

**BAŞKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI
ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ
TEZLİ YÜKSEK LİSANS PROGRAMI**

**İNSAN HAREKETLERİNİN KINECT SENSÖR KULLANILARAK
SINIFLANDIRILMASI**

HAZIRLAYAN

BÜŞRA AÇIŞ

YÜKSEK LİSANS TEZİ

ANKARA- 2022

**BAŞKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI
ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ
TEZLİ YÜKSEK LİSANS PROGRAMI**

**İNSAN HAREKETLERİNİN KINECT SENSÖR KULLANILARAK
SINIFLANDIRILMASI**

HAZIRLAYAN

BÜŞRA AÇIŞ

YÜKSEK LİSANS TEZİ

TEZ DANIŞMANI

DR. ÖĞR. ÜYESİ SELDA GÜNEY

ANKARA – 2022

BAŞKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Elektrik Elektronik Mühendisliği Anabilim Dalı Elektrik Elektronik Mühendisliği Tezli Yüksek Lisans Programı çerçevesinde Büşra AÇIŞ tarafından hazırlanan bu çalışma, aşağıdaki jüri tarafından Yüksek Lisans Tezi olarak kabul edilmiştir.

Tez Savunma Tarihi: 29 / 06 / 2022

Tez Adı: İnsan Hareketlerinin Kinect Sensör Kullanılarak Sınıflandırılması

Tez Jüri Üyeleri (Unvanı, Adı- Soyadı, Kurumu)

İmza

Prof. Dr. Hasan Şakir BİLGE (Gazi Üniversitesi)

.....

Prof. Dr. Hamit ERDEM (Başkent Üniversitesi)

.....

Dr. Öğr. Üyesi Selda GÜNEY (Başkent Üniversitesi)

.....

ONAY

Prof. Dr. Ömer Faruk ELALDI
Fen Bilimleri Enstitüsü Müdürü
Tarih: / / 2022

BAŞKENT ÜNİVERSİTESİ
FEN BİLİMLER ENSTİTÜSÜ
YÜKSEK LİSANS TEZ ÇALIŞMASI ORJİNALLİK RAPORU

Tarih: 29 / 06 / 2022

Öğrencinin Adı, Soyadı : Büşra AÇIŞ

Öğrencinin Numarası : 21910158

Anabilim Dalı : Elektrik-Elektronik Mühendisliği Anabilim Dalı

Programı : Elektrik-Elektronik Mühendisliği Tezli Yüksek Lisans Programı

Danışmanın Unvanı/Adı, Soyadı : Dr. Öğr. Üyesi Selda GÜNEY

Tez Başlığı : İnsan Hareketlerinin Kinect Sensör Kullanılarak Sınıflandırılması

Yukarıda başlığı belirtilen Yüksek Lisans tez çalışmamın; Giriş, Ana Bölümler ve Sonuç Bölümünden oluşan, toplam 90 sayfalık kısmına ilişkin, 29 / 06 / 2022 tarihinde şahsım/tez danışmanım tarafından Turnitin adlı intihal tespit programından aşağıda belirtilen filtrelemeler uygulanarak alınmış olan orijinallik raporuna göre, tezimin benzerlik oranı %4'tür. Uygulanan filtrelemeler:

1. Kaynakça hariç
2. Alıntılar hariç
3. Beş (5) kelimedenden daha az örtüşme içeren metin kısımları hariç

“Başkent Üniversitesi Enstitüleri Tez Çalışması Orijinallik Raporu Alınması ve Kullanılması Usul ve Esaslarını” inceledim ve bu uygulama esaslarında belirtilen azami benzerlik oranlarına tez çalışmamın herhangi bir intihal içermediğini; aksinin tespit edileceği muhtemel durumda doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi ve yukarıda vermiş olduğum bilgilerin doğru olduğunu beyan ederim.

Öğrenci İmzası:

ONAY

Tarih: 29 / 06 / 2022

Öğrenci Danışmanı Unvan, Adı, Soyadı, İmza:

Sevgili kedim Mira'ya ithafen...

Büşra AÇIŞ

Ankara - 2022

TEŞEKKÜR

Sayın tez danışmanım Dr. Öğr. Üyesi Selda GÜNEY'e, çalışmanın sonuca ulaştırılmasında ve karşılaşılan güçlüklerin aşılmasında her zaman yardımcı ve yol gösterici olduğu için teşekkür ederim.

Sayın Dr. Begüm CANGÖZ'e, bu çalışmada gösterdiği tüm yardım ve destekleri için teşekkür ederim.

Sevgili eşim İsmail AÇIŞ'a desteklerinden ve anlayışından dolayı teşekkür ederim.

Sevgili annem Birgül BÖLÜK ve sevgili babam Soner BÖLÜK'e üniversitesi hayatım boyunca bana verdikleri emek ve destekleri için sonsuz teşekkürlerimi iletiyorum.

Tüm başarılar sizlerle anlamlı.

ÖZET

Büşra AÇIŞ

İNSAN HAREKETLERİNİN KINECT SENSÖR KULLANILARAK SINIFLANDIRILMASI

Başkent Üniversitesi Fen Bilimleri Enstitüsü

Elektrik Elektronik Mühendisliği Anabilim Dalı

2022

Son yıllarda sağlık ve güvenlik gibi birçok alanda insan hareketlerini sınıflandırmaya yönelik çalışmalar yapılmaktadır. Son yıllarda kullanım alanı daha çok gelişen görüntü işleme ve sınıflandırma algoritmaları bu alanda da kullanılmaya başlanmıştır. Bu sınıflandırma algoritmalarının öncül yöntemleri makine öğrenmesi ve derin öğrenmedir.

Bu çalışmada, Kinect sensör kullanılarak elde edilen veri seti üzerinde insan hareketlerinin sınıflandırılması yapılmıştır. Kullanılan veri seti içerisinde gerçek zamanlı insan duruş bilgilerinin ve görüntülerinin bulunduğu literatürde hazır halde bulunan CAD60 veri setidir. Bu veri setinde, farklı insanların farklı hareketlerini/duruşlarını içeren veriler bulunmaktadır. Bu çalışma kapsamında, MATLAB uygulaması kullanılarak derin öğrenme tabanlı ve makine öğrenme tabanlı yöntemlerle insan hareketlerini sınıflandırma yapılmıştır. Ham veri seti üzerinde, geriye doğru öznitelik seçme (BFS) ile elde edilen verilerle ve Uzun Kısa Süreli Bellek (Long Short Term Memory – LSTM) ile öznitelik çıkarma ile elde edilen verilere makine öğrenmesi tabanlı metotlar uygulanarak sınıflandırma çalışmaları yapılmıştır. Bunu yanı sıra LSTM, Evrişimsel Sinir Ağları (Convolutional Neural Network, CNN) ile sıfırdan öğrenme (learn from scratch) ve öğrenme aktarımı (transfer learning) yöntemi ile insan hareketleri sınıflandırılmıştır. Bu altı yöntemin sonunda elde edilen başarı değerleri birbirleri ile kıyaslanmıştır ve maksimum başarı değeri LSTM ile öznitelik çıkarma yönteminde elde edilmiştir.

ANAHTAR KELİMELER: Sınıflandırma, Evrişimli Sinir Ağları, Derin Öğrenme, İnsan Hareket Tanıma, Makine Öğrenmesi, Kinect Sensor.

ABSTRACT

Büşra AÇIŞ

CLASSIFICATION OF HUMAN MOVEMENTS BY USING KINECT SENSOR

Baskent University Institute of Science and Engineering

Department of Electrical and Electronics Engineering

2022

In recent years, studies have been carried out to classify human movements in many areas such as health and safety. Image processing and classification algorithms, which have been used more in recent years, have started to be used in this field as well. The predecessor methods of these classification algorithms are machine learning and deep learning.

In this study, the classification of human movements was made on the data set obtained using the Kinect sensor. It is the CAD60 dataset, which is available in the literature, which contains real-time human posture information and images in the data set used. In this dataset, there are data containing different movements/stances of different people. Within the scope of this study, human movements were classified using deep learning-based and machine learning-based methods using MATLAB application. Classification studies were carried out by applying machine learning-based methods to the data obtained by backward feature selection (BFS) and feature extraction with Long Short Term Memory (LSTM) on the raw data set. In addition, human movements are classified with LSTM, Convolutional Neural Networks (CNN), learn from scratch and transfer learning method. The success values obtained at the end of these six methods were compared with each other and the maximum success value was obtained in the feature extraction method with LSTM.

KEYWORDS: Classification, Convolutional Neural Networks, Deep Learning, Human Activity Recognition, Machine Learning, Kinect Sensor.

İÇİNDEKİLER

	Sayfa
TEŞEKKÜR.....	i
ÖZET	ii
ABSTRACT	iii
İÇİNDEKİLER.....	iv
TABLolar LİSTESİ.....	vii
ŞEKİLLER LİSTESİ.....	ix
SİMGELER VE KISALTMALAR LİSTESİ.....	xii
1. GİRİŞ.....	1
1.1. Çalışmanın Konusu	1
1.2. Çalışmanın Amacı ve Önemi.....	2
1.3. Genel Bakış	4
1.4. Literatür Taraması	5
2. MATERYAL VE METOTLAR.....	16
2.1. Kinect Sensör	16
2.2. Veri Seti.....	20
2.3. Makine Öğrenmesi	23
2.3.1. Makine öğrenmesinde sınıflandırma	23
2.3.2. Karar ağaçları	24
2.3.3. Doğrusal ayrık analiz	25
2.3.4. Destek vektör makineleri.....	26

2.3.5.	K-En yakın komşu algoritması	27
2.3.6.	Topluluk sınıflandırıcı	28
2.3.7.	Öznitelik seçme	28
2.3.7.1.	Geriye doğru öznitelik seçimi (BFS).....	30
2.3.7.2.	İleriye doğru öznitelik seçimi (FFS)	30
2.4.	Derin Öğrenme	31
2.4.1.	Sıfırdan öğrenme (Learn from scratch)	35
2.4.2.	Öğrenme aktarımı (Transfer learning) tabanlı sınıflandırma.....	35
2.4.2.1.	AlexNet	38
2.4.2.2.	ResNet.....	39
2.4.2.3.	DenseNet-201	40
2.4.2.4.	Inception v3.....	42
2.4.2.5.	Xception.....	42
2.4.3.	Yinelemeli sinir ağı (RNN)	43
2.4.4.	LSTM.....	44
2.5.	Sınıflandırma Performansını Ölçmede Kullanılan Teknikler	46
3.	YAPILAN ÇALIŞMALAR VE SONUÇLAR	49
3.1.	Makine Öğrenmesi Yöntemleri.....	49
3.2.	Geriye Doğru Öznitelik Seçme (BFS).....	57
3.3.	LSTM.....	62
3.4.	LSTM ile Öznitelik Çıkarma	68

3.5.	Sıfırdan Öğrenme (Learn from Scratch)	75
3.6.	Öğrenme Aktarımı (Transfer Learning).....	77
3.7.	Elde Edilen Sonuçlar.....	88
4.	SONUÇ VE ÖNERİLER.....	91
	KAYNAKLAR.....	92



TABLÖLAR LİSTESİ

	Sayfa
Tablo 2.1. Veri seti gözlem değerleri.....	21
Tablo 2.2. Etiket değerleri.....	21
Tablo 3.1 Makine öğrenmesi yönteminde veri seti ayrımları için test başarıları (%).....	50
Tablo 3.2. Makine öğrenmesi yönteminde veri seti ayrımları için maksimum sınıflandırma başarıları	56
Tablo 3.3. Çıkarılan öznitelik bilgisi.....	59
Tablo 3.4. Eğitim seti:134 için BFS ile elde edilen sınıflandırma sonuçları	60
Tablo 3.5. BFS sonucunda veri seti ayrımları için elde edilen başarı değerleri.....	61
Tablo 3.6. Eğitim seti:134 ve test seti:2 için LSTM sonuçları.....	63
Tablo 3.7. LSTM Yönteminde eğitim seti:123, test seti:4 için sınıflandırma başarıları	65
Tablo 3.8. LSTM yönteminde eğitim seti:124, test seti:3 için sınıflandırma başarıları	66
Tablo 3.9. LSTM yönteminde eğitim seti:134, test seti:2 için sınıflandırma başarıları	66
Tablo 3.10. LSTM yönteminde eğitim seti:234, test seti:1 için sınıflandırma başarıları	67
Tablo 3.11. LSTM için ortalama sınıflandırma sonuçları.....	67
Tablo 3.12. LSTM ile öznitelik çıkarma yönteminde veri seti ayrımları için sınıflandırma başarıları (%)	68
Tablo 3.13. LSTM ile öznitelik çıkarma için ortalama başarı değerleri	74
Tablo 3.14. RGB görüntü sayıları	75
Tablo 3.15. Sıfırdan öğrenme ile sınıflandırma başarıları	76
Tablo 3.16. Ağ mimarilerine göre sınıflandırma başarıları.....	78

Tablo 3.17. Öğrenme aktarımı yönteminde veri seti ayrımları için Resnet50 ile elde edilen sınıflandırma başarıları (%).....	78
Tablo 3.18. Öğrenme aktarımı yönteminde veri seti ayrımları için DenseNet-201 ile elde edilen sınıflandırma başarıları (%).....	81
Tablo 3.19. Öğrenme aktarımı yönteminde veri seti ayrımları için Xception ile elde edilen sınıflandırma başarıları (%).....	83
Tablo 3.20. Resnet50 ağ mimarisine göre maksimum sınıflandırma başarıları.....	85
Tablo 3.21. DenseNet-201 ağ mimarisine göre maksimum sınıflandırma başarıları.....	86
Tablo 3.22. Xception ağ mimarisine göre maksimum sınıflandırma başarıları	87
Tablo 3.23. Sınıflandırma sonuçları.....	90

ŞEKİLLER LİSTESİ

Sayfa

Şekil 2.1. Kinect sensör [46].	16
Şekil 2.2. Kinect sensörden elde edilen RGB görüntü [13].	17
Şekil 2.3. Kinect sensörden elde edilen derinlik görüntüsü [13].	17
Şekil 2.4. Kinect sensör iskelet görüntüsü [13].	18
Şekil 2.5. Kinect sensör eklem bilgileri [50].	19
Şekil 2.6. Kinect sensör yerleşimi [16].	20
Şekil 2.7. Data seti ile ilgili görüntüler [13].	20
Şekil 2.8. Veri setinde kullanılan eklemler.	22
Şekil 2.9. ML sınıflandırma örneği.	24
Şekil 2.10. Karar ağaçları örneği [2].	25
Şekil 2.11. (a) LDA'dan önce (b) LDA'dan sonra.	25
Şekil 2.12. SVM örneği [66].	26
Şekil 2.13. K-NN algoritması [69].	27
Şekil 2.14. Öznitelik çıkarımı ile makine öğrenmesi [78].	29
Şekil 2.15. Derin öğrenme mimarisi [86].	31
Şekil 2.16. Klasik bir derin öğrenme modeli katmanları [87].	31
Şekil 2.17. LeNet mimarisi [88].	32
Şekil 2.18. Evrişim işleminin çizimi [97].	33
Şekil 2.19. 2x2 Filtre kullanan maksimum havuz işlemi.	34

Şekil 2.20. Tam bağlantılı katman çizimi [102].	34
Şekil 2.21. Geleneksek makine öğrenmesi ve öğrenme aktarımı yöntemleri [106].	36
Şekil 2.22. Sıfırdan öğrenme [107].	36
Şekil 2.23. Tam bağlaşımlı katman ile öğrenme aktarımı [107].	37
Şekil 2.24. Destek vektör makineleri ile öğrenme aktarımı [107].	37
Şekil 2.25. AlexNet mimarisi [108].	38
Şekil 2.26. Sinir ağlarında kullanılan aktivasyon fonksiyonları ve türevleri. a) Basamak fonksiyonu, b) Doğrusal fonksiyon, c) Sigmoid fonksiyonu, d) tanh fonksiyonu, e) ReLU fonksiyonu ve f) Sızıntı ReLU fonksiyonu [111].	39
Şekil 2.27. ResNet mimarisindeki artık blok yapısı [107].	40
Şekil 2.28. DenseNet mimarisinin yapı katmanları [115].	41
Şekil 2.29. DenseNet mimarisi [115].	41
Şekil 2.30. Inception v3 mimarisi [69].	42
Şekil 2.31. Xception mimarisi [118].	43
Şekil 2.32. RNN mimarisi [121].	43
Şekil 2.33. (a) RNN mimarisi, (b) LSTM mimarisi [124].	44
Şekil 2.34. LSTM mimarisi [128].	45
Şekil 2.35. Karmaşıklık matrisi [66].	47
Şekil 3.1. Yapılan çalışmaları özetleyen akış diyagramı.	49
Şekil 3.2. Eğitim seti: 123 için en yüksek başarı değerinde karışıklık matrisi.	52
Şekil 3.3. Eğitim seti: 124 için en yüksek başarı değerinde karışıklık matrisi.	53
Şekil 3.4. Eğitim seti: 134 için en yüksek başarı değerinde karışıklık matrisi.	54

Şekil 3.5. Eğitim seti: 234 için en yüksek başarı değerinde karışıklık matrisi.	55
Şekil 3.6. Veri seti ile ilgili örnek görüntüler [13].	57
Şekil 3.7. BFS sonucunda elde edilen sınıflandırma başarısı.	58
Şekil 3.8. LSTM için karışıklık matrisi.	64
Şekil 3.9. LSTM ile öznitelik çıkarma yönteminde eğitim seti:123 ve test seti:4 için karışıklık matrisi.	71
Şekil 3.10. LSTM ile öznitelik çıkarma yönteminde eğitim seti:124 ve test seti:3 için karışıklık matrisi.	71
Şekil 3.11. LSTM ile öznitelik çıkarma yönteminde eğitim seti:134 ve test seti:2 için karışıklık matrisi.	72
Şekil 3.12. LSTM ile öznitelik çıkarma yönteminde eğitim seti:234 ve test seti:1 için karışıklık matrisi.	72
Şekil 3.13. LSTM ile öznitelik çıkarma yönteminin akış şeması.	90

SİMGELER VE KISALTMALAR LİSTESİ

BFS	Backward Feature Selection (Geriye Doğru Öznitelik Seçimi)
CONF	Boole Güven Değeri
CMOS	Complementary Metal Oxide Semiconductor; (Bütünleyici Metal Oksit Yarı İletken)
CNN	Convolutional Neural Network (Evrişimsel Sinir Ağı)
DL	Deep Learning (Derin Öğrenme)
DNN	Deep Neural Network
FFS	Forward Feature Selection (İleriye Doğru Öznitelik Seçimi)
FN	False Negative
FP	False Positive
Hz	Hertz
KNN	K-Nearest Neighbors (K-En Yakın Komşu Algoritması)
LDA	Linear Discriminant Analysis (Doğrusal Ayırık Analiz)
LSTM	Long Short-Term Memory (Uzun Kısa Süreli Bellek)
LVQ	Learning Vector Quantization (Öğrenme Vektör Nicemleme)
MEMM	Maksimum Entropi Markov Modeli
ML	Machine Learning (Makine Öğrenmesi)
NITE	Natural Interface Middleware (Doğal Arayüz Ara Yazılımı)
PLSTM	Partaware Long Short-Term Memory
PSD	Power Spectral Density (Güç Spektral Yoğunluğu)
ReLu	Rectified Linear Unit (Düzleştirilmiş Doğrusal Birim)
ResNet	Residual Network
RGB	Red Green Blue (Kırmızı, Yeşil, Mavi)
RNN	Recurrent Neural Network (Yinelemeli Sinir Ağı)
RULA	Rapid Upper Limb Assessment (Hızlı Üst Ekstremité Değerlendirmesi)
SVM	Support Vector Machine (Destek Vektör Makineleri)
TN	True Negative
TP	True Positive
YSA	Yapay Sinir Ağları

1. GİRİŞ

1.1. Çalışmanın Konusu

İnsanlar, doğumdan ölüme kadar olan bütün zamanlarında deneyimledikleri her olay için bir takım tecrübe elde edinirler. Örneğin, hepimizin bildiği bir atasözünü ele alalım, “Çocuklar düşse kalka büyür”. Yürümenin ilk aşaması emeklemektir. Daha sonra çocuklar ayakta durarak ilk adımlarını atmaya başlarlar. Bu aşamada ise çocuklar sık sık yere düşerler. Ama bu düşme durumlarında dengelerini daha iyi sağlamayı öğrenirler. Yetişkin bireyler de günlük hayatlarında bir hata yaptıklarında genellikle bu hata üzerinden kendilerine bir ders çıkartırlar ve bu hatayı düzeltmeye çalışırlar. Bilgisayarların doğasında ise hatalardan ders çıkartmak gibi bir durum söz konusu değildir. Bilgisayarların temel programları sadece kendisine tanımlanan programın görevini gerçekleştirmektedir, bu görevin sonucunda elde edilecek sonucun ne olduğu ile ilgilenmemektedirler. Bu sebeple, temel bilgisayar problemleri ile bir konu üzerinde bir iyileştirme veya geliştirme yapma durumu söz konusu olamaz. Bu durum kapsamında, verilerden öğrenebilen ve performansını geliştirebilen programlar geliştirilmiştir. Bu programlara örnek olarak makine öğrenmesi ve derin ağlar verilebilir.

Geleneksel bilgisayar programlarının gelişen teknolojiye ayak uyduramaması ile kendi kendine öğrenen bilgisayar programları geliştirilmeye başlanmıştır. Bu konudaki ilk çalışma, dama oyunlarını daha iyi oynayan bir program geliştirmek amacı ile 1952 yılında A. Samuel tarafından yapılmıştır [1]. Bu çalışma geliştirilerek 1967’de yeni verileri bilinen veriler ile karşılaştırarak aralarındaki benzerlikleri bularak verilerdeki örüntüleri tespit edebilen ilk örüntü programı elde edilmiştir [2]. Makine öğrenmesi ise 1990’lı yıllardan bu yana verilerden öğrenerek modellemeye olanak sağlamak için veri madenciliği, uyarlanabilir yazılı sistemler, metin ve dil öğrenme alanlarında yaygın bir şekilde kullanılmaktadır [2]. Örnek olarak: bir e- ticaret mağazasının müşterileri ile ilgili verileri toplayan ve bu bilgilerden müşterilerine kişiselleştirilmiş reklamlar oluşturan uygulama bir nevi yapay zeka sayılmaktadır. Makine öğrenmesi yöntemleri kullanılarak çeşitli işlemler yapılabilmektedir: sınıflandırma, veri setindeki değişkenler arasındaki ilişkinin anlaşılması, görüntü işleme, kümeleme ve tahminleme bunların başında gelmektedir [3]. Makine öğrenmesi yöntemleri

hayatımızın çeşitli alanlarında kullanılmaktadır: sağlık, mühendislik, ekonomi, güvenlik, meteoroloji, müzik ve sinema vs [4].

Yapay sinir ağları (YSA), insanların biyolojik özelliklerinden faydalanarak geliştirilmiş olan bir tür bilgisayar sistemidir. YSA, insan beyninde bulunan sinir ağlarının çalışma şekline göre faydalanılarak, öğrenme sürecinin matematiksel olarak ifade edilmesi ile ortaya çıkmıştır. Bu sistemde öğrenme işlemi, veri setinde var olan bilgilerin kullanılmasıyla gerçekleştirilmektedir. Öğrenme işlemini tamamlanmasının ardından, oluşturulan modelden yeni bilgiler türetilmektedir. YSA, bahsedilen bu durumları, kendi başına otomatik olarak gerçekleştirmektedir [5], [6].

Derin öğrenme (DL), 2010 yılından bu yana kullanım alanı genişleyen bir makine öğrenmesi türüdür [7]. DL, dijital sistemlerin yapılandırılmamış veya etiketlenmemiş verilere dayalı olarak öğrenmesini sağlamak üzere yapay sinir ağlarını kullanmaktadır [8]. Makine öğrenmesi, veri setini inceleyerek öğrenme işlemini gerçekleştirir. Ancak, derin öğrenme veri setini kullanarak bilgi edinir ve bu bilgileri kullanarak tepki verir [9]. DL, büyük ve karmaşık veri setleri üzerinde daha yüksek doğruluk ile çalışmaktadır [10]. DL'in çok geniş bir kullanım alanı vardır: savunma, güvenlik, perakende, sağlık hizmetleri, taşıma, üretim, teknoloji vs. Örneğin, akıllı ev sistemlerinde kullanılan cihazların sesli komut ile çalışmasını sağlayan sistem derin öğrenmedir [11].

Makine öğrenmesi ve derin öğrenme yöntemleri aynı alanlar üzerinde çalışsa da çalışma prensipleri ve gösterdikleri performanslar birbirinden farklı olmaktadır. Bu iki yöntemin kendi içerisinde farklı mimarileri mevcuttur. Bu çalışmada, insan hareketlerinin sınıflandırılmasında hangi yöntem ile daha yüksek doğruluk elde edildiğini bulmak için Kinect sensör ile elde edilen veri seti üzerinde makine öğrenmesi ve derin öğrenme tabanlı yöntemler kullanılarak sınıflandırma başarıları elde edilmiştir.

1.2. Çalışmanın Amacı ve Önemi

İnsanların sahip olduğu duruş stilleri kişinin sağlığı açısından büyük önem taşımaktadır. Farkında olmadan yapılan duruş hataları gerek şimdi gerek ilerleyen dönemde ciddi sağlık sorunlarına yol açmaktadır. İnsanların çoğu zamanının büyük bir kısmını iş yerinde geçirmektedir. Bu durumda iş yerinde yapılan duruşlar daha kritik olmaktadır.

Örneğin, bir fabrika çalışanı yaptığı işi uzun bir süre kambur bir duruş ile gerçekleştirirse bir süre sonra bel ağrıları problemi ortaya çıkabilmektedir veya bir yazılım mühendisinin bilgisayar başında sürekli olarak kambur bir şekilde durması kas spazmına neden olur. Bu gibi duruş bozukluklarından kaynaklı sağlık sorunları fabrika, okul, hastane, banka gibi sayısız iş yerinde gerçekleşebilir. Duruş bozuklukları insanın yaşamını olumsuz bir yönde etkilemektedir. Hatta, hayatının işleyişine ve çalışmasına engel olabilmektedir. Bu durumlar göz önüne alınarak; gelişmiş bir sistem ile çalışanların duruşu belirlenip, yanlış duruş ile ilgili olarak bir uyarı gerçekleştirilebilir. Bu örneklerden de yola çıkarak özellikle sağlık alanında insanların duruşunun doğru ve etkili bir şekilde sınıflandırılması ile ilgili olarak birçok faydalı çalışma gerçekleştiriliyor ve gerçekleştirilmeye devam edecektir.

Bu tez çalışmasında, Kinect sensör ile elde edilen veri seti kullanılarak insan hareketlerini sınıflandıran bir çalışma yapılması amaçlanmıştır. Bu amaç doğrultusunda yapılan tüm çalışmalar maddeler halinde aşağıda verilmiştir:

1. Veri seti üzerinde herhangi bir işlem uygulanmadan makine öğrenmesi yöntemleri ile sınıflandırma çalışması yapılmış ve ham veri başarısı elde edilmiştir.
2. Ham veri seti üzerinde elde edilen başarılar, öznitelik seçimi yöntemi uygulanarak artırılmaya çalışılmıştır.
3. Derin öğrenme tabanlı olan LSTM mimarisi kullanılarak sınıflandırma başarı elde edilmiş ve mimarideki farklı parametrelerin en iyilenerek sınıflandırma başarısının artırılması amaçlanmıştır.
4. LSTM mimarisi kullanılarak veri seti üzerinde öznitelik çıkarma işlemi uygulanmıştır ve elde edilen öznitelik matrisi kullanılarak makine öğrenmesi yöntemleri ile sınıflandırma yapılmıştır.
5. Öğrenim aktarımı kullanılarak farklı derin öğrenme ağları kullanılarak sınıflandırma başarısını elde edilmiştir.
6. Öğrenme aktarımı ile sıfırdan öğrenme yöntemlerinin karşılaştırabilmesi adına, sıfırdan oluşturulan bir Evrimsel Sinir Ağı ile de sınıflandırma performansı ölçülmüştür.

Başarı iyileştirme denemeleri ile toplamda altı farklı yöntem kullanılarak veri seti üzerinde sınıflandırma başarısı elde edilmesi ve bu sınıflandırıcıların başarıları birbirleri ile karşılaştırılarak bu veri seti üzerinde kullanılabilecek olan en başarılı sınıflandırıcının belirlenmesi amaçlanmaktadır.

Tezde kullanılan veri seti üzerinde yukarıda bahsedilen yöntemlerle sınıflandırma başarısı elde edilmiştir. Elde edilen başarı değerleri iyileştirilmeye çalışılmıştır. Bu amaçla bahsedilen yöntem 1’de elde edilen başarı değerleri öznitelik çıkarma yöntemi ile arttırılmaya çalışılmıştır. Yöntem 3’te parametre değerleri değiştirilerek en yüksek başarı değeri elde edilmesi amaçlanmıştır. Bu yöntemden elde edilen sınıflandırma başarısının arttırılması için yöntem 4’te öznitelik çıkartma uygulanmıştır.

1.3. Genel Bakış

Bu tez çalışması 4 bölümden oluşmaktadır. Tez çalışmasında, CAD60 olarak bilinen Kinect sensör ile elde edilmiş hazır veri seti üzerinde makine öğrenmesi ve derin öğrenme mimarileri kullanılarak insan duruşlarının sınıflandırma başarısının elde edilmesi ve karşılaştırılması yapılmıştır.

Tez çalışmasının ilk bölümünde gerçekleştirilen çalışmanın konusu, amacı, önemi ve tez çalışmasının mimarisi sunulmaktadır. Ayrıca, gerçekleştirilen çalışma ile ilgili olarak geçmişte yapılmış çalışmalar üzerinde bir literatür taraması yapılmış ve benzer nitelikteki çalışmaların bir özeti bu bölümde verilmiştir.

Tez çalışmasının ikinci bölümünde, tez çalışmasında kullanılan yöntem ve mimarilerden detaylı olarak bahsedilmiştir. Ayrıca, bu bölümde tez çalışmasında kullanılan veri seti ve veri setinin elde edildiği Kinect sensör ile ilgili bilgilere yer verilmiştir.

Tez çalışmasının üçüncü bölümünde ise transfer öğrenmesi amacı ile kullanılan öğrenme aktarımı tabanlı yöntemler, derin öğrenme tabanlı olan LSTM, Makine öğrenmesi ve öznitelik çıkarma ile ilgili olarak gerçekleştirilen çalışmalar ve sonuçları verilmiştir.

Tezin son ve dördüncü bölümünde, yapılan çalışmalar genel olarak özetlenmiştir. Çalışmada elde edilen sonuçlar yorumlanmıştır ve gelecek çalışmalar ile ilgili olarak önerilerde bulunulmuştur.

1.4. Literatür Taraması

Günümüzde, insan aktivitesini tespit etmek ve tanımlayabilmek çok geniş bir kullanım alanına sahip olduğu için bu konular üzerinde çok fazla çalışmalar ve yenilikler geliştirilmiştir. Bu konu üzerinde yapılan çalışmalarda en çok kullanılan sensörlerden birisi ucuz ve kullanımının kolay olmasından dolayı Kinect sensördür [12]. Jaeyong Sung ve arkadaşlarının yaptığı çalışmada Kinect sensör ile elde edilen RGBD görüntüleri üzerinden insan etkinliği tespiti yapılmıştır. Günlük aktivitelerin gerçekleştirildiği ortamlar, yapılandırılmış ve yapılandırılmamış ortamlar olarak tanımlanmaktadır. Yapılandırılmamış ortam, dağınık ev veya ofis olarak tanımlanmaktadır. Yapılandırılmış ortam ise bunun tam tersidir. Ortamın dağınık olması insan hareketlerini tespit etmekte zorluk çıkarmaktadır. Bu sebeple, bu çalışmada yapılandırılmamış ortamda bulunan hareketlerin tespiti gerçekleştirilmiştir. Bu çalışmada gerçekleştirilen algoritma, bir kişinin aktivitesini bir dizi alt aktiviteden oluşan bir hiyerarşik MEMM göre gerçekleştirmektedir. Algoritmanın farklı ortamlarda elde ettiği başarı karşılaştırılmak istendiği için beş farklı ortamda etkinlik tanıma testi gerçekleştirilmiştir. Algoritma başarısı, kesinlik ve geri çağırma ölçütleri ile değerlendirilmiştir. Ayrıca, algortmada daha önce tanımlanmış ve hiç tanımlanmamış (yeni bir kişi) olarak iki farklı başarı sonucu elde edilmiştir. Algoritma sonucunda, “görmüş” ve “yeni kişi” durumlarında sırasıyla %84,7/83,2 ve %67,9/55,5 hassasiyet/hatırlama ölçümü ile tespit ve sınıflandırma sonuçları elde edilmiştir [13].

Tao Wang ve arkadaşlarının yaptığı çalışmadaysa Kinect sensör kullanılarak otomatik ve gerçek zamanlı depresyon tespiti için bir yürüyüş analizi gerçekleştirilmiştir. Bu çalışmada, yürüyüş verileri için depresyonu tespit etmede kaynaşmış özellikler üzerinde çalışmak için bir hız modeli geliştirilmiştir. Modelin ilk aşamasında yürüyüş anormalliklerini tespit etmek için sözde bir hız modeli oluşturulmuştur. Daha sonra depresyon tespitini daha iyi desteklemek için kullanılacak olan frekans alanı özelliğini çıkarmak için PSD analizi yapılmıştır. Çalışma 95 insan üstünde gerçekleştirilmiştir ve geliştirilen modelin sınıflandırma doğruluğu %93,75'tir. Elde edilen sonuçlar doğrultusunda önerilen yöntemin sınıflandırma başarısı, diğer birkaç yönteme kıyasla daha yüksek olduğu

gözlemlenmiştir. Ayrıca, bu yöntem ile bireysel farklılıkların etkisinin de azaldığı gözlemlenmiştir [14].

S.A. Abdul Shukor ve arkadaşlarının gerçekleştirdiği çalışmada Kinect sensör kullanılarak insan düşüşü tespit edilmiştir. Bu çalışmada, zemin düzlemi ile her eklem mesafesini hesaplayarak çalışan iskelet tabanlı bir yöntem kullanılmıştır. Eklemler, iskelet uzay sistemi kullanılarak tespit edilmektedir. Algoritmanın çalışma mantığına göre bir düşüşün algılanabilmesi için zeminin gözükmemesi ve algılanan Y koordinatı değerinin belirlenen eşik değerinden küçük olması gerekmektedir. Algoritma ve sensör arasında daha az iletişim hatası olması için Visual Studio yazılımında C # ile Kinect sensörü için ara yazılım olarak Window SDK kullanılmıştır. Geliştirilen sistem ideal koşullarda doğru sonuçları vermiştir [15].

Baoqi Yao ve arkadaşlarının yapmış olduğu çalışmada ise sanal basketbol eğitimi için üçlü Kinect sensörleri kullanan insan aktivitesi tanıma sistemi geliştirilmiştir. İnsan hareketlerini sınıflandırmak için PLSTM modeli kullanılmıştır. Birden fazla Kinect sensör kullanıldığı için algılanan çoklu iskelet verilerini entegre etmek için ağırlıklı bir entegrasyon yönetimi kullanılmıştır. İnsan aktivitelerini tanımlamak için, özellik olarak ortak hızları, açıları ve açısal hızları kullanılmıştır. Bu özellikler, üç boyutlu iskelet verisinden çıkarılmıştır ve vücut kısımlarına göre beş parçaya ayrılmıştır. Ayrıca, sanal gerçeklik basketbol uygulaması ile sistem denenmiştir ve deney sonuçlarına göre önerilen sistemin kullanılabilir olduğu gözlemlenmiştir [16].

Timeea Sărătean ve arkadaşlarının gerçekleştirdiği çalışmada ise evde fizyoterapi tedavisi alan bir hastayı izleyebilen kişisel asistana dayanan otonom bir sistem geliştirilmiştir. Sistem, yapılan egzersizlerin doğru olduğunu ve hasta için doktorlar tarafından belirlenen rehabilitasyon programına uygun olduğunu kontrol etme amacı ile tasarlanmıştır. Geliştirilen mimaride hasta, sistem üzerinden fizyoterapist tarafından kendisine sunulan egzersizlerden birisini seçerek yapmaktadır. Hastanın eklem koordinatlarını bulmak için Kinect Sensör kullanılmıştır. Sistem iki farklı uygulamadan oluşturulmuştur: fizyoterapistler için egzersizleri tanımlayabileceği bir web uygulaması ve hastalar için egzersizleri gerçekleştireceği Kinect tabanlı masaüstü uygulamasıdır. Hareket tespiti için çabaya dayalı bir parametreleme yöntemi kullanılmıştır. Sistemin doğruluğu on iki farklı egzersiz üzerinde denenmiştir. Sonuçlar, geliştirilen yöntemin büyük bir fayda

sağladığını ve fizyoterapistler tarafından uzaktan fizyoterapi egzersizlerinin yapılabileceği göstermiştir[17].

Alexandre de Queiroz Burle ve arkadaşlarının gerçekleştirdiği çalışmada klinik çalışmalarda kullanılmak üzere Kinect sensörü ile elde edilen yürüyüş verileri üzerinde gerçek zamanlı bir yaklaşım uygulanmıştır. Bu çalışmada, her bir parametrenin gerçek ölçümleriyle karşılaştırmak için 75 farklı yürüyüş örneği kullanılmıştır. Kinect sensörü ile eklem bilgilerine ulaşılmıştır. Sistem, adımları belirleyerek ve vücudun eklem açılarını tanımlayarak çalışmıştır. Elde edilen sonuçlara göre diz ve kalça açıları ile yüksek doğruluk elde edilmiştir. Ancak, ayak açıları sistemin çalışmasını olumsuz yönde etkilemiştir [18].

Zhenda Xu ve arkadaşlarının gerçekleştirdiği çalışma ise sağlık alanındadır. Omurga taraması klinik tanıya veya X-ışını muayenesine dayanmaktadır. Ancak, bu alanda bir bilgi eksikliği olmasından dolayı ergenlik çağındaki gençlerde omurga deformasyon sorunlarını ailelerinin fark etmesine engel olmaktadır. Bu nedenle bu çalışmada, omurga deformasyonunu değerlendirmek için kullanılacak bir yöntem geliştirilmiştir. Geliştirilen yöntem, omurganın üç boyutlu şeklini hızlı bir şekilde yeniden yapılandırmak için düşük maliyetli, kullanımı kolay, radyasyon içermeyen yapıda ve yüksek doğruluk ile çalışan bir yöntemdir. Üç boyutlu omurgayı yeniden yapılandırmak için kullanılacak olan doğru modeli oluşturmak için Kinect sensör kullanılmıştır. Kinect sensör tarafından toplanan derinlik görüntüleri üç boyutlu nokta bulutlarına dönüştürülmektedir. Daha sonra, anatomik dönüm noktalarının ve spinöz süreçler çizgisinin özellikleri sınıflandırılmıştır. Ardından, deforme olmuş omurgayı yeniden yapılandırmak için spinöz süreçler çizgisi ve omurga çizgisinin korelasyon modeli oluşturulmuştur. Yöntem, deneysel olarak test edilmiştir ve ortalama RMS farkı yaklaşık olarak 8 mm olarak hesaplanmıştır. Önerilen bu yöntemin performansı, literatürde bulunan diğer yöntemler ile karşılaştırıldığında daha yüksek gelmiştir [19].

Jyothsna Kondragunta ve Gangolf Hirtz'in gerçekleştirdiği çalışmada ise yaşlılar üzerinde demansın erken saptanmasında üç boyutlu insan duruş tahmini kullanılarak yürüyüş parametrelerini tahmin etmeye çalışılmıştır. Demans, unutkanlığın belirgin olduğu hastalıklara verilen genel bir isimdir. Demanstan etkilenen bir insanın yürüyüşünde belirgin bir değişiklik gözlemlenmektedir. Bu yürüyüş farklılığından faydalanarak bir insanda bulunan demansı saptamak için bazı yürüyüş parametreleri kullanılmıştır. Bu çalışma için

80 yařın üzerindeki birçok yařlı insana ait yryř verileri toplanmıřtır. alıřma grupları e ayrılmıřtır: biliřsel saėlıklı bireyler, znel biliřsel bozukluėu olan bireyler ve olası hafif biliřsel bozukluėu olan kiřiler veya hafif biliřsel bozukluėu olan kiřiler. Yryř verileri Kinect sensor kullanılarak toplanmıřtır. Kinect sensr ile hem RGB grnt hem de derinlik verileri toplanmıřtır. Yryř verileri zerinde  boyutlu insan pozu tahmini ve yryř analizi yapılmıřtır. Bunun sonucunda, bireylerin saėlık durumu analiz edilip, bu analizin sonucunda yrme anormalliklerindeki modeller sınıflandırılmıřtır. Bu sınıflandırma sonucu nceden bunamanın teřhisinde kullanılmıřtır. Elde edilen sonular sistemin alıřabildiėini gstermiřtir [20].

Naveen Kumar Mangal ve Anil Kumar Tiwari yaptıėı alıřmada kas-iskelet bozukluklarının kinematik analizi iin hareket imzası oluřturmayı amalanmıřlardır. Bu ama doėrultusunda, grlty gidermek ve gvde evresinde bulunan verilerin  boyutlu koordinatlarının kalitesini artırmak iin bir filtre nermiřlerdir. Oluřturulan imzalar, klinikte alıřanlara vcut eklem hareketlerinin teřhisi ve izlenmesine yardımcı olmak amaı ile kullanılmıřtır. nerilen filtrenin performansını deėerlendirmek iin klinik olarak bu deėerlendirmeye uygun olan anatomik hareketler gerekleřtirilmiřtir. alıřma kapsamında nerilen filtreden elde edilen hareket aralıėı deėerleri, altın standart gonyometri deėeri ile karřılařtırılarak performans deėerlendirilmesi yapılmıřtır. Elde edilen sonulara gre bu alıřmada nerilen filtreden elde edilen hareket aralıėı deėerlerinin, Kinect ile iskelet eklemi izlemenin bařarısını attırdıėı gzlemlenmiřtir [21].

Shalini Nehra ve Jagdish Lal Raheja, Kinect sensr kullanarak i ortamda bulunan bir kiřinin gnlk aktivitelerini izlemek ve tespit etmek iin kullanılan bir insan aktivitesi tanıma sistemi geliřtirmiřtir. nerilen sistem gerek zamanlı olarak dřme tespiti yapmak zere geliřtirilmiřtir. Bunun yanı sıra, sistem herhangi bir yanlıř aktiviteyi hesaba katmadan hareket halindeyken gerekleřen dřme eylemini veya otururken gerekleřen dřme eylemini tespit edebilmektedir. Bunun iin Kinect sensr kullanılarak derinlik ve iskelet eklem bilgileri elde edilmiřtir. alıřma kapsamında, sistemin doėru alıřması iin referans on eklem kullanılmıřtır. Deneysel sonulara gre sistemin herhangi bir veri seti zerinde yksek doėruluk ile alıřtıėı gzlemlenmiřtir [22].

Tan-Hsu Tan ve arkadaşlarının gerekleřtirdiėi alıřmada yařlı insanların gnlk ve anormal olan aktivitelerini tanımak zere bir insan aktivitesi tespit sistemi geliřtirilmiřtir.

Bu çalışmada RGB aktivite görüntüleri için Kinect sensör ile elde edilen Cornell Etkinlik Veri Kümesi kullanılmıştır. Yaşlıların günlük aktivitelerini tanımlamak ve sınıflandırmak için evrişimli sinir ağı kullanılmıştır. Dört kat çapraz doğrulama deneysel sonuçlarının başarı değeri kesinlik, geri çağırma, özgüllük, doğruluk ve F1 skorunu ile değerlendirilmiştir. Bu başarı değerleri sırası ile %95,5; % 95,6; % 99,8; % 99,6 ve % 95,3'tür. Bu sonuçlara göre önerilen sistemin performansı literatürde bulunan diğer yöntemlere göre daha başarılı olduğunu göstermiştir [23].

Bir başka çalışmada ise Takuya Kamizono ve arkadaşları tarafından dersteyken öğrencilerin durumlarını anlamak ve dersin kalitesini arttırmak için öğrencilerin aktivitelerini tanımlamak üzerine bir çalışma yapılmıştır. Öğrencilerin aktivitelerini tanımlamak için kullanılacak olan veriler Kinect sensör aracılığı ile elde edilmiştir. Dört farklı aktivite üzerinden veriler toplanmıştır: Gözleri kapalı meditasyon yapmak, ekranda yazıları okumak, yüzünü hareket ettirerek uzağa bakmak ve uzak bir noktada oturmak. Veriler üzerinde ML yönteminde K-NN mimarisi uygulanmıştır ve aktiviteleri tanımada sistemin başarısı 0,936 olarak bulunmuştur [24].

Min Li ve arkadaşlarının çalışmasında ise yatış durumundaki bebekler için üç boyutlu poz tahmin yöntemi denenmiştir. Yöntem, Moving Infants In RGB-D açık veri kümesindeki Kinect sensör tarafından elde edilen derinlik verilerindeki ayrıntılı bilgilerde birleştirilmiş olan mevcut iki boyutlu bir insan vücudu kilit noktası algılama yöntemi kullanmıştır. Yöntemin sonucundaki verilere göre tahmini vücut parçasının uzunluğunun ortalama hatası 13,76 mm olarak bulunmuştur. Ayrıca, Doğru Yerleştirilen Parçaların Yüzdesi ve Doğru Anahtar Nokta Yüzdesi bilgilerinin doğruluğu sırasıyla %80,7 ve %86,1 olarak gözlemlenmiştir [25].

Weiyen Ren ve arkadaşlarının gerçekleştirdiği çalışmada ise Kinect sensör ile yirmi farklı insanın yatakta yatarken gösterdiği duruş bilgisi elde edilmiştir. Bu veriler üzerinde bulanık mantık ve makine öğrenimi birleşik algoritması kullanarak insanların duruşunu doğru ve verimli bir şekilde tanıyan bir yöntem uygulanmıştır. Bu çalışmanın amacı, otonom bir yardımcı robotun veya bir başka otomatik sağlık bakım sisteminin kişinin duruşunu değiştirmeye yardımcı olmak veya onu yataktan tekerlekli sandalyeye taşımak gibi bakım hizmetleri sağlamak için yatakta yatan bir hastanın vücut-uzuv duruşunu tanımaya olanak sağlamaktır. Bu çalışma için yeni bir insan yatma pozisyonu içeren veri seti oluşturulmuştur.

Kinect sensör kullanılarak 32 kişiden veri toplanmıştır ve toplamda 19.800 derinlik görüntüsü elde edilmiştir. Kinect sensör ile elde edilmiş derinlik görüntüleri ve iskelet eklem bilgilerine dayanarak, tüm yatma pozisyonları analiz edilmiştir. Bu analizin sonucunda bir dizi bulanık kural elde edilmiştir. Bu aşamadan sonra, duruş tanıma ile ilgili olarak eklem açıları ve uzuv duruşları arasında ilişkisel bir model elde edilmiştir. Yöntem ML yönteminde SVM mimarisi ile denenmiştir ve sınıflandırma başarısı %97,1 olarak bulunmuştur [26].

Bir başka çalışmada ise Lucio Tommaso De Paolis ve Valerio De Luca tarafından Kinect sensörü tarafından elde edilen veriler üzerinde farklı vücut boyutlarına sahip deneklerde omuz eklemi hareketliliğini değerlendirmek için bir yöntem denenmiştir. Veri setinde, kollarını kaldıran bir hastanın ulaştığı maksimum açı, bu eylem için geçen süre ve zaman bilgileri bulunmaktadır. Farklı omuz eklem açıları olabileceği için hem uzun boylu hem de kısa boylu bir insan ile veriler elde edilmiştir. Ancak, bu durum test sonuçlarında büyük farklara sebep olmamıştır. Bunun haricinde, istatistiksel testler ile büyük açılı genliklerdeki göreceli hatalar ile kollar yukarı konumdayken harcanan zaman arasında bir korelasyon olduğu gözlemlenmiştir [27].

Ömer Faruk İnce ve arkadaşları tarafından gerçekleştirilen çalışmada ise üç boyutlu uzayda insan aktiviteleri tespit etmek için yeni bir biyometrik sistem önerilmiştir. İnsan aktiviteleri tespit etmek için kullanılacak RGB derinlik görüntüleri Kinect sensör ile elde edilmiştir ve elde edilen eklem açıları özellik olarak kullanılmıştır. Yöntemde açı bilgisi kayan çekirdek yöntemi aracılığı ile depolanmıştır. Özellik bilgilerinin kaybolmaması için Haar-dalgacık dönüşümü (HWT) uygulanmıştır. Daha sonra, bir ortalama algoritması aracılığı ile boyut küçültme işlemi uygulanmıştır. Sınıflandırmadan hemen önce, son özellik kümesini çıkarmak için ters Haar-dalgacık dönüşümü ile bir eşikleme yöntemi uygulanmıştır. Sınıflandırma için ML yönteminde K-NN algoritması kullanılmıştır ve sistemin doğruluğu %86,1 olarak elde edilmiştir [28].

Pramod Kumar Pisharady ve Martin Saerbeck tarafından yapılan bir başka çalışmada Kinect kullanılarak elde edilen verilerin geometrik özellikleri kullanılarak konum ve ölçekle değişmeyen, çok sınıflı insan duruş algılama ve tanıma algoritması sunulmuştur. Veri setinde 6000 duruş ve toplamda 10 sınıf etiketi içeren veri bulunmaktadır. Kinect ile elde edilen üç boyutlu 14 iskelet noktası verileri, bir dizi açısal özelliğe dönüştürülmüştür. Bu açısal özellikler 11 vektöre karşılık gelmiştir. Duruşların tespiti, duruş olasılığının

eşiklenmesi ile yapılmıştır. Duruşların sınıflandırılması ise polinom çekirdekli bir SVM sınıflandırıcı kullanılarak yapılmıştır. Algoritmanın hem çevrimdışı yöntemle hem de gerçek zamanlı uygulamayla test edilmesi durumunda %95,78'lik bir sınıflandırma başarısı elde edilmiştir. Ayrıca modelin hassasiyet ve geri çağırma başarıları sırasıyla %100 ve %98,54'tür [29].

Bir başka çalışmada, Wen-June Wang ve arkadaşları tarafından beş farklı insanın farklı duruşlarını başarılı bir şekilde tanımak için bir yöntem önerilmiştir. Bu duruşlar; oturma, ayakta durma, eğilme, diz çökme ve yatmadır. Veri seti, Kinect sensör ile elde edilen derinlik görüntülerinden oluşturulmuştur. Yöntemin ilk aşamasında, bir insanın siluet konturunu çıkarmak için derinlik görüntüsü üzerinde arka plan çıkarma işlemi uygulanmıştır ve sadece insanın silueti içeren derinlik görüntüleri elde edilmiştir. Daha sonra, siluet konturunun yatay bir izdüşümü kullanılmıştır. Bunun amacı ise insanın diz çöküp çökmediğinin belirlenmesidir. Eğer siluet diz çökmüyorsa, yıldız iskelet tekniği, özellik noktalarını elde etmek için siluet konturuna uygulanmıştır. Ardından, derinlik görüntüsünün öznitelik vektörlerini ve derinlik değerlerini hesaplanmıştır. Bunun için öznitelik noktaları ağırlık merkeziyle birlikte kullanılmıştır. Daha sonra duruşların sınıflandırılması için özellik vektörleri ve derinlik değerleri önceden eğitilmiş bir LVQ sinir ağına eklenmiştir ve bunun sonucunda sınıflandırma başarısı %99'un üzerinde bulunmuştur. Bu yöntemin sağladığı avantajlar; çok yüksek bir tanıma oranına sahip, daha az eğitim veri seti ile çalışmaktadır ve literatürdeki diğer yöntemler ile karşılaştırıldığında daha ekonomik bir sensör kullanmaktadır [30].

Ahmed Abobakr ve arkadaşları, bir giriş derinliği görüntüsünden vücut eklem açılarını tahmin edebilen bütünsel bir duruş analizi ile ilgili olarak bir yöntem sunmuşlardır. Yöntemde, vücut eklem açılarını tahmin etmek için evrişimli sinir ağı modeli kullanılmıştır. Yöntemin en büyük avantajları, direkt giriş görüntülerinden vücut eklem açılarını tahmin edebilmesi ve gerçek zamanlı olarak performans sağlaması olarak tanımlanmıştır. Bu sayede, sistem çalışanların işteki kas ve iskelet rahatsızlıklarını belirlemek için RULA gibi ergonomik değerlendirme ölçütlerini gerçek zamanlı değerlendirilmesine olanak sağlamıştır. Önerilen yöntem, genelleştirilebilir bir öğrenme modeli elde etmek için farklı çalışanlar tarafından çeşitli görüş açılarından gerçekleştirilen manuel görevleri simüle eden sentetik eğitim görüntüleri üzerinde eğitilmiştir. Karşılık gelen eklem açıları, ters kinematik aşama

kullanılarak hesaplanmıştır ve önerilen yöntemin 4,67 derecelik bir ortalama MAE ve 6,64 derecelik bir RMSE değerine ulaştığı gözlenmiştir [31].

Omar RIHAWI ve arkadaşları, Otizm Spektrum Bozukluğu (OSB)'na sahip insanlar üzerinde bir çalışma gerçekleştirmiştir. OSB, bazı insanların iletişimi, sosyal becerilerini veya davranışlarını etkileyen bir bozukluktur ve buna sahip insanlar genellikle ortak tekrarlayan davranışlara sahiptir. Bu davranışlar: çırpma, sallanma, dönme vs. olarak tanımlanabilir. Bu çalışmada Kinect sensör ile elde edilen otistik insanların davranışlarını içeren bir veri seti üzerinde bu davranışlar araştırılmıştır. Davranışlar; statik ve dinamik olanlar, basit ve karmaşık olanlar olarak sınıflandırılmıştır. Sistemin sınıflandırma başarısı, yöntemin uygulanabilir olduğunu göstermiştir [32].

Bir başka çalışmada, Ravindu Kumarasiri ve arkadaşları tarafından yürüyüş analizi yakalama aralığını artıran ikili Kinect sistemi ile ilgili bir yöntem sunulmuştur. İnsan yürüyüşü algılamak için kullanılan sensörlerin insana olan mesafesi arttıkça, bu sensörlerin derinlik algılama doğruluğu sınırlamaları nedeniyle, yürüyüş analizinin mesafe aralığı da küçülmektedir. Bu durumun sonucunda elde edilecek veriler etkisiz hale gelmektedir. Bu sebeple bu çalışmada birden fazla Kinect sensör ve Kalman filtresi kullanılarak verilerin birleştirilmesi ile yürüyüş analizinin mesafe aralığı 2,5 m'den 4 m'ye üç yürüyüş döngüsü ile artıran bir sistem geliştirilmiştir. Yöntemin deneysel sonuçlarında, uzunluk ölçümlerinde %4 hata ve fleksiyon ölçümlerinde 5° hata elde edilmiştir [33].

Yifeng Xu ve arkadaşlarının gerçekleştirdiği çalışmada Kinect sensörü kullanılarak elde edilen veri seti üzerinde üç hedef pozu tanıyan gelişmiş bir ileri beslemeli yapay sinir ağı önerilmiştir. Kinect sensörü ile derinlik ve iskelet eklem verileri elde edilmiştir. Çalışma kapsamında kullanılan sinir ağı Kinect tarafından elde edilen veri kümesi ile eğitilmiştir. Ancak eğitimden önce veriler normalleştirilmiştir. Sistemin doğruluğu tespit etmek için test aşamasında farklı bir vücut izleyici kullanılmıştır ve testler gerçek zamanlı olarak uygulanmıştır. Testler sırasında yapay sinir ağı, eklemler güvenilir olarak etiketlenirse bile iskelet izleyicisinin çıktısını işlemeye devam etmiştir. Gerçek zamanlı deneylerin sonucunda sistemin düşme tespitindeki doğruluğu %98,5 olarak bulunmuştur [34].

Heng Sun ve arkadaşları, Kinect sensör aracılığı ile gerçek zamanlı olarak öğrencilerin oturma duruşlarının algılanması ve uyarılması ile ilgili olarak bir yazılım geliştirmişlerdir.

Duruş tanımlaması yapıldıktan sonra eğer sistem yanlış bir duruş algıladıysa, denetlenen kişiye duruşu ile ilgili olarak bir uyarı gönderilmektedir. Sistemin mimarisi üç parçaya ayrılmıştır: ilk olarak Kinect sensör aracılığı ile video verileri elde edilmiştir, daha sonra video verilerinde ilgilenilen böyleyi çıkarmak için video akışındaki gerçek zamanlı görüntü çerçeveleri üzerinde bir takım ön işleme yöntemleri uygulanmıştır ve sonra olarak oturma duruşu tanımlanmıştır. Oturma duruşunu tanımak için iki yöntem kullanılmıştır: ilk olarak kemik verilerine dayalı şablon eşleştirme yöntemi ve daha sonra CNN'e dayalı özellik çıkarma yöntemi kullanılmıştır. İnsan vücudunun üst gövdesinde bulunan ana eklem noktalarının uzamsal koordinatlarını hesaplamak ve hedef noktayı bulmak için veriler farklı uzamsal koordinat sistemine dönüştürülmüştür. Deneysel sonuçlar, sistemin uygulanabilir olduğunu göstermiştir [35].

Shehzan Haider Chowdhury ve arkadaşları kas iskelet rahatsızlıklarının ameliyat sonrasında iyileşmesinde yapılacak olan rehabilitasyon egzersizlerinin kontrol edilmesini sağlayan bir yöntem önermişlerdir. Ameliyat sonrasında uygulanan rehabilitasyon tedavisi genellikle ev ortamında yapılmaktadır. Bu sebeple hastaların tedavi sürecini takip etmek çok zor olmaktadır. Önerilen yöntem ile bu sorunun önüne geçilmesi amaçlanmıştır. Önerilen yöntemde, hasta ev ortamındayken bir kamera önünde doktoru tarafından verilen egzersizleri gerçekleştirmektedir. Elde edilen veriler veya görüntüler sürecin takibi için hastadan sorumlu doktora iletilmektedir. Çalışmada kullanılan iskelet verileri Kinect 3D sensör yardımı ile elde edilmiştir. Sistem makine öğrenimi tabanlı iki farklı model ile test edilmiştir. İlk olarak, HF-LSTM modelini ile egzersizlerin kalitesini tahmin etmek için 3D iskelet verileri kullanılmıştır. Grafik evrişim ağı modelinde ise, 3D iskelet verilerini kullanarak öznitelik çıkarma işlemi uygulanmıştır. Bu iki farklı model, KIMORE veri seti üzerinde eğitilmiş ve test edilmiştir. Elde edilen sonuçlara göre HF-LSTM, 0'lık bir RMSE kaybı elde ederken GCN-LSTM, 0,191 ortalama RMSE kaybı ile çalışmaktadır [36].

Dinesh Kumar Vishwakarma ve Konark Jain gerçekleştirdikleri çalışmada boyut dönüşüm kavramını kullanarak bir insan hareketi tanıma sistemi önermişlerdir. Çalışmada kullanılan veri seti Kinect sensör ile elde edilen 25 eklem için iskelet verisi ile oluşturulmuştur. Üç farklı Kinect sensör yatay açılarda farklı bir şekilde konumlandırılarak video verisi elde edilmiştir. Sistem derin öğrenme mimarisi ile çalışmıştır. Bir görüntü karesinde bulunan insanın iskelet hareketi, eklem konumları aracılığı ile hesaplanmıştır. Daha sonra, bu iskelet eklemleri 2D uzaya ve daha sonrasında 1D uzaya yansıtılmıştır. Açılar

üzerindeki mesafelerin değeri örneklenmiştir ve özellik vektörü olarak saklanması ile SVM kullanılarak sistemin sınıflandırma doğruluğu test edilmiştir. Sistem farklı veri setleri üzerinde test edilmiştir. Elde edilen sonuçlara göre; MSR Action3D veri setinde %94,13; Berkeley MHAD veri setinde %93,34; TST Düşme Tespiti veri setinde %95,70 ve NTU-RGB+D veri setinde %86,84'lük bir başarı elde edilmiştir. Sistem uygulanabilir olarak tanımlanmıştır [37].

Bir başka çalışmada Ahmed Ferdous ve arkadaşları vücut hareketlerinden otomatik olarak duygu tanıma ile ilgili olarak bir yöntem sunmuşlardır. İnsanların içinde bulundukları duygu halleri vücut hareketlerini etkilemektedir. Bu çalışmada bu durumdan faydalanılmıştır ve duygu sınıflandırılması için iki katmanlı bir özellik seçim çerçevesi sunulmuştur. Çalışmada beş farklı duygu üstüne sınıflandırma yapılmıştır. Bunlar: mutluluk, üzüntü, korku, öfke ve duygusuzluktur. İlk olarak, sistemin sınıflandırma başarısını etkilemeyecek veya olumsuz yönde etkileyecek olan gereksiz özelliklerin kaldırılması için Varyans Analizi (ANOVA) ve Çok Değişkenli Varyans Analizi (MANOVA) kombinasyonu kullanılmıştır. Daha sonra, duygu sınıflandırmasını yüksek doğruluk ile elde etmek için gerekli özellikler listesinden faydalanarak özellik alt kümesi seçmek amacı ile ikili kromozom tabanlı bir algoritma geliştirilmiştir. Sistemin sınıflandırma başarısını biraz daha arttırmak için puan ve rütbe düzeyinde bir birleştirme işlemi uygulanmıştır. Sistemin sınıflandırma başarısı halka açık veri seti üzerinde test edilmiştir. Veri setinde toplamda 30 kişiden elde edilen veriler bulunmaktadır. Sistem makine öğrenmesinin farklı mimarileri ile test edilmiştir. Sistem yürüme, oturma ve eylemden bağımsız bir senaryoda test edilmiştir. Elde edilen sınıflandırma sonuçları sırası ile %90, %96 ve %86,66 olarak elde edilmiştir. Sonuçlar, sistemin yüksek doğruluk ile çalıştığı gözlemlenmiştir [38].

D. Arpitha ve arkadaşlarının yaptığı çalışma ise önceki literatür taramalarına göre farklı bir alandır. Yapılan çalışmada, Hint Klasik Dansı Mudra'nın hareketlerinin tanınması üzerine bir sistem geliştirilmiştir. Bu dansa karmaşık insan hareketleri bulunmaktadır. Çalışmada kullanılan veri seti Kinect derinlik sensörü kullanılarak oluşturulmuştur. Sensör yardımı ile 5 farklı oryantasyon ve 10 farklı denek ile veri seti oluşturulmuştur. Oluşturulan veri seti eğitim ve test olarak %70/%30 oranında ayrılmıştır. Veri seti oluşturulduktan sonra HOG özelliklerinin histogramı çıkartılmıştır. Veri setinin sınıflandırılması için, yapay sinir ağları, KNN, spektral grafik eşleştirme (SGM) ve uyarlamalı grafik eşleştirme (AGM) ve destek vektör makineleri (SVM) yöntemleri

kullanılmıştır. SVM yöntemi ile elde edilen kesinlik, geri çağırma ve tanıma değeri sırası ile %0,9581, 0,9434 ve 0,9491 olarak elde edilmiştir [39].

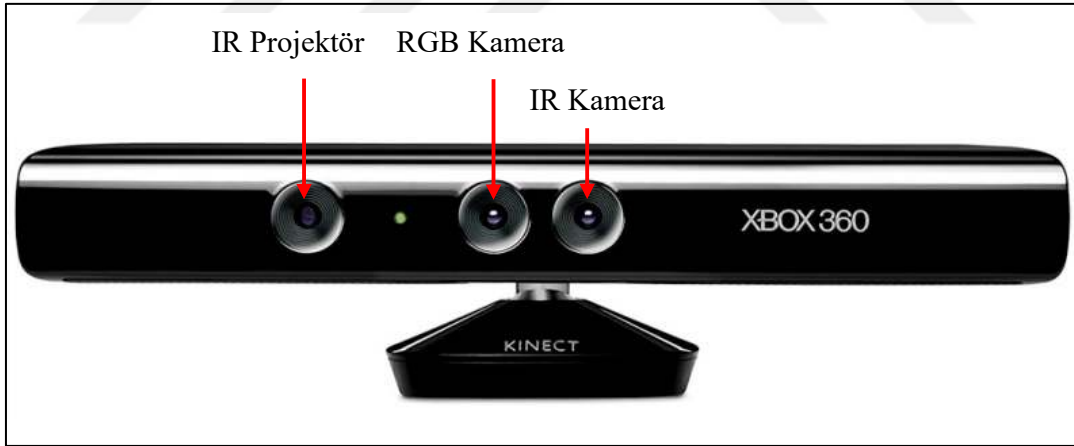
Bu çalışmada iş yerlerinde insanların hatalı duruşlarını tespit ederek gelecekte oluşabilecek sağlık sorunlarının önüne geçilmesi amaçlanmıştır. Literatürde çoğunlukla, güvenlik, fizyoterapi ve düşme tespiti üzerine çalışmalar yapılmaktadır. Bu çalışmanın amacı ile ilgili olarak yapılan çalışma sayısı çok azdır. Bu sebeple, bu alanda olan eksiklik bu alanda çalışma ihtiyacını oluşturmuştur.



2. MATERYAL VE METOTLAR

2.1. Kinect Sensör

Birinci nesil olan Kinect v1 [40], ilk kez Microsoft tarafından 2010 yılında piyasaya sürülmüş, oyun konsolları ve Microsoft Windows bilgisayarlar için geliştirilmiş bir dizi hareket algılama giriş cihazıdır. Bu cihaz, oyuncuların işaret ve konuşma yolu ile herhangi bir kontrol cihazına veya kumandaya dokunmadan temassız bir şekilde oyun oynamalarını sağlamıştır [41]. Daha sonra Microsoft Kinect v1'i geliştirerek 2014 yılında Kinect v2'yi [42] piyasaya sunmuştur. Şekil 2.1'de Kinect sensor verilmiştir. Kinect v2, Kinect v1'e göre daha fazla vücut eklemi algılayabilmektedir [43]. İlk geliştirilen sensör 20 eklem algılayabilirken, yeni geliştirilen sensör 25 eklem algılayabilmektedir. Bu sebeple sensörden elde edilen konum bilgileri daha doğru ve kararlıdır [44]. Kinect sensör dünya çapında ortalama 150\$'a satılmaktadır [45]. Kinect sensörün gelişmiş özellikleri ve düşük maliyeti göz önüne alınca özellikle mühendisler tarafından geliştirilen projelerde yaygın olarak kullanılmaya başlanmıştır [44].



Şekil 2.1. Kinect sensör [46].

Kinect Sensor, bir renkli kamera, bir kızılötesi (IR) yayıcı, bir IR derinlik sensörü, eğme için bir motor, bir mikrofon dizisi ve bir LED ışıktan oluşmaktadır [47]. Kinect cihazları ile üç tür bilgi alınabilir: renkli görüntüler, 3D derinlik görüntüleri ve 3B görüntüleme alanına karşılık gelen kemik bilgisidir [48]. RGB kamera, maksimum 1280 x

960 çözünürlük ile yüz tanıma ve diğer algılama özellikleri için RGB görüntü verilerini toplar [44]. Şekil 2.2’de Kinect Sensör tarafından elde edilen RGB görüntü verilmiştir.



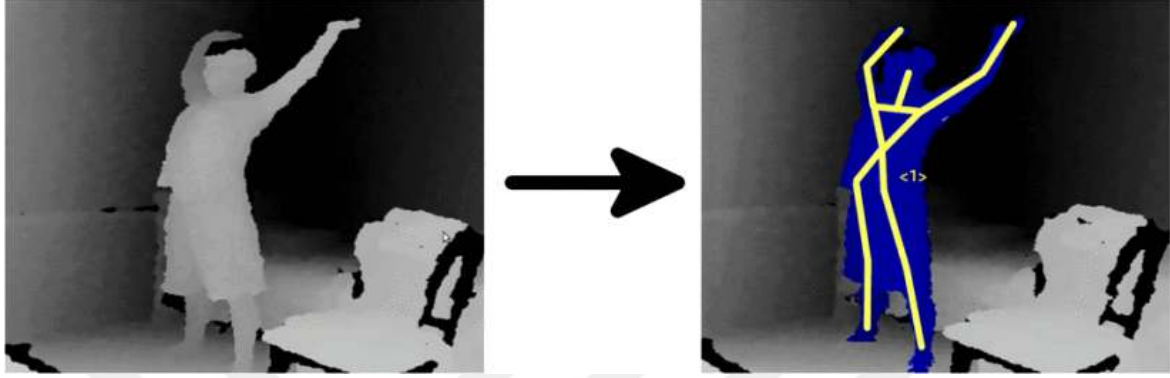
Şekil 2.2. Kinect sensörden elde edilen RGB görüntü [13].

Derinlik sensörü, maksimum 640 x480 çözünürlük üretebilen ve herhangi bir ortam ışığında video verilerini 3D olarak yakalayan monokrom tamamlayıcı bir CMOS sensörü ile birleştirilmiş bir kızılötesi lazer projektörden oluşmaktadır [45]. Şekil 2.3’te, Şekil 2.2’de verilen RGB görüntünün derinlik görüntüsü verilmiştir.



Şekil 2.3. Kinect sensörden elde edilen derinlik görüntüsü [13].

Derinlik görüntüsü ile pikselin hangi renkte olduğunu değil, pikselin sensörden ne kadar uzakta olduğu elde edilmektedir. Bunun haricinde, Kinect sensör API'si elde edilen derinlik görüntülerinden insan eklemlerini içeren bir iskelet görüntüsü üretir [44]. Şekil 2.4'te Kinect sensör ile elde edilen iskelet görüntüsü verilmiştir.



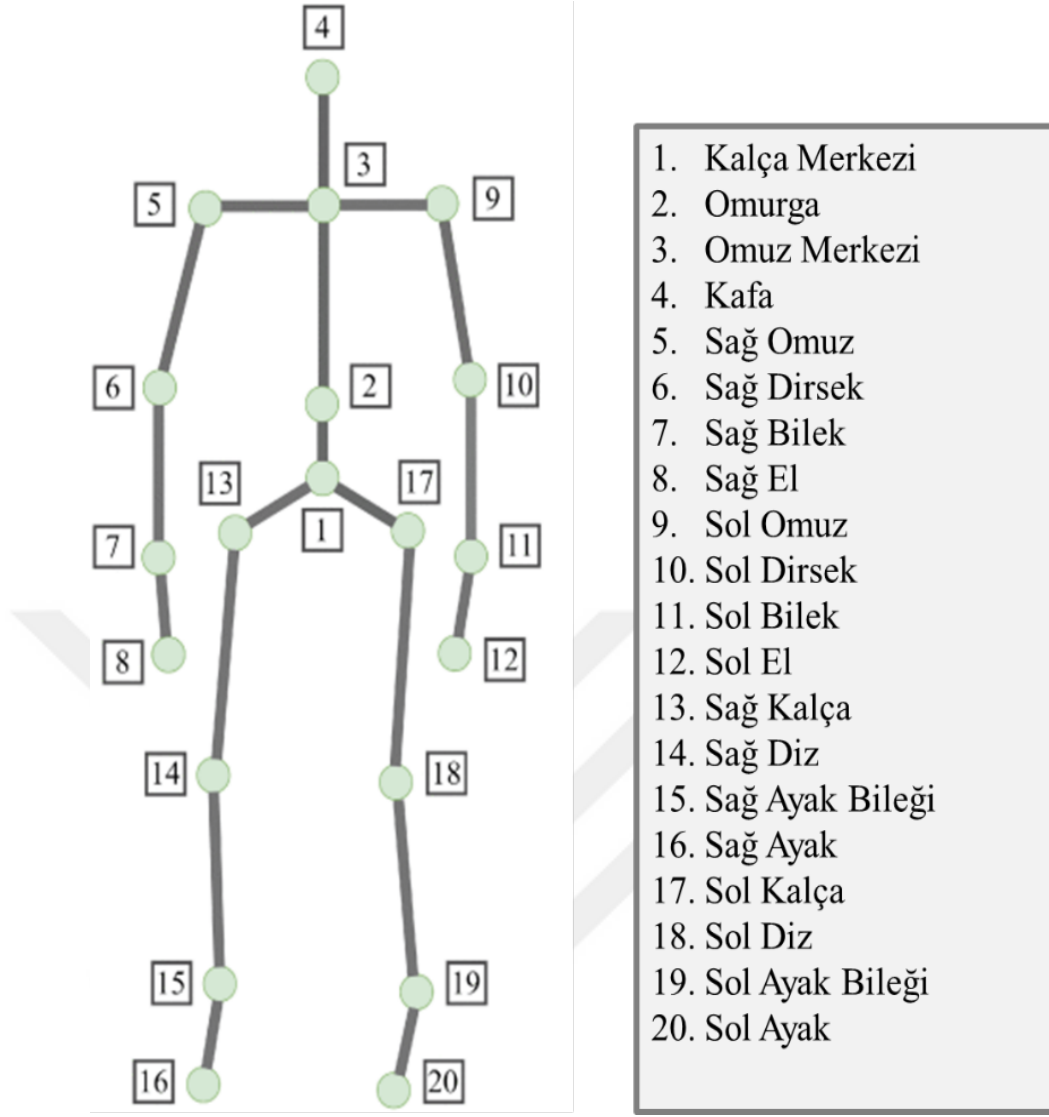
Şekil 2.4. Kinect sensör iskelet görüntüsü [13].

Kinect sensöründen gelen RGB verileri, eklem bilgileri bulunan bir iskelet modeli oluşturmak için kullanılabilir. Bu iskelet modeli, insan faaliyetini tahmin etmek için ham görüntü ve derinliklerle birlikte kullanılmaktadır. Şekil 2.5'te Kinect sensörü ile elde edilen eklemlerin yerleri görülmektedir [13].

Kinect sensörü ile elde edilen videolar aynı anda bir Dump Kinect Skeleton yazılımı kullanılarak elde edilmiş olan eklem bilgilerini kartezyen koordinatlardaki (x,y,z) verilerine dönüştürülebilmektedir. Aşağıda verilen (1.1) ve (1.2)'de (x, y, z) koordinatını (x, y) piksel sayısına yaklaşık olarak 2B görüntüde dönüştürmek için kullanılabilecek eşitlik verilmiştir. Bu sayede, eklemün üç boyutlu konumu göz önüne alındığında, eklemün RGB görüntülerde nerede olduğu bulunabilir [13], [49].

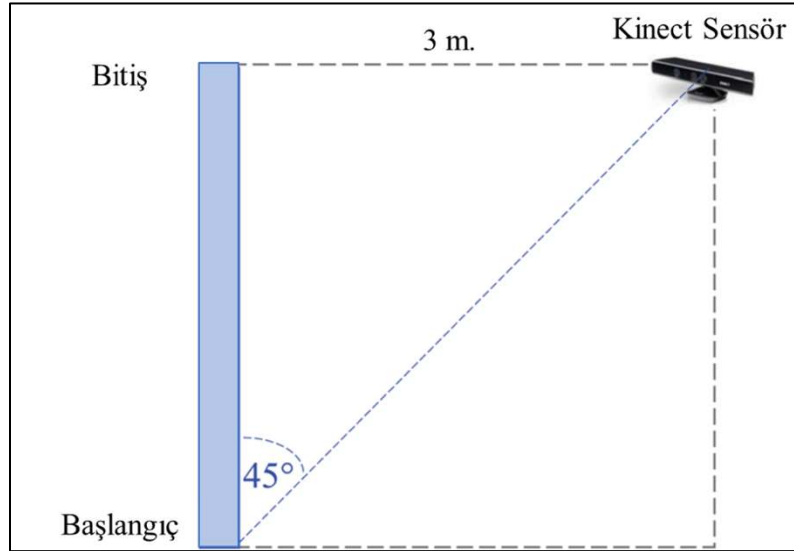
$$x = 156.8584456124928 + 0.0976862095248 * x - 0.0006444357104 * y + 0.0015715946682 * z \quad (1.1)$$

$$y = 125.5357201011431 + 0.0002153447766 * x - 0.1184874093530 * y - 0.0022134485957 * z \quad (1.2)$$



Şekil 2.5. Kinect sensör eklem bilgileri [50].

Kinect sensör ile elde edilen video dizilerinde maksimum başarı elde edilmesi için Kinect sensörü genel olarak yerden 1,3 metre yüksekliğe ve görüntülerin elde edileceği yere çapraz olarak yaklaşık 45 derecelik açı ile yerleştirilmesi gerekmektedir [51]. Kinect sensörde, uzuv başına elde edilen minimum bir tam yürüyüş döngüsüne izin verilen kabul edilebilir mesafe, Kinect ‘ten 1,5 ila 3,5 metre aralığındadır [52]. Bu mesafeler standart olmamakla birlikte, literatürde çoğunlukla bu şekilde kullanılmaktadır. Şekil 2.6’da Kinect sensörün örnek yerleşimi ile ilgili görsel verilmiştir.



Şekil 2.6. Kinect sensör yerleşimi [16].

2.2. Veri Seti

Bu tez çalışmasında insan duruşlarının sınıflandırılmasını yapan algoritmaların başarı değerlerinin birbirleri ile kıyaslanması amaçlanmaktadır. Kinect sensörler, giyilebilir olmaması ve maliyetinin az olması gibi avantajlarından dolayı insan duruşlarının saptanması amacıyla sıkça tercih edilirler. Bu bilgilerin ışığında literatürde Kinect sensör ile elde edilmiş insan duruşlarını içeren veri seti araştırılması sonucunda, Jaeyong Sung ve arkadaşlarının [13] yaptığı çalışmanın uygunluğu tespit edilmiştir. Şekil 2.7’de veri setinden bazı örnek kareler verilmiştir.



Şekil 2.7. Data seti ile ilgili görüntüler [13].

Bu veri setinde 30 Hz kare hızında 1,2 m ila 3,5 m aralığında 640x480 derinlikli RGB görüntü veren Microsoft Kinect sensörü kullanılmıştır. Veri seti elde edilirken ofis, mutfak, yatak odası, banyo ve oturma odası olmak üzere beş farklı ortamda sensör verileri elde edilmiştir. İki erkek ve iki kadın olmak üzere dört kişi ile veri seti oluşturulmuştur. Her bir kişiden her bir aktivite için yaklaşık 45 saniyelik veri toplanmıştır. Veriler toplanırken kişilere nerede durmaları veya ne yapmaları gerektiğine dair talimatlar verilmiştir. Ancak, yaptıkları hareketlerin nasıl etiketleneceğine dair bilgi verilmemiştir. Bu dört kişinin her birinden elde edilen gözlem değerleri Tablo 2.1’de verilmiştir [13].

Tablo 2.1. Veri seti gözlem değerleri

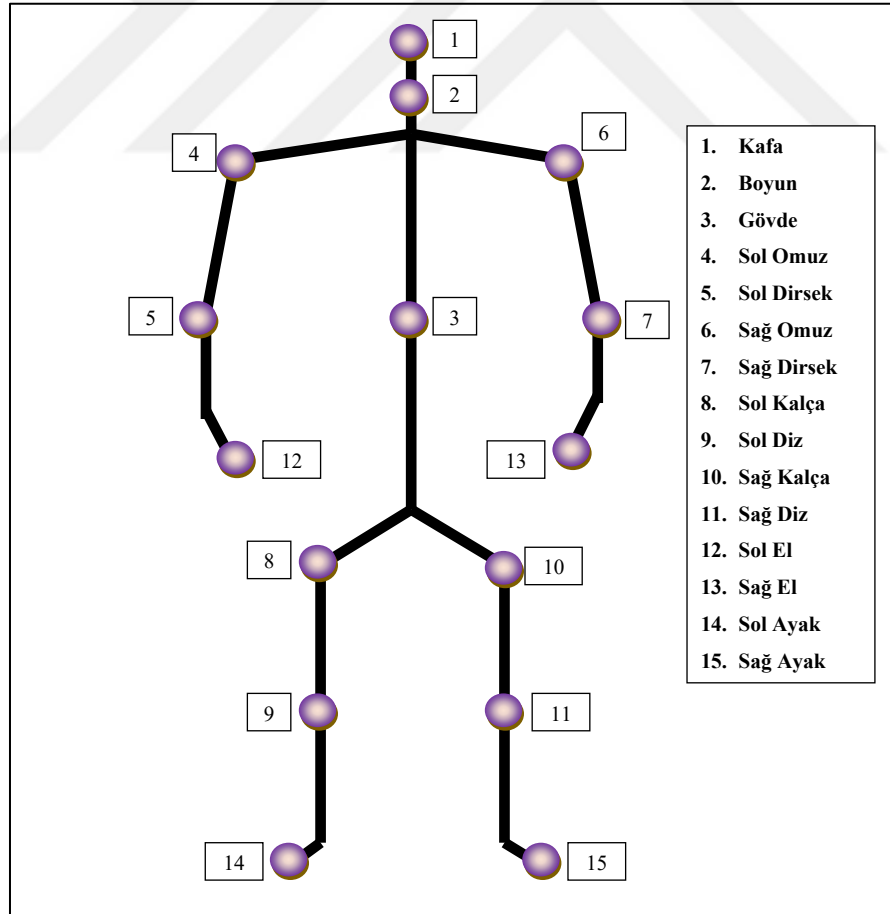
Kişi Numarası	Veri Seti Gözlem Değeri
1	17918
2	16805
3	17577
4	20360

Veri setinde 13 etiket mevcuttur. Etiketler her bir hareketi temsil edecek şekilde 1’den 13’e kadar sıralanmıştır. Bu etiketler ve karşılıkları Tablo 2.2’de verilmiştir.

Tablo 2.2. Etiket değerleri

Etiket Numarası	Etiketin Hareket Karşılığı
1	Telefonda Konuşmak
2	Beyaz Tahtaya Yazmak
3	Su İçmek
4	Su ile Ağızı Çalkalamak
5	Diş Fırçalamak
6	Lens Takmak
7	Sandalyede Konuşmak
8	Sandalyede Dinlenmek
9	Pişirmek (Doğramak)
10	Pişirmek (Karıştırmak)
11	İlaç Kutusu Açmak
12	Bilgisayarda Çalışmak
13	Hareketsiz Durmak

Bu tez çalışmasında veri seti ile iki farklı sınıflandırma uygulanmıştır. Bunlardan ilki Kinect sensör aracılığı ile elde edilen RGB görüntülerdir. Diğeri ise 15 vücut eklemi izleyen Prime Sensor NITE 1.3 algoritması ile elde edilen eklem verileridir. Şekil 2.8’de veri setinde kullanılan eklemler verilmiştir. Eklem verileri içerisinde eklem oryantasyonu ve/veya eklem konumu bilgileri bulunmaktadır. Eklem oryantasyonu, 3x3’lük bir matrsten oluşmaktadır. Eklem konum bilgileri ise x, y ve z olmak üzere 3 değişken ile elde edilmektedir. Hem eklem oryantasyonu hem de eklem konumuna sahip 11 eklem bulunmaktadır. Bunlar; kafa, boyun, gövde, sol omuz, sol dirsek, sağ omuz, sağ dirsek, sol kalça, sağ kalça, sol diz ve sağ dizdir. Bunun haricinde eklem pozisyonuna sahip sadece 4 eklem bulunmaktadır. Bunlar; sol el, sağ el, sol ayak ve sağ ayaktır. Eklemlerin pozisyon bilgilerinden sonra CONF değeri gelmektedir. CONF değeri, NITE [53] çerçevesinde bir güven değeri vermektedir ve 0 veya 1 değeri almaktadır. Bu güven değeri hangi verilerin güvenilir olduğunu bulmakla ilgilenildiği için füzyon amacı ile kullanılabilir. CONF değeri, eklem konumu güvenilir ise 1 verir; veriler güvenilir değil ise 0 vermektedir [13].



Şekil 2.8. Veri setinde kullanılan eklemler.

2.3. Makine Öğrenmesi

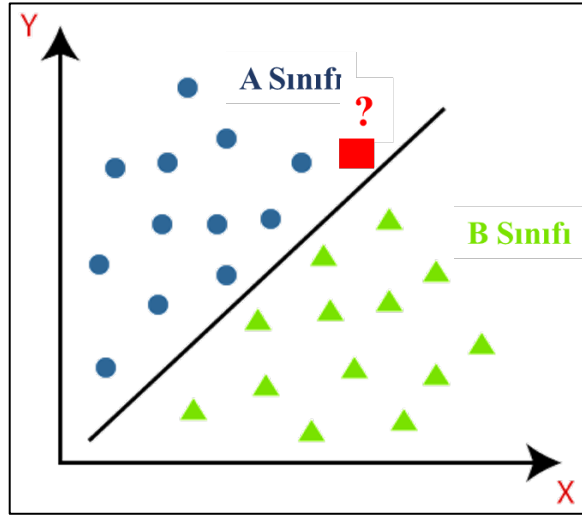
Makine öğrenimindeki yaklaşımlar ile çözümlenmek istenen sorunlar denetimli öğrenme veya deneyimsiz öğrenme olarak sınıflandırılmaktadır. Denetimsiz öğrenmeye, kümeleme; denetimli öğrenmeye ise sınıflandırma ve regresyon örnek verilmektedir [54]. Bu çalışmada ML yaklaşımında sınıflandırma incelenmiştir.

Makine öğrenimi var olan veriler üzerinde tahminler yapmak için algoritmalar kurmaya odaklanır. Bir makine öğrenimi görevi; X giriş alanı verilerini, Y çıkış alanı yani olası tahminleri ile eşleyen bir $f : X \rightarrow Y$ fonksiyonunu öğrenmeyi amaçlamaktadır. f fonksiyonları her öğrenme algoritmasının tipine bağlı olarak farklı olan bir fonksiyon sınıfından seçilir [55]. ML yöntemi aşağıda verilen listedeki sıra ile gerçekleştirilmektedir [56]:

- Çözülmesi istenen sorunun tanımlanması,
- Eğitim ve test verilerinin elde edilmesi,
- Verileri niteleyen özelliklerin belirlenmesi veya çıkarılması,
- Eğitim verisi ile modeli eğitmek,
- Algoritmada daha önce tanımlanmayan test verileri üzerinde eğitilmiş modelin performansını değerlendirmek için modelin edilmesi.

2.3.1. Makine öğrenmesinde sınıflandırma

Makine öğrenmesinde eğitim ve test verileri olmak üzere iki ayrı veri seti bulunmaktadır. Bu veri setinde bulunan her bir veri için bir etiket yani sınıf bilgisi bulunmaktadır. Modelin eğitilmesi kısmında, eğitim verisindeki her bir verinin hangi sınıfa ait olduğuna dair sistem eğitilmektedir. Bunun sonucunda sistemin eğitmediği yeni bir veri geldiğinde (test verisi) model yeni gelen verinin hangi sınıfa ait olduğunu tahmin etmeye çalışmaktadır. Şekil 2.9'da ML sınıflandırma örneği verilmiştir.



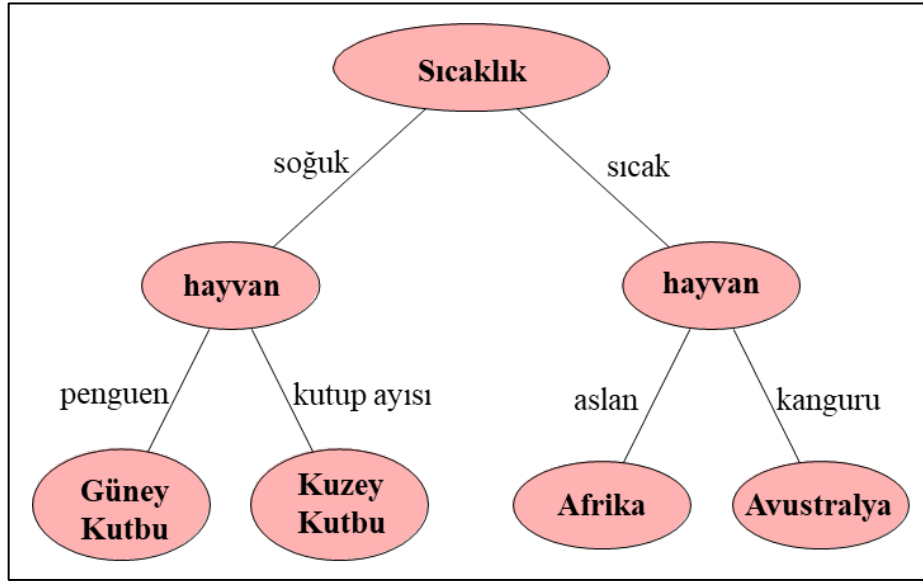
Şekil 2.9. ML sınıflandırma örneği.

Şekil 2.3'e bakılacak olursa, şekilde daireler, A sınıfı; üçgenler ise B sınıfı olarak tanımlanmıştır. Yeni gelen karenin aslında A sınıfına mı B sınıfına mı ait olduğu eğitilmiş model ile belirlenmeye çalışılmaktadır. Verilerin sınıf/etiket bilgisi A, B olabileceği gibi; binary sayılar: 0, 1, 2, ...; isim bilgisi: kedi, köpek, kuş, vs. şeklinde de olabilir. Sınıflandırma için geliştirilmiş birçok yöntem modeli bulunmaktadır.

2.3.2. Karar ağaçları

Karar Ağacı, yapısı itibari ile kolay anlaşılabilir olmasından dolayı en yaygın olarak kullanılan öğrenme yöntemlerinden biri olan sınıflandırma tekniğidir. Karar ağacı algoritmaları, belirlenmiş olan özellik değerlerine verileri sınıflandıran ağaçlardan oluşmaktadır. Bu ağaç yapısında gerçek hayattaki ağaç yapısına benzer olarak düğümler, yapraklar ve kökler bulunmaktadır [57].

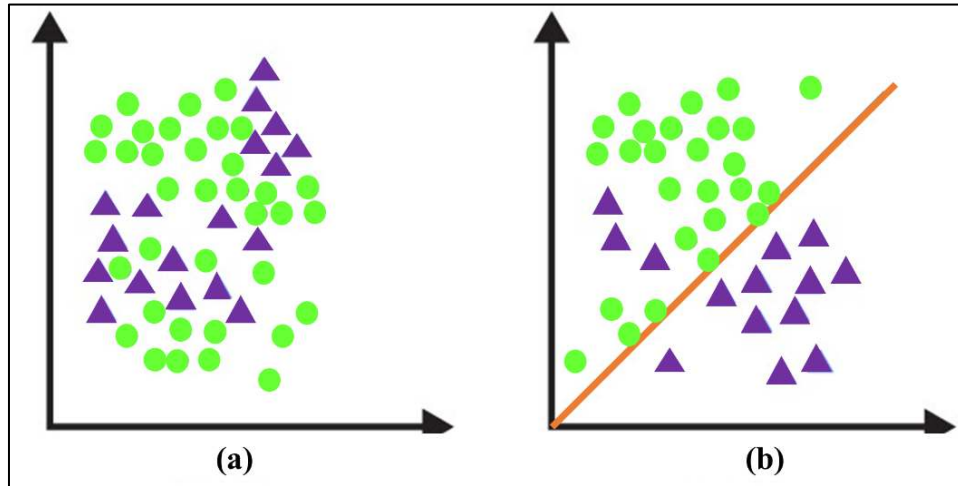
Şekil 2.10'da karar ağaçları yapısının örneği görülmektedir. Şekilde de görüldüğü üzere düğümler (sıcaklık, hayvan) sınıflandırılması istenen özellikleri göstermektedirler. Dallar (soğuk, sıcak, vs.) ise düğümü tanımlamak için kullanılan değerleri göstermektedir. Kök (Güney Kutbu, Kuzey Kutbu vs.) ise algoritma sonunda elde edilmek istenen sınıf bilgilerini temsil etmektedir. Veriler, kök düğüm ile en etkili şekilde sıralanabilmektedir [2], [58], [59].



Şekil 2.10. Karar ağaçları örneği [2].

2.3.3. Doğrusal ayırık analiz

Doğrusal ayırık analizi (LDA), 1936 yılında Ronald Fisher tarafından geliştirilen denetimli bir öğrenme algoritmasıdır [56]. LDA, uygulama açısından basit bir yapıdadır [60]. Bu basit yapısı Şekil 2.11’de verilmiştir.



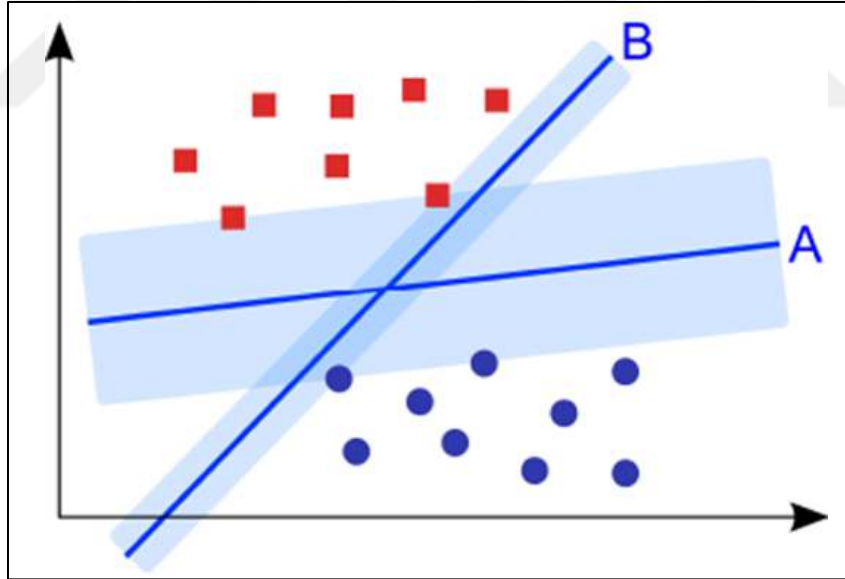
Şekil 2.11. (a) LDA'dan önce (b) LDA'dan sonra.

LDA, doğrusal olarak verilen veri kümelerine uygulanan bir boyutsallık azaltma tekniğidir [61]. Boyutsallık azaltma işlemi var olan verilerdeki özelliklerin azaltılması ile yapılmaktadır. Bu işlemin yapılmasının amacı ise sınıflandırma yapılırken modelin daha

verimli çalışmasıdır [61]. Ayrıca LDA, boyutsallığı azaltırken ayırt edici olan özellikleri saklamaktadır. LDA, boyutluluk azaltma işlemini sınıfların ayrıklık bilgisini koruyarak gerçekleştirmektedir [62]. Bu yöntemde verilerin istatistik özelliklerinden faydalanılarak ortalama, varyans ve çok değişkenli gauss hesaplanmaktadır [62]. Sınıf tahmini ise bayes yöntemi kullanılarak gerçekleştirilmektedir. LDA, küresel bir koordinat sisteminde tek bir doğrusal dönüşümü dikkate aldığı için doğrusal olmayan sistemleri işleyememektedir [63].

2.3.4. Destek vektör makineleri

Destek vektör makineleri 1995 yılında Corinna Cortes ve Vladimir Vapnik tarafından geliştirilmiştir [64]. SVM algoritması, genel olarak denetimli öğrenme problemlerinde kullanıldığı için test verilerini sınıflandırmak için, sınıf bilgisi olan eğitim setine ihtiyaç duymaktadır [2]. Ancak, geliştirilen destek-vektör kümeleme yöntemi [65] ile denetimsiz öğrenme problemlerinde de kullanılabilir [56]. Şekil 2.12’de bir iki tane sınıf bilgisi olan veri setini iki farklı lineer ayırım kullanarak iki sınıfa bölen SVM örneği verilmiştir.

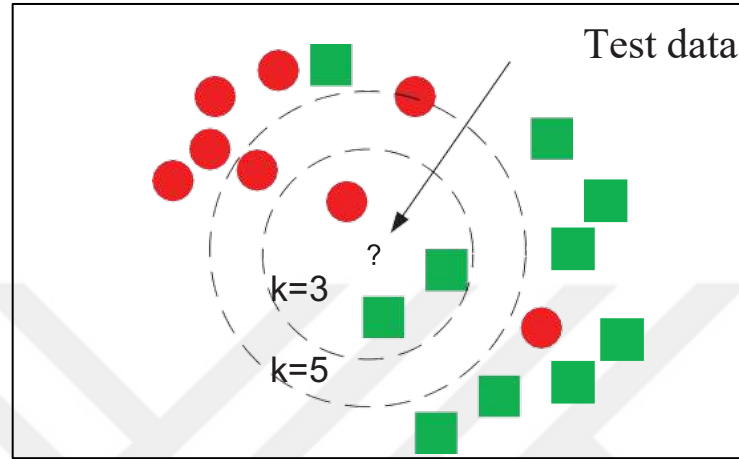


Şekil 2.12. SVM örneği [66].

SVM’ler regresyon ve sınıflandırma işlemlerinde kullanılabilir. Ancak, büyük veri kümelerinde yüksek bir başarı gösterdikleri için genel olarak sınıflandırma problemlerinde kullanılmaktadır [67]. SVM, bir dizi hiper düzlem oluşturur [56]. Bu düzlemlerin olduğu yer sonsuz boyutlu veya yüksek boyutlu bir uzaydır.

2.3.5. K-En yakın komşu algoritması

K-En yakın komşu algoritması, 1951 yılında Fix ve Hodges tarafından geliştirilmiş mesafe hesabına dayalı denetimli bir öğrenme algoritmasıdır [68]. Şekil 2.13'te K-NN örneği verilmiştir.



Şekil 2.13. K-NN algoritması [69].

KNN algoritması, yapısı itibari ile en basit ve en çok kullanılan sınıflandırma algoritmalarından birisidir. K-NN mimarisinde eğitim verilerini öğrenmek yerine eğitim verileri ezberlenmektedir. Bu sebeple büyük veri setlerinde yaygın olarak kullanılmaktadır [70].

K-NN algoritmasının çalışmasında etiket değerleri belirli olan eğitim veri seti oluşturulduktan sonra, sınıflandırılmada kullanılacak olan mesafe fonksiyonu için K parametresi belirlenmektedir. K parametresi, algoritma çalışırken bakılacak eleman sayısını temsil etmektedir. Sınıflandırmaya yeni bir veri geldiğinde, eğitim setinden en yakın K eleman sayısı alınarak yeni gelen verinin belirlenen mesafe fonksiyonu kullanılması ile eğitim setinde olan verilere uzaklığı hesaplanmaktadır. Yeni gelen verinin, veri setine uzaklığının hesaplanmasına göre en yakın sınıfa atanarak sınıflandırma işlemi tamamlanmaktadır [71].

Bu algoritmanın çalışma mantığı mesafe hesabına dayalı olarak çalışmaktadır. En yaygın kullanılan mesafe hesabı Öklid Mesafesi'dir. Öklid Mesafesi'nin hesabı (2.1)'de

verilmiştir. Öklid mesafesinin haricinde mesafe hesabında kullanılan iki farklı yöntem daha vardır: Chebyshev Mesafesi ve Manhattan Mesafesi'dir [72]. Chebyshev Mesafesi (2.2) ve Manhattan Mesafesi (2.3)'te verilmiştir [56], [73].

Öklid Mesafesi:

$$d(x^*, x^i) = \sqrt{\sum_{j=1}^d (x_j^* - x_j^i)^2} \quad (2.1)$$

Manhattan Mesafesi:

$$d(x^*, x^i) = \sqrt{\sum_{j=1}^d |x_j^* - x_j^i|} \quad (2.2)$$

ChebyshevUzaklığı:

$$d(x^*, x^i) = \max(|x_1^* - x_1^i|, |x_2^* - x_2^i|, \dots, |x_d^* - x_d^i|) \quad (2.3)$$

2.3.6. Topluluk sınıflandırıcı

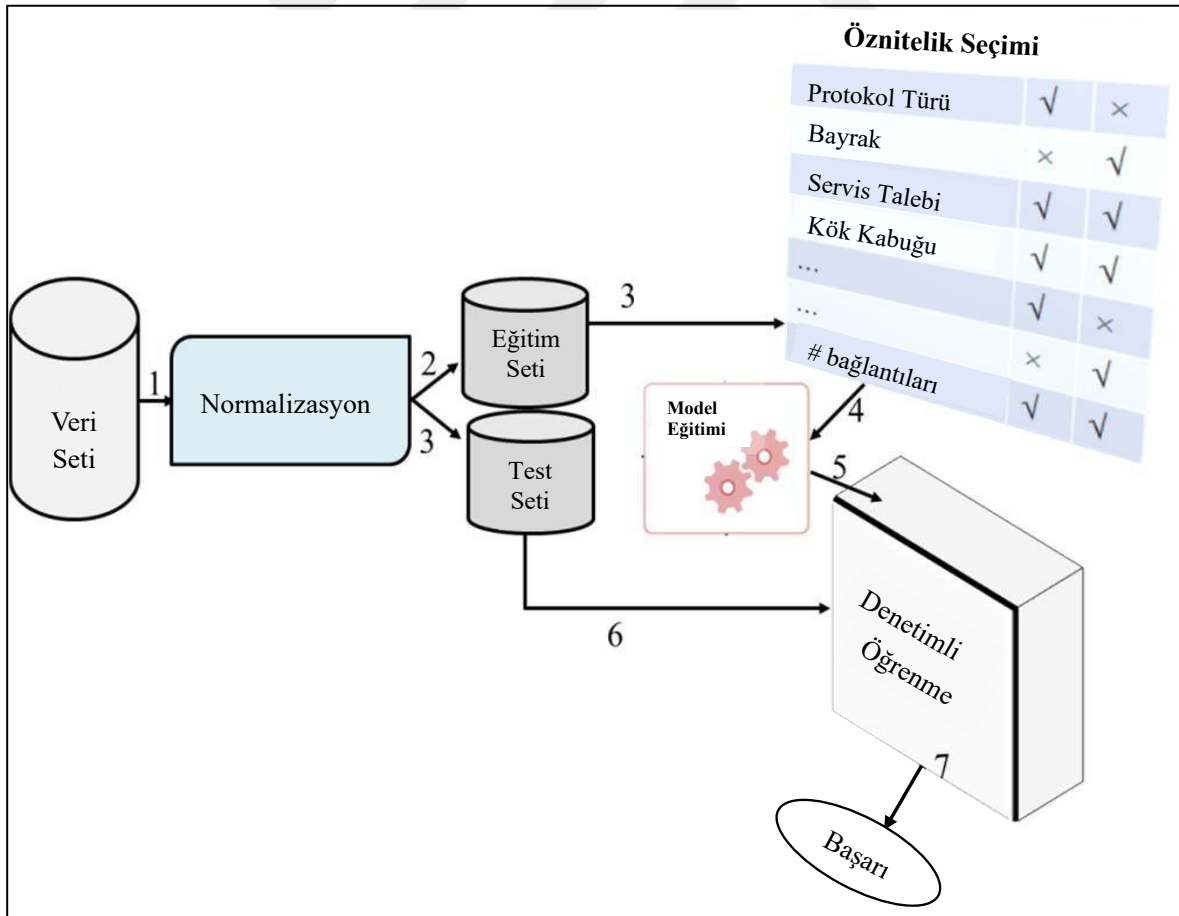
Topluluk sınıflandırıcı [74], son yıllarda geliştirilmiş olan ML'de en yaygın kullanılan yöntemlerinden biri olmuştur. Bu algoritmada, daha iyi tahmin performansı elde etmek için çoklu öğrenme algoritmalarını kullanılmaktadır [75]. Topluluk öğrenmede bir alt prosedür olarak, topluluk budama, bir alt kümesi seçmeyi amaçlar [76]. Topluluk budama, temel sınıflandırıcıların en iyi alt kümesini korumayı ve bu sırada da gereksiz temel sınıflandırıcıları ortadan kaldırmayı amaçlama prosedürüdür [72].

2.3.7. Öznitelik seçme

Bir veri setinde var olan her bir satır bir zaman serisini temsil ediyor denilebilir. Bu veri setinde bulunan zaman serilerini yani satırları tanımlayan sütunlar ise öznitelik olarak tanımlanmaktadır. Örneğin, kediler ile ilgili bir veri setinde kedileri tanımlayan kulak yapısı (kırık kulak veya dik kulak), tüy yapısı (uzun tüylü veya kısa tüylü), cins (tekir, british,

scottish vs.) gibi özellikler öznitelikleri oluşturmaktadır. Her veri setinin kendine özel öznitelik sayısı vardır [77].

ML algoritmaları, kendi kendine veriden öğrenme yeteneğine sahiptir. Bu sebepten dolayı, bu algoritmaların performansı ve algoritma sonucunda elde edilecek olan başarı veri setine bağlıdır. Veri setinde başarıyı ve performansları belirleyen durum ise veri setinin bilgileri olarak tanımlanan özniteliklerdir. Büyük veri setleri üzerinde gerçekleştirilen işlemler çok fazla zaman almaktadır. Ayrıca, veri setinde bulunan özniteliklerin tamamı gerçekleştirilen çalışmada kullanılmak istenmiyor olabilir veya işlemler sırasında bazı öznitelikler bulunan diğer öznitelikleri baskılıyor ve bunun sonucunda da başarı etkileniyor olabilir. Bu sebeple öznitelik seçme işlemleri yapılarak, algoritmanın çalışma süresinin azalmasına, tahmin performansının artmasına ve makine öğrenimi uygulamalarında verinin daha iyi anlamlandırılması sağlanabilir. Şekil 2.14'te öznitelik seçimi ile ilgili görsel verilmiştir.



Şekil 2.14. Öznitelik çıkarımı ile makine öğrenmesi [78].

Öznitelik seçimi işleminde, veri seti üzerinde en etkili olan bir başka deyişle, performansı en çok arttıran özniteliklerin seçilmesi ve bu özniteliklerin haricinde kalan özniteliklerin ise veri setinden ayıklanması yapılmaktadır. Öznitelik seçimi yapıldıktan sonra veri setinin boyutunda bir küçülme olmaktadır. Bu küçülme, algoritmanın çalışma süresinin azalmasına olanak sağlamaktadır. Sonuç olarak, öznitelik çıkarma yapılarak karmaşık veriler daha anlaşılır hale gelmektedir, algoritmanın işlem zamanından kazanç sağlamaktadır ve doğru işlem adımları ile başarı arttırılabilmektedir. Öznitelik seçimi ile ilgili kullanılan pek çok yöntem vardır. Bu yöntemler arasında en sık kullanılanlar; ileriye doğru öznitelik seçimi ve geriye doğru öznitelik seçimidir [58], [79]–[81].

2.3.7.1. Geriye doğru öznitelik seçimi (BFS)

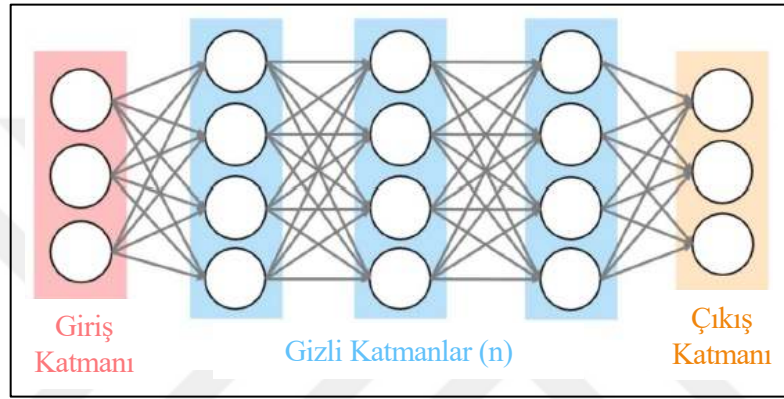
Öznitelik seçme yöntemlerinden birisi olan BFS'nin temel amacı veri setinden zaman ve başarı açısından en verimli eğitim setini oluşturmaktır. Bu yöntemde, öznitelik seçme işlemi öznitelikler çoktan aza olacak şekilde geriye doğru işlenir. İlk olarak veri setindeki tüm öznitelikler ile algoritma sonucu elde edilir. Daha sonra sıra ile öznitelik elenir. Bu işlem yapılırken hangi öznitelik ayıklanınca başarı artıyor ise o öznitelik çıkarılmaktadır. Bu işlem veri setinde kalan son öznitelik için de başarı sonucu elde edilince tamamlanmaktadır. Her bir öznitelik seçme işleminde elde edilen başarı değerleri karşılaştırılarak en yüksek başarıyı veren küme, eğitim seti olarak seçilmektedir [79], [82], [83].

2.3.7.2. İleriye doğru öznitelik seçimi (FFS)

İleriye doğru öznitelik seçimi, geriye doğru öznitelik seçimi ile aynı amacı taşımaktadır. Ancak, iki yöntemin işleyiş şekli birbirinin tam tersidir. BFS'de geriye doğru öznitelik eleme yapılmaktadır. FFS'de ise tam tersi olarak ileriye doğru öznitelik ekleme yapılarak bir başarı sonucu elde edilecektir. Bu yöntemde ilk aşamada, her bir öznitelik için ayrı bir başarı değeri elde edilmektedir. Daha sonra hangi öznitelikte başarı daha yüksek ise o öznitelik veri setine eklenir ve bu şekilde işleme devam edilir. Her aşamada veri setine bir öznitelik eklenerek bir başarı değeri elde edilir. Bu işlem tüm öznitelikler veri setine eklenene kadar devam etmektedir. Her bir öznitelik ekleme işleminde elde edilen başarı değerleri karşılaştırılarak en yüksek başarıyı veren küme, eğitim seti olarak seçilmektedir [79].

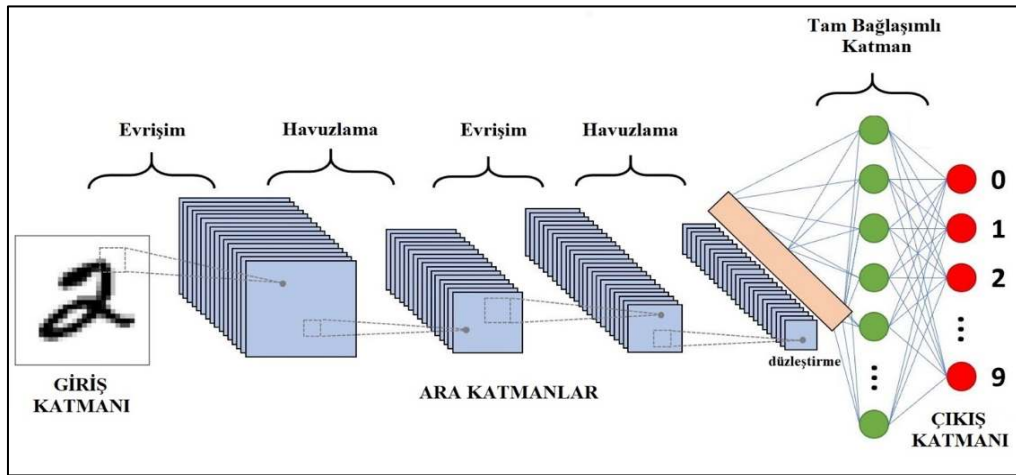
2.4. Derin Öğrenme

Derin öğrenme, birbiri ardından gelen katmanlardan oluşan ve her katman işlendiğinde daha verimli temsiller elde edilebilen, örneklerden öğrenen bir makine öğrenimi tekniğidir [84]. Şekil 2.15'te derin öğrenme mimarisinin genel örneği verilmiştir. Derin öğrenme de derin kelimesi mimarinin içerisinde bulunan modelleri oluşturan katmanların çokluğunu ifade etmektedir [85]. Derin öğrenme mimarisi çoğunlukla görüntülerden oluşan örnekler üzerinde çalışsa da bunun haricinde ses veya metin verisinden de öğrenme sağlayabilir [86].



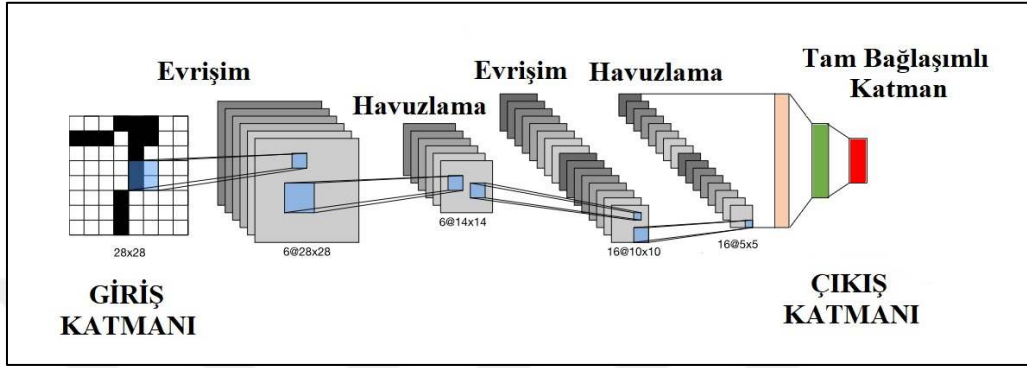
Şekil 2.15. Derin öğrenme mimarisi [86].

Derin öğrenme yöntemi genellikle büyük, etiketli veri setleri üzerinde öğrenme gerçekleştirdiği için makine öğrenimine kıyasla daha yüksek doğruluk ile çalışmaktadır [86]. Şekil 2.16'da derin öğrenme mimarisinin açık gösterimi verilmiştir. Bu çalışma kapsamında derin öğrenme tabanlı olarak LSTM ve CNN yöntemleri incelenmiştir.



Şekil 2.16. Klasik bir derin öğrenme modeli katmanları [87].

CNN, derin öğrenme mimarisi altında ilk olarak LeCun ve arkadaşları tarafından LeNet adı ile geliştirilen ve son yıllarda sınıflandırma alanında çok yaygın bir şekilde kullanılan bir ileri beslemeli sinir ağı çeşididir [88]. CNN'nin son yıllarda çok tercih edilmesinin sebebi ise doğrusal olmayan öznitelikler üzerinde çıkarımlar yaparken yüksek performans göstermesidir [89]. Geliştirilen LeNet mimarisi Şekil 2.21'de verilmiştir.

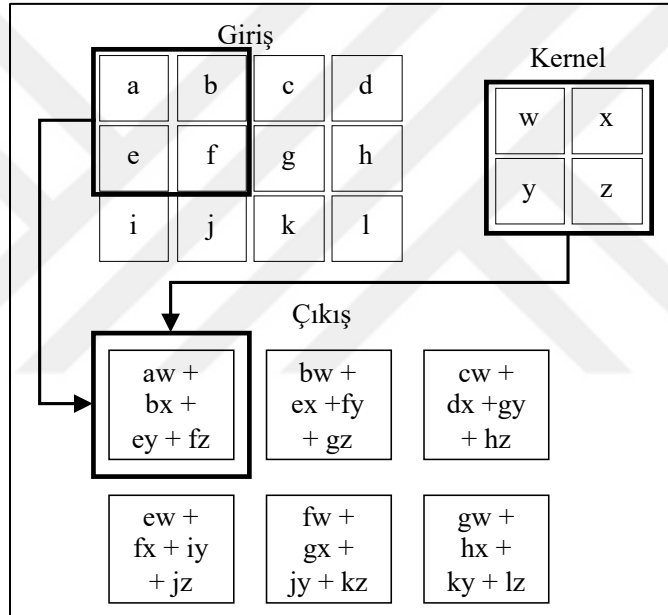


Şekil 2.17. LeNet mimarisi [88].

CNN mimarisinin oluşturulmasında canlıların görme ve algılama sisteminden faydalanılmıştır. Örneğin insan görme sisteminde, insanlar baktıkları noktadaki tüm şekil ve renkleri algılamaktadırlar. Ancak, bu işlem yazıldığı kadar basit değildir; arka planda birçok işlem gerçekleşmektedir. CNN mimarisi geliştirilirken bu arka planda yürütülen işlemlerden yararlanılmıştır. Bu mimaride, sınıf bilgisini elde etmek için öncelikli olarak düşük öznitelikler çıkarılır ve ardından öğrenme işlemi gerçekleştirilmektedir. Bu işlemden sonra yüksek seviyeli öznitelikler ayıklanmaktadır. Bu sayede sınıflandırma işlemi kolaylaşmaktadır [90].

CNN mimarisi çoğunlukla görüntü verilerini sınıflandırmak, tanımlamak; nesne algılama ve nesne izleme sistemlerinde kullanılmaktadır [91]. CNN yöntemi çoğunlukla görüntü verileri üzerinde çalıştığı ve görüntüler çok fazla pikselden oluşabileceği için bu yöntemin işlem açısından gerçekleşme süresi uzundur. CNN yöntemi genellikle özellik çıkarma ve sınıflandırma bölümü olarak iki ana bölümden oluşmaktadır [92]. Özellik çıkarma kısmı, bir veya birkaç evrişim katmanından ve havuzlama katmanından oluşur [93]. Çıkarılan çıktı (özellik haritası), sınıflandırma bölümünün girdisi olur. CNN için literatürde çeşitli katman sayıları ve çeşitlerinden oluşan kombinasyonlar bulunmaktadır. Ancak, katmanlar genel olarak evrişim katmanı, havuz katmanı ve tam bağlantılı katmandır [94].

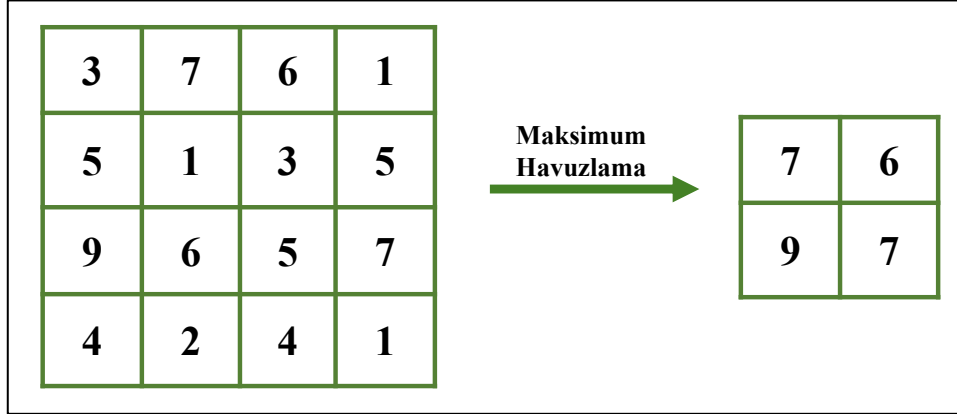
a. Evrişim Katmanı (Konvolüsyon Katmanı): Evrişim katmanı, CNN mimarisinde öznitelik çıkarımı gerçekleştirmek için kullanılan ana katmandır. Bu katman giriş katmanından alınan görüntüler üzerinde filtreler, yani çekirdeklerden oluşan nöronlar [95] kullanarak çeşitli özellikler çıkarmak için kullanılan ilk katmandır. Bu katmanda, giriş görüntüsü ile iki boyutlu bir filtre arasında matematiksel evrişim işlemi gerçekleştirilir. Şekil 2.22’te Kernel filtre kullanılarak gerçekleştirilen bir Konvolüsyon örneği verilmiştir. Genellikle kullanılan filtre boyutları 3x3, 5x5 veya 7x7’dir. Evrişim işlemi için filtre giriş görüntüsü üzerinde kaydırılır ve Filtreyi giriş görüntüsünün üzerine kaydırarak, filtrenin boyutuna (MxM) göre tek bir değer elde edilir. Filtreme işlemi sonucunda önemli öznitelikler elde edilmekte ve daha az önemli olan öznitelikler ortadan kaldırılmaktadır [96].



Şekil 2.18. Evrişim işleminin çizimi [97].

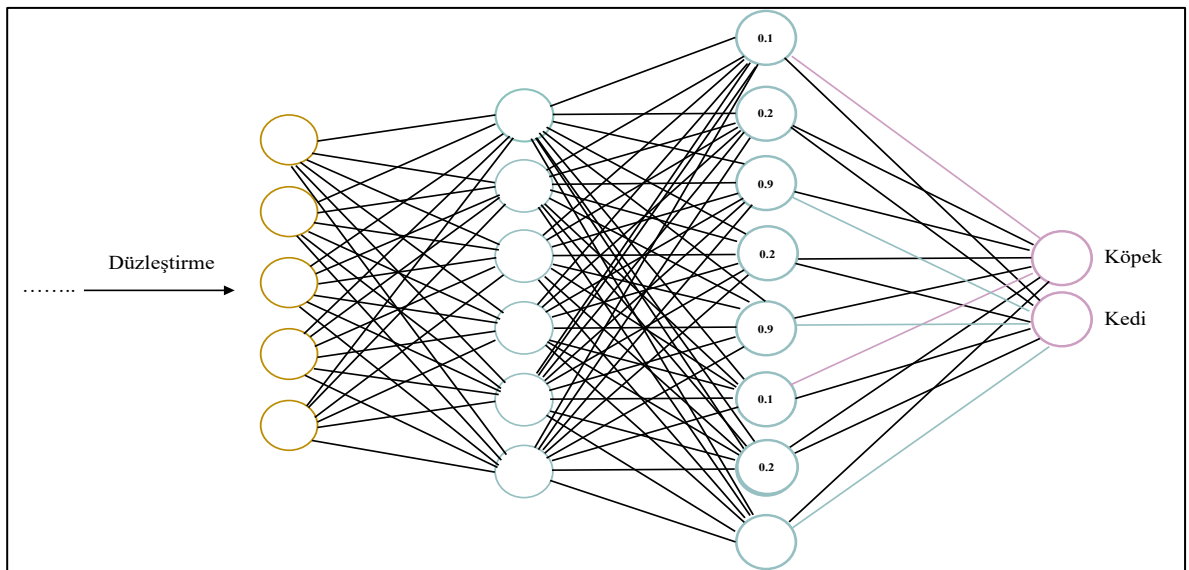
b. Havuzlama (Ortaklama): Bu katmanın birincil amacı, kendisinden sonra gelen katmanlardaki karmaşıklığı azaltarak önemli olan piksellerin vurgulanmasıdır. Ayrıca, bu işlem sonrasında parametre sayısı azalmaktadır ve bu da eğitim süresini azaltmaktadır. Katmanın çalışmasında evrişim katmanında olduğu gibi iki boyutlu filtreler kullanılmaktadır. Filtreler tarafından kapsanan her pencerede maksimum norma göre ayırım yapılmaktadır. Bu katman genellikle iki evrişim katmanı arasında kullanılmaktadır. Maksimum havuzlama ve ortalama havuzlama bu katmanda en çok kullanılan yöntemlerdir. Bu iki yöntem arasında ise maksimum havuzlama, standart işlev olarak kabul edilmektedir. Maksimum havuzlama yönteminde görüntü küçük parçalara ayrılır ve her parçadaki büyük

değer alınmaktadır. Şekil 2.23'te 2x2 filtre ile maksimum havuzlama örneği verilmiştir [96], [98].



Şekil 2.19. 2x2 Filtre kullanan maksimum havuz işlemi.

c. Tam Bağlı Katman: CNN mimarisinde kullanılan son katmandır. Bu katmanda sınıflandırma bilgisini oluşturmak için kendisinden önceki katmanlar tarafından oluşturulan veriler derlenmektedir. Bu katmanda çıktı düğümü sayısı ile sınıf sayısı birbirine eşittir. Tam bağlantılı katmanda sigmoid, tanh veya softmax gibi aktivasyon fonksiyonları kullanılmaktadır. Tam bağlantılı katmanda, sinir ağlarının yapısına benzer şekilde düğümler kullanılmaktadır. Bu katmandaki her düğüm, katmanın kendisinden önceki ve sonraki katmanlardaki her düğüm ile bağlantılıdır. Şekil 2.24'te bu durumun örneği verilmiştir [96], [99]–[101].



Şekil 2.20. Tam bağlantılı katman çizimi [102].

Derin öğrenme yönteminde sınıflandırma amacı ile kullanılabilecek olan üç farklı metodoloji vardır. Bunlar; sıfırdan öğrenme (learn from scratch), öğrenme aktarımı (transfer learning) ve özellik çıkarıcıdır. Bu çalışma kapsamında derin öğrenme yönteminde sıfırdan öğrenme ve öğrenme aktarımı yöntemi üzerinde çalışmalar geliştirilmiştir [103].

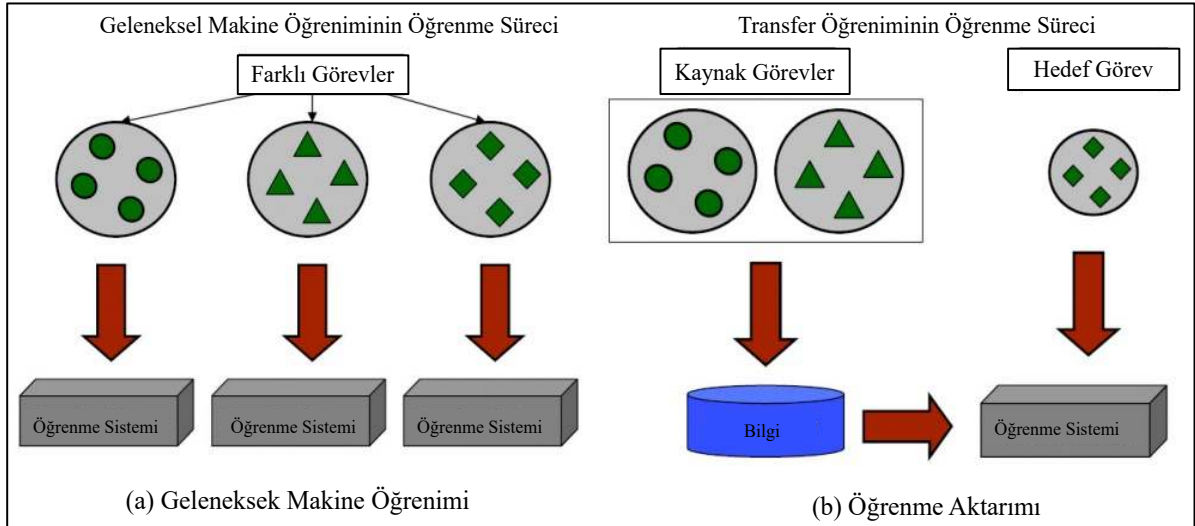
2.4.1. Sıfırdan öğrenme (Learn from scratch)

Geleneksel makine öğrenmesi yönteminde sınıflandırma amacı ile istenilen model sıfırdan öğrenilerek geliştirilebilir. Ancak, bunun için yapılmak istenen çalışma ile ilgili olan büyük veri seti kullanılması gerekmektedir. Örneğin, kedi cinslerinin sınıflandırılmasının istendiği bir çalışmada, kediler ile ilgili yüksek miktarda veri içeren bir veri seti kullanılması ile kedilerin farklı özelliklerini öğrenebilecek istenilen özelliklerde bir ağ mimarisi tasarımı yapılır. Ağ mimarisinin eğitilmesi veri hacmine, öğrenme hızına ve işlem gücüne bağlı olarak uzun bir sürede gerçekleşmektedir [103], [104].

Sıfırdan öğrenme yönteminin dezavantajları, eğitim süresinin uzun olması ve büyük veri setlerine ihtiyaç duymasıdır. Sistemde her zaman büyük veri seti elde etmek mümkün değildir. Bu noktada eğitim süresinin daha kısa olması ve büyük veri setine ihtiyaç duymaması ile transfer öğrenimi yöntemi devreye girmektedir [105].

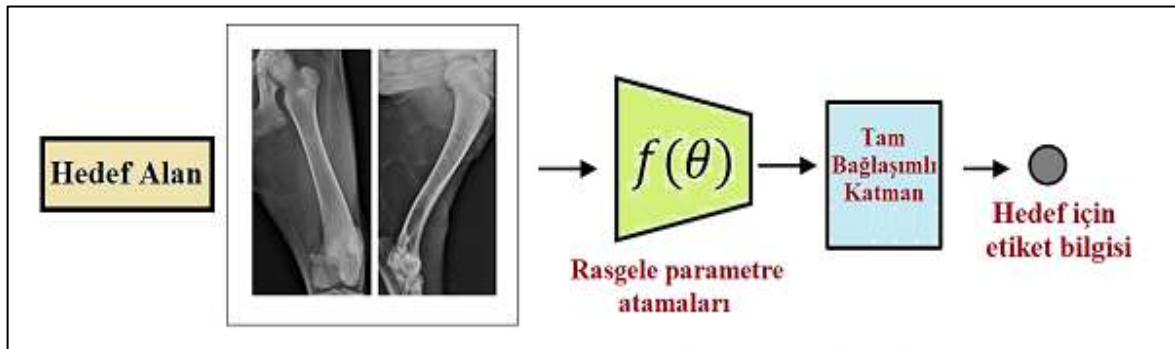
2.4.2. Öğrenme aktarımı (Transfer learning) tabanlı sınıflandırma

Öğrenme aktarımı, insan beyninin öğrenme şekline dayanarak geliştirilmiştir. Bu sınıflandırma yöntemi, bir problem çözümü sırasında elde ettiği bilgileri saklamakta ve elde ettiği bu bilgileri yeni bir problem ile karşılaştığında kullanmaktadır. Bu duruma öğrenme aktarımı denmektedir. Bu yöntem sayesinde daha az veri ile eğitim yapılabilir. Öğrenme aktarımında, direkt olarak önceki problemde elde edilen özellik ve ağırlıklar yeni probleme aktarılabilir. Bu sayede zamandan tasarruf sağlanmasının yanında, daha yüksek başarı ile hızlı öğrenen bir model ortaya çıkmaktadır. Şekil 2.17’de geleneksel makine öğrenmesi ve öğrenme aktarımı yöntemi ile çalışan sistemin temel mimarileri verilmiştir [106].

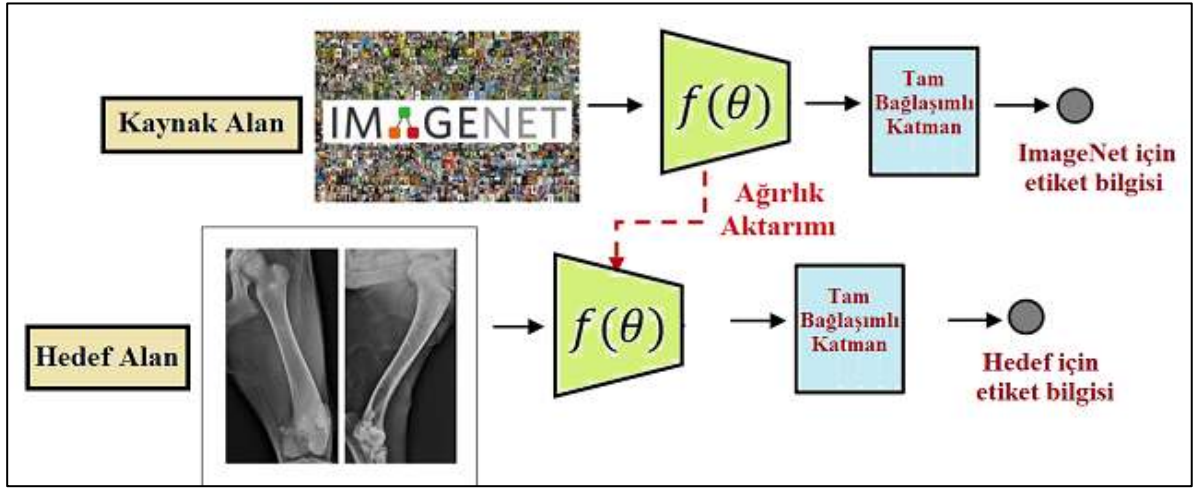


Şekil 2.21. Geleneksel makine öğrenmesi ve öğrenme aktarımı yöntemleri [106].

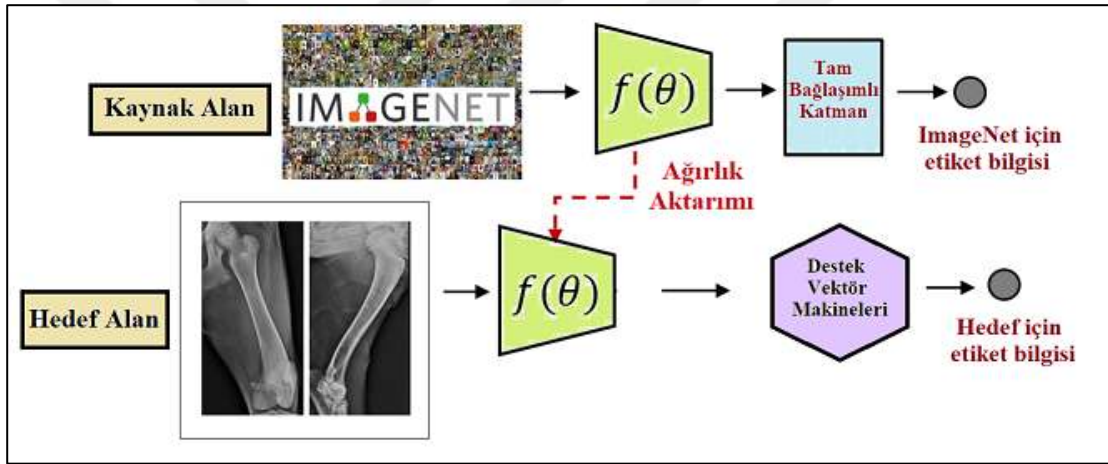
Geleneksel makine öğrenmesi yöntemleri, her problem için yeniden bir öğrenme işlemi gerçekleştirmektedir. Bu durum, ‘sıfırdan öğrenme’ olarak tanımlanmaktadır. Ancak, belirli bazı problemlerin çözümünde kaynak görevden elde edilen özellik gibi bilgiler öğrenilerek, hedef görevin çözümünde kullanılabilir. Öğrenme aktarımı yönteminde, öğrenilen bilgiler sadece kaynak göreve ait olmayacak şekilde genel bilgi olmalıdır. Buradan da anlaşılacağı gibi öğrenilecek olan bilgiler kaynak ve hedef görevler için ortak olmalıdır [106]. Şekilde 2.18’de sıfırdan öğrenme, Şekil 2.19’da tam bağımlı katman ile sınıflandırılan öğrenme aktarımı ve Şekil 2.20’de destek vektör makineleri ile sınıflandırılan öğrenme aktarımı yönteminin görselleri verilmiştir.



Şekil 2.22. Sıfırdan öğrenme [107].



Şekil 2.23. Tam bağlaşımlı katman ile öğrenme aktarımı [107].



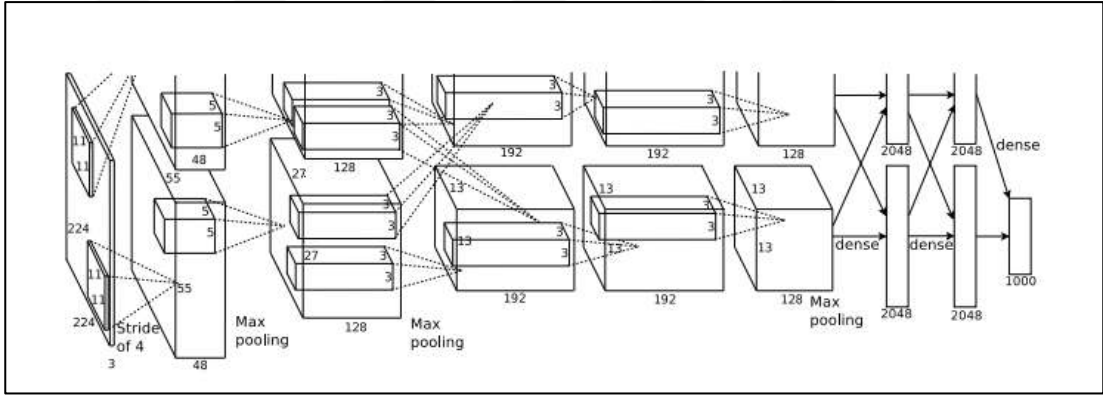
Şekil 2.24. Destek vektör makineleri ile öğrenme aktarımı [107].

Bu çalışma kapsamında, öğrenme aktarımı yöntemi ile ilgili olarak farklı modeller kullanılmıştır. Bu modeller aşağıda listelenmiştir:

- AlexNet,
- ResNet-50,
- DenseNet-201,
- Inception v3,
- Xception.

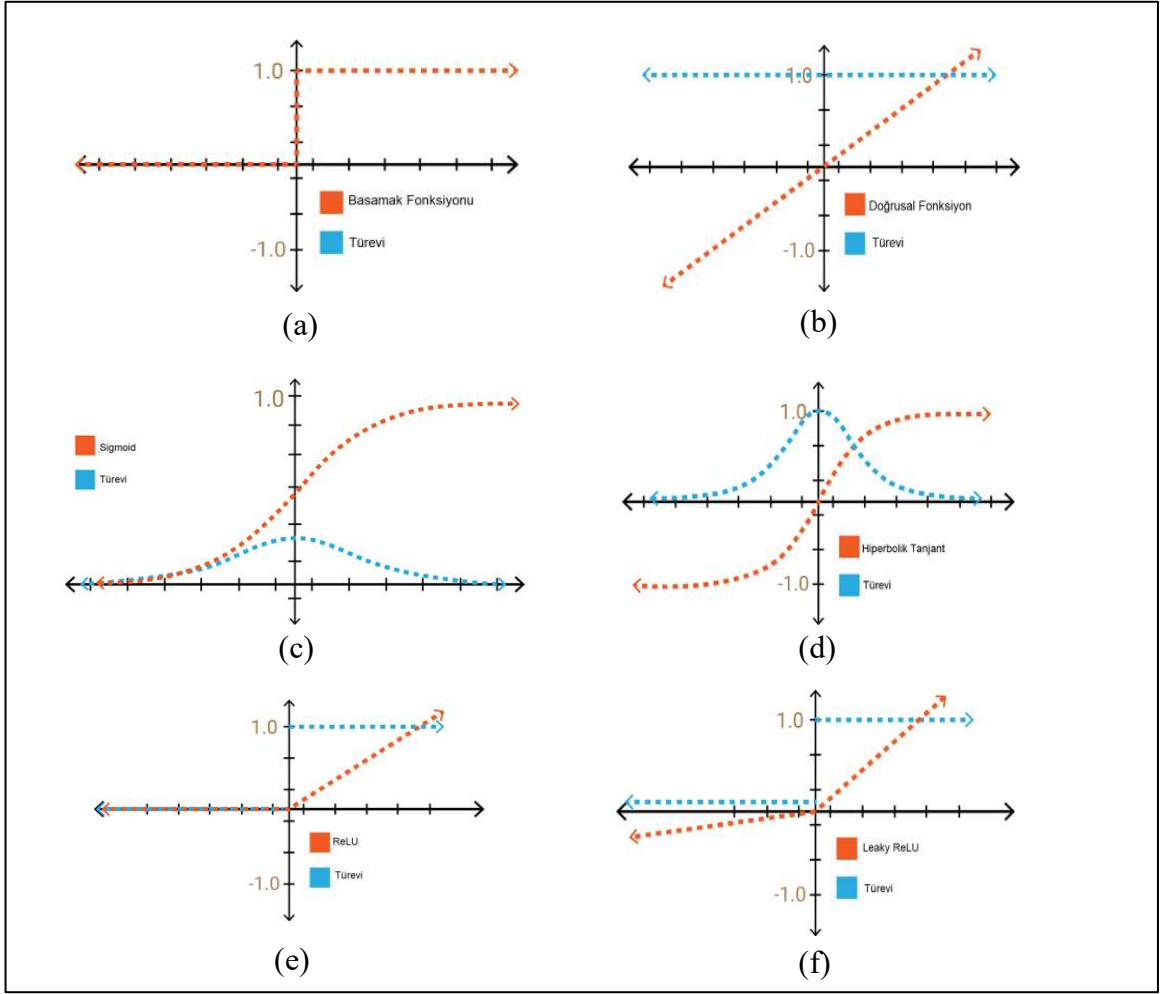
2.4.2.1. AlexNet

AlexNet, 2012 yılında Alex Krizhevsky öncülüğünde ve I. Sutskever ile G. Hinton'un yardımları ile geliştirilmiş bir CNN mimarisidir [108]. Bu ağ, LeNet ile çok benzer bir mimariye sahiptir. Ancak, LeNet'e göre daha fazla filtre ve yığılmış evrişim katmanları ile daha derin bir ağ modeli oluşturmaktadır. AlexNet mimarisi, 5 evrişim katmanı ve 3 tam bağlaşım katmanından oluşmaktadır. Şekil 2.25'te 60 milyon parametreye sahip AlexNet mimarisi verilmiştir. Bu sinir ağı yapısında 11x11, 5x5 ve 3x3 boyutlu filtreler kullanılmıştır [108]. Ayrıca, AlexNet mimarisi ReLU aktivasyon fonksiyonu kullanan ilk derin ağ modelidir. ReLU aktivasyon fonksiyonu, herhangi bir negatif girdi alırsa 0 değerini döndürmektedir. Ancak, herhangi bir pozitif x değeri için bu değeri geri döndürmektedir [109]. Böylece sistem 0 ile sonsuz arasında değişen bir çıktı vermektedir.



Şekil 2.25. AlexNet mimarisi [108].

Aktivasyon işlemi, bir sinir ağında düğümden alınan toplam ağırlıklı girdiyi, bu girdi için belirli aralıklara sahip bir çıktıya dönüştürmek demektir [110]. Eğer aktivasyon fonksiyonu kullanılmazsa, çıkışta basit bir lineer fonksiyon halinde sinyal elde edilmektedir. Buradan da anlaşılacağı gibi aktivasyon fonksiyonu kullanılmayan bir sinir ağı, sınırlı öğrenme gücü ile doğrusal bir regresyon olarak hareket etmektedir [109]. Literatürde, basamak (step), doğrusal (linear), sigmoid, hiperbolik tanjant (tanh), sızıntı ReLU (Leaky ReLU) gibi aktivasyon fonksiyonları bulunmaktadır. Şekil 2.26'de bu fonksiyonlar ile ilgili görsel verilmiştir.



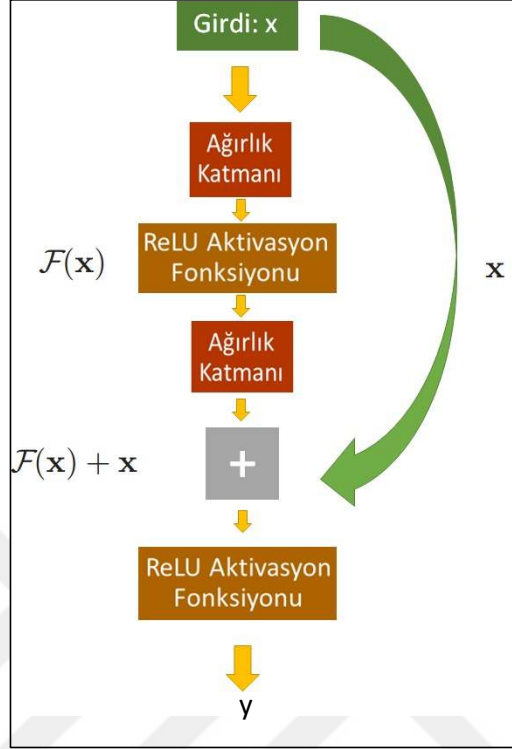
Şekil 2.26. Sinir ağlarında kullanılan aktivasyon fonksiyonları ve türevleri. a) Basamak fonksiyonu, b) Doğrusal fonksiyon, c) Sigmoid fonksiyonu, d) tanh fonksiyonu, e) ReLU fonksiyonu ve f) Sızıntı ReLU fonksiyonu [111].

2.4.2.2. ResNet

ResNet, ilk olarak 2015 yılında Kaiming He, Xiangyu Zhang, Shaoqing Ren ve Jian Sun tarafından tanıtılan bir CNN mimarisidir [112]. Bu derin ağ modeli, artık blok (residual block) olarak adlandırılan yeni bir blok eklendiği için Artık Ağ ismi ile de tanımlanmaktadır. [113].

Bu modelin temel yaklaşımı, katmanların temel haritalanmayı öğrenmesi yerine, ağın atlama bağlantısı ile iki katman arasındaki bazı katmanların işlenmesine engel olacak kısa bir yol eklemektir. Bu şekilde bir atlama bağlantısı kullanımı ile işlem sırasında herhangi bir

katmanın mimarinin performansını olumsuz bir şekilde etkilemesi durumunda, atlama ile bu durum ortadan kaldırılmaktadır. Şekil 2.27’de artık katmanın mimarisi verilmiştir [114].

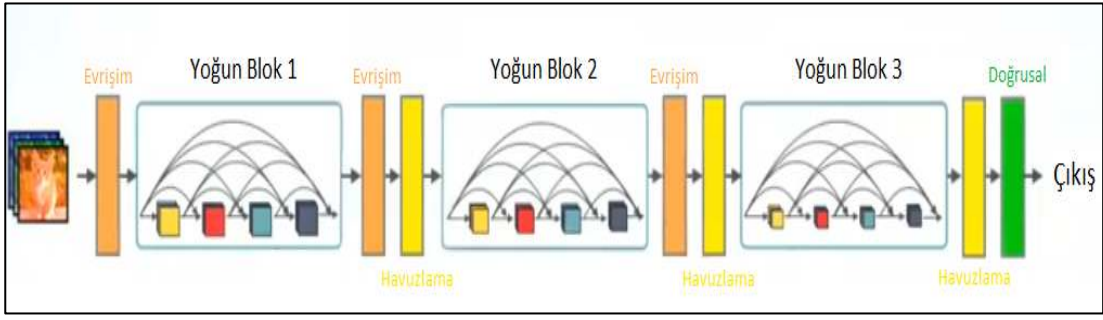


Şekil 2.27. ResNet mimarisindeki artık blok yapısı [107].

ResNet’te kullanılan artık blok yapısı, gradyanın akması için blok yapısında bulunan kısa yolu kullanarak derin ağlarında bulunan kaybolan gradyan sorunu çözmek için geliştirilmiştir. Kaybolan gradyan sorunu, eğitim aşamasında ortaya çıkan hata sinyalinin çok küçük değerler almasından dolayı ağırlıkların gerekli bir şekilde güncellenememesidir ve aynı zamanda eğitim işleminin sonlanması olarak tanımlanmaktadır [114]. ResNet mimarisi, farklı sayılarda katmanlardan oluşan farklı sürümleri mevcuttur. Bunlar; ResNet-18, ResNet-50 ve ResNet-101’ dir.

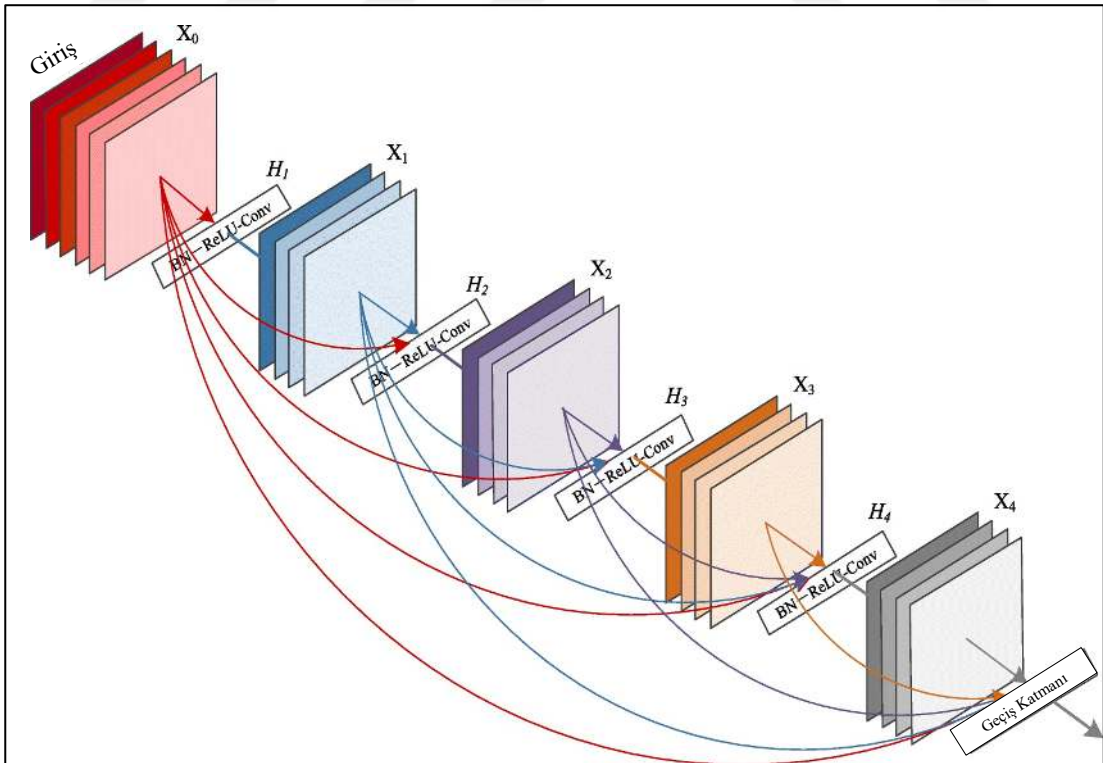
2.4.2.3. DenseNet-201

DenseNet, 2018 yılında G. Huang, Z. Liu, L. Maaten, ve K. Q. Weinberger tarafından geliştirilmiştir [115]. ResNet mimarisinde ortaya çıkan artık blok yapısı, DenseNet mimarisinde yoğun blok (dense block) olarak ortaya çıkmaktadır. DenseNet modelinde; 121, 169, 201 ve 161 derin katmandan oluşan dört farklı ağ yapısı bulunmaktadır. Şekil 2.28’de DenseNet modeline ait yapı katmanları verilmiştir [115].



Şekil 2.28. DenseNet mimarisinin yapı katmanları [115].

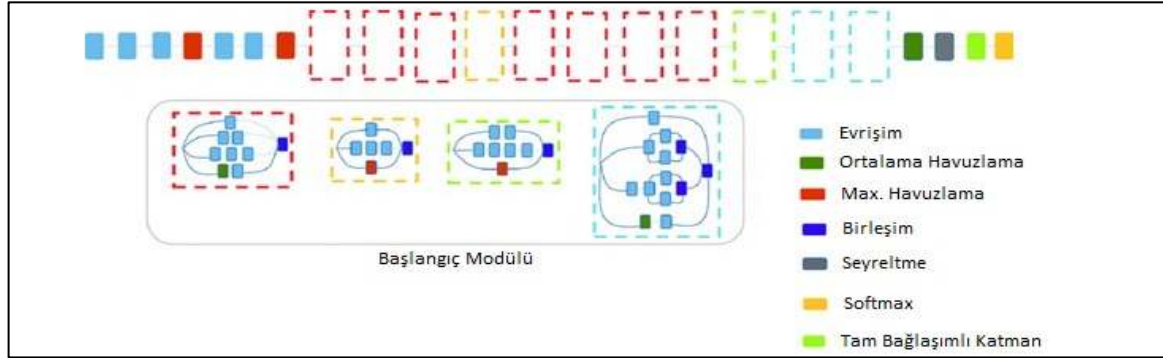
Bir DenseNet mimarisinde bulunan her katman diğer tüm katmanlar ile bağlantılıdır. Bu nedenle, bu mimariye Yoğun Bağlantılı Evrişimsel Ağ (Densely Connected Convolutional Network) adı verilmiştir. Şekil 2.29’da DenseNet mimarisinin örneği verilmiştir. Bu mimari, bazı temel farklar ile ResNet modeline benzemektedir. ResNet mimarisinde, özellik haritalarını içeren katman, bir sonraki katman ile birleştiren bir toplama yöntemi kullanılırken, DenseNet önceki katmanın çıktısını gelecek katman ile birleştirmektedir. Sistemin en büyük avantajları; kaybolan gradyan sorununu hafifletmek, özellik yayılımını güçlendirmek, özelliklerin yeniden kullanımını teşvik etmek ve parametre sayısını önemli ölçüde azaltmaktır [115], [116].



Şekil 2.29. DenseNet mimarisi [115].

2.4.2.4. Inception v3

Inception v3, 2016 yılında C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens ve Z.Wojna tarafından geliştirilmiştir [117]. Bu mimarinin yapısı Şekil 2.30’da verilmiştir [69].



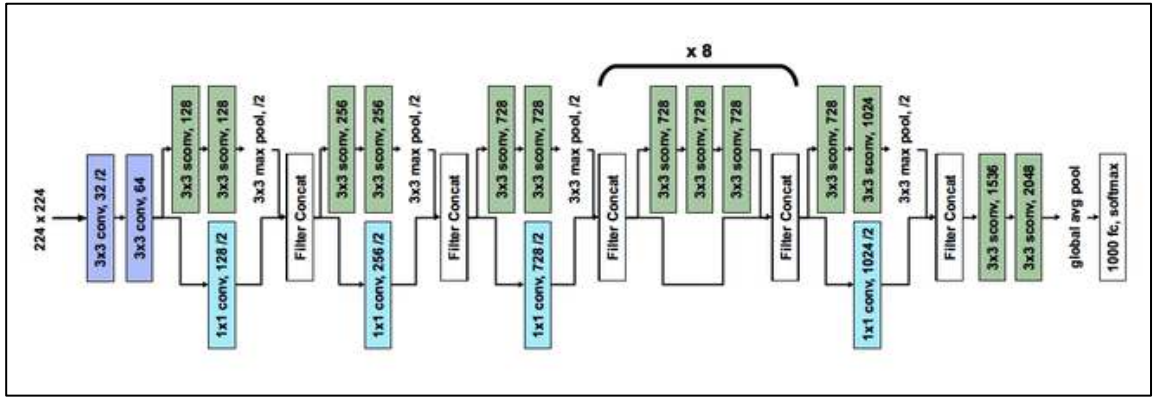
Şekil 2.30. Inception v3 mimarisi [69].

Bu mimari, Google'ın Evrişimsel sinir ağında üçüncü baskısıdır. Inception v3 tasarımında parametre sayısının çok fazla büyümesi engellenirken aynı zamanda derin ağların kullanılmasını amaçlanmıştır. Inception v1 ve Inception v2 sürümlerine göre eklenen bazı katmanlar ile başarı artırımı sağlanmıştır. Bu katmanlar, küme normalizasyonu, evrişim katmanları ve tam bağlaşımlı katmanlardır.

2.4.2.5. Xception

Xception modeli, 2017 yılında François Chollet tarafından ortaya atılmıştır. Xception mimarisi, standart Inception mimarilerini derinlemesine ayrılabilir kıvrımlarla değiştiren Inception mimarisinin bir uzantısı olarak geliştirilmiş bir modeldir. Bu model, Inception v3 modelinden daha iyi bir performans göstermektedir. Şekil 2.31’de Xception’a ait mimari verilmiştir [118], [119].

Xception modeli, GoogLeNet ile yaklaşık olarak sayıda parametreye sahiptir. Ayrıca, bu model ve Inception v3 aynı sayıda parametreye sahip olmasına rağmen Xception mimarisi daha yüksek bir başarı göstermektedir. Bunun sebebi ise Xception mimarisinde parametrelerin daha verimli kullanılmasıdır. Xception mimarisi 3 ana bölümden oluşmaktadır: giriş akışı, orta akış ve çıkış akışıdır [118], [119].

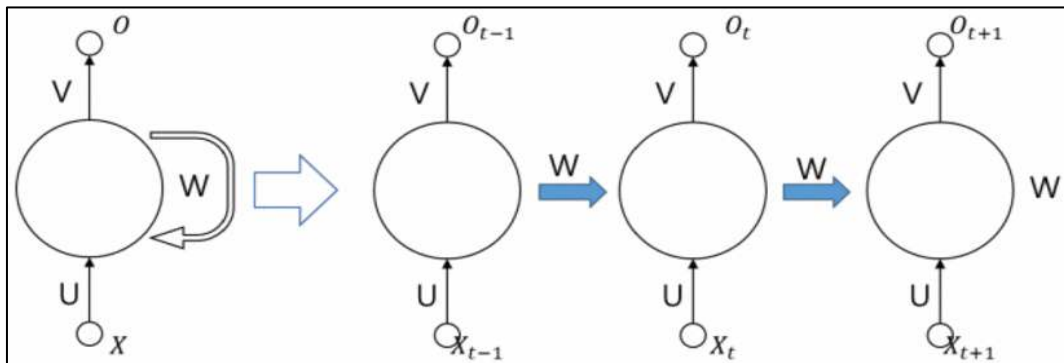


Şekil 2.31. Xception mimarisi [118].

2.4.3. Yinelemeli sinir ağı (RNN)

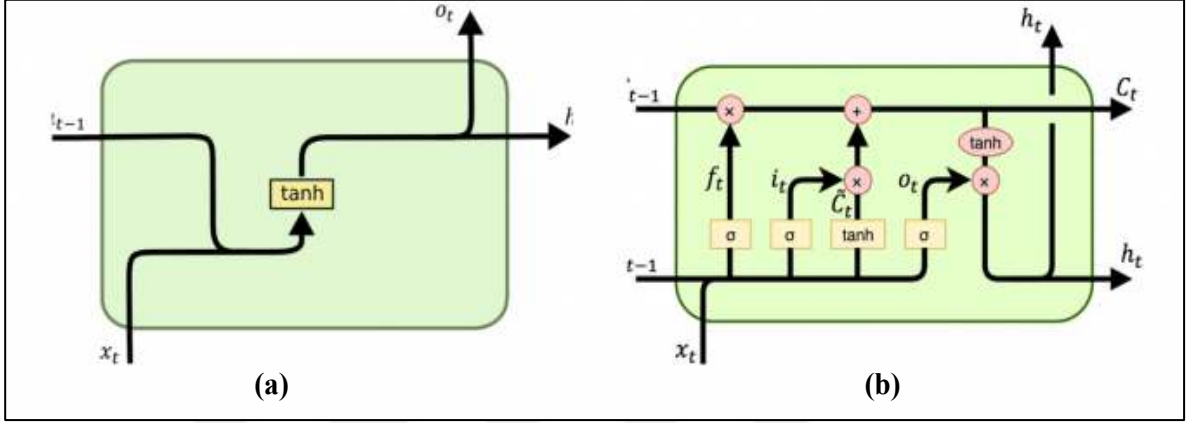
Yinelemeli sinir ağı, 1986 yılında David Rumelhart'ın çalışması ile ortaya çıkmıştır [120]. Şekil 2.32'de RNN mimarisi verilmiştir. RNN algoritması, tekrarlı ve sıralı veri üzerinde öğrenme işleminin gerçekleştirilmesini için geliştirilmiş bir yapar sinir ağı çeşididir [121].

Geleneksel sinir ağı yapısı, var olan girdi üzerinden hedef vektöre eşlenebilen arka arkaya gelen katmanlar arasındaki tam bağlantı ile karakterize edilmektedir. Ancak, RNN algoritmasında girdiler yalnızca anlık değil, önceki girdilere göre de değerlendirilmektedir. Buradan anlaşılacağı gibi, t zamanında elde edilen RNN algoritma çıktısı, $t-1$ zamanında elde edilen RNN çıktısından etkilenebilir. Bu nedenle, sürekli ve sıralı dizi verilerinin modellenmesinde geleneksel sinir ağlarına göre daha yüksek bir performans göstermektedirler [122].



Şekil 2.32. RNN mimarisi [121].

Yinelemeli sinir ağının dezavantajı, sürekli dizi verilerinin modellenmesinde çok etkili olmasına rağmen, geri yayılıma dayalı model eğitimi, uzun dizi modellenirken gradyan kaybolması (Gradient Vanishing) ve patlama [123] problemiyle karşılaşma durumudur. Bu problemten dolayı LSTM algoritması geliştirilmiştir. Şekil 2.33'te RNN ve LSTM mimarilerinin yapıları verilmiştir.



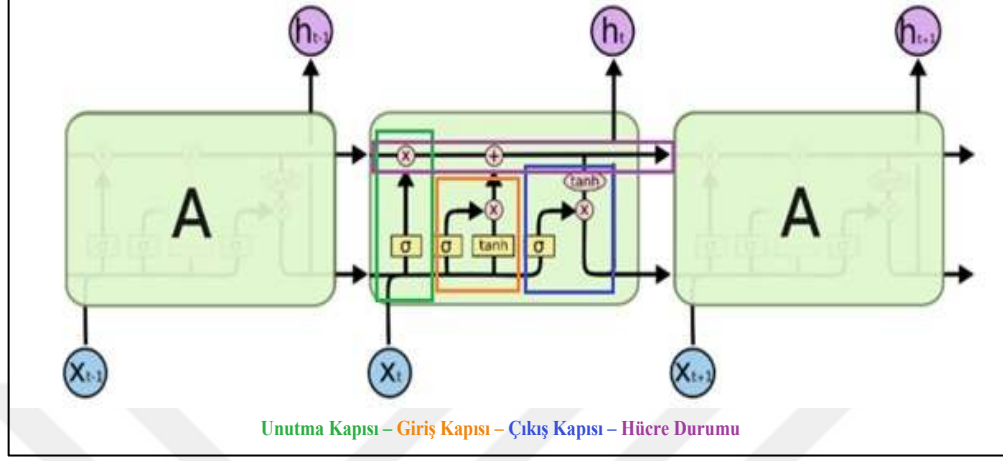
Şekil 2.33. (a) RNN mimarisi, (b) LSTM mimarisi [124].

2.4.4. LSTM

LSTM ağı, 1991 yılında Hochreiter ve arkadaşları tarafından, öğrenmeyi ek alan bilgisi olmadan model eğitimi ile birleştiren bir RNN çeşidi olarak geliştirilmiştir [123]. RNN, ardışık olaylar arasında ilişkiler türetmekte başarılı olmaktadır. Ancak, gradyan kaybolması nedeni ile uzun vadeli olayları ilişkilendirirken yetersiz kalmaktaydı. LSTM de bu problemin çözümü için geliştirilmiştir [125]. Bu çözüm ile, LSTM'nin doğrusal olmayan dinamikleri modellemede bir avantaj sahibi olmasını sağlamıştır. Ayrıca, adından da anlaşılacağı üzere uzun dizi verileri ile işlem yaparken büyük bir avantaj sağlamıştır [125]. Şekil 2.34'te kapılar ile LSTM mimarisi verilmiştir.

LSTM mimarisi, sinyal, ses, metin ve diğer zaman serileri gibi dizi verilerinde uzun vadeli bağımlılıkları öğrenmeyi sağlamaktadır. LSTM, hangi verilerin uzun süre saklanması, hangi verilerin atılması veya hangi verilerin okunması gerektiğini anlayabilen bir algoritmadır [125]. Bunun için, bir bellek hücresi ekleyerek, belleği belirli bir süre tutmaktadır. Eklenen bu bellek hücresi; giriş kapısı, unutma kapısı ve çıkış kapısı olarak belirlenen bölgeler tarafından kontrol edilmektedir [126]. Giriş kapısı ile bellek hücresine

giriş bilgisi aktif hale getirilir [127]. Unutma kapısı, bellek hücresindeki bazı bilgileri yok etmektedir ve bir sonraki giriş için depolamayı etkinleştirmektedir [127]. Çıkış kapısı ise bellek hücresi tarafından hangi bilgilerin çıkarılacağına karar vermektedir [127].



Şekil 2.34. LSTM mimarisi [128].

LSTM mimarisinde sistemin başarısını arttırmak için mimaride kullanılan bazı parametreler üzerinde değişimler yapmak gerekmektedir. Bu parametreler aşağıda verilmiştir [129], [130], [131].

a. Optimize edici: LSTM mimarisinde optimize ediciler modelin doğruluğunu arttırmak için kullanılırlar. Literatürde mevcut olan pek çok optimize edici bulunsada en çok kullanılanları; SGDM, RMSProp ve Adam'dır.

SGDM, artımlı gradyan inişi olarak tanımlanmaktadır. Bu optimize edici, yineleme yolu ile minimum veya maksimum hata bulunmasını amaçlamaktadır. Bu optimize edicinin dezavantajı ise, tanımlanan amaç fonksiyonu dışbükey değilse, yerel bir minimuma yakınsamasının neredeyse kesin olmasıdır.

RMSProp ise gradyan tabanlı bir optimizasyon tekniğidir. Bu yöntemde, Gama ve öğrenme oranını (n) sırası ile 0,9 ve 0,001 olarak ayarlanması gerekmektedir.

Adam, birinci dereceden gradyanlar gerektiren ve az bellek ihtiyacı olan verimli stokastik optimizasyonlardan birisidir. Adam, LSTM'de diğer optimize edicilerden daha doğru bir sınıflandırma sonucu elde edilmesini sağlamaktadır.

b. Mini Yığıt Boyut (Mini Batch Size): Çözücüler her adımda verilerin bir alt kümesini kullanarak parametreleri güncellemektedir. Bu alt kümeye mini parti (mini batch) adı verilmektedir. Mini partinin boyutu, mini yığıt boyut parametresi ile belirlenmektedir.

c. Maksimum Eğitim Turu (Max Epochs): Bir yineleme, her parametrenin güncellenmesi olarak tanımlanmaktadır. Bir dönem ise, eğitim algoritmasının tüm eğitim seti üzerinden tam geçişi olarak tanımlanmaktadır. Maksimum eğitim turu, bir pozitif tam sayıdan oluşan virgülle ayrılmış çift olarak belirtilen, eğitim için kullanılacak maksimum dönem sayısıdır.

d. Gizli Birim Sayısı (Number of Hidden Units): LSTM diyagramı yalnızca gizli hücreleri göstermektedir, ancak bu hücrelerin birimlerini asla göstermemektedir. Her gizli katman, zaman adımı sayısı kadar gizli hücrelere sahiptir. Ayrıca, her gizli hücre, aşağıdaki şemada olduğu gibi, birden çok gizli birimden oluşmaktadır. Bu nedenle, LSTM'deki bir gizli katman matrisinin boyutluluğu zaman adımı sayısı, gizli birim sayısı (number of time steps, number of hidden units) 'dır.

2.5. Sınıflandırma Performansını Ölçmede Kullanılan Teknikler

Sınıflandırma, bir veri setinde bulunan herhangi bir verinin ait olduğu kategoriyi tahmin etme işlemidir. Sınıflandırma işlemi için kullanılabilecek birden çok sınıflandırma metodu bulunmaktadır. Sınıflandırıcılar aynı durum altında aynı performansı göstermezler. Bu performansları değerlendirmek için çeşitli teknikler kullanılmaktadır. Bu teknikler aşağıda listelenmiştir:

- Karışıklık matrisi (confusion matrix),
- Doğruluk (accuracy),
- Eğri altındaki alan (AUC),
- Alıcı çalışma karakteristik eğrisi (ROC),
- Hassasiyet, geri çağırma,
- F1- puanı.

Bu tez çalışması kapsamında sınıflandırma performanslarını karşılaştırmak için karışıklık matrisi ve doğruluk kullanılmıştır.

a. Karışıklık matrisi [132]: Karışıklık matrisi literatürde karmaşıklık matrisi olarak da kullanılmaktadır. Bir veri setinde bulunan tüm sınıflar için ayrı ayrı doğru ve yanlış olan tahminlerin sayılarını karşılaştıran bir matris ile sistemin performansını gözlemlemeyi sağlamaktadır. Basit bir iki sınıflı veri setindeki karmaşıklık matrisi Şekil 2.35’te verilmiştir.

Bir karmaşıklık matrisi dört terimden oluşmaktadır: TP, FP, FN ve TN’dir. Bu terimler iki sınıflı ve sınıf bilgisi 0 ve 1 değerlerinden oluşan bir sistem için aşağıda açıklanmıştır.

Gerçek negatif (True Negative): Sınıflandırıcının sonucunda elde edilen sonuç doğru tahmin değeri içermektedir ve bu değer negatiftir. Bir başka deyişle, sınıf bilgisinin değeri gerçekte 0’dır ve sınıflandırıcı tarafından da bu değer 0 olarak belirlenmiştir.

Gerçek pozitif (True Positive): Sınıflandırıcının sonucunda elde edilen sonuç doğru tahmin değeri içermektedir ve bu değer pozitifdir. Bir başka deyişle, sınıf bilgisinin değeri gerçekte 1’dir ve sınıflandırıcı tarafından da bu değer 1 olarak belirlenmiştir.

Yanlış pozitif (False Positive): Sınıflandırıcı tarafından negatif olan değer pozitif olarak belirlenmesi ile yanlış tahmin işleminin gerçekleştirilmesidir. Bir başka deyişle, sınıf bilgisinin gerçek değeri 0 iken sınıflandırıcı tarafından 1 olarak belirlenmiştir.

Yanlış negatif (False Negative): Sınıflandırıcı tarafından pozitif olan değer negatif olarak belirlenmesi ile yanlış tahmin işleminin gerçekleştirilmesidir. Bir başka deyişle, sınıf bilgisinin gerçek değeri 1 iken sınıflandırıcı tarafından 0 olarak belirlenmiştir.

		GERÇEK	
		Pozitif	Negatif
TAHMIN	Pozitif	TP	FP
	Negatif	FN	TN

Şekil 2.35. Karmaşıklık matrisi [66].

Yukarıda tanımlanan dört terim kullanılarak sistemin başarısının hesaplanması (2.4)'te verilmiştir.

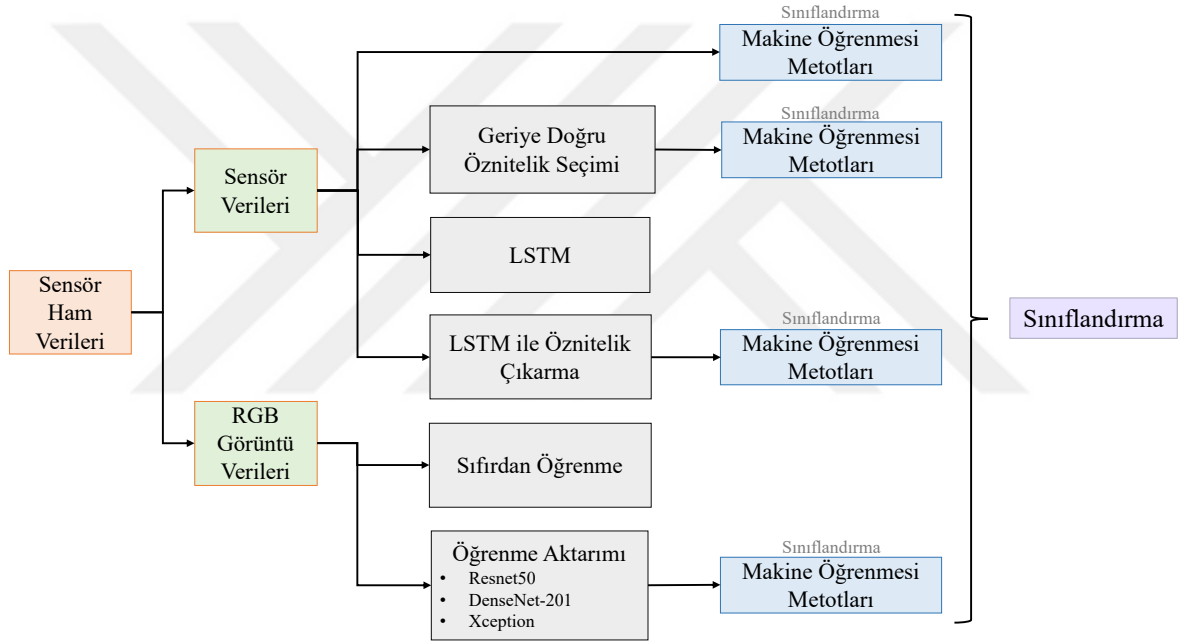
$$\text{Sınıflandırma Başarısı} = \frac{TP+TN}{TP+TN+FN+FP} \quad (2.4)$$

b. Doğruluk (Accuracy) [133]: Bir veri setinde kullanılan sınıflandırıcının performansını değerlendirmek için doğruluk yöntemi oldukça basittir. Bir modelin doğruluğu, doğru tahminlerin toplam tahmin sayısına bölümü ile elde edilmektedir. Sınıflandırma başarısının formülü (2.5)'te verilmiştir. Elde edilen bu doğruluk değeri 0 ve 1 arasında bir değerdir. Değeri yüzde olarak değerlendirmek için 100 ile çarpılmaktadır.

$$\text{Sınıflandırma Başarısı} = \frac{\text{Doğru Tahminler}}{\text{Toplam Tahminler}} \times 100 \quad (2.5)$$

3. YAPILAN ÇALIŞMALAR VE SONUÇLAR

Bu tez çalışması kapsamında kullanılan veri seti Jaeyong Sung ve arkadaşlarının [13] çalışmasında kullanılan CAD60 olarak bilinen hazır veri setidir. Veri setinde dört farklı kişiden elde edilen RGB/derinlik görüntüleri ve bunlarla ilgili eklem konum bilgileri bulunan veri dosyaları bulunmaktadır. Dört farklı kişiden elde edilen görüntü verilerinin sayısı sıra ile; 17.918, 16.805, 17.577 ve 20.360'tır. Toplamda 72.660 veri üzerinde çalışmalar gerçekleştirilmiştir. Veri setinde, derinlik görüntülerinden elde edilen iskelet verileri 15 eklemden oluşmaktadır. Veri seti ile ilgili detaylı bilgi bölüm 2.1'de verilmiştir. Şekil 3.1'de yapılan çalışmaları özetleyen akış şeması verilmiştir.



Şekil 3.1. Yapılan çalışmaları özetleyen akış diyagramı.

3.1. Makine Öğrenmesi Yöntemleri

Makine öğrenmesinde bulunan farklı mimariler ile veri seti üzerinde sınıflandırma başarısı test edilmiştir. Makine öğrenmesi yönteminde Prime Sensor NITE 1.3 algoritması ile elde edilen 15 ekleme ait eklem verileri kullanılarak sınıflandırma işlemi yapılmıştır. Bu tez çalışmasında kullanılan veri setinde dört farklı kişiden elde edilen veriler bulunmaktadır. Bu kişiler 1, 2, 3 ve 4 olarak numaralandırılmıştır. Öğrenme tabanlı yöntemler, modeli oluşturma ve sınamaya aşamasında eğitim ve test setlerine ihtiyaç duymaktadırlar. Kullanılan veri setinde bu ayrımı en verimli şekilde yapabilmek adına farklı denemeler ile veri setinde

bulunan 3 kişinin verileri ile eğitim seti, geriye kalan kişinin verileri ile test seti oluşturulmuştur. Örneğin; 1, 2, 3 numaralı kişilerin verileri ile eğitim seti oluşturulurken, 4 numaralı kişinin verileri ile test seti oluşturulmuştur. Eğitim ve test setleri toplamda 4 farklı şekilde ayrılmıştır. Her bir ayırım için makine öğrenmesi tabanlı farklı metotlar kullanılarak maksimum sınıflandırma başarıları elde edilmiştir. Her bir ayırmada elde edilen test (sınıflandırma) başarıları Tablo 3.1’de verilmiştir.

Tablo 3.1 Makine öğrenmesi yönteminde veri seti ayrımları için test başarıları (%)

Model Numarası	Model İsim	Eğitim Seti:123 Test Seti: 4	Eğitim Seti:124 Test Seti: 3	Eğitim Seti:134 Test Seti: 2	Eğitim Seti:234 Test Seti: 1
1.1	İnce Ağaç (Fine Tree)	48,5	20,4	70,9	55,5
1.2	Orta Ağaç (Medium Tree)	55,1	34	71,1	59,7
1.3	Kaba Ağaç (Coarse Tree)	37,1	12,7	45,9	34,2
1.4	Doğrusal Diskriminant (Linear Discriminant)	74,6	53,8	87,2	58,9
1.5	Doğrusal SVM (Linear SVM)	64,3	54,8	79,4	63,1
1.6	İkinci Dereceden SVM (Quadratic SVM)	67,5	47,3	73,3	62,7
1.7	Kübik SVM (Cubic SVM)	70,5	47,2	66,7	63,3
1.8	İnce Gauss SVM (Fine Gaussian SVM)	9,2	8,6	8,6	9,7
1.9	Orta Gauss SVM (Medium Gaussian SVM)	42,9	30,2	49,8	48,3
1.10	Kaba Gauss SVM (Coarse Gaussian SVM)	65,3	50,8	78,7	55,7

1.11	İnce KNN (Fine KNN)	71,5	51,2	91,1	71,2
1.12	Orta KNN (Medium KNN)	72	52,1	91,1	71,4
1.13	Kaba KNN (Coarse KNN)	71,5	54,6	89,3	76,7
1.14	Kosinüs KNN (Cosine KNN)	74,5	54,4	91,6	69,7
1.15	Ağırlıklandırılmış KNN (Weighted KNN)	71,8	52	91,1	71,4
1.16	Güçlendirilmiş Ağaçlar (Boosted Trees)	45,9	53,9	65,2	52,9
1.17	Torbalı Ağaçlar (Bagged Trees)	67,7	43	75,6	60,6
1.18	Alt Uzay Diskriminant (Subspace Discriminant)	81,1	49,8	76,7	74,1
1.19	Yükseltilmiş Ağaçlar (RUSBoosted Trees)	47,8	47,2	67,6	52,8
1.20	Dar Sinir Ağı (Narrow Neural Network)	50,9	52,7	69,8	46,5
1.21	Orta Sinir Ağı (Medium Neural Network)	64,5	51,8	90,7	52,5
1.22	Geniş Sinir Ağı (Wide Neural Network)	63,8	46,7	70,5	70,5
1.23	İki Katmanlı Sinir Ağı (Bilayered Neural Network)	34,6	39,3	70,4	56,9
1.24	Üç Katmanlı Sinir Ağı (Trilayered Neural Network)	59,8	38,6	45,5	55
1.25	SVM Çekirdeği (SVM Kernel)	44,2	40,2	48,3	48,2
1.26	Lojistik Regresyon Çekirdeği (Logistic Regression Kernel)	50,6	49,6	70,9	52,6

ML yönteminde 26 farklı metot kullanılarak sistemin başarısı elde edilmiştir. Eğitim seti:123 ve test seti:3 için en yüksek sınıflandırma başarısı Alt Uzay Diskriminant yönteminde %79,6 olarak elde edilmiştir ve bu sınıflandırma başarısı için karışıklık matrisi Şekil 3.2’de verilmiştir.

1	171		1136										
2		1792											2
3	172		336		1019								
4				1739	126								
5					1467	113							
6					225	875							
7							1812						
8								1853					
9				1350					548	12			
10		1		10					137	1687			
11			76			6					920		10
12												1662	
13													1102
	1	2	3	4	5	6	7	8	9	10	11	12	13

Şekil 3.2. Eğitim seti: 123 için en yüksek başarı değerinde karışıklık matrisi.

Şekil 3.2 incelendiğinde, bu veri seti ayrımında bulunan 4 hareketin sınıflandırılmasında hata gerçekleşmemiştir. Bunlar; sandalyede dinlenmek, sandalyede konuşmak, bilgisayarda çalışmak ve hareketsiz durmaktır. Bu hareketlerin sıra ile etiket değerleri; 7, 8, 12 ve 13’tür.

Eğitim seti: 124 ve test seti:3 için sistemin sınıflandırma başarısına bakıldığında en yüksek başarı değeri Doğrusal SVM (Linear SVM)’de %54,8 olarak bulunmuştur. Önceki çalışmada elde edilen sonuç ile karşılaştırıldığında bu başarı değeri düşük olarak gözlemlenmektedir. Bu başarı değeri ile ilgili Karışıklık matrisi Şekil 3.3’te verilmiştir.

1			927						52		308		
2									1597				
3											1310		
4				2	116	81			1304				
5					1783								
6					660	162							
7							1684		28				
8							1369	10					
9									1721	33			
10			118						46	1303			
11						6					620		1
12												1222	
13													1119
	1	2	3	4	5	6	7	8	9	10	11	12	13

Doğru Sınıf

Tahmin Edilen Sınıf

Şekil 3.3. Eğitim seti: 124 için en yüksek başarı değerinde karışıklık matrisi.

Bu veri seti ayrımının karışıklık matrisi incelendiğinde sadece 2 hareketin hatasız bir şekilde sınıflandırıldığı görülmektedir. Bu hareketler, 12 ve 13 numaralı etiket değerlerine sahip bilgisayarda çalışmak ve hareketsiz durmaktır.

Veri setinin üçüncü ayrımında eğitim seti 1, 3, 4 ve test seti 2 numaralı kişilerin verileri ile oluşturulmuştur. Bu veri seti ayrımı için en yüksek başarı değeri Kosinüs KNN (Cosine KNN) metodunda %91,6 olarak bulunmuştur. Bu başarı değeri ile ilgili Karışıklık matrisi Şekil 3.4'te verilmiştir.

1			829										
2		1618							19				
3			679						2			97	
4				1437		9							
5	82				1593								
6	5			8	40	1277				85			
7							1539						
8								1497					
9		102							1503	59			
10		11	22							1316			
11			29								922		12
12												1530	
13			5							1			476
	1	2	3	4	5	6	7	8	9	10	11	12	13

Tahmin Edilen Sınıf

Doğru Sınıf

Şekil 3.4. Eğitim seti: 134 için en yüksek başarı değerinde karışıklık matrisi.

Eğitim seti:134 ve test seti:2 için elde edilen karışıklık matrisinde 3 hareketin sınıflandırılması hatasız bir şekilde gerçekleştirilmiştir. Bu hareketler sıra ile; sandalyede konuşmak (7 numaralı etiket), sandalyede dinlenmek (8 numaralı etiket) ve bilgisayarda çalışmaktır (12 numaralı etiket).

Veri setinin son ayrımında eğitim seti 2, 3, 4 ve test seti 1 numaralı kişilerin verileri ile oluşturulmuştur. Tablo 3.1’de görüldüğü üzere en yüksek sınıflandırma başarısı Alt Uzay Diskriminant (Subspace Discriminant) metodunda %74,1 olarak bulunmuştur. Bu başarı değeri ile ilgili Karışıklık matrisi Şekil 3.5’te verilmiştir.

1				66	1458								
2	1792												
3	418				720					394		55	
4		4		1601	1			140					
5					741	601				9			
6					208	627							
7							1681						
8								1447					
9									1442	123			
10									59	1258	29		
11						322					399		28
12												1265	
13													1029
	1	2	3	4	5	6	7	8	9	10	11	12	13

Şekil 3.5. Eğitim seti: 234 için en yüksek başarı değerinde karışıklık matrisi.

Bu veri seti ayrımının sınıflandırılmasında 5 hareketin sınıflandırılmasında hata olmamıştır. Bu hareketlerin etiket değerleri 2, 7, 8, 12 ve 13'tür. Bu etiket değerlerine karşılık gelen hareketler sırası ile; beyaz tahtaya yazmak, sandalyede konuşmak, sandalyede dinlenmek, bilgisayarda çalışmak ve hareketsiz durmaktır.

Veri setinin, eğitim ve test seti olarak oluşturulabileceği bütün ayrımlar denenmiştir. Toplamda 4 farklı ayırım gerçekleştirilmiştir. Her bir test ve eğitim seti ayrımı için makine öğrenmesi tabanlı yöntemler ile sınıflandırma doğruluğu elde edilmiştir ve her bir ayırım için maksimum başarı değeri belirlenmiştir. Eğitim ve test seti ayrımlarından iki tanesi Alt Uzay Diskriminant (Subspace Discriminant) metodunda maksimum başarı vermiştir. Diğer maksimum başarı değerleri Kosinüs KNN (Cosine KNN) ve Doğrusal SVM (Linear SVM) metotlarında gelmiştir. Tablo 3.2'de her bir ayırım için yukarıda verilen metotlar ile elde edilen sınıflandırma başarıları ve kullanılan metotların parametre değerleri verilmiştir. Her bir metot için ortalama başarı değeri alınmıştır. Bunun sonucunda makine öğrenmesi yönteminde Kosinüs KNN metodu ile %72,60 ile en yüksek ortalama başarı elde edilmiştir.

Tablo 3.2. Makine öğrenmesi yönteminde veri seti ayrımları için maksimum sınıflandırma başarıları

Veri Seti	Alt Uzak Diskriminant	Doğrusal SVM	Kosinüs KNN
Eğitim Seti: 123 Test Seti: 4	%81,1	%64,3	%74,5
Eğitim Seti: 124 Test Seti: 3	%49,8	%54,8	%54,6
Eğitim Seti: 134 Test Seti: 2	%76,7	%79,4	%91,6
Eğitim Seti: 234 Test Seti: 1	%74,1	%63,1	%69,7
Ortalama Başarı	%70,43	%65,40	%72,60
Standart Sapma	14,05	10,24	15,24
Parametre Değerleri	Topluluk yöntemi: Alt Uzak Öğretici Tipi: Diskriminant Öğretici Sayısı: 30 Alt Uzak Boyutu: 72	Çekirdek ölçeği: Otomatik Kutu kısıtlama seviyesi: 1 Çok sınıflı yöntem: Bire Karşı Bir Verileri Standartlaştırma: Evet	Komşu Sayısı: 10 Uzaklık Ölçümü: Kosinüs Uzaklık Ağırlığı: Eşit Verileri Standartlaştırma: Evet

Tablo 3.2 incelendiğinde en düşük sınıflandırma doğruluğu eğitim seti:124 ve test seti:3 durumunda elde edilmiştir. Şekil 3.6’da 2 numaralı etiket bilgisine ait görüntüler verilmiştir. Bu kısımda denekler beyaz bir tahtaya yazı yazma işlemi gerçekleştirmektedir. Veri setinde bulunan 4 kişiden 3’ü (kişi 1, kişi 2, kişi 4) sağ elini kullanmaktadır. Ancak, 3 numaralı kişi sol elini kullanmaktadır. Bu veri seti ayrımında, eğitim sağ eklemeler ile yapılmıştır. Ama, test aşaması sol eklemeler ile gerçekleştiği için düşük bir başarı değeri gözlemlenmiştir. Standart sapmalara bakıldığında, en düşük değer Doğrusal SVM metodunda elde edilmiştir. Buradan anlaşıyor ki, bu sınıflandırıcıda veri seti ayrımları için elde edilen başarılarında daha az değişim söz konusudur.

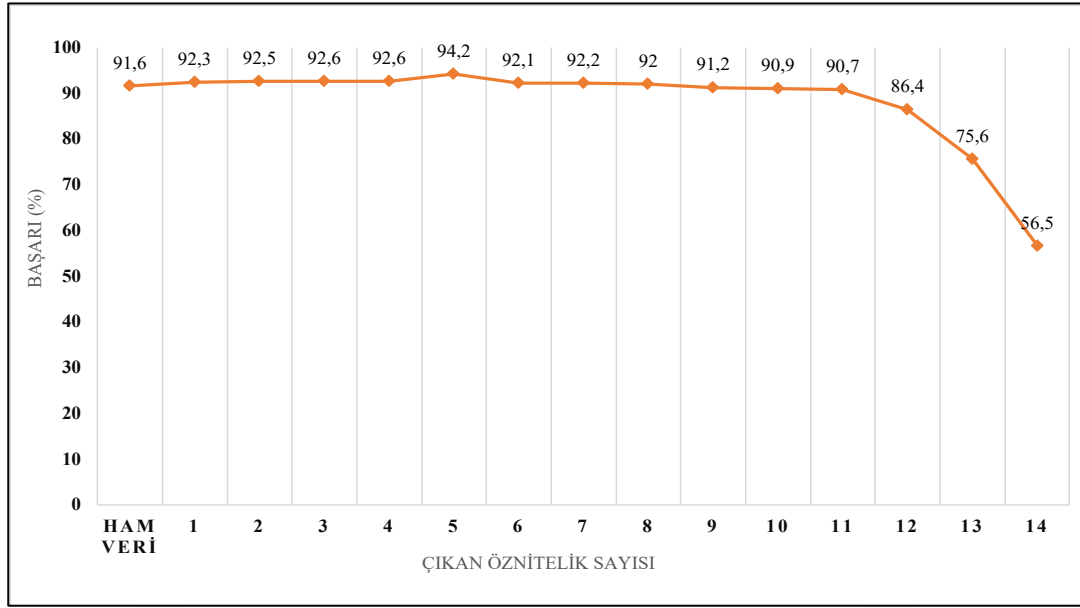


Şekil 3.6. Veri seti ile ilgili örnek görüntüler [13].

Ham veri ile elde edilen maksimum sınıflandırma başarısı, veri setinin elde edildiği çalışmanın başarı değerinin altında kalmıştır. Bunun sebebi ise ham veri seti üzerinde başarıyı olumsuz yönde etkileyecek verilerin veya birbirleri ile ilişkisiz verilerin bulunmasıdır. Bu durum, işlem yükünü de arttırmaktadır. Bu amaçla, ilişkisiz verilerden kurtulmak, doğruluk sonucunu arttırmak ve işlem yükünü azaltmak için öznitelik çıkarma uygulanmıştır.

3.2. Geriye Doğru Öznitelik Seçme (BFS)

Veri setinde Kinect sensör ile elde edilen 15 eklem bulunmaktadır. Makine öğrenmesinde elde edilen ortalama sınıflandırma başarısı %72,60'tı. Bu başarı değerini arttırmak için öznitelik seçme uygulanmıştır. Öznitelik seçme yönteminde BFS kullanılmıştır. Veri seti 4 farklı şekilde ayrılmıştı. Ham veri seti ayrımları ile elde edilen sınıflandırma başarıları incelenerek maksimum sınıflandırma doğruluğunun elde edildiği veri seti ayrımı eğitim seti:134 ve test seti:2 olarak belirlenmiştir. Bu veri seti ayrımı referans alınarak öznitelik seçme uygulanmıştır. Her bir öznitelik ayıklandığında elde edilen sınıflandırma başarısı Şekil 3.7'de verilmiştir.



Şekil 3.7. BFS sonucunda elde edilen sınıflandırma başarısı.

Eğitim seti:134 ve test seti:2 durumu için ham veri başarısı %91,6 iken, son kalan öznitelik (sol dirsek) başarısı %56,5'tir. Geriye doğru öznitelik seçme işleminde sınıflandırma başarısı, azalan öznitelik sayısı ile önce artmıştır ve sonra azalmaya başlamıştır. Sınıflandırma başarısı, 5 tane özellik çıkarıldığı zaman pik yapmıştır (%94,2). Grafiğin bu yapısı pik fenomeni (peaking phenomenon) olarak tanımlanmaktadır [134], [135]. Ayrıca, pik fenomeni için ayıklanan 5 tane öznitelik optimal özellik sayısı olarak tanımlanmaktadır [134]. Tablo 3.3'te her aşamada çıkaran öznitelik bilgisi verilmiştir.

Tablo 3.3 incelendiğinde çıkarılan ilk iki öznitelik sağ diz ve kafa olarak gözlemlenmektedir. Buradan anlaşılıyor ki, bu eklem bilgileri veri setinde bulunan hareketler için daha az belirleyici bir öznitelik olduğundan ötürü bu eklemlerin çıkarılması başarıyı arttırmıştır. Çıkarılan 3. eklem sol kalçadır. Şekil 3.6'da 2 numaralı etiket bilgisine ait görüntüler verilmişti. Görsellerden de görüleceği üzere deneklerin 3 tanesi sağ elini kullanırken, 1 tanesi sol elini kullanarak yazma işlemini gerçekleştirmektedir. Buradan da anlaşılacağı üzere bu veri setinin bu ayrımında sol eklemlerin veri seti üzerinde daha az bir etkisi bulunmaktadır. Bu sebeple, sol kalça başarıyı olumsuz yönde etkilemektedir. Bu durum, çıkarılan bir sonraki eklem sol el için de geçerlidir. 5 tane öznitelikten geriye kalan son ikisi sol el ve sağ eldir. Bu iki eklemin çıkarılması, ilk 3 öznitelik çıkarıldıktan sonra hareketin tanımlanmasına diğer eklemlere oranla daha az katkı sağladığı için başarıyı arttırmıştır. Öznitelik çıkarma işleminde geriye kalan son iki eklem sağ dirsek ve sol

dirsektir. Dirsekler, el ile yapılan birçok aktivitenin gerçekleştirilmesinde önemli bir rol oynamaktadır. Bir başka deęişle, dirsekler harekete yön vermektedir. Bu sebeple bu iki eklem en sona kalmıştır.

Tablo 3.3. Çıkarılan öznitelik bilgisi

Sıra	Öznitelik	
	Numara	İsim
1	11	Sağ Diz
2	1	Kafa
3	8	Sol Kalça
4	12	Sol El
5	13	Sağ El
6	6	Sağ Omuz
7	14	Sol Ayak
8	3	Gövde
9	15	Sağ Ayak
10	10	Sağ Kalça
11	4	Sol Omuz
12	2	Boyun
13	9	Sol Diz
14	7	Sağ Dirsek
Kalan	5	Sol Dirsek

Maksimum başarının elde edildiğı durum için (eğitim seti:134 ve test seti:2) makine öğrenmesi tabanlı 18 farklı metot kullanılarak yeni bir sınıflandırma başarısı elde edilmiştir. Bu başarı değerleri Tablo 3.4'te verilmiştir. Sonuçlar incelendiğinde bazı sonuçlar için sınıflandırma başarısının hiçbir şekilde deęişmediğı gözlemlenmiştir. Ancak genel anlamı ile bakıldığında BFS sonrasında mimariler sonucunda elde edilen sınıflandırma başarılarının arttığı gözlemlenmiştir.

Tablo 3.4. Eğitim seti:134 için BFS ile elde edilen sınıflandırma sonuçları

Model Numarası	Model İsmi	Ham Veri Test Başarısı (%)	Öznitelik Çıkarma Test Başarısı (%)
1.1	İnce Ağaç (Fine Tree)	70,9	77,4
1.2	Orta Ağaç (Medium Tree)	71,1	72,3
1.3	Kaba Ağaç (Coarse Tree)	45,9	47
1.4	Doğrusal Diskriminant (Linear Discriminant)	87,2	87,7
1.5	Doğrusal SVM (Linear SVM)	79,4	82,8
1.6	İkinci Dereceden SVM (Quadratic SVM)	73,3	81
1.7	Kübik SVM (Cubic SVM)	66,7	78,6
1.8	İnce Gauss SVM (Fine Gaussian SVM)	8,6	8,6
1.9	Orta Gauss SVM (Medium Gaussian SVM)	49,8	51,9
1.10	Kaba Gauss SVM (Coarse Gaussian SVM)	78,7	81,5
1.11	İnce KNN (Fine KNN)	91,1	91,5
1.12	Orta KNN (Medium KNN)	91,1	91,8
1.13	Kaba KNN (Coarse KNN)	89,3	89,3
1.14	Kosinüs KNN (Cosine KNN)	91,6	94,2
1.15	Ağırlıklandırılmış KNN (Weighted KNN)	91,1	91,9
1.16	Güçlendirilmiş Ağaçlar (Boosted Trees)	65,2	69,9
1.17	Torbalı Ağaçlar (Bagged Trees)	75,6	86,4
1.18	Alt Uzay Diskriminant (Subspace Discriminant)	76,7	90,1

Referans veri seti ayrımında maksimum başarı değeri 5 tane öznitelik çıkartıldığında elde edilmiştir. Çıkarılan 5 tane özellik sıra ile; sağ diz, kafa, sol kalça, sol el ve sağ eldir. Geriye kalan 3 tane veri seti ayrımında da aynı sıra ile aynı 5 tane öznitelik çıkartılmıştır. Tüm veri seti ayrımlarında öznitelik çıkartılması yapıldıktan sonra her birisi için makine öğrenmesi tabanlı metotlar kullanılarak sınıflandırma başarısı elde edilmiştir. Her veri seti ayrımı için maksimum başarı değeri ve maksimum başarı değerinin elde edildiği metot belirlenmiştir. Eğitim ve test seti ayrımlarından iki tanesi Alt Uzay Diskriminant (Subspace Discriminant) metodunda maksimum başarı vermiştir. Diğer maksimum başarı değerleri Kosinüs KNN (Cosine KNN) metodunda gelmiştir. Tablo 3.5'te her bir ayrım için ham veri setinin maksimum başarı değerlerinin elde edildiği metotlar ile elde edilen sınıflandırma başarıları verilmiştir.

Tablo 3.5. BFS sonucunda veri seti ayrımları için elde edilen başarı değerleri

Veri Seti	Alt Uzay Diskriminant	Doğrusal SVM	Kosinüs KNN
Eğitim Seti: 123 Test Seti: 4	%82,7	%65,7	%76,2
Eğitim Seti: 124 Test Seti: 3	%48,9	%51,2	%55,7
Eğitim Seti: 134 Test Seti: 2	%90,1	%82,8	%94,2
Eğitim Seti: 234 Test Seti: 1	%77,8	%63,7	%70,6
Ortalama Başarı	%74,88	%65,85	%74,18
Standart Sapma	18,04	12,99	15,91
Parametre Değerleri	Topluluk yöntemi: Alt Uzay Öğretici Tipi: Diskriminant Öğretici Sayısı: 30 Alt Uzay Boyutu: 51	Çekirdek (Kernel) ölçeği: Otomatik Kutu kısıtlama seviyesi: 1 Çok sınıflı yöntem: Bire Karşı Bir Verileri Standartlaştırma: Evet	Komşu Sayısı: 10 Uzaklık Ölçümü: Kosinüs Uzaklık Ağırlığı: Eşit Verileri Standartlaştırma: Evet

Her bir metot için ortalama başarı değeri alınmıştır. Kosinüs KNN metodunda elde edilen ortalama başarı değeri BFS ile %72,60'tan, %74,18'e yükselmiştir. Ancak, BFS yönteminde Alt Uzak Diskriminant yöntemi ile ortalama başarı daha yüksek elde edilmiştir. Standart sapmalar, önceki aşamaya göre artmıştır. En düşük değer ise önceki aşamada olduğu gibi Doğrusal SVM yönteminde elde edilmiştir.

3.3. LSTM

Sınıflandırıcıların performanslarını karşılaştırmak için yapılan bu çalışmada sonuçların karşılaştırılabilir olması için LSTM mimarisi uygulanırken eğitim ve test setleri makine öğrenmesi yönteminde olduğu gibi ayrılarak ortalama başarı değeri elde edilmiştir. LSTM algoritması uygulanırken bazı parametreler üstünde değişimler uygulanarak en yüksek başarı elde edilmeye çalışılmıştır. Bunun için önceki aşamalarda maksimum başarının elde edildiği eğitim seti:134 ve test seti:2 veri seti ayrımı referans olarak seçilmiştir.

Referans veri seti üzerinde ilk olarak Maksimum Eğitim Turu (max epochs) parametresi değiştirilmiştir ve sırası ile 25, 17 ve 32 olarak denenmiştir. En yüksek sonucu veren Maksimum Eğitim Turu parametresi 32 olarak bulunmuştur ve diğer işlemlerde bu parametre değeri korunmuştur.

Bir sonraki aşamada ise Mini Yığıt Boyutu (Mini Batch Size) parametresi sırası ile 27, 32 ve 20 olarak değiştirilmiştir ve en yüksek sonucu veren parametre değeri 27 olarak bulunmuştur.

Optimize edici seçiminde en sık kullanılan SGDM, RMSProp ve Adam optimizasyon yöntemleri denenmiştir. Algoritma sonucunda en yüksek sonucu veren optimizasyon yöntemi Adam olarak bulunmuştur.

LSTM blok sayısı artırılarak başarı irdelenmiştir. Blok sayısı 1 ve 2 olarak denenmiştir. Blok sayısı arttığında başarı artsaydı, sıra ile blok sayısı artırılarak maksimum başarıyı veren değer bulunması amaçlanmaktaydı. Ancak, blok sayısı 1 iken başarı daha yüksek gelmiştir.

Son olarak, Gizli Katman Sayısı (Number of Hidden Units) sırası ile 50, 80,100 ve 300 olarak denenmiştir. Elde edilen sonuçlara göre parametre değeri 100 olduğunda daha yüksek başarı değeri gözlemlenmiştir. Eğitim seti:134 ve test seti:2 durumu için elde edilen LSTM algoritması sonuçları Tablo 3.6’da verilmiştir.

Tablo 3.6. Eğitim seti:134 ve test seti:2 için LSTM sonuçları

Gizli Katman Sayısı (numHiddenUnits)	LSTM katman sayısı	Maksimum Eğitim Turu (MaxEpochs)	Mini Yığıt Boyutu (Mini Batch Size)	Optimize Edici	Başarı (%)
100	1	25	27	adam	57
100	1	17	27	adam	71
100	1	32	27	adam	77
100	1	32	32	adam	59
100	1	32	20	adam	71
100	1	32	27	sgdm	70
100	1	32	27	rmsprop	69
100	2	32	27	adam	61
50	1	32	27	adam	54
300	1	32	27	adam	73
80	1	32	27	adam	72
100	1	17	27	rmsprop	11
50	1	17	27	adam	74
300	1	17	27	adam	47
80	1	17	27	adam	66

LSTM yönteminde parametreler sıra ile değiştirilerek maksimum başarının elde edildiği parametre değerleri belirlenmiştir. Bu parametre değerleri aşağıda verilmiştir:

- Gizli katman sayısı (number of hidden units) = 100,
- LSTM katman sayısı = 1,
- Maksimum eğitim turu (Maximum epochs) = 32,
- Mini yığıt boyutu (Mini batch size) = 27,
- Optimize Edici (Optimizer) = Adam.

LSTM yöntemi sonucunda eğitim seti:134 ve test seti:2 durumu için elde edilen en yüksek sınıflandırma başarısı %77 olarak bulunmuştur ve bu başarı değeri için Karışıklık matrisi Şekil 3.8’de verilmiştir.

1	273		533								24		
2		5						87	1545				
3	26		200						492				60
4				1380	66								
5					1671	4							
6	27				8	1380							
7							1539						
8								1497					
9		5	217						288	1154			
10										1349			
11	4				158						704		97
12												1530	
13			482										
	1	2	3	4	5	6	7	8	9	10	11	12	13

Tahmin Edilen Sınıf

Doğru Sınıf

Şekil 3.8. LSTM için karışıklık matrisi.

Yukarıda belirlenen parametre değerleri referans alınarak veri setinin farklı ayrımları için LSTM algoritması 10 kez çalıştırılarak her bir ayırım için ortalama sınıflandırma başarıları elde edilmiştir. Tablo 3.7’de eğitim seti:123, test seti:4 için her bir aşamada elde edilen sınıflandırma başarıları verilmiştir.

Tablo 3.7. LSTM Yönteminde eğitim seti:123, test seti:4 için sınıflandırma başarıları

Deneme Numarası	Sınıflandırma Başarısı
1	%33
2	%55
3	%67
4	%52
5	%63
6	%59
7	%49
8	%49
9	%56
10	%47

Tablo 3.7’de eğitim seti:123, test seti:4 için her bir aşamada elde edilen sınıflandırma başarıları verilmiştir. Buna göre, eğitim seti:123, test seti:4 durumunda LSTM mimarisi için elde edilen ortalama başarı %53’tür. Bir sonraki aşamada eğitim seti:124 ve test seti:3 veri seti ayrımı için LSTM algoritması 10 kez çalıştırılmıştır ve elde edilen başarı değerleri Tablo 3.8’de verilmiştir.

Tablo 3.8. LSTM yönteminde eğitim seti:124, test seti:3 için sınıflandırma başarıları

Deneme Numarası	Sınıflandırma Başarısı
1	%42
2	%36
3	%49
4	%43
5	%43
6	%38
7	%52
8	%41
9	%52
10	%53

Eğitim seti:124, test seti:3 veri seti ayrımı için LSTM algoritması 10 kez çalıştırılmıştır ve elde edilen ortalama başarı %45 olarak bulunmuştur. Bir sonraki aşamada eğitim seti:134 ve test seti:2 veri seti ayrımı için LSTM algoritması 10 kez çalıştırılmıştır ve elde edilen başarı değerleri Tablo 3.9’da verilmiştir.

Tablo 3.9. LSTM yönteminde eğitim seti:134, test seti:2 için sınıflandırma başarıları

Deneme Numarası	Sınıflandırma Başarısı
1	%77
2	%70
3	%57
4	%77
5	%57
6	%71
7	%67
8	%75
9	%69
10	%76

Eğitim seti:134, test seti:2 veri seti ayrımı için LSTM algoritmasında elde edilen ortalama başarı %70 olarak bulunmuştur. Veri setinin son ayrımı için LSTM algoritması 10 kez çalıştırılmıştır ve elde edilen başarı değerleri Tablo 3.10’da verilmiştir.

Tablo 3.10. LSTM yönteminde eğitim seti:234, test seti:1 için sınıflandırma başarıları

Deneme Numarası	Sınıflandırma Başarısı
1	%37
2	%47
3	%50
4	%51
5	%45
6	%46
7	%47
8	%49
9	%40
10	%50

Eğitim seti:234, test seti:1 veri seti ayrımı için LSTM algoritmasında elde edilen ortalama başarı %46 olarak bulunmuştur. Tablo 3.11’de tüm veri seti ayrımları için elde edilen ortalama başarı değeri verilmiştir. Her bir veri seti ayrımında elde edilen başarının ortalaması alınarak LSTM algoritması sonucunda ortalama sınıflandırma başarıları %54 olarak bulunmuştur.

Tablo 3.11. LSTM için ortalama sınıflandırma sonuçları

Data Set	LSTM’de Sınıflandırma Başarısı
Eğitim Seti: 123 Test Seti: 4	%53
Eğitim Seti: 124 Test Seti: 3	%45
Eğitim Seti: 134 Test Seti: 2	%70
Eğitim Seti: 234 Test Seti: 1	%46
Ortalama Başarı	%54
Standart Sapma	11,56

LSTM ile elde edilen sınıflandırma başarısı ham veri seti başarısının altında elde edilmiştir. Bunun sebebi ise LSTM'nin sonunda kullanılan tam bağlantılı katmandır (fully connected layer). Çıkarılan öznitelikler tam bağlantılı katman için yeterli olmamıştır. Bu sebeple, LSTM ile öznitelik çıkarma ve sınıflandırma için makine öğrenmesi tabanlı yöntemler kullanılarak bir sonraki çalışma yapılmıştır.

3.4. LSTM ile Öznitelik Çıkarma

Çalışmanın bu aşamasında sınıflandırma başarısını arttırmak için LSTM mimarisi kullanılarak öznitelik çıkarma işlemi gerçekleştirilmiştir. Bu işlemin sonunda elde edilen öznitelik matrisi kullanılarak eğitim ve test setleri oluşturulmuştur. Bu ayırım önceki aşamalarda olduğu gibi dört farklı şekilde yapılmıştır. Elde edilen veri setleri makine öğrenmesi yöntemleri kullanılarak test edilmiştir. Tablo 3.12'de tüm veri seti ayrımları için elde edilen sınıflandırma sonuçları verilmiştir.

Tablo 3.12. LSTM ile öznitelik çıkarma yönteminde veri seti ayrımları için sınıflandırma başarıları (%)

Model Sırası	Model İsim	Eğitim Seti:123 Test Seti: 4	Eğitim Seti:124 Test Seti: 3	Eğitim Seti:134 Test Seti: 2	Eğitim Seti:234 Test Seti: 1
1.1	İnce Ağaç (Fine Tree)	92,6	81,8	95,7	74,3
1.2	Orta Ağaç (Medium Tree)	87,6	75	97,6	60,7
1.3	Kaba Ağaç (Coarse Tree)	36,2	37,1	46,2	27,9
1.4	Doğrusal Diskriminant (Linear Discriminant)	97,9	87,1	99,3	99,2
1.5	İkinci Dereceden Diskriminant (Quadratic Discriminant)	82	73,3	93	82,9

1.6	Doğrusal SVM (Linear SVM)	92,7	94,2	99,7	90,4
1.7	İkinci Dereceden SVM (Quadratic SVM)	93,1	86,4	99,5	83,5
1.8	Kübik SVM (Cubic SVM)	92,9	85,1	99	82,5
1.9	İnce Gauss SVM (Fine Gaussian SVM)	15,3	81,2	23,6	71,3
1.10	Orta Gauss SVM (Medium Gaussian SVM)	92,1	98,3	98,2	82,3
1.11	Kaba Gauss SVM (Coarse Gaussian SVM)	92,3	95,7	99,6	86,5
1.12	İnce KNN (Fine KNN)	98,4	90	99	99,6
1.13	Orta KNN (Medium KNN)	98,3	89,5	98,9	99,6
1.14	Kaba KNN (Coarse KNN)	98,5	88,9	98,9	99,5
1.15	Kosinüs KNN (Cosine KNN)	98,4	89,4	98,9	99,7
1.16	Kübik KNN (Cubic KNN)	98,3	89,4	98,9	99,5
1.17	Ağırlıklandırılmış KNN (Weighted KNN)	98,3	89,5	98,9	99,6
1.18	Güçlendirilmiş Ağaçlar (Boosted Trees)	96,6	97,4	97,6	92,3
1.19	Torbalı Ağaçlar (Bagged Trees)	98,3	97,7	98,5	92,3
1.20	Alt Uzay Diskriminant (Subspace Discriminant)	97,8	88	99,2	99,7

1.21	Alt Uzay KNN (Subspace KNN)	98,2	90,3	98,8	99,8
1.22	Yükseltilmiş Ağaçlar (RUSBoosted Trees)	97,5	93,7	97,5	95,2
1.23	Dar Sinir Ağı (Narrow Neural Network)	99	98,3	98,3	81
1.24	Orta Sinir Ağı (Medium Neural Network)	97,7	98,8	97,8	90,7
1.25	Geniş Sinir Ağı (Wide Neural Network)	97,4	98,2	90,5	97,1
1.26	İki Katmanlı Sinir Ağı (Bilayered Neural Network)	97,5	88,9	88,8	88
1.27	Üç Katmanlı Sinir Ağı (Trilayered Neural Network)	95,2	95,4	88	98,3
1.28	SVM Çekirdeği (SVM Kernel)	97,8	98,6	99,3	99,7
1.29	Lojistik Regresyon Çekirdeği (Logistic Regression Kernel)	97,9	98,8	98,7	99,8

Tablo 3.12’de tüm veri seti ayrımları için test başarıları verilmiştir. Eğitim seti:123 ve test seti:4 için test başarısında önceki aşamalara göre iyileşme gözlemlenmiştir. Maksimum başarı değeri Dar Sinir Ağı (Narrow Neural Network) modelinde %99 olarak elde edilmiştir. Şekil 3.9’da bu başarı değeri için elde edilen karışıklık matrisi verilmiştir. Eğitim seti:124 ve test seti:3 için maksimum başarı değeri SVM Çekirdeği (SVM Kernel) modelinde %99,3 olarak elde edilmiştir. Şekil 3.10’da bu başarı değeri için elde edilen karışıklık matrisi verilmiştir. Eğitim seti:134 ve test seti:2 için maksimum başarı değeri önceki veri seti ayrımında olduğu gibi SVM Çekirdeği (SVM Kernel) modelinde elde edilmiştir. Elde edilen başarı değeri %99,3’tür ve bu başarısının karışıklık matrisi Şekil 3.11’de verilmiştir. Son olarak, eğitim seti:234 ve test seti:1 için maksimum başarı değeri Lojistik Regresyon Çekirdeği (Logistic Regression Kernel) modelinde %99,8 olarak elde edilmiştir ve Şekil 3.12’de karışıklık matrisi verilmiştir.

1	1308											
2		1792										
3			1526							3		
4				1844	21							
5					1224	356						
6					56	1044						
7							1812					
8								1853				
9									1910			
10		1								1834		
11											1012	
12												1662
13												1102

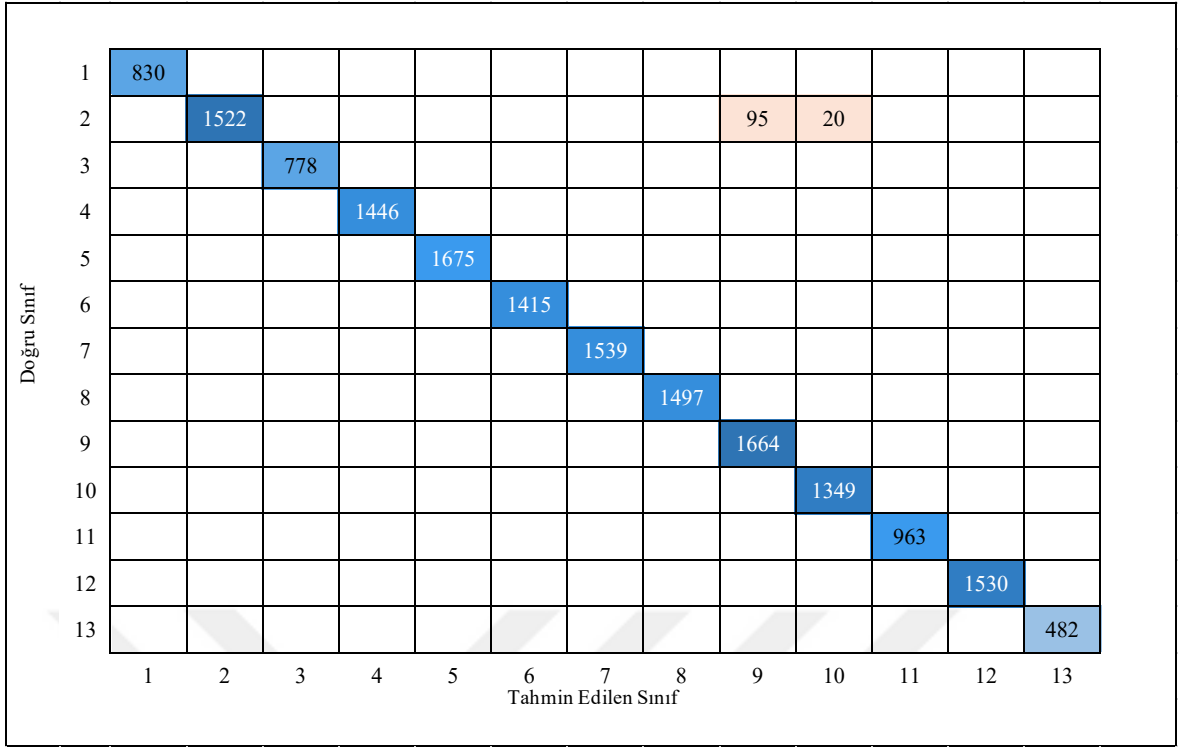
Tahmin Edilen Sınıf

Şekil 3.9. LSTM ile öznitelik çıkarma yönteminde eğitim seti:123 ve test seti:4 için karışıklık matrisi.

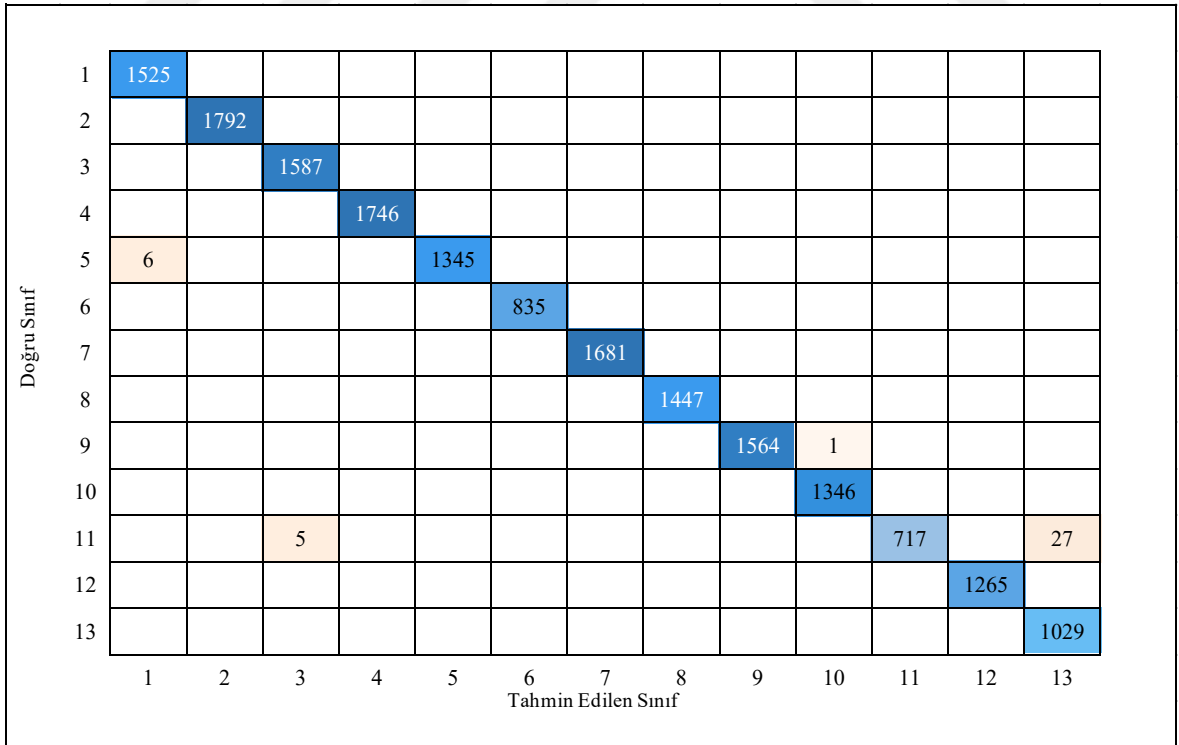
1	1257							15		16		
2		1597										
3		91	1219									
4				1494		7			2			
5					1783							
6						822						
7							1712					
8								1379				
9									1754			
10									3	1464		
11										1	620	
12		67										1155
13										11		1108

Tahmin Edilen Sınıf

Şekil 3.10. LSTM ile öznitelik çıkarma yönteminde eğitim seti:124 ve test seti:3 için karışıklık matrisi.



Şekil 3.11. LSTM ile öznitelik çıkarma yönteminde eğitim seti:134 ve test seti:2 için karışıklık matrisi.



Şekil 3.12. LSTM ile öznitelik çıkarma yönteminde eğitim seti:234 ve test seti:1 için karışıklık matrisi.

Veri seti ayrımları için elde edilen karışıklık matrisleri yukarıda verilmiştir. Şekiller sıra ile incelendiğinde, ilk olarak eğitim seti:123 ve test seti:4 veri seti ayrımı için 3, 4, 5, 6 ve 10 numaralı etikete sahip hareketlerde yanlış sınıflandırılmalar olmuştur. Eğitim seti:124 ve test seti:3 veri seti ayrımı için 7 tane hareketin (1, 3, 4, 10, 11, 12 ve 13 numaralı etiketler) sınıflandırılmasında hata olmuştur. Eğitim seti:134 ve test seti:2 veri seti ayrımında sadece bir hareketin sınıflandırılmasında (2 numaralı etiket – beyaz tahtaya yazmak) hata olmuştur. Son veri seti ayrımında ise 4, 9 ve 11 numaralı etikete sahip hareketlerde yanlış sınıflandırma olmuştur. Genel olarak incelendiğinde sandalyede konuşmak (7 numaralı etiket) ve sandalyede dinlenmek (8 numaralı etiket) aktivitelerinde yanlış sınıflandırma olmamıştır. 13 aktiviteden sadece bu iki aktivite oturarak gerçekleştirilmiştir. Bu sebeple, diğer hareketlerden daha belirgin bir şekilde ayrılarak sınıflandırma başarısının yüksek çıktığı düşünülmektedir.

Veri setinin, eğitim ve test seti olarak oluşturulabileceği bütün ayrımlar denenmiştir. Her bir test ve eğitim seti ayrımı için LSTM ile öznitelik çıkarıldıktan sonra makine öğrenmesi yöntemi ile sınıflandırma yapılmıştır. En yüksek başarı değerlerinin elde edildiği metotlar; Orta Sinir Ağı (Medium Neural Network), Lojistik Regresyon Çekirdeği (Logistic Regression Kernel), SVM Çekirdeği (SVM Kernel) olarak belirlenmiştir. Tablo 3.13'te her bir ayrım için yukarıda verilen metotlar ile elde edilen sınıflandırma başarıları ve kullanılan metotların parametre değerleri verilmiştir. Her bir metot için ortalama başarı değeri alınmıştır. Bunun sonucunda LSTM ile öznitelik çıkarılması yönteminde SVM Çekirdeği metodu ile %98,685 ile en yüksek ortalama başarı elde edilmiştir.

Tablo 3.13'te verilen ortalama başarı ve standart sapma sonuçları incelendiğinde, önceki yöntemlerde elde edilen sonuçlara oranla büyük bir iyileşme olduğu gözükmemektedir. Değerlerden anlaşılıyor ki, veri seti ayrımlarında elde edilen sınıflandırma doğruluklarında fazla bir değişim söz konusu değildir. Bu konuda en başarılı sınıflandırıcı Lojistik Regresyon Çekirdeğidir. LSTM ile öznitelik çıkarıldığında, ham veri setine göre daha fazla öznitelik elde edilmiştir. LSTM öznitelikler arasında ilişki kuran bir yöntemdir. O yüzden zaman serisi problemlerinde sıklıkla kullanılmaktadır [136], [137]. Burada da yüksek başarı göstermesinin sebebi olarak eklemlerden elde edilen sensör verileri arasında iyi bir ilişki kurması olarak düşünülmektedir. Bunun yanı sıra öznitelik sayısının artması, veri setinin

sınıflandırma başarısını arttırmıştır. Ayrıca, SVM metodu öznitelik sayısı fazla olduğunda dayanıklı (robust) bir yöntem olup yüksek doğruluk ile çalışmaktadır [138].

Tablo 3.13. LSTM ile öznitelik çıkarma için ortalama başarı değerleri

Veri Seti	Orta Sinir Ağı	Lojistik Regresyon Çekirdeği	SVM Çekirdeği
Eğitim Seti: 123 Test Seti: 4	%97,7	%97,9	%97,8
Eğitim Seti: 124 Test Seti: 3	%98,8	%98,8	%98,6
Eğitim Seti: 134 Test Seti: 2	%97,8	%98,7	%99,3
Eğitim Seti: 234 Test Seti: 1	%90,7	%99,8	%99,7
Ortalama Başarı	%96,25	%98,80	%98,85
Standart Sapma	3,23	0,79	0,83
Parametre Değerleri	Tam bağlı katman sayısı: 1 İlk katman boyutu: 25 Aktivasyon: ReLU Yineleme sınırı: 1000 Düzenleştirme gücü (Lambda): 0 Verileri standartlaştırma: Evet	Genişletme boyutları sayısı: Otomatik Düzenleştirme gücü (Lambda): Otomatik Çekirdek ölçeği: Otomatik Çoklu sınıf yöntemi: Bire-Bir-Bir Yineleme sınırı: 1000	Genişletme boyutları sayısı: Otomatik Düzenleştirme gücü (Lambda): Otomatik Çekirdek ölçeği: Otomatik Çoklu sınıf yöntemi: Bire-Bir-Bir Yineleme sınırı: 1000

3.5. Sıfırdan Öğrenme (Learn from Scratch)

CNN mimarisi uygulanırken önceki aşamalardan farklı olarak eklem konum bilgilerinden oluşan veri seti yerine RGB görüntü verileri üzerinde çalışmalar gerçekleştirilmiştir. Veri setinde RGB görüntü ve derinlik görüntüleri bir arada bulunmaktadır. Ancak, çalışmada derinlik görüntüsü kullanılmak yerine RGB görüntüler kullanılmıştır. Bunun sebebi ise sınıflandırma yapılacağı zaman RGB görüntülerin, derinlik görüntülerine göre daha iyi sonuç vermesidir, çünkü, aydınlatma koşullarındaki değişiklik önemli bir rol oynamaktadır. Tablo 3.14’te her bir etiket değerine karşılık gelen RGB görüntü sayısı verilmiştir.

Tablo 3.14. RGB görüntü sayıları

Etiket Bilgisi	RGB Görüntü Sayısı
Telefonda Konuşmak	4951
Beyaz Tahtaya Yazmak	6818
Su İçmek	5204
Su ile Ağızı Çalkalamak	6560
Diş Fırçalamak	6389
Lens Takmak	4172
Sandalyede Konuşmak	6744
Sandalyede Dinlenmek	6176
Pişirmek (Doğramak)	6893
Pişirmek (Karıştırmak)	5997
İlaç Kutusu Açmak	3345
Bilgisayarda Çalışmak	5679
Hareketsiz Durmak	3732

Bu çalışma kapsamında, görüntü işlemede en çok kullanılan CNN mimarisinde, 72.660 RGB görüntü kullanılarak sınıflandırma işlemi gerçekleştirilmiştir. Eğitim ve test verileri önceki aşamalarda olduğu gibi ayrılarak oluşturulmuştur. Veri seti, sıfırdan öğrenme ile 7 katmandan oluşan CNN modeli ile sınıflandırılmıştır. Tablo 3.15'te her bir eğitim ve test veri seti ayrımı için elde edilen sınıflandırma başarısı verilmiştir. Elde edilen sınıflandırma başarısı için kullanılan katmanlar aşağıda listelenmiştir:

- Giriş görüntü katmanı (image input layer)
- Konvolüsyon katmanı (convolution layer)
- Relu katmanı
- Maksimum havuzlama katmanı (max pooling layer)
- Tam bağlantılı katman (fully connected layer)
- Softmax katmanı
- Sınıflandırma katmanı (classification layer)

Tablo 3.15. Sıfırdan öğrenme ile sınıflandırma başarıları

Veri Seti	Sınıflandırma Başarısı
Eğitim Seti: 123 Test Seti: 4	%23,51
Eğitim Seti: 124 Test Seti: 3	%19,71
Eğitim Seti: 134 Test Seti: 2	%28,12
Eğitim Seti: 234 Test Seti: 1	%25,61
Ortalama Başarı	%24,24
Standart Sapma	3,56

Sıfırdan öğrenme yönteminde elde edilen ortalama başarı değeri %24,24'tür. Bu yöntemde elde edilen başarı değeri diğer yöntemlere kıyasla çok düşük gelmiştir. Bunun başlıca iki sebebi olduğu düşünülmektedir. İlk olarak, sıfırdan öğrenme yöntemi ağı derinlemesine eğitmek için çok büyük veri setlerine ihtiyaç duymaktadır. Ancak, kullanılan

veri setinde çok fazla veri olmaması ile ağ yeterli düzeyde eğitilememiştir. Bunun sonucunda, yeteri kadar ayırt edici öznelik ortaya çıkamaması ile düşük bir başarı değeri elde edilmiştir. Başarıyı düşüren bir diğer sebep ise, katmanların sayısını ve parametre değerlerini ayarlamanın zorluğudur. Sıfırdan öğrenen bir derin ağ yapısı kurmak için veri setine uygun katmanların belirlenmesi gerekmektedir. Ancak, bu durum hem çok maliyetli hem de çok zaman alıcı bir iş olmaktadır. Ayrıca, bu katmanlar ince ayar yapılması gereken pek çok parametre barındırmaktadır. Bu sebeple oluşturulan ağın tüm parametrelerini en iyilemek çok zorlu bir iş olmaktadır. Bu durumlar sistemin başarısını olumsuz yönde etkilemektedir. Bu durum bizi öğrenme aktarımına yönlendirmektedir.

3.6. Öğrenme Aktarımı (Transfer Learning)

Bir Evrişimli Sinir Ağı, DL alanından güçlü bir makine öğrenme tekniğidir. CNN'ler, çeşitli görüntülerden oluşan geniş veri setleri kullanılarak eğitilmektedirler. Bu geniş veri setlerinden CNN'ler, çeşitli görüntüler için daha iyi performans gösteren özellik temsillerini öğrenebilmektedirler. Eğitim için fazla bir zaman ve çaba göstermeden CNN'ler için verimli bir sonuç elde etme yolu ise özellik çıkarıcı olarak önceden eğitilmiş bir CNN mimarisi kullanmaktır. Bunun için transfer öğrenmesi yöntemi ile literatürde yayın olarak kullanılan farklı mimariler kullanılarak öznelik çıkarma uygulanmıştır. Kullanılan mimariler birbirlerinden farklı özelliklere sahiptirler. Bu mimariler ile ilgili detaylı bilgi Bölüm 2'de verilmiştir.

Veri setinin işlenmesi çok zaman aldığından dolayı veri setinin tüm ayrımları için 5 farklı mimari kullanılarak sınıflandırma başarısının elde edilmesi çok zaman almaktadır. Bu sebeple ilk olarak veri seti bütün halde rastgele olarak %70 eğitim verisi ve %30 test verisi olarak ayrılmıştır. Bu ayırım için öğrenme aktarımı yönteminde farklı mimariler için elde edilen doğrulama başarısı sonuçları Tablo 3.16'da verilmiştir.

Tablo 3.16'ya bakıldığında toplamda 5 mimari için elde edilen sınıflandırma başarıları görülmektedir. Sonuçlara bakıldığında 4 mimari için %100 başarı, diğer mimari için ise %100'e çok yakın bir başarı değeri görülmektedir.

Elde edilen sonuçlardan ilk üçü seçilmiştir ve önceki aşamalarda kullanılan 4 farklı veri seti ayırımı için bu mimariler kullanılarak öznelik çıkarma uygulanmıştır. Öznelik

çıkardıktan sonra makine öğrenmesi yönteminde bulunan 29 farklı sınıflandırıcı kullanılarak sınıflandırma başarılarının elde edilmesi amaçlanmıştır. Ancak, bazı sınıflandırıcılar için sonuç elde edilememiştir ve toplamda 22 farklı sınıflandırıcı için başarı değerleri elde edilmiştir.

Tablo 3.16. Ağ mimarilerine göre sınıflandırma başarıları

Mimari Yapı	Doğrulama Başarısı (%)
ResNet50	100
DenseNet-201	100
Xception	100
Inception v3	100
AlexNet	99,98

Tüm veri seti ayrımlarında ilk olarak Resnet50 mimarisi kullanılarak öznitelikler çıkarılmıştır. Daha sonra 22 farklı sınıflandırıcı ile veri seti ayrımlarının sınıflandırma başarıları elde edilmiştir. Elde edilen sonuçlar Tablo 3.17’de verilmiştir.

Tablo 3.17. Öğrenme aktarımı yönteminde veri seti ayrımları için Resnet50 ile elde edilen sınıflandırma başarıları (%)

Model Sırası	Model İsim	Eğitim Seti:123 Test Seti: 4	Eğitim Seti:124 Test Seti: 3	Eğitim Seti:134 Test Seti: 2	Eğitim Seti:234 Test Seti: 1
1.1	İnce Ağaç (Fine Tree)	34,3	28,2	32,6	25,7
1.2	Orta Ağaç (Medium Tree)	34,7	35	30,3	32,9
1.3	Kaba Ağaç (Coarse Tree)	26,8	37,7	25,3	35,7

1.4	Doğrusal SVM (Linear SVM)	48,9	50,8	59,5	53,1
1.5	İkinci Dereceden SVM (Quadratic SVM)	50,7	43,5	55,3	41,8
1.6	Kübik SVM (Cubic SVM)	45,4	36,5	53,3	43,1
1.7	Orta Gauss SVM (Medium Gaussian SVM)	30,5	41,5	27,9	30,8
1.8	Kaba Gauss SVM (Coarse Gaussian SVM)	49,5	52,4	56,2	46,9
1.9	İnce KNN (Fine KNN)	38,3	41,1	37	45,7
1.10	Orta KNN (Medium KNN)	39,7	42,8	37,5	46,8
1.11	Kaba KNN (Coarse KNN)	39,6	40,3	37,1	49,1
1.12	Kosinüs KNN (Cosine KNN)	52,4	44,4	39,2	49,6
1.13	Ağırlıklandırılmış KNN (Weighted KNN)	39,5	42,5	37,3	46,5
1.14	Güçlendirilmiş Ağaçlar (Boosted Trees)	41,4	36	29,9	39,7
1.15	Torbalı Ağaçlar (Bagged Trees)	48,5	49,5	39,6	43,4
1.16	Alt Uzay Diskriminant (Subspace Discriminant)	63,6	54,9	59,6	65,9
1.17	Yükseltilmiş Ağaçlar (RUSBoosted Trees)	40,3	39,6	26,5	33,4
1.18	Dar Sinir Ağı (Narrow Neural Network)	42,4	42	32,4	56,9

1.19	Orta Sinir Ağı (Medium Neural Network)	46,7	45,2	31,7	64,2
1.20	Geniş Sinir Ağı (Wide Neural Network)	46,5	43,2	41,3	58,6
1.21	İki Katmanlı Sinir Ağı (Bilayered Neural Network)	53,1	36,5	32,9	40,2
1.22	Üç Katmanlı Sinir Ağı (Trilayered Neural Network)	52,3	43,7	29,1	53,7

Tablo 3.17’de tüm veri seti ayrımları için elde edilen test başarıları verilmiştir. Eğitim seti:123 ve test seti:4 ayrımı için Resnet50 ile öznetelik çıkarıldıktan sonra maksimum sınıflandırma başarısı Alt Uzay Diskriminant (Subspace Discriminant) sınıflandırıcısında %63,6 olarak elde edilmiştir. Eğitim seti:124 ve test seti:3 ayrımı için maksimum sınıflandırma başarısı Alt Uzay Diskriminant (Subspace Discriminant) sınıflandırıcısında %54,9 olarak elde edilmiştir. Veri setinin üçüncü ayrımında, eğitim seti:134 ve test seti:2 için sınıflandırma başarısı incelenmiştir. Bu ayrımında, maksimum sınıflandırma başarısı Alt Uzay Diskriminant (Subspace Discriminant) sınıflandırıcısında %59,6 olarak elde edilmiştir. Son olarak, eğitim seti:234 ve test seti:1 ayrımı için Resnet50 ile öznetelik çıkarıldıktan sonra maksimum sınıflandırma başarısı Alt Uzay Diskriminant (Subspace Discriminant) sınıflandırıcısında %65,9 olarak elde edilmiştir.

Bir sonraki aşamada, DenseNet-201 ağ mimarisi kullanılarak öznetelikler çıkarılmıştır. Daha sonra 22 farklı sınıflandırıcı ile veri seti ayrımlarının sınıflandırma başarıları elde edilmiştir. Tüm veri seti ayrımları için elde edilen sınıflandırma başarıları Tablo 3.18’de verilmiştir.

Tablo 3.18. Öğrenme aktarımı yönteminde veri seti ayrımları için DenseNet-201 ile elde edilen sınıflandırma başarıları (%)

Model Sırası	Model İsim	Eğitim Seti:123 Test Seti: 4	Eğitim Seti:124 Test Seti: 3	Eğitim Seti:134 Test Seti: 2	Eğitim Seti:234 Test Seti: 1
1.1	İnce Ağaç (Fine Tree)	40,3	26,7	36,5	36,5
1.2	Orta Ağaç (Medium Tree)	32,8	22,5	23,6	23,6
1.3	Kaba Ağaç (Coarse Tree)	26,3	19,4	18,5	18,5
1.4	Doğrusal SVM (Linear SVM)	47,1	41,8	57,6	57,6
1.5	İkinci Dereceden SVM (Quadratic SVM)	45,4	36,5	50,1	50,1
1.6	Kübik SVM (Cubic SVM)	46,9	45,6	60,5	60,5
1.7	Orta Gauss SVM (Medium Gaussian SVM)	29,6	9	18,5	18,5
1.8	Kaba Gauss SVM (Coarse Gaussian SVM)	46	30,9	54,7	54,7
1.9	İnce KNN (Fine KNN)	33,2	44,2	54,7	54,7
1.10	Orta KNN (Medium KNN)	37,2	44,8	55,1	55,1
1.11	Kaba KNN (Coarse KNN)	41	46,2	59,4	59,4
1.12	Kosinüs KNN (Cosine KNN)	37,6	48,3	53,8	53,8

1.13	Ağırlıklandırılmış KNN (Weighted KNN)	36,6	44,7	55,2	55,2
1.14	Güçlendirilmiş Ağaçlar (Boosted Trees)	38,5	44,2	39,1	39,1
1.15	Torbalı Ağaçlar (Bagged Trees)	46,9	52,1	58,5	58,5
1.16	Alt Uzay Diskriminant (Subspace Discriminant)	41,2	44,4	54,6	54,6
1.17	Yükseltilmiş Ağaçlar (RUSBoosted Trees)	38,9	38	24	24
1.18	Dar Sinir Ağı (Narrow Neural Network)	30,4	46,5	73,1	73,1
1.19	Orta Sinir Ağı (Medium Neural Network)	41,9	56,6	66,7	66,7
1.20	Geniş Sinir Ağı (Wide Neural Network)	29,5	55,2	64,8	64,8
1.21	İki Katmanlı Sinir Ağı (Bilayered Neural Network)	44,6	45,1	57,3	57,3
1.22	Üç Katmanlı Sinir Ağı (Trilayered Neural Network)	29,3	51,5	55,9	55,9

Tablo 3.18’de verilen sonuçlar incelendiğinde eğitim seti:123 ve test seti:4 ayrımı için DenseNet-201 ile öznetelik çıkarıldıktan sonra maksimum sınıflandırma başarısı Doğrusal SVM (Linear SVM) sınıflandırıcısında %47,1 olarak elde edilmiştir. Eğitim seti:124 ve test seti:3 ayrımı için maksimum sınıflandırma başarısı Orta Sinir Ağı (Medium Neural Network) sınıflandırıcısında %56,6 olarak elde edilmiştir. Veri setinin üçüncü ayrımında, eğitim seti:134 ve test seti:2 için sınıflandırma başarısı incelenmiştir. Bu ayrımında, maksimum sınıflandırma başarısı İkinci Dereceden SVM (Quadratic SVM) sınıflandırıcısında %65,3 olarak elde edilmiştir. Son olarak, eğitim seti:234 ve test seti:1 ayrımı için DenseNet-201 ile öznetelik çıkarıldıktan sonra maksimum sınıflandırma başarısı Dar Sinir Ağı (Narrow Neural Network) sınıflandırıcısında %73,1 olarak elde edilmiştir.

Son aşamada, Xception ağ mimarisi kullanılarak öz nitelikler çıkarılmıştır. Daha sonra 22 farklı sınıflandırıcı ile veri seti ayrımlarının sınıflandırma başarıları elde edilmiştir. Tüm veri seti ayrımları için elde edilen sınıflandırma başarıları Tablo 3.19’da verilmiştir.

Tablo 3.19. Öğrenme aktarımı yönteminde veri seti ayrımları için Xception ile elde edilen sınıflandırma başarıları (%)

Model Sırası	Model İsim	Eğitim Seti:123 Test Seti: 4	Eğitim Seti:124 Test Seti: 3	Eğitim Seti:134 Test Seti: 2	Eğitim Seti:234 Test Seti: 1
1.1	İnce Ağaç (Fine Tree)	40,8	33	33,8	38,3
1.2	Orta Ağaç (Medium Tree)	47,5	37,9	41,4	39,2
1.3	Kaba Ağaç (Coarse Tree)	27,4	32,3	24,8	27,8
1.4	Doğrusal SVM (Linear SVM)	64,4	61	53,7	64,1
1.5	İkinci Dereceden SVM (Quadratic SVM)	64,7	58,2	50,6	63
1.6	Kübik SVM (Cubic SVM)	63,6	56,6	48	62,2
1.7	Orta Gauss SVM (Medium Gaussian SVM)	44,5	35,8	33,1	46,1
1.8	Kaba Gauss SVM (Coarse Gaussian SVM)	62,9	69	52,4	63,6
1.9	İnce KNN (Fine KNN)	64,6	58,7	55,1	55,8
1.10	Orta KNN (Medium KNN)	62,9	59,5	55,9	55,9
1.11	Kaba KNN (Coarse KNN)	63	61,7	53,5	59,4

1.12	Kosinüs KNN (Cosine KNN)	60,5	54,7	61,3	49,1
1.13	Ağırlıklandırılmış KNN (Weighted KNN)	63,3	59,1	55,4	43,7
1.14	Güçlendirilmiş Ağaçlar (Boosted Trees)	54,3	47,8	46,2	52,2
1.15	Torbalı Ağaçlar (Bagged Trees)	57,4	46	47	60,7
1.16	Alt Uzay Diskriminant (Subspace Discriminant)	64,3	63,9	52,8	65,9
1.17	Yükseltilmiş Ağaçlar (RUSBoosted Trees)	48,7	47,7	38	42,6
1.18	Dar Sinir Ağı (Narrow Neural Network)	58,1	56,7	43,2	72,8
1.19	Orta Sinir Ağı (Medium Neural Network)	68,3	61	45,1	64,4
1.20	Geniş Sinir Ağı (Wide Neural Network)	71,9	58,9	49,4	66,1
1.21	İki Katmanlı Sinir Ağı (Bilayered Neural Network)	47,3	66,2	48,5	50,3
1.22	Üç Katmanlı Sinir Ağı (Trilayered Neural Network)	47,7	47,9	43	65,7

Tablo 3.19’de verilen sonuçlar incelendiğinde eğitim seti:123 ve test seti:4 ayrımı için Xception ile öznetelik çıkarıldıktan sonra maksimum sınıflandırma başarısı Geniş Sinir Ağı (Wide Neural Network) sınıflandırıcısında %71,9 olarak elde edilmiştir. Eğitim seti:124 ve test seti:3 ayrımı için maksimum sınıflandırma başarısı İki Katmanlı Sinir Ağı (Bilayered Neural Network) sınıflandırıcısında %66,2 olarak elde edilmiştir. Veri setinin üçüncü ayrımında, eğitim seti:134 ve test seti:2 için sınıflandırma başarısı incelenmiştir. Bu ayrımında, maksimum sınıflandırma başarısı Kosinüs KNN (Cosine KNN) sınıflandırıcısında %61,3 olarak elde edilmiştir. Son olarak, eğitim seti:234 ve test seti:1 ayrımı için Xception ile öznetelik çıkarıldıktan sonra maksimum sınıflandırma başarısı Dar Sinir Ağı (Narrow Neural Network) sınıflandırıcısında %72,8 olarak elde edilmiştir.

Veri setinin dört farklı ayrımı için Resnet50, DenseNet-201 ve Xception ağ mimarileri kullanılarak öznitelik çıkarma uygulanmıştır ve daha sonrasında 22 farklı sınıflandırıcı için veri seti ayrımlarının sınıflandırma başarıları elde edilmiştir. Veri seti ayrımları için her bir ağ mimarisinde maksimum başarı değerleri seçilmiştir. Tablo 3.20’de Resnet50 ağ mimarisine göre, Tablo 3.21’de DenseNet-201 ağ mimarisine göre ve Tablo 3.22’de Xception ağ mimarisine göre elde edilen maksimum başarı değerleri verilmiştir.

Tablo 3.20. Resnet50 ağ mimarisine göre maksimum sınıflandırma başarıları

Veri Seti	Alt Uzay Diskriminant (Subspace Discriminant)
Eğitim Seti: 123 Test Seti: 4	%63,6
Eğitim Seti: 124 Test Seti: 3	%54,9
Eğitim Seti: 134 Test Seti: 2	%59,6
Eğitim Seti: 234 Test Seti: 1	%65,9
Ortalama Başarı	%61
Standart Sapma	4,83
Parametre Değerleri	Topluluk yöntemi: Alt Uzay Öğretici Tipi: Diskriminant Öğretici Sayısı: 30 Alt Uzay Boyutu: 500

Tablo 3.21. DenseNet-201 ağı mimarisine göre maksimum sınıflandırma başarıları

Veri Seti	Doğrusal SVM (Linear SVM)	Kübik SVM (Cubic SVM)	Orta Sinir Ağı (Medium Neural Network)	Dar Sinir Ağı (Narrow Neural Network)
Eğitim Seti: 123 Test Seti: 4	%47,1	%46,9	%41,9	%30,4
Eğitim Seti: 124 Test Seti: 3	%41,8	%45,6	%56,6	%46,5
Eğitim Seti: 134 Test Seti: 2	%63,6	%67	%60,7	%54,3
Eğitim Seti: 234 Test Seti: 1	%57,6	%60,5	%66,7	%73,1
Ortalama Başarı	%52,53	%55	%56,48	%51,08
Standart Sapma	9,88	10,46	10,56	17,74
Parametre Değerleri	Çekirdek (Kernel) işlevi: Doğrusal Çekirdek ölçeği: Otomatik Kutu kısıtlama seviyesi: 1 Çok sınıflı yöntem: bire karşı bir Verileri Standartlaştırma: Evet	Çekirdek (Kernel) işlevi: Kübik Çekirdek ölçeği: Otomatik Kutu kısıtlama seviyesi: 1 Çok sınıflı yöntem: bire karşı bir Verileri Standartlaştırma: Evet	Tam bağlı katman sayısı: 1 İlk katman boyutu: 25 Aktivasyon: ReLU Yineleme sınırı: 1000 Düzenleştirme gücü:0 Verileri standartlaştırma: Evet	Tam bağlı katman sayısı: 1 İlk katman boyutu: 10 Aktivasyon: ReLU Yineleme sınırı: 1000 Düzenleştirme gücü:0 Verileri standartlaştırma: Evet

Tablo 3.22. Xception ağ mimarisine göre maksimum sınıflandırma başarıları

Veri Seti	Dar Sinir Ağı (Narrow Neural Network)	Geniş Sinir Ağı (Wide Neural Network)	İki Katmanlı Sinir Ağı (Bilayered Neural Network)	Kosinüs KNN (Cosine KNN)
Eğitim Seti: 123 Test Seti: 4	%58,1	%71,9	%47,3	%60,5
Eğitim Seti: 124 Test Seti: 3	%56,7	%58,9	%66,2	%54,7
Eğitim Seti: 134 Test Seti: 2	%43,2	%49,4	%48,5	%61,3
Eğitim Seti: 234 Test Seti: 1	%72,8	%66,1	%50,3	%49,1
Ortalama Başarı	%57,70	%61,58	%53,08	%56,40
Standart Sapma	12,10	9,70	8,84	5,69
Parametre Değerleri	Tam bağlı katman sayısı: 1 İlk katman boyutu: 10 Aktivasyon: ReLU Yineleme sınırı: 1000 Düzenlileştirme gücü (Lambda): 0 Verileri standartlaştırma: Evet	Tam bağlı katman sayısı: 1 İlk katman boyutu: 100 Aktivasyon: ReLU Yineleme sınırı: 1000 Düzenlileştirme gücü (Lambda): 0 Verileri standartlaştırma: Evet	Tam bağlı katman sayısı: 2 İlk katman boyutu: 10 İkinci katman boyutu: 10 Aktivasyon: ReLU Yineleme sınırı: 1000 Düzenlileştirme gücü (Lambda): 0 Verileri standartlaştırma: Evet	Komşu Sayısı: 10 Uzaklık Ölçümü: Kosinüs Uzaklık Ağırlığı: Eşit Verileri Standartlaştırma: Evet

Öğrenme aktarımı yönteminde 3 farklı ağ mimarisi kullanılarak ortalama başarı değeri elde edilmiştir. Resnet50 ağ mimarisi kullanıldığında Alt Uzak Diskriminant sınıflandırıcısında ortalama başarı değeri %61 olarak bulunmuştur. DenseNet-201 ağ mimarisinde Orta Sinir Ağı sınıflandırıcısı ile ortalama başarı değeri %56,48 olarak elde edilmiştir. Xception ağ mimarisinde Geniş Sinir Ağı sınıflandırıcısı ile ortalama başarı değeri %61,58 olarak elde edilmiştir. Sıfırdan öğrenme yöntemine göre başarı değeri artmıştır. Çünkü, burada kullanılan ağ mimarileri büyük veri kümeleri ile eğitilmiş, iyi genelleme yapabilen modellerdir. Ancak, burada kullanılan ağ mimarileri de bu çalışmada kullanılan veri setine uygun değildir. Öznitelikleri çıkarmak tam anlamıyla sonuç verememiştir. Ayrıca, bu ağ mimarileri ile genellikle görüntü tanıma yönelik çalışma yapılmaktadır. Burada hareket tanınması yapıldığı için düşük bir başarı elde edildiği düşünülmektedir.

3.7. Elde Edilen Sonuçlar

Veri setinde bulunan veriler ve RGB görüntüler dikkatli bir şekilde incelenerek etiketlenmiştir. Veri setinde insan hareketlerini içeren 13 sınıf bilgisi bulunmaktadır. Veri setinde dört farklı kişiden elde edilen veriler bulunmaktadır. Veri setini eğitim ve test verisi olarak ayırmak için 3 kişinin verileri ile eğitim seti, geriye kalan kişinin verileri ile test seti oluşturularak sınıflandırma başarıları elde edilmiştir. İlk aşamada, ham veri setinin sınıflandırma başarısının incelenmesi için her bir veri seti ayrımında maksimum bir sınıflandırma başarısı elde edilmiştir ve sonuçların ortalaması alınarak detaylı bir analiz gerçekleştirilmiştir. Sonuç olarak, elde edilen maksimum ortalama sınıflandırma başarısı Kosinüs KNN sınıflandırıcısında %72,60 olarak bulunmuştur.

Makine öğrenmesinde, sınıflandırma başarısını arttırmak için öznitelik seçme yöntemi uygulanmıştır. Bu aşamada, geriye doğru öznitelik seçme yöntemi kullanılmıştır. Veri setinde bulunan eklemleri temsil eden toplam 15 öznitelik bulunmaktadır. Ham veri seti üzerinde maksimum sınıflandırma başarısının elde edildiği veri seti ayrımı (eğitim seti:134 ve test seti:2) üzerinde sıra ile öznitelikler çıkarılarak maksimum başarı elde edilmeye çalışılmıştır. Bu işlem veri setinde tek bir öznitelik kalana kadar devam etmiştir. Uygulamada, maksimum başarı 5 tane öznitelik çıkarıldığında elde edilmiştir. Bu durumda diğer veri seti ayrımları üzerinde aynı 5 öznitelik çıkarıldığı durum için sınıflandırma

başarısı elde edilmiştir. Bunun sonucunda elde edilen maksimum ortalama başarı değeri Alt Uzay Diskriminant sınıflandırıcısında %74,88 olarak bulunmuştur.

Sınıflandırma başarısını elde etmek için bir sonraki aşamada derin öğrenme tabanlı LSTM mimarisi kullanılmıştır. LSTM mimarisinde, veri seti normalize edilerek işlemler gerçekleştirilmiştir. Sistemin sınıflandırma başarı için parametre değerleri değiştirilerek maksimum başarı elde edilmeye çalışılmıştır. Bunun için makine öğrenmesi yönteminde maksimum başarının elde edildiği veri seti ayrımı (eğitim seti:134 ve test seti:2) kullanılmıştır. Deneysel çalışmayla belirlenen parametre değerleri ile dört veri seti ayrımı için LSTM mimarisi 10 kez çalıştırılmıştır. Bunun sonucunda elde edilen ortalama sınıflandırma başarısı %54 olarak bulunmuştur.

LSTM mimarisinde elde edilen sınıflandırma başarısını arttırmak için bir sonraki aşamada LSTM kullanılarak öznitelik çıkarma uygulanmıştır. Öznitelik çıkarma uygulandıktan sonra makine öğrenmesi tabanlı 29 farklı sınıflandırıcı kullanılarak sistemin sınıflandırma başarısı elde edilmiştir. Her bir veri seti ayrımı için bu işlem sıra ile gerçekleştirilmiştir ve maksimum ortalama başarı değeri SVM sınıflandırıcısında %98,85 olarak elde edilmiştir.

CNN mimarisinde, RGB görüntüler kullanılarak sınıflandırma işlemi gerçekleştirilmiştir. Bu aşamada, RGB görüntüler sıfırdan öğrenme ile 7 katmandan oluşan CNN modeli ile sınıflandırılmıştır. Her bir veri seti ayrımı için sınıflandırma başarısı elde edilmiştir ve ortalama başarı değeri %24,24 olarak bulunmuştur.

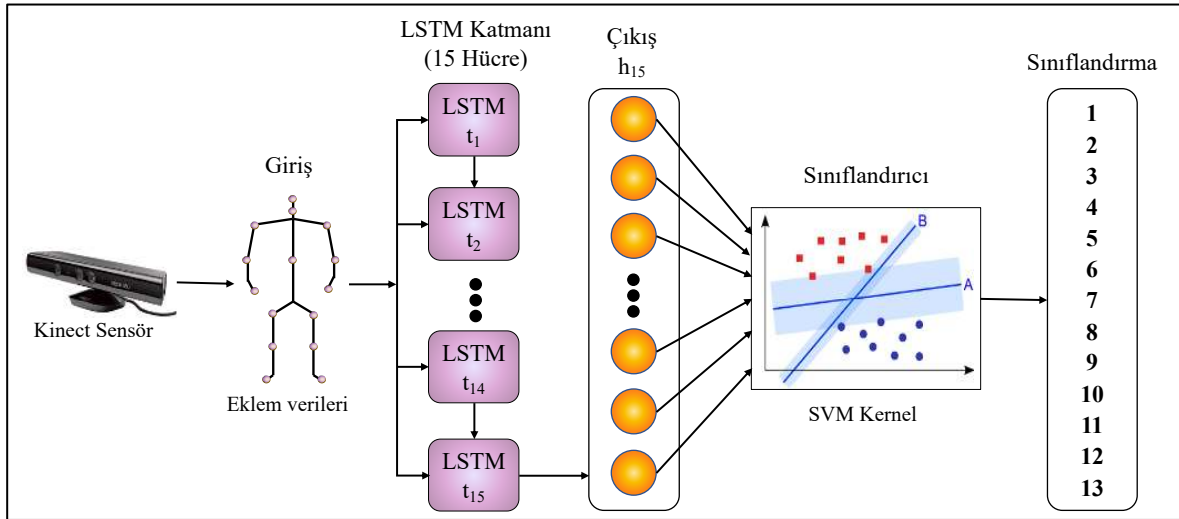
Son aşamada ise öğrenme aktarımı yöntemi kullanılmıştır. Bu yöntemde Resnet50, DenseNet-201 ve Xception ağ mimarileri kullanılarak öznitelik çıkarma uygulanmıştır ve 22 farklı sınıflandırıcı kullanılarak sistemin sınıflandırma başarısı elde edilmiştir. Her bir veri seti ayrımı üzerinde gerçekleştirilen işlemler sonunda Resnet50 ağ mimarisi kullanılarak elde edilen maksimum ortalama başarı değeri Alt Uzay Diskriminant sınıflandırıcısında %61 olarak bulunmuştur. DenseNet-201 ağ mimarisi kullanılarak elde edilen maksimum ortalama başarı değeri Orta Sinir Ağı sınıflandırıcısında %56,48 olarak bulunmuştur. Xception ağ mimarisi kullanılarak elde edilen maksimum ortalama başarı değeri Geniş Sinir Ağı sınıflandırıcısında %61,58 olarak bulunmuştur.

Bu tez çalışmasında farklı mimariler kullanılarak toplamda 72.660 veri üzerinde 13 etiket için sınıflandırma çalışması yapılmıştır. Veri seti ile ilgili elde edilen tüm sonuçlar Tablo 3.23'te verilmiştir.

Tablo 3.23. Sınıflandırma sonuçları

Yöntem	Sınıflandırma Başarısı
Veri setinin elde edildiği çalışma	%84,70
Ham veri	%72,60
Geriye doğru öznitelik çıkarma (BFS)	%74,88
LSTM	%54,00
LSTM ile öznitelik çıkarma	%98,85
Sıfırdan öğrenme	%24,24
Öğrenme aktarımı	%61,58

Tablo 3.23 incelendiğinde, tez çalışması kapsamında gerçekleştirilen LSTM ile öznitelik çıkarma yönteminde en yüksek sınıflandırma başarısı elde edilmiştir. Bu başarı değeri, veri setinin elde edildiği çalışmada elde edilen sonuca göre yüksektir. Bu yöntemin akış şeması Şekil 3.13'te verilmiştir.



Şekil 3.13. LSTM ile öznitelik çıkarma yönteminin akış şeması.

4. SONUÇ VE ÖNERİLER

İnsanın sahip olduğu duruş stilleri kişinin sağlığı açısından büyük önem taşımaktadır. Farkında olmadan yapılan duruş hataları gerek şimdi gerek ilerleyen dönemlerde ciddi sağlık sorunlarına yol açmaktadır. Son yıllarda kullanım alanı oldukça gelişen görüntü işleme ve sınıflandırma algoritmaları bu alanda da kullanılmaya başlanmıştır. Öğrenme tabanlı algoritmaların yardımı ile çeşitli görüntüleme yöntemleri ile elde edilen görüntülerde, insan duruş tanımlaması yapılabilmektedir. Bu sınıflandırma algoritmalarının öncül yöntemleri makine öğrenmesi ve derin öğrenmedir. Bunun yanı sıra son yıllarda insan duruşunu algılamada insan eklemlerini tespit edebilen sensorlerin kullanımı da artmıştır. Bu çalışmada, Microsoft Kinect sensörü kullanılarak elde edilen insan duruş bilgileri ve görüntülerini içeren bir çalışmanın veri seti kullanılmıştır. Bu veri setinde, farklı insanların farklı hareketlerini/duruşlarını içeren veriler bulunmaktadır. Bu çalışma kapsamında veri seti üzerinde sınıflandırma algoritmaları kullanılarak bu algoritmaların performansları elde edilmiş ve bu performanslar karşılaştırılmıştır.

Veri seti iki farklı şekilde incelenmiştir; sensör verileri ve RGB görüntüler. Elde edilen sonuçlar incelendiğinde sensör verileri ile yapılan çalışmalarda daha yüksek bir başarı gözlemlenmiştir. Veri setinde bulunan 4 kişiden 3'ü sağ elini kullanılırken, geriye kalan kişi sol elini kullanmaktadır. Veri setinin bu şekilde ayrılması sonuçları olumsuz yönde etkilemiştir. Gelecek çalışmalarda, daha dinamik bir model kullanılarak daha başarılı sonuçlar elde edilebilir. En başarılı sonuç LSTM algoritması ile öznetelik çıkarıldığında elde edilmiştir. Bu tez çalışması kapsamında, bu yöntem kullanılarak başarı değerinin arttığı gözlemlenmiştir.

Birçok yaşlı insan bel ağrısı, boyun ağrısı gibi rahatsızlıklardan şikayetçi olmaktadır. Bunların en temel sebebi ise gençlik yıllarında yapılan duruş bozukluklarıdır. Özellikle insanlar çalışırken yanlış bir pozisyon sergilediklerinde ve bunu sık sık tekrar ettiklerinde kalıcı bir rahatsızlık ile karşılaşabilmektedir. Bu durumlar düşünüldüğünde, gelecekte çalışanların duruş bozukluklarının algılanması ve duruşların sınıflandırılması; bunun sonucunda da yanlış duruş sergilemesi ile ilgili olarak çalışana bir uyarı mesajı gönderilmesi üzerine bir sistem yapılabilir.

KAYNAKLAR

- [1] W. and eds Sammut, *Encyclopedia of Machine Learning and Data Mining New York : Springer*. ISBN: 978-1-4899-7685-7, 2017.
- [2] M. Luckert and M. Schaffer-Kehnert, ‘Using Machine Learning Methods for Evaluating the Quality of Technical Documents’, M.S. thesis, Dept. Computer Science, Linnaeus Univ., Sweden., 2015.
- [3] M. Wang, Y. Cui, X. Wang, S. Xiao, and J. Jiang, ‘Machine Learning for Networking: Workflow, Advances and Opportunities’, *IEEE Network*, vol. 32, no. 2, pp. 92–99, Mar. 2018, doi: 10.1109/MNET.2017.1700200.
- [4] J. Zhou, A. H. Gandomi, F. Chen, and A. Holzinger, ‘Evaluating the Quality of Machine Learning Explanations: A Survey on Methods and Metrics’, *Electronics*, vol. 10, no. 5, p. 593, Mar. 2021, doi: 10.3390/electronics10050593.
- [5] Elif Yıldırım, ‘Yapay Sinir Ağı (Artificial Neural Network) Nedir?’ <https://www.veribilimiokulu.com/yapay-sinir-agiartificial-neural-network-nedir/>, erişim: 14.01.2022, 11:20.
- [6] M. S. Rozario, A. Thomas, and D. Mathew, ‘Performance Comparison of Multiple Speech Features for Speaker Recognition using Artifical Neural Network’, in *2019 9th International Conference on Advances in Computing and Communication (ICACC)*, Nov. 2019, pp. 234–239, doi: 10.1109/ICACC48162.2019.8986182.
- [7] S. Amiri, S. Salimzadeh, and A. S. Z. Belloum, ‘A Survey of Scalable Deep Learning Frameworks’, in *2019 15th International Conference on eScience (eScience)*, Sep. 2019, pp. 650–651, doi: 10.1109/eScience.2019.00102.
- [8] J. Liu, ‘Survey of the Image Recognition Based on Deep Learning Network for Autonomous Driving Car’, in *2020 5th International Conference on Information Science, Computer Technology and Transportation (ISCTT)*, Nov. 2020, pp. 1–6, doi: 10.1109/ISCTT51595.2020.00007.
- [9] P. Chaudhari and A. Bhusari, ‘Transfer Optimistic Outcome-based Learning for Mature Behavior of Machine in Deep Learning’, in *2020 IEEE International*

Conference on Advent Trends in Multidisciplinary Research and Innovation (ICATMRI), Dec. 2020, pp. 1–4, doi: 10.1109/ICATMRI51801.2020.9398404.

- [10] Y. Dong and G. Liang, ‘Research and Discussion on Image Recognition and Classification Algorithm Based on Deep Learning’, in *2019 International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI)*, Nov. 2019, pp. 274–278, doi: 10.1109/MLBDBI48998.2019.00061.
- [11] Microsoft Azure, ‘Derin Öğrenme nedir’. <https://azure.microsoft.com/tr-tr/overview/what-is-deep-learning/>, erişim: 12.01.2022, 11:55.
- [12] B. Bonnechère, V. Sholukha, B. Jansen, L. Omelina, M. Rooze, and S. Van Sint Jan, ‘Determination of Repeatability of Kinect Sensor’, *Telemedicine and e-Health*, vol. 20, no. 5, pp. 451–453, May 2014, doi: 10.1089/tmj.2013.0247.
- [13] Jaeyong Sung, C. Ponce, B. Selman, and A. Saxena, ‘Unstructured human activity detection from RGBD images’, in *2012 IEEE International Conference on Robotics and Automation*, May 2012, pp. 842–849, doi: 10.1109/ICRA.2012.6224591.
- [14] T. Wang *et al.*, ‘A Gait Assessment Framework for Depression Detection Using Kinect Sensors’, *IEEE Sensors Journal*, vol. 21, no. 3, pp. 3260–3270, 2021, doi: 10.1109/JSEN.2020.3022374.
- [15] N. Asilah Saidin and S. A. Abdul Shukor, ‘An Analysis of Kinect-Based Human Fall Detection System’, *Proceeding - 2020 IEEE 8th Conference on Systems, Process and Control, ICSPC 2020*, no. December, pp. 220–224, 2020, doi: 10.1109/ICSPC50992.2020.9305797.
- [16] B. Yao, H. Gao, and X. Su, ‘Human motion recognition by three-view kinect sensors in virtual basketball training’, *IEEE Region 10 Annual International Conference, Proceedings/TENCON*, vol. 2020-Novem, pp. 1260–1265, 2020, doi: 10.1109/TENCON50793.2020.9293892.
- [17] T. Saratean, M. Antal, C. Pop, T. Cioara, I. Anghel, and I. Salomie, ‘A Physiotherapy Coaching System based on Kinect Sensor’, *Proceedings - 2020 IEEE 16th International Conference on Intelligent Computer Communication and Processing, ICCP 2020*, pp. 535–540, 2020, doi: 10.1109/ICCP51029.2020.9266178.

- [18] A. D. Q. Burle, T. B. D. G. Lafayette, J. R. Fonseca, V. Teichrieb, and A. E. F. Da Gama, 'Real-time approach for gait analysis using the Kinect v2 sensor for clinical assessment purpose', *Proceedings - 2020 22nd Symposium on Virtual and Augmented Reality, SVR 2020*, pp. 144–153, 2020, doi: 10.1109/SVR51698.2020.00034.
- [19] Z. Xu *et al.*, 'Back Shape Measurement and Three-Dimensional Reconstruction of Spinal Shape Using One Kinect Sensor', in *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*, Apr. 2020, pp. 1–5, doi: 10.1109/ISBI45749.2020.9098481.
- [20] J. Kondragunta and G. Hirtz, 'Gait Parameter Estimation of Elderly People using 3D Human Pose Estimation in Early Detection of Dementia', *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, vol. 2020-July, pp. 5798–5801, 2020, doi: 10.1109/EMBC44109.2020.9175766.
- [21] N. K. Mangal and A. K. Tiwari, 'Kinect v2 tracked Body Joint Smoothing for Kinematic Analysis in Musculoskeletal Disorders', *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, vol. 2020-July, pp. 5769–5772, 2020, doi: 10.1109/EMBC44109.2020.9175492.
- [22] S. Nehra and J. L. Raheja, 'Unobtrusive and Non-Invasive Human Activity Recognition using Kinect Sensor', *Indo - Taiwan 2nd International Conference on Computing, Analytics and Networks, Indo-Taiwan ICAN 2020 - Proceedings*, pp. 58–63, 2020, doi: 10.1109/Indo-TaiwanICAN48429.2020.9181359.
- [23] T. H. Tan, M. Gochoo, H. S. Chen, S. H. Liu, and Y. F. Huang, 'Activity Recognition Based on DCNN and Kinect RGB Images', *2020 International Conference on Fuzzy Theory and Its Applications, iFUZZY 2020*, pp. 60–63, 2020, doi: 10.1109/iFUZZY50310.2020.9297815.
- [24] T. Kamizono, H. Abe, K. Baba, S. Takano, and K. Murakami, 'Towards activity recognition of learners by kinect', *Proceedings - 2014 IIAI 3rd International Conference on Advanced Applied Informatics, IIAI-AAI 2014*, pp. 177–180, 2014, doi: 10.1109/IIAI-AAI.2014.45.

- [25] M. Li, F. Wei, Y. Li, S. Zhang, and G. Xu, 'Three-Dimensional Pose Estimation of Infants Lying Supine Using Data from a Kinect Sensor with Low Training Cost', *IEEE Sensors Journal*, vol. 21, no. 5, pp. 6904–6913, 2021, doi: 10.1109/JSEN.2020.3037121.
- [26] W. Ren, O. Ma, H. Ji, and X. Liu, 'Human Posture Recognition Using a Hybrid of Fuzzy Logic and Machine Learning Approaches', *IEEE Access*, vol. 8, pp. 135628–135639, 2020, doi: 10.1109/ACCESS.2020.3011697.
- [27] L. T. De Paolis and V. De Luca, 'The performance of Kinect in assessing the shoulder joint mobility', *IEEE Medical Measurements and Applications, MeMeA 2020 - Conference Proceedings*, pp. 1–6, 2020, doi: 10.1109/MeMeA49120.2020.9137213.
- [28] Ö. F. İnce, I. F. Ince, M. E. Yıldırım, J. S. Park, J. K. Song, and B. W. Yoon, 'Human activity recognition with analysis of angles between skeletal joints using a RGB-depth sensor', *ETRI Journal*, vol. 42, no. 1, pp. 78–89, 2020, doi: 10.4218/etrij.2018-0577.
- [29] P. K. Pisharady and M. Saerbeck, 'Kinect based body posture detection and recognition system', *International Conference on Graphic and Image Processing (ICGIP 2012)*, vol. 8768, no. March 2013, p. 87687F, 2013, doi: 10.1117/12.2009926.
- [30] W. J. Wang, J. W. Chang, S. F. Haung, and R. J. Wang, 'Human Posture Recognition Based on Images Captured by the Kinect Sensor', *International Journal of Advanced Robotic Systems*, vol. 13, no. 2, pp. 1–16, 2016, doi: 10.5772/62163.
- [31] A. Abobakr, D. Nahavandi, J. Iskander, M. Hossny, S. Nahavandi, and M. Smets, 'RGB-D human posture analysis for ergonomic studies using deep convolutional neural network', *2017 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2017*, vol. 2017-Janua, pp. 2885–2890, 2017, doi: 10.1109/SMC.2017.8123065.
- [32] O. Rihawi, D. Merad, and J. L. Damoiseaux, '3D-AD: 3D-autism dataset for repetitive behaviours with kinect sensor', *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS 2017*, no. August, 2017, doi: 10.1109/AVSS.2017.8078544.
- [33] R. Kumarasiri, A. Niroshan, Z. Lantra, T. Madusanka, C. U. S. Edussooriya, and R.

- Rodrigo, ‘Gait Analysis Using RGBD Sensors’, in *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, Nov. 2018, pp. 460–465, doi: 10.1109/ICARCV.2018.8581295.
- [34] Y. Xu, J. Chen, Q. Yang, and Q. Guo, ‘Human Posture Recognition and fall detection Using Kinect V2 Camera’, in *2019 Chinese Control Conference (CCC)*, Jul. 2019, pp. 8488–8493, doi: 10.23919/ChiCC.2019.8865732.
- [35] H. Sun, G. Zhu, X. Cui, and J.-X. Wang, ‘Kinect-based intelligent monitoring and warning of students’ sitting posture’, in *2021 6th International Conference on Automation, Control and Robotics Engineering (CACRE)*, Jul. 2021, pp. 338–342, doi: 10.1109/CACRE52464.2021.9501372.
- [36] S. H. Chowdhury, M. Al Amin, A. K. M. M. Rahman, M. A. Amin, and A. A. Ali, ‘Assessment of Rehabilitation Exercises from Depth Sensor Data’, in *2021 24th International Conference on Computer and Information Technology (ICCIT)*, Dec. 2021, pp. 1–7, doi: 10.1109/ICCIT54785.2021.9689826.
- [37] D. K. Vishwakarma and K. Jain, ‘Three-dimensional human activity recognition by forming a movement polygon using posture skeletal data from depth sensor’, *ETRI Journal*, Jan. 2022, doi: 10.4218/etrij.2020-0101.
- [38] F. Ahmed, A. S. M. H. Bari, and M. L. Gavrilova, ‘Emotion Recognition From Body Movement’, *IEEE Access*, vol. 8, pp. 11761–11781, 2020, doi: 10.1109/ACCESS.2019.2963113.
- [39] D. Arpitha, M. Balasubrahmanyam, and D. A. Kumar, ‘Depth based Indian Classical Dance Mudra’s Recognition using Support Vector Machine’, in *2022 4th International Conference on Smart Systems and Inventive Technology (ICSSIT)*, Jan. 2022, pp. 885–888, doi: 10.1109/ICSSIT53264.2022.9716486.
- [40] Microsoft, ‘Xbox 360 + kinect’, 2012, [Online]. Available: <http://www.xbox.com/en-us/kinect>, erişim: 28.11.2021, 12:35.
- [41] G. S. Lahan, A. K. Talukdar, and K. K. Sarma, ‘Action Recognition from Depth Video Sequences Using Microsoft Kinect’, in *2019 Fifth International Conference on Image Information Processing (ICIIP)*, Nov. 2019, pp. 35–40, doi:

10.1109/ICIIP47207.2019.8985957.

- [42] Microsoft, 'Kinect hardware', 2014. <https://developer.microsoft.com/en-us/windows/kinect/hardware>, erişim : 29.11.2021, 14:05.
- [43] X. Xu and R. W. McGorry, 'The validity of the first and second generation Microsoft Kinect™ for identifying joint center locations during static postures', *Applied Ergonomics*, vol. 49, pp. 47–54, Jul. 2015, doi: 10.1016/j.apergo.2015.01.005.
- [44] Roanna Z. Lun, 'Human Activity Tracking and Recognition Using Kinect Sensor', Ph.D. dissertation, Electrical and Computer Engineering, Cleveland State University, Ohio, 2018.
- [45] Vikipedi, 'Kinect', 2021. <https://tr.wikipedia.org/wiki/Kinect>, erişim: 01.12.2021, 09:45.
- [46] R. Mojtahedzadeh, 'Robot Obstacle Avoidance using the Kinect', M.S. thesis, Computer Science, KTH CSC, Stockholm, Sweden, 2011.
- [47] C. Chen, R. Jafari, and N. Kehtarnavaz, 'A Real-Time Human Action Recognition System Using Depth and Inertial Sensor Fusion', *IEEE Sensors Journal*, vol. 16, no. 3, pp. 773–781, Feb. 2016, doi: 10.1109/JSEN.2015.2487358.
- [48] L. Cruz, D. Lucio, and L. Velho, 'Kinect and RGBD Images: Challenges and Applications', in *2012 25th SIBGRAPI Conference on Graphics, Patterns and Images Tutorials*, Aug. 2012, pp. 36–49, doi: 10.1109/SIBGRAPI-T.2012.13.
- [49] M. H. Nadhif, A. P. Hadiputra, Y. Whulanza, and S. Supriadi, 'Gait Analysis for Biometric Surveillances Using Kinect™: A Study Case of Axial Skeletal Movements', in *2019 16th International Conference on Quality in Research (QIR): International Symposium on Electrical and Computer Engineering*, Jul. 2019, pp. 1–4, doi: 10.1109/QIR.2019.8898273.
- [50] A. S. M. H. Bari and M. L. Gavrilova, 'Artificial Neural Network Based Gait Recognition Using Kinect Sensor', *IEEE Access*, vol. 7, pp. 162708–162722, 2019, doi: 10.1109/ACCESS.2019.2952065.
- [51] T. Sukumpee, N. Theera-Umpon, S. Chamnongkich, and S. Auephanwiriyakul,

- ‘Kinematic-based Knee Angle Correction for Gait Analysis Using Single Kinect Sensor’, in *2018 8th IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, Nov. 2018, pp. 212–216, doi: 10.1109/ICCSCE.2018.8685025.
- [52] R. A. Clark *et al.*, ‘Instrumenting gait assessment using the Kinect in people living with stroke: reliability and association with balance tests’, *Journal of neuroengineering and rehabilitation*, vol. 12, p. 15, 2015, doi: 10.1186/s12984-015-0006-8.
- [53] E. Babaians, N. Khazaei Korghond, A. Ahmadi, M. Karimi, and S. S. Ghidary, ‘Skeleton and visual tracking fusion for human following task of service robots’, in *2015 3rd RSI International Conference on Robotics and Mechatronics (ICROM)*, Oct. 2015, pp. 761–766, doi: 10.1109/ICRoM.2015.7367878.
- [54] V. Nasteski, ‘An overview of the supervised machine learning methods’, *HORIZONS.B*, vol. 4, pp. 51–62, Dec. 2017, doi: 10.20544/HORIZONS.B.04.1.17.P05.
- [55] R. Bekkerman, M. Bilenko, and J. Langford, *Scaling Up Machine Learning: Parallel and Distributed Approaches*. Cambridge: Cambridge University Press. doi:10.1017/CBO9781139042918, 2011.
- [56] O. Güldali, ‘Deep Feature Transfer From Deep Learning Models Into Machine Learning Algorithms To Classify Covid-19 From Chest X-Ray Images’, M.S. thesis, Dept. of Mathematical Engineering, Istanbul Technical University, İstanbul, Turkey, 2021.
- [57] Ning Li, Li Zhao, Ai-Xia Chen, Qing-Wu Meng, and Guo-Fang Zhang, ‘A new heuristic of the decision tree induction’, in *2009 International Conference on Machine Learning and Cybernetics*, Jul. 2009, pp. 1659–1664, doi: 10.1109/ICMLC.2009.5212227.
- [58] B. Görmez, ‘Adli Bilişimde Makine Öğrenmesi: Makine Öğrenmesi Algoritmaları ile Terör Olaylarının Tahmin Edilmesi Çalışması’, MS. Tez, Dept. Disiplinler arası Adli Bilimler, Ankara Üniversitesi, Ankara, Türkiye, 2021.

- [59] N. E. Sahla, ‘A deep learning prediction model for object classification’, M.S. thesis, Dept. Mechanical Engineering, Delft University, Holland, 2018.
- [60] Jing Peng, Peng Zhang, and N. Riedel, ‘Discriminant Learning Analysis’, *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 38, no. 6, pp. 1614–1625, Dec. 2008, doi: 10.1109/TSMCB.2008.2002852.
- [61] J. Ghosh and S. B. Shuvo, ‘Improving Classification Model’s Performance Using Linear Discriminant Analysis on Linear Data’, in *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, Jul. 2019, pp. 1–5, doi: 10.1109/ICCCNT45670.2019.8944632.
- [62] B. M. Gayathri and C. P. Sumathi, ‘Feature selection using Linear Discriminant Analysis for breast cancer dataset’, in *2018 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, Dec. 2018, pp. 1–5, doi: 10.1109/ICCIC.2018.8782399.
- [63] Tae-Kyun Kim and J. Kittler, ‘Locally linear discriminant analysis for multimodally distributed classes for face recognition with a single model image’, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 3, pp. 318–327, Mar. 2005, doi: 10.1109/TPAMI.2005.58.
- [64] C. Cortes and V. Vapnik, ‘Support-vector networks’, *Machine Learning*, vol. 20, no. 3, pp. 273–297, Sep. 1995, doi: 10.1007/BF00994018.
- [65] A. Ben-Hur, D. Horn, H. T. Siegelmann, and V. Vapnik, ‘A support vector clustering method’, in *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, vol. 2, pp. 724–727, doi: 10.1109/ICPR.2000.906177.
- [66] W. Commons, ‘Zwei mögliche trenngeraden mit verschiedenen randgrößen’. https://de.wikipedia.org/wiki/Support_Vector_Machine, erişim: 10:01.2021, 21:15.
- [67] M. Çetintaş, ‘Comparison Of Machine Learning Methods For Fault Analysis In Induction Motors’, M.S. thesis, Dept. of Electrical and Electronic Engineering, Eskişehir Technical University, Eskişehir, Turkey, 2021.
- [68] E. Fix and J. L. Hodges, ‘Discriminatory Analysis. Nonparametric Discrimination:

Consistency Properties’, *International Statistical Review / Revue Internationale de Statistique*, vol. 57, no. 3, p. 238, Dec. 1989, doi: 10.2307/1403797.

- [69] R. Ye, Z. Le, and P. N. Suganthan, ‘K-nearest neighbor based bagging SVM pruning’, *Proceedings of the 2013 IEEE Symposium on Computational Intelligence and Ensemble Learning, CIEL 2013 - 2013 IEEE Symposium Series on Computational Intelligence, SSCI 2013*, no. April, pp. 25–30, 2013, doi: 10.1109/CIEL.2013.6613136.
- [70] A. Glowacz, ‘DC Motor Fault Analysis with the Use of Acoustic Signals, Coiflet Wavelet Transform, and K-Nearest Neighbor Classifier’, *Archives of Acoustics*, vol. 40, no. 3, pp. 321–327, Sep. 2015, doi: 10.1515/aoa-2015-0035.
- [71] P. Harrington, *Machine Learning in Action*. Manning Publications, Shelter Island, 2012.
- [72] Z. Jiang, H. Liu, B. Fu, and Z. Wu, ‘A Novel Bayesian Ensemble Pruning Method’, in *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, Dec. 2016, pp. 1205–1212, doi: 10.1109/ICDMW.2016.0174.
- [73] C. C, ‘Prediction of Heart Disease using Different KNN Classifier’, in *2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS)*, May 2021, pp. 1186–1194, doi: 10.1109/ICICCS51141.2021.9432178.
- [74] J. Wang and S. Luo, ‘Exploiting Ensemble Method in Semi-Supervised Learning’, in *2006 International Conference on Machine Learning and Cybernetics*, 2006, pp. 1104–1107, doi: 10.1109/ICMLC.2006.258568.
- [75] L. Rokach, ‘Ensemble-based classifiers’, *Artificial Intelligence Review*, vol. 33, no. 1–2, pp. 1–39, Feb. 2010, doi: 10.1007/s10462-009-9124-7.
- [76] Z. H. Zhou, *Ensemble methods: foundations and algorithms*, no. Schapire 1990. CRC Press, 2012.
- [77] K. Alan, ‘Veri Madenciliğine Giriş’. <https://medium.com/@kadiralan021/veri-madenciliğine-giriş-fa0a63e655fb>, erişim: 22.12.2021, 12:42.
- [78] K. A. Taher, B. Mohammed Yasin Jisan, and M. M. Rahman, ‘Network Intrusion

- Detection using Supervised Machine Learning Technique with Feature Selection’, in *2019 International Conference on Robotics,Electrical and Signal Processing Techniques (ICREST)*, Jan. 2019, pp. 643–646, doi: 10.1109/ICREST.2019.8644161.
- [79] B. Ergün, ‘Gaz Sensörlerinde Çok Değişkenli Kalibrasyon Yöntemlerinin İncelenmesi’, M.S. tez, Elekt. Elektro. Müh. Ana Bilim Dalı, Başkent Üniversitesi, Ankara, Türkiye, 2018.
- [80] J. Cai, J. Luo, S. Wang, and S. Yang, ‘Feature selection in machine learning: A new perspective’, *Neurocomputing*, vol. 300, pp. 70–79, Jul. 2018, doi: 10.1016/j.neucom.2017.11.077.
- [81] S. Khalid, T. Khalil, and S. Nasreen, ‘A survey of feature selection and feature extraction techniques in machine learning’, in *2014 Science and Information Conference*, Aug. 2014, pp. 372–378, doi: 10.1109/SAI.2014.6918213.
- [82] L. Li *et al.*, ‘Prediction of bacterial protein subcellular localization by incorporating various features into Chou’s PseAAC and a backward feature selection approach’, *Biochimie*, vol. 104, pp. 100–107, Sep. 2014, doi: 10.1016/j.biochi.2014.06.001.
- [83] G. B. Ergün and S. Güney, ‘A Comparison of the Multivariate Calibration Methods with Feature Selection for Gas Sensors ’ Long - Term Drift Effect’, *Uluslararası Teknolojik Bilimler Dergisi*, vol. 11, no. 3, pp. 170–176, 2019.
- [84] H. Wechsler, *Neural Networks for Perception Theory of the Backpropagation Neural Network Based on “nonindent” by Robert Hecht-Nielsen*. Academic Press, 65-92, 1992.
- [85] A. E. Öztürk, ‘Development Of Rw-Uav Detection Model Using Deep Learning Training With A Synthetic Dataset’, Ph.D. thesis, Dept. of Electrical and Electronic Engineering, Gaziantep University, Gaziantep, Turkey, 2021.
- [86] The MathWorks, ‘What Is Deep Learning?’ <https://www.mathworks.com/discovery/deep-learning.html>, erişim: 10.01.2022, 14:55.
- [87] Di Guan, ‘Classical Architectures in CNN’, [Online]. Available:

<https://guandi1995.github.io/Classical-CNN-architecture>, erişim: 04.12.2021, 18.08.

- [88] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, ‘Gradient-based learning applied to document recognition’, *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998, doi: 10.1109/5.726791.
- [89] F. Wang *et al.*, ‘Generative adversarial networks and convolutional neural networks based weather classification model for day ahead short-term photovoltaic power forecasting’, *Energy Conversion and Management*, vol. 181, pp. 443–462, Feb. 2019, doi: 10.1016/j.enconman.2018.11.074.
- [90] B. Dikici, ‘Object Detection Using Convolutional Neural Network’, M.S. thesis, Dept. of Electrical and Electronic Engineering, Gaziantep Univ., Turkey, 2021.
- [91] H. Srivastava and K. Sarawadekar, ‘A Depthwise Separable Convolution Architecture for CNN Accelerator’, in *2020 IEEE Applied Signal Processing Conference (ASPCON)*, Oct. 2020, pp. 1–5, doi: 10.1109/ASPCON49795.2020.9276672.
- [92] Y. Heryadi and H. L. H. S. Warnars, ‘Learning temporal representation of transaction amount for fraudulent transaction recognition using CNN, Stacked LSTM, and CNN-LSTM’, in *2017 IEEE International Conference on Cybernetics and Computational Intelligence (CyberneticsCom)*, Nov. 2017, pp. 84–89, doi: 10.1109/CYBERNETICSCOM.2017.8311689.
- [93] G. B. ERGÜN, S. GÜNEY, and T. G. ERGÜN, ‘Köpeklerdeki Uzun Kemiklerin Evrimsel Sinir Ağları Kullanılarak Sınıflandırılması’, *Fırat Üniversitesi Mühendislik Bilimleri Dergisi*, vol. 33, no. 1, pp. 125–132, Feb. 2021, doi: 10.35234/fumbd.759340.
- [94] J. Gu, M. Zhang, T. Chen, Z. Sun, Y. Ma, and B. Yu, ‘Recent advances in convolutional neural network acceleration’, *Pattern Recognit*, vol. 77, pp. 354–377, 2018, doi: 10.1016/j.neucom.2018.09.038.
- [95] S. Ghimire, R. C. Deo, N. Raj, and J. Mi, ‘Deep solar radiation forecasting with convolutional neural network and long short-term memory network algorithms’, *Applied Energy*, vol. 253, p. 113541, Nov. 2019, doi: 10.1016/j.apenergy.2019.113541.

- [96] N. Mowla, ‘Weed Detection And Classification Using Deep Learning’, M.S. thesis, Dept. of Electrical and Electronic Engineering, Çukurova Univ., Turkey, 2021.
- [97] Y. Bengio, I. Goodfellow, and A. Courville, *Deep learning*. Cambridge, MA, USA: The MIT Press, 2016.
- [98] A. Burkov, ‘The Hundred-Page Machine Learning Book’, pp. 98–105, 2019.
- [99] N. Kwak, ‘Introduction to Convolutional Neural Networks, Korea’, 2016, doi: 10.1007/978-1-4842-5648-0.
- [100] V. Dumoulin and F. Visin, ‘A guide to convolution arithmetic for deep learning’, Mar. 2016, [Online]. Available: <http://arxiv.org/abs/1603.07285>.
- [101] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, ‘Convolutional neural networks: an overview and application in radiology’, *Insights into Imaging*, vol. 9, no. 4, pp. 611–629, Aug. 2018, doi: 10.1007/s13244-018-0639-9.
- [102] W. Chen, ‘The One-Stop Guide to Convolutional Neural Networks’, <https://williamjchen.medium.com/the-one-stop-guide>, erişim: 25.12.2021, 16:23.
- [103] Innova Blog, ‘Derin öğrenme hakkında bilmeniz gereken her şey’. <https://www.innova.com.tr/tr/blog/buyuk-veri-blog/derin-ogrenme-hakkinda-bilmeniz-gereken-her-sey>, erişim: 01.02.2022, 10:30.
- [104] S. Candemir, X. V. Nguyen, L. R. Folio, and L. M. Prevedello, ‘Training Strategies for Radiology Deep Learning Models in Data-limited Scenarios’, *Radiology: Artificial Intelligence*, vol. 3, no. 6, Nov. 2021, doi: 10.1148/ryai.2021210014.
- [105] Niklas Donges, ‘What Is Transfer Learning? Exploring the Popular Deep Learning Approach.’ .
- [106] Y. Gülsüm and M. N. Yeğin, ‘Öğrenme Aktarımı/Transfer Learning’. [https://medium.com/novaresearchlab/öğrenme-aktarımı-transfer-learning-c0b8126965c4#:~:text=Öğrenme aktarımı \(Transfer learning\) makine, karşılaştığında o bilgiyi kullanmasıdır. , Erişim: 16.02.2022, 15:22](https://medium.com/novaresearchlab/öğrenme-aktarımı-transfer-learning-c0b8126965c4#:~:text=Öğrenme aktarımı (Transfer learning) makine, karşılaştığında o bilgiyi kullanmasıdır. , Erişim: 16.02.2022, 15:22).
- [107] G. B. Cangöz, ‘Köpeklerin Uzun Kemiklerinin Ve Uzun Kemiklerindeki Kırıkların

Sınıflandırılması’, PhD. tez, Elekt. Elektro. Müh. Ana Bilim Dalı, Başkent Üniversitesi, Ankara, Türkiye, 2022.

- [108] A. Krizhevsky, I. Sutskever, and G. E. Hinton, ‘ImageNet classification with deep convolutional neural networks’, in *Advances in neural information processing systems*, pp. 1097–1105, May 2012, doi: 10.1145/3065386.
- [109] Great Learning Team, ‘Introduction to ReLU Activation Function’. <https://www.mygreatlearning.com/blog/relu-activation-function/> , erişim : 11.01.22, 11:00.
- [110] J. Brownlee, ‘A Gentle Introduction to the Rectified Linear Unit (ReLU)’. <https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/>, erişim : 11.01.22, 10:45.
- [111] A. Kızrak, ‘Derin Öğrenme İçin Aktivasyon Fonksiyonlarının Karşılaştırılması’. <https://ayyucekizrak.medium.com/derin-öğrenme-için-aktivasyon-fonksiyonlarının-karşılaştırılması-cee17fd1d9cd>, erişim: 11.01.22, 11.30.
- [112] K. He, X. Zhang, S. Ren, and J. Sun, ‘Deep Residual Learning for Image Recognition’, in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.
- [113] S. Das, ‘CNN Architectures: LeNet, AlexNet, VGG, GoogLeNet, ResNet and more...’. <https://medium.com/analytics-vidhya/cnns-architectures-lenet-alexnet-vgg-googlenet-resnet-and-more-666091488df5> , erişim: 10.01.22, 11.00.
- [114] Great Learning Team, ‘Deep Learning Neural Networks Introduction to Resnet or Residual Network’. <https://www.mygreatlearning.com/blog/resnet/>, erişim: 10.01.22, 16.00.
- [115] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, ‘Densely connected convolutional networks’, *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 2261–2269, 2017, doi: 10.1109/CVPR.2017.243.
- [116] Gaurav Singhal, ‘Introduction to DenseNet with TensorFlow’.

<https://www.pluralsight.com/guides/introduction-to-densenet-with-tensorflow>,
erişim : 11.01.22, 15:30.

- [117] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, ‘Rethinking the Inception Architecture for Computer Vision’, in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 2818–2826, doi: 10.1109/CVPR.2016.308.
- [118] M. M. Leonardo, T. J. Carvalho, E. Rezende, R. Zucchi, and F. A. Faria, ‘Deep Feature-Based Classifiers for Fruit Fly Identification (Diptera: Tephritidae)’, in *2018 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, Oct. 2018, pp. 41–47, doi: 10.1109/SIBGRAPI.2018.00012.
- [119] F. Chollet, ‘Xception: Deep Learning with Depthwise Separable Convolutions’, in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, pp. 1800–1807, doi: 10.1109/CVPR.2017.195.
- [120] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, ‘Learning representations by back-propagating errors’, *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986, doi: 10.1038/323533a0.
- [121] D.-J. Choi, J.-H. Han, S.-U. Park, and S.-K. Hong, ‘Comparative Study of CNN and RNN for Motor fault Diagnosis Using Deep Learning’, in *2020 IEEE 7th International Conference on Industrial Engineering and Applications (ICIEA)*, Apr. 2020, pp. 693–696, doi: 10.1109/ICIEA49774.2020.9102072.
- [122] X. Tian, ‘Modeling and application of nonlinear time series based on combinatorial model’, Xi’an University of Electronic Science and Technology, 2017.
- [123] S. Hochreiter and J. Schmidhuber, ‘Long Short-Term Memory’, *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.
- [124] T. Toharudin, R. S. Pontoh, R. E. Caraka, S. Zahroh, Y. Lee, and R. C. Chen, ‘Employing long short-term memory and Facebook prophet model in air temperature forecasting’, *Communications in Statistics - Simulation and Computation*, pp. 1–24, Jan. 2021, doi: 10.1080/03610918.2020.1854302.

- [125] Q. Ye, X. Yang, C. Chen, and J. Wang, ‘River Water Quality Parameters Prediction Method Based on LSTM-RNN Model’, in *2019 Chinese Control And Decision Conference (CCDC)*, Jun. 2019, pp. 3024–3028, doi: 10.1109/CCDC.2019.8832885.
- [126] Y. Jia, Z. Wu, Y. Xu, D. Ke, and K. Su, ‘Long Short-Term Memory Projection Recurrent Neural Network Architectures for Piano’s Continuous Note Recognition’, *Journal of Robotics*, vol. 2017, pp. 1–7, 2017, doi: 10.1155/2017/2061827.
- [127] F. A. Gers, J. Schmidhuber, and F. Cummins, ‘Learning to forget: continual prediction with LSTM’, in *9th International Conference on Artificial Neural Networks: ICANN ’99*, 1999, vol. 1999, pp. 850–855, doi: 10.1049/cp:19991218.
- [128] M. Turkoglu, D. Hanbay, and A. Sengur, ‘Multi-model LSTM-based convolutional neural networks for detection of apple diseases and pests’, *Journal of Ambient Intelligence and Humanized Computing*, Nov. 2019, doi: 10.1007/s12652-019-01591-w.
- [129] B. Basnet, ‘LSTM Optimizer Choice ?’ <https://deepdatascience.wordpress.com/2016/11/18/which-lstm-optimizer-to-use/>, erişim: 01.01.2022, 15:41.
- [130] D. P. Kingma and J. Ba, ‘Adam: A Method for Stochastic Optimization’, Dec. 2014, [Online]. Available: <http://arxiv.org/abs/1412.6980>.
- [131] The MathWorks, ‘trainingOptions’, [Online]. Available: <https://www.mathworks.com/help/deeplearning/ref/trainingoptions.html>, erişim: 01.02.2022, 16:24.
- [132] B. Bilen, ‘Karışıklık Matrisi (Confusion Matrix)’. <https://burhanbilen.medium.com/karışıklık-matrisi-confusion-matrix-990dfc718653>, erişim: 12.01.2022, 21:14.
- [133] Jason Brownlee, ‘Failure of Classification Accuracy for Imbalanced Class Distributions’. <https://machinelearningmastery.com/failure-of-accuracy-for-imbalanced-class-distributions/#:~:text=Classification accuracy is a metric,used for evaluating classifier models>, erişim:22.01.2022, 22:30.

- [134] E. Dougherty, J. Hua, and C. Sima, ‘Performance of Feature Selection Methods’, *Current Genomics*, vol. 10, no. 6, pp. 365–374, Sep. 2009, doi: 10.2174/138920209789177629.
- [135] C. Sima and E. R. Dougherty, ‘The peaking phenomenon in the presence of feature-selection’, *Pattern Recognition Letters*, vol. 29, no. 11, pp. 1667–1674, Aug. 2008, doi: 10.1016/j.patrec.2008.04.010.
- [136] R. Ghanbari and K. Borna, ‘Multivariate Time-Series Prediction Using LSTM Neural Networks’, in *2021 26th International Computer Conference, Computer Society of Iran (CSICC)*, Mar. 2021, pp. 1–5, doi: 10.1109/CSICC52343.2021.9420543.
- [137] J. Wang, W. Wang, S. Wei, Y. Zeng, and F. Luo, ‘Time Series Sequences Classification with Inception and LSTM Module’, in *2019 IEEE International Conference on Integrated Circuits, Technologies and Applications (ICTA)*, Nov. 2019, pp. 51–55, doi: 10.1109/ICTA48799.2019.9012862.
- [138] L. Hosseini and R. Shaghghi Kandovan, ‘Hyperspectral Image Classification Based on Hierarchical SVM Algorithm for Improving Overall Accuracy’, *Advances in Remote Sensing*, vol. 06, no. 01, pp. 66–75, 2017, doi: 10.4236/ars.2017.61005.