



T.C.

ALTINBAŞ ÜNİVERSİTESİ

Lisansüstü Eğitim Enstitüsü

Elektrik ve Bilgisayar Mühendisliği Ana Bilim Dalı

**RESIDUAL NETWORK TABANLI UYGULAMA
İLE GÖZ BAKTERİLERİN SINIFLANDIRILMASI**

Betül ÖZÇINAR

Yüksek Lisans Tezi

Danışman

Doç. Dr. Sefer KURNAZ

İstanbul, 2023

RESIDUAL NETWORK TABANLI UYGULAMA İLE GÖZ BAKTERİLERİN SINIFLANDIRILMASI

Betül ÖZÇINAR

Elektrik ve Bilgisayar Mühendisliği Ana Bilim Dalı

Yüksek Lisans Tezi

ALTINBAŞ ÜNİVERSİTESİ

2023

Betül ÖZÇINAR tarafından hazırlanmış ve 19/12/2023 tarihinde sunulmuş RESIDUAL NETWORK TABANLI UYGULAMA İLE GÖZ BAKTERİLERİN SINIFLANDIRILMASI başlıklı tez Elektrik ve Bilgisayar Mühendisliği Ana Bilim Dalında Yüksek Lisans Tezi olarak **oy birliği** ile kabul edilmiştir.

Doç. Dr. Sefer KURNAZ

Danışman

Tez Savunma Sınavı Jüri Üyeleri:

Doç. Dr. Sefer KURNAZ

Bilgisayar Mühendisliği,
Altınbaş Üniversitesi

Dr. Öğr. Üyesi Abdullahi
ABDUIBRAHİM

Bilgisayar Mühendisliği,
Altınbaş Üniversitesi

Dr. Öğr. Üyesi Zeynep ALTAN

Yazılım Mühendisliği,
Beykent Üniversitesi

Bu tezin Yüksek Lisans tezi olarak bütün şartları sağladığımı beyan ederim.

Bu tezdeki tüm bilgilerin akademik kurallara ve etik davranışlara uygun olarak edinildiğini ve sunulduğunu beyan ederim. Ayrıca bu kuralların ve davranışların gerektirdiği şekilde, bu çalışmada orijinal olmayan tüm materyalleri ve sonuçları tamamen alıntı yaptığımı ve referans gösterdiğimi de beyan ederim.

Betül ÖZÇINAR

İmzası

İTHAF

Bu tez, sadece bir akademik çalışma deęil, aynı zamanda destekleri, cesaret verici sözleri ve inançlarıyla bu yolda yanımda olan anneme babama kardeşlerime ve dostlarıma ithaf edilir.



ÖNSÖZ

İnsanlık tarihinde gözle görülür bir hızla ilerleyen dijital teknolojiler, bilgi işlem alanında devrim niteliğindeki gelişmeleri beraberinde getirmektedir. Bu gelişmelerin önemli bir kolu olan görüntü işleme, bilgisayarla görmenin temel taşıdır. Bu bağlamda, bu tez, görüntü işleminin önemli bir kilometre taşı olan ResNet yapılarının incelenmesi ve analiziyle ilgilenmektedir. Geliştirilen bu yapılar, derin öğrenme alanında çığır açmış, bilgisayarla görme uygulamalarında önemli bir dönüm noktası olmuştur. Bu tez, ResNet'in temel prensiplerini ve uygulamalarını ayrıntılı bir şekilde ele alarak, bilimsel bir perspektif sunmayı amaçlamaktadır. Ayrıca, bu çalışma, bu alandaki gelecek çalışmalara ilham kaynağı olması amacıyla yazılmıştır. Tez boyunca ortaya konulan analizlerin ve sonuçların, görüntü işleme alanındaki bilimsel çıkarımlara ve uygulamalara katkı sağlaması temennisiyle...

ÖZET

RESIDUAL NETWORK TABANLI UYGULAMA İLE GÖZ BAKTERİLERİN SINIFLANDIRILMASI

ÖZÇINAR, Betül

Yüksek Lisans, Elektrik ve Bilgisayar Mühendisliği Anabilim Dalı, Altınbaş Üniversitesi

Danışman: Doç.Dr. Sefer KURNAZ

Tarih: Aralık /2023

Sayfa: 97

Günümüzde görüntü tanıma teknolojileri, her geçen gün daha da popülerleşerek gelişmektedir. Görüntü tanıma teknolojileri; endüstriyel otomasyonlar, Medikal ve sağlık, astronomi, otomatik sürüş ve uçuş sistemleri gibi birçok alanda kullanılmaktadır. Yüz tanıma sistemleri, elektronik görüntü belirleme lensleri, optik okuyucular; görüntü işleme konusunda son yıllarda hem akademik alanda hem de sosyal alanlarda yaygın olarak kullanılan ve araştırılan insan-bilgisayar etkileşimi konuları arasında yer almaktadır. Bu çalışmada yapay sinir ağlarından biri olan ResNet sistemi, deep neural network modellerindeki gradiyent sıkışması problemini çözmek için skip connection'ları kullanır. Bu skip connection'lar, modelin önceki katmanlarından gelen bilginin direkt olarak sonraki katmanlara aktarılmasını sağlar ve böylece gradiyent sıkışmasını önlenabilir ve daha derin network'lerin eğitilebilmesi mümkün olabilir. Çalışmamızda Resnet çoklu görüntü alma mimari yapısını kullanarak gözlerde bulunan bakteri türlerini ayırt edip sınıflandırmaya çalıştık. Sistemde kullandığımız mimari sayesinde, bakteri görüntülerinin daha derin ve daha kompleks modeller oluşturmalarının yanı sıra daha yüksek performanslı görüntü elde etmemizi mümkün kılmaktadır. ResNet, ImageNet gibi büyük veri kümelerinde yüksek performans göstermiş olup güncel deep learning uygulamalarından birçoklarında kullanılan bir sistem olmuştur.

Anahtar Kelimeler: Yapay Zeka, Derin Öğrenme, Makine öğrenmesi, ImageNet, ResNet.

ABSTRACT

CLASSIFICATION OF EYE BACTERIA WITH RESIDUAL NETWORK BASED APPLICATION

ÖZÇINAR, Betül

M.Sc., Department of Electrical and Computer Engineering, Altınbaş University,

Supervisor: Assoc. Prof. Dr. Sefer KURNAZ

Date: December / 2023

Pages: 96

Today, image recognition technologies continue to be popularised day by day. They are used in many fields such as industrial automation, medical and health, astronomy, automatic driving and flight systems, face recognition systems, electronic image detection lenses, optical readers are among the human-computer interaction topics that have been widely used and researched in both academic and social fields in recent years in image processing. In this study, the ResNet system, one of the artificial neural networks, uses skip connections to solve the gradient compression problem in deep neural network models. These skip connections allow the information from the previous layers of the model to be transferred directly to the next layers, so that gradient compression can be avoided and deeper networks can be trained. In our study, we tried to distinguish and classify the types of bacteria found in the eyes by using the ResNet multi-image retrieval architecture. Thanks to the architecture we use in the system, it enables us to obtain higher performance images as well as deeper and more complex models of bacteria images. ResNet has shown high performance on large datasets such as ImageNet and has become a system used in many current deep learning models.

Keywords: Artificial Intelligence, Machine Learning, Deep Learning, ImageNet, ResNet.

İÇİNDEKİLER

Sayfa

ÖZET	vii
ABSTRACT	viii
TABLO LİSTESİ.....	xi
ŞEKİL LİSTESİ.....	xii
KISALTMALAR	xiv
SEMBOL LİSTESİ.....	xv
1. GİRİŞ.....	1
1.1 AMAÇ.....	1
2. YAPAY ZEKA (ARTIFICIAL INTELLIGENCE)	3
2.1 MAKİNE ÖĞRENİMİ (MACHINE LEARNING).....	12
2.1.1 Öğrenim İlişkileri.....	16
2.1.2 Sınıflandırma	17
2.2 DERİN ÖĞRENME (DEEP LEARNING)	21
2.2.1 Derin Öğrenmenin Çalışma Stili.....	29
2.2.2 CNN (Evrişimli Sinir Ağları).....	32
2.2.2.1 Evrişim katmanı (Convolution layer)	35
2.3 IMAGE PROCESSING.....	58
2.3.1 Görüntü İşlemede Nokta Operatörleri	66
2.3.2 Piksel İşleme	68
3. BULGULAR	70
3.1 SİSTEMDE KULLANDIĞIMIZ RESNET 50 VERİ MODELİ YAZILIM MİMARİSİ	70
3.1.1 Sistem Alt ve Üst Kütüphaneleri Tanımlamaları	70

3.1.2 DataSet Tanıtma.....	71
3.1.3 View Modelimiz	72
3.1.4 Resnet Yordamları	73
3.1.5 Görüntü İşleme İçin Kullanılan Parametreler	74
3.1.6 Listdizi Class Yapısı.....	74
3.1.7 Keras Modeli Optimizasyonu	75
3.1.8 Classname	76
3.1.9 Resnet Model Oluşturma	76
3.1.10 Resnet Modelleme	77
3.1.11 Görüntü Yakalama.....	77
4. GEREKSİNİMLER	78
5. ARAŞTIRMAMIZDA KULLANDIĞIMIZ YÖNTEMLER.....	79
6. TARTIŞMA VE SONUÇLAR	80
REFERANSLAR	83

TABLO LİSTESİ

Sayfa

Tablo 2.1: Karşılaştırmalı AlexNet ve ZFNet Tablosu	40
Tablo 2.2: ImageNet'te, Konvolüsyonel Sinir Ağları (AlexNet, Inception, VGG-16, ResNet gibi) Görüntü Tanımanın Yanı Sıra Diğer Birçok Alanda da Devrim Niteliğinde Başarılar Elde Etme Potansiyeli Olduğunu Gösteriyor.	47
Tablo 2.3: Çeşitli ILSVRC Yarışmasının En İyi Performans Gösteren Girişlerindeki Katman Sayısı.....	55
Tablo 3.1: Kütüphane Parametreleri	70
Tablo 3.2: Veritabanı Yordamları.....	71
Tablo 3.3: Görüntü Parametreleri	72
Tablo 3.4: ResNet Element Yordamları	73

ŞEKİL LİSTESİ

Sayfa

Şekil 2.1: Yapay Zekanın Özellikleri ile İlgili Görsel	4
Şekil 2.2: Örnek Bir Sinir Hücresi Resmi	5
Şekil 2.3: Modelin Eğitim Sürecinde Kullanılan Örnek Veri Seti.	18
Şekil 2.4: Makine Öğrenmesi ve Derin Öğrenme Karşılaştırılması.....	23
Şekil 2.5: Yapay Zeka ve Alt Kümelerini Oluşturan Yapısalı.	24
Şekil 2.6: Öğrenme Modellerinin Üst Düzey Yapısı	26
Şekil 2.7: Google Books Verilerine Göre, Yapay Sinir Ağları Araştırmalarının Tarihsel Olarak Üç Ana Dalgada Ölçülen Anahtar Terimler 'Sibernetik' ve 'Bağlantıcılık' veya 'Sinir Ağları' dır. Şekil ise Üç Tarihsel Dalgasından İkisini Göstermektedir.....	28
Şekil 2.8: Derin Ağların Beş Yıllık Verileri.....	29
Şekil 2.9: Derin Öğrenme Nöral Girdiler	30
Şekil 2.10: Support Vektör Machine.....	32
Şekil 2.11: Evrişimli Sinir Ağları Mimari Yapısı	34
Şekil 2.12: Üst Düzey Genel CNN Mimarisi	34
Şekil 2.13: AlexNet Mimarisi.....	37
Şekil 2.14: VGG'de Kullanılan Konfigürasyonlar.....	43
Şekil 2.15: GoogLeNet'in Başlangıç Modülü.....	45
Şekil 2.16: 2010'dan beri ImageNet Üzerinde ILSRV İlk 5 Hatası Gösterilmektedir.	47
Şekil 2.17: Artık Modülü ve ResNet'in İlk Birkaç Katmanı.....	49
Şekil 2.18: Kaybolan ve Patlayan Gradyan Problemi Figürü.....	57
Şekil 2.19: Nesneleri Görüntülemek Amacıyla Radyasyon ve Madde Arasındaki Etkileşimin Şematik Diyagramı.	61
Şekil 2.20: Bazı Yaygın Görüntü İşleme İşlemleri: (a) Orijinal Görüntü; (b) Artırılmış Kontrast; (c) Görüntüdeki Değişim Renk Tonu; (d) "Posterleştirilmiş" (Nicelleştirilmiş	

Renkler); (e) Bulanıklaştırılmış; (f) Döndürülmüş İşlemler.....	62
Şekil 2.21: Görüntüleme için Bir Sistem Yaklaşımı.....	64
Şekil 2.22: Farklı Piksel Bilgi Türleri	65
Şekil 2.23: Yukarıda, (a)Orijinal Görüntünün Yanı Sıra, (b) Artırılmış parlaklığı, (c) Artırılmış Kontrastı, (d) Gamasını ve (e)Histogramını, (f) Kısmi Histogramını Değiştiren Bazı Yerel Görüntü İşleme İşlemleri Gösterilmektedir.	67
Şekil 2.24: Görüntü Verilerinin Görselleştirilmesi: (a) Orijinal Görüntü; (b) Bir Görüntü İnceleme Aracı Kullanılarak Kırpılmış Kısım ve Tarama Çizgisi Çizimi; (c) Sayılar Izgarası; (d) Yüzey Çizimi. Şekil (c)-(d) İçin Görüntü Önce Gri Tonlamaya Dönüştürülmüştür.	68
Şekil 3.1: Dataset İmport Ettiğimiz Verisetimiz.....	74
Şekil 3.2: Data Veri Setimizden Çekilen Görüntüleri ve Oluşturduğumuz Sınıfları Çekmek için Kullandığımız Yordamlarımız.....	74
Şekil 3.3: Bakteri Türünü Çekmek İçin Kullandığımız Metod	75
Şekil 3.4: Oluşturduğumuz Keras Modeli Metodu.....	75
Şekil 3.5: Bakteri Türlerini Çekmek İçin Kullandığımız Metod.....	76
Şekil 3.6: Resnet-50 Metodu	76
Şekil 3.7: Resnet Kod Blokları	77
Şekil 3.8: Görüntü Yakalama Kodları.....	77
Şekil 6.1: Bakteri Sınıflandırmamızdaki Veri Setinden Elde Ettiğimiz Görüntümüz.	81
Şekil 6.2: Resnet-50 için Sistemden Alınan Verideğerleri	82

KISALTMALAR

AI	:	Artificial Intelligence
CNN	:	Convolutional Neural Network
ANN	:	Artificial Neural Network
RCNN	:	Region Based Neural Network
SVM	:	Support Vector Machine
GPU	:	Graphics Processing Unit
ReLU	:	Rectified Linear Unit
ILSVRC	:	ImageNet Large Scale Visual Recognition Challenge
GTX	:	Giga Texel Shader eXtreme
VGG	:	Visual Geometry Group
LSTM	:	Long Short-Term Memory

SEMBOL LİSTESİ

μ	:	Elektronun Mobilitesi
λ	:	Dalgaboyu
ω	:	Açısal Frekans
Σ	:	Toplam Sembolü
θ	:	Tetha



1. GİRİŞ

1.1 AMAÇ

Günümüzde Yapay Zekanın makine veya bilgisayar sistemlerinin insan benzeri bir düşünce ve davranış göstermesini amaçlayan alanlarda kullanımı gittikçe yaygınlaşmış olup bu sistem ve özellikler sayesinde insanların kullanımını kolaylaştıracak gereçler ve yöntemler hızlı bir şekilde inşa edilmeye başlanmıştır. AI teknolojileri, insanlar gibi problemleri çözebilme, öğrenme ve karar verme yeteneğine sahip olma eğiliminde olmakla birlikte yapay zekanın popülerleşerek insan hayatını olumlu yönde etkilemesi ve sürekli yenilenerek gelişmesi olumlu bir gelişme olarak kaydedilmektedir. Yapay zeka içindeki en önemli alt dallarından biri olan derin öğrenme sistemleriyle küçük veya büyük miktarda veri ve çok katmanlı sinir ağları kullanımıyla makineler öğrenme kapasitesine destek vermektedir.

Bu çalışmamızda, Residual Network (Residüel Ağ) yapısını kullanarak derin öğrenme alanındaki bir yapı olan Residüel ağlarla birçok katman arasındaki verilerin zayıflamasını önlemek ve daha doğru tahminler yapılmasını sağlamak amaçlanmıştır. Bu ağlar; görüntü tanıma, ses tanıma ve diğer uygulamalarda oldukça etkili bir yöntemdir.

Bu sistemi kullanmamızdaki amacımız;

- Daha Derin Ağ: ResNet-50, geleneksel ağlara kıyasla çok daha derin ağ mimarilerine olanak tanır ve daha iyi performans sağlar.
- Geliştirilmiş eğitim sayesinde ResNet-50, derin ağların eğitimini kolaylaştırmak için artık bağlantıları kullanarak daha iyi yakınsama ve daha hızlı eğitim süreleri sağlar.
- Geliştirilmiş genelleme yaparak ResNet-50, geleneksel derin ağlara kıyasla gelişmiş genelleme performansı göstermiştir, bu da görünmeyen veriler üzerinde iyi performans gösterebileceği anlamına gelir.
- Önceden Eğitilmiş Modeller ile ResNet-50, ImageNet gibi büyük veri kümeleri üzerinde önceden eğitilmiştir; bu da ağırlıkların diğer görevlerde transfer öğrenimi için bir başlangıç noktası olarak kullanılmasını mümkün kılarak iyi bir model eğitmek için gereken veri miktarını azaltır.

Özetle, ResNet-50'nin daha derin ağları eğitme yeteneği, gelişmiş eğitim ve genelleme ve önceden eğitilmiş modellerin kullanılabilirliği, onu bilgisayarla görme görevleri için güçlü

bir aday haline getirmektedir. Bu üç teknoloji birlikte verileri daha iyi analiz etme, problemleri çözme ve insanlar gibi davranma yeteneği kazanma amacını desteklemektedir. Yapay zeka, derin öğrenme ve residüel ağ teknolojileri, gelecekte daha da gelişecek ve daha fazla alanda kullanılacaktır. Aynı zamanda bu teknolojilerin uygulanması, ekonominin büyümesine ve endüstri alanındaki gelişmeye de katkıda bulunacaktır.



2. YAPAY ZEKA (ARTIFICIAL INTELLIGENCE)

Yapay Zekâ, veri ve algoritmalar kullanarak kendini öğrenen ve uyarlayan sistemler oluşturmak için kullanılan bir teknolojidir. Makine öğrenmesi, yapay sinir ağları ve diğer yapay zekâ teknolojileri, verileri analiz etme ve doğru kararlar verme yeteneği kazandırarak insanların yapabileceklerinden daha hızlı ve hassas sonuçlar elde etmelerine olanak tanır. Yapay zeka; finans, sağlık, eğitim ve otomotiv gibi alanlarda kullanılabilir ve iş verimliliğini artırabilir, kaliteyi arttırabilir ve müşteri deneyimini iyileştirebilir. Ancak yapay zekâ aynı zamanda sosyal ve etik sorunları da ortaya çıkarabilir ve güvenliği tehdit edebilir. Bu nedenle, yapay zekâ kullanımının yanı sıra bu teknolojinin etik ve güvenliği de dikkate alınmalıdır.

Yapay Zekâ (AI) ve Görüntü İşleme, günümüz dünyasında çok sayıda avantaja sahiptir. Geliştirilmiş otomasyon: Yapay zeka, tekrar eden görevleri otomatikleştirerek verimliliği artırabilir ve hataları azaltabilir. Yapay zeka destekli sistemler, büyük miktarda veriyi analiz ederek içgörü elde eder ve bu içgörülerini bilinçli kararlar almak için kullanır. Artan doğruluk oranıyla görüntü işleme tekniklerinde, görüntülerin kalitesini ve doğruluğunu artırarak daha fazla analiz için daha uygun hale getirebilen sistemlerdir. Daha iyi müşteri deneyimi sunan yapay zeka destekli sistemler, önerileri kişiselleştirebilir ve çeşitli sektörlerde müşteri hizmetlerini iyileştirebilir. Geliştirilmiş derin öğrenme yapan Artık Ağlar ile derin ağlardaki kaybolan gradyanlar sorununu ele alabilir ve çeşitli görevlerde doğruluğu artırabilir. Robotikte ilerlemeler de ise Yapay Zeka ve Görüntü İşleme, otonom robotların ve insansız hava araçlarının geliştirilmesinde temel bileşenlerdir.

İnsanlardan daha zeki ve daha akıllı özelliklere sahip makinelerden oluşan Yapay Zeka, çoğu zaman bilgisayarlar tarafından kontrol edilir. Bu, bilim kurgu filmlerinde ve dizilerinde sıklıkla görülen, insan zekası ile eşit veya üstün yapay zeka sistemlerinin yaratılmasına olanak tanır. Yapay zeka araştırmalarında çok kullanılan iki anahtar kelime vardır, bunlar; duyarlılık ve zekâdır [1].

Duyarlılık, çevremizdeki dünyayı deneyimleme biçimimizdir. Bu, fiziksel duyularımız aracılığıyla dünyayı algılamaktan, duygularımızı ve düşüncelerimizi hissetmeye kadar her şeyi içerir. Duyarlılık, bizi canlı ve bağlantılı kılan şeydir. Bilgisayarlar ya da robotlar, bilinçli olmak için gerekli olan duyuları taklit edebilirler. Bir bilgisayar, bir kamera veya tarayıcı aracılığıyla çevresini görebilir. Bir mikrofon aracılığıyla sesleri duyabilir. Hatta, bir kişinin konuştuğu kelimeleri tanıyacak şekilde programlanabilir. Bu, bilgisayarların ve

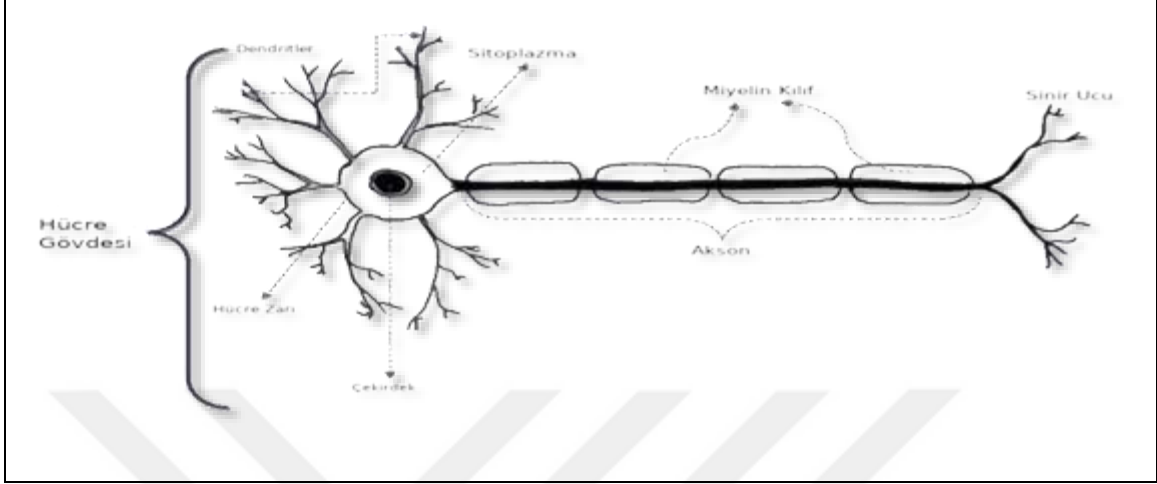
robotların, insan deneyimine benzer bir şekilde çevrelerini algılayabildiklerini gösterir. Örneğin, bilgisayarların duyularını geliştirmek için çalışan bilim insanları, onları insan gibi algılama yeteneğine sahip hale getirmeye çalışıyor. Koku alma, tat alma ve dokunma gibi duyuları taklit eden cihazlar geliştiriyorlar. Bu cihazlar hala geliştirilmekte ve mükemmelleştirilmekte olsa da, bilgisayarların çevrelerindeki dünyayı daha iyi anlamalarına yardımcı oluyorlar. Sonuç olarak, bazı bilgisayarlar ve robotlar dünyayı algılayabilmektedir. Duyularının onlara ne söylediğini anlayıp anlayamadıkları ya da takdir edip edemedikleri başka bir şeydir. Bazı bilim adamları, duyarlılığın sadece bir duygudan ibaret olmadığını daha fazlası olduğunu düşünüyorlar ve o da dünyayı hissedebilmek [1].



Şekil 2.1: Yapay Zekanın Özellikleri ile İlgili Görsel.

Yapay zeka araştırmalarında, zeka kavramının ne anlama geldiği konusunda bir anlaşmazlık vardır. Bazı bilim insanları, zekanın karmaşık problemleri çözme, yeni şeyler öğrenme ve deneyimlerden ders çıkarma yeteneği olarak tanımlanabileceğini savunurken, diğerleri zekanın daha çok akıl yürütme ve problem çözme yeteneğiyle ilgili olduğunu düşünür. Zeka, " Bir problemi çözmek için gerekli bilgiyi, deneyimleri ve yaratıcılığı bir araya getirme yeteneğidir. " [2] şeklinde tanımlamaktadırlar. Zeka tanımının bir kısmı öğrenme yeteneğiyle ilgilidir ve öğrenme olayı beynimize bir şey veya konu hakkında yeni bilgiler eklediğimiz anlamına gelmektedir. Öğrenciler küçükken toplama çıkarma çarpma vs. konularını

öğrenirken 3 toplam 2'nin 5 olduğunu veya 4 çarpı 4'ün 16 olduğunu ve dört mevsimi, ayların ne olduğunu öğrenmektedirler. Onlar bu gerçekleri önceden bilmiyorlardı ve bu öğrenmeleri onlara bilgi kattı [3].



Şekil 2.2: Örnek bir Sinir Hücresi Resmi.

Zihin kısmen mantıklara göre işler ve bu kuralların bazılarını taklit eden fiziksel sistemler inşa etmek başka bir şeydir; zihnin kendisi de böyle bir fiziksel sistemdir. René Descartes, zihin ve bedenin farklı tözlerden oluştuğu ve bu ayrımın birçok sorunu beraberinde getirdiği fikrini ilk kez dile getiren filozoftur. Descartes, bugün rasyonalizm olarak bilinen ve üyeleri arasında Aristoteles ile Leibniz'i de sayan bir felsefe akımının temsilcisi olarak, dünyayı anlamada akıl yürütmenin gücünün güçlü bir taraftarıydı. Bununla birlikte, kendisi aynı zamanda düalizmin savunucularından biri olarak bilinirdi. Descartes, insan zihninin (ya da ruhunun) bir "düşünen şey" olduğunu ve bu nedenle fiziksel dünyanın yasalarından muaf olduğunu savunmuştur [4]. Descartes, zihin ve maddenin iki farklı töz olduğunu savunarak, zihin-beden probleminin temellerini atmıştır. Ancak, hayvanlara makine muamelesi yapabileceğini öne sürerek, bu ayrımı hayvanlara da uygulamıştır. Bu, Descartes'ın zihin-beden ayrımının, hayvanlara yönelik yaklaşımıyla tutarsız olduğunu göstermektedir. Materyalistler, zihnin beynin fiziksel bir ürünü olduğunu savunurlar. Özgür irade ise, bu fiziksel ürünün, sahip olduğumuz seçenekleri nasıl algıladığımıza bağlıdır.

Bilim İnsanları yapay zekânın bazı temel fikirlerini ortaya koymuşlardır fakat resmi olarak bilime sıçrama, üç temel alanda matematiksel bir formalizasyon seviyesi gerektirmiştir. Bunlar, mantık, hesaplama ve olasılıktır. Biçimsel mantık fikri antik Yunan filozoflarına kadar uzanmaktadır ancak matematiksel gelişimi gerçek anlamda önermeler mantığı ya da

Boole mantığının detaylarını çözen George Boole'un (1815-1864) çalışmalarıyla başlamıştır (Boole, 1847). 1879'da Gottlob Frege (1848-1925) Boole'un mantığını nesneleri ve ilişkileri içerecek şekilde genişleterek günümüzde kullanılan birinci dereceden mantığı oluşturdu. Alfred Tarski, mantıktaki nesnelerin gerçek dünyadaki karşılıklarına bir köprü inşa eden bir referans teorisi ortaya koymuştur. Bu köprü, mantıktaki nesnelerin gerçek dünyadaki karşılıklarıyla doğru bir şekilde ilişkilendirilmesini sağlar [5] [6] [7].

Mantık ve hesaplama yoluyla yapılabileceklerin sınırlarını belirlemek için bir sonraki adım, bu sınırların belirsizliğini kabul etmektir. Kayda değer ilk algoritma, Öklid'in en büyük ortak bölen hesaplama algoritması olarak kabul edilir. Algoritma kelimesi (ve algoritmalar üzerinde çalışma fikri), yazılarıyla Arap rakamları ve cebiri, Avrupa'nın matematiksel gelişiminde bir dönüm noktası olmuştur. Bu dönüm noktasının mimarı, 9. yüzyıl Fars matematikçisi El-Harezmi'dir. Boole ve diğerleri mantıksal tümdengelim için algoritmaları kendi aralarında münazara ettiler ve 19. yüzyılın sonlarında, matematikçiler, matematiksel akıl yürütmenin temellerini mantıksal olarak sağlamlaştırmak için, matematiksel akıl yürütmenin mantıksal olarak doğru olan bir öncül kümesi verildiğinde, sonuç olarak doğru olan bir sonuca varmanın mümkün olduğunu göstermeye çalışıyorlardı. 1930'da Kurt Gödel (1906-1978), Frege ve Russell'in birinci dereceden mantığında herhangi bir doğru ifadeyi kanıtlamak için etkili bir prosedür bulduğunu, ancak birinci dereceden mantığın doğal sayıları karakterize etmek için gereken matematiksel tümevarım ilkesini yakalayamadığını göstermiştir. 1931'de Gödel, tümdengelim üzerinde belli ölçülere göre sınırlar olduğunu ve "Eksiklik Teoremi" olarak adlandırılan Peano aritmetiğinde (doğal sayıların temel teorisi), teori içinde ispatı olmayan yani tümdengelim yapılamayan doğru ifadeler olduğunu gösterdi [8]. Bu temel sonuç, tam sayılar üzerindeki bazı fonksiyonların, hesaplama makineleri tarafından gerçekleştirilemeyecek kadar karmaşık olduğunu gösterdi. Bu durum ise Turing'i, hesaplama makinelerinin yapabileceklerinin sınırlarını belirlemek için çalışmaya yöneltti. Bir hesaplama veya etkin prosedür kavramına gerçekten de resmi bir tanım verilemediğinden, Turing makinesinin (Turing, 1936) aritmetik olarak hesaplanabilir herhangi bir problemi hesaplayabildiğini gösteren Church-Turing tezi; bunun için yeterli bir tanım olarak kabul edilmiştir [9]. Turing, hesaplama makinelerinin yapabileceklerinin bir "tavan" olduğunu gösterdi. Bu tavan, hiçbir Turing makinesinin hesaplayamayacağı bazı işlevleri içeriyor. Örneğin, Turing'in teoremini, bir dağa tırmanış olarak düşünebiliriz. Bu yaklaşım, hesaplama makinelerinin yapabilecekleri ve yapamayacakları şeyleri, bir dağa

tırmanabilecekleri ve tırmanamadıkları şeylerle karşılaştırır. Şöyle ki; hiçbir makine genel olarak belirli bir programın belirli bir girdi üzerinde bir cevap verip vermeyeceğini veya sonsuza kadar çalışıp çalışmayacağını söyleyememiştir.

Bir yapay zekanın başarılı olabilmesi için iki şeye ihtiyacımız vardır. Bunlar, zeka ve nesnedir. Bizim için bu işi yapabilecek ve yapay zekâ için tercih edilen nesnemiz bilgisayarlarımızdır. İkinci Dünya Savaşı'nın zorlu koşulları altında, bilim insanları, mühendisler ve askeri yetkililer, savaşın gidişatını değiştirecek yeni teknolojiler geliştirmek için bir araya geldiler. Bu çabaların bir sonucu olarak, İngiltere, Almanya ve Amerika Birleşik Devletleri'nde bağımsız olarak ve hemen hemen aynı anda, modern dijital elektronik bilgisayarlar icat edildi.

Elektromekaniksel olarak çalışabilen ilk bilgisayar, 1940 yılında Alan Turing tarafından yapılan ve tek bir amaç olan Alman mesajlarının deşifresi için kullanılan Heath Robinson'du. Aynı araştırma ekibi, 1943 yılında vakum tüplerine dayalı güçlü ve özel amaçlı bir bilgisayar olan Colossus'u geliştirdi. Dünya genelindeki ilk operasyonel ve programlanabilir bilgisayar ise 1941 yılında Almanya'da Konrad Zuse tarafından icat edilen Z-3'tür [8]. Zuse, bilgisayar biliminin gelişiminde bir dönüm noktasıydı. O, kayan noktalı sayıları ve ilk yüksek seviyeli programlama dilini icat ederek, günümüz bilgisayarlarının geliştirilmesini mümkün kıldı. Bu icatları, günümüz bilgisayarlarının temelini oluşturdu ve dünyanın şekillenmesinde önemli bir rol oynadı. Atanasoff'un araştırması diğerlerine göre çok az destek görmüştür ve maalesef daha az tanınmıştır. Modern bilgisayarların en etkili öncülerinden biri olan John Mauchly ve John Eckert'in de dâhil olduğu bir ekip tarafından Pensilvanya Üniversitesi'nde gizli bir askeri projenin parçası olarak ENIAC geliştirilmiştir [10].

O zamandan beri her yeni nesil bilgisayar, donanımın hızını ve kapasitesini artırdı ve fiyatı düşürdü. Yaklaşık 2005 yılına kadar performansları her 18 ayda bir yaklaşık iki katına çıkardı, ancak güç tüketimi sorunları üreticilerin saat hızları yerine CPU çekirdeklerinin sayısını artırmaya başlamasına neden oldu. Şu anki beklentiler, gelecekteki güç artışlarının, beynin özellikleriyle garip bir yakınlaşma olan devasa paralellikten kaynaklanacağı şeklindedir. Hesaplama, insanlığın varoluşundan beri var olan bir ihtiyaçtır. Bu ihtiyacı karşılamak için, tarih boyunca çeşitli hesaplama cihazları icat edilmiştir. Geçmiş 17. yüzyıla kadar uzanan en eski otomatik makinelerden olan ilk programlanabilir makine 1805 yılında Joseph Marie Jacquard (1752-1834) tarafından icat edilen dokuma tezgahıydı. Jacquard, İcat etmiş olduğu

makine dokuma kartlarının zimbalarını kullanarak desen oluşturmak için talimatlar kaydetmiştir [11].

19. yüzyılın ortalarında, Charles Babbage, modern bilgisayarların temellerini attı. Fark Motoru ve Analitik Motoru, günümüz bilgisayarlarının öncülleri idi. Fark Motoru, mühendislik ve bilimsel projeler için karmaşık hesaplamaları otomatikleştirmeyi amaçlayan bir mekanik cihazdı. Analitik Motor ise, programlanabilir bir dijital bilgisayarın ilk tasarımıydı. Charles Babbage'nin Analitik Motoru, modern bilgisayarların gelişiminde bir dönüm noktasıydı. Bu makine, adreslenebilir bellek, depolanan programlar ve koşullu atlamalar gibi modern bilgisayarların temel özelliklerini içeriyordu. Analitik Motor, günümüz bilgisayarlarının yapabildiği her şeyi yapabilen ilk yapıtı ve modern bilgisayarların geliştirilmesini mümkün kıldı. [11].

Yapay zekâ, bilgisayar biliminin hem donanım hem de yazılım yönlerine borçludur. Donanım, yapay zekâ sistemlerinin çalışması için gerekli olan temel altyapıyı sağlarken, yazılım ise bu sistemlerin nasıl çalışacağını ve ne yapacağını belirler. İşletim sistemleri, programlama dilleri ve modern programlar (ve bunlar üzerine yazılmış makaleler) yazmak için gereken araçları sağlamıştır. Yapay zeka, bilgisayar biliminin gelişiminde bir katalizör görevi gördü. Bu zeka sayesinde, bilgisayar bilimi, daha verimli, daha kullanıcı dostu ve daha güçlü hale geldi. Yapay zeka, bilgisayar bilimine bir tılsım gibi etki etti. Dolayısıyla bilgisayar bilimi, bir büyücü gibi, daha önce mümkün olmayan şeyleri yapmaya başladı.

Modern dil bilimi ve yapay zekâ, aynı zamanda doğan ikiz kardeşler gibidir. Bu iki kardeş, zamanla büyüdükçe ve geliştikçe, hesaplamalı dil bilimi veya doğal dil işleme adı verilen ortak bir alanda buluştular. Çok geçmeden, dili anlama sorununun 1957'de olduğundan çok daha karmaşık olduğu ortaya çıkmıştır. Ancak 1957'de dilin oldukça karmaşık bir şey olduğu ortaya çıktı. Dili anlamak sadece cümle yapısını anlamayı değil, aynı zamanda konu ve bağlamı anlamayı da gerekli görüyordu. Bu gerçek, bir zamanlar gün gibi açıkken, 1960'lara kadar çoğu insan tarafından anlaşılmamıştı. Bu, insan düşüncesinin bir paradoksudur: Bazen en basit gerçekler, en zor anlaşılanlardır. Bilgi temsili (bilginin bilgisayarların muhakeme edebileceği bir biçimde nasıl temsil edileceğinin araştırılması) üzerine yapılan ilk çalışmaların çoğu dile bağlıydı ve dilbilimsel araştırmalar tarafından bilgilendirildi, bu da dilin felsefi analizi üzerine onlarca yıllık çalışmalara bağlıydı [12].

Yapay Zeka (AI) olarak günümüzde genellikle bilinen ilk çalışma, Yapay Zeka (AI) alanında

genellikle bilinen ilk önemli çalışma, 1943 yılında Warren McCulloch ve Walter Pitts tarafından gerçekleştirildi. Bu grup, çalışma için üç tane bilgi kaynağı kullanmışlardır. İlk olarak, Araştırmacılar, nöronların elektriksel ve kimyasal sinyalleri nasıl ilettiğini ve işlediğini açıkladılar; ikincisinde, Russell ve Whitehead tarafından önerme mantığının biçimsel bir analizi ele alındı ve üçüncüsünde, Turing'in hesaplama teorisi açıklandı [12].

Warren McCulloch ve Walter Pitts, her bir nöronun "açık" veya "kapalı" durumda olduğu ve bu değişikliğin yeterli sayıda komşu nörondan gelen uyarana yanıt olarak gerçekleştiği yapay bir nöron modeli önerdi. Bir nöronun durumu, bir uyarıcıya dair bir "inanç" olarak görülebilir. Ancak, bu inanç her zaman doğru değildir. Nöronlar, bazen yanlış bilgilere dayanarak "inançlar" geliştirebilirler. Örneğin, hesaplanabilir fonksiyonların bağlı nöronlardan oluşan bir ağ tarafından hesaplandığını ve tüm mantıksal bağlaçların (ve, veya, değil, vb.) basit bir ağ yapısı tarafından uygulandığını göstermiştir. McCulloch ve Pitts, yapay sinir ağlarının öğrenme yeteneğine sahip olduğunu öne sürerek, yapay zeka araştırmalarında çığır açtı. Donald Hebb, bu yeteneği daha da geliştirmek için, nöronlar arasındaki bağlantıların gücünü değiştiren basit bir kural önerdi. Bu kural, bugün hâlâ yapay sinir ağlarının öğrenmesini sağlayan temel bir mekanizmadır. [13].

Yapay zekâ olarak nitelendirilebilecek pek çok ilk çalışma arasında en etkili olanı muhtemelen Alan Turing'in çalışması idi. Alan Turing'in vizyonu, yapay zeka alanındaki ilerlemeleri hızlandıracak ve insanlığın geleceğini şekillendirecek bir güce sahipti. Daha 1947'de Londra Matematik Derneği'nde konuyla ilgili bir konuşma yapmış ve 1950 tarihli "Hesaplama Makineleri ve Zekâ" başlıklı makalesinde ilgi çekici bir şekilde konuyu ele almıştır. Yapay zeka araştırmaları, Turing Testi'nin tanıtılmasıyla yeni bir dönüm noktasına ulaştı. Bu test, bir makinenin insan seviyesinde akıl yürütebileceğini ölçmek için bir yöntem olarak kullanıldı. Turing Testi'nin başarısı, makine öğrenimi, genetik algoritmalar ve takviye öğrenimi gibi yeni araştırma alanlarının doğmasına yol açtı. Bu alanlar, yapay zekanın insan zekasına yaklaşmasına yardımcı olmak için yeni ve heyecan verici yöntemler sunuyor. "Çocuk programı" fikrini ortaya atarak bir yetişkinin zihnini taklit eden bir program yaratmaya çalışmak yerine, bir çocuğun zihnini taklit eden bir program yaratmaya çalışılabileceğini açıkladı [14].

Princeton Üniversitesinde yapılan araştırmada yapay zekâ üzerinde önemli ve etkili bir araştırması olan John McCarthy, 1951 yılında Princeton'dan doktorasını almış ve Stanford

Üniversitesi'ne, ardından da Dartmouth College'a geçmeden önce iki yıl boyunca burada eğitimci olarak çalışmıştır. McCarthy, yapay zeka araştırmalarının geleceği için bir vizyona sahipti. Bu vizyonu gerçekleştirmek üzere, yapay zekanın temellerini atmak adına farklı disiplinlerden araştırmacıları bir araya getiren, Dartmouth Konferansının düzenlenmesini sağladı. McCarthy, Minsky, Claude Shannon ve Nathaniel Rochester'ı bu konuda ikna ederek, yapay zeka araştırmaları için bir dönüm noktası oluşturmuştur. Bu araştırmacılar, 1956 yazında Dartmouth College'da iki aylık bir atölye çalışması düzenlediler. 1956 yazında Hanover, New Hampshire'daki Dartmouth College'da iki ay süresince 10 kişilik bir yapay zeka çalışması yürütmeyi önerdi. Söz konusu çalışma, öğrenmenin her yönünün ya da zekanın diğer tüm özelliklerinin prensipte büyük bir doğrulukla tanımlanabileceği ve bunları simüle edecek makinelerin yapılabilmesi varsayımından hareket edecekti. Bu, makinelerin dili kullanmasını, soyutlama ve kavramsallaştırma yapmasını ve yalnızca insanların çözebileceği ve kendilerini geliştirebileceği sorunları çözmesini sağlamanın yollarını bulma girişimidir. Seçkin bir grup bilim insanının bir süre bu sorun üzerinde çalışarak geçirmesi halinde, bu konulardan bir ya da birkaçında önemli ilerlemeler kaydedilebileceğine belirtti [15].

Yapay zekâ araştırmacıları artık bu hedeflere 2020 ile 2040 yılları arasında ulaşabileceğine inanıyor. Gerçek yapay zeka, bir insanla sohbet edebilecek ve ondan ayırt edilebilecek kadar akıllı bir makinedir. Bu tür makineler, araştırmacıların yapay zekanın sınırlarını zorlamasıyla giderek daha olası hale geliyor. Bunun nedeni, araştırmacıların yapay zekanın karmaşıklığını daha iyi anladıkça, onu daha da geliştirmek için yeni yollar bulabilmeleridir. Bu nedenle, yapay zekânın üstesinden gelebileceği alanlar çok sınırlıdır. Uzman sistemler, belirli bir konuda uzmanlaşmış makine öğrenimi modelleridir. Bu modeller, insan uzmanlarının bilgi ve deneyimini kullanarak, o alandaki sorunları neredeyse insan kadar iyi çözebilir. Örneğin, bir uzman sistem, bir kalp monitörünün verilerini analiz ederek, bir kalp krizi riskini tahmin edebilir.

Yapay zeka, dilleri ve görüntüleri anlayabilir ve çevirebilir. Bu, makinelerin insan benzeri bir şekilde bilgi işleyebildiğini ve dünyayı algılayabildiğini gösteriyor. Aynı zamanda araştırmacılar, insanlar gibi düşünebilen, öz farkındalığı olan programlar geliştirmek gibi daha zorlu bir sorun üzerinde de ilerleme kaydetmektedirler. Ancak yapay zekâ araştırmaları henüz istenilen düzeyde ilerlemiş değil.

İnsan zihni çok esnek bir cihazdır. Her gün yaptığımız her şeyi düşünün: yürümek, konuşmak, duyguları deneyimlemek, yemek yemek, duyularımızla anlamak, öğrenmek, TV izlemek, oyun oynamak vb. Örneğin, Boş bir bilgisayar ekranı, bir mesaj, bir soru veya bir gizem olabilir. Ne anlama geldiğini anlamak için, ekranın neden karardığını bilmemiz gerekir. Esneklik, bir kumaş gibidir. Her duruma uyum sağlayabilir. Örneğin, bilgisayar ekranı kararır, bunu bir ekran koruyucunun tetiklenmesi olarak değerlendiririz. Boş bir ekran, bir gizem gibidir. Fareyi hareket ettirdiğimizde, aydınlanma umuduyla onu aydınlatmak için bir tılsım gibi kullanırız. Ancak, çoğu zaman tılsım işe yaramaz ve gizem çözülmez. Çevik programlamaya sahip bilgisayarlar, tıpkı insanlar gibi durumlara uyum sağlayabilir. Bulanık mantık ve çevik programlamanın yaygın olarak kullanıldığı alanlardan biri de oyun bilgisayarlarıdır [16].

İnsan zekâsının ayırt edici özelliklerinden biri, yaşanan duruma bağlı olarak tepkileri değiştirme yeteneğidir. Düşünmek sadece bilgisayar monitörleri için değil, araba kullanmak gibi diğer durumlarda da kullanılır. Hız sınırı tabelasına uymak gibi basit bir konuda bile esnek davranırız. Ne yapacağımıza karar vermek için zekâmızı kullanırız. Aynı kuralları aynı şekilde takip etmek yerine, neyin gerekli ve güvenli olduğuna karar veririz. Bulanık dünyayı bizim gibi idare edebilmesi için farklı durumlara esnek bir şekilde yanıt verebilen yazılımlar yaratmak da yapay zekânın geliştirilmesine yardımcı olacak önemli bir araç olacaktır. Bir diğer avantaj ise yapay zekânın önyargıya daha az eğilimli olmasıdır. Örneğin, sürekli endişeli bir hastası olan bir doktor, hastanın sadece kendi sorunlarını uydurduğunu düşünebilir. Doktor hayal kırıklığına uğrayabilir ve hastanın şikâyetlerine daha az duyarlı olabilir. Bunu yaparken de gerçek sorunu gözden kaçırabilir. Öte yandan bir bilgisayarın sinirlenme ve her vakayı aynı şekilde ele alma olasılığı daha düşüktür. Bu tür bir bilgisayar, doktorun gözden kaçırabileceği şeyleri de yakalayabilir. Modern bilgisayarlar mimarların bina tasarlamasına da yardımcı olur. Ayrıca arabaların, köprülerin ve otomobillerin tasarımına da yardımcı olurlar. Gelecekte akıllı bilgisayarlar bir araçtan daha fazlası haline gelebilir, onlara tasarım yapma görevi verilebilir. Hatta insan yardımı olmadan sıfırdan tasarım yapabilirler [17].

Bilim insanları veri toplamak ve analiz etmek için bilgisayarları kullanıyor. Bir gün akıllı bilgisayarlar hipotezleri formüle etmemize ve bunları test etmek için deneyler planlamamıza yardımcı olacak. Gerçekten de bilgisayarlar, insanların kendi başlarına çözemeyeceği sorunları çözmek için şimdiden matematiksel kanıtlar üretiyor. Gerçekten akıllı bilgisayarlar,

insanların yanıtlaması uzun zaman alacak bilimsel sorulara yanıt bulabilecek duruma gelecektir.

2.1 MAKİNE ÖĞRENİMİ (MACHINE LEARNING)

“Makine Öğrenmesi nedir?” sorusunu basit anlamda bir dijital makinaya öğrenmesi gereken komutları ekleyip yaptığı davranışı gözlemlemek olarak tanımlayabiliriz. Bilgisayarların öğrenmesini sağlayabilseydik, onlara yeni şeyler öğretmekten ziyade, onları kendileri için öğrenmeye teşvik edebilirdik. Bu, insanlığın potansiyelini önemli ölçüde genişletecekti. Örneğin, tıbbi kayıtlardan yeni bir hastalık için en etkili tedaviyi öğrenen bir bilgisayar, sakinlerinin deneyimlerinden edindiği özel kullanım modellerine göre enerji maliyetlerini optimize eden bir ev veya kullanıcının değişen ilgi alanlarını öğrenen ve çevrimiçi bir sabah gazetesinden özellikle önemli öğeleri vurgulayan kişisel bir yazılım asistanı hayal edin. Makaleleri vurguladığınızı hayal edin. Bilgisayarların nasıl öğrendiğinin daha iyi anlaşılması, bilgisayarların yeni kullanımlarını, yeni yetenek düzeylerini ve adaptasyonunu mümkün kılacaktır. Bilgisayarların nasıl öğrendiğini anlamak, insanların nasıl öğrendiğini anlamamıza yardımcı olabilir. Bu, öğrenmenin doğası ve sınırlamaları hakkında yeni bilgiler sağlayabilir [18].

Artık "büyük veri" çağındayız. Dünyada artık her şeyin bir verisi var. Bu veriler, bir zamanlar sadece şirketlerin kontrolündeydi. Ancak, şimdi herkes bu verilere erişebilir ve bunları kullanabilir. Ancak kişisel bilgisayarların ortaya çıkması ve kablosuz iletişimin yaygınlaşmasıyla birlikte her birimiz veri üretiyoruz. Bir ürün satın aldığımızda, bir film kiraladığımızda, bir web sitesini görüntülediğimizde, bir blog yazdığımızda, sosyal medya paylaşımları yaptığımızda, hatta yürürken veya araba kullanırken bile veri üretiyoruz. Her birimiz, bir veri akışının hem kaynağı hem de havuzuyuz. Çağımız insanları, kendilerine özel ürün ve hizmetlere sahip olmayı, ihtiyaçlarının anlaşılmasını ve çıkarların tahmin edilebilmesini ister.

Örneğin, bir ülkenin dört bir yanındaki yüzlerce fiziksel mağazada ya da web üzerinden sanal bir mağaza aracılığıyla milyonlarca müşteriye binlerce ürün satan bir süpermarket zincirini düşünün. Süpermarket zincirleri, müşteri davranışlarını anlamak için bir bulmacayı çözmeye çalışır. Her gün milyonlarca parçayı bir araya getirmeye çalışırlar ve her parça, bir müşterinin yaptığı bir satın alma işlemidir. Bulmacayı çözebilirlerse, hangi müşterinin hangi ürünü satın alacağını tahmin edebilirler ve satışlarını artırabilirler. Benzer şekilde müşteriler

de, aradıkları ürün gruplarını bulmak isterler. Bu görev yeterince açık değildir. Ne tür insanların bu dondurmayı satın aldığını veya bu yazarın kitabını satın aldığını veya bu yeni filmi izlediğini, bir şehri ziyaret ettiğini veya bu bağlantıya tıkladığını anlamaya çalışıyoruz, bunu kimin yapacağını tam olarak bilmiyoruz. Müşteri davranışları, bir rüzgarın yön değiştirmesi gibidir. Zaman ve coğrafyaya göre değişir, ancak her zaman bir düzen vardır.

Bilgisayarlar, problemleri çözmek için bir orkestraya ihtiyaç duyar. Bu orkestra, girdiyi çıktıya dönüştürmek için gerekli olan tüm bileşenleri ve süreçleri koordine eder. Bazı problemler için, girdiyi çıktıya dönüştürmek için izlenecek adımlar belli değildir. Örneğin, sıralama için bir algoritma geliştirebiliriz. Bu durumda girdi bir dizi sayı olur ve çıktı, bu sayıların sıralı bir liste halinde olmasıdır. Ancak bazı görevler için belirli bir algoritma mevcut değildir. Müşteri davranışını tahmin etmek gibi, spam e-postalarını yasal olanlardan ayırmak gibi. Girdinin ve çıktının ne olduğunu biliyoruz, ancak girdiyi çıktıya nasıl dönüştüreceğimizi bilmiyoruz çünkü neyin spam olarak kabul edileceği zaman ve kişiden kişiye değişir. Bu, sorunun karmaşık veya değişken olmasından kaynaklanabilir [19].

Bilgi eksikliğimizi veri analizi ile gideririz. Bildiğimiz kadarıyla bazı mesajların spam olduğu ve bazılarının olmadığı binlerce örnek veri bulup toplayabiliriz. Amacımız, bu verilerden neyin spam olduğunu otomatik olarak öğretecek bir algoritma oluşturmaktır. Sayıları sıralamayı öğrenmek gerekli değil çünkü bu alan için zaten belirli algoritmalar mevcut. Fakat algoritmamızın olmadığı birçok uygulama ve çok miktarda veri var.

Süreci tam olarak tanımlamak mümkün olmasa da faydalı yaklaşımlarda bulunmanın mümkün olduğuna inanıyoruz. Bu yaklaşımlar her şeyi açıklayamayabilir ancak verilerin bir kısmını açıklayabilir. Sürecin tamamını tanımlamak mümkün olmasa da bazı kalıpların ve düzenliliklerin tanımlanabileceğine inanıyoruz. Makine öğreniminin temeli burada yatar. Bu örüntüler, süreci anlamamıza yardımcı olabilir ve bu örüntüleri tahminlerde bulunmak için kullanabiliriz. Eğer gelecekte toplanacak örnek veriler, geçmişten çok da farklı değilse, gelecekteki tahminlerin doğru olması muhtemeldir, ya da en azından beklenir.

Veri madenciliği, büyük veri okyanusunda gizli hazineleri bulmak için makine öğrenimi yöntemlerini kullanarak, bir sır perdesini aralayarak, büyük veri tiyatrosunda izleyiciye yeni bir bakış açısı sunan bir sanattır. Bir maden, bir altın madenine benzer. Dışarıdan bakıldığında, toprağın altındaki altın gibi değerli bir şeyin varlığından habersiz büyük bir kaya parçası gibi görünür. Ancak, madenciler bu kayayı kazmaya ve işlemeye

başladıklarında, altın parçacıklarının ortaya çıkmaya başladığını görürler. Bu parçacıklar, kayanın toplam kütesinin küçük bir kısmını oluşturur, ancak yine de çok değerlidir. Benzer şekilde veri madenciliğinde de büyük miktarda veri işlenerek değerli kullanımı olan modeller, örneğin yüksek tahmin doğruluğuna sahip basit bir model oluşturulabilir. Uygulama alanları çok fazladır. Perakendeye ek olarak finans alanında bankaların kredi kontrolleri, dolandırıcılık tespiti ve borsalar için kullanılan modeller uygulama alanlarından bazılarıdır. Makine öğrenimi, üretimde süreçleri optimize etmek, tıpta teşhis koymak, telekomünikasyonda ağları iyileştirmek ve bilimde yeni keşifler yapmak için kullanılan bir araçtır. Bu araç, büyük miktarda veriyi hızlı ve verimli bir şekilde analiz ederek, insan zekasının ulaşamayacağı sonuçlar elde edilmesini sağlar. World Wide Web çok genişler ve giderek büyümektedir ve ilgili bilgilerin elle aranması imkansızdır [20].

Makine öğrenimi, bir veritabanından bilgi çekmekten çok daha fazlasını ifade eder. Yapay zekanın bir parçası olarak, bir sistemde öğrenme ve uyum sağlama yeteneğini temsil eder. Bu, değişen bir ortamda bile sistemlerin verimliliğini ve performansını korumasına olanak tanır.

Makine öğrenimi ayrıca görme, konuşma tanıma ve robotik gibi çeşitli sorunlara çözüm bulunmasına yardımcı olmuştur. Örneğin yüz tanıma. Her gün ailemizin ve arkadaşlarımızın fotoğraflarına bakarız ve farklı pozları, farklı ışıkları, farklı saç stilleri vs. olsa bile onları tanıyabiliriz. Ancak bunu bilinçsizce yaparız ve nasıl yaptığımızı açıklayamayız. Ancak bunu bilinçsizce yaparız ve nasıl yaptığımızı açıklayamayız. Bilgisayar programları yazamayız çünkü yeteneklerimizi açıklayamayız. Bir yüz, bir piksel tablosu değil, bir sanat eseridir. Bu eser, doğanın yarattığı bir düzen ve güzellik sergiler. Bir yüzün yapısı, bu düzenin temelini oluşturur. Simetri ise bu düzeni vurgular. Bir yüzün her biri belirli bir konuma yerleştirilmiş gözleri, burnu ve ağzı vardır. Bir kişinin yüzü, bir alfabenin harfleri gibidir. Bu harfler, bir araya getirildiğinde, benzersiz bir kelime oluşturur. Bu kelime, o kişinin kimliğini ifade eder. Bir öğrenme programı, bir kişinin yüzünün örnek bir görüntüsünü analiz ederek o kişinin karakteristik bir desenini yakalayabilir ve bu desenin o görüntüde mevcut olup olmadığını kontrol ederek onu tanımlayabilir. Örüntü Tanıma Yöntemine verilen bir örnek teşkil eder.

Makine öğrenimi, bilgisayarların geçmiş deneyimlerden öğrenerek yeni bilgiler edinme ve yeni davranışlar geliştirme yeteneğidir. Bu, bilgisayarlara, programlama olmadan, yeni

durumlara uyum sağlama ve daha iyi kararlar verme yeteneği kazandırır.

Makine öğrenimi, çıkarım yaparken matematiksel modeller oluştururken istatistik teorisine dayanır. Bu çünkü temel görev, bir örneğin özelliklerinden genel bir kural veya model çıkarmaktır. Bilgisayar biliminin bu süreçteki rolü iki yönlüdür. Eğitim aşamasında, optimizasyon problemlerini çözmek için büyük veri miktarını işleyebilen etkili algoritmalar gereklidir. Model eğitildikten sonra ise temsil ve algoritmik çözümün verimli olması, çıkarım sürecinin uzamsal ve zamansal karmaşıklığı gibi faktörler açısından önemlidir. Özellikle bazı uygulamalarda, eğitim ve çıkarım algoritmalarının etkinliği, tahmin doğruluğu kadar kritik bir rol oynayabilir.

Bilgisayarların ayrıca insanların öğrendiği şekilde öğrenmesini nasıl sağlayacağımızı henüz bilmiyoruz. Bununla birlikte, belirli öğrenme görevleri için etkili algoritmalar tasarlanmış ve öğrenmeye ilişkin teorik anlayış gelişmeye başlanmıştır. Buna ek olarak, faydalı öğrenme yeteneklerine sahip bazı pratik bilgisayar programları geliştirilmiş ve önemli ticari uygulamalar ortaya çıkmıştır. Makine öğrenimi, bilgi işlemin altın anahtarıdır. Bilgi işlem anlayışımız geliştikçe, bu anahtar daha da parlak ve güçlü hale geliyor.

Konuşma tanıma gibi problemlerde, makine öğrenimi algoritmaları denenen diğer tüm yaklaşımlardan daha iyi performans göstermektedir. Veri madenciliği adı verilen bir alanda, makine öğrenimi algoritmaları genellikle ekipman bakım kayıtları, kredi başvuruları, finansal işlemler, tıbbi kayıtlar vb. içeren büyük ticari veri tabanlarından değerli bilgileri keşfetmek için kullanılır.

Öğrenme deneyimi, bir yolculuktur. Bu yolculuğun türü, yolcunun başarısını veya başarısızlığını büyük ölçüde etkileyebilir. Önemli bir faktör, yolculuğun boyunca yolcuya geri bildirim sağlayıp sağlamadığıdır [21].

Örneğin, bir satranç oyununu öğrenirken, sistem bireysel bir satranç taşının durumundan ve her doğru hamleden oluşan doğrudan öğrenme örneklerinden öğrenebilir. Alternatif olarak, sistem çeşitli oyunlardan hamle ve sonuç şeklinde sadece dolaylı bilgi alabilir. Bu durumda, oyunun başında belirli hamlelerin doğruluğu hakkındaki bilgiler, son oyunun kazananları ve kaybedenlerinden dolaylı olarak çıkarılmalıdır. Burada öğrenci ayrıca ders atama, yani dizideki her hamlenin nihai sonuç için ne kadar krediyi veya suçu hak ettiğini belirleme sorunuyla karşı karşıyadır. Ders atamak çok zor bir sorundur çünkü ilk hamle optimal olsa bile, bunu bir dizi kötü hamle izleyebilir ve bu da kayba yol açabilir. Böylelikle, doğrudan

geri bildirim, bir öğretmen gibidir. Öğrenciye, hatalarını ve doğrularını gösterir. Dolaylı geri bildirim ise bir öğrenci gibidir. Öğrenciye, hatalarını ve doğrularını söylemez, ancak öğrencinin kendi hatalarını ve doğrularını bulmasına yardımcı olur. Alternatif olarak, öğrenci çok kafa karıştırıcı bulduğu bir tabloyu işaret edebilir ve öğretmenden doğru adımları isteyebilir. Alternatif olarak, öğrenci, öğretmen olmadan kendi başına öğrenirken olduğu gibi, tablonun durumu ve eğitimin sınıflandırılması (dolaylı olarak) üzerinde tam kontrole sahiptir. Bu durumda, öğrenciler henüz test edilmemiş yeni satranç durumlarını denemeyi seçebilir veya en umut verici mevcut oyun çizgilerinin küçük varyasyonlarını oynayarak becerilerini geliştirebilirler. İlerleyen bölümlerde, öğrenme deneyiminin öğrencinin kontrolü dışındaki rastgele süreçler tarafından sağlandığı, öğrencinin deneyimli bir öğretmene farklı türde sorular yönelttiği ve öğrencinin çevreyi bağımsız olarak keşfederek öğrenme örnekleri topladığı ortamlar da dâhil olmak üzere çeşitli öğrenme ortamlarını inceleyeceğiz [21].

Genellikle yapay zeka (AI) olarak adlandırılan bu tür sistemlerin sofistike problem çözme yetenekleri, tahminler, kurallar, cevaplar, öneriler veya benzer sonuçlar üreten analitik modellere dayanmaktadır. Analitik modeller oluşturmaya yönelik ilk çabalar, akıllı sistemlere, bir insan uzmanının bilgeliğini ve deneyimini, elle hazırlanmış kurallar aracılığıyla aktarmaya dayanıyordu. Bu kurallar, bir uzmanın, bir problemin nasıl çözüleceğine dair bilgisini yansıtıyordu. Yeni programlama çerçevelerinin pratikliği, veri mevcudiyeti ve gerekli bilgi işlem gücüne kapsamlı erişim sayesinde artık giderek artan sayıda analitik model, genellikle makine öğrenimi olarak adlandırılan yöntem kullanılarak oluşturulmaktadır (Brynjolfsson ve McAfee 2017; Goodfellow vd. 2016). Makine öğrenimi, insanları bilgiyi makine tarafından erişilebilir formlarda tanımlama ve biçimlendirme yükünden kurtararak daha verimli akıllı sistemlerin geliştirilmesini sağlar.

2.1.1 Öğrenim İlişkileri

Bir araştırmacı, dünyayı gözlemledikten sonra gelecekteki görevlerde performansını artırırsa öğrenme içgüdüsüne sahip olur. Öğrenme, bir telefon numarasını yazmak kadar önemsiz bir şeyden Albert Einstein'ın yeni bir evren teorisine ulaşması kadar derin birçok şeye kadar değişebilir. Bu bölümde, bir buzdağı gibi görünen bir öğrenme problemine odaklanacağız. Yüzeyde sınırlı görünebilir, ancak derinlerde çok daha fazlasına sahiptir. Mantıksal etmen bileşenleri için önermeler mantığı ve birinci dereceden mantık ve deterministik etmenlerin

çıkarımsal bileşenleri için Bayesian ağları gibi çeşitli etmen bileşeni temsil örnekleri gördük. Makine öğrenimi araştırmacıları, bu gösterimler için etkili öğrenme algoritmaları geliştirmek için çok çalıştılar. Bu bölümde, bu algoritmaların çoğunun temelini oluşturan faktörlü temsiller hakkında konuşacağız [22].

Örnek yoluyla öğrenme, hem araştırma hem de endüstriyel uygulamalar açısından büyük ilgi görmüştür. Öğrenme algoritmaları tipik olarak özellik kombinasyonları ve sınıflandırılmış sınıflar arasındaki olası ilişkilerin geniş bir uzayında aramaya rehberlik etmek için sezgisel yöntemler kullanır. Öğrenme görevlerindeki nitelikler nominal (kategoriler) veya sürekli (sayılar) olabilir. "Sürekli" terimi literatürde gerçek ve tamsayı öz nitelikleri ifade etmek için kullanılır. Tipik olarak bir öznitelik, doğrusal olarak sıralanmış bir değer aralığına sahip bir özniteliktir. Çoğu öznitelik seçim süreci tüm özniteliklerin nominal olduğunu varsayar. Bu nedenle, sürekli değerlere sahip öznitelikler, öznitelik seçiminden önce ayrıklaştırılmalıdır. Şu anda bir dizi ayrıklaştırma algoritması mevcuttur, örneğin eşit genişlikli aralıklar, eşit frekanslı aralıklar gibi.

2.1.2 Sınıflandırma

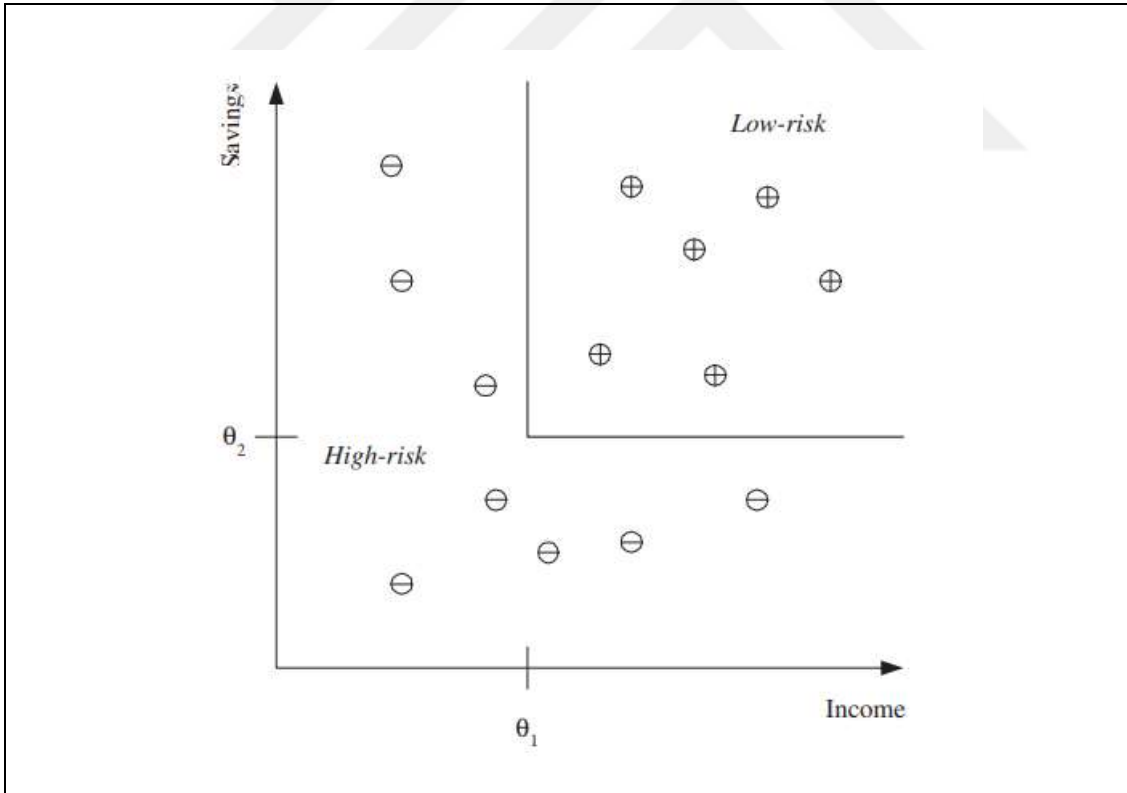
Kredi, banka gibi bir finans kuruluşundan ödünç alınan ve genellikle taksitler halinde faiziyle birlikte geri ödenen bir miktar paradır. Bankalar için kredilerle ilgili riskleri önceden tahmin etmek, bir doktorun bir hastanın durumunu önceden teşhis edebilmesi gibidir. Risk, bir müşterinin temerrüde düşmesi ve kredinin tamamını geri ödememesi olasılığıdır. Bu, bankanın kar elde edebilmesi ve müşterilerin mali kapasitelerini aşan krediler nedeniyle sıkıntıya düşmemeleri için yapılır.

Krediyi değerlendirirken (Hand 1998), bankalar riski kredi miktarına ve müşteri hakkındaki bilgilere dayanarak hesaplar. Müşteri hakkındaki bilgiler, bir bireyin finansal kapasitesini hesaplamada önemli olan gelir, tasarruf, teminat, meslek, yaş ve geçmiş finansal geçmiş gibi erişebileceğimiz verileri içerir. Bankalar müşteri verilerine ve geri ödemelerin veya geri ödememelerin kaydedildiği geçmiş kredi kayıtlarına sahiptir. Buradaki amaç, başvurular için özel verilerden müşteri özellikleri ve risk arasındaki ilişki hakkında genel kurallar türetmektir. Makine öğrenimi sistemi, geçmiş verilerdeki ipuçlarını kullanarak yeni bir başvurunun riskini tahmin eder. Bu, bir doktorun hastanın geçmiş tıbbi kayıtlarını kullanarak hastanın durumunu teşhis etmesine benzer. Makine öğrenimi sistemi, yeni bir başvuruyu bir mayın tarlasından geçerken bir asker gibi değerlendirir.

Geçmiş veriler üzerinde eğitim yapıldıktan sonra, öğrenilen sınıflandırma kuralları aşağıdaki biçimi alır.

IF income > θ_1 AND savings > θ_2 THEN low-risk ELSE high-risk

θ_1 ve θ_2 'nin karşılık gelen değerlerine bölünür (bkz. Şekil 2.3). Bu, bir anahtar gibidir. Bir anahtar, farklı kilitleri açmak için kullanılabilir. Benzer şekilde, diskriminant fonksiyonu, farklı sınıfları ayırmak için kullanılabilir. Bu tür kuralların ana kullanım alanı tahminîdir. Geçmiş verilere uyan kurallarla, gelecek geçmişe benziyorsa yeni vakalar için doğru tahminler yapılabilir. Yeni olgular için doğru öngörüler sağlanabilir. Belirli gelir ve tasarruflara sahip yeni bir başvuru göz önüne alındığında, başvurunun düşük veya yüksek riskli olup olmadığını belirlemek kolaydır. Bazı durumlarda, X'in müşteri özellikleri ve Y'nin düşük risk ile yüksek risk için sırasıyla 0 veya 1 olduğu durumunda, 0/1 tipi bir kesin karar yerine $P(Y|X)$ gibi bir olasılık hesaplamak isteyebiliriz. Bu, belirli bir durumun gerçekleşme olasılığını daha kesin bir şekilde değerlendirmemizi sağlar [23].



Şekil 2.3: Modelin Eğitim Sürecinde Kullanılan Örnek Veri Seti.

Her daire, bir veri örneğinin izini sürer ve onu bir sınıfa yerleştirir. Basitlik açısından, girdi verisi olarak yalnızca iki müşteri özelliği; yani gelir ve tasarruflar ve iki sınıf, yani düşük

risk ("+") ve yüksek risk ("-") kullanılmıştır. İki türü birbirinden ayıran bir ayırım örneği de verilmiştir.

Müşterinin yüksek riskli olma olasılığı, X 'in x değeri için $P(Y = 1|X = x) = 0,8$ ise, bir para atıp tura gelmesine benzer. Yüksek riskli olma olasılığı %80 iken, düşük riskli olma olasılığı %20 'dir. Firma daha sonra kar kaybetme olasılığına bağlı olarak kredinin kabulüne ya da reddine karar verir.

Örüntü tanıma makine öğreniminin birçok uygulaması vardır. Görsel karakter tanıma, görüntülerden karakterlerin kodlarını tanıyan bir tür örüntü tanımadır. Bu, alfabemizdeki karakter sayısı kadar sınıfın olduğu bir örnektir. Zarfın üzerindeki posta kodunu veya bir çekin tutarını okumak gibi el yazısı karakterleri içeren durumlar özellikle ilgi çekicidir. El yazısı harfler farklı kişiler tarafından farklı şekillerde yazılır: küçük ya da büyük, çapraz, tükenmez ya da kurşun kalemle ya da aynı harflerin farklı kalıplarıyla. Yazı bir insan icadıdır ancak hiçbir sistem insan okuması kadar doğru değildir. Tüm "A" ları içeren ve "A" olmayanları hariç tutan resmi bir "A" tanımına sahip değiliz. Bu nedenle, "Aness" tanımını öğrendiğimiz yazarlardan örnekler alıyoruz. Ancak, bir resmi "A" yapan özelliği tam olarak bilemiyoruz; fakat farklı "A"ların paylaştığı bir ortak özelliğin olduğuna eminiz ve bu özelliği elde etmek istediğimiz şey de bu. Bir karakter görüntüsü, sadece rastgele noktaların bir araya gelmesi değil, aynı zamanda bir öğrenme programının algılayabileceği düzenliliklere sahip fırça darbelerinin bir kombinasyonudur. [23].

Metin okurken diğer bir unsur da insan dilindeki fazlalık kullanımıdır. Bir kelime bir harf dizisidir ve bu dizideki harfler tek başına durmaz sadece dildeki kelimelerle sınırlandırılır. Bu, bazı harfler tanınmasa bile kelimenin okunabilmesi gibi bir avantaja sahiptir. Dil, kelimelerin ve cümlelerin birbirleriyle karmaşık etkileşimleri yoluyla anlam kazanır. Bu etkileşimler, makine öğrenimi algoritmaları tarafından öğrenilebilir ve modellenir.

Yüz tanıma durumunda girdi, bir görüntüdür; sınıf tanınacak yüzdür ve öğrenme programı yüz görüntüsü ile kimlik arasındaki ilişkiyi öğrenmelidir. Bu problem, çok sayıda sınıf, büyük girdi görüntüleri ve yüzlerin 3B olması ve görüntülerin farklı duruşlar ve ışıkla önemli ölçüde değişmesi nedeniyle karakter tanımadan daha zordur. Buna ek olarak bazı girdiler gizlenebilir, örneğin gözlükler tarafından gizlenen gözler ve kaşlar veya sakallar tarafından gizlenen çene.

Tıbbi tanısında, hastanın vücudunun bir hikayesini dinlersin ve bu hikayeyi hastalığın bir

şifresi olarak çözmeye çalışırsın. Bazı testler hastaya uygulanmaz, dolayısıyla bu girdiler eksiktir. Bu testler zaman alıcı, maliyetli ve hasta için potansiyel olarak rahatsız edicidir, bu nedenle hasta bu testlerin değerli bilgiler sağlayacağından emin olmadıkça yapılmamalıdır. Tıbbi teşhiste, bir makinenin kararının yanlış olması, hastanın hayatını tehlikeye atabilir. Bu nedenle, sınıflandırıcının kararsız olduğu durumlarda, insan uzmana danışmak en iyisidir.

Konuşma tanımada, girdiler işitseldir ve sınıflar konuşulabilen kelimelerdir. Bu görev çok daha zordur çünkü aynı kelime yaş, cinsiyet ve telaffuz farklılıkları nedeniyle farklı telaffuz edilebilir. Bir diğer fark ise girdinin geçişli olmasıdır. Kelimeler zaman içinde bir dizi fonem olarak telaffuz edilir ve bazı kelimeler diğerlerinden daha uzundur.

Karakter tanımada olduğu gibi konuşma tanımada da bir "dil modeli" oluşturmak önemlidir ve bir dil modeli oluşturmanın en iyi yolu geniş bir örnek veriden öğrenmektir. Makine öğrenimi, doğal dil işlemeye yeni bir soluk getiriyor. Örneğin anti-spam, spam yaratıcılarının ve işlemcilerinin spam ile başa çıkmak için yeni ve daha iyi yollar geliştirerek daha iyi yollar bulmaya çalıştıkları bir alandır. Bir başka ilginç örnek de "trend konuları" belirlemek ve reklam içeriğini belirlemek için blogların ve sosyal medya gönderilerinin analiz edilmesidir. Manuel çeviri kurallarına dayalı onlarca yıllık araştırmalardan sonra, belki de en etkileyici olanı makine çevirisi gelmektedir. En umut verici yöntemin, her iki dil için büyük miktarda metin çifti sağlamak ve programın bir metni diğerine eşleştiren kuralları otomatik olarak bulmasını sağlamak olduğu sonucuna varıldı.

Biyometrik fotoğraflar, insanları tanımlamak ve kimliklerini doğrulamak için insanın fizyolojik ve davranışsal özelliklerini kullanan ve birden fazla yöntemden gelen girdilerin entegrasyonunu gerektiren bir teknolojidir. Fizyolojik özellikler arasında yüzler, parmak izleri, irisler ve avuç içleri gibi görüntüler yer alırken, davranışsal özellikler arasında imzalar, sesler, yürüyüş ve tuş vuruşları gibi dinamik özellikler yer alır. Fotoğraflar, imzalar ve şifreler gibi standart kimlik belirleme yöntemlerinin aksine birçok farklı (ilgisiz) girdinin olduğu durumlarda, sahtecilik (kimliğe bürünme) daha zor olacak ve sistem daha doğru olacak ve umulur ki kullanıcı için daha az can sıkıcı olacaktır. Makine öğrenimi hem bu farklı formatlar için ayrı tanıma amacıyla hem de bu farklı kaynakların güvenilirliğini dikkate alan genel bir kabul/red kararı vermek için bir arada kullanılır.

Verilerden elde edilen bilgilerle kuralların öğrenilmesi, bilgi çıkarımını mümkün kılar. Kurallar basitçe veriyi açıklayan modellerdir ve bu modeller incelenerek verinin altında

yatan sürecin bir açıklaması elde edilebilir. Diyelim ki, düşük riskli ve yüksek riskli müşterileri ayıran kalıbı öğrendikten sonra, düşük riskli müşterilerin kim olduğunu ve ne yaptıklarını anlayabiliriz. Bu bilgi, örneğin reklamcılıkta potansiyel düşük riskli müşterileri daha etkili bir şekilde hedeflemek için kullanılabilir. Kurallar, verilerin daha basit açıklamalarını üretmek için verilere uygulanabildiğinden ve böylece depolama için daha az bellek ve işleme için daha az hesaplama gerektirdiğinden, öğrenmenin bir sıkıştırma etkisi de vardır. Toplama kuralı bir kez bilindiğinde, her sayı kombinasyonunun toplamını hatırlamaya gerek kalmaz.

Makine öğreniminin bir diğer uygulaması, genel kurallara uymayan mükemmel örnekleri belirleyen aykırı değer tespittir. Buradaki fikir, tipik örneklerin kolayca ifade edilebilecek özelliklere sahip olduğu, bu özelliklere sahip olmayan örneklerin ise tipik olmadığıdır. Bu gibi durumlarda, mümkün olduğunca basit ve mümkün olduğunca çok sayıda yaygın vakayı kapsayan bir kural bulmak amaçlanır. Dolandırıcılık gibi dikkat gerektiren bir anormallik olabilir veya daha önce gözlemlenmemiş ancak geçerli olan yeni bir durum birer istisna olarak karşımıza çıkar.

2.2 DERİN ÖĞRENME (DEEP LEARNING)

Makine öğrenimi sistemleri, hem sığ hem de derin mimarilerde deneyim yoluyla öğrenme ve gelişme yeteneğine sahiptir. Makine öğrenimi süreci, karar verme için faydalı bilgileri çıkarmak amacıyla kullanılan ham verilerle başlar ve nihai hedef, makinelerin insanlar gibi faydalı bilgileri öğrenmesini sağlamaktır [24].

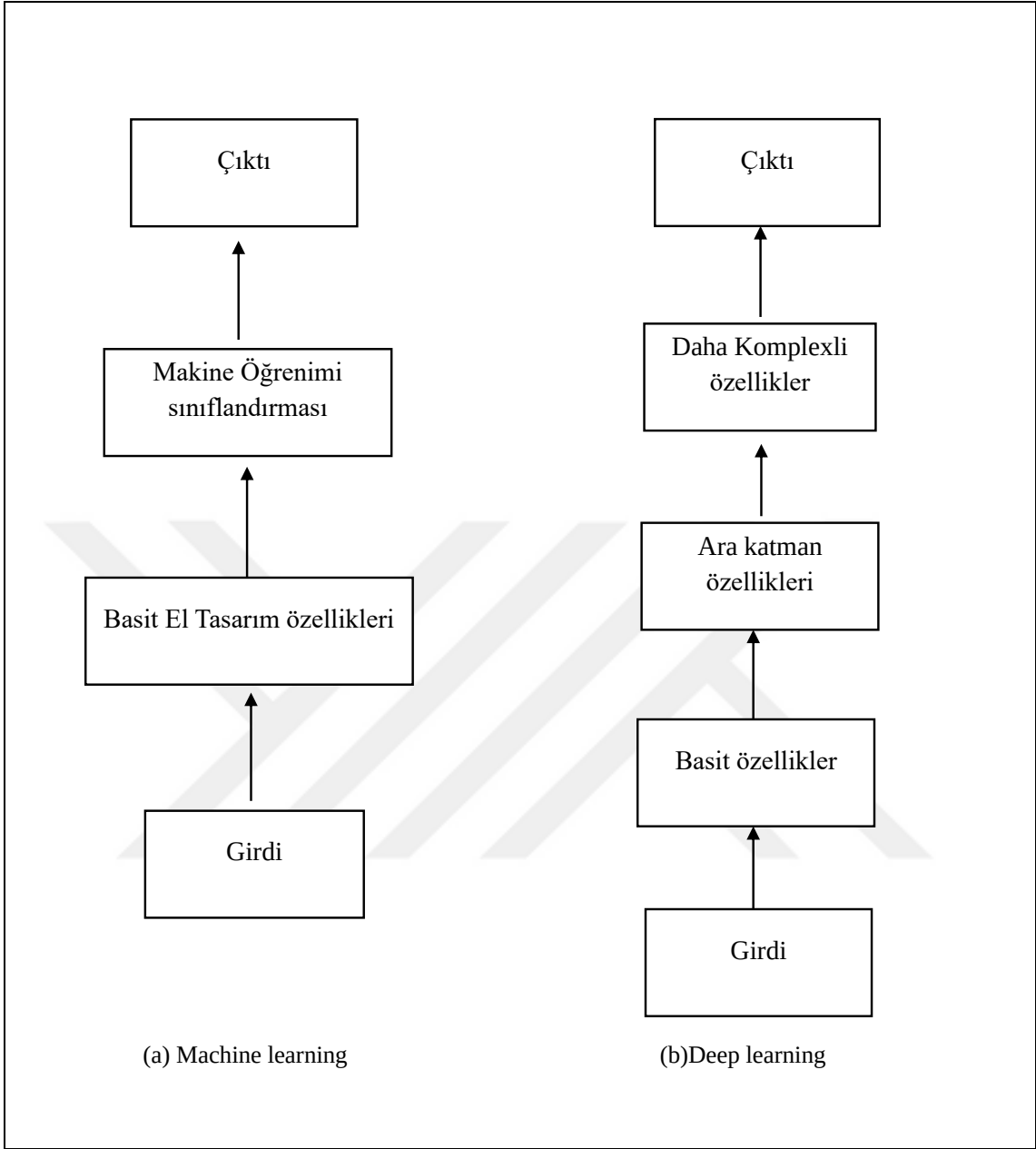
Derin öğrenme, son yıllarda büyük ilgi gören ve hızla gelişen bir makine öğrenimi alanıdır. Derin öğrenme, birçok gizli katmana (deep networks) sahip olan ve farklı soyutlama seviyelerinde farklı özellikleri öğrenen bir mimaridir. Derin öğrenme algoritmaları, düşük seviyeli özellikler tarafından belirlenen yüksek seviyeli öğrenme özellikleri ile genellikle çoklu katmanlarda girdi dağılımının bilinmeyen yapısından yararlanarak üstün bir temsil bulmaya çalışır [25].

Geleneksel makine öğrenimi yöntemleri, verileri doğada var olduğu şekliyle işledikleri gerçeğiyle sınırlıdır. Uzun yıllardır, bir örüntü tanıma veya makine öğrenimi sistemi oluşturmak, ham verileri (örneğin, görüntü piksel değerleri gibi) uygun bir içsel temsile dönüştüren ve özellik vektörleri çıkaran özenli bir uzmanlık ve titiz çalışma gerektirmiştir;

bu özellik vektörü, daha sonra bir sınıflandırıcı gibi öğrenme sistemi tarafından kullanılarak girdideki örüntüleri tanımlamak veya sınıflandırmak için kullanılabilir. Derin öğrenme, ham verilerin (görüntü verileri durumundaki pikseller) önce özellikleri çıkarmadan veya özellik vektörlerini tanımlamadan bir öğrenme algoritmasına sokulmasına izin verir. Derin öğrenme algoritmaları, uygun bir set özellikleri öğrenme yeteneğine sahiptir ve bu özellikleri çıkarmak için manuel programlamadan çok daha etkilidir. Ham verilerden özellik çıkarma işlemini tanımlamak için bir dizi kural veya algoritma oluşturmak yerine, derin öğrenme bu özellikleri öğrenme süreci sırasında otomatik olarak gerçekleştirir [26].

Derin öğrenmede problem, her bir kavramın diğerinin üzerine inşa edildiği bir kavramlar hiyerarşisi olarak uygulanır. Modelin en alt katmanı problemin temel bir temsiliyi kodlarken, daha üst katmanlar ise en alt katmanın üzerine daha karmaşık kavramlar inşa etmektedir.

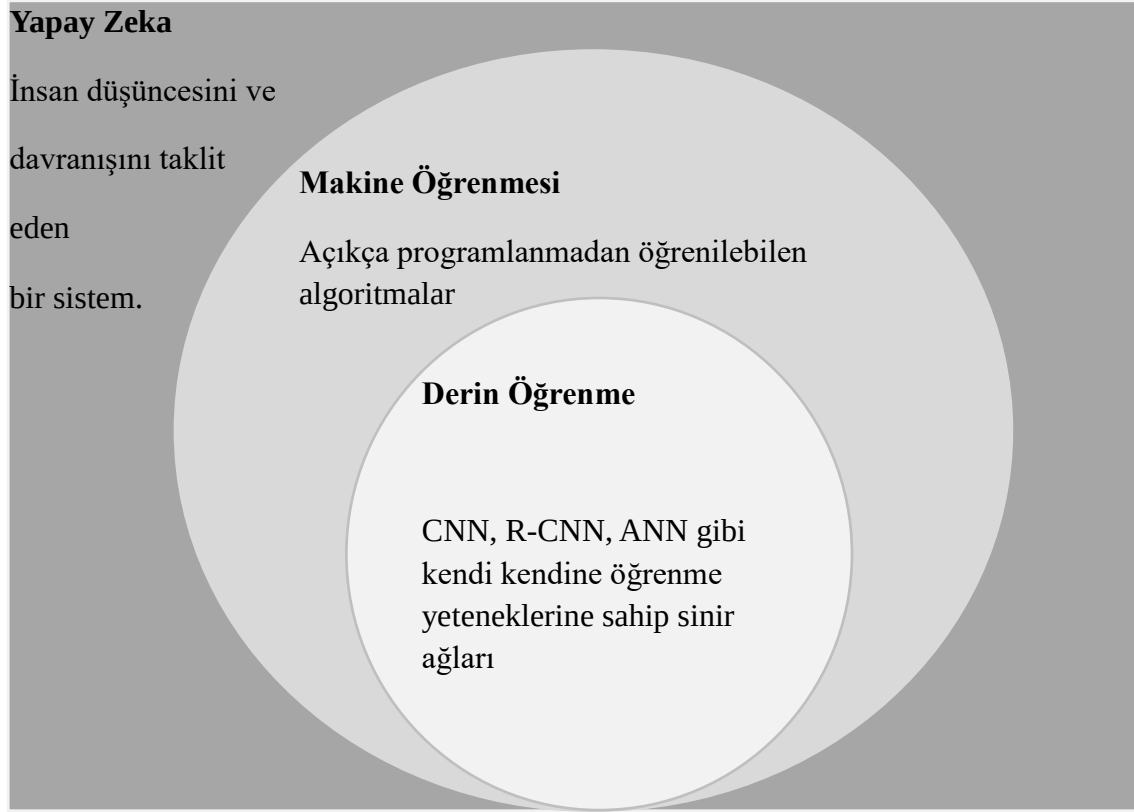
Görüntü verisi sağlandığında, piksel yoğunluk değerleri derin öğrenme sistemi tarafından giriş olarak alınır. Ardından, çoklu gizli katmanlar, giriş görüntüsünden özellikler çıkarır. Bu gizli katmanlar, kendi aralarında hiyerarşik bir yapı oluşturacak şekilde tasarlanmıştır. Ağın en alt katmanı ilk olarak yalnızca kenar benzeri bölgeleri algılar. Bu kenar bölgeleri, köşeleri (kenarların kesiştiği noktalar) ve konturları (nesnelerin çevresi) tanımlamak için kullanılır. En üst seviyede, açılar ve konturlar birleştirilir ve bir sonraki seviyede daha soyut "nesne parçalarına" bağlanır. Derin öğrenme, insan mühendislerinin müdahalesi olmadan, verilerden aşamalı olarak öğrenen ve kendini geliştiren bir makine öğrenmesi tekniğidir. Bu teknik, görüntü işleme, doğal dil işleme, ses tanıma ve makine çevirisi gibi birçok alanda devrim yaratmıştır.



Şekil 2.4: Makine Öğrenmesi ve Derin Öğrenme Karşılaştırılması.

"Derin katmanlar" terimi, girdi verilerinin giderek daha anlamlı temsillerinin sürekli öğrenilmesini ifade eder. Modelin karmaşıklığı, verilerini modellemek için kullandığı katman sayısı ile doğru orantılıdır. Günümüzün derin öğrenmesinde, onlarca ila yüzlerce ardışık temsil katmanı genellikle eğitim verilerinden otomatik olarak öğrenilir. Geleneksel makine öğrenimi yaklaşımları, verileri basit bir şekilde temsil eder. Bu, verilerin karmaşıklığını tam olarak anlamaya yetmeyebilir. Derin öğrenme yaklaşımları, verileri daha karmaşık bir şekilde temsil eder. Bu, verilerin karmaşıklığını daha iyi anlamalarına ve daha doğru tahminler yapmalarına olanak tanır. Ancak, derin öğrenme yaklaşımları daha fazla veri

ve hesaplama gücü gerektirir. Derin öğrenme, makine öğreniminin bir alt kümesidir, ancak verileri temsil etmek için çok katmanlı bir yaklaşım kullanır. Bu, verilerin karmaşıklığını daha iyi anlamalarına ve daha doğru tahminler yapmalarına olanak tanır [27].



Şekil 2.5: Yapay Zeka ve Alt Kümelerini Oluşturan Yapısalı.

Derin öğrenme ağları, verileri daha karmaşık bir şekilde temsil etmek için alt modellerin katmanlarını kullanır. Bu, verilerin karmaşıklığını daha iyi anlamalarına ve daha doğru tahminler yapmalarına olanak tanır. Ancak, derin öğrenme ağları daha fazla veri ve hesaplama gücü gerektirir. Bu, derin öğrenmenin gerçek dünyadaki uygulamalarda kullanımını sınırlayabilir. Derin öğrenme 1980'lerden beri var olmakla birlikte, yetersiz bilgi işlem altyapısı (hem donanım hem de yazılım olarak) ve mevcut veri kümelerinin çok küçük olması nedeniyle birkaç yıldır nispeten popüler değildi. Son zamanlarda geleneksel sinir ağlarının popülaritesi azalırken, derinlik ağları konuşma tanıma ve bilgisayarla görme alanlarında etkileyici sonuçlar ortaya çıkarmıştır [28].

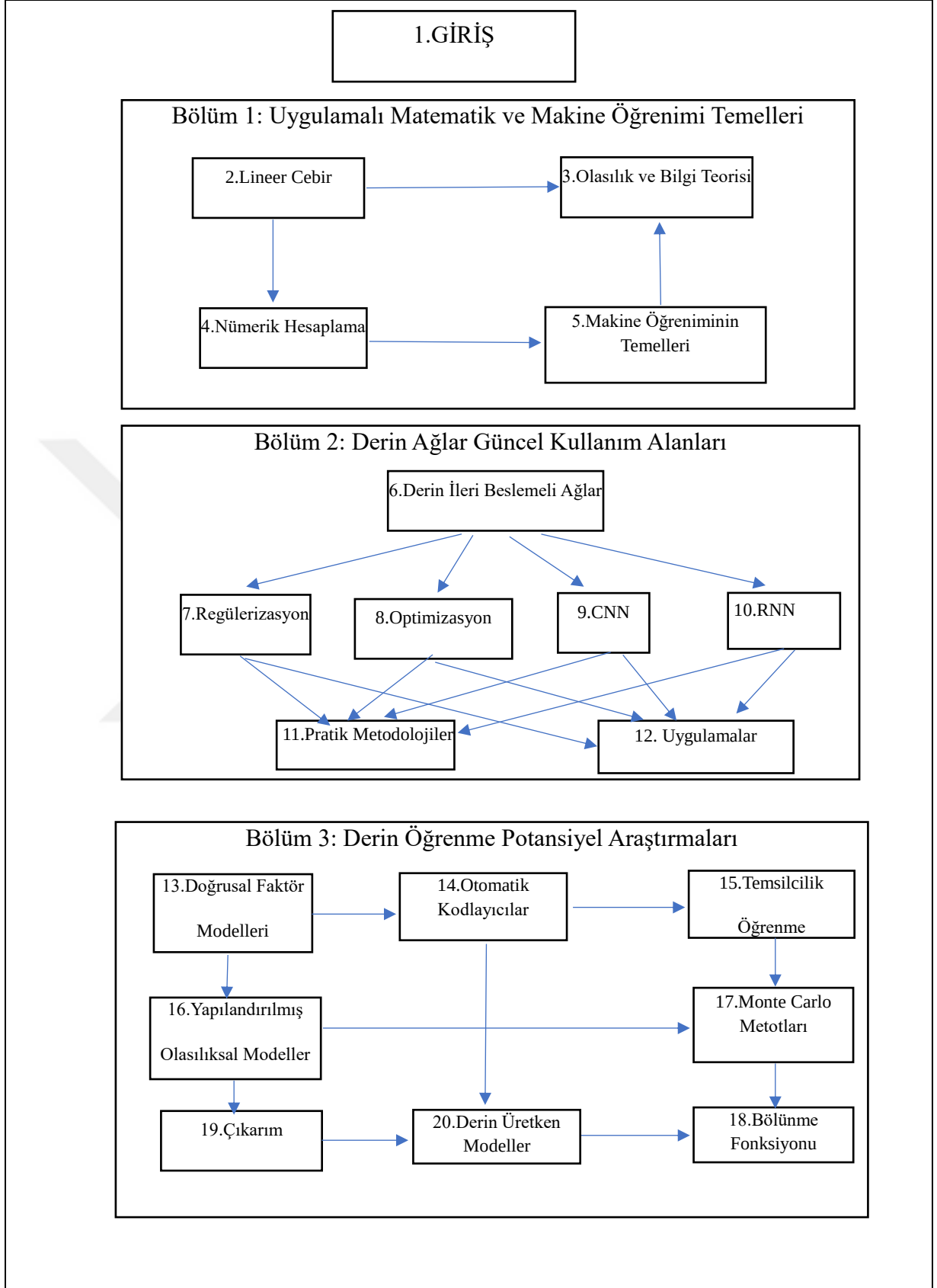
Makine öğrenimi sistemleri, verileri temsil eden özelliklerden ne kadar iyi yararlanırsa, o kadar başarılı olurlar. Bu, çünkü özellikler, sistemlerin veri kümesinin örüntülerini ve ilişkilerini anlamalarına yardımcı olur. Özellik (nesne üzerinde) çıkarma işlemini makine

öğrenimi algoritmaları için daha kullanışlı hale getirmek için probleme dahil edilmesi gereken yönler belirlenmeden daha iyi sonuçlar elde edilemez. Bu nedenle araştırmacılar, makine öğrenimi uzmanlarının faydalı olan özellik setleri elde etmek için alan uzmanlarıyla işbirliği yapılması gerektiğini düşünmüşlerdir. Biyolojik beyin, bir sorunun özünü hızlı bir şekilde kavrayabilir ve bu bilgiyi etkili bir çözüm geliştirmek için kullanabilir. Bu, beyinin bir sorunun karmaşıklığını azaltma ve çözümü basitleştirme konusundaki doğal yeteneğinden kaynaklanmaktadır. Beyin, sorunun önemli özelliklerini hızla belirler ve bu bilgileri sorunu çözmek için kullanır.

Derin öğrenme, insan mühendisler tarafından tasarlanmak yerine geliştirilmiş bir öğrenme süreciyle verilerden özellikler öğrendiği için doğruluk açısından geleneksel algoritmalarından daha iyi performans gösterir.

Derin ağlar, bilgisayarla görme alanında çarpıcı gelişmeler göstermiş, makine çevirisini önemli ölçüde iyileştirmiş ve insan konuşmasını tanımaya yaklaşan konuşma tanıma yetenekleriyle etkili bir yapay zeka teknolojisi olarak odak noktası haline gelmiştir. Makine öğrenimi, diğer disiplinlerden araştırmacıların bile ilgisini çeken, çeşitli alanlarda üstün doğruluk ve genellenebilirlik sağlayan güçlü bir araçtır. Bu, makine öğreniminin, tıp, finans ve üretim gibi kritik karar verme gerektiren alanlarda önemli bir etki oluşturma kapasitesine işaret eder.

Derin öğrenmenin uzun ve zengin bir geçmişi vardır ancak felsefî bakış açılarını yansıtan çeşitli isimlerle popülerliği azalmıştır. Mevcut eğitim verilerinin hacmi arttıkça daha kullanışlı hale gelir. Derin öğrenme için bilgi işlem altyapısı (hem donanım hem de yazılım) geliştikçe derin öğrenme modelleri de zaman içinde evrim geçirmiştir. Derin öğrenme, zaman içinde giderek karmaşıklaştan uygulamaları yüksek bir doğruluk derecesiyle çözmüştür.



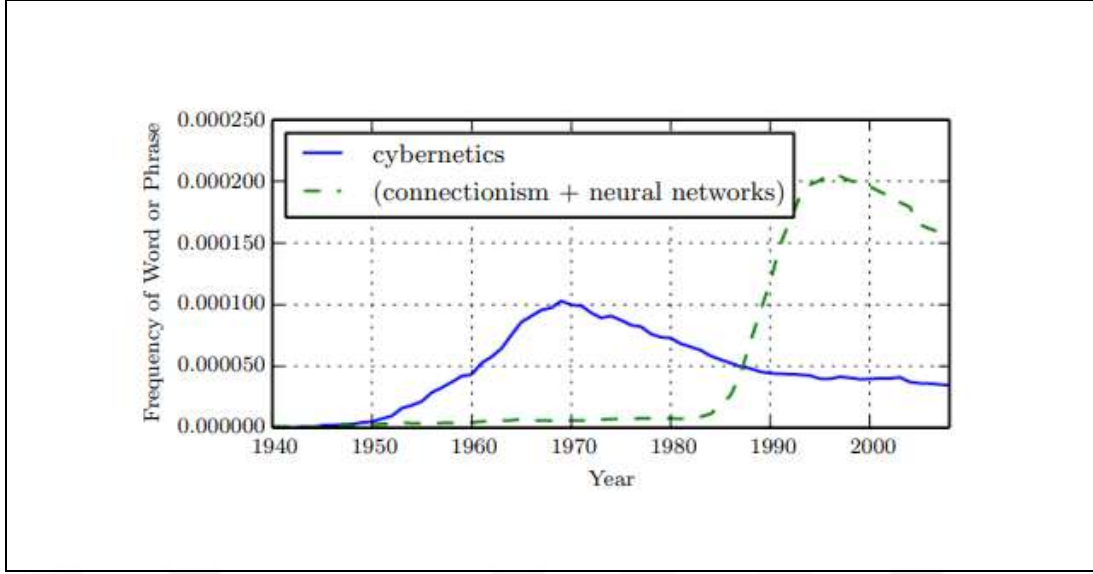
Şekil 2.6: Öğrenme Modellerinin Üst Düzey Yapısı.

Üstteki şekil, öğrenme modelinin üst düzey yapısıdır. Bir bölümden diğerine giden oklar, bir önceki bölümün sonraki bölümleri anlamak için ön koşul niteliğinde olduğunu göstermektedir.

Beynin öğrenme mekanizmalarını taklit eden makine öğrenimi algoritmaları, öğrenmenin temellerini anlamamıza yardımcı olabilir. Bu, öğrenmenin nasıl çalıştığına dair yeni bilgiler elde etmemize ve daha etkili öğrenme algoritmaları geliştirmemize olanak tanır. Sonuç olarak derin öğrenme, yapay sinir ağları (YSA) olarak bilinir hale gelmiştir. Derin öğrenme modelleri, biyolojik beynin yapısını ve işleyişini taklit eden makine sistemleridir. Bu modeller, beyin işlevlerini anlamamıza ve daha etkili makine öğrenimi algoritmaları geliştirmemize yardımcı olabilir. Örneğin, derin öğrenme modelleri, beyin görüntüleme verilerini analiz etmek için kullanılabilir. Bu, beyin hastalıklarının teşhis ve tedavisinde yeni yollar açabilir. (Hinton ve Shallice, 1991), ancak derin öğrenme modelleri, biyolojik beynin işlevini taklit etmek için tasarlanmış olsa da, genellikle tam olarak biyolojik işlevi yansıtmazlar.

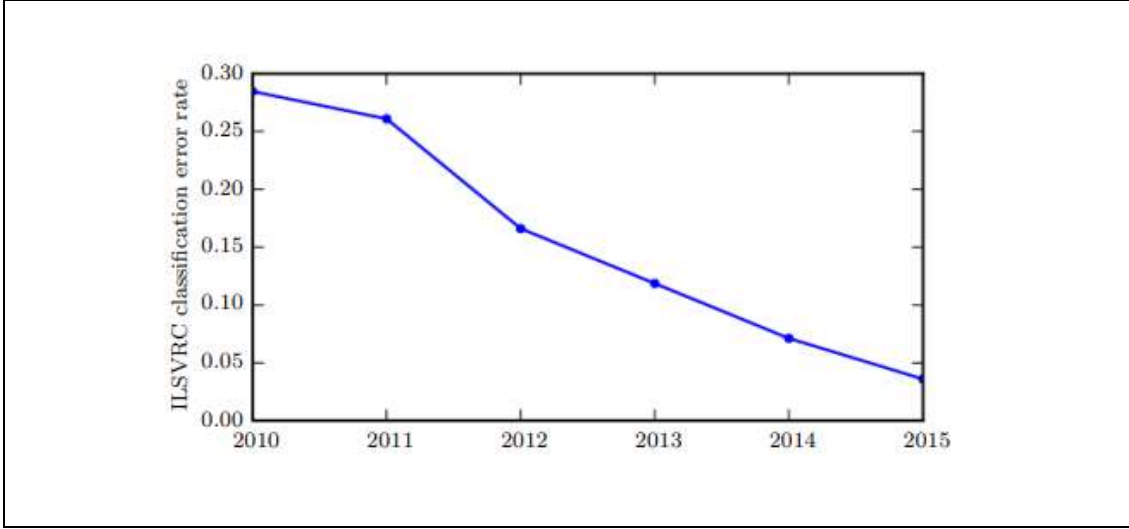
Derin öğrenmenin nöral perspektifi, biyolojik beyinin akıllı davranışın temelini oluşturduğu ve bu temeli yapay sistemlere aktararak insan benzeri zeka elde edebileceğimiz fikrine dayanmaktadır. Bu fikir, biyolojik beynin nasıl çalıştığını anlamamıza ve bu bilgileri yapay zeka sistemlerini geliştirmek için kullanmamıza yardımcı olmuştur. Örneğin, derin öğrenme modelleri, biyolojik beyinin görsel işleme ve dil işleme gibi süreçlerini taklit etmek için kullanılmaktadır. Bu modeller, görüntü tanıma, doğal dil işleme ve makine çevirisi gibi çeşitli uygulamalarda kullanılmaktadır. Bunlar, beyin altında yatan hesaplama ilkelerini anlamak ve işlevlerini kopyalamaktır. İnsan beyninin ve zekasının nasıl çalıştığına dair bilgi edinmek, hem mühendisler hem de bilim insanları için değerlidir. Bu bilgi, yeni teknolojiler geliştirmek, hastalıkları teşhis etmek ve tedavi etmek ve dünyayı daha iyi anlamak için kullanılabilir. İnsan beyninin ve zekasının altında yatan sırları çözmek, hem mühendislik hem de bilimsel açıdan heyecan verici bir hedeftir. Bu nedenle, makine öğrenimi modelleri, bu sırlara ışık tutma potansiyeline sahip oldukları için değerli bir araçtır [29].

Derin öğrenme, biyolojik beyni taklit etmenin ötesine geçerek, kendi kendine öğrenme ve adaptasyon gibi yeni yeteneklere sahip hale geldi. Nörobilimden ilham alması gerekmeyen makine öğrenimi, çerçevelerine uygulanabilen çok seviyeli biçimlendirici öğrenmenin daha genel ilkesini çağırır.



Şekil 2.7: Google Books Verilerine Göre, Yapay Sinir Ağları Araştırmalarının Tarihsel Olarak Üç Ana Dalgada Ölçülen Anahtar Terimler 'Sibernetik' ve 'Bağlantıcılık' veya 'Sinir Ağları' dır. Şekil ise Üç Tarihsel Dalgasından İkisini Göstermektedir.

Derin öğrenme, biyolojik beynin yapısını ve işleyişini taklit etmeye çalışan basit doğrusal modellerden doğmuştur. Bu modeller, biyolojik beynin nasıl öğrendiğini ve akıllı davranışı nasıl ürettiğini anlamamıza yardımcı olmuştur. Ancak, bu modeller sınırlıydı ve karmaşık görevleri yerine getirmekte zorlanıyordu. Nörobilimciler, beyin hücrelerinin nasıl birbirleriyle iletişim kurduğunu ve karmaşık görevleri nasıl yerine getirdiğini öğrendikçe, derin öğrenme modelleri de bu görevleri yerine getirmeyi öğreniyor. Nörobilimciler, gelinciklerin beyinlerini, beynin işitmeyi işleyen bölgelerine görsel sinyaller gönderecek şekilde yeniden düzenlediklerinde bu bölgelerde "görmeyi" öğrenebileceklerini keşfetmişlerdir (Von Melchner vd., 2000). Bu, tek bir algoritmanın memeli beyni tarafından gerçekleştirilen birçok farklı görev için kullanılabileceğini göstermekteydi. Makine öğrenimi araştırmaları, biyolojik beynin nöral ağları aracılığıyla öğrenmesini taklit eden derin öğrenme modellerinin ortaya çıkmasıyla birlikte daha bütünleşik bir yapıya kavuştu [30].



Şekil 2.8: Derin Ağların Beş Yıllık Verileri.

ImageNet büyük ölçekli görsel tanıma yarışmasını, katılmasına izin veren bir boyuta ulaştığından beri, her yıl, düşük bir hata oranıyla tutarlı bir şekilde kazanmaya devam etmiştir. Veriler Russakovsky et al. (2014b) ve He et al. (2015). Derin öğrenme, 2007 yılında insanların üstün olduğu işitme ve görme gibi algısal problemlerde umut verici sonuçlarla ön plana çıktı ancak uzun zamandır makineler için ince bir varlıktı. Sonuç olarak bilgisayar bilimcileri, bilgisayarlara fotoğraflardaki nesneleri bulmak, kelimeleri ve cümleleri tanımak ve ses, video gibi büyük miktarda veri kullanarak belgeleri başka dillere çevirmek gibi insanlar için doğal ve sezgisel olan şeyleri yapmayı öğretmek için büyük bilgi işlem gücünden yararlanabilmektedirler. Bir bilgisayar bunu ilk kez yapabiliyordu [31]. Veri, olasılıkların denizidir. Deniz ne kadar derin olursa, olasılıklar da o kadar geniş olur. Örneğin bir komut dosyası düzenlemek, spam olup olmadığını belirlemek veya bir müşterinin kredi ödemeye istekli olup olmadığını belirlemek için ses tanımayı kullanmak gibi.

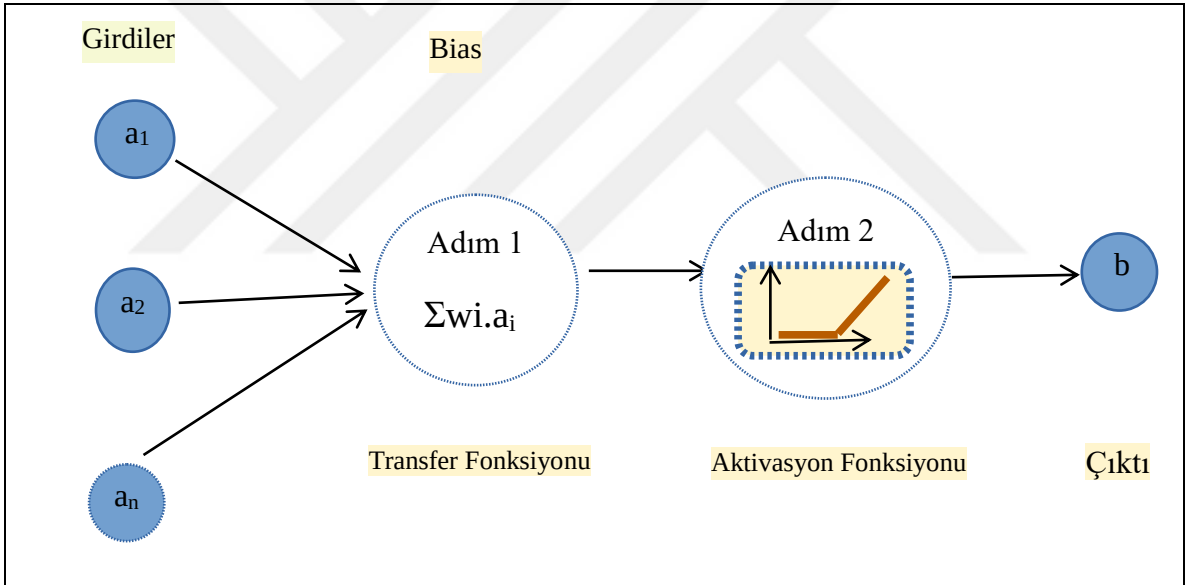
2.2.1 Derin Öğrenmenin Çalışma Stili

Derin ağlar bir dizi çok seviyeli dönüşümle girdileri hedeflerle eşleştirir ve bu çok seviyeli dönüşümler eğitim örneklerine maruz bırakılarak öğrenilir. Girdiye bir seviye uygulayan dönüşüm, esasen bir sayı kümesi olan seviyenin ağırlığı ile belirlenir. Bir katman, ağırlıkları aracılığıyla gerçekliği nasıl yorumlayacağını belirler. Ağırlıklar ne kadar yüksekse, katman o kadar güçlü bir etki uygular. Bu, katmanın gerçekliği nasıl gördüğünü ve bu görüşe dayalı olarak nasıl kararlar verdiğini etkiler. [32].

Derin öğrenme, bir yolculuktur. Yolculuğun amacı, ağırlıklarını doğru değerlere

doğrultmaktır. Bu yolculuk, binlerce parametre arasında doğru yolu bulmak gibi zor bir iş olabilir. Ancak, doğru yoldan sapmadan ve doğru ağırlıkları bularak, derin öğrenme ağıları karmaşık görevleri yerine getirmeyi öğrenebilir. Bu durum özellikle bir parametrenin değeri diğerinin değerini etkilediğinde geçerlidir [32].

Bir derin ağı eğitmek için, ağı hedefe ne kadar yakın olduğunu bilmek gerekir. Bu bilgi, ağı hedefe doğru ilerlemesini sağlamak için kullanılır. Kayıp fonksiyonu, belirli bir durum için öngörülen ağ çıktısı ile gerçek hedef değer arasındaki farkı hesaplar. Bu fark ne kadar büyükse, ağı o kadar yanıldığı anlamına gelir. Ağı yanıldığı anlaşılarak, ağırlıklarını güncelleyerek ağı doğru çıktıyı üretmesini sağlayabiliriz. Bu, ağı belirli bir örneği ne ölçüde öğrendiğinin değerlendirilmesini sağlar. Öğrenme, ağı hatalarını düzeltme sürecidir. Bu süreç, ağı ağırlıklarını güncelleyerek gerçekleştirilir. Ağırlıklar, ağı çıktısını etkileyen parametrelerdir. Ağırlıkları güncelleyerek, ağı çıktısını daha doğru hale getirebiliriz [33].



Şekil 2.9: Derin Öğrenme Nöral Girdiler.

Daha güçlü bir model oluşturmak için birden fazla doğrusal olmayan katman ekleme genel fikri, derin öğrenme olarak adlandırılmasa bile, her zaman sinir ağı araştırmalarının arkasındaki itici güç olmuştur.

Derin öğrenmenin temellerini anlamak için, tek katmanlı doğrusal regresyondan başlayalım. Bu model, derin öğrenmenin temelini oluşturan kavramları anlamamıza yardımcı olacaktır. İnsanlar bu mimariyi öğrenmekte zorlandılar ve bu durum, o dönemde makinelerin sınırlı hesaplama gücüyle birleşince, destek vektör makineleri (SVM'ler) gibi diğer yöntemlerin

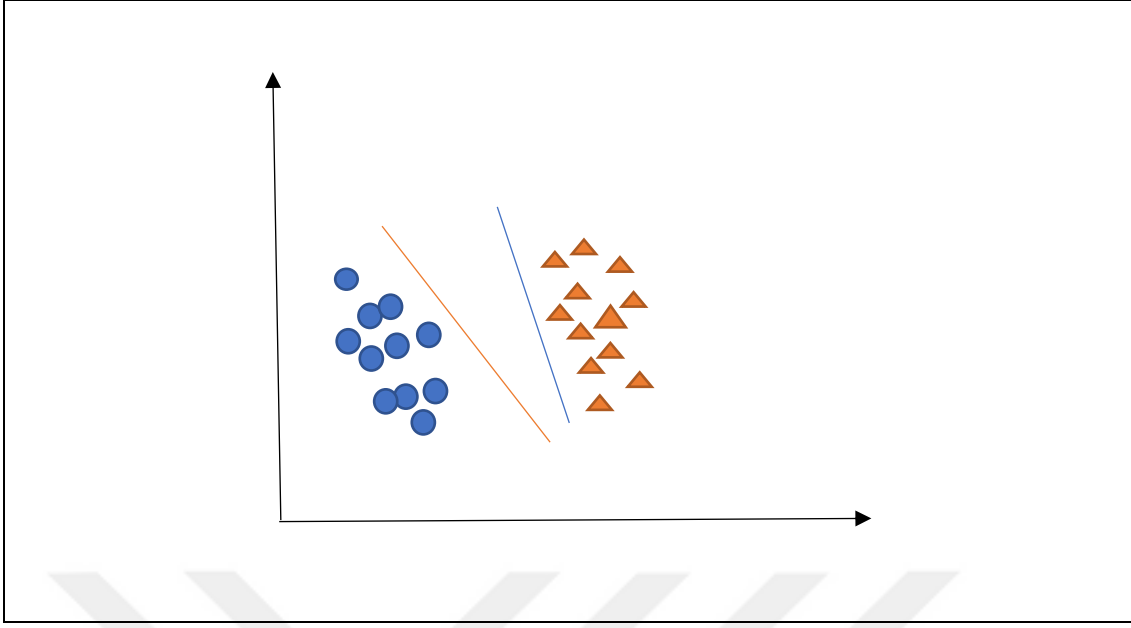
örüntü tanıma sahnesine hakim olmasına yol açtı [34].

Sinir ağlarının görüntü sınıflandırma, segmentasyon ve özellik öğrenme gibi çeşitli uygulamalar için tercih edilen yöntem haline gelmesi yaklaşık yirmi yılı aşkın süre içinde birçok dikkate değer sonuçlar ortaya çıkarmıştır. Bu sonuçların merkezinde SVM'lerden farklı olarak özellikleri öğrenme yeteneği vardır. Özellikle görüntü sınıflandırmada, giriş piksellerinin tek başına temsili çoğu uygulama için makul değildi ve iyi bir görüntü tanımlayıcı kümesi bulunmalıydı.

Nihai derin öğrenme modeli hiç şüphesiz konuşma tanıma gibi görevlerde dizi modelleme için tercih edilen tekrarlayan ağ olarak tanımlanmaktadır [Werbos, 1974; Williams ve Zipser, 1989; Schmidhuber, 1992; Graves, 2008].

Makine öğrenimi, veri dünyasını keşfetmenin yeni bir yoludur. Bu keşfi mümkün olduğunca kolaylaştırmak için, makine öğrenimi yöntemlerinin uygulanmasını kolaylaştırmak gerekir. Bu şekilde, giriş verilerinden bağımsız olarak özellik öğrenimi sonrasında tüm geleneksel makine öğrenimi yöntemleri uygulanabilir olacaktır [35]. Örneğin, SVMler bir özellik çıkarma adımından geçer. İyi özellikler size güçlü sınıflandırma sonuçları verebilir ancak eksik veya kötü özellikler hiçbir şey ifade etmeyecektir. Yakın bir gelecekte, derin öğrenmenin benzersiz ve yenilikçi şekillerde uygulandığını göreceğiz. Bu uygulamalar, günlük hayatımızın çeşitli alanlarında faydalı olmalarını sağlamak için gerçekçi mekaniklerle birleştirilmiş daha gizli zeka (öneriler ve ses tanıma gibi) çeşitliliğine sahip olacaktır [35].

Derin öğrenme ağları, bir yandan birçok avantaj sağlarken, diğer yandan da birçok sorun ve zorluk yaratır. Bu sorunlar ve zorluklar, derin mimarilerin avantajlarından daha ağır basmaktadır. Bu mimariyi genel kullanım için eğitmek acı verici derecede yavaştır. Sınırlı bilgi işlem gücü nedeniyle derin öğrenme ağları, çekirdek yöntemi gibi diğer yaklaşımlardan daha iyi performans göstermiştir. Hesaplama gücünün (özellikle GPU'lar ve dağıtık hesaplama) muazzam artışı ve büyük etiketli veri kümelerine erişim, yeniden canlanmasının önünü açmıştır. Ancak bu alandaki büyük ilerlemelere rağmen çok sayıda serbest parametreye sahip derin modelleri eğitmek, karmaşık ve uygunsuz bir optimizasyon problemidir. Derin mimariler için verimli eğitim yöntemleri oluşturmaya yönelik çok sayıda araştırma çalışması yapılmıştır [36].



Şekil 2.10: Support Vektor Machine.

2.2.2 CNN (Evrişimli Sinir Ağları)

ConvNets veya CNN'ler olarak da bilinen Evrişimli Sinir Ağları, birçok katmandan oluşan derin öğrenme teknikleridir. ConvNets, biyolojik görsel korteksin nasıl çalıştığına dair bir modeldir. Beyin, görsel bilgiyi işlemek için küçük hücre alanlarını kullanır. Bu hücreler, görme alanının belirli bölgelerine duyarlıdır. Örneğin, bazı nöronlar yalnızca belirli bir yönde bir çizgi olduğunda ateşlenir, bazı nöronlar ise dikey bir kenara çarptığında ateşlenir ve diğerleri yatay veya çapraz bir kenar görüldüğünde ateşlenir. ConvNets, beyin gibi bir simülasyondur. Beyin, belirli görevleri yerine getirmek için nöronları gruplar halinde kullanır. fıkriConvNets de aynı şekilde çalışır. [37].

ConvNets, bilgisayarla görmenin geleceğini şekillendiriyor. Görüntüleri tanımak için insan beynini taklit ederek, kendi kendine giden arabaların, robotların ve körlerin tedavisinin daha güvenli ve etkili hale gelmesini sağlıyor.

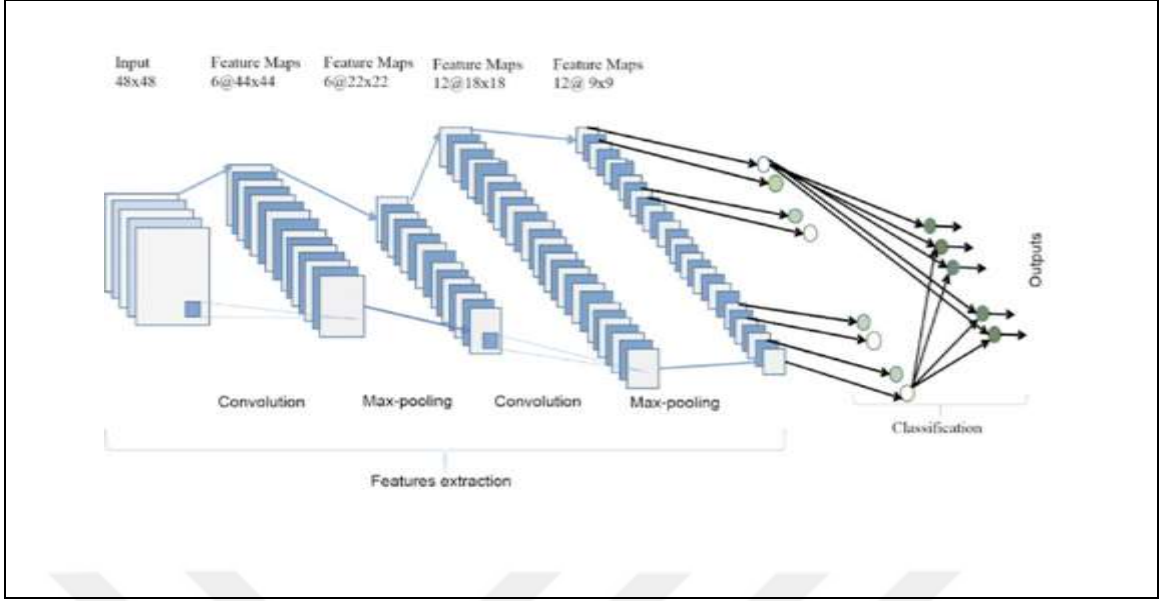
ConvNets, bir görüntüyü, daha küçük parçalara ayırarak ve bunları daha sonra yeniden birleştirerek anlamaya çalışır. Bu süreç, üst katmanlardan başlayarak ve alt katmanlara doğru ilerleyerek gerçekleşir. Üst katmanlar, görüntünün en temel özelliklerini öğrenirken, alt katmanlar daha karmaşık özellikleri öğrenir. Bununla birlikte, çok katmanlı mimarisi onu hesaplama açısından yoğun hale getirir ve böyle bir ağı büyük bir veri kümesi üzerinde eğitmek günler alır [38].

CNN'in amacı, konvolüsyon yoluyla verilerdeki yüksek dereceli özellikleri incelemektir. Görüntüleri kullanarak nesne tanıma konusunda özellikle uygundur ve sürekli olarak en iyi görüntü sınıflandırma yarışmalarını kazanmışlardır. BU sayede yüzler, insanlar, işaretler ve platypus gibi görsel verilerin farklı yönleri tespit edilebilir. CNN, metin analizini görsel karakter tanıma ile örtüştürür ancak CNN mimarisi kelimeleri ayrı metin birimleri olarak analiz etmek için de yararlıdır. Ayrıca ses analizi konusunda da etkilidir.

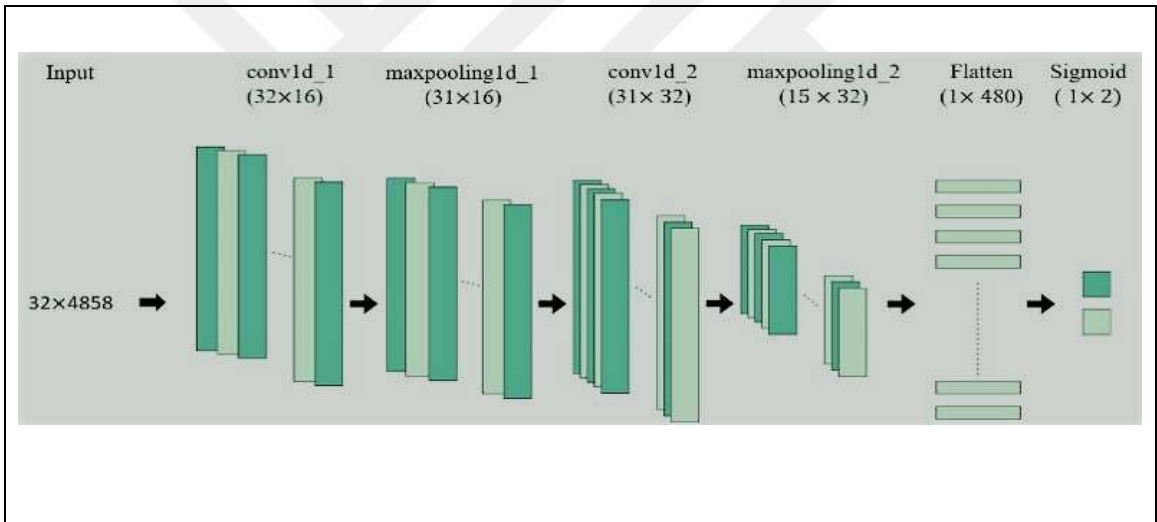
CNN'ler, görüntü tanımda insan seviyesinde veya daha iyi performans göstererek, derin öğrenmenin gerçek dünyadaki etkilerini gösterdi. Bu, insanların derin öğrenmenin gücünü fark etmelerinin ana nedenlerinden biridir. Şekil 2.11'de gösterildiği gibi, CNN'ler ham görüntü verilerinden konumla değişmeyen ve (bir dereceye kadar) rotasyona dayalı özellikler oluşturmak için uygundur. CNN; sürücüsüz arabalar, robotik, insansız hava araçları, körlerin bakımı ve daha birçok alanda net uygulamalarla makine görüşü alanındaki büyük ilerlemelerin arkasındadır.

CNN'ler, girdinin bir modelini oluşturmak için en iyisidir. Bu model, girdinin düzenini ve özelliklerini yansıtır. Bu bilgi, girdiyi anlamlandırmak ve sınıflandırmak için kullanılabilir. Görüntü ve ses verilerinin belirli bir yinelenen desen ve yan yana bir dizi girdi değeriyle uzamsal olarak nasıl ilişkilendirildiği buna bir örnektir. CNN'ler ayrıca doğal dil çevirisi/üretimi ve duygu analizi gibi diğer görevler için de kullanılmaktadır. Konvolüsyon, daha sağlam bir sinyal tabanlı özellik uzayı oluşturmaya yardımcı olan güçlü bir kavramdır.

CNN, giriş verilerini bir dizi katman aracılığıyla sınıf puanlarına dönüştürür. Bu katmanlar, girdinin özelliklerini ve kalıplarını öğrenmek için birlikte çalışır. İlk katmanlar, görüntünün temel özelliklerini öğrenirken, sonraki katmanlar daha karmaşık özellikleri öğrenir. Çıkış katmanı, tüm bu bilgileri bir araya getirerek, görüntünün olası sınıflarını tahmin eder. Şekil 2.12'de gösterilen katman modeline dayanmaktadır [39].



Şekil 2.11: Evrişimli Sinir Ağları Mimari Yapısı.



Şekil 2.12: Üst Düzey Genel CNN Mimarisi.

Şekil 2.12’de 3 katman gösterilmiştir. Bunlar:

- Input yani Giriş Katmanı
- Öğrenme (Learning) Katmanları
- Output Çıkış katmanı

Giriş katmanı, genellikle görüntünün uzamsal boyutu olan (genişlik × yükseklik) biçiminde üç boyutlu girişi kabul etmektedir ve renk kanallarını temsil eden bir derinliğe sahiptir. RGB renk kanalları için genellikle üç katmanlıdır.

Öğrenme (learning) katmanları ise ortak bir tekrar eden dizi modeline sahiptir.

a. Evrişim katmanı

Doğrultulmuş Doğrusal Birim (ReLU) aktivasyon fonksiyonunu diğer literatürle eşleştirmek için buradaki diyagramda bir katman olarak ifade ediyoruz.

b. Havuzlama (pool layer)

Bu katmanlar bir görüntüdeki çok sayıda özelliği tespit eder ve kademeli olarak üst düzey özellikler oluşturur. Geleneksel manuel mühendisliğin aksine özelliklerin otomatik olarak öğrenildiği devam eden derin öğrenme konusunu doğrudan ele alır.

Son olarak, yüksek dereceli özellikleri alan ve olasılıklar veya sınıf puanları üreten bir veya daha fazla tam bağlı katmana sahip bir sınıflandırma katmanı vardır. Bu katmanlar, bir önceki katmanın tüm nöronlarıyla iletişim halindedir. Bu, her nöronun, bir önceki katmanın tüm deneyimlerine erişebileceği anlamına gelir. Bu katmanlardan elde edilen çıktı tipik olarak $[b \times N]$ boyutlarında iki boyutlu bir çıktı üretir. Burada b mini partideki örnek sayısı ve N değerlendirmek istediğimiz sınıf sayısıdır.

Bir evrişimsel sinir ağında, her seviyedeki durum uzamsal bir ızgara yapısına göre düzenlenir. Görüntüdeki yerel özellikler, bir sonraki katmana aktarılır. Bu, her bir özelliğin, bir önceki katmanın küçük bir parçasını temsil etmesi nedeniyle [40].

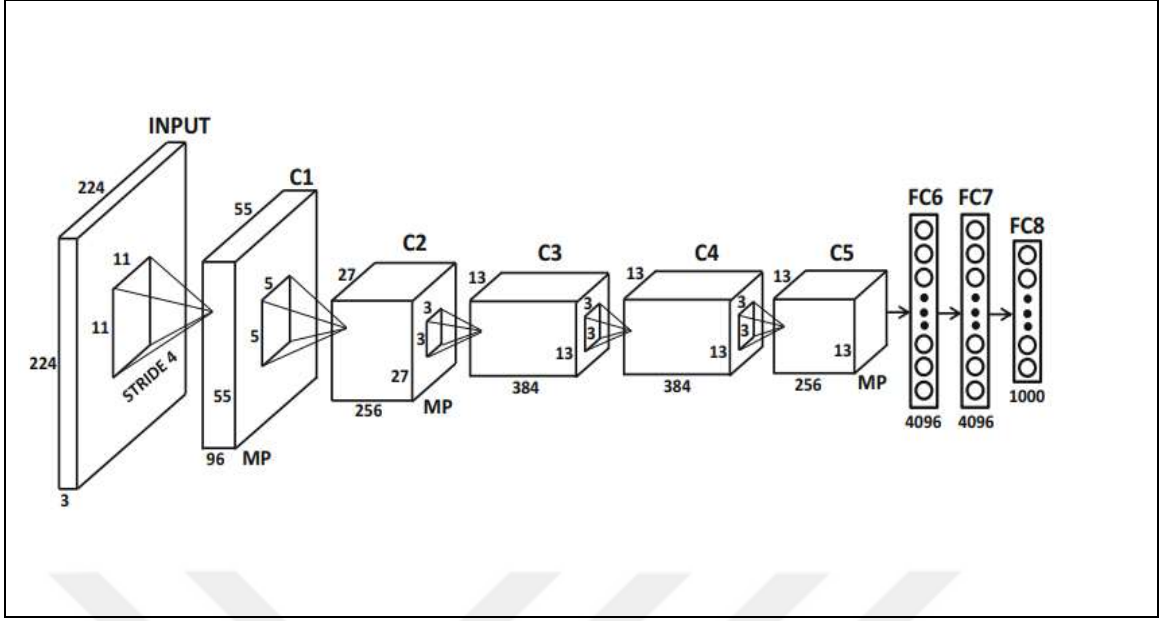
2.2.2.1 Evrişim katmanı (Convolution layer)

Konvolüsyon, iki bilgi kümesinin nasıl birleştirileceğine ilişkin kuralları tanımlayan matematiksel bir işlem olarak tanımlanır. Fizik ve matematikte önemli olan Fourier dönüşümünü, uzay/zaman ile frekans alanı arasındaki köprüyü tanımlamak için kullanır. Girdiyi alır, bir konvolüsyon çekirdeği uygular ve çıktı olarak bir özellik haritası sağlar [41]. Konvolüsyonel katman, görüntüdeki yerel özellikleri çıkarmak için bir filtre kullanır. Bu filtre, bir önceki katmanın küçük bir bölgesine uygulanır ve bu bölgenin özelliklerini temsil eden bir çıktı üretir. Parametreler, görüntüdeki yerel özellikleri tanımak için kullanılan bir dizi filtredir. Bu filtreler, görüntüdeki benzerlikleri ve kalıpları tanımlamak için kullanılır. Bu bilgiler, görüntünün daha yüksek seviyeli özelliklerini öğrenmek için kullanılır. Evrişimsel katmanın ana görevi, giriş görüntüsünün yerel bölgesinde bulunan ve tüm veri kümesi için ortak olan özellikleri tespit etmek ve bunların görünümünü bir özellik haritasına

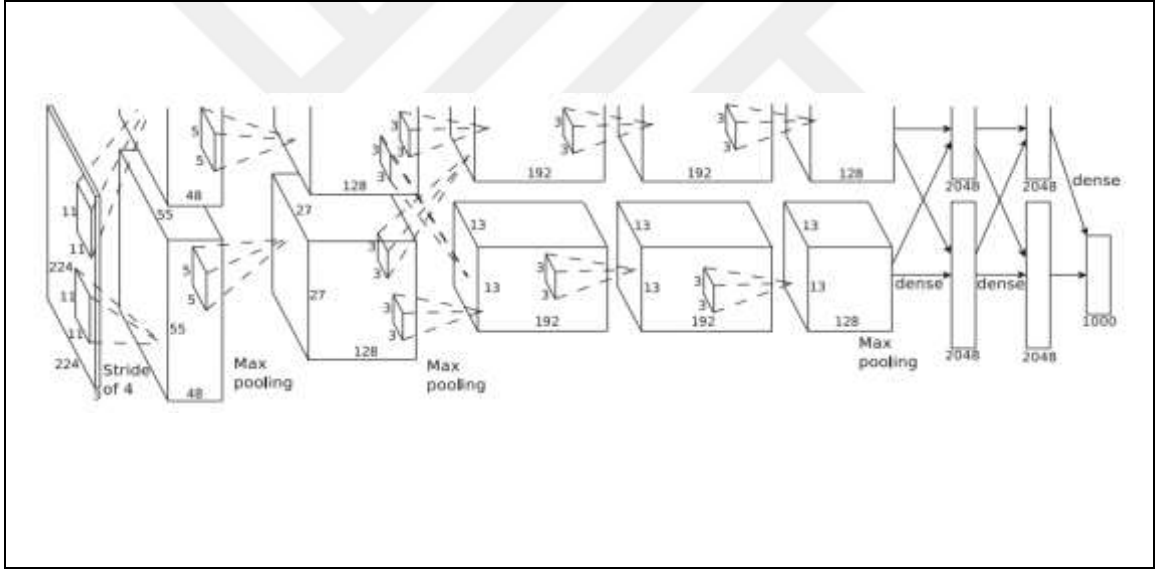
eşlemektir.

Alexnet

AlexNet, 2012 ILSVRC yarışmasında bir devrim yaratarak, görüntü tanıma alanında yeni bir çağır açtı. AlexNet mimarisinin, bu devrimin temelini oluşturduğu düşünülmektedir. Şekil 2.13(a), AlexNet mimarisinin yalnızca bir parçasıdır. Orijinal mimaride, Şekil 2.13(a)'da gösterilmeyen iki paralel işlem yolu daha vardır. Bu yollar, AlexNet'in daha hızlı ve daha verimli çalışmasını sağlar. İki GPU daha hızlı bir öğrenme modeli oluşturmak ve belleği paylaşmak için birlikte çalıştığında iki boru hattı oluşur. AlexNet, başlangıçta 3 GB belleğe sahip bir GTX 580 GPU üzerinde eğitildi. Ancak, ağ büyüdükçe, bu kapasite yetersiz kaldı. Bu nedenle ağ iki GPU'ya bölünmüştür. Şekil 2.13(b) iki GPU'ya bölünmüş orijinal mimariyi göstermektedir. Ayrıca bu şekil, bu bölümde tartışılan diğer konvolüsyon sinir ağı mimarileriyle karşılaştırmayı kolaylaştırmak için GPU'ların neden olduğu değişiklikler olmadan mimariyi göstermektedir.



(a) GPU Bölümlenmesi Yok.



(b) GPU Bölümlenmesiyle (orijinal mimari)

Şekil 2.13: AlexNet Mimarisi.

ReLU aktivasyonu her bir konvolüsyon seviyesini takip eder ve açıkça gösterilmemiştir. Maksimum konsolidasyon katmanının MP olarak etiketlendiğine ve konvolüsyonel-ReLU kombinasyon katmanının sadece bir kısmını takip ettiğine dikkat ediniz. (b)'deki mimari diyagram AlexNet, Krizhevsky, Sutskever ve Hinton tarafından 2012 NIPS konferansında yayınlanan bir makalede tanıtıldı. Makalede, AlexNet'in, görüntü tanımda devrim niteliğinde bir başarı elde ettiği gösterildi.

Şekil 2.13(b)'de GPU'ların yalnızca bazı katmanlarda birbirine bağlı olduğunu ve gerçekte inşa edilen modelde Şekil 2.13 (a) ile (b) arasında küçük bir fark olduğunu belirtmek gerekir. Özellikle GPU'ya bölünmüş mimaride, tüm katmanlar birbirine bağlı değildir ve bu da daha az ağırlığa neden olur. Ara bağlantıların bazılarının kaldırılması işlemciler arasındaki iletişim süresini azaltarak verimliliğe katkıda bulunur. AlexNet 224 x 224 x 3 boyutunda bir görüntü ile başlar ve ilk katmanda 11 x 11 x 3 boyutunda 96 filtre kullanır. Adım sayısı 4'tür. Sonuç olarak, bu ilk katmanın boyutu $55 \times 55 \times 96$ olur. İlk katman hesaplandıktan sonra maksimum havuzlama katmanı kullanılır. Bu katman, Şekil 2.13(a)'da 'MP' olarak gösterilmiştir. Şekil 2.13(a)'daki mimari, iki paralel boru hattını açıkça gösteren Şekil 2.13(b)'nin basitleştirilmiş bir versiyonudur. AlexNet, GPU'nun paralel işlem yeteneklerini en üst düzeye çıkarmak için, ilk konvolüsyon katmanının derinliğini 96'dan 48'e düşürmüştür. Öte yandan Şekil 2.13(a), GPU kullanımını gerektirmez, bu nedenle genişliği açıkça 96 olarak ayarlanmıştır. ReLU aktivasyon fonksiyonu her konvolüsyon seviyesinden sonra uygulanır, ardından yanıt normalizasyonu ve maksimum havuzlama yapılır. Maksimum havuzlama diyagramda belirtilmiştir ancak hiçbir mimari blok tahsis edilmemiştir. ReLU ve yanıt normalleştirme katmanları, diyagramda görünmez bir şekilde çalışır. Bu kısalık, sinir mimarilerinin görüntü temsillerinde yaygındır.

Bu mimaride kullanılan tasarım tercihlerinin birçoğu daha sonraki mimarilerde standartlaştırılmıştır. AlexNet, ReLU aktivasyon fonksiyonları, veri büyütme, GPU kullanımı ve dropout gibi birçok yeni teknik ve kavramın benimsenmesine yol açarak, konvolüsyonel sinir ağları alanında devrim niteliğinde bir başarıya imza attı. Bu teknikler, görüntü tanımanın doğruluğunu ve verimliliğini önemli ölçüde artırarak, görüntü tanımanın geleceğini şekillendirdi. Dropout, çoğu durumda ek bir güçlendirici olduğu için günümüzde neredeyse tüm mimari türlerinde yaygın olarak kullanılmaktadır. Yanıt normalizasyonu, konvolüsyonel sinir ağlarının (CNN) gelişiminde önemli bir rol oynadı, ancak daha sonra daha verimli ve güçlü tekniklerle değiştirildi [42][43].

Alexnet'i geliştirmek için yapılan düzeltmenin nedenleri aşağıda sıralanmıştır:

- a. Toplam 22.000'den fazla kategoriden yaklaşık 15 milyon görüntü etiketi içeren büyük etiketli görüntü veritabanı (ImageNet) kullanılmıştır.
- b. Model, GTX 580 GPU'nun hızını kullanarak, yaklaşık 6 gün boyunca yoğun bir eğitime tabi tutuldu.

- I. ReLU (Rectified Linear Unit) basit ama güçlü bir aktivasyon fonksiyonu olan $f(x) = \max(x, 0)$ kullanılmıştır. ReLU, sigmoid ve tanh gibi geleneksel aktivasyon fonksiyonlarından farklı olarak, negatif değerlere sıfır döndürür. Bu, ReLU'nun hesaplamasını basitleştirir ve GPU'larda daha hızlı bir şekilde yürütülmesine olanak tanır. Ayrıca, ReLU'nun nöronların aşırı uyarılmasını önleyerek modelin kararlılığını artırmasına yardımcı olur. ReLU aktivasyon fonksiyonu kaybolan gradyan problemi gibi sorunlarla karşılaşmaz[42].

AlexNet, görüntülerdeki nesneleri tanımak için, 5 konvolüsyonel katman, 3 havuzlama katmanı, 3 tam bağlı katman ve 1000 softmax sınıflandırıcıdan oluşan, karmaşık bir sinir ağı mimarisidir.

ZFNet

ZFNet'in [44] bir varyantı 2013 ILSVRC yarışmasını kazanmıştır. ZFNet, AlexNet'in üzerine inşa edilmiş, doğruluğu artıran bir mimaridir. AlexNet'in bir nevi süper kahramanı gibidir. Bu değişikliklerin çoğu hiperparametrelerin seçimindeki farklılıkları içerir ve bu nedenle ZFNet, temel düzeyde AlexNet'ten önemli ölçüde farklı değildir; AlexNet'ten ZFNet'e yapılan bir değişiklik, orijinal $11 \times 11 \times 3$ filtrenin $7 \times 7 \times 3$ filtre olarak değiştirilmesidir. Buna ek olarak dört adım yerine iki adım kullanıyoruz. İkinci katmanda 2. adımda 5×5 filtre kullanır. AlexNet'te olduğu gibi, aynı boyutta maksimum konsantrasyon filtrelerine sahip üç maksimum konsantrasyon katmanı vardır. Ancak maksimum konsantrasyon katmanlarının ilk çifti birinci ve ikinci konvolüsyondan sonra gerçekleştirilir (ikinci ve üçüncü konvolüsyon yerine). Sonuç olarak üçüncü katmanın uzamsal ayak izi, 27×27 'den 13×13 'e değişirken diğer tüm uzamsal ayak izleri, AlexNet'e göre değişmeden kalır. AlexNet ve ZFNet katmanlarının boyutları Tablo 2.1'de gösterilmektedir. ZFNet, AlexNet'in konvolüsyonel katmanlarına daha fazla filtre ekleyerek, daha ayrıntılı özellikler öğrenmesini sağlamıştır.

Tablo 2.1: Karşılaştırmalı AlexNet ve ZFNet Tablosu.

	<i>AlexNet</i>	<i>ZfNet</i>
Volume Operations	224 x 224 x 3 Conv 11 x 11 (stride 4)	224 x 224 x 3 Conv 7 x 7 (stride 2), MP
Volume Operations	55 x 55 x 96 Conv 5 x 5, MP	55 x 55 x 96 Conv 5 x 5 (stride 2), MP
Volume Operations	27x 27 x 256 Conv 3 x 3, MP	13 x 13 x 256 Conv 3 x 3
Volume Operations	13x 13 x 384 Conv 3 x 3, MP	13 x 13 x 512 Conv 3 x 3
Volume Operations	13x 13 x 384 Conv 3 x 3, MP	13 x 13 x 1024 Conv 3 x 3
Volume Operations	13x 13 x 256 MP, fully connect	13 x 13 x 512 MP, fully connect
FC6: Operations	4096 Fully connect	4096 Fully connect
FC7: Operations	4096 Fully connect	4096 Fully connect
FC8: Operations	1000 softmax	1000 Softmax

AlexNet ve ZFNet, çoğu katmanda aynı uzamsal ayak izlerine sahiptir. Ancak, son üç konvolüsyon katmanında, derinlikleri farklıdır. Bu, ZFNet'in AlexNet'ten daha fazla ayrıntılı özellik öğrenebilmesine olanak tanır. Genel bir bakış açısıyla, ZFNet AlexNet'e benzer prensipler kullanmaktadır ve ana avantajı AlexNet'in mimari parametrelerinin değiştirilmesinden gelmektedir. Bu mimari, ilk beşteki hata oranını %15,4'ten %14,8'e düşürmüş ve aynı yazarlar tarafından genişlik/derinlikte yapılan bir artış hatayı %11,1'e indirmiştir. AlexNet ve ZFNet arasındaki farkların çoğu iyi tasarım seçimleri olduğundan, derin öğrenme algoritmalarla çalışırken küçük ayrıntıların çok önemli olduğu gerçeğini vurgulamaktadır. Bu nedenle, en iyi performansı elde etmek için nöral mimarilerle kapsamlı deneyler yapmak önemlidir. ZFNet'in mimarisi, daha fazla filtre ve daha fazla katman ekleyerek, daha güçlü hale getirilmiştir. Bu yeni mimari, 2013 yılında ILSVRC yarışmasında, Clarifai adı altında sunulmuştur. Bu şirket, ZFNet'in ilk yazarı tarafından kurulmuştur. Clarifai ve ZFNet arasındaki fark ağın genişliği/derinliği ile ilgiliydi fakat bu farkların kesin ayrıntıları mevcut değildi. 2013 yılında ILSVRC yarışmasında, ZFNet'in girişi, diğer tüm girişleri geride bıraktı ve birinci oldu. Detaylar ve mimarinin görsel bir temsilini görmek için [44] numaralı kaynağa başvurabilirsiniz.

VGGNet

VGG [45], artan ağ derinliği açısından büyüyen eğilimi daha da vurgulamaktadır. Test edilen

ağlar, 11 ila 19 katman arasında boyutlara sahip çeşitli konfigürasyonlarda tasarlanmıştır ve en iyi performans gösteren versiyon, 16'dan fazla katmana sahiptir. VGG, ISLVR 2014'te en iyi performansı göstermiş ancak yarışmayı kazanamamıştır. Kazanan, GoogLeNet, %7,3'lük hata oranının aksine %6,7'lik bir hata oranına sahipken, VGG gelecekteki mimariler için kritik tasarım ilkelerini ortaya koyarak önemli bir referans noktası oluşturmuştur.

VGG'nin önemli yenilikleri filtre boyutunun küçültülmesi ve derinliğin artırılmasıdır. Daha da önemlisi, filtre boyutu küçüldükçe derinlik de artmaktadır. Küçük filtreler, görüntünün yalnızca küçük bir alanını yakalayabilir. Bu nedenle, ızgara derin değilse, küçük filtreler görüntünün tamamını anlayamaz. Örneğin, 3×3 boyutundaki üç ardışık konvolüsyonun sonucu olan tek bir özellik, 7×7 boyutundaki bir giriş bölgesini yakalar. 7×7 giriş bölgesinin görsel özellikleri, doğrudan giriş verileri üzerinde tek bir 7×7 filtre kullanılarak da yakalanabilir. İlk durumda $3 \times 3 \times 3 = 27$ parametre kullanılırken, ikinci durumda $7 \times 7 \times 1 = 49$ parametre kullanılır. Bu nedenle, üç ardışık konvolüsyon kullanıldığında parametrelerin kapladığı alan daha küçüktür. Tek bir konvolüsyon, görüntünün yüzeyini görebilir. Ancak, üç ardışık konvolüsyon, görüntünün derinliklerini ve karmaşıklıklarını keşfedebilir. Bu nedenle, 7×7 filtreli bir ağ, küçük bölgelerin ince ayrıntılarını gözden kaçırabilir.

Derinlik arttıkça, modeller daha karmaşık ve düzensiz özellikler öğrenmeye başlar. Daha derin ağlar, daha fazla ReLU katmanı nedeniyle daha fazla doğrusal olmayanlığa ve konvolüsyonların yinelemeli sentezini kullanarak katmanların derinliğini artıran artan yapı nedeniyle daha fazla düzenliliğe sahip olacaktır. Derinlik ve filtre boyutu, bir ağın parametre sayısı üzerinde zıt etkilere sahiptir. Derinlik, parametre sayısını doğrusal olarak artırır, ancak daha küçük filtre boyutları, parametre sayısını kare olarak azaltır. Bu, daha derin ve daha küçük filtre boyutlarına sahip mimarilerin, daha az parametreye sahip olmasına neden olur. Bu nedenle, filtre boyutu azaltılarak parametre sayısı önemli ölçüde azaltılabilir ve derinlik artırılarak parametreler "tüketilebilir". Derinliğin artırılması aynı zamanda daha fazla doğrusal olmayan tetikleyicinin kullanılmasına olanak tanıyarak modelin ayırt edici gücünü artırır. VGG, küçük bölgelerin ince ayrıntılarını yakalamak için 3×3 ve 2×2 lik filtreleri tercih eder.

Konvolüsyon 1. adımda gerçekleştirilir ve 1 dolgu için kullanılır. Yoğunlaştırma her zaman uzamsal ayak izini sıkıştırır ancak 1 dolgulu 3×3 filtre kullanılarak çıktı hacminin uzamsal

ayak izi korunur. VGG, önceki iki mimarinin aksine örtüşmeyen havuzlama kullanır. Bu, görüntünün uzamsal boyutunu azaltarak, modelin daha karmaşık özelliklere odaklanmasını sağlar. VGG ayrıca, her maksimum birleştirmeden sonra filtre sayısını genellikle iki kat artırır. Bu, modelin daha fazla ayrıntı yakalamasına olanak tanır. Uzamsal ayak izindeki her 2 kat azalma için derinliği her zaman 2 kat artıran bu tasarım ustalığı, katmanlar arasındaki hesaplama yükünün dengelenmesine olanak tanır ve ResNet gibi daha sonraki mimarilere taşınır.

Derin mimarilerle ilgili bilinen bir sorun, artan derinliğin başlangıçta hassasiyeti ve kararsızlığı artırmasıdır. Bu sorun, önce sığ mimariyi eğiterek ve ardından daha fazla katman eklemek için ön eğitim vererek çözülür. Ön eğitim katman başına değil, 11 katmandan oluşan alt kümeler halinde yapılır; önce 11 katmandan oluşan alt küme eğitilir. Eğitilmiş katmanlar, mimarinin daha derin katmanlarına bir tür "başlangıç noktası" sağlar. VGG, ISLVR yarışmasında en iyi performans gösterenlerden biriydi ve ilk beş arasında sadece %7,3'lük bir hata elde etti, yine de yarışmayı kazanamadı, VGG'nin çeşitli konfigürasyonları Tablo 2.2'de gösterilmiştir. D sütunundaki mimari, yarışmada diğer mimarileri alt ederek birinciliği göğüsledi. Maksimum konsantrasyonun her gerçekleştirilişinde filtre sayısının iki katına çıktığını unutmayın. Bu nedenle, maksimum konsantrasyon haznenin yüksekliğini ve genişliğini 2 kat azaltır, bu da derinlikte 2'lik bir artışla dengelenir. Bir 3×3 filtre ve 1 dolgu ile bir konvolüsyon gerçekleştirmek uzamsal ayak izini değiştirmez. Dolayısıyla, Tablo 2.2'nin D sütunundaki farklı maksimum konsantrasyon seviyeleri arasındaki bölgedeki her bir uzamsal boyutun boyutu (yani yüksekliği ve genişliği) 224, 112, 56, 28 ve 14'tür.

C3D64 terimi, konvolüsyonun 3×3 (ve muhtemelen 1×1) uzamsal boyuta sahip 64 filtre ile gerçekleştirildiği durumu ifade eder. Her katman, kendi özelliklerini yakalamak için özel olarak tasarlanmış filtrelerle sahiptir. Filtreler, katmanın uzamsal boyutunu korumak için tasarlanmıştır ve ReLU aktivasyonu kullanılarak işlenir. M ve LRN, sırasıyla maksimum rezervuar katmanı ve yerel yanıt normalizasyonunu temsil eden simgelerdir. S ve FC4096, sırasıyla yumuşak maksimum katmanı ve 4096 birimli tam bağlı katmanı temsil eden simgelerdir. Filtre sayısı, son katman kümesi hariç, her MAX birleşmesinden sonra daima artırılır. Böylece, uzamsal ayak izindeki azalmaya genellikle derinlikteki bir artış eşlik eder.

Name:	A	A-LRN	B	C	D	E
# Layers	11	11	13	16	16	19
	C3D64	C3D64	C3D64	C3D64	C3D64	C3D64
		LRN	C3D64	C3D64	C3D64	C3D64
	M	M	M	M	M	M
	C3D128	C3D128	C3D128	C3D128	C3D128	C3D128
			C3D128	C3D128	C3D128	C3D128
	M	M	M	M	M	M
	C3D256	C3D256	C3D256	C3D256	C3D256	C3D256
	C3D256	C3D256	C3D256	C3D256	C3D256	C3D256
				C1D256	C3D256	C3D256
						C3D256
	M	M	M	M	M	M
	C3D512	C3D512	C3D512	C3D512	C3D512	C3D512
	C3D512	C3D512	C3D512	C3D512	C3D512	C3D512
				C1D512	C3D512	C3D512
						C3D512
	M	M	M	M	M	M
	C3D512	C3D512	C3D512	C3D512	C3D512	C3D512
	C3D512	C3D512	C3D512	C3D512	C3D512	C3D512
				C1D512	C3D512	C3D512
						C3D512
	M	M	M	M	M	M
	FC4096	FC4096	FC4096	FC4096	FC4096	FC4096
	FC4096	FC4096	FC4096	FC4096	FC4096	FC4096
	FC1000	FC1000	FC1000	FC1000	FC1000	FC1000
	S	S	S	S	S	S

Şekil 2.14: VGG'de Kullanılan Konfigürasyonlar.

Son maksimum konsantrasyon, tam bağlantılı bir katmanın oluşturulmasından hemen önce gerçekleşir ve uzamsal ayak izi daha da azaltılarak 7'ye düşürülür. İlk tam bağlantılı katman, 4096 nöronun her birinin $7 \times 7 \times 512$ nöronlardan gelen girdilerle bağlandığı bir "dev beyin" gibidir. Sinir ağının "bilgisi", bu bağlantılarda kodlanmıştır. Parametreler ve aktivasyonlar için belleğin çoğunun nerede bulunduğu dair ilginç bir alıştırma [46] 'da sunulmuştur. Konvolüsyon sinir ağının ilk kısmı, eğitim sırasında "bellek sınırlayıcısı" görevi görür. Bu, ilk kısmın en büyük uzamsal ayak izine sahip olmasından kaynaklanmaktadır. Bu, ilk kısmın en fazla aktivasyonu ve gradyanı depolaması gerektiği anlamına gelir. Bu, eğitim sırasında bellek kullanımını sınırlayabilir. Bu durum önemlidir çünkü mini yığının gerektirdiği bellek mini yığının boyutuna karşılık gelir. Örneğin, [46] bir görüntü için yaklaşık 93 MB gerektiğini göstermektedir. Dolayısıyla, 128 mini parti boyutuyla gereken toplam bellek, yaklaşık 12 GB'dir. İlk katman, büyük uzamsal ayak izi nedeniyle en büyük belleği gerektirir ancak seyrek bağlantı ve yük dengeleme nedeniyle büyük bir parametre ayak izine sahip

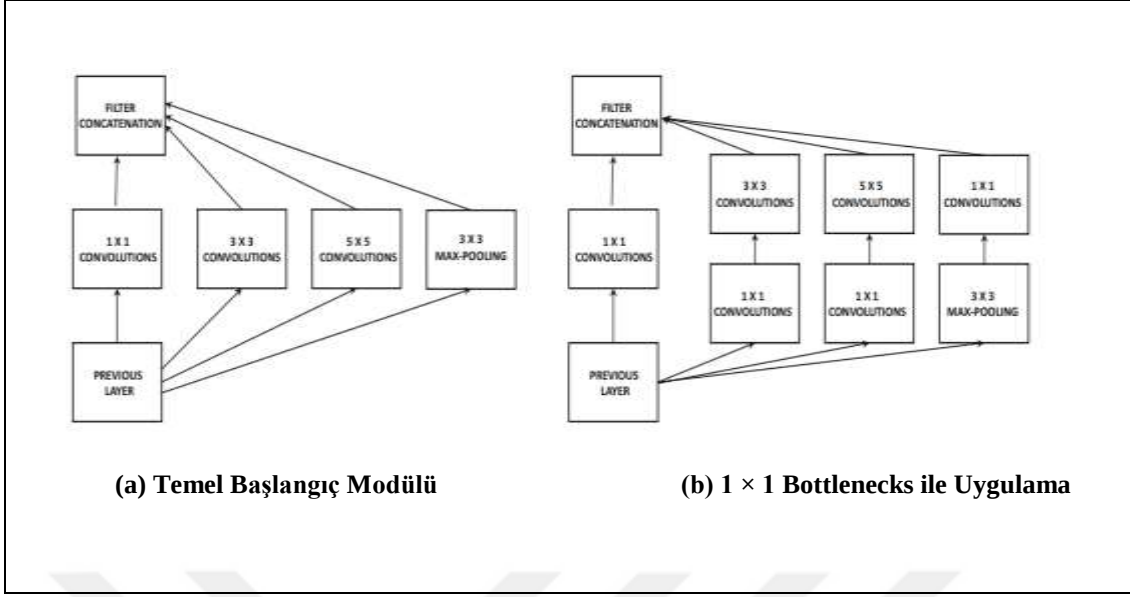
değildir. Konvolüsyon sinir ağının "beyin" kısmı, son iki tam bağlantılı katmanda bulunur. Genel olarak yoğun bağlantılar, sinir ağının parametre ayak izinin %90'ını oluşturmaktadır; bu da GoogLeNet'in son katmanın parametre ayak izini azaltmak için çaba sarf etmesi açısından önemlidir [45].

GoogLeNet

GoogLeNet ise önyükleme mimarisi adı verilen yeni bir kavram önermektedir. Önyükleme mimarisi, bir sinir ağının "kökleri"dir. Dış kısmı, geleneksel konvolüsyonel analiz ağlarına benzer ve görüntünün temel özelliklerini yakalar. Bu, iç kısmın, daha karmaşık özellikler yakalamasına olanak tanır. Ağın önemli kısmı, bootstrap modülü olarak adlandırılan orta katmandır. Bir önyükleme birimi örneği, Şekil 2.14(a)'da gösterilmiştir.

Giriş modülü, bir görüntünün "bütünü" ve "parçaları"nı ayırt etmek için kullanılır. Bu, görüntünün önemli bilgilerini farklı ayrıntı seviyelerinde yakalamasına olanak tanır. Büyük bir filtre, sınırlı varyasyonla geniş bir alan üzerinde bilgi elde etmek için kullanılabilirken, küçük bir filtre daha küçük bir alan üzerinde ayrıntılı bilgi elde etmek için kullanılabilir. Bir çözüm, birkaç küçük filtreyi birbirine bağlamaktır, ancak bu, parametreleri ve derinliği boşa harcar. Oysa daha geniş bir desen yeterli olacaktır. Görüntünün her bir alanı için hangi ayrıntı düzeyinin uygun olduğunu belirlemek zordur. Bu nedenle, sinir ağına görüntüleri farklı ayrıntı düzeylerinde modelleme esnekliği vermek önemlidir. Bu, farklı boyutlarda üç filtreyi paralel olarak yerleştiren bir bootstrap modülü ile sağlanır. Bootstrap modülü, görüntülerin farklı "perspektiflerini" yakalamak için üç farklı filtre (1 x 1, 3 x 3 ve 5 x 5) boyutu kullanır.

Aynı boyuttaki filtre dizisi, bir görüntüdeki her şeyi aynı şekilde "görür". Bu, farklı ölçekteki nesneleri yakalamada verimsizdir. Bu nedenle, farklı boyutlarda filtreler kullanarak, bir görüntünün farklı bölgelerini farklı "perspektiflerden" görebiliriz. Bu nedenle, bu mimariden geçmek için çok sayıda yol seçilebilir ve ortaya çıkan özellikler çok farklı uzamsal alanları temsil edecektir. Örneğin, dört adet 3x3 filtreden ve ardından yalnızca bir adet 1x1 filtreden geçmek nispeten küçük bir uzamsal alanı yakalayacaktır. Buna karşılık, çok sayıda 5x5 filtreden geçmek daha büyük bir uzamsal ayak izi ile sonuçlanacaktır.



Şekil 2.15: GoogLeNet'in Başlangıç Modülü.

Bu, farklı gizli özellikler tarafından yakalanan şekillerin ölçeğindeki farklılıkların sonraki katmanlarda büyütüldüğü anlamına gelir. Son yıllarda, toplu normalizasyon, ağ yapısını "sadeleştirerek" orijinal mimarinin bazı sınırlamalarını aştı.

Bir gözlem, başlangıç modülünün farklı boyutlardaki çok sayıda bağlam nedeniyle hesaplama açısından verimsiz olabileceğidir. 1×1 konvolüsyonu, özellik haritasının derinliğini azaltmak için bir "kısaltma" görevi görür. Bu, ağ yapısını daha verimli hale getirir. Örneğin, 1024 kanallı bir özellik haritasını 128 kanallı bir özellik haritasına küçültmek için, 1×1 konvolüsyonu 1024 filtre yerine yalnızca 128 filtre kullanır. Bu, ağ yapısını önemli ölçüde basitleştirir ve eğitim sürecini hızlandırır. Örneğin, giriş derinliğini ilk olarak 256'dan 64'e düşürmek için 64 farklı 1×1 filtre kullanılabilir. Bu ilave 1×1 konvolüsyon, başlangıç modülünde tıkanıklık modu olarak adlandırılır. İlk olarak özellik haritasının derinliğini azaltarak (ucuz bir 1×1 konvolüsyon ile) Darboğaz (Bottleneck) konvolüsyonu uygulandıktan sonra katmanın derinliği azaltılır, böylece daha büyük konvolüsyon nedeniyle hesaplama verimliliğinden tasarruf edilir. 1×1 konvolüsyon, daha büyük bir uzamsal filtre uygulamadan önce denetlenen boyutta bir azalma olarak görülebilir. Tıkanıklık filtresinin parametreleri geriye yayılma sırasında öğrenildiği için bu boyut azaltma kontrollü bir boyut azaltmadır. Tıkanıklık aynı zamanda yoğunlaşma katmanından sonraki derinliğin daha küçük olmasına katkıda bulunur. Tıkanıklık katmanı hilesi, verimliliği ve çıktı derinliğini artırmaya yardımcı olmak için diğer bazı mimarilerde de kullanılır.

GoogLeNet, çıktı katmanı için bir "ortalama" işlevi kullanır. Bu, ağ yapısını daha basit ve etkili hale getirir. Örneğin, 1024 filtrelili bir son aktivasyon haritaları kümesi için, GoogLeNet yalnızca tek bir değer üretir. Bu değer, kümedeki tüm aktivasyon haritalarının ortalamasıdır. Bu, ağ yapısını önemli ölçüde basitleştirir ve eğitim sürecini hızlandırır. Önemli bir gözlem, çoğu parametrenin son evrimsel katman ile ilk tam bağlantılı katman arasındaki bağlantıyla ilgili olmasıdır. Bu tür ayrıntılı bağlantılar, yalnızca sınıf etiketlerini tahmin etmesi gereken uygulamalar için gerekli değildir. Bu nedenle, bir ortalama yaklaşımı kullanılır. Bununla birlikte, ortalama birleşik temsilde uzamsal bilgi tamamen kaybolduğundan, hangi uygulamalarda kullanıldığına dikkat edilmelidir; GoogLeNet'in önemli bir özelliği, parametre sayısının VGG'den çok daha kompakt olmasıdır. Birincisi büyüklük sırasına göre daha az parametreye sahiptir. GoogLeNet'in başarısı, daha sonraki birçok mimaride standart hale gelen ortalama havuzlama yönteminin kullanılmasına dayanmaktadır. Öte yandan GoogLeNet'in genel mimarisi, hesaplama açısından çok zahmetlidir.

GoogLeNet, farklı ölçeklerdeki nesneleri "görme" yeteneğine sahiptir. Bu, 22 katmanlı önyükleme mimarisinin doğasında bulunan uygun filtre boyutlarına dayanır. Bu başlangıç modülünün izin verdiği çok taneli ayrıştırma esnekliği, performansının anahtarlarından biridir. Parametre ayak izi de tam bağlantılı katmanların ortalama konsantrasyonla değiştirilmesiyle önemli ölçüde azaltılmıştır. Bu mimari 2014 ILSVRC yarışmasını kazanmış, VGG ise ikinci olmuştur. GoogLeNet, VGG'den daha iyi performans gösterir ancak ikincisi, profesyonellerin bazen takdir ettiği basitlik açısından bir avantaja sahiptir. Her iki mimari de, konvolüsyonel sinir ağlarının nasıl çalıştığını ve nasıl tasarlanabileceğini anlamamıza yardımcı olan önemli bilgiler sunmaktadır. İlk mimari önemli araştırmaların odağı olmuştur [47, 48] ve performansı artırmak için birçok değişiklik önerilmiştir. Inception-v4, ResNet'ten ilham alarak, daha basit ve etkili bir yapıya sahip 75 seviyeli bir mimariye dönüştürüldü. Bu mimari, ImageNet veri kümesinde %3,08 hata oranıyla yeni bir rekor kırdı.

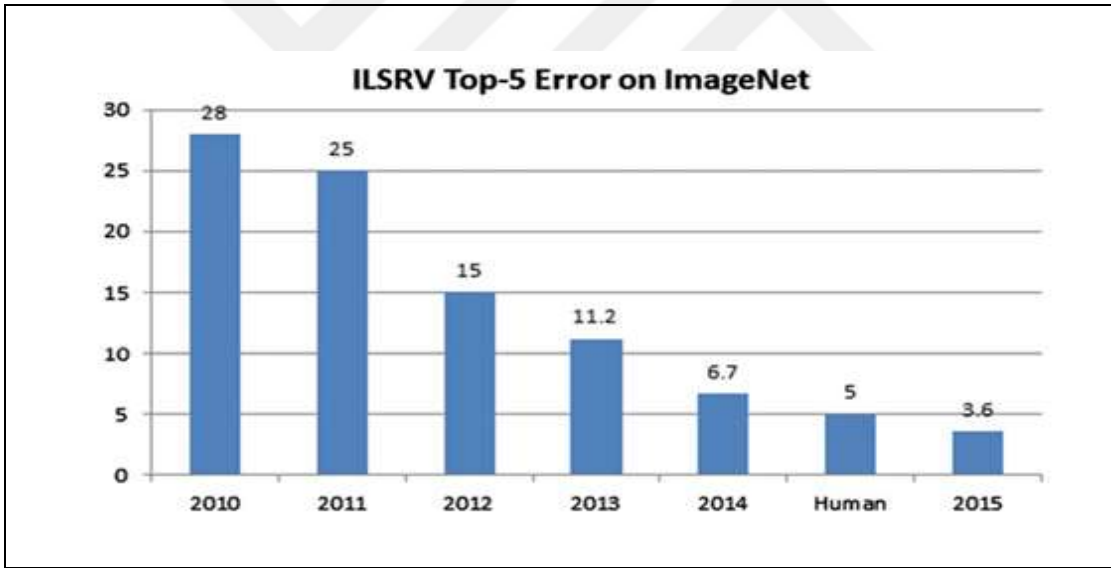
ResNet yapı mimarisi

ResNet[49], derin öğrenmeyi "gökyüzüne fırlattı" ve insan seviyesinin ötesine geçmenin mümkün olduğunu gösterdi. Örneğin, ResNet, %5-10'luk insan hata oranından daha iyi olan %3,6'lık bir hata oranıyla ILSVRC 2015'i kazandı. Bu, derin öğrenmenin, insan gözünden ayırt edilemeyecek kadar doğru görüntü tanıma yapabildiğini gösterdi. ResNet'in

başarısı, derin öğrenmenin yapay zeka alanındaki en önemli gelişmelerden biri olarak kabul ediliyor.

Tablo 2.2: ImageNet'te, Konvolüsyonel Sinir Ağları (AlexNet, Inception, VGG-16, ResNet gibi) Görüntü Tanımanın Yanı Sıra Diğer Birçok Alanda da Devrim Niteliğinde Başarılar Elde Etme Potansiyeli Olduğunu Gösteriyor.

<i>Model</i>	<i>Top-1 accuracy</i>	<i>Top-5 accuracy</i>
<i>AlexNet</i>	0.625	0.86
<i>VGG-16</i>	0.715	0.901
<i>Inception</i>	0.782	0.941
<i>ResNet-152</i>	0.870	0.943
<i>Xception</i>	0.790	0.945



Şekil 2.16: 2010'dan Beri ImageNet Üzerinde ILSRV İlk 5 Hatası Gösterilmektedir.

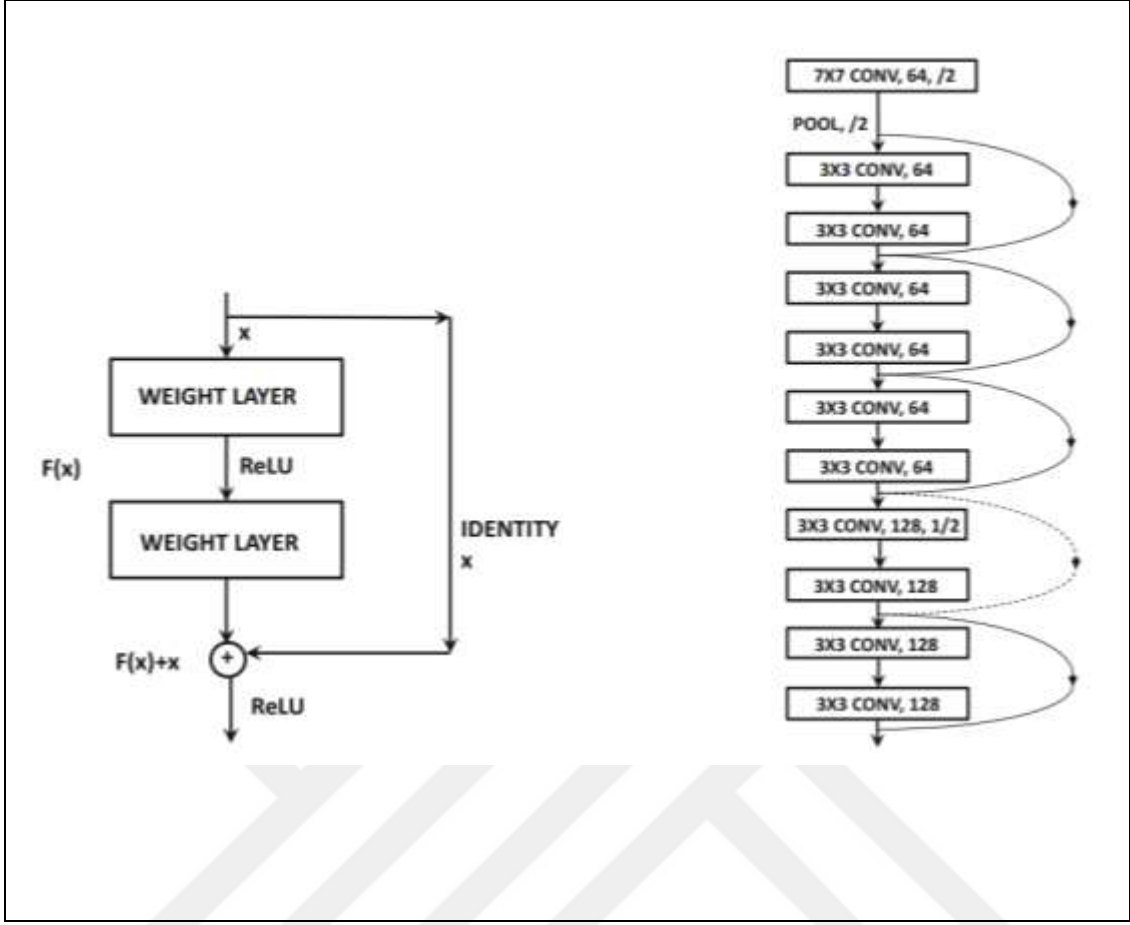
ResNet, daha önce kullanılan diğer mimarilerden neredeyse bir kat daha büyük olan 152 katman kullanır. Bu mimari, 2015 ILSVRC yarışmasını kazanarak ilk 5'te %3,6'lık bir hata elde etmiş ve insan düzeyinde performansa sahip ilk sınıflandırıcı olmuştur. ResNet ağları, derin öğrenmede bir "milyar dolarlık soruyu" cevapladı: 152 katmanlı bir mimariyi eğitmek mümkün mü? Cevap, evet. Ancak, bunun için bazı önemli yenilikler gereklidir. Örneğin, ResNet ağları, kısa bağlantılar adı verilen bir teknik kullanır. Bu teknik, ağın derinliğini artırırken, aynı zamanda eğitim sürecini de hızlandırır. Bu yenilikler sayesinde, ResNet

ağları, 152 katmanlı bir mimariyi eğitmeyi başardı ve görüntü tanımada çığır açtı.

Bu tür derin ağların eğitilmesindeki temel zorluk, daha derin katmanlardaki gradyanların büyüklüğünü artıran veya azaltan birçok işlev nedeniyle katmanlar arasındaki gradyan akışının engellenmesidir (Aşağıda “patlayan gradientler nedir?” bahsedilmiştir). Ancak mevcut araştırmalar, derin ağlardaki temel öğrenme sorunlarının, özellikle de toplu normalleştirme kullanıldığında, bu sorunlardan kaynaklanmadığını göstermektedir. Öğrenme sürecinin doğru bir şekilde yakınsamasını sağlamak, yapay zeka araştırmalarının en önemli zorluklarından biridir. Derin öğrenme ağlarında, karmaşık kayıp yüzeyleri, ağların yakınsamasını engelleyen bir "engel" oluşturabilir. Ayrıca bazı derin ağlar, eğitim ve test hataları arasında büyük bir farka sahip olsa da, birçok derin ağ hem eğitim hem de test verilerinde büyük hatalara sahiptir. Bu, optimizasyonun yeterince sofistike olmadığı anlamına gelir [49].

Hiyerarşik özellik mühendisliği, sinir ağı tabanlı öğrenmenin kutsal kâsesidir ancak katmanlı uygulaması, tüm görüntü kavramlarını aynı soyutlama seviyesini gerektirmeye zorlar. Bazı kavramlar yüzeysel bir ağda öğretilbilirken, diğerleri ince taneli bağlantılar gerektirir. Bir sirk filini kare bir çerçeve içinde düşünün. Fil, kare çerçevenin "gölgesi" gibidir. Filin karmaşık özelliklerini yakalamak için, ağ, gölgeyi oluşturan birçok katmanı anlamalıdır. Kare çerçevenin özelliklerini yakalamak için, ağ yalnızca gölgeyi anlamalıdır. Çok derin bir ağ, sığ bir mimarinin öğrenebileceği birçok kavramı öğrenebilir ve bu ağlar, yakınsama hızında sığ mimarilerden daha iyi olabilir. Yakınsama hızı, ağ derinliği ile her zaman doğru orantılı değildir. Peki neden bir sinir ağının her bir özelliği öğrenmek için kaç katman kullanacağına kendinin karar vermesine izin vermiyorsunuz?

ResNet, katmanlar arasında atlama bağlantıları kullanarak, daha önceki katmanlardan gelen bilgileri kullanarak, daha sonraki katmanların daha iyi bir iş çıkarmasını sağlar. Bu, özellik mühendisliğinin "bir bütün olarak" değil, "parçalar halinde" yapılmasına olanak tanır. Uzun süreli kısa süreli bellek ağları ve yinelemeli geçitli modüller, sıralı verilerde benzer ilkeleri kullanır ve uyarlanabilir geçitler kullanılarak durum bölümlerinin bir katmandan diğerine kopyalanmasına izin verir.



(a) Artık Modüldeki Atlamalı Bağlantılar

(b) ResNet'in Kısmi Mimarisi

Şekil 2.17: Artık Modülü ve ResNet'in İlk Birkaç Katmanı.

ResNet için, var olmayan "portallar" her zaman tamamen açık kabul edilir. ResNet, ileri beslemeli ağları "bir zincir"den "bir ağa" dönüştürür. Bu, ileri beslemeli ağların "bir yönlü" akışını "çift yönlü" bir akışa dönüştürür. Örneğin, bir ResNet, bir resimdeki bir nesnenin şeklini tanımak için, nesnenin kenarlarını ve köşelerini tanımak için daha önceki katmanlardan gelen bilgileri kullanabilir. Bu, daha sonraki katmanların nesnenin genel şeklini daha doğru bir şekilde tanımlamasına olanak tanır. $R = 2$ için atlanmış bir bağlantı örneği olan ResNet çekirdek modülü, Şekil 2.17(a)'da gösterilmiştir. Geriye yayılma algoritması artık atlama bağlantılarını kullanarak gradyanı geriye doğru yaymak için bir hiper yola sahip olduğundan, böyle bir yaklaşım verimli gradyan akışına izin verir. Konvolüsyonel sinir ağları, lego gibi basit parçalardan oluşturulan karmaşık yapılardır. Çoğu katman 1 adımlı hassas yataklı filtreler kullanır, böylece girdilerin uzamsal boyutu ve derinliği bir katmandan diğerine değişmez. Bununla birlikte bazı katmanlar, her bir uzamsal boyutu iki kat azaltmak için kademeli konvolüsyon kullanır. Aynı zamanda, derinliği iki kat

azaltmak için daha fazla filtre kullanılır. Bu durumlarda, bağlantıları atlamak için eşdeğerlik fonksiyonu kullanılamaz. Bu nedenle, Atlama bağlantıları, boyut uyumsuzluğunun üstesinden gelmek için doğrusal bir projeksiyon matrisi kullanabilir. Bu projeksiyon tablosu, uzamsal yayılmayı iki kat azaltmak için adım 2 için bir dizi 1×1 konvolüsyon işlemi tanımlar [50].

ResNet, "dallı" bir yaklaşım kullanır. Bu, yalnızca ardışık katmanlar arasında bağlantılar olduğu anlamına gelir. Örneğin, $r = 2$ kullanırsak, yalnızca ardışık tek katmanlar arasındaki atlama bağlantıları kullanılır. Bu, daha önceki katmanlardan gelen bilgilerin daha sonraki katmanlara iletilmesini zorlaştırabilir. ResNet'te, Şekil 2.17(a)'da gösterilen temel modül tekrarlanır, böylece minimum ileri hesaplama ile girişleri çıkışlara yaymak için atlama bağlantıları tekrar tekrar izlenebilir. Şekil 2.17(b), mimarinin ilk katmanlarından bir örneği göstermektedir. Bu anlık görüntü, 34 katmanlı mimarinin "temelini" göstermektedir. Bu temel, daha sonraki katmanların daha iyi bir şekilde öğrenmesine izin verir. Şekil 2.17(b)'de, atlama bağlantılarının çoğu sürekli çizgiler olarak görünmektedir, bu da değişmez filtre hacmine sahip bir ID fonksiyonunun kullanımına karşılık gelmektedir. Bununla birlikte, bazı katmanlarda uzamsal izi ve derinlik izini değiştiren 2 adım vardır. Bu katmanlar, noktalı atlama bağlantılarıyla gösterilen bir projeksiyon tablosunun kullanılmasını gerektirir. ResNet, "küçükten büyüğe" bir yaklaşım benimsedi. En küçük mimari bile, önceki en iyi yaklaşımı geride bıraktı [50].

Atlama bağlantıları, geriye yayılma algoritmasına "bir otoyol" sağlar. Bu, algoritmanın daha hızlı ve daha verimli öğrenmesine olanak tanır. Örneğin, bir resimdeki bir nesnenin şeklini tanımak için, ResNet, nesnenin kenarlarını ve köşelerini tanımak için daha önceki katmanlardan gelen bilgileri kullanabilir. Bu, daha sonraki katmanların nesnenin genel şeklini daha doğru bir şekilde tanımlamasına olanak tanır. Doğrusal olmayan girdiler, "bir atlama" yapabilir. Daha karmaşık yapılara sahip diğer girdiler, ilgili özellikleri çıkarmak için daha fazla bağlantıdan geçebilir. Dolayısıyla bu yaklaşım, daha uzun yollarda öğrenmenin, daha kısa yollarda, daha basit öğrenmeyi geliştirmek gibi olduğu artık öğrenme olarak da adlandırılır. Başka bir deyişle, bu yaklaşım özellikle görüntünün farklı yönleri farklı karmaşıklık seviyelerine sahip olduğunda uygundur ve [49]'ün çalışması, daha derin katmanların artık tepkilerinin genellikle nispeten küçük olduğu ve sabit bir derinliğin doğru öğrenmeyi engellediği sezgisini doğrulamaktadır. Bu gibi durumlarda, kısa yollar engelsiz gradyan akışı ile önemli miktarda öğrenmeye izin verdiği için yakınsama genellikle bir sorun

değildir. [50]'teki ilginç bir gözlem, ResNet'in sığ bir ağ kümesi gibi davranmasıdır. Bu tür bir mimaride, "daha kısa yollar daha iyidir". Bu, atlama bağlantılarının, daha önceki katmanlardan gelen bilgileri kullanarak, daha sonraki katmanların daha fazla doğrusal olmayan girdileri sınıflandırmasına olanak tanır. Daha kısa yollar, daha az karmaşıktır ve bu nedenle daha az hata yapma olasılığı vardır. Örneğin, bir resimdeki bir nesnenin şeklini tanımak için, ResNet, nesnenin kenarlarını ve köşelerini tanımak için daha önceki katmanlardan gelen bilgileri kullanabilir. Bu, daha sonraki katmanların nesnenin genel şeklini daha doğru bir şekilde tanımlamasına olanak tanır. Derin öğrenme göz ardı edilebilir ve yalnızca kesinlikle gerekliyse bu gerçekleşir. [50]'te ResNet mimarisi, paralel ardışık düzende farklı yolları açıkça göstererek resimsel olarak temsil edilmiştir. Bu dağıtım diyagramı, ResNet'in neden küme merkezli tasarım ilkelerine benzer olduğunun net bir şekilde anlaşılmasını sağlar. Bu bakış açısının bir sonucu, tahmin sırasında eğitilmiş ResNet'in belirli katmanlarının kaldırılmasının VGG gibi diğer ağlar kadar büyük bir doğruluk kaybına yol açmamasıdır. Genişletilmiş artık ağlar üzerine yapılan bir çalışma [51] okunarak daha fazla fikir edinilebilir. Bu çalışma, çok derin yolların çoğu kullanılmadığı için artık ağın derinliğini artırmanın her zaman yararlı olmadığını göstermektedir. Bypass bağlantıları alternatif yollar sağlar ve ağ genişliğini önemli ölçüde artırır.

[51]'un çalışması, toplam katman sayısını belirli bir eşikle sınırlandırarak (örneğin 150 yerine 50) ve her katmanda daha fazla filtre kullanarak daha iyi sonuçlar elde edilebileceğini göstermektedir. 50 değeri, pre-ResNet standartlar için hala oldukça büyüktür ancak son kalıntılar bunun ağ deneylerinde kullanılan derinliğe kıyasla daha düşük olduğunu belirtmektedir. Bu yöntem, işlemleri paralelleştirmek için de kullanışlıdır.

Xception mimarisi

Xception, "katmanlarını bölen" bir CNN mimarisidir. Bu, her katmanın, çıktısında yalnızca belirli bir özellik türüyle ilgilenmesine olanak tanır. Bu, daha az karmaşık ve daha verimli bir mimariye yol açar. Bu mimariye Xception (Extreme Inception) denmesinin nedeni aslında Inception modelinden esinlenilmiş olmasıdır. Xception, "katmanlarını bölerek ve ardından birleştirerek" çalışan bir CNN mimarisidir. Bu, her katmanın, çıktısında yalnızca belirli bir özellik türüyle ilgilenmesine olanak tanır. Bu, daha az karmaşık ve daha verimli bir mimariye yol açar. Ayrıca, artık bağlantılar, katmanların daha önceki katmanlardan gelen bilgilere erişmesine olanak tanıyarak, daha derin mimarilerde bile daha iyi öğrenme sağlar.

Xception'ın ImageNet üzerinde Inception V3'ten biraz daha iyi performans gösterdiği iddia edilmiştir. Tablo 2.1 ve Şekil 2.3, ImageNet üzerinde üç büyük CNN mimarisinin performansını karşılaştırıyor. VGG-16, ResNet-152 ve Xception, sırasıyla %68.3, %76.2 ve %77.3 doğruluk elde etti. Bu sonuçlar, Xception'ın ImageNet üzerinde en iyi performans gösteren CNN mimarisi olduğunu gösteriyor [57].

Veriye dayalı işleme

İnançlarımız, başkalarının bakış açılarını duyma ve anlama yeteneğimizle şekillenir. Başkalarının bize söyledikleri şeyleri, kendi deneyimlerimizle birleştirerek ve yorumlayarak, kendi inançlarımızı oluşturuyoruz. Bu süreç, söylenen şeyin anlamını anlamamıza ve ona göre bir yargıya varmamıza olanak tanıyan söylemin anlama özelliklerine bağlıdır. Nedeni her zaman eksiktir. Mükemmel derecede açık olmaya çalışan metinler ve konuşmalar şaşırtıcı bir şekilde okunamaz, çünkü çok fazla başlık, açıklama, tanım ve yorumla doludurlar.

Yazarlar ve konuşmacılar, mesajlarını daha açık hale getirmek için bunların çoğunu söylemeden bırakmalıdır. Bu gözlem özellikle temel varsayımlar için geçerlidir. Bu tür varsayımlar nadiren açıkça ifade edilir. Şu ifadeye bir bakalım :Ann, Ocak ayında doğdu, Fred ise aynı yılın Aralık ayında doğdu; bu nedenle Ann, Bill'den daha yaşlıdır çünkü doğduğu tarihten itibaren geçen zaman diliminde daha fazla yaşlanmıştır. Ancak, bir hikaye anlatıcısının çok çok uzak bir galakside zamanın doğrusal olduğunu söyleyerek başlaması pek olası değildir. Bu daha derin ilkeler genellikle fark edilmez. Bunun bir nedeni, yazarın veya konuşmacının bu ilkelerin tam olarak farkında olmaması olabilir; diğer bir nedeni ise genel ilkelerin hem gönderici hem de alıcı için ortak kabul edilmesi ve dolayısıyla açıkça ifade edilmesine gerek olmamasıdır [58].

Peki durum böyle ise ve oluşturduğumuz varsayım doğru değilse ne olacak? Eğer bir yazar belirli varsayımlara dayanarak bir metin yazarsa, bu varsayımları paylaşmayan okuyuculara ne olur? Okuyucular farklılıkları fark edip not ediyor mu yoksa metni anlaşılmaz mı buluyor? Tipik bir sonuç yoktur. Bunun yerine, okuyucunun ön bilgilerinden yararlanılır ve tutarlı ve bağlantılı bir yorum yaratılır, ancak bu yazarın aklındaki yorum değildir. Amaçlanan mesaj, okuyucunun önceki algıları ve inançları tarafından özümseenecek ve çarpıtılacaktır [59].

Asimilasyon gerçekleşmekte çünkü söylemi anlama, algılama ve çıkarımda bulunma yorumlama süreçleridir ve yorumlama her zaman bir seçim meselesidir. ercih, alıcının

önceden sahip olduğu bilgilere göre yapılır. Zamanı kısaltmak için, bu tartışma sadece dinlediğini anlama için geliştirilecektir ancak aynı zamanda algılama ve çıkarım için de geçerlidir. Söylemsel anlamada seçim; kelime, cümle ve söylem düzeyinde yapılır. Sözcükler, onları algıladığımızda ve anladığımızda bir anahtar açar. Bu anahtar, sözcüklerin anlam dünyasına açılan kapıdır. Birçok kelime anlamlıdır yani birden fazla anlamı ifade etmek için kullanılırlar. Örneğin, "çizgi" kelimesi bir ödeme hattı, bir olta, bir kalem çizimi vb. gibi birçok farklı şeye atıfta bulunabilir. "Çizgi" kelimesi, zihnimizde bir çağrışım zinciri başlatır [60].

Amaçlanan anlamı bulmak için alıcılar, dil bilgilerinin yanı sıra dünya bilgilerini de kullanırlar. Örneğin, "nehir" gibi kelimeler aynı bağlamda kullanıldığında, "misina ipi" kelimesinin bir yorumunu olta ipi olarak tercih eder ve diğer bir yorum olan "yazar kasa ipi" yorumunu bastırırlar [60].

Öte yandan, "bakkal" ve "mağaza" gibi kelimeler ikincisini desteklerken birincisini bastırır. Bu durumda, kelimelerin anlamı dinleyicinin ön bilgisine bağlıdır, örneğin "gölde olta kullanılır" veya "markette sık sık kuyruklar olur". Ardından analizimizi düşünün. Bir cümledeki kelimeler, birbirleriyle olan ilişkilerinde bir bulmaca gibidir. Bu bulmaca, cümlelerin anlamını ortaya çıkarır. Cümlelerin anlamı, dünyayla olan ilişkimizden doğar. Kelime formları, kelime sırası, edatlar ve noktalama işaretleri kelimelerin nasıl ilişkili olduğu hakkında bilgi verir, ancak aşağıdaki cümle örneğinde görüldüğü gibi bu bilgi yeterli olabilir veya olmayabilir. Casus dürbünün arkasındaki adamı süzdü. Dürbünün sahibi bir casus mu yoksa bir ajan mı? Böyle bir şey söylenebilir. Dünyanın nasıl işlediğine dair biraz bilgiyle, bu soru yapısal olarak benzer bir önermeyle çabucak çözülür. Kuş gözlemcisi dürbünüyle izledi. Başkalarını gözetlemek istemeyen kuş muhtemelen dürbünün sahibi değildir. Yorumlanma şeklini değiştiren dil değil, kuşun bilgisidir.

Konuşmayı anlamak için cümlelerin birbiriyle ilişkili olması gerekir. Örneğin, beş kitap rafın üstünde duruyorsa ve altıncı kitap masanın köşesinde duruyorsa, altıncı kitabın diğer beş kitaba göre konumunu belirlemekte tereddüt etmeyiz. Burada söz konusu olan bilgi çok soyuttur: yukarı ve aşağı gibi uzamsal ilişkilerin geçişliliği. A, B'nin üzerindeyse ve B, C'nin üzerindeyse, A, C'nin üzerindeki her şeyin de üzerindedir. A, C'nin üzerinden baktığında, C'nin altındaki her şeyi görebilir. A, C'nin üzerindeki her şeye de ulaşabilir [61].

Bu tür örtük bağlayıcı çıkarımlar, okuduğumuzda veya dinlediğimizde yüksek düzeyde

gerçekleşir ve bilgiye dayanır. Söylemin anlaşılmasında ön bilginin güçlü ve önemli rolü, söylemin alıcılarının da yazarlar ve konuşmacılarla aynı şekilde söyleme anlam katması gibi özel bir sonuca sahiptir. Alıcı, konuşmacının ya da yazarın aklında ne olduğuna doğrudan erişemediği için, yorumu belirleyen alıcının bilgisidir. Bilgiler, birbirleriyle tutarlıysa, uyumlu bir melodi gibidir. Eğer okuyucular ve dinleyiciler söylemin anlamına büyük ölçüde katkıda bulunurlarsa, söylemden çıkarılan mesajlar kaçınılmaz olarak onların inandıklarıyla örtüşecektir. İnançlar, bir kez oluştuktan sonra, yeni bilgilerle değiştirilmesi zor olabilir.

Dünya'nın yuvarlak olduğunu söyleyen otorite figürü, çocuğun zihninde bir kalıp oluşturur. Bu kalıp, çocuğun Dünya'nın şekli hakkındaki inançlarını şekillendirebilir. Çocuğun, otorite figürünün mesajına inanması daha olasıdır. Büyük olasılıkla çocuk yetişkinin sözüne inanacak ve düz modeli küresel bir modelle değiştirecektir. Joseph Nussbaum, Stella Vosniadou ve diğerleri tarafından yapılan araştırmalar, çocukların dünyayı anlama biçiminin, otorite figürleri tarafından sağlanan bilgilerden daha karmaşık olduğunu gösteriyor. Görünüşte basit olan bu dil, çocukların düz dünya modellerine asimile edilmiştir [62].

Bu durumda, asimilasyon mekanizmasının çalıştığından emin olmak için, çocuğun kullanabileceği seçenekleri göz önünde bulundurun. Yetişkinler, dünyanın bir top gibi olduğunu, yani yüzeyinin her yönden aynı şekilde uzadığını söylüyorlar. "Yuvarlak" terimi muğlaktır ve hem daire (iki boyutlu özellik) hem de küre (üç boyutlu özellik) için kullanılabilir. Çağrılmak ne anlama geliyor? Çocuk dünyanın her yönde sonsuza kadar uzandığına inanmıyorsa, düz bir dünyanın bir yerlerde bir kenarı olmalıdır ve bu kenar bir daire olabilir. Eğer bir yetişkin düz bir dünyanın yuvarlak olduğunu söylerse, çocuklar bunu yargılayacaktır. Jason Lee, Andrew Johnson, Thomas Moher ve ben sık sık çocuklardan dünyayı çizmelerini istediğimizde tereddüt etmeden bir daire çizdiklerini gözlemledik [63]. Dairenin tepesinin hangi yönü gösterdiği sorulduğunda, bazı çocuklar, dünyanın bir krep gibi yukarıdan görüldüğünü varsayarak, kalemlerini çizime dik tutmuş, kalemin ucu tavanı göstermiştir. Başka bir deyişle, eğer dinleyici dünyanın düz olduğuna inanıyorsa, dünyanın yuvarlak olduğuna dair görünüşte çelişkili olan nedenin aksini öğretme gücü yoktur, çünkü dinleyicinin önceki inançları çok fazla yorumlama gücüne sahiptir.

Yetişkinler nihayet çocukları anlayabiliyor mu? Eğer elimizde bir küre tutuyor ve bunun Dünya olduğunu söylüyorsak, Dünya'nın dairesel mi yoksa küresel mi olduğu sorusu çözülmelidir. Bununla birlikte, Vosniadou ve meslektaşları, bir çocuğun gelişiminin farklı

aşamalarında başka özümsemelerin meydana geldiğine dair kanıtlar bulmuştur. Dünyanın küresel olduğunu kabul eden çocuklar, dünyanın düz olduğu fikrini sürdürmek için durumu yanlış anlayabilirler [64]. Dünya hakkındaki kendi gözlemleri ve deneyimlerine dayanarak, bazı çocuklar, dünyanın yuvarlak ama içi boş olduğuna, yarısının toprakla dolu olduğuna ve tepesinde güneşin parladığı bir delik bulunduğuna inanmışlardır. İçi boş bir kürede yaşıyoruz ve kürenin alt yarısını dolduran düz bir toprak yüzey üzerinde yürüyoruz. "Dünyanın bir küre olduğu" iddiası, "neredeyse düz bir yüzeyde yaşadığımız" inancıyla tutarlı olacak şekilde çarpıtılarak, "dünyanın içi boş bir küre olduğu" şeklinde yorumlanmaktadır. Kısacası, yazmak ve konuşmak yeni fikirleri iletmenin kolay yolları gibi görünse de derinlemesine analiz durumun böyle olmadığını göstermektedir. Anlama, kelime, cümle ve söylem düzeyinde çoklu yorumlayıcı seçimleri içerir ve bu seçimler büyük ölçüde alıcının önceki inançlarına bağlıdır, bu nedenle konuşmanın yorumlanması neredeyse kesinlikle bu inançlarla tutarlıdır. Yeni bilgiler kişinin mevcut algı ve inançlarına göre yorumlandığından, bu algı ve inançları değiştirme gücü çok azdır. Dünyamızı anlamlandırmak için kullandığımız ön kavramlar, aynı zamanda, yeni bilgilere karşı önyargılarımızı da oluşturur. Bu paradoks, mantıksal akıl yürütmeyi öğrenmenin önündeki temel bir engeldir.

Atlama mimarilerin versiyonlu karşılaştırılması

ResNet mimarisi önerildiğinden beri, performansı daha da artırmak için çeşitli varyasyonlar önerilmiştir.

Tablo 2.3: Çeşitli ILSVRC Yarışmasının En İyi Performans Gösteren Girişlerindeki Katman Sayısı.

Name	Year	Number of Layers	Top-5 error
---	2012 den önce	≤ 5	$> \%25$
<i>AlexNet</i>	2012	8	$\% 15.4$
ZfNet/Clarifai	2013	8/>8	$\%14.8 / \% 11.1$
VGG	2014	19	$\% 7.3$
GoogLeNet	2014	22	$\% 6.7$
ResNet	2015	152	$\%3.6$

Bunlardan ilki [53]'te önerilmiştir. Başlangıçta önerilen Otoyol Ağı, [52] atlamalı bağlantı kavramını getirmiş ve daha kapsamlı bir mimari olarak değerlendirilebilir. Bir otoyol ağında, kimlik eşlemesi yerine kapılar kullanılır, ancak kapalı bir kapı çok fazla bilginin geçişine

izin vermez. Bu durumlarda, geit ağı artık bir ağı gibi davranmaz. Bununla birlikte, artık ağı, kapıların her zaman tamamen açık olduėu bir ağı geidi ağının özel bir durumu olarak görülebilir. Otoyol ağıları LSTM ve ResNet ile yakından ilişkilidir, ancak ResNet oklu yollarla gradyan akışlarını etkinleştirmeye odaklandıėı için görüntü tanıma görevlerinde hala daha iyi performans gösteriyor gibi görünmektedir. ResNet mimarisi, basit katman blokları kullanarak, atlama bağlantıları arasında bilgi aktarımını sağlar. ResNext mimarisi ise, başlangı blokları kullanarak, atlama bağlantıları arasında daha fazla bilgi aktarımını mümkün kılar [54].

Atlama bağlantıları kullanmak yerine, her bir katman ifti arasında konvolüsyon dönüşümleri kullanılabilir [55]. DenseNet mimarisi, her katmanda önceki tüm katmanlardan gelen bilginin birleştirilmesini sağlar. Bu, ağıdaki bilgiyi daha verimli bir şekilde kullanmaya ve daha iyi performans elde etmeye yardımcı olur. Örneėin, bir DenseNet mimarisi 10 katmandan oluşmuşsa, bu mimaride toplam 100 katman dönüşümü vardır. Bu, geleneksel bir ileri beslemeli ağı mimarisinde olması gerekenden ok daha fazla katman dönüşümü anlamına gelir. Bu, ağıdaki bilgiyi daha verimli bir şekilde kullanmaya ve daha iyi performans elde etmeye yardımcı olur. Böyle bir mimarinin amacının, her katmanın bir atlama bağlantısına benzer şekilde yararlı soyutlamalardan öğrenmesine izin vermek olduğunu unutmayın.

Bu durumda, atlanan bağlantılar arasındaki bazı bloklar eğitim sırasında rastgele atılır, ancak test sırasında ağı tamamı kullanılır. Bu metodun, ağı sıkıştırmak için düğümleri kaldırmak yerine incelten Dropout'a benzediėini aklınızda bulundurun. Bununla birlikte, Dropout, katman başına düğümleri kaldırmaktan biraz farklı bir motivasyona sahiptir, ünkü ikincisi özelliklerin birlikte uyarlanması önlemek yerine gradyan akışını iyileştirmeye odaklanır.

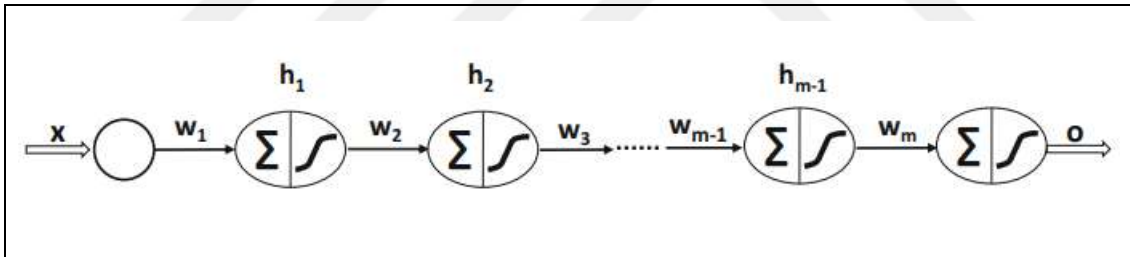
Patlayan gradient

Kaybolan ve Patlayan gradyanlar, derin sinir ağlarının eğitimi sırasında karşılaşılan yaygın sorunlardır. Kaybolan gradyan, geriye yayılma sırasında hesaplanan gradyanların büyüklüėü ok küçük olduėunda ortaya çıkar ve ağı parametrelerini etkili bir şekilde güncellemeyi zorlaştırır. Öte yandan, gradyanların büyüklüėü ok büyüdüėünde patlayan gradyanlar meydana gelir ve ağı kararsız hale gelmesine ve zayıf öğrenmesine neden olur. Her iki sorun da ağırlıkların uygun şekilde başlatılması ve toplu normalleştirme gibi normalleştirme teknikleriyle hafifletilebilir.

Sinir ağı öğrenimi ile ilgili kararlılık sorunları nedeniyle sinir ağlarında başlatma çok önemlidir. Sinir ağları genellikle her katmanın aktivasyonunun zayıfladığı veya güçlendiği kararlılık sorunlarına sahiptir. Bu olgu, ağın derinliği ile üstel olarak ilişkili olduğu için özellikle derin ağlarda şiddetlidir. Bu olguyu bir dereceye kadar düzeltmenin bir yolu iyi bir başlangıç noktası seçmektir, böylece gradyan çeşitli seviyelerde kararlı olur.

Ağırlıkları başlatmak için, ağdaki tüm bağlantıların ağırlıklarını, ortalaması sıfır ve standart sapması 10^{-2} olan küçük bir Gauss dağılımından rastgele değerlerle başlatabiliriz. Bu, ağdaki tüm bağlantıların başlangıçta hem pozitif hem de negatif küçük değerlere sahip olmasına neden olacaktır [56].

Bu başlatma yöntemi, her nöronun girdi sayısından bağımsız olarak aynı ağırlıkları kullanır. Bu, belirli bir nöronun girdi sayısına göre optimize edilmediği anlamına gelir. Örneğin, bir nöronun sadece iki girdisi ve başka bir nöronun 100 girdisi varsa, ilk nöronun çıktısı, daha fazla girdinin (daha büyük bir eğim olarak görünen) ek etkisi nedeniyle ağırlıklı ortalamaya çok daha duyarlı olacaktır. Genel olarak, çıktının varyansının girdi sayısı ile orantılı olduğu ve bu nedenle standart sapmanın girdi sayısının karekökü ile orantılı olduğu gösterilebilir.



Şekil 2.18: Kaybolan ve Patlayan Gradyan Problemi Figürü.

Bu dengeyi sağlamak için, her ağırlık $\sqrt{1/r}$ standart sapmalı bir Gauss dağılımından alınan bir değerle başlatılır; burada r , o nöronun girdi sayısıdır. Bias nöronları, çıktının ortalamasını sıfıra ayarlamak için kullanılır. Bu nedenle, her zaman sıfır ağırlıkla başlatılırlar. Alternatif olarak, ağırlıklar 'de eşit olarak dağıtılmış değerlere başlatılabilir, $[-1/\sqrt{r}, 1/\sqrt{r}]$.

Daha sofistike başlatma kuralları olarak, farklı seviyelerdeki düğümlerin birbirleriyle etkileşime girdiği ve çıktının hassasiyetine katkıda bulunduğu gerçeğini dikkate alır; r_{in} ve r_{out} sırasıyla belirli bir nöronun girişi ve çıkışıdır; Xavier initialization veya Glorot initialization olarak adlandırılan, önerilen bir başlatma kuralı, standart sapması $\sqrt{2/(r_{in} + r_{out})}$ olan bir Gauss dağılımı kullanmaktır. Tüm ağırlıklar aynı değere başlatılırsa, her katmandaki nöronlar, her eğitim döngüsü boyunca aynı miktarda değişecektir. Bu, ağ modelinin yavaş

öğrenmesine neden olabilir. Bu da katmandaki nöronlar tarafından aynı özelliklerin üretilmesiyle sonuçlanacaktır. Nöronlar arasındaki farklılıklar, ağ modelinin daha iyi öğrenmesine yardımcı olur. Bu farklılıklar, başlangıçtan itibaren sağlanmalıdır [56].

2.3 IMAGE PROCESSING

Dijital görüntü işleme, önerilen çözümlerin gerçek dünya verilerinde nasıl performans gösterdiğini görmek için kapsamlı deneysel çalışmalara ihtiyaç duyar. Görüntü işleme sistemleri, kabul edilebilir bir çözüme ulaşmak için genellikle birçok farklı yaklaşımın test edilmesini gerektirir. Bu, yaklaşımları hızlı bir şekilde formüle etme ve prototip haline getirme becerisinin, başarılı bir sistem geliştirmenin anahtarı olduğu anlamına gelir.

Görüntü işleme, bir görüntünün içeriğini ve yapısını çıkarmak veya değiştirmek için dijital bir bilgisayarın kullanıldığı bir alandır. Bu, görüntünün boyutunu değiştirmek, gürültüyü azaltmak, nesneleri tespit etmek veya görüntüyü farklı bir şekilde göstermek için kullanılabilir. Görüntü işleme, birçok farklı uygulamada kullanılır, örneğin:

- a. Fotoğraf düzenleme ve manipülasyon
- b. Yapay zeka ve makine öğrenimi
- c. Tıbbi görüntüleme
- d. Robotik
- e. Uzay araştırmaları

Görüntü işleme, dijital dünyadaki bilgilerin işlenmesi ve anlaşılması için önemli bir araçtır. Dijital bir bilgisayarın, dijital bir görüntüyü işleme işlemine "dijital görüntü işleme" adı verilir [65].

Dijital bir görüntü, insan algısının temelini oluşturan sonlu sayıda pikselden oluşur. Pikseller, görüntünün içeriğini ve yapısını temsil eder. Bu bilgiler, nesneleri tespit etmek, gürültüyü azaltmak veya görüntüyü farklı bir şekilde göstermek için kullanılabilir. Bu nedenle, dijital görüntülerin nasıl işlendiğini ve yorumlandığını anlamak, çevremizdeki dünyayı daha iyi anlamamıza yardımcı olabilir. Görüntüleme cihazları, elektromanyetik spektrumun çok geniş bir yelpazesini kapsar. Bu, bize insan gözüyle göremediğimiz şeyleri görmemizi sağlar. Örneğin, gama kameraları, kanser hücrelerini tespit etmek için kullanılır. Radyo teleskopları, uzak galaksileri gözlemlemek için kullanılır. Bu cihazlar, çevremizdeki dünyayı daha iyi

anlamamıza ve yeni keşifler yapmamıza yardımcı olur.

Normalde insanlar tarafından görüntülenmeyen nesneler tarafından üretilen görüntüleri de işleyebilir. Görüntüleme cihazları, insan gözünün göremediği şeyleri görmemize olanak tanır. Bu, görüntü işleme için geniş bir uygulama yelpazesine yol açar. Ancak, Görüntü işleme, görüntü analizi ve bilgisayarla görme gibi alanlar, bir görüntüden bilgi çıkarmak için birbirleriyle örtüşen ve iç içe geçmiştir. Görüntü işleme bazen girdisi ve çıktısı görüntü olan bir alan olarak tanımlanmasıyla ayırt edilir. Bu durumun kısıtlayıcı ve biraz da yapay bir kısıtlama olduğunu düşünüyoruz. Örneğin, bu tanım altında bir görüntünün ortalama yoğunluğunu hesaplamak gibi önemsiz bir görev bile bir görüntü işleme görevi olarak kabul edilemez. Öte yandan, bilgisayarla görme gibi bazı alanlar, nihai hedefi görsel verilere dayalı öğrenme, muhakeme ve eylemi mümkün kılmak olan insan görüşünü taklit etmek için bilgisayarları kullanmaktadır. Görüntü işleme alanı, bilgisayarların insan gözünün yaptığı gibi dünyayı görmesini sağlayan yapay zekadır. AI, pratik kullanıma girme sürecinde olan ve ilerlemenin beklenenden çok daha yavaş olduğu bir alandır. Görüntü analizi (görüntü anlama olarak da bilinir), görüntü işleme ve bilgisayarla görme arasında kalan bir alandır.

O halde görüntü işlemeyen bilgisayarla görmeye net bir sınır yoktur. Bununla birlikte, bu süreklilik bağlamında, üç tür bilgisayar işlemeyi göz önünde bulundurmak yararlı bir örnektir: düşük seviye, orta seviye ve yüksek seviye. Görüntü ön işleme, gürültüyü azaltmak, kontrastı iyileştirmek ve görüntüyü keskinleştirmek için birincil işlemleri içerir. Örneğin keskinlik işleme, Düşük seviyeli işleme, girdi ve çıktının görüntü olmasıyla karakterize edilir. Orta seviye görüntü işleme, segmentasyon (bir görüntüyü bölgelere veya nesnelere ayırma), nesne tanımlama (bir görüntüyü bilgisayar işlemine uygun bir formata dönüştürme) ve tek tek nesnelerin sınıflandırılmasını (tanıma) içerir. Ara işleme genellikle girdinin bir görüntü olması ve çıktının görüntüden çıkarılan özellikler (örneğin, kenarlar, konturlar, tek tek nesnelerin kimlikleri) olmasıyla karakterize edilir. Görüntünün anlamını çıkarmak için kullanılan üst düzey işleme ise insan görüşünün en karmaşık yönlerini bile taklit edebilir. Genellikle, tipik olarak insan görüşü ile ilişkilendirilen bilişsel işlevlerin yerine getirilmesiyle bilinir [66].

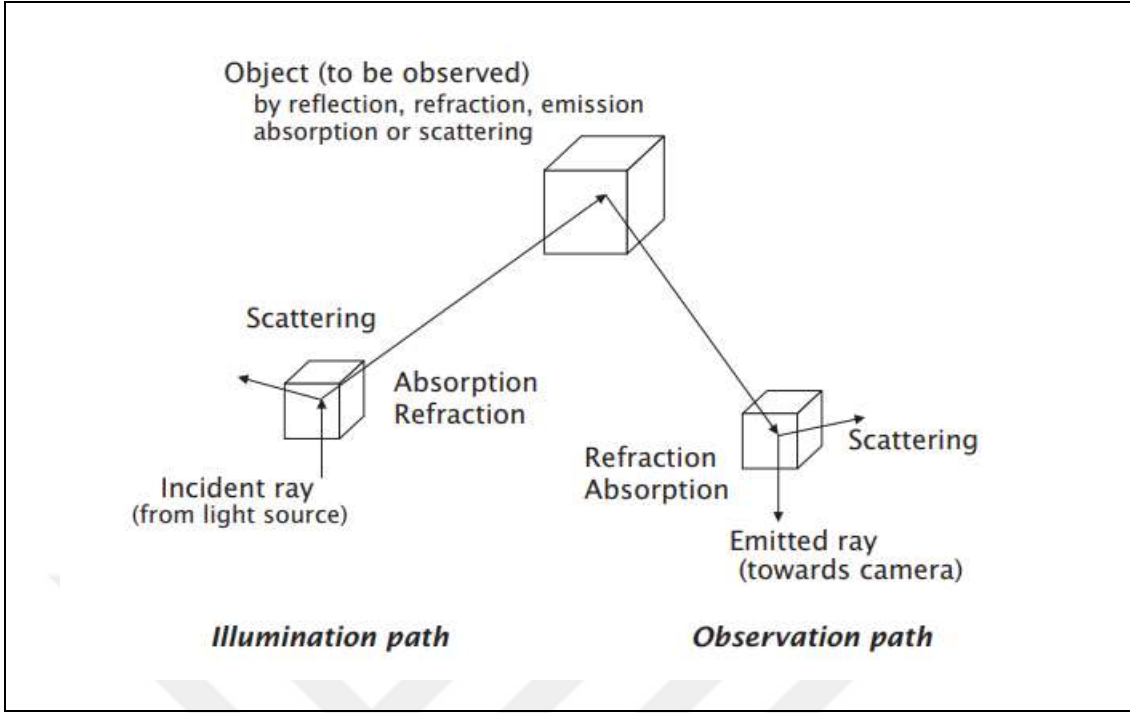
Yukarıdakilerden, görüntü işleme ve görüntü analizinin bir görüntüdeki tek tek alanları veya nesneleri tanımlama açısından mantıksal olarak örtüştüğü görülebilir. Bu nedenle, bu makaledeki dijital görüntü işleme, görüntülerin girdi ve çıktı olarak işlenmesinin yanı sıra,

tek tek nesnelerin tanınmasına kadar görüntülerden özelliklerin çıkarılmasını içerir. Bu kavramları açıklığa kavuşturmak için basit bir örnek olarak otomatik metin analizi alanını ele alalım. Dijital görüntü işleme, metin içeren bir görüntünün anlamını çıkarmak için kullanılan bir dizi tekniktir. Bu teknikler, görüntünün ön işlemesini, tek tek karakterlerin ayrıştırılmasını, karakterlerin tanımlanmasını ve karakterlerin tanınmasını içerir. Metin içeren bir görüntünün anlamını çıkarmak, görüntü analizi ve bilgisayarla görme gibi diğer alanların da kapsamına girer. Dijital görüntü işleme, bu teknikleri kullanarak metin tanıma, görüntü işleme ve hatta gerçek dünyayı anlama gibi çeşitli uygulamalarda kullanılmaktadır.

Görüntüleme sistemleri nesneler tarafından yayılan radyasyonu toplar ve görüntüler. Radyasyon parçacık akımları, elektromanyetik dalgalar ve akustik dalgalardan oluşur. Klasik bilgisayarla görmede, bir sahne veya ışık verildiğinde bu sahne yakalanır ve analiz edilir. Günlük hayatta kullanılan optik sistemler, nesneleri görsel olarak tanımlamak için tasarlanmıştır. Ancak bilimsel ve endüstriyel uygulamalarda kullanılan optik sistemler, nesnelerin özelliklerini ölçmek için tasarlanmıştır. Bu nedenle, bu sistemler, ilgilenilen nesnenin özellikleri ile yayılan ışık arasında niceliksel bir ilişki kurmaya odaklanır. Böyle bir çabanın amacı, diğer parametrelerden toplanan ve yayılan ışığın bozulmasını en aza indirirken ilgilenilen nesnenin özelliklerini haritalamaktır.

Şekil 2.19'de gelen ışının ve nesneden kameraya yayılan ışının ek süreçlerden etkilenebileceğini göstermektedir. Bir nesnenin konumu, yayılan ışınların kırılması nedeniyle kayabilir. Gelen ışınların ve sinkrotron radyasyonunun saçılması ve emilmesi nedeniyle, radyasyon akısının zayıflaması nesnenin kendisinden değil çevreden kaynaklanır ve bu da gözlemleri yanıltabilir.

Uygun bir ortamda, bu ek etkileri en aza indirmek ve alınan radyasyonun ilgilenilen nesnenin özellikleriyle doğrudan ilişkili olmasını sağlamak önemlidir. Aydınlatma veya ayarlardan etkilenmese de en uygun radyasyon türü ve dalga boyu aralığı seçilebilir. Görüntü işleme görevinin karmaşıklığı, nicel görüntüleme sürecinin karmaşıklığını belirler. Tek amaç nesnelerin doğru geometrik ölçümleri ise, aydınlatmayı nesneler eşit şekilde aydınlatılacak ve arka plandan açıkça ayırt edilebilecek şekilde ayarlamak yeterlidir. Bu durumda ilgilenilen nesnenin özellikleri ile kameraya doğru yayılan radyasyon arasında niceliksel bir ilişki kurulmasına gerek yoktur [67].



Şekil 2.19: Nesneleri Görüntülemek Amacıyla Radyasyon ve Madde Arasındaki Etkileşimin Şematik Diyagramı.

Kameraya yayılan radyasyon ile nesnenin özellikleri arasındaki ilişki, gelen ışının saçılması, emilmesi ve kırılmasının yanı sıra senkrotron radyasyonu tarafından bozulabilir. Bununla birlikte, bir nesnenin yoğunluk, sıcaklık, yüzey yönelimi veya kimyasal türlerin konsantrasyonu gibi özellikleri ölçülmek isteniyorsa, seçilen özellik ile yayılan radyasyon arasındaki kesin ilişkinin bilinmesi gerekir. Bir nesnenin rengini belirlemek, nicel görüntüleme sürecinin en basit uygulamalarından biridir.

Nicel görüntüleme, görüntülerin özelliklerinin ölçülmesine odaklanan bir görüntü işleme alt alanıdır. Birçok uygulamada, ilgilenilen parametre ile yayılan radyasyon arasındaki ilişki o kadar net değildir. Bu nedenle, ilgilenilen parametreyi yayılan radyasyondan çıkarmak için daha karmaşık teknikler gerekli olabilir. Örneğin, bir nesnenin bileşimini belirlemek için yapılan bir nicel görüntüleme işlemi, bir nesnenin boyutunu ölçmek için yapılan bir nicel görüntüleme işleminden daha karmaşıktır. Veyahut uydu görüntülerinde kentsel alanlar, ormanlar, nehirler, göller ve tarım arazileri kolayca tanımlanabilir ancak bunları hangi özelliklerine göre tanımlayabiliriz? En önemlisi, görüntüler neden böyle görünüyor?

Benzer şekilde, tıbbi araştırmalardaki görüntülemelerde yaygın sorunlardan biri patolojik anormalliklerin tespit edilmesidir. Patolojik bir anormalliği tanımlayan biyolojik

parametreler ile görüntüdeki görünümü arasındaki ilişkinin iyi anlaşılması, güvenilir kararlar almak için gereklidir [67].



Şekil 2.20: Bazı Yaygın Görüntü İşleme İşlemleri: (a) Orijinal Görüntü; (b) Artırılmış Kontrast; (c) Görüntüdeki Değişim Renk Tonu; (d) "Posterleştirilmiş" (nicelleştirilmiş renkler); (e) Bulanıklaştırılmış; (f) Döndürülmüş İşlem.

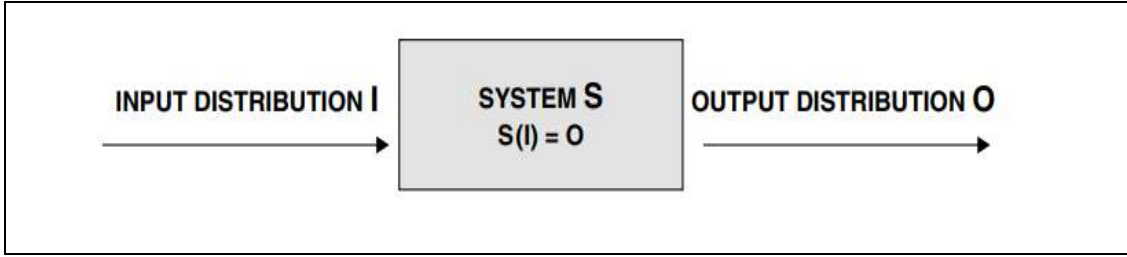
Görüntülerin sahnenin 3D unsurları, aydınlatma, kamera optikleri ve sensörlerin etkileşimiyle nasıl oluştuğunu gördükten sonra, birçok bilgisayarla görme uygulamasının ilk aşamasına bakalım: Görüntüleri ön işleme tabi tutan ve daha fazla analiz için uygun bir formata dönüştüren görüntü işlemedir. Örnekler arasında pozlama telafisi, renk dengesinin ayarlanması, görüntü gürültüsünün azaltılması, keskinliğin iyileştirilmesi ve görüntünün döndürülerek hizalanması yer alır (Şekil 2.20). Görüntü işleme, görüntülerin analiz edilmesi, yorumlanması ve anlaşılması için kullanılan bir dizi tekniktir. Bilgisayarla görme, görüntülerden bilgi çıkarmak için kullanılan bir dizi tekniktir. Görüntü işleme, bilgisayarla görmenin ayrılmaz bir parçasıdır. Bilgisayarla görme uygulamalarının çoğu, görüntü işleme aşamasının iyi bir şekilde tasarlanmasını gerektirir. Bu bölümde, piksel değerlerini bir görüntüden diğerine eşleyen standart görüntü işleme operatörleri incelenmektedir [68].

Görüntü işleme genellikle elektrik mühendisliği bölümlerinde sinyal işlemeye giriş dersinden ileri bir ders olarak öğretilir (Oppenheim ve Schaffer 1996; Oppenheim, Schaffer ve Buck 1999). Bu bölümde, en basit görüntü dönüşümü olan ve her piksel üzerinde komşu piksellerden bağımsız olarak çalışan dönüşümle başlayacağız. Bu tür dönüşümler genellikle nokta operatörleri veya nokta işleme olarak adlandırılır. Komşuluk (bölge tabanlı) operatörler, yeni piksel değerinin, yalnızca birkaç komşu giriş değerine bağlı olduğu operatörlerdir. Bu bölümde, komşuluk operatörlerinin temellerini inceleyeceğiz. Bu tür ortamların çalışmasını analiz etmek (ve bazen hızlandırmak) için yararlı bir araç Fourier dönüşümüdür.

En yakın komşu operatörleri, görüntü piramitleri ve dalgaları oluşturmak için kullanılabilir. Bu, çeşitli çözünürlüklerdeki (ölçeklerdeki) görüntüleri analiz etmek ve belirli işlemleri hızlandırmak için yararlıdır. Global operatörlerin bir diğer önemli sınıfı, geometrik dönüşümler olan rotasyonlar, makaslamalar ve perspektif deformasyonlarıdır. Görüntü iyileştirme, görüntülerin görsel kalitesini ve faydalılığını artırmak için kullanılan bir dizi tekniktir. Bu teknikler, görüntünün gürültüsünü azaltmak, keskinleştirmek, renklerini iyileştirmek ve diğer kusurları gidermek için kullanılabilir. Görüntü iyileştirme, görüntünün özelliklerine ve kullanılacağı uygulamaya göre farklı şekillerde gerçekleştirilebilir. Örneğin, bir fotoğrafın görsel kalitesini artırmak için kullanılan teknikler, bir tıbbi görüntünün özelliklerini çıkarmak için kullanılan tekniklerden farklı olabilir. İnsan ve makine görüşüne

uyması için çeşitli özellikler geliştirilmiştir. Bir başka örnek olarak, medyan filtreleme orijinal görüntüyü daha görünür hale getirmek için nokta gürültüsünü azaltır [69].

Genel matematiksel anlamda görüntü oluşumu, bir girdi dağılımını bir çıktı dağılımına dönüştürme süreci olarak görülebilir. Basit bir mercek, ışığın bir bölgedeki dağılımını başka bir bölgedeki dağılmaya dönüştüren bir "transformatör" olarak görülebilir.



Şekil 2.21: Görüntüleme için Bir Sistem Yaklaşımı.

Görüntüleme, bir görüntünün nasıl üretildiğini anlamanın bir yoludur. Bu yaklaşımda, görüntüleme süreci, girdi dağılımı üzerinde hareket eden bir operatör olarak görülür. Bu operatör, görüntünün çıktısını üretmek için girdi dağılımındaki bilgiyi kullanır. Tıbbi ultrason görüntüleme, bir ultrason probunun, insan vücudunun iç organlarından gelen akustik yansımaları ölçerek çalışır. Bu yansımalar, bir bilgisayar tarafından işlenir ve gri tonlamalı yoğunluk görüntüleri olarak görselleştirilir. Bu dönüşüm, girdi dağılımını (ultrasonun ölçtüğü akustik yansıma verileri) çıktı dağılımına (görüntülenen yoğunluk görüntüleri) eşler.

Görüntüleme, bir nesnenin veya sürecin özelliklerini bir görüntüye dönüştürme sürecidir. Sistem teorisi yaklaşımı, görüntüleme sürecini, girdi dağılımından çıktı dağılımına bir dönüşüm gerçekleştiren bir sistem olarak görür. Bu yaklaşım, görüntüleme sürecinin nasıl çalıştığını anlamamıza ve görüntü kalitesini iyileştirmek için nasıl optimize edilebileceğini anlamamıza yardımcı olur [70]. Yukarıdaki şekil bu olayı göstermektedir. Bir görüntü işleme sisteminin girdiyi çıktıya dönüştürme süreci başka bir perspektiften de görülebilir: Fourier alanı veya frekans alanı. Fourier alanında görüntü, farklı frekanslardaki harmonik fonksiyonların üst üste binmesinden oluşur. Görüntü işleme sistemi, görüntünün uzamsal frekans içeriğini dönüştürerek çalışır. Bu, görüntünün görünümünü ve özelliklerini değiştirebilir.

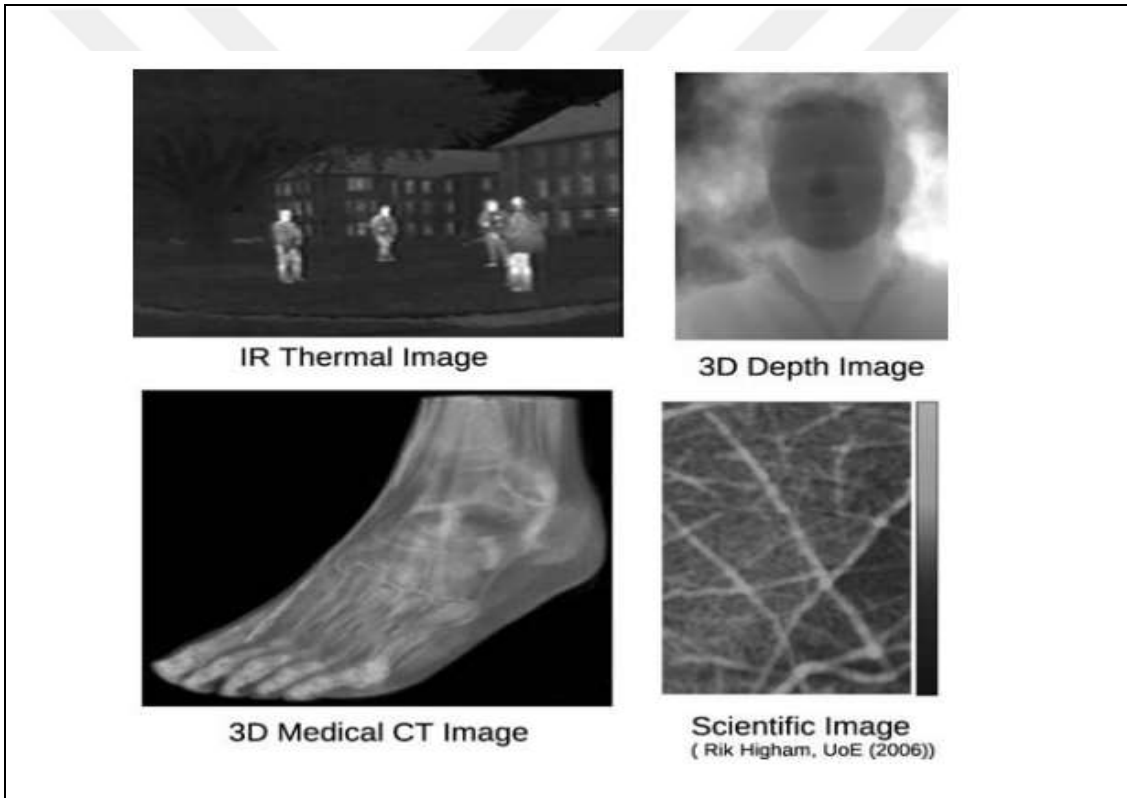
Görüntü işlemedeki en temel işlem, giriş görüntüsündeki tek bir noktanın (yani pikselin) değerini çıkış görüntüsündeki karşılık gelen noktaya (piksele) eşleyen nokta dönüşümüdür.

Matematiksel anlamda, görüntü işleme sistemi, girdiyi çıktıya bire bir eşleyen bir "anahtar" olarak görülebilir.

Görüntü işlemede kullanılan en basit dönüşümler, görüntüler üzerinde temel matematiksel veya mantıksal işlemlerdir. Her biri iki görüntü I_A ve I_B arasında veya bir görüntü ile sabit bir değer C arasında bir işlem olarak gerçekleştirilir:

$$I_{OUTPUT} = I_A + I_B \quad (2.1)$$

$$I_{OUTPUT} = I_A + C \quad (2.2)$$



Şekil 2.22: Farklı Piksel Bilgi Türleri.

Her iki durumda da, çıktı görüntüsündeki tek tek piksellerin (i,j) konum değerleri aşağıdaki gibi eşlenir:

$$I_{OUTPUT}(i,j) = I_A(i,j) + I_B(i,j) \quad (2.3)$$

$$I_{OUTPUT} = I_A(i,j) + C \quad (2.4)$$

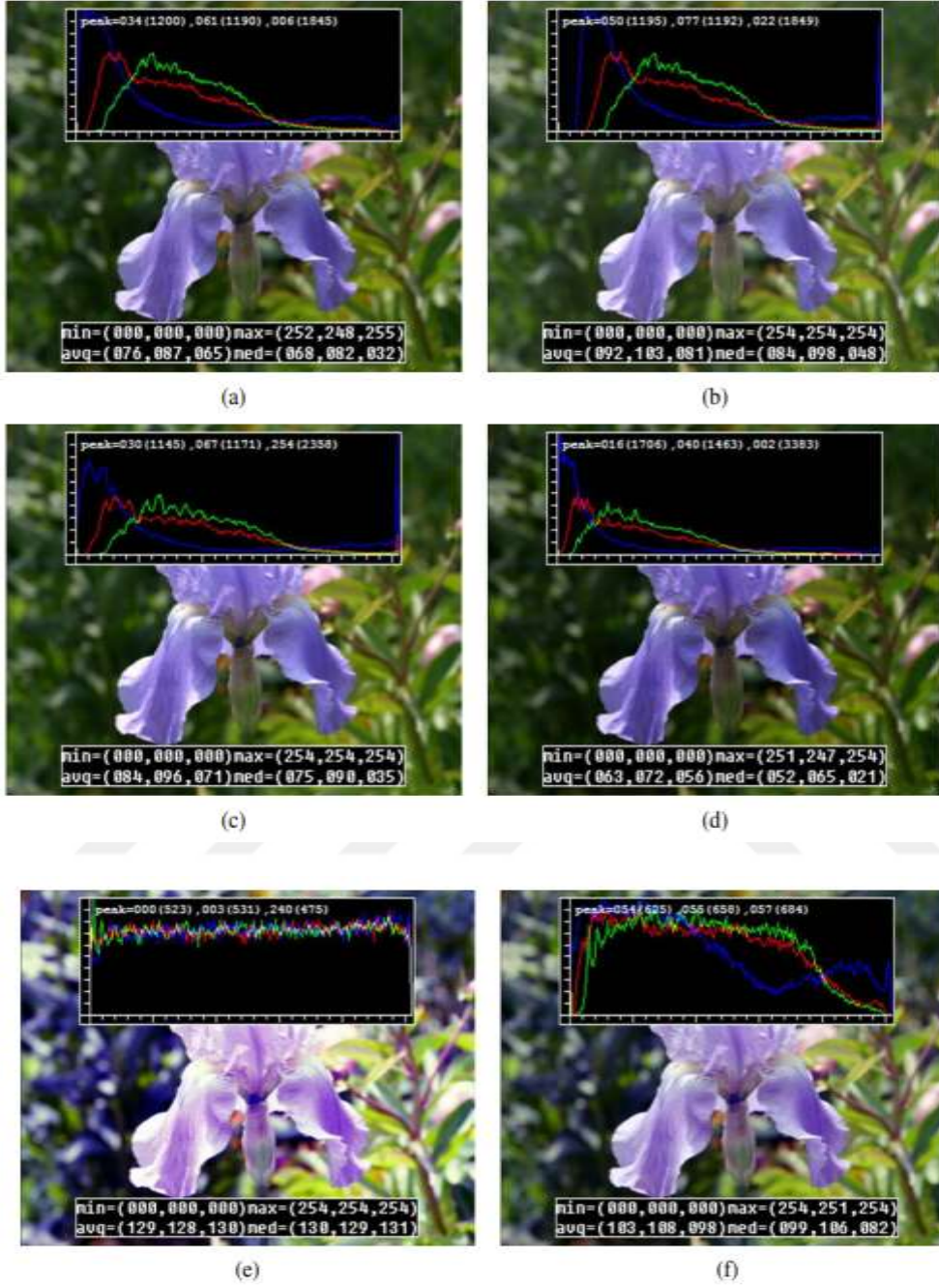
İşlemi $C \times R$ boyutundaki bir görüntünün tamamı üzerine gerçekleştirmek için tüm görüntü indisleri üzerinde yinelememiz yeterli olacaktır. $(i,j) = \{0 \dots C-1, 0 \dots R-1\}$ (N.B in MATLAB piksel indisleri $\{1 \dots C, 1 \dots R\}$)

En basit ancak çok kullanışlı görüntü düzeltme işlemi nokta aritmetiğidir. Çıkış pikseli, bir veya daha fazla giriş pikseli tarafından belirlenir. Eşikleme, görüntü işlemede yaygın olarak kullanılan bir işlemdir. Histogram işleme olarak da adlandırılan kontrast geliştirme için yoğunluk dönüşümü başka bir işleme türüdür. Doğrusal ve doğrusal olmayan filtreleme, çıkış pikselinin giriş pikselinin çok yakınındaki piksellerin bir fonksiyonu olduğu ana işleme türleridir. Doğrusal bir sistem, giriş sinyallerini birbirleriyle etkileşime sokmadan işleme koyar. Bu, giriş sinyallerinin toplamının veya çarpımının çıktıya karşılık gelmesi anlamına gelir.

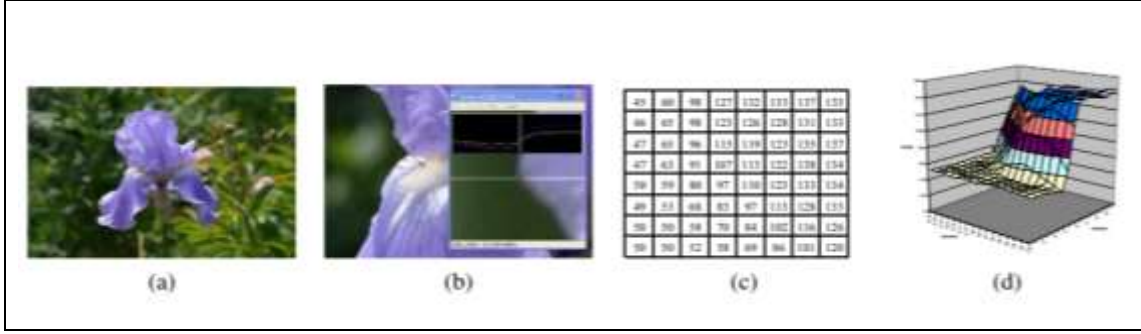
2.3.1 Görüntü İşlemede Nokta Operatörleri

Görüntü işlemede, bir pikseldeki değerin, yalnızca komşu piksellerin değerlerine bağlı olduğu dönüşümler, nokta operatörleri olarak adlandırılır (ek olarak, küresel olarak toplanmış bilgi ve parametreler olabilir). Görüntü işlemede, noktasal işlemler, görüntünün her bir pikseli üzerinde bir veya daha fazla matematiksel işlem gerçekleştirir. Bu işlemler, görüntünün parlaklığını, kontrastını (Şekil 2.22), renklerini veya diğer özelliklerini değiştirebilir (Crane 1997).

Bu bölümde, görüntü işlemede kullanılan en temel noktasal işlemlerden bazılarını kısa bir bakış sunuluyor. Daha sonra görüntülerdeki renklerin nasıl yönlendirilebileceği açıklanmaktadır. Hesaplama fotoğrafçılık ve bilgisayar grafiklerinde, görüntünün farklı parçalarını birleştirmek ve eşleştirmek için kullanılan teknikler tanıtılmaktadır. Bu teknikler, panoramik görüntü oluşturma, 3D modelleme ve video montaj gibi uygulamalarda kullanılır. Son olarak da görüntünün görünümünü iyileştirmek için ton değerlerini (parlaklık ve kontrast) yönlendiren uygulama örnekleri sunulmuştur.



Şekil 2.23: Yukarıda, (a) Orijinal Görüntünün Yanı Sıra, (b) Artırılmış Parlaklığı, (c) Artırılmış Kontrastı, (d) Gamasını ve (e) Histogramını, (f) Kısmi Histogramını Değiştiren Bazı Yerel Görüntü İşleme İşlemleri Gösterilmektedir.



Şekil 2.24: Görüntü Verilerinin Görselleştirilmesi: (a) Orijinal Görüntü; (b) Bir Görüntü İnceleme Aracı Kullanılarak Kırpılmış Kısım ve Tarama Çizgisi Çizimi; (c) Sayılar Izgarası; (d) Yüzey Çizimi. Şekil (c)-(d) için Görüntü Önce Gri Tonlamaya Dönüştürülmüştür.

2.3.2 Piksel İşleme

Dijital bir görüntü içe aktarıldıktan sonra, ilk ön işleme adımı olarak iki tür işlem gerçekleştirilir:

- Nokta İşlemleri
- Geometrik İşlemler

Bu iki işlem temel olarak piksellerin "ne" ve "nerede" olduğunu değiştirir.

Nokta işlemi, tek bir pikselin gri ölçek değerini yalnızca gri ölçek değerine göre veya bazı durumlarda pikselin konumuna göre değiştirir. Genel olarak, bu tür işlemler aşağıdaki gibi ifade edilir.

$$'_{MN} = P_{MN}(G_{MN}) \quad (2.5)$$

P indeksi, nokta işleminin piksel konumuna bağımlılığını gösterir. Geometrik işlem ise yalnızca pikselin konumunu değiştirir; X konumundaki piksel, yeni bir x' konumuna taşınır. İki koordinat arasındaki ilişki geometrik eşleme fonksiyonu tarafından verilir.

$$X' = M(x) \quad (2.6)$$

Noktasal ve geometrik işlemler birbirini tamamlayan işlemlerdir. Görüntüleme sensörünün doğrusal olmayan ve tekdüze olmayan radyosensitivitesi ve görüntüleme sisteminin geometrik bozulmaları gibi görüntü oluşum sürecine zemin hazırlayan bozulmaların düzeltilmesinde etkilidirler. Görüntü aydınlatmasının düzeltilmesi veya optimizasyonu, düşük akış veya taşma tespiti, kontrast geliştirme veya germe, görüntü ortalaması alma, aydınlatma homojensizliğinin düzeltilmesi veya radyometrik kalibrasyon nokta işlemleri ile

gerçekleştirilir. Geometrik işlemler, görüntüyü dönüştürmek için iki ana adım kullanır. İlk adım, görüntünün eşleme fonksiyonunu bulmaktır (2.3). İkinci adım, eşleme fonksiyonunu kullanarak pikselleri yeni konumlarına taşımaktır [71]. Bir görüntü geometrik bir dönüşümle bozulduğunda, orijinal görüntünün pikselleri ile bozulmuş görüntünün pikselleri nadiren örtüşür. Görüntüdeki bazı pikseller, geometrik işlemler nedeniyle orijinal konumlarında olmayabilir. Bu nedenle, bu piksellerin değerlerini komşu piksellerden hesaplamak gerekir.



3. BULGULAR

3.1 SİSTEMDE KULLANDIĞIMIZ RESNET 50 VERİ MODELİ YAZILIM MİMARİSİ

3.1.1 Sistem Alt ve Üst Kütüphaneleri Tanımlamaları

Sistemimizde kullandığımız kütüphaneler aşağıdaki gibidir.

Tablo 3.1: Kütüphane Parametreleri.

Element	Açıklama
matplotlib.pyplot	Veri Kütüphanesi
numpy as np	Çok boyutlu dizileri ve matrisleri destekleyen kütüphane
PIL	Görüntü işleme için oluşturulan Kütüphane
tensorflow as tf	Google'ın makine öğrenmesi için kullandığı açık kaynak kodlu kütüphanesi
Os	Diğer sistemler ile iletişim kurmayı sağlayan kütüphane
tensorflow import keras	Keras kütüphanesi tensorflow üzerinde çalışan deep learning kütüphanesi
keras import layers	Fonksiyonel model oluşturmak için kullanılan kütüphane
keras.layers import Dense, Flatten	Dense yapısı görüntü işlemede katmanlar arası nöron ya da düğüm geçişi sağlar.
keras.models import Sequential	Dense ve layer optimizasyonu için kullanılır.
google.colab import drive	Colab bölümüne eklenen drive kütüphanemiz.

3.1.2 DataSet Tanıtma

Veritabanı olarak oluşturduğumuz nesneleri çekmek için sistemde kullandığımız yordamlar.

Tablo 3.2: Veritabanı Yordamları.

Veri Tabanı Yordamı	Açıklama	Aldığı Değer
!unzip -q	Data set ziplenlen verileri açmak için kullanılan yordam	!unzip -q '/content/gdrive/MyDrive/dataset/bakteri_fotolar.zip'
Pathlib	Veritabanı yolu	import pathlib
dataset_url	Veritabanı çekilen yordam	dataset_url = "/content/bakteri_fotolar"
data_dir	Gözlem yordamı	data_dir = pathlib.Path('/content/bakteri_fotolar')
print(data_dir)	Gözlem yordamı yazdırmak için	content/bakteri/fotolar

3.1.3 View Modelimiz

Sistemde kullandığımız viewler ve yordamlar.

Tablo 3.3: Görüntü Parametreleri.

Parametre	Açıklama
img_height	Görüntü girdi için oluşturulan yükseklik yordamı
img_width	Görüntü girdi için oluşturulan genişlik yordamı
batch_size	Dinamik parti boyutları için kullanılan, parametre güncellemesinin sağlanması için gerçekleştiği ağı verilen alt örneklerin sayısıdır. Yığınınızın büyüklüğü.
train_ds	Test verisi almak için kullanılır.
validation_split	Çoklu sınıflandırma yapmak için kullanılan yapı.
Subset	Tensorflowa veri yüklemek için kullanılıyor.
Seed	Rastgele sayı üretmek için kullandığımız yordam
label_mode	İnt veya float tipli argümanlar için kullanılan yordam.

3.1.4 Resnet Yordamları

Resnet olarak modellemek için kullandığımız yordamlarımız.

Tablo 3.4: ResNet Element Yordamları.

Element	Açıklama
resnet_model	Resnet Modelimizi belirlemek için kullandığımız yapı
pretrained_model	Önceden eğitilmiş modeller kullanmanın amacı, daha yüksek doğruluk değerlerine sahip modeller eğitmektir.
include_top	Ağın tepesinde tam bağlantılı bir katman bulunup bulunmayacağı yordamı.
input_shape	Tam olarak 3 giriş kanalına ve genişlik ve yüksekliğe sahip olmalı.
pooling='avg'	include_top False olduğunda özellik çıkarma için isteğe bağlı havuzlama modu. avg, küresel ortalama konsantrasyonun son konvolüsyon bloğunun çıktısına uygulandığı ve modelin çıktısının bir 2D tensör olduğu anlamına gelir.
weights='imagenet'	Resnet50 Fonksiyon yordamı
pretrained_model.layers	pretrained_model katman yordamı.

3.1.5 Görüntü İşleme İçin Kullanılan Parametreler

Resnet Yapısını kullandığımız derin öğrenme için kullanılan parametreler:

```
from google.colab import drive
drive.mount('/content/gdrive')

Drive already mounted at /content/gdrive; to attempt to forcibly remount, call drive.mount("/content/gdrive", force_remount=True).
```

Şekil 3.1: Dataset İport Ettiğimiz Verisetimiz.

3.1.6 Listdizi Class Yapısı

```
[ ] !unzip -q '/content/gdrive/MyDrive/dataset/bakteri_fotolar.zip'

[ ] import pathlib

dataset_url = "/content/bakteri_fotolar"
data_dir = pathlib.Path('/content/bakteri_fotolar')

[ ] print(data_dir)

/content/bakteri_fotolar
```

Şekil 3.2: Data Veri Setimizden Çekilen Görüntüleri ve Oluşturduğumuz Sınıfları Çekmek için Kullandığımız Yordamlarımız.

```
[ ] print(data_dir)

/content/bakteri_fotolar

[ ] staphylococcus = list(data_dir.glob('staphylococcus/*'))
print(staphylococcus[0])
PIL.Image.open(str(staphylococcus[0]))
```

Şekil 3.3: Bakteri Türünü Çekmek İçin Kullandığımız Metod.

3.1.7 Keras Modeli Optimizasyonu

İmage Dataset çekmek için kullanılan için Keras Modeli:

```
[ ] val_ds=tf.keras.preprocessing.image_dataset_from_directory(
    data_dir,
    validation_split=0.2,
    subset="validation",
    seed=123,
    label_mode="categorical",
    image_size=(img_height,img_width),
    batch_size=batch_size
)
```

Şekil 3.4: Oluşturduğumuz Keras Modeli Metodu.

3.1.8 Classname

Kullandığımız sınıfın görüntü işleme için kullanılan kod parçası bakteri isimlerini bu metodla çekeriz.

```
[ ] class_names=train_ds.class_names
    print(class_names)

['Acetivobacteria', 'Corynebacterium', 'Klebsiella oxytoca', 'Proteobacteria', 'Pseudomonas', 'staphylococcus']
```

Şekil 3.5: Bakteri Türlerini Çekmek için Kullandığımız Metod.

3.1.9 Resnet Model Oluşturma

Derin öğrenme için kullandığımız görüntü işlemede Resnet 50 modeli:

```
[ ] resnet_model=Sequential()

    pretrained_model= tf.keras.applications.ResNet50(include_top=False,
        input_shape=(180,180,3),
        pooling='avg',classes=6,
        weights='imagenet')
    for layer in pretrained_model.layers:
        layer.trainable=False

    resnet_model.add(pretrained_model)
    resnet_model.add(Flatten())
    resnet_model.add(Dense(512, activation='relu'))
    resnet_model.add(Dense(6, activation='softmax'))

[ ] resnet_model.summary()

Model: "sequential"
```

Şekil 3.6: Resnet-50 Metodu.

3.1.10 Resnet Modelleme

Resnet model için kullandığımız çizim kod blokları:

```
[ ] epochs=10
    history = resnet_model.fit(
        train_ds,
        validation_data=val_ds,
        epochs=epochs
    )

[ ] fig1= plt.gcf()
    plt.plot(history.history['accuracy'])
    plt.plot(history.history['val_accuracy'])
    plt.axis(ymin=0.4, ymax=1)
    plt.grid()
    plt.title('Model Accuracy')
    plt.ylabel('Accuracy')
    plt.xlabel('Epochs')
    plt.legend(['train', 'validation'])
    plt.show()
```

Şekil 3.7: Resnet Kod Blokları.

3.1.11 Görüntü Yakalama

Görüntülerimizi yakalamak için kullandığımız kütüphane parça kodları:

```
[ ] import cv2
    image=cv2.imread(str(staphylococcus[0]))
    image_resized=cv2.resize(image, (img_height, img_width))
    image=np.expand_dims(image_resized, axis=0)
    print(image.shape)
```

Şekil 3.8: Görüntü Yakalama Kodları.

4. GEREKSİNİMLER

- a. Donanım: Hesaplama açısından yoğun oldukları için ResNet modellerini eğitmek için iyi bir GPU'ya sahip bir bilgisayara ihtiyacınız olacaktır.
- b. Yazılım: ResNet'i uygulamak için TensorFlow, PyTorch veya Caffe gibi bir derin öğrenme çerçevesine ihtiyacınız olacaktır.
- c. Veri: ResNet eğitme büyük miktarda veri gerektirir, bu nedenle çözmeye çalıştığınız görev için büyük bir açıklamalı veri kümesine erişmeniz gerekir (örneğin, görüntü sınıflandırması için ImageNet).
- d. Teknik Beceriler: ResNet'i uygulamak için derin öğrenme kavramlarını iyi anlamak ve Python'da programlama deneyimi gereklidir.
- e. Zaman: Derin sinir ağlarını eğitmek uzun zaman alabilir, bu nedenle gerekli zamanı ve hesaplama kaynaklarını ayırmaya hazır olmalısınız.
- f. Ağ Mimarisi: ResNet, artık bağlantılara sahip belirli bir sinir ağı mimarisi türüdür, bu nedenle ağ mimarisini görevinizin gereksinimlerine göre tasarlamanız ve uygulamanız gerekir.
- g. Hiperparametre Ayarlama: ResNet'ten en iyi sonuçları almak için öğrenme oranı, yığın boyutu ve ağırlık başlatma yöntemi gibi çeşitli hiperparametreleri ayarlamanız gerekir.
- h. Düzenli hale getirme: Bırakma veya ağırlık azaltma gibi düzenli hale getirme teknikleri, ResNet'te aşırı uyumu önlemeye ve performansı artırmaya yardımcı olabilir.
- i. Veri Ön İşleme: ResNet modelleri, en iyi performansı sağlamak için veri normalleştirme ve veri artırma gibi veri ön işleme gerektirir.
- j. Model Seçimi: Her biri farklı sayıda katmana sahip ResNet-50, ResNet-101 ve ResNet-152 dahil olmak üzere çeşitli ResNet modelleri mevcuttur. Veri kümenizin boyutuna ve karmaşıklığına bağlı olarak göreviniz için doğru modeli seçmeniz gerekecektir.

5. ARAŞTIRMAMIZDA KULLANDIĞIMIZ YÖNTEMLER

Bakteriler, toprakta, denizde, ellerinizde ve bağırsaklarınızda olmak üzere hemen hemen her yerde bulunabilen tek hücreli organizmalardır. İnsan vücudunda birçok bakteri türü bulunmaktadır [72]. Bu makalede, insan gözünde bulunan çeşitli bakteri türlerini inceledik. Araştırmamız, insan gözünde bulunan farklı bakteri türlerini tanımladı ve bu makalede hangi kategoriye ait olduklarını araştırdı.

Gözlerimizde yaşayan tüm mikroorganizmaları içeren bakteriler dört cinse ayrılır. Bunlar stafilocoklar, difteri, propionibakteriler ve streptokoklardır. Sağlıklı insanların gözlerinde, oküler yüzeyde de bulunan TT virüsünün oranı %65'tir. Gözdeki bu mikroorganizmalar yaşa, ırka, sosyal çevreye, yaşanan bölgeye, kullanılan kontakt lenslere ve hastalık durumuna göre değişiklik göstermektedir.[73]

Göz ile yapılmış çalışmalarda elde edilen bazı bakterilerin isimleri şunlardır.

- a. Staphylococcus
- b. Streptococcus
- c. Corynebacterium
- d. Propionibacterium
- e. Proteobacteria
- f. Acetobacteria
- g. Firmicatus
- h. Cyanobacteria
- i. Bacteroidetes
- j. Pseudomonas
- k. Klebsiella oxytoca

Kullandığımız veri seti Image-net, Resnet-50 yapısını kullanarak bulduğumuz sonuçları aktarmaya çalıştık. Sistemimizde birbirinden farklı 6 adet bakteri türü kullandık. Bunların bazıları gözlerimizin içinde yaşamaktadır ve genellikle hastalık yapan bakterilerdir. Elde ettiğimiz bakteriler isimleri ve türleri internet üzerinden elde edilmiştir ve Google'ın kullandığı görüntü kaydetme alt yapı tabanıyla data set oluşturup içine aktarılmıştır.

6. TARTIŞMA VE SONUÇLAR

GoogLeNet'te, katmanın kendisinin birden fazla dala ayrıldığı ve toplanan veya birleştirilen çıktı ürettiği bir katman mimarisi buluyoruz. Toplama veya birleştirme ile biten katmanlar gradyanı eşit parçalara böler ve geriye yayılma sırasında gradyandan geçer. He ve arkadaşları tarafından önerilen artık ağlar veya ResNets, bir katmanı iki dala bölme fikrini kullanan yeni bir mimari olmakla biliniyor. Bir dal sinyale hiçbir şey yapmazken diğer dal sinyali normal bir katman gibi işlemektedir (He et al., 2015). İşlenmemiş veri veya artık, ağdan geçen orijinal sinyale "gürültü azaltıcı bir filtre olarak" eklenir. Bu, ağdaki gürültüyü azaltmaya yardımcı olur. Bu, ağda bir dalın gradyanı değiştirmeden basitçe yaydığı bir bölünme yaratır. Bu da daha derin bir ağ içinden geçen güçlü eğimlerle öğrenir.

Yukarıda gözde bulunan faydalı ve patojen bakterilerden bazıları listelenmiştir. Bu bakteri türlerini kullanarak doku türü yapısına dayalı bir sınıflandırma yaptık. Yaptığımız sınıflandırma, gözdeki bakterilerin hangi türe ait olduklarını bulmaktır. Bu çalışmada türlerini ve isimlerini belirlemek için sistemimizde yaklaşık 600 bakteri resmi kullanılmıştır. Bu şekilde görsellerin boyutlarını eşleştirmeye çalıştık. Çalışmamızda 200 x 200'lük bir görüntü ölçeği kullanılmıştır.

Doğrulama bölünmesini 0.2 olarak ayarladık çünkü bu sayede %80'ini eğitim için ayırmış olduk, %20'sini ise doğrulamayı kullanmak için ayırmış bulunuyoruz. Batch size 132 olarak belirledik. İlk başta eğitim alt kümesini oluşturmamız gerekiyordu. Alt kümemizi oluşturduktan sonra aldığımız çıktı aşağıdaki gibidir:

```
▶ staphylococcus = list(data_dir.glob('staphylococcus/*'))  
  print(staphylococcus[0])  
  PIL.Image.open(str(staphylococcus[0]))
```

📁 /content/bakteri_fotolar/staphylococcus/staphaureus.jpg



Şekil 6.1: Bakteri Sınıflandırmaımızdaki Veri Setinden Elde Ettiğimiz Görüntümüz.


```
[ ] resnet_model.summary()
```

Model: "sequential"

Layer (type)	Output Shape	Param #
resnet50 (Functional)	(None, 2048)	23587712
flatten (Flatten)	(None, 2048)	0
dense (Dense)	(None, 512)	1049088
dense_1 (Dense)	(None, 6)	3078

Total params: 24,639,878
Trainable params: 1,052,166
Non-trainable params: 23,587,712

Şekil 6.2: Resnet-50 için Sistemden Alınan Veri Değerleri.

Kullanılan artık ağ modeline göre (Resnet 50) yaklaşık 20-24 milyon aralığında değer elde edilmektedir. Bu çalışmanın amacı, Resnet 50'yi uygulayarak döngüsel hız öğrenme, veri büyütme ve transfer öğrenme kullanarak Imagenet veri kümesindeki bakteriyel görüntülerden gözdeki bakterileri sınıflandırmadaki performansını değerlendirmektir.

Bu çalışmada yaklaşık %83,33'lük bir doğruluk değeri elde edilmiştir. Burada derin öğrenme model yapısının sınıflandırma gücünü doğrulayabilir ve kendi eğitimimizi eğitebileceğimizi gösterebiliriz. ResNet Model sınıflandırmamızla daha doğru ve tutarlı olan resimler elde etmeye çalışılmıştır.

REFERANSLAR

- [1] Aarup, M., Arentoft, M. M., Parrod, Y., Stader, J., and Stokes, I. (1994). OPTIMUM-AIV: A knowledge-based planning and scheduling system for spacecraft AIV. In Fox, M. and Zweben, M. (Eds.), *Knowledge Based Scheduling*. Morgan Kaufmann.
- [2] Han, X. and Boyden, E. (2007). Multiple-color optical activation, silencing, and desynchronization of neural activity, with single-spike temporal resolution. *PLoS One*, e299.
- [3] Ogawa, S., Lee, T.-M., Kay, A. R., and Tank, D. W. (1990). Brain magnetic resonance imaging with contrast dependent on blood oxygenation. *PNAS*, 87, 9868–9872
- [4] Descartes, R. (1637). *Discourse on method*. In Cottingham, J., Stoothoff, R., and Murdoch, D. (Eds.), *The Philosophical Writings of Descartes*, Vol. I. Cambridge University Press, Cambridge, UK
- [5] Boole, G. (1847). *The Mathematical Analysis of Logic: Being an Essay towards a Calculus of Deductive Reasoning*. Macmillan, Barclay, and Macmillan, Cambridge.
- [6] Gottlob, G., Leone, N., and Scarcello, F. (1999a). A comparison of structural CSP decomposition methods. In *IJCAI-99*, pp. 394–399. Gottlob, G., Leone, N., and Scarcello, F. (1999b). Hypertree decompositions and tractable queries. In *PODS-99*, pp. 21–32.
- [7] Tarski, A. (1941). *Introduction to Logic and to the Methodology of Deductive Sciences*. Dover
- [8] Lucas, J. R. (1961). Minds, machines, and Gödel. *Philosophy*, 36.
- [9] Turing, A. (1936). On computable numbers, with an application to the Entscheidungsproblem. *Proc. London Mathematical Society*, 2nd series, 42, 230–265. Turing, A. (1950). Computing machinery and intelligence. *Mind*, 59, 433–460.
- [10] Berry, D. A. and Fristedt, B. (1985). *Bandit Problems: Sequential Allocation of Experiments*. Chapman and Hall.
- [11] Morrison, P. and Morrison, E. (Eds.). (1961). *Charles Babbage and His Calculating Engines: Selected Writings by Charles Babbage and Others*. Dover.
- [12] Ford, K. M. and Hayes, P. J. (1995). Turing Test considered harmful. In *IJCAI-95*, pp. 972–977.
- [13] Hebb, D. O. (1949). *The Organization of Behavior*. Wiley.
- [14] Shieber, S. (1994). Lessons from a restricted Turing Test. *CACM*, 37, 70–78.
- [15] McCarthy, J., Minsky, M. L., Rochester, N., and Shannon, C. E. (1955). Proposal for

the Dartmouth summer research project on artificial intelligence. Tech. rep., Dartmouth College.

- [16] Clarke, E. and Grumberg, O. (1987). Research on automatic verification of finite-state concurrent systems. *Annual Review of Computer Science*, 2, 269–290.
- [17] Clarke, M. R. B. (Ed.). (1977). *Advances in Computer Chess 1*. Edinburgh University Press.
- [18] Aha, D., Kibler, D., and Albert, M., “Instance-Based Learning Algorithms,” *Machine Learning*, 6, 37-66, 1991.
- [19] Anderson, T. W., *An Introduction to Multivariate Statistical Analysis*, New York: John Wiley, 1958.
- [20] Dietterich, T., “Machine Learning,” *Annu. Rev. Comput. Sci.*, 4:255-306, Palo Alto: Annual Reviews Inc., 1990.
- [21] Holland, J. H., “Escaping Brittleness; The Possibilities of General-Purpose Learning Algorithms Applied to Parallel Rule-Based Systems.” In Michalski, R., Carbonell, J., and Mitchell, T. (eds.) , *Machine Learning: An Artificial Intelligence Approach*, Volume 2, chapter 20, San Francisco: Morgan Kaufmann, 1986.
- [22] Alpaydm, E. 1999. “Combined 5×2 cv F Test for Comparing Supervised Classification Learning Algorithms.” *Neural Computation* 11:1885–1892.
- [23] Yıldız, O. T., and E. Alpaydm. 2006. “Ordering and Finding the Best of $K > 2$ Supervised Learning Algorithms.” *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28:392–402.
- [24] Zeiler, M. D. and Fergus, R. (2014). Visualizing and understanding convolutional networks. In *ECCV’14*.
- [25] Holden, D., Saito, J., Komura, T.: A deep learning framework for character motion synthesis and editing. *ACM Trans. Graph. (TOG)* 35(4), 138:1–138:11
- [26] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778 (2016)
- [27] Hinton, G., Deng, L., Yu, D., Dahl, G.E., Mohamed, A.R., Jaitly, N., et al.: Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. *IEEE Signal Process. Mag.* 29(6), 82–97
- [28] Bengio, Y.: Learning deep architectures for AI. *Found. Trends Mach. Learn.* 2(1), 1–127 (2009)

- [29] Cruz-Roa, A.A., Ovalle, J.E.A., Madabhushi, A., Osorio, F.A.G.: A deep learning architecture for image representation, visual interpretability and automated basal-cell carcinoma cancer detection. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, Berlin, Heidelberg, pp. 403–410, Sept 2013
- [30] Von Melchner, L., Pallas, S. L., and Sur, M. (2000). Visual behaviour mediated by retinal projections directed to the auditory pathway. *Nature*, 404(6780), 871–876. 16
- [31] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2014a). ImageNet Large Scale Visual Recognition Challenge.
- [32] Deng, L., Yu, D.: Deep learning: methods and applications. *Found. Trends Signal Process.* 7(34), 197–387 (2013)
- [33] Collobert, R., Bengio, S., and Bengio, Y. (2001). A parallel mixture of SVMs for very large scale problems. Technical Report IDIAP-RR-01-12, IDIAP. 450
- [34] Werbos, P., Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences, Ph.D. Thesis, Harvard University, 1974.
- [35] Knowles-Barley, S., Jones, T. R., Morgan, J., Lee, D., Kasthuri, N., Lichtman, J. W., and Pfister, H. (2014). Deep learning for the connectome. *GPU Technology Conference*. 26
- [36] Steinkrau, D., Simard, P. Y., and Buck, I. (2005). Using GPUs for machine learning algorithms. 2013 12th International Conference on Document Analysis and Recognition, 0, 1115–1119. 445
- [37] Wani, M.A., Bhat, F.A., Afzal, S., Khan, A.I. (2020). Introduction to Deep Learning. In: *Advances in Deep Learning. Studies in Big Data*, vol 57. Springer, Singapore.
- [38] Wani, M.A., Bhat, F.A., Afzal, S., Khan, A.I. (2020). Introduction to Deep Learning. In: *Advances in Deep Learning. Studies in Big Data*, vol 57. Springer, Singapore.
- [39] Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: *Advances in Neural Information Processing Systems*, pp. 1097–1105 (2012)
- [40] Nguyen, A., Yosinski, J., Clune, J.: Deep neural networks are easily fooled: high confidence predictions for unrecognizable images. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2015*, pp. 427–436
- [41] Sabour, S., Frosst, N., Hinton, G.E.: Dynamic routing between capsules. In: *Advances in Neural Information Processing Systems 2017*, pp. 3859–3869

- [42] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. NIPS Conference, pp. 1097–1105, 2012.
- [43] code.google.com/archive/p/cuda-convnet/
- [44] M. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. European Conference on Computer Vision, Springer, pp. 818–833, 2013.
- [45] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556, 2014.
- [46] A. Karpathy, J. Johnson, and L. Fei-Fei. Stanford University Class CS321n: Convolutional neural networks for visual recognition, 2016.
- [47] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. IEEE Conference on Computer Vision and Pattern Recognition, pp. 2818–2826, 2016.
- [48] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. AAAI Conference, pp. 4278–4284, 2017.
- [49] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778, 2016.
- [50] A. Veit, M. Wilber, and S. Belongie. Residual networks behave like ensembles of relatively shallow networks. NIPS Conference, pp. 550–558, 2016.
- [51] S. Zagoruyko and N. Komodakis. Wide residual networks. arXiv:1605.07146, 2016.
- [52] K. Greff, R. K. Srivastava, and J. Schmidhuber. Highway and residual networks learn unrolled iterative estimation. arXiv:1612.07771, 2016.
- [53] I. Grondman, L. Busoniu, G. A. Lopes, and R. Babuska. A survey of actor-critic reinforcement learning: Standard and natural policy gradients. IEEE Transactions on Systems, Man, and Cybernetics, 42(6), pp. 1291–1307, 2012.
- [54] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. arXiv:1611.05431, 2016.
- [55] G. Huang, Z. Liu, K. Weinberger, and L. van der Maaten. Densely connected convolutional networks. arXiv:1608.06993, 2016.
- [56] Y. LeCun, J. Denker, and S. Solla. Optimal brain damage. NIPS Conference, pp. 598–605, 1990.
- [57] Wani, M.A., Bhat, F.A., Afzal, S., Khan, A.I. (2020). Introduction to Deep Learning. In: Advances in Deep Learning. Studies in Big Data, vol 57. Springer, Singapore.

- [58] Grice, H. P. (1975). Logic and conversation. In P. Cole & J. L. Morgan (Eds.), *Syntax and semantics* (vol. 3): Speech acts (pp. 41–58). New York: Seminar. [Reprinted in Grice, P. (1989). *Studies in the way of words* (pp. 22–40). Cambridge, MA: Harvard University Press.]
- [59] Gaskell, G. (Ed.). (2007). *Oxford handbook of psycholinguistics*. New York: Oxford University Press.
- [60] Swinney, D. A. (1979). Lexical access during sentence comprehension: (Re) Consideration of context effects. *Journal of Verbal Learning and Verbal Behavior*, vol. 18, pp. 645–659.
- [61] Goldstein, E. B. (2008). *Cognitive psychology: Connecting mind, research, and everyday experience* (2nd ed.). Belmont, CA: Thomson Wadsworth.
- [62] Bransford, J. D., Barclay, J. R., & Franks, J. J. (1972). Sentence memory: A constructive versus interpretative approach. *Cognitive Psychology*, vol. 3, pp. 193–209.
- [63] Vosniadou, S. (1994a). Capturing and modeling the process of conceptual change. *Learning and Instruction*, vol. 4, pp. 45–69.
- [64] Ohlsson S., *How the Mind Overrides Experience* , University of Illinois, Chicago, March 2011
- [65] Gonzalez, R. C. and Woods, R. E. [1992]. *Digital Image Processing*, Addison-Wesley, Reading, MA.
- [66] Abidi, M. A. and Gonzalez, R. C. (eds.) [1992]. *Data Fusion in Robotics and Machine Intelligence*, Academic Press, New York.
- [67] T. Scheuermann, G. Pfundt, P. Eyerer, and B. Jähne. Oberflächenkonturvermessung mikroskopischer Objekte durch Projektion statistischer Rauschmuster. In G. Sagerer, S. Posch, and F. Kummert, eds., *Mustererkennung 1995*, Proc. 17. DAGM-Symposium, Bielefeld, 13.-15. September 1995, pp. 319–326. DAGM, Springer, Berlin, 1995.
- [68] R. Szeliski, *Computer Vision: Algorithms and Applications*, Texts in Computer Science, Springer-Verlag London Limited 2011
- [69] Oppenheim, A. V., Schafer, R. W., and Buck, J. R. (1999). *Discrete-Time Signal Processing*. Prentice Hall, Englewood Cliffs, New Jersey, 2nd edition.
- [70] Solomon and Toby Breckon, *Fundamentals of Digital Image Processing A Practical Approach with Examples in Matlab* Chris. 2011 John Wiley & Sons, Ltd.
- [71] B. Jähne, E. Barth, R. Mester, and H. Scharr, eds. *Complex Motion*, Proc. 1th Int. Workshop, Günzburg, Oct. 2004, vol. 3417 of *Lecture Notes in Computer Science*, 2005. Springer, Berlin.

- [72] Aparna Vidyasagar, Stephanie Pappas published October 14, 2021. “*What are bacteria?*”, <https://www.livescience.com/51641-bacteria.html>
- [73] Dr. Mahir E. (2019). “*Gözümüzde Yaşayan Yararlı Bakteriler*” <https://bilimgenc.tubitak.gov.tr>

