



T.C.

ALTINBAŞ ÜNİVERSİTESİ

Lisansüstü Eğitim Enstitüsü

Elektrik ve Bilgisayar Mühendisliği

**DERİN ÖĞRENME İLE ÜÇ BOYUTLU (3B)
KARACİĞER TÜMÖR TESPİTİ**

Fırat ÖZCAN

Doktora Tezi

Danışman

Prof. Dr. Osman Nuri UÇAN

İstanbul, 2023

DERİN ÖĞRENME İLE ÜÇ BOYUTLU (3B) KARACİĞER TÜMÖR TESPİTİ

Fırat ÖZCAN

Elektrik ve Bilgisayar Mühendisli

Doktora Tezi

ALTINBAŞ ÜNİVERSİTESİ

2023

“Fırat ÖZCAN” tarafından hazırlanmış ve 06/04/2023 tarihinde sunulmuş “DERİN ÖĞRENME İLE ÜÇ BOYUTLU (3B) KARACİĞER TÜMÖR TESPİTİ” başlıklı tez Elektrik ve Bilgisayar Mühendisliği Anabilim Dalında Doktora Tezi olarak **oy birliği** ile kabul edilmiştir.

Prof. Dr. Osman Nuri UÇAN

Danışman

Tez Savunma Sınavı Jüri Üyeleri:

Prof. Dr. Osman Nuri UÇAN

Elektrik-Elektronik
Mühendisliği Bölümü,

Altınbaş Üniversitesi

Prof. Dr. Oğuz BAYAT

Elektrik-Elektronik
Mühendisliği Bölümü,

Altınbaş Üniversitesi

Doç. Dr. Adil Deniz DURU

Antrenörlük Eğitimi Bölümü,

Marmara Üniversitesi

Dr. Öğr. Üyesi Sefer KURNAZ

Bilgisayar Mühendisliği
Bölümü,

Altınbaş Üniversitesi

Dr. Öğr. Üyesi Serdar KARGIN

Biyomedikal Mühendisliği
Bölümü,

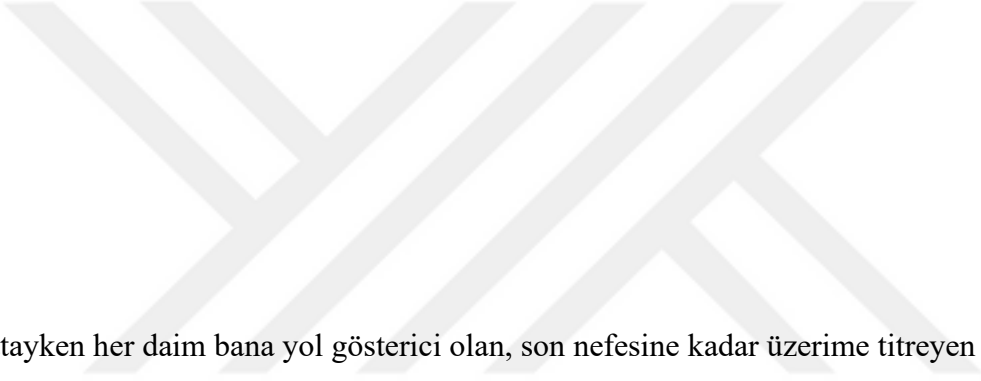
İstanbul Arel Üniversitesi

Bu tezin Doktora tezi olarak bütün şartları sağladığını beyan ederim.

Tezin Lisansüstü Eğitim Enstitüsüne teslim tarihi: ____/____/____//

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.

Fırat ÖZCAN



Hayattayken her daim bana yol gösterici olan, son nefesine kadar üzerime titreyen
ANNEM İÇİN,

ÖNSÖZ

Tez konumun belirlenmesinden bu yana her türlü konuda desteğini ve sabrını sonuna kadar gösteren değerli tez danışmanım Prof. Dr. Osman Nuri UÇAN'a, araştırmalarım sırasında ortaya koyduğu fikirlerle bana destek olan ve yön gösteren Prof. Dr. Gülay Durmuş ALTUN'a, verileri elde etmemde ve işaretlemde emeği geçen Dr. Öğr. Üyesi Ülkü KORKMAZ'a,

Hayatımın her alanında bana değer katan ve sevgileri ile güç katan aile büyüklerim ve kardeşlerime, ayrıca hayatıma girdiğinden beni bana her konuda güvenen ve destek veren eşim Esra ÖCAN ve dünyalar tatlısı oğlum Can ÖZCAN'a,

Eğitim öğretim hayatımda bana değer katan bütün öğretmen ve hocalarıma sonsuz teşekkürlerimi sunarım.

Fırat ÖZCAN

ÖZET

DERİN ÖĞRENME İLE ÜÇ BOYUTLU (3B) KARACİĞER TÜMÖR TESPİTİ

ÖZCAN, Fırat

Doktora, Elektrik ve Bilgisayar Mühendisliği, Altınbaş Üniversitesi

Danışman: Prof. Dr. Osman Nuri UÇAN

Tarih: Nisan/2023

Sayfa: 128

Bilgisayarlı tomografi (BT) görüntülerinde her kesitte meydana gelen şekil, sınır ve yoğunluk değişiklikleri nedeniyle karaciğerin segmentasyonu zor bir işlemdir. Bu çalışmada, karın BT taramalarından karaciğer ve karaciğer tümörlerinin bilgisayar destekli otomatik bölütlenmesi için konvolüsyonel sinir ağları tabanlı Unet ve Inception modellerinin hibritleşmesinden meydana gelen Adding Inception Module-Unet (AIM-Unet) modeli önerilmiştir. Bir tanesi bu tez çalışması için özel olarak hazırlanmış, diğer üçü (CHAOS, LiTS ve 3DIRCADb) halka açık olan toplam dört farklı karaciğer BT görüntü veri seti üzerinde deneysel çalışmalar yapılmıştır. Önerilen yöntem kullanılarak elde edilen sonuçlar ve uzman tarafından işaretlenen bölütleme sonuçları, Zar benzerlik katsayısı (DSC), Jaccard benzerlik katsayısı (JSC) ve doğruluk (ACC) ölçüm parametreleri ile karşılaştırılmıştır. Bu çalışmada, karaciğer görüntülerini içeren veri kümeleri (LiTS, CHAOS ve veri setimiz) önerdiğimiz AIM-Unet modeli ile ayrı ayrı eğitildi. Karaciğer bölütlemesi için en iyi DSC, JSC ve ACC performans ölçümlerini sırasıyla %97.86, %96.10 ve %99.75 ile CHAOS veri setinden elde edildi. Ayrıca, önerilen model LiTS ve 3DIRCADb veri setlerinde tümör bölütlemesi için sırasıyla %75,6 ve %65,5 DSC değeri olarak hesaplandı ve AIM-Unet modeli ile veri kümeleri üzerindeki bölütleme başarı sonuçları önceki çalışmalarla karşılaştırıldı. Bu sonuçlarla bu çalışmada önerilen yöntemin karaciğer segmentasyonu ve karaciğer tümörlerinin tespiti için hekimlerin karar verme süreçlerinde yardımcı bir araç olarak

kullanılabileceđi görölmüştür. Bu çalışma medikal görüntüler için faydalı olup, geliştirilen model farklı organ ve diğeri medikal alanlardaki uygulamalar için kolaylıkla geliştirilebilir.

Anahtar Kelimeler: Derin Öğrenme, Bilgisayarlı Tomografi (BT), Karaciğeri Tümörü, Karaciğeri Bölütlemesi, U-net, AIM-Unet



ABSTRACT

THREE-DIMENSIONAL (3D) LIVER TUMOR DETECTION WITH DEEP LEARNING

ÖZCAN, Fırat

PhD, Electrical and Computer Engineering, Altınbaş University,

Supervisor: Prof. Dr. Osman Nuri UÇAN

Date: April/2023

Pages: 128

The segmentation of the liver is a difficult process due to the changes in shape, border, and density that occur in each section in computed tomography (CT) images. In this study, the Adding Inception Module-Unet (AIM-Unet) model, which is a hybridization of convolutional neural networks-based Unet and Inception models, is proposed for computer-assisted automatic segmentation of the liver and liver tumors from CT scans of the abdomen. Experimental studies were carried out on four different liver CT image datasets, one of which was prepared for this study and three of which were open (CHAOS, LITS, and 3DIRCADb). The results obtained using the proposed method and the segmentation results marked by the specialist were compared with the Dice similarity coefficient (DSC), Jaccard similarity coefficient (JSC), and accuracy (ACC) measurement parameters. In this study, we obtained the best DSC, JSC, and ACC liver segmentation performance metrics on the CHAOS dataset as 97.86%, 96.10%, and 99.75%, respectively, of the AIM-Unet model we propose, which is trained separately on three datasets (LITS, CHAOS, and our dataset) containing liver images. Additionally, 75.6% and 65.5% of the DSC tumor segmentation metrics were calculated on the proposed model LITS and 3DIRCADb datasets, respectively. In addition, the segmentation success results on the datasets with the AIM-Unet model were compared with the previous studies. With these results, it has been seen that the

method proposed in this study can be used as an auxiliary tool in the decision-making processes of physicians for liver segmentation and detection of liver tumors. This study is useful for medical images, and the developed model can be easily developed for applications in different organs and other medical fields.

Keywords: Deep Learning, Computed Tomography, Liver Tumor, Liver Segmentation, U-Net, AIM-Unet



İÇİNDEKİLER

Sayfa

ÖZET	vii
ABSTRACT.....	ix
TABLO LİSTESİ.....	xiv
ŞEKİL LİSTESİ.....	xv
KISALTMALAR	xx
1. GİRİŞ.....	1
1.1 PROBLEM TANIMI.....	3
1.2 TEZİN AMACI	3
1.3 ARAŞTIRMA KAPSAMI.....	4
2. LİTETATÜR TARAMASI.....	5
3. YAPAY SİNİR AĞLARI.....	12
3.1 YAPAY SİNİR AĞLARININ TARİHÇESİ.....	12
3.2 YAPAY ZEKÂ VE YAPAY SİNİR AĞLARININ UYGULAMA ALANLARI	18
3.2.1 Görüntü Sınıflandırma ve Yerini Belirleme	18
3.2.2 Nesne Algılama (Object Detection)	18
3.2.3 Semantik Bölümleme (Semantic Segmentation).....	19
3.2.4 Sahne Bölümleme (Scene Segmentation).....	19
3.2.5 Resim Altyazısı Oluşturma(Image Captioning)	19
3.2.6 Sağlık ve Sağlık Hizmetlerinde	19
3.2.7 Gazetecilik Endüstrisinde	20
3.2.8 Diğer Alanlar	21
4. DERİN ÖĞRENME	23
4.1 EVRİŞİMSEL SİNİR AĞLARI	23
4.1.1 ESA'yı Oluşturan Katmanlar.....	25

4.1.1.1	Giriş katmanı.....	25
4.1.1.2	Evrişim (Konvolüsyon) katmanı.....	26
4.1.1.3	Havuzlama (Boyut indirgeme) katmanı.....	29
4.1.1.4	Tam bağlantı katman	31
4.1.1.5	Seyreltme (DropOut) katmanı	32
4.1.1.6	Çıktı katmanı.....	33
4.2	DERİN ÖĞRENME MODELLERİ	34
4.2.1	LeNet-5.....	34
4.2.2	AlexNet.....	40
4.2.3	GoogLeNet	45
4.2.4	VGG-16	49
4.2.5	U-Net	52
4.2.6	ResNet	53
4.3	DERİN ÖĞRENME MODELLERİNDE SIK KULLANILAN HİPER-PARAMETRELER.....	56
4.3.1	Veri Kümesinin Boyutu.....	58
4.3.2	Batch Boyutu	59
4.3.3	Öğrenme Oranı ve Momentum Katsayısı.....	62
4.3.4	Optimizasyon Algoritmasının Seçimi	63
4.3.5	Eğitim Tur (Epoch) Sayısı.....	67
4.3.6	Ağırlık (Weight) Başlangıç Değerlerinin Belirlenmesi.....	68
4.3.7	Aktivasyon Fonksiyonu.....	68
4.3.7.1	Sık kullanılan aktivasyon fonksiyonları	70
5.	MATERYAL ve YÖNTEM.....	77
5.1	DERİN ÖĞRENME MODELİ GELİŞTİRME.....	77
5.2	INCEPTION MODULE PART (IMP).....	80
5.3	VERİ SETLERİ.....	84
5.3.1	Hastalar ve Veri edinme protokolü.....	84

5.3.2	BT Edinme Protokolü	85
5.3.3	Veri Setin Oluşturulması	85
5.3.4	LITS Görüntü Veri Seti	86
5.3.5	CHAOS Görüntü Veri Seti	86
5.3.6	3DIRCADb Görüntü Veri Seti	87
5.4	VERİ HAZIRLAMA VE İŞLEME	87
5.5	İŞLEM SONRASI ADIMLAR	90
6.	DENEY SONUÇLARI VE ANALİZ	91
7.	SONUÇ ve ÖNERİLER.....	103
8.	KAYNAKLAR.....	104

TABLO LİSTESİ

Sayfa

Tablo 4.1: LeNet'te katmanlardaki özellik haritaları arasındaki bağıntı.	37
Tablo 4.2: LeNet mimarisi özet tablosu.	40
Tablo 4.3: VGG-16 mimarisinin özeti.	51
Tablo 5.1: AIM-Unet için giriş, çıkış, filtre sayısı ve filtre boyutları	79
Tablo 5.2: Kullandığımız bilgisayarın özellikleri.	82
Tablo 5.3: Veri setlerindeki görüntülerin dağılımları.	84
Tablo 5.4: Veri artırma sonrasında elde edilen görüntüler.	88
Tablo 6.1: AIM-Unet'in veri setleri üzerindeki performansı.	93
Tablo 6.2: AIM-Unet'in test setleri üzerindeki çapraz performansı.	93
Tablo 6.3: Karaciğer için önerilen yöntemin (AIM-Unet) diğer mimarilerle karşılaştırılması.	98
Tablo 6.4: Karaciğer tümörü için önerilen yöntemin (AIM-Unet) diğer mimarilerle karşılaştırılması.	99

ŞEKİL LİSTESİ

Şekil 1.1: 2012 yılı Dünya geneli kadın ve erkeklerdeki kanser çeşitleri ve yakalanan kişi sayıları [1].	1
Şekil 1.2: 2012 yılı Dünya geneli kadın ve erkeklerdeki kanser türlerine göre ölü sayıları [1].....	2
Şekil 2.1: Akciğer kanser teşhis yöntemi blok diyagramı [10].	5
Şekil 2.2: ROI saptayan 8 eksenli süzgeç [10].....	6
Şekil 2.3: 3B morfolojik süzgeç [10].	6
Şekil 2.4: a) Akciğer BT verisi b) HYSA ile segmentasyonu sonucu c) BT ham veri ve HYSA çıktısının AND'lenmiş hali d) 8 eksenli süzgeç çıktısı: Akciğer ROI'si e) Kestirilen nodül [10].....	7
Şekil 2.5: Giriş görüntüsü (BT karın görüntüsü) [13].	8
Şekil 2.6: Bilgisayar ekran görüntüsü [13].	8
Şekil 2.7: Ekran görüntüsünün histogramı [13].	9
Şekil 2.8: Ayıklama yapıldıktan sonraki histogram [13].	9
Şekil 2.9: Karaciğer segmantasyonu [13].	9
Şekil 2.10:Tümör segmantasyonu [13].	10
Şekil 2.11: Tümör tespiti için oluşturulan DESA [19].....	11
Şekil 3.1: Perceptron yapısı.	14
Şekil 3.2: XOR problemi.	15
Şekil 3.3: Çok katmanlı algılayıcı şeması.	16
Şekil 3.4: XOR probleminin çözümü.....	17
Şekil 3.5: Sinir ağlarının kısa tarihçesi.	18
Şekil 4.1: Yapay zekânın süreçleri [30].	23

Şekil 4.2: Evrişimsel sinir ağının genel mimarisi [32].....	25
Şekil 4.3: Renkli girdiler için kanal, genişlik ve yükseklik kavramlarının gösterimi.	26
Şekil 4.4: Tek bir konvolüsyon katmanının gösterimi [33].	27
Şekil 4.5: 5x5x3 boyutundaki bir giriş verisine 3x3'lük filtrenin uygulandığı konvolüsyon işlemi [15]....	28
Şekil 4.6: ESA için giriş görüntüsü [32].	29
Şekil 4.7: ESA'nın konvolüsyon katmanından sonraki oluşan görüntüsü [32].	29
Şekil 4.8: Maksimum, minimum ve ortalama havuzlama.....	30
Şekil 4.9: 5x5'lik giriş görüntüsüne 2x2 filtre ile bir ve iki adım kaydırmalı maksimum havuzlama işleminin uygulanması [32].....	31
Şekil 4.10: Seyreltme katmanının Uygulanması [32].	33
Şekil 4.11: LeNet-5 yapısı.	35
Şekil 4.12: C1:Konvolüsyon katmanı.	35
Şekil 4.13: S2:Havuzlama katmanı.	36
Şekil 4.14: C3:Evrişim Katmanı.	37
Şekil 4.15: Ortalama havuzlama katmanı.	38
Şekil 4.16: Tam bağlı evrişim katmanı.	38
Şekil 4.17: Tam bağlı katman.	39
Şekil 4.18: Tam bağlı çıkış katmanı.....	39
Şekil 4.19: AlexNet mimarisi [35].....	41
Şekil 4.20: AlexNet yapısı [36].	42
Şekil 4.21: Konvolüsyon katmanı.....	43

Şekil 4.22: AlexNet'in ikinci katmanı.	44
Şekil 4.23: AlexNet 6. Katman.	45
Şekil 4.24: Başlangıç bloğunun yapısı [38].	46
Şekil 4.25: GoogLeNet mimarisi [38].....	47
Şekil 4.26: VGG-16 mimarisi [39].	50
Şekil 4.27: U-Net mimarisi [41].	52
Şekil 4.28: U-Net Encoding(sol taraf) ve Decoding(sağ taraf) İşlemleri [42].	53
Şekil 4.29: Residual bloğu [43].	54
Şekil 4.30: ImageNet için örnek ağ mimarisi. Solda VGG-19 modeli[39]. Ortada 34 parametre katmanı plain ağı ve sağda 34 parametre katmanı residual ağı [44].	55
Şekil 4.31: Yapay zeka uygulamalarında parametre ve hiper parametre.	57
Şekil 4.32: Batch boyutu değiştirildiğinde hata değerinin değişimi [45].....	60
Şekil 4.33: MNIST seti üzerindeki çeşitli optimizasyon algoritmalarının çalışma zamanı grafiği [51].	65
Şekil 4.34: Optimizasyon algoritmalarının hız performans gösterimi [52].	65
Şekil 4.35: (a) Sigmoid fonksiyonu. (b) Sigmoid fonksiyonu gradyanı.....	71
Şekil 4.36: (a) <i>Tanh</i> fonksiyonu. (b) <i>Tanh</i> fonksiyonu gradyanı.	72
Şekil 4.37: (a) ReLU fonksiyonu. (b) ReLU fonksiyonu gradyanı.	73
Şekil 4.38: (a) Sızıntı ReLU fonksiyonu. (b) Sızıntı ReLU fonksiyonu gradyanı.	74
Şekil 4.39: Swish fonksiyonu ve gradyanı.....	75
Şekil 4.40: Aktivasyon fonksiyonları.....	76
Şekil 5.1: AIM-Unet modelinin şematik gösterimi.....	77

Şekil 5.2: AIM-Unet mimarisi.	78
Şekil 5.3: Inception modül kısmının diyagramı (IMP).	81
Şekil 5.4: Önerdiğimiz AIM-Unet modeli.	83
Şekil 5.5: Yukarıdaki şekilde bir hastanın herhangi bir kesit görüntüsünün RAW hali (a), filtreleme yapılmış hali (b) ve uzman radyolog tarafından etiketlenmesi yapılan görüntü (c) yer almaktadır.	89
Şekil 5.6: Uzman radyolog tarafından etiketlenmiş karaciğer (gri renkli bölge) ve tümörlü alan (beyaz renkli bölge).	89
Şekil 5.7: Yukarıdaki şekilde bir hastanın herhangi bir kesit görüntüsünün RAW hali (a), uzman tarafından işaretlenmiş karaciğer alanı (b) ve yine uzman radyolog tarafından etiketlenmesi yapılan tümörlü alan (c) yer almaktadır.	90
Şekil 6.1: Tahmin sonuçları karşılaştırırken gerçek pozitif, gerçek negatif, yanlış pozitif ve yanlış negatif bölgelerin grafik gösterimi.	91
Şekil 6.2: Yukarıda üç farklı veri seti üzerinde eğittiğimiz AIM-Unet modelinin her epoch esnasındaki Zar katsayısı, kayıp miktarı ve IoU değerlerinin değişimini gösteren grafikler yer almaktadır.	94
Şekil 6.3: Yukarıda eğitim sırasındaki birinci konvolüsyon katmanındaki 64 adet 3x3 ve dördüncü katmanındaki 64 adet 256x256 filtreleri yer almaktadır.	96
Şekil 6.4: Önerilen modelin (AIM-Unet) veri setlerinin görüntüleri üzerindeki performansı. Karşılaştırma görüntüsündeki kırmızı alanlar modelimizin kaçırdığı bölgeleri, yeşil alanlar modelin yanlış tahmin ettiği alanları ve sarı alanlar modelin doğru tahmin ettiği bölgeleri göstermektedir.	97
Şekil 6.5: Üç boyutlu elde ettiğimiz bir karaciğerin farklı açılardan kesit görüntüleri.	98
Şekil 6.6: Önerilen modelin (AIM-Unet) veri setlerinin görüntüleri üzerindeki performansı. Karşılaştırma görüntüsündeki yeşil alanlar, modelimizin karaciğer alanı için kaçırdığı bölgeleri, mavi alanlar ise tümör alanı için modelin kaçırdığı bölgeleri göstermektedir.	99
Şekil 6.7: Üstteki resim uzman doktorun işaretlediği 3 boyutlu karaciğer kısmı, alttaki görüntü ise önerdiğimiz modelin 3 boyutlu tahmin ettiği karaciğer görüntüsünü göstermektedir.	101

Şekil 6.8: LITS veri setindeki 104 ve 108 nolu hastaların uzmanlar tarafından işaretlenmiş görüntüleri (üst satırdakiler) ve modelimizin tahmin ettiği görüntüler (alt satırdakiler). 102

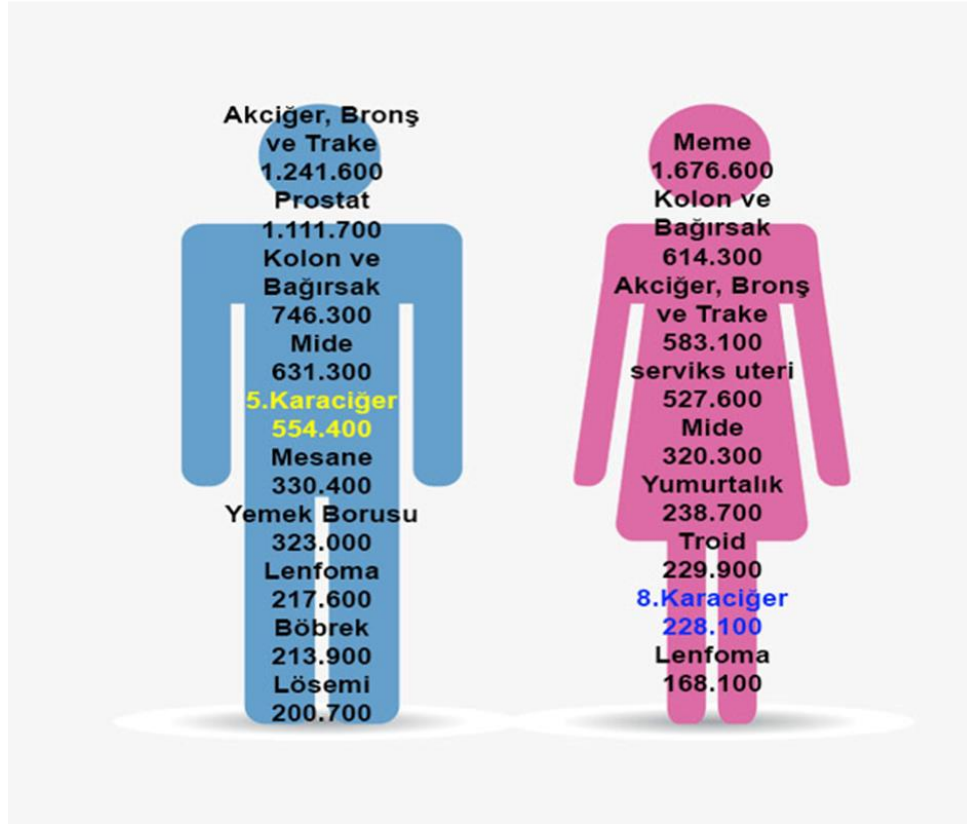


KISALTMALAR

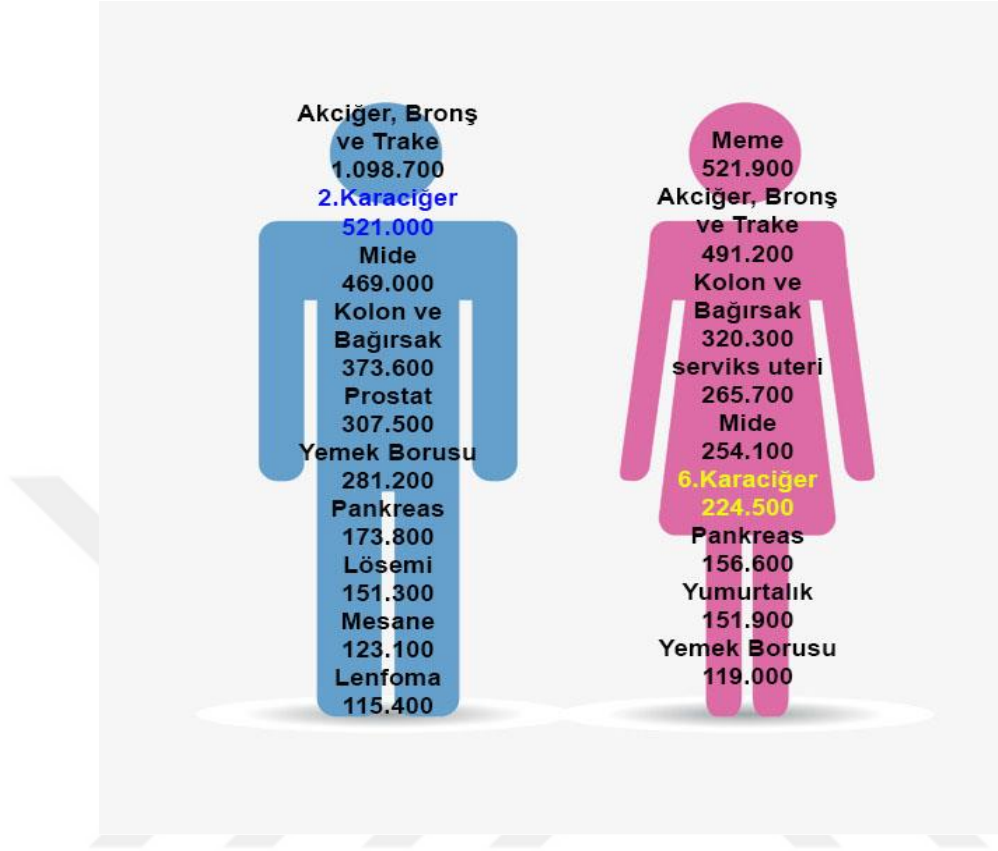
AI	: Artificial Intelligence
YSA	: Yapay Sinir Ağı
EYSA	: Evrişimsel Yapay Sinir Ağı
ESA	: Evrişimsel Sinir Ağı
YZ	: Yapay Zekâ
BT	: Bilgisayarlı Tomografi
HYSA	: Hücresel Yapay Sinir Ağları
GT	: Graph Cut
3B	: ÜÇ BOYUTLU

1. GİRİŞ

İnsan vücudundaki organlar büyükten küçüğe doğru sıralandığında ikinci sırayı karaciğer almaktadır. Karaciğer hastalıkları, insanların sağlığı ve yaşamları için ciddi bir tehdit oluşturmaktadır. Karaciğer kanseri, erkeklerde en sık ikinci kanser ölüm nedeni iken kadınlarda kanser ölümleri nedenleri arasında altıncı sırada olduğu görülmüştür. 2012 yılında dünya çapında yaklaşık 782.500 yeni karaciğer kanseri vakasına rastlanmış olup (Şekil 1.1 ve Şekil 1.2) bu vakaların 745.500'ü ölüm ile sonuçlanmıştır [1]. Dünya sağlık örgütünün 2018 yılında yaptığı araştırmaya göre ise yılda yaklaşık 841.000 yeni vaka ve bu vakaların 782.000'i ölümlle sonuçlanan dünya çapında altıncı kanser ve kansere bağlı ölümlerde ise dördüncü sıraya karaciğer kanseri yer almaktadır. Ayrıca dünyanın birçok bölgesinde erkeklerde yakalanma oranı kadınlara göre 2 ila 3 kat daha fazladır. Bu sebeple karaciğer kanseri, küresel vakalar açısından beşinci ve erkek ölümleri açısından ikinci sırada yer almaktadır [2].



Şekil 1.1: 2012 yılı Dünya geneli kadın ve erkeklerdeki kanser çeşitleri ve yakalanan kişi sayıları [1].



Şekil 1.2: 2012 yılı Dünya geneli kadın ve erkeklerdeki kanser türlerine göre ölü sayıları [1].

Tıbbi görüntülemelerde, bir hastalığın doğru teşhisi ve değerlendirmesi hem görüntü edinimi hem de görüntü yorumlamasına bağlıdır. Görüntü elde etme, son yıllarda daha hızlı veri elde eden ve daha yüksek çözünürlükte veri elde eden cihazlarla büyük ölçüde iyileşmiştir. Bununla birlikte, görüntü yorumlama süreci, bilgisayar teknolojisinden sadece son zamanlarda faydalanmaya başlamıştır [3]. Karaciğer kanseri teşhisinde çeşitli medikal görüntüleme (MRG, PET, v.b.) ve laboratuvar testleri yapılmaktadır. Günümüzde en gelişmiş medikal görüntüleme çeşitlerinden biri de BT çekimidir. BT (Bilgisayarlı Tomografi) görüntülerinden karaciğerin ayrılması, canlı donör karaciğer transplantasyonu, ışın tedavisi ve hacim ölçümü için cerrahi planlama gibi bilgisayar destekli bazı klinik müdahalelerde önemli bir adımdır. Bu çalışmada, karın BT görüntüleri kullanılacaktır. Kullanılacak olan BT görüntülerinden otomatik olarak karaciğer ve tümörünü segmentlere ayıracak derin bir öğrenme algoritması geliştirmeyi amaçlıyoruz.

1.1 PROBLEM TANIMI

BT taramaları, MRI, Ultrason gibi görüntülerin analizi, uzmanların teşhis, tedavi, planlama, ameliyat ve takip için sıkça başvurdukları bir yöntemdir. Tıbbi görüntü bölütleme, görüntülerden istenilen alanların çıkarılması istenmeyen bölge veya organların ise atılması anlamına gelmektedir. Karaciğer ve tümör bölütlemesi, tıbbi görüntü metotlarından herhangi birine başvurarak karaciğer ve tümörünün yerini, sınırlarını ve şeklini otomatik bir şekilde çıkarmayı hedefler; BT taramalarında işaretlenmiş olan görüntüde karaciğer veya tümörlü kısım 1 (beyaz) ve karaciğer ya da tümör içermeyen alanlar ise sıfır (siyah) olarak görüntülenir. Karaciğer ve tümör bölütlemesi, karaciğer hastalıklarının teşhis, tedavi ve takip süreçlerinde önemli rol almaktadır. Karaciğeri ve tümörleri bölümlere ayırmak, klinik doktorlara karaciğer ve tümörlerin konumu, şekli ve boyutu hakkında net bir görüş sağlar; bu ayrıntılar, tedavi yöntemine ve dozuna karar vermek için gereklidir.

Özellikle son yıllarda ülkemizin de üzerinde titizlikle durduğu kişisel verilen korunması kanunu kapsamında tıbbi görüntü elde etmek çok zor bir durum almıştır. Durum böyle olunca tıbbi veri seti oluşturmak ve araştırmalar yapmak meşakkatli bir hal almaktadır. Araştırmaları için mevcut karaciğer veri setlerinin sayısı oldukça azdır. Karaciğer ve tümörü için manuel olarak açıklamalı işaretlenmiş veri oluşturmak, klinik uzmanlık gerektiren zaman alıcı bir görevdir. Manuel olarak işaretlenmiş veriler, model eğitimi, doğrulama ve test için çok büyük öneme sahiptir.

Segmentasyon doğruluğu, özellikle karaciğer için tıbbi görüntü segmentasyonunda hala açık bir sorundur. Ayrıca şuana kadar geliştirilen hiçbir yöntem tüm organları başarılı bir şekilde segmentlere ayırmış değildir. BT taramalarından karaciğer için geliştirilen aynı yöntemle bile, farklı veri kümelerinde modellerin doğruluğuna göre değişiklik göstermektedir.

1.2 TEZİN AMACI

Üç boyutlu (3B) medikal görüntülerden doğru karaciğer segmentasyonu yardımı ile hepatik hastalıkların patolojik tanısı gibi birçok klinik uygulamada, cerrahi planlama ve postoperatif değerlendirilmesinde önemli bir yer teşkil etmektedir. Ancak otomatik karaciğer segmentasyonu, tıbbi görüntülerde karaciğer görünümünü ve sınırlarını belirlemek, karmaşık arka plan ve karaciğerin kişilerde farklı büyüklükte olmasından dolayı hala zorlu bir görevdir.

Son yıllarda karaciğer segmentasyonu ile ilgili birçok çalışma yapıp yayınlanmıştır. Bu çalışmaların birçoğunda, karaciğerin boyutunu belirlemede ve tümörlü hücrelerini tespit etmede yüksek oranda başarı sağlanmıştır. Çağdaş dünyada bilgisayar ve bilgisayar sistemleri hayatımızın vazgeçilmez birer parçası haline gelmiştir. Evimizdeki çoğu elektrikli alette, özellikle elimizden düşürmediğimiz cep telefonlarında akıllı kelimesi kullanılmaktadır. Çünkü artık çoğu cihazda Yapay zekâ (YZ) teknolojileri kullanılmaktadır.

Görüntülerden tümörlerin teşhisi doktorlar için çok zor ve fazla süre almaktadır ve ayrıca hata yapma olasılığı görülebilmektedir. Doktorların teşhis becerilerini geliştirmek kesin ve doğru teşhis için harcanan sürelerini azaltmak, bilgisayar destekli otomatik tespit ve teşhis sistemleri ile mümkün olmaktadır. Bu sistemlerin amacı, uzmanların daha hızlı ve doğru karar verebilmelerine yardımcı olmaktır [4].

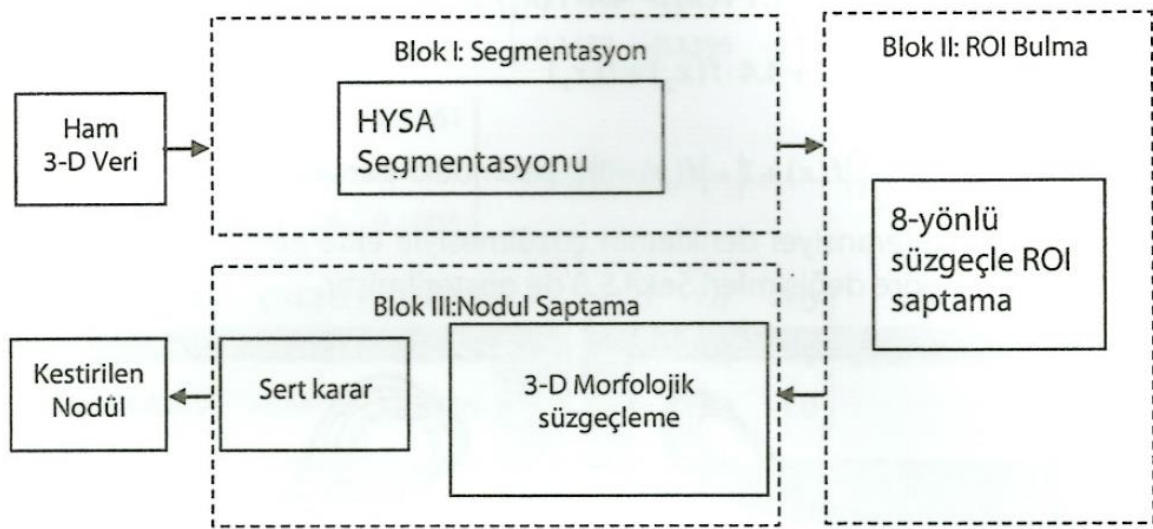
Bu tezin amacı, YZ problemlerinde çok fazla kullanılan bir yöntem olan makine öğrenmenin bir sınıfı olan derin öğrenme yaklaşımını medikal görüntüleme alanına uygulamak, kullanılacak olan 50 hastanın BT sonuçlarından optimal karaciğer ve tümörlü hücrelerin segmentasyonunu farklı derin öğrenme modelleri ile çıkarmaktır. Bu sayede cerrahın tümörü görmesini, büyüklüğü hakkında bilgi sahibi olmasını ve tedaviyi kolaylaştırabilmesini sağlayacaktır. Ayrıca geliştirilecek bu model ile teşhis için tıbbi veri analizini daha kısa sürede sağlamayı ve deneyimsizlik veya yorgunluk sebebi ile meydana gelen insan hatalarını engellemeyi hedeflemektedir.

1.3 ARAŞTIRMA KAPSAMI

Araştırma, ana tıbbi görüntü olarak yalnızca BT taramalarına odaklandı ve başka herhangi bir yöntemi araştırmadı, örneğin PET-BT veya MRI çünkü BT taramaları karaciğer tümörlerinin teşhis ve tedavi sürecinde yaygın olarak kullanılmaktadır. Tez çalışmasında, Karaciğer ve tümör segmentasyonuna odaklanmıştır. Ancak karaciğer ve tümörü dışında farklı bir organ araştırılmamıştır. Diğer organlar, örneğin böbrek ve kan damarları, veri setlerinin yetersizliği ve karaciğer ile diğer organlar arasında özellikler, doku ve şekil açısından önemli farklılıklar nedeniyle dâhil edilmemiştir. Karaciğer tümörleri hastanedeki verilerde olmadığı için halka açık olan ve internet üzerinden kolayca erişilebilen iki veri seti üzerinde araştırılmıştır.

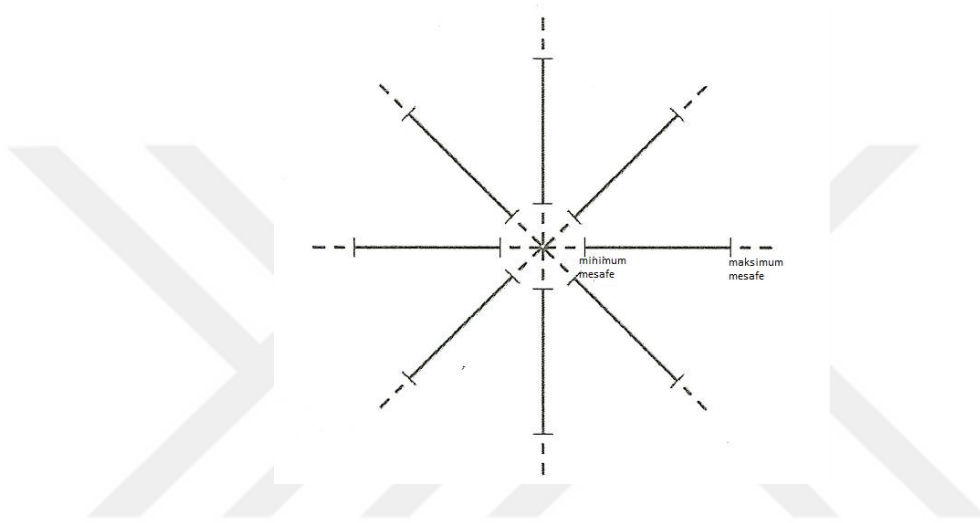
2. LİTETATÜR TARAMASI

Karaciğer segmentasyonu ve karaciğer tümör teşhisi için çok sayıda teknik uygulanmış ve sonuçları sunulmuştur. Özellikle kontrastlı BT görüntülerindeki damarları segmente etmek için özel yöntemler önerilmiştir[5],[6],[7],[8]. Karaciğer segmentasyonu için kullanılan metotlar iki kategori altında sınıflandırılabilir: Düşük seviyeli bilgi tabanlı metotlar ve öncül bilgi destekli metotlar[9]. Düşük seviyeli bilgi tabanlı metotlar genellikle voksellerin(bir pikselin üç boyutlu görünümü) belirlenmesi için görüntü gri tonlama, gradiyent ve doku kullanılmaktadır. Eşikleme yolu ve GT (graph cutting) ile sınırların belirlenmesi sağlanır. Ön ve son işlem adımları çoğu zaman, alt bölümlendirmeyi hafifletmek için morfolojik operatörler ile uygulanır. Birçok çalışmada Tıbbi Görüntü Hesaplama ve Bilgisayar Destekli Müdahale konferansında (Medical Image Computing and Computer Assisted Intervention-MICCAI) veri seti kullanılmıştır. Osman vd. 2006 yılında yayımladığı “Görüntü İşleme Teknikleri ve Mühendislik Uygulamaları” kitabında HYSA (Hücrel Yapay Sinir Ağları) yönteminin biyomedikal görüntülere uygulaması başlıklı bölümde HYSA’nın kullanıldığı biyomedikal alanları hakkında bilgi vermiştir. Yaptığı çalışmada akciğer kanserinde bölge teşhisine dikkat çekmiştir. “Lung Image Database Consortium (LIDC)” veri setine uyguladığı yöntem Şekil 2.1’teki gibidir.



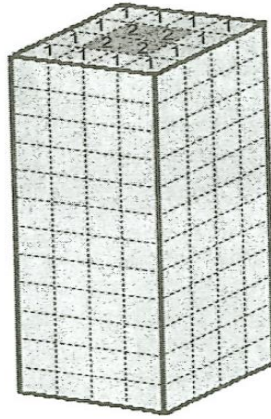
Şekil 2.1: Akciğer kanser teşhis yöntemi blok diyagramı [10].

Görüntü işlemede temel alınacak alanın sınırlarını belirlemek için 3B alınan BT (Bilgisayar Tomografi)'ler HSYA algoritması kullanılarak segmentasyona tabi tutmuştur. Sonrasında ROI (Region of Interest) [11], saptaması için 8 yönlü bir süzgeçten (Şekil 2.2) geçirmiştir [10].



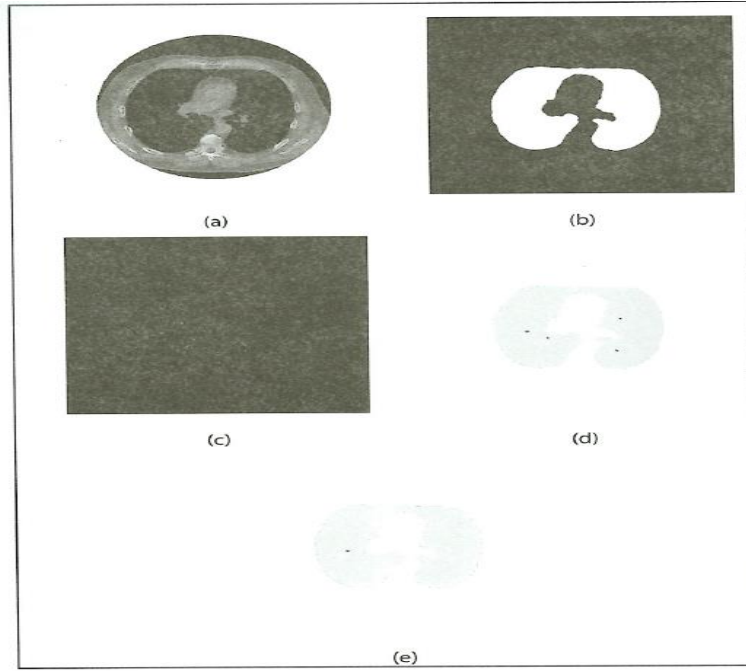
Şekil 2.2: ROI saptayan 8 eksenli süzgeç [10].

ROI belirlendikten sonra 3B'li özel bir süzgeçten geçirmiştir. Bu süzgeç olası kanserli bölgeleri kuvvetlendirirken, sağlam olan bölgeleri zayıflatmıştır (Şekil 2.3). Amaç morfolojik özelliklere uygun 3B nodül yapısına uygunluk aranmıştır.



Şekil 2.3: 3B morfolojik süzgeç [10].

Üç veri seti üstünde denenmiş olan bu yöntem başarılı sonuçlar veriş olup uygulama adımlarının görünüşleri Şekil 2.4'teki gibidir [10].

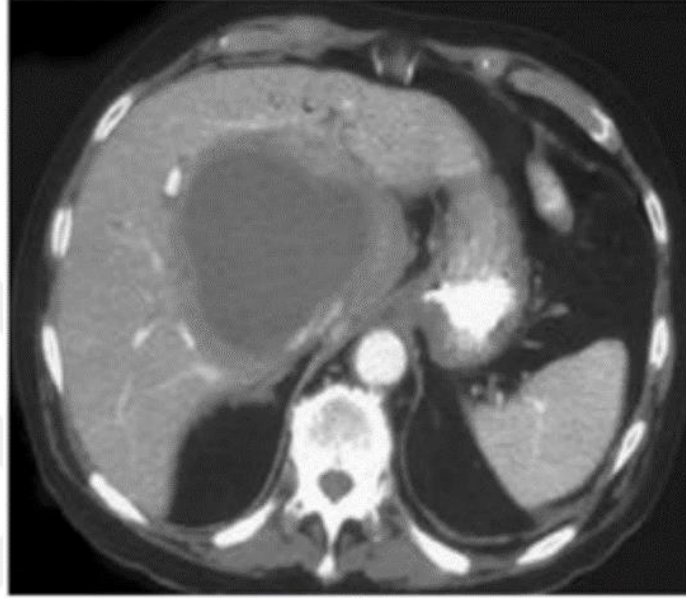


Şekil 2.4: a) Akciğer BT verisi b) HYSA ile segmentasyonu sonucu c) BT ham veri ve HYSA çıktısının AND'lenmiş hali d) 8 eksenli süzgeç çıktısı: Akciğer ROI'si e) Kestirilen nodül [10].

Zheng vd. 2007 yılında yapmış olduğu çalışmada karaciğer BT'nin hem anatomik bilgisine hem de özelliklerine dayanan eşik değerlerini seçmek için yeni bir yöntem geliştirmiştir [12]. Karaciğeri diğer yapışık hücrelerden ayırmak için morfolojik operatörler uygulanmıştır. Bu işlemde sonra filtreleme yaparak karaciğerin üç boyutlu son halini elde etmiştir. Zheng vd.'lerinin geliştirmiş olduğu bu yöntem 45 hastaya uygulamıştır ve elde edilen karaciğer segmentasyonlarında %87.7 oranında başarı sağlamıştır.

Mala vd. 2008 yılında yaptığı çalışmada, karaciğer bilgisayarlı tomografi (BT) görüntülerini kullanarak şüpheli tümör bölgelerini otomatik çıkarmıştır. Elde ettikleri dokusal bilgileri olasılıksal sinir ağını (Probabilistic Neural Network) eğitmekte kullanıp tümör hücrelerini sınıflandırmışlardır. Elde ettikleri sonuçları radyologlar ile değerlendirdiklerinde, sistemin tümör hücrelerini yüksek bir oranda doğru tespit edildiği görülmüştür [13]. Tümör bölgelerini belirlemek için Fuzzy C Means Clustering (FCM)'yi kullanmıştır. Çıkarılan tümörlü bölgelere biortogonal dalgacık dönüşümü (Biortogonal wavelet transform) yatay, dikey ve diyagonal olarak

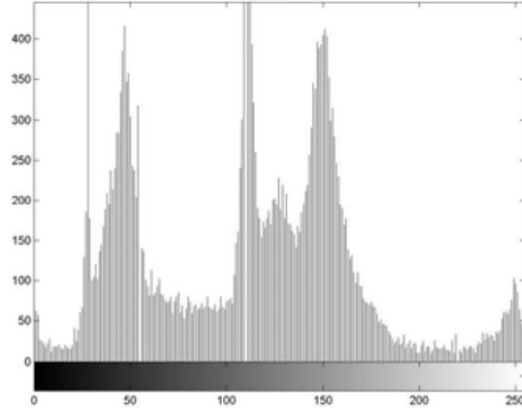
uygulanmıştır. Bu sayede istatistiksel özellikler veya eş oluşum özellikleri elde edilmiştir. Matlab'taki mevcut olan fonksiyonlar yardımı ile DICOM (Digital Imaging and Communications in Medicine) formatındaki görüntülere uygulanmıştır. Elde edilen görüntüler aşağıdaki gibidir.



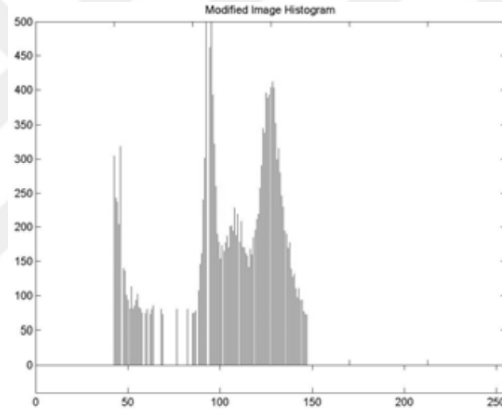
Şekil 2.5: Giriş görüntüsü (BT karın görüntüsü) [13].



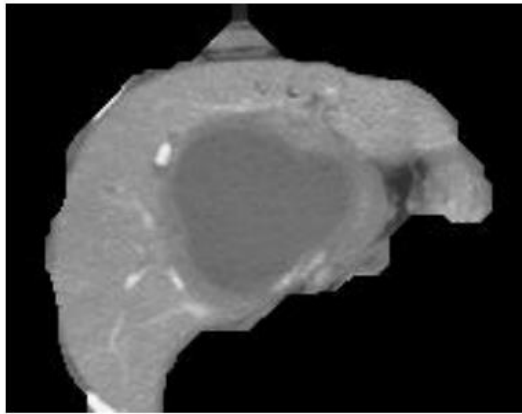
Şekil 2.6: Bilgisayar ekran görüntüsü [13].



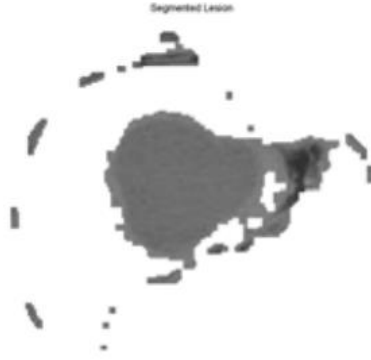
Şekil 2.7: Ekran görüntüsünün histogramı [13].



Şekil 2.8: Ayıklama yapıldıktan sonraki histogram [13].



Şekil 2.9: Karaciğer segmentasyonu [13].



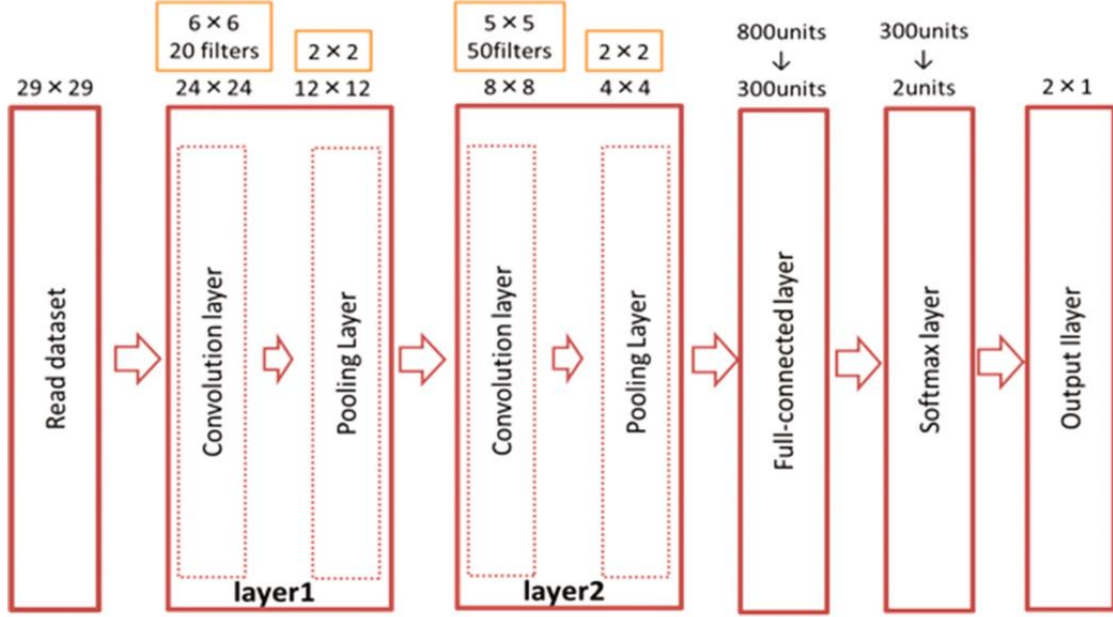
Şekil 2.10: Tümör segmentasyonu [13].

Linguraru ve diğ. 2010 [14] yılında yaptığı çalışmada olasılıksal atlas yöntemi ile Karaciğer ve dalak BT görüntüleri kullanarak bilgisayar destekli otomatik karaciğer ve dalak bölümlerini bölütlemeye çalıştı. Arı ve Hanbay'ın 2012 yılında yapmış olduğu çalışmada MR görüntülerinden tümör tespit edilmiş ve tümörün yerini belirleyen bir bilgisayarlı destekli sistem önerilmiştir [15]. Bilgisayar destekli tümör tespit sisteminde kullanılan mimari ESA kullanan derin öğrenme mimarisidir. Çalışma sonucunda ise farklı mimariler tasarlanarak veri setleri üzerinde test sonuçları %99.10 doğruluk değeri elde edilmiştir.

Bilgisayar destekli hepatit hatalıkların tanısını koymak ve tedavisi için yapmak için BT hacimlerinden otomatik karaciğer segmentasyonu önemli bir ön koşuldur. 2016 yılında yapılan bir çalışmada yeni bir üç boyutlu derin denetimli ağ (3B DSN) geliştirilmiştir [16]. Geliştirilen 3B DNS, uçtan uca öğrenme ve çıkarım yapan tam evrışimsel bir mimariden faydalanmıştır. Daha da önemlisi, öğrenme sürecinde potansiyel optimizasyon zorluklarıyla mücadele etmek için derin bir denetim mekanizması sunmuş ve bu nedenle model daha hızlı bir yakınsama oranı ve daha güçlü bir ayırmacılık yeteneği kazanabilmiştir. MICCAI-SLiver07 veri seti üzerinde uygulanan 3B DSN oldukça başarılı sonuçları ortaya koymuştur.

Todoriki vd. 2017 yılında yaptığı çalışmada Derin Evrışimsel Sinir Ağlarını (DESA) kullanarak karaciğer tümör bölgesini ve çeşidini tespit etmeye çalıştı. Önerdiği tümör tespit yöntemi iki adımdan meydana gelmiştir. İlk adım, geliştirdiği karaciğer segmentasyon algoritması kullanarak karaciğer BT görüntüsünü segmente etmiştir [17],[18]. İkinci adım ise, ESA ağı kullanarak her bir

pikselin tümör olma olasılığını hesaplamaktır. Kullanılan ağda iki tane evrişimsel katman, iki adet havuzlama (seyreltme) katmanı ve bir adet tam bağlantı katmanı vardır. Evrişim katmanları yararlı özellikleri çıkarmak için kullandı. Havuz katmanları, özelliklerin uzamsal çeşitliklerini en aza indirmek için kullandı. Son olarak tam bağlı katman ise sınıflandırma için kullandı. Kullandığı Derin Evrişimsel Sinir Ağ'ın yapısı Şekil 2.11'de olduğu gibidir [19].



Şekil 2.11: Tümör tespiti için oluşturulan DESA [19].

3. YAPAY SİNİR AĞLARI

Çağdaş dünyada bilgisayar ve bilgisayar sistemleri yaşantımızın vazgeçilmez birer parçaları haline gelmiştir. Önceleri sadece elektronik veri transferi yapmak ve karmaşık hesaplamalar yapmak için kullanılan bilgisayarlar şimdilerde çok büyük verileri filtreleyerek özet istenilen verileri çıkartabilen ve mevcut bilgileri kullanarak durumlar hakkında yorum yapan nitelikler kazandığı görülmüştür. Bilgisayar bilimleri ile ilgilenen araştırmacılar için her zaman önemli bir çalışma konusu olan yapay zekâ sistemlerini tasarlariken insan gibi düşünen, insanların sorunlara bakış açısı ile bakabilen ve onlar gibi çözün üretebilen sistemleri oluşturmayı hedeflemişlerdir. Temelde belli bir alanda uzmanlaşmış sistem yerine insan gibi sorunu algılayabilen, öğrenebilen, insan gibi yorumlayabilen, çözüm üretebilen, bulunduğu ortama ve şartlara adapte olan sistemler oluşturma düşüncesi yatmaktadır. Bu sistemlerin insan gibi modellenmesi için öncelikle insan beynine uygun bir şekilde tasarlanması gerekmektedir. Bunun içinde insan beyninin nasıl çalıştığını ve nasıl davrandığını biyolojik olarak bilmek gerekir. İnsan beyninin biyolojik olarak nasıl çalıştığı ve fonksiyonları üzerine uzun yıllar araştırmalar yapılmıştır. İnsan beyninin fonksiyonları ile ilgili ilk eser 1980 yılında yayınlanmıştır.

3.1 YAPAY SİNİR AĞLARININ TARİHÇESİ

Yapay zekâ ve yapay zekânın alt dalları ile ilgilenen araştırmacılar sürekli daha zeki sistemleri geliştirmeyi amaçlamışlardır. Matematiksel olarak formülize edilemeyen ve çözülmesi mümkün olmayan problemler sezgisel yöntemlerle bilgisayarlar tarafından çözülebilmektedir. Bilgisayarlara problemleri çözme yeteneğini sağlayan ve çözme becerilerini donatan çalışmalar “yapay zekâ” çalışmaları olarak bilinmektedir. İlk defa 1950’li yıllarda ortaya atılan yapay zekâ terimleri zamanla çok fazla ilgi görmüştür. Daha sonrasında hayatın vazgeçilmez parçası olan zeki sistemlerin doğmasına sebep olmuştur. 1940’lı yıllardan beri zeki sistemler üzerinde çalışmaların olduğu bilinmektedir. Fakat bu çalışmalar mühendislik dışındaki çalışmalardır. 1943’te McCulloch ve Pitts [20], 1949’da Hebb [21] gibi bilim adamları araştırmalarını mühendislik alanına kaydırmaya başlamış ve şimdiki yapay sinir ağlarının temellerini oluşturmuşlardır. Bu yıllarda ilk yapay sinir hücresinin yapısını oluşturdular. Yapay sinir hücreleri ile her türlü mantıksal ifadeyi (ve, veya ve değil) fomülize etmenin mümkün olduğunu gösterdiler. Böylece insanlar yapay sinir ağlarının altın çağını yaşadıklarına inanırlar. 1949 yılında Donald Hebb, yapay

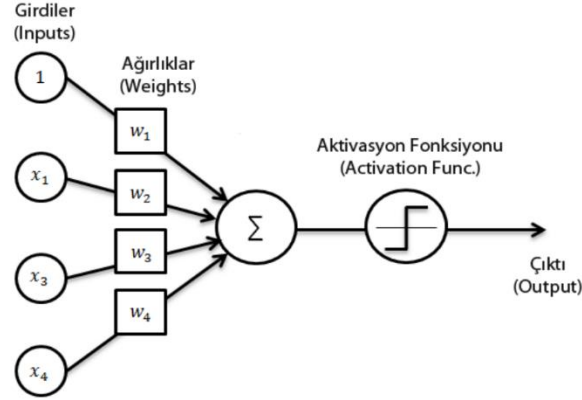
hücrelerden oluşan bir yapay sinir ağının değerlerini değiştiren bir öğrenme kuralı geliştirdi[21]. Bu kural günümüzde de birçok öğrenmek kuralının temelini oluşturan “Hebbian öğrenme” kuralının kendisidir. 1950’li yıllarda çalışmalar daha açık bir şekilde hızlandı ve 1951 yılında ilk nuro-bilgisayar üretildi. 1954 yılında Farley ve Clark [22] tarafından rassal ağlar (random networks) ile adaptif tepki üretme kavramı ortaya atıldı ve bu kavram daha sonralarında birçok araştırmacının ilgisini çekmiştir. Bu kavram 1958 yılında Rosenblatt tarafından geliştirildi. Rosenblatt tarafından geliştirilen algılayıcı model (perceptron Şekil 3.1) yapay sinir ağları tarihinde önemli bir yer elde etmiştir. Çünkü bu model daha sonraları geliştirilecek ve yapay sinir ağlarında devrim niteliğinde olacak olan çok katmanlı algılayıcıların temelini oluşturmaktadır.

Rosenblatt 1958 yılında yaptığı çalışmasında 3 temel soruya yanıt aramıştır. Bunlar:

- i. Fiziksel dünyadaki olaylar biyolojik sistem tarafından nasıl algılanır veya tespit edilir?
- ii. Sağlanan bilgiler hangi formda saklanır veya hatırlanır?
- iii. Depolamada veya bellekte yer alan bilgiler tanıma ve davranışları nasıl etkiler?

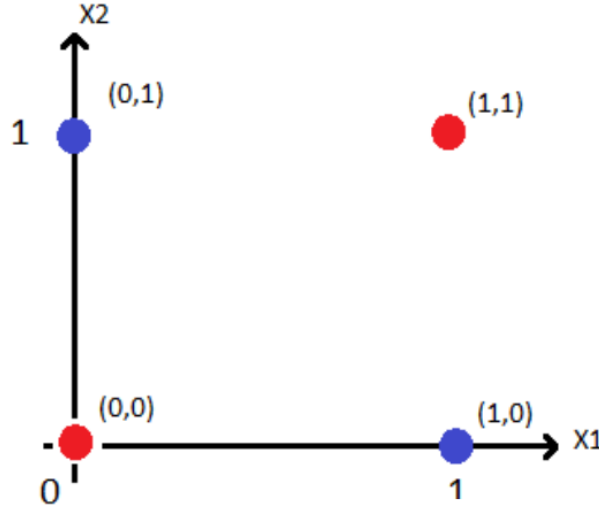
Bu yaklaşım da insanoğlunda var olan sinir ağlarının çalışma prensibinden esinlenilerek geliştirilmiştir. Eşik değeri (threshold) ve aktivasyon fonksiyonu kavramları bu çalışmayı önemli kılan noktalardır [23].

Benzer şekilde 1960 yılında Windrow ve Hoff ADALINE (ADative LInear NEuron) modelini geliştirdiler [24].



Şekil 3.1: Perceptron yapısı.

Geliştirilen bu model yapay sinir ağlarının mühendislik uygulamalarının başlamasında ilk adım olarak sayılmaktadır. ADALINE modeli hem adaptif öğrenmenin hem de 1970 yıllarında MADALINE modellerinin çıkmasında temel olmuştur. 1960'larda yapay sinir Ağa çalışmaları duraklama devrine girmiştir. Yapay sinir ağlarının duraklama devrine girmesine sebep olan olay, Yapay Zekâ bilimcilerinin o yıllarda yazdığı algılayıcılar (Perceptrons) başlıklı kitap olmuştur. Bu kitapta yazarlar, doğrusal olmayan problemlerin yapay sinir ağları ile çözülemeyeceğini iddia ettiler. Bu tezlerini ispatlamak için ise meşhur XOR probleminin çözülememesini örnek gösterdiler (Şekil 3.2). Bu örnek birçok araştırmacının yapay sinir ağları üzerindeki çalışmalarını sekteye uğrattı. Hatta Amerika Birleşik Devletlerinde araştırma ve geliştirme çalışmalarına destek sağlayan DARPA atlı organizasyon yapay sinir ağları ile ilgili çalışmalara destek vermeyi kesti. Durum böyle olunca sadece birkaç araştırmacı bu konuda çalışmaya devam etti. Çalışmalar o kadar azaldı ki XOR problemi çözülünceye kadar dikkatleri yapay sinir ağlarına çekmek mümkün olmadı.

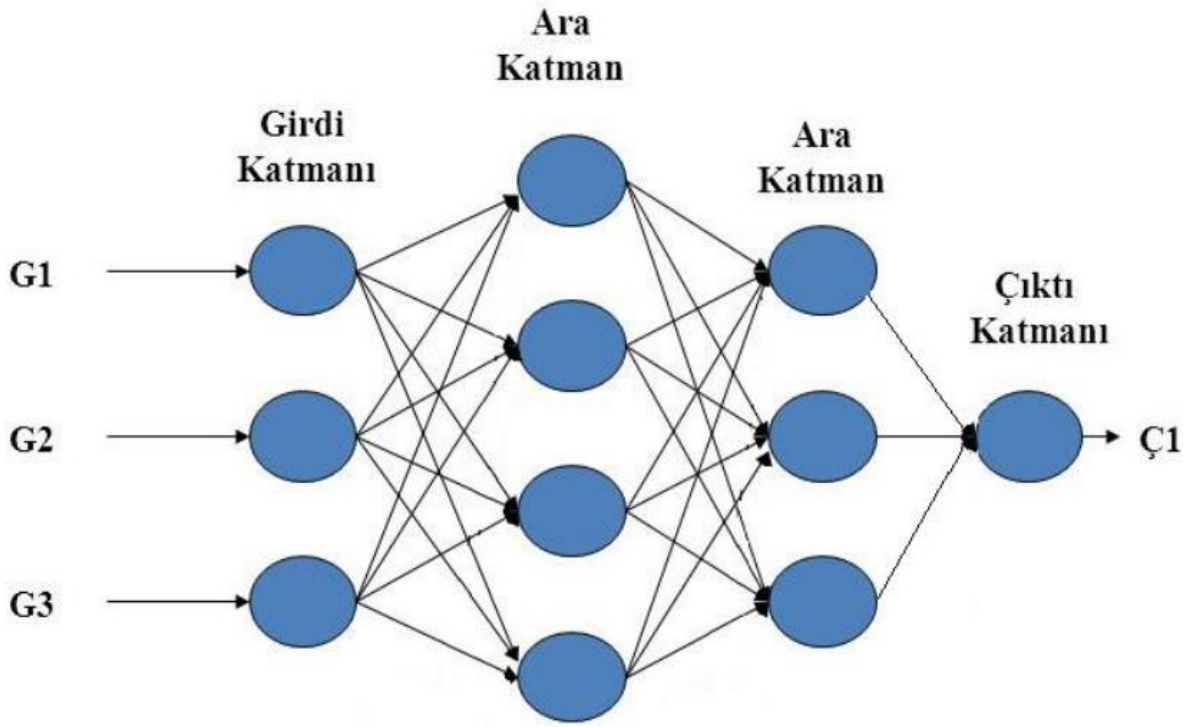


Şekil 3.2: XOR problemi.

Yapay sinir ağlarındaki çalışmaların 1969 yılında sekteye uğraması ve gerekli finansal desteklerin sağlanmamasına rağmen bazı bilim adamları çalışmalarına devam ettiler. Bu çalışmalar 1980 li yıllarda meyvelerini vermeye başladılar. Bu sayede yapay sinir ağları çalışmalarındaki sessizlik sona erdi. Kohonen 1982 yılında “ kendi kendine öğrenme nitelik haritaları (self organizing feature maps-SOM)” alanındaki çalışmasını yayınladı [25]. Hopfield tarafından 1982 ve 1984 yıllarında yayınlanan çalışmalarda yapay sinir ağlarının geliştirilebileceği özellikle geleneksel bilgisayar programlama ile çözülmesi zor olan problemlere çözüm geliştirilebileceğini gösterdi[26],[27]. Bu çözümlere en güzel örnek gezgin satıcı problemi olmuştur. Çalışmalarını kolay ve anlaşılır bir şekilde sunmasından mühendislerin yapay sinir ağlarına ilgisini yeniden artırmaya başladı. Bu çalışmaların ilerlemesi Hinton ve arkadaşlarının geliştirdikleri Boltzman makinesinin doğması ile taçlanıyordu.

1986 yılında Rumelhart ve arkadaşları paralel programlama konularındaki çalışmalarını ilerletiyor ve bugün ki derin öğrenme modellerinin de temellerini oluşturacak bir yaklaşım ortaya atıldı. G. Hinton, D. Rumelhart ve R. Williams tarafından yayınlanan çalışmada geri yayılım olarak isimlendirilen yapay sinir ağları için yeni bir öğrenme metodu geliştirildi [28]. Daha önce Perceptron yapısında kullanılan katman sayısı artırılarak “Çok Katmalı Algılayıcı(Multi Layer Perceptron)” yapısı elde edildi (Şekil 3.3). Bu yapının eğitilmesi, ağırlıkların doğru bir şekilde güncellenmesi için geri yayılım algoritması kullanıldı. Geri yayılım algoritması, ağda bulunan nöronlar arasındaki bağlantıları, ağırlıkları gerçek çıktı vektörü ile hesaplanan çıktı vektörü

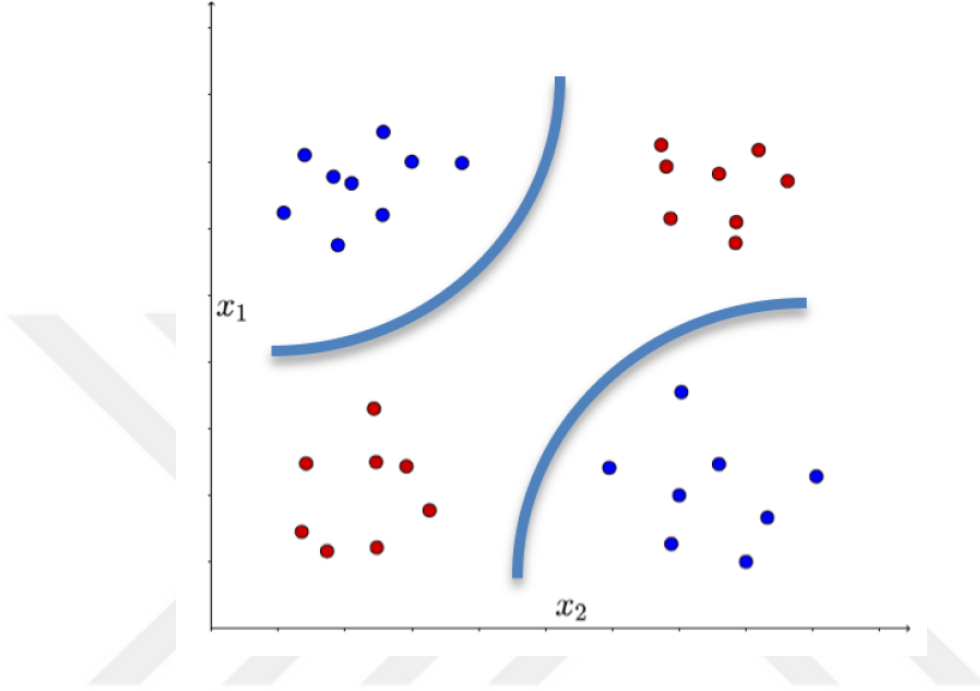
arasındaki farkın minimize edilmesini sağlayacak şekilde günceller [28]. Girdi ve çıktı katmanları arasındaki katmanlar gizli katman(hidden layer) olarak isimlendirilir. Öğrenme prosedürü, verilen girdi için istenen çıktıyı elde edebilmek adına gizli birimlerin hangi durumlarda aktif olacağına karar vermelidir [28]. Hatanın minimuma indirgenmesi için katmanlar arası ağırlıkların doğru bir şekilde güncellenmesi gerekir. Geri yayılım algoritması ağırlıkların güncellenmesini, isteğe bağlı olarak her bir örnekten sonra ya da belirli örnek topluluklarından (batch) sonra hesaplanan hataya göre yineleyerek yapar. Ağırlıkların güncellenmesi hataların türevi alınacak şekilde yapıldığından ve katman sayısı ya da katmanlardaki nöron sayılarının fazla olmasından dolayı yüksek işlem gücü gerektirir. Fakat geri yayılım algoritması ve çok katmanlı algılayıcı yapısı doğrusal olmayan, lineer olarak ayıramayan problemlerin çözümü için başarılı sonuçlar elde etmiştir ve yapay zekâ çalışmalarını tekrar ivmelendirmiştir.



Şekil 3.3: Çok katmanlı algılayıcı şeması.

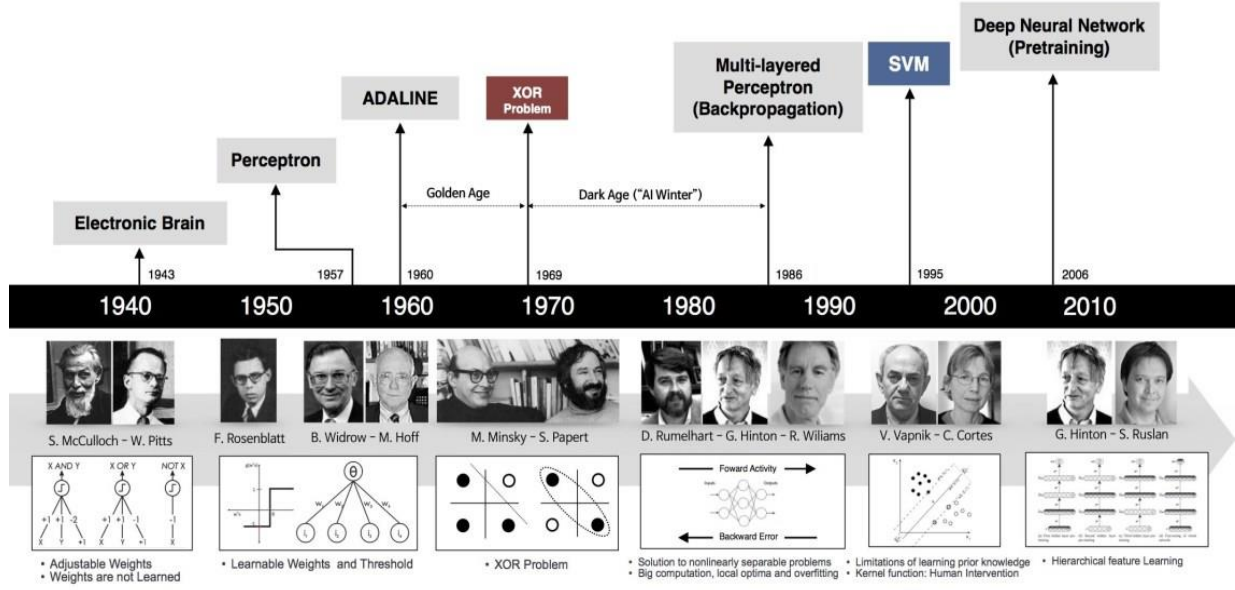
Bu çalışmalardan sonra yapay sinir ağlarına olan ilginin ateşlenmesi sonucu yapılan seminerlerde çok fazla yapay sinir ağları ile ilgili makaleler sunuluyor ve yapay sinir ağları bütün disiplinlerde kendini göstermeye başlıyordu. Çünkü XOR problemini çözemeyen tek katmanlı algılayıcı yerine

çok katmanlı algılayıcılar bulunmuş ve yapay sinir ağlarının çalışmadığını söyleyen bütün tezler çürümüştür.



Şekil 3.4: XOR probleminin çözümü.

1985 yılında Parker çok katmanlı algılayıcı ile ilgili bazı çalışmalar yürütüyordu. Çok katmanlı algılayıcı sadece XOR problemini çözmekle (Şekil 3.4) kalmamış diğer taraftan Hopfield ve Boltzman makinelerinin sınırlamalarını da çözmüştür. Sinir ağlarının gelişim sürecinin yıllara göre değişimi Şekil 3.5'te gösterilmiştir.



Şekil 3.5: Sinir ağlarının kısa tarihçesi.

3.2 YAPAY ZEKÂ VE YAPAY SİNİR AĞLARININ UYGULAMA ALANLARI

3.2.1 Görüntü Sınıflandırma ve Yerini Belirleme

Konvolüsyonel sinir ağları verilen görüntülerin sınıflandırılması için etkin bir şekilde kullanılır. 2015 yılında ILSVCR yarışmasında ResNet modeli ile insan hata oranının altında bir sonuç elde etmiştir. Verilen görüntü için sınıflandırma yaparken sınıflandırılan nesnenin görüntü içerisindeki konumu(Classification + Localization) da konvolüsyonel sinir ağları ile belirlenebilir.

3.2.2 Nesne Algılama (Object Detection)

Konvolüsyonel sinir ağları nesnelerin algılanması için kullanılabilir. Görüntü sınıflandırmadan farklı olarak, bu yöntemde verilen görüntü içerisinde bulunan farklı nesneler tespit edilebilir ve tespit edilen her farklı nesne farklı renkte bir dörtgen ile belirtilen alan içerisinde çevrelenerek gösterilir. Özellikle Çin gibi devletler, sokak kameralarından gelen dinamik görüntülerde nesne algılama yöntemini kullanmaktadır. Görüntü içerisinde bulunan farklı nesneler tespit edilip, belirli dörtgensel alanlarda etiketleri ile beraber işaretlenir. Bu alanda geliştirilen, gerçek zamanlı nesne algılama(Real Time Object Detection) projesi olan YOLO(You only look once) günümüzde çok popülerdir.

3.2.3 Semantik Bölümleme (Semantic Segmentation)

Konvolüsyonel sinir ağlarının başarılı olduğu alanlardan bir diğeri ise semantik bölümleme alanıdır. Semantik bölümleme, görüntüyü piksel seviyesinde anlayabilmektir. Yani görüntüde bulunan her bir pikseli bir nesne sınıfına, kategoriye atamak istiyoruz[23]. Görüntüleri tanımanın yanında, görüntüde bulunan her bir nesnenin sınırlarını da çizmek, belirlemek istenir[23]. Semantik bölümleme, görüntü sınıflandırmadan farklı olarak görüntüler için piksel yoğunluğuna dayalı tahminler oluşturuyoruz. Bu alanda kullanılan en önemli veri setleri şunlardır: VOC2012 ve MSCOCO[23].

3.2.4 Sahne Bölümleme (Scene Segmentation)

Konvolüsyonel sinir ağları sahne bölümleme işlemleri için sıklıkla kullanılmaktadır. Sürücüsüz araçların çevreyi daha iyi algılayabilmeleri ve nesneleri tespit edebilmeleri için bu yöntem kullanılabilmektedir.

3.2.5 Resim Altyazısı Oluşturma (Image Captioning)

Konvolüsyonel sinir ağları ve tekrarlayan sinir ağı(RNN) kullanılarak resim altyazısı oluşturma işlemi gerçekleştirilmektedir. Bu teknik diğerlerinden farklı olarak iki farklı derin öğrenme mimarisini beraber kullanır. İlk olarak konvolüsyonel sinir ağları kullanılarak girdi üzerinde nesne tespiti gerçekleştirilir. Daha sonra tekrarlayan sinir ağları kullanılarak, tespit edilen nesnelerden anlamlı bir cümle oluşturulur.

3.2.6 Sağlık ve Sağlık Hizmetlerinde

Sağlık hizmetleri verilerinin artan kullanılabilirliği ile AI, sağlık hizmetlerine bir paradigma kayması getirdi. Sağlık sektöründe Yapay Zekanın birincil odak noktası, hasta sonuçları ile kullanılan tedavi veya önleme tekniği arasındaki ilişkilerin analizidir. AI programları, hasta teşhisi, tedavi protokolü oluşturma, ilaç geliştirme ile hasta izleme ve bakımı için başarıyla geliştirilmiştir. Güçlü AI teknikleri, büyük miktarda klinik veriyi eleayabilir ve karar vermeye yardımcı olmak için klinik olarak ilgili bilgilerin kilidini açmaya yardımcı olabilir.

- i. Radyoloji – Yapay zekanın görüntüleme sonuçlarını yorumlama yeteneği, uzmanların bir görüntüdeki insan gözünün kolayca gözden kaçırabileceği değişiklikleri algılayma

yeteneğini destekler. Stanford Üniversitesi'nde yakın zamanda oluşturulan bir YZ algoritması, pnömoni hastalarının akciğerlerindeki belirli bölgeleri tespit edebilmektedir.

- ii. Elektronik Sağlık Kayıtları – Bilgi yayılımını ve erişimi optimize etmek için dijital sağlık kayıtlarına duyulan ihtiyaç, sistemlerde sağlıkla ilgili tüm verilerin hızlı ve doğru bir şekilde günlüğe kaydedilmesini gerektirir. Bir insan hatalara eğilimlidir ve bilişsel aşırı yüklenme ve tükenmişlikten etkilenebilir. Bu süreç, YZ tarafından başarıyla otomatikleştirildi. Elektronik sağlık kayıtları verilerinde Öngörücü modellerin kullanılması, başlangıçta %70-72 doğrulukla bireyselleştirilmiş tedavi yanıtının tahmin edilmesini sağladı.
- iii. Görüntüleme – Devam eden yapay zekâ araştırmaları, yüz çekiciliğini tahmin etmek için yarık damak tedavisini değerlendirmenin yanı sıra düzeltici çene cerrahisinin sonucunu değerlendirmede doktorlara yardımcı oluyor.
- iv. Yapay zekâ destekli robotik cerrahi – Robot destekli cerrahinin en büyük avantajı, büyük kesikler gerektirmemeleri ve ameliyat sonrası iyileşme süresinin düşük olmasıyla minimal invaziv olarak kabul edilmeleridir. Robotlar, ameliyat öncesi hasta tıbbi kayıtlarından gelen verileri analiz edebilir ve ardından ameliyat sırasında cerrahin aletlerine rehberlik edebilir. Bu robot destekli ameliyatlarda, hastaların hastanede kalış sürelerinde %21'e varan azalma bildirmiş tir. Robotlar ayrıca geçmiş ameliyatlardan gelen verileri kullanabilir ve cerrahı olası yeni teknikler hakkında bilgilendirmek için yapay zekâyı kullanabilir. En gelişmiş cerrahi robot olan “Da Vinci”, cerrahların geleneksel yöntemlerden daha yüksek doğruluk ve daha fazla kontrol ile karmaşık cerrahi prosedürleri gerçekleştirmesini sağlar.

3.2.7 Gazetecilik Endüstrisinde

Biraz şaşırtıcı değil mi, raporların yerini bilgisayarlar alıyor? Gazetecilik kadar öznel ve duygusal olarak yönlendirilen bir şeyin makineler tarafından nasıl yapılabileceğini anlamak kolay değil. Ancak “Bloomberg News” tarafından yayınlanan içeriğin yaklaşık üçte biri şu anda bir tür Yapay Zekâ kullanıyor. Bloomberg, gazetecilere şirketin kazanç raporuyla ilgili her sorguda binlerce haber makalesi yazmasına yardımcı olmak için tasarlanmış “Cyborg” adlı bir sistem kullanıyor. Program, finansal verileri elde eder etmez raporu inceler ve en uygun gerçekler ve rakamlarla bir haber hikâyesi oluşturur. Bu tür haberciliği sıkıcı ve yorucu olarak değerlendiren çoğu iş

muhabirinin dışında, Cyborg doğru bir şekilde ve yorulmadan çalışıyor. Bu sayede Bloomberg, yüksek hızlı ticari finans gazeteciliği alanında Reuters ile rekabet ediyor ve Cyborg'un yardımıyla Bloomberg'in büyük kazançlar yapmayı planlanıyor.

3.2.8 Diğer Alanlar

Evrimsel sinir ağları günümüzde birçok farklı alanda başarılı bir şekilde kullanılmaktadır. Örneğin bankacılık sektörü, milyonlarca müşterisine kaliteli müşteri hizmeti sağlama, dolandırıcılığı önleme, mobil teknoloji ve diğerlerinin yanı sıra veri güvenliği gibi birçok geleneksel bankacılık sorununu çözmek için parasını yapay sinir ağlarını kullanan YZ tabanlı çözümlere yatırım yapmaktadır.

Seyahat endüstrisi, bilgisayar tabanlı havayolu rezervasyon sistemlerinin ortaya çıkmasıyla büyük ölçekli endüstriyel hesaplama öncülük etti. Çoğu seyahat şirketi, kullanıcılardan veri toplar ve doğru Yapay Zekâ araçlarıyla, müşterilere geçmiş aramalarına dayalı olarak geleceğe yönelik seyahat programları sunabilir. Pazar talebine, hava durumuna ve diğer belirleyici faktörlere bağlı olarak uçuş fiyatlarını otonom olarak ayarlayabilen Yapay Zekâ fiyatlandırma araçlarının kullanımı, havayollarının işlerini optimize etmesine yardımcı olmaktadır.

Ulaştırma endüstrisi, insan hatalarından kaynaklanan iki soruna, trafik ve kazalara karşı oldukça hassastır. Bu problemlerin doğası gereği öngörülemeyen yapıları nedeniyle modellemek çok zordur, ancak gözlemlenen verileri analiz edebilen ve uygun kararları verebilen veya tahmin edebilen Yapay zekâ destekli araçların kullanımıyla kolayca üstesinden gelinebilir. Artan seyahat talebi, güvenlik endişeleri, CO2 emisyonları ve çevresel bozulma gibi zorluklar yapay zekânın gücüyle karşılanabilir. Yapay Sinir Ağlarından Arı kolonisi optimizasyonuna kadar, ulaşım endüstrisini verimli ve etkili hale getirmek için birçok yapay zekâ tekniği kullanılmaktadır. Nüfus için seyahat süresini daha güvenilir hale getirirken trafik sıkışıklığından önemli ölçüde kurtulmak için ulaşım yetkilileri çeşitli yapay zekâ tabanlı çözümler deniyor. Gelişmiş yol altyapısı ve sürücüler için yardım için yapay zekânın potansiyel uygulamasıyla, ulaşım endüstrisi, maliyet etkin olmakla birlikte çevre üzerinde hiç bir etkisi olmayacak daha güvenilir bir ulaşım sistemi gerçekleştirmeye odaklanmıştır.

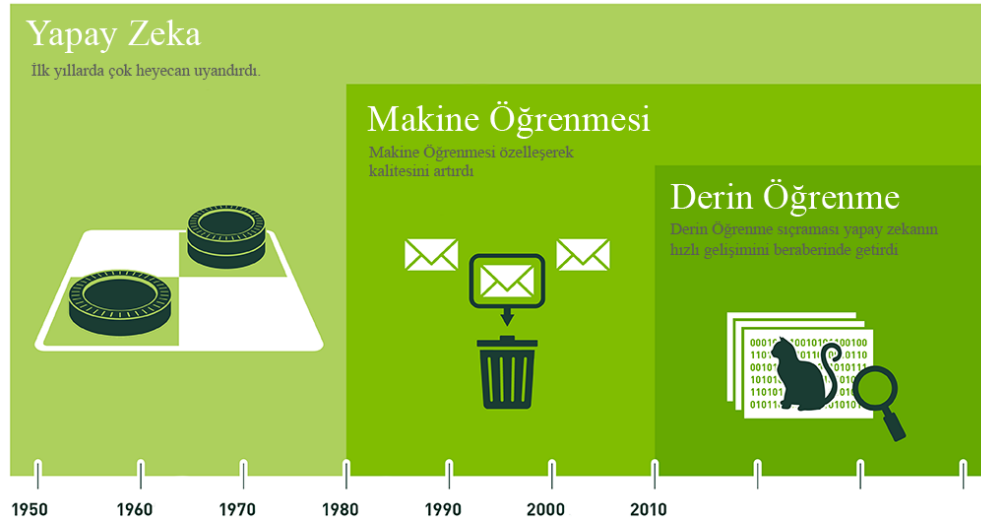
Eğitimde yapay zekâ destekli araçların artan kullanımıyla birlikte bir öğretmenin işinde ve eğitimdeki en iyi uygulamalarda birden fazla değişiklik beklenebilir, ancak öğretmenlerin kritik

varlığı yeri doldurulamaz kalacaktır. Yapay zekânın eğitim için kullanımının artmasıyla birlikte akademik dünya daha da kullanışlı ve kişisel hale gelmektedir. İnsanların öğrenme şekli hızla değişiyor; eğitim materyalleri bilgisayarlar ve akıllı cihazlar tarafından kolayca erişilebilir durumdadır. Bir eğitimi almak için geleneksel sınıfta olma yönteminin yerine, öğrencilerin evde bilgisayarlarında bir internet bağlantı rahatlığıyla sağlanmaktadır. Yapay zekâ, yapay zekâ kullanılarak kolayca otomatikleştirilebilen idari görevlere harcanan süreyi azaltarak göstergelere ve akademik kurumlara da kolaylık sağlamaktadır.

Yapay zekâ, video oyunlarından filmlere kadar eğlence kaynaklarının arka planında giderek daha fazla çalışmakta ve bize daha zengin, ilgi çekici ve daha gerçekçi bir deneyim sunmaktadır. Netflix ve Disney+ gibi eğlence sağlayıcıları, kullanıcılara bireysel kullanıcıların geçmiş etkinliklerinden ve davranışlarından türetilen kişiselleştirilmiş öneriler sağlamak için yapay zekâ kullanıyor. Bilgisayar grafikleri ve dijital medya içerik üreticileri, üretim süreçlerinin hızını ve verimliliğini artırmak için Yapay Zekâ tabanlı araçlardan yararlanıyor. Film şirketleri, film fragmanları ve reklamlarının yanı sıra yapım öncesi ve sonrası süreçlerinde makine öğrenimi algoritmalarını giderek daha fazla kullanmaktadır.

4. DERİN ÖĞRENME

Makine öğrenmesi, bir sistemin belirli bir problem üzerinde karar vermesini sağlayan metotların geliştirildiği bilgisayar biliminin bir alanıdır. Temel olarak makine öğrenme, ham veriden bilgi çıkarmak ve bazı modellerde temsil etmek için algoritmalar kullanmaktadır. Yapay Sinir Ağları (YSA) ise, problemleri bilgisayar ortamında yapay olarak çözmek için insan beyninden etkilenecek geliştirilmiş bir tekniktir. YSA yönünden “Derin Öğrenme” (Deep Learning) ifadesi ilk kez Igor Aizenberg ve arkadaşları tarafından 2000 yılın kullanılmıştır[29]. Derin öğrenmenin tanımı birçok kaynakta geçmekte olup hepsinde farklı farklı tanımlar ve yüksek seviyeli açıklamalar yapılmıştır. Ancak Derin öğrenme, hiyerarşik mimarilerdeki birçok katmandan oluşan, büyük miktarlarda etiketlenmiş eğitim verilerinden özellik saptama yapabilen sistemler oluşturmak için çok seviyeli “derin” sinir ağların kullanılmasıdır. Derin öğrenme, makine öğrenmesi yaklaşımı olup son yıllarda bilgisayar bilimcilerin çalışmalarını yoğunlaştırdığı popüler bir çalışma alanı olmuştur. Yapay Zekâ, Makine Öğrenmesi ve Derin Öğrenmenin gelişim süreçleri ve aralarındaki ilişki Şekil 4.1’deki gibidir.



Şekil 4.1: Yapay zekânın süreçleri [30].

4.1 EVRİŞİMSEL SİNİR AĞLARI

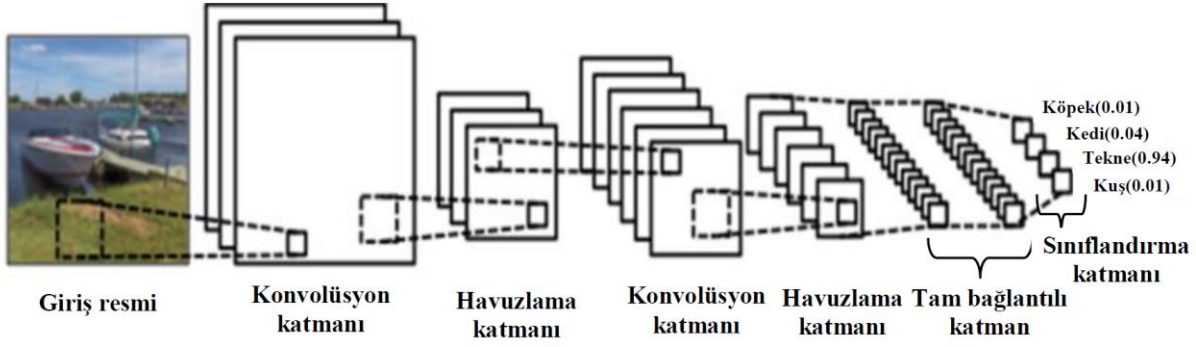
Bilgisayarlar evrişimli sinir ağları ile görüyorlar desek yanlış olmaz. Çünkü yaşadığımız zamanda teknoloji devlerinin tümü bu ağları kullanıyor. İlk uygulamaları Yann LeCun tarafından 1990 yıllarında yapılmasına rağmen 2012 yılından sonra yaygınlaşmaya başlamıştır. Öncelikle evrişim

kelimesinin anlamına bakalım. Evrişim matematiksel olarak iki sinyalin birleşip, üçüncü bir sinyal elde edilmesidir. Sinir ağlarında ise evrişim işlemi, sık sık bir filtreleme operasyonu olarak tanımlanır. Evrişim kernellerine filtre adı verilir. Matematiksel olarak kernel ise, $AX=0$ veya $BX=0$ gibi homojen denklemin çözümüne karşılık gelen çözüm kümelerine denir. Evrişimli sinir ağlarında filtreler ağırlıklardan oluşan bir vektör ile gösterilir. Filtreler sisteme tanıtılan bir girdi parçasının bir özelliğe ne kadar benzediğini ölçmemize imkân sağlar.

Derin öğrenme yöntemlerinden biri olan ESA mimarileri; görüntü sınıflama, nesne tanıma ve tespit yöntemi olarak literatürde sıklıkla kullanılmaktadır. ESA, yapay sinir ağlarından esinlenerek oluşturulmuş ve toplanmış bilgileri uçtan uca öğrenebilen bir mimaridir. ESA, yeterli kapasitesi ve akıllı model yapısı sayesinde büyük ölçekli verilerin de üstesinden gelebilmektedir. ESA'nın dezavantajları ise eğitim sürecinin uzun olması ve eğitim sürecinde yerel bir çözüme takılma ihtimalidir[15].

Evrişimsel Sinir Ağları (ESA), bir dizi makine öğrenme algoritması içeren, ileriye dönük işlem yapan, düşük seviyeli verilerden üst düzey bilgi oluşturan bir bilgi hiyerarşisi öğrenme yeteneğine sahip derin öğrenme yaklaşımıdır[31]. ESA' temel olarak iki katmandan oluşur. Bu katmalardan ilki evrişim (Konvolüsyon) katmanı, ikincisi ise havuzlama katmanıdır[31]. ESA'lar bu iki katmanda temel bazı operatörler ile (hem doğrusal hem de doğrusal olmayan) görüntünün önemli olan özelliklerini çıkarmayı hedeflemektedir[31]. Böylelikle yapay sinir ağına, görüntünün istenmeyen kısımlarının gitmesi engellenir. Bu durum sistemin hem hızlanmasını hem de doğruluk oranını artırmaktadır[15]. ESA'nın geliştirilmesi ile birlikte performansı arttırmak için bırakma katmanları ve aktivasyon fonksiyonları da eklenmiştir.

ESA'nın son katmanında ise görev türüne göre kayıp katmanlar da yer alabilmektedir. ESA'nın genel Şekil 4.2'de gösterilmiştir.



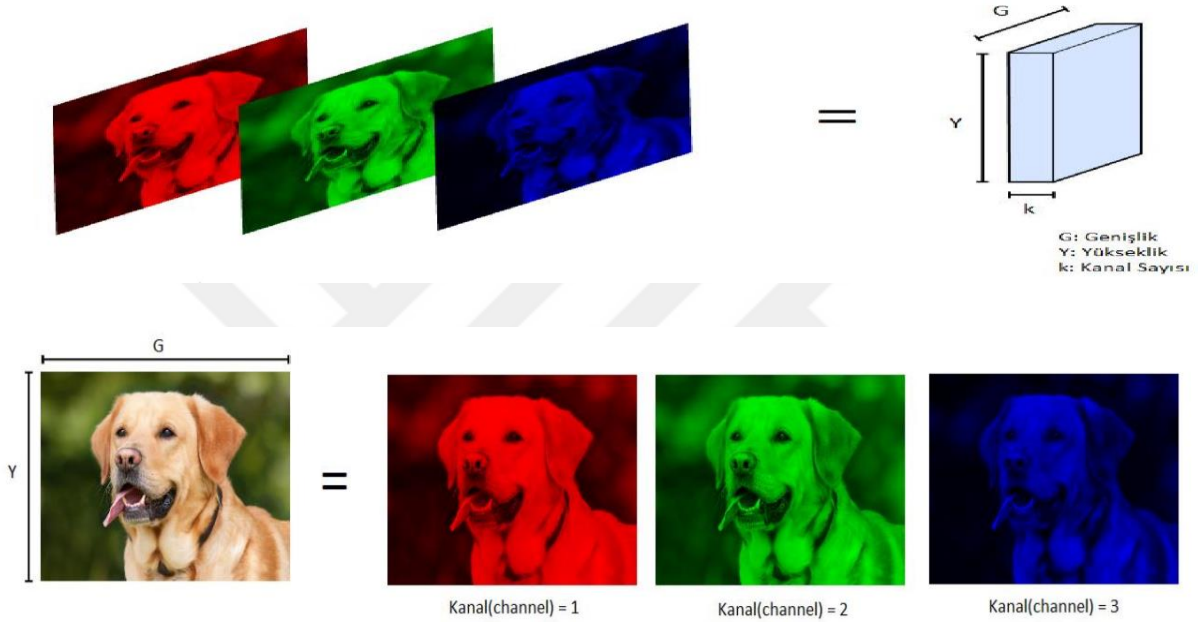
Şekil 4.2: Evrişimsel sinir ağının genel mimarisi [32].

4.1.1 ESA'yı Oluşturan Katmanlar

4.1.1.1 Giriş katmanı

Bu katman isminden de anlaşılacağı üzere ESA'nın ilk katmanıdır. Bu katmanda veri (sayısallaştırılmış ve düzeltilmiş) ham olarak ağı verilmektedir. Tasarlanacak modelin başarımı için bu katmandaki verinin boyutu önem kazanmaktadır. Genelde verilen girdiler resim formatında olduğundan sisteme tanıtılmadan önce bu resimler üzerinde çeşitli işlemler uygulanır. Veri olarak girdinin resim olduğunu düşünürsek bu resmin her pikselini 0 ile 255 arasına ifade edilen bir sayısal değere dönüştürüp, bu değerleri de bir matrisin elemanı olarak saklanır. Genellikle seçilen resimlerin en-boy değerleri eşittir ancak bu bir zorunluk değildir. Resmin en ve boy uzunluğuna göre matrisin boyutları değişebilmektedir. Örneğin: 128x128 piksel boyutunda gri tonlamalı bir resim girdi olarak kullanılacak ise bunun için 128x128x1 boyutunda bir matrise ihtiyaç vardır. Bu resim vektör haline getirilecek ise $128 \times 128 \times 1 = 1536$ elemanlı bir vektör ile temsil edilebilir. Ancak aynı boyutlardaki resminin renkli olduğunu düşünürsek 128x128x3 boyutunda bir matris kullanmamız gerekecektir. Yine bu matrisi vektör formatına getirmek için $128 \times 128 \times 3 = 4608$ elemanlı bir vektöre ihtiyaç duyulacaktır. Renkli olan girdiler direkt olarak 3 boyutta temsil edilmektedir. Her bir boyut, kanal(channel) olarak isimlendirilir. Bunun sebebi her bir girdinin RGB (Red, Green, Blue) kanallarına sahip olmasından kaynaklanmaktadır (şekil 4.3). Yani renkli bir girdinin her bir pikseli “Red” kanalında ayrı, “Blue” kanalında ayrı, “Green” kanalında ayrı bir sayısal değer ile ifade edilmektedir. Bu yüzden seçilen girdi boyutları sinir ağının hızı ve çıktıların doğruluğu için çok önemlidir. Hatta resmin boyutu çok büyük ise işleme sokulmadan önce resim tekrar boyutlanır. Giriş görüntü boyutunun yüksek seçilmesi hem yüksek bellek ihtiyacını hem

eğitim süresini hem de görüntü başına düşen test süresini uzatabilir. Bunun yanında ağ başarısını arttırabilir. Giriş görüntü boyutunun düşük seçilmesi bellek ihtiyacını azaltır ve eğitim süresini kısaltır. Fakat kurulacak ağın derinliği azalır ve performansı düşük olabilir. Görüntü analizinde hem ağ derinliği hem donanımsal hesaplama maliyeti hem de ağ başarısı için uygun bir giriş görüntü boyutu seçilmelidir[32].



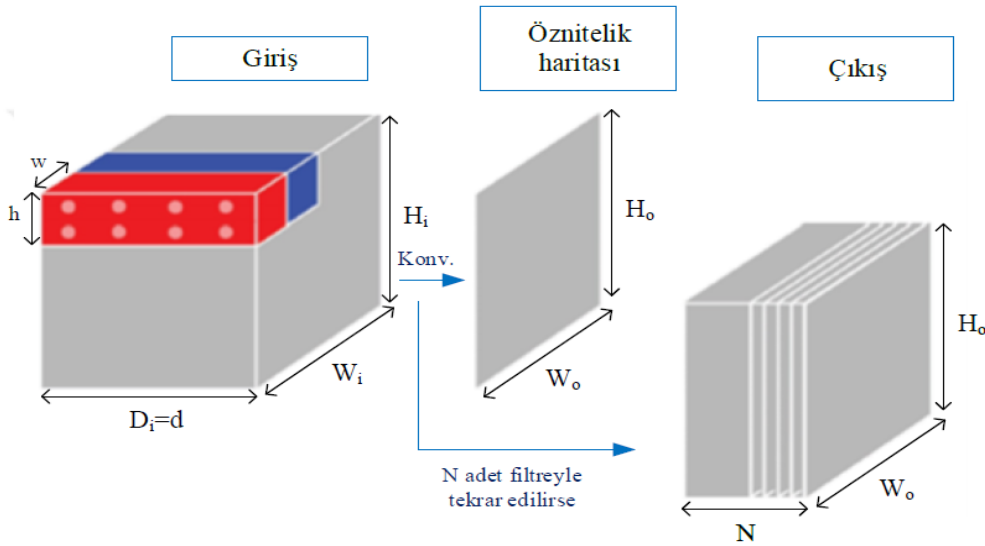
Şekil 4.3: Renkli girdiler için kanal, genişlik ve yükseklik kavramlarının gösterimi.

Her kanal, düzenlenebilecek orijinal görüntü hakkında bilgi tutar. KIRMIZI kanal, renkçilerin görüntünün tonu ve gradyan değerleri ile çalışması için mükemmel olan kontrast haritasını sağlar. YEŞİL kanal, kaynak görüntünün en fazla detayını içerir, böylece bir rötuşçu görüntünün belirli kısım (örneğin cilt lekeleri, güzellik izleri, saç görünümü, gözenekler, vb.) ile çalışmak isterse üzerinde çalışılacak kanaldır. Son olarak, MAVİ kanalı görüntüdeki gürültüyü ayarlamaktan sorumludur, böylece rötuşlar daha pürüzsüz bir sonuç elde etmek veya görüntünün ne kadar pürüzsüz görünmesini istediklerini ayarlamak için bu kanalla çalışabilir.

4.1.1.2 Evrişim (Konvolüsyon) katmanı

ESA'nın temelini oluşturan bu katman dönüşüm katmanı olarak da bilinir. Bu dönüşüm işlemi belirli bir filtrenin tüm görüntü üzerinde dolaştırılması işlemine dayanmaktadır. Bu sebeple

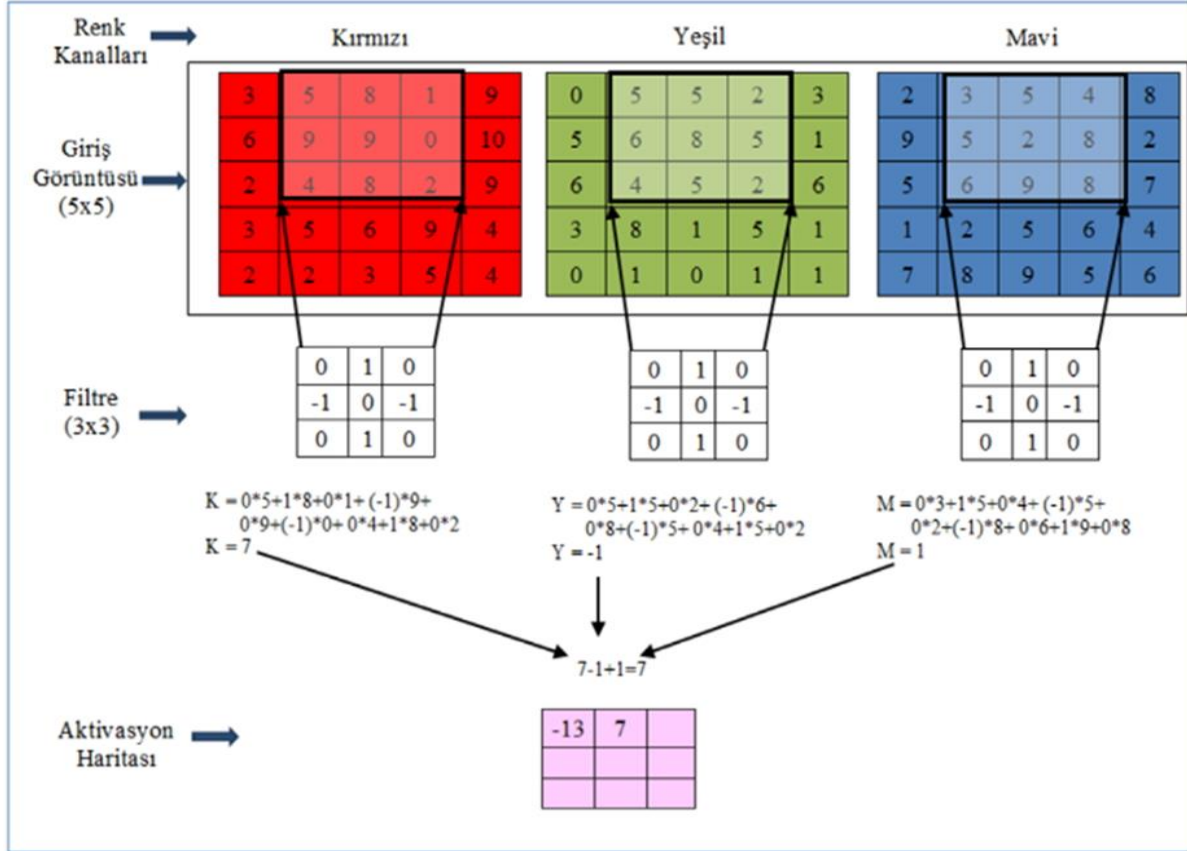
filtreler katmanlı mimarinin ayrılmaz bir bileşenidir[32]. Filtreler genellikle 2x2, 3x3, 7x7 ve 11x11 gibi boyutlarda seçilir. Filtreler, bir önceki katmandan gelen görüntülere konvolüsyon işlemini uygulayarak çıkış verisini oluştururlar. Bu konvolüsyon işlemi sonucu aktivasyon haritası (Özellik haritası) oluşur. Aktivasyon haritası, her bir filtreye özgü özelliklerin keşfedildiği bölgelerdir. ESA'ların eğitimi esnasında bu filtrelerin katsayıları, eğitim kümesindeki her öğrenme yinemesiyle değişir. Böylelikle ağ, özelliklerin belirlenmesi için, verinin hangi bölgelerinin önem taşıdığını belirler. Bir filtrenin görüntü üzerine uygulanması Şekil 4.4'te gösterilmiştir. Şekil 4.5'te görüldüğü gibi giriş görüntüsü renkli (RGB) ve 5x5 bir matris olduğu düşünülürse giriş veri boyutu 5x5x3 olacaktır. Konvolüsyon işlemi için 3x3'lük bir filtre ile giriş görüntüsü üzerinle sağa veya sola doğru belirli Adım (Stride) kaydırılarak dolaşma yapılır. Bu dolaşma esnasında matris sınırına gelindiğinde ise bir basamak aşağı kayıp tekrar devam edilir. Bu dolaşma işlemi görüntü matrisinin tümü üzerinde yapılır. Filtre katsayıları her bir renk kanalındaki değerlerle çarpılıp bunların toplamı alınır. Her üç kanal üzerinde bu işlem yapıldıktan sonra üçünün toplamı Aktivasyon haritasını oluşturur. Her bir renk kanalı matrisine uygulanan filtre katsayıları farklı olabilir. Bu filtre katsayılarındaki değişimler tasarımcılar tarafından modellerine uygun olacak şekilde yapılır.



Şekil 4.4: Tek bir konvolüsyon katmanının gösterimi [33].

Aktivasyon haritasındaki değerler, giriş ve çıkış boyutu arasında aynı yoğunluk aralığını sağlamak için normalize edilir. Bu normalize için her bir renk kanalı için hesaplanan değerler filtre

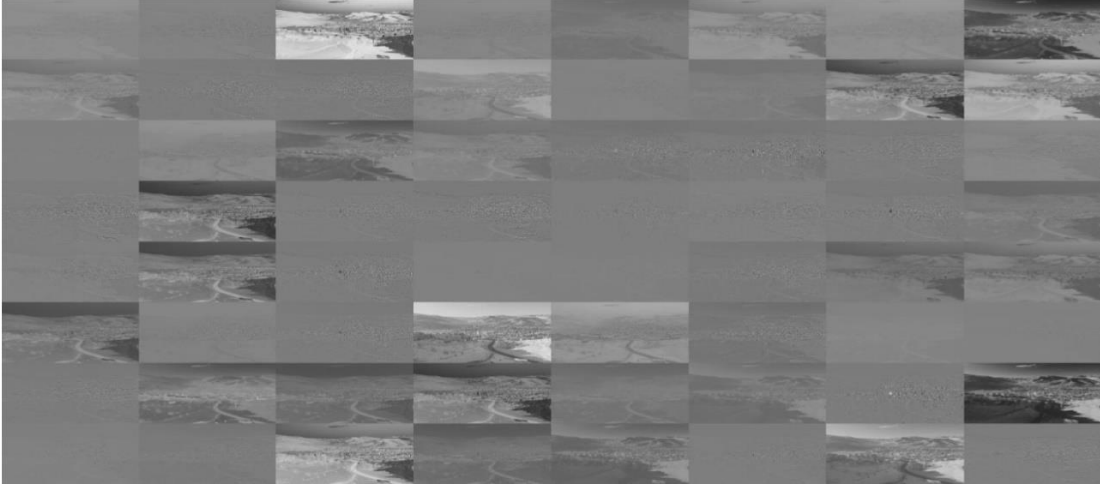
katsayılarının toplamına bölünür. Şekil 4.4'te gösterilen örnekte filtre katsayılarının toplamı sıfır olduğu için bu işlem yapılmamıştır. Konvolüsyon katmanının ESA üzerinde gösterimi Şekil 4.4 ve Şekil 4.5'te verilmiştir. Ağın birinci konvolüsyon katmanı için 3x3'lük boyutta ve toplam 64 filtreden oluşan bir ESA tasarlanmıştır. Her bir filtrenin Şekil 4.6'daki giriş görüntüsüne uygulanması sonucu Şekil 4.7'deki görüntüler elde edilmiştir. Şekil 4.7'e bakıldığında her bir filtrenin giriş görüntüsüne etkisi farklı olduğu görülmektedir[32].



Şekil 4.5: 5x5x3 boyutundaki bir giriş verisine 3x3'lük filtrenin uygulandığı konvolüsyon işlemi [15].



Şekil 4.6: ESA için giriş görüntüsü [32].

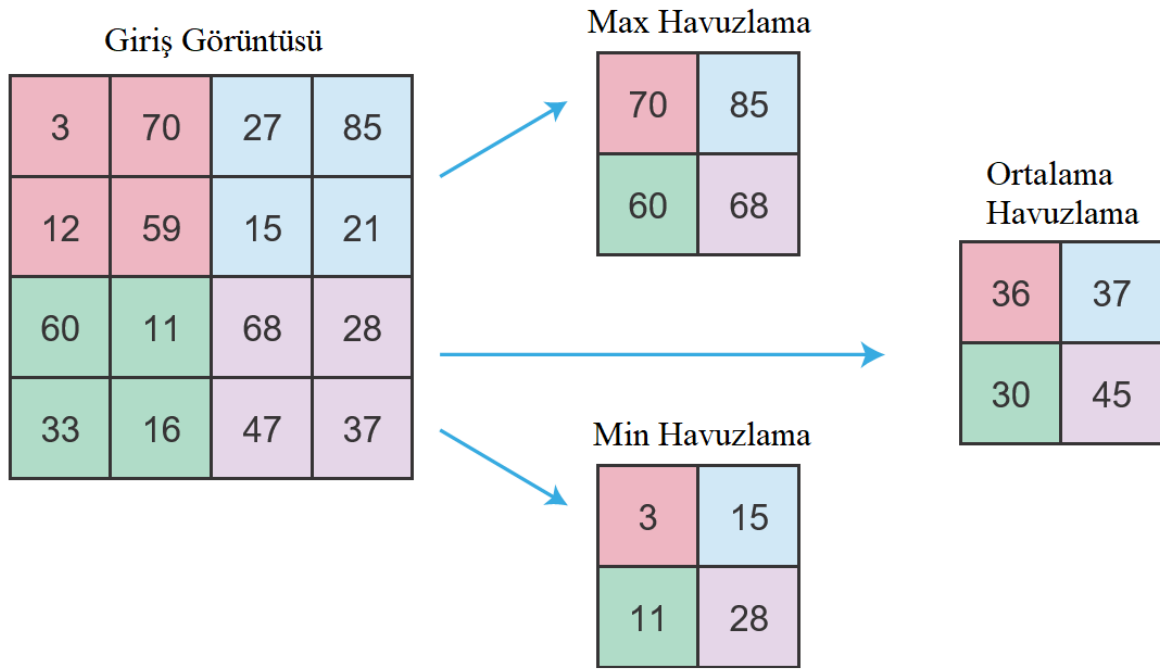


Şekil 4.7: ESA'nın konvolüsyon katmanından sonraki oluşan görüntüsü [32].

4.1.1.3 Havuzlama (Boyut indirgeme) katmanı

ESA'nın havuzlama katmanında kullanılan filtreler ya ortalama değer (meanpooling) ya da maksimum değer (maxpooling) hesaplama prensibine göre çalışır. Ancak hangi tekniği uygularsak uygulayalım, görüntünün bir bölgesinde elde edilen özellikler diğer bölgeler içinde geçerli olmalıdır. Bu özellik “durağanlık” özelliği olarak da bilinir. Havuzlama işlemi genellikle 2x2, 3x3, ..., nxn boyutunda filtreler uygulanarak yapılır. Temel amacı, sonraki konvolüsyon katmanı için

giriş boyutunu (Genişlik x Yükseklik) azaltmaktır. Ortalama havuzlama, filtre boyutunun alanına giren piksel değerlerin toplamının filtre pencere boyutuna bölünmesi prensibiyle çalışır. Maksimum havuzlama[34] ise filtre boyutunun alanına giren piksel değerlerinden büyük olan değerin belirlenmesi işlemidir. İki temel havuzlama katmanı farklı ESA mimarilerinde kullanılarak mimarilerin başarımları incelenmiştir. Maksimum havuzlama yönteminin daha etkili olduğu görülmüştür. Havuzlama katmanlarına ait işlemlerin uygulanması Şekil 4.8’de gösterilmiştir.



Şekil 4.8: Maksimum, minimum ve ortalama havuzlama.

Evrişim işleminden sonraki resme havuzlama işlemi uygulanarak özellik vektöründe boyut indirgenmiş olur. Filtre boyutu F ve adım sayısı S olsun ve veri girdi boyutu $X \times Y \times Z$ olarak kabul edilsin. $\hat{X} \times \hat{Y} \times \hat{Z}$ boyutlu bir çıkış özellik haritası üretmek için denklem (4.1) kullanılmaktadır.

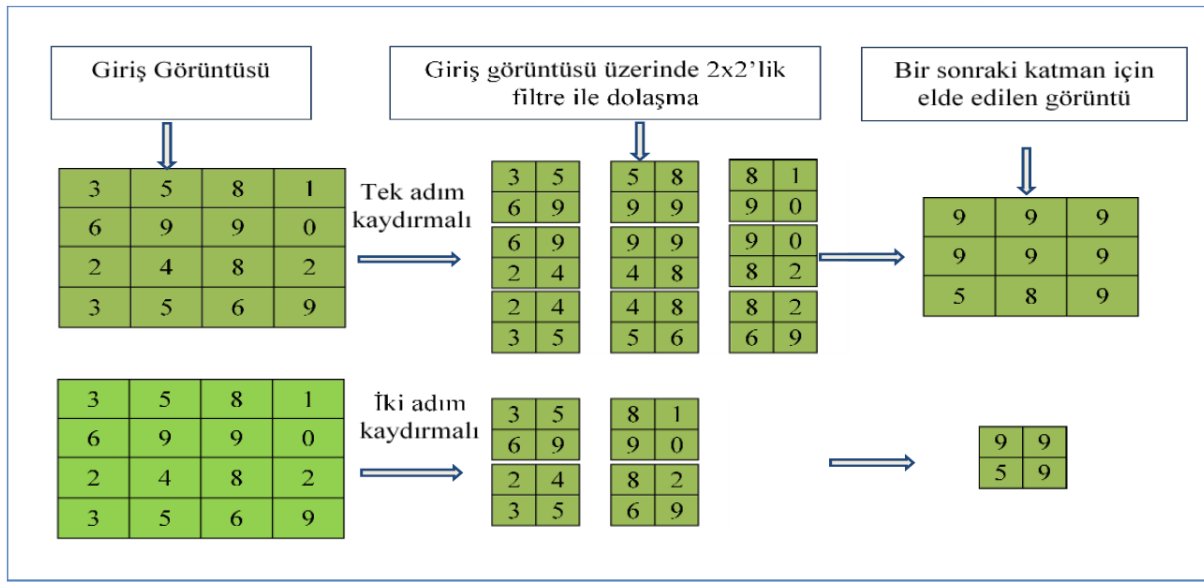
$$\hat{X} = \frac{X - F + S}{S}, \hat{Y} = \frac{Y - F + S}{S}, \hat{Z} = Z \quad (4.1)$$

X = Giriş görüntü boyutunun genişliği

F= Filtre boyutu

S= Adım kaydırma sayısı

Havuzlama işleminde genellikle F=2 ve S=2 olarak seçilir. ESA'larda Havuzlama katmanı isteğe bağlıdır ve bazı mimariler bu işlemi gerçekleştirmez. Havuzlama işleminin yapılışı ile ilgili örnek Şekil 4.9'te verilmiştir. Şekil 4.9'da giriş görüntü boyutu 4x4 ve filtre boyutu 2x2. Bir adım kaydırılacak şekilde oluşan görüntünün boyutu 3x3 olur. İki adım kaydırılacak şekilde oluşan görüntünün boyutu ise 2x2 olur.



Şekil 4.9: 5x5'lik giriş görüntüsüne 2x2 filtre ile bir ve iki adım kaydırmalı maksimum havuzlama işleminin uygulanması [32].

4.1.1.4 Tam bağlantı katman

Derin öğrenmede, yoğun katman olarak da bilinen tam bağlantılı katman, tüm nöronların önceki katmandaki tüm nöronlara bağlı olduğu bir tür yapay sinir ağı katmanıdır. Tamamen bağlı katman, girdiye bir dizi ağırlık uygulayarak ve bir yanlılık terimi ekleyerek girdi verileri üzerinde matematiksel bir işlem gerçekleştirir. Bu işlem, girdi verilerini yeni bir özellik kümesiyle eşleyen doğrusal bir dönüşüme benzer. Tam olarak bağlı bir katmanın çıktısı, ağı doğrusal olmamayı getirmek için genellikle düzeltilmiş doğrusal birim (ReLU) gibi bir etkinleştirme işlevinden geçirilir.

Tamamen bağı bir katmanın amacı, sinir ağı girdi özellikleri ile çıktısı arasındaki doğrusal olmayan ilişkileri öğrenmektir. Eğitim sırasında ağırlıkları ve önyargıları ayarlayarak, sinir ağı kalıpları tanımayı öğrenebilir ve girdi verileri hakkında tahminler yapabilir.

Tamamen bağlantılı katmanlar, ileri beslemeli sinir ağı ve evrişimli sinir ağı gibi derin öğrenme modellerinde yaygın olarak kullanılır. Öğrenilen özelliklerin çıktı tahminine nihai eşleşmesini sağladıkları için tipik olarak ağı sonuna doğru yerleştirilirler.

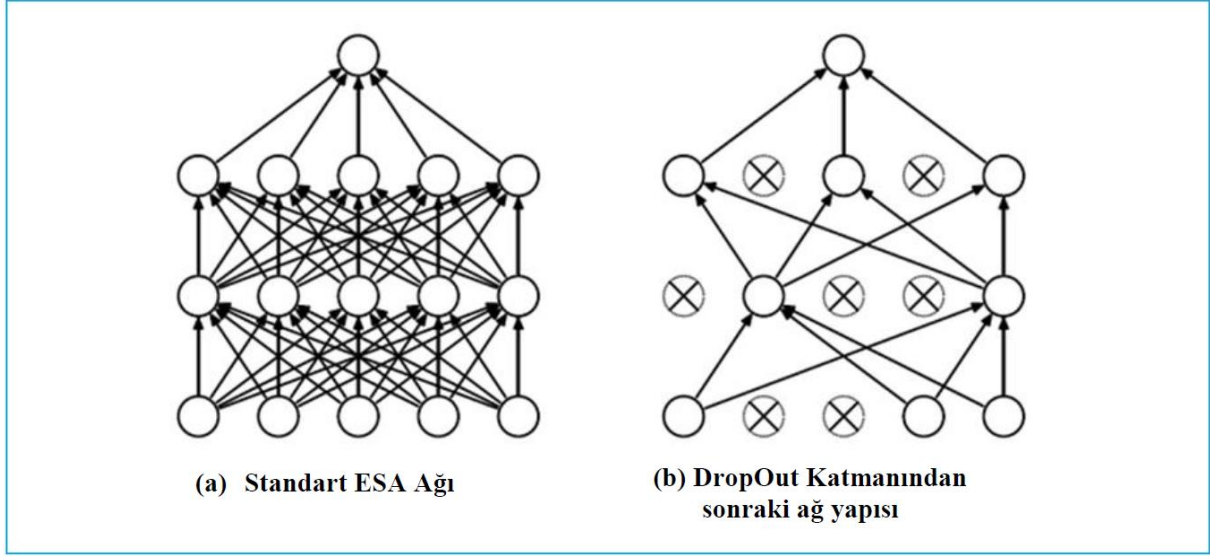
4.1.1.5 Seyreltme (DropOut) katmanı

Bu teknik, en ilginç düzenleme tekniklerinden biridir. Aynı zamanda çok iyi sonuçlar verir ve sonuç olarak derin öğrenme alanında en sık kullanılan düzenleme tekniğidir.

Seyreltmeyi anlamak için sinir ağı yapımızın aşağıda gösterilene benzer bir yapı taşıdığını varsayalım: Peki seyreltme ne yapar? Her yinelemede, bazı düğümleri rastgele seçer ve onları tüm gelen ve giden bağlantılarıyla birlikte kaldırır. Dolayısıyla, her yinelemede farklı bir düğüm kümesi vardır ve bu farklı bir çıktı kümesiyle sonuçlanır. Makine öğreniminde bir topluluk tekniği olarak da düşünülebilir.

Kaç düğümün seyreltmesi gerektiğini seçme olasılığının oranı olarak, seyreltme fonksiyonunun hiper parametresi olarak alınır. Bu oran 0 ile 1 arasında bir değer olmak zorundadır. Genellikle bu oran 0,5 olarak seçilir. Tüm katmanlar için aynı oran kullanılmak zorunda değildir. Farklı katmanlar için farklı değer uygulanabilir. Seyreltme, gizli katmanlara, tam bağı katmana ve giriş katmanlarına uygulanabilir. Seyreltme, hesaplama açısından ucuz ama güçlü bir düzenleme yöntemidir. Seyreltme, eğitim için ucuz bir yaklaşım sağlar ve üstel olarak çok sayıda sinir ağı paketlenmiş bir topluluğunu değerlendirir. Seyreltmenin bir diğer önemli avantajı, kullanılabilecek model türünü veya eğitim prosedürünü önemli ölçüde sınırlamamasıdır. Dağıtılmış bir temsil kullanan ve stokastik gradyan inişiyle eğitilebilen hemen hemen tüm modellerle iyi çalışır. Bu nedenlerden dolayı, daha fazla rastgelelik sağlamak için büyük bir sinir ağı yapısına sahip olduğumuzda genellikle seyreltme tercih edilir.

ESA'da büyük veriler ile eğitim işlemi yapıldığı için bazen ağı ezberleme yapar. Ağı ezberlemesinin önüne geçmek için bu katman kullanılır. Bu katmanda uygulanan temel mantık ağı bazı düğümlerinin kaldırılmasıdır. Şekil 4.10 (a)'da ağı orijinal yapısı gösterilirken, (b)'de seyreltme katmanından sonraki hali gösterilmiştir.



Şekil 4.10: Seyreltme katmanının Uygulanması [32].

4.1.1.6 Çıktı katmanı

Sınıflandırma katmanı, bir sinir ağındaki modelin çıktısını veya tahminini üretmekten sorumlu olan son katmandır. Tam bağlı katmandan hemen sonra gelmektedir. Tipik olarak, evrişimli katmanlar tarafından oluşturulan özellik haritalarının düzleştirilmiş bir temsili veya tekrarlayan katmanlar tarafından üretilen gizli durum değerlerinin bir vektörü olabilen önceki katmanın çıktısını girdi olarak alan tam olarak bağlı bir katmandır.

Sınıflandırma katmanının amacı, bir önceki katmanın çıktısını olası çıktı sınıfları üzerinden bir olasılık dağılımına dönüştürmektir. Bu, tipik olarak, katmanın çıktısına, çıktı değerlerini toplamı 1 olan olasılıklara dönüştüren bir softmax aktivasyon fonksiyonu uygulanarak elde edilir. Çıktı katmanındaki çıktı adedi çözeceğimiz problemin sınıf boyutu ile aynı olmalıdır. Örneğin çözeceğimiz problem rakamları bulan bir model ise çıktı katmanının boyutu 0'dan 9'a kadar olan rakamları temsil eden 10 sınıf olmalıdır. Daha sonra, en yüksek olasılığa sahip sınıf, modelin tahmin edilen çıktısı olarak seçilir.

Bazı durumlarda, sınıflandırma katmanı, aşırı uydurmayı önlemek ve modelin genelleştirme performansını iyileştirmek için bırakma veya toplu normalleştirme gibi ek düzenleme tekniklerini de içerebilir.

Hata Fonksiyonları

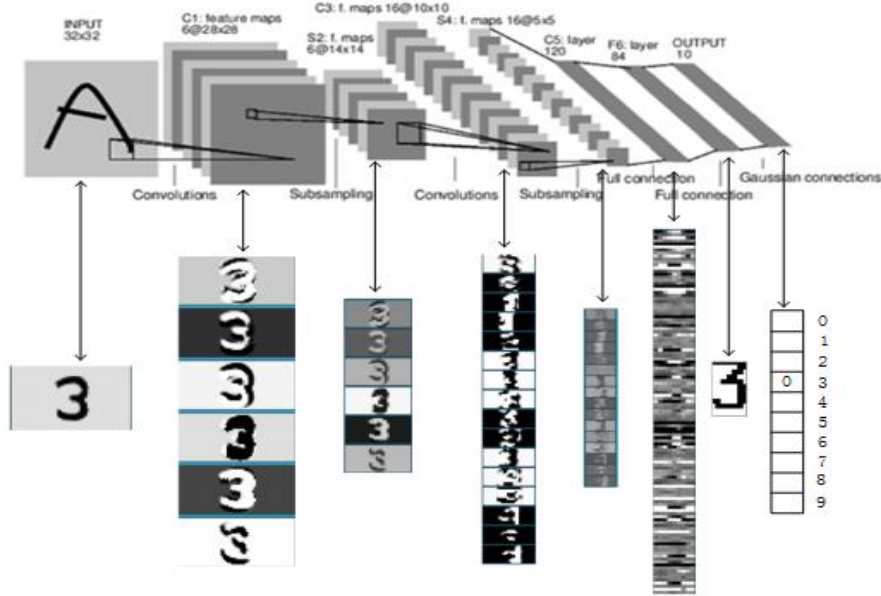
Bir sinir ağının eğitimindeki temel hedef ağın hata fonksiyonunu azaltmaktır. İyi eğitilen bir yapay sinir ağı en düşük hataya sahiptir. Yapay sinir ağındaki hata, hata ölçüm fonksiyonları ile hesaplanır. Bu hata fonksiyonları eğitim kümesindeki örneklerin hatalarının tümünü temsil eder. Problemin türüne ve ağı tasarlayan kişinin tercihinine göre değişik hata fonksiyonları kullanılabilir. En yaygın kullanılan hata fonksiyonları: ikinci dereceden (kuadratik) hata fonksiyonu (Aynı zamanda Mean Squared Error (MSE) olarak bilinir), Çapraz entropi hata fonksiyonu ve Log-likelihood hata fonksiyonu'dır.

4.2 DERİN ÖĞRENME MODELLERİ

Bir Evrişimli Sinir Ağı (CNN veya ConvNet), minimum ön işleme sahip piksel görüntülerinden görsel kalıpları doğrudan tanımak için tasarlanmış özel birçok katmanlı sinir ağı türüdür. ImageNet projesi, yazılım programlarının nesneleri ve sahneleri doğru şekilde sınıflandırmak ve tespit etmek için rekabet ettiği ImageNet Büyük Ölçekli Görsel Tanıma Yarışması (ImageNet Large Scale Visual Recognition Challenge-ILSVRC) yüzlerce obje kategorisi ve milyonlarca resimden obje sınıflandırmaya yönelik yapılan bir faaliyettir. Bu yarışma 2010 yılından beri düzenli olarak yapılmakta ve 50'den fazla kuruluş yarışmaya katılım sağlamaktadır. Burada ImageNet Büyük Ölçekli Görsel Tanıma Yarışmasının en iyi rakiplerinin CNN modellerine değinilecektir.

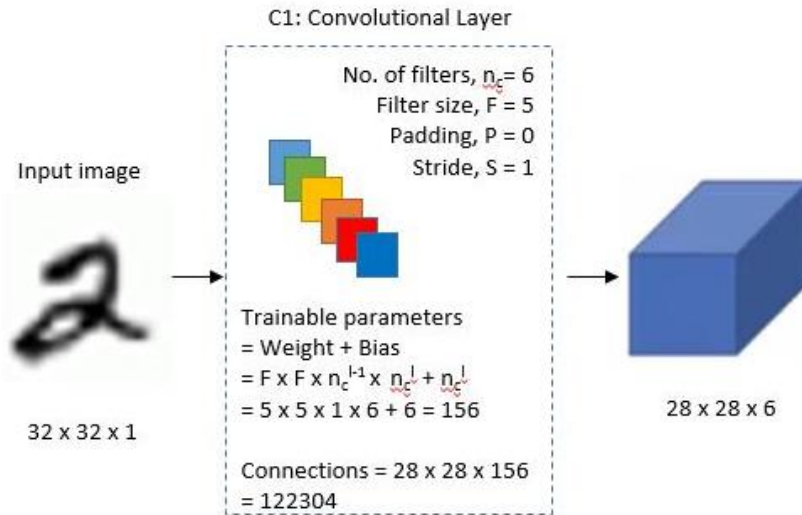
4.2.1 LeNet-5

Yann LeCun, Leon Bottou, Yosuha Bengio ve Patrick Haffner, 1990'lı yıllarda LeNet-5 (bkz. Şekil 4.11) adını verdiği el yazısı ve makine baskısı karakter tanıma için bir sinir ağı mimarisi önerdi. LeNet, 32x32 piksel gri tonlamalı giriş resimlerinde sayısallaştırılmış çekler üzerindeki el yazısı rakamları tanımak için birkaç banka tarafından uygulandı. Mimarinin basit ve anlaşılması kolaydır, bu yüzden çoğunlukla Konvolüsyonel Sinir Ağının öğretilmesinde ilk adım olarak kullanılır. LeNet mimarisinde, üç farklı katman türü olduğunu görüyoruz: Konvolüsyon Katmanlar, Havuz Katmanları ve Tamamen Bağlantılı Katmanlar. Bunlar, tüm CNN'lerde bulunan üç temel katman türüdür. LeNet-5 mimarisi iki kümeli evrişimli ve ortalama havuzlama katmanını, ardından yassılaştırılmalı evrişimli bir katmanı, devamında iki tam bağlı katmanı ve son olarak softmax sınıflandırıcısını içerir.



Şekil 4.11: LeNet-5 yapısı.

İlk Katman: Girişte görüntü boyutu 32x32 piksel olan gri tonlamalı sayı veya harf karakterleri vardır. Bu görüntüler 5x5 boyutundaki 6 farklı filtre ile evrimleştirme sonucu 32x32x1 boyutundaki görüntü 28x28x6 boyutundaki görüntüye dönüşür.



Şekil 4.12: C1:Konvolüsyon katmanı.

Parametre hesabı çoğumuz biliyoruz ancak burada hesaplama yöntemine tekrar değinecek olursak, giren görüntüler 32x32 boyutunda tek kanallı (n_c) yani gri tonlamalı görüntülerdir. Filtre (F)

boyutu ise 5x5 yani (5,5) tir. Kullanılan filtre sayısı (n_c^1) ise 6 ve kullanılan bias ağırlık sayısı da 6 dır. Çıkış görüntü boyutu ($G_c \times G_c$), 28x28x6 olmaktadır. Bu durumda:

$$Toplam\ Ağırlık\ Sayısı = F \times F \times n_c \times n_c^l$$

$$= 5 \times 5 \times 1 \times 6 = 150$$

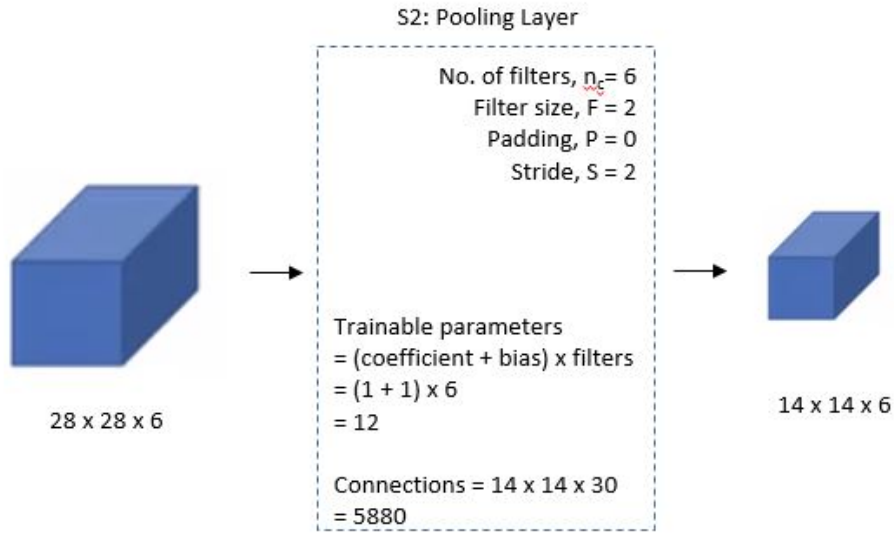
$$Toplam\ Parametre\ Sayısı = Toplam\ Ağırlık\ Sayısı + Bias\ sayısı$$

$$= 150 + 6 = 156$$

$$Bağıntı\ sayısı = G_c \times G_c \times Toplam\ Parametre\ Sayısı$$

$$= 28 \times 28 \times 156 = 122304$$

İkinci Katman: Birinci katmandan sonara havuzlama katmanına gelen görüntü, 2x2 filtre boyutunda ve 2 adım kaydırma yöntemi ile ortalama havuzlamaya veya örnekleme uygulanarak görüntünün boyutu 14x14x6'ya düşürülür.



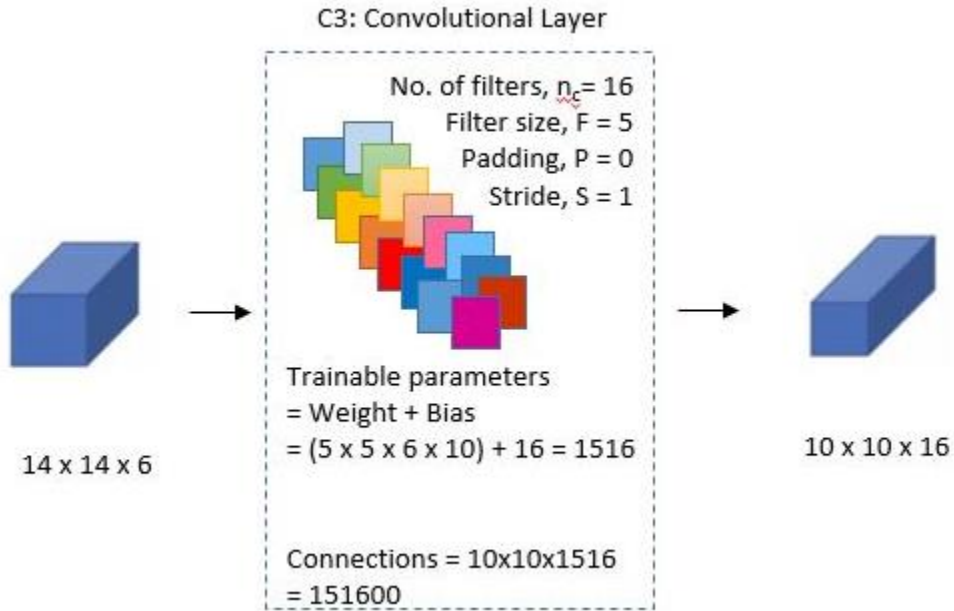
Şekil 4.13: S2:Havuzlama katmanı.

Üçüncü Katman: Daha sonra, gelen görüntüye 5x5 boyutunda 16 özellik haritası bir adım kaydırmalı olarak ikinci bir evrişime tabi tutulur. Yalnız bu katmanda 16 özellik haritasının sadece 10 tanesi aşağıdaki Tablo 4.1'de gösterildiği gibi bir önceki 6 özellik haritasına bağlanır.

Tablo 4.1: LeNet’te katmanlardaki özellik haritaları arasındaki bağıntı.

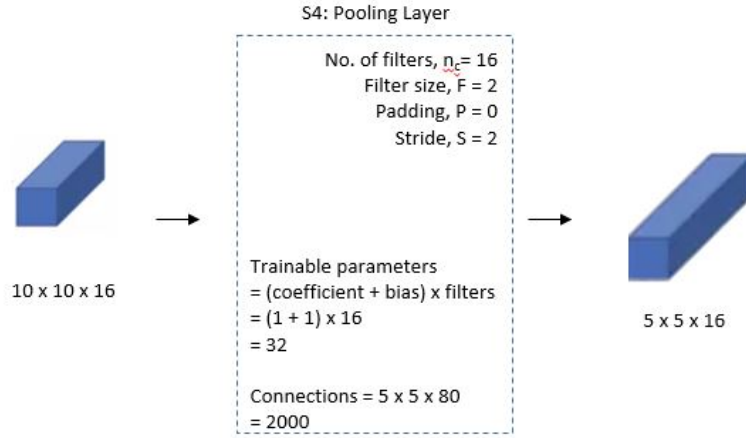
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	X				X	X	X			X	X	X	X		X	X
1	X	X				X	X	X			X	X	X	X		X
2	X	X	X				X	X	X			X		X	X	X
3		X	X	X			X	X	X	X			X		X	X
4			X	X	X			X	X	X	X		X	X		X
5				X	X	X			X	X	X	X		X	X	X

Bunun temel sebebi ağdaki simetriyi kırmak ve bağıntı sayısını makul sınırlar içinde tutmaktır. Bu nedenle, bu katmanlardaki parametre sayısı 2400 yerine 1516’ya düşmektedir. Benzer şekilde bağıntı sayısı da 240000 yerine 151600’ya düşmektedir. Bu adım sonunda görüntü boyutu 10x10x16 olmaktadır.



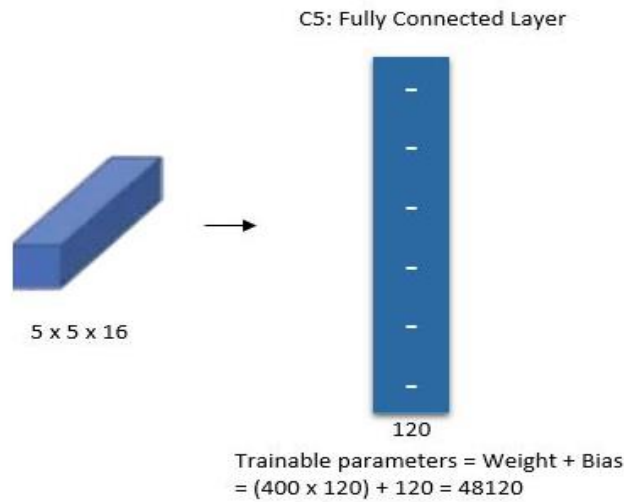
Şekil 4.14: C3:Evrişim Katmanı.

Dördüncü katman: Dördüncü katman yine filtre büyüklüğü 2x2 ve 2 adım kaydırmalı bir ortalama havuzlama katmanıdır. Bu katman, 16 özellik haritasına sahip olması dışında ikinci katmanla aynıdır. Böylece çıkış görüntüsü 5x5x16 boyutuna düşürülmektedir.



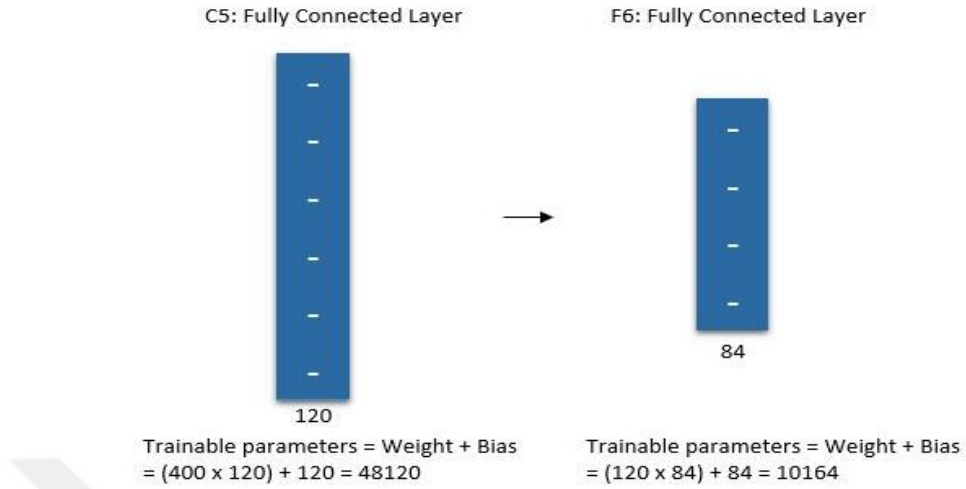
Şekil 4.15: Ortalama havuzlama katmanı.

Beşinci katman: Beşinci katman, her biri 1×1 boyutunda haritalara sahip 120 özellikli, tamamen bağlı bir evrişim katmandır. Beşinci katmandaki 120 birimin her biri, dördüncü katmandaki tüm 400 düğüme ($5 \times 5 \times 16$) bağlanır.



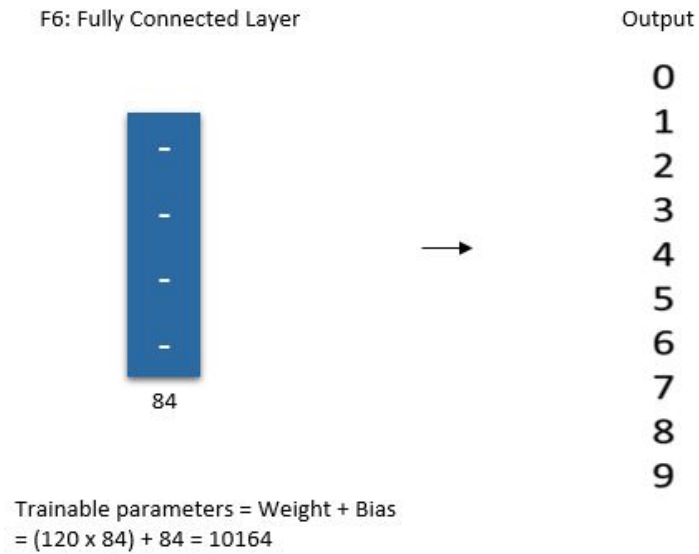
Şekil 4.16: Tam bağlı evrişim katmanı.

Altıncı katman: Altıncı katman, beşinci katmandan gelen 120 birim bu katmanda 84 birime indirgenen tam bağlı katmandır.



Şekil 4.17: Tam bağlı katman.

Çıkış katmanı: Son olarak da 0 ile 9 arasındaki rakamlara karşılık gelen 10 olası değere sahip, tamamen bağlı bir softmax çıkış katmanı vardır. Tüm katmanlarda aktivasyon fonksiyonu olarak tanjant hiperbolik fonksiyonu kullanılmıştır.



Şekil 4.18: Tam bağlı çıkış katmanı.

LeNet mimarisinin detaylı yapısı Tablo 4.2’de kullanılan katmanlar, giriş-çıkış boyutları ve kullanılan aktivasyon fonksiyonlarının isimleri yer almaktadır.

Tablo 4.2: LeNet mimarisi özet tablosu.

Layer		Feature Map	Size	Kernel Size	Stride	Activation
Input	Image	1	32x32	-	-	-
1	Convolution	6	28x28	5x5	1	tanh
2	Average Pooling	6	14x14	2x2	2	tanh
3	Convolution	16	10x10	5x5	1	tanh
4	Average Pooling	16	5x5	2x2	2	tanh
5	Convolution	120	1x1	5x5	1	tanh
6	FC	-	84	-	-	tanh
Output	FC	-	10	-	-	softmax

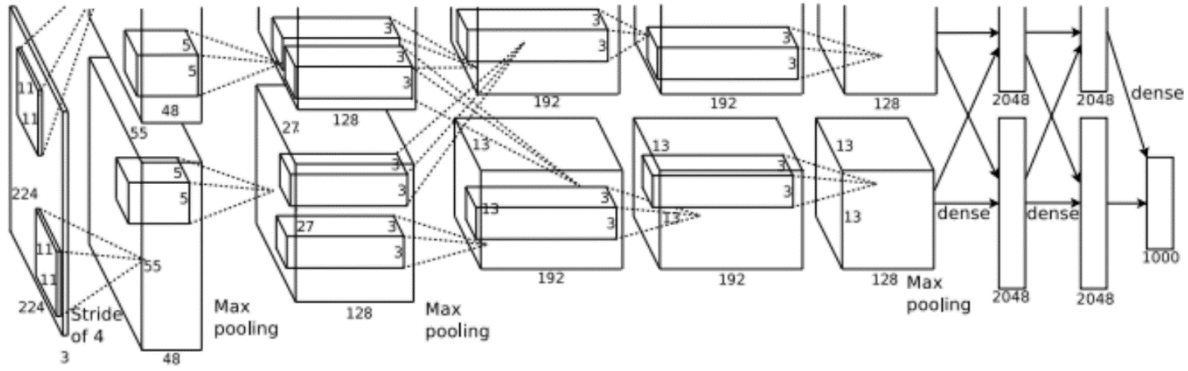
4.2.2 AlexNet

Her ne kadar LaNet'i kullanan, bilgisayarlı görme ve makine öğrenimi ile uğraşan kişiler evrimsel sinir ağlarını biliyor olmalarına rağmen bu alana hakim değillerdi. Çünkü LeNet küçük veri setleri üzerinde başarılı sonuçlar elde etse de, evrimsel ağların daha büyük, daha gerçekçi veri kümeleri üzerinde eğitilme performansı ve uygulanabilirliği hakkında henüz çalışmalar yapılmamıştı. Sinir ağları genellikle destek vektör makinaları gibi diğer makine öğrenme yöntemleri ile aşılmaya çalışılıyordu.

1990'larda her ne kadar bazı sinir ağlarını geliştirme çalışmaları yapılmış olsa da, bu çalışmalar çok kanallı, çok katmanlı evrimsel sinir ağlarını çok sayıda parametreye sahip yapacak kadar güçlü değildiler. Dahası veri setleri hala nispeten küçüktü. Bu engellere ek olarak, parametrelere ilk değerin atanması, geliştirilmiş stokastik gradyan iniş çeşitleri, farklı aktivasyon fonksiyonları ve etkili düzenleme teknikleri gibi sinir ağlarının eğitimi için önemli püf noktaları hala eksikti.

2012 yılında yapılan 2012 ILSVRC ImageNet Büyük ölçekli görsel tanıma yarışmasında birincilik elde eden derin öğrenme mimarisi AlexNet olmuştur. Örüntü tanımda hata oranını %26'lardan %15'lere indirmişti. Bu başarı sayesinde AlexNet en çok bilinen derin öğrenme mimarilerinden biri haline gelmiştir. Ağ, Yann LeCun ve arkadaşlarının LeNet ile çok benzer bir

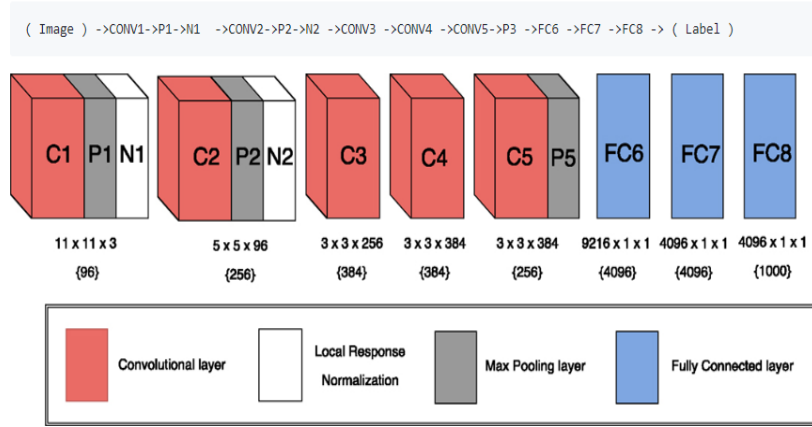
mimarisine sahipti , ancak daha derin, katman başına daha fazla filtre ve yığılmış evrişimli katmanlar kullanılmıştır. Mimari 11x11, 5x5, 3x3 boyutlu filtreler, konvolüsyon, maksimum havuz, bırakma (DropOut), veri büyütme (data augmentation), momentumlu SGD'den oluşuyordu. Her kıvrımlı ve tam bağlantılı katmandan sonra ReLU aktivasyonlarını eklenmiştir. Aktivasyon fonksiyonu olarak ReLu kullanılmasının sebebi klasik tanh fonksiyonundan daha hızlı olmasıdır. Bu sebepten dolayı eğitim süresi kısalmıştır. Eğitim sırasında ezberlemeyi ve takılmaları önlemek için bırakma (DropOut) katmanı kullanılmıştır. Ağırlık gecikmeleri ve momentum değerleri için gradyan iniş modeli (SGD) kullanılmıştır. AlexNet mimarisi aşağıdaki şekilde yer almaktadır.



Şekil 4.19: AlexNet mimarisi [35].

Her katman kendi üzerindeki işlemi yaptıktan sonra bir sonraki katmana çıktısını giriş olarak aktarmaktadır. AlexNet'in girişi 256x256 boyutunda bir RGB görüntüsüdür. Bu, eğitim setindeki tüm görüntülerin ve tüm test görüntülerinin 256x256 boyutunda olması gerektiği anlamına gelir. Giriş verileri 1000 farklı sınıfa sahip (kedi, köpek vb.) görüntülerden oluşmaktadır. Çıkış katmanı ise 1000 sayıdan oluşan bir vektördür ve bu vektörün tüm elemanlarının toplamı 1'dir. Giriş verisi ağ içerisine aktarılırken katmanlar arasındaki parametre değerleri oldukça fazladır. Normal bir işlemci ile bu işlemin yapılması çok uzun süre alacağından işlem süresini düşürmek için ekran kartı işlemcileri olan GPU'lar kullanılmıştır. GPU'lar sayesinde paralel işlemler yapılmaktadır. Yani aynı anda daha fazla işlemin yapılması anlamına gelmektedir. Bu sayede haberleşme yükü düşük tutuluyor ve bu da genel olarak iyi performans elde edilmesine yardımcı oluyor. Derin

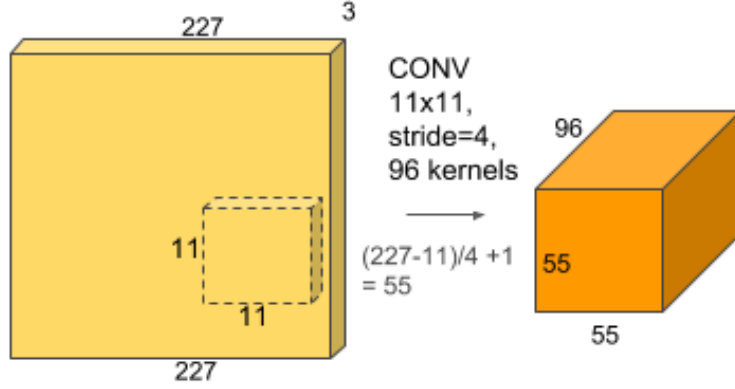
öğrenme mimarilerinde GPU (Grafical Processing Unit) işlemcileri daha iyi performanslar ile çalışmaktadır.



Şekil 4.20: AlexNet yapısı [36].

25 katmandan oluşan AlexNet derin öğrenme modelinde temel olarak 5 katman yer almaktadır.

Birinci katman: AlexNet'te ilk katman evrişim katmanıdır. Bu katmana giren görüntülerin boyutları 227x227x3'tür. Bu katmanda toplam 96 filtreye kullanılmakta olup, her filtrenin boyutu 11x11x3 şeklindedir. Evrişim sırasında kullanılan kaydırma adım sayısı 4'tür. Bu katmandan çıkan görüntü boyutu ise $(227-11/4 + 1) \times (227-11/4 + 1) \times 96$ (bölümlerde yer alan 4 adım sayısından gelmektedir) yani 55x55x96'dır. Ancak sistemde 2 adet GPU kullanıldığından her GPU'ya 55x55x48 boyutunda veri çıkışı olmaktadır. Burada evrişim sırasında yapılan işlem normal matris çarpma işlemidir. Ancak Fourier dönüşümü veya Winograd gibi farklı algoritmalar da kullananlar vardır.



Şekil 4.21: Konvolüsyon katmanı.

$$\text{Toplam Ağırlık Sayısı} = F \times F \times n_c \times n_c^l$$

$$= 11 \times 11 \times 3 \times 96 = 34.848$$

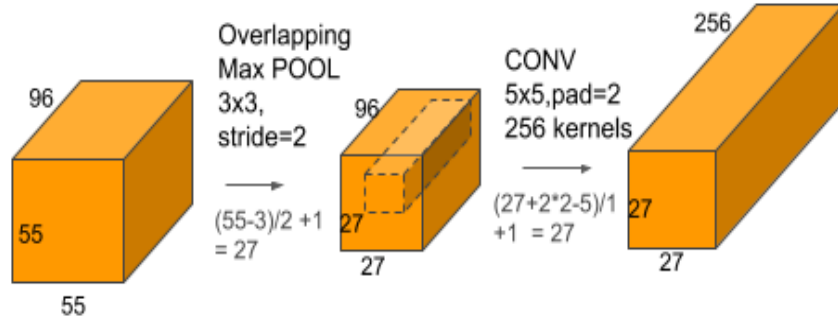
$$\text{Toplam Parametre Sayısı} = \text{Toplam Ağırlık Sayısı} + \text{Bias sayısı}$$

$$= 34.848 + 96 = 34.944$$

$$\text{Bağıntı sayısı} = G \times G \times \text{Toplam Parametre Sayısı}$$

$$= 55 \times 55 \times 34.944 = 105.705.600$$

İkinci katman: İkinci katman maksimum havuzlama işlemi takip eden konvolüsyon katmanıdır. Bu katmana giren görüntünün boyutları 55x55x96'dır. Maksimum havuzlama işlemi ile görüntü boyutu $(55-3/2 + 1) \times (55-3/2 + 1) \times 96$ yani 27x27x27 ye dönüşür. Çünkü havuzlama işleminde kullanılan pencereler 3x3 boyutunda ve kaydırma adım sayısı 2'dir. Havuzlama işleminin en önemli amacı hesaplamayı azaltmak ve fazla uyumu kontrol etmektir. Bu katmanda kullanılan filtre sayısı 256 ve her filtrenin boyutları da 5x5x48'dir. Bu katmanın çıkışında elde edilen görüntü 27x27x256 boyutundadır ancak her bir GPU'ya düşen görüntü boyutu 27x27x128'dir.



Şekil 4.22: AlexNet'in ikinci katmanı.

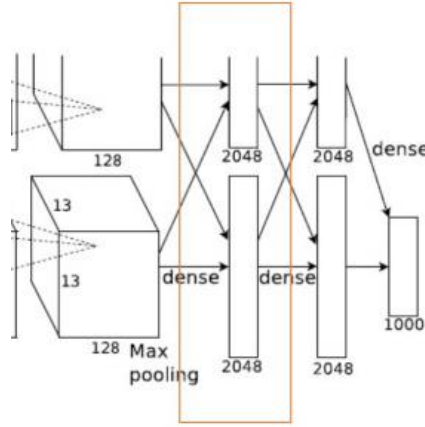
$$\begin{aligned} \text{Toplam Ağırlık Sayısı} &= F \times F \times n_c \times n_c^l \\ &= 5 \times 5 \times 96 \times 256 = 614.400 \end{aligned}$$

$$\begin{aligned} \text{Toplam Parametre Sayısı} &= \text{Toplam Ağırlık Sayısı} + \text{Bias sayısı} \\ &= 614.400 + 256 = 614.656 \end{aligned}$$

$$\begin{aligned} \text{Bağıntı sayısı} &= G \times G \times \text{Toplam Parametre Sayısı} \\ &= 27 \times 27 \times 614.656 = 448.084.224 \end{aligned}$$

Bu katmandan sonraki üçüncü, dördüncü ve beşinci katmanlar benzer şekilde işlem görmektedir. Sadece kullandıkları filtre sayısı ve boyutu birde kaydırma adım sayıları değişmektedir. Örneğin üçüncü katmanda kullanılan filtre boyutu 3x3, kullanılan filtre sayısı 384 adet ve kaydırma adım sayısı birdir.

Altıncı katman: Altıncı katman tam bağlı katmandır. Bu katmana giren görüntünün boyutu 6x6x256 dir. Bu giriş görüntüsü öncelikle bir vektöre dönüştürülür. Daha sonra buna bildiğimiz matris çarpma işlemi uygulanarak her GPU için 2048x1 boyutunda bir çıkış matrisi elde edilir.



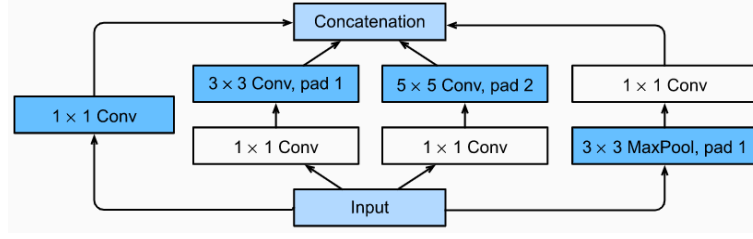
Şekil 4.23: AlexNet 6. Katman.

Bu katmandan sonraki yedinci ve sekizinci katmanlarda da benzer şekilde işlemler görülmektedir. Bu mimaride aktivasyon fonksiyonu olarak ReLU kullanılmakta olup, düzenlileştirme (droupout) oranı olarakta 0.5 kullanılmaktadır. Ayrıca yığın boyutu (batch size) 128 alınmakta olup, öğrenme (learning rate) oranı olarak da 0.01 değeri seçilmiştir.

4.2.3 GoogLeNet

GoogLeNet 2014’de NiN’in güçlü yönlerini ve tekrarlanan blokların paradigmalarını birleştiren bir yapı önererek ImageNet yarışmasını kazandı[37]. Makalenin odak noktalarından biri hangi boyuttaki evrişim çekirdeklerinin en iyi olduğu sorusunu ele almaktı. Bu bakış açısıyla bazen çeşitli boyutlarda çekirdeklerden oluşan bir kombinasyon kullanmanın avantajlı olabileceğini gösterdiler.

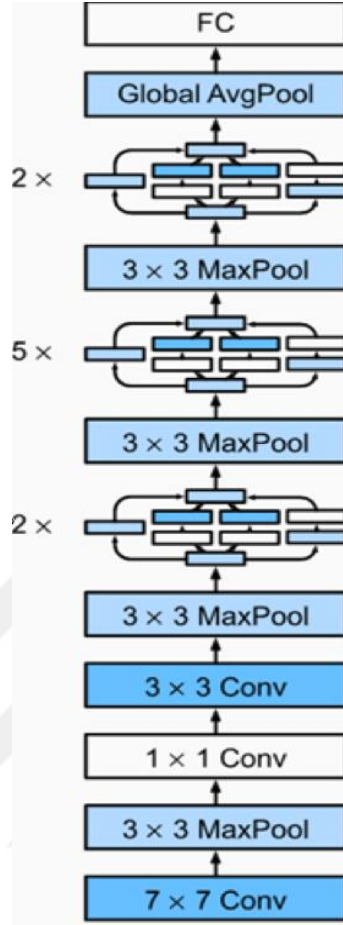
GoogLeNet deki en temel evrişimli blok başlangıç(inception) bloğu olarak adlandırılır.



Şekil 4.24: Başlangıç bloğunun yapısı [38].

Şekil 4.24'te gösterildiği gibi başlangıç bloğu 4 paralel yoldan oluşur. İlk üç yol farklı boyutlardan bilgi çekmek için 1x1, 3x3, 5x5 boyutlu evrişimli katmanları kullanır. Ortadaki iki yol, kanalların sayısını azaltmak için girdi üzerinde 1x1 boyutunda bir evrişim gerçekleştirerek model karmaşıklığını azaltır. Dördüncü yolda girdiye ve çıktıya aynı yüksekliği ve genişliği vermek için uygun havuzlama kullanılır. Sonuç olarak her yoldaki çıktılar, kanal boyutu boyunca birleştirilir ve bloğun çıktısı oluşturulur. Başlangıç bloğunun yaygın olarak ayarlanmış üst değişkenleri(hyperparameter) her katmandaki çıktı kanallarının sayısıdır.

GoogLeNet Mimarisi toplam 9 başlangıç bloğundan oluşan bir yığın ve tahminlerini üretmek için küresel ortalama havuzlama(global average pooling) kullanır (Şekil 4.25).



Şekil 4.25: GoogLeNet mimarisi [38].

Başlangıç bloklarının arasındaki maksimum havuzlama boyutsallığı azaltır. İlk modül AlexNet ve LeNet'e benzerdir. Blok yığını VGG'den miras alınır ve küresel ortalama havuzlama, sonunda tamamen bağlı(fully-connected) katman yığınlarını önler.

İlk modül 64 kanallı 7x7 evrişimli katman kullanılır.

İkinci modül de iki tane evrişimli katman kullanılır. İlk olarak 64 kanallı 1x1 evrişim katmanı, ardından kanal sayısını 3 katına çıkaran 3x3 evrişim katman kullanılır. Bu başlangıç bloğundaki ikinci yola karşılık gelir.

Üçüncü modül seri olarak iki tam başlangıç bloğunu birbirine bağlar. İlk başlangıç bloğunun çıktı kanallarının sayısı

$$64 + 128 + 32 + 32 = 256$$

dır. Dört yol arasındaki çıktı kanal sayısı oranı

$$64 : 128 : 32 : 32 = 2 : 4 : 1 : 1$$

dir. İkinci ve üçüncü yollar ilk olarak girdi kanallarının sayısını sırasıyla

$$\frac{96}{192} = \frac{1}{2} \text{ ve } \frac{16}{192} = \frac{1}{12}$$

oranına düşürür ve ikinci evrişim katmanına bağlanır. İkinci başlangıç bloğunun çıktı kanallarının sayısı

$$128 + 192 + 96 + 64 = 480$$

e çıkarılır. Dört yol arasındaki çıktı kanal sayısı oranı

$$128 : 192 : 96 : 64 = 4 : 6 : 3 : 2$$

dir. İkinci ve üçüncü yollar ilk olarak girdi kanallarının sayısını sırasıyla

$$\frac{128}{256} = \frac{1}{2} \text{ ve } \frac{32}{256} = \frac{1}{8}$$

oranına düşürür.

Dördüncü modül daha karmaşıktır. Burada seri olarak beş başlangıç bloğu birbirine bağlanır. Bunlar sırasıyla

$$192 + 208 + 48 + 64 = 512 ,$$

$$160 + 224 + 64 + 64 = 512 ,$$

$$128 + 256 + 64 + 64 = 512 ,$$

$$112 + 288 + 48 + 64 = 528 ,$$

$$256 + 320 + 128 + 128 = 832$$

çıktı kanallarına sahiptirler. Bu yollara atanan kanalların sayısı üçüncü modüldekine benzerdir. 3x3 evrişimli katmana sahip ikinci yol en fazla sayıda kanalı çıkarır, ardından yalnızca 1x1

evrişimli katmanla birinci yol, 5x5 evrişimli katmanla üçüncü yol ve 3x3 maksimum havuzlama katmanı ile dördüncü yol izler. İkinci ve üçüncü yollar ilk olarak orana göre kanal sayısını azaltır. Bu oranlar farklı başlangıç bloklarında farklıdırlar.

Beşinci modülde iki tane başlangıç bloğu vardır. Bu blokların çıktı kanalları sırasıyla

$$256 + 320 + 128 + 128 = 832 ,$$

$$384 + 384 + 128 + 128 = 1024$$

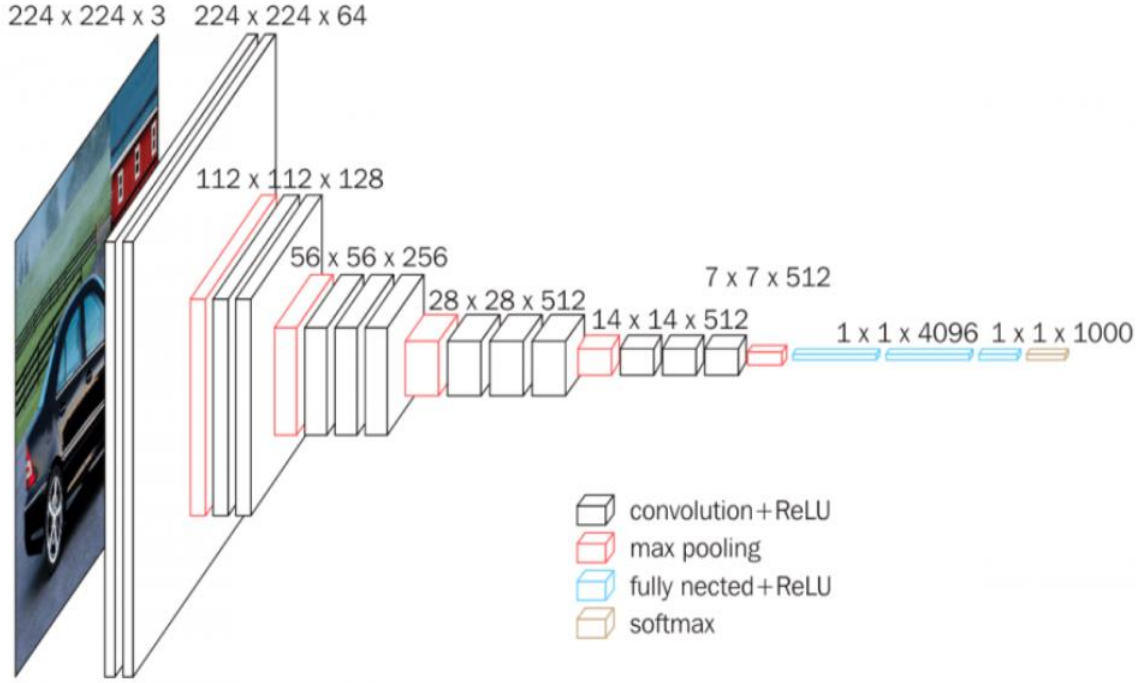
dür. Her yola atanan kanal sayısı üçüncü ve dördüncü modüllerdeki ile aynıdır, ancak belirli değerlerde farklılık gösterir. Beşinci bloğun ardından çıktı katmanı gelmektedir. Bu blok, tıpkı NiN’de olduğu gibi her kanalın yüksekliğini ve genişliğini 1’e değiştirmek için genel ortalama havuzlama katmanı kullanır.

Sonuç olarak, çıktı sayısı etiket sınıfı sayısı olan tamamen bağlı (fully-connected) katmanın ardından çıktıyı iki boyutlu bir diziye dönüştürüyoruz.

4.2.4 VGG-16

VGG’nin tam adı, Oxford Üniversitesi Bilim ve Mühendislik Bölümüne ait olan Görsel Geometri Grubu(Visual Geometry Group)’dur[1]. Karen Simonyan ve Andrew Zisserman, evrişimli ağ derinliğinin büyük ölçekli görüntü tanıma ayarındaki doğruluğu üzerindeki etkisini araştırdılar[39]. 3x3 evrişim filtreleri ile mimarilerinin derinliğini 16 katmana çıkardılar. Bulgularının adını VGG-16 adını verdiler.

VGG-16 mimarisi(Şekil 45), 12 evrişimli katmandan oluşur ve bunlardan bazılarını maksimum havuzlama katmanları, ardından dört tam bağlı katmanla son olarak 1000 çıktıli softmax sınıflandırıcısı izler.



Şekil 4.26: VGG-16 mimarisi [39].

Birinci ve ikinci katman: VGG-16 için girdi olarak 24x24 piksel boyutundaki renkli görüntüler ve bu renkli görüntüler 3 kanaldan yani kırmızı, yeşil ve mavi kanallarından oluştuğundan 24x24x3'tür. 3x3 boyutunda ve kaydırma adım sayısı 1 olan 64 filtre ile bir ve ikinci evrişimli katmanlardan girdi geçerek görüntü boyutu 24x24x64 olur.

Daha sonra VGG-16 kaydırma adım sayısı 2 olan 3x3 boyutunda bir maksimum havuzlama filtresi uygular. Bunun sonucunda görüntü boyutu 112x112x64 olarak elde edilir.

Üçüncü ve dördüncü katman: Burada kaydırma adım sayısı 1 ve 3x3 boyutunda 128 filtreli 2 tane evrişimli katman vardır. Bu katmanlar uygulandığında elde edilen çıktı boyutu indirgenerek 56x56x128 olur.

Beş ve altıncı katman: Beş ve altıncı katmanlar 1 kaydırma adım sayılı 3x3 boyutlu filtreye sahip evrişimli katmanlardır. Bunların her ikisi de 256 özellik kullanır. Bu katmanların ardından 2 kaydırma adım sayılı 3x3 boyutunda filtreye sahip maksimum havuzlama katmanları uygulanır ve 256 özellik elde edilir.

Yedi – on iki arasındaki katmanlar: Yediden on ikinci katmana ve on iki dahil toplam iki set 3 evrişimli katman ve ardından maksimum havuzlama katmanı vardır. Tüm evrişimli katmanlar 1 kaydırma adım sayılı 3x3 boyutunda 512 filtreye sahiptir. Bu katmanların çıktısı indirgenerek 7x7x512 olarak elde edilir.

On üçüncü katman: Evrişimli katman çıktısı burada her biri 1x1 boyutunda 25088 özelliğe sahip tam bağlı bir katman aracılığı ile düzleştirilir.

On dördüncü ve on beşinci katman: Tekrar bu katmanlarda 4096 nöronlu tam bağlı iki tane katman vardır.

Çıktı katman: Son olarak, 1000 olası değerli softmax çıktı katmanı vardır.

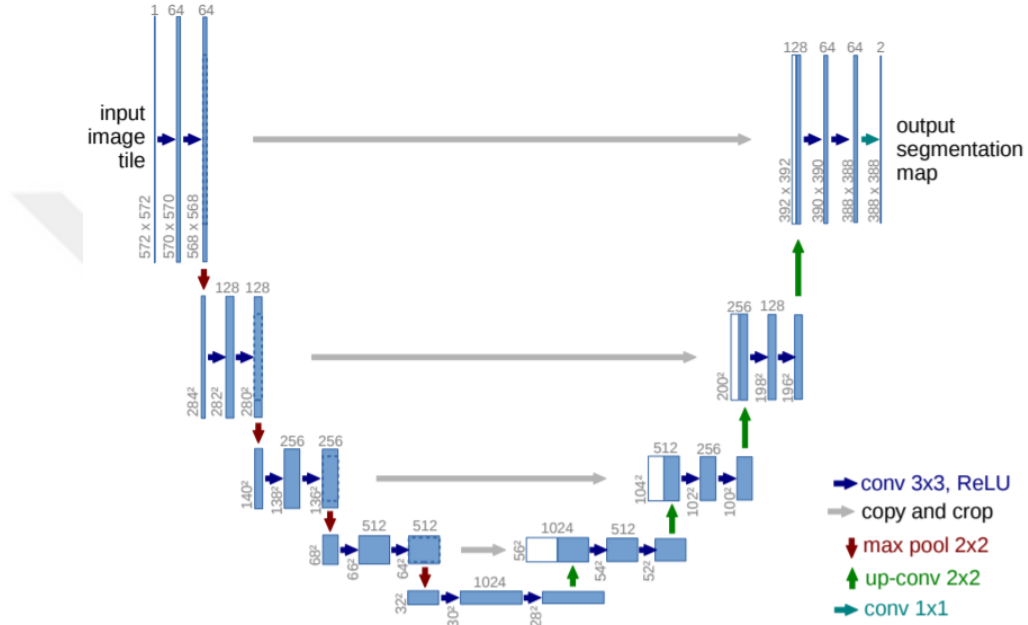
VGG-16 mimarisinin detaylı yapısı Tablo 4.3'te kullanılan katmanlar, giriş-çıkış boyutları ve kullanılan aktivasyon fonksiyonlarının isimleri yer almaktadır.

Tablo 4.3: VGG-16 mimarisinin özeti.

	Layer	Feature Map	Size	Kernel Size	Stride	Activation
Input	Image	1	224 x 224 x 3	-	-	-
1	2 X Convolution	64	224 x 224 x 64	3x3	1	relu
	Max Pooling	64	112 x 112 x 64	3x3	2	relu
3	2 X Convolution	128	112 x 112 x 128	3x3	1	relu
	Max Pooling	128	56 x 56 x 128	3x3	2	relu
5	2 X Convolution	256	56 x 56 x 256	3x3	1	relu
	Max Pooling	256	28 x 28 x 256	3x3	2	relu
7	3 X Convolution	512	28 x 28 x 512	3x3	1	relu
	Max Pooling	512	14 x 14 x 512	3x3	2	relu
10	3 X Convolution	512	14 x 14 x 512	3x3	1	relu
	Max Pooling	512	7 x 7 x 512	3x3	2	relu
13	FC	-	25088	-	-	relu
14	FC	-	4096	-	-	relu
15	FC	-	4096	-	-	relu
Output	FC	-	1000	-	-	Softmax

4.2.5 U-Net

U-Net ağ mimarisi Şekil 4.27’deki gibidir. Mimari “U” harfine benzediğinden bu isim verilmiştir. U-Net evrişimli sinir ağlarını baz alır. Encoding(sol taraf) ve decoding(sağ taraf) olmak üzere iki bölümden oluşur[40].

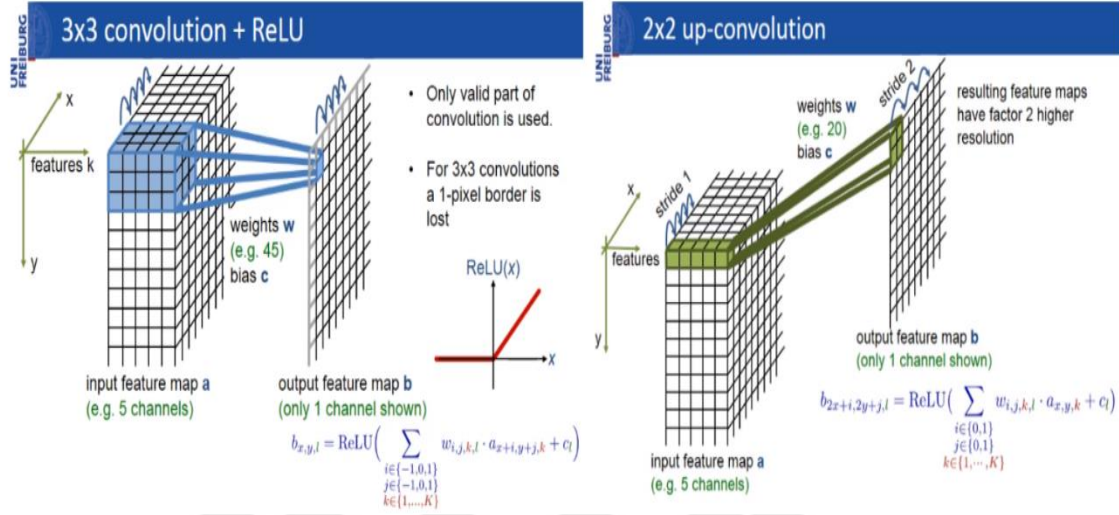


Şekil 4.27: U-Net mimarisi [41].

Encoding bölümünde ilk iki tane 1 kaydırma adım sayılı 3x3 evrişim katmanı art arda uygulanır ve her birini relu ve dördüncü seviyede 2 kaydırma adım sayılı 2x2 maksimum havuzlama işlemi takip eder. Daha sonra seyreltme(dropout) katmanı takip eden ilk evrişimli katmana uygulanır. Girdi görüntüsünün boyutu her alt örnekleme(down-sampling) adımında yarıya indirgenir ve özellik kanallarının sayısı iki katına çıkarılır. Mimarinin en alt bölümünde havuzlama katmanı olmayan 2 iki tane 3x3 evrişimli katman vardır.

Decoding bölümünde özellikleri üst örnekleme(up-sampling) yaparak girdi görüntüsünün boyutu eski haline getirilir (bkz. Şekil 4.28). Bu bölümde her adımda özellik kanallarının sayısını ikiye bölen 2x2 boyutunda evrişim katmanı ile üst örnekleme yapılır ve bunların ardından encoding bölümünde karşılık gelen özellik kanallarıyla bir birleştirme yapılır. Daha sonra bunu daha sonra her adımda bunları ilk olarak ReLU, seyreltme katmanı ve tekrardan ReLU uygulanır. Son

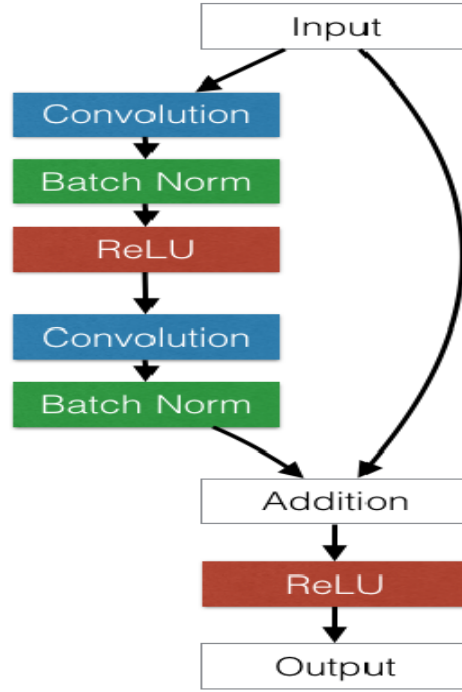
katmanda, her 64 bileşenli özellik vektörünü istenen sayıda sınıfa eşlemek için 1x1 evrişim kullanılır. Toplamda 23 tane evrişim katmanı kullanılmıştır[41].



Şekil 4.28: U-Net Encoding(sol taraf) ve Decoding(sağ taraf) İşlemleri [42].

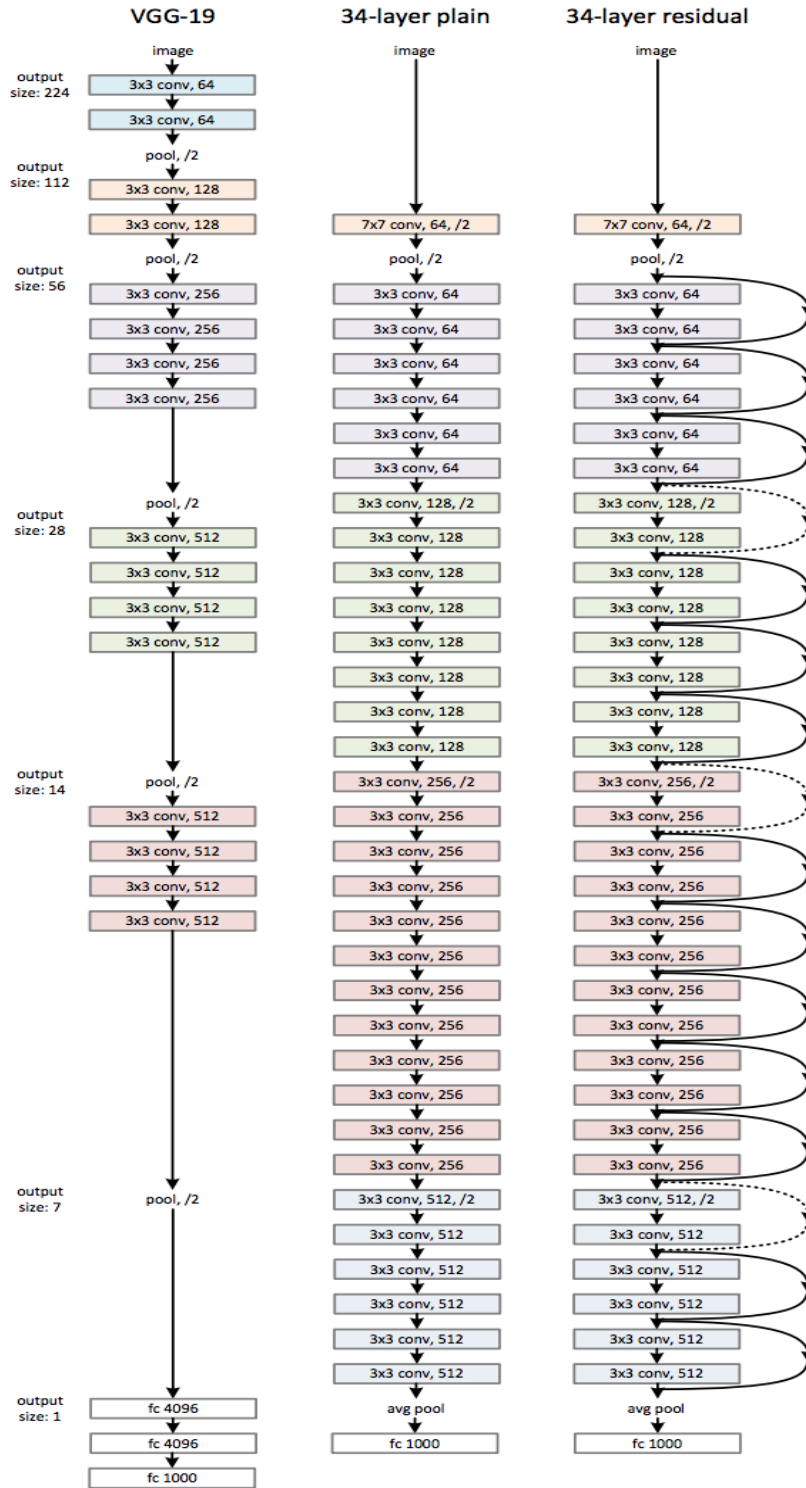
4.2.6 ResNet

Belli bir derinlikten sonra ileri beslemeli CNN'lere ek katmanlar eklemek, daha yüksek bir eğitim hatası ve daha yüksek doğrulama hatasıyla sonuçlanır. Katman eklediğimizde performans belirli bir derinliğe kadar artar ve daha sonra performans hızla azalır. ResNet makalesinde yazarlar, bu eksik öğrenmenin(underfitting) kaybolan gradient probleminden kaynaklanmadığını, çünkü bu yığın(batch) normalizasyon tekniğini kullanırken bile meydana geldiğini savunmuşlardı. Bu yüzden residual block olarak adlandırılan yeni bir kavram tanımladılar. ResNet takımı katmanları atlayabilen bağlantılar ekledi.



Şekil 4.29: Residual bloğu [43].

2015’de İmageNet yarışmasını kazanan %3.57 hata oranıyla Microsoft’dan ResNet olmuştur[43]. ResNet, ağı derinleştirmek için aynı yapının tekrar tekrar tekrarlanması anlamında bir çeşit VGG’dir. VGG’den farklı olarak 34, 50, 101 ve 152 katman gibi farklı derinlik varyasyonlarına sahiptir.



Şekil 4.30: ImageNet için örnek ağ mimarisi. Solda VGG-19 modeli[39]. Ortada 34 parametre katmanıyla plain ağı ve sağda 34 parametre katmanıyla residual ağı [44].

AlexNet 8, VGG 19 ve GoogLeNet 22 katmanlı olmasına karşın ResNet'in 152 katmana sahiptir. ResNet mimarisi residual bloklardan oluşan bir yığındır. Ana fikir sinir ağlarına bağlantılar ekleyerek katmanları atlamaktır. Her residual bloğu 3x3 evrişim katmanına sahiptir (bkz. Şekil 4.30). Son evrişim katmanından sonra bir boşluk(GAP) katmanı eklenir. 1000 sınıfı sınıflandırmak için tek bir tam bağlı katman vardır. ImageNet veri kümesi için 34, 50, 101 veya 152 katman gibi farklı derinlik çeşitlerine sahiptir. Daha derin bir ağ için, örneğin 50'den fazla katman için, verimliliği arttırmak için darboğaz(bottleneck) özelliklerini kullanır. Bu ağ yapısında dropout kullanılmamıştır.

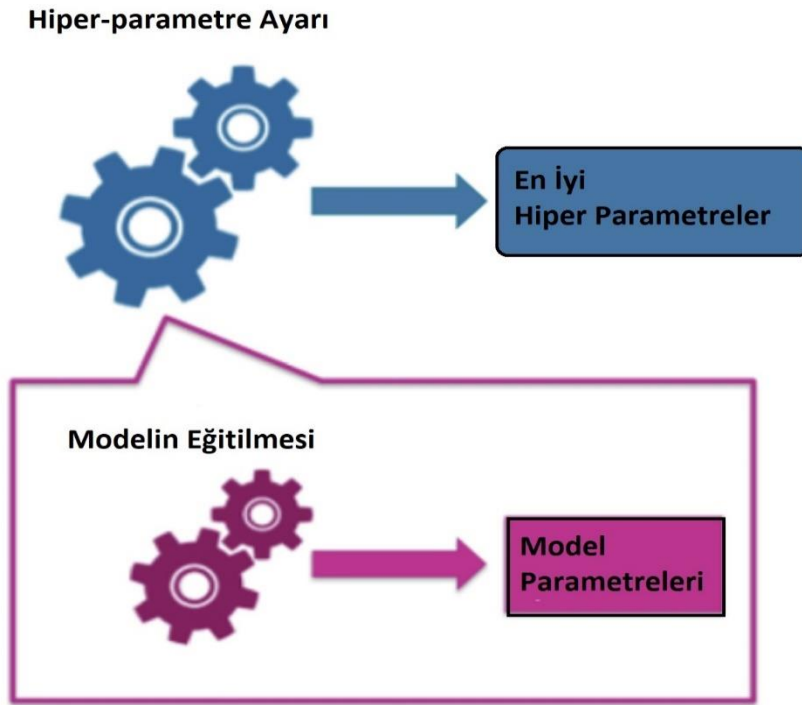
4.3 DERİN ÖĞRENME MODELLERİNDE SIK KULLANILAN HİPER-PARAMETRELER

Bundan birkaç yıl önceki bilim adamları genellikle ilgili problemi çözebilmek için problemin çözümünü modelleyen en uygun öznetelik kümelerini bulmayı amaçlamışlardır. Yani özneteliklerin çıkarılması ve bu öznetelikler arasındaki bağlantıyı en yüksek oranda sağlayacak haritalamalarla ilgili araştırma yapıyorlardı. Derin öğrenme modellerindeki farklı yaklaşımlarının gelişimi ile derin katmanlı yapay sinir ağının nasıl tasarlanacağı; mimarinin kaç katmandan oluşacağı, mimariye en uygun optimizasyon algoritmasının ne olacağının, mimaride kaç adet nöron kullanılacağı, ya da hangi aktivasyon fonksiyonunun kullanılacağı problem çözümünde daha önemli bir hale geldi.

Özellikle ekran kartlarındaki gelişim; grafik işlemcilerdeki hafızalarının artması, grafik işlemcilerin veri yolu üzerindeki anlık bit düzeyindeki anlık hesaplama sayısının artırılması ile bu kartlarda eş zamanlı işlemlerin daha hızlı yapılabilmesi bilim adamlarının daha çok gizli katmandan oluşan fazla karmaşık mimarilerin oluşturulmasına fırsat verdi. Buda bir problemi çok katmanlı ağ yapısını derin öğrenme ile çözmeyi en uygun şekilde mimarisini oluşturmak ile eşdeğer duruma geldi. Bu mimari mühendisliğinde bilim adamlarının izlenimlerinden ve deneyimlerinden sonra en çok başvurdukları araçlar hiper parametreler olmuştur.

Veri setlerinden modeller tasarlanırken, modeli optimum hale getirmek için mimaride kullanılacak algoritmaların veya tekniklerin neler olması gerektiğine mimar karar verirken hangi parametreleri kullanacağına ve parametrelerin değerlerini de göz önünde bulundurması gerekir. Örneğin destek vektör makinalarının algoritmalarında hangi filtreleri ve fonksiyonları kullanacağı veya KNN

sınıflandırma algoritmasında kaçta sınıfın olacağına (yani k değerine) modeli tasarlayan kişinin kendisi karar vermektedir. Benzer şekilde kullanılacak derin ağ modellerindeki gizli katman sayısına, her katmandaki nöron sayısına, optimizasyon fonksiyonu ve öğrenme oranına, filtre boyutuna, filtrenin kaydırma miktarına, seyreltme (dropout) değerine modelin mimarcısı karar vermektedir. Bir problemi çözmek için kullanılacak modeldeki parametrelerin neler olduğu ve oranlarının neler olacağı başlangıçta net ve kesin değildir. Parametrelerin seçimi probleme, veri kümesine, veri tipine vb. gibi faktörlere göre farklılıklar göstermektedir. Bu sebeple hangi parametrelerin kullanılacağı ve değerlerinin neler olacağı tamamen modelin mimarına kalmıştır. Bir probleme, veri setine veya veri setindeki verilerin türüne göre farklılık gösteren parametreler hiper-parametre (hyperparameters) olarak adlandırılmaktadır.



Şekil 4.31: Yapay zeka uygulamalarında parametre ve hiper parametre.

Bir modelde farklı hiper parametre grupları kullanılması modelin başarımını da farklı çıkarabilmektedir. Bu grupların her birinin model tasarımında kullanılmasında sakınca yoktur. Bununla birlikte en iyi başarıya sahip hiper parametre grubunun seçilmesi baş edilmesi gereken

önemli problemlerden biridir. Hiper parametrelerin seçimi genelde tasarımcının sezgisine, problemin türüne, daha önceki problemlerden elde edilen tecrübeler, tasarlanacak mimariye, farklı alanlardaki uygulamaların kendi problemimize yansımalarına, model içindeki tasarım bağımlılığı vb. göre değişmektedir. Son zamanlarda tasarlanacak mimariye en uygun hiper parametrelerin otomatik olarak belirlenmesi için farklı çalışmalar yapılmaktadır. Ancak halen en iyi hiper parametreleri veren bir teknik yada model tasarlanamamıştır.

4.3.1 Veri Kümesinin Boyutu

Bir veri kümesinin boyutunu ayarlamak, gereksiz yere büyük bir veri kümesi kullanmanın hesaplama ve zaman maliyetlerinden kaçınırken iyi bir model oluşturmak için yeterli veriye sahip olmak arasında doğru dengeyi sağlamakla olur. Bir veri kümesinin boyutunun nasıl ayarlanacağına ilişkin bazı genel yönergeler aşağıda verilmiştir:

- i. Sorunun karmaşıklığını göz önünde bulundurun: Çözmeye çalıştığınız sorun karmaşıksa, sorunun tüm nüanslarını yakalamak için daha büyük bir veri kümesine ihtiyacınız olabilir. Öte yandan, problem nispeten basitse, daha küçük bir veri seti yeterli olabilir.
- ii. Model performansınızı değerlendirin: Modelinizi çeşitli veri kümesi boyutları üzerinde eğitin ve modelin performansını bir doğrulama kümesi üzerinde değerlendirin. Bu size sorunuz için en uygun veri kümesi boyutu hakkında bir fikir verecektir.
- iii. Örneklem yöntemlerini kullanın: Büyük bir veri kümeniz varsa ancak hesaplama kaynaklarıyla sınırlıysanız, orijinal veri kümesinin istatistiksel özelliklerini koruyan daha küçük veri alt kümeleri oluşturmak için örneklem yöntemlerini kullanabilirsiniz. Bu, modelin kalitesinden ödün vermeden hesaplama ve zaman maliyetlerini azaltmanıza yardımcı olabilir.
- iv. Veri dağıtımını göz önünde bulundurun: Veri kümeniz dengesizse, yani bazı sınıflar diğerlerinden önemli ölçüde daha fazla veriye sahipse, modelin her sınıf için eşit miktarda veriyle eğitildiğinden emin olmak için her sınıf için veri kümesi boyutunu ayarlamamız gerekebilir.
- v. Veri artırmayı kullanın: Küçük bir veri kümeniz varsa, veri kümesinin boyutunu yapay olarak artırmak için veri artırma tekniklerini kullanabilirsiniz. Bu, yeni örnekler

oluşturmak için mevcut verilere döndürme, çevirme ve yakınlaştırma gibi dönüşümlerin uygulanmasını içerir.

Genel olarak, veri kümesinin boyutu, modelinizin performansını etkileyebilecek önemli bir hiperparametredir. Sorununuz için doğru miktarda veriye sahip olduğunuzdan emin olmak için bu hiperparametreyi dikkatlice ayarlamanız önemlidir.

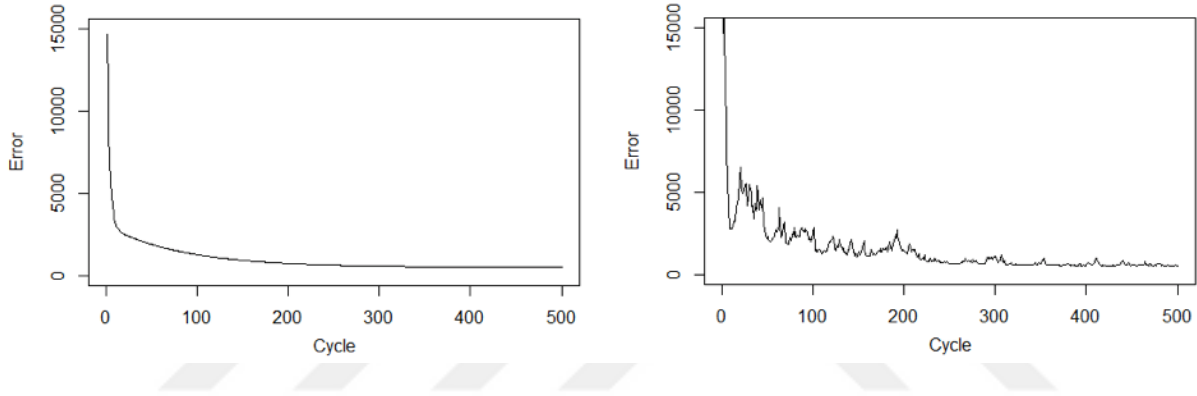
4.3.2 Batch Boyutu

Batch boyutu, her eğitim yinelemesinde işlenen numune sayısını tanımlayan bir hiperparametredir. Batch boyutunun seçimi, eğitim süreci ve ortaya çıkan modelin kalitesi üzerinde önemli bir etkiye sahip olabilir. Batch boyutunu ayarlamak için bazı adımlar şunlardır:

- i. Küçük bir toplu iş boyutuyla başlayın: Yeni bir modeli eğitirken, genellikle küçük bir toplu iş boyutuyla başlamak iyi bir fikirdir. Bu, hızlı bir şekilde yinleme yapmanıza ve modelin nasıl performans gösterdiğine dair bir fikir edinmenize olanak tanır. Küçük bir batch boyutu, eğitim sürecinin başlarında aşırı uyumu önlemeye de yardımcı olabilir.
- ii. Batch boyutunu kademeli olarak artırın: Küçük bir batch boyutuyla temel bir performans oluşturduğunuzda, modelin iyileşip iyileşmediğini görmek için batch boyutunu kademeli olarak artırabilirsiniz. Batch boyutunu küçük artışlarla artırmak (ör. batch boyutunu iki katına çıkarmak) ve her değişiklikten sonra modelin performansını değerlendirmek genellikle en iyisidir.
- iii. Performans ölçümlerini izleyin: Grup boyutunu ayarlarken, eğitim kaybı, doğrulama kaybı ve doğruluk gibi performans ölçümlerini izlediğinizden emin olun. Bu, batch boyutunu artırmanın modelin performansını iyileştirip iyileştirmediğini veya modelin aşırı takılmasına neden olup olmadığını belirlemenize yardımcı olacaktır.
- iv. Donanım sınırlamalarını göz önünde bulundurun: Batch boyutu, mevcut donanım tarafından da sınırlandırılabilir. Örneğin, sınırlı belleğe sahip bir GPU üzerinde egzersiz yapıyorsanız, verileri belleğe sığdırmak için batch boyutunu küçültmeniz gerekebilir.

- v. Çapraz doğrulama kullanın: Modelin yeni veriler üzerinde nasıl performans göstereceğine dair daha doğru bir tahmin elde etmek için, modelin performansını farklı toplu iş boyutlarıyla değerlendirmek üzere çapraz doğrulamayı kullanın.

Genel olarak, batch boyutunun ayarlanması, performans ölçütlerinin denemesini ve dikkatli bir şekilde izlenmesini gerektiren yinelemeli bir süreçtir. Yapılan çeşitli uygulamalarda veri kümesindeki girdilerin gruplar halinde eğitilmesi (batch-size) durumunda kayıp değerinin arttığı ancak hesaplama zamanının daha kısa sürdüğü tespit edilmiştir.



Şekil 4.32: Batch boyutu değiştirildiğinde hata değerinin değişimi [45].

Batch boyutu uyguladığımızda Batch boyutu değiştirildiğinde hata değerinin değişimi [45] görüleceği üzere hata miktarlarında ani değişiklikler olur. Çünkü her iterasyonda farklı veri işlendiği için bazı iterasyondaki veriler için seçilen parametreler tam uygun iken bazısında uygun olmayabilir. Bununla birlikte iterasyon değeri arttıkça bu dalgalanma miktarında düşmeler olacaktır. Batch boyutu durumunda ortaya çıkan bu gürültü durumunu azaltmanın bir yolu da küçük öğrenem değeri (learning rate) belirlemektir.

Batch boyutu değerinin seçiminde en uygun değer 1 ile eğitim kümesindeki tüm verilerin sayısı arasında bir değer olmalıdır. Bu belirlenen değer ne çok küçük ne de çok büyük olmayan bir değer olmalıdır. Bu sayede modelimizin hızlı şekilde öğrenmesi sağlayacaktır. Batch boyutunun bu şekilde belirlenebilmesiyle ilgili bazı özellikler aşağıdaki gibidir;

- i. Kullanılabilir bellek: Batch boyutu, modeli eğitmek için kullanılan donanımın kullanılabilir belleği ile sınırlıdır. Toplu iş boyutu çok büyükse, bellek bunu kaldıramayabilir ve eğitim işlemi başarısız olabilir.

- ii. Veri kümesi boyutu: Veri kümesinin boyutu, toplu iş boyutunu da etkiler. Veri kümesi büyükse, fazla uydurmayı önlemek için daha küçük bir toplu iş boyutu tercih edilebilirken, daha küçük veri kümeleri için daha büyük bir toplu iş boyutu kullanılabilir.
- iii. Model karmaşıklığı: Daha karmaşık modeller daha fazla bellek ve hesaplama gerektirir, bu nedenle bellek sorunlarını veya yavaş eğitim sürelerini önlemek için daha küçük batch boyutları tercih edilebilir.
- iv. Öğrenme oranı: Öğrenme oranı, batch boyutundan etkilenen başka bir hiperparametredir. Daha büyük bir batch boyutu, modelin yerel minimumda takılıp kalmasını önlemek için tipik olarak daha büyük bir öğrenme oranı gerektirir.
- v. Eğitim süresi: Batch boyutu, modelin eğitim süresini de etkileyebilir. Daha büyük batch boyutları, daha hızlı eğitim sürelerine yol açabilir, ancak daha az doğru bir modelle sonuçlanabilir.
- vi. Donanım yapılandırması: GPU veya CPU çekirdeği sayısı gibi donanım yapılandırması da etkin bir şekilde kullanılabilecek toplu iş boyutunu etkileyebilir. Örneğin, daha fazla sayıda GPU, bellek tükenmeden daha büyük batch boyutlarının kullanılmasına izin verebilir.
- vii. Batch size değeri çoğunlukla 16 ile 512 arasındaki 2'nin katı olan bir değer olarak seçiliyor[46].
- viii. ESA'lar çoğu zaman batch değerine karşı duyarlıdır. Batch değerindeki ufak değişiklikler modelin başarımında çoğu zaman yüksek tesirler meydana getirebilir.
- ix. Batch işleminde, veri kümesi batch değeri olarak belirlenen değere göre parçalara ayrılmakta ve her iterasyonda modelin eğitimi bu parça üzerinden yapılmaktadır. Bununla birlikte bazı durumlarda veri kendi içinde gruplanmış olabilmektedir. Bu durum veri kümesi içinde korelasyon oluşturacak; bu veri kümesinden seçilecek test kümesinde yüksek başarımlı vermesini sağlayacak böylece ezberleme ("overfitting") olacaktır. Bunu önlemek için eğitim başlamadan veri kümesi parçalara ayrılmadan önce veri kümesi karıştırılmalıdır (shuffle). Batch boyutunun seçiminde verilerin rastgele seçilmesi önemlidir.

4.3.3 Öğrenme Oranı ve Momentum Katsayısı

Öğrenme hızını ve momentumu ayarlamak, derin sinir ağlarını optimize etmede önemli bir adımdır. Öğrenme hızı ve momentumu ayarlanırken aşağıdaki adımlar dikkate alınmalıdır:

- i. Makul bir aralıkla başlayın: Tipik olarak, öğrenme hızı ve momentum sırasıyla 0,1 ve 0,9 gibi küçük değerlere ayarlanır. Ancak kesin değerler, belirli göreve ve ağ mimarisine bağlıdır. Makul bir aralıkla başlamak ve sonuçlara göre ayarlamak iyi bir fikirdir.
- ii. Bir ızgara araması kullanın: Öğrenme oranını ve momentumu ayarlamanın bir yolu, bir ızgara araması kullanmaktır. Her hiperparametre için bir dizi olası değer belirtebilir ve ağı her bir değer kombinasyonu için eğitebilirsiniz. Daha sonra ağın performansını değerlendirebilir ve en iyi hiperparametre kombinasyonunu seçebilirsiniz.
- iii. Rastgele arama kullan: Başka bir seçenek de, belirli bir aralıktaki her hiperparametre için değerleri rastgele örneklediğiniz rastgele arama kullanmaktır. Bu yaklaşım, özellikle hiperparametre aralığı genişse, ızgara aramasından daha verimli olabilir.
- iv. Kaybı izleyin: Egzersiz sırasında, azaldığından emin olmak için kaybı izleyin. Kayıp azalmıyorsa, öğrenme hızını ve momentumu ayarlamak gerekebilir. Kayıp çok yavaş azalırsa, öğrenme oranını artırmanız gerekebilir. Kayıp salınır veya farklılaşırsa, öğrenme oranını düşürmeniz veya momentumu artırmanız gerekebilir.
- v. Öğrenme oranı çizelgelerini kullanın: Öğrenme oranı çizelgeleri, öğrenme oranını zamanla ayarlar ve genellikle eğitim ilerledikçe düşürür. Bu, kaybın dalgalanmasını veya farklılaşmasını önlemeye yardımcı olabilir.
- vi. Momentum çizelgelerini kullanın: Öğrenme hızı çizelgelerine benzer şekilde, momentum çizelgeleri momentumu zaman içinde ayarlar. Bu, ağın daha hızlı yakınsamasına ve salınımlardan kaçınmasına yardımcı olabilir. Popüler momentum çizelgeleri, kademeli azalmayı ve doğrusal bozulmayı içerir.
- vii. Uyarlanabilir yöntemleri kullanın: Adam, Adagrad ve RMSProp gibi, öğrenme hızını ve momentumu eğime göre ayarlayan birçok uyarlanabilir yöntem mevcuttur. Bu yöntemler,

özellikle büyük ağlar ve karmaşık görevler için hiperparametreleri manuel olarak ayarlamaktan daha verimli olabilir.

Öğrenme hızı ve momentum, derin öğrenme modellerini eğitmek için kullanılan birçok optimizasyon algoritmasında önemli hiperparametrelerdir. Öğrenme hızı ve momentumun özelliklerinden bazıları şunlardır:

- i. Öğrenme oranı: Öğrenme oranı, eğitim sırasında model parametrelerinin güncellendiği adım boyutunu belirler. Daha yüksek bir öğrenme oranı, modelin daha hızlı yakınsamasına neden olur, ancak optimum parametrelerin aşılmasına neden olabilir. Daha küçük bir öğrenme oranı, daha yavaş yakınsamaya neden olur, ancak modelin aşma olasılığı daha düşüktür. Öğrenme oranı tipik olarak deneme yanılma yoluyla veya öğrenme oranı programlarının kullanılmasıyla belirlenir.
- ii. Momentum: Momentum, eğitim sırasında modelin yakınsamasını hızlandırmaya yardımcı olan bir parametredir. Bunu, önceki parametre güncellemesinin bir kısmını mevcut güncellemeye ekleyerek yapar ve parametre güncelleme sürecini etkili bir şekilde yumuşatır. Bu, daha istikrarlı bir yakınsama ile sonuçlanır ve modelin yerel minimumlardan kaçınmasına yardımcı olabilir. Daha yüksek bir momentum değeri, gradyan biraz yön değiştirse bile modelin aynı yönde daha tutarlı hareket edeceği anlamına gelir. Daha düşük bir momentum değeri, modelin daha kolay yön değiştirmesini sağlayacaktır.

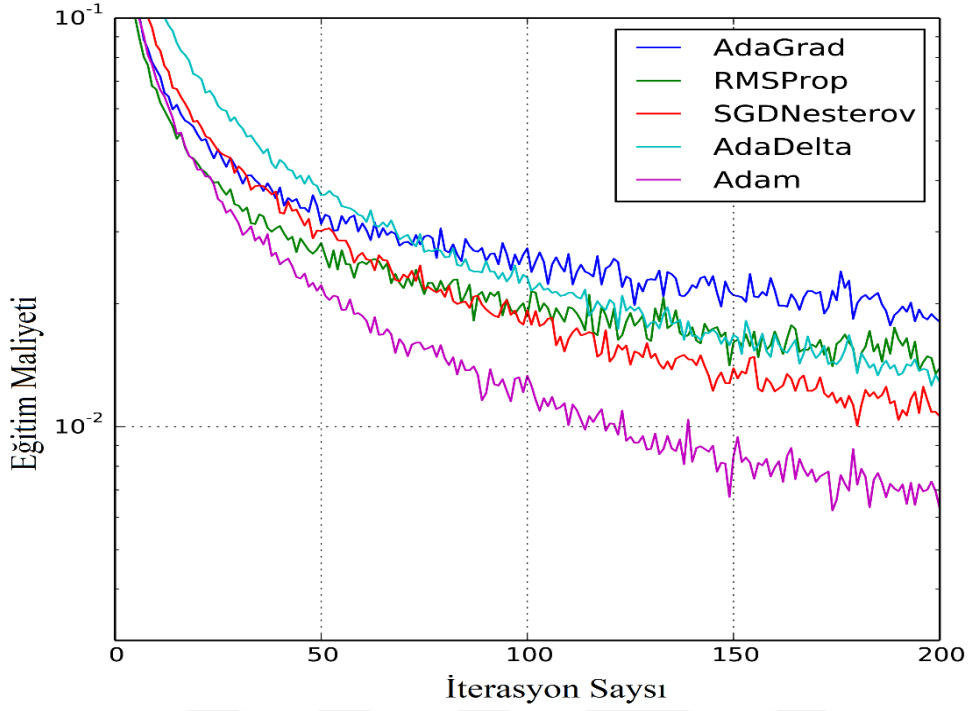
Özetle, öğrenme hızı ve momentum, derin öğrenme modellerinin eğitimini önemli ölçüde etkileyebilecek önemli hiperparametrelerdir. Bu hiperparametrelerin optimal değerleri, çözülmekte olan belirli probleme bağlıdır ve tipik olarak deneme yanılma veya otomatikleştirilmiş optimizasyon teknikleri yoluyla bulunur.

4.3.4 Optimizasyon Algoritmasının Seçimi

Optimizasyon “en iyi amaç kriterinin en iyi değerini veren kısıtlardaki değişkenlerin değerini bulmaktır” [47]. Eğitim optimizasyonu algoritmasının ayarlanması, daha iyi performans elde etmek için algoritmanın hiper parametrelerinin ayarlanmasını içerir. Eğitim optimizasyonu algoritmasını ayarlamak için izleyebileceğiniz bazı genel adımlar şunlardır:

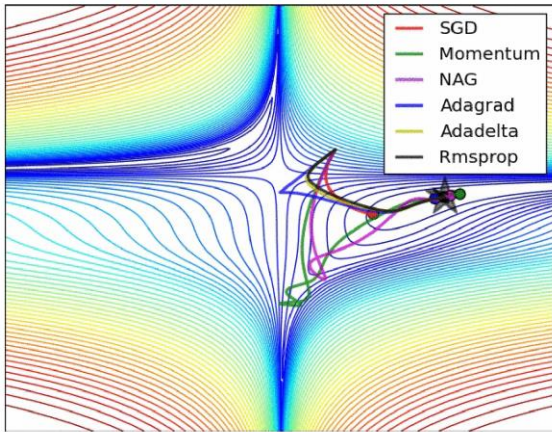
- i. Uygun bir deęerlendirme metrięi sein: Hiper parametreleri ayarlamadan nce, modelinizin performansını len uygun bir deęerlendirme metrięi tanımlamanız gerekir. Deęerlendirme ltlerine rnek olarak doęruluk, kesinlik, duyarlılık, F1 puanı ve ortalama karesel hata (MSE) verilebilir.
- ii. Ayarlanacak hiper parametreleri tanımlayın: ęrenme hızı, parti boyutu, gizli katman sayısı ve etkinleştirme işlevleri gibi her optimizasyon algoritması iin ayarlanacak birok hiper parametre vardır. Hangi hiper parametrelerin modelinizin performansını etkileyebileceęini belirlemeniz gerekir.
- iii. Hiper parametreler iin bir arama alanı tanımlayın: Ayarlanacak hiper parametreleri belirledikten sonra, her hiper parametre iin bir arama alanı tanımlamanız gerekir. Arama alanı, hiper parametrenin alabileceęi deęer aralıęını tanımlar.
- iv. Bir optimizasyon algoritması sein: Aralarından seim yapabileceęiniz, stochastic gradyan azaltma (stochastic gradient descent) [48], adagrad [49], adadelta [50], adam (Adaptive Moment Estimation) [51], adamax [52], RMS-Prop (Root Mean Square Propagation) yntemi [53][54], gibi optimizasyon algoritmalarıdır. Sorununuzun doęasına gre uygun bir optimizasyon algoritması semeniz gerekir.
- v. Hiper parametre ayarı iin bir yntem sein: Hiper parametre ayarı iin kılavuz arama, rastgele arama ve Bayes optimizasyonu gibi eşitli yntemler vardır. Izgara araması, tm hiper parametre uzayını kapsamlı bir şekilde araştıran, rastgele arama ise hiper parametre uzayını rasgele rnekleyen bir kaba kuvvet yntemidir. Bayes optimizasyonu, en iyi hiper parametreleri bulmak iin olasılıksal bir model kullanır.
- vi. Modelin performansını deęerlendirin: Hiper parametreleri ayarladıktan sonra, modelin performansını bir doęrulama setinde deęerlendirmeniz gerekir. Performans tatmin edici deęilse, 3. adıma geri dnmeniz ve hiper parametreler iin arama alanını ayarlamanız gerekebilir.
- vii. Nihai modeli test edin: Hiper parametreleri ayarladıktan ve doęrulama setindeki performansı deęerlendirdikten sonra, modelin yeni verilere iyi bir genelleme yaptıęından emin olmak iin son modeli ayrı bir test setinde test edebilirsiniz.

Genel olarak, eęitim optimizasyonu algoritmasını ayarlamak zaman alıcı bir sre olabilir, ancak makine ęrenimi grevlerinizde iyi performans elde etmek iin ok nemlidir.

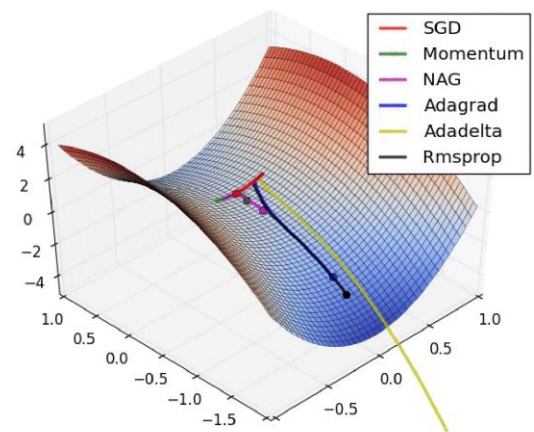


Şekil 4.33: MNIST seti üzerindeki çeşitli optimizasyon algoritmalarının çalışma zamanı grafiği [51].

Bazı optimizasyon algoritmalarının MNIST veri kümesi üzerindeki iterasyon sayısına bağlı eğitim maliyetinin grafiği yukarıdaki şekilde görüldüğü gibidir. Ayrıca optimizasyon algoritmalarının optimum değeri bulma yöntemi ve çalışma zamanı ile ilgili grafiği Şekil 4.33'te yer almaktadır.



(a) SGD optimization on loss surface contours



(b) SGD optimization on saddle point

Şekil 4.34: Optimizasyon algoritmalarının hız performans gösterimi [52].

Optimizasyon algoritmaları, olası çözümler kümesi arasından bir soruna en iyi çözümü bulmak için kullanılan matematiksel yöntemler kümesidir. Optimizasyon algoritmalarının özellikleri, kullanılan özel algoritmaya bağlı olarak değişebilir, ancak bazı ortak özellikler şunları içerir:

- i. Amaç fonksiyonu: Amaç fonksiyonu, optimize edilen problemi tanımlar. Bir veya daha fazla girdi değişkenini alır ve çözümün kalitesini temsil eden bir çıktı değeri üretir.
- ii. Kısıtlar: Kısıtlar, çözümün geçerli kabul edilmesi için yerine getirmesi gereken koşullardır. Bunlar, girdi değişkenleri veya aralarındaki ilişkiler üzerindeki sınırlamaları içerebilir.
- iii. Arama alanı: Arama alanı, algoritmanın dikkate alabileceği tüm olası çözümlerin kümesidir. Arama alanının boyutu ve karmaşıklığı, algoritmanın performansını etkileyebilir.
- iv. Optimizasyon yöntemi: Optimizasyon yöntemi, algoritmanın optimum çözümü nasıl aradığını belirler. Bazı yaygın optimizasyon yöntemleri arasında gradyan iniş ve genetik algoritmalar bulunur.
- v. Başlatma: Başlatma, algoritma tarafından kullanılan başlangıç noktasını veya ilk çözümü ifade eder. Bu, algoritma tarafından elde edilen nihai çözümü etkileyebilir.
- vi. Durdurma kriterleri: Durdurma kriterleri, algoritmanın ne zaman sonlanması gerektiğini belirleyen koşullardır. Bunlar, belirli bir doğruluk düzeyine ulaşmayı veya maksimum yineleme sayısını aşmayı içerebilir.
- vii. Performans metrikleri: Performans metrikleri, algoritma tarafından üretilen çözümün kalitesini değerlendirmek için kullanılır. Bunlar, doğruluk, kesinlik ve duyarlılık gibi ölçümleri içerebilir.
- viii. Ölçeklenebilirlik: Ölçeklenebilirlik, algoritmanın büyük veya karmaşık sorunları çözme yeteneğini ifade eder. Algoritmanın verimliliği ve hızı, ölçeklenebilirliğini etkileyebilir.

Genel olarak, optimizasyon algoritmaları, herhangi bir kısıtlamayı veya sınırlamayı yerine getirirken bir soruna en iyi çözümü bulmayı amaçlar. Bu algoritmaların özellikleri, optimal çözümlerin bulunmasında etkinlik ve verimliliklerinin belirlenmesinde önemli bir rol oynamaktadır.

4.3.5 Eğitim Tur (Epoch) Sayısı

Derin öğrenme problemlerinde model eğitime başlanırken veri kümesindeki tüm veriler aynı anda yani tek seferde eğitilmez. Veriler eşit sayıda küçük gruplara ayırarak tüm gruplar tek tek eğitime sokulur. İşte tüm gruplar eğitimi bitip tekrar modelin eğitim için başa dönmesine bir tur sayısı (epoch) denir. Tur sayısı ayarı, derin öğrenme modellerini eğitmenin önemli bir yönüdür. Tur sayısı, eğitim sırasında algoritmanın tüm veri kümesi üzerinde yineleme sayısıdır. Amaç, doğrulama verilerinde iyi bir genellemeye yol açacak en uygun tur sayısını bulmaktır.

Tur sayısını ayarlamak için bazı genel adımlar şunlardır:

- i. Verileri eğitim, doğrulama ve test kümelerine ayırın. Eğitim seti, modeli eğitmek için kullanılır, doğrulama seti, modelin eğitim sırasındaki performansını değerlendirmek için kullanılır ve test seti, modelin eğitimden sonraki performansını değerlendirmek için kullanılır.
- ii. Modeli sabit sayıda priyotlarla, örneğin 10 turla eğitin. Her turdan sonra doğrulama kümesinde modelin performansını değerlendirin. Her turdan sonra doğrulama doğruluğunu ve kaybını kaydedin.
- iii. Tur sayısı üzerinden doğrulama doğruluğunu ve kaybını çizin. Doğrulama kaybı artmaya başlarken doğrulama doğruluğunun düzleşmeye veya azalmaya başladığı bir nokta arayın. Bu, aşırı uyumun bir işaretidir ve bu noktada eğitimi durdurmalısınız.
- iv. Doğrulama doğruluğu ve kaybı gelişmeye devam ederse, tur sayısını artırın ve işlemi tekrarlayın. Doğrulama doğruluğu düzleşmeye veya azalmaya başlayana kadar bu işleme devam edin.
- v. En uygun tur sayısını bulduğunuzda, modeli bu tur sayısına sahip eğitim ve doğrulama kümeleri üzerinde eğitin. Modelin performansının tarafsız bir tahminini elde etmek için test setindeki modelin performansını değerlendirin.

Optimum tur sayısının belirli soruna ve veri kümesine bağlı olarak değişebileceğini unutmayın, bu nedenle en iyisini bulmak için farklı sayıda tur sayıları ile denemeler yapmak önemlidir.

4.3.6 Ağırlık (Weight) Başlangıç Değerlerinin Belirlenmesi

Bir sinir ağının ağırlıkları için uygun başlangıç değerlerinin seçilmesi, etkili bir model oluşturmada önemli bir adımdır. Zayıf başlatma, modelin yerel minimumda takılıp kalmasına veya hiç yakınsamamasına neden olabilir. Ağ ağırlığı başlatmayı ayarlamak için bazı ipuçları aşağıda yer almaktadır:

- i. Rastgele başlatma kullan: Ağırlıkları başlatmanın en yaygın yöntemi, değerleri normal bir dağılımdan rastgele örnekleme. Tüm ağırlıkları aynı değere ayarlamaktan veya aktivasyon işlevinin doyuma ulaşmasına neden olabilecek büyük değerlerle başlatmaktan kaçınmak önemlidir.
- ii. Farklı dağılımlarla denemeler yapın: Normal dağılıma ek olarak, ağırlıkları başlatmak için kullanılabilecek tek biçimli, kesik normal ve Xavier başlatma gibi başka dağılımlar da vardır. Sorunuza en uygun olanı görmek için farklı dağıtımları deneyin.
- iii. Önceden eğitilmiş ağırlıkları kullanın: Benzer veriler veya görevler üzerinde eğitilmiş önceden eğitilmiş ağırlıklara erişiminiz varsa, bunları modeliniz için bir başlangıç noktası olarak kullanabilirsiniz. Bu, eğitimi hızlandırmaya ve performansı artırmaya yardımcı olabilir.
- iv. Ağırlık ölçeğini ayarlayın: Ağırlıkları bir faktöre göre ölçeklendirmek, yakınsamayı hızlandırmaya ve doygunluğu önlemeye yardımcı olabilir. Bu, ağırlıkları sabit bir faktörle çarparak veya ağırlıkları hâlihazırda ölçekleyen bir başlatma yöntemi kullanılarak yapılabilir.
- v. Performansı izleyin: Eğitim sırasında modelin performansını takip edin ve model öğrenmiyorsa veya eğitim verilerini gereğinden fazla uyduruyorsa ağırlık başlatmayı ayarlayın.

Genel olarak, doğru ağırlık başlatma tekniğini seçmek, sorunun özelliklerine ve model mimarisine bağlıdır. Optimum başlatma yöntemini bulmak biraz deneme yanılma gerektirebilir.

4.3.7 Aktivasyon Fonksiyonu

Aktivasyon fonksiyonu, modele doğrusal olmayanlık (non-linearity) getiren bir sinir ağının önemli bir bileşenidir ve buda modelin öğrenme sırasında önemli bir görev almaktadır. Bir nöronun

çıktısının etkinleştirilip etkinleştirilmeyeceğini belirlemek için kullanılır. Aktivasyon fonksiyonun seçimi, sinir ağının performansı üzerinde çok büyük bir tesire sahip olabilir ve bu nedenle, onu dikkatli bir şekilde ayarlamak çok önemlidir. Nöron aktivasyon fonksiyonu ayarlamak için bazı adımlar şunlardır:

- i. Doğru aktivasyon fonksiyonu seçme: Aktivasyon fonksiyonlarından bazıları; adım fonksiyonu, birim fonksiyonu, tanjant hiperbolik fonksiyonu ($\tanh$), sigmoid fonksiyonu, doğrultulmuş doğrusal birim (ReLU) fonksiyonu, softmax fonksiyonu ve son zamanlarda Google tarafından geliştirilen swish fonksiyonu v.b. fonksiyonlarıdır. Halen sinir ağları için en kullanışlı ve en popüler olan aktivasyon fonksiyonu, ilk önce sınırlı Boltzmann makineleri için önerilen[55] ve daha sonra sinir ağları için başarılı [56] bir şekilde kullanılan doğrultulmuş doğrusal birim (ReLU) fonksiyonudur. Bu fonksiyonların her birinin avantajları ve dezavantajları vardır ve bunları anlamak, sinir ağınız için doğru aktivasyon fonksiyonunu seçmenize yardımcı olabilir.
- ii. Probleminiz için aktivasyon fonksiyonunu seçin: Çözmeye çalıştığınız probleme bağlı olarak, belirli aktivasyon fonksiyonu diğerlerinden daha iyi çalışabilir. Örneğin, sigmoid fonksiyonu genellikle ikili sınıflandırma görevlerinde kullanılırken softmax fonksiyonu, çok sınıflı sınıflandırma problemlerinde yaygın olarak kullanılır.
- iii. Farklı aktivasyon fonksiyonu denemeleri yapın: Farklı aktivasyon fonksiyonu kullanmayı deneyin ve probleminizdeki performanslarını karşılaştırın. Sinir ağınızın performansını farklı aktivasyon fonksiyonu değerlendirmek için bir doğrulama seti kullanabilirsiniz.
- iv. Hiper parametreleri ayarlayın: Aktivasyon fonksiyonuna bağlı olarak, öğrenme hızı, düzenleme ve başlangıç değerlerini ayarlama gibi hiper parametreleri ayarlamanız gerekebilir.
- v. Aşırı uyumdan (ağın ezberlemesinden) kaçınma: Bazı aktivasyon fonksiyonu, özellikle modele çok fazla doğrusalsızlık getirirlerse, aşırı uyuma yol açabilir. Seyreletme gibi düzenleme teknikleri, aşırı uydurmayı önlemeye yardımcı olabilir.

- vi. Eğitim sürecini izleme: Eğitim sırasında modelinizin kaybını ve doğruluğunu takip edin. Modelin gelişmediğini fark ederseniz, aktivasyon fonksiyonu veya hiper parametreleri ayarlamanız gerekebilir.

Genel olarak, bir sinir ağının aktivasyon fonksiyonu ayarlamak, dikkatli deneyler ve izleme gerektirir. Bu adımları izleyerek göreviniz için doğru aktivasyon fonksiyonunu seçebilir ve sinir ağınızın performansını iyileştirebilirsiniz.

4.3.7.1 Sık kullanılan aktivasyon fonksiyonları

Aktivasyon fonksiyonları, bir katmana gelen net bilgiyi hesaplayarak nöronun bu girdiye karşılık üreteceği çıktıyı hesaplayıp bir sonraki katmana iletmek için kullanılır (çıkış katmanına kadar ve çıkış katmanı da dahil olmak üzere). Genellikle aktivasyon fonksiyonları doğrusal olmayan bir fonksiyon seçilir. Aktivasyon fonksiyonu seçilirken türevinin kolay hesaplanabilir olmasına dikkat edilir. Özellikle geri beslemeli ağlarda hatanın dağılımı türev yardımı ile hesaplandığından türevi kolay hesaplanan bir fonksiyon seçilir. Birçok aktivasyon fonksiyonu vardır ve bu fonksiyonlardan en çok kullanılanları aşağıda verilmiştir.

Sigmoid Veya Lojistik

Elde ettiğimiz çıktı değerlerini bir olasılık değeri olarak yorumlamak için kullanılan aktivasyon fonksiyonlarından birisi sigmoid fonksiyonudur. Sigmoid fonksiyonu genellikle ikili sınıflandırma problemlerinde kullanılır. Bu fonksiyon girdi değerlerini 0 ile 1 arasında bir çıktıya dönüştürür ve bu çıktı olasılık olarak yorumlanabilir. Sigmoid fonksiyonunun formülü denklem (4.2)'de verilmektedir.

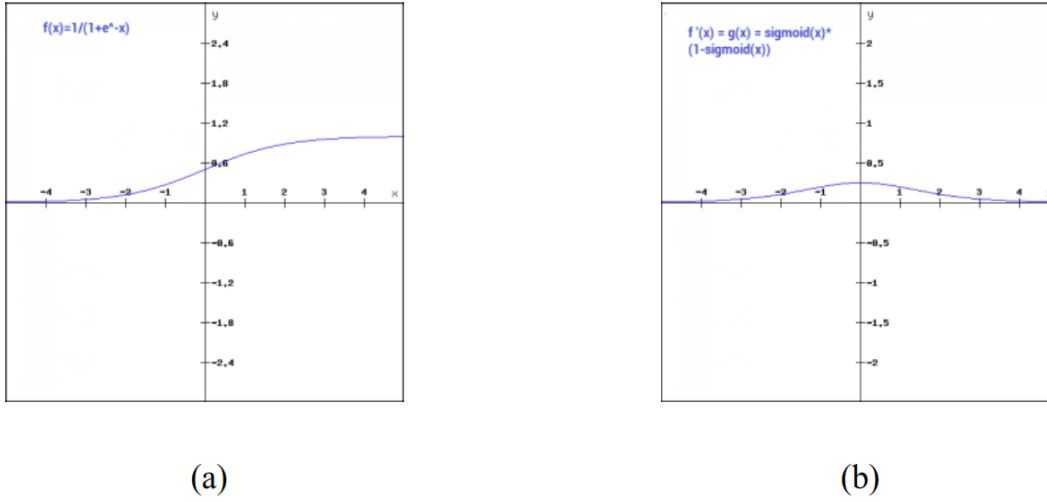
$$f(x) = \frac{1}{1+e^{-x}} \quad (4.2)$$

Burada x girdi değerini, e doğal logaritmanın tabanını (euler sayısını), $f(x)$ ise sigmoid fonksiyonunun çıktı değerini ifade eder.

Sigmoid fonksiyonunun yapısından kaynaklanan birkaç özelliği vardır. İlk olarak, türevi her zaman hesaplanabilir, bu nedenle geriye doğru yayılım sırasında kolayca hesaplanabilir. İkinci olarak, sigmoid fonksiyonu 0 ile 1 arasında sınırlıdır, bu nedenle olasılık değerlerinin modellenmesi için kullanılabilir. Üçüncü olarak, sigmoid fonksiyonu simetrik bir yapıya sahiptir ve orta noktası

$x = 0.5$ 'tir. Bu, sigmoid fonksiyonunun girdisi 0'a yakın olduğunda, çıktısının da 0.5'e yakın olacağı anlamına gelir.

Sigmoid fonksiyonunun bir dezavantajı, girdi değeri çok büyük veya çok küçük olduğunda doyuma ulaşabilmesidir. Bu duruma gradyan kaçış noktası olarak ifade edilmektedir ve Şekil 4.35'de bu durum görülmektedir. Gradyanın her iki tarafta da başlangıca çok yatay olması ve bu nedenle öğrenme sürecini yavaşlatmaktadır. Buda, sigmoid fonksiyonunun türevinin çok küçük olması ve eğitim sırasında sorunlar yaratması anlamına gelir. Gradyan kaçış noktasına takılmamak için ağırlıkları başlatırken büyük değerler ile başlamak mantıklı olacaktır.



Şekil 4.35: (a) Sigmoid fonksiyonu. (b) Sigmoid fonksiyonu gradyanı.

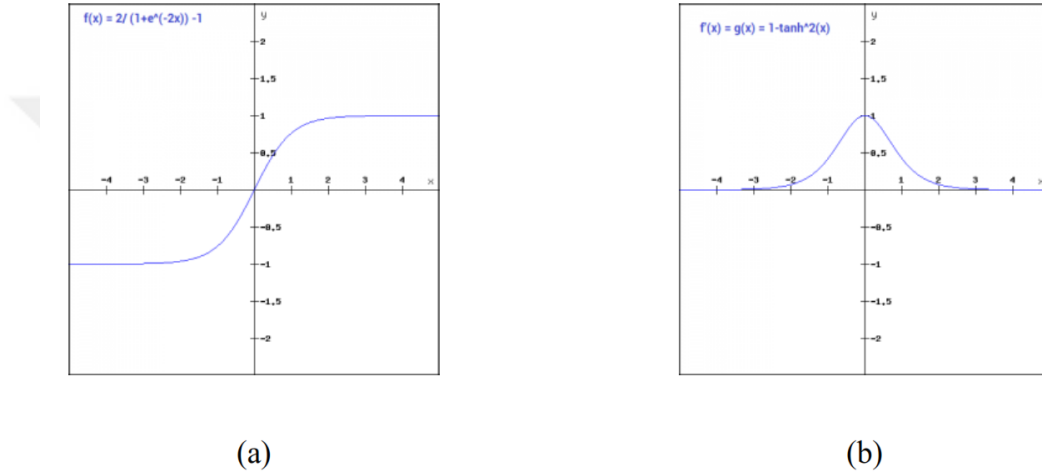
Tanjant Hiperbolik (Tanh)

Tanh, bir yapay sinir ağı için yaygın olarak kullanılan bir aktivasyon fonksiyonudur. *Tanh*, hiperbolik tanjantın kısaltmasıdır ve *Tanh* denklem (4.3)'de verilmiştir.

$$\text{Tanh}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (4.3)$$

Burada, x herhangi bir gerçektek sayı olabilir. *Tanh* fonksiyonu, sigmoid fonksiyonuna benzer, ancak değerleri -1 ile 1 arasındadır. Bu, bir sinir hücresinin çıkışının her zaman pozitif veya her zaman negatif olmasını önler. *Tanh* fonksiyonu ayrıca, girdinin sıfıra yakın olduğu durumlarda sigmoid fonksiyonundan daha fazla duyarlıdır. Bu, aktivasyon fonksiyonunun, ağı daha hızlı öğrenmesine yardımcı olabileceği anlamına gelir. Sigmoid fonksiyonunda olduğu gibi ne yazık ki bu fonksiyonda da gradyan kaçış noktası problemi yer almaktadır.

Tanh fonksiyonu aynı zamanda bir simetriye sahiptir. Yani, $x = 0$ 'da *tanh* fonksiyonunun değeri 0'dır (bkz. Şekil 4.36). Bu, sinir ağındaki simetriyi koruyarak, eğitim verilerine daha iyi uyum sağlayabilir. Son olarak, *tanh* fonksiyonu, sigmoid fonksiyonuna kıyasla daha büyük bir gradyan üretir. Bu, geriye yayılım algoritması ile ağı eğitiminde daha hızlı ve daha iyi sonuçlar elde edilmesine yardımcı olabilir. Tüm bu nedenlerden dolayı, *tanh* fonksiyonu, yapay sinir ağlarında sıklıkla kullanılan bir aktivasyon fonksiyonudur.



Şekil 4.36: (a) *Tanh* fonksiyonu. (b) *Tanh* fonksiyonu gradyanı.

ReLU (Rectified Linear Unit)

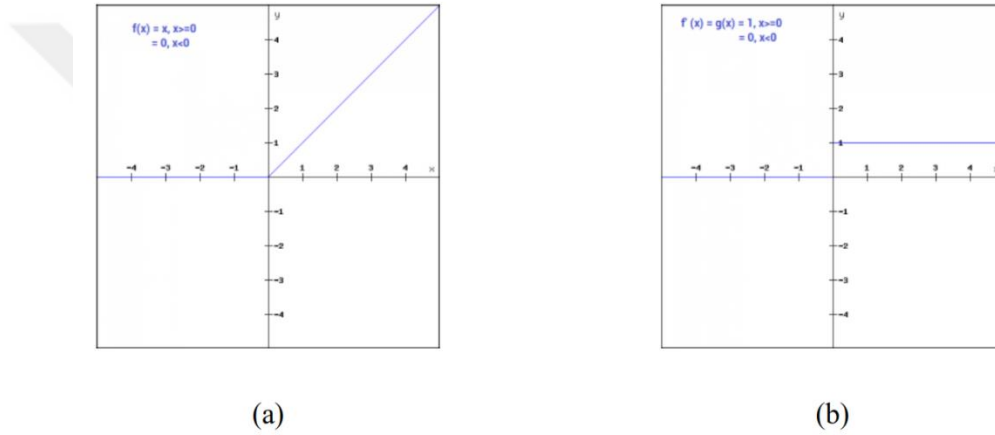
ReLU (Rectified Linear Unit) aktivasyon fonksiyonu, yapay sinir ağlarındaki en yaygın kullanılan aktivasyon fonksiyonlarından biridir. Bu fonksiyon, bir nöronun çıktı değerini hesaplamak için kullanılır ve ağı non-lineerlik kazanmasına yardımcı olur. ReLU fonksiyonu, Şekil 4.37'de görüldüğü üzere girdi olarak aldığı değerin sıfırdan küçük olması durumunda sıfırı, sıfırdan büyük olması durumunda ise girdi değerini doğrudan çıkış olarak verir. Matematiksel olarak ifade denklem (4.4)'te yer almaktadır.

$$f(x) = \max(0, x) \quad (4.4)$$

Burada x , bir nöronun toplam girdi değeri (ağırlıklı toplamı ve bias değeri ile birlikte) ve $f(x)$ ise çıktı değeridir. ReLU'nun birçok avantajı vardır. Öncelikle, hesaplama açısından çok ucuzdur, çünkü sadece bir karşılaştırma işlemi gerektirir. Ayrıca, sigmoid veya *tanh* gibi diğer aktivasyon

fonksiyonlarına göre, sıfırdan farklı değerler üretmesi nedeniyle gradiyanlar üzerinde daha iyi performans gösterir. Bu, daha hızlı öğrenmeyi ve daha iyi sonuçlar elde etmeyi sağlar.

Ancak, ReLU fonksiyonunun bazı dezavantajları da vardır. Özellikle, girdi değeri negatif olduğunda türevi sıfır olur ve bu durumda gradiyanın da sıfır olması nedeniyle geri yayılım algoritmasında sorunlar yaşanabilir. Bu, "ölü nöron" problemini doğurabilir, yani bir nöronun hiçbir zaman aktif hale gelmemesi ve dolayısıyla öğrenme sürecine katkıda bulunmaması. Bu sorunun çözümü için, LeakyReLU gibi değişik varyasyonlar kullanılabilir.



Şekil 4.37: (a) ReLU fonksiyonu. (b) ReLU fonksiyonu gradiyanı.

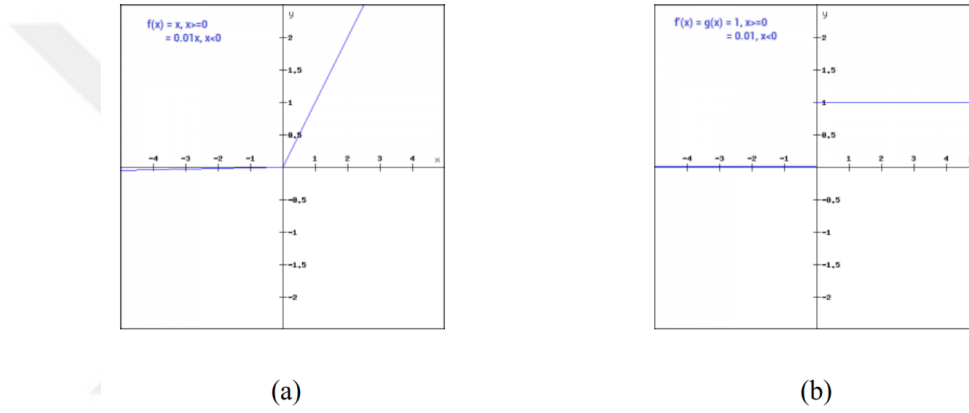
Sızıntı (Leaky) ReLU

Sızıntı ReLU fonksiyonu, derin öğrenmede en çok kullanılan aktivasyon fonksiyonlarından biri olan standart ReLU aktivasyon fonksiyonunun bir varyasyonudur. Standart ReLU fonksiyonu denklem(4.3)'te tanımlanmıştır. ReLU fonksiyonu basit ve hesaplama açısından verimlidir, ancak "ölü nöron" olarak adlandırılan bir sorunla karşı karşıyadır. Bu sorunda, bazı nöronlar eğitim sırasında hiçbir zaman pozitif aralığa ulaşmayan girdilerle etkisiz hale gelebilir ve hiçbir zaman kurtulamazlar. Ölü nöron sorununu çözmek için Sızıntı ReLU fonksiyonu geliştirilmiştir. Sızıntı ReLU fonksiyonu denklem(4.5)'te tanımlanmıştır.

$$f(x) = \begin{cases} x, & x > 0 \\ \alpha x, & x \leq 0 \end{cases} \quad (4.5)$$

burada α küçük bir sabit olup genellikle 0.01 ile 0.3 arasında bir değer olarak ayarlanır ve α küçük pozitif bir değer olarak ayarlandığında, negatif girdiler için Şekil 4.38'de görüldüğü üzere fonksiyonun eğimi sıfır olmadığından, ölü nöron sorununu önlemeye yardımcı olur.

Sızıntı ReLU fonksiyonunun standart ReLU fonksiyonuna göre birkaç avantajı vardır. İlk olarak, ölü nöron sorununu önlemeye yardımcı olur, bu da derin sinir ağlarının yakınsama ve performansını artırabilir. İkinci olarak, çıktıda negatif değerlerin kaybolmasını önler ve gradient kaybına engel olur. Üçüncüsü, hesaplama açısından verimlidir ve kolayca uygulanabilir.



Şekil 4.38: (a) Sızıntı ReLU fonksiyonu. (b) Sızıntı ReLU fonksiyonu gradyanı.

Genel olarak, Sızıntı ReLU fonksiyonu, derin sinir ağlarının performansını iyileştirmeye yardımcı olan faydalı bir aktivasyon fonksiyonudur, özellikle toplu normalleştirme ve seyreltme gibi diğer tekniklerle birlikte kullanıldığında.

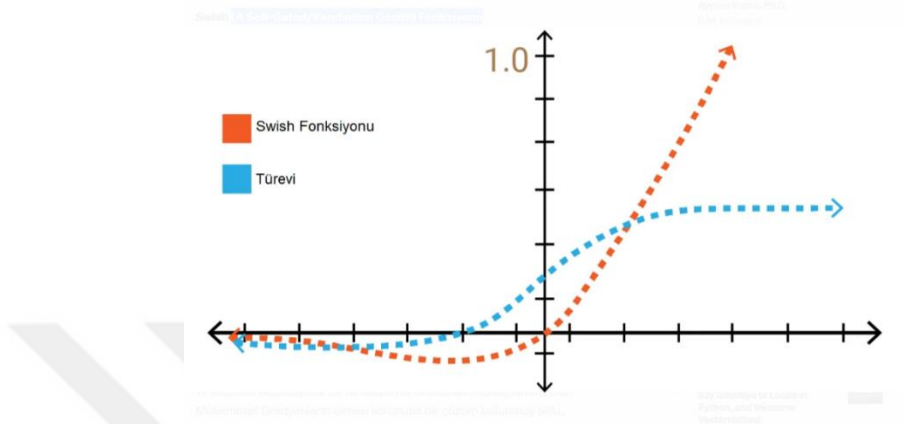
Swish (A self-Gated/Kendinden Geçitli) Fonksiyonu

Swish aktivasyonu, 2017 yılında Google'daki araştırmacılar tarafından önerilen, derin sinir ağlarında kullanılan bir aktivasyon fonksiyonu türüdür. Swish aktivasyon fonksiyonu denklem (4.6) te tanımlandığı gibidir.

$$f(x) = \frac{2x}{1 + e^{-\beta x}} \quad (4.6)$$

burada x , aktivasyon fonksiyonunun girdisini, $f(x)$ çıktıyı ve beta (β) ise eğitilebilir bir parametreyi ifade etmektedir. Denklemden eğer $\beta=0$ alınırsa fonksiyon $f(x)=x$ olur ki buda bize birinci açıortay doğrusu olan birim doğrusal fonksiyonu olur. Diğer taraftan eğer β çok büyük bir değer alırsa

fonksiyon ReLU fonksiyonuna yakınsar. Bu sebeple standart swish fonksiyonunda $\beta=1$ seçilir. Bu sayede hem gradyan iniş yumuşatılmış olunur ve nöron ölmesi problemini de çözülmüş olunur.



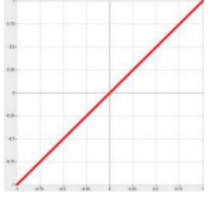
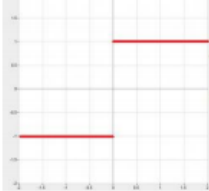
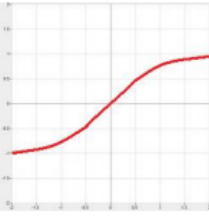
Şekil 4.39: Swish fonksiyonu ve gradyanı.

Swish aktivasyon fonksiyonu, tüm negatif değerleri sıfıra ayarlayan ve tüm pozitif değerleri değiştirmeden geçiren popüler ReLU (Düzeltilmiş Doğrusal Birim) aktivasyon fonksiyonuna benzer (bkz. Şekil 4.39). Ancak, Swish aktivasyon fonksiyonu, negatif değerlerin sıfır olarak ayarlanması yerine sorunsuz bir şekilde ölçeklenmesine izin veren düzgün bir fonksiyondur. Bu düzgünlük, Swish etkinleştirme işlevinin çıkışı (0, 1) aralığıyla sınırlayan x giriş değerine sigmoid fonksiyonunun uygulanmasıyla elde edilir.

Swish aktivasyon fonksiyonunun bir avantajı, ReLU aktivasyon fonksiyonundan daha iyi doğrusal olmama sağlayarak derin sinir ağlarının performansını artırabilmesidir. Bu geliştirilmiş doğrusal olmama, derin ağlarda nöronların doygunluğunu önlemeye yardımcı olabilen Swish aktivasyon fonksiyonunun düzgünlüğünden kaynaklanmaktadır.

Ayrıca, Swish aktivasyon işlevi çoğu derin öğrenme problemlerine kolayca uygulanabilir ve ReLU, sigmoid ve tanh gibi diğer popüler aktivasyon işlevlerinden daha iyi performans gösterdiği çeşitli kıyaslama veri setlerinde gösterilmiştir.

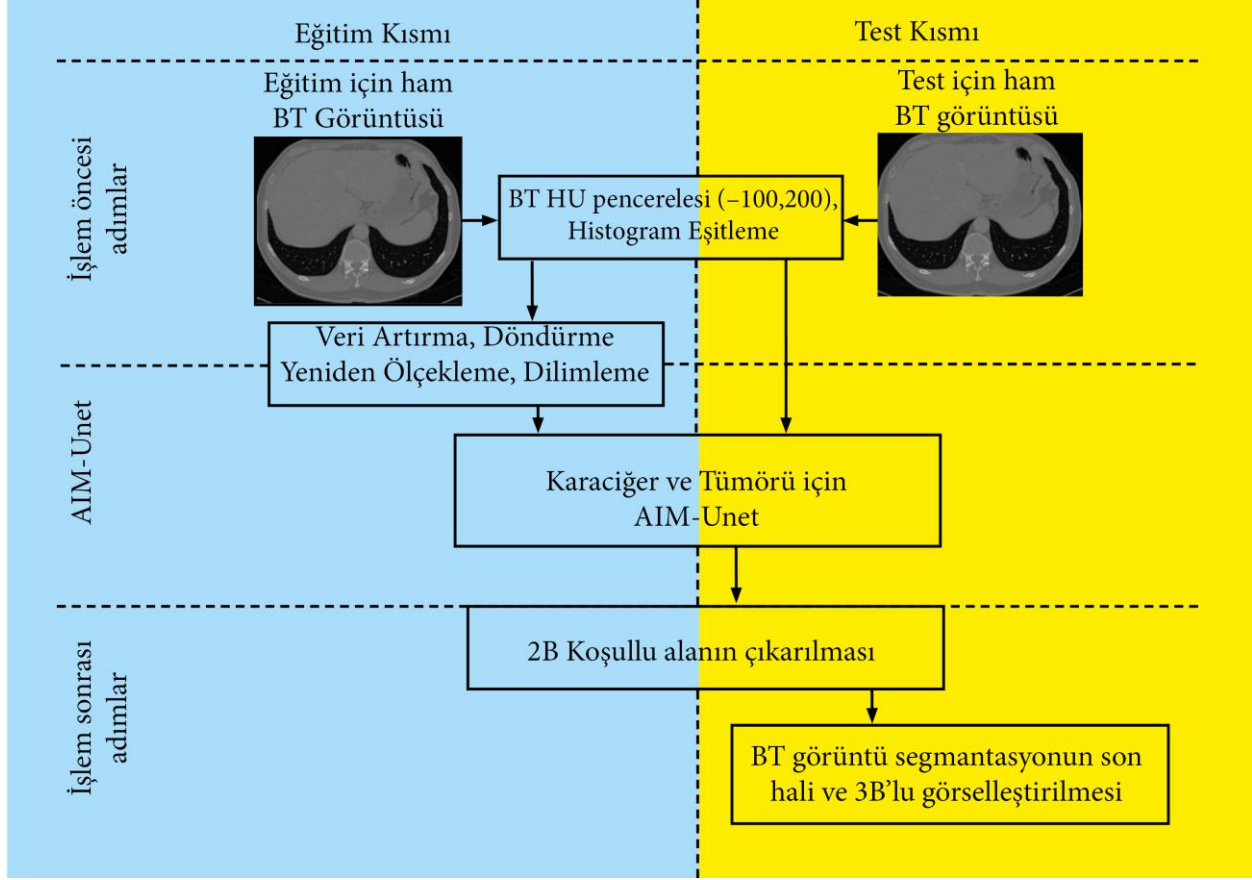
Bu fonksiyonlar ile birlikte kullanılan diğer aktivasyon fonksiyonlarının grafikleri Şekil 4.40'ta yer almaktadır.

Doğrusal (Lineer) Aktivasyon Fonksiyonu		$F(Net) = A * Net$ (A sabit bir sayı)	Doğrusal problemler çözmek amacıyla aktivasyon fonksiyonu doğrusal bir fonksiyon olarak seçilebilir. Toplama fonksiyonundan çıkan sonuç, belli bir katsayı ile çarpılarak hücrenin çıktısı olarak hesaplanır.
Adım (Step) Aktivasyon Fonksiyonu		$F(Net) = \begin{cases} 1 & \text{if } Net > \text{Eşik Değer} \\ 0 & \text{if } Net \leq \text{Eşik Değer} \end{cases}$	Gelen Net girdinin belirlenen bir eşik değerin altında veya üstünde olmasına göre hücrenin çıktısı 1 veya 0 değerini alır.
Sigmoid Aktivasyon Fonksiyonu		$F(Net) = \frac{1}{1 + e^{-Net}}$	Sigmoid aktivasyon fonksiyonu sürekli ve türevi alınabilir bir fonksiyondur. Doğrusal olmayışı dolayısıyla yapay sinir ağı uygulamalarında en sık kullanılan fonksiyondur. Bu fonksiyon girdi değerlerinin her biri için 0 ile 1 arasında bir değer üretir.
Tanjant Hiperbolik Aktivasyon Fonksiyonu		$F(Net) = \frac{e^{Net} + e^{-Net}}{e^{Net} - e^{-Net}}$	Tanjant hiperbolik fonksiyonu, sigmoid fonksiyonuna benzer bir fonksiyondur. Sigmoid fonksiyonunda çıkış değerleri 0 ile 1 arasında değişirken hiperbolik tanjant fonksiyonunun çıkış değerleri -1 ile 1 arasında değişmektedir.
Eşik Değer Fonksiyonu		$F(Net) = \begin{cases} 0 & \text{if } Net \leq 0 \\ Net & \text{if } 0 < Net < 1 \\ 1 & \text{if } Net \geq 1 \end{cases}$	Gelen bilgilerin 0 dan küçük-eşit olduğunda 0 çıktısı, 1 den büyük-eşit olduğunda 1 çıktısı, 0 ile 1 arasında olduğunda ise yine kendisini veren çıktılar üretilebilir.
Sinüs Aktivasyon Fonksiyonu		$F(Net) = \sin(Net)$	Öğrenilmesi düşünülen olayların sinüs fonksiyonuna uygun dağılım gösterdiği durumlarda kullanılır.

Şekil 4.40: Aktivasyon fonksiyonları.

5. MATERYAL VE YÖNTEM

Bu çalışmada, abdominal BT görüntülerinde karaciğer ve tümörün otomatik bölütlenmesi için Unet ve Inception tabanlı bir model önerilmiştir. Blok diyagram, önerilen modelin metodolojisini göstermektedir ve üç aşamada özetlenmiştir: veri hazırlama, eğitim-test etme ve segmentasyon (Şekil 5.1).

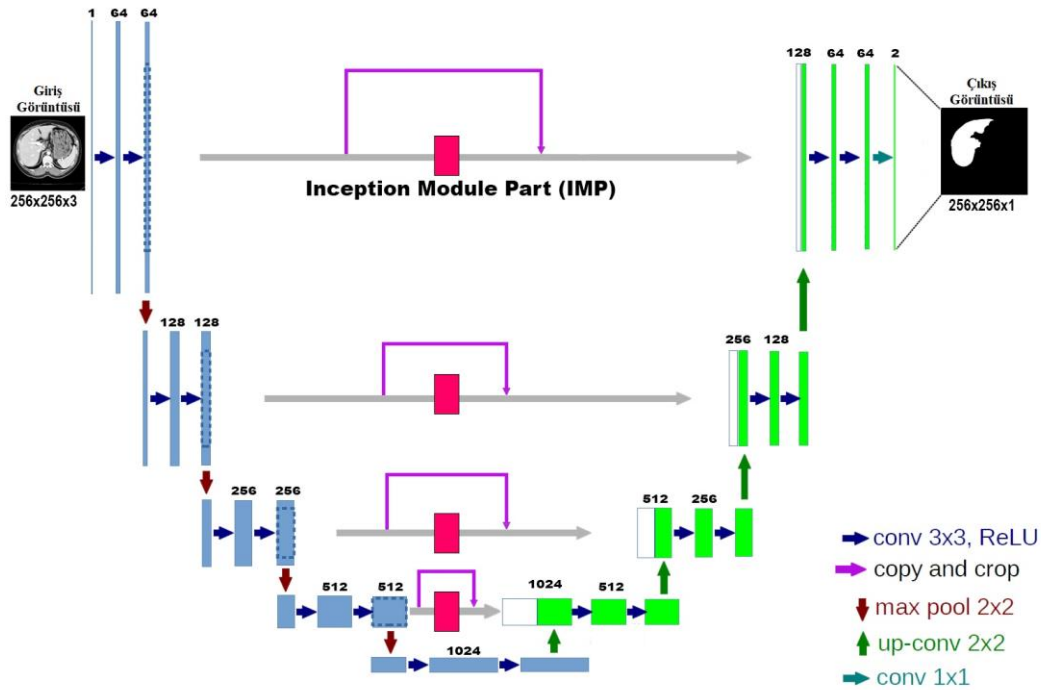


Şekil 5.1: AIM-Unet modelinin şematik gösterimi.

5.1 DERİN ÖĞRENME MODELİ GELİŞTİRME

Derin CNN'ler, son yıllarda tıbbi görüntü sınıflandırmasında iyi sonuçlar elde etti. Derin CNN'lerin performansını iyileştirmenin en basit yolu, derinliklerini ve genişliklerini artırmaktır [57]. Szegedy ve ark. her aşamanın genişliğini artıran Inception adlı derin bir CNN mimarisi önerdi [58]. Inception-Net'in birden çok sürümü vardır. Bunlardan Inception-v2 [59], Inception-v3 [60]-[61] ve Inception-v4 [62], sürümleri sınıflandırma görevlerinde yaygın olarak

kullanılmaktadır. He ve ark. tarafından tanıtılan artık bağlantılar, çok derin ağırları eğitmeyi kolaylaştırmıştır [63]. Her iki fikri birleştiren Inception-ResNet, yapılan uygulamalarda önceki ağlardan daha iyi performans göstermiştir. Ancak medical görüntü segmentasyonunda U-net en çok tercih edilen modellerdendir. Model ismini şeklinden de görüleceği gibi “U” harfine benzemesinden almıştır. Çok tercih edilmesinin sebebi başarılı sonuçlar vermesinin yanında eğitim süresinin de kısa olmasıdır. Orijinal U-Net modeli, biyomedikal görüntü segmentasyonu için geliştirilmiş başarılı bir evrişim ağıdır. U-net temelde kodlayıcı/küçülme yolu ve kod çözücü/genişleme yolu olarak iki bölümden oluşur. U-net’in de farklı versiyonları araştırmacılar tarafından geliştirilmiştir. Bu versiyonlardan bir kaçı Recurent Residual-Unet[64], Attention-Unet[65] ve Attention Residual-Unet’tir[66]. Karaciğer segmentasyonu için önerdiğimiz model AIM-Unet Şekil 5.2’de yer almaktadır. Bu model U-net modeli ile Inception modelinin bir modülünün birleşiminden oluşmaktadır. Modelimizde toplam 188 katman yer almakta olup toplam parametre sayısı 41,695,169 adettir. Bu parametrelerin 41,672,129 adedi eğitilebilir, 23,040 tanesi eğitilemez (Non-trainable)’dir. Önerilen modelde, katmanlardaki girdi, çıktı boyutları ve filtre sayıları Tablo 5.1’yer almaktadır.



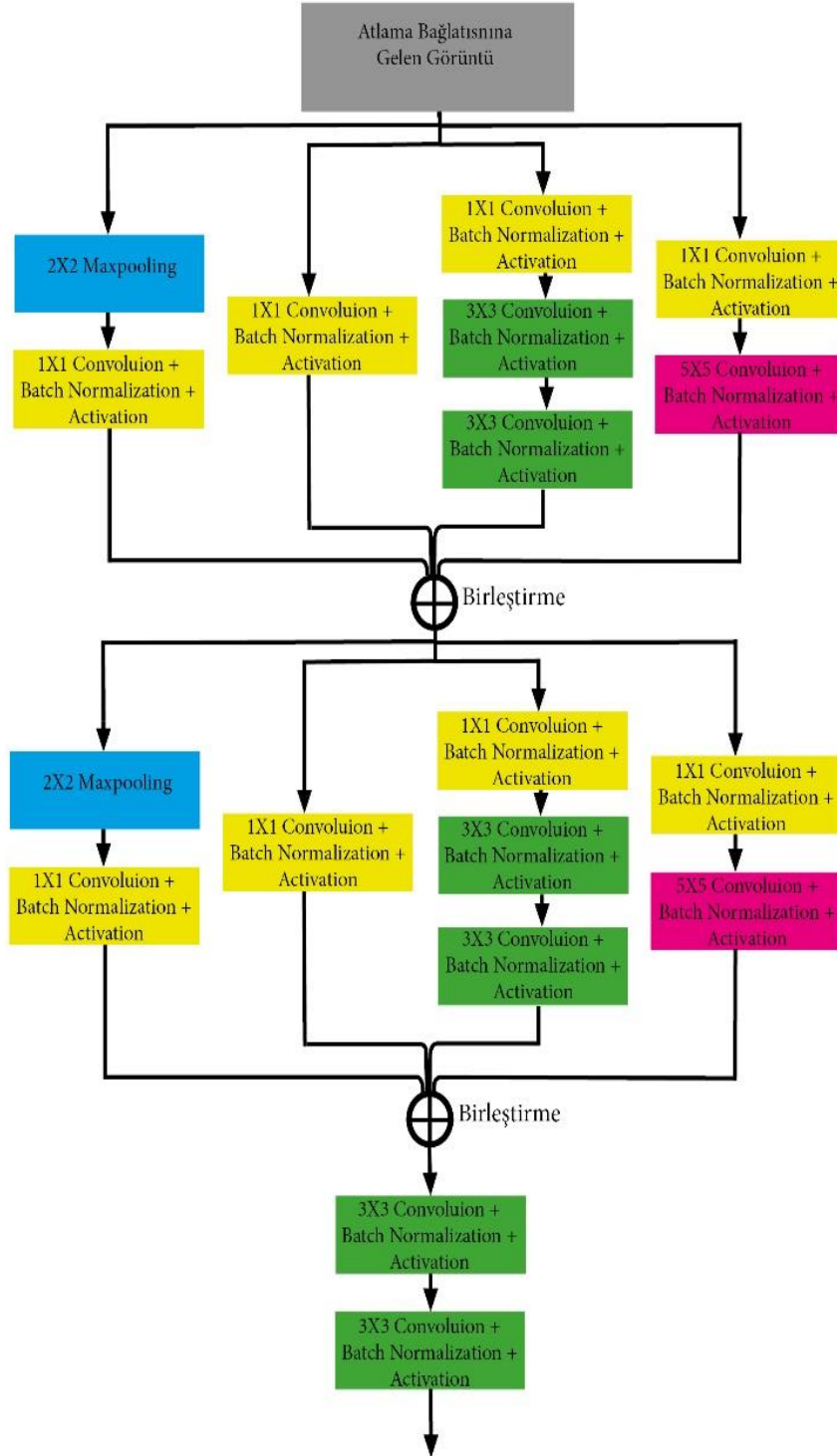
Şekil 5.2: AIM-Unet mimarisi.

Tablo 5.1: AIM-Unet için giriş,çıkış, filtre sayısı ve filtre boyutları

Kodlayıcı Bölümü					Inception Module Bölümü					Kod çözücü Bölümü				
Block	Giriş	Katman(filter boyutu)	Çıkış	filtreler	Giriş	Katman(filt er boyutu)	Çıkış	Giriş	filtreler	Block	Giriş	Katman(filter boyutu)	Çıkış	filtreler
Kodlayıcı 1	(256,256)	Conv2D(3,3)	(128,128)	64	Atlama Bağıntısı 1	(256,256)	IMP 1	(256,256)	128	K. Çözücü 1	(128,128)	Conv2D(3,3)	(256,256)	64
		Conv2D(3,3)										Conv2D(3,3)		
Kodlayıcı 2	(128,128)	Conv2D(3,3)	(64,64)	128	Atlama Bağıntısı 2	(128,128)	IMP 2	(128,128)	256	K. Çözücü 2	(64,64)	Conv2D(3,3)	(128,128)	128
		Conv2D(3,3)										Conv2D(3,3)		
Kodlayıcı 3	(64,64)	Conv2D(3,3)	(32,32)	256	Atlama Bağıntısı 3	(64,64)	IMP 3	(64,64)	512	K. Çözücü 3	(32,32)	Conv2D(3,3)	(64,64)	256
		Conv2D(3,3)										Conv2D(3,3)		
Kodlayıcı 4	(32,32)	Conv2D(3,3)	(16,16)	512	Atlama Bağıntısı 4	(32,32)	IMP4	(32,32)	1024	K. Çözücü 4	(16,16)	Conv2D(3,3)	(32,32)	512
		Conv2D(3,3)										Conv2D(3,3)		
Taban	(16,16)	Conv2D(3,3)	(16,16)	1024						Taban	(16,16)	Conv2D(3,3)	(16,16)	1024

5.2 INCEPTION MODULE PART (IMP)

Bu çalışmada karaciğer segmentasyonu için geliştirdiğimiz model Unet modeline inception modelinin konvolüsyon ve havuzlama katmanını içeren modülü Şekil 5.3’de gösterildiği gibi eklenmiştir. Bu modül atlama bağlantısına eklenmiştir. Eklediğimiz modül giriş verisini 4 farklı kolda işleme tutmaktadır. Bir kolunda sadece 1x1 konvolüsyon işlemi yapılırken, ikinci kolunda önce 1x1 konvolüsyon işlemi, sonrasında 3x3 konvolüsyon işlemi gerçekleştirilmektedir. Üçüncü kolda yine önce 1x1 konvolüsyon işlemi daha sonrasında iki defa 3x3 konvolüsyon işlemi gerçekleştirilmektedir. Son kolda ise önce maksimum havuzlama işlemi yapıp ardından 1x1 konvolüsyon işlemi gerçekleştirilmektedir. Bu dört kol işlemi gerçekleştirildikten sonra çıktıların birleşme işlemi uygulanır. Bu yapılan işlemler bir kez daha tekrarlanır. Ardından iki defa 3x3 konvolüsyon ve aktivasyon işlemlerine tabi tutulur. Bu işlemler Şekil 5.3’te detaylı bir şekilde gösterilmiştir.



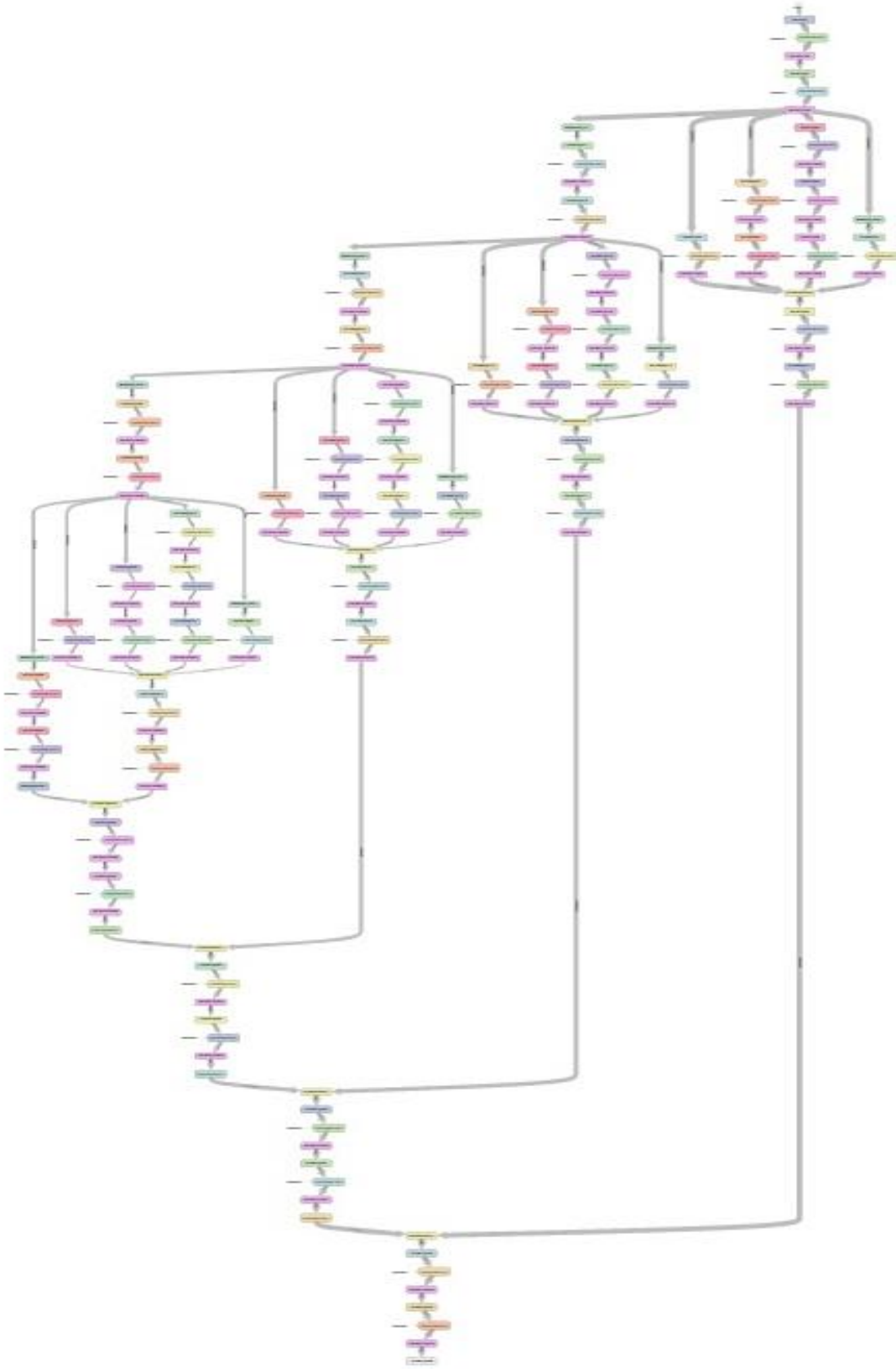
Şekil 5.3: Inception modül kısmının diyagramı (IMP).

Çalışmamızın tüm adımlarını aşağıdaki tabloda yer alan konfigürasyona sahip bir bilgisayar ile gerçekleştirdik.

Tablo 5.2: Kullandığımız bilgisayarın özellikleri.

CPU	Intel Core i7-10875H 2.3 GHZ
Memory	16384Mb RAM
GPU	NVIDIA GeForce RTX 2080 Super with Max-Q Design
İşletim Sistemi	Windows 10, x64

Bunla birlikte araştırmamızı Tensorflow 2.2.0 sürümüne sahip framework, Python 3.6 versiyonu and Keras-nightly 2.5.0 sürümü ile gerçekleştirdik. Çalışmada doğruluk ve kaybın ölçümü için zar katsayısı ve zar kaybı fonksiyonlarını kullandık. Ayrıca eğitim sırasında IoU (Intersection Over Union), hatırlama ve kesinlik değerlerini gözlemledik. Eğitim boyunca Öğrenme oranı 0,001 ve batch size değerini 3 aldık. Yine eğitim boyunca optimize fonksiyonu olarak en yayın kullanılan Adam optimize fonksiyonunu kullandık.



Şekil 5.4: Önerdiğimiz AIM-Unet modeli.

5.3 VERİ SETLERİ

Bu çalışmada karaciğer görüntülerini içeren 3 farklı veri sti kullandık. Bunlardan ikisi halka açık ve internet ortamında rahatlıkla ulaşılabilen veri setleri olan LİTS ve CHAOS verisetleridir. Üçüncü veri seti iste Trakya üniversitesi tıp fakültesinden aldığımız görüntüler ile oluşturduğumuz kendi veri setimizdir. Karaciğer tümörü için ise iki farklı veri seti kullanıldı ve bu iki veri seti de yine halka açık olan veri setleri olan LİTS ve 3DIRCADb veri setleridir. Karaciğer veri setleri oluşturulurken veri setlerindeki hastaların karaciğer içermeyen kesitlerinin çoğu atıldı. Tüm görüntüler 512x512 boyutundandır. Ayrıca veri setlerinde görüntülerin %30 doğrulama (validation) için kullanıldı. Veri setlerinde kullanılan görüntü sayılarının detayları aşağıdaki tabloda yer almaktadır.

Tablo 5.3: Veri setlerindeki görüntülerin dağılımları.

Veri Seti	Eğitim	Doğrulama	Test	Toplam
CHAOS	1935	828	212	2975
LİTS	13171	5644	618	19433
Veri Setimiz	2362	1011	254	3627

5.3.1 Hastalar ve Veri edinme protokolü

Bu çalışma, sağlık bakanlığının belirlediği kriterlere uyularak, Trakya Üniversitesi tıp fakültesinden 12.02.2020 tarihli ve 600-E.411005 sayılı gerekli izin ve TÜTF-BAEK 2020-7254 numaralı etik kurul onayı alınarak yapılmıştır. Ayrıca hasta verileri Türkiye Cumhuriyeti 6698 sayılı kişisel verilerin korunması kanununa uygun bir şekilde alınıp çalışmada işlenmiştir. Bu çalışmada yaş aralığı 25 ile 83 olan toplamda 50 farklı hastadan (28 erkek, 22 kadın) bir kısmı hastalıklı bir kısmı sağlam olan hasta veriler kullanılmıştır. Her bir hastanın ortalama 73 kesitten oluşan abdominal görüntüleri ile veri seti oluşturulmuştur.

5.3.2 BT Edinme Protokolü

Karın MRG'leri, GE MR Singna HDX 3.0T, GE MR Singna HD Excite tarayıcılar 1.5T, Simen MR Skyra 3.0T, Simen magnetom avanto Dot1 dahil olmak üzere 1.5-T, 3-T ve 750 W MR tarayıcılarla sırtüstü pozisyonda gerçekleştirildi. 5T ve GE discovery MR 750 tarayıcılar. T2 ağırlıklı dizi, difüzyon ağırlıklı dizi (b değerleri: 800 s/mm²) standart kurumsal karaciğer MR görüntüleme protokollerinden 2-2.5 ve 2-2.5 dakikalık alım süreleriyle gerçekleştirilmiştir. Kontrastlı T1 dizileri, 12-18 s'lik alım süreleriyle kullanıldı. Sırasıyla 0.2 ve 0.1 mmol/kg dozlarında Gadopentetate dimeglumin ve Gadoxetic Acid Disodium (Primovist) olmak üzere iki farklı kontrast maddesi kullanıldı. Kontrast öncesi T1, T2 ve difüzyon görüntüleri dışında, geç arteriyel faz (enjeksiyondan ~15 s sonra), portal venöz faz (enjeksiyondan ~60 s sonra) ve denge fazı (~60 s) dahil kontrast sonrası görüntüler analiz edildi (Enjeksiyondan 3 dakika sonra). Görüntüleme parametreleri farklı tarayıcılar ve zaman dilimleri arasında değişiklik göstermiştir.

Çalışmada kullanılan Karaciğer BT kesitlerinin elde edilmesi ve sistem bilgisi:

Karın BT (Şekil 2.5) görüntüleri tüm vücut PET/BT görüntülerinin ilgili bölümlerinden elde edildi. Görüntüler, birleşik tüm vücut PET/BT sistemi kullanılarak elde edildi. (Keşif STE; GE Medical Systems, Milwaukee, Wisconsin). Emisyon görüntüleri 3D modunda yatak pozisyonu başına 3 dakika, 3.75 mm dilim kalınlığı, $4.7 \times 6.3 \times 30$ mm BGO kristali, 256×256 matris, enine görüş alanı 570 mm ve eksenel görüş alanı 157 mm için kaydedildi. Veriler iv kontrast olmadan elde edildi ve zayıflama düzeltmesi için kullanıldı. Bununla birlikte, GI yolunu ayırt etmek için oral kontrast kullanıldı.

5.3.3 Veri Setin Oluşturulması

Bir segmentasyon problemin çözülmesi için toplanmasına ihtiyaç duyulan veri miktarına çoğu zaman gücümüz yetmez. Burada güç ile kast edilen zaman, donanım ve belki de en önemlisi verilerin etiketlenmesidir. Çünkü etiketleme için sıradan bir göz yeterli olmayacak, özellikle biyomedikal ya da savunma gibi hassas yaklaşımlar gerektiren alanlarda her bir pikselin lokalizasyon/konumlama bilgisine ihtiyaç vardır. Bunun için çokça ilgili görüntü konusunda uzman gözü ve tecrübesine ihtiyaç duyulmaktadır.

Uygun BT görüntüleri hasta hanenin veri tabanından Tıpta Dijital Görüntüleme ve işleme (DICOM) formatında saklandı. Saklanan tüm görüntüler, 4.7x6.3x30 mm çözünürlüğe yeniden boyutlandırıldı. Uzman radyologlar tarafından karaciğer alanları (ROI) belirlenerek etiketleme işlemleri yapıldı. Etiketleme işleminden sonra orijinal görüntüler 512x512x3 piksel boyutlarında 3 kanal renk derinliğine sahip jpg uzantılı, etiketlenmiş görüntüler ise 512x512x1 boyutlarında tek renk kanalına sahip png uzantılı olarak veri setine eklendi.

5.3.4 LITS Görüntü Veri Seti

Bu veri集中的 görüntüler, dünyanın çeşitli ülkelerindeki kliniklerden ve üniversite hastanelerinden toplanmıştır. IRCAD Institute Strasbourg, Kudüs İbrani Üniversitesi, Tel Aviv Üniversitesi, Sheba Tıp Merkezi, Münih Ludwig Maximilian Üniversitesi, Polytechnique ve CHUM Araştırma Merkezi Montreal ve Nijmegen Radboud Üniversitesi Tıp Merkezi bunlardan bazılarıdır. Bu veri kümesindeki hastaların çoğu karaciğer hastalıklarından muzdariptir. Hastaların görüntülerinde primer (yani karaciğer benzeri hepatosellüler karsinoma, HCC kaynaklı) ve sekonder (yani karaciğer benzeri kolorektal kansere yayılan) tümör hastalıkları ile meme ve akciğer kanserlerinden kaynaklanan metastazlar yer almaktadır. Görüntüler, farklı cihazlardan farklı edinim protokolleri altında alındı. Bu nedenle, verilerin farklı çözünürlükleri, görüntü kalitesi, ve kontrast yoğunlukları. LITS veri seti, eğitim ve test için sırasıyla 131 ve 70 kontrastlı 3D karın BT taraması içerir [67]. Bu hastaların görüntülerindeki kesit sayısı 42 ile 1026, karaciğeri içeren kesit sayısı ise 30 ile 253 arasında değişiyordu. Her hastada ortalama 150 görüntü bulunuyor. Kesitler 0,45 mm ile 6,0 mm arasında farklı kalınlıklara sahiptir. Veriler nfi dosya formatında indirildi ve modelin daha hızlı sonuçlanması için karaciğersiz görüntüler çıkarıldı. Karaciğerin eğitim ve test setlerinde hacimler normal dağılım gösterir ve bilinen dağılımlara benzerdir. Veri setinde toplam 19.433 adet görsel bulunmakta olup, bu görsellerin kullanım dağılımları

Tablo 5.3'de detaylı olarak gösterilmiştir [67].

5.3.5 CHAOS Görüntü Veri Seti

Veri kümeleri Dokuz Eylül Üniversitesi Hastanesinin PACS'sinden toplanmıştır. Tüm hastaların aynı şartlar altında çekilmiş olduğu 40 farklı hastanın (20 eğitim ve 20 test vakası) BT görüntüleri

yer almaktadır. Bu hastalar, sağlıklı bir karaciğere sahip potansiyel karaciğer donörleridir. Yani karaciğerde tümör, lezyon veya başka bir hastalık yoktur. Her veri seti 512×512 çözünürlüklü DICOM görüntüleri içerir. Eğitim verileri hem DICOM görüntülerini hem de bunların uzman radyologlar tarafından etiketlenilmiş (maskelenmiş) görüntülerini içerir. Test veri seti ise yalnızca DICOM görüntülerini içerir. 0,7-0,8 mm aralığında ve 3 ila 3,2 mm dilimler arası mesafeye (ISD) sahip x-y mesafesi (yani, rutin klinik prosedürler nedeniyle bu alımlar için daha küçük ISD kullanılamaz). Bu veri setinde hasta başına ortalama 90 kesit görüntü yer almaktadır (yani minimum 77, maksimum 105 dilim). Eğitim kümesinde toplam 2975 kesit görüntü vardır. Bu BT görüntüleri kontrast madde enjeksiyonu sonrası bir hastanın üst karın bölgesinden alınmıştır. Enjeksiyon periyodu sırasında, karaciğer ve karaciğer veninin hepatik parankimi de portal ven tarafından kanın aşırı verilmesiyle güçlendirilir ve maksimize edilir. Portal venler iyi bir şekilde artar ve bu aşamada hepatik venlerin bir miktar güçlenmesine yardımcı olur. Sonuç olarak, bu aşama, ameliyattan önce karaciğer ve vasküler segmentasyon için yaygın olarak uygulanır. Bu veri kümesi yalnızca sağlıklı bir karaciğeri içermesine rağmen, kontrast enjeksiyonu nedeniyle gelişmiş vasküler yapılar, bitişik organın benzer bir Hounsfield değer aralığına sahip olmasını sağlar [37], bu görevin zorluğu Şekil 2.9'deki gibidir. LITS veri kümesi, bu veri kümesi düzensiz karaciğer şekillerini (yani, çeşitli boyut veya yönelim) içerir ve hastalar arasında anatomik yapıların önemli şekil farklılıklarına sahiptir.

5.3.6 3DIRCADb Görüntü Veri Seti

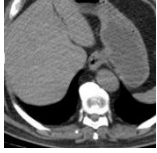
Bu veri kümesi, 131 hasta kaydından oluşan LITS veri kümesinden 28-47 hasta alt kümesidir. 3DIRCADb veri setine toplam 20 hasta dâhil edilmiştir. Bu hastaların 16 tanesinde tümör görüntüsü var. Tümörlü dilimlerin toplam sayısı 588'dir.

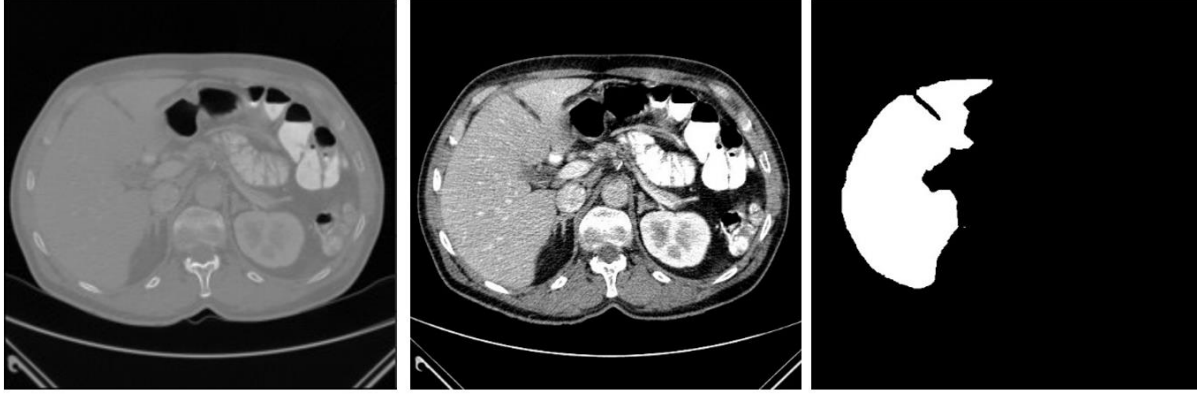
5.4 VERİ HAZIRLAMA VE İŞLEME

Ön işleme, kesitler şekilde gerçekleştirildi. İlk adımda, veri setindeki her hastanın karaciğer içermeyen kesitleri veri setinden çıkarıldı. Alakasız organ ve nesneleri dışarda bırakmak için Hounsfield birim değerleri bir fonksiyon yardımıyla otomatik olarak $[-100, 200]$ aralığında pencerelendi ve ardından histogram eşitleme ile kontrastı artırıldı. Veriler jpg ve png formatlarına dönüştürülerek ön işleme, yeniden örnekleme ve yapılandırma sonrası eğitimlere hazır hale getirilmiştir. Bir modeli eğitirken, veri kümesi küçükse ağı etkili bir şekilde eğitmek için veri

artırma önemlidir. Bu nedenle mevcut eğitim görüntülerine farklı afin deformasyonlar uygulanarak veri artırma işlemi uygulanmaktadır [68]. Özellikle medikal görüntü veri kümeleri oluşturulurken verilerin etiketlenmesi işlemi oldukça maliyetli, zahmetli ve uzun zaman aldığı için veri kümesindeki görüntü sayısı yetersiz kalmaktadır. Az sayıda eğitim verisi, aşırı uyum sorununu gündeme getirir. Veri kümemizdeki her görüntüye merkez kırpma, saat yönünde döndürme, saat yönünün tersine döndürme, ızgara bozulması, dikey çevirme ve rastgele parlaklık kontrastı uygulandı. Her görselden ekstra 6 farklı görünüm oluşturuldu. Bu sayede veri seti 7 kat arttırılmıştır. Tablo 5.4’de bir görüntüye uygulanan yöntemlerin sonuçlarının görüntüleri yer almaktadır. Veri setindeki resimlerin orijinal boyutları 512×512’dir. Bu boyutlardaki resimlerin kullanılması GPU’yu yoracağından, batch boyutunu ayarlama esnek olunamayacağı ve yetersiz hafıza hatalarını önlemek için resimler 256×256’ya yediden boyutlandırılmıştır. Ayrıca görüntüler 0 ile 1 arasında normalize edilmiştir.

Tablo 5.4: Veri artırma sonrasında elde edilen görüntüler.

Original görüntü	Merkezi kırpma	Saat yönünün tersine döndürme	Izgara bozulması	Dikey aynalama	Rastgele parlaklık kontrast	Saat yönünde döndürme
						



(a)

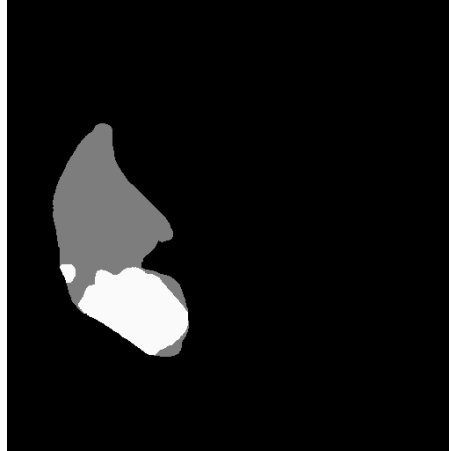
(b)

(c)

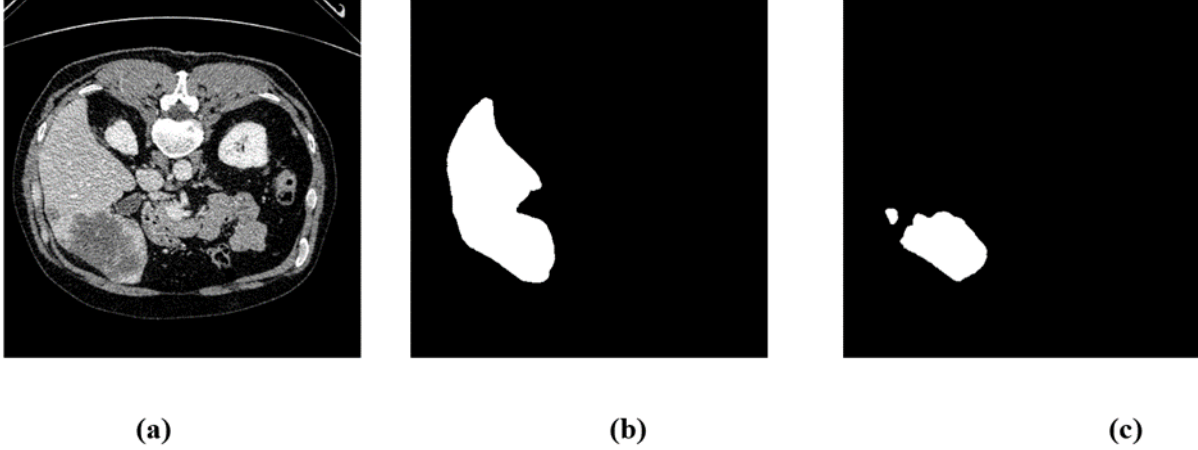
Şekil 5.5: Yukarıdaki şekilde bir hastanın herhangi bir kesit görüntüsünün RAW hali (a), filtreleme yapılmış hali (b) ve uzman radyolog tarafından etiketlenmesi yapılan görüntü (c) yer almaktadır.

Tümörlü alanın tespit edilebilmesi için LITS veri kümesinde uzman radyologlar tarafından etiketlenmiş olan

Şekil 5.6:'da görüldüğü gibi karaciğer ve tümörlü bölgeleri içeren görüntüleri iki boyutlu olarak kayıt edildi. Sadece tümör içeren veri kümesindeki görüntülerin %10'nunu ayırdıktan sonra geri kalan görüntülerin %80'nini modeli eğitirken, geri kalan verilerin %20'sini doğrulama için kullandık.



Şekil 5.6: Uzman radyolog tarafından etiketlenmiş karaciğer (gri renkli bölge) ve tümörlü alan (beyaz renkli bölge).



Şekil 5.7: Yukarıdaki şekilde bir hastanın herhangi bir kesit görüntüsünün RAW hali (a), uzman tarafından işaretlenmiş karaciğer alanı (b) ve yine uzman radyolog tarafından etiketlenmesi yapılan tümörlü alan (c) yer almaktadır.

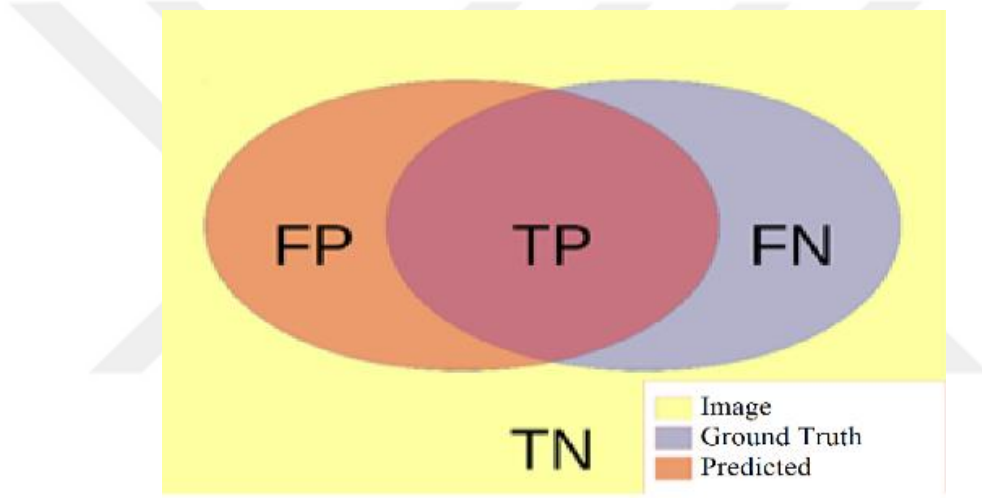
5.5 İŞLEM SONRASI ADIMLAR

Herhangi bir görüntü model tarafından tahmin edilirken veya test edilirken, görüntünün her pikseli bağımsız olarak karaciğer veya arka plan olarak sınıflandırılır. Görüntünün her bir pikselini sınıflandırmak için modelimizin son katmanı olan çıktı katmanında “sigmoid” fonksiyonu kullanıldı. Tahmin sonucunda elde edilen görüntü değeri, oluşturulan bir eşik fonksiyonuna gönderilir. Fonksiyon gelen görüntünün, her pikselin değerini ayrı ayrı inceler. Piksel değeri 0,5'ten büyükse, o piksel karaciğer olarak kabul edilir ve o pikselin değerine 1 değeri atar. Piksel değeri 0,5'ten küçük ise bu pikseli arka plan olarak kabul eder ve piksele 0 değerini atanır. Bu işlem benzer şekilde tümör görüntüsü içinde uygulanır.

Hastanın tüm görüntülerinden 3 boyutlu bir görünüm elde edilecek ise hastaya ait tüm görüntülerin kesitlerine yukarıdaki adımlar uygulandıktan sonra bu kesitler numpy kütüphanesi tarafından oluşturulan çok boyutlu bir dizide katmanlar üst üste birleştirilir ve STL uzantılı olarak kaydedilir. Daha sonra MeshLAB programı ile mesh (Laplacian smoot, Faces oryantasyon) işlemleri uygulanır. Elde ettiğimiz bir hastanın 3 boyutlu görüntüsü Şekil 6.7:'de yer almaktadır.

6. DENEY SONUÇLARI VE ANALİZ

Deneysel sonuçları analiz etmek için çeşitli ölçüm performansları dikkate alınır [49 , 50]. Doğruluk (AC), hatırlama (hassasiyet) (RE), yaygınlık (PV), özgülük (SP), kesinlik (PR), Zar benzerlik katsayısı (DSC) ve Jaccard benzerlik katsayısı (JSC) ölçüm performansları bunlardan bazılarıdır. Bu performans ölçümlerini hesaplamak için, Şekil 5'te Gerçek Pozitif (TP), Yanlış Pozitif (FP), Gerçek Negatif (TN) ve Yanlış Negatif (FN) değişkenlerini içeren bir karışıklık matrisi kullanılır.



Şekil 6.1: Tahmin sonuçları karşılaştırırken gerçek pozitif, gerçek negatif, yanlış pozitif ve yanlış negatif bölgelerin grafik gösterimi.

Eğitim sırasında doğruluğu (Accuracy) hesaplamak için denklem (6.1) kullanıldı ve duyarlılığı (Recall) hesaplamak için ise denklem (2) kullanıldı.

$$AC = \frac{TP + TN}{TP + TN + FP + FN} \quad (6.1)$$

$$RE = \frac{TP}{TP + FN} \quad (6.2)$$

Ayrıca, Kesinlik (Precision) [69] değerini hesaplamak için de denklem (3)'e başvurulmuştur.

$$PR = \frac{TP}{TP + FP} \quad (6.3)$$

Çapraz entropi kaybı, anlamsal bölümlleme görevlerinde yaygın olarak kullanılan kayıp işlevidir. Çapraz entropi kaybı, her pikseli tek tek inceler ve her piksel kategorisi için tahmin edilen sonuçları (olasılık dağılım vektörü) ısı kodlu etiket vektörüyle karşılaştırır. Yalnızca iki kategori olduğunda, BCE kaybı adı verilen bir ikili çapraz entropi kaybı kullanılır ve Denklem (4) ile gösterilir. Veri setlerini eğitirken, ikili çapraz entropi fonksiyonunu kayıp fonksiyonu olarak seçtik. Çapraz entropi [4], belirli bir rastgele değişken veya olaylar dizisi için iki olasılık dağılımı arasındaki farkın bir ölçüsü olarak tanımlanabilir. Piksel düzgün çalışıyorsa, nesneleri ve segmentasyon seviyelerini sınıflandırmak için yaygın olarak kullanılır.

İkili çapraz entropi şu şekilde tanımlanır:

$$L_{BCE} = -[P \log(G) + (1 - P) \log(1 - G)] \quad (6.4)$$

Burada P, model tarafından tahmin edilen görüntüyü ve G ise işaretlenmiş görüntüdür.

Zar puanını, tahmin edilen ve işaretlenmiş görüntü arasındaki eşleşme düzeyi olarak kabul edebiliriz. Zar skorunun aralığı 0 ile 1 arasında bir değerdir. Skor ne kadar yüksekse performans o kadar iyidir. Bu değerın sıfır olması tahmin edilen görüntü ile işaretlenmiş görüntü arasında bir ilişki olmadığını gösterir. Tahmin edilen görüntü ve işaretlenmiş görüntü tamamen örtüşüyorsa, maksimum Zar puanının 1 olduğu açıktır. İkili kesinlik hesaplamasına ek olarak 0,5 olarak belirlediğimiz eşik parametresi hesaplanır. Tahmin edilen her piksel, eşik ile karşılaştırılır. Piksel değeri eşik değerinden küçük veya ona eşitse, o pikselin değerine 0, piksel değeri eşik değerinden büyükse o piksel değerine 1 atanır.

Zar katsayısı, denklem (5) kullanılarak hesaplanır ve Zar kaybı, denklem (6) kullanılarak hesaplanır.

$$DSC = \frac{2|P \cap G|}{|P| + |G|} \quad (6.5)$$

$$DL = 1 - \frac{2|P \cap G|}{|P| + |G|} \quad (6.6)$$

Burada P, önerdiğimiz model tarafından tahmin edilen görüntüyü ve G ise işaretlenmiş görüntüyü ifade etmektedir.

Jaccard indeksi çok popülerdir ve aşağıdaki formda gösterildiği gibi ikili veriler için benzerlik indeksi olarak kullanılır.

$$JSC = \frac{P \cap G}{P \cup G} \quad (6.7)$$

Ayrıca ortalama IoU, hatırlama ve kesinlik metriklerinin yanı sıra Zar skoru metriğini de hesapladık. Bu metrik değerler 3 veri seti için Tablo 6.1 ve Tablo 6.2'da verilmiştir.

Tablo 6.1: AIM-Unet'in veri setleri üzerindeki performansı.

Veri seti	Accuracy (%)	Recall (%)	Precision (%)	Mean IoU (%)
Veri setimiz	99.54 ± 0.27	96.35 ± 2.64	98.47 ± 1.23	97.21 ± 1.46
CHAOS	99.75 ± 0.10	96.39 ± 6.23	99.69 ± 0.55	97.91 ± 3.13
LITS	99.48 ± 0.48	95.12 ± 6.15	96.78 ± 3.78	95.82 ± 3.52

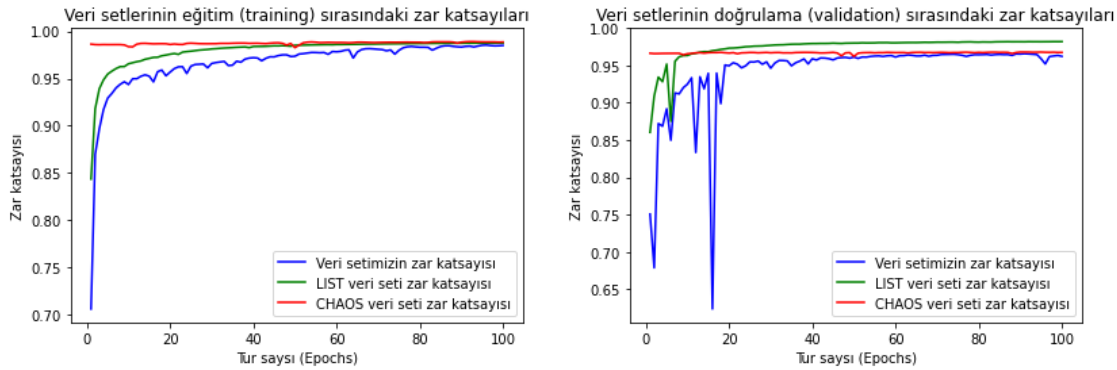
Tablo 6.2: AIM-Unet'in test setleri üzerindeki çapraz performansı.

Model ismi	Eğitim veri setleri	Test veri setleri					
		Veri setimiz		LITS		CHAOS	
		Dice (%)	Jaccard (%)	Dice (%)	Jaccard (%)	Dice (%)	Jaccard (%)
AIM-Unet	Veri setimiz	97.38 ± 1.63	94.95 ± 3.00	93.60 ± 7.49	88.60 ± 9.10	97.64 ± 6.74	95.91 ± 7.81
	LITS	77.43 ± 27.60	69.61 ± 28.96	95.77 ± 5.16	92.22 ± 6.90	94.12 ± 5.97	89.38 ± 8.70
	CHAOS	97.06 ± 3.89	94.53 ± 6.02	93.46 ± 9.46	88.64 ± 10.49	97.86 ± 4.65	96.10 ± 6.21

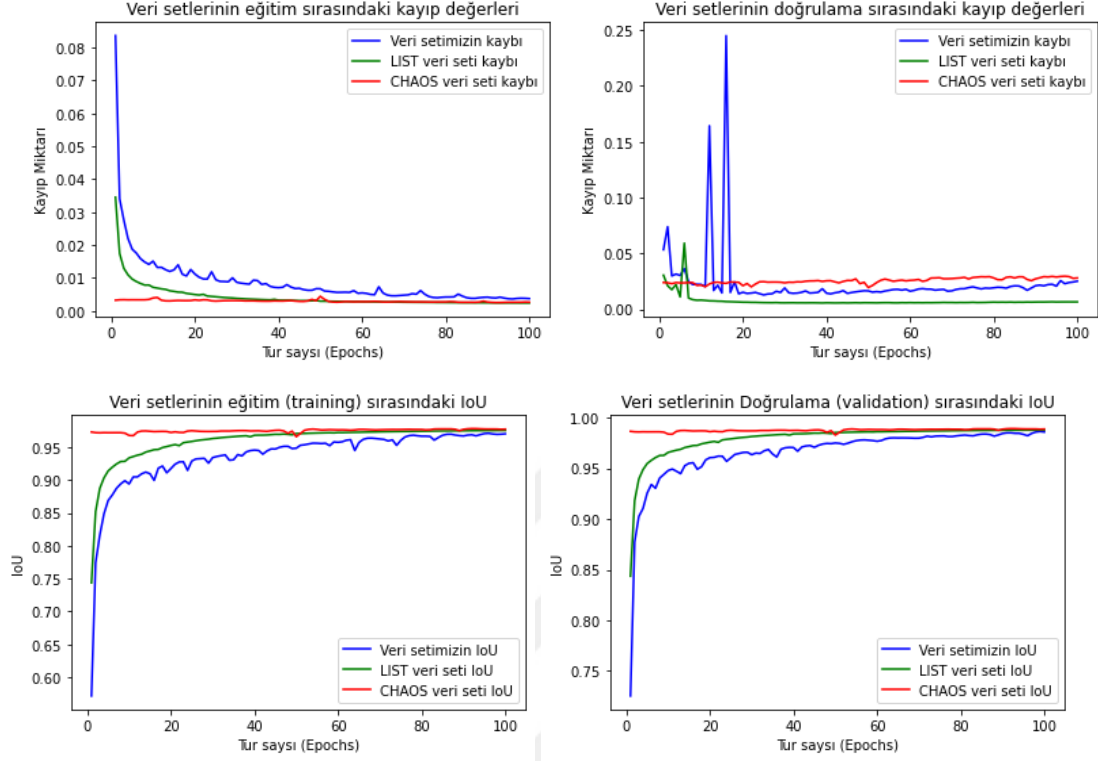
Bu çalışmanın amacı, U-Net modeline eklenen Inception modülündeki farklı boyutlardaki filtrelerin otomatik olarak karaciğer bölgesini çıkarabilme yeteneğini ortaya koymaktır. Yöntemin değerlendirilmesi, bir radyolog tarafından tahmin edilen alan ile gerçek alan karşılaştırılarak ölçülmüştür. Sonuçların doğru çıkması yönünde en sık kullanılan ölçüm değerleri Zar katsayısı ve Jaccard indeksidir (IOU). Modelimizde bu değerlerdeki değişiklikler Tablo 6.3'te gösterildiği gibidir. Ancak biz zar katsayısı ve Jaccard indeksinin yanında diğer metrikleri de gözlemledik ve bunları Tablo 6.1'e kaydettik. Şekil 6.2'deki dalgalanmaların nedenini ve kullandığımız Adam optimizasyon algoritmasındaki öğrenme oranının yenilenmesinden kaynaklı hata miktarının değişiminden kaynaklanmaktadır. Modelin eğitimi sonucunda elde ettiğimiz bazı maskeler Şekil 6.3'te alttaki 64 adet görüntü ise filtreleme sonucu elde edilen maskeleri ifade etmektedir. Önerdiğimiz modelin başarısını ölçmek için ağı sadece kendi veri setimiz ile eğittik

ve ardından LIST ve CHAOS veri setleri üzerinde ayrı ayrı çaprazlama olarak test ettik. Daha sonra, bu işlemi diğer veri seti için de tekrarladık ve sonuçları Tablo 6.2'da gösterildiği gibi raporladık. Tablo 6.2'da ki değerler, önerilen modelin her bir veri kümesi üzerinde ayrı ayrı eğitimden sonraki test sonuçlarını göstermektedir. Eğitim ve test veri setlerinde kullanılan görüntü sayıları

Tablo 5.3'te verilmiştir. Tablo 6.2'ya dikkat edersek, LIST veri setinde eğitilen ve bizim veri setimizde test edilen sonuçlar en büyük standart sapmalara sahiptir. Bunun sebebi ise veri setimizdeki görüntünün morfolojik işlemler sonucunda farklılık göstermesidir. Yine Tablo 6.2'da görüldüğü gibi, en başarılı test sonuçları CHAOS veri setinde eğitilen modelden alınmıştır. Ancak veri setimiz üzerinde eğitilen ve Chaos veri seti üzerinde test edilen AIM-Unet modelin test sonuçları en yüksek değerlere oldukça yakındır. Ayrıca tablodaki verilere bakıldığında AIM-Unet modelinin tüm veri kümelerinde başarılı sonuçlar elde ettiği görülmektedir. Tablo ayrıca bize en düşük performansın ve en büyük standart sapmanın LITS veri seti üzerinde eğitilip, veri setimiz üzerinde test edilen sonuç olduğunu göstermektedir. Bunun en büyük sebebi LIST veri setinin görüntülerini kendimiz bazı morfolojik işlemler yaptıktan sonra kesit kesit kaydetmiş olup, veri setimize gelen görüntülerin kesit olarak uzaman tarafından ve kendi belirlediği kontrast ayarlarında olmasından kaynaklandığını düşünmekteyiz.

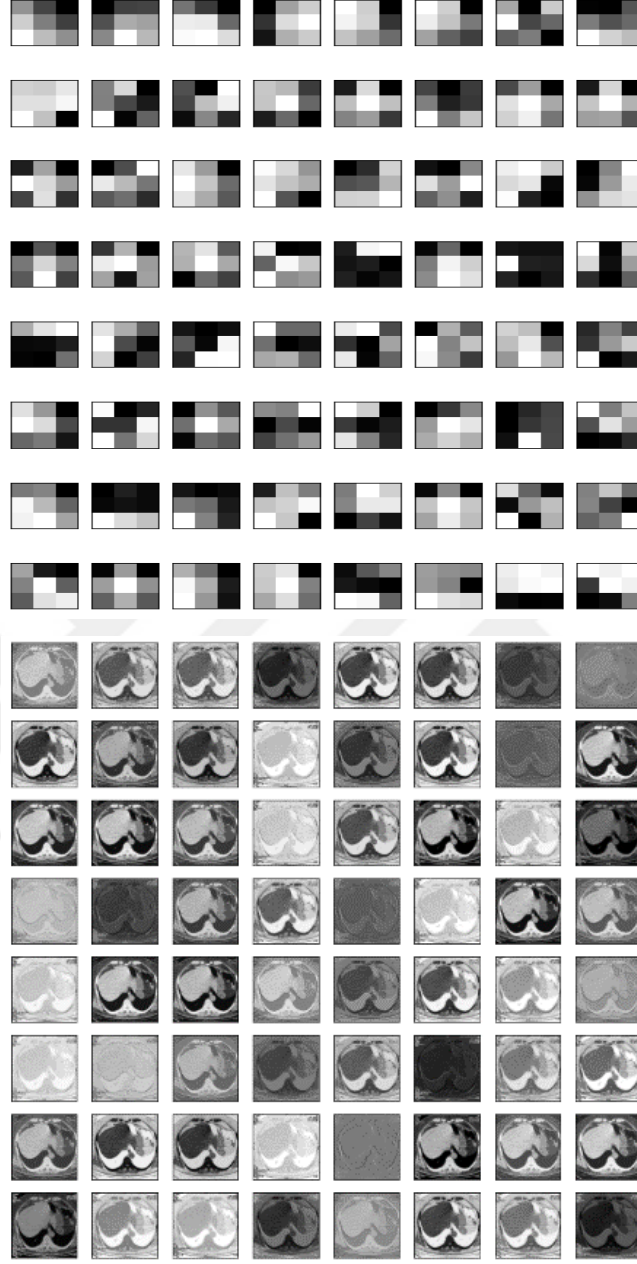


Şekil 6.2: Yukarıda üç farklı veri seti üzerinde eğittiğimiz AIM-Unet modelinin her epoch esnasındaki Zar katsayısı, kayıp miktarı ve IoU değerlerinin değişimini gösteren grafikler yer almaktadır.



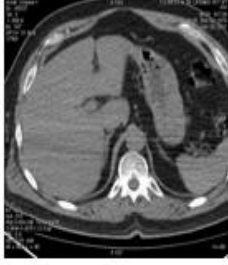
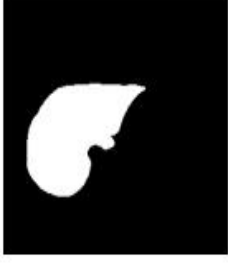
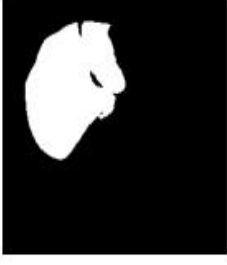
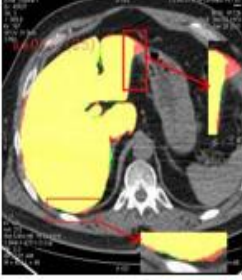
Şekil 6.2: Yukarıda üç farklı veri seti üzerinde eğittiğimiz AIM-Unet modelinin her epoch esnasındaki Zar katsayısı, kayıp miktarı ve IoU değerlerinin değişimini gösteren grafikler yer almaktadır.(devam)

Bu çalışmanın amacı, Unet modeline eklenen Inception modülündeki farklı boyuttaki filtrelerin karaciğer bölgesini otomatik çıkarabilme yeteneğini ortaya koymaktır. Metodun değerlendirilmesini bir radyolog tarafından tahmin edilen alan ile gerçek alanın karşılaştırılması yapılarak ölçüldü. En yaygın kullanılan ölçüm değerleri: Zar katsayısı ve Jaccard index (IOU) 'dir. Bu değerlerin modelimizdeki değişimleri Tablo 6.3'te gösterildiği gibidir.



Şekil 6.3: Yukarıda eğitim sırasındaki birinci konvolüsyon katmanındaki 64 adet 3x3 ve dördüncü katmanındaki 64 adet 256x256 filtreleri yer almaktadır.

Önerilen mimarının performansını değerlendirmek için bilgisayarlı tomografi BT tarama veri setini tartışıyor ve analiz ediyoruz. AIM-Unet tarafından farklı veri setleri üzerindeki görüntülerden tahmin edilen sonuçlar aşağıdaki tabloda görülmektedir. Karşılaştırma görüntüsündeki kırmızı alanlar modelimizin kaçırdığı, yeşil alanlar modelin yanlış tahmin ettiği ve sarı alanlar ise modelin doğru tahmin ettiği bölgeleri göstermektedir.

	Veri setimizdeki Görüntü	LITS veri stindeki görüntü	CHAOS veri setindeki görüntü
BT kesit			
İşaretlenmiş görüntü			
Tahmin edilen görüntü			
Karşılaştırma (işaretlenmiş görüntü & AIM-Unet)			

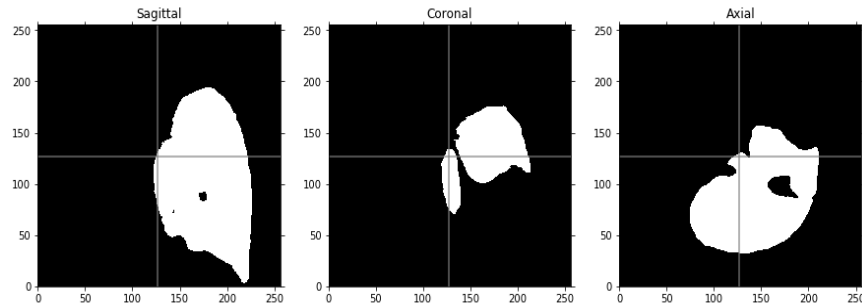
Şekil 6.4: Önerilen modelin (AIM-Unet) veri setlerinin görüntüleri üzerindeki performansı. Karşılaştırma görüntüsündeki kırmızı alanlar modelimizin kaçırdığı bölgeleri, yeşil alanlar modelin yanlış tahmin ettiği alanları ve sarı alanlar modelin doğru tahmin ettiği bölgeleri göstermektedir.

Tablo 6.3: Karaciğer için önerilen yöntemin (AIM-Unet) diğer mimarilerle karşılaştırılması.

Model	Veri seti	Dice (%)	Jaccard (%)
Ronneberger et al. [70]	LITS	95.2	90.8
Lin et al. [71]	LITS	90.9	86.7
Li et al. [72]	LITS	96.5	92.6
Yuan et al. [73]	LITS	96.7	92.9
Thi et al. [71]	LITS	96.0	90.4
Unet	LITS	87.86 $\pm$ 20.28	82.10 $\pm$ 21.05
AIM-Unet	LITS	95.77 $\pm$ 5.16	92.22 $\pm$ 6.90
Ronneberger et al. [70]	CHAOS	74.75	
Thi et al. [71]	CHAOS	94.92	
Mourya et al. [74]	CHAOS	97.0 $\pm$ 0.03	
Unet	CHAOS	68.30 $\pm$ 33.96	60.35 $\pm$ 33.24
AIM-Unet	CHAOS	97.86 $\pm$ 4.65	96.10 $\pm$ 6.21
Unet	Veri setimiz	46.64 $\pm$ 35.52	37.69 $\pm$ 31.72
AIM-Unet	Veri setimiz	97.38 $\pm$ 1.63	94.95 $\pm$ 3.00

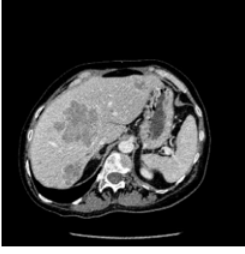
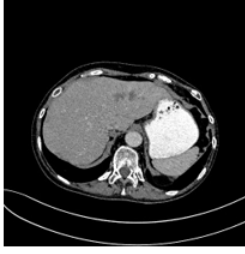
Not: *'lı sonuçlar [37] referans numaralı makaleden alınmıştır.

Önerilen modelin ortalama zar katsayısı ve jaccard index değerleri Tablo 6.3'te yer almaktadır. Tablodan da görüldüğü gibi LITS veri seti üzerinde standart U-net ve FPN modellerinden daha yüksek başarıya sahip iken MFunet ve diğerlerinden biraz daha düşük başarısı göstermiştir. Ancak Chaos veri seti ve oluşturduğumuz veri set üzerinde önerdiğimiz model daha fazla başarılı model olmuştur. Ayrıca Jaccard indexlerine baktığımızda üç veri set üzerinde en başarılı performansı gösteren yine önerdiğimiz model olmuştur.

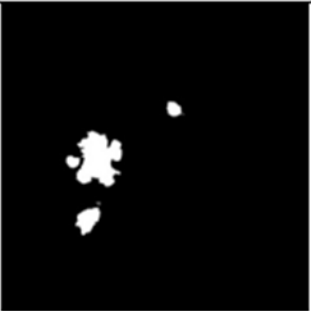
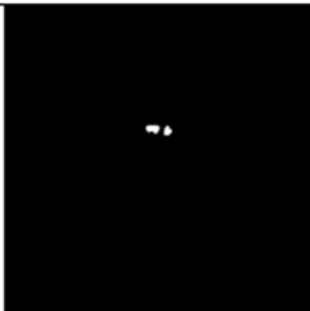
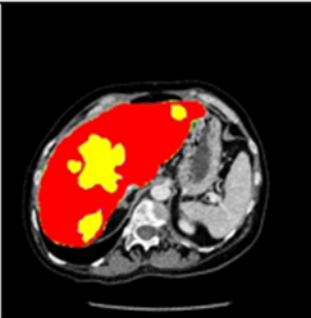
**Şekil 6.5:** Üç boyutlu elde ettiğimiz bir karaciğerin farklı açılardan kesit görüntüleri.

Tablo 6.4: Karaciğer tümörü için önerilen yöntemin (AIM-Unet) diğer mimarilerle karşılaştırılması.

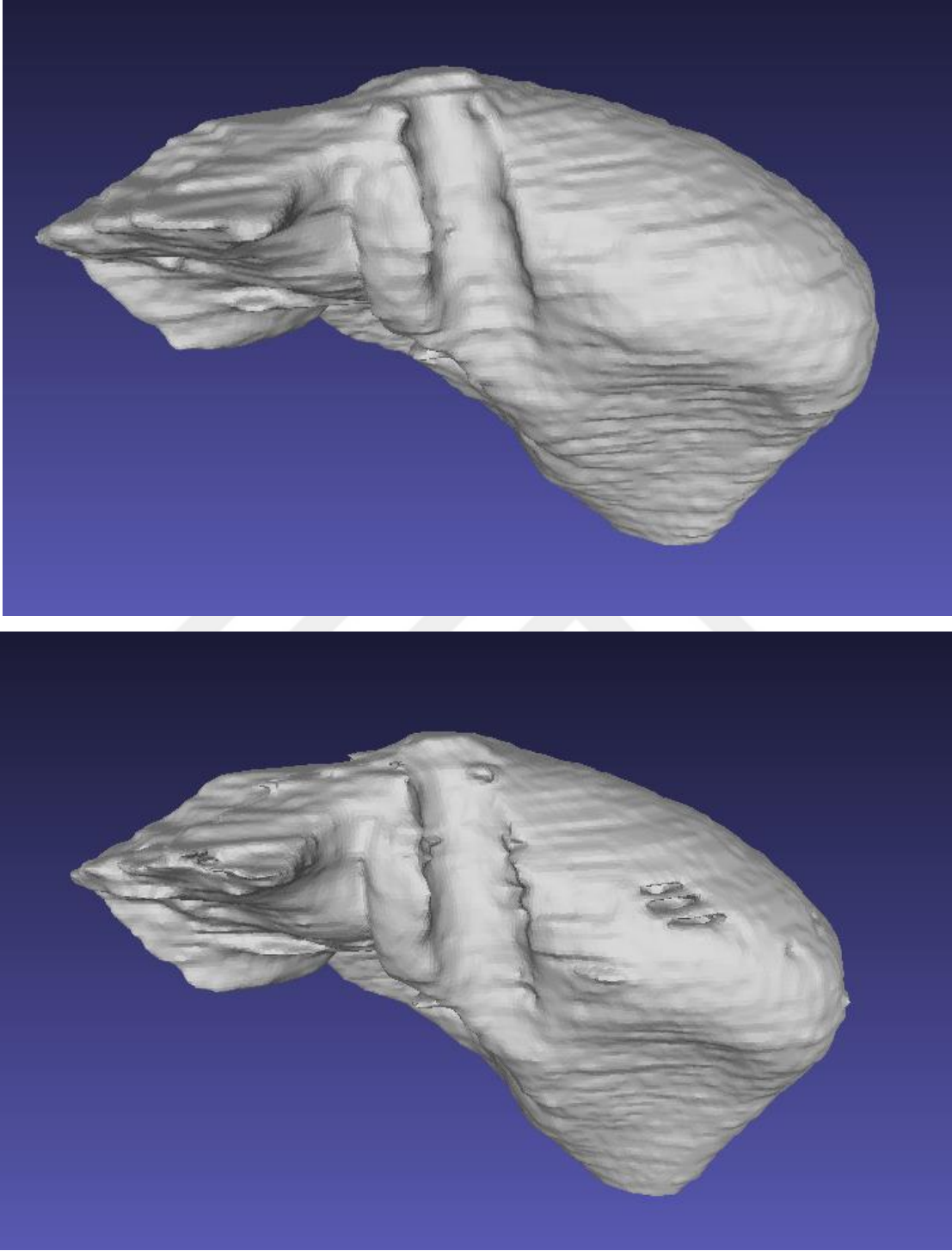
Model	Veri seti	Dice (%)
AIM-Unet	3DIRCADb2	65.5 ± 12.9
Li et al. [75]	3DIRCADb2	66.3
Omar et al. [76]	3DIRCADb2	75.0
Christ et al. [77]	3DIRCADb2	56.0
Chen et al. [78]	LITS	66.6
Vorontsov et al. [79]	LITS	66.1
Bi et al. [80]	LITS	64.5
Yuan et al. [73]	LITS	65.7
Zhang et a	LITS	58.7 ± 28.3
Li et al. [75]	LITS	74.1
AIM-Unet	LITS	75.6 ± 13.4

Veri seti	LITS veri seti görüntüsü (hasta 16 kesit 406)	3DIRCADb veri seti görüntüsü (hasta 12 kesit 64)
BT kesiti		
İşaretli karaciğer alanı		

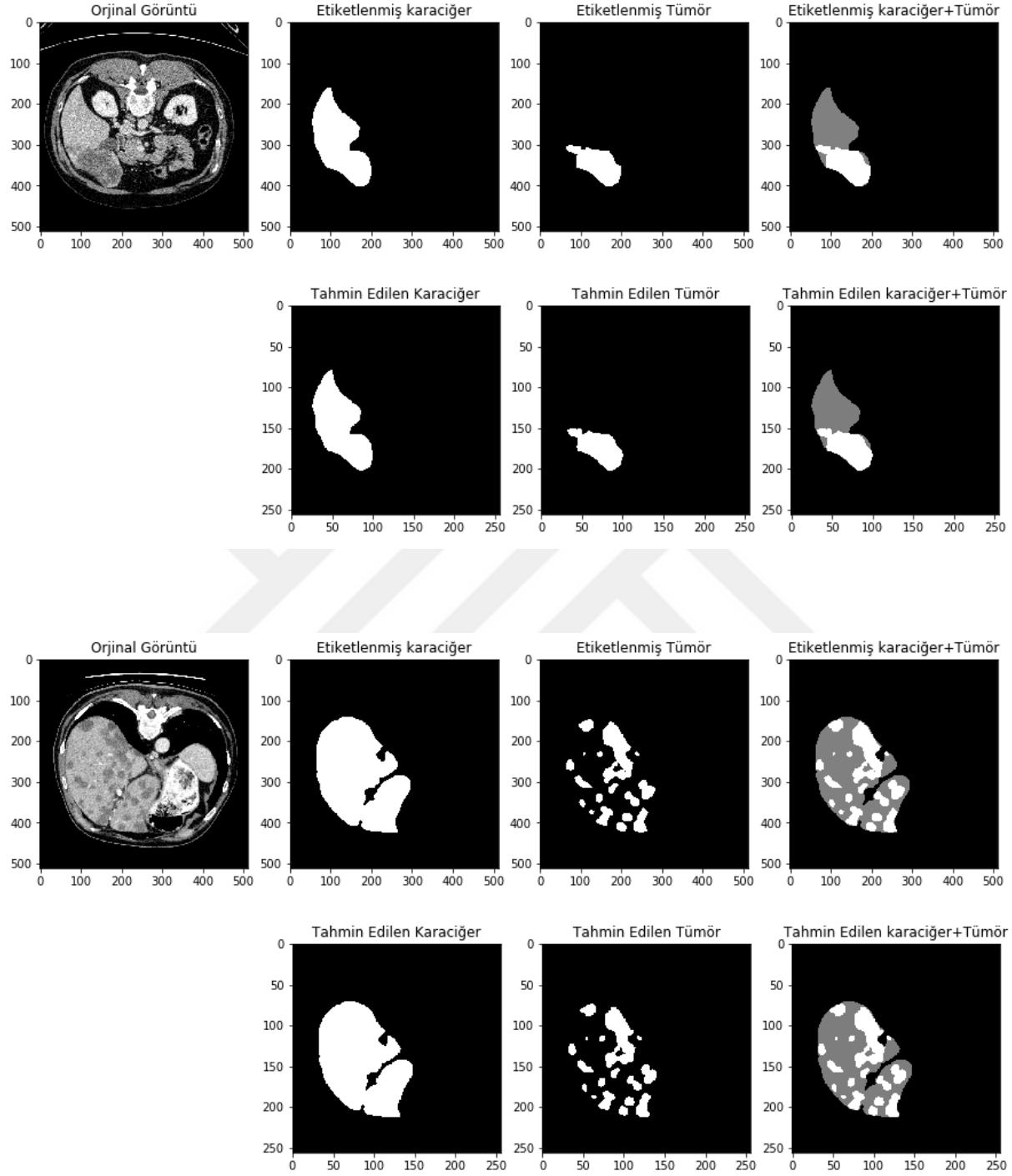
Şekil 6.6: Önerilen modelin (AIM-Unet) veri setlerinin görüntüleri üzerindeki performansı. Karşılaştırma görüntüsündeki yeşil alanlar, modelimizin karaciğer alanı için kaçırdığı bölgeleri, mavi alanlar ise tümör alanı için modelin kaçırdığı bölgeleri göstermektedir.

Veri seti	LITS veri seti görüntüsü (hasta 16 kesit 406)	3DIRCADb veri seti görüntüsü (hasta 12 kesit 64)
Tahmin edilen karaciğer alanı		
İşaretlenmiş tümör alanı		
Tahmin edilen tümör alanı		
Karşılaştırma (işaretlenmiş alan & AIM-Unet)		

Şekil 6.6: Önerilen modelin (AIM-Unet) veri setlerinin görüntüleri üzerindeki performansı. Karşılaştırma görüntüsündeki yeşil alanlar, modelimizin karaciğer alanı için kaçırdığı bölgeleri, mavi alanlar ise tümör alanı için modelin kaçırdığı bölgeleri göstermektedir. (devam)



Şekil 6.7: Üstteki resim uzman doktorun işaretlediği 3 boyutlu karaciğer kısmı, alttaki görüntü ise önerdiğimiz modelin 3 boyutlu tahmin ettiği karaciğer görüntüsünü göstermektedir.



Şekil 6.8: LITS veri setindeki 104 ve 108 nolu hastaların uzmanlar tarafından işaretlenmiş görüntüleri (üst satırdakiler) ve modelimizin tahmin ettiği görüntüler (alt satırdakiler).

7. SONUÇ VE ÖNERİLER

Bu çalışmada Unet modelinin atlama bağlantısına Inception modelinin farklı filtreleme modülünü ekleyip BT karın görüntülerindeki yerel özelliklerden yararlanarak segmentasyona nasıl etki edeceğini inceledik. Bu çalışmayı 3 farklı veri seti üzerine uyguladık. Sonuçlardan da anlaşacağı gibi eklediğimiz farklı filtre modülü standart ağıdan daha fazla yerel özelliklerin öğrenmesine yardımcı olduğunu gördük. Modelimiz uçtan uca eğitilebilir bir modeldir. Önceden eğitilmiş ağırlıklara ihtiyaç duymamaktadır. Modelimizin standart Unet modeline göre daha karmaşık ve daha fazla parametre kullandığını gördük. Algoritmanın tamamı, tipik bir masaüstü bilgisayar kullanarak bir dilimi 1.12 saniyede segmentlere ayırabildi. Bir BT kesit görüntüsünün bir uzman tarafından bölütlenmesi için gereken süre ortalama 2 dakikadır. Bu nedenle önerdiğimiz yöntem kabul edilebilir doğruluk seviyelerinde manuel bölütlemekten yaklaşık 100 kat daha hızlıdır. Ayrıca sonuçlar, önerilen bu modelin sadece eğitim sırasında değil, aynı zamanda test aşamasında da daha iyi performans sağladığını göstermektedir. Önerdiğimiz mimari karaciğer lezyonlarının bulunmasında çok yüksek başarılar sağlamaktadır. Ancak şunu söylemeyi unutmamalıyız ki atlama bağlantısına eklediğimiz modül sayesinde önerdiğimiz modelin parametre sayısı çok fazla arttı. Buda bilgisayar için eğitim ve doğrulama sırasında hem fazla işlem hem de fazla zaman alması anlamına gelmektedir. Gelecekte önerdiğimiz bu modeli direk 3B hale getirip görüntüyü 3 boyutlu verip sonucu direk 3 boyutlu çıkaracak hale getirip bunu farklı veri setlerine ve farklı medikal organlarına uygulamayı istiyoruz.

8. KAYNAKLAR

- [1] L. A. Torre, F. Bray, R. L. Siegel, and J. Ferlay, "Interpretation of model in modelling ecological niche.pdf," vol. 65, no. 2, pp. 87–108, 2015, doi: 10.3322/caac.21262.
- [2] F. Bray, J. Ferlay, I. Soerjomataram, R. L. Siegel, L. A. Torre, and A. Jemal, "Global cancer statistics 2018: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries," *CA. Cancer J. Clin.*, vol. 68, no. 6, pp. 394–424, 2018, doi: 10.3322/caac.21492.
- [3] H. Greenspan, B. Van Ginneken, and R. M. Summers, "Guest Editorial Deep Learning in Medical Imaging: Overview and Future Promise of an Exciting New Technique," *IEEE Trans. Med. Imaging*, vol. 35, no. 5, pp. 1153–1159, 2016, doi: 10.1109/TMI.2016.2553401.
- [4] H. Kaya, A. Çavuşoğlu, H. B. Çakmak, B. Şen, and D. Delen, "Görüntü bölütleme ve görüntü benzetimi yöntemleri yardımı ile hastalığın teşhis ve tedavi sonrası süreçlerinin desteklenmesi: Keratokonus örneği," *J. Fac. Eng. Archit. Gazi Univ.*, vol. 31, no. 3, pp. 737–747, 2016, doi: 10.17341/gummfd.85749.
- [5] D. Lesage, E. D. Angelini, I. Bloch, and G. Funka-Lea, "A review of 3D vessel lumen segmentation techniques: Models, features and extraction schemes," *Med. Image Anal.*, vol. 13, no. 6, pp. 819–845, 2009, doi: 10.1016/j.media.2009.07.011.
- [6] H. Li and A. Yezzi, "Vessels as 4-D Curves: Global Minimal 4-D Paths to Extract 3-D Tubular Surfaces and Centerlines," *IEEE Trans. Med. Imaging*, vol. 26, no. 9, pp. 1213–1223, 2007, doi: 10.1109/TMI.2007.903696.
- [7] S. Esneault, C. Lafon, and J. Dillenseger, "Liver Vessels Segmentation Using a Hybrid Geometrical Moments/Graph Cuts Method.pdf," *IEEE Trans. Biomed. Eng.*, vol. 57, no. 2, pp. 276–283, 2010, doi: 10.1210/jcem-52-1-34.
- [8] Jens N. Kaftan, H. Tek, and T. Aach, "A Two-Stage Approach for Fully Automatic Segmentation of Venous Vascular Structures in Liver CT Images Jens," *Med. Imaging 2009 Image Process. Orlando, USA*, vol. 7259, pp. 725911-1–12, doi: 10.1117/12.812407.
- [9] J. Peng, P. Hu, F. Lu, Z. Peng, D. Kong, and H. Zhang, "3D liver segmentation using multiple region appearances and graph cuts," *Med. Phys.*, 2015, doi: 10.1118/1.4934834.
- [10] O. N. Ucan, O. Osman, and A. M. Albora, "Görüntü işleme teknikleri ve mühendislik uygulamaları.pdf," istanbul: Nobel yayın dağıtım, 2006, pp. 63–70.
- [11] G. Ertas, H. O. Gulcur, M. Tunaci, O. Osman, and O. N. Ucan, "A preliminary study on computerized lesion localization in MR mammography using 3D nMITR maps, multilayer cellular neural networks, and fuzzy c-partitioning," *Med. Phys.*, vol. 35, no. 1, pp. 195–205, 2008, doi: 10.1118/1.2805477.
- [12] Y. Zheng, X. Yang, X. Ye, and X. Lin, "Fully automatic segmentation of liver from multiphase liver CT," vol. 6512, p. 65122X, 2007, doi: 10.1117/12.709711.

- [13] K. Mala, V. Sadasivam, and S. Alagappan, "Neural Network based Texture Analysis of Liver Tumor from Computed Tomography Images," vol. 2, no. 1, pp. 12–19, 2008.
- [14] M. G. Linguraru, J. K. Sandberg, Z. Li, F. Shah, and R. M. Summers, "Automated segmentation and quantification of liver and spleen from CT images using normalized probabilistic atlases and enhancement estimation," *Med. Phys.*, vol. 37, no. 2, pp. 771–783, 2010, doi: 10.1118/1.3284530.
- [15] A. Arı and D. Hanbay, "Bölgesel evrişimsel sinir ağları tabanlı mr görüntülerinde tümör tespiti," *J. Fac. Eng. Archit. Gazi Univ.*, 2018, doi: <https://doi.org/10.17341/gazimmfd.460535>.
- [16] Q. Dou, H. Chen, Y. Jin, L. Yu, J. Qin, and P. A. Heng, "3D deeply supervised network for automatic liver segmentation from CT volumes," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9901 LNCS, pp. 149–157, 2016, doi: 10.1007/978-3-319-46723-8_18.
- [17] C. Dong *et al.*, "Simultaneous Segmentation of Multiple Organs Using Random Walks," *J. Inf. Process. Soc. Japan*, vol. 24, no. 2, pp. 320–329, 2016, doi: 10.2197/ipsjjip.24.320.
- [18] C. Dong, Y. Chen, A. Hossein, L. Lin, and X. Han, "Segmentation of liver and spleen based on computational anatomy models," *Comput. Biol. Med.*, pp. 1–15, 2015, doi: 10.1016/j.compbiomed.2015.10.007.
- [19] Y. Todoroki, X. Han, Y. Iwamoto, and L. Lin, "Detection of Liver Tumor Candidates from CT Images Using Deep Convolutional Neural Networks," *Innov. Med. Healthc.*, vol. 71, pp. 1–6, 2017, doi: 10.1007/978-3-319-59397-5.
- [20] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity (reprinted from bulletin of mathematical biophysics, vol 5, pg 115-133, 1943)," *Bull. Math. Biol.*, vol. 52, no. 1--2, pp. 99–115, 1990, doi: Doi 10.1016/S0092-8240(05)80006-0.
- [21] D. O. Hebb, "Hebb," *Organ. Behav. New York Wiley , Introd. Chapter 4., "The first stage Percept. growth Assem. pp. xi- xix, 60- 78*, 1949.
- [22] B. G. Farley and W. A. Clark, "Simulation of self-organizing systems by digital computer," *IRE Trans. Inf. Theory, PGIT-4*, 76-84., 1954.
- [23] F. Rosenblatt, "The Perceptron: A Probabilistic Model For Information Storage And Organization In The Brain," *Psychol. Rev.*, vol. 65, no. 6, pp. 1–23, 1958, doi: 10.1098/rspb.1976.0087.
- [24] B. Widrow and M. E. Hoff, "Adaptive switching circuits," *WEST-CON Convention Record Part IV*, pp. 96–104, 1960.
- [25] T. Kohonen, "Self-Organized Formation of Topologically Correct Feature Maps," *Biol. Cybern.*, vol. 69, pp. 59–69, 1982.
- [26] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proc. Natl. Acad. Sci.*, vol. 79, no. April, pp. 2554–2558, 1982,

doi: 10.1201/9780429500459.

- [27] J. J. Hopfield, “Neurons with graded response have collective computational properties like those of two-state neurons (associative memory/neural network/stability/action potentials),” *Proc. Natl. Acad. Sci.*, vol. 81, no. May, pp. 3088–3092, 1984, [Online]. Available: <http://www.pnas.org/content/pnas/81/10/3088.full.pdf>.
- [28] R. D. E., H. G. E, and and Williams R. J., “Learning representations by back-propagating errors,” *Nature*, vol. 323, pp. 533–536, 1986.
- [29] I. N. Aizenberg, N. N. Aizenberg, J. Vandewalle, I. N. Aizenberg, N. N. Aizenberg, and J. Vandewalle, “Multiple-Valued Threshold Logic and Multi-Valued Neurons,” *Multi-Valued Univers. Bin. Neurons*, no. 1971, pp. 25–80, 2000, doi: 10.1007/978-1-4757-3115-6_2.
- [30] A. Şeker, B. Diri, and H. Hüseyin, “Derin Öğrenme Yöntemleri ve Uygulamaları Hakkında Bir İnceleme A Review about Deep Learning Methods and Applications,” vol. 3, no. 3, pp. 47–64, 2017, doi: 10.1155/2017/1419676 [doi].
- [31] X. Gao, W. Li, M. Loomes, and L. Wang, “A fused deep learning architecture for viewpoint classification of echocardiography,” *Inf. Fusion*, vol. 36, pp. 103–113, 2017, doi: 10.1016/j.inffus.2016.11.007.
- [32] Ö. İnik *et al.*, “Derin Öğrenme ve Görüntü Analizinde Kullanılan Derin Öğrenme Modelleri Deep Learning and Deep Learning Models Used in Image Analysis,” *GBAD) Gaziosmanpaşa J. Sci. Res.*, vol. ISSN, pp. 2146–8168, 2017.
- [33] S. Muruganandham, “Semantic Segmentation of Satellite Images using Deep Learning,” *Czech Tech. Univ. Prague Lulea Univ. Technol.*, pp. 1–94, 2016, doi: 10.1007/s00211-012-0485-5.
- [34] L. Deng, “Deep Learning: Methods and Applications,” *Found. Trends® Signal Process.*, vol. 7, no. 3–4, pp. 197–387, 2014, doi: 10.1561/20000000039.
- [35] P. Anand, “Architecture of AlexNet and its current use.” <https://iq.opengenus.org/architecture-and-use-of-alexnet/> (accessed Nov. 25, 2019).
- [36] SKK Saikia, “CNN-AlexNet,” 2018. .
- [37] C. Szegedy *et al.*, “Going deeper with convolutions,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 07-12-June, pp. 1–9, 2015, doi: 10.1109/CVPR.2015.7298594.
- [38] J. M. Czum, “Dive Into Deep Learning,” *J. Am. Coll. Radiol.*, vol. 17, no. 5, pp. 637–638, 2020, doi: 10.1016/j.jacr.2020.02.005.
- [39] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–14, 2015.
- [40] M. Livne *et al.*, “A U-net deep learning framework for high performance vessel

- segmentation in patients with cerebrovascular disease,” *Frontiers in Neuroscience*, vol. 13, no. FEB. 2019, doi: 10.3389/fnins.2019.00097.
- [41] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9351, pp. 234–241, 2015, doi: 10.1007/978-3-319-24574-4_28.
 - [42] U. of Freiburg, “<https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net/>,” 2020. .
 - [43] M. Sewak, R. Karim, and P. Pujari, *Practical Convolutional Neural Networks*. 2018.
 - [44] S. Wu, S. Zhong, and Y. Liu, “Deep Residual Learning for Image Recognition,” *Multimed. Tools Appl.*, pp. 1–17, 2017, doi: 10.1007/s11042-017-4440-4.
 - [45] S. Schmatz, “In a neural network (unsure if it is keras specific) what is the difference between batch-size and batch-training? When would I need both?,” 2017. <https://www.quora.com/In-a-neural-network-unsure-if-it-is-keras-specific-what-is-the-difference-between-batch-size-and-batch-training-When-would-I-need-both>.
 - [46] A. Gülcü and Z. Kuş, “Konvolüsyonel Sinir Ağlarında Hiper-Parametre Optimizasyonu Yöntemlerinin İncelenmesi,” *Gazi Üniversitesi Fen Bilim. Derg. Part C Tasarım ve Teknol.*, vol. 7, no. 2, pp. 503–522, 2019, doi: 10.29109/gujsc.514483.
 - [47] A. C. ÇINAR, “Optimizasyon Nedir?,” 2020. <https://www.ahmetcevahircinar.com.tr/2016/03/09/optimizasyon-nedir/>.
 - [48] N. E. C. L. America and P. Nj, “Large-Scale Machine Learning with Stochastic Gradient Descent,” 2010, doi: 10.1007/978-3-7908-2604-3.
 - [49] J. Duchi, “Adaptive Subgradient Methods for Online Learning and Stochastic Optimization Adaptive Subgradient Methods for Online Learning and Stochastic Optimization,” no. March, 2015.
 - [50] M. D. Zeiler, “Adadelata: An adaptive learning rate method,” *arXiv Prepr. arXiv1212.5701*, 2012.
 - [51] D. P. Kingma and J. L. Ba, “Adam: A Method for Stochastic Optimization,” pp. 1–15, 2015.
 - [52] S. Ruder, “An overview of gradient descent optimization,” pp. 1–14, 2016.
 - [53] L. Deng *et al.*, “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups,” no. November, pp. 82–97, 2012.
 - [54] G. Hinton and T. Tieleman, “Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitud.COURSERA: Neural Networks for Machine Learning,” 2012.
 - [55] G. E. Hinton and V. Nair, “Rectified Linear Units Improve Restricted Boltzmann Machines,” *Proc. 27th Int. Conf. Mach. Learn.*, no. 3, pp. 807–814, 2010.

- [56] X. Glorot, A. Bordes, and Y. Bengio, "Deep Sparse Rectifier Neural Networks," *JMLR W&CP Proc. Fourteenth Int. Conf. Artif. Intell. Stat. (AISTATS 2011)*, vol. 15, pp. 315–323, 2011.
- [57] S. H. Zhen *et al.*, "Deep Learning for Accurate Diagnosis of Liver Tumor Based on Magnetic Resonance Imaging and Clinical Data," *Front. Oncol.*, vol. 10, no. May, pp. 1–14, 2020, doi: 10.3389/fonc.2020.00680.
- [58] G. Zeng, Y. He, Z. Yu, X. Yang, R. Yang, and L. Zhang, "Preparation of novel high copper ions removal membranes by embedding organosilane-functionalized multi-walled carbon nanotube," *J. Chem. Technol. Biotechnol.*, vol. 91, no. 8, pp. 2322–2330, 2016, doi: 10.1002/jctb.4820.
- [59] L. J. Halawa, A. Wibowo, and F. Ernawan, "Face Recognition Using Faster R-CNN with Inception-V2 Architecture for CCTV Camera," *ICICOS 2019 - 3rd Int. Conf. Informatics Comput. Sci. Accel. Informatics Comput. Res. Smarter Soc. Era Ind. 4.0, Proc.*, pp. 2–7, 2019, doi: 10.1109/ICICoS48119.2019.8982383.
- [60] C. Lin, L. Li, W. Luo, K. C. P. Wang, and J. Guo, "Transfer learning based traffic sign recognition using inception-v3 model," *Period. Polytech. Transp. Eng.*, vol. 47, no. 3, pp. 242–250, 2019, doi: 10.3311/PPtr.11480.
- [61] X. Xia, C. Xu, and B. Nan, "Inception-v3 for flower classification," *2017 2nd Int. Conf. Image, Vis. Comput. ICIVC 2017*, pp. 783–787, 2017, doi: 10.1109/ICIVC.2017.7984661.
- [62] M. Längkvist, L. Karlsson, and A. Loutfi, "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning," *Pattern Recognit. Lett.*, vol. 42, no. 1, pp. 11–24, 2016, doi: 10.48550/arXiv.1602.07261.
- [63] V. Sangeetha and K. J. R. Prasad, "Deep Residual Learning for Image Recognition Kaiming carbaldehydes," *Indian J. Chem. - Sect. B Org. Med. Chem.*, vol. 45, no. 8, pp. 1951–1954, 2006, doi: 10.1002/chin.200650130.
- [64] Y. Huang, X. Li, C. Yan, L. Liu, and H. Dai, "MIRD-Net for Medical Image Segmentation," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 12085 LNAI, pp. 207–219, 2020, doi: 10.1007/978-3-030-47436-2_16.
- [65] O. Oktay *et al.*, "Attention U-Net: Learning Where to Look for the Pancreas," no. Midl, 2018, [Online]. Available: <http://arxiv.org/abs/1804.03999>.
- [66] C. Guo, M. Szemenyei, Y. Hu, W. Wang, W. Zhou, and Y. Yi, "Channel Attention Residual U-Net for Retinal Vessel Segmentation," pp. 1185–1189, 2021, doi: 10.1109/icassp39728.2021.9414282.
- [67] K. Kubota *et al.*, "Measurement of liver volume and hepatic functional reserve as a guide to decision-making in resectional surgery for hepatic tumors," *Hepatology*, vol. 26, no. 5, pp. 1176–1181, 1997, doi: 10.1002/hep.510260514.
- [68] M. Sevi and I. Aydin, "COVID-19 Detection Using Deep Learning Methods," *2020 Int.*

- Conf. Data Anal. Bus. Ind. W. Towar. a Sustain. Econ. ICDABI 2020*, 2020, doi: 10.1109/ICDABI51230.2020.9325626.
- [69] B. Stasiak, P. Tarasiuk, I. Michalska, and A. Tomczyk, "Application of convolutional neural networks with anatomical knowledge for brain MRI analysis in MS patients," *Bull. Polish Acad. Sci. Tech. Sci.*, vol. 66, no. 6, pp. 857–868, 2018, doi: 10.24425/bpas.2018.125933.
 - [70] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9351, no. Cvd, pp. 12–20, 2015, doi: 10.1007/978-3-319-24574-4.
 - [71] V. Thi, T. Vi, A. Oh, G. Lee, H. Yang, and S. Kim, "Automatic Extraction of Liver Region from Medical Images by Using an MFUnet," vol. 9, no. 3, pp. 59–70, 2020, doi: 10.30693/SMJ.2020.9.3.59.
 - [72] X. Li, H. Chen, X. Qi, Q. Dou, C. W. Fu, and P. A. Heng, "H-DenseUNet: Hybrid Densely Connected UNet for Liver and Tumor Segmentation from CT Volumes," *IEEE Trans. Med. Imaging*, vol. 37, no. 12, pp. 2663–2674, 2018, doi: 10.1109/TMI.2018.2845918.
 - [73] Y. Yuan, M. Buckstein, and Y. Lo, "Automatic Liver and Tumor Segmentation Using Hierarchical Convolutional-Deconvolutional Neural Networks with Jaccard Distance," *Med. Phys.*, vol. 45, no. 6, pp. E516–E516, 2018.
 - [74] G. K. Mourya, S. Paul, A. Handique, U. Baid, P. V. Dutande, and S. N. Talbar, "Modified U-Net for fully automatic liver segmentation from abdominal CT-image," *Int. J. Biomed. Eng. Technol.*, vol. 40, pp. 1–17, 2022, doi: 10.1504/IJBET.2022.125099.
 - [75] Q. Li *et al.*, "CC-DenseUNet: Densely Connected U-Net with Criss-Cross Attention for Liver and Tumor Segmentation in CT Volumes," *Proc. - 2021 IEEE Int. Conf. Bioinforma. Biomed. BIBM 2021*, pp. 966–971, 2021, doi: 10.1109/BIBM52615.2021.9669823.
 - [76] O. I. Alirr, "Deep learning and level set approach for liver and tumor segmentation from CT scans," *J. Appl. Clin. Med. Phys.*, vol. 21, no. 10, pp. 200–209, 2020, doi: 10.1002/acm2.13003.
 - [77] P. F. Christ *et al.*, "Automatic Liver and Tumor Segmentation of CT and MRI Volumes using Cascaded Fully Convolutional Neural Networks," pp. 1–20, 2017, doi: <https://doi.org/10.48550/arXiv.1702.05970>.
 - [78] L. C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 11211 LNCS, pp. 833–851, 2018, doi: 10.1007/978-3-030-01234-2_49.
 - [79] E. Vorontsov, A. Tang, C. Pal, and S. Kadoury, "Liver lesion segmentation informed by joint liver segmentation," *Proc. - Int. Symp. Biomed. Imaging*, vol. 2018-April, pp. 1332–1335, 2018, doi: 10.1109/ISBI.2018.8363817.
 - [80] L. Bi, J. Kim, A. Kumar, and D. Feng, "Automatic Liver Lesion Detection using Cascaded

Deep Residual Networks,” *Comput. Vis. Pattern Recognit.*, vol. abs/1704.0, 2017, doi: arXiv:1704.02703.



EK A

ETİK KURUL ONAYI

A.1 Çalışma İzni

Evrak Tarih ve Sayısı: 13.02.2020-E.172574



T.C.
TRAKYA ÜNİVERSİTESİ REKTÖRLÜĞÜ
Sağlık Araştırma ve Uygulama Merkezi Müdürlüğü



Sayı : 79056779-600-E.172574
Konu : Çalışma hk.

13/02/2020

Sayın Fırat ÖZCAN

İlgi : a) 04/02/2020 tarihli dilekçe,
b) 12/02/2020 tarihli ve 600-E.411005 sayılı yazı

İlgi (a) yazımıza istinaden Hastanemiz Nükleer Tıp Anabilim Dalı Başkanlığının düzenlemiş olduğu ilgi (b) yazısı ekte gönderilmiştir.

Bilgilerinizi rica ederim.

e-imzalıdır
Prof. Dr. Abdullah TAŞ
Merkez Müdürü

Ek: İlgi (b) yazı (1 adet, 1 sayfa)

Evrak Doğrulamak İçin : https://ebys.trakya.edu.tr/en/Vision/Validate_Doc.aspx?V=BE6ESHJFT



Adres: Trakya Üniversitesi Rektörlüğü Balkan Yerleşkesi Edirne 22030
Telefon: (0284) 235 27 31 Faks: (0284) 235 27 30
E-Posta: bashekim@trakya.edu.tr Elektronik Ağ: <http://tuh.trakya.edu.tr/>

Bilgi için: Neriman ÜNAL
Unvanı: Bilgisayar İşletmeni



Evrak Tarih ve Sayısı: 12/02/2020-411005



T.C.
TIP FAKÜLTESİ DEKANLIĞI
Dahili Tıp Bilimleri Bölüm Başkanlığı
Nükleer Tıp Anabilim Dalı Başkanlığı



Sayı : 64330354-600-E.411005
Konu : Çalışma hk.

12/02/2020

SAĞLIK ARAŞTIRMA VE UYGULAMA MERKEZİ MÜDÜRLÜĞÜNE

İlgi : 06/02/2020 tarihli ve 600-E.409495 sayılı yazı

Kırklareli Üniversitesi Teknoloji Fakültesi Mekatronik Mühendisliği bölümünde öğretim görevlisi kadrosunda görev yapmakta olup; aynı zamanda Altınbaş Üniversitesinde doktora eğitimi görmekte olan Fırat ÖZCAN'ın bölümümüzde yapmak istediği doktora tez çalışmasının Prof.Dr.Gülşay DURMUŞ ALTUN danışmanlığında yürütülmesi Anabilim Dalımız kürsü kurulunda katılanların oy birliği ile uygun bulunmuştur.

Gereği hususunda bilgilerinize arz ederim.

e-imzalıdır
Prof. Dr. Funda ÜSTÜN
Anabilim Dalı Başkanı

Adres:Trakya Üniversitesi Rektörlüğü Tıp Fakültesi Dekanlığı Balkan Yerleşkesi 22030 Edirne

Bilgi için: Nurşen BAYRAKTAR

Telefon:2842357653 Faks:2842357652

Unvanı: Sekreter

E-Posta:dekanlik@trakya.edu.tr Elektronik Ağ:http://tipfak.trakya.edu.tr/



A.1.1 Etik Kurul Onayı

Evrak Tarih ve Sayısı: 09/11/2020-473049



T.C.
TRAKYA ÜNİVERSİTESİ REKTÖRLÜĞÜ
Tıp Fakültesi Dekanlığı



Sayı : 22121724-050.04.02-E.473049
Konu : Davetler ve Gündemler

09/11/2020

Sayın Prof. Dr. Gülay DURMUŞ ALTUN

Trakya Üniversitesi Tıp Fakültesi Bilimsel Araştırmalar Etik Kurulu'nun 10.08.2020 tarihli toplantısında almış olduğu, yürütücüsü olduğunuz; "**Derin Öğrenme ile Üç Bpyutlu (3B) Karaciğer Tümör Tespiti**" adlı TÜTF-BAEK 2020/254 protokol no.lu Doktora Öğrencisi Fırat ÖZCAN'ın tez çalışmasına ait 05 no.lu karar ekte gönderilmiştir.
Bilgilerinizi rica ederim.

e-imzalıdır

Prof. Dr. Mustafa İNAN
Dekan a.
Dekan Yardımcısı V.

Adres:Trakya Üniversitesi Rektörlüğü Tıp Fakültesi Dekanlığı Balkan Yerleşkesi 22030 Edirne

Telefon:2842357653 Faks:2842357652

E-Posta:dekanlik@trakya.edu.tr Elektronik Ağ:http://tipfak.trakya.edu.tr/

Bilgi için: Yeşim ERDOĞAN
Unvanı: Bilgisayar İşletmeni



BİLİMSEL ARAŞTIRMALAR ETİK KURULU Edirne, Türkiye

ARAŞTIRMA BAŞVURUSU ONAYIBAŞVURU BİLGİLERİ	PROTOKOL KODU	TÜTF-BAEK 2020/254	
	PROTOKOL ADI	Derin Öğrenme ile Üç Bpyutlu (3B) Karaciğer Tümör Tespiti	
	SORUMLU ARAŞTIRICI ÜNVANI / ADI	Prof. Dr. Gülay ALTUN	
	ARAŞTIRMA MERKEZİ		
	DESTEKLEYİCİ		
	ARAŞTIRMAYA KATILAN MERKEZLER	Tek Merkez Ulusal	Çok Merkez Uluslararası
KARAR BİLGİLERİ	Karar No: 12/05	Tarih: 10.08.2020	
	Fakültemiz Nükleer Tıp Anabilim Dalı Öğretim Üyesi Prof. Dr. Gülay ALTUN'un sorumluluğunda yapılması planlanan ve yukarıda başvuru bilgileri verilen Doktora Öğrencisi Fırat ÖZCAN'ın tez çalışmasının araştırma başvuru dosyası ve ilgili belgeler araştırmanın gerekçe, amaç, yaklaşım ve yöntemleri dikkate alınarak incelenmiş; araştırmaya ilişkin giderlerin gönüllüye ve/veya bağlı bulunduğu sosyal güvenlik kurumuna ödenilmediği koşullarda ve veri toplanacak yerlerden gerekli izinler alındıktan sonra gerçekleştirilmesinde etik bilimsel standartlar açısından sakınca bulunmadığına mevcudun oy birliği ile karar verilmiştir.		
ETİK KURUL BİLGİLERİ			
ÇALIŞMA ESASI	Helsinki Bildirgesi, İyi Klinik Uygulamalar Kılavuzu, TÜTF-BAEK Yönergesi		

ÜYELER

Ünvan/Ad/ Soyadı	Uzmanlık Dalı	Kurumu	Cinsiyeti	İlişki(*)	Katılım (**)	İmza
Dr. Öğr. Üyesi Fatma Gülşüm ONAL Başkan	Tıp Tarihi ve Etik	T.Ü.T.F. Tıp Tarihi ve Etik A.D.	K	E H	E H	
Doç. Dr. Hakan GÜRKAN Başkan Yardımcısı	Tıbbi Genetik	T.Ü.T.F. Tıbbi Genetik A.D.	E	E H	E H	
Dr. Öğr. Üyesi Ruhan Deniz TOPUZ Üye	Tıbbi Farmakoloji	T.Ü.T.F. Tıbbi Farmakoloji A.D.	K	E H	E H	Mazeretli
Doç. Dr. F. Nesrin TURAN Üye	Biyoistatistik	T.Ü.T.F. Biyoistatistik A.D.	K	E H	E H	
Doç. Dr. Rugül KÖSE ÇINAR Üye	Ruh Sağlığı ve Hastalıkları	T.Ü.T.F. Ruh Sağ. ve Has. A.D.	K	E H	E H	
Prof. Dr. Hasan ÜMIT Üye	İç Hastalıkları	T.Ü.T.F. İç Hastalıkları A.D.	E	E H	E H	
Dr. Öğr. Üyesi Oktay KAYA Üye	Fizyoloji	T.Ü.T.F. Fizyoloji A.D.	E	E H	E H	
Prof. Dr. Galip EKUKLU Üye	Halk Sağlığı	T.Ü.T.F. Halk Sağlığı A.D.	E	E H	E H	
Prof. Dr. Filiz TÖTÜCÜLER KÖKENLİ Üye	Çocuk Sağlığı ve Hastalıkları	T.Ü.T.F. Çocuk Sağlığı ve Hastalıkları A.D.	K	E H	E H	
Dr. Öğr. Üyesi Fatma Gülşüm ONAL Üye	Tıp Tarihi ve Etik	T.Ü.T.F. Tıp Tarihi ve Etik A.D.	K	E H	E H	Mazeretli
Öğr. Gör. Dr. Sinan ATEŞ Üye	Kadın Hastalıkları ve Doğum	T.Ü.T.F. Kadın Hastalıkları ve Doğum A.D.	E	E H	E H	
Prof. Dr. Sevtap HEKİMOĞLU ŞAHİN Üye	Anestezi ve Reanimasyon	T.Ü.T.F. Anestezi ve Reanimasyon A.D.	K	E H	E H	Mazeretli
Prof. Dr. Atakan SEZER Üye	Genel Cerrahi	T.Ü.T.F. Genel Cerrahi A.D.	E	E H	E H	Mazeretli
Avukat Emine NURLU Üye		T.C. Rektörlüğü	K	E H	E H	Mazeretli
Emekli Öğretmen Sinan SEÇKİN Üye		Serbest Üye	E	E H	E H	

*Araştırma ile ilişki
**Toplantıda Bulunma

Prof. Dr. Ahmet TEZEL