
GAZİ ÜNİVERSİTESİ
Gazi
1926

DERİN ÖĞRENME YÖNTEMİ İLE BİLGİSAYARLI TOMOGRAFİ GÖRÜNTÜLERİNDEN KARACİĞER TÜMÖRÜ TESPİTİ

Hanife VEÇEYİŞ

YÜKSEK LİSANS TEZİ
ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ ANA BİLİM DALI

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

OCAK 2025

ETİK BEYAN

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

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

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

İmza

Hanife VEÇEYİŞ

07/01/2025

DERİN ÖĞRENME YÖNTEMİ İLE BİLGİSAYARLI TOMOGRAFİ GÖRÜNTÜLERİNDEN KARACİĞER TÜMÖRÜ TESPİTİ

(Yüksek Lisans Tezi)

Hanife VEÇEYİŞ

GAZİ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

Ocak 2025

ÖZET

Bu çalışmada, karaciğer tümörlerinin tıbbi görüntülerde bölütlenmesi üzerine derin öğrenme tabanlı bir model geliştirilmiştir. Karaciğer tümör bölütlemesi, karaciğer kanserinin tedavi planlamasında, cerrahi müdahalelerin hassasiyetinde, hastalığın evrelenmesinde ve patolojik teşhisin doğruluğunda hayati önem taşımaktadır. Bu bağlamda, derin öğrenme modellerinin yaygın kullanımı, etiketlenmiş veri miktarının artması ve görüntü işleme teknolojilerindeki gelişmeler sayesinde bu görevde önemli ilerlemeler kaydedilmiştir. Bununla birlikte, karaciğer tümörü bölütlemesi, birçok teknik zorluğu da beraberinde getirir. Tıbbi görüntülerde düşük kontrast, tümör sınırlarının belirsizliği, karaciğerin diğer karın organlarına yakın konumu ve benzer doku özellikleri, bu zorluklardan bazılarıdır. Ayrıca, karaciğer yapısının kişiden kişiye değişiklik göstermesi, bu tür bölütleme görevlerinin doğruluğunu etkileyen faktörler arasındadır. Bu faktörler, manuel bölütleme süreçlerini zorlaştırırken, otomatik derin öğrenme tabanlı modellerin geliştirilmesini de zor hale getirmektedir. Tıbbi veri etiketleme süreci genellikle uzmanlık gerektirdiğinden, yeterli miktarda etiketlenmiş veri bulunmaması ve hasta mahremiyeti gibi kısıtlamalar, model performansını artırmada karşılaşılan önemli engellerdendir. Bu çalışmada, sınırlı etiketli veri üzerinde eğitilen ve ortalama bir bölütleme başarımı sağlayan etkili bir derin öğrenme modeli sunulmaktadır. Bu sayede, sağlık çalışanlarına klinik karar süreçlerinde destek sağlanması ve karaciğer hastalıklarının tanı ve tedavisinde etkin bir araç olarak kullanılması amaçlanmaktadır. Bu yaklaşım, klinik ortamlarda zamandan tasarruf sağlamayı, teşhis doğruluğunu artırmayı ve karaciğer tümör bölütlemesi görevinde derin öğrenme yöntemlerinin uygulanabilirliğini kanıtlamayı hedeflemektedir.

Bilim Kodu : 93418
Anahtar Kelimeler : Derin öğrenme, karaciğer, tümör, bölütleme
Sayfa Adedi : 83
Danışman : Prof. Dr. Hasan Şakir BİLGE

LIVER TUMOR DETECTION FROM COMPUTED TOMOGRAPHY IMAGES WITH DEEP LEARNING METHOD

(M. Sc. Thesis)

Hanife VEÇEYİŞ

GAZİ UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

January 2025

ABSTRACT

In this study, a deep learning-based model has been developed for liver tumor segmentation in medical images. Liver tumor segmentation is critical for treatment planning of liver cancer, precision in surgical interventions, staging of the disease, and accuracy in pathological diagnosis. In this context, significant progress has been achieved in this task with the widespread use of deep learning models, the increase in the amount of labeled data, and advancements in image processing technology. However, liver tumor segmentation also presents various technical challenges. These challenges include low contrast in medical images, unclear tumor boundaries, the proximity of the liver to other abdominal organs, and similar tissue characteristics. Additionally, variations in liver structure between individuals impact the accuracy of segmentation tasks. While these factors complicate manual segmentation processes, they also make developing automated deep learning-based models challenging. Labeling medical data typically requires expertise, so the lack of sufficient labeled data and constraints such as patient privacy are significant obstacles to improve model performance. In this study, an effective deep learning model with average segmentation performance trained on limited labeled data is presented. Thus, the aim is to support healthcare professionals in clinical decision-making processes and provide an effective tool for diagnosing and treating liver diseases.

Science Code	:	93418
Key Words	:	Deep learning, liver, tumor, segmentation
Page Number	:	83
Supervisor	:	Prof. Dr. Hasan Şakir BİLGE

TEŞEKKÜR

Bu çalışmasının hazırlanması sürecinde bilgisi, tecrübesi ve sabrı ile her türlü desteği sunan değerli danışmanım Prof. Dr. Hasan Şakir BİLGE'ye minnet ve teşekkürlerimi sunarım.

Ev, profesyonel iş hayatı ve okul dengesini kurmamda her zaman destek olan, araştırma sürecinde beni motive eden, cesaretlendiren değerli çocuklarım Serra VEÇEYİŞ, Azra VEÇEYİŞ ve pek değerli eşim Gökhan VEÇEYİŞ'e sevgi ve şükranlarımı sunmak isterim.

İÇİNDEKİLER

Sayfa

ÖZET iv

ABSTRACT v

TEŞEKKÜR vi

İÇİNDEKİLER vii

ÇİZELGELERİN LİSTESİ ix

ŞEKİLLERİN LİSTESİ x

SİMGELER VE KISALTMALAR xiii

1. GİRİŞ 1

2. TEORİK BİLGİLER 9

3. LİTERATÜR TARAMASI 11

4. KAPSAM, YÖNTEM VE ARAÇLAR 15

4.1. Yapay Zekâ, Makine Öğrenmesi ve Derin Öğrenme 15

4.2. Yapay Sinir Ağları (YSA) 16

4.3. Derin Sinir Ağları (DSA) 19

4.4. Evrişimli Sinir Ağları (ESA) 24

4.5. UNet mimarisi 29

4.6. Karmaşıklık Matrisi 31

4.7. ROC Analizi 34

5. YÖNTEM VE ÇALIŞMA 39

5.1. Kullanılan Materyaller 39

5.2. Veri Setinin Seçilmesi ve İncelenmesi 41

5.3. Veri Önişleme 46

5.4. Derin Öğrenme Modeli ve Parametreler 53

Sayfa

6. DENEYSEL SONUÇLAR 61

6.1. Öğrenme Hızı = 0,00001 61

6.2. Öğrenme Hızı = 0,0001 73

7. SONUÇ VE ÖNERİLER 77

KAYNAKLAR 78

ÖZGEÇMİŞ 83

ÇİZELGELERİN LİSTESİ

Çizelge **Sayfa**

Çizelge 3.1. Karaciğer ve tümör bölütlemesi çalışmalarından elde edilen bazı Dice Benzerlik Katsayısı değerleri 14

Çizelge 5.1. Karaciğer ve karaciğer tümörü halka açık veri setlerinin karşılaştırması .. 42

Çizelge 5.2. Veri seti klasör yapısı 47

Çizelge 5.3. Unet Ağ Modeli Parametreleri..... 53

Çizelge 6.1. Lr = 0.00001 için bir görüntü dilimi çıktılarından hesaplanan FPR-TPR değerleri 72

Çizelge 6.2. Lr = 0.0001 için bir görüntü dilimi çıktılarından hesaplanan FPR-TPR değerleri 75

ŞEKİLLERİN LİSTESİ

Şekil **Sayfa**

Şekil 1.1. Karaciğerin yapısı 2

Şekil 1.2. IARC-Tüm bireylerden ve yaş gruplarından 2020 yılındaki yeni kanser vaka sayıları 3

Şekil 1.3. IARC-Tüm bireylerden ve yaş gruplarından 2020 yılındaki kanserden ölüm sayıları 4

Şekil 4.1. Biyolojik ve Yapay Sinir Hücrelerinin Yapısı 17

Şekil 4.2. Tek Katmanlı Yapay Sinir Ağı 19

Şekil 4.3. DSA 20

Şekil 4.4. DSA Çalışma Şekli 23

Şekil 4.5. ESA Modeli Çalışma Prensibi 24

Şekil 4.6. ESA Katmanları 24

Şekil 4.7. Evrişim İşlemi 25

Şekil 4.8. Relu Aktivasyon Fonksiyornu 26

Şekil 4.9. Maksimum Havuzlama 26

Şekil 4.10. Ortalama Havuzlama 27

Şekil 4.11. Düzleştirme Katmanı 27

Şekil 4.12. Tam Bağlantı İşlemi 28

Şekil 4.13. (a) Standart Esa Modeli, (b) Seyreltme Katmanı Sonrası ESA Modeli 29

Şekil 4.14. UNet Mimarisi 31

Şekil 4.15. İki Sınıflı Karmaşıklık Matrisi 32

Şekil 4.16. Eşik değer – Karmaşıklık Matrisi ilişkisi 35

Şekil 4.17. Eşik Değer seçimi 36

Şekil 4.18. ROC eğrisi gösterimleri 36

Şekil 4.19. ROC eğrisi ve sınıflandırma performansı ilişkisi 37

Şekil 4.20. AUC ve sınıflandırma performansı ilişkisi 38

Şekil **Sayfa**

Şekil 5.1. LiTS Veri Seti Lezyon Sayısı ve Örnek Sayısı Histogram Grafiği 43

Şekil 5.2. LiTS veri seti dosya yapısı 44

Şekil 5.3. Sıfır (0) etiketli karın içi diğer dokular..... 45

Şekil 5.4. Bir (1) etiketli karaciğer dokuları 45

Şekil 5.5. İki (2) etiketli tümörlü dokular 46

Şekil 5.6. NIfTI dosyalarından DICOM serisi oluşturma..... 47

Şekil 5.7. Dönüşüm fonksiyonu uygulanmış veriler..... 53

Şekil 6.1. Lr=0,00001 için Dice Benzerlik Katsayısı-Epok Değişim Grafiği..... 61

Şekil 6.2. Lr=0,00001 için Dice Kaybı-Epok Değişim Grafiği 61

Şekil 6.3. Test verileri ile elde edilen tahminler 63

Şekil 6.4. Eşik değer 0.0 için model çıktısı..... 66

Şekil 6.5. Eşik değer 0.1 için model çıktısı..... 67

Şekil 6.6. Eşik değer 0.0 için model çıktısı DN pikseller 67

Şekil 6.7. Eşik değer 0.2 için model çıktısı..... 68

Şekil 6.8. Eşik değer 0.3 için model çıktısı..... 68

Şekil 6.9. Eşik değer 0.4 için model çıktısı..... 69

Şekil 6.10. Eşik değer 0.5 için model çıktısı..... 69

Şekil 6.11. Eşik değer 0.6 için model çıktısı..... 70

Şekil 6.12. Eşik değer 0.7 için model çıktısı..... 70

Şekil 6.13. Eşik değer 0.8 için model çıktısı..... 71

Şekil 6.14. Eşik değer 0.9 için model çıktısı..... 71

Şekil 6.15. Eşik değer 1.0 için model çıktısı..... 72

Şekil 6.16. Lr=0,00001 için ROC Eğrisi ROC Eğrisi..... 73

Şekil 6.17. Lr=0,0001 için Dice Benzerlik Katsayısı-Epok Değişim Grafiği 74

Şekil **Sayfa**

Şekil 6.18. Lr=0,0001 için Dice Kaybı-Epok Değişim Grafiği 75

Şekil 6.19. Lr=0,0001 için ROC Eğrisi 76

SİMGELER VE KISALTMALAR

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

Kısaltmalar	Açıklamalar
AUC	Area Under The Curve
BT	Bilgisayarlı Tomografi
CT	Computed Tomograpy
DICOM	Digital Imaging and Communications in Medicine
DP	Doğru Pozitif
DN	Doğru Negatif
ESA	Evrişimli Sinir Ağı
FPR	False Positive Rate
IARC	The International Agency for Research on Cancer
LiTS	Liver Tumor Segmentation
MICCAI	International Conference on Medical Image Computing and Computer Asisted Intervention
MONAI	Medical Open Network for Artificial Intelligence
NIfTI	Neuroimaging Informatics Technology Initiative
Relu	Rectified Linear Unit
ROC	A Receiver Operator Characteristic
TPR	True Positive Rate
YSA	Yapay Sinir Ağları
YP	Yanlış Pozitif
YN	Yanlış Negatif

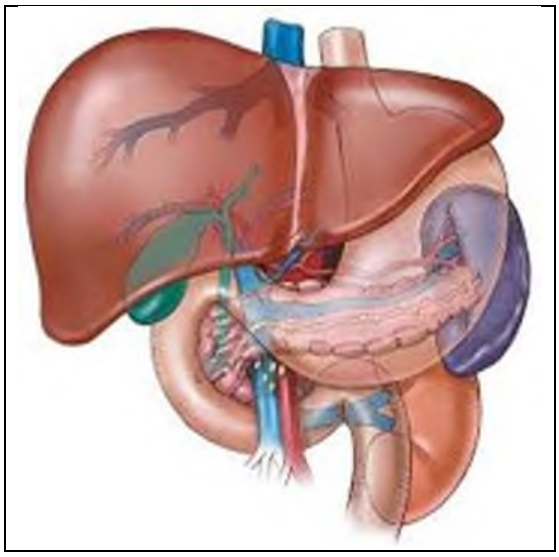
1. GİRİŞ

Karaciğer kanserinin ilk evrelerde fark edilerek gerekli tedbirlerin alınması hastalığın tedavisi aşamasında çok değerlidir. Tıbbi görüntüleme tekniklerinin gelişmesine paralel olarak erken teşhis imkânı her geçen gün artmaktadır. Ancak teşhis ve tedavide tıbbi görüntüleme teknikleri kullanılırken bunun yanında bu görüntüleri yorumlayıp kullanacak kısıtlı uzman personel faktörü hala bir kısıt olarak yer tutmaktadır. Bu sebeple teşhis ve tedavi planının önemli bir parçasını oluşturan görüntülerin yorumlanması işleminin tam otomatik olarak yapılması halen gelişime açık bir alan olarak görülmektedir. Otomatik organ ve nodül bölütleme çalışmasının yeterli başarıya ulaşması sayesinde uzman personel ihtiyacından kaynaklanan gecikme ve aksamaların önüne geçilebileceği değerlendirilmektedir.

Bu çalışmada tıbbi görüntülerin işlenmesi özelinde oldukça zengin bir kaynak olan ve birçok kullanım kolaylığı sunan MONAI (Medical Open Network for Artificial Intelligence) derin öğrenme kütüphanesi kullanılarak karaciğer tümörlerinin tam otomatik (insan faktörü olmadan) bölütlenmesine yönelik bir modelleme gerçekleştirilmiş, yöntem ve çalışma adımları açıklanmış, deneysel sonuçlar paylaşılmış ve bu sonuçların değerlendirmesi yapılmıştır.

Karaciğer

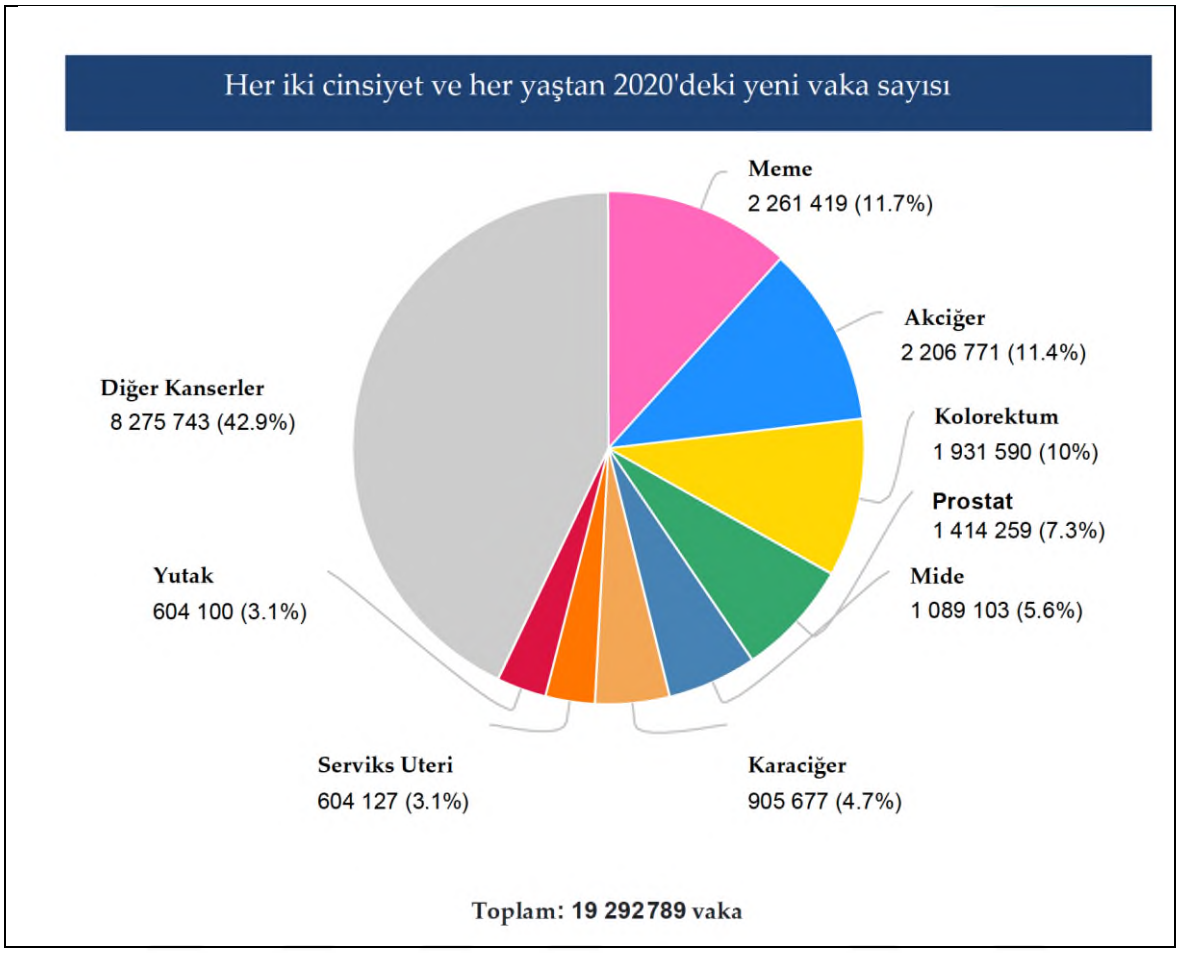
Karaciğer karın boşluğunda ve bu bölgenin sağ üst tarafında yer alan ve diğer iç organlarla yakın ilişki içinde bulunan; 1,4-1,7 kg ağırlığında, pembemsi-kahverengi, yumuşak, üçgen şekli ile vücudumuzun en büyük iç organıdır. Başlıca görevleri metabolizma, safra üretimi, detoksifikasyon, protein sentezi ve depo olmak üzere beş başlık altında toplanabilmektedir. Vücuttaki bu kritik görevlerinden de anlaşılacağı üzere karaciğer sağlığı yaşamsal öneme sahiptir. Bu sebeple karaciğeri etkileyen hastalıkların erken aşamada teşhisi ve tedavi planının hızlı ve etkili biçimde uygulanmaya başlanması hastaların yaşam süresi ve kalitesi üzerinde belirleyici rol oynamaktadır. Şekil 1.1.'de karaciğerin yapısı görülmektedir. [1]



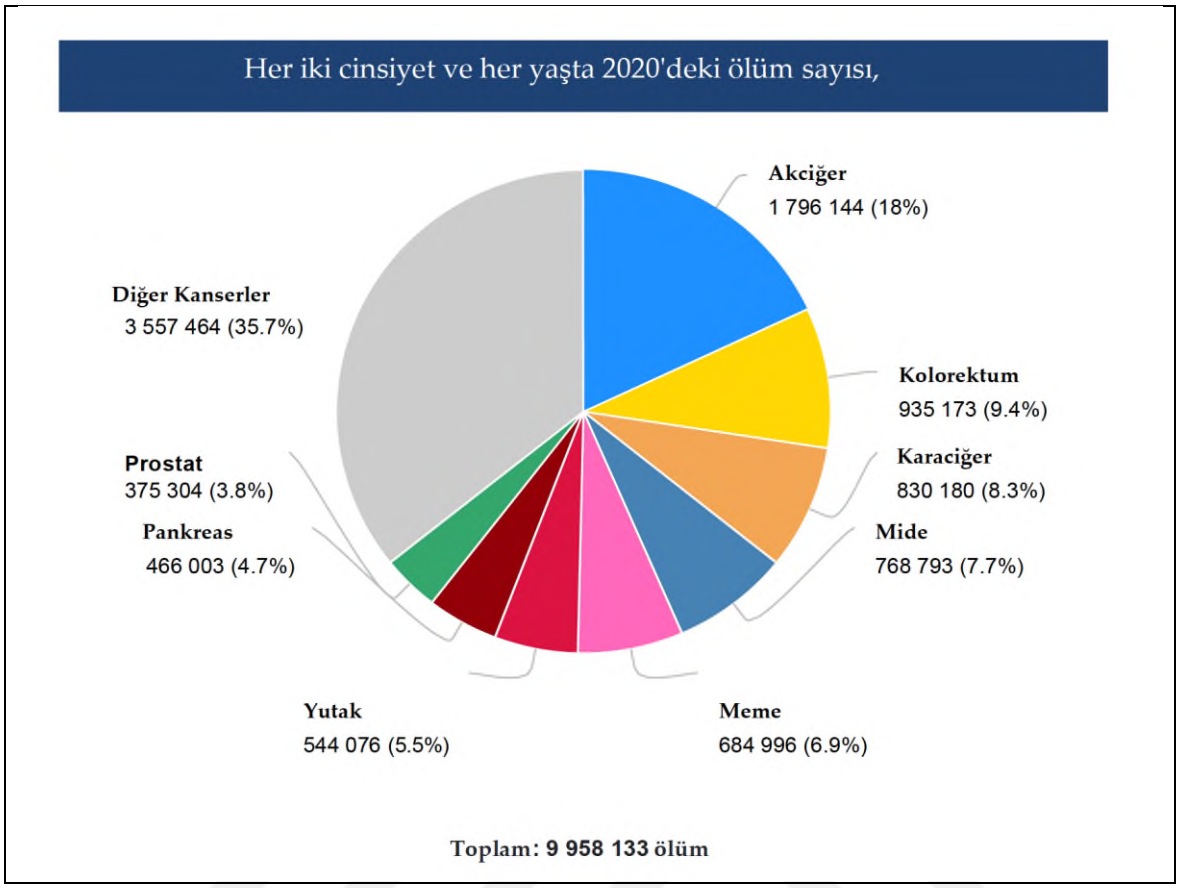
Şekil 1.1. Karaciğerin yapısı [1]

Karaciğer kanserleri

Dünya Sağlık Örgütüne (DSÖ) (World Health Organization-WHO) bağlı bir kuruluş olan Uluslararası Kanser Araştırma Ajansının (IARC-The International Agency for Research on Cancer) 2020 yılında yaptığı dünya kanser istatistikleri raporuna göre dünya çapında 2020 yılında tahmini olarak 19,3 milyon adet yeni kanser vakası (melanom dışı cilt kanseri hariç 18,1 milyon) görülmüş ve yaklaşık olarak 10,0 milyon adet kansere dayalı ölüm (melanom dışı cilt kanseri hariç 9,9 milyon) meydana gelmiştir. 2020 yılında 19,3 milyon yeni kanser vakası ve 10,0 milyon kişilik ölüm şeklinde bildirilen oranlar, 2018 yılındaki 18,1 milyon yeni kanser vakası ve 9,6 milyon kişilik ölüm oranlarıyla kıyaslandığında dünya üzerindeki kanser vakalarının arttığı tahmin edilmektedir. 2020 yılı istatistiklerine göre yeni kanser vakası olarak kadın meme kanserini 2,3 milyon (%11,7), akciğer (%11,4), kolorektal (%10,0), prostat (%7,3), mide (%5,6) ve karaciğer (%4,7) kanserleri izlemiştir. Akciğer kanseri, tahmini olarak 1,8 milyon kişinin ölümüyle (%18) kanser sebepli ölümlerinin önde gelen nedeni olmaya devam etmektedir ve bunu kolorektal kanser vakalarındaki ölüm (%9,4), karaciğer kanser vakaları (%8,3), mide kanser vakaları (%7,7) ve kadın meme (%6,9) kanserlerinden ölüm izlemiştir [2].



Şekil 1.2. IARC-Tüm bireylerden ve yaş gruplarından 2020 yılı yeni kanser vakası sayıları [2]



Şekil 1.3. IARC-Tüm bireylerden ve yaş gruplarından 2020 yılı kanser ölüm sayıları [2]

Şekil 1.2. ve Şekil 1.3.'teki grafiklerde Uluslararası Kanser Araştırma Ajansı 2020 yılı küresel kanser istatistiklerine göre, tüm bireylerden ve yaş gruplarından 2020 yılı yeni kanser vakası sayıları oranı olarak (%4,7) ile 6. sırada yer alırken; tüm bireylerden ve yaş gruplarından 2020 yılındaki kanserden ölüm sayıları oranı olarak (%8,3) ile 3. sırada yer almaktadır. Bu da bize karaciğer kanserlerinin ne kadar ölümcül olduğu hakkında bir karşılaştırma yapma imkânı vermektedir.

Tıbbi görüntüleme

Tıbbi görüntüleme, klinik analiz ve tıbbi müdahale süreçlerinde vücudun iç yapılarının non-invaziv (cerrahi işlem gerektirmeyen) yöntemlerle görsel olarak incelenmesini sağlayan bir teknolojidir. Bu işlem; bilgisayarlı tomografi (BT), X ışınları, Manyetik Rezonans Görüntüleme (MRG ya da MRI) vb. çeşitli teknikleri içerir ve sağlık hizmetlerinde çok önemli bir role sahiptir.

Tıbbi görüntüleme, başta tanı koyma olmak üzere birçok alanda kritik öneme haizdir. Tanı süreçlerinde, doktorlara vücudun iç yapıları hakkında detaylı görüntüler sunarak tümörler, kırıklar, enfeksiyonlar ve organ anormallikleri gibi çeşitli sağlık durumlarını saptama ve teşhis etme imkânı sağlar. Ayrıca tanı aşamasında tümörlerin şekil ve yoğunluğu değerlendirilerek türü (iyi huylu kötü huylu) belirlenebilmektedir. Tedavi planlama aşamasında ise sağlık profesyonellerinin hastanın durumuna göre kişiselleştirilmiş tedavi planları oluşturmasına yardımcı olur. Ayrıca, tedavi veya müdahale sürecinde izleme yaparak tedavi etkinliğini değerlendirme ve gerekli değişiklikleri yapma olanağı sunar.

Bunların yanı sıra, tarama amacıyla da kullanılan tıbbi görüntüleme, mamografi veya akciğer kanseri taramaları gibi yüksek risk grubundaki hastalıkların erken teşhisine olanak tanır. Cerrahi ve biyopsi gibi tıbbi müdahaleler sırasında rehberlik sağlaması da tıbbi görüntülemenin bir diğer önemli fonksiyonudur; bu yöntemler, doğru ve kesin müdahaleler için gerekli olan gerçek zamanlı görsel bilgiyi sağlar. Bu özellikleri ile tıbbi görüntüleme, tanıdan tedaviye, taramadan rehberliğe kadar sağlık hizmetlerinde birçok temel sürece katkı sağlamaktadır.

Bilgisayarlı tomografi (BT)

Etimolojik olarak "Tomografi" ifadesi Yunancadan gelmektedir. “Tomos” sözcüğü Yunancada dilim anlamına gelirken ve “grafein” sözcüğü Yunancada yazmak anlamına gelmektedir. Bu iki sözcüğün birleşimi ile tomografi kelimesi doğmuştur. Bilgisayarlı tomografi cihazı ile yapılan taramalar sonucunda iki boyutlu (2B) dilim (slice) biçiminde kesitsel görseller elde edilebilmekte ve bu 2B görseller özel olarak hazırlanmış bilgisayar yazılımları aracılığı ile bir araya getirilerek taranan yapının üç boyutlu (3B) hacimsel verisi dijital ortamda oluşturulabilmektedir. Tıbbi görüntüleme tekniklerinden biri olan BT sayesinde farklı açılardan alınan X-ışını görüntüleri birleştirilerek vücudun detaylı kesitsel görüntüleri oluşturulur. Özellikle kemikleri, yumuşak dokuları ve kan damarlarını detaylı şekilde görselleştiren BT taramaları, bu bölgelerdeki hasar ve hastalıklar hakkında değerli bilgiler sunar.

Elde edilen tıbbi görüntülerin paylaşılması, işlenmesi ve yaygın kullanılabilmesi için bu verilen toplanması ve saklanması aşamasında bazı standart veri tipleri geliştirilmiştir. Bu

veri tiplerine uyulması hem standardizasyon hem de verilerin işlenmesinde kolaylık sağlamaktadır.

DICOM (Digital Imaging and Communications in Medicine) terimi tıp alanında dijital görüntüleme ve iletişim olarak değerlendirilmektedir. Tıpta kullanılan görüntüleme cihazlarının sunmuş oldukları görüntülerin saklanması ve görselleştirilmesi için geliştirilmiş olan bir veri formatıdır. Bu cihazlardan gelen DICOM verileri sadece iki boyutlu görüntüler içerir.

NIfTI (Neuroimaging Informatics Technology Initiative), Nöro Görüntüleme Bilişim Teknolojisi Girişimi'nin kısaltması olup, Bilgisayarlı Tomografi (BT), Manyetik Rezonans Görüntüleme (MRG) vb. taramalardan elde edilen veriler de dahil olmak üzere tıbbi görüntüleme verilerini depolamak için kullanılan bir dosya formatıdır. Tıbbi görüntü analizi alanında yaygın olarak kullanılan bir format olup, birçok yazılım aracı ve kütüphane tarafından desteklenmektedir.

NIfTI dosyalarıyla görüntüler 3 boyutlu biçimde saklanır. Diğer tıbbi görüntü dosyası formatlarının uzamsal yönelim zorluklarının üstesinden gelmek için özel olarak bu şekilde tasarlanmıştır. Öte yandan DICOM görüntü dosyaları ve ilgili veriler 2 boyutlu katmanlardan oluşur. Bu sayede, bir görüntünün farklı dilimleri teker teker görüntülenebildiğinden insan vücudunu ve çeşitli organları analiz ederken kullanışlıdır. Sağlık profesyonellerinin ve veri bilimcilerinin, görüntü kalitesini kaybetmeden DICOM dosyalarını NIfTI dosyalarına dönüştürebilmektedir. Verilere farklı bir formatta ihtiyaç duyulduğunda NIfTI dönüşümü yararlı olur. Üstesinden gelinmesi gereken asıl zorluk, herhangi bir ek açıklama, etiket veya meta veri kaybı olmadan katmanları tek bir görüntüde birleştirmek olduğundan, çeşitli açık kaynaklı ve özel yazılım paketleri bunu yapmanıza yardımcı olabilmektedir.

Amaç

Karaciğerin toplam hacminin belli bir kısmını kaybetse bile kendi kendini yenileyebilen bir organ olduğundan tümörlerinin erken aşamada tespit edilmesi ve tümörlü bölgenin uzaklaştırılması tedavinin başarısını ve hastanın yaşam süresinin uzatılmasını önemli ölçüde etkilemektedir. Bu sebeple karaciğer tümörünün otomatik olarak bölütlenmesi karaciğer

tümörlerinin teşhisi ve tedavisi için çok önemli bir uygulamadır [3]. Ayrıca spesifik olarak, tümör lezyonlarının lokalizasyonun verilen tıbbi görüntü üzerinde taranması, radyo terapi gibi birçok tedavi seçeneği için bir ön koşuldur [4].

Diğer birçok tıbbi görüntüleme uygulamasında olduğu gibi, manuel olarak üç boyutlu BT görüntülerinden hedef lezyonların tanımlanması zaman alıcı, uzmanlık gerektiren, tekrar edilebilirliği zayıf işlemlerdir ve operatöre bağlı sonuçlar ortaya çıkabilir [5].

Karaciğerin ve karaciğer lezyonlarının otomatik bölütlenmesi hala birçok açıdan zorluklar barındırmaya devam etmektedir. İlk olarak, lezyonlar ve arka plan kontrastları arasındaki varyasyonlar. İkinci olarak farklı odak lezyonların (iyi huylu, kötü huylu vb.) bir arada bulunmasından kaynaklanan, tıbbi görüntülerdeki değişiklikler. Üçüncü olarak karaciğer kanseri varlığının yaygın öncü göstergesi olan karaciğer dokusunun arka plan sinyallerinin büyük oranda değişiklik göstermesidir. Sayılan bu zorluklar otomatik tümör bölütlemesini hala gelişime açık bir alan olarak bırakmaktadır [6].

Bu tez çalışmasının amacı; MONAI derin öğrenme kütüphanesini kullanarak oluşturulan derin öğrenme modelini, karın içi BT görüntüleri içeren LiTS veri seti ile eğiterek karaciğer tümörlerini otomatik bölütlemek; uzmanlık gerektiren, zaman alıcı ve uzmana göre çeşitlilik gösteren (hataya açık) bu işlemi otomatik olarak gerçekleştirip karaciğer tümörü tanısını hızlandırmak ve tedavi sürecinde sağlık profesyonellerine ve hastalara katkı sağlamaktır.

Çalışmasının birinci bölümünde tezin konusu, arka planı ve amacı hakkında bilgiler verilmiştir. İkinci bölümde bu alanda yapılan çalışmaların teorik altlığı verilmiştir. Üçüncü bölümde literatür taraması yapılmış ve önceki çalışmalar paylaşılmıştır. Dördüncü bölümde tezin kapsamı, kullanılan yöntemler ve araçlar hakkında detaylı bilgiler paylaşılmıştır. Beşinci bölümde tez çalışması kapsamında önerilen yöntem açıklanmıştır. Altıncı bölümde yapılan çalışmada elde edilen deneysel sonuçlar paylaşılmış, yedinci bölümde de çalışmanın sonuçları değerlendirilerek literatüre yaptığı katkılara ve gelecekteki çalışmalar için önerilere yer verilmiştir.

2. TEORİK BİLGİLER

2016'dan önce otomatik karaciğer ve karaciğer tümörü bölütleme çalışmalarında genel olarak geleneksel makine öğrenmesi metotları kullanılmıştır. Ancak 2016'da MICCAI (International Conference on Medical Image Computing and Computer-Assisted Intervention) Tıbbi Görüntü İşleme ve Bilgisayar Destekli Müdahaleler Konferansındaki ilk ilgili yayından bu yana söz konusu çalışmalarda derin öğrenme metotları ilk tercih olarak karşımıza çıkmaktadır. Bu çalışmada da belirtildiği gibi karaciğer lezyonları çok farklı şekil, boyut ve kontrast aralığına sahiptir. Tümörler karaciğerin neredeyse her bölgesinde olabilir ve genellikle sınırları net değildir. Tıbbi görüntü elde edilirken kullanılan kontrast maddelerinin çeşitliliği de ek varyasyonlar meydana getirebilir. Bu sebeple karaciğer tümörlerinin otomatik bölütlenmesi karaciğerin bölütlenmesine kıyasla çok daha zorlu bir görevdir [7].

Karaciğer tümörlerini bölütlenmesi için yayınlanmış çalışmalarda kullanılan metotlar üç grup altında toplanabilir;

- Eşikleme ve uzamsal düzenleme (Thresholding and spatial regularization)
- Yerel özellikler ve öğrenme algoritmaları (Local features and learning algorithms)
- Derin öğrenme (Deep learning)

Eşikleme ve uzamsal düzenleme metotlarını kullanan yöntemler; bu yöntemler tümör alanlarının gri ton değerlerinin, tümör dışındaki bölgelere ait 2D piksellerin ya da 3D voksellerin gri ton değerlerinden farklı olduğu varsayımına dayanarak, eşikleme işlemi ile tümörü karaciğerden ve arka plandan otomatik olarak ayıran basit ancak etkili metotlardır ve ilk olarak Soler ve diğerleri (2001) tarafından kullanılmıştır [8]. O zamandan beri tümör bölütlemesi sonuçlarını iyileştirmek için eşikleme [9], sınıflar arasındaki maksimum varyans [10] ve iteratif algoritma [11] gibi yöntemler üzerinde çalışmalar yapılmıştır. Uzamsal düzenleme teknikleri ise görüntü veya morfolojiler hakkında, örneğin tümör boyutu, şekli, yüzeyi veya uzam bilgilerine dayanır. Heterojen lezyon bölütlemesi için adaptif eşikleme yöntemlerini, model tabanlı morfolojik işleme ile birleştiren çalışmalar yapılmıştır [12]. Aktif kontur tabanlı tümör bölütlemesi [13] şekil ve yüzey bilgisine dayanır ve olasılıksal modeller [14] veya histogram analizi [15] kullanarak otomatik bölütleme haritaları oluşturur.

Seviye seti [16] yöntemleri, yüzeylerin ve şekillerin sayısal analizinde bir araç olarak seviye kümelerini kullanmak için kavramsal bir çerçevedir ve nesneleri parametrelendirmek zorunda kalmadan tümör şekillerinin sayısal olarak hesaplanmasını sağlar. Karaciğer tümörü bölütlemesi için seviye seti yaklaşımları, denetimli bulanık piksel sınıflandırması [17] ve üç boyutlu voksel sınıflandırması teknikleri ile [18] birleştirilmiştir.

Yerel özellikler ve öğrenme algoritmaları kullanan yöntemler; k-means içeren kümeleme yöntemleri [19] ve bulanık c-means kümeleme yöntemleridir [20].

Derin öğrenme kullanan yöntemler; LiTS (Liver Tumor Segmentation) veri setinden önce, derin öğrenme yöntemleri karaciğer tümörü bölütlemesi görevleri için nadiren kullanılmıştır. Bunun en büyük sebebi yeterli sayıda, güvenilir etiketli veri bulunmamasıdır. Daha sonraki birçok derin öğrenme yaklaşımı, LiTS veri seti ile birlikte geliştirilmiş ve test edilmiştir.

3. LİTERATÜR TARAMASI

Karaciğer ve karaciğer tümörü bölütlemesi için 3D UNet modelinin kullanılmasını ilk olarak Christ ve diğerleri (2016) gerçekleştirmiştir. 2016 yılında, karaciğer ve lezyonlarını kaskad derin sinir ağları ve 3D koşullu rastgele alanlar (conditional random fields) kullanarak bölütleyen bir algoritma önermişlerdir. Makalede, karaciğer ve karaciğer tümörü için iki kaskad UNet ağı kullanılmıştır. Ardından, bu ağlardan gelen çıktılar iyileştirme için 3D koşullu rastgele alanlara girdi olarak kullanılmıştır. Çalışmada karaciğer bölütlemesinde dice benzerlik katsayısı değeri 0,943 olarak elde edilmiş, ancak tümör bölütlemesi sonuçları makalelerinde sunulmamıştır [21].

Yaklaşık bir yıl sonra Sun ve diğerleri (2017) kontrastı artırılmış çok fazlı BT tarama görüntüsünden karaciğer tümörlerini çok kanallı tam evrişimli ağ (FCN-Fully Convolutional Network) kullanarak bölütleme fikrini ortaya koymuştur. Tam evrişimli ağın tek kanalı, sekiz evrişim katmanı, üç alt örnekleme katmanı, üç ters evrişim katmanı ve iki özellik birleştirme katmanından oluşmuştur. Evrişim katmanları, önceki katmanın çıktısına farklı çekirdek boyutlarıyla evrişim işlemleri gerçekleştirir ve mekansal korelasyonları koruyarak görüntüden özellikler çıkartır. Alt örneklem katmanı, görüntü çözünürlüğünü etkilemeden görüntünün boyutunu azaltır. Daha sonra ters evrişim katmanı ve birleştirme katmanları, sırasıyla yükseltme ve birleştirme işlemi için kullanılmıştır. Çalışmaları için, BT tarama görüntüsünün üç farklı fazından görüntünün özelliklerini çıkarmak için üç adet tam evrişim katmanı kullanmışlardır. Her tam evrişimli ağ kanalı bağımsız olarak eğitilmiş ve özellikleri daha iyi bölütleme sonucu için birleştirilmiştir. Değerlendirme için farklı performans metrikleri kullanılmıştır ve hacimsel örtüşme hatası (VOE – Volumetric Overlapping Error) 8,1±4,5 elde edilmiştir [22].

Bir yıl sonra Chlebus ve diğerleri (2018), Christ ve diğerlerinin (2016) fikrine katılmış ve karaciğer tümörlerini bölütlemek için UNet mimarisini kullanmıştır. Ancak, uygulanan UNet mimarisini değiştirerek, ağın eğitimi sırasında parametre güncellemeleri ve ağın hızını artırmak için bir seyreltme (dropout) katmanı ve ek kısa atlama bağlantıları (short skip connections) eklemişlerdir. Ardından, değiştirilmiş UNet'ten gelen çıktıları, daha fazla iyileştirme için üç boyutlu bağlantılı bileşen kullanılarak işlemiş, bir şekil tabanlı işleme yöntemi ve daha sonra bir rastgele orman sınıflandırıcısı kullanılarak tekrar sınıflandırmıştır.

Bölütleme sonuçlarında ortalama bir dice benzerlik katsayı değeri olan 0,58 elde etmiştir. Ancak burada, tümörlerin değişken bir şekle sahip olması nedeniyle şekle dayalı işlemin düşünüldüğü kadar etkili olmayabileceği ortaya çıkmıştır [23].

Ve sonra Liu ve diğerleri (2019), Christ ve diğerleri (2016) ile Chlebus ve diğerlerinin (2018) çalışmasını geliştirmiş ve Unet'i grafik kesim algoritmasıyla birleştiren GIU-Net'i kullanarak karaciğer sıralı görüntülerini bölütleme fikrini ortaya atmıştır. Geliştirdikleri UNet mimarisinde, daha iyi bir bölütleme sonucu elde etmek için yapıyı derinleştirmiş ve atlama bağlantılarını havuzlama katmanlarının çıktılarından elde etmek suretiyle değiştirmiştir. Orijinal UNet'ten farklı olarak, bu yeni yapıda yükseltme katmanının çıktısını, karşılık gelen evrişim katmanının çıktısı ile birleştirmemiş ve bu yeni Unet yapısına IU-NET adını vermiştir. Daha sonra, bu yapıyı grafik kesim yöntemiyle birleştirerek GIU-Net adında yeni bölütleme algoritmalarını sunmuştur. Algoritmalarının performansını değerlendirmek için farklı metrikler kullanmış ve dice benzerlik katsayısı olarak 0,9505 değerini elde etmişlerdir. Bu makalede yazarlar yalnızca karaciğer bölütlemesi için bir algoritma sağlamıştır ve karaciğer patolojilerini bölütlemek için bir çözüm sunmamıştır [24].

Qin ve diğerleri (2018), yaptıkları çalışmada öncelikle tüm BT görüntülerini, benzer BT numarasına sahip yakın piksellerin bir araya getirildiği süper piksel bölgelerine ayırmıştır. Daha sonra, geleneksel ikili bölütlemeyi çoklu sınıflı bir sınıflandırmaya dönüştürerek süper piksel bölgelerini üç sınıfa ayırmıştır: karaciğer içi, karaciğer sınırı ve karaciğer olmayan arka plan. Bu şekilde, karaciğerin sınır bölgesi açıkça tanımlanmış ve sonraki sınıflandırma için öne çıkarılmıştır. Üçüncü olarak, her BT hacmi için entropi tabanlı bir belirginlik haritası hesaplamışlar ve bu haritayı süper piksel bölgeler üzerinden görüntü yaması örneklemesini yönlendirmek için kullanmışlardır. Bu sayede, bilgilendirici bölgelerden (örneğin, düzensiz değişikliklere sahip karaciğer sınırı) daha fazla yama, homojen bölgelerden ise daha az yama çıkarılmıştır. Son olarak, derin bir ESA (Evrişimli Sinir Ağı) hattı oluşturulmuş ve karaciğer sınırının olasılık haritasını tahmin etmek için eğitilmiştir. Önerilen algoritma 100 hastadan oluşan bir hasta grubu üzerinde test edilmiştir. 10 kat çapraz doğrulama ile SBBS-CNN, %97,31 dice benzerlik katsayısı elde etmiştir [25].

Han ve diğerleri (2018), gerçekleştirdikleri çalışmada, derin konvolüsyonel frameletler teorisinden ilham alarak, UNet'in sınırlamalarını ortaya koymayı ve yeni çok çözünürlüklü derin öğrenme yaklaşımlarını önermeyi hedeflemiştir. Özellikle, çift çerçeve ve sıkı çerçeve

UNet gibi alternatif UNet varyantlarının, seyrek-görünüm BT'de yüksek frekanslı kenarların daha etkili bir şekilde yeniden elde edilmesini sağlayan çerçeve koşulunu sağladığını göstermişler ve dice benzerlik katsayısı: %89,72 sonucu elde etmişlerdir [26].

Men ve diğerleri (2018), kademeli evrişimli sinir ağları kullanarak CAC-SPP adıyla yeni bir model geliştirmiş ve performansı, yaygın olarak kullanılan UNet ve ResNet-101 ile karşılaştırmıştır. Tümörlerinin bağımsız bölütlemesi için iki farklı görüntü seti kullanılmıştır: (1) 70 T2-ağırlıklı MR görüntüsü ve (2) 100 planlama BT görüntüsü. Sonuçlar, önerilen CAC-SPP' nin her iki görüntü setinde de UNet ve ResNet-101'yi geride bıraktığını göstermiştir [27].

Li ve diğerleri (2018), yaptıkları çalışmada 2 boyutlu (2B) evrişimli sinir ağlarının üçüncü boyuttaki mekansal bilgileri tam olarak kullanamaması, 3 boyutlu (3B) evrişimli sinir ağlarının ise yüksek hesaplama maliyeti ve GPU bellek tüketimi sorunlarını çözmek için, verimli bir şekilde intra-dilim özelliklerini çıkaran 2D DenseUNet ve hacimsel bağlamları hiyerarşik olarak birleştiren 3D karşılıkları içeren yeni bir hibrit yoğun bağlantılı UNet (H-DenseUNet) önermiştir. Bu yöntem, karaciğer ve tümör bölütlemesi için auto-context algoritmasının ruhuna uygun şekilde, intra-dilim temsillerinin ve inter-dilim özelliklerinin birleştirilerek optimize edilebileceği bir öğrenme süreci formüle etmektedir. Yöntem, MICCAI 2017 Karaciğer Tümör Segmentasyonu Yarışması veri seti ve 3DIRCADb veri seti üzerinde kapsamlı bir şekilde değerlendirilmiş ve tümör bölütleme sonuçlarında diğer ileri düzey yöntemleri geride bırakırken, tek bir modelle karaciğer bölütlemesi için de çok rekabetçi bir performans sergilemiştir [28].

Seo ve diğerlerinin (2019) gerçekleştirdiği çalışma, karaciğer görüntülemesi ve özellikle karaciğer tümör bölütlemesi üzerine önemli bir katkı sağlamıştır. Çalışma, tıbbi görüntüleme verilerini kullanarak karaciğer tümörlerinin otomatik bölütlenmesini geliştirmeyi amaçlamış ve bu süreçte derin öğrenme tabanlı yaklaşımları incelemiştir. Yazarlar yaptıkları çalışmalarda UNet mimarisinin görüntü analizi görevlerinde karşılaşılan bazı önemli dezavantajlara sahip olduğunu belirtmiştir. Bu dezavantajlar, düşük çözünürlüklü bilgilerin atlama bağlantılarıyla yeniden gönderilmesi nedeniyle bulanıklaşma, yüksek çözünürlüklü kenar bilgisi eksikliği ve optimal birleşme sayısının belirlenmesindeki zorluklardır. Bu sorunları çözmek için yazarlar, UNet'teki atlama bağlantılarına ters evrişim ve aktivasyon işlemleri eklemişlerdir. Bu şekilde modifiye edilmiş UNet (mU-Net) modeli, 2017 Global

Liver Tumor Segmentation (LiTS) veri seti üzerinde test edilmiş ve Karaciğer bölütlemesi için; dice benzerlik katsayısı: %98,51 sonucu elde edilmiştir. Bu sonuç, modelin karaciğer dokusunu oldukça yüksek doğrulukla bölütlendiğini göstermektedir. Tümör bölütlemesi için; dice benzerlik katsayısı: %89,72 sonucu elde edilmiştir. Bu sonuç değerlendirildiğinde modelin tümör bölütlemesinde karaciğer dokusuna göre daha düşük bir başarı elde etmiş olduğunu göstermektedir. Bunun sebebi, tümörlerin boyutlarının ve şekillerinin oldukça değişken olmasıdır.

Bu model 3dIRCADB veri seti ile test edildiğinde ise tümör bölütlemesi için dice benzerlik katsayısı: %96,01 elde edilmiştir [29].

Karaciğer ve tümör bölütlemesi ile ilgili literatürdeki bazı sonuçlar Çizelge 3.1'deki gibi verilmiştir.

Çizelge 3.1. Karaciğer ve tümör bölütlemesi çalışmalarından elde edilen bazı Dice Benzerlik Katsayısı değerleri

	DSC (%) (Karaciğer)	DSC (%) (Tümör)
Qin ve diğerleri	97,18	39,82
Han ve diğerleri	97,36	55,42
Men ve diğerleri	97,50	83,14
Li ve diğerleri	98,05	86,53
Seo ve diğerleri	98,51	89,72

Sonraki yıl Budak ve diğerleri (2020), karaciğer ve karaciğer tümörün verimli bir şekilde bölütlenmesi için iki kademeli enkoder-dekoder evrişimli sinir ağı geliştirmiştir. İki simetrik kodlayıcı ve dekoder bölümleri içeren EDCNN algoritmasını sunmuştur. Her bir bölüm, maksimum havuzlama katmanından sonra toplu normalleştirme ve ReLU aktivasyonu içeren on evrişim katmanından oluşur. Ve her evrişim katmanında 64 adet filtre kullanmışlar ve bu sayede gerekli olan parametre sayısını azaltmışlardır. Bölütleme işleminde bir tanesi karaciğer diğeri ise tümör için olmak üzere iki kademeli derin sinir ağı kullanmışlardır. İlk ağın çıktısı, bir sonraki ağ için giriş olmuştur. Karaciğer ve tümör bölütlemesinde sırasıyla ortalama dice benzerlik katsayısı değeri olan 0,9522 ve 0,634 değerlerine ulaşmışlardır [30]. Son yıllarda yapılan çalışmalar veri eksikliğinden dolayı daha çok denetimsiz öğrenme üzerine yoğunlaşmıştır.

4. KAPSAM, YÖNTEM VE ARAÇLAR

4.1. Yapay Zekâ, Makine Öğrenmesi ve Derin Öğrenme

Genel bir terim olarak yapay zekâ, bir bilgisayarın veya bir makinenin, genellikle insana özgü (anlama, çözüm yolu bulma, genelleme yapma ve geçmişteki tecrübelerden faydalanarak öğrenme gibi) mantık gerektiren süreçlerle ilgili işleri yerine getirebilme yeteneği olarak tarif edilebilir [31]. Bu bağlamda bakıldığında yapay zekâ herhangi bir karar sürecini işletebilen veya tahmin işlemini gerçekleştirebilen teknolojiler olarak tanımlanabilir.

Makine öğrenmesi ise kendisine gösterilen verilerden öğrenme sürecini temel alan ve algoritmanın, önceden belirlenmiş kurallarla değil, tamamen verilere dayanarak bilgi edindiği bir yapay zekâ yaklaşımıdır. Bu yönüyle makine öğrenmesi, “hard-coded” olarak tabir edilen, sabit kurallara dayalı sembolik yapay zekâ algoritmalarından ayrılmaktadır. Makine öğrenmesi modelleri, Naive Bayes, Destek Vektör Makineleri (Support Vector Machines), Karar Ağaçları (Decision Trees), Rastgele Orman (Random Forest) ve xGBoost gibi çeşitli algoritmalar kullanılarak tanımlanabilir. Her biri belirli veri türleri ve problem yapıları için farklı avantajlar sunan bu algoritmalar, modelin veriden öğrenme yeteneğini etkinleştiren temel yapı taşlarıdır.

Derin öğrenme, makine öğrenmesinin alt dalı olarak değerlendirilen ve çok katmanlı sinir ağları (neural networks) üzerinden öğrenmeyi gerçekleştiren özel bir algoritma ailesidir. Veriden öğrenme yaptığından tüm derin öğrenme algoritmaları aynı zamanda bir makine öğrenmesi algoritmasıdır fakat bunun tersi doğru değildir. Yani her makine öğrenmesi derin öğrenme değildir. Derin öğrenme, geleneksel makine öğrenmesi algoritmalarından farklı olarak öğrenme sürecini tek bir matematiksel model yerine, katman olarak adlandırılan ve verinin özel bir tür temsili ile çalışan sinir ağlarına dayandırmaktadır. Bu ağlar, verinin en küçük bilgi parçacığından başlayarak, örneğin bir fotoğraf gibi karmaşık bir bütün haline dönüşene kadar farklı seviyelerde temsil edilmesini sağlayan katmanlar dizisidir.

Makine öğrenmesi modellerinde, hangi özelliklerin (parametrelerin) modele dahil edilmesi gerektiği, geliştiricinin belirlemek zorunda olduğu bir sorundur. Bu süreç, "feature

engineering" (özellik mühendisliği) olarak adlandırılmakta olup, veri analizinden veya parametre seçim algoritmalarından faydalanılarak çözümlenir. Ancak, derin öğrenme algoritmalarında bu süreç yer almaz; derin öğrenme modelleri, verinin yapısına dayanarak hangi özelliklere ne derecede önem vereceğini ve parametreleri nasıl değerlendireceğini kendiliğinden keşfetme yeteneğine sahiptir. Bu durum, derin öğrenme modellerinin, veri içindeki karmaşık örüntüleri ve ilişkileri daha etkili şekilde ortaya çıkarmalarını sağlar.

Derin öğrenme algoritmalarının, geleneksel makine öğrenmesi algoritmalarına kıyasla bazı dezavantajları da bulunmaktadır. Bu algoritmaların, yüksek doğruluk oranına sahip ve karmaşık sonuçlar üretebilmeleri için büyük veri setlerine ve güçlü donanımlara (özellikle grafik işlemciler gibi yüksek performanslı işlemcilere) ihtiyaçları vardır. Derin öğrenme modellerinin geliştirilmesi ve uygulanması, bu nedenle, makine öğrenmesine kıyasla daha fazla kaynak gerektirmektedir.

Bu bağlamda, derin öğrenme ve makine öğrenmesi arasındaki temel farklar dikkate alındığında, her iki yaklaşımın da belirli problem türleri için kendine has avantaj ve dezavantajlar sunduğu anlaşılmaktadır. Derin öğrenme, veri miktarının büyük olduğu durumlarda öne çıkarken, makine öğrenmesi algoritmaları daha az verinin bulunduğu ve daha basit modellerin yeterli olduğu durumlar için daha uygun olabilmektedir.

4.2. Yapay Sinir Ağları (YSA)

Derin öğrenme, insan beyninden ilham alarak bilgisayar sistemlerine dataları işlemeyi öğreten bir yapay zekâ yöntemidir. Derin öğrenme modelleri karmaşık resimler, metinler ve sesler gibi veri modellerini tanıyabilir. Yapay Sinir Ağları (YSA) ise derin öğrenme teknolojilerinin altlığıdır. Çok fazla sayıda ve katman katman birbiri ile bağlantılı düğümlerden oluşur. Bu düğümler dataları sistematik ve uyarlanabilir bir şekilde işlerler. Elde edilen tahmin (çıktı) hakkında geri bildirimlerde bulunurlar ve hataları düzelterek sürekli gelişirler. Bu yüzden, yapay sinir ağları derin öğrenme teknolojisinin temel yapıtaşıdırlar [32].

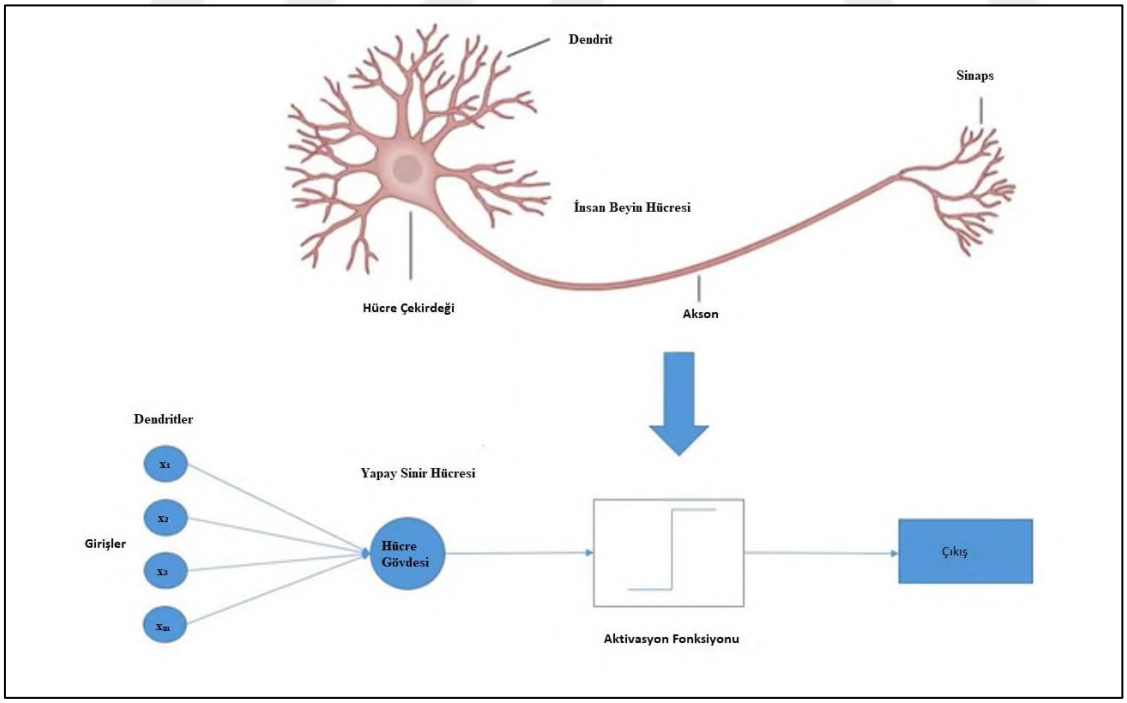
Diğer bir ifadeyle Yapay Sinir Ağları (YSA), insanlara özgü öğrenme yolu ile yeni bilgiler ortaya çıkarabilme işlevini gerçekleştirebilmek maksadı ile geliştirilen bilgisayar

teknolojileridir. YSA, insanların yaşayarak elde ettiği deneyimlerden faydalanarak öğrenme işlevini kendisine sunulan veriler yoluyla sağlamaktadır.

YSA'dan beklenen temel fonksiyon, benzer durumlara karşılık benzer kararlar verebilmesidir. YSA'nın güvenilir bir şekilde çalıştırılabilmesi için eğitim sürecinin başarı ile bitirilmesi, tüm örnek olayların yapay sinir ağına gösterilmesi ve örnekler ile olay arasındaki ilişkilerin ortaya çıkarılması gerekmektedir [33].

YSA, insan beyninin çalışma biçimi temel alınarak, öğrenme adımlarının matematiksel olarak modellenmeye çalışılması uğraşları ile ortaya çıkmıştır. Bu sebeple, bu konu hakkındaki araştırmalar öncelikle insan beynini oluşturan biyolojik sinir hücrelerinin modellenmesi ve bilgisayar sistemlerinde uygulanması ile başlamıştır. Sonrasında bilgisayar donanımlarının gelişimi ile beraber birçok alanda kullanılır hale gelmiştir.

Biyolojik sinir sistemini oluşturan temel unsur sinir hücreleridir ve Şekil 4.1.'de gösterildiği gibi dört temel bölüme ayrılmaktadır; dendrit, akson, hücre çekirdeği ve sinaps [34].

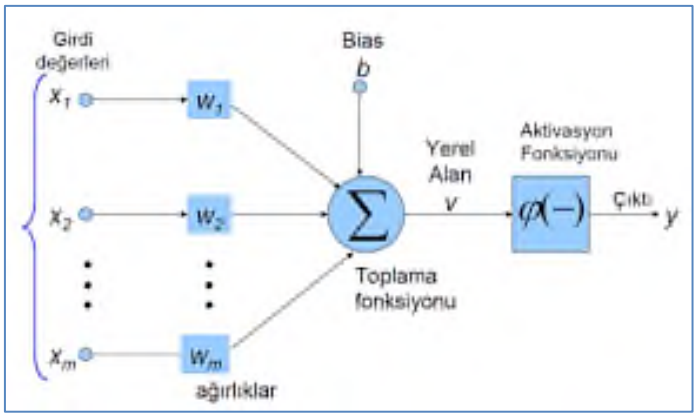


Şekil 4.1. Biyolojik ve Yapay Sinir Hücrelerinin Yapısı [35]

YSA'ları oluşturan yapay sinir hücreleri, biyolojik sinir hücrelerinin modellenmesiyle gerçekleştirilir. Genel olarak biyolojik sinir sisteminin tersine, Yapay Sinir Ağları (YSA),

çözülmek istenen sorunla baş edebilmek için belirlenen sabit bir topoloji kullanılarak eğitilmişlerdir. Topolojileri sabittir ve süreç ilerledikçe değişim göstermez. Ağırlık değerleri başlangıçta rastgele seçilir ve girdi değerlerinin ağırlıkları istenen bir çıktı fonksiyonu değerini elde edecek biçimde bir optimizasyon algoritması yardımı ile ayarlanır.

Eğitim safhası ister sıfırdan olsun ister ince ayardan olsun, ağırlıkları güncelleme işlemi, veriyi yapay sinir ağından geçirmek, çıkan sonucu ölçmek ve ağırlıkların değerini bu sonuca göre değiştirmek olarak tanımlanabilir. Ağırlıklar, modelden beklenen görevin performansını yükseltecek şekilde kademeli olarak değiştirilir. Buradaki öğrenme işlevi gündelik hayatta nesneleri tanıyan bir çocuğunkine benzerdir. Birçok başarısız deneme ve elde edilen tecrübelerden sonra çocuk farklı bir açıdan yeniden dener. Yapay sinir ağları da öğrenme süreçlerini işletirken bu şekilde modellenmişlerdir. Doğru sonuçlar (etiketli eğitim veri seti) ile beslenirler ve bir öğrenme süreci olarak yapay sinir ağları doğru cevapların sayısını maksimuma çıkaracak şekilde ağırlıkları ayarlarlar. Dışarıdan veya diğer yapay sinir hücrelerinden alınan girdiler, ağırlıklar vasıtası ile diğer yapay sinir hücrelerine bağlanırlar. Toplama fonksiyonu yardımı ile net girdi değeri elde edilir. Net girdi değerinin aktivasyon fonksiyonundan geçirilmesi ile Eşitlik 4.1.'deki gibi net çıktı değeri hesaplanır. Bu net çıktı değerleri modelin verdiği tahmin değerleridir. Her bağlantının bir ağırlık değeri vardır ve bu değer değişkendir. YSA kendisine veriler gösterildikçe bu ağırlık değerlerini değiştirir. Bunu yapmaktaki amaç ağın gösterilen örnekler için doğru çıkışları verecek ağırlıkları bulmasını sağlamaktır. Şekil 4.2. Tek Katmanlı Yapay Sinir Ağı şematik olarak görülmektedir.



Şekil 4.2. Tek Katmanlı Yapay Sinir Ağı [36]

$x = x_1, x_2, x_3 \dots x_m$ girdi değerleri,

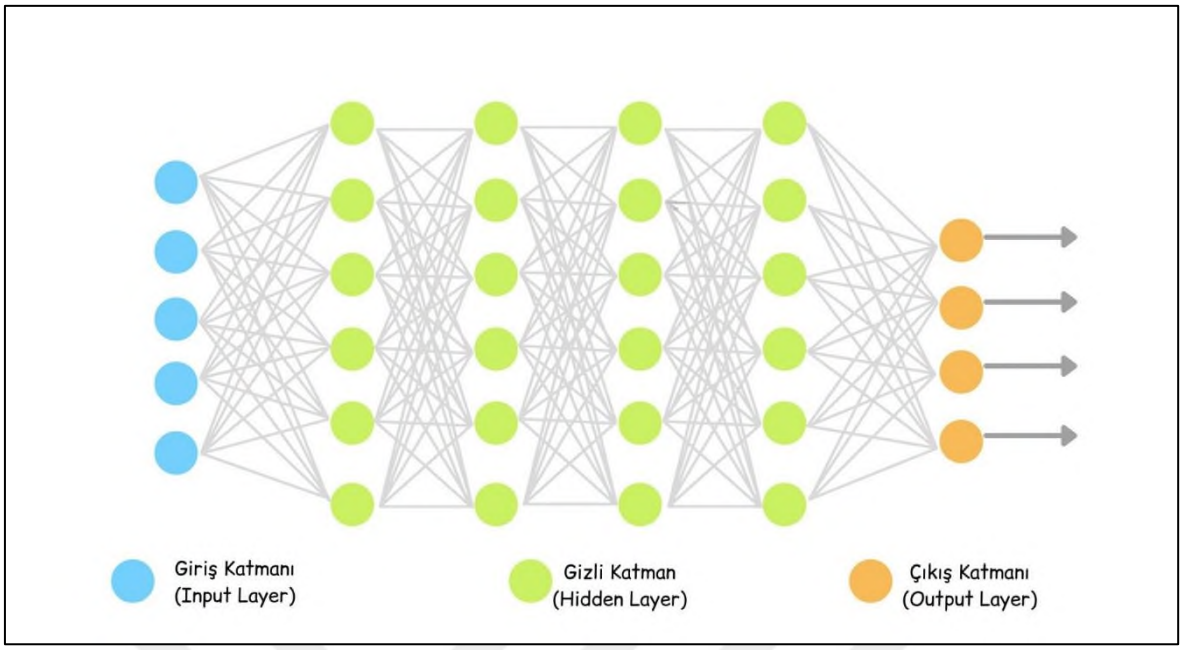
$w = w_1, w_2, w_3 \dots w_m$ ağırlıklar,

b bias,

$$Y' = f(x_1w_1 + x_2w_2 \dots x_mw_m + b) = f(\Sigma x_i w_i + b) \quad (4.1)$$

4.3. Derin Sinir Ağları (DSA)

Giriş katmanı ile çıkış katmanı arasında çok sayıda gizli katman yer alan yapay sinir ağına (YSA), derin sinir ağı (DSA) denilmektedir. Derin sinir ağları, derin mimarileri kullanan sinir ağlarıdır. "Derin" terimi, tek bir katmanda daha fazla sayıda katmana ve birime sahip olan işlevleri ifade eder. Daha yüksek düzeyde desenler yakalamak için daha fazla ve daha büyük katmanlar ekleyerek daha doğru modeller oluşturmak mümkündür. Şekil 4.3.'te bir derin sinir ağı yapısı görülmektedir.



Şekil 4.3. DSA [37]

Bir derin sinir ağı birçok bileşenden oluşur. Bu bileşenler şöylece sıralanabilir;

Girdi katmanı

Bir derin sinir ağında özniteliklerinin öğrenilmesi istenen örneğin modele verildiği katmandır. Girdi katmanı, ağın içine veri girişi yapılan kısımdır ve her bir nöron, girdi verisinin benzersiz bir özelliğini temsil eder. Bu katmanın temel görevi, ham veriyi almak ve daha fazla işlem yapılması için bir sonraki katmana aktarmaktır.

Düğümler (Nöronlar)

Nöronlar veya düğümler, bir sinir ağının temel işleme birimleridir. Her nöron bir girdi alır, bunu (ağırlıklı bir toplam ve bir aktivasyon fonksiyonu uygulayarak) işler ve çıktıyı bir sonraki katmana gönderir.

Aktivasyon fonksiyonları

Aktivasyon fonksiyonlarının amacı ağırlık ve bias değerlerini ayarlamaktır. Aktivasyon fonksiyonları, sinir ağında neyin öğrenilip neyin göz ardı edileceğine karar vererek, ağı esnek hale getirir. Bu fonksiyonlar, ağın karmaşık desenleri yakalayıp öğrenmesini sağlar.

Sigmoid Fonksiyonu, tanh (Hiperbolik Tanjant) Fonksiyonu, Relu Fonksiyonu vb. birçok aktivasyon fonksiyonu mevcuttur. Problemin ne olduğuna göre bu fonksiyonlar arasından seçim yapmak gerekebilir. Mesela iki sınıflı bölütleme problemlerinin çözümü için sigmoid aktivasyon fonksiyonu uygun iken, çok sınıflı bölütleme problemlerinin çözümünde softmax aktivasyon fonksiyonu uygun olabilmektedir.

Ağırlıklar ve biaslar

Ağırlıklar, ağın katmanlarındaki girdilerin çıktılar üzerindeki etkisini belirleyen parametrelerdir. Biaslar ise, aktivasyon fonksiyonlarının sıfır olmayan çıktılar üretebilmesini sağlamak için verilen ilk değerler olarak tanımlanabilirler. Ağın etkili bir şekilde aktive olmasına ve öğrenmesine katkıda bulunurlar.

Gizli katman

Girdi ve çıktı katmanları arasında yer alan gizli katmanlar, sinir ağının ana hesaplamalarını gerçekleştirir. “Gizli” olarak adlandırılırlar çünkü girdi ve çıktı katmanlarının aksine dış ortamla etkileşime girmezler. Bir sinir ağının karmaşıklığı ve kapasitesi, gizli katmanların sayısı ve mimarisine bağlıdır.

Çıktı katmanı

Sinir ağının son katmanı olan çıktı katmanı, sonuçları sunar ve gizli katmanlardan gelen bilgiyi, sınıflandırma, regresyon veya diğer tahmin türleri gibi göreve uygun bir formata dönüştürür.

Kayıp fonksiyonu

Kayıp fonksiyonu, modelin tahmin ettiği çıktı ile gerçek değer (ground truth) arasındaki farkı ölçer. Eğitimin amacı bu fonksiyonu minimize ederek modelin daha doğru tahminlerde bulunabilmesini sağlamaktır.

Optimizasyon algoritmaları

Bu algoritmalar, modelin doğruluğunu zaman içinde iyileştirmek için ağırlıkları ve biasları ayarlayarak hataları azaltır.

Geriye yayılım

Bu derin öğrenme algoritması, modelin hatalarından öğrenmesini ve gelişmesini sağladığı için oldukça önemlidir. Modelin ağırlıklarındaki değişikliklerin doğruluğunu nasıl etkilediğini hesaplar ve bu ayarları, modelin tahminleri daha iyi yapabilmesi için hataları geriye doğru izleyerek ayarlar.

Düzenleme teknikleri

Modeller bazen eğitim verisini aşırı öğrenir (overfitting), bu durum da modelin yeni veriler üzerinde iyi performans gösterememelerine neden olur. Bunu dengelemek için, katman normalizasyonu (batch normalization) gibi düzenleme teknikleri kullanılarak ağırlıkların boyutu ayarlanır ve eğitim süreci hızlandırılır.

Katman normalizasyonu

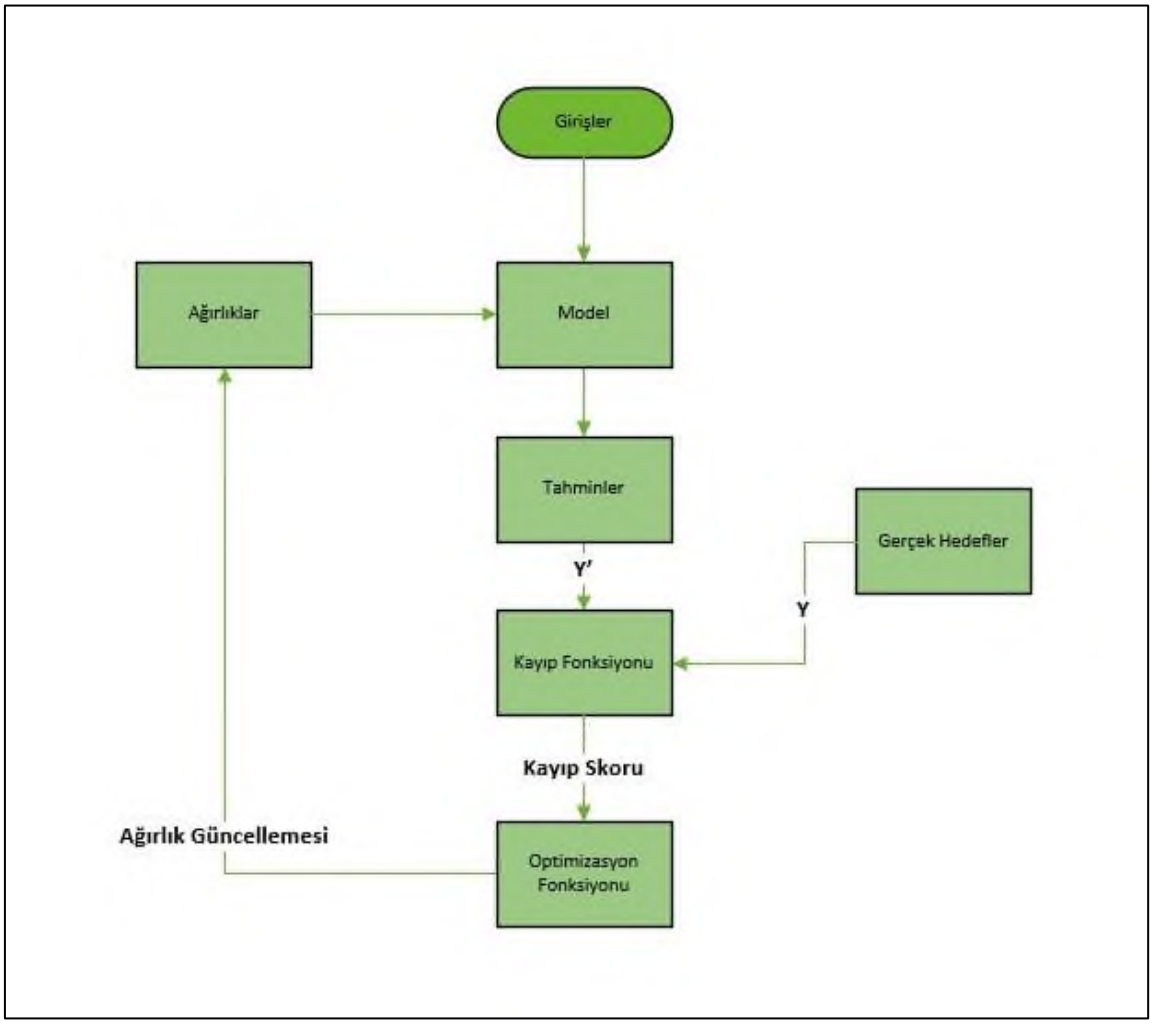
Bu teknik, her katmanın girdisini normalize ederek sinir ağının kararlılığını, performansını ve hızını artırmayı hedefler. Ayrıca, başlangıç ağırlıklarına olan duyarlılığı azaltmaya yardımcı olur.

Seyreltme (Dropout)

Başka bir düzenleme yöntemi olan seyreltme, eğitim sırasında bir dizi nöronu rastgele devre dışı bırakır. Bu, ağın tek bir nörona aşırı bağımlı olmasını engelleyerek aşırı uyumu azaltmaya yardımcı olur.

Derin öğrenme modeli yapay sinir ağlarının çok katmanlı yapısıyla tasarlanmaktadır. İlk katman girdi verilerini alır ve işler. Derin öğrenme modelinin tasarlanmasına bağlı olarak girdi katmanından sonra birçok gizli katman bulunabilmektedir. Bu katmanlar özellikleri

öğrenmek ve veriyi temsil etmek için kullanılmaktadırlar. Her nöron bir aktivasyon fonksiyonu ile işlenerek modelin karmaşıklığı artırılmış olur. Modelin son katmanı olan çıkış katmanı ise sonucu vermektedir. Modelin çıktısı (tahmin) ile gerçek değerler arasındaki fark kayıp fonksiyonları ile hesaplanmaktadır. Modelde kayıp fonksiyonlarından elde edilen kayıp değerini minimuma indirmek için geriye yayılım ve optimizasyon algoritmaları kullanılmaktadır [38]. Şekil 4.4.'te bir derin sinir ağının çalışma şekli şematik olarak gösterilmiştir.

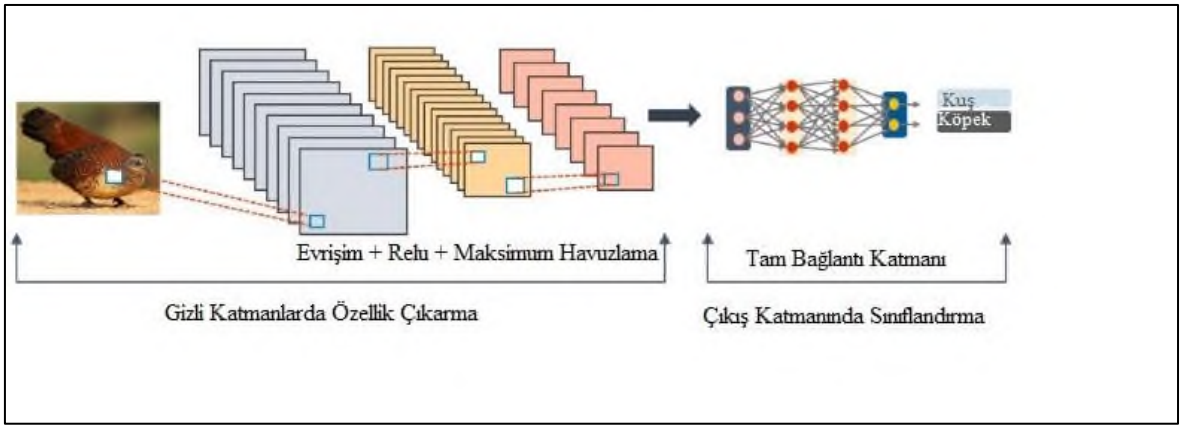


Şekil 4.4. DSA Çalışma Şekli

Derin öğrenme, tıbbi görüntü analizi problemlerin çözümünde de oldukça etkilidir. Derin Sinir Ağları kullanılarak oluşturulan modeller sayesinde tıbbi görüntüler üzerinden hastalıklar teşhis edilebildiği gibi organ ve tümörler bölütlenip tümörler sınıflandırılabilir. Görüntü işleme problemlerinde en çok kullanılan derin öğrenme modellerinden biri de Evrişimli Sinir Ağları (ESA)'dır.

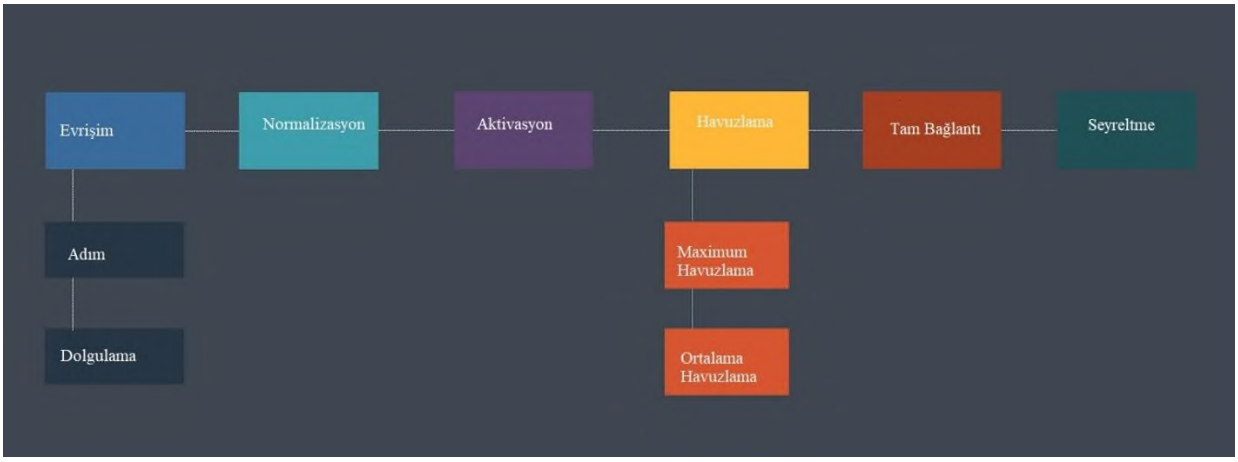
4.4. Evrişimli Sinir Ağları (ESA)

ESA görüntü işleme ve nesne tanıma gibi problemlerin çözümünde oldukça etkili ve sık kullanılan bir yapay sinir ağı modelidir. Evrişimli Sinir Ağı (ESA), bir giriş görselini alabilen, görsel içindeki çeşitli özelliklere ağırlık değerleri ve bias değerleri atayabilen, gösterilenler arasında ayrım yapabilen bir derin öğrenme sistemidir. Diğer sınıflandırma algoritmaları ile karşılaştırıldığında ESA önemli ölçüde az ön işleme gerektirir. Temel yaklaşımlar filtrelerin elle tasarlanmasını gerektirirken, ESA bu filtreleri/özellikleri yeterli eğitimle öğrenebilir [39]. Şekil 4.5.'te ESA Modeli çalışma prensibi verilmiştir.



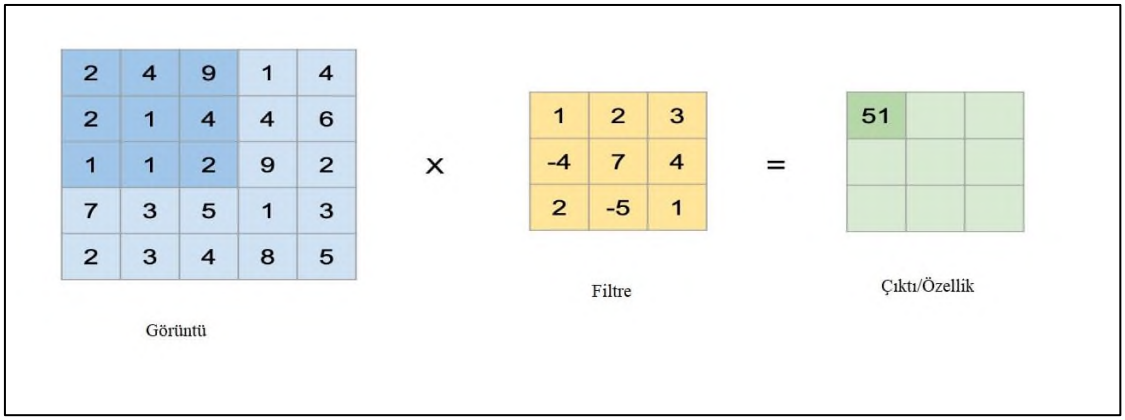
Şekil 4.5. ESA Modeli Çalışma Prensibi

ESA'lar, verilerden özellikleri işleyen ve çıkaran birden çok katmana sahiptir. Bu katmanlar Şekil 4.6.'te verilmiştir.



Şekil 4.6. ESA Katmanları [40]

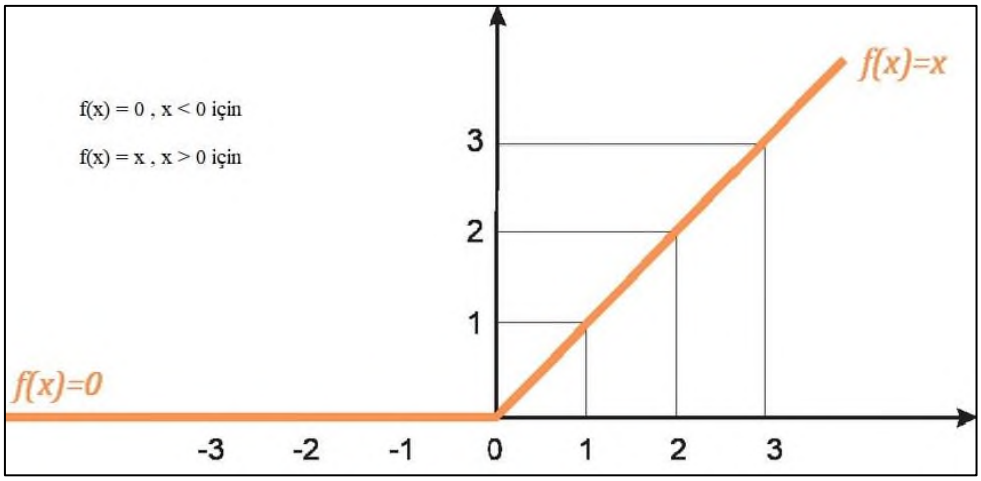
Evrişim Katmanı: Evrişimli sinir ağlarında işlenecek görseller matrisler ile ifade edilmektedir ve tüm işlemler matris işlemleri olarak gerçekleştirilmektedir. ESA, evrişim işlemlerini gerçekleştirmek için birkaç filtreye sahip bir evrişim katmanına sahiptir. Evrişim katmanında görsel üzerinde görsel boyutundan daha küçük bir filtre gezdirilerek bu görselden belirli özellikler çıkartılmaktadır. Filtrenin görsel üzerinde ne kadar kaydırılacağı adım (stride) parametresi ile belirlenmektedir. Daha sonra dolgulama (pixel ekleme) yöntemi kullanılarak elde edilen özellik matrisinin boyutu arttırılır. Eklenen pikseller sonucu etkilememesi için sıfır değerindedirler. Birden çok filtre ile birçok öznitelik çıkartılıp özellik haritaları elde edilir. Şekil 4.7.'da evrişim işlemi gösterilmiştir.



Şekil 4.7. Evrişim İşlemi [40]

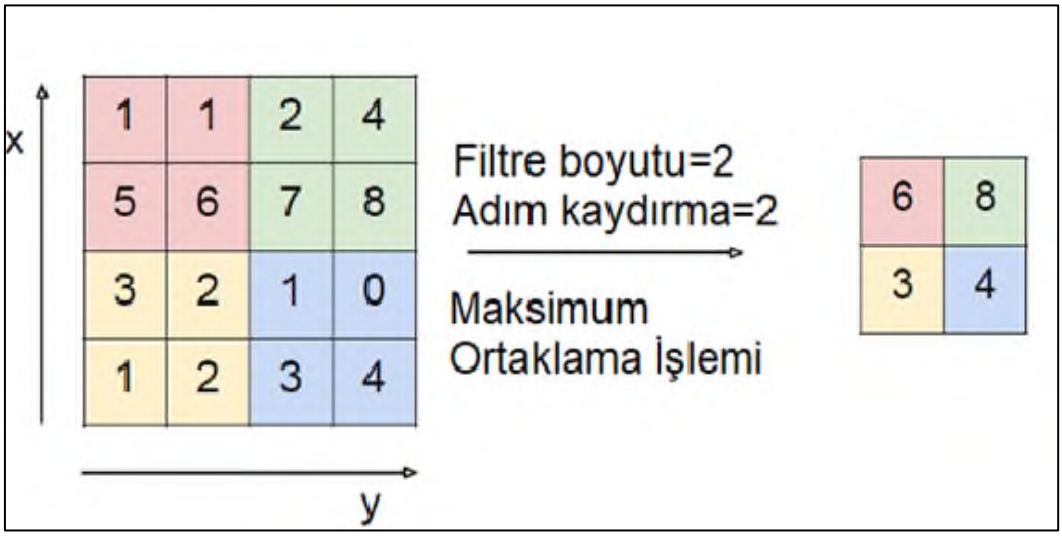
Normalizasyon Katmanı: Verileri standart hale getirmek için kullanılan bir ön işleme adımıdır. Normalizasyon, verileri 0 ve 1 arasında yeniden ölçekler. Bir sinir ağının katmanları arasında yapılan normalleştirme işleminde matrisin tüm bileşenleri en büyük değere bölünerek matrisin 0 ile 1 arasında değerler alması sağlanır. Normalizasyon, girdileri daha dengeli hale getirip ve verilerdeki herhangi bir aykırı değerin etkisini azaltarak eğitimi hızlandırmaya ve öğrenmeyi kolaylaştırmaya yardımcı olur.

Aktivasyon Katmanı (Relu): Yapay sinir ağlarına lineer olmayan özellikleri tanıtmak için aktivasyon fonksiyonları kullanılmaktadır. Relu aktivasyon fonksiyonu negatif olan değerleri 0 değerine değiştirirken pozitif olan değerleri sayının kendisi olarak bırakır. Şekil 4.8.'de Relu Aktivasyon Fonksiyonu verilmiştir.

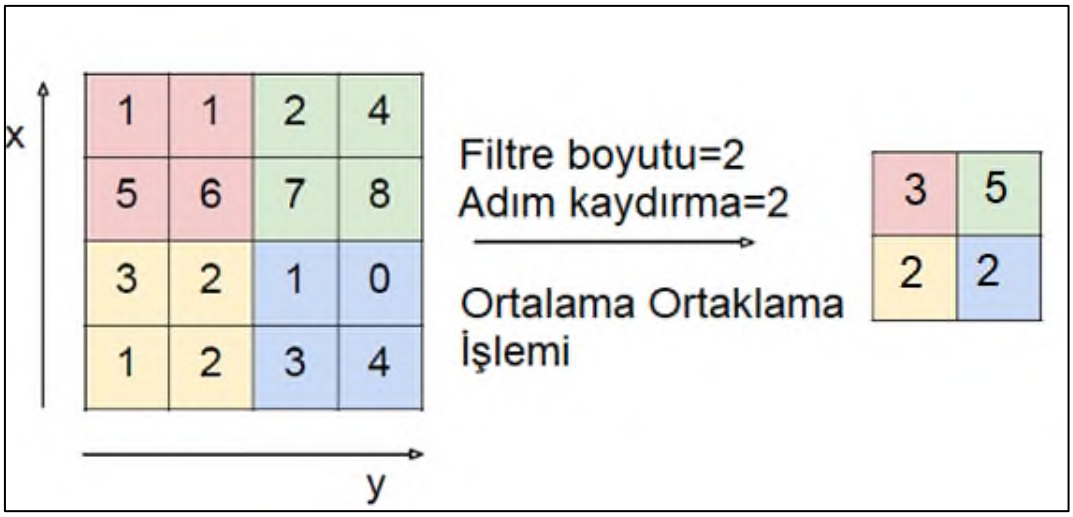


Şekil 4.8. Relu Aktivasyon Fonksiyornu

Havuzlama Katmanı: Havuzlama, özellik haritasının boyutlarını azaltan bir alt örnekleme (down sampling) işlemidir. Havuzlama katmanı daha sonra havuzlanmış özellik haritasından elde edilen iki boyutlu dizileri düzleştirerek tek, uzun, sürekli, doğrusal bir vektöre dönüştürür. Burada yapılan işlem giriş matrisinin kanal sayısını değiştirmeksizin boyut (yükseklik ve genişlik) değerini düşürmektir. Bu sayede hesaplamalardaki karmaşıklık azaltılmış olmaktadır. Şekil 4.9.'de en yaygın iki havuzlama türünden biri olan maksimum havuzlama ve Şekil 4.10.'da ortalama havuzlama gösterilmektedir.

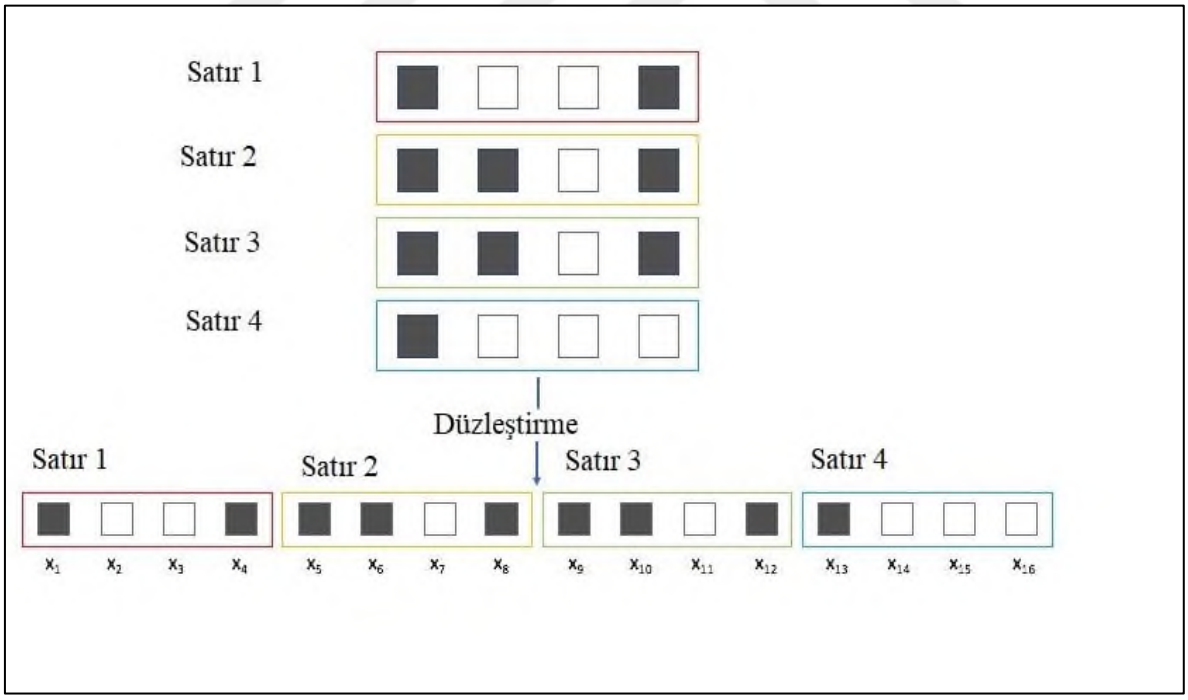


Şekil 4.9. Maksimum Havuzlama [41]



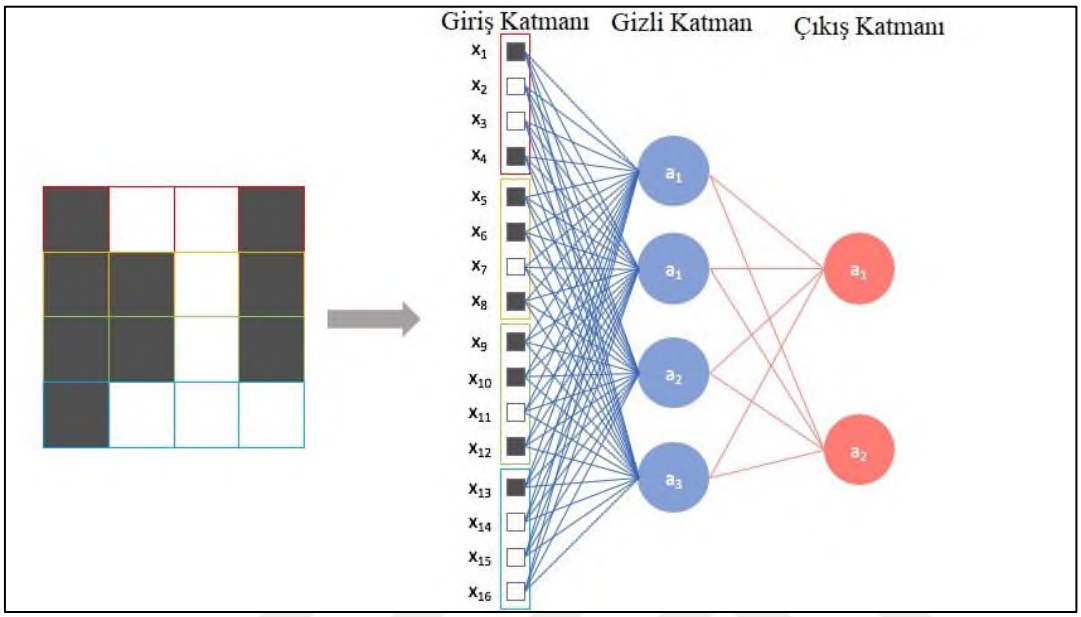
Şekil 4.10. Ortalama Havuzlama [41]

Düzleştirme Katmanı: Düzleştirme katmanı gizli katmanlardan gelen (özellik çıkarma) matris tipindeki verileri dizi tipine çevirme işleminin gerçekleştirildiği katmandır. Şekil 4.11.'da Düzleştirme Katmanı gösterilmiştir.



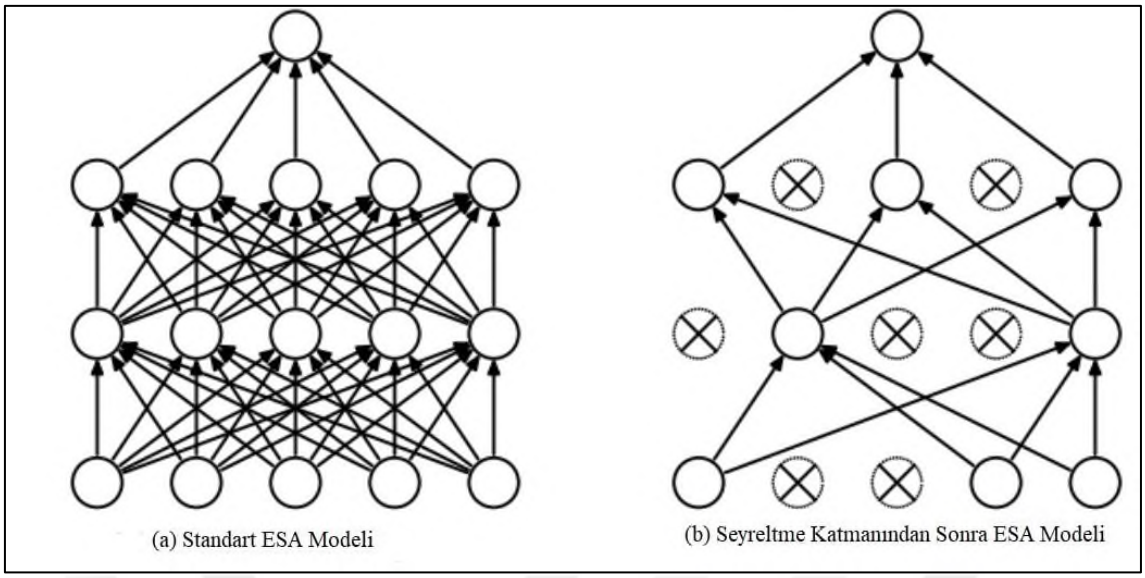
Şekil 4.11. Düzleştirme Katmanı [41]

Tam Bağlantı Katmanı: Evrişimli sinir ağının son ve en önemli katmanıdır. Düzleştirme katmanından aldığı veriler ve sinir ağını kullanarak öğrenme işlemini gerçekleştirir. Şekil 4.12.'de Tam Bağlantı İşlemi şematik olarak verilmiştir.



Şekil 4.12. Tam Bağlantı İşlemi [41]

Seyreltme (DropOut) Katmanı: ESA'da eğitim işlemi büyük veriler ile yapıldığından bazı durumlarda ağ aşırı örenebilir. Ağın aşırı öğrenmesini engellemek için bu seyreltme işlemi uygulanır [42]. Bu katmanda uygulanan temel mantık ağın bazı düğümlerinin kaldırılmasıdır. Kaldırma işlemi herhangi bir hesaplamaya dayanmaz, hangi düğümlerin kaldırılacağı rasgele belirlenir. Örneğin seyreltme oranı 0,5 olarak seçilmiş ise bu, her eğitim yinelemesi sırasında o katmandaki nöronların yarısının rastgele devre dışı bırakılacağı anlamına gelir. Şekil 4.13 (a)'da ağın standart yapısı yer alırken, (b)'de seyreltme katmanı sonrasındaki yapısı şematik olarak yer almaktadır.



Şekil 4.13. (a) Standart Esa Modeli, (b) Seyreltme Katmanı Sonrası ESA Modeli [43]

4.5. UNet mimarisi

UNet asıl amacı görüntüleri bölütlemek olarak tasarlanmış bir evrişimli sinir ağı modelidir ve mimarisi Şekil 4.14'te gösterilmiştir. Mimari olarak yapısı U harfine benzediğinden bu ismi almıştır. UNet'in en büyük artısı eğitim süresinin diğer modellere göre kıyasla daha kısa olmasıdır. Ayrıca daha basit bir yapıya sahip olup daha az parametreye ihtiyaç duymaktadır. Bunun yanında diğer ağlara göre nispeten sınırlı sayıda veriye ihtiyaç duymaktadır. Bir diğer üstünlüğü ise yüksek kalitede (piksel düzeyinde) bölütleme performansı göstermesidir [44].

Simetrik bir yapıya sahip olan UNet'te, alt örnekleme hattı ile üst örnekleme hattı arasında birleştirme bağlantıları mevcuttur. Bu birleştirme bağlantıları olmasa UNet aslında bir 'Autoencoder'dır. Bir adet daralan alt örnekleme hattından ve genişleyen üst örnekleme hattından meydana gelmektedir.

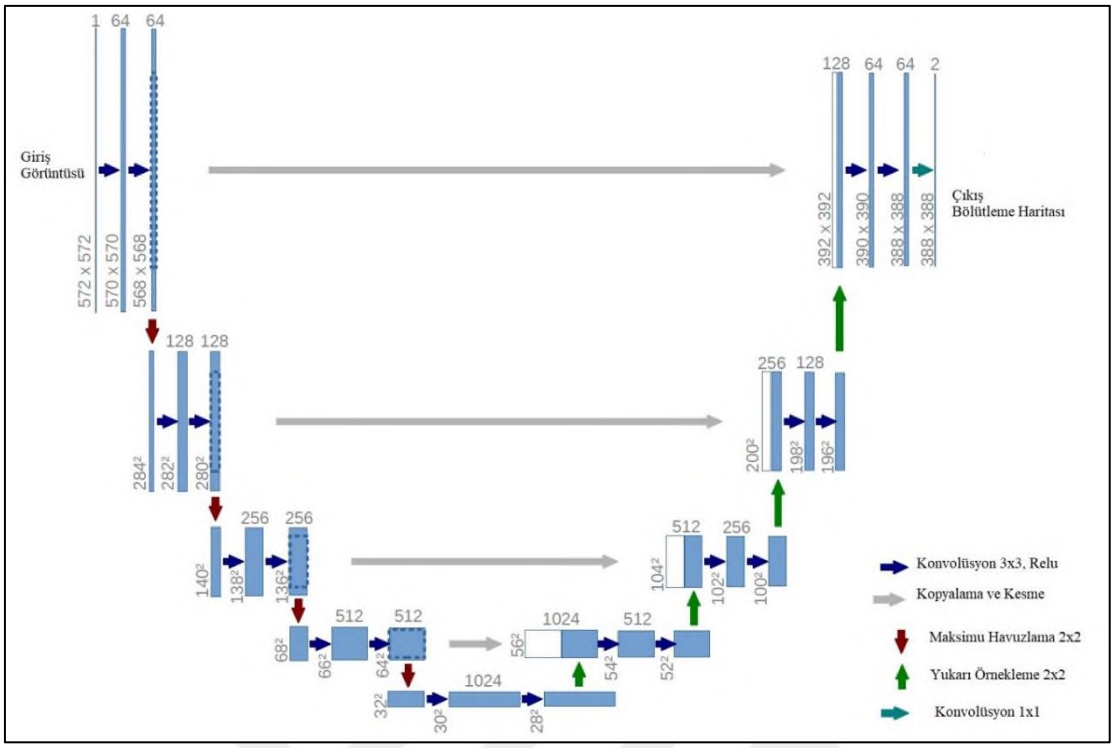
Alt örnekleme hattı dört adet bloktan oluşmaktadır. Her bir blok ikişer adet (3,3) konvolüsyon ile relu aktivasyon fonksiyonundan ve birer adet (2,2) maksimum havuzlama katmanından oluşmaktadır. Örnek olarak ilk bloğun çalışmasını anlatacak olursak (572,572) görüntü ilk (3,3) konvolüsyon işleminden sonra (570,570) boyutuna düşer çünkü konvolüsyonda adım (stride) değeri bir olarak tanımlanmıştır. Bu bloktaki ikinci (3,3) konvolüsyon işlemi sonucu görüntü boyutu (568,568) olur. Daha sonra (2,2) maksimum havuzlama işlemi yapılır ve bu işlemde doldurma (padding) sıfır olduğundan görüntü

boyutunu yarıya (284,284) düşürür. Aşağı örnekleme hattında görüntü boyutu bu şekilde düşerken özellik haritalarının sayısı her maksimum havuzlama işleminden sonra ikiye katlanır. Yani her alt örnekleme bloğunda boyut azalırken özellik bilgilerinin sayısı artar çünkü alt örnekleme hattının amacı, bölütleme yapabilmek için görüntünün içeriğini elde etmektir. Sonraki üç alt örnekleme bloğunda da bu matris işlemleri ve özellik çıkarma işlemleri devam eder. Sonuç olarak alt örnekleme hattında "Ne" bilgisi artarken, "Nerede" bilgisi azalır.

Üst örnekleme hattı da yine dört bloktan meydana gelmektedir. Her bir blok bir adet alt örnekleme hattından gelen özellik haritası ile birleştirme katmanı, ikişer adet (3,3) konvolüsyon katmanı ile aktivasyon fonksiyonu ve bir adet (2,2) yukarı örnekleme katmanından meydana gelir. Örnek olarak anlatmak gerekirse üst örnekleme katmanının son bloğunda önceki bloktan gelen (196,196) boyutlarındaki görüntü önce (392,392) boyutuna üst örneklenir. Yani boyutu iki katına çıkar. Sonra ilk (3,3) konvolüsyon işlemi ile boyut (390,390)'a ikinci konvolüsyon işlemi ile boyutu (388,388)'e düşer. Görüldüğü gibi üst örnekleme katmanında boyut arttırılırken özellik bilgileri alt örnekleme hattından gelen özellik bilgileri ile birleştirilir. Üst örnekleme hattının görevi alt örnekleme hattından gelen özellikleri kullanarak yüksek çözünürlüklü bölütleme haritasını oluşturmaktır. Birleştirme katmanının görevi kaybolan "Nerede" bilgisini yeniden kazanmaktır [45].

Üst örnekleme hattının dördüncü bloğunun sonunda (1,1) ek bir konvolüsyon daha uygulanır. Bunun amacı bölütleme haritasını gerekli sayıda sınıfa indirgeyerek bölütlenmiş görüntüyü üretmektir [44]. Görüldüğü gibi iki kanallı bir bölütleme haritası oluşturulmuştur.

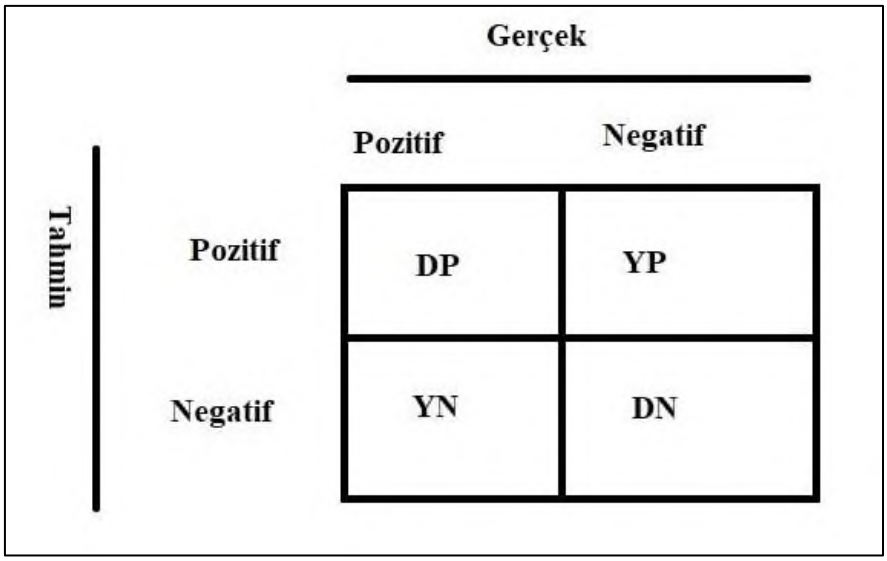
2015 yılında yayınlanan makalesinden bugüne, UNet'in tıbbi görüntü işleme alanında tercih edilir ve kullanılır olması konusunda büyük bir artış görmüştür. Ayrıca UNet'in sadece tıbbi görüntü işleme konusunda değil diğer birçok konuda da kullanımı o günden bugüne arttırmıştır.



Şekil 4.14. UNet Mimarisi [45]

4.6. Karmaşıklık Matrisi

Karmaşıklık matrisi, sınıflandırma problemlerinin çözümünde kullanılan derin öğrenme modellerinin doğruluk sonuçlarını ölçmek için kullanılan tekniklerden biridir. Karmaşıklık matrisi, sınıflandırma modelinizin gösterdiği performansı görsel olarak simgeleyen şematik bir matristir. Modelin tahminleri (çıktılar) ile gerçek değerler arasındaki ilişkiyi göstermeye yarar. İki sınıflı karmaşıklık matrisi Şekil 4.15'te görülmektedir.



Şekil 4.15. İki Sınıflı Karmaşıklık Matrisi

Doğru Pozitif (DP): Gerçek pozitif değerlerin doğru biçimde pozitif olarak tahmin edildiği durumlardır. Yani doğruya doğru demek olarak tanımlanabilir.

Doğru Negatif (DN): Gerçek negatif değerlerin doğru biçimde negatif olarak tahmin edildiği durumlardır. Yani yanlışa yanlış demek olarak tanımlanabilir.

Yanlış Pozitif (YP): Gerçek negatif değerlerin pozitif olarak tahmin edildiği durumlardır. Yani doğruya yanlış demek olarak tanımlanabilir.

Yanlış Negatif (YN): Gerçek pozitif değerlerin negatif olarak tahmin edildiği durumlardır. Yani yanlışa doğru demek olarak tanımlanabilir.

Ölçülebilir ve karşılaştırılabilir standartta doğruluk sonuçları elde etmek için karmaşıklık matrisini oluşturan bileşenler kullanılır ve bazı metrikler hesaplanır.

Doğruluk (Accuracy), bir derin öğrenme modelinin hangi oranda doğru tahmin yaptığını belirten temel metriktir. Bunu belirlerken modelin doğru tahminlerinin, tüm veri noktalarına göre yüzdesini hesaplar. Yüksek doğruluk oranı modelin performansının iyi olduğunun göstergesidir. Fakat, doğruluk her zaman en uygun ölçüm olmayabilir. Doğruluk karmaşıklık matrisi kullanılarak aşağıdaki formül ile hesaplanır.

Doğruluk=(DP+DN) / (DP+DN+YP+YN) (4.2)

Kesinlik (Precision), bir derin öğrenme modelinin pozitif çıktılarının (tahmin ettiği değerlerin) kaç adedinin gerçekten pozitif olduğunu gösteren metriktir. Aşağıdaki formül ile hesaplanır.

Kesinlik=DP/(DP+YP) (4.3)

Duyarlılık (Recall), bir derin öğrenme modelinin pozitif olarak tahmin etmesi gereken sonuçların ne kadarını pozitif olarak tahmin ettiğini gösteren metriktir. Bu anlamda duyarlılık modelin gerçek pozitifleri ne kadar yakaladığını ölçer. Aşağıdaki formül ile hesaplanır.

Duyarlılık=DP/(DP+YN) (4.4)

F1 Skor (F1 Score), kesinlik metriği ile duyarlılık metriği arasındaki ilişkiyi ölçmeye yarayan metriktir. Kesinlik ve Duyarlılık değerlerinin harmonik ortalamasını göstermektedir. Aşağıdaki formül ile hesaplanır.

F1 Skor=2*(Kesinlik*Duyarlılık) / (Kesinlik+Duyarlılık) (4.5)

Dice benzerlik katsayısı (Dice similarity coefficient), etiketlenmiş bölgeler ile tahmin edilmiş bölgeler arasındaki örtüşmeyi ölçen bir metriktir. Dice benzerlik katsayısı 0 ile 1 aralığında bir değer alır. Ölçülen kesişme alanı ile birleşme alanı aynı ise bu katsayının değeri 1 olur ve bu durum kusursuz bir bölütlemenin gerçekleştiğini gösterir. Aşağıdaki formül ile hesaplanır.

$$\text{Dice Benzerlik Katsayısı} = 2 * |A \cap B| / (|A| + |B|) \quad (4.6)$$

Dice kayıp değeri (Dice Loss); dice kayıp fonksiyonu, dice benzerlik katsayısına dayanmaktadır. Dice benzerlik katsayısından elde edilen değer 1'den çıkarılarak dice kayıp değeri elde edilmiş olur. Bu metrikte bölütlemenin ne kadar başarılı yapıldığını gösteren bir metriktir. Dice kayıp değeri ne kadar küçük ise dice benzerlik katsayısı o kadar büyük

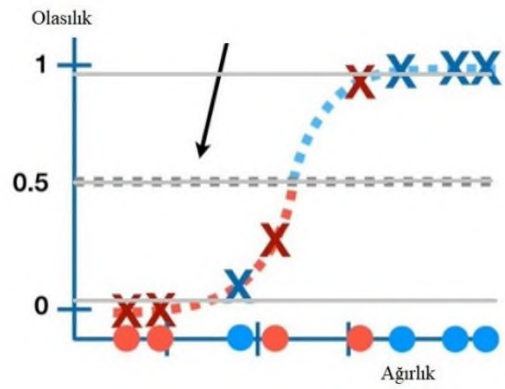
demektir ve bölütleme görevi bir o kadar o kadar iyi gerçekleştirilmiş demektir. Dice kayıp değeri aşağıdaki gibi hesaplanır.

$$\text{Dice Kayıp Değeri} = 1 - 2|A \cap B| / (|A| + |B|) \quad (4.7)$$

4.7. ROC Analizi

ROC (A Receiver Operator Characteristic) analizi sınıflandırma problemlerinde kullanılan etkili analiz yöntemlerinden biridir. İkinci Dünya Savaşı sırasında radar kullanımı ile beraber Amerikan askerlerinin radar ekranında gördükleri sinyali düşman (Alman) uçağı, dost uçak veya parazit olarak ayırma ihtiyaçları ile ortaya çıkmıştır. Radar kullanımı arttıkça ve radar teknolojileri geliştikçe algılama doğruluğunu nicel olarak ölçecek standart bir metodoloji gereksinimi de artmıştır. İlk olarak sinyal algılama teorisinde kullanılmış olsa da şu anda birçok alanda (radyoloji, tıp, makine öğrenmesi vb.) kullanılmaktadır [46].

Şekil 4.16'da görüldüğü üzere bir sınıflandırma probleminde farklı eşik değerler için farklı karmaşıklık matrisleri elde edilebilir. Bu problemde obez fareler mavi, obez olmayan fareler ise kırmızı renkte gösterilmiştir ve farelerin ağırlıkları x ekseninde yer almaktadır. Değeri 0 ile 1 arasında değişen y ekseninde ise farelerin obez olma olasılıkları belirtilmiştir. Mesela 0,5 eşik değeri için 3 obez fare obez (DP) olarak, obez olmayan 3 fare obez değil (DN) olarak, obez olmayan 1 fare obez (YP) olarak, obez olan 1 fare ise obez değil (YN) olarak sınıflandırılmış ve karmaşıklık matrisi hazırlanmıştır. Görüldüğü üzere farklı eşik değerlerde farklı karmaşıklık matrisleri oluşmaktadır. 0 ile 1 arasında birçok eşik değer seçilebilir fakat bunlardan hangisinin seçilmesi gerektiğini belirlemek karar verilmesi gereken bir konu olarak durmaktadır.



Eşik=0,5

3	1
1	3

Eşik=0,95

3	0
1	4

Eşik=0,05

4	2
0	2

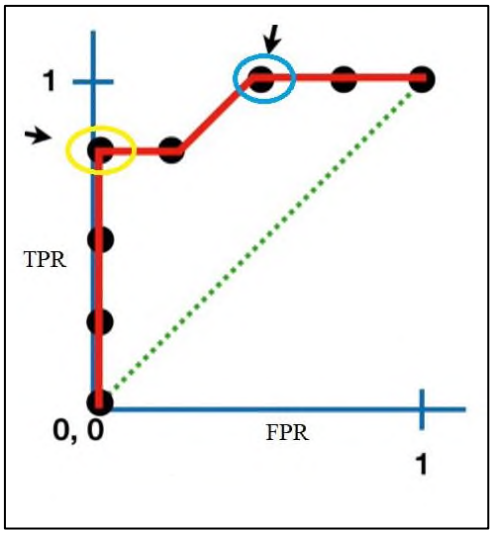
Şekil 4.16. Eşik değer – Karmaşıklık Matrisi ilişkisi [47]

Hangi eşik değerin seçileceği kararını vermek için; bir sınıfa ait olma derecesini yansıtan bir puan veren olasılıksal sınıflandırıcılarda puan eşiği değiştirilerek bir eğri oluşturulabilir. Bu eğri bize farklı eşiklerde elde edilen karmaşıklık matrislerini özetleyen bir eğri olacaktır. Buna da ROC eğrisi adı verilir. Eğrinin x eksenini FPR (False Positive Rate – Yanlış Pozitif Oranı), y eksenini ise TPR (True Positive Rate – Doğru Pozitif Oranı) oluşturur ve bu değerler aşağıdaki formüller ile hesaplanır.

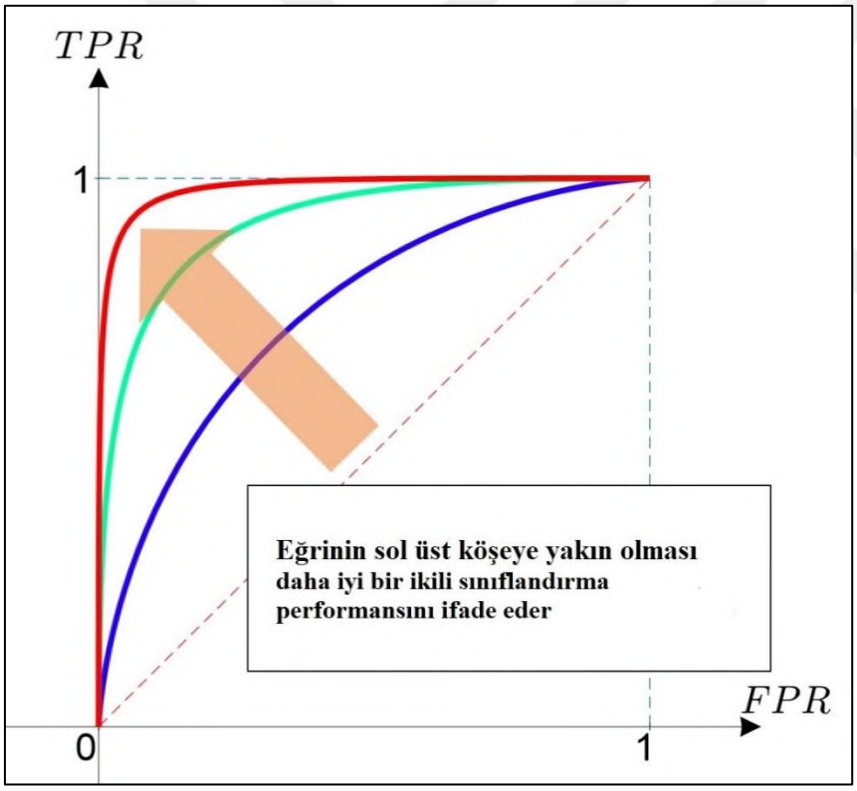
$$TPR = \text{Duyarlılık} = DP/(DP+YN) \tag{4.8}$$

$$FPR = \text{Özgüllük} = YP/(YP+DN) \tag{4.9}$$

Eğri oluşturulduktan sonra hangi eşik değerin seçileceğine yanlış pozitif toleransına göre karar verilebilir. Örneğin Şekil 4.17'deki eğriyi değerlendirdiğimizde sarı ve mavi çerçeveli eşik değerden birinin seçilmesinin uygun olacağı anlaşılmaktadır. Çünkü ikisi arasındaki noktada sarıya göre TPR aynı iken FPR daha fazladır. Mavi ile sarı noktalar arasındaki seçim ise FPR toleransımıza göre belirlenebilir. Mavide sarıya göre TPR fazla iken FPR de fazladır. FPR toleransımız yüksek ise mavi nokta seçilebilir.



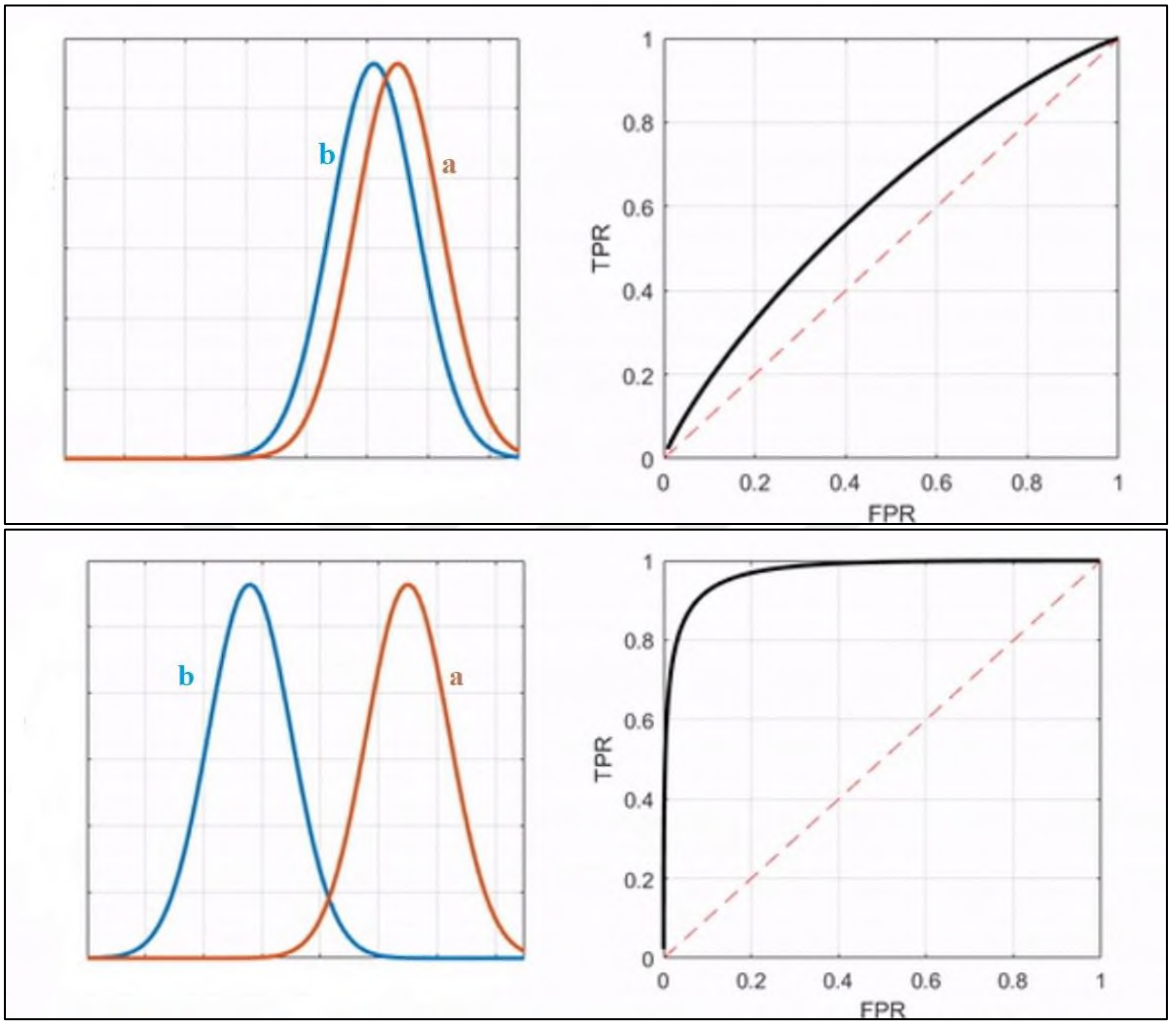
Şekil 4.17. Eşik Değer seçimi [47]



Şekil 4.18. ROC eğrisi gösterimleri [47]

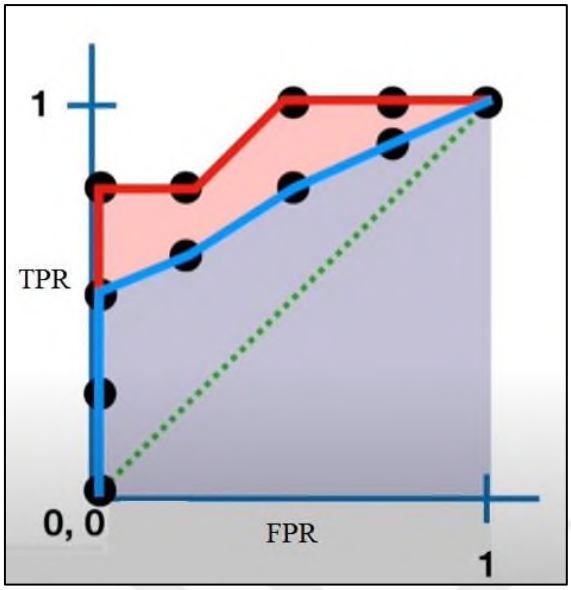
Şekil 4.18'da ROC Eğrisi temsili gösterimleri verilmiştir. Mavi kesikli çizgiler mükemmel sınıflandırmayı tarifler. Eğrinin sol üst köşeye yakın olması daha iyi bir ikili sınıflandırma performansını ifade eder. Kırmızı kesikli çizgiler 45 derece açıyla yer almaktadır ve bu şekilde verilen bir eğri modelin sınıflandırma performansının rastgele olduğunu belirtir.

Yani model bir sınıflandırma yapmayıp rasgele karar vermektedir. Eğrinin konumu ile sınıflandırma performansı arasındaki ilişki Şekil 4.19'de gösterilmiştir.



Şekil 4.19. ROC eğrisi ve sınıflandırma performansı ilişkisi [47]

Çizilen ROC eğrisi altında kalan alan AUC olarak adlandırılır ve bu alan arttıkça da modelin doğru tahmin yapma olasılığı (oranı) artar. AUC değeri 0,5 ile 1 arasında değişir. Şekil 4.19'da görüldüğü gibi rastgele performans gösteren eğride eğri altında kalan alan 0,5 tir. Eğri sol üste kaydıkça altta kalan alan artar. Mükemmel eğride ise bu alan 1 olarak hesaplanır. 0,5'in üzerindeki değerlerde model az da olsa doğru tahmin üretmektedir. Eğri altındaki alan 1'e doğru gittikçe (0,8 – 0,9 vb.) modelin tahminleri de iyileşir. Örneğin Şekil 4.20'de görülen iki eğriden kırmızı olan ikili sınıflandırma konusunda daha başarılıdır. Yani iki farklı yöntem ile elde edilen noktalardan eğriler çizilip bu şekilde karşılaştırılabilir. Bu eğrilerden kırmızı lojistik regresyon mavi rasgele orman ile elde edilmiş olsun. Bu durumda lojistik regresyon kullanılması daha uygundur denilebilir.



Şekil 4.20. AUC ve sınıflandırma performansı ilişkisi [47]

5. YÖNTEM VE ÇALIŞMA

5.1. Kullanılan Materyaller

Bu alt bölümde tez çalışmasında kullanılan yazılım ve kütüphaneleri kısaca tanıtılmaktadır.

ITK-SNAP, açık kaynak olarak geliştirilmiş ve çoklu platform desteği olan kullanıcıların üç boyutlu tıbbi görüntülerde gezinmesine ve tıbbi görüntüleri bölütlemesine olanak tanıyan ücretsiz bir yazılımdır. Bu yazılım sayesinde DICOM ve NIfTI uzantılı tıbbi görüntüler incelenebilir, işlenebilir ve üç boyutlu manuel bölütleme işlemleri gerçekleştirilebilir.

3D SLICER, üç boyutlu tıbbi görüntüleri analiz etme ve işlemede kullanılan açık kaynaklı ücretsiz bir yazılımdır. Bu yazılım sayesinde de aynı şekilde DICOM ve NIfTI uzantılı tıbbi görüntüler incelenebilir, işlenebilir ve üç boyutlu manuel bölütleme işlemleri gerçekleştirilebilir.

MONAI, tıbbi görüntülemede araştırmayı ve klinik iş birliğini hızlandırmak için oluşturulmuş açık kaynaklı, ücretsiz ve iş birliğine dayalı bir dizi çerçevedir. Amaç tıbbi görüntülemenin ve derin öğrenme araştırmalarının neredeyse her düzeyinden yararlanabileceği sağlam bir yazılım çerçevesi oluşturarak inovasyonun ve klinik dönüşümün hızını artırmaktır.

Monai Veri Dönüşüm Modülleri, tıbbi görüntülerin analizi öncesinde uygulanması gereken veri ön işleme adımlarını destekler. Bu adımlar arasında yeniden boyutlandırma, normalizasyon, kırpma, döndürme ve ölçekleme gibi çeşitli veri artırma teknikleri bulunmaktadır. Ayrıca, sağlık verilerinin güvenli ve etkin bir şekilde yönetilebilmesi için bu dönüşümler özelleştirilebilir.

Derin Öğrenme Model Mimarisi Desteği, özellikle bölütleme işlemleri için yaygın olarak kullanılan UNet, UNetr ve Attention UNet gibi üç boyutlu veri işleme desteği olan çok sayıda model bulundurmaktadır. Bu modeller, tıbbi görüntü analizi için gereken derin mimarilerde veri hacmini yönetebilmekte ve tıbbi görüntülerin çok boyutlu yapısına uygun bir şekilde çalışabilmektedir.

Kayıp fonksiyonları ve değerlendirme metrikleri, tıbbi görüntülerin yapısal analizinde sıkça kullanılan bileşenlerdir. MONAI çerçevesi Dice kayıp fonksiyonu, Focal kayıp fonksiyonu, Tversky kayıp fonksiyonu ve daha birçok kayıp fonksiyonunu kullanıcılarına sunar. Bu fonksiyonlar, bölütleme süreçlerinde organ veya tümör sınırlarının doğru tespiti için kritik öneme sahiptir. Bunun yanı sıra, modelin doğruluğunu ve performansını ölçmek için Dice Benzerlik Katsayısı gibi tıbbi analizlerde yaygın kullanılan metrikleri sağlamaktadır. MONAI'nin bünyesindeki tüm bu bileşenler ayrıntılı olarak açıklanmış ve kaynak kodları ile beraber sunulmaktadır.

Optimizasyon araçları kullanılarak ağın eğitimi için gerekli parametreler en uygun şekilde seçilebilir. Örneğin 'LearningRateFinder' eğitim oranı aralık testi, bir ön eğitim çalışması sırasında öğrenme oranını iki sınır arasında doğrusal veya üstel bir şekilde arttırabilmektedir. Bu da ağın farklı öğrenme oranları aralığında ne kadar iyi eğitilebileceği ve en uygun öğrenme oranının ne olduğu hakkında değerli bilgiler sağlamaktadır.

MONAI, tıbbi görüntü analizi alanında çalışan araştırmacılar için kapsamlı bir derin öğrenme kütüphanesi olarak önemli bir avantaj sağlamaktadır. Kütüphanenin esnekliği, özelleştirilebilir olması ve tıbbi görüntüleme süreçlerine uygunluğu, bu alandaki çalışmaların hızlanmasına ve doğruluğunun artmasına katkıda bulunmaktadır.

PYTORCH, derin öğrenme modelleri oluştururken grafik işlem birimlerinin gücünü kullanabilen Pytorch oldukça zengin bir kütüphanesidir.

VISUAL STUDIO CODE, Windows, Linux ve MacOS üzerinde çalışabilen ve Microsoft tarafından geliştirilen gelişmiş bir kaynak kodu düzenleyicisidir. En büyük avantajı açık kaynak kodlu olması ve ücretsiz sunulmasıdır. Bunun yanında diğer avantajları zengin dokümantasyonu sayesinde öğrenilmesinin kolay olması, editör içinde birden fazla terminal penceresi açılabilmesi sayesinde geliştiricilerin hem kod yazmasını hem de terminal komutlarını aynı yerden çalıştırmasını sağlaması, kod yazarken akıllı öneriler ve otomatik tamamlamalar sunarak kod yazımını hızlandırıp ve hataları azaltması, hataları ve uyarıları kod yazımı sırasında gerçek zamanlı olarak göstermesi ve gelişmiş hata ayıklama (debugging) desteğinin olmasıdır.

BİLGİSAYAR DONANIMI, modelin çalışması için aşağıdaki özelliklerde donanım kullanılmıştır. Pytorch ve Monai kütüphanelerinin CUDA desteği olmasına rağmen bilgisayar üzerinde donanım olarak ayrılmış grafik kartı bulunmadığından ayrıca bir GPU kullanılmamış model CPU üzerinde eğitilmiştir.

İşlemci: 12th Gen Intel(R) Core(TM) i7-1260P 2.10 GHz
Bellek: 16,0 GB
İşletim Sistemi: Windows10

5.2. Veri Setinin Seçilmesi ve İncelenmesi

Bu çalışmada öncelikle kullanılacak veri setini belirleme çalışması yapılmıştır. Sonuç olarak da karaciğer tümör bölütlemesi konusunda çalışmak üzere LiTS veri setinin kullanılmasına karar verilmiştir. Bu veri setini seçmenin büyük sebeplerden biri verilerin hem karaciğer hem de tümör etiketleri içeriyor olmasıdır. Diğer bir sebep ise verilerin üç boyutlu olmasıdır. Yani 3 boyutlu bölütleme için uygun bir veri seti olmasıdır. Veri seti içinde yer alan tümörler genellikle heterojen (farklı) yoğunluk ve şekil sergilemektedir. Bu da veri setini daha zorlayıcı hale getirerek modeli farklı çeşitlilikte lezyonun daha kolay tanınır olmasını sağlamaktadır.

LiTS (Liver Tumor Segmentation) yarışmasında kullanılan görüntü verileri, dünya genelindeki yedi farklı klinik merkezden toplanmıştır. Bu merkezler şunlardır:

- Rechts der Isar Hastanesi, Münih Teknik Üniversitesi, Almanya
- Radboud Üniversitesi Tıp Merkezi, Hollanda
- Polytechnique Montréal ve CHUM Araştırma Merkezi, Kanada
- Sheba Tıp Merkezi, İsrail
- İbrani Üniversitesi, İsrail
- Hadassah Üniversitesi Tıp Merkezi, İsrail
- IRCAD, Fransa

LiTS**,** verilerinin ve karşılık gelen maskelerinin dünya çapındaki çeşitli klinik siteler tarafından sağlandığı, halka açık, dengeli ve etiketli tıbbi görüntü veri setidir. Farklı ülkeler ve cihazlardan gelen taramalar, veride çeşitlilik sağlamaktadır. Diğer organlarla

karşılaştırıldığında, mevcut karaciğer veri setleri ya etiketli değildir ya da nispeten az sayıda görüntü içerir. Veri seti toplam 201 adet 3B karın içi BT (Bilgisayarlı Tomografi) taramasından oluşmaktadır. Bu taramalardan 194'ü tümör içermektedir. 7 adet tarama ise tümör içermemektedir. Veri seti içindeki tümörler geniş bir çeşitlilik gösterir. Küçük, lokalize tümörlerden büyük, karaciğeri kaplayan tümörlere kadar birçok lezyon veri seti içinde mevcuttur. Tümörlerin boyutları, şekilleri, yerleşimleri ve yoğunlukları oldukça farklıdır. Bazı tümörler kontrast farkıyla kolayca görülebilirken, bazı tümörler ise sağlıklı dokuya çok benzemekte ve çok zor ayırt edilebilmektedir. Çizelge 5.1 başlıca halka açık karaciğer ve karaciğer tümörü veri setlerinin karşılaştırmasını vermektedir [48].

Tüm veriler anonimleştirilmiştir ve kullanımları için onay gerektirmemektedir. Kişisel bilgileri içerebilecek herhangi bir unsuru ortadan kaldırmak için görüntüler görsel olarak gözden geçirilmiştir.

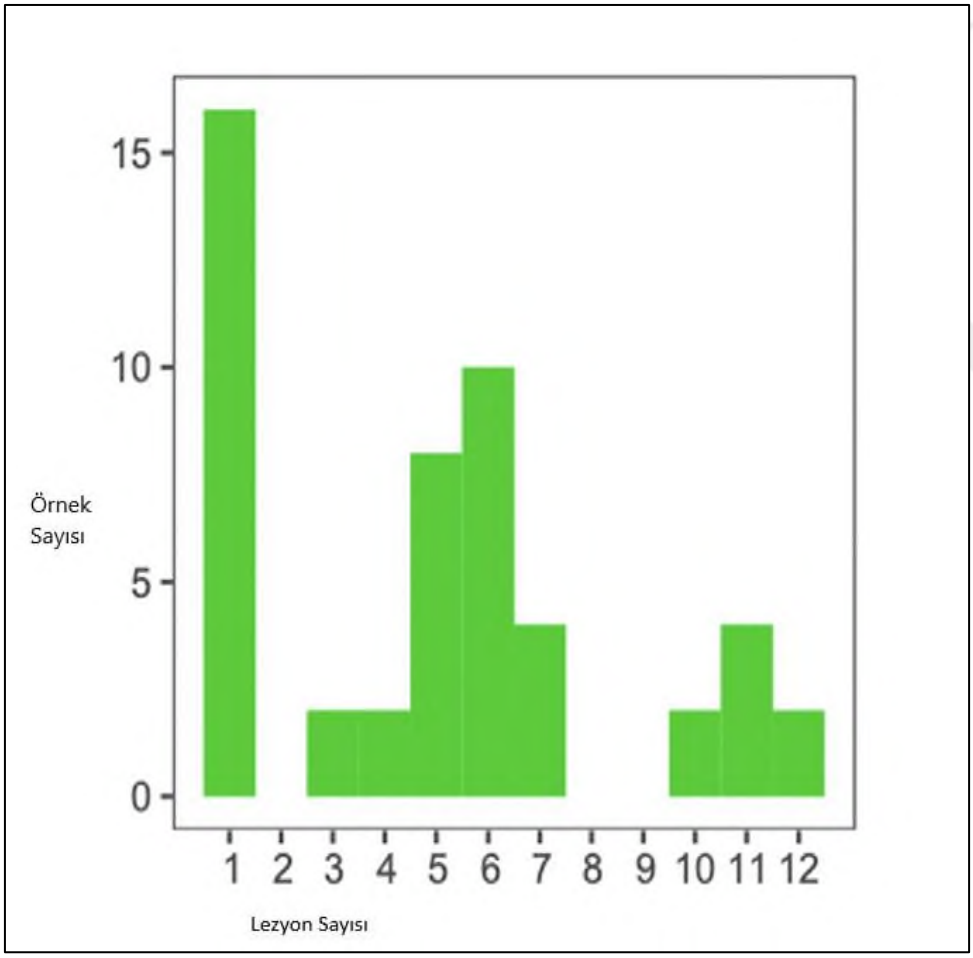
Görüntülere, yalnızca tüm taramaları ortak bir formata dönüştürme işlemi uygulanmıştır. Bunun için Python dilinde NiBabel kütüphanesi kullanılarak görüntüler NIfTI formatına dönüştürülmüştür.

Çizelge 5.1.Karaciğer ve karaciğer tümörü halka açık tıbbi veri setlerinin karşılaştırması [48]

Veri Seti	Kurum	Karaciğer	Tümör	Bölütleme	Sayı	Yöntem
TCGA-LIHC (Erickson ve diğ., 2016)	TCIA	✓	✓	✗	1688	BT, MR, PT
MIDAS (Cleary, 2017)	IMAR	✓	✓	✗	4	BT
3Dircadb-01 ve 3Dircadb-02 (Soler ve diğ., 2010)	IRCAD	✓	✓	✓	22	BT
SLIVER'07 (Heimann ve diğ., 2009)	DKFZ	✓	✗	✓	30	BT
LTSC'08 (Heimann ve diğ., 2009)	Siemens	✗	✓	✓	30	BT
ImageCLEF'15 ()	Bogazici Uni.	✓	✗	✓	30	BT
VISCERAL'16 (Jimenez-del Toro ve diğ., 2016)	Uni. of Geneva	✓	✗	✓	60/60	BT/MRI
CHAOS'19 (Kavur ve diğ., 2021)	Dokuz Eylul Uni.	✓	✗	✓	40/120	BT/MRI
LiTS	TUM	✓	✓	✓	201	BT

Çizelge 5.1 incelendiğinde de anlaşılacağı üzere LiTS veri seti hem nispeten sayıca fazla tıbbi görüntü içermekte ve bu durum da ağın eğitimini pozitif yönde etkilemektedir. Hem de karaciğer ile beraber karaciğer tümörlerini de etiketlenmiş olarak sunulmakta ve bu durum da karaciğer ile beraber karaciğer tümörü bölütlemesine olanak sağlamaktadır. Tüm bu faktörler de veri seti araştırmasında LiTS'i ön plana çıkarmaktadır.

LiTS veri seti içindeki örneklerin çoğu sadece bir lezyon içerirken, büyük bir grup örnekte (2-7) ve birçok örnek de (10-12) lezyon içermektedir. Bu dağılımı gösteren histogram diyagramı Şekil 5.1.'de verilmiştir.

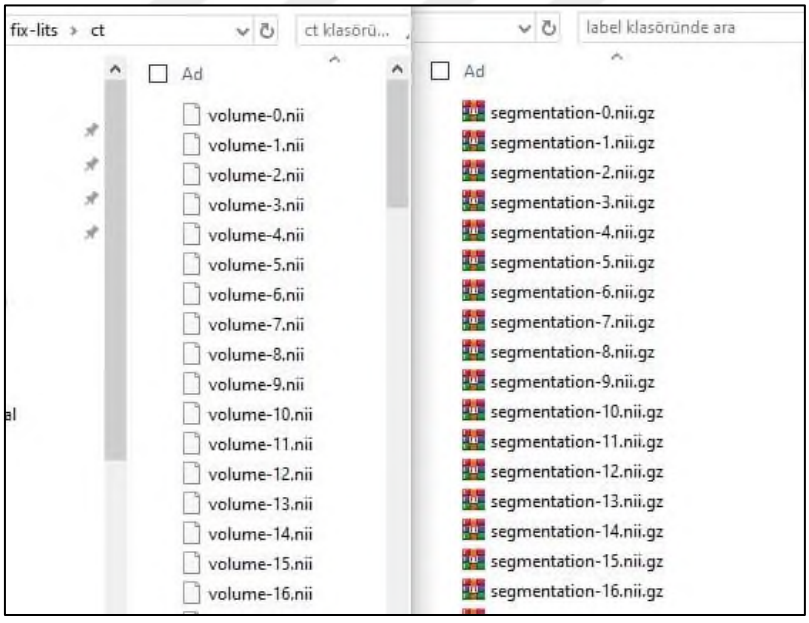


Şekil 5.1. LiTS Veri Seti Lezyon Sayısı ve Örnek Sayısı Histogram Grafiği [48]

LiTS veri seti halka açık olduğu için http://medicaldecathlon.com/dataaws/ web adresinden Task03_Liver bağlantı linkine tıklanıp "Task03_Liver.tar" karaciğer veri seti dosyası indirilmiştir.

LiTS veri seti incelendiğinde içerisinde 201 adet hasta görüntüsüne rastlanmıştır. 131 adet hasta için (512×512) boyutlarında karın içi BT görüntüsünün etiketlenmiş olarak, 70 adet hasta için ise yine (512×512) boyutlarında karın içi BT görüntüsünün etiketlenmemiş olarak bulunduğu belirlenmiştir. Her bir örnek (hasta dosyası) NIfTI uzantılı dosyadır. Bu çalışmada 131 hastanın etiketli karın içi BT görüntüsü kullanılmış etiketsiz veriler kullanılmamıştır. 131 adet etiketli veriden yaklaşık %15'i test veri seti için (18 adet), yaklaşık %5'i (7 adet) doğrulama veri seti için, yaklaşık %80'i eğitim veri seti için (106 adet) manuel olarak ayrılmıştır.

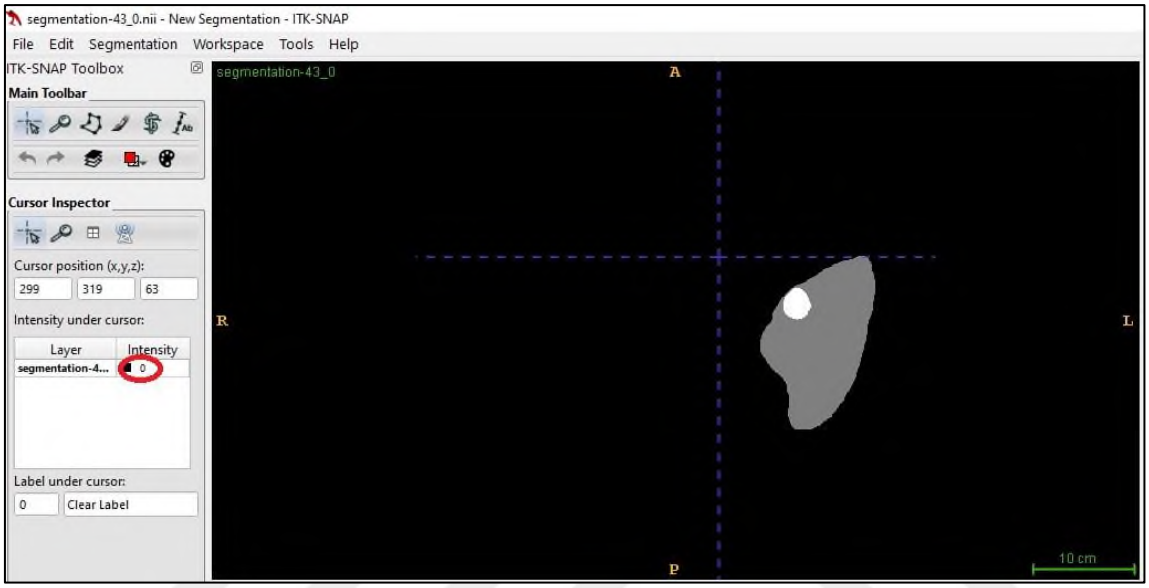
Veri seti içindeki her bir hastanın NIfTI dosyaları incelendiğinde bu dosyaların farklı sayılarda (65-517 arasında) DICOM dosyası içeren DICOM serilerinden oluştuğu anlaşılmıştır. Birinci NIfTI dosyasının karın içi görüntüyü, ikinci NIfTI dosyasının ise bu görüntünün etiketli halini, yani karaciğer, karaciğer tümörü ve karın içi diğer dokuların etiketlendiği maskesini barındırdığı görülmüştür. Her bir hasta için her iki NIfTI dosyası kontrol edildiğinde bu itibarla eşit sayıda DICOM dilimi içerdikleri teyit edilmiştir. Şekil 5.2.'de bilgisayarda açılmış olan LiTS veri setinin dosya yapısı gösterilmiştir.



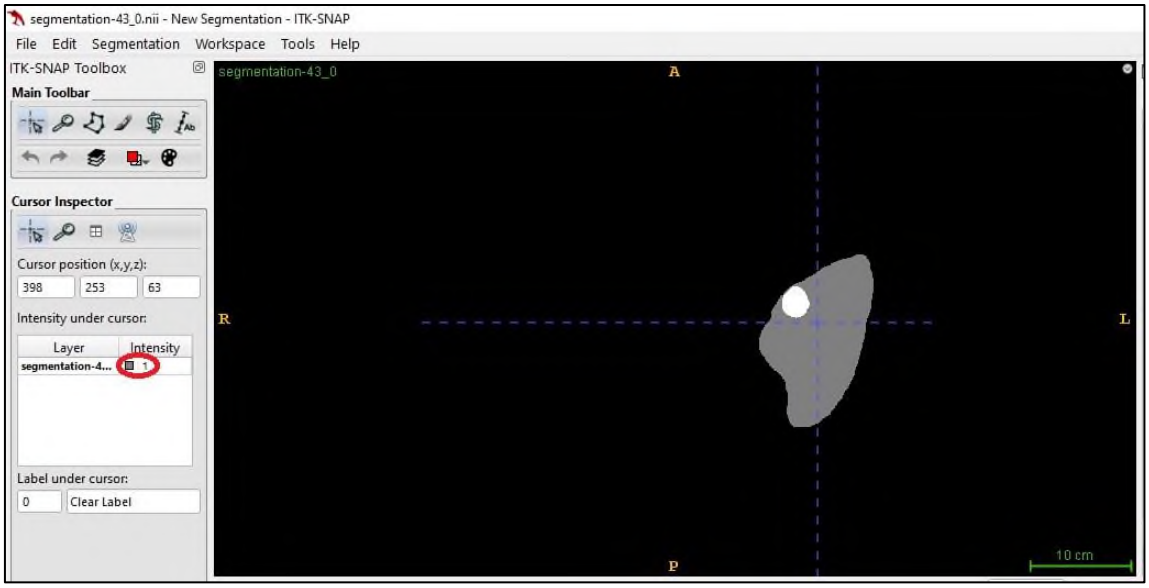
Şekil 5.2. LiTS veri seti dosya yapısı

LiTS veri seti görüntü ve etiketleri görüntü işleme programları kullanılarak görselleştirilmiş, karın içi BT görüntülerinin maskelerinde üç etiketin de mevcut olduğu teyit edilmiştir. Bunlardan arka plan olan etiketin yani karın içi diğer dokuların sıfır (0), karaciğer

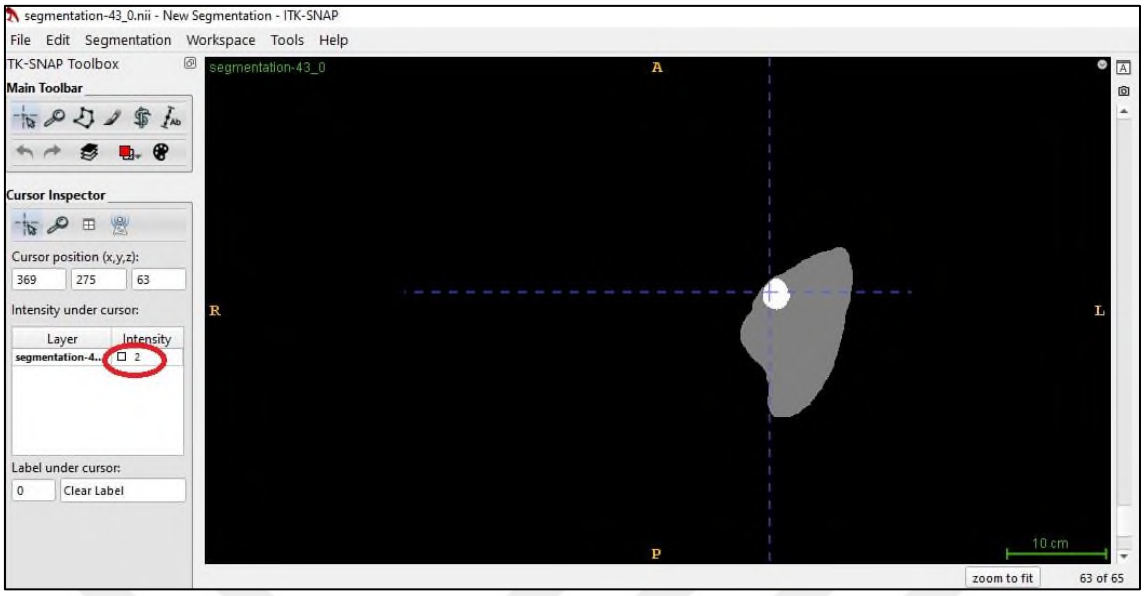
dokularının bir (1), tümör dokularının ise iki (2) olarak etiketlendiği görülmüştür. Bu işlem ITK-SNAP ve 3D SLICER programları ile gerçekleştirilmiştir. Şekil 5.3.'de sıfır (0) etiketli karın içi diğer dokuların, Şekil 5.4.'de bir (1) etiketli karaciğer dokularının, Şekil 5.5.'de iki (2) etiketli tümörlü dokuların ITK-SNAP programından alınmış ve işaretlenmiş görselleri verilmiştir.



Şekil 5.3. Sıfır (0) etiketli karın içi diğer dokular



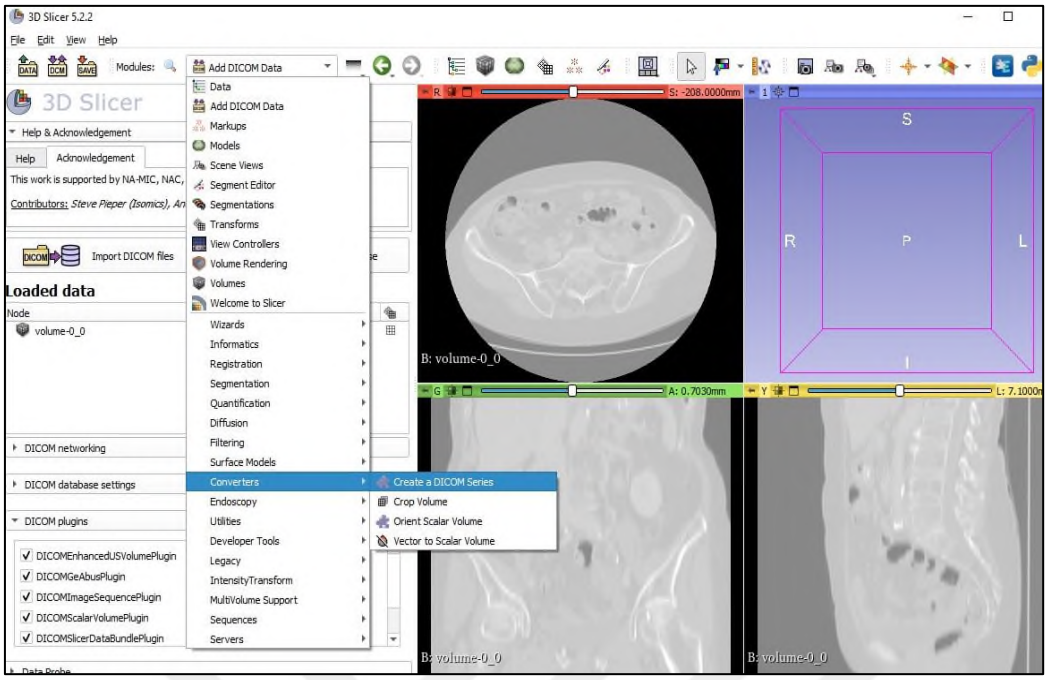
Şekil 5.4. Bir (1) etiketli karaciğer dokuları



Şekil 5.5. İki (2) etiketli tümörlü dokular

5.3. Veri Önişleme

Veri seti içindeki her bir NIfTI dosyası farklı sayıda (65-517 arasında) DICOM dosyası içeren DICOM serilerinden oluştuğundan birinci adımda eşit sayıda DICOM dosyası içeren NIfTI dosyalarının oluşturulması hedeflenmiştir. Bunun için Şekil 5.5'te görüldüğü gibi her bir NIfTI dosyası "3D Slicer" programı ile kendi adı ile anılan bir klasör altına DICOM serilerine açılmıştır. Açılan bu serilerden 'dcomtonifti' python kütüphanesi kullanılarak otomatik olarak eşit sayıda DICOM dosyası içeren NIfTI dosyaları oluşturulmuştur.



Şekil 5.6. NIfTI dosyalarından DICOM serisi oluşturma

Veri yapısı

Tüm bu çalışmalarla beraber eşit sayıda DICOM dosyasından oluşan tıbbi görüntü NIfTI dosyaları ve onlara karşılık gelen eşit sayıda DICOM dosyasından oluşan tıbbi görüntü maskesi NIfTI dosya çiftleri oluşturulmuştur. Bu sayede bölütleme projelerinde kullanılan ve sözlük veri yapısına uygun olan veriler elde edilmiştir. Bu verileri depolamak için bölütleme problemlerinde kullanılması uygun olan klasör yapısı Çizelge 5.2.'deki gibi kurulmuştur.

Çizelge 5.2. Veri seti klasör yapısı

Klasör ismi	İçerik
EgitimVerileri	Egitim veri seti görüntüleri NIfTI dosyaları
EgitimEtiketleri	Egitim veri seti maskeleri NIfTI dosyaları
TestVerileri	Test veri seti görüntüleri NIfTI dosyaları
TestEtiketleri	Test veri seti maskeleri NIfTI dosyaları
DogrulamaVerileri	Doğrulama veri seti görüntüleri NIfTI dosyaları
DogrulamaEtiketleri	Doğrulama veri seti maskeleri NIfTI dosyaları

EğitimVerileri klasörüne eğitim veri setinin tıbbi görüntüleri, EğitimEtiketleri klasörüne eğitim veri setinin maske dosyaları, DogrulamaVerileri klasörüne doğrulama veri setinin tıbbi görüntüleri, DogrulamaEtiketleri klasörüne de doğrulama veri setinin maske dosyaları, TestVerileri klasörüne test veri setinin tıbbi görüntüleri, TestEtiketleri klasörüne de test veri setinin maske dosyaları manuel olarak taşınmıştır.

Sonrasında bu dosya çiftleri bölütleme problemlerinde yaygın olarak kullanılan sözlük (dictionary) veri yapısına getirilerek model içinde kullanılmıştır. Sözlüğün her bir öğesi bir görüntü ve ona karşılık gelen maskeyi içerir hale getirilmiştir. Böylece model içinde her bir tıbbi görüntü ve onun etiketi birleştirilip bir öğe haline getirilmiştir.

Veri dönüşümleri

Kullanacağımız verileri istediğimiz forma getirmek, verileri standartlaştırmak, iyileştirmek ve çoğaltmak için veri dönüşümleri (transforms) kullanılabilir. Bu çalışmada sözlük veri yapısı oluşturulduktan sonra eğitim verilerine, doğrulama verilerine ve test verilerine MONAI kütüphanesinin sağlamış olduğu aşağıdaki veri dönüşüm fonksiyonları uygulanmıştır.

Öncelikle dönüşüm fonksiyonları için birer isim belirlenmiştir. “train_donusum” eğitim verilerine uygulanacak veri dönüşüm fonksiyonunun adı, “valid_donusum” doğrulama verilerine uygulanacak dönüşüm fonksiyonunun, “test_donusum” ise test verilerine uygulanacak dönüşüm fonksiyonunun adıdır. Ardından her bir dönüşüm foksiyonu uygulanacak dönüşümlerle beraber oluşturulmuştur [49].

```
train_donusum, test_donusum, valid_donusum = Compose(
   [
      LoadImaged(keys=["grs", "etk"]),
      AddChanneld(keys=["grs", "etk"]),
      EnsureChannelFirstd(keys=["grs","etk"], strict_check=True, allow_missing_keys=False,
channel_dim=None),
      Resized(keys=["grs", "etk"], spatial_size=[256,256,64]),
      Spacingd(keys=["grs", "etk"], pixdim=pixel_dimension, mode=("bilinear", "nearest"),
      Orientationd(keys=["grs", "etk"], axcodes="LAS"),
```

```
    ScaleIntensityRanged(keys=["grs"], a_min=a_min, a_max=a_max, b_min=b_min, b_max=b_max,
clip=True),
    CropForegroundd(keys=['grs', 'etk'], source_key='grs'),
    LabelFilterd(keys=["etk"], applied_labels=(2)),
    ToTensord(keys=["grs", "etk"]),
  ]
)
```

Dönüşüm fonksiyonu birçok veri dönüşümü içermektedir. Veri dönüşümlerinin sonunda yer alan "d" harfi veri tipi sözlük olan verilere bu dönüşümlerin uygulanabileceğini belirtmektedir. MONAI kütüphanesinde sözlük veri tipine uygun dönüşüm fonksiyonları ile tekli veri yapısına uygun dönüşüm fonksiyonları ayrı ayrı sunulmaktadır.

Dönüşüm fonksiyonlarında yer alan composed ifadesi ile seçilen dönüşümlerin her birinin sıra ile uygulanacağı belirtilmiştir.

LoadImaged(keys=["grs", "etk"]);

Bu dönüşüm verileri modele yükleme amaçlıdır ve bu dönüşüm hem tıbbi görüntüler sütununa hem etiketlenmiş maskeler sütununa uygulanmıştır. Veri yüklenmemiş ise üzerinde dönüşüm uygulanamayacağından bu dönüşüm ilk sırada olmak zorundadır. Sözlük veri yapısında yer alan "grs" ifadesi tıbbi görüntülerin olduğu sütunu, "etk" ifadesi ise bu görüntülerin etiketlenmiş maskelerinin olduğu sütunu ifade etmektedir. LoadImaged dönüşümü hem tıbbi görüntülere hem de maskelerine uygulanmıştır.

AddChanneld(keys=["grs", "etk"]),

Bu dönüşüm veri normalizasyonu için kullanılır ve modelin girişine sunulan verilerin boyutlarının tutarlı bir yapıya sahip olmasını sağlar. 3B veya 2B görüntü verisine bir kanal boyutu ekler. MONAI veya PyTorch tabanlı modeller, genellikle giriş görüntülerinde bir kanal boyutu gerektirir. Bu işlem, görüntüyü C, H, W (kanal, yükseklik, genişlik) veya C, H, W, D (kanal, yükseklik, genişlik, derinlik) formatına dönüştürür. Kanal boyutu varsayılan olarak 1'dir. Örneğin giriş görüntüsü (256, 256, 64) olsun çıkış görüntüsünün boyutları (1,256, 256, 64) olacaktır. AddChanneld dönüşümü tıbbi görüntü ve maskesinin aynı boyutta olması gerektiğinden hem tıbbi görüntülere hem de maskelerine uygulanmıştır.

EnsureChannelFirstd(keys=["grs","etk"], strict_check=True, allow_missing_keys=False, channel_dim=None),

Bu dönüşümde yine 3B veya 2B görüntü verisine bir kanal boyutu eklemek için kullanılır. Bu işlem, görüntüyü C, H, W (kanal, yükseklik, genişlik) veya C, H, W, D (kanal, yükseklik, genişlik, derinlik) formatına dönüştürür. MONAI'nin yeni versiyonlarında artık "AddChanneld" dönüşümü yerine "EnsureChannelFirstd" dönüşümü sunulmaktadır. Bu projede de önce "AddChanneld" dönüşümü kullanılmış olmakla beraber sonradan bunun yerine "EnsureChannelFirstd" dönüşümü kullanılmıştır.

Resized(keys=["grs", "etk"], spatial_size=[256,256,64]),

MONAI'deki resized dönüşümü, tıbbi görüntüleri ve ilgili verileri (örneğin, segmentasyon maskeleri) belirli bir boyuta yeniden ölçeklemek (resize) için kullanılır. Bu dönüşüm, farklı boyutlardaki verileri model eğitimi veya analizi için standart bir boyuta getirmek amacıyla uygulanır. Hedef boyut (Spatial size): Görüntünün yeniden ölçekleneceği uzamsal boyutları belirler. Tıbbi görüntü ve maskenin aynı boyutta olması gerektiğinden bu dönüşüm hem görüntü hem de maskeye uygulanır.

Spacingd(keys=["grs", "etk"], pixdim=pixdim, mode=("bilinear", "nearest");

Bu dönüşüm medikal görüntülerin uzamsal çözünürlüklerini yeniden örneklemek için kullanılan bir işlemdir. Görüntülerin piksel boyutlarını (veya voksel boyutlarını) belirli bir hedef boyuta uyacak şekilde ayarlar. Özellikle farklı cihazlardan gelen medikal görüntülerin standartlaştırılması veya belirli bir çözünürlükte analiz yapılması gerektiğinde sıkça kullanılır. Pixdim parametresi hedeflenen piksel/voksel boyutları belirtmeye yarar ve içindeki boyutlar sırası ile yükseklik, genişlik ve derinliktir. Tek bir sayı belirtilmiş ise, ilk boyut için bu sayı kullanılacaktır. Pixdim dizisinin öğeleri giriş görüntüsünün uzamsal boyutlarıyla eşleşir; eğer pixdim dizisinin uzunluğu görüntünün uzamsal boyutlarından daha uzunsa, daha uzun kısım göz ardı edilir, daha kısaysa 1,0 ile doldurulur. Spacingd dönüşümü, orijinal görüntü çözünürlüğü ile hedef çözünürlük arasındaki farkı kullanarak interpolasyon yapar. Mode parametresi çıkış değerlerini hesaplamak için kullanılacak interpolasyon modunu belirtir. "Linear (Doğrusal)" sürekli değerler için uygundur (örneğin, MR

görüntüleri). “Nearest (En yakın)” Kategorik verilerde (örneğin, bölütleme maskeleri) tercih edilir. Spacingd dönüşümü hem tıbbi görüntülere hem de maskelerine uygulanmıştır.

Orientationd(keys=["grs", "etk"], axcodes="LAS");

MONAI'deki Orientationd dönüşümü, medikal görüntülerin uzamsal yönlerini standart bir formata dönüştürmek için kullanılan veri ön işleme aracıdır. Medikal görüntüler, tarayıcı cihazlarının farklı çıktıları nedeniyle farklı yönelimlerde olabilir (örneğin, aksiyal, koronal veya sagital düzlemler). Bu tutarsızlıklar, verileri işlemek ve modellemek için sorunlara neden olabilir. Orientationd dönüşümü, verilerin yönelimini standart bir referans sistemine (örneğin, RAS - Right-Anterior-Superior) dönüştürerek bu sorunu çözer. Bu işlem sırasında NIfTI görüntü dosyalarının başlık bilgileri kullanılır ve veriler belirtilen oryantasyona uygun şekilde döndürülür veya yeniden düzenlenir. LAS (Left-Anterior-Superior) anatomik referans sisteminde, bir görüntünün sol (left), ön (anterior) ve üst (superior) taraflarını tanımlayan bir kodlamadır. LAS sistemi, görüntüyü bu referans noktaları doğrultusunda hizalar. Sol (L): Görüntüleme cihazına göre hastanın sol taraf, Ön (A): Hastanın ön kısmı (yüz ve göğüs tarafı), Üst (S): Baş ya da anatomik olarak daha yukarıda bulunan taraf.

ScaleIntensityRanged(keys=["grs"];

MONAI'deki ScaleIntensityRanged dönüşümü, tıbbi görüntülerin yoğunluk değerlerini belirli bir aralıktan normalize edilmiş bir hedef aralığa ölçeklendirmek için kullanılır. Bu işlem, medikal görüntülerin farklı cihazlardan veya protokollerden kaynaklanan yoğunluk farklarını gidermeye ve modeli eğitmek için uygun bir giriş formatı oluşturmaya yardımcı olur. Maskede yoğunluk bilgisi olmadığından bu dönüşüm sadece tıbbi görüntüye uygulanmıştır.

Giriş Aralığı (a_degermin, a_degermax): Yoğunluk değerlerinin seçileceği alt ve üst sınır. Bu sınırların dışındaki değerler kesilir.

Çıkış Aralığı (b_degermin, b_degermax): Yoğunluk değerlerinin ölçeklendirileceği hedef aralık.

clip: True ise, giriş aralığının dışındaki değerler sınırlandırılır (kesilir). Varsayılan değer olarak clip:True tanımlanmıştır..

CropForegroundd(keys=['grs', 'etk'], source_key='grs'),

MONAI'nin CropForegroundd dönüşümü, bir medikal görüntünün anlamlı bölgesini (foreground) otomatik olarak kesmek ve bu bölgeye odaklanmak için kullanılır. Bu işlem, genellikle görüntü verisinin gereksiz arka plan alanlarını kaldırarak daha verimli bir veri işleme ve analiz süreci sağlamak için uygulanır. Foreground bölgesinin belirlenmesi için kullanılan referans veri anahtarı (genellikle görüntü) kesme işlemi hem tıbbi görüntüye hem de maskeye uygulanacağından bu dönüşüm sözlüğün iki sütununa da uygulanır.

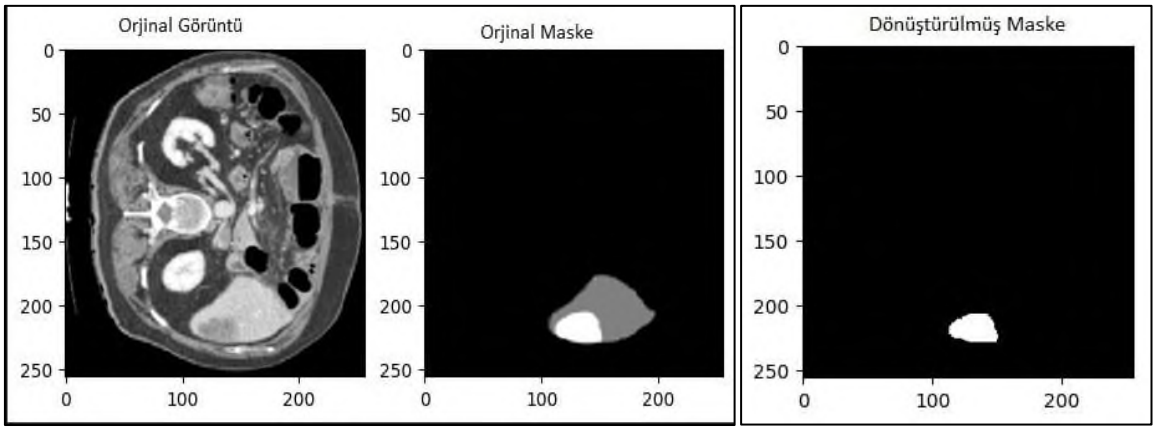
LabelFilterd(keys=["etk"], applied_labels=(2));

Bu dönüşüm etiketlenmiş maskelerdeki bazı etiketleri kullanmak amaçlıdır ve sadece etiketlenmiş maskeler sütununa uygulanmıştır. applied_labels=(2) ifadesi diğer tüm etiketler arka plan yani sıfır (0) etiketi olarak değiştirilip sadece sıfır (0) ve iki (2) etiketli piksellerden oluşan yeni bir maske oluşturur.

ToTensord(keys=["grs", "etk"]);

Bu dönüşüm verileri tensör veri tipine dönüştürme amaçlıdır ve hem tıbbi görüntüler sütununa hem etiketlenmiş maskeler sütununa uygulanmıştır. Veri diğer tüm dönüşümler uygulandıktan sonra tensör tipine dönüştürüleceğinden bu dönüşüm son sırada uygulanır.

Tüm bu dönüşümlerin ve daha fazla dönüşümün ayrıntılı anlatımları ve örnek kullanımları https://docs.monai.io/en/stable/transforms.html#dictionary-transforms bağlantısı altında mevcuttur. Dönüşüm fonksiyonu uygulandıktan sonra verileri görselleştirdiğimizde Şekil 5.7.'deki görseller elde edilmiştir. Şekilde görüldüğü üzere orjinal maske üzerindeki karaciğere ait pikseller arka plana eklenmiş, yeni maske sadece arka plan ve tümör etiketi içeren piksellerden meydana gelmiştir.



Şekil 5.7. Dönüşüm fonksiyonu uygulanmış veriler

5.4. Derin Öğrenme Modeli ve Parametreler

Yapılan çalışmada MONAI yapay zekâ kütüphanesi ve bölütleme problemlerinin çözümünde en çok kullanılan ESA modellerinden biri olan UNet ağ modeli kullanılmıştır. Bölütleme problemlerinde başarılı sonuçlar ürettiği için öncelikle UNet ağ modeli tercih edilmiştir. Çizelge 5.3. kullanılan UNet ağ modeli parametrelerini ayrıntılı olarak göstermektedir.

Çizelge 5.3.Unet Ağ Modeli Parametreleri

UNet
class monai.networks.nets.UNet(
spatial_dims,
in_channels,
out_channels,
channels,
strides,
kernel_size=3,
up_kernel_size=3,
num_res_units=0,
act='PRELU',
norm='INSTANCE',
dropout=0.0,
bias=True,
adn_ordering='NDA')

spatial_dims – Modelin çalışacağı uzamsal boyutları belirler. 2 ise 2B görüntüler, 3 ise 3B görüntüler ile çalışır.

in_channels – Giriş görüntüsünün kanal sayısıdır. Değer 1 ise giriş siyah-beyaz (tek kanallı) tıbbi görüntülerdir. Değer 3 ise giriş renkli (RGB) görüntülerdir.

out_channels – Modelin üreteceği çıktı kanal sayısını ifade eder. Genellikle çıktı kanal sayısı sınıf sayısını temsil eder. Eğer model regresyon (sürekli değer tahmini) yapıyor ise 1, model ikili sınıflandırma/bölütleme (örneğin, "arka plan" ve "foreground") yapıyor ise 2, eğer model çok sınıflı bölütleme (örneğin, N farklı sınıf) yapıyor ise sınıf sayısı (N) değerini alır.

channels – UNet'in her bir katmanında kullanılacak kanal (filtre) sayısını belirtir. Tuple veri tipinde verilir: (kanal1, kanal2, ..., kanalN) ve channels uzunluğu en az 2 olmalıdır.
channels=(16, 32, 64, 128, 256) olarak verilmiş ise ilk kodlayıcı katmanında 16 kanal, sonraki kodlayıcı katmanlarında sırasıyla 32, 64, 128 ve 256 kanal vardır.

strides – Evrişim katmanında kodlayıcı aşamasında kullanılan aşağı örnekleme (downsampling) işlemleri sırasında uygulanacak adım boyutlarıdır. Tuple veri tipinde verilir: (stride1, stride2, ..., strideN).

Her bir değer, bir önceki katmandan bir sonraki katmana geçişte uzamsal boyutların nasıl küçüleceğini belirler. Örneğin, stride=2 ile bir boyut yarıya indirilir. Strides uzunluğu, channels uzunluğundan 1 eksik olmalıdır.

kernel_size – Evrişim çekirdeği boyutunu belirler. Evrişim çekirdeği, genellikle sabit boyutlu bir kare veya dikdörtgen şeklinde olup, giriş verisini tarar ve her bir konvolüsyon işlemi için yeni bir özellik haritası oluşturur. Varsayılan değeri 3 olup herhangi bir değer verilmemiş ise kernel_size 3 olarak kabul edilir.

up_kernel_size – Bu parametre, yukarı örnekleme işlemi sırasında kullanılan evrişim çekirdeğinin boyutunu tanımlar. Yukarı örnekleme, modelin çıktı boyutlarını (yükseklik, genişlik, derinlik) artıran bir işlemdir. Varsayılan değeri 3 olup herhangi bir değer verilmemiş ise up_kernel_size 3 olarak kabul edilir.

num_res_units – Kullanılan artık blokların sayısını verir. Artık blok, derin öğrenmede kullanılan bir yapı taşıdır ve özellikle çok derin sinir ağlarının eğitimi sırasında karşılaşılan "kaybolan gradyan" sorununu çözmek için tasarlanmıştır. Bu yapı, bir sinir ağına, giriş verilerini doğrudan çıkışa aktaran bir shortcut (kestirme) ekleyerek, ağın daha etkili bir şekilde öğrenmesini sağlar.

Artık bloklar, ilk kez ResNet (Residual Network) mimarisinde tanıtılmıştır ve günümüzde birçok derin öğrenme mimarisinde önemli bir yer tutmaktadır. Artık bloklar, derin öğrenme ağlarında bilgi akışını kolaylaştırarak öğrenmeyi iyileştirir. Varsayılan değer 0'dır.

act – derin öğrenme modelindeki aktivasyon fonksiyonunu belirtmek için kullanılır. Aktivasyon fonksiyonu, bir yapay sinir ağının katmanlarındaki doğrusal olmayan işlemleri kontrol eder ve modelin daha karmaşık verileri öğrenmesini sağlar. MONAI kütüphanesinde de bu parametre, modelin her bir katmanında kullanılacak aktivasyon fonksiyonunu belirtmek için kullanılır. Varsayılan: PReLU' dur.

norm – Kullanılan norm parametresi, modelin katmanlarında hangi normalizasyon tekniğinin kullanılacağını belirtmeye yarar. Varsayılan değer Norm.INSTANCE'dır. Herhangi bir parametre belirtilmediğinde bu normalizasyon kullanılır.

dropout – seyreltme oranını ifade eder. 0,1 değeri bağlantıların yüzde on, 0,2 değeri bağlantıların yüzde yirmi oranında seyreltildiği anlamına gelir.

bias – Evrişim bloklarında bias terimi olup olmadığını belirtir. Varsayılan olarak "True"dur.

```
model = UNet(
    spatial_dims=3,
    in_channels=1,
    out_channels=2,
    channels=(16, 32, 64, 128, 256),
    strides=(2, 2, 2, 2),
    num_res_units=2,
    norm=Norm.BATCH,
    dropout=0.1,)
```

Yukarıdaki kod parçacığı bu çalışmada kullanılan ve parametreleri detaylı biçimde anlatılan UNet ağ modelinin nasıl uygulandığını göstermektedir. Kullandığımız veri seti üç boyutlu olduğundan "spatial_dims=3" eklenmiş, görüntülerimiz renkli olmadığından "in_channels=1" eklenmiş, çıkış değerlerimiz sadece binary yani 0 ve 1 olduğundan "out_channels=2" eklenmiş, "channels=(16, 32, 64, 128, 256)" ifadesi ile her kodlayıcı katmanındaki kanal sayısı belirtilmiş, aşağı örnekleme işleminde adım boyutunu belirtmek için "strides=(2, 2, 2, 2)" eklenmiş, kullanılan residual blok sayısını belirtmek için "num_res_units=2" eklenmiş, kullanılan normalizasyon türünü belirtmek için "norm=Norm.BATCH" eklenmiştir.

Daha önce de bahsedildiği gibi veri seti içindeki (512×512) boyutundaki görüntüler kaynak ihtiyacını azaltmak ve eğitim süresini kısaltmak için (256×256) olarak ölçeklenmiş ve ağ modeli bu şekilde çalıştırılmıştır.

Hiperparametreler (Hyperparameters)

Derin sinir ağ modelleri iki farklı parametre türü barındırırlar: Bunlardan ilki "model parametreleridir" ve bu parametreler kısaca "parametreler" olarak ta anılmaktadırlar. Model parametreleri baştan belirlenmezler, model eğitilirken modelin çıktıları bir kayıp fonksiyonu ile optimize edilip adım adım öğrenilirler. Örneğin YSA'daki her bir ağırlık baştan belirlenmez modelin çalışması esnasında öğrenilir. Girdilerin istenen çıktılara nasıl dönüştürüleceği model parametreleri sayesinde belirlenir.

İkinci parametre türü ise "hiperparametreler" olarak adlandırılırlar. Bu parametreler model parametreleri gibi girdi ve çıktı değerlerinden eğitilemezler. Daha çok modelin kurgulanmasını tanımlarlar ve model eğitilmeye başlanmadan önce kullanıcı tarafından kişisel tecrübe, önceki denemeler ve genel kabul görmüş öngörüler ile belirlenirler. Örneğin bir derin ağ modelinde adım sayısı (epok) değeri, öğrenme hızı (learning rate), seyreltme (dropout) yapılıp yapılmayacağına yapılacaksa seyreltme oranına, modelin derinliğine vb. modelin kullanıcısı baştan karar verir. Bu karar verilirken çözüm bekleyen problemin durumu, kullanılan veri seti vb. bileşenler tasarımcı tarafından göz önüne alınır. Yapılan denemeler sonucunda modelin gösterdiği en iyi performansa göre bu parametrelerden hangilerinin kullanılacağına karar verilir. Bunun yanında eldeki problem için optimum parametre grubu seçiminin yapılabilmesi için bazı araçlar da mevcuttur.

Öğrenme hızı

Öğrenme hızı, bir derin öğrenme modelinin her adımda ağırlıklarını ne kadar değiştireceğini belirleyen bir hiperparametredir. Yüksek bir öğrenme hızı, modelin hızlıca öğrenmesini sağlar, ancak çok yüksek olduğunda modelin dengeli bir şekilde öğrenememesine ve kararsız sonuçlar vermesine yol açabilir. Düşük bir öğrenme hızı ise daha hassas bir güncelleme yapar, ancak öğrenme sürecini yavaşlatabilir. Dolayısıyla, doğru öğrenme hızının seçilmesi, modelin doğruluğu ve eğitim süresi açısından kritik bir öneme sahiptir. Öğrenme hızına statik olarak değer verilebildiği gibi, model için uygun öğrenme hızı sürekli olarak artan bir değer olarak da verilebilir. Bu artış doğrusal olabildiği gibi üstel olarak ta belirlenebilir. MONAI kütüphanesinde bu görevin optimum sonuçla yerine getirilmesi için sunulan araç "Learning Rate Finder" fonksiyonudur.

Çalışmada kullanılan UNet Modeli farklı öğrenme hızlarında çalıştırılmış (lr=1e-4=0,0001) ve lr=1e-5=0.00001) ve sonuçlar karşılaştırılmıştır. lr =1e-5 ifadesi Unet Modelinin eğitimi esnasında modele girilen bir parametredir ve ifadedeki "e" "exponent" ("üs") anlamındadır. Bu durumda öğrenme hızı 10^{-5} olarak hesaplanabilir.

Optimizasyon fonksiyonu

Optimizasyon fonksiyonları, derin öğrenme modellerinde kayıp fonksiyonunu minimize etmek için kullanılan algoritmalardır. Bu fonksiyonlar, model parametrelerini (örneğin, ağırlıklar ve biaslar) her iterasyonda güncelleyerek modelin öğrenmesini sağlar. Amaç, kayıp fonksiyonunun çıktısını mümkün olan en düşük değere ulaştırmaktır. Derin ağların karmaşıklığı nedeniyle optimizasyon fonksiyonları, bu süreçte verimli ve etkili bir şekilde çalışmalıdır.

Çalışmada kullanılan UNet ağ modeli farklı optimizasyon fonksiyonları (Adam ve SGD) ile çalıştırılmış en başarılı sonuç Adam optimizasyon fonksiyonunun uygulandığı model ile elde edilmiştir.

Kayıp fonksiyonu

Kayıp fonksiyonu (loss function), bir derin öğrenme modelinin tahminleri ile gerçek değerler arasındaki farkı ölçen matematiksel bir fonksiyondur. Amaç, bu farkı en aza indirmek için model parametrelerini optimize etmektir. Kayıp fonksiyonu, modelin başarımını doğrudan etkiler ve öğrenme sürecinde kritik bir rol oynar. Kayıp fonksiyonu, modelin öğrenme sürecinin merkezinde yer alır. Modelin doğru performans göstermesi için problemin yapısına uygun bir kayıp fonksiyonu seçmek kritik öneme sahiptir. Bu sebeple probleme uygun kayıp fonksiyonu seçilmelidir. Örneğin ikili sınıflandırma problemleri için Binary Cross-Entropy kayıp fonksiyonunu seçmek uygun olurken, çok sınıflı sınıflandırma problemleri için Categorical Cross-Entropy kayıp fonksiyonu uygun olmaktadır.

Problemimiz bir bölütleme problemi olduğundan ve kullandığımız tıbbi görüntüler göz önüne alındığında aranan tümörlü alan görüntü içinde küçük bir alan kapladığından bu durumlar için daha uygun olan dice kayıp fonksiyonu kullanılmasına karar verilmiştir.

Adım sayısı (Epok)

Çalışmada kullanılan UNet modeli 200 epok çalıştırılmış ve en iyi model sonucuna 104. epokta erişmiştir. 104. epok geçildikten sonra modelin öğrenme performansının düştüğü hatta bozulduğu görülmüştür.

Eşik değer (Threshold)

Eşik değer, bölütleme problemlerinde bir pikselin veya birimin bir sınıfa ait olup olmadığına karar vermek için kullanılan bir sınır değeridir. Bölütleme problemleri, özellikle görüntü verilerinde bir objeyi arka plandan veya farklı objeleri birbirinden ayırmayı hedeflediğinden eşikleme, bu ayrımı gerçekleştirmek için en temel yöntemlerden biridir ve çoğu bölütleme algoritmasında temel bir rol oynar.

Modelin çıktı haritası, her bir piksel için [0, 1] aralığında bir olasılık tahmini içerir. Eşik değer uygulanarak, çıktılar aşağıdaki gibi binarize edilir.

$$T(x,y) = \begin{cases} 255 & \text{if } I(x,y) > T \\ 0 & \text{if } I(x,y) \leq T \end{cases} \quad (5.1)$$

T eşik değerini;

(x, y) ilgili pikselin görsel içindeki koordinatını temsil eder.

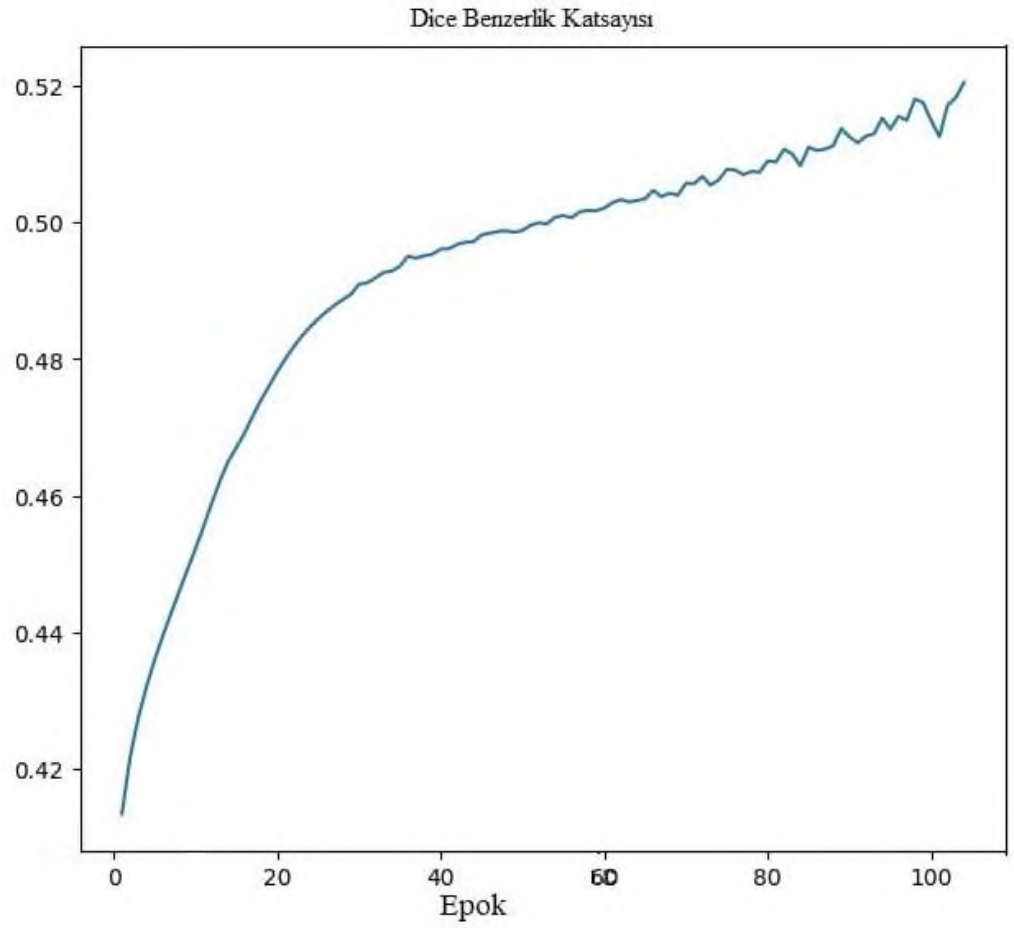
Eşik değer, hangi piksellerin nesneye, hangilerinin arka plana ait olduğunu belirler. Düşük bir eşik değer daha fazla pikseli nesne olarak sınıflandırabilir, yüksek bir eşik değer daha seçici davranarak yanlış pozitifleri azaltır, bölütleme kalitesini iyileştirir. Eşik değer problemden bağımsız olarak genellikle hiperparametre optimizasyonunda ayarlanır.

6. DENEYSEL SONUÇLAR

6.1. Öğrenme Hızı = 0,00001

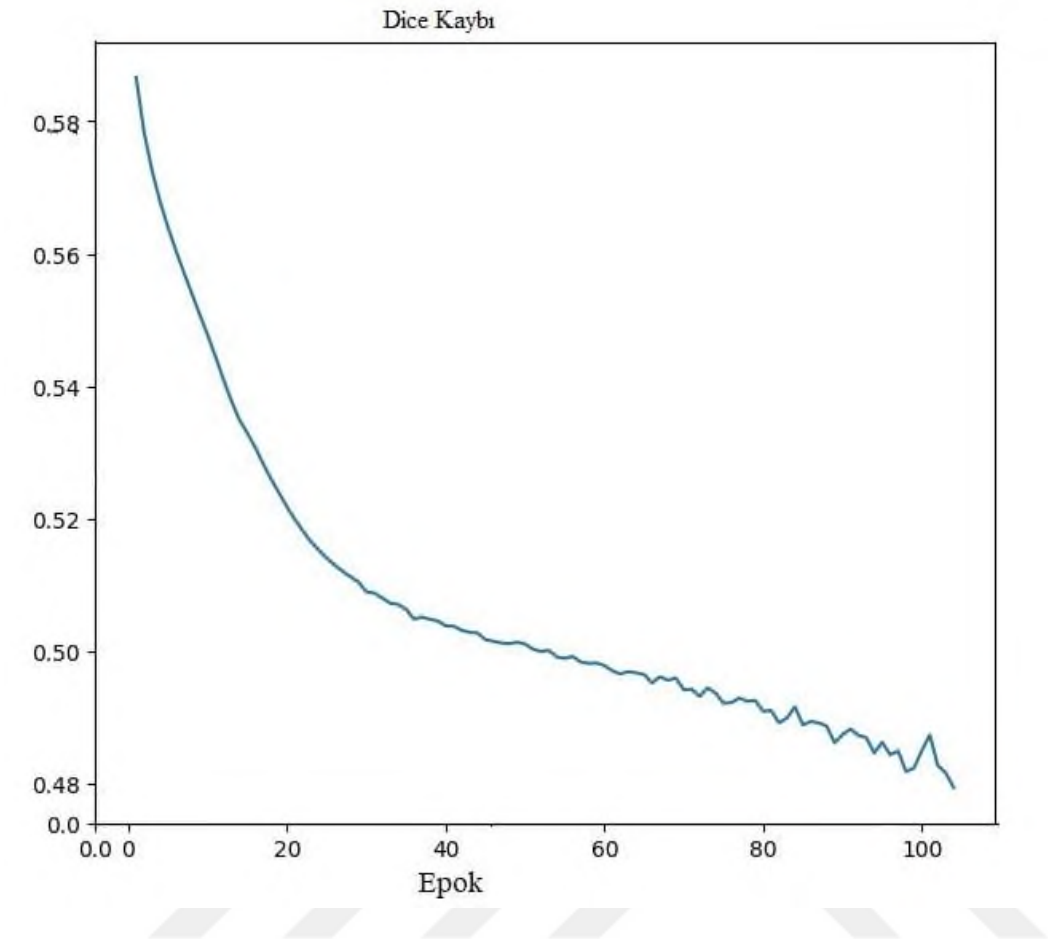
Bu alt bölümde öğrenme hızı Lr = 0,00001 alındığında elde edilen sonuçlar verilmiştir.

Şekil 6.1.'de Lr=0,00001 için doğrulama verileri ile elde edilen Dice Benzerlik Katsayısı-Epok Değişim Grafiği verilmiştir. Grafikte de görüldüğü gibi doğrulama verileri ile elde edilen en iyi Dice Benzerlik Katsayısı: 0,5205 olarak sağlanmıştır.



Şekil 6.1. Lr=0,00001 için Dice Benzerlik Katsayısı-Epok Değişim Grafiği

Şekil 6.2.'de doğrulama veri seti ile elde edilen Dice Kaybı-Epok Değişim Grafiği verilmiştir. Grafikte de görüldüğü gibi doğrulama verileri ile elde edilen en iyi Dice Kayıp Değeri: 0,4795 olarak sağlanmıştır.



Şekil 6.2. Lr=0,00001 için Dice Kaybı-Epok Değişim Grafiği

Çalıştırılan modelin çıktıları test veri seti içindeki tıbbi görüntüler ile test edildiğinde Şekil 6.3'deki sonuçlar (tahminler) elde edilmiştir. Şekil 6.3 incelendiğinde;

Birinci sütun test veri seti içindeki dönüşüm fonksiyonu uygulanmış tıbbi görüntüleri yani test için kullanılan görüntüleri ifade etmektedir.

İkinci sütun test veri seti içindeki tıbbi görüntülerin dönüşüm fonksiyonu uygulanmış maskelerini yani manuel olarak etiketlenmiş ve dönüşüm fonksiyonları uygulanmış doğru kabul edilen maskeleri ifade etmektedir.

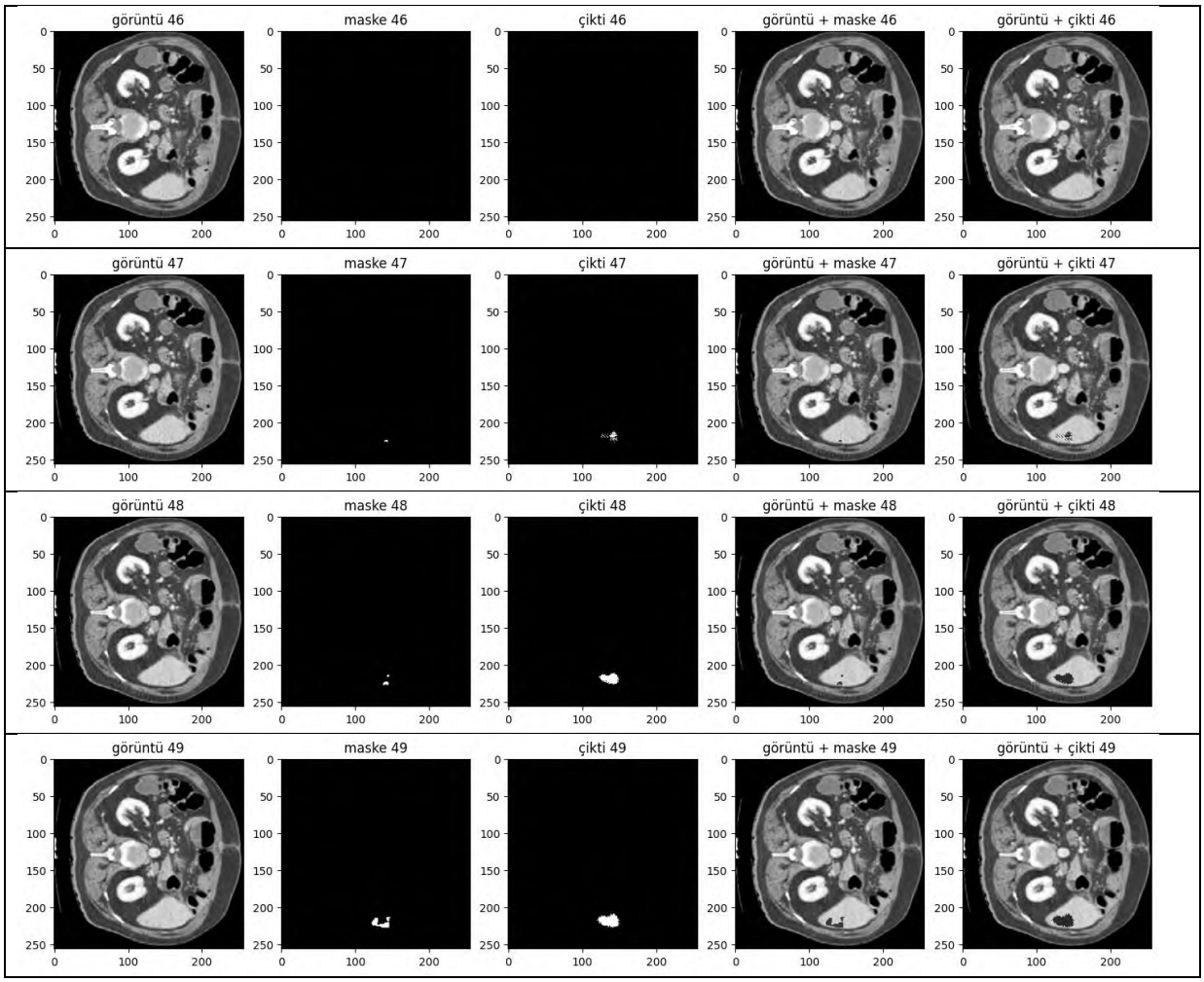
Üçüncü sütun dönüşüm fonksiyonları uygulanmış test veri seti üzerinde modelin tahmin ettiği sonuçları (çıktı) ifade etmektedir.

Dördüncü sütun birinci ve ikinci sütunların üst üste bindirilmiş görüntüsünü (blended image), yani test veri seti içindeki veri dönüşümleri uygulanmış tıbbi görüntü ile o görüntüye

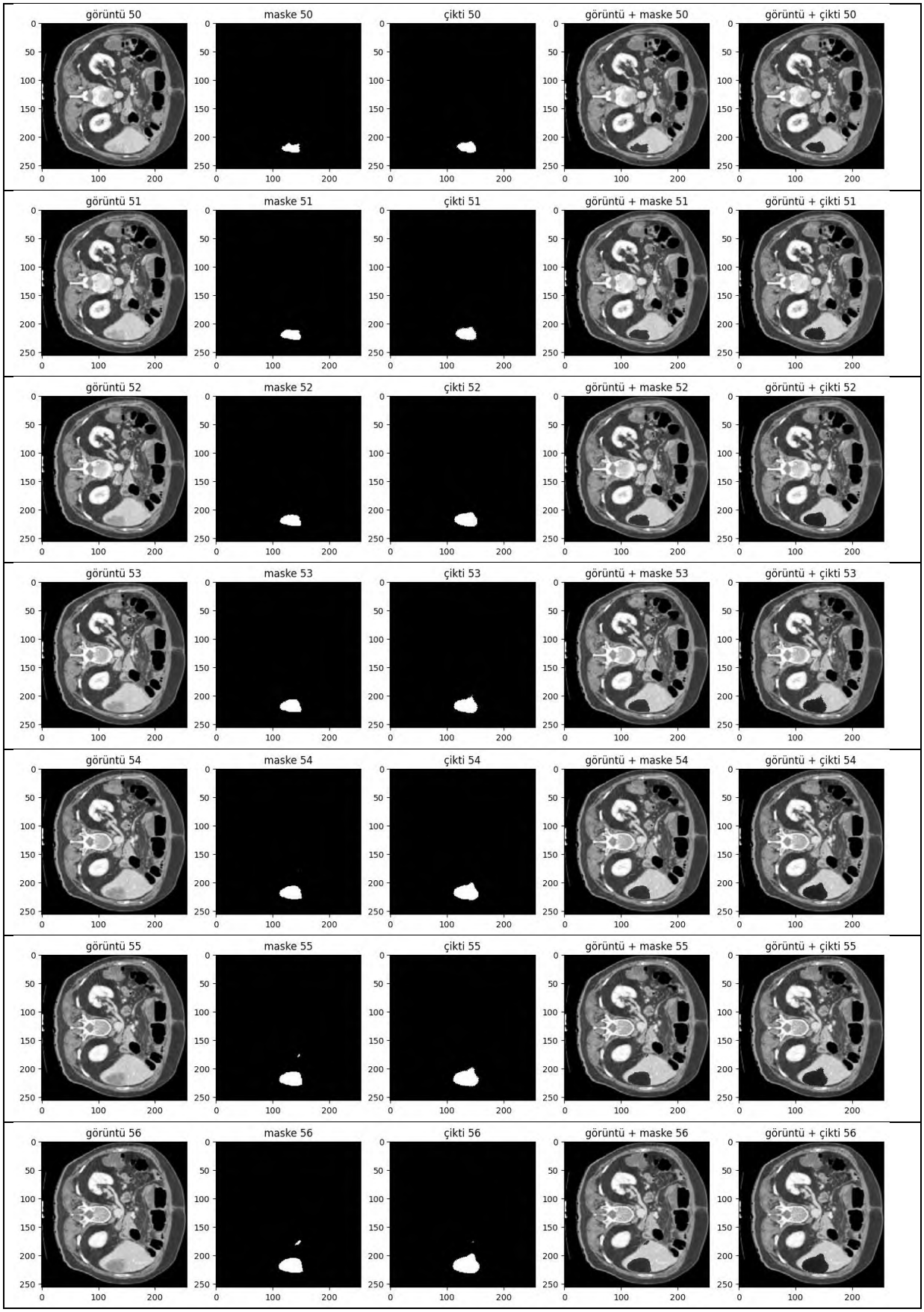
ait maskenin üst üste bindirilmiş görselini ifade etmektedir. Görüntüleri üst üste bindirme işlemi için yine MONAI kütüphanesinden faydalanılmış ve “blend_images” fonksiyonu kullanılmıştır.

Beşinci sütun birinci ve üçüncü sütunların üst üste bindirilmiş görüntüsünü (blended image), yani test veri seti içindeki dönüşüm fonksiyonları uygulanmış tıbbi görüntü ile modelin tahmin ettiği maskenin (çıktı) üst üste bindirilmiş görselini ifade etmektedir. Yine bu işlem için de MONAI kütüphanesinden “blend_images” fonksiyonu kullanılmıştır.

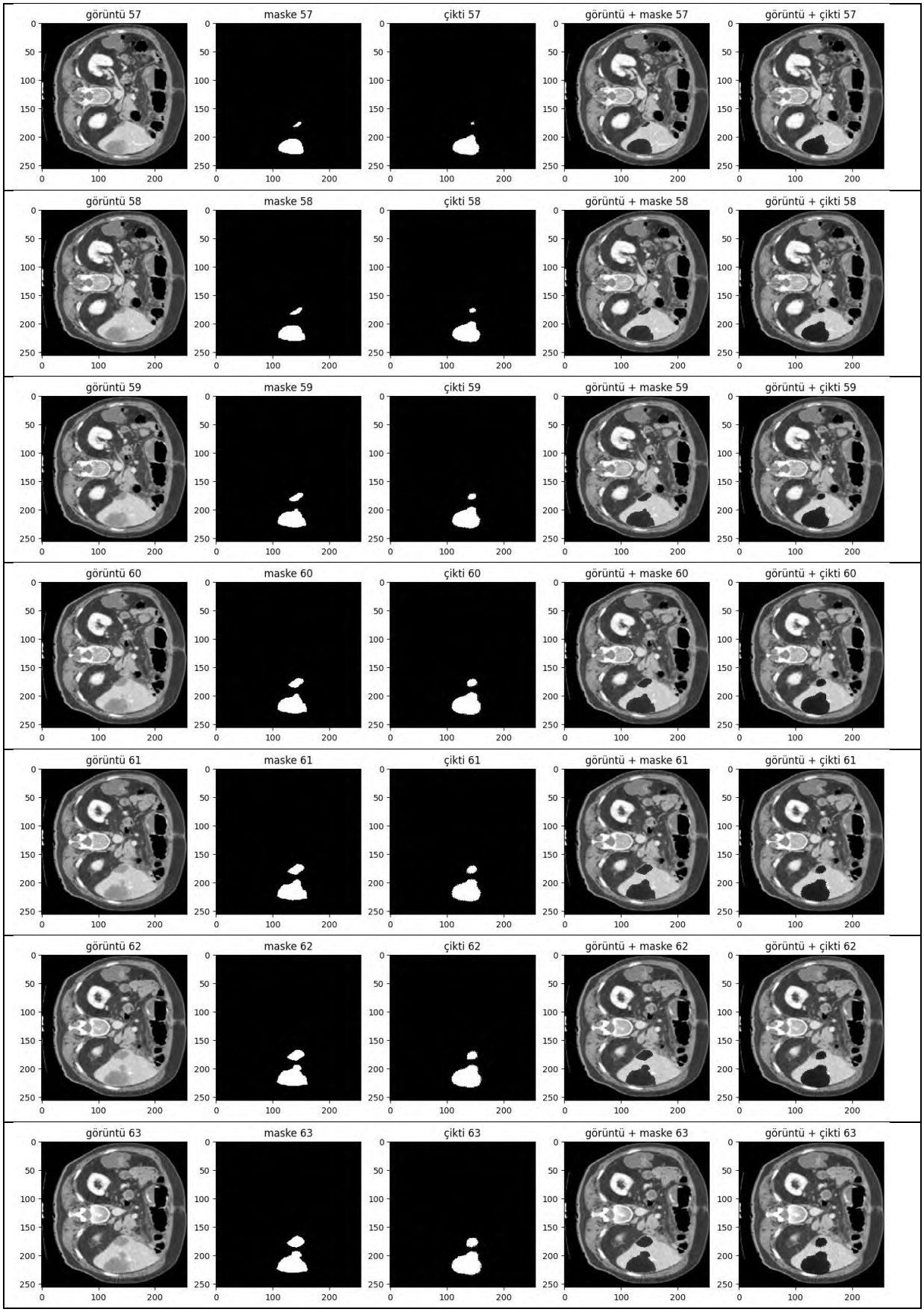
Şekil 6.3. Test verileri ile elde edilen çıktılar



Şekil 6.3. (devam) Test verileri ile elde edilen çıktılar



Şekil 6.3. (devam) Test verileri ile elde edilen çıktılar

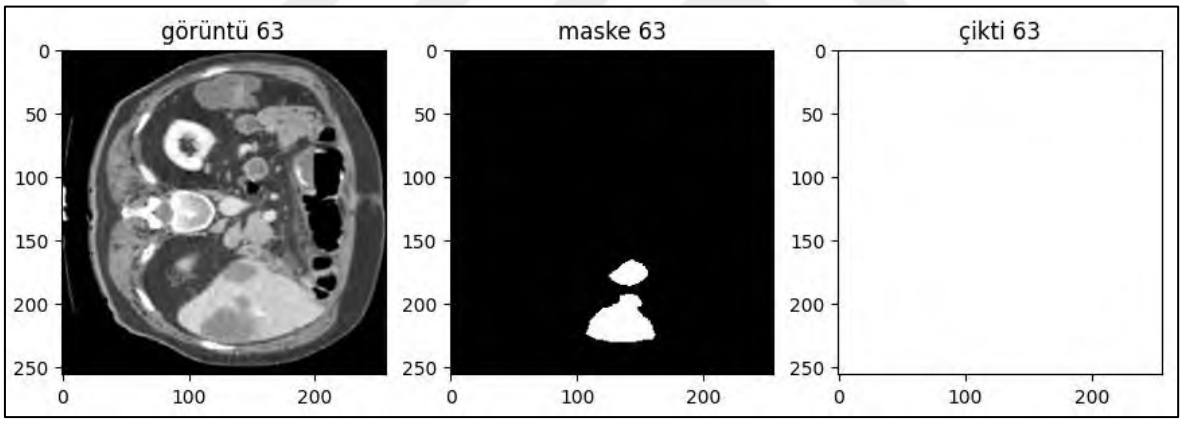


ROC Analizi

Test verileri ile elde edilen görsellerden her bir pikselin değeri modelin çıkışına farklı eşik değerler (0,0-0,1-0,2-0,3-0,4-0,4-0,5-0,6-0,7-0,8-0,9-1,0) uygulanarak ölçülmüş ve her bir eşik değer için modelin çıktıları alınmış, karmaşıklık matrisleri ayrı ayrı oluşturulup FPR (False Positive Rate – Yanlış Pozitif Oranı), TPR (True Positive Rate – Doğru Pozitif Oranı) değerleri hesaplanmış ve ROC eğrisi çizilmiştir.

Eşik Değer=0,0 iken Karmaşıklık Matrisi, FPR ve TPR Değerleri;

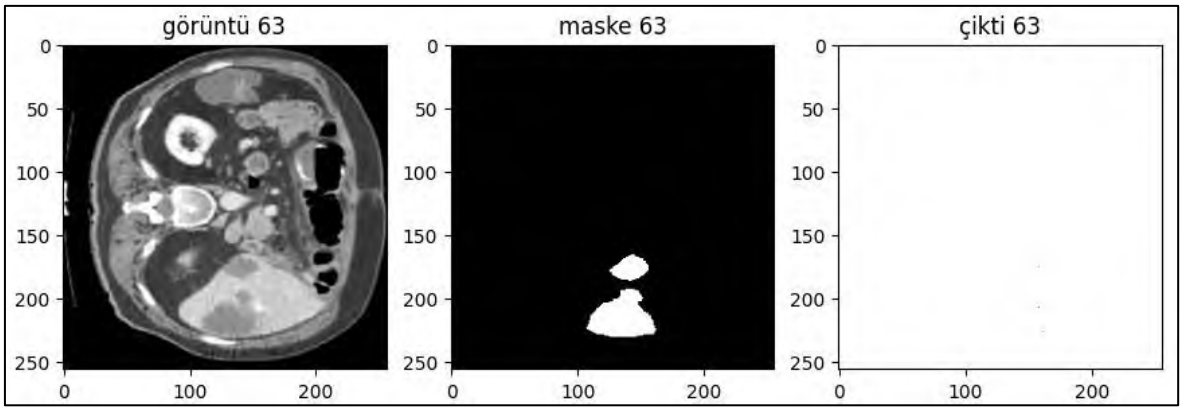
DP=1781, YP=63755
YN=0, DN=0
FPR=1
TPR=1



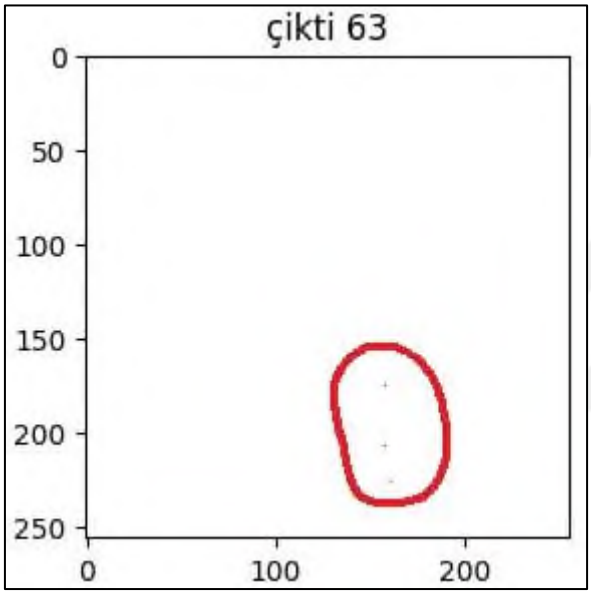
Şekil 6.4. Eşik değer 0,0 için model çıktısı

Eşik Değer=0,1 iken Karmaşıklık Matrisi, FPR ve TPR Değerleri;

DP=1781, YP=63752
YN=0, DN=3
FPR=0,9999
TPR=1



Şekil 6.5. Eşik değer 0,1 için model çıktısı



Şekil 6.6. Eşik değer 0.1 için model çıktısı DN pikseller

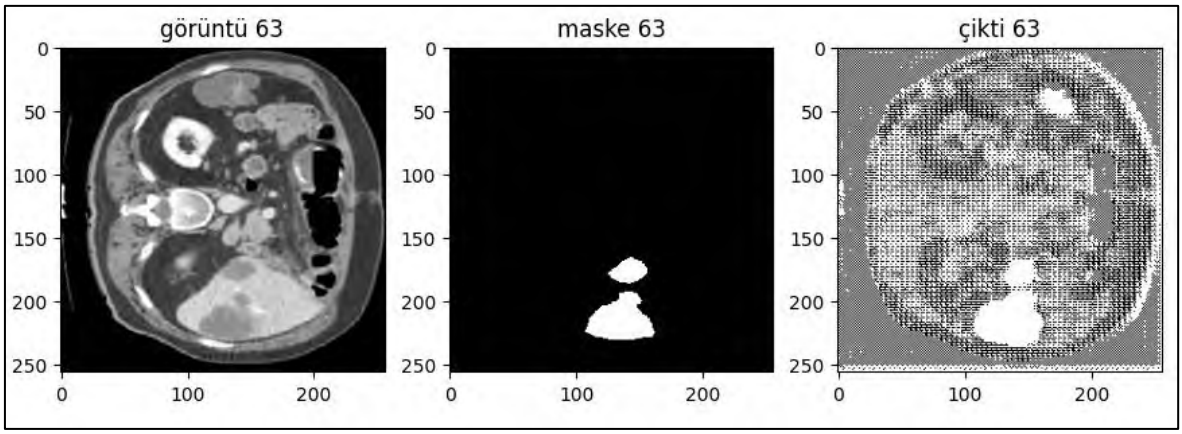
Eşik Değer=0,2 iken Karmaşıklık Matrisi, FPR ve TPR Değerleri;

DP=1745, YP=37371

YN=36, DN=26381

FPR= 0,5861

TPR=0,9797



Şekil 6.7. Eşik değer 0,2 için model çıktısı

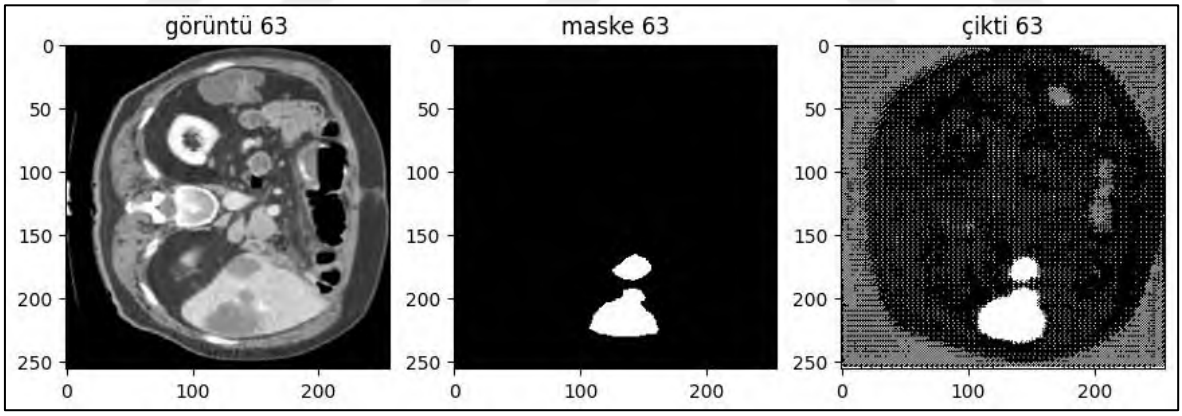
Eşik Değer=0,3 iken Karmaşıklık Matrisi, FPR ve TPR Değerleri;

DP=1658, YP=11093

YN=123, DN=52659

FPR= 0,1741

TPR=0,9311



Şekil 6.8. Eşik değer 0,3 için model çıktısı

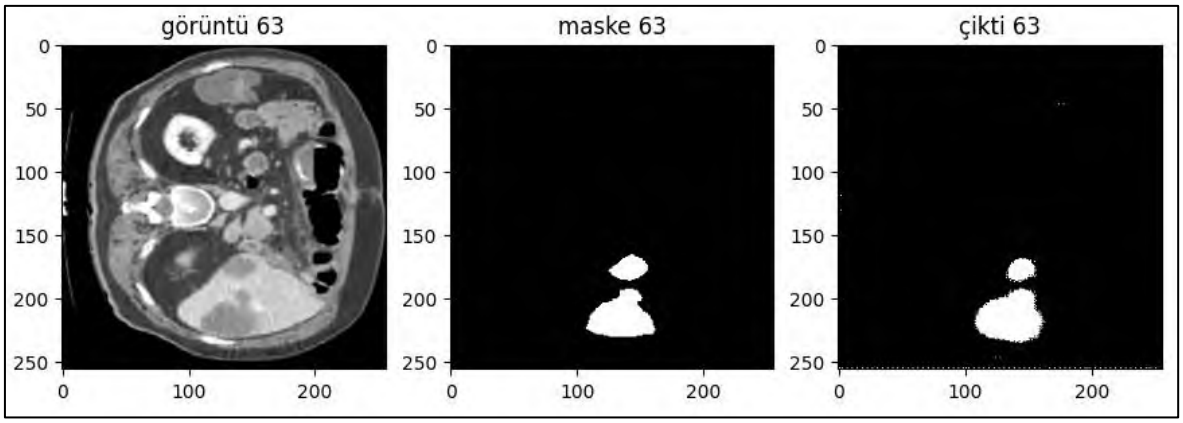
Eşik Değer=0,4 iken Karmaşıklık Matrisi, FPR ve TPR Değerleri;

DP=1603, YP=475

YN=178, DN=63277

FPR=0,0074

TPR=0,9001



Şekil 6.9. Eşik değer 0,4 için model çıktısı

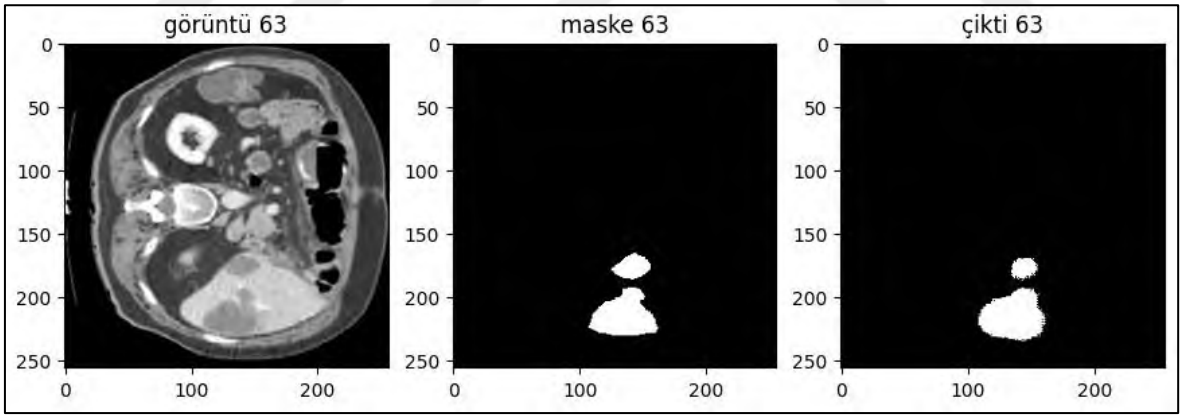
Eşik Değer=0,5 iken Karmaşıklık Matrisi, FPR ve TPR Değerleri;

DP=1566, YP=272

YN=215, DN=63480

FPR=0,0042

TPR=0,8792



Şekil 6.10. Eşik değer 0,5 için model çıktısı

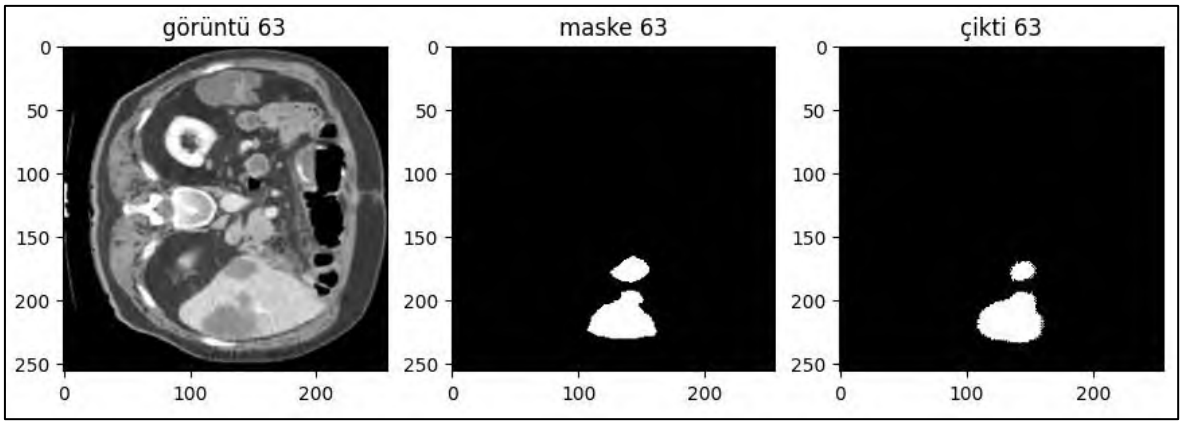
Eşik Değer=0,6 iken Karmaşıklık Matrisi, FPR ve TPR Değerleri;

DP=1528, YP=221

YN=253, DN=63531

FPR=0,0034

TPR=0,8579



Şekil 6.11. Eşik değer 0,6 için model çıktısı

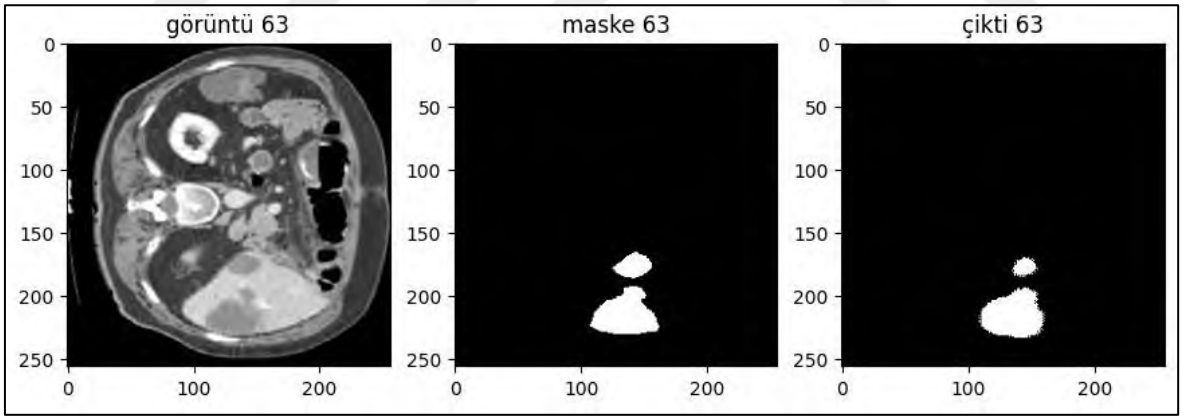
Eşik Değer=0,7 iken Karmaşıklık Matrisi, FPR ve TPR Değerleri;

DP=1475, YP=184

YN=306, DN=63568

FPR=0,0028

TPR=0,8281



Şekil 6.12. Eşik değer 0,7 için model çıktısı

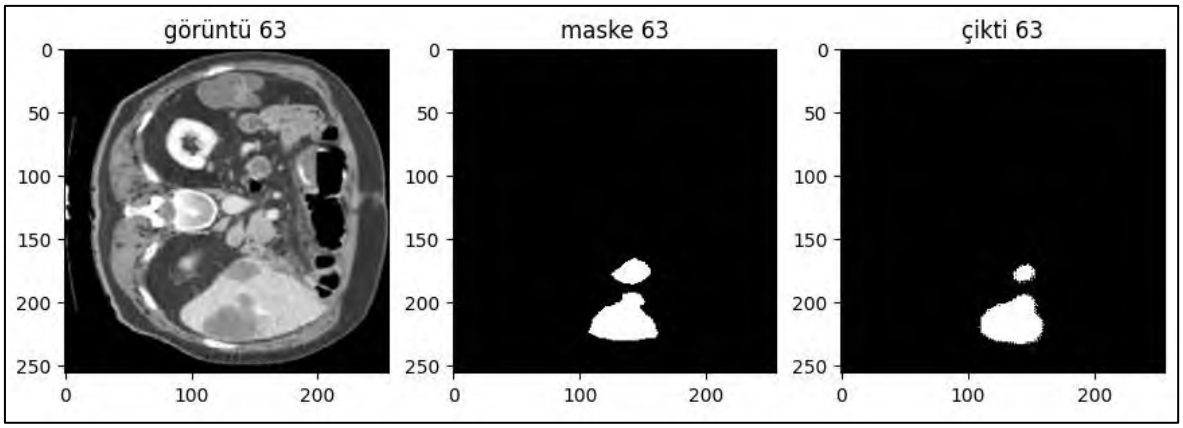
Eşik Değer=0,8 iken Karmaşıklık Matrisi, FPR ve TPR Değerleri;

DP=1425, YP=147

YN=258, DN=63605

FPR=0,0023

TPR=0,8001



Şekil 6.13. Eşik değer 0,8 için model çıktısı

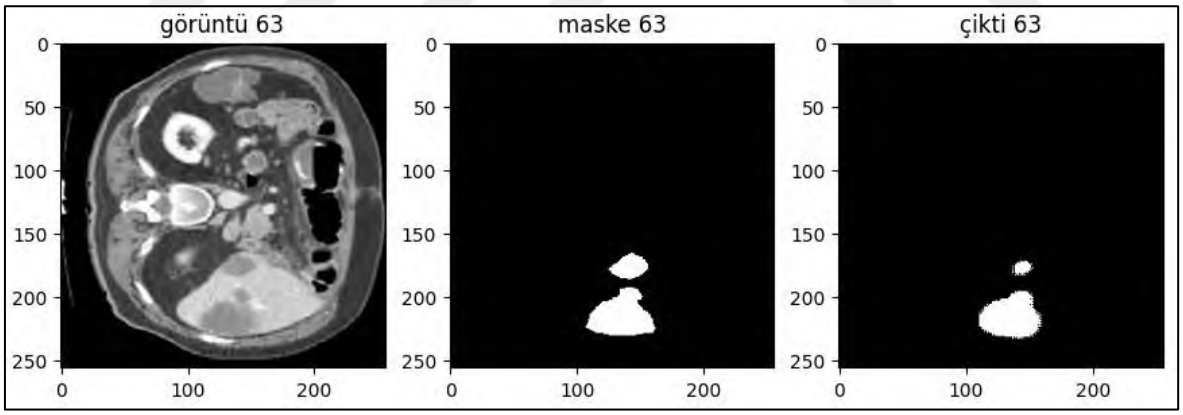
Eşik Değer=0,9 iken Karmaşıklık Matrisi, FPR ve TPR Değerleri;

DP=1342, YP=96

YN=439, DN=63656

FPR=0,0015

TPR=0,7535



Şekil 6.14. Eşik değer 0,9 için model çıktısı

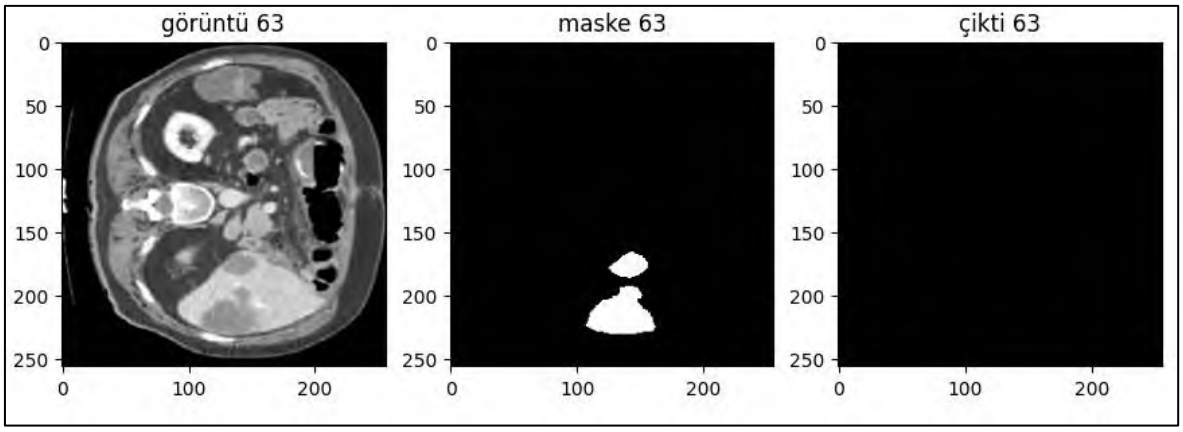
Eşik Değer=1,0 iken Karmaşıklık Matrisi, FPR ve TPR Değerleri;

DP=0, YP=0

YN=0, DN=65533

FPR=0

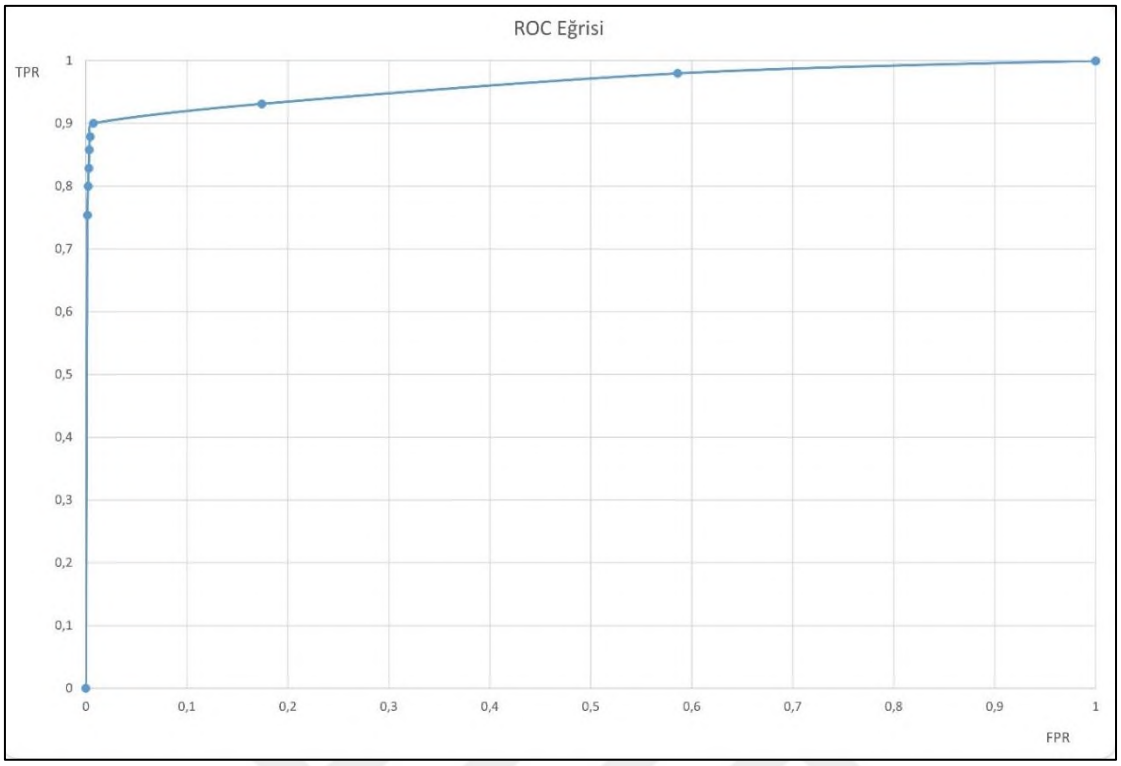
TPR=0



Şekil 6.15. Eşik değer 1,0 için model çıktısı

Çizelge 6.1 Bir görüntü dilimi çıktılarından hesaplanan FPR-TPR değerleri

Eşik Değer	YP	YN	DP	DN	FPR	TPR
0,0	63752	0	1781	0	1	1
0,1	63752	0	1781	3	0,9999	1
0,2	37371	36	1745	26381	0,5861	0,9797
0,3	11093	123	1658	52659	0,1741	0,9311
0,4	475	178	1603	63277	0,0074	0,9001
0,5	272	215	1566	63480	0,0042	0,8792
0,6	221	253	1528	63531	0,0034	0,8579
0,7	184	306	1475	63568	0,0028	0,8281
0,8	147	358	1425	63605	0,0023	0,8001
0,9	96	439	1342	63656	0,0015	0,7535
1,0	0	0	0	65533	0	0



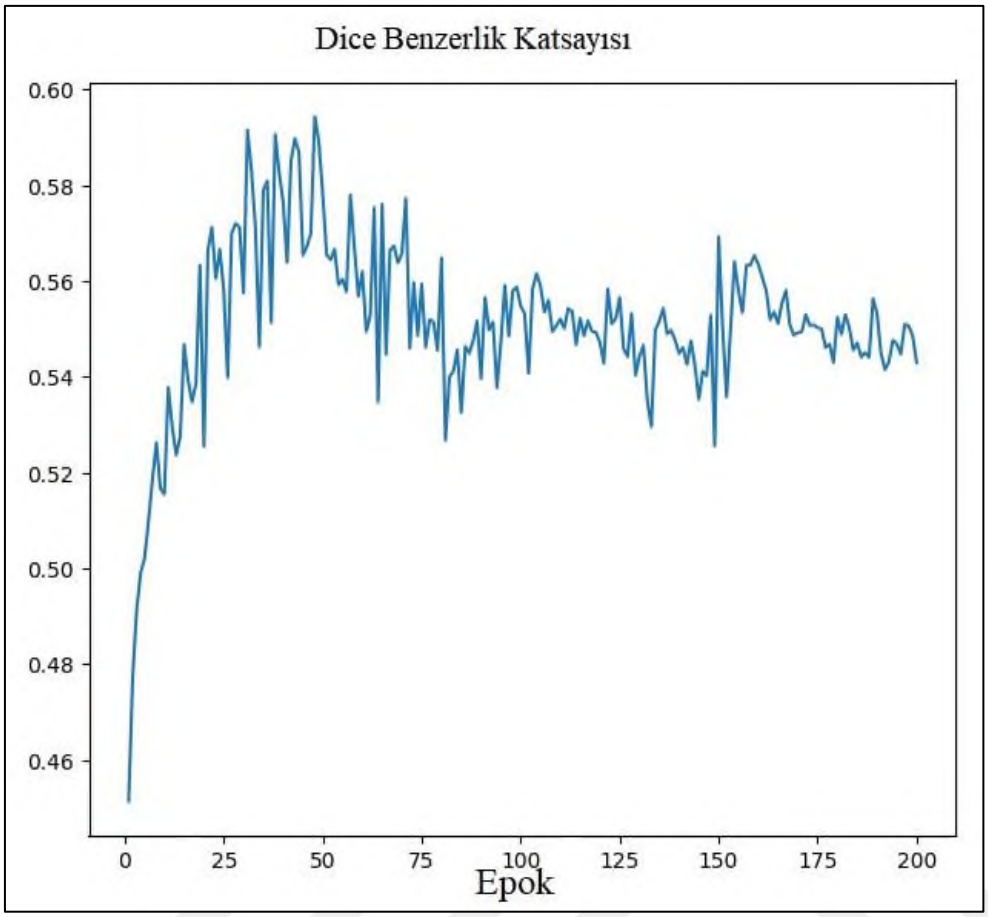
Şekil 6.16. lr=0,00001 için ROC Eğrisi

Şekil 6.16.'da görülen ROC eğrisinin altında kalan alan AUC=0,96 çıkmıştır. Bu rakam ideal durum olan 1'e yakın olması nedeniyle çok iyi bir değerdir.

6.2. Öğrenme Hızı = 0,0001

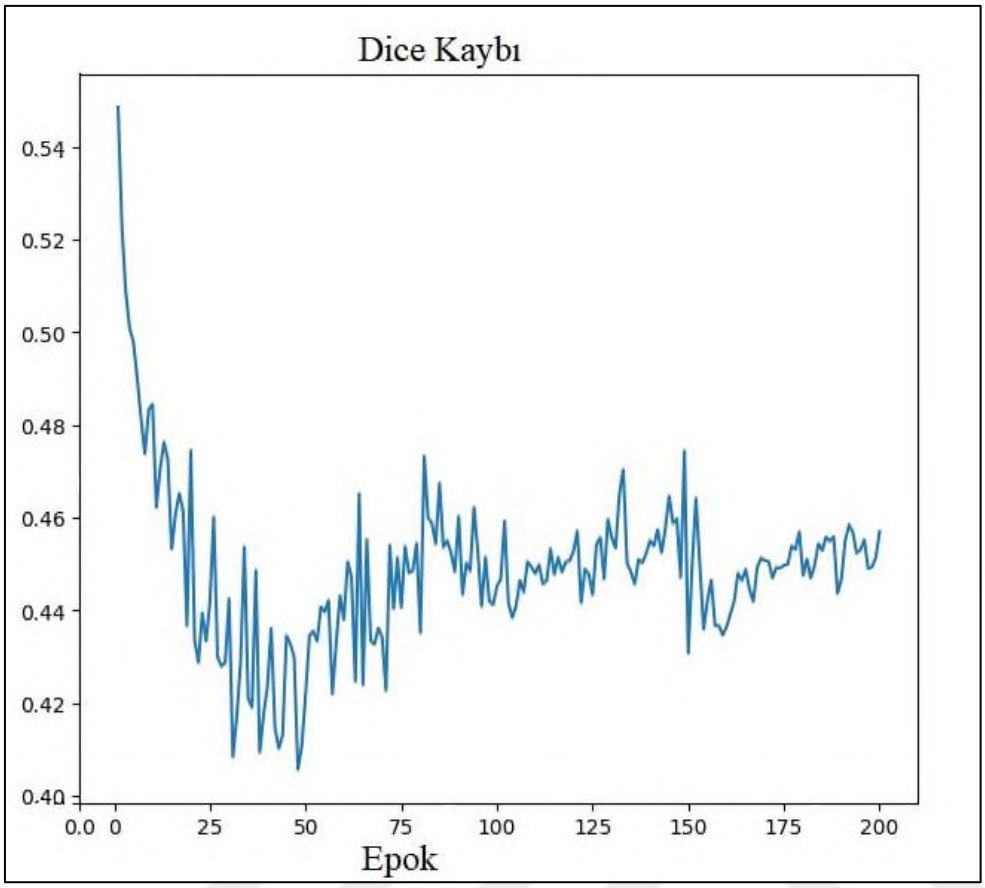
Bu alt bölümde öğrenme hızı Lr = 0,0001 alındığında elde edilen sonuçlar verilmiştir.

Şekil 6.17.'de Dice Benzerlik Katsayısı-Epok Değişim Grafiği verilmiştir. Grafikte de görüldüğü gibi doğrulama verileri ile elde edilen en iyi Dice Benzerlik Katsayısı: 0.5943 olarak sağlanmıştır.



Şekil 6.17. Lr=0,0001 için Dice Benzerlik Katsayısı-Epok Değişim Grafiği

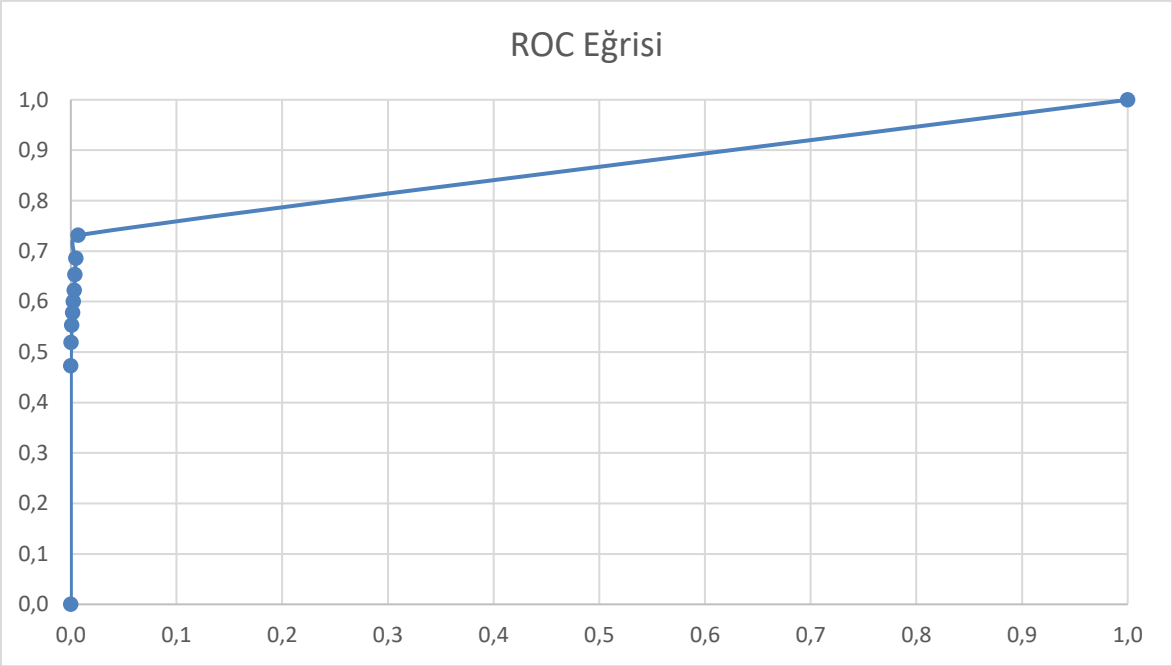
Şekil 6.18.'de doğrulama veri seti ile elde edilen Dice Kaybı-Epok Değişim Grafiği verilmiştir. Grafikte de görüldüğü gibi doğrulama verileri ile elde edilen en iyi Dice Kayıp Değeri: 0,4057 olarak sağlanmıştır.



Şekil 6.18. Lr=0,0001 için Dice Kaybı-Epok Değişim Grafiği

Çizelge 6.2. Bir görüntü dilimi çıktılarından hesaplanan FPR-TPR değerleri

Eşik Değer	YP	YN	DP	DN	FPR	TPR
0,0	63755	0	1781	0	1	1
0,1	443	478	1303	63312	0,0069	0,7316
0,2	312	559	1222	63443	0,0048	0,6861
0,3	255	617	1164	63500	0,0039	0,6535
0,4	206	672	1109	63349	0,0032	0,6226
0,5	156	712	1069	63599	0,0024	0,6002
0,6	119	751	1030	63636	0,0018	0,5783
0,7	63	796	985	63692	0,0009	0,5531
0,8	25	857	924	63730	0,0003	0,5188
0,9	1	939	842	63754	0	0,4727
1,0	0	1781	0	63755	0	0



Şekil 6.19. lr=0,0001 için ROC Eğrisi

Şekil 6.19.'da görülen ROC eğrisinin altında kalan alan AUC=0,86 çıkmıştır. Bu rakam oldukça iyi bir değerdir. Önceki sonuçlarla karşılaştırıldığında; lr=0,00001 ve 104 epok eğitilen modelin ROC eğrisi altında kalan alanın lr=0,0001 ve 200 epok eğitilen modelin ROC eğrisi altında kalan alandan daha büyük olduğu görülmüştür.

7. SONUÇ VE ÖNERİLER

Bu çalışmada, literatürde yayınlanan otomatik karaciğer ve tümör bölütlemesi ile ilgili çalışmalarda elde edilen en iyi sonuçlara göre kısmen daha düşük sonuçlar elde edilmiştir. Elde ettiğimiz sonuç, en iyi skorlar ile kıyaslandığında bir miktar düşük olsa bile iyi görüntü sonuçları sağlanabilmiştir. Bu çalışmada otomatik karaciğer ve tümör bölütlemesi için MONAI yapay zeka kütüphanesinin etkin bir şekilde kullanılabileceği gösterilmiş ve söz konusu kütüphanenin kolaylıkla kullanılabiliyor olması konusunda önemli bir tespit yapılmıştır. MONAI kütüphanesinin diğer biyomedikal görüntü işleme problemleri için de faydalı olacağı anlaşılmıştır.

İki farklı öğrenme hızında farklı epok sayılarında çalıştırılan modelin test verilerinden elde edilen çıktılarına (olasılıkları) piksel bazında farklı eşik değerler uygulanarak elde edilen tahmin görüntüleri çizdirilmiş, FPR ve TPR değerleri hesaplanmış ve ROC eğrileri çizdirilmiştir. İki farklı öğrenme hızında elde edilen sonuçlar için çizdirilen ROC eğrilerini karşılaştırmak için eğriler altında kalan alanlar (AUC) hesaplanmıştır. Bu hesaplamada lr=0,00001 ve 104 epok eğitilen modelin ROC eğrisi altında kalan alanın lr=0,0001 ve 200 epok eğitilen modelin ROC eğrisi altında kalan alandan daha büyük olduğu görülmüştür. Ayrıca lr=0,00001 ve 104 epok eğitilen modelin ROC eğrisinde TPR değerinin daha yüksek olduğu görülmektedir.

Bu çalışma ile tıbbi görüntüleri işlemek için özelleştirilmiş bir çerçeve olan MONAI derin öğrenme kütüphanesi olabildiğince etkin kullanılmaya çalışılmış, çok zengin olan bu kütüphanenin ancak sınırlı sayıdaki bileşenleri ile çalışılabilmiştir.

Derin öğrenmede çok fazla deneme yapmak, farklı parametre ve bileşenler kullanarak modeli sürekli iyileştirmek imkânı olduğundan gelecekte çok daha iyi sonuçlar elde etmek için bu kütüphane ile farklı modeller, farklı dönüşümler, farklı parametreler vb. denenebilir. Ayrıca sürekli güncellenen bu kütüphane ile belli aralıklarla çalışmalar yapılıp geliştirmeler izlenebilir.

Sonraki çalışmalarda tümörün türü, kanserse evresi gibi ek incelemeler yapılabilir. Bu çalışmadan elde edilen tecrübeler farklı organların bölütlenmesinde ve diğer tümör çeşitlerinde uygulanabilir.

KAYNAKLAR

1. Çakaloğlu, Y. (2018) Hepatoselüler Karsinomada Tanı, Evreleme ve Tedavi: 2018-Aasld Practice Guidance. *Hepatology*, 68,723-750.

2. İnternet: Dünya Sağlık Örgütü IARC 2020 Dünya Kanser Araştırmaları Web: https://www.iarc.who.int/news-events/new-report-on-global-cancer-burden-in-2022-by-world-region-and-human-development-level/., Son Erişim Tarihi: 20.08.2024.

3. Albain, K.S., Swann, R.S., Rusch, V.W., Turrisi, A.T., Shepherd, F.A., Smith C., Chen, Y., Livingston, R.B., Feins, R.H. (2009). Radiotherapy plus chemotherapy with or without surgical resection for stage III non-small-cell lung cancer: A phase III randomised controlled trial, *The Lancet*, 374, 379-386.

4. Shiina, S., Sato, K., Tateishi R., Shimizu M., Ohama H., Hatanaka T., Takawa, M., Nagamatsu, H., Imai, Y. (2018). Percutaneous ablation for hepatocellular carcinoma: Comparison of various ablation techniques and surgery. *Canadian Journal of Gastroenterology and Hepatology*, 2018(3),1-8.

5. Todorov, M., Paetzold, J.C., Schoppe, O., Tetteh, G., Shit, S., Efremov, V., Todorov-Völgyi, K., Düring, M., Dichgans, M., Piraud, M. (2020). Machine learning analysis of whole mouse brain vasculature, *Nature Methods*, 17(5), 1-8.

6. Moghbel, M., Mashohor, S., Mahmud, R., Saripan, M.I.B. (2017). Review of liver segmentation and computer assisted detection/diagnosis methods in computed tomography. *Artificial Intelligence Review*, 50(4), 497- 537.

7. Christ, P., Elshaer, M.E.A., Ettlinger, F., Tatavarty, S., Bickel, M., Bilic, P., Rempfler, M., Armbruster, M., Hofmann, F., D'Anastasi, M., Sommer, W.H., Ahmadi, S.A., Menze, B.H. (2016). *Automatic Liver and Lesion Segmentation in CT Using Cascaded Fully Convolutional Neural Networks and 3D Conditional Random Fields.* In: Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016: 19th International Conference, Anhens, 415–423.

8. Soler, L., Delingette, H., Malandain, G., Montagnat, J., Ayache, N., Koehl, C., Dourthe, O., Malassagne, B., Smith, M., Mutter, D. (2001). Fully automatic anatomical, pathological, and functional segmentation from CT scans for hepatic surgery. *Computer Aided Surgery, 6(3), 131-142.*

9. Ciecholewski, M., Ogiela, M.R. (2007). *Automatic segmentation of single and multiple neoplastic hepatic lesions in CT images.* International Work-Conference on the Interplay Between Natural and Artificial Computation, La Manga del Mar Menor, 63-71.

10. Nugroho, H.A., Ihtatho, D., Nugroho, H. (2008). Contrast enhancement for liver tumor identification. *Medical Image Computing and Computer-Assisted Intervention, MICCAI Workshop*, 41 (2008), 201.

11. Abdel-Massieh, N.H., Hadhoud, M.M., Amin, K.M. (2010). Fully automatic liver tumor segmentation from abdominal CT scans. *Computer Engineering and Systems. (ICCES), 2010 International Conference on, Institıde of Electrical and Electronics Engineers IEEE (2010*), Cairo, 197-202.

12. Moltz, J.H., Bornemann, L., Kuhnigk, J.-M., Dicken, V., Peitgen, E., Meier, S., Bolte, H., Fabel, M., Bauknecht, H.C., Hittinger, M. (2009). Advanced segmentation techniques for lung nodules, liver metastases, and enlarged lymph nodes in CT scans. *Institıde of Electrical and Electronics Engineers Journal of Selected Topics in Signal Processing*., 3 (1), 122-134.

13. Kass, M., Witkin, A., Terzopoulos, D. (1988). *Active contour models*, International Journal of Computer Vision, 321-331.

14. Ben-Dan, I., Shenhav, E. (2008). *Liver tumor segmentation in CT images using probabilistic methods.* MICCAI Workshop 2008, 41, 43.

15. Linguraru, M.G., Richbourg, W.J., Liu, J., Watt, J.M., Pamulapati, V., Wang, S., Summers, R.M., (2012). Tumor burden analysis on computed tomography by automated liver and tumor segmentation, *IEEE Transactions on Medical Imaging,* 31(10),1965-1976.

16. Osher, S., Sethian, J.A. (1988). Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations, *Journal of Computational Physics*, 79(1), November 1988, 12-49.

17. Smeets, D., Stijnen, B., Loeckx, D., De Dobbelaer, B., Suetens, P. (2008). Segmentation of liver metastases using a level set method with spiral-scanning technique and supervised fuzzy pixel classification. *Medical Image Analysis,* 14(1), 13-20.

18. Jiménez Carretero, D., Fernández de Manuel, L., Pascau González Garzón, J., Tellado, J.M., Ramon, E., Desco Menéndez, M., Santos, A., Ledesma Carbayo, M.J. (2011). *Optimal multiresolution 3D level-set method for liver segmentation incorporating local curvature constraints.* 2011 Annual International Conference of the Institute of Electrical and Electronics Engineers Engineering in Medicine and Biology Society, Boston, 3419-3422.

19. Massoptier, L., Casciaro, S., (2008). A new fully automatic and robust algorithm for fast segmentation of liver tissue and tumors from CT scans. *European Radiology Journal.*, 18(9), 1867–1874.

20. Häme, Y. (2008). Liver tumor segmentation using implicit surface evolution. *Medical Image Analysis,* 12(6), 709-725.

21. Christ, P., Elshaer, M.E.A., Ettlinger, F., Tatavarty, S., Bickel, M., Bilic, P., Rempfler, M., Armbruster, M., Hofmann, F., D'Anastasi, M., Sommer, W.H., Ahmadi, S.-A., Menze, B.H., (2016). *Automatic Liver and Lesion Segmentation in CT Using Cascaded Fully Convolutional Neural Networks and 3D Conditional Random Fields*. Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016: 19th International Conference, Lyon, 415–423.

22. Sun, C., Guo, S., Zhang, H., Li, J., Chen, M., Ma, S., Jin L., Liu X., Li, X., Qian, X., (2011). Automatic segmentation of liver tumors from multiphase contrast-enhanced CT images based on FCNs. *Artificial Intelligence in Medicine. 53(3)*, 139-147.

23. Chlebus, G., Schenk, A., Moltz, J.H., van Ginneken, B., Hahn, H.K., Meine, H. (2018). Automatic liver tumor segmentation in CT with fully convolutional neural networks and object-based postprocessing. *Scintific Reports,* 8(1), 1-7.

24. Zhe, L., Yu-Qing, S., Sheng, Z., Victor, S., Liangmin, W., Rui, J., Xiaolin, Z., Deqi, Y. (2019). Liver CT sequence segmentation based with improved U-Net and graph cut. *Expert Systems with Applications,* 126, 54-63.

25. Qin, W. (2018). Superpixel-based and boundary-sensitive convolutional neural network for automated liver segmentation. *Institute of Physics and Engineering in Medicine,* 63(9), 324.

26. Han, Y. (2017). Framing U-Net via deep convolutional framelets: Application to sparse-view CT. *Institıde of Electrical and Electronics Engineers Transactions on Medical Imaging,* 37(6), 1417-1419.

27. Men, K., Boimel, P, Janopaul-Naylor, J., Zhong, H., Huang, M., Geng, H. (2018). Cascaded atrous convolution and spatial pyramid pooling for more accurate tumor target segmentation for rectal cancer radiotherapy. *Institute of Physics and Engineering in Medicine,* 63(18), 68.

28. Li, X., Chen, H., Qi, X., Duo, Q., Fu, C., Heng, P. (2018). H-DenseUNet: Hybrid Densely Connected UNet for Liver and Tumor Segmentation from CT Volumes. *Institıde of Electrical and Electronics Engineers Transactions on Medical Imaging,* 37(12)., 2663-2674.

29. Sao, H., Huang, C., Basenne, M., Xiao, R., Xing, L. (2020). Modified U-Net (mU-Net) with Incorporation of Object-Dependent High Level Features for Improved Liver and Liver-Tumor Segmentation in CT Images. *Institıde of Electrical and Electronics Engineers Transactions on Medical Imaging,* 39(5), 1316-1325.

30. Budak, Ü., Guo, Y., Tanyildizi, E., Şengür, A. (2020). Cascaded deep convolutional encoder-decoder neural networks for efficient liver tumor segmentation. *Medical Hypotheses,* 142(1), 109711.

31. Nabiyev, V. V. (2012). *Yapay Zekâ: İnsan-Bilgisayar Etkileşimi*. Ankara: Seçkin Yayıncılık, 20-38.

32. İnternet: Sinir ağı nedir? Web: https://aws.amazon.com/tr/what-is/neural-network. Son Erişim Tarihi: 12.06.2024.

33. Öztemel, E. (2020). *Yapay Zekâ ve İnsanlığın Geleceği, Bilişim Teknolojileri ve İletişim: Birey ve Toplum Güvenliği.* (1. Baskı). Ankara: Türkiye Bilimler Akademisi ,75-90.

34. İnternet: Orhan, U. Makine Öğrenmesi Ders Notları. Web: https://ceng.cu.edu.tr/uorhan/DersNotu/Ders06.pdf. Son Erişim Tarihi: 19.04.2023

35. İnternet: What is neuron? Web: http://www.mplsvpn.info/2017/11/what-is-neuron-and-artificial-neuron-in.html. Son Erişim Tarihi:11.04.2024.

36. İnternet: Yapay Sinir Ağları. Web: https://www.veribilimiokulu.com/yapay-sinir-aglari/ Son Erişim Tarihi:18.04.2024.

37. İnternet: Artificial Neural Network. Web: https://www.sciencedirect.com/topics/physics-and-astronomy/artificial-neural-network Son Erişim Tarihi: 03.05.2024.

38. Krizhevsky, A., Sutskever, I. and Hinton, G.E. (2017). *Imagenet classification with deep convolutional neural networks*. Communications of the ACM, 60(6), 84-90.

39. Kiran, P., Razman, A.M., Abdullah, M.A., Ab Nasir. A.F., (2023). Sign Language Recognition using Deep Learning through LSTM and CNN. *Journal of Scientific & Industrial Research*, 82(07), 67-71.

40. Anirudha, G., (2020). Fundamental Concepts of Convolutional Neural Network. *Recent Trends and Advances in Artificial Intelligence and Internet of Things, 5-10.*

41. İnternet: Deep Learning. Web: https://medium.com/deep-learning-from-deepest/katmanlar%C4%B1-anlamak-e6b33371f0f8 Son Erişim Tarihi: 03.02.2024.

42. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *The Journal of Machine Learning Research, 15(1), 1929.*

43. Han, S., (2016). *Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding*. International Conference on Learning Representation, San Juan, 409(1), 244–258.

44. Ronneberger, O., Fisher, P., Brox, T. (2015). *U-Net: Convolutional Networks for Biomedical Image Segmentation*. Medical Image Computing and Computer-Assisted Intervention, Munich, 387-395.

45. Liu, L., Cheng, J., Quan, Q., Wu, F.-X., Wang, Y.-P. ve Wang, J. (2020). A survey on U-shaped networks in medical image segmentations. *Neurocomputing*, 409, 244–258.

46. Mason, S.J., Graham, N.E. (2002). Areas beneath the relative operating characteristics (ROC) and relative operating levels (ROL) curves. *Statistical significance and interpretation. Quarterly Journal of the Royal Meteorological Society*, 128, 2145–2166.

47. İnternet: Deep Learning Courses. Web: https://www.youtube.com/watch?v=Joh3LOaG8Q0&t=1018s Son Erişim Tarihi:18.08.2024.

48. Bilic, P., Christ P., Voronhsov, E. (2022). The Liver Tumor Segmentation Benchmark (LiTS). *Medical Image Analysis*, 84(1), 102680.

49. İnternet: Online Açık Kaynak Kodlu Monai Yapay Zaka Kütüphanesi Web: https://docs.monai.io/en/stable/networks.html#unet Son Erişim Tarihi:11.10.2024



Gazili olmak ayrıcalıktır