

**T.C.  
AKDENİZ ÜNİVERSİTESİ**



**DERİN ÖĞRENME İLE KORPUS KALLOZUM SEGMENTASYONU VE İNDEKS  
HESAPLAMASI**

**Tahirhan YILDIZOĞLU**

**FEN BİLİMLERİ ENSTİTÜSÜ  
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ  
ANABİLİM DALI  
YÜKSEK LİSANS TEZİ**

**OCAK 2022**

**ANTALYA**

**T.C.  
AKDENİZ ÜNİVERSİTESİ**



**DERİN ÖĞRENME İLE KORPUS KALLOZUM SEGMENTASYONU VE İNDEKS  
HESAPLAMASI**

**Tahirhan YILDIZOĞLU**

**FEN BİLİMLERİ ENSTİTÜSÜ  
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ  
ANABİLİM DALI  
YÜKSEK LİSANS TEZİ**

**OCAK 2022**

**ANTALYA**

**T.C.  
AKDENİZ ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**DERİN ÖĞRENME İLE KORPUS KALLOZUM SEGMENTASYONU VE İNDEX  
HESAPLAMASI**

**Tahirhan YILDIZOĞLU**

**ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ  
ANABİLİM DALI**

**YÜKSEK LİSANS TEZİ**

Bu tez 21/01/2022 tarihinde jüri tarafından Oybirliği / Oyçokluğu ile kabul edilmiştir.

Dr. Öğr. Üyesi Yalçın ALBAYRAK (Danışman)  
Dr. Öğr. Üyesi Onur KARAMAN  
Dr. Öğr. Üyesi Fatih Ahmet ŞENEL

## ÖZET

### DERİN ÖĞRENME İLE KORPUS KALLOZUM SEGMENTASYONU VE İNDEKS HESAPLAMASI

Tahirhan Yıldızoğlu

Yüksek Lisans Tezi, Elektrik-Elektronik Mühendisliği Anabilim Dalı

Danışman: Dr. Öğr. Üyesi Yalçın ALBAYRAK

Ocak 2022; 42 sayfa

Bu çalışmada, Beyin kesiti resmi üzerinden korpus kallozum indeksini otomatik olarak hesaplayabilecek bir sistem geliştirilmiştir. Derin öğrenme metotları kullanılarak geliştirilen bu çözüm ile radyoloji uzmanları için yardımcı bir karar destek sistemi oluşturulmuştur. MS, Alzheimer, şizofreni, disleksi, ve epilepsi hastalıklarının tespiti ve takibi işlemlerinde, Radyoloji uzmanlarının otomatize edilmiş KKİ hesabı ile zaman tasarrufu sağlamaları amaçlanmıştır.

Sunulan çözümde öncelikle seçilen derin öğrenme mimarisinin eğitimi gerçekleştirilmiştir. Eğitim işlemi için gerekli olan veri seti Akdeniz Üniversitesi Radyoloji bölümü yardımı ile edinilmiştir. Google Colaboratoy platformunda TensorFlow ve Keras kütüphaneleri ile gerçekleştirilen eğitim sonucunda elde edilen model ile segmentasyon sistemi hazırlanmıştır. Girdi olarak beyin MR kesiti kullanılmaktadır. Alınan medikal resim üzerinde öncelikle eğitilmiş U-Net derin öğrenme modeli ile segmentasyon işlemi gerçekleştirilir. Ardından segmente edilmiş resim üzerinde korpus kallozum indeks hesaplama algoritması ile kalınlık ve uzaklık değerleri elde edilir. Kalınlık ve uzaklık değerleri indeks denklemine yerleştirilerek sonuç değeri bulunmaktadır. Çalışma sonucunda MS subtiplerine göre KKİ değer dağılımları elde edilmiştir.

**ANAHTAR KELİMELER:** Derin Öğrenme, Korpus Kallozum, Korpus Kallozum İndeksi, , Segmentasyon.

**JÜRİ:** Dr. Öğr. Üyesi Yalçın Albayrak

Dr. Öğr. Üyesi Onur KARAMAN

Dr. Öğr. Üyesi Fatih Ahmet ŞENEL

## **ABSTRACT**

### **CORPUS CALLOSUM INDEX CALCULATION AND SEGMENTATION WITH DEEP LEARNING**

**Tahirhan YILDIZOĞLU**

**MSc Thesis in Electrical-Electronics Engineering**

**Supervisor: Assist. Prof. Dr. Yalçın ALBAYRAK**

**January 2022; 42 pages**

In this study, a system with capability of calculating corpus callosum index automatically is developed. Deep learning techniques are used in the solution that aims to be an assistant recommender system to radiology specialists. Main goal of the provided system is to make a positive effect on time management of radiology specialists by automating CCI calculation at detection and control of MS, Alzheimer, schizophrenia, dyslexia, and epilepsy.

First step for the study was training the selected deep learning architecture. Training dataset is gathered from the Radiology department of the Akdeniz University. Segmentation pipeline built with trained model. Input for the segmentation pipeline is brain MR image. For the CCI calculation, first step is to detect corpus callosum with trained U-Net model. After the segmentation process, an algorithm is used for finding the thickness and length values of detected corpus callosum. With the obtained values, corpus callosum index is calculated. As a result of the thesis work CCI distribution with respect to MS sub-types is obtained.

**KEYWORDS:** Corpus Callosum, Corpus Callosum Index, Deep Learning, Segmentation.

**COMMITTEE:** Assist. Prof. Dr Yalçın Albayrak

Assist. Prof. Dr Onur KARAMAN

Assist. Prof. Dr Fatih Ahmet ŞENEL

## ÖNSÖZ

Gelişmiş teknolojinin en ileri seviye uygulama alanlarından biri olan Bilgisayar Görü alanında yapılan bu çalışma ile Radyoloji alanında medikal görüntüler üzerinde otonom tespit ve hesaplama yapabilen bir sistem geliştirilmiştir. Çalışma kapsamında derin öğrenme uygulama alanlarından biri olan segmentasyon işlemi gerçekleştirilmiştir. Projede kullanılan U-Net mimarisi üzerinde tecrübe ve bilgi edinilmiştir.

Çalışma boyunca bilgi ve tecrübelerini benden esirgemeyen, tez konumun belirlenmesinde yardımcı olan, öneri ve yardımları ile tezime katkıda bulunan, tez danışman hocam Sayın Dr. Öğr. Üyesi Yalçın ALBAYRAK’a teşekkür ve saygılarımı sunarım.

Eğitim-öğretim ve çalışma hayatım boyunca bana desteklerini esirgemeyen, her konuda bana sonsuz güvenen aileme ve Sayın Y. Mim. Nurcan İLERİ’ye teşekkürlerimi ve minnettarlığımı sunarım.

## İÇİNDEKİLER

|                                                            |      |
|------------------------------------------------------------|------|
| ÖZET .....                                                 | i    |
| ABSTRACT .....                                             | ii   |
| ÖNSÖZ .....                                                | iii  |
| AKADEMİK BEYAN .....                                       | vi   |
| KISALTMALAR .....                                          | vii  |
| ŞEKİLLER DİZİNİ .....                                      | viii |
| 1. GİRİŞ .....                                             | 1    |
| 2. KAYNAK TARAMASI .....                                   | 6    |
| 3. MATERYAL VE METOT .....                                 | 10   |
| 3.1. Veri Seti .....                                       | 10   |
| 3.2. Python Programlama Dili .....                         | 12   |
| 3.3. Pandas Kütüphanesi .....                              | 12   |
| 3.4. Google Colaboratory Platformu .....                   | 13   |
| 3.5. Tensorflow ve Keras Derin Öğrenme Kütüphaneleri ..... | 13   |
| 3.6. Numpy Kütüphanesi .....                               | 14   |
| 3.7. OpenCV Kütüphanesi .....                              | 14   |
| 3.8. Matplotlib Görselleştirme kütüphanesi .....           | 14   |
| 3.9. Scikit-learn Kütüphanesi .....                        | 15   |
| 3.10. Labelbox Platformu .....                             | 15   |
| 3.11. U-Net Mimarisi .....                                 | 16   |
| 3.12. Segmentasyon Metrikleri .....                        | 21   |
| 4. BULGULAR VE TARTIŞMA .....                              | 25   |
| 4.1 Model Seçimi .....                                     | 25   |
| 4.2 Veri Seti Oluşturma .....                              | 25   |
| 4.3. Model Eğitimi .....                                   | 29   |
| 4.4. Korpus Kallozum İndeks Hesaplama Algoritması .....    | 32   |
| 5. SONUÇLAR .....                                          | 34   |
| 6. KAYNAKLAR .....                                         | 38   |

## ÖZGEÇMİŞ



## AKADEMİK BEYAN

Yüksek Lisans Tezi olarak sunduğum “Derin Öğrenme ile Korpus Kallozum Segmentasyonu ve İndeks Hesaplaması” adlı bu çalışmanın, akademik kurallar ve etik değerlere uygun olarak yazıldığını belirtir, bu tez çalışmasında bana ait olmayan tüm bilgilerin kaynağını gösterdiğimi beyan ederim.

21/01/2022

Tahirhan YILDIZOĞLU

## KISALTMALAR

### **Kısaltmalar**

|        |                                                                                                              |
|--------|--------------------------------------------------------------------------------------------------------------|
| KKİ    | : Korpus kallozum indeksi                                                                                    |
| MS     | : Multipl Skleroz                                                                                            |
| API    | : Uygulama programlama arayüzü (Application Programming Interface)                                           |
| CNN    | : Evrişimsel sinir ağı (Convolutional Neural Network)                                                        |
| IoU    | : Birleşim üzerindeki kesişim (Intersection Over Union)                                                      |
| IoT    | : Nesnelerin interneti (Internet of Things)                                                                  |
| ISBI   | : Uluslararası Biyomedikal Görüntüleme Sempozyumu (International Symposium on Biomedical Imaging)            |
| CSV    | : Virgül ile ayrılmış değerler (Comma-Separated Values)                                                      |
| MR     | : Manyetik rezonans                                                                                          |
| AMS    | : Uyarlamalı ortalama kayması (Adaptive Mean Shift)                                                          |
| GAC    | : Geometrik aktif eş düzey çizgisi (Geometric Active Contour)                                                |
| ACI    | : Otomatik korpus kallozum eş düzey çizgisi tanımlama (Automatic Corpus Callosum Contour Initialization)     |
| MRI    | : Manyetik rezonans görüntüleme (Magnetic Resonance Imaging)                                                 |
| SVM    | : Destek vektör makinesi (Support Vector Machine)                                                            |
| STAPLE | : Zaman uyumlu doğruluk ve performans seviyesi tahmini (Simultaneous Truth and Performance Level Estimation) |

## ŞEKİLLER DİZİNİ

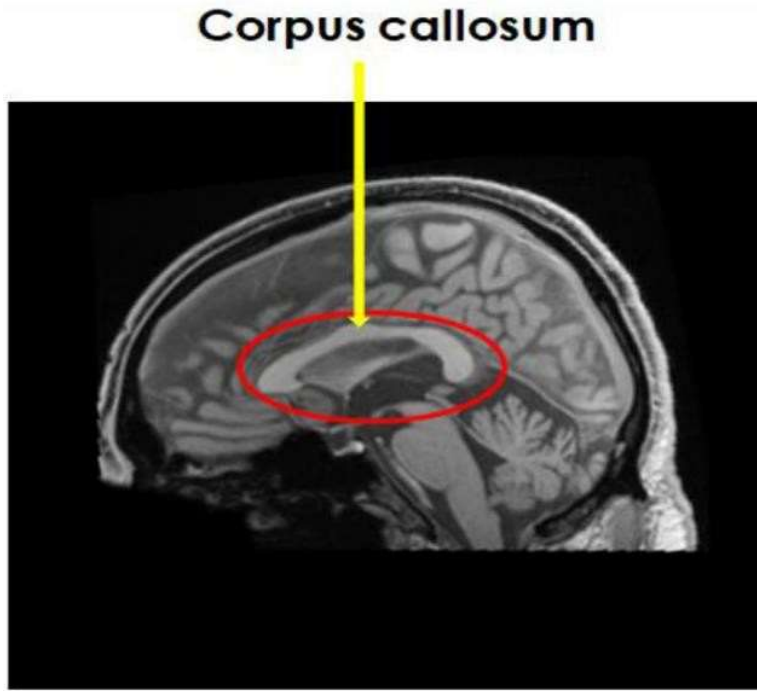
|                                                                                                                           |    |
|---------------------------------------------------------------------------------------------------------------------------|----|
| Şekil 1.1 İnsan beyni içerisinde yer alan Korpus Kallozum dokusunun 2 boyutlu resmi .....                                 | 1  |
| Şekil 1.2 Sağlık birey ile MS hastası bireyin korpus kallozum dokusunun karşılaştırması...                                | 2  |
| Şekil 1.3 Korpus kallozum indeks denklemi ve morfolojik ölçüm gösterimi .....                                             | 3  |
| Şekil 1.4 Derin öğrenme metodları ile medikal resimler üzerinde elde edilen segmentasyon sonuçları .....                  | 4  |
| Şekil 1.5 Çalışma kapsamında sunulan çözüm diyagramı .....                                                                | 5  |
| Şekil 2.1 Park vd. (2018) çalışmasına ait segmentasyon sonuçları.....                                                     | 6  |
| Şekil 2.2 Li vd. (2017)'nin sundukları çözümün sonuçları.....                                                             | 7  |
| Şekil 2.3 DeepnCCA algoritmasının çıktılarından örnekler .....                                                            | 8  |
| Şekil 2.4 Herrera vd. (2018) çalışmasında sundukları piksel tabanlı sınıflandırıcının sonuçları .....                     | 9  |
| Şekil 3.1.1 ABIDE ve OASIS setleri ile eğitilen modelin Radyoloji bölümü örnekleri üzerindeki segmentasyon sonuçları..... | 10 |
| Şekil 3.1.2 Tez çalışması kapsamında kullanılan veri setlerinden MRI örnekleri .....                                      | 11 |
| Şekil 3.10.1 Labelbox platformunda korpus kallozum etiketi .....                                                          | 15 |
| Şekil 3.11.1 U-Net Mimarisi.....                                                                                          | 17 |
| Şekil 3.11.2 ReLU cevap grafiği.....                                                                                      | 18 |
| Şekil 3.11.3 2x2 filtre ile gerçekleştirilen Max-Pool işlemi .....                                                        | 18 |
| Şekil 3.11.4 3x3 filtre ile gerçekleştirilen yukarı evrişim işlemi .....                                                  | 19 |
| Şekil 3.11.5 64x64x192 boyutlu öznetelik haritasına uygulanan 1x1 evrişim işlemi .....                                    | 19 |
| Şekil 3.11.6 5 katmanlı öznetelik aktarım örneği .....                                                                    | 20 |
| Şekil 3.11.7 Evrişim katmanı ile ilgili görselleştirme.....                                                               | 20 |
| Şekil 3.12.1 Pixel Accuracy metriği için segmentasyon örneği.....                                                         | 21 |

|                                                                                                                          |    |
|--------------------------------------------------------------------------------------------------------------------------|----|
| <b>Şekil 3.12.2</b> Şekil 3.12.1 için çıkan segmentasyon sonucu.....                                                     | 22 |
| <b>Şekil 3.12.3</b> IoU görseli .....                                                                                    | 23 |
| <b>Şekil 3.12.4</b> Dice Katsayısı hesaplama yöntemi görseli .....                                                       | 23 |
| <b>Şekil 4.2.1</b> Etiket dosyasının işlenmesi ile ilgili akış diyagramı .....                                           | 26 |
| <b>Şekil 4.2.2</b> Labelbox platformundan alınan etiket verilerinin işlenmesi ile elde edilen veri seti görselleri ..... | 27 |
| <b>Şekil 4.2.3</b> Veri setinin model eğitime uygun hale dönüştürülmesi için uygulanan işlemlerin akış diyagramı .....   | 28 |
| <b>Şekil 4.2.4</b> Rotasyon işlemi ile veri zenginleştirme örneği.....                                                   | 29 |
| <b>Şekil 4.3.1</b> U-Net modelinin katmanları, çıkış boyutları, ve bağlantı noktaları .....                              | 29 |
| <b>Şekil 4.3.2</b> Model eğitimi süresince Dice Katsayısı değişim grafiği .....                                          | 32 |
| <b>Şekil 5.1</b> Kontrol grubu segmentasyon sonuçları .....                                                              | 34 |
| <b>Şekil 5.2</b> Kontrol grubu segmentasyon sonuçları .....                                                              | 34 |
| <b>Şekil 5.3</b> Örnek Korpus Kallozum İndeksi hesabı.....                                                               | 35 |
| <b>Şekil 5.4</b> Çalışılan veri seti üzerinde elde edilen korpus kallozum indeksi dağılımı.....                          | 36 |
| <b>Şekil 5.5</b> MS subtiplerine göre Korpus Kallozum İndeksi dağılımları.....                                           | 37 |

## 1. GİRİŞ

Radyoloji uzmanları, hastalık tespiti, erken tanı ve hastalık takibi konusunda beyin dokuları üzerinde elde ettikleri ölçüm değerlerini referans olarak kullanabilmektedir. Beyin dokusu ölçümleri ile tespiti ve takibi yapılabilen hastalıklardan biri olan MS hastalığı, 20-40 yaş grubundaki genç insanları etkileyen ve merkezi sinir sisteminde meydana gelen kronik iltihaplanmalı bir hastalıktır (Ozer 2021). Radyoloji uzmanlarının MS hastalığı tespiti ve takibi için referans olarak kullandığı değerlerden biri de korpus kallozum indeksidir.

Korpus kallozum beyindeki en büyük beyaz madde yapısıdır ve latince “sert cisim” anlamına gelir. Kalın bir sinir lifleri bantı olan korpus kallozum, serebral korteks loblarını, sol ve sağ yarım küreler olarak ikiye böler. Görevi iki yarım küre arasında iletişimi sağlamaktır. Korpus kallozum, hareketssel, duyuusal ve bilişsel bilgiyi, yarım küreler arasında aktarır (Fitsiori vd. 2011).

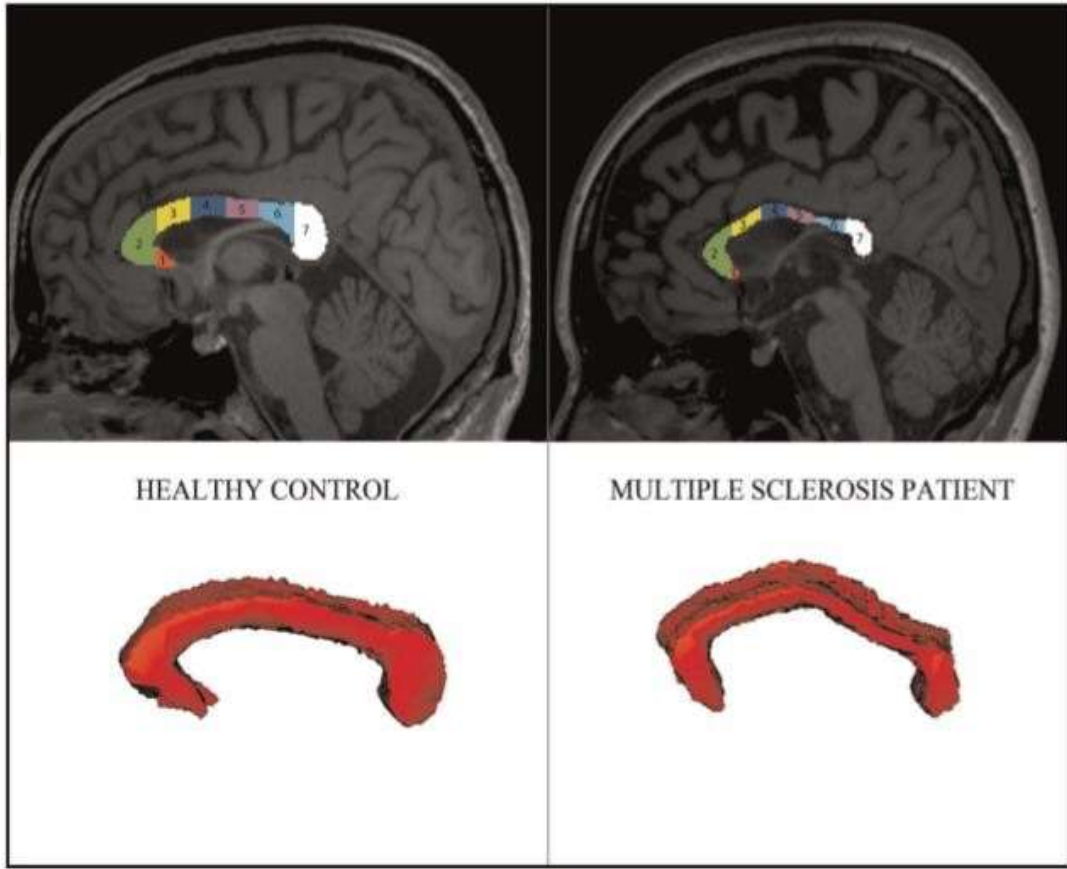


**Şekil 1.1** İnsan beyni içerisinde yer alan Korpus Kallozum dokusunun 2 boyutlu resmi (Chandra vd. 2022)

Korpus kallozum indeksi, korpus kallozum üzerinde yapılan morfolojik ölçümlerle elde edilir ve beyin hacmindeki değişiklikleri yansıtan bir değerdir. Beyin atrofisi

incelemelerinde KKI gösterge değeri olarak kullanılmaktadır (Yaldizli vd. 2010). Önemli bir beyin dokusu olan korpus kallozum'un hacmi, ölçüm değerleri, ve şekil çeşitleri Alzheimer, şizofreni, disleksi, epilepsi ve MS hastalıkları ile doğrudan korelasyon içindedir.

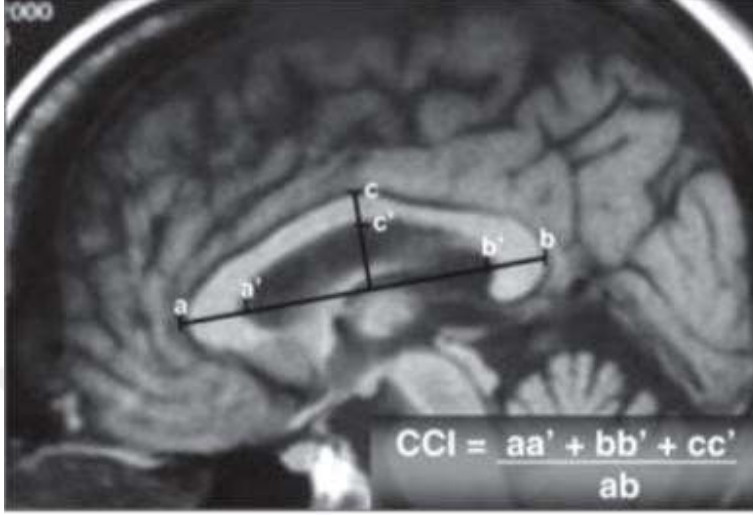
Şekil 1.2.'de sağlıklı birey ile MS hastası bireyin midsagittal beyin kesitleri yer almaktadır. Korpus kallozum dokusu 7 bölgeye ayrılarak incelemeler yapılmıştır. Hacim ise 11 ardışık merkezi sagittal kesitleri ile elde edilmiştir. MS hastası bireyin korpus kallozum dokusundaki atrofi açıkça görülmektedir (Llufriu vd. 2012).



**Şekil 1.2.** Sağlıklı birey ile MS hastası bireyin korpus kallozum dokularının karşılaştırması

Korpus kallozum indeksinin hesaplanması için öncelikle MR kesitleri içerisinde en uygun mid-sagittal T1W resminin seçilmesi gerekir. Ardından korpus kallozum üzerindeki en büyük anteroposterior çap üzerine düz bir çizgi çekilir. Bu çizginin tam ortasına çekilecek dik bir çizgi ile de cc' ölçümü yapılır. Şekil 1.3'de görüldüğü üzere aa' ve bb' ölçümleri de yapılarak en büyük anteroposterior çap değeri ile normalize edilir. Figürdeki aa' genu

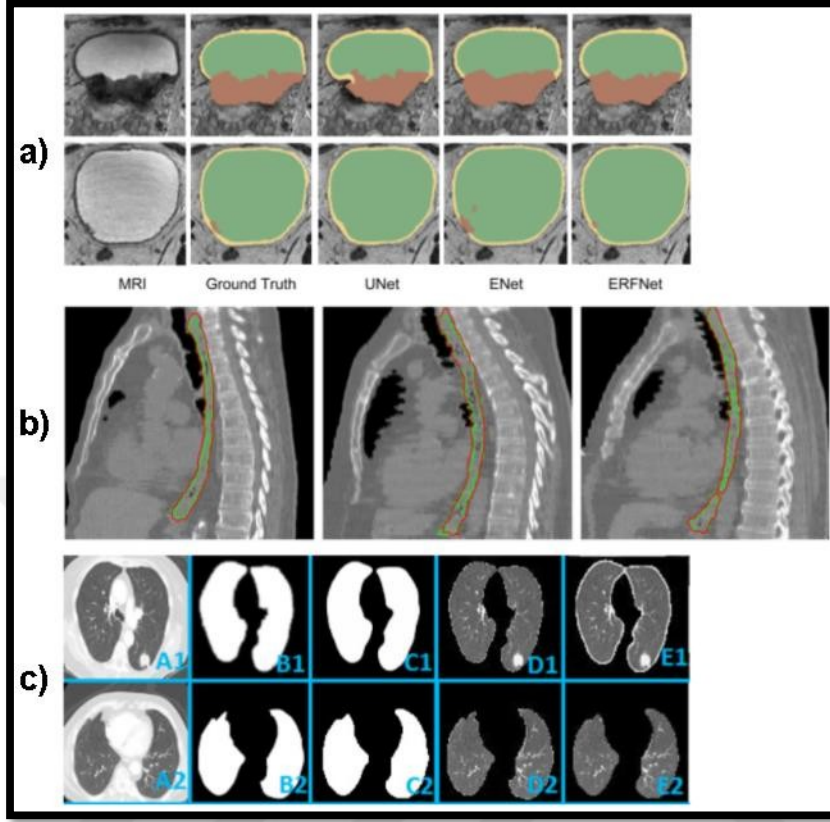
kalınlığını, bb' splenium kalınlığını, cc' korpus kalınlığını tanımlamaktadır. Denklemdaki ab ise korpus kallozum ön-arka toplam uzunluğunu ifade eder (Figueira vd. 2008).



**Şekil 1.3.** Korpus kallozum indeks denklemi ve morfolojik ölçüm gösterimi

Korpus kallozum indeksinin, hastalık takibi sürecinde belirli periyotlarda radyoloji uzmanı tarafından elde edilip incelenmesi gerekmektedir. Yapılan çalışma kapsamında, derin öğrenme metodlarından faydalanılarak korpus kallozum dokusunun otomatik olarak tespit edilmesi amaçlanmıştır.

Derin öğrenme modelleri, obje tanıma ve tespit konusundaki başarıları nedeniyle medikal uygulamalarda sıkça kullanılmaktadır (Krizhevsky vd. 2012). Medikal görüntüleme yöntemleri sonucu elde edilen resimler üzerinden sınıflandırma, tespit ve segmentasyon işlemleri ile hastalık veya doku teşhislerinde yardımcı sistemler geliştirilen bilmektedir. Şekil 1.4.'de derin öğrenme metodu kullanılarak medikal resimler üzerinde elde edilen segmentasyon sonuçları görülmektedir.

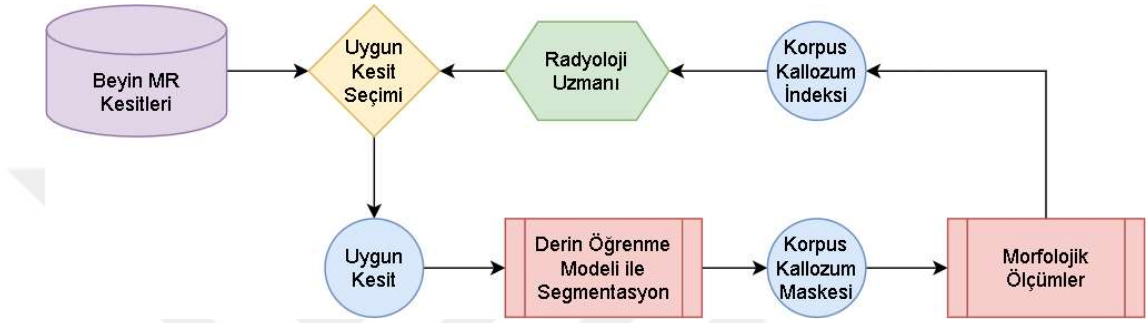


**Şekil 1.4.** Derin öğrenme metodları ile medikal resimler üzerinde elde edilen segmentasyon sonuçları. **a)** Mesane kanseri tespiti için yapılan çalışmada iç ve dış duvarlar yeşil ve sarı renkli pikseller ile tanımlanırken tümörlü bölgeler kahverengi pikseller ile segmente edilmiştir; **b)** Yemek borusu segmentasyonu çalışmasında yeşil renkli çizgiler referansı temsil ederken kırmızı çizgili hatlar kullanılan modelin çıktılarıdır; **c)** A orijinal resim; B akciğer dokusu etiketi, C çalışmada sunulan çözümün çıktısı, D etiket haritası ile oluşturulan segmentasyon, E U-net ile elde edilen segmentasyon çıktısı (Dolz vd. 2018; Fechter vd. 2017; Skourt vd. 2018)

Bu çalışmada da derin öğrenme metodlarından yararlanılarak, seçilen beyin MR kesiti üzerinde korpus kallozum dokusu segmente edilecektir. En uygun kesitin seçilmesi işlemi uzman radyolojist tarafından yapılacaktır. Segmentasyon işlemi ile korpus kallozum bir maske içerisine alınarak morfolojik ölçümlere hazır hale getirilecektir.

Derin öğrenme metodu ile elde edilen maske üzerinde en soldaki ve en sağdaki pikseller bulunacaktır. Belirlenen piksellerin tam ortasındaki değerin x değeri kullanılarak korpus kalınlığı elde edilecektir. Genu ve splenium kalınlıkları da orta noktaya doğru piksel değerleri takip edilerek bulunacaktır.

Korpus kallozum indeksi, morfolojik ölçümler ile elde edilen değerler kullanılarak belirlenecektir. Şekil 1.5’de genel olarak proje akış şeması gösterilmiştir.

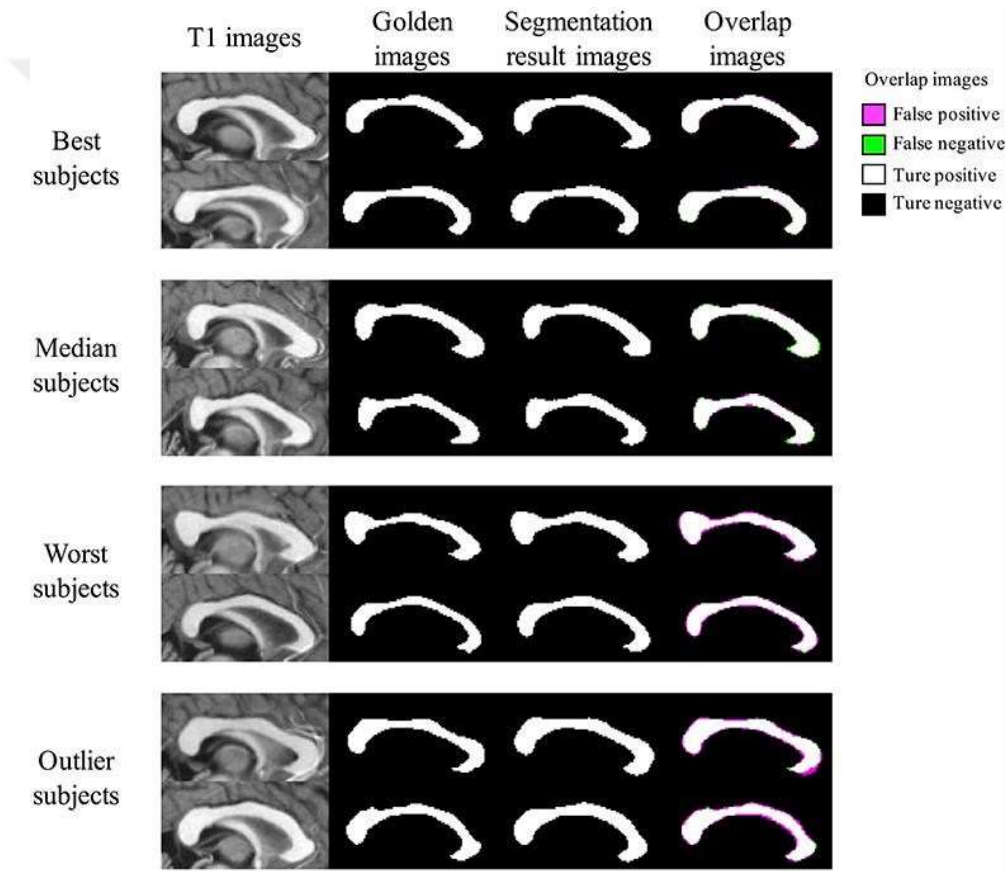


**Şekil 1.5.** Çalışma kapsamında sunulan çözüm diyagramı

Çalışma sonucunda radyoloji uzmanları MS hastalığı tespiti ve takibi konusunda önemli bir referans değeri olan KKİ otomatik olarak hızlı bir şekilde hesaplanabilir hale gelecektir. Sunulan sistemin özellikle zaman kazanımı açısından faydalı olması amaçlanmıştır.

## 2. KAYNAK TARAMASI

Park vd. (2018), çalışmasında medikal görüntüleme uygulamalarında başarılı olan seyrek temsil (sparse representation) ve çoklu-atlas oylama metotları kullanılarak geliştirilmiş olan Bayes çıkarımı kullanılmıştır. Sunulan Bayes çıkarımındaki başlıca elde edilen bilgi, çoklu-atlas oylaması tarafından üretilen olasılık haritasından elde edilmiştir. Olasılık haritası, hedef resimdeki korpus kallosum lokasyonunu ve şekil bilgisini içermektedir. Bayes çıkarımındaki olasılık değeri ise seyrek temsil hatasından üretilmiş olan gamma dağılımı fonksiyonundan elde edilmiştir. Şekil 2.1’de ilgili çalışma ile elde edilmiş segmentasyon çıktıları yer almaktadır.

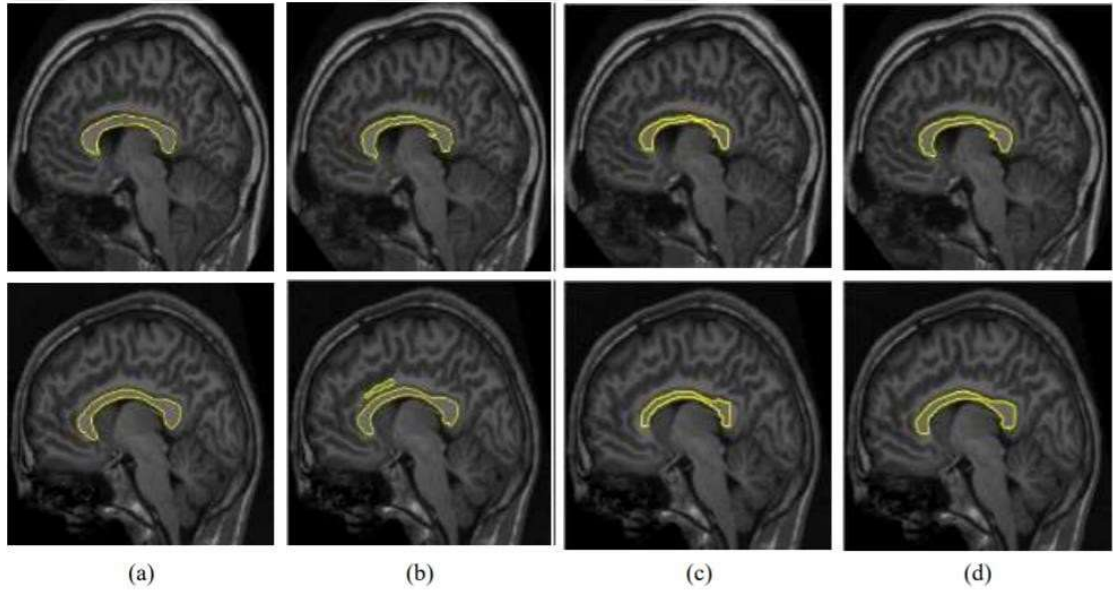


**Şekil 2.1.** Park vd. (2018) çalışmasına ait segmentasyon sonuçları

Korpus kallosum segmentasyonu ile ilgili bir diğer çalışmada, İçer (2013) Gauss karışım modeli ve bulanık C-Means kümeleme metotlarının başarısını ve performansını karşılaştırmıştır. Gerçek MR resimleri ve Simüle edilmiş resimler üzerinde yapılan çalışmada

Gauss karışım modeli 98.3% başarı oranı ile daha doğru sonuçlar verirken bulanık C-Means kümeleme metodu hız konusunda daha başarılı olmuştur.

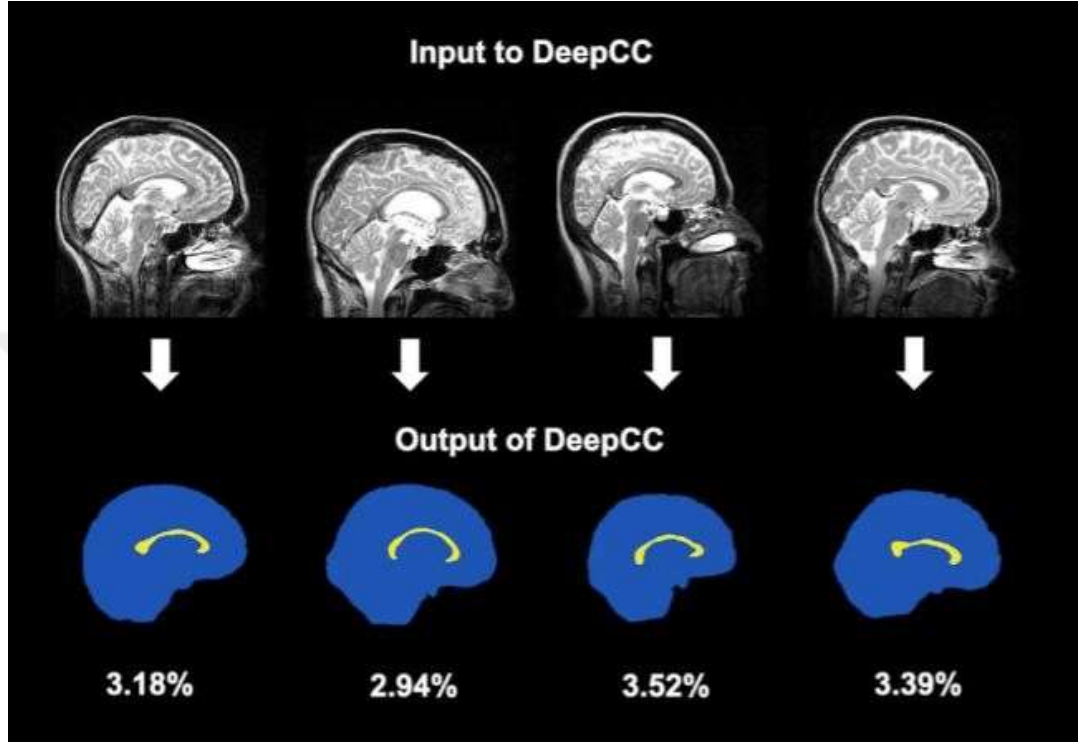
Li vd. (2017), Beyin MR resimleri üzerinde otomatik korpus kallozum segmentasyonu için yaptıkları çalışmalarında, 3 modülden oluşan bir çözüm sunmuşlardır. Birinci modülde, AMS (Adaptive Mean Shift) tekniği ile resimdeki tüm homojen alanlar kümelendirilmiştir. İkinci modülde ise kümelendirilmiş bölgelerin analizi, taslak eşleme, ve konum analizi sonucunda korpus kallozum bölgesi tespit edilerek ACI (Otomatik Korpus Kallozum Kontur Tanımlaması) işlemi tamamlanmıştır. Son olarak üçüncü modülde, GAC (Geometrik Aktif Kontur) modeli kullanılarak başlangıç konturu olan korpus kallozum bölgesi sınırları son haline geliştirilerek sonuç elde edilmiştir. Çalışmaya ait sonuçlar Şekil 2.2’de gösterilmiştir.



**Şekil 2.2.** Li vd. (2017)’nin sundukları çözümün sonuçları **a)** Etiketli resimler; **b)** Tipik GAC ile elde edilen sonuçlar; **c)** AMS-ACI tekniği ile alınan sonuçlar; **d)** AMS-ACI-GAC tekniği ile elde edilen sonuçlar (Li vd. 2017)

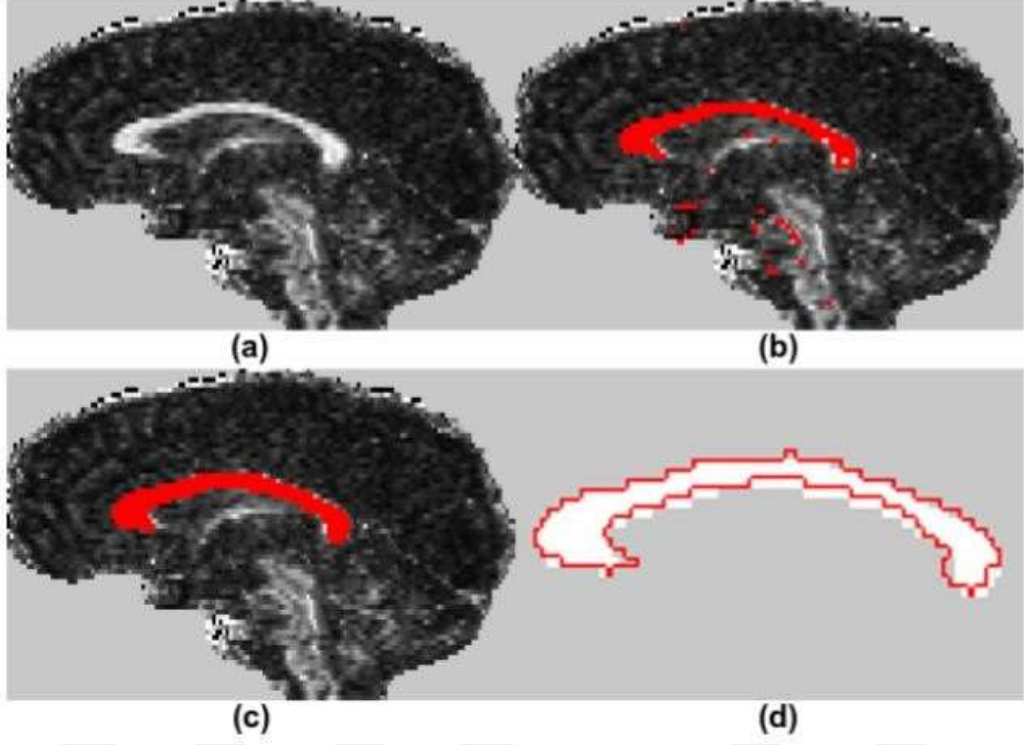
Michael vd. (2021), çalışmalarında intrakranyal ve korpus kallozum bölgelerini segmente ederek MS hastalığında nörodejeneratif belirteç olan korpus kallozum atrofisini otomatik olarak belirleyebilecek bir çözüm sunmuşlardır. DeepnCCA adını verdikleri algoritmada iki farklı bölgenin segmentasyonu için de U-Net mimarisi kullanılmışlardır.

Aynı mimariye sahip iki model ile yapılan segmentasyon sonuçlarında, beyin dokusu içerisinde korpus kallozum alanının oranı normalize edilerek bulunmuştur. Çalışmaya ait sonuçlar Şekil 2.3’de yer almaktadır.



**Şekil 2.3.** DeepnCCA algoritmasının çıktılarından örnekler. 4 farklı beyin kesiti üzerinde yapılan segmentasyon sonuçları ve korpus kallozum alanının intrakranyal bölgeye oranı (Michael vd. 2021)

Herrera vd. (2018) çalışmalarında Difüzyon Manyetik Rezonans Görüntüleme üzerinde Destek Vektör Makinesi (SVM) algoritmasını kullanarak geliştirdikleri piksel tabanlı sınıflandırıcı ile korpus kallozum dokusunu segmente etmişlerdir. Dengesiz piksel sınıflandırma sorununu çözmek için ise K-means algoritması tabanlı bir alt-örnekleme tekniği kullanmışlardır. Çalışma sonucunu değerlendirmek için hassaslık, özgüllük, ve Dice katsayısı metrikleri aracılığıyla STAPLE algoritmasını kullanarak nicel analiz gerçekleştirmişlerdir. Sundukları çalışmada 88% Dice Katsayısı medyan değerine ulaşılmıştır.



**Şekil 2.4.** Herrera vd. (2018) çalışmalarında sundukları piksel tabanlı sınıflandırıcının sonuçları. **a)** Kesirli Anizotropi haritası (Fractional Anisotropy); **b)** SVM ile sınıflandırma sonucu (Dice = 82%); **c)** Morfolojik işlemler ile elde edilen final segmentasyonu (Dice = 89%); **d)** STAPLE gümüş-standardı (koyu sınır çizgisi) ve piksel tabanlı sınıflandırma sonucunun (açık renkli maske) karşılaştırması

### 3. MATERYAL VE METOT

#### 3.1. Veri Seti

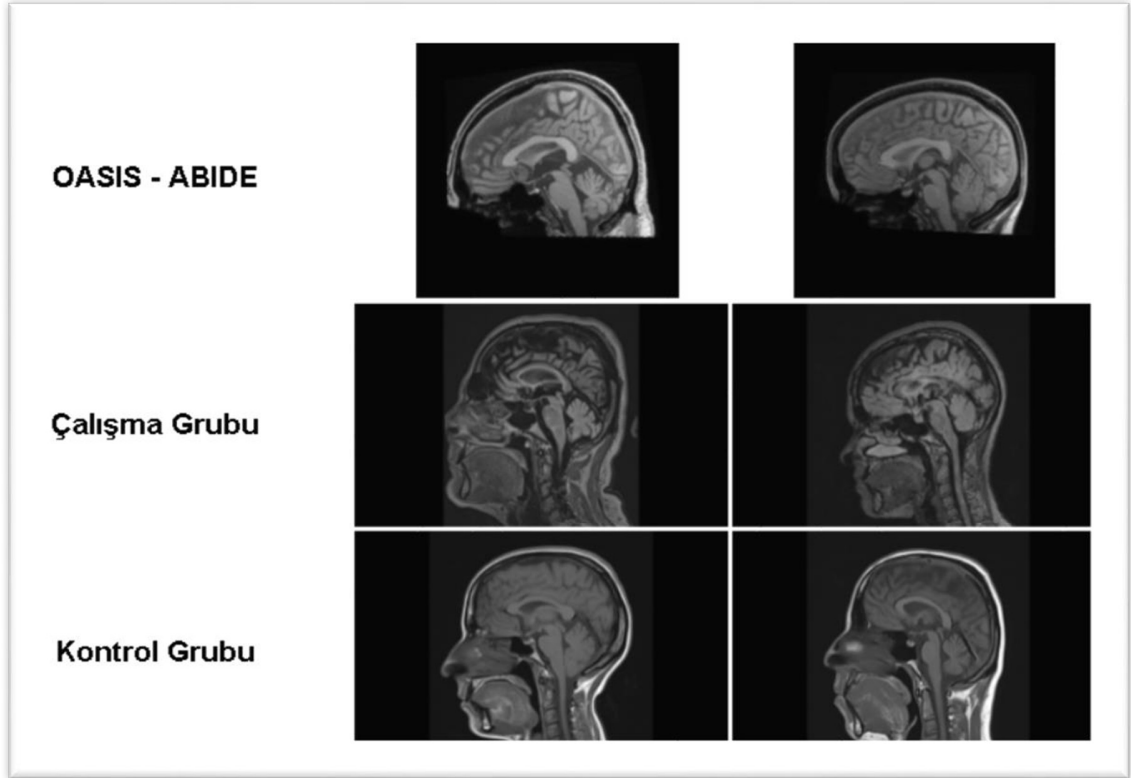
Segmentasyon modelinin eğitilebilmesi için etiketlenmiş örnek resimlere ihtiyaç duyulmaktadır. Bu etiketli örnek resimler veri seti olarak adlandırılmaktadır. Çalışma kapsamında öncelikle açık kaynaklı ABIDE ve OASIS beyin MR kesiti veri setlerinden yararlanılarak model eğitimleri gerçekleştirilmiştir (ABIDE Veri Seti; OASIS Veri Seti). Açık kaynaklı veri setleri ile yapılan eğitim sonucunda elde edilen model, Akdeniz Üniversitesi Radyoloji bölümünden elde edilen örnek resimler üzerinde test edilmiştir. Radyoloji bölümünden elde edilen örnekler eğitim setindeki örneklerden farklı protokollerde alındığından başarılı segmentasyon sonuçları alınamamıştır. İlgili durum ile Şekil 3.1.1’de gösterilmiştir.



**Şekil 3.1.1.** ABIDE ve OASIS setleri ile eğitilen modelin Radyoloji bölümü örnekleri üzerindeki segmentasyon sonuçları

Akdeniz üniversitesi Radyoloji bölümünden iki farklı protokolde alınmış veri setleri edinilmiştir. Bu veri setleri Çalışma grubu ve Kontrol grubu olarak adlandırılmıştır. İki grupta da ayrı ayrı 148 resim bulunmaktadır. Elde edilen sonuçların üzerine Radyoloji bölümünden elde edilen veri setleri üzerinde etiketleme çalışması yapılarak eğitime uygun hale getirilmiştir. ABIDE ve OASIS veri setleri ile eğitilmiş olan ön-eğitimli (pre-trained) model ağırlıkları ile yeni eğitimler başlatılmıştır. Yeni eğitim işlemlerinde etiketleme yapılmış olan Çalışma ve Kontrol grubu veri setleri kullanılmıştır. Bu eğitimlerin tamamlanmasıyla elde edilen modellerin segmentasyon çıktıları sonuçlar bölümünde anlatılacaktır.

Proje kapsamında üzerinde çalışılan OASIS ve ABIDE veri setleri ile Akdeniz Üniversitesi Radyoloji bölümünden alınan Çalışma ve Kontrol grubuna ait örnek MR resimleri Şekil 3.1.2’de gösterilmiştir.



**Şekil 3.1.2.** Tez çalışması kapsamında kullanılan veri setlerinden MRI örnekleri

Akdeniz Üniversitesi Radyoloji bölümünden alınan Beyin MR kesitlerini veri seti ile ilgili 2012-KAEK-20 kodlu etik kurulu raporu Ekler bölümünde yer almaktadır.

Çalışma ve Kontrol grubu veri setlerinin içinde yer alan resimler 3 Tesla ve 1,5 Tesla MRI cihazları ile alınmıştır. Kontrol grubu hastalarında rutin MRI görüntüleme; T1 sagittal, T2 transvers, T2 FLAIR transvers, SWI (Susceptibility Weighted Imaging); ve Diffüzyon ağırlıklı MRI sekansları alınmıştır. Ölçümler için kullanılacak olan T1 sagittal sekansta TE değeri 8,7 msn, TR değeri 490 msn, kesit kalınlığı ise 5 mm olarak belirlenmiştir. Çalışma grubu hastalarında (MS tanılı hastalar) rutin MRI görüntüleme; sagittal FLAIR, T1 transvers, T2 transvers, T2-FLAIR transvers, SWI ve Diffüzyon ağırlıklı MRI sekansları alınmaktadır. Ölçümler için kullanılacak olan sagittal FLAIR sekansta TE değeri 336 msn, TR değeri 5000 msn, kesit kalınlığı ise 1,4 mm olarak belirlenmiştir.

### 3.2. Python Programlama Dili

Proje kapsamında geliştirilecek olan kod blokları için Python yazılım dili kullanılacaktır. Derin öğrenme kütüphaneleri ile uyumlu olan Python dili, gelişmiş bir kömüniteye sahiptir. Proje kapsamında kullanılması planlanan Tensorflow ve Keras kütüphaneleri ile Python dili kullanılarak çalışılabilmektedir. Ayrıca belirtilen kütüphaneler ile model oluşturma, eğitme ve değerlendirme adımlarında yararlanılabilecek kaynaklar mevcuttur (Python).

Versiyon seçiminde ise Python dilinin en son versiyonu ve kullanılacak kütüphanelerin güncel versiyonları ile uyumlu olan “3.7.12” versiyonu tercih edilmiştir.

### 3.3. Pandas Kütüphanesi

Pandas kütüphanesi Python programlama dili için geliştirilmiş olan açık kaynaklı bir veri analizi aracıdır. Kütüphanenin geliştirilme amacı yüksek seviye kullanım dili ile temel bir veri analiz aracı olmasıdır (Pandas). Pandas modülü ile ilgili ana özellikler aşağıda anlatılmıştır:

- Entegre edilmiş indeksleme ile DataFrame objeleri üzerinde hızlı ve verimli veri manipülasyonuna imkan sağlar.
- CSV ve text dosyaları, Microsoft Excel, SQL veritabanları, ve HDF5 formatı gibi veri yapıları ve formatları arasında veri okuma ve yazma yapabilmektedir.
- Kayıp veriler için otomatik veri atama özelliğine sahiptir.
- Veri setleri üzerinde esnek yeniden şekillendirme işlemleri yapabilmektedir.
- Büyük veri setleri üzerinde kesme, alt küme oluşturma, birleştirme, çoklu filtreleme yapma kabiliyetlerine sahiptir.

### 3.4. Google Colaboratory Platformu

Google Colaboratory, tarayıcı üzerinde Python dosyalarını çalıştırmaya ve ücretsiz GPU servisi aracılığıyla derin öğrenme modellerini eğitmeye imkan sunan bir platformdur. Platform üzerinde derin öğrenme kütüphaneleri için gereken konfigürasyonlar hazır olarak gelmektedir. Ayrıca yapılan çalışmalar farklı kullanıcılar ile paylaşılabilmekte ve birlikte çalışılabilmektedir (Google Colaboratory).

Çalışma kapsamında veri seti oluşturma, serileştirme, zenginleştirme gibi ön hazırlık (preprocessing) işlemleri ve model eğitimi, test sonuçlarının alınması işlemleri Google Colaboratory platformu üzerinde gerçekleştirilmiştir.

### 3.5. Tensorflow ve Keras Derin Öğrenme Kütüphaneleri

Tensorflow büyük çaplı ve heterojen ortamlarda çalışabilen bir makine öğrenimi sistemidir. Tensorflow, işlemleri, paylaşımlı durumu, ve durumu manipüle eden işlemleri, veri akışı grafikleri ile temsil eder. Kütüphane, özellikle derin sinir ağlarının eğitimi ve kullanımına sunduğu güçlü destek ile geliştiricilere çeşitli uygulama alanlarını mümkün kılmaktadır (Abadi).

Tensorflow kütüphanesi, mobil ve IoT cihazları için TensorFlow Lite, tarayıcılar için TensorFlow.js, üretim seviyesindeki makine öğrenimi uygulamaları için ise TensorFlow Extended modüllerini sunmaktadır.

Tensorflow kütüphanesi ile gerçekleştirilen uygulama örnekleri aşağıda sıralanmıştır:

- Airbus firması TensorFlow yardımı ile uydu görüntülerinden anormal durum tespiti modelleri ile değerli bilgi çıkarımı yaparak müşterilerine iletmektedir (Airbus TensorFlow Project).
- Texas Instruments İşlemci SDK'ı TensorFlow Lite modülünü entegre ederek makine öğrenimi modellerinin işlemciler üzerinde çalıştırılabilmesini sağlamaktadır (Texas Instruments).
- InSpace video konferans platformu TensorFlow.js yardımı ile mesajlaşma filtreleri oluşturarak güvenli iletişim sağlamaktadır (Inspace TensorFlow Project).

Keras sinir ağı kütüphanesi Python ile yazılmış yüksek seviye, kullanımı basit bir modüldür. TensorFlow altyapısı üzerine düzenleyici bir modül olarak kurulmuştur (Keras ve TensorFlow). Proje kapsamında eğitim işlemleri için modelin tanımlanması aşamasında Keras kütüphanesinde tanımlanmış olan Giriş (Input), Birleşim (merge), 2 boyutlu evrişim (Conv2D), 2 boyutlu maksimum havuzlama (MaxPooling2D), 2 boyutlu yukarı örnekleme

(UpSampling2D), Seyreltme (Dropout), Tensor Birleştirme (concatenate) katmanları kullanılmıştır.

Ayrıca model katsayılarının normalize edilmesinde düzenleyici olarak BatchNormalization metodu kullanılmıştır. Optimizasyon algoritması olarak “Adam” algoritması tercih edilmiştir. Seçilen bu metodlar Keras kütüphanesi yardımı ile model için tanımlanmıştır.

### 3.6. Numpy Kütüphanesi

NumPy kütüphanesi, Python için temel bilimsel işlem paketidir. Çok-boyutlu dizi objeleri, çeşitli türev objeleri (maskelenmiş diziler ve matrisler gibi), diziler için birbirinden farklı hızlı işlem rutinlerini barındırmaktadır. Hızlı işlem rutinleri içerisinde, ayrık Fourier dönüşümü, şekil manipülasyonu, temel lineer cebir, temel istatistiksel işlemler, rastgele simülasyon gibi işlemleri barındırmaktadır (NumPy).

### 3.7. OpenCV Kütüphanesi

OpenCV (Open Source Computer Vision Library) açık kaynaklı bir bilgisayar görü ve makine öğrenimi yazılım kütüphanesidir. Bilgisayar görü uygulamaları için temel bir altyapı sağlamak ve makine algısı için gereken işlemleri hızlandırmak için geliştirilmiştir. BSD lisanslı bir ürün olduğundan dolayı işletmeler tarafından kullanılması ve modifiye edilmesi kolaydır.

Görüntü işleme ve bilgisayar görü alanında büyük kolaylıklar sağlayan OpenCV kütüphanesi, Klasik ve yeni geliştirilmiş olan algoritmalar da dahil olmak üzere 2500’den fazla optimize edilmiş algoritmayı bünyesinde barındırmaktadır. Bu algoritmalar sayesinde yüz tespiti ve tanıma, obje tanıma ve ayırt etme, insan hareketlerini sınıflandırma, kamera hareketlerini takip etme, hareketli objelerin takibi, objelerin 3D modelinin çıkarılması, 3D nokta bulutlarının çıkarılması, yüksek çözünürlüklü resim üretimi için birleştirme vb. gibi uygulamalar gerçekleştirilebilmektedir (OpenCV).

### 3.8. Matplotlib Görselleştirme kütüphanesi

Matplotlib kütüphanesi Python programlama dili için geliştirilmiş bir görselleştirme aracıdır. Oldukça kapsayıcı olan Matplotlib kütüphanesi yardımı ile statik, animasyon içeren, ve etkileşimli görselleştirmeler oluşturmak mümkündür (Matplotlib).

Kütüphanenin sunduğu başlıca avantajlar şunlardır:

- Yayın kalitesinde grafikler oluşturulabilir.
- Yakınlaştırma ve güncelleme yapılabilen etkileşimli figürler oluşturulabilir.
- Görsel stil ve arayüz modifiye edilebilir.
- Birçok farklı dosya formatında çıktı alınabilir.
- Grafik arayüzlerine entegre edilebilir.

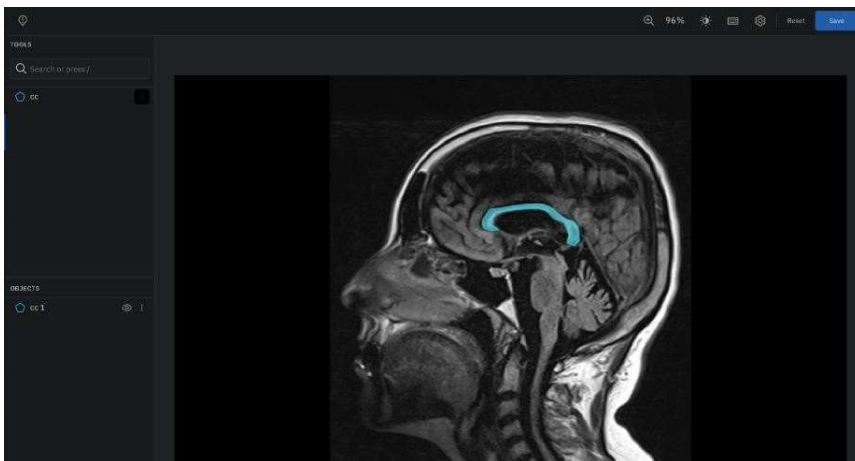
### 3.9. Scikit-learn Kütüphanesi

Scikit-learn Python modülü orta seviye gözetimli ve gözetimsiz problemler için en yeni makine öğrenimi algoritmalarını entegre etmektedir. Scikit-learn kütüphanesi, genel amaçlı ve yüksek seviye bir dilde kullanım sunmaktadır. Kullanım kolaylığı, performans, dokümantasyon, ve API sürekliliği önemli bulunarak geliştirilmiştir. BSD lisansına sahip olan modül hem akademik hem de ticari kullanıma uygundur (Pedregosa vd. 2012).

### 3.10. Labelbox Platformu

Labelbox, Yapay zeka çalışma takımları için geliştirilmiş veri merkezli altyapı platformudur. Platform, Yapay zeka takımlarının minimum insan gözetimi ile aynı anda aynı proje üzerinde birlikte çalışarak eğitim verisi oluşturmalarını ve model performansını iyileştirmelerini sağlamaktadır (Labelbox).

Labelbox platformunun sunduğu etiketleme otomasyon teknikleri sayesinde model eğitimi için gerekli olan etiketli veri seti hızlı bir şekilde ve ücretsiz olarak hazırlanabilmektedir. Şekil 3.10.1’de platform içerisinden görsel paylaşılmıştır.



**Şekil 3.10.1.** Labelbox platformunda korpus kallozum etiketi. Turkuaz renkli maske ile korpus kallozum bölgesi gösterilmiştir

### 3.11. U-Net Mimarisi

U-Net, hızlı ve kesin resim segmentasyonu için geliştirilmiş olan evrimsel ağ mimarisidir. ISBI elektron mikroskopu görüntülerinde nöron yapıları segmentasyonu yarışmasında önceki en iyi metod olan Kayan-Pencere Evrimsel Ağ metodundan daha iyi performans göstermiştir. Ayrıca 2015 yılında Biyomedikal görüntüleme alanındaki ISBI yarışmalarını kazanmıştır (ISBI Challenge; Cell Tracking Challenge).

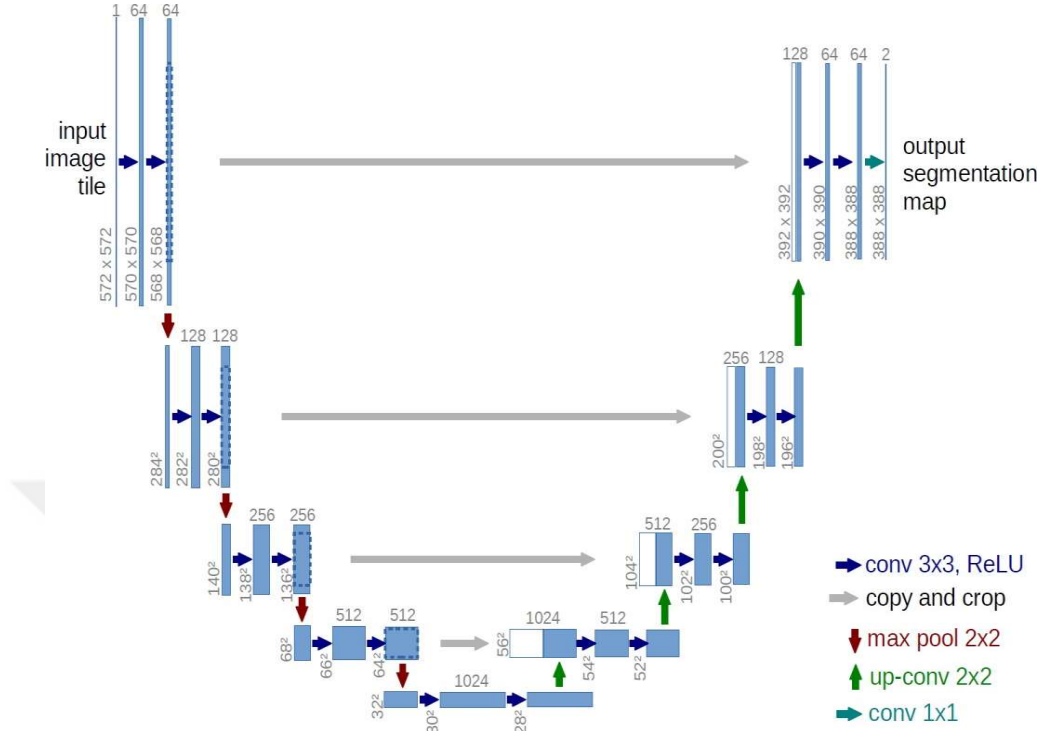
U-Net mimarisi, daraltma bölümü ile başlayıp genişletme bölümü ile sonlanmaktadır. Daraltma bölümü tipik bir konvolüsyon ağı olarak oluşturulmuştur. Başlangıç bölümündeki her adımda aşağıda verilen işlemler 2 birim adımlar (stride) ile uygulanır:

- 2 adet 3x3 konvolüsyon (Dolgu uygulaması olmadan)
- ReLU: Rectifier Linear Unit
- 2x2 maksimum pooling

Her alt örnekleme aşamasında öznitelik kanal sayısı 2 katına çıkarılmaktadır. Mimarinin ikinci aşaması olan genişletme bölümünde yukarı örnekleme yapılmaktadır. Genişletme bölümündeki her adımda aşağıdaki işlemler uygulanır:

- Öznitelik haritasının yukarı örnekleme
- 2x2 yukarı konvolüsyon (up convolution)
- 3 adet 3x3 konvolüsyon
- ReLU: Rectifier Linear Unit
- Her yukarı örnekleme işlemi sırasında öznitelik kanal sayısı yarıya düşmektedir.

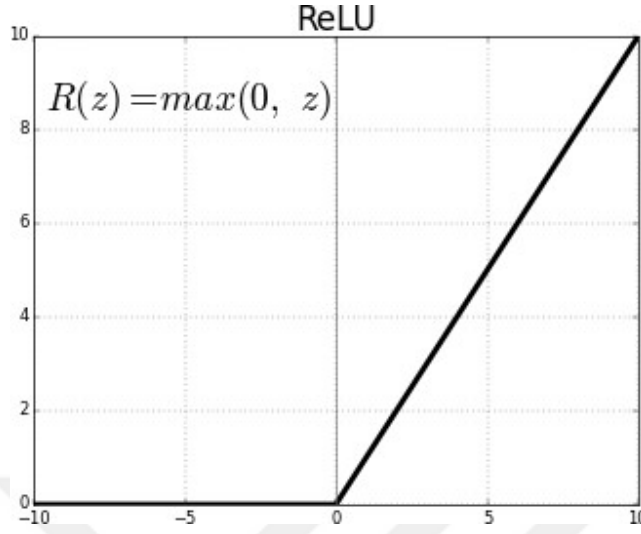
U-Net mimarisinin final katmanında ise 1x1 konvolüsyon işlemi ile 64-bileşenli öznitelik vektörü istenen sayıdaki sınıfa uygulanır.



**Şekil 3.11.1.** U-Net Mimarisi. Her mavi kutu çok kanallı öznetelik haritasını temsil etmektedir. Kanal sayısı kutunun üstünde bulunmaktadır. Boyut bilgisi sol alt köşelerde bulunmaktadır. Beyaz kutular kopya öznetelik haritalarıdır. Farklı oklar ile uygulanan işlemler belirtilmiştir (Ronneberger vd. 2015)

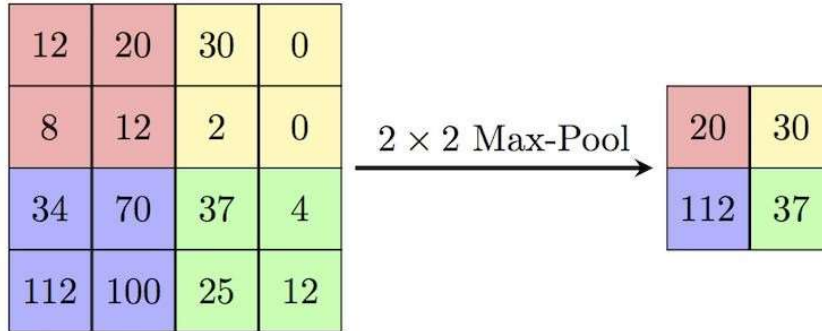
Şekil 3.11.1.'de U-Net mimarisi görselleştirilmiştir. Mimaride kullanılan operasyonlar aşağıda açıklanmıştır:

- **Conv 3x3, ReLU:** 3x3 maske ile gerçekleştirilen konvolüsyon işlemidir. Maskele Rectifier Linear Unit aktivasyon fonksiyonu ile yapılmaktadır. Bu fonksiyon girişe gelen değeri pozitif ise aynı şekilde çıkışa vermektedir. Girişin negatif olması durumunda ise çıkışta 0 değeri vermektedir. ReLU cevap grafiği Şekil 3.11.2'de gösterilmiştir.



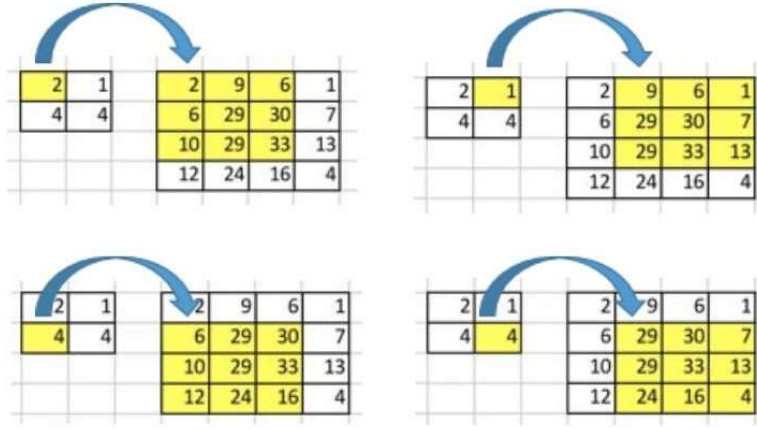
Şekil 3.11.2. Rectifier Linear Unit fonksiyonu cevap grafiği (ReLU)

- **Copy and crop:** Öznitelik haritasının kopyalanıp kırılması işlemidir.
- **Max pool 2x2:** 2x2 maske ile uygulanan en büyük değerin çıktı olarak alındığı fonksiyondur.



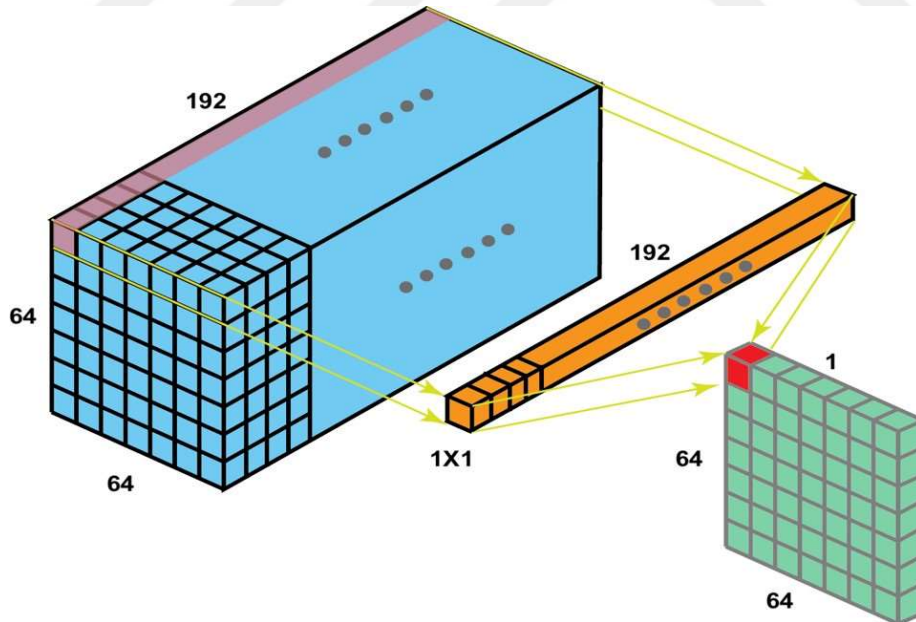
Şekil 3.11.3. 2x2 filtre ile gerçekleştirilen Max-Pool işlemi

- **Up-conv 2x2:** Düşük çözünürlükten yüksek çözünürlüğe geçiş için kullanılır. Eğitimde öğrenilen parametreler ile piksel değerinin çözünürlüğü artırılır. 2x2 maskeler ile gerçekleştirilir. Şekil 3.11.4.'de yukarı evrişim işlemine ait örnek yer almaktadır.



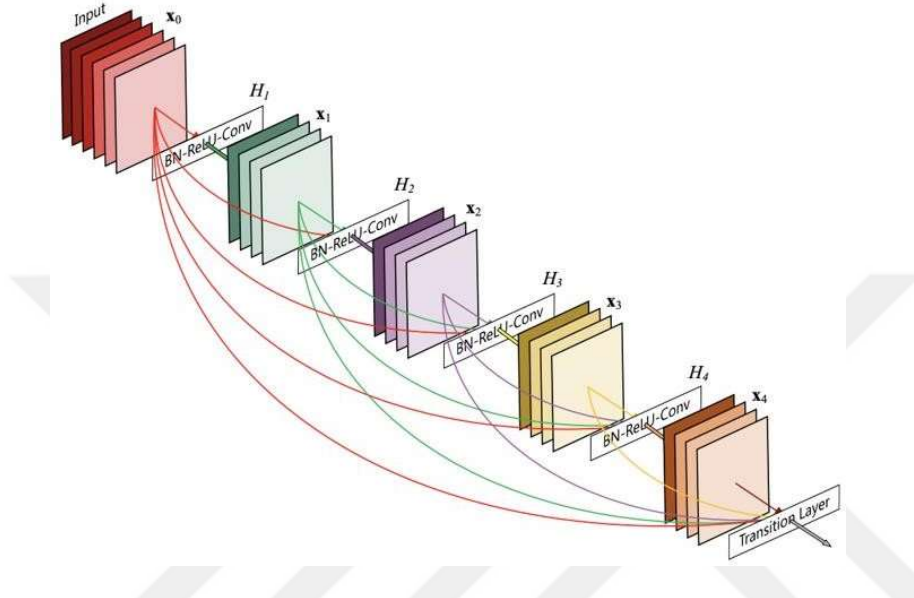
Şekil 3.11.4. 3x3 filtre ile gerçekleştirilen yukarı evrişim (Up-Convolution) işlemi

- **Conv 1x1:** 1x1 maske ile uygulanan konvolüsyon işlemidir.

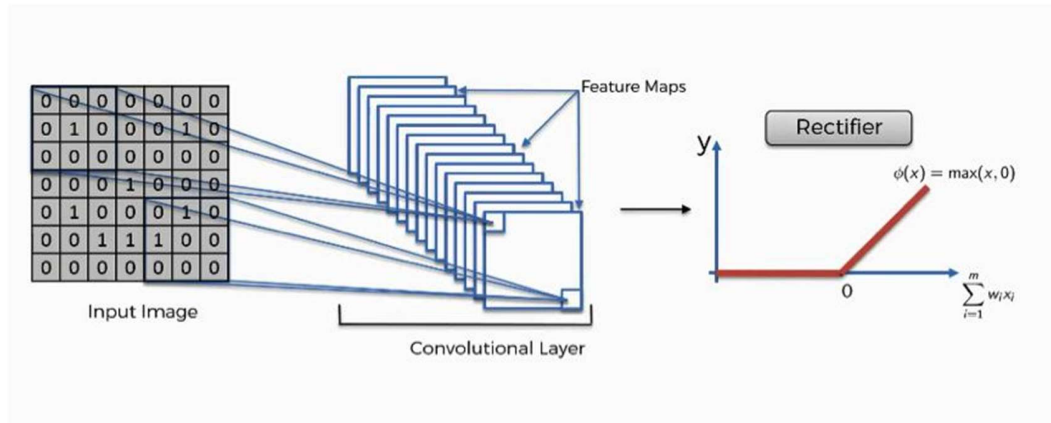


Şekil 3.11.5. 64x64x192 boyutlu öznetelik haritasına uygulanan 1x1 evrişim işlemi

- **Concatenated Skip Connection:** Elde edilen öznetelik haritalarının aktarılan katmanlarda tekrar kullanılabilmesini sağlayan işlemidir. Ağ yapısı içerisindeki önceki katmanlardaki bilginin aktarılmasını ve ilerleyen katmanlarda da etkili olmasını sağlar. Bu işleme ait görselleştirme örneği Şekil 3.11.6.'da yer almaktadır.



Şekil 3.11.6. 5 katmanlı öznetelik aktarım örneği. Her katman, önceki öznetelik haritalarını girdi olarak almaktadır (Concatenated Skip Connection)



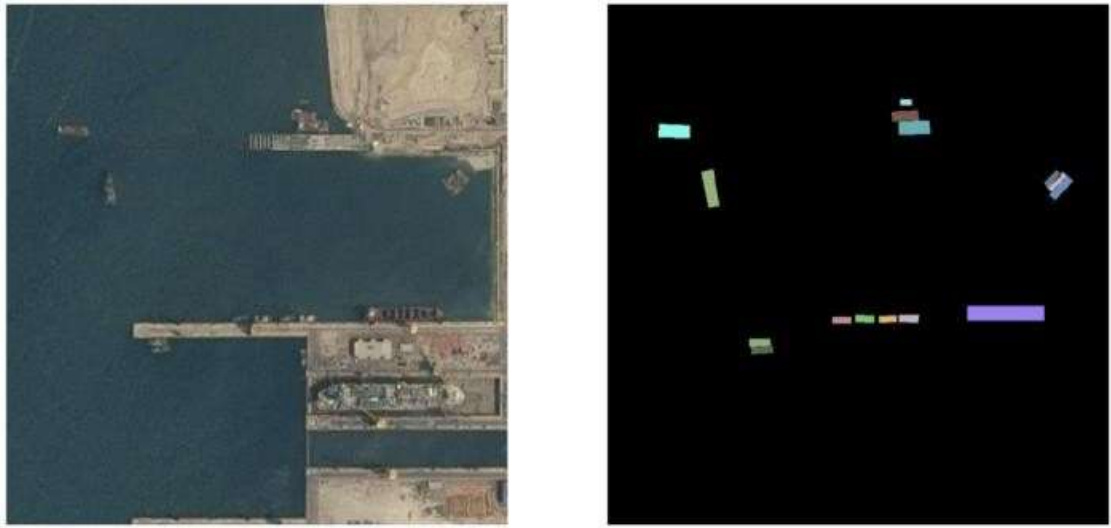
Şekil 3.11.7. Evrişim katmanı ile ilgili görselleştirme (Evrişim Katmanı)

Şekil 3.11.7’de evrişim katmamına ait işlem görselleştirilmiştir. Giriş resmine evrişim katmanındaki aktivasyon fonksiyonu uygulanarak öznitelik haritaları elde edilmektedir.

### 3.12. Segmentasyon Metrikleri

Model eğitiminde önemli noktalardan biri de model başarısını ölçmek için kullanılan metrik değerlerdir. Farklı uygulamalar için kullanılan metrikler de model mimarilerinde olduğu gibi değişim göstermektedir. Projede gerçekleştirilen işlem segmentasyon olduğundan dolayı bu uygulama alanında kullanılan metrikler incelenmiştir. Aşağıda segmentasyon başarı metrikleri olan Piksel Doğruluğu (Pixel Accuracy), Birleşim üzerindeki kesişim (Intersection Over Union - Jaccard Index), ve Dice Katsayısı (F1 Skoru) anlatılmıştır (Semantic Segmentation Metrics).

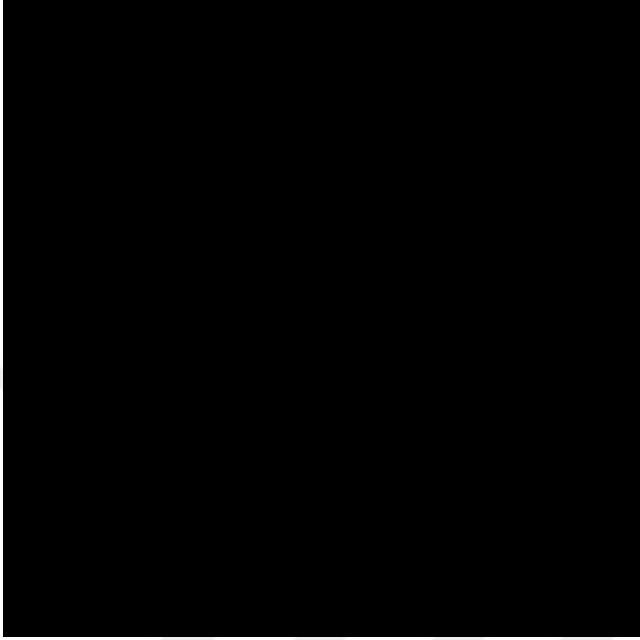
Piksel doğruluğu metriği anlaşılması en basit segmentasyon metriğidir. Resim üzerinde doğru segmente edilen piksellerin oranı ile temsil edilir. İlk bakışta oldukça kullanılabilir gözüken metrik oldukça aldatıcı sonuçlar verebilmektedir. Bu durumun anlaşılması için Şekil 3.12.1’de görülen resim üzerine üretilen segmentasyon sonucu incelenecektir.



**Şekil 3.12.1.** Pixel Accuracy metriği için segmentasyon örneği. Sol tarafta orijinal resim ve sağ tarafta etiketlenmiş maske resmi yer almaktadır (Pixel Accuracy)

Şekil 3.12.2’de yer alan segmentasyon sonucu için Piksel Doğruluğu metriği 95% başarı sonucunu vermiştir. Gemi etiketleri haricinde kalan büyük alan doğru segmente

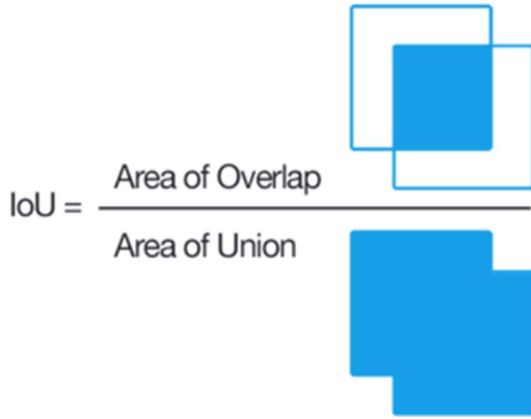
edildiğinden başarı puanı yüksek çıkmıştır. Uygulamaya göre metrik seçiminin ne kadar önemli olduğu bu örnek üzerinden anlaşılabılır.



**Şekil 3.12.2.** Şekil 3.12.1 için çıkan segmentasyon sonucu

Örnekte görülen problem sınıf dengesizliği (class imbalance) olarak adlandırılmaktadır. Sınıf dengesizliği, herhangi bir sınıfın resimde oldukça baskın şekilde yer alması ile oluşur. Bu durum gerçek hayat problemlerinde oldukça yaygın olduğundan göz ardı edilebilir değildir. Aşağıda anlatılan diğer segmentasyon metrikleri ile bu problem aşılabilmektedir.

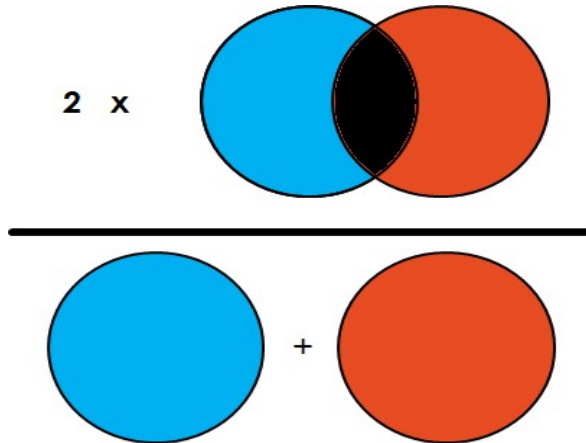
Birleşim üzerindeki kesişim metriği segmentasyon uygulama en yaygın kullanılan metriklerden biridir. Oldukça verimli olan IoU metriği için kullanılan hesaplama yöntemi Şekil 3.12.3’de görselleştirilmiştir.



**Şekil 3.12.3.** Birleşim üzerindeki kesişim metriği için kullanılan hesaplama yönteminin görseli (Jaccard Index)

Şekil 3.12.3’de görüldüğü üzere, IoU metriği, segmentasyon çıktısı ile etiket alanının kesişiminin segmentasyon çıktısı ve etiket alanı toplamına bölünmesi ile elde edilir. Metrik, 0 ile 1 (0-100%) arasında bir değer alabilir. 0 değeri başarısız segmentasyonu temsil ederken 1 değeri tam başarılı segmentasyonu ifade eder.

Dice Katsayısı, IoU metriği ile oldukça benzerdir. F1 skoru olarak da anılan metrik, segmentasyon çıktısı ile etiketli resmin kesişim alanının 2 katının, segmentasyon çıktısı ve etiketli resim alanlarının toplamına bölünmesi ile elde edilir. Şekil 3.12.4’de hesaplama yöntemi görselleştirilmiştir.



**Şekil 3.12.4.** Dice Katsayısı hesaplama yöntemi görseli

F1 skoru ortalama başarı performansına odaklanırken, IoU skoru en kötü duruma göre puanlamasını şekillendirmektedir. Bu farkın anlaşılması için şu örnek kullanılabilir. Bir uygulamada A sınıflandırıcısı ile elde edilen çıktılarda genel olarak kesişimler B sınıflandırıcısı ile elde edilenlerden daha iyi yani başarı değeri daha yüksek olsun. Ancak A sınıflandırıcısı ile elde edilen bazı çıktılarda kesişim değerlerinin oldukça kötü olduğunu kabul edelim. Bu durumda F1 skoru metriği ile A sınıflandırıcısı tercih edilecek iken IoU metriği ile B sınıflandırıcısı tercih edilecektir (Dice Skoru ve IoU).



## 4. BULGULAR VE TARTIŞMA

### 4.1. Model Seçimi

Tez çalışması kapsamında ilk aşamada korpus kallozum bölgesinin kesit resmi üzerinde segmente edilmesi gerekmektedir. Bu işlemin gerçekleştirilebilmesi için korpus kallozum etiketli verisi ile eğitilecek olan bir derin öğrenme modeli kullanılacaktır. Modelin başarılı olmasında probleme uygun bir mimarinin seçilmesi kritik bir noktadır.

Resim segmentasyonu işlemi için geliştirilmiş olan mimarilerden bazıları aşağıda sıralanmıştır:

- U-Net
- FastFCN —Fast Fully Convolutional Network
- Gated-SCNN
- DeepLab
- Mask R-CNN

Tez çalışması medikal bir alan olan radyoloji alanındaki bir probleme çözüm sunacaktır. Bundan dolayı medikal resimlerde başarılı olan model mimarileri araştırılmıştır. Segmentasyon işlemi için derin öğrenme mimarilerinden olan U-Net mimarisi seçilmiştir. Seçilen mimari, medikal görüntüleme alanındaki ISBI yarışmalarında en iyi performansı gösteren mimari olmuştur ( Arganda-Carreras vd. 2015; U-Net).

### 4.2. Veri Seti Oluşturma

Derin öğrenme mimarilerinden olan U-Net modelinin eğitimi için etiketli resimlere ihtiyaç duyulmaktadır. Etiketli veri setinin oluşturulması için Labelbox platformu kullanılmıştır. Platform üzerinde sunulan otomatik etiketleme araçları sayesinde orjinal resim üzerinde korpus kallozum bölgesi poligon kullanılarak maske içerisine alınır ve etiket oluşturulmuş olur. Çalışmanın yapılacağı örnek beyin kesiti resimleri için Akdeniz üniversitesi Radyoloji bölümünden iki farklı protokolda alınmış veri setleri edinilmiştir. Bu veri setleri Çalışma grubu ve Kontrol grubu olarak adlandırılmıştır. Her iki veri seti grubu için etiketleme işlemi gerçekleştirilmiştir.

Labelbox platformunda etiketleme işlemleri tamamlanınca çıktı olarak her resim için poligon maske değerlerini barındıran JSON formatında bir dosya elde edilmiştir. Dosyadaki değerler kullanılarak “tiff” formatındaki etiketli resimler elde edilir.

JSON formatındaki etiket dosyasındaki bilgilerin işlenerek “tiff” formatındaki etiketli resimlerin oluşturulması için Python programlama dili ile geliştirilmiş olan kod

kullanılmıştır. İlgili kodun algoritma adımları aşağıda gösterilmiştir. Ayrıca algoritmanın akış diyagramı Şekil 4.2.1’de görülmektedir. Sonuç olarak elde edilen veri seti görselleri Şekil 4.2.2’de yer almaktadır.

*Çalışma dosyasına OpenCV, JSON, NumPy kütüphanelerini ekle*

*LabelBox’den alınmış JSON dosyasını oku*

*JSON içerisindeki her etiket için:*

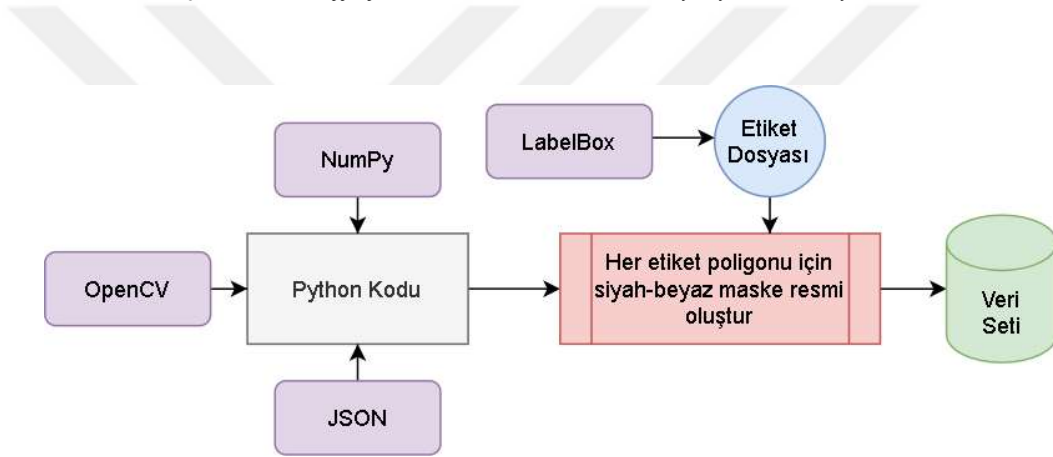
*Etiketin ait olduğu resmi OpenCV ile oku*

*Etiket içerisindeki poligon noktalarının x ve y değerlerini NumPy dizisine al*

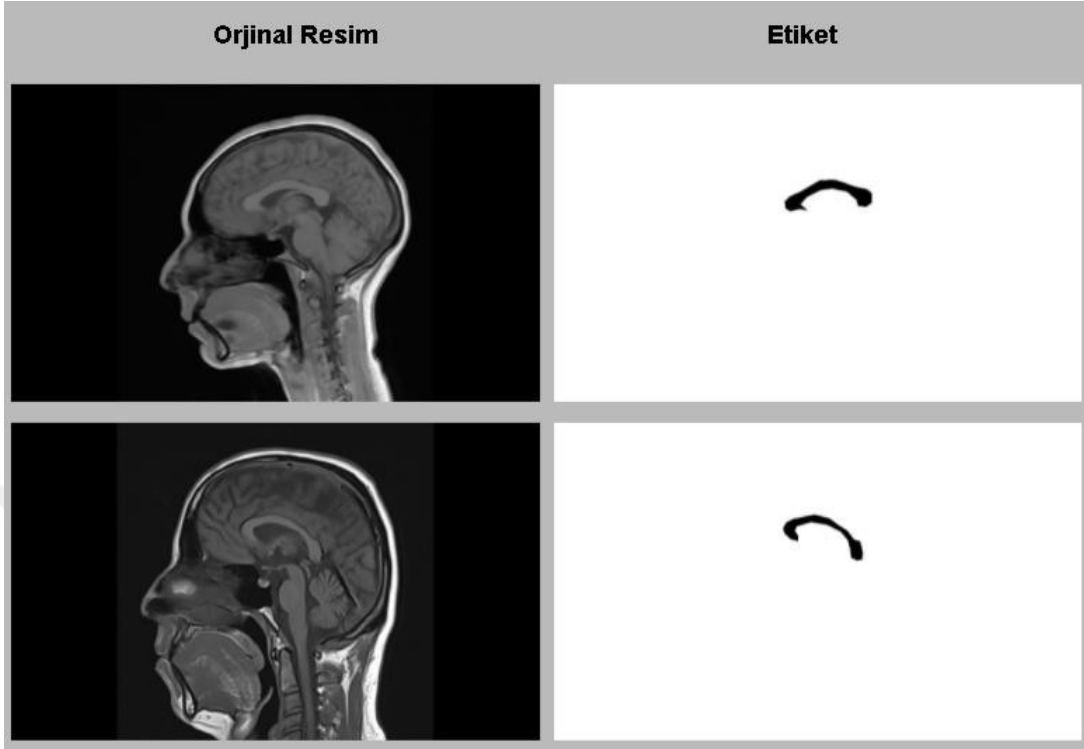
*NumPy ile etiketin ait olduğu resim boyutunda 255 değerli (beyaz) matris oluştur*

*OpenCV ile beyaz matris üzerinde, elde edilen poligonu 0 değeri (siyah) ile doldur*

*Sonuç resmini ‘tiff’ formatında belirtilen dosya yoluna kaydet*



**Şekil 4.2.1.** Etiket dosyasının işlenmesi ile ilgili akış diyagramı



**Şekil 4.2.2.** Labelbox platformundan alınan etiket verilerinin işlenmesi ile elde edilen veri seti görselleri

Veri setinin, eğitim aşamasında model girişine hazır hale getirilmesi için serileştirme işlemi ile “PKL” formatına çevrilmiştir. PKL dosyasının elde edilmesi için Python uyumlu Joblib modülü kullanılmıştır. İşlem aşamaları Şekil 4.2.3’de görselleştirilmiştir ve aşağıda Pseudo kodu yer almaktadır:

*NumPy, Pandas, Joblib, Math, ve PIL kütüphanelerini kod dosyasına ekle*

*Veri setindeki resimlerin dosya yolunu tanımla*

*Resim boyutunu 128 x 128 olarak ayarla*

*Girdi ve Çıktı resimleri için boş NumPy dizileri oluştur*

*Veri setindeki resimlerin ismini ve indeksini barındırmak için Pandas DataFrame’i oluştur*

*Veri seti dosya yolundaki her resim için:*

*‘jpg’ formatındaki orjinal resmi (girdi) oku*

*‘tiff’ formatındaki segmentasyon maskesini (çıktı) oku*

*Döngü indeksini ve okunan resmin ismini DataFrame’e aktar*

*PIL modülü ile orjinal resmi aç ve gri skalaya çevir. Ardından 128 x 128 hale boyutla*

*Manipüle edilen orjinal resmi NumPy dizisi olarak tanımla ve Girdi dizisine ekle  
PIL modülü ile segmentasyon resmini aç ve gri skalaya çevir.*

*Ardından 128 x 128 hale boyutla*

*Manipüle edilen segmentasyon resmini NumPy dizisi olarak tanımla ve Çıktı dizisine ekle*

*Oluşturulan DataFrame'i 'csv' olarak kaydet*

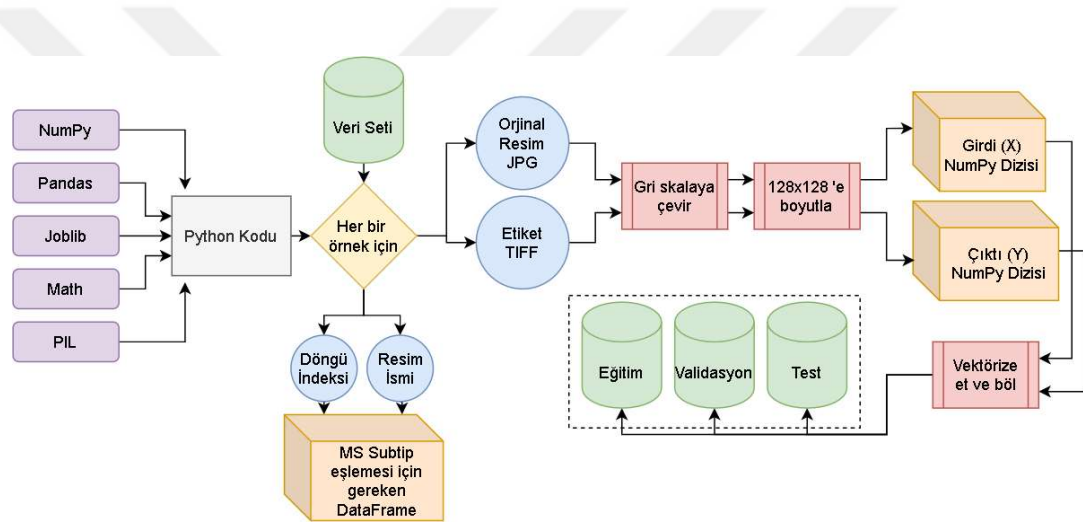
*Girdi ve Çıktı dizilerini vektör haline çevir*

*Eğitim yüzdesini 70% olarak belirle*

*Validasyon yüzdesini 10% olarak belirle*

*Girdi dizisini NumPy modülü ile rastgele karıştır*

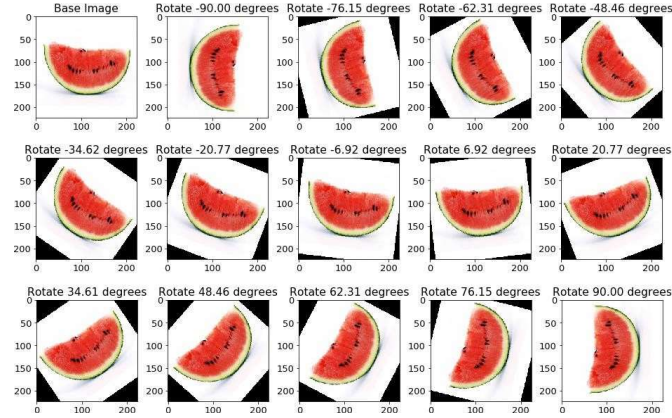
*Eğitim-Validasyon ve Test dizilerini ayrı ayrı 'pkl' formatında kaydet*



**Şekil 4.2.3.** Veri setinin model eğitime uygun hale dönüştürülmesi için uygulanan işlemlerin akış diyagramı

Veri seti üzerinde son olarak zenginleştirme işlemi gerçekleştirilmiştir. Bu işlem sayesinde veri setinde yer alan örnek sayısının çeşitlendirilmesi ve veri setinin daha kapsayıcı bir hale getirilmesi amaçlanmıştır.

Zenginleştirme teknikleri serileştirilmiş veri üzerinde uygulanmıştır. Zenginleştirme işlemi için veri setindeki girdi ve çıktı resimlerine rotasyon uygulanmıştır. Rotasyon sonucu oluşan yeni girdi ve çıktı veri setine eklenmiştir. Şekil 4.2.4'de rotasyon işlemi için örnek görselleştirme yer almaktadır.



Şekil 4.2.4. Rotasyon işlemi ile veri zenginleştirme örneği

### 4.3. Model Eğitimi

Model eğitimine başlamak için öncelikle Python kod bloğu içerisinde Tensorflow ve Keras kütüphaneleri yardımı ile seçilen U-Net mimarisi oluşturulmuştur. Oluşturulan model ile ilgili katman ve parametre bilgilerinin özeti Şekil 4.3.1’de yer almaktadır.

| Layer (type)                        | Output Shape          | Param # | Connected to                |
|-------------------------------------|-----------------------|---------|-----------------------------|
| input_1 (InputLayer)                | [(None, 128, 128, 1)] | 0       |                             |
| conv2d (Conv2D)                     | (None, 128, 128, 64)  | 1664    | input_1[0][0]               |
| conv2d_1 (Conv2D)                   | (None, 128, 128, 64)  | 102464  | conv2d[0][0]                |
| batch_normalization (BatchNormaliza | (None, 128, 128, 64)  | 256     | conv2d_1[0][0]              |
| max_pooling2d (MaxPooling2D)        | (None, 64, 64, 64)    | 0       | batch_normalization[0][0]   |
| conv2d_2 (Conv2D)                   | (None, 64, 64, 128)   | 204928  | max_pooling2d[0][0]         |
| conv2d_3 (Conv2D)                   | (None, 64, 64, 128)   | 409728  | conv2d_2[0][0]              |
| batch_normalization_1 (BatchNor     | (None, 64, 64, 128)   | 512     | conv2d_3[0][0]              |
| max_pooling2d_1 (MaxPooling2D)      | (None, 32, 32, 128)   | 0       | batch_normalization_1[0][0] |
| conv2d_4 (Conv2D)                   | (None, 32, 32, 256)   | 819456  | max_pooling2d_1[0][0]       |
| conv2d_5 (Conv2D)                   | (None, 32, 32, 256)   | 1638656 | conv2d_4[0][0]              |
| batch_normalization_2 (BatchNor     | (None, 32, 32, 256)   | 1024    | conv2d_5[0][0]              |
| max_pooling2d_2 (MaxPooling2D)      | (None, 16, 16, 256)   | 0       | batch_normalization_2[0][0] |
| conv2d_6 (Conv2D)                   | (None, 16, 16, 512)   | 3277312 | max_pooling2d_2[0][0]       |
| conv2d_7 (Conv2D)                   | (None, 16, 16, 512)   | 6554112 | conv2d_6[0][0]              |

Şekil 4.3.1. U-Net modelinin katmanları, çıkış boyutları, ve bağlantı noktaları

|                                 |                       |          |                                                      |
|---------------------------------|-----------------------|----------|------------------------------------------------------|
| batch_normalization_3 (BatchNor | (None, 16, 16, 512)   | 2048     | conv2d_7[0][0]                                       |
| max_pooling2d_3 (MaxPooling2D)  | (None, 8, 8, 512)     | 0        | batch_normalization_3[0][0]                          |
| conv2d_8 (Conv2D)               | (None, 8, 8, 1024)    | 13108224 | max_pooling2d_3[0][0]                                |
| conv2d_9 (Conv2D)               | (None, 8, 8, 1024)    | 26215424 | conv2d_8[0][0]                                       |
| batch_normalization_4 (BatchNor | (None, 8, 8, 1024)    | 4096     | conv2d_9[0][0]                                       |
| up_sampling2d (UpSampling2D)    | (None, 16, 16, 1024)  | 0        | batch_normalization_4[0][0]                          |
| concatenate (Concatenate)       | (None, 16, 16, 1536)  | 0        | up_sampling2d[0][0]<br>batch_normalization_3[0][0]   |
| dropout (Dropout)               | (None, 16, 16, 1536)  | 0        | concatenate[0][0]                                    |
| conv2d_10 (Conv2D)              | (None, 16, 16, 512)   | 19661312 | dropout[0][0]                                        |
| conv2d_11 (Conv2D)              | (None, 16, 16, 512)   | 6554112  | conv2d_10[0][0]                                      |
| up_sampling2d_1 (UpSampling2D)  | (None, 32, 32, 512)   | 0        | conv2d_11[0][0]                                      |
| concatenate_1 (Concatenate)     | (None, 32, 32, 768)   | 0        | up_sampling2d_1[0][0]<br>batch_normalization_2[0][0] |
| dropout_1 (Dropout)             | (None, 32, 32, 768)   | 0        | concatenate_1[0][0]                                  |
| conv2d_12 (Conv2D)              | (None, 32, 32, 256)   | 4915456  | dropout_1[0][0]                                      |
| conv2d_13 (Conv2D)              | (None, 32, 32, 256)   | 1638656  | conv2d_12[0][0]                                      |
| up_sampling2d_2 (UpSampling2D)  | (None, 64, 64, 256)   | 0        | conv2d_13[0][0]                                      |
| concatenate_2 (Concatenate)     | (None, 64, 64, 384)   | 0        | up_sampling2d_2[0][0]<br>batch_normalization_1[0][0] |
| dropout_2 (Dropout)             | (None, 64, 64, 384)   | 0        | concatenate_2[0][0]                                  |
| conv2d_14 (Conv2D)              | (None, 64, 64, 128)   | 1228928  | dropout_2[0][0]                                      |
| conv2d_15 (Conv2D)              | (None, 64, 64, 128)   | 409728   | conv2d_14[0][0]                                      |
| up_sampling2d_3 (UpSampling2D)  | (None, 128, 128, 128) | 0        | conv2d_15[0][0]                                      |
| concatenate_3 (Concatenate)     | (None, 128, 128, 192) | 0        | up_sampling2d_3[0][0]<br>batch_normalization[0][0]   |
| dropout_3 (Dropout)             | (None, 128, 128, 192) | 0        | concatenate_3[0][0]                                  |
| conv2d_16 (Conv2D)              | (None, 128, 128, 64)  | 307264   | dropout_3[0][0]                                      |
| conv2d_17 (Conv2D)              | (None, 128, 128, 64)  | 102464   | conv2d_16[0][0]                                      |
| conv2d_18 (Conv2D)              | (None, 128, 128, 1)   | 65       | conv2d_17[0][0]                                      |
| =====                           |                       |          |                                                      |
| Total params: 87,157,889        |                       |          |                                                      |
| Trainable params: 87,153,921    |                       |          |                                                      |
| Non-trainable params: 3,968     |                       |          |                                                      |

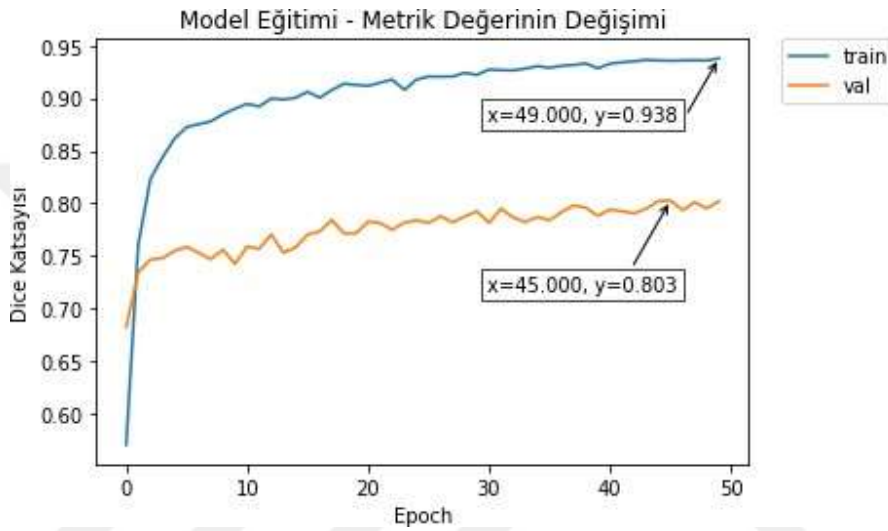
Şekil 4.3.1'in devamı

Ön hazırlık (preprocessing) işlemlerinin tamamlanmasıyla birlikte oluşturulan veri seti ile tanımlanan U-Net derin öğrenme modelinin eğitim aşamasına geçilmiştir. Eğitim esnasında kullanılan parametreler aşağıda anlatılmıştır:

- **smooth:** Dice katsayısı hesaplanırken hem çıktı hemde etiketli maske değerleri 0 olabilir. Bu durumda kullanılan denklem sonsuz değeri verdiğinden eğitim işleminde istenmeyen durumlar oluşur. Bu durumu engellemek için ‘smooth’ üst değişkeni kullanılır [43](Smooth Hiperparametresi). Denklemin hem pay hem de payda kısmına eklenerek normal durumlarda herhangi bir etkisinin olmaması sağlanır. Bu değişkene float türünde 1 değeri atanmıştır.
- **use\_dropout:** Seyreltme (Dropout) işlemi ile girdi setindeki değerlerden rastgele seçim yapılarak 0 ile değiştirilir. Değer atama işlemi, tanımlanacak olan “dropout\_rate” değerine bağlı bir sıklıkta her adımda uygulanır. Seyreltme işlemi sayesinde aşırı öğrenme (Overfitting) durumu önlenmiş olur. Tüm girdilerin toplamının değişmemesi için sıfır değeri atanmayan girdiler  $1 / (1 - \text{‘dropout\_rate’})$  ile ölçeklenir (Dropout Katmanı).
- **use\_regularizers:** Boolean türünde olan bu üst değişken düzenlileştiricilerin kullanılıp kullanılmamasını belirler. Model eğitimi için düzenlileştiriciler kullanılmıştır.
- **dropout\_rate:** Seyreltme işleminin hangi sıklıkta yapılacağını belirten üst değişkendir. Eğitim işlemi için 0.5 değeri atanmıştır.
- **number\_of\_epochs:** Öğrenme algoritmasının tüm eğitim seti üzerinde kaç defa çalışacağını belirten üst değişkendir (hyperparameter). Eğitim işlemi için 50 Epoch uygulanmıştır.
- **batch\_size:** Dahili model parametrelerini güncellemeden önce üzerinde çalışılacak örnek sayısını belirtir. Kullanılan donanımın işlem gücüne bağlı olarak belirlenebilir. Eğitim işleminde ‘batch\_size’ üst değişkeni 32 olarak belirlenmiştir (Batch Size).
- **kernel\_size:** Filtre maskesinin piksel türünden genişlik ve yükseklik bilgilerini içeren boyut üst değişkenidir. Eğitim işlemi için 5x5 filtre maskeleri tercih edilmiştir.
- **initial\_volume\_size:** Evrişim katmanındaki çıkış filtrelerinin sayısıdır (Conv2D Katmanı). 64 değeri atanmıştır.

Bu proje kapsamında genel başarı değeri daha önemli bulunduğundan dolayı segmentasyon modellerinin başarı ölçümleri için Dice katsayısı metriği seçilmiştir.

Model eğitimi işlemi boyunca seçilen başarı metriğinin grafiği ve sonuç değerleri Şekil 4.3.2’de görülmektedir. Şekildeki grafikte mavi çizgi eğitim seti üzerinde elde edilen Dice katsayısını, turuncu çizgi ise validasyon seti üzerinde elde edilen Dice katsayısını temsil etmektedir. Modelin başarı parametresi olan Dice katsayısı, eğitim seti için 49. Epoch’ta 0.938 değeri ile maksimum noktasına ulaşmıştır. Validasyon seti için ise maksimum 0.803 değerine ulaşılmıştır. Test seti üzerinde ise ortalama başarı değeri hesaplaması ile 0.811 değeri elde edilmiştir.



Şekil 4.3.2. Model eğitimi süresince Dice Katsayısı değerinin değişim grafiği

#### 4.4. Korpus Kallozum İndeks Hesaplama Algoritması

Çalışma kapsamında sunulan çözümün esas amacı KKİ değerinin hesaplanmasının otomatize edilmesidir. Segmentasyon işlemi ile ilk aşama tamamlanıp ardından KKİ hesaplama algoritması ile indeks değeri bulunacaktır. Kullanılan algoritmanın Pseudo kodu aşağıda yer almaktadır.

*Eğitilen model yardımı ile girdi resimleri için maskelenmiş çıktı resimlerini al*

*Her bir maskelenmiş çıktı resmi için:*

*Aşağıdan yukarıya ve soldan sağa tarama yaparak maskenin en soldaki piksel konumunu bul*

*Bulunan konumdan sağa doğru piksel değerlerini kontrol et ve maske dışına çıkana kadar devam et ve piksel sayısını say*

*Elde edilen piksel sayısını genu kalınlığı olarak belirle*

*Aşağıdan yukarıya ve sağdan sola tarama yaparak maskenin en sağındaki piksel konumunu bul*

*Bulunan konumdan sola doğru piksel değerlerini kontrol et ve maske dışına çıkana kadar devam et ve piksel sayısını say*

*Elde edilen piksel sayısını splenium kalınlığı olarak belirle*

*En sol ve en sağdaki pikselin ortasında yer alan pikselin  $x$  değerini al*

*Belirlenen  $x$  değerinde en üstten aşağıya doğru piksel değerlerini kontrol et*

*Maske değerini bulunca sayacı başlat ve maskeden çıkana kadar piksel değerlerini kontrol et*

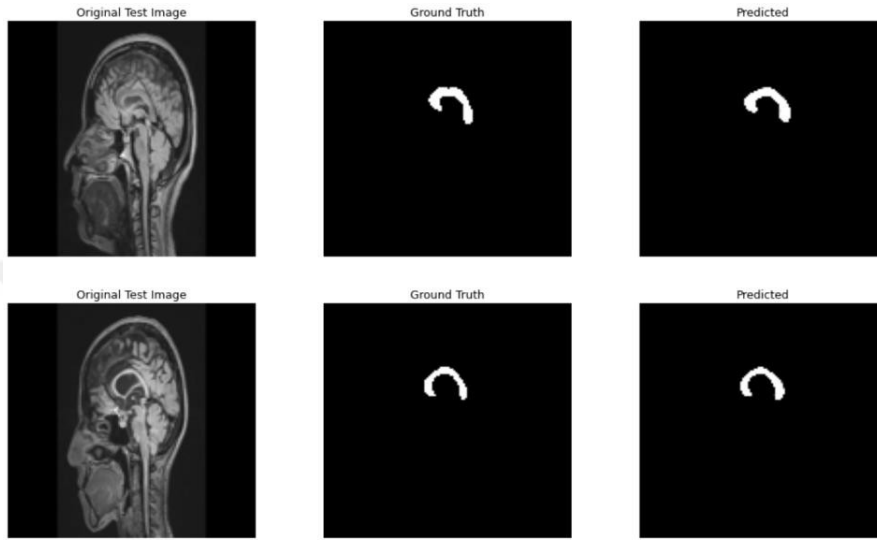
*Piksel sayacındaki değeri korpus kalınlığı olarak tanımla*

*En sol ve en sağdaki pikselin arasındaki uzaklığı hipotenüs yardımı ile bul*

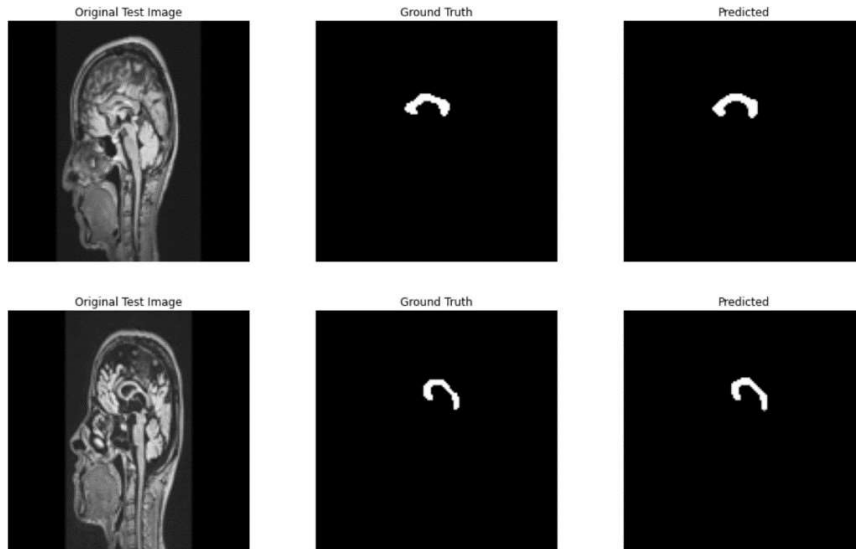
*KKİ denklemini kullanarak sonucu üret*

## 5. SONUÇLAR

Çalışma kapsamında kullanılan veri setleri için eğitim işlemi ayrı ayrı gerçekleştirilmiş ve ayrı modeller elde edilmiştir. Eğitilen model mimarileri ile elde edilen segmentasyon çıktıları Şekil 5.1 ve Şekil 5.2’de yer almaktadır.

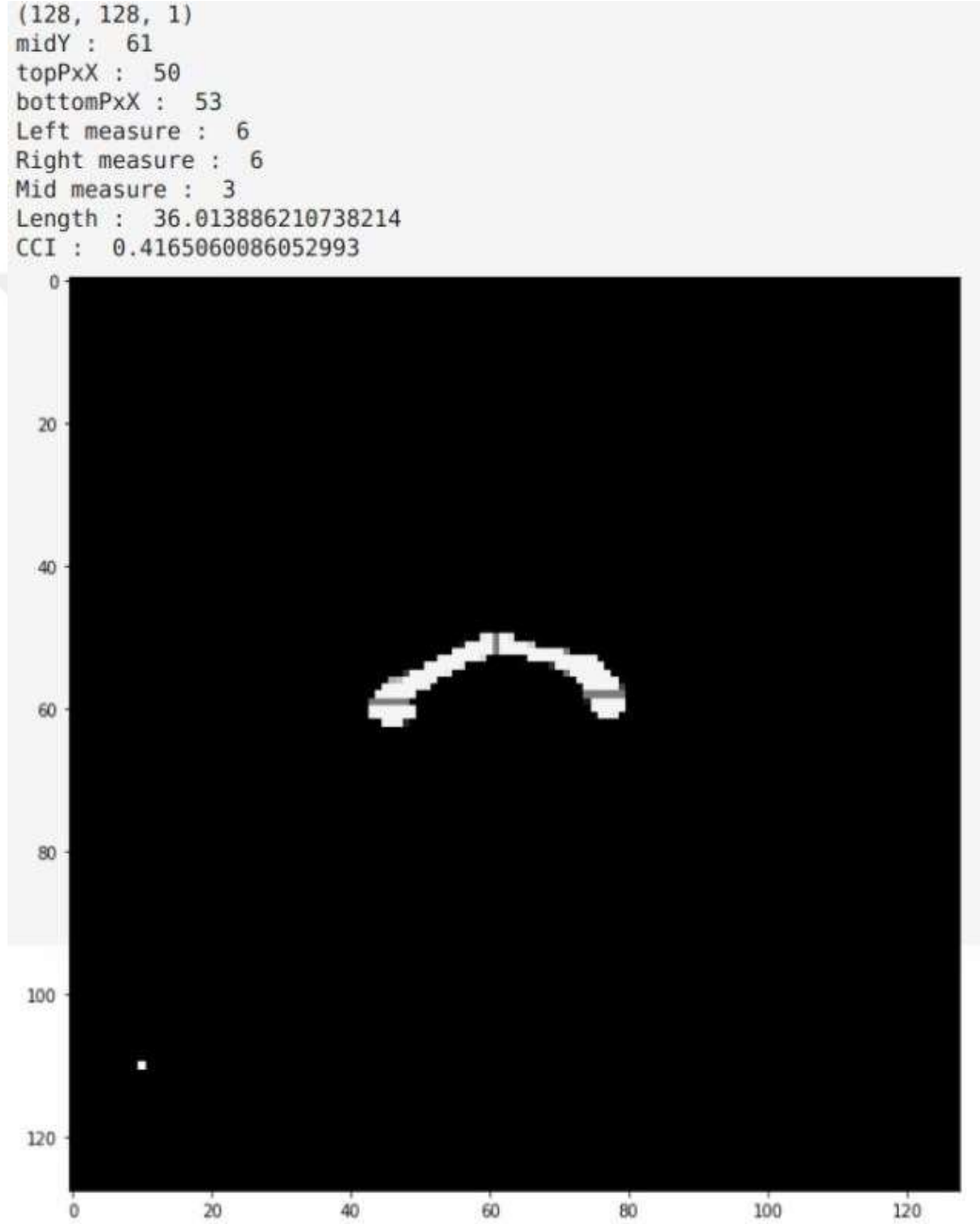


**Şekil 5.1.** Şekilde orjinal resim, etiket (Ground Truth), ve segmentasyon çıktıları (Predicted) yer almaktadır



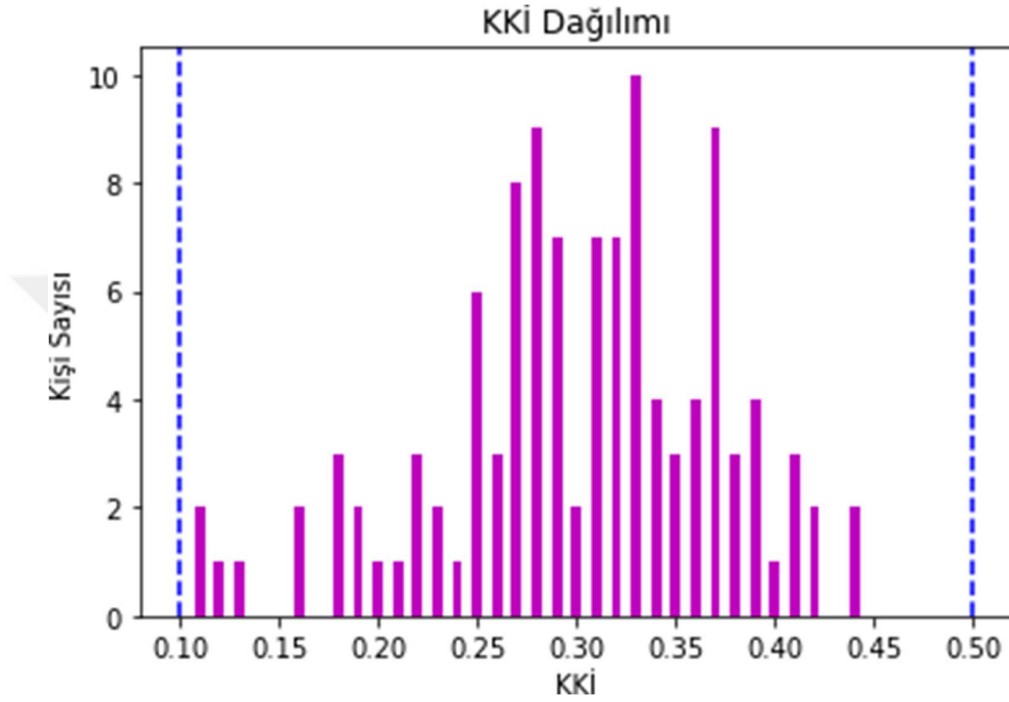
**Şekil 5.2.** Kontrol grubu örnekleri üzerinde elde edilen segmentasyon sonuçları

Elde edilen segmentasyon çıktılarının üzerinde KKİ hesap algoritması uygulanmıştır. Uygulama esnasında görselleştirme amacıyla genu, splenium, ve korpus kalınlıkları için algoritmanın tarama yaptığı hatlara gri şerit çekilmiştir. Şekil 5.3’de örnek KKİ hesabı yer almaktadır.



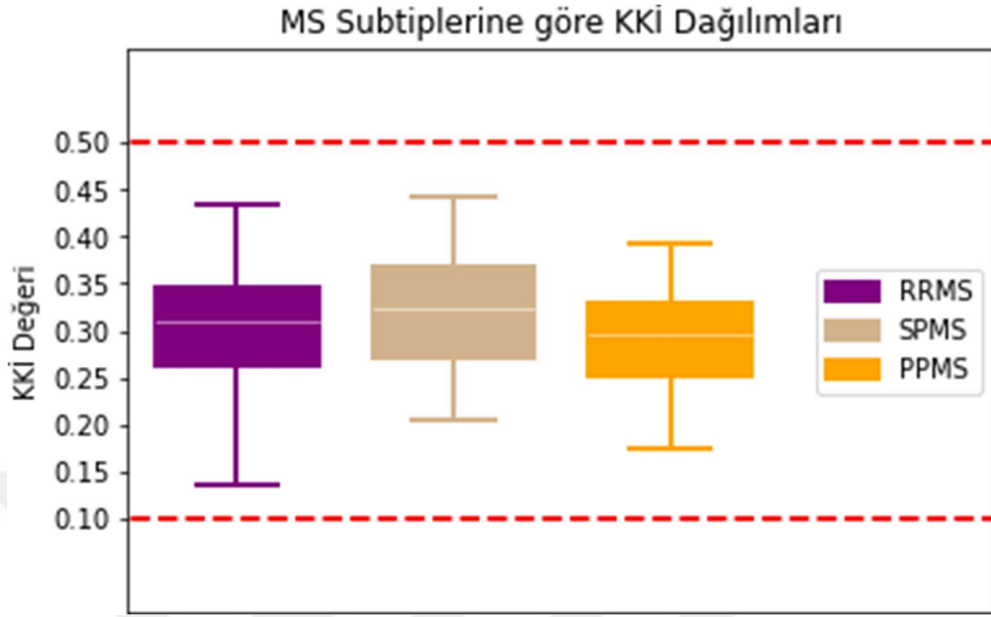
Şekil 5.3. Örnek Korpus Kallozum İndeksi hesabı

Çalışılan veri seti üzerinde elde edilen KKİ değerlerini dağılımı Şekil 5.4’de gösterilmiştir. Şekilde de görüldüğü üzere KKİ değerleri 0.1 ile 0.5 değerleri arasında yer almıştır. Grafikte x ekseninde korpus kallosum indeksi değeri ve y ekseninde ilgili değer için veri setindeki kişi sayısı yer almıştır.



**Şekil 5.4.** Çalışılan veri seti üzerinde elde edilen korpus kallosum indeksi dağılımı

Model eğitimi için ön-işlem aşamasında oluşturulan ve resim indekslerini barındıran DataFrame ile MS subtip durumu belirli olan hastaların resimleri eşleştirilmiştir. Bu sayede MS subtipi belirli olan hastaların KKİ değerlerinin ilişkisinin incelenmesi amaçlanmıştır. Eşleştirme işlemi kapsamında Python uyumlu Pandas modülü kullanılmıştır. Eşleştirme sonucunda, Şekil 5.5’de görüldüğü üzere MS subtiplerine göre KKİ dağılımları elde edilmiştir.



**Şekil 5.5.** MS subtiplerine göre Korpus Kallozum İndeksi dağılımları

## 6. KAYNAKLAR

- Abadi, Martín & Barham, Paul & Chen, Jianmin & Chen, Zhifeng & Davis, Andy & Dean, Jeffrey & Devin, Matthieu & Ghemawat, Sanjay & Irving, Geoffrey & Isard, Michael & Kudlur, Manjunath & Levenberg, Josh & Monga, Rajat & Murray, Derek & Steiner, Benoit & Tucker, Paul & Vasudevan, Vijay & Warden, Pete & Zhang, Xiaoqiang. (2016). *TensorFlow: A system for large-scale machine learning*.
- ABIDE Veri Seti. [https://sites.google.com/site/hpardoe/cc\\_abide](https://sites.google.com/site/hpardoe/cc_abide) [Son erişim tarihi: 08.01.2022].
- Airbus TensorFlow Project. Blog. <https://blog.tensorflow.org/2020/04/how-airbus-detects-anomalies-iss-telemetry-data-tfx.html> [Son erişim tarihi: 08.01.2022].
- Batch Size. Difference Between a Batch and Epoch in a Neural Network. <https://machinelearningmastery.com/difference-between-a-batch-and-an-epoch/> [Son erişim tarihi: 16.12.2021].
- Cell Tracking Challenge. <http://celltrackingchallenge.net/> [Son erişim tarihi: 08.01.2022]
- Chandra, Anjali & Verma, Shrish & Raghuvarshi, Ajay & Bodhey, Narendra. (2022). CCsNeT: Automated Corpus Callosum segmentation using fully convolutional network based on U-Net. *Biocybernetics and Biomedical Engineering*, 42. 10.1016/j.bbe.2021.12.008.
- Concatenated Skip Connection. <https://paperswithcode.com/method/concatenated-skip-connection> [Son erişim tarihi: 22.01.2022].
- Conv2D Katmanı. [https://keras.io/api/layers/convolution\\_layers/convolution2d/](https://keras.io/api/layers/convolution_layers/convolution2d/) [Son erişim tarihi: 16.12.2021].
- Dice Skoru ve IoU. <https://stats.stackexchange.com/questions/273537/f1-dice-score-vs-iou/276144#276144> [Son erişim tarihi: 14.12.2021].
- Dolz, Jose & Xiaopan, Xu & Rony, Jérôme & Yuan, Jing & Liu, Yang & Granger, Eric & Desrosiers, Christian & Zhang, Xi & Ben Ayed, Ismail & Lu, Hongbing. (2018). *Multi-region segmentation of bladder cancer structures in MRI with progressive dilated convolutional networks*.
- Dropout Katmanı. [https://keras.io/api/layers/regularization\\_layers/dropout/](https://keras.io/api/layers/regularization_layers/dropout/) [Son erişim tarihi: 16.12.2021].
- Evrişim Katmanı. Convolutional Neural Networks. <https://www.superdatascience.com/blogs/convolutional-neural-networks-cnn-step-1b-relu-layer/> [Son erişim tarihi: 04.12.2021].
- Fechter, Tobias & Adebahr, Sonja & Baltas, Dimos & Ben Ayed, Ismail & Desrosiers,

- Christian & Dolz, Jose. (2017). Esophagus segmentation in CT via 3D fully convolutional neural network and random walk. *Medical Physics*, 44. 10.1002/mp.12593.
- Figueira, Fernando & Santos, Valeria & Figueira, Gustavo & Silva, Angela. (2008). Corpus Callosum Index: A practical method for long-term follow-up in multiple sclerosis. *Arquivos de neuro-psiquiatria*, 65. 931-5. 10.1590/S0004-282X2007000600001.
- Fitsiori, Aikaterini & Nguyen, Duy & Karentzos, Alexandros & Delavelle, J & Vargas, Maria. (2011). The corpus callosum: White matter or terra incognita. *The British journal of radiology*, 84. 5-18. 10.1259/bjr/21946513.
- Google Colaboratory. About. <https://colab.research.google.com/> [Son erişim tarihi: 15.12.2021].
- Herrera, William & Cover, Giovana & Rittner, Leticia. (2018). Pixel-Based Classification Method for Corpus Callosum Segmentation on Diffusion-MRI. *Lecture Notes in Computational Vision and Biomechanics*, 27. 217-224. 10.1007/978-3-319-68195-5\_24.
- İçer, S. (2013). Automatic segmentation of corpus callosum using Gaussian mixture modeling and Fuzzy C means methods. *Comput. Methods Progr. Biomed*, 112, 38–46. doi: 10.1016/j.cmpb.2013.06.006
- Ignacio Arganda-Carreras, Srinivas C. Turaga, Daniel R. Berger, Dan Ciresan, Alessandro Giusti, Luca M. Gambardella, Jürgen Schmidhuber, Dmitry Laptev, Sarvesh Dwivedi, Joachim M. Buhmann, Ting Liu, Mojtaba Seyedhosseini, Tolga Tasdizen, Lee Kamentsky, Radim Burget, Vaclav Uher, Xiao Tan, Chanming Sun, Tuan D. Pham, Eran Bas, Mustafa G. Uzunbas, Albert Cardona, Johannes Schindelin, and H. Sebastian Seung. Crowdsourcing the creation of image segmentation algorithms for connectomics. *Frontiers in Neuroanatomy*, vol. 9, no. 142, 2015.
- Inspace TensorFlow Project. Blog. <https://blog.tensorflow.org/2020/12/inspace-new-video-conferencing-platform-uses-tensorflowjs-for-toxicity-filters-in-chat.html> [Son erişim tarihi: 08.01.2022].
- ISBI Challenge. Computer-Automated Detection of Caries in Bitewing Radiography. <http://www-o.ntust.edu.tw/~cweiwang/ISBI2015/challenge2/index.html> [Son erişim tarihi: 08.01.2022].
- Jaccard Index. [https://en.wikipedia.org/wiki/Jaccard\\_index](https://en.wikipedia.org/wiki/Jaccard_index) [Son erişim tarihi: 14.12.2021].
- Keras ve TensorFlow. <https://www.upgrad.com/blog/the-whats-what-of-keras-and-tensorflow/> [Son erişim tarihi: 13.12.2021].
- Krizhevsky, A., Sutskever, I., Hinton, G.E. 2012. *Imagenet classification with deep*

- convolutional neural networks*. In: NIPS. pp. 1106–1114
- Labelbox. Documents. <https://docs.labelbox.com/docs> [Son erişim tarihi: 26.12.2021].
- Li, Yue & Wang, Huiquan & Ahmed, Nizam & Mandal, Mrinal. (2017). AUTOMATED CORPUS CALLOSUM SEGMENTATION IN MIDSAGITTAL BRAIN MR IMAGES. *ICTACT Journal on Image and Video Processing*, 8. 1554-1565. 10.21917/ijivp.2017.0220.
- Llufriu, Sara & Blanco, Yolanda & Martínez-Heras, Eloy & Casanova-Molla, Jordi & Gabilondo, Iñigo & Sepulveda, Maria & Falcón, Carles & Berenguer, Joan & Bargalló, Nuria & Villoslada, Pablo & Graus, Francesc & Valls-Sole, Josep & Saiz, lbert. (2012). Influence of Corpus Callosum Damage on Cognition and Physical Disability in Multiple Sclerosis: A Multimodal Study. *PloS one*, 7. e37167. 10.1371/journal.pone.0037167.
- Matplotlib. About. <https://matplotlib.org/> [Son erişim tarihi: 26.12.2021].
- Michael, Platten & Brusini, Irene & Andersson, Olle & Ouellette, Russell & Piehl, Fredrik & Wang, Chunliang & Granberg, Tobias. (2021). Deep Learning Corpus Callosum Segmentation as a Neurodegenerative Marker in Multiple Sclerosis. *Journal of neuroimaging : official journal of the American Society of Neuroimaging*, 31. 10.1111/jon.12838.
- NumPy. About. <https://numpy.org/doc/stable/user/whatisnumpy.html> [Son erişim tarihi: 09.12.2021].
- OASIS Veri Seti. <https://www.oasis-brains.org/> [Son erişim tarihi: 08.01.2022].
- OpenCV. About. <https://opencv.org/about/> [Son erişim tarihi: 10.12.2021].
- Ozer, Pınar. (2021). Correlation of corpus callosum index and optic coherence tomography findings in multiple sclerosis with or without optic nerve involvement. *European Eye Research*, 10.14744/eer.2021.03522.
- Pandas. Python Data Analysis Library. <https://pandas.pydata.org/about/> [Son erişim tarihi: 09.01.2022].
- Park G, Kwak K, Seo SW, Lee JM. Automatic Segmentation of Corpus Callosum in Midsagittal Based on Bayesian Inference Consisting of Sparse Representation Error and Multi-Atlas Voting. *Front Neurosci*, 2018 Sep 11;12:629. doi: 10.3389/fnins.2018.00629. PMID: 30271320; PMCID: PMC6142891.
- Pedregosa, Fabian & Varoquaux, Gael & Gramfort, Alexandre & Michel, Vincent & Thirion, Bertrand & Grisel, Olivier & Blondel, Mathieu & Prettenhofer, Peter & Weiss, Ron & Dubourg, Vincent & Vanderplas, Jake & Passos, Alexandre & Cournapeau, David

- & Brucher, Matthieu & Perrot, Matthieu & Duchesnay, Edouard & Louppe, Gilles. (2012). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*. 12.
- Pixel Accuracy. <https://towardsdatascience.com/image-segmentation-kaggle-experience-9a41cb8924f0> [Son erişim tarihi: 14.12.2021].
- Python. About. <https://www.python.org/about/> [Son erişim tarihi: 08.12.2021].
- ReLU. Not a Differentiable Function. <https://medium.com/@kanchansarkar/relu-not-a-differentiable-function-why-used-in-gradient-based-optimization-7fef3a4cecec> [Son erişim tarihi: 04.12.2021].
- Ronneberger, Olaf & Fischer, Philipp & Brox, Thomas. (2015). *U-Net: Convolutional Networks for Biomedical Image Segmentation*.
- Semantic Segmentation Metrics. <https://towardsdatascience.com/metrics-to-evaluate-your-semantic-segmentation-model-6bcb99639aa2> [Son erişim tarihi: 14.12.2021].
- Skourt, Brahim & El hassani, Abdelhamid & Majda, Aicha. (2018). Lung CT Image Segmentation Using Deep Neural Networks. *Procedia Computer Science*, 127. 109-113. 10.1016/j.procs.2018.01.104.
- Smooth Hiperparametresi. <https://stackoverflow.com/a/63243092/8712569> [Son erişim tarihi: 16.12.2021].
- Texas Instruments. Machine Learning. [http://software-dl.ti.com/processor-sdk-linux/esd/docs/latest/linux/Foundational\\_Components\\_Machine\\_Learning.html](http://software-dl.ti.com/processor-sdk-linux/esd/docs/latest/linux/Foundational_Components_Machine_Learning.html) [Son erişim tarihi: 08.01.2022].
- U-Net. Convolutional Networks for Biomedical Image Segmentation. <https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net/> [Son erişim tarihi: 18.04.2021].
- Yaldizli O, Atefy R, Gass A, Sturm D, Glassl S, Tettenborn B, Putzki N. Corpus callosum index and long-term disability in multiple sclerosis patients. *J Neurol*, 2010 Aug;257(8):1256-64. doi: 10.1007/s00415-010-5503-x. Epub 2010 Mar 3. PMID: 20198382.

## ÖZGEÇMİŞ

**Tahirhan YILDIZOĞLU**

### ÖĞRENİM BİLGİLERİ

|               |                                                                           |
|---------------|---------------------------------------------------------------------------|
| Yüksek Lisans | Akdeniz Üniversitesi                                                      |
| 2019-2022     | Fen Bilimleri Enstitüsü, Elektrik-Elektronik Mühendisliği Bölümü, Antalya |
| Lisans        | Akdeniz Üniversitesi                                                      |
| 2013-2018     | Fen Bilimleri Enstitüsü, Elektrik-Elektronik Mühendisliği Bölümü, Antalya |

### MESLEKİ VE İDARİ GÖREVLER

|                   |                                    |
|-------------------|------------------------------------|
| Yazılım Mühendisi | Notilus Yacht Design & Engineering |
| 2020-Devam Ediyor | Antalya                            |
| Yazılım Mühendisi | Art-In Systems                     |
| 2018-2020         | Antalya                            |