



**FATİH SULTAN MEHMET VAKIF ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR MÜHENDİSLİĞİ PROGRAMI**

**BEYİN DAMAR GÖRÜNTÜLERİNDE
DALLANMA VE UÇ NOKTALARIN
DERİN ÖĞRENME YÖNTEMLERİ İLE TESPİTİ**

DOKTORA TEZİ

SAMET KAYA

İSTANBUL, 2024



**FATİH SULTAN MEHMET VAKIF ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR MÜHENDİSLİĞİ PROGRAMI**

**BEYİN DAMAR GÖRÜNTÜLERİNDE
DALLANMA VE UÇ NOKTALARIN
DERİN ÖĞRENME YÖNTEMLERİ İLE TESPİTİ**

DOKTORA TEZİ

**SAMET KAYA
(191221001)**

**Danışman
(Doç. Dr. Berna Kiraz)**

**Eş Danışman
(Prof. Dr. A. Yılmaz Çamurcu)**

**Fatih Sultan Mehmet Vakıf Üniversitesi tarafından
22022B1Ç01D proje numarası ile desteklenmiştir.**

İSTANBUL, 2024

11 /12/2024

LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ MÜDÜRLÜĞÜNE

Bilgisayar Mühendisliği Anabilim Dalı Bilgisayar Mühendisliği programı öğrencisi 191221001 numaralı Samet KAYA'nın hazırladığı “Beyin Kan Damarlarının ve İlişkili Hücrelerin Derin Öğrenme Yöntemleri ile Üç Boyutlu Modellenmesi” konulu Doktora Yeterlik tezi ile ilgili Tez Savunma Sınavı, 11/12/2024 Çarşamba günü saat 15:30’da yapılmış, sorulara alınan cevaplar sonunda adayın tezinin **Kabulüne Oy Çoğunluğu/Oy Birliği** ile karar verilmiştir.

Tez adı değişikliği yapılması halinde: Tez adının “ Beyin Damar Görüntülerinde Dallanma ve Uç Noktaların Derin Öğrenme Yöntemleri ile Tespiti ” şeklinde değiştirilmesi uygundur.

Jüri Üyesi	Karar
1. (Danışman) Doç. Dr. Berna KİRAZ	Kabul.....
2. Dr. Öğr. Üyesi Şaban SAHMOUD	Kabul.....
3. Prof. Dr. Alper KİRAZ	Kabul.....
4. Dr. Öğr. Üyesi Ali NİZAM	Kabul.....
5. Doç. Dr. Önder DEMİR	Kabul.....
6. (İkinci Danışman)*. Prof. Dr. Ali Yılmaz ÇAMURCU	Kabul.....

*2. Danışman varsa doldurulması gerekmektedir.

ETİK BİLDİRİM

Bu tezin yazılmasında bilimsel ahlak kurallarına uyulduğunu, başkalarının eserlerinden yararlanılması durumunda bilimsel normlara uygun olarak atıfta bulunulduğunu, kullanılan verilerde herhangi bir tahrifat yapılmadığını, tezin herhangi bir kısmının bağlı olduğum üniversite veya bir başka üniversitedeki başka bir çalışma olarak sunulmadığını beyan ederim.

Samet KAYA

BEYİN DAMAR GÖRÜNTÜLERİNDE DALLANMA VE UÇ NOKTALARIN DERİN ÖĞRENME YÖNTEMLERİ İLE TESPİTİ

SAMET KAYA

ÖZET

Mikroskopi teknolojilerindeki gelişmeler, biyomedikal araştırmalarda önemli bir yer edinmiş ve özellikle nöroloji gibi karmaşık biyolojik sistemlerin incelenmesinde kritik bir rol oynamıştır. Beyin damarlarının doğru ve etkin bir şekilde görüntülenmesinin, nörogörüntüleme alanında önemli yeri vardır. Beyin damarlarının yapısal bütünlüğündeki bozulmalar, nörolojik hastalıkların ortaya çıkmasına yol açabilir. Bu nedenle, damar yapılarının dallanma noktaları ve uç noktalarının doğru tespiti, hastalıkların tanısı ve tedavisinde kritik bir rol oynar. Ancak geleneksel görüntü işleme yöntemleri, artan algoritmik karmaşıklık ve görüntü boyutu nedeniyle sınırlamalar taşımaktadır.

Bu tez çalışması, beyin damarlarının iki boyutlu (2B) mikroskop görüntülerinde dallanma ve uç noktalarının tespiti için derin öğrenme tabanlı bir nesne tespiti yöntemi geliştirmeyi ve bu yöntemin hiperparametre optimizasyonunu gerçekleştirmeyi amaçlamaktadır. Laboratuvar ortamında elde edilen fare beyni dilimlerinin ham görüntüler ön işleme süreci işletilerek nesne tespiti yöntemleri için uygun hale getirilmiştir. BM3D algoritması ile gürültü temizleme, üçgen eşikleme yöntemi ile segmentasyon ve 2B inceltme algoritması ile iskelet çıkarma adımlarından geçirilmiştir. Elde edilen dallanma ve uç noktaları, 5x5 piksel boyutunda sınırlayıcı kutularla etiketlenmiş ve COCO formatına dönüştürülmüştür. Daha sonra, Detectron2 çerçevesinde dört farklı derin öğrenme nesne tespiti algoritması kullanılarak modeller eğitilmiş ve bu modellerin performansları Birleşimin Kesişime Oranı (Intersection over Union (IoU)) metriği ile

değerlendirilmiştir. Bazı modellerde %90'ın üzerinde bir başarı oranı elde edilmiştir. Ancak model performansını daha da artırmak için, Bayesci Optimizasyon yöntemi kullanılarak hiperparametre optimizasyonu gerçekleştirilmiştir. Bu yöntem sayesinde model doğruluğu %98'in üzerine çıkarılmıştır. Optimizasyon işlemini otomatikleştirmek için optuna kütüphanesinden faydalanılmıştır.

Tezde ayrıca, Python ile geliştirilen BrainVasculyzer isimli yazılım da geliştirilmiştir. Bu yazılım, beyin damarlarının otomatik analizini sağlayan kullanıcı dostu bir arayüz sunmakta ve mevcut analiz programları ile karşılaştırılarak performansı değerlendirilmiştir. BrainVasculyzer, damarların uzunluğu, dallanma ve uç noktaları gibi bilgileri otomatik olarak çıkarıp görselleştirilmektedir. Bu tez çalışması, derin öğrenme ve hiperparametre optimizasyonunun nörogörüntüleme alanında önemli ilerlemeler sağladığını ve beyin damarlarının analizinde büyük bir potansiyele sahip olduğunu ortaya koymaktadır.

Anahtar kelimeler: Derin öğrenme, Beyin damar analizi, Hiperparametre optimizasyonu, Nesne tespiti, Görüntü işleme, Dallanma ve uç noktaları tespiti

DETECTION OF BRANCHES AND END POINTS IN CEREBRAL VESSEL IMAGES WITH DEEP LEARNING METHODS

SAMET KAYA

ABSTRACT

Advances in microscopy technologies have gained an important place in biomedical research and have played a critical role in the study of complex biological systems, especially in neurology. Accurate and efficient imaging of brain vessels is of vital importance in the field of neuroimaging. Disruptions in the structural integrity of brain vessels can lead to the development of neurological diseases. Therefore, accurate detection of branching points and endpoints of vascular structures plays a critical role in diagnosing and treating diseases. However, traditional image processing methods have limitations due to increasing algorithmic complexity and image size.

This thesis is a significant step forward in the field of neuroimaging. It aims to develop a deep learning-based object detection method for detecting branches and endpoints in two-dimensional (2D) microscope images of cerebral vessels. This method, coupled with hyperparameter optimization, promises to revolutionize the way we analyze brain vessels. Raw images of mouse brain slices obtained in the laboratory environment are subjected to noise removal with the BM3D algorithm, segmentation with the triangle thresholding method and skeletonization with the 2D thinning algorithm. The resulting branches and endpoints were labeled with bounding boxes of 5x5 pixels and converted to COCO format.

Within the scope of the thesis, models were trained using four different deep-learning object detection algorithms within the Detectron2 framework, and their performance was evaluated with the Intersection over Union (IoU) metric. With

some models, a success rate of over 90% was achieved. To improve the model performance further, hyperparameter optimization was performed using Optuna software, and the model accuracy increased by over 98%.

The thesis also introduces BrainVasculyzer, a software developed in Python. This software provides a user-friendly interface for automatic analysis of brain vessels and its performance is evaluated by comparing it with existing analysis programs. BrainVasculyzer can automatically extract and visually present information such as the length, branching and endpoints of the vessels. This thesis not only demonstrates the significant advances made in neuroimaging through deep learning and hyperparameter optimization but also instills hope for the future, showing the great potential these technologies have in brain vessels.

Keywords: *Deep learning, Brain vessel analysis, Hyperparameter optimization, Object detection, Image processing, Branching and endpoint detection.*

ÖNSÖZ

Bu tez çalışmasında, beyin damarlarının iki boyutlu mikroskop görüntülerinde dallanma ve uç noktalarının tespiti için derin öğrenme tabanlı nesne tespiti yöntemi geliştirilmiştir. Çalışmada, beyin damarlarının yapısal analizinde derin öğrenme tekniklerinin etkin bir şekilde nasıl kullanılabileceğini göstermeyi amaçlamaktadır. Çalışma kapsamında, derin öğrenme modellerinin performansını artırmak amacıyla hiperparametre optimizasyonu yapılmış ve mevcut modellerde performans artışı sağlanmıştır. Önerilen yöntem, farklı görüntüler üzerinde test edilmiş ve iyi sonuçlar elde edilmiştir. Tez içerisinde kullanılan algoritma ve modeller ile laborantlar tarafından kullanılabilecek bir program geliştirilmiştir. Araştırmada kullanılan yöntemler, nörogörüntüleme ve diğer biyomedikal görüntüleme alanlarında kullanılabilecek potansiyele sahip olup, çeşitli nörolojik hastalıkların tanısı ve tedavisi için değerli bilgiler sunmaktadır.

Bu tez çalışmasının gerçekleştirilmesinde, her aşamada değerli bilgilerini ve deneyimlerini benimle paylaşan, sabrı ve desteği ile bana yol gösteren çok kıymetli danışman hocam Dr. Öğr. Üyesi Berna KİRAZ'a en içten teşekkürlerimi sunarım. Kendisinin rehberliği, bu tez sürecinde akademik gelişimime büyük katkı sağlamıştır. Aynı zamanda, ikinci danışmanım Prof. Dr. Ali Yılmaz ÇAMURCU'ya da sağladığı değerli katkılar ve yönlendirmeler için şükranlarımı sunarım.

Bu çalışmada kullanılan veri setlerini sağlayarak katkıda bulunan ve desteklerini esirgemeyen Doç. Dr. Atay Vural ve KUTTAM ekibine içtenlikle teşekkür ederim. Sağladıkları bu değerli veri setleri, araştırmamın başarıyla tamamlanmasında önemli bir rol oynamıştır. Destek ve iş birlikleri için minnettarım.

Tez önerisi aşamasından itibaren her adımda desteğini ve bilgisini benimle paylaşan, farklı bakış açıları kazanmamı sağlayan Dr. Öğr. Üyesi Musa AYDIN'a ve Dr. Öğr. Üyesi Zeki KUŞ'a da teşekkür ederim.

Tez ara sunumları ve savunma sürecimde bilgi ve tecrübeleriyle desteklerini esirgemeyen Prof. Dr. Alper Kiraz ve Dr. Öğr. Üyesi Şaban Sahmoud'a en içten

teşekkürlerimi sunarım. Ayrıca, tez savunmasında jüri üyesi olarak yer alarak çalışmalarımı değerlendiren Doç. Dr. Önder Demir ve Dr. Öğr. Üyesi Ali Nizam'a katkıları, önerileri ve yapıcı eleştirileri için minnettarım.

Son olarak, bu sürecin her anında yanımda olan sevgili eşim Gökçe'ye ve sürecin içine doğan kızım Gülce Sena'ya, gösterdikleri sabır, anlayış ve verdikleri sevgi için minnettarım. Onların desteği, bu çalışmanın en önemli dayanağı olmuştur. Süreç boyunca her zaman yanımda olan ve desteklerini esirgemeyen sevgili aileme, bana verdikleri moral, motivasyon ve sınırsız sevgi için sonsuz teşekkür ederim. Onların varlığı, bu çalışmanın arkasındaki en büyük güç olmuştur.

Aralık, 2024

Samet KAYA

İÇİNDEKİLER

ÖZET.....	v
ABSTRACT	vii
ÖNSÖZ.....	ix
TABLO LİSTESİ	xiii
ŞEKİL LİSTESİ.....	xiv
KISALTMALAR	xvi
GİRİŞ	1
BİRİNCİ BÖLÜM.....	8
1. LİTERATÜR TARAMASI	8
1.1. Derin Öğrenme Nesne Tanıma Metotları ile Damar Analizi	8
1.1.1. Derin Öğrenme Çalışmalarında Kullanılan Modeller	11
1.1.2. Derin Öğrenme Çalışmalarında Kullanılan Kütüphaneler	13
1.2. Derin Öğrenme Metotlarında Hiperparametre Optimizasyonu	14
1.3. Damar Analizi İçin Geliştirilmiş Programlar	17
İKİNCİ BÖLÜM	19
2. GÖRÜNTÜLERE ÖN İŞLEME YAPILMASI VE VERİ SETİ OLUŞTURULMASI	19
2.1. Görüntü Analizi ve Ön İşleme Çalışması	19
2.1.1. Gürültü Temizleme	20
2.1.2. Görüntü Keskinleştirme	22
2.1.3. Görüntü Yumuşatma	24
2.1.4. Görüntü Eşikleme	25
2.1.5. İkili Kapanma Filtresi	29
2.1.6. Küçük Nesnelerin Silinmesi	30
2.1.7. Görüntü İskeletinin Çıkarılması	31
2.2. Dallanma ve Uç Noktalarının Geleneksel Görüntü İşleme Yöntemleri ile Tespiti	34
2.2.1. Damar Ağı Özellikleri	35
2.2.1.1. Damarsal Alan Fraksiyonu	35
2.2.1.2. Vasküler Uzunluk	36
2.2.1.3. Vasküler Dallanma Sayısı	37
2.2.1.4. Vasküler Uç Nokta Sayısı	38
2.3. Tespit Edilmiş Nesnelerin Veri Seti Haline Getirilmesi	41
ÜÇÜNCÜ BÖLÜM	46
3. DERİN ÖĞRENME METODLARI İLE NESNE TESPİTİ.....	46

3.1. Derin Öğrenme Metotları İle Dallanma Ve Uç Noktaların Bulunması.....	49
3.1.1. Model ve Kütüphane Seçimi	49
3.1.2. Eğitim ve Test.....	51
3.1.3. Değerlendirme Yöntemi	53
3.1.3.1. Referans ve Tahmin Nesnelerinde Birleşimin Kesişime Oranı	54
3.1.3.2. Ortalama Kesinlik ve Duyarlılık	56
3.1.3.3. Değerlendirme Çıktısı	56
3.1.4. Deney Sonuçların Gösterimi	58
3.1.5. Deneysel Çalışma 1	59
3.1.6. Deneysel Çalışma 2	63
3.1.7. Zaman Karşılaştırması.....	65
3.1.8. Bulgular ve Tartışma.....	66
3.2. Hiperparametre Optimizasyonu.....	68
3.2.1. Metodoloji.....	69
3.2.2. Optimizasyon Çalışması.....	70
3.2.3. En İyi Parametrelerin Test Edilmesi	74
3.2.4. Bulgular ve Tartışma.....	76
DÖRDÜNCÜ BÖLÜM	77
4. PROGRAM GELİŞTİRİLMESİ.....	77
4.1. BrainVasculyzer: 2B Beyin Damar Görüntü Analiz Programı	77
4.1.1. Çalışmanın Program Haline Getirilme Süreci.....	77
4.1.2. Programın Çalışma Ekranları.....	78
4.2. Geliştirilen Programın Test Edilmesi	85
4.2.1. Referans Program ile Karşılaştırma	85
4.2.2. Bulgular ve Tartışma.....	86
SONUÇ.....	87
KAYNAKÇA	89

TABLO LİSTESİ

Tablo 2.1: Damar Yollarının Gösterilmesi	37
Tablo 3.1: Modellerde Kullanılan Başlangıç Parametreleri	52
Tablo 3.2: HFHS13 veri kümesi üzerinde Dört Model Tarafından Elde Edilen Ortalama Kesinlik ve Duyarlılık Değerleri	60
Tablo 3.3: HFHS13 veri kümesi üzerinde Dört Model Tarafından Elde Edilen Dallanma ve Uç Nokta Sayıları Tablosu	61
Tablo 3.4: NC15 veri kümesi üzerinde Dört Model Tarafından Elde Edilen Ortalama Kesinlik ve Duyarlılık Değerleri	63
Tablo 3.5: NC15 veri kümesi üzerinde Dört Model Tarafından Elde Edilen Dallanma ve Uç Nokta Sayıları Tablosu	64
Tablo 3.6: HFHS13 (20 görüntü) Eğitim ve Test Süreleri Karşılaştırması	65
Tablo 3.7: KUVESG (156 görüntü) Test Süresi Karşılaştırması	66
Tablo 3.8: Parametre Arama Uzayı	70
Tablo 3.9: Deneylerin En İyi On Sonucu	74
Tablo 3.10: NC15 Veri Seti ile Optimize Edilmiş Hiperparametlerin Test Sonuçları	75
Tablo 4.1: Fiji ile BV Karşılaştırması	85

ŞEKİL LİSTESİ

Şekil 2.1: Görüntülerden Damar Haritasının Çıkarılması	20
Şekil 2.2: İşlenmemiş Görüntü	21
Şekil 2.3: Gürültüden Arındırılmış Görüntü	22
Şekil 2.4: Keskinleştirme Filtresi Uygulanmış Görüntü	23
Şekil 2.5: Gauss Filtresi Uygulanmış Görüntüler	25
Şekil 2.6: Farklı Filtrelerle Eşiklenmiş Görüntüler	27
Şekil 2.7: Eşiklenmiş Görüntüde Üzerine İkili Kapatma Filtresi İyileştirmesi Uygulanmış Görüntü Karşılaştırması. (a): Triangle Eşiklenmiş Görüntü, (b): İkili Kapatma Filtresi Uygulanmış Görüntü	30
Şekil 2.8: Referans Görüntü ve Küçük Bölütleri Silinmiş Görüntü Karşılaştırması. (a): Referans Görüntü, (b): Küçük Bölütler Silinmiş Görüntü	31
Şekil 2.9: Orijinal Görüntü Temizlenmesi ve İskelet Şeması Görüntüleri. (a): Orijinal Görüntü, (b): Bölütlenmiş Görüntü, (c): Görüntü İskeleti	32
Şekil 2.10: Damar İskeleti ile Temizlenmiş Görüntünün Üstüste Hali	33
Şekil 2.11: İskelet Çıkarma Çalışmasının Diğer Yöntemler ile Karşılaştırılması. (a): Orijinal , (b): ResNet , (c): UNAS-Net, (d): Uygulanan Metot	34
Şekil 2.12: Örnek Dallanma Noktaları	38
Şekil 2.13: Örnek Uç Noktalar	38
Şekil 2.14: Uç Nokta Bulma Filtreleri	40
Şekil 2.15: Dallanma Noktası Bulma Filtreleri	40
Şekil 2.16: Örnek Görüntüdeki Dallanma Noktaları ve Uç noktalar. (a): İskelet Görüntüsü, (b): İşaretlenmiş Görüntü. Yeşil Daireler Uç Noktaları, Kırmızı Daireler ise Dallanma Noktalarını Göstermektedir	41
Şekil 2.17: 5x5 Piksel Çerçeveler ile İşaretlenmiş Dallanma ve Uç Noktaları	43
Şekil 2.18: COCO Formatındaki Özet Dosya	44
Şekil 3.1: Üstüste Gelmiş Nesne Tespit Çerçeveleri	54
Şekil 3.2: Birleşimin Kesişime Oranı (IoU) Formülü	55

Şekil 3.3: COCO Evaluator Çıktısı	57
Şekil 3.4: Hiperparametrelerin Paralel Koordinat Grafikleri. (a) AP IOU 0,50, (b) AP IOU 0,75 ve (c) AR	71
Şekil 3.5: Hiperparametrelerin Dağılım Grafikleri. (a) AP IOU 0.50, (b) AP IOU 0.75 ve (c) AR	73
Şekil 4.1: Görüntü Yükleme	79
Şekil 4.2: Gürültü Temizleme	79
Şekil 4.3: Bölütleme	80
Şekil 4.4: İskelet Çıkarımı	80
Şekil 4.5: Analiz İşeminin Başlatılması	81
Şekil 4.6: Damar Yollarının Gösterilmesi	81
Şekil 4.7: Uç noktaların Gösterimi	82
Şekil 4.8: Dallanma Noktalarının Gösterilmesi	82
Şekil 4.9: Analiz Verilerinin Gösterimi. (a): Damar Id ve Uzunluk Gösterimi, (b): Dallanma (Kırmızı) ve Uç (Yeşil) Noktaları Gösterimi	83
Şekil 4.10: Programın Tüm Görüntüsü	83
Şekil 4.11: Raporlama Seçenekleri	84
Şekil 4.12: Analiz Raporu	84

KISALTMALAR

AP	Avarage Precision
AR	Avarage Recall
BM3D	Block-matching and 3D filtering algorithm
BV	BrainVasculyzer
CNN	Convolutional Neural Networkten
COCO	Common Objects in Context
DÖ	Derin Öğrenme
F_BP_C	Filtered Branch Point Count
F_TP_C	Filtered Tip Point Count
GGİ	Geleneksel Görüntü İşleme
GT_BP_C	Ground-Truth Branch Point Count
GT_TP_C	Ground-Truth Tip Point Count
MAT	Medical Axis Transformation
NVU	Nörovasküler Ünite
OH	Ortalama Hassasiyet
OK	Ortalama Kesinlik
R_BP_C	Raw Branch Point Count
R_TP_C	Raw Tip Point Count
SAT	Symmetry Axis Transformation

GİRİŞ

Damar sistemi tüm vücudun düzgün fonksiyon göstermesi için kritik öneme sahiptir. Vücutta gereken ilgili yerlere oksijen, glikoz gibi gerekli maddeler damarlar yoluyla iletilir ve atık maddeler damar sistemi tarafından toplanır. Bunlar da metabolizmanın sağlıklı sürdürülebilirliği açısından vücutta bulunan damar sistemlerinin önemini göstermektedir.

Vücutta bulunan damar sistemleri bir şekilde birbirlerine bağlıdır. Bu sistemleri kısmi olarak ele alınıp incelemek daha olanaklı ve anlaşılırdır. Bu tezde tezde beyin damar sistemi ele alınmaktadır. Beynin sağlıklı fonksiyon göstermesi için bu kadar kritik öneme sahip olan damarsal sistemin organizasyonu diğer dokulara göre çok daha gelişmiştir, damar yoğunluğu fazladır ve nöronlarla yakın ilişki içerisindedir. Beyinde, diğer organlardan farklı olarak kan akımı kan tüm organ düzeyinde düzenlenmez ve sadece belirli bir görevi yerine getirirken aktif çalışması gereken konfokal mikro bölgelerde giden kan akımı düzenlenir. Böylesi hassas bir regülasyon nörovasküler ünite (NVU) adı verilen yapı ile başlayan ve pek çok hücrenin dahil olduğu özel bir sistemle sağlanır.

Yukarıda da bahsedildiği gibi damar sistemi pasif bir borular sistemi olmayıp, çok hücrenin katıldığı ve bir arada çalıştığı aktif ve hassas bir sistemdir. Bu sistemin yapısında endotel hücreler, perisitler, düz kas hücreleri, astrositler bulunur ve tüm bu hücreler birbirleriyle ve diğer nöral hücreler ile yakın ve sürekli iletişim halindedirler. Farklı hastalık durumlarında ve yaşlanma sürecinde bu sistemin farklı bileşenlerinin beyin farklı bölgelerinde nasıl etkilendiğini anlayabilmek için insan beyni otopsi materyallerinin yanı sıra, daha sık olarak fare ve zebra balığı gibi sistemlerde hastalık modelleri kullanılır (Özkan vd., 2023). Damar sistemini incelerken karşılaşılan en büyük zorluk, son derece karmaşık ve çok olması aynı anda organ genelinde mikro seviyede birden fazla parametreyi aynı anda inceleyebilecek yöntemlerin eksikliğidir.

Beyin damarlarının incelenmesi, nörogörüntüleme alanında önemli bir yer tutar. Beyin damarları, beyin farklı bölgelerine oksijen ve besin maddeleri taşıyan

karmaşık bir ağ oluşturur. Bu damarların yapısal bütünlüğünde meydana gelen herhangi bir bozulma, çeşitli nörolojik hastalıkların ortaya çıkmasına neden olabilir. Örneğin, beyin damarlarındaki bir tıkanıklık, inme gibi ciddi sağlık sorunlarına yol açabilir. Bu nedenle, beyin damarlarının doğru ve etkin bir şekilde analiz edilmesi, bu tür hastalıkların erken tanısında kritik bir rol oynar. Nörolojik hastalıkların tanısı ve tedavisi için beyin damarlarının detaylı bir şekilde incelenmesi, yapısal özellikleri, sağlıklı olup olmadığını belirlemede ve olası hastalıkları tespit etmede kritik bir rol oynar. Modern mikroskopi teknolojileri sayesinde elde edilen bu görüntülerin doğru ve etkin bir şekilde analiz edilmesi için gelişmiş görüntü işleme teknikleri ve yazılımlar gerekmektedir.

Tüm organlar ve özellikle de beyin dokusu yoğun lipit (doymuş, doymamış yağ) içerdiği için şeffaf değildir. Bazı kimyasallar yardımıyla bu dokuları lipitlerden arındırılarak şeffaf yapılabilmektedir. İşlemler sonrası dokulardaki antijen yapılar büyük oranda korunmaktadır. Böylelikle immünfloresan yöntemlerle organlar üzerindeki damar sistemi bütün olarak işaretlenmesi mümkün olmaktadır (Özkan vd., 2023). Beyin dokusu şeffaflaştırıldıktan sonraki adımda dokunun dijital ortama aktarılması işlemi vardır. Burada devreye gelişmiş mikroskop cihazları girmektedir. Bu cihazlardan biri floresan mikroskobudur.

Floresan mikroskopları, biyolojik ve kimyasal numunelerin detaylı incelenmesinde kullanılan özel optik mikroskop türlerindendir. Bu mikroskoplar, numunede bulunan floresan maddelerin spesifik dalga boylarında ışık yaymasını sağlayarak yüksek kontrastlı görüntüler elde eder. Floresan mikroskopları, özellikle hücrelerin, moleküllerin ve proteinlerin yapılarını ve işlevlerini incelemek için yaygın olarak kullanılır. Modern floresan mikroskoplar, çoklu floresan işaretleyicilerle çalışabilme ve üç boyutlu görüntüleme gibi özelliklerle gelişmiş analiz imkanı sunar.

Floresan mikroskopları incelenen şeffaflaştırılmış dokunun görüntüsünü katman katman kaydedilebilmektedir. Bu katmanlı görüntülere z-yığın görüntü olarak adlandırılmaktadır. Böylece sistem dokuyu iki boyutlu resimlerden oluşan z-yığın görüntülere dönüştürmüş olur.

Floresan mikroskobu ile alınmış bu görüntülerin otomatik olarak analiz edilmesi, araştırmacıların daha hızlı ve daha doğru sonuçlar elde etmesini sağlar. Bu

çalışmada, beyin damarlarının iki boyutlu (2B) mikroskop görüntülerinde dallanma ve uç noktalarının tespiti yapılmaktadır. Bu tez kapsamında görüntülerin analizinden, geleneksel görüntü işleme yöntemleri ve derin öğrenme yöntemleri ile çözüm ayrı ayrı gösterilen bir yazılım geliştirilmiştir. Geliştirilen program arka planında derin öğrenme tabanlı bir nesne tespiti yönteminin yapılabilirliğinin test edilmesi, geliştirilmesi ve bu yöntemin hiperparametre optimizasyonu yapılmıştır.

Klasik nesne tanıma yöntemlerinde, damarların dallanma ve bifurkasyon noktalarını bulmak ve sınıflandırmak için bazı yöntemler kullanılmıştır (Bhuiyan vd., 2012). Bu tez çalışması için, derin öğrenme modelinin eğitimi ve değerlendirilmesini kolaylaştırmak amacıyla, öncelikle laboratuvar deneylerinden elde edilen ham veri seti temizlenmiş ve doğruluğu sağlanmıştır. Veri seti daha sonra geliştirdiğimiz algoritma ile etiketlenerek, nesne tespiti algoritmasının eğitimi için zemin gerçek-doğru veri seti oluşturulmuştur. Derin öğrenme çerçevesiyle uyumluluğu artırmak için, etiketlenmiş veri seti daha sonra nesne tespiti görevleri için yaygın olarak kullanılan bir standart olan COCO formatına dönüştürülmüştür (Lin vd., 2014).

Önerilen yöntemimizin değerlendirilmesi bilimsel yöntemler ile yapılmış olup, beyin damar görüntülerinde dallanma ve uç noktalarını doğru bir şekilde tespit etme performansına odaklanılmıştır. Kesim üzerindeki birlik metriğini (IoU) kullanarak, tahmin edilen ve zemin doğruluğu notasyonları arasındaki örtüşme derecesini ölçen güvenilir bir ölçüm metodu kullanılmıştır. Çalışmamızın bulguları, derin öğrenme tabanlı yaklaşımın sağladığı büyük başarıyı göstermekte olup, tespit doğruluğu %90'ı aşmıştır. Bu sonuçlar, derin öğrenmenin beyin damar görüntülerinin analizini otomatikleştirmede değerli bir araç olarak potansiyelini vurgulamakta ve geleneksel görüntü işleme tekniklerine kıyasla gelişmiş verimlilik ve doğruluk sunmaktadır. Çalışmamızda başarılı derin öğrenme modelleri önerdikten sonra bunun hiperparametre optimizasyonu yapılmıştır. Bu tarz makine öğrenmesi metotlarında optimizasyon oldukça önemlidir. Nitekim optimizasyon çalışmasıyla birlikte başarı oranı %98 üzerine çıkarılmıştır. Böylelikle bu yöntemler ile bir çalışma yürütecek araştırmacılara optimum parametreler ile başlangıç yapabilmelerinin yolu açılmıştır.

Yapay zeka, makine öğrenmesi ya da derin öğrenme gibi bir çok akademik çalışma sadece çalışılmış olarak kalmakta kullanıcılara açılmamaktadır. Tez

kapsamında yapılan çalışmaların tamamı bir program haline getirilmiş ve kullanıcıların kullanabileceği bir arayüz sunulmuştur. Sonuç olarak tezde uçtan uca bir uygulama geliştirilmiştir. Bu da arka planda yapayzeka metotları kullanılarak nasıl bir uygulama geliştirilebileceğinin örneğini vermesi açısından değerlidir. Kısaca, problemin başlangıcından çözümler üretilerek, bulunan çözümler test edilerek en sonunda laborantların kullanabileceği bir yazılım ortaya çıkarılmıştır.

Tez Çalışmasının Literatüre Katkısı:

1) Derin Öğrenme ile Damar Analizi Alanına Katkı:

Bu tez çalışması, beyin damarlarının dallanma ve uç noktalarının tespitinde derin öğrenme tabanlı yöntemlerin etkinliğini ortaya koyarak, tıbbi görüntüleme alanında önemli bir yenilik sunmaktadır. Geleneksel görüntü işleme yöntemlerinin sınırlamalarını aşarak, derin öğrenme modellerinin daha yüksek doğruluk ve verimlilik sağladığı gösterilmiştir. Bu durum, beyin damar yapılarının daha hassas ve otomatik bir şekilde analiz edilmesine olanak tanıyarak, nörovasküler bozuklukların teşhis ve araştırmalarında yeni kapılar açmaktadır.

2) Büyük Ölçekli Tıbbi Görüntü Verilerinin Analizine Katkı:

Çalışma, büyük ölçekli tıbbi görüntü veri setlerinin otomatik analizine yönelik derin öğrenme yöntemlerinin nasıl uygulanabileceğini göstermektedir. Özellikle, veri setlerinin COCO formatına dönüştürülmesi ve nesne algılama modellerinin eğitilmesi sürecinde sunulan yaklaşımlar, tıbbi görüntüleme alanında büyük veri analizi için kullanılabilecek bir çerçeve sağlamaktadır. Bu çerçeve, araştırmacıların ve klinisyenlerin büyük veri kümelerini daha etkili bir şekilde işleyebilmesine ve analiz edebilmesine yardımcı olacaktır.

3) Nörovasküler Bozuklukların Teşhisine Yönelik Uygulamalara Katkı:

Tezde geliştirilen derin öğrenme tabanlı damar tespiti algoritması, nörovasküler bozuklukların teşhisi ve izlenmesi sürecinde önemli bir araç olarak kullanılabilir. Yüksek doğruluk oranı ve otomatik analiz yetenekleri sayesinde, bu yöntem klinik uygulamalarda hastalık ilerlemesini değerlendirmek ve tedavi etkinliğini izlemek için kullanılabilir. Bu, tıbbi teşhislerde daha hızlı ve güvenilir sonuçlar alınmasına katkıda bulunarak, nörovasküler araştırmalara da değerli bir katkı sunmaktadır.

4) Tıbbi Analiz Programlarının Geliştirilmesine Katkısı:

Bu tez çalışmasında geliştirilen program, iki boyutlu beyin damar görüntülerindeki dallanma ve uç noktalarının otomatik tespiti için özel olarak tasarlanmıştır. Python tabanlı bu yazılım, derin öğrenme algoritmalarını entegre ederek kullanıcı dostu bir arayüz sunmakta ve biyomedikal araştırmalarda, medikal çalışmalarda rahatlıkla kullanılabilecektir. Program, ham görüntü verilerini işleyerek damar yapılarının segmentasyonunu, iskeletleştirilmesini ve önemli noktaların tespitini otomatikleştirir. Bu süreçler sayesinde, araştırmacılar, beyin damarlarının yapısal özelliklerini hızlı ve hassas bir şekilde analiz edebilir, bu da nörovasküler hastalıkların anlaşılmasını ve tedavi süreçlerinin geliştirilmesini kolaylaştırır. Tüm bunları bir araya getiren program tıbbi analiz yazılımlarının nasıl geliştirilebileceğine bir örnek oluşturmaktadır.

Tez Çalışmasından Türetilen Yayınlar ve Projeler:

❖ Konferans yayınları:

- Kaya S., Dik S.Z., Kiraz B., Aydın M., Çamurcu A.Y. (2023) “BRAINVASCULYZER: 2B Beyin Damar Görüntü Analiz Programı”. MAS 18th International European Conference on Mathematics, Engineering, Natural & Medical Sciences 121-130.
- Kaya S., Kiraz B., Çamurcu A.Y. (2024) “Hyperparameter Optimization in Deep Learning-Based Object Detection of Branching and Endpoints on 2D Brain Vessel Images”. Akıllı Sistemlerde Yenilikler ve Uygulamaları Konferansı (ASYU). IEEE

❖ Dergi yayınları:

- Kaya S., Kiraz B., Çamurcu A.Y. (2024) “Branch and End Points Detection in Cerebral Vessels Images Using Deep Learning Object Detection Techniques”. Journal of Polytechnic. ESCI

❖ Projeler:

- Kaya S., Dik S.Z., Kiraz B., Çamurcu A.Y., Aydın, M. (2023) "Beyin Kan Damarlarının ve İlişkili Hücrelerin Makine Öğrenmesi Yaklaşımı ile Üç Boyutlu Analizi", 22022B1Ç01D, Bilimsel Araştırma Projesi (BAP), Fatih Sultan Mehmet Vakıf Üniversitesi.

Tez Organizasyonu:

Bu tez, beyin damarlarının mikroskop görüntülerinde dallanma ve uç noktalarının derin öğrenme tabanlı nesne tespiti algoritmaları ile bulunmasına ve bu yöntemler kullanılarak bir program geliştirilmesine yönelik çalışmaların detaylarını sunan altı ana bölümden oluşmaktadır.

Birinci bölümde, çalışmanın amacı, önemi, kapsamı ve literatür taraması ele alınmaktadır. Beyin damarlarının yapısal analizinin nöro görüntüleme alanındaki önemi vurgulanmakta ve bu alanda derin öğrenme tekniklerinin kullanımına dair genel bir bakış sunulmaktadır. Literatür taraması üç ana kısma ayrılmıştır, derin öğrenme ile nesne tespit çalışmaları, hiperparametre optimizasyonu üzerine çalışmalar ve damar analizi üzerine geliştirilen programlar.

İkinci Bölüm: görüntülerin ön işlemler ile temizlenmesi aşamaları kesin referans veri setini oluşturabilmek için dallanma ve uç noktaların görüntü işleme teknikleriyle bulunması açıklanmaktadır. Yapılan işlemler sonucunda tüm görüntülerin veri seti haline getirilmesi anlatılmıştır. Özetle beyin damarlarının iki boyutlu mikroskop görüntülerinde dallanma ve uç noktalarının tespiti için izlenen adımlar, görüntülerin hazırlanması, etiketlenmesi ve eğitim süreçleri açıklanmaktadır.

Üçüncü bölümde, derin öğrenme yöntemleri ile eğitim ve testlerin nasıl yapıldığı ve değerlendirme yöntemi, yapılan deneyleri ve kullanılan derin öğrenme modellerinin performansları, geleneksel yöntem ile derin öğrenme yöntemlerinin zaman karşılaştırması, hiperparametre optimizasyonu, değerlendirme açıklanmaktadır. Optimizasyon çalışması, deneyi ve sonuç değerlendirmesi bu bölümde yer almaktadır.

Dördüncü bölümde, yapılan çalışmayı kullanıcı odaklı bir yazılım haline dönüştürme süreci ve bu programın referans bir program ile karşılaştırmalı analizi açıklandı. Programın kullanımı da bu bölümde anlatılmaktadır. En son olarak program referans bir program ile karşılaştırmalı analizi yapılmıştır.

Beşinci Bölümde, çalışmanın genel sonuçları özetlenmekte ve beyin damarlarının analizi için geliştirilen derin öğrenme tabanlı yöntemin literatüre katkıları vurgulanmaktadır. Ayrıca, bu çalışma kapsamında elde edilen sonuçların gelecekteki araştırmalara ve uygulamalara yönelik öneriler sunulmaktadır.



BİRİNCİ BÖLÜM

1. LİTERATÜR TARAMASI

1.1. Derin Öğrenme Nesne Tanıma Metotları ile Damar Analizi

Tıbbi görüntüleme teknolojisindeki ilerlemeler, yüksek çözünürlüklü iki boyutlu beyin damar görüntülerinin elde edilmesini sağlamıştır (Kuş vd., 2023; Todorov vd., 2019; Zhang vd., 2018). Kan damarlarının doğru bir şekilde incelenmesi, hastalık nedeniyle damar yapısında meydana gelen değişikliklerin belirlenmesi için kritik öneme sahiptir. Bu analiz, hastalığın durumu, ilerlemesi veya olası tedavilerin etkinliği hakkında önemli bilgiler sağlayabilir (Bollmann vd., 2022; Zhang vd., 2018). Görüntülerdeki bu bilgi zenginliği, biyolojik çalışmalarda genellikle insanlar tarafından manuel olarak analiz edilemez (McGarry vd., 2022). Bu tür görüntüleri analiz etmenin kritik görevlerinden biri, damar yapısındaki dallanma ve uç noktalarının tespit edilmesidir; bu özellikler, teşhis ve araştırma açısından büyük bir değere sahiptir (Gu vd., 2019; Zudaire vd., 2011). Geleneksel olarak, bu süreç genellikle manuel müdahale gerektiren ve büyük ölçekli analizler için ölçeklenebilirlikten yoksun olan geleneksel görüntü işleme tekniklerine dayanmıştır. Geleneksel nesne tespit yöntemlerinin aksine, derin öğrenme, sinir ağlarının gücünü kullanarak ham verilerden otomatik olarak ilgili özellikleri belirleyebilir ve bu da damar yapılarını tespit etmede potansiyel olarak geliştirilmiş doğruluk ve verimlilik sunar. Ölçeklenebilirliği göz önünde bulundurmadığımızda ve eğitilebilir bir veri seti olmadığında, klasik nesne tespiti daha iyi bir yaklaşım olabilir. Bununla birlikte, bilgi miktarı arttıkça derin öğrenme yaklaşımları performans açısından görüntü işleme yaklaşımlarını geride bırakmıştır (Bhuiyan vd., 2012; Chamarty Anusha, 2018).

Derin öğrenmenin tıbbi görüntüleme alanında devrim niteliğinde bir değişim yarattığı açıktır. Özellikle nesne tespiti alanında, derin öğrenme tabanlı teknikler, tıbbi görüntülerde anatomik yapıları ve patolojileri tanımlama ve sınıflandırma yeteneklerini kanıtlamıştır. Bu gelişmeler, teşhis doğruluğunu artırmış ve klinik iş

akışlarını hızlandırarak hasta sonuçlarını iyileştirmiştir. Ayrıca, derin öğrenme modellerinin büyük veri kümelerinden karmaşık özellikleri otomatik olarak öğrenme yetenekleri, manuel özellik mühendisliği ihtiyacını azaltmış ve daha sağlam ve ölçeklenebilir çözümler sunmuştur (Chamarty Anusha, 2018; Livne vd., 2019).

Yapılan literatür çalışmasında derin öğrenme yöntemleri ile damar analizi çalışmalarının birçoğu bölütleme üzerine olduğu görülmektedir. Bir çalışmada, X-ışını koroner anjiyografide damar segmentasyonu için konvolüsyonel ve tekrarlayan katmanları birleştiren hibrit bir derin sinir ağı tanıtılmaktadır. Ağ, segmentasyon doğruluğunu artırmak için CNN'lerin uzamsal özelliklerinden ve RNN'lerin zamansal özelliklerinden yararlanmaktadır. Yazarlar, yöntemlerini geleneksel yaklaşımlarla karşılaştırmış ve ince damarları tespit etmede ve yanlış pozitifleri azaltmada önemli gelişmeler olduğunu göstermiştir (Yuan vd., 2020). Retina görüntülerinde damar segmentasyonu hakkındaki bir makalede U-Net, ResNet ve GAN tabanlı modeller gibi farklı mimarilerin güçlü ve zayıf yönlerini vurgulamakta ve retinal damar segmentasyonundaki uygulamalarını tartışmaktadır. Bu çalışmada, küçük damar boyutları ve düşük kontrast gibi retinal görüntülerle çalışmanın zorluklarını ve derin öğrenme modellerinin bu sorunları nasıl ele aldığını vurgulamaktadır (C. Chen vd., 2021). Yapılmış diğer bir çalışmada, tıbbi görüntü segmentasyonuna uygulanan derin öğrenme (DÖ) alanındaki son gelişmelere kapsamlı bir genel bakış sunmaktadır. DÖ tekniklerinin, özellikle de konvolüsyonel sinir ağlarının (CNN'ler), görüntü segmentasyon görevlerinin doğruluğunu ve verimliliğini nasıl önemli ölçüde geliştirdiğini vurgulamaktadır. Çalışma, tümör tespiti ve organ segmentasyonu gibi tıbbi görüntü segmentasyonunda yaygın olarak kullanılan çeşitli DL algoritmalarını, veri kümelerini ve ön işleme tekniklerini incelemektedir. Ayrıca, sınırlı etiketli veri ve sınıf dengesizliği de dahil olmak üzere bu alanda karşılaşılan zorlukları tartışmakta ve bu sorunları ele almak için gelecekteki araştırma yönlerine ilişkin içgörüler sunmaktadır (Rayed vd., 2024). Birçok bölütleme çalışması bulunmak ile birlikte Alvaro ve arkadaşlarının yaptığı çalışmada, el ile etiketlenmiş veriler kullanmadan retinal damar segmentasyonu için derin öğrenme tabanlı bir yaklaşım sunmaktadır. Çalışmada, manuel etiketlemeye ihtiyaç duyulmadan, multimodal görüntü verilerinden otomatik olarak elde edilen etiketler kullanılarak sinir ağlarının eğitilmesi amaçlanmaktadır. Retinografiden (göz dibi fotoğrafı) anjiyografi türevi

damar haritaları üretmek için bir sinir ağı eğitmişlerdir. Bu işlemde, anjiyografi görüntüleri basit görüntü işleme teknikleriyle damar haritalarına dönüştürülmekte ve bu haritalar sinir ağına etiket olarak verilmektedir. Bu sayede, etiketleme süreci otomatikleştirilerek insan müdahalesi olmadan büyük miktarda eğitim verisi elde edilebilmektedir (Hervella vd., 2019).

Dallanma ve uç noktalarının tespiti üzerine yapılan çalışmalar kısıtlı da olsa bulunmaktadır. Retinal görüntülerdeki damar birleşim noktalarını (bifurkasyon ve çapraz noktaları) tespit etmek ve sınıflandırmak için derin öğrenme tabanlı bir yöntem öneren bir çalışmada, iki aşamalı bir işlem hattı kullanılarak damar birleşim noktalarının doğru bir şekilde tespit edilmesi ve sınıflandırılması hedeflenmiştir. İlk aşamada, RCNN tabanlı Birleşim Öneri Ağı (JPN) kullanılarak olası birleşim noktaları tespit edilir ve ardından Birleşim İyileştirme Ağı (JRN) ile yanlış tespitler elenir. İkinci aşamada, tespit edilen birleşim noktaları bir sınıflandırma ağı kullanılarak çapraz veya bifurkasyon olarak sınıflandırılır. Çalışma, DRIVE ve IOSTAR veri setleri üzerinde test edilmiş ve en son yöntemlerden daha yüksek bir performans elde etmiştir. Özellikle, DRIVE veri setinde F1-skoru %70, IOSTAR veri setinde ise %60 olarak elde edilmiştir. Özetle, bu yöntem, retinal görüntülerdeki damar birleşim noktalarını doğrudan tespit ve sınıflandırmak için derin öğrenme tabanlı bir yaklaşım sunmakta ve diğer mevcut yöntemlere kıyasla daha iyi sonuçlar vermektedir (Zhao vd., 2020).

Bir makale, damar ağlarının nicel analizi için tasarlanmış bir hesaplama aracını sunmaktadır (Zudaire vd., 2011). Gelişmiş matematiksel modeller veya makine öğrenimi tekniklerinden yararlanan bu araç, damar yapılarını hassas bir şekilde ölçmeyi ve karakterize etmeyi sağlayarak araştırmacıların damar morfolojisini, fonksiyonunu ve patolojisini anlamalarına yardımcı olur. Bu araç, damar ağları tarafından kaplanan alan ve damar sayısı gibi çeşitli morfolojik ve mekansal parametreleri hesaplar. 2012 yılında yapılan bir çalışma, retinal damar görüntülerinde bifurkasyon ve dallanma noktalarını otomatik olarak tespit etmek ve sınıflandırmak için yenilikçi bir yöntem sunmuştur. Bu yaklaşım, vasküler ağ yapılarını doğru bir şekilde tanımlamak ve sınıflandırmak için morfolojik işlemleri yerel gradyan bilgisiyle birleştirmiştir (Bhuiyan vd., 2012). Başka bir çalışma, retinal vaskülerdeki bağlantıların tespitini kolaylaştırmak için çoklu görevli bir ağ mimarisi

kullanarak, aynı çerçevede birden fazla görevin aynı anda gerçekleştirilmesini sağlamıştır (Uslu & Bharath, 2018). Bu yaklaşım, retinal görüntülerde kritik özelliklerin tanımlanmasının verimliliğini ve doğruluğunu optimize ederek, hastalık teşhisi ve tedavi planlamasını kolaylaştırmıştır.

Todorov vd. tarafından yapılan çalışmada, fare beyninin tüm damar yapısını analiz etmek için makine öğrenimi yöntemlerini, özellikle konvolüsyonel sinir ağlarını uygulamıştır(Todorov vd., 2020). Bu çalışmada, fare beyninin tamamı üzerinde güvenilir bir segmentasyon sağlanmış ve karmaşık damar ağı uzunlukları ve bifurkasyon noktaları çıkarılmıştır. Fare beyin damarlarını immüno-fluoresans etiketleme için kapsamlı bir protokol sunulmuştur (Freitas-Andrade vd., 2022). Ardından, yazarlar iki veya üç boyutlu dijital görüntü işleme algoritmaları kullanarak damar yoğunluğu, dallanma ve tortioziteyi tarafsız bir şekilde ölçmüşlerdir. Çalışma, 2B veya 3B floresan görüntüler kullanılarak fare beyin endotel ağlarının tarafsız analizini yapmak için gelişmiş görüntü işleme algoritmalarını, makine öğrenimi ve bilgisayarla görme tekniklerini kullanmıştır.

McGarry vd. tarafından Beyin damarları üzerine yapılan bir çalışmada, "Vessel Metrics" yazılım aracı, konfokal görüntülemeye vasküler yapıların otomatik analizini gerçekleştirmek için Python tabanlı teknikler kullanmaktadır (McGarry vd., 2022). Bu uygulama, biyoloji ve tıp gibi çeşitli alanlarda araştırmaları kolaylaştırarak damar özelliklerinin etkin ve hassas bir şekilde ölçülmesini sağlar. Birçok derin öğrenme nesne tespiti algoritması bulunmaktadır (Chamarty Anusha, 2018).

1.1.1. Derin Öğrenme Çalışmalarında Kullanılan Modeller

Derin öğrenme yöntemleri, görüntü işlemede devrim yaratarak nesne tespiti gibi karmaşık problemler için etkili çözümler sunmuştur. Bu bağlamda, Convolutional Neural Networks (CNN'ler) ve türevleri, görsel verilerden anlamlı bilgiler çıkarmada kilit rol oynamaktadır. CNN tabanlı yöntemler, görüntüdeki nesnelerin konumunu ve türünü belirleyebilmek adına katmanlar aracılığıyla özellikler çıkarır ve bu özellikleri sınıflandırır. Nesne tespiti alanında kullanılan çeşitli CNN tabanlı yaklaşımlar, hız, doğruluk ve esneklik açısından farklı avantajlar sunmaktadır. R-CNN, YOLO, SSD ve Mask R-CNN gibi popüler modeller, farklı uygulama alanlarında yüksek performans göstermiştir. Bu bölümde, bu yöntemlerin

temel prensipleri, avantajları ve uygulama alanları kısa bilgiler olarak aşağıda anlatılmıştır.

Convolutional Neural Networks (CNNs):

Convolutional Neural Networks (CNN'ler), derin öğrenme tabanlı nesne tespiti yöntemlerinin temelini oluşturur. CNN'ler, görüntülerdeki nesneleri tespit etmek için konvolüsyon katmanları kullanır. Bu katmanlar, görüntünün farklı bölgelerinden özellikler çıkarır ve bu özellikleri sınıflandırarak nesneleri tespit eder. CNN tabanlı yöntemler, nesne tespiti alanında büyük başarı elde etmiş ve birçok modern modelin temel yapı taşı olmuştur (Lecun vd., 1998).

Region-Based Convolutional Neural Networks (R-CNN):

R-CNN, nesne tespiti için bölge tabanlı bir yaklaşım sunar. Bu yöntem, görüntüdeki belirli bölgeleri analiz eder ve her bölgeyi ayrı ayrı sınıflandırır. R-CNN'in geliştirilmiş versiyonları olan Fast R-CNN ve Faster R-CNN, bu süreci hızlandırarak tespit doğruluğunu artırmıştır (Girshick vd., 2013). Faster R-CNN, bölge öneri ağıları (Region Proposal Networks, RPN) kullanarak tespit sürecini hızlandırır ve gerçek zamanlı uygulamalarda kullanılabilir hale getirir.

You Only Look Once (YOLO):

YOLO, nesne tespitinde hız ve doğruluk açısından öne çıkan bir yöntemdir. Bu yöntem, görüntüyü tek bir geçişte analiz eder ve bu sayede hızlı sonuçlar elde edilir. YOLO'nun ana avantajı, nesneleri gerçek zamanlı olarak tespit edebilmesidir (Redmon vd., 2016). YOLO'nun çeşitli versiyonları olan YOLOv3, YOLOv4 ve YOLOv5, her geçen yıl geliştirilerek daha yüksek doğruluk ve performans sunmaktadır.

Single Shot Multibox Detector (SSD):

SSD, nesne tespitinde hızlı ve etkili bir yöntem sunar. YOLO'ya benzer şekilde, tek bir geçişte görüntüdeki nesneleri tespit eder. Ancak, SSD özellikle küçük nesneleri tespit etme konusunda daha başarılıdır (W. Liu vd., 2015). Bu yöntem, farklı çözünürlüklerde özellik haritaları kullanarak nesneleri tespit eder ve bu sayede daha esnek bir yapı sunar.

Mask R-CNN:

Mask R-CNN, R-CNN yönteminin bir uzantısı olarak nesneleri tespit etmenin yanı sıra her nesne için bir maske oluşturur. Bu yöntem, nesnelerin segmentasyon

gerektirdiği durumlarda kullanılır (K. He vd., 2017). Mask R-CNN, özellikle tıbbi görüntüleme ve otonom araçlar gibi uygulamalarda yaygın olarak kullanılmaktadır.

1.1.2. Derin Öğrenme Çalışmalarında Kullanılan Kütüphaneler

Derin öğrenme tabanlı nesne tespiti, çeşitli kütüphane ve araçlarla daha erişilebilir hale gelmiştir. Bu kütüphaneler, araştırmacıların ve mühendislerin karmaşık modelleri hızlı bir şekilde geliştirmesine ve uygulamalarına olanak tanır. TensorFlow, PyTorch, OpenCV, Darknet ve MMDetection gibi önde gelen kütüphaneler, nesne tespiti alanında güçlü çözümler sunmaktadır. Her biri, farklı özellikler ve avantajlar sunarak, kullanıcıların spesifik gereksinimlerine göre en uygun aracı seçmesine imkan tanımaktadır. Bu bölümde, nesne tespiti için yaygın olarak kullanılan bu kütüphanelerin temel özellikleri, kullanım alanları ve sundukları olanaklar detaylandırılmıştır.

TensorFlow:

TensorFlow, Google tarafından geliştirilen ve geniş bir kullanıcı kitlesine sahip olan bir derin öğrenme kütüphanesidir. TensorFlow'un Object Detection API'si, nesne tespiti için önceden eğitilmiş modeller ve araçlar sunar (Abadi vd., 2016). Bu API, kullanıcıların hızlı bir şekilde nesne tespiti modelleri oluşturmaya ve bu modelleri çeşitli uygulamalarda kullanmasına olanak tanır.

PyTorch ve Detectron2:

PyTorch, Facebook tarafından geliştirilen ve esnek yapısı sayesinde popüler hale gelen bir derin öğrenme kütüphanesidir. PyTorch üzerinde çalışan Detectron2, nesne tespiti için güçlü bir kütüphanedir ve Mask R-CNN, Faster R-CNN gibi ileri seviye modelleri destekler (Wu vd., 2019). Detectron2, modüler yapısı ve kullanıcı dostu API'si sayesinde geniş bir kullanım alanına sahiptir. Araştırma ve endüstri projelerinde yaygın olarak kullanılmaktadır.

OpenCV:

OpenCV, görüntü işleme ve bilgisayarla görme alanında kullanılan popüler bir kütüphanedir. OpenCV, derin öğrenme modelleriyle entegre çalışarak nesne tespiti uygulamalarında da etkili bir şekilde kullanılabilir (Bradski, 2000). Bu kütüphane, nesne tespiti için hızlı ve optimize edilmiş araçlar sunar.

Darknet:

Darknet, YOLO algoritmasının geliştirilmesi için kullanılan bir kütüphanedir. C dilinde yazılmış olan Darknet, hız ve performans açısından avantaj sağlar ve YOLO'nun farklı versiyonlarının uygulanmasına olanak tanır (Redmon & Farhadi, 2018). Bu kütüphane, özellikle gerçek zamanlı uygulamalar için idealdir.

MMDetection:

MMDetection, OpenMMLab tarafından geliştirilen bir nesne tespiti kütüphanesidir. PyTorch tabanlı olan bu kütüphane, çeşitli model ve yöntemlerin entegre edilebildiği modüler bir yapı sunar (K. Chen vd., 2019). MMDetection, geniş bir model yelpazesi sunarak araştırma ve endüstri projelerinde yaygın olarak kullanılmaktadır.

1.2. Derin Öğrenme Metotlarında Hiperparametre Optimizasyonu

Nörovasküler çalışmalarda derin öğrenmenin uygulanması, beyin damarlarının tespiti ve analizinde daha doğru ve verimli araçlar sağlayarak alanı kökten değiştirmiştir. Geleneksel yöntemler genellikle manuel segmentasyona ve klasik görüntü işleme tekniklerine dayanırken, derin öğrenme, özellikle evrişimli sinir ağları (CNN'ler), bu süreçleri yüksek hassasiyetle otomatikleştirme konusunda önemli bir potansiyel göstermiştir. Bununla birlikte, hiperparametre optimizasyonu, bu modellerin performansını en üst düzeye çıkarmak için kritik bir rol oynamaktadır (Bischi vd., 2023).

Hiperparametre optimizasyonu, modelin genel performansını artırmak ve optimize etmek için çeşitli model parametrelerinin titizlikle ayarlanmasını içerir, bu da model karmaşıklığı, eğitim verimliliği ve tahmin doğruluğu arasında denge sağlar. Doğru ayarlama, modelin eğitim verilerinden genel sonuçlara ulaşma yeteneğini önemli ölçüde artırabilir, aşırı öğrenmeyi azaltabilir ve güvenilirliği artırabilir. Ayrıca, hesaplama kaynaklarının verimli bir şekilde kullanılmasını sağlar ve modelin performansını düşürmeden daha hızlı eğitim sürelerine olanak tanır. Hiperparametreler üzerinde ince ayar yaparak, araştırmacılar modelleri belirli veri kümelerine ve görevlere uyarlayabilir, bu da gerçek dünya tıbbi senaryolarında uygulamalarını ve başarılarını artırır. Bu süreç, eğitim verilerinden genelleme yeteneğini geliştirmek ve aşırı uyumu azaltmak için büyük önem taşır. Bu bağlamda,

Bayes optimizasyonu, grid arama ve rastgele arama gibi gelişmiş teknikler, özellikle küçük anatomik yapıları tespit etmek gibi spesifik veri setleri için modelleri uyarlamak için önemlidir (Snoek vd., 2012a).

Hiperparametre optimizasyonu, derin öğrenmede model performansını doğrudan etkileyen kritik bir süreçtir. Eğitim sırasında öğrenilen model parametrelerinin aksine, hiperparametreler öğrenme sürecinden önce belirlenir ve modelin nasıl eğitileceğini yönetir. Bu kısımda, derin öğrenmede hiperparametre optimizasyonu için çeşitli yöntemleri ve bunların avantajlarını, sınırlamalarını ve uygulamaları açıklanmaktadır.

Rastgele Arama ve Izgara Arama:

Rastgele arama ve ızgara arama, hiperparametre ayarlama için iki geleneksel yöntemdir. Izgara araması, önceden tanımlanmış bir parametre uzayı üzerinde, hiperparametrelerin olası tüm kombinasyonlarını değerlendiren kapsamlı bir aramayı içerir (Bergstra vd., 2012). Etkili olmakla birlikte, ızgara araması özellikle büyük modeller için hesaplama açısından işlem süresi uzundur. Rastgele arama hiperparametre uzayından rastgele örnekleme yapar. Rastgele aramanın özellikle yüksek boyutlu uzaylarda verimlilik açısından ızgara aramasından daha iyi performans gösterebileceğini göstermiştir (Bergstra vd., 2012).

Bayes Optimizasyonu:

Bayes optimizasyonu, amaç fonksiyonunun olasılıksal bir modelini oluşturan ve bunu hiperparametreleri seçmek için kullanan daha sofistike bir yaklaşımdır (Snoek vd., 2012). Yöntem tipik olarak amaç fonksiyonunu modellemek için Gauss Süreçlerini (GP'ler) ve değerlendirilecek bir sonraki noktaya karar vermek için edinim fonksiyonlarını içerir. Bayesian optimizasyonu çeşitli derin öğrenme uygulamalarında başarı göstermiş, ızgara ve rastgele aramaya göre daha az değerlendirme ile daha iyi performans sağlamıştır.

Gradyan Tabanlı Yöntemler:

Hypergradient Descent (Baydin vd., 2017) gibi gradyan tabanlı yöntemler, hiperparametrelerle ilgili doğrulama hatasının gradyanına yaklaşarak hiperparametreleri optimize eder. Bu yöntemler özellikle öğrenme oranları gibi sürekli hiperparametreleri optimize etmek için kullanışlıdır. Bununla birlikte, farklılaştırılamayan veya ayrık hiperparametrelerle mücadele edebilirler.

Hiper bant:

Hyperband, hiperparametre optimizasyonu için hesaplama kaynaklarını verimli bir şekilde tahsis eden yeni bir yaklaşımdır (Li vd., 2018). Eğitim sürecinin başlarında gelecek vaat eden konfigürasyonları belirlemek için rastgele aramayı erken durdurma ile birleştirerek kapsamlı hesaplama ihtiyacını azaltır. Hiper bant, özellikle eğitimin pahalı olduğu ve erken performansın nihai sonuçları öngördüğü modeller için kullanışlıdır.

Hiperparametre optimizasyonu, problemin karmaşıklığına ve hesaplama kaynaklarına bağlı olarak çeşitli yöntemlerin mevcut olduğu derin öğrenmede önemli bir zorluk olmaya devam etmektedir. Izgara ve rastgele arama gibi geleneksel yöntemler hala yaygın olarak kullanılırken, Bayesian optimizasyonu, gradyan tabanlı yöntemler ve popülasyon tabanlı yöntemler gibi gelişmiş teknikler gelişmiş verimlilik ve performans sunar. Yöntem seçimi, hiperparametrelerin doğası ve mevcut hesaplama kaynakları dahil olmak üzere uygulamanın özel gereksinimlerine bağlıdır.

Tez kapsamında optimizasyon işlemi için optuna kütüphanesi tercih edilmiştir. Optuna, makine öğrenmesi ve yapay zeka projelerinde hiperparametre optimizasyonu için kullanılan güçlü ve esnek bir araçtır. Hiperparametre optimizasyonu, bir modelin performansını önemli ölçüde artırabilen ancak manuel olarak yapılması oldukça zahmetli olan bir süreçtir. Optuna, bu süreci otomatikleştirerek araştırmacılara ve veri bilimcilerine zaman kazandırır (Akiba vd., 2019).

Optuna'nın temel işlevi, belirli bir model için en iyi hiperparametreleri bulmaktır. Bunu yaparken, kullanıcılar arama alanlarını dinamik olarak tanımlayabilir ve Optuna'nın sunduğu çeşitli optimizasyon algoritmalarından yararlanabilirler. Optuna'nın en dikkat çekici özelliklerinden biri, define-by-run yaklaşımı ile kullanıcıların optimizasyon sürecini koşarken hiperparametre arama alanlarını esnek bir şekilde tanımlamalarına olanak tanır.

Optuna, özellikle büyük veri setleri ve karmaşık modeller üzerinde çalışan araştırmacılar için idealdir. Tree-structured Parzen Estimator (TPE) ve CMA-ES gibi sofistike optimizasyon algoritmalarını desteklerken, düşük performanslı denemeleri

erken durdurma mekanizması olan budama ile kaynakları verimli kullanır. Optuna, birçok akademik çalışma tarafından hiperparametre optimizasyonu için tercih edilmektedir. Bu çalışmalar, genellikle derin öğrenme modellerinin performansını artırmak için Optuna'nın sağladığı esnekliği ve gücü vurgular. Yapayzeka ve makine öğrenmesi alanındaki literatürde, Optuna kullanan çalışmalar vardır (Akiba vd., 2019; Hadiani, 2023; Rani vd., 2024; Srinivas & Katarya, 2022).

1.3. Damar Analizi İçin Geliştirilmiş Programlar

Gelişmiş mikroskopi teknikleri, vasküler ağların 2 boyutlu görüntülerinin elde edilmesine olanak sağlamaktadır. Bu görüntüler, damarların doğası ve görüntüleme teknolojisi nedeniyle genellikle karmaşık ve gürültülü olsa da manuel olarak analiz edilmesi zordur. Bunu ele almak için, 2B serebrovasküler görüntü analiz programları, beyin vasküler hastalıklarının teşhisi ve değerlendirilmesi için umut verici araçlar olarak ortaya çıkmıştır. Bu programlar, görüntü işleme ve/veya makine öğrenimi tekniklerini birleştirerek biyomedikal görüntüleri analiz etmek ve yorumlamak için tasarlanmıştır ve uzmanlara hastalık tespiti için hızlı ve doğru bir araç sağlar. Vasküler yapıların analizi, özellikle kanser ve vasküler bozukluklar gibi hastalıkların anlaşılmasında tıbbi araştırmaların kritik bir bileşenidir. Vasküler analiz araçları hem görüntüleme teknolojisindeki hem de hesaplama gücündeki ilerlemelerden yararlanarak yıllar içinde önemli ölçüde gelişmiştir.

Bu amaçla geliştirilen en eski araçlardan biri, görüntü analizi yoluyla in vitro anjiyogenezin sağlam bir şekilde ölçülmesi için tasarlanmış MATLAB tabanlı bir program olan Angioquant'tır(Niemistö vd., 2005). Bu araç özellikle deneysel ortamlarda yeni kan damarlarının oluşumunu analiz etmede etkili olmuştur.

2011 yılında iki önemli program tanıtılmıştır: Angiotool ve RAVE. Java ile geliştirilen Angiotool, vasküler ağların nicel analizi için tasarlanmıştır (Zudaire vd., 2011). Araştırmacıların damar uzunluğu, dallanma noktaları ve damar alanı gibi parametreleri yarı otomatik bir şekilde ölçmelerine olanak sağladı. RAVE veya Damar Elemanlarının Hızlı Analizi, aynı yıl tanıtılan bir başka araçtı. Fizyolojik, patolojik ve tümör anjiyogenezini incelemek için kolaylaştırılmış bir yaklaşım sunmuştur (Seaman vd., 2011).

Fiji/ImageJ: ImageJ tabanlı açık kaynaklı bir görüntü işleme paketi olan Fiji, vasküler analiz de dahil olmak üzere çeşitli biyomedikal görüntü analizi görevleri için yaygın olarak kullanılmaktadır. Fiji'nin eklenti ekosistemi, kullanıcıların iskeletleştirme, damar izleme ve segmentasyon gibi karmaşık görevleri yerine getirmelerine olanak tanır. Fiji'yi özel eklentilerle genişletmenin kolaylığı, onu araştırmacılar arasında popüler bir seçim haline getirmiştir (Schindelin vd., 2012). Fiji için “Basit Nörit İzleyici Eklentisi” çalışması bulunmaktadır. Öncelikle nöronları izlemek için tasarlanmış olsa da bu eklenti 2B vasküler analiz için de kullanılabilir. Uzunluk ve dallanma noktaları gibi faydalı ölçütler sağlayarak 2B görüntülerde tübüler yapıların manuel ve yarı otomatik olarak izlenmesini sağlar. Bu araç, basitliği ve Fiji platformuna entegrasyonu nedeniyle (Longair vd., 2011).

Daha yakın zamanda, 2020 yılında tanıtılan bir program olan REAVER, yüksek çözünürlüklü vasküler ağ görüntülerinin analizini geliştirmeyi amaçlamıştır. Corliss ve arkadaşları tarafından geliştirilen REAVER, yüksek çözünürlüklü görüntülerde karmaşık vasküler yapıları analiz eden araştırmacılar için daha iyi doğruluk ve kullanım kolaylığı sunmaya odaklanmıştır (Corliss vd., 2020).

Bahsedilen programlar bazı özellikleri paylaşmakla birlikte damar uzunlukları, dallanma noktaları, uç noktalar ve yoğunluk gibi metrikleri analiz ederek araştırmacılara değerli bilgiler sağlarlar. Bu programlar güçlü olmakla birlikte, genellikle kurulum ve kullanımda karmaşıklıklarla birlikte gelir ve bu da bazı araştırmacılar için daha az erişilebilir hale getirir. Bu nedenle, ayrıntılı vasküler çalışmalar için gereken analiz derinliğini sunmaya devam eden daha kullanımı kolay olan araçlar geliştirmek için devam eden bir çaba vardır.

Bu tez kapsamında, serebrovasküler görüntülerin otomatik karakterizasyonu için Python'da geliştirilen bir 2B beyin damarı görüntü analiz programı geliştirildi. Bu yazılım, çoklu foton mikroskobundan elde edilen 2B beyin damarı görüntülerini işlemek için tasarlandı ve analiz için çeşitli görüntü işleme teknikleri kullandı. Tez kapsamında geliştirilen derin öğrenme modelleri yazılımın içerisine gömüldü program ile çalışabilir hale getirildi.

İKİNCİ BÖLÜM

2. GÖRÜNTÜLERE ÖN İŞLEME YAPILMASI VE VERİ SETİ OLUŞTURULMASI

Bu bölümde, floresan mikroskop ile elde edilen damar görüntülerinin analizi, ön işleme süreci ve veri seti haline getirilmesi ele alınmıştır. Bu kısmın amacı, derin öğrenme çalışması yapılmadan önce veri setlerinin nasıl hazırlandığını çalışmanın tekrar edilebilirliği açısından araştırmacılara sunmaktır. Mevcut birçok çalışmada hazır veri setleri kullanılırken çalışmamızda kendi veri setimizi oluşturmamız ve oluşturma yönteminin ayrıntılı açıklanması iyi bir katkı olarak görülmektedir.

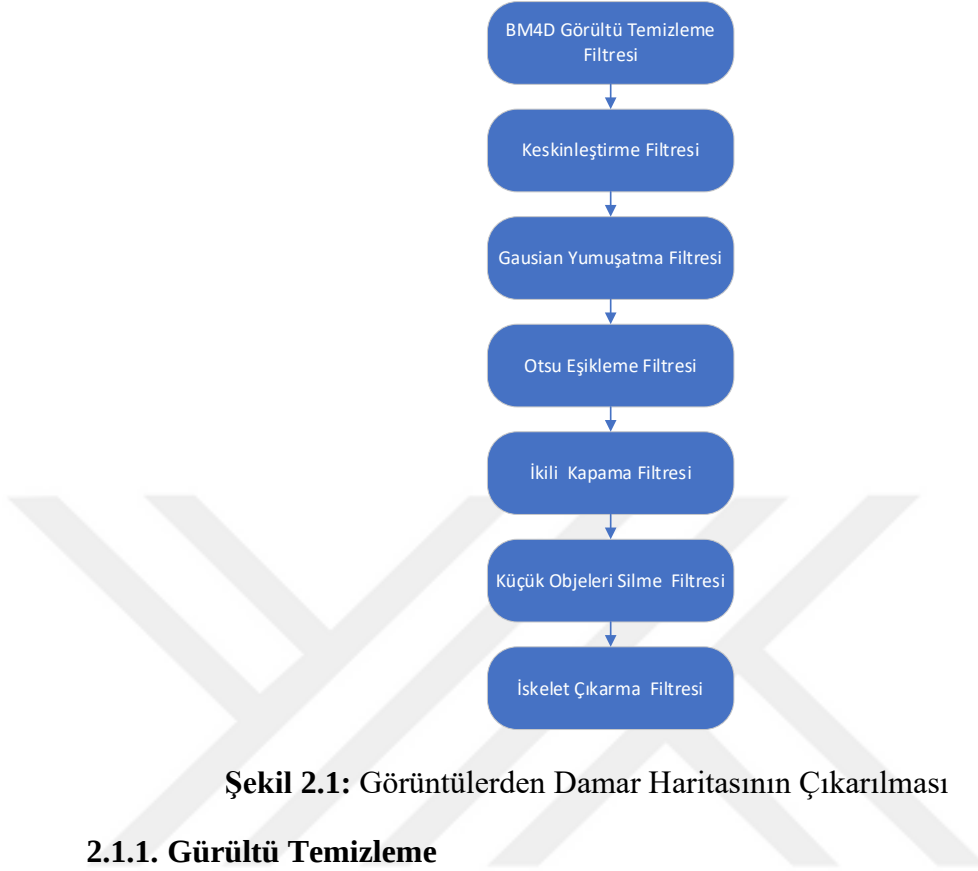
İlk önce damar yollarının doğru bir şekilde çıkarılması için gerekli olan adımları detaylandırmaktır. Bu doğrultuda, görüntülerin gürültüden arındırılması, keskinleştirilmesi, yumuşatılması ve damar iskeletinin çıkarılması gibi temel işlemlerin uygulanışı anlatılmıştır.

Sonraki başlıkta damar analizi çalışmalarının nasıl yapıldığı anlatılmış ve ardından damar yapılarından ve iskelet üzerinden dallanma ve uç noktalarının bulunması işlemlerinin geleneksel görüntü işleme yöntemleri ile nasıl yapıldığı detaylandırılmıştır.

Son olarak tespit edilen dallanma ve uç noktalarını derin öğrenme eğitimleri ve testlerinde kullanılabilecek bir formata dönüştürülmesi açıklanarak, tercih edilen formatın neden, nasıl seçildiği ve dönüşümün nasıl yapıldığı ayrıntılı olarak gösterilmiştir.

2.1. Görüntü Analizi ve Ön İşleme Çalışması

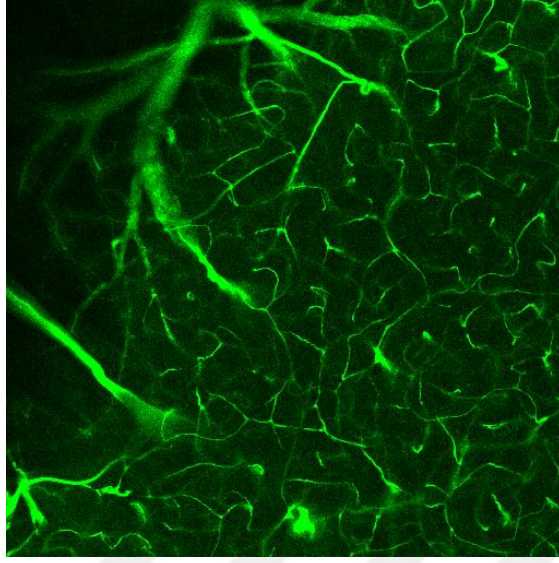
Floresan mikroskoptan çıkan görüntülerden gerekli damar yollarının çıkarılması ve tezde izlenen görüntünün temizlenmesi adımları **Şekil 2.1**'da görüntüden damar iskeletinin çıkarılması adımları gösterilmiştir.



2.1.1. Gürültü Temizleme

Çeşitli etmenler ile görüntü üzerinde oluşabilecek bozulmalar, gürültü olarak tanımlanabilir. Gürültü, görüntüyü alan lens, görüntü alınan ortam ve görüntüsü alınan nesne kaynaklı fiziksel olabilmekte ya da görüntü sayısallaştırılırken oluşabilmektedir. Literatürde birçok çeşit gürültü tipi tanımlanmıştır (Jebur vd., 2024). Gürültünün görüntüden çıkarılabilmesi için görüntü üzerindeki gürültü hakkında bilgi sahibi olunması gerekir. Görüntü üzerindeki gürültü çeşidi tespit edildikten sonra uygun teknikler ile belirli dereceye kadar temizleme ya da düşürme işlemi yapılabilir (Lone & Siddiqui, 2018).

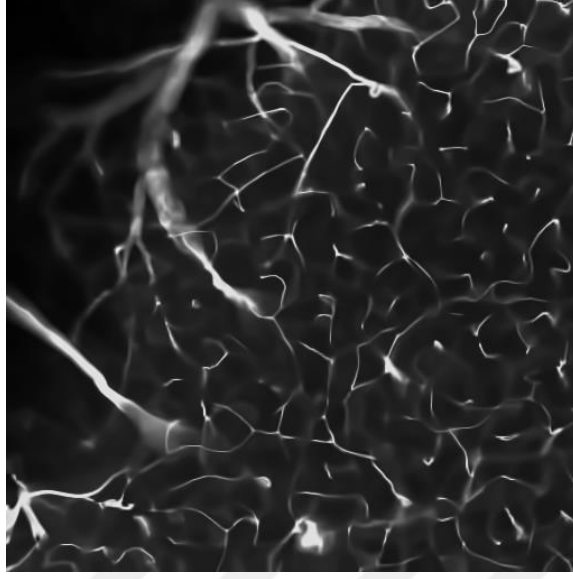
Bu tezde temel alınan görüntülerdeki damar izleri dışında her şey gürültü sayılmaktadır. Bundan dolayı görüntü üzerindeki damar izleri dışında her bilginin silinmesi hedeflenmiştir. **Şekil 2.2**'de bu tez kapsamında kullanılacak olan ham bir görüntü örneği gösterilmiştir.



Şekil 2.2: İşlenmemiş Görüntü

Görüntüler incelendiğinde görüntüler üzerinde rasgele gürültü olduğu tespit edilmiştir. Rasgele gürültülerin temizlenmesi periyodik gürültülere göre daha zordur ve referanslı olarak temizlenebilmektedir (İlesanmi & İlesanmi, 2021). Buradaki referans daha önceden işaretlenmiş veriler olabilirken, görüntü kendi içerisindeki referansları da kullanabilmektedir. Mevcut veri setlerimizde işaretlemiş bir referans verimiz bulunmamaktadır. Bu sebeple çalışmada, iç referans kullanan BM3D yöntemi tercih edilmiştir.

BM3D (Block-matching and 3D filtering algorithm) en güncel görüntü temizleme algoritmalarından biridir. Mevcut diğer algoritmalara kıyasla daha iyi gürültü giderme sonuçları elde etme kapasitesine sahiptir. Önerilen metotta görüntü bloklara bölünerek birbirlerine referans olmaları sağlanmaktadır. Bloklardaki benzerlikler kullanılarak görüntüdeki gürültüler giderilmektedir (Dabov vd., 2006). **Şekil 2.3**'te BM3D yöntemi ile gürültüden arındırıldıktan sonra elde edilen görüntü gösterilmiştir.



Şekil 2.3: Gürültüden Arındırılmış Görüntü

2.1.2. Görüntü Keskinleştirme

Görüntü işleme alanında keskinleştirme, görüntünün netliğini artırmak amacıyla sıklıkla kullanılan bir tekniktir. Keskinleştirme işlemi, görüntüdeki ince detayları belirgin hale getirir, kenarları vurgular ve bulanıklığı azaltarak görüntünün daha net ve anlaşılır olmasını sağlar. Özellikle nesne tanıma, kenar tespiti ve detay analizi gibi alanlarda keskinleştirme önemli bir rol oynar (Gonzalez & Woods, 2006a; Marr & Hildreth, 1980).

Keskinleştirme işlemi, genellikle düşük kontrastlı veya bulanık görüntülerde uygulanır. Bunun başlıca nedenleri şunlardır:

Detayların Belirginleştirilmesi: Görüntüdeki ince detayların ortaya çıkarılması ve nesnelerin daha belirgin hale getirilmesi amacıyla keskinleştirme yapılır. Bu, özellikle tıbbi görüntüleme, uydu görüntüleri ve mikroskopik görüntüleme gibi alanlarda önemlidir.

Kenar Vurgulama: Kenarlar, görüntüdeki nesnelerin sınırlarını belirler. Keskinleştirme işlemi, kenarların daha belirgin hale gelmesini sağlayarak nesnelerin tanınmasını kolaylaştırır.

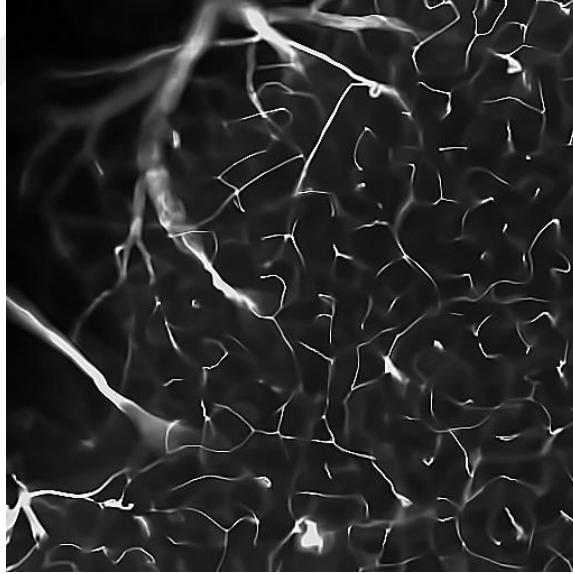
Bulanıklığın Giderilmesi: Çeşitli nedenlerle (örneğin, hareket bulanıklığı, odak dışı görüntüleme) bulanık hale gelen görüntüler, keskinleştirme teknikleri ile daha net hale getirilebilir.

Algısal İyileştirme: İnsan gözü, keskin ve net görüntüleri daha kolay algılar. Bu nedenle, görüntülerin görsel kalitesini artırmak amacıyla keskinleştirme yapılır.

Tez kapsamında ele alınan görüntü veri seti ne kadar gürültüden arındırılrsa da damar görüntülerinin arka-plan görüntüsünden biraz daha ayrışmasını sağlamak için bir kere “Sharpen” morfolojik filtresinden geçirilmiştir. Bu, özellikle damar ve arka plan gibi kontrast farkı düşük alanlarda, sınırların daha net ayrışmasını ve analiz için daha uygun bir görüntü elde edilmesini mümkün kılar. Kullanılan filtrenin çekirdeği **Denklem (2.1)**’de gösterilmiştir.

$$A = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 7 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad (2.1)$$

Verilmiş olan filtre uzaysal bir şekilde görüntüye uygulanmıştır. **Şekil 2.4** keskinleştirme filtresi uygulanmış görüntüyü göstermektedir.



Şekil 2.4: Keskinleştirme Filtresi Uygulanmış Görüntü

Bu keskinleştirme işlemi, damar yapılarının daha belirgin hale getirilmesini sağlamış ve görüntü üzerinde yapılan analizlerin doğruluğunu artırmıştır. Keskinleştirme filtresi, damar sınırlarının netleştirilmesine katkıda bulunarak, dallanma noktalarının ve uç noktalarının daha doğru tespit edilmesini kolaylaştırmıştır. Böylece, damar analizi sırasında elde edilen sonuçların

güvenilirliği artmıştır. Yapılan bu işlemin, tez kapsamında gerçekleştirilen diğer görüntü işleme adımlarına olumlu bir katkı sağladığı gözlemlenmiştir.

2.1.3. Görüntü Yumuşatma

Görüntü işleme alanında yumuşatma, bir görüntünün pürüzsüzleştirilmesi ve istenmeyen gürültülerin azaltılması amacıyla kullanılan temel tekniklerden biridir. Yumuşatma işlemi, özellikle görüntüdeki yüksek frekanslı bileşenleri (ani değişim gösteren alanlar) baskılayarak, daha homojen bir görüntü elde etmeye çalışır. Bu, belirli nesnelerin daha belirgin hale gelmesini ve görüntüdeki gürültülerin (rastgele sapmaların) etkisinin azaltılmasını sağlar (C. Tomasi & R. Manduchi, 1998; Gonzalez & Woods, 2006a)

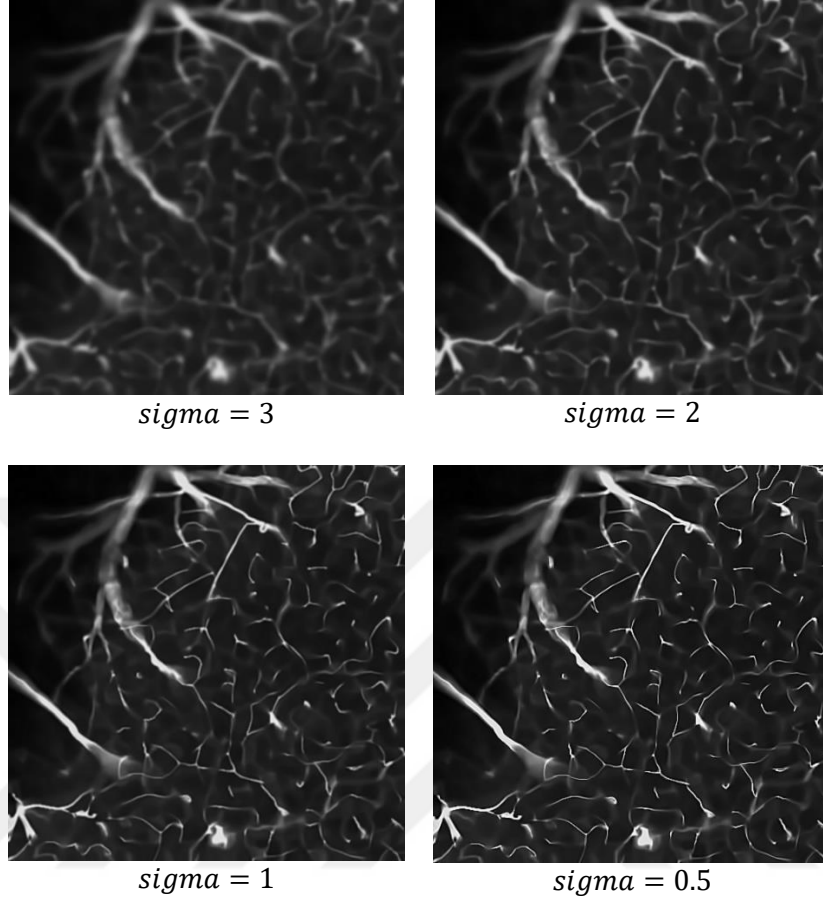
Yumuşatma işlemi, genellikle aşağıdaki nedenlerle uygulanır:

Gürültü Azaltma: Yumuşatma, görüntüdeki rastgele gürültüleri azaltarak, daha temiz bir görüntü elde etmeyi sağlar. Bu, özellikle düşük ışık koşullarında veya görüntüleme cihazlarındaki hatalar nedeniyle oluşan gürültülerin giderilmesi için önemlidir.

Detayların Yumuşatılması: Yumuşatma işlemi, ince detayları ve keskin kenarları yumuşatarak, görüntünün daha pürüzsüz ve doğal görünmesini sağlar. Bu, özellikle portre fotoğrafları veya grafik tasarımları gibi estetik amaçlarla yapılan görüntü işlemlerinde yaygındır.

Kenarlıkların Yumuşatılması: Yumuşatma, kenarlıkların daha az belirgin olmasını sağlayarak, keskin geçişlerin yumuşatılmasına yardımcı olur. Bu, özellikle görüntüdeki keskin geçişlerin rahatsız edici olduğu durumlarda tercih edilir.

Görüntü keskinleştirme ile görüntüdeki nesneler ön plana çıkarken aşırı ön plana çıkan kesimleri normalleştirmek için görüntüye “Gaussian Blur” uygulanır. Gauss filtresinde varyant değeri gauss filtresinin eğimini belirtir, eğim azaldıkça yumuşatma derecesi artar (Gonzalez & Woods, 2006a). **Şekil 2.5** çalışma kapsamında yaptığımız aynı damar görüntüsüne farklı sigma değerleri ile gauss filtresi uygulamasının çıktısıdır. Görüntüler arasında en iyi ve yeterli gördüğümüz 0.5 değerli yumuşatma işlemi ile devam edilmiştir.



Şekil 2.5: Gauss Filtresi Uygulanmış Görüntüler

Tıbbi görüntüleme, özellikle damar yapılarının analizi gibi kritik uygulamalarda hem detayların korunmasını hem de gürültülerin azaltılmasını gerektirir. Gauss filtresi gibi yumuşatma teknikleri, bu tip görüntülerde düşük seviyeli gürültüleri azaltarak, damar yapılarının daha net bir şekilde ortaya çıkmasını sağlar. Farklı sigma değerleri ile yapılan testlerde, aşırı yumuşatmanın damar detaylarını kaybetme riski taşıdığı görülmüştür. Bu nedenle, uygun bir sigma değeri seçmek, hem gürültünün etkili bir şekilde azaltılmasını hem de damarların doğru şekilde segmentasyonunu sağlamak için kritik öneme sahiptir. Özellikle 0.5 sigma değeri, bu dengeyi korumak adına en uygun seçenek olarak belirlenmiştir.

2.1.4. Görüntü Eşikleme

Eşikleme, görüntü işleme alanında en basit ve en yaygın kullanılan yöntemlerden biridir. Eşikleme işlemi, bir görüntüyü belirli bir eşik değeri kullanarak iki veya daha fazla bölgeye ayırmayı amaçlar. Görüntü eşikleme (Image

Thresholding) görüntüyü iki değer arasına çekme işlemidir. Eşiklenmiş görüntüdeki pikselleri 0 ya da 255 değerlerini alırlar. Eşikleme işleminde görüntünün histogramı çıkarıldıktan sonra belirli piksel değeri altındaki değerleri 0 üsteki değerleri 255 verilerek görüntü ikili hale getirilir. Literatürde birçok görüntü eşikleme yöntemi bulunmaktadır(Gonzalez & Woods, 2006b). Bu teknik, görüntüdeki önemli nesnelerin veya yapıların belirgin hale getirilmesini ve görüntü analizinin daha kolay yapılmasını sağlamaktadır. (Sezgin & Sankur, 2004).

Eşikleme işlemi, genellikle aşağıdaki nedenlerle yapılmaktadır:

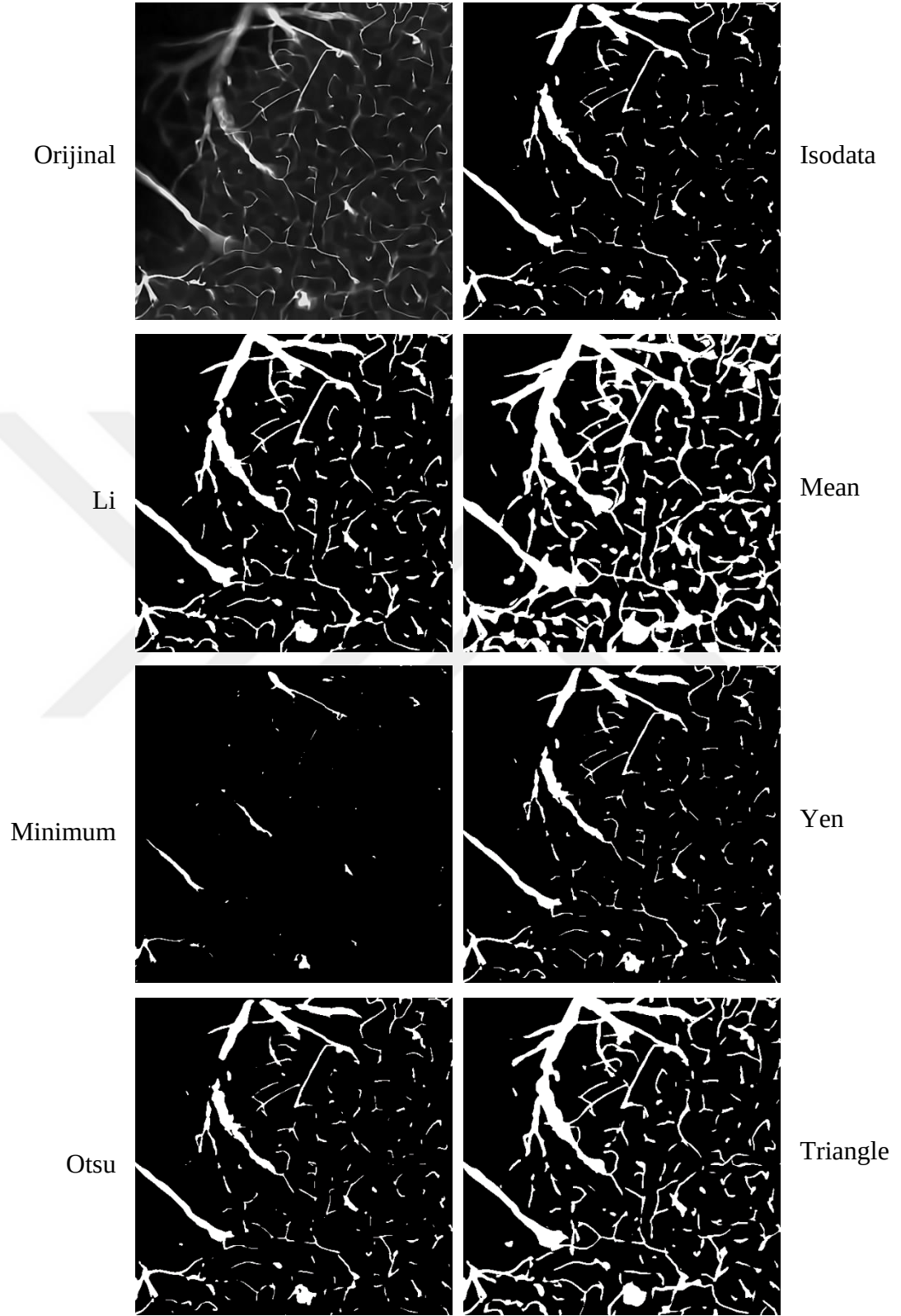
Nesne Segmentasyonu: Eşikleme, görüntüdeki belirli nesnelerin arka plandan ayrılması için kullanılır. Örneğin, bir tıbbi görüntüde tümör gibi belirli bir yapı, belirli bir parlaklık seviyesinin üzerinde olabilir. Eşikleme, bu yapının izole edilmesine yardımcı olur.

Görüntü Sadeleştirme: Görüntüyü iki seviyeye indirgemek, daha basit ve daha hızlı analiz yapılmasını sağlar. Bu, özellikle gerçek zamanlı sistemler veya düşük hesaplama gücüne sahip sistemler için önemlidir.

Gürültü Azaltma: Eşikleme, görüntüdeki düşük yoğunluklu gürültülerin filtrelenmesine yardımcı olabilir. Örneğin, belirli bir parlaklık seviyesinin altındaki pikselleri sıfırlamak, gürültüyü azaltabilir.

Kenar Tespiti: Eşikleme, kenarlık tespitinde de kullanılır. Kenarlar, parlaklıkta ani değişimlerin olduğu bölgelerde bulunur ve bu bölgeleri belirlemek için eşikleme uygulanabilir.

Tez çalışmasında damarların bölütlenmesi işlemi için eşikleme yapılmıştır. Temelde görüntülerimizi analiz ettiğimizde en çok bilgi çıkaran yöntem tahmin edilebilse dahi sonuçları görmek açısından birçok filtreyi deneyip sonuçları **Şekil 2.6**'te gösterilmiştir.



Şekil 2.6: Farklı Filtrelerle Eşiklenmiş Görüntüler

Isodata Eşikleme: Damarları nispeten iyi şekilde belirgin hale getirmektedir ancak bazı küçük detaylar eksik olduğu gözlemlenebilir.

Li ve Mean Eşikleme: Damar yapısını orta derecede öne çıkarmaktadır ancak çok fazla detay sunmadığı görülmektedir.

Minimum ve Yen Eşikleme: Daha fazla detay sağlamaktadır, fakat görüntüde bazı bölgeler fazla sadeleştirilmiş olduğu belirgin.

Otsu ve Triangle Eşikleme: Damar yapısının en iyi şekilde çıkarıldığı yöntemler olarak gözükmektedir. Bu yöntemlerle damarların en belirgin ve net şekilde görüldüğü, daha fazla bilgi içeren görüntüler elde edilir.

Gözleme ile yapılan analizde görüntüden en çok anlamlı damar bilgisi çıkaran Otsu, Triangle olarak tespit edilmiştir. Bunlar arasında en çok bilgi veren eşikleme algoritmasının Triangle olduğuna karar verilmiş ve bu filtre ile çalışmaya devam edilmiştir.

Triangle eşikleme filtresi, bir görüntünün histogramını analiz ederek eşik değerini belirlemek için kullanılan otomatik bir yöntemdir. Bu yöntem, histogramın biçimini dikkate alarak ideal bir eşik değeri seçer ve genellikle ışıklandırma koşullarından bağımsız olarak nesneleri arka plandan ayırmak için uygundur.

- Çalışma Prensipleri:
 - Histogram Oluşturma:
 - İlk olarak, görüntüdeki piksel yoğunluklarının dağılımını temsil eden bir histogram oluşturulur.
 - En Yüksek Yoğunluk Bulma:
 - Histogramda en yüksek frekansa sahip yoğunluk değeri bulunur (tepe noktası).
 - Histogramun Kuyruğu:
 - Histogramın tepe noktasından başlayarak, bir tarafında kalan kuyruk (genellikle düşük yoğunluk veya yüksek yoğunluk tarafı) analiz edilir.
 - Doğru Çizme:
 - Tepe noktası ile histogramın kuyruk kısmının son noktası arasında bir doğru çizilir. Bu doğru, histogramın yatay eksenini ile belirli bir açı yapar.

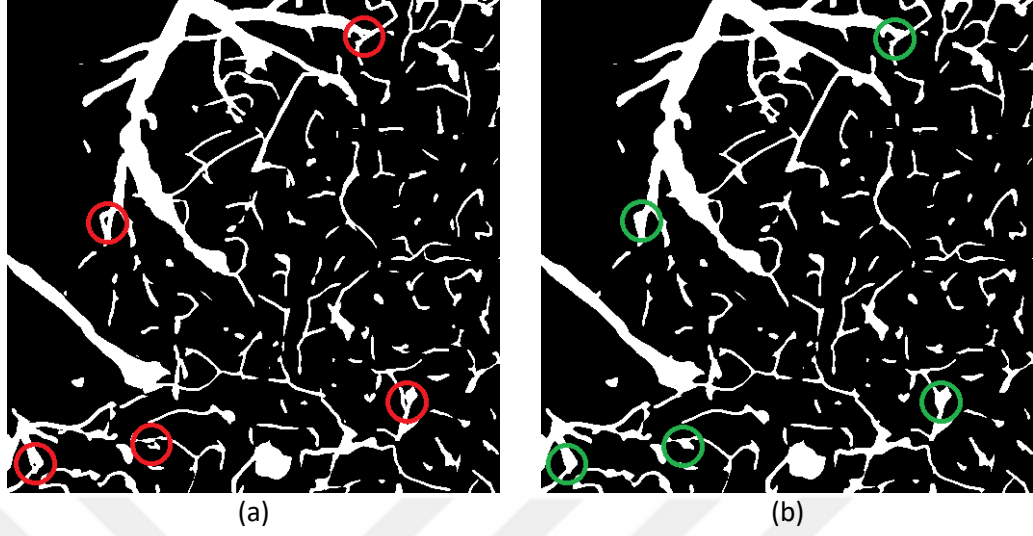
- Eşik Değerinin Belirlenmesi:
 - Çizilen doğruya en uzak olan histogram noktası (maksimum dik mesafeye sahip olan) eşik değeri olarak seçilir. Bu nokta, histogramdaki veri yoğunluğuna göre nesne ve arka planı en iyi şekilde ayırır.

Triangle eşikleme filtresi, biyomedikal görüntüleme, endüstriyel görüntü analizleri ve metin/simge ayıklama gibi alanlarda sıkça tercih edilir.

2.1.5. İkili Kapanma Filtresi

İkili kapama filtresi, görüntü işlemede yaygın olarak kullanılan morfolojik bir operasyondur. Bu filtre, genellikle ikili (binary) görüntülerdeki küçük deliklerin doldurulması ve nesnelerin sınırlarının düzgünleştirilmesi amacıyla kullanılır. Genişletme (Dilation) işlemi ile başlar, ardından aşındırma (Erosion) işlemi gelir. Bu sayede, görüntüdeki nesneler arasındaki küçük boşluklar kapatılır ve nesnelerin parçaları birleştirilir. Özellikle tıbbi görüntüleme ve belge analizinde, bu filtre, küçük gürültülerin ortadan kaldırılması ve nesnelerin kesintisiz hale getirilmesi için oldukça önemlidir. Örneğin, tıbbi görüntülerde hücre veya doku sınırlarının düzgünleştirilmesi veya belgelerde metin parçalarının birleştirilmesi gibi işlemlerde kullanılır (Fasano vd., 2021).

Eşikleme uygulandıktan sonra damar görüntüleri içlerinde küçük boşluk olduğu durumlar ortaya çıkmaktadır. Bu boşluklar iskeletleştirme işleminde algoritmanın kötü sonuçlar vermesine neden olmaktadır. Sorundan kaçınmak için İkili Kapanma (Binary Closing) filtresi uygulanması yoluna gidilmiştir. Burada genişletme (dilation) işlemi uygulanabilir ancak genişletme damar kalınlıklarını da artırmaktadır. Çok önemli olmayan küçük bir damar daha büyük görünmesine sebebiyet verdiği için genişletme küçük nesnelerin silinmesinden sonra uygulanmıştır. İkili kapanma bir görüntünün kısımlarını aynen korurken sadece iç kısmındaki boşlukları doldurmaktadır.



Şekil 2.7: Eşiklenmiş Görüntüde Üzerine İkili Kapatma Filtresi İyileştirmesi Uygulanmış Görüntü Karşılaştırması. (a): Triangle Eşiklenmiş Görüntü, (b): İkili Kapatma Filtresi Uygulanmış Görüntü

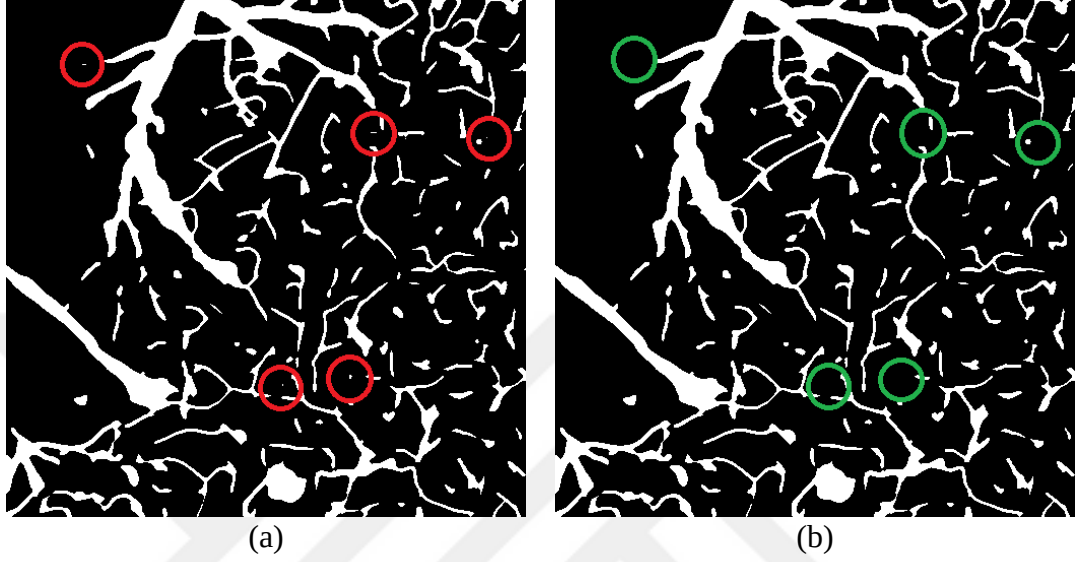
Bu çalışmada, **Şekil 2.7**'de eşikleşmiş görüntü ile ikili kapatma filtresi uygulanmış görüntünün karşılaştırması verilmiştir. İlk görüntüde damarların ortalarında boşluklar varken ikinci görselde ikili kapatma ile yapılmış iyileştirmeler gösterilmiştir. Bu filtre, damar yapılarını daha düzgün hale getirmiş ve küçük boşlukların kapanmasını sağlamıştır. İkili kapatma filtresi, özellikle iskeletleştirme gibi algoritmaların daha doğru sonuçlar vermesini sağlamak amacıyla kullanılmaktadır. **Şekil 2.7.a**'da damarların arasında bazı boşluklar varken, ikinci görselde bu boşlukların doldurulmuş olduğu gözlenmektedir. Bu tür iyileştirmeler, görüntüdeki yapılar üzerinde daha fazla doğruluk elde edilmekte ve algoritmaların daha kararlı sonuçlar üretilmektedir. Bu durum, tıbbi görüntü işleme gibi alanlarda son derece kritik olabilmektedir.

2.1.6. Küçük Nesnelerin Silinmesi

Damar görüntüleri kesit yığıtları şeklindedir. Kimi görüntülerde bir sonraki görüntüde daha ön plana çıkan damar bir öncekinde belirli belirsiz görünmektedir ve bölütleme yapıldığında bölüt gürültüsü oluşturmaktadır. Bu küçük gürültülerin silinmesi iskelet çıkarma işleminin daha sağlıklı olmasını sağlamaktadır.

Küçük nesneleri silmek için bağlantılı bileşen analizi yöntemi kullanılmıştır. Yöntem, görüntüdeki beyaz piksellerin (True olan piksellerin) birbiriyle bağlantısını kontrol ederek bağlantılı bölgeleri (connected components) tespit eder. Bağlantılı

bileşenler genellikle 4-bağlantı (komşu pikseller yatay ve dikey olarak bağlanır) veya 8-bağlantı (komşu pikseller çapraz da bağlanır) kurallarına göre belirlenir. Tespit edilen bağlantılı bileşenlerin boyutları (piksel sayıları) hesaplanır.



Şekil 2.8: Referans Görüntü ve Küçük Bölütleri Silinmiş Görüntü Karşılaştırması.
(a): Referans Görüntü, (b): Küçük Bölütler Silinmiş Görüntü

Şekil 2.8 referans görüntü ve küçük bölütleri silinmiş görüntü arasındaki farklar gösterilmiştir. Şekil 2.8.b’de bazı küçük bölütler filtrelenmiş görüntüde yoktur. Uygulanan filtrede ne kadar küçük bölütlerin silineceği parametre olarak verilebilmektedir. Şekilde 10 piksel boyutundan küçük bölütler silinmiştir. Silinmediği durumlarda bu küçük partiküller de damar olarak değerlendirilmektedir. Bu da hata durumunu artırmaktadır.

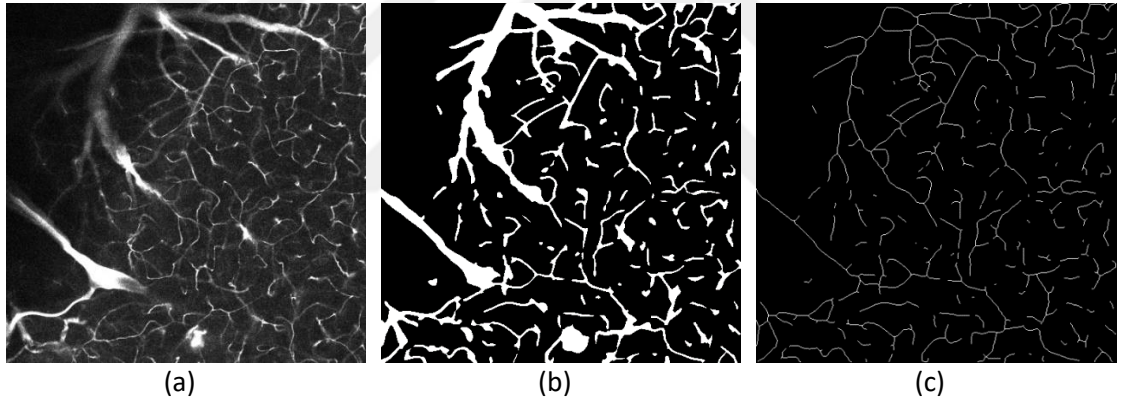
2.1.7. Görüntü İskeletinin Çıkarılması

Görüntüler üzerine yapılacak son işlem damar iskeletinin çıkarılmasıdır. İskeletleştirme işlemi tespit edilen bölütlerin bir piksel olarak inceltilmesidir. Bu işlemdeki temel amaç alan bazlı bölütlenmiş kısımların genel formunun çıkarılmasıdır. İskelet kavramı, H. Blum tarafından Medikal Eksen Dönüşümü (Medical Axis Transformation-MAT) veya Simetri Eksen Dönüşümü (Symmetry Axis Transformation-SAT) sonucunda tanıtıldı. MAT, bir nesnedeki her nokta için en yakın sınır noktalarını belirler. Bir iç nokta, en az iki en yakın sınır noktasına sahipse iskeleti oluşturan noktalardan biridir (Duan vd., 2008).

İskeletin çok açıklayıcı bir tanımı çayır-ateş benzetmesi ile verilmektedir: Bir nesnenin sınırı ateşe verilir ve iskelet, yangın cephelerinin buluştuğu ve birbirini söndürdüğü yerdir.

Üçüncü yaklaşım resmi bir tanım sağlar: iskelet, tüm maksimum yazılı hiper-kürelerin (yani sırasıyla 2B ve 3B diskler ve toplar) merkezlerinin yeridir. (Bir yazılı hiper-küre, başka herhangi bir yazılı hiper-küre tarafından kapsanmıyorsa maksimumdur.) (He vd., 2021).

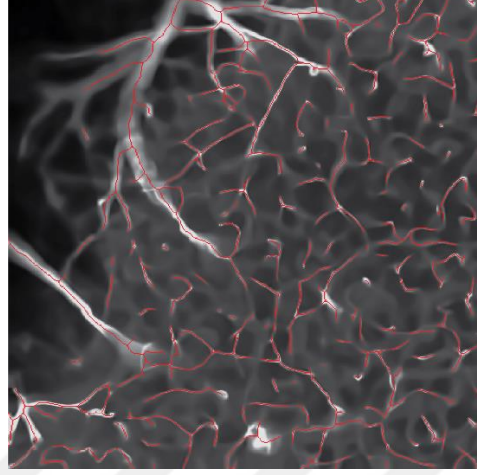
Bir bölütün iskeletinin çıkarılması yöntemleri çeşitli olsa da bu tez kapsamında scikit-image kütüphanesinin hazır fonksiyonu kullanılmıştır(Y. He vd., 2021). Gürültüden temizleyip ikili forma dönüştürdüğümüz görüntünün iskelet formu çıkarılmıştır. **Şekil 2.9**'de orijinal görüntü ve işlenmiş görüntülerin çıktıları verilmiştir.



Şekil 2.9: Orijinal Görüntü Temizlenmesi ve İskelet Şeması Görüntüleri. (a): Orijinal Görüntü, (b): Bölütlenmiş Görüntü, (c): Görüntü İskeleti

İskeletleştirme işlemi, özellikle damar yapıları gibi karmaşık ve dallı yapılarda topolojik özelliklerin korunmasını sağlar. Bu işlem, nesnelerin sadece dış hatlarını değil, iç yapısını da anlamamıza yardımcı olur. Böylece, damar gibi karmaşık yapılarda dallanma noktaları, uç noktalar ve genel form hakkında daha net bilgi elde edilir. Ayrıca, bu yöntem, tıbbi görüntü analizinde damarların yapısal özelliklerini incelemek, bölütleme sonrası nesnelerin temel yapısını çıkarmak için kritik bir araçtır. Scikit-image kütüphanesi gibi hazır araçlar, bu süreci hızlandırarak iskelet çıkarma işlemini daha erişilebilir kılmaktadır. İskeletleştirmenin sağladığı bu sadeleştirilmiş yapılar, özellikle damarların topolojik analizlerinde büyük bir avantaj

sağlar. Bunun yanında, iskelet formu, nesnelerin genel yapısal karakteristiklerini korurken veri boyutunu küçülterek işlem süresini de optimize etmektedir.

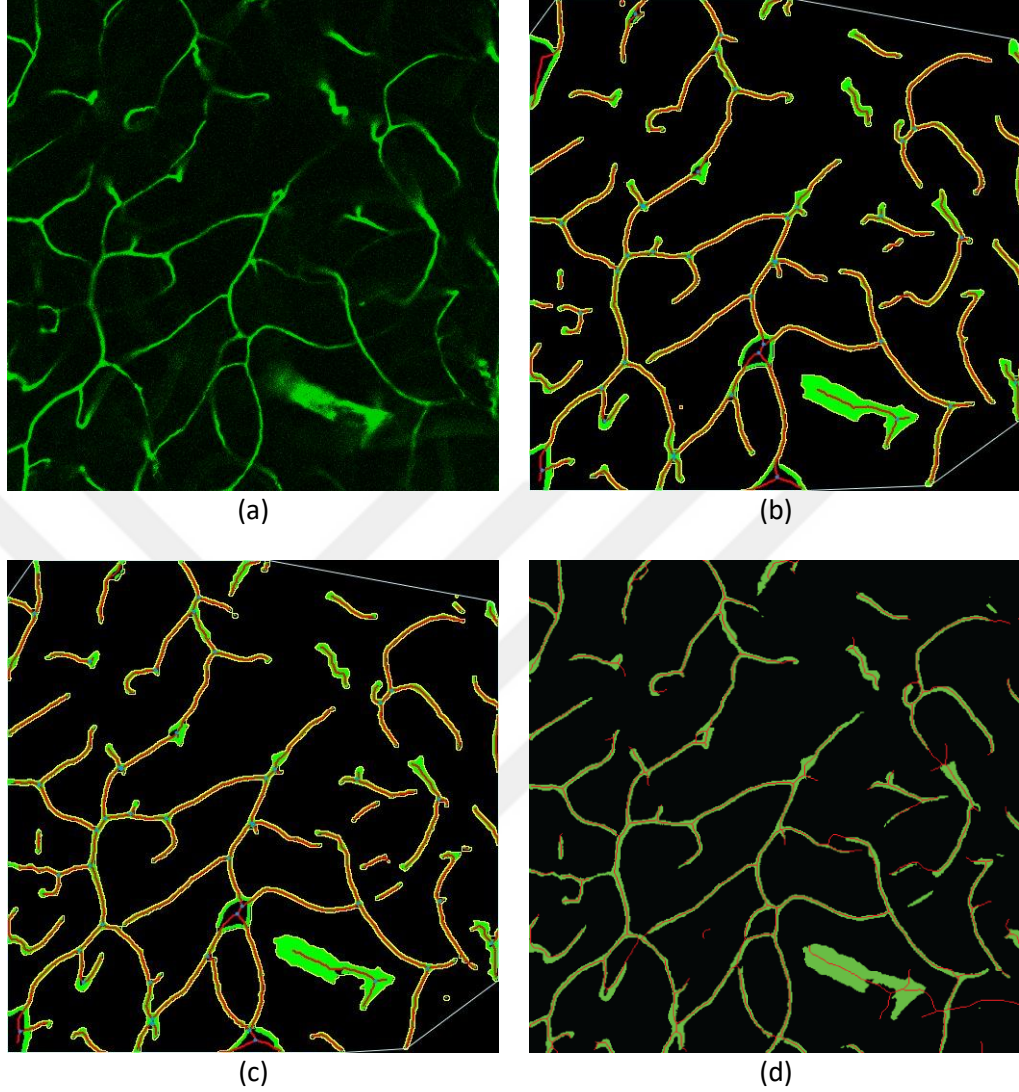


Şekil 2.10: Damar İskeleti ile Temizlenmiş Görüntünün Üstüste Hali

İskelet ve orijinal görüntünün temizlenmiş halini üst üste konulduğunda işlemlerin başarımını daha iyi görülmektedir. **Şekil 2.10**'da iki görüntü üst üste hali gözlemlenebilmektedir. Henüz iskelet görüntüsü sayısallaşmadığından herhangi bir sayısal başarımlar hesaplanmamıştır. Ancak gözle yapılan tetkikte görüntünün başarılı bir iskelet çıkarma işlemi olduğu söylenebilir. Bu seviye bir başarımlar görüntülerdeki dallanma ve uç noktaların işaretlenmesinde kullanılacak temel iskeleti vermektedir.

Ulaşılan aşamada görüntüden çıkarılmış iskelet sayısallaştırılmamıştır. Damar kalınlıkları, boyutları ve dallanma noktaları gibi bilgiler elde edilmediğinden karşılaştırma sadece gözlem yoluyla olabilmektedir.

Daha önce denetimli öğrenme yoluyla yapılan bir çalışmanın sonuçları ile uyguladığımız metodun sonuçlarını gözlemlemek için bir deneme yapılmıştır. **Şekil 2.11**'de ResNet ve UnasNet ile denetimli öğrenme yöntemiyle çıkarılmış görüntü iskeletleriyle yapılan çalışmadaki denetimsiz sadece görüntü işleme teknikleri kullanılarak çıkarılmış iskelet görüntüsü görülmektedir.



Şekil 2.11: İskelet Çıkarma Çalışmasının Diğer Yöntemler ile Karşılaştırılması. (a): Orijinal , (b): ResNet , (c): UNAS-Net, (d): Uygulanan Metot

Orijinal görüntü ile karşılaştırıldığında uygulanan metodun daha doğru çıkardığı iskelet kısımları varken daha başarısız yaptığı yerler de bulunmaktadır. Ancak genel gözlemde çalışmaların başarımlarının yakın olduğu söylenebilir.

2.2. Dallanma ve Uç Noktalarının Geleneksel Görüntü İşleme Yöntemleri ile Tespiti

Vasküler analiz çalışmaları, tıbbi görüntüleme alanında büyük bir öneme sahiptir ve bu çalışmaların doğru bir şekilde gerçekleştirilmesi, hastalıkların teşhisi ve tedavi sürecinde kritik rol oynar. Damarların yapısal özelliklerinin incelenmesi, özellikle tümör gelişimi ve doku hasarı gibi durumların değerlendirilmesi için önemli

bilgiler sunar. Bu bağlamda, damarsal alan, vasküler uzunluk, dallanma sayısı ve uç nokta sayısı gibi temel özellikler üzerinde durulmuştur. Bu özellikler, damar ağının yapısını ve işlevini anlamak için gereklidir. Bu kısımda, bu özelliklerin tespit edilmesine yönelik çeşitli görüntü işleme teknikleri açıklanmış ve “vur ya da ıskala” algoritması gibi yöntemlerin nasıl uygulandığı ayrıntılı olarak ele alınmıştır. Ayrıca, dallanma ve uç noktaların tespiti için kullanılan algoritmaların etkinliği ve doğruluğu üzerine yapılan çalışmalar da incelenmiştir. Son olarak, tespit edilen bu özelliklerin derin öğrenme nesne tespiti metotlarına nasıl entegre edileceği ve veri setinin bu doğrultuda nasıl oluşturulacağı hakkında bilgiler verilmiştir.

2.2.1. Damar Ağı Özellikleri

Vasküler analiz çalışmaları, tıbbi görüntülemede önemli bir yere sahiptir. Bu çalışmalar, damarların uzunluk, kalınlık bilgileri, dallanma ve uç noktaları gibi çeşitli özellikleri üzerinde durarak doku ve organların sağlığı hakkında bilgi sağlar. Vasküler yapıların doğru bir şekilde tespit edilmesi, hastalıkların teşhisinde ve tedavi sürecinde kritik bir rol oynamaktadır. Bu bağlamda, damar analizlerinde dört temel özellik sıklıkla öne çıkar: damarsal alan, dallanma sayısı, uç nokta sayısı ve toplam uzunluk. Bu dört özellik, doku içindeki kan akışını, damarların dağılımını ve yapısını anlamaya yönelik önemli bilgiler sunar. Bu bilgiler, özellikle tümör gelişimi veya doku hasarı gibi durumların değerlendirilmesinde kullanılır. Tez kapsamında dallanma ve uç nokta tespitine odaklanılsa da diğer özellikler üzerine de çalışmalar yapılmıştır. Aşağıda bu özellikler üzerine yapılmış çalışma gösterilmiştir.

2.2.1.1. Damarsal Alan Fraksiyonu

Damarsal alan fraksiyonu (Vascular Area Fraction), belirli bir doku veya organ içindeki kan damarlarının yüzdesel bir ölçüsünü ifade eder. Bu terim, özellikle organların perfüzyonu, yani kan akışının derecesi hakkında bilgi sağlar. Örneğin, bir tümörün veya organın damarsal alan fraksiyonu, tümör büyümesi, anjiyogenez (damar oluşumu) veya doku hasarını değerlendirmek için kullanılabilir. Görüntüleme teknikleri ile elde edilen verilerde, damarsal alan fraksiyonu genellikle segmentasyon işlemi sonrası beyaz piksellerin toplam piksellere oranı olarak hesaplanır. Bu

hesaplama, belirli bir alan içindeki vasküler yapıların oranını verir ve dokunun ne kadarının kan damarı ile kaplandığını gösterir.

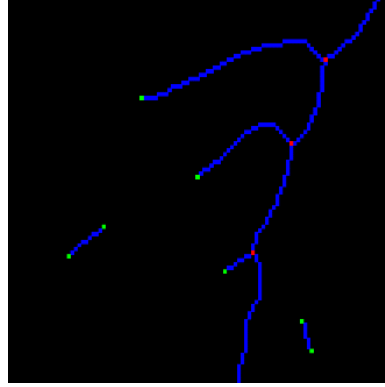
Damar alan fraksiyonu hesaplamasında kullanılan **Denklem (2.2)** matematiksel hesabı göstermektedir:

$$(ToplamBeyazPiksel * 100) / ToplamPiksel \quad (2.2)$$

Bu formül, bölütlenmiş (segment edilmiş) damarların tüm görüntü içindeki oranını temsil eder. Görüntülerde damarlar beyaz piksellerle, damar olmayan alanlar ise siyah piksellerle ifade edilir. Bu yöntem, vasküler alan fraksiyonunun hızlı ve doğru bir şekilde hesaplanmasını sağlar. Çeşitli çalışmalar, bu yöntemin klinik karar destek sistemlerinde ve araştırma projelerinde kullanılabileceğini göstermiştir (Smith et al., 2020; Johnson & Lee, 2021).

2.2.1.2. Vasküler Uzunluk

Vasküler uzunluk, damar yapılarının toplam uzunluğunu ifade eder. Görüntü analizinde, damar uzunluklarının hesaplanması, damar dallanma noktalarının ve uç noktalarının doğru bir şekilde tespit edilmesine dayanır. Dallar uç noktalarda sonlanırken, dallanma noktalarında birleşirler. Çalışmamız kapsamında, damar yollarını bulmak için öncelikle dallanma noktaları 3x3 piksel filtrelerle silinmiştir. Bu işlem, damarları birbirlerinden ayırarak sadece iki uç noktaya sahip hale getirilmektedir. Damar ağında her bir dal ayrı bir damarmış gibi uzunluğu hesaplanmıştır.

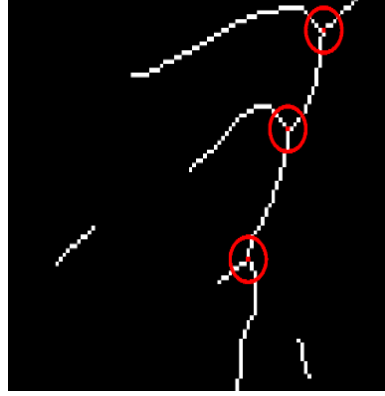


Tablo 2.1: Damar Yollarının Gösterilmesi

Tablo 2.1'de, damar yollarının gösterildiği bir örnek görüntü sunulmuştur. Damar uzunluğunun hesaplanması, öklid mesafesi kullanılarak gerçekleştirilmiştir. Bir damar iskeletinde başlangıç noktasından bitiş noktasına kadar tüm piksellerin konumları kaydedilir ve her bir piksel arası mesafe hesaplanarak kümülatif uzaklığa eklenir. Bu sayede, damarların tüm uzunluğu doğru bir şekilde hesaplanır. Vasküler uzunluğun doğru bir şekilde belirlenmesi, birçok klinik uygulamada önemli bir yere sahiptir ve çeşitli çalışmalar bu yöntemin doğruluğunu göstermektedir (K. Xu vd., 2024; Yao vd., 2020).

2.2.1.3. Vasküler Dallanma Sayısı

Vasküler dallanma sayısı, birbirine bağlı damarların bağlantı noktalarının sayısını ifade eder. Bir damar, diğer damarlarla birleşebilir veya bağlantısız kalabilir. Bu bağlamda, dallanma noktaları, damar yapısının karmaşıklığını ve dallanma modellerini anlamak için kritik bir bilgi sunar. Dallanma sayısının tespiti için görüntü işleme teknikleri kullanılarak iskelet çıkarımı yapılır. Bu işlem sonrası, dallanma noktalarını belirlemek amacıyla 3x3 piksel filtreler uygulanır ve vur ya da ıskala yöntemi kullanılarak dallanma noktaları tespit edilir.

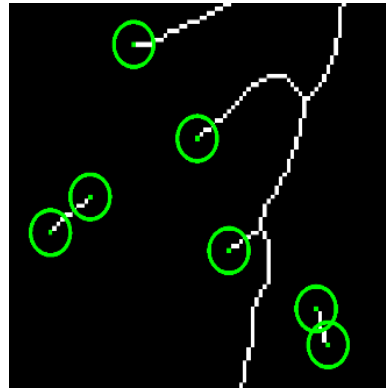


Şekil 2.12: Örnek Dallanma Noktaları

Şekil 2.12'de görüldüğü gibi, dallanma noktaları damarların birleşim yerlerinin tam ortasında yer alır. Bu tür tespitler, özellikle damarların birbirleriyle olan bağlantılarını incelemek ve bu bağlantıların sayısını belirlemek için oldukça önemlidir. Dallanma noktalarının toplam sayısı, damarsal dallanma sayısını verir. Dallanma sayısının tespiti, genellikle tıbbi görüntüleme verilerinde kullanılan yöntemlerle gerçekleştirilir ve bu yöntemlerin doğruluğu çeşitli çalışmalarda gösterilmiştir (Seaman vd., 2011; Zudaire vd., 2011)

2.2.1.4. Vasküler Uç Nokta Sayısı

Vasküler uç nokta sayısı, bir damarın başka bir damara bağlanmadan sona erdiği noktaları ifade eder. Uç noktaların tespiti, dallanma noktalarının tespiti ile benzer yöntemlerle gerçekleştirilir. Ancak bu noktada kullanılan filtreler, uç nokta tespitine uygun olacak şekilde özelleştirilmiştir. Uç nokta tespiti, özellikle damar yapısının sınırlarını ve sonlanma noktalarını belirlemek için kullanılır.



Şekil 2.13: Örnek Uç Noktalar

Şekil 2.13'da uç noktaların gösterildiği örnek bir görüntü sunulmuştur. Uç noktalar, damarların sonlandığı ve başka bir damar ile birleşmediği alanlardır. Bu tür noktaların tespiti, damarların uzunluğunu ve yapısını analiz etmek için gereklidir. Uç nokta sayısının belirlenmesi, birçok çalışmada damar yapısının özelliklerini anlamak ve damarların uzaysal dağılımını incelemek için kullanılmıştır(Freitas-Andrade vd., 2022; Kaya vd., 2024a, 2024b).

2.2.2. Dallanma ve Uç Noktaların Bulunması

Dallanma ve uç noktalarının bulunması, damar görüntüleme ve analizinde kritik bir öneme sahiptir. Bu noktalar, damar ağının yapısal özelliklerini anlamada ve çeşitli medikal teşhis süreçlerinde kullanılmaktadır. Özellikle damar dallanma noktaları, kan akışının yönünü, damar bifurkasyonlarını (çatallanmaları) ve damar ağının genel topolojisini belirlemeye yardımcı olurken, uç noktalar, damar ağının sınırlarını ve sonlanma bölgelerini tanımlamak için kullanılır. Bu noktaların doğru bir şekilde tespit edilmesi, birçok tıbbi uygulama için temel bir adımdır (X. Xu vd., 2012).

Örneğin, retina damarlarının analizinde, dallanma ve uç noktalarının tespiti, göz hastalıklarının erken teşhisinde kritik bir rol oynar. Benzer şekilde, kardiyovasküler hastalıkların değerlendirilmesi, tümör vaskülaritesinin incelenmesi ve organların beslenme durumlarının belirlenmesi gibi alanlarda da bu noktaların doğru bir şekilde bulunması, hastalığın tanı ve tedavisinde büyük bir önem taşır(Guo & Peng, 2020).

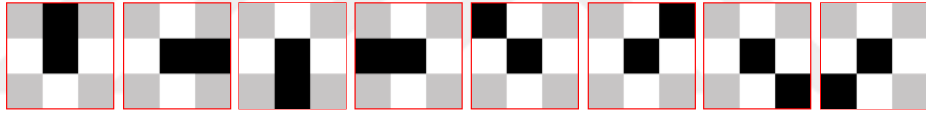
Bu noktaların tespiti, aynı zamanda otomatik görüntü işleme sistemlerinin doğruluğunu artırır. Bu nedenle, dallanma ve uç noktalarının güvenilir ve hassas bir şekilde belirlenmesi hem klinik uygulamalarda hem de araştırma çalışmalarında kritik bir adımdır.

Bu tez kapsamında eğitim ve test veri setlerini oluşturmak için vur ya da ıskala (Hit-or-Miss) algoritması kullanılarak dallanma ve uç noktalar bulunmuştur. Bu nesneler hatasız olarak tespit edildikten sonra derin öğrenme nesne tespiti metotları denenmiştir.

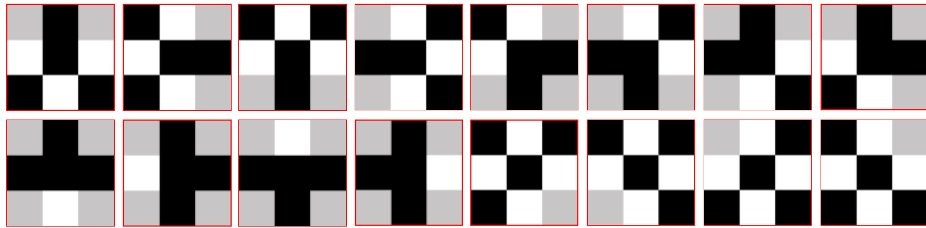
“Vur ya da ıskala” algoritması, görüntü işleme alanında şekil tanıma için kullanılan morfolojik bir operasyondur. Bu algoritma, ön plan (foreground) ve

arkaplan (background) piksellerinin belirli bir düzenlemesini tanımlayan desenleri tespit etmek için kullanılır. Temel olarak, görüntüde bir desenin eşleştiği bölgeleri belirlemek amacıyla erozyon ve tamamlayıcı işlemler gerçekleştirilir. “Vur ya da ıskala” dönüşümü, özellikle ikili görüntülerde şekil algılama, nesne tanıma ve görüntü analizi gibi görevler için oldukça yararlıdır. Bu yöntemin etkinliği, karmaşık desenleri bir görüntüde doğru bir şekilde bulabilmesinden gelir (Murray & Marshall, 2013).

Bu yöntemin temel amacı, dallanma ve uç noktalarının doğru ve kesin bir şekilde tespit edilmesidir. “Vur ya da ıskala” algoritması, bu noktaları tespit etmek için 3x3 boyutunda uzamsal filtreleri görüntü üzerinde kaydırır. Filtrelerin her biri, istenilen desenin bulunduğu noktaları tespit etmek için özel olarak tasarlanmıştır. Dallanma ve uç noktalarının tespiti için farklı filtreler kullanılır; uç noktalar için 8, dallanma noktaları için ise 16 filtre geliştirilmiştir. **Şekil 2.14** ve **Şekil 2.15**'te bu filtreler gösterilmiştir.

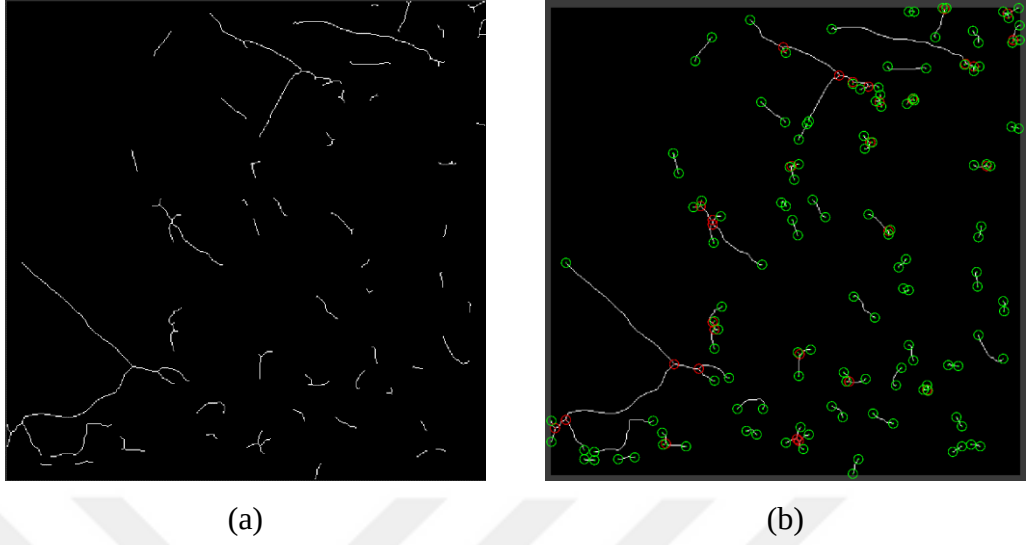


Şekil 2.14: Uç Nokta Bulma Filtreleri



Şekil 2.15: Dallanma Noktası Bulma Filtreleri

Bu işlem sonucunda, görüntünün iskelet yapısı üzerinde dallanma ve uç noktalarının tam yerleri tespit edilir. **Şekil 2.16**'da ise bu noktaların görselleştirilmesi sağlanmıştır. Yeşil daireler uç noktaları, kırmızı daireler ise dallanma noktalarını temsil eder. Geliştirilen Python programı, filtreleri uzamsal olarak görüntü üzerinde kaydırarak bu noktaları tespit eder ve görselleştirir.



Şekil 2.16: Örnek Görüntüdeki Dallanma Noktaları ve Uç noktalar. (a): İskelet Görüntüsü, (b): İşaretlenmiş Görüntü. Yeşil Daireler Uç Noktaları, Kırmızı Daireler ise Dallanma Noktalarını Göstermektedir

Görsellerden de anlaşılabileceği gibi geleneksel görüntü işleme yöntemi uygulayarak tespit ettiğimiz dallanma ve uç noktaların tamamı doğru olarak belirlenmiştir. Bu işlemi veri setinde olan tüm görüntülere uygulanarak her bir görüntü için dallanma ve uç noktaları tespiti gerçekleştirilir. Tüm işlemlerin sonucunda sadece dallanma ve uç noktaların derin öğrenme yöntemlerinde kullanılabilecek formatta işaretleme yapmak ve uygun formatta dönüştürmek kalmaktadır. Bir sonraki bölümde veri setinin oluşturulması detaylı bir şekilde anlatılmaktadır.

2.3. Tespit Edilmiş Nesnelerin Veri Seti Haline Getirilmesi

Yapay zeka ve özellikle derin öğrenme yöntemleri, görüntü işleme ve nesne tanıma alanlarında önemli gelişmeler olmaktadır. Nesnelerin doğru bir şekilde tespit edilmesi, bu yöntemlerin başarısında kritik bir rol oynamaktadır. Ancak, derin öğrenme algoritmalarının etkin bir şekilde çalışabilmesi için geniş ve doğru şekilde etiketlenmiş veri kümelerine ihtiyaç duyulmaktadır. Bu noktada, nesne tanıma işlemleri için geliştirilen veri işaretleme formatları, tespit edilen nesnelerin düzenlenmesi ve öğrenme süreçlerine entegre edilmesi açısından büyük önem taşır. Farklı veri işaretleme formatları, çeşitli uygulamalara uyum sağlamak üzere

tasarlanmıştır. Bu bölümde, tespit edilen nesnelerin uygun formatlara dönüştürülmesi ve veri seti haline getirilmesi süreci açıklanacaktır.

Nesne tanıma için birçok veri işaretleme formatı vardır. Bunlar arasında en yaygın olarak kullanılanlar PASCAL VOC (Visual Object Classes), YOLO (You Only Look Once), LabelImg XML ve COCO formatlarıdır (Dave vd., 2020; Everingham vd., t.y.; Lin vd., 2014; Redmon vd., 2016). Veri kümesi anotasyonları için Microsoft'un geliştirmiş olduğu “Common Object in Context” (Bağlam İçinde Ortak Nesne) COCO formatını tercih ediyoruz (Lin vd., 2014). COCO, bilgisayarla görmede nesne yerleştirme ve anlamsal segmentasyon gibi görevler için Microsoft tarafından geliştirilen bir veri kümesi formatıdır ve derin öğrenme modellerinin eğitimi ve değerlendirilmesi için yaygın olarak kullanılmaktadır.

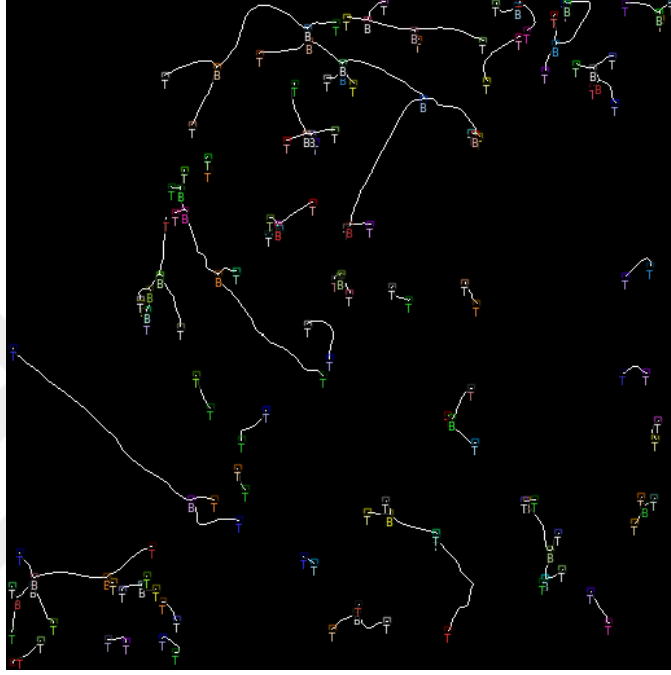
COCO veri seti formatı JSON formatı ile geliştirilmiştir. Formatta üç önemli kategori bulunmaktadır: “resimler (images)”, “kategoriler (categories)”, “notasyonlar (annotations)”.

- Resimler kısmında dosyada bulunan resimlerin kimlik değeri(id), boyut ve dosya ismi bulunmaktadır.
- Kategoriler kısmı veri setinde bulunan nesne tiplerini ifade eder ve tip kimlik bilgisi ile isim bilgisini tutar.
- Notasyonlar kısmı görüntü üzerinde bulunan nesnelerin bilgilerini verir. Burada nesne hakkında kimlik bilgisi (id), görüntü aitlik bilgisi (image_id), kategori bilgisi (category_id), kesişimli bir nesne olup olmadığı (iscrowd), nesne çerçeve bilgisi (bbox), çerçeve alan bilgisi tutulmaktadır (area).

Tezde kullanılmış olan görüntü seti 3DISCO prosedüründen geçirilmiş temizlenmiş fare beyin örneklerinden elde edilen fare korteks görüntüleri kullanılmıştır (Özkan vd., 2023). Bu veri kümesi, 512×512 boyutlarında 96 görüntüye sahiptir.

Ön işlemde geçirilerek iskeleti çıkarılan görüntülerde 3x3 piksel boyutundaki filtrelerle yapılan geleneksel görüntü işleme, tüm nesneleri doğru bir şekilde bulunduğu bir önceki bölümde açıklanmıştır. Derin öğrenme nesne tanıma yöntemi için, bu nesnelerin çerçeve içinde alınarak işaretlenmesi ve bu işaretleme

COCO formatına dönüştürülmesi gerekmektedir. Bu nedenle, her görüntüde mekansal olarak tespit edilen dallanma ve uç noktalarını 5x5 piksellik bir çerçeve içine alınmıştır. "B" ve "T", sırasıyla dallanma noktalarını ve uç noktaları temsil etmektedir.



Şekil 2.17: 5x5 Piksel Çerçeveler ile İşaretlenmiş Dallanma ve Uç Noktaları

COCO formatlı veri kümesinde işaretlerin doğruluğunu metinsel bir dosyadan kontrol etmek zor bir işlem olduğundan işaretlerin konum ve tür doğruluğunu hızlı bir şekilde gözlemleyebilmek için çerçevelenmiş ham dallanma ve uç noktalarının görüntülerini kaydettik. **Şekil 2.17**'de örnek bir işaretlenmiş görüntü gösterilmiştir. Veri setindeki her görüntü için ayrı ayrı bu kayıt alınmış ve gözlem yoluyla doğruluğu kontrol edilmiştir.

COCO (Common Objects in Context), bilgisayarlı görü konusundaki araştırmalar için bir veri seti formatıdır. COCO, Microsoft tarafından oluşturulan ve genellikle nesne tanıma, nesne sınıflandırma, nesne yerelleştirme ve semantik segmentasyon gibi görevler için kullanılan bir veri setidir. COCO veri seti hem eğitim hem de test amaçlı kullanılabilecek geniş bir veri yelpazesi sunar ve bilgisayarlı görü üzerine yapay zeka ve derin öğrenme modellerinin eğitilmesi ve değerlendirilmesi için yaygın olarak kullanılır. Bu veri seti, algoritmaların nesneleri

tanıma, sınıflandırma, konumlandırma ve segmentasyon gibi görevlerdeki performansını ölçmek için genellikle bir referans olarak kullanılır.

Bu çalışmada yerleri görüntü işleme yöntemiyle bulunmuş nesneler kod yardımıyla nesne merkezde olacak şekilde çerçeve boyutu ve tip değerleri verilerek COCO formatına çevrilmiştir.

```
{
  "images": [
    {
      "id": 1,
      "width": 520,
      "height": 520,
      "file_name": "HFHS13_z11.png"
    }
  ],
  "categories": [
    {
      "id": 0,
      "name": "T"
    },
    {
      "id": 1,
      "name": "B"
    }
  ],
  "annotations": [
    {
      "id": 1,
      "image_id": 1,
      "category_id": 1,
      "iscrowd": 0,
      "bbox": [123.5,493.5,5,5],
      "area": 25
    },
    {
      "id": 2,
      "image_id": 1,
      "category_id": 2,
      "iscrowd": 0,
      "bbox": [279.5,489.5,5,5],
      "area": 25
    }
  ]
}
```

Şekil 2.18: COCO Formatındaki Özet Dosya

Oluřturulan COCO formatındaki dosya ierięi zet halinde **řekil 2.18**'de gsterilmiřtir. rnekke tek bir grnt, iki kategori, 2 notasyon bulunmaktadır. Nesne sayıları ok olduęundan dosya boyutları olduka byktr.

Grntlerin hepsi ayrı ayrı iřlemlerden geirilse dahi tek bir tespit edilen nesneler grnt numarasıyla tek bir COCO dosyasına kayıt edilmektedir. İřlem grselleřtirme aısından tm grntler iin eęitim ve test fark etmeksizin yapılmaktadır. Dallanma noktaları mavi-mor tonlarında renkler alırken u noktalar yeřil tonlarıyla olarak belirtilmiř ve tipleri “B”, “T” olarak alta yazdırılmıřtır.



ÜÇÜNCÜ BÖLÜM

3. DERİN ÖĞRENME METODLARI İLE NESNE TESPİTİ

Geleneksel görüntü işleme yöntemleri, belirli kurallar ve eşik değerlerine dayalı sabit algoritmalarla çalışır. Özellikle biyomedikal görüntüleme alanında, bu yöntemler karmaşık ve değişken yapılarla başa çıkmada sınırlamalar sergiler. Örneğin, fare beyni gibi dokuların analizinde geleneksel yöntemlerin yetersizlikleri belirgindir. Bu yöntemler, manuel olarak tanımlanmış özelliklere (örneğin, kenarlar, dokular, renkler) dayanır ve özel olarak tasarlanmış algoritmalar gerektirir. Bu durum, esnekliği azaltır ve genel uygulanabilirliklerini sınırlı hale getirir. Ayrıca, geleneksel yöntemler genellikle insan müdahalesine ihtiyaç duyar, bu da zaman alıcı ve hata yapmaya yatkın bir süreçtir. Son olarak, karmaşıklık ve ölçeklenebilirlik açısından, geleneksel yöntemler büyük veri setlerini verimli bir şekilde işlemek konusunda zorlanabilirler.

Bu sınırlamaları aşmak için derin öğrenme yöntemleri etkili bir alternatif sunar. Özellikle Convolutional Neural Networks (CNN'ler) gibi derin öğrenme modelleri, karmaşık görüntü özelliklerini otomatik olarak öğrenebilir ve tanımlayabilir. Bu modeller, ham veriden anlamlı özellikleri otomatik olarak çıkarma konusunda son derece başarılıdır, bu da insan müdahalesi ve manuel özellik mühendisliğine olan ihtiyacı önemli ölçüde azaltır. Derin öğrenme modelleri, geniş ve çeşitli veri setleri üzerinde eğitildiklerinde, farklı veri setlerine ve koşullara karşı daha genel ve sağlam performans sergileyebilirler. Ayrıca, derin öğrenme genellikle görüntü sınıflandırma, segmentasyon ve tanıma görevlerinde geleneksel yöntemlere kıyasla daha yüksek doğruluk sağlar. GPU'larla ve büyük veri setleriyle eğitilen derin öğrenme modelleri, büyük miktarda veriyi hızlı ve verimli bir şekilde işleyebilir. Bu bağlamda, derin öğrenme, özellikle biyomedikal görüntüleme gibi alanlarda, doğruluk ve verimlilik açısından önemli bir ilerleme olarak kabul edilir. Bu tür çözümler, araştırmaların ve uygulamaların kalitesini önemli ölçüde artırabilir.

Geleneksel görüntü işleme yöntemleri ve derin öğrenme teknikleri, bu iki yaklaşım görüntü analizi alanında yaygın olarak kullanılan iki farklı yöntemdir. Her iki yöntemin de kendine özgü avantajları ve dezavantajları bulunmaktadır. Geleneksel yöntemler genellikle basit ve belirli durumlarda etkili olurken, esneklik ve genel uygulanabilirlik açısından sınırlı kalmaktadır. Buna karşılık, derin öğrenme teknikleri karmaşık ve büyük veri setlerinde üstün performans gösterir, ancak yüksek hesaplama gücü ve büyük miktarda veri gerektirir. Bu bağlamda, her iki yöntemin güçlü ve zayıf yönleri, uygulama alanına ve ihtiyaçlara bağlı olarak değerlendirilmeli ve doğru seçim yapılmalıdır. Bu yöntemlerin avantaj ve dezavantajlarını karşılaştırabiliriz.

Geleneksel görüntü işleme yöntemleri:

Geleneksel görüntü işleme yöntemleri genellikle sabit algoritmalar ve elle tanımlanmış özelliklere dayanır. Bu yöntemler belirli kurallar ve eşiklerle çalışır, bazı görüntü türlerinde etkili olabilir, ancak karmaşık ve değişken yapılarla başa çıkmada genellikle yetersiz kalır. Örneğin, biyomedikal görüntülemelerde, bu yöntemlerin sıçan beyni gibi dokuları analiz etmede önemli sınırlamaları vardır. Geleneksel yöntemlerin sınırlamaları şu şekilde kategorize edilebilir (Calvo vd., 2011; Murray & Marshall, 2013; Scholl vd., 2011) :

Öznitelik Seçimi: Geleneksel yöntemler, kenarlar, dokular, renkler gibi özelliklerin elle tanımlanmasını gerektirir. Bu durum, özelleştirilmiş, duruma özel algoritmalar gerektirir ve esnekliği azaltır.

Genel Uygulanabilirlik: Sabit kurallar, görüntüdeki değişikliklere (örneğin, aydınlatma, kontrast farkları) duyarlı olabilir ve farklı veri setlerinde tutarlı sonuçlar elde etmeyi zorlaştırır.

İnsan Müdahalesi: Bu yöntemler genellikle manuel müdahale gerektirir, bu da zaman alıcı ve hataya açık olabilir. Ayrıca, bu yöntemler uzmanlık gerektirir, bu da uygulama alanını sınırlar.

Karmaşıklık ve Ölçeklenebilirlik: Geleneksel yöntemler, karmaşık ve büyük veri setlerini işlemekte zorlanır. Büyük miktarda veriyi verimli bir şekilde işlemek için yetersiz kalabilirler.

Derin Öğrenme Yöntemleri:

Derin öğrenme, bu sınırlamaların üstesinden gelmek için güçlü bir alternatif sunar. Özellikle Evrişimli Sinir Ağları (CNN) gibi derin öğrenme modelleri, karmaşık görüntü özelliklerini otomatik olarak öğrenebilir ve tanıyabilir. Derin öğrenmenin başlıca avantajları şunlardır (Chamarty Anusha, 2018; Eunhwa & Kyungho, 2006):

Özellik Öğrenimi: Derin öğrenme modelleri, ham veriden anlamlı özellikler çıkarmada mükemmeldir, bu da insan müdahalesini ve manuel özellik mühendisliğini önemli ölçüde azaltır.

Genel Uygulanabilirlik ve Uyarlanabilirlik: Derin öğrenme modelleri, çeşitli veri setlerine ve koşullara karşı dikkat çekici bir uyarlanabilirlik gösterir. Geniş ve çeşitli veri setlerinde eğitildiklerinde, bu modeller daha genel ve sağlam bir performans sergiler.

Yüksek Doğruluk: Derin öğrenme, genellikle görüntü sınıflandırma, segmentasyon ve tanıma görevlerinde geleneksel yöntemlere kıyasla daha yüksek doğruluk sağlar.

Verimlilik ve Ölçeklenebilirlik: Derin öğrenme modelleri, GPU'larla eğitildiğinde ve büyük veri setleriyle çalıştığında, büyük miktarda veriyi hızlı ve verimli bir şekilde işleyebilir.

Tez kapsamında yapılan dallanma ve uç noktalarının tespiti çalışmasının geleneksel görüntü işleme algoritmaları ile tespitinin algoritmik karmaşık hesabı **Denklem (3.1)**'de verilmiştir. Denklemde 3x3 boyutunda sekiz uç nokta filtresi on altı dallanma noktası filtresi nxn boyutunda bir resme uygulanmaktadır. Bunun sonucunda işlem karmaşıklık değeri n^2 olarak çıkmaktadır.

$$\begin{aligned} &\Rightarrow \left(\begin{matrix} 3 \times 3 \\ \text{Filter} \\ \text{Size} \end{matrix} \times \left(\begin{matrix} 8 \\ \text{End} \\ \text{Point} \\ \text{Filters} \end{matrix} + \begin{matrix} 16 \\ \text{Branch} \\ \text{Point} \\ \text{Filters} \end{matrix} \right) \right) \times \begin{matrix} n \times n \\ \text{Image} \\ \text{Size} \end{matrix} \\ &\Rightarrow 216 \times n^2 \\ &\Rightarrow O(n^2) \end{aligned} \tag{3.1}$$

Yapılan karmaşıklık hesabından da görülebileceği gibi, zaman karmaşıklığı (n^2) görüntü boyutu büyüdükçe katlanarak artmaktadır. Bu da algoritmanın ölçeklenebilirliğini etkilemektedir.

Tez çalışmasında dallanma ve uç noktaların tespitinde derin öğrenme algoritmaları denenmiş, test edilmiş ve başarımı hem derin öğrenme metotları ile hem de geleneksel metot ile karşılaştırılmıştır.

3.1. Derin Öğrenme Metotları İle Dallanma Ve Uç Noktaların Bulunması

Bu kısımda derin öğrenme metotları ile nesne tanıma işlemini uygulama biçimi anlatılmaktadır.

Nesne tespiti, bilgisayarla görme alanında önemli bir araştırma konusudur. Derin öğrenme tabanlı yöntemler, son yıllarda bu alanda büyük ilerlemeler sağlamıştır. Derin sinir ağlarının (Deep Neural Networks) kullanımının yaygınlaşması, nesne tespitinde daha yüksek doğruluk oranlarına ulaşılmasını mümkün kılmıştır. Bu çalışmada, derin öğrenme tabanlı nesne tespiti yöntemleri ve bu alanda kullanılan popüler kütüphaneler ele alınmaktadır.

3.1.1. Model ve Kütüphane Seçimi

Derin öğrenme ile nesne tespiti, bilgisayarla görme alanında büyük ilerlemeler kaydetmiştir. Convolutional Neural Networks, R-CNN, YOLO, SSD ve Mask R-CNN gibi yöntemler, bu alanda yüksek doğruluk ve performans sağlamıştır (K. He vd., 2017; Redmon & Farhadi, 2018). Bu yöntemler, TensorFlow, PyTorch, Detectron2, OpenCV, Darknet ve MMDetection gibi güçlü kütüphaneler ile desteklenmektedir. Hepsinin kendi avantajları ve dezavantajları bulunması ile birlikte tez çalışması için kütüphane olarak Detectron 2 seçilmiştir.

Tez çalışmasında, derin öğrenme ile nesne tanıma için oldukça güçlü bir kütüphane olan “detectron 2” kullanılmıştır. Detectron 2, Facebook AI Research (FAIR) tarafından geliştirilen bir bilgisayarlı görü modelleme platformudur. Özellikle nesne tespiti, nesne tanıma ve görüntü segmentasyonu gibi görevleri gerçekleştirmek için kullanılır. Detectron 2, PyTorch tabanlı bir derin öğrenme kütüphanesi üzerine inşa edilmiştir ve araştırmacıların ve geliştiricilerin ölçeklenebilir ve esnek bilgisayarlı görü modelleri oluşturmasına olanak tanır(Wu vd., 2019).

Kütüphanenin bazı temel özellikleri şunlardır:

- Modüler ve Esnek Yapı: Detectron 2, modüler bir yapıya sahiptir ve farklı veri kümeleri, mimariler ve hedef görevler için kolayca yapılandırılabilir.
- Önceden Eğitilmiş Modeller: Çeşitli önceden eğitilmiş modeller sunar ve bu modeller kullanılarak yeni veri kümeleri veya görevler için transfer öğrenme yapılabilir.
- Performans: Yüksek performanslı ve verimli bir şekilde çalışabilme özelliği vardır, bu da büyük veri kümeleriyle ve karmaşık görevlerle başa çıkabilir.
- Nesne Tespiti ve Segmentasyonu: Nesne tespiti, nesne tanıma ve görüntü segmentasyonu gibi görevlerde kullanılabilecek çeşitli önceden eğitilmiş modeller sunar.

Detectron 2 içerisinde birçok derin öğrenme modeli barındırmaktadır. Hazırlanmış veri seti ile modellerden biri seçilerek eğitimi yapılmakta ardından eğitilmiş model ile test yapılabilmektedir. Detectron 2 kütüphanesinde nesne tespiti için bulunan derin öğrenme modelleri şu şekildedir:

- fast_rcnn_R_50_FPN_1x.yaml
- faster_rcnn_R_50_C4_1x.yaml
- faster_rcnn_R_50_C4_3x.yaml
- faster_rcnn_R_50_DC5_1x.yaml
- faster_rcnn_R_50_DC5_3x.yaml
- faster_rcnn_R_50_FPN_1x.yaml
- faster_rcnn_R_50_FPN_3x.yaml
- faster_rcnn_R_101_C4_3x.yaml
- faster_rcnn_R_101_DC5_3x.yaml
- faster_rcnn_R_101_FPN_3x.yaml
- faster_rcnn_X_101_32x8d_FPN_3x.yaml
- retinanet_R_50_FPN_1x.yaml
- retinanet_R_50_FPN_3x.yaml
- retinanet_R_101_FPN_3x.yaml
- rpn_R_50_C4_1x.yaml
- rpn_R_50_FPN_1x.yaml

Modellerin hepsi nesne tespitinde kullanılabilir. Tez kapsamında mümkün olduğu kadar fazla derin öğrenme nesne tespiti metodu denenmeye çalışılmıştır. Araştırmalar sonucu en çok tercih edilen modeller bulunmuş bunların dört tanesi seçilmiştir.

Çalışma için seçilen modeller:

- fast_rcnn_R_50_FPN_1x
- faster_rcnn_R_50_FPN_1x
- retinanet_R_50_FPN_1x
- rpn_R_50_FPN_1x

Detectron2 çerçevesinde, fast-rcnn-R-50-FPN-1x, faster-rcnn-R-50-FPN-1x, retinanet-R-50-FPN-1x ve rpn-R-50-FPN-1x gibi dört farklı model uygulanmıştır. Bu modellerde kullanılan "R-50", ResNet-50 omurgasını ifade eder. ResNet, kalıntı ağı anlamına gelir ve ResNet-50, 50 katmanlı bir ResNet mimarisini belirtir. FPN, yani Özellik Piramidi Ağı, farklı ölçeklerdeki nesneleri tespit etme zorluğunu, farklı çözünürlük ve anlamsal anlamlara sahip özellik haritalarından oluşan bir piramit oluşturarak ele alır. '1x' terimi, modelin tüm eğitim veri seti üzerinde bir epoch boyunca eğitildiğini gösterir. Bu, modelin veri seti üzerinde iki veya üç kez eğitildiği '2x' veya '3x' eğitim programlarıyla karşılaştırılır. '1x' eğitim programı genellikle standart bir başlangıç noktası olarak kullanılmaktadır.

3.1.2. Eğitim ve Test

Veri setimiz 512x512 boyutlarında 96 görüntü içermektedir. Her bir görüntüdeki nesnelerin etrafına, 5x5 piksel boyutlarında sınırlayıcı kutular (bounding boxes) çizilmiş ve COCO formatında dosya haline getirilmiş dosyayı, görüntüler ile birlikte eğitim modeline verilmektedir. Nesne tespiti için dört farklı model kullanılmıştır: Fast R-CNN, Faster R-CNN, RetinaNet ve RPN. Bu modeller, aynı veri seti ve aynı parametreleri kullanılarak ayrı ayrı eğitilmiş ve test edilmiştir. Eğitim ve test sonuçlarını, her model için ayrı ayrı sunulmuştur.

Tablo 3.1: Modellerde Kullanılan Başlangıç Parametreleri

Parametre	Değer
BASE-LR	0.0001
MAX-ITER	10000
BATCH-SIZE-PER- IMAGE	16
NUM-CLASSES	2
PRE-NMS-TOPK- TRAIN	15000
POST-NMS-TOPK- TRAIN	1500

Tablo 3.1'de modellerde kullanılan başlangıç parametreleri verilmiştir. Kullanılan parametrelerin ayrı ayrı açıklamaları aşağıdadır.

Temel Öğrenme Oranı (Base-LR): Bu, ağırlıkların başlangıçta ne kadar hızla güncelleneceğini belirleyen bir parametredir. Öğrenme oranı, her güncelleme sırasında ağırlıkların ne kadar değiştirileceğini kontrol eder. Düşük bir öğrenme oranı, modelin daha yavaş ama daha kesin öğrenmesine yol açarken, yüksek bir öğrenme oranı daha hızlı fakat daha az kesin sonuçlara neden olabilir.

Maksimum İterasyon (Max-Iter): Bu, eğitim sürecinde yapılacak maksimum iterasyon (yineleme) sayısını belirtir. Bir iterasyon, tüm eğitim veri setinin bir kez modelden geçirilmesi anlamına gelir. Maksimum iterasyon sayısı, modelin eğitim süresince kaç kez eğitileceğini belirler.

Görüntü Başına Yığın Boyutu (Batch-Size-Per-Image): Bu, bir görüntü için kullanılan yığın (batch) boyutunu ifade eder. Yığın boyutu, her güncelleme için kullanılan örnek sayısını belirler. Daha büyük yığın boyutları, daha fazla bellek gerektirir, ancak eğitim süresini kısaltabilir. Bu parametre, modelin eğitim performansını ve bellek kullanımını dengelemeye yardımcı olur.

Sınıf Sayısı (Num-Classes): Bu, eğitilen modelin sahip olacağı sınıf sayısını belirtir. Nesne tespiti modelleri, önceden belirlenmiş bir nesne sınıfı setini tespit etmek için eğitilir. İki sınıf içeren bir model, pozitif ve negatif örnekler arasında ayrım yapar.

Pre-NMS-Topk-Train: Bu, eğitim sırasında NMS (Non-Maximum Suppression) adımından önce seçilecek en yüksek puanlı bölgelerin sayısını belirtir. Bu, bölge öneri sürecinde aday bölgelerin sayısını sınırlamak için kullanılır.

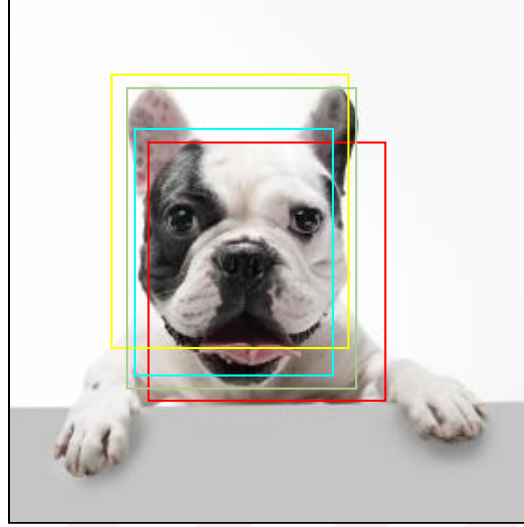
Post-NMS-Topk-Train: Bu, eğitim sırasında NMS adımından sonra seçilecek en yüksek puanlı bölgelerin sayısını ifade eder. Bu parametre, nihai aday bölgelerin belirlenmesinde ve çoklu nesne tespiti sürecinde kullanılan aday bölgelerin sayısını sınırlamak için kullanılır.

3.1.3. Değerlendirme Yöntemi

Görüntüler üzerinde nesne tespiti işleminde nesneler bir çerçeve ile görüntü üzerinde işaretlenmektedir. Bu işlemten sonra görüntüde işaretli nesnenin özellikleri ile sistem eğitilmektedir. Ardından model kullanılarak tahmin edilen nesne çerçeve içerisinde gösterilmektedir (Strafforello vd., 2024).

Teorik olarak olması gereken tam olarak yukarıda bahsedildiği adımlardır, ideal olarak tek çerçeve ile tespit edilen nesne gösterilir. Ancak tahmin işleminde tespit edilen bir nesne için çerçeve boyutunun tam nesneye oturmaması, birden fazla çerçeve çizilmesi gibi durumlar ortaya çıkmaktadır. Bu durumlarda bir hata yoktur, çerçeveler nesneyi doğru bulmuştur ancak birden fazla çerçeve olması ve çerçeve boyutu gibi durumlar değerlendirmeyi etkilemektedir (C. Liu vd., 2022).

Birden fazla çerçeve çizimi durumu nesne görüntü içerisinde aranırken birden fazla defa nesnenin bulunmasıdır. Böyle durumlar başarıyı düşürmektedir. **Şekil 3.1** bu konuda bir örnek verilmiştir.



Şekil 3.1: Üstüste Gelmiş Nesne Tespit Çerçeveleri

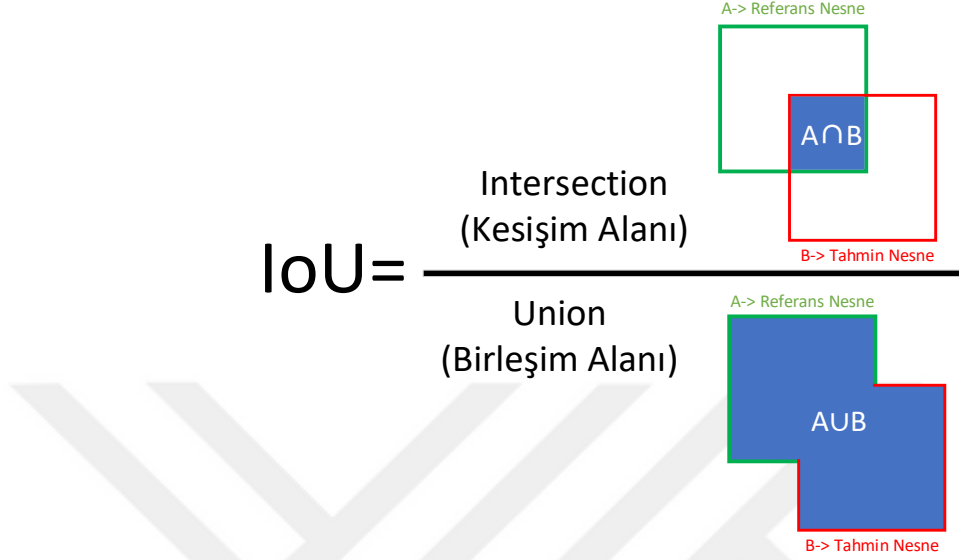
Bir nesne için birden fazla çerçeve çiziminin önüne geçilmesi için “maksimum olmayan bastırma (Non-Maximum Suppression)” yöntemi kullanılmaktadır. Bu yöntem üst üste gelen tahmin çerçevelerinden en uygun olanı bırakmaktadır (Hosang vd., 2017). En uygun seçiminde çeşitli teknikler kullanılabilmektedir: ilk bulunan, en merkezi olan, birleştirme, en yüksek orana sahip olan.

Çalışmada üst üste gelen çerçevelerden ilki seçilmiştir. Örneğin bir nesne üç defa tespit edilmiş ise üst üste üç çerçeve ile çizilerek gösterilmektedir. Çalışmada bu üç tespit aynı nesne olduğundan ilki seçilmektedir. Bu basit bir seçimdir ancak ilerleyen çalışmalarımızda başka seçim tekniklerini etkisi araştırılabilir. Bununla beraber nesne tipi ayrımı yapılmıştır. Yani bir çerçeve diğer bir çerçeve ile üst üste gelmesi durumunda eğer aynı tip nesne ise eleme yapılmaktadır. Fazladan elemeleri engellemektedir.

3.1.3.1. Referans ve Tahmin Nesnelerinde Birleşimin Kesişime Oranı

Nesne tespiti çalışmalarında referans nesnesi ve tahmin edilen nesnenin doğru konumda bulunup bulunmadığının tespiti için Birleşimin Kesişime Oranı (Intersection Over Union (IoU)) kullanılmaktadır. Bu çalışmada benzer yaklaşım kullanılmıştır. Birleşimin Kesişime Oranı (IoU), bir nesne algılama veya nesne tanıma modelinin performansını değerlendirmek için kullanılan bir metriktir. Bu

metrik, özellikle obje tespiti gibi görevlerde modelin tahminlerinin gerçek etiketlerle ne kadar iyi örtüştüğünü ölçmek için kullanılır (Zhou vd., 2019).



Şekil 3.2: Birleşimin Kesişime Oranı (IoU) Formülü

IoU, gerçek etiketlenmiş bir nesnenin (örneğin bir dikdörtgen kutu içindeki bir nesnenin) model tarafından tahmin edilen bölgenin ne kadar doğru olduğunu gösterir. İki bölgenin (gerçek etiketlenmiş bölge ve tahmin edilen bölge) birlikteki (intersection) ve birbirlerine göreli olarak kesişimdeki (union) alanların oranı olarak hesaplanır.

Şekil 3.2’de IoU formülü verilmiştir. Formüle göre "Intersection" bölgenin, hem gerçek etiketlenmiş bölge hem de tahmin edilen bölge içindeki kesişim alanını, "Union" bölgenin ise bu iki bölgenin toplam alanını ifade ettiği bir orandır.

Özellikle nesne algılama modellerinde, bu metrik, modelin nesneleri ne kadar doğru bir şekilde tespit ettiğini ve nesne sınırlarını ne kadar iyi tahmin ettiğini değerlendirmek için kullanılır. Yüksek IoU değerleri, modelin gerçek etiketlerle tespitlerde ve sınırlarda daha doğru olduğunu gösterirken, düşük IoU değerleri modelin hatalı veya kısmi tespitler yaptığını gösterebilir. IoU, nesne tespiti, nesne tanıma ve nesne sınıflandırma gibi görevlerde model performansının değerlendirilmesinde yaygın olarak kullanılan bir ölçümdür.

3.1.3.2. Ortalama Kesinlik ve Duyarlılık

Nesne tespiti performansını değerlendirmek için kullanılan iki önemli metrik olarak, Ortalama Kesinlik (OK) (Average Precision (AP)) ve Ortalama Hassasiyet (OH) (Average Recall (AR)) kullanılmaktadır. Doğru işaretleme çerçevesi ve tahmin çerçeveleri kullanılarak bulunan IoU değerleri ile hesaplanmaktadır. Özellikle, bir nesne algılama modelinin nesneleri doğru bir şekilde tespit edip etmediğini ve bu tespitlerin ne kadar kesin olduğunu değerlendirmek için önemlidir.

Ortalama Kesinlik: Nesne tespiti sırasında OK, modelin belirli bir eşik değerinde (genellikle farklı kesme değerleri veya eşiklerle) ne kadar doğru olduğunu ölçer. Precision-Recall eğrisi altındaki alanın (AUC - Area Under the Curve) bir ölçüsüdür. Yüksek OK değerleri, modelin farklı eşiklerde nesneleri tespit etme konusunda daha hassas olduğunu gösterir.

Ortalama Hassasiyet: AR, modelin belirli bir eşik değeri için ortalama duyarlılığını temsil eder. Yani, farklı eşik değerleri için elde edilen duyarlılık (recall) değerlerinin ortalamasıdır. Yüksek AR değerleri, modelin genellikle daha geniş bir yelpazedeki nesneleri tespit etme kabiliyetinin daha yüksek olduğunu gösterir.

Bu metrikler, nesne tespiti yapan bir modelin performansını ölçmek için kullanılır. Nesne tespiti modelleri genellikle kesinlik-duyarlılık eğrisi altındaki alanlar ve farklı eşik değerlerindeki duyarlılıkların ortalamalarını değerlendirerek değerlendirilir.

Yüksek OK ve OH değerleri, genellikle modelin nesneleri doğru bir şekilde tespit etme yeteneğinin daha iyi olduğunu gösterir. Bu metrikler, nesne tespiti modellerinin performansını anlamak ve farklı modeller arasında karşılaştırma yapmak için kullanılır.

3.1.3.3. Değerlendirme Çıktısı

Çalışmada sonuçların değerlendirmesi COCOEvaluator ile yapılmıştır. COCOEvaluator, önceden tanımlanmış COCO metriklerini hesaplamak ve modelin performansını analiz etmek için kullanılan bir araçtır. Bu, nesne algılama modellerinin geliştirilmesi ve karşılaştırılması için yaygın olarak kullanılan bir araçtır. Bu tür araçlar, modelin ne kadar iyi çalıştığını anlamak ve iyileştirmek için önemli bir geri bildirim sağlar.

Nesne tespiti metotlarının deęerlendirmelerinde genellikle doęruluk, kesinlik, duyarlılık gibi metrikler üzerinden yapılır. Bu metrikler referans ve tahmin nesnelerinin IoU deęeri ve tip örtüşmesine göre, modelin tahminlerinin gerçek etiketlerle karşılaştırılmasıyla hesaplanır. TP (True Positive - Doğru Pozitif), FP (False Positive - Yanlış Pozitif), FN (False Negative - Yanlış Negatif) gibi kavramlar üzerinden deęerlendirme yapılır ve bu sayede modelin performansı ölçülür.

Şekil 3.3'de yapılan çalışmadaki bir modelin testinin çıktısı verilmiştir. Görüldüğü gibi kesinlik ve duyarlılık deęerleri ayrı ayrı verilmiştir. Yanlarda bulunan "IoU=..." deęerleri ise referans ve tahmin nesne çerçevesinin örtüşme oranıdır. Örneğin IoU=0.50 iki nesne çerçevesinin %50 ve üzeri örtüştüğü anlamına gelir. Nesne tanıma çalışmalarında %50 nesne çerçevesinin doğru çizilmesi yeterli bir oran

```
Average Precision (AP) @[ IoU=0.50:0.95 | area= all | maxDets=2000 ] = 0.379
Average Precision (AP) @[ IoU=0.50 | area= all | maxDets=2000 ] = 0.804
Average Precision (AP) @[ IoU=0.75 | area= all | maxDets=2000 ] = 0.288
Average Precision (AP) @[ IoU=0.50:0.95 | area= small | maxDets=2000 ] = 0.379
Average Precision (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=2000 ] = -1.000
Average Precision (AP) @[ IoU=0.50:0.95 | area= large | maxDets=2000 ] = -1.000
Average Recall (AR) @[ IoU=0.50:0.95 | area= all | maxDets= 1 ] = 0.006
Average Recall (AR) @[ IoU=0.50:0.95 | area= all | maxDets= 10 ] = 0.063
Average Recall (AR) @[ IoU=0.50:0.95 | area= all | maxDets=2000 ] = 0.469
Average Recall (AR) @[ IoU=0.50:0.95 | area= small | maxDets=2000 ] = 0.469
Average Recall (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=2000 ] = -1.000
Average Recall (AR) @[ IoU=0.50:0.95 | area= large | maxDets=2000 ] = -1.000
```

Şekil 3.3: COCO Evaluator Çıktısı

olarak görülmektedir. Çalışmada %50 oranı karşılaştırmalarda kullanılmıştır.

3.1.4. Deney Sonuçların Gösterimi

Sonuçlar iki tablo şeklinde her iki veri seti için de gösterilmiştir. Sonuçlar özeti aşağıdaki gibidir.

- HFHS13
 - Dört derin öğrenme modelinin kesinlik, duyarlılık tablosu
 - Dallanma ve uç nokta sayıları tablosu
- NC15
 - Dört derin öğrenme modelinin kesinlik, duyarlılık tablosu
 - Dallanma ve uç nokta sayıları tablosu

Kesinlik ve Duyarlılık Tablosu:

Kesinlik ve duyarlılık sonuç tablosu özellikler ve model sonuçları içeren iki kısma sahiptir. Özellikler hangi tip değerlendirme olduğu ve bu değerlendirmenin özelliklerini barındırmaktadır.

- Özellikler
 - Değerlendirme Tipi
 - Ortalama kesinlik
 - Ortalama duyarlılık
- IoU
 - 0.50:0.95 => Tahmin nesnelerinin %50 ile %95 arası referans objeleriyle örtüşme oranı.
 - 0.50 => Tahmin nesnelerinin %50 referans objeleriyle örtüşme oranı.
 - 0.75 => Tahmin nesnelerinin %75 referans objeleriyle örtüşme oranı.
- Alan
 - Tüm, büyük, orta, küçük alanlar olarak karşılaştırma.
- Maksimum Tahmin
 - En fazla kaç tahmin üzerinden değerlendirildiği
- Model sonuçları
 - Hangi derin öğrenme modeli ile ilgili sonuç gösteriliyorsa başlıkta yazılmıştır.

Dallanma ve Uç Nokta Sayıları Tablosu:

Tablo 3.2'da referans nesne sayıları ile tahmin edilen nesne sayıları nicel olarak gösterilmiştir. Gösterimde dallanma ve uç nokta sayıları her model için ayrı ayrı gösterilmiştir. Bununla birlikte ham tahmin ile filtrelenmiş tahmin de ayrı ayrı verilmiştir. Ham tahminde bir nesneden birden fazla çerçeve çizilebilmektedir, filtrelenmiş tahmin “maksimum olmayan bastırma” algoritması uygulanmış halidir.

- Belge ismi
 - İlgili görüntü adı => hangi görüntünün sonuçları olduğunu gösterir.
- Model isimleri
 - Hangi derin öğrenme modeli ile çalışıldığını gösterir.
- Kısaltmalar
 - GT_BPC => Grant Truth Branch Point Count (Referans Dallanma Noktası Sayısı)
 - RD_BPC => Raw Detection Branch Point Count (Ham Tespit Dallanma Noktası Sayısı)
 - FI_BPC => Filtered Branch Point Count (Filtrelenmiş Dallanma Noktası Sayısı)
 - GT_TPC => Grant Truth Tip Point Count (Referans Uç Nokta Sayısı)
 - RD_TPC => Raw Detection Tip Point Count (Ham Tespit Uç Nokta Sayısı)
 - FI_TPC => Filtered Tip Point Count (Filtrelenmiş Uç Nokta Sayısı)

3.1.5. Deneysel Çalışma 1

Veri Seti:

- HFHS13
 - 512x512 boyut
 - 76 Eğitim görüntüsü
 - 20 Test görüntüsü

Kullanılan derin öğrenme modelleri:

- fast_rcnn_R_50_FPN_1x
- faster_rcnn_R_50_FPN_1x
- retinanet_R_50_FPN_1x
- rpn_R_50_FPN_1x

Tablo 3.2'de tüm derin öğrenme modellerine bakıldığında “retinanet” harici diğer modellerin %50 örtüşme kesinlik skorlarının 0.95’ten büyük olduğunu görülmektedir. Bununla beraber duyarlılık değerleri max 2000 nesne için “small” ve “all” özellikleri için 0.65 üzeri geldiği görülmektedir. Bu da fast_rcnn_R_50_FPN_1x, faster_rcnn_R_50_FPN_1x, rpn_R_50_FPN_1x modellerinin başarılı olduğunu ancak retinanet’in başarımının düşük olduğunu göstermektedir.

Tablo 3.2: HFHS13 veri kümesi üzerinde Dört Model Tarafından Elde Edilen Ortalama Kesinlik ve Duyarlılık Değerleri

HFHS13							
Özellikler				Model Sonuçları			
Değerlendirme Tipi	IoU	Area	MaxDets	fast_rcnn R_50 FPN_1x	faster_rcnn R_50 FPN_1x	retinanet R_50 FPN_1x	rpn R_50 FPN_1x
Average Precision (AP)	0.50:0.95	all	2000	0.592	0.596	0.121	0.585
Average Precision (AP)	0.5	all	2000	0.967	0.963	0.369	0.962
Average Precision (AP)	0.75	all	2000	0.7	0.709	0.037	0.662
Average Precision (AP)	0.50:0.95	small	2000	0.592	0.596	0.121	0.585
Average Precision (AP)	0.50:0.95	medium	2000	-1	-1	-1	-1
Average Precision (AP)	0.50:0.95	large	2000	-1	-1	-1	-1
Average Recall (AR)	0.50:0.95	all	1	0.009	0.009	0.003	0.01
Average Recall (AR)	0.50:0.95	all	10	0.085	0.088	0.026	0.088
Average Recall (AR)	0.50:0.95	all	2000	0.684	0.681	0.303	0.676
Average Recall (AR)	0.50:0.95	small	2000	0.684	0.681	0.303	0.676
Average Recall (AR)	0.50:0.95	medium	2000	-1	-1	-1	-1
Average Recall (AR)	0.50:0.95	large	2000	-1	-1	-1	-1

Tablo 3.3'te her görüntü için ayrı ayrı nicel değerler görülmektedir. Ham değerler ile filtrelenmiş değerlere bakılınca filtreleme ihtiyacı gözlemlenmektedir.

Üç öğrenme modelinin filtrelenmiş tespit edilen nesne sayıları referans görüntü değerine yakın olduğu görülmektedir. “Retinanet” başarısızlığı burada da görülmektedir.

Tespit edilen noktaların sayısını verdiğimiz Tablo 3'te, önce hibe-gerçek dalların ve uç noktaların sayısını verdik. Her satırda dört modelin ham ve filtrelenmiş tahminlerini görebilirsiniz. NMS'nin 5x5 piksel çerçeve boyutuna sahip çalışmalarda post-processing olarak uygulanması gerektiği bu tabloda daha net anlaşılabilir.

Tablo 3.3: HFHS13 veri kümesi üzerinde Dört Model Tarafından Elde Edilen Dallanma ve Uç Nokta Sayıları Tablosu

No	fast_rcnn R_50_FPN_1x						faster_rcnn R_50_FPN_1x						retinanet R_50_FPN_1x						rpn R_50_FPN_1x					
	GT_BPC	RD_BPC	FL_BPC	GT_TPC	RD_TPC	FL_TPC	GT_BPC	RD_BPC	FL_BPC	GT_TPC	RD_TPC	FL_TPC	GT_BPC	RD_BPC	FL_BPC	GT_TPC	RD_TPC	FL_TPC	GT_BPC	RD_BPC	FL_BPC	GT_TPC	RD_TPC	FL_TPC
z72	97	147	100	146	169	152	97	127	96	146	160	149	97	201	201	146	229	229	97	143	101	146	157	150
z78	9	18	9	70	94	73	9	14	10	70	94	74	9	45	45	70	215	215	9	12	9	70	86	75
z65	80	113	81	143	153	143	80	109	82	143	157	145	80	164	164	143	237	237	80	112	82	143	154	147
z95	62	86	58	142	165	140	62	80	59	142	161	142	62	149	149	142	235	235	62	76	60	142	156	141
z27	52	82	52	118	130	117	52	74	51	118	137	121	52	128	128	118	293	293	52	84	54	118	129	118
z16	70	100	69	182	199	184	70	90	64	182	202	183	70	112	112	182	277	277	70	106	67	182	193	177
z88	0	0	0	6	8	6	0	0	0	6	9	6	0	59	59	6	374	371	0	0	0	6	8	6
z23	78	119	77	163	178	166	78	120	78	163	183	163	78	173	173	163	255	255	78	125	76	163	178	165
z35	27	42	29	71	74	72	27	39	27	71	77	74	27	129	129	71	241	240	27	42	27	71	73	71
z91	41	59	40	149	179	151	41	54	41	149	175	154	41	98	98	149	258	258	41	49	38	149	164	149
z07	18	32	20	46	59	49	18	29	18	46	58	48	18	172	172	46	224	224	18	30	21	46	53	50
z47	65	94	66	132	152	132	65	87	64	132	152	137	65	139	139	132	243	243	65	95	63	132	139	130
z40	53	82	55	119	129	119	53	77	52	119	134	122	53	124	124	119	239	239	53	76	53	119	129	120
z84	0	0	0	6	6	6	0	0	0	6	7	7	0	78	78	6	397	389	0	0	0	6	6	6
z11	40	67	41	114	132	116	40	64	40	114	138	117	40	124	124	114	233	233	40	67	43	114	125	112
z58	88	130	90	165	192	168	88	113	89	165	187	173	88	164	164	165	217	217	88	124	89	165	185	171
z53	52	72	52	167	193	173	52	71	51	167	186	175	52	95	95	167	282	282	52	73	53	167	186	171
z10	26	48	27	82	98	84	26	38	27	82	102	86	26	136	136	82	237	237	26	38	26	82	96	86
z80	1	3	2	11	11	11	1	1	1	11	13	12	1	96	96	11	333	327	1	1	1	11	12	11
z62	141	198	144	114	137	117	141	178	140	114	129	114	141	294	294	114	149	149	141	188	143	114	128	118

Sonuç tablolarında verilen sonuçlara dayanarak, nesne tespitinde ortalama kesinliğin, ortalama hassasiyetten (AR) daha yüksek olması, tipik olarak modelin

görüntüdeki tüm örnekleri yakalamaktan ziyade ayrıntılı öğeleri tam olarak tanımlama ve tanımlamada daha iyi olduğunu gösterir.

Kesinlik-Duyarlılık Ödünleşimi (Precision-Recall Trade-off): Kesinlik, model tarafından yapılan iyimser tahminlerin doğruluğunu ölçerken, geri çağırma veri setindeki pozitif örneklerin kapsamını ölçer. Kesinlik ve duyarlılık arasında genellikle bir denge vardır. Bir modelin veri setine bazı pozitif örnekleri dahil etmesi gerekse bile, daha az sayıda, daha doğru, iyimser tahminler yaparak yüksek doğruluk elde edebilir, bu da daha düşük bir geri çağırma ile yapılabilir.

Eşikleme (Thresholding): Nesne algılama modelleri, tahmin edilen bir nesnenin ne zaman pozitif olarak değerlendirilmeyeceğini belirlemek için genellikle güven eşiklerini kullanır. Bunu ayarlayarak modelin hassasiyeti ayarlanabilir. Daha yüksek bir ışık daha yüksek hassasiyete yol açar, ancak model pozitif tahminler yaparken daha seçici hale geldiğinden daha düşük geri çağırma sağlar.

Algılama Zorluğu (Detection Difficulty): Veri kümesindeki bazı nesneleri tespit etmek boyut, oklüzyon veya görünümdeki değişkenlik nedeniyle diğerlerine göre daha zor olabilir. Model, tespit edilmesi daha kolay nesneleri doğru bir şekilde tespit etmeye ve konumlandırmaya öncelik verebilir, bu da daha yüksek hassasiyete yol açar, ancak tespit edilmesi daha zor bazı örnekleri dahil etmesi gerekiyorsa potansiyel olarak daha düşük geri çağırma sağlar.

Sınıf Dengesizliği (Class Imbalance): Veri kümesi dengesizse, pozitif örneklerden çok daha fazla negatif örnek varsa, model negatif örneklerin pozitif olarak yanlış sınıflandırılmasını önlemek için yanlış pozitifleri en aza indirmeye öncelik verebilir. Model pozitif tahminler yapma konusunda seçiciyse, bu daha yüksek hassasiyete ancak potansiyel olarak daha düşük geri çağırmaya yol açabilir.

Özetle, nesne tespitindeki ortalama kesinliğin ortalama geri çağırmadan daha yüksek olması, modelin veri kümesindeki bazı pozitif örnekleri potansiyel olarak kaçırma pahasına bile olsa doğru iyimser tahminler yapmaya daha fazla odaklandığını göstermektedir. Bu değiş tokuş birçok makine öğrenimi görevinde yaygındır ve uygulamanın özel gereksinimlerine ve kısıtlamalarına göre değerlendirilmelidir.

3.1.6. Deneysel Çalışma 2

Veri Seti:

- NC15
 - 1032X1032 boyut
 - 175 Eğitim görüntüsü
 - 41 Test görüntüsü

Kullanılan derin öğrenme modelleri:

- fast_rcnn_R_50_FPN_1x
- faster_rcnn_R_50_FPN_1x
- retinanet_R_50_FPN_1x
- rpn_R_50_FPN_1x

Tablo 3.4: NC15 veri kümesi üzerinde Dört Model Tarafından Elde Edilen Ortalama Kesinlik ve Duyarlılık Değerleri

NC15							
Özellik				Model Sonuçları			
Değerlendirme Tipi	IoU	Area	MaxDets	fast_rcnn R_50 FPN_1x	faster_rcnn R_50 FPN_1x	retinanet R_50 FPN_1x	rpn R_50 FPN_1x
Average Precision (AP)	0.50:0.95	all	2000	0.384	0.372	0.025	0.328
Average Precision (AP)	0.5	all	2000	0.723	0.724	0.089	0.666
Average Precision (AP)	0.75	all	2000	0.365	0.32	0.006	0.234
Average Precision (AP)	0.50:0.95	small	2000	0.384	0.372	0.025	0.328
Average Precision (AP)	0.50:0.95	medium	2000	-1	-1	-1	-1
Average Precision (AP)	0.50:0.95	large	2000	-1	-1	-1	-1
Average Recall (AR)	0.50:0.95	all	1	0.006	0.006	0	0.006
Average Recall (AR)	0.50:0.95	all	10	0.06	0.058	0.005	0.06
Average Recall (AR)	0.50:0.95	all	2000	0.524	0.528	0.12	0.427
Average Recall (AR)	0.50:0.95	small	2000	0.524	0.528	0.12	0.427
Average Recall (AR)	0.50:0.95	medium	2000	-1	-1	-1	-1
Average Recall (AR)	0.50:0.95	large	2000	-1	-1	-1	-1

Tablo 3.4'te NC15 veri kümesi üzerinde bu dört model tarafından elde edilen ortalama kesinlik ve duyarlılık değerleri gösterilmektedir. Veriler incelendiğinde, genel olarak fast_rcnn_R_50_FPN_1x ve faster_rcnn_R_50_FPN_1x modellerinin diğer modellere göre daha iyi sonuçlar verdiği gözlemlenmektedir. Bununla birlikte,

retinanet_R_50_FPN_1x modelinin performansı NC15 veri setinde de düşük kalmıştır.

Tablo 3.5: NC15 veri kümesi üzerinde Dört Model Tarafından Elde Edilen Dallanma ve Uç Nokta Sayıları Tablosu

No	fast_rcnn R_50_FPN_1x						faster_rcnn R_50_FPN_1x						retinanet R_50_FPN_1x						rpn R_50_FPN_1x					
	GT_BPC	RD_BPC	FI_BPC	GT_TPC	RD_TPC	FI_TPC	GT_BPC	RD_BPC	FI_BPC	GT_TPC	RD_TPC	FI_TPC	GT_BPC	RD_BPC	FI_BPC	GT_TPC	RD_TPC	FI_TPC	GT_BPC	RD_BPC	FI_BPC	GT_TPC	RD_TPC	FI_TPC
z077	53	73	48	264	318	258	53	86	48	264	330	262	53	48	48	264	515	515	53	43	31	264	286	251
z192	29	48	31	169	201	172	29	50	25	169	201	170	29	34	34	169	418	418	29	18	14	169	184	166
z061	62	84	52	252	312	251	62	107	63	252	316	250	62	41	41	252	493	493	62	49	34	252	282	245
z051	61	107	56	245	301	241	61	125	70	245	302	245	61	94	94	245	499	499	61	43	31	245	286	237
z134	61	77	46	215	263	212	61	91	56	215	271	214	61	98	98	215	427	427	61	47	29	215	240	205
z220	26	66	29	146	185	142	26	63	31	146	181	146	26	88	88	146	425	425	26	29	19	146	165	139
z104	38	61	38	224	274	227	38	86	45	224	284	228	38	37	37	224	486	486	38	33	23	224	241	212
z108	40	57	32	210	260	216	40	86	46	210	276	217	40	38	38	210	479	479	40	30	24	210	236	205
z178	40	57	36	224	283	222	40	68	40	224	294	227	40	56	56	224	451	451	40	27	17	224	250	219
z044	77	112	69	254	295	243	77	131	70	254	303	246	77	71	71	254	528	528	77	67	44	254	268	230
z063	55	92	54	253	304	248	55	97	53	253	336	261	55	66	66	253	488	488	55	43	26	253	279	242
z181	39	62	37	203	239	199	39	89	47	203	249	202	39	61	61	203	425	425	39	30	20	203	227	197
z176	41	59	37	216	260	210	41	84	45	216	267	216	41	52	52	216	447	447	41	24	18	216	240	209
z071	64	94	62	270	324	269	64	122	73	270	344	272	64	70	70	270	515	515	64	54	40	270	304	262
z139	38	55	33	218	255	213	38	75	42	218	272	212	38	58	58	218	458	458	38	35	23	218	251	207
z210	35	52	32	199	254	200	35	54	32	199	254	201	35	53	53	199	476	476	35	29	22	199	222	197
z012	54	105	56	88	130	87	54	177	84	88	141	90	54	349	349	88	323	323	54	70	40	88	124	87
z169	46	61	44	230	270	224	46	86	46	230	278	229	46	37	37	230	481	480	46	28	21	230	251	219
z165	55	103	56	230	296	237	55	108	55	230	297	237	55	83	83	230	480	480	55	48	29	230	278	231
z222	27	45	28	155	200	157	27	51	28	155	210	163	27	60	60	155	389	389	27	28	19	155	177	155
z119	55	86	45	209	243	208	55	92	47	209	253	212	55	72	72	209	453	453	55	53	31	209	231	202
z212	42	56	30	196	220	187	42	75	40	196	243	195	42	62	62	196	426	426	42	36	24	196	213	188
z205	24	51	26	172	216	171	24	54	26	172	215	173	24	49	49	172	443	443	24	18	15	172	202	171
z056	65	109	59	253	302	250	65	116	64	253	321	257	65	88	88	253	487	487	65	44	30	253	284	247
z152	40	72	39	200	240	197	40	84	43	200	250	201	40	57	57	200	436	436	40	38	29	200	222	194
z093	60	80	47	245	288	235	60	103	58	245	292	239	60	46	46	245	464	464	60	51	36	245	268	231
z032	81	121	72	264	314	253	81	161	83	264	351	267	81	77	77	264	570	570	81	77	56	264	297	247
z100	47	74	42	233	264	222	47	79	46	233	291	235	47	86	86	233	461	461	47	28	23	233	256	221
z039	64	94	57	296	359	290	64	116	62	296	364	287	64	58	58	296	574	574	64	47	35	296	312	264

Tablo 3.5'te gösterildiği gibi, dallanma ve uç nokta sayıları her model için ayrı ayrı verilmiştir. Ham tespit değerleri ile filtrelenmiş tespit değerleri

karşılaştırıldığında, filtreleme işleminin sonuçlar üzerinde önemli bir etkisi olduğu görülmektedir. Özellikle, retinanet_R_50_FPN_1x modelinin bu veri kümesindeki başarısızlığı filtrelenmiş sonuçlarda da belirgin bir şekilde ortaya çıkmıştır.

Deney 2 sonuçlarına göre, NC15 veri kümesi üzerinde en yüksek performansı fast_rcnn_R_50_FPN_1x ve faster_rcnn_R_50_FPN_1x modelleri göstermiştir. Diğer modellerin performansının düşük olduğu ve bu veri kümesinde daha fazla iyileştirmeye ihtiyaç duyulduğu söylenebilir.

3.1.7. Zaman Karşılaştırması

Bu bölümdeki amaç, farklı derin öğrenme modelleri ile geleneksel görüntü işleme tekniklerinin uygulama sürelerini karşılaştırmaktır. Bu karşılaştırma, bu modellerin uygulama süreleri açısından verimliliği hakkında fikirler sunmayı amaçlamaktadır.

Test için kullanılan sistem:

- İşletim Sistemi: Ubuntu
- İşlemci: Intel I7-12700H
- Ekran Kartı: RTX3070TI 8GB
- Ram: 16 GB

Karşılaştırma yönteminde, HFHS13 veri seti için eğitim süresini ve tüm test verileri için toplam nesne tespit süresini verilmiştir. Ardından, farklı bir veri seti olan KUVESG veri seti ile yaptığımız test sürelerini gösterilmiştir. Sonuçlar **Tablo 3.6** ve **Tablo 3.7**'da gösterilmiştir. (GGİ – Geleneksel Görüntü İşleme).

Tablo 3.6: HFHS13 (20 görüntü) Eğitim ve Test Süreleri Karşılaştırması

Model	Eğitim Süresi	Test Süresi (SS:DD:SS)
Fast R-CNN	2:22:44	00:00:03:781
Faster R-CNN	2:19:14	00:00:03:762
RetinaNet	0:36:48	00:00:01:788
RPN	2:21:38	00:00:03:726
GGİ	N/A	00:00:00:780

Tablo 3.7: KUVESG (156 görüntü) Test Süresi Karşılaştırması

Model	Test Süresi (SS:DD:SS)
Fast R-CNN	00:00:23:225
Faster R-CNN	00:00:22:729
RetinaNet	00:00:09:263
RPN	00:00:22:740
GGI	00:00:01:116

Sonuçlara göre, Fast R-CNN, Faster R-CNN ve RPN modelleri benzer eğitim ve test süreleri sunmaktadır. Bu modeller arasında süre açısından çok az fark bulunmaktadır. RetinaNet modeli daha kısa sürede çözüm sağlasa da, daha önce gördüğümüz gibi, RetinaNet modelinin tahmin skorları düşük olduğundan dikkate alınmayabilir. Geleneksel yöntem, nesne tespitinde en düşük performans süresini göstermiştir. Bunu küçük bir veri seti olarak yorumluyoruz. Daha önce karmaşıklık formülünü verdiğimiz geleneksel yöntemin performans süresinin üstel olarak artması beklenmektedir.

Karşılaştırma, derin öğrenme modelleri ile geleneksel görüntü işleme teknikleri arasındaki ödünleşimleri vurgulamaktadır. Derin öğrenme modelleri ölçeklenebilirlik ve doğruluk sunarken, daha yüksek eğitim sürelerine ve karmaşıklığa sahiptir. Geleneksel görüntü işleme, küçük veri setlerinde daha hızlıdır ve uygulanması daha basittir, ancak daha büyük veya daha karmaşık veri setlerinde ölçeklenebilirlik ve doğruluk konusunda sıkıntı yaşar. Bu yaklaşımlar arasında seçim yaparken, hız, doğruluk, ölçeklenebilirlik ve kaynakların kullanılabilirliği arasında denge sağlanmalıdır.

3.1.8. Bulgular ve Tartışma

Dört farklı modelin aynı parametreler ve nesne çerçevesi ile iki farklı veri seti üzerinde çalıştırılması farklı sonuçlar verdiği görülmektedir. Buna yol açabilecek sebepler aşağıda açıklanmaktadır.

Görüntü Boyutu ve Çözünürlüğü:

Deney 1'de kullanılan HFHS13 veri seti 512x512 boyutunda görüntüler içerirken, Deney 2'de kullanılan NC15 veri seti 1032x1032 boyutundadır. Bu fark,

her iki deneyde de nesne boyutları sabit olmasına rağmen, modellerin daha büyük görüntü alanlarında nesneleri tespit etme konusunda zorlanmasına yol açmış olabilir. Daha büyük bir görüntüde, 5x5 boyutundaki küçük nesneleri doğru bir şekilde tespit etmek, model için daha zorlu hale gelebilir.

Görüntüdeki Nesne Sayısı:

Daha büyük boyutlu görüntüler, aynı alan içerisinde daha fazla sayıda nesne barındırabilir. Bu durumda, modellerin bu nesneleri tespit etme ve ayırt etme konusunda daha fazla işlem yapması gerekebilir. Deney 2'de daha fazla sayıda nesnenin bulunması, özellikle küçük nesneleri tespit etme performansını olumsuz etkileyebilir.

Model Mimarisi ve Performans:

Modellerin farklı veri setleri üzerinde nasıl performans gösterdiği de önemlidir. Deney 1'deki küçük boyutlu görüntüler, modellerin daha az karmaşık işlemler yapmasını sağlarken, Deney 2'deki daha büyük görüntüler, modelin daha fazla sayıda nesneyi ve ayrıntıyı işlemek zorunda kalmasına neden olabilir. Bu durum, özellikle daha büyük ve karmaşık veri setlerinde RetinaNet gibi modellerin performansının düşmesine neden olmuş olabilir.

Görüntü Çözünürlüğü ve Detay Seviyesi:

Deney 2'deki daha yüksek çözünürlüklü görüntüler, daha fazla ayrıntı içerebilir ve bu da modellerin bu ayrıntıları doğru bir şekilde tanımlamasını zorlaştırabilir. Yüksek çözünürlüklü görüntülerde, 5x5 piksel boyutundaki nesneleri doğru bir şekilde tespit etmek, özellikle küçük nesneler arasında ayırım yapmayı gerektiren durumlarda zorlayıcı olabilir.

NMS ve Post-Processing Etkisi:

Her iki deneyde de NMS (Non-Maximum Suppression) algoritmasının kullanılması, özellikle yoğun nesne tespiti durumlarında önemli bir rol oynayabilir. Deney 2'deki daha büyük boyutlu ve yüksek çözünürlüklü görüntülerde, birden fazla çerçevenin aynı nesne üzerinde tespit edilme olasılığı artırmaktadır. Bu durum, NMS algoritmasının daha fazla hata yapmasına neden olabilir ve modelin başarımını olumsuz etkileyebilmektedir.

Veri Setindeki Çeşitlilik ve Zorluk Derecesi:

Veri setlerinin içerdği nesne türleri ve zorluk dereceleri de modellerin performansında farklılık yaratabilir. NC15 veri setindeki görüntülerin daha karmaşık yapıya sahip olması veya daha zorlu koşullar içermesi, Deney 2'deki başarımların farklarını açıklayabilir.

Sonuç olarak, her iki deneyde de nesnelerin boyutları ve nesne çerçevesi boyutları sabit tutulmuş olsa da görüntü boyutu, çözünürlük, model mimarisi ve veri setinin karmaşıklığı gibi faktörler, Deney 1 ve Deney 2 arasındaki başarımların farklarının başlıca nedenleri olabilir.

3.2. Hiperparametre Optimizasyonu

Hiperparametre ayarı, tıbbi görüntüleme için derin öğrenme modellerinin optimize edilmesinde kritik bir rol oynar. Bu, modelin performansını optimize etmek için çeşitli model parametrelerini titizlikle ayarlamayı içerir, karmaşıklık, eğitim verimliliği ve tahmin doğruluğu arasında bir denge kurar. Uygun bir ayar, bir modelin eğitim verilerinden görünmeyen vakalara genelleme yeteneğini önemli ölçüde artırabilir, aşırı uyumu azaltabilir ve güvenilirliği artırabilir. Ayrıca, hesaplama kaynaklarının verimli bir şekilde kullanılmasını sağlar, eğitim sürelerinin hızlanmasına izin verirken modelin performansını da düşürmez. Hiperparametrelerin ince ayarı, araştırmacıların modelleri belirli veri kümelerine ve görevlere uyarlamalarını sağlayarak gerçek dünya tıbbi senaryolarında başarımlarını daha da artırır.

Model parametrelerinden farklı olarak, hiperparametreler eğitimden önce belirlenir ve modelin eğitim verilerinden genelleme yapabilme yeteneğini önemli ölçüde etkileyebilir. Yaygın hiperparametreler arasında öğrenme oranları, batch boyutları ve iterasyon sayısı yer alır. Bu hiperparametrelerin en uygun kombinasyonunu bulmak hesaplama açısından yoğun olabilir, ancak makine öğrenimi modellerinin doğruluğunu ve verimliliğini maksimize etmek için gereklidir (Bartz vd., 2022).

Hiperparametre optimizasyonu için her biri kendi avantaj ve dezavantajlarına sahip birkaç teknik vardır. Grid arama, belirli bir aralıkta her hiperparametre kombinasyonunu denemeyi içeren brute-force bir yöntemdir. Bu yöntem etkili

olabilir ancak genellikle hesaplama açısından pahalıdır. Öte yandan, “random search” rastgele hiperparametre kombinasyonlarını seçer ve genellikle daha geniş bir değer yelpazesi araştırarak daha verimli bir yaklaşım sunar. Daha ileri teknikler, örneğin Bayesian optimizasyonu, yeni hiperparametre setlerinin performansını tahmin etmek için önceki değerlendirme sonuçlarını kullanarak daha az iterasyonla en uygun ayarları bulmayı hedefler. Genetik algoritmalar ve gradyan tabanlı optimizasyon gibi diğer yöntemler de hiperparametre alanında gezinmek için sofistike araçlar sunar (Bischl vd., 2024).

Bu çalışmada, bir makine öğrenimi modeli için üç kritik hiperparametrenin optimizasyonuna odaklanılmıştır: öğrenme hızı (BASE LR), maksimum iterasyon sayısı (MAX ITER) ve görüntü başına batch boyutu (BATCH SIZE PER IMAGE). Ana hedef, modelin performansını üç temel metrikle iyileştirmektir: AP IoU 0.50, AP IoU 0.75 ve AR. Hiperparametre optimizasyonu, hiperparametre alanını verimli bir şekilde aramak için en son teknikleri kullanan güçlü ve esnek bir yazılım çerçevesi (framework) olan Optuna kullanılarak gerçekleştirilmiştir.

3.2.1. Metodoloji

Optuna, makine öğrenimi için iyi bir şekilde uygun olan otomatik bir hiperparametre optimizasyon yazılım çerçevesidir. Dinamik bir arama alanına izin veren ve optimizasyon sırasında ayarlanabilir bir yapı sunan bir “define-by-run” yaklaşımına sahiptir. Optuna'nın temel optimizasyon algoritması, olasılıksal bir model oluşturur ve bu modeli, değerlendirilecek en umut verici hiperparametreleri seçmek için kullanır. Bu yöntem, grid arama ve rastgele arama gibi geleneksel yöntemlere kıyasla daha az iterasyonla en uygun hiperparametre setini verimli bir şekilde bulmaya yardımcı olur. Ayrıca, Optuna'nın esnekliği ve ölçeklenebilirliği hem küçük ölçekli deneyler hem de büyük ölçekli üretim sistemleri için mükemmel bir seçimdir.

Bu çalışmada, başarılı modellerden biri (faster RCNN) seçilmiş eğitim ve testler bu modelle yapılmıştır. Base Learning Rate (Base LR), Maksimum İterasyon Sayısı (Max Iter) ve Görüntü Başına Batch Boyutu (Batch Size Per Image) olmak üzere üç kritik hiperparametreye odaklanılmıştır. **Tablo 3.8**'de parametre uzayı verilmiştir. Tüm parametreler 210 olası kombinasyon oluşturmaktadır.

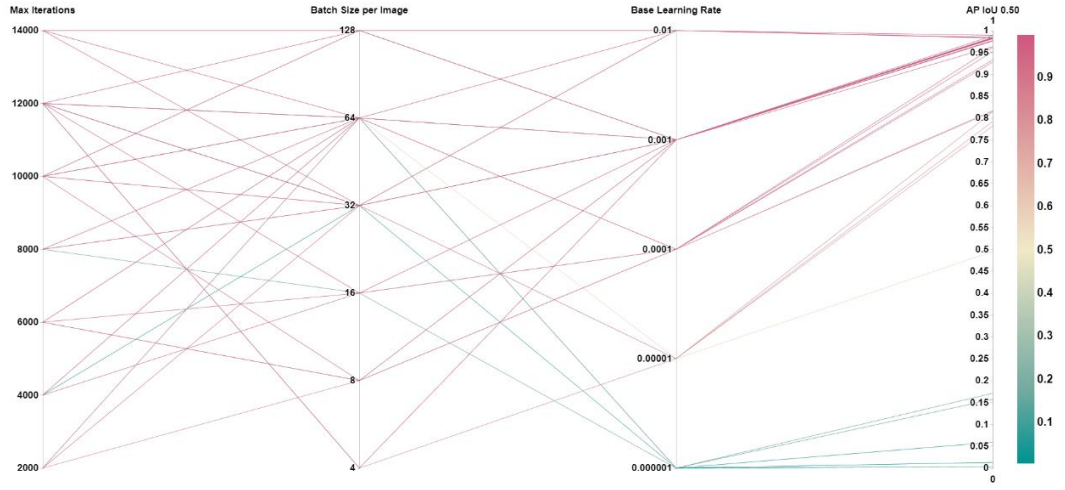
Tablo 3.8: Parametre Arama Uzayı

Parametre	Değerler
Base Lr	1e-02, 1e-03, 1e-04, 1e-05, 1e-06
Max Iter	2000,4000,6000, 8000, 10000, 12000, 14000
Batch Size Per Image	4, 8, 16, 32, 64, 128

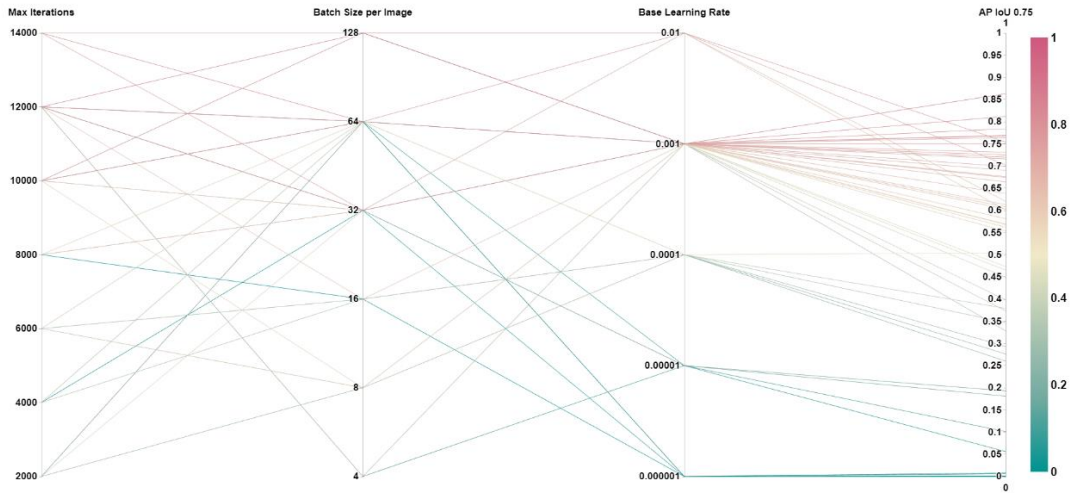
Hiperparametre aramasını, her biri benzersiz bir hiperparametre kombinasyonunu temsil eden birden fazla deneme üzerinden Optuna ile gerçekleştirilmiştir ve Optuna'nın optimizasyon algoritmasını kullanarak stratejik olarak seçilen 50 deneme yapılmıştır. Bu yöntem, hiperparametre alanını verimli bir şekilde keşfetmemizi ve en etkili yapılandırmaları tanımlamamızı sağladı. Bu sayede, modellerimizin doğruluğunu ve performansını artırarak sağlam ve güvenilir bir nesne algılama sistemi oluşturulmuştur. Optimizasyon süreci, her hiperparametrenin etkisine dair değerli bilgiler sağladı ve daha hedefli ve bilgilendirici bir optimizasyon stratejisine katkıda bulunmuştur.

3.2.2. Optimizasyon Çalışması

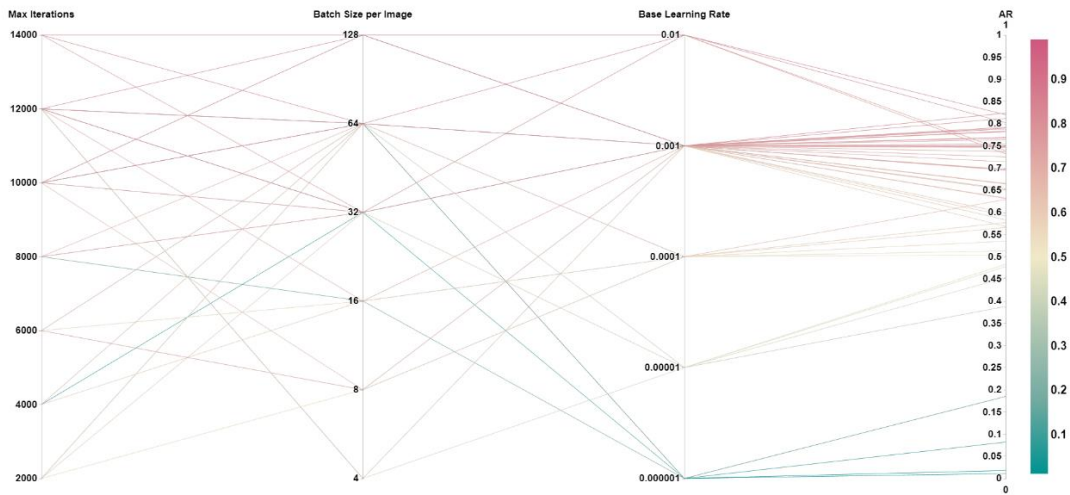
Hiperparametre optimizasyon sürecimizin kapsamlı sonuçlarını etkili bir şekilde göstermek için paralel koordinat grafiklerini kullanılmıştır. Her grafik, temel öğrenme oranı, maksimum iterasyonlar ve görüntü başına batch boyutu gibi üç kritik hiperparametrenin farklı performans metrikleri üzerindeki etkisini görselleştirdi: IoU eşik değerlerinde ortalama kesinlik ve ortalama duyarlılık oranı. Bu görselleştirmeler, hiperparametreler ve model performansı arasındaki karmaşık ilişkilerini vurgulanmıştır.



(a)



(b)



(c)

Şekil 3.4: Hiperparametrelerin Paralel Koordinat Grafikleri. (a) AP IOU 0,50, (b) AP IOU 0,75 ve (c) AR

İlk grafik olan **Şekil 3.4:(a)**, IoU 0.50'de AP'yi göstermektedir. Bu grafikte, farklı denemeleri temsil eden çizgiler, daha yüksek AP IoU 0.50 değerlerinin yaklaşık 0.001 öğrenme oranı ve özellikle 12000 ile 14000 arasında maksimum iterasyonlarla elde edildiğini göstermektedir. 32 ile 64 arasındaki batch boyutları da AP IoU 0.50 değerlerine olumlu katkıda bulunmaktadır.

Şekil 3.4:(b), IoU 0.75'te AP'ye odaklanmaktadır ve benzer eğilimler ortaya koymaktadır. En yüksek doğruluk değerleri yine yaklaşık 0.001 öğrenme oranı, daha yüksek maksimum iterasyonlar ve orta ile büyük batch boyutları ile ilişkilendirilmektedir. Farklı IoU eşik değerleri arasındaki bu tutarlılık, bu hiperparametre ayarlarının sağlamlığını vurgulamaktadır.

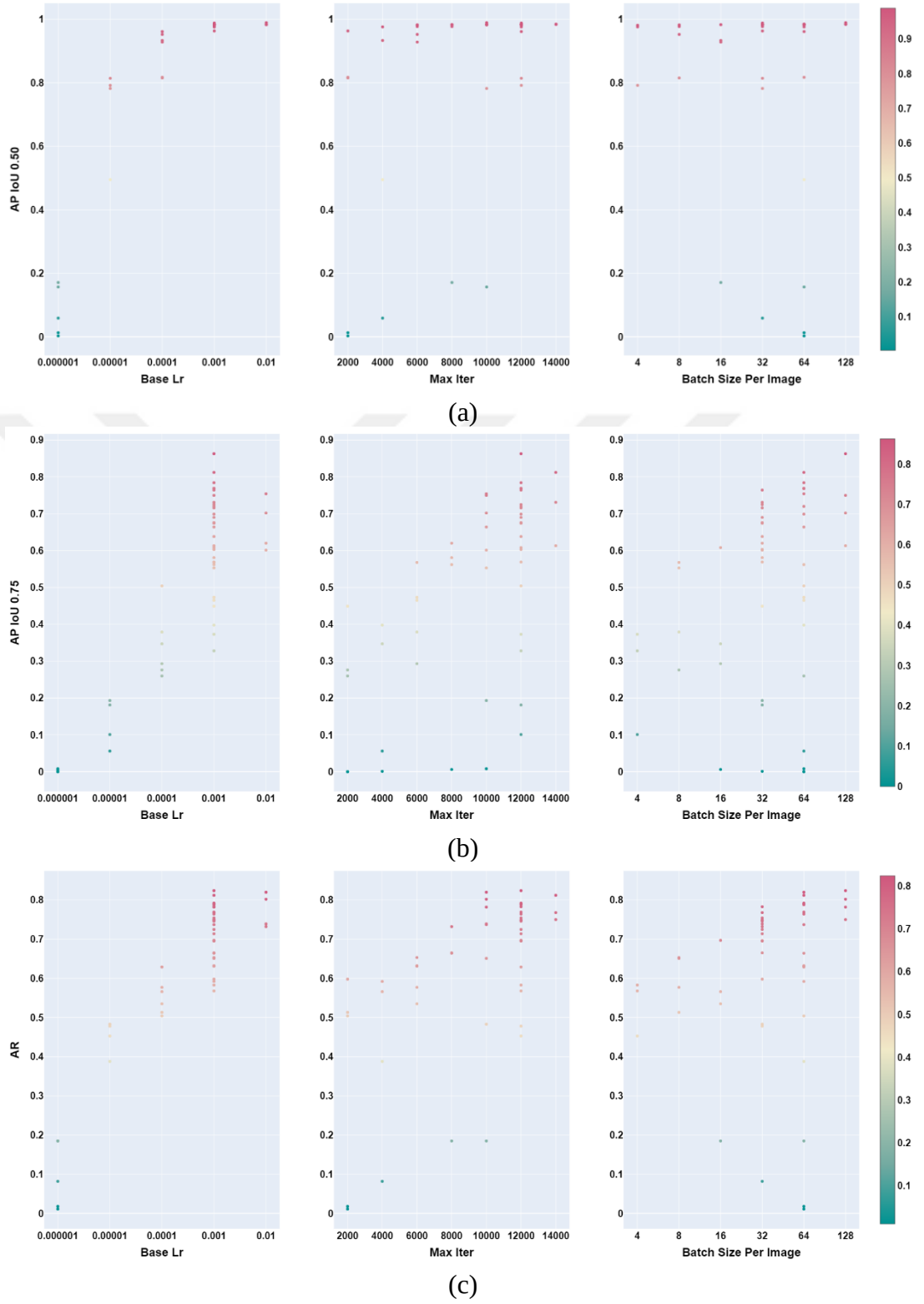
Şekil 3.4:(c), ise ortalama duyarlılık oranını göstermekte ve yine 0.001 öğrenme oranı, daha yüksek maksimum iterasyonlar ve 32 ile 64 arasındaki batch boyutlarının en iyi çağırma değerlerini sağladığını vurgulamaktadır.

Dağılım grafikleri, üç ana hiperparametrenin temel öğrenme oranı, maksimum iterasyonlar ve görüntü başına batch boyutu nesne algılama modellerimizin performans metrikleri üzerindeki etkisini görsel olarak temsil etmektedir.

Şekil 3.5:(a)'da IoU 0.50 için, veriler temel öğrenme oranı 0.001, maksimum iterasyonlar 10000 ile 14000 arasında ve 32 ile 64 arasındaki batch boyutlarının tutarlı bir şekilde yüksek ortalama doğruluk değerleri sağladığını göstermektedir. Bu kombinasyon, performans ölçeğinin üst ucuna doğru yoğunlaşan çizgilerle kanıtlandığı üzere optimal görünmektedir.

Şekil 3.5:(b), IoU 0.75 için benzer kalıplar göstermektedir. En yüksek doğruluk değerleri yaklaşık 0.001 öğrenme oranı, daha yüksek maksimum iterasyonlar ve 32 ile 64 arasındaki batch boyutları ile elde edilebilir. Farklı IoU eşik değerleri arasındaki bu tutarlılık, bu hiperparametre ayarlarının sağlamlığını göstermektedir.

Şekil 3.5:(c), Ortalama duyarlılığı göstermektedir ve yine aynı hiperparametre ayarlarının en iyi çağırma değerlerini sağladığını gözlemliyoruz. Bu, bu ayarların yalnızca doğruluğu artırmakla kalmayıp modelin ilgili nesneleri geri çağırma yeteneğini de geliştirdiğini göstermektedir.



Şekil 3.5: Hiperparametrelerin Dağılım Grafikleri. (a) AP IOU 0.50, (b) AP IOU 0.75 ve (c) AR

Genel olarak, dağılım grafikleri, nesne algılama modellerimiz için en iyi performansı sağlayan optimal hiperparametre yapılandırmalarını etkili bir şekilde vurgulamaktadır. Bu görselleştirme, daha hedefli ve bilgilendirilmiş bir optimizasyon stratejisini kolaylaştırarak modellerin doğruluğunu ve güvenilirliğini artırmaktadır.

Tablo 3.9: Deneylerin En İyi On Sonucu

Test	BASE LR	MAX ITER	BATCH SIZE PER IMAGE	AP IoU 0.50	AP IoU 0.75	AR
45	1E-02	10000	128	0.989	0.702	0.802
14	1E-03	12000	32	0.988	0.690	0.714
41	1E-03	12000	128	0.984	0.863	0.824
46	1E-02	10000	64	0.984	0.754	0.820
42	1E-03	14000	64	0.984	0.812	0.812
33	1E-03	12000	64	0.984	0.769	0.792
31	1E-03	12000	64	0.984	0.784	0.790
26	1E-03	12000	64	0.984	0.768	0.788
44	1E-03	10000	128	0.984	0.750	0.782

Tablo 3.9, hiperparametre optimizasyonunun en iyi on sonucunu sunmaktadır ve Bu sonuçlar sırasıyla AP IoU 0.50, AR ve AP IoU 0.75'e göre sıralanmıştır. En iyi yapılandırma, 1E-02 temel öğrenme oranı, 10000 maksimum iterasyon ve 128 batch boyutu ile elde edilmiş olup, 0.989 AP IoU 0.50, 0.702 AP IoU 0.75 ve 0.802 AR değerlerine ulaşılmıştır. Özellikle, 1E-03 temel öğrenme oranı ve 12000 maksimum iterasyon, birkaç üst düzey yapılandırmada sürekli olarak ortaya çıkmış ve bu yapılandırmaların model performansını artırmadaki önemini göstermiştir. Yüksek batch boyutları, özellikle 64 ve 128, algılama metriklerini olumlu yönde etkilemiştir.

3.2.3. En İyi Parametrelerin Test Edilmesi

Yapılan optimizasyon çalışması tek bir model ve HFHS13 veri seti ile yapılmıştır. Tespit edilen parametrelerin doğruluğunu test etmek için NC15 veri setini kullanarak bir eğitim ve test gerçekleştirdik. Eğitim için kullanılan parametreler **Tablo 3.9**'de sıralanmış en iyi sonuç olan kırk beşinci optimizasyon testinin parametreleridir. Base LR: 0.01, Maksimum iterasyon: 10000 ve Görüntü başına batch boyutu: 128 olarak girilmiştir.

Tablo 3.10: NC15 Veri Seti ile Optimize Edilmiş Hiperparametlerin Test Sonuçları

NC15					
Özellik				Model Sonuçları	
Değerlendirme Tipi	IoU	Area	MaxDets	faster_rcnn R_50 FPN_1x Optimizasyon Öncesi	faster_rcnn R_50 FPN_1x Optimize Edilmiş
Average Precision (AP)	0.50:0.95	all	2000	0.372	0.477
Average Precision (AP)	0.5	all	2000	0.724	0.905
Average Precision (AP)	0.75	all	2000	0.32	0.414
Average Precision (AP)	0.50:0.95	small	2000	0.372	0.477
Average Precision (AP)	0.50:0.95	medium	2000	-1	-1
Average Precision (AP)	0.50:0.95	large	2000	-1	-1
Average Recall (AR)	0.50:0.95	all	1	0.006	0.007
Average Recall (AR)	0.50:0.95	all	10	0.058	0.074
Average Recall (AR)	0.50:0.95	all	2000	0.528	0.622
Average Recall (AR)	0.50:0.95	small	2000	0.528	0.622
Average Recall (AR)	0.50:0.95	medium	2000	-1	-1
Average Recall (AR)	0.50:0.95	large	2000	-1	-1

Tablo 3.10'te yer alan sonuçlar, NC15 veri seti ile optimize edilmiş hiperparametrelerin test edilmesi sonucunda elde edilen model performansını göstermektedir. Optimize edilmemiş ve optimize edilmiş modeller arasındaki performans farkları açıkça gözlemlenebilir.

Ortalama Kesinlik:

IoU 0.5 tüm alanlar için, OK değeri optimize edilmeden önce 0.724 olarak ölçülmüş, ancak optimizasyon sonrasında 0.905'e kadar çıkmıştır.

IoU 0.75 için OK değeri ise optimize edilmeden önce 0.32 iken, optimizasyon sonrası 0.414'e yükselmiştir. Bu durum, modelin daha yüksek doğruluk gerektiren senaryolarda bile performansının iyileştiğini ortaya koyar.

Ortalama Hassasiyet:

Tüm alanlar ve IoU 0.50:0.95 için, optimize edilmeden önce AR değeri 0.528 iken, optimizasyon sonrasında bu değer 0.622'ye çıkmıştır. Bu artış, modelin nesneleri daha doğru bir şekilde hassasiyet kapasitesinin arttığını gösterir.

Sonuç olarak, **Tablo 3.10**'te görüldüğü gibi, hiperparametre optimizasyonu NC15 veri seti üzerinde önemli bir iyileşme sağlamıştır. Özellikle OK ve OH değerlerinde gözlemlenen iyileşmeler, optimize edilen modelin daha doğru ve güvenilir tahminler ürettiğini gösterir. Bu, optimize edilmiş modelin gerçek dünya senaryolarında daha iyi performans gösterebileceğini ve daha güvenilir sonuçlar sağlayabileceğini vurgulamaktadır.

3.2.4. Bulgular ve Tartışma

Beyin damar analizi için derin öğrenmeye dayalı nesne tespiti üzerine yapılan hiperparametre optimizasyonu çalışması, model performansını ve tanısallık verimliliği artırmada önemli ilerlemeler göstermiştir. Araştırmanın bu kısmı, temel öğrenme oranı, maksimum iterasyonlar ve görüntü başına batch boyutu gibi kritik hiperparametrelere odaklanarak, Optuna yazılım çerçevesini ile 210 olası kombinasyonu incelemiş ve stratejik olarak seçilmiş 50 deneme gerçekleştirmiştir. Bu yaklaşım, hiperparametre alanının verimli bir şekilde keşfedilmesini sağlayarak, modelin ortalama kesinlik ve ortalama duyarlılık oranını önemli ölçüde iyileştiren optimal yapılandırmaların bulunmasına olanak tanımıştır.

Paralel koordinat grafikleri, hiperparametreler ile performans metrikleri arasındaki ilişkileri kapsamlı bir şekilde görselleştirmektedir. Sonuçlar, 0.001 temel öğrenme oranı, daha yüksek maksimum iterasyonlar (özellikle 12000 ila 14000 arasında) ve 32 ila 64 arasında batch boyutlarının AP IoU 0.50, AP IoU 0.75 ve AR için en iyi sonuçları verdiğini göstermiştir. Bu ayarlar, sürekli olarak daha yüksek doğruluk ve çağırma oranları ile ilişkilendirilmiş ve bu hiperparametre yapılandırmalarının sağlamlığı ve güvenilirliği vurgulanmıştır.

Buna ek olarak, dağılım grafikleri bu hiperparametrelerin etkisini daha da açıklığa kavuşturmuş, paralel koordinat grafikleriyle uyumlu net eğilimler göstermiştir. Belirtilen optimal ayarlar etrafında yoğunlaşan yüksek performanslı denemeler, seçilen hiperparametrelerin model performansını artırmadaki etkinliği gösterilmiştir. Sonuç olarak böyle bir çalışma yapılması durumunda çalışmamızın bu kısmında tavsiye ettiğimiz parametreler ile başlanabilir. Bu zamandan tasarruf sağlayacaktır. Ek olarak başka parametreler de optimizasyon işlemine katılabilir ancak ileriki başka bir çalışmanın konusu olabilir.

DÖRDÜNCÜ BÖLÜM

4. PROGRAM GELİŞTİRİLMESİ

Yapay zeka kullanılarak yapılan binlerce akademik çalışma bulunmaktadır. Ancak bunların çok büyük çoğunluğu insanların kullanımına geçmeden teorik olarak kalmaktadır. Yapılan çalışmaların program olarak kullanıcılara sunulması farklı bir süreçtir. Akademik çalışmaya göre farklı yetenekler ve zaman gerektirmektedir. Bu çalışmada yapılan akademik çalışmamızdan laboratuvarlarda kullanılabilecek bir yazılım geliştirdik. Görüntülerin doğru bir şekilde analiz edilmesi için laborantların işlerini kolaylaştıracak bir araç ortaya koyma hedeflenmiştir. Geliştirdiğimiz program uzmanların bu görüntüleri etkili bir şekilde analiz etmelerine, damarların dallanma noktalarını, uzunluklarını, yoğunluklarını ve uç noktalarını belirlemelerine yardımcı olmaktadır. Programımız hastalıkların erken teşhisi ve tedavisinde kritik bir araç olarak hizmet etmesini akademik, bilimsel çalışmalara yardımcı olacaktır. Ayrıca, yazılımımız araştırmacılara da çeşitli bilimsel çalışmalarında değerli veriler sunacaktır.

4.1. BrainVasculyzer: 2B Beyin Damar Görüntü Analiz Programı

4.1.1. Çalışmanın Program Haline Getirilme Süreci

Programımıza "**Brain Vasculyzer**" (BV) ismini verdik. Programının geliştirilmesi, beyin damar görüntülerinin otomatik geleneksel ya da derin öğrenme tekniğiyle analiz edilmesi ve raporlanması amacıyla yapılan bir çalışmanın sonucudur. Bu sürecin başlıca adımları şunlardır:

Gereksinimlerin Belirlenmesi: İlk olarak, beyin damar görüntülerinin analizi için ihtiyaç duyulan temel fonksiyonlar belirlenmiştir. Mevcut programların yetersiz olduğu noktalar tespit edilerek, bu eksiklikleri giderecek özellikler tanımlandı. Kullanım kolaylığı, doğruluk ve hız öncelikli gereksinimler arasında yer almıştır.

Teknoloji Seçimi: Programın geliştirilmesinde Python programlama dili tercih edildi. Bu tercih, Python'un zengin kütüphane desteği ve geniş topluluğu sayesinde esnek bir geliştirme ortamı sunmasından kaynaklanmaktadır. Arayüz için ise modern ve kullanıcı dostu bir deneyim sağlamak amacıyla Pyside6 kütüphanesi kullanılmıştır.

Görüntü İşleme Teknikleri: Programın çekirdek işlevselliği, beyin damarlarının görüntü işleme teknikleri ile analiz edilmesine dayanmaktadır. İlk olarak, mikroskop ile elde edilen görüntüler üzerindeki gürültüler BM3D algoritması ile temizlenmiştir. Gürültü temizliği sonrasında, görüntüler “triangle thresholding” yöntemi ile bölütlenmiştir. Ardından, 2B inceltme algoritması kullanılarak damar iskeletleri çıkarılmıştır. Bu aşamadan sonra seçimli olarak geleneksel ya da derin öğrenme yöntemiyle dallanma ve uç noktalar bulundu. Ardından damar uzunlukları hesaplanmıştır.

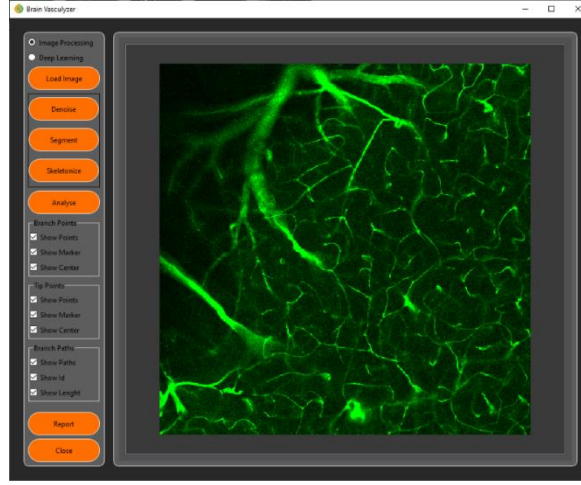
Görselleştirme ve Arayüz Geliştirme: Programın kullanıcı arayüzü, işlemlerin kolaylıkla yönetilebilmesi için sade ve modern bir tasarımla geliştirildi. Kullanıcılar, görüntü yükleme, gürültü temizleme, bölütleme, iskelet çıkarma ve analiz işlemlerini adım adım gerçekleştirebilir. Ayrıca, analiz edilen sonuçlar grafiksel olarak kullanıcıya sunulmuştur ve detaylı raporlar oluşturulabilir.

Test ve Karşılaştırma: Geliştirilen program, literatürde yer alan diğer referans programlarla karşılaştırıldı. Özellikle Fiji programı ile yapılan karşılaştırmalarda, BrainVasculyzer'ın doğruluğu ve etkinliği test edildi. Karşılaştırmalar sonucunda, programın farklı algoritmalar kullanarak daha doğru sonuçlar verdiği tespit edilmiştir.

Raporlama: Programın son adımı, analiz edilen verilerin raporlanmasıdır. Kullanıcılar, analiz sonuçlarını hem ekranda görüntüleyebilir hem de bu sonuçları CSV ve PDF formatlarında kaydedebilir. Bu raporlar, damar haritalarını inceleyen araştırmacılar için gerekli olan tüm bilgileri içerecek şekilde tasarlandı.

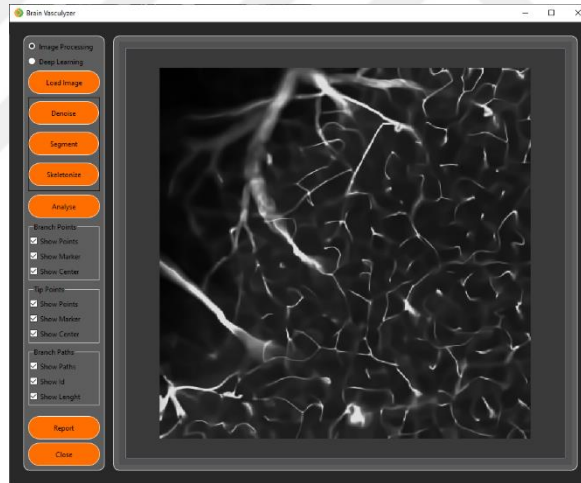
4.1.2. Programın Çalışma Ekranları

Program basit bir arayüz ile sunulmuştur. En tepede görüntü işleme mi yoksa derin öğrenme yöntemiyle mi dallanma ve uç noktaların tespiti yapılacağını seçimi yapılmaktadır.



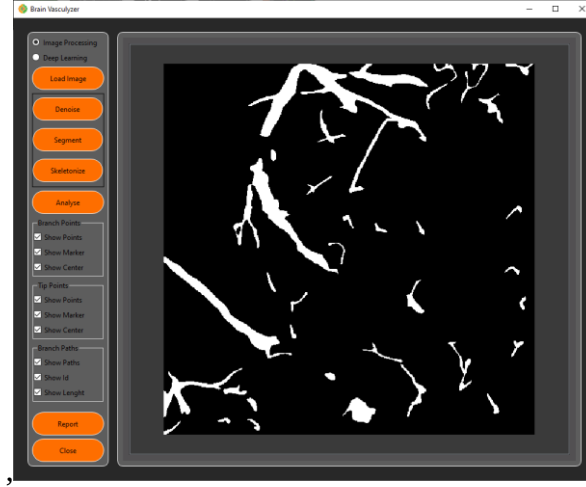
Şekil 4.1: Görüntü Yükleme

Ardından bir görüntü yükleme butonuna basıldığında dosya ekranı açılmaktadır, kullanıcı bir görüntü seçerek bir damar görüntüsü yüklenebilmektedir (Şekil 4.1).



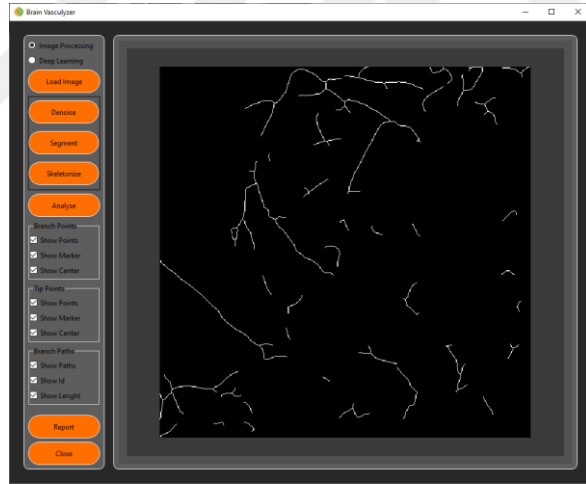
Şekil 4.2: Gürültü Temizleme

Gürültü temizleme (Denoise) butonunda daha önce bahsettiğimiz BM3D algoritması ile görüntü gürültülerinden temizlenmektedir.



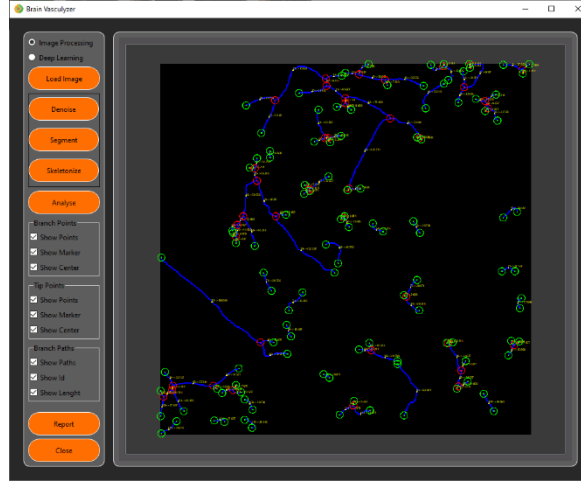
Şekil 4.3: Bölütleme

Bölütleme (Segment) işleminde görüntü “Triangle” eşik değeri ile eşiklenmektedir. Böylece damar izleri diğer izlerden arındırılmaktadır.



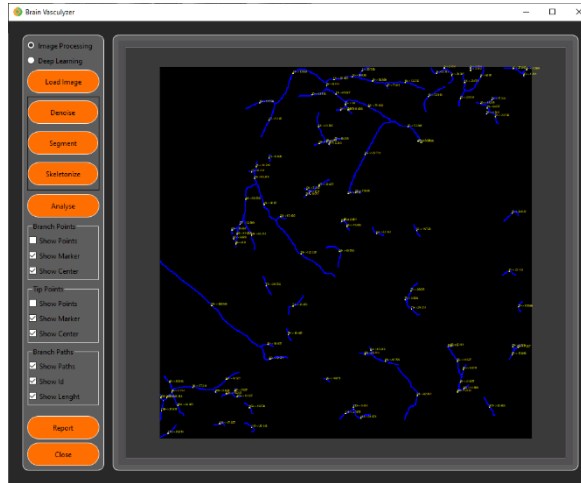
Şekil 4.4: İskelet Çıkarımı

İskelet Çıkarımı işleminde 2b inceltme algoritması kullanılmıştır. Segmente edilmiş görüntüyü incelterek damar haritasının çıkarılmasını sağlamaktadır.



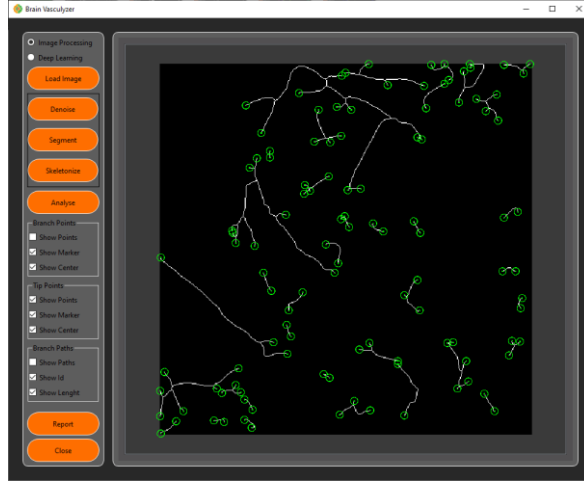
Şekil 4.5: Analiz İşeminin Başlatılması

Analiz işlemi iki türlü olabilmektedir: geleneksel görüntü işleme ya da derin öğrenme metodu. En tepede bulunan seçime göre görüntünün dallanma ve uç noktaları çıkarılmaktadır. Ardından dallanma ve uç noktaları kullanılarak tüm damar yolları piksel piksel bulunmaktadır. Tüm bu bileşenler geliştirdiğimiz algoritmalar yardımıyla veri yapısı haline getirilmektedir. Böylece görselleştirme için alt yapı hazır hale getirilmektedir.



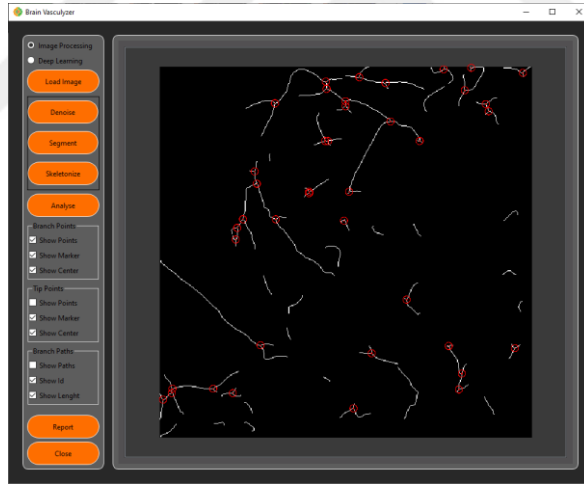
Şekil 4.6: Damar Yollarının Gösterilmesi

Damar yolları her bir damar koluna bir “id” verilerek işaretlenmiştir ve “mavi” ile gösterilmiştir. Kullanıcı istediği urumda bu damar yollarını seçebilir, seçilen yol “pembe” görünür. Damar yolu “id” değeri ve uzunluğu ekranda damarın ortasında görünür. **Şekil 4.6**’da bir görüntünün damar yolları görülebilir.



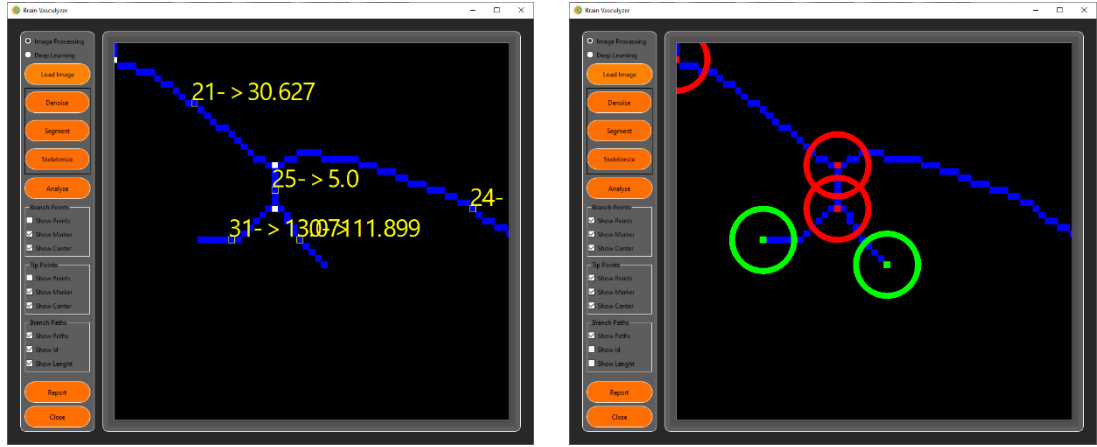
Şekil 4.7: Uç noktaların Gösterimi

Damar ağının uç noktaları “yeşil” renk ile işaretlenmiştir. Tek bir piksel ortada olacak şekilde etrafına halk şeklinde bir işaretçi çizilmiştir. Kullanıcı istediği durumda uç noktaları ve işaretçilerin görünürliğini değiştirebilmektedir.



Şekil 4.8: Dallanma Noktalarının Gösterimi

Damar ağının dallanma noktaları “kırmızı” renk ile işaretlenmiştir. Uç noktalara benzer şekilde tek bir piksel ortada olarak etrafına halk şeklinde bir işaretçi çizilmiştir. Kullanıcı istediği durumda uç noktaları ve işaretçilerin görünürliğini değiştirebilmektedir.

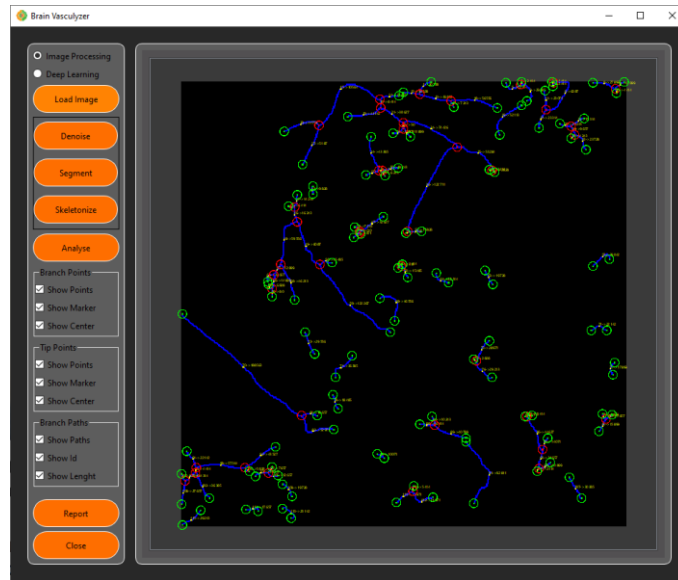


(a)

(b)

Şekil 4.9: Analiz Verilerinin Gösterimi. (a): Damar Id ve Uzunluk Gösterimi, (b): Dallanma (Kırmızı) ve Uç (Yeşil) Noktaları Gösterimi

Şekil 4.9’de program içerisinde yakınlaştırılmış bir görüntü verilmiştir. Görüntüde damar id değerleri ve uzunlukları, dallanma ve uç noktalar işaretlenmiş şekilde gösterilmiştir. Program bu açıdan oldukça anlaşılır bir kullanıma sahiptir. Şekil 4.10’da tüm analiz işaretlemelerini görülebilmektedir.



Şekil 4.10: Programın Tüm Görüntüsü

Programa son olarak raporlama eklenmiştir. Raporlama kısmında kullanıcıya raporda istediği bilgileri seçebileceği bir ekran verilmiştir. Kullanıcı seçimine göre rapor “pdf” ve “csv” formatında kaydedilmekte ve ekranda da görülebilmektedir.



Şekil 4.11: Raporlama Seçenekleri

The screenshot shows the 'Brain Vasculyzer' software interface with the 'VESSEL ANALYSE' report displayed. The report is a table with columns for 'vessl(%)', 'branch points count', 'tip point count', 'vein count', 'total vein length', and 'average vein length'. The report also includes a table with columns for 'id', 'p1.x, p1.y', 'p2.x, p2.y', 'length', 'p1_type', and 'p2_type'.

vessl(%)	branch points count	tip point count	vein count	total vein length	average vein length
7.456	40.0	112.0	115.0	2952.321	25.451

id	p1.x, p1.y	p2.x, p2.y	length	p1_type	p2_type
1	287,0	274,14	20.728	TIP	BRANCH
2	391,0	390,3	2.414	TIP	BRANCH
3	426,0	428,1	1.414	TIP	BRANCH
4	509,0	499,8	12.899	TIP	BRANCH
5	373,1	390,3	22.314	TIP	BRANCH
6	428,1	419,32	62.87	BRANCH	BRANCH
7	427,5	428,1	3.414	TIP	BRANCH
8	473,1	499,8	27.899	TIP	BRANCH
9	397,22	390,3	29.385	TIP	BRANCH
10	500,13	499,8	4.414	TIP	BRANCH
11	418,10	419,32	23.071	TIP	BRANCH
12	255,12	274,14	18.828	TIP	BRANCH
13	274,14	310,22	36.899	BRANCH	BRANCH

Şekil 4.12: Analiz Raporu

Şekil 4.11, Şekil 4.12’de görüldüğü gibi tez kapsamında geliştirilmiş olan program bir analiz programından beklenen temel işlemleri yerine getirebilmektedir.

4.2. Geliştirilen Programın Test Edilmesi

4.2.1. Referans Program ile Karşılaştırma

Çalışmanın doğruluğunu değerlendirmek amacıyla Fiji programı ile karşılaştırmalar yapılmıştır. Fiji programında görüntü gürültüsünü temizlemeye yönelik çeşitli eklentiler bulunmasına rağmen, bu eklentilerle başarılı sonuçlar elde edilememiştir. Bu nedenle, geliştirilen görüntü işleme programı Fiji programına yüklenmiş ve sırasıyla iskelet oluşturma ile iskelet analizi eklentileri kullanılarak test edilmiştir. Fiji programı, analiz sonuçlarını Excel formatında vermekte olup, gerekli hesaplamalar yapılarak karşılaştırmalar yapılabilmektedir. Sonuçlar, aşağıda yer alan **Tablo 4.1**'de gösterilmiştir.

Tablo 4.1: Fiji ile BV Karşılaştırması

Image	Program	VAF(%)	Branch points count	Tip point count	Branch count	Total branch length	Average branch length
z09	BV	12.765	95	239	255	5404.697	21.112
	FIJI	12.765	69	214	213	5364.546	25.185
z11	BV	13.593	82	240	236	5618.905	23.708
	FIJI	13.593	60	219	200	5537.398	27.444
z18	BV	7.539	72	264	238	4474.877	18.723
	FIJI	7.539	49	242	195	4366.568	22.392
z21	BV	7.382	73	224	218	4407.102	20.123
	FIJI	7.382	48	195	169	4325.965	25.597
z26	BV	8.424	68	212	207	4491.526	21.593
	FIJI	8.424	56	201	185	4487.219	24.255
z30	BV	6.322	52	180	167	3433.836	20.439
	FIJI	6.322	36	164	138	3366.819	24.397

Karşılaştırma sonuçlarına göre belirgin farklılıklar tespit edilmiştir. Yapılan analizler, bu farklılıkların temel nedeninin kullanılan algoritmaların farklılığı olduğunu ortaya koymaktadır. Fiji programı, tespit ettiği dalları analiz ederek gereksiz gördüğü kısımları budamaktadır. Ayrıca, bazı çatallanma noktalarını birleştirmektedir. Bu sebeple, dal sayıları, dallanma ve uç noktalarının sayısı daha az çıkmaktadır. Bununla birlikte, iskelet çıkarıldıktan sonra BV programında herhangi bir işlem yapılmadığı durumda, BV programının tüm bilgileri doğru bir şekilde tespit ettiği gözlemlenmiştir.

4.2.2. Bulgular ve Tartışma

Tez çalışmasında, mikroskopla elde edilen beyin görüntülerinin işlenerek analiz edilebileceği, görsel açıdan zengin ve kullanıcı dostu bir program sunmak hedeflenmiştir. Mevcut referans program ile yapılan karşılaştırmalar sonucunda bazı farklılıklar tespit edilmiştir. Ancak bu farklılıkların iskelet çıkarımı ve damar ağının değerlendirilmesi aşamalarında ortaya çıktığı belirlenmiştir. Gelecekte, algoritmalar farklı açılardan ele alınabilir geliştirmeler yapılabilir.

Bu süreçte elde edilen başarılar, beyin damar hastalıklarının teşhisinde ve bilimsel araştırmalarda kullanılabilecek etkin bir araç ortaya koymuş olduk. Geliştirilen yazılım, mevcut referans programlarla yapılan karşılaştırmalarda üstün performans sergilemiş ve kullanım kolaylığı ile dikkat çekmiştir. Bu tür programların geliştirilmesi, beyin damarlarının daha iyi anlaşılmasına ve nörolojik hastalıkların tedavisinde yeni yöntemlerin geliştirilmesine olanak sağlayacaktır.

SONUÇ

Bu tez çalışmasında, beyin damarlarının iki boyutlu mikroskop görüntülerinde dallanma ve uç noktalarının tespit edilmesi amacıyla derin öğrenme tabanlı nesne tespiti algoritmaları kullanılarak, nörovasküler analiz için yenilikçi bir yaklaşım sunulmuştur. Çalışmanın temel amacı, mevcut geleneksel yöntemlerin yetersizliklerini aşarak, derin öğrenme modellerinin sağladığı yüksek doğruluk oranlarıyla beyin damarlarının otomatik analizini gerçekleştirmek olmuştur. Bu doğrultuda, fare beyni dilimlerinden elde edilen ham mikroskop görüntüleri üzerinde çeşitli görüntü işleme teknikleri uygulanmış, özellikle BM3D algoritması ile gürültü temizleme, üçgen eşikleme yöntemi ile segmentasyon ve 2B inceltme algoritması ile iskelet çıkarma adımları başarılı bir şekilde gerçekleştirilmiştir. Ardından geleneksel görüntü işleme yöntemleri ile dallanma ve uç noktaları bulunarak referans doğru veri seti oluşturulmuştur.

Çalışma kapsamında kullanılan dört farklı derin öğrenme modeli, Detectron2 çerçevesinde eğitilmiş ve bu modellerin performansları “Intersection over Union” (IoU) metriği ile değerlendirilmiştir. Model sonuçları incelendiğinde, bazı modellerin %90'ın üzerinde bir başarı oranı elde ettiği görülmüştür. Bununla birlikte, Optuna yazılımı ile yapılan hiperparametre optimizasyonu sayesinde, model doğruluğunun %98'in üzerine çıkarılması mümkün olmuştur. Hiperparametre optimizasyonu, derin öğrenme modellerinin performansını artırmak için kritik bir adım olarak öne çıkmış ve bu sürecin modeli daha etkin hale getirdiği görülmüştür. Bu süreç, derin öğrenme modellerinin gerçek dünya verileri üzerinde daha başarılı sonuçlar vermesini sağlamış, beyin damarlarının tespiti ve analizinde önemli ilerlemeler kaydedilmiştir.

Bu çalışmada geliştirilen Python tabanlı BrainVasculyzer yazılımı, beyin damarlarının otomatik analizini gerçekleştiren kullanımı kolay bir araç olarak tasarlanmıştır. Yazılım, mikroskop görüntülerinden elde edilen verilerin hızlı ve doğru bir şekilde işlenmesini sağlamış ve mevcut analiz programlarıyla karşılaştırıldığında yüksek performans göstermiştir. BrainVasculyzer, beyin

damarlarının uzunluđu, dallanma ve uç noktaları gibi bilgileri otomatik olarak çıkararak görsel bir şekilde sunabilmektedir. Bu yazılım, nörogörüntüleme alanında araştırmacılara ve klinisyenlere önemli bir araç sunmakta, beyin damarlarının yapısal analizini daha erişilebilir ve hassas hale getirmektedir.

Bu tez çalışmasının literatüre katkısı, derin öğrenme tabanlı nesne tespiti yöntemlerinin nörovasküler analizdeki uygulanabilirliğini göstermesi ve bu alanda mevcut yöntemlerin ötesine geçerek daha hassas ve doğru sonuçlar elde edilmesini sağlamasıdır. Derin öğrenme yöntemlerinin, geleneksel görüntü işleme tekniğıyle yapılabilen çalışmanın yüksek doğruluk ve verimlilik ile başarılabildiğı bu çalışmada kanıtlanmıştır. Özellikle nörovasküler bozuklukların teşhis ve tedavisinde kullanılabilecek yeni yöntemlerin geliştirilmesi, bu tez çalışmasının önemli bir çıktısı olarak değerlendirilmektedir. BrainVasculyzer yazılımı, klinik uygulamalarda hastalık ilerlemesinin değerlendirilmesi ve tedavi etkinliğinin izlenmesi açısından değerli bir araç olarak kullanılabilir.

Bu çalışma, beyin damarlarının otomatik analizi ve nörovasküler bozuklukların teşhisi için yeni bir yaklaşım sunmaktadır. Derin öğrenme ve hiperparametre optimizasyonu, nörogörüntüleme alanında önemli ilerlemeler sağlamak ve beyin damarlarının yapısal analizinde büyük bir potansiyele sahip olduğunu göstermektedir. Geliştirilen yöntemler ve yazılım, biyomedikal araştırmalar ve klinik uygulamalarda kullanılabilir, bu sayede nörolojik hastalıkların erken teşhis ve tedavi süreçlerine katkı sunabilir. Bu çalışma, derin öğrenmenin tıbbi görüntülemede sunduğı potansiyeli bir kez daha ortaya koymakta ve bu alandaki gelecekteki çalışmalara ışık tutmaktadır.

Son söz olarak, beyin damarlarının otomatik analizi, nörovasküler bozuklukların daha hızlı ve doğru bir şekilde teşhis edilmesine olanak tanımakta, bu da tıbbi tedavi süreçlerinde büyük bir avantaj sağlamaktadır. Bu çalışmanın bulguları, derin öğrenme yöntemlerinin tıbbi görüntüleme alanında daha yaygın bir şekilde kullanılması gerektiğini ortaya koymakta ve bu alandaki teknolojik gelişmelerin klinik uygulamalara entegrasyonu konusunda önemli bir referans niteliğı taşımaktadır.

KAYNAKÇA

- Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., ... Zheng, X. (2016). *TensorFlow: A system for large-scale machine learning*. <http://arxiv.org/abs/1605.08695>
- Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019). *Optuna: A Next-generation Hyperparameter Optimization Framework*. <http://arxiv.org/abs/1907.10902>
- Bartz, E., Bartz-Beielstein, T., Zaefferer, M., & Mersmann, O. (2022). *Hyperparameter Tuning for Machine and Deep Learning with R* (1. bs). Springer. <https://doi.org/978-981-19-5170-1>
- Baydin, A. G., Cornish, R., Rubio, D. M., Schmidt, M., & Wood, F. (2017). *Online Learning Rate Adaptation with Hypergradient Descent*. <http://arxiv.org/abs/1703.04782>
- Bergstra, J., Ca, J. B., & Ca, Y. B. (2012). Random Search for Hyper-Parameter Optimization Yoshua Bengio. İçinde *Journal of Machine Learning Research* (C. 13). <http://scikit-learn.sourceforge.net>.
- Bhuiyan, A., Nath, B., & Ramamohanarao, K. (2012). Detection and classification of bifurcation and branch points on retinal vascular network. *2012 International Conference on Digital Image Computing Techniques and Applications (DICTA)*, 1-8. <https://doi.org/10.1109/DICTA.2012.6411742>
- Bischi, B., Binder, M., Lang, M., Pielok, T., Richter, J., Coors, S., Thomas, J., Ullmann, T., Becker, M., Boulesteix, A. L., Deng, D., & Lindauer, M. (2023). Hyperparameter Optimization: Foundations, Algorithms, Best Practices and Open Challenges. İçinde *WIREs Data Mining and Knowledge Discovery* (C. 13, Sayı 2). John Wiley and Sons Inc. <https://doi.org/10.1002/widm.1484>
- Bischi, B., Binder, M., Lang, M., Pielok, T., Richter, J., Coors, S., Thomas, J., Ullmann, T., Becker, M., Boulesteix, A. L., Deng, D., & Lindauer, M. (2024). Benchmarking Automated Machine Learning (AutoML) Frameworks for Object Detection. *Information*, 15(1). <https://doi.org/10.3390/info15010063>
- Bollmann, S., Mattern, H., Bernier, M., Robinson, S. D., Park, D., Speck, O., & Polimeni, J. R. (2022). Imaging of the pial arterial vasculature of the human brain in vivo using highresolution 7T time-of-flight angiography. *eLife*, 11, 1-35. <https://doi.org/10.7554/eLife.71186>

- Bradski, G.** (2000). The OpenCV library. *Dr. Dobb's Journal*, 25(11), 120-125.
- C. Tomasi, & R. Manduchi.** (1998). Bilateral Filtering for Gray and Color Images. *Sixth International Conference on Computer Vision*, 839-846.
- Calvo, D., Ortega, M., Penedo, M. G., & Rouco, J.** (2011). Automatic detection and characterisation of retinal vessel tree bifurcations and crossovers in eye fundus images. *Computer Methods and Programs in Biomedicine*, 103(1), 28-38. <https://doi.org/10.1016/j.cmpb.2010.06.002>
- Chamarty Anusha, P. S. A.** (2018). Object Detection using Deep Learning. *International Journal of Computer Applications*, 182(32), 18-22. <https://doi.org/10.5120/ijca2018918235>
- Chen, C., Chuah, J. H., Ali, R., & Wang, Y.** (2021). Retinal vessel segmentation using deep learning: A review. İçinde *IEEE Access* (C. 9, ss. 111985-112004). Institute of Electrical and Electronics Engineers Inc. <https://doi.org/10.1109/ACCESS.2021.3102176>
- Chen, K., Wang, J., Pang, J., Cao, Y., Xiong, Y., Li, X., Sun, S., Feng, W., Liu, Z., Xu, J., Zhang, Z., Cheng, D., Zhu, C., Cheng, T., Zhao, Q., Li, B., Lu, X., Zhu, R., Wu, Y., ... Lin, D.** (2019). *MMDetection: Open MMLab Detection Toolbox and Benchmark*. <http://arxiv.org/abs/1906.07155>
- Corliss, B. A., Doty, R. W., Mathews, C., Yates, P. A., Zhang, T., & Peirce, S. M.** (2020). REAVER: A program for improved analysis of high-resolution vascular network images. *Microcirculation*, 27(5), 1-14. <https://doi.org/10.1111/micc.12618>
- Dabov, K., Foi, A., Katkovnik, V., & Egiazarian, K.** (2006). Image denoising with block-matching and 3D filtering. *Image Processing: Algorithms and Systems, Neural Networks, and Machine Learning*, 6064(February), 606414. <https://doi.org/10.1117/12.643267>
- Dave, A., Khurana, T., Tokmakov, P., Schmid, C., & Ramanan, D.** (2020). TAO: A Large-Scale Benchmark for Tracking Any Object. *Computer Vision -- ECCV 2020*, 436-454. https://doi.org/10.1007/978-3-030-58558-7_26
- Duan, H., Wang, J., Liu, X., & Liu, H.** (2008). A scheme for morphological skeleton pruning. *Proceedings of 2008 IEEE International Symposium on IT in Medicine and Education, ITME 2008*, 1112-1117. <https://doi.org/10.1109/ITME.2008.4744043>
- Eunhwa, J., & Kyungho, H.** (2006). Automatic retinal vasculature structure tracing and vascular landmark extraction from human eye image. *Proceedings - 2006 International Conference on Hybrid Information Technology, ICHIT 2006*, 2, 161-167. <https://doi.org/10.1109/ICHIT.2006.253606>

- Everingham, M., Luc, P., Gool, V., Christopher, P., Williams, K. I., Winn, J., Zisserman, A., Everingham, M., Van Gool, L., Leuven, K. U., Williams, B. C. K. I., Winn, J., & Zisserman, A. (t.y.). *The PASCAL Visual Object Classes (VOC) Challenge*. <http://www.flickr.com/>
- Fasano, A., Rebaudo, G., Durante, D., & Petrone, S. (2021). A closed-form filter for binary time series. *Statistics and Computing*, 31(4). <https://doi.org/10.1007/s11222-021-10022-w>
- Freitas-Andrade, M., Comin, C. H., da Silva, M. V., Costa, L. da F., & Lacoste, B. (2022). Unbiased analysis of mouse brain endothelial networks from two- or three-dimensional fluorescence images. *Neurophotonics*, 9(03). <https://doi.org/10.1117/1.nph.9.3.031916>
- Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2013). *Rich feature hierarchies for accurate object detection and semantic segmentation*. <http://arxiv.org/abs/1311.2524>
- Gonzalez, R. C., & Woods, R. E. (2006a). *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc.
- Gonzalez, R. C., & Woods, R. E. (2006b). *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc.
- Gu, Z., Cheng, J., Fu, H., Zhou, K., Hao, H., Zhao, Y., Zhang, T., Gao, S., & Liu, J. (2019). CE-Net: Context Encoder Network for 2D Medical Image Segmentation. *IEEE Transactions on Medical Imaging*, 38(10), 2281-2292. <https://doi.org/10.1109/TMI.2019.2903562>
- Guo, Y., & Peng, Y. (2020). BSCN: Bidirectional symmetric cascade network for retinal vessel segmentation. *BMC Medical Imaging*, 20(1). <https://doi.org/10.1186/s12880-020-0412-7>
- Hadianti, S. (2023). Optimization of The Machine Learning Approach using Optuna in Heart Disease Prediction. *Journal Medical Informatics Technology*, 59-64. <https://doi.org/10.37034/medinftech.v1i3.15>
- He, K., Gkioxari, G., Dollar, P., & Girshick, R. (2017). Mask R-CNN. *Proceedings of the IEEE International Conference on Computer Vision, 2017-October*, 2980-2988. <https://doi.org/10.1109/ICCV.2017.322>
- He, Y., Kang, S. H., & Alvarez, L. (2021). Finding the skeleton of 2d shape and contours: Implementation of hamilton-jacobi skeleton. *Image Processing On Line*, 11(February), 18-36. <https://doi.org/10.5201/ipol.2021.296>
- Hervella, A. S., Rouco, J., Jose Jorge, Novo, J., & Ortega, M. (2019). Self-Supervised Deep Learning for Retinal Vessel Segmentation Using Automatically Generated Labels from

- Multimodal Data. *International Joint Conference on Neural Networks (IJCNN)*, 1-8. <http://www.ieee.org/publications>
- Hosang, J., Benenson, R., & Schiele, B.** (2017). *Learning non-maximum suppression*. <http://arxiv.org/abs/1705.02950>
- Ilesanmi, A. E., & Ilesanmi, T. O.** (2021). Methods for image denoising using convolutional neural network: a review. *Complex and Intelligent Systems*, 7(5), 2179-2198. <https://doi.org/10.1007/s40747-021-00428-4>
- Jebur, R. S., Zabil, M. H. B. M., Hammood, D. A., & Cheng, L. K.** (2024). A comprehensive review of image denoising in deep learning. *Multimedia Tools and Applications*, 83(20), 58181-58199. <https://doi.org/10.1007/s11042-023-17468-2>
- Kaya, S., Kiraz, B., & Çamurcu, A. Y.** (2024a). Branch and End Points Detection in Cerebral Vessels Images Using Deep Learning Object Detection Techniques. *Journal of Polytechnic*.
- Kaya, S., Kiraz, B., & Çamurcu, A. Y.** (2024b). Hyperparameter Optimization in Deep Learning-Based Object Detection of Branching and Endpoints on 2D Brain Vessel Images. *Akıllı Sistemlerde Yenilikler ve Uygulamaları Konferansı (ASYU)*.
- Kuş, Z., Kiraz, B., Göksu, T. K., Aydın, M., Özkan, E., Vural, A., Kiraz, A., & Can, B.** (2023). Differential evolution-based neural architecture search for brain vessel segmentation. *Engineering Science and Technology, an International Journal*, 46. <https://doi.org/10.1016/j.jestch.2023.101502>
- Lecun, Y., Bottou, E., Bengio, Y., & Haffner, P.** (1998). *Gradient-Based Learning Applied to Document Recognition*.
- Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A., & Talwalkar, A.** (2018). Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization. *Journal of Machine Learning Research*, 1-52. <http://arxiv.org/abs/1603.06560>
- Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L.** (2014). Microsoft COCO: Common Objects in Context. *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, 8693, 740-755*. https://doi.org/https://doi.org/10.1007/978-3-319-10602-1_48
- Liu, C., Wang, K., Lu, H., Cao, Z., & Zhang, Z.** (2022). *Robust Object Detection With Inaccurate Bounding Boxes*. <http://arxiv.org/abs/2207.09697>
- Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C.** (2015). *SSD: Single Shot MultiBox Detector*. https://doi.org/10.1007/978-3-319-46448-0_2

- Livne, M., Rieger, J., Aydin, O. U., Taha, A. A., Akay, E. M., Kossen, T., Sobesky, J., Kelleher, J. D., Hildebrand, K., Frey, D., & Madai, V. I.** (2019). A U-net deep learning framework for high performance vessel segmentation in patients with cerebrovascular disease. *Frontiers in Neuroscience*, 13(FEB). <https://doi.org/10.3389/fnins.2019.00097>
- Lone, A. H., & Siddiqui, A. N.** (2018). Noise models in digital image processing. *Global Sci-Tech*, 10(2), 63. <https://doi.org/10.5958/2455-7110.2018.00010.1>
- Longair, M. H., Baker, D. A., & Armstrong, J. D.** (2011). Simple neurite tracer: Open source software for reconstruction, visualization and analysis of neuronal processes. *Bioinformatics*, 27(17), 2453-2454. <https://doi.org/10.1093/bioinformatics/btr390>
- Marr, D., & Hildreth, E.** (1980). Theory of edge detection. İçinde *Proc. R. Soc. Lond. B* (C. 207).
- McGarry, S. D., Adjekukor, C., Ahuja, S., Greysson-Wong, J., Vien, I., Rinker, K. D., & Childs, Sarah. J.** (2022). Vessel Metrics: A python based software tool for automated analysis of vascular structure in confocal imaging. *bioRxiv*, 151(0026-2862), 104610. <https://doi.org/10.1101/2022.12.22.521670>
- Murray, P., & Marshall, S.** (2013). *A Review of Recent Advances in the Hit-or-Miss Transform* (P. W. Hawkes, Ed.; C. 175, ss. 221-282). Elsevier. <https://doi.org/https://doi.org/10.1016/B978-0-12-407670-9.00005-6>
- Niemistö, A., Dunmire, V., Yli-Harja, O., Zhang, W., & Shmulevich, I.** (2005). Robust quantification of in vitro angiogenesis through image analysis. *IEEE Transactions on Medical Imaging*, 24(4), 549-553. <https://doi.org/10.1109/TMI.2004.837339>
- Özkan, E., Çetin-Taş, Y., Şekerdağ, E., Yiğit, B., Shomalizadeh, N., Sapancı, S., Ozler, C., Kesibi, J., Kızılırmak, A. B., Pekmez, M., Yapıcı-Eser, H., Zeybel, M., Karahüseyinoğlu, S., & Gürsoy-Özdemir, Y.** (2023). Hyperglycemia with or without insulin resistance triggers different structural changes in brain microcirculation and perivascular matrix. *Metabolic Brain Disease*, 38(1), 307-321. <https://doi.org/10.1007/s11011-022-01100-7>
- Rani, M., Yadav, J., Rathee, N., & Panjwani, B.** (2024). Optifusion: advancing visual intelligence in medical imaging through optimized CNN-TQWT fusion. *Visual Computer*. <https://doi.org/10.1007/s00371-024-03495-z>
- Rayed, M. E., Islam, S. M. S., Niha, S. I., Jim, J. R., Kabir, M. M., & Mridha, M. F.** (2024). Deep learning for medical image segmentation: State-of-the-art advancements and challenges. İçinde *Informatics in Medicine Unlocked* (C. 47). Elsevier Ltd. <https://doi.org/10.1016/j.imu.2024.101504>
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A.** (2016). You Only Look Once: Unified, Real-Time Object Detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 779. <https://doi.org/10.1109/CVPR.2016.91>

- Redmon, J., & Farhadi, A.** (2018). *YOLOv3: An Incremental Improvement*. <http://arxiv.org/abs/1804.02767>
- Schindelin, J., Arganda-Carreras, I., Frise, E., Kaynig, V., Longair, M., Pietzsch, T., Preibisch, S., Rueden, C., Saalfeld, S., Schmid, B., Tinevez, J. Y., White, D. J., Hartenstein, V., Eliceiri, K., Tomancak, P., & Cardona, A.** (2012). Fiji: An open-source platform for biological-image analysis. *İçinde Nature Methods* (C. 9, Sayı 7, ss. 676-682). <https://doi.org/10.1038/nmeth.2019>
- Scholl, I., Aach, T., Deserno, T. M., & Kuhlen, T.** (2011). Challenges of medical image processing. *Computer Science - Research and Development*, 26(1-2), 5-13. <https://doi.org/10.1007/s00450-010-0146-9>
- Seaman, M. E., Peirce, S. M., & Kelly, K.** (2011). Rapid analysis of vessel elements (RAVE): A tool for studying Physiologic, Pathologic and Tumor Angiogenesis. *PLoS ONE*, 6(6). <https://doi.org/10.1371/journal.pone.0020807>
- Sezgin, M., & Sankur, B.** (2004). Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging*, 13(1), 146-165. <https://doi.org/10.1117/1.1631316>
- Snoek, J., Larochelle, H., & Adams, R. P.** (2012a). *Practical Bayesian Optimization of Machine Learning Algorithms*. <http://arxiv.org/abs/1206.2944>
- Snoek, J., Larochelle, H., & Adams, R. P.** (2012b). *Practical Bayesian Optimization of Machine Learning Algorithms*. <http://arxiv.org/abs/1206.2944>
- Srinivas, P., & Katarya, R.** (2022). hyOPTXg: OPTUNA hyper-parameter optimization framework for predicting cardiovascular disease using XGBoost. *Biomedical Signal Processing and Control*, 73. <https://doi.org/10.1016/j.bspc.2021.103456>
- Strafforello, O., Kayhan, O. S., Inel, O., Schutte, K., & van Gemert, J.** (2024). *Aligning Object Detector Bounding Boxes with Human Preference*. <http://arxiv.org/abs/2408.10844>
- Todorov, M. I., Paetzold, J. C., Schoppe, O., Tetteh, G., Efremov, V., Völgyi, K., Düring, M., Dichgans, M., Piraud, M., Menze, B., & Ertürk, A.** (2019). Automated analysis of whole brain vasculature using machine learning. *bioRxiv*, 0-34. <https://doi.org/10.1101/613257>
- Todorov, M. I., Paetzold, J. C., Schoppe, O., Tetteh, G., Shit, S., Efremov, V., Todorov-Völgyi, K., Düring, M., Dichgans, M., Piraud, M., Menze, B., & Ertürk, A.** (2020). Machine learning analysis of whole mouse brain vasculature. *Nature Methods*, 17(4), 442-449. <https://doi.org/10.1038/s41592-020-0792-1>

- Uslu, F., & Bharath, A. A.** (2018). A multi-task network to detect junctions in retinal vasculature. *Lecture Notes in Computer Science, 11071 LNCS*, 92-100. https://doi.org/10.1007/978-3-030-00934-2_11
- Wu, Y., Kirillov, A., Massa, F., Lo, W.-Y., & Girshick, R.** (2019). *Detectron2*. <https://github.com/facebookresearch/detectron2>
- Xu, K., Wu, H., Iwahori, Y., Yu, X., Hu, Z., & Wang, A.** (2024). A Vascular Feature Detection and Matching Method Based on Dual-Branch Fusion and Structure Enhancement. *Sensors, 24*(6). <https://doi.org/10.3390/s24061880>
- Xu, X., Reinhardt, J. M., Hu, Q., Bakall, B., Tluczek, P. S., Bertelsen, G., & Abràmoff, M. D.** (2012). Retinal Vessel Width Measurement at Branchings Using an Improved Electric Field Theory-Based Graph Approach. *PLoS ONE, 7*(11). <https://doi.org/10.1371/journal.pone.0049668>
- Yao, X., Alam, M. N., Le, D., & Toslak, D.** (2020). Quantitative optical coherence tomography angiography: A review. İçinde *Experimental Biology and Medicine* (C. 245, Sayı 4, ss. 301-312). SAGE Publications Inc. <https://doi.org/10.1177/1535370219899893>
- Yuan, A. Y., Gao, Y., Peng, L., Zhou, L., Liu, J., Zhu, S., & Song, W.** (2020). Hybrid deep learning network for vascular segmentation in photoacoustic imaging. *Biomedical Optics Express, 11*(11), 6445. <https://doi.org/10.1364/boe.409246>
- Zhang, L. Y., Lin, P., Pan, J., Ma, Y., Wei, Z., Jiang, L., Wang, L., Song, Y., Wang, Y., Zhang, Z., Jin, K., Wang, Q., & Yang, G. Y.** (2018). CLARITY for high-resolution imaging and quantification of vasculature in the whole mouse brain. *Aging and Disease, 9*(2), 262-272. <https://doi.org/10.14336/AD.2017.0613>
- Zhao, H., Sun, Y., & Li, H.** (2020). Retinal vascular junction detection and classification via deep neural networks. *Computer Methods and Programs in Biomedicine, 183*. <https://doi.org/10.1016/j.cmpb.2019.105096>
- Zhou, D., Fang, J., Song, X., Guan, C., Yin, J., Dai, Y., & Yang, R.** (2019). *IoU Loss for 2D/3D Object Detection*. <http://arxiv.org/abs/1908.03851>
- Zudaire, E., Gambardella, L., Kurcz, C., & Vermeren, S.** (2011). A computational tool for quantitative analysis of vascular networks. *PLoS ONE, 6*(11), 1-12. <https://doi.org/10.1371/journal.pone.0027385>

