

**T.C.
ERCIYES ÜNİVERSİTESİ
FEN BİLİMLERLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**

**DERİN ÖĞRENME TABANLI GERÇEK ZAMANLI POLİP
TESPİT YÖNTEMLERİNİN GELİŞTİRİLMESİ VE
İNCELENMESİ**

**Hazırlayan
İshak PAÇAL**

**Danışman
Prof. Dr. Derviş KARABOĞA**

Doktora Tezi

**Haziran 2022
KAYSERİ**

**T.C.
ERCIYES ÜNİVERSİTESİ
FEN BİLİMLERLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**

**DERİN ÖĞRENME TABANLI GERÇEK ZAMANLI POLİP
TESPİT YÖNTEMLERİNİN GELİŞTİRİLMESİ VE
İNCELENMESİ
(Doktora Tezi)**

**Hazırlayan
İshak PAÇAL**

**Danışman
Prof. Dr. Derviş KARABOĞA**

**Haziran 2022
KAYSERİ**

BİLİMSEL ETİĞE UYGUNLUK

Bu tez çalışmadaki tüm bilgilerin, akademik ve etik kurallara uygun bir şekilde elde edildiğini beyan ederim. Aynı zamanda bu kural ve davranışların gerektirdiği gibi, bu çalışmanın özünde olmayan tüm materyal ve sonuçları tam olarak aktardığımı ve referans gösterdiğimi belirtirim.

İshak PAÇAL

YÖNERGEYE UYGUNLUK

“Derin Öğrenme Tabanlı Gerçek Zamanlı Polip Tespit Yöntemlerinin Geliştirilmesi ve İncelenmesi” adlı Doktora tezi, Erciyes Üniversitesi Lisansüstü Tez Önerisi ve Tez Yazma Yönergesi ’ne uygun olarak hazırlanmıştır.

Tezi Hazırlayan

İshak PAÇAL

Danışman

Prof. Dr. Derviş KARABOĞA

Bilgisayar Mühendisliği ABD Başkanı

Prof. Dr. Veysel ASLANTAŞ

TEŞEKKÜR

Doktora sürecimin başarılı geçmesinde bilgisini, samimiyetini ve rehberliğini asla unutamayacağım kıymetli bilim insanı olan doktora tez danışmanım Prof. Dr. Derviş Karaboğa'ya teşekkürlerimi sunuyorum. Tez izleme jürimdeki birbirinden değerli bilim insanlılarından olan Prof. Dr. Nurhan Karaboğa, Prof. Dr. Bahriye Akay ile doktora sürecimde değerli bilgilerini esirgemeyen Prof. Dr. Ahmet Karaman, Prof. Dr. Alper Baştürk ve Doç. Dr. Özkan Ufuk Nalbantoğlu hocalarıma teşekkürlerimi sunuyorum.

Bu hayatı anlamlandıran ve maddi-manevi her türlü desteği sunan, ayrıca tebessümlerini hiçbir zaman eksik etmeyen eşime, aileme ve çocuklarıma müteşekkirim.

Son olarak Erciyes Üniversitesi'nde bilgilerini esirgemeyen değerli öğretim elemanı arkadaşlarıma teşekkürlerimi iletiyorum.

İshak PAÇAL

Haziran 2022, KAYSERİ

DERİN ÖĞRENME TABANLI GERÇEK ZAMANLI POLİP TESPİT YÖNTEMLERİNİN GELİŞTİRİLMESİ VE İNCELENMESİ

İshak PAÇAL

Erciyes Üniversitesi, Fen Bilimleri Enstitüsü

Doktora Tezi, Kasım 2021

Danışman: Prof. Dr. Derviş KARABOĞA

ÖZET

Kolorektal kanseri (KRK), dünya çapında en yaygın üçüncü kanser türüdür ve görülme sıklığı sürekli artmaktadır. Kolonoskopi, KRK taramasında altın standart olarak kabul edilir ve KRK'nın öncüsü olan poliplerin kanserli yapıya dönüşmeden anında çıkarılmasına olanak tanır. Klasik bilgisayar destekli teşhis (CAD) temelli otomatik polip tespit sistemleri gerçek zamanlı çalışma, duyarlılık, kesinlik ve özgüllük açısından yoksundur. Son zamanlarda, derin öğrenme mimarilerinin birçok alanda başarılı sonuçlar vermesiyle beraber otomatik polip tespitinde bu zorluklar aşılmıştır. Bu tez, otomatik polip tespitinde yaşanan zorlukların araştırılmasını ve katkıda bulunulması için derin öğrenme yaklaşımlarını ve optimizasyon algoritmalarını kullanarak klinikte uygulanabilecek bir CAD sistemi geliştirmiştir. Önerilen sistem, halka açık veri setlerinin yetersiz olmasından dolayı literatürde var olan bütün veri setlerinden daha büyük miktarda eşsiz polip ve görüntüsü içeren yeni bir veri seti sunuyor. YOLO nesne tespit algoritmaları temelli dört model polip tespiti için önerilmiştir. Yapay arı koloni (ABC) algoritması, YOLO algoritmalarına başarılı bir şekilde entegre edilerek aktivasyon fonksiyonu ve hiper-parametre optimizasyonunu gerçekleştirerek başarıma ciddi bir katkı sağlamıştır. Bunlara ek olarak, farklı veri artırma teknikleri, eğitim stratejileri ve özellik çıkarıcı yapılar kullanılarak sistemin başarısı optimal düzeye getirilmiştir. Bu yapıların entegrasyonu sonucu, gerçek zamanlı otomatik polip tanıma sistemi klinik seviyesinde performans göstermektedir.

Anahtar Kelimeler: YOLO, ABC, Kolon kanseri, Kolorektal kanseri, Rektal kanseri, Derin öğrenme, Tıbbi görüntü işleme, Hiper-parametre optimizasyonu, Aktivasyon fonksiyonları optimizasyonu.

DEVELOPMENT AND INVESTIGATION OF DEEP LEARNING BASED REAL-TIME POLYP DETECTION METHODS

İshak PAÇAL

Erciyes University, Graduate School of Natural and Applied Sciences

PhD Thesis, May 2022

Supervisor: Prof. Dr. Derviş KARABOĞA

ABSTRACT

Colorectal cancer (CRC) is the third most common type of cancer worldwide and its incidence is constantly increasing. Colonoscopy is considered the gold standard in CRC screening and allows the timely removal of polyps, which are the precursors of CRC, before they turn into cancerous structures. Conventional computer-aided detection (CAD)-based automated polyp detection systems lack real-time operation, sensitivity, precision, and specificity. Recently, these difficulties have been overcome in automatic polyp detection, with deep learning architectures giving successful results in many areas. This thesis has developed a clinically applicable CAD system using deep learning approaches and optimization algorithms to investigate and contribute to the challenges in automatic polyp detection. The proposed system introduces a new dataset containing a larger amount of unique polyps and images than any existing dataset in the literature due to the lack of publicly available datasets. Four models based on YOLO object detection algorithms have been proposed for polyp detection. Artificial bee colony (ABC) algorithm has been successfully integrated into YOLO algorithms, making a significant contribution to the success by performing activation function and hyper-parameter optimization. In addition to these, the success of the system has been optimized by using different data augmentation techniques, training strategies and feature extractor structures. As a result of the integration of these structures, the real-time automatic polyp detection system performs at the clinical level.

Keywords: YOLO, ABC, Colon cancer, Colorectal cancer, Rectal cancer, Deep learning, Medical image processing, Hyper-parameter optimization, Activation functions optimization.

İÇİNDEKİLER

DERİN ÖĞRENME TABANLI GERÇEK ZAMANLI POLİP TESPİT YÖNTEMLERİNİN GELİŞTİRİLMESİ VE İNCELENMESİ

BİLİMSEL ETİĞE UYGUNLUK.....	ii
YÖNERGEYE UYGUNLUK.....	iii
KABUL VE ONAY.....	iv
TEŞEKKÜR.....	v
ÖZET.....	vi
ABSTRACT.....	vii
İÇİNDEKİLER.....	viii
KISALTMALAR.....	xiii
TABLolar LİSTESİ.....	xiv
ŞEKİLLER LİSTESİ.....	xv
GİRİŞ.....	1

1. BÖLÜM

GENEL BİLGİLER

1.1. Kanser Hakkında.....	5
1.1.1. Kolon Kanseri.....	6
1.1.2. Polip ve Çeşitleri.....	7
1.1.3. KRK Görüntüleme Yöntemleri.....	8
1.2. Derin Öğrenme.....	9
1.3. Problemin Durumu.....	10
1.4. Araştırmanın Amacı.....	11
1.5. Araştırmanın Önemi.....	11

2. BÖLÜM

DERİN ÖĞRENME VE POLİP TESPİTİNDE KULLANILAN GÜNCEL NESNE TESPİT ALGORİTMALARI

2.1. Giriş.....	13
2.2. Yapay Sinir Ağları	15
2.3. Derin Öğrenme Mimarileri	18
2.3.1. Evrişimsel Sinir Ağları.....	18
2.3.2. Diğer Derin Öğrenme Mimarileri	24
2.4. Polip Tespitinde Güncel Nesne Tespit Algoritmaları.....	26
2.4.1. Tek Aşamalı Yaklaşımlar	28
2.4.1.1. YOLO Algoritmaları.....	31
2.4.2. İki Aşamalı Yaklaşımlar	46
2.4.3. Diğer Yaklaşımlar	48
2.5. CNN Omurga Ağları	50
2.6. KRK Teşhisi için Derin Öğrenme Algoritmalarının Kullanımına İlişkin Literatür Çalışması.....	56
2.6.1. KRK Teşhisi için Yapılmış Çalışmalar	57
2.6.2. Gerçek Zamanlı Polip Tespiti için Yapılmış Güncel Çalışmalar	62

3. BÖLÜM

VERİ SETLERİ VE DEĞERLENDİRME METRİKLERİ

3.1. Giriş.....	65
3.2. Açık Veri Setleri.....	66
3.3. Önerilen Veri Seti	71
3.4. Değerlendirme Metrikleri	74

4. BÖLÜM

GERÇEK ZAMANLI POLİP TESPİTİ İÇİN DERİN ÖĞRENME MODELLERİ

4.1. Önerilen CAD Sistemi.....	77
4.2. Kayıp Fonksiyonları.....	81
4.3. Aktivasyon Fonksiyonları	83
4.4. Veri Artırma Teknikleri	87
4.5. Uygulama Detayları	89
4.6. Küçük Veri Setleri için Önerilen Modeller	89
4.6.1. Önerilen Modellere Ait Sonuçlar	95
4.6.1.1. Güncel Yöntemlerle Karşılaştırılması	100
4.7. Büyük Veri Setleri için Önerilen Modeller	103
4.7.1. Önerilen Modellere Ait Sonuçlar	108
4.7.1.1. SUN Veri Seti için Sonuçlar	108
4.7.1.2. PICCOLO Veri Seti için Sonuçlar	112
4.8. Özel Veri Setleri için Sonuçlar	116

5. BÖLÜM

GERÇEK ZAMANLI POLİP TESPİTİ İÇİN ÖNERİLEN DERİN ÖĞRENME MODELLERİNİN OPTİMİZASONU

5.1. Giriş.....	119
5.2. Genetik Algoritma.....	121
5.3. Yapay Arı Koloni Algoritması	122
5.4. Önerilen Hiper-parametre Optimizasyon Yöntemi	126
5.5. Model Geliştirme.....	128
5.6. Topluluk Öğrenme	130
5.7. Hiper-parametre Optimizasyonu için Sonuçlar	130
5.8. Önerilen Modellerin En Son Teknoloji Yöntemlerle Karşılaştırılması	136
5.9. ABC Algoritması ile Aktivasyon Fonksiyonlarının Optimizasyonu	138
5.9.1. Aktivasyon Fonksiyonlarının Optimizasyonu için Yöntem	141
5.9.2. Aktivasyon Fonksiyonlarının Optimizasyonu için Sonuçlar	143

6. BÖLÜM

TARTIŞMA-SONUÇ ve ÖNERİLER

6.1. Sonuçlar ve Tartışma 147

6.2. Öneriler 150

KAYNAKÇA 151

EKLER 166

ÖZGEÇMİŞ 167

KISALTMALAR

ABC	:	Artificial Bee Colony Algorithm
ADR	:	Adenoma Detection Rate
AEs	:	Autoencoders
ANN	:	Artificial Neural Network
CNN	:	Convolutional Neural Network
CAD	:	Computer-aided detection
CSP	:	Cross-Stage-Partial
CIoU	:	Complete-IoU Loss
DBN	:	Deep Belief Network
DVM	:	Destek Vektör Makinesi
DIoU	:	Distance-IoU Loss
DNN	:	Deep Neural Network
ESA	:	Evrişimsel Sinir Ağları
FC	:	Fully Connected Layer
FCN	:	Fully Convolutional Network
FP	:	False Positive
FN	:	False Negative
FPR	:	False Positive rate
FPS	:	Frame Per second
FPN	:	Feature Pyramid Network
GA	:	Genetik Algoritma
ILSVRC	:	Imagenet large-scale Visual Recognition Challenge
IoU	:	Intersection Over Union
KRK	:	Kolorektal kanser
NMS	:	Non Maximum Suppression
RNN	:	Recurrent Neural Network
R-CNN	:	Region-based Convolutional Neural Network
RBM	:	Restricted Boltzmann Machines
ROI	:	İlgi Bölgesi
RPN	:	Region Proposal Network
SPP	:	Spatial Pyramid Pooling
SGD	:	Stochastic gradient descent
TP	:	True Positive
TN	:	True Negative
YSA	:	Yapay Sinir Ağları
YOLO	:	You Only Look Once

TABLOLAR LİSTESİ

Tablo 4.1. Etis-Larib test veri setine ait sonuçlar	97
Tablo 4.2. Önerilen yöntemlerin diğer güncel yöntemlerle karşılaştırılması (Etis-Larib) .	101
Tablo 4.3. Önerilen yöntemin diğer yeni yöntemlerle karşılaştırılması (CVC-ColonDB).	103
Tablo 4.4. SUN polip veri seti için uçtan uca eğitim sonuçları	109
Tablo 4.5. SUN polip veri seti için transfer öğrenme ile eğitim sonuçları	110
Tablo 4.6. SUN polip veri seti için transfer öğrenme ve negatif eğitim sonuçları	111
Tablo 4.7. PICCOLO polip veri seti için uçtan uca eğitim sonuçları	112
Tablo 4.8. PICCOLO polip veri seti için transfer öğrenme ile eğitim sonuçları	113
Tablo 4.9. PICCOLO polip veri seti için transfer öğrenme ve negatif eğitim sonuçları	114
Tablo 4.10. Özel veri seti için eğitilmiş modellerin sonuçları	117
Tablo 5.1. Scaled-YOLOv4 nesne algılama algoritmasının hiper-parametre değerleri.....	119
Tablo 5.2. Scaled-YOLOv4'ün hiper-parametre değerleri.....	131
Tablo 5.3. SUN polip veri seti için sonuçlar	132
Tablo 5.4. PICCOLO polip veri seti için sonuçlar	133
Tablo 5.5 Önerilen yöntemlerin son teknoloji yöntemlerle karşılaştırılması	137
Tablo 5.6. Aktivasyon fonksiyonlarının SUN polip veri seti için sonuçları	139
Tablo 5.7. Aktivasyon fonksiyonlarının PICCOLO polip veri seti için sonuçları	140
Tablo 5.8. YOLOv5s modelinin aktivasyon fonksiyonu optimizasyonu ve bazı hiper- parametre değerleri	144
Tablo 5.9. SUN ve PICCOLO veri setleri için sonuçlar	145

ŞEKİLLER LİSTESİ

Şekil 1.1. Kolonoskopi cihazı ve farklı yapıda ve boyutta polip çeşitleri	9
Şekil 2.1 İnsan sinir hücresi (solda) ve YSA'na ait diagram (sağda).	16
Şekil 2.2. Tek katmanlı (solda) ve çok katmanlı (sağda) YSA'ya ait yapılar.	17
Şekil 2.3. Genel bir CNN mimarisi ve CNN'nin temel katmanları.	19
Şekil 2.4. Bir giriş görüntüsü (4×4), bir çekirdek (3×3) ve adım sayısı 1 ile evrişim	21
Şekil 2.5. Bazı doğrusal olmayan aktivasyon fonksiyonları.	22
Şekil 2.6. Bir giriş (4×4), bir filtre (2×2) ve adım sayısı 2 ile havuzlama işlemi	23
Şekil 2.7. Beş ana derin öğrenme mimarileri.	26
Şekil 2.8. Tek aşamalı yaklaşımlara ve iki aşamalı yaklaşımlara ait blok diyagramları	28
Şekil 2.9. YOLOv1 mimarisinin yapısı.	29
Şekil 2.10. SSD mimarisinin yapısı.	30
Şekil 2.11. RetinaNet mimarisinin yapısı.	31
Şekil 2.12. Polip görüntüsünün ızgaralara bölünmüş hali.	33
Şekil 2.13 Polip görüntüsü ait sınırlayıcı kutu ve sınırlayıcı kutunun oluşma süreci. ..	34
Şekil 2.14. Polip görüntüsüne ait gerçek kutu ve tahmin edilen kutu.	35
Şekil 2.15. IoU'nun nasıl hesaplandığını gösteren temsiller ve formüller.	35
Şekil 2.16. Polip tahmini için YOLO'nun ürettiği tahmin kutuları	36
Şekil 2.17. NMS öncesi ve sonrasına ait polip görüntüleri ve tahmin kutuları.	37
Şekil 2.18. YOLO algoritmasının polip görüntüsü üzerinde çalışmasını gösterimi.	37
Şekil 2.19. YOLOv2 mimarisinin yapısı.	40
Şekil 2.20. DarkNet53 mimarisinin yapısı.	42
Şekil 2.21. YOLOv4 mimarisinin yapısı.	44
Şekil 2.22. YOLOv4 (solda) ve YOLOv4-CSP (sağda) mimarilerinin yapısı.	45
Şekil 2.23. Faster R-CNN mimarisinin yapısı.	47
Şekil 2.24. Mask R-CNN mimarisinin yapısı.	48
Şekil 2.25. CenterNet mimarisinin yapısı.	50

Şekil 2.26. LeNet mimarisinin yapısı.	51
Şekil 2.27. AlexNet mimarisinin yapısı.	51
Şekil 2.28. ZFNet mimarisinin yapısı.	52
Şekil 2.29. GoogleNet mimarisinin yapısı.	52
Şekil 2.30. VGG mimarisinin yapısı.	53
Şekil 2.31. ResNet ve VGG19 mimarilerinin yapısı..	54
Şekil 2.32. Farklı katmanlara sahip ResNet mimarilerinin parametreleri	55
Şekil 2.33. DarkNet53 mimarisinin farklı bir yapısı.	56
Şekil 3.1. CVC-ClinicDB veri setine ait bazı görüntüler.	67
Şekil 3.2 CVC-ColonDB veri setine ait bazı görüntüler.	68
Şekil 3.3 Etis-Larib veri setine ait bazı görüntüler.	69
Şekil 3.4 PICCOLO veri setine ait bazı görüntüler.	70
Şekil 3.5 SUN veri setine ait bazı görüntüler.	71
Şekil 3.6. Kolonoskopi görüntülerinin işaretlenmesi/etiketlenmesi için ara yüz.	72
Şekil 3.7 Önerilen (özel veri seti) veri setine ait bazı görüntüler.	74
Şekil 4.1. Önerilen otomatik polip tespiti sisteminin akış şeması ve diyagramı.	80
Şekil 4.2. Scaled-YOLOv4 nesne tespit algoritmalarına ait bütün modellerin yapısı... ..	81
Şekil 4.3. SiLU ve ReLU'nun aktivasyon fonksiyonlarının davranışları..	86
Şekil 4.4. Çalışmada kullanılmış bazı veri artırma teknikleri.....	88
Şekil 4.5. CSPNet'in ResNet ağına uygulanmasını gösteren diyagram.....	90
Şekil 4.6. YOLOv4 ağının temel yapısı..	91
Şekil 4.7. Scaled-YOLOv4 (YOLOv4-CSP) mimarisine ait yapı.	92
Şekil 4.8. Önerilen Model 1 mimarisine ait temel yapı.	93
Şekil 4.9. Önerilen Model 2 mimarisine ait temel yapı.	94
Şekil 4.10. ETIS-LARIB veri seti için sonuçlar	99
Şekil 4.11. CVC-ColonDB veri seti için sonuçlar.....	100
Şekil 4.12. Önerilen Model 3 ağına ait temel yapı.	105
Şekil 4.13. Önerilen Model 4 ağına ait temel yapı.	106

Şekil 4.14. Negatif örneklere ait görüntüler.	107
Şekil 4.15. ETIS-LARIB veri seti için sonuçlar (Büyük Modeller)	115
Şekil 5.1. ABC algoritmasının temel akış şeması.	125
Şekil 5.2. Önerilen ABC Tabanlı YOLO algoritmasının yapısı.	126
Şekil 5.3. Derin öğrenme modelleri optimize etmek için TensorRT adımları	129
Şekil 5.4. Scaled-YOLOv4'ün modelinin SUN veri seti üzerindeki performansı	135
Şekil 5.5. Scaled-YOLOv4'ün modelinin PICCOLO veri seti üzerindeki performansı	136
Şekil 5.6. Aktivasyon fonksiyonlarının optimizasyonu için önerilen ABC tabanlı YOLO algoritmasının yapısı	142

GİRİŞ

Kolorektal kanser (KRK), dünya çapında en yaygın üçüncü kanser olmakla beraber en ölümcül kanser türlerinden biridir. KRK, bağırsakta (kolon) başlayan bir kanser türüdür. Amerikan Kanser Derneği'ne [1,2] göre, Amerika Birleşik Devletleri'nde tahmini 104.270 yeni kolon kanseri (52.590 erkek ve 51.680 kadın) ve 45.230 yeni rektal kanser vakası ile toplam 149.500 yetişkine kolorektal kanser teşhisi (2021'de 26.930 erkek ve 18.300 kadın) konulacak. Ayrıca 2021'de Amerika Birleşik Devletleri'nde KRK'ya bağlı toplam 52.980 ölümün (28.520 erkek ve 24.460 kadın) olacağı tahmin edilmektedir [1,3]. Bu istatistiksel veriler, KRK'nın ne kadar yaygın ve ölümcül olduğunu vurgulamaktadır. KRK, kolon ve rektum astarındaki sağlıklı hücrelerin zamanla farklı yapılar veya kitleler oluşturmasıyla başlar [4,5]. Kitleler, yani tümörler kötü huylu olduklarında hızla büyüebilir ve vücudun diğer bölgelerine yayılarak yaşamı tehdit eden durumlara sebep olabilirler [4,6]. Polipler çoğunlukla kolon veya rektumun iç duvarında zamanla gelişebilen zararsız yapılardır [7]. Ancak bu polipler erken dönemde alınmazsa hayatı tehdit eden kanserli bir forma dönüşebilirler. Polipleri kanserli dokuya dönüşmeden önce çıkarmak KRK'yı önlemenin etkili bir yoludur. Altın standart olarak kabul edilen kolonoskopi taraması ile polipler tespit edilebilir. Kolonoskopi taraması yardımıyla polipler anında çıkarılabildiği için KRK'yı önlemede de oldukça etkili bir işlemdir [6–8]. Adenomlar, spesifik polip türleri de kanserli hale gelebilen yapılardır [9,10]. Düz polip tiplerini bulmak zordur, bu nedenle genellikle kolonoskopi yardımıyla, onları vurgulamak için boya kullanılarak bu yapılar bulunabilirler. Başka bir polip türü olan hiperplastik polipler kanser öncesi kabul edilmez ve kolon ve rektumda gelişebilir[9,11,12]. Yapılan bir çalışmada [11], 136 gastroenterolog toplam 314.872 kolonoskopiyi değerlendirdi. Bu çalışmada, adenom saptama oranları %7,4 ile %52,5 arasında değişmekteydi ve sonuçlar, adenom saptama oranındaki her %1,0'lık artışın KRK riskinde %3,0'lık bir azalma ile ilişkili olduğunu ortaya koydu [11]. Bu çalışma, gözden kaçan poliplerin tespitinin hayati bir önem

taşıdığını göstermiştir. CNN mimarilerinin kullanımı, çeşitli görüntü işleme uygulamalarında hızla büyümekte ve birçok alanda kullanılmaktadır. Derin öğrenme, özellikle CNN'lerdeki gelişmeler sayesinde daha çekici hale gelmekte ve tıp gibi hayati alanlarda kullanılmasına olanak sağlamaktadır [13–16]. Örneğin, KRK analizi için CNN tabanlı polip tespit ve sınıflandırma çalışmaları 2014 yılında başlatılmıştır [13,17]. İlerleyen zamanlarda özellikle You Only Look Once (YOLO) [18] ve Region-based Convolutional Neural Network (R-CNN) [19] gibi nesne tespit algoritmalarının yüksek performansları görülmüş ve polip tespitine uygulanarak daha başarılı sonuçlar vermiştir. CNN tabanlı nesne algılama algoritmalarının sağlamlığına ve yüksek verimliliğine rağmen, bağırsak içeriğindeki dışkı ve kabarcık partikülleri polip benzeri yapılar oluşturabilir [12,20]. Ayrıca kolonoskopi cihazından çıkan ışıklar polip benzeri yapılar oluşturulabilir. Bu faktörler derin öğrenme algoritmalarının performansının düşmesine neden olabilir. Özellikle bu yapılar CNN tarafından polip olarak algılandığında yanlış pozitif ve yanlış negatif tespit sayıları artabilir. Bu sorunların üstesinden gelmek için bazı çalışmalar sunulmuştur. Örneğin, Shin ve ark. [10] yanlış pozitif algılamaları azaltmak için polip eğitim veri setine negatif örnekler, yani polipsiz görüntüler ekledi. Daha sonra modeli Faster R-CNN algoritması ile eğiterek polip tespitinde performans artışı gözlemlendiler. Ancak çalışmada performansın ne kadar arttığı belirtilmedi. CNN tabanlı polip tespit çalışmalarında negatif örneklerin eğitime dâhil edilmemesinin en önemli nedenlerinden biri negatif örnekleri içeren halka açık veri setlerinin olmamasıydı. Bununla birlikte, toplam 102.761 negatif çerçeve içeren yeni SUN polip veri seti, 2021'de tanıtıldı [11]. Böylece CNN eğitiminde negatif örneklerin polip tespiti üzerindeki etkisi daha iyi incelenebilme imkânına sahip oldu. Bu veri seti kolonoskopi cihazı kaynaklı negatif örnekleri ve polipsiz görüntüler içeren negatif örnekler içermektedir. Polip tespiti konusundaki son çalışmaların çoğu, nesne tespitine dayalı algoritmalar ve bunların değiştirilmiş sürümlerinden oluşmaktadır. Çalışmaların çoğu özel veri setlerini kullanırken, bazı çalışmalarda kamuya açık veri setleri bulunmaktadır. Özellikle halka açık veri setlerini barındıran Kolonoskopide Otomatik Polip Tespiti ile ilgili MICCAI Alt Mücadelesini izleyen birkaç çalışma vardır. Bu alt mücadeleyi takip eden çalışmaların performansı rahatlıkla karşılaştırılabilirken, spesifik veri setlerindeki çalışmaları karşılaştırmak zordur. Polip tespiti ile ilgili son zamanlarda yapılan bazı çalışmalar şunlardır. Bu çalışmada [8] gastrointestinal poliplerin sınıflandırılması için Xception, ResNet-101 ve VGG-19 mimarilerini eğitilmiştir.

Çalışmada halka açık veri seti kullanılmış ve topluluk öğrenme yöntemi kullanılarak eğitilen modellerin performansı artırılmıştır. Diğer bir çalışmada [12] geleneksel CNN modellerinden farklı olarak, yüksek oranda kaçırılan algılama ve yanlış pozitifleri (FP'ler) azaltmak için iki boyutlu (2-D) CNN tabanlı gerçek zamanlı nesne algılama ağını mekânsal-zamansal bilgilerle birleştirerek YOLOv3 tabanlı bir yaklaşım sunuldu. Bu yöntem, YOLOv3 ile eğitilmesine rağmen gerçek zamanlı polip tespiti sağlayamazken, Etis-Larib veri setinde %77'lik bir F1 performansı göstermiştir. Farklı polip türlerinin verimli bir şekilde tespit edilmesini sağlamak ve yanlış pozitiflerin sayısını azaltmak için tek seferlik bir ileri beslemeli tam evrişimli sinir ağı (F-CNN) yaklaşımı sunuldu [16]. Bu yaklaşımda ikili maskeler yerine 2D Gauss maskeleri kullanılarak polip tespit performansı artırılmış ve F1 değeri Etis-Larib açık veri seti üzerinden %86,33 değerine ulaşmıştır. Yapılan diğer bir çalışmada ise [11] %90'ın üzerinde hassasiyet ve özgüllük ile gerçek zamanlı tespit sağlayan YOLOv3'e dayalı gelişmiş bir polip tespit sistemi önerildi. Bu sistem 5 akademik merkezden toplanan 56.668 eğitim görüntüsü üzerinde eğitilmiştir. Bu çalışmada yeni SUN Polip veri seti de sunulmuş ve bu veri seti ile önerilen modelin performansı değerlendirilmiştir. Yapılan en güncel çalışmada ise [14] YOLOv4 algoritması ölçeklendirilerek gerçek zamanlı otomatik polip algılama sistemi önerilmiş. Önerilen sistemin başarısı, MICCAI Şub-Challenge on Automatic Polyp Detection in Colonoscopy veri setlerinde diğer çalışmalara kıyasla daha yüksek bir başarı ve hız sunmuştur. Bu çalışma, veri setlerinin optimize kullanımını ve hiper-parametre optimizasyonunu sağlayarak polip tespit performansını önemli ölçüde artırmıştır.

Bu tez, otomatik polip tespitinde yaşanan sorunlara bir çözüm sunmakla beraber, otomatik polip tespitinin başarımını ve performansını önemli bir şekilde artırmıştır. Polip tespitindeki başarım ve performans artışı için şu çözümler sunulmuştur. İlk olarak halka açık veri setlerindeki yetersiz eşsiz polip ve polipli görüntü sayısına çözüm olarak yeni ve büyük miktarda eşsiz polip ve görüntü içeren özgün bir veri seti sunulmuştur. Bu veri seti ile derin öğrenme algoritmaları etkili bir şekilde eğitilerek, başarımları klinik bir seviyeye ulaşabilir. Diğer çözümlerden biri ise gerçek zamanlı polip tespitinde çıkarım hızının iyileştirilmesi için YOLO algoritmalarının ölçeklendirilmesi ve çeşitli yapıların entegre edilmesidir. YOLO algoritmalarının hiper-parametre optimizasyonu daha efektif bir şekilde sağlamak için yapay arı koloni algoritmasının (ABC) YOLO'ya

entegre edilmesi ile modellerin başarımı ciddi şekilde artırılmıştır. Ayrıca, YOLO algoritmalarının kullanılacak veri setlerinin boyutlarına göre çeşitli yapılarla yeniden ölçeklendirilmesi ve optimize edilmesiyle polip tespitiindeki başarımlar ve performanslar maksimize edilmiştir.



1. BÖLÜM

GENEL BİLGİLER

1.1. Kanser Hakkında

Kanser teriminin kökeni MÖ 460-370 yıllarında, tıbbın babası olarak kabul edilen Yunan Hekim Hipokrat'a dayanmaktadır [21,22]. Hipokrat, ülser oluşturan ve ülser oluşturmeyan tümörleri tanımlamak için Yunanca 'da yengeç anlamına gelen karsinoz ve karsinom terimlerini kullanmıştır. Bu kelimenin yengeç anlamında kullanılmasının sebebi ise oluşan kanserin parmak benzeri yayılan çıkıntılar oluşturmasydı ve bu terim daha sonra Romalı Hekimler tarafından Latince kanser (cancer) kelimesine çevrilerek günümüze kadar uzanmıştır ve güncel halini almıştır [21–23]. Kanser, vücudun herhangi bir yerinde ortaya çıktığında, kontrolsüz bir şekilde bölünebilme özelliğine sahip olan ve normal vücut dokusuna zarar verme eğilimine sahip anormal hücrelerin hızlı gelişimi ile karakterize edilen çok sayıda hastalıkların herhangi birine verilen isimdir [24]. Vücudun diğer bölgelerine kısa sürede yayılan kanserler, hücrelerin ölmesine ve zamanla insan yaşamına mal olan ölümcül bir hastalıktır [25].

Kanserlerin insan vücudundaki başlama serüveni şu şekilde özetlenebilir. İnsan vücudu trilyonlarca hücrelerden oluşur ve vücudun ihtiyaç duyduğu yeni hücreler oluşturmak için büyür ve çoğalırlar. Yaşlanan hücreler ölürlar ve yerlerine yeni hücreler oluşmaktadır. Bazen, bu süreçte ciddi beklenmedik durum olan anormal veya hasarlı hücreler çoğalırlar ve böylece doku topakları, yani tümörler oluşurlar. Tümörler ise iyi ve kötü huylu tümörler olarak iki sınıfta ele alınır. İyi huylu tümörler vücuda çoğu zaman zararlı olmamakla beraber çoğalmazlar ve yakındaki dokulara yayılmaz veya onları istila etmez [25]. İyi huylu tümörler kansere neden olmazken, kötü huylu tümörler yakındaki dokulara yayılır veya onları istila eder ve vücutta uzak yerlere bile

hızla ilerleyebilme kabiliyetine sahiptirler. Kanserli tümörler bir diğer adıyla, malign, yani kötü tümörler olarak da adlandırılabilir ve hayati tehdit eden ölümcül tümörlerdir.

1.1.1 Kolon Kanseri

Kolorektal Kanserler (KRK) dünyada kadınlarda en sık görülen ikinci kanser iken erkeklerde ise en sık görülen üçüncü kanserdir. Amerika Birleşik Devletleri'nde ise KRK, hem erkeklerde hem de kadınlarda teşhis edilen en yaygın üçüncü kanser türüdür. Amerikan Kanser Derneği'ne göre 2020 yılında 104.610 yeni kolon kanseri vakası, 43.340 yeni rektum kanseri vakası ve 53.200 ölüm olacağı tahmin edilmektedir [2,3]. 2020 yılında, Türkiye de KRK erkek ve kadınlarda görülen en yaygın üçüncü kanserdir. 2020 yılında 11989 KRK erkeklerde teşhis edilirken, bu sayı kadınlarda ise 9202 civarındadır [2]. KRK, adenomatöz polipler (adenomlar) olarak bilinen glandüler dokunun iyi huylu büyümesi olarak başlar. Bu polipler zamanla malign tümörlere dönüşerek KRK'ya neden olur [26]. KRK, evresine göre çeşitli doku ve organlara metastaz yayılımı yaparak hayati fonksiyonların durmasına ve ölüme neden olur.

Kalın bağırsak ya da kolon ve rektum kanserinin diğer genel ismi KRK'dır. Bazen bu kanserler literatürde kolon ya da kolorektal olarak geçmektedir, fakat ABD gibi ülkelerde genelde kolorektal yani kolon + rektal kanserlerinin toplamını içeren daha genel bir anlamda kullanılır. Kolorektal (KRK) kanserinin üçte ikisi sol kolonda, üçte biri sağ kolonda görülmekte, %3'ünde ise aynı anda birden fazla odakta birden belirmektedir [1,27]. KRK erkeklerde bir miktar daha sık görülmekle beraber erkek/kadın oranı genelde 1,34'tür. KRK'nın erkeklerde görülme oranı %6,14 iken bu oran kadınlarda ise %5,92 olarak bildirilmesine rağmen ölüm oranları kadınlarda daha yüksektir [27]. KRK kanseri çoğunlukla poliplerden gelişmektedir. Kolon ya da rektum mukozasından kaynaklanan polipler yani iyi huylu tümörler uzun yıllar herhangi bir şikâyetle yol açmadıkları için fark edilmeleri oldukça güçtür. Adenomların sıklığı yaş ile beraber artmaktadır. Adenomlar kanser gelişme sürecini büyük oranda açıklasalar da başka dış nedenler de olmakta ve bu konuda halen tartışmalar sürmektedir. KRK kanseri, kolonda yer alan hücrelerde başlayarak hayati dokulara saldırmaya başlar. Hücre sayısı çoğaldıkça, peçete halkası gibi dairesel şekilde kolon etrafına yayılır. Erken tanı konması halinde, kanser hücreleri sadece kolon içi ile sınırlı olarak tespit edilebilir. Erken tanı konamaması halinde ise, kanser yakın organlara, lenf bezlerine ve

kan dolaşımı yoluyla karaciğer, akciğer ve diğer organlara yayılım göstererek yaşamı tehdit eden ölümcül bir hastalık olur.

1.1.2 Polip ve Çeşitleri

Genel anlamda polip, bağırsak boşluğu içine doğru kabarıklık oluşturarak büyüyen bir tümör kitlesidir. Polipler saplı ve sapsız olarak bağırsağa tutunabilir. Bir sapla bağırsak duvarına tutunabilen polip çeşitlerine saplı polip denilirken, sap içermeyip geniş bir taban ile bağırsak duvarına tutunanlara ise sessil polipler denilir. Kansereleşme potansiyeline göre ise polipler, kansereleşme potansiyeli olanlar (neoplastik) ve olmayanlar (non-neoplastik) şeklinde sınıflandırılır [28,29]. Bir polipin bu gruplardan hangisinde yer aldığının belirlenebilmesi ancak patolojik bir inceleme ile mümkündür [29]. Bu nedenle bağırsak içerisinde görülen her polipin çıkarılması ve patolojik olarak incelenmesinin yapılması gerekir. Genelde, büyük ve sapsız poliplerde, üzerinde kanama ve düzensizlik olanlarda ve birden fazlasının bir arada bulunduğu durumlarda kansereleşme potansiyeli artmakla beraber kesin çözüm yine patolojik inceleme ile elde edilir. Neoplastik polipler, kolorektal kanser gelişiminde bir aşamayı temsil eden malign bir potansiyel barındırdıkları için birincil öneme sahiptir. Neoplastik (adenomatöz) kolorektal polipler, mukus salgılayan kolonik epitel hücrelerinden kaynaklanan iyi huylu tümörlerdir [7,28].

Adenomatöz polipler: Kalın bağırsakta en yaygın polip tipi adenom veya adenomatöz poliptir [7,28–30]. Bu polip çeşidi, sadece en yaygın olduğu için değil, aynı zamanda kolon kanserinin en sık nedeni olduğu için önemli bir polip türüdür. Bir adenomun kansere dönüşme olasılığı genellikle boyutuna bağlıdır; yani polip ne kadar büyükse, polipin habis olma olasılığı da o kadar yüksektir. Genellikle adenom poliplerde boyutlar bir santimetreden daha büyük ise bu polipin habis potansiyel olma potansiyeli artar [12]. Bununla beraber, bağırsakta tutunan polipin tek polip mi yoksa çoklu polip mi olduğu da önemlidir. Birden fazla polipi olan hastalarda, patolojik inceleme yapıldığında kötü huylu olmasalar bile, bunların ilerleyen zamanlarda kötü huylu olabilecek ek polipler geliştirme olasılığı daima daha yüksektir [29]. Kötü huylu olma potansiyeli genelde üç veya daha fazla polip aynı yerde olduğunda başlar. Son olarak, bir adenomatöz polipin kötü huylu olma eğilimi, mikroskop altında görüldüğü gibi polip hücrelerinin kendilerini organize etme şekli ile de kendini belli edebilir. Kendilerini tübüler yapılar (tübüler adenomlar) halinde organize eden hücrelerin kanserli olma olasılığı, kendilerini

parmak benzeri yapılar (villus adenomları) halinde organize eden hücrelere göre genellikle daha düşüktür.

Hiperplastik polipler: Bağırsak poliplerinde, ikinci en yaygın kolon polip tipi ve kolonda en sık görülen polip tipi hiperplastik poliplerdir [30]. Bu polip tipi, tüm yaş gruplarında görülebilen polip olmakla beraber ileri yaşlarda daha sık görülen ve genellikle en sık sol kolon, inen kolon ve rektumda görülür ve genellikle 0.5cm'den küçük boyuttadır [7,28,29]. Bu polipler, genel olarak kanserleşme potansiyeli taşımadığı kabul edilmekle beraber ailede kolon kanseri olan kişilerde önemsenmesi gerekir ve çok nadir olarak hiperplastik polip zemininde kanserleşme görülebildiği için dikkate alınması gerekir.

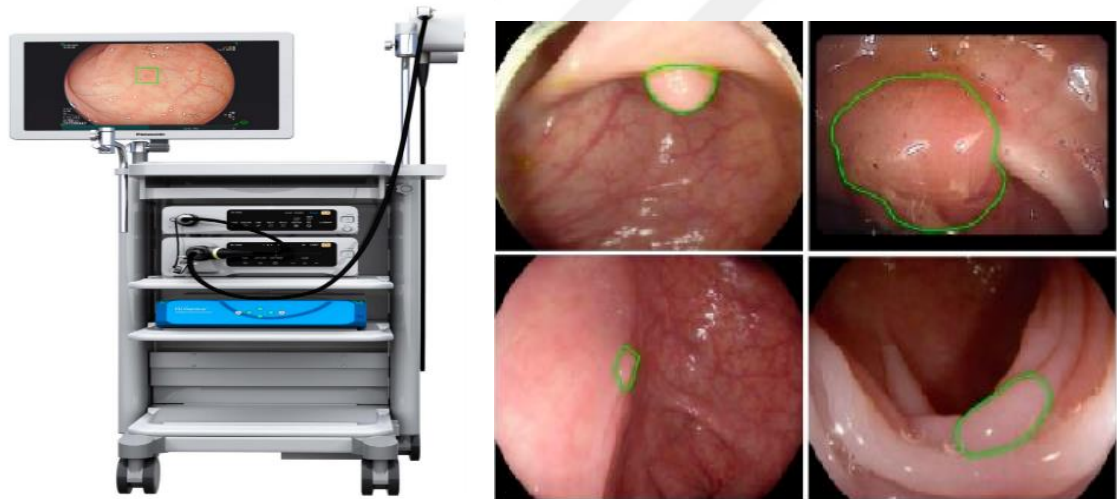
Diğer polip çeşitleri: Diğer polip tipleri ise hamartomatöz, juvenil ve inflamatuvar polipler diye ayrılabilir [28–30]. Bu polip tipleri, çok daha az yaygın olan kolon polip tipleridir ve kanserli olma eğilimleri büyük ölçüde değişir ve genellikle diğer polip tiplerine göre oldukça azdır.

1.1.3 KRK Görüntüleme Yöntemleri

KRK erken evrede teşhis edildiğinde büyük ölçüde tedavi edilebilir bir hastalık olduğundan görüntüleme yöntemlerinin etkinliği hayati bir önem taşımaktadır. KRK'de erken teşhis, mortalite, morbidite azalmasının yanında; tedavi maliyetlerini de önemli ölçüde düşürecektir. Kolorektal kanseri erken evrede teşhis etmenin en etkili yolu ise hastalığı asemptomatik evrede tarama programları ile yakalamaktır. Tarama programlarında gaitada gizli kan testi, sigmoidoskopi, kolonoskopi ve diğer görüntüleme yöntemleri kullanılmaktadır. Bununla beraber, KRK tarama önerisinde sırasıyla dikkat edilmesi gerekenlerin başında; etkinlik, duyarlılık, yalancı pozitiflik, güvenilirlik, maliyet, etkinlik ve hastanın tercihi göz önünde bulundurulması gereken ana etkenlerdir [31]. Tüm bunlar göz önüne alındığında, kolonoskopi KRK erken teşhisinde altın standart olarak herkesçe kabul görmektedir [6,32].

Kolonoskopi, kalın bağırsak (kolon) ve rektumdaki değişiklikleri veya anormallikleri tespit etmek için ve genelde KRK taramasında kullanılan bir tıbbi cihazdır. Kolonoskopi sırasında rektuma uzun, esnek bir tüp (kolonoskop) yerleştirilir. Tüpün ucundaki küçük bir video kamera ve ışığın olduğu, bükülebilen, yaklaşık bir insan

parmağı kalınlığında olan, böylece doktorun tüm kolonun içini görmesini sağlayarak kanser öncüsü olan poliplerin ya da anormalliklerin tespit edilmesini sağlar [33]. Kolonoskopi sayesinde dokulardan anında biyopsi alınabilir ve polipler de aynı anda alınabilir. Bu sayede, polipli dokulara anında müdahale edilebilir ve bu da onu diğer yöntemlerden farklı kılmaktadır [34]. Kolonoskopi taraması hastanın bir-iki gün öncesinde, hastanın sulu, yumuşak bir diyeti ile beslenmesi gerekiyor ve aynı zamanda bağırsak temizliği yapacak ilaçlarını da hekiminin belirttiği gibi işlem öncesinde alması gerekiyor. Aksi halde, istenilen net görüntü elde edilmeyebilir. Ayrıca, kolonoskopi riski ise diğer görüntüleme yöntemlerine göre oldukça az olmakla beraber, hasta daha rahat ve kesin bir kontrolden geçmektedir [6,32]. Kolonoskopi cihazının ve bazı polipli görüntülere ait görüntüler Şekil 1.1’de gösterilmiştir.



Şekil 1.1. Kolonoskopi cihazı (solda) ve farklı yapıda ve boyutta polip çeşitleri (sağda).

1.2 Derin Öğrenme

Derin öğrenme, bünyesinde bulundurduğu hiyerarşik mimariler sayesinde modele daha fazla “derinlik” (karmaşıklık) ekleyerek üst düzey soyutlamaları öğrenmeye çalışan bir makine öğrenimi alt alanıdır [35]. Derin öğrenmenin en güçlü avantajlarından birisi özellik öğrenmesini hiyerarşinin daha üst seviyelerindeki özelliklerin daha alt seviyeli özelliklerin birleşimi ile oluşturulduğu ham verilerden otomatik bir şekilde yeni işlevsel özellikler çıkarmasıdır [13,17]. Bu avantajlara ek olarak derin öğrenme, paralel programlamaya izin vererek daha karmaşık modellerin eğitime olanak sağlayarak

daha karmaşık sorunları iyi ve hızlı bir şekilde çözebilme kabiliyetine sahiptir. Bu karmaşık modeller çok daha büyük veri setleri ile eğitildiğinde daha büyük başarımlar sağlayarak regresyon problemlerindeki hatayı daha da azaltabilirler [36]. Derin öğrenme, son zamanlarda sürekli bir gelişme trendi ile özellikle görüntü işleme, doğal dil işleme, medikal görüntü işleme, otonom araçlar ve savunma sanayi gibi birçok alanda en üst sıralarda kendine yer edinmiştir [37]. Günümüzde derin öğrenmenin bu denli dikkatleri üzerine çekmesindeki en büyük etkenler ise hiç şüphesiz ki, önemli ölçüde artan yonga işleme yetenekleri; özellikle grafik işleme kartları, önemli ölçüde düşük bilgi işlem donanımı maliyeti ve makine öğrenme algoritmalarındaki önemli gelişmelerdir.

Derin öğrenme algoritmaları kanserlerin teşhisinde son zamanlarda geniş bir alanda oldukça yaygın bir şekilde kullanılmaktadırlar [38,39]. Kanser teşhisi için derin öğrenmeyle ilgili yayınların sayısı her geçen gün hızla artıyor. KRK teşhisinde ise derin öğrenme algoritmalarının elde ettikleri başarılar gastroenterologlarla karşılaştırılabilir bir seviyeye gelmiş ve hatta onlardan daha iyi performans gösterdiği çalışmalarda dile getiriliyor [13,15,40,41]. Bununla birlikte, derin öğrenme uygulamalarının tıbbi alanlarda faydaları kanıtlanmıştır [42,43]. Özellikle hastanelerde, üniversitelerde her gün milyonlarca görüntünün saklanabildiği ortamların oluşması ve bu verilerin gelecekte derin öğrenme alanında daha aktif bir şekilde kullanılacağı düşünülmekle beraber, kanser tespitlerinde başarımlarının artacağı öngörülmektedir [44]. Özellikle son zamanlarda kanser tespitlerinde derin öğrenme algoritmalarının uzman hekimlerin seviyesine yakın ve ya daha üstünde başarımlar göstermesi, derin öğrenmenin kanser tespitlerinde ilerleyen zamanlarda daha iyi bir gelecek vaat ettiği bir gerçektir [45,46].

1.3. Problemin Durumu

KRK dünyada ve Türkiye’de en yaygın ve ölümcül kanserler arasında ilk üçte yer alması, bu ölümcül kansere karşı erken teşhise dayalı sistemlerin, özellikle bilgisayar destekli sistemlerin geliştirilmesi önem arz etmektedir. Bu bağlamda, sürekli bir araştırma konusu olan KRK kanserlerin öncüsü olan poliplerin otomatik tespiti süregelen bir problemidir. Hem poliplerin yüksek bir başarımla tespiti hem de aynı zamanda gerçek zamanlı olarak tespit edilmesi problemin önem derecesini artırmıştır. Daha önce geliştirilmiş olan bazı sistemlerin gerek hız konusunda (gerçek zamanlı

tespit) gerekse de yeteri yüksek başarıyı gösterememesi problemin güncelliğini korumasını sağlamıştır [47]. Bu bağlamda, problemin durumuna bağlı olarak hem gerçek zamanlı tespit yapabilen hem de yüksek başarıyla otomatik polip tespit sistemlerinin geliştirilmesi problemin durumunu göstermektedir.

1.4. Araştırmanın Amacı

Bu çalışmanın birincil amacı, gerçek zamanlı otomatik KRK'ler için gelişmiş bir CAD sisteminin oluşturulmasıdır. Bu CAD birimi ilk başta kolonoskopi görüntülerinden daha önce eğitilen modeller yardımı ile otomatik etiketleme yeteneğine sahip olması amaçlanmaktadır. Geliştirilen ilk model, uzman hekimler yardımı ile etiketlenmiş görüntüler ilk aşamayı oluştururken, sistem sürekli olarak kendini geliştirebilen bir birim üzerinde oluşturulmuştur. Bu birimde diğer amaç ise CAD sisteminin endoskopi cihazı ile bütünleştirilerek gerçek zamanlı çalışan bir sistemin oluşturulmasıdır. Bu bütünleşik sistem gastroenterologların sürekli kullanacağı bir yardımcı asistan görevini görecektir. Böylelikle bu sistem sayesinde, uzman gastroenterologların teşhis süresi kısalarırken, polip kaçırma oranı düşürülerek hastaların sağ kalım oranı ciddi derecede artırılmış olacaktır. Ayrıca, otomatik etiketleme ile istenilen etiketli görüntülerin elde edilmesi ile hem uzmanlara olan ihtiyaç azaltılmış olacak hem de sürekli kendini geliştirebilen bir sistemin de ana temeli atılmış olmaktadır. Tüm bu geliştirmeler ek, CAD sisteminin başarımının artırılması için hiper-parametre optimizasyonu kullanılarak daha yüksek başarı sunan bir yapı inşa edilecektir. Böylece daha esnek bir CAD sistemi geliştirilmiş olacaktır.

1.5. Araştırmanın Önemi

KRK dünyada en yaygın ve ölümcül kanserler arasında ilk üç sırada yer alır. Kolorektal poliplerin otomatik tespiti, birçok nedenden dolayı zor bir problem dâhilindedir. Çünkü her bir polip, boyut, şekil, renk ve doku açısından farklı varyasyonlara sahiptir. Ek olarak, kolonoskopi cihazına bağlı olarak gelişen ışık kaynaklı farklılıklar ve hareket halinde polip bölgesinde meydana gelen değişimler ile polip görüntülerinde ciddi farklılıklar olmaktadır. Derin öğrenme temelli, polip görüntülerinden ve videolarından oluşan etiketli büyük bir veri seti, her türlü polipi tespit edebilen verimli bir model geliştirmek için temel şarttır. Böyle bir sistemin geliştirilmesi halinde, bu zorlukların

üstesinden gelinecektir ve böylece polip tespiti oranlarında klinik seviyesinde başarımlar sağlanacaktır. Bir gastroenteroloğun adenom bulma yeteneği, Adenom Tespit Oranı (ADR) olarak bilinir. Çalışmalar, kolonoskopi taraması sırasında adenomların %30'a varan oranda gözden kaçabileceğini göstermiştir [15,48]. ADR'deki her %1'lik artış, kanser oranında %3-6'lık bir azalma ile sonuçlanır [4,14,15,48]. Ayrıca ikinci bir gözlemci yardımıyla polip tespit oranının iyileştirilebileceğini gösteren çalışmalar da mevcuttur. Böyle bir durumda, hastaların sağ kalımlarında artış olacağından, otomatik polip tespitine ihtiyaç daha fazla olmaktadır. KRK'nın hayati bir öneme sahip olması ve bu tür kanserlerin tespiti ise insan hayatı için kritik bir öneme sahip olması, problemin önem derecesini ortaya koymaktadır. Bu tür zorlu problemler için CAD sistemlerinin geliştirilmesi büyük önem arz etmektedir.

2. BÖLÜM

DERİN ÖĞRENME VE POLİP TESPİTİNDE KULLANILAN GÜNCEL NESNE TESPİT ALGORİTMALARI

2.1. Giriş

Derin öğrenme, yapay zekânın en yaygın kullanılan araçlarından biridir ve makine öğrenmesinin bir alt dalıdır [13]. Genel olarak derin öğrenme, derin bir mimaride çoklu doğrusal ve doğrusal olmayan işlem birimlerini düzenleyerek yararlı özellikleri otomatik olarak çıkarma yöntemidir. Derin Öğrenme kavramı, 2006 yılında Hinton ve arkadaşları tarafından Science dergisinde yayınlanan bir makaleden kaynak almıştır [35]. Derin öğrenme mimarilerindeki temel fikir sırasıyla şu şekilde izah edilebilir. Veri analizi ve özellik çıkarımı için yapay sinir ağlarını kullanmaktır. Sinir ağındaki döngü ise, öncelikle verilerdeki özellikler birden fazla gizli katman tarafından çıkarılır. Her gizli katman ise bir algılayıcı olarak kabul edilirken, bu algılayıcılar düşük seviyedeki özellikleri çıkarır ve daha sonra ise bunları birleştirerek daha yüksek seviyeli özellikleri çıkararak yerel minimum problemini büyük ölçekte düşürür [35]. Derin öğrenme, geleneksel algoritmalarda yapay olarak tasarlanmış özelliklere nazaran özellik çıkarımı gibi önemli işleri otomatik bir şekilde matematiksel olarak yapması onu üst seviyelere taşıyarak daha fazla araştırmacıyı bu alana yönlendirmiştir. Çünkü şu an da derin öğrenme örüntü tanıma, otonom araçlar, konuşma tanıma, doğal dil işleme ve medikal görüntü işlemeden savunma sanayisine kadar nerdeyse özellik çıkarımı gerektiren her yerde aktif bir şekilde kullanılmaktadır.

Geleneksel makine öğrenimi yaklaşımları büyük ölçüde önceden tanımlanmış mühendislik özelliklerine dayanır ve genellikle yalnızca belirli problemler için tasarlanmıştır. Ancak, derin öğrenme algoritmaları açık özellik tanıma gerektirmez, bunun yerine verileri kullanır ve bir sonuca ulaşmak için yüksek boyutlu olarak

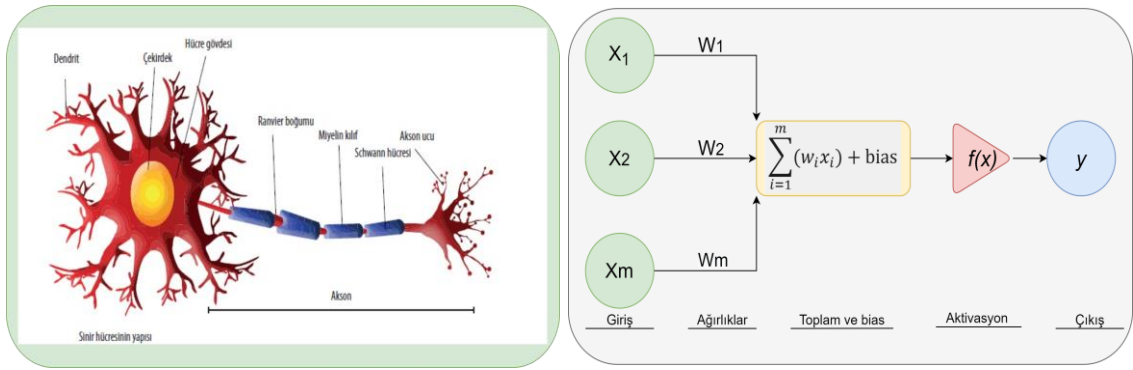
yorumlanması zor özellikleri birleştirir [49]. Derin öğrenme algoritmalarının üstün performansı ve başarısı nedeniyle klasik makine öğrenmesi algoritmaları arasında yer alan rastgele orman, destek vektör makineleri [SVM] [50] ve Gauss karışım modelleri gibi birçok algoritma yerini derin öğrenme algoritmalarına bırakmıştır. Bu geleneksel yöntemler, sağlam ve esnek değildir. Bu yüzden kanser teşhislerinde ve özellikle kolon kanseri analizinde pek kullanılmazlar çünkü manuel olarak tasarlandıkları için zaman alıcıdır. Örneğin, bu yöntemler, yüksek yanlış pozitif oranları nedeniyle kolonoskopi görüntüleri için gerçek zamanlı polip tespitinde başarılı değildir [13].

Derin ve karmaşık görüntülerin özelliklerini çıkarmak oldukça zor iken, derin öğrenme bu dar boğazı kolay bir şekilde çözer. Örneğin, düşük seviyeli özellikler, ara özellikler ve yüksek seviyeli anlamsal özellikler gibi çok seviyeli görüntü özellik bilgilerini elde etmek için orijinal görüntüden doğrudan denetimsiz öğrenme gerçekleştirerek otomatik bir şekilde yapma kabiliyetine sahiptir [51]. Kanser tespitinde daha önce probleme özgü olan ve özellik çıkarımı için uzmanlık bilgisi gerektiren geleneksel algoritmalara nispeten derin öğrenme hem denetimli öğrenme hem de denetimsiz öğrenme gerçekleştirerek gerek zamandan gerekse de uzman bilgisine fazla ihtiyaç duymadan en anlamlı özellikleri daha efektif bir şekilde çıkarabilir. Derin öğrenme, manuel güdümlenme olmadan büyük verilerden özellikleri otomatik olarak daha rahat öğrenebilir. Model, otonom öğrenme yeteneği ve özellik ifade yeteneği olan ve görüntü sınıflandırma ve tanıma için görüntü özelliklerini otomatik olarak çıkarabilen çoklu katmanlardan oluşmaktadır. Bu nedenle, derin öğrenme, kanser tespiti ve görüntü tanıma alanında büyük bir rol almaktadır. Günümüzde derin öğrenme yöntemleri temelde sırasıyla derin inanç ağları (DBNs) [52], derin Boltzmann makineleri (DBMs) [52], otomatik kodlayıcılar (AEs) [53] ve derin evrişimli sinir ağları (CNNs) [54] olmak üzere dört ana sınıfa ayrılabilir. Aslında, bazı özel derin öğrenme mimarileri elde ettikleri başarı ve kullanım yönünden farklı sınıflara ayrılabilir da temelde dört ana sınıfta belirtmek daha efektif olmaktadır. Bilindiği üzere derin öğrenme mimarileri veriye açık algoritmalarlardır. Yani, eğitim örneklerinin sayısı arttıkça ve hesaplama gücü arttıkça, derin sinir ağlarının karakterizasyon gücü daha da artmaktadır ve daha güçlü olmaktadır. Günümüzde derin öğrenmenin mimarilerin çeşitli yapılarla ortaya çıkmasıyla beraber hem endüstri hem de akademi bu alandan oldukça istifade etmektedir ve derin sinir ağı modellerinin performansı geleneksel modellerin oldukça

önünde olmasıyla beraber, daha da cazibeli hale gelmektedir. Son yıllarda, hiç şüphesiz ki en popüler derin öğrenme mimarisinin başında derin evrişimli sinir ağları gelmektedir. Bu bölümde, dört ana sınıfa ayrılan mimarilerden kısaca bahsedilecek, evrişimli sinir ağları ise detaylıca incelenecektir. Çünkü otomatik polip tespitinde kullanılan algoritmaların neredeyse hepsi CNN tabanlıdır. Bu yapılara geçmeden önce derin öğrenmenin türediği yapay sinir ağlarının temellerine bakıldıktan sonra, derin öğrenme mimarilerine geçilecektir.

2.2. Yapay Sinir Ağları

Yapay sinir ağları (YSA) [55,56], aldığı çeşitli sayıdaki parametreleri daha fazla sayıda belirsiz veri noktasıyla etkili bir şekilde ilişkilendirdikten sonra bu ilişkiyi biyolojik sinir ağına benzeten matematiksel bir modeldir. Bu şekilde makinelerin insansı bir öğrenme ve karar verme yeteneği kazandırması hedeflenmektedir. YSA modelleri, çeşitli parametreleri ilişkilendirmek için klasik yöntemlere kıyasla herhangi bir karmaşık matematiksel yapı gerektirmez [57]. Aslında YSA temelde, geleneksel yöntemlere kıyasla, herhangi bir sayıda parametreyi büyük miktarda belirsiz veri noktalarıyla ilişkilendirmeye çalıştığı zaman daha az bir maliyet yani hesaplama gerektirir [55]. YSA, insan beynindeki nöronlar gibi yapılardan oluşur yani aslında modellemeye çalışır. Bu nöronlar, ağırlık olarak adlandırılan bazı kesirli sayılarla birbirine belli bir sistemde bağlanır. Eğitime başlayan bir sinir ağı ise doğru sonuçları tahmin etmek için bu ağırlıkları ayarlar ve sonuçta belli bir oranda doğru tahmin ile gerçek doğru arasında bir hata oluşur [58]. YSA bu hatayı minimize etmeye çalışır ve böylece hata istenilen seviyeye ulaştığında ise eğitim tamamlanmış olur. YSA mimarilerin temel yapısı, giriş katmanı, gizli katman ve çıkış katmanından oluşur. YSA yapılarında en az bir gizli katman içerir. Bütün katmanların toplanması ile düğüm yani nöron oluşur. Her düğüm veya yapay nöron diğerine bağlanır ve bu nöronlar ilişkisel bir ağırlık ve eşik değeri sahiptir. Herhangi bir düğümün çıktısı belirtilen eşik değerinin üzerinde olduğu sürece, o düğüm etkinleştirilir ve böylece iletim ağı bir sonraki katmanına gönderilir [59]. Eğer düğümdeki çıktı eşik değerinin üzerinde değilse, ağı bir sonraki katmanına hiçbir iletim olmaz ve böylece veri aktarılmamış olur. İnsana ait sinir hücresi ve YSA'ya ait temel yapılar beraber Şekil 2.1'de gösterilmiştir.

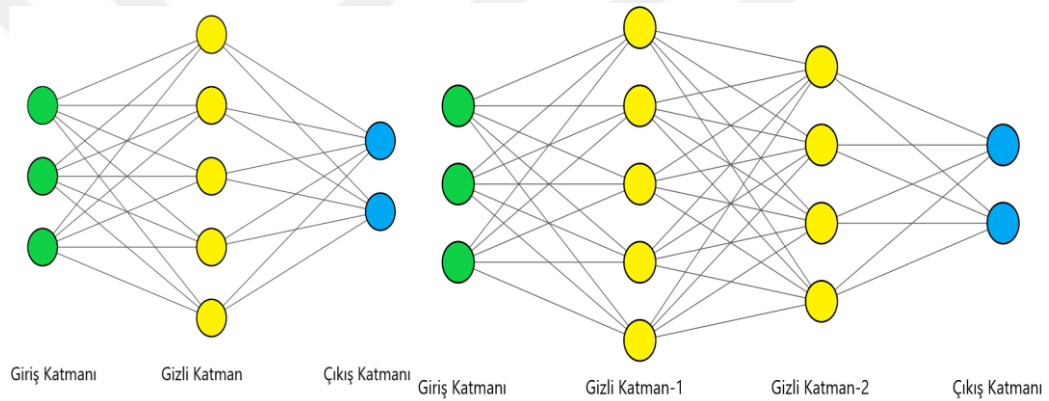


Şekil 2.1 İnsan sinir hücresi (solda) ve YSA'na ait diagram (sağda).

Şekil 2.1'de gösterildiği üzere insan nöron hücresi ile YSA'daki yapay nöron hücrelerinin benzetimi gösterilmiştir. Burada birbirine benzer görevleri üstlenen kısımlar sırasıyla şu şekildedir. İnsan sinir sistemindeki nöronun YSA'daki birimi; işlemci eleman olarak anlamlandırılırken, dendritler; toplama fonksiyonu, hücre gövdesi; transfer fonksiyonu, aksonlar; yapay nöron çıkışı ve sinapslar ise ağırlıkları temsil ederler. Böylelikle insan sinir sisteminin YSA'ya benzetimi yani modellenmesi sağlanmış olmaktadır.

YSA'daki temel amaç ise daha önce belirtildiği üzere, derin öğrenme modellerinde olduğu gibi modelin en iyi skoru vereceği ağırlık parametresi ve bias (önyargı) değerini optimize etmektir, bu şekilde hata da minimize edilmiş olmaktadır. Bu şekilde her sinir hücresi yani nöronlar hesaplanır ve böylece nöronlar arasında veri iletimi tamamlanmış olur ve bu yapı paralel ya da seri bir düzende bağlanabilir. Şekil 2.1'de (sağda) gösterildiği üzere 5 birimden oluşan yapay sinir ağlarındaki yapıların temel görevleri ise sırasıyla şu şekildedir. Giriş birimi ya da girdi olarak tanımlanan birimde nöronlara gelen veriler temsil edilir. İnsan sinir hücresinde olduğu gibi bu veriler nöron çekirdeğine iletilir. Ağırlıklar ise bir diğer temel birimdir. Burada ise girdilerden gelen veriler buradaki ağırlık birimleri ile çarpılarak çekirdeğe gönderilir, yani aslında bu ağırlıklar çıktıların üzerinde en temel role sahip birimlerin başında gelir. Bir sonraki birim olan toplama fonksiyonu ise girdi verilerinin ağırlıklar ile çarpılması sonucunda oluşan verileri toplayarak hücrenin net girdisinin hesaplanmasını sağlayan fonksiyondur. Dördüncü birim olan aktivasyon fonksiyonu ise toplama fonksiyonundan gelen verileri bünyesinde bulundurduğu çeşitli aktivasyon fonksiyonları ile beraber doğrusal olmayan bir forma sokarak, bir sonraki katmana iletmeyi sağlar. Son birim

olan çıkış birimi ise hücrenin çıktısını yani sonucunu bir sonraki hücreye iletmekle sorumludur. Buradaki yapı aslında insandaki sinir yapısının matematiksel modellenmesi ile beraber böyle bir yapıyı izlemiştir. Ama temelde olan mantık ise hatanın minimize edilerek en iyi çıktı değerinin elde edilmesini sağlamaktır. Bir YSA içerisinde bulundurduğu nöron ve katman sayısı bakımından temelde iki sınıfa ayrılabilir. Bir YSA’da birçok nöron ve gizli katmandan hiyerarşik bir yapı oluşmuş ise buna çok katmanlı sinir ağı yani multilayer artificial neural network ya da kısa adıyla MLP denir [60]. Eğer tek bir katmandan oluşuyorsa bu yapı tek katmanlı sinir ağı yani single layer artificial neural network olarak adlandırılır [60]. Tek katmanlı ve çok katmanlı yapay sinir ağlarına ait yapılar aşağıdaki Şekil 2.2’de gösterilmiştir.



Şekil 2.2. Tek katmanlı (solda) ve çok katmanlı (sağda) YSA’na ait yapılar.

YSA aynı zamanda ileri ve geri beslemeli ağlar olmak üzere iki ayrı sınıfta incelenebilir. İleri beslemeli sinir ağları, tek yönlü veri iletimine izin verirler. Giriş katmanı aldığı verileri hiçbir değişikliğe sokmadan bir sonraki katmana yani gizli katmana aktarır. Ara katmanda işlenen veri çıkış katmanına çıktı olarak iletilir. Diğer taraftan geri beslemeli sinir ağları ise en az bir sinir hücresinin çıkışı ya kendisine ya da diğer sinir hücrelerine giriş olarak verilmesi şartı aranır. Böylelikle geri beslemeli sinir ağlarında geciktirme elemanları kullanılmış olur. YSA’lar temelde böyle bir yapı içerirler. Derin öğrenmenin temelini oluşturan YSA kısaca bu şekilde açıklanabilir. Otomatik polip tespitinde derin öğrenme tabanlı gelişmiş nesne tespit algoritmaları kullanıldığı için derin öğrenme daha detaylı açıklanacaktır. Buradaki amaç YSA hakkında genel bir bilgiye sahip olunarak, derin öğrenmede otomatik polip tespiti sistemlerinin temel mantığının anlaşılır bir hale getirilmesidir.

2.3 Derin Öğrenme Mimarileri

Makine öğreniminden farklı olarak, derin öğrenme, ham verilerden temsillerin nasıl öğrenildiği konusunda farklı bir yol izler. Derin öğrenme, çok ve karmaşık katmanlardan oluşan sinir ağlarının daha fazla hesaplama ve soyutlama yaparak verilerden özelliklerin daha iyi bir şekilde temsil edilmesini yani öğrenilmesini sağlar [61,62]. Derin öğrenme ve YSA arasındaki en temel farklar; gizli katmanların sayısı, katmanlardaki bağlantıların hiyerarşisi ve girdilerin anlamlı özelliklerini öğrenme yeteneğidir. Diğer taraftan YSA ise yalnızca belirli bir görev için optimize edilmiş ve genellikle genelleştirmeden yoksun temsiller elde etme kabiliyetine sahiptir. Bununla beraber, derin öğrenmenin esas caydırıcı yönü ise özellik katmanlarının mühendisler tarafından tasarlanmasına ihtiyaç duymadan genel amaçlı bir öğrenme prosedürü kullanılarak verilerden öğrenmenin yapılmasıdır [61]. Buna binaen bu bölümde derin öğrenme mimarilerinin detaylarına inilerek daha çok belirli görevler için yetkilendirilmiş yani sınıflandırılmış derin öğrenme mimarileri tartışılacaktır. Derin öğrenmeyi zirveye götüren ve onu cazibeli hale getiren derin öğrenme mimarisinin başında hiç şüphesiz ki CNN'ler gelir. Bu yüzden CNN yapıları biraz daha detaylı anlatılırken, diğer mimariler temel olarak detaylandırılacaktır.

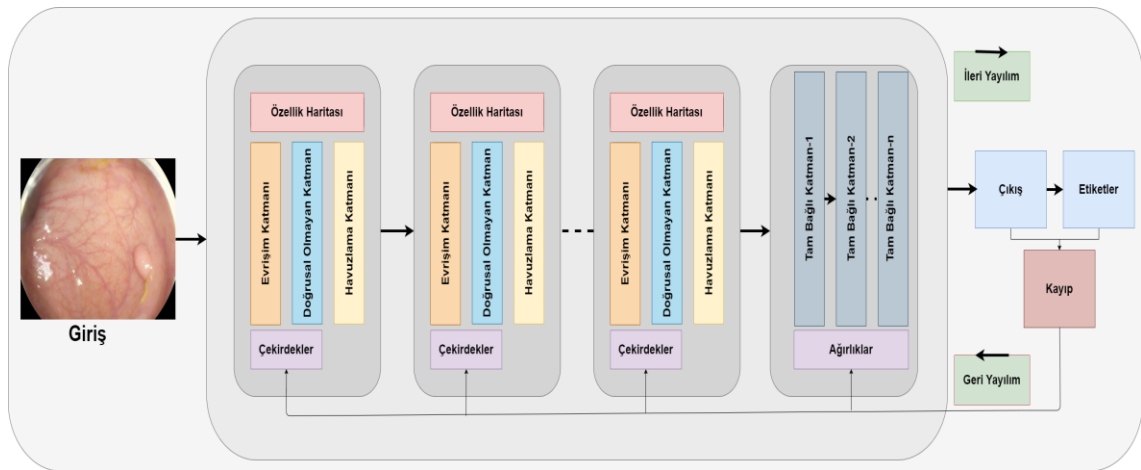
2.3.1 Evrimsel Sinir Ağları (CNN)

Evrimsel sinir ağları son zamanlarda elde ettiği performanslar sayesinde hem derin öğrenmeyi zirveye taşımıştır hem de birçok alanda CNN'lerin daha çok kullanılmasına olanak sağlamıştır [13,54,61]. Böylece, daha çok araştırmacının ilgi alanına girerek kullanımı en üst sıraya çıkmıştır. Burada, CNN mimarilerinin şekil ve temel yapılarına ait veriler Pacal ve arkadaşları tarafından derleme makalesindeki verilerden yararlanılmıştır. CNN mimarilerinin temel yapısı ve anlam kazandığı hususlar şu şekildedir. CNN'ler, çok katmanlı algılayıcının varyantlarıdır ve biyolojik olarak Hubel'in bir kedinin görsel korteksindeki çalışmasından esinlenmiştir. CNN'ler, görsel kalıpları doğrudan ham görüntü piksellerinden tanıma eğilimindedir. El yazısı karakterleri tanımak için önerilen ilk CNN modeli LeNet-5'tir [63]. CNN, katmanlarda matris çarpımı yerine evrişim adı verilen matematiksel bir işlemi gerçekleştiren yapay sinir ağı olarak tanımlanabilirler [61]. Derin öğrenmenin en yaygın kullanılan mimarisi olan CNN'ler, tıbbi görüntü işleme ve analizi gibi alanlarda en çok araştırılan ve

kullanılan makine öğrenme algoritmasıdır [13,62]. Bunun temel nedeni, CNN'lerin giriş görüntülerini alıp işlerken uzamsal ilişkileri ise koruyarak anlamsal özellikleri çıkarmasıdır. Kanserli dokunun normal doku ile olan ilişkileri ve etkileşimleri uzamsal ilişkiler aracılığıyla okunabildiğinden, tıbbi görüntü analizinde uzamsal ilişkiler çok önemli bir noktadır.

CNN'ler, görüntü işleme problemlerini çözmek için önerilen ve görüntülere oldukça iyi uyum sağlayabilen bir derin öğrenme mimarisidir. Bir CNN mimarisinde, katmanlar doğrudan bağlı olmak yerine çeşitli bloklarda bağlanır. Bu bloklar arasındaki bilgi aktarımı görsel kortekse benzer bir yapı taşır. CNN'ler klasik metotların problemlerini ortadan kaldırır. Ayrıca, ham verilerden özellikleri otomatik olarak öğrenebilir, böylece manuel özellik çıkarmanın zorluklarından kaçınır [64]. CNN'lerde özellikler paylaşıldığı için parametre sayısı olması gerekenden daha azdır. Bu şekilde, ağ daha hızlı öğrenmeyi sağlayabilir ve aşırı uyum problemlerinden kaçınılır. CNN'ler, verilere aç derin öğrenme mimarileridir ve milyonlarca parametre eğitilebilir, bu nedenle tıbbi görüntü analizinde iyi performans gösterirler. Benzer birçok avantajının olması, CNN'lerin daha birçok alanda kullanılmasının önünü açmaktadır [62].

CNN'ler, görüntü sınıflandırma, nesne algılama, görüntü bölümlleme, ses tanıma, metin işleme, video işleme ve tıbbi görüntü analizi gibi birçok bilgisayarlı görü çalışmasında kullanılmaktadır [13,64]. Bir CNN mimarisi genellikle evrişim, doğrusal olmayan, havuzlama, bırakma ve tamamen bağlı gibi temel katmanlardan oluşur. Genel bir CNN mimarisi ve CNN'nin temel katmanlarının bir örneği Şekil 2.3'de gösterilmektedir.

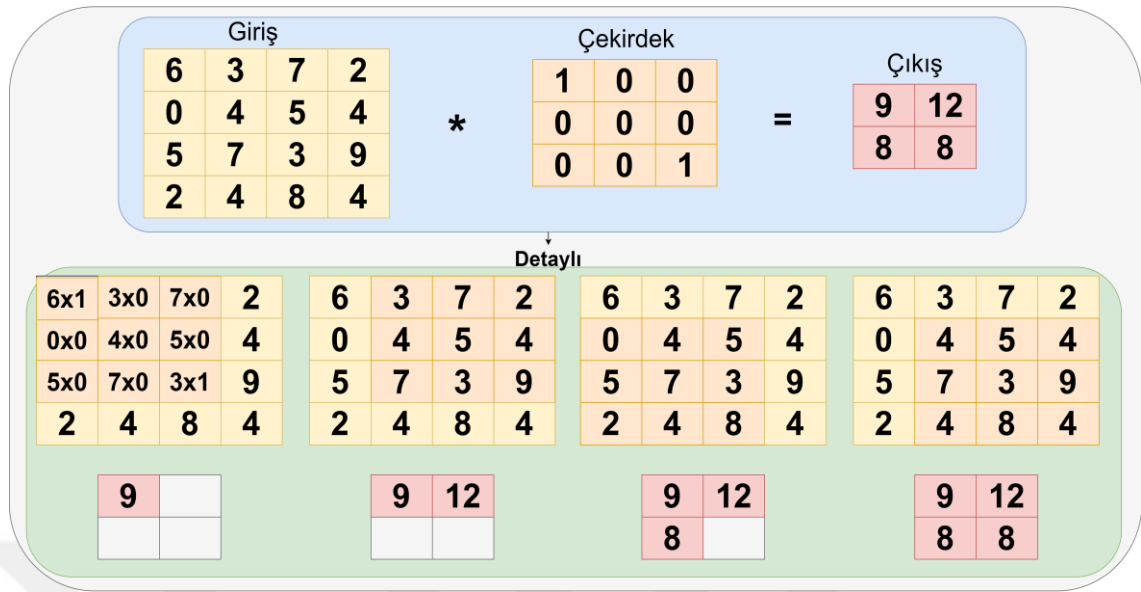


Şekil 2.3. Genel bir CNN mimarisi ve temel katmanları.

Evrişim katmanı: evrişimsel sinir ağlarının temelini oluşturan en önemli katman olan evrişim katmanı, belirli bir filtrenin tüm görüntü üzerinde dolaştırılması işlemine dayanmaktadır. Başka bir deyişle, bir çekirdek çerçevesi, giriş görüntüsünün veya önceki katmandan gelen görüntünün üzerinde sola veya sağa hareket ettirilerek görüntünün piksel değerleri ile çarpılır, daha sonra bunların toplamı alınır ve çıktı verilerinin karşılık gelen piksel alanına yazılır. Matrisin sınırına gelen çekirdek veya filtre istenilen birimde aşağı kaydırılır ve aynı işlemler tekrarlanır. Görüntünün tamamı işlendikten sonra bir özellik matrisi yani bir özellik haritası oluşturulur. Özellik haritası, her filtreye özgü özelliklerin keşfedildiği verileri içerir. Evrişim katmanında genellikle en önemli ve anlamlı özellikler çıkarılır. Evrişim sürecinin matematiksel ifadesi Eşitlik 2.1'de [24] gösterilmiştir.

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(i + m, j + n)K(m, n) \quad (2.1)$$

Eşitlikte, evrişim süreci '*' ile sembolize edilir, I girdidir, S evrişimden sonraki çıktıyı ifade eder ve K ise çekirdektir. Evrişimli bir katmanda, giriş katmanındaki görüntüye veya bir önceki katmanın çıkışındaki matrislere uygulanan birden fazla özellik matrisi vardır. Bu özellik matrisindeki parametreler eğitilebilir ve özellik matrislerinin her satır ve sütunundaki özellikler, her döngüden sonra yayılan gradyan değerine göre güncellenir. Evrişim katmanında, aynı öznelik haritasında aynı filtreler kullanılarak ağırlıklar paylaşılır, böylece ağırlık toplam parametre sayısı önemli ölçüde azaltılır. Filtre genişliği, filtre numarası, adım ve dolgu parametreleri, evrişim katmanında çok önemli parametrelerdir ve tasarımcılar tarafından belirlenir [25]. Evrişim işlemlerinde kullanılan filtrelerin boyutları, filtrenin bir merkeze sahip olması ve simetrik bir dolgu işlemi sağlaması için genellikle tek sayıdan oluşur. Çekirdek matrisi görüntü üzerinde hareket ederken, her adımda ne kadar hareket edileceği bilgisi stride (kayma miktarı) adı verilen parametre ile belirlenir. Doldurma ise çıktının boyutunu girdideki gibi tutmak için kullanılan bir tekniktir ve girdi matrisine simetrik olarak sıfır ekleme işlemidir. Şekil 2.4'te, bir giriş görüntüsü (4×4), bir evrişim çekirdeği (3×3) ve adım sayısı 1 ile evrişim işleminin bir sonucu olarak çıktı görüntüsünün bir örneğini göstermektedir.



Şekil 2.4. Bir giriş görüntüsü (4×4), bir çekirdek (3×3) ve adım sayısı 1 ile evrişim işlemi.

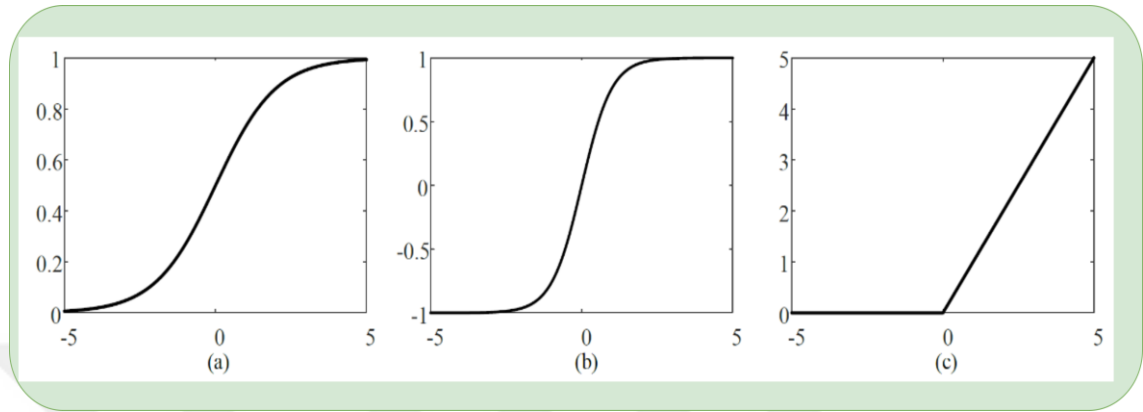
Farklı çekirdekler kullanılarak farklı özellikler keşfedilir ve istenen özellik haritalarını elde etmek için bu işlem kullanıcı tarafından istenilen parametre sayısı kadar tekrarlanır. Evrişim işleminden sonra elde edilen çıktı görüntüsünün boyutu aşağıdaki gibi hesaplanır.

$$\text{Çıkış matris boyutu} = \frac{(\text{giriş boyutu} + 2 \times \text{dolgu} - \text{çekirdek boyutu})}{\text{adım}} + 1 \quad (2.2)$$

Örneğin, giriş görüntüsü 128×128 ise, filtre boyutu 3×3 , dolgu 1 ve adım kaydırma sayısı 1 ise çıktı matrisi 128×128 'dir. Dolgu 0 ise, çıktı matrisi 126×126 olarak elde edilir.

Doğrusal olmayan katman: CNN mimarilerinin performansını önemli ölçüde etkileyen katmanlardan ve işlemlerden bir diğeri ise doğrusal olmayan aktivasyon fonksiyonlarının uygulanmasıdır yani doğrusal olmayan katmandır. Doğrusal olmayan aktivasyon fonksiyonları, önceki katmanlardan sonra doğrusal hale gelen ağı doğrusal olmayan bir forma getiren fonksiyonlardır. Ayrıca, aşırı veri değerleri olmadan bir sonraki katmandaki nöronların bağımsızlığını artırarak bir özellik haritası oluşturur, böylece tüm ağı kararlılığı artırılmış olmaktadır. Otomatik polip tespiti için kullanılan nesne tespit algoritmaları sıklıkla ReLU [65], Mish [66], Swish [67], SiLU [68] gibi daha karmaşık formda olan aktivasyon fonksiyonları kullanılmaktadır. Bu fonksiyonların

sayesinde CNN ağlarının öğrenme kabiliyeti artırılmış olmaktadır. Sigmoid, Tanh ve ReLU olarak adlandırılan bazı popüler doğrusal olmayan aktivasyon fonksiyonları Şekil 3'te gösterilmektedir [67].



Şekil 2.5. Bazı doğrusal olmayan aktivasyon fonksiyonları. (a) sigmoid; (b) tan; (c) ReLU.

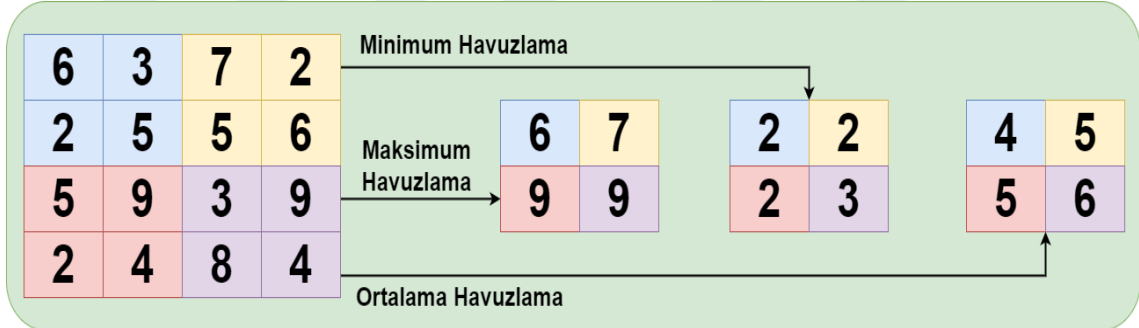
Sigmoid fonksiyonu için değer aralığı 0 ile 1 arasındadır. Bu nedenle, özellikle çıktı olarak olasılığı tahmin etmemiz gereken modellerde sıklıkla kullanılır. Hiperbolik tanjant işlevi (Tanh), sigmoid işlevine benzer şekilde, -1 ile 1 arasında bir değer aralığına sahiptir. Bunun avantajı, sıfır değerlerinin sıfıra yakından eşlenmesi ve negatif değerlerin güçlü bir şekilde negatif eşlenmesidir. Ayrıca, Tanh fonksiyonu orijine göre simetrik olmasına rağmen, sigmoid fonksiyonu simetrik değildir. Öte yandan, Rectified Linear Unit (ReLU), son zamanlarda evrimsel sinir ağlarında en çok kullanılan aktivasyon fonksiyonudur. ReLU için değer aralığı $\max(0, x)$ olarak tanımlanır. Negatif girdileri sıfır, pozitif girdileri ise değişmeden çıktıya yansıtan bir fonksiyondur. ReLU, sigmoid ve Tanh işlevlerinden çok daha hızlı hesaplayabildiği için CNN'lerde daha fazla kullanılır ve gradyan kaybolma problemini de hafifletebilir. Bu üç popüler fonksiyonun matematiksel formülleri ise sırasıyla Eşitlik 2.3, Eşitlik 2.4 ve Eşitlik 2.5'te gösterilmektedir.

$$f_{ReLU}(x) = \max(0, x) = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases} \quad (2.3)$$

$$f_{sigmoid}(x) = \frac{1}{1+e^{-x}} \quad (2.4)$$

$$f_{tanh}(x) = \frac{2}{1+e^{-2x}} - 1 \quad (2.5)$$

Havuzlama katmanı: bu katman bir önceki katman sonucunda elde edilen öz niteliklerin boyutunu küçültür ve benzer öz nitelikleri daha baskın hale getirir. Bunu, tanımlı filtreleri görüntü üzerinde dolaştırarak yapar. Başka bir deyişle, alt örnekleme işlemini gerçekleştirir. Böylece bir sonraki katman için veri boyutu küçülecek ve modelin ağırlığının büyümesi önlenmiş olacaktır. Ancak bu azalmanın bazı önemli bilgilerin kaybolmasına neden olduğu unutulmamalıdır. Bu katmanın en önemli faydası, ağıda hesaplanacak parametre sayısının azalması, böylece ağıın hesaplama karmaşıklığının azalması ve ağıın daha hızlı eğitim yapmasının gerçekleştirilmesidir. Havuzlama işlemi maksimum havuzlama, ortalama havuzlama ve minimum havuzlama olmak üzere üç şekilde yapılır. Havuzlamanın en yaygın şekli olan maksimum havuzlama, filtre boyutu alanındaki piksel değerleri arasında en büyük değerin belirlenmesi, minimum havuzlama ise en küçük değerin seçilmesidir. Ortalama havuzlama, filtre boyutu alanına düşen piksel değerlerinin toplamının filtre penceresi boyutuna bölünmesi ilkesine göre çalışır. Maksimum, minimum ve ortalama havuzlama işlemleri örnekleri Şekil 2.6’da gösterilmektedir.



Şekil 2.6. Bir giriş (4×4), bir filtre (2×2) ve adım sayısı 2 ile havuzlama işlemi örneği.

Seyreltme katmanı: Bir düzenlilik katmanı olan seyreltme katmanı, ağıın eğitim verisini çok fazla ezberlemesini önlemek için bazı düğümlerin bağlantısını keser [26]. Bu sayede ağıdaki bazı parametreler rastgele yok edildiğinden daha başarılı öğrenme gerçekleşebilir. Ağıdaki herhangi bir katmana uygulanabilir ve bu şekilde ağıın performansı arttırılabilir.

Tam bağlı katman: Tam bağlı katman yani Fully Connected Layers (FC Layers), CNN'lerde evrişim, doğrusal olmayan ve havuzlama katmanı katmanlarından sonra gelir. Bu katmanların çıktılarında elde edilen öz nitelik matrisleri alt alta düzenlenerek öz nitelik vektörüne dönüştürülür. Diğer bir deyişle düzleştirme işlemi yapılmış olur. Bu

katmanın sayısı mimariye bağılı olarak birden fazla olabilir. Bu özellik vektörü daha sonra klasik bir sinir ağı gibi bir etikete (yani sınıfa) sınıflandırılır. Çoklu sınıflandırması ve başarıları nedeniyle SoftMax, sınıflandırma görevi için en popüler yapılardan biridir. Çıktı, her nesne sınıfı için $[0,1]$ aralığında oluşturulur. Sınıflandırılacak nesne bir sınıfa dâhil edilir ve ardından en yüksek değere sahip olan çıkış nöronu ile eşleştirilir.

CNN'yi oluşturan katmanların dışında bazı önemli parametreler genellikle eğitim aşamasında kullanıcı deneyimine bırakılır. Bu parametrelerden bazıları momentum, eğitim oranı, mini küme boyutu, karıştırma, ilk öğrenme hızı ve hiper-parametreler olarak da bilinen bu parametrelerdir. Bu parametreler sayesinde ağ istenilen doğruluğu sağlamaya çalışır. Özellikle nesne tespit algoritmalarında bu katmanların yeri ve sayısı ile beraber yeni katmanlarda eklenebilir. Bu detayları her nesne tespit algoritmasının altında ayrıca detaylıca açıklanacağı için bu bölümde sadece genel ve temel katmanlar anlatıldı.

2.3.2. Diğer Derin Öğrenme Mimarileri

Bu bölümde, daha önce belirtildiği üzere dört sınıfa ayrılan derin öğrenme mimarilerden CNN hariç diğer üç önemli mimarinin kısa bir açıklaması sunulacak. Derin öğrenme tabanlı yapılan polip tespiti çalışmaların neredeyse hepsi CNN tabanlı olmasına rağmen, çok az çalışmada özellik çıkarımı gibi konularda az olsa diğer derin öğrenme mimarileri kullanılmaktadır. Bu yüzden bu bölümde bu mimarilerin kısaca genel hatlarına bakılacaktır. CNN mimarileri hariç, derin öğrenme mimarileri; tekrarlayan sinir ağları, oto kodlayıcılar ve kısıtlanmış Boltzmann makineleri ile beraber derin inanç ağları tabanlı mimarilerdir.

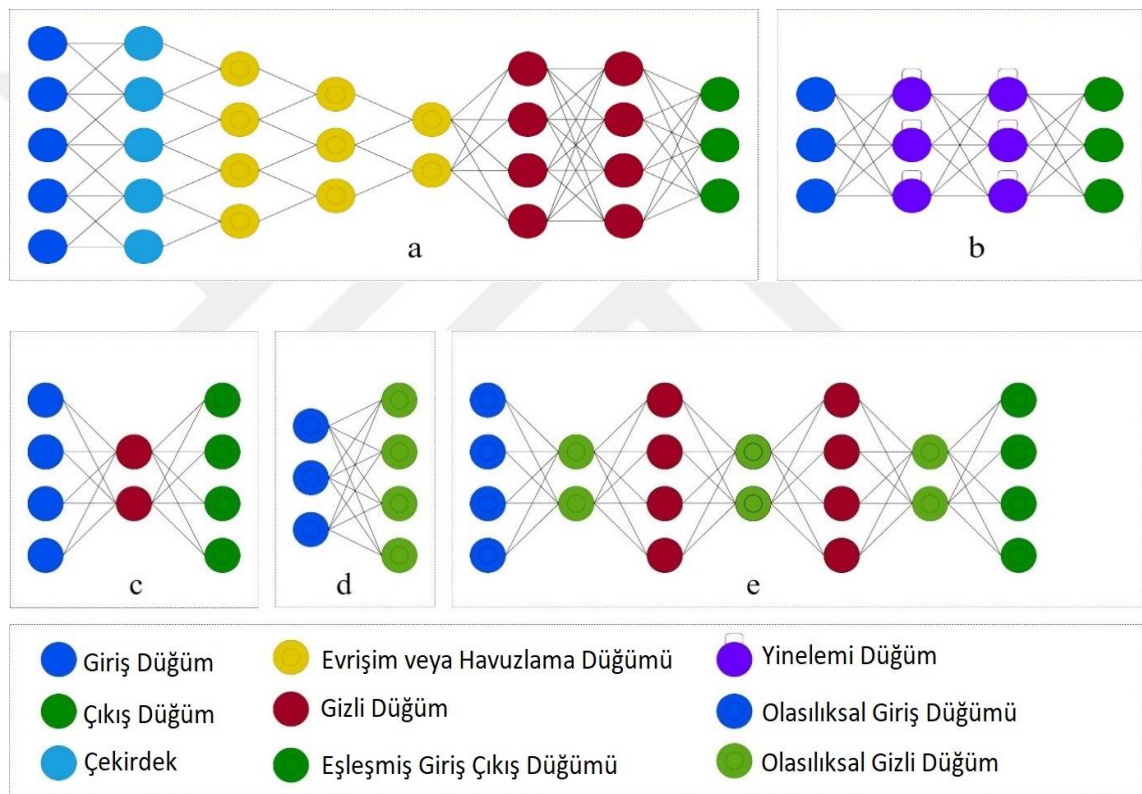
Tekrarlayan sinir ağları (RNNs): Tekrarlayan sinir ağları (RNN'ler) [69], sıralı verileri analiz etme ve bunlarla başa çıkmada diğer derin öğrenme yöntemlerinden daha sık kullanılır ve daha iyi performans gösterir. Özellikle, metin işleme ve metin üretme yetenekleri nedeniyle, RNN'ler konuşma tanıma, metin tahmini ve makine çevirisi gibi metin analizinde sıkça kullanılırlar [69]. Sıralanan verilerin doğasına bağılı olarak, RNN modelinin farklı zaman aralıklarındaki parametreler paylaşılır. Normal olarak, düz RNN'lerde, bir katmanın çıktısı bir sonraki girişe eklenir ve o katmana geri beslenir, bu da bir bellek kapasitesi sorununa neden olur. Bu zamansal gradyan problemleri, pratikte

ağların eğitimini zorlaştırır ve zamansal bağımlılıkları modellemeyi zorlaştırır. Bu soruna bir çare olarak LSTM'ler geliştirilmiş ve RNN'lerin sorunu bellek hücreleri ve çeşitli kapılar eklenerek giderilmiştir. Bu şekilde zamansal verilerdeki kısa veya uzun örüntüler öğrenilebilir hale gelmektedir. RNN'ler tıbbi görüntü işlemede nadiren kullanılır ve genellikle CNN'ler veya diğer derin öğrenme modelleri kullanılarak hibrit modeller şeklinde uygulanır. Bu hibrit modeller genellikle bölümleme işlemlerinde kullanılmaktadır. Tipik bir RNN şeması Şekil 2.7 (b)'de gösterilmektedir.

Otomatik kodlayıcılar (AEs): Otomatik kodlayıcılar [53], etiketli veriler olmadan girdi verilerinden özellikleri öğrenen denetimsiz bir derin öğrenme yöntemidir. AE'ler giriş katmanı, gizli katman ve çıkış katmanı olmak üzere üç katmandan oluşurlar. Yapı ileri beslemeli sinir ağlarına benzese de amaç, gizli katmandaki girdinin farklı bir temsilini oluşturmaktır. AE'ler girdi verilerini alır, onlardan kodlar toplar ve ardından bu kodları çıktı verilerini yeniden oluşturmak için kullanır. Temel mantık, giriş verilerinin çıkış verilerine mümkün olduğunca benzer olmasıdır. Bu nedenle girdiler ve çıktılar farklı olduğunda modeli cezalandıran bir maliyet fonksiyonu içerirler. AE'lerin etiketlenecek eğitim verilerine ihtiyacı yoktur. Kodlamalar genellikle daha küçük boyuttadır, bu da hesaplama karmaşıklığını azaltır. Ayrıca, girdi eğitim verilerine benzer çıktılar üretirler. Tüm bu avantajları ile etiketli verilerin kıt olduğu tıbbi veri işlemede CNN'den sonra sıklıkla kullanılmaktadır. AE'lerin pek çok türü vardır ve en yaygın olarak kullanılanları; Yığılmış Otomatik Kodlayıcılar, Gürültü Giderici Otomatik Kodlayıcılar, Seyrek Otomatik Kodlayıcılar ve Varyasyonlu Otomatik Kodlayıcılar olarak sayılabilirler.

Kısıtlanmış Boltzmann makineleri (RBMs) ve derin inanç ağları (DBNs): Kısıtlı Boltzmann Makineleri (RBM) ilk olarak 1986'da tanıtıldı, ancak daha sonra 2006'da Geoffrey Hinton ve diğerleri tarafından hızlı bir öğrenme algoritması olarak tanındı [69]. RBM, Boltzmann Makinelerinin bir çeşididir ve girdi setindeki olasılık dağılımını öğrenebilen üretken bir rastgele sinir ağıdır. RBM'ler, aralarında simetrik bağlantılar bulunan, gizli ve görünür olan iki parçalı grafiklerdir. RBM'ler, özellik öğrenme, sınıflandırma ve boyut küçültme gibi alanlarda yaygın olarak kullanılmaktadır. RBM'lerin ve DBN'lerin tipik bir grafiği sırasıyla Şekil 3.7. (d) ve Şekil 3.7 (e)'de gösterilmektedir. Derin İnanç Ağları, her seferinde bir katman oluşturmak için denetimsiz bir açgözlü öğrenme algoritmasıyla Hinton ve diğerleri tarafından önerildi

[35]. Temel fikir, önceki katmanın çıktısını modelleyen bir RBM ağı kullanmak ve bu RBM ağını ağın her katmanını bağımsız olarak eğitmek için kullanmaktır. Birden fazla ve daha basit RBM modellerini bu süreçle birleştirmek, farklı bir modeli öğrenmenin etkili bir yoludur. Bir DBN, denetlenmeyen bir sıralı eş görünüm üzerinde eğitildiğinde, girdilerin olasılıksal yeniden üretimini öğrenebilir. DBN'nin katmanları daha sonra bir özellik çıkarıcı görevi görür. Bu şekilde öğrendikten sonra, bir DBN, sınıflandırmak için denetimli bir şekilde daha fazla eğitilebilir. Ana derin öğrenme mimarilerinin tipik bir tablosu Şekil 2.7'de gösterilmektedir.



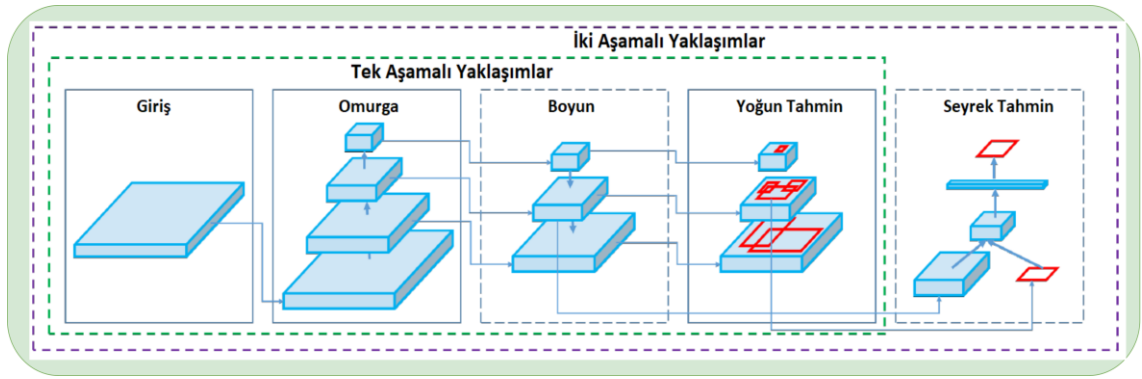
Şekil 2.7. Beş ana derin öğrenme mimarileri iki ana kategoriye ayrılabilir: denetimli öğrenme algoritmaları; CNN'ler (a) ve RNN'ler(b) ve denetimsiz öğrenme algoritmaları; AE'ler (c), RBM'ler (d) ve DBN'ler (e).

2.4. Polip Tespitinde Güncel Nesne Tespit Algoritmaları

Bu bölümde nesne tespit algoritmalarının detaylı açıklamalarıyla beraber güncel yaklaşımlar sunulacak. Özellikle otomatik polip tespitinde başarılı sonuçlar üreten nesne tespit algoritmalarının geniş bir tanımı olacaktır. Genel nesne algılama, herhangi bir görüntüdeki mevcut nesneleri bulmayı, bu nesneleri sınıflandırmayı ve bu nesnelerin

yerini göstermek için bunları dikdörtgen formda yer alan bir sınırlayıcı kutularla etiketlemeyi amaçlar [65]. Böylelikle görüntüde yer alan nesne hem sınıflandırılmış olur hem de görüntüdeki yeri dikdörtgen içeresine alınarak kesin yeri bulunmuş olur. Ya da başka bir deyişle; nesne algılama, nesne sınıflandırma ve nesne yerelleştirme olmak üzere iki görevi kapsayan bir tekniktir. Birden çok nesne sınıfının varlığını ve konumunu tespit etmek için eğitilmiş bir modeldir. Bu, statik görüntülerde veya hatta videolarda gerçek zamanlı olarak kullanılabilir.

Derin öğrenme tabanlı nesne tespit algoritmaları genel olarak iki ana kategoriye ayrılabilir. Bu kategori sırasıyla iki aşamalı ve tek aşamalı olarak ayrılır. Tek aşamalı nesne tespit algoritmaları yüksek çıkarım hızlarına sahip iken, iki aşamalı nesne tespit algoritmaları ise yüksek yer belirleme ve tanıma doğruluğuna sahiptir. Fakat son zamanlarda tek aşamalı nesne tespit algoritmaları bu mustarip durumdan kurtularak iki aşamalı yaklaşımlarla aynı doğruluğu yakalamaktadırlar. İki aşamalı yaklaşımlarda ilk önce bir dizi temel öneri oluşturulur ve sonraki aşamada bu önerilerden öznelik vektörleri çıkarılır, bunlar daha sonra derin evrişimli sinir ağları ile kullanılır ve ardından önceden tanımlanmış sınıf tahminleri yapılır. Öte yandan, tek aşamalı yaklaşımlarda ise giriş görüntüsü üzerindeki tüm konumlar olası hedef nesneler olarak değerlendirir ve her bir ilgi alanını bir hedef nesne veya basitçe arka plan olarak kategorize etmeye çalışır. Tek aşamalı yaklaşımlar oldukça hızlıdır ve gerçek zamanlı nesne algılama uygulamaları için en ideal yaklaşımlardır. Öne çıkan iki aşamalı nesne algoritmalarının en bilineni Daha hızlı R-CNN (Faster R-CNN) [70] iken, tek aşamalı yaklaşımlarda ise YOLO [18] algoritması en ön planda rol oynar. Bunlara ek olarak nesne tespit algoritmaları bölge önerisi tabanlı (region proposal based) yöntemler ve sınıflandırma (classification based) tabanlı yöntemler olmak üzere iki kategoride sınıflandırılmaktadır. Bölge önerisi tabanlı yöntemler arasında R-CNN, Daha Hızlı R-CNN vb. bulunurken, sınıflandırma tabanlı yöntemler ise YOLO ve SSD [71] gibi algoritmaları içerir. Nesne tespit algoritmaları burada tek aşamalı ve iki aşamalı yöntemler olmak üzere iki alt ana başlıkta incelenecektir ve bu algoritmaların en önemli mimarilerinden bahsedilecek ve son olarak ise otomatik polip tespiti en başarılı sonuçlar üreten YOLO algoritmasından daha detaylı bahsedilecektir. Tek ve iki aşamalı yöntemlere ait genel yapı Şekil 2.8’de gösterilmiştir.



Şekil 2.8. Tek aşamalı yaklaşımlara ve iki aşamalı yaklaşımlara ait blok diyagramları.

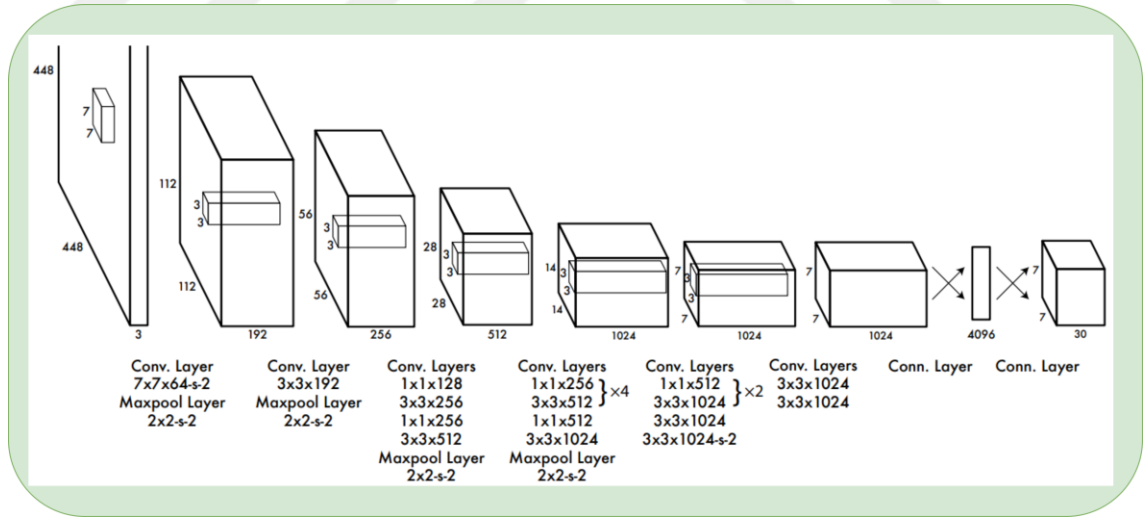
Şekil 2.8’de de görüldüğü üzere bir modern nesne tespit algoritması ya da yaklaşımı giriş verisi ya da görüntüsü, omurga, boyun ve sınıfları veya nesne sınırlayıcı kutularını tahmin etmek için kullanılan bir kafa olmak üzere dört bölümden oluşmaktadır. Bazı çalışmalarda bu yapılar iki ana sınıf yani boyun ve kafa olmak üzere iki sınıfa ayrılabilir. Omurgalar genelde VGG, ResNet, DenseNet gibi CNN mimarilerinden kullanılarak faydalı özellikler çıkarılır. Boyun kısmı ise omurga ve kafa arasındaki farklı aşamalardan özellik haritalarını toplamak için görevlendirilmiştir. Kafa kısmı ise sınıflandırma ve tespit işlemlerinin yürütülmesinde nesne tespiti yaklaşımlarının son bölümünü oluşturur.

2.4.1. Tek Aşamalı Yaklaşımlar

Tek aşamalı yaklaşımlar ya da detektörler, bölge önerme veya teklif bölgelerini üretmek için ayrı bir fonksiyonel göreve sahip olmamakla beraber görüntüler üzerinden sınırlayıcı kutular tahmin edilir ve böylece algılama hızı artırılmış olur. Başka bir deyişle, bu yaklaşımlar, temel olarak bir giriş görüntüsünün tüm konumlarını olası nesneler olarak değerlendirir ve ayrıca her bir ROI’yi (ilgi bölgesi) arka plana veya önceden tanımlanmış bir nesne sınıfına kategorize etmeye çalışan tek bir ileri beslemeli tam evrişimli ağ içerir. Bu nedenle YOLOv4 [72] çıkmadan önce, bu yaklaşımlar düzensiz şekilli nesneleri veya bir grup küçük nesneyi tanımada o kadar iyi değildiler. Fakat YOLOv4 ve sonraki nesil nesne tespiti yaklaşımlarında doğruluktan mustarip bu yaklaşımlarda bu sorunlar çözüme kavuşarak bu yapıların daha da popüler olmasını sağlamıştır. Tek aşamalı yaklaşımların en büyük avantajı ise bu algoritmaların genellikle çok aşamalı yaklaşımlardan daha hızlı ve yapısal olarak daha basit olmasıyla

beraber gerçek zamanlı uygulamalarda kullanılan tek yapılarıdır. Şimdi bu yapıları en popüler bazı algoritmaların yapılarına bakalım.

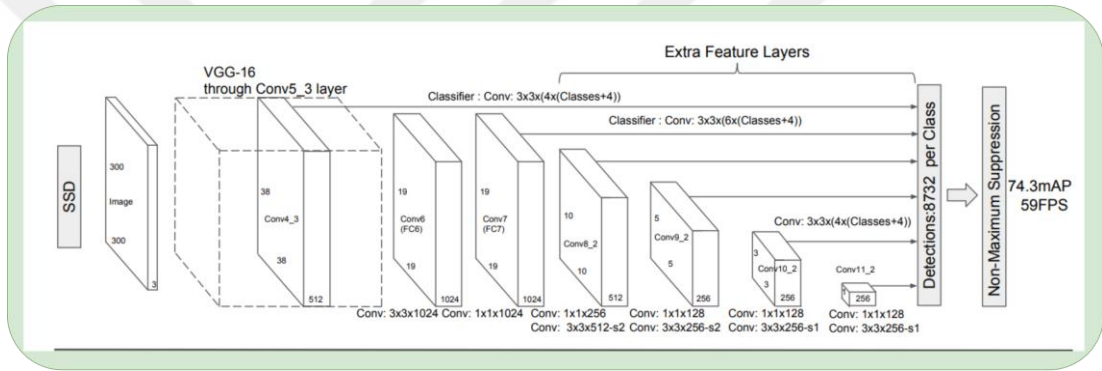
YOLO (you only look once): YOLO ilk kez 2015 yılında, R. Joseph ve ark. tarafından, CNN tabanlı derin öğrenmeye dayalı gerçek zamanlı tek aşamalı bir yaklaşım olarak önerildi. YOLO, algılanan nesnelerin farklı bileşenlerinin tek bir sinir ağına toplayarak daha hızlı işlem yapılmasına yani çıkarım hızını artıran bir metottur. YOLO, öncelikle giriş görüntüsünü alır ve her bir sınırlayıcı kutuyu tahmin etmek için özelliklerini kullanır. Her görüntü $S \times S$ ızgaralarına bölünür ve her ızgara, N sınırlayıcı kutu ve bu kutular için güven puanı oluşturur. Güven, doğru görüntüyü kapsasa da içermese de, sınırlayıcı kutunun doğruluğunu ortaya çıkarır. YOLO bu sayede oldukça hızlıdır ve gerçek zamanlı ortamlar için popüler bir tekniktir. R. Joseph ve ark. YOLO üzerinde bir dizi iyileştirme sağlayarak YOLOv2 [73] ve YOLOv3 sürümlerini önerdiler. Bu sürümler sadece tespit doğruluğunu iyileştirmekle beraber aynı zamanda tespit hızını da iyileştiriyor. YOLOv1 mimarisine ait yapı Şekil 2.9'da gösterilmiştir. YOLO algoritmalarının detayları YOLO algoritmaları adlı bölümde detaylıca incelenecektir.



Şekil 2.9. YOLOv1 mimarisinin yapısı.

SSD (Single Shot Multibox Detektör): SSD, adından da anlaşılacağı gibi, bir görüntüdeki çok sayıda nesneyi algılamak için yalnızca tek bir seferde işlemleri gerçekleştiren yani evrişimli ağı yalnızca bir kez çalışacağı ve özellik haritasını tahmin edileceği anlamına gelir. SSD'nin en önemli katkılarının başında ise küçük nesneleri algılamak için tek aşamalı yaklaşımların algılama doğruluğunu istisnai olarak geliştiren

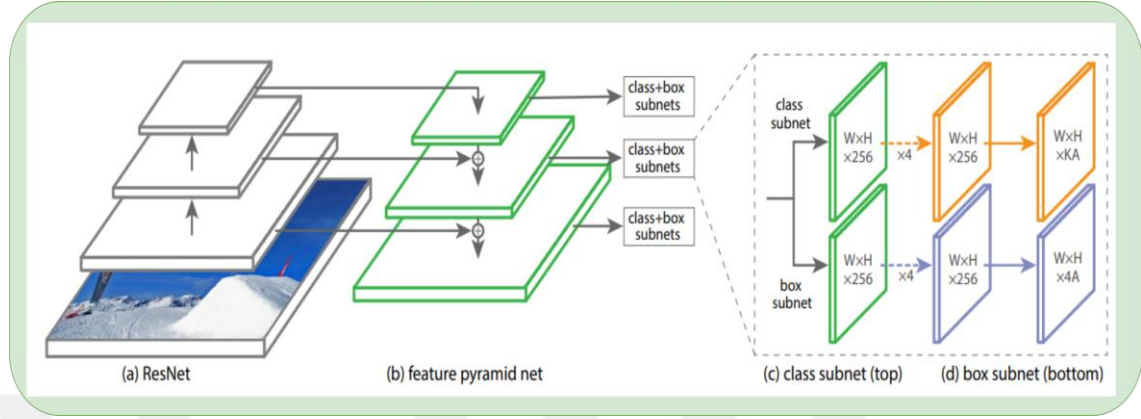
çoklu referans ve çoklu çözünürlük tekniklerinin kullanılmasıdır. Yani SSD, çeşitli en boy oranlarına ve ölçeklere sahip bağlantı kutularını kullanır. Değişken boyutlu nesneleri tahmin etmek için ağ, çeşitli özellik haritası tahminlerini farklı çözünürlüklerle birleştirir. SSD de giriş görüntüsünü matris hücrelerine böler, ancak YOLO'daki gibi önceden tanımlanmış ızgara hücrelerinden nesneleri tanımanın aksine, sınırlayıcı kutuları iyileştirmek için çeşitli ölçek, boyut ve en boy oranlarına sahip bir dizi bağlantı üretilir. SSD nesne tespit algoritmasını eğitilmesi ve nesne algılama bileşeni gerektiren yazılım sistemlerine dâhil edilmesi kolaydır. Diğer tek aşamalı yöntemlere kıyasla SSD, daha küçük girdi görüntü boyutlarında bile çok daha iyi bir doğruluğa sahiptir. SSD algoritmasının temel yapısı Şekil 2.10'da gösterilmiştir.



Şekil 2.10. SSD mimarisinin yapısı.

RetinaNet: Lin ve ark. tarafından önerilen odak kaybını bir sınıflandırma kaybı olarak kabul ederek çalışan tek aşamalı bir nesne tespit algoritmasıdır [74]. RetinaNet, bir omurga ağından ve iki göreve özel alt ağdan oluşan tek, bütünleşik bir yapıdır. Bu yapıda, omurga, tüm bir girdi görüntüsü üzerinde özellik haritası çıkarmaktan sorumlu olmaktadır. İki ana ağ yapısından oluşan RetinaNet algoritması, ilk alt ağ ile omurganın çıktısını kullanarak nesne sınıflandırması gerçekleştirmekle görevli iken, ikinci alt ağ ile de omurganın çıktısını kullanarak sınırlayıcı kutu regresyonu gerçekleştirmekle sorumludur. Ayrıca, RetinaNet odak kaybı ile tek aşamalı yaklaşımlarda var olan bazı sorunlara çözüm bulmuştur. Odak kaybı, zorlu eğitim örneklerine odaklanarak eğitim sırasında yapıyı kayba uğratan çok sayıda kolay olumsuz örnekle uğraşmaktan kaçınır. RetinaNet, önceki tek aşamalı yaklaşımların yüksek hızını devir almakla beraber aynı zamanda dengesiz pozitif ve negatif örnekleri eğitmekle tek aşamalı yaklaşımların sorun

yaşadığı kısmı büyük ölçüde üstesinden çözüme kavuşturur. RetinaNet'e ait genel yapı Şekil 2.11'te gösterilmiştir.



Şekil 2.11. RetinaNet mimarisinin yapısı.

Tek aşamalı yaklaşımlardan olan RetinaNet ağ mimarisi özellik çıkarıcı olarak CNN omurgası olarak ResNet mimarisini kullanırken, ResNet üzerinde ise Özellik Piramidi Ağı (FPN) omurgası kullanılmaktadır. Bu yapılardan sonra iki alt ağı ayrılan yapıda ilk ağ bağlantı kutularını sınıflandırmakla görevli iken diğer alt ağ ise bağlantı kutularından yer gerçeği nesne kutularına döndürmesiyle görevlidir.

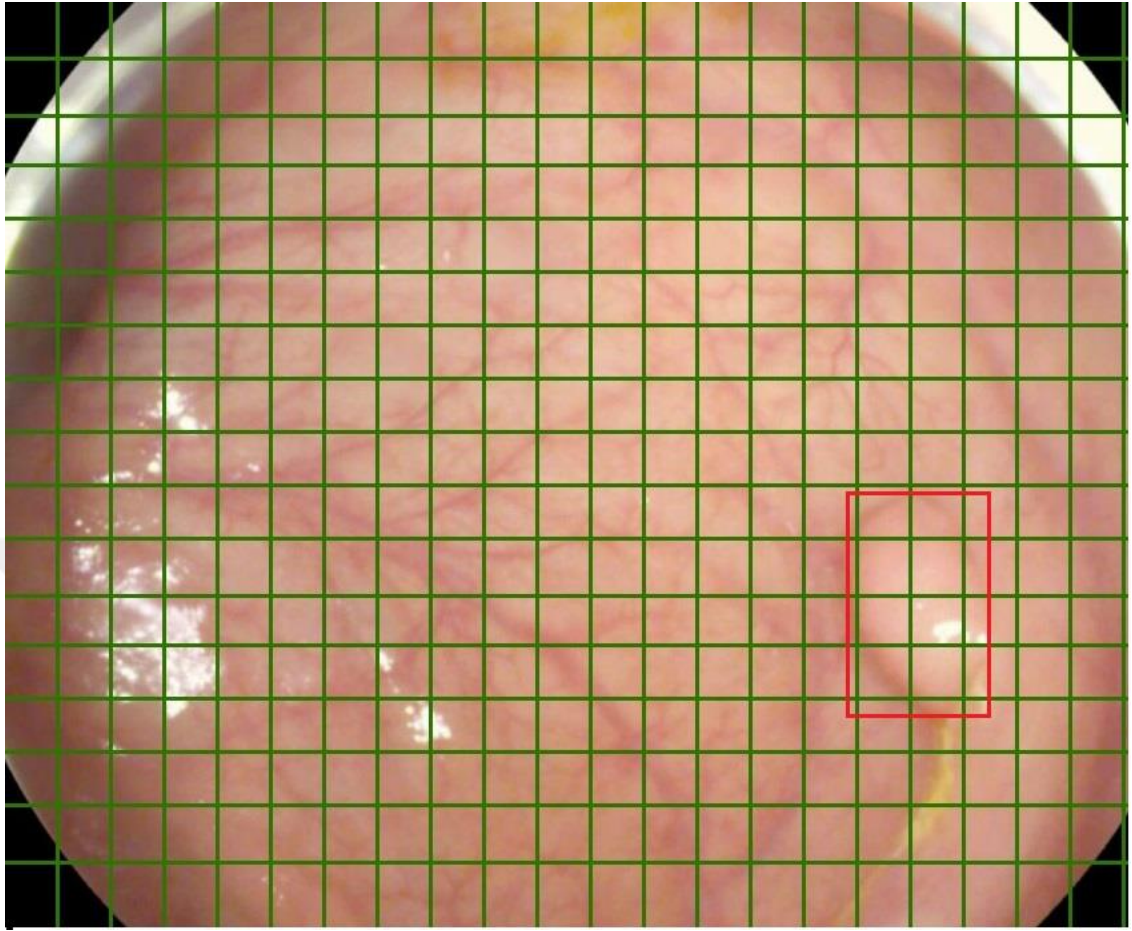
2.4.1.1. YOLO algoritmaları

YOLO algoritmaları ilk çıktıkları 2015 (YOLO) yılından bu yana gelişerek birçok gelişmiş sürümlere sahip olmuştur. Bu algoritmaların detaylarına geçmeden önce kısaca YOLO algoritmalarının temel çalışma mantıkların bakarak algoritmanın temelini anlamaya çalışacağız. YOLO algoritmalarının temelindeki mantık özellik çıkarma işlemini bir CNN vasıtasıyla yaparak bütün işlemleri tek bir seferde yapma eğilimindedirler ve bu yüzden tek aşamalı yaklaşımların popüler bir nesne tespit algoritması haline gelmiştir. YOLO, gerçek zamanlı nesne algılama sağlamak için sinir ağlarını kullanan bir algoritmadır. Bu algoritma, hızı ve doğruluğu nedeniyle popülerdir. Trafik sinyallerini, insanları, medikal görüntüleri ve daha birçok nesneyi hem sınıflandırmak hem de algılamak için çeşitli uygulamalarda kullanılır.

YOLO, 'You Only Look Once' teriminin kısaltmasıdır. Bu, bir resimdeki çeşitli nesneleri (gerçek zamanlı olarak) algılayan ve tanıyan bir algoritmadır. YOLO'da nesne tespiti, bir regresyon problemi olarak yapılır ve tespit edilen görüntülerin sınıf

olasılıklarını sağlar. YOLO algoritması, nesneleri gerçek zamanlı olarak algılamak için CNN kullanır. Adından da anlaşılacağı gibi, algoritma nesneleri algılamak için bir sinir ağı üzerinden yalnızca tek bir ileri yayılım gerektirir. Bu, tüm görüntüdeki tahminin tek bir algoritma çalıştırmasında yapıldığı anlamına gelir. CNN, çeşitli sınıf olasılıklarını ve sınırlayıcı kutuları aynı anda tahmin etmek için kullanılır. YOLO algoritması neden önemlidir diye bir soru sorulduğunda şu avantajlar ön plana çıkmaktadır. İlk avantaj olarak hız kabul edilir. Bu algoritma, nesneleri gerçek zamanlı olarak tahmin edebildiği için algılama hızını artırır. İkinci plandaki avantajı ise yüksek doğruluk sunmasıdır. YOLO, minimum arka plan hatasıyla doğru sonuçlar sağlayan tahmine dayalı bir tekniktir. Son olarak ise YOLO ve YOLO varyantlarında gittikçe gelişmiş olan öğrenme yetenekleridir. Algoritmanın, nesnelerin temsillerini öğrenmesini ve bunları nesne algılamada uygulamasını sağlayan mükemmel öğrenme yetenekleri vardır.

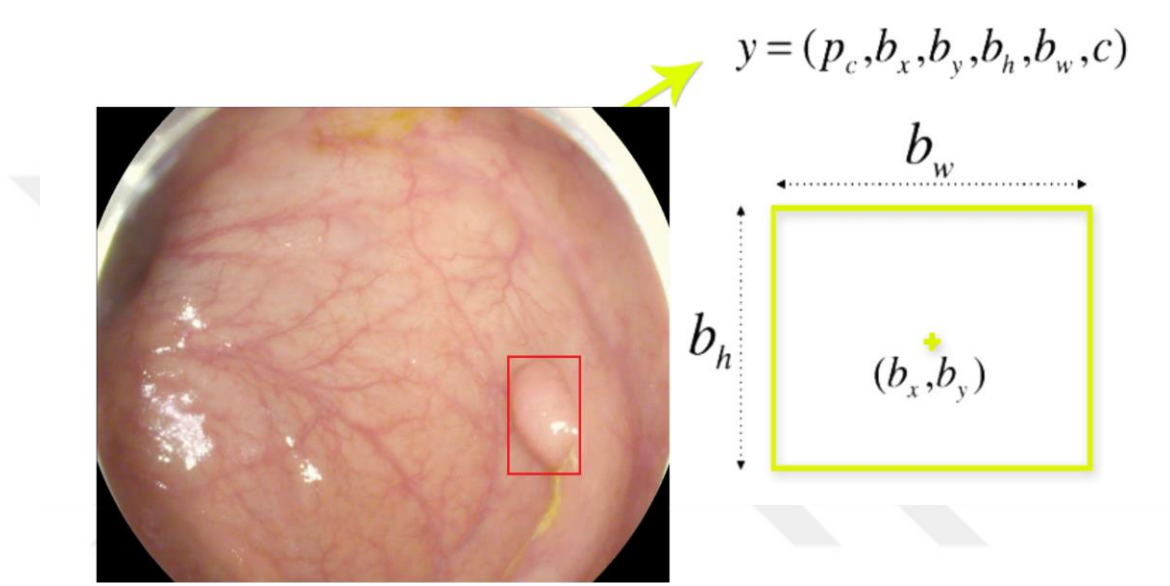
YOLO algoritmasının çalışma mekanizmasına bakıldığında sırasıyla artık bloklar (Residual blocks), Sınırlayıcı kutu regresyonu (Bounding box regression) ve birleşme üzerinde kesişme IOU (Intersection Over Union) gibi yapıların açıklanmasıyla başlanabilir. Artık bloklardaki çalışma mantığı şu şekildedir. İlk olarak, görüntü çeşitli ızgaralara bölünür. Her ızgara $S \times S$ boyutuna sahiptir. Aşağıdaki görüntü, bir girdi görüntüsünün ızgaralara nasıl bölündüğünü gösterir.



Şekil 2.12. Polip görüntüsünün ızgaralara bölünmüş hali.

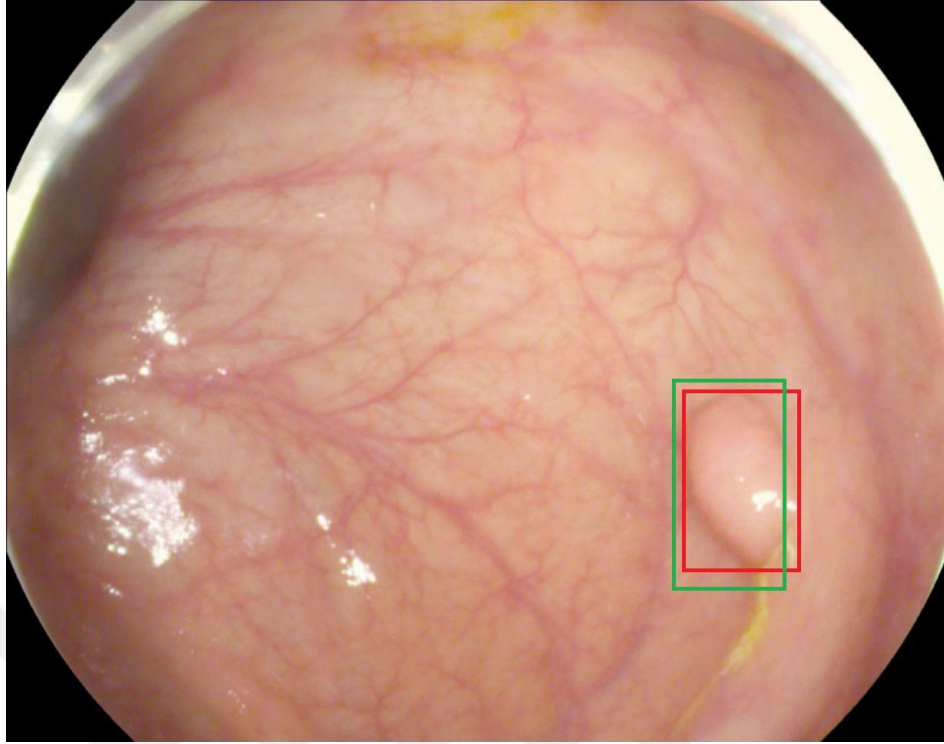
Şekil 2.12, eşit boyutta birçok ızgara hücresi olabilir ama bu örnekte yeşil renkte $S \times S$ 'lık bir ızgara kullanılmıştır. Kırmızı renkte olan kısım ise polipli alandır. Bu gibi durumlarda polipli resim küçük olduğu için daha küçük ızgara boyutlarının kullanılması gereklidir. Her ızgara hücresi, içinde görünen nesneleri algılayacaktır. Örneğin, belirli bir ızgara hücresi içinde bir nesne merkezi belirirse, bu hücre onu algılamaktan sorumlu olacaktır. Görüntüde bulunan her nesne için, bir ızgara hücresinin onu tahmin etmekten "sorumlu" olduğu söylenir. Bu, nesnenin merkezinin içine düştüğü hücredir. Her ızgara hücresi, B sınırlayıcı kutuların yanı sıra C sınıfı olasılıkları da tahmin eder. Sınırlayıcı kutu tahmininin 5 bileşeni vardır: $(x, y, w, h, \text{güven})$, (x, y) koordinatları, ızgara hücresi konumuna göre kutunun merkezini temsil eder (kutunun merkezi ızgara hücresinin içine düşmezse, bu hücrenin bundan sorumlu olmadığını unutmayın). Bu koordinatlar 0 ile 1 arasında olacak şekilde normalleştirilir. (w, h) kutu boyutları da görüntü boyutuna göre $[0, 1]$ 'e normalleştirilir. Aşağıdaki resimde bir sınırlayıcı kutu örneği gösterilmektedir. Sınırlayıcı kutu sarı bir ana hatla temsil edilmiştir.

Sınırlayıcı kutu regresyonu ise şu şekilde açıklanabilir. Sınırlayıcı kutu, görüntüdeki bir nesneyi vurgulayan bir ana hatıdır. Görüntüdeki her sınırlayıcı kutu sırasıyla, genişlik (b_w), yükseklik (b_h), sınıf (örneğin, polip, kişi, araba vb.) c harfi ile gösterilir ve sınırlayıcı kutu merkezi (b_x, b_y) gibi özniteliklerden oluşur. Aşağıdaki resimde bir sınırlayıcı kutu örneği gösterilmektedir. Sınırlayıcı kutu yeşil bir ana hatla temsil edilmiştir.



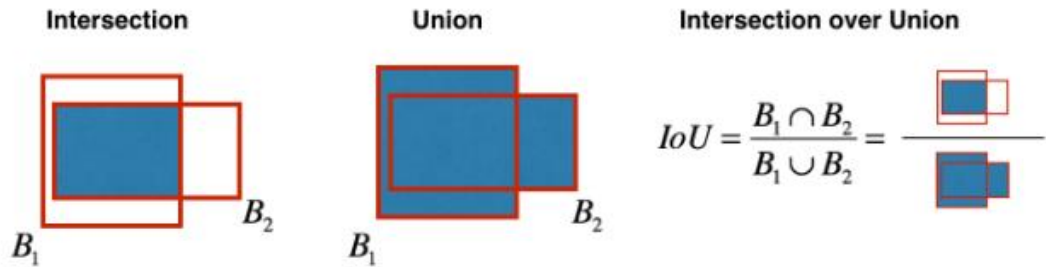
Şekil 2.13. Polip görüntüsüne ait sınırlayıcı kutu ve sınırlayıcı kutunun oluşma süreci.

YOLO, nesnelerin yüksekliğini, genişliğini, merkezini ve sınıfını tahmin etmek için tek bir sınırlayıcı kutu regresyonu kullanır. Yukarıdaki resimde, sınırlayıcı kutuda görünen bir nesnenin olasılığını temsil eder. Birleşme üzerinde kesişme (IOU) işlemi ise şu şekilde açıklanabilir. Birleşim üzerinden kesişme (IOU), kutuların nasıl üst üste bindiğini açıklayan nesne algılamada bir olgudur. YOLO, nesneleri mükemmel bir şekilde çevreleyen bir çıkış kutusu sağlamak için IOU'yu kullanır. Her ızgara hücresi, sınırlayıcı kutuları ve bunların güven puanlarını tahmin etmekten sorumludur. Öngörülen sınırlayıcı kutu gerçek kutuyla aynıysa, IOU 1'e eşittir. Bu mekanizma, gerçek kutuya eşit olmayan sınırlayıcı kutuları ortadan kaldırır. Aşağıdaki görüntü, IOU'nun nasıl çalıştığına dair basit bir örnek göstermektedir.



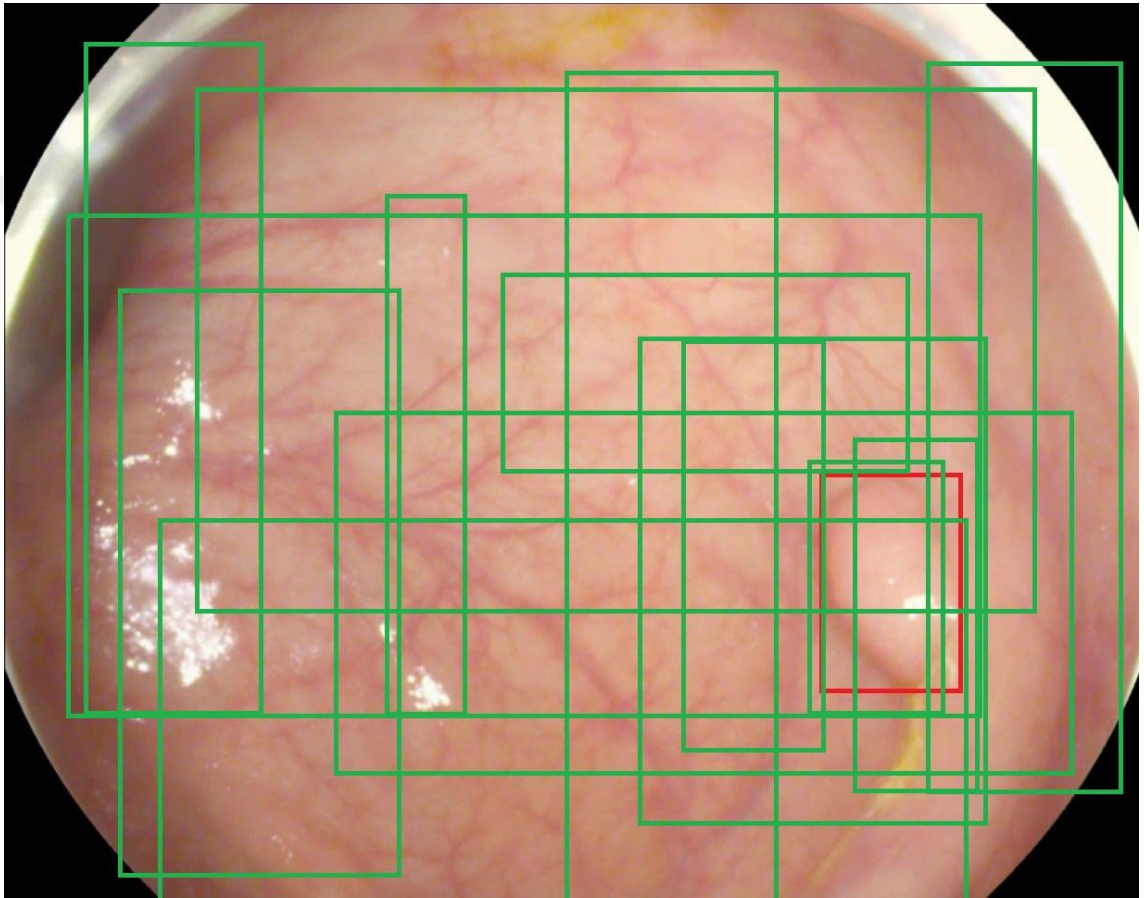
Şekil 2.14. Polip görüntüsüne ait gerçek kutu (kırmızı renk) ve tahmin edilen kutu (yeşil renk).

Şekil 2.14'te biri yeşil, diğeri kırmızı olan iki sınırlayıcı kutu vardır. Kırmızı kutu gerçek kutu iken yeşil kutu ise tahmin edilen kutudur. YOLO, iki sınırlayıcı kutunun eşit olmasını sağlar. Böyle bir durumda yani kutuların eşit olması durumunda güven skoru 1 olur iken, yukarıdaki resimdeki gibi bir durum olduğunda ise güven skoru 1'in altındadır yani 0.8 gibi bir değere yakındır denilebilir. IoU'ya ait aşağıdaki Şekil 2.15 hesaplamamanın nasıl yapıldığını görselleştirmektedir.



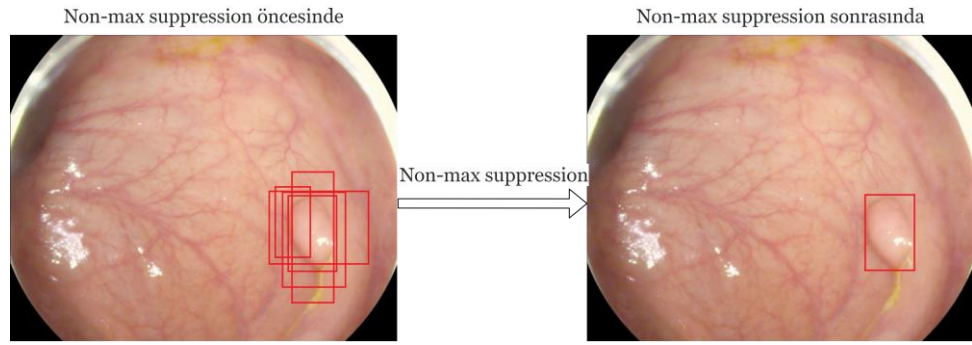
Şekil 2.15. IoU'nun nasıl hesaplandığını gösteren temsiller ve formüller.

YOLO algoritmasında birçok ızgara tespit işlemine katılabilir. Bu, her ızgara hücresi için sınıf olasılıklarını tahmin etmenin öncesini ve sonrasını gösterir. Sınıf olasılıklarını tahmin ettikten sonraki adım, maksimum olmayan bastırma (Non-max suppression), algoritmanın gereksiz bağlantı kutularından kurtulmasına yardımcı olur, aşağıdaki Şekil 2.16 görebileceğiniz gibi, sınıf olasılıklarına göre hesaplanmış çok sayıda bağlantı kutusu vardır.



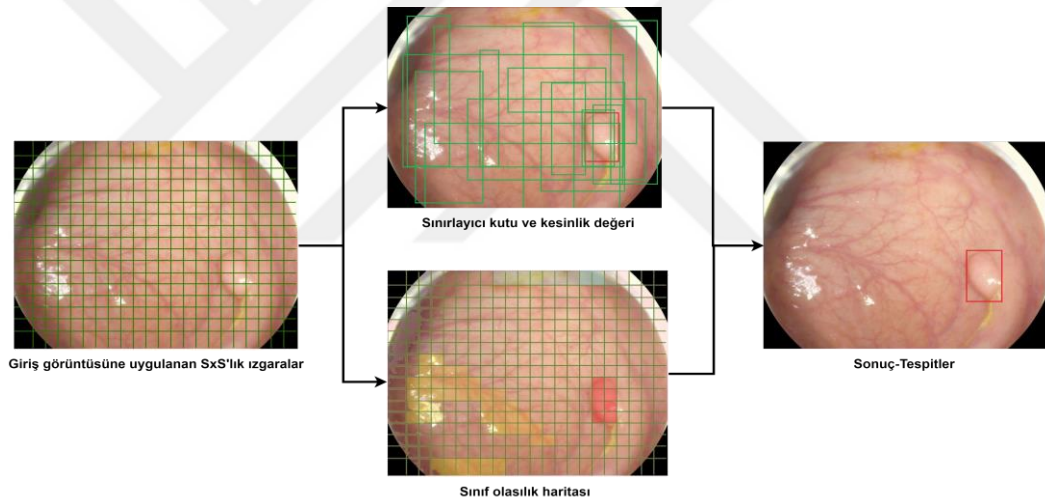
Şekil 2.16. Polip tahmini için YOLO'nun ürettiği tahmin kutuları

Bu sorunu çözmek için maksimum olmayan bastırma, aralarında en yüksek sınıf olasılığına sahip olan IoU'yu önceden oluşturarak çok yakın olan sınırlayıcı kutuları ortadan kaldırır. En yüksek sınıf olasılığına sahip olana ilişkin tüm sınırlayıcı kutular için IoU değerini hesaplar, ardından IoU değeri bir eşikten büyük olan sınırlayıcı kutuları reddeder. Bu iki sınırlayıcı kutunun aynı nesneyi kapsadığını, ancak diğerinin aynı olma olasılığının düşük olduğunu, dolayısıyla elendiğini belirtir. Bir kez yapıldıktan sonra, algoritma bir sonraki en yüksek sınıf olasılıkları ile donatılan kutuyu bulur ve aynı işlemi yapar, tüm farklı sınırlayıcı kutular kalana kadar yapılır.



Şekil 2.17. NMS öncesi ve sonrasına ait polip görüntüleri ve tahmin kutuları.

Sonuç olarak bu üç tekniğin kombinasyonu ile YOLO algoritmasının çıktısı elde edilir. Aşağıdaki Şekil 2.18, nihai tespit sonuçlarını üretmek için üç tekniğin nasıl uygulandığını göstermektedir.



Şekil 2.18. YOLO algoritmasının polip görüntüsü üzerinde çalışmasını gösteren diyagram.

İlk olarak, görüntü ızgara hücrelerine bölünür. Her ızgara hücresi B sınırlayıcı kutuları tahmin eder ve güven puanlarını sağlar. Hücreler, her nesnenin sınıfını oluşturmak için sınıf olasılıklarını tahmin eder. Örneğin, en az iki nesne sınıfı görebiliriz: polip ve normal doku. Tüm tahminler, tek bir evrişimsel sinir ağı kullanılarak aynı anda yapılır. Birleşim üzerinden kesişme, öngörülen sınırlayıcı kutuların nesnelerin gerçek kutularına eşit olmasını sağlar. Bu fonksiyon, nesnelerin özelliklerini (yükseklik ve genişlik gibi) karşılamayan gereksiz sınırlayıcı kutuları ortadan kaldırır. Nihai algılama, nesnelere mükemmel şekilde uyan benzersiz sınırlayıcı kutulardan oluşacaktır.

YOLOv1: İlk YOLO modeli Joseph Redmon ve diğerleri tarafından 2015 yılında yayınlanan “Yalnızca Bir Kez Bakarsınız: Birleşik, Gerçek Zamanlı Nesne Algılama” başlıklı makalesinde tanıtıldı. O zamana kadar R-CNN [75] tabanlı modeller, nesne tespiti için en çok aranan modellerdi. Her ne kadar RCNN model ailesi doğru olsa da sınırlayıcı kutu için önerilen bölgeyi bulmanın çok adımlı bir süreci olduğu için nispeten yavaştı ve daha sonra bu bölgeler üzerinde sınıflandırma yaptı ve son olarak çıktıyı iyileştirmek için son işleme yaptı. Orijinal YOLO ağı, omurga ağı olarak değiştirilmiş bir GoogLeNet [76] kullanır. Redmond daha sonra, 3×3 filtrelerin genel tasarımını takip eden ve her havuzlama adımında kanal sayısını ikiye katlayan DarkNet-19 adlı yeni bir model sundu. 1×1 filtreler ayrıca ağ boyunca özellik gösterimini periyodik olarak sıkıştırmak için kullanılır. YOLO, algılama görevine uyarlanmadan önce görüntü sınıflandırıcıları olarak önceden eğitilir. YOLO $N \times N \times B$ sınırlayıcı kutu tahminlerini üretmek için $B(5+C)$ filtreleri ile bir evrişim katmanını eklemektedir YOLO modelinin ilk yinelemesi, bir sınırlayıcı kutuyu tanımlayan dört değeri de doğrudan tahmin eder. Her bir sınırlayıcı kutunun x ve y koordinatları, her grid hücresinin sol üst köşesine göre tanımlanır ve koordinat değerleri 0 ile 1 arasında sınırlanacak şekilde hücre boyutlarına göre normalleştirilir. Redmond bu formülasyonu seçti çünkü "büyük kutulardaki küçük sapmalar küçük kutulardakinden daha az önemlidir. Sınırlayıcı kutu genişliği ve yüksekliği şu şekilde normalleştirilir: görüntü genişliği ve yüksekliği ve dolayısıyla 0 ile 1 arasında sınırlandırılmıştır. Eğitim sırasında bir L2 kaybı uygulanır.

YOLOv1 sürümünde, "nesne olup olmadığı" puanı p_{obj} , tahmin edilen kutu ile kesinlik etiketi arasındaki Birleşim Üzerinden Kesişmeyi (IoU) yaklaşık olarak tahmin etmek için eğitilir. Eğitim sırasındaki kayıp hesaplandığında, nesneleri (aynı ızgara hücresinde) en yüksek IoU puanına sahip olan sınırlayıcı kutu tahminiyle eşleştirilir. Eşleşmeyen kutular için, kayıp fonksiyonuna dâhil edeceğimiz tek tanımlayıcı p_{obj} 'dir. YOLOv1'de, sınıf tahmini, ızgara hücresi düzeyinde gerçekleştirilir. YOLOv1 modeli, omurga ağının çıktısını kullanarak $N \times N \times B$ sınırlayıcı kutuları basitçe tahmin eder. YOLOv1 modeline ait diyagram Şekil 5.2'de gösterilmiştir.

YOLOv2: YOLOv2, 2016 yılında Joseph Redmon ve Ali Farhadi tarafından “YOLO9000: Better, Faster, Stronger” başlıklı makalelerinde yayınlandı. 9000, YOLOv2'nin 9000'den fazla nesne kategorisini algılayabildiğini gösteriyordu. Bu sürüm, önceki YOLOv1 sürümüne göre çeşitli iyileştirmelere sahipti. YOLOv2,

YOLOv1'in yerine geçti. Bu modelde daha hızlı ve daha iyi hale getirmek için çeşitli iyileştirmeler yapılmıştır. 19 evrişim katmanı ve 5 maksimum havuzlama katmanından oluşan Darknet-19'a dayanmaktadır ve YOLOv1'in tüm tam evrişimli katmanları kaldırılmıştır. YOLO v2, sınıflandırılmış algılama verilerine sahip olmayan nesne sınıfları için algılamaları tahmin etmek için COCO veri kümesi ve ImageNet veri kümesinin birleşimi üzerinde eğitilmiştir. Daha basit, daha hızlı ve daha güçlü hale getirmek için çeşitli geliştirmeler yapılmıştır. mAP değeri %2'den fazla iyileşmeye yol açan toplu normalizasyon tanıtıldı. Normalizasyon işlemi ile model düzenlenir ve fazla ağırlığı örnekten düşmenin silinmesi gerçekleşebilir. Çözünürlüğü 448x448'e yükselten yüksek çözünürlüklü sınıflandırıcı eklenmiştir. Yüksek Çözünürlük sınıflandırması nedeniyle mAP'de %4'ten fazla iyileştirme vardır. Çok sınıflı tanımlama için Çapa Kutuları (Anchor) eklendi, fakat az da olsa çapa kutularının kullanımı nedeniyle doğrulukta azalma var. YOLOv1 her resim için 98 kutu öngörürken, YOLOv2 modeli bağlantı kutuları ile 1.000'den fazla kutu öngörür. Boyut kümeleme, yapay nesneler için ızgaraları kümelemek ve sınırlayıcı kutular oluşturmak için kümeleme kullanılarak elde edilir. İnce taneli yüzeyler, hareket ettirme adı verilen yenilenmiş bir teknik kullanarak katmanları yeniden şekillendirerek küçük artefaktları ayırt etmeye yardımcı olur. İnce girenleme sayesinde görüntü, küçük nesnelerin de gözlemlenebileceği 13x13'lük ızgaralara ayrılır. Bu hassasiyet, hız ile birlikte gerçekleştirilir. Daha fazla nesne tanımlamasını sağlamak için hiyerarşik bir mimari eklendi. Bu doğrulukla hız ile birlikte elde edilir. Bu mimaride, her kutuyla ilişkili güven puanı, kutunun bir nesnesi olup olmadığını ve varsa nesne sınıfının belirlenmesine yardımcı olur. YOLOv2, VOC 2012 [77] veri kümesinde 78.6 mAP kaydetti. YOLOv2 sürümü bağlantı kutuları kavramını tanıttı. Bağlantı kutuları, algılanacak nesnelerin idealleştirilmiş konumunu gösteren bir görüntü için önceden tanımlanmış alanlardan başka bir şey değildir. Öngörülen sınırlayıcı kutunun ve önceden tanımlanmış bağlantı kutusunun birleşme üzerindeki örtüşme oranını (IoU) hesaplarız. IoU değeri, tespit edilen nesnenin olasılığının bir tahmin yapmak için yeterli olup olmadığına karar vermek için bir eşik görevi görür. Farklı en boy oranlarına uyum sağlamak için YOLOv2 modeli eğitim süreci boyunca rastgele yeniden boyutlandırılır (buna çok ölçekli eğitim denir). Modelin sağlam olması için YOLOv2 modeli, COCO [78] veri kümesi (sınırlayıcı kutularla birlikte 80 sınıf) ve ImageNet veri kümesinin (sınırlayıcı kutular olmadan 22k sınıf) bir kombinasyonu üzerinde eğitilmiştir. Model, etiketli bir görüntüyü işlediğinde algılama

ve sınıflandırma hatası hesaplanır. Model etiketsiz bir görüntü gördüğünde, yalnızca sınıflandırma hatasını geri yayar. Darknet19 (YOLO'nun omurgası) adı verilen bir sınıflandırma ağı mimarisi kullanılarak 200 FPS'ye kadar çıkarım hızları ve 75.3 mAP'ye ulaşıldı. YOLOv2 mimarisin kullandığı DarkNet19 yapısı aşağıdaki Şekil 2.19'da gösterilmiştir.

Type	Filters	Size/Stride	Output
Convolutional	32	3×3	224×224
Maxpool		$2 \times 2/2$	112×112
Convolutional	64	3×3	112×112
Maxpool		$2 \times 2/2$	56×56
Convolutional	128	3×3	56×56
Convolutional	64	1×1	56×56
Convolutional	128	3×3	56×56
Maxpool		$2 \times 2/2$	28×28
Convolutional	256	3×3	28×28
Convolutional	128	1×1	28×28
Convolutional	256	3×3	28×28
Maxpool		$2 \times 2/2$	14×14
Convolutional	512	3×3	14×14
Convolutional	256	1×1	14×14
Convolutional	512	3×3	14×14
Convolutional	256	1×1	14×14
Convolutional	512	3×3	14×14
Maxpool		$2 \times 2/2$	7×7
Convolutional	1024	3×3	7×7
Convolutional	512	1×1	7×7
Convolutional	1024	3×3	7×7
Convolutional	512	1×1	7×7
Convolutional	1024	3×3	7×7
Convolutional	1000	1×1	7×7
Avgpool		Global	1000
Softmax			

Şekil 2.19. YOLOv2 mimarisinin yapısı.

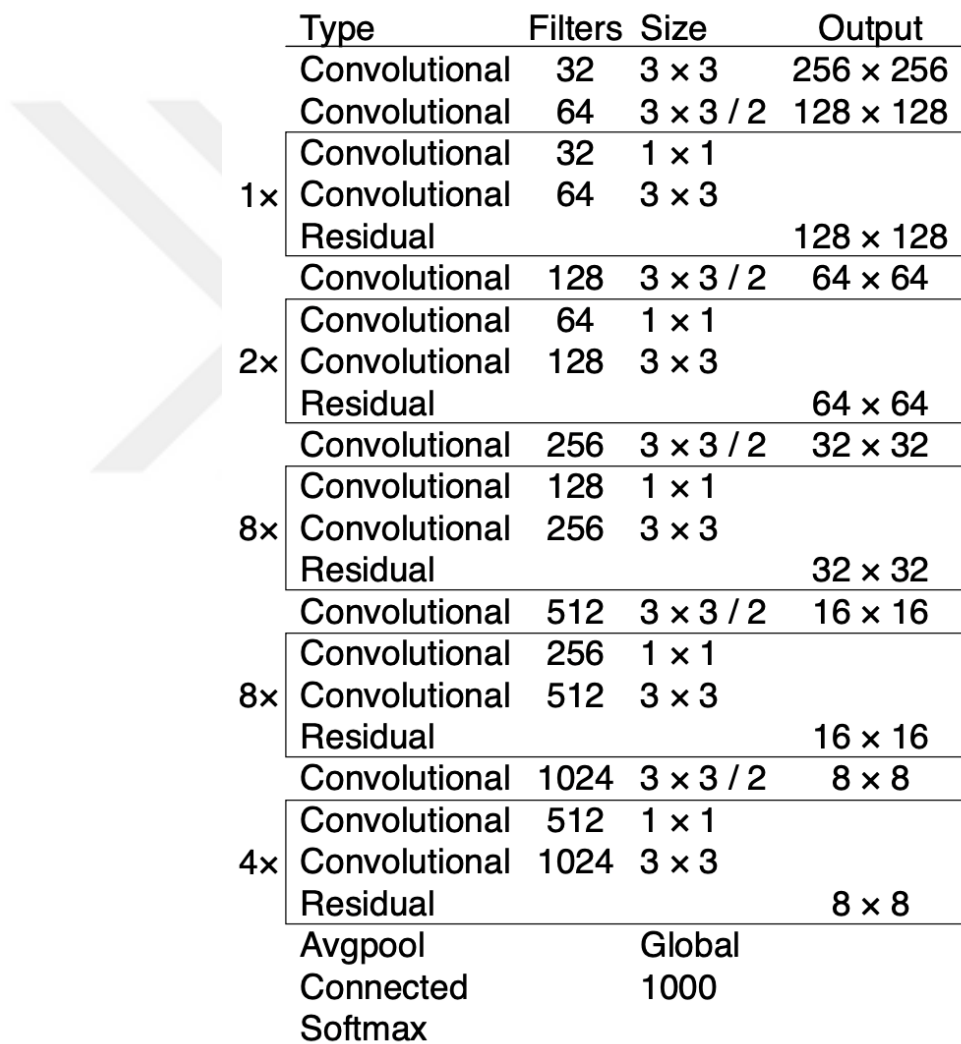
YOLOv3: 2018'de Joseph Redmon ve Ali Farhadi, "YOLOv3: Artan İyileştirme" adlı makalelerinde YOLO'nün üçüncü sürümünü tanıttı. Bu YOLO sürümü Redmon tarafında önerilen son YOLO modeli olmakla da biliniyor. Bu model öncekilerden biraz daha büyüktü ama daha doğrudu ve yine de yeterince hızlıydı. YOLOv3, 22 milisaniyelik bir çıkarım süresi ile 28.2'lik bir mAP'ye sahiptir. (COCO veri setinde).

Bu, SSD nesne algılama tekniğinden 3 kat daha hızlı, ancak benzer bir doğruluğa sahiptir.

YOLOv3, model boyutunu ve ağırlığını büyük ölçüde azaltan tam bağlantılı veya havuzlama katmanları kullanmadan 75 evrişim katmanından oluşuyordu. YOLOv3, minimum çıkarım sürelerini korurken özellik piramit ağı (FPN) ile çoklu özellik öğrenimi için artık modelleri (ResNet modelinden) kullanır. Özellik piramidi ağı, tek bir görüntü için farklı boyutta özellik çıkaran bir özellik çıkarıcıdır. Modelin yerel ve genel özellikleri öğrenebilmesi için tüm özellikleri birleştirir. YOLOv3 modeli omurga olarak DarkNet53 mimarisini kullanır. Lojistik sınıflandırıcıların ve aktivasyonların kullanılmasıyla, YOLOv3'te kullanılan DarkNet53 mimarisi, sınıf tahminleri doğruluk açısından ResNet-50 ve 101'in ötesine geçer.

YOLOv3, YOLOv2'nin geliştirilmiş bir şeklidir. YOLOv3, lojistik regresyon aracılığıyla her sınırlayıcı kutu için nesnellik değerini hesaplar. YOLOv3, maliyet fonksiyonunun ölçülme şeklini değiştirir. Sınırlayıcı kutu çapaları, diğerleri yerine zemin gerçeklik nesnesini geçmedikçe, elde edilen nesne değeri 1 olacaktır. Ön tanımlı eşikten (varsayılan 0,5) daha büyük örtüşen diğer çapalar herhangi bir maliyete tabi değildir. Her zemin gerçeği nesnesi, yalnızca önceden bir sınır kutusuyla ilişkilendirilir. Bir sınırlayıcı kutu bağlantısı atanmamışsa, sınıflandırma ve yerelleştirme kaybı olmaz, yalnızca nesnellik konusunda güven kaybı olur. Kaybı hesaplamak için toplam kayıp kullanılır. YOLOv3, diğer YOLO tekniklerden daha iyi bir sınıf tahminine sahiptir. YOLOv3 çok etiketli bir sınıflandırıcı olduğundan, softmax işlevi, girdi bölgesinin belirli bir sınıfa ait olma olasılığını ölçen farklı lojistik sınıflandırıcılar ile değiştirilir. Sınıflandırma kaybının tahmininde ortalama kare hatası kullanmak yerine, her sayı için sigmoid çapraz entropi kaybı kullanılır. İstatistiksel lojistik sınıflandırıcıların kullanılması da istatistiksel karmaşıklığı azaltır. İçinde birkaç nesnenin bulunduğu bu veri kümesinde, softmax işlevi, genellikle YOLOv3'te durum böyle değildir, bir seferde tek bir nesne için bir sınıfı algılar. YOLOv3'teki çok etiketli bir yaklaşım, her konum için daha iyi bir tahmin doğruluğu sağlayan 3 tahmin verir. Her tahmin çıktısı, nesnellikten, bir sınır kutusundan ve bir $N \times N$ ızgarasındaki 80 sınıf puanından, yani $N \times N / [3 \times (4 + 1 + 80)]$ tahminlerden oluşur. YOLOv3 daha iyi bir özellik çıkarıcıya sahiptir. Bu modelde, eski Darknet-19'un yerine yeni bir özellik çıkarıcı olan 53 katmanlı DarkNet-53 mimarisi uygulanmıştır. DarkNet53'te 3x3 ve 1x1 filtreler ile

alternatif bağlantılar gibi artık bir ağ oluşturulur. Bu artık ağ yapısı, GPU'nun daha verimli kullanılmasına yardımcı olur, değerlendirmeyi daha sağlam hale getirir ve çok daha hızlı hale getirir. Doğruluktaki büyük gelişme, artık bağlı ağlardan kaynaklanmaktadır denilebilir. YOLOv3, YOLOv2'ye göre, hata fonksiyonundaki belirli değişikliklerin dâhil edilmesi ve küçük ile önemli nesnelerin üç ölçekli tespiti gibi avantajlara sahiptir. YOLOv3 nesne tespit algoritmasına ait DarkNet53 mimarisine ait diyagram Şekil 2.20'de gösterilmiştir.



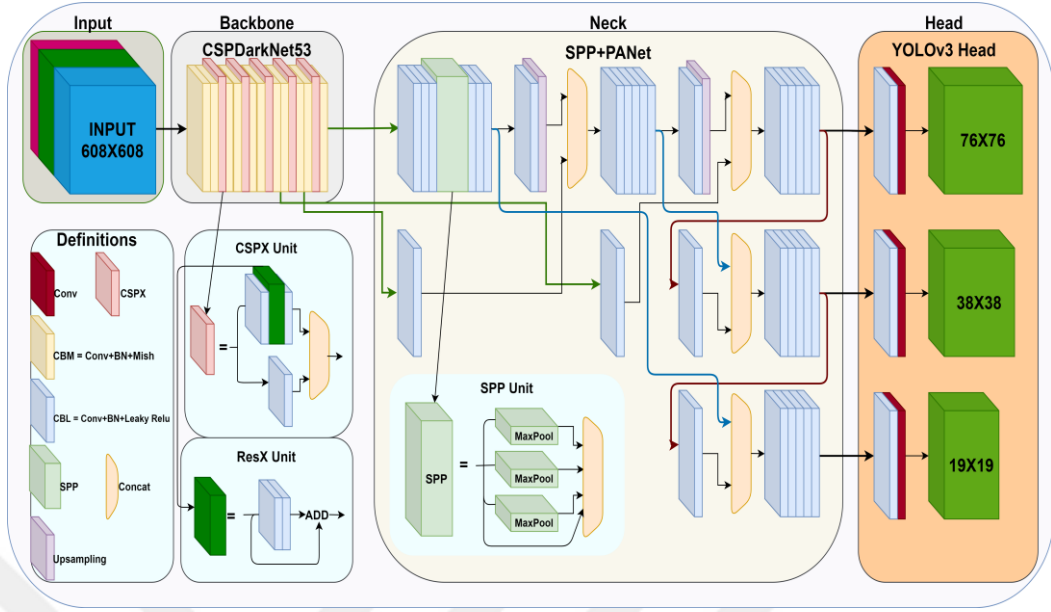
Type	Filters	Size	Output
Convolutional	32	3 × 3	256 × 256
Convolutional	64	3 × 3 / 2	128 × 128
1× Convolutional	32	1 × 1	
1× Convolutional	64	3 × 3	
Residual			128 × 128
Convolutional	128	3 × 3 / 2	64 × 64
2× Convolutional	64	1 × 1	
2× Convolutional	128	3 × 3	
Residual			64 × 64
Convolutional	256	3 × 3 / 2	32 × 32
8× Convolutional	128	1 × 1	
8× Convolutional	256	3 × 3	
Residual			32 × 32
Convolutional	512	3 × 3 / 2	16 × 16
8× Convolutional	256	1 × 1	
8× Convolutional	512	3 × 3	
Residual			16 × 16
Convolutional	1024	3 × 3 / 2	8 × 8
4× Convolutional	512	1 × 1	
4× Convolutional	1024	3 × 3	
Residual			8 × 8
Avgpool		Global	
Connected		1000	
Softmax			

Şekil 2.20. DarkNet53 mimarisinin yapısı.

YOLOv4 ve Diğer Sürümler: Alexey ve arkadaşları tarafından önerilen YOLOv4 [72], son teknoloji bir nesne algılama algoritmasıdır. YOLOv4, artırılmış nesne algılama hızı, optimize edilmiş paralel hesaplama ve çoklu etiket sınıflandırmasıyla

diğer algoritmalarından sıyrılıyor. Bir nesne algılama algoritması üç yapıdan oluşur: omurga, boyun ve kafalar. Omurga bloğunun ana işlevi, özellik çıkarma işlemidir. Omurga seçimi, nesne algılama algoritmasının performansını artırmak için en önemli adımlardan biridir. Boyun bloğunun amacı, farklı aşamalardan özellik haritalarının birleştirilebilmesi için omurga ve kafa arasına ekstra katmanlar eklemektir. Genellikle, bir boyun birkaç aşağıdan yukarıya yoldan ve birkaç yukarıdan aşağıya yoldan oluşur. Son aşama, tek aşamalı yaklaşımlardaki baş blok, son tahmini gerçekleştirir. Bu tahmin, tahmin edilen sınırlayıcı kutunun koordinatlarını, güven puanını ve tahminin etiketini içeren bir vektörden oluşur.

YOLOv4 algoritması, özellik çıkarımı için CSPNet [79] ve DarkNet53'ü birleştiren CSPDarkNet53 omurgasını kullanır. DarkNet53 mimarisi YOLOv3'te ilk kez kullanılmıştır ve ResNet'in gradyan sorununu hafifletmesinden yararlanıyor. CSPNet ise Wang ve diğerleri tarafından önerilen ve hesaplama ile bellek maliyetlerini azaltırken CNN'nin öğrenme yeteneğini geliştiren bir mimaridir [79]. CSPDarkNet53, 29 evrişim katmanı (3×3), daha fazla parametre (27.6 M) ve daha büyük alıcı alan boyutu (725×725) ile diğer mimarilerden üstündür. YOLOv4'ün temel omurga yapısı, YOLOv3'te yaygın olarak kullanılan ve bir evrişim katmanı, toplu normalleştirme ve Leaky Relu'dan (L) oluşan CBL (Conv + BN + Leaky ReLU) modülüdür. CBM modülü (Conv + BN + Mish), evrişim, toplu normalleştirme (BN) ve Mish etkinleştirme işlevlerinden oluşan Aşamalar Arası kısmi bağlantılar (CSP) modülüyle birlikte kullanılır. YOLOv4'te beş CSP (CSP1, CPS2, CSP8, CSP8, CSP4) birimi vardır. CSP1 bir artık birimi (Residual) içerirken CSP8 8 Residual birimi içerir. Mish [28] aktivasyon fonksiyonunun sağladığı düzgünlük, ağır genelleme kabiliyetini ve sınıflandırma doğruluğunu artırır. YOLO'da ağır derinleşmesini sağlayan artık birimler kullanılır ve bu birimler CBL modülü ve tensörlerin örtüşmesini gerçekleştirmeyi amaçlar. Concat katmanları, tensörün karakteristik eklemesini gerçekleştirir. Şekil 2.21, YOLOv4 mimarisinin temelini oluşturan yapıları detaylandırmaktadır.

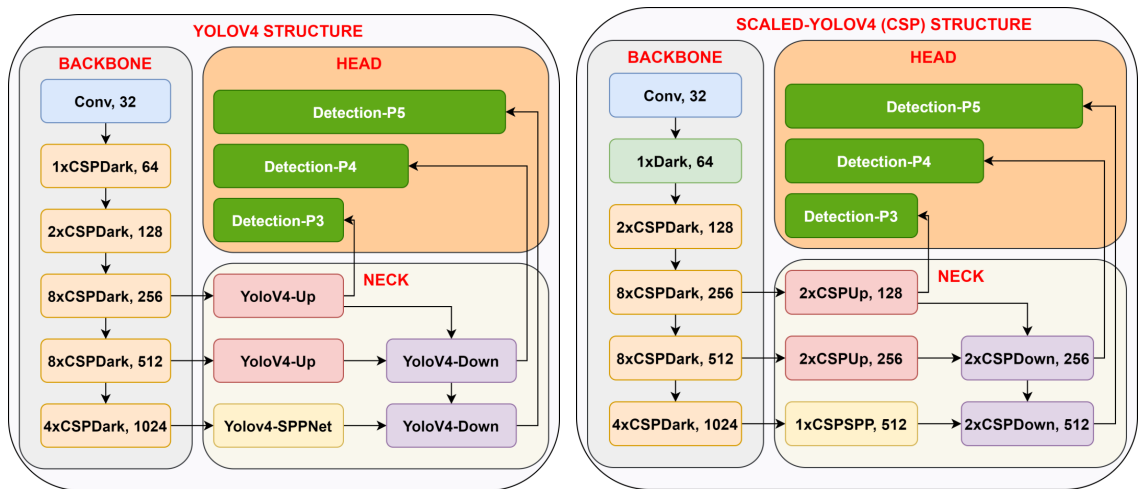


Şekil 2.21. YOLOv4 mimarisinin yapısı.

YOLOv4'teki Uzamsal Piramit Havuzlaması (SPP), YOLO algoritmasındaki uzamsal boyutu korumak için kullanılır. SPP'de kullanılan havuzlama blokları 1×1 , 5×5 , 9×9 ve 13×13 gibi yüzen çekirdeklere uygulanır. Daha sonra farklı boyuttaki çekirdeklerden elde edilen özellikler birleştirilir. Ek olarak, SPP blokları artan bir alıcı alan sağlar, ancak ağ çalışma hızında neredeyse hiç azalma olmaz. YOLOv4, YOLOv3'te kullanılan özellik piramit ağlarının (FPN) yerini almak için değiştirilmiş yol toplama ağlarını (PANet) kullanır. PANet, daha semantik ve zenginleştirilmiş özellikler sağlamak için daha düşük seviyeli özellik haritaları ile daha yüksek seviyeli özellik haritalarını birleştirir. Böylece farklı omurga seviyelerinden parametre bilgileri toplanır ve daha yüksek doğruluk elde edilir. YOLOv4'te kullanılan diğer bir teknik, değiştirilmiş Uzamsal Dikkat Modülüdür (SAM). Bu modül, girdinin belirli bir bölümüne odaklanır. Değiştirilen SAM modülünde maksimum havuzlama ve ortalama havuzlama katmanları kaldırılmıştır. YOLOv4, en son teknolojiler olan Bag-of-Free (BoF) ve Bag-of-Specials (BoS) yöntemleriyle doğruluk ve performansta iyileştirmeler sağlar. BoF yöntemi, nesne tespiti aşamasının çıkarım hızını koruyarak daha iyi doğruluk elde etmek için geliştirilmiş farklı eğitim tekniklerinin genel adıdır. Öte yandan BoS, nesne tespiti aşamasının doğruluğunu büyük ölçüde artırırken çıkarım maliyetini çok az artıran son işlem modülleridir. Özetlemek gerekirse: BoF, mozaik veri büyütme, CutMix, DropBlock düzenleme ve sınıf etiketi yumuşatma ile omurgayı geliştirir. BoF ayrıca nesne tespiti aşamasının CIOU-loss, eğitim görüntülerinin

gruplarını normalleştirmek için Çapraz mini-Yığın Normalleştirme (CmBN), aşırı ezberleme sorunlarını önlemek için bir bırakma birimi olarak DropBlock düzenleme, mozaik veri büyütme, Kendine Karşı Eğitim (SAT) ile geliştirir ve ızgara hassasiyetini ortadan kaldırır, tek bir temel gerçeği için birden fazla çapa sağlar ve optimal hiperparametrelere ve rastgele eğitim şekillerine sahiptir. BoS tarafından omurga üzerinde yapılan iyileştirmeler, Mish aktivasyonu, CSP ve Çok girişli ağırlıklı artık bağlantılarıdır (MiWRC). BoS tarafından detektör için yapılan iyileştirmeler, Mish aktivasyonu, SPP bloğu, SAM bloğu, PANet ve Maksimum Olmayan Bastırma (NMS) ile DIOU'dur.

YOLOv4'ün geliştirilmesinden sonra Wang ve ark. [80] , farklı uygulamalar için ölçeklendirmeyi etkinleştirmek için YOLOv4'ün yapısını değiştirdi. YOLOv4-tiny, hızı en üst düzeye çıkarmak ve mümkün olan minimum hesaplama maliyetini elde etmek için tasarlanmıştır. Ardından, değişen hesaplama gereksinimleriyle doğruluğu en üst düzeye çıkarmak için YOLOv4-CSP ve YOLOv4'ün diğer daha büyük sürümleri geliştirildi. YOLOv5 [81] , YOLOv3'ün PyTorch sürümünü geliştiren ve Haziran 2020'de piyasaya sürülen aynı şirket olan Ultralytics tarafından geliştirilen en son nesne algılama modellerinden biridir ve temeli YOLOv3'e dayanmaktadır. YOLOv5'in YOLOv5s, YOLOv5m, YOLOv5l ve YOLOv5x olmak üzere dörtten fazla farklı modelleri vardır. Genel olarak, YOLOv5 sırasıyla omurga olarak bir SPP katmanı, Boyun olarak PANet ve YOLO algılama başlığı ile CSPDarknet53 mimarisini kullanır. Aşağıdaki Şekil 2.22'de solda YOLOv4 yer alırken, sağdaki şekilde ise YOLOv4-CSP yani ölçeklenebilir YOLOv4 yer almaktadır.

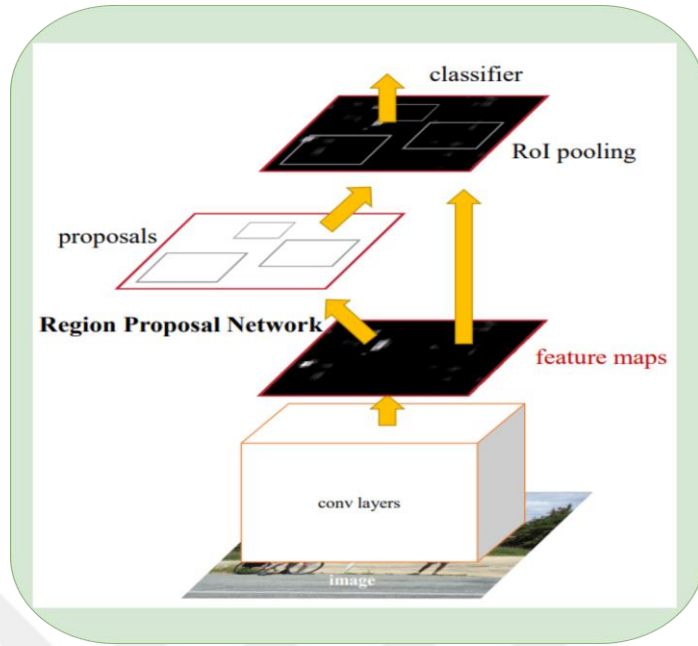


Şekil 2.22. YOLOv4 (solda) ve YOLOv4-CSP (sağda) mimarilerinin yapısı.

2.4.2. İki Aşamalı Yaklaşımlar

İki aşamalı yaklaşımlar, algılama sürecini iki ana aşamaya böler. İlk aşamadan, bölge önerisi yani teklifler oluşturulur. Sınırlayıcı kutular kullanılarak teklif bölgeleri oluşturulur. Diğer aşamada ise sınıflandırma aşaması yani sınıf tahminleri yapılır, bu da yerelleştirmeyi sağlar. Başka bir deyişle, temel olarak, görüntüdeki tüm nesnelerin kendilerini önerilen bölgelerden birine kategorize edecek şekilde daha büyük geri çağırma değerlerine sahip bölgeleri bulmaktır, bu ilk aşamada yapılır. İkinci aşamada ise, bu oluşturulan önerileri ya önceden tanımlanmış sınıf kategorilerinden bir nesneye ya da basitçe arka plana kategorize etmek için evrişimli sinir ağı tabanlı modeller kullanılır. Bazı popüler iki aşamalı yaklaşımlar ele alınacaktır ve detaylandırılacaktır.

Faster R-CNN (Daha Hızlı R-CNN): Ren ve ark. tarafından 2015 yılında Fast R-CNN'nin [82] piyasaya sürülmesinden kısa bir süre önerilmiştir. Önceki muadilleri olan R-CNN ve Fast R-CNN'de ortaya çıkan problemlere çözüm getirerek daha optimize ve başarılı bir algoritma sunmuştur. Daha önce Fast RCNN'de bölge önerisi için sistemi yavaşlatan seçici bir arama yöntemi kullanılmaktadır. Daha hızlı RCNN, bu yöntemi tam bağlantılı bir CNN olan yeni bir bölge teklif ağı (RPN) ile değiştirir. RPN, çok çeşitli en boy oranları ve ölçeklerde nesne bölgelerini daha verimli bir şekilde tahmin etme kabiliyetine sahiptir. Faster RCNN'de, bölge teklifinin oluşturulması için gereken süre Fast RCNN'ye kıyasla daha kısadır ve başarımları ise daha iyidir. Daha hızlı R-CNN, ilgi bölgelerini, yani teklif bölgelerini oluşturmak için kullanılan bölge teklif ağı ve nesneyi tespit etmek için bu teklifleri kullanan ağ olmak üzere iki ağdan oluşur. Sonuçta ise, bölge önerisi oluşturma görevi ve nesne tespiti görevi benzer ağ tarafından gerçekleştirilir. Faster RCNN'de üç farklı ölçek ve üç farklı en boy oranına sahip bağlantı kutuları kullanılır. Bağlantı kutusu tabanlı bölge önerisi genellikle sınırlayıcı kutuyu tahmin etmek için görevlendirilir. Daha sonra, tahmin edilen kutunun konumunu optimize etmek için gerçek doğruluk kutusu ile tahmin edilen sınırlayıcı kutu arasındaki mesafe hesaplanır. Bu nedenle, Faster R-CNN, en doğru nesne algılama tekniklerinden biri olarak kabul edilir. Faster R-CNN ağında ait yapı Şekil 2.23'te gösterilmiştir.

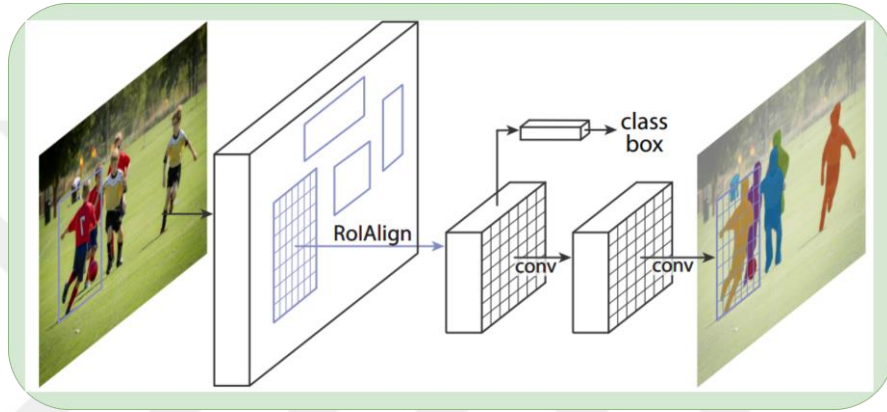


Şekil 2.23. Faster R-CNN mimarisinin yapısı.

Faster R-CNN nesne algılama ağı diğer nesne tespit algoritmaları gibi tipik olarak önceden eğitilmiş bir CNN omurgası olan bir özellik çıkarma ağından oluşur. Bunu daha sonra eğitilebilir iki ana alt ağ takip eder. Birinci ağ adından da anlaşılacağı gibi, nesne teklifleri oluşturmak için kullanılan bir bölge teklif ağı (RPN) iken, ikinci ağ ise nesnenin gerçek sınıfını tahmin etmek için kullanılan yapıdır. Dolayısıyla Faster R-CNN için birincil farklılaştırıcı son evrişim katmanından sonra eklenen RPN'dir denilebilir. Bu, daha önceki sürümlerinde var olan seçici arama gibi herhangi bir harici mekanizmaya ihtiyaç duymadan doğrudan bölge önerileri üretmek üzere eğitilmiştir. Bundan sonra, ROI havuzu ve Fast R-CNN'ye benzer bir yukarı akış sınıflandırıcı ve sınırlayıcı kutu regresörü kullanılarak nesne tespiti gerçekleştirilmiş olur.

Mask R-CNN: Maske R-CNN He ve ark. [83] tarafından Daha Hızlı R-CNN'nin bir uzantısı olarak, temeldeki önerilere paralel olarak segmentasyon (bölümleme) için kullanılan sınırlayıcı kutu regresyonuna paralel olarak RoI'lerin tahmin segmentasyonu için bir Maske ağı dalı ekleyen gelişmiş bir nesne tespit ve segmentasyon algoritmasıdır. Mask R-CNN nesne algılama ve örnek segmentasyonunu aynı anda tamamlayabilme özelliğine sahip bir algoritmadır. Mask R-CNN ağı, her bir RoI için bölünmüş maske oluşturmak için kullanılan tamamen evrişimli ağın aerodinamik bir sürümüdür. RoI havuzunun tamsayı nicelemesi nedeniyle, özellik haritası bölgesi ve

orijinal görüntü bölgesi hizalı değildir. Bu nedenle, piksel düzeyinde maskeyi doğru bir şekilde tahmin etmek için bir bias üretir. Mask R-CNN havuzlama katmanını yerine, piksel düzeyinde hizalama elde etmek için çift doğrusal enterpolasyon kullanan bir RoI-Align katmanını içerir. Bu katman, nesne algılama dalının doğruluğunu da iyileştirebilme kapasitesine sahiptir. Mask R-CNN nesne tespitinde ve nesne bölümlenme problemlerinde popüler olarak kullanılan gelişmiş bir algoritmadır. Mask R-CNN ağında ait yapı Şekil 2.24'te gösterilmiştir.



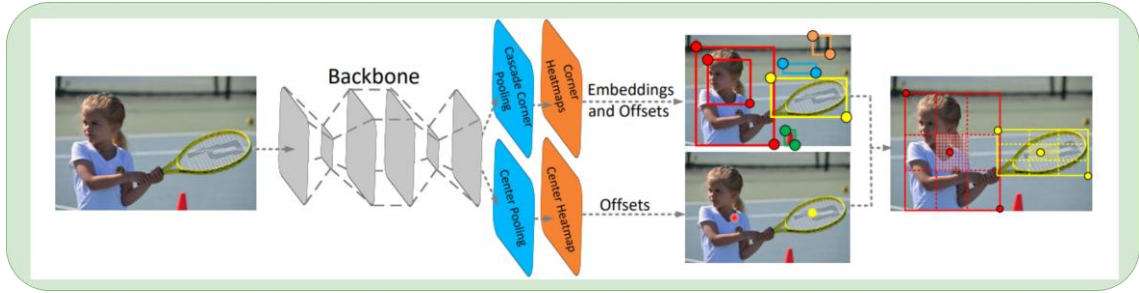
Şekil 2.24. Mask R-CNN mimarisinin yapısı.

2.4.3. Diğer Yaklaşımlar

Derin öğrenmenin hızlı bir şekilde gelişimi ile beraber, nesne algılama, geleneksel algoritmalarla evrimsel sinir ağları (CNN) ile bir üst aşamaya geçmiştir. Derin öğrenmeye dayalı nesne algılama, geleneksel algılamanın darboğazını aşarak algılamanın doğruluğunu ve verimliliğini önemli ölçüde artırmıştır. Farklı eğitim yöntemlerine göre, Anchor-Based (Çapa Tabanlı) modül iki tür modüle ayrılabilir: Bölge önerilerine dayalı iki aşamalı detektörler ve regresyon tabanlı Tek Aşamalı detektörler önceki bölümlerde anlatıldı. Genel olarak, iki aşamalı algılama yönteminin iki adımı vardır: ilk olarak, bölge önerileri belirli bir algoritma tarafından çıkarılır ve ardından sınırlayıcı kutunun konumunu sınıflandırmak ve iyileştirmek için CNN kullanılır. Bu yapıya örnek olarak ise; R-CNN, SPP-Net, Fast R-CNN, Faster R-CNN gibi örnekler verilebilir. Diğer taraftan tek aşamalı yaklaşımların ise bölge önerileri oluşturmaya gerek yoktur ve konumu doğrudan sınırlayıcı kutunun regresyonuna dönüştürür, tipik yöntemler olarak ise YOLO ve SSD gibi yaklaşımlar örneklendirilebilir.

Bu anlatılan yaklaşımlardan farklı olarak, özellikle son zamanlarda, Anchor-free (çapa içermeyen) yapılar ortaya çıkmıştır. Bu yaklaşımların çıkmasının en temel sebepleri ise çapanın önceki algoritmalarda (tek ve iki aşamalı) önceden ayarlanma ve probleme özgü olarak optimize edilmesinin kusurlarında kaynaklıdır. Çünkü çapa, farklı veri kümeleri için farklı gereksinimlere sahip olan, çapanın önceden tanımlanmış ölçeği ve en boy oranı gibi belirli kuralları içermesidir. Çünkü çapa otomatik olarak ayarlanamaz (YOLOv4 ile beraber genetik algoritma ile otomatik yapılması hariç), bu da aşırı ölçekli nesneleri eşleştirmeyi zorlaştırır. Çapanın çok fazla ön ayarı, eğitim sırasında pozitif ve negatif örnekler arasında dengesizliğe yatkın hale getirmesi, Anchor-free nesne tespit algoritmalarının da popüler olmasını sağlamıştır. Bazı popüler çapasız nesne tespit algoritmalarına örnek olarak CenterNet [84], CornerNet [85] ve FCOS [86] verilebilir. Bu bölümde popüler çapa içermeyen nesne tespit algoritmalarından bir tanesinden bahsedilecektir. Buradaki amaç ise genel yapı hakkında bilgi edinmektir.

CenterNet: CenterNet, çapasız bir nesne algılama mimarisidir. Bu yapının önemli bir avantajı ise klasik NMS'yi (Non Maximum Suppression) kullanmadan daha hızlı işlem yapar. Aslında CenterNet, CornerNet'in devamı niteliğindedir. CornerNet, çapa tabanlı yöntemleri kullanmanın sakıncalarının üstesinden gelmek için bir çift köşe anahtar noktası kullanır. Bununla birlikte, CornerNet'in performansı, nesnenin genel bilgisine göndermede zayıf bir yeteneğe sahip olduğundan, nesnelerin sınırlarını tespit etmede sınırlıdır. CenterNet, CornerNet'te karşılaşılan kısıtlamaları aşmaya çalışmak için geliştirilmiştir. Adından da anlaşılacağı gibi ağ, önerilen her bölgedeki görsel kalıpları algılamak için ek bilgiler (merkezlilik bilgisi) kullanır. Artık iki köşe bilgisini kullanmak yerine, nesneleri yerelleştirmek için üçlülere kullanır. Çalışma, tahmin edilen bir sınırlayıcı kutunun, yer gerçeği kutusuyla yüksek bir IoU'ya sahip olması durumunda, merkez bölgesindeki merkez anahtar noktasının aynı sınıf olarak tahmin edilme olasılığının yüksek olduğunu ve bunun tersi olduğunu belirtir. Çıkarım sırasında, teklifler olarak köşe noktaları verildiğinde, ağ, aynı sınıfın kendi merkez bölgesi içinde kalan bir merkez anahtar noktası olup olmadığını kontrol ederek köşe teklifinin gerçekten bir nesne olup olmadığını doğrular. Nesne merkezliliğin ek kullanımı, ağı tek aşamalı detektör olarak tutar, ancak iki aşamalı detektörlerde kullanıldığı gibi RoI yoklamanın işlevselliğini devralır. CenterNet algoritmasının temel yapısı Şekil 2.25'te gösterilmiştir.

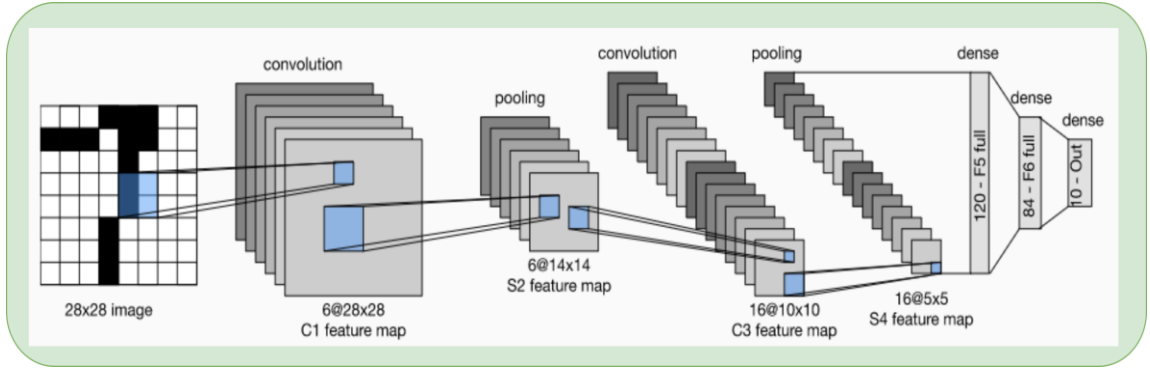


Şekil 2.25. CenterNet mimarisinin yapısı.

2.5. CNN Omurga Ağları

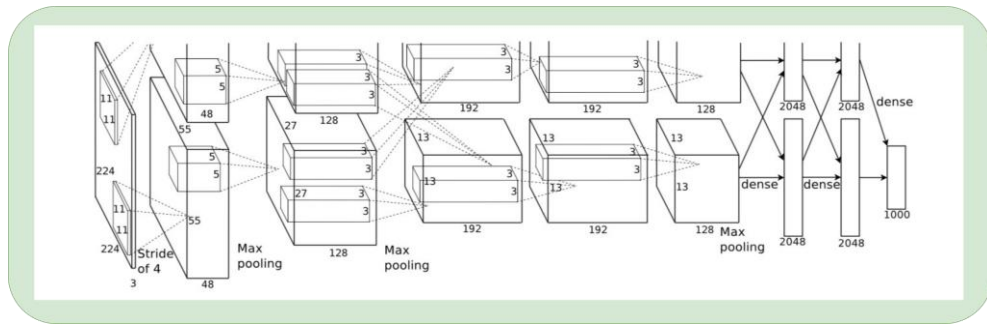
Bu bölümde temel CNN mimarilerinden bahsedilecektir. Özellikle, nesne tespit algoritmalarında özellik çıkarma işlemini gerçekleştiren CNN omurgaları nesne tespiti yaklaşımlarında en önemli yeri tutmaktadır. İlk başlarda sınıflandırma işlemleri için kullanılan CNN mimarileri daha sonralarında nesne tespit algoritmalarında ise en önemli görev olan özellik çıkarma işlemlerine odaklanılmışlardır. Popüler ve temel bazı CNN mimarileri temel halleriyle anlatılacaktır.

LeNet: Bu mimari MNIST el yazısı rakam tanıyıcı modellerine başarıyla uygulanan geleneksel basit bir CNN sinir ağıdır [63]. Modern standartlara göre, LeNet-5 çok basit bir ağıdır. Yalnızca 7 katmanı vardır, bunların arasında 3 evrişim katmanı (C1, C3 ve C5), 2 alt örnekleme (havuzlama) katmanı (S2 ve S4) ve 1 tam bağlı katman (F6) vardır ve bunları çıktı izler. LeNet, evrişim katmanlarına ve ardından alt örnekleme katmanına giden $32 \times 32 \times 1$ (gri tonlamalı görüntü) bir girdi görüntüsü alır. Bundan sonra, bir havuzlama katmanı tarafından takip edilen başka evrişim katmanları dizisi vardır. Son olarak, sonunda çıktı katmanını içeren 3 tam bağlantılı (FC) katmanı vardır. Mimarinin temel amacı, el yazısıyla yazılan rakam kalıplarını ve postanelerdeki posta kodu tanıma özelliğini kabul etmektir. Adım 1 ile birlikte 5×5 filtre kullanır. LeNet mimarisine ait yapı Şekil 2.26'da gösterilmiştir.



Şekil 2.26. LeNet mimarisinin yapısı.

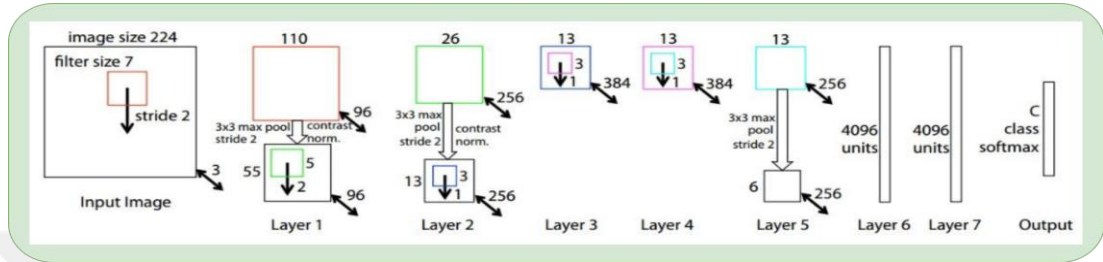
AlexNet: Bu mimari 2012'de Alex Krizhevsky ve diğerleri tarafından geliştirilen derin bir CNN sinir ağıdır [87]. En gelişmiş sonuçları elde ettiği ImageNet [88] LSVRC-2010 yarışması için görüntüleri sınıflandırmak üzere tasarlandı. Ayrıca birden fazla GPU ile çalışabilme kabiliyetine sahiptir. AlexNet, alanındaki en etkili modellerden biridir. AlexNet, 2012 ImageNet LSVRC-2012 yarışmasını büyük bir farkla kazandı. İkinci en iyi girişin aldığı %26,2'ye kıyasla %15,3'lük bir ilk 5 hata oranına ulaşmıştır. AlexNet, görüntü çevirisi, yama çıkarma ve yatay yansımadan oluşan veri büyütme yöntemlerini kullandı. Bu CNN modeli, eğitim verilerine aşırı uyum sorunuyla mücadele etmek için belirli bir son hedefi olan bırakma katmanlarını uygular. Ağırlık düşüşü ve momentum için belirli değerlerle birlikte toplu stokastik gradyan inişi ile eğitilir. AlexNet mimarisine ait temel yapı Şekil 2.27'de gösterilmiştir.



Şekil 2.27. AlexNet mimarisinin yapısı.

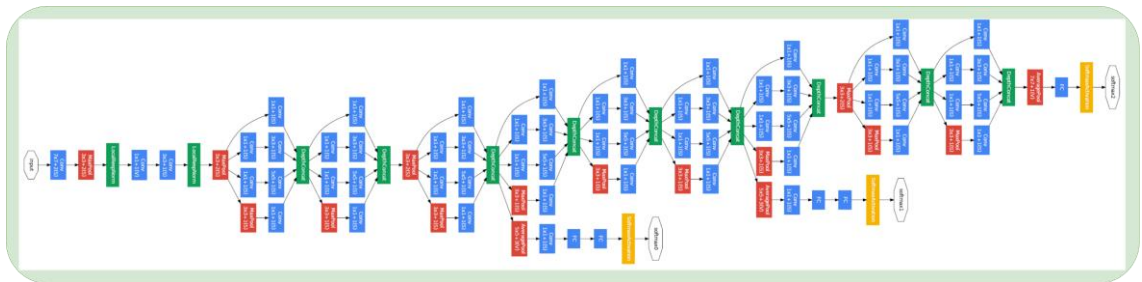
ZfNet: ILSVRC 2013 yarışmasının kazananı ZFNet olarak bilinen CNN mimarisi, geçmişteki AlexNet modeline yapılan bir ince ayardan oluşmaktadır [89]. Şu anda daha önce bahsedilen nöral olmayan hata oranının yarısı olan %14,8'lik bir ilk 5 hata oranına ulaştı. Bu denemede daha önce tartışıldığı gibi ek derin öğrenme öğeleriyle aynı yapıyı

korurken AlexNet'in hiper-parametreleri değiştirilerek elde edilen başarıdır. Bu model, 7×7 boyutundaki filtreleri kullanır ve adım sayısını azaltmaktadır. Bu değişikliğin ötesindeki fikir, birincil evrişim katmanındaki küçük bir filtre boyutunun, giriş hacminde önemli miktarda benzersiz piksel verisi tutmasıdır. ZFNet mimarisine ait temel yapı Şekil 2.28'de gösterilmiştir.



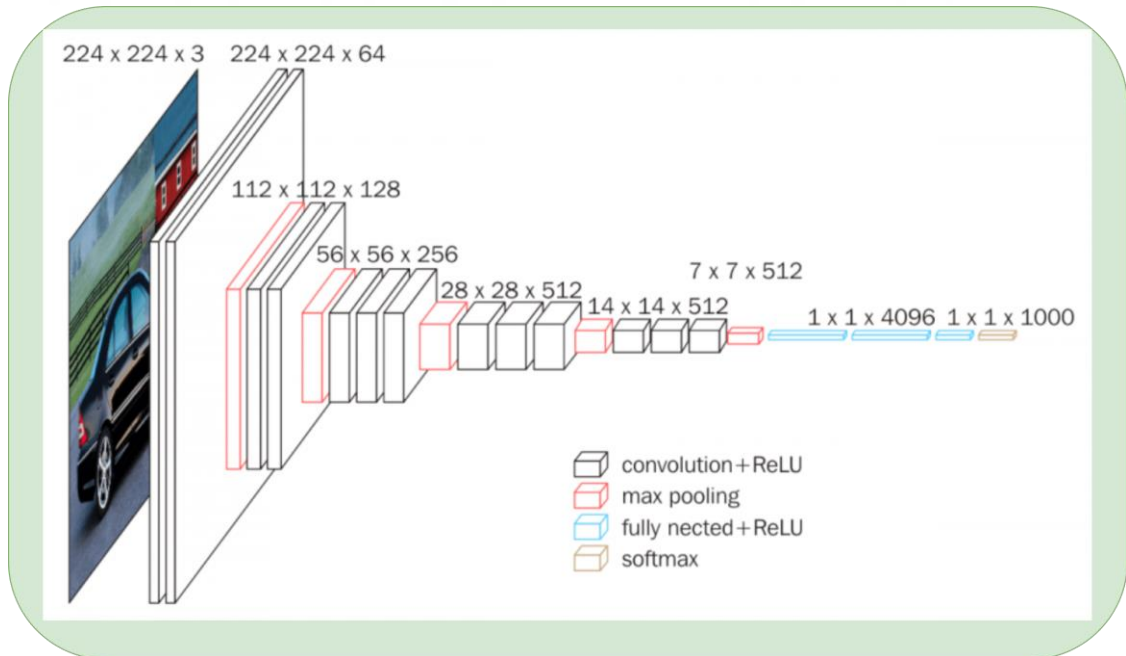
Şekil 2.28. ZFNet mimarisinin yapısı.

GoogleNet: Bu mimari %93,3'lük ilk 5 test doğruluğu ile ImageNet ILSVRC14 algılama ve sınıflandırma zorluğunda en gelişmiş performansı sunmuştur [76]. Bu modelin birincil göstergesi, bilgisayar kaynaklarının modelde daha fazla kullanılmasıdır. Bu, modelin genişletilmiş derinliğini ve genişliğini hesaba katan “başlangıç modülü” olarak bilinen bir yapı taşı aracılığıyla gerçekleştirilmesiydi. Bu, ortak olandan sapan ilk CNN mimarilerinden biriydi. Ağ, aslında LeNet'ten ilham alan bir CNN kullandı, ancak başlangıç modülü olarak adlandırılan yeni bir öge uygulamıştır. Toplu normalleştirme, görüntü bozulmaları ve RMSprop kullanmıştır. Bu modül, parametre sayısını büyük ölçüde azaltmak için birkaç çok küçük konvolüsyona dayanmaktadır. Mimarileri 22 katmanlı bir CNN'den oluşmakla beraber parametre sayısını 60 milyondan (AlexNet) 4 milyona indirdi. GoogleNet mimarisin temel yapısı Şekil 2.29'da gösterilmiştir.



Şekil 2.29. GoogleNet mimarisinin yapısı.

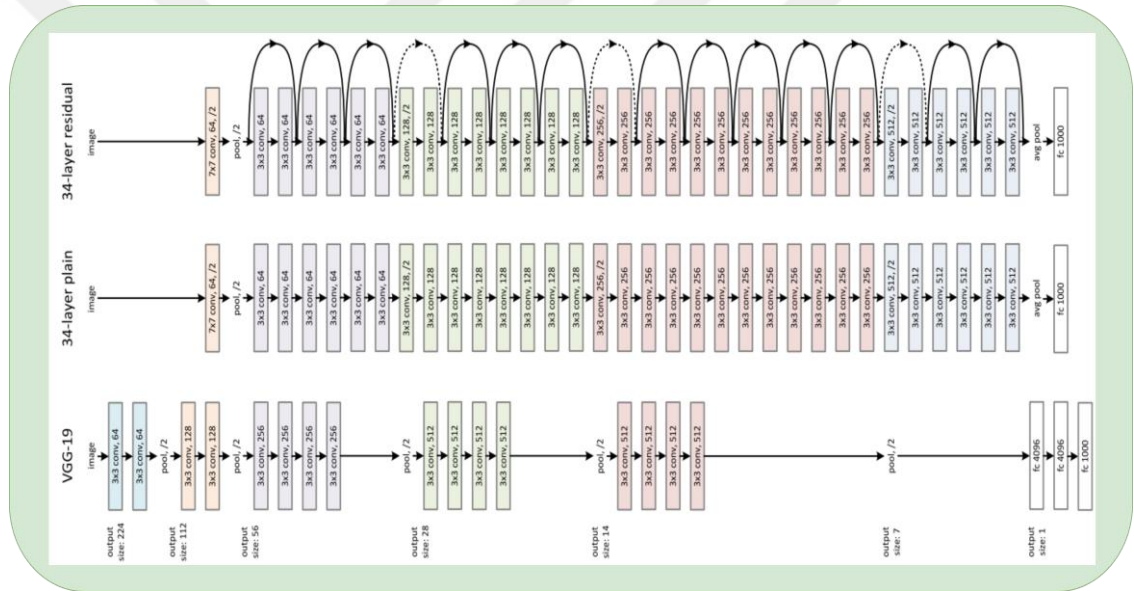
VGG: Görsel geometri grubu (VGG) tarafından Oxford Üniversitesi'nde geliştirilen bu CNN modeli VGG olarak adlandırılmıştır [90]. VGG mimarisi, sadece ILSVRC veri kümeleri üzerinde son teknoloji kesinliği başarmakla kalmayıp, ayrıca başka görüntü tanıma veri tabanlarıyla da oldukça başarılı sonuçlar üretmiştir. VGG mimarisi bulundurduğu katman sayısı bakımından isimlendirilir. Örneğin, 16 katmana sahip olan VGG-16, 3 FC katmanı ile 13 evrişim katmanı içerir; diğer yandan, VGG-19 3 ekstra evrişim katmanı içerir. VGGNet'in ağırlık yapılandırması herkese açıktır ve temel özellik çıkarıcı olarak diğer birçok uygulamada ve zorlukta kullanılmıştır. Diğer taraftan VGGNet, ele alınması biraz zor olabilen 138 milyon parametreden oluşur. AlexNet ve ZFNet, ilk evrişim katmanında 11×11 (4 adım) boyutunda çekirdeği ve 7×7 boyutunda (2 adım) çekirdeği kullanır. Evrişim çekirdeğinin boyutu VGGNet'te daha da azaltılır ve her katmanda 3×3 evrişim çekirdeği (1 adım) kullanılır. Küçük çekirdek ve adım, görüntüdeki nesnenin konum bilgisini çıkarmak için daha elverişlidir. Büyük çekirdek yerine küçük çekirdek, ağırlık derinliğini artırma ve alıcı alanı değişmeden tutma avantajına sahiptir. Parametreler düşürüldükten sonra ağ modelinin özellik temsil kabiliyeti arttırılmıştır. VGGNet mimarisini temel yapısı Şekil 2.30'da gösterilmiştir.



Şekil 2.30. VGG mimarisinin yapısı.

ResNet ağları: ResNet ağları Microsoft'taki bir ekibin derin sinir ağların eğitimini iyileştirmek için artık bir öğrenme çerçevesi sunmasıyla başlatılmıştır [91]. Bu

araştırma 2015 yılında Deep Residual Learning for Image Recognition başlıklı makalede yayınlandı. Böylece ünlü ResNet ("Residual Network"ün kısaltması) ağıları doğmuş oldu. Derin ağıları eğitirken, derinlikteki bir artışın doğruluğun doymasına ve ardından hızla düşmesine neden olduğu bir nokta oluşur. Buna "bozunma sorunu" denir. Bu, tüm sinir ağı mimarilerinin optimize edilmesinin eşit derecede kolay olmadığını vurgular. ResNet ise bu sorunla mücadele etmek için "artık eşleme" adı verilen bir teknik kullanır. Her birkaç yığılmış katmanın doğrudan istenen temel eşlemeye uymasını ummak yerine, Artık Ağ bu katmanların artık bir eşlemeye uymasını açıkça sağlar. Şekil 2.31'de artık bir ağı yapı taşı ile beraber VGG ve 34 katmanlı düz sinir ağı yer almaktadır.



Şekil 2.31. ResNet ve VGG19 mimarilerinin yapısı.

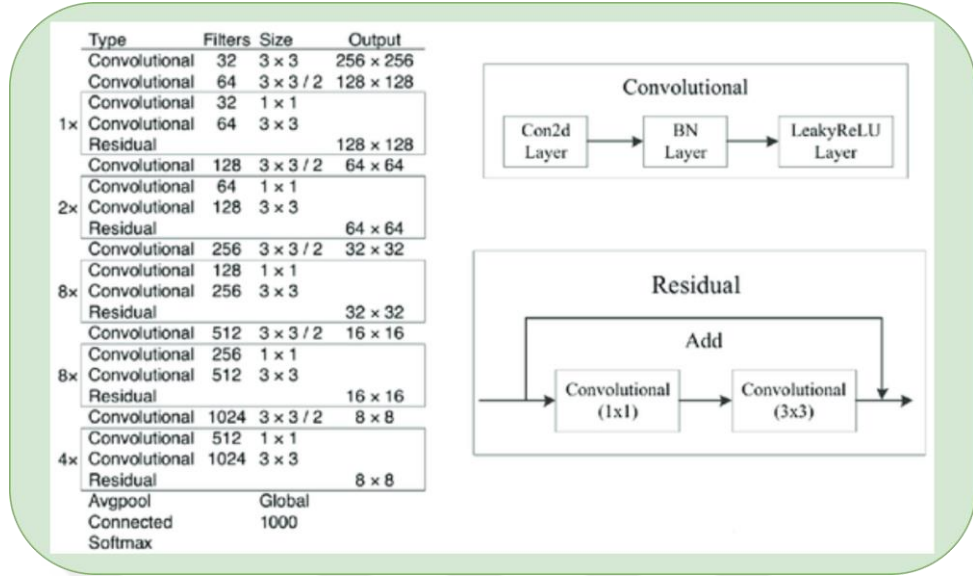
Geleneksel sinir ağı mimarileriyle karşılaştırıldığında, ResNet'lerin anlaşılması nispeten kolaydır. Aşağıda bir VGG ağının, 34 katmanlı düz bir sinir ağının ve 34 katmanlı bir artık sinir ağının görüntüsü bulunmaktadır. Düz ağda, aynı çıktı özellik haritası için katmanlar aynı sayıda filtreye sahiptir. Çıktı özelliklerinin boyutu yarıya indirilirse, filtre sayısı iki katına çıkar ve bu da eğitim sürecini daha karmaşık hale getirir. Bu arada, Artık Sinir Ağı'nda, gördüğümüz gibi, VGG'ye göre eğitim sırasında çok daha az filtre ve daha düşük karmaşıklık var. Ağı muadili artık sürümüne dönüştüren bir kısa yol bağlantısı eklenir. Bu kısayol bağlantısı, artan boyutlar için ekstra sıfır girişleri ile

kimlik eşleme gerçekleştirir. Bu seçenek ek bir parametre getirmez. Şekil 2.32 ise farklı katmanlara sahip ResNet mimarilerindeki katmanları ve parametreleri göstermektedir.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
		3×3 max pool, stride 2				
conv2_x	56×56	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Şekil 2.32. Farklı katmanlara sahip ResNet mimarilerinin parametreleri

DarkNet: YOLO nesne tespiti algoritmalarında özellik çıkarıcı olarak DarkNet ağı önerilmiştir. YOLOv3 omurgası olarak sunulan en son sürüm DarkNet53 iken, YOLOv4 mimarisinde ise CSPNet ile beraber DarkNet53 kullanılmasıyla CSPDarkNet53 sürümü kullanılmıştır. Bu ağ, yakınsamayı hızlandırmak için küresel ortalama havuzlama ve toplu normleştirme ile ardışık 3 x 3 ve 1 x 1 evrişim katmanlarından oluşan önceki sürümü DarkNet19'a dayanmaktadır. Darknet-53 mimarisi, evrişim katmanlarının sayısını 52'ye kadar artırır, dolayısıyla öncekinden önemli ölçüde daha derindir. Derinlikteki bu artışla DarkNet artık hafif bir omurga olarak kabul edilemez. Ayrıca, ResNet mimarilerinden ilham alan DarkNet53 artık bağlantılar içermektedir. Ayrıca, daha güçlü semantik özellikler oluşturmak için birleştirmeyi kullanan atlamalı bağlantılar ve örnekleme tekniklerini de içerir. YOLOv3 mimarisi, piramit ağlara benzer şekilde çok ölçekli tahmine izin vermek için DarkNet omurgasının farklı aşamalarına 3 tahmin başlığı yerleştirir. DarkNet53 önceki sürümden çok daha güçlü ve ResNet omurgalarından hem hesaplama hem de bellek açısından daha verimlidir. ResNet mimarileriyle karşılaştırıldığında, DarkNet53, ResNet-101 ve ResNet-152'ye benzer ancak daha az kayan nokta işlemiyle sonuçlar elde eder, böylece daha hızlı çıkarım oranları elde eder. DarkNet53 mimarisine ait yapı Şekil 2. 34'te gösterilmiştir.



Şekil 2.33. DarkNet53 mimarisinin farklı bir yapısı.

2.6. KRK Teşhisi için Derin Öğrenme Algoritmalarının Kullanımına İlişkin Literatür Çalışması

Bu bölümde KRK teşhisinde derin öğrenme mimari ve algoritmaları ile ilgili yapılmış güncel çalışmalar özetlenmiştir. Bu çalışmalar ise iki ana başlığa ayrılmıştır. Burada amaç, öncelikle KRK teşhisinde derin öğrenme alanında yapılmış çalışmalar özetlendikten sonra derin öğrenme tabanlı gerçek zamanlı polip tespitiindeki güncel çalışmalara değinilmiştir. Bilindiği üzere, derin öğrenme algoritmalarının hızlı bir gelişim sürecindedirler. Daha önce sadece nesne sınıflandırmada popüler olan derin öğrenme mimarileri zamanla nesne tespiti, otonom araçlar ve dil işleme gibi birçok alanda başarılı bir şekilde kullanıldı. En önemli gelişmeler ise derin öğrenme algoritmalarının sağlık ve savunma gibi kritik alanla başarılı sonuçlar vermesiyle derin öğrenme alanlarındaki uygulamaların yelpazesi oldukça genişledi. KRK teşhisinde ise asıl sıçrama derin öğrenme tabanlı nesne tespit algoritmalarında meydana gelen ciddi değişimler oluşturmıştır. Özellikle tek aşamalı yaklaşımlardan olan YOLO-tabanlı algoritmalarda gerçek zamanlı tespit ile beraber ciddi bir doğruluğun sunulması ile beraber polip tespitiinde de ciddi başarımlar elde edilmeye başlanmıştır. YOLO algoritmaları hem gerçek zamanlı teşhis komsunda hem de tespit doğruluğu açısından diğer muadil algoritmalarından oldukça başarılıdır. Literatüre bakıldığında da YOLO

tabanlı algoritmalar polip tespiti konusunda en çok araştırılan ve kullanılan algoritmadır.

2.6.1. KRK Teşhisi için Yapılmış Çalışmalar

Bu bölümde kolon, kolorektal ve rektal kanserlerinde derin öğrenme yöntemlerinin kullanımına ait literatür taraması yapılacaktır. Literatür çalışmasında kolon, rektal ve kolorektal kanserleri tek bir başlık altında, yani kolon kanseri başlığı altında incelenmiştir. Literatür taramasına bu başlıklarla beraber “polip tespiti” eklenerek daha geniş bir arama kümesi seçilerek derin öğrenmenin kolon kanseri teşhisinde kullanım alanları araştırılmıştır. Son zamanlarda, GPU donanımlarının ekonomik hale gelmesi ve daha derin öğrenme modellerinin sunulması ile beraber kolon kanseri üzerine yapılan çalışmaların büyük çoğunluğu artık gerçek zamanlı polip tespiti üzerine yoğunlaşmıştır. Daha önceki çalışmalar ise daha çok sınıflandırma ya da patolojik görüntülerin sınıflandırmasına dayanıyordu. Bununla beraber robot cerrahisinde de kullanılmak üzere son dönemlerde polip bölümlene çalışmaları da oldukça artmıştır. Burada, derin öğrenme yöntemlerinin kolon kanserinde kullanılmasından ziyade, daha özel bir alan olan kolon kanseri tespitinde yapılan çalışmalara yer veriyoruz. Daha sonra ise daha özel bir alan olan gerçek zamanlı polip tespiti üzerine yapılan çalışmalara odaklanıyoruz.

Derin öğrenme, nesne algılama ile büyük başarı elde ederken, nesne algılamaya benzeyen tıbbi görüntü analizinde de derin öğrenme yeni başarılarla imza atmıştır. Derin öğrenme uygulamaları, kanser tarama ve tespiti, hastalık izleme ve tedavi yöntemleri gibi sağlık hizmetlerinde kullanılmaktadır. Sürekli artan tıbbi veriler ve bu verilerin faydalı bilgilere dönüştürülmesi, derin öğrenmeyi CAD sistemlerinde merkezi bir yere koyuyor. Derin öğrenmenin kolon kanseri teşhisindeki performansı gerçekten dikkat çekicidir [92]. Patoloji görüntülerinin analizi veya kolonoskopi görüntülerinin analizi gibi konularda başarı, kolon kanseri tanısında önemli bir yere sahiptir. Derin öğrenmenin kolon kanseri analizindeki başarısını daha iyi göstermek için, toplanan çalışmaları beş ana kategoriye ayırdık. Bu kategoriler şunlardır: tespit, sınıflandırma, bölütleme, hayatta kalma tahmini ve iltihaplı bağırsak hastalıklarıdır. Beş uygulamanın kısa açıklamaları aşağıdaki sırayla verilmiştir. Kısaca tespit modülü, koloskopi görüntülerinde polipleri tespit etmeyi veya patoloji görüntülerinde tümör hücrelerini

bulmayı amaçlar. Bölümleme, patoloji görüntülerindeki bölümlerin bölümlenmesini veya polip parçalarının bölümlenmesini amaçlar. Sınıflandırma modülü, polipin veya polip hücresinin kanserli olmasını veya farklı tiplerde sınıflandırılmasını sağlar.

Tümörlerin, çekirdeklerin veya lezyonların tespiti, tıbbi görüntü analizinde hastalığın teşhisinde en önemli rolü oynar. Nesne algılama sisteminin amacı, nesne bölütleme, kötü huylu ve iyi huylu tümörler arasında ayırım veya tümör veya lezyonların saptanması için güçlü destek sağlar [93]. Tam otomatik uçtan uca uygulamalarda lokalizasyon kullanılarak tıbbi görüntülerin detaylı analizi insan müdahalesi olmadan yapılabilmektedir [94]. Yapılan çalışmalarda bu yöntem kolon kanseri patoloji görüntülerinde malign ve iyi huylu hücre tespiti veya çekirdek tespiti yaparak kolon kanseri teşhisine yardımcı olur. Özellikle kolonoskopi görüntülerinde poliplerin saptanması KRK'nin erken evresinde en önemli adımdır.

Kolonoskopi, kolon kanseri taramasında en çok kullanılan teknik ve altın standarttır. Kolonoskopinin amacı, kolon kanserlerinin öncüsü olan kolon poliplerini bulmak ve çıkarmaktır. Bu poliplerin şekilleri, renkleri ve boyutları değişebilir. Ayrıca kullanılan kolonoskopi cihazının kalitesi ve elde edilecek poliplerin görüntüleri değişkenlik gösterebilmektedir. Tüm bu etkenlerle polip benzeri yapılarında olmasıyla beraber, poliplerin tespit edilmesi zor bir problemdir. Bu da uzman hekimlerin zorlu bir süreç içerisine girmesine ve bazen yanlış sonuçlar almasına neden olabilir. Genelde polip kaçırma oranı %6 ile %27 arasında değişen bir orandadır [9,11,15]. Yanlış sonuçlar almak kolon kanserinin yayılmasına ve metastatik hale gelmesine neden olabilir. Metastatik hale gelen kolon kanseri hayatta kalma oranını azaltır. Bu nedenle CAD tabanlı derin öğrenme sistemleri, polip tespitinde önemli bir yere sahiptir. Özellikle son yıllarda yapılan çalışmalar ile polip tespit oranlarındaki başarı artmış ve daha fazla çalışma yapılmıştır. Bu çalışmaların bazıları şu şekildedir. Ozawa ve ark. [95] CNN tabanlı Single Shot MultiBox Detector (SSD) mimarisini kullanarak hem polip tespiti hem de polip sınıflandırma yöntemini önerdi. Önerdikleri yöntemleri ise özel bir veri seti uyguladılar. Deneysel sonuçlar, SSD nesne tespit algoritmasının KRK analizinde başarılı olduğunu gösterdi.

Saito ve ark. [96] kablosuz kapsül endoskopi (WCE) görüntülerinde çeşitli çıkıntılı lezyonları otomatik olarak tespit etmek için CNN tabanlı bir SSD mimarisi önerdiler.

SSD mimarisi, 292 hastadan çıkıntılı lezyonların 30.584 WCE görüntüsünden oluşan özel bir veri seti ile eğitildi. Deneysel sonuçlar, SSD mimarisinin KRK taramasında ve analizinde başarılı sonuçlar verdiği gözlemlenmiştir. Qadir ve ark. [79] kolonoskopi videolarında polip tespit performansını iyileştirmek için yeni bir polip tespit çerçevesi tanıttı. Bu çerçeve herhangi bir nesne detektörü ile kullanılabilir. Önerilen yöntem, geçici video analizini ve bireysel çerçeve analizini birleştirerek eldeki durum için nihai bir karar vermeye odaklıdır. Önerilen şema, gözden kaçan polipleri daha iyi saptayabilir ve algılama çıktısını daha da iyileştirmektedir. Bu yöntem, güncel CNN yaklaşımları olan SSD ve Faster R-CNN ile denenmiştir. MobileNet omurgasıyla SSD kullanılırken, Inception-ResNet omurgasıyla Faster R-CNN kullanılmıştır. Önerilen yöntemin sonuçları önceki yöntemlerden daha iyi olduğu gözlemlenmiştir.

Wang ve ark. [97] gerçek zamanlı polip tespiti için yeni bir çapa içermeyen yöntem kullanmayı önerdiler. Bu kapsamda CenterNet tabanlı mimari ve VGG16 omurgası ile bazı iyileştirmeler gerçekleştirdiler. Gerçek zamanlı tespitler yapabilmek için CenterNet yapısındaki merkez havuzunu kaldırdılar ve böylece CenterNet yapısı daha hızlı bir tespit yapabilme kabiliyetine ulaştı. Ek olarak, geri çağırma kaybının önüne geçmek için bir kosinüs yer-gerçek izdüşümü stratejisi önerdiler ve daha etkili CenterNet tabanlı bir polip tespit dedektörü elde ettiler. Daha sonra eğitim için EndoSceneStill veri setini ve test süreci için ise CVC-ClinicDB ve Etis-Larib veri setlerini kullandılar. En iyi sonuçları ResNet-50 ve ResNet-101 yerine VGG16 omurgası ile elde ettiler. Nadimi ve ark. [98] , WCE sırasında yakalanan görüntülerde kolorektal poliplerin otonom tespiti için uçtan uca bir CNN tabanlı derin öğrenme modeli önerdiler. Kolorektal polip görüntülerinin bölgelerini sınırlandırmak için Faster R-CNN algoritmasını kullandılar. Veri artırma teknikleri ile birlikte eğitim ve test aşamaları için toplam 11.300 WCE görüntüsü topladılar. Bu çalışma, daha önce yayınlanmış diğer çalışmalardan ve modern yaklaşımlardan daha yüksek doğruluk, kesinlik ve özgüllük elde etti. Liu ve ark. [99] üç özellik çıkarıcı ile yeni nesne algılama yöntemlerinden biri olan SSD mimarisi uyguladı. Bu çalışmada, polipleri saptamak için InceptionV3, ResNet-50 ve VGG16 kullanıldı. Çalışmada, halka açık veri kümelerini kullandılar ve modellerini diğer polip tespit yöntemleriyle karşılaştırdılar. Ayrıca SSD'ye birkaç katman ekleyerek modellerin daha derin yapılara sahip olarak başarımının artmasını sağladılar. Mostafiz ve ark. [100] kolonoskopi videolarında poliplerin tespiti için CNN ve deneye dayalı mod

ayrışmasından (CEMD) oluşan bir füzyon modeli önerdi. Bu iki yöntem birleştirildikten sonra polipler SVM ile sınıflandırıldı. Bu model, daha önce yayınlanan diğer çalışmalardan daha başarılı sonuçlar verdi. Hasan ve ark. [101] daha sonra hem video hem de görüntü verilerinden polipleri tespit etmek için bir CNN ve kontur dönüşümü içeren bir füzyon modeli önerdi. Bu model, genel bir veri seti ile polip yamaları üzerinde değerlendirildi. Deneysel sonuçlarda, birden fazla CNN mimarisi kullanılmasına rağmen, VGG19 mimarisi en iyi sonuçları verdi.

Ladha ve ark. [102] gastrik polip tespiti için CNN tabanlı bir YOLOv3 nesne tespit algoritması kullanmıştır. Bu model ile ortalama 0,91 hassasiyet elde ettiler. Bu algoritma polip tespitinde hız ve doğruluk açısından başarılı sonuçlar vermiştir. Wang ve ark. [40] , derin öğrenme yöntemlerine dayalı otomatik polip tespit sisteminin ADR ve polip tespit oranı üzerindeki etkisini araştırdı. Deneyleri, düşük prevalanslı ADR bölgesinde hem kolorektal polip hem de adenom tespit oranlarında önemli bir artış olduğunu göstermiştir. Sornapudi ve ark. [19] , hareketsiz çerçevelerden tespit edilen poliplerin etrafında maskeler oluşturmak için değiştirilmiş bir R-CNN tabanlı algoritma önerdi. Bu yöntem, Resnet-50 ve Resnet-101 omurgası kullanılarak özellik çıkarımı sağlarken, model ince ayar ve önceden eğitilmiş ağırlıklarla daha etkili olmuştur. Polip tespit oranını iyileştiren Balon verilerini ve ince ayarını kullandılar. Ayrıca bu çalışmada, bu modeller daha önce yayınlanmış çalışmalara kıyasla WCE görüntülerinde ve kolonoskopi görüntülerinde daha iyi performanslar sağlamıştır. Wittenberg ve ark. [103] , ResNet-101 omurgasına sahip Mask R-CNN mimarisine dayalı otomatik polip tespiti için bir yaklaşım sundular. İki bağımsız veri kümesinden ve 2484 görüntü içeren özel bir veri kümesinden oluşan kolonoskopi görüntüleriyle Mask R-CNN'yi eğittiler ve test ettiler. Test veri setini üç parçaya böldüler ve bu veri setinin her bir parçası için modelleri test ettiler. Deneysel sonuçlar, eğitilen modellerin otomatik polip tespitinde başarılı sonuçlar ortaya koyduğu gözlemlendi. Urban ve ark. [15] , gerçek zamanlı videolarda polip algılama ve yerleştirme yapısını tanımlamak için gelişmiş bir evrişimli sinir ağı geliştirdi. VGG16, VGG19 ve ResNet50 modellerindeki ağırlıklarla önceden başlatılmış ağırlıkları kullandılar. Bu çalışmada, 2.000 hastadan 4.088 benzersiz polip içeren 8.641 elle etiketlenmiş görüntüden oluşan devasa bir veri seti kullanıldı. CNN modelleri önce önceden eğitilmiş ağırlıklar eğitildi ve daha sonra poliplerin tespiti için kullanılmıştır. Bu gerçek zamanlı model, yüksek doğrulukta

sonuçlar gösterme yeteneğine sahip olmakla beraber, klinik araştırmalar için kullanılabilir bir çıkarım hızına sahipti. Yuan ve ark. [104] kolonoskopi videolarında polipleri tespit etmek için birkaç aşamadan oluşan bir yöntem önerdi. İlk aşama, görüntülerin çok düzeyli süper piksellere bölünmesiydi. Bir sonraki aşama ise yararlı özellikleri öğrenmek için seyrek bir otomatik kodlayıcı kullanmaya dayanıyordu. Daha sonra aşağıdan yukarıya ve yukarıdan aşağıya belirginlik tekniğinin birleştirilmesiyle polipleri tespit ettiler. Önerilen metodları belirginlik için 0.818 hatırlama başarısına sahipti. Çalışmada elde edilen bu başarı, otomatik polip tespiti için iyi bir doğruluk ve kesinlik sunmaktadır.

Shin ve ark. [20] Inception ResNet'in omurgasını içeren Faster R-CNN tabanlı bir derin öğrenme yöntemini temel alan bir polip algılama yöntemi sundu. Modelleri, hem algılama görüntülerinde hem de videolarda hassasiyet, geri çağırma ve tepki süresi metrikleri açısından önceki modellerden daha iyiydi. Modelleri, polip ve polip olmayan gibi yamalar yerine tam görüntülerle eğitildi. Deneysel sonuçlar, önerilen metodun polip tespiti konusunda başarılı olduğunu ancak gerçek zamanlı tespit konusunda ise yetersiz olduğunu göstermektedir. Misawa ve ark. [105] , daha önceki derin öğrenme teknikleri ile karşılaştırıldığında video veri kümeleri alanında yeterli olan, uzay-zamansal veriler için tasarlanmış 3D CNN'ye dayalı yeni bir orijinal algoritma geliştirdi. Bu algoritma, polip tespiti konusunda sundukları özel veri setlerinde başarılı sonuçlar vermiştir. Wang ve ark. [99] kolonoskopi sırasında kolon poliplerinin tespiti için CNN'ye dayalı bir algoritma geliştirdi. Metrikleri doğrulamak için dört bağımsız farklı veri seti kullandılar. Veri setinin yarısı görüntü analizleri için yarısı ise video analizleri için kullanıldı. Deneyleri, gerçek zamanlı inceleme ile yüksek duyarlılık ve özgüllüğe ulaştıklarını gösterdi. Zhang ve ark. [106] kolonoskopi sırasında polipleri saptamak için yeni bir regresyon tabanlı CNN önerdi. Algoritmaları iki parçaya dayanıyordu. İlk bölüm, poliplerin uzamsal yönlerini çıkarmak için önceden eğitilmiş bir model olan ResYOLO'yu kullanmayı amaçladı. İkinci kısım, etkin evrişim operatörleri (ECO) adlı bir izleyici aracılığıyla birleştirilmiş zamansal bilgiler kullanılarak ResYOLO tarafından elde edilen algılama sonuçlarını sınırlamak için kullanıldı. Bu metod, kolonoskopi esnasında polipleri başarılı bir şekilde saptamayı başarmıştır. Pogorelov ve ark. [107] kolonoskopi görüntüleri için gerçek zamanlı polip tespiti için birkaç makine öğrenimine dayalı yaklaşımlar sunmuş ve değerlendirmiştir. En iyi sonuçları, üretken ağlara

dayanıyordu. Hem eğitim hem de test aşamaları için birkaç genel veri seti kullandılar. Ayrıca, en son teknolojiye sahip çalışmalarla üretken çekişmeli ağ algoritmalarına dayanan en iyi ağlarıyla karşılaştırdılar. Yapılan deneysel sonuçlarda, GAN tabanlı algoritma, diğer polip tespit yöntemlerine göre daha az veri seti ile daha iyi bir başarı oranı sağlamıştır.

2.6.2. Gerçek Zamanlı Polip Tespiti için Yapılmış Güncel Çalışmalar

Bu bölümde gerçek zamanlı polip tespiti için önerilmiş güncel çalışmalardan bahsedilecektir. Bilindiği üzere bir nesne tespit algoritmasının gerçek zamanlı olarak çalışabilmesi için en az 33 saniyedeki kare sayısı (FPS) üretmesi lazım. Bu nedenle iki aşamalı yaklaşımlardan olan R-CNN tabanlı algoritmalar gerçek zamanlı tespit yapmada oldukça uzaktırlar. Gerçek zamanlı tespit konusunda yapılmış çalışmaların neredeyse tamamına yakını tek aşamalı yaklaşımlardan ve özellikle YOLO tabanlı algoritmalarından oluşmaktadır. Şimdi en güncel bazı yaklaşımlara bakalım. Rodriguez ve ark. [47] önceden eğitilmiş bir YOLOv3 mimarisine dayanan ve yanlış pozitifleri azaltmak için bir nesne izleme algoritmasına dayalı bir son işleme adımıyla tamamlanan gerçek zamanlı polip tespiti için bir derin öğrenme modeli önerdiler. YOLOv3 tabanlı ağ, 941 polip konumlarıyla etiketlenmiş 28.576 görüntüden oluşan bir veri kümesi kullanılarak eğitilmiştir. Bu model büyük bir veri seti ile eğitildiği için bir CAD sistemine etkin bir şekilde bütünleşmiş edilebileceğini göstermektedir, fakat yeni veriler üzerindeki başarımı tam olarak nesnel değildir. Pacal ve Karaboğa [14] YOLOv4 algoritmasını ölçeklendirerek gerçek zamanlı polip tespiti için yeni bir yapı önerdiler. Bunun için önce tüm yapıyı Çapraz Aşamalı Kısmi Ağlar (CSPNet) ile değiştirdiler, ardından Sızdıran ReLU aktivasyon fonksiyonu yerine Mish aktivasyon fonksiyonunu kullandılar. Buna ek olarak, CIOU kayıp fonksiyonu yerine DIOU kaybını kullandılar. Yazarlar daha sonra, ResNet, VGG, DarkNet53, Transformers gibi farklı yapıları kullanarak YOLOv3 ve YOLOv4 mimarilerinin performansını iyileştirdiler. Önerilen yöntemin başarısını artırmak için, ön işleme için çeşitli veri büyütme yaklaşımları, bir topluluk öğrenme modeli ve son işleme için ise NVIDIA TensorRT yapılarını kullandılar. Önerilen yöntem gerçek zamanlı performansı ve son teknoloji algılama doğruluğu ile diğer yöntemlerden farklıdır. Önerilen yöntem (toplu öğrenme olmadan) literatürde bulunanlardan daha yüksek sonuçlar elde etti. Bu metoda ait bazı metrikler

ise şu şekildedir. Etis-Larib veri seti üzerindeki metrikler sırasıyla; kesinlik: %91,62, hatırlama: %82,55, F1 puanı: %86,85 elde ederken, CVC-ColonDB veri setinde ise kesinlik: %96,04, hatırlama: %96,68 %, F1-puanı: %96,36 elde ettiler. Önerilen metot ile elde edilen bu metrikler, diğer çalışmalardan hem hız hem de doğruluk olarak daha ilerdedir. Li [108] YOLOv3 algoritmasını sırasıyla 6038 polip görüntüsü ile eğittikten sonra 2571 polip görüntüsü ile test etmişlerdir. Algoritma, CVC-Video ClinicDB veri setinden eğitime dahil edilmeyen 10 video üzerinde test edildi. Yalnızca birleşim alanı üzerinde kesişimi $> 0,3$ olan sınırlayıcı kutular olumlu tahminler olarak kabul edildi. Modelin klinik öncesi uygulamalar için başarılı olduğu deneysel olarak kanıtlanmıştır.

Qadir ve ark. [16] gerçek zamanlı polip algılama sistemi geliştirmek için tek atış ileri beslemeli tamamen evrişimli sinir ağları (F-CNN) kullandılar. F-CNN'ler genellikle nesne bölümlleme için ikili maskeler üzerinde eğitilir. Bu modellerin farklı polip tiplerini daha etkili ve verimli bir şekilde tespit etmesini sağlamak ve yanlış pozitiflerin sayısını azaltmak için ikili maskeler yerine 2D Gauss maskelerinin kullanılmasını önerdiler. Deneysel sonuçlar, önerilen 2D Gauss maskelerinin, arka plan ve polip parçaları arasında belirsiz sınırları olan düz ve küçük poliplerin tespiti için etkili olduğunu göstermiştir. Maskeler, polipleri polip benzeri yanlış pozitiflerden ayırt etmek için daha iyi bir eğitim etkisi oluşturduğu gözlemlenmiştir. Önerilen yöntem, iki polip veri kümesi üzerinde başarılı sonuçları elde etti. Etis-Larib veri setinde %86,54 geri çağırma, %86,12 kesinlik ve %86,33 F1 puanı elde ederken, CVC-ColonDB'de ise %91 geri çağırma, %88,35 kesinlik ve %89,65 F1 puanı elde ettiler. Saito ve ark. [78], WCE görüntülerinde çeşitli çıkıntılı lezyonları otomatik olarak tespit etmek için CNN tabanlı bir SSD mimarisi geliştirdi. SSD mimarisi, 292 hastadan çıkıntılı lezyonların 30.584 WCE görüntüsünden oluşan özel bir veri seti ile eğitildi. Deneysel sonuçlar SSD mimarisinin WCE görüntülerinde gerçek zamanlı otomatik polip tespitinde başarılı sonuçlar verdiği gözlemlenmiştir. Qadir ve ark. [17] kolonoskopi videolarında polip tespit performansını iyileştirmek ve geçici bilgileri bütünleşmiş etmek için yeni bir polip tespit çerçevesi tanıttı. Bu çerçeve herhangi bir nesne dedektörü ile kullanılabilir. Önerilen yöntem, geçici video analizini ve bireysel çerçeve analizini birleştirerek eldeki durum için nihai bir karar verdi. Önerilen şema, gözden kaçan polipleri daha iyi saptadı ve algılama çıktısını iyileştirdi. Önerilen yöntem, güncel CNN yaklaşımları olan SSD ve Faster R-CNN ile denenmiştir. MobileNet omurgasıyla SSD kullanılırken, Inception-

ResNet omurgasıyla Faster R-CNN kullanıldı. Önerilen yöntemin sonuçları çoğu durumda önceki yöntemlerden daha iyi olduğu gözlemlenmiştir. Wang ve ark. [109], gerçek zamanlı polip tespiti için yeni bir çapa içermeyen yöntem kullanmayı önerdi. Bu kapsamda CenterNet tabanlı mimari ve VGG16 omurgası ile bazı iyileştirmeler gerçekleştirdiler. Gerçek zamanlı tespitler yapabilmek için CenterNet yapısındaki merkez havuzunu kaldırdılar ve daha hızlı tespite kavuştular. Ek olarak, geri çağırma kaybının önüne geçmek için bir kosinüs yer-gerçek izdüşümü stratejisi önerdiler ve daha etkili bir polip tespit detektörü elde ettiler. Daha sonra eğitim için EndoSceneStill veri setini ve test süreci için CVC-ClinicDB ve Etis-Larib veri setlerini kullandılar. En iyi sonuçları ResNet-50 ve ResNet-101 yerine VGG16 omurgası ile elde ettiler. Deneysel sonuçlar çapa içermeyen CenterNet nesne tespit algoritmalarının otomatik polip tespitinde başarılı bir şekilde kullanılabileceğini göstermektedir.

3. BÖLÜM

VERİ SETLERİ VE DEĞERLENDİRME METRİKLERİ

3.1. Giriş

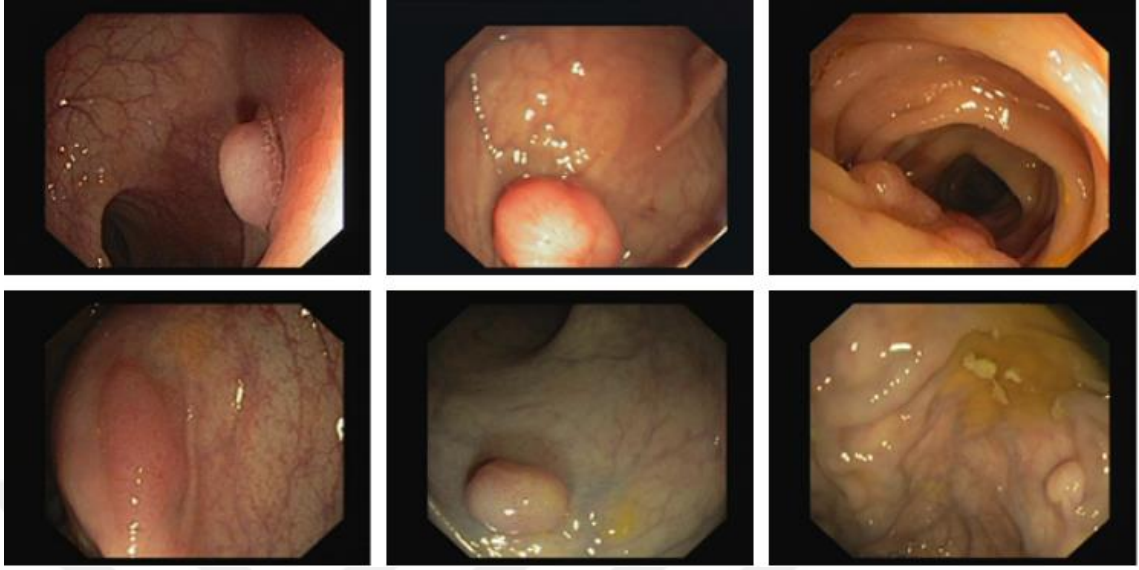
Derin öğrenme modelleri klasik makine öğrenme algoritmalarından farklı olarak, bünyesinde bulundurduğu hiyerarşi sayesinde öğrenmeyi veri temsillerine dayandıran algoritmalar. Bu modeller otonom araçlar, nesne tespiti, bilgisayarlı görü, konuşma tanıma ve tıp alanında giderek daha fazla kullanılmaktadır [110]. Bunların hepsinin temelinde, derin öğrenme algoritmalarının tabiatında var olan veriye aç özelliğidir. Veriye aç özellik ise, veri setlerinin klasik makine öğrenme algoritmalarındaki gibi yüzler seviyesinden ziyade, her bir sınıfa ait binlerce ya da yüzbinlerce veriye ihtiyaç hissetmeleridir. Yüzler seviyesinde olan bir veri setindeki başarımlar %80'lerde iken, bu oran binlere çıktığında ise %90'lara çıkabilir [111]. Yani, derin öğrenmenin en temelinde ve en başında yeterli bir veri seti gelmektedir. Bir veri kümesi bazen veriye aç derin öğrenme mimarilerinin eğitimi için yeterli veri içermeyebilir. Bu gibi durumlarda genellikle veri büyütme uygulanır veya transfer öğrenme kullanılır. Anlamsal düzeydeki görüntü etiketini etkilemeyen basit görüntü dönüşümleri uygulanarak daha fazla örnek eklemek için veri büyütme yapılır. Fakat bu gibi durumlar tamamen o anki uygulamayı kurtarmaya yöneliktir, yani tam anlamıyla modelin ihtiyaç duyduğu veriyi sunamıyor. Veri setinden sonraki aşamalar ise daha çok matematiksel ilişkiler, veri temsili ve hatayı minimize etme yani optimizasyonlardır. Bu nedenle, etkili ve kullanışlı bir derin öğrenme modeli oluşturmanın zorluğu, kullanılacak veri tabanının doğru boyutunu bilmekten geçiyor. Çok küçük bir veri tabanının kullanılması, modelin tüm seviyelerde aşırı basitleştirilmiş sonuçlar vermesiyle ilişkilendirilirken, yani sadece minimum uygulama için etkilidir denilebilir. Öte yandan, çok büyük bir veri tabanı kullanmak için ise daha büyük bir ağ mimarisi ve yeterli seviyede donanım

ihtiyaç olacaktır [112]. Bununla beraber, zaman ve maliyet hesabı da eklenecektir. Sonuç olarak, yeterli ve etkili bir veri tabanı ile probleme ait bir derin öğrenme mimarisi kullanılarak, çözüme gidilebilir. Aksi halde, ya model kısıtlı bir sonuç verecek ya da çok uzun sürmekle beraber ciddi oranda maliyet getirecektir. Bu bölümde, dünyada kullanılan halka açık veri setlerinin yetkinliği ve etkinliği araştırılmakla beraber, klinikte uygulanabilirliği de araştırılmıştır. Özellikle son zamanlarda kolonoskopi alanında yeni ve etkili veri setleri halka açık hale getirilmekle beraber yeni çalışmaların önü de açılmış olmaktadır.

3.2. Açık Veri Setleri

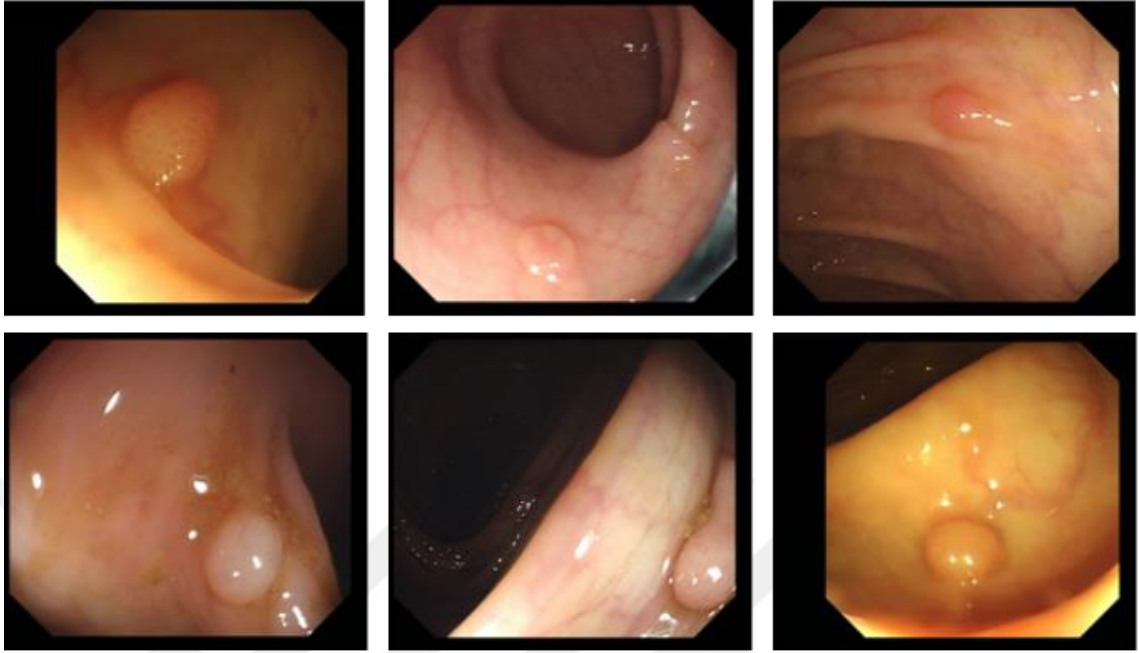
Bu bölümde, dünyada açık kolonoskopi veri tabanları incelenmiştir. 2015'te MICCAI polip yarışmasıyla beraber halka açık veri tabanları sunulmaya başlanmış ve son zamanlarda bu sayı artmıştır [113]. Herkese açık veri setlerinin sayesinde önerilen modellerin etkinliği ve başarımı daha objektif olmakla beraber, bu alanlarda daha fazla araştırmanın yapılmasına olanak sağlar. Diğer taraftan özel veri setlerinin çok kullanılması ile beraber çoğu zaman modellerin başarımı karşılaştırılma zorluğu ile beraber, yeni yöntem ve metotların gelişmesine engel oluyor. Bu yüzden, özellikle polip tespitinde yeni metot ve başarımların oluşmasına en büyük katkıyı açık veri setleri sağlamaktadır. Şimdi bu veri setlerine ve kullanım amaçlarına sırasıyla bakalım.

CVC-ClinicDB: bu veri seti [113], kolonoskopi videosundan çıkarılan bir video karesi veri tabanıdır. 31 farklı video dizisinden elde edilen bu veri seti .tiff formatında olmak üzere 612 adet görüntüden ve toplam 31 adet eşsiz polipten oluşmaktadır. Her görüntüye ait yer gerçeği yani etiket yer almaktadır. Herhangi bir csv ya da xml dosyası olmayan bu veri tabanı 384x288 standart görüntü kalitesine sahiptir. Birden fazla uzman hekim tarafından etiketlenmiş bu görüntülerde bütün görüntülerde en az bir tane polip bulunmakla beraber bazı görüntülerde ise 3 adet polip görülebilir. 2015 MICCAI polip yarışmasıyla beraber bu veri seti halka açık hale getirilmiş ve birçok çalışmada kullanılmıştır. Bu veri setine ait bazı görüntüler Şekil 3.1'de gösterilmiştir.



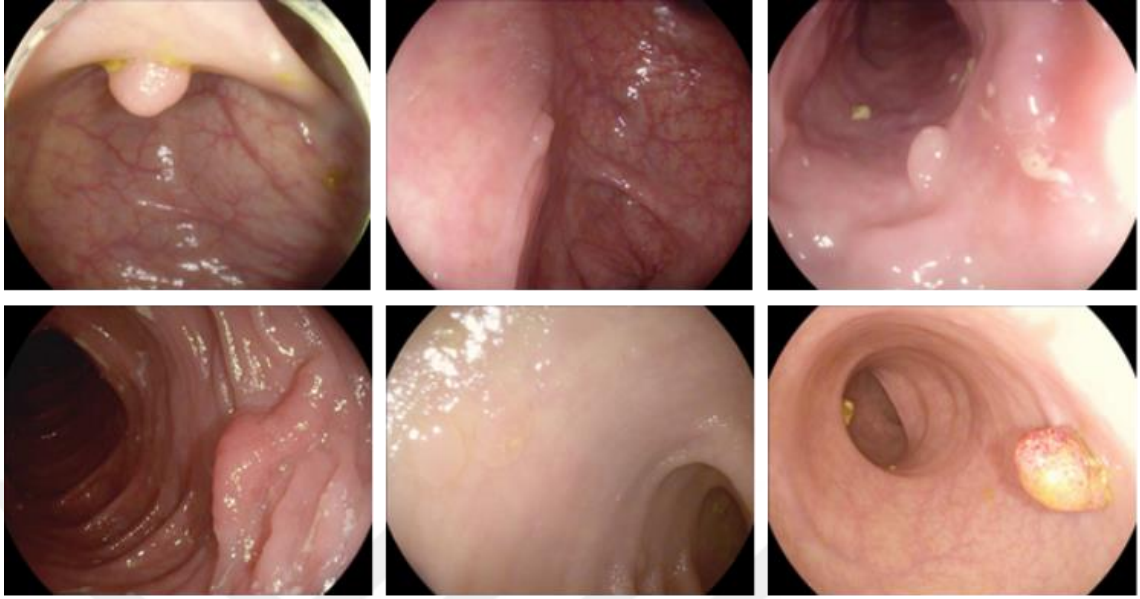
Şekil 3.1. CVC-ClinicDB veri setine ait bazı görüntüler.

CVC-ColonDB: Bu veri setinde [113] toplamda 13 hastadan alınmış video dizisinden elde edilmiş görüntüler mevcuttur. Görüntülerin çözünürlükleri standart kalitede ve 574x500 çözünürlüğündedir. Toplam görüntü sayısı 300 olan bu veri setinde bazı görüntülerde birden fazla polip bulunmaktadır. Bu veri seti genellikle polip bölümlleme işlemlerinde kullanılmasına rağmen polip tespitinde de kullanılmaktadır. Görüntüler uzmanlar tarafından etiketlenmekle beraber CVC-ClinicDB veri setindeki formatlar takip edilmiş yani her bir polip görüntüsüne ait yer gerçeği bulunmaktadır. Bu veri setine ait bazı görüntüler aşağıda gösterilmiştir.



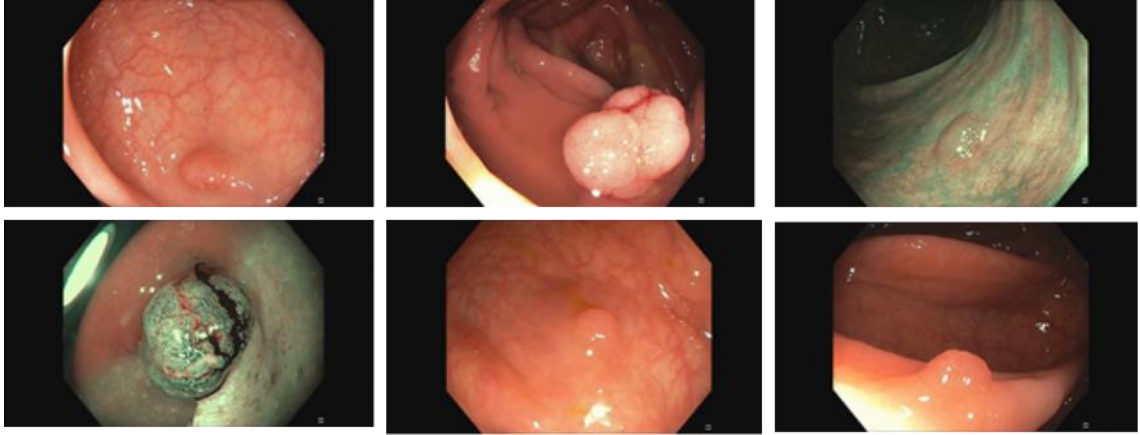
Şekil 3.2. CVC-ColonDB veri setine ait bazı görüntüler.

Etis-Larib: 34 farklı kolonoskopi video dizisinden elde edilen bu veri [113] setinde toplam eşsiz polip sayısı 44 tanedir. Görüntü kalitesi ise 1225×966 piksel HD (yüksek tanımlı) çözünürlüğe sahip olan bu veri setinde ise toplam 196 polip görüntü vardır [16]. 196 görüntünün tamamında en az bir polip mevcuttur ve bazılarında ise ikiden fazla polip olmakla beraber toplamda polip sayısı 212'dir. Bununla beraber yine CVC-ClinicDB olduğu gibi herhangi bir CSV ya da XML dosyası olmayan bu veri setinde, etiketler segmentasyon maskı şeklinde yani yer gerçeği şeklinde verilmiştir. Bütün görüntüler uzmanlar tarafından etiketlenmiş olup herhangi bir yanlışlık bulunmamaktadır. CVC-ClinicDB, CVC-ColonDB ve Etis-Larib veri setleri hem polip tespitinde hem de polip bölümlleme işlemlerinde kullanılmaktadır ve MICCAI polip yarışmasına göre kuralları mevcuttur. Bu veri setine ait bazı görüntüler Şekil 3.3'te gösterilmiştir.



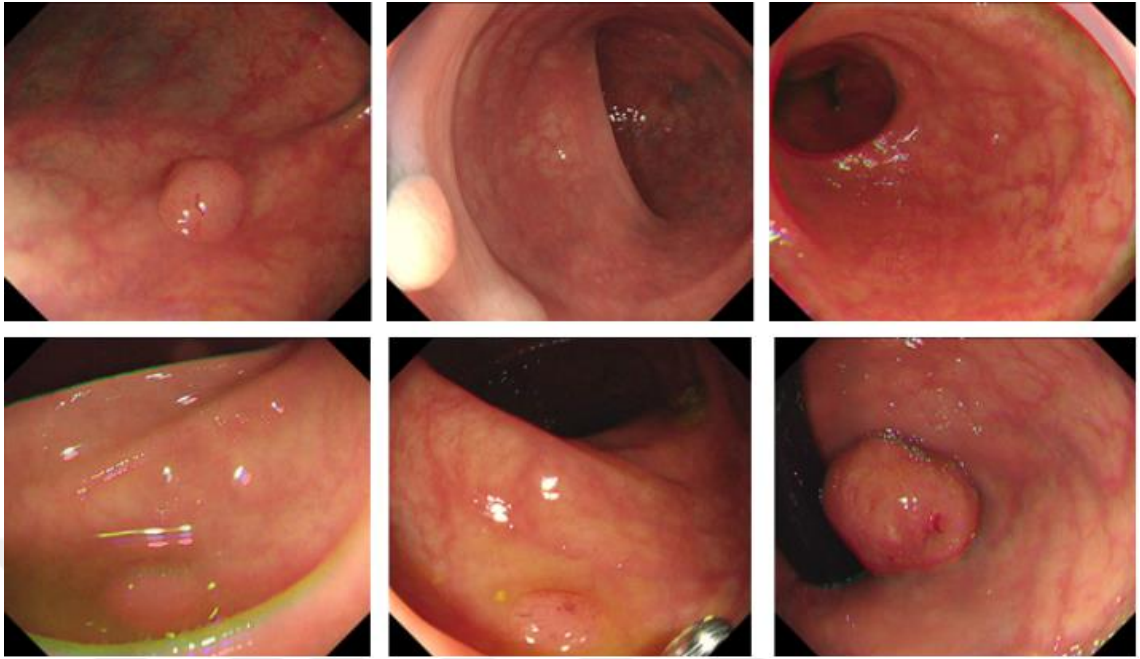
Şekil 3.3. Etis-Larib veri setine ait bazı görüntüler.

PICCOLO veri seti: 2020 yılında sunulan bu veri seti [114] aynı zamanda halka açık olarak sunulmuştur. Veri seti 40 hastadan alınmış ve toplamda 76 adet lezyon yani eşsiz polip içermektedir. Toplamda 3433 adet görüntüden oluşan bu veri setinde, eğitim için 2203 görüntü, doğrulama için 897 görüntü ayrılırken, test işlemleri için ise toplamda 333 adet görüntü ayrılmıştır. 3433 görüntünün 2131 tanesi beyaz ışık görüntüsü iken, 1302 adet görüntü ise dar bant görüntüsü üç içermektedir. Görüntü çözünürlüğü ise standart görüntüden yani 854x480 görüntü çözünürlüğü ile yüksek kalite görüntüye yani 1920x1080 görüntü çözünürlüğüne doğru değişmektedir. Ayrıca bu veri setinde toplam 43 adet negatif yani polip içermeyen görüntü mevcuttur. Bazı görüntülerde bir polip bulunmakla beraber, bazı görüntülerde ise birden fazla polip görüntüleri bulunmaktadır. Bu veri setinin yayınlanma amacı ise polip bölümlleme işlemlerinde diğer veri setlerine nazaran daha başarılı sonuçlar ürettiğini göstermektir. Fakat biz çalışmamızda yeteri kadar veri seti olmadığı için bu veri setinin de polip tespiti için kullandık. Etiketleme işlemlerinde ise bu veri seti CVC-ClinicDB ve Etis-Larib gibi veri setlerinin formatlarını içermektedir yani her bir polip resmine ait yer gerçeği (ground truth) bulunmaktadır. Veri setindeki polip görüntüleri birden fazla uzman hekim tarafından işaretlenmiştir. Veri seti çok yeni olduğu için bu konuda yapılmış birkaç çalışma mevcuttur [115]. Bu veri setine ait bazı görüntüler Şekil 3.4'te gösterilmiştir.



Şekil 3.4. PICCOLO veri setine ait bazı görüntüler.

SUN (Showa Üniversitesi ve Nagoya Üniversitesi) Kolonoskopi Video Veritabanı: otomatik kolorektal polip tespitinin değerlendirilmesi için kolonoskopi videolarından oluşan halka açık veri tabanıdır [116] . Bu veri tabanı 2020 yılında Misawa ve ark. tarafından önerilen metodun değerlendirilmesi için kullanılan veri tabanıdır. SUN veri tabanı, eşsiz 100 polip çerçevelerinden oluşmaktadır. Toplamda 49.136 polipli görüntüye sahip olan bu veri tabanı, 109.554 adet polip içermeyen arka plan görüntülerini taşımasıyla literatürde hem en büyük veri tabanı olmakla beraber hem de ilk kez negatif görüntülerini içeren halka açık ilk veri tabanıdır. Bu veri tabanı bünyesinde bulundurduğu toplam görüntü nedeniyle polip tespitinde en etkili veri tabanıdır. Veri tabanındaki polip görüntüleri birden fazla uzman tarafından etiketlenmiştir. Diğer veri setlerinden farklı olarak bu veri setindeki etiketlenmeler CSV dosyası şeklinde sınırlayıcı kutuların koordinatlarını içermektedir. Yani aslında polip tespiti için sunulmuş etkili bir veri setidir. 100 adet eşsiz polipten oluşan bu veri setinden çok yüksek miktarda görüntü bulunmasından dolayı veri artırma ihtiyacı azaltılmakla beraber, veriye aç derin öğrenme algoritmaları için eşsiz bir veri setidir denilebilir. Bu veri setine ait bazı görüntüler Şekil 3.5'te gösterilmiştir.

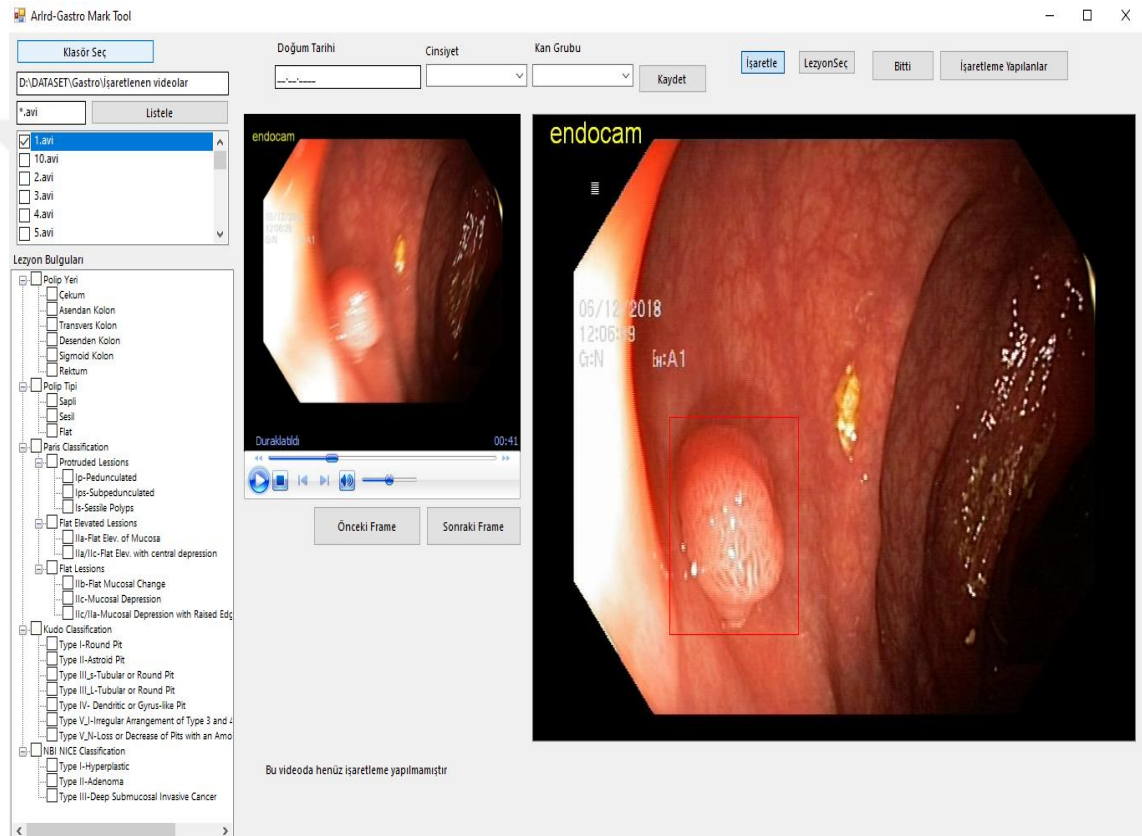


Şekil 3.5. SUN veri setine ait bazı görüntüler.

3.3. Önerilen Veri Seti

Otomatik polip tespiti uygulamalarında veri setinin etkinliği tartışılmazdır. Özellikle veriye aç derin öğrenme tabanlı polip tespiti algoritmalarının klinik seviyesinde bir sonuç vermesi için ise veri setindeki eşsiz polip sayısı ve görüntü sayısının önemi tartışılmaz bir ölçektir. Ayrıca, veri setindeki polip görüntülerin birkaç uzman gastroenterolog tarafından teyit edilmesi de aynı derecede öneme sahiptir. Özellikle SUN veri setinden önceki veri setlerinin etkinliği oldukça düşük olmakla beraber, klinik seviyede uygulama geliştirilmesi oldukça zor olmaktadır. Bu veri kısıtlamasının önüne geçmek için doktora çalışmaları süresince yerel ve daha özgün bir veri setinin oluşturulması amacıyla çalışmalar başlatılmıştır. Bilindiği üzere hastane veya üniversite ortamlarında hastaların özel bilgi ve verilerine ulaşmak için etik izinlerinin alınması ve hastaların önceden bilgilendirilmesi lazım. Bu konuda gerekli izinlerin alınması için uzman gastroenterolog ile beraber etik izin belgeleri alınmış ve buna binaen hastaların videoları izin dâhilinde kayıt altına alınmıştır. Bilindiği üzere, kolonoskopi taramalarında hasta anlık olarak incelendikten sonra kaydedilen video tekrar oynatılarak uzman tarafından yeniden incelenir ve buna bağlı olarak hastaya gerekli teşhis ve tedaviler uygulanır. Yıl bazında ortalama 100 adet polip içeren kolonoskopi videoları kayıt altına alınmıştır. Toplam da kayıt altına alınan video sayısı 300'ün üzerindedir.

Videolar daha önceden geliştirilen etiketleme uygulaması ile beraber derin öğrenme ortamına alınmaya başlanmıştır. Böylece, öncelikle geliştirilen uygulamanın özellikleri sırasıyla şu şekildedir. Uygulamada, kolonoskopi videoları kayıt altına altıktan sonra videolar, yavaş ya da hızlı şekillerde yürütülerek etiketlenmesine olanak sağlamıştır. Etiketleme programı polipteki lezyonun cinsini, polipin cinsini veya boyutu gibi tıbbi olarak gerekli olan bütün özellikleri içermektedir. Aşağıdaki şekilde polip etiketleme programına ait bilgiler bulunmaktadır.

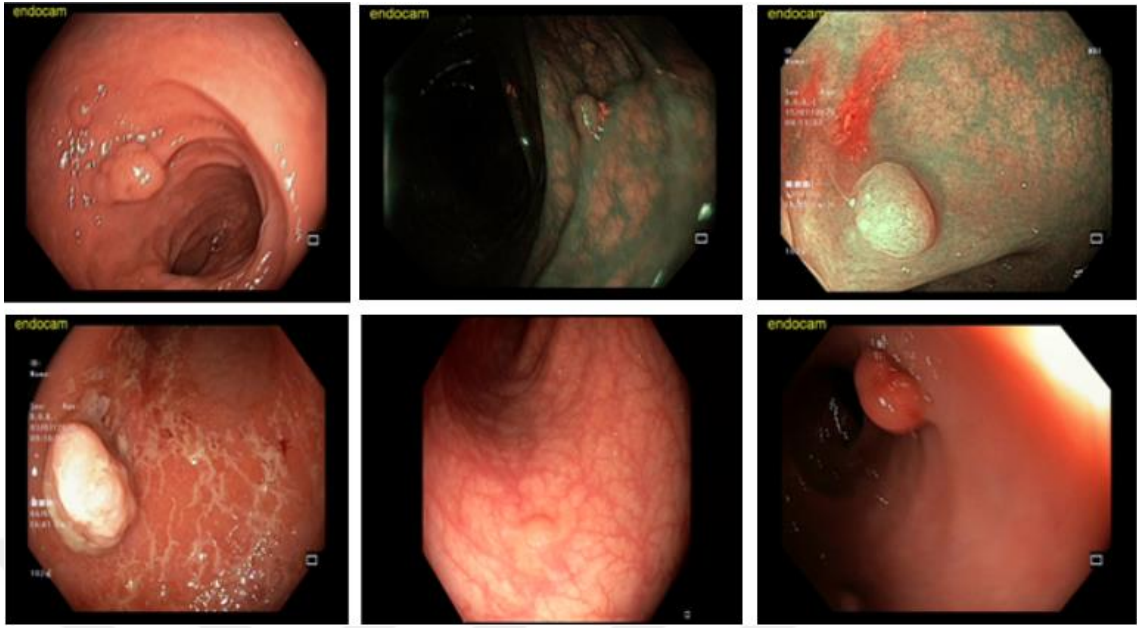


Şekil 3.6. Kolonoskopi görüntülerinin işaretlenmesi/etiketlenmesi için geliştirilmiş olan mevcut ara yüz.

Yukardaki şekilde de görüldüğü üzere, her bir polipe ait özellikler uygulamaya özellik olarak eklenerek, uzmanın gerek duyacağı bütün bilgiler bulunmaktadır. Burada, sırasıyla şu aşamalar izlenmektedir. Uzman gastroenterolog uygulamayı açtıktan sonra işaretlenmesi gereken videoyu açar. Daha önceden işaretlenme yapıldıysa ilgili çerçevenin nerde olduğu ve polipe ait bilgiler bulunmaktadır. İlk kez işaretlenme yapılacaksa istenilen şekilde video yürütülerek polipli yerde bekletilir. Daha sonra ise uzman gastroenterolog polip bulunan çerçeveyi sınırlayıcı kutu içeresine alır yani

koordinatlarını belirler. Bununla beraber bu polipli bölgenin birkaç saniye öncesi ve sonrası işaretlenerek polipli alanın farklı açılardan etiketlenmesi sağlanır. Böylece, bir videoda var olan bir polipten istenilen düzeyde (onlar ya da yüzler seviyesinde) polip çerçevesi yani görüntüsü alınarak derin öğrenme algoritmalarının ihtiyaç duyduğu yüksek sayıda etiketli görüntü elde edilmiş olmaktadır. Bundan sonraki süreçler ise daha çok bilgisayar bilimcilerini ilgilendiren verilerin belli bir düzene sokulması ve görüntü işleme yöntemleri ile gerekli görülen gürültülerin giderilmesi, verilerin artırılması için farklı tekniklerin uygulanmasıdır. En nihayetinde ise uzman gastroenterologlar elde edilen bütün görüntüleri (bilgisayar bilimcilerinden sonra) tekrar kontrol eder. Son olarak ise bilgisayar bilimcisi ise bu polipe ait temiz görüntülerden istenilen seviyede görüntüler üretir. Etiketlemeleri ve işaretlemeleri aynı anda yapabilen bu program sayesinde daha kısa sürede daha fazla görüntü üretilerek hem zamandan tasarruf hem de uzmandan yani maliyetten tasarruf sağlanmış olmaktadır.

Toplamda 300'den fazla kolonoskopi videolarından elde edilmiş özel veri setinde en nihayetinde şu kazanımlar elde edilmiştir. 400 adet eşsiz polip elde edilirken, her bir polipten en az 50 adet ve bazılarında ise video uzunluğuna bağlı olarak binler seviyesinde görüntüler elde edilmiştir. Toplamda 100.000 bine yakın polipli görüntü sayısı elde edilirken, 200.000 adette negatif yani polip içermeyen arka plan görüntüleri elde edilmiş, yani toplamda 300.000 adet görüntü elde edilmiştir. Bu veri seti bu haliyle, literatürde en fazla polipli görüntü içeren veri seti olmakla beraber aynı zamanda en çok negatif polip içeren veri seti olmaktadır. Böyle bir veri seti ile eğitilen bir derin öğrenme modeli bu sayede klinik araştırmalarda daha verimli bir şekilde kullanılabilir. Yapılan deneylerde de polip tespitinin başarımı diğer çalışmalardan oldukça ilerde olduğu teyit edilmiş ve klinikte gerekli izinler alınarak gerçek zamanlı klinik uygulamalarında kullanılarak başarımı güvenilir bir seviyede test edilmiştir. Veri setindeki görüntülerin büyük çoğunluğu yüksek kalitede iken (1920x1080 çözünürlüğünde), az bir kısmı ise standart kalitede (720x576 çözünürlüğünde) çözünürlüğe sahiptir. Bu özel veri setimize ait bazı görüntüler Şekil 3.7'de gösterilmiştir.



Şekil 3.7. Önerilen (özel veri seti) veri setine ait bazı görüntüler.

Yukardaki görüntülerden de görüldüğü üzere önerilen veri集中的 görüntü kaliteleri oldukça yüksek olmakla beraber hem beyaz ışık görüntüleri kullanılmış hem de dar bant görüntüleri kullanılmıştır. Bu iki görüntü formatlarının beraber kullanılması sayesinde otomatik polip tespiti modelinin daha çok polip tespiti yapmasına ve yanlış negatif ve pozitif oranının daha da düşürülmesine olanak sağlar. Böylece, daha sağlam ve gelişmiş bir polip tespit sistemi oluşturulmuş olmaktadır.

3.4. Değerlendirme Metrikleri

Makine öğrenimi algoritmalarının performansını ve kalitesini ölçmek için değerlendirme metrikleri kullanılır. Eğitilmiş bir derin öğrenme modelinin, test verisi yani yeni görünmeyen veriler üzerindeki başarımının ölçülmesi ancak efektif değerlendirme metrikleri kullanılarak anlaşılabilir. Bir modeli test etmek için birçok farklı değerlendirme ölçütü literatürde mevcuttur. Eğitimi tamamlanmış bir derin öğrenme modelin performansını değerlendirmek için birden çok değerlendirme metriği kullanılması birçok açıdan faydalı olur, çünkü bir model bir değerlendirme metriği kullanılarak iyi performans gösterebilirken aynı model başka bir değerlendirme metriği için düşük performans gösterebilir. Bu nedenle değerlendirme metriklerini doğru kullanmak, modelimizin başarılı olup olmadığını ve en iyi şekilde çalışıp çalışmadığını gösteren kritik bir faktördür. Bu bölümde, KRK için oluşturulan bilgisayar destekli

teşhis sistemi için kullanılan ve literatürde otomatik polip tespitinde kullanılan değerlendirme ölçütlerinin kısa bir açıklamasını ve bazı formüllerini sunuyoruz.

Değerlendirme metriklerinde en temel öğeler hata matrisi ya da karışıklık matrisindeki temel birimlerden oluşur. Makine öğrenmesinde kullanılan sınıflandırma modellerinin performansını değerlendirmek için hedef niteliğe ait tahminlerin ve gerçek değerlerin karşılaştırıldığı hata matrisi polip tespitinde de sıklıkla kullanılmaktadır. Her ne olursa olsun sınıflandırma tahminleri şu dört temel birimlerden birine sahip olmalıdır ve bunlar sırasıyla şu şekildedir. Bir model pozitif sınıfı doğru bir şekilde tahmin ediyorsa, Doğru Pozitif (TP) olur. Ancak, bir model pozitif sınıfı yanlış tahmin ederse, bu bir Yanlış Pozitif (FP) olur. Öte yandan, bir model negatif sınıfı doğru tahmin ediyorsa, bu Gerçek Negatiftir (TN). Ancak, bir model negatif sınıfı yanlış tahmin ederse, bu Yanlış Negatiftir (FN). Kanseri tespiti için bu değerleri aşağıdaki gibi bir örnekle netleştirebiliriz. Bir görüntü kanserli hücreler içeriyorsa ve model kanserli kısmı başarıyla sınıflandırıyorsa, bu TP'dir. Ama kanserli bir hücre olsa bile, model onu kanserli hücre olarak sınıflandırmıyorsa, o zaman bu FP'dir. Öte yandan, görüntüde kanserli hücre yoksa ve model kanserli hücre olmadığını sınıflandırırsa, bu TN'dir. Ancak kanserli hücre olmasa bile model kanserli hücre olarak sınıflandırıyorsa bu FN'dir. Otomatik polip tespiti problemi göz önüne alındığında bu değerler şu şekilde örneklendirilebilir. TP, öngörülen sınırlayıcı kutunun (bounding box) polipin temel gerçeğine düştüğünü (ground truth) gösterir. FP, öngörülen sınırlayıcı kutunun polipin temel gerçeğinin dışında kaldığını gösterir. FN, tahmin edilen bir sınırlayıcı kutu olmadığını ancak çerçevenin bir polip içerdiğini ve dolayısıyla polipin kaçırıldığını gösterir. Son olarak, TN, polipsiz görüntülerde polip tespit edilmediğini gösterir.

Kesinlik (precision): pozitif olarak tahmin edilen değerlerden kaçının gerçekte pozitif olduğunu gösterir. Bu nedenle, FP numarasının maliyetini çıkarmak için önemli bir metriktir ve aşağıdaki gibi formüle edilmiştir.

$$Precision = \frac{TP}{TP+FP} \quad (3.1)$$

Duyarlılık ya da hatırlama (recall, sensitivity): tüm pozitiflerden kaç gerçek pozitifin tahmin edildiğini gösterir. Bazen duyarlılık olarak adlandırılır ve aşağıdaki gibi formüle edilir.

$$Recall = \text{Sensitivity} = \frac{TP}{TP+FN} \quad (3.2)$$

F1 Puanı (F1-score): kesinlik ve duyarlılık metriklerinin harmonik ortalamasını temsil eder. Özellikle, kesinlik ve hatırlama arasında bir denge aranırken F1 puanı seçilmelidir ve aşağıdaki gibi formüle edilir.

$$F1 = \frac{2*Precision*Recall}{Precision+Recall} \quad (3.3)$$

Ortalama Kesinlik (mAP), tüm sınıfların Ortalama Hassasiyet (AP) değerlerinin ortalamasını alan popüler bir nesne algılama metriğidir. Böylece modelin performansı tek bir metrik ile ölçülebilir hale getirilebilir. Formüle edilmiş hali aşağıdaki gibidir.

$$mAP = \frac{\sum_{q=1}^Q AveP(q)}{Q} \quad (3.4)$$

Eşitlikte Q, kümedeki sorgu sayısıdır, q, ortalama kesinlik sorgusudur. Bu formül ile mAP'yi şu şekilde açıklayabiliriz: Verilen bir sorgu için AP değerini hesaplıyoruz ve ardından tüm bu AP puanlarının ortalaması bize tek bir sayı veren mAP değeri oluyor. Bu sayede model performansımızı tek bir sayı ile değerlendirebiliriz. Bu yöntem, nesne algılama algoritmalarında en çok kullanılan değerlendirme metriğidir. mAP 0.5 metriği IoU değerinin 0.5 değerine göre TP ve FP gibi hesaplamaları belirlerken, mAP 0.5:0.95 ise 0.5'ten başlayarak 0.05 artarak 0.95'e kadar olan değerlerin ortalamasının belirlenmesinde sorumludur ve bu metrik nesne tespit algoritmalarında daha değerli ve güvenli bir metriktir. Çalışmalarımızda bütün bu metrikleri kullandık.

Saniyedeki Kare Sayısı (FPS): algılama hızını ifade eder ve saniyede iletilen kare sayısını gösterir. Yüksek bir FPS değeriyle, saniyede daha fazla kare oluşur, bu nedenle görüntü efekti daha pürüzsüz ve net olur. Ek olarak, bir nesne algılama algoritmasının bir nesneyi gerçek zamanlı olarak algılaması için en az 30 FPS sağlaması veya çıkarım hızının 33 ms'den az olması gerekir. FPS ve çıkarım hızı ters orantılıdır.

4. BÖLÜM

GERÇEK ZAMANLI POLİP TESPİTİ İÇİN DERİN ÖĞRENME MODELLERİ

4.1. Önerilen CAD sistemi

Bu bölümde gerçek zamanlı polip tespiti için önerilen CAD sisteminin elemanlarından bahsedilecektir. Sistem en temel anlamda, veri seti birimi, eğitici birimi, test birimi ve CAD teşhis birimi olmak üzere 4 parçadan oluşmaktadır. Sırasıyla bu yapılarda gerçekleştirilen temel işlemler şu şekildedir. Veri seti birimi; uzman hekimler, veri seti ve görüntü ön işleme birimleri olmak üzere 3 ana kısımdan oluşmaktadır. Burada uzman hekimlerin yaptığı işlem kolonoskopi videolarında bulunan polipli görüntülerin koordinatlarının belirlenmesidir. Bu birim en az 3 uzman hekim tarafından oluşmaktadır. Bu hekimlerin belirlediği koordinatlardaki polipler diğer hekimler tarafından onaylandıktan sonra veri seti kısmına geçmektedir. Veri seti kısmı ise uzman hekimlerden gelen bütün görüntülerin bir araya toparlanması ve etiketlerinin düzenlenmesi olayıdır. Burada önemli olan polipli görüntülerin koordinatlarının YOLO ya da SSD gibi nesne tespit algoritmalarının kullandığı etiketlemeye uygun bir şekilde hazır bir etiket dosyasına dönüşmesidir. Nesne tespit algoritmaları etiketli verilere ihtiyaç duymaktadır. Her bir görüntü için poliplerin bulunduğu koordinatların gerçek ve doğru bir şekilde seçilmesi çok önemli bir yere sahiptir. Yanlış bir etikete sahip bir görüntü modelin başarımının düşürülmesini destekler.

Veri ön işlemi çeşitli görüntü işleme teknikleri ile görüntülerde var olan kusurların giderilmesi sağlanır. Buna ek olarak gerekli duyulması halinde veri artırma teknikleri ile veri seti hacminin artırılmasında kullanılabilir. Burada özel filtreleme teknikleri ile beraber mosaic, mixup, cutmix, flip gibi birçok yeni ve etkili veri artırma tekniklerinin beraber ya da sırayla kullanılması ile veri seti boyutunda ve hacminde ciddi bir artış

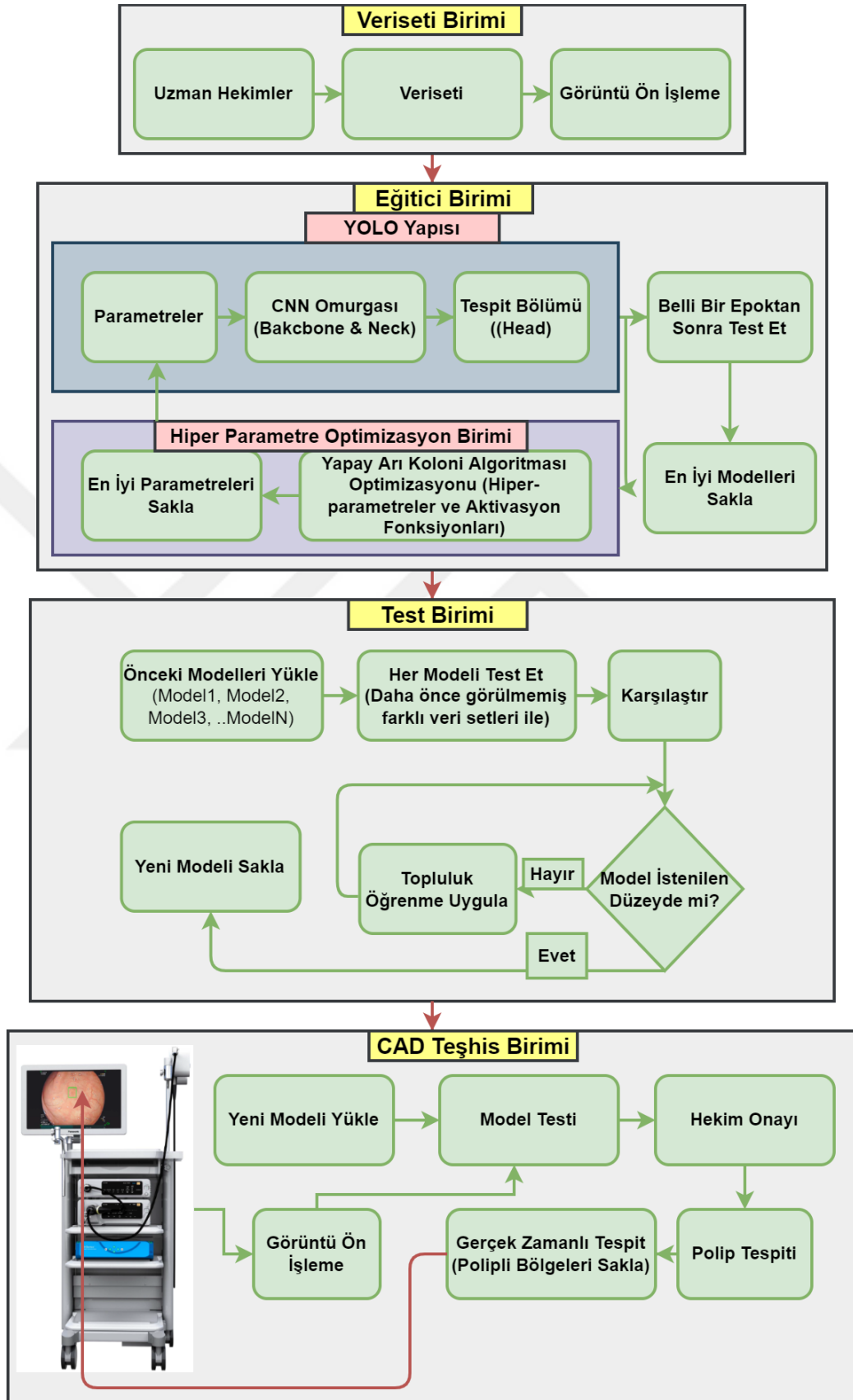
sağlanır. Özellikle küçük ölçekli veri setlerinin bu tür teknikleri kullanarak veriye aç derin öğrenme mimarilerinin veri azlığı konusuna bir çözüm olabilir.

Polip tespiti sisteminin diğer önemli bir parçasını oluşturan eğitici birimi nesne tespiti algoritması olan YOLO tabanlı algoritmalar ile beraber yapay arı koloni (ABC) algoritmasının kullanılmasıdır [117]. ABC algoritması sayesinde hiper-parametre optimizasyonunu ve aktivasyon fonksiyonlarının optimizasyonu esnek ve etkili bir şekilde yapılabilmektedir. Bazı YOLO algoritmalarında hiper-parametre optimizasyonu için tek popülasyonlu genetik algoritma [118] kullanılmaktadır. Bu iki algoritmanın polip tespitindeki başarıları ve karşılaştırmaları tartışma bölümünde verilecektir. Nesne tespit algoritması yeni özelliklerin keşfedilmesinde ve öğrenmenin en uygun düzeyde gerçekleşmesini sağlamaktadır. Bununla beraber gerçek zamanlı tespitin yapılması için ise tek aşmalı yaklaşımlardan olan YOLOv3 ve YOLOv4 algoritmaları seçilmiştir. Özellikle YOLOv4 son zamanların en popüler ve başarılı algoritması iken, YOLOv3 endüstride daha fazla kullanılan bir algoritmadır. Bu iki algoritma piyasadaki muadil algoritmalara nazaran hem gerçek zamanlı tespit yapıyor hem de doğruluk açısından diğer algoritmalarından oldukça ilerdedir.

Test birimi nesne tespit algoritmanın sonucunda elde edilmiş ağırlıklar ile en uygun ağırlık bulunmasını amaçlar. Eğitici birimde farklı hiper-parametreler için eğitilen ağırlıklar saklanır. Bu saklanan ağırlıklar, polip test verileri üzerinde test biriminde teste tabi tutularak bu modellerin genelleme kabiliyetleri ölçülür. Yapılan deneylerde genellikle en başarılı ilk 5 model saklanılır. Buradaki amaç test biriminde en uygun ağırlıklar bulunmakla beraber çok fazla karmaşıya girmeden modelin genelleme yeteneğine odaklanmayı amaçlamaktır. Zaten seçilen modeller daha sonra topluluk öğrenme sürecinden geçilerek başarımın daha da artırılması hedeflenir. Genellikle topluluk öğrenmede başarım artarken çok az durumda bazen hızdan feragat edilebilir ama bu kâr zarar hesabına göre ortak verilen bir karardır. Topluluk öğrenmedeki amaç kısaca şu şekilde açıklanabilir; birçok ağdan elde edilen bilgiler dâhilinde başarımı daha yüksek ve tek bir ağ gibi davranan bir modelin oluşturulmasıdır.

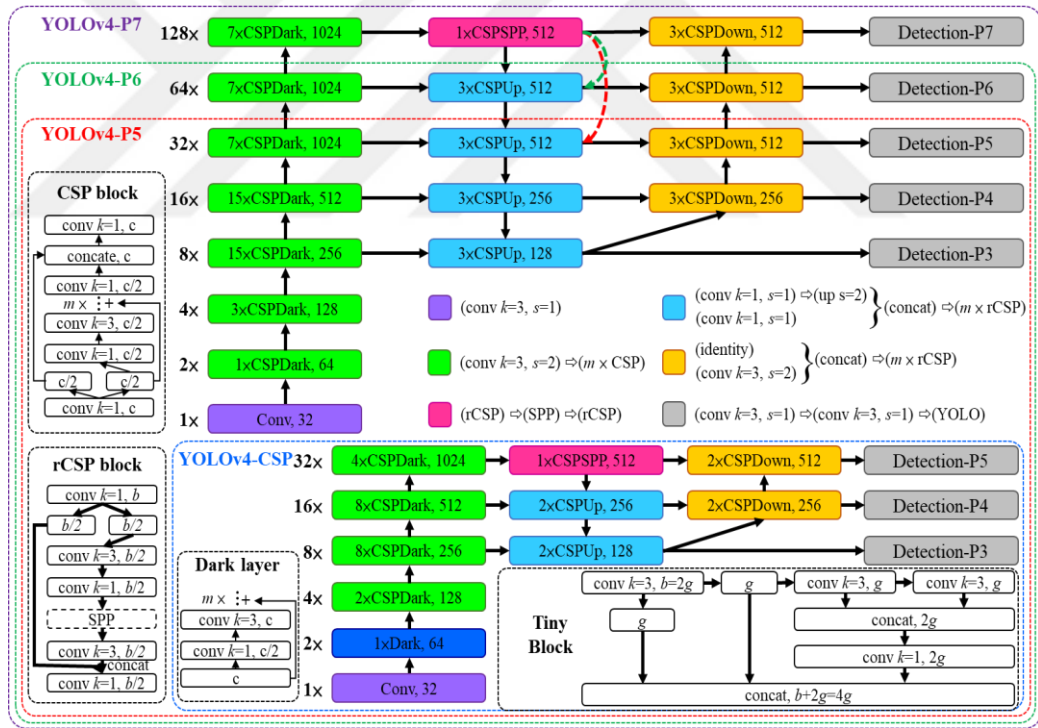
CAD teşhis biriminde elde edilen en iyi modeller hekimlerin onayına sunulur. Bunun için model bir PC yardımı ile kolonoskopi cihazına bağlanarak gerçek zamanlı olarak kullanılması sağlanır. Bazı durumlarda gerçek zamanlı uygulamaya geçilmeden önce

model kolonoskopi videoları üzerinde de test edilebilir. Böylece gerçek zamanlı uygulamada yaşanabilecek olumsuzluklar önceden giderilmiş olur. Kolonoskopi cihazına bir PC yardımı ile bağlanan bu yapı CAD teşhis birimini oluşturur. GPU tabanlı PC kolonoskopi cihazına görüntü yakalama kartı ya da kolonoskopi cihazında görüntü veren portlara da bağlanarak veriler PC'ye aktarılabilir. Yaptığımız çalışmada bir video yakalama kartı ile görüntüler PC'ye aktarıldıktan sonra tekrar kolonoskopi monitörüne gelerek tek bir ekran üzerinde daha optimize bir sistem elde edilmiştir. Böylece gastroenteroloğun daha iyi bir gözlem yapmasına ve CAD biriminden daha fazla fayda sağlamasına olanak sunmuştur. Diğer taraftan, yapılan işlemlerin eş zamanlı bir şekilde çalışması gerektir. Bunun için ise son teknolojik GPU kartlarını seçilmesi önem arz etmektedir. Sistemin başarısı uzman hekimler tarafından onaylandıktan sonra, CAD birimi hem polip tespiti yapar hem de tespit edilen polip görüntülerinin ilgili hastaya göre dosyalandırmasını ve kaydedilmesini sağlamaktadır. Böylece, daha sistematik bir şekilde hem tespit hem de kayıt işlemi gerçekleştirilmiş olmaktadır. Önerilen otomatik polip tespiti sisteminin akış şemasıyla beraber temel diyagramı Şekil 4.1'de gösterilmiştir.



Şekil 4.1. Önerilen otomatik polip tespiti sisteminin akış şeması ve diyagramı.

Şekil 4.1 göz önüne alındığında, tüm sistemin modüler ve esnek bir yapıda olmasından dolayı, herhangi bir medikal görüntü işleme sistemine basit bir şekilde entegre edilebileceğini göstermektedir. Burada tek değiştirilecek olan yapı sadece veri seti birimidir ve hangi alana uygulanacaksa o alandaki uzman hekimlerin değişikliğidir. Sistem donanımsal ve yazılımsal olarak herhangi bir ciddi değişikliği ihtiyaç hissetmemektedir. Ayrıca, tüm sistem için kullanılan nesne tespit algoritması olan Scaled-YOLOv4 ölçeklenebilir olması nedeniyle de sistem daha esnek bir hal almaktadır. Burada sistem, küçük veri setinden büyük veri setine kadar, düşük seviye GPU donanımında yüksek seviye GPU donanımlarını destekleyebilmektedir. Bu da ölçeklenebilir YOLOv4 mimarisinin sistemimize göre yeniden tasarlanmasıyla gerçekleşebilmektedir. Sistemimizde kullanılan Scaled-YOLOv4 nesne tespit algoritmasına ait yapı Şekil 4.2’de gösterilmektedir.



Şekil 4.2. Scaled-YOLOv4 nesne tespit algoritmalarına ait modellerin tamamını içeren temel yapı.

4.2. Kayıp Fonksiyonları

YOLO algoritmalarında ızgara hücresi başına, birden çok sınırlayıcı kutu öngörür. Kaybı gerçek pozitifler için hesaplamak, yapıdan yalnızca bir kısmı sorumludur. Bu

amaçla, zemin gerçeği ile en yüksek IoU'ya sahip olanı seçilir. Bu strateji ile sınırlayıcı kutu tahminleri arasında daha iyi çözümler gerçekleşir. Her tahmin, belirli boyutları ve en boy oranlarını tahmin etmede daha iyi hale gelir. YOLO algoritmalarında, kaybı hesaplamak için tahminler ile temel gerçek arasındaki toplam-kare hatasını kullanır. Kayıp fonksiyonu toplamda üç kayıp fonksiyonun toplamından oluşmaktadır. Bu fonksiyonlar sırasıyla sınıflandırma kaybı, yerleştirme kaybı yani tahmin edilen sınır kutusu ve temel gerçek arasındaki hatalar ve güven kaybı yani kutunun nesnelliğinden oluşmaktadır. Sırasıyla bu kayıpları açıklamaya çalışalım. Bir nesne algılanırsa, her hücredeki sınıflandırma kaybı, her bir sınıf için sınıf koşullu olasılıklarının karesi alınmış hatasıdır. Yerleştirme kaybı, tahmin edilen sınır kutusu konumları ve boyutlarındaki hataları ölçer. Sadece nesneyi algılamaktan sorumlu kutuyu sayılır. Önerdiğimiz modellerde en popüler üç yerleştirme kayıp fonksiyonlarından olan GIoU [119] , DIoU [109] ve CIoU [109] fonksiyonlarını kullanarak polip tespiti için en ideal kayıp fonksiyonlarını bulmayı gözlemledik. Nesne algılama algoritmalarında kayıp işlevi, nesne yerleştirme, nesne sınıflandırma ve nesne güven kaybı işlevlerinin toplamından oluşur. Çoğu nesne algılama algoritması, tahmin edilen kutu ile yer hedef kutusu arasındaki örtüşme derecesini belirlemek için IoU kullanır. Kayıp fonksiyonu şu şekilde formüle edilir:

$$IoU = \frac{|B \cap B^{gt}|}{|B \cup B^{gt}|} \quad (4.1)$$

$$Loss_{IoU} = 1 - IoU \quad (4.2)$$

Burada B^{gt} hedef kutu ve B tahmin edilen kutudur. IoU kaybı, yalnızca sınırlayıcı kutular çakışırsa çalışır. Örtüşme olmadan gradyan sağlanmaz. Önceki sürümlerin aksine, YOLOv4, nesne yerleştirme kaybı işlevi için IoU'nun geliştirilmiş bir sürümü olan CIOU algoritmasını kullanır. CIOU, hedef kutular ile tahmin edilen kutular arasındaki mesafeyi dikkate alır ve ayrıca sınırlayıcı kutu regresyonunda en boy oranı gibi geometrik faktörler sunarak sınırlayıcı kutular için en boy oranlarının tutarlılığını dayatır. Böylece, CIOU, hızlı yakınsama hızına ve sınırlayıcı kutu regresyonunda daha iyi performansa sahip bir kayıp fonksiyonudur ve aşağıdaki gibi formüle edilmiştir:

$$Loss_{CIOU} = 1 - IoU + \frac{p^2(b, b^{gt})}{c^2} + av \quad (4.3)$$

$$v = \frac{4}{\pi^2} \quad (4.4)$$

$$a = \frac{v}{(1-IoU)+v} \quad (4.5)$$

Eşitlikte b ve b^{gt} , B ve B^{gt} 'nin merkez noktalarını gösterir, $p(\cdot)$ ise öklid mesafesi ve c iki kutuyu kaplayan en küçük kutunun köşegen uzunluğudur. Burada α pozitif bir değişken parametredir ve v en boy oranının tutarlılığını ölçer. Polip tespit sistemimizde en iyi doğruluğu elde etmek için CIoU, DIoU ve GIoU kayıp fonksiyonlarını kullandık. DIoU, hedef sınırlayıcı kutu ile tahmin edilen sınırlayıcı kutular arasındaki mesafeyi en aza indirerek daha iyi doğruluk sunar, böylece özellikle örtüşmeyen senaryoda daha hızlı yakınsar.

$$Loss_{DIoU} = 1 - IoU + \frac{p^2(b, b^{gt})}{c^2} \quad (4.6)$$

GIoU kayıp fonksiyonu ise IoU kayıp fonksiyonu ($1-IoU$) ile birlikte bir ceza terimine sahip bir kayıp fonksiyonudur; burada C , B (tahmin edilen sınırlama kutusu) ve B^{gt} (temel gerçeği sınırlama kutusu) kapsayan en küçük kutudur.

$$Loss_{GIoU} = 1 - IoU + \frac{|C - BUB^{gt}|}{|C|} \quad (4.7)$$

Ceza teriminin getirilmesi nedeniyle, GIoU kaybı, hedef kutu ile örtüşene kadar tahmin edilen kutunun boyutunu genişletmeye devam eder ve daha sonra IoU terimi, sınırlayıcı kutunun örtüşme alanını en üst düzeye çıkarmak için çalışacaktır. DIoU işlevi, NMS ile kullanıldığında IoU'dan daha başarılı sonuçlar sunar. DIoU, bastırma kriterlerinde iki kutu arasındaki merkez nokta mesafesini hesaba katarken, IoU yalnızca örtüşme alanını hesaba katar. Yapılan deneylerde, DIoU-NMS ile birlikte DIoU kayıp fonksiyonunu kullanarak daha iyi polip tespiti sağladığı görülmüştür ve bunların geniş ve ayrıntılı sonuçları tartışma bölümünde sunulmuştur.

4.3. Aktivasyon Fonksiyonları

Aktivasyon süreci, sinir ağlarının çıktısını belirlemek için kullanılan ve performansı ciddi şekilde etkileyen sinir ağının bir parçasıdır. Eğitimi ve sinir ağının etkinleştirilip etkinleştirilmeyeceğini daha iyi karakterize etmek için kullanılır. Diğer bir deyişle

aktivasyon fonksiyonları, bir nöronun ateşlenip ateşlenmeyeceğine karar veren ve sinir ağlarına bir tür doğrusal olmamayı eklemeyi amaçlayan fonksiyonlardır. Bu işlevler, girdilerin ve biasların ağırlıklı toplamını hesaplamaktan sorumludur ve ayrıca derin sinir ağları için girdi verileri ile çıktı verileri arasındaki doğrusal olmayan eşleştirmede önemli bir rol oynar. Aktivasyon işlevleri olmayan bir sinir ağı, yalnızca x girişlerinden y çıkışlarına kadar doğrusal eşlemeler gerçekleştirebilir. Böylece matematiksel olarak karmaşık eşleşmeler oluşmayacak ve ağa istenilen görevler istenildiği gibi yürütülmeyecektir. Bu tez çalışmasında kullanılan bazı önemli aktivasyon fonksiyonlarının genel ve kısa bir tanımını veriyoruz.

Sigmoid işlevi, karakteristik S şeklinde bir eğriye sahip matematiksel bir işlevdir. Lojistik fonksiyon, hiperbolik tanjant gibi fonksiyonlar sigmoid fonksiyonlara dâhildir. Sigmoid aktivasyon fonksiyonu, doğası gereği doğrusal değildir ve düzgün bir türevi vardır. Çıktı olarak olasılığı tahmin etmek için özellikle modeller için kullanılır. Bu işlev, $(-\infty; +\infty)$ giriş aralığını $[0; 1]$, aşağıdaki gibi formülize eder.

$$\text{sigmoid}(x) = \frac{1}{1 + e^{-x}} \quad (4.8)$$

ReLU (Rectified Linear Unit), derin öğrenmede en çok kullanılan ve başarılı aktivasyon fonksiyonlarından biridir. Basit bir yapıya sahip olmasına rağmen lojistik sigmoid ve hiperbolik tanjant gibi diğer aktivasyon fonksiyonlarından daha iyi performans gösterir. ReLU, pozitifse girişi doğrudan çıkaran, aksi takdirde sıfır döndüren parçalı doğrusal bir işlevdir.

$$\text{ReLU}(x) = \max(0, x) \quad (4.9)$$

Orijinal YOLOv4'te algoritmasının CSPDarknet53 omurga kısmında Mish [66] aktivasyon fonksiyonu kullanılırken, boyun ve baş kısımlarında Leaky ReLU kullanılmaktadır. YOLOv4-CSP'de ise Leaky ReLU aktivasyon fonksiyonu yerine sadece Mish aktivasyon fonksiyonu kullanılmaktadır. Leaky ReLU aktivasyon fonksiyonu, CNN'lerde en çok kullanılan aktivasyon fonksiyonlarından biridir. Yeni Alpha parametresi ile ReLU'daki ölü nöron problemini çözer ve gradyanların sıfır olması engellenir ve şu şekilde formüle edilmiştir:

$$f(x) = \begin{cases} x & \text{for } x \geq 0 \\ ax & \text{for } x < 0 \end{cases} \quad (4.10)$$

Swish aktivasyon fonksiyonu [67], görüntü sınıflandırma gibi çeşitli zorlu alanlar için geçerli olan pürüzsüz, monoton olmayan bir işlevdir ve daha derin modellerde ReLU'dan daha iyi performans gösterir. Swish'in basitliği ve ReLU'ya benzerliği, sinir ağlarında Swish birimlerinin RELU için değişimini kolaylaştırır. Swish aktivasyonu, modelleme ve optimizasyona yardımcı olan ve derin ağlarda daha yüksek performans sağladığı RELU'dan daha kullanışlı olan bir pürüzsüzlük özelliğine sahiptir ve şu şekilde formüle edilir:

$$f(x) = \frac{x}{1+e^{-x}} \quad (4.11)$$

Mish, Swish işlevine benzer, tanımlanabilir, pürüzsüz, monoton olmayan bir etkinleştirme işlevidir. Mish yapısındaki pürüzsüz, monoton olmayan, limitsiz alt ve üst limitli özellikler antrenman performansı için önemlidir. Böylece, Mish aktivasyon fonksiyonunun sağladığı düzgünlük, ağırlık genelleme kabiliyetini ve sınıflandırma doğruluğunu arttırdığından, ancak daha uzun eğitim süresi ve hafıza maliyetine sahip olduğundan, Mish daha iyi tahminler yapabilir. Şu şekilde formüle edilmiştir:

$$f(x) = x * \tanh(\ln(1 + e^x)) \quad (4.12)$$

Diğer aktivasyon fonksiyonu ise Sigmoid Ağırlıklı Doğrusal Birimler (SiLU) aktivasyon fonksiyonudur [68]. SiLU, pekiştirmeli öğrenme (RL) tabanlı bir yaklaşım işlevi olmasına rağmen, son zamanlarda derin sinir ağlarında popüler hale gelmiştir. SiLU işlevi aslında Sigmoid'in girdisi ile çarpılmasıyla hesaplanır ve şu şekilde formüle edilir:

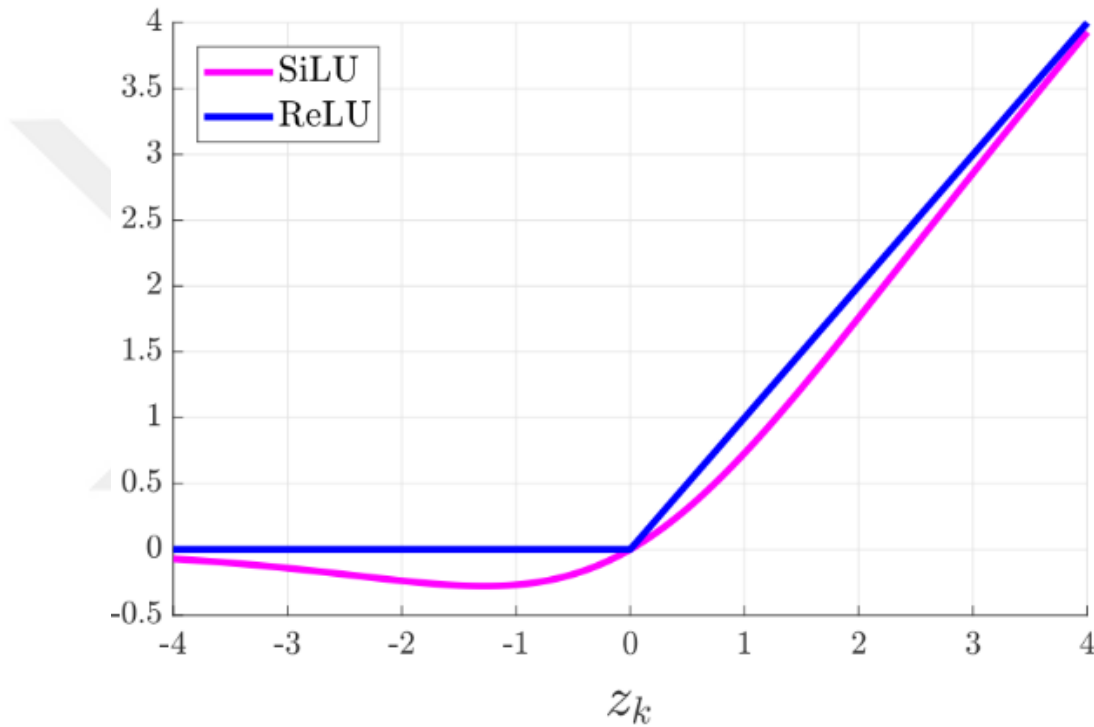
$$a_k(s) = z_k \sigma(z_k) \quad (4.13)$$

SiLU (a_k), s girdi vektörüdür, z_k gizli birimlerin k'ye girdisidir. Gizli katmanlara giriş ise şu şekilde formüle edilir:

$$z_k = \sum_i w_{ik} s_i + b_k \quad (4.14)$$

Eşitlikte b_k sapma ve w_{ik} gizli k birimlerine bağlanan ağırlıktır.

Büyük veri setleri için önerilen modellerde SiLU aktivasyon fonksiyonu kullanıldı. Tüm ağda YOLOv3 ve YOLOv4'ün önceden var olan Leaky ReLU ve Mish aktivasyon işlevleri SiLU ile değiştirildi. SiLU aktivasyon fonksiyonunun yazarı orijinal makalede SiLU'nun ReLU aktivasyon fonksiyonundan daha iyi performans gösterdiğini bildirmiştir [68]. Bu nedenle, özellikle büyük veri setlerindeki polip algılama performansını artırmak için tüm aktivasyon fonksiyonları SiLU ile değiştirildi. SiLU aktivasyon fonksiyonunun grafiği Şekil 4.3'te gösterilmektedir.



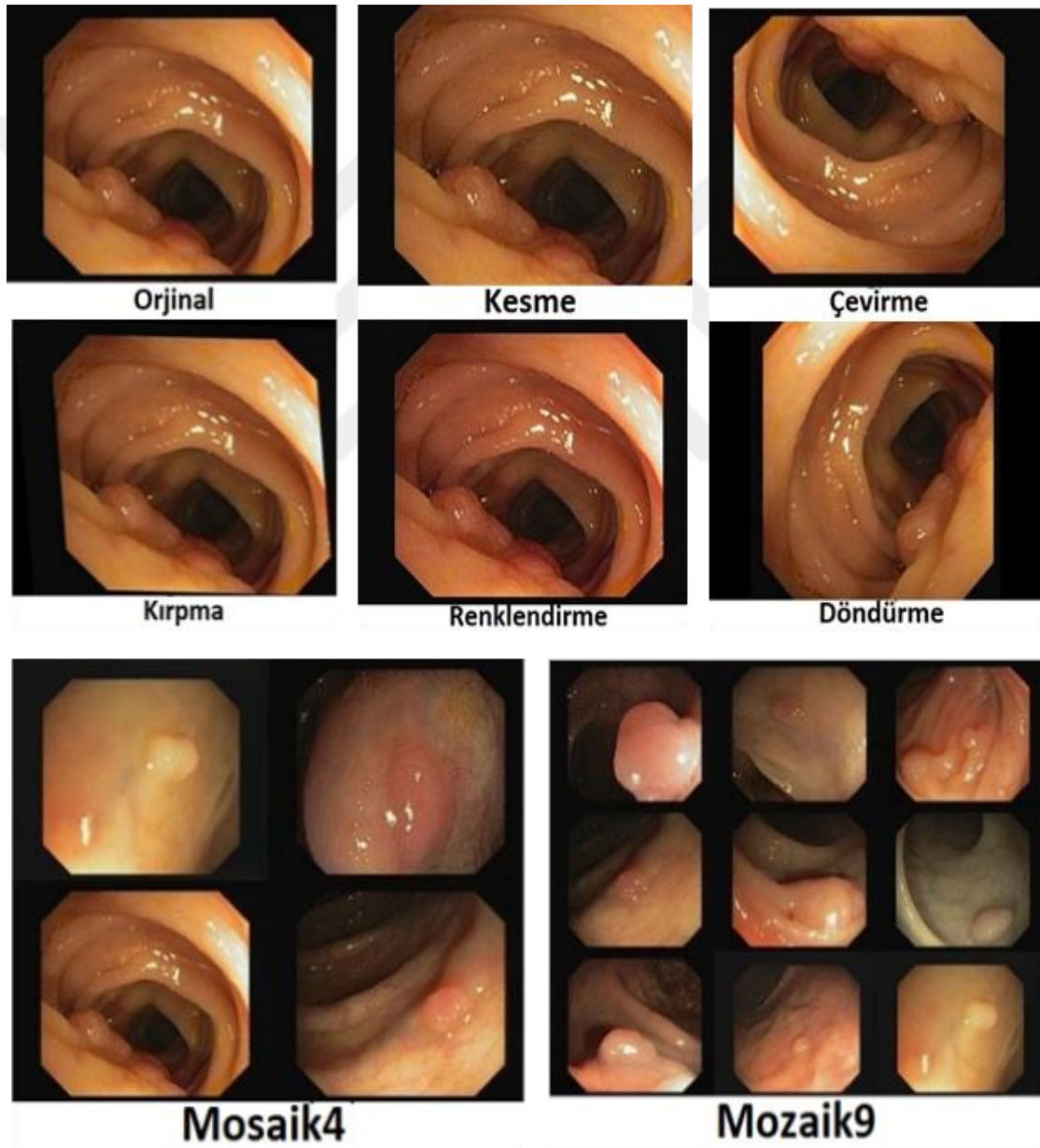
Şekil 4.3. SiLU ve ReLU'nun aktivasyon fonksiyonlarının davranışları.

Polip tespit sisteminde, modeli verimli bir şekilde eğitmek ve özellikle küçük veri setlerinde tahmini daha doğru hale getirmek için, YOLOv4'ün omurga, boyun ve baş kısımlarında Mish ve Swish aktivasyon fonksiyonlarının kullanımını genişletmeyi önerdik. Tüm yapıda CBL bloğu yerine CBM veya CBS kullanıyoruz. Deneylerimizde, küçük veri setlerindeki polip tespit sistemi için Mish aktivasyon fonksiyonu en iyi olurken, büyük veri setlerinde ise tüm yapıda CBL ve CBM blokları yerine CBS bloklarını yani SiLU aktivasyon fonksiyonunu içeren birimleri kullanıyoruz. Yapılan deneylerde büyük veri setlerinde en iyi sonuç veren aktivasyon fonksiyonunun SiLU olduğu gözlemlenmiştir. Bu yapıların sonuçlarını içeren detaylı karşılaştırmalar tartışma bölümünde sunulmuştur.

4.4. Veri Artırma Teknikleri

Veriye aç derin öğrenme algoritmaları için yeterli bir veri seti ve veri büyütme teknikleri çok önemlidir. Örneğin, 612 görüntüden 31 benzersiz polipten oluşan eğitim için CVC-ClinicDB ele alalım. Bu veri seti tek başına eğitim için yetersiz olduğundan, modelin test verileri üzerinde genelleme kabiliyetini arttırmak için birçok farklı veri büyütme tekniği kullanılır. Veri artırma tekniklerini eğitim öncesi veya eğitim sırasında olmak üzere iki ana kategoriye ayrıldı. Eğitim öncesi veri artırmanın amacı küçük polipleri tespit etmek ve duyarlılık değerini arttırmaktır. Ayrıca CVC-ClinicDB veri setini iki ana gruba ayrılması ile daha detaylı bir veri artırma tekniği geliştirildi. CVC-ClinicDB veri seti 325 küçük polipten oluşur ve geri kalanı büyük ve orta boy poliplerdir. Küçük polipler için yakınlaştırma tekniğini (%5–25) ve büyük polipler için uzaklaştırma tekniğini (%5–25) kullanılarak veriler dengelenmeye çalışıldı. Ek olarak, küçük poliplerin sağlam bir şekilde tespiti için yeni, basit bir veri büyütme tekniği geliştirildi. Bu teknik, görüntüleri ve etiketlerini (325 polip) kopyalayıp yapıştırarak küçük poliplerin sayısını artırır ve eğitim sırasında ağ tarafından daha iyi öğrenilmelerini sağlar. Böylece model daha iyi genelleme yeteneğine sahiptir ve küçük polipleri daha iyi tespit eder. Bir diğer önemli teknik, orijinal YOLOv4 makalesinde önerilen mozaik veri büyütmedir. Bu teknik, 4 eğitim görüntüsünü belirli oranlarda tek bir görüntüde birleştirir. Buna ek olarak, 1 görüntü elde etmek için 9 görüntünün karıştırılmasını içeren mozaik9'u da kullanıldı. Mozaik veri büyütme, modelin nesneleri normalden daha küçük bir ölçekte tanımlamasına ve çerçevenin farklı bölümlerinde farklı görüntüleri lokalize etmesine olanak tanır. Ayrıca küçük nesnelerin daha iyi algılanmasına da katkıda bulunur. Mozaik4, mozaik9, kesme, ölçekleme ve Cutmix veri artırma tekniklerini bir arada kullanarak, normal bağlamlarının dışındaki nesnelerin algılanmasının daha etkili olması sağlandı. Ek olarak, görüntünün nispeten yüksek ve düşük değerleri arasındaki kontrastı artıran histogram eşitleme tekniğini kullanarak yeni görüntüler elde edildi. Eğitim sırasında ayrıca mozaik, cutmix, kırpma, ölçekleme, aynalama (yukarı-aşağı, sol-sağ), karıştırma, döndürme, kırpma, doygunluk, ton, renk değeri, parlaklık, çeviri, pozlama, bulanıklık ve rastgele gürültü gibi birçok veri artırma tekniğini kullanıldı ve değerlendirildi. Tüm bu veri artırma tekniklerini kolonoskopi sırasında oluşabilecek gürültü, açı değişikliği ve renklerden kaynaklanan sorunları göz önünde bulundurularak seçildi. Daha sonra, optimizasyon bölümünde açıklanan ABC

algoritması yardımıyla hiper-parametre optimizasyonu seçilerek her bir veri geliştirme tekniğinin en optimal değeri bulunuldu. Eğitim sonrasında kullanılan veri büyütme tekniklerinin en önemli katkılarından biri, önceden belirlenmiş olasılık değerlerine göre eğitim sırasında her görüntünün yalnızca bir kez kullanılmasıdır. Bu sayede eğitim süresi kısaltmakta ve veri seti rastgele hale getirilerek daha iyi sonuçlar elde edilebilmektedir. Şekil 4.4 bu çalışmada kullanılan bazı veri artırma tekniklerini göstermektedir.



Şekil 4.4. Çalışmada kullanılmış bazı veri artırma tekniklerinin polip görüntüsü üzerindeki etkileri.

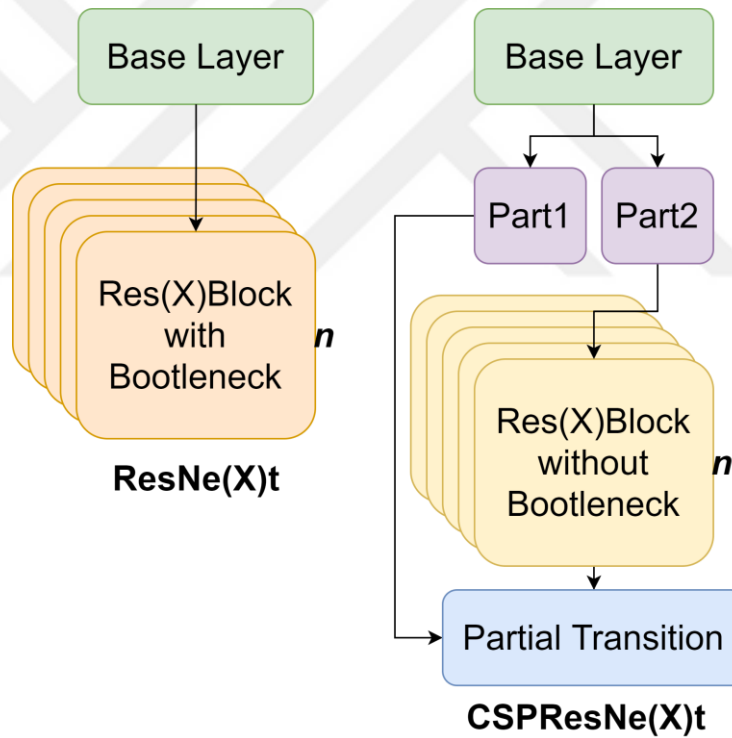
4.5. Uygulama Detayları

Bu çalışmadaki tüm deneyler, önceden yüklenmiş Ubuntu 20.04 ile bir Linux makinesinde yapıldı. Derin öğrenme modellerini eğitmek ve test etmek için kullanılan bilgisayarın özellikleri sırasıyla şu şekildedir. Intel Core i9 9900X (10 çekirdekli 3.50 GHz, 19.25 MB Intel® Smart Cache) işlemci, 32 GB DDR4 RAM ve tekli RTX 2080TI (11 GB GDDR6) 4352 cuda çekirdekli gibi donanımlar kullanmıştır. Buna ek olarak bazı deneysel çalışmalarda ise RTX 8000 (4.608 CUDA çekirdeği ve 576 NVIDIA Tensör Çekirdeği ile 48 GB GDDR6) grafik kartı kullanılmıştır. Ayrıca, NVIDIA CUDA Toolkit 11.5 ve NVIDIA GPU-Hızlandırılmış Kitaplık (cuDNN) 8.3 ile PyTorch ve DarkNet çerçevelerinin en son kararlı sürümü kullanılarak deneyler yapılmıştır. Eğitim sürecinde yinelemeli parametre güncellemeleri için SGD yerine Adam optimize ediciyi kullandık. Adam, eğitimin ilk aşamalarında genelleştirme kriterlerinde SGD optimize ediciden daha iyi performans gösterir, ancak performansı zamanla sabitlenir. Deneylerde, Adam seçildi çünkü SGD, öğrenme oranı düşük olduğunda normal gradyan inişi ile benzer sonuçlar veriyor ve bu gözlemler deneylerde de teyit edildi. YOLO algoritmalarındaki hiper-parametreler için ise GA ve ABC algoritmaları kullanıldı.

4.6. Küçük Veri Setleri için Önerilen Modeller

Bu bölümde YOLOv4 ve Scaled-YOLOv4 tabanlı önerilen modellerden bahsedilecektir. Bilindiği üzere nesne tespit algoritmaları, farklı görüntü çözünürlükleri ve cihazlar için ölçeklenebilir. EfficientDet [120] makalesindeki en son bulgulardan biri, nesne algılama modellerinin nasıl yukarı ve aşağı ölçeklendirileceğini düşünmektir. Ölçeklendirme farklı yöntemlerle gerçekleştirilebilirken, en yaygın ölçekleme yöntemi, omurganın derinliğini ve genişliğini değiştirmek veya tüm yapıyı değiştirmektir. Omurga genişliğinde yapılan değişiklikler CNN'deki evrişim filtrelerinin sayısındaki azalma veya artış ile ilgilidir. Öte yandan, omurganın derinliğinde yapılan değişiklikler, CNN'deki evrişim katmanlarının sayısındaki değişiklik ile ilgilidir. ResNet, ResNeXt ve DarkNet için temel ölçekleme yöntemlerini kullanırken, hesaplama maliyetinde kare, doğrusal ve kare artışlar vardır. Hesaplama maliyeti arttıkça, model daha iyi doğruluk sunabilir ancak gerçek zamanlı uygulamalar için ideal olmayabilir ve daha güçlü donanım gerektirir. Model ölçeklemede bu sorunun çözümü YOLOv4'ün omurgası olan CSPNet ile daha esnek bir şekilde yapılabilir. Çünkü CSPNet, hesaplama maliyetini

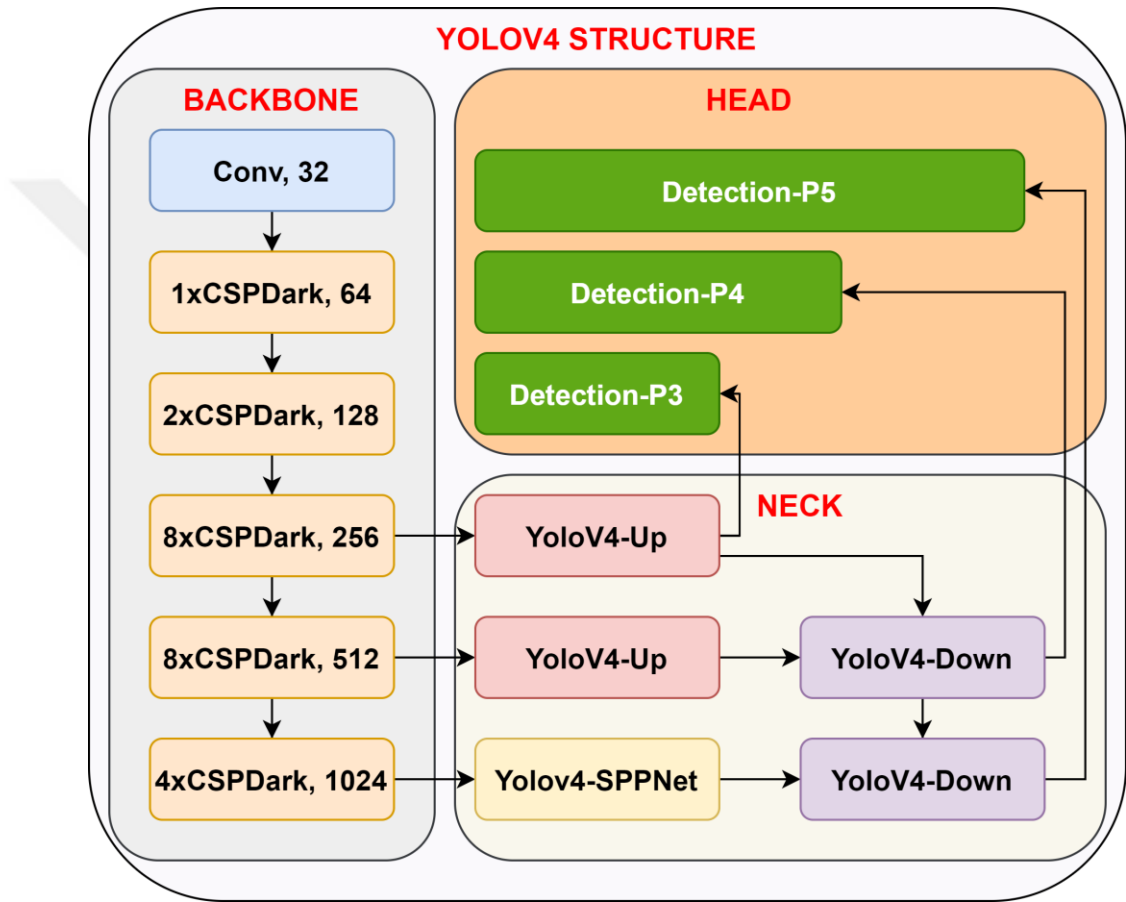
düşürür, doğruluğu artırır ve çıkarım süresini azaltır, böylece modeli gerçek zamanlı uygulamalar için daha avantajlı hale getirir. Yazarlar, YOLOv4'teki bu iyileştirmeleri Scaled YOLOv4 adlı makalelerinde sunmuşlardır. Örneğin, CNN'leri CSPNet'e dönüştürdükten sonra ResNet, ResNeXt ve DarkNet'teki hesaplama miktarı sırasıyla %23,5, %46,7 ve %50,0 azalmıştır. CSPNet, hesaplama maliyetini düşürme, doğruluğu artırma, ayrıca çıkarım süresini azaltma ve modeli gerçek zamanlı uygulamalar için daha başarılı hale getirme gibi çeşitli avantajlara sahip olmasıyla YOLO mimarisini çok daha faydalı bir nesne tespit algoritması yapmaktadır. Gerçek zamanlı polip tespiti için önerdiğimiz YOLOv4 modellerinin performansını CSPNet kullanarak bazı yapısal değişiklikler uygulayarak iyileştirdik. Şekil 4.5 ResNet için CSPNet'nin nasıl uygulandığını göstermektedir.



Şekil 4.5. CSPNet'in ResNet ağına uygulanmasını gösteren diyagram.

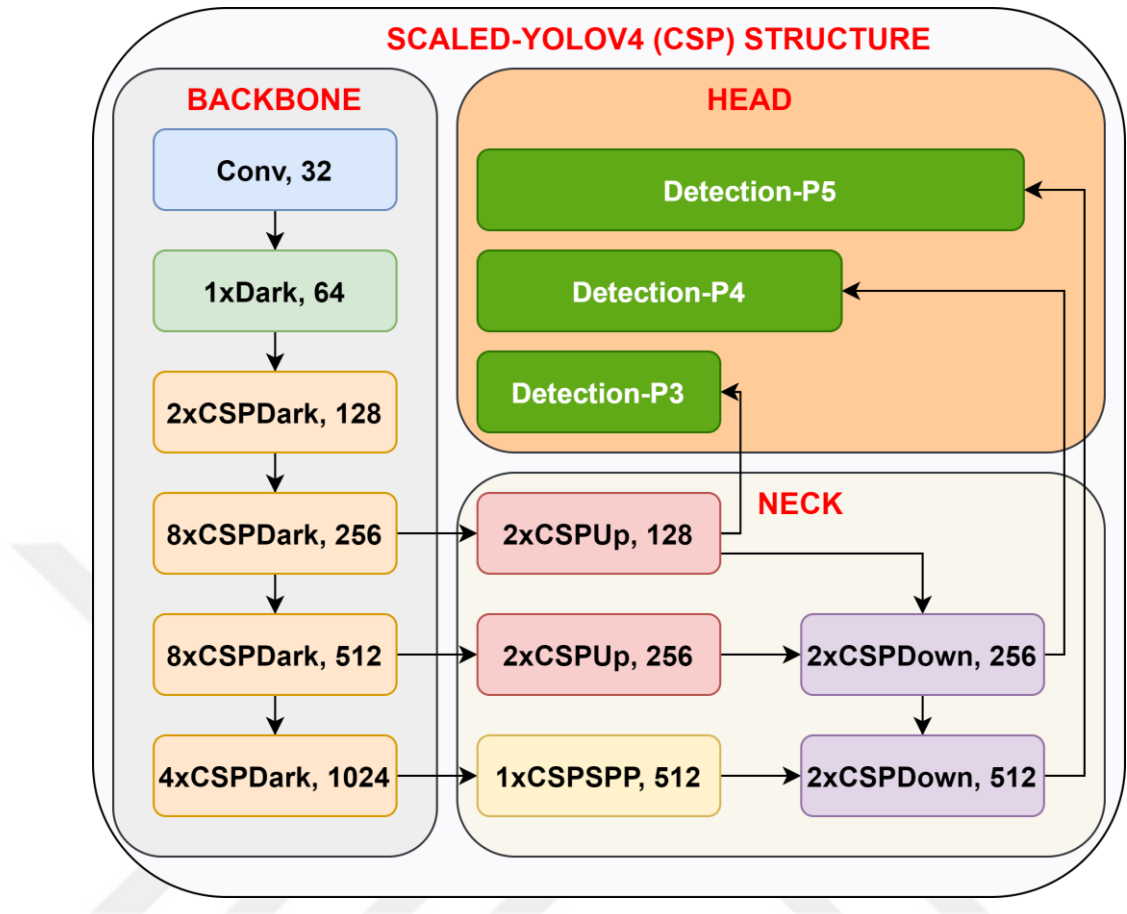
CSPNet, gradyanlardaki değişiklikleri dikkate almak için bir ağ aşamasının başlangıcından sonuna kadar özellik haritalarını entegre eden, aşamalar arası kısmi bir ağıdır. CSPNet, yukarıdaki Şekil 4.5'te gösterildiği gibi algılama performansını iyileştirmek için daha zengin bir gradyan kombinasyonu sunar. Temel katmanın özellik haritasını iki kola böler. Kollardan biri daha fazla gelişme için birkaç katmandan geçerken diğeri bu katmanları atlar. Böylece gradyan bilgisi farklılaşarak zenginleşir.

Sonuç olarak, bellek maliyeti azalır ve nesne algılama modelinin performansı artar. Bu nedenle YOLOv4 mimarisinin ölçeklendirmesini gerçekleştirmek için CSP uyumlu modeller kullanıyoruz. Güçlü gerçek zamanlı polip tespiti için, hem hız hem de doğruluk açısından daha iyi performans elde etmek için YOLOv4 mimarisinde iyileştirmeler yaparak daha optimize bir yöntem öneriyoruz. Şekil 4.6, YOLOv4 mimarisinin yapısını göstermektedir.



Şekil. 4.6. YOLOv4 ağının temel yapısı.

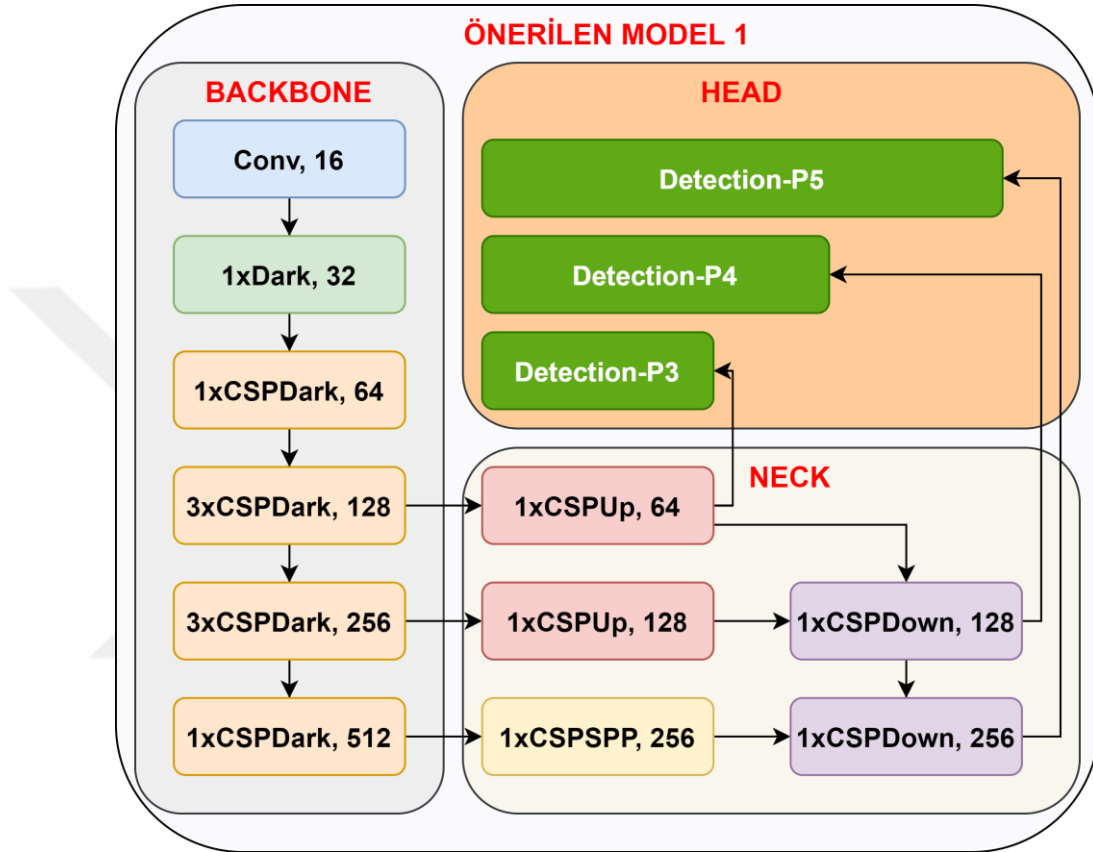
CSPNet, YOLOv4'ün yalnızca omurga kısmında kullanılır. YOLOv4-CSP'de (Scaled YOLOv4) ise tüm mimaride CSPNet kullanılır. Şekilde de görüldüğü üzere en belirgin fark CSPNet'in Scaled YOLOv4'te kullanılmasıdır. Omurga ve boyun yapılarında değişiklikler görülürken, kafa kısmında her iki modelde de YOLOv3 kafası kullanılmaktadır. Şekil 4.7, Scaled YOLOv4 yani YOLOv4-CSP mimarisinin yapısını göstermektedir.



Şekil 4.7. Scaled-YOLOv4 (YOLOv4-CSP) mimarisine ait yapı.

Burada önerdiğimiz modelin amacı YOLOv4 mimarisinin ölçeklendirilmesi ve gerçek zamanlı otomatik polip algılama sistemi için daha optimize bir model olmasını sağlamaktır. Ana hedefimiz, çıkarım hızını düşürürken veya korurken doğruluğu artırmak veya sabit etmektir. Detektörün girişi, model omurgası ve boyun daha sonra ölçekleme faktörlerini ayarlayarak en iyi kombinasyonu bulmak için ayarlanır. Giriş görüntü boyutu artırıldığında, ağırlık derinliği veya aşama sayısı artırılarak daha iyi algılama sağlanabilir. Ancak, ağırlık gerçek zamanlı algılama gereksinimi de dikkate alınmalıdır. Önerilen YOLOv4 mimarimiz sırasıyla şu önemli değişiklikleri içerir. İlk olarak, YOLOv4'ü yeniden tasarladık ve YOLOv4-CSP formatına dönüştürdük, böylece yeni modeli istenen veri seti ve donanıma uyarlamayı kolaylaştırdık. Omurga için, YOLOv4 ile aynı CSPDarkNet53'ü kullanıyoruz, ancak ilk CSPNet katmanını bir DarkNet artık katmanı ile değiştiriyoruz. Scaled-YOLOv4 makalesinde belirtildiği gibi, CSPDarkNet, yalnızca çekirdek değeri birden büyük olduğunda DarkNet aşamasına göre daha iyi bir hesaplama avantajına sahiptir. Boyun için CSPNet ile birlikte önceden

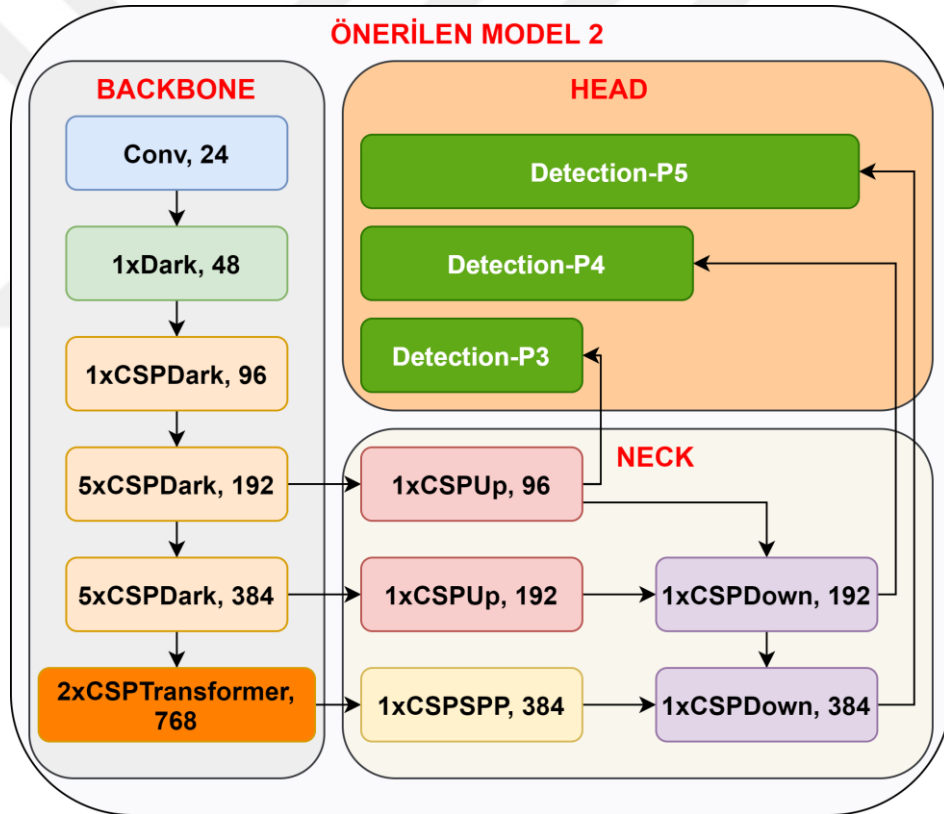
var olan PANet mimarisini kullanıyoruz. Son olarak, SPP modülü için mevcut yapıya CSP eklenmiş ve CSPPAN'ın ilk hesaplama listesi grubunun orta konumuna eklenmiştir. Bu, CSPNet'i omurga ve boyun kısımlarında daha etkin kullanmamızı sağlar. Polip algılama sistemleri için önerdiğimiz Model 1 Şekil 4.8'de gösterilmektedir.



Şekil 4.8. Önerilen Model 1 mimarisine ait temel yapı.

Yaptığımız ikinci değişiklik, YOLOv4 mimarisine Transformers ekleyerek performansı artırmaktı. Transformers, Vaswani ve diğerleri [121] tarafından önerilen doğal dil işlemede kullanılan en gelişmiş yöntemlerden birisidir. Ancak, bilgisayarla görmede sınırlı kalmıştır. Hem Facebook hem de Google, nesne tespitinde Transformers'ı nesne tespit algoritmalarına uyarlamışlardır. Özellikle, Vision Transformer (ViT) [122], son teknoloji CNN'lere kıyasla daha az bilgi işlem kaynağı gerektirirken mükemmel sonuçlar elde eder. Vizyon yapısına benzer şekilde, Transformers'ı CNN ağı ile hibrit bir yapıda kullandık, böylece yeni yapının daha az FLOPS'a ihtiyacı vardı ama aynı zamanda çıkarım ve eğitim sürelerine de olumlu katkı sağladı. Transformatör bloğunun omurga üzerindeki son bloğa (1024 kanallı P5 bloğu, parametre numaralarının çoğundan sorumlu blok) yerleştirilmesi, modelin nesne algılama kabiliyetini, özellikle

büyük nesnelerin algılanmasını arttırır. Deneylerimizde omurga üzerindeki diğer bloklara (128 kanal, 256 kanal vb.) transformerlerin yerleştirilmesinin algılamayı artırmadığı gözlemlenmiştir. Aslında, tam tersine, daha fazla hesaplama yüküne neden oldu. En iyi seçenek, son CSP bloğunu kaldırmak ve transformer bloğunu CSP ile birlikte yerleştirmektir. Ardından, ilk artık blok dâhil olmak üzere kalan tüm blokları CSPNet ile kullandık. CSPNet'in ilk bloğa eklenmesi, hesaplama maliyetini biraz artırsa da, CNN'nin öğrenme yeteneğini de artırdı. Deneylerimizdeki verilere dayanarak, sığ modellerin küçük veri setleri üzerinde daha derin modellerden daha iyi performans sağladığı sonucuna vardık. Polip algılama sistemleri için önerdiğimiz Model 2 (Transformatörlü Orta blok) Şekil 4.9'da gösterilmektedir.



Şekil 4.9. Önerilen Model 2 mimarisine ait temel yapı.

Daha detaylı olarak, önerdiğimiz yöntem, yukarıdaki şekilde gösterildiği gibi filtre ve blok sayısının azaltılması ile daha sığ bir model elde edilmiştir ve bu model YOLOv4-CSP'den farklıdır. Örneğin, YOLOv4-CSP'deki CSPDarkNet blokları (1xDark, 2xCSPDark, 8xCSPDark, 8xCSPDark, 4xCSPDark) sırasıyla (1xDark, 1xCSPDark, 3xCSPDark, 3xCSPDark, 1xCSPDark)'a düşürülmüştür. Transformer bloğunun

eklenmesiyle aynı işlemler gerçekleşir. Mevcut yapıyı (1xDark, 2xCSPDark, 8xCSPDark, 8xCSPDark, 4xCSPDark) yerine (1xCSPDark, 1xCSPDark, 5xCSPDark, 5xCSPDark, 2xCSPTransformer)'a çevirerek daha sağlam bir yapı sunuyoruz ve ayrıca filtre sayısını azaltmış oluyoruz. Model1 ve Model2 için bu iki ana değişikliği anlamak için iki yapı birlikte düşünülmelidir. Boyundaki bu değişikliklerle hem katman sayısındaki azalma hem de filtre sayısındaki azalma görülebilir. Bu yeni yapı ile hem omurgadaki hem de boyundaki filtre sayısı azaltılarak daha sıkı modeller elde edilmiştir. Tüm yapıyı sıkılaştırarak daha iyi doğruluk ve hız elde ettik ve böylece polip tespit sisteminin eğitim süresini kısalttık. Ama şunu unutmamak gerekir ki bu modellerde ki asıl amaç daha çok küçük veri setleri üzerinde daha iyi sonuçlara ulaşmak için modellerin sıkılaştırılmasıdır. Ayrıca, bu yapılar büyük veri setinde uyarlanması için modellerin daha da derinleştirilmesi ile ancak daha başarılı sonuçlar verir. Bu değişikliklerin yanı sıra ResNet ağları, VGG ağları, DarkNet53 ve versiyonlarını omurga kısmına dâhil ederek farklı yapıların polip tespitine katkısını inceledik. YOLOv3'te DarkNet53 yerine CSPDarkNet53 mimarisini de kullandık. Kısacası, YOLOv3 ve YOLOv4 mimarilerine farklı omurgalar, boyunlar ve aktivasyon ve kayıp fonksiyonları ekleyerek bu modellerin gerçek zamanlı polip algılama performansını iyileştirdik.

4.6.1. Önerilen Modellere Ait Sonuçlar

Bu bölümde sıkı modellere ait deneysel sonuçlar paylaşılarak polip tespitinin derin öğrenme algoritmaları yüksek düzeyde performans gösterdikleri görülmüştür. Önerilen modeller, YOLOv3 ve YOLOv4 algoritmaları ile beraber, kayıp fonksiyonları, aktivasyon fonksiyonları, omurga yapıları ve çeşitli yapıların eklenmesi ile bu yapıların polip tespitine katkısı incelenmiştir. Bu tezde dört ana model önerilmiş, bu modeller sıkı ve derin modeller olarak ikiye ayrılmaktadır. Sıkı ya da küçük modeller küçük veri setleri için en ideal modeller iken, derin ya da büyük modeller ise ve büyük veri setleri için en ideal modeller olmakla beraber deneysel olarak kanıtlanmıştır. Daha öncede belirtildiği üzere bunu yapmamızın sebebi nesne tespit algoritmasının ölçeklenmesidir. Nesne tespit algoritmaları ölçeklenerek giriş görüntüsü ve donanımına bağlı olarak optimize edilir. Bu sayede büyük veri seti için önerilen bir model küçük bir veri setinde istenmeyen başarıyı gösterebilir. Biz burada, birçok modelle beraber birçok yapıyı bütünleşmiş ederek bu sistemin birçok medikal görüntü işleme alanında kullanılmasına

olanak veriyoruz. Bu sayede daha esnek bir yapı sunulmuş olmaktadır. Öncelikle önerilen modellerin ve doğal YOLO mimarilerin halka açık veri setleri üzerindeki başarımı incelenmiştir. Daha sonra ise özel veri setimiz üzerindeki başarımı incelenmiş ve bu veri setinin klinik uygulamalar için uygulanabilirliğini gösterilmektedir. Son olarak ise nesne tespit algoritmalarında çok önemli bir yer tutan ve özellikle tespitin doğruluğunu önemli derece artıran hiper-parametre optimizasyonunu için kullandığımız ABC ve GA algoritmalarının polip tespiti doğruluğunu artırmadaki performansına bakılmıştır. Böylece, polip tespiti için hangi optimizasyon algoritmasının daha iyi sonuç verdiğini gözlemlemiş olduk. Bu bölümde halka açık olan veri setleri ile önerilen modellerle beraber YOLO modellerinin detaylı bir karşılaştırılması sunulmuştur. YOLOv3 ve YOLOv4 ile beraber önerilen modellerin farklı yapılarla beraber entegrasyonu sonucu halka açık veri setlerinden olan CVC-ClinicDB, CVC-ColonDB, Etis-Larib, PICCOLO ve SUN veri setleri üzerindeki başarımları ölçülmüştür. Halka açık veri setlerini de daha iyi bir karşılaştırma sunulması açısından küçük ve büyük veri setleri diye iki kısmı ayırdık. Buradaki büyük ve küçük manası ise eğitimde kullanılan veri setinin binler düzeyinde ya da yüzler düzeyinde olup olmamasına göre sınıflandırılmıştır.

Bu bölümde sıg yani küçük modellerin küçük veri setleri üzerinde eğitilmesi sonucu başarımları karşılaştırılmıştır. Burada sadece CVC-ClinicDB (612 polip görüntüsü) veri seti eğitimde kullanılmıştır. Böylelikle sıg modellerin küçük veri setleri üzerindeki başarımı ölçülmüştür. Test veri seti olarak ise CVC-ColonDB (300 polip görüntüsü) ile beraber Etis-Larib (196 polip görüntüsü) veri setleri seçilmiştir. Bunları seçmemizin nedeni ise çalışmamızı diğer çalışmalarla daha objektif bir şekilde karşılaştırmak için sadece halka açık veri setlerinin hem eğitimde hem de test verisinde kullanarak Kolonoskopide Otomatik Polip Tespitinde MICCAI Alt-Yarışmasını takip edilmesidir. Böylelikle önerilen modellerin güncel yüksek performans gösteren modellerle karşılaştırılması ve başarımın ölçülmesi daha adil bir şekilde olmaktadır. Daha önceki bölümlerde anlatıldığı üzere iki sıg model küçük veri setleri için önerilirken (Model 1 ve Model 2), iki derin model ise büyük veri setleri için önerilmiştir. Şimdi önerilen iki modelin polip tespitindeki başarımına bakalım. Daha öncede anlatıldığı üzere, YOLOv3 ve YOLOv4 algoritmalarında yapısal değişiklikler yaptık ve bunları gerçek zamanlı polip tespitine uyarladık. Bunu yapmak için aktivasyon fonksiyonu, kayıp fonksiyonu

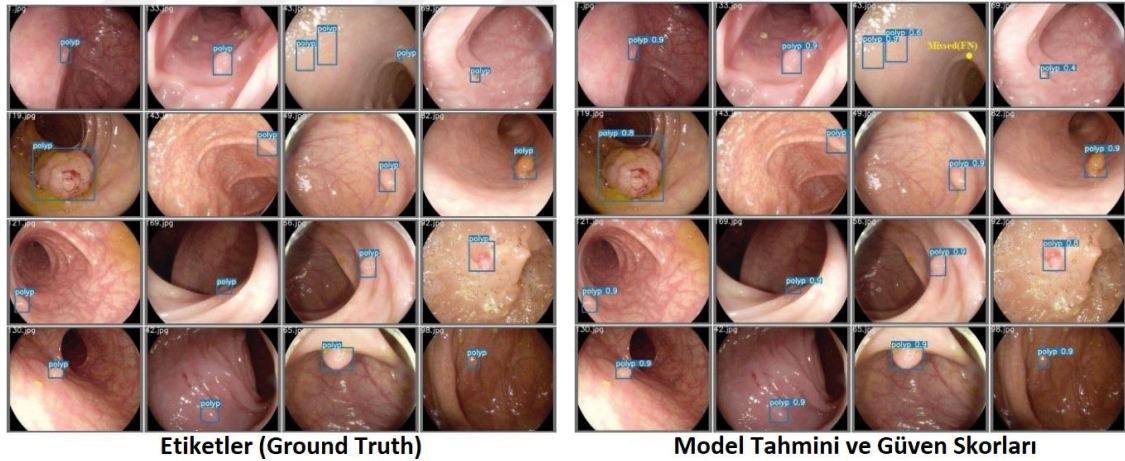
ve transformers gibi farklı yapılar ekleyerek bu yapıların polip tespiti üzerindeki etkilerini inceledik. DarkNet53, CSPDarkNet53, ResNet ve VGG gibi yapıları karşılaştırarak omurgada en büyük değişiklikleri yaptık. YOLOv3 ve YOLOv4 algoritmaları ve değiştirilmiş yapıları, daha adil bir performans karşılaştırması yapmak için CVC-ClinicDB (612 görüntü) veri seti ile eğitildi. Özellikle YOLOv4 mimarisine odaklanarak polip tespiti için ideal yapılar sunduk ve bu yapıları optimize ettik. Değerlendirme için, Otomatik Polip Tespitinde MICCAI Alt-Yarışmasını izledik ve eğitim için CVC-ClinicDB'yi ve test için ise Etis-Larib veri setini kullandık. YOLOv4 algoritmasını farklı yapılarla modifiye ederek en iyi yapıyı bulmayı amaçladık ve YOLOv4 üzerinde yapılmış en büyük polip tespit çalışması ile kapsamlı bir yol haritası sunduk. Tüm çalışmamızın ayrıntılı bir incelemesi Tablo 4.1'de gösterilmektedir. CSP'ler, model ölçeklemenin sadece boyunda yapıldığını gösterirken, Full-CSP, CSP bloğunun tüm boyunda ve DarkNet yerine ilk katmanda kullanıldığını gösterir. Model değerlendirmesinde, adil sonuçlar için 0,5'lik IOU eşliğini, küme boyutunu 1, güven eşliğini 0,25 ve tek grafik kartı RTX 2080TI'yi kullandık. Tablo 5.1'de eğitim veri seti olarak Otomatik Polip Tespitinde MICCAI Alt-Yarışmasını dikkate alarak CVC-ClinicDB halka açık veri setini seçtik. Test veri seti olarak ise Etis-Larib veri setini seçerek, metrikleri ise yine aynı kuralları izleyerek yaptık.

Tablo 4.1. Etis-Larib test veri setine ait sonuçlar (Eğitim veri seti: CVC-ClinicDB, IOU: 0.5, küme boyutu 1, güven eşliği: 0.25, ekran kartı: 1xRTX 2080TI).

Model	Omurga	Kayıp Fonksiyonu	Aktivasyon Fonksiyonu	Keskinlik %	Duyarlılık %	F1 %	Hız (mili-saniye)
YOLOv3	DarkNet53	GIoU	Relu	79.44	76.10	77.73	10.3
YOLOv3-SPP	DarkNet53	GIoU	Relu	80.19	78.61	79.39	10.4
YOLOv3-SPP	DarkNet53	DIoU	Mish	83.01	80.66	81.82	10.4
YOLOv3-SPP	CSPDarkNet53	CIoU	Mish	83.43	77.28	80.24	10.5
YOLOv3s-SPP	CSPDarkNet53	CIoU	Mish	87.99	77.32	82.31	7.2
YOLOv3-SPP	DarkNet53	CIoU	Swish	82.81	79.88	81.32	10.4

YOLOv4-Tiny	CSPVoVNet	CIoU	Mish	70.27	61.32	65.49	2.6
YOLOv4	CSPDarkNet53	CIoU	Leaky Relu +Mish	81.78	79.25	80.50	13
YOLOv4-CSP	CSPDarkNet53	CIoU	Mish	82.55	82.55	82.55	12.2
YOLOv4-CSP	VGG16	CIoU	Mish	82.56	75.94	79.12	9.8
YOLOv4-CSPs	VGG16	CIoU	Mish	84.66	75.47	79.80	8.2
YOLOv4-CSP	ResNet18	CIoU	Mish	85.48	73.58	79.59	7.8
YOLOv4-CSP	ResNet34	CIoU	Mish	87.08	73.11	79.49	9.5
YOLOv4-CSPs	ResNet34	CIoU	Mish	89.27	74.53	81.23	8.4
YOLOv4-CSP	ResNet50	CIoU	Mish	78.95	77.83	78.38	11
YOLOv4-CSPs	ResNet50	CIoU	Mish	80.12	79.28	79.70	10
YOLOv4-CSP	ResNeXt50	CIoU	Mish	87.36	75.00	80.71	13.1
YOLOv4-CSP-S	ResNeXt50	CIoU	Mish	83.44	78.69	81.00	12.2
Model 1	CSPDarkNet53	DIoU	Mish	91.62	82.55	86.85	8.2
Model 1	CSPDarkNet53	CIoU	Mish	91.55	82.23	86.64	8.5
Model 1	CSPDarkNet53	DIoU	Swish	89.10	83.21	86.05	8.4
Model 2	CSPDarkNet53 + Transformer	DIoU	Mish	88.87	80.95	84.85	9.9
Model 1 Full-CSP	CSPDarkNet53	DIoU	Mish	88.60	80.66	84.44	8.6
Model 1 (X-Small)	CSPDarkNet53	DIoU	Mish	86.22	79.49	82.72	6.2
Model 3	DarkNet53	CIoU	SiLU	88.05	80.21	83.95	10
Model 4	CSPDarkNet53	CIoU	SiLU	86.98	80.65	83.70	11

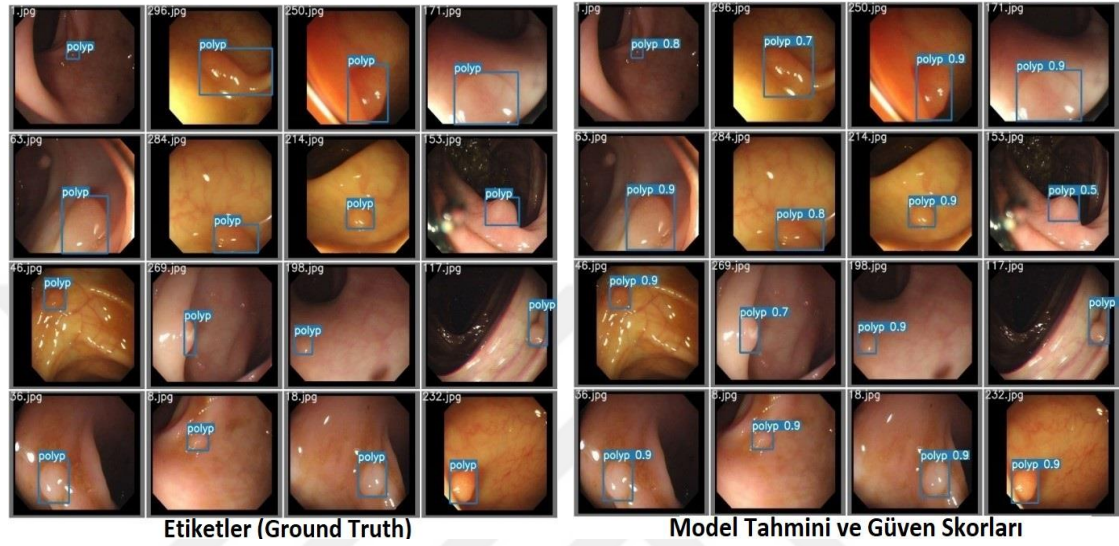
Tablo 4.1 düşünüldüğünde, modelin performansına en fazla katkıda bulunan değişikliklerin omurga, aktivasyon fonksiyonu ve kayıp fonksiyonu olduğunu göstermektedir. Veriye aç derin öğrenme modelleri, yeterli veri kümelerine sahip olduklarında maksimum performans sunar. Tablo 5.1'de gösterilen daha küçük modeller, polip tespit sistemine daha fazla katkıda bulunmuştur. Bu önemlidir çünkü CVC-ClinicDB veri setinde çok az sayıda benzersiz polip vardır (31), veri setindeki görüntüler düşük çözünürlüklüdür (384×288) ve veri setinde çok fazla görüntü yoktur. Bu durumda, daha derin modellerin MS-COCO gibi yeterli ve iyi etiketlenmiş veri setine ihtiyaç duyduğunu, sık modellerin ise daha küçük bir veri seti ile daha iyi performans gösterebildiğini gözlemledik. Önerilen YOLOv4 Model 1 hem hız hem de doğruluk açısından en iyi performansı sunarken, orta modelde (Model 2) CSPDarknet53 ile Transformer bloğu da benzer sonuçlar vermiştir. Diğer taraftan, büyük veri setleri için önerilen Model 3 ve Model 4 ise sık modellere oranla hem hız hem de doğruluk açısından daha düşük performans göstermiştir. İlginç bir şekilde, YOLOv3 hem hız hem de doğruluk açısından YOLOv4'e çok yakın sonuçlar verdi.



Şekil 4.10. Soldaki şekil ETIS-LARIB veri setinin temel gerçeğini gösterirken, sağdaki şekil en iyi modelin tahminini ve tahminlerin güven puanlarını göstermektedir.

CSPNet, YOLOv3'ün hem omurgasında hem de boynunda kullanıldığında Scaled-YOLOv4'e benzer ve rekabetçi sonuçlar verdiği görülmektedir. ResNet ve VGG omurga olarak kullanıldığında ve sadece boynunda CSPNet kullanıldığında DarkNet mimarisine göre daha az özellik çıkardığı yani daha düşük sonuçlar verdiği gözlemlenmiştir. Bu mimariler, boyun sık bir modele ölçeklendiğinde daha iyi sonuçlar verdi. Özetlemek gerekirse, CSPNet'in hem omurga hem de boyun (tüm ağ) üzerinde kullanıldığında ve

özel problem için ölçeklendiğinde diğer mimarilerden daha başarılı olduğu gözlemlenmiştir. Önerilen Model 1'in Etis-Larib ve CVC-ColonDB veri setlerinin bazı görüntüleri üzerindeki tespit performansı sırasıyla Şekil 4.10 ve 4.11'de gösterilmektedir.



Şekil 4.11. Soldaki şekil CVC-ColonDB veri setinin temel gerçeğini gösterirken, sağdaki şekil en iyi modelin tahminini ve tahminlerin güven puanlarını göstermektedir.

4.6.1.1 Güncel Yöntemlerle Karşılaştırılması

Bu bölümde önerilen küçük modellerden en iyi sonucu alan Model 1'in polip tespit sistemi için, eğitim ve test için tam olarak aynı veri setini kullanan ve MICCAI Otomatik Polip Tespiti Alt Yarışmasında yer alan ekiplerin sonuçlarıyla karşılaştırdık. İlk olarak, MICCAI'ye dâhil edilen CUMED, OUS ve UNS-UCLAN olmak üzere her takımın en iyi üç sonucunu dâhil ettik. Bu üç ekip, polip tanıma için geleneksel CNN yapılarını kullandılar. Ardından, polip tespiti için son teknoloji performansa sahip farklı yöntem ve modeller kullanarak nesne tespit algoritmalarını başarıyla uyarlayan modelleri dahil ettik. Buradaki amaç, gerçek zamanlı otomatik polip tespiti için önerilen yöntemin diğer yöntemlerle adil bir karşılaştırmasının sağlanmasıdır. Tablo 4.2'de değerlendirme metrikleri; kesinlik, duyarlılık ve F1'in yanı sıra gerçek pozitif (TP), yanlış pozitif (FP), yanlış negatif (FN), çıkarım süresi yani gerçek zamanlı performans (hız) gibi metrikleri gösterilmektedir.

Tablo 4.2. Önerilen yöntemlerin diğer güncel yöntemlerle karşılaştırılması (Eğitim seti: CVC-ClinicDB, test seti: Etis-Larib, IOU: 0.5, küme boyutu 1, güven eşiği: 0.25, ekran kartı: 1xRTX 2080TI).

Metot	Metot	TP	FP	FN	Kesinlik %	Duyarlılık %	F1 %	Hız (mili-saniye)	Gerçek zamanlı tespit
CUMED [113]	CNN	144	55	64	72.30	69.20	70.7	200	Hayır
OUS [113]	CNN	131	57	77	69.70	63.00	66.1	5000	Hayır
Shin ve ark. 2018 [20]	Faster R-CNN	167	26	41	81.50	80.30	80.90	390	Hayır
Kang et al, 2019 [123]	Mask-RCNN	N/A	N/A	N/A	73.84	74.37	74.11	N/A	Hayır
Sornapudi ve ark. 2019 [19]	R-CNN	167	62	41	72.93	80.29	76.43	317	Hayır
Qadir ve ark. 2019 [17]	Mask R-CNN	N/A	N/A	N/A	80.00	72.59	76.12	430	Hayır
Jia et al, 2020 [124]	PLPNet (Faster R-CNN)	170	96	38	63.90	81.70	71.70	N/A	Hayır
Qadir ve ark. 2021 [16]	MDeNetplus (F-CNN)	180	28	28	86.54	86.12	86.33	39	Hayır
Liu ve ark. 2021 [42]	Consolidated Domain Adaptive Framework	182	52	26	77.80	87.50	82.40	108	Hayır
Xu ve ark. 2021 [125]	Modified-YOLOv3	149	30	59	83.24	71.63	77.00	35	Hayır
Model 1	YOLOv4-CSP	175	16	37	91.62	82.55	86.85	8.2	Evet
Topluluk Öğrenme (Model 1+Model 2)	YOLOv4-CSP	180	18	28	90.91	86.54	88.67	16.3	Evet
Topluluk Öğrenme (Üç Model)	YOLOv4-CSP	178	20	33	89.90	84.36	87.07	24.1	Evet
Model 1 ve TensorRT	YOLOv4-CSP	175	16	37	91.62	82.55	86.85	4	Evet
Topluluk Öğrenme (Model 1+Model 2) ve TensorRT	YOLOv4-CSP	180	18	28	90.91	86.54	88.67	8.3	Evet

Tablo 4.2'de sunulan sonuçlara dayanarak, önerdiğimiz Model 1'in kesinlik, F1 ve hız açısından diğer modellerden daha başarılı olduğunu ancak duyarlılık metriğinde sadece ikinci olduğunu görebiliyoruz. Ancak, topluluk öğrenmeyi kullanarak en iyi iki modelimizi birleştirdiğimizde, yeni topluluk modelimizin neredeyse tüm değerlerde en iyi performansa sahip olduğunu görüyoruz. Topluluk öğrenme polip tespiti başarımı için daha yüksek sonuçlar vermektedir fakat diğer taraftan ise FPS oranında ise ciddi oranda düşüş sağlamaktadır. Bu tür durumlarda modellerin gerçek manada gerçek zamanlı işlem yapma durumu sorgulanmalıdır. Tablo 4.2 düşünüldüğünde bizim önerdiğimiz modeller dışındaki tüm modellerin, R-CNN veya F-CNN tabanlı ağlar gibi iki aşamalı yaklaşımları nedeniyle gerçek zamanlı performansta başarılı olmaktan uzak olduğunu göstermektedir. Özellikle polip tespitinin hem gerçek zamanlı olması gerekir aynı zamanda ise yüksek FPS sunması gerekir ki bu sistemin kolonoskopi cihazında entegre edilmesi amaçlansın aksi takdirde gerçek zamanlı tespitin avantajlarında feragat edilmesi gereklidir. Bir modelin polipleri gerçek zamanlı olarak tespit edebilmesi için gereken minimum FPS hızı en az 30 olmalı veya çıkarım süresi 33 ms'den fazla olmamalıdır. Xu ve ark. [43], değiştirilmiş YOLOv3 algoritmasına sahip RTX 2080 TI grafik kartı ile 35 ms'lik bir çıkarım hızı elde etti. Ancak bu hız gerçek zamanlı algılama için yetersizdir. Önerilen yöntemimiz 8,2 ms'lik bir çıkarım süresi, bu şekilde gösterilen en düşük çıkarım (gecikme) ve tek RTX 2080 TI grafik kartıyla 122'lik bir FPS'lik performans sunmaktadır. Nvidia TensorRT (FP16, küme boyutu 1, görüntü boyutu 416) çerçevesini kullandığımızda, bu gecikme 4 ms'nin altına düşerek 250 FPS'den daha yüksek bir hız sağlar. Önerilen yöntemimizi Etis-Larib veri seti üzerinde değerlendirdikten sonra, aynı modelin performansını başka bir genel veri seti olan CVC-ColonDB üzerinde de değerlendirdik. Eğitimin sadece CVC-ClinicDB veri seti üzerinde yapıldığını vurgulamak isteriz. Yani Etis-Larib veri seti birincil bağımsız veri seti olarak kullanıldıktan sonra ikincil bağımsız veri seti ise CVC-ColonDB seçilmiştir. Böylece, eğitilen bir modelin başka bağımsız veri setleri üzerindeki başarımları görebilmektedir. CVC-ColonDB veri kümesindeki en iyi modelin performansı ve diğer son teknoloji algoritmalarla ayrıntılı bir karşılaştırma Tablo 4.3'te gösterilmektedir.

Tablo 4.3 Önerilen yöntemin diğer yeni yöntemlerle karşılaştırılması (Eğitim veri seti: CVC-ClinicDB, test veri seti: CVC-ColonDB, IOU: 0.5, küme boyutu 1, güven eşiği: 0.25, ekran kartı: 1xRTX 2080Ti).

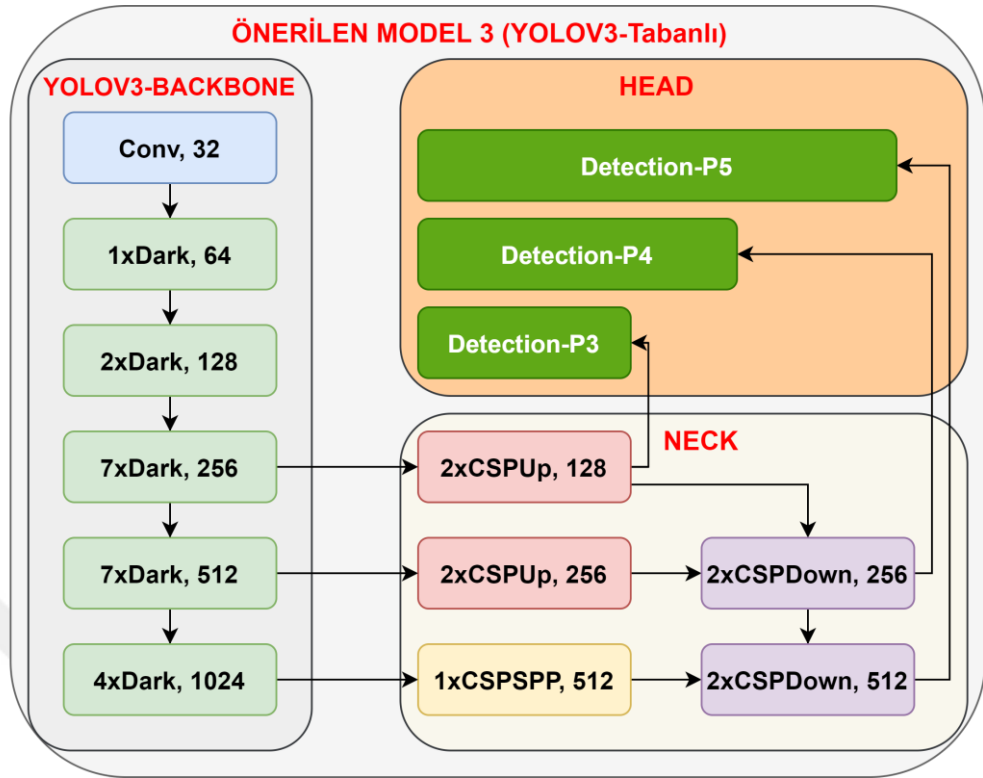
Metot	Metot	TP	FP	FN	Kesinlik %	Duyarlılık %	F1 %	Hız (mili-saniye)	Gerçek zamanlı tespit
Sornapudi ve ark. [19]	R-CNN	287	77	13	78.85	95.67	86.58	220	Hayır
Qadir ve ark. [16]	MDeNetplus (F-CNN)	273	36	27	88.35	91.00	89.65	39	Hayır
Model 1	YOLOv4-CSP	291	12	9	96.04	96.68	96.36	8.2	Evet

Tablo 4.3 göz önünde bulundurulduğunda, önerilen modelin (topluluk ve TensorRT'siz Model 1) önceki modellere kıyasla CVC-ColonDB veri setinde en yüksek değerlere sahip olduğunu göstermektedir. Yöntemimize en yakın çalışma Qadir ve ark. ait olan çalışmadır. Bu çalışma 39 ms çıkarım hızı ile 89.65 F1 değerine sahipken, yöntemimiz 8.2 ms çıkarım hızı ile 96.36 F1 değerine sahiptir. Sonuç olarak, önerilen yöntemimiz, rakiplerine göre gerçek zamanlı polip tespitinde önemli bir gelişme sağlamıştır çünkü hız ve doğruluk açısından oldukça başarılı olduğu aşikârdır.

4.7. Büyük Veri Setleri için Önerilen Modeller

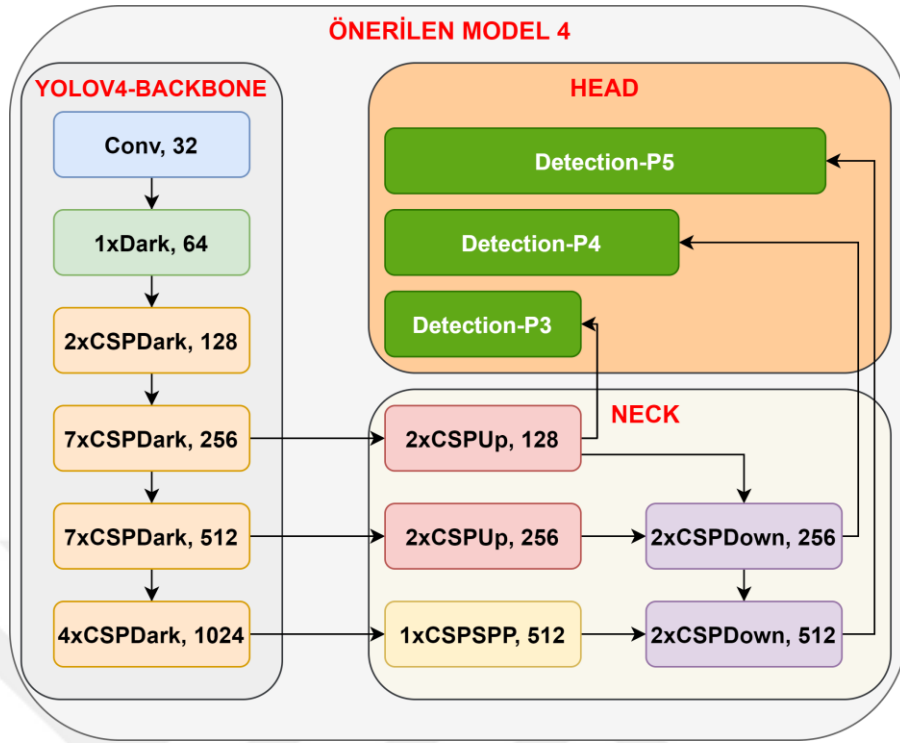
Bu bölümde, önceki bölümden farklı olarak büyük polip veri setleri için önerilen modellerden bahsedilecektir. Bu nedenle, YOLO mimarileri, veri artırma teknikleri, mimarilerdeki bazı temel yapı ve fonksiyonlar gibi daha önce anlatılan temel yapılardan bahsedilmeyecek. Daha çok veri setine bağlı olarak mimarinin nasıl derin ya da sığ hale getirilebildiğinden bahsedilmiştir. Diğer bir deyişle daha derin nesne tespit algoritmalarının polip tespiti sistemleri için ölçeklenmesi ve optimize edilmesini ele alınmaktadır. Bunun için YOLOv3 ve YOLOv4 mimarilerinin CSPNet ile ölçeklenmesi ve derin ağların elde edilmesi için bazı modeller önerdik. Önerilen yöntemimiz, önceki bölümlerde bahsedilen Model1 ve Model2 mimarilerinin benzer yapılarını içermekle beraber, ölçeklenmiş YOLO mimarisinin büyük veri setlerine uyarlanmış halini içermektedir. Diğer bir deyişle, tekrarlamak gerekirse mimarinin yapısal değişikliği şu

şekildedir. YOLO mimarilerinin boyun ve baş kısımları tamamen aynı olmasına rağmen, omurga kısmında temel değişiklikler bulunmaktadır. YOLOv3'te omurga kısmı için değiştirilmiş bir DarkNet53 yapısı kullanılır. Ancak YOLOv4 mimarisinin omurgasında CSPNet ile birlikte değiştirilmiş bir CSPDarkNet53 yapısı kullanılır ve bu yapı YOLOv4-CSP'ye benzer. YOLOv3 ve YOLOv4'ün ortak boyun ve baş kısımları, orijinal YOLOv4'ün aksine CSPNet ile birlikte kullanılır. Böylece, CSPNet mimarisinin YOLO nesne algılama algoritmaları üzerindeki etkisi de incelenmiştir. Bu yapılarıdaki temel amaç, büyük veri setleri için en ideal yapıyı bulmak için CSPNet ile YOLO mimarilerinin ölçeklenmesini gerçekleştirmektir. Önceki bölümlerde anlatıldığı üzere, küçük veri setleri için en ideal yapının sığ modeller olduğu deneysel olarak ispatlanmıştı ve tartışma bölümünde tablolar halinde verilmiştir. Fakat büyük veri setleri için daha derin modellerin daha yüksek tespit doğruluğu sunduğu deneysel çalışmalarda kanıtlanmıştır. Dikkat edilmesi gereken nokta ise model derinleştirilmesine çalıştırılınca, çıkarım hızı yani FPS hızının korunmasının önemlilik arz ettiğidir. Bu yüzden yapılan deneysel çalışmalarda en uygun modellerin bulunması amaçlanmıştır. Orijinal YOLOv3 ve YOLOv4 mimarilerinin çıkarım hızı ve tespit doğruluğu ölçüldükten sonra modeller ölçeklenmiştir. Böylece, üzerinde ilk geliştirmeler yapmak üzere YOLOv3 nesne tespit algoritmasını ele aldık. YOLOv3 algoritması, özellik çıkarımı için bir omurga olarak DarkNet53'ü kullanır. DarkNet53, eğitim sürecinde gradyan kaybolması problemlerini çözme yeteneğine sahiptir. Artık biriminin yapısı sırasıyla bir evrişim katmanı, bir toplu normalizasyon alt katmanı ve Leaky ReLU yapılarından oluşur ve buna CBL denir. Önerilen geliştirilmiş YOLOv3 mimarimizde, CBL ünitesindeki Leaky ReLU aktivasyon fonksiyonu yerine SiLU aktivasyon fonksiyonunu kullandık. DarkNet53 mimarisindeki evrişim katmanlarının sayısını (1xDark, 2xDark, 8xDark, 8xDark, 4xDark) yerine (1xDark, 2xDark, 7xDark, 7xDark, 4xDark) olarak değiştirdik. Bu yapılarıdaki filtre sayısını değiştirmedik, böylece modelin doğruluğunu sabit tuttuk ve modelin gerçek zamanlı algılamasını yani FPS oranını arttırdık. YOLOv3 mimarisinin CSPNet ile geliştirilmiş sürümü olan modele ait diyagram Şekil 4.12'deki gibidir.



Şekil 4.12. Önerilen Model 3 ağına ait temel yapı.

Sistemde nesne tespit algoritmasında yapılan diğer değişikliklerden bir tanesi de YOLOv4 nesne tespit algoritması üzerinde yapılan yapısal değişikliklerdir. Aynı durumu YOLOv4 mimarisinin özellik çıkarıcısı olan CSPDarkNet53 omurgasında da gerçekleştirdik. CSPDarkNet53 omurgasındaki filtre sayısını sabit tutarken evrişim katmanlarının sayısını (1xDark, 2xCSPDark, 8xCSPDark, 8xCSPDark, 4xCSPDark) yerine (1xDark, 2xCSPDark, 7xCSPDark, 7xCSPDark, 4xCSPDark) olarak yapıyı değiştirdik. Çünkü, yapılan deneysel çalışmalarda bu yapının orijinal YOLOv3 ve YOLOv4 algoritmalarına göre hem hız konusunda hem de tespit doğruluğu açısından daha iyi sonuçlar verdiği gözlemlenmiştir. Ek olarak, CSPDarkNet'in DarkNet aşamasına göre daha iyi hesaplama avantajı nedeniyle, yalnızca çekirdek değeri birden büyük olma koşulu karşılandığında, Scaled-YOLOv4 makalesinde belirtilen bir nokta olan ilk CSP katmanını DarkNet artık katmanıyla değiştirdik. YOLOv4 mimarisinin CSPNet ile geliştirilmiş sürümü olan modele ait diyagram aşağıdaki Şekil 4.13 'deki gibidir.



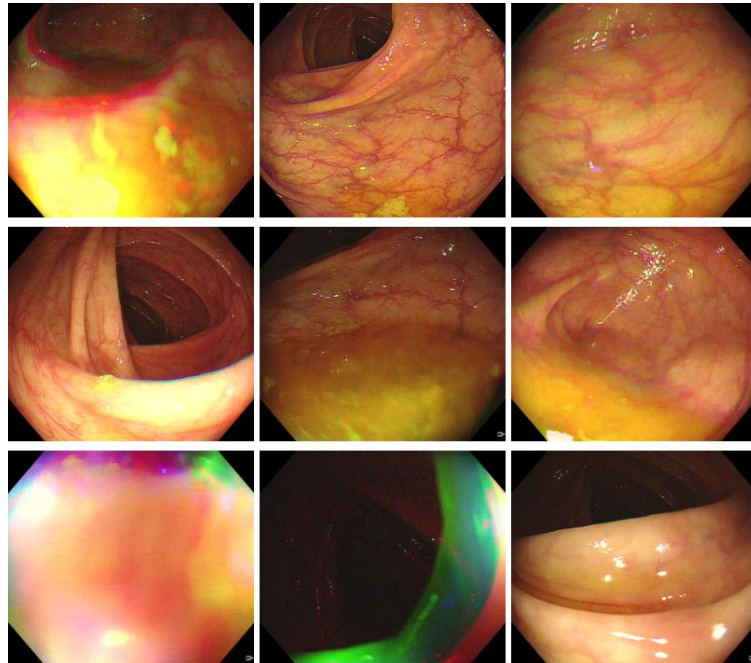
Şekil 4.13. Önerilen Model 4 ağına ait temel yapı.

Yukarıdaki şekillerde de görüldüğü üzere, buradaki amaç nesne tespit algoritmalarından olan YOLO mimarisinin polip tespit sistemi için yeniden ölçeklenmesi ve optimize edilmesine dayanmaktadır. Bu bağlamda, nesne tespit algoritmasının omurga, boyun kısımları yeniden ölçeklenirken, kafa kısmı YOLOv3 kafasıyla aynı kalmıştır. Bununla beraber filtre ve katman sayıları değiştiği için modelin hızı ve başarımında değişiklikler olmuştur. Fakat yapılan deneylerde küçük veri setleri ve büyük veri setleri için en ideal yapılar sunulmuştur. Bununla beraber daha birçok model sunulabilir, fakat en ideal modellerin hangisi olduklarını göstermekle beraber çok daha detaylı karşılaştırmalar tartışma bölümünde sunulmuştur.

Kısaca, büyük veri setleri ve negatif örneklerinde eğitime katılması için (toplamda daha büyük veri seti elde edildiğinden dolayı), YOLOv3 ve YOLOv4 mimarilerinin boyun ve baş kısımlarının ortak yapıları aşağıdaki gibi yeniden düzenlenmiştir. Buradaki temel amaç, tüm yapıyı tamamen CSPNet ile değiştirerek maliyeti düşürmek ve nesne algılama performansını artırmaktır. PANet yapısını ve SPP yapısını CSPNet ile birlikte kullanarak, CNN'nin öğrenme kabiliyetini tüm yapı boyunca daha etkin bir şekilde artırmayı hedefledik. Aktivasyon işlevleri ağırlık verilerdeki karmaşık kalıpları öğrenmesini sağlar. Tüm ağda YOLOv3 ve YOLOv4'ün önceden var olan Leaky ReLU

ve Mish etkinleştirme işlevlerinin yerine SiLU aktivasyon fonksiyonu ile değiştirdik çünkü SiLU aktivasyon fonksiyonu hem deneylerde hem de orijinal makalede SiLU'nun ReLU aktivasyon fonksiyonundan daha iyi performans gösterdiğini bildirmiştir. Bu nedenle, polip algılama performansını artırmak için tüm aktivasyon fonksiyonlarını SiLU ile değiştirdik.

Modelin başarımını artırmak için negatif örnekler (polip içermeyen) yani etiketli olmayan görüntüler ekledik. Bu görüntüler halka açık veri setleri içerisinde sadece SUN veri setinde yer alırken, Önerilen Özel Polip veri setinde de etiketli görüntünün iki katı kadar negatif örnekler (200.000 adet görüntü) yer almaktadır. SUN veri setine tekrar bacak olursak şu özelliklerden oluşuyor. SUN Kolonoskopi Video Veritabanı, uzman endoskopistler tarafından etiketlenmiş otomatik kolorektal polip tespiti için kolonoskopi tabanlı halka açık bir veri setidir. Bu veri seti, sınırlayıcı kutulu 100 tam açıklamalı polipten 49.136 polip çerçevesi ve polipsiz 109.554 çerçeve içerir. Bu sayılar, mevcut diğer herhangi bir halka açık polip veri setinden daha fazla polip gösterir. Aynı zamanda polipsiz görüntüleri içeren ilk halka açık polip veri setidir. SUN veri setine ait bazı polip içermeyen görüntüler yani negatif örnekler Şekil 4.14'te gösterilmektedir.



Şekil 4.14. Negatif örneklere (polip içermeyen kolonoskopi görüntüleri) ait görüntüler.

Şekil 4.14'te görüldüğü üzere herhangi bir polip içermeyen kolonoskopi görüntüleri negatif ya da etiketsiz veriler olarak tanımlanabilir. Bu tarz görüntüler sayesinde yanlış negatif ve yanlış pozitif değerleri düşürülerek model daha başarılı ve daha iyi polip tespit etme kabiliyetine kavuşuyor. Bununda ilgili deneysel çalışmalar ve sebep-sonuç ilişkileri tartışma bölümünde verilmiştir.

4.7.1. Önerilen Derin Modellere Ait Sonuçlar

Bu bölümde derin yani daha büyük modellerin büyük veri setleri üzerinde eğitilmesi sonucu başarımları karşılaştırılmıştır. Burada PICCOLO, SUN ve diğer küçük veri setleri ile bütün verilerin birleşimiyle daha büyük veri setlerinin elde edilmesi sonucunda polip tespitinin başarımı incelenmiştir. Test veri seti olarak ise aynı veri setleri ile beraber CVC-ColonDB (300 polip görüntüsü) ile beraber Etis-Larib (196 polip görüntüsü) veri setleri seçilmiştir. Bunları burada yine kullanmamızın nedeni ise çalışmamızı diğer çalışmalarla daha objektif bir şekilde karşılaştırmak ve Kolonoskopide Otomatik Polip Tespitinde MICCAI Alt-Yarışmasını takip edilmesidir. Böylelikle önerilen modellerin güncel yüksek performans gösteren modellerle karşılaştırılması ve başarımın ölçülmesi doğrulanmış olacaktır. Daha önceki bölümlerde anlatıldığı üzere iki sığ model küçük veri setleri için önerilirken (Model1 ve Model2), iki derin model (Model 3 ve Model 4) büyük veri setleri için önerilmiştir. Şimdi önerilen iki modelin polip tespitindeki başarımlarına bakalım.

4.7.1.1 SUN Veri Seti için Sonuçlar

Bu bölümde önerilen yöntemler için SUN veri seti üzerinde deneysel çalışmalar gerçekleştirdik. SUN veri setinde 49136 polip görüntüsü bulunmaktadır ancak bu görüntüler eğitim veya test olarak ayrı klasörlere ayrılmamaktadır. SUN veri seti, her dosya bir hastadaki benzersiz polip sayısını temsil eden 100 dosyadan oluşur. Nesne tespit algoritmalarındaki çalışmalarda, eğitimde SUN polip veri seti kullanan herhangi bir çalışma yapılmamıştır. Misawa ve ark. değerlendirme sürecinde sadece SUN polip veri setini kullandılar. Bu nedenle daha objektif bir çalışma için SUN Polyp verilerini aşağıdaki gibi ayırdık. 1-80 arasındaki hasta dosyalarını topladık ve sadece eğitim için kullandık. Bu dosyalardaki toplam polip görüntü sayısı 40707'dir. Geriye kalan 81 ile 100 arasındaki dosyaları test işlemi için kullandık. Bu şekilde oluşturulan toplam görüntü sayısı yani test görüntüsü sayısı 8429'dur. Önerilen modeller ve diğer

modellerin eğitim sürecini üç ana başlık altında inceledik. Modelleri önce sıfırdan eğittik. Daha sonra modelleri transfer öğrenme tekniği ile eğittik. Son olarak ise eğitime dâhil edildiğinde negatif polip (polipsiz) görüntülerinin polip tespit performansına katkı sağlayıp sağlamadığını inceledik.

Ek olarak, negatif örneklerin SUN veri setine katkısını incelemek için SUN veri setinden negatif polip örnekleri kullandık. Pozitif örneklerin negatif örneklere oranını 1:1, 1:0.5 ve 1:10 olarak tuttuk ve bu oranlar da Tablo 10.6'da gösterilmektedir. Model adı, eğitim görüntü boyutu, kesinlik, duyarlılık, F1-skor ve mAP gibi metrik değerlerini vererek çalışmamızı değerlendirdik. Tüm deneylerde, diğer çalışmalara benzer şekilde IoU değerini 0,5 ve güven eşiği 0,25 olarak sabitledik.

Tablo 4.4. SUN polip veri seti için uçtan uca örneklerle (sıfırdan eğitim) eğitilmiş modellerin sonuçları (IOU: 0.5, küme boyutu 1, güven eşiği: 0.25, ekran kartı: 1xRTX 2080TI).

Model	Görüntü boyutu	Kesinlik %	Duyarlılık %	F1 %	mAP (mAP@.5) %	mAP@.5:.95 %
YOLOv3-SPP	448	91.57	72.54	80.95	81.58	49.58
YOLOv4-CSP	448	91.01	76.27	82.99	96.79	65.22
Önerilen Model 3 (YOLOv4 tabanlı)	448	91.67	77.90	83.63	97.18	64.86
Önerilen Model 4 (YOLOv3 tabanlı)	448	91.65	83.83	87.56	96.69	64.36

Tablo 4.4'te gösterildiği gibi, önce YOLOv3, YOLOv4, önerilen YOLOv4 (Model 3) ve önerilen YOLOv3 (Model 4) modellerini sıfırdan eğittik. Sıfırdan eğitim modelleri transfer öğrenme gibi yöntemlere göre daha uzun sürdüğü için eğitimde görüntü boyutunu (Image-Size) 448 olarak sabitledik. Tablo 4.4 dikkate alındığında, kesinlik ve duyarlılığın harmonik ortalamasını alan F1 değeri; önerilen YOLOv3 modeli yani Model 4 diğer modellere göre oldukça başarılı iken (en yakın modele en az %3'lik başarı), önerilen YOLOv4'ün başarısı YOLOv4'ten biraz daha iyidir. YOLOv3-SPP modeli, performans açısından diğer modellere göre ise en düşük performansı sergilemektedir.

Tablo 4.5. SUN polip veri seti için transfer öğrenme ile eğitilmiş modellerin sonuçları (IOU: 0.5, küme boyutu 1, güven eşiği: 0.25, ekran kartı: 1xRTX 2080TI).

Model	Görüntü boyutu	Kesinlik %	Duyarlılık %	F1 %	mAP (mAP@.5) %	mAP@.5:.95 %
YOLOv3-SPP	448	94.62	77.05	84.94	86.66	53.71
YOLOv3-SPP	640	95.27	75.01	83.93	84.39	52.54
YOLOv4-CSP	448	94.77	83.65	88.87	98.32	68.90
YOLOv4-CSP	640	92.78	79.87	85.84	97.44	67.53
Önerilen YOLOv4 (Model 3)	448	93.79	85.00	89.18	98.45	68.05
Önerilen YOLOv4 (Model 3)	640	93.01	80.18	86.12	97.89	67.51
Önerilen YOLOv3 (Model 4)	448	93.92	86.39	90.00	98.46	68.08
Önerilen YOLOv3 (Model 4)	640	95.77	80.76	87.62	98.76	68.62

Tablo 4.5, SUN veri seti üzerinde gerçekleştirilen transfer öğrenmenin sonuçlarını göstermektedir. Sonuçlar, sıfırdan eğitime kıyasla en az %2 iyileştirme olduğunu gösteriyor. Örneğin, önerilen YOLOv3 modeli (Model 4) sıfırdan eğitimde %87,56 F1 değerine sahipken, transfer öğrenmede F1 değeri %90.00'dır. Ayrıca modellerin yakınsaması sıfırdan eğitimde yaklaşık 200 epoktan sonra gerçekleşirken, transfer eğitiminde ise 90 epoktan sonra meydana gelmektedir. Başka bir deyişle, transfer öğrenme hem daha hızlı eğitim imkânı sunar hem de daha iyi bir tespit doğruluğu sunar. Burada dikkat edilmesi gereken bir diğer nokta, SUN polip veri setinin, veri setinin yapısından dolayı 640 eğitim görüntüsü boyutu yerine 448'de daha iyi performans göstermesidir. Bu, modelin bu görüntü çözünürlüğünde daha iyi öğrendiğini gösterir. Bu da eğitimde görüntü boyutunun doğru seçilmesi gerektiğini vurgulamakla beraber, bu noktanın göz ardı edilmemesinin gerekliliğini göstermektedir.

Tablo 4.6. SUN polip veri seti için transfer öğrenme ve negatif örneklerle eğitilmiş modellerin sonuçları (IOU: 0.5, küme boyutu 1, güven eşiği: 0.25, ekran kartı: 1xRTX 2080Ti).

Model	Görüntü boyutu	Pozitif negatif örneklerin oranı	Kesinlik %	Duyarlılık %	F1 %	mAP (<u>mAP@.5</u>) %	<u>mAP@.5:.95</u> %
YOLOv3-SPP	448	1:1 (40707 pozitif, 40800 negatif)	96.51	71.60	82.20	81.58	52.38
YOLOv4-CSP	448	1:1(40707 pozitif, 40800 negatif)	96.87	77.53	86.13	98.93	69.81
Önerilen YOLOv4	448	1:1 (40707 pozitif, 40800 negatif)	93.87	77.92	85.15	98.45	68.29
Önerilen YOLOv3	448	1:1(40707 pozitif, 40800 negatif)	95.26	82.19	88.24	98.71	68.36
Önerilen YOLOv4	448	1:0.5 (40707 pozitif, 20000 negatif)	96.98	82.74	89.30	99.19	70.73
Önerilen YOLOv3	448	1:0.5 (40707 pozitif, 20000 negatif)	95.62	83.28	89.03	98.98	69.72
Önerilen YOLOv3	448	1:0.2 (40707 pozitif, 8000 negatif)	96.46	85.27	90.52	99.05	69.20
Önerilen YOLOv4	448	1:0.2 (40707 pozitif, 8000 negatif)	95.09	82.70	88.46	98.67	69.39
YOLOv4-CSP	448	1:0.1 (40707 pozitif, 4000 negatif)	95.66	84.78	89.89	98.88	69.52
Önerilen YOLOv3 (Model 4)	448	1:0.1 (40707 pozitif, 4000 negatif)	96.49	86.48	91.21	99.21	69.22
Önerilen YOLOv4 (Model 3)	448	1:0.1 (40707 pozitif, 4000 negatif)	95.82	83.94	89.49	98.88	69.26

Tablo 4.6, transfer öğrenme ile negatif örneklerin kullanılmasının sonuçlarını göstermektedir. Tablo 10.6'da gösterilen pozitif örneklere göre belirli oranlarda negatif örnekleri kullanarak polip performansına etkisini inceledik. Aynı sayıda pozitif örnek oranlarında kullanılan negatif örneklerin model performansını tam tersi etkilediği görülürken, pozitif örnek oranı en fazla %10 olduğunda negatif örneklerin daha iyi metrik değerleri verdiği sonucuna varılmıştır. Ayrıca modellerin yakınsama hızı transfer öğrenmeye göre %50 daha hızlıdır. Kısaca özetlemek gerekirse, SUN polip veri seti üzerinde yapılan deneyler sonucunda modellerin eğitimi %30 daha kısa sürede tamamlanır ve transfer öğrenme ile birlikte negatif örneklerin kullanımı ile doğruluğu en az %1,5 oranında artırıldığı sonucuna varılmıştır.

4.7.1.2 PICCOLO Veri Seti için Sonuçlar

Bu bölümde önerilen yöntemler için PICCOLO veri seti üzerinde deneysel çalışmalar yaptık. SUN veri setinde yapılan deneylerin aynısı PICCOLO veri setinde de yapılmıştır. Tek fark, PICCOLO veri seti önceden var olan eğitim, doğrulama ve test klasörleriyle birlikte geldiğinden burada herhangi bir değişiklik yapmadık. Yani eğitim klasöründe 2203 resim, doğrulama klasöründe 897 resim ve test klasöründe 333 resimden oluşmaktadır. Bu çalışmadaki deneyde eğitim, doğrulama ve test süreçleri için PICCOLO veri seti tarafından belirlenen kurallar dikkate alınarak eğitim gerçekleştirilmiştir. Ek olarak, negatif örneklerin PICCOLO veri setine katkısını incelemek için SUN veri setinden negatif polip örnekleri kullandık. Negatif örneklerin pozitif oranını (polipli görüntüler) SUN veri seti ile yapılan deneylerde olduğu gibi tuttuk. Bu oranlar da Tablo 10.9'da gösterilmiştir. Diğer çalışmalara benzer şekilde, tüm deneylerde IoU değerini 0,5 ve güven eşiğini 0,25 olarak sabitledik.

Tablo 4.7. PICCOLO polip veri seti için uçtan uca eğitilmiş (sıfırdan eğitim) modellerin sonuçları (IOU: 0.5, küme boyutu 1, güven eşiği: 0.25, ekran kartı: 1xRTX 2080Ti).

Model	Görüntü boyutu	Keskinlik %	Duyarlılık %	F1 %	mAP (mAP@.5) %	mAP@.5:.95 %
YOLOv3-SPP	448	82.33	71.01	76.25	74.52	56.44
YOLOv4-CSP	448	85.80	75.51	80.33	94.91	82.75
Önerilen YOLOv4 (Model 3)	448	88.62	75.51	81.54	96.22	84.10

Önerilen YOLOv3 (Model 4)	448	87.36	77.55	82.16	96.03	82.22
---------------------------	-----	-------	-------	-------	-------	-------

Tablo 4.7'de gösterildiği gibi, önce modeli sıfırdan eğittik. F1 metriği baz alındığında, önerilen YOLOv3 modeli diğer modellerden çok daha başarılı olurken (en yakın modele yani Model 3'e göre yaklaşık olarak %2 daha başarılı), önerilen YOLOv4, orijinal YOLOv3'e göre yaklaşık olarak %6 ve YOLOv4 modeline göre ise yaklaşık olarak %2 daha başarılıdır. YOLOv3-SPP modeli performans açısından diğer modellere göre daha az başarılıdır. Modelin eğitimi uzun sürdüğü için eğitimdeki görüntü boyutunu 448 olarak sabitledik. Kısaca özetlemek gerekirse, önerilen Model 4 en başarılı sonuçları SUN veri setinde verdiği gibi PICCOLO veri setinde vermiştir.

Tablo 4.8. PICCOLO polip veri seti için transfer öğrenme ile eğitilmiş modellerin sonuçları (IOU: 0.5, küme boyutu 1, güven eşiği: 0.25, ekran kartı: 1xRTX 2080TI).

Model	Görüntü boyutu	Kesinlik %	Duyarlılık %	F1 %	mAP (mAP@.5) %	mAP@.5:.95 %
YOLOv3-SPP	448	88.65	72.72	79.89	81.71	68.02
YOLOv3-SPP	640	88.89	72.95	80.13	82.17	69.87
YOLOv4-CSP	448	93.05	71.68	80.98	97.14	85.72
Önerilen YOLOv4 (Model 4)	448	92.01	73.47	81.70	97.34	85.77
YOLOv4-CSP	640	92.50	75.51	83.15	97.92	87.01
Önerilen YOLOv4 (Model 3)	640	92.62	76.57	83.84	98.04	87.20
Önerilen YOLOv3 (Model 4)	448	91.24	77.04	83.54	96.75	84.67
Önerilen YOLOv3 (Model 4)	640	91.28	80.10	85.33	97.36	85.96

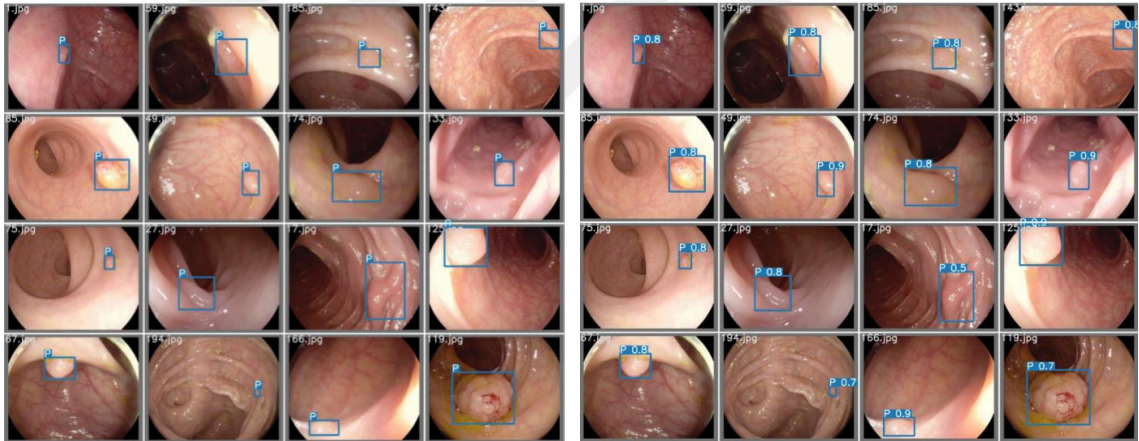
Tablo 4.8, PICCOLO veri setinin transfer öğrenmesi ile deneysel çalışmalarını göstermektedir. SUN veri setinden farklı olarak PICCOLO veri seti daha yüksek çözünürlükte (640) daha iyi sonuçlar verir. Deneylerde önerilen YOLOv3 modeli

YOLOv3-SPP'ye göre yaklaşık olarak %6 daha başarılı sonuçlar verirken, diğer modeller birbirine yakın sonuçlar vermektedir. Kısaca özetlemek gerekirse, önerilen Model 4 en başarılı sonuçları SUN veri setinde verdiği gibi PICCOLO veri setinde de en başarılı sonuçlara sahip model olmuştur.

Tablo 4.9. PICCOLO polip veri seti için transfer öğrenme ve negatif örneklerle eğitilmiş modellerin sonuçları (IOU: 0.5, küme boyutu 1, güven eşiği: 0.25, ekran kartı: 1xRTX 2080Ti).

Model	Görüntü boyutu	Pozitif negatif örneklerin oranı	Kesinlik %	Duyarlılık %	F1 %	mAP (mAP@.5) %	mAP@.5: 95 %
YOLOv3-SPP	448	1:1 (2203 pozitif, negatif)	82.25	71.18	76.31	85.22	74.13
YOLOv4-CSP	448	1:1 (2203 pozitif, negatif)	92.51	72.45	81.26	96.85	85.85
Önerilen YOLOv3 (Model 4)	448	1:1 (2203 pozitif, negatif)	88.62	75.51	81.54	95.35	83.14
Önerilen YOLOv4 (Model 3)	448	1:1 (2203 pozitif, negatif)	87.85	75.79	81.37	94.59	83.08
Önerilen YOLOv3 (Model 4)	448	1:0.5 (2203 pozitif, negatif)	90.39	75.26	82.15	95.13	83.53
Önerilen YOLOv4 (Model 3)	448	1:0.5 (2203 pozitif, negatif)	89.28	74.47	81.45	94.82	82.78
Önerilen YOLOv3 (Model 4)	448	1:0.1 (2203 pozitif, negatif)	92.60	79.85	85.75	96.98	84.97
Önerilen YOLOv4 (Model 3)	448	1:0.1 (2203 pozitif, negatif)	91.49	78.12	84.27	96.98	84.97

Tablo 4.9, PICCOLO veri seti için transfer eğitime dayalı negatif örneklerin kullanılmasının sonuçlarını göstermektedir. SUN veri seti ile aynı kurallar izlenerek deneyler yapılmıştır. Aynı sayıda pozitif polip oranlarında kullanılan negatif örneklerin performansı olumsuz etkilediği gözlemlendi. Deneylerde, en iyi sonuçların, negatif örneklerin %10'u kullanıldığında elde edildiği gözlemlendi. Ayrıca eğitimde negatif örnekler kullanıldığında modelin transfer yalınlaştırmaya göre daha hızlı yakınsama sağladığı ve performansta en az %2 iyileşme olduğu gözlemlenmiştir. Yine aynı şekilde önerilen Model 4 (Önerilen YOLOv3) en başarılı sonuçları SUN veri setinde verdiği gibi PICCOLO veri setinde de en başarılı sonuçlara sahip model olmuştur. Başka deyişle Model 4 büyük veri setleri için bütün veri setlerinde ve çeşitli konfigürasyonlara rağmen en başarılı metriklerle sunan yani en iyi tespiti yapan modeldir. Şekil 4.15 SUN veri seti üzerinde eğitilen en iyi modelin, Etis-Larib test veri setine ait bazı görüntüler üzerindeki tespit başarımını göstermektedir.



Şekil 4.15. Soldaki şekil Etis-Larib veri setinin temel gerçeğini gösterirken, sağdaki şekil ise en iyi modelin (Önerilen YOLOv3 modeli, yani, Model 4) tahminini ve tahminlerin güven puanlarını göstermektedir.

Şekil 4.15 düşünüldüğünde, önerilen YOLOv3 modelinin yani Model 4'ün çok başarılı sonuçlar ortaya koyduğu görülmektedir. Sonuç olarak önerilen YOLOv3 algoritmasının performansı tüm veri setlerinde YOLOv3, YOLOv4 ve YOLOv4-CSP (Scaled-YOLOv4) algoritmalarından daha iyi performans göstermiş ve negatif örneklerle transfer öğrenmenin kullanılması polip tespit performansını yaklaşık olarak %5'lik ortak bir artış sağlamıştır. Aynı şekilde, modelin polip tespit performansının transfer

öğrenme, negatif örnekler ve çok miktarda polip görüntüsü kullanımı ile önemli ölçüde arttığı görülmüştür

4.8. Özel Veri Seti için Sonuçlar

Bu bölümde daha önce hastane ortamında toplamda 400 eşsiz poliple beraber, 100.000 adet polip görüntüsü ve 200.000 adet negatif yani polip içermeyen kolonkopsi görüntülerden elde edilen özel (yerli) veri seti üzerinde modellerin başarımı incelenmiştir. Ayrıca, YOLOv3 ve YOLOv4 ile beraber önerilen modellerin (Model 1, Model 2, Model 3 ve Model 4) başarımlarına bakılmıştır. Daha önce belirtildiği üzere Model 1 ve Model 2 sığ modeller iken, Model 3 ve Model 4 ise daha derin CNN yapılarından oluşmuştur. Bu yapılardan kasıt ise filtre sayısı, katman sayısı ve bunların genişliği ile beraber farklı yapıların beraber kullanılmasıdır. Elimizde yer alan özel veri setinden oluşan görüntülerin %70'i eğitim için, %20'si validasyon ve geriye kalan %10'u ise sadece test işlemlerinde kullanılmıştır. Burada önemli olan nokta, eğitimde yer alan görüntülerdeki hastalar ile test işlemlerinde yer alan hastalar birbirinden farklıdır. Başka bir deyişle, eğitim ve test verileri birbirilerinden bağımsızdır ve hastalar kesinlikle aynı değildir. Böylece, bu şekilde eğitilen modeller genelleme kabiliyeti daha yüksek olmaktadır. Ayrıca eğitim ve test verileri farklı veri setleri gibi davranmakla beraber daha objektif sonuçlar elde edilmektedir. Literatürde yer alan çoğu çalışma çapraz doğrulama ya da eğitim ve test verilerinde hastaların aynı olup olmamasına bakmıyorlar, böylece eğitim ve test verilerindeki görüntüler aynı olduklarından dolayı sonuçlar yüksek çıkmakla beraber objektifliği azalmaktadır. Özellikle bu konuya dikkat ederek test ve eğitim görüntülerini ayırarak deneysel sonuçları elde ettik. Model eğitiminde negatif yani polip içermeyen görüntülerde eklenerek eğitim süreci tamamlanmıştır. Daha önceki deneylerde negatif görüntülerin modelin genelleme kabiliyeti artırdığını, doğruluğu da önemli derece artırdığı sonucuna varılmıştı. Burada, diğer deneysel sonuçlarla aynı negatif görüntü oranını yani pozitif görüntülerin 1/10 katı kadar negatif görüntü kullandık. Buna ek olarak aktivasyon fonksiyonu olarak SiLU ve CIOU'yu ise kayıp fonksiyonu olarak kullandık. Eğitim giriş görüntü çözünürlüğü ise 448x448 olarak seçilmiştir, çünkü SUN veri setinde bu çözünürlük en iyi sonucu vermişti. Kısaca SUN ve PICCOLO gibi büyük veri setlerine

uygulanan bütün teknikler özel veri seti içinde aynen uyguladık. Özel veri setine ait deneysel sonuçlar Tablo 4.10’da gösterilmiştir.

Tablo 4.10. Özel veri seti için eğitilmiş modellerin sonuçları (IOU: 0.5, küme boyutu 1, güven eşiği: 0.25, ekran kartı: 1xRTX 2080TI).

Model	Görüntü boyutu	Kesinlik %	Duyarlılık %	F1 %	mAP (mAP@.5) %	mAP@.5:.95 %
YOLOv3	448	95.88	92.41	94.11	97.25	71.33
YOLOv3-SPP	448	95.98	93.23	94.59	97.68	72.68
YOLOv4	448	96.19	92.96	94.55	98.09	72.84
YOLOv4-CSP	448	97.15	93.27	95.17	98.88	74.29
Model 1-scratch	448	90.22	88.78	89.49	86.21	61.51
Model 1-transfer	448	91.98	90.17	91.06	88.11	62.31
Model 1-transfer+negatif	448	93.18	90.87	92.01	89.85	64.02
Model 2	448	90.77	88.10	89.41	86.15	60.98
Model 2-transfer	448	90.32	89.91	90.11	86.19	60.55
Model 2-transfer+negatif	448	91.35	90.15	90.75	86.92	60.48
Model 3	448	96.09	94.71	95.39	94.29	72.17
Model 3-transfer	448	97.25	95.69	96.46	95.98	73.78
Model 3-transfer+negatif	448	98.14	95.88	96.99	96.77	74.51
Model 4	448	97.75	95.89	96.81	97.94	76.63
Model 4-transfer	448	98.07	96.37	97.21	98.34	77.98
Model 4-transfer+negatif	448	98.93	97.47	98.46	98.87	79.31

Yukarıdaki Tablo 4.10’da görüldüğü üzere, önerilen derin modeller diğer bütün modellerden daha başarılı sonuçlar vermiştir. Önerilen sığ modellerden olan Model 1 ve Model 2 ise YOLOv3 ve YOLOv4’ün gerisinde kalmıştır. Daha öncede belirtildiği üzere bu modeller sadece küçük veri seti setleri için optimize edilmiş modellerdir. Yine

tabloya bakıldığında yüksek sonuçları veren model ise YOLOv3 tabanlı olan Model 4'e aittir. Daha sonra ise YOLOv4 tabanlı Model 3 diye devam ediyor. Burada en önemli husus şudur ki büyük veri setlerindeki özelliklerin daha iyi çıkarılmasında derin modeller daha iyi bir öğrenme gerçekleştirirken, sığ modellerde ise yeteri kadar filtre, katman ve sinir ağının olmamasından kaynaklı daha az başarımlar ortaya çıkıyor. Çıkarım hızı ya da saniyede üretilen çerçeve oranına bakıldığında ise Model 1 en yüksek FPS değeri sunarken, Model 3 ise yine oldukça hızlı olduğu görülmektedir. Kısaca böyle bir sistemde hem hız hem de doğruluk açısından Model 3 en uygun model olarak karşımıza çıkmaktadır.



5. BÖLÜM

GERÇEK ZAMANLI POLİP TESPİTİ İÇİN ÖNERİLEN DERİN ÖĞRENME MODELLERİNİN OPTİMİZASONU

5.1. Giriş

Bu bölümde eğitici birimde yer alan hiper-parametrelerin optimizasyonu için kullanılan genetik algoritma (GA) ile beraber yapay arı koloni algoritmasından (ABC) bahsedilecektir. Bu algoritmalar YOLO algoritmalarında kullanılan önemli parametrelerin optimize edilerek başarımı daha yüksek modellerin elde edilmesini sağlamaktır. YOLO algoritmalarında iki temel hiper-parametre grubu olduğu söylenebilir. Birinci grup, öğrenme hızı, momentum, ağırlıkların azalması gibi temel hiper-parametreleri içerirken, ölçek, kesme, çevirme gibi hiper-parametreler, veri artırma teknikleriyle yani veri seti ile ilgili hiper-parametrelerdir. Böylece veri artırma tekniklerinde daha yüksek doğruluk için değerlerin optimize edilmesini sağlar. YOLOv4 ve YOLOv3 algoritmalarında optimize edilmeye çalışılan standart değerler Tablo 5.1’de gösterilmiştir. Tabloda her bir parametrenin adı, optimizasyonda kullanılacak alt ve üst limit değerleri verilmiştir. Bu değerler ABC ve GA gibi optimizasyon algoritmalarıyla optimize edilerek her bir veri seti için en ideal hiper-parametre değerleri elde edilmededir.

Tablo 5.1. Scaled-YOLOv4 nesne algılama algoritmasının hiper-parametre değerleri.

Hiper-parametre adı (temel hiper-parametreler)	Alt değer	Üst değer	Hiper-parametre adı (veri seti hiper-parametreleri)	Alt değer	Üst değer
initial learning rate	1e-5	1e-1	image HSV-Hue (fraction)	0.0	0.1
momentum	0.6	0.98	image HSV-Saturation (fraction)	0.0	0.9

optimizer weight decay	0.0	0.001	image HSV-Value (fraction)	0.0	0.9
GIoU loss gain	0.02	0.2	image rotation (+/- deg)	0.0	45.0
classification loss gain	0.2	4.0	image translation (+/- fraction)	0.0	0.9
classification BCELoss	0.5	2.0	image scale (+/- gain)	0.0	0.9
objectness loss gain	0.2	4.0	image shear (+/- deg)	0.0	10.0
objectness BCELoss	0.5	2.0	image perspective (+/- fraction)	0.0	0.001
IoU training threshold	0.1	0.7	image flip up-down (probability)	0.0	1.0
anchor-multiple threshold	2.0	8.0	image flip left-right (probability)	0.0	1.0
focal loss gamma	0.0	2.0	image mix-up (probability)	0.0	1.0

Buradaki her bir hiper-parametrenin ayrı bir önemi olmakla beraber, öğrenme adımı momentum ve ağırlık düşümü gibi değerler daha önemlidir denilebilir. Sırasıyla bu değerlerin açıklamaları şu şekildedir. Lr0 değeri ilk öğrenme oranıdır. Bu oran SGD optimizeri için $1e-2$ gibi bir değer iken, Adam optimizeri için ise $1e-3$ 'dür. Lrf değeri ise nihai OneCycleLR öğrenme oranı ($lr0 * lrf$) olarak formülize edilmiştir. Momentum değeri, gradyan tabanlı optimizasyon tekniklerinin yakınsamasını hızlandırmak için yaygın olarak kullanılan bir stratejidir. Weight decay değeri ise Adam ya da SGD optimizelerin ağırlık düşüşünü belirtir. Warm up ise yani ısınma, erken eğitim örneklerinin öncelikli etkisini azaltmanın bir yoludur. Diğer değerler ise genelde sabit olmakla beraber çok fazla etkisi yoktur. Daha fazla açıklama YOLOv4 orijinal makalesinde belirtilmiştir. Bu değerler GA ve ABC algoritmalarının yardımı ile optimize edilerek polip tespiti için en ideal değerler sağlanarak, eğitim bu değerler üzerinde ince ayar olarak ya da yeniden eğitilebilir. Ama baştan eğitimle ince ayar eğitimi nerdeyse aynı doğruluğu vermektedir. Yine de ideal olan her iki eğitimin gerçek uygulama öncesinde denenerek daha ideal bir sistemin geliştirilmesine planlanmalıdır. ABC ve GA algoritmalarının hiper-parametre optimizasyonunda kullanıldığı diğer önemli yer ise veri artırma teknikleri ile ilgilidir. Veri artırmada kullanılan teknikler

alana göre farklı şekillerde sonuçlar verdiklerinden dolayı bu tekniklere ait değerlerin optimize edilmesi ile başarımlar oldukça yükselebilir.

5.2. Genetik Algoritma

Bir optimizasyon yöntemi, yani arama ve eniyileme yöntemlerinden biri olan GA, doğada gözlemlenen evrimsel sürece benzerlik taşıyan ve çalışan güçlü bir optimizasyon algoritmasıdır. Genetik algoritmadaki mantık arama uzayındaki en iyi bireyin hayatta kalma ilkesine dayanan bütünsel bir çözüm sunar. John Holland tarafından ortaya atılan GA yine aynı kişi tarafından 1975 yılında yaptığı çalışmaları “Adaptation in Natural and Artificial Systems” adlı kitabında toplayarak GA’nın gelişme sürecini başlatmıştır. Başka deyişle, bu algoritma, bir sonraki neslin yavrularını üretmek için üreme için en uygun bireylerin seçildiği doğal seçim sürecini yansıtır. Ayrıca, problemlere sadece tek bir çözüm üretmez yani farklı çözümlerden oluşan bir çözüm kümesi üreterek bütünsel bir arama gerçekleştirir.

Optimizasyon algoritmalarında doğal seleksiyon önemli bir yer kaplar. Doğal seçim süreci, bir popülasyondan en uygun bireylerin seçilmesiyle başlar. Ebeveynlerin özelliklerini miras alan ve bir sonraki nesle eklenecek yavrular üretirler. Ebeveynler daha iyi zindeliğe sahipse, yavruları ebeveynlerden daha iyi olacak ve hayatta kalma şansı daha yüksek olacaktır. Bu süreç yinelenir ve sonunda en uygun bireylere sahip bir nesil bulunur. Bu kavram bir arama problemi için uygulanabilir. Genetik bir algoritmada beş aşama göz önünde bulundurulur. Bunlar sırasıyla; başlangıç popülasyonu, amaç fonksiyonu (fitness fonksiyonu), seleksiyon (seçilim), çaprazlama ve mutasyondur.

Başlangıç popülasyonundaki süreç popülasyon adı verilen bir grup bireyle başlar. Her birey, çözmek istediğiniz soruna bir çözümdür. Bir birey, Genler olarak bilinen bir dizi parametre (değişken) ile karakterize edilir. Genler, bir Kromozom oluşturmak için bir dizi halinde birleştirilir. Bir genetik algoritmada, bir bireyin gen kümesi, bir alfabe cinsinden bir dize kullanılarak temsil edilir. Genellikle ikili değerler kullanılır (1'ler ve 0'lar dizisi). Amaç fonksiyonu, bir bireyin ne kadar uygun olduğunu belirler (bir bireyin diğer bireylerle rekabet etme yeteneği). Her bireye bir uygunluk puanı verir. Bir bireyin üreme için seçilme olasılığı, uygunluk puanına bağlıdır. Seleksiyon aşaması fikri, en

uygun bireyleri seçmek ve genlerini bir sonraki nesle aktarmalarına izin vermektir. Uygunluk puanlarına göre iki çift birey (ebeveyn) seçilir. Uygunluğu yüksek bireylerin üreme için seçilme şansları daha fazladır. Çaprazlama, genetik algoritmanın en önemli aşamasıdır. Çiftleştirilecek her bir ebeveyn çifti için genler içinden rastgele bir çaprazlama noktası seçilir. Mutasyon ise, oluşan bazı yeni yavrularda, genlerinden bazıları düşük rastgele olasılıkla mutasyona tâbi tutulabilir. Bu, bit dizisindeki bazı bitlerin çevrilebileceği anlamına gelir. Mutasyondaki amaç popülasyon içindeki çeşitliliği korumak ve erken yakınsamayı önlemektir. Algoritma, popülasyon yakınsadığında sona erer (önceki nesilden önemli ölçüde farklı olan yavrular üretmez). GA'nın temel adımları Algoritma 1'de gösterilmiştir.

Algoritma 1 Basit genetik algoritmanın temel adımları

- 1: İlk popülasyonu oluştur
 - 2: Amaç fonksiyonunu hesapla
 - 3: **REPEAT**
 - 4: Seleksiyon
 - 5: Çaprazlama
 - 6: Mutasyon
 - 7: **UNTIL** (sonlandırma kriteri)
-

5.3. Yapay Arı Koloni Algoritması

ABC algoritması, Karaboğa tarafından 2005 yılında sunulan popülasyon temelli sezgisel algoritmalarından biridir. ABC algoritması, doğadaki gerçek bal arılarının yiyecek arama davranışından ilham alınarak modellenmiştir. Algoritmadaki temel amaç, her bir besin kaynağının probleme olası bir çözümle örtüşmesi, böylece en iyi besin kaynağının en iyi sonucu vermesidir. ABC algoritmasında toplayıcı arılar iş bölümlerinde üç göreve göre ayrılır: işçi arılar, gözcü arılar ve kâşif arılar. İşçi arılar hafızalarında tuttukları kaynakları araştırırken, gözcü arılar işçi arılardan aldıkları bilgilere göre potansiyel olarak daha iyi besin kaynağı alanları ararlar. Öte yandan kâşif arılar, yeni yiyecek alanlarında rastgele aramalar yaparak küresel bir arama yapmaktan sorumludur. ABC algoritmasının temel adımları Algoritma 2'de verilmiştir.

Algoritma 2 ABC algoritmasının temel adımları

1: Başlangıç yiyecek kaynakları

2: Değerlendirme aşaması

3: **REPEAT**

4: İşçi arı fazı

5: Gözcü arı fazı

6: Kâşif arı fazı

7: **UNTIL** (sonlandırma kriteri)

ABC algoritmasında ilk adımda kontrol parametre değerleri belirlenir ve ardından gıda kaynaklarının başlangıç konumları rastgele belirlenir. İlk rastgele popülasyon, değişkenlerin alt ve üst sınırlarına göre oluşturulur ve Eşitlik 6.1'deki gibi belirlenir.

$$x_{ij} = x_j^{\min} + \text{rand}(0,1) (x_j^{\max} - x_j^{\min}) \quad (5.1)$$

Eşitlikte, $x_j^{\max}$ üst sınırdır, $x_j^{\min}$ alt sınırdır, x_{ij} çözüm vektörüdür ($j=1,2,...,D$) ve $i=1,2,...,SN$. Burada D parametre sayısı, SN ise çözüm sayısıdır. Görevli arı, gözcü ve kâşif arı aşamaları, en iyi çözüm hafızada tutularak bir durdurma kriteri sağlanana kadar tekrarlanır. İşçi arı evresi, mevcut çözümlerin komşuluğunu araştırarak yeni çözümler üretir ve Eşitlik 6.2'deki formülü kullanır.

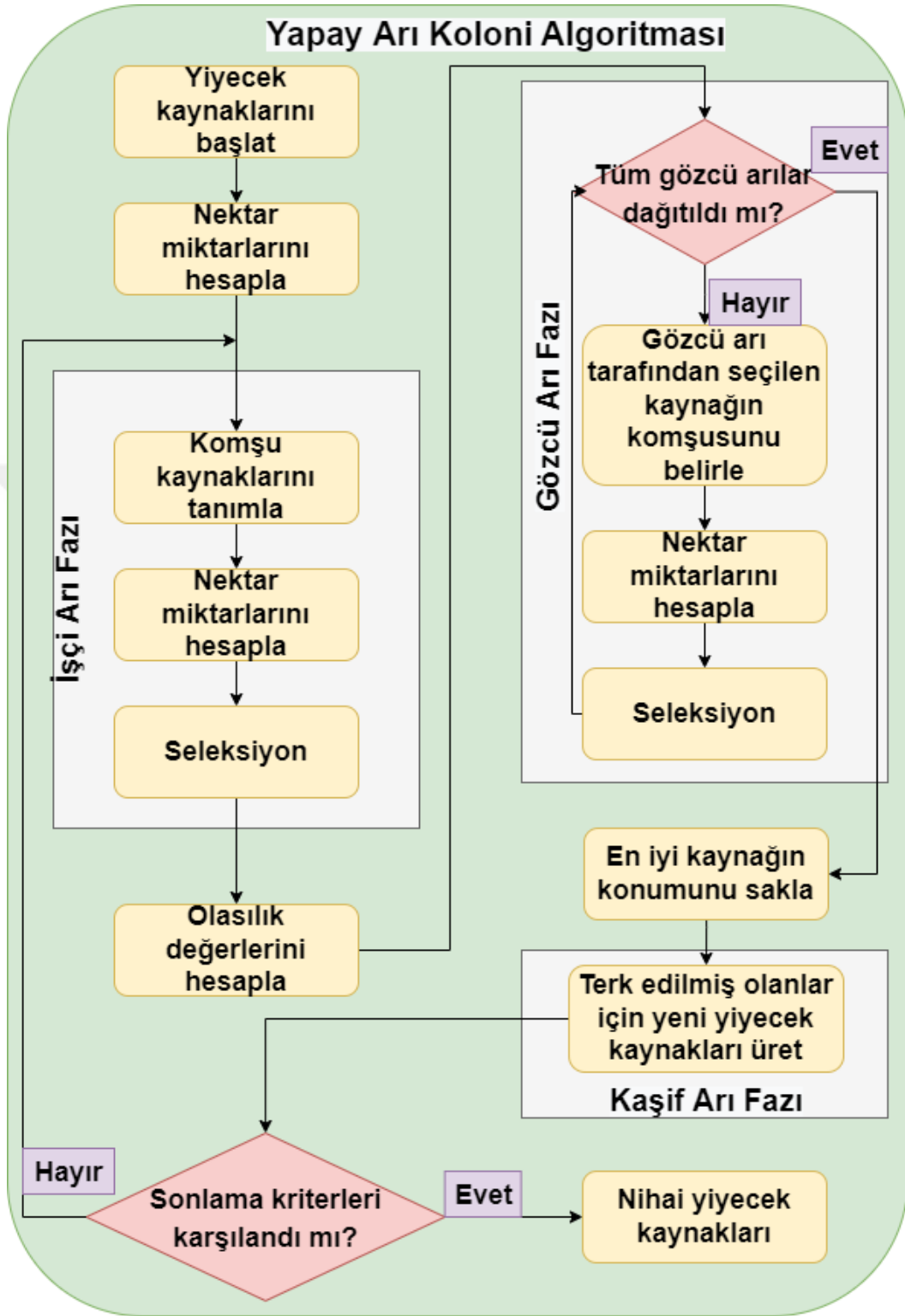
$$x'_{ij} = x_{ij} + \phi_{ij} (x_{ij} - x_{kj}) \quad (5.2)$$

Eşitlikte, x' yeni aday çözüm vektörü, ϕ_{ij} ise $[-1, 1]$ arasında rastgele bir değerdir ve x_k rastgele seçilen bir komşu çözüm vektörüne karşılık gelir. Eşitlik 9.2'ye göre, yeni bir çözüm üretildikten sonra açgözlü seçim mekanizması ile yeni ve mevcut çözümler arasından daha iyi olan çözüm seçilir. Yeni çözüm, mevcut çözümden daha iyi bir uygunluk değerine sahipse, yeni çözüm mevcut çözümün yerini alır, aksi takdirde mevcut çözüm korunur ve karşılık gelen sayaç bir artırılır. Bu sayacın amacı, besin kaynağının tükenip tükenmediğine karar vermektir. Gözcü arı aşamasında, işçi arı aşamasında olduğu gibi Eşitlik 6.1 kullanılarak yeni bir çözüm üretilir. Gözcü bir arının besin kaynağını belirlemek için olasılıksal bir seçim yöntemi kullanılır. Bu, daha yüksek olasılıkla daha iyi çözümlerin seçilmesini sağlar ve düşük kaliteli çözümler etrafında

arama şansını azaltırken, bu çözümler etrafında yerel arama yapılmasına izin verir. Bireylerin olasılık hesaplamaları Eşitlik 6.3 ile yapılır.

$$P_i = \frac{Fitness_i}{\sum_{i=1}^{NP} Fitness_i} \quad (5.3)$$

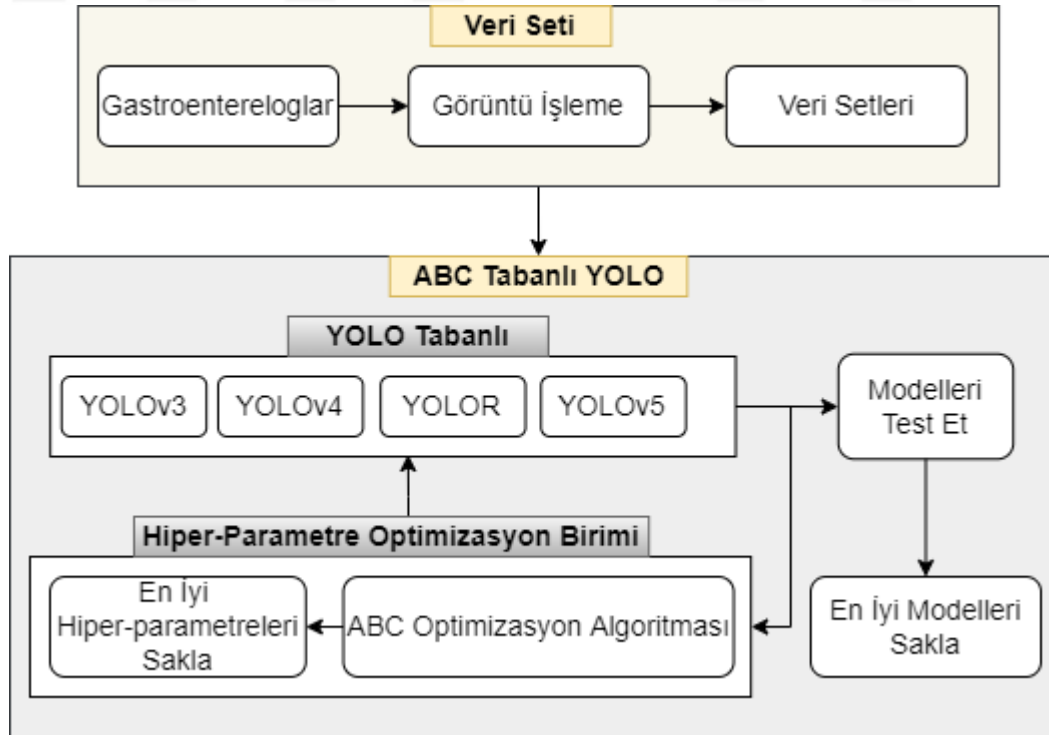
Eşitlikte, p_i değeri i çözümünü seçme olasılığıdır ve uygunluğu çözümün kalitesine karşılık gelir. İşçi arı aşamasında olduğu gibi burada da açgözlü seçim mekanizması uygulanmakta ve iyileşmeyen çözümlerin ilgili sayaç (deneme) değerleri birer birer artırılmaktadır. Arıların nektar toplaması sonucunda besin kaynakları zamanla azalır ve sonunda tükenir. Bu, çözümün artık geliştirilemeyeceği gerçeğine karşılık gelir. Kaynağın tükenip tükenmediğinin tespiti "limit" adı verilen kontrol parametresi ile belirlenir. Bir çözümün deneme sayacı, x_i , "sınırı" aştığında, ilgili kaynak terk edilir. Daha sonra o kaynağın arısı kâşif arı olarak rastgele arama yapar. Sürü zekasına dayalı algoritmalarından biri olan ABC algoritması, doğası gereği az sayıda kontrol parametresine sahip olup, ABC algoritmasının diğer algoritmalara göre basit ve kullanımı oldukça esnek olmasına olanak sağlamaktadır. Besin kaynaklarının sayısı, maksimum döngü sayısı ve sınır değeri, ABC'nin temel kontrol parametreleridir. ABC algoritmasının temel akış şeması Şekil 5.1'de gösterilmiştir.



Şekil 5.1. ABC algoritmasının temel akış şeması.

5.4. Önerilen Hiper-parametre Optimizasyon Yöntemi

YOLO algoritmaları, gerçek zamanlı ve hızlı nesne algılama için tasarlanmış bir dizi uçtan uca derin öğrenme modelidir. YOLO ve diğer derin öğrenme algoritmalarında olduğu gibi, model öğrenme aşamasına başlamadan önce ayarlanması gereken modele özgü hiper-parametreler vardır. Modelin bu parametrelerle eğitilmesi ağın davranışında ve performansında oldukça etkilidir. Nesne algılama algoritmaları genellikle MSCOCO için ilk hiper-parametreleri kullanır, ancak polip tespiti veya akciğer nodülü tespiti gibi farklı görevler için bu hiper-parametreler optimal olmayabilir. Kullanıcılar genellikle bu hiper-parametreleri kendi deneyimlerinden başlatırlar, ancak yine de en iyi sonucu elde edemeyebilirler. Manuel ayarlanan hiper-parametre optimizasyonunda, arama alanı yüksek boyutlu olduğunda, en iyi hiper-parametrelere ulaşmak çok fazla çaba ve ayrıca daha fazla zaman gerektirir. Bu gibi nedenlerle, hesaplama maliyetini ve kullanıcı katılımını en aza indirmek için otomatik temelli hiper-parametre optimizasyonu kaçınılmazdır. Burada, Şekil 5.2'de gösterilen YOLO algoritmalarında hiper-parametreleri optimize etmek için ABC algoritmasının uygulamasını sunuyoruz.



Şekil 5.2. Önerilen ABC Tabanlı YOLO algoritmasının yapısı.

Şekil 5.2'de görüldüğü gibi önerilen yöntem veri seti ve ABC tabanlı YOLO algoritmasından oluşmaktadır. Optimizasyon algoritması için ABC, YOLO tabanlı algoritma için Scaled-YOLOv4 algoritması seçilmiştir. Scaled-YOLOv4 algoritması birçok model içerdiğinden diğer versiyonlara göre daha geniş deneysel çalışma yapılabilmektedir. Algoritma 3, YOLO algoritmalarında hiper-parametre optimizasyonu için ABC algoritmasının nasıl kullanıldığını daha ayrıntılı olarak gösterir.

Algoritma 3 ABC'nin YOLO algoritmaları için hiper-parametre optimizasyon adımları

- 1: Veri kümesini yükle
 - 2: Veri kümesini eğitim ve test olarak ayır
 - 3: Amaç fonksiyonun mAP olarak değiştir
 - 4: Önceden eğitilmiş ağırlıkları yükle
 - 5: Varsayılan hiper-parametreleri başlat
 - 6: YOLO tabanlı algoritmaların eğitimini başlat (hızlı eğitim için küçük model)
 - 7: En iyi modeli kaydet
 - 8: ABC algoritmasının parametrelerini başlat
 - 9: En iyi modeli yükle
 - 10: REPEAT**
 - 11: ABC algoritması ile en iyi modeli birkaç defa ince ayar için eğit
 - 12: **UNTIL** (sonlandırma kriteri)
 - 13: Optimize edilmiş hiper-parametre değerlerini al
 - 14: Optimize edilmiş hiper-parametrelerle YOLO tabanlı algoritmayı yeniden eğit veya ince ayar yapın
 - 15: Modellerin performansını değerlendir.
-

Algoritma 3'te gösterildiği gibi, önerilen yöntemin ilk adımları veri setini sisteme yüklemektir. Yüklenen veri seti daha sonra eğitim ve test verisi olarak ayrılır. Buraya kadar olan kısım veri seti ile ilgili iken sonraki kısım YOLO algoritmasında yapılan değişikliklerden oluşmaktadır. Önceden eğitilmiş ağırlıklar, uygunluk işlevi olarak mAP güncellemesinden sonra yüklenir. Aynı zamanda, varsayılan hiper-parametreler başlatılır. Ardından YOLOv4s modeli eğitime başlanır. Burada YOLOv4s modelini seçmemizin nedeni, hiper-parametre optimizasyonunun zaman alan bir süreç olması ve küçük ölçekli modellerin eğitiminin diğer büyük modellere göre zaman kazandırmasıdır. Model eğitildikten sonra, ABC optimizasyonu için en iyi ağırlıklar tutulur. Bir sonraki kısım, ABC algoritması ve YOLO algoritmasının birleşimine

dayanmaktadır. İlk olarak, ABC algoritmasının parametreleri başlatılır (popülasyon büyüklüğü, limit değeri, deneme değeri ve toplam değerlendirme sayısı gibi). ABC algoritmasının YOLOv4 algoritmasının hiper-parametrelerini optimize etmesi için bu parametreler için alt ve üst değerler tanımlanır. Ardından YOLOv4s modelinin en iyi ağırlığı yüklenir. ABC algoritması, belirtilen dönem sayısı ile YOLOv4 algoritmasının en iyi modeline ince ayar yapmaya çalışır. Bu işlem ABC algoritmasının çalışma prensibine göre yapılır. ABC algoritması tarafından kullanılan uygunluk fonksiyonu, nesne algılama algoritması tarafından üretilen ve Eşitlik 5.4 ile verilen mAP değerine bağlı olarak tanımlanır.

$$fitness_i = mAP_i \quad (5.4)$$

Eşitlikte, i çözüm vektörünün sayısıdır. Her çözüm vektörü, Tablo 6.1'de listelenen olası hiper-parametre değerlerinden oluşur ve Eşitlik 5.5'te verilen bir dizi ile temsil edilir.

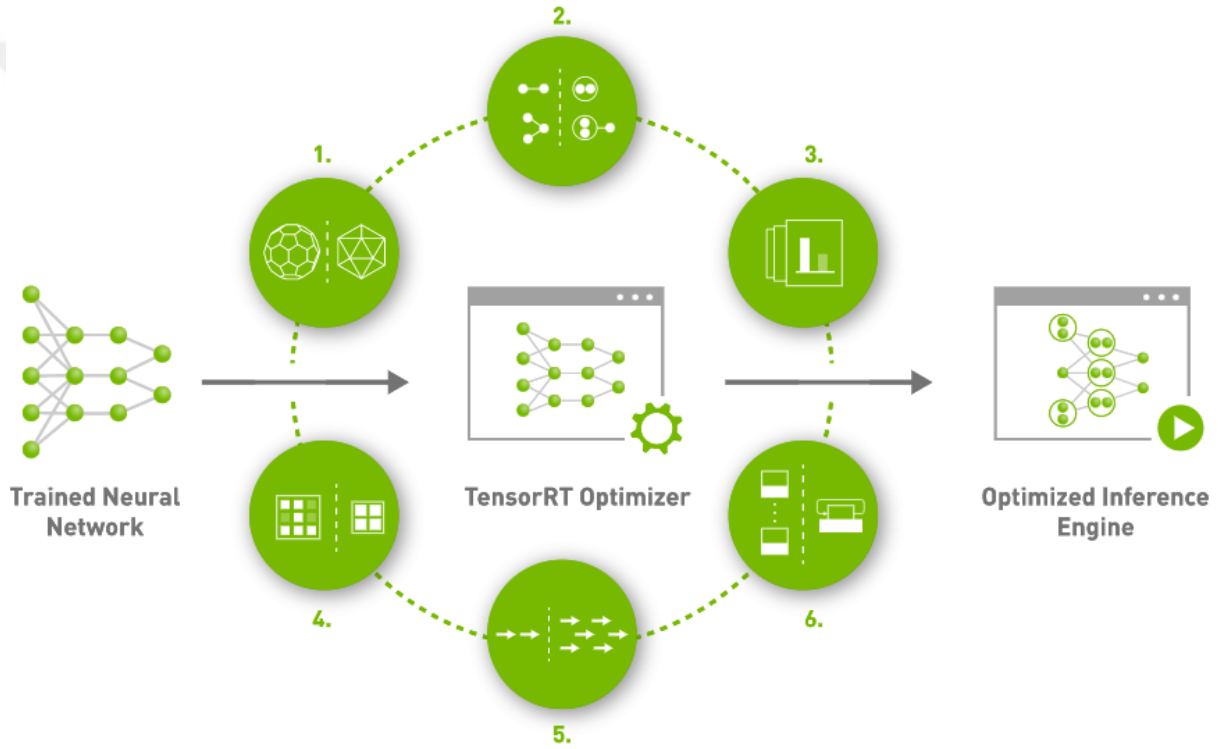
$$A \text{ solution} = [x_1, x_2, x_3, \dots, x_i, \dots, x_N] \quad (5.5)$$

Eşitlikte, x_i bir hiper-parametredir (YOLO tabanlı algoritmalarda optimize edilecek) ve N ise toplam hiper-parametre sayısıdır. Olası bir çözümdeki hiper-parametre değerleri, YOLOv4'lerle elde edilen en iyi modelle yalnızca birkaç dönem için eğitilir (ince ayar yapılır). Bu döngü, bir sonlandırma kriteri karşılanana kadar devam eder. Döngü sonunda en iyi besin kaynağı yani en iyi hiper-parametreler ABC algoritması bazında üretilir. Böylece en iyi hiper-parametrelerin başarısı değerlendirilir. En iyi hiper-parametreler daha sonra YOLOv4 modeliyle yeniden eğitilebilir veya ince ayar yapılabilir. Son bölümde ise orijinal YOLOv4 ile elde edilen modelin performansı ve ABC ile elde edilen YOLOv4 modelinin performansı karşılaştırılmıştır. Burada aynı işlemler YOLOv5, YOLOR, YOLOv3 gibi herhangi bir YOLO tabanlı algoritma kullanılarak gerçekleştirilebilir. Bu da önerilen yöntemin ne kadar esnek olduğunu göstermektedir. Hiper-parametre optimizasyonuna ait sonuçlar ve başarımlar, deneysel sonuçlar adlı bölümde verilmiştir.

5.5 Model Geliştirme

Bu bölümde, model geliştirme ile ilgili uygulanan tekniklerden bahsedilmiştir. Bir model eğitildikten sonra hız ve doğruluğunun artırılması için birkaç teknik uygulanabilir.

Burada biz modelin çıkarım hızını yani FPS oranını artırmak için NVIDIA TensorRT uygularken, modelin doğruluğunu artırmak için ise topluluk öğrenmeyi uyguladık. Eğitimli derin öğrenme mimarilerimizin gerçek zamanlı uygulamalarda daha iyi performans sunmasını sağlamak için Nvidia TensorRT platformunu kullandık. TensorRT, Nvidia GPU'larında yüksek performanslı çıkarım uygulamak için NVIDIA tarafından önerilen bir platformdur. TensorRT tabanlı uygulamalar, çıkarım sırasında CPU platformlarından 40 kata kadar daha hızlı performans gösterir. NVIDIA TensorRT mimarisini gösteren yapı Şekil 5.3'te gösterilmiştir.



Şekil 5.3. Derin öğrenme modelleri optimize etmek için TensorRT tarafından gerçekleştirilen adımlar.

TensorRT, başarılı optimizasyon teknikleriyle eğitilmiş derin öğrenme modellerini daha düşük hassasiyet için daha yüksek doğrulukla kalibre edebilir. Bu işlemler ağ mimarisini basitleştirmeye odaklanır ve herhangi bir derin öğrenme modeli için yapılabilir. Yinelenen öğeler varsa, bunlar birleştirilir, ağın boyutu küçülür ve çıkarım performansı artar. Ayrıca, INT8 ve FP16 gibi farklı optimizasyonlar sağlayarak uygulamaların üretim dağıtımını başarıyla sağlar. Bu işlemleri sadece birkaç model üzerinden deneyerek NVIDIA TensorRT ile polip tespiti çıkarım hızının artırılabilirliğini gösterdik. Böylece bu tür geliştirmeler ile özellikle klinik

uygulamalarında daha yüksek performanslı sistemlerin olabileceğini gösterdik. Özellikle polip tespiti gibi gerçek zamanlı bir gereksinim için böyle teknikler kritik önem taşımaktadır. Eğitilen modellerin kolonoskopi cihazına entegre edilmeden önce, yüksek çıkarım hızı için (gerçek zamanlı uygulamalar) NVIDIA TensorRT ciddi bir öneme sahiptir. Bu yapıya ait sonuçlar ise deneysel sonuçlar adlı bölümde verilmiştir.

5.6. Topluluk Öğrenme

Topluluk öğrenme teknikleri, bir sonucu tahmin etmek için birçok model tarafından üretilen tahminleri birleştirmeye odaklanır. Tek bir model olarak hareket eder ve çalışır. Genel olarak, topluluk öğrenme teknikleri kullanılarak genelleme hatası en aza indirilir. Bu çalışmada, modelin doğruluğunu artırmak için NMS, ortalama ve max topluluk tekniklerini uyguladık. NMS topluluğu, her modelden kutu koordinatlarını alır ve bunları bir listeye ekler, böylece farklı modeller birleştirilebilir ve mAP değerleri artırılabilir. Ortalama ve maksimum birleştirme ise, birleştirilmiş modellerin güven puanlarına göre kutu koordinatlarının sırasıyla ortalamasını veya maksimumunu alarak algılamayı artırmayı amaçlamaktadır. NMS topluluk öğrenimini uygularken, aynı zamanda gerçek zamanlı uygulama için çıkarım süresinin de dikkate alınması gerektiğini belirtmek önemlidir. Daha yüksek performans elde etmek için farklı parametreler, çözünürlükler ve modeller için eğitilmiş bir YOLOv3 ve YOLOv4 modellerinin koleksiyonunu kullandık. Burada, birçok YOLO modellerini tek bir model gibi davranmasını sağlayarak başarıyı artırdık, fakat topluluk öğrenme tekniklerinde başarıyı artarken, gerçek zamanlı tespitte de kazanımlar yerine kayıplar gerçekleşir. Topluluk öğrenme teknikleri uyguladıktan sonra elde edilen sonuçlar, deneysel sonuçlar adlı bölümde detaylandırılmıştır.

5.7. Hiper-parametre Optimizasyonu için Sonuçlar

Bu bölümde, nesne tespit algoritmalarının tespit başarımını büyük ölçüde etkileyen hiper-parametre optimizasyonuna ait sonuçlar yer almaktadır. YOLO algoritmalarında optimize edilecek hiper-parametre sayıları yaklaşık olarak birbirine yakındır. Örneğin YOLOv5 algoritmasında optimize edilecek hiper-parametre sayısı 29 iken YOLOv4 algoritmasında 28, YOLOR algoritmasında 28 ve Scaled-YOLOv4 algoritmasında 22'dir. Herhangi bir YOLO algoritmasının tüm hiper-parametreleri, ABC optimizasyon

algoritması ile optimize edilebilir. Nesne algılama algoritmalarında ve YOLO algoritmalarında, MSCOCO veri seti için genellikle iki farklı hiper-parametre dosyası kullanılır. Örneğin, YOLO algoritmalarında, MSCOCO eğitiminde sıfırdan (scracth) hiper-parametre dosyası kullanılırken, MSCOCO ince ayarı için optimize edilmiş ince ayar dosyası kullanılır. Bu çalışmada önce SUN ve PICCOLO açık veri setlerini sıfırdan hiper-parametre dosyası ile eğittik ve ardından ince ayar hiper-parametre dosyası ile aynı adımları yaptık. Bu veri setlerini seçmemizin nedeni ise çalışmanın daha nesnel olması için halka açık ve büyük veri setleri seçildi. Son olarak Tablo 5.1'de belirtilen hiper-parametreleri ABC algoritması ile optimize ederek ABC hiper-parametre dosyasını elde ettik. Bu dosyaların hiper-parametre değerleri Tablo 5.2'de gösterilmiştir.

Tablo 5.2. Scaled-YOLOv4'ün hiper-parametre değerleri (SUN ve PICCOLO polip veri kümeleri için Scratch, Fine-tune ve ABC algoritması).

Hiper-parametreler	scratch	fine-tune	ABC (SUN)	ABC (PICCOLO)	Hiper-parametreler	scratch	fine-tune	ABC (SUN)	ABC (PICCOLO)
initial learning rate	0.01	0.01	0.095	0.09	image HSV-Hue	0.015	0.015	0.016	0.0137
momentum	0.937	0.937	0.930	0.923	image HSV-Saturation	0.7	0.7	0.721	0.62
optimizer weight decay	0.0005	0.0005	0.00054	0.00055	image HSV-Value	0.4	0.4	0.446	0.497
GIoU loss gain	0.05	0.05	0.052	0.0479	image rotation	0.0	0.0	0.15	0.125
classification loss gain	0.5	0.5	0.55	0.514	image translation	0.5	0.5	0.35	0.197
classification BCELoss	1.0	1.0	1.12	0.9	image scale	0.5	0.8	0.62	0.48
objectness loss gain	1.0	1.0	0.9172	1.1	image shear	0.0	0.0	0.15	0.2
objectness BCELoss	1.0	1.0	0.9	1.07	image perspective	0.0	0.0	0.0	0.0
IoU training threshold	0.2	0.20	0.2	0.2	image flip up-down	0.0	0.0	0.35	0.42

anchor-multiple threshold	4.0	4.0	3.5	2.87	image flip left-right	0.5	0.5	0.4	0.48
focal loss gamma	0.0	0.0	0.0	0.0	image mix-up	0.0	0.2	0.31	0.11

Tablo 5.2'de görüldüğü gibi ABC optimizasyon algoritması alt ve üst değerler arasındaki hiper-parametreleri optimize etmeye çalışarak polip tespiti doğruluğunu arttırmaktadır. Çalışmamızda ABC algoritmasının YOLO algoritmasını daha iyi optimize etmek için kullandığı bazı parametre değerleri aşağıdaki gibi seçilmiştir. Popülasyon büyüklüğü 50, limit değeri 60 ve toplam değerlendirme sayısı 5000 olarak alınmıştır. Algoritma 3'de özetlendiği üzere, öncelikle Scaled-YOLOv4 algoritmasının en küçük modeli olan YOLOv4s SUN veri seti yeterli sayıda epok değeri ve varsayılan hiper-parametre değerleri ile eğitilmiştir. Buradan elde edilen en iyi model ABC algoritması ile üst ve alt hiper-parametre değerlerine bağlı olarak küçük bir epok değeri ile ince ayar yapılmıştır. Sonuç olarak, ABC algoritması, elde ettiği en iyi gıda kaynağına karşılık gelen en iyi model ve eğitilmiş hiper-parametrelerle YOLOv4'lerin ağırlığına ince ayar yaparak performansını iyileştirdi. Bu çalışmada uygunluk fonksiyonu olarak mAP metriğini seçtik. Bu nedenle, ABC algoritması mAP metriğini optimize etmeye çalışır. Uygunluk fonksiyonu olarak herhangi bir metrik seçilebilir, mAP seçilmesinin nedeni, nesne algılama algoritmalarında ve polip algılama performansı diğer metriklere göre daha net ortaya koymasıdır. Tablo 5.4 ve Tablo 5.5'te görüldüğü gibi polip tespitinde yaygın olarak kullanılan kesinlik (P), duyarlılık (R) ve F1 gibi metrikler eklenmiştir. Sctracth, fine-tune (ince ayar) ve ABC hiper-parametreleri ile eğitilen Scaled-YOLOv4 algoritmasının sonuçları Tablo 5.3 ve Tablo 5.4'te verilmiştir.

Tablo 5.3. SUN polip veri seti için sonuçlar (IOU: 0,65, küme boyutu 1, güven eşiği: 0,001, cihaz: 1xRTX 8000).

Scaled-YOLOv4			Scratch için sonuçlar				Fine-tune için sonuçlar				ABC için sonuçlar			
Model	Görüntü boyutu	FPS	P	R	F1	mAP	P	R	F1	mAP	P	R	F1	mAP
YOLOv4s	416	100	76.32	76.11	76.21	81.13	79.14	80.22	79.86	83.08	76.14	81.4	78.68	84.14

YOLOv4m	416	87	83.28	74.34	78.56	82.12	74.86	81.65	78.11	84.27	83.43	83.68	83.55	88.82
YOLOv4-CSP	416	71	80.90	78.61	79.74	84.92	81.30	85.11	83.16	87.43	86.23	84.06	85.13	89.39
YOLOv4-P5	416	53	79.18	80.96	80.06	85.24	82.46	84.14	83.29	87.78	83.76	87.51	85.28	90.79
YOLOv4-P6	448	42	78.68	83.96	81.24	87.63	82.77	85.94	84.32	88.18	83.36	87.71	85.47	90.98
YOLOv4-P7	512	35	80.36	84.92	82.57	87.89	83.49	85.87	84.66	88.75	84.42	89.33	86.80	92.93

Tablo 5.4. PICCOLO polip veri seti için sonuçlar (IOU: 0,65, küme boyutu 1, güven eşiği: 0,001, cihaz: 1xRTX 8000).

Scaled-YOLOv4			Scratch için sonuçlar				Fine-tune için sonuçlar				ABC için sonuçlar			
Model	Görüntü boyutu	FPS	P	R	F1	mAP	P	R	F1	mAP	P	R	F1	mAP
YOLOv4s	416	100	60.75	76.67	65.33	71.41	64.01	72.47	67.98	72.81	63.74	76.28	69.45	75.14
YOLOv4m	416	87	61.89	72.96	66.97	73.98	70.76	72.47	71.60	74.54	68.79	75.26	71.88	75.68
YOLOv4-CSP	416	71	84.42	67.35	74.92	74.33	77.74	72.70	75.14	74.96	81.35	78.57	79.93	78.11
YOLOv4-P5	416	53	76.75	72.96	74.81	74.65	74.52	79.08	76.73	76.69	81.60	78.79	80.17	79.67
YOLOv4-P6	448	42	80.24	74.59	77.31	78.81	75.15	78.06	76.58	78.98	82.14	80.65	81.39	82.27
YOLOv4-P7	512	35	78.85	77.55	78.20	79.56	82.39	77.55	79.90	79.85	82.69	81.85	82.26	83.18

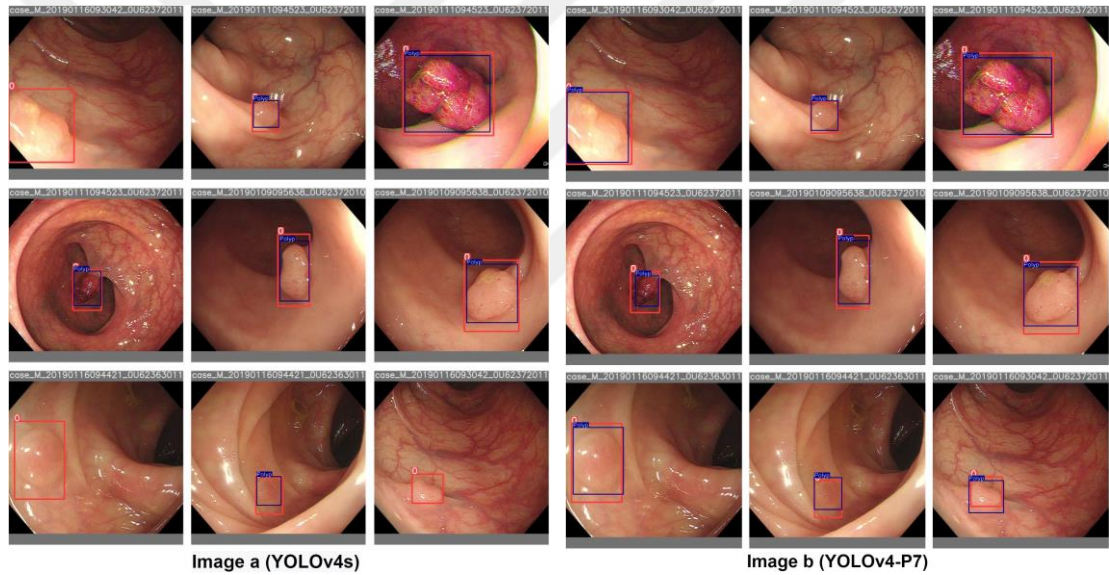
Tablo 5.3 ve Tablo 5.4, SUN ve PICCOLO polip veri kümeleri için 3 farklı hiperparametre içeren dosyalara göre Scaled-YOLOv4 algoritmasının performanslarını göstermektedir. Burada FPS değeri modellerin gerçek zamanlı çalışma hızını, görüntü boyutu ise eğitim ve testte kullanılan görüntülerin boyutlarını gösterir. Deneylerde 416x416 çözünürlüklü görüntü boyutunu seçmeye çalıştık ancak Scaled-YOLOv4-P6 ve Scaled-YOLOv4-P7 gibi daha büyük modeller için farklı çözünürlükler seçtik. Bunun ana nedeni, Scaled-YOLOv4-P6 modelinin maksimum adımı 64'ün katı olduğu için 416'ya en yakın değeri 448'e güncellememiz ve Scaled-YOLOv4-P7 modelini 512'ye güncellememizdir çünkü Scaled-YOLOv4-P7 modeli 128'in katıdır. P kesinliği, R hatırlamay, F1 ise F1 puanını ve mAP ortalama kesinliği temsil eder. Modeller

NVIDIA RTX 8000 grafik kartı ile eğitilmiş ve test edilmiştir ve Scaled-YOLOv4'ün varsayılan parametreleri kullanılmıştır. Buradaki temel amaç, hiper-parametrelerin performans üzerindeki etkisini incelemek ve ABC algoritmasının hiper-parametreleri optimize etmedeki başarısını ortaya koymaktır. Bu nedenle, daha kararlı bir deneysel çalışma için her model 10 kez eğitilmiştir. Daha sonra her modele ait metriklerin ortalaması alınarak değerlendirme yapılmıştır.

Tablo 5.3 ve Tablo 5.4 dikkate alındığında burada en dikkat çekici nokta ABC optimizasyon algoritmasının hiper-parametreleri ile eğitilen Scaled-YOLOv4 modellerinin diğer hiper-parametrelerle eğitilen modellere göre önemli ölçüde performans artışı göstermesidir. ABC hiper-parametreleriyle ince ayar yapılmış Scaled-YOLOv4 modellerinde aşağıdaki artışlar gözlemlendi. SUN veri seti için mAP değerinde %3'ün üzerinde bir artış olurken, PICCOLO veri seti için mAP değerinde %2'nin üzerinde bir artış olmuştur. Aynı şekilde her iki veri seti için F1 değerinde %2'den fazla artış sağlanmıştır. Yani, ABC optimizasyon algoritması, herhangi bir veri seti için hiper-parametreleri başarılı bir şekilde optimize ederek performansını artırır. Bir diğer önemli nokta ise modelin büyümesiyle (YOLOv4s->YOLOv4-P7) performansın önemli ölçüde artmasıdır. Bunun temel nedeni, derin öğrenme mimarileri yeterli bir veri seti ile eğitilirse, modelin katman sayısı ve filtre sayısı gibi mimarinin derinleşmesi ile modellerin performansının artmasıdır.

YOLOv4-P7 modeli polip tespiti için en iyi tespit doğruluğunu göstermiş ve bu modelin en iyi performansı ABC algoritması tarafından üretilen hiper-parametreler ile elde edilmiştir. Scaled-YOLOv4 algoritmasının en küçük modeli olan YOLOv4s, diğer modellere göre en düşük tespit doğruluğuna sahiptir. Tüm bunlara rağmen gerçek zamanlı performans açısından yani FPS değerlerine göre YOLOv4s en iyi model olurken, YOLOv4-P7 en düşük FPS değerine sahip modeldir. Modellerin F1 ve mAP metrikleri incelendiğinde ABC'nin hiper-parametreleri ile eğitilen modellerin bu metriklerdeki performansı sürekli olarak artırdığı ancak aynı zamanda daha kararlı ve düzenli bir gelişim gösterdiği görülmektedir. Hem tespit doğruluğu hem de gerçek zamanlı performansa sahip modeller, bir sonraki aşama olan polip tespit sisteminin klinik uygulamasına geçebilir. Bir modelin gerçek zamanlı sayılabilmesi için FPS değerinin en az 30 olması gerekir. Model bir endoskopi cihazına entegre edilecekse FPS değerinin 30'dan bile yüksek olması gerekir, aksi takdirde gerçek uygulamalarda

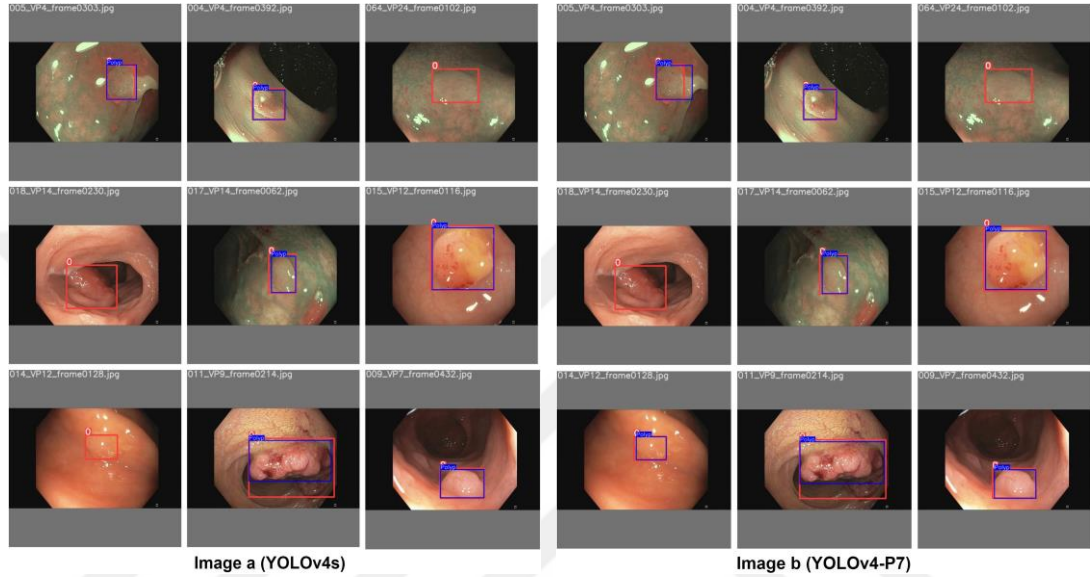
gecikmeler olacaktır, modelin uygulanabilirliğini sınırlamaktadır. Model TensorRT ile geliştirilirse ve devreye alınırsa FPS oranı artar ve daha büyük modeller daha etkili bir şekilde kullanılabilir. Gerçek zamanlı olmayan durumlar için en iyi model, en büyük model olan YOLOv4-P7'dir. Öte yandan, YOLOv4s, YOLOv4m, YOLOv4-CSP gerçek zamanlı uygulama için yeterli görünmektedir. Scaled-YOLOv4 modelleri TensorRT aşamasına geçilirse, YOLOv4-P7 gerçek zamanlı uygulamalar için daha kullanılabilir hale gelebilir. Ayrıca, YOLOv4-P7 ve YOLOv4-P6 gibi daha büyük modeller için bulut GPU'lar veya üst düzey GPU'lar kullanılarak bu sorun ortadan kaldırılabilir. Şekil 5.4 SUN polip veri setinde ABC algoritması ile optimize edilmiş Scaled-YOLOv4'ün (YOLOv4s ve YOLOv4-P7) performansını göstermektedir.



Şekil 5.4. Scaled-YOLOv4'ün modelinin bazı SUN test verileri üzerindeki algılama performansı (Kırmızı sınırlayıcı kutular temel gerçeği gösterir. Image a (solda) YOLOv4s modelinin sonuçlarını gösterirken, Image b ise (sağda) YOLOv4-P7 modeline ait sonuçları göstermektedir).

Şekil 5.4. SUN veri setinin test görüntülerinde Scaled-YOLOv4 algoritmasının (YOLOv4s ve YOLOv4-P7) algılama performansının bir örneğini göstermektedir. Tablo 5.4'te görüldüğü gibi, YOLOv4s modelinin algılama performansı, YOLOv4-P7 modeline göre daha düşüktür. ABC algoritmasının ise modellerin algılama performansını önemli ölçüde arttırdığı görülmektedir. YOLOv4s modeli SUN veri setindeki test verilerinden rastgele seçilen 3 görüntüyü algılayamazken kısacası 3 FN değeri oluşurken YOLOv4-P7 modelinde FN değeri yoktur yani rastgele seçilen tüm

görüntüleri algılamaktadır. Sonuç olarak Scaled-YOLOv4 nesne algılama algoritmasında model büyüdükçe polip algılama doğruluğunda önemli bir artış gösterdiği açıktır. Şekil 5.5. optimize edilmiş Scaled-YOLOv4'ün (YOLOv4s ve YOLOv4-P7) ABC algoritması ile PICCOLO polip veri seti üzerindeki performansını göstermektedir.



Şekil 5.5. Scaled-YOLOv4'ün modelinin bazı PICCOLO test verileri üzerindeki algılama performansı (Kırmızı sınırlayıcı kutular temel gerçeği gösterir. Image a (solda) YOLOv4s modelinin sonuçlarını gösterirken, Image b ise (sağda) YOLOv4-P7 modeline ait sonuçları göstermektedir).

Şekil 5.5, PICCOLO veri setinin test görüntülerinde Scaled-YOLOv4 algoritmasının (YOLOv4s ve YOLOv4-P7) algılama performansının bir örneğini göstermektedir. Tablo 5.3'te görüldüğü gibi YOLOv4s modelinin algılama performansı YOLOv4-P7 modeline göre daha düşüktür. Tablo 5.3 ve Tablo 5.4 incelendiğinde ABC algoritmasının modellerin tespit performansını önemli ölçüde arttırdığı görülmektedir. Sonuç olarak Scaled-YOLOv4 nesne tespit algoritmasında model büyüdükçe polip tespit doğruluğunda önemli bir artış olduğu görülmektedir.

5.8. Önerilen Modellerin En Son Teknoloji Yöntemlerle Karşılaştırılması

Bu bölümde eğitilmiş tüm modellerin performansı, diğer son teknoloji modellerle karşılaştırmak için Etis-Larib veri seti üzerinde değerlendirilmiştir. Önerilen YOLOv3

modeli özel veri seti, SUN ve PICCOLO veri kümeleri için en iyi performansı sağladığından, bu model halka açık Etis-Larib veri seti üzerinde değerlendirilmiştir. Bunlara ek olarak farklı veri kombinasyonları ile eğitimler gerçekleştirilmiştir. Buradaki amaç, eğitimde veri artışının polip tespiti üzerindeki etkisini incelemek ve önerilen yöntemin başarısını değerlendirmektir. Farklı veri kombinasyonlara ve bunlara ait sonuçlar aşağıdaki tabloda gösterilmiştir.

Tablo 5.5 Önerilen yöntemlerin son teknoloji yöntemlerle karşılaştırılması (IOU: 0.5, küme boyutu 1, güven eşiği: 0.25, ekran kartı: 1xRTX 2080TI).

Model	Eğitim veri seti	Test veri seti	Kesinlik %	Duyarlılık %	F1 %	FPS	Gerçek Zamanlı
Önerilen YOLOv3	Özel veri seti +SUN+PICCOLO+CVC-ClinicDB	Etis-Larib	98.95	95.93	97.41	100	Evet
Önerilen YOLOv3	Özel veri seti	Etis-Larib	93.20	92.56	92.87	100	Evet
Önerilen YOLOv3	SUN+PICCOLO+CVC-ClinicDB	Etis-Larib	90.61	91.04	90.82	100	Evet
Önerilen YOLOv3	SUN+ CVC-ClinicDB	Etis-Larib	90.14	90.57	90.35	100	Evet
Önerilen YOLOv3	SUN	Etis-Larib	88.24	88.24	88.24	100	Evet
Önerilen YOLOv3	PICCOLO	Etis-Larib	87.29	75.53	80.99	100	Evet
Wittenberg ve ark. 2019 [103]	Özel	Etis-Larib	83.00	74.00	79.00	N/A	Hayır
Liu ve ark. 2021 [126]	CVC-ClinicDB	Etis-Larib	77.80	87.50	82.40	10	Hayır
Wang ve ark. 2019 [97]	GIANA 2017	Etis-Larib	88.89	80.77	84.63	53	Evet

Qadir ve ark. 2021 [16]	CVC-ClinicDB	Etis-Larib	86.54	86.12	86.33	26	Hayır
Pacal ve ark. 2021 [14]	CVC-ClinicDB	Etis-Larib	91.62	82.55	86.85	122	Evet
Shen ve ark. 2021 [127]	CVC-ClinicDB	Etis-Larib	91.49	82.69	86.87	N/A	Hayır

Tablo 5.5'e bakıldığında ilk üç satırdaki metrik değerlerden de açıkça görüleceği üzere önerilen modelin performansı veri sayısı arttıkça artmaktadır. Bu durumda Özel veri seti + SUN + CVC-ClinicDB + PICCOLO veri setleri (en fazla veriyi içeren) üzerinde eğitilen model, diğer tüm modellerden daha başarılı görünmektedir. Sadece FPS yani hız düşünüldüğünde Pacal ve Karaboğa tarafından önerilen metot, önerilen YOLOv3 modelinden hız olarak biraz daha ilerdedir. Önerilen YOLOv3 modeli yani Model 4 NVIDIA TensorRT ile kullanıldığında FPS oranı 200'den fazla olmaktadır. Önerilen özel veri seti SUN ve PICCOLO gibi büyük veri setlerinden daha yüksek performans göstermiştir. Bunun temel nedeni ise önerilen veri setindeki eşsiz polip sayısının daha fazla olmasıyla beraber polipli görüntü sayısının oldukça fazla olması ve görüntü çözünürlüğünün de yüksek olmasından kaynaklıdır. PICCOLO veri seti ile eğitilen model, başarı açısından diğer modellerin biraz gerisindedir. Önerilen özel veri seti ile beraber diğer veri setlerinden olan SUN + CVC-ClinicDB + PICCOLO veri seti üzerinde eğitilen modeller eğer ince ayar yapılırsa Etis-Larib veri setinde daha başarılı sonuçlar verecektir. Özetlemek gerekirse veri sayısı özellikle eşsiz polip sayısı ve polipli görüntü sayısı arttıkça modellerin başarımı artmaktadır ve özel veri seti ile SUN veri seti gibi büyük veri setleri ile eğitilen model çok yüksek bir genelleme yeteneğine sahiptir.

5.9. ABC Algoritması ile Aktivasyon Fonksiyonlarının Optimizasyonu

Bu bölümde derin öğrenme mimarilerinin başarımına ciddi bir etkisi olan aktivasyon fonksiyonlarının ABC optimizasyon algoritması ile nasıl optimize edileceğinden bahsedilmiştir. Aktivasyon fonksiyonları, sinir ağlarının eğitiminde önemli bir rol oynamaktadır. Günümüzde birçok aktivasyon fonksiyonu CNN mimarilerinde yerini almasına rağmen bazı aktivasyon fonksiyonlarının doğal davranışı diğerlerinden daha

başarılı olabilmektedir. Aktivasyon fonksiyonlarının sınıflandırma, tahmin veya tespit başarıları alandan alana değişiklik gösterebilir. Otomatik polip tespitinde aktivasyon fonksiyonlarının performansını ölçmek için temel aktivasyon fonksiyonları ile güncel ve popüler aktivasyon fonksiyonlarının birlikte kullanıldığı kapsamlı bir çalışma gerçekleştirdik. Bu çalışmada kullanılan bazı önemli aktivasyon fonksiyonlarının genel ve kısa bir tanımını 4.bölümde aktivasyon fonksiyonları adlı başlıkta bakılabilir. Çalışmada daha önce anlatılan aktivasyon fonksiyonlarına benzer şekilde ELU, GELU, SELU, Tanh vb. bazı aktivasyon fonksiyonları da kullanılmıştır.

Çalışmada öncelikle var olan aktivasyon fonksiyonlarına ait başarılar ölçülmüştür. Bunun için YOLOv5s algoritması kullanılmıştır çünkü YOLOv5s algoritması YOLOv5'in diğer modellerine göre daha hızlı bir eğitim sunmaktadır. YOLOv5s modeli 5'er defa eğitildikten sonra 5'er koşmanın ortalaması alınmıştır. Bunu yapmamızın nedeni ise daha hassas bir değerlendirme gerçekleştirmektir çünkü derin öğrenme algoritmalarındaki deterministik olmayan özelliklerinden dolayı yeniden aynı sonuçları almak zordur. Ayrıca, aynı çekirdekler kullanılsa bile, CPU ve GPU yürütmeleri arasında sonuçlar yeniden üretilmeyebilir. Bu nedenle en az 5'er defa modeller koşturularak ortalamaları alınmıştır. Optimize edilmeden önce her bir aktivasyon fonksiyonunun SUN ve PICCOLO veri setleri üzerindeki başarımı Tablo 5.7'de gösterilmişti. Tablolarda her bir aktivasyon fonksiyonlarına ait GPU kullanım miktarı, kesinlik, duyarlılık, mAP@.5 ve mAP@.5:.95 gibi önemli metrikleri vererek detaylı bir karşılaştırma sunduk. Böylece, aktivasyon fonksiyonlarının polip tespiti için duyarlılığı ve veri setlerine karşı etkileri ölçülmüş olmaktadır.

Tablo 5.6. SUN polip veri seti için sonuçlar (IOU: 0.6, küme boyutu 128, güven eşiği: 0,001, cihaz: 1xRTX 8000).

Model	Aktivasyon fonksiyonu	GPU kullanımı	Kesinlik %	Duyarlılık %	mAP (mAP@.5) %	mAP@.5:.95 %
YOLOv5s	ELU	11.40	90.63	70.56	82.7	47.93
YOLOv5s	GELU	11.40	88.92	68.62	80.37	45.5
YOLOv5s	SELU	11.40	89.95	72.27	83.28	48.15
YOLOv5s	SiLU	11.40	90.98	66.38	77.98	45.60

YOLOv5s	Swish	11.40	89.80	67.42	78.72	45.38
YOLOv5s	Mish	11.40	88.70	66.93	78.41	44.98
YOLOv5s	Identity	8.79	48.37	36.10	34.94	0.11
YOLOv5s	ReLU	9.45	86.92	65.95	75.65	42.7
YOLOv5s	Leaky ReLU	9.45	87.60	70.05	79.52	44.04
YOLOv5s	Sigmoid	9.45	88.80	70.23	80.5	45.71
YOLOv5s	Tanh	9.44	86.32	69.17	78.80	43.87

Tablo 5.6 düşünüldüğünde, Identity yani lineer aktivasyon fonksiyonu hariç diğer aktivasyon fonksiyonlarının başarımlarının birbirine çok da uzak olmadığı görülmektedir. Polip tespitinde ve nesne tespit algoritmalarında mAP metrikleri temel metrik olarak düşünülmektedir. Bu yüzden modellerimiz en iyi ağırlığı bulmak için mAP metriğini kullandı. Yani en iyi ağırlık karşılaştırılması için 0,9 oranında mAP@.5:.95 ağırlığı kullanılırken 0,1 oranında ise mAP@.5 metriğinin ağırlığı temel alındı. Tablo 5.7'ye bakıldığında en başarılı aktivasyon fonksiyonu SUN veri seti için SELU olduğu görülmektedir. Daha sonra ise ELU ve SiLU aktivasyon fonksiyonları gelmektedir. Yine aynı tabloya bakıldığında en az GPU kullanımı lineer aktivasyon fonksiyonunda iken en yüksek GPU kullanımı ise SiLU, Mish gibi aktivasyon fonksiyonları kullanmaktadır. Sigmoid aktivasyon fonksiyonu ise GPU kullanımı ve başarımları daha optimal olduğu söylenebilir.

Tablo 5.7. PICCOLO polip veri seti için sonuçlar (IOU: 0.6, küme boyutu 128, güven eşiği: 0,001, cihaz: 1xRTX 8000).

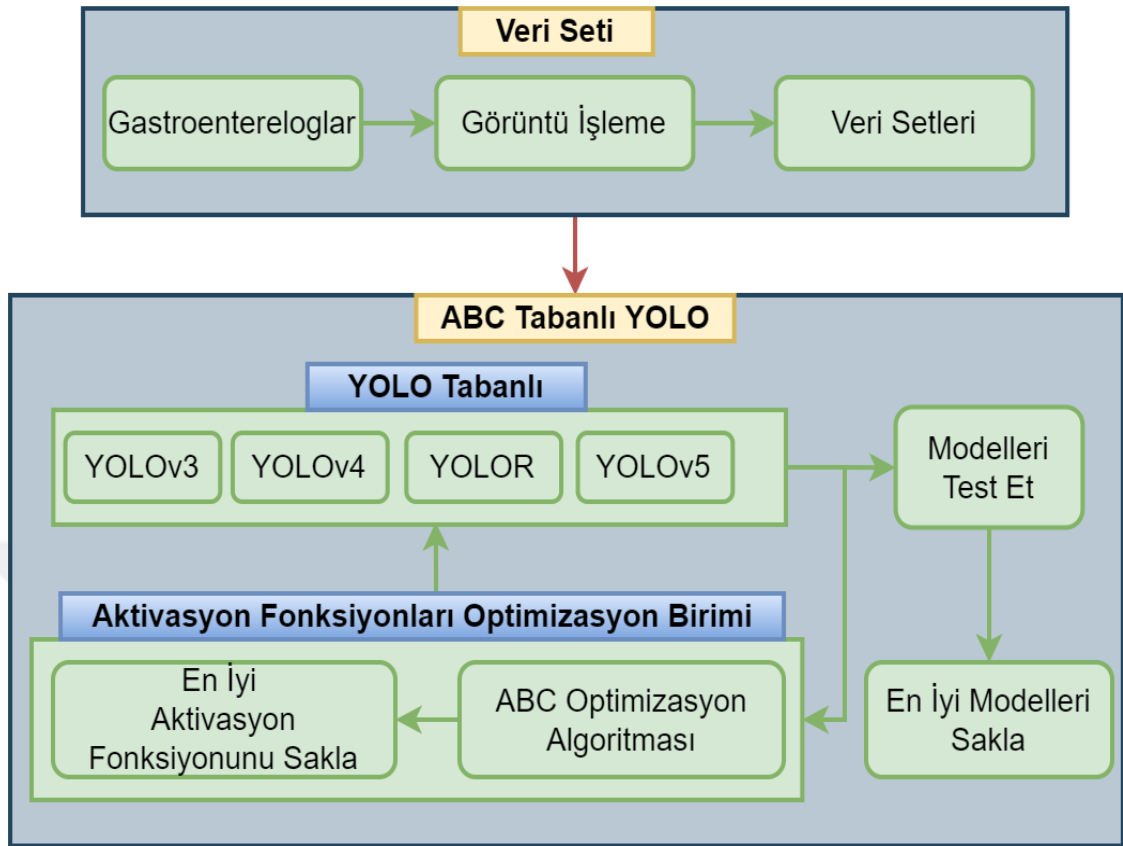
Model	Aktivasyon fonksiyonu	GPU kullanımı	Kesinlik %	Duyarlılık %	mAP (mAP@.5) %	mAP@.5:.95 %
YOLOv5s	ELU	11.40	88.97	64.82	74.77	54.74
YOLOv5s	GELU	11.40	86.01	66.97	75.37	55.04
YOLOv5s	SELU	11.40	85.10	63.30	72.55	53.73

YOLOv5s	SiLU	11.40	89.42	62.75	73.65	56.33
YOLOv5s	Swish	11.40	88.13	67.78	77.30	58.18
YOLOv5s	Mish	11.40	86.98	65.18	75.37	57.95
YOLOv5s	Identity	8.79	30.50	25.00	14.63	03.94
YOLOv5s	ReLU	9.45	90.57	64.45	75.10	54.65
YOLOv5s	Leaky ReLU	9.45	88.43	64.60	75.08	56.25
YOLOv5s	Sigmoid	9.45	89.27	67.36	75.86	55.90
YOLOv5s	Tanh	9.44	87.44	67.98	76.98	54.98

Tablo 5.7 düşünüldüğünde, yine SUN veri setinde olduğu gibi Identity yani lineer aktivasyon fonksiyonu hariç diğer aktivasyon fonksiyonlarının başarımlarının birbirine benzer olduğu görülmektedir. Swish aktivasyon fonksiyonun diğer fonksiyonlara nazaran daha başarılı olduğu görülürken Mish, SiLU ve Leaky ReLU aktivasyon fonksiyonlarının da benzer performansa sahip olduğu görülmektedir. Leaky ReLU aktivasyonu daha az GPU kullanımı ile başarımlarından optimal bir performans gösterdiği görülmektedir.

5.9.1. Aktivasyon Fonksiyonlarının Optimizasyonu için Yöntem

Hiper-parametre optimizasyonu için kullanılan ABC optimizasyon algoritması aktivasyon fonksiyonu için yine aynı şekilde kullanıldı. Burada bütün fonksiyonlarına integer değerler atandı. Örneğin 20 aktivasyon fonksiyonlu bir çalışmada 1-20 arasında her bir aktivasyon fonksiyonuna kimliğine değerler atandı. Böylece hiper-parametre optimizasyonu bölümünde olduğu gibi aktivasyon fonksiyonlarında bir hiper-parametre olarak ele alındı. Böylece eğitilen model üzerinde ince ayar yapılarak en ideal aktivasyon fonksiyonu bulunmuş olmaktadır. Şekil 5.6 ABC optimizasyon algoritmasının aktivasyon fonksiyonlarının optimizasyonu için nasıl kullanıldığını göstermektedir.



Şekil 5.6. Aktivasyon fonksiyonlarının optimizasyonu için önerilen ABC tabanlı YOLO algoritmasının yapısı.

Algoritma 3'te gösterildiği gibi, önerilen yöntemin ilk adımları veri setini sisteme yüklemektir. Yüklenen veri seti daha sonra eğitim ve test verisi olarak ayrılır. Buraya kadar olan kısım veri seti ile ilgili iken sonraki kısım YOLO algoritmasında yapılan değişikliklerden oluşmaktadır. Önceden eğitilmiş ağırlıklar (YOLOv5s MSCOCO), uygunluk işlevi olarak mAP güncellemesinden sonra yüklenir. Aynı zamanda, varsayılan hiper-parametreler başlatılır. Ardından YOLOv5s modeli eğitime başlanır. Burada YOLOv5s modelini seçmemizin nedeni, aktivasyon fonksiyonu ve hiper-parametre optimizasyonunun zaman alan bir süreç olması ve küçük ölçekli modellerin eğitiminin diğer büyük modellere göre zaman kazandırmasıdır. Modeller her bir aktivasyon fonksiyonu için eğitildikten sonra, ABC optimizasyonu için en iyi ağırlıklar tutulur. Bir sonraki kısım, ABC algoritması ve YOLO algoritmasının birleşimine dayanmaktadır. İlk olarak, ABC algoritmasının parametreleri başlatılır (popülasyon büyüklüğü, limit değeri, deneme değeri ve toplam değerlendirme sayısı gibi). ABC algoritmasının YOLOv5 algoritmasının için aktivasyon fonksiyonlarını hiper-parametre

olarak ele alır ve bu parametreler için alt ve üst değerler tanımlanır (örneğin 20 aktivasyon fonksiyonu ise bu aralık 1-20 arasında tutulur). Ardından YOLOv5s modelinin en iyi ağırlığı yüklenir. ABC algoritması, belirtilen dönem sayısı ile YOLOv5 algoritmasının en iyi modeline ince ayar yapmaya çalışır. Bu işlem ABC algoritmasının çalışma prensibine göre yapılır. Böylece, elimizde var olan aktivasyon fonksiyonları (bu çalışmada 11 aktivasyon fonksiyonu kullanılmıştır) üzerine ABC ince ayar yapacaktır. Bu işlem sadece aktivasyon fonksiyonları ile de yapılabilir ya da aktivasyon fonksiyonları ile beraber hiper-parametrelerde eş zamanlı çalışabilir. Bu işlem zaman aldığı için kullanıcının donanım ve zaman gereksinimleri göz önünde bulundurularak esnek tutulmuştur.

Burada aktivasyon fonksiyonlarının optimize edilmesindeki temel amaç her bir aktivasyon fonksiyonu için yapılacak eğitimlerin ABC ile tek seferde bütün aktivasyon fonksiyonlarını optimize etmesinden onlarca kat daha fazla zaman almasıdır. ABC bunu herhangi bir aktivasyon fonksiyonu ile eğitilen bir modeli ince ayar şeklinde optimize ederek en ideal aktivasyon fonksiyonunu bularak zaman tasarrufu sağlamakla beraber hiper-parametre optimizasyonunu da aktivasyon fonksiyonlarının doğasına uygun değerler sunarak başarımda ciddi bir katkı sağlar.

5.9.2. Aktivasyon Fonksiyonlarının Optimizasyonu için Sonuçlar

Bu bölümde ABC algoritması ile optimize edilen YOLOv5s algoritmasının en ideal aktivasyon fonksiyonu ve hiper-parametre değerleri sunulmuştur. Bu çalışmada hem hiper-parametre optimizasyonu hem de aktivasyon fonksiyonu optimizasyonu eş zamanlı yapılarak her bir aktivasyon fonksiyonunu doğasına uygun en ideal hiper-parametreler sunulmuştur. 11 tane aktivasyon fonksiyonu için eğitilen YOLOv5s modeline ait modellerin bir havuzda biriktirildikten sonra optimizasyon için bunlardan rastgele şekilde bir tane aktivasyon fonksiyonu seçildikten sonra optimizasyon süreci ince ayar şeklinde 20 epoktan az olmamak şartıyla diğer bütün değerler sabit tutularak yapılmıştır. Yani kısaca YOLOv5s modelinin her şeyi sabit tutularak sadece ABC ile ince ayar yapılarak en ideal değerler bulunmaya çalışılmıştır. Tablo 5.9 YOLOv5 algoritmasına ait hiper-parametre ve aktivasyon fonksiyonlarının optimal yani ABC algoritması ile optimize edilmiş değerlerini göstermektedir.

Tablo 5.8. YOLOv5s modelinin aktivasyon fonksiyonu optimizasyonu ve bazı hiper-parametre değerleri (SUN ve PICCOLO polip veri kümeleri için Scratch ve ABC algoritması).

Hiper-parametreler	scratch	ABC (SUN)	ABC (PICCOLO)	Hiper-parametreler	scratch	ABC (SUN)	ABC (PICCOLO)
Aktivasyon Fonksiyonu	SiLU	SELU	Swish	Aktivasyon Fonksiyonu	SiLU	SELU	Swish
initial learning rate	0.01	0.0091	0.0078	image HSV-Hue	0.015	0.0155	0.01433
momentum	0.937	0.8830	0.79056	image HSV-Saturation	0.7	0.7321	0.77078
optimizer weight decay	0.0005	0.00064	0.00066	image HSV-Value	0.4	0.346	0.36839
classification loss gain	0.5	0.52	0.517	image translation	0.1	0.095	0.09975
classification BCELoss	1.0	1.154	1.1581	image scale	0.5	0.424	0.4191
objectness loss gain	1.0	0.8172	0.7844	image shear	0.0	0.147	0.02
objectness BCELoss	1.0	0.754	0.66742	image perspective	0.0	0.0	0.0
IoU training threshold	0.2	0.2	0.2	image flip left-right	0.0	0.55	0.5
anchor-multiple threshold	4.0	4.14	4.9415	mosaic	1.0	0.91	0.96226

Tablo 5.8 düşünüldüğünde, YOLOv5s modeline ait aktivasyon fonksiyonlarının ve hiper-parametrelerin çoğunun değiştiği görülmektedir. ABC algoritmasının aktivasyon

fonksiyonlarını ve hiper-parametreleri başarılı bir şekilde optimize ettiği görülmektedir. Şimdi bu aktivasyon fonksiyonlarına ve hiper-parametrelere ait başarımların metriklerine bakalım. Tablo 5.10 ABC optimizasyonu sonucunda SUN ve PICCOLO veri setlerine ait değerlendirme metriklerini sunmaktadır. Bu metrikler doğrultusunda ABC optimizasyon algoritmasının YOLO-tabanlı nesne tespit algoritmalarını optimize etmedeki başarısı değerlendirilebilir.

Tablo 5.9. SUN ve PICCOLO polip veri setleri için sonuçlar (IOU: 0,65, küme boyutu 1, güven eşiği: 0,001, cihaz: 1xRTX 8000).

Model	Veri Seti	Görüntü boyutu	Aktivasyon fonksiyonu	Keskinlik %	Duyarlılık %	mAP (mAP@.5) %	mAP@.5:.95 %
YOLOv5s	SUN	416	SiLU	90.98	66.38	77.98	45.60
YOLOv5s + ABC	SUN	416	SELU	90.35	73.21	83.77	48.88
YOLOv5s	PICCOLO	416	SiLU	89.42	62.75	73.65	56.33
YOLOv5s +ABC	PICCOLO	416	Swish	89.15	65.06	78.29	58.98

Tablo 5.9 düşünüldüğünde, ABC algoritmasının aktivasyon fonksiyonları ve hiper-parametre optimizasyonu sonucunda başarımları gösteren metriklerde ciddi bir artış kaydedilmiştir. Hem SUN hem de PICCOLO veri setleri YOLOv5s’de yer alan SiLU aktivasyon fonksiyonu yerine SELU ve Swish gibi farklı aktivasyon fonksiyonları ile daha yüksek bir başarımlar elde etmiştir. Demek ki nesne tespit algoritmalarında daha yüksek performans gösteren SiLU aktivasyon fonksiyonu polip tespiti için diğer aktivasyon fonksiyonlarına nazaran biraz geride kaldığı görülmüştür. Bunun temel nedeni ise derin öğrenme mimari ve algoritmalarının alana göre başarımlarının farklı olmasından kaynaklıdır. ABC algoritması manuel olarak elde edilen aktivasyon fonksiyonları ile SUN ve PICCOLO veri setlerinde en iyi başarımları gösteren aktivasyon fonksiyonunu otomatik bularak daha kararlı bir mekanizma sunmaktadır. Örneğin SUN veri setinde manuel olarak koşturulan aktivasyon fonksiyonlarından en başarılı

performansı SELU gösterirken ABC ile de aynı aktivasyon fonksiyonunun daha başarılı olduğunu Tablo 5.6 ve 5.9’da gösterilmektedir. Bu da ABC algoritmasının başarılı ve daha etki bir şekilde aktivasyon fonksiyonlarını ve hiper-parametreleri optimize ettiği görülmektedir. Ayrıca aktivasyon fonksiyonları ile beraber hiper-parametreleri de optimize ederek başarımda ciddi bir artışa sebep olmuştur. Aynı durum PCCOLO veri seti içinde geçerli olduğu görülmektedir. Kısaca, ABC algoritması hiper-parametre ve aktivasyon fonksiyonlarının istenilen alana göre daha esnek ve zamanla beraber donanımdan tasarruf ederek daha efektif bir şekilde optimize edebilmektedir. Bu da YOLO-tabanlı nesne tespit algoritmalarına ABC algoritmasının ciddi bir katkı sağladığını göstermektedir.

6. BÖLÜM

TARTIŞMA-SONUÇ ve ÖNERİLER

6.1. Tartışma ve Sonuçlar

Bu tez çalışmasında, kolorektal kanserlerinin öncüsü olan polip yapılarının gerçek zamanlı otomatik tespiti için derin öğrenme tabanlı mimariler ele alınmıştır. Ele alınan mimariler incelenmiş ve gerçek zamanlı otomatik polip tespiti için en ideal yapılar tespit edilmiş ve yeni mimariler sunulmuştur. Gerçek zamanlı otomatik polip tespitindeki başarımın artırılması için ve klinik bir seviyede doğrulanması için bazı yeni metotlar ve yapılar sunulmuştur.

Derin öğrenme tabanlı bilgisayar destekli otomatik polip tespiti sistemi üç temel yapıdan oluşmaktadır. Bu yapıların en başında etkin bir veri setinin olması gelmektedir. Çünkü derin öğrenme mimarilerin yapısında veriye aç bir nitelik söz konusudur, yoksa bu tür problemler klasik makine öğrenme yöntemleri ile de sürdürülebilirdi. Etkili bir veri setinden kasıt ise veri setinin yeteri kadar eşsiz hasta ile beraber eşsiz polip içermesi, polipli görüntü sayısının en az on binler seviyesinde olması ve aynı zamanda negatif örneklerinde içermesi denilebilir. 2021 yılına kadar halka açık veri setlerinde maalesef böyle bir veri seti niteliği bulunmamaktaydı. En yüksek veri içeren CVC-ClinicDB veri seti ise toplamda 612 görüntü ve 31 eşsiz polip görüntüne sahipti. Böyle bir veri seti ile derin öğrenme mimarileri eğitilebilir fakat klinik seviyeye taşımak oldukça zor hatta imkânsız denilebilir. Sonraki yıllarda büyük veri setleri halka açık olmalarına rağmen yine de istenilen seviyede oldukları söylenemez. Bu yüzden, bu tez çalışmasında, literatürde var olan bütün veri setlerinden hem eşsiz polip sayısı hem de görüntü olarak daha büyük veriye sahip özel bir veri seti sunulmuştur. Bu veri seti ile derin öğrenme mimarileri etkili bir şekilde eğitilebilir ve bu modeller klinik seviyeye taşınabilir bir konuma gelmektedir.

Otomatik polip tespiti sisteminin diğer önemli yapısını ise en ideal derin öğrenme mimarisinin seçimi oluşturmaktadır. Derin öğrenme tabanlı gerçek zamanlı otomatik polip tespiti çalışmalarının tamamına yakını nesne tespit algoritmaları temelli bir yaklaşımla sunulmuştur. Bu yaklaşımların çoğu ise tek aşamalı yaklaşımlardan oluşmaktadır. Bunun temel sebebi ise iki aşamalı yaklaşımlardan ziyade, sadece tek aşamalı yaklaşımların gerçek zamanlı tespit yapabilme kabiliyetlerinin olmasıdır. YOLO tabanlı algoritmalar otomatik polip tespiti için en ideal yaklaşımların başında gelmektedir. YOLO algoritmaları nesne tespiti için önerilmiş en gelişmiş nesne tespit algoritmalarının başında gelir fakat polip tespitine uyarlanması için bu tarz algoritmaların bu alana göre yeniden yapılandırılması gereklidir. Bunun nedeni ise bu algoritmaların doğasına göre sadece nesne tespiti için optimize edilmiş ve buna göre parametreler içermektedirler. Buna binaen, YOLO tabanlı dört model gerçek zamanlı otomatik polip tespiti sistemleri için önerilmiştir. Bu modellerden ilk ikisi sıg modeller olmakla beraber daha çok küçük veri setleri için önerilirken, diğer iki model ise ilk iki modele göre oldukça derin modeller olmakla beraber büyük veri setleri için daha uygundur. Bu dört model, CSPNet mimarisinin yardımıyla YOLOv3 ve YOLOv4 algoritmalarının otomatik polip tespiti için yeniden yapılandırılmasını içeriyor. Bu yapılandırma, filtre sayısı ve boyutu, Transformers gibi blokların, CNN özellik çıkarıcı yapılarının ve aktivasyon fonksiyonu gibi birçok yapının optimize edilmesiyle elde edilmiştir. Bu önerilen modellerin orijinal YOLO modellerine göre daha başarılı oldukları deneysel sonuçlarla da kanıtlanmıştır.

Aktivasyon fonksiyonları ve hiper-parametre optimizasyonu, derin öğrenme mimarilerinin başarımını ciddi oradan artıran yöntemlerden biridir. Otomatik polip tespiti sisteminin son önemli yapısını ise aktivasyon fonksiyonlarının ve hiper-parametrelerin optimizasyonu oluşturmaktadır. Nesne tespit algoritmalarının çoğu hiper-parametre optimizasyonunu manuel olarak yapmakla kalmakla beraber genelde tek bir probleme ait hiper-parametreler paylaşılır. Polip tespit sisteminin başarımı artırmak için en ideal hiper-parametrelerin olması gereklidir. Bu tez çalışmasında, aktivasyon fonksiyonlarının ve hiper-parametrelerin optimizasyonunu daha etkin hale getirmek için ABC algoritmasını kullanarak YOLO algoritmalarının hiper-parametrelerini optimize ettik. Aktivasyon fonksiyonlarının başarımı alandan alana farklılık gösterir ve bu yüzden zaman ve donanım açısından tasarrufla beraber tespit

doğruluğunun artırılması için aktivasyon fonksiyonları optimize edilmiştir. Hiper-parametreler temel hiper-parametreler ve veri seti bağlantılı hiper-parametreler olarak iki başlıkta düşünülebilir. Temel hiper-parametreler modelin öğrenme kabiliyetini değiştirebilen en etkili parametreler iken, veri seti ile ilgili hiper-parametreler ise eğitim esnasında veri artırma tekniklerinin hangi sıklıkla ve hangi oranda olacağını belirleyen parametrelerdir. Bu yüzden, her bir hiper-parametrenin başarıma etkisi vardır. Yapılan deneysel çalışmalarda, ABC algoritması, YOLO tabanlı algoritmaları (YOLOv3, YOLOv4, YOLOv5, YOLOR) başarılı bir şekilde optimize ettiği gözlemlendi. ABC algoritması ile YOLO algoritmalarındaki başarımlar en az %3'lük bir artış ile deneysel olarak kanıtlanmıştır. Sonuç olarak, ABC algoritması nesne tespit algoritmalarında hiper-parametreleri tek bir seferde başarılı bir şekilde optimize edebilme kabiliyetine sahiptir.

Gerçek zamanlı otomatik polip tespitinin başarımlarının artırılması için yapılan bazı önemli diğer teknikler ise şu şekildedir. YOLO tabanlı bütün modeller ve önerilen modeller üç farklı şekilde eğitilerek polip tespiti için en ideal eğitim modelleri deneysel sonuçlar yardımıyla tanımlanmıştır. İlk eğitim modeli, YOLO tabanlı modellerin uçtan uca yani sıfırdan eğitimlerini içermektedir. Bu tür eğitimlerde herhangi bir ağırlık başlangıcı sunulmadan sıfırdan modeller eğitilir. Bu tür eğitim, diğer eğitim tekniklerine kıyasla daha uzun bir zamanda tamamlanır ve yakınsama süresi de uzundur. İkinci eğitim modeli olan transfer eğitiminde ise eğitimin başlangıcı için önceden farklı veri setleri üzerinde eğitilmiş modellerin ağırlıkları ile model başlatılır yani ağırlıklar transfer edilir. Böylece, daha kısa sürede hem yakınsama gerçekleşir hem de kısa sürede model eğitimi tamamlanmış olur. Derin öğrenme mimarilerinde en sık uygulanan eğitim modeli transfer eğitimidir. Son eğitim modeli ise yakın zamanda popüler olan negatif örneklerin transfer eğitimi ile beraber eğitime dâhil edilmesidir. Böyle durumda, eğitim transfer eğitimi ile benzer bir sürede tamamlanırken, asıl amaç ise yanlış pozitif ve yanlış negatif değerlerinde düşüşün sağlanmasına dayanır. Yapılan deneysel çalışmalarda, en güçlü ve en başarılı modeller negatif örneklerin transfer eğitimi ile beraber eğitimi dâhil edilmesi ile elde edildi. Negatif örneklerin eğitime dâhil edilmesi ile beraber başarımda önemli bir artış sağlanmış oldu. Yapılan diğer önemli değişiklikler ise özellik çıkarıcı rolünde olan CNN yapılarının ve aktivasyon fonksiyonlarının karşılaştırılmalı performans analizinden sonra en ideal olanlarının seçilmesidir. Ayrıca,

tespit başarımının artırılması için topluluk öğrenme tekniklerinin kullanılması, modellerin kolonoskopi cihazlarına entegrasi için çıkarım hızında iyileştirme yapılmasıdır. Ek olarak, farklı veri artırma tekniklerinin uygulanarak polip tespit başarımının artırılmasıdır. Özetle söylemek gerekirse, polip tespit başarımının artırılması için derin öğrenme algoritmaları ile birçok yapının hibrit bir şekilde performans analiz süzgecinden geçtikten sonra seçilerek en ideal yapıların beraber kullanılmasına dayanmaktadır.

6.2. Öneriler

Gerçek zamanlı otomatik polip tespiti sistemlerinin başarımının daha da artırılması için gelecekte şu çalışmalar yapılabilir. Veriye aç derin öğrenme mimarileri hâlihazırda olan halka açık veri setleri ile klinik seviyede bir doğruluğa ulaşması güçtür. Bazı çalışmalarda sunulan özel veri setleri ile bu klinik seviye aşılabılır, fakat çalışmaların objektifliği az olmaktadır. Yakın zamanda, polip tespiti ile ilgili daha büyük veri setlerinin sunulabilir olmasıyla beraber, gerçek zamanlı otomatik polip tespitinin başarımı klinik seviyeye ulaşabilir düzeylere çıkacaktır. Böylelikle, daha fazla çalışmalar yapılabilir ve polip tespiti için yeni tekniklerle sunularak polip tespitinin başarımı daha da artabilir.

Nesne tespit algoritmalarının son sürümü olan YOLOv4 ve YOLOv5 algoritmaları hâlihazırda en güçlü YOLO algoritmalarıdır. İlerleyen zamanlarda yeni YOLO sürümlerinin sunulması ile beraber polip tespiti sistemlerinin başarımları daha da artırabilir bir düzeye gelebilir. Burada önemli husus, herhangi bir nesne tespit algoritmasının polip tespitine en ideal şekilde uyarlanması ve gerekirse yapılarının yeniden inşa edilmesini sağlamaktadır. Böylece, güçlü ve etkili bir veri seti ve gerçek zamanlı nesne tespit algoritması ile beraber gerçek zamanlı otomatik polip tespitinin başarımı da en uygun düzeye çıkmış olacaktır.

KAYNAKÇA

1. Xi, Y. and P. Xu., 2021. Global colorectal cancer burden in 2020 and projections to 2040. **Translational Oncology**, **14**(10): 101174.
2. Zheng, Y. and Z.Z. Wang., 2021. [Interpretation of global colorectal cancer statistics]. **Zhonghua Liu Xing Bing Xue Za Zhi = Zhonghua Liuxingbingxue Zazhi**, **42**(1): 149—152.
3. <https://www.cancer.net/cancer-types/colorectal-cancer/statistics>. (Eriřim tarihi: Temmuz 2022).
4. Bien, J. and A. Lin., 2021. A Review of the Diagnosis and Treatment of Metastatic Colorectal Cancer. **JAMA**, **325**(23): 2404–2405.
5. <https://www.mayoclinic.org/diseases-conditions/colon-polyps/symptoms-causes/syc-20352875>. (Eriřim tarihi: Temmuz 2022).
6. Issa, I.A. and M. NouredDine., 2017. Colorectal cancer screening: An updated review of the available options. **World Journal of Gastroenterology**, **23**(28): 5086–5096.
7. Morimoto, L.M., P.A. Newcomb, C.M. Ulrich, R.M. Bostick, C.J. Lais and J.D. Potter., 2002. Risk factors for hyperplastic and adenomatous polyps: Evidence for malignant potential? **Cancer Epidemiology Biomarkers and Prevention**, **11**(10 I): 1012–1018.
8. https://www.cdc.gov/cancer/colorectal/basic_info/what-is-colorectal-cancer.htm. (Eriřim tarihi: Temmuz 2022).
9. Jass, J.R., 2004. SPECIAL REPORTS AND REVIEWS Hyperplastic Polyps and Colorectal Cancer: Is There a Link? **Clinical Gastroenterology and Hepatology**, **2**: 1–8.

10. Theodosi, A., S. Ouzounis, S. Kostopoulos, D. Glotsos, I. Kalatzis, V. Tzelepi, P. Ravazoula, P. Asvestas, D. Cavouras and G. Sakellaropoulos., 2021. Design of a hybrid deep learning system for discriminating between low- and high-grade colorectal cancer lesions, using microscopy images of IHC stained for AIB1 expression biopsy material. **Machine Vision and Applications**, **32**(3): 1–17.
11. Corley, D.A., C.D. Jensen, A.R. Marks, W.K. Zhao, J.K. Lee, C.A. Doubeni, A.G. Zauber, J. De Boer, B.H. Fireman, J.E. Schottinger, V.P. Quinn, N.R. Ghai, T.R. Levin and C.P. Quesenberry., 2014. Adenoma Detection Rate and Risk of Colorectal Cancer and Death A BS TR AC T. **N Engl j Med**, **14**(3): 1298–306.
12. Hacking, S.M., B. Chakraborty, R. Nasim, T. Vitkovski and R. Thomas., 2021. A Holistic Appraisal of Stromal Differentiation in Colorectal Cancer: Biology, Histopathology, Computation, and Genomics. **Pathology Research and Practice**, **220**(October 2020): 153378.
13. Pacal, I., D. Karaboga, A. Basturk, B. Akay and U. Nalbantoglu., 2020. A comprehensive review of deep learning in colon cancer. **Computers in Biology and Medicine**, **126**: 104003.
14. Pacal, I. and D. Karaboga., 2021. A robust real-time deep learning based automatic polyp detection system. **Computers in Biology and Medicine**, **134**..
15. Urban, G., P. Tripathi, T. Alkayali, M. Mittal, F. Jalali, W. Karnes and P. Baldi., 2018. Deep Learning Localizes and Identifies Polyps in Real Time With 96% Accuracy in Screening Colonoscopy. **Gastroenterology**, **155**(4): 1069-1078.e8.
16. Qadir, H.A., Y. Shin, J. Solhusvik, J. Bergsland, L. Aabakken and I. Balasingham., 2021. Toward real-time polyp detection using fully CNNs for 2D Gaussian shapes prediction. **Medical Image Analysis**, **68**: 101897.
17. Qadir, H.A., Y. Shin, J. Solhusvik, J. Bergsland, L. Aabakken and I. Balasingham., 2019. Polyp Detection and Segmentation using Mask R-CNN: Does a Deeper Feature Extractor CNN Always Perform Better? **International Symposium on Medical Information and Communication Technology, ISMICT, 2019-May**:

18. Redmon, J., S. Divvala, R. Girshick and A. Farhadi., 2016. You only look once: Unified, real-time object detection. **Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016-Decem:** 779–788.
19. Sornapudi, S., F. Meng and S. Yi., 2019. Region-based automated localization of colonoscopy and wireless capsule endoscopy polyps. **Applied Sciences (Switzerland), 9(12):.**
20. Shin, Y., H.A. Qadir, L. Aabakken, J. Bergsland and I. Balasingham., 2019. Automatic colon polyp detection using region based deep CNN and post learning approaches. **ArXiv**, 1906.11463
21. Faguet, G.B., 2015. A brief history of cancer: Age-old milestones underlying our current knowledge database. **International Journal of Cancer**, **136(9):** 2022–2036.
22. Baykara, O., 2016. Current Modalities in Treatment of Cancer. **Balıkesir Health Sciences Journal**, **5(3):** 154–165.
23. <https://www.mayoclinic.org/diseases-conditions/colon-cancer/symptoms-causes/syc-20353669>. Accessed July 8, 2021.
24. <https://www.mayoclinic.org/diseases-conditions/cancer/symptoms-causes/syc-20370588>. Accessed August 7, 2021.
25. Martínez-Reyes, I. and N.S. Chandel., 2021. Cancer metabolism: looking forward. **Nature Reviews Cancer**, **21(10):** 669–680.
26. Stintzing, S., 2014. Management of colorectal cancer. **F1000Prime Reports**, **6(November):**
27. Yoshihara, M., T. Hiyama and S. Tanaka., 2007. Epidemiology of colorectal cancer. **Nihon Naika Gakkai Zasshi. The Journal of the Japanese Society of Internal Medicine**, **96(2):** 200–206.
28. Kakoi, H. and F. Hiraide., 1987. A Histological Study of Formation and Growth

of Nasal Polyps. **Acta Oto-Laryngologica**, **103**(1–2): 137–144.

29. Goldstein, N.S., 2006. Serrated pathway and APC (conventional)-type colorectal polyps: Molecular-morphologic correlations, genetic pathways, and implications for classification. **American Journal of Clinical Pathology**, **125**(1): 146–153.
30. Thompson-Fawcett, M.W., V.A. Marcus, M. Redston, Z. Cohen and R.S. McLeod., 2001. Adenomatous polyps develop commonly in the ileal pouch of patients with familial adenomatous polyposis. **Diseases of the Colon and Rectum**, **44**(3): 347–353.
31. Schreuders, E.H., A. Ruco, L. Rabeneck, R.E. Schoen, J.J.Y. Sung, G.P. Young and E.J. Kuipers., 2015. Colorectal cancer screening: A global overview of existing programmes. **Gut**, **64**(10): 1637–1649.
32. Regula, J., M. Rupinski, E. Kraszewska, M. Polkowski, J. Pachlewski, J. Orłowska, M.P. Nowacki and E. Butruk., 2006. Colonoscopy in Colorectal-Cancer Screening for Detection of Advanced Neoplasia. **New England Journal of Medicine**, **355**(18): 1863–1872.
33. Bretthauer, M., 2011. Colorectal cancer screening. **Journal of Internal Medicine**, **270**(2): 87–98.
34. Winawer, S.J., 2007. Colorectal cancer screening. **Best Practice and Research: Clinical Gastroenterology**, **21**(6): 1031–1048.
35. Lecun, Y., Y. Bengio and G. Hinton., 2015. Deep learning. **Nature**, **521**(7553): 436–444.
36. Schiele, S., T.T. Arndt, B. Martin, S. Miller, S. Bauer, B.M. Banner, E.M. Brendel, G. Schenkirsch, M. Anthuber, R. Huss, B. Märkl and G. Müller., 2021. Deep learning prediction of metastasis in locally advanced colon cancer using binary histologic tumor images. **Cancers**, **13**(9):.
37. Tajbakhsh, N., J.Y. Shin, S.R. Gurudu, R.T. Hurst, C.B. Kendall, M.B. Gotway and J. Liang., 2017. Convolutional Neural Networks for Medical Image Analysis: Full Training or Fine Tuning? **ArXiv**, **35**(5): 1299–1312.

38. Wang, J., H. Zhu, S.H. Wang and Y.D. Zhang., 2021. A Review of Deep Learning on Medical Image Analysis. **Mobile Networks and Applications**, **26**(1): 351–380.
39. Ibrahim, D.M., N.M. Elshennawy and A.M. Sarhan., 2021. Deep-chest: Multi-classification deep learning model for diagnosing COVID-19, pneumonia, and lung cancer chest diseases. **Computers in Biology and Medicine**, **132**: 104348.
40. Wang, P., P. Liu, J.R. Glissen Brown, T.M. Berzin, G. Zhou, S. Lei, X. Liu, L. Li and X. Xiao., 2020. Lower Adenoma Miss Rate of Computer-Aided Detection-Assisted Colonoscopy vs Routine White-Light Colonoscopy in a Prospective Tandem Study. **Gastroenterology**, **159**(4): 1252-1261.e5.
41. Pacal, I., A. Karaman, D. Karaboga, B. Akay, A. Basturk, U. Nalbantoglu and S. Coskun., 2022. An efficient real-time colonic polyp detection with YOLO algorithms trained by using negative samples and large datasets. **Computers in Biology and Medicine**, **141**(September 2021): 105031.
42. Liu, X., X. Guo, Y. Liu and Y. Yuan., 2021. Consolidated domain adaptive detection and localization framework for cross-device colonoscopic images. **Medical Image Analysis**, 102052.
43. Guo, Y., Y. Liu, A. Oerlemans, S. Lao, S. Wu and M.S. Lew., 2016. Deep learning for visual understanding: A review. **Neurocomputing**, **187**: 27–48.
44. Bora, K., M.K. Bhuyan, K. Kasugai, S. Mallik and Z. Zhao., 2021. Computational learning of features for automated colonic polyp classification. **Scientific Reports**, **11**(1): 1–16.
45. Durak, S., B. Bayram, T. Bakırman, M. Erkut, M. Doğan, M. Gürtürk and B. Akpınar., 2021. Deep neural network approaches for detecting gastric polyps in endoscopic images. **Medical and Biological Engineering and Computing**, **59**(7–8): 1563–1574.
46. Ali, M. and R. Ali., 2021. Multi-input dual-stream capsule network for improved lung and colon cancer classification. **Diagnostics**, **11**(8): 1–18.

47. Nogueira-Rodríguez, A., R. Domínguez-Carbajales, F. Campos-Tato, J. Herrero, M. Puga, D. Remedios, L. Rivas, E. Sánchez, Á. Iglesias, J. Cubiella, F. Fdez-Riverola, H. López-Fernández, M. Reboiro-Jato and D. Glez-Peña., 2021. Real-time polyp detection model using convolutional neural networks. **Neural Computing and Applications**, 4:.
48. Hoang, M.C., K.T. Nguyen, J. Kim, J. Park and C. Kim., 2021. Automated Bowel Polyp Detection Based on Actively Controlled Capsule Endoscopy: Feasibility Study. **Diagnostics**, 11(10): 1878.
49. Wang, T., Y. Lei, Y. Fu, J.F. Wynne, W.J. Curran, T. Liu and X. Yang., 2021. A review on medical imaging synthesis using deep learning and its clinical applications. **Journal of Applied Clinical Medical Physics**, 22(1): 11–36.
50. Noble, W.S., 2006. What is a support vector machine? **Nature Biotechnology**, 24(12): 1565–1567.
51. Ganaie, M.A., M. Hu, M. Tanveer* and P.N. Suganthan*., 2021. Ensemble deep learning: A review.
52. Movahedi, F., J.L. Coyle and E. Sejdic., 2018. Deep belief networks for electroencephalography: A review of recent contributions and future outlooks. **IEEE Journal of Biomedical and Health Informatics**, 22(3): 642–652.
53. <https://dx.doi.org/10.1109%2FJBHI.2017.2727218>Vieira, S., R. Garcia-Dias and A. Mechelli., 2019. Autoencoders. **Machine Learning: Methods and Applications to Brain Disorders**, 193–208.
54. Patil, A. and M. Rane., 2021. Convolutional Neural Networks: An Overview and Its Applications in Pattern Recognition. **Smart Innovation, Systems and Technologies**, 195: 21–30.
55. Abiodun, O.I., A. Jantan, A.E. Omolara, K.V. Dada, A.M. Umar, O.U. Linus, H. Arshad, A.A. Kazaure, U. Gana and M.U. Kiru., 2019. Comprehensive Review of Artificial Neural Network Applications to Pattern Recognition. **IEEE Access**, 7: 158820–158846.

56. Hasson, U., S.A. Nastase and A. Goldstein., 2020. Direct Fit to Nature: An Evolutionary Perspective on Biological and Artificial Neural Networks. **Neuron**, **105**(3): 416–434.
57. Zador, A.M., 2019. A critique of pure learning and what artificial neural networks can learn from animal brains. **Nature Communications**, **10**(1):.
58. Nasser, I.M. and S.S. Abu-Naser., 2019. Predicting Tumor Category Using Artificial Neural Networks. **International Journal of Academic Health and Medical Research**, **3**(2): 1–7.
59. Santos, I., L. Castro, N. Rodriguez-Fernandez, Á. Torrente-Patiño and A. Carballal., **2021**. Artificial Neural Networks and Deep Learning in the Visual Arts: a review.
60. Han, S.-H., K.W. Kim, S. Kim and Y.C. Youn., 2018. Artificial Neural Network: Understanding the Basic Concepts without Mathematics. **Dementia and Neurocognitive Disorders**, **17**(3): 83.
61. Emmert-Streib, F., Z. Yang, H. Feng, S. Tripathi and M. Dehmer., 2020. An Introductory Review of Deep Learning for Prediction Models With Big Data. **Frontiers in Artificial Intelligence**, **3**(February): 1–23.
62. Sengupta, S., S. Basak, P. Saikia, S. Paul, V. Tsalavoutis, F. Atiah, V. Ravi and A. Peters., 2020. A review of deep learning with special emphasis on architectures, applications and recent trends. **Knowledge-Based Systems**, **194**: 105596.
63. Lecun, Y., L. Bottou, Y. Bengio and P. Ha., 1998. LeNet. **Proceedings of the IEEE**, (November): 1–46.
64. Shrestha, A. and A. Mahmood., 2019. Review of deep learning algorithms and architectures. **IEEE Access**, **7**: 53040–53065.
65. Redmon, J. and A. Farhadi., 2018. YOLOv3: An incremental improvement. **ArXiv**.

66. Misra, D., 2019. Mish: A self regularized non-monotonic neural activation function. **ArXiv**.
67. Ramachandran, P., B. Zoph and Q. V. Le., 2017. Searching for activation functions. **ArXiv**, 1–13.
68. Elfving, S., E. Uchibe and K. Doya., 2018. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. **Neural Networks**, **107**(2015): 3–11.
69. Sherstinsky, A., 2020. Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network. **Physica D: Nonlinear Phenomena**, **404**(March): 1–43.
70. Ren, S., K. He, R. Girshick and J. Sun., 2017. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, **39**(6): 1137–1149.
71. Liu, W., D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.Y. Fu and A.C. Berg., 2016. SSD: Single shot multibox detector. **Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, **9905 LNCS**: 21–37.
72. Bochkovskiy, A., C.Y. Wang and H.Y.M. Liao., 2020. YOLOv4: Optimal Speed and Accuracy of Object Detection. **ArXiv**.
73. Redmon, J. and A. Farhadi., 2017. YOLO9000: Better, faster, stronger. **Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, 2017-Janua**: 6517–6525.
74. Lin, T.Y., P. Goyal, R. Girshick, K. He and P. Dollar., 2020. Focal Loss for Dense Object Detection. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, **42**(2): 318–327.
75. Girshick, R., J. Donahue, T. Darrell and J. Malik., 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. **Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern**

Recognition, 580–587.

76. Szegedy, C., W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich., 2015. Going deeper with convolutions. **Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 07-12-June**: 1–9.
77. M., E., V.-G. L., W.C.K. I., W. J. and Z. A., 2010. The Pascal Visual Object Classes (VOC) Challenge. **International Journal of Computer Vision**, **88**(2): 303–338.
78. Lin, T.Y., M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár and C.L. Zitnick., 2014. Microsoft COCO: Common objects in context. **Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, **8693 LNCS**(PART 5): 740–755.
79. Wang, C.Y., H.Y. Mark Liao, Y.H. Wu, P.Y. Chen, J.W. Hsieh and I.H. Yeh., 2020. CSPNet: A new backbone that can enhance learning capability of CNN. **IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2020-June**: 1571–1580.
80. Wang, C.Y., A. Bochkovskiy and H.Y.M. Liao., 2020. Scaled-YOLOv4: Scaling cross stage partial network. **ArXiv**.
81. Jocher, G., A. Stoken, J. Borovec, NanoCode012, ChristopherSTAN, L. Changyu, Laughing, tkianai, yxNONG, A. Hogan, lorenzomamma, AlexWang1900, A. Chaurasia, L. Diaconu, Marc, wanghaoyang0106, ml5ah, Doug, Durgesh, F. Ingham, Frederik, Guilhen, A. Colmagro, H. Ye, Jacobsolawetz, J. Poznanski, J. Fang, J. Kim, K. Doan and 于力军 L.Y., 2021. ultralytics/yolov5: v4.0 - nn.SiLU() activations, Weights & Biases logging, PyTorch Hub integration.
82. Girshick, R., 2015. Fast R-CNN. **Proceedings of the IEEE International Conference on Computer Vision, 2015 Inter**: 1440–1448.

83. He, K., G. Gkioxari, P. Dollár and R. Girshick., 2020. Mask R-CNN. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, 42(2): 386–397.
84. Duan, K., S. Bai, L. Xie, H. Qi, Q. Huang and Q. Tian., 2019. CenterNet: Keypoint triplets for object detection. **Proceedings of the IEEE International Conference on Computer Vision, 2019-Octob**: 6568–6577.
85. Law, H. and J. Deng., 2020. CornerNet: Detecting Objects as Paired Keypoints. **International Journal of Computer Vision**, 128(3): 642–656.
86. Tian, Z., C. Shen, H. Chen and T. He., 2019. FCOS: Fully convolutional one-stage object detection. **Proceedings of the IEEE International Conference on Computer Vision, 2019-Octob**: 9626–9635.
87. Gonzalez, T.F., 2007. Handbook of approximation algorithms and metaheuristics. **Handbook of Approximation Algorithms and Metaheuristics**, 1–1432.
88. Russakovsky, O., J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A.C. Berg and L. Fei-Fei., 2015. ImageNet Large Scale Visual Recognition Challenge. **International Journal of Computer Vision**, 115(3): 211–252.
89. Zeiler, M.D. and R. Fergus., 2012. Visualizing and Understanding Convolutional Networks.
90. Simonyan, K. and A. Zisserman., 2015. Very deep convolutional networks for large-scale image recognition. **3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings**, 1–14.
91. He, K., X. Zhang, S. Ren and J. Sun., 2016. Deep residual learning for image recognition. **Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016-Decem**: 770–778.
92. Sánchez-Peralta, L.F., L. Bote-Curiel, A. Picón, F.M. Sánchez-Margallo and J.B. Pagador., 2020. Deep learning to find colorectal polyps in colonoscopy: A systematic literature review. **Artificial Intelligence in Medicine**, 108(March):

101923.

93. Zhou, S.K., H. Greenspan, C. Davatzikos, J.S. Duncan, B. Van Ginneken, A. Madabhushi, J.L. Prince, D. Rueckert and R.M. Summers., 2021. A Review of Deep Learning in Medical Imaging: Imaging Traits, Technology Trends, Case Studies with Progress Highlights, and Future Promises. **Proceedings of the IEEE**, **109**(5): 820–838.
94. Gulum, M.A., C.M. Trombley and M. Kantardzic., 2021. A review of explainable deep learning cancer detection models in medical imaging. **Applied Sciences (Switzerland)**, **11**(10):.
95. Ozawa, T., S. Ishihara, M. Fujishiro, Y. Kumagai, S. Shichijo and T. Tada., 2020. Automated endoscopic detection and classification of colorectal polyps using convolutional neural networks. **Therapeutic Advances in Gastroenterology**, **13**:.
96. Saito, H., T. Aoki, K. Aoyama, Y. Kato, A. Tsuboi, A. Yamada, M. Fujishiro, S. Oka, S. Ishihara, T. Matsuda, M. Nakahori, S. Tanaka, K. Koike and T. Tada., 2020. Automatic detection and classification of protruding lesions in wireless capsule endoscopy images based on a deep convolutional neural network. **Gastrointestinal Endoscopy**, **92**(1): 144-151.e1.
97. Wang, D., N. Zhang, X. Sun, P. Zhang, C. Zhang, Y. Cao and B. Liu., 2019. AFP-Net: Realtime anchor-free polyp detection in colonoscopy. **Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI, 2019-Novem**: 636–643.
98. Nadimi, E.S., M.M. Buijs, J. Herp, R. Kroijer, M. Kobaek-Larsen, E. Nielsen, C.D. Pedersen, V. Blanes-Vidal and G. Baatrup., 2020. Application of deep learning for autonomous detection and localization of colorectal polyps in wireless colon capsule endoscopy. **Computers and Electrical Engineering**, **81**:.
99. Liu, M., J. Jiang and Z. Wang., 2019. Colonic Polyp Detection in Endoscopic Videos with Single Shot Detection Based Deep Convolutional Neural Network. **IEEE Access**, **7**: 45058–45066.

100. Mostafiz, R., M. Hasan, I. Hossain and M.M. Rahman., 2020. An intelligent system for gastrointestinal polyp detection in endoscopic video using fusion of bidimensional empirical mode decomposition and convolutional neural network features. **International Journal of Imaging Systems and Technology**, 30(1): 224–233.
101. Hasan, M.M., N. Islam and M.M. Rahman., 2020. Gastrointestinal polyp detection through a fusion of contourlet transform and Neural features. **Journal of King Saud University - Computer and Information Sciences**, (xxxx):
102. Laddha, M., S. Jindal and J. Wojciechowski., 2019. Gastric polyp detection using deep convolutional neural network. **ACM International Conference Proceeding Series**, (304): 55–59.
103. Wittenberg, T., P. Zobel, M. Rathke and S. Mühldorfer., 2019. Computer Aided Detection of Polyps in White-light-Colonoscopy Images using Deep Neural Networks. **Current Directions in Biomedical Engineering**, 5(1): 231–234.
104. Yuan, Y., D. Li and M.Q.H. Meng., 2018. Automatic Polyp Detection via a Novel Unified Bottom-Up and Top-Down Saliency Approach. **IEEE Journal of Biomedical and Health Informatics**, 22(4): 1250–1260.
105. Misawa, M., S. ei Kudo, Y. Mori, K. Hotta, K. Ohtsuka, T. Matsuda, S. Saito, T. Kudo, T. Baba, F. Ishida, H. Itoh, M. Oda and K. Mori., 2021. Development of a computer-aided detection system for colonoscopy and a publicly accessible large colonoscopy video database (with video). **Gastrointestinal Endoscopy**, 93(4): 960-967.e3.
106. Zhang, R., Y. Zheng, C.C.Y. Poon, D. Shen and J.Y.W. Lau., 2018. Polyp detection during colonoscopy using a regression-based convolutional neural network with a tracker. **Pattern Recognition**, 83: 209–219.
107. Pogorelov, K., O. Ostroukhova, M. Jeppsson, H. Espeland, C. Griwodz, T. De Lange, D. Johansen, M. Riegler and P. Halvorsen., 2018. Deep Learning and Hand-Crafted Feature Based Approaches for Polyp Detection in Medical Videos. **Proceedings - IEEE Symposium on Computer-Based Medical Systems**, 2018-

June: 381–386.

108. Li, L., 2020. Recognizing Polyps in Wireless Endoscopy Images Using Deep Stacked Auto Encoder with Constraint Image Model in Flexible Medical Sensor Platform. **IEEE Access**, **8**: 60653–60663.
109. Zheng, Z., P. Wang, W. Liu, J. Li, R. Ye and D. Ren., 2019. Distance-IoU loss: Faster and better learning for bounding box regression. **ArXiv**, (2):.
110. Esteva, A., A. Robicquet, B. Ramsundar, V. Kuleshov, M. DePristo, K. Chou, C. Cui, G. Corrado, S. Thrun and J. Dean., 2019. A guide to deep learning in healthcare. **Nature Medicine**, **25**(1): 24–29.
111. Shamshirband, S., M. Fathi, A. Dehzangi, A.T. Chronopoulos and H. Alinejad-Rokny., 2021. A review on deep learning approaches in healthcare systems: Taxonomies, challenges, and open issues. **Journal of Biomedical Informatics**, **113**: 103627.
112. Miotto, R., F. Wang, S. Wang, X. Jiang and J.T. Dudley., 2017. Deep learning for healthcare: Review, opportunities and challenges. **Briefings in Bioinformatics**, **19**(6): 1236–1246.
113. Bernal, J., N. Tajkbaksh, F.J. Sanchez, B.J. Matuszewski, H. Chen, L. Yu, Q. Angermann, O. Romain, B. Rustad, I. Balasingham, K. Pogorelov, S. Choi, Q. Debard, L. Maier-Hein, S. Speidel, D. Stoyanov, P. Brandao, H. Cordova, C. Sanchez-Montes, S.R. Gurudu, G. Fernandez-Esparrach, X. Dray, J. Liang and A. Histace., 2017. Comparative Validation of Polyp Detection Methods in Video Colonoscopy: Results from the MICCAI 2015 Endoscopic Vision Challenge. **IEEE Transactions on Medical Imaging**, **36**(6): 1231–1249.
114. Sánchez-Peralta, L.F., J.B. Pagador, A. Picón, Á.J. Calderón, F. Polo, N. Andracka, R. Bilbao, B. Glover, C.L. Saratxaga and F.M. Sánchez-Margallo., 2020. Piccolo white-light and narrow-band imaging colonoscopic dataset: A performance comparative of models and datasets. **Applied Sciences (Switzerland)**, **10**(23): 1–14.

115. Pacal, I., A. Karaman, D. Karaboga, B. Akay, A. Basturk, U. Nalbantoglu and S. Coskun., 2022. An efficient real-time colonic polyp detection with YOLO algorithms trained by using negative samples and large datasets. **Computers in Biology and Medicine**, **141**(September 2021): 105031.
116. Misawa, M., S. ei Kudo, Y. Mori, K. Hotta, K. Ohtsuka, T. Matsuda, S. Saito, T. Kudo, T. Baba, F. Ishida, H. Itoh, M. Oda and K. Mori., **2021**. Development of a computer-aided detection system for colonoscopy and a publicly accessible large colonoscopy video database (with video). American Society for Gastrointestinal Endoscopy.
117. Karaboga, D. and B. Basturk., 2007. Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization Problems, In , pp: 789–798.
118. Katoch, S., S.S. Chauhan and V. Kumar., **2021**. A review on genetic algorithm : past , present , and future. Multimedia Tools and Applications.
119. Rezatofighi, S.H., N. Tsoi, J. Gwak, A. Sadeghian, I.D. Reid and S. Savarese., 2019. Generalized Intersection over Union: {A} Metric and {A} Loss for Bounding Box Regression. **CoRR**, **abs/1902.0**:
120. Tan, M., R. Pang and Q. V. Le., 2020. EfficientDet: Scalable and efficient object detection. **Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition**, 10778–10787.
121. Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser and I. Polosukhin., 2017. Attention is all you need. **Advances in Neural Information Processing Systems, 2017-Decem**(Nips): 5999–6009.
122. Dosovitskiy, A., L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit and N. Houlsby., 2020. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. 1–21.
123. Kang, J. and J. Gwak., 2019. Ensemble of Instance Segmentation Models for Polyp Segmentation in Colonoscopy Images. **IEEE Access**, **7**: 26440–26447.

124. Jia, X., X. Mai, Y. Cui, Y. Yuan, X. Xing, H. Seo, L. Xing and M.Q.H. Meng., 2020. Automatic Polyp Recognition in Colonoscopy Images Using Deep Learning and Two-Stage Pyramidal Feature Prediction. **IEEE Transactions on Automation Science and Engineering**, 17(3): 1570–1584.
125. Xu, J., R. Zhao, Y. Yu, Q. Zhang, X. Bian, J. Wang, Z. Ge and D. Qian., 2021. Real-time automatic polyp detection in colonoscopy using feature enhancement module and spatiotemporal similarity correlation unit. **Biomedical Signal Processing and Control**, 66(February): 102503.
126. Liu, X., X. Guo, Y. Liu and Y. Yuan., 2021. Consolidated domain adaptive detection and localization framework for cross-device colonoscopic images. **Medical Image Analysis**, 102052.
127. Shen, Z., C. Lin and S. Zheng., 2021. COTR: Convolution in Transformer Network for End to End Polyp Detection. 1–10.

EKLER

Derin öğrenme tabanlı gerçek zamanlı polip tespiti yöntemlerinin geliştirilmesi ve incelenmesi adlı tezde yapılan deneysel hesaplamalar “Erciyes Üniversitesi Büyük Veri ve Yapay Zekâ Merkezinde” bulunan hesaplama birimleri üzerinde gerçekleştirilmiştir. Önerilen özel veri setine ait etik izinler “Acıbadem Hastanesi tarafından ATADEK-2021/02 izin numarası ve 2021-02/15 karar numarası” ile verilmiştir.



ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı: İshak PAÇAL

Uyruğu: Türkiye (T.C)

EĞİTİM

Derece	Kurum	Mezuniyet Tarihi
Yüksek Lisans	The University of Nottingham, Elektronik Haberleşme ve Bilgisayar Mühendisliği	2016
Lisans	Harran Üniversitesi, Bilgisayar Mühendisliği	2013
Lise	Atatürk YDA Lisesi, Iğdır	2007

İŞ DENEYİMLERİ

Yıl	Kurum	Görev
2017-Halen	Iğdır Üniversitesi	Araştırma Görevlisi
20013-2015	MEB YLSY	MEB

YABANCI DİL

İngilizce

YAYINLAR

SCI, SSCI, AHCI 'İndekslerine Giren Dergilerde Yayınlanan Makaleler

1. Pacal, Ishak & Karaboga, Dervis & Basturk, Alper & Akay, Bahriye & Nalbantoglu, Ozkan. (2020). A comprehensive review of deep learning in colon cancer. Computers in biology and medicine. 126. 104003. 10.1016/j.compbiomed.2020.104003.
2. Pacal, Ishak & Karaboga, Dervis. (2021). A Robust Real-Time Deep Learning Based Automatic Polyp Detection System. Computers in Biology and Medicine. 134. 104519. 10.1016/j.compbiomed.2021.104519.
3. Pacal, Ishak & Karaman, Ahmet & Karaboga, Dervis & Akay, Bahriye & Basturk, Alper & Nalbantoglu, Ozkan & Coskun, Seymanur. (2021). An efficient real-time colonic polyp detection with YOLO algorithms trained by using negative samples and large datasets. Computers in Biology and Medicine. 141. 105031. 10.1016/j.compbiomed.2021.105031.

SCI İndekslerine Giren Dergi Sistemlerine Yüklenen ve İncelemedeki Makaleler

1. Karaman, Ahmet & Karaboga, Dervis & Pacal, Ishak & Akay, Bahriye & Basturk, Alper & Nalbantoglu, Ozkan & Coskun, Seymanur & Sahin, Omur. (2022). Hyper-parameter optimization of deep learning architectures using artificial bee colony (ABC) algorithm for high performance real-time automatic colorectal cancer (CRC) polyp detection. Applied Intelligence. (Kabul aşamasında)
2. Karaman, Ahmet & Pacal, Ishak & Karaboga, Dervis & Akay, Bahriye & Basturk, Alper & Nalbantoglu, Ozkan & Coskun, Seymanur & Sahin, Omur. (2022). Evaluation and optimization of activation functions for deep learning architectures using artificial bee colony (ABC) algorithm for real-time polyp detection. Computers Methods and Programs in Medicine. (Yüklemede)