

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

**DERİN ÖĞRENME İLE NESNE ALGILAMADA
TRANSFER ÖĞRENME VE İNCE AYAR İŞLEMLERİNİN
ETKİNLİKLERİNİN ARAŞTIRILMASI**

**Hazırlayan
Mehmet Uğur TÜRKDAMAR**

**Danışman
Doç. Dr. Celal ÖZTÜRK**

Yüksek Lisans Tezi

**Ocak 2023
KAYSERİ**

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

**DERİN ÖĞRENME İLE NESNE ALGILAMADA
TRANSFER ÖĞRENME VE İNCE AYAR İŞLEMLERİNİN
ETKİNLİKLERİNİN ARAŞTIRILMASI
(Yüksek Lisans Tezi)**

**Hazırlayan
Mehmet Uğur TÜRKDAMAR**

**Danışman
Doç. Dr. Celal ÖZTÜRK**

**Ocak 2023
KAYSERİ**

BİLİMSEL ETİĞE UYGUNLUK

Bu 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.

Mehmet Uğur TÜRKDAMAR

“DERİN ÖĞRENME İLE NESNE ALGILAMADA TRANSFER ÖĞRENME VE İNCE AYAR İŞLEMLERİNİN ETKİNLİKLERİNİN ARAŞTIRILMASI” adlı Yüksek Lisans Tezi, Erciyes Üniversitesi Lisansüstü Tez Önerisi ve Tez Yazma Yönergesi’ ne uygun olarak hazırlanmıştır.

Hazırlayan

Mehmet Uğur TÜRKDAMAR

İmza

Danışman

Doç. Dr. Celal ÖZTÜRK

İmza

Bilgisayar Mühendisliği ABD Başkanı

Prof. Dr. Veysel ASLANTAŞ

İmza



TEŞEKKÜR

Bana çalışmalarım süresince her türlü yardımı ve fedakârlığı sağlayan Doç. Dr. Celal Öztürk ve Dr. Öğr. Üyesi Murat Taşyürek hocalarıma ve aileme teşekkür ederim.

Mehmet Uğur TÜRKDAMAR

Ocak 2023, KAYSERİ



DERİN ÖĞRENME İLE NESNE ALGILAMADA TRANSFER ÖĞRENME VE İNCE AYAR İŞLEMLERİNİN ETKİNLİKLERİNİN ARAŞTIRILMASI

Mehmet Uğur TÜRKDAMAR

**Erciyes Üniversitesi, Fen Bilimleri Enstitüsü
Yüksek Lisans Tezi, Ocak 2023
Danışman: Doç. Dr. Celal ÖZTÜRK**

ÖZET

Birçok alanda yerini alan makine insan etkileşimi sayesinde dil bilmeme sorunu ortadan kalkmakta, ses ve görüntü tanıma ile kişisel ve toplumsal ihtiyaçlar sağlanmaktadır. Uydu görüntüleri yeni teknolojiler ile anlamlandırılarak doğal afet takibi yapılabilmekte ve sınır güvenliği gibi stratejik adımlar keskin bir şekilde atılabilmektedir. Bilgisayarlı görü alt dalı olan nesne tespiti sayesinde sayılan örnek işler ve dahası yapılmaktadır. Hesap yükü az algoritmaların bilimsel çalışmalarla, veriyi paralel işleyen ekran kartlarının yatırımlarla, veri setine erişimin yaygınlaşması sonucu ile de derin öğrenme çalışmaları hız kazanmıştır. Çokça görüntü verisi üzerinde çalışan ESA tabanlı derin öğrenme ağları ile uygulanan nesne tespiti hayatımızı kolaylaştıran teknolojilerde daha etkin rol oynamaktadır. Bu tez kapsamında donanım, zaman ve veri kaynaklarını daha az harcamaya adına insan öğrenmesinden ilham alınarak geliştirilen transfer öğrenme yaklaşımı kullanılmıştır. Transfer öğrenme 3 farklı veri seti üzerinde çalışan güncel ESA tabanlı nesne tespitleyici derin öğrenme ağlarından YOLO v4, YOLO v5 ve Faster R-CNN'e uygulanarak ağların performansları, transfer öğrenmenin uygulanmadığı aynı ağların performansları ile karşılaştırılmıştır. Çalışmada ayrıca görüntülerdeki küçük nesnelerin tespitini mümkün kılan bir yöntem olarak ince ayar tekniği önerilmiştir. Daha önceki çalışmalarda mimari değişikliğiyle veya ağın genelleme kabiliyetini kaybetmesine neden olan probleme özel veri seti seçimi ile küçük nesne algılamanın üstesinden gelinirken, bu tez ile mimari değişimden ve veri setinden bağımsız bir çözüm önerilmiştir.

Anahtar Kelimeler: Derin öğrenme, nesne tespiti, ESA, transfer öğrenme, ince ayar

STUDYING EFFECTIVENESS OF TRANSFER LEARNING AND FINE TUNING PROCESSES IN OBJECT DETECTION WITH DEEP LEARNING

Mehmet Uğur TÜRKDAMAR

Erciyes University, Graduate School of Natural and Applied Sciences

Master Thesis, January 2023

Supervisor: Associate Professor Celal ÖZTÜRK

ABSTRACT

With machine-human interaction, which has taken its place in many areas, the problem of not knowing the language is eliminated, and personal and social needs are met with voice and image recognition. By making sense of satellite images with new technologies, natural disaster monitoring can be carried out and strategic steps such as border security can be taken sharply. With object detection, which is a sub-branch of computer vision, exemplary things and more are done. Deep learning studies have gained momentum as a result of the scientific studies of algorithms with low computational cost, the investments in graphics cards that process data in parallel, and the widespread access to the data set. Object detection, implemented with ESA-based deep learning networks working on a lot of image data, plays a more effective role in technologies that make our lives easier. In this thesis, the transfer learning approach which is inspired by human learning, was used in order to spend less on hardware, time and data. Transfer learning was applied to YOLO v4, YOLO v5 and Faster R-CNN, one of the current ESA-based object detection deep learning networks working on 3 different data sets, and the performances of the networks were compared with the performances of the same networks without transfer learning. In the study, fine-tuning technique is also proposed as a method that makes it possible to detect small objects in images. While in previous studies, small object detection can be overcome with architectural change or with the selection of a problem-specific dataset that caused the network to lose its generalization ability, with this thesis, a solution which is independent of architectural change and dataset is proposed.

Keywords: Deep learning, object detection, CNN, transfer learning, fine-tuning

İÇİNDEKİLER

DERİN ÖĞRENME İLE NESNE ALGILAMADA TRANSFER ÖĞRENME VE İNCE AYAR İŞLEMLERİNİN ETKİNLİKLERİNİN ARAŞTIRILMASI

YÖNERGEYE UYGUNLUK.....	ii
KABUL VE ONAY	iv
TEŞEKKÜR	v
ÖZET.....	vi
ABSTRACT	vii
İÇİNDEKİLER	viii
KISALTMALAR	xi
TABLolar LİSTESİ.....	xii
ŞEKİLLER LİSTESİ.....	xiii
GİRİŞ	1

1. BÖLÜM

PROBLEM DURUMU, AMAÇ VE ÖNEM

1.1. Problem Durumu	3
1.2. Araştırmanın Amacı	5
1.3. Araştırmanın Önemi.....	6

2. BÖLÜM

GENEL BİLGİLER ve LİTERATÜR ÇALIŞMASI

2.1. Bilgisayarlı Görü.....	7
2.2 Makine Öğrenme.....	7

2.3. Derin Öğrenme.....	8
2.4. Evrişimli Sinir Ağları.....	9
2.5. Transfer Öğrenme.....	10
2.6. İnce Ayar.....	12
2.7. Konu ile İlgili Yapılan Çalışmalar.....	13

3. BÖLÜM

YÖNTEM ve MATERYAL

3.1. Yöntem	19
3.1.1. Transfer Öğrenme Yönteminin Kullanılması	21
3.1.2. İnce Ayar Yönteminin Kullanılması	23
3.1.3. Modellere Uygulanan Ayarlar	28
3.2. Materyal.....	31
3.2.1. Faster R-CNN Yöntemi	34
3.2.2. YOLO Algoritması.....	36

4. BÖLÜM

BULGULAR

4.1. Araştırma Modeli.....	45
4.2. Deneyler	47

5. BÖLÜM

TARTIŞMA-SONUÇ ve ÖNERİLER

5.1. Tartışma.....	60
5.2. Sonuç	65
5.3. Öneriler	69
KAYNAKÇA	70
ÖZGEÇMİŞ.....	78



KISALTMALAR

ESA:	Evriřimli Sinir Ađı
YOLO:	You Only Look Once (Bir Kere Bakarsın)
Faster R-CNN:	Faster Region CNN (Daha hızlı Bölgesel Evriřimli Sinir Ađı)
IoU:	Intersection over Union (Kesiřimlerin Birleřimi)
CIoU:	Complete Intersection over Union (Komple Kesiřimlerin Birleřimi)
GIoU:	Generalizable Intersection over Union (Genelleřtirilebilir Kesiřimlerin Birleřimi)
HOG:	Histogram of Oriented Gradients (Yönlendirilmiş Gradyanların Histogramı)
SIFT:	Scale-Invariant Feature Transform (Ölçekle Deđiřmeyen Özellik Dönüřümü)
SURF:	Speeded Up Robust Features (Hızlandırılmış Sağlam Özellikler)
RoI:	Region of Interests (İlgilenilen Bölge)
RPN:	Region Proposal Network (Bölge Öneren Ađ)
CSPNet:	Cross Stage Partial Network (Çapraz Ařamalı Kısmi Ađ)
PANet:	Path Aggregation Network (Yol Birleřtirme Ađı)
SPP:	Spatial Pyramid Pooling (Uzamsal Piramit Havuzu)
ResNet:	Residual Neural Network (Artan Sinir Ađı)
Darknet:	Dark Net (Karanlık Ađ - YOLO v4'ü çalıştırmaya yarayan çerçeve yazılım)
PyTorch:	Python Torch (Python Meřalesi - YOLO v4'ü çalıştırmaya yarayan çerçeve yazılım)

TABLÖLAR LİSTESİ

Tablo 1. Görüntü Çözünürlüğü ve Nesne Tespiti İlişkisi	25
Tablo 2. YOLO Varsayılan İzgara Sayıları.....	37



ŞEKİLLER LİSTESİ

Şekil 1. Yapay zeka kapsamı.....	8
Şekil 2. ESA	10
Şekil 3. ESA Tabanlı Derin Öğrenme Ağı Çalışma Yapısı.....	20
Şekil 4. Transfer Öğrenmenin Kullanımı	23
Şekil 5. Farklı Çözünürlüklerdeki Görüntüler	25
Şekil 6. Küçük Nesneler İçeren Görüntü Üzerinde Transfer Öğrenmeli YOLO v4 tespiti	31
Şekil 7. Küçük Nesneler İçeren Görüntü Üzerinde İnce Ayarlı Transfer Öğrenmeli YOLO v4 tespiti	32
Şekil 8. Zatürreli/Sağlıklı akciğer ayrımı yapan Faster R-CNN mimarisi	33
Şekil 9. YOLO çalışma mantığı	37
Şekil 10. İnce ayar için YOLO ızgara sayı artırımı	39
Şekil 11. YOLO V4 Mimarisi	40
Şekil 12. CIoU çalışma hızı.....	40
Şekil 13. YOLO v5 Mimarisi	42
Şekil 14. YOLO v5 öğrenen çapa kutuları	42
Şekil 15. Öğrenen çapa ve çapa kutu farkı	43
Şekil 16. Çalışma Akış Diyagramı	46
Şekil 17. ResNet50 ve CSPDarknet53	46
Şekil 18. Örnek etiketleme işlemi.....	48
Şekil 19. Örnek txt etiketi (MS COCO)	49
Şekil 20. Örnek xml etiketi (PASCAL VOC)	49
Şekil 21. Transfer Öğrenmesiz Darknet YOLO v4 Kask Eğitim Grafiği	50
Şekil 22. Transfer Öğrenmeli Darknet YOLO v4 Kask Eğitim Grafiği.....	51
Şekil 23. Transfer Öğrenmesiz PyTorch YOLO v4 Satranç Eğitim Grafiği.....	52
Şekil 24. Transfer Öğrenmeli PyTorch YOLO v4 Satranç Eğitim Grafiği	52
Şekil 25. Transfer Öğrenmesiz PyTorch YOLO v4 Zatürre Eğitim Grafiği	53
Şekil 26. Transfer Öğrenmeli PyTorch YOLO v4 Zatürre Eğitim Grafiği	53
Şekil 27. Transfer Öğrenmesiz YOLO v5 Kask Eğitim Grafiği	54
Şekil 28. Transfer Öğrenmeli YOLO v5 Kask Eğitim Grafiği.....	54

<i>Şekil 29.</i> Transfer Öğrenmesiz YOLO v5 Satranç Eğitim Grafiği.....	55
<i>Şekil 30.</i> Transfer Öğrenmeli YOLO v5 Satranç Eğitim Grafiği.....	55
<i>Şekil 31.</i> Transfer Öğrenmesiz YOLO v5 Zatürre Eğitim Grafiği.....	56
<i>Şekil 32.</i> Transfer Öğrenmeli YOLO v5 Zatürre Eğitim Grafiği.....	56
<i>Şekil 33.</i> Transfer Öğrenmesiz Faster R-CNN Kask Eğitim Grafiği	57
<i>Şekil 34.</i> Transfer Öğrenmeli Faster R-CNN Kask Eğitim Grafiği	57
<i>Şekil 35.</i> Transfer Öğrenmesiz Faster R-CNN Satranç Eğitim Grafiği	57
<i>Şekil 36.</i> Transfer Öğrenmeli Faster R-CNN Satranç Eğitim Grafiği.....	58
<i>Şekil 37.</i> Transfer Öğrenmesiz Faster R-CNN Zatürre Eğitim Grafiği.....	58
<i>Şekil 38.</i> Transfer Öğrenmeli Faster R-CNN Zatürre Eğitim Grafiği.....	58
<i>Şekil 39.</i> Sınıflandırmada kullanılan YOLO v5 mozaik veri çoklama işlemi.....	63
<i>Şekil 40.</i> Örnek bir görüntü üzerinde karşılaştırma metrik değerlerini oluşturan doğru pozitif, yanlış pozitif ve yanlış negatif değerlerinin gösterimi	64
<i>Şekil 41.</i> YOLO v4 İnce Ayar Etkinliği F1 Skor Karşılaştırması	67
<i>Şekil 42.</i> Farklı Ağlardaki Transfer Öğrenme Etkinliği F1 Skor Karşılaştırması	67
<i>Şekil 43.</i> Ağlar Arası Yöntemlerin Aldığı En Yüksek F1 Skor Karşılaştırması	68

GİRİŞ

Gün içinde yapılacak işleri söyleyen sesli asistan, bizim yerimize evi süpüren akıllı robot, yüzümüzü görünce kilidini açan telefon hayatımızın içindedir. Benzeri teknolojilerin yaygınlaşması ve gün geçtikçe geliştirilmesi yaşamımızı büyük ölçüde kolaylaştırmaktadır. Bu kolaylaşma, farklı ve birçok verinin yapay zekâ uygulamaları tarafından tespit edilerek öğrenilmesi sonucu meydana gelmektedir. Akıllı temizlik robotu çalışacağı evin kroki görüntüsünü birçok defa görüp öğrenirken sesli asistan sahibinin sesini duydukça daha iyi öneriler vermektedir.

Gerçek görüntüyü ve nasıl tahmin yapılması gerektiğinin kuralını makineye söyleyerek tahmin işlemlerini gerçekleştirmeye eğitim denmektedir. Derin ve makine öğrenmede gerçekleşen bu eğitim sayesinde daha çok gerçek veri görülerek daha başarılı tahminler gerçekleştirilmektedir. Bu görme uygulamasının temelinde ise nesne tespiti yer almaktadır. Nesne tespiti, tarım, tıp, savunma başta olmak üzere birçok alanda geniş bir kullanıma sahiptir [1]. Tüm işlemin esas amacı girdi görüntülerindeki nesneleri tespit edip tanımadır. Bu çalışma son yıllarda hakkında çokça çalışma gerçekleştirilen derin öğrenme temelli bir nesne tespiti çalışması olarak düşük donanım özelliklerinde küçük veri setleriyle yüksek doğruluk elde etmeyi ve küçük nesneleri tespit etmeyi içermektedir.

Tezin amacı doğrultusunda, birinci bölümde çalışmanın problem durumu, önemi ve amacına değinilmiştir. İkinci bölümde, çalışmanın kavramsal çerçevesini oluşturan ESA, bilgisayarlı görü, makine ve derin öğrenme temel kavramları açıklanmış ve transfer öğrenme ve ince ayar için yapılan önceki çalışmalar irdelenmiştir. Üçüncü bölümde sırasıyla transfer öğrenme, önerilen metot olarak ince ayar açıklanarak model ayarlarına yer verilmiş, kullanılan materyal olarak derin öğrenme ağlarından YOLO v4, v5 ve Faster R-CNN ağları karşılaştırmalı olarak mimari ve çalışma mantıkları yönünden sunul-

muştur. Dördüncü bölümde, çalışmada gerçekleştirilen deneyler grafiklerle ve deneylere ilişkin sonuçlar f1 skor metriğiyle ortaya konulmuştur. Tezin son bölümünde ise, deneysel sonuçlar yorumlanarak, elde edilen başarı literatürdeki diğer çalışmalarla karşılaştırmalı olarak sunulurken ileride yapılması planlananlara ve önerilere yer verilmiştir.



1. BÖLÜM

PROBLEM DURUMU, AMAÇ VE ÖNEM

1.1. Problem Durumu

Son yıllarda nesne tespitinde derin öğrenme kullanımı klasik yaklaşıma göre çok daha iyi sonuçlar vermesine karşın, gerçekleştirilen uygulamalarda karşılaşılan bazı sınırlılıklar söz konusudur. Bu kapsamda ilk olarak değinilebilecek husus ağ eğitim süreleridir. Derin öğrenme çalışmaları ilk olarak bir bankanın gri renkteki banknotlarındaki rakamları tanıyan LeNet-5 ağı ile başlamışken, ImageNet yarışını kazanan AlexNet ile gelişimi hız kazanmıştır. Fakat bu ve devamında geliştirilen ağların çalıştırılıp yüksek doğruluklar elde etmesi için büyük veri setlerine ihtiyaç vardır. Büyük veri setlerinin toplanması ve bu veri setleri üzerindeki ağ eğitimleri ise uzun süreler almaktadır. Nesne tespitinde yüksek başarı elde etmek için çalışmada kullanılan büyük ölçekli modellerin içerdikleri ağlardan; ResNet ile transfer öğrenmesiz bir eğitimin sonucu olarak ağırlık çıktı eldesi için; normal özellikli bir ekran kartının donanım özelliklerinden daha gelişmiş, v2-8 tensör işleme birimi ile 7,3 saat çalışması, çalışmada kullanılmayan daha katmanlı yapısıyla VGG16'nın ise 15 saat çalışması gerekmektedir [2]. Kısacası uzun bekleme süreleri derin öğrenmede problem teşkil etmektedir.

İkinci sınırlılık olarak internetteki çoğu derin öğrenme uygulamasının çalıştırılabilmesi için geçmiş yıllarda oluşturulmuş büyük veri setleri PASCAL VOC [3] veya MS COCO'yu [4] zorunlu tutması dile getirilebilir. Bu veri setleri yüksek boyutlu bilgisayar depolama alanına ihtiyaç duydukları için bu durum, özellikle gerekli bilgisayar depolama alanına sahip olmayan araştırmacılarda kısıtlayıcı bir durum yaratır [5] [6].

Üçüncü bir problem olarak görüntünün derin kısımlarında yer alan küçük nesnelerin bilgisayar tarafından tespit edilememesi vardır. Derin kısımlarda bulunan nesneler, ekran düzleminden uzak, görüntüde az piksel harcayarak az yer kaplamakta ve küçük olarak adlandırılmaktadır. Bu nesneler görüntünün derin kısımlarının herhangi bir yerinde olabilmektedir ve tespit edilememeleri özellikle sağlık ve güvenlik gibi alanlarda önemli sonuçlar doğurabilmektedir. Dolayısıyla bu nesneleri tespit edecek derin öğrenme ağlarının genelleme yeteneğinin yüksek olması beklenmektedir [7] [8].

Ek olarak konuyla ilgili çalışmalar irdelendiğinde görüntü verisi kullanan ve nesne tespiti yapan birçok çalışma olmasına rağmen görüntüdeki küçük nesneleri tespit etmeye çalışan çalışma sayısının çok az olduğu görülmektedir. Sayılı olan çalışmalardan kimi küçük nesneyi tespit etmek için hiper parametre veya mimari değişikliğine giderken kimi sadece küçük nesnelerin yer aldığı veri setlerini kullanmıştır [9][10][11]. Ağır optimizasyon algoritmaları uygulamaları ise modelin daha başarılı eğitimine yöneliktir ve küçük nesne tespitinden bağımsızdır [12]. Mimari değişikliği, FLOPs sayısını değiştirerek eğitim süresini artırmakta [13], bu sebeple de hesaplamaya ek yük gelmekte, sadece küçük nesnelerin yer aldığı veri setlerini kullanım ise ağırların sadece küçük nesneleri öğrenerek orta-büyük boyutlu nesneler ile karşılaştığında tahmin yapamaması gibi hatalı, dengesiz bir durumun oluşmasına neden olmaktadır. Geleneksel hiper parametre arama algoritmalarından ızgara ve rastgele arama, başarının öncelendiği bu çalışma için kullanılan derin öğrenme ağlarında uygulanamamaktadır. Bunun yerine YOLO v5 için genetik algoritma ile en uygun hiper parametreleri bulma uygulaması vardır [14] fakat tüm hiper parametrelerin aynı anda değişimi, ağırların ek yük getireceği için tercih edilen bir yöntem olarak görülmemektedir [15]. Bu sebeplerden ötürü çalışmada, problemin farklı bir yolla, kullanılan ağırlara göre değişen hiper parametre değişimi şeklinde çözülmesi planlanmıştır.

İstenilen veri setiyle, istenilen ağırlarda uzun beklemler yaşamadan çalıştırmalar yapmak, günümüz dünyasında birçok alan için önem teşkil etmektedir. Ayrıca anlamlı görüntü verisindeki önemli olabilecek küçük nesneleri bilgisayara keşfettirmek teknolojilerin belkemiğini oluşturmaktadır. Bu kapsamda gerçekleştirilen çalışmada transfer öğrenme ve ince ayar teknikleri görüntü verisi üzerinde çalıştırılarak başarılı sonuçlar elde edilmiştir.

1.2. Araştırmanın Amacı

Bu araştırmada 1.1. problem durumunda dile getirilen nesne tespitindeki derin öğrenme çalışmalarının bazı sınırlılıklarını gidermek amaçlanmıştır. Bu kapsamda derin öğrenmenin fazla veriye ihtiyaç duyması sonucu oluşan uzun eğitim süreleri ve dolayısıyla donanım kaynağının harcanması problemine transfer öğrenme aracılığı ile çözüm bulmak. Ayrıca güvenlik, sağlık veya herhangi bir alanda önem arz edebilecek, gözden kaçabilen küçük nesneleri makineye tespit ettirmek amaçlanmıştır. Çalışmada cevabı aranacak sorular somut olarak aşağıdaki gibidir:

Araştırma soruları:

- Görüntüde yer alan küçük nesneler veri setine bağlı olmadan tespit edilebilir mi?
- Transfer öğrenme, derin öğrenme ağlarında doğruluğu artıran bir yöntem midir?
- Transfer öğrenme, küçük veri setinde derin öğrenme ağını çalıştırabildiği için büyük veri setinde derin öğrenme ağı çalıştırmaya göre süreyi azaltan bir teknik midir?

Alt araştırma soruları:

- Farklı ağlar transfer öğrenmeye nasıl tepki vermektedir? (Hangi ağ için doğruluk az artarken hangi ağ için çok artmaktadır?)
- Farklı ağlar, uygulanan ince ayar işlemine karşı nasıl tepki vermektedir? (Ağ genel doğruluğu düşüyor mu, gerçekten küçük nesne tespiti gerçekleşiyor mu, görüntüde küçük boyutta olmayan diğer nesnelerin tespiti kaçırılıyor mu?)

Araştırma hedefleri:

- Transfer öğrenme sayesinde nispeten küçük veri seti üzerinde düşük donanım özellikleri ile derin öğrenme ağ eğitimlerinin tamamlanması.
- İnce ayarla görüntü üzerinde yer alan derinde ve küçük nesneleri tespit edebilme.

Hedeflerimizin ölçülebilir olması için kullanılan tüm veri setlerinde test ayrımı yapılmış ve ağların çalışma süreleri not edilmiştir. Test görüntüleri üzerinde doğruluk oranları hesaplanarak başarı ölçümleri gerçekleştirilmiştir. İlk hedef olarak gerekli model ayarlarının yapılması, daha sonra veri setleri üzerinde model eğitimlerinin başarılı bir şekilde gerçekleştiğinin eğitim grafikleri üzerinden yorumlanması ve tespit işlemleri sonucunda doğruluk oranlarının hesaplanması vardır.

1.3. Araştırmanın Önemi

Problem durumunda belirtildiği gibi derin öğrenme çalışmalarının mevcut bazı sınırlılıklarını gidermek bakımından bu çalışma önemlidir. Çalışmada kullanılacak olan transfer öğrenme yöntemi sayesinde bilgisayar donanımını uzun süreler boyunca çalıştırmadan çok da büyük olmayan (ortalama 1.000 adet) görüntü verisi üzerinde derin öğrenme çalışmalarının yapılabilirliğinin test edilmesi, nesne tespiti çalışmalarının daha az zaman ve finansal maliyetle gerçekleştirilmesine katkı sağlayacaktır.

Ek olarak görüntünün derin kısımlarında yer alan küçük nesnelerin tespitinin yapılabilmesi gündelik yaşam kalitemizi artırmasının yanı sıra bilgisayarlı görü teknolojilerinin gelişimi için de önemlidir. Örneğin, can güvenliği için önemli olan trafik işaret levhaları veya yayalar yüksek çözünürlüklü görüntülerde az yer kapladığından kolay ayırt edilemezler. Medikal görüntülemelerde tümör gibi küçük cisimlerin erken tespiti hayati derecede önemlidir. Endüstride makinedeki küçük bir hatanın tespiti koca bir işi baştan yapmanın önüne geçebilmektedir. Havadan görüntü sağlayan araçlarla görüntülemelerde ise neredeyse tüm nesneler küçük kategorisine girmektedir. Kapalı alanlara küçük silah sokmak güvenlik zafiyetine yol açmaktadır. Akıllı araçlar, askeri projeler, akıllı ulaşım gibi güvenliğin öncelendiği alanlarda küçük nesne tespitinin önemi çok büyüktür.

Çalışmada elde edilecek sonuçların bilgisayarlı görü teknolojilerinin gelişmesi ve bu bağlamda ilerleyen dönemlerde donanım uygulanmasına katkı sağlayacağı düşünülmektedir. Bu sayede engeli olan bireylere yardımcı uygulamaların geliştirilmesinden, daha doğru tıbbi teşhislerin konulmasına, ülkenin savunmasında düşman hedeflerinin daha kolay tespit edilmesinden, daha doğru tarımsal analizlere kadar birçok alana katkı sağlanabileceğine inanılmaktadır.

2. BÖLÜM

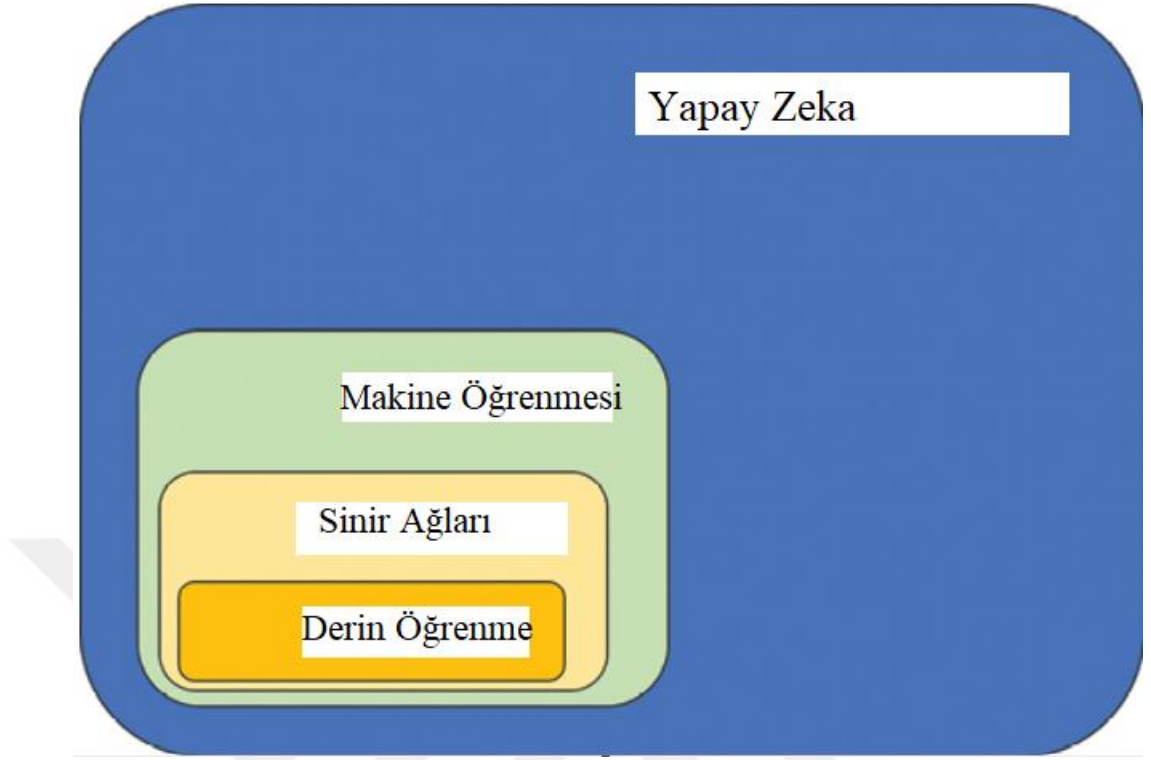
GENEL BİLGİLER ve LİTERATÜR ÇALIŞMASI

2.1. Bilgisayarlı Görü

İnsan görmesine dayanan bilgisayarlı görü teknolojisi, ESA ağları aracılığıyla görüntü verisinden anlam çıkarmaktadır. Sınıflandırma, segmentasyon ve nesne tespiti gibi birçok alanda bu teknoloji kullanılmaktadır. Tespit işleminde, görüntü üzerindeki nesneye sınıflandırmanın yanında konumlandırma da uygulanmaktadır. Bu işlem, sınıflandırılan nesnenin görüntüdeki yerini belirten ve kullanılan ağa göre değişen sınırlayıcı kutu çizme veya nesne içini farklı renklerle doldurarak ayırma olarak gerçekleşmektedir. Tespit işlemi geçmiş yıllarda SIFT, SURF gibi klasik görüntü işleme algoritmalarına dayanırken şimdilerde derin öğrenme tabanlı makine öğrenmesi ile birçok yönden avantaj sağlanmaktadır [16].

2.2. Makine Öğrenme

Adımlarını yazması çok uzun veya tanımlaması çok zor olan görevler için makine öğrenmesi kullanmak idealdir. Örneğin bir topun atılması için kişinin nasıl durması gerektiği, topu kaç şiddetinde fırlatması gerektiği gibi adımları kişiye öğretmek yerine çokça farklı top atma yöntemlerini kişiye göstermeye, onun bu yolla öğrenmesine, makine öğrenmesi yaklaşımıyla eğitim denir [17]. Makine ve derin öğrenme yapay zeka tarafından Şekil 1’de gösterildiği gibi kapsamaktadır fakat her yapay zeka uygulaması bir makine veya derin öğrenme uygulaması olmak zorunda değildir.



Şekil 1. Yapay zeka kapsamı [17]

Örneğin yapay zeka gelişiminde dönüm noktası olarak gösterilen 1997’de satranç şampiyonunu yenen Deep Blue’da oyun kuralları ve yenme stratejileri programlanmıştır. Kısaca kendisine verilen girdi ve çıktı verilerinin arasındaki ilişkiyi öğrenmeye çalışan makine öğrenmesinin derin öğrenmeden temel farkları şunlardır;

- Öğrenmesi için kendisine verilen özelliklerin el ile çıkarılmış olması,
- Yapay sinir ağları kullanarak 3 katmandan oluşması ile daha az veriye ihtiyaç duyması sonucu genelde başarımının daha düşük çıkmasıdır.

Makine öğrenmesi en genel başlık olarak denetimli ve denetimsiz’e ayrılmaktadır. Denetimli makine öğrenmesinde ham verinin eşleştiği etiket gibi yardımcı elemanlara ihtiyaç duyulurken denetimsiz makine öğrenmesinde ağ, ham verilerden tamamen kendi başına öğrenmeye çalışmaktadır.

2.3. Derin Öğrenme

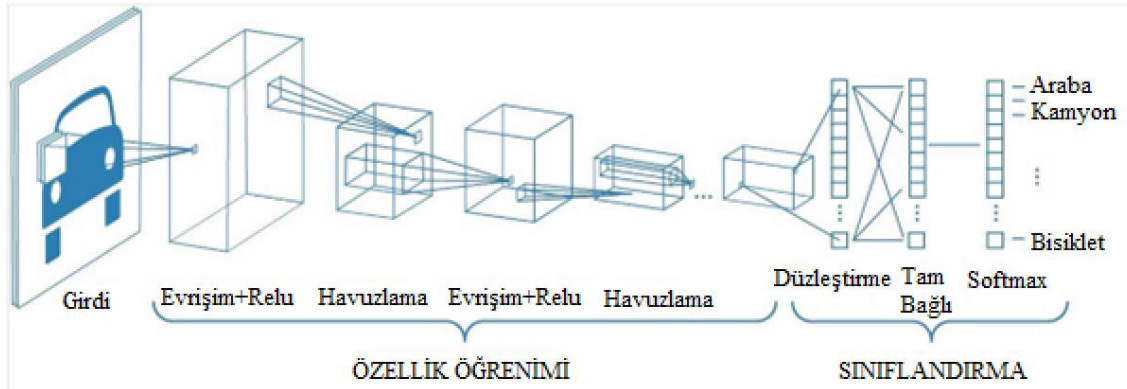
Derin öğrenme, makine öğrenmesine göre başarı ve kullanım kolaylığı gibi çeşitli avantajlara sahiptir. Yapay sinir ağları kullanan makine öğrenmesi girdi, gizli ve çıktı olmak

üzere sadece üç katmana sahipken derin öğrenme, gizli katman sayısını artırarak daha çok veriden daha fazla özellik çıkarımını mümkün kılmaktadır. Katman sayısı arttıkça veriden çıkarılan bilgi miktarı da orantılı olarak artmaktadır. Günümüzde bir derin öğrenme ağının katman sayısı 152'lere kadar ulaşabilmektedir [18]. Derin öğrenme verinin her katman geçişinde iç parametreleri nasıl değiştirmesi gerektiğini yani bilgi tutan ağırlığı nasıl güncellemesi gerektiğini geri yayılım algoritması sayesinde keşfederek kaliteli çıktılar -ağırlıklar- üretmektedir. Bu sebeple derin öğrenme, klasik görüntü işleme göre devrim niteliğindedir. Klasik görüntü işlemede programcının algoritmanın tüm adımlarını belirlemesiyle özellik çıkarımı yapılırken derin öğrenmede bu işlemler girdi ve çıktının ağa verilerek kuralın (özellik) eldesi şeklinde gerçekleşmektedir. Makine öğrenmesinden bir farkı da özellik çıkarımını otomatikleştirmesidir.

ESA tabanlı derin öğrenme ağının evrişim ve havuzlama katmanları aracılığıyla çıkarılan özellikler, ağın sonunda yer alan tam bağlı katman aracılığıyla sınıflandırma işlemine tâbi tutulmaktadır. Sınıflandırılan yani tahmin edilen özellikler ise ağın başında ağa verilen gerçek girdilerle karşılaştırılarak bu girdilere ne kadar yakın tahminler yapıldığının ölçümü yapılmaktadır. Ölçümler neticesinde ise sınıflandırma skor değeri üretilmektedir.

2.4. Evrişimli Sinir Ağları

ESA görüntüden anlam çıkarma işini görüntü üzerinde çok ve çeşitli sayıda filtre kaydırarak yapmaktadır. Filtrelerin çeşitleri görüntüden çıkarılmak istenen kenar, leke, doku özelliklerine göre farklılaşmaktadır. Temelde evrişim, havuzlama ve tam bağlı katmanlardan oluşan ESA'nın unutma, beyazlatma ve düzleştirme katmanları da vardır. Ağda istenilen bir yere belli bir oranda eklenebilen unutma ile ağın ezber yapması engellenirken beyazlatma katmanı ile ağın genelleştirme yeteneği artırılmaktadır. Düzleştirme katmanı ise tam bağlı katmandan önce gelerek iki boyutta olan özellikleri tek boyuta dönüştürmektedir. Ayrıca katmanlarla birlikte isteğe bağlı olarak seçilen birtakım aktivasyon fonksiyonları vardır. Her biri farklı matematiksel işlem yapan aktivasyon fonksiyonlarından softmax, ağın son tam bağlı katmanında yer alarak çoklu sınıflandırma işlemini yapabilmektedir. Şekil 2'de ESA'nın temel bir gösterimi sunulmuştur.



Şekil 2. ESA [19]

2.5. Transfer Öğrenme

Transfer öğrenme en genel başlık olarak bir makine öğrenmesi tekniğidir. Derin öğrenmedeki en popüler yaklaşımlardan biri transfer öğrenmedir [20]. Transfer öğrenme, insan öğrenim aktarımından ilham alınarak geliştirilmiştir. Günlük hayatta karşılaştığımız problemlere istemli veya istemsiz olarak çözümler üretmekteyiz [21]. Örneğin bisiklet sürmeyi bilen biri, araba sürmeyi öğrenirken fazla zorlanmayacak ya da matematiği iyi olan birinin fizik dersi de iyi olacaktır. Yani bir işi iyi bilen biri benzer başka bir işi de yapabilecektir mantığından türeyen transfer öğrenmenin derin öğrenmedeki çalışma prensibi ise belirli bir tip veride çalışan ağ bilgisinin, yine aynı tip veri üzerinde çalışacak başka bir ağa aktarılabilmesi şeklindedir. Transfer öğrenmenin pratikteki uygulaması; ImageNet [22], MS COCO [4], PASCAL VOC [3] gibi büyük veri setlerinde çalışmış ağların çıktı olarak verdiği ağırlıklarda depolanan bilgilerin, probleme özgü olarak ayarlanan ağa aktarılması şeklindedir. Transfer öğrenme, ilgili literatürde [23] “bir problemde öğrenilen özellikleri alarak bu özellikleri yeni ve benzer problemde kullanmak” şeklinde tanımlanırken diğer bir tanım da, “yüksek miktarda görüntü üzerinde eğitilmiş modellerin güçlü özelliklerini transfer ederek eğitimi hızlandıran metot” olarak karşımıza çıkmaktadır [24]. En genel formuyla transfer öğrenme aşağıdaki 4 adımdan oluşmaktadır:

1. Önceden eğitilmiş ağın katman çıktı ağırlıklarını al.
2. Yeni eğitilebilir katmanların eğitilebilmesi için bu ağırlıkları yeni ağa transfer et.
3. Önceki bilgiyi, yeni veri setinde tahminler yapmaya dönüştürmeyi öğrenebilen yeni, eğitilebilir katmanları kullan.

4. Yeni veri setinde yeni katmanları eğit.

Kaynak ve hedef alanlarını içeren transfer öğrenmede, transferin; nereden yapılacağını kaynak, nereye yapılacağını ise hedef belirtmektedir. Kaynak ve hedef alanları, birer derin öğrenme ağını ve veri setini içermektedir. Kaynağın veri seti oldukça büyük yani çok sayıda veriden oluşmakta, hedef veri seti ise kaynak veri setine göre daha az veriden oluşmaktadır. Öğrenmenin aktarımı; kaynak ağıın kaynak veri setinde eğitilmesiyle oluşan ağırlıkların, hedef ağıın hedef veri setindeki eğitiminin başlangıç noktası olarak kullanılması şeklinde gerçekleşmektedir.

Transfer öğrenme kullanımı, yetersiz veri seti problemi etrafında birleşmektedir. İHA simülasyon verisi üzerinde derin öğrenme ağı çalıştırmak isteyen araştırmacılar; yetersiz veriyle karşılaştıklarında transfer öğrenmeyi kullanmışlardır [25]. Diğer bir çalışmadaki araştırmacılar akciğer, karaciğer bölümlene ve nodül sınıflandırmada transfer öğrenmeyi kullanarak bu organların 3 boyutlu görüntü ve etiketini içeren nispeten küçük veri setinden yararlanmışlardır [26]. Başka bir çalışmada yapılan uygulamada az sayıda MRI görüntüsü üzerinde eğittikleri transfer öğrenmeli AlexNet ağından %100 doğruluk elde etmişlerdir [27]. Tüm hasadı bitirebilecek ölçüde zararlı olan bitki hastalığını derin öğrenme ile otomatik tespiti üzerine yapılan çalışmada ImageNet veri seti kaynak olarak kullanılmıştır. Aslında ImageNet'te bitki nesnesi yoktur fakat bitkide de diğer nesnelerde de ortak olan kenar, doku, leke gibi genel özellikleri öğrenerek eğitime başlayan ağı için bitkiyi tanıma işi kolaylaşmıştır. Bu bilgileri öğrenmeden eğitime başlayan ağı göre bu bilgileri öğrenen ağı, daha başarılı olmaktadır [28]. Başka bir çalışmanın yazarları ise teleskop ile görüntülenen uzaydaki galaksi çarpışma görüntülerini toplayarak çarpışmaları transfer öğrenmeli derin öğrenme ağı ile tespit etmişlerdir [29]. Farklı bir çalışmada da bilinen internet ağı atakları ile eğitilmiş derin öğrenme ağıının çıktı ağırlıkları kullanılarak bilinmeyen internet ağı atakları tespit edilmiştir [30]. Tarımda işe yarayacak balya [31], arabalarda hasara yol açacak bozuk yol [32], hastalığa neden olan akciğer nodülü [33], depremin habercisi su yüzeyindeki dalga [34], can ve mal kaybına yol açabilecek yangın [35], pirincin kanseri olarak bilinen pirinçlerin içindeki kir [36] tespiti gibi daha birçok alanda transfer öğrenme kullanılmıştır.

Transfer öğrenmenin özellikle sağlık alanında çok sık kullanım oranına sahip olmasının nedeni hasta mahremiyeti ve idari sebeplerden ötürü veri paylaşımının kısıtlı olmasıdır.

Daha önce de bahsedildiği gibi az veride çalışabilen transfer öğrenme bu duruma çözüm olmaktadır. Bu çalışmalardan birinde yazarlar az sayıda göğüs sitoloji görüntülerinden meme kanserini, transfer öğrenme uygulamasıyla tespit ederek hastalığın erken teşhisinin, ölüm oranlarını düşürebileceğini söylemişlerdir [37]. Başka bir çalışmada MR beyin görüntüleri kullanılarak 3 tip beyin tümörünü sınıflandırmada transfer öğrenmeden yararlanılmıştır [38]. Diğer bir sağlık alanındaki çalışmada ise göğüs tomografi görüntüleri üzerinde transfer öğrenmeli derin ağ çalıştırılarak Covid-19 tespiti yapılmıştır [39]. [40] çalışmasındaki yazarlar bitki türlerini el ile sınıflandırmanın yorucu, hataya açık, zaman alıcı ve uzman desteğine ihtiyaç oluşturduğunu söyleyerek derin öğrenme ile otomatik sınıflandırma metodu önermişlerdir. Bunun için 4 farklı veri setine transfer öğrenme uygulamışlardır. Başka bir çalışmada tarıma zararlı 10 tip bitki hastalığı derin öğrenme ile tanımlanarak uzmanların tanımlama başarıları ve geleneksel ağlarla karşılaştırmalar yapılmıştır. En yüksek başarı derin öğrenme çalışmaları ile elde edilmiştir [41]. Pandemi süresince maske takmanın öneminden yola çıkarak maske tespit sistemi geliştirme adına transfer öğrenme, derin öğrenme ve geleneksel makine öğrenmesi metodlarını birleştirmişlerdir [42]. Farklı bir çalışmada ise günümüz kirliliğine dikkat çekmek için toplam 5.904 görüntüden oluşan organik, inorganik ve medikal atık görüntüsünü derin öğrenme eğitim işlemine alarak atık geri dönüştürmeyi amaçlamışlardır ve transfer öğrenme sayesinde %98 oranında başarıya ulaşmışlardır [43].

2.6. İnce Ayar

İnce ayar çeşitli şekillerde kullanıma sahiptir. Görüntü kırpma, nöron sayısını ayarlama için de kullanılan ince ayar terimi, literatürde genelde katman dondurma şeklinde anılmaktadır. İstenilen ağların dondurularak eğitilmemesi dolayısıyla ağırlık transferiyle gerçekleştirilen transfer öğrenmenin katman transferiyle gerçekleştirilmesine de ince ayar denmektedir. Bu çalışmada ise ince ayarın kullanımı, transfer öğrenmeden bağımsız olarak hiper parametre optimizasyonu şeklinde uygulanmıştır.

Derin öğrenme ağındaki katmanları dondurma mantığına dayanan katman tabanlı ince ayar ile eğitim süresi kısalabilmekte ve sadece istenilen katmanların özellikleri ağda tutulabilmektedir. 2.5 başlığında görüldüğü gibi veri seti azlığından dolayı transfer öğrenmenin kullanıldığı çalışma sayısı çok olmasına rağmen az sayıdaki katman tabanlı ince ayarlı transfer öğrenme çalışmaları şu şekildedir; [44] çalışmasındaki yazarlar

transfer öğrenmeyi kullanmak için 10 nesneli CIFAR-10 ve 1.000 nesneli ImageNet veri setlerinin birleşimini, transfer öğrenmenin kaynağı olarak seçerek hedef veri seti olan küçük boyutlu termogram görüntülerinin eğitiminde kullanmışlardır. Katman tabanlı ince ayar işlemi ile ağın sadece son katmanını eğitime açmışlar, bu sayede özellik çıkarım aşamasını geçmişlerdir. Başka bir ince ayarlı transfer öğrenme çalışmasında ise görüntülerden elma, dal ve kamyon tespit ederek dalları sallayan dolayısıyla elmaları kamyonlara düşüren sistemi geliştirerek elma toplamayı büyük ölçüde otomatize etmişlerdir. Bu çalışma için önceden büyük veri setlerinde eğitilmiş AlexNet, VGG16 ve VGG19 ağlarında katman tabanlı ince ayar işlemi gerçekleştirmişlerdir [45]. Bir başka çalışmada kameranın olduğu ortamlardaki insanları derin öğrenme ile belirlemek için çok sayıda insan görüntüsüne ihtiyaç duyulduğundan bahsederek bu ihtiyacı, transfer öğrenmeyle aşmışlar ve eğitim süresini düşürmek için iki aşamalı katman tabanlı ince ayar uygulamışlardır [46]. Tümör tespitini; 2.000 görüntülü 4 beyin MRI veri seti üzerinde AlexNet, GoogLeNet ve VGGNet ağlarıyla transfer öğrenmesiz ve katman tabanlı ince ayarlı transfer öğrenmeyle eğiterek gerçekleştirmişlerdir. Daha doğru sonuçları katman tabanlı ince ayarlı transfer öğrenmeyle almışlardır. İlk aşamada ince ayarlı ağları eğiterek özellik çıkarımı yapmışlar, ikinci aşamada ise sınıflandırma yapacak ağa, çıkarılan bu özellikleri girdileyerek yüksek başarı elde etmişlerdir [47]. Aslında transfer öğrenme ile ince ayar işlemlerini birleştirmişlerdir. Uydu görüntüleri kullanılarak bina hasar tespiti yapılan çalışmada ise kullanılan derin öğrenme ağının sonuna geleneksel makine öğrenme modeli ekleyip oluşan yeni ağa katman tabanlı ince ayar uygulayarak neredeyse tüm test görüntülerinde doğru tahminler yapacak %99.29 f1 skoru elde etmişlerdir [48]. Diğer bir çalışmadaki yazarlar, Google Earth web sitesinin sağlamış olduğu kullanımı herkese açık uydu görüntülerinde yer alan uçak görüntülerini, katman tabanlı ince ayarlı transfer öğrenme uygulanmış ağın eğitiminde kullanarak tespitte yüksek kesinlik skoru elde etmişlerdir [49].

2.7. Konu ile İlgili Yapılan Çalışmalar

Bu çalışmanın birinci amacı; küçük veri setiyle, düşük donanım özelliklerinde derin öğrenme ağ eğitimleri gerçekleştirmek ve yüksek başarımlar elde etmektir. İkinci amacı da herhangi bir veri setindeki görüntünün üzerinde yer alan küçük nesneleri derin öğrenme ağına tespit ettirmektir. Küçük nesne tespiti etmek için ağ mimarisinde değişiklik

yapmayarak hesaplamaya ek yük getirmeme ve kullanılan veri setlerinde deęiřtirme yapmadan çözüm yolu geliřtirmek bu çalışmanın amacıdır.

Görüntü algılama algoritmaları YOLO ve Faster R-CNN'in farklı hava koşullarındaki nesne tespitleri sonucu oluşan doğruluk oranlarının karşılařtırması yapılmıřtır [50]. Kullanılan çok sayıda veri üzerinde YOLO aęı daha başarılı olmuřtur. Karanlık, parlama gibi durumlar farklı hava koşullarında çokça olduęundan nesne tespiti zorlayıcı olmaktadır. Arka planın karıřık olduęu veri setleri bu tez çalışması kapsamında seğıilmemiřtir çünkü derin öğrenme aęlarının bilgisayarlı görüyle birleřtięi kısım araştırılmamıřtır.

Bilgisayarlı nesne tespitinin, insandan ilham alınarak geliřtięini vurgulayarak bir iři yapmadan önce o iře giden yoldaki nesnelerin tespit edilerek iřin yapılması gibi bilgisayar için de eğitim ařamasından sonra tespit iřleminin yapıldıęı ifade edilmiřtir [51]. Çalışmanın yazarları piksel başına derinlik bilgisi saęlayan görüntüler üzerinde Faster R-CNN aęı çalıştırmıřlardır. Eğitim ařamasındaki aęın giriřte kabul ettięi görüntü çözünürlük deęerini 224x224'e, tespit ařamasındaki aęın IoU deęerini ise 0.3'e ayarlamıřlardır. Bu tez çalışmasında ise kullanılan Faster R-CNN aę girdi çözünürlüęü deęeri varsayılan deęer olarak 704x704'te tutulmuř, ayrıca tespit ařamasında daha fazla nesne tespit edilmeye çalışılarak IoU deęeri 0.2'ye ayarlanmıřtır.

Yazarlar, yař ve cinsiyet verisini son tam baęlı katmanın nöron sayısına ekleyerek sınıflandırmaya ek tahmin iřlemleri gerekleřtirmiřlerdir [52]. Tez kapsamında kullanılan sayısal veri, sınırlayıcı kutu çizdirmede kullanılan parametrelerdir ve kullanımı için ek bir mimari deęiřiklięine gerek duyulmamaktadır. Fakat çalışmada transfer öğrenme kullanıldıęından son tam baęlı katmanların sınıflandırıcı nöron sayıları; satran için 6'ya, röntgen ve kask için 2'ye ayarlanmıřtır.

Derin öğrenme eğitimleri GPU'da yapıldıęında daha hızlı ve verimli olmaktadır. Buradan yola çıkan yazarlar gömülü GPU kullandıkları çalışmada az kaynak harcayarak gerek zamanda çalıştırmalar gerekleřtirmiřlerdir [53]. YOLO v4 algoritmasının girdi ızgara paralama sayılarını deęiřtirerek tespit iřlemleri yapmıřlardır. En yüksek FPS'yi 416x416'dan, en yüksek doğruluęu ise deęiřtirdikleri YOLO v4 mimarisine uyguladıkları 832x832 ızgara paralamadan almıřlardır. Bu tez çalışmasında ise hızdan çok doğruluęa odaklanıldıęından gerek zamanda çalışan cihazlar kullanılmamıřtır. Ayrıca bu

tür küçük cihazlar daha az kaynak harcadığından kararlılığı masaüstü bilgisayarlar kadar fazla değildir.

Bir başka çalışmada sera alanları tespit edilerek ekonomiye katkı sağlayan unsurlara değinilmiştir [54]. Sera tespiti için kullandıkları görüntüler; havadan çekilmiş kızılötesi hiper spektral görüntülerdir. Kızılötesi hiper spektral görüntü, gündelik hayatta sürekli karşılaştığımız RGB görüntüler gibi kolay elde edilememektedir. Sera yerleri plastikten olduğundan yansması çok olmakta ve tespiti zorlaşmaktadır. Görüntüdeki sera alanlarına karartma ön işlemi uygulanarak zorluk durumu aşılabılır.

[55] Bu çalışmada da bir üst paragrafta bahsedilen çalışma gibi hiper spektral görüntüler kullanılmıştır. Fakat bu sefer hiper spektral görüntülerin içeriği, suyun altıdır. Çalışmanın amacı; arama kurtarma ekiplerine, yansıma yapan su kolonunu tespit ederek yardımcı olmaktır. Hiper spektral görüntünün, multispektral (görünür banttaki) görüntüye göre daha detaylı özellikler içerdiği vurgulanmıştır. Fakat hiper spektral görüntü internetten elde edilebilecek, kolay toplanabilen bir görüntü türü değildir.

Tümyönlü kameralar ile nesne tespiti için doğrudan bir yaklaşım [56] adlı çalışmada tüm yönlü kamera; sağ, sol, ön olmak üzere 3 yönü de görerek geniş görüş açısına sahiptir. Yazarlar, ilk önce kayan pencere yaklaşımı HOG ile tüm yönlü olmayan normal kameralarda filtre kaydırarak özellik çıkarmışlar daha sonra HOG tekniğini tüm yönlü kameraya uygulamaya çalışmışlardır. Derin öğrenme kullanmadan gerçekleştirilen SIFT, SURF gibi görüntü tanımlayıcıları ve özellik vektörleri, kayan pencere, Bayesçi, DVM gibi makine öğrenmesi sınıflandırıcılarının etkinliğinden bahsetmişlerdir. Çok yönlü kameralara, standart kameralar için uygulanan HOG gibi algoritmaların uygulanamayacağını belirterek HOG + SVM yaklaşımında değiştirmeler yaparak yeni bir algoritma geliştirmişlerdir. SIFT ve SURF nesne tespitinde derin öğrenmesiz çözüm olarak kullanılmaktadır. Bu gibi çözümler özellik çıkarmada derin öğrenme ağları kadar başarı getirmemektedir.

Derin Evrişimli Sinir Ağları Tabanlı Kısıtsız Kulak Tanıma [24] adlı tez çalışmasının başlığındaki kısıtsız ifadesi, kulak tespit probleminin tek görüntü üzerinden çözülmediğini vurgulamak içindir. Gürültü ekleme, rasgele kırpma, döndürme, çevirme bu çalışmada denenen veri çoklama metotlarıdır. Döndürülmüş birçok kulak görüntüsünü elde ettikten sonra bu görüntüleri aynı yöne getirmek için diğer çalışmalardan farklı olarak

veri hizalama uygulanmıştır. Görüntülere ayrıca iki aşamalı kırpma ince ayar işlemi uygulanmıştır. İlk olarak yüz veri setinden yüz çevresi kırılmış, daha sonra çıkarılan bu yüzlerdeki kulaklar kırılarak derin öğrenme ağına verilmiştir.

Önceki çalışmaların öğrenme oranı ve yığın boyutu hiper parametrelerinin değişikliği ile yapılan ince ayar işlemini araştırdığını fakat momentumun etkinliğini araştırmadığını öne sürerek yaptıkları Rethinking the Hyperparameters for Fine-tuning [57] adlı çalışmada hiper parametrelerin sadece veri setine bağlı olmayarak kaynak ve hedefteki ağların arasındaki ilişkiye de bağlı olduğunu ortaya koymuşlardır. Ayrıca ağırlık kayıplı olmayan momentumun, ağırlık kayıplı momentuma göre test hatası daha yüksek çıkmıştır. Hata değeri arttıkça ağırlık doğruluk değeri düşmektedir. Buradan yola çıkarak ağırlık kaybının, ağırlık doğruluğunu iyileştiren bir etmen olduğu yorumu yapılabilmektedir. Hiper parametrelerden momentum devre dışı bırakıldığında ise öğrenme oranı hata değerinin arttığı gözlenmiştir. Buna göre momentum değerinin de aynı ağırlık kaybı gibi ağırlık doğruluğunu artıran bir etmen olduğu söylenebilmektedir.

Augmentation for small object detection [58] çalışmasında yazarlar, küçük nesneleri içeren görüntüleri çoğaltmışlardır. Bu işlem veri setinde küçük nesne içeren görüntülerin daha çok olmasıyla dengesizliğe neden olmuştur. Ağ, bu veri setinden küçük nesne özelliklerini daha iyi öğrenip tespit ederken büyük ve orta boyutlu nesneleri daha zor tespit etmiştir. Yani ağırlık genelleme yeteneğinde düşüş meydana gelmiştir. Buna kanıt olarak COCO veri setini tespit etme yarışında bu çalışmanın kurduğu takımın, başarıyı ölçen AP değerinde; küçük nesne tespit başarısı, büyük nesne tespit başarısından neredeyse yarı yarıya fazla çıkmıştır. Bu çalışmadaki gibi küçük nesne tespit ederken genel doğruluğumuzun düşmesini istemiyoruz. Bu yüzden çalışmada veri setinden bağımsız bir ince ayar yöntemini önermenin yanında başarıyı çoğu zaman artıran transfer öğrenme yöntemi kullanılmaktadır.

Small-Object Detection in Remote Sensing Images with End-to-End Edge-Enhanced GAN and Object Detector Network [59] adlı çalışmada Dünya üzerinde yer alan yağ ve gaz depolama tanklarını tespit için uydu görüntülerinden faydalanmışlardır. Uydu görüntülerinin; düşük çözünürlüklü ve geometrik bozulmalarının olması ve uydu sensör arızası, atmosferik durumlar gibi gürültülü olduğunu bu yüzden uydunun sağladığı görüntülerdeki küçük nesnelerin derin öğrenme ağı ile tespitinde zorlanıldığı yazılmıştır.

Bu duruma çözüm olarak düşük çözünürlüklü girdi görüntüsünü ilk önce Generator ağından geçirerek orta seviyede çözünürlüklü görüntü oluşturulmuş daha sonra kenar geliştirme ağından geçirdikleri orta çözünürlüklü görüntüyü, yüksek çözünürlüklü görüntüye çevirmişlerdir. Oluşan bu görüntüleri son olarak tek aşamalı ağ olan SSD'ye ve çift aşamalı ağ olan Faster R-CNN'e vermişlerdir. Fakat veri setinde validasyona yer ayırmayarak model ayarlarının doğru yapıp yapılmadığını kontrol etmemişlerdir. Kullandıkları kenar geliştirme metodu ile bütünleşik Faster R-CNN ağından en yüksek skoru %84.9 AP ile bulmuşlardır.

[60] Real-time gun detection in CCTV: An open problem adlı çalışmada bilindik veri setlerinden COCO, ImageNet ve Open Images'da küçük nesnelerin, orta boyuttaki nesnelerden yaklaşık olarak %38 daha az olduğu söylenmiş, bu yüzden küçük nesneleri içeren gerçek dünya görüntüsü bulamadıklarını öne sürerek sentetik görüntü oluşturmak için oyun motorlarını kullanmışlardır. Bu yolla görüntülere küçük boyutlu nesne ve karakterler yerleştirmişlerdir. Sonuçta bu görüntüler üzerinde tespit işlemleri sağlamışlardır. Fakat sentetik veri, dünya verisine benzemediği için sentetik görüntüler üzerinde eğitilen ağın daha önce hiç görmediği gerçek dünya verisi üzerindeki tahmin başarısı bir hayli düşük olacaktır.

[61] Başka bir çalışmada Romero ve Salamea, iki aşamalı ince ayar denebilecek kırpma işlemlerini yaparak görüntüleri iki farklı YOLO ağına vermişlerdir. İlk aşamada eğitim sonucu tespit edilen insan nesnesini görüntüden çıkarıp oluşan bu yeni görüntüleri farklı bir YOLO ağına vererek insan üzerindeki silah nesnesini tespit etmişlerdir. Aslında her iki YOLO ağı da bir sınıf tespit etmektedir. Çoklu sınıflandırma problemlerinde kullanılması daha uygun olan YOLO ağına bir nesne tespit ettirerek gereksiz kaynak harcamasına neden olunmuştur. Ayrıca elde ettikleri %86 doğruluk skoru, derin öğrenmede ortalama istenen %90 doğruluk skoru barajını geçememiştir.

[36] Küçük nesneleri tespit için iki aşama önermişlerdir; ilk aşamada Gauss Karıştırma Modeli sayesinde odaklanılmış bölgeler olan nesne kümeleri üretmişler daha sonra üretilen bu kümeler üzerinde nesneleri tespit edebilen tespitleyici ağ çalıştırmışlardır. Gauss Karıştırma Modeli, tahmin edilen bölgelerde ekstra hesaplama yapmadan ölçek normalizasyonu sağlamaktadır. Bu sayede tahmin edilen bölgeler sabit boyuta getirilmektedir. YOLO ve Faster R-CNN ağları sabit boyuta getirme işlemini boyun bölgele-

rinde kullandıkları SPP ağı sayesinde yapmaktadır. Gauss Karıştırma Modelinin dezavantajları ise aşağıdaki gibi sıralanabilir; kategorik özellikler arasında iş birliği sağlamada zorluklar, özellikler için normal dağılımı kabul etme, küme şekli için bazı varsayımlar yapması, her bir küme için yeterli sayıda veriye ihtiyaç duyması, küme sayısının belirli olmasına ihtiyaç duymasıdır [62]. Bu yönlerden Gauss Karıştırma Modeli kullanmak ağda olumsuz etki oluşturmaktadır.

[63] Feature-Fused SSD: Fast Detection for Small Objects adlı çalışmada, çoğu çalışmanın, görüntüdeki küçük nesneleri tespit için ağıın çalışma hızından feragat ettiği söylenmiştir. Bu çalışmada ise tek aşamalı ağ olan SSD ile hıza odaklanılmıştır. Çalışmada çok seviyeli özellik birleştirme metodu önerilmiştir. Çok seviyeli özellik birleştirme metodunu SSD'ye ekleyip içeriksel bilgiyi artırarak derin öğrenme ağlarında normalde yaşanan hız artarken doğruluğun düşmesini yaşamamışlardır. Önerdikleri SSD ile orijinal SSD'ye göre %1.7'lik doğruluk artışı elde etmişlerdir.

Literatürde yapılan çalışmaların “ince ayarı” ağıın son tam bağı katman nöron sayısını değiştirme, katman dondurma, hiper parametre ayarı, özellik birleştirme, görüntü kırma gibi çok farklı şekillerde kullandığı özelliklerle küçük nesne tespiti için genelde gerçek veri yerine sentetik veri kullanma, küçük nesneleri içeren görüntüleri çoklama veya mimari değişikliğine gidildiği söylenebilir. Mimari değişikliği, ağıın hesaplama yükünü artırmakta, sadece küçük nesne içeren görüntüleri çoklama ise veri setinde dengesizliğe dolayısıyla ağıın genelleme yeteneğinde düşüşe neden olmaktadır. Bu sebeplerden ötürü veri setine bağı olmayacak şekilde küçük nesne tespiti bu çalışma ile önerilmiştir.

3. BÖLÜM

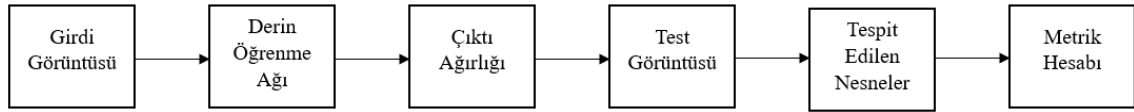
YÖNTEM VE MATERYAL

3.1. Yöntem

Nesne tespiti doğruluğu; doğru mimari seçimine, kullanılan tekniğe ve veri setine bağlıdır. Genel olarak derin olmayan bir mimari, az katmanlı olduğundan görüntülerdeki nesnelerin tüm özelliklerini çıkarmada yetersiz kalırken mimariye uygun olarak seçilemeyen hiper parametreler veya veri seti büyüklüğü sonucunda başarılı sonuçların alınması zorlaşmaktadır.

İnsan görmesinden ilham alınarak geliştirilen bilgisayarla görmede görüntü tanıma uygulamaları yer almaktadır. Bilgisayarlı görü görüntüdeki nesnelerin kategorisini bulup listeleyen sınıflandırma ve sınıflandırdığı nesnelere konumlandırma da uygulayan tespit olarak ikiye ayrılmaktadır. Her iki uygulamada da görüntü üzerinde filtreler kaydırılarak (yani matrisler üzerinde işlemler yapılarak) görüntüden anlamlar çıkarılmaktadır.

Bu uygulamalarda geçerli olan eğitim aşamasından sonra üretilen çıktı ağırlıkları ile derin öğrenme ağına daha önce eğitim aşamasında görmediği etiketsiz (tespit işlemi için nesnelerin sınıfı ve konumu belirsiz) görüntüler üzerinde tahminler yapması sağlanmaktadır. Tahmin işlemlerinin tüm test görüntülerinde tamamlanmasıyla nesnelerin görüntü üzerindeki tespitleri gerçekleşmektedir. Görüntülerde tespit edilen nesnelerin matematiksel formüller ile sayılması sonucu oluşan metrik değerlerinin hesaplanmasıyla da çalışmada çalıştırılan ağlar arası başarı karşılaştırmalarının yapılması mümkün olmaktadır. Bu çalışmada görüntüde filtre kaydırıcı ESA tabanlı derin öğrenme ağları kullanılarak bir bilgisayarlı görü teknolojisi olan nesne tespit işlemleri gerçekleştirilmiştir. Şekil 3, bu işlemleri özetler niteliktedir.



Şekil 3. ESA Tabanlı Derin Öğrenme Ağı Çalışma Yapısı

Çalışmada kullanılan görüntüde nesne algılamaya izin veren ESA tabanlı derin öğrenme ağları; tek aşamalı YOLO v4, YOLO v5 ve çift aşamalı Faster R-CNN'dir. Tek aşamalı YOLO v4 ve v5 ağları, tespit işleminin basamaklarını oluşturan sınıflandırma ve konumlandırmayı tek seferde yaparken çift aşamalı Faster R-CNN ağı, bu işlemleri iki aşamada yapmaktadır. Her iki tip ağ da görüntünün tamamına bakmaktadır ve çoklu nesne sınıflandırmadaki başarıları ile bilinmektedir.

Ayrıca derin öğrenme ağlarını az bir kaynak harcayarak çalıştırmaya imkân tanıyan transfer öğrenme ve küçük nesneleri tanımaya yardımcı ince ayar teknikleri, çalışma kapsamında kullanılan ağlara denenmiştir. Fazla parametrelili (katmanların içeriğinde farklı sayılarda ve çeşitlerde filtre olan), çok katmanlı büyük çaptaki derin öğrenme ağları, yüksek başarımlar için normalde büyük, etiketli veri setlerine ihtiyaç duymaktadır. Fakat büyük veri setine erişim her zaman mümkün olmamakta, olsa bile büyük veri setindeki görüntülere karşılık etiketler çoğu zaman oluşturulamamaktadır. Çünkü görüntü verisi her ne kadar web, mobil tabanlı uygulamalarda veya paylaşım için kullanılsa da etiket verisi, gündelik hayatta işe yarar bir unsur değildir. Gene de görüntüler ile eşleşen etiketleri oluşturmuş ve oluşturmaya devam eden ekipler vardır [3] [4] [22]. Her derin öğrenme çalıştırması için ilgili görüntüye ait etiketi oluşturmamak veya büyük veri setleri için oluşturulanları ilgili görüntü veri setini kullanarak kaynakları fazla harcamamak için transfer öğrenme tekniği geliştirilmiştir. Bu teknik sayesinde önceden toplanmış veri setinin ve veri setiyle ilişkili etiketlerinin istenilen bir ağda eğitilmiş çıktısı, çok miktarda olmayan farklı ve kendi etiketli veri setimize (probleme özgü) uygulanabilmesi sağlanmaktadır. Tez kapsamında transfer öğrenme, ağırlık optimizasyonu olarak önceden büyük veri setlerinde eğitilmiş ağı ağırlık çıktıları, problemimizdeki ağların eğitimini başlatan nokta olarak; mimari değişiminde ise ağı son katmanının nöron sayısının değiştirilmesi şeklinde uygulanmıştır.

Uygulaması tüm ağlara denenen fakat sadece YOLO v4 için sonuç aldığımız transfer öğrenmeli ince ayar işlemi hem transfer öğrenmeyi hem de ince ayarı bünyesinde barındırmaktadır. Yani bu işlem, iki tekniğin aynı anda tek ağa uygulanması şeklindedir. Bu tekniklerin ayrı ayrı denendiği ağların genel doğruluğunda ise transfer öğrenme uygulanan ağ, sadece ince ayar uygulanan ağa göre daha fazla başarı getirmiştir. Zaten ince ayar tekniği ile amaçlanan, ağ doğruluğunu artırmaktan çok görüntüdeki küçük nesneleri tespit etmektir. Derin öğrenme ağ eğitiminden önce hiper parametre değişikliği ile gerçekleştirilen ince ayar işlemi sayesinde derindeki küçük nesnelerin tespiti kolaylaşmıştır.

3.1.1. Transfer Öğrenme Yönteminin Kullanılması

Transfer öğrenmeye verebileceğimiz başka isimler; öğrenmeyi öğrenme, hayat boyu öğrenme ve ikincil öğrenmedir. Örnek olarak bir bebek ilk önce gördüğü yüzleri tanımakta daha sonra bu bilgiyi kullanarak çevresindeki nesneleri tanımaktadır. Önceden eğitilmiş ağ ağırlıklarını kullanan ağ, çoğu nesne özelliklerini öğrenerek eğitime başlar. Pedagojik perspektiften bir örnek de bir işin yaptıkça geliştiğidir, derin öğrenmede önceden eğitilen ağın çıktılarının başka ağları eğitmede kullanıldıkça ağ doğruluğunun artması bu duruma örnek olarak verilebilir.

E. L. Thorndike'nin "Benzer veya ayırt edici araçlar içeren durumlar, daha kolay öğrenilir" ve Charles Judd'un "'A' görevinde öğrenilen bilginin genel hatları 'B' bilgisinin tamamına veya bir kısmına aktarılabilir." sözlerinden yola çıkarak hayatta karşılaşılan 3 tip transfer öğrenme kullanımını aşağıdaki gibi sıralayabiliriz:

- Pozitif transfer: Bir durumda öğrenilen bilginin, diğer öğrenme durumuna yaramasıdır. Viyola çalma becerisinin piyano çalmayı öğrenmeye, scooter sürme bilgisinin, motor sürmeye, matematik bilgisinin, fizik öğrenmeye yaraması bu tip transfere örnek olarak verilebilir.
- Negatif transfer: Bir işi öğrenme, diğer işi öğrenmede zorluk çıkartıyorsa negatif transfer gerçekleşiyor demektir. Sondan eklemeli bir dil bilen birinin, baştan eklemeli bir dili öğrenmede, direksiyonu solda olan aracı sürmeyi bilen birinin, sağ direksiyonlu aracı sürmede zorlanmasıdır.

- Nötr transfer: Ne pozitif ne de negatif yönlü transferdir. Yüzebilme becerisinin motor tamir etmeye yaramaması gibi bir alanda öğrenilen bilginin başka bir alana katkısının olmamasıdır.

İnsan öğrenmesinden ilham alınarak geliştirilen transfer öğrenme yaklaşımında kaynak ve hedef alanları bulunmaktadır. Her iki alan da birbirinden bağımsız olarak birer derin öğrenme ağını ve veri setini ifade etmektedir. İki alanın tek ortak noktası, alanlardaki ağların aynı tip veri üzerinde çalışmasıdır. Kaynak veri setinin, hedef veri setinden; büyük (daha fazla sayıda görüntü içerdiği) olduğu durumlarda transfer öğrenmenin uygulanması mantıklı (çok bilginin az bilgi alanına transferi) olurken kaynak veri setinin hedef veri setinden az olduğu durumlarda transfer öğrenmeyi kullanmak mantıksızdır. Kaynak ağırlığı bir kere hedef ağda kullanıldığında, artık hedef de bir kaynak alanıdır. Yani bu durum gerçekleştiğinde hedef alanı, başka hedef alanlarına kaynaklık edebilir.

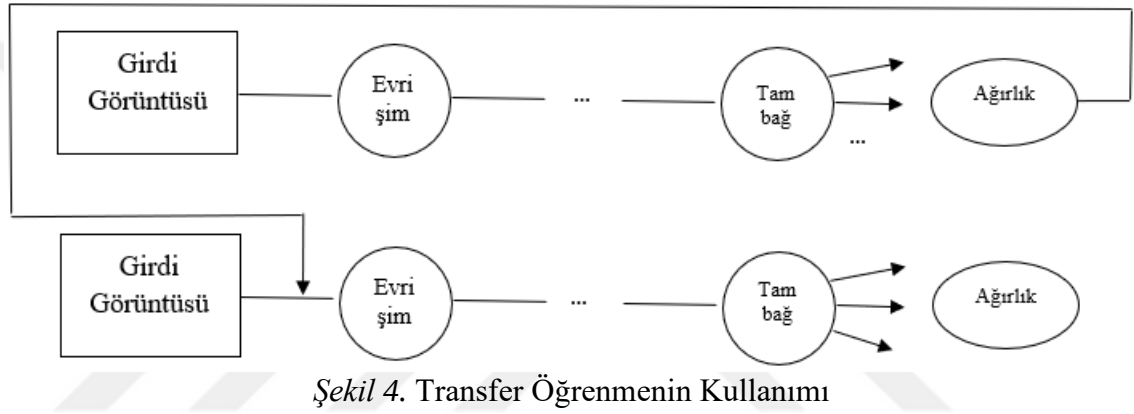
Transfer öğrenmedeki amaç, büyük veri setinde çalışarak bu veri setini oluşturan görüntü özelliklerini, ağırlık çıkartısına kaydettikten sonra bu çıktıyı, küçük veri seti üzerinde çalışacak ağı eğitiminin başlangıç ağırlığı olarak kullanmadır. Yani büyük veri setindeki nesnelerin özelliklerini, hedef ağı öğrenmesinde kullanmaktır. Bu sayede ağırlıklardaki nesne görüntü özellikleri, bilgileri; bu bilgileri bilmeyen ağa aktarılmış olmaktadır. Diğer biri ifadeyle öğrenilmiş bilgiler, başka bir ağı öğrenmesinde kullanılabilir.

Kaynak veri setinde yer alan nesnelerin özellik bilgileri, kaynak ağı son tam bağı katmanından önce gelen evrişim ve havuzlama katmanları aracılığıyla çıkarılmıştır. Hedef ağa transfer edilen bu özellikler hedef ağı evrişim ve havuzlama katmanlarında işlem görmeyerek geçerken hedef veri setindeki görüntü özellikleri bu katmanlarda çıkarılmaktadır. Yani sonuçta katmanlardan; direk ve çıkarıldıkça geçen bu özelliklerin birleşimiyle güçlü özellikler ağda tutulmaktadır. Bu sayede evrişim ve havuzlama katmanlarından sonra gelen son tam bağı katman ile yüksek skorlu tahminler gerçekleştirilmektedir.

Özet olarak kaynaktan transfer edilen bilgileri öğrenmiş olarak eğitime başlayan hedef ağ, öğrendiği yeni bilgileri, transfer edilen bilgilerin üzerine inşa ederek eğitimini tamamlar. Eğitimin tamamlanması esnasında, kaynak ağı yaptığı gibi bağımsız olarak fakat kaynak bilgilerini de içeren ağırlık çıkartısını üretir. Bu ağırlık artık başka hedeflere

kaynak oluşturabilir çünkü hem önceki kaynağın bilgisi hem de kendi veri setindeki görüntülerin bilgisi, oluşturduğu ağırlık çıktısında yer almaktadır.

ESA tabanlı derin öğrenme ağı tarafından kullanılan veri setinden sınıflandırılacak nesne sayısı, büyük etiketli veri setlerini oluşturan gruplar tarafından 80 veya 1.000 adet olarak belirlenmişse kendi problemimize has olarak bu sayıyı ağ mimarisinde değiştirebiliriz. Başka bir ifadeyle Mimarinin sınıflandırıcı katmanının/son tam bağlı katmanının düğüm (nöron) sayısını problemdeki sınıflandırılacak nesne sayısına ayarlamalıyız.



Şekil 4. Transfer Öğrenmenin Kullanımı

Şekil 4’te transfer öğrenmenin ESA mimarisine uygulanışı verilmiştir. Şekil 4’ün üst satırındaki ağ, büyük veri setinde öğrenerek büyük veri setinden sınıflandırılacak nesne sayısınınca tam bağlı katmanında nöron sayısı olan ağıdır. Şeklin alt satırında ise örnek olarak 3 nesne sınıflandırmaya çalışılan bir ağ verilmiştir. Şeklin üst kısmındaki ağ eğitimi tamamlayıp çıktı ağırlığını oluşturduktan sonra bu ağırlığı, şeklin alt satırındaki ağın eğitime başlamasında kullanmasına transfer etmektedir. Altındaki ağ daha sonra kendi girdi görüntülerinden eğitim boyunca öğrendiği bilgiler ile transfer edilen ağırlıklardaki bilgiyi birleştirerek son tam bağlı katmanında bu bilgileri sınıflandırıp kendi ağırlık çıktısını üretmektedir.

3.1.2. İnce Ayar Yönteminin Kullanılması

Çoğu çalışmada görüntüde 25 pikselden; az yer kaplayan nesneler küçük, çok yer kaplayanlar ise normal boyutlu olarak tanımlanmaktadır [11]. Küçük nesne tespitinde yaşanan zorluklar; bilgisayarla görmenin yaşadığı zayıf ışık, parlama, oklüzyon, karanlık gibi problemlerin dışında görüntüde az yer kaplayarak eksik görüntü bilgisine sahip

olması ve bu yüzden arka plan veya benzer sınıflardan kolaylıkla ayırt edilememesi şeklinde gerçekleşmektedir. Bu nesneleri görüntülerde etiketlemek ayrıca kullanıcıya zorluk yaratmaktadır. Veriler etiketlense bile varsayılan ağ ayarları ve mimari gereği eğitim aşamasında bu nesnelerin gözden kaçması, öğrenilememesi muhtemeldir.

Bir derin öğrenme modelinin gelişimini sağlayan unsurlar aşağıdaki gibidir;

- Daha fazla veri toplama: Çalışma kapsamında kullanılan satranç taşları görüntüleri birçok kaynaktan dengeli taş dağılımı gözetilerek mümkün olduğunca fazla sayıda toplanmıştır. Kullanılan diğer veri setleri olan kask ve x-ray görüntüleri ise arama motorlarından bulunamayacağından Kaggle web sitesinden [64] hazır olarak elde edilmiştir.
- Görüntü ön işleme ve çoklama: Seçilen başarılı nesne tespitleyici mimarileri, bu işlemleri barındırdıkları algoritmalar aracılığıyla otomatik olarak yapmaktadır. Doygunluk, renk ve parlaklık ayarı gibi fotometrik ön işlemler ağ girişinde, görüntüyü rasgele döndürme gibi geometrik sanal veri çoklama ise ağı eğitimi esnasında gerçekleşmektedir.
- Görüntü girdi çözünürlük boyutunu ayarlama: Girdi görüntülerinin çözünürlük değerleri kullanılan mimariye göre ayarlanmaktadır. Yani 250x250 boyutlu bir görüntü, girişinde 500x500'lük görüntüyü kabul eden bir ağa gireceği zaman görüntünün içeriği değiştirilmeden eksik 250x250'lik alan, anlam ifade etmeyen siyah renkle doldurulmakta, bu durumun tersi olarak da 500x500'lük bir görüntü 250x250'lik görüntüleri kabul eden bir ağa gireceği zaman merkezden uzak olan 250x250'lik fazla kısım kırılarak atılmaktadır.
- Model boyutu ve mimariyi seçme: Çoklu nesneleri tespit etme başarılarıyla bilinen, görüntünün tümüne bakan, çift ve tek aşamalı başarılı mimariler seçilmiştir. Her iki ağ da fazla parametrelili olarak büyük boyutludur ve bu yüzden ağlar, bolca veriye ihtiyaç duymaktadır. Transfer öğrenme tekniği ağlara uygulanarak büyük veri setlerine ihtiyaç duyulmadan küçük veri setleri üzerinde çalıştırmalar gerçekleştirilmiştir.
- Hiper-parametre optimizasyonu: Görüntüde yer alan küçük nesnelerin tespitine çözüm olmuştur. Faster R-CNN için ağ çözünürlük değeri ve YOLO v5 için ağ kanal derinliği değişimi başarısız, YOLO v4 için ızgara sayı artırımı başarılı olmuştur.

İnce ayar işleminin uygulama aşaması, transfer öğrenmeye benzer şekilde ağ eğitiminin öncesidir. Mimaride ufak bir değişiklik yapılarak eğitim ve dolayısıyla tespit aşamalarında görüntüdeki küçük nesnelerin ağ tarafından görülmesi amaçlanmaktadır. Uygulanan ince ayar işlemi, çalışmada kullanılan farklı ağların farklı hiper parametrelerinin değiştirilmesi şeklinde gerçekleştirilmiştir. YOLO v4 için Şekil 9’da verilen YOLO’nun çalışma mantığından, Faster R-CNN için girdi çözünürlük değeri, YOLO v5 içinse kanal derinliği parametre değişiminden türetilmiştir. Fakat sadece YOLO v4’ten doğru sonuçlar alınabilmektedir. Şekil 9’dan görüleceği üzere YOLO çalışma mantığında görüntüyü ızgaralara bölme işlemi vardır. Varsayılan değer olarak 416x416 sayıda ızgaraya bölünen görüntü için bu değer 832x832 olarak ayarlandığında görüntüde daha fazla sayıda ızgara oluşmakta dolayısıyla önceden 416x416 sayıdaki ızgaranın içine giremeyen, görüntüdeki küçük nesneler bu sayede ızgaraların içine girebilmektedir. Bu durumun basitleştirilmiş bir versiyonu Şekil 10 ile gösterilmiştir. Görüntüden tespiti istenen nesne, ızgara içine girdikten sonra YOLO için tüm tahminlerde ortak olan; ızgara için oluşturulacak vektör içinde güven skoru ve sınırlayıcı kutu parametreleri yer alacaktır.



Şekil 5. Farklı Çözünürlüklerdeki Görüntüler

Şekil 5 (a)’da düşük çözünürlüklü görüntüde yer alan orta/büyük ile Şekil 5 (b)’de yüksek çözünürlüklü görüntüde yer alan küçük boyutlu nesne örneği verilmiştir. İnce ayarla birebir alakası olan bu durumda, Şekil 6’daki gibi yüksek çözünürlüklü görüntüye çok

sayıda küçük boyutlu nesne sığabilmekte dolayısıyla görüntüde tespiti yapılacak küçük boyutlu nesne sayısı fazla olmaktadır. Buna karşılık düşük çözünürlüklü görüntüye küçük boyutlu nesneler sığamamakta veya düşük çözünürlüklü görüntü ile gösterilmek istenen nesne, yalnızca normal/büyük boyutta olmaktadır. Bu sebeple ince ayar işlemi, küçük nesneleri içeren yüksek çözünürlüklü görüntülerde daha başarılı olurken küçük nesneleri içermeyen düşük çözünürlüklü görüntülerde daha başarısız olmaktadır. Bu durum yapılan deneylerdeki taş tespit sayılarının gösterildiği Tablo 1 ile verilmiştir.

Tablo 1 ile verilen satranç test görüntülerinin %10'u olan 20 görüntü üzerinde YOLO v4'ün tespit edip/edemediği taş sayıları verilmiştir. Tablo 1'den görüleceği 20 test görüntüsünün çözünürlükleri düşükten yükseğe doğru çeşitlidir. YOLO v4'e uygulanan ince ayarlı transfer öğrenme metodu, yüksek çözünürlüklü görüntülerde neredeyse tespitini kaçırdığı nesne bırakmamıştır. Buna karşılık ince ayarın hiç uygulanmadığı sadece transfer öğrenmeli ağ ise yüksek çözünürlüklü görüntülerde yer alan küçük nesnelerin tespitini çokça kaçırmıştır. 800x800 gibi orta çözünürlüklü görüntülerde her 3 metot da benzer performans sergilemiş, düşük çözünürlüklü görüntülerde ise ince ayarlı transfer öğrenme metodu dışındaki metotlar daha yüksek performanslar sergilemişlerdir. Bu durumun nedeni olarak ince ayarlı transfer öğrenmenin düşük çözünürlüklü görüntüyü çok sayıda ızgaraya bölerek tespiti istenen az sayıdaki nesne boyutuna uygun ızgarayı denk getirememesidir.

Tablo 1. Görüntü Çözünürlüğü ve Nesne Tespiti İlişkisi [65]

Çözünürlük	Transfer Öğrenme		İnce Ayarlı Transfer Öğrenme		İnce Ayar	
	Tespit Edilen	Tespit edilemeyen	Tespit Edilen	Tespit edilemeyen	Tespit Edilen	Tespit edilemeyen
1600x1167	23	8	30	1	0	31
1300x956	9	23	30	2	1	31

390x280	5	0	2	3	5	0
1600x1167	14	18	27	5	0	32
794x447	6	0	5	1	6	0
1024x689	12	17	23	6	8	21
1280x853	14	16	28	2	2	28
1280x720	9	21	29	1	5	25
1024x576	21	11	28	4	15	17
1023x682	13	19	31	1	9	23
458x350	6	0	6	0	6	0
1114x1087	13	6	18	1	7	12
1023x682	13	17	27	3	9	21
1023x682	11	19	27	3	8	22
414x320	8	2	3	7	8	2
1920x1097	16	16	31	1	0	32
800x800	5	0	5	0	5	0
640x380	6	0	4	2	5	1
1600x1300	19	15	33	1	10	24

1600x900	7	25	31	1	1	31
----------	---	----	----	---	---	----

Küçük nesneleri tespit etmek için uygulanabilecek diğer teknikler ise ağın küçük nesnelerin tespitini sağlaması için evrişim katmanlarında yer alan filtrelerin boyutlarını küçültmek veya ağa fazladan evrişim katmanları eklemektir. Fakat bu işlemler mimarinin hesaplama maliyetine ek yük getirmekte ve orta/büyük boyutlu nesnelerin tespitinde kaçırımlara neden olmaktadır. Diğer bir küçük nesne tespiti yöntemi ise eğitimden önce yer alan veri ön işlem aşamasında sadece küçük nesnelere yakınlaştırma uygulayarak yakınlaştırılan alanların tüm görüntünden kırılmasıyla ayrı görüntülerin oluşturulmasıdır. Kırılan bu görüntüler, eğitim aşamasında ayrı birer görüntü olarak kullanılabilir fakat bu durumda kırılmadan önceki orijinal görüntünün yapısı bozulmuş olarak bu görüntülerin eğitime alınamaması durumu gerçekleşmektedir. Sadece kırılan küçük nesneleri içeren görüntüler eğitime alındığından ağ; küçük nesnelerin özelliklerini öğrenirken günlük hayatta sıkça karşılaşılan orta/büyük nesnelerin özelliklerini öğrenemeyerek bu nesneleri tespit edemeyecek bu yüzden de ağın genelleme yeteneğinde düşüş meydana gelecektir. Küçük nesne tespitinde uygulanan üçüncü yöntem ise çıkarım aşamasında kullanılan küçük nesneleri içeren test görüntülerine yakınlaştırma uygulanmasıdır. Bu sayede ağın küçük nesneleri, eğitimde gördüğü büyük nesneler olarak algılaması sağlanacaktır. Fakat uygulanan bu işlem ile bir önceki duruma benzer şekilde ağın orijinal küçük nesne görüntülerini tanıyamaması, sadece büyük nesne görüntülerini tanıması durumu meydana gelecektir.

3.1.3. Modellere Uygulanan Ayarlar

Hiper parametrelerin hepsinin de ortak amacı yitim değerini en aşağı seviyede tutarak başarılı ağırlıklar üretmektir. İlk belirlenmesi gereken değerler bu yüzden diğer hiper parametrelerin de bağlı olduğu epok ve öğrenme oranıdır. Yüksek tutulan epok değeri, ağın ezberlemesine yol açarken düşük tutulan bu değer ile ağın veriden öğrenememesi gerçekleşebilmekte ve gereksiz kaynak harcaması meydana gelmektedir. Doğru epok değeri seçiminin yapıldığı ancak yitim değerinin, eğitim boyunca 0'a yaklaşmasından anlaşılabilmektedir. Başarılı bir ağ eğitimi için önemli olan hiper parametreler şunlardır;

Epok – Eğitimin kaç tur süreceğinin sayısıdır. Her epokta tüm veri seti baştan sona taranmaktadır. Ağıdaki epok sayısını bir olarak belirlersek veri seti hem ileri hem de geri yönlü geçişte birer kere taranır.

İleri ve geri yönlü geçiş – Gerçek değerler üzerinde tahminler, ileri yönlü geçişte yapılırken geri yönlü geçiş ile gerçek ile tahmin edilen arasındaki fark olan hata değeri azaltılmaya çalışılmaktadır.

Yığın – Veri setindeki görüntüler yığın boyutuna göre gruplara ayrıldığında oluşan görüntü grubu sayısıdır.

Yığın boyutu – Her bir iterasyonda ağa kaç adet görüntünün kabul edileceğinin sayısıdır.

İterasyon – Veri seti boyutu / Yığın ile bulunur. Örnek olarak 1.000 görüntümüz var diyelim, yığını da 64 seçtik, iterasyon sayımız 15.625 çıktı, yani bu örnekteki 1 epoku tamamlamak için 15.625 iterasyona ihtiyaç vardır.

Öğrenme Oranı – Her bir iterasyonda ağırlıkların hangi oranda güncelleneceğinin değeridir. Eğitim esnasında isteğe bağlı olarak değiştirilebilmektedir.

Ağırlık – $(\text{Girdi} \times \text{Ağırlık} = \text{Çıktı})$ Girdi ve çıktı değerleri ağa verilerek ağı eğitimi esnasında ağırlık parametresi oluşturulmaktadır. Girdi verisinin olması gereken çıktı verisine ne kadar yaklaşıldığının ifadesidir.

Yitim Değeri – Olması gereken ile tahmin edilen arasındaki fark değeridir. Başarıyı veren doğruluk skoru ile arasında ters orantı vardır. Bu değer eğitimi boyunca mümkün olan en düşük seviyelerde tutulması beklenir.

Ağırlık Kaybı – Ağı genelleştirme yeteneğini büyük ölçüde artırmaktadır. L1 ve L2 normalizasyon değerlerini barındırmaktadır. Normalizasyon sayesinde yitim değeri düşürülmektedir. Ağırlık kaybı aynı zamanda modelin ezber yapmaması için ayarlanmaktadır [36].

Momentum – Geçmiş ağırlıkların oluşacak yeni ağırlıklara ne kadar etkisinin olacağının oranıdır. Derin öğrenme gradyan tabanlı bir yaklaşıma sahip olduğundan ağı yakınsamasını (amaç yitim fonksiyonunu düşürme) hızlandırıcı etkisi vardır.

Model eğitiminde kullanılan optimizasyon algoritmaları ise tespit hatasını azaltmaya yöneliktir. Tespit hatası, olması gereken gerçek değerler ile tahmin edilenler arasındaki farktır. Model gerçek, olması gereken değerlere ne kadar yakın tahminlerde bulunursa ağırlıklar da o derece kaliteli çıkmaktadır. Başka bir deyişle ağırlıklar, ağırlık veriye ne kadar uyum sağladığının göstergesidir. Büyük ağ için büyük veri seti, küçük ağ için küçük veri seti seçilmelidir. Hata oranı düştükçe yitim değeri, yitim değeri düştükçe de tespit hatası düşmektedir.

Çalışmada kullanılan YOLO v4 ve v5 ağlarının geri yayılımında “Adam” optimizasyon algoritması kullanılarak öğrenme oran değeri, her bir ağ ağırlığı için bağımsız olarak güncellenmiştir. Faster R-CNN’de ise “momentumlu SGD” optimizasyon algoritması tercih edilmiştir. Her üç veri setinden de birden çok nesne tespit edilmeye çalışıldığından yitim fonksiyonu olarak; çoklu sınıflandırma problemlerinde sık kullanımı olan “kategorik çapraz entropi” seçilmiştir.

Eğitim boyunca oluşan ağırlıkların güncellenme miktarını, öğrenme oranı belirlemektedir. Bu yüzden öğrenme oranına adım boyutu da denmektedir. Yetersiz kaynak varsa epok ve öğrenme oranını düşürmek mantıklı olurken bu yöntem eğitim süresinin uzamasına neden olmaktadır. Öğrenme oranı, önceki derin öğrenme çalışmalarından ortalama bir değer olarak tüm ağ çalıştırmalarında 0.00013 olarak belirlenmiştir.

Modelin genellemesine veri çoklamanın yardım etmesi gibi ağırlık kaybı da yardım etmektedir. Veri çoklama, veri tarafında yapılırken; ağırlık kaybı, model ayarlama tarafında yapılmaktadır. Veri çoklama genelde eğitim esnasında sanal bir çoklama ile yapılırken ağırlık kaybı ve momentum ise eğitimden önce model ayarlaması ile gerçekleştirilmektedir. Önceki çalışmalara bakarak momentum değeri 0.9, ağırlık kaybı ise 0.005 olarak tüm ağlarda aynı değerlere ayarlanmıştır.

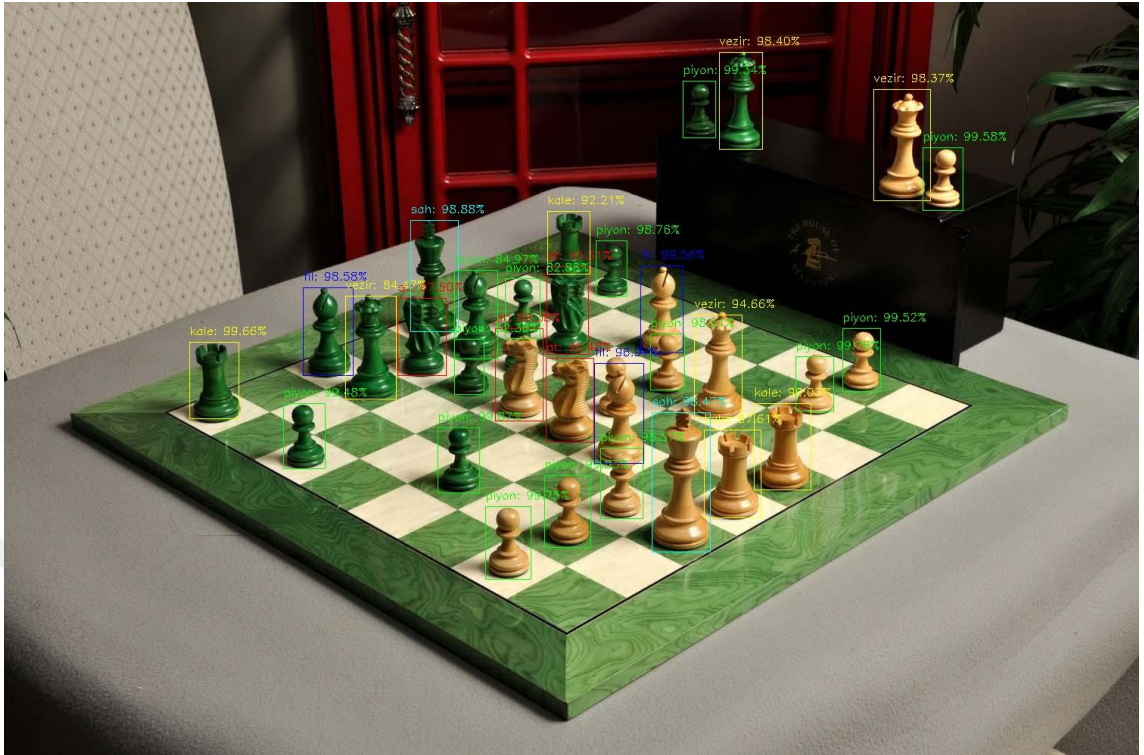
Yığın boyutu ile dolaylı yoldan ağa her iterasyonda ne kadar görüntü kabul edileceği belirlenmektedir. Bu değer az tutulursa eğitim uzun sürecek fakat doğrulukta artış olacak, çok tutulursa eğitim kısa sürecek ama doğrulukta düşüş olacaktır. Doğrulukta artışın olma sebebi; ağırlık analizini, her iterasyonda daha az sayıda görüntü üzerinde gerçekleştirmesinden kaynaklanmaktadır. Bu sebeple yığın boyutu, tüm çalıştırmalarda ortalama bir değer olan 64 olarak alınmıştır.

3.2. Materyal

Çalışmalarda kullanılan veri setlerinden ikisi doğal, biri sentetik görüntülerden oluşmaktadır. Çalışmaların birinde [37] kullanılan veri seti sentetik ve tamamen ince ayarın etkinliğini araştırmaya yönelik olarak, arama motorlarından taş çeşidi dengesi göz önünde bulundurularak satranç taşı içeren görüntüler toplanmıştır. Kullanılan ince ayar tekniğiyle beraber ESA tabanlı YOLO v4 derin öğrenme ağının başka veri setlerine uygulanması halinde küçük nesnelerin tespiti, aynı satranç veri setinde olduğu gibi mümkün olacaktır. Şekil 6'da sadece transfer öğrenme uygulanan ağın, derindeki küçük nesneleri tespit etmede zorlandığı, Şekil 7'de ise ince ayarın uygulandığı aynı ağın, Şekil 6'da tespiti kaçırılan ve kaçırılmayan ile yanlış tespit edilen nesneleri yüksek skorlarla tespit ettiği görülmektedir.



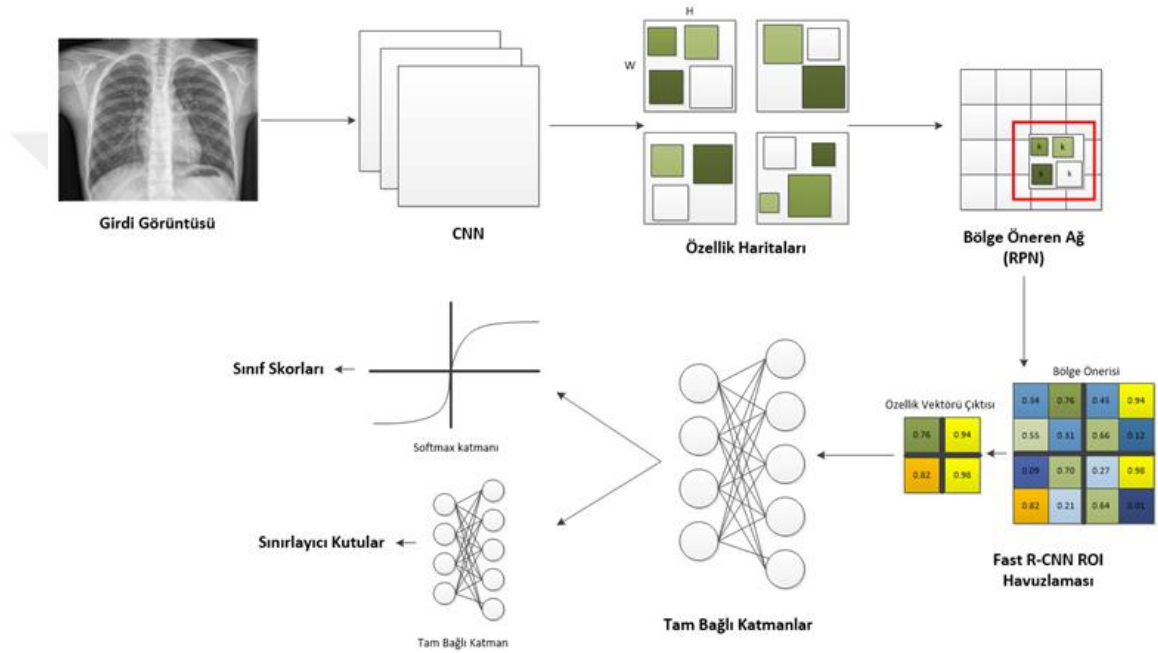
Şekil 6. Küçük Nesneler İçeren Görüntü Üzerinde Transfer Öğrenmeli YOLO v4 tespiti



Şekil 7. Küçük Nesneler İçeren Görüntü Üzerinde İnce Ayarlı Transfer Öğrenmeli YOLO v4 tespiti

[66] çalışmasında kişinin yüzde kaç oranında zatürreli veya sağlıklı olduğunu söyleyen, uzmanlara yardımcı yapay zekâ tabanlı bir sistem önerilmiştir. Röntgen görüntüsüne bakan kişinin yeterince deneyimli olmaması veya göz yorulması sonucu görüntüyü yanlış yorumlayarak yanlış teşhis koyması gibi hataları en aza indirerek yanlış teşhisin önüne geçen ve bazı durumlarda uzmandan daha başarılı teşhisler koyan, ayrıca teşhis süresini kısaltarak uzun bekleme süre sorununu ortadan kaldıran bu yöntem sayesinde yapay zekâ tabanlı sistemlerin geleceğine yönelik bir perspektif çizilmiştir. Bu çalışmada 1-5 yıl arasında zatürre hastası olan ve sağlıklı kişilerin akciğer röntgen görüntüleri üzerinde transfer öğrenmeli ve öğrenmesiz Faster R-CNN ağları çalıştırılmış ve transfer öğrenmeli Faster R-CNN ağından %98 ile transfer öğrenmesiz Faster R-CNN'e göre %1'lik daha yüksek f1 skor alınmıştır. Şekil 8'de bu çalışmada kullanılan Faster R-CNN ağının mimarisi gösterilmiştir. Sağlıklı ve zatürreli olan akciğer röntgen görüntüleri etiketleri ile birlikte ilk olarak Faster R-CNN omurga ESA'sına girmekte, ESA'dan çıkarılan özellik haritaları -matrisler- Bölge Öneren Ağ ile tespiti istenen nesneleri içermeye ihtimali olan bölgeleri oluşturarak yine aynı ağ tarafından tüm mimariye önerilmektedir. Son olarak hesaplama maliyetini düşürücü en yüksek havuzlamaya benzer

işleviyle RoI havuzlamasından geçen özellikler iki boyutlu matris tipinden tek boyutlu vektör tipine dönüştürülerek tam bağlı katmanlar aracılığıyla sınıflandırma ve kutu çizimi işlemlerine tâbi tutulmaktadır. Bu çalışmanın omurga ESA'sında daha katmanlı VGG19 yerine ResNet50 seçilerek FLOPs sayısında düşürme meydana getirilmiştir. Çünkü katman sayısı arttıkça hesaplama yükünü veren FLOPs sayısı da artmaktadır, VGG19 ile ResNet50 arasında neredeyse 6 kat oranında FLOPs sayısı mevcuttur [67].



Şekil 8. Zatürreli/Sağlıklı akciğer ayrımı yapan Faster R-CNN mimarisi [67]

[68] çalışmasında ise transfer öğrenmenin etkinliği tek veri seti üzerinde üç farklı ağın çalışması ile incelenmiştir. Çalışmada güncel ESA tabanlı derin öğrenme ağlarından YOLO v4, v5 ve Faster R-CNN kullanılmıştır. Aynı şartlar altında transfer öğrenmeli ve öğrenmesiz olmak üzere toplamda 6 farklı ağ eğitimi gerçekleştirilmiştir. Transfer öğrenmeli ağlar, ağ içi karşılaştırmalarda transfer öğrenmesiz ağlara göre daha başarılı olmuştur. Tüm karşılaştırmalarda ise transfer öğrenmeli YOLO v5 %98 f1 skor ile en yüksek başarıya ulaşmıştır. Bu çalışmada ayrıca transfer öğrenmeli ve öğrenmesiz ağların eğitim süreleri karşılaştırılmıştır. Transfer öğrenmenin aynı epok ve aynı büyüklükteki veri setleri üzerindeki çalıştırmalarında eğitim süresini düşürmediği ama artırmadığı da yani eşit şartlar altında çalışan ağların eğitim süresinde ciddi farklar oluşturmadığı sonucu ortaya çıkmıştır. En kısa çalıştırma süresini CPU üzerinde çalıştırmalarda; trans-

fer öğrenmesiz YOLO v5, 4.65 saat transfer öğrenmeli YOLO v4, 4.84 saat ile almıştır. Büyük veri setinde çalışan transfer öğrenmesiz ResNet ağı ise GPU'nun gelişmiş v2-8 tensör işleme birimi ile 7.65 saat çalışma süresine sahip olmuştur. Buradan yola çıkarak transfer öğrenmenin düşük donanım ve veri kaynaklarındaki çalıştırmalarında süreyi kısaltan bir teknik olduğu söylenebilir.

3.2.1. Faster R-CNN Yöntemi

Derin öğrenme çalışmaları, bir bankanın gri renkteki banknotların üzerindeki rakamları tanıyan LeNet-5 ağı ile başlamış daha sonra ImageNet yarışında 1. olan AlexNet ile gün geçtikçe hız kazanmıştır. R-CNN'e kadar geliştirilen ağlar, tek nesne içeren görüntülerde başarılı olurken R-CNN ile görüntünün tamamına bakılarak görüntülerde yer alan çoklu nesneler tanınabilmektedir. R-CNN ve sonraki versiyonları, tüm görüntüyü bölgelere bölerek tespit işlemlerini gerçekleştirmektedir. Bir görüntü için 2.000 sayıda bölge önerisi üreterek her bir öneri için bir ESA çalıştıran R-CNN, mimariye ek yük getirmektedir. Fast R-CNN ise R-CNN gibi bir görüntü için 2.000 bölge önerisi üretmekte fakat tüm bu önerileri tek bir ESA'dan geçirmektedir. Buna karşılık R-CNN ailesinin sonuncusu olarak geliştirilen Faster R-CNN, önerilen bölgeleri tüm ağ ile paylaşarak neredeyse 0 maliyetle çalışmaktadır. Versiyonlar arası kullanılan algoritma farklılıkları da vardır. R-CNN ve Fast R-CNN hesaplama yükü fazla, Seçici Arama algoritmasıyla görüntüdeki yakın ve benzer pikselleri gruplandırarak bölge önerileri yaparken Faster R-CNN, omurga ESA'sından çıkardığı özellikleri maliyeti düşük RPN'e vererek önceki versiyonlarıyla benzer bölge önerme işlemini yapmaktadır. PASCAL VOC yarışlarında yapılan testlerde RPN'li Fast R-CNN'ler, Seçici Aramalı Fast R-CNN'lere göre daha iyi performans sergilemişlerdir. RPN, tam bağlı bir ağıdır ve her bir pozisyonda nesne sınırlarını ve nesnesizlik skorlarını eş zamanlı olarak tahmin etmektedir. Faster R-CNN'de; RPN ve Fast R-CNN ağları birleştirilerek birbirlerinin evrişim özelliklerini paylaşabilen tek bir ağ oluşturulmuştur. RPN, Fast R-CNN'e nereye bakması gerektiğini söyleyerek dikkat mekanizmasını kullanmaktadır. Faster R-CNN ağı iki modülden oluşmaktadır. Bunlardan ilki özellik çıkaran omurga ESA'sı ve RPN ağı iken ikincisi bu özellikleri sınıflandıran Fast R-CNN ağıdır.

Faster R-CNN'in seçilen omurga ağından RPN'e gelen özellikler ile bölge önerileri üretilmektedir. Aslında üretilen bölge önerileri, tespiti istenen nesneleri içermesi olasılığı

olan görüntü özellik haritalarıdır. Daha sonra RPN'den çıkan öneriler, RoI havuzlamasına girerek sadece en güçlü özelliklerin ağda tutulması sağlanmakta ve iki boyutlu matris tipinde olan özelliklerin tek boyutlu vektör tipine çevrilmesi işlemi gerçekleştirilmektedir. Ağda kalan vektör tipindeki özellikler son olarak üç tam bağlı katmana girdi olarak verilmektedir. Bu tam bağlı katmanlardan ilki sınıflandırma performansını artırmak için, diğer ikisi kutu sınıflandırmada kullanılan “cls” katmanı ve kutu parametrelerinin belirlenmesinde kullanılan “reg” katmanıdır. RPN kayan pencere kullanarak k sayıda öneri için; $2k$ sayıda sınıflandırma (nesne var/yok) ve $4k$ sayıda kutu üretimi (merkez x y , genişlik, yükseklik) yapmaktadır. Her bir kayma konumunda varsayılan $k = 3$ için ölçek ve çerçeve oranı parametrelerince 9 çapa kutusu üretilmektedir. Çapa adı verilen k önerileri, görüntü dönüşümlerine karşı değişmezdir yani görüntü çevrilse, döndürülse bile çapa nesneye tam olarak uymaktadır.

Ağ tarafından rastgele olarak 256 çapa seçilerek mini-yığında 1:1 oranında olan pozitif ve negatif örnek çapalar için yitim fonksiyonu hesaplanmaktadır. Alternatif eğitim adı verilen eğitim türünde ilk olarak RPN eğitilmekte, daha sonra buradan çıkan öneriler Fast R-CNN eğitiminde kullanılmaktadır. Fast R-CNN tarafından ayarlanan ağ, daha sonra RPN eğitimini başlatan nokta olarak kullanılmaktadır. Bu işlemler en düşük yitim değerine ulaşana veya kullanıcı eğitimi kesene kadar tekrarlanmaktadır. Böylece neredeyse sıfır maliyetle bölge öneri problemi çözülerek önerilen bölgelerin kalitesi artırılmaktadır. Ayrıca gerçek zamana yakın performanslar bu sayede sergilenmektedir. Bu çalışmada da strateji olarak alternatif eğitim yöntemi tercih edilmiştir. RPN eğitmek için her bir çapaya nesne var/yok yani ikili sınıf etiketi ataması yapılmaktadır. Çapaya pozitif etiket atama, şu iki yolla yapılmaktadır;

çapa veya çapalardan (i) en yüksek IoU değerine sahip olanlar,

(ii) IoU değeri 0.7'den yüksek olanlar.

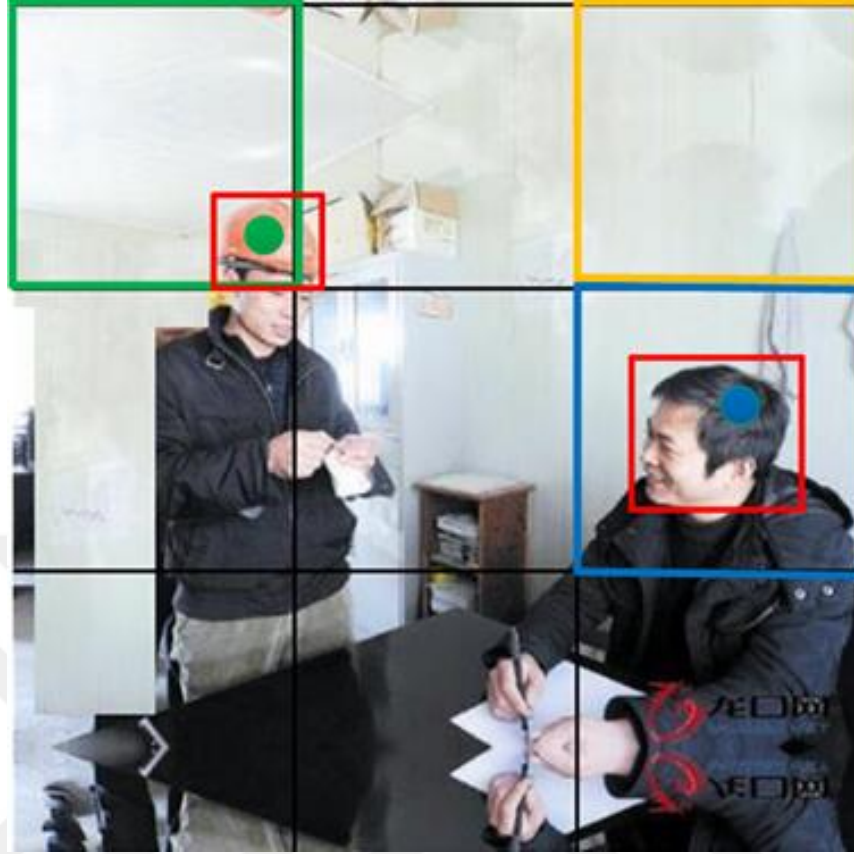
Tek bir olması gereken kutu, birden çok pozitif etiketli çapaya atanabilmektedir. Genelde ikinci kural pozitif örnekleri belirlemede yeterli olurken yine de olasılığı az olan durumlar (ikinci şarttan pozitif etiket bulamayan) için birinci şarta da bakılmaktadır. Pozitif olmayan çapaya negatif etiket atamak içinse çapanın IoU değeri 0.3'ten küçük mü diye bakılmaktadır. Ne pozitif ne de negatif olan çapalar eğitime alınmamaktadır.

Daha sonra RoI havuzlamasından geçen bölgelerin boyutu düşürülerek hesaplama maliyeti ciddi oranda azaltılmaktadır. Özellikler son olarak nesne skoru ve regresyon için iki dala yönlendirilerek ağı sonlandırmaktadır. Ağ ayrıca ölçek ve çerçeve oranı parametreleri sayesinde tespiti istenen nesneyi bulma ihtimalini artırmak için sınırlayıcı kutu üretimini artırmaktadır. YOLO'daki ızgara sistemini kullanmadığı, ölçek ve çerçeve oranı parametrelerini kullandığı için yan yana olan nesneleri daha kolay tespit etmektedir. Faster R-CNN'in tek dezavantajı olarak konumlandırma ve sınıflandırma işlerini iki aşamada yaptığı için tek aşamalı ağlara göre daha yavaş olmasıdır.

Transfer öğrenmenin Faster R-CNN için kullanımında ağı omurga ESA'sında yer alan önceden büyük veri setinde eğitilmiş ResNet50 ağı kullanılmıştır. Tek dallı ve daha derin yapılı VGG19 yerine bu ağı kullanılma sebebi, ResNet50 ağının VGG19'a göre FLOPs sayısının yani hesaplama maliyetinin daha düşük olmasıdır [67]. Ayrıca YOLO omurga ağına göre daha az katmanlı olan bu ağ sayesinde daha küçük veri setlerine uyum sağlayan Faster R-CNN, gerçekleştirilen deneylerde yer alan küçük veri seti üzerinde çalıştırmalarda YOLO ağlarına göre daha başarılı olmuştur.

3.2.2. YOLO Algoritması

Tek bir ileri yönlü ağ geçişinde görüntü üzerindeki tüm nesneleri tespit ettiği için “bir kere bakarsın” manasına gelmektedir. Regresyon tabanlı olduğu için kutu çizme işleminde olduğu gibi sınıflandırma işlemini de matematiksel yolla çözmektedir. Ayrıca Faster R-CNN gibi sınıflandırmayı ve kutu çizmeyi iki aşamada yapmayarak tek aşamada yaptığı için daha hızlıdır.



Şekil 9. YOLO çalışma mantığı [69]

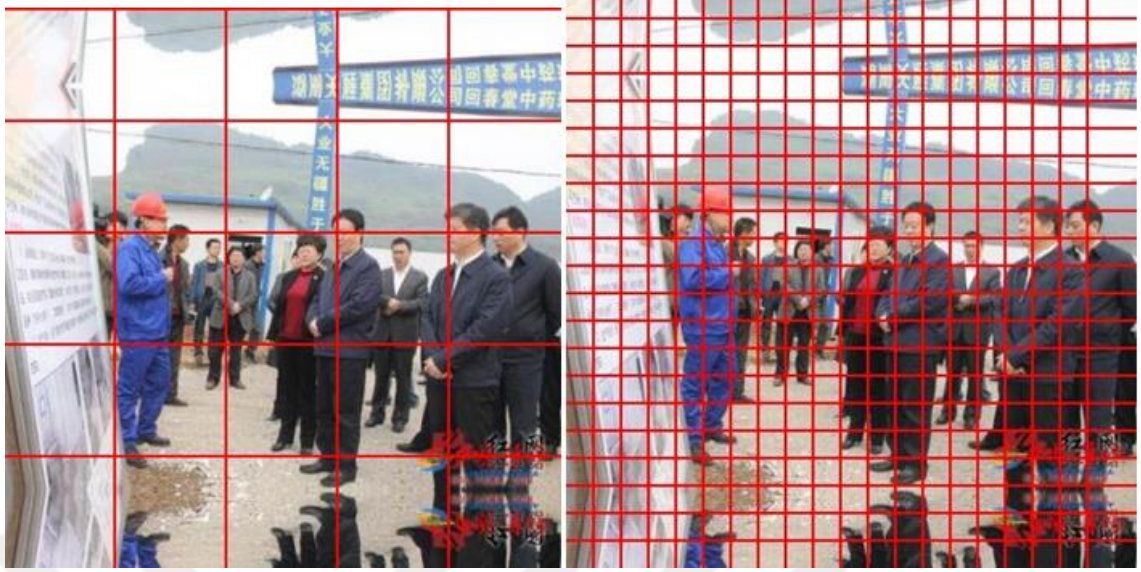
YOLO, Şekil 9'deki gibi mimarisine kabul ettiği girdi görüntülerini ızgaralara bölmektedir. Görüntüyü normalde daha fazla sayıda ızgaraya bölmektedir. Fakat örneğin daha iyi anlaşılması için görüntü 3x3 sayıda ızgaraya bölünmüştür. Izgaralara bölündükten sonra her ızgaranın ortasında kalan nesne, ağ tarafından tespit edilmeye çalışılmaktadır. Bu örnekte tespit edilmek istenen nesneler kafa ve kask olduğu için bu nesneler ağ tarafından tespit edilmeye çalışılmaktadır. Kırmızı dikdörtgenler olması gerekenleri yani gerçek nesne alanlarını göstermektedir ve gösterim için hem kafa hem de kask nesnesinin etrafına mümkün olan en dar, kutu dışında nesne uzantısı kalmayacak şekilde çizilmişlerdir. Mavi ızgara için orta nokta kafa nesnesine, yeşil ızgara içinse kask nesnesine denk geldiği için bu ızgaralar bu nesneleri tespitten sorumlu olmaktadır. Daha sonra her bir sorumlu ızgara için $[p \ x \ y \ w \ h \ c]$ vektörü üretilmektedir. Izgarada yer alan tespiti istenen nesnenin varlığına göre 1 veya 0 değerini alan p , güven skorunu ifade etmektedir. p , mavi ve yeşil ızgaralar için 1 -çünkü tespiti istenen nesne ızgarada var- değerini alırken sarı ve diğer ızgaralar için 0 -tespiti istenen nesne ızgarada yok- değerini almaktadır. YOLO tek seferde sınıflandırma ve konumlandırma işlemlerini yaptığı için tek

vektöre; güven skoru p 'nin yanına x, y, w, h kutu oluşturma değerlerini de eklemiştir. x, y sınırlayıcı kutunun; sırasıyla yataydaki ve düşeydeki merkez noktalarını, w, h ise genişlik ve yüksekliğini göstermektedir. Vektörün son elemanı olan c ise aslında p ile anlam kazanmaktadır. c değeri geçerli ızgaranın içinde olan tespiti istenen nesnelerden hangisinin olduğunun kategorik sırasını göstermektedir ve yalnızca p değeri 1 olduğunda aktif olmaktadır. Bu örnek için yeşil ızgarada üretilen c değeri; 0, mavi ızgarada 1, sarı ızgarada ise p değeri sıfır olduğundan; c ve vektörün diğer değerleri üretilmemektedir.

Tablo 2. YOLO Varsayılan Izgara Sayıları

Az	416x416
Orta	608x608
Çok	832x832

Tablo 2’de YOLO’nun varsayılan olarak kullandığı, mimari girişinde kabul ettiği görüntüleri bu değerlerce ızgaraya parçaladığı sayılar verilmiştir. Izgara sayısı arttıkça ızgaraların içine girebilen nesne sayısı da artmaktadır. Diğer bir ifadeyle tespit edilmek istenen nesneler, ızgaraların içine girdikçe tespit ihtimali artmaktadır. Şekil 10 (a)’da 5x5 (az) ve Şekil 10 (b)’de 21x21 (çok) ızgara sayısı parçalama örnekleri verilmiştir. Örneğin daha iyi anlaşılması için küçük değerler kullanılmıştır. Normalde görüntü, Tablo 2’de verilen sayıca ızgaralara parçalanmaktadır. Şekil 10 (a)’nın baştan 3. satır ve 3. sütunundaki ızgaraya tespiti istenen 3 adet kafa nesnesi girmiştir. Bu durumda YOLO hangi kafa nesnesini tespit edeceğini (hangi biri için vektör üreteceğini) şaşıracaktır. Çünkü YOLO’nun çalışma mantığında her tespiti istenen nesneyi içeren ızgara için sadece bir vektör üretilmesi vardır. Buna karşın Şekil 10 (b)’deki görüntü için aynı lokasyondaki her bir kafa nesnesine birer ızgara denk geldiği için algoritma bu nesneleri rahatlıkla tespit edebilecektir.



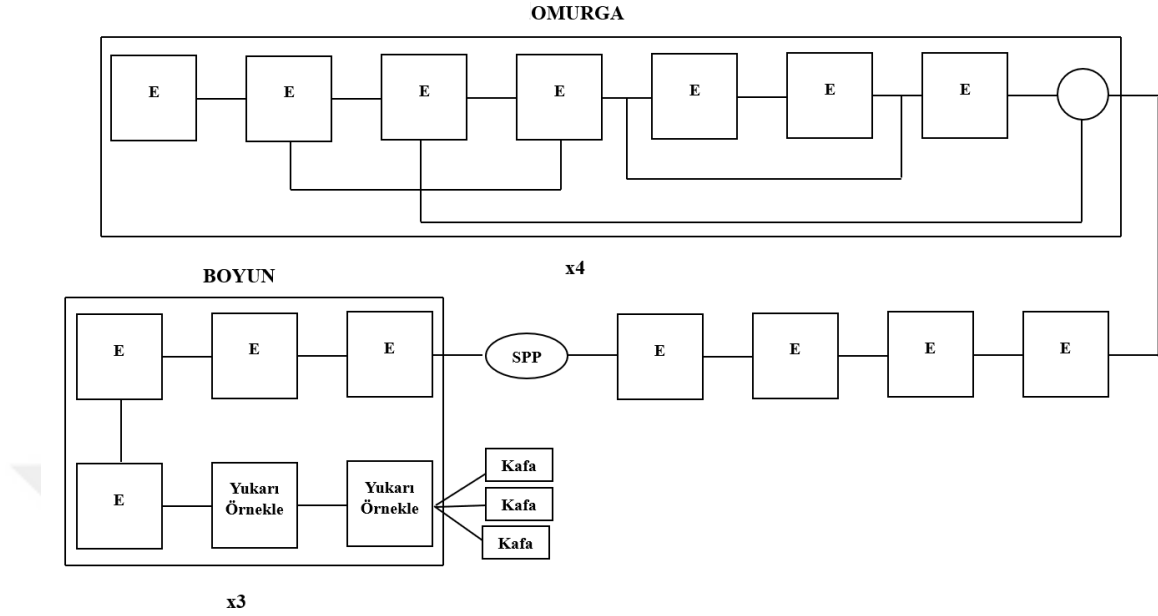
(a) az ızgara

(b) çok ızgara

Şekil 10. İnce ayar için YOLO ızgara sayı artırımı

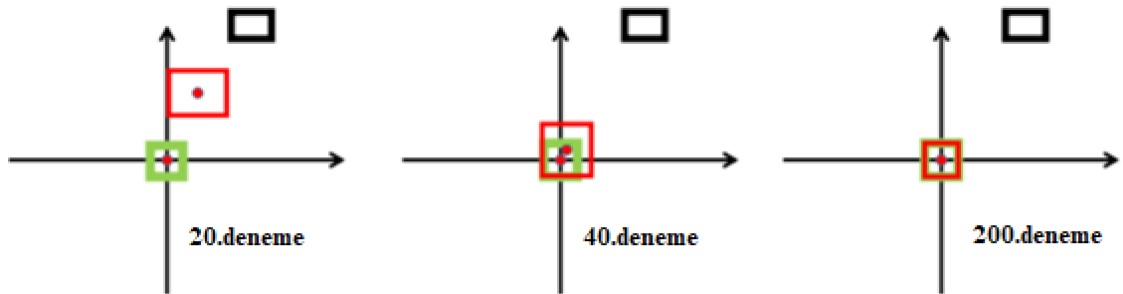
Nesne tespitleyicilerinde ortak olarak yer alan kavramlar; omurga, boyun ve kafa kısımlarıdır. YOLO v1 ve v2'nin omurga ağında 19 katman kullanıldığı için derin özellikler çıkarılamazken YOLO v3 ve YOLO v4; diğer versiyonlarıyla aynı konumda yani ağın başlangıcında olan omurga ağında 19 katman yerine 53 katman kullanarak daha derin özellikleri görüntülerden çıkarabilmektedir. Ayrıca YOLO v3 ve v4'te, önceki versiyonlarında olan girdi ve test görüntüsü çözünürlük değerinin aynı olması gibi bir kısıtlama yoktur. YOLO v4'ün tahminleyici katmanları da YOLO v3 ile aynı olarak 3 adettir. Her iki model de sınırlayıcı kutu üretmede k ortalamalar algoritması kullanmaktadır. YOLO v4, YOLO v2'deki gibi son katmanda 5 sınırlayıcı kutu uygulamak yerine 3 farklı konuma 9 sınırlayıcı kutu uygulamaktadır ($9 \times 9 \times 9 = 729$). Bu da yaklaşık 146 kat daha fazla ihtimalle tespiti istenen nesneyi bulmak demektir. Bu yüzden YOLO v4, önceki versiyonlarına göre daha başarılı tespitler yapmaktadır.

YOLO v4'ün omurgası Darknet53'ün gelişmiş versiyonu CSPDarknet53, boynu Faster R-CNN'de de yer alan değişken boyutu, sabit boyuta getiren SPP ve özellikleri birleştiren PAN, kafası YOLO v3'ün kafasıyla aynı olan tam bağlı katman görevi gören tespitleyici katmanlardır. Şekil 11'de YOLO v4 mimarisinin basitleştirilmiş bir versiyonu gösterilmiştir. Şekilde görüldüğü üzere ağ; görüntü özelliklerinin çıkarıldığı omurga ile başlayıp sırasıyla özelliklerin birleştirildiği ve sabit boyuta getirildiği boyun ve tespit işlemlerinin gerçekleştiği kafa bölümleri ile sonlanmaktadır.



Şekil 11. YOLO V4 Mimarisi

YOLO v4'ün tespit yeteneğini artırmak için kullandığı çeşitli algoritmalar vardır. Omurgasında (yani görüntüden özellik çıkarım aşamasında) kullanılan veri çoklama, kafasında (tespitleyicisinde) isteğe bağlı olarak seçilen CIoU yitim fonksiyonu bunlara örnek verilebilir. Kullandığı yitim fonksiyonu, tahmin edilen ile olması gerekenin kesişimini Şekil 12'de gösterildiği gibi 200. denemede, YOLO v5'te kullanılan GIoU fonksiyonuna göre daha az iterasyonda bulabilmektedir. Şekilde olması gerekeni ve tahmin edileni gösteren dikdörtgenlerin renkleri sırasıyla; yeşil ve kırmızıdır. Kırmızı şekil, iterasyon sayısı arttıkça yeşil şekle uyarak 200. iterasyonda yeşil şekle tam olarak oturmaktadır. Bu, tahminin %100 doğru olarak gerçekleştiğini göstermektedir.

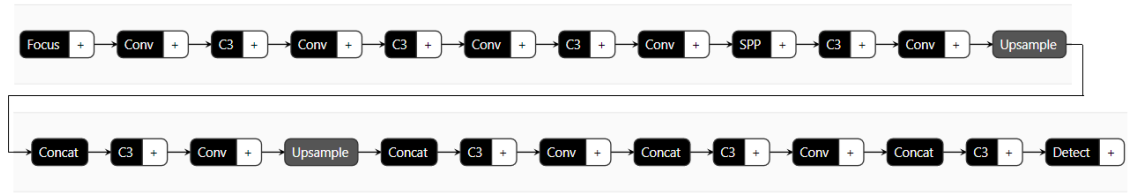


Şekil 12. CIoU çalışma hızı

Transfer öğrenmenin YOLO v4'e uygulanması için önceden MS COCO veri setinde eğitilmiş YOLO v4 ağırlık çıktısı olan "yolov4.conv.137" ile standart ESA'daki son katman nöron sayısı değiştirme benzeri, ayarları değiştirilen ağına eğitime başlanmıştır. İnce ayar için eğitime başlamadan önce ağına hiper parametrelerinden biri olan ızgara sayısı 416x416'dan 832x832'ye getirilmiştir. İnce ayarlı transfer öğrenme işlemi için bu iki yöntem birlikte uygulanmıştır. Transfer öğrenmesiz eğitim içinse ağı ayarları transfer öğrenme uygulamasındaki gibi değiştirilmiş fakat eğitim için önceden eğitilmiş ağına çıktı ağırlıkları kullanılmamış ve ızgara sayısı değişikliği yapılmamıştır.

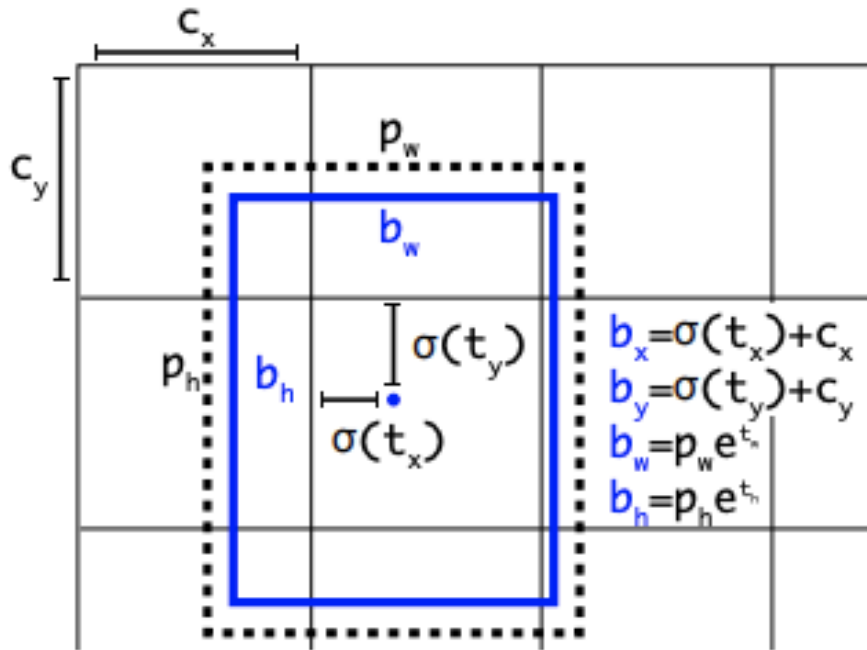
Şekil 13'te YOLO v5 mimarisi verilmiştir. Şekilden görüldüğü üzere YOLO v5'te omurga ağı olarak, derin ağlarla işlem süresinde önemli gelişme sağlayan CSPNet - YOLO v4'ün de omurga ağı olan- ve 3 adet YOLO v4 evrişim katmanı yerine geçen Odak modülü kullanılmaktadır. Odak modülü, eğitim için gerekli CUDA (GPU üzerinde çalışan derin öğrenme ağlarının çalışma süresini kısaltan kütüphane) hafızasını azaltırken ileri ve geri yönlü yayılmayı artırmaktadır [70]. Mimaride yer alan boyun kısmında ise YOLO v4'e benzer şekilde PANet ve SPP katmanları, genelde modellerin nesne ölçeklemesindeki genelleştirmesine yardım eden özellik piramitlerinin üretimini sağlamaktadır. Modelin kafa bölümü de YOLO v3 ve v4 ile aynıdır. YOLO v5'e has Odak Yitim fonksiyonunu dengesiz bir veri setimiz olmadığından kullanmadık. İkili sınıflandırma da yapmadığımızdan Logit Fonksiyonlu İkili Çapraz Entropi kullanmayarak çoklu sınıflandırma için ideal Kategorik Çapraz Entropi fonksiyonunu kullandık.

IoU yitim yerine genelleştirilebilir yitim (GIoU yitim) algoritmasını işe koşması, YOLO v4'ten diğer bir farkı ise oto öğrenme sınırlayıcı kutuları kullanarak genel doğruluğu artırmasıdır [71]. Fakat YOLO v4'te kullanılan CIoU'nun, GIoU'ya göre avantajı Şekil 12'de verildiği gibidir. Her iki algoritmanın; omurgasında da CSPDarknet53 ağı kullanıldığından görüntüden derin özellikleri çıkarma yetenekleri benzerdir, kullandıkları yöntemler (veri çoklama) de aynı olduğundan ağa kazandırdıkları doğruluklar benzer oranda artış göstermektedir.



Şekil 13. YOLO v5 Mimarisi [70]

YOLO'nun önceki versiyonları C programlama diliyle yazılmışken YOLO v5, Python programlama diliyle yazılmıştır. Böylelikle YOLO v5'i IoT cihazlara yüklemek ve farklı cihazlarla entegre etmek kolaylaşmıştır. YOLO v5'in boyutu sadece 27 MB iken Darknet kullanan YOLO v4'ün boyutu 244 MB, PyTorch kullanan YOLO v4'ün boyutu ise 2 GB'tır. Öğrenme verimini artırmak için derinleşen ağların parametre sayıları da artmaktadır. YOLO v5'in farklı versiyonlarında, n'den x'e doğru parametre sayısı ve ağ doğruluğu artarken tespit hızı düşmektedir. Mevcut çalışma gerçek zamanlı olmadığı için en yüksek hızlı versiyonlar (n ve s gibi) yerine en yüksek doğruluk veren 444 katmanlı, 86.186.872 parametrelili versiyon (x versiyonu) çalışmada kullanılmıştır. Aslında tüm versiyonlar aynı ağ üzerinde çalışmaktadır. Versiyonlar arası değişim, ağın ölçek ve ince detayları yakalama kapasitesinde ortaya çıkmaktadır. Bu değerler arttıkça parametre sayısı da artmaktadır.



Şekil 14. YOLO v5 öğrenen çapa kutuları [72]

Şekil 14'te YOLO v5 öğrenen çapa kutularının oluşum aşamaları verilmiştir. Öğrenen çapa kutusu, ızgara kenar uzunluğunun üzerine ızgara merkezinden ızgara kenarına kadar olan uzunluğu ekleyerek en uygun sınırlayıcı kutuyu bulmaktadır. Bu sayede olması gerekene en yakın tahminler yapılmaktadır. Öğrenen çapa kutularını anlamak için öncelikle sınırlayıcı kutu ve çapa kutu arasındaki farkı tanımlamamız gerekir. Çapa kutu ve sınırlayıcı kutu programcı tarafından tanımlanan bir en ve boya sahip içi boş dikdörtgen şeklindedir. Öğrenen çapa kutusu ise farklı boyutlardaki nesnelerin şekline uymaya çalışarak tespit işlemi sonunda daha doğru bir görünüm elde edilmesine yardımcı olmaktadır.



(a) YOLO v5



(b) YOLO v4



(c) Faster R-CNN

Şekil 15. Öğrenen çapa ve çapa kutu farkı

Şekil 15 (a)'da görüldüğü üzere YOLO v5'in kullanmış olduğu öğrenen çapa kutusu sayesinde tespit edilen nesnelere daha çok uyan kutular elde edilmiştir. YOLO v4 ve Faster R-CNN'in kullanmış olduğu çapa kutular ise sırasıyla Şekil 15 (b) ve (c)'de gösterildiği gibi daha belirsiz bir yapıya sahiptir.

YOLO v5 için uygulanan transfer öğrenme işlemi de aynı YOLO v4 için seçilen MS COCO veri setinde eğitilmiş ağ çıktısı “yolov.conv.137” gibi “yolov5m.pt” seçimi yapılarak gerçekleştirilmiştir. Bu sayede MS COCO veri seti üzerinde eğitilmiş modelin m versiyonu ağırlık çıktısını kullanarak kendi problemimizdeki x versiyonu modelin eğitime başlanmış olmaktadır.

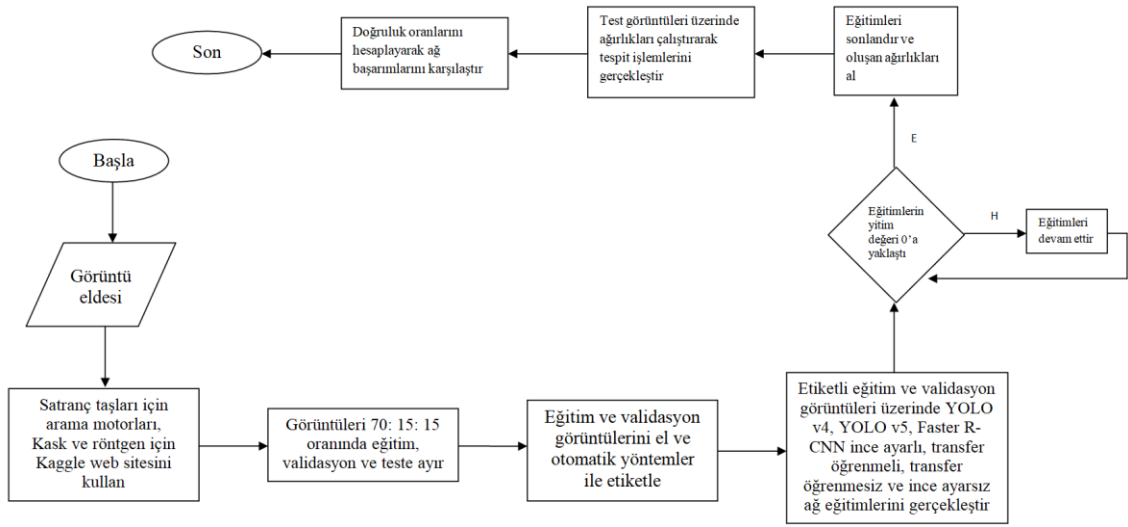


4. BÖLÜM

BULGULAR

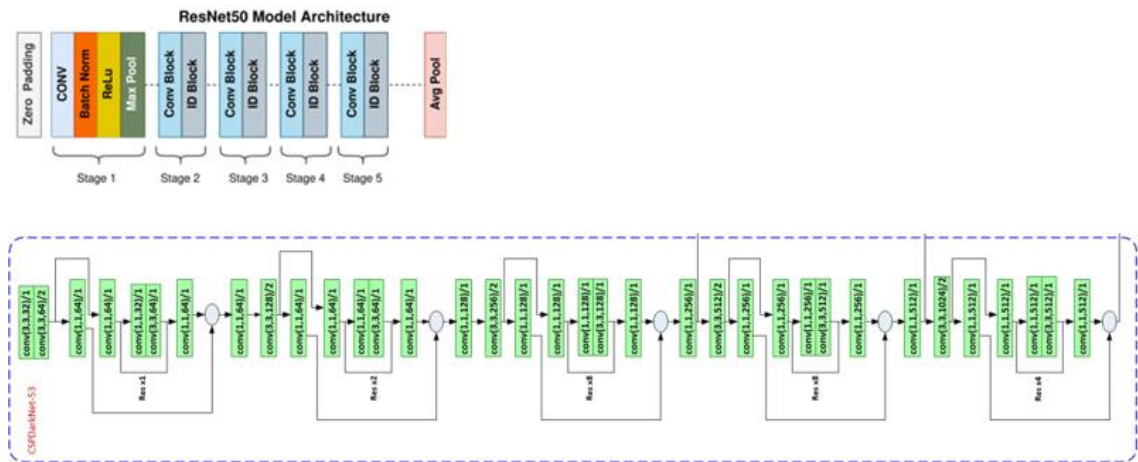
4.1. Araştırma Modeli

Şekil 16'da çalışmanın akış diyagramı verilmiştir. Her bir veri seti farklı yöntemler ile edinildikten sonra ön işlem aşaması olarak eğitim ve validasyona ayrılan bu görüntüler etiketlenmiştir. Akciğer görüntülerinin her birindeki akciğer adedi bir ve ekranın tamamını kapladığı için el ile etiketlemeye ihtiyaç duyulmayarak bu görüntülere tam ekran, otomatik etiketleme yöntemi uygulanmış; kask ve satranç veri setleri ise her görüntüde çoklu nesne içerdiklerinden etiketleme işlemi, el ile web tabanlı açık kaynak kodlu yazılım üzerinde gerçekleştirilmiştir. Etiketleme ile aslında derin öğrenme modelinin nereye bakması gerektiği ,görüntü üzerindeki nesnelerin sınıfı ve konumu belirlenerek, söylenmektedir. Daha sonra etiketli görüntüler ilgili ağlara girdi olarak verilerek eğitim işlemleri gerçekleştirilmiştir. Eğitimlerin başarıyla tamamlandığı, yitim değerinin 0'a yaklaşmasından anlaşılmıştır. Eğitimlerin sonucunda her bir ağın birbirinden bağımsız olarak çıktılacağı ağırlık dosyaları ile veri setlerinin kendi içlerinde belirlenmiş test görüntüleri üzerinde tespit işlemleri gerçekleştirilmiştir. Tespit işlemlerinin sonucu karşılaştırma metrikleri hesaplanarak denenen transfer öğrenmeli, transfer öğrenmesiz ve ince ayarlı, ince ayarsız ağ başarımları birbirleri arasında karşılaştırılmıştır.



Şekil 16. Çalışma Akış Diyagramı

Şekil 17’de sırasıyla Faster R-CNN ve YOLO v4, v5 mimarilerinin omurga bölümlerinde kullanılan ağlar verilmiştir. Bu ağlar her iki mimaride de özellik çıkarıcı olarak görev yapmaktadır yani sınıflandırma işlemi gerçekleştirememektedir. Transfer öğrenmeli Faster R-CNN için kullanılacak ResNet50 ağı, 1.000 sınıflı ImageNet veri setinde; transfer öğrenmeli YOLO v4, v5 için kullanılacak CSPDarknet53 ağı ise 80 sınıflı MS COCO veri setinde eğitilmiştir. Eğitim işlemleri sonucu oluşan ağırlık dosyaları, tekrar içlerinde yer aldıkları mimarilerin eğitiminde fakat bu sefer farklı sınıfları tespit etmek için kullanılmıştır.



Şekil 17. ResNet50 [73] ve CSPDarknet53

Metot kısmında detaylı olarak bahsedilen Faster R-CNN için RPN'ye giren veya YOLO için CSPDarknet53 ağından çıkarak boyun kısmına geçen özelliklerin çoğu, ImageNet ve MS COCO veri setlerinde yer alan nesne özellikleridir. Bu veri setlerinin içerdiği nesne sayısı, fazla olmasına rağmen her iki veri seti de 6 farklı satranç taşını, sağlıklı ve zatürreli akciğer x-ray görüntüsü ile kask ve kafa görüntüsünü içermemektedir. Bu yüzden transfer öğrenme tekniği kullanılarak gerçekleştirilen eğitimlerde, transfer öğrenmesize yani nesne özelliklerini öğrenmeyerek rasgele ağırlıklar ile eğitime başlamaya göre çok yüksek bir başarımlı artışı gerçekleşmemiştir. Yine de sayılan bu tespiti istenen nesnelerin her nesnede olduğu gibi kenar, leke, doku, kıvrım gibi genel özellikleri vardır ve bu bilgiler Şekil 17'de verilen ağların, önceden büyük veri setlerinde eğitilmesi sonucu oluşan ağırlık çıktılarındadır mevcuttur. Bu sebeple kendi ağ türünde yapılan deneylerin çoğunda transfer öğrenmeli eğitim sonuçları, transfer öğrenmesiz eğitimlere göre daha yüksek başarıya ulaşmıştır.

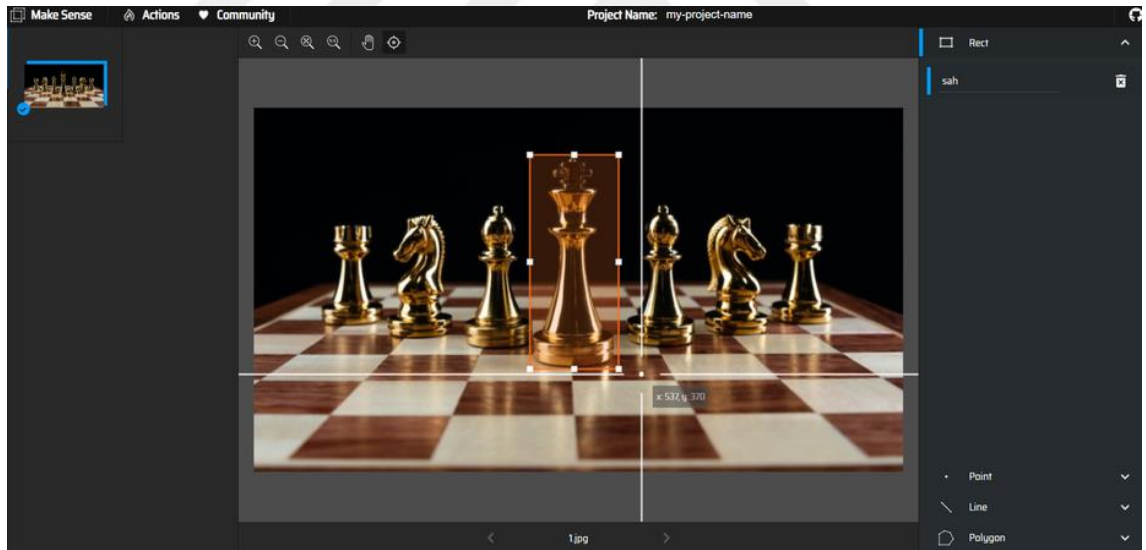
4.2. Deneyler

Bir sınıflandırıcı ağın doğru çalışması için üzerinde çalışacağı veri setinde hem negatif hem de pozitif örnekler olmalıdır. Tespiti istenen nesneleri barındıran görüntüler pozitif örnekler, arka plan görüntüsü ve tespiti istenmeyen nesneler de negatif örnekler olarak adlandırılmaktadır. Veri setlerinden; zatürrede röntgen görüntüsüne giren doktor malzemesi, kaskta inşaat sahasında kullanılan araç-gereçler ile arka plan görüntüsü ve satrançta tespiti istenmeyen gündelik nesneler ile arka plan görüntüsü mevcuttur. Zatürreli-sağlıklı akciğer röntgen görüntüleri 5.863 [74], kafa-kask nesnelerini içeren inşaat sahası görüntüleri 5.000 [75], satranç taşlarını içeren görüntüler ise 1.000 adettir. Satranç taşlarını içeren görüntüler arama motorlarından el ile elde edildiği için en az sayıdadır. Röntgen görüntüleri, ilerleyen aşamalarda gerçekleştirilen ağ eğitimlerinin sağlıklı bir şekilde yürütülebilmesi için iki sınıflı görüntü sayı dengesi sağlanarak 800 adete indirilmiştir. Tüm veri setlerinde; gerçekleştirilecek eğitimlere daha fazla ağırlık verilerek %70 eğitim, %15 validasyon ve %15 test ayrımı yapılmıştır.

Çalışma denetimli makine öğrenimi kapsamına girdiği için eğitilecek yani eğitim ve validasyona ayrılan görüntülerdeki nesneler bilgisayara tanıtılmıştır. Bu tanıtma işlemi için görüntüler etiketlenmiştir. Etiketlemede; satranç taşları, dosya adında normal veya zatürreli olduğu yazan tam ekran akciğer görüntüsü ve kask, kafa nesneleri için uzmana

ihtiyaç duyulmamıştır. Masaüstü uygulaması olan LabelImg etiketleme için ilk olarak denendiyse de uzun süren etiketleme işlemi için bilgisayarın kapanmaması gerekmektedir. Ancak bu programda export-import seçeneklerinin olmaması, yapılan işlerin gün sonunda kaybedilmesine ve bir sonraki gün kalınan yerden devam edilememesine yol açmıştır. Bu yüzden export-import işlemleri olan web tabanlı makesense.ai [76], çalışma kapsamında (iki çalışmada) tercih edilmiştir. Bu sayede gerçekleştirilen uygulamalar kayıpsız bir şekilde sürdürülmüştür. Etiketleme, tespiti yapılacak nesneleri ve bu nesnelerin konumlarını bilgisayara tanıtmadır. Etiketleme işlemi ile görüntüde yer alan tespiti yapılacak nesnelerin sınıfı ve konumu; eğitim esnasında kullanılmak üzere her bir görüntü için bir dosyaya (YOLO için txt, Faster R-CNN için xml) yazdırılmıştır.

Şekil 18’de eğitimde kullanılan satranç veri setindeki görüntülerin içeriğindeki tespiti istenen satranç taşı nesnelerinin etiketlenmesinden örnek bir görüntü sunulmuştur.



Şekil 18. Örnek etiketleme işlemi [76]

Şekil 18’den anlaşılabileceği üzere mümkün olan en dar (olması gerekeni en iyi ifade eden) dikdörtgen, nesne etrafına, dışarısında nesne uzantısı kalmayacak şekilde çizilmiştir. YOLO ve Faster R-CNN, eğitim için sınırlayıcı kutu olarak dikdörtgen şekline ihtiyaç duymaktadır. Program tarafından kullanıcıya sunulan her bir sınıf için farklı renkte olan dikdörtgen şekli ile hatanın minimize edilmesi amaçlanmaktadır. Görüntünün sağ kısmında yer alan diğer etiketleme seçenekleri; nokta, çizgi ve poligon olarak farklı amaçlara hizmet eden (bölütleyen UNet için poligon gibi) nesne tespitleyicileri içindir.

MS COCO ve PASCAL VOC gibi büyük veri setlerinin içerdikleri görüntülerin çeşitli olması gibi etiketlerinin formatları da birbirinden farklıdır. MS COCO veri setinin etiket dosya tipi txt, PASCAL VOC veri setininki ise xml'dir. YOLO ağ eğitimi PASCAL VOC görüntüleri kullanılsa bile bu görüntülere karşılık gelen etiket formatı PASCAL VOC'un xml'i yerine MS COCO'nun txt'si olmalıdır. Aynı şekilde Faster R-CNN için kullanılan etiket formatı PASCAL VOC formatındadır fakat ağın eğitimi için MS COCO görüntüleri de kullanılabilir. MS COCO görüntüleri kullanılsa bile bu görüntülere karşılık gelen etiket formatları PASCAL VOC'un xml'i olmalıdır. MS COCO veri formatı txt dosya tipi örneği Şekil 19'da, PASCAL VOC veri formatı xml dosya tipi örneği ise Şekil 20'de gösterilmiştir. Şekil 19 ile gösterilen örnek içeriğin ilk değeri, tespiti istenen nesnenin hangi rakamsal kategoride olduğunu; ikinci ve üçüncü değeri, sırasıyla sınırlayıcı kutunun; merkez x ve y noktalarını, son iki değeri ise genişliği ve yüksekliğini belirtmektedir. Faster R-CNN için oluşturulan etiket ise Şekil 20'de gösterildiği gibi PASCAL VOC veri seti formatındadır ve dosya tipi xml'dir. "name" anahtar kelimesi ile hangi nesnenin tespit edildiği belirtilmiş, "xmin" nesneye çizilen kutunun; sol alt köşesinin yataydaki, "xmax" sağ alt köşesinin yataydaki, "ymin" sol üst köşesinin düşeydeki, "ymax" ise sağ üst köşesinin düşeydeki ekran konumunu göstermektedir.

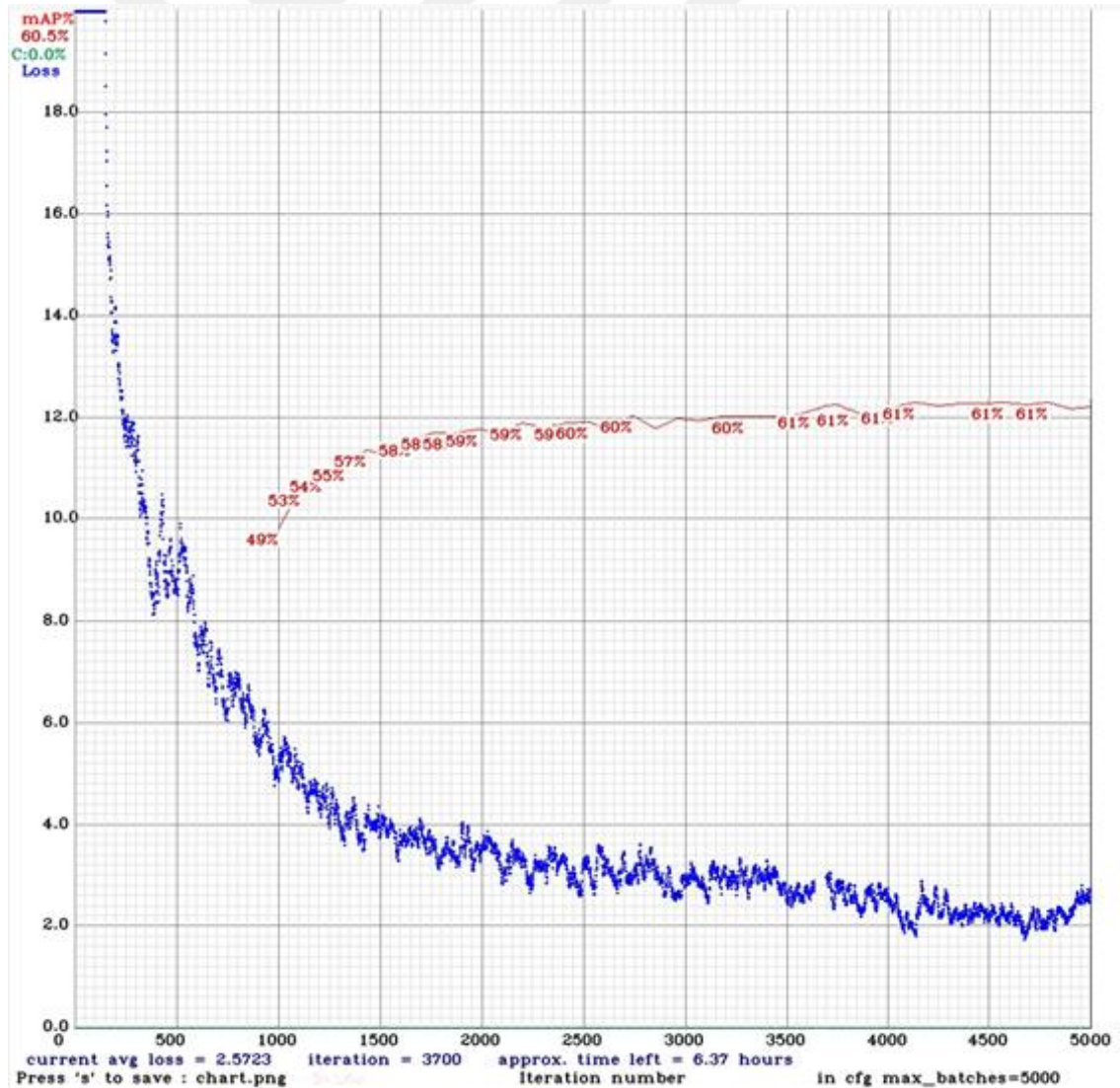
```
0 0.674254 0.069859 0.263736 0.139717
0 0.921900 0.073783 0.156200 0.147566
```

Şekil 19. Örnek txt etiketi (MS COCO)

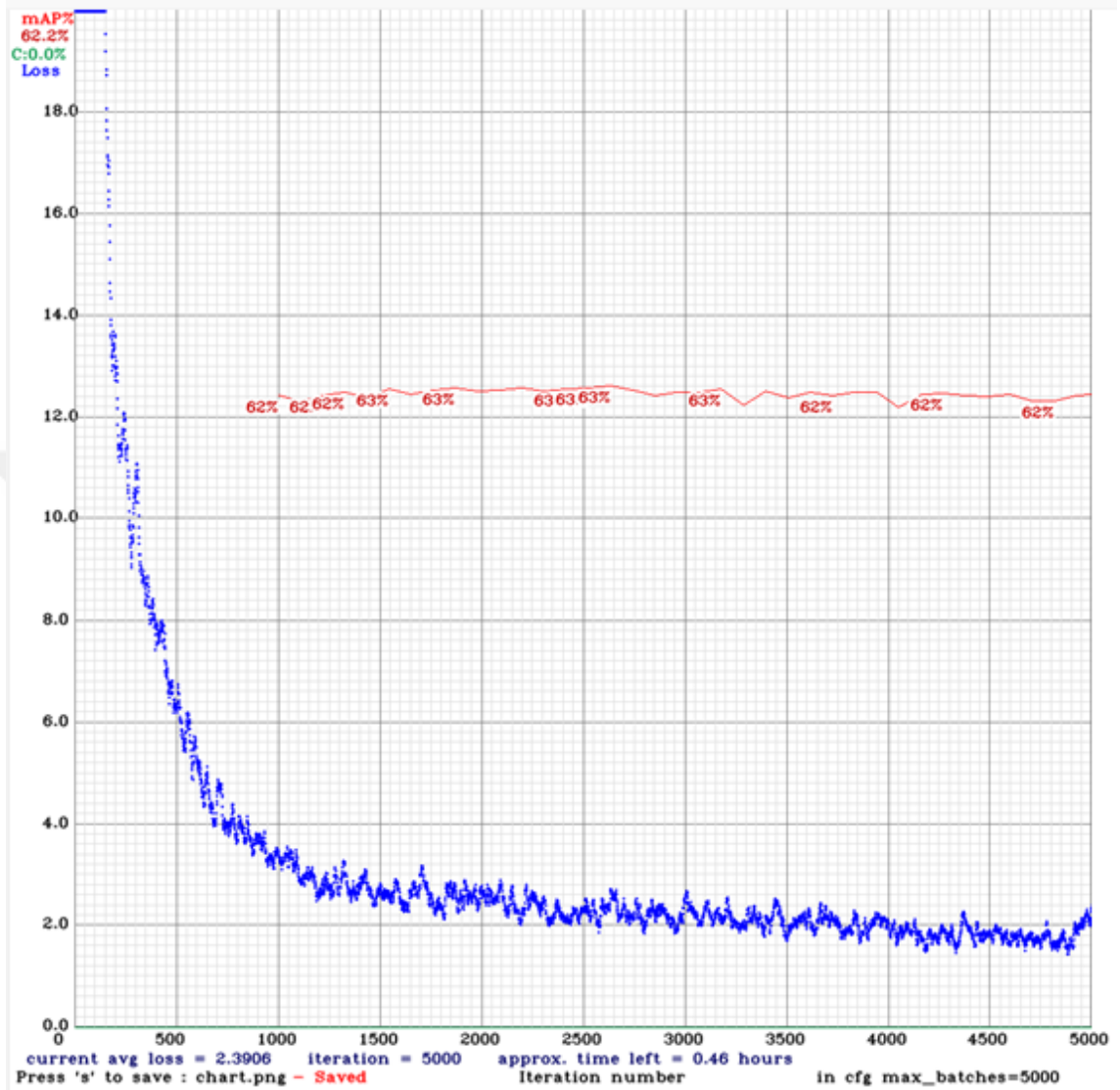
```
<object>
  <name>asfalt</name>
  <pose>Unspecified</pose>
  <truncated>0</truncated>
  <difficult>0</difficult>
  - <bndbox>
    <xmin>20.73388429752066</xmin>
    <ymin>156.56198347107437</ymin>
    <xmax>56.70082644628099</xmax>
    <ymax>253.03801652892562</ymax>
  </bndbox>
</object>
```

Şekil 20. Örnek xml etiketi (PASCAL VOC)

Görüntüler etiketlendikten sonra ağ eğitimlerine hazır hale gelmişlerdir. Ağ eğitimlerinde kullanılan kişisel bilgisayar AMD Ryzen 5 işlemci, 32 GB RAM, 12 GB ekran kartı özelliklerine sahiptir ve tüm deneyler eşit şartlar altında gerçekleştirilmiştir. Şekil 21 ve 22’de kask veri seti üzerinde çalıştırılan sırasıyla transfer öğrenmesiz ve transfer öğrenmeli YOLO v4 eğitim grafikleri verilmiştir. Grafiklerden görüleceği üzere 5.000 epok boyunca; mavi noktayla gösterilen yitim değeri sürekli olarak düşerken kırmızı çizgiyle gösterilen mAP değeri artmaktadır. Bu eğitimin başarılı bir şekilde tamamlandığını ve 5.000 epokluk eğitimin Darknet çerçeve yazılımı kullanan YOLO v4 için yeterli olduğunu göstermektedir.



Şekil 21. Transfer Öğrenmesiz Darknet YOLO v4 Kask Eğitim Grafiği

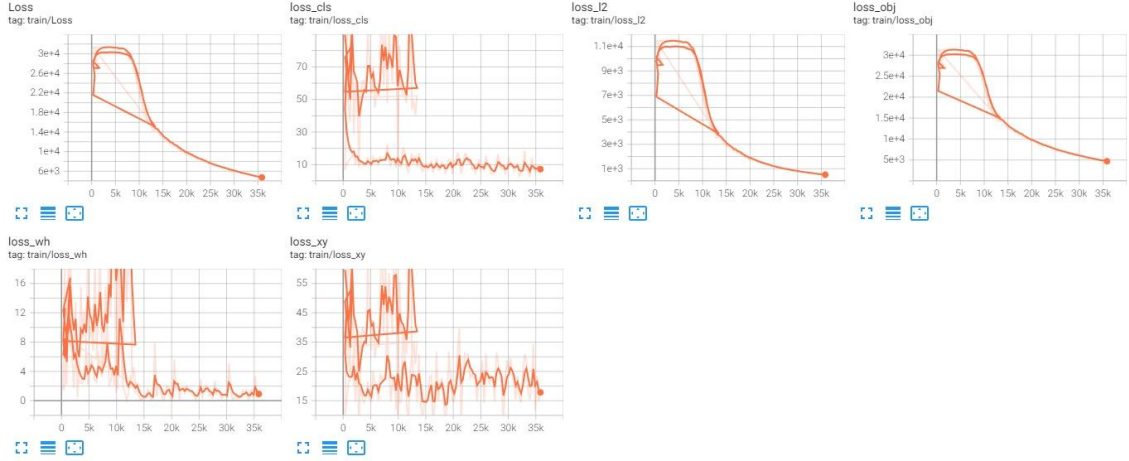


Şekil 22. Transfer Öğrenmeli Darknet YOLO v4 Kask Eğitim Grafiği

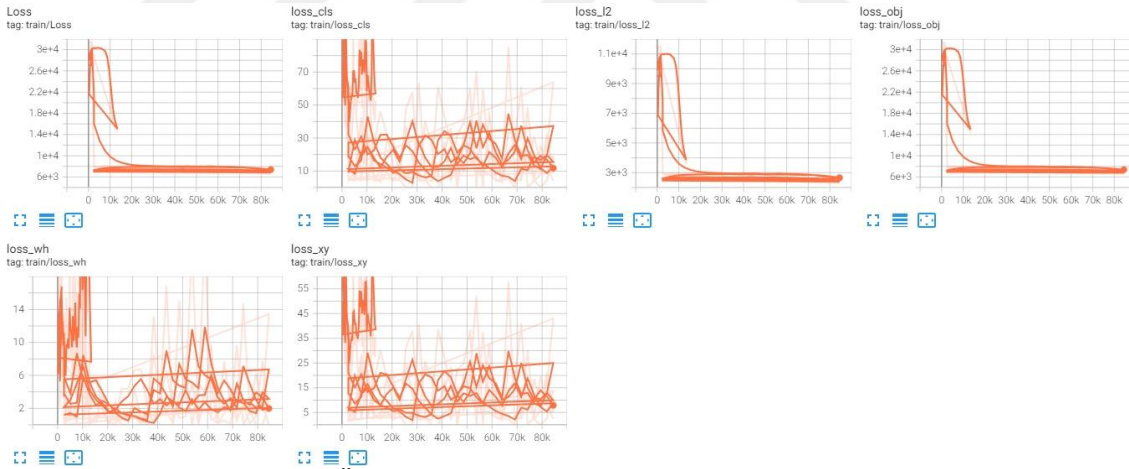
Şekil 23-26'da ise YOLO v4'ün; diğer veri setleri olan zatürre ve satrançtaki çalışma grafikleri verilmiştir. Grafiklerde “loss_cls” ile verilen yitim değeri genelde kararsız bir performans sergilemiştir. Eğitim boyunca yitim değerinin bazen 0'a yaklaşması (1 civarı) eğitimin başarılı geçtiğine işaret etmektedir. Performans metriklerinin oluşması için gerçekleştirilen tespit işlemleri de bu durumu kanıtlar niteliktedir.

Sadece PyTorch [77] çerçeve yazılımı kullanan YOLO v4 ve v5'te olan “loss_obj” ile gösterilen değer, nesnesizlik skoru olarak önerilen bölgede nesnenin olamama ihtimalini, sınırlayıcı kutu parametreleri ile alakalı olan “loss_wh” ve “loss_xy” ise YOLO v4

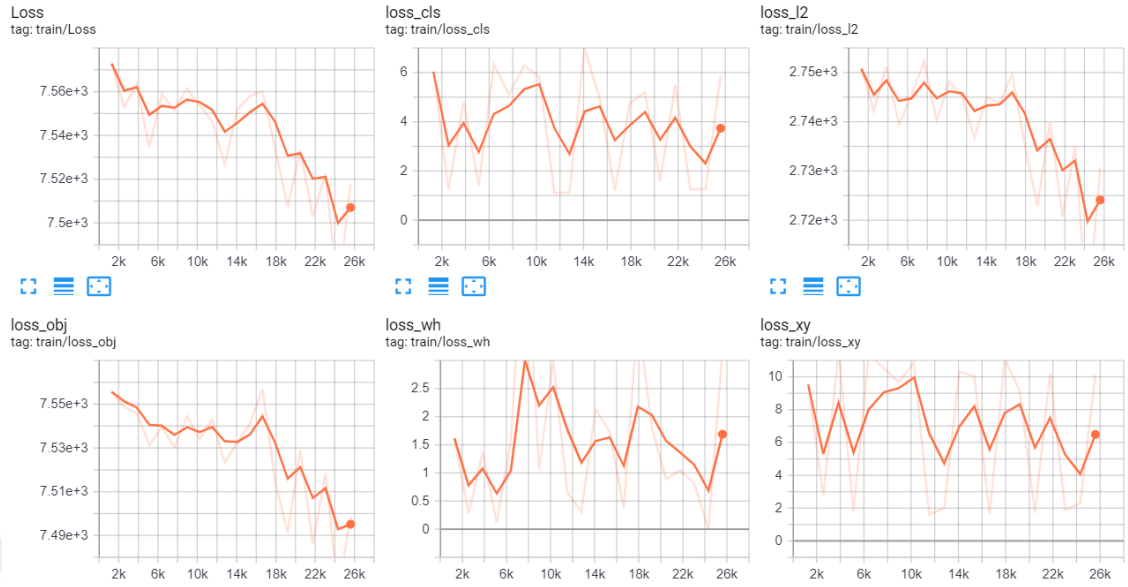
için CIoU yitim, YOLO v5 için GIoU yitim fonksiyonu değerlerini göstermektedir. Tüm grafiklerdeki bu değerler de “loss_cls” gibi eğitim boyunca 0’a yaklaşarak eğitimlerin başarılı bir şekilde gerçekleştiğini göstermektedir.



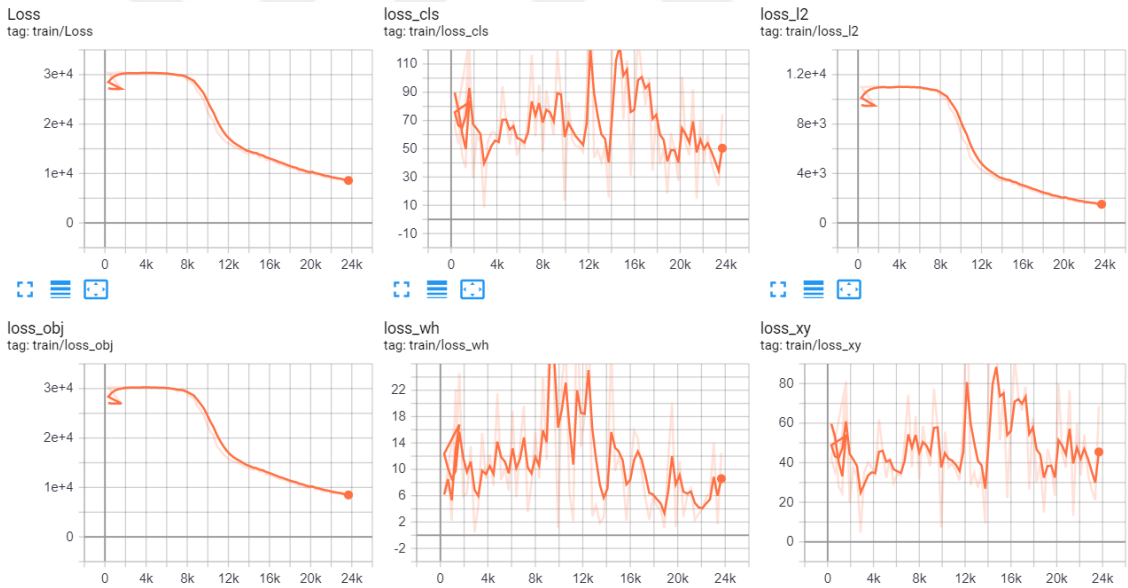
Şekil 23. Transfer Öğrenmesiz PyTorch YOLO v4 Satranç Eğitim Grafiği



Şekil 24. Transfer Öğrenmeli PyTorch YOLO v4 Satranç Eğitim Grafiği



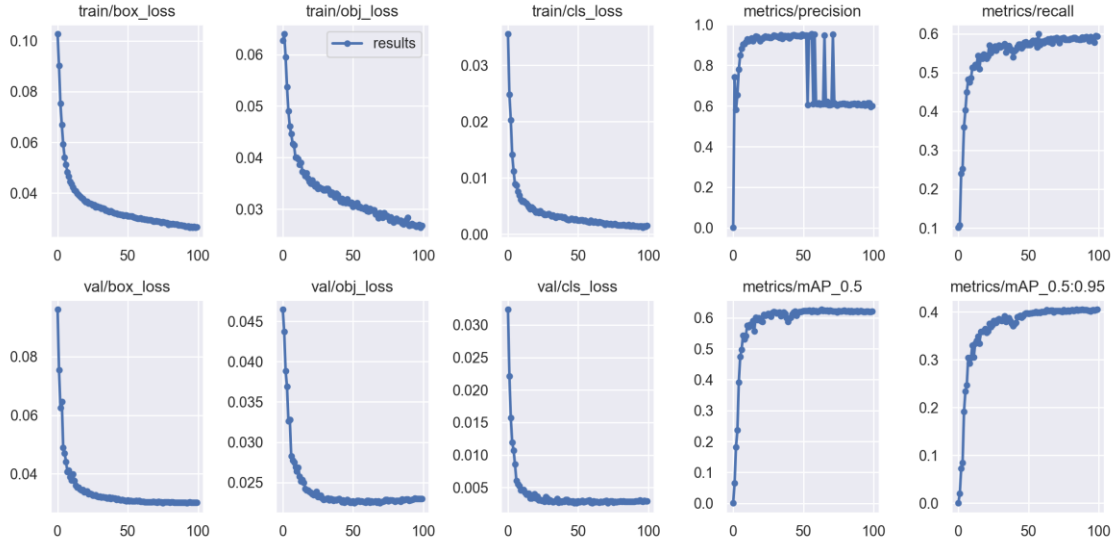
Şekil 25. Transfer Öğrenmesiz PyTorch YOLO v4 Zatürre Eğitim Grafiği



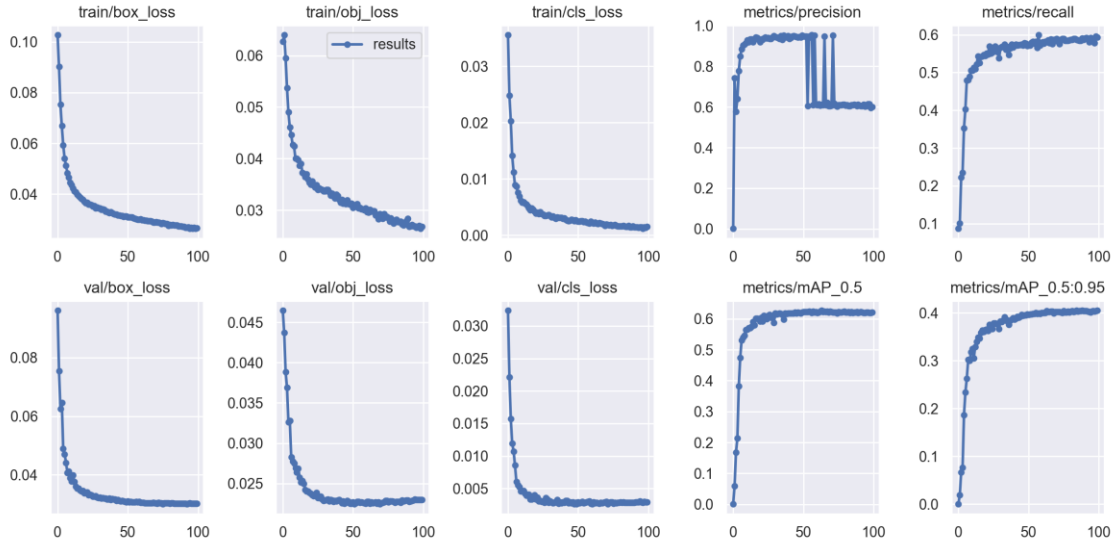
Şekil 26. Transfer Öğrenmeli PyTorch YOLO v4 Zatürre Eğitim Grafiği

Şekil 27-32’de 3 farklı veri setinde, iki farklı yöntemin denendiği YOLO v5 eğitim grafikleri sunulmuştur. Eğitimin başarısını gösteren “train/cls_loss” değeri tüm grafiklerde 0’a yaklaşmıştır. Bu durum 100 epokun PyTorch çerçeve yazılımını kullanan YOLO v5 için yeterli olduğunu göstermektedir. PyTorch YOLO v4’te olduğu gibi kutu ve nesne-sizlik yitim değerleri 0’a yaklaşmıştır. Ayrıca tüm YOLO v5 grafiklerinde “val/cls_loss” değeri, “train/cls_loss” değerinden daha düşük çıkarak ağırlık transfer yap-

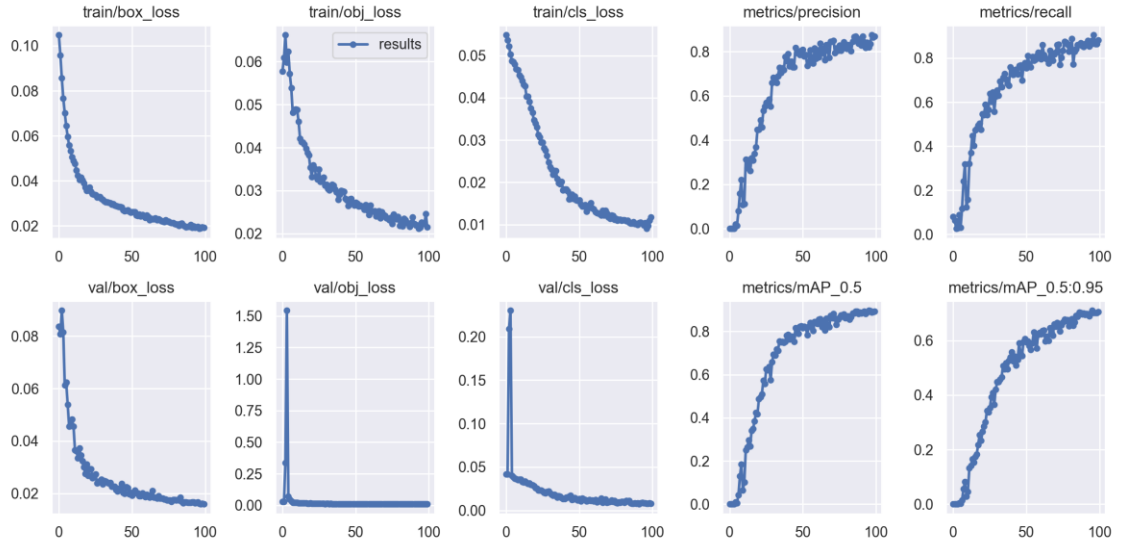
madığı yani daha önce görmediği veriler üzerinde başarılı tahminler yapacağı sonucu çıkarılmaktadır.



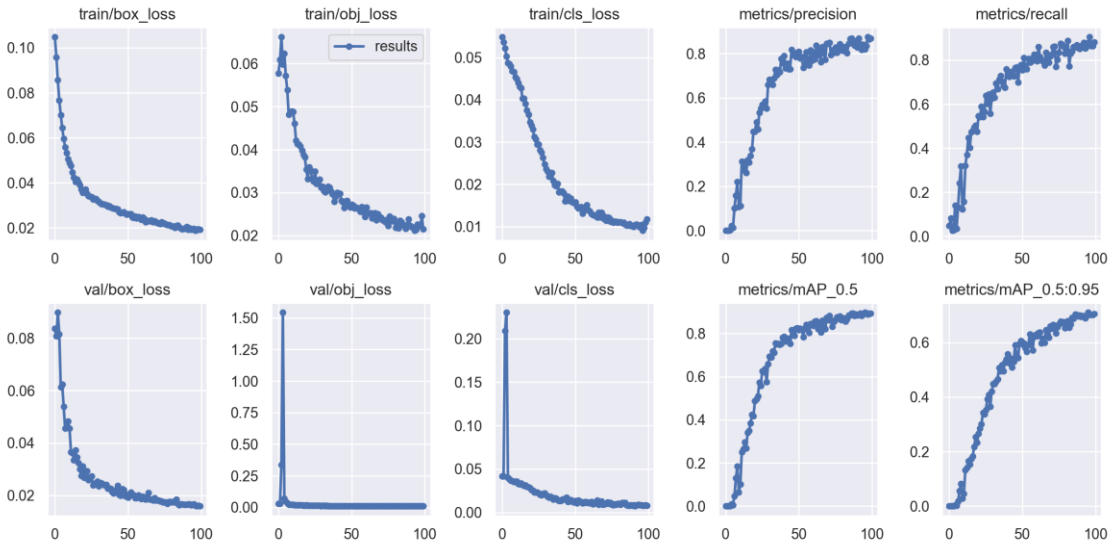
Şekil 27. Transfer Öğrenmesiz YOLO v5 Kask Eğitim Grafiği



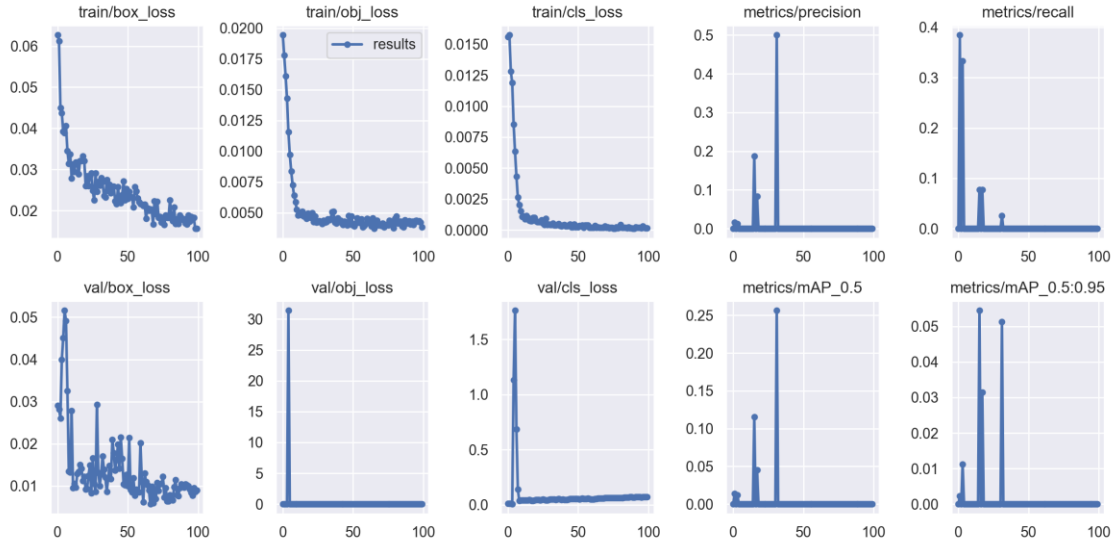
Şekil 28. Transfer Öğrenmeli YOLO v5 Kask Eğitim Grafiği



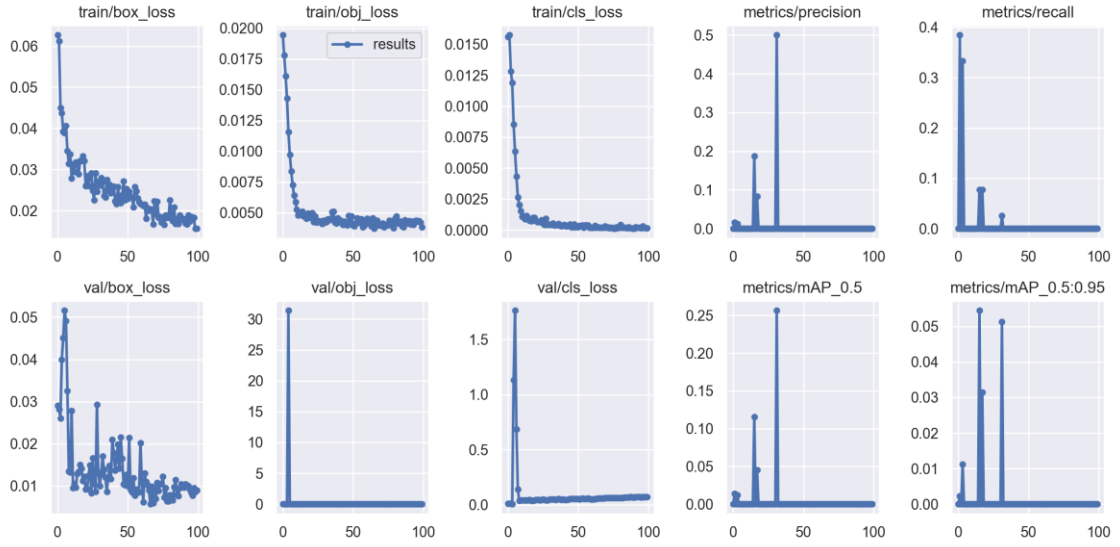
Şekil 29. Transfer Öğrenmesiz YOLO v5 Satranç Eğitim Grafiği



Şekil 30. Transfer Öğrenmeli YOLO v5 Satranç Eğitim Grafiği

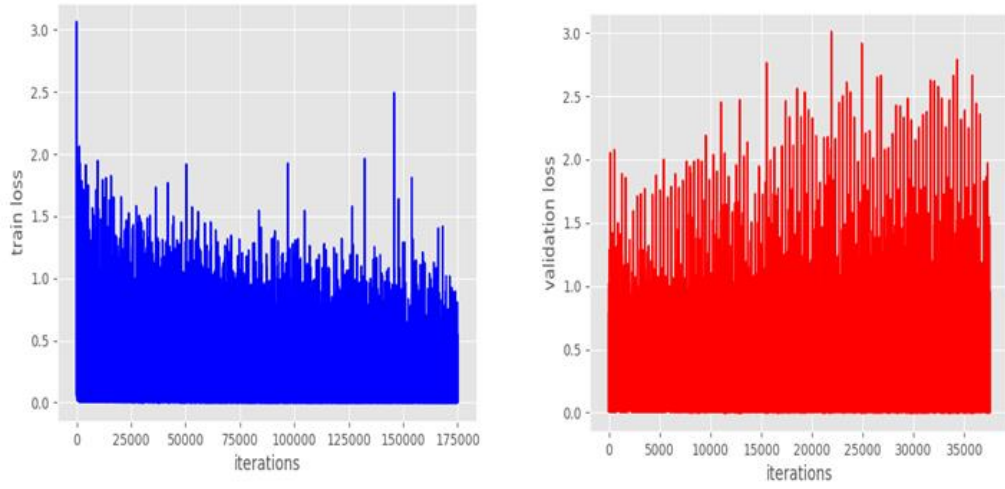


Şekil 31. Transfer Öğrenmesiz YOLO v5 Zatürre Eğitim Grafiği

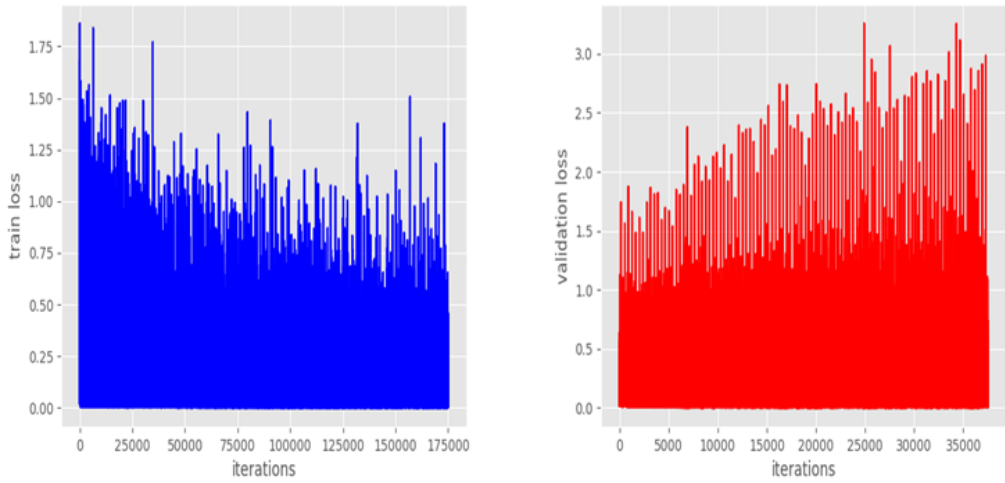


Şekil 32. Transfer Öğrenmeli YOLO v5 Zatürre Eğitim Grafiği

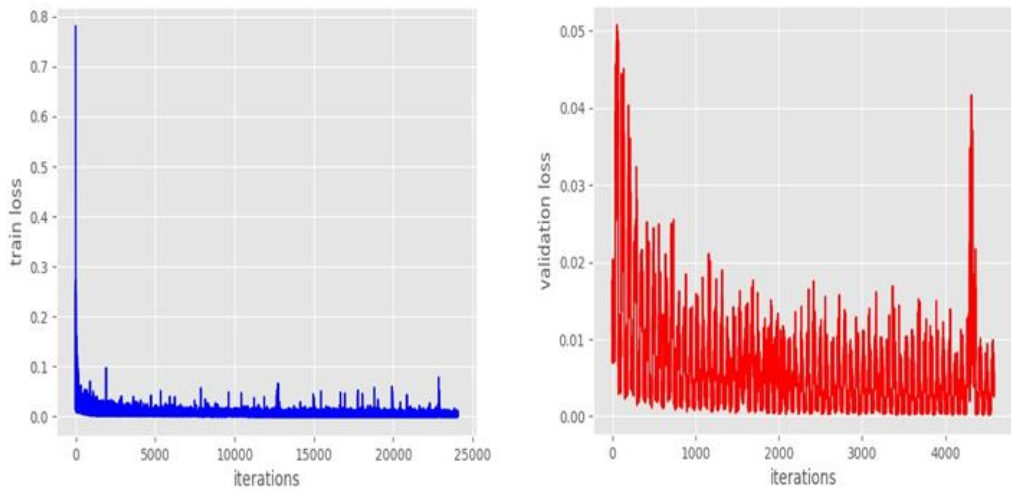
Şekil 33-38’de 6 farklı Faster R-CNN eğitim grafiği verilmiştir. Bu grafiklerde eğitim boyunca oluşan eğitim ve validasyon yitim değerleri verilmiştir. Mavi ile gösterilen eğitim yitim değeri tüm grafiklerde 0’a yaklaştığı için eğitimlerin başarılı bir şekilde sonlandığı ve seçilen epok değerinin ağ için yeterli olduğu belirtilebilir [69].



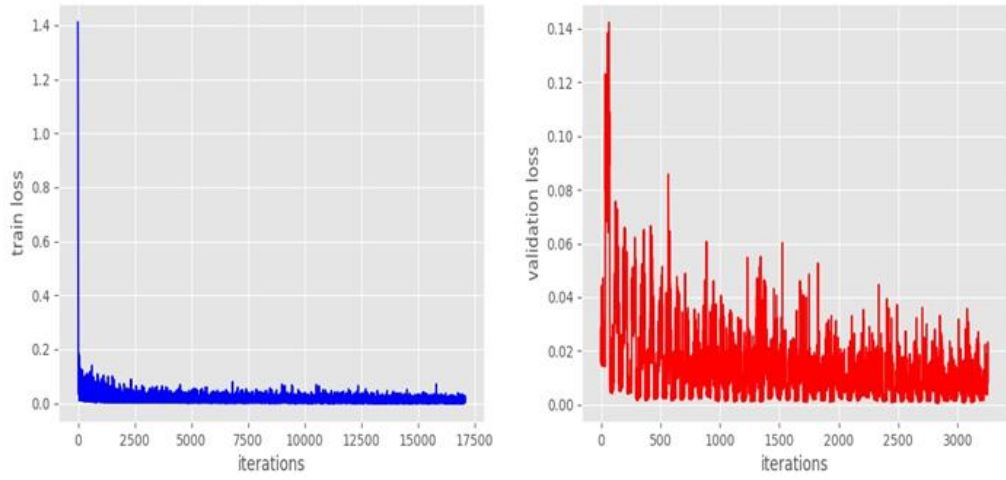
Şekil 33. Transfer Öğrenmesiz Faster R-CNN Kask Eğitim Grafiği



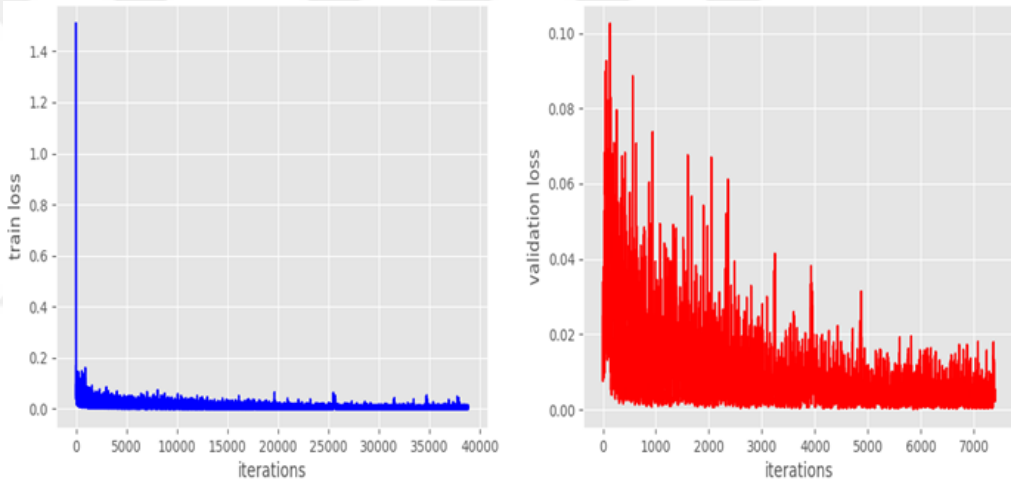
Şekil 34. Transfer Öğrenmeli Faster R-CNN Kask Eğitim Grafiği



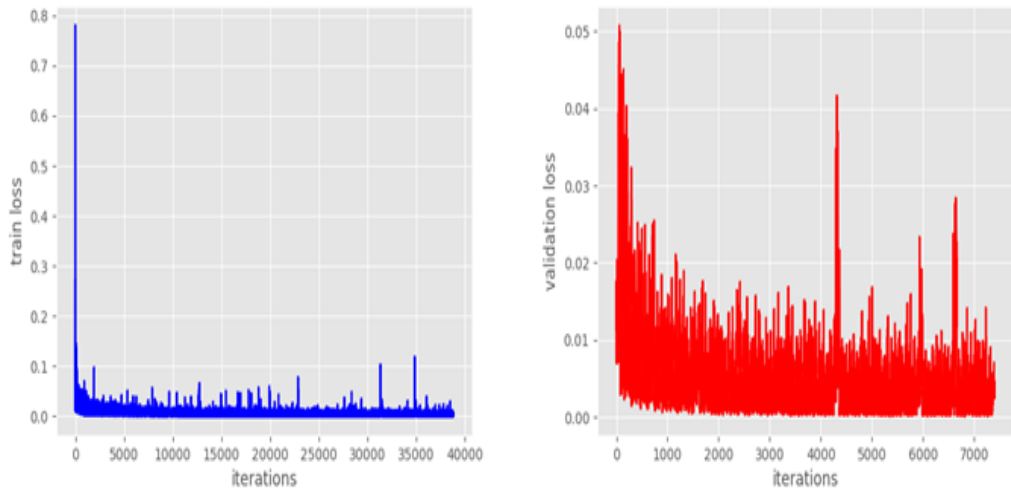
Şekil 35. Transfer Öğrenmesiz Faster R-CNN Satranç Eğitim Grafiği



Şekil 36. Transfer Öğrenmeli Faster R-CNN Satranç Eğitim Grafiği



Şekil 37. Transfer Öğrenmesiz Faster R-CNN Zatürre Eğitim Grafiği



Şekil 38. Transfer Öğrenmeli Faster R-CNN Zatürre Eğitim Grafiği

Faster R-CNN için PyTorch, eğitim ve validasyon yitimini birlikte yazdırırken Darknet kullanan YOLO v4, eğitim yitimiyle birlikte eğitim boyunca değişen mAP değerini de yazdırmaktadır. mAP orta ortalama kesinlik değeridir ve tartışma kısmında değinilecek olan sınıfların “tp”, “fp” ve “fn” sayımlarıyla ortaya çıkan Kesinlik ve Hassaslık eğrilerinin kesişimlerinin alanıdır. PyTorch YOLO v4 ve YOLO v5 eğitim grafiklerini çizdirmede sırasıyla Tensorboard [78] ve seaborn [79] paketleri kullanıldığından sınıf ve kutu parametreleri için olan yitim değerlerinin tümüne birlikte yer verilmiştir.



5. BÖLÜM

TARTIŞMA-SONUÇ ve ÖNERİLER

5.1. Tartışma

Görüntüden anlam çıkaran ESA tabanlı derin öğrenme ağlarının oldukça yaygın kullanım alanına sahip olmasının birincil unsuru transfer öğrenmedir. Transfer öğrenme sayesinde düşük donanım özelliklerinde küçük veri setini kullanabilme mümkün olmaktadır. Yazı ve ses verisinde olduğu gibi görüntü verisinde de çalışabilen transfer öğrenme tekniği sayesinde güçlü özellikler, ağlar arasında transfer edilebilmektedir.

Görüntü verilerini ve nesne tespiti teknolojisini halihazırda yaşantımızın her alanında sıklıkla kullanmaktayız. Özellikle tarım, güvenlik, sağlık ve şehir planlama gibi alanlarda nesne tespiti yıllardır hayatımızı kolaylaştıran, daha iyi bir dünya oluşturma çabamıza katkı sunan, veriden hareketle anlam çıkarımı yapan temel teknolojilerden birisidir. Nesne tespiti teknolojisi gelecekte insanlığa çok daha önemli faydalar sağlayacak bir potansiyele sahiptir. Örneğin, nesne tespiti ile görme engelli bireylere yardımcı, görülen nesneyi sesle söyleyen cihaz geliştirilebilir veya savunma sanayide düşman hedeflerin daha kolay tespit edilerek vurulmasını kolaylaştıran ekipmanların yapılması mümkün olabilir.

İnsan görmesinden ilham alınarak geliştirilen bilgisayarlı görü teknolojisi, veriden anlam çıkarma eylemini ESA tabanlı derin öğrenme ağı kullanarak yapmaktadır. Az katmanlı ESA tabanlı derin öğrenme ağları daha düşük, çok katmanlı ESA tabanlı derin öğrenme ağları ise daha yüksek seviyede anlamlı özellik çıkarmaktadır. Yüksek seviye özellik çıktıkça da daha başarılı tespitler gerçekleştirilmektedir. Daha başarılı tespitler sonucunda gündelik hayatımızda kullandığımız teknolojiler dönüşmekte ve bu kapsam-

da düşman hedefi, sulama, hastalık tespiti, güvenlik için tespit işleri gerçekleştirilmektedir.

Görüntüde derinde yer alan küçük nesnelerin tespitinden bağımsız olarak küçük veri setiyle derin öğrenmeyi çalıştırma ve ağın doğruluğunu artırma için uygulanan transfer öğrenme tekniği ile çalıştırılan ağların çoğunda, bu tekniğin kullanılmadığı ağlara göre başarı artışı sağlanmıştır.

Bu çalışmada kullanılan YOLO v5, Faster R-CNN ve YOLO v4 ağlarından sadece YOLO v4 ile derindeki küçük nesneler tespit edilmiştir. Faster R-CNN'de çözünürlük artırımı ile ağa kabul edilen görüntüler belirlenmiş, YOLO v5'te kanal genişliği ve derinliği artırılmış fakat derin nesne tespiti bağlamında sonuç alınamamıştır. YOLO v4 mimarisinin hiper parametrelerinden olan ızgara sayısını artırma probleme çözüm olmuştur.

Literatürde [59] [71] derindeki küçük nesnelerin tespiti uygulamaları, genelde mimari değişikliği şeklinde yapılmaktadır. [59]'da ağa; özellik birleşimi katmanı eklemişler, [71]'da ise Gauss karıştırma modeli uygulamışlardır. Her iki çalışmadaki uygulamalar, ağı hesap yükünü artırmıştır.

Mevcut çalışmanın konu alanı ile ilişkili literatürde gerçekleştirilen yayınlar incelendiğinde farklı türde ince ayar uyguladıkça doğruluk değerinin arttığı görülmektedir. Cilt hastalığı görüntüleri sınıflandıran tez çalışmasında [80] en yüksek f1 skor 0.987 bulunmuştur. Çalışmada DenseNet, Xception, MobileNet tercih edilmiş ve özellikle katman tabanlı ince ayarın etkinliğine bakılmıştır. [81] çalışmasında ise iki malzeme arasındaki mesafe değeri ölçülerek çıkan mesafe değeri belirli bir değerin üzerindeyse makinede hata var denmiştir, bu değerin ölçülmesinde görüntüde yer alan malzemeler kullanılmıştır. SSD kullanılan çalışmada 500. epok sonucunda mAP değeri 0.729 bulunmuştur. [82] çalışması sağlık alanında gerçekleştirilmiş ve transfer öğrenmeli ağ ile meme kanseri tespiti yapılmıştır. Araştırmacılar VGG16 ve VGG19 ağları ile en yüksek f1 skor değerini 0.9774 bulmuşlardır. Cinsiyet ve kişi sınıflandırmalarını ayrı ayrı yapan [52] çalışmasında ise cinsiyet sınıflandırmasını ResNet50 ağı ile yaparak kişi sınıflandırmasına göre en yüksek doğruluğu 0.9158 olarak bulmuşlardır. Son olarak 10 katlı çapraz doğrulama uygulayarak bu değeri 0.9641'e getirmişlerdir. Sağlık alanında yapılan ikinci bir çalışmada ise yazarlar, göğüs röntgen görüntülerinden Covid-19 tespiti yapmak için

ESA'nın sonuna DVM ekleyerek 0.856 f1 skor elde etmişlerdir [83]. Yine sağlık alanında yapılan bir tez çalışmasında ise transfer öğrenmeli ağlar ile Covid-19 tespiti yapılmıştır. Covid-19 teşhisinde kullanılan tomografi cihazının x-ray cihazına göre daha kısa sürede sonuç vermesine rağmen daha yüksek seviyede radyasyon yaydığından bahsedilen bu çalışmada en yüksek f1 skor %96.7 olarak bulunmuştur [84].

Küçük nesneleri tespitite ızgara yapısını kullanan YOLO ve ölçek, çerçeve oranı parametrelerini kullanan Faster R-CNN tercih edilmiştir. Yapılan deneylerde YOLO v5'in büyük veri setlerinde daha iyi olduğu anlaşılmıştır [85]. Çünkü Faster R-CNN'in YOLO v5'e göre daha az katmanlı yapısı gereği az sayıda görüntüyle işlem yaptığından başarısı daha yüksek çıkmaktadır. YOLO algoritmaları arasında ise YOLO v5'in YOLO v4'e üstün geldiği birçok çalışmada görülmüştür [10] [86]. Veri setlerinin YOLO'nun ızgara yapısına uymayan yan yana nesneleri fazla sayıda içermemesi bu başarıyı YOLO'ya getirmiştir. Çalışmada diğer veri setlerine göre en fazla yan yana nesne, kask veri setindedir fakat bu veri seti üzerindeki eğitimlerde Faster R-CNN yerine YOLO v5 birinci gelmiştir. Çünkü bu veri setindeki görüntülerin çoğunda, yan yana kafa ve kask nesneleri yoktur, ayrıca veri setindeki görüntü sayısı çoktur. Göğüs röntgen görüntü sayısı az olduğundan burada beklenildiği gibi YOLO'ya göre daha az katmanlı Faster R-CNN daha başarılı olmuştur. Faster R-CNN'in çalışmada kullanılan diğer ağlara göre bir dezavantajı, tespit aşamasında çakışan kutulardır. Problem, NMS eşik değerinin düşürülmesiyle aşılmış fakat bu işlem, yanlış tespit sayısını veren "yanlış pozitifin" artmasına neden olmuştur.

Şekil 42'de çalışmada gerçekleştirilen deneylerin f1 skorlarına baktığımızda YOLO v5'in diğer ağlara üstünlüğü gözükmemektedir. YOLO v5 mimarisinin kompakt yapısı YOLO v5'e bu başarıyı getirmiştir. Özellikle YOLO v5'te kullanılan Odak katmanının YOLO v4'teki 4 evrişim katmanı yerine geçmesi, YOLO v4'te sadece regresyon problemlerinde kullanılabilen mozaik veri çoklamanın YOLO v5 ile birlikte sınıflandırmada da kullanılabilmesi ve kendi kendine öğrenen sınırlayıcı kutu çapaları bu başarının temel yapı taşlarını oluşturmuştur.

YOLO v4'ün geliştiricisi tarafından oluşturulan ve eğitim stratejilerini geliştirmeye yarayan birtakım algoritmalar vardır. Bu algoritmalar fotometrik ve geometrik veri çoklama metotlarını içererek Bag Of Freebies adı altında toplanmıştır. Algoritmalarından biri

olan Kes ve Karıştır, nesneleri sınırlarından kırpmaktadır. Şekil 39 ile 4 adet görüntüye Kes ve Karıştır uygulanması ve ağa dörderli bir şekilde alınma aşaması olan mozaik veri çoklama gösterilmektedir. Mozaik veri çoklama sayesinde ağı; tek seferde birçok görüntüyü görmesi sağlanarak genelleştirme yeteneği artırılmaktadır.



Şekil 39. Sınıflandırmada kullanılan YOLO v5 mozaik veri çoklama işlemi

Transfer öğrenme ile büyük veri setlerinde eğitilen ağların ağırlık çıktıları nispeten küçük veri setleri olan kask, x-ray ve satranç görüntülerinde eğilecek ağların başlangıç noktaları olarak kullanılmıştır. Bu sayede ağlar günümüzde her yerde olan nesne özelliklerini öğrenerek eğitime başlamıştır. Ağların kendi içlerindeki evrişim ve havuzlama katmanlarından nesne özellikleri çıkarılarak bu özellikler, önceden öğrenilen özelliklerin üstüne eklenmiştir, ağların son aşamasında sınıflandırılacak nesne sayısına ayarlanan tam bağlı katmanın nöron sayısınca nesne sınıflandırılmıştır. Kask ve x-ray görüntüleri için bu sayı iki nesne sınıflandırması yapan ağ için ikiye, satranç içinse taş sayısı olan 6'ya ayarlanmıştır. Son olarak bu ağlardan alınan çıktılar, tespit işlemlerinin yapılabilmesi için her veri setinin teste ayrılan kısmında ayrı ayrı çalıştırılmıştır.



Şekil 40. Örnek bir görüntü üzerinde karşılaştırma metrik değerlerini oluşturan doğru pozitif, yanlış pozitif ve yanlış negatif değerlerinin gösterimi

Örnek bir tarla tespit problemi için eğitilen ağ sonucu oluşan çıktı ağırlıkları, tarla test görüntülerinde kullanıldığında Şekil 40'dakine benzer bir tablo ile karşılaşılmaktadır. Şekil 40'da mavi ile gösterilen kare, görüntüde tespiti istenen tarlanın ağ tarafından doğru tespit edildiğini kullanıcıya söyleyerek genel toplamdaki doğru pozitif değerini bir artırmakta, sarı kare, tarla yerine yol nesnesi tespit edildiğini, kırmızı kare ise tespiti istenen tarla nesnesinin bulunamayışını göstermektedir ve normalde oluşmamıştır, örneğin daha iyi anlaşılabilmesi için konmuştur. Yani görüntüde tespiti istenen tarla var fakat ağ tarafından tespit edilemeyişin ifadesi, normalde oluşmayan kırmızı karedir. Sarı kare sayısı, genel toplamdaki yanlış pozitif değerini, kırmızı kare ise genel toplamdaki yanlış negatif değerini bir artırmaktadır.

Şekil 40'daki bir test görüntüsü için hesaplanan “doğru pozitif”, “yanlış pozitif” ve “yanlış negatif” değerleri birer adet çıkmıştır. Normalde bu değerler, bir görüntüde daha fazla sayıda çıkabilmektedir. Ayrıca bu tespitler test görüntülerinin tümü için yapılmak-

tadır. Test görüntülerinin tamamı için hesaplanan bu değerlerin toplanmasıyla Eşitlik 1-4'te verilen Doğruluk, Kesinlik, Duyarlılık ve F1 Skor oranları oluşturulmaktadır.

$$\text{Doğruluk} = (tp + tn) / (tp + tn + fp + fn) \quad (1)$$

$$\text{Kesinlik} = tp / (tp + fp) \quad (2)$$

$$\text{Duyarlılık} = tp / (tp + fn) \quad (3)$$

$$\text{F1 Skor} = 2 * \text{Kesinlik} * \text{Duyarlılık} / (\text{Kesinlik} + \text{Duyarlılık}) \quad (4)$$

Eşitlik 3'te verilen "tn" değeri görüntüde olmayan nesneye kutu çizilmemesi manasına gelmektedir ve her zaman matematikte etkisiz eleman olan 0 olarak alınmaktadır. "Yanlış pozitif" ve "yanlış negatif" değerlerinin dengede olmadığı durumlarda "Doğruluk" yerine "F1 Skor" kullanımı daha uygundur çünkü F1 skor hesabında bu değerlerin kullanıldığı oranların harmonik ortalamasının alımı söz konusudur.

5.2. Sonuç

Bu çalışmanın amaçlarından biri kafa, kask; sağlıklı, zatürreli göğüs röntgen; satranç taşları görüntülerini içeren veri setlerini kullanarak yüksek doğrulukta nesne tespiti yapmaktır. Bu kapsamda ESA tabanlı derin öğrenme ağlarından YOLO v4, YOLO v5 ve Faster R-CNN ağlarıyla deneyler gerçekleştirilmiş, deneyler sonucunda ağlar arasındaki f1 skor oranları karşılaştırılmıştır. Çalışmada ayrıca transfer öğrenme ve ince ayarın etkinliği incelenmiştir. Deneylerde transfer öğrenmeli ve transfer öğrenmesiz eğitimler gerçekleştirilerek transfer öğrenmenin doğruluğu artırıcı bir etkisinin olup olmadığına bakılmıştır. Derindeki küçük nesneleri tespit etmek için uygulanan ince ayar yöntemi ise gene tüm ağlar için denenmiş fakat sadece YOLO v4'ten sonuç alınabilmektedir. Bunun nedeni olarak YOLO v5'in ızgara sayısını artırmaya izin vermeyişi ve Faster R-CNN'in ızgara yapısını kullanmamasıdır.

Bu tez çalışmasında süre kısıtı olmadığından eğitimi kısaltan metot olarak katman tabanlı ince ayara gerek duyulmamış ancak derindeki küçük nesne tespiti aşamasında hiper parametre optimizasyonu ile ince ayar işlemi gerçekleştirilmiştir. Transfer öğrenme uygulaması ise ağırlık transferi şeklinde gerçekleştirilmiştir. Yani önceden büyük veri setinde eğitilen ağın ağırlık çıktısı, mevcut problemlerdeki ağların eğitiminde kullanıl-

mıştır. Bu çalışma önceki paragrafta değinilen amaçlardan biri olan yüksek doğruluğu, Faster R-CNN'in röntgen görüntü veri setinden aldığı %99 ve YOLO v5'in yüksek skorları ile sağlamıştır. Bu başarının ana etmenlerinden biri de Faster R-CNN'in omurga ağında kullanılan ResNet50'nin katman sayısının az olmasıdır. Röntgen görüntü sayısı da az olarak katman sayısına uyum gösterdiği için ağ öğrenebileceği her şeyi öğrenerek tam öğrenmiştir [85]. Bu yüzden Faster R-CNN ağı, daha katmanlı omurga ağına sahip YOLO ağlarına göre bu veri setinde daha başarılı olmuştur. Diğer veri setlerinde ise YOLO v5, diğer ağlara üstün gelmiştir.

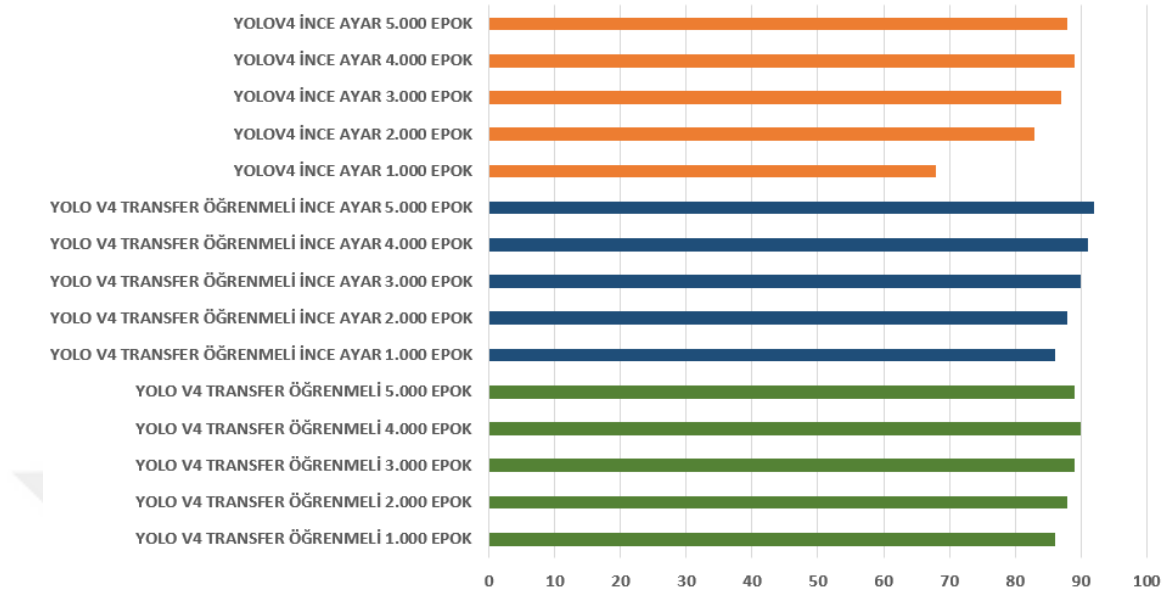
İkinci bir araştırma sorusu olan transfer öğrenmenin doğruluğu artıran bir etken olup olmadığı deneyler ile araştırılmıştır. Deneylerde transfer öğrenme uygulanan 9 ağı 7'sinde transfer öğrenme daha başarılı gelirken sadece ikisinde transfer öğrenmesiz ağlar daha başarılı gelmiştir. Ayrıca transfer öğrenme sayesinde küçük veri seti ve düşük donanım özellikleri kullanarak derin öğrenme ağ çalıştırmaları gerçekleştirilebilmiştir.

Transfer öğrenme hangi ağlarda daha fazla etki bırakıyor sorusuna cevap ise sırasıyla YOLO v4, Faster R-CNN ve YOLO v5'tir. Şekil 42'den görüleceği üzere tüm veri setleri üzerindeki çalıştırmalar dikkate alınarak; transfer öğrenmeli YOLO v4, transfer öğrenmesiz YOLO v4'e 0.04; transfer öğrenmeli Faster R-CNN, transfer öğrenmesiz Faster R-CNN'e 0.03; transfer öğrenmeli YOLO v5 ise transfer öğrenmesiz YOLO v5'e 0.0034 f1 skoru farkıyla üstün gelmiştir.

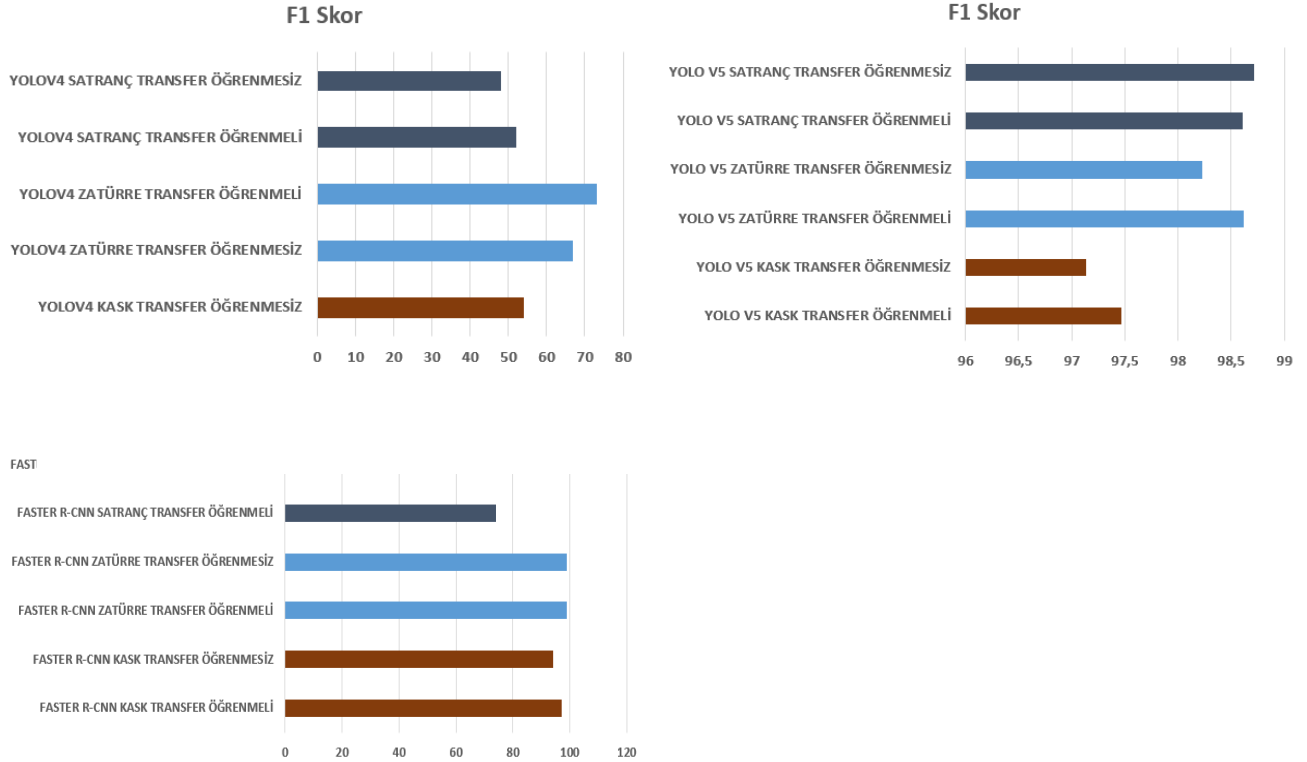
Şekil 41'den görüleceği üzere ince ayarlı ağların genel doğruluğu olarak ifade edebileceğimiz f1 skorları düşük değildir hatta bazı durumlarda sadece transfer öğrenme uygulanan ağlara üstün gelmişlerdir. Yani ince ayar uygulanan ağlar küçük nesnelerin dışında orta/büyük boyuttaki nesneleri de tespit etmiştir. Tez kapsamında uygulanan ince ayar, ağ doğruluğunu düşüren bir yöntem değildir.

Derindeki küçük nesneler tespit edildikçe savunma, güvenlik ve sağlık gibi kritik alanlarda gözden kaçırmaların minimum düzeye indirilmesi planlanmaktadır. Bu tez kapsamında bu tür nesneler, veri setinden bağımsız olarak yani üzerinde küçük nesne barındıran tüm görüntü veri setlerine benzer şekilde ama bir derin öğrenme ağına uygulanabilmektedir.

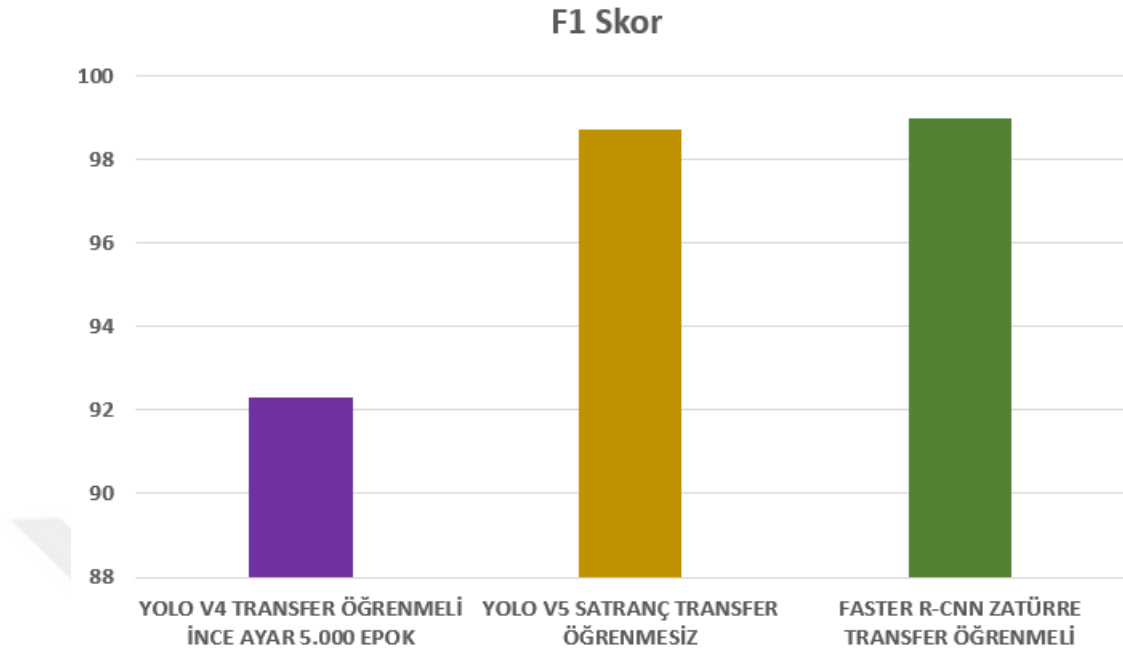
YOLO V4 İNCE AYAR ETKİNLİĞİ F1 SKORLARI



Şekil 41. YOLO v4 İnce Ayar Etkinliği F1 Skor Karşılaştırması



Şekil 42. Farklı Ağlardaki Transfer Öğrenme Etkinliği F1 Skor Karşılaştırması



Şekil 43. Ağlar Arası Yöntemlerin Aldığı En Yüksek F1 Skor Karşılaştırması

Şekil 41 ve 42’de çalışma kapsamında uygulanan deneylerin f1 skorlarına yer verilmiştir. Şekil 43’te ise ağların en yüksek başarıyı elde ettikleri metotların f1 skor karşılaştırması sunulmuştur. Şekil 41’den görüldüğü üzere transfer öğrenme ve ince ayarlı transfer öğrenme metotlarından beklenildiği gibi 1.000 epoktan 5.000 epoka doğru f1 skor artışı söz konusudur. Sadece ince ayarın uygulandığı metotta 4.000 epoktan 5.000 epoka geçişte skor düşüşü yaşanmıştır. İnce ayar deneylerinde YOLO v4 kararsız bir performans sergilemiştir. Bu 3 metot arasındaki 5.000 epok değerlerine baktığımızda en yüksek skoru 0.923 ile ince ayarlı transfer öğrenme metodu almıştır. İnce ayar, denemeler sonucu sadece YOLO v4 ağına uygulanabildiği için Şekil 41’de metodun diğer ağlara farklı yollarla denendiği uygulamaların f1 skorlarına yer verilmemiştir.

YOLO v4, zatürreli ve normal akciğer görüntü ayrımını, satranç taşı ayrımından daha başarılı bir şekilde yapmıştır. Çünkü iki nesne sınıflandırma problemini çözmek, 6 nesne sınıflandırma problemini çözmekten daha kolaydır. Faster R-CNN ise tekli nesne içeren görüntülerde diğer modellerden daha başarılı olarak 0.99 f1 skorunu zatürreli ve sağlıklı akciğer röntgen görüntülerinden elde etmiştir. Önceden de bahsedildiği üzere zatürre veri setinin boyutu küçük olduğundan Faster R-CNN burada daha başarılı çıkmıştır. YOLO v5 ise akciğer röntgen görüntüleri dışındaki tüm veri setlerinde en yüksek başarıya ulaşmıştır.

İki aşamalı, bölgesel tabanlı ağ ile tek aşamalı regresyon tabanlı, kendi içlerinde versiyon farkı olan ağların mimarileri ve çalıştırdıkları algoritmalar tez kapsamında sunulmuş ve karşılaştırılmıştır. Toplanan 3 farklı veri setinde; bu ağlar transfer öğrenme uygulanarak ve uygulanmayarak çalıştırılmış, ayrıca ince ayar tekniği farklı yollarla tüm ağlarda denenmiştir. İnce ayarın başarılı olduğu ağ çıktılarının 1.000 epokta bir alınması ile birlikte toplamda 32 ağırlık çıktısı elde edilmiştir. Çıktılar, test görüntüleri üzerinde çalıştırılıp tespit işlemleri gerçekleştirilmiş ve tespitlerin f1 skoru hesaplanarak performans karşılaştırmaları yapılmıştır.

5.3. Öneriler

Çalışmanın sonuçlarıyla birlikte gelecekteki çalışmalara yönelik olarak öneriler aşağıda sıralanmıştır:

- Küçük veri seti üzerinde çalıştırmalarda Faster R-CNN kullanılan YOLO yöntemlerine göre daha başarılı nesne tespitleri gerçekleştirdiği için önerilmektedir. Orta ve büyük ölçekli veri setlerinde ise YOLO ağları daha başarılı çıktığı için önerilmektedir.
- Bütün derin öğrenme ağ eğitimlerinde transfer öğrenmenin kullanılması, donanım kaynaklarını az harcaması, nispeten küçük veri seti üzerinde çalıştırmaya izin vermesi ve ağ doğruluğunu artırması yönüyle önerilmektedir.
- Çalışmada uygulanan ince ayarın tüm derin öğrenme ağlarına uygulanamaması çalışmanın eksik kalan yönüdür. Çalışmada bahsedilen ince ayar işleminin çalışmada kullanılmayan, farklı derin öğrenme ağlarına da denenmesi gelecek çalışmalar kapsamında planlanmaktadır.

KAYNAKÇA

1. Benjdira, B., Khursheed, T., Koubaa, A., Ammar, A., Ouni, K, 2019. Car detection using unmanned aerial vehicles: comparison between faster r-cnn and yolo3, **arXiv**; **1812**: 10968v1.
2. Javaid, U. How long will it take to train the VGG-16 model on IMAGENET using GOOGLE COLAB TPU? (Web sayfası: <https://stackoverflow.com/questions/58582739/how-long-will-it-take-to-train-the-vgg-16-model-on-imagenet-using-google-colab-t>), (Erişim Tarihi: Aralık 2022).
3. Everingham, M., Gool, L. V., Williams, C. K. I., Winn, J., Zisserman, A., 2010. The pascal visual object classes (voc) challenge, **International journal of computer vision** **88**(2), 303-338.
4. Lin, T-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C. L., 2014. Microsoft coco: Common objects in context. *European-conference on computer vision*. Springer, Cham, 2014.
5. Jwyang. A Faster Pytorch Implementation of Faster R-CNN. (Web sayfası: <https://github.com/jwyang/faster-rcnn.pytorch>), (Erişim Tarihi: Aralık 2022).
6. Amdegroot. SSD: Single Shot MultiBox Object Detector, in PyTorch. (Web sayfası: <https://github.com/amdegroot/ssd.pytorch>), (Erişim Tarihi: Aralık 2022).
7. Liu, Y., Sun, P., Wergeles, N., Shang, Y, 2021. A survey and performance evaluation of deep learning methods for small object detection. **Expert Systems with Applications** **172**: 114602.
8. Weiss, Y., Levi, K., 2004. Learning object detection from a small number of examples: the importance of good features, *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.
9. Ahmed, K. R., 2021. Smart pothole detection using deep learning based on dilated convolution, **Sensors** **21**(24), 8406.

10. Kırarak, O., Gürbüz, M. S., 2022. Performance Comparison of YOLOv3, YOLOv4 and YOLOv5 algorithms: A Case Study for Poultry Recognition, **Avrupa Bilim ve Teknoloji Dergisi** **38**, 392-397.
11. Koyun, O. C., Keser, R. K., Akkaya, İ. B., Töreyin, B. U., 2022. Focus-and-detect: a small object detection framework for aerial images. *Computer Vision and Pattern Recognition 2022*, 2022, New Orleans Louisiana.
12. Tong, K., Wu, Y., Zhou, F., 2020. Recent advances in small object detection based on deep learning: A review, **Image and Vision Computing**, **97**.
13. H. Qi, E. R. Sparks, and A. Talwalkar, "Paleo: A performance model for deep neural networks," 2016. [Online]. (Web sayfası: **<https://openreview.net/pdf?id=SyVVJ85lg>**), (Erişim Tarihi: Ocak 2023)
14. Hyperparameter Evolution (Web sayfası: **<https://docs.ultralytics.com/tutorials/hyperparameter-evolution/>**), (Erişim Tarihi: Aralık 2022).
15. Solawetz, J. YOLOv4 - Ten Tactics to Build a Better Model. (Web sayfası: **<https://blog.roboflow.com/yolov4-tactics/>**; 2020), (Erişim Tarihi: Şubat 2022)
16. LeCun, Y., Bengio, Y., Hinton, G. A., 2015. Deep learning. **Nature**; **521**(7553): 436-444.
17. Lane, D. Machine Learning For Kids. No Starch Press ve Amerika, s 392.
18. He, K., Zhang, X., Ren, S., Sun, J., 2015. Deep Residual Learning for Image Recognition, **Computer Vision and Pattern Recognition**, **10.48550/ARXIV.1512.03385**.
19. Saha, S. (Web sayfası: **<https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>**), (Erişim Tarihi: Aralık 2022).
20. Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., Liu, C., 2018. A Survey on Deep Transfer Learning, **arXiv**, **1808.01974v1**.
21. Koçer, B., 2012, Transfer Öğrenmede Yeni Yaklaşımlar. Selçuk Üniversitesi Fen Bilimleri Enstitüsü, Doktora Tezi, Konya, 93 s.
22. Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., Fei-Fei, L., 2009. Imagenet: A large-scale hierarchical image database. *2009 IEEE conference on computer vision and pattern recognition*. IEEE, 2009.

23. Fchollet. Transfer learning & fine-tuning. (Web sayfası: <https://keras.io/guides/transfer-learning/>), (Erişim Tarihi: Aralık 2021).
24. Eyiokur, F. İ., 2018, Deep Convolutional Neural Network Based Unconstrained Ear Recognition. İstanbul Teknik Üniversitesi Bilim Mühendislik ve Teknoloji Lisansüstü, Yüksek Lisans Tezi, İstanbul, 85 s.
25. Rui C., Youwei, G., Huafei, Z., Hongyu, J., 2021. A comprehensive approach for uav small object detection with simulation-based transfer learning and adaptive fusion, **arXiv : 01800v1**.
26. Chen, S., Ma, K., Zheng Y., 2019. Med3d: transfer learning for 3d medical image analysis, **arXiv, 1904.00625v4**.
27. Lu, S., Lu, Z., Zhang Y., 2018. Pathological brain detection based on AlexNet and transfer learning, **Journal of Computational Science, 30**, 41-47.
28. Chen, J., Chen, J., Zhang, D., Sun, Y., Nanekaran, Y. A., 2020. Using deep transfer learning for image-based plant disease identification, **Computers and Electronics in Agriculture, 173** :105393.
29. Ackermann, S., Schawinski, K., Zhang, C., Weigel, A. K., Turp, M. D., 2018. Using transfer learning to detect Galaxy mergers, **Monthly Notices of the Royal Astronomical Society, 479(1)**: 415-425.
30. Zhao, J., Shetty, S., Pan, J. W., Kamhoua, C., Kwiat, K., 2019. Transfer learning for detecting unknown network attacks. **EURASIP Journal on Information Security**.
31. Zhao, W., Yamada, W., Li, T., Digman, M., Runge T., 2021. Augmenting crop detection for precision agriculture with deep visual transfer learning—a case study of bale detection, **Remote Sensing, 13(1)**: 23.
32. Arya, D., Maeda, H., Ghosh, SK., Toshniwal, D., Mraz, A., Kashiya, T., Sekimoto, Y., 2020. Transfer learning-based road damage detection for multiple countries, **arXiv; 13101v1**.
33. Harsono, IW., Liawatimena, S., Cenggoro, TW., 2020. Lung nodule detection and classification from thoraxct-scan using retina net with transfer learning, **Journal of King Saud University -Computer and Information Sciences**.
34. Zhang, L., Zhang, Y., Zhang, Z., Shen, J., Wang, H., 2019. Real-time water surface object detection based on improved faster r-cnn. **Sensors, 19(16)**: 3523.

35. Chaoxia, C., Shang, W., Zhang, F., 2020. Information-guided flame detection based on faster r-cnn, **IEEE Access**, **99**, 1-1.
36. Sethy, P. K., Barpanda, N. K., Rath, A. K., Behera, S. K., 2020. Rice false smut detection based on faster r-cnn, **Indonesian Journal of Electrical Engineering and Computer Science**.
37. Khan, SU., Islam, N., Jan, Z., Din, I. U., Rodrigues, J. J. P. C., 2019. A novel deep learning based framework for the detection and classification of breast cancer using transfer learning, **Pattern Recognition Letters**, **125**, 1-6.
38. Deepak, S., Ameer, P. M., 2019. Brain tumor classification using deep CNN features via transfer learning, **Computers in Biology and Medicine**, **111**: 1033345.
39. Pathak, Y., Shukla, P. K., Tiwari, A., Stalin, S., Singh, S., Shukla, P. K., 2020. Deep transfer learning based classification model for covid-19 disease, **IRBM**.
40. Kaya, A., Keceli, A. S., Catal, C., Yalic, H. Y., Temucin, H., Tekinerdogan, B., 2019. Analysis of transfer learning for deep neural network based plant classification models, **Computers and Electronics in Agriculture**, **158**, 20-29.
41. Dawei, W., Limiao, D., Jiangong, N., Jiyue, G., Hongfei, Z., Zhongzhi, H., 2019. Recognition pest by image-based transfer learning, **Journal of the Science of Food and Agriculture** **99**:10. 4524-4531.
42. Loey, M., Manogaran, G., Taha, M. H. N., Khalifa, N. E. M., 2021. A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic, **Measurement**, **167**: 208288.
43. Vo, Anh. H., Son, L. H., Vo, Minh. T., Le, T., 2018. A novel framework for trash classification using deep transfer learning, **IEEE Access**, **7**, 178638-178639.
44. Saeed, N., King, N., Said, Z., Omar, MA., 2019. Automatic defects detection in cfrp thermograms, using convolutional neural networks and transfer learning, **Infrared Physics & Technology**, **102**: 103048.
45. Zhang, J., Karkee, M., Zhang, Q., Zhang, X., Yaqoob, M., Fu, L., Wang, S., 2020. Multi-class object detection using faster R-CNN and estimation of haking locations for automated shake-and-catch apple harvesting, **Computers and Electronics in Agriculture**; **173**: 105384.
46. Geng, M., Wang, Y., Xiang, T., Tian, Y. Deep transfer learning for person re-identification. **arXiv**, **1611: 05244v2**, 2016.

47. Rehman, A., Naz, S., Razzak, M. I., Akram, F., Imran, M., 2020. A deep learning-based framework for automatic brain tumors classification using transfer learning, **Circuits, Systems, and Signal Processing**, **39**, 757–775.
48. Gao, Y., Mosalam, K. M., 2018. Deep transfer learning for image-based structural damage recognition. **Health Monitoring of Structures**, **33**: 9, 748-768.
49. Chen, Z., Zhang, T., Ouyang, C., 2018. End-to-end airplane detection using transfer learning in remote sensing images, **Remote Sensing**, **10**(1): 139.
50. Abdulghani, A. M. A.G., 2021. A Comparison Of Image Detection Algorithms YOLO And Faster R-CNN In Different Conditions. Atılım Üniversitesi Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Ankara, 105 s.
51. Al-Nussairawi A. H. N., 2021. Object Detection And Recognition In RGBD Images Using Faster R-CNN. Altınbaş Üniversitesi Lisansüstü Çalışmalar Enstitüsü, Yüksek Lisans Tezi, 69 s.
52. Fathi, Z., 2020. Bilgisayar Görmesine Dayalı ve Evrimsel Sinir Ağları Kullanarak Yaş ve Cinsiyet Tahmini. Sakarya Üniversitesi Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Sakarya, 63 s.
53. Ali, S., 2021. Deep Learning Model Optimization For Real-time Small Object Detection On Embedded GPUs. İstanbul Medipol Üniversitesi Mühendislik ve Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, İstanbul, 88 s.
54. Diri, M. F., 2019. Plastic Object Detection With An Infrared Hyperspectral Image. Ortadoğu Teknik Üniversitesi Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Ankara, 114 s.
55. Eken, İ. C., 2017. Underwater Target Detection With Hyperspectral Imagery For-Search And Rescue Operations. Ortadoğu Teknik Üniversitesi Bilgi Sistemleri Departmanı, Yüksek Lisans Tezi, Ankara, 146 s.
56. Çınaroğlu, İ., 2014. Tümyönlü Kameralar İle Nesne Tespiti İçin Doğrudan Bir Yaklaşım. İzmir Yüksek Teknoloji Enstitüsü Mühendislik ve Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, İzmir, 61 s.
57. Li, H., Chaudhari, P., Yang, H., Lam, M., Ravichandran, A., Bhotika, R., Soatto, S., 2020. Rethinking the hyper parameters for fine-tuning. *ICLR 2002 Conference*, 30 Nisan 2020, Addis Ababa, Ethiopia.

58. Kisantal, M., Wojna, Z., Murawski, J., Naruniec, J., Cho, K., 2019. Augmentation for small object detection. *Computer Vision and Pattern Recognition 2019*, 16-20 Haziran 2019, LongBeach CA.
59. Rabbi, J., Ray, N., Schubert, M., Chowdhury, S., Chao, D., 2020. Small-Object detection in remote sensing images with end-to-end edge-enhanced GAN and object detector network. *Computer Vision and Pattern Recognition 2020*, 16-18 Haziran 2020.
60. González, J. L. S., Alvarez-Garcia, J. A., Morillo, L. M. S., Zaccaro, C., 2020. Real-time gun detection in CCTV: an open problem, **Neural networks: the official journal of the International Neural Network Society**, 132: 297-308.
61. Romero, D., Salamea, C., 2019. Convolutional models for the detection of firearms in surveillance videos, **Artificial Intelligence Applications to Smart City and Smart Enterprise**, 9 (15).
62. Ellis, C. (Web sayfası: <https://crunchingthedata.com/when-to-use-gaussian-mixture-models/>), (Erişim Tarihi: Aralık 2022).
63. Cao, G., Xie, X., Yang, W., Liao, Q., Shi, G., Wu, J., 2018. Feature-fused SSD: fast detection for small objects. *Computer Vision and Pattern Recognition 2018*, Salt Lake City.
64. Kaggle, Inc. (Web sayfası: <https://kaggle.com>), (Erişim Tarihi: Nisan 2022)
65. Öztürk, C., Taşyürek M., Türkdamar M. U. 2022. Transfer learning and fine-tuned transfer learning methods' effectiveness analyse in the CNN-based deep learning models, **Concurrency and Computation Practice and Experience**.
66. Türkdamar, M. U., Taşyürek M., Öztürk C., 2022. X-Ray Filmlerinden Transfer Öğrenmeli Faster R-CNN Derin Öğrenme ile Zatürre Tespiti. *TURK-COSE 2022: IV. Uluslararası Türk Dünyası Fen Bilimleri ve Mühendislik Kongresi*.
67. Shan, B. Fang, Y., 2020. A Cross Entropy Based Deep Neural Network Model for Road Extraction from Satellite Images, **Entropy** 22(5).
68. Türkdamar, M. U., Taşyürek M., Öztürk C. 2023. Transfer öğrenmeli ve transfer öğrenmesiz derin ağlar ile inşaat alanında kask tespiti, **Niğde Ömer Halisdemir Üniversitesi Mühendislik Bilimleri Dergisi**, 12(1), ss 1-1.
69. Taşyürek, M., Öztürk, C., 2022. DDL: A new deep learning based approach for multiple house numbers detection and clustering, **Journal Of The Faculty Of Engineering And Architecture Of Gazi University**, vol.37(2), pp.843-856.

70. Gutta, S. Object Detection Algorithm – YOLO v5 Architecture. (Web sayfası: <https://medium.com/analytics-vidhya/object-detection-algorithm-yolo-v5-architecture-89e0a35472ef>), (Erişim Tarihi: Kasım 2022).
71. Wang J., Xu, C., Yang, W., Yu, L., 2021. A normalized Gaussian Wasserstein distance for tiny object detection, **arXiv preprint** arXiv:2110.13389.
72. Zhang, K., Wang, C., Yu, X., 2022. Research on mine vehicle tracking and detection technology based on YOLOv5, **Systems Science & Control Engineering An Open Access Journal**, **10**(1): 347-366.
73. Mukherjee, S. The Annotated ResNet-50. (Web sayfası: <https://towardsdatascience.com/the-annotated-resnet-50-a6c536034758>), (Erişim Tarihi: Aralık 2022).
74. Kermany, D. S., Goldbaum, M., Cai, W., Valentim, C. C. S., Liang, H., Baxter, S. L., McKeown, A., Yang, G., Wu, X., Yan, F., Dong, J., Prasadha, M. K., Pei, J., Ting, M. Y. L., Zhu, J., Li, C., Hewett, S., Dong, J., Ziyar, I., Shi, A., Zhang, R., Zheng, L., Hou, R., Shi, W., Fu, X., Duan, Y., Huu, V. A. N., Wen, C., Zhang, E. D., Zhang, C. L., Li, O., Wang, X., Singer, M. A., Sun, X., Xu, J., Tafreshi, A., Lewis, M. A., Xia, H. ve Zhang, K., 2018. Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning, **Cell**, **172**(5), pp. 1122-1131.
75. Make ML, Hard Hat Dataset, (Web sayfası: <https://makeml.app/datasets/hard-hat-workers>), (Erişim Tarihi: Nisan 2022).
76. Skalski, P., 2019. Make-sense. (Web sayfası: <https://github.com/SkalskiP/make-sense/>), (Erişim tarihi: Ocak 2022).
77. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., De Vito, Z., Lin, Z., Demaison, A., Antiga, L., Lerer, A., 2017. Automatic differentiation in PyTorch. *NIPS-W*. 2017
78. Tensor Board.dev (Web sayfası: <https://tensorboard.dev/>), (Erişim Tarihi: Aralık 2022).
79. Waskom, M. Seaborn. (Web sayfası: <https://seaborn.pydata.org/>), (Erişim Tarihi: Kasım 2022).
80. Sarı, A., 2022, Evrişimsel Sinir Ağları İle Cilt Hastalık Görüntülerinin Sınıflandırılmasında Transfer Öğrenme Yönteminin Etkinliğinin Araştırılması. Fatih Sul-

tan Mehmet Vakıf Üniversitesi Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, İstanbul, 108 s.

81. Uyanık, İ. C., 2022, Evrişimli Sinir Ağları Ve Transfer Öğrenme Yöntemleri İle Endüstriyel Ürünlerde Otomatik Örüntü Tespiti. Millî Savunma Üniversitesi Atatürk Stratejik Araştırmalar Ve Lisansüstü Eğitim Enstitüsü, Yüksek Lisans Tezi, İstanbul, 73 s.
82. Seyıdbaylı, C., 2022, Transfer Öğrenmesi Tabanlı Hibrit Evrişimsel Sinir Ağı Modelleri Kullanılarak Meme Kanseri Teşhisi. Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, İstanbul, 80 s.
83. Üstündağ, E., 2022, Evrişimli Sinir Ağları, Transfer Öğrenme Ve Destek Vektör Makineleri Kullanarak Akciğer Röntgen Görüntülerinden Covid-19 Sınıflandırması, Afyon Kocatepe Üniversitesi Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Afyon, 98 s.
84. Foto, Ö., 2022, Transfer Öğrenme Modelleri İle Akciğer X-Ray Görüntülerinden Covid-19 Tespiti, Aksaray Üniversitesi Sosyal Bilimler Enstitüsü, Yüksek Lisans Tezi, Aksaray, 108 s.
85. Nakamura, A., Chotitamnavee, K. E., Swaroop, S., Durante, Z., Adeli, E., Li, F. F. Transfer Learning for the ICU: Comparing the Performance of YOLOv5 and-Faster R-CNN X101-FPN. (Web sayfası: <http://cs231n.stanford.edu/reports/2022/pdfs/110.pdf>), (Erişim Tarihi: Aralık 2022).
86. Nepal, U., Eslamiat, H., 2022. Comparing YOLOv3, YOLOv4 and YOLOv5 for autonomous landing spot detection in faulty UAVs, **Sensors** 22(2), 464.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı: Mehmet Uğur Türkdamar

EĞİTİM

Derece	Kurum	Mezuniyet Tarihi
Lisans	Erciyes Üniversitesi, Bilgisayar Mühendisliği	2019
Lise	Hacı Ömer Tarman Anadolu Lisesi, Ankara	2015

İŞ DENEYİMLERİ

Yıl	Kurum	Görev
2020-Halen	Niğde Ömer Halisdemir Üniversitesi	Araştırma Görevlisi
2019-2020	Niğde 112 Acil Çağrı Merkezi	Sistem Yöneticisi

YABANCI DİL

İngilizce, Fransızca

YAYINLAR

1. Öztürk, C., Taşyürek M. ve Türkdamar M. U. (2022). Transfer learning and fine-tuned transfer learning methods' effectiveness analyse in the CNN-based deep learning models. *Concurrency and Computation Practice and Experience*. <https://doi.org/10.1002/cpe.7542>.

2. Türkdamar, M. U., Taşyürek M. ve Öztürk C. (2023). Transfer öğrenmeli ve transfer öğrenmesiz derin ağlar ile inşaat alanında kask tespiti. *Niğde Ömer Halisdemir Üniversitesi Mühendislik Bilimleri Dergisi*, 12(1),ss 1-1.
3. Türkdamar, M. U., Taşyürek M. ve Öztürk C. (2022). X-Ray Filmlerinden Transfer Öğrenmeli Faster R-CNN Derin Öğrenme ile Zatürre Tespiti. *TURK-COSE 2022: IV. Uluslararası Türk Dünyası Fen Bilimleri ve Mühendislik Kongresi*.

