



AKDENİZ ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ



Hüseyin Umut YÜKSEL

BİLGİSAYARLI GÖRÜ VE DERİN ÖĞRENME YÖNTEMLERİ İLE BALON
BALIĞI TESPİTİ

Yönetim Bilişim Sistemleri Anabilim Dalı
Yüksek Lisans Tezi

Antalya, 2023



AKDENİZ ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ



Hüseyin Umut YÜKSEL

BİLGİSAYARLI GÖRÜ VE DERİN ÖĞRENME YÖNTEMLERİ İLE BALON
BALIĞI TESPİTİ

Danışman

Dr. Öğr. Üyesi Güray TONGUÇ

Yönetim Bilişim Sistemleri Anabilim Dalı

Yüksek Lisans Tezi

Antalya, 2023

T.C.
Akdeniz Üniversitesi
Sosyal Bilimler Enstitüsü Müdürlüğüne,

Hüseyin Umut Yüksel'in bu çalışması, jürimiz tarafından Yönetim Bilişim Sistemleri Anabilim Dalı Tezli Yüksek Lisans Programı tezi olarak kabul edilmiştir.

Başkan : Doç. Dr. Kerim Kürşat ÇEVİK (İmza)

Üye (Danışman) : Dr. Öğr. Üyesi Güray TONGUÇ (İmza)

Üye : Dr. Öğr. Üyesi Mevlüt ERSOY (İmza)

Tez Başlığı: Bilgisayarlı Görü ve Derin Öğrenme Yöntemleri ile Balon Balığı Tespiti

Onay: Yukarıdaki imzaların, adı geçen öğretim üyelerine ait olduğunu onaylarım.

Tez Savunma Tarihi :/....../202...

Mezuniyet Tarihi :08/12/2023

AKADEMİK BEYAN

Yüksek Lisans Tezi olarak sunduğum “Bilgisayarlı Görü ve Derin Öğrenme Yöntemleri ile Balon Balığı Tespiti” adlı bu çalışmanın, akademik kural ve etik değerlere uygun bir biçimde tarafımda yazıldığını, yararlandığım bütün eserlerin kaynakçada gösterildiğini ve çalışma içerisinde bu eserlere atıf yapıldığını belirtir; bunu şerefimle doğrularım.

(İmza)

Hüseyin Umut YÜKSEL





T.C.
AKDENİZ ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ



.... / / 20...

TEZ ÇALIŞMASI ORJİNALLİK RAPORU BEYAN BELGESİ

Öğrenci Bilgileri	
Adı-Soyadı	Hüseyin Umut Yüksel
Öğrenci Numarası	202052096003
Anabilim Dalı	Yönetim Bilişim Sistemleri
Programı	Yönetim Bilişim Sistemleri Tezli Yüksek Lisans
Danışman Öğretim Üyesi Bilgileri	
Unvanı, Adı-Soyadı	Dr. Öğr. Üyesi Güray TONGUÇ
Yüksek Lisans Tez Başlığı	Bilgisayarlı Görü ve Derin Öğrenme Yöntemleri ile Balon Balığı Tespiti
Turnitin Bilgileri	
Ödev Numarası	2201767978
Rapor Tarihi	20.10.2023
Benzerlik Oranı	Alıntılar hariç: %3 Alıntılar dahil: %3
SOSYAL BİLİMLER ENSTİTÜSÜ MÜDÜRLÜĞÜNE, Yukarıda bilgileri bulunan öğrenciye ait tez çalışmasının a) Kapak sayfası, b) Giriş, c) Ana Bölümler ve d) Sonuç kısımlarından oluşan toplam 59 sayfalık kısmına ilişkin olarak Turnitin adlı intihal tespit programından Sosyal Bilimler Enstitüsü Tez Çalışması Orijinallik Raporu Alınması ve Kullanılması Uygulama Esaslarında belirlenen filtrelemeler uygulanarak yukarıdaki detayları verilen ve ekte sunulan rapor alınmıştır. Danışman tarafından uygun olan seçenek işaretlenmelidir: (X) Benzerlik oranları belirlenen limitleri aşmıyor ise: Yukarıda yer alan beyanın ve ekte sunulan Tez Çalışması Orijinallik Raporunun doğruluğunu onaylarım. () Benzerlik oranları belirlenen limitleri aşıyor, ancak tez/dönem projesi danışmanı intihal yapılmadığı kanısında ise: Yukarıda yer alan beyanın ve ekte sunulan Tez Çalışması Orijinallik Raporunun doğruluğunu onaylar ve Uygulama Esaslarında öngörülen yüzdelik sınırlarının aşılmasına karşın, aşağıda belirtilen gerekçe ile intihal yapılmadığı kanısında olduğumu beyan ederim.	
Gerekçe:	
Benzerlik taraması yukarıda verilen ölçütlere uygun olarak tarafımda yapılmıştır. İlgili tezin orijinallik raporunun uygun olduğunu beyan ederim. Danışman Öğretim Üyesi Dr. Öğr. Üyesi Güray Tonguç İmza	

İÇİNDEKİLER

ŞEKİLLER LİSTESİ	iii
TABLolar LİSTESİ	v
KISALTMALAR LİSTESİ.....	vi
ÖZ	vii
ABSTRACT	ix
TEŞEKKÜR.....	xi
GİRİŞ	1

BİRİNCİ BÖLÜM TEZİN TANIMI

1.1. TEZİN AMACI.....	6
1.2. LİTERATÜR TARAMASI	6
1.3. ALANDAKİ ÇALIŞMALAR	8
1.4. HİPOTEZ	11
1.5. TEZİN ORGANİZASYONU	11

İKİNCİ BÖLÜM MATERYAL YÖNTEM

2.1. VERİ KAYNAĞI	13
2.2. PYTHON	16
2.3. PYCHARM IDE	16
2.4. OPENCV (OPEN SOURCE COMPUTER VİSİON)	16
2.5. ROBOFLOW PLATFORMU	17
2.6. GOOGLE COLAB	17
2.7. SUALTI ORTAMI KARŞILAŞILAN ZORLUKLAR.....	18
2.8. NESNE TESPİT MODELİ.....	19
2.8.1. Yapay Zeka.....	20
2.8.2. Makine Öğrenmesi	21
2.8.3. Yapay Sinir Ağları.....	22
2.8.4. Derin Öğrenme	23
2.8.5. Evrişimli Sinir Ağları	24
2.8.6. YOLO	25
2.8.7. YOLOv1	27

2.8.8. YOLOv2	29
2.8.9. YOLOv3	29
2.8.10. YOLOv4	29
2.8.11. YOLOv5	30
2.8.12. YOLOv6	31
2.8.13. YOLOv7	31
2.8.14. YOLOv8	32
2.9. DEĞERLENDİRME METRİKLERİ	34
2.9.1. Hata Matrisi(Confusion Matrix).....	35
2.9.2. Hassasiyet (Kesinlik-Precision).....	35
2.9.3. Duyarlılık (Recall).....	35
2.9.4. Ortalama Hassasiyet (Mean Average Precision-mAP)	36
2.9.5. Değerlerin İncelenmesi.....	36
2.9.6. Kayıp Fonksiyonları	36

ÜÇÜNCÜ BÖLÜM BULGULAR

3.1. EĞİTİM ORTAMI.....	39
3.2. EĞİTİMİN SONUÇLARI.....	39

DÖRDÜNCÜ BÖLÜM TARTIŞMA

SONUÇ	58
KAYNAKÇA	59
ÖZGEÇMİŞ	66

ŞEKİLLER LİSTESİ

Şekil 2.1 Roboflow görsel etiketleme aracı	14
Şekil 2.2 Veri setinden etiketlenmiş bazı balon balığı görselleri	14
Şekil 2.3 Eğitim sırasında kullanılan etiketlenmiş görsel örnekleri	15
Şekil 2.4 Zorlu sualtı ortamlarını gösteren temsili örnekler	18
Şekil 2.5 Yapay zeka, makine öğrenmesi, yapay sinir ağları, derin öğrenme, bilgisayarlı görüş ilişkisi	20
Şekil 2.6 YOLO tek aşamada nesne tespit işlemi (Redmon vd., 2016).....	26
Şekil 2.7 YOLOv1 Mimarisi (Redmon vd., 2016)	27
Şekil 2.8 YOLO girdi görüntülerinin ızgaraları bölünmesi (Redmon vd., 2016).....	28
Şekil 2.9 YOLOv5 mimari yapısı (Jocher vd., 2022).....	31
Şekil 2.10 YOLOv8 mimarisi (RangeKing, 2023).....	33
Şekil 2.11 YOLO versiyonlarının başarımlarının kıyaslanması (Jocher & Munawar, 2023).33	
Şekil 3.1 YOLOv8m eğitim sonuçlarına ait grafikler	41
Şekil 3.2 YOLOv8n modeli eğitim sonuçlarına ait grafikler.....	41
Şekil 3.3 YOLOv8l modelinin eğitim sonuçlarına ilişkin grafikler	42
Şekil 3.4 YOLOv5n modelinin eğitim sonuçlarına ilişkin grafikler	43
Şekil 3.5 YOLOv5m modeli eğitim sonuçlarına ait grafikler	43
Şekil 3.6 YOLOv8n doğrulama veri seti hata matrisi	44
Şekil 3.7 YOLOv8n test veri seti hata matrisi	44
Şekil 3.8 YOLOv8m doğrulama veri seti hata matrisi	45
Şekil 3.9 YOLOv8m test veri seti hata matrisi	45
Şekil 3.10 YOLOv8l doğrulama veri seti hata matrisi	46
Şekil 3.11 YOLOv8l test veri seti hata matrisi	46
Şekil 3.12 YOLOv8m doğrulama hassasiyet duyarlılık eğrisi	47
Şekil 3.13 YOLOv8m test hassasiyet duyarlılık eğrisi.....	47
Şekil 3.14 YOLOv8n doğrulama hassasiyet duyarlılık eğrisi	48
Şekil 3.15 YOLOv8n test hassasiyet duyarlılık eğrisi.....	48
Şekil 3.16 YOLOv8l doğrulama hassasiyet duyarlılık eğrisi	49
Şekil 3.17 YOLOv8l test hassasiyet duyarlılık eğrisi.....	49
Şekil 3.18 Eğitilmiş model ağırlığı ile tespit edilen balon balığı görüntüsü	50

Şekil 3.19 Gece çekimi, az ışık olan ortamda YOLOv8m model ağırlığı ile tespit edilen balon balığı	51
Şekil 3.20 Gece çekimi, çok ışık olan ortamda YOLOv8m model ağırlığı ile tespit edilen balon balığı	51
Şekil 3.21 Gece çekimi, yüksek gürültülü ortamda YOLOv8m model ağırlığı ile tespit edilen balon balığı	52
Şekil 3.22 Gündüz çekimi, normal ışık altında YOLOv8m modeli ile tespit edilmiş balon balığı görseli	52
Şekil 3.23 Gündüz çekimi, normal ışık altında YOLOv8m modeli ile tespit edilmiş balon balıkları	53
Şekil 3.24 Gürültülü ortamda balon balığı	53



TABLÖLAR LİSTESİ

Tablo 2.1 YOLOv8 nesne tespit modellerinin kıyaslanması (Jocher & Munawar, 2023)	
.....	34
Tablo 2.2 Hata Matrisi	35
Tablo 3.1 Modellerin eğitimi sonrası doğrulama veri setine ait eğitim sonuçları	40



KISALTMALAR LİSTESİ

Box loss: Kutu Kaybı

Cls loss: Classification Loss (Sınıflandırma Kaybı)

CNN: Evrişimli Sinir Ağı

ESA: Evrişimli Sinir Ağı

CSP: Çapraz Aşama Kısmi Bağlantılar

Dfl loss: Distribution Focal Loss

DPM: Deformable Parts Model

FPN: Feature Pyramid Network

GAR: En İyi Gerçek Kabul Oranı

GPU: Grafik İşlem Birimi

IDE: Entegre Geliştirme Ortamı

mAP: mean Average Presicion (Ortalama Hassasiyet)

MS COCO: Microsoft COCO (Common Objects in Context)

OpenCV: Open Source Computer Vision (Açık Kaynak Bilgisayarla Görü)

PANet: Path Aggregation Network

R-CNN: Regions With Convolutional Neural And Networks

SSD: Single Shot Multibox Detector

TTX: Tetradotoksin

YSA: Yapay Sinir Ağları

YOLO: You Only Look Once

ÖZ

Bu tez çalışması, Lagocephalus sceleratus olarak bilinen balon balıklarını tespit eden nesne tespiti modelinin eğitimi üzerine odaklanmaktadır. Süveyş Kanalı'nın açılması ve Asvan Barajı'nın tamamlanması, Akdeniz'in ekosistemini önemli ölçüde değiştirmiştir. Bu değişim, Kızıldeniz'den Akdeniz'e olan canlı göçlerini artırmış ve yeni türlerin Akdeniz'e yayılmasına neden olmuştur. Lagocephalus sceleratus gibi zararlı türler, ekonomik kayıplara ve insan sağlığına yönelik tehditlere yol açmaktadır. Bu balık türü, doğal avcısı olmayan, geniş alanlarda yayılan ve zehirli olmasıyla bilinen bir türdür.

Bu çalışma, bilgisayarlı görü ve derin öğrenme tekniklerini kullanarak balon balığı tespiti konusunda bir modelin eğitilmesi amacı taşımaktadır. Yapılan literatür incelemeleri ve mevcut teknoloji dikkate alındığında, en güncel You Only Look Once (YOLO) algoritmaları üzerine odaklanarak bu tür balıkların otomatik olarak tespit edilmesini hedeflemektedir. Elde edilen veriler, Akdeniz'deki dalış okullarından ve dalış eğitmenlerinden sağlanmıştır. Videolardan elde edilen fotoğraf kareleri YOLO formatında kullanılmak üzere uygun şekilde etiketlenmiş ve 2473 görselden oluşan veri seti hazırlanmıştır. Bu verilerle YOLOv8 derin öğrenme modelleri eğitilmiş ve balon balığı tespiti için eğitilen modellerle denemeler yapılmıştır. YOLOv8 sürümü, YOLOv5 sürümünü geliştiren Ultralytics şirketi tarafından geliştirilen en güncel YOLO sürümü olup önceki sürümlerin başarımı geçmeyi hedeflemektedir.

Çalışma kapsamında YOLO derin öğrenme modelinin v8 ve v5 sürümleri ve farklı boyutlarda parametre içeren paketleri eğitilerek denemeler tamamlanmış ve en iyi başarımlar YOLOv8m sürümü ile ortalama hassasiyet %96,90 olarak elde edilmiştir. Çalışmada YOLOv8 modelleri üzerinde durulmakta olup YOLOv5 modellerinin başarımları kıyaslama yapmak için verilmiştir.

Eğitimlerde balon balıklarının kafa kısmından ve yan açılardan daha iyi tespit edilebildiği gözlenmiştir. Bu, modelin balon balığının göz ve çevresini yakalamaya çalıştığı ve balığın şeklini yan açıdan tespit edebildiği gerçeğinden kaynaklanmaktadır. Ancak, manuel etiketleme sürecindeki bazı zorluklar nedeniyle balıkların kuyruk ve yüzgeçleri tam olarak dikkate alınmamış, bu da modelin odaklanmasını etkilemiştir. Ayrıca, farklı ışık ve ortam koşullarındaki balıkların farklı açılardan seçilen görselleri, modelin genelleme yeteneğini artırmıştır.

Bu çalışmanın sonuçları, sualtı robotları ve diğer otomatik tespit sistemlerinde kullanılarak balon balığı popülasyonunun kontrol altında tutulması ve ekolojik dengenin

korunması konularında önemli bir katkı sağlayabilir. Bu model, balon balığı probleminin çözümüne daha etkili ve sürdürülebilir bir yaklaşım sunarak gelecekte yapılacak çalışmalara temel oluşturabilir. Bu bağlamda, çalışmanın ileri sualtı teknolojileri ve ekoloji çalışmaları için değerli bir kaynak olması beklenmektedir.

Anahtar Kelimeler: Derin öğrenme, Bilgisayarla görü, YOLO, Balon balığı, Nesne tespiti



ABSTRACT
PUFFERFISH DETECTION WITH COMPUTER VISION AND DEEP LEARNING
METHODS

This thesis focuses on the training of an object detection model to identify pufferfish known as *Lagocephalus sceleratus*. The opening of the Suez Canal and the completion of the Aswan High Dam have significantly altered the ecosystem of the Mediterranean Sea. This change has increased the migration of living organisms from the Red Sea to the Mediterranean and led to the spread of new species in the Mediterranean. Harmful species like *Lagocephalus sceleratus* pose threats to economic losses and human health. This species is known for being non-native, spreading in vast areas, and being poisonous, without natural predators.

This study aims to train a model for pufferfish detection using computer vision and deep learning techniques. Considering the existing literature and technology, it aims to automatically detect such fish, focusing on the most recent You Only Look Once (YOLO) algorithms. The data used in this research were obtained from diving schools and instructors in the Mediterranean. Frames extracted from videos were appropriately labeled in YOLO format, creating a dataset consisting of 2473 images. YOLOv8 deep learning models were trained with this dataset, and experiments were conducted using the trained models for pufferfish detection. YOLOv8 is the latest version developed by Ultralytics, surpassing the performance of previous versions, including YOLOv5.

Within the scope of the study, experiments were conducted by training versions 8 and 5 of the YOLO deep learning model, as well as packages with different sets of parameters. The best performance was achieved with the YOLOv8m version, with a mean average precision of 96.9%. The study focuses on YOLOv8 models, and the achievements of YOLOv5 models are provided for comparison purposes.

During the experiments, it was observed that pufferfish could be better detected from the head and side angles. This is due to the model attempting to capture the eyes and surroundings of the pufferfish and being able to detect its shape from a side angle. However, due to some challenges in the manual labeling process, the tails and fins of the fish were not fully considered, affecting the focus of the model. Additionally, selecting images of fish from different angles and lighting conditions enhanced the model's generalization ability.

The results of this study could contribute significantly to controlling the pufferfish population and maintaining ecological balance by being employed in underwater robots and other automatic detection systems. This model could provide a more effective and sustainable

approach to solving the pufferfish problem and serve as a foundation for future research. In this context, the study is expected to be a valuable resource for advanced underwater technologies and ecological studies in the future.

Keywords: Deep learning, Computer vision, YOLO, Pufferfish, Object detection



TEŞEKKÜR

Bu tez çalışmasını tamamlamamda emeği geçen danışman hocam Dr. Öğr. Üyesi Güray Tonguç’a sonsuz teşekkür ederim. Bana her zamanki desteklerinden dolayı aileme teşekkürü borç bilirim. Ayrıca çalışmamız sırasında bize destek veren Profesör Dr. Mehmet Gökoğlu, Doç. Dr. Ahmet Balcı, Star Diving Academy Dalış Merkezi, Poseidon Kemer Diving ve dalış eğitmeni Orkun Tekin’e teşekkür ederim.



GİRİŞ

İnsanlık tarih boyunca yaşadığı çevreyi de kendi ihtiyaçlarına uyacak şekilde değiştirmeye çalışmıştır. Bu değişimler beraberinde değişimin gerçekleştiği coğrafyada birtakım etkilere sebep olmuştur. Bu etkilerin tamamı karalar üzerinde gerçekleşmemiştir. İnsanlığın denizleri kullanarak gerçekleştirdiği ticari faaliyetlerde yeni rotalar oluşturmak ve limanlar arası mesafeleri kısaltmak için açtıkları kanallar denizlerin ekosistemi üzerinde de bazı öngörülemeyen değişikliklere sebebiyet vermiştir (El-Raey, 2010; Mavruk & Avsar, 2008; Rodrigue, 2020).

Akdeniz olarak bilinen deniz ve bu denizin ekosistemi 1869 yılında Süveyş Kanalı'nın yaklaşık yüz yıl sonra da Nil Nehri üzerine inşa edilen Asvan Barajı'nın inşasının tamamlanmasıyla gerçekleşen olaylar neticesinde önemli derece değişimlere uğramıştır (Golani, 1999). Süveyş Kanalı'nın inşasının tamamlanması ile bu kanal benzer özelliklere sahip Akdeniz ve Hint Okyanusu'na açılan Kızıldeniz'in sularını birbirine bağlamıştır. Bunun doğal sonucu olarak da bu iki deniz arasında canlı göçlerinin önü açılmış olup göçlerin büyük çoğunluğu Kızıldeniz'den Akdeniz'e uzanan rotada gerçekleşmiştir (Golani, 1998).

İncelemeler neticesinde Kızıldeniz'den Akdeniz'e göç eden İndo-Pasifik orijinli türler kayıt altına alınmaya çalışılmış, 2008 yılında toplanan veriye göre bir yeni türün görülmesi sıklığı 9 günde bir olmak üzere 903 türün göç ettiğine dair kayda ulaşılmıştır. Bu oranın tahmin edilen en kötü senaryodan daha yüksek bir oran olduğu görülmektedir (Zenetos vd., 2008). 2010 yılında toparlanan verilere göre Türkiye kıyılarında 400 yabancı türün varlığı kayıt altına alınmıştır. Bu türlerin bazıları çeşitliliği arttırıp başta balıkçılık sektöründe olmak üzere ekonomik katkı sağlarken bazıları zararlı türlerdir (Cinar vd., 2011). 2020 yılında güncellenen verilere göre Türkiye kıyılarında gözlenen yabancı tür sayısı 539'a ulaşırken yerli olmayan balık türlerinin sayısının 80 olduğu raporlanmıştır (Çinar vd., 2021).

Akdeniz ve ekosistemini etkilen bu türler Lesepsiyen türler ve bu göç hareketi Lesepsiyen göç adıyla kendisine özel bir isimlendirme bulmuş olup (Por, 1971). Lesepsiyen türler Kızıldeniz'den Akdeniz'e göç eden balık türlerine verilen isimdir ve bu göç hareketine de Lesepsiyen göçler denir. İsmin fikir babası olan Francis Dov Por, Lesepsiyen adını bulurken, kendisine isim noktasında ilham kaynağı olan kişide Fransız diplomat Ferdinand de Lesseps'dir (Por, 1971). Lesseps Süveyş Kanalı inşa projesinde büyük emekleriyle tanınan bir mühendistir (Eryılmaz & Dalyan, 2006).

Lesepsiyen türler arasında Akdeniz'e göç etmiş bir balon balığı türü olan Lagocephalus scleratus 2003 yılı Gökova Körfezi'nde örneklenerek Süveyş Kanalı aracılığıyla Akdeniz'e

geçen türler arasında yer almıştır. (Akyol vd., 2005). Akdeniz'e sonradan gelmiş yabancı bir tür olmasına rağmen bölgeye kolay uyum sağlamış ve geniş bir alanda varlığını sürdürmektedir. Bu balığın benekli balon balığı, zehirli balon balığı, gümüş yanaklı kurbağa balığı gibi farklı bilinen isimleri de vardır (Ulman vd., 2021). Balığın Akdeniz de yayılım gösterdiği alanın büyük olmasında küresel ısınmanın etkisi de vardır (Demertzis & Iliadis, 2018). Akdeniz'in sıcaklığının yükselmesi bu balık türünün Akdeniz sularındaki istilasını kolaylaştırmıştır (Çinar vd., 2021).

Lagocephalus sceleratus vücudundaki tetradotoksin (TTX) yüzünden sadece deniz ekosistemini değil insan sağlığı için risk oluşturmakta ve ekonomik aktiviteleri de olumsuz etkilemektedir (Zenetos vd., 2009). Balon balıklarının içeren familya Tetraodontidae ailesidir. En belirgin fenotipleri yapışmış çene, gözle görülen üstte ve altta yer alan iki büyük diştir. Familya adı da buradan gelmektedir. Bu türün bireyleri tehlike hissettiklerinde ya da stres altında bırakıldıklarında vücutlarının ventral kısmının yarısını suyla şişirerek balonu da andıran bir tepki vermektedir. Yüzgeçlerinde sert ışın bulundurmazlar (Akyol vd., 2005). Ailenin üyeleri her şeyi tüketirler, omnivor ve herbivor üyeleri vardır. Karides, yengeç, mercan, yumuşakça vb. omurgasızları ve küçük balıkları yemeyi tercih ederler ve yayılım gösterdikleri alanlarda bu türlerin popülasyonunda azalmalar görülür bu da yol açtıkları ekonomik kayıpların arasında yer alan sebeplerden biridir (Kulbicki vd., 2005). Balon balıkları, balon balığı dışında isimlendirmelere de sahiptir, kurbağa, küre, cüce ve domuz balon balığı olarak isimlendirildikleri görülmüştür. (Tüzün, 2012). *Lagocephalus sceleratus* barındırdığı savunma mekanizması olan tetradotoksin yüzünden diğer balıklar tarafından av olarak görülmemektedir. Balığın ekosistemde doğal avcısının olmayışı çoğalmasını kolaylaştırmıştır. Hemen hemen her şeyi tüketen bu balık türünün bir ekonomik getirisi de yoktur. Balığın zehrinin çıkarılması deneyimli aşçılar tarafından yapılması gereken bir iş olduğu ve balığın tüketimine bağlı ölümler olduğu için çoğu ülkede tüketimi yasaklanmış ya da kısıtlanmıştır. Böyle olunca balıkçılar tarafından normal koşullarda avlanan bir tür değildir. Bu türün verdiği zararlar sadece bununla kısıtlı kalmamıştır. Yaşadığı bölgede diğer balık ve yumuşakça türlerinin popülasyonunda düşüş yaşanmasına sebep olmuş bu yüzden de avcılıktan gelen gelirin düşmesine sebep olmuştur. Yine balıkçılık faaliyeti ile uğraşan kıyı balıkçılarına sorunlar çıkarmıştır. Balıkçıların ağlarını dişleri ile yırttığı ve avlanan diğer balık türleri arasında istenmeden de olsa bu türün avlandığı görülmüştür. İnsanların bilmeden ya da yanlışlıkla tüketmesiyle de ölüm vakaları gözlenmiştir (Tüzün, 2012).

Sualtı çalışmalarında doğrudan gözlem çalışmaları yapmak bir hayli zor olmakla birlikte su tankları ya da akvaryumlar içerisinde bu yaklaşım daha çok tercih edilmektedir. Sualtı

çalışmalarında genellikle sualtı kameralarını içeren sistemler kullanılır. Kameralarla toplanan büyük miktardaki verinin doğru işlenebilmesi için elde edilen görüntülerdeki canlı türlerini otomatik olarak tanıyan ve tespit eden modellerin kullanılması gerekmektedir. Bu tarz modellerin en çok zorlandığı kısım balıkların doğal ortamındaki ışık ve sudaki bulanıklık tarzı çevre etkileri olmaktadır. Kamera teknolojilerindeki gelişmeler de bu tarz çevre sorunlarının etkisinin azaltılmasına olanak sağlamaktadır (Knausgard vd., 2022). Bu tarz modellerin geliştirilmesinde bilgisayarlı görü ve derin öğrenme algoritmaları kullanılmaktadır. Bilgisayarlı görü bilgisayarların dijital fotoğraflardan, videolardan, canlı yayın görüntülerinden bilgi, anlam çıkarmasıyla uğraşan yaklaşımdır. Bilgisayarın gördüğü şeyleri insanların gördüğü şekilde anlamlandırmasını sağlamaya yönelik çalışmaları içerir (Ballard & Brown, 1982). Bilgisayarlı görüde bilgisayarın gördüğü şeyleri algılaması temeline makine öğrenmesi ve derin öğrenme algoritmalarını alır. Derin öğrenme de denetimli, yarı denetimli veya denetimsiz olarak ayrılmak üzere gerçekleştirilebilen yapay sinir ağları ve benzeri makine öğrenmesi algoritmalarını kapsayan makine öğrenmesinin çözmekte yetersiz kaldığı problemlerde tercih edilen bir çalışma alanıdır (LeCun vd., 2015). Nesne tespiti de videolarda ve dijital görüntülerle çalışan ve burada sınıfları tespit etmeye çalışan bilgisayarlı görü ve görüntü işleme ile ilgilenen teknolojilerdir. Nesne sınıflarının kendilerine has olan özelliklerini yakalamaya ve bu yakaladığı özelliklerle nesneleri tanımaya çalışır. Sinir ağlarını kullanan yaklaşımlar olduğu gibi sinir ağları kullanmayan yöntemlerde bulunur (Dasiopoulou vd., 2005). Sinir ağı kullanan yöntemlerden oluşan YOLO algoritması gerçek zamanlı anlık olarak nesne tespiti gerçekleştirmeye odaklanmıştır. Bu alanda performans, doğruluk ve keskinlik gibi değerleri ele aldığımızda önde gelen en güncel algoritmalarındandır. YOLO algoritması merkezine sinir ağlarını alır ve farklı nesne tespit yapılarıyla birleştirilip farklı sürümleri geliştirilmiştir (H. Wang vd., 2022).

Balon balığı probleminin çözümünde geleneksel tabir edebileceğimiz aşırı avlanma tarzı balıkçılık yöntemleri ile çözüm üretilmeye çalışılmaktadır. Literatürdeki çalışmalar daha çok balon balığının varlığının tespitine, türün nasıl bir tür olduğuna ve türün çevreye ve insana verdiği zararlara yönelik çalışmaları içermektedir. Güncel teknoloji ve yöntemler özellikle bilgisayarlı görü ve derin öğrenme kullanarak farklı balık türleri, deniz kaplumbağaları, balinalar ve yunus gibi sualtı canlılarının tespitine yönelik çalışmalar olsa da balon balığı probleminin çözümüne odaklanan çalışmalara rastlanmamıştır (Arvind vd., 2019; Chuang vd., 2017; Kezebou vd., 2019; Spampinato vd., 2008; Xu & Cheng, 2017). Bu çalışmada literatürde eksikliği hissedilen balon balığı probleminin insansız sualtı robotları ile çözümüne yönelik çalışmalarda kullanılabilmesi açısından başlangıç noktası olan derin öğrenme ile balon balığı

tespiti modelinin oluşturulmasına ve başarımının test edilmesine odaklanılmaktadır. Bu çalışma ile ortaya konulacak model, ileride sualtı robotları veya sualtı dronlarında kullanılarak insansız bir şekilde balon balığı tespiti ve avlanmasının mümkün hale gelmesi için diğer çalışmalara temel oluşturacaktır. Bu kapsamda bu çalışma balon balığı problemine daha etkili ve sürdürülebilir bir çözüm üretmeye odaklanmaktadır.

Tez çalışmasının amacı doğrultusunda ekonomik ve ekolojik zararları olan ve insan sağlığı için tehdit oluşturan balon balıklarının bilgisayarlı görü ve derin öğrenme ile tespiti amaçlanmakta ve özellikle popüler en güncel You Only Look Once (YOLO) algoritmalarına odaklanmaktadır. Bu balıkların sebep olduğu sorunları anlamak kadar en son teknoloji ve yöntemlerden faydalanarak yeni çözümler geliştirmeye çalışmakta önemlidir. Bu yeni çözümler balon balığı vb. istilacı türlerin popülasyonunu daha kolay ve etkili bir şekilde kontrol altında tutmaya odaklanırken bu türlerin çevreye ve insanlara verdiği zararları en aza indirmeye yönelik olmalıdır. Son yıllarda, bilgisayar görme ve derin öğrenme teknikleri, otomatik nesne tespiti ve tanıma görevlerini mümkün kılarak çeşitli alanlarda devrim yaratmıştır. Bu ilerlemelerden büyük ölçüde fayda sağlayabilecek alanlardan biri, şekil değiştirme yeteneği, canlı renkler sergileme ve çevresiyle uyum sağlama özellikleriyle bilinen balon balığının tespitidir. Doğru ve verimli balon balığı tespiti hem ekoloji çalışmaları hem de halk sağlığı endişeleri açısından büyük önem taşımaktadır (Demertzis & Iliadis, 2018).

Bu tez kapsamında yürütülen çalışmada balon balığı ile ilgili görsellerin Akdeniz’de faaliyet yürüten dalış okullarından ve dalış eğitmenlerinden toplanması, toplanan görseller ile veri setinin oluşturulması ve oluşturulan veri setini kullanarak YOLO modelinin eğitilmesi amaçlanmaktadır. Oluşturulan veri setinde görsellerin sınıf etiketleriyle etiketlenmesi, etiketlenmiş görsellerin eğitim, doğrulama ve test veri setlerine ayrılması, veri setlerindeki veriler üzerinde ön işleme yapılması, YOLOv8 derin öğrenme modeli kullanılarak eğitimin gerçekleştirilmesi, modelin optimizasyonu, modelin son hali ile denemeler yapılması ve duyarlılık, kesinlik, ortalama hassasiyet değerleri dikkate alınarak sonuçların yorumlanması amaçlanmaktadır.

Çalışmamızda ortaya koyduğumuz modelin sualtı robotları vb. cihazlarla kullanımı durumunda otomatik olarak tespit edilebilecek balon balıklarının popülasyonunun kontrol altında tutulması ve bu balıkların vereceği zararın en aza indirilmesi süreci daha iyi bir şekilde ele alınabilecektir. Balon balığı gibi istilacı bir balık türü olan aslan balığı için gerçekleştirilmiş tespit ve takip robotları düşünüldüğünde balon balıkları için geliştirilebilecek sualtı robotlarıyla balon balıklarının yakalanması ya da etkisiz hale getirilmesi süreci otomatikleştirilebilecektir (Anderson, 2014; Antaya vd., 2019). Çalışmanın güncel ve güncelliği devam eden bir konuda

olması sebebiyle de ileride gerçekleştirilecek çalışmalara yöntem ve literatür noktasında katkıda bulunması beklenmektedir.



BİRİNCİ BÖLÜM

TEZİN TANIMI

2.1. Tezin Amacı

Akdeniz’de popülasyonu giderek artan ve birçok olumsuz etkisi bulunan istilacı bir tür olan balon balıklarının tespiti bu balıkların vermekte olduğu ve ileride verebileceği zararlarla mücadelede atılması gereken önemli adımlardan biridir. Sualtı ortamında ışığın kırılması, suyun bulanık olması ve arka plan gürültüsünün yoğun olması balık tespitini ve takibini zorlaştırmaktadır. Sualtı ortamı vb. ortamlarda insan gözlemleriyle balıkların tespit edilmesi ve hareketlerinin izlenmesi zordur. Bu durum alandaki araştırmaların hızına ve başarısına olumsuz etki etmektedir. Balon balığının yol açtığı sorunların giderilmesinde bu istilacı türün tespiti ve takibi popülasyonunun kontrol altında tutulabilmesi için gerekli olan temellerden biridir. Bu problemin giderilmesi ve sualtı ortamında balon balığının tespit edilmesine odaklanan tez çalışmamızda Akdeniz Bölgesinde dalış gerçekleştiren dalış okullarından ve profesyonel dalgıçlardan balon balıklarını içeren video görüntüleri toplanmıştır. Toplanan video görüntüleri fotoğraf karelerine ayrılmış, fotoğraflarda yer alan balon balıkları etiketlenmiş ve etiketlenen fotoğraf kareleri ile nesne tespit modelinin eğitiminde kullanılacak veri seti oluşturulmuştur. Videolardan elde edilen etiketlenmiş fotoğraf kareleri YOLOv8 ve YOLOv5 modelleriyle eğitilmiş ve kesinlik, duyarlılık ve ortalama hassasiyet değerleri üzerinden eğitim başarısının karşılaştırılması yapılmıştır. Bu modelin balon balığı tespitinde kullanılabilmesi ve ileride balon balığı popülasyonunu kontrol altında tutabilecek sualtı robotlarında kullanılabilmesi amaçlanmaktadır.

2.2. Literatür Taraması

Bilgisayarlı görü, dijital görüntülerin veya videoların yorumlanmasını otomatikleştirmek için bilgisayar algoritmalarının kullanımını araştıran bir araştırma alanı olarak bilimde önemli kullanım örneklerine sahiptir (Beyan & Browman, 2020). Daha geniş tanımıyla bilgisayarla görme ya da bilgisayarlı görü, insanlardaki görme sisteminin taklidi prensibi üzerine geliştirilmiş olmakla birlikte bilgisayar sistemlerinin dijital görüntüler veya video akışları üzerinden nesneleri algılamasına ve anlamlandırmasına olanak tanıyan bir alandır. Bu teknoloji, görüntü veya video verilerini işleyerek nesneleri tanıma, sınıflandırma, yerini belirleme ve hareketi takip etme gibi görevleri gerçekleştirebilir (Zhao vd., 2019). Bu sebeplerle yapılan çalışmalarda sık sık başta derin öğrenme algoritmaları olmak üzere yapay

zekâ algoritmalarıyla kullanımları görülmektedir. Akdeniz de yaşanan balon balığı istilasına gibi biyolojik çeşitlilik dünya çapında önemli ve hızlı değişimler geçirirken sualtı biyolojik ortamının izlenmesinde ve problemlerin tespiti ve çözümlenmesinde doğru kararlar alabilmek adına bu alanda doğru veriye ihtiyaç vardır (Cardinale vd., 2012). Balık tespiti gibi uygulamalarda bilgisayarla görme tekniklerinin geleneksel teknikler yerine kullanımı sadece kazandırdığı avantajlardan ötürü değil sualtı ortamının yapısından dolayı var olan birtakım özelliklerden ötürü bir zorunluluk haline gelmiştir çünkü sualtı ortamları insan gözü ile doğrudan gözlem yapmaya uygun bir ortam değildir. Suyun optik özellikleri, ışığın kırılması ve yansması gibi faktörler, elde edilecek verilerin hassasiyetine olumsuz etki etmekte, balıkların doğru bir şekilde tespit edilmesini ve izlenmesini zorlaştırmaktadır (Cui vd., 2020). Sualtı ortamındaki bulanıklık gibi sorunlar geleneksel yöntemlerle elde edilen kamera görüntülerinden bilgi elde edimini zorlaştırmaktadır (Boswell vd., 2008). Son yapılan çalışmalarda, büyük veri miktarı toplamak için uyarlanabilir çözünürlüklü görüntüleme sonar kameraları kullanan araştırmacıların, örneklerinde balıkları manuel olarak sayma zorunluluğu nedeniyle sınırlandıkları görülmüştür (Lankowicz vd., 2020). Bu nedenle bilgisayarla görme, su altında hareket eden balıkları tespit etmek ve izlemek için kullanışlı bir araç haline gelmiştir. Yine bilgisayarla görme algoritmaları, görüntü veya video verilerini otomatik olarak analiz edebilmemize imkân sağlayarak bu alandaki çalışmaları hızlandırmıştır. Bu, büyük miktardaki veriyi hızlı bir şekilde işleyebilme ve nesneleri tanıma, sınıflandırma veya hareketi takip etme gibi görevleri otomatikleştirme imkânı sağlamaktadır. Sadece balıkların tespitinde değil balıkların taşınması muhtemel hastalıkların tespitinde ve balık çiftliklerinde ağların, balıkların ve suyun durumunun izlenmesinde de kullanım alanı bulmaktadır (D. Li vd., 2022). Bilgisayarla görme, yüksek çözünürlüklü kameralar ve gelişmiş algoritmalar kullanarak çok küçük ayrıntıları algılayabilir (Bicknell vd., 2016). Bu sayede su altında bulunan balıkların veya diğer nesnelerin ince detaylarını tespit edebilir (Schneider vd., 2019). Bilgisayarla görme sistemleri, sualtı ortamında bulunan kameralardan gelen verilere uzaktan erişebilir. Bu, sualtı balık tespiti çalışmalarında denizaltında veya uzaktaki bir kontrol merkezindeki bilgisayarlarda analiz yapabilme imkânı sağlar ve sualtı ekosistemlerinde nesne algılama ve nesne takibi kombinasyonu ile balıkların otomatik olarak izlenmesi konusunda yapılan çalışmaların güvenilir ve doğru olduğuna dair kanıtlar bulunmaktadır (Lantsova vd., 2016; Mohamed vd., 2020; Spampinato vd., 2008). Derin öğrenme, yapay zekânın bir alt kümesidir ve yapılandırılmamış veriler içindeki daha yüksek boyutlu temsilleri öğrenme ve desenleri tespit etme yeteneğine sahip ağlar kullanır (Schmidhuber, 2015). Derin öğrenme modelleri, sualtı ortamları gibi zorlu tespit, takip ve sınıflandırma problemlerinde özellikle başarılı olmuştur

(Ditria vd., 2020). Bu sebeple farklı derin öğrenme algoritmaları bu alanda kullanılmaktadır (N. Wang vd., 2022). Derin öğrenme, belirli bir alanda toplanan büyük veri setlerine dayanmaktadır. Kapsamlı veri kaynaklarından öğrenme son derece önemlidir. "Derin" terimi, sinir ağının beynimizi taklit etmek için birçok katmana sahip olduğunu ifade eder. Yüksek performanslı grafik işlem birimlerinin(GPU), uygulamaya özgü entegre devrelerin(ASIC - Application-Specific Integrated Circuit), bulut depolama ve güçlü bilgi işlem altyapılarının ortaya çıkmasıyla, büyük veri kümelerini toplamak, yönetmek ve analiz etmek mümkün hale gelmiştir. Derin öğrenmede, aşırı uydurma sorunlarının çözülebilmesi için yeterince büyük veri kümelerine ihtiyaç vardır. Ayrıca, artan bilgi işlem gücü, zaman alıcı eğitim sürecinin hızını artırabilir (Cui vd., 2020). Nesne tanıma ve sınıflandırmada yaygın olarak kullanılan derin öğrenme algoritmaları Evrişimli Sinir Ağı (ESA) türevleridir. ESA prensiplerine dayanan birçok yöntem önerilmiş çeşitli alanlarda başarı ve performansları gösterilmişlerdir (B S vd., 2020; Krizhevsky vd., 2017; Simonyan & Zisserman, 2015).

2.3. Alandaki Çalışmalar

Hızlı R-CNN (Regions with Convolutional Neural and Networks) özelliklerinden yararlanmayı amaçlayan Li ve ark. (2015) bir çalışmada evrensel VOC ve ImageNet veri setlerinde Evrişimli Sinir Ağları (CNN'ler) tarafından elde edilen güçlü tanıma sonuçlarından ilham alarak, bu popüler derin ConvNet'leri, su altı ortamına uygulamak için 12 sınıfa ait 24277 ImageCLEF balık görüntüsünden oluşan yeni bir veri seti kullanılmıştır. Hızlı R-CNN, Deformable Parts Model (DPM) temeline göre ortalama hassasiyeti (mAP) %11.2 artırarak 81.4 mAP elde etmekte ve tek bir balık görüntüsünde önceki R-CNN'den %80 daha hızlı tespit yapabilmektedir.

Bir evrişimli sinir ağı (ESA), balık tespiti noktasında insanların hız ve doğruluk açısından performansı ile karşılaştırılmıştır. Villon ve ark. (2018) çeşitli fotoğraf veri tabanlarıyla eğitilen CNN'nin performansı test edilmiş ve 20 farklı balık türünü tanımlama başarısı ölçülmüştür. En iyi CNN modeli, 900,000 görüntü kullanılarak eğitilmiş olup, tam balık gövdeleri, kısmi balık gövdeleri ve çevreyi içermektedir. Doğru tanımlama oranı %94,90 olarak bulunmuş ve bu oran insanların %89,3 doğruluk oranını geride bırakmıştır. Ayrıca, CNN'nin mercanlar veya diğer balıkların arkasında kısmen gizlenmiş balıkları tanımlama konusunda insanlardan daha etkili olduğu belirlenmiştir. Bu çalışma, su altı görüntülerinde etkili bir balık tanımlama yöntemi olan derin öğrenme yöntemlerinin kullanımının ve balık çeşitliliğini ucuz ve etkili bir şekilde izlemek için yeni video tabanlı protokollerin geliştirilmesinin umut vadettiğini ortaya koymaktadır.

ImageNet üzerinde Hridayami ve ark. (2019) önceden eğitilmiş olan VGG16 gibi derin evrişimli sinir ağlarına dayalı bir balık tanıma ve transfer öğrenme yöntemiyle gerçekleştirdikleri bir çalışmada balık veri seti, her biri 15 görüntüden oluşan 50 türü içermektedir, bunların 10 görüntüsü eğitim amacıyla kullanılmış ve 5 görüntüsü test için ayrılmıştır. Çalışmada, model dört farklı tür veri kümesi üzerinde eğitilmiştir: RGB renk uzayı görüntüsü, canny filtreli görüntü, karışım görüntüsü ve RGB görüntü ile karışım yapılmış görüntü. Sonuçlar, RGB görüntü ile karışım yapılmış görüntü eğitilmiş modelin en iyi gerçek kabul oranı (GAR) değerine sahip olduğunu (%96,40) göstermektedir. Bunu, %92,40 GAR değerine sahip RGB renk uzayı görüntüsü eğitilmiş model izlemekte, ardından %80,4 GAR değerine sahip canny filtreli görüntü eğitilmiş model ve en düşük GAR değeri olan %75,6 GAR değerine sahip karışım görüntüsü eğitilmiş model gelmektedir.

Genellikle denizel ortamlarda balıkçılık faaliyetleri için kullanılan Geniş açı güvenlik kamerası sistemi su altı yaşamının incelenmesi, balık stoklarının değerlendirilmesi, tür sınıflandırması ve çevresel izleme gibi amaçlarla kullanılır. Bu sistem, su altı ortamında gerçek zamanlı görsel verilerin elde edilmesini sağlar ve balıkçılık endüstrisinde, deniz bilimlerinde ve çevre koruma çalışmalarında önemli bir rol oynar. Tarama izindeki bilinen konumlardan elde edilen görüntüleri incelemek, türlerin varlığı için yüksek çözünürlüklü gerçek veri sağlar. Allken ve ark. (2019) yaptıkları çalışmada, Derin Görüş Tarama Kamera Sistemi görüntülerinde bulunan türlerin sınıflandırılmasını otomatikleştirmek için derin öğrenme sinir ağı geliştirilmiş ve eğitim verilerinin azlığı sorununu çözmek için, derin görüş görüntülerinin gerçekçi bir şekilde simülasyonunu temel alan yeni bir eğitim düzeni geliştirilmiştir. Mavi mezgit, Atlantik hamsi ve Atlantik uskumru için %94 sınıflandırma doğruluğuna ulaşan çalışma sentetik verilerin eğitim verilerinin eksikliğini etkin bir şekilde gidermekte etkili olabileceğini göstermektedir.

Jalal ve ark. (2020) çalışmalarında optik akışı, Gaussian karışım modellerini ve YOLO derin sinir ağını birleştiren hibrit bir çözüm modelinde YOLO'nun hareketli balıkları ve arka planda kamuflaj yapmış balıkları tespit etmedeki başarısını arttırmak için Gaussian karışım modelleri ve optik akış yoluyla elde edilen zamansal bilgiler kullanılmıştır. Fish4Knowledge veri tabanındaki LifeCLEF 2015 ve Batı Avustralya Üniversitesi tarafından derlenen bir veri seti üzerinde çalışma gerçekleştirilmiştir. Her iki veri setinde de balık tespit F-skorları sırasıyla %95,47 ve %91,2, balık türü sınıflandırma doğrulukları ise sırasıyla %91,64 ve %79,8 olarak elde edilmiştir.

Serbestçe hareket eden balıkları tespit etmek için Salman ve ark. (2020) Bölge Tabanlı Evrişimli Sinir Ağı(R-CNN) kullanılarak sualtı ortamlarında balıkların hareket bilgileri

kullanılarak sinir ağı eğitilmiş ve ham görüntülerle birleştirilerek balığa bağımlı aday bölgeleri oluşturulmuştur. Fish4Knowledge den temin edilen Karmaşık Sahne veri seti ve LifeCLEF 2015 balık veri seti kullanılarak yapılan testlerde %87,44 ve %80.02 oranında tespit doğruluğu elde edilmiştir.

Hussain ve ark. (2021) dört evrişimli katman ve iki tam bağlantılı katman içeren AlexNet modelinin indirgenmiş bir versiyonunu kullanarak yaptıkları bir çalışmada AlexNet ve VGGNet gibi diğer derin öğrenme modelleriyle karşılaştırma yapılmıştır. Dört parametre dikkate alınmıştır. Bunlar evrişimli katman sayısı, tam bağlantılı katman sayısı, eğitim verilerinde %100 doğruluk elde etmek için yapılan iterasyon sayısı, gruplama boyutu ve atlama katmanı olarak ele alınmıştır. Sonuçlar, önerilen ve değiştirilmiş AlexNet modelinin daha az katmana sahip olmasıyla test doğruluğunun %90,48 olduğunu göstermektedir, orijinal AlexNet modeli eğitilmemiş referans balık veri setinde %86,65 doğruluk elde edilmiştir. Atılma katmanı katmanının dâhil edilmesi, önerilen modelin genel performansını artırmıştır. Ayrıca, daha az eğitim görüntüsü, daha az bellek kullanımı ve daha az hesaplama karmaşıklığına sahiptir.

Akvaryumlar içindeki , balıkların anormal davranışı genellikle kötü su kalitesi, hipoksi veya çeşitli hastalıklardan kaynaklanır. Balıklardaki bu davranışın gecikmeli fark edilmesi, büyük sayıda balık ölümüne yol açmaktadır. Bu nedenle, anormal davranış sergileyen balıkların gerçek zamanlı olarak tespiti ve takibi, balık refahını artırmak ve akuakültürün hayatta kalma oranını ve ekonomik faydalarını artırmak için etkili bir yoldur. H. Wang ve ark. (2022) Porfir deniz levreğinin anormal davranışını tespit etmek ve takip etmek için birleşik bir uçtan uca sinir ağı kullanarak algılama algoritması, hedefin başlangıç değerini takip algoritmasına ileterek, takip algoritması sonraki kareleri izleyerek uçtan uca anormal balık davranışı tespiti gerçekleştirir ve anormal davranış bireylerini yüksek hızda ve doğru bir şekilde izlemektedir. Hedef tespiti bölümünde, YOLOV5s, çok seviyeli özellikleri dahil ederek ve özellik eşlemeyi ekleyerek geliştirilmiştir. Orijinal ağ ile karşılaştırıldığında, algılama doğruluğu AP50:95 8.8% artarken AP50 99.4%'e ulaşmıştır. SiamRPN++ temelinde anormal balıkların çoklu hedefli takibi gerçekleştirilmiştir. Takip doğruluğu %76,70'tir. İki yaklaşımın birleştirilmesiyle, anormal davranış sergileyen bireysel balıklar hassas bir şekilde tespit edilip gerçek zamanlı olarak takip edilebilir.

Süs balığı çiftliklerini izlemek, balık göletini kontrol etmek gibi işler, çiftçiler için oldukça zorlu bir görevdir, çünkü çiftçiler, zaman, çaba ve para maliyetiyle manuel prosedürlerle doğrudan ilgilenmek zorundadır. Günümüz dünyasında her iş, belirli bir süre içinde ve çaba harcayarak yapılır. Çiftçilerin balıkları donmuş su veya karanlık ortamlarda tespit edebileceği bir sistem öneren Patro ve ark. (2023) çalışma için nesne tespiti için gelişmiş

bir model olarak kabul edilen YOLOv5-CNN tespit modeli kullanılmıştır, çünkü hız ve doğruluk açısından R-CNN modelinden daha iyidir. Görüntüler, Labelimg kullanılarak etiketlenmiş ve kodlama işlemi Google Colab'da gerçekleştirilmiştir. Model, balıkları gerçek zamanlı olarak bir olasılık puanıyla tespit etmektedir. Modelin ortalama hassasiyet değeri %86 olarak bulunmuştur.

2.4. Hipotez

Sualtı canlı türlerinin izlenmesi ve sualtı ile ilgili sorunların çözümü için kullanılan geleneksel yöntemler, yapay zeka alanındaki ilerlemelerle birlikte bilgisayarla görü ve derin öğrenme gibi daha modern tekniklere yerini bırakmıştır. Bulanıklık, ışığın kırılması ve gürültü gibi zorluklarla karşılaşılacak sualtı ortamında, YOLO modeli gibi evrimsel sinir ağı tabanlı (ESA) modelleri kullanılarak balon balığının tanınması ve izlenmesi amacıyla balon balığı görsellerini içeren veri seti kullanılmaktadır. Bu çalışmada, çevresel, ekonomik ve insan sağlığına zararları olan Akdeniz'de ciddi bir sorun haline gelen balon balığı problemine güncel teknolojiler ve daha modern yaklaşımlar ile etkili bir çözüm sunabileceği hipoteziyle, balon balığı tespiti yapılacaktır.

2.5. Tezin Organizasyonu

Bu tez çalışmasında, balon balığı tespit etmek amacı ile evrimsel sinir ağı (ESA) tabanlı YOLOv8 ve V5 modelleri kullanılarak, videolardan elde edilen veri seti üzerinde nesne tespiti yapılmış olup sonuçları incelenmiştir. Tez çalışmasının Birinci Bölüm'ünde tezin amacı, kapsamı, literatür taraması, alandaki benzer çalışmalar ve hipotezi açıklanmıştır. İkinci Bölüm' de sualtı ortamlarında balık tespiti için kullanılan YOLO versiyonları incelenmiş ve önerdiğimiz YOLO v8 ve V5 modelleri hakkında temel bilgi verilmiştir. Tezde kullanılan çalışmaya konu olan balon balıkları görselleri ile veri toplama yöntemleri, veri setleri, kullanılan yazılımlar, model eğitimi ve test süreçlerine bu bölümde değinilmiştir. Üçüncü Bölüm' de balon balığı için önerilen YOLOv8 ve v5 modelleri farklı boyutlara sahip sürümleri ile eğitilmiş eğitilen modeller ile nesne tespiti işlemi gerçekleştirilmiş tezin ana sonuçları ve bulguları sunulmuştur. YOLOv8 ve v5 modelinin eğitim başarısı YOLOv8'i de geliştiren Ultralytics şirketinin geliştirdiği önceki YOLO versiyonu olan YOLOv5 modelinin eğitim başarısı ile karşılaştırılmıştır. Dördüncü Bölüm 'de bulguların değerlendirildiği ve yorumlandığı bir tartışma bölümü yer almaktadır. Bu bölümde tezin amacına ulaşıp ulaşılmadığı, yöntemlerin gücü ve sınırlamaları gibi konular ele alınmıştır. Ayrıca gelecekteki araştırmalar ve iyileştirmeler için öneriler sunulmaktadır. Sonuç bölümünde tezin genel

sonuçları ve elde edilen bulguların özetlendiği kısım yer alır. Bu bölümde çalışmanın, sınırlamaları ve sonuçlarına ilişkin genel bir değerlendirme yapılmıştır.



İKİNCİ BÖLÜM

MATERYAL YÖNTEM

3.1. Veri Kaynağı

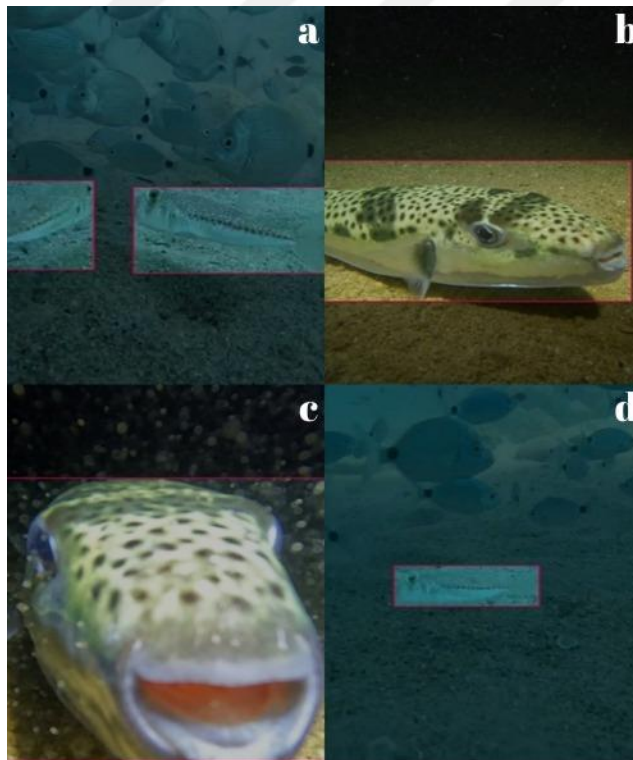
Bu tez çalışmasında balon balığı tespiti için kullanılan videolar Akdeniz bölgesinde faaliyet gösteren dalış okullarından ve profesyonel dalış eğitmenlerinden temin edilmiştir. Çalışma kapsamında elde edilen videolar YOLO modelinin eğitimi için fotoğraf karelerine dönüştürülmüştür. Videoların fotoğraf karelerine dönüştürülmesi için gerekli kodlar Python programlama dili kullanılarak PyCharm IDE (Entegre Geliştirme Ortamı)'nda yazılmış, OpenCV kütüphanesinden faydalanılmıştır. Eğitim için veri seti oluşturulmadan önce fotoğraf karelerinin etiketlenmesi gerekmektedir. Bu görsellerin etiketlenmesinde Roboflow platformunun sunduğu etiketleme arayüzü kullanılmıştır. Etiketlenen görsellerde otomatik yön düzeltme(auto-orient) işlemi yapılmıştır. YOLOv8 ve YOLOv5 modelleri 640x640 boyutundaki görselleri eğitimde kullanabildiği için yeniden boyutlandırma (resize) işlemi ile görseller 640x640 boyutuna dönüştürülmüştür. Görseller üç parçaya ayrılmıştır. Bu parçalar veri setinin eğitim (train), doğrulama (validation) ve test parçalarıdır. YOLO versiyonunun eğitiminde kullanılacak görsel çeşitliliğini arttırmak için veri artırma (data augmentation) işlemi veri setinin eğitim (train) parçası üzerinde gerçekleştirilmiştir. Görsel artırma için yatay aynalama (horizontal flip), dikey aynalama(vertical flip), saat yönünde ve saat yönünün tersinde 90 derece açıyla döndürme işlemleri seçilmiştir. Özellikle sınırlı miktarda veriye sahip olduğumuz durumlarda veri artırma modelin daha iyi genelleme yapmasını ve aşırı uyum (overfitting) sorununu azaltmasını sağladığı gözlemlenmiştir (Alimovski & Erdemir, 2021).Veri artırma sonrasında veri seti 1773 eğitim görseli, 450 doğrulama görseli ve 250 test görselinden oluşmaktadır. YOLOv8 modelinin eğitiminde ve bu modelin eğitim başarısının karşılaştırılmasında kullanılan YOLOv5 modelinin eğitiminde bu veri seti kullanılmıştır.

Bu çalışma Akdeniz Üniversitesi Bilimsel Araştırma ve Yayın Etiği Kurulu'nun 05.09.2023 tarih ve 370 sayılı kararıyla etik kurul izni alınmıştır.



Şekil 3.1 Roboflow görsel etiketleme aracı

Şekil 2.1’de Roboflow sitesine yüklenen görsellerin etiketlenmesinde kullanılan etiketleme aracı görülmektedir. Videolardan elde edilen fotoğraf karelerinde balon balıkları sınırlayıcı kutular ile işaretlenmiş ve sınıf etiketleri ile etiketlenmiştir. Etiketleme işlemi manuel şekilde insan eliyle gerçekleştirilmiştir.



Şekil 3.2 Veri setinden etiketlenmiş bazı balon balığı görselleri

Şekil 2.2’de Roboflow sitesinde etiketlenmiş örnek balon balığı görselleri verilmiştir. Görsellerde görüldüğü üzere etiketlenen balon balıkları farklı sualtı ortamlarında kayıt altına alınmıştır. Şekil 2.2 a etiketli görselde balon balıklarının diğer balık türleri ile olduğu ortam

içerisine alma işlemlerinde ağırlık balık gövdesine verilmiştir, balıkların yüzgeç ve kuyrukları etiketlemelerde daha az dikkate alınmıştır.

3.2. Python

Python programlama dili, Guido van Rossum tarafından Amsterdam'daki Centrum Wiskunde & Informatica (CWI) araştırma enstitüsünde 1980'lerin sonlarında geliştirilmeye başlanmıştır. Daha basit ve okunabilir bir dil oluşturma amacıyla hayata geçirilen bu projenin ilk sürümü, 0.9.0 sürümü ile 20 Şubat 1991'de yayımlanmıştır. Python, ABC dilinden ilham alarak tasarlanmıştır (Van Rossum, 2007). 2008'de Python 3.0 sürümü yayımlanmıştır, yapay zeka, veri bilimi, web geliştirme ve bilgisayar ağları gibi birçok farklı alan ve endüstride popülerlik kazanmış ve tercih edilmiş bir programlama dilidir (Nagpal & Gabrani, 2019). Tez çalışmasında Python programlama dilinin 3.10.12 sürümü kullanılmıştır.

3.3. PyCharm IDE

PyCharm, JetBrains şirketi tarafından geliştirilen ve Python programlama dili için özel olarak tasarlanmış bir bütünleşmiş geliştirme ortamıdır (IDE). Python programlama dilini kullanan projelerinin oluşturulması, kod düzenleme, hata ayıklama, test etme, derleme ve dağıtım gibi birçok geliştirme sürecini desteklemektedir. Pycharm bütünleşmiş geliştirme ortamının ücretli ve ücretsiz sürümleri bulunmaktadır. Biz çalışmada PyCharm geliştirme ortamının normalde ücretli olan ve üniversite öğrencilerinin de öğrenci e-posta adresleri ile yıllık ücretsiz üyelik kapsamında kullanabildikleri Pycharm Professional 2023.2.1 sürümü kullandık. Ücretsiz olan Community Edition ile ücretli olan Professional Edition arasında çalışmamızda etki edecek bir fark bulunmamaktadır. Bizim çalışmada Professional sürümünü tercih etmemizin tek sebebi JetBrains şirketinin üniversite öğrencileri için sağladığı ücretsiz lisansdır (Saabith vd., 2021).

3.4. OpenCV (Open Source Computer Vision)

OpenCV (Open Source Computer Vision) bir bilgisayarla görü ve görüntü işleme kütüphanesidir. Bilgisayarla görü, görüntülerin ve videoların analizi, anlama ve işlenmesiyle ilgilenen bir alandır. OpenCV, bu alandaki birçok işlemi gerçekleştirmek için kullanılan bir araç seti sunar. OpenCV, 1999 yılında Intel tarafından başlatılan bir proje olarak ortaya çıkmıştır ve o zamandan beri sürekli olarak geliştirilmekte ve topluluk tarafından desteklenmektedir. Başlangıçta C ve C++ dillerinde geliştirilen OpenCV, günümüzde birçok farklı programlama dili için bağlantılar ve arabirimler sunmaktadır, bu nedenle Python, Java, MATLAB gibi dillerde de kullanılabilir (Kaehler & Bradski, 2016).

3.5. Roboflow Platformu

Roboflow, kullanıcıların ücretli ya da ücretsiz kullanabileceği bilgisayar vizyonu projeleri oluşturmayı ve yönetmeyi kolaylaştıran bir platformdur. Roboflow, görüntülerle ilgili projelerde baştan sona çözümler sunmaktadır. Kullanıcılar, görüntü verilerini yükleyebilmekte, veri setlerini etiketleyebilmekte, veri setlerini düzeltebilmekte, model eğitebilmekte ve sonuçları değerlendirebilmektedir. Roboflow'un sunduğu hizmetler arasında veri artırma, veri seti yönetimi, model eğitimi ve çıkarımı, model değerlendirme, model konfigürasyonları ve uygulama entegrasyonları gibi işlevler bulunmaktadır. Bu şekilde, araştırmacılar, geliştiriciler ve veri bilimciler, bilgisayar vizyonu projelerini daha etkili bir şekilde yönetebilir ve geliştirebilmektedir. Tez çalışmasında Roboflow platformu görsellerin etiketlenmesinde ön işlenmesinde, veri artırma işlemlerinin yapılmasında ve platforma ait API (Application Programming Interface-Uygulama Programlama Arayüzü) aracılığıyla YOLO formatına uygun olarak etiketlenmiş görselleri içeren veri setinin Google Colab ortamında model eğitiminde kullanılmasına olanak sağlamıştır. Google Colab'ta gerçekleştirilen çalışmanın sonuçları Roboflow üzerinden görüntülenebilmekte ve aynı zamanda Roboflow platformu tarafından ücretsiz bir şekilde sunulan Roboflow 3.0 nesne tespit modeli ile de eğitim gerçekleştirebilmekte ve eğitilen YOLO model sonuçları Roboflow'un kendine ait olan nesne tespit modelinin eğitim sonuçları ile istenildiği takdirde karşılaştırılabilmektedir (Čakić vd., 2022).

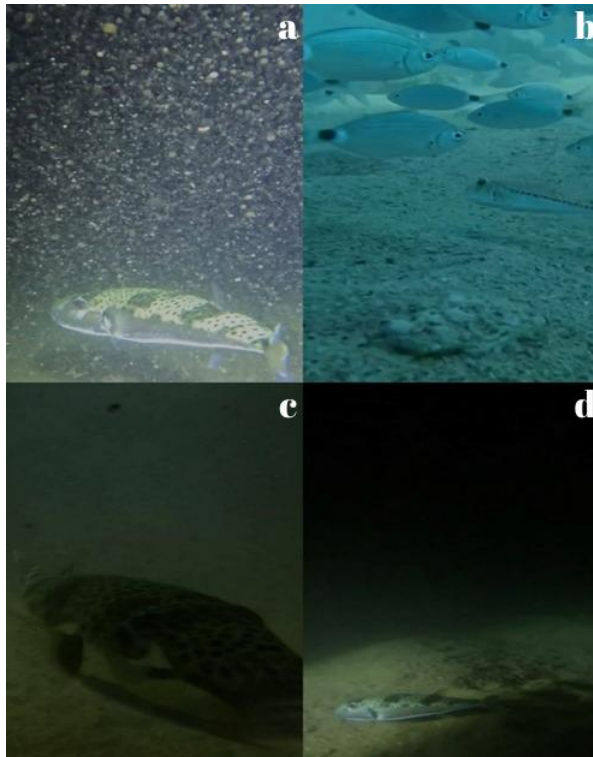
3.6. Google Colab

Tez çalışmasında eğitim ortamı olarak Google Colab seçilmiştir. Google Colab (Colaboratory), Google şirketi tarafından sunulan bir bulut tabanlı Jupyter Notebook hizmetidir. Google Colab, araştırma ve geliştirme yapmak için kullanıcıların Python kodlarını, metinleri, grafikleri ve verileri etkileşimli bir şekilde çalıştırmalarını sağlamayı amaçlamaktadır. Kullanıcıların web tarayıcıları üzerinden Python kodlarını yazıp çalıştırabilecekleri etkileşimli bir ortam sunar. Google'ın sunucularında çalışır, bu nedenle kullanıcıların kendi bilgisayarlarına Python veya çeşitli kütüphaneleri yüklemelerine gerek kalmaz. Bulut tabanlı olması, yüksek işlem gücü ve bellek gerektiren derin öğrenme projelerinde gerçekleştirilen çalışmalarda avantaj sağlamaktadır. Ücretsiz bir hizmettir ve Google hesabı olan herkesin kullanımına açıktır. Kullanıcılar, Colab üzerindeki tüm temel özelliklere ve kaynaklara ücretsiz olarak erişebilirler. Python programlama dilinin tüm yaygın kütüphaneleri (numpy, pandas, matplotlib, TensorFlow, PyTorch, Keras, scikit-learn vb.) kolayca kullanılabilir durumdadır. Colab, kullanıcılara Jupyter defterini paylaşma ve iş birliği

yapma imkânı sunar. Bu sadece araştırmacılar çalışmalarını birlikte yürütebilir ve başkalarıyla aynı anda aynı defter üzerinde çalışabilir ve gerçek zamanlı olarak kod düzenlemelerini gerçekleştirebilirler. Bu, proje ekipleri, araştırmacılar veya öğrenci grupları arasında kolay iş birliği yapılmasını sağlar. Google Colab, kullanıcılara dosyaları ve veri kümesini kolayca yükleyip saklamalarını sağlar. Google Drive entegrasyonu sayesinde, veri setlerini depolayabilir ve çeşitli projelerde kullanabilirsiniz. Ayrıca, Colab'ın birçok popüler hizmet ve API ile entegrasyonu vardır, bu da verileri hızlı ve kolay bir şekilde içe aktarmanıza ve dışa aktarmanıza olanak tanır (Carneiro vd., 2018).

3.7. Sualtı Ortamı Karşılaşılan Zorluklar

Sualtı ortamında balon balığı tespiti için yeterli sayıda doğru etiketlenmiş örnek bulmak zor olmaktadır. Bu, modelimizi eğitmek ve doğru sonuçlar elde etmek için gereken veriye ulaşmamızı zorlaştırmaktadır. Sualtında çekilen görüntüler genellikle düşük ışık koşullarında veya bulanık olabilmektedir. Bu da balon balığı dâhil sualtı ortamında balık türlerinin tespitini zorlaştırmaktadır. Görüntü kalitesinin düşük olması, etiketleme zorluklarına ve modelimizin balon balığı örneklerini doğru bir şekilde algılamasına engel olabilmektedir. Sualtında balon balıkları farklı perspektiflerden ve boyutlardan görülebilir. Bazı balon balıkları yakından çekilirken, diğerleri uzaktan çekilmiştir bu modelin bu farklı perspektifleri ve boyutları tanıyabilmesi için ek verilere ihtiyaç duymasına sebep olmaktadır.



Şekil 3.4 Zorlu sualtı ortamlarını gösteren temsili örnekler

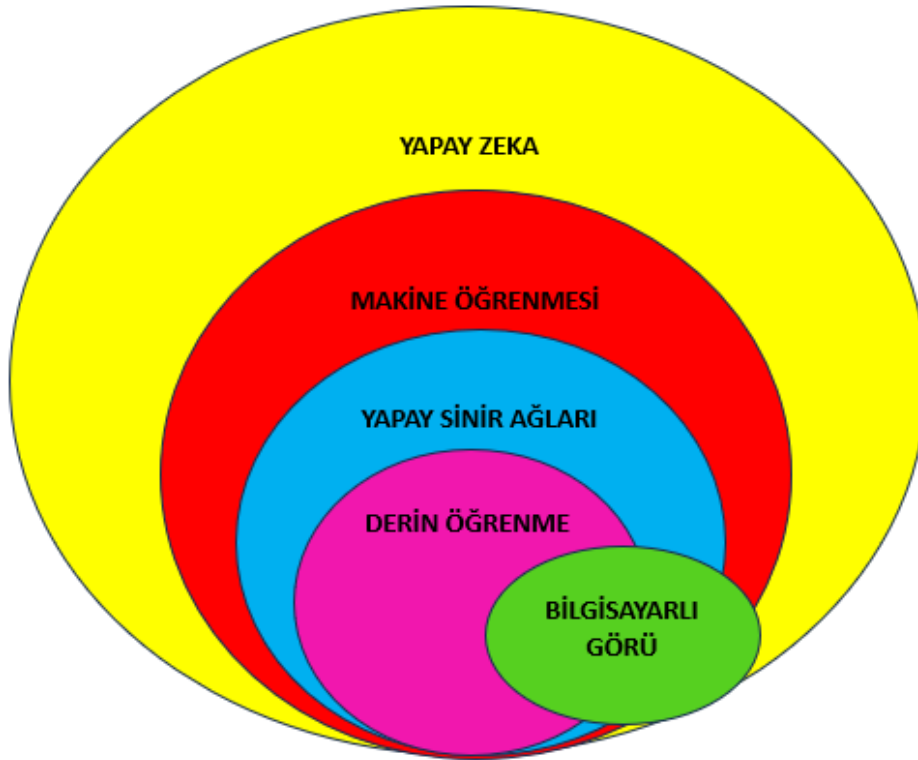
Sualtı ortamı doğal olarak karmaşık bir ortamdır. Yosunlar, kayalar, mercanlar gibi nesneler, balon balığı tespitini zorlaştırabilir ve modelin doğru sonuçlar üretmesini engelleyebilir. Modelin çevredeki gürültüyü (noise) filtreleyebilmesi ve yalnızca balon balığına odaklanabilmesi başarıyı etkileyen faktörlerdendir. Şekil 2.4'te yer alan görsellerde bahsedilen zorlu sualtı ortamlarına ait durumlara örnekler verilmiştir. Şekil 2.4 a'da gürültünün bulunduğu gece kayıt altına alınan su altı ortamı görülmektedir. Şekil 2.4 b'de başka balıklarında olduğu ve balon balığının deniz tabanındaki kum tabakasıyla benzer görünüşte olduğu ortam görülmektedir. Balığın kum yüzeye yaklaştıkça çevresinden ayırt edilmesi zorlaşmaktadır. Şekil 2.4 c'de ışığın yetersiz olduğu ortam gözükmemektedir. Şekil 2.4 d'de yapay ışık ile balığın kayda alındığı karanlık sualtı ortamı görülmektedir.

3.8. Nesne Tespit Modeli

Tez çalışmasında balon balığı tespit etmek için eğittiğimiz YOLOv8 modeli hakkında bilgi verilmeden önce yapay zeka, makine öğrenmesi, derin öğrenme ve evrimsel sinir ağları konularına değinilmiştir. Şekil 2.5'te aralarındaki ilişkiyi içeren görsel yer almaktadır.

Nesne tespiti (object detection), bilgisayarlı görü işleme alanında kullanılan bir tekniktir. Bu teknik, dijital görüntülerdeki nesneleri tanımlama ve sınıflandırma yeteneği sağlar. Nesne tespiti algoritmaları, bir görüntüde bulunan nesneleri algılamak için kullanılır. Bu nesneleri algımlarken, nesnelerin konumlarını genellikle dikdörtgen kutular(sınırlayıcı kutu) içine alarak, türlerini ve sayılarını belirtmektedir (Grauman & Leibe, 2011; Jafri vd., 2014; Mohan vd., 2001).

Bilgisayarlı görü, bilgisayarların ve yazılım algoritmalarının görüntülerle çalışabilme yeteneği olarak tanımlanır. Nesne tespiti, bilgisayarlı görünün önemli bir uygulama alanıdır çünkü birçok uygulamada nesneleri algılama ve tanıma gereklidir. YOLO nesne tespiti için kullanılan popüler ve önemli algoritmalarından birisidir (Hussain, 2023; Terven & Cordova-Esparza, 2023).



Şekil 3.5 Yapay zeka, makine öğrenmesi, yapay sinir ağları, derin öğrenme, bilgisayarlı görü ilişkisi

3.8.1. Yapay Zeka

Yapay zeka, bilgisayar sistemlerinin insan benzeri zekaya sahip olma yeteneğini anlamaya çalışan bir bilim dalıdır. Yapay zeka alanındaki çalışmaların tarihi 20. yüzyılın ortalarına kadar uzanmaktadır. 1940'ların sonları ve 1950'ler, yapay zeka da ilk adımların atıldığı yıllardır. Alan Turing'in "Bilgisayarlar ve Zeka" adlı makalesi, yapay zekanın temelini atan önemli bir çalışma olmuştur. Turing, bir bilgisayarın zeki davranışı taklit edebilme yeteneğine sahip olup olamayacağını sorgulayan ünlü Turing Testi'ni geliştirmiştir (Turing, 2021).

1960'lar ve 1970'lerde, yapay zeka araştırmalarında daha karmaşık problemleri çözmek için kullanılan bilgisayar programları geliştirilmeye odaklanılmıştır. Oyunları oynayabilen ve dil yapısını anlayabilen programların ilk örnekleri bu dönemde ortaya çıkmıştır (Jones, 1994). Bu dönemde yapay zeka araştırmaları belirli sınırlamalarla karşılaşmıştır. O yıllarda bilgisayarların bellek ve işlem gücünün sınırlı olmasıyla karmaşık problemleri çözmek için yeterli olmadığı anlaşılmıştır (Lesgold & Mandl, 1988).

1980'lerden itibaren, yapay zeka alanında yaşanan canlanma, özellikle uzman sistemler, doğal dil işleme ve veri tabanı yönetimi gibi alanlarda önemli ilerlemelere yol açmıştır. Yapay zeka uygulamaları, endüstriyel ve ticari alanlarda da kullanılmaya başlanmıştır ancak, bu

dönemde yaşanan ekonomik durgunluk ve bazı projelerin beklenen başarıyı elde edememesi yapay zeka alanındaki ilginin azalması ile alandaki çalışmaların duraklamasına neden olmuştur(de la Fuente Garcia vd., 2020; Frank vd., 2019; Sutton vd., 2016; Webb, 2019; Yasnitsky, 2020).

2000'li yılların başlarından itibaren, yapay zeka alanında büyük bir patlama yaşanmıştır. Bilgisayarların gücü arttıkça, derin öğrenme ve sinir ağları gibi teknolojiler geliştirilmiştir. Bu teknolojiler, büyük veri setlerini analiz edebilme ve karmaşık problemleri çözebilme yeteneklerini büyük ölçüde artırmıştır. Ayrıca, yapay zeka uygulamaları mobil cihazlar, otomobiller, sağlık hizmetleri, finans ve diğer birçok endüstriyel sektörde yaygın olarak kullanılır hale gelmiştir (Ahmad vd., 2021; Bharadiya, 2023; Crawford, 2021; Duan vd., 2019; Hwang, 2018; Lu, 2019; Zhang & Lu, 2021).

3.8.2. Makine Öğrenmesi

Temelleri, 1950'lerde ve 1960'larda istatistiksel öğrenme teorisi ve yapay zeka araştırmaları ile atılmıştır. Makine öğrenmesi alanındaki büyük ilerlemeler 2000'lerden itibaren yaşanmıştır. Bu dönemde, büyük veri setlerinin ve güçlü bilgisayar sistemlerinin kullanılabilir hale gelmesiyle birlikte, derin öğrenme gibi karmaşık algoritmalar geliştirilebilmiştir (Audry, 2021).

Makine öğrenmesi istatistik, veri madenciliği, matematik ve bilgisayar bilimleri gibi alanların kesişim noktasında yer almaktadır. Temel amacı, veri setlerindeki örüntüleri tanıyarak gelecekteki olayları tahmin etmektir. Bu tahmin yeteneği, genellikle algoritmalar ve istatistiksel modeller aracılığıyla gerçekleştirilir. Makine öğrenmesi, deneme-yanılma yöntemlerine dayanan bir öğrenme sürecine sahiptir. Algoritmalara doğru parametreler verildiğinde, sistem, verilerdeki örüntüleri algılayabilir ve gelecekteki verileri tahmin edebilir (Fayyad vd., 1996; Friedman, 1998; Ivezić vd., 2020).

Makine öğrenmesi, denetimli, denetimsiz ve takviyeli öğrenme olmak üzere temel üç kategoriye ayrılır. Denetimli öğrenme, sistemlerin etiketlenmiş veri setleri üzerinden öğrenme yapmasını içerir. Bu veri setlerinde giriş verileri (input) ile doğru çıktılar (output) beraber bulunur. Denetimli öğrenme, genellikle sınıflandırma (classification) ve regresyon (regression) problemleri için kullanılır (Mahdavinejad vd., 2018; Mahesh, 2020; L. Zhou vd., 2017).

Denetimsiz öğrenme, etiketlenmemiş veri setleri üzerinde çalışır. Bu tür veri setlerinde çıktılar (etiketler) bulunmaz. Sistem, veri setindeki örüntüleri kendiliğinden keşfeder. Kümelenme (clustering) ve boyut indirgeme (dimensionality reduction) bu kategoriye örnek

olarak verilebilir. Bu tür öğrenme, veri setindeki doğal yapıları anlamak için kullanılır (Alashwal vd., 2019; Patel, 2019).

Takviyeli öğrenme, bir ajanın belirli bir çevrede ödül maksimizasyonu yapmasını amaçlar. Bu öğrenme türünde, ajan bir çevrede çeşitli aksiyonlar gerçekleştirir ve çevreden gelen geri bildirimlere göre öğrenir. Bu öğrenme türü, oyun teorisi, robotik ve otonom araçlar gibi alanlarda kullanılır (Aradi, 2020; Kiran vd., 2021; Zhao vd., 2020).

Günümüzde, makine öğrenmesi birçok endüstride önemli uygulamalara sahiptir. Sağlık sektöründe, kanser teşhisi ve tedavisi gibi alanlarda makine öğrenmesi teknikleri kullanılarak hastalıkların erken teşhisi ve tedavi yöntemlerinin belirlenmesi üzerinde çalışmalar yapılmaktadır (Bakrania vd., 2023; Bhardwaj vd., 2017; Coccia, 2020; Ngiam & Khor, 2019). Finans sektöründe, piyasa tahmini ve risk yönetimi gibi konularda makine öğrenmesi algoritmaları kullanılarak analizler gerçekleştirilmektedir (Goodell vd., 2021; Kou vd., 2019; Leo vd., 2019). Otomotiv endüstrisinde, otonom araçlar için makine öğrenmesi teknikleri kullanılarak araçların çevrelerini algılaması ve güvenli bir şekilde hareket etmesi sağlanır (Arnold vd., 2019; Grigorescu vd., 2020; Sallab vd., 2017). Doğal dil işleme alanında, makine öğrenmesi, otomatik çeviri, dil modelleme ve metin analizi gibi birçok uygulamada kullanılmaktadır (Otter vd., 2020; Young vd., 2018).

3.8.3. Yapay Sinir Ağları

Sinir ağları ya da bilgisayar bilimlerinde diğer kullanılan adıyla Yapay Sinir Ağları (YSA), insan beyninin bilgi işleme yeteneklerinden ilham alınarak geliştirilmiş güçlü matematiksel modellerdir. Biyolojik nöronların çalışma şeklini taklit eden YSA'lar, sinaptik bağlantıların dijital olarak modellenmesiyle oluşturulur. Nöronlar, çeşitli şekillerde birbirlerine bağlanarak öğrenme, hafıza ve veri analizi gibi işlemleri gerçekleştirebilen ağlar oluşturur. YSA'lar, insanın doğal düşünme ve gözlem yeteneklerini taklit ederek karmaşık problemleri çözebilme yeteneğini bilgisayarlara kazandırır. Bu teknoloji, insan beyninin yaşayarak ve deneyerek öğrenme yeteneğini temel alır (Maind & Wankar, 2014).

İnsan doğduğu andan itibaren çevresindeki deneyimlerle sürekli olarak öğrenme sürecine girer. Öğrenme süreci, nöronlar arasındaki sinaptik bağlantıların ayarlanması ile gerçekleşir. Beyin, yaşantılarla şekillenir, sinaptik bağlantıları ayarlar ve yeni bağlantılar oluşturur. YSA'lar bu benzer prensibi taklit eder. Öğrenme, eğitim verileriyle gerçekleştirilir. Bu süreçte, ağın sinaptik bağlantıları, eğitim algoritmaları tarafından girdi ve çıktı verileri kullanılarak tekrar tekrar ayarlanır (Abraham, 2005) (Kasai vd., 2021).

YSA'lar, birbiriyle bağlantılı işlem birimlerinden (nöronlar) oluşan matematiksel sistemlerdir. Her bir işlem birimi, diğer nöronlardan aldığı bilgiyi birleştirir, dönüştürür ve sayısal bir çıktı üretir. Bu işlem birimleri, gerçek nöronlara benzer şekilde çalışır ve bir ağ içinde birbirleriyle bağlanır. Sinir ağları, dağıtılmış, adaptif ve doğrusal olmayan işlemler temelinde çalışır. YSA'lar, geleneksel işlemcilerden farklı olarak, çok sayıda basit işlem biriminden oluşurlar ve her biri büyük problemin bir parçasıyla ilgilenir. Bu yapısı sayesinde YSA'lar, paralel ve dağıtılmış işlem kapasitesiyle karmaşık problemleri etkili bir şekilde çözebilirler (Albawi vd., 2017; Kukreja vd., 2016).

Sinir ağlarının davranışı, nöronların transfer fonksiyonlarından ve bu nöronların birbirleriyle nasıl bağlandığından etkilenir. Ayrıca, ağdaki bağlantıların ağırlıkları, ağın temel davranışını şekillendirir. Bu şekilde, YSA'lar, girdi verileriyle çıktı verileri arasındaki karmaşık ilişkileri modelleyebilir ve bu özellikleriyle bilgi işleme, görüntü ve ses tanıma gibi birçok alanda etkili bir şekilde kullanılabilmektedirler (Bengio vd., 2015; Liu vd., 2017).

3.8.4. Derin Öğrenme

Derin öğrenme, yapısında birden fazla yapay sinir ağını bulunduran çok katmanlı yapıya sahip bir makine öğrenmesi alt dalıdır. Büyük veri setlerindeki karmaşık örüntüleri algılamak, temsil etmek ve öğrenmek için kullanılır. Derin öğrenme algoritmaları, veri setlerindeki yüksek seviyeli özellikleri otomatik olarak öğrenmek için çok katmanlı sinir ağları kullanır. Bu, daha kenarlar, renkler vb. düşük seviyeli özelliklerin kombinasyonlarından daha karmaşık ve soyut olan yüz tanıma, nesne tanıma gibi özelliklerin öğrenilmesine olanak tanımaktadır (Guo vd., 2016; LeCun vd., 2015; Pouyanfar vd., 2018).

Derin öğrenme, özellikle büyük veri setlerinde ve karmaşık problemlerde geleneksel makine öğrenmesi tekniklerinden daha etkili olmaktadır. Görüntü tanıma, doğal dil işleme, ses tanıma ve oyun stratejileri gibi birçok uygulama alanında kullanılmaktadır. Derin öğrenme, aynı zamanda otomatik özellik mühendisliğine ihtiyaç duymaz, yani, veri setindeki önemli özellikleri manuel olarak seçmek veya tanımlamak yerine, bu özellikleri veriden otomatik olarak çıkarabilmektedir (Guo vd., 2016; LeCun vd., 2015; Pouyanfar vd., 2018).

Temel olarak, derin öğrenme, bir sinir ağı modelini çok sayıda katmana sahip yapılara dönüştürme yeteneğine dayanır. Her bir katman, önceki katmanın çıktıları alır, bu çıktıları belirli bir şekilde işler, dönüştürür ve ardından bir sonraki katmana iletir. Bu şekilde, veri setindeki karmaşık örüntüler katmanlar arasında giderek daha soyut ve yüksek seviyeli özelliklere dönüştürülür. Bu soyut özellikler, veri setindeki temsil yeteneğini artırır ve sonuçta

daha doğru tahminler veya sınıflandırmalar yapmak için kullanılabilir (Guo vd., 2016; LeCun vd., 2015; Miikkulainen vd., 2019; Shinde & Shah, 2018).

Derin öğrenme modelleri, büyük miktarda veriden öğrenerek çalışır. Bu veriler, görüntüler, sesler, metinler veya diğer veriler olabilir. Model, bu verilerden kalıplar ve ilişkiler öğrenir. Derin öğrenme modelleri, öğrendikleri kalıpları ve ilişkileri kullanarak, yeni verileri tahmin edebilir veya sınıflandırabilir ancak derin öğrenme modelleri, geleneksel makine öğrenimi modellerine göre daha fazla veriye ihtiyaç duymaktadır. Derin öğrenme modelleri, geleneksel makine öğrenimi modellerine göre daha fazla hesaplama gerektirir. Derin öğrenme modelleri, geleneksel makine öğrenimi modellerine göre daha karmaşıktır ve genellikle büyük miktardaki veri setlerinde ve yeterli hesaplama gücüne sahip bilgisayar sistemlerinde etkili bir şekilde çalışır (Najafabadi vd., 2015; Xiao vd., 2015; Zhang vd., 2018).

3.8.5. Evrişimli Sinir Ağları

Bilgisayar bilimlerinde, özellikle de yapay zeka, makine öğrenmesi ve derin öğrenme alanında, evrişimsel sinir ağları (ESA) son yılların en etkili ve yaygın kullanılan algoritmalarından biridir (Abiodun vd., 2018). Evrişimsel sinir ağları, özellikle görüntü tanıma ve analizinde büyük bir başarı elde etmiştir, ancak bununla sınırlı kalmayıp birçok farklı alanda da kullanılmaktadır (Liu vd., 2017). Evrişimsel sinir ağları, biyolojik sinir sistemlerinden esinlenen, çok katmanlı bir yapay sinir ağı türüdür (Kriegeskorte, 2015). Temelde, evrişimsel sinir ağları, görüntü verilerindeki özellikleri öğrenmek ve tanımak için tasarlanmıştır. Bu ağlar, özellik çıkarımı yapmak için konvolüsyon katmanları, aktivasyon fonksiyonları, havuzlama katmanları ve tam bağlantılı katmanlardan oluşur (Albawi vd., 2017).

Evrişimsel sinir ağları, öğrenme sürecinde gerçek veri ile eğitilir. Bu süreçte, ağ, belirli bir problemi çözmek için gerekli olan özellikleri otomatik olarak öğrenir. Eğitim verileri, ağın ağırlıklarını ve bias değerlerini ayarlamak için kullanılır. Eğitim süreci ilerledikçe, ağ doğru tahminler yapabilme yeteneğini geliştirir. Evrişimsel sinir ağlarının bu öğrenme yeteneği, büyük veri setleri üzerinde yapılan derin öğrenme (deep learning) çalışmalarında çok değerli hale gelmiştir (Pinaya vd., 2020).

Evrişimsel sinir ağları (ESA) temelde dört ana tür katmandan oluşur: Giriş Katmanı (Input Layer), Evrişim Katmanları (Convolutional Layers), Havuzlama Katmanları (Pooling Layers) ve Tam Bağlantılı Katmanlar (Fully Connected Layers). Bu katmanlar, görsel verilerde (genellikle görüntülerde) özellikleri çıkarmak ve sınıflandırmak için tasarlanmıştır (Krizhevsky vd., 2012).

3.8.5.1. Giriş Katmanı

Giriş katmanı, ağı dış dünyadan (örneğin, bir görüntü) gelen veriyi alır. Giriş katmanında, görüntüler gibi veri öğeleri pikseller (veya özellik vektörleri) şeklinde temsil edilir. Giriş katmanı, verinin doğru boyut ve formatlara dönüştürülmesinden sorumludur, böylece evrimsel sinir ağının ilk evrişim katmanı ile uyumlu olur (Krizhevsky vd., 2012).

3.8.5.2. Evrişim Katmanları

Evrışim katmanları, görüntülerde veya diğer veri türlerindeki lokal desenleri algılamak için kullanılır. Bu katmanlardaki her bir birim (nöron), giriş verisinde (örneğin, bir görüntüde) belirli bir bölgeye odaklanarak bu bölgedeki desenleri öğrenir. Bu desenler, filtreler veya çekirdekler (kernels) olarak adlandırılır ve evrişim işlemi ile giriş verisi üzerinde kaydırılır. Bu sayede, evrişim katmanları veriden özellik haritalarını (feature maps) çıkarır (Krizhevsky vd., 2012).

3.8.5.3. Havuzlama Katmanları

Havuzlama katmanları, evrişim katmanlarının çıkardığı özellik haritalarını küçültmek ve aynı zamanda çıkarılan özellikleri daha genelleştirmek için kullanılır. Tipik olarak maksimum havuzlama veya ortalama havuzlama gibi tekniklerle çalışırlar. Maksimum havuzlama, belirli bir bölgedeki en büyük özellik değerini alırken, ortalama havuzlama ise belirli bir bölgedeki özellik değerlerinin ortalamasını alır. Havuzlama, hesaplama yükünü azaltır ve özelliklerin konum değişikliklerine karşı daha dirençli hale gelmelerini sağlar (Krizhevsky vd., 2017).

3.8.5.4. Tam Bağlantılı Katmanlar

Tam bağlantılı katmanlar, evrişim katmanlarından elde edilen özellikleri kullanarak sınıflandırma yapar. Bu katmanlar, öğrenilen özellikleri belirli sınıflara (örneğin, kedi veya köpek gibi) atayan bir dizi ağırlık ile ilişkilendirilir. Bu ağırlıklar, ağı eğitim sürecinde veri setinden öğrenilir. Tam bağlantılı katmanlar genellikle bir veya daha fazla gizli katman (hidden layer) içerir ve genellikle son katmanda softmax aktivasyon fonksiyonu kullanılarak sınıflandırma sonuçları elde edilir (Krizhevsky vd., 2017).

3.8.6. YOLO

Evrimsel sinir ağları mimarisine dayalı nesne tanıma ve sınıflandırma algoritmaları genel olarak iki aşamalı ve tek aşamalı olarak ayrılmışlardır. İki aşamalı algoritmalar çalışma sırasında öznitelik çıkarımı ve görüntü tespiti olmak üzere iki aşamadan oluşur ve doğruluk

noktasında tek aşamalı algoritmalarından daha yüksek tanıma doğruluğuna ulaşabilir ancak ilgi bölgesi çıkarma aşamasında elde edilen artan hesaplama gücü, algoritmanın hedefi tespit etme hızını da sınırlar (Sultana vd., 2020). İki aşamalı bir algoritma ile karşılaştırıldığında, tek aşamalı olan YOLO hedefi doğrudan görüntü üzerinde tespit etme ve sınırlayıcı kutu regresyon işlemlerini gerçekleştirme yoluyla daha yüksek bir hedef tespit hızına ulaşmaktadır. Bununla birlikte, tanıma doğruluğu, iki aşamalı algoritma modelinden biraz daha düşüktür. YOLO, geleneksel nesne algılama yöntemlerinden farklı olarak, bir görüntüyü tek bir işlemle analiz ederek nesne algılama ve sınıflandırmayı gerçekleştirir. Bölge tabanlı yöntemlerin aksine, YOLO tüm nesne algılama sürecini bir aşamada tamamlar. Bu hem hızlı hem de gerçek zamanlı uygulamalar için avantaj sağlar (Redmon vd., 2016).



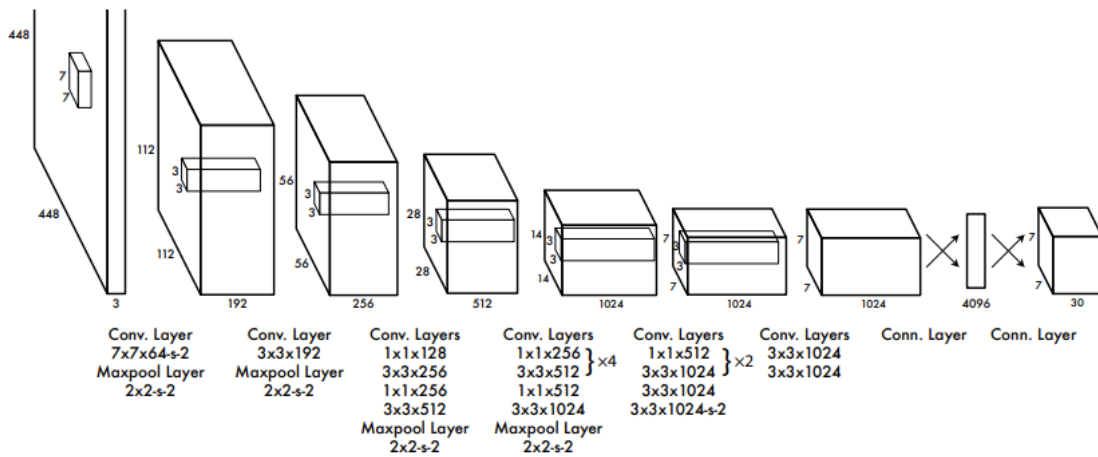
Şekil 3.6 YOLO tek aşamada nesne tespit işlemi (Redmon vd., 2016)

YOLO algoritması, görüntüyü bir ızgara şeklinde böler ve her hücrede potansiyel nesnelerin varlığını ve sınıflarını tahmin etmek için tek aşamalı ESA tabanlı bir algoritma kullanır. Her hücre, bir veya daha fazla nesne sınıfı için sınıflandırma tahmini yapar ve aynı zamanda nesnenin konumunu ve boyutunu da tahmin eder. Böylece, tek bir geçişte tüm nesne bilgileri elde edilir. YOLO'nun bu tek seferlik yaklaşımı, nesne algılama sürecini hızlandırırken aynı zamanda genel bağlamı da dikkate alır. Diğer yöntemler genellikle belli bölgelerde nesne araması yaparken, YOLO tüm görüntüyü aynı anda analiz eder ve böylece nesnelerin genel bağlamını daha iyi anlar. YOLO'nun performansını değerlendirmek için PASCAL VOC veri seti ve COCO veri seti gibi yaygın kullanılan veri setleri kullanılmıştır. PASCAL VOC, bilgisayarla görü işleme ve öğrenim alanında yaygın olarak kullanılan bir veri setidir. "PASCAL Visual Object Classes" (PASCAL Görsel Nesne Sınıfları) olarak bilinen bu veri seti, nesne tespiti, nesne sınıflandırma ve nesne segmentasyonu gibi görevler için kullanılmak üzere oluşturulmuştur. PASCAL VOC, özellikle derin öğrenme algoritmalarını eğitmek ve değerlendirmek için kullanılan bir referans veri seti olarak önemli bir rol oynamıştır. PASCAL VOC veri seti, 20 farklı nesne sınıfını içerir (X. Zhou vd., 2017). COCO (Common Objects in

Context) veri seti tıpkı PASCAL VOC veri seti gibi, bilgisayarla görü işleme alanında yaygın olarak kullanılan bir veri setidir. Bu veri seti, nesne tanıma, nesne sınıflandırma ve nesne segmentasyonu gibi görevler için kullanılmak üzere Microsoft tarafından oluşturulmuştur. COCO veri seti, daha geniş ve karmaşık sahnelerde bulunan nesneleri içerdiği için diğer veri setlerinden farklıdır. COCO veri seti, özellikle nesne tanıma alanında derin öğrenme modellerini eğitmek ve değerlendirmek için kullanılır. COCO veri setinde 80 farklı nesne sınıfı bulunmaktadır (Lin vd., 2014). Elde edilen sonuçlar, YOLO'nun diğer yöntemlere kıyasla daha yüksek algılama doğruluğuna sahip olduğunu ve gerçek zamanlı uygulamalar için uygun olduğunu göstermiştir. YOLO'nun bir diğer avantajı, tek bir ağ yapısının kullanılması sayesinde eğitim ve tahmin sürecinin basitleştirilmesidir. Bu, YOLO'nun daha kolay kullanılabilir ve uygulanabilir olmasını sağlamaktadır (Redmon vd., 2016; Terven & Cordova-Esparza, 2023).

3.8.7. YOLOv1

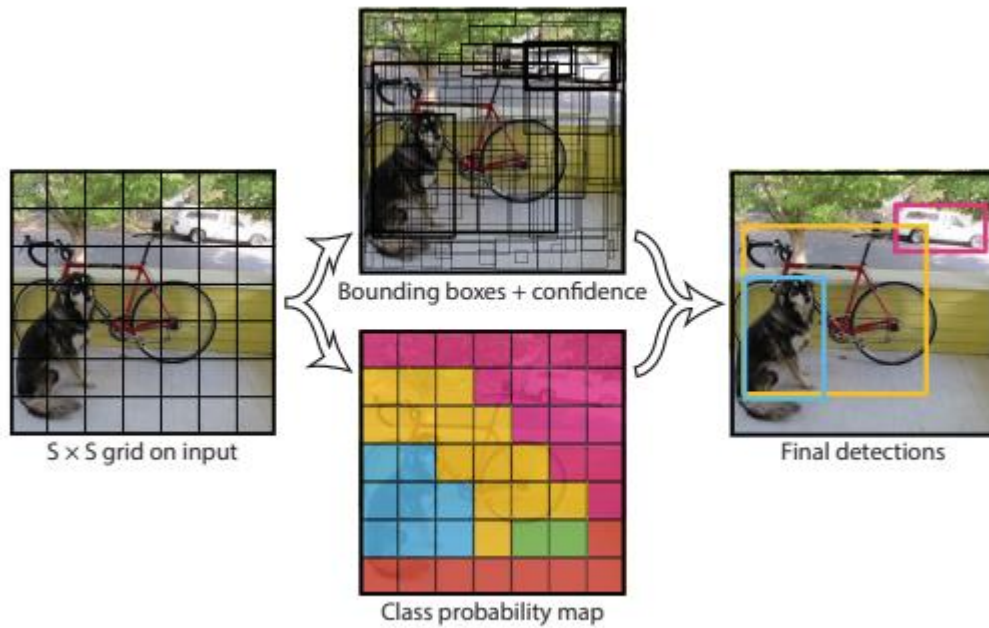
YOLO, önceden eğitilmiş bir derin sinir ağı kullanır. Şekil 2.7'de mimarisi verilen YOLOv1 de VGG-16 adı verilen Oxford Üniversitesi'nde geliştirilen ve 16 katmanlı derin bir sinir ağı kullanılmaktadır. Bu sinir ağı, büyük bir görüntü veri kümesi üzerinde eğitilerek görüntü sınıflandırması görevlerinde yüksek performans göstermiştir (Tammina, 2019). YOLOv1, VGG-16'nın önceden eğitilmiş ağırlıklarını kullanarak nesne tespiti yeteneklerini elde etmek için transfer öğrenme yöntemi olarak adlandırılan yöntemi kullanır.



Şekil 3.7 YOLOv1 Mimarisi (Redmon vd., 2016)

YOLOv1, VGG-16'nın özellik çıkarıcı katmanlarını kullanırken, son katmanları çıkarır ve kendi özel sınıflandırıcı ve lokalizasyon katmanlarını ekler. Bu şekilde, tek bir geçişte nesne tespiti yapabilen bir baştan sona mimari oluşturulur. Bu sinir ağı, genellikle büyük veri kümesiyle eğitilerek nesne tespiti yetenekleri kazanır. Eğitim sürecinde, sinir ağı, binlerce etiketlenmiş görüntüyü kullanarak nesne sınıflandırması ve konumlandırması becerilerini

öğrenmektedir. İkinci adımda, girdi kare, önceden eğitilmiş sinir ağından geçirilir. Bu geçiş sırasında, sinir ağı, farklı ölçeklerdeki özellik haritalarını üretmektedir. Özellik haritaları, nesnelerin farklı özelliklerini (kenarlar, şekiller, renkler vb.) yakalamak için kullanılmaktadır. Özellik haritaları, her bir hücreyi bir nesnenin var olabileceği bir bölge olarak temsil etmektedir. Her hücre, birden fazla sınırlayıcı kutu tahmini yapmaktadır. Her bir sınırlayıcı kutu, nesnenin yaklaşık konumunu ve boyutunu ifade eden dört koordinat değeri (sol üst köşe, sağ alt köşe) ile sınıf olasılıklarını tahmin eden bir dizi sayı üretmektedir. Her sınırlayıcı kutu, sınıf olasılıklarını tahmin etmekte ve genellikle bir softmax aktivasyon fonksiyonu kullanılarak, her sınıf için bir olasılık değeri hesaplanmaktadır. Bu, görüntüdeki nesnenin hangi sınıfa ait olduğunu belirlemek için kullanılır. Sınırlayıcı kutular ve sınıf olasılıkları, eşik değerleriyle filtrelenir. Örneğin, belirli bir olasılık değerinin altındaki tahminler göz ardı edilir. Ardından, nesneleri tespit etmek için en yüksek olasılığa sahip sınırlayıcı kutular seçilir. Bu seçim, çakışan sınırlayıcı kutuları birleştirme ve aynı nesneyi birden fazla kez tespit etmeyi engelleme gibi bazı işleme adımlarını içerebilir (Redmon vd., 2016; Terven & Cordova-Esparza, 2023).



Şekil 3.8 YOLO girdi görüntülerinin ızgaraları bölünmesi (Redmon vd., 2016)

Şekil 2.8’de görüldüğü üzere girdi görüntüsünü bir ızgara şeklinde böler ve sınırlayıcı kutuları ve sınıf olasılıklarını doğrudan ızgara hücrelerinden tahmin eder. Bu benzersiz yaklaşım, YOLO’nun yüksek verimlilikle gerçek zamanlı nesne tespiti yapmakta ve bu alandaki çalışmalarda sıklıkla tercih edilmesine olanak sağlamaktadır. YOLOv1, nesne tespiti için bir baştan sona mimari (girdiden çıktıya) derin öğrenme modeli olarak tasarlanmıştır. Geleneksel nesne tespiti yöntemlerinden farklı olarak, YOLO 7x7 çözünürlüğe sahip bir özellik haritası

oluşturur ve bu harita üzerinden sınırlayıcı kutuları ve sınıf olasılıklarını tahmin eder. Nesne tespiti konusunda başarılı olsa da, düşük boyutlu nesneleri doğru bir şekilde tespit etme zorluğu vardır (Redmon vd., 2016).

3.8.8. YOLOv2

YOLOv2, YOLOv1'in iyileştirilmiş bir sürümüdür YOLO9000 olarak tanıtılmıştır YOLOv2, çeşitli önemli geliştirmeler içermektedir. Öncelikle, YOLOv2, daha küçük nesneleri daha iyi tespit edebilmek için çeşitli ölçekli özellik haritaları kullanmaktadır. Darknet-19 adı verilen özel bir sinir ağı mimarisini kullanılmaktadır. Önceden eğitilmiş bir model kullanarak başlangıç ağırlıkları ile eğitilir. Referans kutusu (Anchor Box) yöntemini kullanarak sınırlayıcı kutuların tahminlerini iyileştirir. Çok ölçekli test tekniği ile farklı çözünürlüklerdeki özellik haritalarını kullanarak nesne tespiti doğruluğunu artırır. Orta boyutlu nesneleri daha iyi tespit edebilir ve genelde YOLOv1'e kıyasla daha iyi performans gösterir. Ayrıca, anlık olay tabanlı sabit nokta temelli bir yaklaşım benimser ve daha fazla evrişimli sinir ağı katmanı eklenerek daha derin bir ağ yapısı kullanır. YOLO'daki tüm evrişim katmanlarına toplu normalizasyon eklenerek, ortalama hassasiyet değerinde %2'den fazla bir iyileşme elde edilmiştir. Toplu normalizasyon ile modele aşırı uyum sağlamadan düşme (dropout) yöntemi çıkartılabilir hale gelmiştir. YOLOv2'nin bu iyileştirmeleri, doğruluğunu önemli ölçüde artırmıştır (Hussain, 2023; Redmon & Farhadi, 2016; Terven & Cordova-Esparza, 2023).

3.8.9. YOLOv3

YOLOv3, Darknet-53 adı verilen daha derin bir sinir ağı mimarisini kullanmaktadır. Residual(kalan) blokları ve atlama bağlantılarını içeren daha karmaşık bir yapıya sahiptir. Feature Pyramid Network (FPN-Özellik Pramidi Ağları) kullanarak farklı ölçeklerdeki özellik haritalarını entegre eder. YOLOv2'ye kıyasla daha fazla sınıflandırma ve konumlandırma özelliği içerir. Kafa bölümünde daha fazla ölçekli sınırlayıcı kutu kullanarak nesne tespit doğruluğunu artırır. YOLOv2'ye göre daha hızlıdır ve daha geniş bir nesne sınıfı yelpazesini destekler. Bu sürümde, daha fazla konvolüsyon tabakası, ölçekli özellik haritaları ve farklı boyutlardaki anlık olaylar (sabitleme kutuları) kullanılır. Ayrıca, farklı ölçeklerdeki nesneleri daha iyi tespit etmek için çoklu ölçekli çıktılar sunmaktadır (Hussain, 2023; Redmon & Farhadi, 2018; Terven & Cordova-Esparza, 2023).

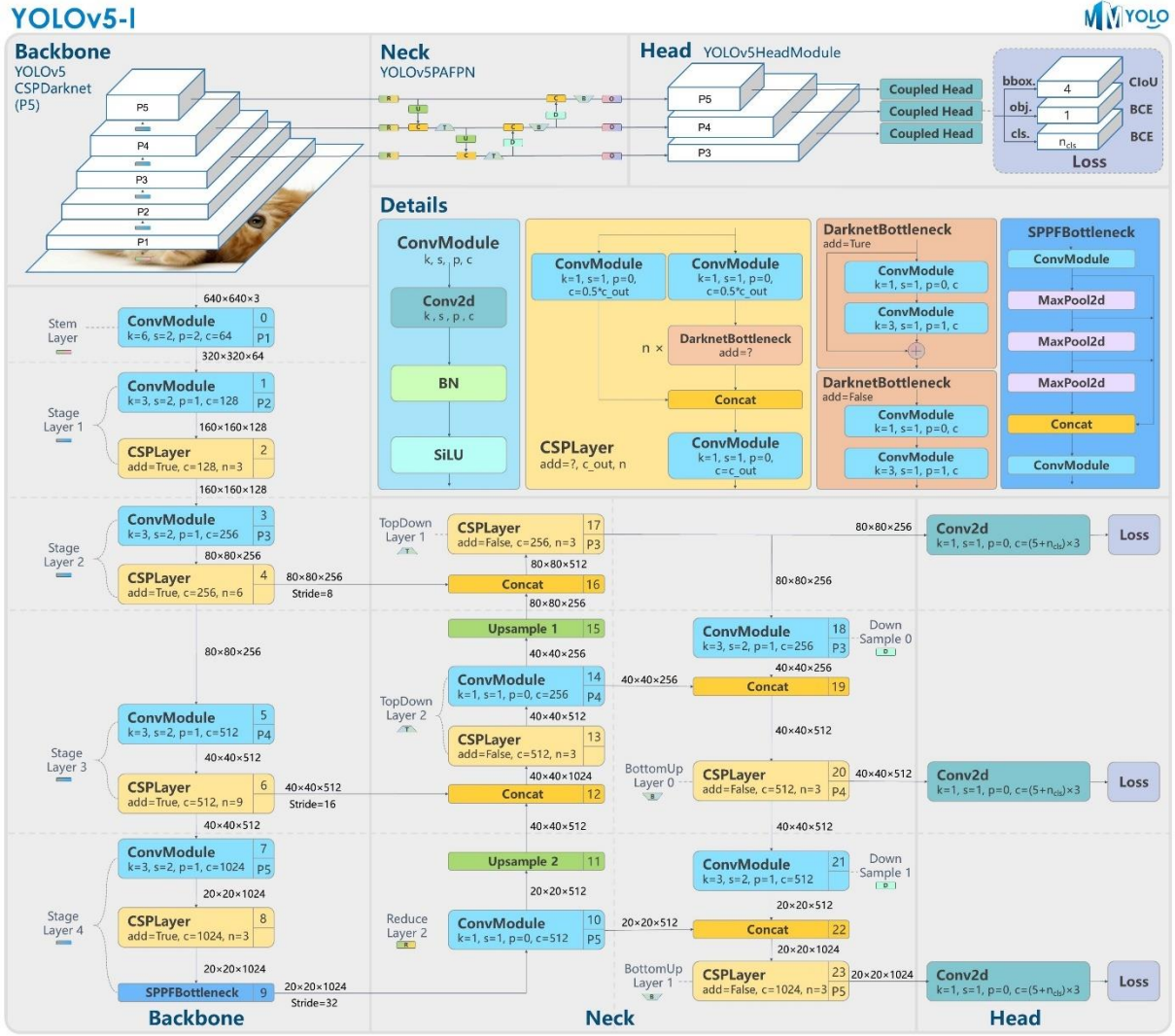
3.8.10. YOLOv4

YOLOv4, CSPDarknet53 adı verilen yeni bir sinir ağı mimarisini kullanır. YOLOv3 sonrası Redmon etik sebeplerden ötürü bilgisayarlı görü projelerini bırakmıştır. Aynı Darknet-

53 mimarisini geliştirilmiş bir versiyonunu kullanan YOLOv4, Alexey Bochkovskiy tarafından geliştirilmiştir. CSPDarknet53, daha derin ve daha karmaşık bir yapıya sahiptir. Mish aktivasyon fonksiyonunu ve yol bağlantılarını içeren yeni özelliklere sahiptir. Bu aktivasyon fonksiyonu geleneksel olarak yaygın olarak kullanılan ReLU aktivasyon fonksiyonunun yerine kullanılır. Mish, daha yumuşak bir gradyan özelliği olan ve sınırlarda daha iyi bir performans gösteren bir fonksiyondur. PANet (Path Aggregation Network) kullanarak farklı ölçeklerdeki özellik haritalarını birleştirir. PANet, ölçekli bir yapıda özellik haritalarını birbirine bağlar ve daha geniş bir konumsal bilgi yelpazesini kapsar. Bu, nesnelerin farklı boyutlarda ve konumlarda doğru bir şekilde tespit edilebilmesini sağlar. Ayrıca, kalıntı blokları ve Çapraz Aşama Kısmi Bağlantılar(CSP) bağlantıları atlama bağlantıları gibi özellikler içerir. YOLOv3'e kıyasla daha yüksek doğruluk ve hız sunar. Optimize edilmiş hız ve bellek kullanımı ile daha gelişmiş bir nesne tespit performansı sağlar. Bu, ağı daha iyi özellik çıkarabilmesini sağlar ve daha yüksek bir temsil yeteneği sunmaktadır (Bochkovskiy vd., 2020; Hussain, 2023; Terven & Cordova-Esparza, 2023).

3.8.11. YOLOv5

Şekil 2.9'da mimari yapısı verilen YOLOv5 ise YOLOv4'e alternatif olarak geliştirilmiştir. YOLOv5, YOLOv4'ten farklı bir mimariye sahiptir ve Ultralytics adlı bir yapay zekâ şirketi tarafından geliştirilmiştir. YOLOv4, CSPDarknet53 adı verilen özel bir sinir ağı mimarisini kullanırken, YOLOv5, özelleştirilmiş bir hafif sinir ağı mimarisine dayanır. Bu mimari, daha hızlı ve daha hafif olmayı hedeflerken, nesne tespiti performansını korumayı amaçlamaktadır. YOLOv5, YOLOv4'ten daha küçük boyutlara sahip bir modeldir. YOLOv5, daha az parametreye ve daha düşük bellek gereksinimlerine sahip olacak şekilde tasarlanmıştır. Bu, YOLOv5'in daha hızlı çalışmasına ve daha kolay dağıtılmasına olanak sağlar. YOLOv5, YOLOv4'ten farklı optimizasyon teknikleri ve algoritmalara sahiptir. Modelin hızını ve performansını artırmak için çeşitli optimizasyon adımları kullanılır. Özellikle, çeşitli büyütme, daraltma ve yol bağlantıları gibi teknikler YOLOv5'te yer alır. YOLOv5, daha hızlı çalışma süreleri ve daha hafif bir yapı sağlamak üzere optimize edilmiştir. Bu sayede gerçek zamanlı uygulamalarda daha yüksek hızlar elde edilebilir. Aynı zamanda, nesne tespiti doğruluğunu korumak için gelişmiş bir yapı sunar (Hussain, 2023; Jiang vd., 2022; Terven & Cordova-Esparza, 2023).



Şekil 3.9 YOLOv5 mimari yapısı (Jocher vd., 2022)

3.8.12. YOLOv6

YOLOv6, 2022 yılında Çin merkezli büyük bir e ticaret şirketi olan Meituan şirketi araştırmacıları tarafından yayınlanan açık kaynak kodlu bir YOLO versiyonudur ve şirketin otonom teslimat robotlarında kullanılmaktadır (Jiang vd., 2022).

3.8.13. YOLOv7

Chien-Yao ve arkadaşları tarafından 2022 yılında yapılan bir çalışmada YOLOv7, 5 FPS ile 160 FPS aralığında hem hız hem de doğruluk açısından bilinen tüm nesne tespit yöntemlerini geride bıraktığı ve GPU(Grafik işlem birimi) V100 üzerinde 30 FPS veya daha yüksek hızlarda bilinen gerçek zamanlı nesne tespit yöntemleri arasında en yüksek doğruluğa (%56.8 AP) sahip olduğunu bulmuştur. Ayrıca, YOLOv7'nin eğitimi sırasında yalnızca MS COCO veri kümesi kullanılmış olup başka veri kümesi veya önceden eğitilmiş ağırlıklar kullanılmamıştır (Chien-Yao vd., 2022; Hussain, 2023; Terven & Cordova-Esparza, 2023).

3.8.14. YOLOv8

Tez çalışmamızda önerilen Şekil 2.10’da mimarisi verilen YOLOv8 modeli Ultralytics şirketi tarafından geliştirilmekte olan Ocak 2023’te yayınlanmış olan en güncel YOLO modelidir. YOLOv5 sürümünü geliştirmiş şirket tarafından önceki modelleri baz alarak geliştirilen son teknoloji ve en gelişmiş model olarak öne çıkmaktadır. Performansın, esnekliğin ve verimliliğin artırılması için önceki modeller üzerine geliştirmeler yapılmıştır. COCO veri seti üzerinde eğitilmiştir. Ultralytics şirketinin paylaştığı verilere göre önceki YOLO versiyonlarından daha başarılı sonuçlar vermiştir. YOLOv8 modelinin nesne tespiti alanında kullanılan 5 farklı modeli vardır. Versiyon farkları başarı ve kaynak kullanımı farklılıkları içerir. Donanım kaynak farklarından ötürü, modelin farklı versiyonları tercih edilir versiyonlar arasında bir dizi iyileştirme, optimize etme veya özellik ekleme farklılıkları vardır. Daha büyük boyutlu modeller daha yüksek başarılar gösterirken gereken kaynak tüketimleri de bunlara doğru orantılı olarak daha fazladır. Gerçek zamanlı nesne tespitinde donanım kaynakları da önemli kısıtlardan birisidir ve optimum çözüm gözetilerek model seçilmelidir (Jocher & Munawar, 2023)(Hussain, 2023; Terven & Cordova-Esparza, 2023).

Tez çalışmasında YOLO sürümlerinden YOLOv8 sürümünün tercih edilmesinin ana sebebi Şekil 2.11’de yer alan başarı tablosudur. Ultralytics şirketinin Github hesabında verdiği başarı tablosuna göre YOLOv8 önceki popüler YOLO versiyonları performans anlamında geçmiştir. Bu sebeple bu çalışmada YOLOv8 modelleri tercih edilmiştir. Çalışmada YOLOv8’in performansı YOLOv5 ile karşılaştırılmış ve Şekil 2.11’de verilen tabloyu destekler niteliktedir.

Tablo 3.1 YOLOv8 nesne tespit modellerinin kıyaslanması (Jocher & Munawar, 2023)

Model	size (pixels)	mAPval 50-95	Speed CPU ONNX (ms)	Speed A100 TensorRT (ms)	params (M)	FLOPs (B)
YOLOv8n	640	37.3	80.4	0.99	3.2	8.7
YOLOv8s	640	44.9	128.4	1.20	11.2	28.6
YOLOv8m	640	50.2	234.7	1.83	25.9	78.9
YOLOv8l	640	52.9	375.2	2.39	43.7	165.2
YOLOv8x	640	53.9	479.1	3.53	68.2	257.8

Tablo 2.1 incelendiğinde YOLOv8, beş farklı ölçeklendirilmiş versiyona sahiptir: YOLOv8n (nano), YOLOv8s (small), YOLOv8m (medium), YOLOv8l (large) ve YOLOv8x (extra large). Bu versiyonlarda, evrişim modüllerinin genişliği ve derinliği, belirli uygulamalara ve donanım gereksinimlerine uygun olarak değişir. Örneğin, YOLOv8n ve YOLOv8s, düşük kaynaklı cihazlar için hafif modellerdir, YOLOv8x ise hızdan ödün vererek yüksek performans için optimize edilmiştir. YOLOv8, nesne tespiti, segmentasyon, poz tahmini, takip ve sınıflandırma gibi birden çok görüntü işleme görevini desteklemektedir.

mAPval değerleri, COCO val2017 veri kümesi üzerinde tek model tek ölçek kullanılarak hesaplanmıştır. Hız, Amazon EC2 P4d sunucusu kullanılarak COCO val görüntüleri üzerinde ortalama olarak hesaplanmıştır. MS COCO veri seti test-dev 2017 üzerinde değerlendirildiğinde, YOLOv8x, 640 piksel görüntü boyutunda (aynı giriş boyutunda YOLOv5’in 50.7%’ine kıyasla) 53.9% AP(ortalama hassasiyet) elde etmiştir ve NVIDIA A100 ve TensorRT üzerinde 280 FPS hızında çalışmıştır (Jocher & Munawar, 2023).

3.9. Değerlendirme Metrikleri

Makine öğrenmesinde ve derin öğrenmede modellerin eğitim sonrası başarısının değerlendirilmesinde, performansının ölçülmesinde ve sonuçların ifade edilebilir şekilde sayısallaştırılmasında metrikler kullanılır. Bu metriklerin geneli gerçek ve tahmin edilen sınıflar arasındaki ilişkiyi gösteren hata matrisi(confusion matrix) isimli bir tablodan türetilmiştir.

YOLO modellerinin değerlendirmesinde de başvurulana ana metrikler precision (hassasiyet), recall (duyarlılık) ve mAP (mean Average Precision-ortalama hassasiyet) değerleridir. Bu değerler hesaplanırken hata matrisindeki değerlere başvurulur. Bunun yanında modelin eğitim başarısında eğitim hızı ve süresi gibi değerlerde dikkate alınabilir.

3.9.1. Hata Matrisi(Confusion Matrix)

Hata matrisi, gerçek ve tahmin edilen sınıf etiketleri arasındaki ilişkiyi gösteren bir tablodur. Pozitif ve negatif sınıflandırmaların doğru ve yanlış tahminlerini gösterir. Bu matris, modelin doğruluğunu, hassasiyetini, duyarlılığını ve F1 skorunu gibi metrikleri hesaplamak için kullanılır. Hata matrisi, modelin hangi sınıfları daha iyi tahmin ettiğini ve hangi sınıflarda hatalar yaptığını anlamak için önemlidir. Bu bilgi, modelin geliştirilmesi ve iyileştirilmesi için kritik ipuçları sağlamaktadır. Hata matrisine ilişkin yapı Tablo 2.2’de gösterilmiştir.

Tablo 3.2 Hata Matrisi

Hata Matrisi		Gerçek	
		Pozitif	Negatif
Tahmin Edilen	Pozitif	Doğru Pozitif	Yanlış Pozitif
	Negatif	Yanlış Negatif	Doğru Negatif

3.9.2. Hassasiyet (Kesinlik-Precision)

Hassasiyet(Precision), doğru pozitif (True Positive) tahminlerin (sınıflandırılan nesnelerin doğru bir şekilde belirlenmesi) toplam tahminlere (Doğru Pozitif + Yanlış Pozitif) oranını temsil eder. Formülü aşağıdaki gibi ifade edilir:

$$Hassasiyet = \frac{Doğru\ Pozitif}{(Doğru\ Pozitif + Yanlış\ Pozitif)}$$

Hassasiyet, bir modelin ne kadar kesin olduğunu gösterir. Yani, modelin doğru tahminlerinin toplam tahminlere oranını verir.

3.9.3. Duyarlılık (Recall)

Duyarlılık(Recall), doğru pozitif tahminlerin gerçek pozitiflerin (sınıflandırılması gereken tüm nesnelerin doğru bir şekilde belirlenmesi) toplamına oranını temsil eder. Formülü şu şekildedir:

$$Duyarlılık = \frac{Doğru\ Pozitif}{(Doğru\ Pozitif + Yanlış\ Negatif)}$$

Duyarlılık, bir modelin ne kadar kapsayıcı olduğunu gösterir. Yani, gerçek pozitiflerin ne kadarını doğru bir şekilde belirleyebildiğini ifade eder.

3.9.4. Ortalama Hassasiyet (Mean Average Precision-mAP)

mAP (mean Average Precision-Ortalama Hassasiyet), bir nesne tespit modelinin performansını genel olarak ölçmek için kullanılan bir metriktir. Ortalama hassasiyet, her bir sınıf için hesaplanan ortalama hassasiyet değerlerinin ortalamasıdır. Ortalama hassasiyet, modelin farklı hassasiyet seviyelerinde nasıl performans gösterdiğini ölçen bir metriktir. Farklı kesinlik seviyeleri için kesinlik-duyarlılık eğrisi çizilir ve bu eğri altındaki alan hesaplanarak ortalama hassasiyet elde edilir. Daha sonra tüm sınıflar için ortalama hassasiyet değerleri alınarak ortalama hassasiyet hesaplanır.

$$\text{Ortalama Hassasiyet}(mAP) = \frac{1}{N} \sum_{i=1}^N AP_i$$

3.9.5. Değerlerin İncelenmesi

Yüksek bir kesinlik değeri, modelin sınıflandırmalarının çoğunun doğru olduğunu gösterir. Duyarlılık değerinin yüksek olması, modelin sınıflandırması gereken nesnelerin büyük bir kısmını doğru bir şekilde tespit ettiğini gösterir. Yüksek bir ortalama hassasiyet değeri, modelin genel olarak iyi bir nesne tespit performansı sergilediğini gösterir.

3.9.6. Kayıp Fonksiyonları

Bütün bu metriklerin yanında modelin eğitimi devam ederken öğrenme ve doğrulama sırasında kullanılan box loss, cls loss ve dfl loss değerleri vardır. Bunlar modelin eğitim performansının iyileştirilmesi için kullanılan kayıp fonksiyonlarıdır.

3.9.6.1. Kutu Kaybı(Box Loss) Kayıp Fonksiyonu

Box loss nesne tespiti algoritmalarında kullanılan bir kayıp fonksiyonudur. Genellikle YOLO gibi tek aşamalı nesne algılama yöntemlerinde kullanılır. YOLO, görüntüyü tek bir geçişte tarar ve nesneleri tespit ederken, aynı zamanda bu nesnelerin sınıfını ve sınırlayıcı kutularını (bounding box) tahmin eder. Kutu kaybı, tahmin edilen sınırlayıcı kutuların gerçek sınırlayıcı kutulara olan benzerliğini değerlendirerek bir kayıp değeri hesaplar. Yani, tahmin edilen sınırlayıcı kutuların gerçek sınırlayıcı kutularla ne kadar iyi örtüştüğünü ölçer. Kutu kaybı genellikle iki bileşenden oluşur. Bunlar kutu konum kaybı (box localization loss) ve kutu boyutu kaybı (box size loss). Kutu konum kaybı (box localization loss) tahmin edilen sınırlayıcı kutuların gerçek sınırlayıcı kutulara olan konum farkını hesaplar. Örneğin, tahmin edilen

kutunun merkez noktasıyla gerçek kutunun merkez noktası arasındaki mesafeyi ve tahmin edilen kutunun genişliği ve yüksekliği ile gerçek kutunun genişliği ve yüksekliği arasındaki farkı ölçer. Bu, kutunun konumunun doğru tahmin edilip edilmediğini değerlendirir. Kutu boyutu kaybı (box size loss) tahmin edilen sınırlayıcı kutuların gerçek sınırlayıcı kutulara olan boyut farkını hesaplar. Örneğin, tahmin edilen kutunun genişlik ve yükseklik değerlerinin, gerçek kutunun genişlik ve yükseklik değerleriyle ne kadar uyumlu olduğunu ölçer. Bu, kutunun boyutunun doğru tahmin edilip edilmediğini değerlendirir. Kutu kaybı, bu iki bileşenin birleşimi olarak hesaplanır ve genellikle toplam nesne algılama kaybının bir parçasıdır. Box loss, nesne tespiti algoritmasının eğitim sürecinde kullanılır ve gerçek sınırlayıcı kutulara daha iyi uyum sağlamak için modelin sınırlayıcı kutu tahminlerini iyileştirmeye yönelik bir geriye yayılım optimizasyonu yapılmasını sağlar (C. Li vd., 2022).

3.9.6.2. Sınıflandırma Kaybı(Classification Loss -Cls loss) Kayıp Fonksiyonu

Sınıflandırma kaybı nesne tespiti algoritmasının sınıf tahminlerini değerlendirir. Model, görüntüde bulunan her bir nesne için sınıf etiketi tahmin eder. Sınıf kaybı, bu tahminlerin gerçek sınıf etiketleriyle ne kadar uyumlu olduğunu ölçer. Sınıf kaybı genellikle çapraz entropi (cross-entropy) kaybı olarak hesaplanır. Çapraz entropi kaybı, tahmin edilen sınıf etiketlerinin gerçek sınıf etiketlerine ne kadar yakın olduğunu ölçer. Tahmin edilen sınıflar genellikle softmax aktivasyon fonksiyonu kullanılarak olasılık dağılımı olarak hesaplanır. Sınıf kaybı, tahmin edilen olasılık dağılımının gerçek sınıf etiketlerini ne kadar iyi temsil ettiğini değerlendirir. Sınıflandırma kaybı, nesne tespiti algoritmasının sınıflandırma yeteneğini optimize etmeye yardımcı olur. Modelin sınıfları doğru bir şekilde tahmin etmesini sağlayarak, nesne tespiti performansını artırır. Kayıp değeri, gerçek sınıf etiketiyle tahmin edilen sınıf etiketi arasındaki farkın büyüklüğüne bağlı olarak artar veya azalır. Eğer tahmin edilen sınıf etiketi gerçek sınıf etiketiyle tamamen örtüşüyorsa, sınıf kaybı düşük olur. Ancak tahmin edilen sınıf etiketi gerçek sınıf etiketinden uzaklaştıkça, sınıf kaybı değeri yükselir. Sınıflandırma kaybı değeri, modelin sınıflandırma yeteneğini ölçmektedir(C. Li vd., 2022).

3.9.6.3. Distribution Focal Loss(Dfl loss) Kayıp Fonksiyonu

Dfl kayıp fonksiyonu, nesne tespiti algoritmasının sınırlayıcı kutu tahminlerini değerlendirir. Model, görüntüdeki nesneleri tespit etmek için sınırlayıcı kutular tahmin eder. Dfl kayıp fonksiyonu, bu tahmin edilen kutuların gerçek sınırlayıcı kutularla ne kadar iyi örtüştüğünü ölçer. Dfl kayıp fonksiyonu, kutuların tahmin edilen konumu ve gerçek konumu arasındaki farkı hesaplar. Ayrıca, kutuların sınırlayıcı kutu olup olmadığını da değerlendirir. Böylece, modelin sınırlayıcı kutu tahminlerini iyileştirmeye odaklanır. Dfl kayıp fonksiyonu,

farklı bir ağırlıklandırma mekanizması kullanarak zorlu örnekleri vurgular. Zorlu veya nadir sınıflara ait nesnelerin daha iyi tahmin edilmesini sağlar. Bu sayede, modelin daha iyi sınırlayıcı kutu tahminleri yapması ve nesne tespiti performansının artması hedeflenir. Dfl kayıp fonksiyonunun yüksek olması genellikle iki durumu ifade eder. Bunlar zayıf öğrenme ve zorluk örnekler sebebiyledir. Zayıf öğrenme gözlenen durumlarda yüksek bir dfl kayıp fonksiyonu değeri, modelin nesne tespiti görevini henüz tam olarak öğrenemediğini veya doğru tahminler yapmak için yeterli öğrenme sürecinden geçmediğini gösterebilir. Bu durumda, modelin sınırlayıcı kutu tahminleri ve sınıf tahminleri hatalı veya düşük doğrulukta olabilir. Zorlu örnekler olduğu durumda da yüksek dfl kayıp fonksiyonu değerleri görülür. Bu modelin zorlu örneklerle başa çıkma konusunda zorlandığını gösterebilir. Zorlu örnekler, nadir sınıflara ait veya zor ayırt edilebilir özelliklere sahip nesneleri temsil edebilir. Model, bu zorlu örnekleri doğru bir şekilde sınıflandırmak veya sınırlayıcı kutularını doğru bir şekilde tahmin etmek için daha fazla eğitim veya odaklanma gerektirebilir (C. Li vd., 2022; Lin vd., 2017).

ÜÇÜNCÜ BÖLÜM

BULGULAR

4.1. Eğitim Ortamı

Balon balığı tespiti için yürütülen çalışmada 2473 görsel içeren veri seti Google Colab ortamlarında eğitilmiştir. Colab ortamında eğitilen modelin eğitiminde Ultralytics YOLOv8.0.134 sürümü ve YOLOv5 v7.0-72 kullanılmıştır. Python-3.10.12 versiyonu ve torch-2.0.1+cu118 versiyonu kullanılmıştır.

Google Colab üzerinden üyelik satın alınmış olup çalışma sırasında A100 ve V100 grafik işlem birimleri kullanılmıştır. Google Colab üzerindeki çalışma zamanında kullanılan sistem özellikleri 12 çekirdek işlemci, 83,5 GB Ram bellek, 166.8 GB disk alanı ve 40GB Nvidia A100 SXM grafik işlem birimi veya 16GB Nvidia V100 grafik işlem birimidir. Kullanılan grafik işlem birimi sürücü versiyonu 525.105.17'dir. Kullanılan CUDA versiyonu 12.0'dır.

4.2. Eğitimin Sonuçları

Çalışmada her model için epok değeri 300 ve patience (sabır) değeri 100 olarak belirlenmiştir. Sabır değeri üst üste gerçekleşen epokları takip eder ve verdiğimiz değer kadar epokta eğitimde daha iyi değerler yakalayamazsa bütün epokların bitmesini beklemeden eğitimi sonlandırır. Modelin artık öğrenemediği durumlarda daha fazla kaynak harcamamasını istediğimiz için bu değeri 100 olarak belirledik. Bu sabır değeri epok sayısı kadar verildiğinde model eğitimine bütün epoklar sonlanana kadar devam edecektir yine sabır değerini 0 olarak verdiğimizde eğitim bütün epoklar sonlanana kadar devam edecektir. Eğitim sonucu elde edilen kesinlik, duyarlık ve ortalama hassasiyet değerleri Tablo 3.1'deki gibidir. Çalışmada YOLOv8 modellerinin üzerinde durulmaktadır. YOLOv5 modelleri YOLOv8 modellerinin eğitim başarılarını desteklemek için çalışmaya eklenmiştir.

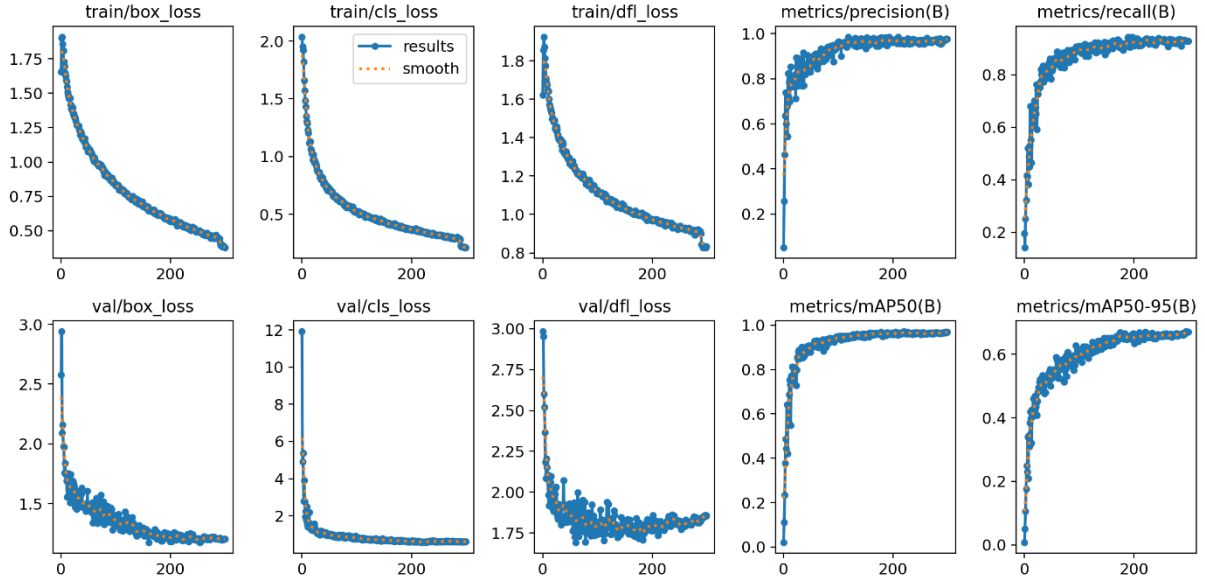
Tablo 4.1 Modellerin eğitimi sonrası doğrulama veri setine ait eğitim sonuçları

Model	Hassasiyet	Duyarlılık	Ortalama Hassasiyet	Grafik İşlem Birimi	Süre
YOLOv5n	%97,20	%88,70	%92,40	V100	1 saat 6dk 53s
YOLOv5m	%95,10	%89,70	%93,40	V100	2 saat 45dk 50s
YOLOv8n	%93,90	%93,90	%95,80	V100	1 saat 6dk 18s
YOLOv8m	%97,60	%92,70	%96,90	V100	1 saat 55dk 2s
YOLOv8l	%97,40	%92,30	%96,70	A100	1 saat 47dk 13s

Tablo 3.1'deki hassasiyet, duyarlılık, ortalama hassasiyet değerleri Google Colab ortamında gerçekleştirilen eğitim sonrası elde edilen değerlerdir. Tablodaki değerler doğrulama veri setine ait hata matrisleri üzerinde hassasiyet ve duyarlılık formülleri baz alınarak eğitim sonu otomatik oluşturulmuştur. Ortalama hassasiyet değerinin elde edilmesinde hassasiyet-duyarlılık eğrisinin altında kalan alan kullanılmıştır olup yine bu değer eğitim sonrası otomatik hesaplanmıştır. Google Colab ortamında görüntü işleme birimi için A100 grafik işlem birimi seçilmiş olup Colab bu grafik işlem birimi yerine V100 grafik işlem birimini kullanıma sunmuştur. Colab kaynaklarına erişim taleplerinde Colab istenilen grafik işlem birimi yerine alternatiflerinden birisini sunabilmektedir. Grafik işlem birimlerinde yaşanan farklılık buna dayanmaktadır. Eğitim sürecinde farklı grafik işlem birimlerinin kullanılması eğitim süresine etki etmektedir.

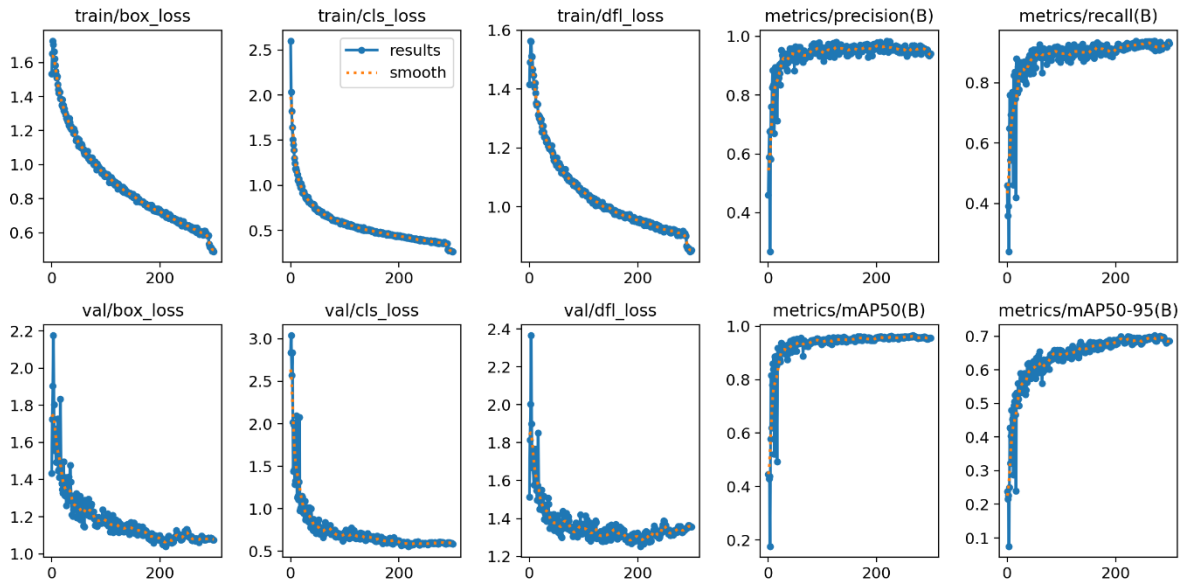
Tablo 3.1'de yer alan YOLOv5n ait ortalama hassasiyet değeri ve eğitim süresi YOLOv8n ile karşılaştırıldığında YOLOv8n modeliyle daha iyi sonuçlar elde edildiği gözlenmiştir.

Tablo 3.1'de yer alan YOLOv5m ait ortalama hassasiyet değeri ve eğitim süresi YOLOv8m ile karşılaştırıldığında YOLOv8m modeliyle daha iyi sonuçlar elde edildiği gözlenmiştir. Aynı grafik işlem birimlerinin kullanıldığı iki eğitim arasında eğitimlerin tamamlanma süresi açısından yaklaşık 51 dakikalık fark gözlenmiştir.



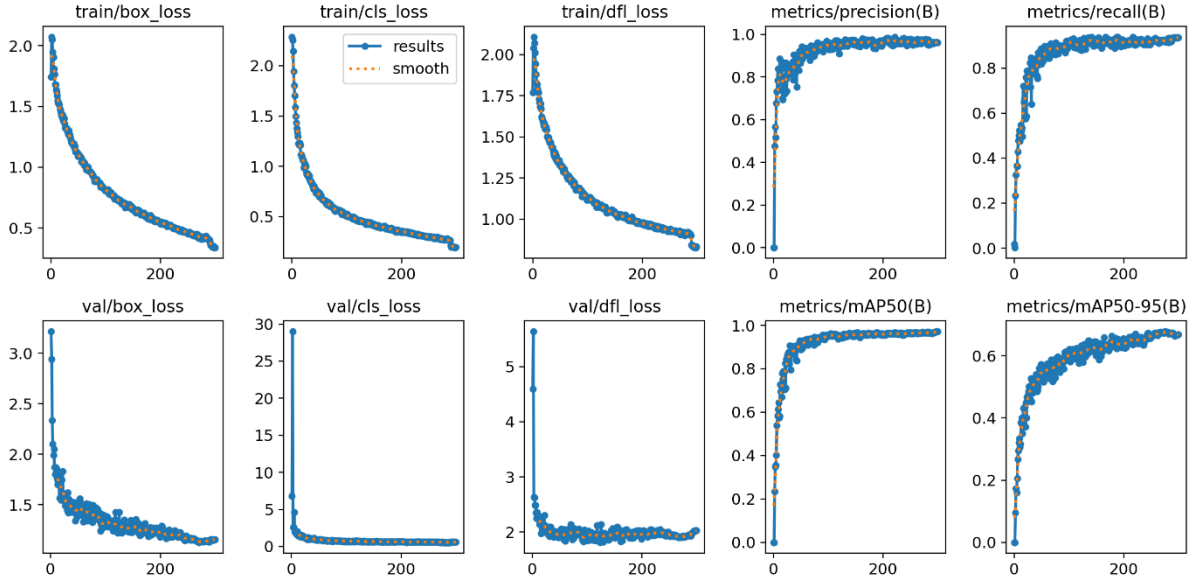
Şekil 4.1 YOLOv8m eğitim sonuçlarına ait grafikler

Şekil 3.1’de yer alan YOLOv8m modeline ait eğitim sonuçlarına dair grafiklerde 200 epok sonrasında epoklar incelendiğinde eğitim sırasında modelin duyarlılık, hassasiyet ve ortalama hassasiyet değerlerinin yatayda sabitleştiği görülmektedir. 250-300 arası ortalama hassasiyette modelin stabil bir performans sergilediği gözlenmiştir. Doğrulama(val) grafiklerinde 250epok sonrasında dfl kaybı değerinde artış gözlenmiştir. Bu YOLOv8m kullanılan örneklerde 300 epok Google Colab çalışma ortamında 1 saat 55dk gibi bir sürede tamamlanırken diğer modellere ait ortalama hassasiyet ve tamamlanma süreleri Tablo 3.1’de yer almaktadır. Elde edilmiş en yüksek ortalama hassasiyet değeri YOLOv8m modeli ile gözlenmiştir.



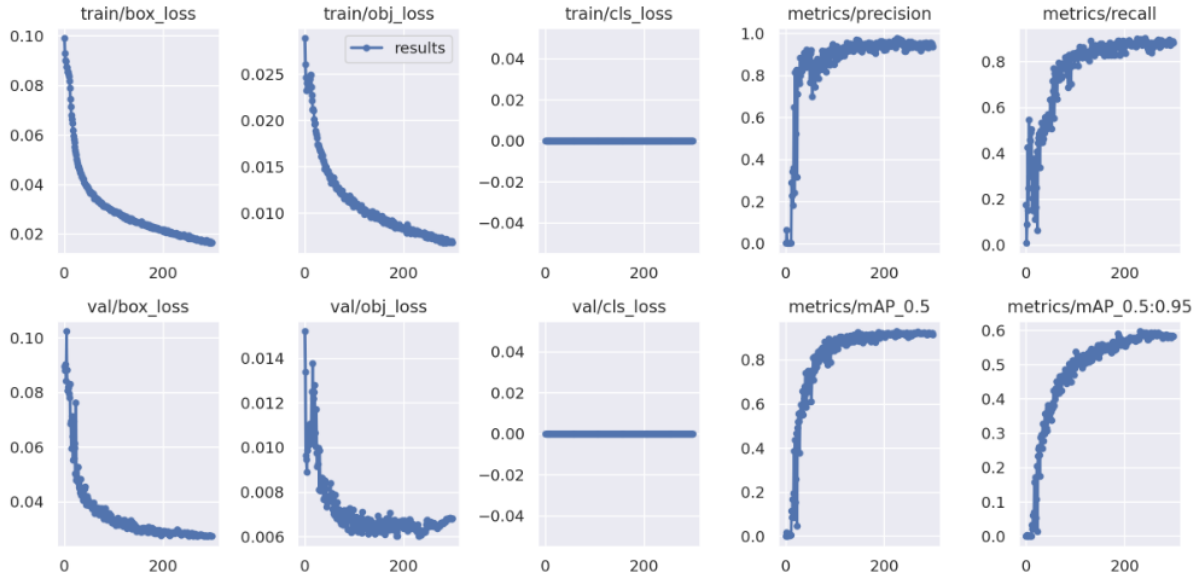
Şekil 4.2 YOLOv8n modeli eğitim sonuçlarına ait grafikler

Şekil 3.2’de YOLOv8n modeline ait eğitim sonuçlarını içeren grafikler yer almaktadır. Grafikler 300 epok olarak gerçekleştirilen eğitim sürecinde yer alan değerlerin değişimini göstermektedir. 200-300 epok arası doğrulama kutu kaybı ve dfl kaybında değerlerin yukarı yönlü arttığı gözlenmiştir.



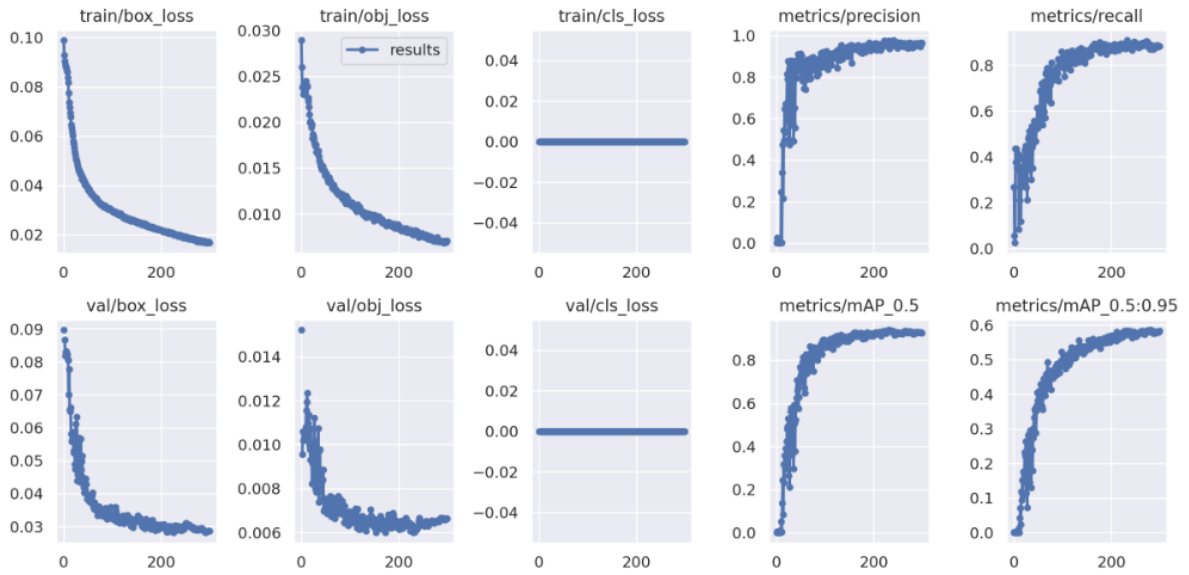
Şekil 4.3 YOLOv8l modelinin eğitim sonuçlarına ilişkin grafikler

Şekil 3.3’te YOLOv8l modeline ait eğitim sonuçlarını içeren grafikler yer almaktadır. Tablo 2.1’de yer alan YOLOv8’e ait nesne tespiti ölçeklendirilmiş paketlerine bakıldığında eğitimde kullanılan diğer YOLOv8 modellerine göre en fazla parametre içeren ve aynı koşullarda diğer eğitimde kullanılan YOLOv8 modellerine göre en yavaş çalışan en fazla grafik işlem birimine ihtiyaç duyan bu modeldir. Eğitim sonuçları incelendiğinde doğrulama(val) kutu kaybı ve dfl kaybı değerlerinin YOLOv8n ve YOLOv8m değerlerine göre yüksek kaldığı gözlenmiştir. 250 epok sonrası dfl değerinin yukarı yönlü hareket edildiği de gözlenmektedir.



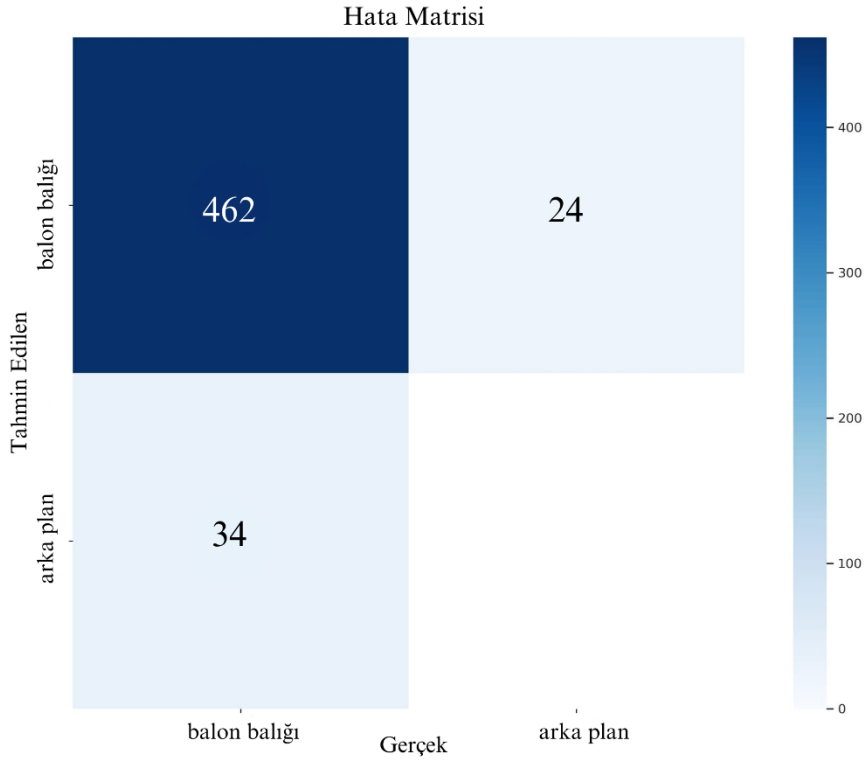
Şekil 4.4 YOLOv5n modelinin eğitim sonuçlarına ilişkin grafikler

Şekil 3.4'te YOLOv5n modeline ait eğitim sonuçlarını içeren grafikler yer almaktadır. YOLOv5 modelleri ile YOLOv8 modelleri arasında yer alan farklılıklardan ötürü YOLOv5 tek sınıflı örneklerde sınıflandırma kaybını 0 olarak hesaplamaktadır. 200 epok sonrası doğrulama(val) objektif kaybı değerlerinin yükseldiği gözlenmiştir.



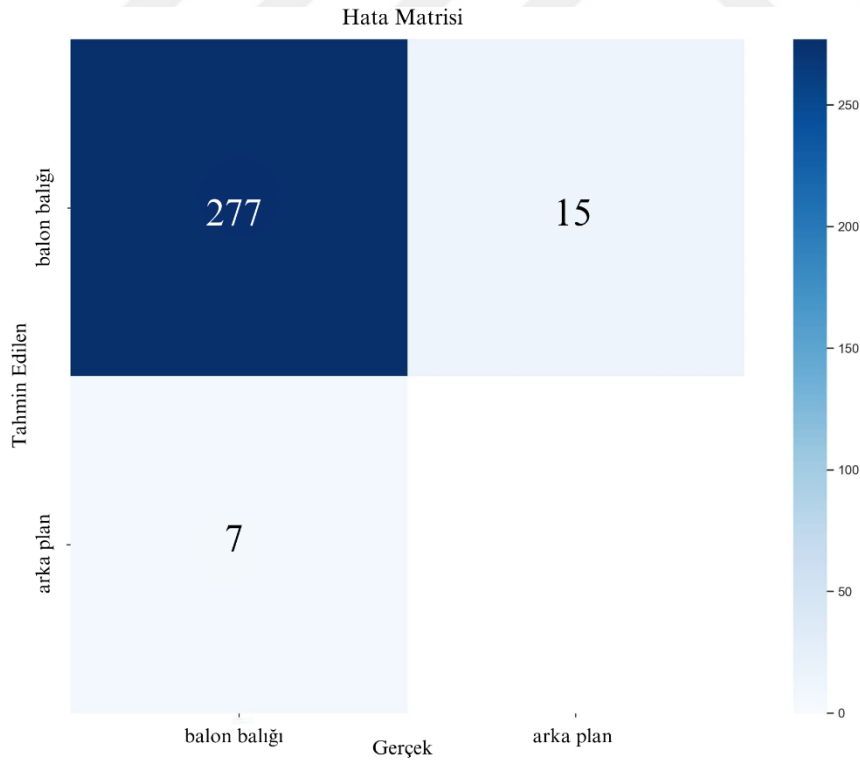
Şekil 4.5 YOLOv5m modeli eğitim sonuçlarına ait grafikler

Şekil 3.5'te YOLOv5m modeline ait eğitim sonuçlarını içeren grafikler yer almaktadır. Doğrulama(val) objektif kaybı değerinin 200 epok sonrası yükselme eğilimi gösterdiği gözlenmiştir.



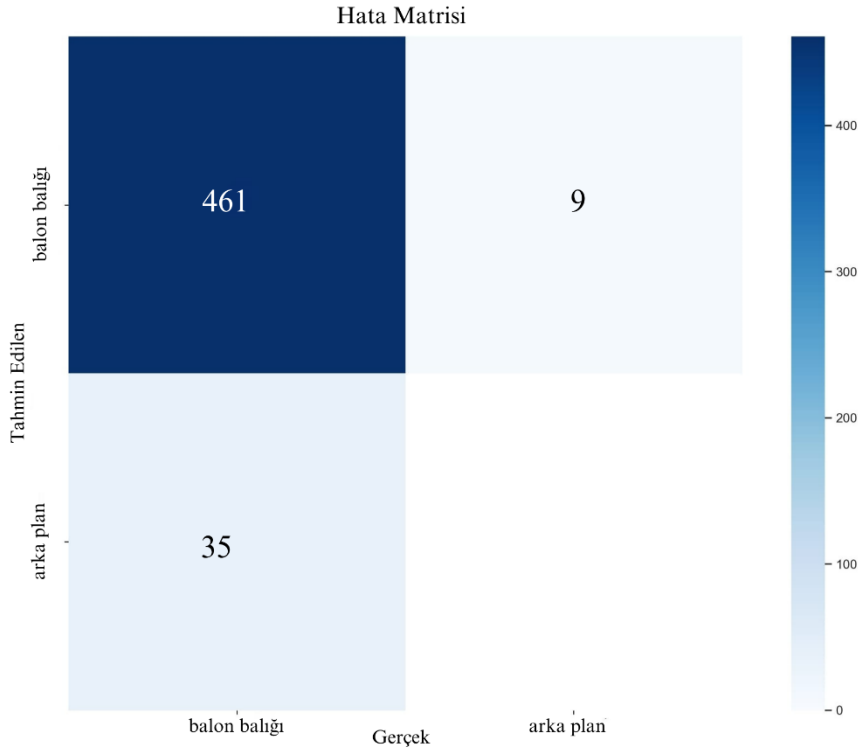
Şekil 4.6 YOLOv8n doğrulama veri seti hata matrisi

Şekil 3.6'da yer alan hata matrisine göre YOLOv8n modeli doğrulama veri setinde balon balıklarını 462 doğru pozitif, 24 yanlış pozitif ve 34 yanlış negatif olarak tahmin etmiştir.



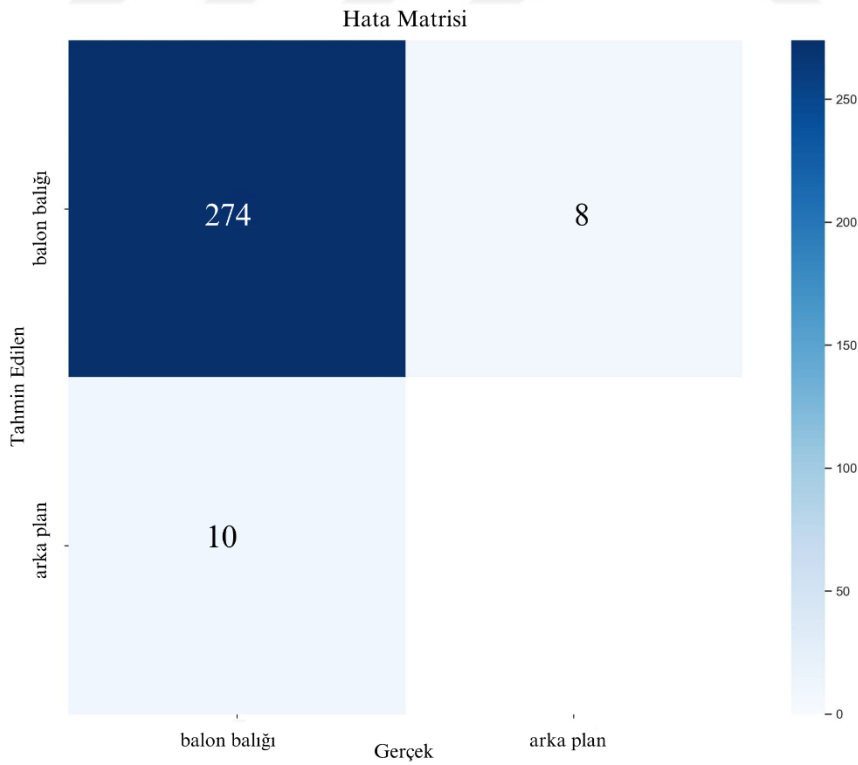
Şekil 4.7 YOLOv8n test veri seti hata matrisi

Şekil 3.7'de yer alan hata matrisine göre YOLOv8n modeli test veri setinde balon balıklarının 277 doğru pozitif, 15 yanlış pozitif ve 7 yanlış negatif olarak tahmin etmiştir.



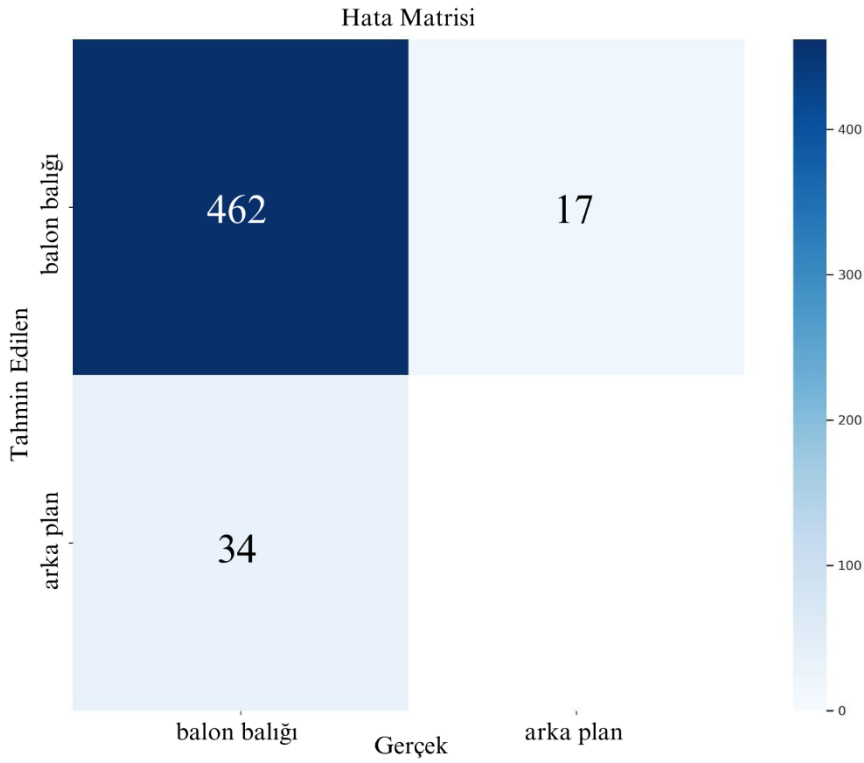
Şekil 4.8 YOLOv8m doğrulama veri seti hata matrisi

Şekil 3.8’de yer alan YOLOv8m hata matrisine göre YOLOv8m modeli doğrulama veri setinde balon balıklarının 461 doğru pozitif, 9 yanlış pozitif ve 35 yanlış negatif olarak tahmin etmiştir.



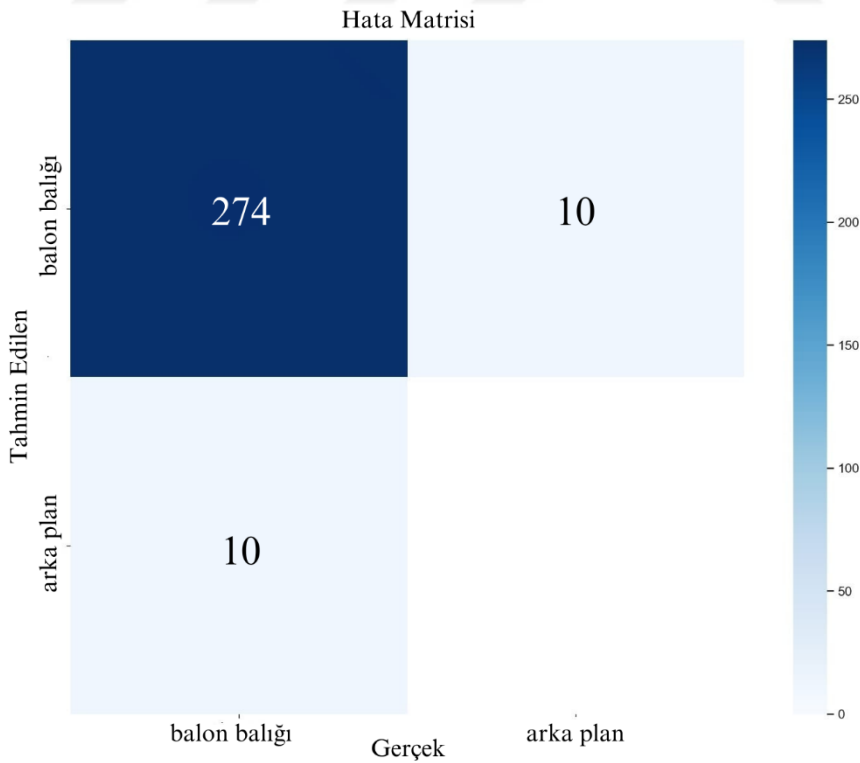
Şekil 4.9 YOLOv8m test veri seti hata matrisi

Şekil 3.9’da yer alan hata matrisine göre YOLOv8m modeli test veri setinde balon balıklarının 274 doğru pozitif, 8 yanlış pozitif ve 10 yanlış negatif olarak tahmin etmiştir.



Şekil 4.10 YOLOv8l doğrulama veri seti hata matrisi

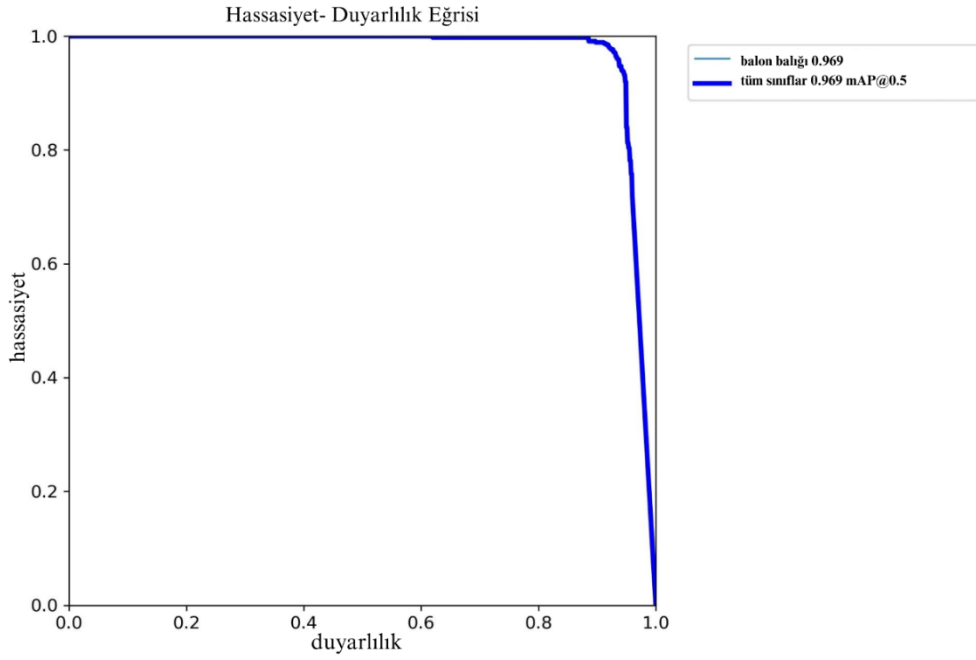
Şekil 3.10’da yer alan hata matrisine göre YOLOv8l modeli doğrulama veri setinde balon balıklarının 462 doğru pozitif, 17 yanlış pozitif ve 34 yanlış negatif olarak tahmin etmiştir.



Şekil 4.11 YOLOv8l test veri seti hata matrisi

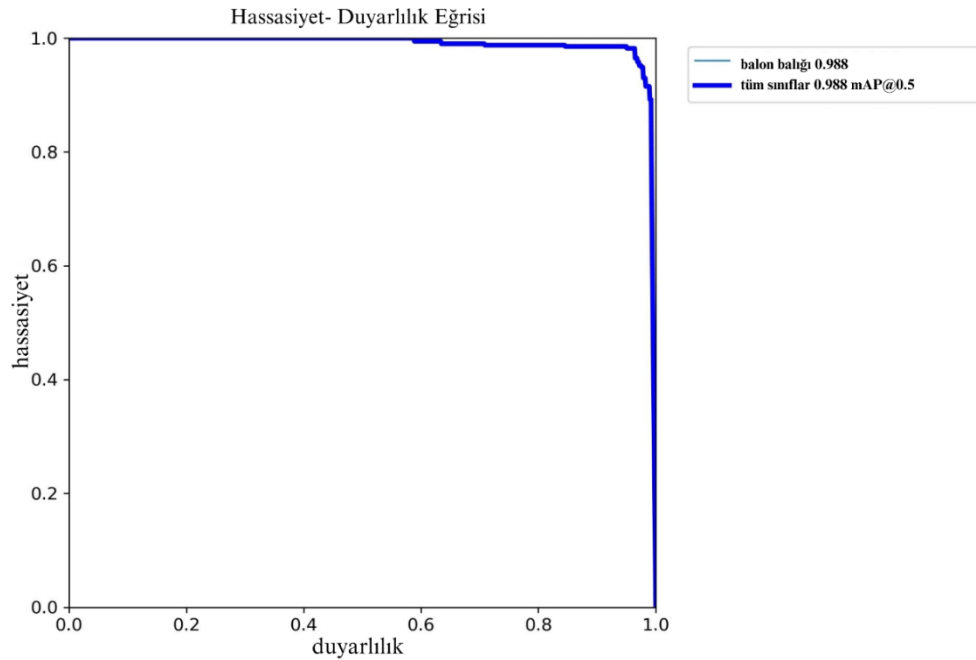
Şekil 3.11’de yer alan hata matrisine göre YOLOv8l modeli test veri setinde balon balıklarının 274 doğru pozitif, 10 yanlış pozitif ve 10 yanlış negatif olarak tahmin etmiştir.

YOLOv8 modellerinde tek sınıf içeren çalışmalarda YOLO tarafından üretilen hata matrislerinde doğru negatif değeri hata matrisinde gösterilmemektedir.



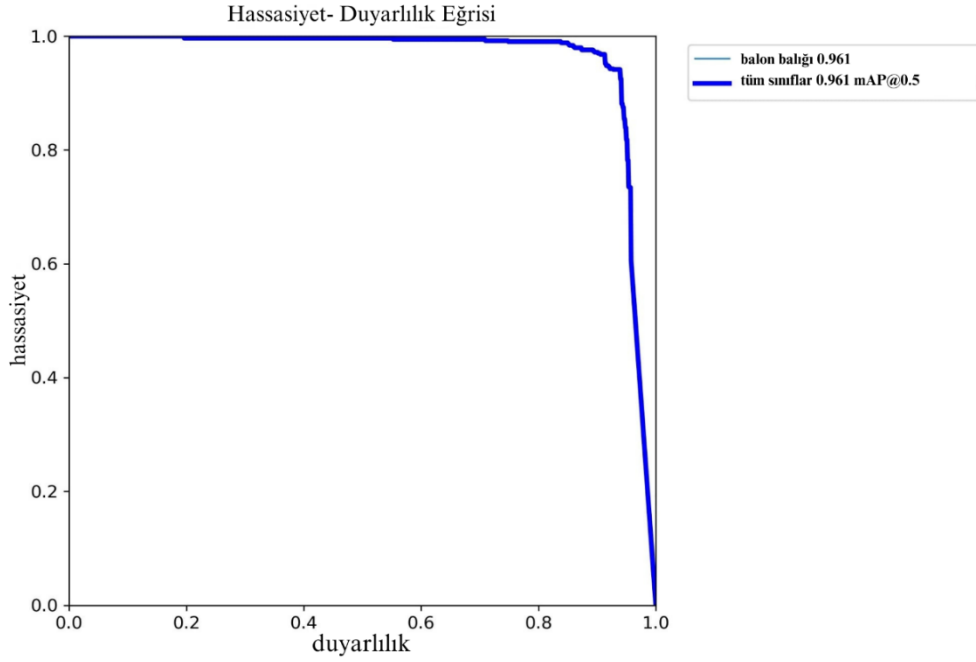
Şekil 4.12 YOLOv8m doğrulama hassasiyet duyarlılık eğrisi

Şekil 3.12’de YOLOv8m doğrulama veri setinde elde edilen ortalama hassasiyeti gösteren hassasiyet duyarlılık eğrisi yer almaktadır. YOLOv8m modelinde balon balığı için ortalama hassasiyet 0,50 eşik değerinde %96,90 bulunmuştur.



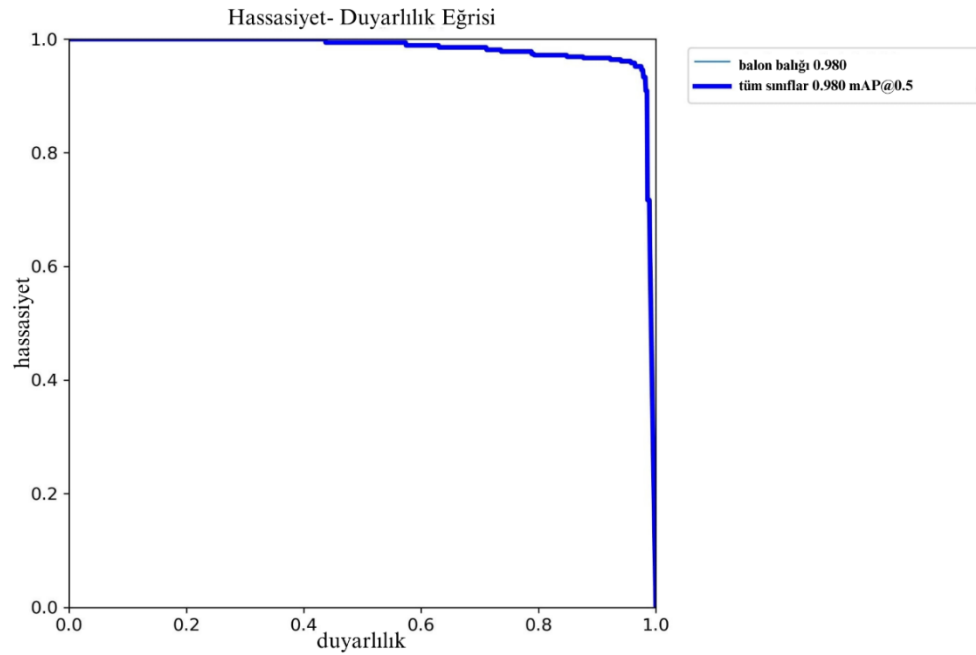
Şekil 4.13 YOLOv8m test hassasiyet duyarlılık eğrisi

Şekil 3.13'te YOLOv8m test veri setinde elde edilen ortalama hassasiyeti gösteren hassasiyet duyarlılık eğrisi yer almaktadır. YOLOv8m modelinde balon balığı için ortalama hassasiyet 0,50 eşik değerinde %98,80 olarak bulunmuştur.



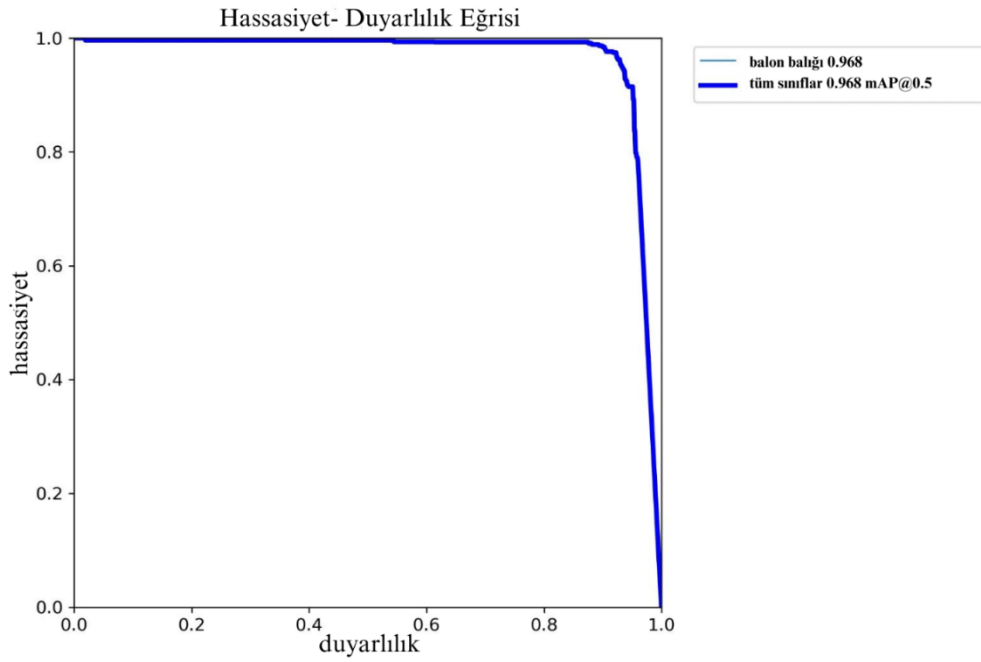
Şekil 4.14 YOLOv8n doğrulama hassasiyet duyarlılık eğrisi

Şekil 3.14'te YOLOv8n doğrulama veri setinde elde edilen ortalama hassasiyeti gösteren hassasiyet duyarlılık eğrisi yer almaktadır. YOLOv8n modelinde balon balığı için ortalama hassasiyet 0,50 eşik değerinde %96,10 olarak bulunmuştur.



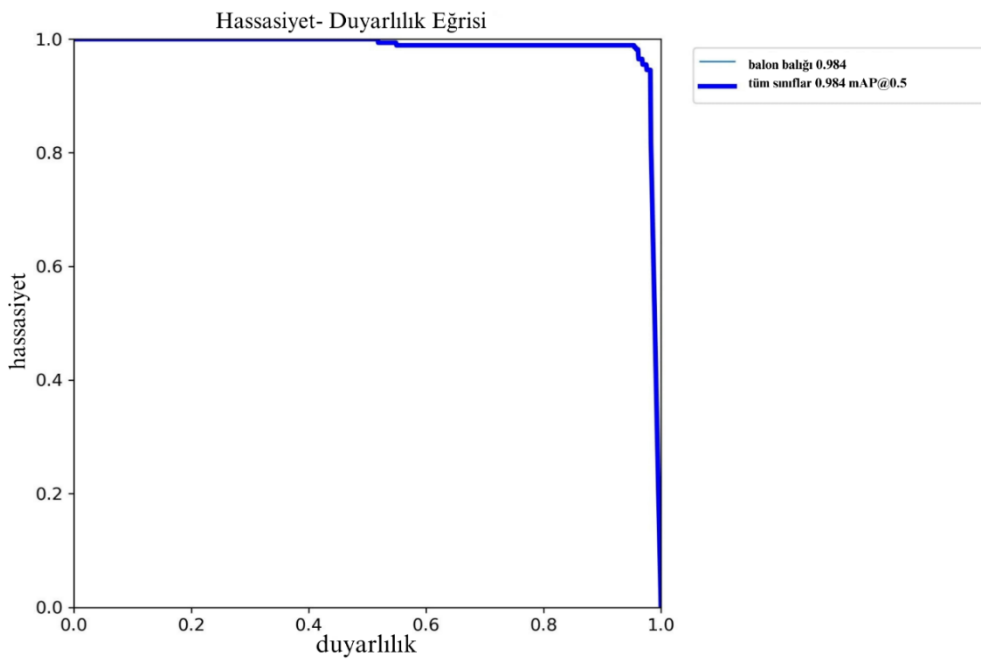
Şekil 4.15 YOLOv8n test hassasiyet duyarlılık eğrisi

Şekil 3.15'te YOLOv8n test veri setinde elde edilen ortalama hassasiyeti gösteren hassasiyet duyarlılık eğrisi yer almaktadır. YOLOv8n modelinde balon balığı için ortalama hassasiyet 0,50 eşik değerinde %98 olarak bulunmuştur.



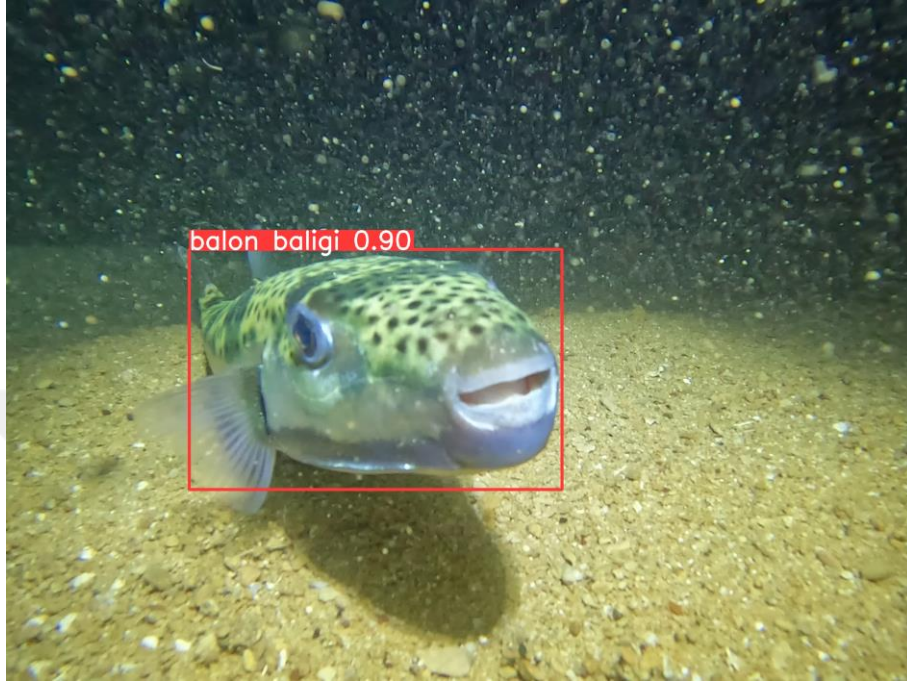
Şekil 4.16 YOLOv8l doğrulama hassasiyet duyarlılık eğrisi

Şekil 3.16'da YOLOv8l doğrulama veri setinde elde edilen ortalama hassasiyeti gösteren hassasiyet duyarlılık eğrisi yer almaktadır. YOLOv8l modelinde balon balığı için ortalama hassasiyet 0,50 eşik değerinde %96,80 olarak bulunmuştur.



Şekil 4.17 YOLOv8l test hassasiyet duyarlılık eğrisi

Şekil 3.17’de YOLOv8l test veri setinde elde edilen ortalama hassasiyeti gösteren hassasiyet duyarlılık eğrisi yer almaktadır. YOLOv8l modelinde balon balığı için ortalama hassasiyet 0,50 eşik değerinde %98,40 olarak bulunmuştur.



Şekil 4.18 Eğitilmiş model ağırlığı ile tespit edilen balon balığı görüntüsü

Şekil 3.18’deki görsel YOLOv8m modelinin başarısı, eğitim sonrası elde edilmiş en iyi ağırlıklar ile gürültülü ortam, gece ve yapay ışık altında balığın önden çekilmiş fotoğrafı üzerinde tahmin edilmiştir.



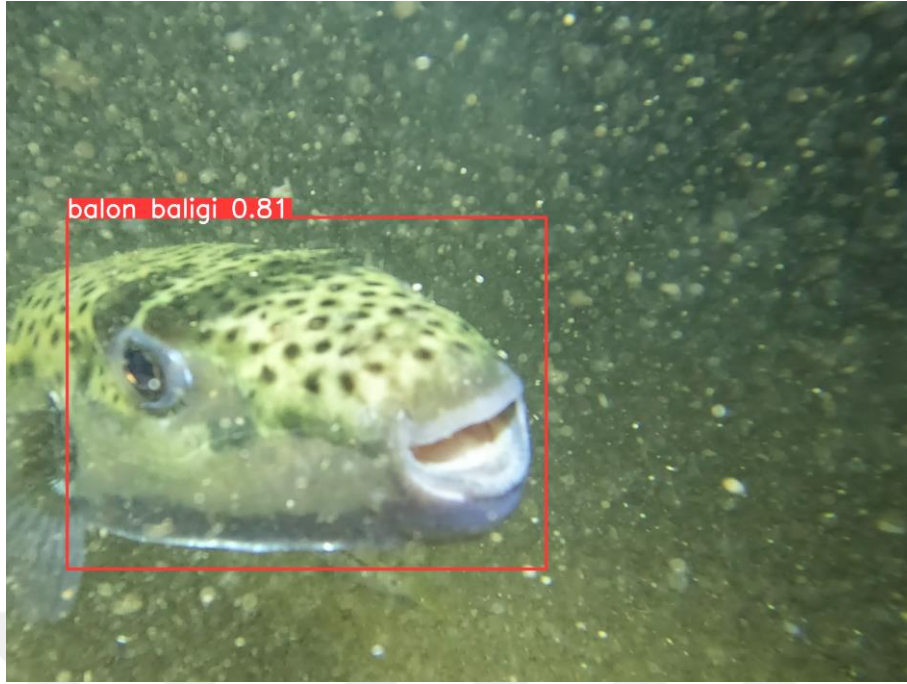
Şekil 4.19 Gece çekimi, az ışık olan ortamda YOLOv8m model ağırlığı ile tespit edilen balon balığı

Şekil 3.19’da gece gerçekleşen çekimde loş ışık altında önden çekilen balon balığı fotoğrafı üzerinde gerçekleştirilmiş tahmin yer almaktadır.



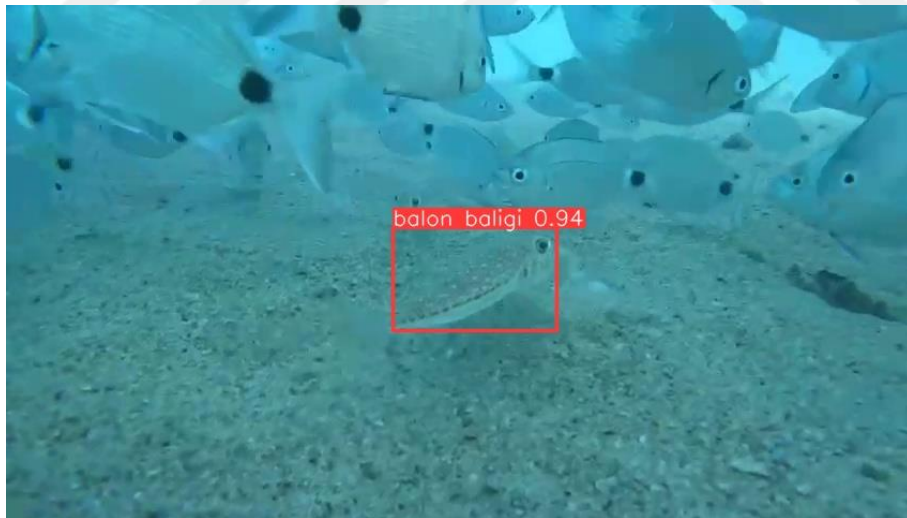
Şekil 4.20 Gece çekimi, çok ışık olan ortamda YOLOv8m model ağırlığı ile tespit edilen balon balığı

Şekil 3.20’de yer alan görsel YOLOv8m modelinin eğitilmesi sonucu elde edilmiş en iyi ağırlık ile tespit edilmiştir. Gece çekimi, yüksek yapay ışık altında, gürültülü ortam, yandan ve üstten çekilen fotoğraf üzerinde gerçekleştirilen tahmin değerleri görselde yer almaktadır.



Şekil 4.21 Gece çekimi, yüksek gürültülü ortamda YOLOv8m model ağırlığı ile tespit edilen balon balığı

Şekil 3.21 de yüksek gürültülü ortam, yakın çekim, yapay yeterli ışık seviyesi altında eğitilmiş YOLOv8m modeli ağırlıkları ile tespit edilen balon balığı görülmektedir.



Şekil 4.22 Gündüz çekimi, normal ışık altında YOLOv8m modeli ile tespit edilmiş balon balığı görseli

Şekil 3.22’de gündüz, normal, doğal ışık altında balığın kamufle olabildiği ortamda arkayandan çekilen fotoğrafta eğitilmiş YOLOv8m ağırlıklarıyla gerçekleştirilmiş tahmine ait görüntü yer almaktadır.



Şekil 4.23 Gündüz çekimi, normal ışık altında YOLOv8m modeli ile tespit edilmiş balon balıkları

Şekil 3.23'te gündüz, normal, doğal ışık altında yandan çekilen görseller üzerinde eğitilmiş YOLOv8m model ağırlıklarıyla gerçekleştirilen tahminlere ait görüntü yer almaktadır.



Şekil 4.24 Gürültülü ortamda balon balığı

Şekil 3.24'te gürültünün fazla olduğu ortamda balığın arka sırt kısmından çekilen görüntüsü üzerinden YOLOv8m modeli ile balon balığı tespiti gerçekleştirilmiştir.

DÖRDÜNCÜ BÖLÜM

TARTIŞMA

Bu tezin amacı, bilgisayarla görü ve derin öğrenme algoritmalarının, özellikle en güncel YOLOv8 algoritmasının, balon balığı tespitindeki potansiyelini keşfetmektir. Özellikle gerçek zamanlı olarak balon balığı örneklerini doğru bir şekilde tespit edebilen otomatik bir balon balığı tespit sistemi geliştirmeyi hedefliyoruz ve bu sayede manuel tespit yöntemlerinin zamansal iş gücü ve finansal sınırlamalarını geride bırakmayı hedeflediğimiz çalışmamızda bilgisayar görüntüleme tekniklerindeki ilerlemeleri kullanarak ve güvenilir bir veri kümesini kullanarak, otomatik balon balığı tespit sistemlerinin yetenekleri ve sınırlamaları hakkında bir çalışma ortaya koyduk.

300 epok üzerinden gerçekleştirilen eğitimlerde balon balıklarının kafa kısmından ve yan açılardan daha iyi tespit edilebildiği görülmektedir. Bu durum modelin balon balığı tespitinde balon balığının göz ve çevresini yakalamaya çalıştığından ve balığın şeklini yan açıdan tespit edebildiğinden oluşmaktadır. Bu durumunun ortaya çıkmasında görsellerin manuel etiketlenmesi süreci de etkili olmuştur. Görseller manuel olarak etiketlenirken balon balıklarının yandan görüntüleri ve gövdeleri sınırlayıcı kutular içerisine alınırken balıkların kuyruk ve yüzgeçleri daha az dikkate alınmıştır. Balıkların yüzgeçleri ve kuyrukları ancak net olarak seçilebiliyorlarsa sınırlayıcı kutu içerisine dahil edilmişlerdir. Özellikle balon balıklarının erişkinlik öncesi hallerini içeren görsellerin etiketlenmesinde balon balığı kuyruk ve yüzgeçleri dikkate alınmamış, kuyruk ve yüzgeçleri daha belirgin olan erişkin balon balıklarında kuyruk ve yüzgeçler sınırlayıcı kutu içerisine dahil edilmişlerdir. Bu durum modelin genellemesinde balığın kafa kısmına ve yan açılardan şekline odaklanmasına yol açmıştır.

Şekil 3.18'deki görselde görüldüğü gibi model yeterli ışık altında önden kafa tarafından yaptığı tahminlerde iyi başarımlar göstermektedir.

Şekil 3.19'daki görsel incelendiğinde model başarımının düşük ışık altında düştüğü görülmektedir. Model, balığı kafa kısmından daha düşük bir doğrulukla tahmin etmiştir.

Şekil 3.20'deki görsel incelendiğinde yüksek ışık altında model balığı tespit edebilmiş ancak sınırlayıcı kutuyu çizdirmekte sorun yaşamıştır.

Şekil 3.21'de ortamın yüksek gürültüsü balığın tamamının tespit edilmesinin önüne geçmiş model kafa kısmından daha düşük bir doğrulukla balığı tespit etmeyi başarmıştır.

Şekil 3.22 ve Şekil 3.23'te balon balıkları yüksek doğrulukla tespit edilmiş iken model balıkların kuyruklarını sınırlayıcı kutu dışında bırakmıştır. Bu durum manuel etiketlemede yapılan etiketlemeye bağlı bir sonuç olarak kendini göstermektedir.

Şekil 3.24'teki gibi gürültülü ortamlarda ve balon balığının arkadan yapılan tahminlerinde başarımlar düşük çıkmaktadır. Bu durum da balığın dokusu ve arka plandaki gürültüye bağlı başarımın düştüğünü göstermektedir.

Çok fazla gürültünün olduğu, ışığın farklı olduğu, balon balığının bulunduğu ortam ve diğer balıklar arasında kamufle olduğu görsellerde balığın sınırları ve görselde yer alıp almadığı durumları manuel etiketlemekte karşılaşılan diğer zorluklardandır. Bazen etiketlenmiş bir görselde fark edilmeyen bir balon balığı veri seti üzerinde tekrar yapılan kontrollerde gözlenmiş ve eğitim öncesi etiketler düzeltilmiştir. Çalışmanın ön hazırlığında kullanılan veri setinde arka plan görselleri yer almakta olup eğitim sonuçlarında kötü sonuçlar alınınca tezde kullanılan veri setine bu arka plan görselleri dahil edilmemiştir. Manuel yapılan incelemeler sonrasında tezde kullanılan veri setinde tespit edilememiş 3 arka plan görseli eğitim sırasında Colab ortamında tespit edilmiş ve veri setinden çıkartılmamıştır. Veri seti 1773 görsel(3 tanesi arka plan) eğitim verisi, 450 görsel doğrulama verisi ve 250 görsel test verisinden oluşmaktadır.

Eğitim sürecinde karşılaşılan bir diğer problem videolardan elde edilen fotoğraf karelerinin(frame) tamamının eğitime dahil edilip edilmeyeceği noktasındadır. Araştırmalar sonrası balon balığı içeren bütün fotoğraf karelerini eğitime sokabileceğimizi görmüş olsak da bu durum manuel etiketleme sürecindeki iş yükünü çok arttıracığı için tercih edilmemiştir. Balon balığını temsil eden ve balon balığının daha iyi genellenebilmesi için sualtı ortamındaki zorluklar dikkate alınarak, farklı ışık ve ortam koşullarındaki balıkların farklı açılardan görselleri manuel olarak seçilmiştir. Bu görseller Roboflow sitesinde etiketlendiği ve eğitim doğrulama ve test veri setlerine belirlediğimiz oranlarda rastgele dağıtıldığı için eğitim, doğrulama ve test veri setlerinde benzer görseller yer almıştır, bu da doğrulama ve test veri setlerine ilişkin hassasiyet, duyarlılık ve ortalama hassasiyet değerlerin birbirine daha yakın olmasına neden olmuştur.

Eğitimlerin sonunda YOLOv8 modeli ile beklenen başarı elde edilmiş olup YOLOv8'i geliştiren Ultralytics şirketinin önceki YOLO versiyonu olan YOLOv5 ile 2023 Ocak'ta yayınlanan ve geliştirme süreci devam eden YOLOv8'in başarısı kıyaslanmıştır. YOLOv8 modelleri YOLOv5 modellerine göre daha iyi ortalama hassasiyet başarıları göstermiştir. YOLOv8 modelinin YOLOv5 modeli karşısındaki başarısı Ultralytics şirketinin Şekil 2.11'de verdiği YOLO versiyonlarının başarı kıyaslaması ile paralellik göstermektedir.

İki YOLO sürümünde de balon balığı tespiti için kullanılan modeller eğitilmiş olup eğitim sonuçlarına dair hassasiyet, duyarlılık ve ortalama hassasiyet değerlerini içeren Tablo 3.1 incelendiğinde sonuçlar modelin başarılı olduğu noktasında kabul edilebilir sonuçlardır.

YOLOv8 eğitim sonuçlarına dair grafikleri içeren Şekil 3.1, Şekil 3.2 ve Şekil 3.3'te 250 epok sonrası dfl kaybı değerlerinin yükseldiği görülmektedir. Yine YOLOv5 modellerin eğitimlerine ait Şekil 3.4 ve Şekil 3.5'te yer alan grafiklerde 250 epok sonrası objektif kaybı değerinin yükseldiği gözlenmiştir. Bu durum modelin eğitimi sırasında 250 epok sonrası aşırı öğrenme (overfitting) probleminin ortaya çıktığını göstermektedir.

Modellerin eğitimi sırasında sabır(patience) değerini 100 olarak belirlenmiştir bu değer 100 yerine 50 olarak belirlense eğitim 300 epok tamamlanmadan sonuçlanacaktır. Sabır değerini çok düşük tutmak, model eğitiminin çok erken durdurulması problemine yol açabilmekte ve bu, modelin gerçekten de genellemeyi öğrenme şansını azaltabilmektedir. Sabır değerini dikkatli bir şekilde ayarlamak, modelin yeterli süre boyunca eğitimini sürdürerek hem eğitim verilerine uyum sağlamasını hem de genellemeyi öğrenmesini sağlamak açısından önemlidir. Genel olarak, sabır değeri, modelin eğitim sürecini dengeli bir şekilde yönetmek ve aşırı öğrenme problemini kontrol etmek için kullanılır. Bu değeri doğru bir şekilde ayarlamak, modelin doğrulama verilerindeki performansını izlerken esneklik sağlar ve aşırı öğrenmenin oluşma olasılığını azaltır. Çok düşük bir sabır değeri seçmek, modelin gerçek potansiyelini gösterme şansını engelleyebilir. Bu nedenle sabır değerini seçerken dikkatli olmak önemlidir. Çalışma sonunda modellerin eğitimi için seçtiğimiz sabır değerinin yüksek olduğu görülmektedir. Daha düşük bir sabır değeri belirleyerek eğitimin 250 epok ve öncesi tamamlanması sağlanabilir.

Kutu kaybı, sınıflandırma kaybı ve dfl değerleri modelin karmaşıklığına, veri setine ve eğitim sürecine bağlı olarak değişebilmektedir. Daha düşük dfl kaybı değerleri, modelin daha iyi bir nesne tespiti performansına sahip olduğunu göstermektedir. Elde ettiğimiz sonuçlar, modelin kafa kısmından ve yan açıdan çekilen görüntülerde balon balıklarını doğru sınıflandırabilirken, onların konumunu tam olarak belirlemede zorlandığını göstermektedir. Dfl kaybının diğer değerlere göre yüksek olması, modelin özellikle sınırlayıcı kutu tahminleri konusunda hatalar yaptığını göstermektedir. Bu durumda, tahmin edilen sınırlayıcı kutuların gerçek sınırlayıcı kutularla uyumlu olmadığı veya örtüşme konusunda sorunlar olduğu düşünülebilir. Modelin nadir sınıflara veya zorlu örneklerle başa çıkma konusunda güçlük çektiği ve bu nesneleri doğru bir şekilde sınıflandırmak ve sınırlayıcı kutularını tahmin etmek için daha fazla veriye ihtiyaç duyduğu düşünülebilir. Bu durumda, dfl kaybı bileşeninin toplam

kayıp değeri üzerinde daha büyük bir etkisi olduğunu ve modelin nesne tespiti performansını olumsuz yönde etkilediğini düşünebiliriz.



SONUÇ

Bilgisayarlı görü ve derin öğrenme kullanılarak balon balığı tespiti gerçekleştiren modeli eğittiğimiz bu tez çalışmasında, ESA tabanlı YOLOv8 modelleri ve bu modelin eğitim başarısını karşılaştırmak için seçtiğimiz YOLOv5 modelleri nesne tespitinde kullanılmak için eğitilmiştir. Dalış okullarından temin edilen balon balığı içeren videolardan elde edilen fotoğraf kareleri kullanılarak oluşturulan veri seti 2473 görselden oluşmaktadır. Çalışmaya konu olan görselleri içeren videolar Akdeniz de kayıt altına alınmış videolardır. Akdeniz sualtı ortamına uygun görsellerden özgün bir etiketli veri seti oluşturulmuştur. Çalışma sonrası hassasiyet duyarlılık ve ortalama hassasiyet başarımları üzerinden karşılaştırılarak veri setinde YOLOv8m modeli ile %96,90 ortalama hassasiyet elde edilmiştir.

Sonuç olarak bu tez çalışmasında kullanılan YOLOv8 modelleri ile sualtı ortamlarında balon balığı gibi özel türlerin tespit edilmesinde verim alınabileceği ve tespitin başarı ile yapılabildiği görülmektedir. Sualtı ortamından elde edilecek her türlü video veya fotoğraf görüntüsünün ön işlemeden geçirildikten sonra yürütülecek çalışmalarda kullanılabileceği ve bu çalışmalar sonrası alınan sonuçların değerlendirilebileceği ancak sualtı ortamının kendine has zorlu koşulları ve balıkların sınıflandırmada yaratacağı özel sorunlar, balıkların yüzgeç ve kuyruk gibi yapılarının etiketlemeye dahil edilip edilmeyeceği göz önüne alınırsa daha çok veri içeren veri setleri ile çalışmanın tekrarlanması gerekmektedir. Tüm çalışma süresince elde edilen kesinlik, duyarlılık ve ortalama hassasiyet değerleri yüksek olsa da dfl kaybı değerini azaltmak için farklı optimizasyon stratejileri, hiper parametre (dfl kaybı) ayarlamaları denenebilir. 300 epok üzerinden gerçekleştirilen çalışmada 250 epok sonrası modellerde aşırı öğrenme problemi gözlenmiş olup aşırı öğrenme probleminden kaçınacak adımlar atmak gerekmektedir. Bu tür durumları değerlendirmek için modelinizi ve veri setinizi detaylı bir şekilde analiz etmek önemlidir. Genel olarak değerlendirdiğimizde balon balığı probleminin bilgisayarlı görü yaklaşımı ve derin öğrenme temelli algoritmaları temel alan modellerle çözülebileceği ancak burada çözülmesi gereken kısıtlar olduğu görülmektedir. Modelin bu kısıtların üzerinden gelmesi için nadir sınıflara veya zorlu örneklerle başa çıkma konusunda zorlanması durumunda, daha çok veri içeren veri setleriyle ve veri setinin tespit edilmeye çalışılan nesneyi daha iyi temsil ettiğinden emin olunarak çalışmalar tekrarlanmalıdır. Daha fazla nadir sınıf veya zorlu örnek içeren verilerle modeli performansı artırılabilir.

KAYNAKÇA

- Abiodun, O. I., Jantan, A., Omolara, A. E., Dada, K. V., Mohamed, N. A., & Arshad, H. (2018). State-of-the-art in artificial neural network applications: A survey. *Heliyon*, 4(11).
- Abraham, A. (2005). Artificial neural networks. *Handbook of measuring system design*.
- Ahmad, T., Zhang, D., Huang, C., Zhang, H., Dai, N., Song, Y., & Chen, H. (2021). Artificial intelligence in sustainable energy industry: Status Quo, challenges and opportunities. *Journal of Cleaner Production*, 289, 125834.
- Akyol, O., Unal, V., Ceyhan, T., & Bilecenoglu, M. (2005). First confirmed record of *Lagocephalus sceleratus*(Gmelin, 1789) in the Mediterranean Sea. *Journal of Fish Biology*, 66(4), 1183-1186. <https://doi.org/10.1111/j.0022-1112.2005.00667.x>
- Alashwal, H., El Halaby, M., Crouse, J. J., Abdalla, A., & Moustafa, A. A. (2019). The application of unsupervised clustering methods to Alzheimer's disease. *Frontiers in computational neuroscience*, 13, 31.
- Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017). Understanding of a convolutional neural network. 2017 international conference on engineering and technology (ICET),
- Alimovski, E., & Erdemir, G. (2021). Veri Artırma Tekniklerinin Derin Öğrenmeye Dayalı Yüz Tanıma Sisteminde Etkisi. *İstanbul Sabahattin Zaim Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, 3(1), 76-80. <https://doi.org/10.47769/izufbed.880581>
- Anderson, E. J. (2014). *Bio-inspired robotic fish with vision based target tracking* (Publication Number 1584765) [M.S.M.E., Purdue University]. ProQuest Dissertations & Theses Global. United States -- Indiana. <https://www.proquest.com/dissertations-theses/bio-inspired-robotic-fish-with-vision-based/docview/1660971851/se-2?accountid=15340>
- Antaya, A., Peterson, E., Ralph, T., & Conroy, K. (2019). Lionfish Bot 2.0. Worcester Polytechnic Institute.
- Aradi, S. (2020). Survey of deep reinforcement learning for motion planning of autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 23(2), 740-759.
- Arnold, E., Al-Jarrah, O. Y., Dianati, M., Fallah, S., Oxtoby, D., & Mouzakitis, A. (2019). A survey on 3d object detection methods for autonomous driving applications. *IEEE Transactions on Intelligent Transportation Systems*, 20(10), 3782-3795.
- Audry, S. (2021). *Art in the age of machine learning*. Mit Press.
- B S, R., G N, D. S., Reddy, S., Kakwani, D., & Bhattad, N. (2020). Fish Detection and Classification Using Convolutional Neural Networks. In (pp. 1221-1231). https://doi.org/10.1007/978-3-030-37218-7_128
- Bakrania, A., Joshi, N., Zhao, X., Zheng, G., & Bhat, M. (2023). Artificial intelligence in liver cancers: Decoding the impact of machine learning models in clinical diagnosis of primary liver cancers and liver cancer metastases. *Pharmacological Research*, 189, 106706.
- Bengio, S., Vinyals, O., Jaitly, N., & Shazeer, N. (2015). Scheduled sampling for sequence prediction with recurrent neural networks. *Advances in neural information processing systems*, 28.
- Beyan, C., & Browman, H. I. (2020). Setting the stage for the machine intelligence era in marine science. *ICES Journal of Marine Science*, 77(4), 1267-1273. <https://doi.org/10.1093/icesjms/fsaa084>
- Bharadiya, J. P. (2023). A Comparative Study of Business Intelligence and Artificial Intelligence with Big Data Analytics. *American Journal of Artificial Intelligence*, 7(1), 24.
- Bhardwaj, R., Nambiar, A. R., & Dutta, D. (2017). A study of machine learning in healthcare. 2017 IEEE 41st annual computer software and applications conference (COMPSAC),

- Bicknell, A. W. J., Godley, B. J., Sheehan, E. V., Votier, S. C., & Witt, M. J. (2016). Camera technology for monitoring marine biodiversity and human impact [<https://doi.org/10.1002/fee.1322>]. *Frontiers in Ecology and the Environment*, 14(8), 424-432. <https://doi.org/https://doi.org/10.1002/fee.1322>
- Bochkovskiy, A., Wang, C.-Y., & Hong, Y. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv pre-print server*. <https://doi.org/None> arxiv:2004.10934
- Boswell, K. M., Wilson, M. P., & Cowan Jr, J. H. (2008). A Semiautomated Approach to Estimating Fish Size, Abundance, and Behavior from Dual-Frequency Identification Sonar (DIDSON) Data [<https://doi.org/10.1577/M07-116.1>]. *North American Journal of Fisheries Management*, 28(3), 799-807. <https://doi.org/https://doi.org/10.1577/M07-116.1>
- Čakić, S., Popović, T., Krčo, S., Nedić, D., & Babić, D. (2022). Developing object detection models for camera applications in smart poultry farms. 2022 IEEE International Conference on Omni-layer Intelligent Systems (COINS),
- Cardinale, B. J., Duffy, J. E., Gonzalez, A., Hooper, D. U., Perrings, C., Venail, P., Narwani, A., Mace, G. M., Tilman, D., Wardle, D. A., Kinzig, A. P., Daily, G. C., Loreau, M., Grace, J. B., Larigauderie, A., Srivastava, D. S., & Naeem, S. (2012). Biodiversity loss and its impact on humanity. *Nature*, 486(7401), 59-67. <https://doi.org/10.1038/nature11148>
- Carneiro, T., Da Nóbrega, R. V. M., Nepomuceno, T., Bian, G.-B., De Albuquerque, V. H. C., & Reboucas Filho, P. P. (2018). Performance analysis of google colab as a tool for accelerating deep learning applications. *IEEE Access*, 6, 61677-61685.
- Chien-Yao, W., Bochkovskiy, A., & Hong-Yuan, M. L. (2022). YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Ithaca: Cornell University Library, arXiv.org.
- Cinar, M. E., Bilecenoglu, M., Ozturk, B., Katagan, T., Yokes, M. B., Aysel, V., Dagli, E., Acik, S., Ozcan, T., & Erdogan, H. (2011). An updated review of alien species on the coasts of Turkey. *Mediterranean Marine Science*, 12(2). <https://doi.org/10.12681/mms.34>
- Coccia, M. (2020). Deep learning technology for improving cancer care in society: New directions in cancer imaging driven by artificial intelligence. *Technology in Society*, 60, 101198.
- Crawford, K. (2021). *The atlas of AI: Power, politics, and the planetary costs of artificial intelligence*. Yale University Press.
- Cui, S., Zhou, Y., Wang, Y., & Zhai, L. (2020). Fish Detection Using Deep Learning. *Applied Computational Intelligence and Soft Computing*, 2020, 13. <https://doi.org/https://doi.org/10.1155/2020/3738108>
- Çinar, M. E., Bilecenoglu, M., Yokes, M. B., Öztürk, B., Taskin, E., Bakir, K., Dogan, A., & Açik, S. (2021). Current status (as of end of 2020) of marine alien species in Turkey. *PLoS One*, 16(5). <https://doi.org/ARTN e0251086> 10.1371/journal.pone.0251086
- de la Fuente Garcia, S., Ritchie, C. W., & Luz, S. (2020). Artificial intelligence, speech, and language processing approaches to monitoring Alzheimer's disease: a systematic review. *Journal of Alzheimer's Disease*, 78(4), 1547-1574.
- Demertzis, K., & Iliadis, L. (2018). The Impact of Climate Change on Biodiversity: The Ecological Consequences of Invasive Species in Greece. In W. Leal Filho, E. Manolas, A. M. Azul, U. M. Azeiteiro, & H. McGhie (Eds.), *Handbook of Climate Change Communication: Vol. 1: Theory of Climate Change Communication* (pp. 15-38). Springer International Publishing. https://doi.org/10.1007/978-3-319-69838-0_2

- Ditria, E. M., Lopez-Marcano, S., Sievers, M., Jinks, E. L., Brown, C. J., & Connolly, R. (2020). Automating the Analysis of Fish Abundance Using Object Detection: Optimizing Animal Ecology With Deep Learning. *Frontiers in Marine Science*, 7. <https://doi.org/ARTN10.3389/fmars.2020.00429>
- Duan, Y., Edwards, J. S., & Dwivedi, Y. K. (2019). Artificial intelligence for decision making in the era of Big Data—evolution, challenges and research agenda. *International journal of information management*, 48, 63-71.
- El-Raey, M. (2010). Impacts and implications of climate change for the coastal zones of Egypt. *Coastal zones and climate change*, 7, 31-50.
- Eryilmaz, L., & Dalyan, C. (2006). First record of Apogon queketti Gilchrist (Osteichthyes: Apogonidae) in the Mediterranean Sea. *Journal of Fish Biology*, 69(4), 1251-1254. <https://doi.org/10.1111/j.1095-8649.2006.01185.x>
- Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). From data mining to knowledge discovery in databases. *AI magazine*, 17(3), 37-37.
- Frank, M. R., Wang, D., Cebrian, M., & Rahwan, I. (2019). The evolution of citation graphs in artificial intelligence research. *Nature Machine Intelligence*, 1(2), 79-85.
- Friedman, J. H. (1998). Data Mining and Statistics: What's the connection? *Computing science and statistics*, 29(1), 3-9.
- Golani, D. (1998). Distribution of Lessepsian migrant fish in the Mediterranean. *Italian Journal of Zoology*, 65, 95-99. <https://doi.org/Doi10.1080/11250009809386801>
- Golani, D. (1999). The Gulf of Suez ichthyofauna - Assemblage pool for Lessepsian migration into the Mediterranean. *Israel Journal of Zoology*, 45(1), 79-90. <Go to ISI>://WOS:000080371800007
- Goodell, J. W., Kumar, S., Lim, W. M., & Pattnaik, D. (2021). Artificial intelligence and machine learning in finance: Identifying foundations, themes, and research clusters from bibliometric analysis. *Journal of Behavioral and Experimental Finance*, 32, 100577.
- Grauman, K., & Leibe, B. (2011). *Visual object recognition*. Morgan & Claypool Publishers.
- Grigorescu, S., Trasnea, B., Cocias, T., & Macesanu, G. (2020). A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics*, 37(3), 362-386.
- Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S., & Lew, M. S. (2016). Deep learning for visual understanding: A review. *Neurocomputing*, 187, 27-48.
- Hussain, M. (2023). YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection. *Machines*, 11(7), 677.
- Hussain, M. A. I., Wang, Z.-J., Ali, Z., & Riaz, S. (2021). Automatic Fish Species Classification Using Deep Convolutional Neural Networks. *Wireless Personal Communications*, 116. <https://doi.org/10.1007/s11277-019-06634-1>
- Hwang, T. (2018). Computational power and the social impact of artificial intelligence. *arXiv preprint arXiv:1803.08971*.
- Ivezić, Ž., Connolly, A. J., VanderPlas, J. T., & Gray, A. (2020). *Statistics, data mining, and machine learning in astronomy: a practical Python guide for the analysis of survey data*. Princeton University Press.
- Jafri, R., Ali, S. A., Arabnia, H. R., & Fatima, S. (2014). Computer vision-based object recognition for the visually impaired in an indoors environment: a survey. *The Visual Computer*, 30, 1197-1222.
- Jiang, P., Ergu, D., Liu, F., Cai, Y., & Ma, B. (2022). A Review of Yolo algorithm developments. *Procedia Computer Science*, 199, 1066-1073.
- Jocher, G., Chaurasia, A., Stoken, A., Borovec, J., Kwon, Y., Michael, K., Fang, J., Wong, C., Yifu, Z., & Montes, D. (2022). ultralytics/yolov5: v6. 2-yolov5 classification models, apple m1, reproducibility, clearml and deci. ai integrations. *Zenodo*.

- Jocher, G., & Munawar, M. R. (2023). *Ultralytics YOLOv8 Github Repository*. <https://github.com/ultralytics/ultralytics>
- Jones, K. S. (1994). Natural language processing: a historical review. *Current issues in computational linguistics: in honour of Don Walker*, 3-16.
- Kaehler, A., & Bradski, G. (2016). *Learning OpenCV 3: computer vision in C++ with the OpenCV library*. " O'Reilly Media, Inc."
- Kasai, H., Ziv, N. E., Okazaki, H., Yagishita, S., & Toyoizumi, T. (2021). Spine dynamics in the brain, mental disorders and artificial neural networks. *Nature Reviews Neuroscience*, 22(7), 407-422.
- Kiran, B. R., Sobh, I., Talpaert, V., Mannion, P., Al Sallab, A. A., Yogamani, S., & Pérez, P. (2021). Deep reinforcement learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 23(6), 4909-4926.
- Kou, G., Chao, X., Peng, Y., Alsaadi, F. E., & Herrera Viedma, E. (2019). Machine learning methods for systemic risk analysis in financial sectors.
- Kriegeskorte, N. (2015). Deep neural networks: a new framework for modeling biological vision and brain information processing. *Annual review of vision science*, 1, 417-446.
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet Classification with Deep Convolutional Neural Networks. *Communications of the Acm*, 60(6), 84-90. <https://doi.org/10.1145/3065386>
- Kukreja, H., Bharath, N., Siddesh, C., & Kuldeep, S. (2016). An introduction to artificial neural network. *Int J Adv Res Innov Ideas Educ*, 1, 27-30.
- Kulbicki, M., Bozec, Y.-M., Labrosse, P., Letourneur, Y., Mou-Tham, G., & Wantiez, L. (2005). Diet composition of carnivorous fishes from coral reef lagoons of New Caledonia. *Aquatic Living Resources*, 18(3), 231-250. <https://doi.org/10.1051/alr:2005029>
- Lankowicz, K. M., Bi, H., Liang, D., & Fan, C. (2020). Sonar imaging surveys fill data gaps in forage fish populations in shallow estuarine tributaries. *Fisheries Research*, 226, 105520. <https://doi.org/10.1016/j.fishres.2020.105520>
- Lantsova, E., Voitiuk, T., Zudilova, T., & Kaarna, A. (2016). Using Low-Quality Video Sequences for Fish Detection and Tracking. *Proceedings of the 2016 Sai Computing Conference (Sai)*, 426-433. <Go to ISI>://WOS:000389451900060
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436-444. <https://doi.org/10.1038/nature14539>
- Leo, M., Sharma, S., & Maddulety, K. (2019). Machine learning in banking risk management: A literature review. *Risks*, 7(1), 29.
- Lesgold, A. M., & Mandl, H. (1988). *Learning issues for intelligent tutoring systems*. Springer.
- Li, C., Li, L., Jiang, H., Weng, K., Geng, Y., Li, L., Ke, Z., Li, Q., Cheng, M., & Nie, W. (2022). YOLOv6: A single-stage object detection framework for industrial applications. *arXiv preprint arXiv:2209.02976*.
- Li, D., Li, X., Wang, Q., & Yinfeng, H. (2022). Advanced Techniques for the Intelligent Diagnosis of Fish Diseases: A Review. *Animals*, 12(21), 2938. <https://doi.org/10.3390/ani12212938>
- Li, X., Shang, M., Qin, H., & Chen, L. (2015). *Fast accurate fish detection and recognition of underwater images with Fast R-CNN*. <https://doi.org/10.23919/OCEANS.2015.7404464>
- Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. *Proceedings of the IEEE international conference on computer vision*,
- Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft coco: Common objects in context. *Computer Vision—ECCV*

- 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13,
- Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y., & Alsaadi, F. E. (2017). A survey of deep neural network architectures and their applications. *Neurocomputing*, 234, 11-26.
- Lu, Y. (2019). Artificial intelligence: a survey on evolution, models, applications and future trends. *Journal of Management Analytics*, 6(1), 1-29.
- Mahdavinejad, M. S., Rezvan, M., Barekatin, M., Adibi, P., Barnaghi, P., & Sheth, A. P. (2018). Machine learning for Internet of Things data analysis: A survey. *Digital Communications and Networks*, 4(3), 161-175.
- Mahesh, B. (2020). Machine learning algorithms-a review. *International Journal of Science and Research (IJSR)*. [Internet], 9(1), 381-386.
- Maind, S. B., & Wankar, P. (2014). Research paper on basic of artificial neural network. *International Journal on Recent and Innovation Trends in Computing and Communication*, 2(1), 96-100.
- Mavruk, S., & Avsar, D. (2008). Non-native fishes in the Mediterranean from the Red Sea, by way of the Suez Canal. *Reviews in fish biology and fisheries*, 18, 251-262.
- Miikkulainen, R., Liang, J., Meyerson, E., Rawal, A., Fink, D., Francon, O., Raju, B., Shahrzad, H., Navruzian, A., & Duffy, N. (2019). Evolving deep neural networks. In *Artificial intelligence in the age of neural networks and brain computing* (pp. 293-312). Elsevier.
- Mohamed, H. E., Fadl, A., Anas, O., Wageeh, Y., ElMasry, N., Nabil, A., & Atia, A. (2020). MSR-YOLO: Method to Enhance Fish Detection and Tracking in Fish Farms. *11th International Conference on Ambient Systems, Networks and Technologies (Ant) / the 3rd International Conference on Emerging Data and Industry 4.0 (Edi40) / Affiliated Workshops*, 170, 539-546. <https://doi.org/10.1016/j.procs.2020.03.123>
- Mohan, A., Papageorgiou, C., & Poggio, T. (2001). Example-based object detection in images by components. *IEEE transactions on pattern analysis and machine intelligence*, 23(4), 349-361.
- Nagpal, A., & Gabrani, G. (2019, 4-6 Feb. 2019). Python for Data Analytics, Scientific and Technical Applications. 2019 Amity International Conference on Artificial Intelligence (AICAI),
- Najafabadi, M. M., Villanustre, F., Khoshgoftaar, T. M., Seliya, N., Wald, R., & Muharemagic, E. (2015). Deep learning applications and challenges in big data analytics. *Journal of big data*, 2(1), 1-21.
- Ngiam, K. Y., & Khor, W. (2019). Big data and machine learning algorithms for health-care delivery. *The Lancet Oncology*, 20(5), e262-e273.
- Otter, D. W., Medina, J. R., & Kalita, J. K. (2020). A survey of the usages of deep learning for natural language processing. *Ieee Transactions on Neural Networks and Learning Systems*, 32(2), 604-624.
- Patel, A. A. (2019). *Hands-on unsupervised learning using Python: how to build applied machine learning solutions from unlabeled data*. O'Reilly Media.
- Patro, K. S. K., Yadav, V. K., Bharti, V. S., Sharma, A., & Sharma, A. (2023). Fish Detection in Underwater Environments Using Deep Learning. *National Academy Science Letters*, 46(5), 407-412. <https://doi.org/10.1007/s40009-023-01265-4>
- Pinaya, W. H. L., Vieira, S., Garcia-Dias, R., & Mechelli, A. (2020). Convolutional neural networks. In *Machine learning* (pp. 173-191). Elsevier.
- Por, F. D. (1971). 100 Years of Suez-Canal - Century of Lessepsian Migration - Retrospect and Viewpoints. *Systematic Zoology*, 20(2), 138-&. <https://doi.org/Doi 10.2307/2412054>
- Pouyanfar, S., Sadiq, S., Yan, Y., Tian, H., Tao, Y., Reyes, M. P., Shyu, M.-L., Chen, S.-C., & Iyengar, S. S. (2018). A survey on deep learning: Algorithms, techniques, and applications. *ACM Computing Surveys (CSUR)*, 51(5), 1-36.

- RangeKing. (2023). *Brief summary of YOLOv8 model structure*. Retrieved 25.09 from <https://github.com/ultralytics/ultralytics/issues/189>
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. In. Ithaca: Cornell University Library, arXiv.org.
- Redmon, J., & Farhadi, A. (2016). YOLO9000: Better, Faster, Stronger. *arXiv pre-print server*. <https://doi.org/None>
arxiv:1612.08242
- Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. In. Ithaca: Cornell University Library, arXiv.org.
- Rodrigue, J.-P. (2020). *The geography of transport systems*. Routledge.
- Saabith, S., Vinothraj, T., & Fareez, M. (2021). A review on Python libraries and Ides for Data Science. *Int. J. Res. Eng. Sci*, 9(11), 36-53.
- Sallab, A. E., Abdou, M., Perot, E., & Yogamani, S. (2017). Deep reinforcement learning framework for autonomous driving. *arXiv preprint arXiv:1704.02532*.
- Salman, A., Siddiqui, S., Shafait, F., Mian, A., Shortis, M., Khurshid, K., Ulges, A., & Schwanecke, U. (2020). Automatic fish detection in underwater videos by a deep neural network based hybrid motion learning system. *ICES Journal of Marine Sciences*, 77(4), 1295-1307. <https://doi.org/10.1093/icesjms/fsz025>
- Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, 61, 85-117. <https://doi.org/10.1016/j.neunet.2014.09.003>
- Schneider, S., Taylor, G. W., Linquist, S., & Kremer, S. C. (2019). Past, present and future approaches using computer vision for animal re-identification from camera trap data [<https://doi.org/10.1111/2041-210X.13133>]. *Methods in Ecology and Evolution*, 10(4), 461-470. <https://doi.org/https://doi.org/10.1111/2041-210X.13133>
- Shinde, P. P., & Shah, S. (2018). A review of machine learning and deep learning applications. 2018 Fourth international conference on computing communication control and automation (ICCUBEA),
- Simonyan, K., & Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv pre-print server*. <https://doi.org/None>
arxiv:1409.1556
- Spampinato, C., Chen-Burger, Y. H., Nadarajan, G., & Fisher, R. B. (2008). Detecting, tracking and counting fish in low quality unconstrained underwater videos. *Visapp 2008: Proceedings of the Third International Conference on Computer Vision Theory and Applications, Vol 2*, 514-+. <Go to ISI>://WOS:000256791600081
- Sultana, F., Sufian, A., & Dutta, P. (2020). A Review of Object Detection Models Based on Convolutional Neural Network. In (pp. 1-16). https://doi.org/10.1007/978-981-15-4288-6_1
- Sutton, S. G., Holt, M., & Arnold, V. (2016). “The reports of my death are greatly exaggerated”—Artificial intelligence research in accounting. *International Journal of Accounting Information Systems*, 22, 60-73.
- Tammina, S. (2019). Transfer learning using vgg-16 with deep convolutional neural network for classifying images. *International Journal of Scientific and Research Publications (IJSRP)*, 9(10), 143-150.
- Terven, J., & Cordova-Esparza, D. (2023). A comprehensive review of YOLO: From YOLOv1 to YOLOv8 and beyond. *arXiv preprint arXiv:2304.00501*.
- Turing, A. M. (2021). Computing Machinery and Intelligence (1950). *Ideas That Created the Future*, 147-164. <Go to ISI>://WOS:001038769700016
- Tüzün, S. (2012). *GROWTH CHARACTERISTICS OF THE SILVERSTRIPE BLAASOP (Lagocephalus sceleratus Gmelin, 1789) IN ANTALYA BAY* Adnan Menderes University]. <http://adudspace.adu.edu.tr:8080/xmlui/bitstream/handle/11607/977/SUNA%20T%c3%9cZ%c3%9cN.pdf?sequence=3&isAllowed=y>

- Ulman, A., Yildiz, T., Demirel, N., Canak, O., Yemişken, E., & Pauly, D. (2021). The biology and ecology of the invasive silver-cheeked toadfish (*Lagocephalus sceleratus*), with emphasis on the Eastern Mediterranean. *NeoBiota*, 68, 145-175.
- Van Rossum, G. (2007). Python Programming Language. USENIX annual technical conference,
- Villon, S., Mouillot, D., Chaumont, M., Darling, E., Subsol, G., Claverie, T., & Villéger, S. (2018). A Deep learning method for accurate and fast identification of coral reef fishes in underwater images. *Ecological Informatics*, 48. <https://doi.org/10.1016/j.ecoinf.2018.09.007>
- Wang, H., Zhang, S., Zhao, S., Wang, Q., Li, D., & Zhao, R. (2022). Real-time detection and tracking of fish abnormal behavior based on improved YOLOV5 and SiamRPN++. *Computers and Electronics in Agriculture*, 192, 106512. <https://doi.org/https://doi.org/10.1016/j.compag.2021.106512>
- Wang, N., Wang, Y., & Er, M. J. (2022). Review on deep learning techniques for marine object recognition: Architectures and algorithms. *Control Engineering Practice*, 118, 104458.
- Webb, M. (2019). The impact of artificial intelligence on the labor market. *Available at SSRN* 3482150.
- Xiao, T., Xia, T., Yang, Y., Huang, C., & Wang, X. (2015). Learning from massive noisy labeled data for image classification. Proceedings of the IEEE conference on computer vision and pattern recognition,
- Yasnitsky, L. N. (2020). Whether be new “Winter” of artificial intelligence? Integrated Science in Digital Age: ICIS 2019,
- Young, T., Hazarika, D., Poria, S., & Cambria, E. (2018). Recent trends in deep learning based natural language processing. *ieee Computational intelligenCe magazine*, 13(3), 55-75.
- Zenetos, A., Meric, E., Verlaque, M., Galli, P., Boudouresque, C. F., Giangrande, A., Cinar, M. E., & Bilecenoglu, M. (2008). Additions to the annotated list of marine alien biota in the Mediterranean with special emphasis on Foraminifera and Parasites. *Mediterranean Marine Science*, 9(1). <https://doi.org/10.12681/mms.146>
- Zenetos, A., Pancucci-Papadopoulou, M. A., Zogaris, S., Papastergiadou, E., Vardakas, L., Aligizaki, K., & Economou, A. N. (2009). Aquatic alien species in Greece (2009): tracking sources, patterns and effects on the ecosystem. *Journal of Biological Research-Thessaloniki*, 12, 135-172. <Go to ISI>://WOS:000272794100003
- Zhang, C., & Lu, Y. (2021). Study on artificial intelligence: The state of the art and future prospects. *Journal of Industrial Information Integration*, 23, 100224.
- Zhang, Q., Yang, L. T., Chen, Z., & Li, P. (2018). A survey on deep learning for big data. *Information Fusion*, 42, 146-157.
- Zhao, W., Queralta, J. P., & Westerlund, T. (2020). Sim-to-real transfer in deep reinforcement learning for robotics: a survey. 2020 IEEE symposium series on computational intelligence (SSCI),
- Zhao, Z. Q., Zheng, P., Xu, S. T., & Wu, X. D. (2019). Object Detection With Deep Learning: A Review. *Ieee Transactions on Neural Networks and Learning Systems*, 30(11), 3212-3232. <https://doi.org/10.1109/Tnnls.2018.2876865>
- Zhou, L., Pan, S., Wang, J., & Vasilakos, A. V. (2017). Machine learning on big data: Opportunities and challenges. *Neurocomputing*, 237, 350-361.
- Zhou, X., Gong, W., Fu, W., & Du, F. (2017). Application of deep learning in object detection. 2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS),

ÖZGEÇMİŞ

Adı ve SOYADI	Hüseyin Umut YÜKSEL
EĞİTİM DURUMU	
Mezun Olduğu Lise	Hacı Malike Mehmet Bileydi Anadolu Lisesi, Antalya, 2011
Lisans Diploması	Süleyman Demirel Üniversitesi Bilgisayar Mühendisliği, Isparta, 2018
Tezsiz Yüksek Lisans Diploması	
Dönem Proje Konusu	
Tezli Yüksek Lisans Diploması	
Yüksek Lisans Tez Konusu	Bilgisayarlı Görü ve Derin Öğrenme ile Balon Balığı Tespiti
Doktora Diploması	
Doktora Tez Konusu	
Yabancı Dil / Diller	İngilizce
BİLİMSEL FAALİYETLER	
İŞ DENEYİMİ	
Stajlar	Karayolları 13. Bölge Müdürlüğü Bilgi Teknolojileri Başmühendisliği
Projeler	
Çalıştığı Kurumlar	Teomanpaşa Halk Eğitimi Merkezi, Çamlıbel Halk Eğitimi Merkezi