

T.C.
BİTLİS EREN ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

DÖRT MOTORLU İNSANSIZ HAVA ARACIYLA
İNSAN TESPİTİ

YÜKSEK LİSANS TEZİ

VEYSİ AKGÜN

DANIŞMAN
DOÇ. DR. YAKUP ŞAHİN

İKİNCİ DANIŞMAN
DOÇ. DR. VEDAT TÜMEN

EYLÜL 2024
BİTLİS

T.C.
BİTLİS EREN ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

DÖRT MOTORLU İNSANSIZ HAVA ARACIYLA
İNSAN TESPİTİ

YÜKSEK LİSANS TEZİ

VEYSİ AKGÜN

ORCID: 0009-0001-2192-7371

DANIŞMAN

DOÇ. DR. YAKUP ŞAHİN

İKİNCİ DANIŞMAN

DOÇ. DR. VEDAT TÜMEN

EYLÜL 2024

BİTLİS

T.C.

BİTLİS EREN ÜNİVERSİTESİ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

YÜKSEK LİSANS TEZ ÇALIŞMASI

ETİK BEYANI

Lisansüstü Eğitim Enstitüsü Elektrik Elektronik Mühendisliği Anabilim Dalı Yüksek Lisans öğrencisiyim. Hazırlamış olduğum “Dört Motorlu İnsansız Hava Aracıyla İnsan Tespiti” başlıklı tezde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi; tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu; tez çalışmasında yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi; kullanılan verilerde herhangi bir değişiklik yapmadığımı; bu tezde sunduğum çalışmanın özgün olduğunu bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim./...../20.....

İmza

Veysi AKGÜN

T.C.
BİTLİS EREN ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
TEZ YAZIM KILAVUZU UYGUNLUK BEYANI

“Dört Motorlu İnsansız Hava Aracıyla İnsan Tespiti” başlıklı yüksek lisans Bitlis Eren Üniversitesi Lisansüstü Eğitim Enstitüsü Tez Yazım Kılavuzuna uygun olarak hazırlanmıştır./...../20.....

Tezi Hazırlayan

İmza

Veysi AKGÜN

Danışman

İmza

Doç. Dr. Yakup ŞAHİN

Elektrik Elektronik Mühendisliği Anabilim Dalı Başkanı

İmza

Prof. Dr. Sabir RÜSTEMLİ

T.C.
BİTLİS EREN ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
TEZ ONAY SAYFASI

Bitlis Eren Üniversitesi Lisansüstü Eğitim Enstitüsü Elektrik Elektronik Mühendisliği Anabilim Dalı öğrencisi Veysi AKGÜN tarafından hazırlanan “Dört Motorlu İnsansız Hava Aracıyla İnsan Tespiti” adlı Yüksek Lisans Tezi ile ilgili tez savunma sınavı,/...../20.... tarihinde yapılmış ve tezin oy birliği/ oy çokluğu ile kabul/ ret edilmesine karar verilmiştir.

JÜRİ:

İMZA

Danışman: Doç. Dr. Yakup ŞAHİN
(Bitlis Eren Üniversitesi)

Üye: Doç. Dr. Rıdvan UMAZ
(Bitlis Eren Üniversitesi)

Üye: Doç. Dr. Naim Süleyman TINGİ
(Erzincan Binali Yıldırım Üniversitesi)

Bitlis Eren Üniversitesi Lisansüstü Eğitim Enstitüsü Yönetim Kurulunun tarih ve sayılı kararıyla jüri tarafından kabul edilmiş bu çalışmanın yüksek lisans tezi olarak kabulü onaylanmıştır.

....../...../2024

Doç. Dr. Mehmet Bakır ŞENGÜL
Enstitü Müdürü

T.C.
Bitlis Eren Üniversitesi Lisansüstü Eğitim Enstitüsü
Elektrik Elektronik Mühendisliği Anabilim Dalı
DÖRT MOTORLU İNSANSIZ HAVA ARACIYLA İNSAN TESPİTİ
Yüksek Lisans Tezi
Veysi AKGÜN
Danışman: Doç. Dr. Yakup ŞAHİN
İkinci Danışman: Doç. Dr. Vedat TÜMEN
Eylül 2024
ÖZET

Bu tezde, dört rotorlu insansız hava aracı (İHA) ile çekilen fotoğraflardan elde edilen bir veri seti kullanarak, insan tespiti alanında en son teknolojilerden olan YOLO (You Only Look Once) algoritmalarının (YOLOv7, YOLOv8 ve YOLOv10) performansını karşılaştırmayı amaçlamaktadır. Veri seti, çeşitli insan pozisyonlarını içeren zengin bir görüntü koleksiyonundan oluşmakta olup, eğitim süreci Google Colab ortamında gerçekleştirilmiştir. Bu tezde, her bir YOLO sürümünün eğitim süreçleri detaylı bir şekilde ele alınmış ve karşılaştırmalar yapılmıştır. YOLOv7, YOLOv8 ve YOLOv10'un performansları, insan tespiti doğruluğu ve genel verimlilik açısından değerlendirilmiştir. Algoritmaların performansları, doğruluk, hassasiyet, duyarlılık, F1 skoru ve mAP50 gibi kriterler üzerinden değerlendirilmiştir. YOLOv8, en yüksek doğruluk (%89,6) ve mAP50 (%97,4) değerleri ile diğer iki algoritmadan daha etkili bulunmuştur. YOLOv7 ve YOLOv10 ise sırasıyla %82,8 ve %88,0 doğruluk oranlarına sahiptir. Bu bulgular, insansız hava araçları ile gerçekleştirilen görüntüleme ve nesne tanıma uygulamalarında hangi algoritmanın daha etkili olduğunu belirlemek için önemli bir temel sunmaktadır. Çalışmanın sonuçları, gelecekteki araştırmalar ve uygulamalar için yol gösterici niteliktedir.

Anahtar Kelimeler: İnsansız Hava Aracı, Nesne Tanıma, YOLO Algoritmaları, YOLOv7, YOLOv8, YOLOv10

Republic of Türkiye
Bitlis Eren University Graduate School
Department of Electrical and Electronics Engineering
HUMAN DETECTION WITH FOUR ENGINE UNMANNED AERIAL
VEHICLE
Master's Thesis
Veysi AKGÜN
Supervisor: Doç. Dr. Yakup ŞAHİN
Co-Supervisor: Doç. Dr. Vedat TÜMEN
September 2024
ABSTRACT

In this thesis, it aims to compare the performance of YOLO (You Only Look Once) algorithms (YOLOv7, YOLOv8 and YOLOv10), which are among the latest technologies in the field of human detection, by using a data set obtained from photographs taken with a four-rotor unmanned aerial vehicle (UAV). The dataset consists of a rich collection of images containing various human positions, and the training process was carried out in the Google Colab environment. In this thesis, the training processes of each YOLO version are discussed in detail and comparisons are made. The performances of YOLOv7, YOLOv8 and YOLOv10 have been evaluated for human detection accuracy and overall efficiency. The performance of the algorithms was evaluated on criteria such as accuracy, sensitivity, sensitivity, F1 score and mAP50. YOLOv8 was found to be more effective than the other two algorithms with the highest accuracy (89.6%) and mAP50 (97.4%) values. YOLOv7 and YOLOv10, on the other hand, have accuracy rates of 82.8% and 88.0%, respectively. These findings provide an important basis for determining which algorithm is more effective in imaging and object recognition applications performed with unmanned aerial vehicles. The results of the study are a guide for future research and applications.

Keywords: Unmanned Aerial Vehicle, Object Recognition, YOLO Algorithms, YOLOv7, YOLOv8, YOLOv10

TEŞEKKÜR

Öncelikle, tez danışmanım Sayın Doç. Dr. Yakup ŞAHİN ve Doç. Dr. Vedat TÜMEN'e, bana sağladıkları değerli rehberlik ve sürekli destekleri için minnettarım. Bilgi ve deneyimleri ile bu süreci daha anlamlı hale getirdiler. Eleştirileri ve önerileriyle çalışmamı geliştirmeme yardımcı oldular.

Bu çalışma, Bitlis Eren Üniversitesi Bilimsel Araştırma Projeleri Koordinatörlüğü tarafından 2022.05 numaralı proje kapsamında desteklenmiştir.

Ayrıca, aileme sonsuz teşekkürlerimi iletmek istiyorum. Her zaman yanımda oldunuz, beni desteklediniz ve bu zorlu süreçte motivasyon kaynağım oldunuz. Sabrınız ve sevginiz, bu başarıyı elde etmemde büyük rol oynadı.

İÇİNDEKİLER

ÖZET.....	I
ABSTRACT	II
TEŞEKKÜR	III
İÇİNDEKİLER	IV
ŞEKİLLER DİZİNİ	VIII
KISALTMALAR	X
1. GİRİŞ	1
1.1. Literatür Taraması	1
1.2. Tezin Amacı	5
2. GENEL BİLGİLER	6
2.1. İnsansız Hava Araçları (İHA)	6
2.2. Sabit kanatlı İHA'lar	7
2.3. Döner kanatlı İHA'lar	8
2.4. Dört Rotorlu İnsansız Hava Aracı	9
2.5. İnsan Tespiti	12
2.6. Görüntü işleme	12
2.6.1. Filtreleme	12
2.6.1.1. Ortalama Filtre (Mean Filter - Box Blur)	13
2.6.1.2. Medyan Filtresi (Median Filter)	13
2.6.1.3. Gauss Düzleştirme Filtresi (Gaussian Smoothing)	14
2.6.2. Kenar Tespiti	15
2.6.3. Hough Dönüşümü	15
2.6.4. Nesne Tanıma ve Sınıflandırma	16
2.6.5. Segmentasyon	17
2.6.6. Özellik Çıkarma	17
2.6.7. Tuz – Biber kirliliğinin kaldırılması	18

2.6.8. Keskinliği artırma.....	18
2.6.8.1. Unsharp Masking (Keskinleştirme)	19
2.6.8.2. Laplacian Kenar Algılama	19
2.6.8.3. Kontrast Artırma	19
2.6.8.4. Kenar Belirginleştirme Filtreleri.....	19
2.6.8.5. Wavelet Dönüşümü	20
2.6.8.6. Histogram Eşitleme.....	20
2.6.9. Kenar çıkarma algoritması	20
2.7. Isı haritası	20
2.8. Derin Öğrenme.....	20
2.9. Derin Öğrenme Modelleri	22
2.10. YOLOv7.....	23
2.10.1. Input Size (Giriş Boyutu)	23
2.10.2. Batch Size (Toplu Boyut).....	24
2.10.3. Number of Classes (Sınıf Sayısı)	24
2.10.4. Confidence Threshold (Güven Eşiği)	24
2.10.5. NMS Threshold (NMS Eşiği)	25
2.10.6. Training Parameters (Eğitim Parametreleri).....	25
2.10.7. Inference Time (Çıkarım Süresi).....	26
2.11. YOLOv8.....	29
2.12. YOLOv10.....	30
2.13. Ultralytics YOLO Performans Metrikleri	31
2.13.1. Doğruluk (Accuracy)	31
2.13.2. Hassasiyet (Precision)	31
2.13.3. Duyarlılık (Recall).....	32
2.13.4. F1 Skoru.....	32
2.13.5. mAP50.....	32
2.14. K-Fold Cross Validation.....	32

3. MATERYAL VE YÖNTEM.....	34
3.1. HERİDAL.....	35
3.2. Okutama-Action.....	36
3.3. Derin Öğrenme Modeli Seçilmesi ve Kullanımı	36
3.4. Nemrut Kalderası Kırsalından Elde Edilen Veri Seti	38
3.5. Bitlis Eren Üniversitesi Kampüsü İçinde Elde Edilen Veri Seti	39
3.6. Model Eğitim Ortamının Hazırlanması ve Eğitimlerin Yapılması	40
3.7. Kurulumlar.....	40
3.8. Eğitim Parametrelerinin Ayarlanması.....	41
3.9. Veri Seti.....	42
3.10. Eğitim Süreci	43
3.11. YOLOv7 ile Elde Edilen Sonuçlar.....	44
3.12. YOLOv8 ile Elde Edilen Sonuçlar.....	49
3.13. YOLOv10 ile Elde Edilen Sonuçlar.....	54
4. SONUÇ VE ÖNERİLER.....	61
KAYNAKLAR	62

ÇİZELGELER DİZİNİ

Çizelge 2.1. Sabit kanat İHA ile döner kanat İHA karşılaştırılması.....	11
Çizelge 2.2. Performans Karşılaştırması.....	26
Çizelge 2.3. Verim Karşılaştırması.....	29
Çizelge 3.1. Kullanılan hiperparametreler	34
Çizelge 3.2. Google Colab'daki T4 GPU'nun teknik özellikleri	35
Çizelge 3.3. Yazılım kütüphaneleri ve versiyonları	41
Çizelge 3.4. Gerçekleştirilen eğitimde kullanılan hiper parametreler	60
Çizelge 3.5. Nesne sınıflarının koordinat ve boyut bilgileri.....	42
Çizelge 3.6. Veri setimizdeki nesne sınıflarının koordinat ve boyut bilgileri örneği.....	43
Çizelge 3.7. Fold bazında mAP50 ve mAP değerleri	44
Çizelge 3.8. YOLOv7 Algoritmasının Farklı Katmanlar İçin Performans Değerleri.....	48
Çizelge 3.9. Fold bazında mAP50 ve mAP değerleri	49
Çizelge 3.10. YOLOv8 algoritmasının farklı katmanlar için performans değerleri	54
Çizelge 3.11. Fold bazında mAP50 ve mAP değerleri	55
Çizelge 3.12. YOLOv10 algoritmasının farklı katmanlar için performans değerleri	59
Çizelge 3.13. Farklı nesne tanıma modellerinin performans metrikleri karşılaştırması	60
Çizelge 4. 1. YOLO algoritmalarının performans karşılaştırması	61

ŞEKİLLER DİZİNİ

Şekil 2.1. Sistem yapısı.....	7
Şekil 2.2. Hava aracı kategorileri.....	7
Şekil 2.3. Sabit kanatlı İHA'lar	7
Şekil 2.4. Döner kanatlı İHA'lar.....	8
Şekil 2.5. Drone hareket sistemi	9
Şekil 2.6. Quadcopter hareketleri	9
Şekil 2.7. Nesne tespit uygulamalar.....	12
Şekil 2.8. Mean (Ortalama).....	13
Şekil 2.9. Ortalama filtre örneği	13
Şekil 2.10. Medyan filtresi.....	14
Şekil 2.11. Medyan filtresi örneği	14
Şekil 2.12. Gauss filtresi	15
Şekil 2.13. Hough dönüşümünde yarı çap tespiti	16
Şekil 2.14. Segmentlere ayrılmış nesneler örnek görsel	17
Şekil 2.15. Tuz ve biber gürültüsü eklenmiş görüntü üzerinde filtrelerin etkileri.....	18
Şekil 2.16. Unsharp masking örneği.....	19
Şekil 2.17. Derin Öğrenme	21
Şekil 2.18. Sinir Hücresi Yapısı.....	21
Şekil 2.19. Yapay sinir yapısının basit modellemesi	22
Şekil 2.20. Perceptron (Algılayıcı)	22
Şekil 2.21. CIFAR-10 veri setinden örnek	24
Şekil 2.22. Maksimum Olmayan Bastırmanın etkisi	25
Şekil 2.23. Inference Time.....	26
Şekil 2.24. Normal model	27
Şekil 2.25. Modele aux head eklenmiş hali	27
Şekil 2.26. Independent assigner	28
Şekil 2.27. Lead guided assigner and coarse-to-fine lead guided assigner.....	28
Şekil 2.28. YOLOv8 Performans Karşılaştırması	29
Şekil 2.29. YOLOv10 eğitim sürecinde çift etiketleme ve eşleşme metrikleri	30
Şekil 2.30. Gecikme-doğruluk (solda) ve boyut-doğruluk (sağda) ödünleşimleri açısından diğerleriyle karşılaştırmalar	30
Şekil 2.31. Karışıklık matrisi	31

Şekil 2.32. K-Katlı çapraz doğrulama	33
Şekil 3.1. HERIDAL veri kümesinde bulunan ve insan içeren görüntülerden örnekler	35
Şekil 3.2. Okutama-Action veri kümesinde insan bulunan görüntüler	36
Şekil 3.3. Global modellerin başarımların grafikleri	37
Şekil 3.4. Yolo modellerinin performans karşılaştırması	38
Şekil 3.5. Nemrut kalderası veri kümesinden görüntüler	39
Şekil 3.6. Bitlis Kampüs veri kümesinden görüntüler	40
Şekil 3.7. YOLO etiket dosyasının içerik örneği.....	42
Şekil 3.8. Veri setimize ait görüntü örneği	43
Şekil 3.9. Google Colab üzerinde eğitim görseli	44
Şekil 3.10. Eğitilen modele ait karmaşıklık matrisi.....	45
Şekil 3.11. Yolov7 ile eğitim sonucunda elde edilen F1 skor - güven grafiği.....	45
Şekil 3.12. Eğitilen modelin test görüntülerinden tespit örnekleri	46
Şekil 3.13. YOLOv7 Nesne tespiti modeli eğitim ve değerlendirme metrikleri	47
Şekil 3.14. Yolov7 modeline ait hatalı tespit örnekleri	49
Şekil 3.15. Eğitilen modele ait karmaşıklık matrisi.....	50
Şekil 3.16. Eğitim sonucunda elde edilen F1 skor - güven grafiği.....	51
Şekil 3.17. Eğitilen modelin test görüntülerinden tespit örnekleri	52
Şekil 3.18. YOLOv8 Nesne tespiti modeli eğitim ve değerlendirme metrikleri	53
Şekil 3.19. Yolov8 modeline ait hatalı tespit örnekleri	54
Şekil 3.20. Eğitilen modele ait karmaşıklık matrisi.....	55
Şekil 3.21. Eğitim sonucunda elde edilen F1 skor - güven grafiği.....	56
Şekil 3.22. Eğitilen modelin test görüntülerinden tespit örnekleri	57
Şekil 3.23. YOLOv10 Nesne tespiti modeli eğitim ve değerlendirme metrikleri	58
Şekil 3.24. Yolov10 modeline ait hatalı tespit örnekleri	59

KISALTMALAR

AP	: Ortalama Hassasiyet
CNN	: Convolutional Neural Networks
GPS	: Küresel Konumlama Sistemi
İHA	: İnsansız Hava Aracı
NMS	: Non-Maximum Suppression
R-CNN	: Region-based Convolutional Neural network
RPN	: Bölgesel Öneri Ağı
UAV	: Unmanned Aerial Vehicle
YOLO	: You Only Look Once

1. GİRİŞ

Teknolojik alanlarda yaşanan makinelerin kontrol ve tasarım uygulamalarında yeni bir döneme hızla yaklaşıyoruz. Yapay zekâ uygulamaları hayatımızın birçok alanına entegre oluyor ve küresel ölçekte büyük bir fark yaratmaktadır. Sağlıktan ulaşım, üretimden güvenliğe, eğitimden eğlenceye pek çok sektörü dönüştürmektedir[1]. İnsanların olduğu her yerde kendi can ve mal güvenliklerini korumak birinci öncelikleri olmuştur. Geçmişte insanlar bu güvenlik ihtiyaçlarını ilkel metotlarla sağlarken son yıllarda teknolojinin gelişmesine paralel olarak insansız hava araçlarını bu amaçla kullanma eğilimi oldukça artmıştır. İnsanların can ve mal güvenliklerini manuel olarak kontrol etmek oldukça meşakkatli bir süreci gerektirmekte ve bunun için ciddi bir insan kaynağına yatırım yapılmasını gerektirmektedir. Bu insan yatırımından kurtulmanın ekonomik yollarından biri de İHA'lar la (İnsansız Hava Aracı) insan tespiti yapmaktır. Ve dünya genelinde kullanılan insansız hava aracı sayısı sürekli artış göstermektedir[2].

İHA'lar birçok alanda önemli rol oynuyor ve insanları tespit edebilmektedir. Bunlar, askeri personelin düşman varlığını tespit etme, sınırlardan izinsiz giriş ve çıkışları takip etme, gibi işlerde görev almasına olanak tanıyarak sahada risk almadan bilgi toplamasına olanak tanıyor. Kayıp kişilerin bulunması için hızlı ve etkili bir yöntem sunarlar. Hasar tespiti ve kurtarma operasyonlarında insanların yerini tespit etme sürecinde büyük önem taşırlar. Kalabalık insan gruplarını izlemek ve güvenliği sağlamak için kullanılabilirler. Bu araçların kullanım alanlarına baktığımızda toplumu ilgilendiren farklı alanlar karşımıza çıkmaktadır. Kullanım alanlarının en dikkat çekici amaçlarından biri de korunması gereken özel alanlar gelmektedir. Bazı özel alanlar hayvanlar veya insanlardan korunması gerekebilir. Bunun yanında çevresel etkileri inceleme, askeri alanlar, ziraat ve ormancılık, medya ve haberleşme, bilimsel araştırmalar, doğal afetlerde afet sonrası etkilenen bölgeleri insansız hava araçları ile taramak ve tahliye edilenleri tespit etmektir. Başka bir kullanım amacı da dezavantajlı bölgelerde acil hastalara tıbbi malzeme ulaştırmak ve gıda malzemesine ulaşma imkânı olmayanlara gıda tedariki sağlamaktır[3].

1.1. Literatür Taraması

Literatür çalışmalarına bakıldığında insan tespiti ile ilgili çalışmaların görüntü işleme, makine öğrenimi ve yapay zekâ ile yapıldığı karşımıza çıkmaktadır. Konu insan

olunca alıřmalar daha ok askeri alanda gvenlik amalı olarak insansız hava araları ile yapılmaktadır. İnsan tespiti iin grntnn iřlenmesi, grntlerden z niteliklerin ıkarılması, ıkarılan z niteliklerin makine ğrenimi veya derin ğrenme ile eğitilmesi neticesinde bir sınıflama iřleminin yapılması ve eğitilen model ile grntlerin kıyaslanarak bir sonuca varılması basamaklarından oluřmaktadır.

Literatr dikkatle incelendiğinde, bu konu ile ilgili sınırlı sayıda alıřmanın yapıldığı ve alıřmaların yoğunluğ gnmze yaklařtıka arttığı grlecektir. Bu konudaki bazı nemli alıřmalar řu řekildedir;

Derinlik kamerası ile derin ğrenme tabanlı nesne tespit etmeye alıřmıştır. Kamkat meyvesinin otonom bir řekilde hasat edilmesi iin yolo algoritması kullanarak Kamkat meyvesinin toplama noktasını bulmaya alıřmış kullanmış olduğı algoritmalar arasında YOLOv7 modelinde %93 lk bir doğruluk oranı elde etmiştir. Bu alıřmasında yapraklar arkasında kalan meyvelerin tespitinin zorlařtığı yaprakların ve dalların bulunmadığı noktaların meyve toplama iin en uygun nokta olduğı tespiti yapılmıştır[4].

Tm ynl kameralarda insan tespiti ve takibi ile dikdrtgen battaniyesi problemi doktora tez alıřmasında otonom makinelerle insan tespiti ve yařlılara yardım edilmesi iin gerekli tespit ve analiz alıřmaları yapılmıştır. ok ynl kameralarla ile ekilmiş 46 videodaki grntlerde hareketle BOMNI ismini verdiğı bir veri kmesi zerinde bir Bayes erevesi geliřtirerek izleme ve dřme tespiti yapmış %93 izleme doğruluğna ulařmıştır[5].

Yayaların tespitinde yayaların arka grnmleri ve yayaların kendi grnmlerinden kaynaklı zorluklardan dolayı hızlı bir řekilde tespit edilemeyen řerinden hareketle NVIDIA CPU’da kullanarak YOLOv3 ve YOLOv3-tiny modelleri karřılařtırma yaparak YOLOv3 modelinde %91,07’lik doğruluk bařarısı elde ederek sunmuş olduğı modelin kalabalık ynetimine bir zm getirdiğini kanıtlamıştır[2].

Yksek yapılarda derin ğrenmeyi kullanarak gerek zamanlı beton hasar tespiti alıřmasında derin ğrenme metodu olan YOLOv3 algoritması kullanılarak gerek zamanlı hasar tespiti yapılmaya alıřılmış nerilen yaklařımda bir grnty 0,033 saniyede iřleyerek grntlerde %94,24 lk bir doğruluk oranı elde etmiştir[6].

Değışebilen paralı modellerde veri arttırımı modeli kullanarak arama kurtarma alıřmalarında insan tespiti alıřması tezinde istenilen verilere ulařmadaki zorluklar ve

verilere ulaşılsa dahi yeteri düzeyde verinin bulunmayışı yazarı yapay yollarla veri kümesini çoğaltmaya çalışıp elde ettiği sonuçlarda başarı oranını arttırmayı hedefine yöneltmiştir. Yapmış olduğu veri attırma yöntemlerinde veriyi çevirme, öteleme ve ölçeklendirme gibi, renk bazlı değişlerde aydınlık düzeyine müdahale, Gauss gürültüsü ekleme ve karşıtlık değerine müdahale etme gibi değişimler, etiket değerleri, piksel konumları gibi değişiklikler yapmıştır [7].

E. Şimşek ve arkadaşları geleneksel yöntemlerle foto kapan ile elde edilen görüntülerdeki karmaşıklık, arka plan hareketleri, ışık şiddeti, yetersizliği ve benzeri etkenlerden dolayı geleneksel görüntü tanımlı yöntemlerinin burada sınırlı kaldığını, buna alternatif olarak YOLO mimarisini kullanarak gerçek zamanlı nesne tespitinde daha hızlı ve yüksek doğruluk oranı yakalamayı hedeflemiştir. YOLO mimarisi ve PASCAL VOC 2012 veri seti ile 80 Sınıf için eğitilmiş ağırlıklar kullanılarak foto kapanın görüntü alanına giren insanların tespiti sağlanmıştır. Foto kapan veri seti içerisinde 100 adet farklı arka plana sahip hem gece hem gündüz çekilmiş 2.880 görüntü bulunmaktadır. Bu görüntülerden 2.150 tanesinde nesnelere ait tür, konum ve boyut bilgileri bulunmaktadır. 314 tanesinde ise sınırlayıcı kutuları belirlenmiş tam veya parçalı insan görüntüsü bulunmaktadır. 1.836 tanesinde ise insan bulunmamaktadır. 730 görüntüde ise hiç etiketlenmemiş olup bunları insan olup olmadığını tespit etme amacıyla kullanmıştır. Çalışmada güven parametresi 0,5 değeri. Eşitleme parametresi 0,3 değeri olarak kullanıldığında 314 insan içeren görüntüden 297 tanesi tespit edilmiş 17 tanesi tespit edilememiştir. Tespit edilemeyen görüntüler incelendiğinde kişilerin ağaç, nesne ve diğer bitki örtüleri ile kapandığı tespit edilmiştir. Bu verilere dayalı olarak %94,6 lık yüksek bir başarı oranı elde ettiklerini belirterek sonraki çalışmalarında kendi veri setlerini oluşturarak gömülü sistemlerde gerçek zamanlı nesne tespiti yapmayı hedeflemişlerdir[8][9]

Bir videodaki veya görseldeki nesneyi tespit etmek için ya görsel öznitelikler (şekil ve renk gibi) ya da nesneye ait hareket bilgileridir. Bu ilgilere ulaşmak belli bir zaman dilimini kapsar ve işlem gerektirmektedir. Görsellerden hareketle nesne tespiti yöntemini ve hareket bilgisinden faydalanıp nesne tespiti yöntemlerini ayrı ayrı kullanılabildiği gibi iki yöntemi bir arada kullanmakta mümkün olabilir[10].

DroMOD ismini verdikleri drone tabanlı çok kapsamlı nesne algılama çalışmalarında kişi, araç, yangın gibi nesneleri tespit etmek için ihtiyaca binaen hangi

nesne tespit edilmek isteniyorsa o nesneye ait referans verilerle karşılaştırma işlemi yapılarak sonuca ulaşma çalışmalarını simülasyon ortamında gerçekleştirmişlerdir[11].

Dronlarla küçük nesnelerin tespitinde yaşanan zorluklara çözüm getirmek amacıyla YOLOAL ismini verdikleri modele evrişim ve koordinat dikkat modülü (CCAM) ismini verdikleri yeni bir mekanizma dahil edilerek koordinat bilgisinden faydalanılarak nesneleri tespit etme yeteneğinin geliştiğine ve yapılan çalışma sonuçlarının YOLOv8 geride bıraktığını vurgulamıştır[12].

Trafikte araç yoğunluğunun olduğu yollarda trafikteki araç sayını insansız hava aracı ile tespit edip yer kontrol istasyonu ile paylaşan ve sayının belli bir sınırın üzerine çıkması durumunda sesli ve görüntülü ikaz veren çalışmalarında İHA üzerine monte edilen kamera ile görüntüler alınarak Watershed algoritması kullanılarak nesne sınırları tespit edilmiş, Gaussian modeline göre arka plan çıkarımı yapılarak MATLAB programında görüntüler gerçek zamanlı işlenerek araç sayısı tespit edilerek gerekli uyarılar verilmiştir[13].

Bu çalışmada, arama kurtarma operasyonlarının simülasyonu sırasında drone hava görüntülerinden insan tespiti için YOLOv5'in kullanılması önerilmiştir. Ayrıca mankenler insan yerine kullanılarak arama kurtarma sahasının her tarafına dağılmıştır. Toplanan hava görüntüleri, farklı büyütme seviyeleri ve drone uçuş modelleri açısından varyasyonlarla YOLOv5'in insan tespit etme etkinliğini kanıtlamak için kullanılmıştır[14].

F. H. K. Zaman ve arkadaşlarının yazmış olduğu bu makalede, açık alan olay yeri incelemelerinde İHA'ların gerçek zamanlı delil tespitine katkısını incelemeyi amaçlamaktadır. Olay yeri inceleme simülasyonu kullanılarak kanıt tespiti için İHA tabanlı ve geleneksel arama yöntemlerinin nicel karşılaştırmalı bir analizini sağlayarak projenin getirdiği yeniliği vurgulamaktadır. İHA'ların adli tespit araçları olarak kullanışlılığını, insanların mı yoksa insansız hava araçlarının mı kullanılması gerektiği ikilemini ele alarak test etmeye çalışılmıştır. Drone'ların performans sonuçlarını farklı arazilerde yeniden üretme kabiliyeti incelenirken, drone'lar tarafından kaydedilen videolar, kanıt tespitinin doğruluğunu test etmek için bilgisayarlı görü tekniklerine tabi tutulmuştur. Deneysel sonuçlar, kanıt tespitinde dronların kullanılmasının çeşitli arazi türlerinde daha fazla doğruluk ve algılama hızı sağlayabileceğini göstermektedir[15].

K. Jiang ve arkadaşlarının yazmış olduđu bu makalede, kenevir ördek sürülerinin popölasyonunu izlemeye yönelik yeni bir görüntü veri kümesi oluşturmak için derin öğrenme algoritmalarını kullandık. Deneysel sonuçların, CBAM-YOLOv7 algoritmasının daha yüksek doğruluk ve gelişmiş performans gösterdiği ortaya koymaktadır. Bu çalışma, akıllı bir kenevir ördek sayma yönteminin uygulanabilir olduğunu ve otomatik ördek sayımı geliştirebileceğini göstermektedir[16].

1.2. Tezin Amacı

Literatür çalışmalarına bakıldığında İHA ile insan tespit çalışmalarının yapıldığı görölmektedir. Yapılan çalışmalarda havadan çekilen görüntülerde insanların küçük görünmesinden dolayı insan tespitinde yaşanan zorluklar ön plana çıkmaktadır. Bu tespit göz önünde bulundurularak geniş bir insan veri kümesi oluşturulmuştur. Bu veri kümesi ön işlemeden geçirilmiştir. YOLOv7, YOLOv8 ve YOLOv10 algoritmaları ile eğitim işlemi gerçekleştirildi. Bu üç algoritmandan hangisinin daha iyi sonuçlar verdiği karşılaştırmalı olarak tezin ilerleyen kısımlarında verilecektir.

2. GENEL BİLGİLER

Bu bölümde, literatür genel bir şekilde incelenmiş ve çalışmada adı geçen teknolojiler hakkında genel bir bilgi sunulacaktır.

2.1. İnsansız Hava Araçları (İHA)

İnsansız Hava Araçları (İHA); içinde insan bulunmayan gündelik hayatta drone olarak tabir edilen ve uzaktan verilen direktiflere göre veya özerk bir şekilde hareket edebilen ve kullanım alanına göre çeşitli teknolojik ekipmanlarla donatılabilen bir çeşit uçaktır[17]. Literatür incelendiğinde İHA'ların birçok kullanım alanının olduğunu görmekteyiz. Belli başlı kullanım alanları şöyle sıralanabilir.

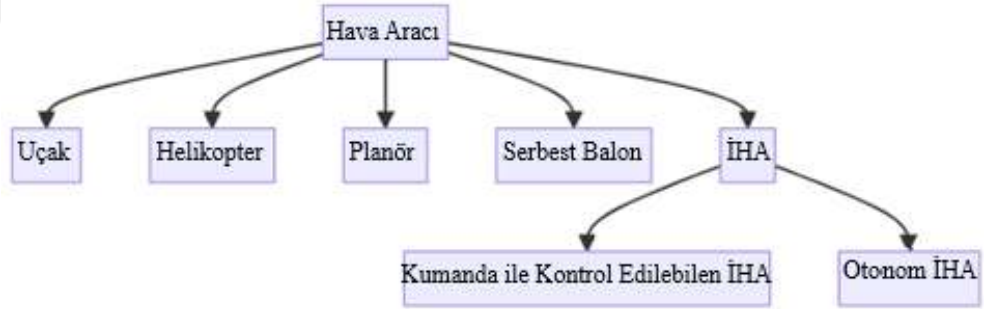
1. Askeri Uygulamalar
2. Endüstriyel Uygulamalar
3. Madencilik Uygulamaları
4. Hassas Tarım Uygulamaları
5. Arkeolojik Uygulamalar
6. Çevre ve Doğa Uygulamaları
7. Emniyet ve Güvenlik Uygulamaları
8. Doğal Afet Uygulamaları
9. Lojistik Uygulamaları
10. Ticari Uygulamalar
11. Hobi ve Spor Uygulamaları
12. İletişim Uygulamaları
13. Adli Uygulamalar
14. Harita Mühendisliği Uygulamaları[18].

Tez çalışmamda, İHA ile insan tespiti yapan bir sistem tasarlanmıştır. Şekil 2.1'de görülen dört motorlu döner kanatlı İHA, görüntüleri almak için bir kameradan ve görüntüleri işlemek için bir bilgisayardan oluşmaktadır.



Şekil 2.1. Sistem yapısı

İHA Çeşitleri



Şekil 2.2. Hava aracı kategorileri

İHA'lar kanat yapılarına göre ikiye ayrılmaktadır.

2.2. Sabit kanatlı İHA'lar



Şekil 2.3. Sabit kanatlı İHA'lar

Sabit kanatlı İHA'lar, genellikle tahrik sistemi olarak bir motor ve pervane kullanan ve çeşitli arazi şartlarında ve yerleşim yerleri dahil olmak üzere çok çeşitli olası çalışma ortamlarında çalışabilen kanat, gövde ve kuyruktan oluşan İHA'lardır. Sabit kanatlı İHA'ların dikey kalkış ve inişleri olmaması ve iniş ve kalkışta belli bir piste ihtiyaç duymalarına rağmen en çok kullanım alanına sahip İHA türlerindendir. Havada süzülerek yol almaları daha fazla menzil gitmelerine sebep olduğundan geniş bir kullanım alanı bulmaktadır.

2.3. Döner kanatlı İHA'lar



Şekil 2.4. Döner kanatlı İHA'lar

İsmi döner kanat yapılarından alan ve helikoptere benzerliği ile tanınan fakat helikopterden farklı yönleri olan bu dronlar zemine paralel dönen pervaneleri sayesinde havayı iten döner kanatlı İHA'lar, dikey bir şekilde yükselip alçalabilmektedir. Dar alanlarda dikey iniş ve kalkış yapabilen ve havada sabit kalabilme özelliklerinden dolayı kendilerine oldukça fazla bir kullanım alanı bulmaktadır.

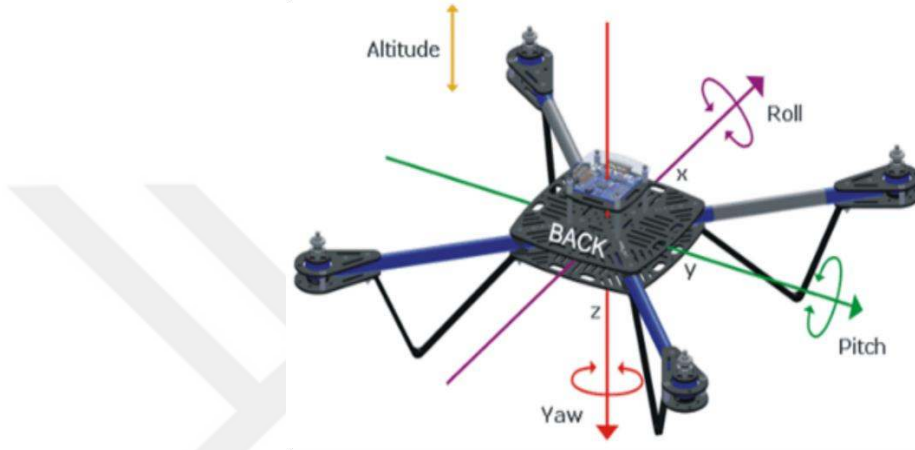
Farklı gövdeleri ve motor sayıları dikkate alındığında döner kanatlı İHA'lar;

1. Tricopter
2. Hex-Tricopter
3. Quadcopter Plus
4. Quadcopter X
5. Qcto Quadcopter Plus
6. Qcto Quadcopter X
7. Hexacopter X
8. Hexacopter Plus
9. Hexacopter H
10. Octocopter X
11. Octocopter Plus
12. Octocopter H[19].

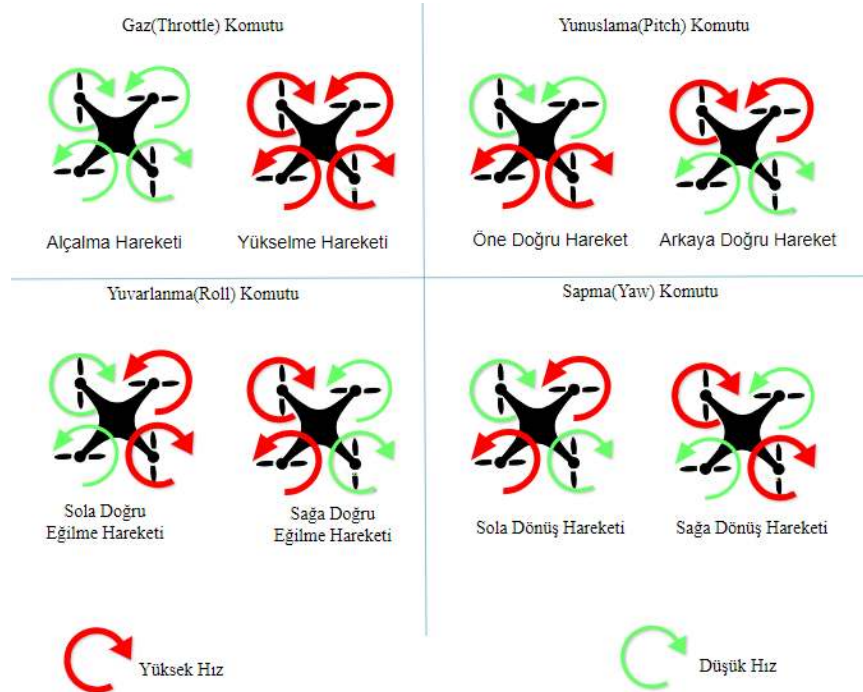
Şeklinde sıralanabilir.

2.4. Dört Rotorlu İnsansız Hava Aracı

Kullanım alanı oldukça geniş olan dört rotorlu insansız hava araçları askeri uygulamalar başta olmak üzere tarımsal faaliyetlerde, arama kurtarma faaliyetlerinde ve birçok alanda nesne tespiti amacıyla kendine kullanım alanı bulmaktadır[20]. Quadcopterlar genellikle kararlı uçuş için bir dizi sensör kullanır, bu sensörler arasında jiroskoplar, ivmeölçerler ve barometreler bulunabilir. Ayrıca, GPS gibi konum belirleme sistemlerini kullanarak da uçuşlarını kontrol ederler.



Şekil 2.5. Drone hareket sistemi[21]



Şekil 2.6. Quadcopter hareketleri[22]

Dört motorlu bir insansız hava araçları incelendiğinde genelde şu parçalardan oluştukları görülmektedir:

Gövde (Airframe): İHA'nın ana yapısal bileşeni olmakla birlikte İHA'ların uçuş süresini, manevra kabiliyetini ve performansını direk etkilediğinden genellikle hafif malzemelerden yapılmakta ve uçuş stabilitesini sağlamak için aerodinamik bir tasarıma sahiptir. Drone gövdelerinin tasarımı kullanım amacına göre değişmektedir. Yarış dronu olarak kullanılacaksa hafif malzemeden yapılması ve aerodinamik bir tasarım ön plana çıkarken yük taşıma amaçlı dronlarda ekipmanlar göz önünde bulundurularak tasarım yapılmaktadır.

Motorlar: Dört adet motor, UAV'nin havada kalmasını sağlar. Bu motorlar genellikle elektrikli motorlardır. Motor seçiminde İHA'nın hangi amaçla kullanılacağı, gövde tipi ve ağırlığı, ihtiyaç olan itme kuvveti, motorun ağırlığı ve fiyat gibi etkenler göz önünde bulundurulur.

Pervaneler: Her motor için bir pervane bulunur. Pervaneler, motordan gelen gücü dönme hareketine çevirerek itme kuvveti üretir ve İHA'nın ileri doğru hareket etmesini sağlar. Kullanım amacına göre seçilmesi gereken pervanelerde plastik pervaneler daha hafif ve sert malzemeden yapılmış olan pervanelere göre kaldırma kuvveti daha fazladır. Burada İHA'nın bir nesneye çapması durumu da göz önünde bulundurularak karbon fiber malzemeden yapılmış olan pervaneler günümüzde sıklıkla kullanılmaktadır. Pervanelerin dönüş yönleri birçok insansız hava aracında saat yönü ve tersi olacak şekildedir.

Uçuş Kontrol Sistemi (Flight Control System): İHA'nın uçuşunu kontrol eden ana bilgisayar sistemidir. Bu sistem, uçuş stabilitesini sağlamak, rotayı belirlemek ve otomatik pilotaj gibi görevleri yerine getirir. İHA'nın uçuşunu kontrol eden beyin olarak da adlandırılabilir. Hız, yön ve diğer parametreleri kontrol eder.

Navigasyon ve İletişim Sistemleri: İHA'nın konumunu belirlemek ve diğer sistemlerle iletişim kurmak için kullanılan sistemlerdir. Bu sistemler genellikle GPS ve radyo iletişimi gibi teknolojileri içerir.

Yedek Güç Kaynağı: Acil durumlarda veya ana güç kaynağının başarısız olduğu durumlarda kullanılan bir yedek güç kaynağı bulunabilir.

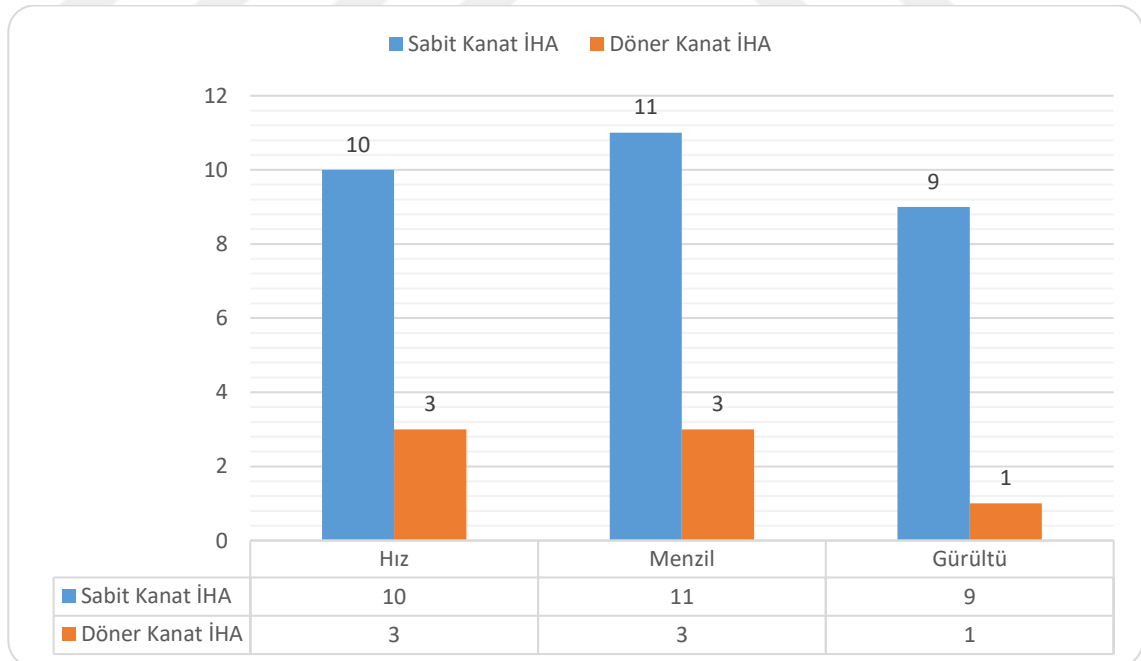
Sensörler: Çevresel koşulları ve uçuş parametrelerini ölçmek için kullanılan sensörler bulunur. Bu sensörler genellikle irtifa, hız, sıcaklık, basınç ve diğer değişkenleri izler.

Kamera veya Diğer Yükler: İHA genellikle bir kamera veya diğer sensörlerle donatılır. Bu sensörler, gözetleme, keşif veya haritalama gibi çeşitli görevler için kullanılabilir. İHA'nın çevresini gözlemlemesini ve veri toplamasını sağlayan kameralar ve diğer sensörlerdir. Bu sensörler genellikle GPS, barometre, hava basıncı sensörleri vb. olabilir.

Batarya: İHA'nın gücünü sağlayan bataryadır. LiPo (Lityum Polimer) bataryalar sıkça kullanılır, çünkü hafif olmalarına rağmen yüksek enerji yoğunluğuna sahiptirler.

Sabit ve döner kanatlı İHA'lar hız, menzil ve gürültü bakımından birbiriyle karşılaştırılması Çizelge 2.1'de sunulmuştur.

Çizelge 2.1. Sabit kanat İHA ile döner kanat İHA karşılaştırılması[23]



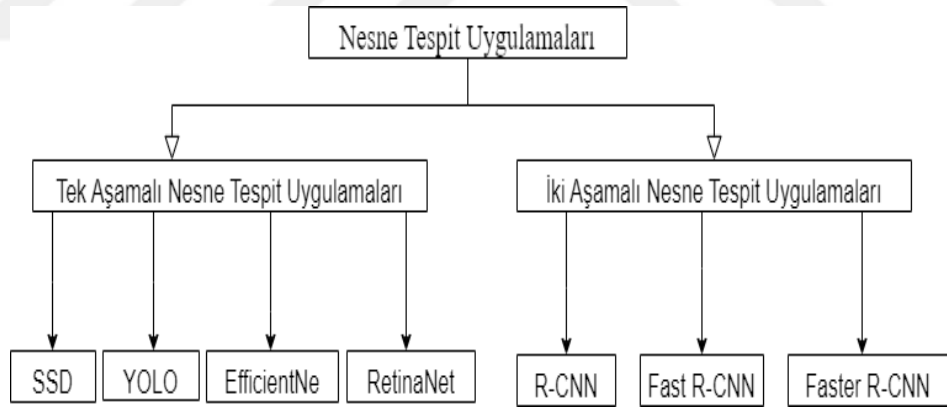
Sabit kanat İHA ve döner kanat İHA hız menzil ve gürültü bakımından karşılaştırıldığında Çizelge 2.1'de görüldüğü gibi sabit kanat İHA'ların hız ve menzil bakımından döner kanat İHA'lara göre daha avantajlı oldukları fakat gürültü seviyesi

bakımından döner kanat İHA'ların avantajlı oldukları görülmektedir[23]. İHA'ların birçok avantajı olmasına rağmen dezavantajları da bulunmaktadır. Sınırlı taşıma kapasitesi, kısa uçuş süresi, rüzgârlı havalarda sınırlı kullanım ve iniş, kalkış ve uçuş aşamalarındaki sorunlar nedeniyle geniş alan uygulamaları için uygun görülmemektedir.

2.5. İnsan Tespiti

İnsanlar genellikle fiziki ve biyolojik yapıları diğer nesnelerden farklı olduğundan dronlara takılacak çeşitli kameralar vasıtasıyla insan tespit edilebilir. Gerçek zamanlı insan tespitinde insanların bir arada bulunmasından dolayı üst üste çakışmalardan, hava koşulları, kamera parlaması gibi nedenlerden dolayı insan tespitinde zorluklar yaşanmaktadır[2].

Çeşitli algoritmalar kullanılarak insansız hava araçlarından alınan görüntülerle nesne ve insan tespiti yapılabilir. Derin öğrenme algoritmaları, yapay sinir ağları üzerine kurulu, karmaşık veri yapılarını analiz etmek, nesneleri tanımak ve kararlar almak için kullanılan özel makine öğrenimi algoritmalarıdır. En popüler derin öğrenme algoritmaları tek aşamalı nesne tespiti ve iki aşamalı nesne tespiti olarak ayırabiliriz. Bu algoritmalarından en çok kullanılanlar Şekil 2.7'de gösterilmiştir[24].



Şekil 2.7. Nesne tespit uygulamaları[24].

2.6. Görüntü işleme

Görüntü işleme genellikle bir resim, fotoğrafı anlama, yorumlama ve analiz çalışmalarında kullanılan bir yöntem olarak tanımlanabilir[25].

2.6.1. Filtreleme

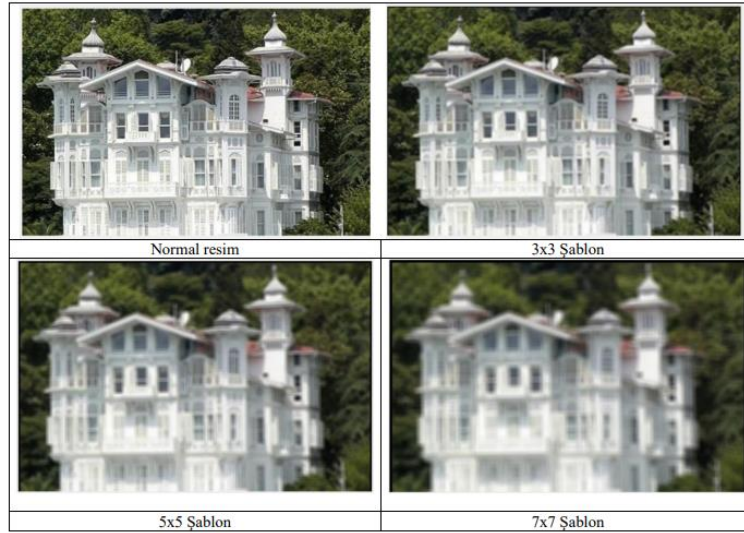
Görüntü işleme filtreleme, bir görüntü üzerinde belirli bir etkiyi elde etmek için kullanılan işlemlerdir. Bu filtreler genellikle görüntü üzerinde belirli bir etkiyi vurgulamak, gürültüyü azaltmak veya görüntüyü iyileştirmek için uygulanır.

2.6.1.1. Ortalama Filtre (Mean Filter - Box Blur)

Gürültüyü azaltmak için her pikselin komşularının ortalama değeri alınarak görüntüyü yumuşatmada kullanılır. Bu işlem, çevrelerini temsil etmeyen piksel değerlerinin yumuşatılmasına yol açar[26]. Ortalama filtre, bir tür evrişim filtresidir[27].

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

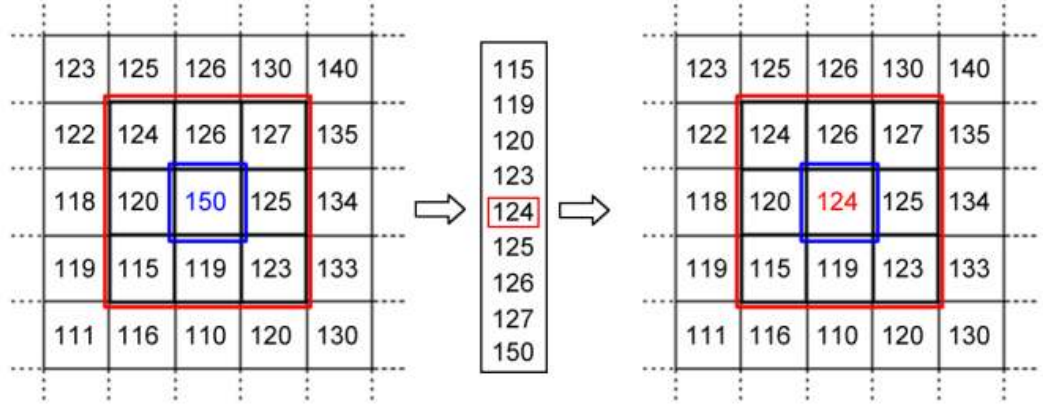
Şekil 2.8. Mean (Ortalama)



Şekil 2.9. Ortalama filtre örneği

2.6.1.2. Medyan Filtresi (Median Filter)

Medyan Filtre, görüntülerdeki gürültüyü yumuşatmak için görüntü işlemede kullanılan bir filtreleme yöntemidir[26]. Görüntüyü yumuşatan ortalama filtrenin aksine, medyan filtre ayrıntıların kaybolmamasını sağlar. Her pikselin komşularının değerleri sıralanır ve değerlerdeki aşırı sıçramaları ortadan kaldırmak amacıyla medyan değer alınır. Kenarların belirlenmesine ve ayrıntıların korunmasına yardımcı olduğu için görüntülerdeki gürültüyü azaltmak için tercih edilir[27].



Şekil 2.10. Medyan filtresi

Şekil 2.10'da 150 değeriyle temsil edilen pikselin etrafındaki işlemler, çevreleyen pikselleri düzgün şekilde yansıtmaz. Bu pikselin değerini değiştirirken öncelikle çevredeki piksellerin değerlerini sırayla düzenleyelim. Bunlar (115, 119, 120, 123, 124, 125, 126, 127, 150) değerlerinden oluşur. Bu değerlerin ortasında ise 124 sayısı yer almaktadır. Bu nedenle, 150 sayısı 124 sayısı ile değiştirilir. Burada 124 sayısı medyan değer olur[27].

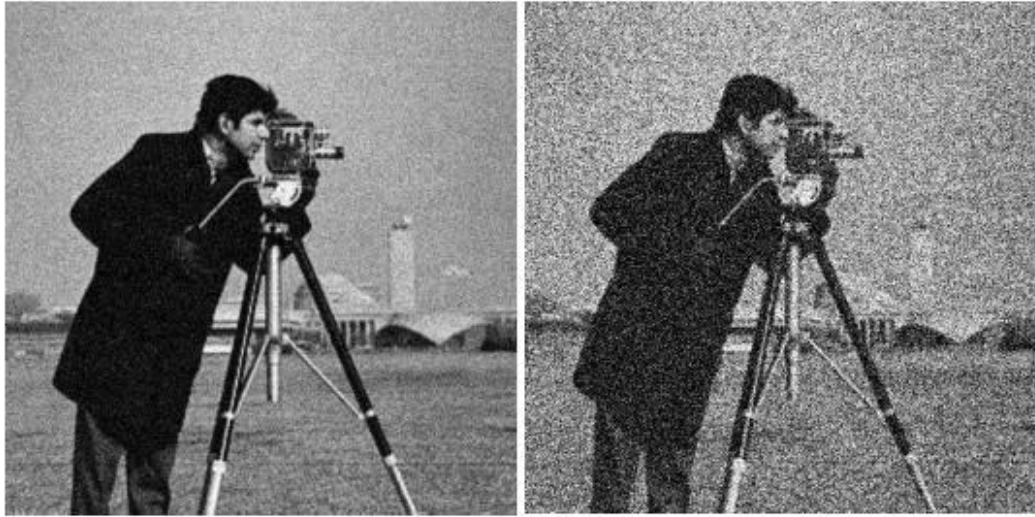


Şekil 2.11. Medyan filtresi örneği

2.6.1.3. Gauss Düzleştirme Filtresi (Gaussian Smoothing)

Gauss filtresi, görüntü işleme alanında yaygın olarak kullanılan bir filtreleme yöntemidir[28]. Bu yöntemin amacı, görüntüdeki gürültüyü azaltmak ve görüntüyü yumuşatmaktır. İşleyişini matematiksel olarak ifade edecek olursak: Her pikselin komşularının değerleri ağırlıklarla hesaplanır ve piksel değeri bu ağırlıklı ortalama ile güncellenir. Komşuluk bilgisi Gauss dağılımına göre belirlenir[28]. Gauss filtresi, görüntülerdeki keskin ton değişikliklerini azaltmada ve pürüzsüz bir görüntü elde etmede etkilidir. Bu filtre özellikle görüntülerdeki gürültüyü azaltırken ve daha homojen

bir görüntü elde edilmek istendiğinde kullanılır. Kenarları tanımlamak ve ayrıntıları korumak için tasarlanmıştır.



Şekil 2.12. Gauss filtresi[28]

2.6.2. Kenar Tespiti

Kenar tespiti, bir görüntüdeki cisimlerin veya nesnelerin çerçevesini belirlemeyi amaçlayan bir görüntü işleme tekniğidir. Kenar tespiti, bir görüntüyü kolay analiz edebilmek için önemli olan özellikleri belirlemeyi amaçlar. Kenarlar genellikle nesnelerin şeklini ve yapısını tanımlayan önemli bilgiler içerir, bu nedenle kenar tespiti, nesne tanıma, görüntü sınıflandırma ve benzeri uygulamalarda önemli bir adımdır.

2.6.3. Hough Dönüşümü

Hough dönüşümü iki boyutlu düzlemdeki bir noktanın parametre uzayına dönüşümü olarak tanımlanmaktadır. Hough Dönüşümü genellikle düz çizgilerin veya dairelerin algılanmasında kullanılır[29].

Bir doğru için genel denklem:

$$y = mx + b \quad (2.1)$$

Hough Dönüşümünde doğruların gösterimin de kullanılan denklem[29].

$$\rho = x \cos(\theta) + y \sin(\theta) \quad (2.2)$$

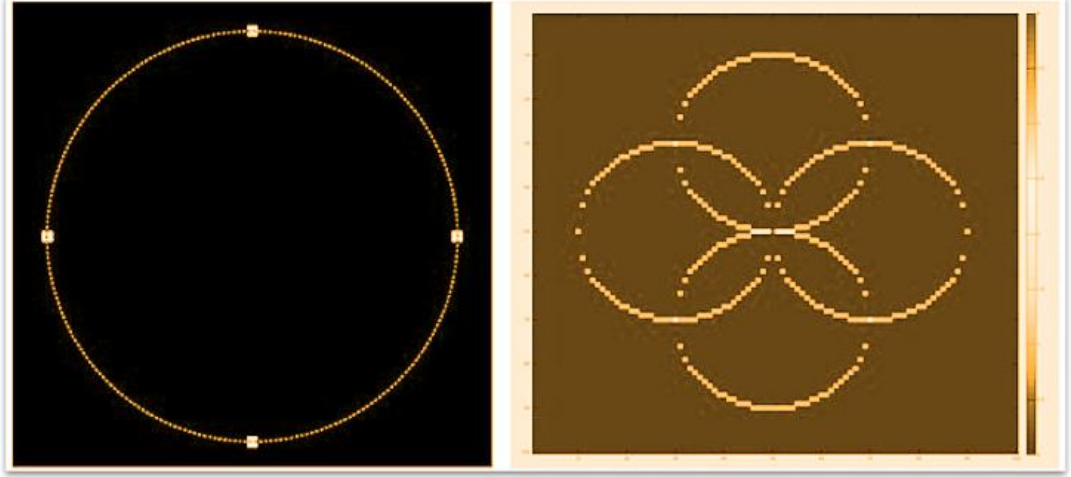
Bir daire için genel denklem:

$$(x - a)^2 + (y - b)^2 = r^2 \quad (2.3)$$

Burada, (a,b) dairenin merkezi ve r ise dairenin yarıçapıdır.

$$x = a + r \cos \theta \quad (2.4)$$

$$y = b + r \sin \theta \quad (2.5)$$



Şekil 2.13. Hough dönüşümünde yarı çap tespiti[30]

Hough Dönüşümünde önemli noktalardan biri tespit edilmeye çalışılan dairesel nesnenin yarı çapının yaklaşık olarak tahmin edilebilmesidir[30]

2.6.4. Nesne Tanıma ve Sınıflandırma

Nesne tanıma ve sınıflandırma, bilgisayar görüşü alanında yaygın olarak kullanılan tekniklerdir. Bu teknikler, görüntülerdeki nesneleri algılamak, tanımak ve farklı sınıflara ayırmak için kullanılır. İşte nesne tanıma ve sınıflandırma için kullanılan bazı algoritmalar:

YOLO (You Only Look Once): YOLO, gerçek zamanlı nesne tespiti için popüler bir algoritmadır. Görüntüyü bir matris olarak ele alır ve tek bir geçişte nesneleri sınıflandırır ve konumlarını belirler. YOLO (You Only Look Once) çerçevesi, farklı nesne algılama algoritmaları arasında görüntülerdeki nesnelerin hızlı ve güvenilir bir şekilde tanımlanmasına olanak tanıyan olağanüstü hız ve doğruluk dengesi ile öne çıkıyor [31].

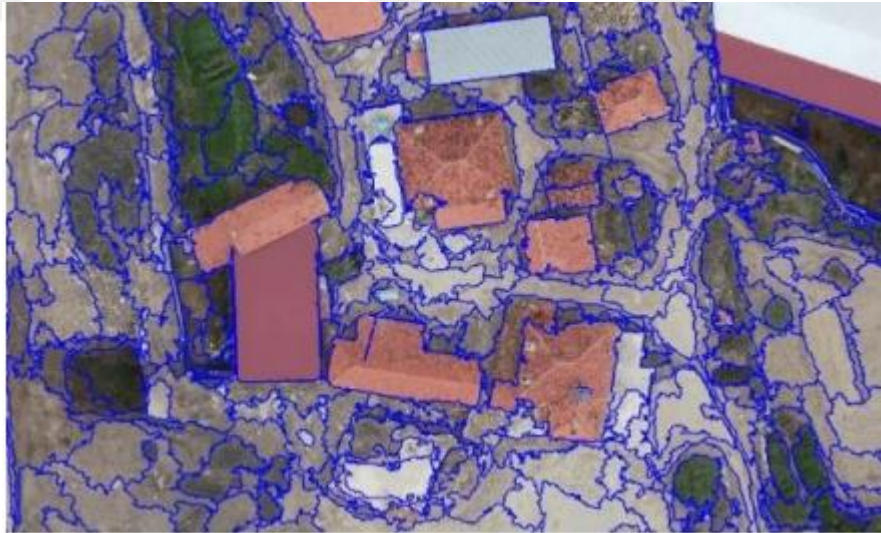
Faster R-CNN (Region-based Convolutional Neural Network): Faster R-CNN, nesne tespiti için bir başka etkili yöntemdir. RPN (Bölgesel öneri ağı) kullanarak nesne bölgelerini önerir ve ardından bu bölgeleri sınıflandırır.

SSD (Single Shot MultiBox Detector): SSD, çoklu ölçekli öneri bölgeleri kullanarak nesne tespiti yapar. Hem hızlı hem de doğru sonuçlar verir.

Mask R-CNN: Mask R-CNN, nesne tespiti ve piksel düzeyinde sınıflandırma yapabilen bir modeldir. Nesnelerin sınıflandırılmasının yanı sıra, nesne maskelerini de çıkarır.

2.6.5. Segmentasyon

Segmentasyon, bir görüntüdeki nesneleri veya bölgeleri piksel düzeyinde ayırma işlemidir. Piksellere verilmiş olan etiketler üzerinden tahminde bulunma işlemi yapılır[32]. Görüntüdeki bilgileri daha anlamlı hale getirerek nesneleri veya bölgeleri tanımlar. Bu şekilde istenmeyen bilgileri veya arka planı filtreleyerek, sadece ilgi alanlarına odaklanmayı sağlar. Buradaki amaç, görüntüdeki farklı nesne veya bölgeleri belirlemek ve ayrı olarak kullanmaktır. Tıpta hastalıklı dokuların ve organların teşhisi ve analizi için[33], sürücüsüz araçlar için çevresindeki nesneleri algılamada, güvenlik gerektiren yerlerde ve gözetim sistemlerinde insanları ve nesneleri tespit etmek için kullanılır.



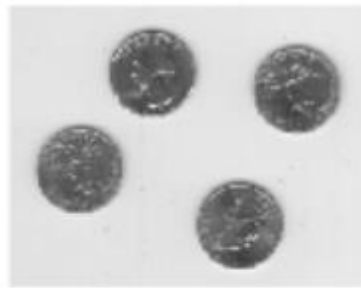
Şekil 2.14. Segmentlere ayrılmış nesneler örnek görsel[34].

2.6.6. Özellik Çıkarma

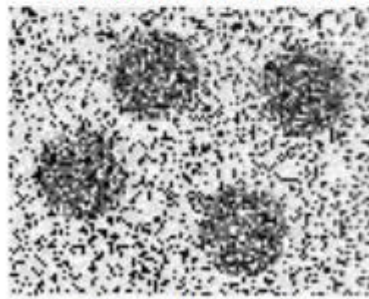
Özellik çıkarma, bir görüntünün çizgiler, köşeler veya dokular gibi önemli özelliklerinin tanımlanmasını ve çıkarılmasını içerir. Bu işlem, görüntülerdeki önemli bilgileri tanımlamak ve kullanılabilir hale getirmek için görüntü işleme ve makine öğrenimi uygulamalarında kullanılır.

2.6.7. Tuz – Biber kirliliğinin kaldırılması

Tuz-biber gürültüsü, genellikle düşük kaliteli sensörlere veya veri aktarımındaki sorunlara bağlı olarak ortaya çıkabilir. Tuz ve biber gürültüsü, görüntülerde rastgele beyaz ve siyah noktaların olduğu bir tür gürültüdür. Bu noktalar, beyaz arka plan üzerinde siyah pikseller ve siyah arka plan üzerinde beyaz pikseller şeklinde görünür. Siyah noktalar genellikle 0 değerini alırken, beyaz noktalar 255 değerini alır. Bu tür gürültü, görüntü işleme algoritmalarında düzeltilmesi gereken yaygın bir sorundur[26]. Burada tüm piksellerde gürültü oluşmayabilir ama oluşan gürültüler ise görüntü işleme esnasında olumsuz sonuçları doğuracağından bunların yok edilmesi gerekmektedir[35]. Gürültü, görüntü işleme uygulamalarında, özellikle nesne tanıma veya görüntü iyileştirme gibi alanlarda sorunlara neden olabilir. Bu nedenle, tuz-biber gürültüsünü azaltmak veya kaldırmak için çeşitli filtreleme teknikleri kullanılır[36].



Orjinal Görüntü



Gürültülü Görüntü

Şekil 2.15. Tuz ve biber gürültüsü eklenmiş görüntü üzerinde filtrelerin etkileri [35]

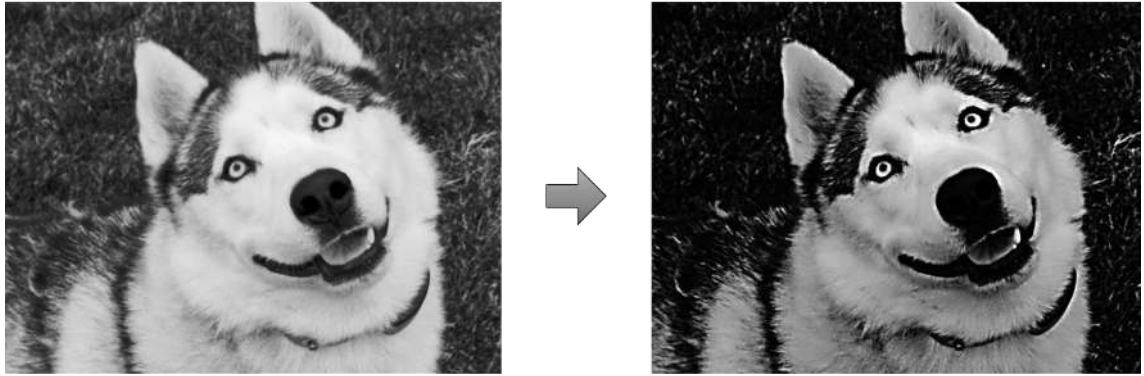
2.6.8. Keskinliği arttırma

Kullanılacak bir görüntüden amacımızın doğru bilgiye ulaşmak veya en iyi yorumu yapabilmemizi sağlayacak olması görüntüler üzerinde keskinliği arttırma algoritmaları

bu amaca hizmet eder. Uzaktan alınan görüntüler üzerinde keskinleştirme işlemi uygulamaları sıklıkla başvurulmuş algoritmalarındandır[37]

2.6.8.1. Unsharp Masking (Keskinleştirme)

Unsharp Masking, bir görüntüden orijinal görüntüyü çıkardıktan sonra elde edilen bir maske ile görüntüyü keskinleştirir. Bu işlem, görüntüdeki kenarları ve detayları vurgulayarak keskinliği artırır. Bu süreçte, orijinal görüntü taban katmanı olarak düşünülebilirken, elde edilen maske ise detay katmanını temsil eder. Unsharp Masking yöntemi taban ve detay katmanlarını kullanarak görüntü keskinleştirme işlemi yapar[38]



Şekil 2.16. Unsharp masking örneği[39].

2.6.8.2.Laplacian Kenar Algılama

Laplacian filtresi, görüntüdeki kenarları tespit eder ve bu kenarları belirginleştirir. Bu, görüntünün daha keskin görünmesini sağlar.

2.6.8.3. Kontrast Artırma

Kontrast artırma, görüntüdeki farklı tonların arasındaki farkı artırarak keskinlik etkisini artırabilir.

2.6.8.4. Kenar Belirginleştirme Filtreleri

Kenar belirginleştirme filtreleri, görüntüdeki kenarları vurgular ve bu da görüntünün daha keskin görünmesini sağlar. Örnek olarak, Sobel veya Prewitt filtreleri kullanılabilir.

2.6.8.5. Wavelet Dönüşümü

Wavelet dönüşümü, görüntüdeki farklı ölçeklerdeki detayları ortaya çıkararak keskinliği artırabilir. Bu yöntem genellikle çoklu ölçekli kenar belirginleştirme için kullanılır.

2.6.8.6. Histogram Eşitleme

Histogram eşitleme, görüntünün histogramını değiştirerek kontrastı artırır ve detayları belirginleştirir. Ancak aşırı uygulandığında görüntüdeki gürültüyü artırabilir[38].

2.6.9. Kenar çıkarma algoritması

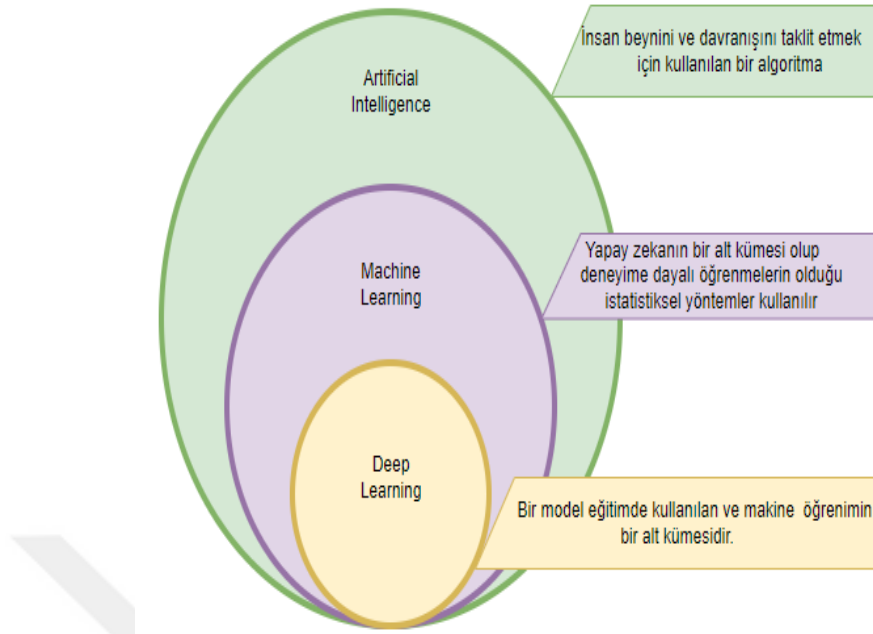
Kenar algılama, doğal görüntülerden görsel olarak önemli kenarları ve nesne sınırlarını çıkarmak için kullanılan bir teknolojidir. Bu, bilgisayarlı görmenin temel zorluklarından biri olarak uzun yıllardır kabul edilmektedir. Görüntü üzerinde kenar tespiti yapmak, nesneleri tespit etmek, saymak ve özelliklerini belirlemek için kullanılabilir. Kenar belirleme algoritmaları, temel olarak görüntüdeki piksellerin renk değerlerinin birbirlerinden farklılaşması ile kenarları belirler[40].

2.7. Isı haritası

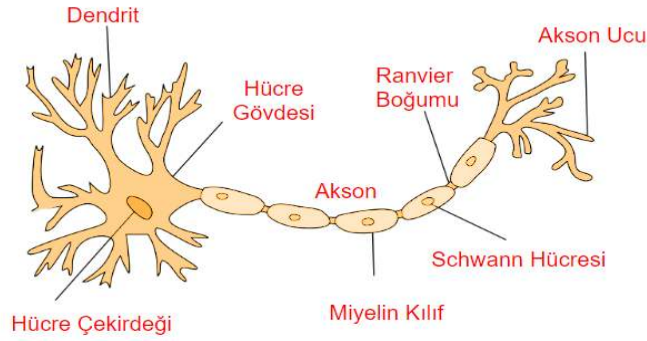
Termal kamera veya termal sensörler vasıtasıyla bir bölgedeki insan olup olmadığı varsa insanların hareketlerini tespit etmeye yarar. Görüntülerden yüksek düzeyde özellikleri çıkarmak suretiyle kullanılabilecek yöntemlerden biri makine öğreniminin bir alt kümesi olan derin öğrenme yöntemidir[41]. Termal kameralardaki düşük çözünürlüklü görüntülerden yüksek çözünürlüklü görüntüleri derin öğrenme modeli ile eğitilerek belli bir bölgedeki insan varlığı tespit edilebilir[42].

2.8. Derin Öğrenme

İnsan beyni gibi hesaplama yapmaya çalışan resim, ses, metin gibi karmaşık verilerden anlam çıkarmaya çalışan yapay zekâ alt alanıdır[43].



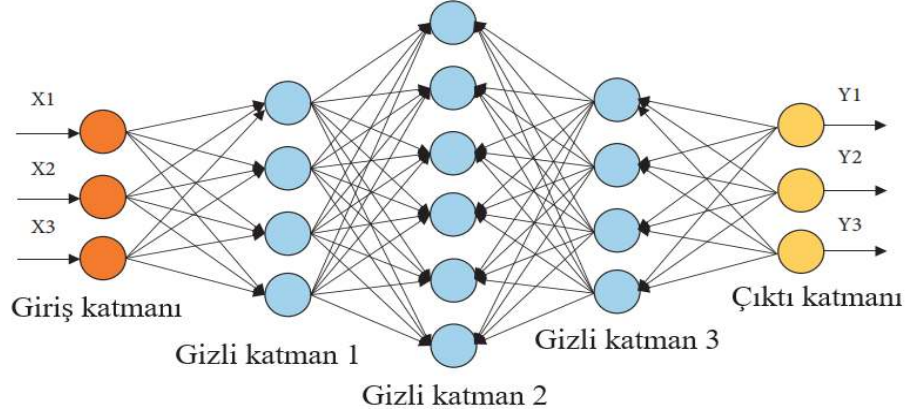
Şekil 2.17. Derin Öğrenme[44]



Şekil 2.18. Sinir Hücresi Yapısı[45]

Çevreden gelen kimyasal ve elektriksel sinyaller veya diğer hücrelerden gelen sinyalleri Dendrit vasıtasıyla hücre gövdesinde işlenmek üzere iletilir. Burada işlenen sinyallerin belli bir değer üzerine ulaşması neticesinde akson boyunca diğer hücrelere iletilerek bu durum sonunda bir öğrenme gerçekleşmektedir. Bu öğrenme neticesinde sinir hücreleri arasındaki bağlantılar şekillenir hatta bu bağlantılar gelişim sürecine bağlı olarak yeni bağlantıları doğurur[46].

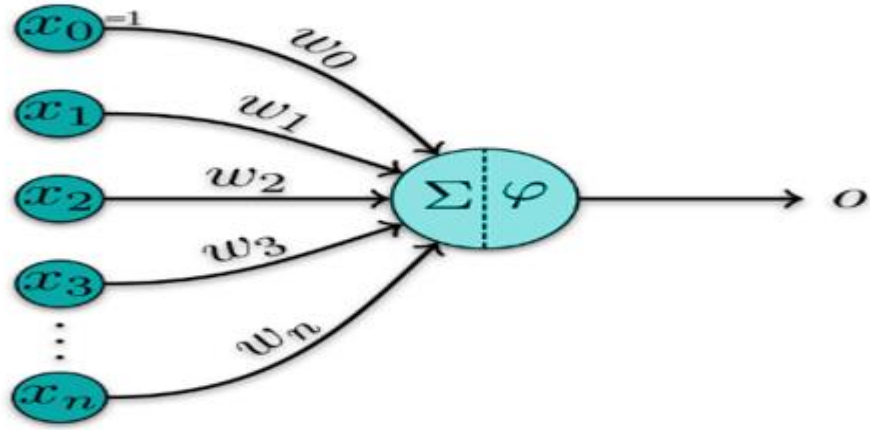
Yapay sinir ağları; insanların doğayı taklit etmeleri neticesinde geliştirmiş oldukları birçok nesne gibi kendi biyolojik sinir hücrelerini de taklit ederek bunu bir bilgisayar programı haline getirmeleri neticesinde oluşturulmuştur[46].



Şekil 2.19. Yapay sinir yapısının basit modellemesi[47]

Genel olarak giriş katmanı, gizli katmanlar ve çıkış katmanı olmak üzere üç katmandan oluşmaktadır. Yapay sinir ağlarında giriş katmanında bulunan veriler belli bir ağırlık katsayıları ile çarpılarak çıkışta bir veri üretmektedir. Bu işlem, sinir ağının öğrenme sürecindeki temel adımlardan biridir[48].

$$y = wx + b \quad (2.6)$$



Şekil 2.20. Perceptron (Algılayıcı)

2.9. Derin Öğrenme Modelleri

Joseph Redmon ve Ali Farhadi tarafından bulunan YOLO (You Only Look Once) algoritması nesne tespitine yeni bir yaklaşım getirmektedir. YOLO'nun günümüzde popüler hale gelmesini sağlayan temel özellikler şöyle sıralanabilir:

1. Gerçek zamanlı nesne tespiti yapılabilmesi
2. Nesne tespiti yaparken hızlı sonuçlar üretmesi

3. Tek bir ağ mimarisinin olması ile basit bir yapıya sahip olması
4. Küçük nesneleri algılamada diğer yöntemlere göre daha başarılı sonuçlar vermesi
5. Birden fazla nesneyi aynı anda algılayabilmesi
6. Farklı boyutlardaki nesneleri algılayabilmesi
7. Açık kaynak kodlu olması[49].

YOLO algoritmasının farklı versiyonları olmakla beraber son çıkan versiyon genellikle bir önceki versiyonların geliştirilmesi ve iyileştirilmesi neticesinde ortaya çıkmaktadır. YOLO, nesne tespiti için kullanılan günümüzde hızla yayılan bir derin öğrenme algoritmasıdır. YOLO'nun farklı versiyonları olmakla beraber her versiyon değişikliğinde iyileştirmeler yapılarak performanslarını arttırmayı hedefler. Kullanacağımız projedeki ihtiyaçlar hangi versiyonu kullanacağımızı belirleyecektir.

2.10. YOLOv7

Genel olarak YOLOv7, YOLO'nun önceki sürümlerine göre önemli bir gelişmedir ve nesne algılama görevlerinde daha iyi performans ve verimlilik sunar. YOLOv7 kendinden önceki YOLO modellerine göre daha doğru sonuçlar daha hızlı bir şekilde vermesiyle ön plana çıkmakta ayrıca diğer derin öğrenme algoritmalarından donanım olarak daha az maliyetlidir[50] Yolov7 modeli, Yolov5 ile entegre edilmiş bir ön işleme yöntemine sahiptir ve mozaik veri artırma yöntemi küçük nesnelerin tespiti için uygun bir kullanıma sahiptir[16].

2.10.1. Input Size (Giriş Boyutu)

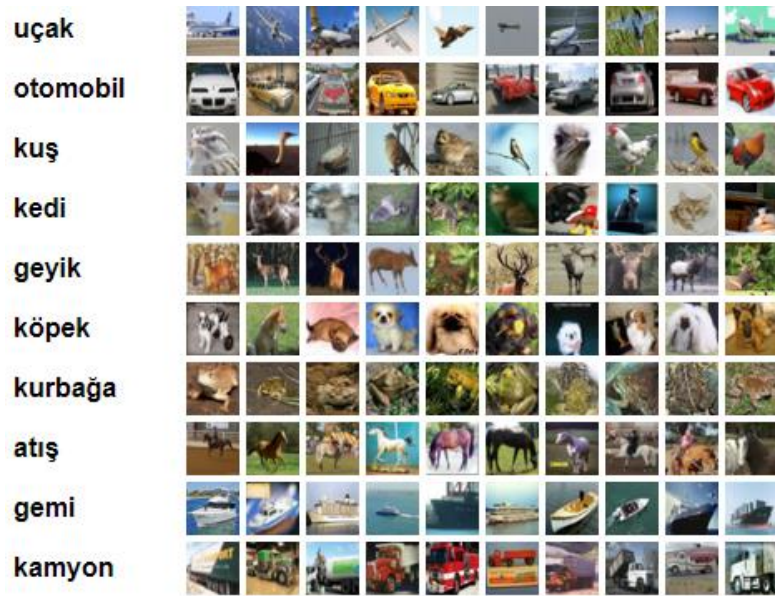
Modelin işleyeceği görüntülerin boyutunu belirler. YOLOv7'de giriş boyutu genellikle bir çift olarak belirtilir, örneğin "416x416" veya "608x608" gibi. Bu boyutlar, genellikle modelin performansı ve hassasiyeti arasında bir denge kurmak için seçilir. Daha büyük boyutlar genellikle daha detaylı sonuçlar sağlarken, daha küçük boyutlar daha hızlı işleme süreleri sunabilir. Giriş boyutu aynı zamanda modelin hiper parametrelerinden biridir ve genellikle modelin başarımını etkiler. Bu nedenle, giriş boyutunu seçerken dikkatli olmak önemlidir ve uygulamanın gereksinimlerine uygun bir boyut seçmek önemlidir.

2.10.2. Batch Size (Toplu Boyut)

Eğitim ve tahmin için kullanılacak veri örneklerinin toplu halde verilip verilmeyeceğini belirler. Batch Size eğitim sırasında sinir ağının ne kadar hızlı yakınsadığını belirlediği için önemli bir parametredir[51]. Yığın boyutları, CPU'nun bellek kapasitesine uygun olarak genellikle 2'nin katları şeklinde belirlenir. Küçük yığın boyutları, modelin daha hızlı yakınsamasını sağlar; ancak bu süreçte daha fazla gürültü içerebilir. Öte yandan, daha büyük yığın boyutları, yakınsama sürecini yavaşlatır, fakat daha az gürültü ile daha kararlı bir öğrenme sunar[52].

2.10.3. Number of Classes (Sınıf Sayısı)

Algılanabilecek farklı nesne sınıflarının sayısını belirtir. Örneğin, bir nesne algılama modeli, arabalar, insanlar, köpekler gibi farklı nesne sınıflarını tanıyabilir. Örneğin CIFAR-10 veri seti, sınıf başına 6000 resim olmak üzere 10 sınıfta 60000 adet 32x32 renkli resimden oluşur. 50000 eğitim resmi ve 10000 test resmi vardır.



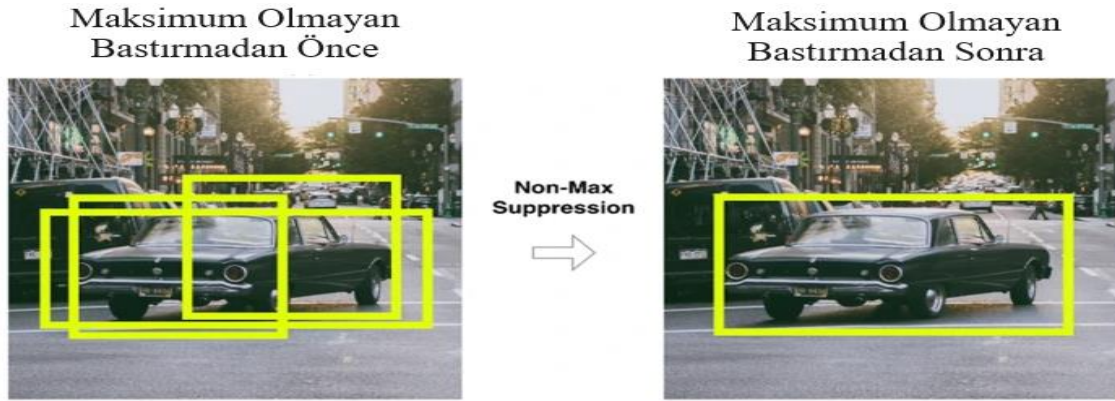
Şekil 2.21. CIFAR-10 veri setinden örnek[53]

2.10.4. Confidence Threshold (Güven Eşiği)

Algılama sonuçlarının ne kadar güvenilir olması gerektiğini belirler. Bu değer, algılama sonuçlarının ne kadar doğru olması gerektiğini kontrol eder. Örneğin bir nesne tanıma modelinde güven eşiği %70 olarak ayarlanırsa yalnızca modelin %70 veya daha yüksek güvenle tahmin ettiği sonuçlar kabul edilir.

2.10.5. NMS Threshold (NMS Eşiğı)

NMS (Non-Maximum Suppression), nesne algılama gibi bilgisayarla görme görevlerinde sahte veya ilgisiz sonuçları ortadan kaldırmak için yaygın olarak kullanılan bir tekniktir. İşleyiş prensibi, bir bağlama çerçevesinin (bounding box) başka bir bağlama çerçevesiyle olan IoU (Intersection over Union) değerini kullanarak, belirli bir eşiğı aşan kutuları daha büyük kabul etmektir. Bu sayede, üst üste binen kutulardan en yüksek skora sahip olanı seçilirken, diğerleri bastırılır. Son zamanlarda, NMS'de daha fazla sınırlayıcı kutu bilgisi kullanarak doğruluğu artırma yöntemleri geliştirilmiştir. Güven eşiğı, yanlış pozitif hataları azaltarak, gereksiz tespit sonuçlarını bastırır ve bu da tespit modelinin genel doğruluğunu artırır. Böylece, daha doğru ve güvenilir nesne algılama sonuçları elde edilir[54]. Aşırı örtüşmeyi azaltmak için kullanılan "Non-Maximum Suppression" (NMS) adı verilen teknik eşik değerini belirler.



Şekil 2.22. Maksimum Olmayan Bastırmanın etkisi[55]

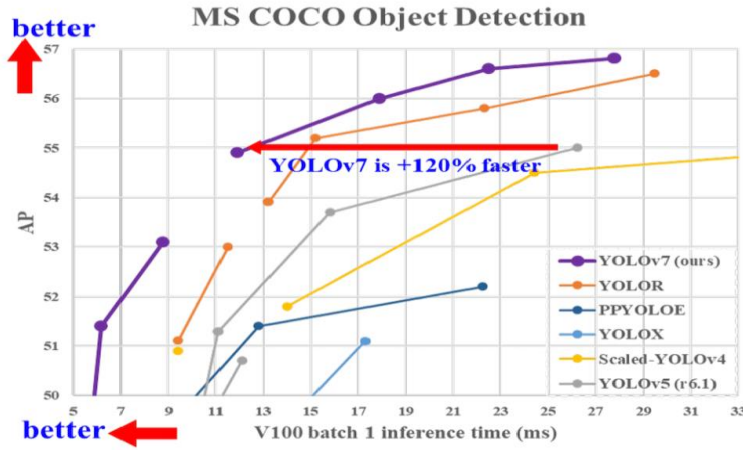
2.10.6. Training Parameters (Eğitim Parametreleri)

Modelin eğitilmesi sırasında kullanılacak öğrenme oranı, momentum değeri, epoch sayısı gibi parametreleri belirler. Bu değerler kritik öneme sahiptirler. Öğrenme oranı modelin öğrenme sürecinde belirlenir ve belirli bir değere ayarlanır. Bu oran örnek ağırlıklarının güncellenme hızını etkiler. Zamanla öğrenme oranı, düşük bir seviyeye ulaşana kadar giderek azalır. Bu modelin daha doğru öğrenmesini sağlar[55]. Momentum, bir optimizasyon yöntemidir. Bu yöntem, optimum sonuca ulaşmak için gerekli olan fonksiyon değerlendirme sayısını azaltarak optimizasyon sürecini hızlandırmayı ve algoritmanın etkinliğini artırmayı amaçlar. Hem iyi başlangıç değerleri hem de uygun momentum ayarları, bir sinir ağının başarılı bir şekilde öğrenmesi için kritik öneme sahiptir[56]. Model eğitimi sırasında veriler, tamamı yerine belirli parçalar halinde kullanılır. İlk parça ile model eğitilir ve başarıyı test edilir. Başarıya göre

ağırlıklar geriye yayılım yöntemiyle güncellenir. Ardından yeni bir veri kümesi ile eğitim süreci devam eder ve ağırlıklar tekrar güncellenir. Bu süreç, modelin en uygun ağırlık değerlerini bulmak için her eğitim adımında tekrarlanır. Bu eğitim adımlarına "epoch" denir.

2.10.7. Inference Time (Çıkarım Süresi)

Modelin bir görüntüyü işleyip sonuç üretme süresini belirler. Daha hızlı sonuçlar için optimize edilmiş modeller, daha yüksek çıkarım hızına sahip olabilir.

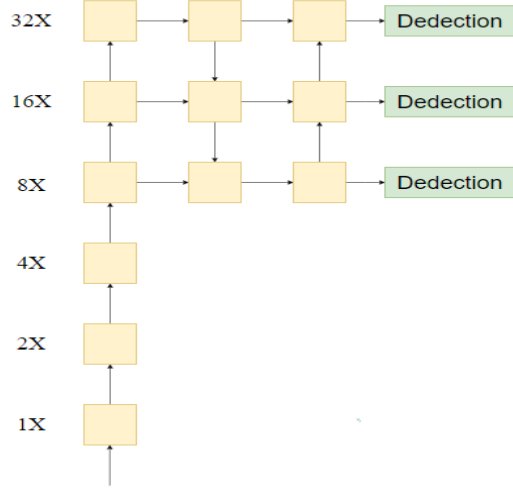


Şekil 2.23. Inference Time[57].

Farklı YOLO (You Only Look Once) modellerinin çeşitli performans metrikleri temelinde Çizelge 2.2’de karşılaştırmalı olarak verilmiştir.

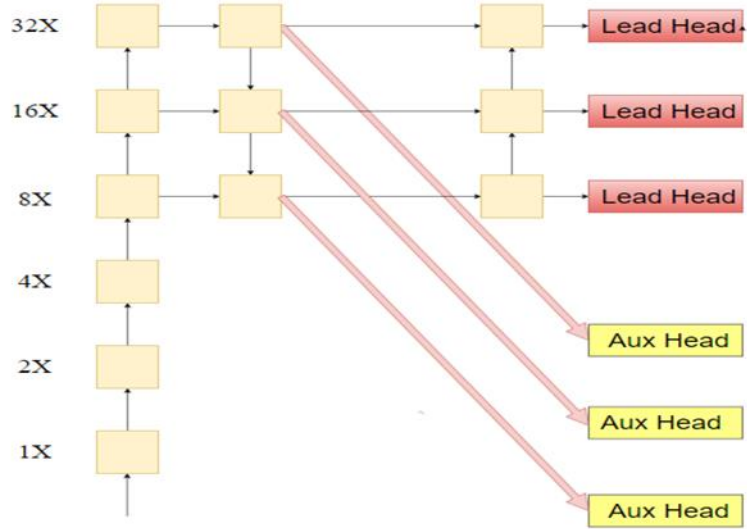
Çizelge 2.2. Performans Karşılaştırması[58]

Model	Test size	APtest	AP50test	AP75test	Batch 1 fps	Batch 32 average time
YOLOv7	640	51.4%	69.7%	55.9%	161 fps	2.8 ms
YOLOv7-X	640	53.1%	71.2%	57.8%	114 fps	4.3 ms
YOLOv7-W6	1280	54.9%	72.6%	60.1%	84 fps	7.6 ms
YOLOv7-E6	1280	56.0%	73.5%	61.2%	56 fps	12.3 ms
YOLOv7-D6	1280	56.6%	74.0%	61.8%	44 fps	15.0 ms
YOLOv7-E6E	1280	56.8%	74.4%	62.1%	36 fps	18.7 ms



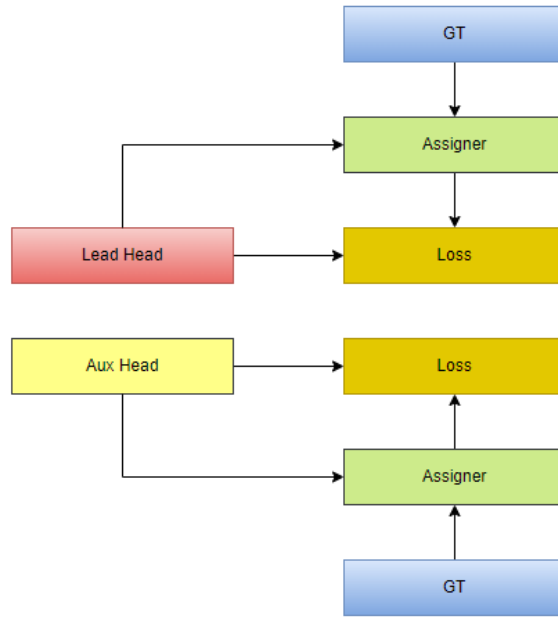
Şekil 2.24. Normal model[59]

Derin ağların eğitiminde kullanılan derin denetimin olmadığı örnek bir nesne dedektör mimarisi görülmektedir[59].



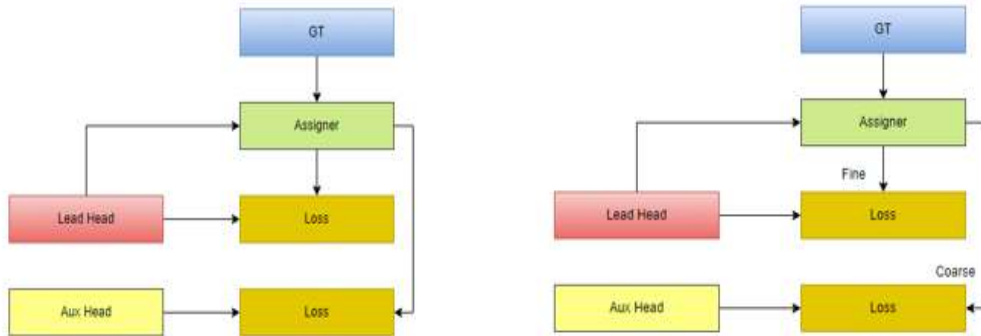
Şekil 2.25. Modele aux head eklenmiş hali

[59] Tarafından Şekil 2.24'deki nesne dedektör mimarisine çıktıdan sorumlu Lead Head kısmına ek olarak Şekil 2.25'de görüldüğü üzere Aux Head ismini verdikleri ağın eğitilmesine yardımcı olacak bir yapı eklenmiş halidir. Bu iki yapı arasındaki iş birliği, modelin performansını artırabilir ve daha karmaşık öğrenme görevlerini daha etkili bir şekilde gerçekleştirmesine olanak tanır. Ana yapı, genellikle daha güçlü öğrenme yeteneklerine sahipken, yardımcı yapı, öğrenme sürecini destekleyerek ve modelin daha geniş bir bilgi yelpazesini öğrenmesine yardımcı olarak önemli bir rol oynar[59].



Şekil 2. 26. İndependent assigner

Şekil 2.26’da etiket atamada verilmiş olan modelde hem lead head hem de aux head ismini verdikleri ana ve yardımcı yapılar tarafından bağımsız etiket atama işlemi için iki yapı yapının birbirinden ayrıldığı görülmektedir[59].

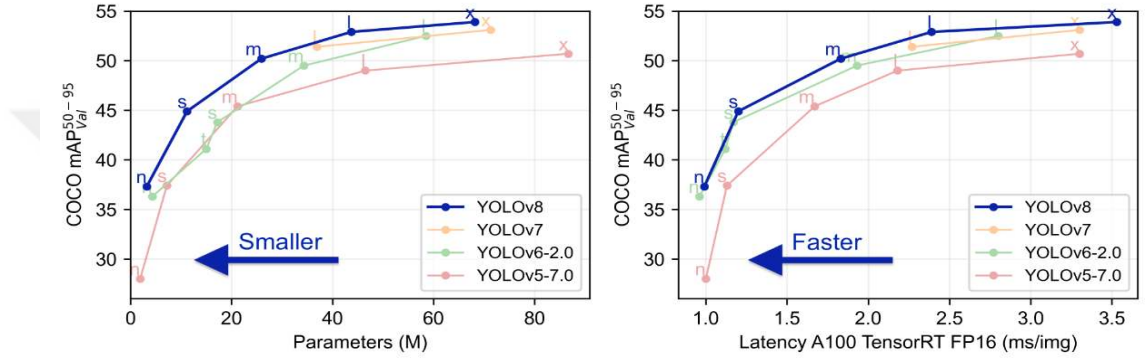


Şekil 2.27. Lead guided assigner and coarse-to-fine lead guided assigner

Şekil 2.27’da yardımcı yapı ana yapı tarafından öğrenilmiş bilgileri direk alarak ana yapının daha fazla öğrenme sürecine katılmasına katkı sunarak modelin performansının artırılmasına ve daha karmaşık öğrenme görevlerinin yerine getirilmesine katkı sunar[59].

2.11. YOLOv8

Ocak 2023'te piyasaya sürülmüş olan ve Ultralytics [60] tarafından oluşturulmuş olan yolov8 önceki yolo modellerinin güncellenmesi neticesinde oluşturulmuş ve hız, doğruluk ve daha az hesaplama gücü ve bellek gerektirmesi gibi özellikleriyle ön plana çıkmaktadır[61]. YOLOv8, YOLO'nun önceki sürümüne göre performansı artıran hızlı, doğru ve kullanımı kolay bir model olarak karşımıza çıkarken modelinin omurgası YOLOv5 CSP modülünü sürdürmektedir[62]. YOLOv8, nesne algılama, segmentasyon, poz tahmini, izleme ve sınıflandırma gibi çoklu görme görevlerini destekler[31].

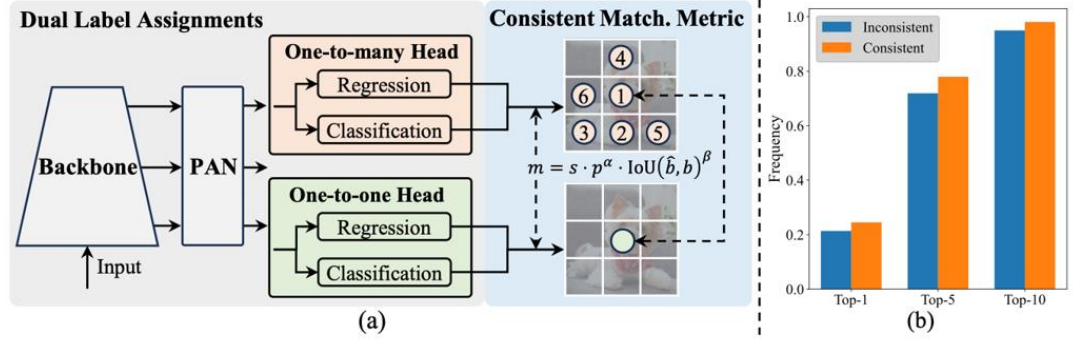


Şekil 2.28. YOLOv8 Performans Karşılaştırması

Çizelge 2.3. Verim Karşılaştırması[63]

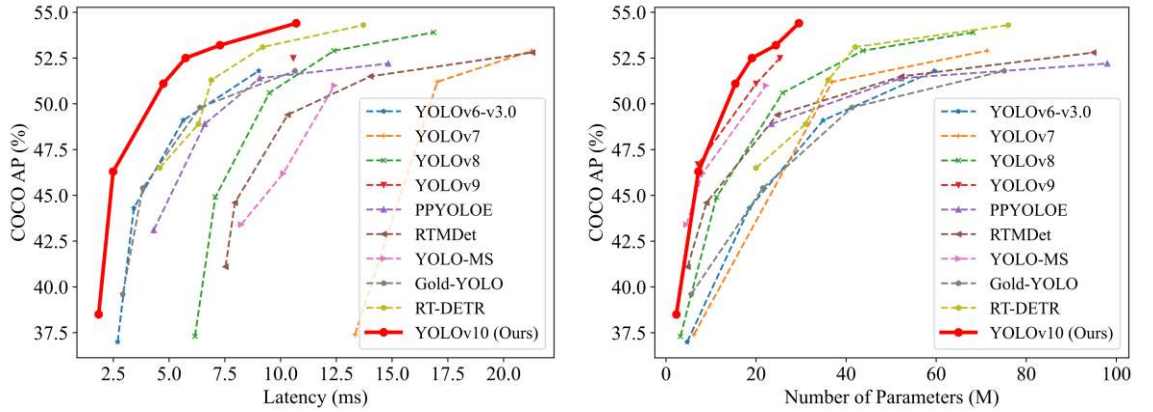
Modeli	Boyut (piksel)	mAP değeri 50-95	Hız CPU ON cNX (ms)	Hız A100 Tensor RT (ms)	Parametreler (M)	FLOP'lar (B)
YOLOv8n	640	37.3	80.4	0,99	3.2	8.7
YOLOv8s	640	44.9	128.4	1.20	11.2	28.6
YOLOv8m	640	50.2	234.7	1.83	25.9	78.9
YOLOv8l	640	52.9	375.2	2.39	43.7	165.2
YOLOv8x	640	53.9	479.1	3.53	68.2	257.8

2.12. YOLOv10



Şekil 2.29. YOLOv10 eğitim sürecinde çift etiketleme ve eşleşme metrikleri[64]

Şekil 2.29’da, YOLOv10 modelinin eğitim sürecini ve nesne tespitindeki çift etiket atamaları ile tutarlı eşleşme metriklerini özetlemektedir. Model, görüntü verilerini işleyerek özellik haritaları çıkarır, ardından birden fazla tahmin yapan "one-to-many" ve tek tahmin yapan "one-to-one" bölümlerini kullanarak nesnelerin konum ve sınıf bilgilerini belirler. [64]

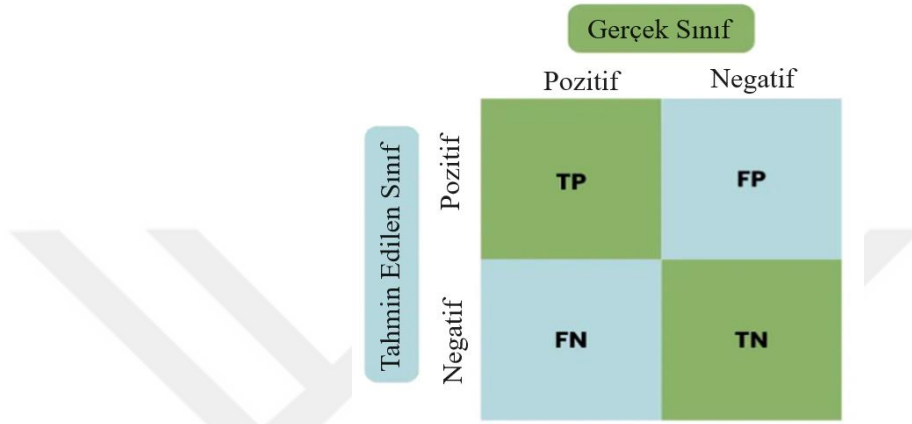


Şekil 2.30. Gecikme-doğruluk (solda) ve boyut-doğruluk (sağda) ödünleşimleri açısından diğerleriyle karşılaştırmalar.[64]

YOLOv10, gerçek zamanlı nesne tespitinde öne çıkan YOLO serisinin en yeni sürümüdür ve hesaplama maliyeti ile tespit performansı arasında etkili bir denge sağlamaktadır. Önceki sürümlerin Non-Maximum Suppression (NMS) algoritmasına bağımlılığı, sonlandırma gecikmesini artırırken, YOLOv10, NMS gereksinimini ortadan kaldırarak tutarlı ikili atamalarla eğitim yapmaktadır. Ayrıca, verimlilik ve doğruluğu artırmak için çeşitli bileşenlerin kapsamlı optimizasyonunu içeren bütünsel bir model tasarımı sunmaktadır. Deneysel sonuçlar, YOLOv10'un çeşitli model ölçeklerinde en iyi performansı sergilediğini göstermekte; örneğin, YOLOv10-S modeli, benzer AP değerine sahip RT-DETR-R18 modelinden 1.8 kat daha hızlıdır. [64]

2.13. Ultralytics YOLO Performans Metrikleri

YOLO (You Only Look Once), gerçek zamanlı nesne algılamada üstün başarı sağlayan popüler bir derin öğrenme algoritmasıdır. YOLO modelinizin performansını değerlendirmek, iyileştirmek ve diğer modelleri karşılaştırmak için Doğruluk (Accuracy), Hassasiyet (Precision), Duyarlılık (Recall), F1 Skoru, mAP50 kritik öneme sahip bazı metriklerdir.



Şekil 2.31. Karışıklık matrisi

2.13.1. Doğruluk (Accuracy)

Doğruluk, bir modelin doğru bir şekilde sınıflandırdığı örneklerin toplam örnek sayısına oranını ifade eder. YOLO modelinin genel performansını ölçmek için temel bir metriktir. Ancak, doğruluk tek başına tüm durumu anlatmaz. Özellikle dengesiz veri kümeleriyle çalışırken, doğruluk aldatıcı olabilir. Örneğin, bir modelin %96 doğruluk oranına sahip olması, düşük sayıda örnek içeren bazı sınıflarda kötü performans göstermesi anlamına gelebilir. Doğruluk değerinin hesaplaması için kullanılan matematiksel ifade denklem (2.6)'de yer almaktadır.

$$\text{Doğruluk} = (TP + TN) / (TP + TN + FP + FN) \quad (2.6)$$

2.13.2. Hassasiyet (Precision)

Hassasiyet, modelin tahmin ettiği pozitif örnekler içinde doğru pozitif örneklerin yüzdesini ölçer. Başka bir deyişle, modelin yanlış pozitif örnekleri en aza indirmesi için ne kadar başarılı olduğunu belirtir. Yüksek hassasiyet, modelin gereksiz veya yanlış tahminleri azaltması anlamına gelir. Hassasiyet değerinin hesaplaması için kullanılan matematiksel ifade denklem (2.7)'de yer almaktadır.

$$\text{Hassasiyet} = TP / (TP + FP) \quad (2.7)$$

2.13.3. Duyarlılık (Recall)

Duyarlılık, modelin gerek pozitif rnekler iinde doėru pozitif rneklerin yzdesini ler. Baėka bir deyiėle, modelin gerek pozitif rnekleri kaırmadan ne kadar baėarılı olduėunu belirtir. Yksek duyarlılık, modelin gerek pozitif rnekleri doėru bir ėekilde tanımlama olasılıėının yksek olduėunu gsterir. Duyarlılık deėerinin hesaplaması iin kullanılan matematiksel ifade denklem (2.8)'de yer almaktadır.

$$\text{Duyarlılık} = TP / (TP + FN) \quad (2.8)$$

2.13.4. F1 Skoru

F1 skoru, hassasiyet ve duyarlılıėın harmonik ortalamasıdır. Bu metrik, iki metrik arasında dengeli bir lt saėlar. Yksek bir F1 skoru, modelin hem yksek hassasiyet hem de yksek duyarlılıėa sahip olduėunu gsterir. Bu, modelin hem yanlış pozitifleri hem de yanlış negatifleri en aza indirmek iin iyi bir iė ıkardıėını gsterir. F1 Skoru, 0 ile 1 arasında deėer alır ve 1 en iyi skordur. F1 skoru deėerinin hesaplaması iin kullanılan matematiksel ifade denklem (2.9)'de yer almaktadır.

$$F1 \text{ Skor} = 2 * ((\text{Precision} * \text{Recall}) / (\text{Precision} + \text{Recall})) \quad (2.9)$$

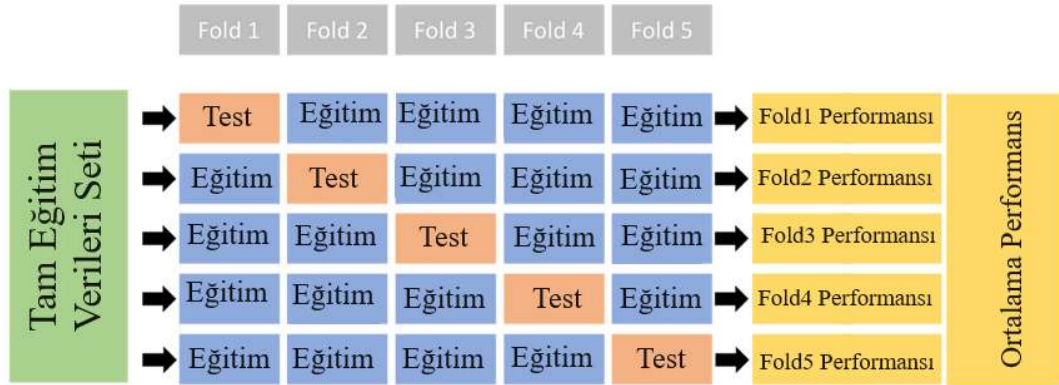
2.13.5. mAP50

mAP50 (Ortalama Ortalama Hassasiyet %50 Eėik Deėeriyle), YOLO modelinin farklı sınıflar iin hassasiyet-duyarlılık eėrilerinin altında kalan alanı ler. %50 eėik deėeri, modelin bir nesneyi doėru bir ėekilde tanımlaması iin gereken minimum eėleėme oranını belirtir. Yksek bir mAP50 skoru, modelin nesneleri doėru bir ėekilde tespit etme ve sınıflandırma yeteneėinin yksek olduėunu gsterir.

2.14. K-Fold Cross Validation

K-katlı apraz doėrulama ynteminin temel amacı, makine ėrenmesi modellerinin gerek dnya performansını daha doėru bir ėekilde tahmin etmektir. Burada veri seti, k eėit paraya (kat) blnr. Her seferinde bir kat test seti olarak kullanılır, geri kalan k-1 kat ise eėitim seti olarak kullanılır. Model, eėitim seti zerinde eėitilir ve test seti zerinde deėerlendirilir. Bu iėlem k kez tekrarlanır ve modelin performansının ortalaması alınır. Bu yntem, tek bir eėitim-test blm yerine tm veri seti zerinde modelin performansını deėerlendirir ve daha doėru performans tahmini saėlar. K-Katlı apraz Doėrulama, bir veri setini eėitmek ve test etmek iin kullanılan bir makine ėrenimi tekniėidir. Veri seti, k katlı bir ėekilde blnr ve her bir blm

sırayla test için kullanılırken diğerleri eğitim için kullanılır. Bu işlem k kere tekrarlanır ve sonuçlar ortalaması alınarak modelin performansı değerlendirilir. Bu teknik, modelin genelleme yeteneğini artırmak ve aşırı öğrenmeyi önler[65].



Şekil 2.32. K-Katlı çapraz doğrulama[65]

Toplam görüntü sayısı: 18613

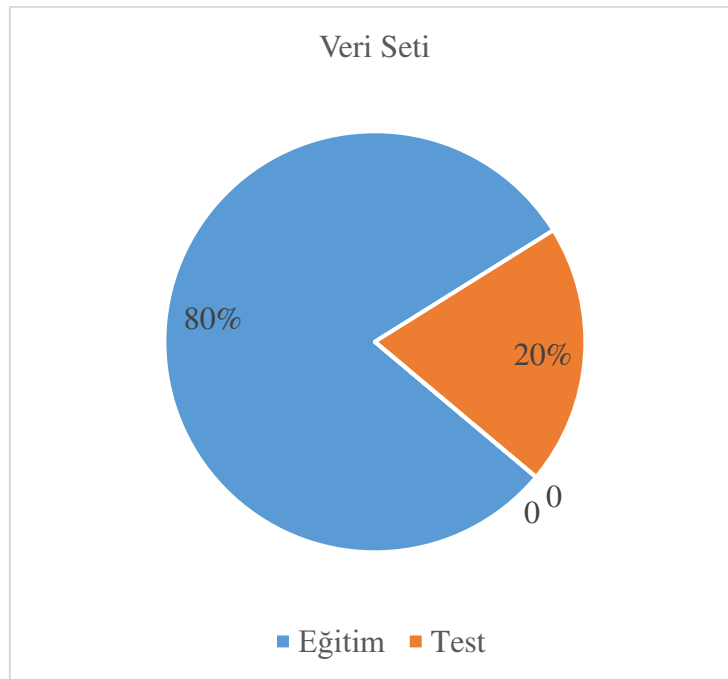
Fold 1: Train dosya sayısı: 14890, Val dosya sayısı: 3723

Fold 2: Train dosya sayısı: 14890, Val dosya sayısı: 3723

Fold 3: Train dosya sayısı: 14890, Val dosya sayısı: 3723

Fold 4: Train dosya sayısı: 14891, Val dosya sayısı: 3722

Fold 5: Train dosya sayısı: 14891, Val dosya sayısı: 3722



3. MATERYAL VE YÖNTEM

Tez çalışmamızda literatürdeki açık kaynak olarak paylaşılmış “İHA’lar üzerinden insan tespiti” veri kümelerinin detaylı incelenmesi yapılarak “İHA’lar üzerinden insan tespiti” probleminde kullanılması uygun görülen Yolov7, Yolov8 ve Yolov10 üzerinde çalışılıp denemeler yapılmıştır.

Probleme özel geniş çaplı bir veri kümesinin oluşturulmuş ve literatürde yeterince rastlanmayan kırsal bölgelerde toplanmış bir veri seti elde edilmiştir. Bu veri setleri hazırlanırken HERIDAL ve Okutama-Action veri kümesinden faydalanılmıştır.

YOLOv7, YOLOv8 ve YOLOv10 gibi popüler derin öğrenme tabanlı nesne tespit modellerinin yaklaşık 18.613 görüntülü bir veri seti üzerinde gerçekleştirilen performans karşılaştırmasını sunmaktadır. Google Colab platformu kullanılarak eğitilen modellerin, çeşitli performans ölçütleri üzerinden karşılaştırılması ve sonuçlarının yorumlanması bu çalışmanın temel amacını oluşturmaktadır.

Çizelge 3.1. Kullanılan hiperparametreler

Hyperparameters	Value or Types
Network size (width x height)	640 x 640
Learning rate	0.01
Momentum	0.937
Training batch size	16
Flipud	0
Scale	0.5
Fliplr	0.5

Çizelge 3.2. Google Colab'daki T4 GPU'nun teknik özellikleri

Özellik	Açıklama
GPU	NVIDIA Tesla T4
CUDA Cores	2560
Tensor Cores	320
VRAM	16 GB GDDR6
RAM	Genellikle 12 GB veya daha fazla
Disk Alanı	Geçici olarak yaklaşık 100 GB
Google Drive Entegrasyonu	Ekstra depolama alanı için kullanılabilir
İşlemci	Genellikle Intel Xeon
Yazılım Desteği	Python 3, TensorFlow, PyTorch, Keras vb.

3.1. HERIDAL

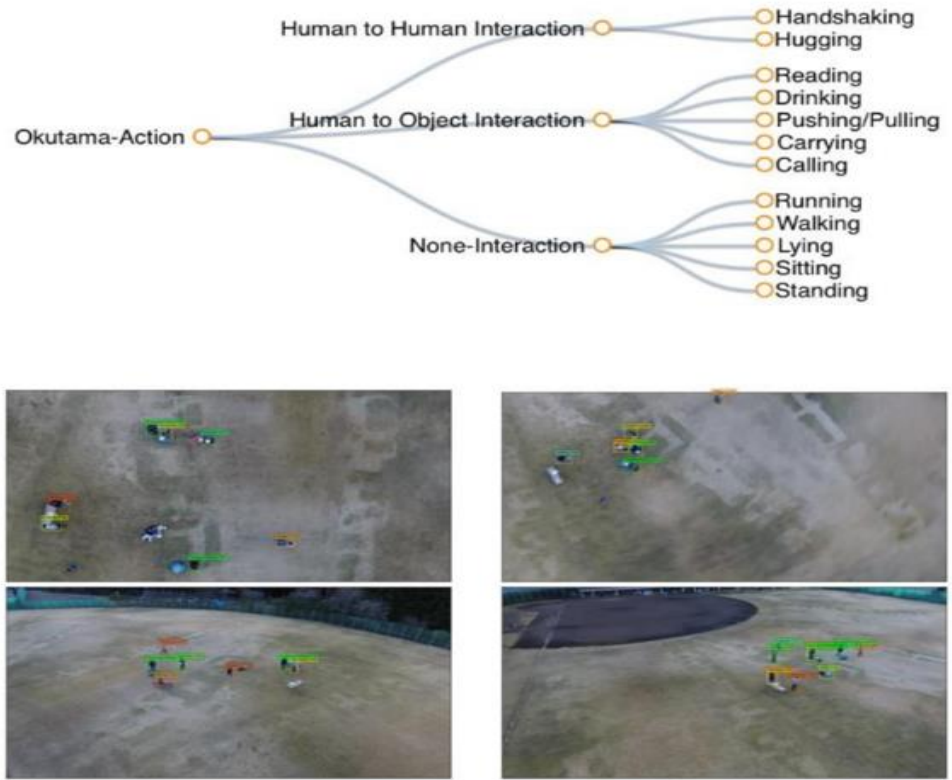
HERIDAL veri kümesinde, hava perspektifinden elde edilen 68.750'den fazla görüntü parçası bulunmaktadır. Bu görüntülerin içerisinde 29.050 pozitif örnek (insan içeren) ve 39.700 negatif örnek bulunmaktadır. Ayrıca bu veri kümesi yaklaşık 500 etiketli, 4.000 x 3.000 piksel çözünürlüklü gerçek dünya görüntüsü içermektedir. Bu veri kümesi insan tespiti konusunda kullanılabilecek en iyi veri kümesidir. Bunun nedeni, birçok kırsal alan arka planı bulunması ve gerçekçi zorlukları içermesidir. Örneğin; insanlar kimi zaman tepeden kimi zaman yandan, bazen gölgede bazen ise güneşin parlak olduğu anlarda, otururken, yatarken, koşarken ve çeşitli elbiseler içinde bulunmuşlardır. Veri kümesindeki bu çeşitlilik, nesne tespiti algoritmasının daha gerçekçi bir özellik çıkarabilmesini ve zorlu koşullar altında unsur tespiti yapılabilmesini sağlayacaktır.



Şekil 3.1. HERIDAL veri kümesinde bulunan ve insan içeren görüntülerden örnekler

3.2. Okutama-Action

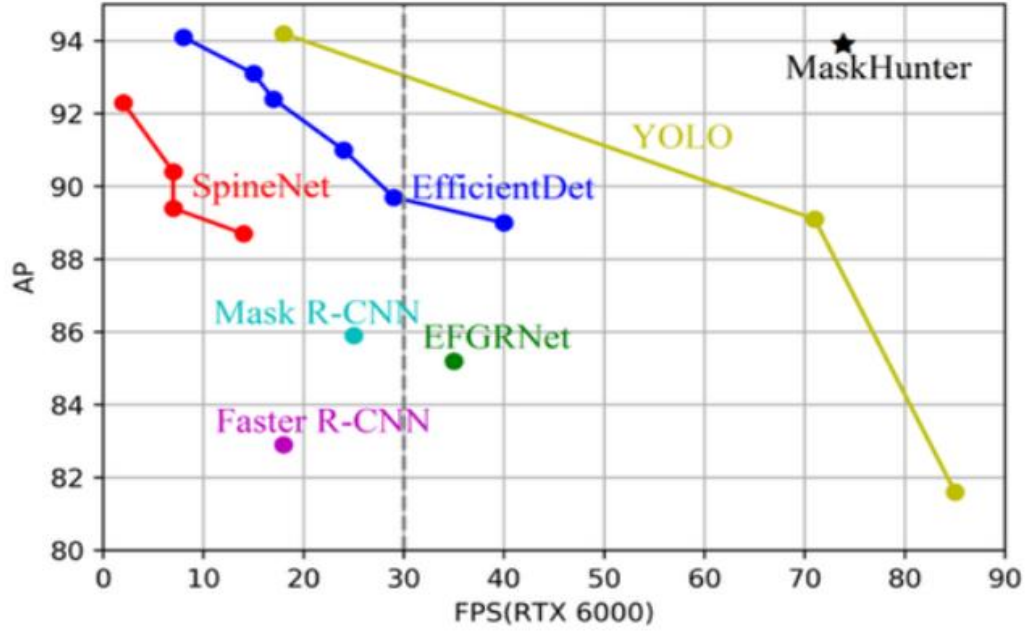
Okutama-Action, havadan çekilmiş fotoğraflar üzerinden gerçek zamanlı insan eylemi tespiti için toplanmış, açık kaynaklı bir video veri kümesidir. Bu veri kümesi 3840x2160 çözünürlüğünde, 12 eylem sınıfı ile 43 dakikalık tam açıklamalı dizilerden oluşmaktadır. Okutama-Action, eylemlerin dinamik geçişi, ölçek ve en boy oranındaki önemli değişiklikler, ani kamera hareketi ve çok etiketli aktörler dahil olmak üzere mevcut veri kümelerinde eksik olan birçok zorluğu içermektedir. Ayrıca, farklı arka planlara ait sahnelerden çekilmiş ve farklı görünümlerdeki insan görüntülerini içeren bu veri kümesinin geliştirilen derin öğrenme modellerinde kullanılmasıyla doğruluğun artması sağlanacaktır. Şekil 3.2’ de Okutama-Action veri kümesi görülmektedir.



Şekil 3.2. Okutama-Action veri kümesinde insan bulunan görüntüler

3.3. Derin Öğrenme Modeli Seçilmesi ve Kullanımı

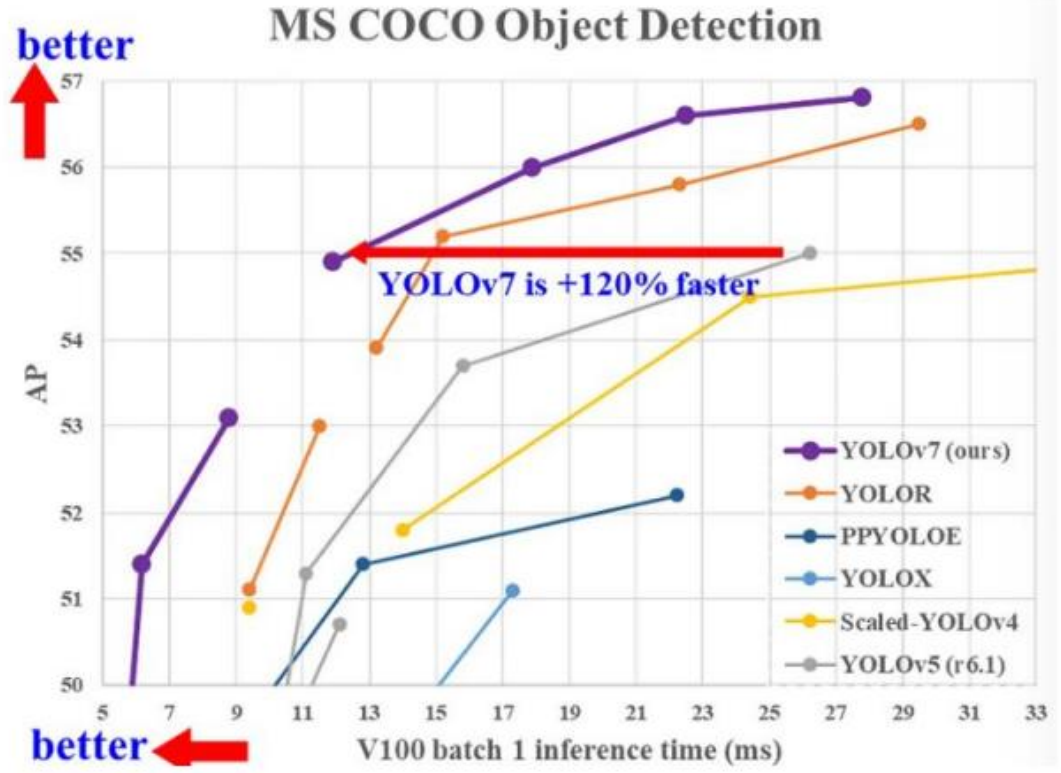
Global olarak geliştirilen sinir ağı mimarileri genellikle yeteneklerini kanıtlamak için COCO veri kümesi ile eğitilir ve test edilir. COCO veri kümesi ile geliştirilen ve tespit başarısı ile saniye başına işlem kabiliyeti oranı en yüksek olan açık kaynaklı yapay zekâ mimarisi YOLO mimarisidir. Şekil 3.3’te incelenen modellerin başarı performansları görülmektedir.



Şekil 3.3. Global modellerin başarımları grafikleri

YOLO, tek aşamalı bir yapıya sahip olup, evrimsel (konvolüsyonel) sinir ağlarını kullanarak nesne tespiti ve takibi yapabilen bir algoritmadır. YOLO algoritmasının diğer modellerden ayıran önemli özelliklerinden biri, nesne tespitini oldukça hızlı bir şekilde gerçekleştirebilmesidir. YOLO algoritması, giriş olarak verilen görüntülerin tamamını tek seferde sinir ağlarından geçirdiği için nesne tespitini daha hızlı yapabilmektedir. Ayrıca YOLO mimarisi, görüntüyü bölgelere ayırarak, bu bölgelerdeki nesneleri içine alan kutular çizer. Bu kutulara "sınırlayıcı kutu" (bounding box) adı verilir. Her bir ayrılan bölge, içerisinde bulunan nesnelerin olasılıklarını hesaplar. Ayrıca her bir sınırlayıcı kutu içerisinde bulunan nesnelerin yüzde kaç oranında tahmin edilen nesneler olduğunu veya benzeri olduğunu hesaplar.

Bu çalışmada, YOLO mimarilerinden en güncel olan YOLOv7 mimarisinin nesne tespit modeli olarak kullanılması planlanmaktadır. Bu seçimin sebebi, diğer YOLO modellerine kıyasla YOLOv7'nin daha hızlı ve doğru sonuçlar üretebilmesidir. YOLOv7 ve diğer YOLO modellerinin doğruluk/hız kıyaslaması Şekil 3.4'te gösterilmektedir. Şekil 3.4'te kırmızı oklar ile gösterilen dikey eksen AP (Ortalama Hassasiyet) değeri yani ortalama doğruluğun yukarı doğru arttığını, yatay eksen yani her bir fotoğrafın modelde işleme süresinin ise sola doğru azaldığını ve modelin hızlandığını göstermektedir. Şekil 3.4'ün ortasındaki kırmızı ok (sağdan sola doğru uzanan ok), YOLOv7 modelinin en doğru çalışan versiyonundan %120 daha hızlı olduğunu göstermektedir.



Şekil 3.4. Yolo modellerinin performans karşılaştırması

Derin Öğrenme kullanan nesne tespiti mimarileri, öğrenirken ilgili nesneyi ortaya çıkarmaya ve arka plan özelliklerinin etkisini azaltmaya yönelik ağırlıklar elde etmeye çalışmaktadır. Yapay zekâ modelinin unsur tespitinde istenilen başarıya ulaşabilmesi için farklı arka planlar kullanılarak, çeşitli pozisyonlarda, farklı renkte kıyafetler içinde, farklı ışık koşullarında ve çeşitli açılardan çekilmiş görüntüler kullanılmalıdır. Literatürdeki açık kaynaklı verilerin yanı sıra çeşitliliği artırmak adına ve tez çalışmasında ele alınan tarzda verilerin elde edilebilmesi için bir veri kümesi oluşturulmaya başlanmıştır. Tez kapsamında tespit edilmesi hedeflenen insanların, çeşitli kamuflajlar ile örtünüp arazideki bitki örtüsü, kayalık gibi şeylerin arkasına saklanacağı ön görüldüğünden, arka plan renginde kıyafet giyen ve saklanan insanlara ait görüntüler toplanmaya çalışılmıştır. Veri toplama işlemi DJI Phantom 4 pro ve 4096 x 2160 çözünürlüğündeki kamera ile gerçekleştirilmiştir.

3.4. Nemrut Kalderası Kırsalından Elde Edilen Veri Seti

Veri kümesinin hazırlanması için Nemrut Kalderası'nın seçilmesinin nedeni, farklı arka planlara sahip birçok arazi tipini barındırmasıdır. Bu sayede hedeflenen veri çeşitliliğini sağlamanın daha kolay olması hedeflenmiştir. Bölgenin sık ağaçlık, çalılık,

çorak bitki örtüsü ve yer yer kayalıklar içermesi çekim için uygun görülmüştür. Veri toplama sürecinde 20 ila 50 metre yükseklikten çekim yapılmıştır. Toplamda 15 dakikalık 4096 x 2160 çözünürlüğünde video görüntüsü toplanmıştır. Çekimler sırasında kullanılan kamera yatay olarak 90, 80, 70 ve 60 derecelere ayarlanmıştır. Figüranlar sürekli olarak hareket halinde olmuş ve ağaçlar, çalılıklar, kayalıklar gibi nesnelerin arkasına gizlenmiş, yer yer uzanmış, sürünmüş, oturmuş veya koşmuştur. Açık bir günde, maksimum 3 m/s rüzgâr hızında öğlen saatlerinde çekimler gerçekleştirilmiştir. Veriler öncelikle gereksiz sahnelerden ve tekrar eden verilerden ayıklanmış, yaklaşık 1000 adet fotoğraf bırakılarak sadeleştirilmiştir. Tamamen manuel olarak etiketlenen veriler, ilk modelin eğitiminde kullanılmıştır.



Şekil 3.5. Nemrut kalderası veri kümesinden görüntüler

3.5. Bitlis Eren Üniversitesi Kampüsü İçinde Elde Edilen Veri Seti

Nemrut bölgesinde gerçekleştirilen veri toplama işlemlerinin yeterli bulunmaması üzerine farklı bölgelerde çekimler yapılmıştır. Bu kapsamda, Bitlis Eren Üniversitesi kampüsü içerisinde bulunan çayır, fidanlık ve kırsal alanlarda çekimler gerçekleştirilmiştir. Bu aşamada insan tespiti problemi ile ilgili farklı olasılıklar üzerine odaklanılmıştır. Örneğin, insan boylarında fidanların bulunduğu bir arazide oldukça uzak mesafelerden (40m - 60m+) çekimler yapılarak çok sayıda negatif (insan bulunmayan) ve pozitif (insan bulunan) veri elde edilmiştir. Ayrıca eğitilen insan tespiti modellerinde gözlemlenen bir zayıf noktaya odaklanılarak, insanlar birbirlerine çok yakın olduğunda tek insan algılama problemi için özellikle yan yana çekimler de

yapılmıştır. 40-60 metre irtifadan elde edilen bu görüntüler, önceki çekimler gibi dik değil, yatay eksen ile 10-30 derece açı olacak şekilde çekilmiş ve mesafe daha da artırılmıştır. Elde edilen veri kümesi yaklaşık 20 dakikalık video formatında olmuştur. Çekimler açık havada ve ikindi vaktinde gerçekleştirilmiştir.

Bitlis Kampüs veri kümesinin etiketlenmesinde yarı otomatik bir yöntem kullanılarak daha az iş gücü ihtiyacı oluşturulmuş ve daha hızlı ilerleme sağlanmıştır. Öncesinde gereksiz sahneler ve tekrar eden verilerden ayıklanarak model sadeleştirilmiştir. Daha sonra farklı bir dokuya sahip Nemrut verileri ile eğitilen YOLO modeli ile tespit işlemine sokularak yapılan işaretlemeler txt uzantılı olarak kaydedilmiş ve otomatik etiketleme işlemi gerçekleştirilmiştir. Ardından bu bilgilerle görüntüler etiketleme platformu üzerinden denetlenmiş ve hatalı işaretlemeler düzeltilmiştir.



Şekil 3.6. Bitlis Kampüs veri kümesinden görüntüler

3.6. Model Eğitim Ortamının Hazırlanması ve Eğitimlerin Yapılması

Bu çalışmada, model eğitiminin hızını ve doğruluğunu artırmak amacıyla Google Colab üzerinde güçlü bir donanım altyapısı kullanılmıştır.

3.7. Kurulumlar

Model tespit işlemini gerçekleştirecek YOLO mimarisinin GPU üzerinden çalıştırılabilmesini sağlayan CUDA sürümünü ve temel kütüphaneleri sisteme kurmak gerekmektedir. CUDA sürümü yapay zekâ modelinin çalıştırılacağı donanımın türüne göre değişiklik göstermektedir. Bu hususta GPU donanımının desteklediği ve pytorch geliştiricilerinin önerdiği CUDA 11.3 sürümü kurulmuştur. YOLOv7 modelinin uygun

bir şekilde eğitilebilmesi için geliştiriciler gereklilikler (requirements) adlı txt dosyasını oluşturmuşlardır. TXT dosyası kullanılıp gerekli kütüphaneler sisteme entegre edilerek model kullanıma hazır hale getirilmiştir.

Çizelge 3.3. Yazılım kütüphaneleri ve versiyonları

Kütüphane	Versiyon	Açıklama
matplotlib	$\geq 3.3.0$	Versiyon 3.3.0 ve üzeri
numpy	$< 2.0.0, \geq 1.23.0$	Versiyon 1.23.0 ile 2.0.0 arasında
opencv-python	$\geq 4.6.0$	Versiyon 4.6.0 ve üzeri
pillow	$\geq 7.1.2$	Versiyon 7.1.2 ve üzeri
pyyaml	$\geq 5.3.1$	Versiyon 5.3.1 ve üzeri
requests	$\geq 2.23.0$	Versiyon 2.23.0 ve üzeri
scipy	$\geq 1.4.1$	Versiyon 1.4.1 ve üzeri
torch	$\geq 1.8.0$	Versiyon 1.8.0 ve üzeri
torchvision	$\geq 0.9.0$	Versiyon 0.9.0 ve üzeri
tqdm	$\geq 4.64.0$	Versiyon 4.64.0 ve üzeri
pandas	$\geq 1.1.4$	Versiyon 1.1.4 ve üzeri
seaborn	$\geq 0.11.0$	Versiyon 0.11.0 ve üzeri

3.8. Eğitim Parametrelerinin Ayarlanması

Yapay zekâ modelinin farklı ışık, ortam, unsur ve irtifa durumlarında başarılı bir şekilde çalışabilmesi için eğitim öncesi parametreleri ayarlanarak veri arttırma (data augmentation) işlemi gerçekleştirilmektedir. YOLO mimarisinin geliştiricileri, eğitim öncesi parametreleri oluşturarak döndürme, kırpma, parlatma, karartma, renk doygunluğu gibi veri çoğaltma işlemlerini kolayca gerçekleştirmeyi sağlamışlardır.

Çizelge 3.4. Gerçekleştirilen eğitimde kullanılan hiper parametreler

	Hyperparameters	Value or Types
1	Network size (width x height)	640 x 640
2	Learning rate	0.01
3	Momentum	0.937
4	Training batch size	16
5	Flipud	0
6	Scale	0
7	Fliplr	0.5

3.9. Veri Seti

Çalışmada kullanılan veri seti, yaklaşık 18.613 görüntüyü içermektedir. Bu görüntüler çeşitli ortamlarda ve koşullarda çekilmiş olup, farklı nesne türlerini ve pozisyonlarını içermektedir. Veri setinin geniş ve çeşitli olması, modellerin gerçek dünya senaryolarında daha iyi genelleme performansı göstermelerine olanak sağlamaktadır. Veri setinin etiketlenmesi, her görüntünün içindeki nesnelerin sınıfını ve konumunu belirten sınırlayıcı kutular kullanılarak gerçekleştirilmiştir. Etiketleme işlemi, nesne tespit modellerinin eğitimi için olmazsa olmaz bir adımdır, çünkü modeller bu etiketlere göre nesneleri tanımayı öğrenmektedir.



Şekil 3.7. YOLO etiket dosyasının içerik örneği

Çizelge 3.5. Nesne sınıflarının koordinat ve boyut bilgileri

Nesne Sınıfı	X Değeri	X Değeri	Genişlik	Yükseklik
0	0.48	0.63	0.69	0.71
0	0.74	0.52	0.31	0.93
27	0.36	0.79	0.07	0.40



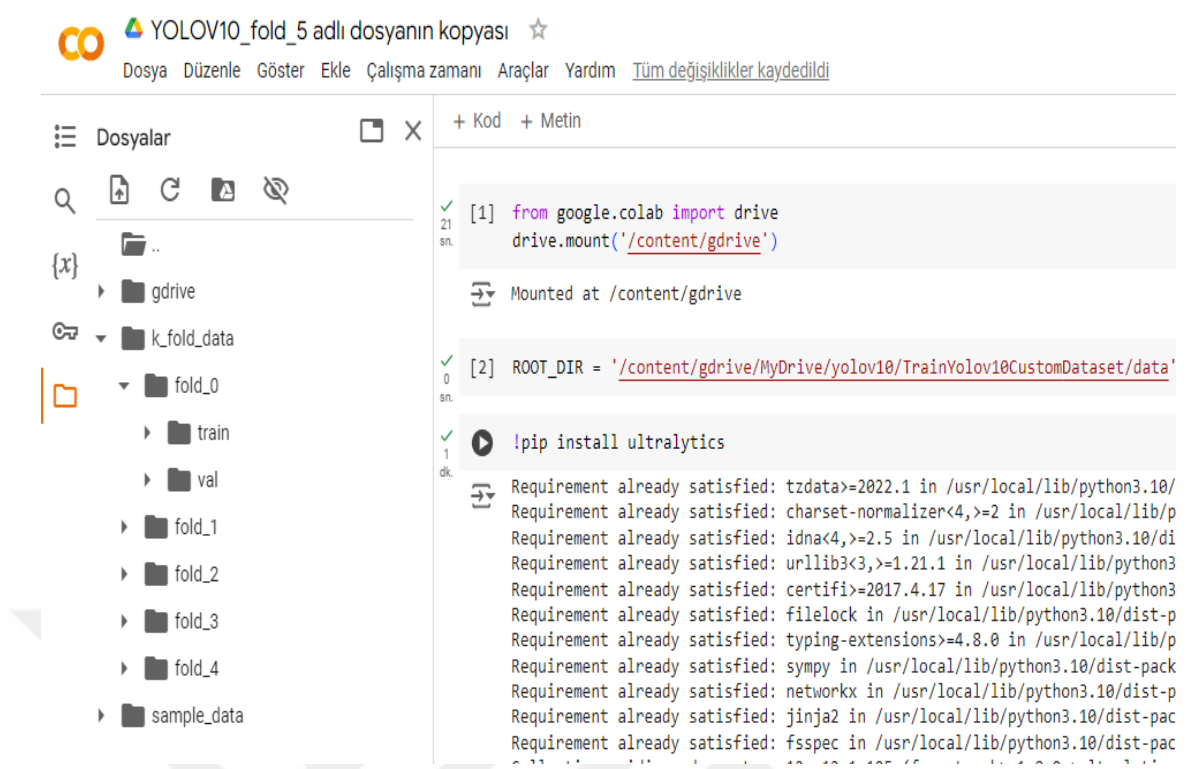
Şekil 3.8. Veri setimize ait görüntü örneği

Çizelge 3.6. Veri setimizdeki nesne sınıflarının koordinat ve boyut bilgileri örneği

Nesne Sınıfı	X Değeri	X Değeri	Genişlik	Yükseklik
0	0.09453125	0.31171875	0.0171875	0.0328125
0	0.4875	0.4265625	0.021875	0.053125

3.10. Eğitim Süreci

Modellerin eğitimi, Google Colab platformu kullanılarak gerçekleştirilmiştir. Google Colab, kullanıcılara ücretsiz olarak GPU ve TPU erişimi sağlayan bulut tabanlı bir makine öğrenmesi platformudur. Bu platform, modellerin eğitilmesi için gereken büyük hesaplama gücünü sağlar. Eğitim süreci, modellerin veri setindeki görüntülere göre ayarlanması ve en iyi performans gösteren parametreleri bulması için gerçekleştirilmiştir. Bu süreç, modellerin eğitim verilerindeki nesneleri tanımayı öğrenmesini içerir. Eğitim işlemi tamamlandıktan sonra, modellerin performansı, test verileri üzerinde değerlendirilmiştir.



Şekil 3.9. Google Colab üzerinde eğitim görseli

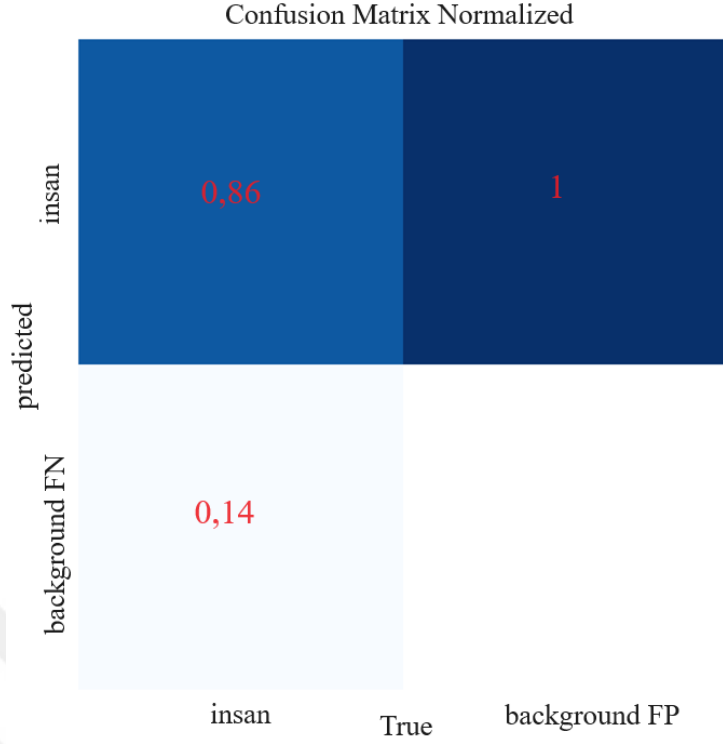
3.11. YOLOv7 ile Elde Edilen Sonuçlar

Google Colab platformunda YOLOv7 algoritmasının eğitim süreci, K-Fold çapraz doğrulama tekniği kullanılarak yapıldı ve her bir fold için 100 epochs değerine ait sonuçlara ait veriler Çizelge 3.7’teki gibidir.

Çizelge 3.7. Fold bazında mAP50 ve mAP değerleri

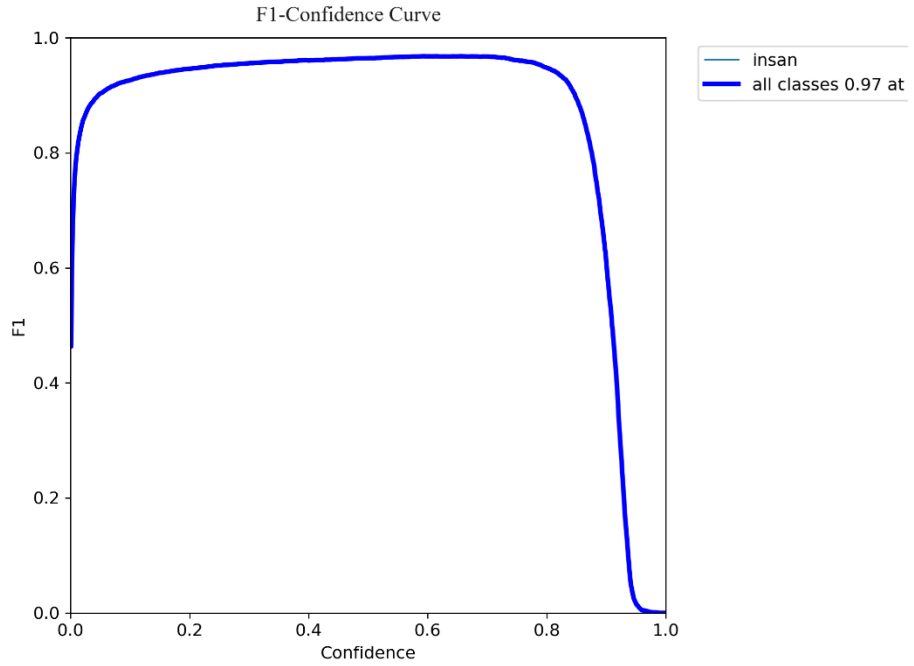
	mAP50	mAP
Fold 1	0.894	0.565
Fold 2	0.909	0.575
Fold 3	0.910	0.587
Fold 4	0.918	0.590
Fold 5	0.987	0.714

Çizelge 3.7’deki sonuçlar incelendiğinde mAP50 değerleri Fold1, Fold2, Fold3, Fold4 0.894- 0.918 arasında değişirken, Fold 5 mAP50 değeri ise 0.987 gibi yüksek bir değer ayrıştığı göze çarpmaktadır. Buda farklı veri dağılımlarının nesne tespitinde önemli olduğu göstermektedir.



Şekil 3.10. Eğitilen modele ait karmaşıklık matrisi

Modelimiz içinde insan olan resimleri doğru bir şekilde %86 oranında ve arka planları tamamen doğru bir şekilde tanımaktadır. Model, arka plan olarak tanımlanan görüntülerde %14 oranında yanlış sınıflandırma yapmıştır.



Şekil 3.11. Yolov7 ile eğitim sonucunda elde edilen F1 skor - güven grafiği

Şekil 3.11'deki grafik, modelin güven düzeyine göre F1 skorunu göstermektedir. Grafikte iki ana eğri bulunmaktadır:

İnsan Sınıfı (Açık Mavi Eğri): Bu eğri, modelin "insan" sınıfı için güven düzeyine göre F1 skorunu göstermektedir. Güven düzeyi 0,0'dan 1,0'a çıktıkça, F1 skoru genel olarak yüksek kalmaktadır. Bu, modelin insan sınıfını tanımada tutarlı bir performans sergilediğini göstermektedir.

Tüm Sınıflar (Koyu Mavi Eğri): Bu eğri, tüm sınıflar için F1 skorunun güven düzeyine göre değişimini göstermektedir. Eğrinin 0,86 değerinde sabitlenmektedir. Güven düzeyi arttıkça, modelin performansı da artmakta; ancak belirli bir noktadan sonra, performansın daha fazla iyileşmediği görülmektedir. Bu durum, modelin belirli bir güven seviyesinde sınıflandırma yeteneğinin sınırlı olduğunu göstermektedir.



Şekil 3.12. Eğitilen modelin test görüntülerinden tespit örnekleri

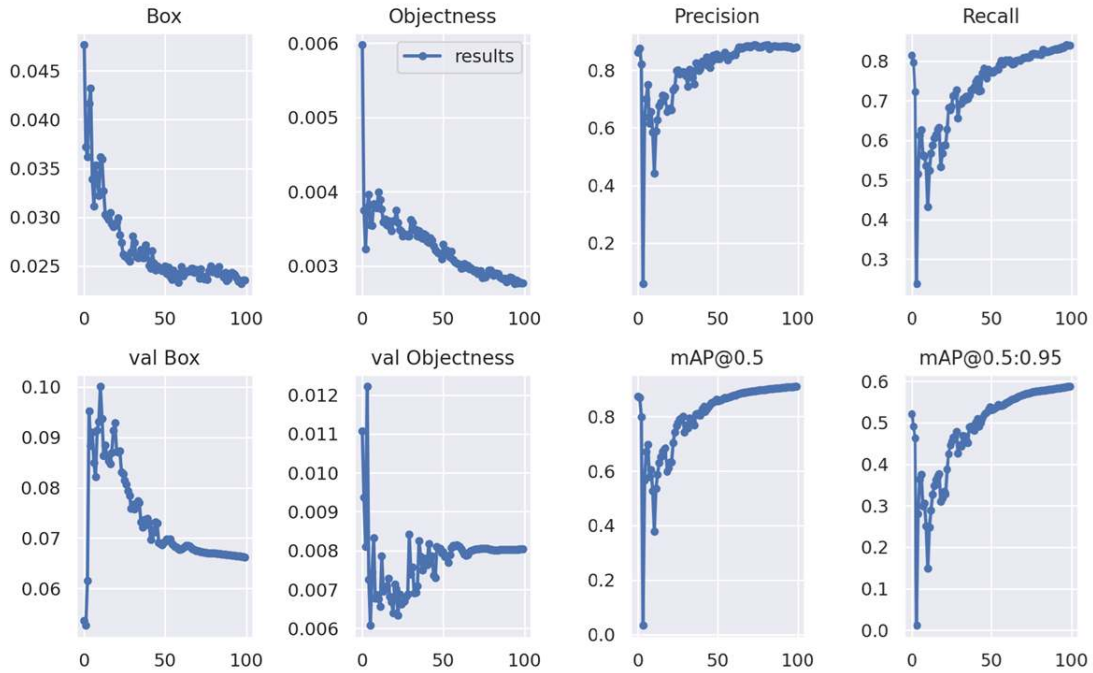
Şekil 3.12’de YOLOv7 modelimizin test aşamasında elde ettiği tespit örneklerini görmekteyiz. Görsellerde, modelin "insan" sınıfını tanıma yeteneği belirli bir güven düzeyi ile gösterilmektedir.

Her bir kutunun içinde yer alan sayılar, modelin tespit ettiği nesnelerin güven düzeyini göstermektedir. Örneğin, "insan 0.9" ifadesi, modelin bu tespiti yaparken %90 güvenle çalıştığını belirtmektedir. Bu, modelin genel olarak yüksek bir doğruluk oranına sahip olduğunu göstermektedir.

Farklı ortamlardaki tespitler, modelin çeşitli koşullarda nasıl performans gösterdiğini anlamamıza yardımcı olmaktadır. Görseller, açık alanlardan kapalı alanlara kadar geniş bir yelpazeyi kapsamaktadır.

Genel olarak, modelin "insan" sınıfını tanıma yeteneği oldukça başarılıdır. Ancak, bazı durumlarda güven düzeyinin daha düşük olduğu gözlemlenmiştir. Bu, modelin daha fazla veriye ihtiyaç duyabileceğini veya belirli durumlarda iyileştirilmesi gerektiğini gösteriyor.

Bu tespit örnekleri, modelimin gerçek dünya uygulamalarında nasıl çalıştığını ve hangi alanlarda geliştirilmesi gerektiğini değerlendirmekte önemli bir rol oynamaktadır.



Şekil 3.13. YOLOv7 Nesne tespiti modeli eğitim ve değerlendirme metrikleri

Şekil 3.13'deki grafikler, YOLOv7 modelinin eğitim sonuçlarını göstermekte olup, 18,613 görüntü üzerinde gerçekleştirilen 5 katlı (k-fold) çapraz doğrulama işleminin 3.katına ait sonuçları sunmaktadır. Eğitim sürecinde "Box" ve "Objectness" kayıplarının zamanla azaldığı, "Precision" ve "Recall" değerlerinin ise artış gösterdiği gözlemlenmektedir. Bu durum, modelin nesne tanıma yeteneğinin geliştiğini ve doğru sınıflandırma oranının yükseldiğini göstermektedir. Doğrulama setindeki kayıpların ve ortalama kesinlik (mAP) değerlerinin de benzer bir ilerleme sergilemesi, modelin genel başarısını desteklemektedir. Sonuç olarak, 3.katın verileri, modelin eğitim sürecindeki etkinliğini ve güvenilirliğini ortaya koymaktadır.

Çizelge 3.8'de YOLOv7 algoritmasının 100 epoch boyunca 5 farklı katlama (fold) için elde edilen sonuçlar, modelin genel performansının değişkenlik gösterdiğini ortaya koymaktadır. Doğruluk oranı %79 ile %91 arasında değişirken, en yüksek doğruluk Fold_5'te %91 olarak kaydedilmiştir. Hassasiyet, %91 ile %92 arasında sabit kalmış, bu da modelin yanlış pozitif oranının düşük olduğunu göstermektedir. Duyarlılık ise %86 ile %98 arasında değişmekte olup, Fold_5'te %98 ile en yüksek değere ulaşmıştır; bu durum modelin gerçek pozitifleri yakalama kapasitesinin oldukça yüksek olduğunu göstermektedir. F1 skoru %88 ile %95 arasında değişirken, bu da modelin hem hassasiyet hem de duyarlılık açısından dengeli bir performans sergilediğini göstermektedir. mAP (mean Average Precision) değerleri ise %565 ile %987 arasında değişiklik göstermekte olup, Fold_5'te en yüksek mAP değeri elde edilmiştir. Genel olarak, elde edilen veriler, YOLOv7'nin etkili bir nesne tespiti modeli olduğunu ancak bazı fold'larda performansın daha düşük olduğunu göstermektedir. Ayrıca Fold_5'te %98 gibi yüksek değeri görmesi veri setinin dağılımının ne kadar önemli olduğunu bize göstermektedir.

Çizelge 3.8. YOLOv7 Algoritmasının Farklı Katmanlar İçin Performans Değerleri

Algoritma	Fold	Doğruluk (Accuracy)	Hassasiyet (Precision)	Duyarlılık (Recall)	F1 Skoru	mAP50
YOLOv7	Fold_1	0.79	0.91	0.86	0.88	0.565
YOLOv7	Fold_2	0.80	0.91	0.87	0.89	0.575
YOLOv7	Fold_3	0.82	0.92	0.88	0.90	0.587
YOLOv7	Fold_4	0.82	0.91	0.89	0.90	0.918
YOLOv7	Fold_5	0.91	0.92	0.98	0.95	0.987



Şekil 3.14. Yolov7 modeline ait hatalı tespit örnekleri

Yolov7 modeli, bir bireyi yüksek bir güven düzeyiyle (0.9) tespit etmiştir. Ancak, aynı zamanda daha düşük güven düzeyine sahip (0.3) hatalı tespitler de mevcuttur. Bu durum, modelin bazı nesneleri insan olarak yanlış sınıflandırdığını göstermektedir. Özellikle, 'insan 0.4' ve 'insan 0.5' gibi düşük güven düzeyleri, modelin bu tespitlerde belirsizlik yaşadığını ortaya koymaktadır. Ayrıca, bazı tespitlerin güven düzeylerinin (örneğin 0.3) oldukça düşük olması, modelin yanlış sınıflandırma yapma olasılığını artırmaktadır.

3.12. YOLOv8 ile Elde Edilen Sonuçlar

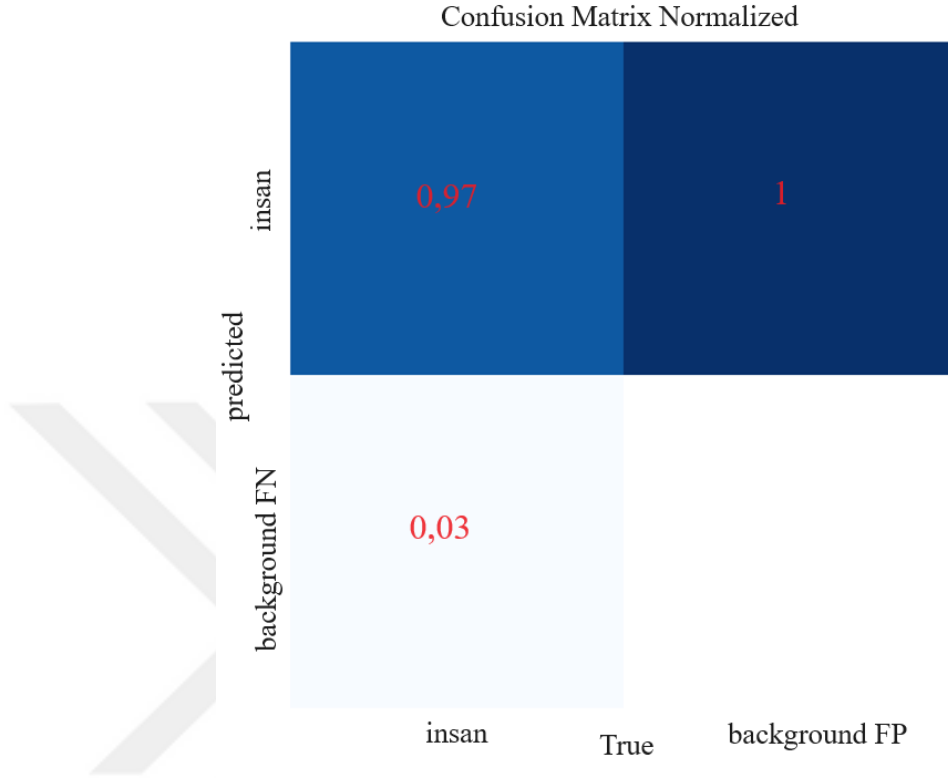
Google Colab platformunda YOLOv8 algoritmasının eğitim süreci, K-Fold çapraz doğrulama tekniği kullanılarak yapıldı ve her bir fold için 100 epochs değerine ait sonuçlara ait veriler Çizelge 3.9'deki gibidir.

Çizelge 3.9. Fold bazında mAP50 ve mAP değerleri

	mAP50	mAP
Fold 1	0.974	0.695
Fold 2	0.971	0.686
Fold 3	0.973	0.690
Fold 4	0.975	0.695
Fold 5	0.975	0.691

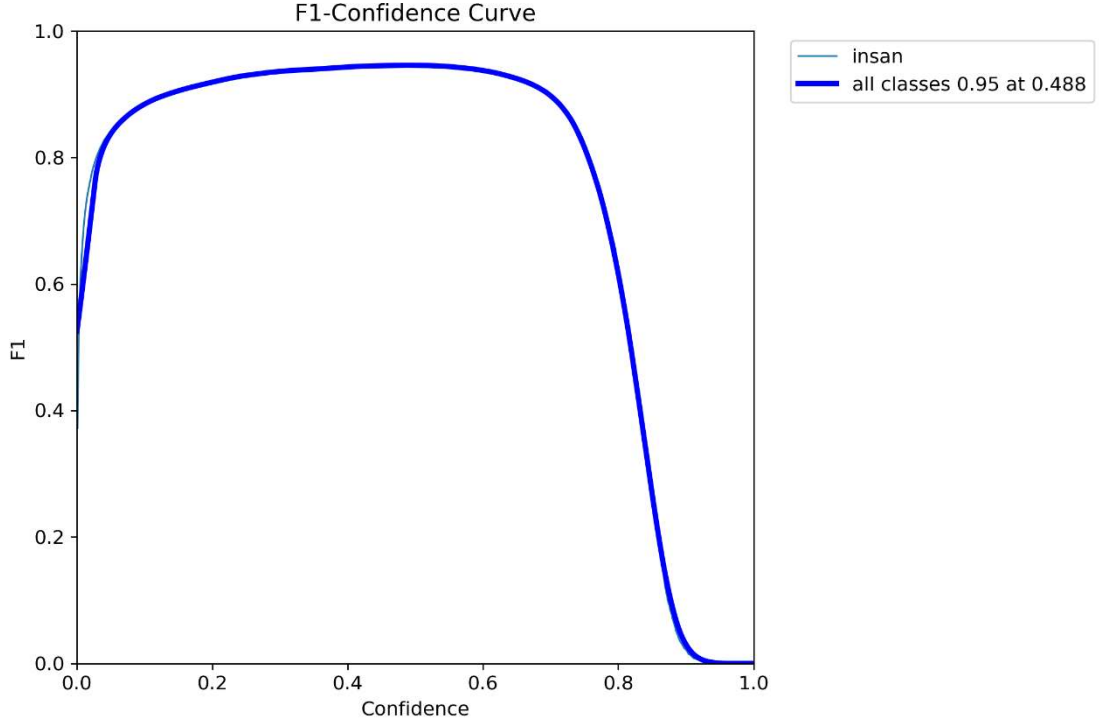
Çizelge 3.9'daki sonuçlar incelendiğinde mAP50 değerleri 0.97 civarında, yani modelin nesneleri %97 doğrulukla tespit ettiği anlamına geliyor. mAP değerleri ise 0.69 civarında olduğu görülmektedir. mAP, nesne tespitindeki genel performansı ölçen bir metrik ve bu değer 0.69 olması oldukça iyi bir performansa işaret etmektedir.

Fold'lar arasındaki sonuçlar da oldukça tutarlı olduğundan dengesiz bir veri dağılımının olmadığı anlaşılmaktadır. En yüksek mAP50 0.975 ile 4. Fold'da, en yüksek mAP ise 0.695 ile 1. Fold ve 4. Fold'da elde edilmiştir.



Şekil 3.15. Eğitilen modele ait karmaşıklık matrisi

Şekil 3.15'deki görsel, modelin performansını değerlendirmek için kullanılan normalize edilmiş bir karmaşıklık matrisini göstermektedir. Matris, modelin doğru tespit oranını ve yanlış sınıflandırma durumlarını ortaya koymaktadır. Üst hücredeki 0.97 değeri, modelin "insan" sınıfını doğru bir şekilde tanıma oranını ifade etmekte ve bu durum modelin yüksek bir doğrulukla çalıştığını göstermektedir. Alt hücredeki 0.03 değeri ise, modelin "insan" olarak tanımlaması gereken bir nesneyi kaçırdığı (False Negative) oranını belirtmektedir.



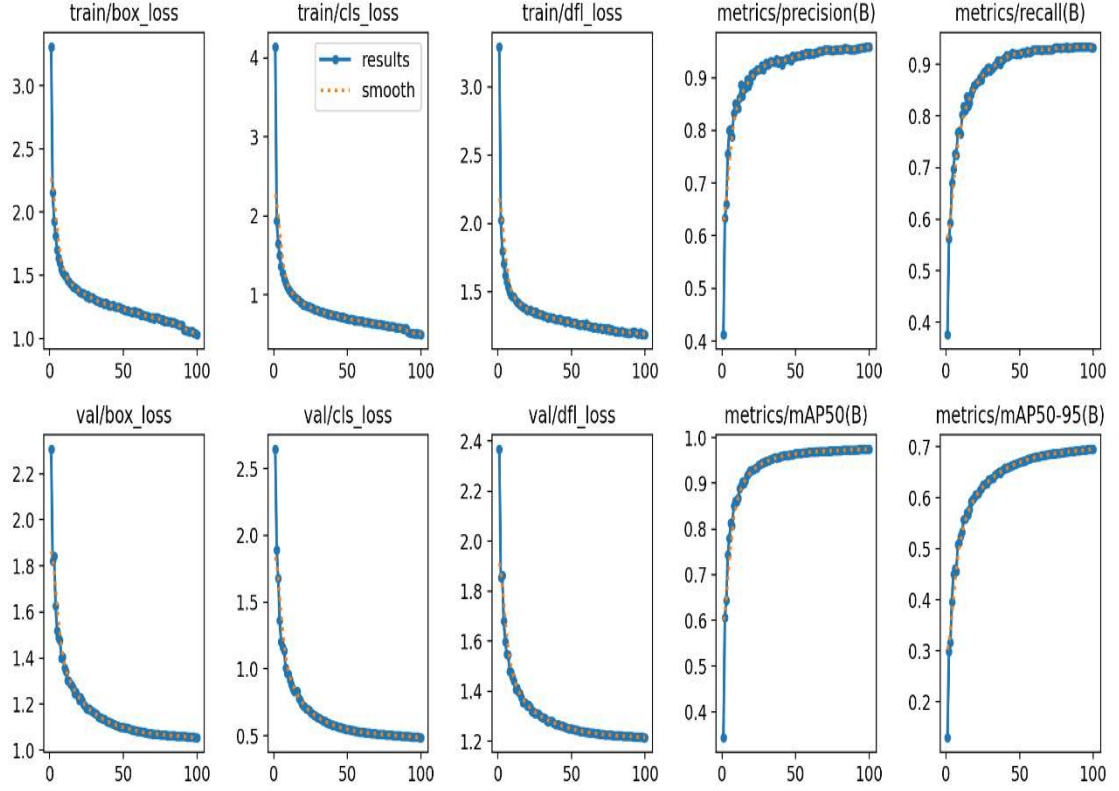
Şekil 3.16. Eğitim sonucunda elde edilen F1 skor - güven grafiği

F1 skoru, modelin doğruluğunu ve geri çağırma oranını dengeleyen bir metriktir. Yüksek F1 skoru, modelin hem doğru tahminler yapma hem de tüm ilgili nesneleri tespit etme yeteneğinin iyi olduğunu gösterir. Şekil 3.16'daki grafik, düşük güven seviyelerinde F1 skorunun hızla arttığını ve yaklaşık 0.2 güven seviyesinde maksimuma ulaştığını gösteriyor. Bu, modelimizin düşük güven seviyelerinde bile yüksek doğrulukla çalıştığını gösterir. Ayrıca tüm sınıflar için yaklaşık 0.488 güven seviyesinde 0.95 F1 skoru elde edilmiştir. Bu, modelimizin genel performansının oldukça iyi olduğunu göstermektedir.



Şekil 3.17. Eğitilen modelin test görüntülerinden tespit örnekleri

Şekil 3.17’de Yolov8 ile bir nesne tanıma modelinin insanları tespit etme yeteneğini gösteren çeşitli örnekler içermektedir. Her görüntüdeki sınırlayıcı kutular ve güven skorları, modelin her tespit için ne kadar güven duyduğunu göstermektedir.



Şekil 3.18. YOLOv8 Nesne tespiti modeli eğitim ve değerlendirme metrikleri

Şekil 3.18'deki grafikler, YOLOv8 modelin eğitim sırasında kayıpların azaldığını ve hassasiyet ile geri çağırma değerlerinin arttığını, bununla birlikte mAP (mean Average Precision) skorlarının iyileştiğini gözlemlemek mümkündür. Bu sonuçlar, modelin başarılı bir şekilde optimize edildiğini ve zamanla daha iyi performans gösterdiğini gösterir.

Çizelge 3.10'daki YOLOv8 algoritmasının 100 epoch boyunca 5 farklı katlama (fold) için elde edilen sonuçları değerlendirdiğimizde, modelin genel olarak yüksek performans sergilediğini görüyoruz. Doğruluk %89 ile %90 arasında değişirken, hassasiyet %92 ile %93 arasında sabit kalmıştır, bu da modelin yanlış pozitif oranının düşük olduğunu gösterir. Tüm fold'lar için duyarlılık %97 olarak yüksek kalmış, bu da modelin gerçek pozitifleri yakalama kapasitesinin mükemmel olduğunu ortaya koymaktadır. F1 skoru %94 ile %95 arasında değişirken, bu da modelin hem hassasiyet hem de duyarlılık açısından dengeli bir performansa sahip olduğunu gösterir. Ayrıca, mAP (mean Average Precision) değerleri %97 civarında olup, nesne tespiti görevinde modelin oldukça başarılı olduğunu göstermektedir. Genel olarak, elde edilen veriler, YOLOv8'in güvenilir ve etkili bir nesne tespiti modeli olduğunu ortaya koymaktadır.

Çizelge 3.10. YOLOv8 algoritmasının farklı katmanlar için performans değerleri

Algoritma	Fold	Doğruluk (Accuracy)	Hassasiyet (Precision)	Duyarlılık (Recall)	F1 Skoru	mAP50
YOLOv8	Fold_1	0.89	0.92	0.97	0.94	0.974
YOLOv8	Fold_2	0.89	0.92	0.97	0.94	0.971
YOLOv8	Fold_3	0.90	0.93	0.97	0.95	0.973
YOLOv8	Fold_4	0.90	0.93	0.97	0.95	0.975
YOLOv8	Fold_5	0.90	0.93	0.97	0.95	0.975



Şekil 3.19. Yolov8 modeline ait hatalı tespit örnekleri

Şekil 3.19’da Yolov8’in insan tespit sonuçları yer almakta. Soldaki görüntüde, modelin "insan" olarak tanımladığı nesnelerin güven düzeyleri sırasıyla 0.4 ve 0.8 olarak belirtilmiş. Sağdaki görüntüde ise, modelin tespit ettiği "insan" nesneleri için güven düzeyleri 0.7, 0.8 ve 0.6 olarak gösterilmiş. Düşük güven düzeyleri (örneğin, 0.4) modelin bazı durumlarda yanlış sınıflandırmalar yapabileceğini veya belirsizlik yaşadığını ortaya koymaktadır. Bu durum, modelin daha fazla eğitim ve iyileştirme gerektirdiğini göstermektedir. Sonuç olarak, bu görsel, modelin insan tespiti konusundaki doğruluk oranlarını ve potansiyel hatalarını değerlendirmek için önemli bir referans sunmaktadır.

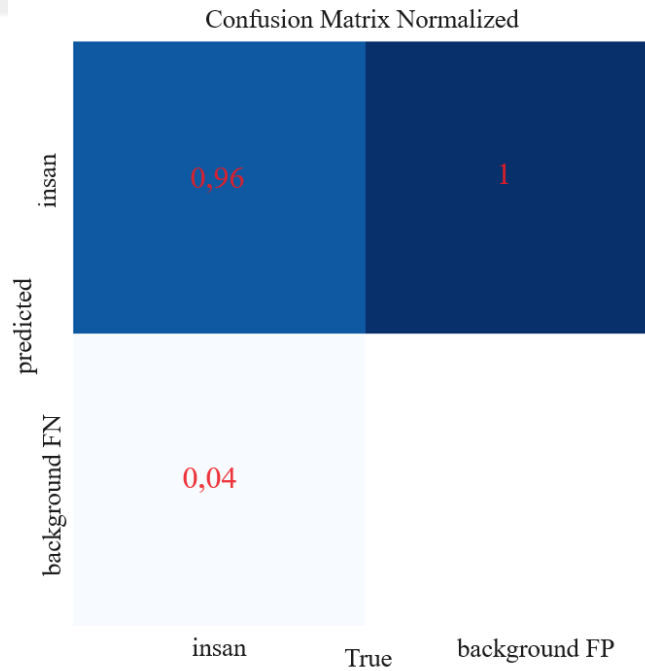
3.13. YOLOv10 ile Elde Edilen Sonuçlar

Google Colab platformunda YOLOv10 algoritmasının eğitim süreci, K-Fold çapraz doğrulama tekniği kullanılarak yapıldı ve her bir fold için 100 epochs değerine ait sonuçlara ait veriler Çizelge 3.11’deki gibidir.

Çizelge 3.11. Fold bazında mAP50 ve mAP değerleri

	mAP50	mAP
Fold 1	0.971	0.686
Fold 2	0.970	0.681
Fold 3	0.969	0.682
Fold 4	0.974	0.687
Fold 5	0.968	0.679

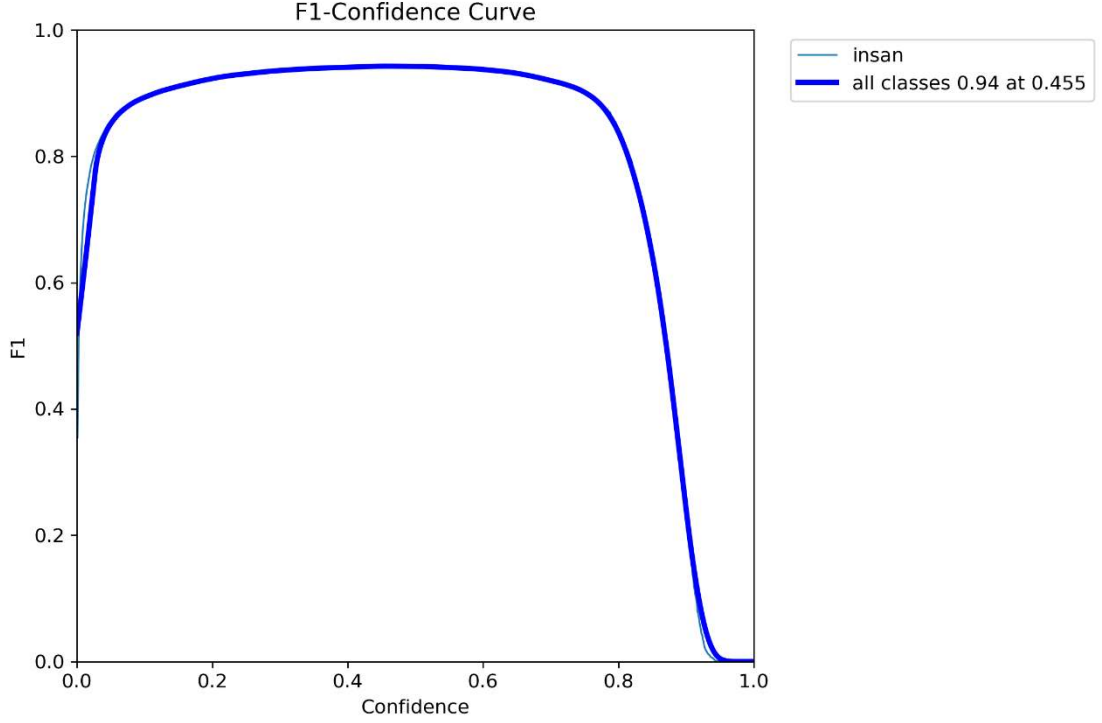
Çizelge 3.11'deki sonuçlar incelendiğinde mAP50 değerleri 0.97 civarında, yani modelin nesneleri %97 doğrulukla tespit ettiği anlamına geliyor. mAP değerleri ise 0.68 civarında olduğu görülmektedir. mAP, nesne tespitindeki genel performansı ölçen bir metrik ve bu değer 0.68 olması oldukça iyi bir performansa işaret etmektedir. Fold'lar arasındaki sonuçlar da oldukça tutarlı olduğundan dengesiz bir veri dağılımının olmadığı anlaşılmaktadır. En yüksek mAP50 0.974 ile 4. Fold'da, en yüksek mAP ise 0.687 ile yine 4. Fold'da elde edilmiştir.



Şekil 3.20. Eğitilen modele ait karmaşıklık matrisi

Şekil 3.20'deki görsel, modelin performansını değerlendirmek için kullanılan normalize edilmiş bir karmaşıklık matrisini göstermektedir. Matris, modelin doğru tespit

oranını ve yanlış sınıflandırma durumlarını ortaya koymaktadır. Üst hücredeki 0.96 değeri, modelin "insan" sınıfını doğru bir şekilde tanıma oranını ifade etmekte ve bu durum modelin yüksek bir doğrulukla çalıştığını göstermektedir. Alt hücredeki 0.04 değeri ise, modelin "insan" olarak tanımlaması gereken bir nesneyi kaçırdığı (False Negative) oranını belirtmektedir.



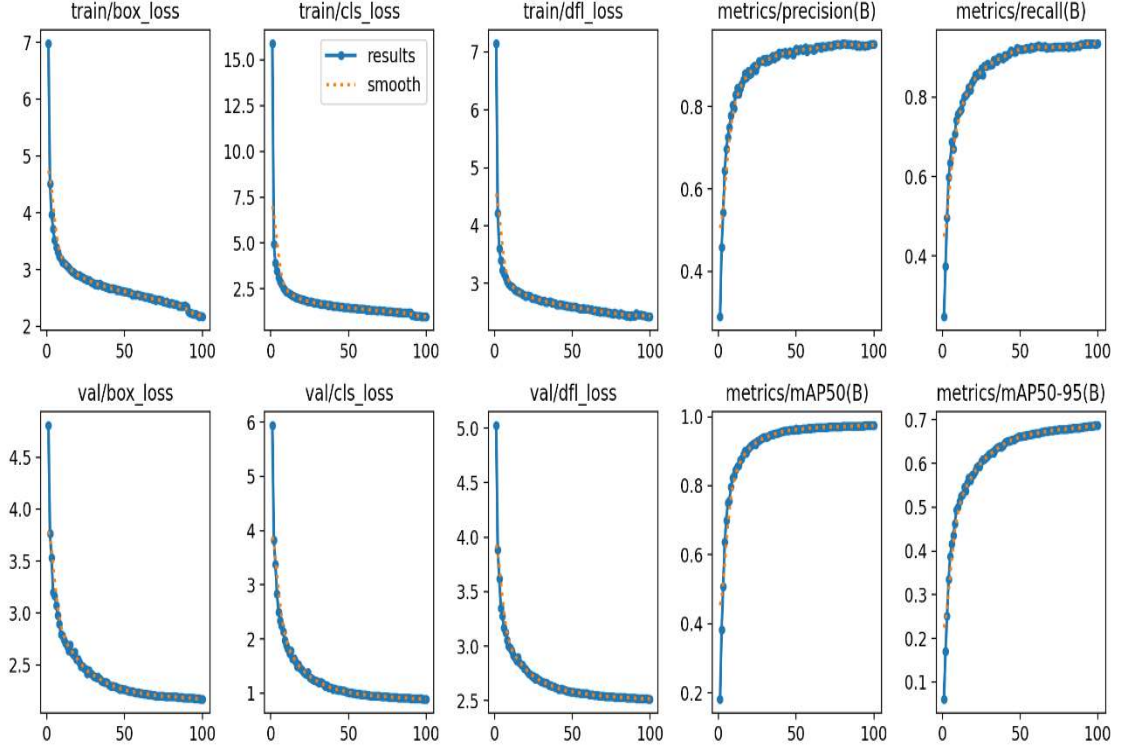
Şekil 3.21. Eğitim sonucunda elde edilen F1 skor - güven grafiği

Şekil 3.21'deki grafik, bir modelin insan tespiti için F1 skoru ile güven arasındaki ilişkiyi gösteren F1-Güven Eğrisi'ni temsil eder. Eğri, farklı güven seviyelerinde F1 skorunu gösterir ve modelin performansını değerlendirmek için kullanılır. Grafikte, tüm sınıflar için en yüksek F1 skoru 0.94 olarak 0.455 güven seviyesinde elde edilmiştir. Bu noktadan sonra, güven arttıkça F1 skoru hızla düşmektedir, bu da modelin güven eşiği yükseldikçe doğru tahmin sayısının azaldığını ve hatalı negatiflerin arttığını gösterir. Bu eğri, modelin optimal performans eşiğini belirlemede önemli bir araçtır.



Şekil 3.22. Eğitilen modelin test görüntülerinden tespit örnekleri

Şekil 3.22'deki görsel, bir UAV (insansız hava aracı) ile elde edilen görüntülerde insan tespiti yapan Yolov10 modelinin sonuçlarını gösteren bir dizi örnek içermektedir. Her bir kare, model tarafından "insan" olarak tanımlanan nesneleri mavi bir kutu içinde göstermekte ve tespit edilen her insan için modelin güven skoru (örneğin, "insan 0.9") belirtilmektedir. Görsellerdeki tespitler, genellikle açık alanlarda yapılan ve farklı mesafelerde çekilmiş görüntülerde insanları tanımlama başarısını göstermektedir. Güven skorları, modelin insanları tanımlama konusundaki güvenini yansıtır ve bu değerler 0.5 ile 0.9 arasında değişmektedir. Bu tür bir görsel, modelin sahadaki gerçek dünya performansını ve farklı koşullarda ne kadar doğru çalıştığını değerlendirmek için kullanılır.



Şekil 3.23. YOLOv10 Nesne tespiti modeli eğitim ve değerlendirme metrikleri

Şekil 3.23'deki grafikler, YOLOv10 modelin eğitim sırasında kayıpların azaldığını ve hassasiyet ile geri çağırma değerlerinin arttığını, bununla birlikte mAP (mean Average Precision) skorlarının iyileştiğini gözlemlemek mümkündür. Bu sonuçlar, modelin başarılı bir şekilde optimize edildiğini ve zamanla daha iyi performans gösterdiğini gösterir.

Çizelge 3.12'deki YOLOv10 algoritmasının 100 epoch boyunca 5 farklı katlama (fold) için elde edilen sonuçlar, modelin genel olarak iyi bir performans sergilediğini göstermektedir. Doğruluk oranı %87 ile %89 arasında değişirken, en yüksek doğruluk Fold_4'te %89 olarak kaydedilmiştir. Hassasiyet %90 ile %92 arasında değişmekte olup, bu durum modelin yanlış pozitif oranının düşük olduğunu göstermektedir. Duyarlılık ise tüm fold'lar için %96 olarak sabit kalmış, bu da modelin gerçek pozitifleri yakalama yeteneğinin oldukça yüksek olduğunu ortaya koymaktadır. F1 skoru %93 ile %94 arasında değişirken, bu da modelin hem hassasiyet hem de duyarlılık açısından dengeli bir performansa sahip olduğunu göstermektedir. mAP (mean Average Precision) değerleri %968 ile %974 arasında olup, nesne tespiti görevinde modelin başarılı olduğunu göstermektedir. Genel olarak, elde edilen veriler, YOLOv10'un etkili bir nesne tespiti modeli olduğunu ortaya koymaktadır.

Çizelge 3.12. YOLOv10 algoritmasının farklı katmanlar için performans değerleri

Algoritma	Fold	Doğruluk (Accuracy)	Hassasiyet (Precision)	Duyarlılık (Recall)	F1 Skoru	mAP50
YOLOv10	Fold_1	0.88	0.91	0.96	0.94	0.971
YOLOv10	Fold_2	0.88	0.91	0.96	0.94	0.970
YOLOv10	Fold_3	0.88	0.91	0.96	0.94	0.969
YOLOv10	Fold_4	0.89	0.92	0.96	0.94	0.974
YOLOv10	Fold_5	0.87	0.90	0.96	0.93	0.968



Şekil 3.24. Yolov10 modeline ait hatalı tespit örnekleri

Şekil 3.24’de Yolov10’un insan tespit sonuçları yer almakta. Soldaki görüntüde, toplamda dört insan bulunmakta, ancak biri kenarda kaldığı için algılanamamıştır. Diğer üç kişi, model tarafından güven düzeyi 0.9 ile başarıyla tanımlanmıştır. Sağdaki görüntüde ise, güven düzeyi 0.8 ile gösterilen iki kişi kucaklaşmakta. Model, bu iki kişiyi tek bir varlık olarak algılamış. Bu durum, nesne algılama algoritmalarının birbirine yakın konumda bulunan nesneleri ayırt etme konusunda zorluk yaşadığını göstermektedir. Kucaklaşma gibi durumların daha iyi tespit edilebilmesi için modelin ek eğitim alması gerekebilir. Sonuç olarak, bu görsel, modelin insan tespiti konusundaki doğruluk oranlarını ve potansiyel hatalarını değerlendirmek için önemli bir referans sunmaktadır.

3.14. Karşılaştırmalı Analiz

Çizelge 3.13. Farklı nesne tanıma modellerinin performans metrikleri karşılaştırması

Çalışma	Model	Doğruluk (Accuracy)	Hassasiyet (Precision)	Duyarlılık (Recall)	F1 Skoru	mAP50
[66]	YOLOv8	0.9678	-	-	-	-
[67]	YOLOv7	-	0.42	0.41	-	0.38
[68]	YOLOv7	-	0.772	0.927	-	0.865
[69]	YOLOv8n	-	0.974	0.952	0.962	0.986
[70]	YOLOv7	-	0.477	0.275	-	0.249
[71]	YOLOv7	-	0.788	0.714	-	0.737
[72]	YOLOv7	-	0.9392	0.7890	-	-
[73]	YOLOv7	-	0.9700	0.9680	-	-
[74]	YOLOv7	0.65	0.6875	0.8462	0.7586	-
[75]	YOLOv8n	-	0.844	0.817	0.83	0.887
Çalışmam	YOLOv7	0.828	0.914	0.896	0.904	0.726
Çalışmam	YOLOv8	0.896	0.926	0.970	0.946	0.974
Çalışmam	YOLOv10	0.880	0.910	0.960	0.938	0.970
[76]	YOLOv4	0.867	-	-	-	-
[76]	Faster R-CNN	0.862	-	-	-	-
[76]	EfficientDetD0	0.661	-	-	-	-
[76]	EfficientDetD4	0.918	-	-	-	-
[76]	EfficientDetD7	0.929	-	-	-	-
[77]	YOLOv7	0.924	-	-	-	-
[78]	Improved YOLOv7		0.902	0.833	-	0.881

Çizelge 3.13’de, farklı nesne tanıma modellerinin performans metrikleri karşılaştırılmaktadır. Bu karşılaştırmada doğruluk, hassasiyet, duyarlılık, F1 skoru ve mAP50 değerleri sunulmuştur. Bu çizelge kullanılan modellerin etkinliğini ve performansını değerlendirerek, hangi modelin belirli görevlerde daha başarılı olduğunu anlamaya yardımcı olmaktadır.

4. SONUÇ VE ÖNERİLER

Çizelge 4. 1. YOLO algoritmalarının performans karşılaştırması

Algoritma	Doğruluk (Accuracy)	Hassasiyet (Precision)	Duyarlılık (Recall)	F1 Skoru	mAP50
YOLOv7	0.828	0.914	0.896	0.904	0.726
YOLOv8	0.896	0.926	0.970	0.946	0.974
YOLOv10	0.880	0.910	0.960	0.938	0.970

Çizelge 4.1’de yer alan değerler, k-5 fold çapraz doğrulama sonuçlarının ortalaması alınarak hesaplanmıştır. Bu bağlamda, YOLOv8 algoritması en yüksek genel performansı sergileyerek, doğruluk (0,896), hassasiyet (0,926), duyarlılık (0,970) ve F1 skoru (0,946) değerleri ile dikkat çekmektedir. YOLOv10, yüksek duyarlılığı (0,960) ile öne çıksa da diğer metriklerde YOLOv8’in gerisinde kalmaktadır. YOLOv7 ise tüm metriklerde en düşük değerlere sahip olup, bu durum onun diğer iki algoritmaya göre daha az etkili olduğunu göstermektedir. Ayrıca, mAP50 değerleri de benzer bir trend izleyerek, YOLOv8’in 0,974 ile en yüksek başarıyı elde ettiğini, YOLOv10’un ise 0,970 ile onu takip ettiğini ortaya koymaktadır. Sonuç olarak, k-5 fold doğrulama ile elde edilen bu değerler, YOLOv8’in en güçlü seçenek olduğunu göstermektedir.

Bu çalışma, YOLOv7, YOLOv8 ve YOLOv10 gibi farklı derin öğrenme tabanlı nesne tespit modellerinin performansını karşılaştırmıştır. Bu yönüyle güncel literatüre önemli bir katkı sağlamaktadır. Çalışmanın sonuçları, YOLO modelleriyle eğitilen modellerin, özellikle YOLOv8’in, gerçek dünya senaryolarında daha yüksek doğrulukla nesne tespiti yapabildiğini göstermiştir. Gelecekte, bu çalışmanın kapsamı genişletilebilir. Örneğin, daha büyük bir veri seti kullanılabilir veya farklı derin öğrenme modelleri eğitilerek karşılaştırılabilir. Ayrıca, modellerin performansı, farklı nesne türleri ve pozisyonları üzerinde ayrıntılı olarak incelenebilir. Geliştirilen modellerin gerçek zamanlı İHA uygulamalarında test edilmesi, pratik senaryolardaki performanslarını değerlendirmek için önemlidir. Bu çalışmalar, derin öğrenme tabanlı nesne tespit modellerinin geliştirilmesine katkıda bulunabilir.

KAYNAKLAR

- [1] N. Yeşilkaya, “Yapay Zekâda Etik Sorunlar”, *e-Şarkiyat İlmi Araştırmaları Dergisi/Journal of Oriental Scientific Research (JOSR)*, vol. 14, sy 3, ss. 949-963, Ara. 2022, doi: 10.26791/sarkiat.1189864.
- [2] H. Mokayed, T. Z. Quan, L. Alkhaled, and V. Sivakumar, “Real-Time Human Detection and Counting System Using Deep Learning Computer Vision Techniques”, *Artificial Intelligence and Applications*, vol. 1, no. 4, pp. 221-229, 2023, doi: 10.47852/bonviewAIA2202391.
- [3] B. Erdil, “İnsansız Hava Araçlarının Kullanım Alanları ile Bu Araçların Türkiye’nin Yurtdışı Operasyonlarındaki Yeri ve Önemi”, *Bölgesel Araştırmalar Dergisi*, c. 5, sy 2, ss. 581-607, Ara. 2021.
- [4] T. Gündüz, “Derinlik kamerası ile derin öğrenme tabanlı nesne tespiti”, Yüksek Lisans Tezi, Mühendislik ve Fen Bilimleri Enstitüsü, İskenderun Teknik Üniversitesi, Hatay, Türkiye, 2023.
- [5] B. E. Demiröz, “Person Detection and Tracking Using Omnidirectional Cameras, And Rectangle Blanket Problem” Doktora Tezi, Fen Bilimleri Enstitüsü, Boğaziçi Üniversitesi, İstanbul, Türkiye, 2019.
- [6] P. Kumar, S. Batchu, N. Swamy S., and S. R. Kota, “Real-time concrete damage detection using deep learning for high rise structures”, *IEEE Access*, no. 9, pp. 112312-112331, 2021, doi: 10.1109/ACCESS.2021.3102647.
- [7] B. Yılmaz, “İnsan tespitinde yapay veri artırımı”, Yüksek Lisans Tezi, Fen Bilimleri Enstitüsü, Yıldız Teknik Üniversitesi, İstanbul, Türkiye, 2016.
- [8] E. Şimşek, B. Özyer, ve G. T. Özyer, “Deep Learning Based Human Detection in Camera-trap Images”, *Bayburt Üniversitesi Fen Bilimleri Dergisi*, no. 3, s.1, 2020.
- [9] E. Şimşek, B. Özyer, ve G. Tümöklü Özyer, “Fotokapan Görüntülerinde Yerel Öznitelikler ile Nesne Tespiti”, *Gümüşhane Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, vol.9, no. 4, pp. 633-644, Eki. 2019, doi: 10.17714/gumusfenbil.510717.

- [10] K. Hanbay ve H. Üzen, “Nesne tespit ve takip metotları: Kapsamlı bir derleme”, *Türk Doğa ve Fen Dergisi*, c. 6, sy 2, 2017.
- [11] T. Abdellatif, M. A. Sedrine, and Y. Gacha, “DroMOD: A Drone-Based Multi-Scope Object Detection System”, *IEEE Access*, no. 11, pp. 26652-26666, 2023, doi: 10.1109/ACCESS.2023.3253767.
- [12] X. Chen, W. Yang, S. Zeng, L. Geng, and Y. Jiao, “YOLOAL: Focusing on the Object Location for Detection on Drone Imagery”, *IEEE Access*, no. 11, pp. 128886-128897, 2023, doi: 10.1109/ACCESS.2023.3332815.
- [13] G. Gol ve E. Kiyak, “İnsansız hava araçları kullanılarak araç sayısı tespiti”, içinde *2014 22nd Signal Processing and Communications Applications Conference, SIU 2014 - Proceedings*, 2014. doi: 10.1109/SIU.2014.6830305.
- [14] F. H. K. Zaman, N. M. Tahir, Y. M. Yusoff, N. M. Thamrin, and A. H. Hasmi, “Human Detection from Drone using You Only Look Once (YOLOv5) for Search and Rescue Operation”, *Journal of Advanced Research in Applied Sciences and Engineering Technology*, no. 30, pp 3, 2023, doi: 10.37934/araset.30.3.222235.
- [15] A. Georgiou, P. Masters, S. Johnson, and L. Feetham, “UAV- assisted real- time evidence detection in outdoor crime scene investigations”, *J Forensic Sci*, vol. 67, no. 3, pp. 1221-1232, May. 2022, doi: 10.1111/1556-4029.15009.
- [16] K. Jiang *et al.*, “An Attention Mechanism-Improved YOLOv7 Object Detection Algorithm for Hemp Duck Count Estimation”, *Agriculture (Switzerland)*, vol. 12, no. 10, 2022, doi: 10.3390/agriculture12101659.
- [17] N. Can ve M. Kahveci, “İnsansız Hava Araçları: Tarihçesi, Tanımı, Dünyada ve Türkiye Deki Yasal Durumu”, *Selcuk University Journal of Engineering ,Science and Technology*, c. 5, sy 4, ss. 511-535, Ara. 2017, doi: 10.15317/scitech.2017.109.
- [18] O. Villi ve M. Yakar, “İnsansız Hava Araçlarının Kullanım Alanları ve Sensör Tipleri”, *Türkiye İnsansız Hava Araçları Dergisi*, c. 4, sy 2, ss. 73-100, Ara. 2022, doi: 10.51534/tiha.1189263.

- [19] Ç. Çiçekdemir, “Döner kanatlı İHA otonom iniş denetleme yazılımının geliştirilmesi”, Yüksek Lisans Tezi, Fen Bilimleri Enstitüsü, Bilecik Şeyh Edebali Üniversitesi, Bilecik, Türkiye, 2019.
- [20] B. Kaycı, “Güneş panellerinin dört rotorlu iha kullanılarak termografi yöntemiyle derin öğrenme tabanlı hata tespit ve teşhisi”, Yüksek Lisans Tezi, Lisansüstü Eğitim Enstitüsü, Karabük Üniversitesi, Karabük, Türkiye, 2021.
- [21] “Drone Nedir ve Nasıl Uçar?” Erişim: 10 Ağustos 2024. [Çevrimiçi]. Erişim adresi: <https://www.emrehaninci.com/drone-nedir-ve-nasil-ucar/>
- [22] E. Balta, “Dört rotorlu bir insansız hava aracının otonom uçuşu için optimum kontrolcü tasarımı”, Yüksek Lisans Tezi, Fen Bilimleri Enstitüsü, Necmettin Erbakan Üniversitesi, Konya, Türkiye, 2023.
- [23] “İnsansız Hava Araçları”. Erişim: 10 Ağustos 2024. [Çevrimiçi]. Erişim adresi: <https://bilimgenc.tubitak.gov.tr/bunu-biliyor-muydunuz/insansiz-hava-araclari>
- [24] S. Murat, “İnsansız hava aracı görüntülerinden derin öğrenme yöntemleriyle nesne tanıma”, Yüksek Lisans Tezi, Lisansüstü Eğitim Enstitüsü, Maltepe Üniversitesi, İstanbul, Türkiye, 2021.
- [25] A. E. Aytan, Y. Öztürk, ve E. K. Örgen, “Görüntü İşleme”, *İ.Ü. Diş Hekimliği Fakültesi Dergisi*, c. 27, sy 4, 1993.
- [26] C. Küpeli ve F. Bulut, “Görüntüdeki Tuz Biber ve Gauss Gürültülerine Karşı Filtrelerin Performans Analizleri”, *Haliç Üniversitesi Fen Bilimleri Dergisi*, c. 3, sy 2, ss. 211-239, Eyl. 2020, doi: 10.46373/hafebid.768240.
- [27] İ. Çayiroğlu, “Görüntü İşleme Ders Notları - 5. Hafta”, 2024. Erişim: 09 Ağustos 2024. [Çevrimiçi]. Erişim adresi: https://ibrahimcayiroglu.com/Dokumanlar/GoruntuIsleme/Goruntu_Isleme_Ders_Notlari-5.Hafta.pdf.
- [28] A. Değirmenci, İ. Çankaya, ve R. Demirci, “Gradyan Anahtarlama Gauss Görüntü Filtresi”, *Düzce Üniversitesi Bilim ve Teknoloji Dergisi*, c. 6, sy 1, ss. 196-215, Oca. 2018, doi: 10.29130/dubited.345116.
- [29] R. B. Bayram ve E. Yılmaz, “Gömülü Sistem Tabanlı Bir Hatalı Ürün Tespit Sistemi”, *Uludağ University Journal of The Faculty of Engineering*, c. 24, sy 1, 2019, doi: 10.17482/uumfd.525696.

- [30] M. Kılınç ve H. Gözde, “Termal görüntülerdeki gömülü plastik anti-personel mayınlarının dairesel hough dönüşümü destekli aktif termografi yöntemi ile tespiti”, *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*, c. 35, sy 2, 2019, doi: 10.17341/gazimmfd.507225.
- [31] J. Terven, D.-M. Córdova-Esparza, and J.-A. Romero-González, “A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS”, *Mach Learn Knowl Extr*, vol. 5, no. 4, pp. 1680-1716, Kas. 2023, doi: 10.3390/make5040083.
- [32] A. Eker ve N. Duru, “Medikal Görüntü İşlemede Derin Öğrenme Uygulamaları”, *Acta Infologica*, c. 5, sy 2, ss. 459-474, 2021, doi: 10.26650/acin.927561.
- [33] S. Uçkun, M. Ağralı, and V. Kılıç, “Deep Learning-Based Ischemic Stroke Segmentation on Brain Computed Tomography Images”, *European Journal of Science and Technology*, no. 50, pp. 105-112, 2023, doi: 10.31590/ejosat.1258247.
- [34] A. Kabadayi ve M. Uysal, “İnsansız Hava Aracı ile Elde Edilen Verilerden Binaların Tespiti”, *Türkiye İnsansız Hava Araçları Dergisi*, c. 1, sy 1, ss. 08-14, 2019.
- [35] G. E. Güraksın, “Tuz Biber Gürültülerinin Giderilmesi için k-Ortalama Algoritması Tabanlı Filtre Tasarımı”, *Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, c. 22, sy 2, 2018, doi: 10.19113/sdufbed.59275.
- [36] C. Küpeli ve F. Bulut, “Görüntüdeki Tuz Biber ve Gauss Gürültülerine Karşı Filtrelerin Performans Analizleri”, *Haliç Üniversitesi Fen Bilimleri Dergisi*, c. 3, sy 2, 2020, doi: 10.46373/hafebid.768240.
- [37] Ö. Akar, “Göktürk-2 ve Worldview-2 Uydu Görüntüleri için Görüntü Keskinleştirme Yöntemlerinin Değerlendirilmesi”, *Erzincan Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, c. 12, sy 2, 2019, doi: 10.18185/erzifbed.495854.
- [38] S. Asghar *et al.*, “Water Classification Using Convolutional Neural Network”, *IEEE Access*, vol. 11, pp. 78601-78612, 2023, doi: 10.1109/ACCESS.2023.3298061.
- [39] “Unsharp Masking”. Erişim: 10 Ağustos 2024. [Çevrimiçi]. Erişim adresi: <https://www.pngwing.com/en/search?q=Unsharp>

- [40] X. Li, G. Niu, X. Li, X. Wang, and G. Shi, “Correct and Crisp Edge Detection Approach Based on Dense Network”, *IEEE Access*, vol. 11, pp. 20940-20951, 2023, doi: 10.1109/ACCESS.2021.3089597.
- [41] F. M. Şenalp ve M. Ceylan, “Düşük Çözünürlüklü Termal Yüz Görüntü Çözünürlüğünün Derin Öğrenme ile Artırılması”, *Avrupa Bilim ve Teknoloji Dergisi*, sy Special, ss. 131-135, Eki. 2020, doi: 10.31590/ejosat.802174.
- [42] C. Zhang and C. Berger, “Pedestrian Behavior Prediction Using Deep Learning Methods for Urban Scenarios: A Review”, *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 10, pp. 10279-10301, Eki. 2023, doi: 10.1109/TITS.2023.3281393.
- [43] A. Yılmaz ve U. Kaya, *Derin Öğrenme*, Kodlab. İstanbul, 2021.
- [44] P. Khan *et al.*, “Machine Learning and Deep Learning Approaches for Brain Disease Diagnosis: Principles and Recent Advances”, *IEEE Access*, vol. 9, 2021, doi: 10.1109/ACCESS.2021.3062484.
- [45] “Sinir Hücresi Yapısı”. Erişim: 10 Ağustos 2024. [Çevrimiçi]. Erişim adresi: https://tr.wikipedia.org/wiki/Dosya:Neuron_Hand-tuned.svg
- [46] F. Gökmen, “Yapay sinir ağları kullanılarak öğütücülerde gürültü ve titreşim analizi”, Yüksek Lisans Tezi, Fen Bilimleri Enstitüsü, Niğde Üniversitesi, Niğde, Türkiye, 2010.
- [47] Q. Luo and J. Yang, “The Artificial Intelligence and Neural Network in Teaching”, *Comput Intell Neurosci*, vol. 2022, pp. 1-11, Haz. 2022, doi: 10.1155/2022/1778562.
- [48] S. Dikici ve V. Altuntaş, “Yapay Sinir Ağları Kullanılarak Protein Katlanması Tanıma”, *Bilişim Teknolojileri Dergisi*, c. 16, sy 2, ss. 95-105, Nis. 2023, doi: 10.17671/gazibtd.1141468.
- [49] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection”, içinde *IEEE Conference on Computer Vision and Pattern Recognition*, IEEE Computer Society, Ara. 2016, pp. 779-788. doi: 10.1109/CVPR.2016.91.

- [50] B. Özcan ve H. Bakır, “Yapay Zekâ Destekli Beyin Görüntüleri Üzerinde Tümör Tespiti”, *International Conference on Pioneer and Innovative Studies*, c. 1, 2023, doi: 10.59287/icpis.847.
- [51] S. Altun ve M. F. Talu, “Derin Sinir Ağları için Hiperparametre Metodlarının ve Kitlerinin İncelenmesi”, *DÜMF Mühendislik Dergisi*, c. 12, sy 2, ss. 187-199, Mar. 2021.
- [52] R. Anita Jasmine, P. Arockia Jansi Rani, and J.Ashley Dhas, “Hyper Parameters Optimization for Effective Brain Tumor Segmentation with YOLO Deep Learning”, *J Pharm Negat Results*, vol. 13, no. 6, pp. 2247-2257, Eki. 2022, doi: 10.47750/pnr.2022.13.S06.292.
- [53] “CIFAR-10 veri setinden örnek”. Erişim: 13 Ağustos 2024. [Çevrimiçi]. Erişim adresi: <https://www.cs.toronto.edu/~kriz/cifar.html>
- [54] W. xue, “Redefining Audience Participation: The Immersive Interaction in ‘Wolf and Spices VR’ and the Enlightenment of Journalism and Communication”, *Journal of Multimedia Information System*, vol. 11, no. 1, pp. 35-44, Mar. 2024, doi: 10.33851/JMIS.2024.11.1.35.
- [55] S. Xu, R. Wang, W. Shi, and X. Wang, “Classification of Tree Species in Transmission Line Corridors Based on YOLO v7”, *Forests*, vol. 15, no. 1, s. 61, Ara. 2023, doi: 10.3390/f15010061.
- [56] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," *Proceedings of the 30th International Conference on Machine Learning (ICML)*, Atlanta, Georgia, USA, 2013
- [57] “Inference Time ”. Erişim: 13 Ağustos 2024. [Çevrimiçi]. Erişim adresi: <https://docs.ultralytics.com/tr/models/yolov7/>
- [58] “Yolov7 Modelleri İçin Performans Karşılaştırması”. Erişim: 10 Ağustos 2024. [Çevrimiçi]. Erişim adresi: <https://github.com/YisrealHung/WongKinYiu-yolov7>
- [59] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, “YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern*

- Recognition (CVPR)*, Jul. 2023, pp. 7464-7475, doi: 10.1109/cvpr52729.2023.00721.
- [60] J. Terven, D.-M. Córdova-Esparza, and J.-A. Romero-González, “A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS”, *Mach Learn Knowl Extr*, vol. 5, no. 4, pp. 1680-1716, Kas. 2023, doi: 10.3390/make5040083.
- [61] M. Sevi and İ. Aydın, “Detection of Foreign Objects Around the Railway Line with YOLOv8”, *Journal of Computer Science*, vol. IDAP-2023, pp. 19-23, Ağu. 2023, doi: 10.53070/bbd.1346317.
- [62] B. Xiao, M. Nguyen, and W. Q. Yan, “Fruit ripeness identification using YOLOv8 model”, *Multimed Tools Appl*, vol. 83, no. 9, 2024, doi: 10.1007/s11042-023-16570-9.
- [63] “YOLOv8”. Erişim: 12 Mart 2024. [Çevrimiçi]. Erişim adresi: <https://github.com/ultralytics/ultralytics?tab=readme-ov-file>
- [64] A. Wang *et al.*, “YOLOv10: Real-Time End-to-End Object Detection”, May. 2024, [Çevrimiçi]. Erişim adresi: <http://arxiv.org/abs/2405.14458>
- [65] “K-Fold Cross Validation”. Erişim: 15 Temmuz 2024. [Çevrimiçi]. Erişim adresi: <https://docs.ultralytics.com/guides/kfold-cross-validation/>
- [66] M. Q. F. Alsamurai, “Development and implementation of yolov8-based model for human and animal detection during forest fires”, Yüksek Lisans Tezi, Lisansüstü Eğitim Enstitüsü, Altınbaş Üniversitesi, İstanbul, Türkiye, 2023.
- [67] S. Ersan, “YOLOv7 algoritması ile diş röntgenlerinde lezyon ve çürük tespiti”, Yüksek Lisans Tezi, Fen Bilimleri Enstitüsü, Dicle Üniversitesi, Diyarbakır, Türkiye, 2024.
- [68] O. E. Ogunboye, “YOLOv7 kullanarak bir yol trafik algılamasını incelemek”, Yüksek Lisans Tezi, Fen Bilimleri Enstitüsü, Üsküdar Üniversitesi, İstanbul, Türkiye, 2023.
- [69] M. T. Çelik, “YOLOv8 nesne tespit modeli ile plastik parça yüzey kusurlarının gerçek zamanlı tespit edilmesi”, Yüksek Lisans Tezi, Lisansüstü Eğitim Enstitüsü, Bursa Teknik Üniversitesi, Bursa, Türkiye, 2024.

- [70] F. Tang, F. Yang, and X. Tian, "Long-Distance Person Detection Based on YOLOv7," *Electronics (Basel)*, vol. 12, no. 6, p. 1502, Mar. 2023, doi: 10.3390/electronics12061502.
- [71] L. A. Silva, V. R. Q. Leithardt, V. F. L. Batista, G. Villarrubia González, and J. F. De Paz Santana, "Automated Road Damage Detection Using UAV Images and Deep Learning Techniques," *IEEE Access*, vol. 11, pp. 62918-62931, 2023, doi: 10.1109/ACCESS.2023.3287770.
- [72] W. Zhang et al., "LS-YOLO: A Novel Model for Detecting Multiscale Landslides With Remote Sensing Images," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 17, pp. 4952-4965, 2024, doi: 10.1109/JSTARS.2024.3363160.
- [73] R. Mahadshetti, J. Kim, and T.-W. Um, "Sign-YOLO: Traffic Sign Detection Using Attention-based YOLOv7," *IEEE Access*, pp. 1-1, 2024, doi: 10.1109/ACCESS.2024.3417023.
- [74] S. Aulia, A. B. Suksmono, T. R. Mengko, and B. Alisjahbana, "A Novel Digitized Microscopic Images of ZN-Stained Sputum Smear and Its Classification Based on IUATLD Grades", *IEEE Access*, vol. 12, pp. 51364-51380, 2024, doi: 10.1109/ACCESS.2024.3386208.
- [75] T.-N. Pham, V.-H. Nguyen, K.-R. Kwon, J.-H. Kim, and J.-H. Huh, "Improved YOLOv5 Based Deep Learning System for Jellyfish Detection", *IEEE Access*, vol. 12, pp. 87838-87849, 2024, doi: 10.1109/ACCESS.2024.3405452.
- [76] N. Aldahoul, H. A. Karim, A. Q. Md. Sabri, M. J. T. Tan, Mhd. A. Momo, and J. L. Fermin, "A Comparison Between Various Human Detectors and CNN-Based Feature Extractors for Human Activity Recognition via Aerial Captured Video Sequences", *IEEE Access*, vol. 10, pp. 63532-63553, 2022, doi: 10.1109/ACCESS.2022.3182315.
- [77] M. Rukaiya, M. Muhtadee Faiaz, and K. Soumik, "Multimodal Approach to Human Detection in Unconstrained Environments using YOLOV7 for Conventional, Infrared & Thermal Cameras", ,Brac University, 2023.

- [78] Z. Bowen, L. Huacai, Z. Shengbo, C. Xinqiang, and X. Hongwei, “Night target detection algorithm based on improved YOLOv7”, *Sci Rep*, vol. 14, no. 1, pp. 15771, Tem. 2024, doi: 10.1038/s41598-024-66842-z.

