T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

GÖRÜNTÜ İŞLEME VE MAKİNE ÖĞRENİMİ YÖNTEMLERİYLE DRONE TESPİTİ

Batuhan BAYSAL

YÜKSEK LISANS TEZİ
Elektronik ve Haberleşme Mühendisliği Anabilim Dalı
Haberleşme Programı

Danışman
Doç. Dr. Bülent BOLAT

Temmuz, 2025

T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

GÖRÜNTÜ İŞLEME VE MAKİNE ÖĞRENİMİ YÖNTEMLERİYLE DRONE TESPİTİ

Batuhan BAYSAL tarafından hazırlanan tez çalışması 08.07.2025 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Elektronik ve Haberleşme Mühendisliği Anabilim Dalı Haberleşme Programı **YÜKSEK LISANS TEZİ** olarak kabul edilmiştir.

Doç. Dr. Bülent BOLAT
Yıldız Teknik Üniversitesi
Danışman

Jüri Üyeleri

Doç. Dr. Bülent BOLAT, Danışman
Yıldız Teknik Üniversitesi _____

Prof. Dr. Ufuk SAKARYA, Üye
Yıldız Teknik Üniversitesi _____

Dr. Öğr. Üyesi Nurullah ÇALIK, Üye
İstanbul Medeniyet Üniversitesi _____

Danışmanım Doç. Dr. Bülent BOLAT sorumluluğunda tarafımca hazırlanan GÖRÜNTÜ İŞLEME VE MAKİNE ÖĞRENİMİ YÖNTEMLERİYLE DRONE TESPİTİ başlıklı çalışmada veri toplama ve veri kullanımında gerekli yasal izinleri aldığımı, diğer kaynaklardan aldığım bilgileri ana metin ve referanslarda eksiksiz gösterdiğimi, araştırma verilerine ve sonuçlarına ilişkin çarpıtma ve/veya sahtecilik yapmadığımı, çalışmam süresince bilimsel araştırma ve etik ilkelerine uygun davrandığımı beyan ederim. Beyanımın aksinin ispatı halinde her türlü yasal sonucu kabul ederim.

Batuhan BAYSAL

İmza

Bu çalışma, “Yıldız Teknik Üniversitesi Bilimsel Araştırma Proje Koordinatörlüğü’nün 1234A” numaralı projesi ile desteklenmiştir.

Dedicated to my family
and my best friend

TEŞEKKÜR

Bu tezin hazırlanmasında büyük emek ve katkıları olan danışman hocam Doç. Dr. Bülent BOLAT'a en içten teşekkürlerimi sunarım. Kendisi, çalışma boyunca bilgi birikimi, yol göstericiliği ve sabrıyla her aşamada bana rehberlik etti. Onun desteği ve teşvikleri olmadan bu çalışmayı tamamlamak mümkün olmazdı.

Hayatım boyunca her adımımda yanımda olan, sevgileri ve fedakârlıklarıyla beni destekleyen eşim Aleyna BAYSAL'a da sonsuz minnettarlığımı ifade etmek isterim. Özellikle sağladığı manevi güç, karşılaştığım zorluklarla baş etmemde en önemli dayanağım olmuştur. Onun inancı ve desteği, bu yolculuğu başarıyla tamamlamamı sağladı.

Ayrıca, çalışmalarım sırasında moral kaynağı olan ve motive edici sözleriyle beni yüreklendiren kıymetli arkadaşlarıma teşekkür ederim. Adlarını anmasam da, katkıları ve varlıkları benim için her zaman çok kıymetlidir. Sağladıkları destek, çalışmalarımı sürdürürken karşılaştığım engelleri aşmamda önemli bir rol oynamıştır.

Son olarak, bu tezin gerçekleşmesine katkıda bulunan, beni cesaretlendiren ve destekleyen herkese minnettarım. İster doğrudan, ister dolaylı şekilde yardım eden herkesin emekleri bu çalışmayı daha anlamlı kılmıştır.

Batuhan BAYSAL

İÇİNDEKİLER

SİMGE LİSTESİ viii

KISALTMA LİSTESİ ix

ŞEKİL LİSTESİ x

TABLO LİSTESİ xi

ÖZET xii

ABSTRACT xiv

1 GİRİŞ 1

1.1 Literatür Özeti 1

1.2 Tezin Amacı 2

1.3 Hipotez 3

2 DERİN ÖĞRENME İLE GÖRÜNTÜ İŞLEME YÖNTEMLERİ 4

2.1 Klasik Yöntemler İle Nesne Tespiti 4

2.1.1 Veri Girişi 5

2.1.2 Ön işleme 5

2.1.3 Öz Nitelik Çıkarma 6

2.1.4 Tanımlama 7

2.2 Derin Öğrenme ile Görüntü işleme 8

2.3 Çalışmada Kullanılan Ana Nesne Tespit Yöntemi 11

3 YOLO AİLESİ VE NESNE TESPİTİNDE KULLANILAN VERİ SETLERİ 12

3.1 Yolo Ailesi 12

3.1.1 Orijinal YOLO (You Only Look Once) Algoritması 12

3.1.2 YOLOv3 14

3.1.3 YOLOv5 16

3.1.4 YOLOv8 17

3.2 Nesne Tespitinde Kullanılan Veri Setleri 18

4 KULLANILAN VERİ SETLERİ VE KLASİK YÖNTEMLERLE DESTEKLENMİŞ YOLOv8 İLE EĞİTİLMESİ **20**
4.1 Veri Setleri ve Etiketleme Süreçleri 20
4.1.1 Kaggle Veri Seti 20
4.1.2 Toplanan Video Verileri 20
4.1.3 Verinin Etiketlenmesi 21
4.2 Eğitim Süreci ve Sonuçları 24
4.3 Test Süreci ve Sonuçları 30
4.3.1 Görüntü Bazlı Test 30
4.3.2 Video Bazlı Test 31

5 SONUÇ **34**
5.1 Karşılaştırmalı Sonuç Analizi 34
5.2 Uygulama ve İyileştirme Önerileri 35

KAYNAKÇA **38**

TEZDEN ÜRETİLMİŞ YAYINLAR **42**

SİMGE LİSTESİ

Ω	Angular Velocity (Açısal Hız)
IoU	Intersection over Union (Birleşim Üzerinde Kesişim)
λ	Learning Rate (Öğrenme Oranı)
Δt	Time Interval (Zaman Aralığı)

KISALTMA LİSTESİ

CVAT	Computer Vision Annotation Tool
HOG	Histogram of Oriented Gradients
IoU	Intersection over Union
SIFT	Scale-Invariant Feature Transform
SURF	Speeded Up Robust Features
SVM	Support Vector Machine
YOLO	You Only Look Once

ŞEKİL LİSTESİ

Şekil 2.1 Klasik nesne tespiti aşamaları[9]. 5
Şekil 2.2 Contrast Limited Adaptive Histogram Equalization (CLAHE) uygulaması[11]. 6
Şekil 2.3 Histogram of Oriented Gradients (HOG) algoritması ile çıkarılan öznitelikler[16]. 7
Şekil 2.4 Farklı nesnelerin sınıflandırılması ve her bir nesneye ait sınır kutularının (bounding boxes) gösterilmesi[20]. 9
Şekil 3.1 Grid hücreleri fikrini temsil eden örnek bir görsel [16] 13
Şekil 3.2 YOLOv3 Modelinde Grid Hücreleri ve Anchor Kutularının Yapısı[36] 16
Şekil 4.1 Çalışmada kullanılan veri setlerinden örnek görseller. 21
Şekil 4.2 Etiketlenmesi gereken görev ve iş seçildikten sonra CVAT ana ekranı görünümü. 23
Şekil 4.3 Etiketlendikten sonra CVAT ana ekranı. 23
Şekil 4.4 Etiketleme işlemi sonrasında oluşturulan örnek metin dosyası formatı. 24
Şekil 4.5 Eğitim Sonuçları (Train) 25
Şekil 4.6 Eğitim Sonuçları (Validation) 25
Şekil 4.7 Eğitim Sonuçları (Precision ve Recall) 26
Şekil 4.8 Eğitim Sonuçları (mAP50 ve mAP50-95) 26
Şekil 4.9 Eğitim Sonuçları (Confusion Matrix) 28
Şekil 4.10 Takip entegrasyonu devre dışı kaldığı durumlarda, piksel değişim analizi sayesinde takip süreci devam etmektedir. 29
Şekil 4.11 Test sonuçları (Confusion Matrix) 31
Şekil 4.12 Test sonuçları (Confusion Matrix) 32
Şekil 5.1 Klasik Anti-Drone Sistemleri[44]. 36
Şekil 5.2 Mimarinin Entegre Edilmesi Önerilen Sistem[45] 36

TABLO LİSTESİ

Tablo 5.1 Tablo 5.1 Karşılaştırmalı Sonuç Analizi 35

ÖZET

GÖRÜNTÜ İŞLEME VE MAKİNE ÖĞRENİMİ YÖNTEMLERİYLE DRONE TESPİTİ

Batuhan BAYSAL

Elektronik ve Haberleşme Mühendisliği Anabilim Dalı
Yüksek Lisans Tezi

Danışman: Doç. Dr. Bülent BOLAT

Son yıllarda dronların sivil ve askeri alanlardaki kullanımının hızla arttığı gözlemlenmektedir. Özellikle güvenlik, gözlem ve lojistik gibi birçok farklı alanda yaygınlaşan dronlar, hem avantajları hem de potansiyel tehditleriyle dikkat çekmektedir. Yetkisiz dronların havaalanları, askeri tesisler, kamu binaları, askeri konvoylar ve özel mülkler gibi kritik bölgeler için ciddi bir güvenlik riski oluşturduğu bilinmektedir. Bu tehditlerle etkili bir şekilde mücadele edebilmek için güvenilir, hızlı ve gerçek zamanlı dron tespit sistemlerine ihtiyaç duyulmaktadır.

Bu çalışmada, derin öğrenme tabanlı bir nesne tespiti mimarisi olan YOLOv8 kullanılarak etkili bir drone tespit sistemi geliştirilmiştir. YOLOv8, önceki versiyonlarına (YOLOv3 ve YOLOv5) göre daha yüksek hız ve doğruluk oranları sunarak, nesne tespiti problemlerinde önemli bir avantaj sağlamaktadır. Çalışmada kullanılan veri seti, Kaggle ve diğer açık kaynaklardan toplanan, 1300 adet drone ve kuş görüntüsünden oluşmaktadır. Bu görüntüler manuel olarak etiketlenmiş ve rasgele olarak %10'u test için ayrıldıktan sonra modelin eğitiminde kullanılmak üzere yine rastgele olarak %80 eğitim, %20 doğrulama (validasyon) şeklinde ikiye ayrılmıştır. Eğitim sürecinde YOLOv8'in farklı kayıp fonksiyonları ve hiperparametreleri optimize edilmiştir. Sonuç olarak, modelin farklı boyut ve şekillerdeki dronları başarıyla tespit ettiği gözlemlenmiştir.

Geliştirilen modelin performansı, saniyede 25-30 kare hızında çalışarak gerçek zamanlı video akışlarında test edilmiştir. Elde edilen sonuçlar, %95'in üzerinde bir doğruluk oranına ulaşarak dronları kuşlardan ayırt ettiği ve bu sistemin pratik

uygulamalarda kullanılabilirliğini kanıtlamaktadır. Modelin, son dönemde özellikle savaş bölgelerinde ortaya çıkan asimetrik dron saldırılarına karşı etkili bir savunma mekanizması olarak kullanılabileceği de düşünülmektedir.

Anahtar Kelimeler: Dron tespiti, derin öğrenme, YOLOv8, gerçek zamanlı nesne tespiti, güvenlik uygulamaları, asimetrik saldırılar.

ABSTRACT

DRONE DETECTION WITH IMAGE PROCESSING AND MACHINE LEARNING METHODS

Batuhan BAYSAL

Department of Electronics and Communication Engineering
Master of Science Thesis

Supervisor: Assoc. Prof. Bülent BOLAT

In recent years, the use of drones in civilian and military fields has been rapidly increasing. Especially becoming widespread in many different areas such as security, surveillance, and logistics, drones attract attention with both their advantages and potential threats. Unauthorized drones are known to pose a serious security risk to critical areas such as airports, military facilities, public buildings, military convoys, and private properties. To effectively combat these threats, reliable, fast, and real-time drone detection systems are required.

In this study, an effective drone detection system was developed using YOLOv8, a deep learning-based object detection architecture. YOLOv8, compared to its previous versions (YOLOv3 and YOLOv5), provides higher speed and accuracy rates, offering a significant advantage in solving object detection problems. The dataset used in the study consists of 1300 drone and bird images collected from Kaggle and other open sources. These images were manually labeled and randomly split, with 10% set aside for testing. The remaining data was further divided into 80% training and 20% validation for model training. During the training process, different loss functions and hyperparameters of YOLOv8 were optimized. As a result, it was observed that the model successfully detected drones of different sizes and shapes.

The performance of the developed model was tested on real-time video streams, operating at a frame rate of 25-30 frames per second. The obtained results reached

an accuracy rate of over 95%, proving that the system can distinguish drones from birds and can be utilized in practical applications. It is also considered that the model could serve as an effective defense mechanism against asymmetric drone attacks, especially those emerging in recent times in conflict zones.

Keywords: Drone detection, deep learning, YOLOv8, real-time object detection, security applications, asymmetric attacks.

1
GİRİŞ

1.1 Literatür Özeti

Gelişen donanım teknolojisi sayesinde derin öğrenme ve yapay zeka tabanlı sistemler, nesne tanıma ve sınıflandırma[1], anlamsal segmentasyon, nesne tespiti ve takibi[2] [3] gibi birçok karmaşık bilgisayarla görüntü işleme görevini çözmek için kullanılan tatmin edici sonuçlar sağlamaktadır. Ayrıca girdi verileri sadece normal kamera görüntüsü değil aynı zamanda kızılötesi görüntülerde olabilir[4]. Derin öğrenme tabanlı yaklaşımlar, klasik makine öğrenimi yöntemlerine kıyasla daha iyi bir performans sergilediğinden, pek çok alanda yaygın şekilde kullanılmaktadır. Birden fazla işlem katmanına sahip olan derin öğrenme modellerinin, temsil edici özellikleri, klasik yöntemlere göre daha iyi çıkardığı genel olarak kabul edilmektedir.Bu katmanlar, girdiyi birden fazla soyutlama düzeyinde temsil edebilmekte ve büyük ölçekli verilerin küçük ayrıntılarını örtük olarak yakalayıp öğrenebilmektedir. Bu veriler görsel, sesli, metinsel, işaret (biyometrik ve sensör gibi) veya uzlamsal (konum verileri veya 3D veriler gibi) gibi herhangi bir türde olabilir. Derin öğrenme modellerini ihtiyaçlarımıza göre genişletmek mümkündür ve bu nedenle, farklı görüntü işleme görevleri için mevcut olan zengin bir yöntem ailesi ve çok sayıda denetimli ve denetimsiz öğrenme modeli [5] bulunmaktadır.

Büyük ölçekli verilerin hızlı işlenmesi ve yüksek doğruluk oranlarıyla tahmin yapılabilmesi, derin öğrenme yöntemlerinin önemli bir avantajıdır. Bu sayede, videolar içerisinde yer alan nesnelerin dahi tespit edilmesi mümkündür. Nesne tespiti, verilen bir görüntüde her bir nesnenin konumunu ve boyutunu belirlemek amacıyla sınır kutularını (nesneyi tanımlamak için kullanılan dikdörtgen alanlar) oluşturmayı hedefleyen bir görüntü işleme problemidir. Yüksek çözünürlüklü ve yüksek FPS sağlayan 360° kameraların yaygınlaşması ve ucuzlaması, Dronların ve diğer uçan cisimlerin hızlı ve kaliteli veri ile kayıt altına alınmasını kolaylaştırmış ve güvenilir nesne tespit algoritmalarına olan talebi artırmıştır. Bununla birlikte, mevcut nesne tespit algoritmalarının performansı, doğal ortamda çekilen

görüntülerdeki algoritmalara kıyasla, dron görüntülerinde yeterince yüksek değildir.

Bu tezde, dronların ve kuşların tespiti ile takibi, aynı zamanda bu iki farklı objenin ayrımını yapmak için iyi bilinen YOLOv8 nesne tespiti algoritmaları kullanılmıştır. Sürüm olarak YOLOv8'in tercih edilmesinin temel nedenleri, önceki sürümlere kıyasla sağladığı önemli iyileştirmeler ve yeniliklerdir. YOLOv8, daha yüksek doğruluk oranı (mAP: %55.2) ve daha düşük inferans süresi (25 ms) sunarak hem hız hem de doğruluk açısından üstünlük sağlamaktadır. Daha küçük model boyutu (12 MB), kaynak verimliliği ve daha geniş bir kullanım yelpazesi sunarken, CSPDarknet omurgası ve PANet++ gibi gelişmiş mimariler, özellik çıkarımını ve ölçekler arası nesne tespitini iyileştirmiştir. Anchor-free yaklaşımı, hiperparametre ayarlarını basitleştirerek daha iyi ölçeklenebilirlik sağlarken, mozaik ve Mixup veri çoğaltma teknikleri ile farklı boyut ve pozisyonlardaki nesnelerin algılanmasında modelin performansı artırılmıştır. Ayrıca, karışık hassasiyetli eğitim (16-bit kayan nokta) ile eğitim süresi hızlandırılmış ve bellek tüketimi azaltılmıştır. Bu yenilikler, YOLOv8'in hem performans hem de verimlilik açısından önceki sürümleri geride bırakmasını sağlamış ve tercih edilmesinde önemli bir rol oynamıştır[6]. Algoritmaların performansını artırmak amacıyla, farklı klasik nesne takip yöntemleriyle desteklenmiştir. Çalışmada, nesne tespitine ilişkin farklı yönler incelenmiş ve bu yöntemlerin algoritmaya olan etkileri sunulmuştur.

Bu tez, özellikle YOLO tabanlı tek aşamalı nesne tespit algoritmalarının klasik yöntemlerle harmanlanmasının, tespit performansını nasıl artırdığını ortaya koymaya odaklanmaktadır.

1.2 Tezin Amacı

Günümüz çatışmalarında insansız hava araçlarının (İHA) kullanımındaki artış, özellikle asimetrik tehditler bağlamında, güvenlik güçleri için ciddi bir sorun teşkil etmektedir[7]. Bu tez, video görüntüleri üzerinden YOLOv8 algoritmasını klasik nesne takip yöntemleriyle destekleyerek İHA'ların kuş gibi diğer hareketli nesnelerden ayrımını gerçekleştirmeyi ve böylece yanlış alarmların önüne geçerek güvenlik sistemlerinin etkinliğini artırmayı amaçlamaktadır. Bu çalışma, İHA tabanlı tehditlerin önlenmesine yönelik yüksek doğrulukta bir tespit ve sınıflandırma sistemi geliştirmeyi hedeflemektedir.

1.3 Hipotez

Yetkisiz dronların kritik alanlarda güvenlik riski oluşturduğu göz önüne alındığında, derin öğrenme tabanlı YOLOv8 algoritmasının, yüksek doğruluk oranı ve gerçek zamanlı performansıyla dronları diğer nesnelerden (örneğin kuşlar) ayırt ederek bu tehditlere karşı etkili bir tespit ve sınıflandırma sistemi sunabileceği öngörülmektedir.

2
DERİN ÖĞRENME İLE GÖRÜNTÜ İŞLEME YÖNTEMLERİ

Bu bölüm, derin öğrenme ile görüntü işleme prosedürlerinin bir incelemesini ve nesne tespitiyle ile ilgili temel terimleri tanıtır. Nesne tespiti görevinin amacı, verilen bir görüntüde mevcut nesnelerin sınıflarını belirlemek ve bu nesnelerin sınır kutularının koordinatlarını tespit etmektir. Literatürde nesne tespiti üzerine çok sayıda araştırma yapılmıştır. Son dönem çalışmaları bu alanda umut verici sonuçlar sunsa da, nesne tespiti görüntü işleme alanında uzun süredir ele alınan bir görevdir. Bu nedenle, yöntemleri iki ana kategori altında sınıflandırmak uygun olacaktır: (i) derin öğrenme tabanlı olmayan (klasik yöntemler) ve (ii) derin öğrenme tabanlı yöntemler.

Derin öğrenme tabanlı yöntemlerden bahsederken, mevcut veri kümelerinin tartışılması ekstra önemli hale gelir; çünkü veri, derin öğrenme tabanlı modellerin performansında kritik bir rol oynar. Sonuç olarak, bu bölüm üç alt başlık içermektedir: (i) Klasik yöntemler (ii) Derin öğrenme tabanlı nesne tespiti yöntemleri (iii) Bu çalışmada kullanılan ana nesne tespiti veri kümeleri

Nesne tespiti, günümüzde görüntü işleme alanında sıklıkla araştırılan bir konu haline gelmiştir. Görüntülerde yüz, hareketli ve sabit platformlar, hayvanlar gibi çeşitli sınıflara ait görsel nesneleri belirlemek için farklı algoritmalar ve yöntemler geliştirmektedir. Bu alandaki asıl hedef, "Görüntüde yer alan nesnenin türünü belirlemek ve görüntü üzerinde nerede olduğunu tespit etmek" sorularını yanıtlamaktır.

2.1 Klasik Yöntemler İle Nesne Tespiti

Öncelikle klasik yöntemlerden bahsedecek olursak, nesne tespiti veri girişi, ön işleme , nitelik çıkarma , kategori atama (sınıf atama) süreçlerinden oluşur[8] .



Şekil 2.1 Klasik nesne tespiti aşamaları[9].

2.1.1 Veri Girişi

Nesne tespiti süreçlerinin ilk ve en temel adımı olan veri girişi, ham verilerin sisteme alınmasına denir. Bu aşamada, sistem genellikle görüntü veya video gibi ham verilerle çalışır. Verilerin kalitesi, sonraki işlemlerin doğruluğunu doğrudan etkiler. Örneğin, düşük çözünürlüklü veya gürültülü görüntüler, nesne tespit performansında düşüşe neden olabilir. Bu nedenle, verilerin özelliklerine uygun bir şekilde hazırlanması büyük önem taşır. Etkin bir veri girişi aşaması, algoritmaların doğru sonuçlar üretebilmesi için sağlam bir temel oluşturur.

2.1.2 Ön işleme

Ön işleme aşaması, ham verilerin analize hazır hale getirilmesini sağlayan kritik bir adımdır. Bu süreçte, görüntülerdeki gürültü azaltılır, boyutlar normalize edilir ve özellik çıkarımına uygun hale getirilir. Örneğin, düşük kaliteli bir görüntüde kontrast artırılarak detayların belirginleşmesi sağlanabilir. Şekil2.2'de ön işleme yöntemlerinden birisi olan CLAHE (Contrast Limited Adaptive Histogram Equalization) uygulamasının sonucu sunulmuştur. Ayrıca, veri formatlarının algoritmaya uygun şekilde dönüştürülmesi de bu aşamada gerçekleştirilir. Ön işleme, ham verilerde bulunan hataları gidererek veya verilerin eksikliklerini tamamlayarak, tespit ve sınıflandırma aşamalarında daha doğru sonuçlar elde edilmesine katkıda bulunur. Bu sayede, algoritmaların performansı optimize edilir ve işlem süresi kısaltılabilir[10].



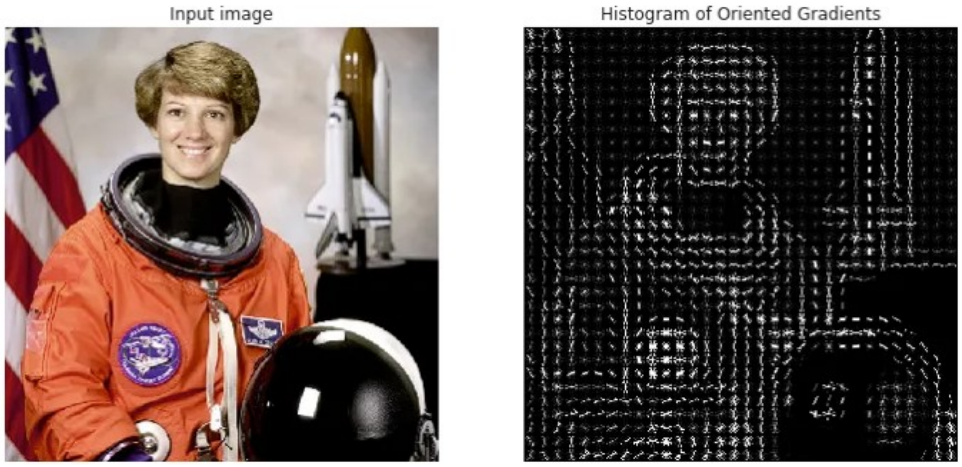
Şekil 2.2 Contrast Limited Adaptive Histogram Equalization (CLAHE) uygulaması[11].

2.1.3 Öz Nitelik Çıkarma

Öznitelik çıkarma, görüntü işleme ve nesne tanıma süreçlerinde temel bir adımdır. Bu adımda, ham veriden ayırt edici ve anlamlı özellikler elde edilerek daha üst düzey işlemler için bir temel sağlanır. Öznitelikler, görüntüdeki kenar, köşe, doku veya renk gibi belirli unsurların matematiksel temsilidir. Doğru özelliklerin seçilmesi, nesne tanıma ve sınıflandırma performansını doğrudan etkiler [12] .

Öznitelik çıkarma sürecinde genellikle SIFT (Scale-Invariant Feature Transform), HOG (Histogram of Oriented Gradients) ve SURF (Speeded Up Robust Features) gibi algoritmalar kullanılır. SIFT, görüntüde ölçek ve döndürme değişimlerine dayanıklı öznitelikler çıkarırken [12], HOG, yönlendirilmiş gradyanların histogramlarını kullanarak nesne sınırlarını belirginleştirir [13]. SURF ise SIFT'e kıyasla daha hızlı çalışarak gerçek zamanlı uygulamalarda kullanılabilir [14].

Bu süreç, makine öğrenimi modellerinin daha anlamlı ve yüksek doğruluk oranlarına ulaşmasını sağlar. Örneğin, SIFT veya HOG gibi algoritmalarla çıkarılan öznitelikler, bir sınıflandırıcıya (örneğin SVM) girdiler olarak verilerek nesnelerin türü belirlenebilir. Örnek olarak HOG algoritması ile çıkartılan öz nitelik görseli Şekil2.3'deki gibidir. Özniteliklerin doğru bir şekilde çıkarılması, modelin genelleştirme kabiliyetini artırır ve sınıflandırma hatalarını azaltır [13–15].



Şekil 2.3 Histogram of Oriented Gradients (HOG) algoritması ile çıkarılan öznitelikler[16].

2.1.4 Tanımlama

Tanımlama, görüntü işleme ve nesne tanıma süreçlerinde önemli bir adımdır ve genellikle (bazı durumlarda doğrudan öznitelik çıkarımı yapılmaksızın da tanımlama gerçekleştirilebilir) öznitelik çıkarma aşamasının ardından gelir. Bu aşama, çıkarılan özniteliklerin belirli bir sınıfla ilişkilendirilmesi ve tanımlanması sürecidir. Tanımlama aşamasında, çıkarılan özellikler kullanılarak görüntüdeki nesnelerin hangi kategoriye ait olduğu belirlenir. Sonuçlar, nesnenin doğru sınıflandırılmasında belirleyici rol oynar.

Bu aşamada kullanılan en yaygın yöntemler arasında, örüntü tanıma, sınıflandırma algoritmaları ve özniteliklerin sınıflar arası ayrım gücüne dayalı teknikler bulunur. Klasik nesne tanımlama yöntemlerinde, genellikle sınıflandırıcılar kullanılarak bir nesneye ait özellikler değerlendirilir. Örneğin, HOG (Histogram of Oriented Gradients) ile elde edilen öznitelikler[15], sınıflandırıcı algoritmaları aracılığıyla tanımlanır ve hangi nesnenin ne türde olduğu belirlenir.

Tanımlama süreci, aynı zamanda sınıflandırma hatalarının azaltılmasına ve doğru sonuçların elde edilmesine yardımcı olur. Bu aşamada doğru karar mekanizmaları ve doğru özniteliklerin seçilmesi, nesnenin doğru tanımlanması için kritik öneme sahiptir. Klasik yöntemlerde, genellikle karar ağaçları, doğrusal sınıflandırıcılar ve benzeri teknikler kullanılarak bu işlem gerçekleştirilir. Bu yöntemler, sınıflandırma ve tanımlama performansını arttırarak nesne tespitinde daha doğru sonuçlar elde edilmesini sağlar.

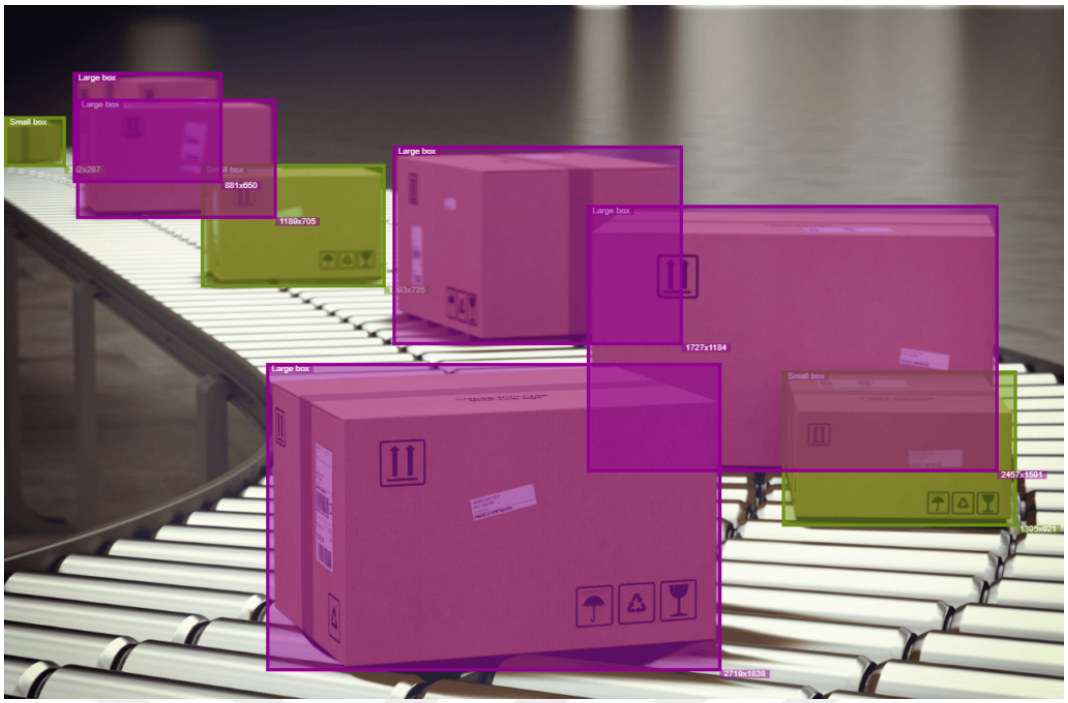
2.2 Derin Öğrenme ile Görüntü işleme

Donanım teknolojilerindeki ilerlemeler ve hız ile boyut kısıtlamalarının ortadan kalkmasıyla birlikte, derin öğrenme alanında önemli gelişmeler kaydedilmiştir. Bu gelişmeler, görüntü işleme alanında bazı klasik yöntemlerin yerini almıştır. Özellikle öznitelik çıkarma ve el ile yapılan hesaplamalar gibi iş yükleri, Konvolüsyonel Sinir Ağları (CNN'ler) kullanılarak daha hızlı ve doğru bir şekilde gerçekleştirilebilir hale gelmiştir.

Günümüzde derin öğrenme tabanlı yöntemler, güvenlik kameralarında nesne tespiti ve sınıflandırması, otonom sürüş sistemleri ve sağlık sektöründe kullanılan uygulamalar gibi birçok farklı alanda başarıyla uygulanmaktadır. Örneğin, derin öğrenme yöntemleri, kanserli hücrelerin tespiti ve sınıflandırılmasında, ayrıca tıbbi görüntülerde anomali tespiti gibi kritik sağlık uygulamalarında kullanılmaktadır[17–19].

Derin öğrenme algoritmalarının temel çalışma prensibi şu şekildedir: Önceden kimliklendirilmiş bir veri setindeki her bir nesne, algoritmaya tanıtılır ve algoritma, her bir nesne için benzersiz bir kimlik numarası (ID) geliştirir. Bu kimlikler, hafızada tutulur ve daha sonra benzer nesnelerle karşılaşıldığında, bu nesnelerin tanınmasını sağlar. Ayrıca, video akışındaki her bir kare ile birlikte (burada video, sırasıyla akan resimler olarak düşünülebilir), daha önce tespit edilen nesneler, sonraki karelerdeki nesnelerle ilişkilendirilerek izlenebilir. Bu süreç, video içeriğindeki nesnelerin sürekli olarak takip edilmesini ve tanımlanmasını sağlar.

Nesne tespiti, nesnelerin sınıflandırılması ve bunların kutucuklar (sınır kutuları) içine alınması sayesinde gerçekleştirilir. Bu nedenle, nesne tespiti uygulamasının temel iki görevi vardır: Nesneleri sınıflandırmak ve bu nesneleri, başka objelerin içine girmeyecek şekilde (eğer mümkünse, nesneler iç içe geçmişse bu durum dikkate alınarak) ve sınıflandırdığımız nesnenin tamamen içinde bulunduğu bir kutu ile sınırlamak. Şekilde5.1, farklı kimliklerle (ID) sınıflandırılmış birden fazla obje ve bu objelerin sınır kutularını görebilirsiniz.



Şekil 2.4 Farklı nesnelerin sınıflandırılması ve her bir nesneye ait sınır kutularının (bounding boxes) gösterilmesi[20].

Görüntülerdeki belirli nesneleri tespit etmek için kullanılan yöntemler, aynı zamanda çok sayıda nesne sınıfını da tanıyabilecek şekilde genişletilebilir. Bu tür bir işlem, nesnelerin hem türünü hem de görüntüdeki konumlarını belirlemeyi amaçlar. Her bir nesne için bir etiket ve sınırlayıcı kutu eklenir, ayrıca nesnenin doğru sınıflandırılma olasılığına dair güven puanı da atanır. Bu süreç, doğru sonuçlar elde edilmesini sağlar.

Convolutional Neural Networks (CNN'ler), görüntü işleme alanındaki en önemli gelişmelerden biri olarak kabul edilmektedir. 1980'lerin ortasında, Yann LeCun ve ekibi tarafından geliştirilmiş ve özellikle el yazısı rakamları tanıyan LeNet-5 modeli ile büyük başarı elde etmiştir. Ancak CNN'lerin gerçek potansiyelini gösteren ve modern bilgisayarla görme uygulamalarının temelini atan gelişme, 2012'de AlexNet'in ImageNet yarışmasında birinci olmasıyla gerçekleşmiştir. AlexNet, derin öğrenme yöntemlerinin gücünü ve CNN'lerin bilgisayarla görme alanındaki önemini gözler önüne serdi.

RCNN (Regions with CNN), CNN'lerin klasik yöntemlerle birleştiği ilk önemli adımlardan biridir. 2014 yılında Ross B. Girshick tarafından geliştirilen RCNN, nesne tespiti alanında büyük bir yenilik getirmiştir. Bu model, ilk olarak görüntüdeki potansiyel nesne bölgelerinin (region proposals) çıkarılmasını sağlar. Çıkarılan bu bölgeler, daha sonra bir CNN modeline verilerek sınıflandırılır. RCNN,

nesne tespiti için CNN'leri kullanmaya başlamış ancak işlem süresi ve hesaplama maliyetleri nedeniyle bazı sınırlamaları da beraberinde getirmiştir[21].

Fast-RCNN (2015) ise RCNN'in temel sorunlarını çözmek için geliştirilmiştir. Girshick, Fast-RCNN'de, bölge önerisi oluşturma adımının ardından her bir bölgenin CNN üzerinden geçirilmesini daha verimli bir şekilde yaparak eğitim sürecini hızlandırmıştır. Ancak, hala bölge önerisi adımının dışarıdan yapılması gerektiği için, süreçte ek bir işlem süresi bulunmaktadır[22].

Faster-RCNN (2015), Fast-RCNN'i bir adım daha ileri götürerek, "Region Proposal Network" (RPN) adı verilen yeni bir yapı eklemiştir. RPN, bölge önerisi üretme adımını tamamen CNN içerisinde yapabilen bir yöntemdir. Bu sayede Faster-RCNN, end-to-end bir model haline gelmiş ve bölge önerisi sürecinden kaynaklanan zaman kayıplarını ortadan kaldırmıştır. Bu gelişme, nesne tespitinde daha hızlı ve verimli çözümler sunmaya imkan tanımıştır[23].

SPPNet (2014) ise CNN'lerin işlem verimliliğini artıran bir başka önemli yöntemdir. Spatial Pyramid Pooling (SPP) katmanı, farklı boyutlardaki görüntülerden sabit boyutta özellikler çıkarmaya olanak tanır. Bu, geleneksel CNN'lerin görüntüleri yeniden ölçeklendirmeye duyduğu ihtiyacı ortadan kaldırarak, farklı boyutlardaki görüntülerle çalışabilmesini sağlamıştır. Ayrıca, her bir görüntü üzerinden yalnızca bir kez özellik hesaplanması, işlem süresini önemli ölçüde azaltır[24].

Feature Pyramid Networks (FPN), nesne tespitinin ölçekle ilgili sorunlarını çözmeye yönelik bir başka önemli gelişmedir. CNN modellerinin genellikle yalnızca üst katmanlarını kullanarak nesneleri tespit etmesi, farklı ölçeklerdeki nesnelerin tespitinde sınırlamalara yol açmaktadır. FPN, üstten aşağıya doğru olan bir yapı kullanarak, tüm ölçeklerdeki yüksek seviyedeki özellikleri yakalamayı mümkün kılar. Bu sayede, farklı boyutlardaki nesnelerin tespitinde önemli bir iyileşme sağlanmıştır[25].

SVM (Support Vector Machines), özellikle klasik nesne tanıma yöntemlerinde yaygın olarak kullanılan güçlü bir sınıflandırıcıdır. SVM, genellikle öznitelik çıkarma aşamasında elde edilen verilerle nesneleri sınıflandırmak için kullanılır. HOG (Histogram of Oriented Gradients) gibi özellik çıkarma teknikleriyle elde edilen veriler, bir SVM sınıflandırıcısına verilir ve bu sınıflandırıcı, nesnenin hangi sınıfa ait olduğunu belirler. SVM, derin öğrenme modellerine geçiş yapılana kadar, nesne tespiti ve sınıflandırmasında çok önemli bir araç olmuştur[26].

Bu yöntemlerin birleşimi, nesne tespitinde kullanılan tekniklerin evrimini ve derin

öğrenmenin sağladığı ilerlemeleri gösterir. İlk başta CNN'lerin kullanımı oldukça sınırlıyken, RCNN, Fast-RCNN ve Faster-RCNN gibi modeller, bu teknikleri daha verimli ve hızlı hale getirmiştir. Sonraki adımlar ise, daha iyi ölçekleme ve daha verimli hesaplamalar yapabilmek adına geliştirilen yöntemlerle bu ilerlemeyi pekiştirmiştir.

Derin öğrenme tabanlı tek aşamalı tespit modelleri, nesneleri yoğun örnekleme yöntemiyle tek bir işlemde hem sınıflandırır hem de konumlandırır. İki aşamalı ve tek aşamalı nesne tespit modelleri arasında belirgin farklılıklar bulunmaktadır. Tek aşamalı modeller, farklı ölçek ve oranlara sahip önceden tanımlanmış sınırlayıcı kutuları kullanarak nesnelerin konumlarını belirler. Bu yaklaşım, iki aşamalı modellerdeki bölge önerisi adımına olan gereksinimi ortadan kaldırır ve tespit sürecini önemli ölçüde hızlandırır. Sonuç olarak, tek aşamalı nesne tespit modelleri daha yüksek tespit hızları sunar.

2.3 Çalışmada Kullanılan Ana Nesne Tespit Yöntemi

Bu çalışmada, dronların hızları ve video görüntülerinin FPS değerleri göz önünde bulundurularak, incelenen veri türü, hassasiyet gereksinimleri ve daha küçük cihazlarda çalıştırılabilirlik açısından optimize edilebilirlik özellikleri nedeniyle, derin öğrenme tabanlı tek aşamalı nesne tespit modeli olan YOLO[5, 6, 17, 27–32] tercih edilmiştir. Tek aşamalı modeller (YOLO Family, SSD, DSSD, RetinaNet, M2Det, RefineDet, CenterNet, CornerNet vb.), farklı boyut ve en-boy oranlarına sahip önceden tanımlanmış sınır kutularını kullanarak nesnelerin konumlarını belirler ve bu sayede, iki aşamalı modellerdeki bölge önerisi sürecine olan ihtiyacı ortadan kaldırır. Böylelikle, bu modellerle tespit hızında önemli bir artış sağlanmıştır.

3
YOLO AİLESİ VE NESNE TESPİTİNDE KULLANILAN VERİ SETLERİ

3.1 Yolo Ailesi

YOLO ailesi, en popüler nesne tespiti modellerinden biridir. Bunun başlıca nedenleri arasında açık kaynaklı olması ve modelin performansının yıllar içinde sürekli olarak geliştirilmiş olması yer alamaktadır. YOLO ailesi, birleştirilmiş bir modelde tek aşamalı ağlara aittir. Yıllar boyunca, her biri en son teknolojik yeniliklerle geliştirilen birçok YOLO sürümü piyasaya sürülmüştür. Bu bölümde, YOLO ailesinin gelişimi ve neden YOLOv8 modeli kullanıldığı ayrıntılı bir şekilde açıklanmış ve önerilen diğer modeller YOLOv3 [27], YOLOv5 [33] ve YOLOV8[6] modelleri ile ilgili bilgi verilmiştir.

3.1.1 Orijinal YOLO (You Only Look Once) Algoritması

YOLO, 2016 yılında Joseph Redmon ve ekibi tarafından nesne tespiti için tavsiye edilen bir derin öğrenme algoritmasıdır[34]. Bu algoritma, nesne tespiti problemini bir sınıflandırma problemi gibi ele alarak yenilikçi bir yaklaşım sunmuştur. Geleneksel iki aşamalı (region proposal ve sınıflandırma) yöntemlerden farklı olarak, YOLO, tek bir ileri besleme işlemiyle bir görüntüdeki tüm nesneleri hem sınıflandırır hem de konumlandırır.

Algoritmasının temelinde, giriş görüntüsünün eşit boyutlu grid hücrelerine bölünmesi yatmaktadır. Her bir grid hücresi, bir nesnenin sınırlayıcı kutusunun merkezi olup olmadığını tahmin etmekle görevlidir. Bu yaklaşım, her grid hücresi için belirli bir sınır kutusu ve sınıf etiketi önerilmesini sağlar. Ayrıca, grid hücreleri, bir nesne içerip içermediğini göstermek için bir "Güven Skoru" üretir. Güven Skoru, bir nesnenin sınır kutusu içinde bulunma olasılığını ve bu kutunun gerçek nesneye olan yakınlığını ifade eden "Intersection over Union" (IoU) değeri ile hesaplanır. YOLO'nun bu verimli ve hızlı yapısı, özellikle gerçek zamanlı uygulamalarda

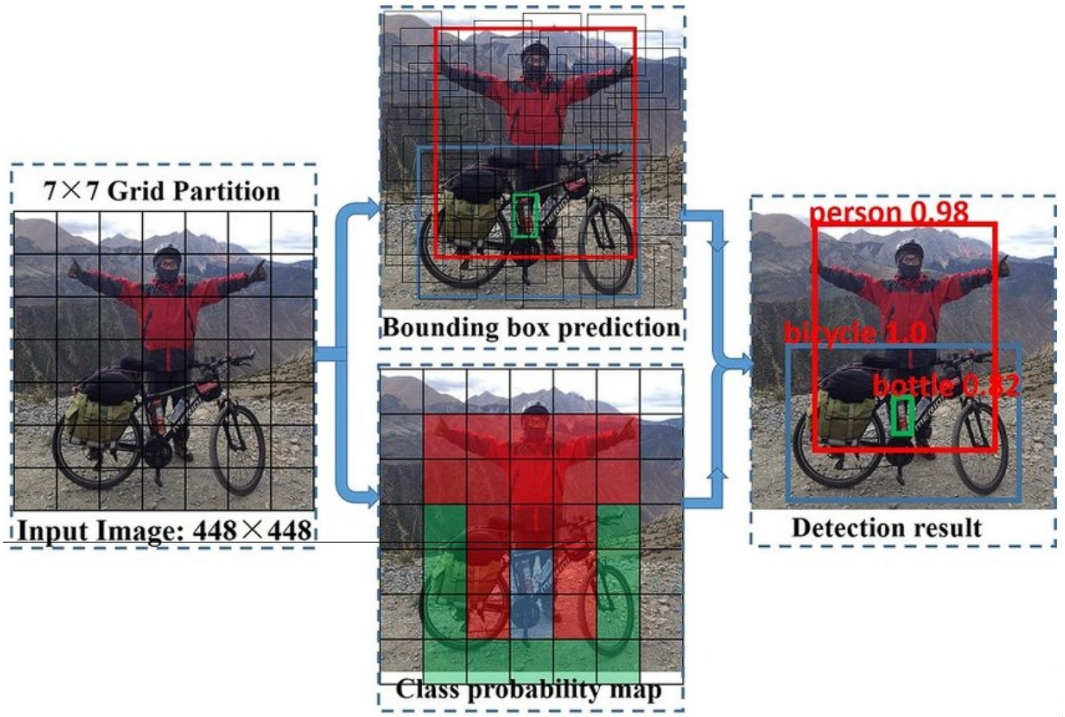
kullanılmasını mümkün kılar. Güvenlik kameraları, otonom araçlar ve insansız hava araçları gibi alanlarda modelin yüksek hız ve verimlilik sağlaması, algoritmanın öne çıkan özelliklerinden biridir. YOLO algoritmasının temel denklemi, "Güven Skoru" hesaplamasını şu şekilde ifade etmektedir:

$$\text{Confidence Score} = P_r(\text{Object}) \times \text{IoU}^{\text{gt}}_{\text{pred}} \tag{3.1}$$

Teorem 3.1. Güven skoru, nesnenin sınır kutusunda yer alma olasılığı ve tahmin edilen kutu ile gerçek kutu arasındaki Intersection over Union (IoU) değeri kullanılarak hesaplanır.

Önerilen sınırlayıcı kutunun merkez noktası, 0 ile 1 arasında bir değer almalıdır. Bu durum, merkezin kesinlikle grid hücresinin içerisinde yer almasını sağlar. Değerlerin 0 ile 1 aralığında tutulması kritiktir çünkü her grid hücresi yalnızca tek bir merkez noktası belirlemekten sorumludur.

Başlangıç noktası olarak hücrenin sol üst köşesi baz alındığında, değer 1'i aşarsa, bu durum bir sonraki ya da alttaki hücrenin merkez noktasını önermesine yol açar ve modelin bütünleşik yapısını bozar. Bunun yanı sıra, yalnızca merkez koordinatları (x, y) değil, aynı zamanda sınırlayıcı kutunun genişlik ve yüksekliği de hesaplanır ve bu değerler görüntünün boyutlarıyla orantılı olarak belirlenir.



Şekil 3.1 Grid hücreleri fikrini temsil eden örnek bir görsel [16]

.

3.1.2 YOLOv3

YOLOv3, J. Redmon ve A. Farhadi tarafından 2018 yılında geliştirilmiş ve orijinal YOLO algoritmasının önemli iyileştirmeler sunan bir versiyonudur [27]. YOLOv3, hem doğruluk hem de esneklik açısından geliştirilmiş, daha karmaşık nesne tespit problemlerine yanıt verebilecek şekilde tasarlanmıştır. Orijinal 2016 YOLO algoritmasına kıyasla birçok teknik yenilik ve algoritmik iyileştirme içermektedir.

İlk olarak, algoritmada nesne tespiti için Darknet-53 omurgasını kullanır. Darknet-53, 53 katmanlı bir konvolüsyonel sinir ağı olup, hem hız hem de doğruluk açısından VGG-16 ve ResNet-101 gibi eski omurga modellerine kıyasla önemli avantajlar suna[35] . Bu yapı, daha derin özelliklerin daha iyi çıkarılmasını sağlar ve karmaşık nesne yapılarını tespit etme kabiliyetini artırır [27].

YOLOv3'ün getirdiği bir diğer önemli yenilik, çok ölçekli özellik haritalarının kullanılmasıdır. YOLOv3, nesne tespiti sırasında üç farklı ölçek seviyesi üzerinde çalışır ve bu, farklı boyutlardaki nesnelerin tespit edilmesinde doğruluk oranını artırır. Özellikle küçük nesnelerin tespitinde yaşanan zorlukları gidermek için, alt katmanlardan gelen düşük seviyeli özellikler, üst katmanlardan gelen yüksek seviyeli özelliklerle birleştirilmiştir.

Algoritmada yeni bir sınıflandırma yöntemide benimsenmiştir. Sınıf etiketi tahmini için sigmoid aktivasyon fonksiyonunu kullanır ve her sınıf için ayrı bir "binary cross-entropy" kayıp fonksiyonu uygular. Bu yaklaşım, çok sınıflı nesne tespit problemlerinde doğruluğu artırmıştır. Algoritmada kullanılan bu kayıp fonksiyonu şu şekilde ifade edilmektedir:

$$\text{Loss} = \text{Binary Cross-Entropy}(y, \hat{y}) \tag{3.2}$$

Teorem 3.2. Loss değeri, tahmin edilen çıktılar ile gerçek çıktılar arasındaki farkı ölçmek için Binary Cross-Entropy (Çapraz Entropi) yöntemi kullanılarak hesaplanır. Bu yöntem, modelin tahminlerinin doğruluğunu değerlendirmek ve hataları minimize etmek amacıyla tanımlanır.

YOLOv3, nesne tespiti süreçlerini daha etkili hale getirmek için "Anchor Boxes" konseptini entegre etmiştir. Bu kutular, her bir grid hücresinde nesnelerin tanımlanmasını kolaylaştırarak farklı boyutlardaki nesnelerin daha iyi algılanmasını sağlar. Ayrıca, bu yöntem algoritmanın gerçek zamanlı uygulamalarda daha yüksek verimlilikle çalışmasına katkıda bulunur.

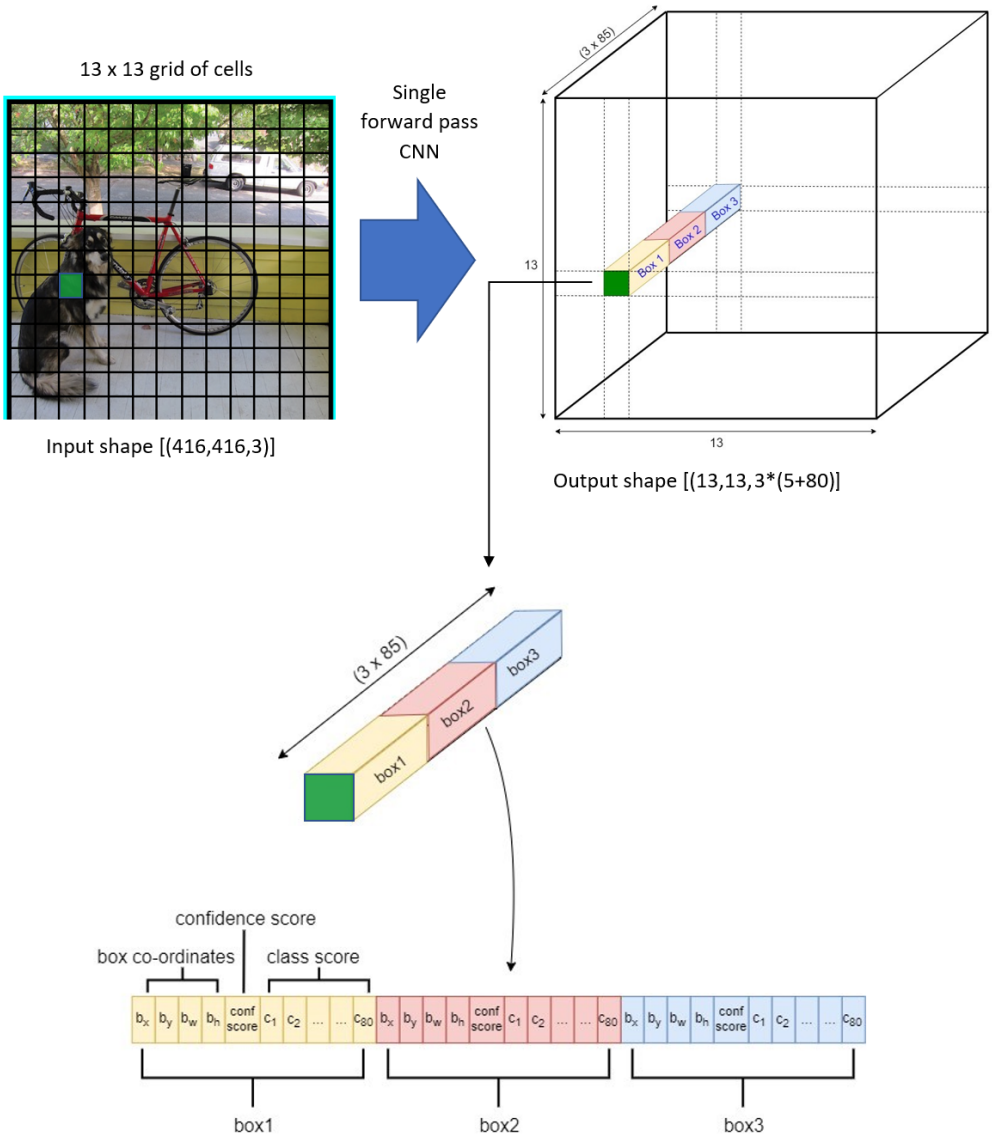
Tahmin edilen sınırlayıcı kutuların doğruluğunu artırmak amacıyla, "Güven Skoru"

kavramı da bu modelde önemli bir yer tutmaktadır. Güven skoru, bir sınır kutusunun nesne içerip içermediğini ve tahmin edilen kutunun gerçek nesneye olan yakınlığını değerlendirmek için kullanılmaktadır. Bu skor, aşağıdaki denklem ile ifade edilmektedir:

$$\text{Confidence} = P(\text{Object}) \times \text{IoU}_{\text{pred, truth}} \tag{3.3}$$

Teorem 3.3. Güven skoru(Confidence), tahmin edilen sınırlayıcı kutunun bir nesne içerme olasılığı ile tahmin edilen kutu ve gerçek kutu arasındaki Intersection over Union (IoU) değeri kullanılarak hesaplanır. Bu yöntem, tahmin edilen kutunun doğruluğunu değerlendirmek amacıyla tanımlanır.

Bu gelişmeler, algoritmanın gerçek zamanlı performansını önemli ölçüde artırmıştır. Hem hız hem de doğruluk açısından önceki sürümleri geride bırakan model, özellikle güvenlik sistemleri, otonom araç teknolojileri ve insansız hava araçları gibi birçok alanda yaygın bir şekilde kullanılmaktadır [27].



Şekil 3.2 YOLOv3 Modelinde Grid Hücreleri ve Anchor Kutularının Yapısı[36] .

3.1.3 YOLOv5

YOLOv5, nesne tespiti alanında YOLO ailesinin önemli bir üyesi olarak 2021 yılında tanıtılmıştır [33]. Bu model, YOLOv3'e kıyasla çeşitli iyileştirmeler sunarak hız, doğruluk ve optimizasyon açısından önemli ilerlemeler kaydetmiştir. YOLOv3'ün temel mimarisi olan Darknet-53[35]'ün yerine, YOLOv5'te daha hafif ve optimize edilmiş bir model kullanılmaktadır. Bu sayede, YOLOv5, daha küçük cihazlarda çalışabilme yeteneğini artırarak özellikle gömülü sistemler ve gerçek zamanlı uygulamalar için daha uygun hale gelmiştir [27].

YOLOv3 ile YOLOv5 arasındaki en belirgin farklardan biri, eğitim sürecindeki

verimlilik ve esnekliktir. YOLOv5, PyTorch tabanlı bir model olarak geliştirilmiş olup, eğitim ve uygulama aşamalarında esnek bir yapıya sahiptir. Bu durum, kullanıcıların modeli kolayca özelleştirmesine ve farklı veri kümeleri üzerinde hızla eğitmesine olanak tanır [32]. Ayrıca, YOLOv5, veri ön işleme, veri artırma (data augmentation) ve hiperparametre optimizasyonu konularında otomatikleştirilmiş teknikler kullanarak daha tutarlı sonuçlar elde etmektedir [29].

Bunun yanı sıra, "Anchor-Free" ve "Anchor-Based" yöntemlerini birleştirerek tespit doğruluğunu artırmaktadır. YOLOv3, anchor kutuları konseptini kullanarak nesne tespiti yaparken, YOLOv5 bu yaklaşımı daha esnek hale getirmiş ve küçük nesnelerin tespitinde performansı önemli ölçüde iyileştirmiştir. Ek olarak, YOLOv5, "Mosaic Augmentation" gibi veri artırma tekniklerini kullanarak eğitimi güçlendirmiş ve çeşitli ölçeklerde nesne tespitindeki başarısını artırmıştır [31].

Sonuç olarak, YOLOv5, YOLOv3'e kıyasla daha hızlı, daha hafif ve daha doğruluklu bir model olarak öne çıkmaktadır. Özellikle gömülü sistemlerde ve gerçek zamanlı uygulamalarda yaygın olarak kullanılmaktadır. Bu iyileştirmeler sayesinde, YOLOv5, nesne tespiti problemlerinde pratikte daha yaygın bir çözüm haline gelmiştir [6, 27, 31, 33].

3.1.4 YOLOv8

YOLOv8, YOLO ailesinin en güncel sürümü olarak nesne tespiti, segmentasyon ve sınıflandırma görevlerinde büyük gelişmeler sunmaktadır. YOLOv5'in başarılarını temel alan bu sürüm, hız ve doğruluk dengesini bir üst seviyeye taşıyarak daha hızlı ve daha hassas bir tespit performansı sağlamaktadır. Tek aşamalı tespit mimarisi üzerinde çalışan YOLOv8, anchor-free yapı, geliştirilmiş backbone ve optimize edilmiş kayıp (loss) fonksiyonları ile dikkat çeker. Bu yenilikler, YOLOv5'e kıyasla hem performans hem de hesaplama verimliliği açısından ciddi iyileştirmeler sunmaktadır [6, 27].

YOLOv5'te anchor tabanlı yöntem kullanılırken, YOLOv8 tamamen anchor-free bir yapıya geçmiştir. Bu yaklaşım, önceden tanımlanmış kutu boyutlarına ihtiyaç duymadan nesne tespiti yaparak daha esnek ve hızlı bir çözüm sunmaktadır [32]. Ayrıca YOLOv5'te bulunan Focal Loss ve IoU (Intersection over Union) tabanlı kayıp fonksiyonlarının yerine, YOLOv8'de Complete IoU (CIoU) kullanılmıştır. CIoU, yalnızca sınır kutuları arasındaki örtüşmeyi değil, aynı zamanda merkezler arası mesafeyi ve kutu yönelimlerini de dikkate alarak daha doğru bir tespit sağlar:

$$\text{CIoU} = IoU - \frac{\rho^2(b, b^{gt})}{c^2} - \alpha v \tag{3.4}$$

Burada b, tahmin edilen sınır kutusunu; b^{gt}, gerçek sınır kutusunu; ρ, merkezler arası Öklid mesafesini; c, en küçük çevreleyen kutunun diyagonal uzunluğunu; ve v, kutuların yönelim farkını ifade etmektedir [6]. Bu sayede YOLOv8, özellikle küçük nesnelerin tespitinde önceki sürümlere göre daha yüksek doğruluk sunmaktadır.

Model mimarisi açısından YOLOv8, CSPNet gibi optimize edilmiş backbone'lar kullanarak daha derin ve anlamlı özellik haritaları elde etmektedir. YOLOv5'in başarılı performansını daha da geliştiren bu yapı, daha karmaşık nesne yapılarını tanımlama kabiliyetini artırırken, model boyutunu küçülterek düşük donanım kaynaklarında çalıştırılabilirliğini sağlamıştır [27]. Özellikle gerçek zamanlı uygulamalarda YOLOv8, YOLOv5'e göre %15-20 oranında daha hızlı tespit yaparken doğruluk değerlerinde de artış göstermektedir[6].

Ayrıca YOLOv8, gelişmiş veri genişletme teknikleri (data augmentation) ve mükemmel optimize edilmiş eğitim stratejileri sayesinde, eğitim süresini kısaltarak daha hızlı ve verimli bir öğrenme süreci sağlar. Özellikle otonom sürüş sistemleri, güvenlik kameraları ve tıbbi görüntü analizleri gibi zaman kritik uygulamalarda bu performans farkı, YOLOv8'i bir adım öne çıkarmaktadır [6, 32].

Bu tez çalışmasında YOLOv8'in kullanılması, sağladığı hız, doğruluk ve donanım verimliliği nedeniyle uygun ve isabetli bir tercih olmuştur. Modelin sunduğu yenilikler, nesne tespiti gibi zaman duyarlı görevlerde önemli avantajlar sağlamaktadır.

3.2 Nesne Tespitinde Kullanılan Veri Setleri

Derin öğrenme modelleri, büyük miktarda veriye ihtiyaç duyan veri odaklı yapılardır. Bu modellerin başarısı, yalnızca veri miktarıyla değil, aynı zamanda verinin kalitesiyle de doğrudan ilişkilidir. Özellikle YOLO gibi derin öğrenme tabanlı nesne tespiti modellerinin eğitimi için yüksek kaliteli veri setleri büyük bir öneme sahiptir. Yetersiz veya düşük kaliteli veri sağlanması durumunda, modelin performansında ciddi düşüşler gözlenebilir ve beklenen doğruluk seviyelerine ulaşmak mümkün olmayabilir.

Derin öğrenme modellerinin geliştirilme sürecinde, zamanın ve iş gücünün büyük bir kısmı veri setlerinin toplanması, temizlenmesi ve analiz edilmesine harcanmaktadır. Bu süreç, modelin nihai performansını doğrudan etkileyen kritik

bir adımdır. Her ne kadar transfer öğrenme ve gözetimsiz öğrenme gibi alternatif yöntemler veri gereksinimini bir ölçüde azaltsada, gerçek dünya uygulamalarında çalışan etkili bir derin öğrenme mimarisi oluşturmak için yüksek kaliteli ve kapsamlı veri setlerine sahip olmak vazgeçilmez bir gereklilik olmaya devam etmektedir.

Görüntü işleme alanında kullanılan veri setleri, yüksek sayıda veri örneği içeren birçok açık kaynak seçeneği sunmasına rağmen, genellikle toplandıkları yöntemler ve bulundukları ortamlar nedeniyle çeşitli sınırlamalara sahiptir. Bununla birlikte, mevcut veri setleri aşağıdaki sınırlamalarla karşı karşıya kalmaktadır:

* Yetersiz Veri: Derin öğrenme algoritmaları, özellikle Konvolüsyonel Sinir Ağları (CNN'ler) için gerekli olan geniş veri örneklerinin eksikliği, algoritmaların performansını sınırlayabilir.

* İnsan Hatası: Verilerin toplanmasında kullanılan yöntemler, önyargıya veya insan kaynaklı hatalara neden olabilir.

* Veri Kalitesi: Gerçek dünyadan elde edilen görüntü veya video tabanlı veri setleri genellikle düzensiz ve karmaşıktır, bu da model eğitimi sürecini zorlaştırabilir.

* Gizlilik ve Uyumluluk: PDPC[37] veya Türkiye'de bilinen adıyla KVKK[38] gibi gizlilik düzenlemeleri nedeniyle birçok kaynak, veri setlerini paylaşmamaktadır.

4.Etiketleme Süreci: Veri setlerinin çoğu, kaliteli etiketlemeler sağlamak için manuel olarak işlenmektedir. Ancak, bu süreç son derece zaman alıcı ve maliyetlidir. Bu sınırlamalar, derin öğrenme modellerinin geliştirilmesi ve uygulanması sürecinde karşılaşılan temel zorluklar arasında yer almaktadır.

Video görüntüleri üzerinden insansız hava araçlarının (İHA), yaygın adıyla dronların, tespiti ve takibini sağlamak için YOLO algoritması ile klasik yöntemleri bir araya getirerek performans iyileştirmesi sağlamaktır. Dronların irtifa ve yükseklik gibi bileşenlerine bağlı olarak değişen yakalama açıları ve ölçekleri, tespit ve takip süreçlerinde belirli zorluklar oluşturmaktadır. Tezin müteakip bölümlerinde, kullanılan veri setinin detaylı bir şekilde incelenmesi ile birlikte, YOLO algoritmasının klasik yöntemlerle entegre edilmesi ve bu entegrasyonun sağladığı katkılar ele alınacaktır.

4
KULLANILAN VERİ SETLERİ VE KLASİK YÖNTEMLERLE DESTEKLENMİŞ YOLOv8 İLE EĞİTİLMESİ

4.1 Veri Setleri ve Etiketleme Süreçleri

Bu çalışmada, drone ve kuşların tespiti için iki farklı veri seti kullanılmıştır. Kullanılan veri setlerinin detayları ve işleme süreçleri aşağıda açıklanmaktadır. Her iki veri seti de işaretleme işlemleri için CVAT platformu kullanılarak elle etiketlenmiştir.

4.1.1 Kaggle Veri Seti

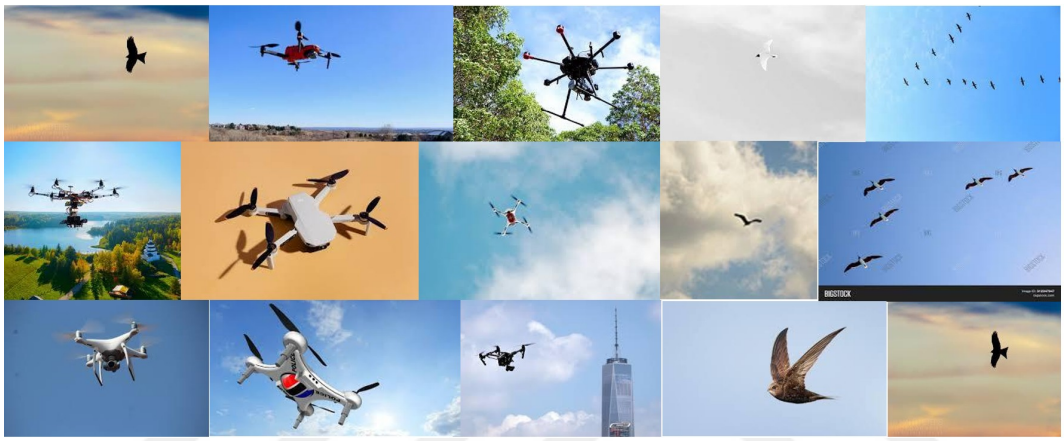
İlk veri seti, Kaggle platformundan temin edilen Birds vs Drone Dataset başlıklı veri setidir. Bu veri seti, drone ve kuş görüntülerini içermesi nedeniyle, bu iki nesne türünün ayrımına yönelik nesne tespiti çalışmalarında oldukça önemlidir. Toplamda 1300 adet görüntüden oluşan bu veri setinde görseller, kuş ve drone olmak üzere iki ana sınıfa ayrılmıştır. Görsellerin farklı perspektiflerden ve çevresel koşullardan alınması, modelin çeşitli senaryolarda yüksek doğruluk göstermesine katkı sağlamıştır. Veri setindeki tüm görseller CVAT platformu aracılığıyla işaretlenmiş, her bir nesnenin sınır kutuları dikkatlice belirlenmiştir.

4.1.2 Toplanan Video Verileri

İkinci veri seti, internetten toplanan 77 adet video (44 saat) görüntüsünden alınan örnekler kullanılarak oluşturulmuştur. Bu videolar, kuş ve drone görüntülerini gerçek dünya koşullarında yansıtan içeriklerdir. Videolardan belirli aralıklarla kareler çıkarılmış ve bu kareler görüntü formatına dönüştürülmüştür. **Toplamda kullanılan 1300 adet görüntüden ratgele olarak %10'u test için ayrıldıktan sonra modelin eğitiminde kullanılmak üzere yine rastgele olarak %80 eğitim, %20 doğrulama(validasyon) şeklinde ikiye ayrılmıştır.** Elde edilen görseller,

yine CVAT platformu kullanılarak detaylı bir şekilde etiketlenmiştir. Bu süreçte, tüm drone ve kuşların konumları sınır kutuları ile belirtilmiş ve farklı ışık, açı ve çevresel faktörlerden kaynaklanan çeşitlilik veri setine eklenmiştir. Bu veri seti, modelin gerçek dünya senaryolarındaki performansını artırmayı amaçlamıştır.

Çalışmada kullanılan veri setine ait görsellerin örnekleri aşağıda sunulmuştur. Görseller, hem Kaggle veri setinden hem de internetten toplanan video verilerinden elde edilen görüntüleri içermektedir. Bu örnekler, drone ve kuşların farklı açılardan ve çevresel koşullardan yakalanmış görsellerini içermektedir. Etiketleme işlemi sırasında tüm nesneler dikkatlice işaretlenmiş ve sınır kutuları belirlenmiştir.



Şekil 4.1 Çalışmada kullanılan veri setlerinden örnek görseller.

Bu örnek görseller, kullanılan veri setinin çeşitliliğini ve modelin eğitiminde kullanılan verilerin özelliklerini vurgulamaktadır. Her iki veri seti de, farklı senaryoları ve zorluk seviyelerini (Işık, mesafe vs.) kapsayacak şekilde çeşitlendirilmiştir.

4.1.3 Verinin Etiketlenmesi

Bu tezde kullanılan veri setleri, yalnızca nesne tespiti görevleri için değil, aynı zamanda çoklu nesne izleme problemleri için de uygun bir temel sunmaktadır. Ancak, drone ve kuş görüntülerinden oluşan veri setlerinin hazırlanması, özellikle farklı koşulları içermesi açısından önemli bir çaba gerektirmiştir. Bu doğrultuda, hem Kaggle platformundan alınan hazır bir veri seti hem de internet üzerinden toplanan videolar, işlenerek çalışmaya uygun hale getirilmiştir.

Kullanılan veri setlerinde, nesne tespiti için gerekli olan anotasyonlar titizlikle oluşturulmuştur. Bu anotasyonlar, her bir görseldeki nesnelerin sınır kutularını doğru bir şekilde belirlemek için tasarlanmıştır. Veri setlerinin işlenmesi ve

etiketlenmesi sürecinde CVAT adlı GUI tabanlı bir anotasyon aracı kullanılmıştır. CVAT, veri setlerindeki tüm görsellerin hızlı ve doğru bir şekilde etiketlenmesini sağlamış ve bu süreç, veri setlerinin YOLOv8 algoritması ile eğitim için hazır hale getirilmesini mümkün kılmıştır[39].

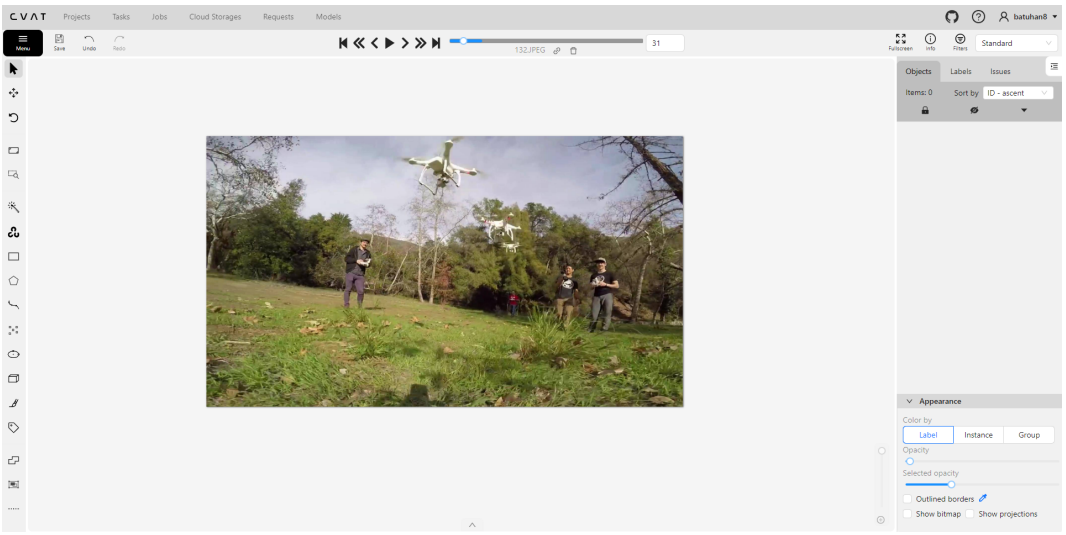
Anotasyonlu veriler, her tür denetimli öğrenme çözümü için hayati bir öneme sahiptir. Derin Sinir Ağları'nın (DNN) bir parçası olan Konvolüsyonel Sinir Ağları (CNN'ler), bilgisayarla görü alanındaki çoğu görev için çözümler sunmaktadır. CNN'ler, anotasyonlu görüntü ve videolar kullanılarak eğitilir. Ancak, bir veri setini etiketleme süreci oldukça zorlu ve zaman alıcı bir süreçtir. Örneğin, Oxford Üniversitesi'nin Visual Geometry Group (VGG) ekibi, ImageNet veri setinde yaklaşık 14 milyon görseli elle etiketlemek için geniş bir ekip oluşturmuş ve binlerce saat harcamıştır. Bu süreçte kullanılan yöntemler, günümüzde veri seti oluşturma alanında birçok araştırmanın temelini oluşturmuştur.[40]

Bu nedenle, veri setleri oluşturmanın, görüntü veya kareleri etiketlemek için daha verimli yöntemler bulmak ve daha kullanıcı dostu araçlar geliştirmek adına başlı başına bir araştırma alanıdır.

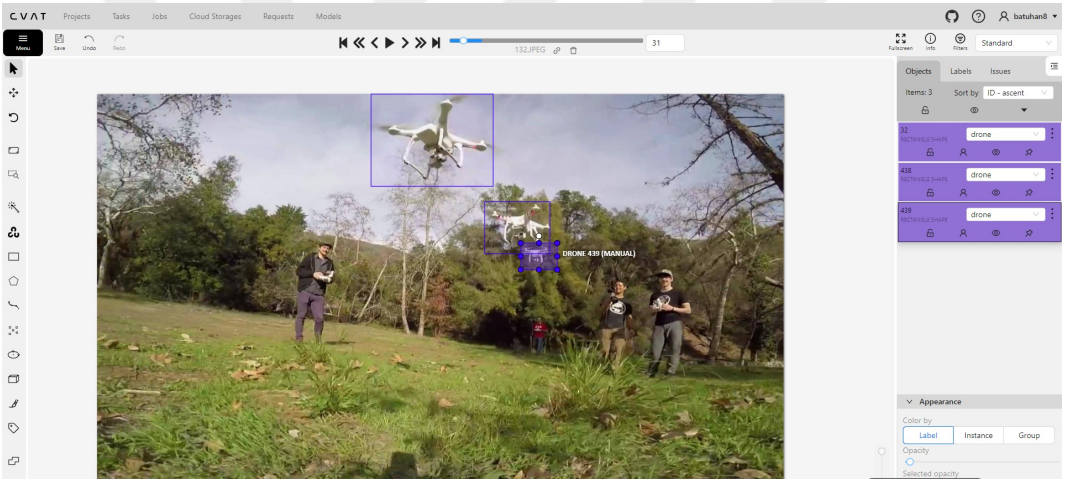
Bu tezin bir parçası olarak, açık kaynaklı çevrimiçi bir etiketleme aracı olan Computer Vision Annotation Tool (CVAT) incelenmiştir[39]. CVAT, görüntülerin ve video karelerinin veri etiketleme sürecinin verimliliğini artırmaya yardımcı olmaktadır. Ayrıca, çeşitli kuruluşların gizlilik politikaları, verilerine veya etiketlemelerine erişim izni vermeyebilir. Dolayısıyla, belirli bir görev için kendi veri setinizi etiketlemek, birçok durumda faydalı bir çözüm olabilmektedir.

Genel olarak, bilgisayarla görü araştırmacılarının yeni bir veri seti elde etmesinin iki yolu vardır:

Etiketleme konusunda uzmanlaşmış üçüncü taraf bir şirketle çalışmak. Ancak bu seçenek, veri doğrulama sorunlarına yol açabilir ve toplam etiketleme süresini artırabilir. Dahili bir veri etiketleme ekibi oluşturmak. Bu yöntem, işin kalitesi ile etiketleme süresine harcanan zaman arasında bir denge kurmayı kolaylaştırabilir. Bu veri setinin etiketleme sürecinde ikinci seçenek tercih edilmiş ve dahili bir ekip oluşturulmuştur. Bu süreçte Computer Vision Annotation Tool (CVAT) kullanılmıştır. CVAT, hem çevrimiçi hem de çevrimdışı olarak kullanılabilmektedir. CVAT'in resmi GitHub sayfasında da belirtildiği üzere, geliştiriciler, kullanıcı arayüzünü birçok veri seti etiketleme ekibinden alınan geri bildirimlere göre tasarlamıştır. Bu açık kaynaklı araç hakkında daha fazla bilgi için https://github.com/opencv/cvat ya da https://www.cvat.ai adresine başvurabilirsiniz.

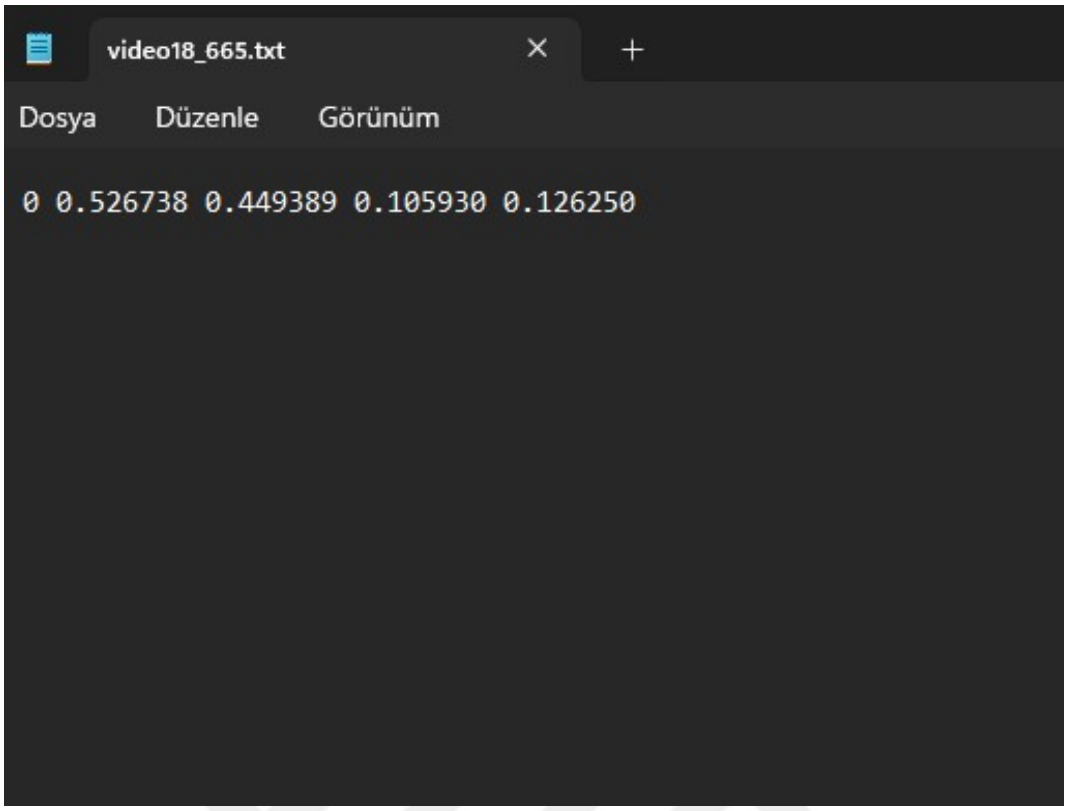


Şekil 4.2 Etiketlenmesi gereken görev ve iş seçildikten sonra CVAT ana ekranı görünümü.



Şekil 4.3 Etiketlendikten sonra CVAT ana ekranı.

Bu süreçte, CVAT tarafından oluşturulan anotasyon dosyası, genellikle bir metin dosyası ya da JSON formatında parametreler içeren bir dosya olarak sağlanır. Bu dosya, her bir görüntü veya video karesine ait sınıf bilgilerini, sınır kutusu koordinatlarını (bounding box) ve diğer ilgili parametreleri içerir. Örneğin, bir veriye ait anotasyonlar, aşağıdaki gibi bir yapı içerebilir:



Şekil 4.4 Etiketleme işlemi sonrasında oluşturulan örnek metin dosyası formatı.

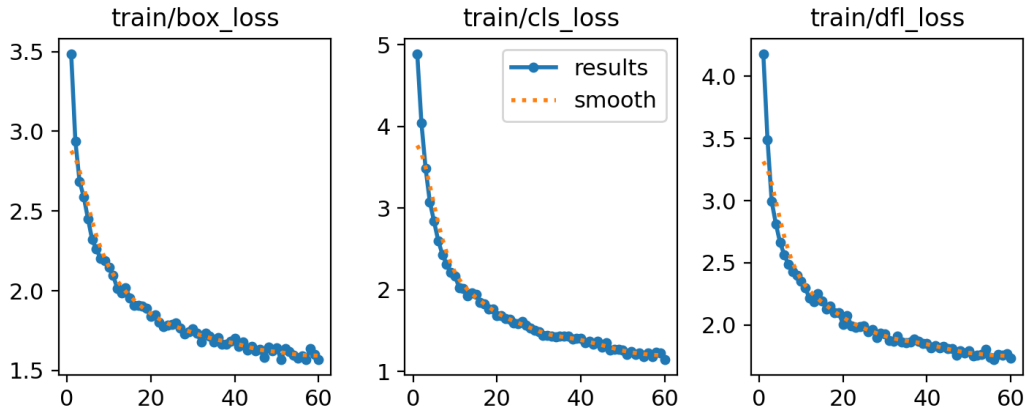
Etiketleme işleminin tamamlanmasının ardından, elde edilen veriler eğitim sürecine uygun hale getirilmiştir. CVAT aracı ile etiketlenen veriler, modelin eğitimine doğrudan kullanılabilecek bir formatta dışa aktarılmıştır. Bu aşamadan sonra, veriler, YOLOv8 algoritması gibi nesne tespiti modellerinin eğitimine hazırdır. Modelin eğitimine ilişkin detaylar bir sonraki bölümde ele alınacaktır.

4.2 Eğitim Süreci ve Sonuçları

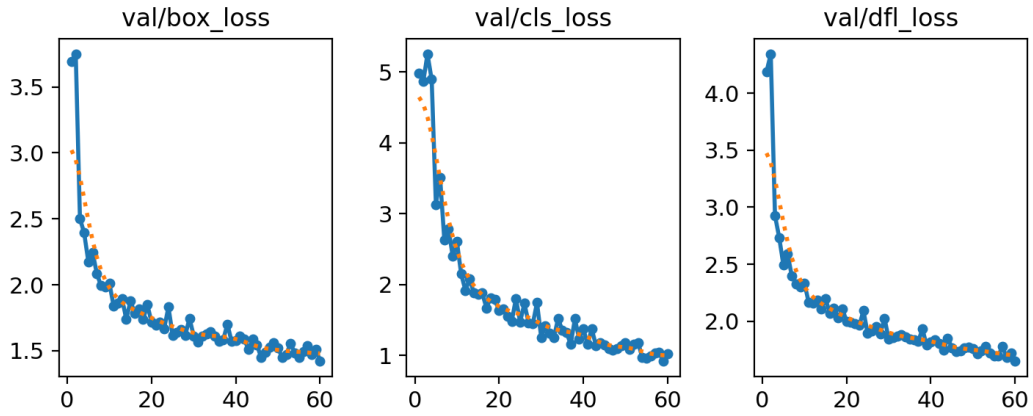
YOLOv8 modeli kullanılarak gerçekleştirilen eğitim sürecinde, eğitim parametreleri dikkatlice seçilerek modelin overfitting (aşırı öğrenme) ve underfitting (eksik öğrenme) gibi istenmeyen durumlarından kaçınılması hedeflenmiştir. Bu bağlamda, maksimum epoch değeri 300 olarak belirlenmiştir. Ancak, modelin gereksiz yere daha fazla eğitilmesinin önüne geçmek ve erken durdurma mekanizması sağlamak amacıyla patience=5 değeri kullanılmıştır.

Patience parametresi, modelin doğrulama kaybında (validation loss) belirli bir süre iyileşme görülmediği takdirde eğitimi durdurmasını sağlar. Bu değer 5 olarak seçilerek, modelin 5 epoch boyunca doğrulama kaybında iyileşme gözlenmemesi durumunda eğitimin otomatik olarak sonlanması sağlanmıştır. Bu, hem zaman

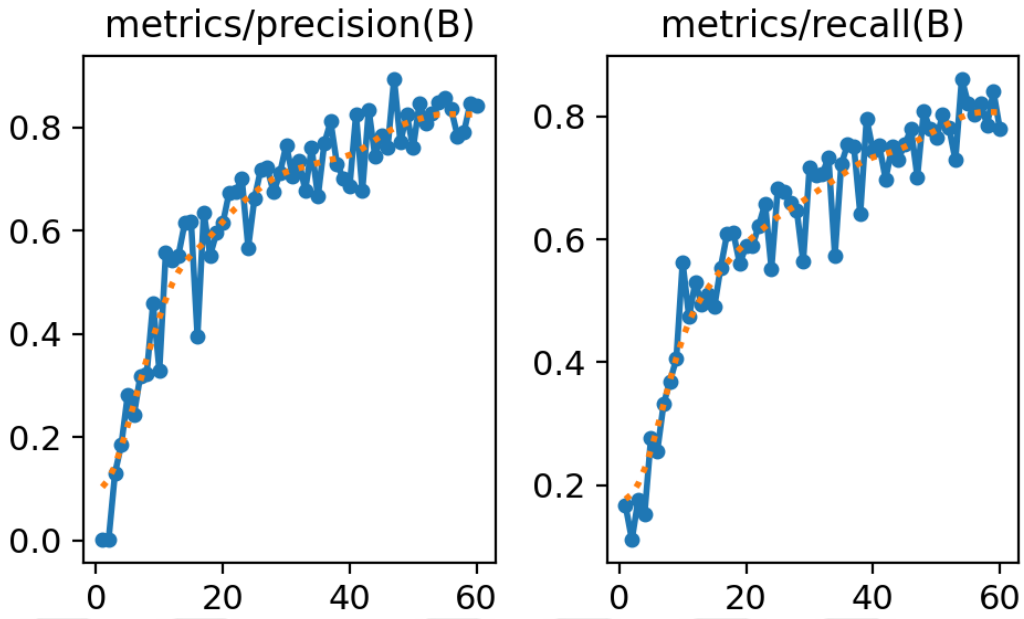
tasarrufu hem de modelin gereksiz yere aşırı öğrenme yapmasını önlemek açısından önemlidir.



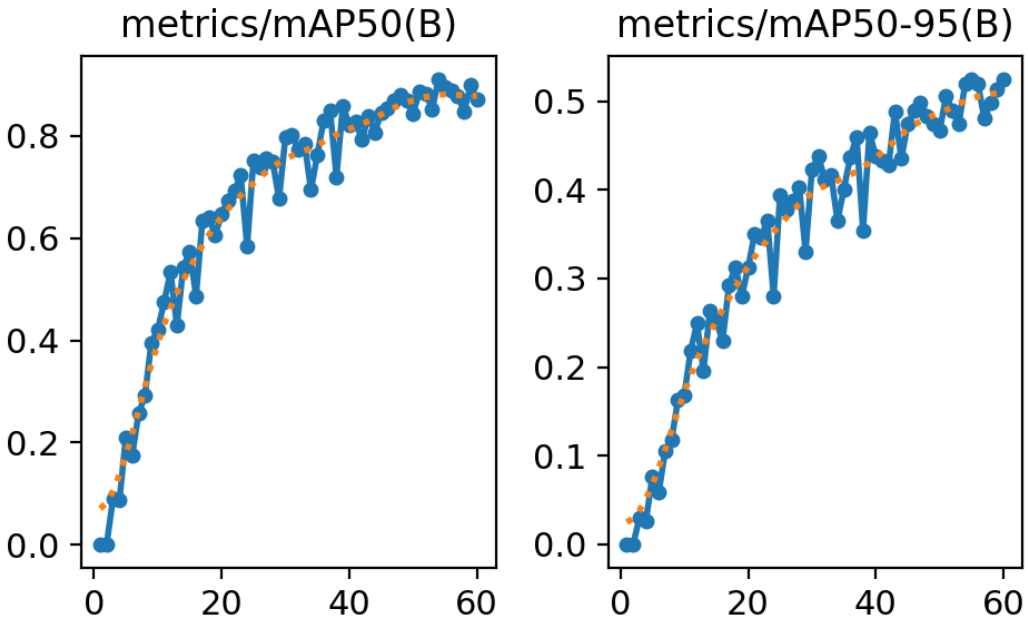
Şekil 4.5 Eğitim Sonuçları (Train)



Şekil 4.6 Eğitim Sonuçları (Validation)



Şekil 4.7 Eğitim Sonuçları (Precision ve Recall)



Şekil 4.8 Eğitim Sonuçları (mAP50 ve mAP50-95)

Eğitim süreci, sonuçları görselleştiren results grafiğinde de gözlemlendiği gibi, 60. epoch'ta sona ermiştir. Bu, patience değerinin devreye girdiği anlamına gelir ve modelin doğrulama kaybında daha fazla iyileşme göstermediği için eğitimin durdurulduğunu ifade eder. Böylece, optimum performansa ulaşılan epoch noktasında modelin eğitimi tamamlanmıştır. Bu yaklaşım, hem hesaplama kaynaklarının etkin kullanılmasını hem de modelin genel doğruluğunun

artırılmasını sağlamıştır.

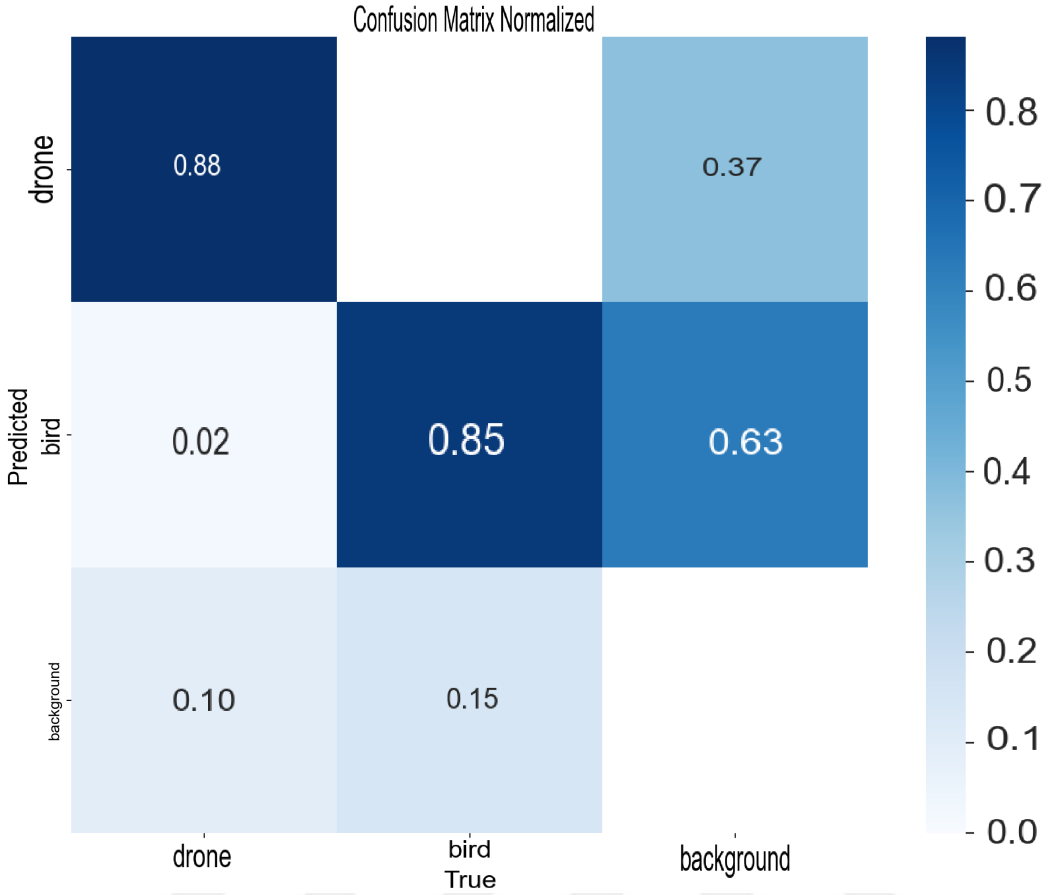
Eğitim süreci, sonuçları görselleştiren results grafiğinde de gözlemlendiği gibi, 60. epoch'da sona ermiştir. Bu, patience değerinin devreye girdiği anlamına gelir ve modelin doğrulama kaybında daha fazla iyileşme göstermediği için eğitimin durdurulduğunu ifade eder. Böylece, optimum performansa ulaşılan epoch noktasında modelin eğitimi tamamlanmıştır. Bu yaklaşım, hem hesaplama kaynaklarının etkin kullanılmasını hem de modelin genel doğruluğunun artırılmasını sağlamıştır.

Eğitim sonuçları, modelin eğitim sürecindeki performansını farklı metrikler üzerinden detaylı bir şekilde göstermektedir. Train/box loss, train/cls loss ve train/dfl loss metriklerinde, eğitim sürecinin başından itibaren sürekli bir azalma gözlemlenmiştir. Bu durum, modelin nesnelerin sınıflandırılması ve kutu tahminleme görevlerinde daha iyi bir performans sergilemek üzere optimize edildiğini göstermektedir. Ayrıca, doğrulama sırasında ölçülen val/box loss, val/cls loss ve val/dfl loss değerlerinde de benzer bir iyileşme eğilimi dikkat çekmektedir. Bu, eğitim sırasında elde edilen kazanımların doğrulama verisetinde de başarılı bir şekilde genelleştirildiğini ifade eder.

Eğitim sonuçlarında önemli bir diğer metrik olan metrics/precision(B) (doğruluk) ve metrics/recall(B) (duyarlılık) değerlerinde de belirgin bir artış gözlenmiştir. Precision eğrisi, modelin tahmin ettiği pozitif sonuçların doğru olma oranını ifade ederken, recall eğrisi, modelin mevcut tüm pozitif örnekleri ne kadar iyi tespit edebildiğini gösterir. Her iki metrikte de eğitimin ilerleyen safhalarında sabit bir artış eğilimi görülmesi, modelin performansının genel olarak başarılı olduğunu kanıtlar.

Metrics/mAP50(B) ve metrics/mAP50-95(B) grafikleri, modelin farklı IoU eşik değerlerinde ne kadar iyi tahmin yaptığını ortaya koymaktadır. Bu metrikler, modelin hem yüksek hem de düşük kesinlik gerektiren senaryolarda iyi bir performans sergilediğini gösterir. mAP (mean Average Precision) değerlerindeki artış, modelin eğitim süreci boyunca öğrenme yeteneğinin sürdürülebilir olduğunu ve eğitim verisetindeki örneklerin doğru bir şekilde genelleştirildiğini ifade etmektedir.

Son olarak, kayıp değerlerindeki azalma ile birlikte doğruluk ve duyarlılık metriklerindeki artış, modelin iyi bir eğitim sürecinden geçtiğini ve kullanılan parametrelerin etkili bir şekilde seçildiğini desteklemektedir. Bu sonuçlar, hem eğitim sırasında hem de doğrulama aşamasında modelin güvenilir ve doğru tahminler yapabildiğini açıkça ortaya koymaktadır.



Şekil 4.9 Eğitim Sonuçları (Confusion Matrix)

Yapılan eğitim sonuçlarının detaylı bir şekilde değerlendirilmesi için kullanılan karmaşıklık matrisi (Confusion Matrix), modelin sınıflandırma performansını görselleştirmek açısından oldukça önemlidir.

Drone sınıfı için:
- Doğru sınıflandırma oranı: %88
- Yanlış sınıflandırma oranları;
- Bird olarak yanlış sınıflandırma: %2
- Arka plan olarak yanlış sınıflandırma: %10

Bird sınıfı için:
- Doğru sınıflandırma oranı: %85
- Yanlış sınıflandırma oranları;
- Drone olarak yanlış sınıflandırma: %0
- Arka plan olarak yanlış sınıflandırma: %15

Sonuçlar, modelin drone tespitinde üstün bir performans sergilediğini açıkça ortaya koymaktadır. Ancak, Bird sınıfında arka plan ile karışıklık oranının (%15) nispeten daha yüksek olduğu gözlemlenmiştir. Bu durum, özellikle arka plan özelliklerinin bazı kuş özelliklerine benzerlik göstermesi ve bu benzerliğin modelin genelleştirme yeteneğini sınamasıyla ilişkilendirilebilir. Bu tür durumlar, genellikle eğitim veri setinin sınıf çeşitliliği ve derinliğinin artırılmasıyla iyileştirilebilmektedir.

Modelin performansı üzerine yapılan analiz, her sınıf için genel başarı oranlarını ve sınıflar arasındaki karışıklık oranlarını net bir şekilde göstermektedir. Özellikle Bird sınıfında karşılaşılan karışıklıklar, arka plan tespitine yönelik özel augmentasyon yöntemlerinin uygulanması veya eğitim veri setine daha çeşitli örneklerin eklenmesiyle azaltılabilir. Bu tür önlemler, modelin genel doğruluğunu ve genelleştirme kapasitesini artırmak adına etkili olacaktır.

Ek olarak, bu çalışmada kullanılan sistem, yapay zeka destekli tespit mekanizmasını klasik piksel değişimi analiziyle destekleyerek önemli bir avantaj sağlamıştır. Modelin drone tespitindeki performansının yanı sıra, yapay zeka tabanlı takibin devre dışı kaldığı durumlarda, piksel değişim analizi sayesinde takip sürecinin kesintiye uğramaması sağlanmaktadır. Bu iki yöntemin entegrasyonu, özellikle düşük ışık koşulları veya karmaşık arka plan özellikleri gibi zorlayıcı durumlarda sistemin güvenilirliğini artırmakta ve hedeflerin sürekli olarak takip edilebilmesini garanti etmektedir. Bu yaklaşım, sistemin operasyonel etkinliğini artırmak ve kullanım senaryolarında daha geniş bir uygulama alanı sunmak açısından kritik bir rol oynamaktadır.



Şekil 4.10 Takip entegrasyonu devre dışı kaldığı durumlarda, piksel değişim analizi sayesinde takip süreci devam etmektedir.

Bu sonuçlar ve öneriler, modelin genelleme yeteneğini geliştirmeye yönelik bazı fikirler sunmakta olup, yapay zeka destekli sistemlerin daha verimli kullanılmasına katkı sağlayabilir.

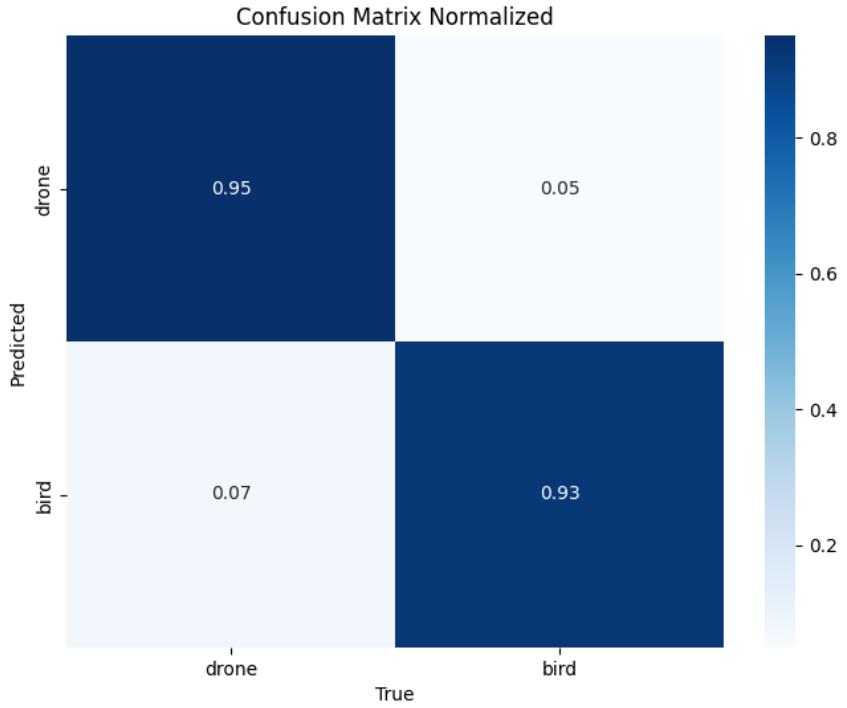
4.3 Test Süreci ve Sonuçları

Test süreci, modelin eğitim sırasında öğrendiği bilgileri gerçek dünya senaryolarında nasıl uygulayabildiğini değerlendirmek amacıyla iki farklı test yöntemi ile gerçekleştirilmiştir. İlk test yöntemi, eğitim veri setinden rastgele seçilen %10'luk bir örneklem kümesi kullanılarak yapılmıştır. Bu veri kümesinde her resimde yalnızca bir adet kuş veya drone bulunmakta olup, toplamda veri setinin %10'unu temsil etmektedir. İkinci test yöntemi ise, rastgele seçilen videolardan elde edilen durağan görüntüler üzerinde gerçekleştirilmiştir. Bu test, modelin farklı açılardan, aydınlatma koşullarından ve çevresel faktörlerden etkilenerek ne ölçüde doğru tahminler yapabildiğini analiz etmek amacıyla uygulanmıştır. Aşağıdaki bölümlerde her iki test yönteminin detayları ve elde edilen sonuçlar açıklanmaktadır.

4.3.1 Görüntü Bazlı Test

Bu test, eğitim veri setinden rastgele seçilen %10'luk bir görüntü alt kümesi üzerinde gerçekleştirilmiştir. Her bir test görüntüsünde yalnızca bir adet kuş veya drone bulunmaktadır. Bu testin amacı, modelin standart koşullarda, veri setinden alınan bireysel görüntüler üzerindeki performansını değerlendirmektir. Veri setinden alınan bu örnekler, doğrudan eğitim veri setine benzer özellikler taşıdığı için ışık, açı veya arka plan değişkenliklerinin etkisinin daha az olacağı öngörülmüştür. Dolayısıyla, modelin teorik olarak en yüksek performans göstermesi beklenmektedir.

Aşağıda, bu testin sonuçlarını içeren normalize edilmiş Confusion Matrix sunulmuştur:



Şekil 4.11 Test sonuçları (Confusion Matrix)

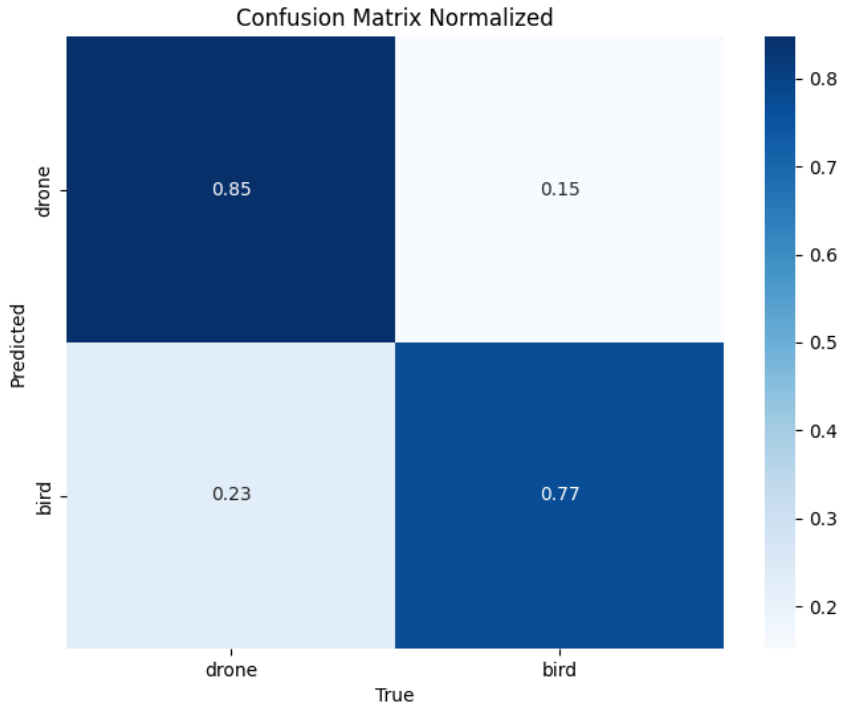
Drone sınıfı için:
- Doğru sınıflandırma oranı: %95
- Yanlış sınıflandırma oranı %5
- Sensitivity: %94
- Specificity: %93

Bird sınıfı için:
- Doğru sınıflandırma oranı: %93
- Yanlış sınıflandırma oranı %7
- Sensitivity: %92
- Specificity: %95

4.3.2 Video Bazlı Test

Video bazlı test sürecinde, farklı senaryolarda çekilmiş rastgele videolardan belirli kareler seçilerek modelin performansı değerlendirildi. Bu yöntemde, gerçek dünya

koşullarına daha yakın bir test ortamı oluşturulması amaçlanmıştır. Videolardan alınan kareler, farklı açılar, ışık koşulları ve hareketli nesneler içermesi nedeniyle modelin genel performansını ölçmek için önemlidir. Bu test, modelin dinamik ortamlarda ve değişen görüntü koşullarında nasıl performans gösterdiğini analiz etmek için gerçekleştirilmiştir.



Şekil 4.12 Test sonuçları (Confusion Matrix)

Drone sınıfı için:
- Doğru sınıflandırma oranı: %85
- Yanlış sınıflandırma oranı %15
- Sensitivity: %83
- Specificity: %82

Bird sınıfı için:
- Doğru sınıflandırma oranı: %77
- Yanlış sınıflandırma oranı %23
- Sensitivity: %75
- Specificity: %80

Bu test, yalnızca tekil kareler üzerinden yapılan tahminlere dayanmaktadır ve herhangi bir piksel değişimi tabanlı takip mekanizması içermemektedir. Ancak, hareketli nesnelerin takibini sağlayan bir takip mekanizması entegre edildiğinde, modelin doğruluk oranı önemli ölçüde artmaktadır. Takip algoritmaları, nesnenin bir önceki karedeki konum ve hareket bilgilerini kullanarak daha tutarlı tahminler yapılmasını sağlar. Böylece, özellikle video bazlı testlerde gözlemlenen yanlış sınıflandırma oranları düşmekte ve genel model performansı artmaktadır.

5
SONUÇ

Bu tez çalışması, sabit ve hareketli askeri birliklerin asimetrik drone tehditlerine karşı korunması amacıyla geliştirilmiştir. Özellikle, klasik piksel değişimi takibi yönteminin optimize edilerek güçlendirilmesi, tespit ve takip sistemlerinin etkinliğini artırmada kritik bir rol oynamıştır. Önerilen sistem, drone ve kuş gibi farklı nesneleri algılayıp takip edebilme kapasitesiyle, sahada karşılaşılabilecek çeşitli tehdit senaryolarında kullanılabilecek şekilde tasarlanmıştır.

5.1 Karşılaştırmalı Sonuç Analizi

Bu doğrultuda, geliştirilen sistemin etkinliğini değerlendirmek amacıyla literatürdeki benzer çalışmalar incelenmiş ve kullanılan yöntemler, veri setleri ile elde edilen doğruluk oranları açısından karşılaştırmalı bir analiz gerçekleştirilmiştir. Drone tespiti üzerine yapılan akademik araştırmalar, genellikle derin öğrenme tabanlı nesne tespit algoritmalarını, görüntü işleme tekniklerini ve çeşitli takip mekanizmalarını içeren farklı yaklaşımlar üzerine yoğunlaşmaktadır. Özellikle YOLO mimarisi, CNN tabanlı modeller ve gelişmiş takip algoritmaları, bu alanda sıkça tercih edilen yöntemler arasında yer almaktadır. Ancak, her çalışmanın kullanılan veri seti, model mimarisi ve test koşulları farklı olduğundan, elde edilen performans metrikleri de değişkenlik göstermektedir.

Önerilen sistem, klasik piksel değişimi takibi ile YOLOv8 mimarisini birleştirerek nesne tespitinde doğruluğu artırmayı amaçlamıştır. Literatürdeki diğer çalışmalarla kıyaslandığında, özellikle drone ve kuş gibi hareketli nesnelerin ayrımını yapma konusunda yüksek başarı gösterdiği görülmektedir. Bununla birlikte, modelin gerçek zamanlı tespit kapasitesi ve doğruluk oranı, mevcut çözümlerle kıyaslandığında umut verici sonuçlar ortaya koymaktadır. Optimizasyon tekniklerinin entegrasyonu sayesinde, işlem süresi ve doğruluk oranı açısından belirli avantajlar sağlanmış olup, sistemin saha koşullarında uygulanabilirliği artırılmaya çalışılmıştır.

literatürdeki çalışmaların yöntemleri ve sonuçları incelenerek yapılan bu kıyaslamanın detayları, aşağıdaki tabloda özetlenmiştir:

Referans Numarası	**Kullanılan Yöntemler**	**Veri Seti**	**En Yüksek Doğruluk Oranı**
[41]	Dört farklı tespit mimarisi ve üç takip çerçevesi	241 video (331.486 görüntü)	%98,6
[42]	SqueezeNet, MobileNetV2, ResNet18, ResNet50	Drone vs. Bird Detection veri seti	%77,0 (SqueezeNet ile)
[43]	YOLOv7, YOLOv8	YOLO Drone Detection Dataset	%50,16 (YOLOv8 ile)
Önerilen Yöntem	YOLOv3, YOLOv5, YOLOv7, YOLOv8 + Klasik Yöntemler	Drone vs. Bird Detection + internetten toplanan 1300 görüntü	%95

Tablo 5.1 Tablo 5.1 Karşılaştırmalı Sonuç Analizi

5.2 Uygulama ve İyileştirme Önerileri

Sistem, hafif zırhlı araçlar üzerine veya binaların üzerine yerleştirilen optik algılama sistemleriyle entegre edilmek üzere geliştirilmiştir. Optik algılama sistemleri, özellikle drone tespiti için yüksek doğruluk oranları sağlamaktadır. Bunun yanında, bu sistemlerin yanına entegre edilen parçacıklı mühimmat kullanan silahlarla birleştiğinde, etkili bir savunma mekanizması oluşturulması hedeflenmiştir. Şekil 5.1'te gösterilen klasik yapıyla karşılaştırıldığında Şekil 5.2'deki mimari hem otonom çalışabilecek hemde operatörün can güvenliğini sağlayacaktır. Hem hareketli hem de sabit hedeflerin korunması açısından önemli bir potansiyel taşımaktadır.



Şekil 5.1 Klasik Anti-Drone Sistemleri[44].



Şekil 5.2 Mimarinin Entegre Edilmesi Önerilen Sistem[45] .

Modelin doğruluğunu artırmak ve genelleme kapasitesini geliştirmek için daha büyük ve daha çeşitli bir veri setine ihtiyaç duyulmaktadır. Farklı coğrafi bölgelerden, çeşitli ışık koşullarında ve değişken arka plan özellikleriyle toplanan görüntüler, modelin performansını önemli ölçüde iyileştirebilir. Bu tür bir genişleme, modelin daha çeşitli senaryolarda daha iyi sonuçlar vermesine olanak tanıyacaktır.

Sistemin gerçek zamanlı çalışmalarda daha etkili olması için performans optimizasyonu kritik öneme sahiptir. YOLOv8 algoritması, 35-40 FPS hızında çalışabilen tek aşamalı nesne tespiti modellerinden biri olarak bu gereksinimi büyük ölçüde karşılamaktadır[6] . Modelin gerçek zamanlı kullanımda işlem hızını artırmak için GPU hızlandırma ve algoritma optimizasyon teknikleri uygulanabilir.

Böylelikle, sahada etkinliği artırmak ve daha yüksek doğrulukla tehditlerin tespitini sağlamak mümkün olacaktır.

Ayrıca optik algılama sistemlerine ek olarak, termal kameralar ve radar sistemleri gibi farklı sensörlerin entegrasyonu ve buna bağlı veri setlerinin eğitilmesi, sistemin farklı çevresel koşullarda daha iyi performans göstermesini sağlayabilir. Sensör füzyonu, hem tespit doğruluğunu artıracak hem de sahte alarmların azaltılmasına katkıda bulunacaktır.

Sonuç olarak, bu çalışma asimetrik tehditlerin algılanması ve önlenmesi konusunda önemli bir katkı sağlamış ve gelecekteki çalışmalara ışık tutacak bir temel oluşturmuştur. Geliştirilen sistem, sahada uygulanabilir bir çözüm sunmakta ve tehditlerin bertaraf edilmesinde etkili bir mekanizma olarak öne çıkmaktadır.

KAYNAKÇA

[1] A. Krizhevsky, I. Sutskever, G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Communications of the ACM*, c. 60, no. 6, ss. 84–90, Haz. 2017. doi: 10.1145/3065386.

[2] B. M. Albaba, S. Özer, "Synet: An Ensemble Network for Object Detection in UAV Images," *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2020, ss. 407–412. doi: 10.1109/ICMLA51294.2020.00071.

[3] Z. Yan, "Beyond 3D Siamese Tracking: A Motion-Centric Paradigm for 3D Single Object Tracking in Point Clouds," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, ss. 8856–8865. doi: 10.1109/CVPR52688.2022.00863.

[4] S. Ervural, M. Ceylan, "Kızılötesi Termal Görüntülemenin Yapay Zekâ Tabanlı Biyomedikal Uygulamalarına Bakış," *Sağlık Bilimlerinde Yapay Zeka Dergisi*, c. 1, no. 1, ss. 24–34, 2021. doi: 10.52309/jaihs.2021.5.

[5] R. K. Sinha, B. Chaurasia, P. K. Gupta, "Deep Learning for Computer Vision Tasks: A Review," *arXiv preprint*, 2018. arXiv: 1804.03928 [cs.CV].

[6] Ultralytics, *"YOLOv8 Live: Real-Time Object Detection"*, Erişim Tarihi: 10 Mart 2025, 2023. erişim adresi: https://ultralytics.com/yolov8.

[7] A. Kara, M.Yılmaz, "Silahlı İnsansız Hava Araçlarının Uluslararası Alanda ve İç Güvenlikte Sevk ve İdaresine İlişkin Hukuki Saptamalar," *Ankara Hacı Bayram Veli Üniversitesi Hukuk Fakültesi Dergisi*, c. 24, no. 3, ss. 229–281, 2020. erişim adresi: https://dergipark.org.tr/tr/pub/ahbvuhfd/issue/57329/812038.

[8] Z.-Q. Zhao, P. Zheng, S.-t. Xu X. Wu, "Object Detection with Deep Learning: A Review," *IEEE Transactions on Neural Networks and Learning Systems*, c. 30, no. 11, ss. 3212–3232, 2019. doi: 10.1109/TNNLS.2018.2876865.

[9] M. Piskin, *"Nesne Tespiti ve Nesne Tanıma"*, Erişim Tarihi: 10 Mart 2025, 2025. erişim adresi: https://mesutpiskin.com/blog/nesne-tespiti-ve-nesne-tanima.html.

[10] A. S. A. Al-Jumaily, A. A. Al-human, H. M. Al-Mashhadi, "Optimization Contrast Enhancement and Noise Reduction for Semantic Segmentation of Aerial Imagery," *International Journal of Intelligent Engineering and Systems*, c. 16, no. 1, ss. 599–608, 2023. doi: 10.22266/ijies2023.0228.51.

[11] N. Kandemir, *"Histogram Eşitleme & CLAHE"*, Erişim Tarihi: 10 Mart 2025, 2025. erişim adresi: https://medium.com/@nupelkandemir/histogram-e%C5%9Fitleme-clahe-9e01cf75da25.

[12] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, c. 60, no. 2, ss. 91–110, 2004. doi: 10.1023/B:VISI.0000029664.99615.94.

[13] N. Dalal, B.Triggs, "Histograms of Oriented Gradients for Human Detection," *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, c. 1, 2005, ss. 886–893. doi: 10.1109/CVPR.2005.177.

[14] H. Bay, A. Ess, T. Tuytelaars, L. V. Gool, "SURF: Speeded Up Robust Features," *Computer Vision and Image Understanding*, c. 110, no. 3, ss. 346–359, 2008. doi: 10.1016/j.cviu.2007.09.014.

[15] A. S. A. Al-Jumaily, A. A. Al-human H. M. Al-Mashhadi, "Optimization Contrast Enhancement and Noise Reduction for Semantic Segmentation of Aerial Imagery," *International Journal of Intelligent Engineering and Systems*, c. 16, no. 1, ss. 599–608, 2023. doi: 10.22266/ijies2023.0228.51.

[16] A. V. Tatachar, D. Dheeraj, "An Automated Attendance System Based on the Concept of Histogram of Oriented Gradients," *Journal of Image Processing and Artificial Intelligence*, c. 6, no. 3, ss. 13–21, 2020. doi: 10.46610/JOIPAI.2020.v06i03.003.

[17] M. T. Ağdaş, S. Gülseçen, "Güvenlik Kameralarında Otomatik Silah ve Bıçak Tespit Sistemi: Karşılaştırmalı YOLO Modelleri," *Avrupa Bilim ve Teknoloji Dergisi*, c. 41, ss. 16–22, 2022. doi: 10.31590/ejosat.1163675.

[18] K. Aki, A. E. Dirik, "Derin Öğrenme Tabanlı ve PID Kontrol Tabanlı Sürücüsüz Araç Sistemleri," *Mühendislik Bilimleri ve Tasarım Dergisi*, c. 8, no. 5, ss. 306–316, 2020. doi: 10.21923/jesd.829598.

[19] S. Çelebi, *Otonom Araçlar İçin TensorFlow.js Tabanlı Nesne Tespiti Uygulaması*, Bitirme Projesi, İstanbul Arel Üniversitesi, 2023. erişim adresi: https://www.academia.edu/124499799/Otonom_ara%C3%A7lar_i%C3%A7in_Tensorflow_js_tabanl%C4%B1_nesne_tespiti_uygulamas%C4%B1.

[20] SentiSight.ai, *"Annotating Bounding Boxes for Object Detection"*, Erişim Tarihi: 10 Mart 2025, 2025. erişim adresi: https://www.sentisight.ai/annotating-bounding-boxes-for-object-detection/.

[21] R. Girshick, J. Donahue, T. Darrell, J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, ss. 580–587. doi: 10.1109/CVPR.2014.81.

[22] R. Girshick, "Fast R-CNN," *Proceedings of the IEEE international conference on computer vision*, 2015, ss. 1440–1448. doi: 10.1109/ICCV.2015.169.

[23] S. Ren, K. He, R. Girshick, J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *Advances in Neural Information Processing Systems 28 (NIPS 2015)*, 2015, ss. 91–99.

[24] K. He, X. Zhang, S. Ren, J. Sun, "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition," *European conference on computer vision*, Springer, Cham, 2014, ss. 346–361. doi: 10.1007/978-3-319-10578-9_23.

[25] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan S. Belongie, "Feature Pyramid Networks for Object Detection," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, ss. 2117–2125. doi: 10.1109/CVPR.2017.106.

[26] N. Dalal, B.Triggs, "Histograms of oriented gradients for human detection," *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, c. 1, 2005, ss. 886–893. doi: 10.1109/CVPR.2005.177.

[27] J. Redmon, A. Farhadi, "YOLOv3: An Incremental Improvement," *arXiv preprint*, 2018. arXiv: 1804.02767 [cs.CV].

[28] G. Jocher, *"ultralytics/yolov5: v6.0 - YOLOv5n 'Nano' models, Roboflow integration, TensorFlow export, OpenCV DNN support"*, https://github.com/ultralytics/yolov5, v6.0 Release, Ekim 2021.

[29] S. S. A. Zaidi, M. S. Ansari, A. Aslam, N. Kanwal, M. Asghar B.-W. Lee, "A survey of modern deep learning based object detection models," *Digital Signal Processing*, c. 126, s. 103 514, 2022. doi: 10.1016/j.dsp.2022.103514.

[30] J. Redmon, A. Farhadi, "YOLO9000: Better, Faster, Stronger," *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, ss. 7263–7271. doi: 10.1109/CVPR.2017.690.

[31] Z. Ge, S. Liu, F. Wang, Z. Li, J. Sun, "YOLOX: Exceeding YOLO Series in 2021," *arXiv preprint*, 2021. arXiv: 2107.08430 [cs.CV].

[32] A. Bochkovskiy, C. Wang, H. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," *arXiv preprint*, 2020. arXiv: 2004.10934 [cs.CV].

[33] D. Thuan, *"Evolution of YOLO Algorithm and YOLOv5: The State-of-the-Art Object Detection Algorithm"*, Published on Medium, 2021.

[34] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, ss. 779–788. doi: 10.1109/CVPR.2016.91.

[35] J. Redmon, A. Farhadi, *"YOLOv3: An Incremental Improvement"*, Accessed: December 2024, 2018. arXiv: 1804.02767 [cs.CV].

[36] A. T. Cong, H. T. Nguyen, V. L. N. Huu, N. D. Trung, "Pesticide Label Detection Using Bounding Prediction-Based Deep Convolutional Networks," *Future Data and Security Engineering*, seri Lecture Notes in Computer Science, c. 13057, Springer, Cham, 2021, ss. 195–210. doi: 10.1007/978-3-030-91387-8_13.

[37] Personal Data Protection Commission Singapore, *"Managing Personal Data"*, Erişim Tarihi: 10 Mart 2025, n.d. erişim adresi: https://www.pdpc.gov.sg/guidelines-and-e-learning/managing-personal-data.

[38] Kişisel Verileri Koruma Kurumu, *"Kişisel Verilerin Korunması Kanunu"*, Kanun Numarası: 6698, 2016. erişim adresi: https://www.mevzuat.gov.tr/mevzuatmetin/1.5.6698.pdf.

[39] B. Sekachev, *"OpenCV/CVAT: Computer Vision Annotation Tool"*, https://github.com/openvinotoolkit/cvat, v1.1.0 Release, Ağus. 2020.

[40] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A, Karpathy, A. Khosla, M. Bernstein, A. C. Berg, F. Li, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision*, c. 115, no. 3, ss. 211–252, 2015. doi: 10.1007/s11263-015-0816-y.

[41] B. K. S. Isaac-Medina, M. Poyser, D. Organisciak, C. G. Willcocks, T. P. Breckon H. P. H. Shum, "Unmanned Aerial Vehicle Visual Detection and Tracking using Deep Neural Networks: A Performance Benchmark," *arXiv preprint*, 2021. arXiv: 2103.13933 [cs.CV].

[42] H. J. A. Dawasari, M. Bilal, M. Moinuddin, K. Arshad K. Assaleh, "DeepVision: Enhanced Drone Detection and Recognition in Visible Imagery through Deep Learning Networks," *Sensors*, c. 23, no. 21, s. 8711, 2023. doi: 10.3390/s23218711.

[43] K. Aggarwal, *"Performance Analysis of YOLOv7 and YOLOv8 Models for Drone Detection"*, https://github.com/KhushiAgg/Performance-Analysis-of-YOLOv7-and-YOLOv8-Models-for-Drone-Detection, GitHub Repository, 2024.

[44] The Times of India, *"India warns arms companies of action for default in offset commitments"*, Erişim Tarihi: 10 Mart 2025, Ağus. 2021. erişim adresi: https://timesofindia.indiatimes.com/india/india-warns-arms-companies-of-action-for-default-in-offset-commitments/articleshow/85752556.cms.

[45] Babel, *"Ukraine buys about a hundred RMS combat modules from Australia, the value of the contract is $80 million"*, Erişim Tarihi: 10 Mart 2025, 2025. erişim adresi: https://babel.ua/en/news/92345-ukraine-buys-about-a-hundred-rms-combat-modules-from-australia-the-value-of-the-contract-is-80-million.

TEZDEN ÜRETİLMİŞ YAYINLAR

Konferans Bildirisi

1. B. Baysal, "Görüntü İşleme ve Makine Öğrenimi Yöntemleriyle Drone Tespiti," in *GradColloquium'24: Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Lisansüstü Araştırmalar Kolokyumu, Özet Kitabı*, İstanbul, Türkiye, 24 Mayıs 2024, s. 41. [Çevrimiçi]. Erişim adresi: https://fbe.yildiz.edu.tr/sayfa/17/GradColloquium/467