

**T.C.**  
**FIRAT ÜNİVERSİTESİ**  
**FEN BİLİMLERİ ENSTİTÜSÜ**



**DERİN ÖĞRENME YÖNTEMLERİNİ KULLANARAK İHA  
GÖRÜNTÜLERİNDEN ÇEVRESEL ATIK TESPİTİ**

**Serkan ÇELİK**

Yüksek Lisans Tezi

EKOBİLİŞİM ANABİLİM DALI

Ekobilisim Bilim Dalı

EKİM 2022

**T.C.**  
**FIRAT ÜNİVERSİTESİ**  
**FEN BİLİMLERİ ENSTİTÜSÜ**

Ekobilişim Anabilim Dalı

Yüksek Lisans Tezi

**DERİN ÖĞRENME YÖNTEMLERİNİ KULLANARAK İHA**  
**GÖRÜNTÜLERİNDEN ÇEVRESEL ATIK TESPİTİ**

Tez Yazarı  
**Serkan ÇELİK**

Danışman  
Dr. Öğr. Üyesi Ayhan ALTINÖRS

EKİM 2022  
ELAZIĞ

**T.C.**  
**FIRAT ÜNİVERSİTESİ**  
**FEN BİLİMLERİ ENSTİTÜSÜ**

Ekobilşim Anabilim Dalı

Yüksek Lisans Tezi

---

Başlığı: Derin Öğrenme Yöntemlerini Kullanarak İHA Görüntülerinden Çevresel Atık Tespiti

Yazarı: Serkan ÇELİK

İlk Teslim Tarihi: 12.09.2022

Savunma Tarihi: 07.10.2022

---

**TEZ ONAYI**

Fırat Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına göre hazırlanan bu tez aşağıda imzaları bulunan jüri üyeleri tarafından değerlendirilmiş ve akademik dinleyicilere açık yapılan savunma sonucunda OYBİRLİĞİ ile kabul edilmiştir.

*İmza*

Danışman: Dr. Öğr. Üyesi Ayhan ALTINÖRS Onayladım  
Fırat Üniversitesi, Teknik Bilimler Meslek Yüksek Okulu

---

Başkan: Doç. Dr. Orhan YAMAN Onayladım  
Fırat Üniversitesi, Teknoloji Fakültesi

---

Üye: Dr. Öğr. Üyesi Mehmet ÜSTÜNDAĞ Onayladım  
Malatya Turgut Özal Üniversitesi, Mühendislik ve Doğa Bilimleri Fakültesi

---

Bu tez, Enstitü Yönetim Kurulunun ...../...../20..... tarihli toplantısında tescillenmiştir.

*İmza*

Prof. Dr. Burhan ERGEN  
Enstitü Müdürü

## BEYAN

Fırat Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırladığım “Derin Öğrenme Yöntemlerini Kullanarak İHA Görüntülerinden Çevresel Atık Tespiti” Başlıklı Yüksek Lisans Tezimin içindeki bütün bilgilerin doğru olduğunu, bilgilerin üretilmesi ve sunulmasında bilimsel etik kurallarına uygun davrandığımı, kullandığım bütün kaynakları atıf yaparak belirttiğimi, maddi ve manevi desteği olan tüm kurum/kuruluş ve kişileri belirttiğimi, burada sunduğum veri ve bilgileri unvan almak amacıyla daha önce hiçbir şekilde kullanmadığımı beyan ederim.

07.10.2022

**Serkan ÇELİK**



# ÖNSÖZ

Günümüzde teknolojinin ilerlemesi ve artan ihtiyaçlar doğrultusunda birçok alanda İnsansız Hava Araçları kullanılmaya başlanmıştır. Bu alanlardan biri de nesne tespit sistemleri için görüntü çekimdir. Teknolojinin gelişmesiyle nesnelerin tespit edilmesi ve tespit edilen nesnelere kullanılarak sorunlara çözümler bulunması günümüzde oldukça yaygınlaşmıştır. Teknolojinin bu yönde ilerlemesine, gelişen donanım teknolojileri ve geliştirilen nesne tanıma algoritmalarının artmasının büyük etkisi olmuştur. Bununla birlikte bu alanda yapılan çalışmalar artmış ve nesne tespiti konusunda yeni teknikler, algoritmalar ve yöntemler geliştirilmiştir. Bu tez çalışmasında, teknolojinin bu yöndeki gelişmelerinden ilham alarak İnsansız Hava Araçları ile doğal ortamda çekilen görüntüler kullanılarak ekosistem için en büyük problemlerden biri olan pet ve cam şişe atıklarının tespiti derin öğrenme yöntemleri ile gerçekleştirilmiştir. Hazırlanan bu tez çalışmasının bu amaç doğrultusunda bundan sonraki çalışmalara fayda sağlayacağı düşüncesindeyim.

Bu çalışmada değerli vaktini bana ayırarak çalışmamın her aşamasında desteğini ve yardımlarını esirgemeyen danışman hocam sayın Dr. Öğr. Üyesi Ayhan ALTINÖRS'e teşekkürlerimi sunarım.

Ayrıca, hayatımın her alanında ilgi, anlayış ve her türlü desteğini esirgemeyen eşim ve aileme teşekkür ederim.

**Serkan ÇELİK**  
ELAZIĞ, 2022

# İÇİNDEKİLER

Sayfa

ÖNSÖZ.....	iv
İÇİNDEKİLER .....	v
ÖZET .....	vi
ABSTRACT .....	vii
ŞEKİLLER LİSTESİ .....	viii
TABLolar LİSTESİ .....	x
KISALTMALAR .....	xi
<b>1. GİRİŞ .....</b>	<b>1</b>
<b>2. LİTERATÜR ÖZETİ .....</b>	<b>2</b>
<b>3. PET VE CAM ŞİŞELERİN ÇEVREYE ETKİLERİ .....</b>	<b>8</b>
<b>4. İNSANSIZ HAVA ARAÇLARI.....</b>	<b>11</b>
4.1. DJI Mavic Air 2 İHA.....	13
<b>5. DERİN ÖĞRENME .....</b>	<b>16</b>
5.1. Evrişimsel Sinir Ağları .....	18
5.2. Hızlı Bölge Bazlı Evrişimsel Sinir Ağı (Faster R-CNN).....	19
5.3. Tek Atış Nesne Algılama (SSD) Algoritması.....	20
5.4. YOLO Nesne Algılama Algoritması .....	20
5.4.1. YOLOv3.....	25
<b>6. MATERYAL VE METOT .....</b>	<b>31</b>
6.1. Sistem Mimarisi.....	31
6.2. Kullanılan Veri Seti .....	33
6.3. Veri Etiketleme.....	34
6.4. Kullanılan Yöntem .....	35
6.4.1. Eğitim Önışlemleri .....	36
6.4.2. Eğitim İşlemleri.....	39
6.5. Nesne Tespit Test İşlemleri .....	41
<b>7. SONUÇLAR VE ÖNERİLER .....</b>	<b>55</b>
KAYNAKLAR.....	56
ÖZGEÇMİŞ	

## ÖZET

---

### Derin Öğrenme Yöntemlerini Kullanarak İHA Görüntülerinden Çevresel Atık Tespiti

**Serkan ÇELİK**

Yüksek Lisans Tezi

FIRAT ÜNİVERSİTESİ

Fen Bilimleri Enstitüsü

Ekobilisim Anabilim Dalı

Ekim 2022, Sayfa: xi + 58

---

Pet ve cam şişeler genellikle tek kullanımlık malzemeler olduklarından kullanıldıktan sonra uygun olmayan yerlere özensiz bir şekilde atılmaktadır. Pet ve cam şişe atıkları, çevre kirliliğinin önemli sebeplerinden biridir. Ayrıca cam şişelerin orman yangınlarının çıkmasında çok büyük etkileri vardır. Bu nedenlerden dolayı pet ve cam şişelerin tespit edilmesi ve çevreye zarar vermeden toplanması gerekmektedir. İnsansız Hava Araçları (İHA) günümüzde birçok sektörde oldukça yaygın olarak kullanılmaktadır. İHA'ların kullanım alanlarının artması ve geçmişe göre erişiminin kolay olması birçok alanda gelişmenin de önünü açmıştır. Daha az insan gücü kullanılarak daha düşük maliyetlerle birçok sorun çözüme kavuşturulmuştur.

Bu tez çalışmasında, İHA'lardan alınan görüntülerdeki pet ve cam şişelerin tespit edilmesi için görüntü işleme teknikleri kullanılmıştır. Yapılan çalışmada öncelikle eğitim için tespit edilecek nesnelere bulunduğu görüntüler belirlenmiştir. Bu görüntülerde tespit edilmesi istenen nesnelere etiketlenmiş, gerekli düzenlemeler yapılarak eğitim ve test verileri elde edilmiştir. Bu veriler kullanılarak YOLOv3 derin öğrenme algoritması ile eğitim gerçekleştirilmiş, elde edilen veriler ile nesne tespit testi yapılmıştır. Bu test işlemi için pet ve cam şişe atıklarının bulunduğu görüntüler kullanılmıştır. Yapılan tez çalışması sonucunda, doğal ortamlardaki pet ve cam şişelerin büyük bir doğruluk oranı ile tespiti gerçekleştirilmiştir. Ekosistemin korunması amaçlı özellikle pet ve cam şişelerin tespitinin yapıldığı benzer bir çalışmanın olmaması sebebiyle hazırlanmış olduğum tez çalışmasının bundan sonraki çalışmalara fayda sağlayacağı düşüncesindeyim.

**Anahtar Kelimeler:** İHA, Ekosistem, Nesne Tespit, Derin Öğrenme, Sınıflandırma, YOLO.

## ABSTRACT

---

### Detection of Environmental Waste Out of UAV Images by Using Deep Learning Methods

Serkan ÇELİK

Master's Thesis

FIRAT UNIVERSITY  
Graduate School of Natural and Applied Sciences  
Department of Eco-informatics

October 2022, Pages: xi + 58

---

Since plastic and glass bottles are generally disposable materials, they are thrown away carelessly after use. One of the main reasons of environmental pollution is plastic and glass bottle waste. Furthermore, glass bottles have a huge impact on forest fires. For these reasons, it is necessary to detect plastic and glass bottles and collect them without harming the environment. Unmanned aerial vehicles (UAV) are widely used in various sectors today. The increase in the use of unmanned aerial vehicles and the ease of access -when compared to the past- have paved the way for development in many fields. Many problems have been solved at lower costs by using less manpower.

In this thesis, image processing techniques were used to detect plastic and glass bottles in images taken from UAVs. Principally, the images with the objects to be detected for training were determined. Objects in these images were labeled, necessary arrangements were made, then training and test data were obtained. Using these data, training was carried out with the YOLOv3 deep learning algorithm, and object detection test was applied with the obtained data. Images of plastic and glass bottle wastes were used for this test process. As a result, plastic and glass bottles in the nature were detected with a great accuracy rate. Since there is no similar study for the protection of the ecosystem, especially for the detection of plastic and glass bottles, I think that the thesis I have prepared will benefit future studies.

**Keywords:** UAV, Ecosystem, Object Detection, Deep Learning, Classification, YOLO.

# ŞEKİLLER LİSTESİ

	Sayfa
Şekil 3.1. Çevreye atılmış pet ve cam şişe atıkları .....	9
Şekil 4.1. İnsansız Hava Aracı kullanım alanları .....	11
Şekil 4.2. Farklı amaçlar için kullanılan İnsansız Hava Araçları .....	12
Şekil 4.3. DJI Mavic Air 2 İHA resimleri .....	13
Şekil 5.1. Basit derin sinir ağı yapısı .....	16
Şekil 5.2. CNN, SSD ve YOLO algoritmalarının algılama doğruluğu açısından karşılaştırılması .....	17
Şekil 5.3. CNN, SSD ve YOLO algoritmalarının algılama hızlarının kıyaslanması .....	18
Şekil 5.4. Faster R-CNN mimarisi.....	19
Şekil 5.5. YOLO algoritmasının genel çalışma prensibi .....	21
Şekil 5.6. Sınırlayıcı kutu tahmin hesaplaması.....	22
Şekil 5.7. Kutu güven puanının (IoU) hesaplanması .....	23
Şekil 5.8. YOLO mimarisi.....	24
Şekil 5.9. DarkNet-53 mimarisi.....	26
Şekil 5.10. Görüntüyü üç farklı ölçekte ızgaralar ile bölme .....	27
Şekil 5.11. DarkNet-53 kullanılmış 3 ölçekli tahmin yapabilen YOLOv3 evrimsel sinir ağı yapısı .....	28
Şekil 6.1. Sistem mimarisi.....	32
Şekil 6.2. Veri setinde kullanılan jpg uzantılı dosyalar .....	33
Şekil 6.3. Veri setinde kullanılan txt uzantılı dosyalar .....	34
Şekil 6.4. Etiketlenmiş örnek veri dosyası içeriği .....	34
Şekil 6.5. Veri setinde kullanılan örnek görüntüler ve etiketlenmiş veri dosyası içerikleri .....	35
Şekil 6.6. Colab platformuna DarkNet kurulumu.....	36
Şekil 6.7. Makefile Konfigrasyonu .....	36
Şekil 6.8. NVIDIA-CUDA kurulumu.....	37
Şekil 6.9. YOLOv3 standart ağırlıkların yüklenmesi .....	37
Şekil 6.10. Google drive bağlantısı .....	37
Şekil 6.11. Veri seti dosyalarının yüklenmesi .....	38
Şekil 6.12. yolov3_custom.cfg, obj.names ve obj.data dosyalarının yüklenmesi.....	38
Şekil 6.13. yolov3_custom.cfg, obj.names ve obj.data dosyalarının içeriği.....	39
Şekil 6.14. Ağırlık değerlerinin hesaplanması (1. iterasyon) .....	40
Şekil 6.15. Ağırlık değerlerinin hesaplanması (3000. iterasyon).....	40
Şekil 6.16. Eğitim süreci hata (loss) / iterasyon grafiği.....	41

<b>Şekil 6.17.</b> İnternette elde edilen görüntü dosyaları için nesne tespit test sonuçları.....	42
<b>Şekil 6.18.</b> Görüntü çekimlerinin yapıldığı coğrafi bölge .....	43
<b>Şekil 6.19.</b> Nesne tespit test görüntülerinin çekildiği ortamlar.....	43
<b>Şekil 6.20.</b> Nesne tespit test işlemi .....	44
<b>Şekil 6.21.</b> Tek nesnenin bulunduğu farklı ortamlarda pet şişe tespiti .....	45
<b>Şekil 6.22.</b> Tek nesnenin bulunduğu farklı ortamlarda cam şişe tespiti .....	46
<b>Şekil 6.23.</b> İki nesnenin bulunduğu farklı ortamlarda pet şişe tespiti .....	47
<b>Şekil 6.24.</b> İki nesnenin bulunduğu farklı ortamlarda cam şişe tespiti .....	48
<b>Şekil 6.25.</b> Üç nesnenin bulunduğu farklı ortamlarda pet şişe tespiti.....	49
<b>Şekil 6.26.</b> Üç nesnenin bulunduğu farklı ortamlarda cam şişe tespiti .....	50
<b>Şekil 6.27.</b> Dört nesnenin bulunduğu farklı ortamlarda şişe tespiti.....	51
<b>Şekil 6.28.</b> Beş ve daha fazla nesnenin bulunduğu farklı ortamlarda şişe tespiti .....	52
<b>Şekil 6.29.</b> İki farklı nesnenin bulunduğu farklı ortamlarda şişe tespiti .....	53

## TABLÖLAR LİSTESİ

	Sayfa
<b>Tablo 3.1.</b> 2020 yılı üretilen, piyasaya sürülen ambalaj ve ambalaj atık miktarları.....	10
<b>Tablo 4.1.</b> DJI Mavic Air 2 İHA özellikleri.....	14
<b>Tablo 6.1.</b> Nesne tespit test sonuçları.....	54



## KISALTMALAR

CNN	: Evrişimsel Sinir Ağları (Convolutional Neural Network)
Colab	: Google Colaboratory
Faster R-CNN	: Hızlı Bölge Bazlı Evrişimsel Sinir Ağı (Faster Region Based Convolutional Neural Network)
GPS	: Küresel Konumlama Sistemi (Global Positioning System)
GPU	: Grafik İşlemci Birimi (Graphics Processing Unit)
IoU	: Birleşim Üzerinden Kesişim (Intersection over Union)
İHA	: İnsansız Hava Aracı
PASCAL VOC	: Örüntü Analizi, İstatistiksel Modelleme ve Hesaplamalı Öğrenme Görsel Nesne Sınıfları (Pattern Analysis, Statistical Modelling and Computational Learning Visual Object Classes)
R-CNN	: Hızlı Bölge Bazlı Evrişimsel Sinir Ağları (Faster Region Based Convolutional Neural Network)
SSD	: Tek Atış Dedektörü (Single Shot Detector)
YOLO	: Sadece Bir Kez Bakarsın (You Only Look Once)
UAV	: Unmanned Aerial Vehicle

# 1. GİRİŞ

Ekosistemde canlıların yaşamlarını sağlıklı olarak sürdürebilmeleri için öncelikle ekosistemin sorunlarından haberdar olup bu sorunların ortadan kaldırılması için gerekli çalışmaları yapmaları gerekmektedir. Çevre ve iklim sorunlarının sürekli arttığı günümüzde en büyük çevre sorunlarının başında ambalaj atıklarının çevrede oluşturdukları tahribat gelmektedir. Toplum olarak sürekli tüketmekte ve bu süreçte ambalaj atık miktarları sürekli artmaktadır. Tüketilen ürünlerin çoğu pet ve cam ambalajlar içerisinde bulunmaktadır. Günümüz toplumu bu pet ve cam ambalaj atıklarını çoğu zaman atık toplama kutuları yerine doğaya gelişigüzel atmaktadır. Pet ambalajlar günümüzde ve gelecekte üretildikleri malzemelerin yapısından dolayı çevre kirliliği ve canlı yaşamları için çok büyük sorunlar teşkil etmektedir. Cam ambalajlar ise buldukları yerlerde çok büyük oranda yangın riski oluşturduklarından ve yine uzun süre çözünmediklerinden çevre için sorun teşkil etmektedir [1].

İnsansız Hava Araçları (İHA), günümüz teknolojileri kullanılarak sürekli geliştirilen ve birçok alanda kullanılan hava araçlarıdır. Neredeyse tüm sektörlerde İHA'lar kullanılmaktadır. Bunun en önemli nedeni ise yapılan işleri daha hızlı ve daha güvenilir olarak yerine getirebilmeleridir. İlk üretildikleri zaman daha çok askeri alanda kullanılan İHA'lar teknolojinin gelişimi ve maliyetlerin düşmesiyle çok geniş bir kullanım alanı bulmuşlardır.

Tez çalışmamızda nesne tespiti için derin öğrenme algoritmaları ile görüntü işleme yöntemi kullanılmıştır. Derin öğrenme, bir makine öğrenmesi tekniğidir. İnsanların beyninden ilham alınarak geliştirilmiştir. İnsan beyninin yapabildiği bazı işlemlerin makineler tarafından yapılabilmesi için yapılan birçok çalışma yapılmaktadır. Bunlardan en önemlisi ise nesne tanıma işlemidir. Bu işlem ile belirlenen nesnelerin tespiti yapıp hızlı bir şekilde sınıflandırılması sağlanmaktadır. Gelecekte yapılacak çalışmalarla insanların yaptıkları birçok iş bu sayede makineler tarafından yapılacaktır [2].

Gerçekleştirdiğimiz tez çalışmasında çevre için çok büyük sorun oluşturan pet ve cam şişelerin İHA tarafından çekilen görüntüler kullanılarak tespit edilmesi amaçlanmıştır. Bu işlem için bir derin öğrenme algoritması olan YOLOv3 algoritması kullanılmıştır. İnternet ortamından elde edilen birçok resim düzelenmiş ve derin öğrenmede eğitim için kullanılmıştır. Uzun bir eğitim süreci sonunda makine tarafından nesnelerin algılanıp algılanmadığı test edilmiştir.

Yapılan test ile tezin amacına uygun olarak doğada çekilen görüntülerden pet ve cam şişelerin tespitinin yapılabildiği görülmüştür. Bu sonuçlar bize nesne tanıma işlemi makinelerin insan kadar hızlı ve doğru olmasa da gerçekleştirebildiğini göstermiştir. Bu tespit işlemi ile ekosistem içerisindeki sorunların çözümünde derin öğrenme algoritmalarının başarılı bir şekilde kullanılabileceği de görülmüştür.

## 2. LİTERATÜR ÖZETİ

Günümüzde İHA birçok alanda etkin olarak kullanılmaktadır. İHA'lar ilk zamanlarda sadece askeri alanlarda kullanılırken günümüzde hem sivil hem de askeri alanlarda sıklıkla kullanılmaktadırlar. İHA ile insanların ulaşmakta zorluk çektiği bölgelere ulaşip bu bölgelerde amaca uygun görüntüler ve resimler alınıp görüntü işleme algoritmaları ile bu veriler değerlendirilmekte ve sorunlara çözümler üretilmektedir. Özellikle çevrenin korunması ile ilgili İHA kullanılarak yapılan birçok çalışma mevcuttur. Bu çalışmalar incelendiğinde genellikle tarım, hayvancılık ve çevrenin korunması ile ilgili görüntü işleme çalışmalarının yoğunlukta olduğu görülmektedir.

Kabadayı ve Uysal tarafından yapılan çalışmada, İnsansız Hava Araçları ile elde edilen verilerden binaların tespiti yapılmıştır. İHA ile havadan görüntüleme yapılarak bindirmeli bir şekilde 15 küçük, 13 orta ve 4 büyük olmak üzere 32 adet binanın bulunduğu bir alanda toplanan görüntü verileri fotogrametrik yöntem ile değerlendirilmiş ve çalışma alanına ait ortofoto ve sayısal yüzey modeli oluşturulmuştur. Elde edilen veriler ile nesne tabanlı sınıflandırma yöntemi kullanılarak bina detayları tespit edilmiştir. Yapılan bu çalışma sonucunda 32 adet binadan 30 tanesi tespit edilerek %90 başarı oranı yakalanmıştır [3].

Yanık ve diğerleri tarafından yapılan çalışmada, orman yangınlarında görsel algılamaya yeni ve kesin bir bakış açısı getirilmesi amaçlanmıştır. Bu amaçla bir İHA sistemi geliştirmiş, sistemde bulunan mikrodenetleyici, derin öğrenme yöntemleri ile eğitim verilerek programlanmış ve İnsansız Hava Aracına yangın algılamasının en erken işareti olan dumanı tanıma özelliği kazandırılmıştır. Bu sayede yangın algılama sistemlerinin ortak sorunu olan yanlış alarm ve gözden kaçma oranlarının azaltılabileceği ve hızlı bir şekilde yangına müdahale edilebileceği öngörülmüştür. İnsansız Hava Aracının mobil görüş kabiliyeti sayesinde verilerin her açıdan net ve sürekli olarak kontrol edilmesi sonucu duman tespitinde %98 oranında başarı sağlanmıştır [4].

Mutlu tarafından yapılan çalışmada yaban hayatı geliştirme bölgesi içerisinde bulunan çitle çevrilmiş bir alanda kaç tane Anadolu Yaban Koyununun yaşadığını ve popülasyon yoğunluğunu İnsansız Hava Aracı teknolojisi kullanarak tespit etmiştir. Yapılan çalışma ile belirlenen alan içerisinde çekilen 535 resim kullanılarak 585 adet Anadolu Yaban Koyununun tespiti yapılmıştır. Bu tip envanter çalışmalarında havadan alınan veriler ve vücut ısıları ile hayvanların yerinin ve sayısının tespit edilebileceği gösterilmiştir [5].

Yakar ve Mırdan tarafından yapılan çalışmada, tarihi eserlerin üç boyutlu modellenmesi için bir yöntem geliştirmiştir. Bu yöntemde yerden fotoğraf makinesi ve havadan İnsansız Hava Aracı ile çekilen resimler birleştirilerek üç boyutlu modellemeye doğruluk, hız ve maliyet anlamında yeni bir soluk getirildiği görülmüştür. Belirlenen noktalardan çekilen resimler ile iki tarihi mekânın katı

modeli, 3B modeli ve ortofotoları elde edilmiştir. Bu bilgiler ile tarihi kültürel mirasın korunması ve gelecek nesillere aktarılması sağlanmıştır [6].

Demiröz tarafından yapılan çalışmada, toprak yüzey pürüzlülüğünün tespiti için İnsansız Hava Aracı ve sayısal yükselti modeli kullanılarak bir yöntem sunulmuştur. Bu yöntem ile toprak yüzey pürüzlülüğü doğrusal ve hacimsel olarak ölçülüp küçük sapmalarla toprağın erozyona elverişli olup olmadığı belirlenmiştir. Yüzey pürüzlülüğünün ölçülmesinde, altı pervaneli bir İnsansız Hava Aracı kullanılarak elde edilen sayısal yükselti modeli kullanılmıştır. Görüntülerin elde edilmesi için ise çok bantlı RGB kamera kullanılmıştır. Yapılan çalışma ile uzaktan algılamanın kullanımı sayesinde, coğrafi bilgi sistemi ortamında elde edilen verilerle, toprak yüzey pürüzlülük değerleri hacimsel, doğrusal ve alansal olarak ölçülebilmekte ve elde edilen veriler doğrultusunda ölçülen alanın haritalanması da sağlanabilmektedir [7].

Körez ve Barışçı tarafından yapılan çalışmada, Faster R-CNN algoritması kullanılarak yayaların tespiti için otomatik bir nesne tanıma yöntemi ortaya koyulmuştur. Bu yöntem ile 6 sınıf kullanılarak (insan, kayak kullanan insan, bisiklet süren insan, araba, otobüs ve golf arabası) İnsansız Hava Aracı ile 8 farklı sahneden çekilen 60 görüntü kullanılmıştır. 4 gün süren eğitim ve yaklaşık 7500 iterasyon sonucunda yayalar %58,02 oranında tespit edilmiştir. Yayaların gölgeleri, tepeden bakış açısı ve veri setindeki resimlerin bulanık veya net olması gibi durumlar başarı oranının düşük olmasındaki etken olarak görülmüştür. Bu olumsuz durumlara rağmen R-CNN metodu kullanılarak diğer yöntemlerden %2 daha başarılı bir yöntem geliştirmiştir [8].

Kim ve diğerleri tarafından yapılan çalışmada, binalardaki yüzey çatlaklarının izlenmesi için yeni bir yöntem önerilmiştir. İnsansız Hava Aracı ile çekilen 40000 çatlak olan ve olmayan yüzey görüntüsü, Evrişimsel Sinir Ağı (CNN) tabanlı bir mimari kullanılarak analiz edilmiş ve binalardaki çatlakların tespiti ve sınıflandırılması sağlanmıştır. Bu yöntem sonucunda bina çatlaklarının %99,8 oranında başarı ile tespit edilebildiği görülmüştür [9].

Yücel tarafından yapılan çalışmada, terk edilmiş maden göllerinin 3 boyutlu modellenmesi üzerinde bir yöntem geliştirilmiştir. Maden gölleri alüminyum, demir, nikel, çinko ve manganez gibi metalleri içermesi nedeniyle çok ciddi kirlilik içeren ve kullanımı mümkün olmayan su birikintisidir. Bu göllerin boşaltılması veya su miktarının azalması çevre açısından önemli riskler oluşturmaktadır. Bu yöntemde İHA kullanılarak belirlenen yüksekliklerde görüntü çekimleri yapılmıştır. Belirlenen yer kontrol noktaları vasıtasıyla ölçümler yapılmış maden göllerindeki yükseklikler belirlenmiştir. Bu yüksekliklerdeki değişimler görüntü işleme yazılımıyla işlenmiş, maden göllerinin ortofoto ve sayısal yükseklik modeli elde edilmiştir. Sonuçta maden göllerindeki yükseklik farkları tespit edilmiş bu yükseklik farkları ile göllerdeki azalma ve bu azalmanın çevre için oluşturduğu etkilerin belirlenmesi sağlanmıştır [10].

Doğan tarafından bitkilerde görülen hastalıkların derin öğrenme yöntemleri ile tespiti ve sınıflandırılması üzerine bir çalışma yapılmıştır. Yapılan çalışmada domates ve elma

yapraklarındaki hastalıkların tespiti ve sınıflandırılması derin öğrenme yöntemleri ile gerçekleştirilmiştir. Konvolüsyonel Sinir Ağı (KSA) modeli ve Learning Vector Quantization (LVQ) derin öğrenme algoritmaları kullanılarak deneysel çalışmalar domates ve elma yaprakları için ayrı ayrı yapılmıştır. Domates yapraklarından oluşan veri setinde 4 farklı hastalıklı yaprak ve 1 sağlıklı yaprak toplam 5 farklı sınıf için deneysel çalışmalar gerçekleştirilmiştir. Elma yaprakları üzerinde gerçekleştirilen çalışmalarda ise Yalova ilinde bulunan elma bahçelerindeki yaprakların İHA kamerası ile görüntüleri alınmış ve deneysel çalışmalar ağaç üzerindeki yaprak görüntüleri üzerinde gerçekleştirilmiştir. Çalışma sonucu konvolüsyonel sinir ağları tabanlı yöntemin yapraklardaki hastalıkların tespitinde başarılı olduğunu ve tarımsal alanlarda hastalık tespiti ve sınıflandırma çalışmaları için etkili bir şekilde kullanılabileceğini göstermiştir [11].

Solak ve Altınışık tarafından yapılan çalışmada, görüntü işleme teknikleri kullanılarak fındık meyvesinin tespiti ve sınıflandırılması için bir yöntem geliştirmiştir. Bu yöntemde kamera ile görüntüsü çekilen fındık meyveleri tespit edilerek düzlem üzerindeki kapladıkları boyut ve alan bilgileri hesaplanmıştır. Çalışma ortamında bulunan nesnelere tespit ve sınıflandırılması amacıyla üç aşamalı bir yöntem önerilmiştir. Önerilen yöntemin ilk aşaması olan görüntü ön işleme bölümünde kameradan alınan görüntü üzerinde filtreleme, grileştirme, ikili resme çevirme ve morfolojik işlemler uygulanmıştır. Nesne tespiti ve özellik çıkarımı aşamasında ise, ortamda yer alan nesnelere bulunması ve alan, boyut ve konum gibi özellik bilgileri elde edilmiştir. Yapılan çalışmada fındıklar gerçek zamanlı olarak %100 başarı ile tespit edilmiştir. Son aşama olan sınıflandırma aşamasında ise bilgi veri tabanında bulunan veriler, ortalama tabanlı ve K-means algoritmaları kullanılarak fındıklar küçük, orta ve büyük olarak sınıflandırılmıştır [12].

Altun ve Türker tarafından yapılan çalışmada, İHA görüntülerinden kentsel alanlarda araç tespiti için bir yöntem geliştirmiştir. Bu geliştirilen yöntemde İHA ile çekilen çok yüksek çözünürlüklü renkli ve üç bantlı (RGB) görüntülerden kentsel alanlarda araçların tespiti sağlanmıştır. İlk önce çekilen görüntülerden, görüntü eşleme ve otomatik kolerasyon tekniği ile sayısal yüzey modeli ve ortofoto üretilmiştir. Ortofoto görüntüleri üzerinden 3 test alanı içerisindeki sabit araçlar geometrik şekil itibarıyla dış sınırları üzerinden kapalı birer alan olarak çizilerek referans veri seti oluşturulmuştur. Bu referans veri seti ile sabit araçlar karşılaştırılarak doğruluk analizleri yapılmıştır. Yapılan çalışma sonucunda 1. test alanı için araç tespit yüzdesi %88,99, 2.test alanı için araç tespit yüzdesi %78,53 ve 3. test alanı için araç tespit yüzdesi %92,15 olarak bulunmuştur. Bu sonuçlar incelendiğinde yüksek çözünürlüklü resimlerden sabit araçların yüksek doğrulukta tespitinin yapılabildiği görülmüştür [13].

Murat tarafından İnsansız Hava Aracı görüntülerinden derin öğrenme yöntemleri ile nesne tanıma konusunda çalışma yapılmıştır. Bu çalışmada geliştirilmiş hızlı ve başarılı evrimsel sinir ağları kullanılarak havadan İHA ile çekilen fotoğraflardan nesne algılama yapılmıştır. 6 farklı türdeki yaya, bisikletli, araba, patenli, kampüs arabası ve otobüs nesnelere çeşitli algoritmalar kullanılarak tespit edilip karşılaştırılması yapılmıştır. YOLOv3, YOLOv5, RetinaNet ve EfficientDet algoritmalarıyla yapılan çalışmada en yüksek doğruluk oranı YOLOv5 algoritmasında elde edilmiştir. YOLOv5 algoritmasının küçük nesne tanıma da daha başarılı olduğu görülmüştür. Yapılan çalışma sonucunda doğruluk oranı havadan çekilen fotoğrafların veri kaynağı olarak kullanılmasından dolayı en fazla %69 olarak bulunmuştur [14].

Şen tarafından İnsansız Hava Aracı kullanılarak obrukların tespiti üzerine bir çalışma yapılmıştır. Bir afet türü olan obrukların canlı hayatını, tarım sahalarını ve çiftçileri tehdit edecek düzeyde olması, yerleşim yerine çok yakın olması, tarımsal arazilerin her hangi bir yerinde oluşuyor olması, gelirinin önemli bir kısmını çiftçilikten sağlayan halk için büyük tehlike arz etmektedir. Çalışmada ilk olarak İHA ile alınan görüntüler üzerinden fotogrametrik veriler elde edilmiş, daha sonra elde edilen bu veriler birleştirilerek ortofoto verileri oluşturulmuştur. Bu görüntüler üzerinde yapılan çalışmalar ile de oluşan obrukların genişlik, derinlik gibi bilgileri elde edilmiştir [15].

Gerdan tarafından yapılan çalışmada görüntü işleme teknikleri kullanılarak bazı meyvelerin sınıflandırılması için bir yöntem geliştirilmiştir. Çalışmada değişik türlerdeki elma, portakal ve ayvalardan 50 şer örnek alınarak toplamda 300 çeşit meyve kullanılmıştır. Boyut ve renk bilgileri görüntü işleme algoritması kullanılarak sınıflandırılmıştır. Yapılan çalışmada sınıflandırma için K en yakın komşuluk, Karar Ağacı, Naive Bayes sınıflandırma ve Çok Katmanlı Algılayıcı Sinir Ağı algoritmalarından faydalanılmıştır. Çalışma sonucunda Ekmek ayvada %95, Eşme ayvada %86, Elma çeşitlerinde %100, Washington Navel portakal çeşidinde %96 ve Valencia Midnight portakal çeşidinde ise %82 oranında sınıflandırma başarısı sağlanmıştır [16].

Öztürk tarafından yapılan çalışmada makine öğrenmesi ve görüntü işleme tekniklerini kullanarak İnsansız Hava Aracı ile çekilen resimler kullanılarak yaprakların sınıflandırılması için bir yöntem geliştirilmiştir. Yapılan çalışmada İnsansız Hava Aracıyla otonom bir şekilde yaprak çeşitleri tanınarak bitkisel hastalıklar ve böceklenme gibi ortaya çıkabilecek sorunlar önceden tespit edilip gerekli tedbirler alınarak, kimyasal ve biyolojik iyileşme sağlanmıştır. Öncelikle çekilen resimlerden özellik çıkarımı yapılmış, çıkarılan özellikler öğretilerek SVM ile sınıflandırma yapılmıştır. Yapılan çalışma sonucunda belirli bir bölgede görüntü işleme teknikleri ve makine öğrenmesi ile %91,3 oranında başarı sağlanan yaprak sınıflandırma işlemini gerçekleştiren bir sistem tasarlanmıştır [17].

Kaymak ve diğerleri tarafından yapılan çalışmada, İnsansız Hava Aracı ve görüntü işleme teknikleri kullanılarak meyve bahçelerinde bulunan ağaçların üzerindeki meyvelerin tespiti ve

sayımı için bir yöntem geliştirilmiştir. Çalışmada İnsansız Hava Aracı kullanılarak rota üzerinde yer alan meyve ağaçları etrafından görüntüler alınmış, bu görüntüler bilgisayar ortamında analiz edilip görüntü işleme teknikleri ile elmaların tespiti ve sayımı gerçekleştirilmiştir. Yapılan çalışma sonucunda ağaçtaki elmalar renk bakımından %78,4 başarı oranı ile tespit edilmiştir [18].

Burgaz tarafından yapılan çalışmada, Derin öğrenme algoritmaları ve İnsansız Hava Aracı kullanılarak silah tespiti için bir yöntem geliştirilmiştir. İnsansız Hava Aracı ile çekilen resimlerden R-CNN ve ResNet algoritmaları ile silah tespiti yapılmıştır. İnsanoğlunun çıplak gözle bile görmekte zorlandığı bu nesnelere daha az zaman ve maliyetle tespit işlemi gerçekleştirilmiştir. Çalışmada 200 adet görüntü kullanılarak gerçekleştirilen eğitim sonucunda resimlerdeki silah %99 doğruluk oranı ile tespit edilmiştir [19].

Bozkurt tarafından İnsansız Hava Araçları ile zeytin ağacı taç genişliğinin belirlenmesi için bir çalışma yapılmıştır. Ağaç taç genişliğinin belirlenmesi için 8 m yükseklikten dik açıyla İnsansız Hava Aracından çekilen görüntüler kullanılmıştır. Çekilen görüntüler bilgisayar ortamına aktarılmış ve görüntü işleme programında amaca uygun olarak işlenmiştir. Çalışma sonucunda İHA ile belirlenen taç yarıçap genişliği 1,10 m olarak bulunmuştur. Bu sonuç şeritmetre ile ve uydu ile yapılan ölçümlere göre ortalama değerlere daha yakın bulunmuştur [20].

Ceylan ve diğerleri tarafından İnsansız Hava Aracı ile elde edilen görüntüleri kullanarak ağaç tespiti yapmak için bir çalışma yapılmıştır. Yapılan bu çalışmada, İnsansız Hava Aracı ile çekilen görüntüler bilgisayar ortamına aktarılıp, uygun programlar kullanılarak değerlendirilmiştir. Bu değerlendirme sonucunda ortofoto, sayısal yükseklik modeli ve sayısal arazi modeli oluşturulmuştur. Bu veriler kullanılarak nesne tabanlı sınıflandırma işlemi yapılmış ve ağaçlık alanların tespiti gerçekleştirilmiştir [21].

Güzel tarafından İnsansız Hava Aracından çekilen görüntüleri kullanarak yeşil alanları tespit etmek için bir çalışma yapılmıştır. Çalışmada İnsansız Hava Aracından çekilen 159 adet görüntü kullanılmıştır. Çalışma sonucunda Yıldız Teknik Üniversitesi kampüs alanındaki gölgede yer alan hem seyrek hem de sık yeşil alanlar 1217 adet görüntü ile yapılan 2,5 saatlik eğitimler sonucunda U-Net-VGG19 derin öğrenme mimarisi ile %95 oranında doğrulukla tespit edilmiştir [22].

Albayrak ve diğerleri tarafından yapılan çalışmada, İnsansız Hava Aracı ve derin öğrenme algoritmaları kullanılarak araç tespiti üzerine bir yöntem geliştirilmiştir. Bu yöntemde iki derin öğrenme modeli (R-CNN ve YOLO sinir ağıları) kullanılmıştır. İnsansız Hava Aracı ile çekilen görüntülerden her iki algoritmaya göre veri setleri oluşturulmuştur. Pazaryerinde çekilen 80 adet görüntüden araçlar her iki algoritma kullanılarak ayrı ayrı tespit edilmiştir. Yapılan çalışma sonucunda YOLO algoritması R-CNN algoritmasına göre daha hızlı sonuç vermiş, R-CNN algoritması ise YOLO algoritmasına göre daha doğru sonuç üretmiştir [23].

Ekiz tarafından yapılan çalışmada, İnsansız Hava Aracı kullanılarak karpuz tespiti üzerine bir yöntem geliştirilmiştir. Bu yöntemde İnsansız Hava Aracı ile 3 m mesafeden çekilen görüntüler

Matlab R2013a programı kullanılarak işlenmiştir. İnsansız Hava Aracı ile çekilen görüntülerden analiz programları kullanılarak 90 gün önce ekimi gerçekleşen karpuzun tekstür ve desen özellikleri belirlenmiş ve Crimson Sweet karpuz türü %84,4 doğruluk oranıyla tespit edilmiştir [24].

Eyili tarafından yapılan çalışmada, marul bahçelerinde İnsansız Hava Aracı ile çekilen görüntüler kullanılarak yabancı ot tespiti için bir çalışma yapılmıştır. Yapılan çalışmada İnsansız Hava Aracı ile 10 farklı alanda ekimi yapılan marul bahçelerinde çekilen görüntüler kullanılmıştır. Çekilen resimlerden görüntü işleme teknikleri kullanılarak yabancı otların tespiti yapılmış ve bu otlar dar ve geniş yapraklı olarak sınıflandırılmıştır. Çalışma sonucunda yabancı otların görüntü işleme teknikleri ile havadan tespit edilebileceği ve sadece tespit edilen bu noktalarda yabancı otlarla mücadele edilerek daha sağlıklı ve hızlı bir çalışma yapılabileceği sonucuna varılmıştır [25].

Arvas tarafından yapılan çalışmada, derin öğrenme algoritmaları kullanılarak İnsansız Hava Aracı görüntülerinden haşhaş, kenevir ve tütün bitkilerinin tespiti için bir çalışma yapılmıştır. Yapılan bu çalışmada İnsansız Hava Aracından çekilen 300 haşhaş, 100 kenevir ve 100 tütün bitkisi olmak üzere 500 görüntü kullanılmıştır. Eğitim için ise 28774 haşhaş, 4172 kenevir ve 2902 tütün bitkisi olmak üzere 35848 görüntü etiketlenmiştir. Oluşturulan veri seti YOLOv5 algoritma ağı kullanılarak eğitilmiş ve eğitim sonucunda İnsansız Hava Aracından çekilen görüntülerden %77,1 doğruluk oranı ile nesnelerin tespiti yapılmıştır [26].

Yapılan literatür taramasında İnsansız Hava Araçları ile çekilen görüntüler kullanılarak görüntü işleme ve derin öğrenme algoritmaları ile birçok farklı nesnenin tespiti ve sınıflandırılmasının yapıldığı görülmüştür. Fakat bu tez çalışmasında gerçekleştirilen pet ve cam şişe tespitine ilişkin bir çalışmaya rastlanmamıştır.

### 3. PET VE CAM ŞİŞELERİN ÇEVREYE ETKİLERİ

Pet ve cam şişeler üretim süreci açısından önemli bir miktarda enerji kaynağı tüketimine neden olurlar. Şişelerin kullanımı nedeniyle hem üretim aşamasında ortaya çıkan zehirli gazlar sebebiyle hem de kullan at malzemeler olduklarından çevre kirliliği oluşturması açısından çok büyük zararlar meydana gelmektedir.

Pet şişelerin hammaddesi dünyanın en değerli doğal kaynağı olan petrol ve kömür gibi fosil yakıtlardır. Plastikler fosil yakıtların petrokimya tesislerinde işlenmesi sonucunda üretildiğinden, bunların üretimi için milyonlarca varil ham petrol ve tonlarca kömürün işlenmesi gerekmektedir. Günümüzde ham petrolün %4'ünün plastik üretimi için kullanıldığı belirtilmektedir. Üretimi ve kullanımı diğer fosil yakıtlara göre daha kolay olduğu için petrolden plastik şişe üretimi dolaylı yoldan petrole olan talebi artırmaktadır. Bu durum çok daha fazla miktarda petrolün işlenmesine yol açmakta ve dolayısıyla doğaya salınan zehirli gazlar çevre kirliliğine neden olmaktadır. Fosil yakıtların plastik üretmek için işlenmesi, çevreye geri dönüşü olmayan zararlar vermektedir. Plastik imalatı sırasında çok miktarda zararlı gaz atıkları çevreye bırakılmaktadır [27].

Cam şişeler silika, kum, soda ve kalsit gibi maddelerin uygun şartlarda ergitilmesi ile oluşturulur. Cam şişeler sert, saydam, kimyasal dayanıklılığı yüksek ve kırılğan bir yapıya sahip olan maddelerdir. Cam şişe üretimi enerji ve hammadde yoğun endüstri süreçlerinden biridir. Ham maddelerin ergitilmesi için yüksek sıcaklık gerektiğinden dolayı cam şişe üretiminde yoğun miktarda termal enerji kullanılmaktadır. Cam şişe üretiminde fosil yakıtların yanması sonucunda oluşan sera gazı emisyonları da yoğundur [28].

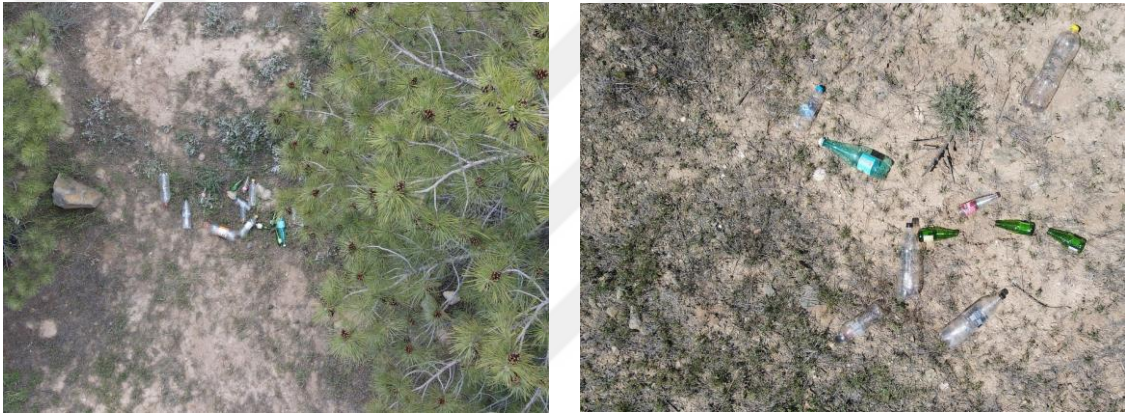
Plastik malzemeler ucuz ve dayanıklı olmaları, kolay işlenebilmeleri, hafif olmaları ve temizlik gibi avantajlarından dolayı oldukça yaygın olarak kullanılan malzemelerdir. Plastikler ambalaj ve paketleme sanayi uygulamalarında kullanım ömürlerinin kısa olmasından dolayı, üretildikten çok kısa süre sonra atık olarak karşımıza çıkmakta, doğa koşullarında çok uzun süre çözünmeden kalmaktadırlar.

Hem yaşadığımız çevreyi, dünyamızı ve enerji kaynaklarımızı korumak hem de üstün özelliklerinden sürekli faydalanabilmek için plastik malzemelerin geri dönüşümünü sağlamak gerekli ve önemlidir [29].

Cam malzemeler hem sağlıklı hem de hammaddesinin %100 doğal olması nedeniyle ambalaj sektöründe oldukça yaygın olarak kullanılmaktadır. Cam malzemelerin geri dönüşümü sonu olmayan bir süreçtir. Plastik ve kâğıdın aksine cam malzemeler sınırsız olarak yeniden değerlendirilirler. Teorik olarak cam şişeler kalite kaybı olmadan neredeyse %100 oranında tekrar kullanılıp yeniden imal edilebilirler. Cam şişeler geri dönüşüm tesislerinde kırılır ve hammadde karışımına katılarak eritme ocaklarına dökülür ve tekrar cam olarak kullanıma sunulur. Cam asfalta %30 oranında geri dönüştürülmüş cam katılmaktadır. Cam bu şekilde sonsuz bir döngü içerisinde

geri dönüştürülebilir, yapısında bozulma olmaz. Cam malzemelerin geri dönüşümü ile üretim sürecinde meydana gelen enerji tüketiminde %25, hava kirliliğinde %20, maden atıklarında %80 ve su tüketiminde ise %50 azalma sağlanabilir [30].

Çevre ve insan birbirlerini etkilemekte ve aynı zamanda birbirlerinden de etkilenmektedir. Bu etkileşimse esas olan denge ve uyumdur. Sanayi devrimine kadar bu denge ve uyum sağlanmıştır. Sanayi devriminden sonra insanoğlu ile çevre arasındaki denge bozulmuş ve insanoğlu çevre üzerinde değişikliklere sebep olmuştur. Bu değişiklikler sonucunda çevre sorunları oluşmaya başlamıştır. Günümüzde karşılaşılan en önemli çevre sorunlarından birisi katı atıklardır. Katı atıkların çevre üzerindeki olumsuz etkileri bu konuyu önemli bir çevre problemi haline getirmiştir. Şekil 3.1’de çevreye atılmış pet ve cam şişe atıkları görülmektedir [31].



**Şekil 3.1.** Çevreye atılmış pet ve cam şişe atıkları

İnsanların gelir seviyesi arttıkça kullanım alışkanlıkları da değişmiştir. İnsanoğlu daha fazla tüketmeye ve buna paralel olarak da daha fazla atığa sebep olmuştur. Özellikle kullan-at kültürünün de yaygınlaşmasıyla birlikte atık miktarında oldukça hızlı bir artış meydana gelmiştir. 2016 yılında yapılan plastik sektörü raporuna göre plastiklerin sadece %9’u geri dönüştürülmüş, %12’si yakılmış ve %79’u ise doğada plastik atık olarak kalmıştır. Doğada ayrışma ve kaybolma süreleri yaklaşık olarak 450 yıl olan plastik atıklar, uzun yok olma sürecinde besin zincirine girerek toprak ve su kirleticileri olarak ekosistemin dengesini de bozmaktadır.

Yapılan araştırmalar sonucunda cam atıkların doğada ayrışma ve kaybolma süreleri yaklaşık olarak 1 milyon yıl olarak belirlenmiştir [31]. Cam şişelerin doğada ayrışma ve kaybolma sürelerinin çok uzun olması uzun yıllar çevreye olumsuz etki etmelerine neden olmaktadır. Bu etkilerin başında ormanlık alanlarda bulunan cam şişe atıklarının, güneş ışınlarına mercek görevi görüp orman yangınlarına sebep olmaları gelmektedir. Orman yangınlarının söndürülmesi için çok fazla işgücü ve maliyet gerekmektedir. Ayrıca yanan alanların tekrar yeşillendirilmesi için uzun

süreler gerekmede ve ekonomik olarak da çok büyük kayıplar oluşmaktadır. Uzun sürelerde oluşan ormanlık alanların bir kıvılcımla yok olması aynı zamanda çevre ekosistemi açısından da olumsuz etkiler oluşturmaktadır.

**Tablo 3.1.** 2020 yılı üretilen, piyasaya sürülen ambalaj ve ambalaj atık miktarları [32]

Cinsi	Üretilen Ambalaj (ton)	Bertarafı Ambalaj Atıklarının Kontrolü Yönetmeliği Kapsamında Gerçekleştirilen Ambalajlar			
		Piyasaya Sürülen (ton)	Geri Kazanılan (ton)	Gerçekleşen Geri Kazanım Oranı (%)	Ambalaj Atık Oranı (%)
<b>Plastik</b>	2.870.185	926.551	615.988	66	24
<b>Metal</b>	525.627	129.545	72.666	56	3
<b>Kompozit</b>	273.495	95.412	66.973	70	3
<b>Kâğıt Karton</b>	3.260.126	1.198.603	1.079.857	90	34
<b>Cam</b>	1.742.759	802.067	242.226	30	22
<b>Ahşap</b>	776.551	555.152	222.714	40	14
<b>Toplam</b>	9.448.743	3.707.329	2.300.425	62	

Tablo 3.1’de 2020 yılına ait üretilen, piyasaya sürülen ambalaj ve ambalaj atık miktarları görülmektedir. Piyasaya sürülen ambalajların %46’sını plastik ve cam ambalajlar oluşturmaktadır. Plastik ambalajların %66’sı, cam ambalajların ise %30’u ancak geri dönüştürülebilmektedir. Bu oranlara dikkat edilirse geri dönüşüm oranının düşük olduğu görülmektedir. Bu geri dönüşüm oranlarının çok daha yukarılara çıkarılması için pet ve cam şişelerin tespiti büyük önem taşımaktadır.

Sonuç olarak kullanımı oldukça yaygın olan pet ve cam şişelerin doğada uzun süre kalması hem ekonomik olarak hem de çevre kirliliği ve diğer etkilerinden dolayı istenmeyen bir durumdur. Bu atıkların tespit edilerek geri dönüşüme kazandırılması, hem çevre kirliliğinin ve orman yangınlarının önlenmesi hem de ekonomik açıdan büyük önem taşımaktadır.

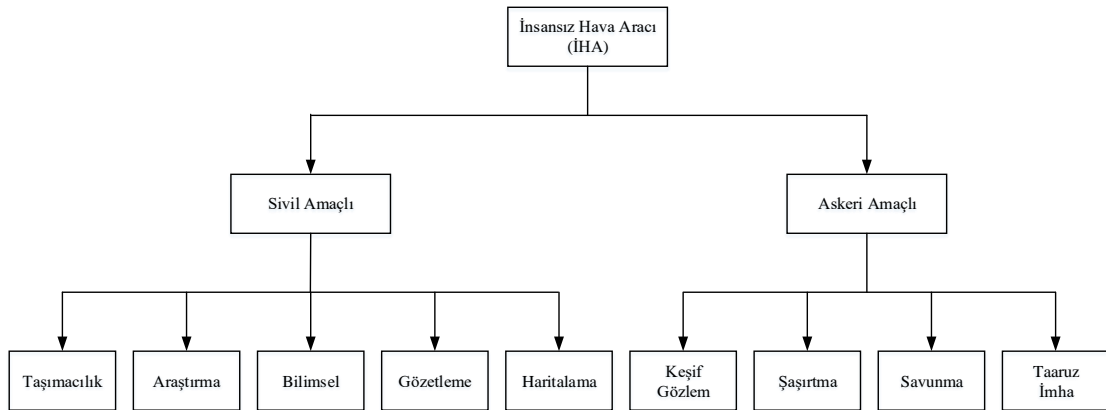
## 4. İNSANSIZ HAVA ARAÇLARI

İnsansız Hava Aracının farklı kurum ve kuruluşlar tarafından belirlenmiş birçok tanımı bulunmaktadır. İnsansız Hava Aracı (İHA) içerisinde insan bulunmayan hava araçlarına verilen genel isimdir. İHA içinde yolcusu ve pilotu olmayan, amaca uygun ekipman (kamera, GNSS, lazer tarama cihazı, silah sistemleri vb.) taşıyabilen, uzaktan kontrol edilen ve otomatik olarak görevini icra edebilen bir tür uçak olarak tanımlanmıştır. Havada insansız uçabilen ve sivil amaçlı kullanılan İnsansız Hava Araçları drone olarak da adlandırılırlar [3].

Günümüzde İHA'lar komuta merkezinde bulunan bir pilot tarafından uzaktan kumanda sistemleri ile mesafe sınırı olmaksızın kontrol edilebilen, verilen görevi yerine getiren ve otomatik olarak uçurulan hava araçlarıdır. İHA'lar genel olarak gövde, motor, güç ünitesi, pervane, uçuş kontrol ünitesi, kamera ve sensör bileşenlerinden oluşmaktadır [33].

İHA'lar ilk olarak askeri amaçla 1916 yılında 1. Dünya savaşı sırasında kullanılmıştır. Sivil amaçla ilk kullanımı ise 1950 yıllarında olmuştur. İlk uzaktan kumandalı İHA 1935 yılında geliştirilmiştir. II. Dünya savaşı sırasında askeri amaçlı olarak birçok farklı model geliştirilmiş ve kullanılmıştır [3].

Teknolojinin her geçen gün gelişmesi ile popülerliği sürekli artan İHA'lar, günümüzde neredeyse tüm sektörlerde kullanılan araçlar haline gelmişlerdir. İHA'lar ilk üretildiklerinde özellikle askeri alanlarda insan kaybı riskinin bulunduğu kritik görevlerde kullanılmışlardır. İHA'lar uzun süre havada kalmaları, uydudan kontrol edilebilmeleri, menzillerinin çok yüksek olması ve tek seferde sahip olduğu birçok silah sistemi ile birden fazla hedefi yok edebilmeleri nedeniyle tüm ordular tarafından ihtiyaç duyulan bir askeri araç haline gelmiştir. Günümüzde İHA'lar askeri amaçlar doğrultusunda taarruz, gözetleme, keşif, izleme ve haritalama görevlerini yerine getirmektedirler [34].



Şekil 4.1. İnsansız Hava Aracı kullanım alanları [35]

Şekil 4.1’de İnsansız Hava Araçlarının kullanım alanlarına ait şema verilmiştir. Askeri alanların dışında doğal afetler, trafik denetimi, kaçak yapılaşmanın kontrol edilmesi, spor faaliyetlerinin izlenmesi, orman yangınlarının tespiti ve söndürülmesi, tarım alanlarının ilaçlanması, sınır güvenliğinin sağlanması, fotoğrafçılık, kargo taşımacılığı, madencilik, haritacılık, habercilik ve gazetecilik, arama kurtarma ve ilk yardım gibi birçok alanda İHA’lar sıklıkla kullanılmaktadır [35]. Şekil 4.2’de farklı amaçlar için kullanılan İHA resimleri görülmektedir.



Şekil 4.2. Farklı amaçlar için kullanılan İnsansız Hava Araçları

İHA’lar havada kalış sürelerine, ağırlıklarına, irtifa yüksekliklerine ve kullanım alanlarına göre farklı sınıflandırmalara tabi tutulmuşlardır. En yaygın yapılan sınıflandırma havada kalış

süreleri ve uçuş irtifalarına göre yapılan sınıflandırmadır. Genel sınıflandırmada Mikro/Mini, Taktik, Operatif ve Stratejik olmak üzere 4 ana gruba ayrılırlar. Bunun yanında kanat tipine göre sabit ve döner kanatlı, uçuş irtifasına göre alçak orta ve yüksek irtifalı, motor tipine göre elektrikli, gaz türbinli ve pistonlu olarak da sınıflandırılırlar [36].

Tez çalışmamızda çevreye atılan pet ve cam şişe atıklarına ait resimlerin elde edilmesi için DJI firmasının üretmiş olduğu Mavic Air 2 model İHA kullanılmıştır.

#### 4.1. DJI Mavic Air 2 İHA

Doğal ortamda çevreye atılmış pet ve cam şişelere ait görüntüler DJI Mavic Air 2 İHA tarafından çekilmiştir. Kullanılan İHA 570 gr ağırlığa sahip, maksimum 68 km/h hız yapabilen, uzun süre havada kalabilen, uzak mesafelerden görüntü iletebilen ve çok kaliteli resimler çekebilen gelişmiş bir İHA'dır. Şekil 4.3'de kullanılan DJI Mavic Air 2 İHA'sına ait resimler görülmektedir.



Şekil 4.3. DJI Mavic Air 2 İHA resimleri

**Tablo 4.1.** DJI Mavic Air 2 İHA özellikleri [37]

Fonksiyonel Özellikler		Teknik Özellikler	
<b>360 Derece Takla</b>	Yok	<b>Ağırlık:</b>	570 gr
<b>Başsız Mod</b>	Yok	<b>Batarya:</b>	3500 mAh LiPo 3S
<b>Beni Takip Et</b>	Var	<b>Ebat:</b>	21-25 cm
<b>Engel Sensörü</b>	3 Yönlü	<b>GPS Modu:</b>	GPS Var
<b>FPV</b>	Var	<b>Kamera:</b>	4K
<b>Hareketli Gimbal</b>	3 Eksenli Gimbal	<b>Kamera Sensör</b>	1/2" CMOS
<b>Kaza Koruma</b>	Var	<b>Fotoğraf Çözünürlüğü</b>	48 Mp 8000x6000
<b>Otomatik Eve Dönüş</b>	Var	<b>Maksimum Hız:</b>	50-70 km/H
<b>Otomatik Rota Takip:</b>	Var	<b>Uçuş Mesafesi:</b>	5000-6000 m
<b>Sabit İrtifa:</b>	Var	<b>Uçuş Süresi:</b>	31-35 Dakika
<b>Tek Tuş İniş:</b>	Var	<b>Max iletişim Mesafesi</b>	10 km
<b>Tek Tuş Kalkış:</b>	Var	<b>Video Aktarım Mesafesi</b>	6 km
<b>İzinler</b>		<b>Hafıza</b>	8 GB
<b>Sivil Havacılık İzni:</b>	Gerektirir	<b>Genişletilebilir Hafıza</b>	256 GB'a kadar
<b>Kumanda Özellikleri</b>		<b>Max Bitrate</b>	120 Mbps
<b>GHZ:</b>	5.8 GHz	<b>Bağlantı Konnektörleri</b>	Lightning, Micro USB, USB Type-C
<b>Kanal:</b>	12 Kanallı	<b>Zoom</b>	Mevcut
<b>Uygulama Kontrol:</b>	Var	<b>Şarj Girişi</b>	13.2 V=2.82 A

Tablo 4.1.'de tez çalışmalarında kullanılan DJI Mavic Air 2'ye ait teknik bilgiler verilmiştir. Tablo 4.1.'de verildiği gibi DJI Mavic Air 2 İHA, zoom özellikli 1/2 inç cmos sensöre sahip, 48 mega piksele kadar JPEG formatında fotoğraf ve 4K/60 fps çözünürlüğe kadar MP4 formatında video çekebilmektedir. Ayrıca 10 km mesafeye kadar canlı görüntü iletebilmektedir.

Sahip olduğu uzaktan kumanda, Lightning, Mikro USB ve Type-C bağlantı uçları ile Android veya IOS işletim sistemine sahip cep telefonlarına bağlanabilmektedir. Cep telefonlarına yüklenen yazılım yardımı ile İHA, 18,5 km mesafeye kadar 2.4/5.8 GHz veri iletim frekansında kontrol edilebilmektedir.

İHA'lar için en önemli hususlardan biri de güç gereksinimidir. Çünkü enerji kaynağı İHA'nın ne kadar süre havada kalacağını belirleyen en önemli etkidir. Kullandığımız İHA 3500 mAh'lık LiPo 3S bataryaya sahiptir. Bu sayede yaklaşık olarak 34 dakikaya kadar havada kalabilmektedir. 13.2 V gerilim 2,82 A akım çıkışına sahip 38 wattlık bir şarj cihazı ile yaklaşık 1,5 saat sürede şarj edilebilmektedir.

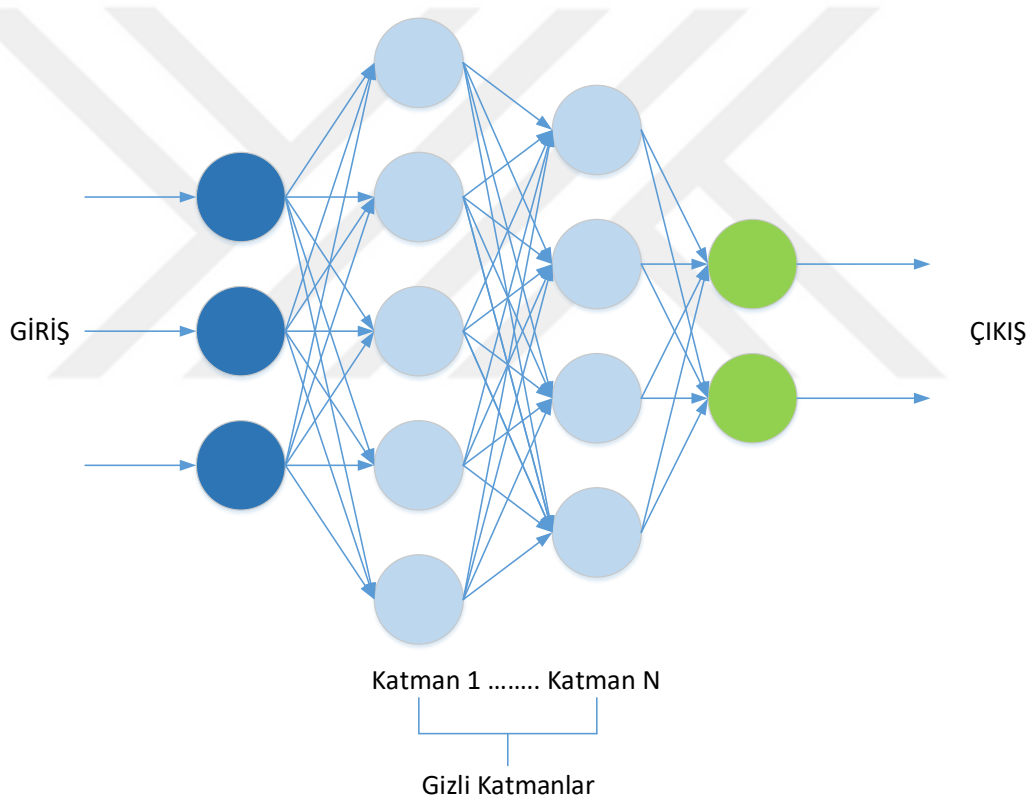
Kullanılan İHA görüntüleri kaydedebilmek için 8 GB dahili hafızaya sahiptir. İstenilirse sahip olduğu SD kart yuvası sayesinde hafıza 256 GB'a kadar artırılarak daha fazla fotoğraf ve daha uzun süreli görüntü kaydı yapılabilir.

Ayrıca GPS+GLONASS sistemlerine sahip olduğundan, konum belirleme yeteneğine ve belirlenen konumda sabit olarak durabilme özelliğine de sahiptir [37].



## 5. DERİN ÖĞRENME

Derin öğrenme, 2006 yılından itibaren popüler olarak kullanılmaya başlanmış makine öğrenmesi alanının en güncel alt dallarından biridir. Derin öğrenme yöntemlerinde insan beyninin yapısından esinlenerek oluşturulan yapay sinir ağı algoritmaları kullanılmıştır. İnsan beyni bir nöron ağıdır, derin öğrenmede ise nöronlar düğümlere karşılık gelir. Lineer cebire dayanan bu sistemde insan beynindeki nöronlara benzeyen düğüm adı verilen yapılar bulunur. Düğüm yapılarının birleştirilmesi ile katmanlar, katmanların birleştirilmesi ile de sinir ağı denilen yapılar oluşturulur. Bu ağlara derin sinir ağı da denir. Şekil 5.1'de basit bir derin sinir ağı yapısı görülmektedir [2].



Şekil 5.1. Basit derin sinir ağı yapısı [2]

Derin öğrenme, makine öğrenimi alanında yapılan çalışmalar ile yapay zekâ kapsamının genişlemesine de öncülük etmiştir. Yüzeysel öğrenme ile karşılaştırıldığında derin öğrenme kavramı daha soyut bilgilere ulaşmak için derin (gizli) katmanlar inşa etme avantajına sahiptir. Bu sayede çok katmanlı mimarilerde çok boyutlu veriler ile çalışma imkânı sağlamıştır [38].

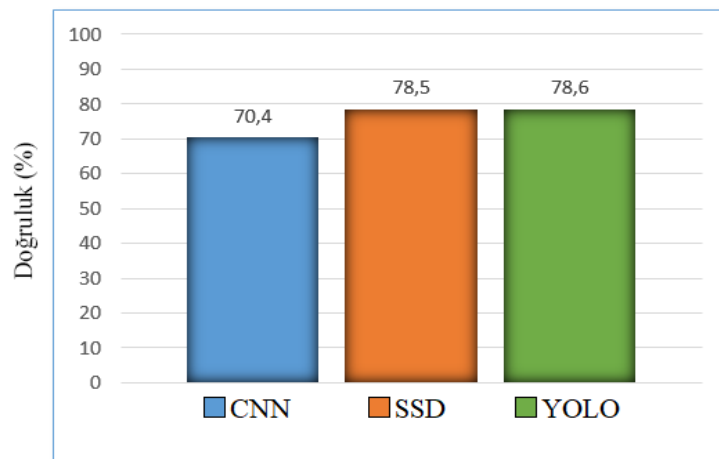
Yapay zekâ fikrinin ilk ortaya çıktığı dönemde insanlar tarafından güçlükle çözülebilen, uzun zaman alan ve matematiksel hesaplama gerektiren problemlerin çözümü için bilgisayarlar

kullanılmakta ve bu problemler çok hızlı bir şekilde çözülmekteydi. Bunun yanında insanların kolay bir şekilde ve sürekli yapabildiği yüz tanıma, nesne algılama ya da takip etme, konuşma tanıma gibi problemlerin çözümü ise bilgisayarlar için çok büyük bir sorun teşkil etmekteydi. Bilgisayarların kavramlar hiyerarşisi mantığı, karmaşık kavramların daha basit kavramlardan yola çıkarak örüntü kurma yoluyla bulunabilmesine imkân sağlamaktadır. Bu kavramlar birbiri üzerine inşa edilerek birden fazla katmandan oluşan ve derin öğrenme adı verilen bir mimari ortaya çıkmıştır. Bilgisayarlı nesne tanıma çalışmaları günümüzde derin öğrenme tabanlı mimariler ve algoritmalar kullanılarak geliştirilmektedir [14].

Derin öğrenme tabanlı yöntemler yüz tanıma, plaka tanıma, nesne algılama, pozisyon tanıma, hareket algılama, şerit tespiti, yaya algılama, duygu algılama, otomatik park etme, resim tanıma ve görüntü işleme gibi birçok alanda karmaşık problemleri çözmek için kullanılmış ve ciddi bir başarı oranı elde edilmiştir [39].

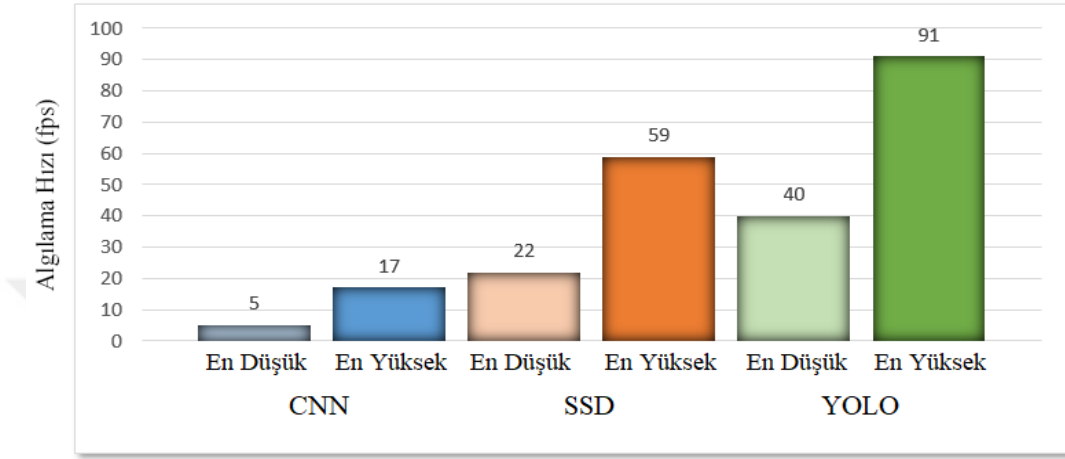
Farklı nesne algılama algoritmalarının adil bir şekilde karşılaştırılması çok zordur. Hangi algoritmanın en iyisi olduğunun kesin bir yanıtı yoktur. Bununla birlikte gerçek hayatta karşılaştığımız uygulamalarda hız ve doğruluğun dengesini sağlamak amacıyla bazı seçimler yapılabilir. Çünkü bazen hızlı işlem yapabilen algoritmalar doğru algılama konusunda sıkıntı yaşayabilmektedirler [38].

Görüntü işleme algoritmaları içerisinde en sık kullanılanlar Faster R-CNN algoritması, SSD algoritması ve YOLO algoritmasıdır. Bu algoritmalar algılama doğruluğu ve algılama hızları bakımından aynı veri seti ile (Pascal VOC 2007 ve 2012) kıyaslanmış ve sonuçlar Şekil 5.2 ve Şekil 5.3'de grafik olarak verilmiştir.



Şekil 5.2. CNN, SSD ve YOLO algoritmalarının algılama doğruluğu açısından karşılaştırılması [40]

Şekil 5.2’de CNN, SSD ve YOLO algoritmaları algılama doğruluğu açısından karşılaştırılmıştır. CNN algoritması %70,4, SSD algoritması %78,5, YOLO algoritması ise %78,6 algılama doğruluk oranına sahiptir. Her üç algıtmada birbirine yakın yüzdelik değerlere sahiptirler. YOLO algoritması az farkla diğer 2 algıtmadan daha doğru algılama yapabilmektedir.



Şekil 5.3. CNN, SSD ve YOLO algoritmalarının algılama hızlarının kıyaslanması [40]

Şekil 5.3’de ise CNN, SSD ve YOLO algoritmaları algılama hızı açısından karşılaştırılmıştır. Bu karşılaştırmada saniyedeki kare sayısı (fps) değerlerinin en düşük ve en yüksek olduğu değerler grafiğe aktarılmıştır. CNN algoritması en düşük 5 fps en yüksek 17 fps, SSD algoritması en düşük 22 fps en yüksek 59 fps, YOLO algoritması ise en düşük 40 fps en yüksek 91 fps algılama hızına sahiptir. YOLO algoritması diğer 2 algıtmadan daha hızlı algılama işlemi yapabilmektedir. Bu durum gerçek zamanlı nesne tespitinde YOLO algoritmasının kullanımının en önemli nedenidir [40].

## 5.1. Evrimsel Sinir Ağları

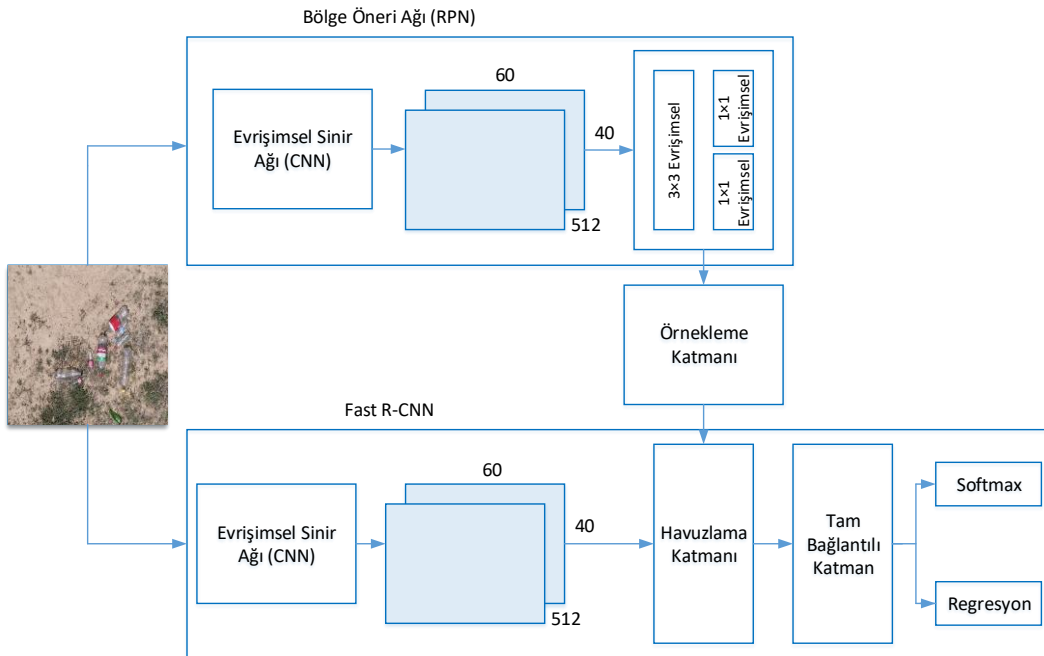
Tüm derin öğrenme algoritmalarının temelinde evrimsel sinir ağları vardır. Evrimsel sinir ağı (CNN), görüntü analizi için tasarlanmış bir derin sinir ağıdır. Bu ağ diğer sınıflandırma algoritmalarına kıyasla çok fazla ön işlem gerektirmez. Yüksek boyutlu verilerden düşük boyutlu verilerin elde edilmesi prensibiyle çalışır. Evrimsel sinir ağı, çok fazla eğitim ile filtreleri/özellikleri öğrenebilir. Bu ağ, Evrimsel Katman, Havuzlama Katmanı ve Tam Bağlantılı Katmanı olmak üzere 3 ana katmana sahiptir [38]. Evrimsel katman, girdi görüntüsünü alarak evrim işlemine tabi tutan filtreleri barındıran katmandır. Burada kullanılan filtreler farklı boyutlarda (2×2, 3×3, 5×5) olabilir. Evrimsel işlem sonucunda aktivasyon haritası elde edilir ve her bir filtreye ait özellikler çıkarılır.

Havuzlama katmanı genellikle evrişimsel katmanının hemen ardından ağıdaki parametre ve hesaplama miktarını azaltmak için kullanılır. Bu işlem için görüntünün boyutu küçültülür bu işlem aynı zamanda ezber öğrenmeyi de önler. Tam bağlantılı katman ise evrişimsel sinir ağlarının en önemli ve en son katmanıdır. Bu katman evrişimsel sinir ağını ileri yönde besler. Tam bağlantılı katman kendisinden önce gelen evrişim ve havuzlama katmanlarının çıktısını alır ve bu çıktıyı vektör olacak şekilde düzleştirir. Bu vektör daha sonra ağırlık ve yanlılık değerleriyle matematiksel olarak hesaplanır [41].

CNN algoritması üzerinde iyileştirmeler yapılarak 2013 yılında R-CNN algoritması geliştirilmiştir. Daha sonra 2015 yılında, R-CNN algoritması ile çoklu objelerde nesne tespitinde yaşanan sorunları çözmek ve algoritmanın iş yükü ve işlem zamanından kaynaklanan problemleri gidermek için Fast R-CNN algoritması kullanılmaya başlanmıştır. Yine 2015 yılında, Fast R-CNN algoritmasının bölge teklifleri üretme adımında iyileştirmeler yapılarak Faster R-CNN algoritması geliştirilmiştir [39].

## 5.2. Hızlı Bölge Bazlı Evrişimsel Sinir Ağı (Faster R-CNN)

Hızlı bölge bazlı evrişimsel sinir ağı (Faster R-CNN) modeli, R-CNN ve Fast R-CNN'in karmaşık eğitim hattını düzenleyerek ve paralel işlem yaparak işlem zamanında iyileştirme sağlamıştır. Günümüzde nesne tanıma uygulamalarında genellikle Faster R-CNN modeli kullanılmaktadır.



Şekil 5.4. Faster R-CNN mimarisi [41]

Şekil 5.4’de görüldüğü gibi Faster R-CNN, birbirine paralel Bölge Öneri Ağı (RPN) ve Fast R-CNN ağından meydana gelmektedir. Bölge Öneri Ağı bölümünde girdi görüntüsü omurga ağından geçirilerek öznitelik haritası elde edilir. Bölge Öneri Ağı öznitelik haritasının tüm noktalarında nesne bulunup bulunmadığını belirler ve nesne var ise girdi görüntüsü üzerinde bulunduğu alanı tahmin eder. Bölge önermeli öznitelik haritası, öznitelik vektörüne dönüştürülerek ağın son kısmında yer alan ve iki çıkışı bulunan tam bağlantılı katmana giriş olarak verilir. Regresyon çıkışı sınırlayıcı kutu tahminlerinin iyileştirilmesi için sınıflandırma çıkışı ise sınıflandırma tahminleri için kullanılır.

PASCAL VOC veri seti üzerinde çalıştırılıp karşılaştırılan bu üç CNN algoritmasından Faster R-CNN algoritması görüntü test işlemini Fast R-CNN algoritmasından 10 kat, R-CNN algoritmasından ise 100 kat daha hızlı gerçekleştirmiştir [41].

### **5.3. Tek Atış Nesne Algılama (SSD) Algoritması**

Liu ve arkadaşları tarafından 2016 yılında gerçek zamanlı sistemler için geliştirilen Tek Atış Nesne Algılama (SSD) algoritması arka plan bilgisini kullanarak nesneyi algılamaktadır. SSD gerçek zamanlı nesne algılama için tasarlanmış bir algoritmadır. SSD, nesne algılama işlemini iki aşamada gerçekleştirir. Bu aşamalar özellik haritalarını çıkarmak ve evrimsel filtreleri kullanarak nesneyi algılamaktır. Bu modelde düşük çözünürlükteki görüntülerin kullanılması ile daha hızlı R-CNN elde edilebilir. SSD, hem yüksek doğruluk oranı hem de hız gerektiren uygulamalarda kullanılan bir algoritmadır [39].

### **5.4. YOLO Nesne Algılama Algoritması**

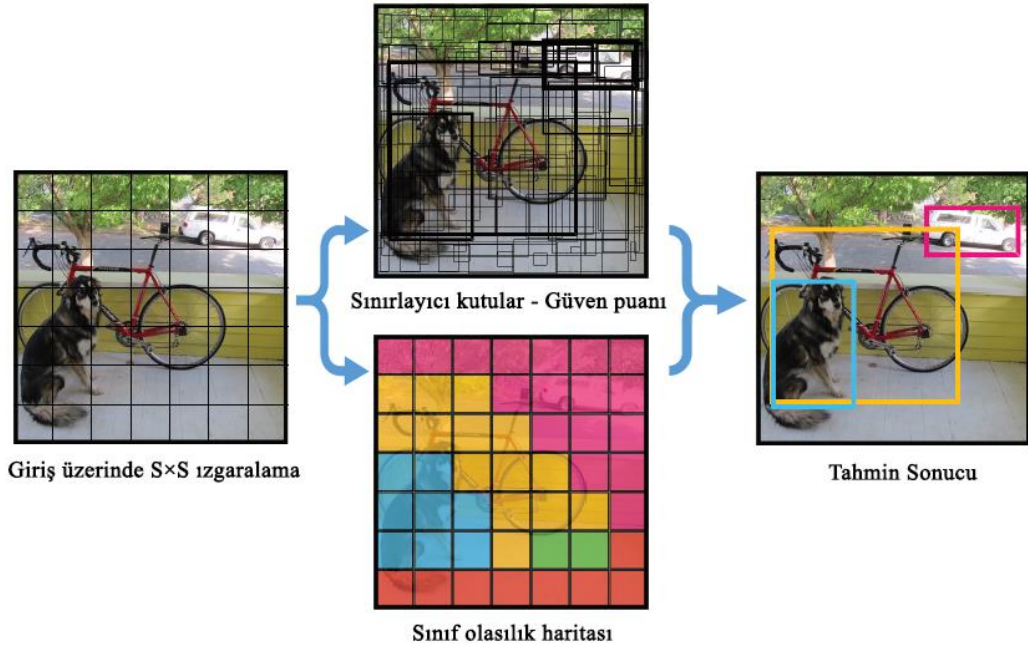
YOLO (You Look Only Once) “Sadece bir kere bak” kelimelerinin kısaltılmasından elde edilmiş, yüksek performansa sahip derin öğrenmeye dayalı gelişmiş bir nesne algılama algoritmasıdır. Yakın zamanda geliştirilen YOLO, nesne tespitinde kullanılan bir algoritma olup özellikle gerçek zamanlı görüntü işleme uygulamalarında kullanılmaktadır [42].

Nesne tespiti için kullanılacak olan algoritma hem doğru hem de hızlı olmalı ayrıca birçok nesneyi de tanıyabilmelidir. Mevcut nesne algılama veri kümeleri, sınıflandırma ve etiketlendirme işlemleri için kullanılan veri kümelerine göre yetersizdir. Günümüzde kullanılan büyük boyutlardaki sınıflandırma verilerinden yararlanmak ve nesne dedektörlerini sadece sınıflandırma için değil aynı zamanda algılama konusunda da eğitmek için ortak bir eğitim algoritmasına ihtiyaç duyulmuş ve böylece YOLO algoritması ortaya çıkmıştır [14].

YOLOv1 algoritması ilk olarak 2015 yılında Joseph Redmon tarafından tasarlanmıştır. Daha sonra 2016 yılında Joseph Redmon ve Ali Farhadi YOLOv1 versiyonunda iyileştirmeler yaparak 9000 nesne tanıyabilen YOLOv2 versiyonunu, 2018 yılında da en son yöntemleri deneyerek diğer

versiyonlardan daha kapsamlı, daha doğru sonuçlar veren ve daha hızlı bir versiyon olan YOLOv3'ü geliştirmişlerdir [43].

Bir resim üzerinde yapılan sınıflandırma işlemi sonucunda insan ya da giysiyi bulmak aşılması gereken bir problemdir. Ancak insan ve giysiyi aynı resimde yerleriyle beraber algılamak daha zor bir problemdir. Birinci problem sınıflandırma yöntemiyle yapay sinir ağlarıyla çözülebilirken, ikinci problemi çözmek daha farklıdır ve farklı bir yaklaşım gerektirmektedir. YOLO güçlü bir evrimsel sinir ağıdır ve bize resmin nerede olduğunu sınırlayıcı bir kutu çizerek göstermektedir. YOLO'nun günümüzde popüler olmasının nedeni ise resme bir defa bakarak algılamayı gerçekleştirmesidir. Bu da YOLO'nun diğer nesne algılama algoritmalarından daha hızlı çalışmasını sağlamaktadır [40].



Şekil 5.5. YOLO algoritmasının genel çalışma prensibi [44]

Şekil 5.5'de YOLO algoritmasının temel çalışma yapısı gösterilmektedir. Şekilde görüldüğü gibi YOLO giriş resmini öncelikle S×S boyutunda ızgaralara böler ve her bir ızgara hücresi o ızgara hücresinde ortalanmış nesneyi tahmin eder. Tahmin sonucunda tahmin değeri üretilir. Tahmin sonuçları toplanarak en düşük olasılığa sahip sınırlayıcı kutu silinir. En yüksek tahmin olasılık değerlerine sahip sınırlayıcı kutu ise nihai sonuç olarak belirlenir [44]. Izgara hücreleri aynı zamanda B tane sınırlayıcı kutuyu ve C sınıf olasılıklarını da tahmin eder. Bu toplamda S×S×C tane sınıf olasılığı demektir. Sınıf tahminleri çıktı vektörüne eklenerek çıktı boyutu  $Y = S \times S \times (B \times 5 + C)$  ifadesi ile elde edilir [45].

Kullanılan sembollerin anlamları;

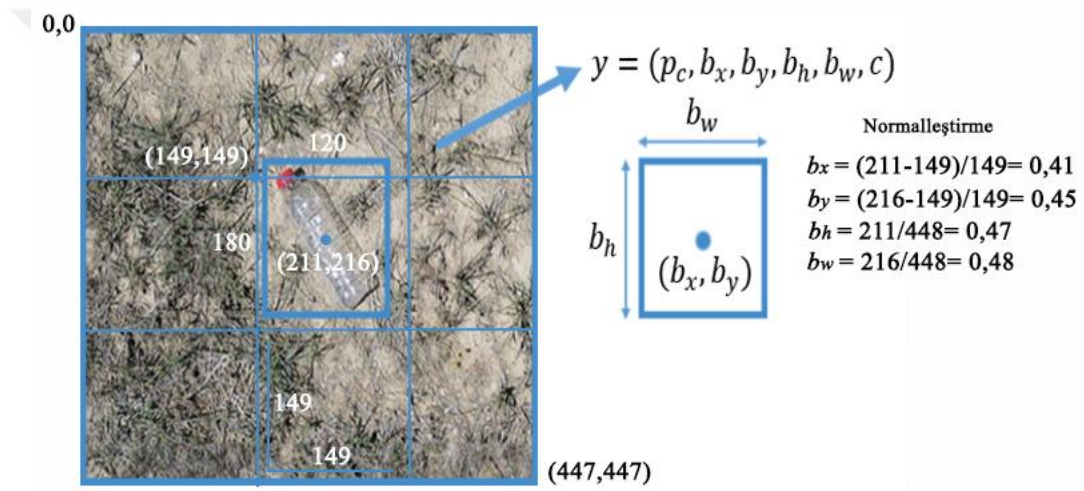
$Y$  : Çıktı boyutu

$S$  : Izgara satır ve sütun sayısı

$B$  : Her ızgaradan istenen tahmin sayısı

$C$  : Tahmin edilecek sınıf sayısı

Sınırlayıcı kutu tahmini yapılırken 6 bileşen kullanılır. Bunlardan 4 tanesi nesne sınırlayıcı kutunun yüksekliği, genişliği ve merkez noktasının koordinatlarından oluşur. Diğer iki bileşen ise Sınıf ve Güven puanından oluşmaktadır. Şekil 5.6'da sınırlayıcı kutu tahmin hesaplaması görülmektedir.



Şekil 5.6. Sınırlayıcı kutu tahmin hesaplaması

Hesaplama kullanılan bileşenlerin açıklamaları;

$b_x$  : Sınırlayıcı kutu merkez x koordinatı

$b_y$  : Sınırlayıcı kutu merkez y koordinatı

$b_h$  : Sınırlayıcı kutu yüksekliği

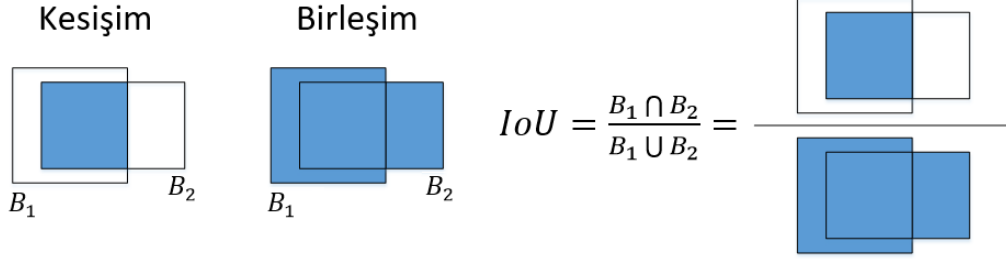
$b_w$  : Sınırlayıcı kutu genişliği

$p_c$  : Güven puanı

$c$  : Sınıf

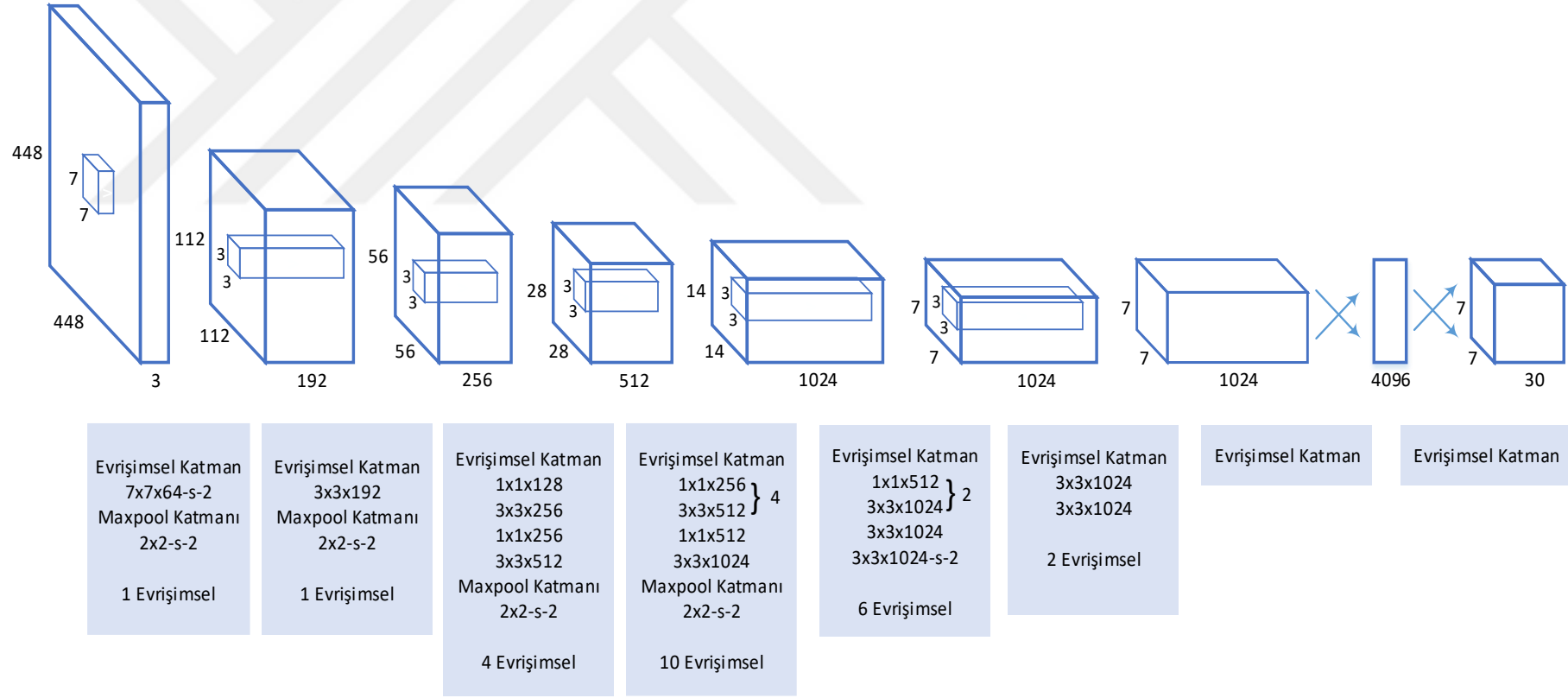
Sınırlayıcı kutu tahmin hesaplamasında 5 parametre tahmin edilir. Bunlar  $b_x$ ,  $b_y$ ,  $b_h$ ,  $b_w$  ve  $p_c$  parametreleridir. Sınırlayıcı kutunun genişliği, yüksekliği, merkez noktasının x ve y koordinatı, görüntünün genişliği ve yüksekliği ile normalleştirilir. Bu normalleştirme sonucunda  $b_x$ ,  $b_y$ ,  $b_h$ ,  $b_w$  parametreleri 0 ile 1 arasında değerler alır [45].

Kutu güven puanı (IoU), bir sınırlayıcı kutuda nesne bulunma olasılığını ve sınırlayıcı kutunun doğruluğunu temsil eden bir puandır [46]. Şekil 5.7’de en solda kesişim alanı, ortada birleşim alanı ve en sağda ise kutu görev puanının nasıl hesaplandığı görülmektedir.



Şekil 5.7. Kutu güven puanının (IoU) hesaplanması [45]

Şekil 5.7’de kutu güven puanının gerçek nesne sınırlayıcı kutusu  $B_1$  ile tahmin edilen nesne sınırlayıcı kutusu  $B_2$ ’nin kesişim alanı ile birleşim alanının oranlanması sonucu elde edilen değer olduğu görülmektedir. Hücre içerisinde hiçbir nesne yoksa kutu güven puanı sıfırdır. Kutu güven puanının nesne tespitinde 1 veya 1’e yakın olması istenir. Eğer kutu güven puanı 1 ise gerçek nesne sınırlayıcı kutusu ile tahmin edilen nesne sınırlayıcı kutusunun kesişim alanı ile birleşim alanı birbirine eşittir. Kutu güven puanının yüksek veya düşük olması, herhangi bir sınıftaki nesnenin varlığını veya yokluğunu belirler. YOLOv3 algoritmasında IoU (Intersection over Union) değeri %50’nin üzerinde tutulmaktadır [45].



Şekil 5.8. YOLO mimarisi [14]

Şekil 5.8’de YOLO mimarisi gösterilmiştir. YOLO ağ mimarisi GoogLeNet’ten ilham alınarak tasarlanmıştır. YOLO mimarisinde görüldüğü gibi algılama ağıımız 24 evrişimsel katman ve bunların devamında 2 tam bağlantılı katmandan oluşmuştur [47]. Ağın çıkışında ise  $7 \times 7 \times 30$  tensör (çok boyutlu veri) tahmin elde edilmiştir.

#### 5.4.1. YOLOv3

YOLOv2 algoritmasında 19 katman kullanılması, küçük nesnelerin tanınmasında meydana gelen sorunlar ayrıca gerçek zamanlı uygulamalardaki hız problemleri YOLOv3 algoritmasının geliştirilmesini gerekli kılmıştır. Geliştirilen YOLOv3 algoritmasında 53 katmanlı DarkNet kütüphanesi kullanılarak daha küçük nesnelerin tespit performansı ve doğruluğu artırılmıştır [43].

YOLOv3 algoritması çoklu kutu yapısı kullanarak birden çok nesneyi gerçek zamanlı olarak yüksek frame hızlarında tanıyabilmektedir [41].

YOLOv3 algoritması nesne algılamayı, doğrudan görüntü piksellerinden sınırlandırıcı kutu koordinatlarına ve olası sınıflara dönüştürmek suretiyle tek bir regresyon problemi olarak ele alır. Bu model kullanıldığında hangi nesnelerin mevcut olduğunu ve nerede olduklarını tahmin etmek için bir resme yalnızca bir kez bakmak yeterlidir. Başka bir deyişle YOLOv3 algoritması bir girdi görüntüyü genel olarak  $S \times S$  ızgaralara bölüp bu ızgaralardan nesnelere ait olan sınırlayıcı kutuları belirleyerek sınıflandırmak üzere yapay sinir ağından yalnız bir kere geçirek fotoğraftaki tüm nesnelerin sınıfını ve koordinatlarını tahmin eder. Böylelikle nesne tespitini tek bir regresyon problemi olarak ele almış olur. YOLO, tam görüntüler üzerinden eğitim alır ve algılama performansını doğrudan optimize eder [14].

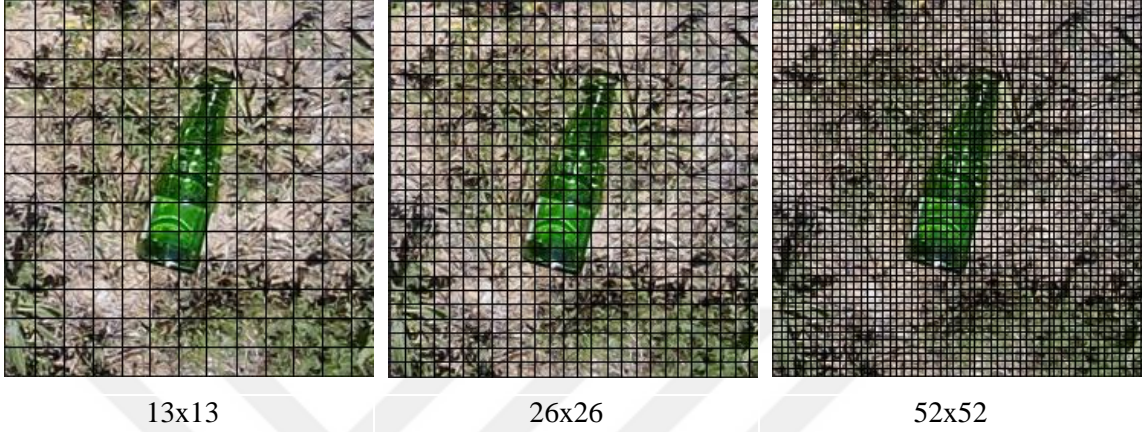
YOLOv3 ağı DarkNet-53 öznelik tanımlayıcısı ve YOLO dedektörü olmak üzere iki bölümden oluşmaktadır. DarkNet-53 mimarisi ImageNet’te eğitilmiş bir evrişimsel sinir ağıdır. C ve CUDA programlama dillerinde yazılmış, hızlı, kurulumu basit hem dahili hem de harici ekran kartlarını destekleyen ve github platformu üzerinden ücretsiz erişilebilen bir evrişimsel sinir ağı yapısıdır [40].

YOLOv3’ün kullandığı orijinal DarkNet-53 mimarisinde 53 adet katman bulunmaktadır. Ancak algılama görevi için 53 katman daha istiflenir ve toplam 106 katman oluşur. Bu katman yapısı ardışık olarak  $1 \times 1$  ve  $3 \times 3$  boyutlarındaki evrişimsel katmanı takip eden artık (residual) katmandan oluşmaktadır [48]. DarkNet-53 ağı Resnet-152 ağı ile aynı doğruluğu sağlarken 2 kat hızlı çalışır. Şekil 5.9’da DarkNet-53 katman mimarisi gösterilmiştir.

Katman	Filtre	Boyut	Çıktı	
	Evrişimsel	32	3x3	256x256
	Evrişimsel	64	3x3/2	128x128
<b>1x</b>	Evrişimsel	32	1x1	
	Evrişimsel	64	3x3	
	Artık			128x128
	Evrişimsel	128	3x3/2	64x64
<b>2x</b>	Evrişimsel	64	1x1	
	Evrişimsel	128	3x3	
	Artık			64x64
	Evrişimsel	256	3x3/2	32x32
<b>8x</b>	Evrişimsel	128	1x1	
	Evrişimsel	256	3x3	
	Artık			32x32
	Evrişimsel	512	3x3/2	16x16
<b>8x</b>	Evrişimsel	256	1x1	
	Evrişimsel	512	3x3	
	Artık			16x16
	Evrişimsel	1024	3x3/2	8x8
<b>4x</b>	Evrişimsel	512	1x1	
	Evrişimsel	1024	3x3	
	Artık			8x8
Havuzlama	Global			
Tam Bağlantılı	1000			
Softmax				

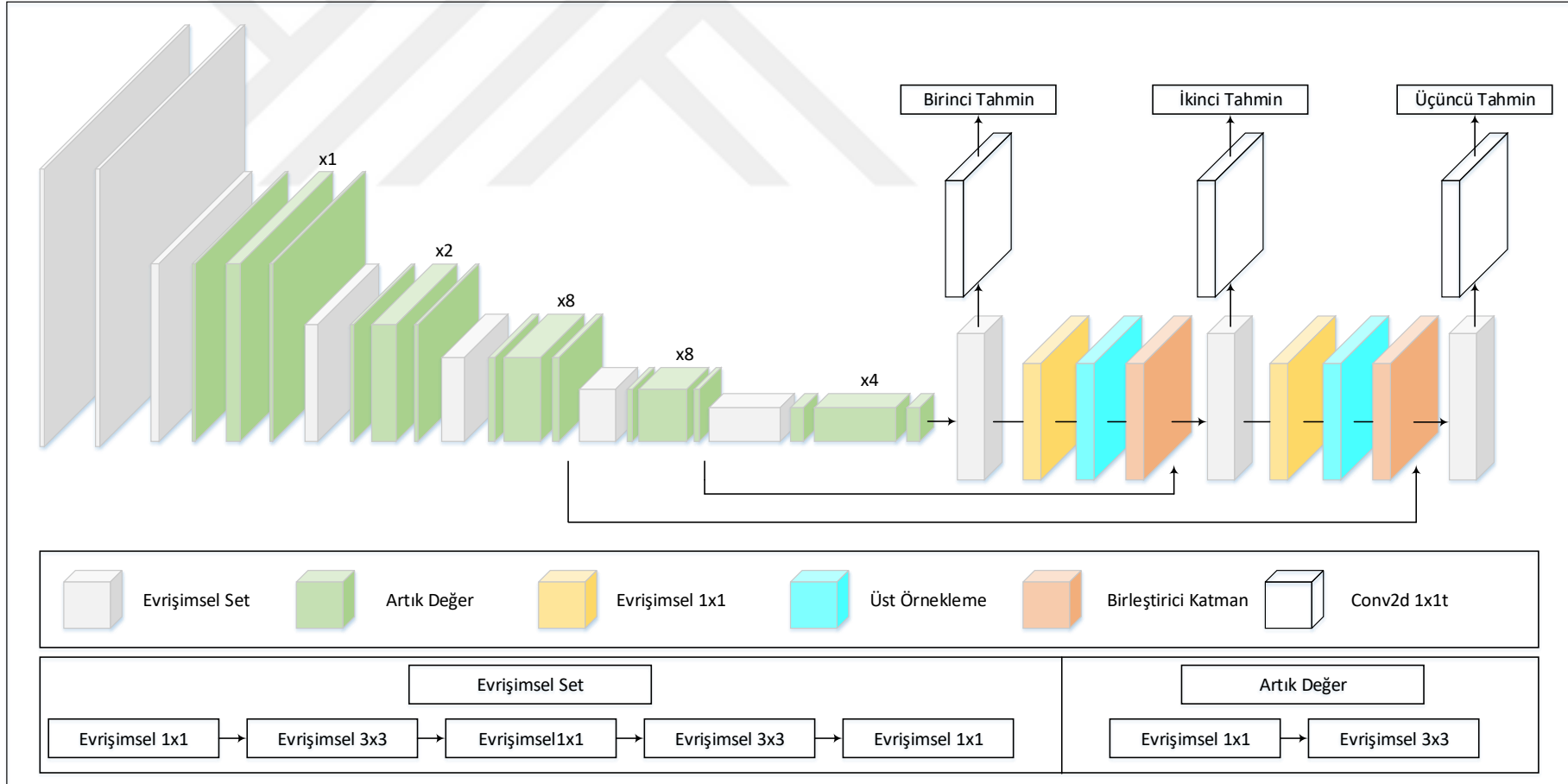
Şekil 5.9. DarkNet-53 mimarisi [40]

YOLOv3 algoritmasında DarkNet-53 mimarisinin kullanılabilmesi için standart YOLO algoritmasında bulunan havuzlama, tam bağlantılı ve softmax katmanları kaldırılmıştır. YOLOv3 farklı boyutlardaki nesnelere algılamak için görüntüyü 3 farklı ölçekte ızgaralara böler. Şekil 5.10'da bir görüntünün 3 farklı ölçekte ızgaralara bölünmüş hali görülmektedir.



**Şekil 5.10.** Görüntüyü üç farklı ölçekte ızgaralar ile bölme

YOLOv3 evrimsel sinir ağı algoritmasında girdi görüntüsünün  $416 \times 416$  boyutunda olması durumunda dedektör katmanlarına bağlanan vektör boyutları  $13 \times 13$ ,  $26 \times 26$  ve  $52 \times 52$  olacaktır. Burada  $13 \times 13$ 'lük ızgaralama büyük boyuttaki nesnelere algılanmasını,  $26 \times 26$ 'lık ızgaralama orta boyuttaki nesnelere algılanmasını,  $52 \times 52$ 'lik ızgaralama ise küçük boyuttaki nesnelere algılanmasını sağlar [41].



Şekil 5.11. DarkNet-53 kullanılmış 3 ölçekli tahmin yapabilen YOLOv3 evrişimsel sinir ağı yapısı [49]

Şekil 5.11’de DarkNet-53 kullanılmış 3 ölçekli tahmin yapabilen YOLOv3 evrişimsel sinir ağı yapısı görülmektedir. Bu yapıda YOLOv3 çoklu ölçekler kullandığından tahmin doğruluğu da artmıştır. YOLOv3 evrişimsel sinir ağında kullanılan DarkNet-53 yapısı DarkNet-19 yapısından daha güçlü, ResNet-101 ve ResNet-152 yapılarından daha verimlidir. Şekil 5.11’de görüldüğü gibi YOLOv3 test süreci 7 aşamadan oluşmaktadır.

- 1. Aşamada;** girişteki görüntü standart boyutlara dönüştürülür.
- 2. Aşamada;** giriş görüntüsü 3 farklı ölçekte ( $13 \times 13$ ,  $26 \times 26$  ve  $52 \times 52$ ) ızgaralara bölünür. Bir nesnenin merkez noktası ızgaranın içerisine denk geliyorsa ızgara birimi nesneyi tahmin eder.
- 3. Aşamada;** sınırları belirlemek için K-means kümeleme kullanılır. Her ızgara biriminde 3 küme vardır. 3 farklı ölçek kullanıldığından toplamda  $3 \times 3 = 9$  küme oluşur.
- 4. Aşamada;** özellik çıkarımı için görüntü ağı girilir. Ağda öncelikle  $13 \times 13$  küçük ölçekte özellik haritası çıkarılır.
- 5. Aşamada;**  $13 \times 13$  küçük ölçekte özellik haritası ilk olarak evrişim setine tabi tutulur, 2 kez örneklenir daha sonra 1. tahmin sonucu ortaya çıkar.  $26 \times 26$  orta ölçekte özellik çıkarım aşamasına geçilir ve özellik haritası çıkarılır.
- 6. Aşamada;**  $26 \times 26$  orta ölçekli özellik haritası evrişim setine tabi tutulur, 2 kez örneklenir daha sonra 2. tahmin sonucu ortaya çıkar.  $52 \times 52$  büyük ölçekte özellik çıkarım aşamasına geçilir ve özellik haritası çıkarılır.
- 7. Aşamada;** 3. tahmin sonucu da ortaya çıktıktan sonra 3 tahmin sonucu birleştirilir. Bu birleşim sonucunda  $((52 \times 52) + (26 \times 26) + (13 \times 13)) \times 3 = 10.647$  adet sınır kutu tahmini yapılır. Daha sonra tahmin sonuçları kullanılarak kutu güven puanı elde edilir. Bu sınırlayıcı kutuların büyük bir kısmı çok düşük kutu güven puanına sahip olacağından Non Maximum Suppression (NMS) yöntemiyle kutu güven puanı eşik değer (YOLOv3 te genellikle 0,5 kabul edilir) altında olanlar silinir. En yüksek kutu güven puanına sahip kutu nihai sonuç olarak bulunur [49].

R-CNN algoritması, verilen veri setine bağlı olarak görüntü üzerinde nesnenin bulunacağı muhtemel bölgeleri belirleyerek bu bölgeler üzerinde evrişimsel sinir ağları (CNN) işlemlerini yürütmektedir. Birden fazla bölgede CNN algoritması yürütüldüğünde bu sistem gerçek zamanlı nesne tespitinde oldukça yavaş kalmaktadır [50]. YOLOv3 algoritması dışındaki derin öğrenme algoritmaları özellikle gerçek zamanlı nesne tespitinde gerekli olan en düşük görüntü işleme hızı olan 20 saniye başına kareye (FPS) ulaşamamışlardır. YOLOv3 algoritması ise görüntüleri 40-90 FPS’de işleyebilmektedir. Bu değerler videonun gerçek zamanlı olarak birkaç milisaniye gecikme ile YOLOv3 algoritması tarafından işlenebileceğini gösterir. Bundan dolayı diğer evrişimsel sinir ağı algoritmalarına göre oldukça hızlı çalışır. Anlık algılamaların önemli olduğu durumlarda algılama zamanının düşük olması çok büyük fayda sağlamaktadır.

YOLOv3 ile diğerk nesne tespit algoritmaları olan R-CNN ve Faster R-CNN algoritmaları karşılaştırıldığında, YOLOv3'ün R-CNN den 1000 kat, Faster R-CNN den ise 100 kat daha hızlı olduđu söylenebilir. Bu yüzden gerçek zamanlı görüntü işlemede hızlı işlem yapabilen YOLOv3 tercih edilmektedir [42]. Tez çalışmasında hızlı işlem yapılabilmesi ve tekrar eğitildiğinde yüksek nesne tespit doğruluğuna ulaşabilmesinden dolayı YOLOv3 evrişimsel sinir ağı yapısı kullanılmıştır.



## 6. MATERYAL VE METOT

Tez çalışmasının ilk aşaması olan literatür araştırmasına göre, son yıllarda sıklıkla kullanılan görüntü işleme ve derin öğrenme algoritmalarının nesne tespitinde hız ve doğruluk açısından gösterdikleri üstün başarı, gerçek zamanlı nesne tespitinde tercih edilmelerinin en büyük sebeplerinden biridir. Bu çalışmada hız ve doğruluk açısından diğer algoritmalarından daha üstün performans gösteren YOLOv3 nesne tanıma algoritması kullanılarak pet ve cam şişe tespiti yapılmıştır.

### 6.1. Sistem Mimarisi

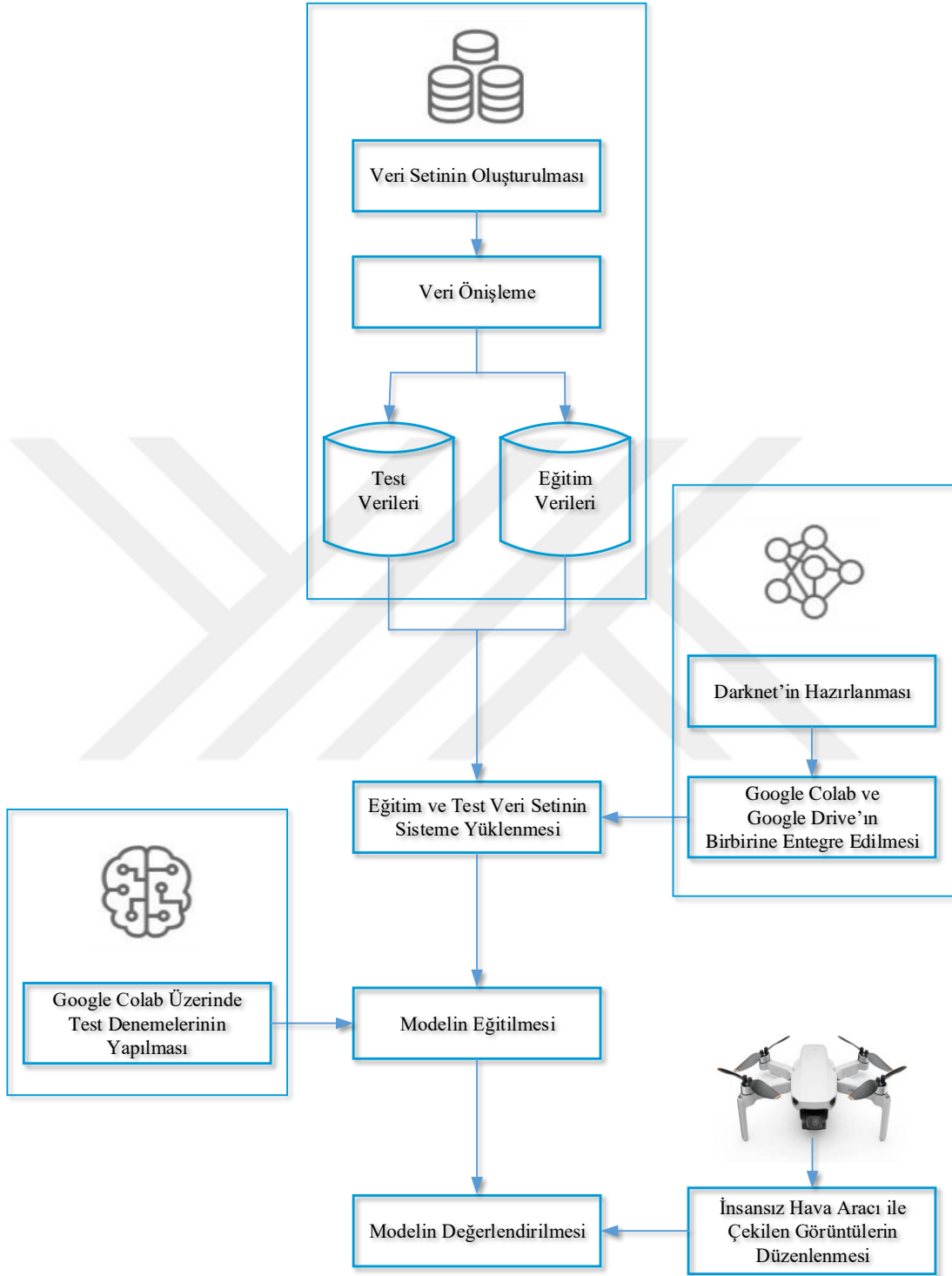
Gerçekleştiren tez çalışmasında Şekil 6.1'de gösterilen sistem mimarisi kullanılmıştır. Bu mimari genel olarak 3 bölümden oluşmaktadır.

Birinci bölümde veri seti oluşturulmaktadır. İnternet ortamından elde edilen içerisinde pet ve cam şişelerin bulunduğu 2500 görüntü veri ön işleme yapılarak incelenmiş, gerekli ayıklamalar sonucunda veri seti için kullanılacak 2250 görüntü belirlenmiştir. Bir sonraki aşamada her bir görüntüye ait nesne etiketleme dosyaları YOLOv3 formatına uygun olarak düzenlenmiştir. Veri setinin %70'i eğitim, %30'u da test verisi olmak üzere ikiye ayrılarak sisteme yüklenmiştir.

İkinci bölümde DarkNet-53 mimarisinin çalışması için gerekli ayarlamalar yapılmış Colab arayüzü ile Google drive'ın birbirine entegre işlemi gerçekleştirilmiştir. Bu bölümde ayrıca eğitim işlemine geçilmeden önce YOLOv3 standart ağırlıkları Colab ortamına yüklenmiş ve tüm veriler eğitim işlemine hazır hale getirilmiştir.

Üçüncü bölümde ise oluşturulan veri seti, YOLOv3 standart ağırlık dosyası ve diğer gerekli veriler kullanılarak evrimsel sinir ağı vasıtasıyla nesne tanıma için gerekli eğitim işlemi gerçekleştirilmiştir. Uzun süren eğitim işlemi ile değerlendirme işleminde kullanılacak ağırlıklar belirlenmiş ve değerlendirme işlemine geçilmiştir.

Değerlendirme işleminin gerçekleştirilmesi için nesne tespiti yapılacak görüntülerin elde edilmesi gerekmektedir. Bu işlem için DJI firmasının üretmiş olduğu Mavic Air 2 İHA kullanılmıştır. Fırat üniversitesi Teknik Bilimler Meslek Yüksekokulu kampüs alanında görüntülerin çekileceği coğrafi bölge belirlenmiş, bu bölgede farklı yerlerde belirlenen yükseklikten içerisinde cam ve pet şişelerin bulunduğu görüntüler çekilmiştir. Çekilen bu görüntüler incelenmiş, belirlenen 48 görüntü ve eğitim sonucunda bulunan ağırlıklar kullanılarak değerlendirme işlemini gerçekleştirilmiştir. Yapılan değerlendirme işlemi sonucunda görüntülerdeki pet ve cam şişe nesnelерinin tespit işlemi yapılmıştır. Tüm görüntü dosyaları tek tek değerlendirme işlemine tabi tutulmuş ve değerlendirme sonuçları elde edilmiştir.



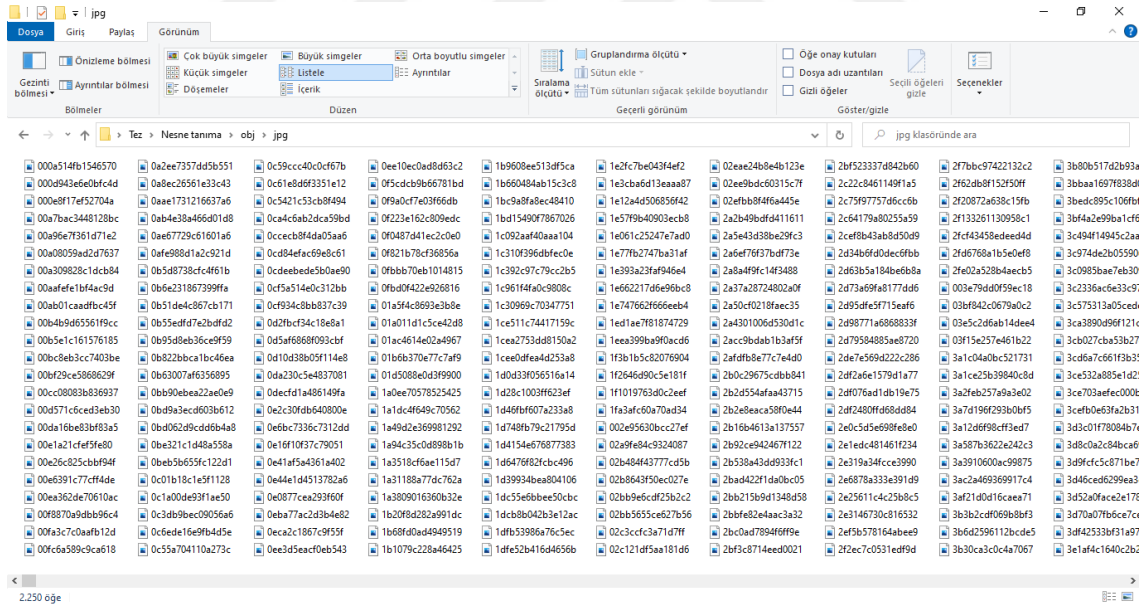
**Şekil 6.1.** Sistem mimarisi

## 6.2. Kullanılan Veri Seti

Evrişimsel sinir ağları kullanılarak nesne tespiti çalışmaları için son yıllarda farklı veri setleri geliştirilmiştir. Nesne tanıma ve sınıflandırma işleminin başarılı olabilmesinin ön şartı kullanılan veri setleridir. Veri setlerindeki görüntü dosya sayısı arttıkça nesne tanıma doğruluğu da artmaktadır. Daha doğru sonuçlar elde etmek için tez çalışmamızda çok fazla görüntü dosyası kullanarak veri seti oluşturulmuştur. Veri seti için çok fazla miktarda görüntü dosyasının toplanması ve etiketlenmesi uzun zaman alacağından bu işlem için kategori bazında hazırlanmış hazır veri setleri kullanılmıştır.

Google Open Image hazır veri setinden içerisinde tespit yapacağımız pet ve cam şişe bulunan 2500 görüntü dosyası ve her bir görüntüye ait txt formatındaki etiketleme dosyaları internet ortamından indirilmiştir. Bu dosyalar incelenerek tekrar eden resimler ayıklanmıştır. Ayıklama sonucunda kalan 2250 görüntü ve YOLOv3 formatına dönüştürülmüş etiketleme dosyası veri setini oluşturmak için kullanılmıştır.

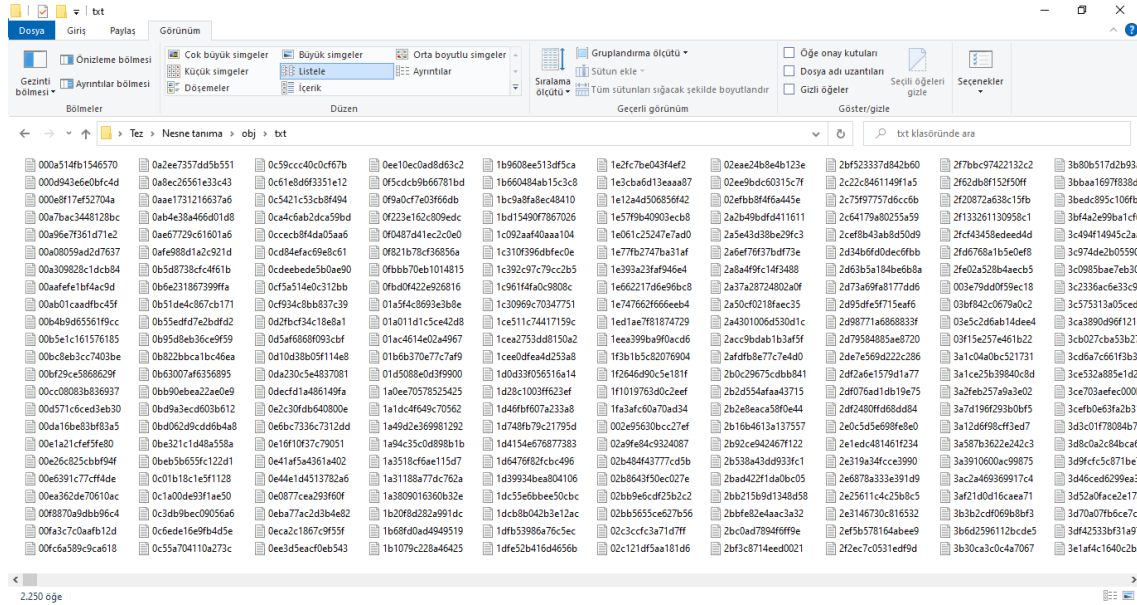
Oluşturulan veri setinde rastgele olarak verilerin %70'i eğitim, %30'u ise test verisi olarak belirlenmiştir.



Şekil 6.2. Veri setinde kullanılan jpg uzantılı dosyalar

Şekil 6.2'de veri setinde kullanılan görüntü dosyaları görülmektedir. Kullanılan veri seti dosyaları jpg formatındadır. 2250 adet jpg uzantılı dosya eğitim için hazırlanmıştır. Şekil 6.3'de ise veri setinde kullanılan görüntü dosyalarına ait etiketleme dosyaları görülmektedir. Etiketleme dosyaları txt uzantılıdır. Her bir görüntü dosyası için bir tane txt dosyası oluşturulmuştur. Bu

dosyaların içerisinde görüntü dosyalarında tespit edeceğimiz nesneleri sınırlayan sınır kutusunun merkez koordinat bilgileri, genişlik ve yükseklik bilgileri bulunmaktadır.

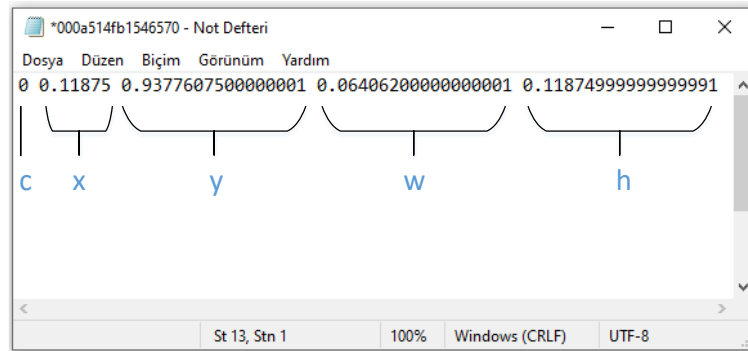


Şekil 6.3. Veri setinde kullanılan txt uzantılı dosyalar

Veri setimiz 2250 adet jpg uzantılı 2250 adet txt uzantılı olmak üzere toplam 4500 adet dosyadan oluşmaktadır.

### 6.3. Veri Etiketleme

Veri setinde kullanılan her bir görüntü dosyası için oluşturulan txt uzantılı etiketleme dosyasında, görüntü dosyasında etiketlenen bir veya birden fazla nesneye ait koordinat ve sınıf bilgileri bulunmaktadır. Şekil 6.4'de txt uzantılı etiketleme yapılmış örnek veri dosyası içeriği gösterilmiştir.



Şekil 6.4. Etiketlenmiş örnek veri dosyası içeriği

Şekil 6.4’de gösterilen dosya içeriğine bakıldığında tespit edilecek nesneye ait beş bilgi rakamsal olarak görülmektedir. Bu bilgiler  $c$ ,  $x$ ,  $y$ ,  $w$  ve  $h$  ifadeleridir.

Bu ifadeler;

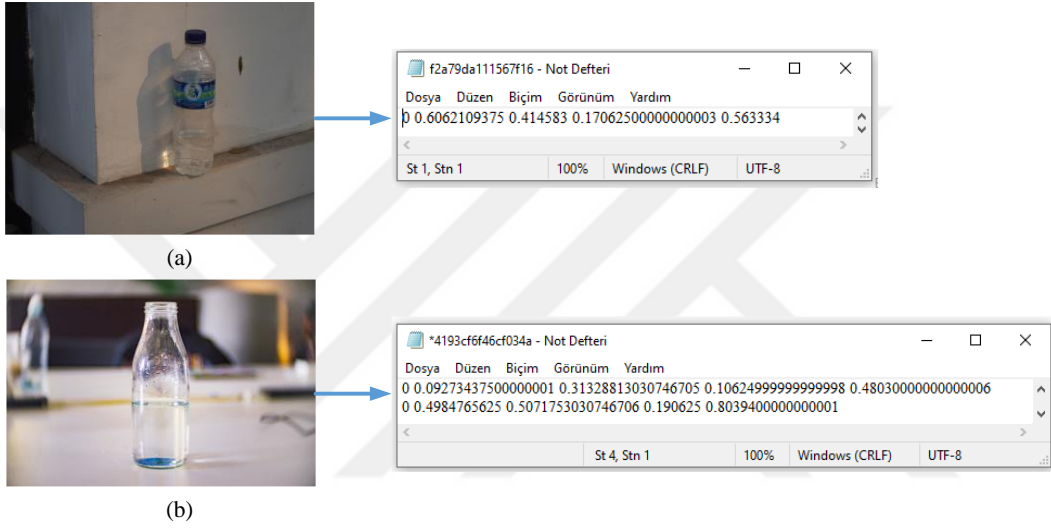
$c$  : Ait olduğu sınıf

$x$  : Etiketlenen bölgenin merkez noktasının  $x$  koordinatı

$y$  : Etiketlenen bölgenin merkez noktasının  $y$  koordinatı

$w$  : Etiketlenen bölgenin genişliği

$h$  : Etiketlenen bölgenin yüksekliği



Şekil 6.5. Veri setinde kullanılan örnek görüntüler ve etiketlenmiş veri dosyası içerikleri

Şekil 6.5’de veri setinde kullanılan iki görüntü dosyası ve bu dosyalara ait etiketlenmiş veri dosyası içerikleri görülmektedir. Şekil 6.5.a’daki görüntüde tek bir şişe nesnesi olduğundan tek satırlık rakamsal etiketleme verisi görülmektedir. Şekil 6.5.b’de ise görüntüde iki şişe nesnesi olduğundan 2 satırlık rakamsal veri görülmektedir. Burada tek bir nesne türü tespit edileceğinden  $c$  (sınıf) değeri 0 olarak belirtilmiştir. Birden fazla nesne türünün tespiti yapıldığında ise nesne türüne bağlı olarak  $c$  1,2,3... gibi değerler almaktadır.

## 6.4. Kullanılan Yöntem

Veri seti hazırlandıktan sonra eğitim işleminin gerçekleştirilmesine geçilir. Eğitim işlemini gerçekleştirmek için YOLOv3 derin öğrenme algoritması kullanılmıştır. Bu algoritma ile eğitimin gerçekleşmesi için Google Research tarafından sağlanan Colobratory (Colab) hizmeti kullanılmıştır. Colab, Python programlama dilini destekleyen ve derin öğrenme için gerekli yüksek hızlı GPU hizmetine ücretsiz erişim sağlanan Jupyter not defteri hizmetidir. Bu hizmeti

kullanabilmek için Google hesabına ihtiyaç vardır. Colab vasıtasıyla eğitim işleminde gerekli olan Python programlama diline ait kodlar yazılıp çalıştırılabilir. Tesla K80 GPU hizmetini ücretsiz olarak sunan bu ortamda daha hızlı ve daha uzun süre eğitim gerçekleştirebilmek için Colab Pro'nun kullanılması gerekmektedir.

#### 6.4.1. Eğitim Önışlemleri

Colab ile eğitimin gerçekleştirilmesi için öncelikle ön ayarların yapılması gerekmektedir. Colab kullanımı için GPU tanımlaması yapıldıktan sonra YOLOv3 DarkNet-53 altyapısını kullandığından DarkNet'in kurulumu gerekmektedir. Bunu için öncelikle Şekil 6.6'da gösterilen DarkNet dizin kurulumunun yapılması gereklidir.

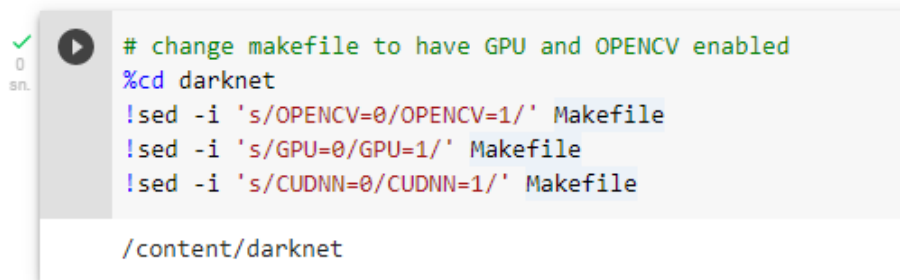


```
[1] # clone darknet repo
!git clone https://github.com/AlexeyAB/darknet

Cloning into 'darknet'...
remote: Enumerating objects: 15460, done.
remote: Total 15460 (delta 0), reused 0 (delta 0), pack-reused 15460
Receiving objects: 100% (15460/15460), 14.07 MiB | 22.62 MiB/s, done.
Resolving deltas: 100% (10385/10385), done.
```

Şekil 6.6. Colab platformuna DarkNet kurulumu

DarkNet dizin kurulumu gerçekleştirildikten sonra Şekil 6.7'de görüldüğü gibi Makefile dosyası içerisinde OPENCV, GPU ve CUDNN değerleri 1 yapılır.



```
# change makefile to have GPU and OPENCV enabled
%cd darknet
!sed -i 's/OPENCV=0/OPENCV=1/' Makefile
!sed -i 's/GPU=0/GPU=1/' Makefile
!sed -i 's/CUDNN=0/CUDNN=1/' Makefile

/content/darknet
```

Şekil 6.7. Makefile Konfigrasyonu

Makefile dosyasının konfigürasyonu bittikten sonra Şekil 6.8'de görüldüğü gibi ekran katı için gerekli kurulumlar yapılır.

```
✓ 0 sn. # verify CUDA
! /usr/local/cuda/bin/nvcc --version

nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2020 NVIDIA Corporation
Built on Mon_Oct_12_20:09:46_PDT_2020
Cuda compilation tools, release 11.1, V11.1.105
Build cuda_11.1.TC455_06.29190527_0
```

Şekil 6.8. NVIDIA-CUDA kurulumu

NVIDIA ekran kartı için CUDA kurulumu gerçekleştirildikten sonra DarkNet yapısının kurulumuna geçilir. DarkNet yapısının kurulum aşamasından sonra Şekil 6.9’da görüldüğü gibi YOLOv3 standart ağırlıkları sisteme yüklenir.

```
✓ 6 sn. # get yolov3 pretrained coco dataset weights
! wget https://pjreddie.com/media/files/yolov3.weights

--2022-08-01 09:14:03-- https://pjreddie.com/media/files/yolov3.weights
Resolving pjreddie.com (pjreddie.com)... 128.208.4.108
Connecting to pjreddie.com (pjreddie.com)|128.208.4.108|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 248007048 (237M) [application/octet-stream]
Saving to: 'yolov3.weights'

yolov3.weights      100%[=====>] 236.52M  41.6MB/s   in 6.1s

2022-08-01 09:14:10 (38.7 MB/s) - 'yolov3.weights' saved [248007048/248007048]
```

Şekil 6.9. YOLOv3 standart ağırlıkların yüklenmesi

Ağırlıkların yükleme işlemi bittikten sonra Şekil 6.10’da görüldüğü gibi veri setimizi ve değerlendirme için kullanacağımız görüntüleri yüklemek ve kullanabilmek için Google drive ile ilişkilendirme işlemi yapılır.

```
✓ 53 sn. %cd ..
from google.colab import drive
drive.mount('/content/gdrive')

/content
Mounted at /content/gdrive

[ ] # this creates a symbolic link so that now the path /content/gdrive/My\ Drive/ is equal to /mydrive
! ln -s /content/gdrive/My\ Drive/ /mydrive
! ls /mydrive
```

Şekil 6.10. Google drive bağlantısı

Google Drive bağlantısı yapıldıktan sonra daha önce oluşturduğumuz veri seti dosyalarımızı Şekil 6.11’de görüldüğü gibi Google drive içerisine yükledikten sonra DarkNet dizinine aktarma işlemi gerçekleştirilir.

```
✓ 7 sn. # copy the .zip file into the root directory of cloud VM
!cp /mydrive/yolov3/obj.zip ../

[ ] # unzip the zip file and its contents should now be in /darknet/data/obj
!unzip ../obj.zip -d data/
```

Şekil 6.11. Veri seti dosyalarının yüklenmesi

Veri seti dosyaları yüklendikten sonra eğitim işlemine geçmek için son olarak yolov3\_custom.cfg, obj.names ve obj.data dosyalarının DarkNet dizini yapısına yüklenmesi gerekmektedir. Şekil 6.12’de bu dosyaların yüklenme aşası görülmektedir.

```
✓ 0 sn. # upload the custom .cfg back to cloud VM from Google Drive
!cp /mydrive/yolov3/yolov3_custom.cfg ./cfg

# upload the custom .cfg back to cloud VM from local machine (uncomment to use)
#%cd cfg
#upload()
#%cd ..

✓ 0 sn. # upload the obj.names and obj.data files to cloud VM from Google Drive
!cp /mydrive/yolov3/obj.names ./data
!cp /mydrive/yolov3/obj.data ./data

# upload the obj.names and obj.data files to cloud VM from local machine (uncomment to use)
#%cd data
#upload()
#%cd ..
```

Şekil 6.12. yolov3\_custom.cfg, obj.names ve obj.data dosyalarının yüklenmesi

obj.names dosyası içerisinde sınıf sayısı, eğitim ve test dosya isimleri ve eğitim sonundaki ağırlıkların bulunduğu dosyanın yükleneceği klasörün isim bilgileri, obj.data dosyası içerisinde ise tespit edilecek nesneye ait tür ismi bulunmaktadır. yolov3\_custom.cfg dosyasında sınıf sayısına bağlı olarak yapılması gereken ayarlar bulunmaktadır. Bu ayarlardan biri kullanılacak filtre sayısıdır. Filtre sayısı (sınıf +5)×3 formülüne göre bulunur. Bu tez çalışmasında sınıf sayısı 1 olduğundan filtre sayısı (1+5)×3=18 olarak bulunur. Şekil 6.13’de obj.names, obj.data ve yolov3\_custom.cfg dosyalarının içeriği gösterilmiştir.

```
obj.names - Not Defteri
Dosya Düzen Biçim Görünüm Yardım
sise

obj data - Not Defteri
Dosya Düzen Biçim Görünüm Yardım
classes =1
train = data/train.txt
valid = data/test.txt
names = data/obj.names
backup = /mydrive/yolov3/backup/

*yolov3_custom - Not Defteri
Dosya Düzen Biçim Görünüm Yardım
[convolutional]
size=1
stride=1
pad=1
filters=18
activation=linear

[yolo]
mask = 6,7,8
anchors = 10,13, 16,30, 33,23, 30,61, 62,45, 59,119, 116,90, 156,198, 373,326
classes=1
num=9
jitter=.3
ignore_thresh = .7
truth_thresh = 1
random=1
```

Şekil 6.13. yolov3\_custom.cfg, obj.names ve obj.data dosyalarının içeriği

Tüm dosyalar yüklendikten ve gerekli ayarlar yapıldıktan sonra eğitim işlemi aşamasına geçilir.

#### 6.4.2. Eğitim İşlemleri

Eğitim işlemi, evrişimsel sinir ağı tarafından veri seti ve standart YOLOv3 ağırlıkları kullanılarak gerçekleştirilir. Bu işlem sonunda son ağırlık değerleri hesaplanmış ve bu ağırlıklar değerlendirme işleminde kullanılmıştır. Eğitimin süresi, kullanılan grafik işlemci birimi (GPU) ve seçilen veri setinin büyüklüğüne göre değişir. Bu zamanın kısaltılması için hızlı ve kaliteli grafik işlemci birimi kullanılması gerekir. Ağırlıkların istenilen değere düşmemesi sonucunda eğitim işleminin sürekli tekrarlanması da eğitim süresini arttırır.

Eğitim işleminin sonucunda toplam hata ve ortalama hata (avg loss) değerlerinin mümkün olduğunca sifıra yakın olması istenmektedir. Eğer toplam hata ve ortalama hata değerleri sifıra yakınsamıyorsa nesne tespitinde istenilen başarı elde edilemez. Tez çalışmasında eğitim işlemi için 3000 iterasyon gerçekleştirilmiştir. Bu işlem ile ağırlıklar 3000 defa güncellenmiştir. İşlem sonucunda hesaplanan toplam hata ve ortalama hatanın sifıra yaklaştığı görülmüştür. Yapılan çalışmalarda toplam hata ve ortalama hata değerlerinin 2'nin altına düşmesi durumunda yapılan eğitimin başarılı olduğu kabul edilmiştir.

```
+ Kod + Metin Drive'a kopyala RAM Disk Düzenleme
# train your custom detector
./darknet detector train data/obj.data cfg/yolov3_custom.cfg darknet53.conv.74 -dont_show
total_bbox = 152, rewritten_bbox = 0.000000 %
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 82 Avg (IOU: 0.461002), count: 6, class_loss = 199.00)
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 94 Avg (IOU: 0.382449), count: 13, class_loss = 1130.4)
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 106 Avg (IOU: 0.000000), count: 1, class_loss = 4287.4)
total_bbox = 171, rewritten_bbox = 0.000000 %
1: 1872.135010, 1872.135010 avg loss, 0.000000 rate, 5.238960 seconds, 64 images, -1.000000 hours left
loaded: 0.000138 seconds
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 82 Avg (IOU: 0.320821), count: 4, class_loss = 199.81)
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 94 Avg (IOU: 0.000000), count: 1, class_loss = 1129.7)
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 106 Avg (IOU: 0.000000), count: 1, class_loss = 4279.4)
total_bbox = 175, rewritten_bbox = 0.000000 %
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 82 Avg (IOU: 0.339078), count: 9, class_loss = 200.25)
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 94 Avg (IOU: 0.000000), count: 1, class_loss = 1128.2)
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 106 Avg (IOU: 0.000000), count: 1, class_loss = 4277.4)
```

Şekil 6.14. Ağırlık değerlerinin hesaplanması (1. iterasyon)

Şekil 6.14'de 1. iterasyonda yani eğitime başlandığında ağırlık değerlerinin hesaplanma işlemi görülmektedir. Başlangıçta toplam hata ve ortalama hata değerlerinin 1872.135010 olduğu görülmektedir. Bu değerler iterasyon sayısı arttıkça azalmakta ve sıfıra yaklaşmaktadır. Eğitimin aşamasının ortasına gelindiğinde ise toplam hatanın 2.106135 ve ortalama hatanın ise 1.412934 değerini aldığı görülmüştür. İterasyon sayısı arttıkça hata değerlerinde çok küçük aralıklarla azalma devam etmiştir.

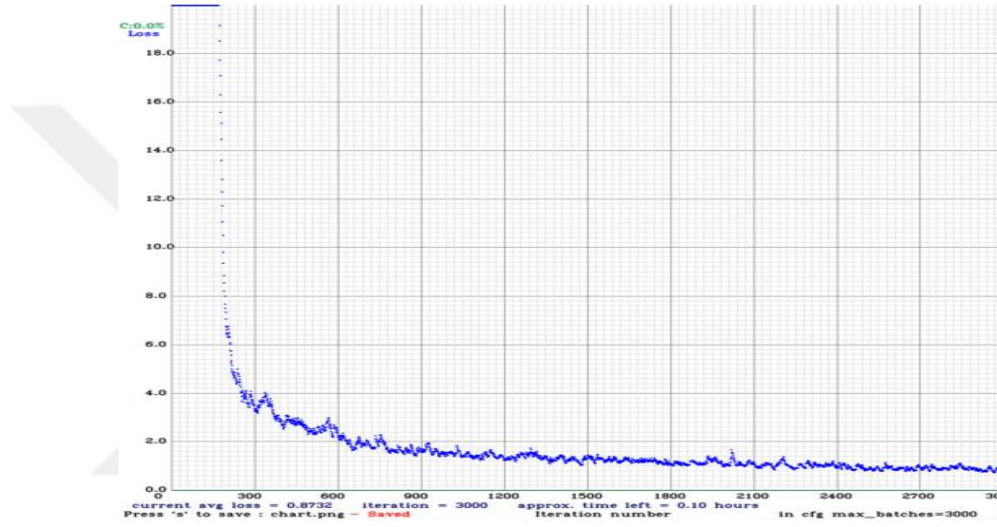
Şekil 6.15'de ise 3000. iterasyonda yani eğitim bitirildiğinde ağırlık değerlerinin hesaplanma işlemi görülmektedir. Eğitim bittiğinde toplam hata değerinin 0.818663 ve ortalama hata değerlerinin 0.873166 olduğu görülmektedir. Bu değerlerin eğitim işleminin sonunda sıfıra yaklaşıldığı görülmektedir. Bu değerler aynı zamanda eğitim işleminin başarıyla gerçekleştiğini de göstermektedir.

```
+ Kod + Metin Drive'a kopyala RAM Disk Düzenleme
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 94 Avg (IOU: 0.738835), count: 1, class_loss = 0.0000)
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 106 Avg (IOU: 0.000000), count: 1, class_loss = 0.0000)
total_bbox = 591522, rewritten_bbox = 0.622293 %
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 82 Avg (IOU: 0.794146), count: 6, class_loss = 1.0949)
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 94 Avg (IOU: 0.000000), count: 1, class_loss = 0.0017)
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 106 Avg (IOU: 0.000000), count: 1, class_loss = 0.0000)
total_bbox = 591528, rewritten_bbox = 0.622287 %
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 82 Avg (IOU: 0.805552), count: 5, class_loss = 0.8659)
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 94 Avg (IOU: 0.774195), count: 2, class_loss = 0.0486)
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 106 Avg (IOU: 0.000000), count: 1, class_loss = 0.0000)
total_bbox = 591535, rewritten_bbox = 0.622279 %
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 82 Avg (IOU: 0.814905), count: 13, class_loss = 0.502)
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 94 Avg (IOU: 0.749030), count: 1, class_loss = 0.3302)
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 106 Avg (IOU: 0.000000), count: 1, class_loss = 0.0000)
total_bbox = 591549, rewritten_bbox = 0.622265 %
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 82 Avg (IOU: 0.818317), count: 3, class_loss = 0.2126)
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 94 Avg (IOU: 0.605975), count: 1, class_loss = 0.2164)
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 106 Avg (IOU: 0.000000), count: 1, class_loss = 0.0000)
total_bbox = 591553, rewritten_bbox = 0.622260 %
3000: 0.818663, 0.873166 avg loss, 0.000010 rate, 5.219235 seconds, 192000 images, 0.104664 hours left
Saving weights to /mydrive/yolov3/backup//yolov3_custom_3000.weights
Saving weights to /mydrive/yolov3/backup//yolov3_custom_last.weights
Saving weights to /mydrive/yolov3/backup//yolov3_custom_final.weights
If you want to train from the beginning, then use flag in the end of training command: -clear
```

Şekil 6.15. Ağırlık değerlerinin hesaplanması (3000. iterasyon)

Şekil 6.15'deki işaretli bölümdeki 3000 değeri iterasyon sayısını, 0.818663 değeri toplam hatayı, 0.873166 değeri ortalama hatayı, 0.000010 değeri mevcut öğrenme oranını, 5.219235 değeri iterasyon süresini, 192000 değeri (3000×64) eğitim süresince kullanılan görüntü sayısını, 0.104664 değeri ise eğitimin bitmesi için kalan süreyi belirtmektedir.

Şekil 6.16'da tüm eğitim süreci boyunca hata değerlerinin değişim grafiği görülmektedir. Grafik incelendiğinde başlangıçta yüksek olan hata değerleri eğitimin sonuna doğru sıfıra oldukça yaklaşmıştır. 3000 iterasyon sonucunda eğitim tamamlanmış ve nesne tespitinde kullanılacak ağırlık değerleri elde edilmiştir.



Şekil 6.16. Eğitim süreci hata (loss) / iterasyon grafiği

Nesne tespiti için elde edilen bu ağırlık değerleri yolov3\_custom\_final isimli weight uzantılı bir dosya halinde Google drive içerisine kaydedilmiştir. Bu dosya gelecekte yapılacak test işlemlerinde tekrar uzun süren eğitim işlemi gerçekleştirilmeden kullanılabilir.

## 6.5. Nesne Tespit Test İşlemleri

Nesne tespit test işlemlerini gerçekleştirmek için eğitim sonucunda oluşturulmuş weight uzantılı yolov3\_custom\_final isimli dosya kullanılmıştır. Öncelikle bulunan ağırlık değerleri, içerisinde cam ve pet şişelerin bulunduğu görüntü dosyaları ile test edilmiştir. Bu işlem için internet ortamından alınan, içerisinde pet ve cam şişe bulunan aşağıdaki görüntü dosyaları kullanılmıştır.



Şekil 6.17. İnternette elde edilen görüntü dosyaları için nesne tespit test sonuçları

Şekil 6.17’de görüldüğü gibi nesne tespit test işlemi sonucunda görüntülerdeki pet ve cam şişeler farklı doğruluk oranlarında tespit edilmiştir. Bu sonuç, ağırlık dosyamızın doğal ortamda çekilen görüntü dosyalarındaki nesnelerin tespiti için de kullanılabileceğini göstermiştir.



Şekil 6.18. Görüntü çekimlerinin yapıldığı coğrafi bölge

Tez çalışmasında, ekosistem için sorun teşkil eden pet ve cam şişe atıklarının tespiti için, Fırat Üniversitesi Teknik Bilimler Meslek Yüksek Okulu kampüs alanında belirlenen coğrafi bölge içerisinde üç farklı ortamda 3 metre yükseklikten çekilen görüntüler kullanılmıştır. Şekil 6.18’de görüntü çekimlerinin yapıldığı coğrafi bölge görülmektedir.



Şekil 6.19. Nesne tespit test görüntülerinin çekildiği ortamlar

Şekil 6.19’da görüldüğü gibi nesne tespit test işlemleri için üç farklı ortam kullanılmıştır. 1. Ortam yeşilliklerin fazla olduğu, 2.ortam yeşilliklerin az olduğu, 3.ortam ise gri toprak yapısının

olduğu ortamdır. Farklı ortamlar kullanılması nesne tespit işlemimizin güvenilirliğini ve doğruluğunu arttırmaktadır.

Üç farklı ortamda İHA tarafından çekilen görüntüler kullanılarak nesne tespit test işlemleri yapılmış ve test işlemleri sonucunda görüntülerdeki pet ve cam şişe nesnelerinin tespiti gerçekleştirilmiştir.

```
105 conv 18 1 x 1/1 52 x 52 x 256 -> 52 x 52 x 18 0.025 BF
106 yolo
[yolo] params: iou loss: mse (2), iou_norm: 0.75, obj_norm: 1.00, cls_norm: 1.00,
delta_norm: 1.00, scale_x_y: 1.00
Total BFLOPS 65.304
avg_outputs = 516723
Allocate additional workspace_size = 52.43 MB
Loading weights from /mydrive/yolov3/backup/yolov3_custom_final.weights...
seen 64, trained: 192 K-images (3 Kilo-batches_64)
Done! Loaded 107 layers from weights-file
Detection layer: 82 - type = 28
Detection layer: 94 - type = 28
Detection layer: 106 - type = 28
/mydrive/images/resim24.jpg: Predicted in 17.725000 milli-seconds.
sise: 66%
Unable to init server: Could not connect: Connection refused
```

**Şekil 6.20.** Nesne tespit test işlemi

Şekil 6.20'de İHA tarafından çekilen görüntü üzerinde nesne tespit test işlemini gerçekleştiren 106 katmanlı YOLOv3 yapısı ve tespit edilen şişe nesnesinin tespit tahmin oranı görülmektedir. Çekilen tüm görüntüler bu şekilde tek tek test işlemine tabi tutulmuştur. Test sonucu elde edilen tespit değerleri ve tespit edilen nesneyi sınırlayan kutu görüntüler üzerine işaretlenmiş ve görüntü dosyalarına son hali verilmiştir. İşaretlenmiş bölüm test işlemi yapılan görüntü dosyası üzerinde 1 adet şişe nesnesinin tespit edildiğini göstermektedir.

Nesne tespit test işleminde nesne ismi olan şişenin yanında bulunan yüzdelerik bilgisi tespit edilen kutu sınırları içerisinde şişe olma ihtimalinin oranıdır. Başka bir ifade ile sınırlandırılmış kutu içerisinde %66 ihtimalle şişe olduğu tahmin edilmiştir.

İHA tarafından belirlenen coğrafi bölgedeki farklı ortamlarda çekilen 50 adet görüntü üzerinde nesne tespit test işlemi gerçekleştirilmiştir.



(a)



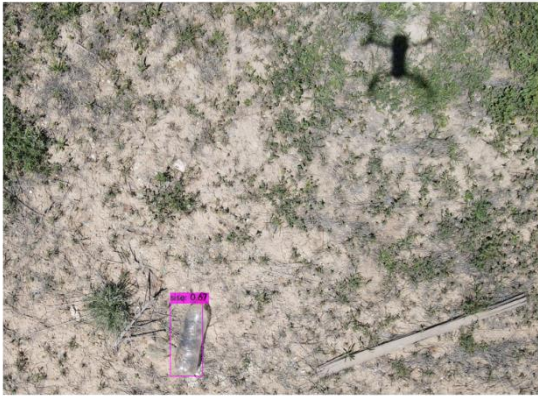
(b)



(c)



(d)



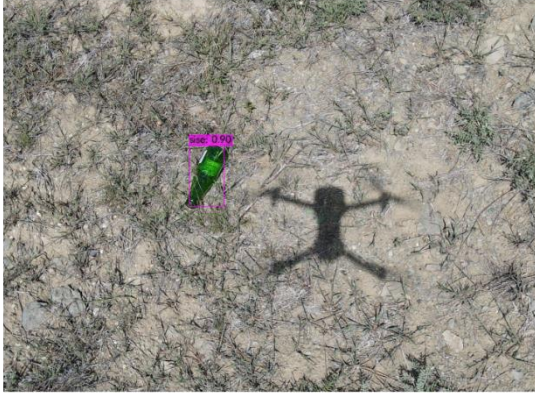
(e)



(f)

**Şekil 6.21.** Tek nesnenin bulunduğu farklı ortamlarda pet şişe tespiti

Şekil 6.21’de farklı ortamlarda bulunan pet şişelerin nesne tespit test sonuç görüntüleri verilmiştir. Çevreye atılmış pet şişenin, Şekil 6.21.a’da %79, Şekil 6.21.b’de %90, Şekil 6.21.c’de %80, Şekil 6.21.d’de %75, Şekil 6.21.e’de %67, Şekil 6.21.f’de ise %61 doğruluk oranı ile tespiti gerçekleştirilmiştir. Görüntüler incelendiğinde tek nesnenin olduğu tüm görüntülerde nesne tespitinin gerçekleştiği görülmektedir.



(a)



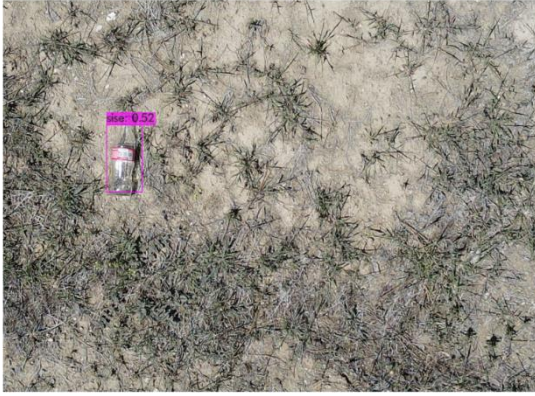
(b)



(c)



(d)



(e)



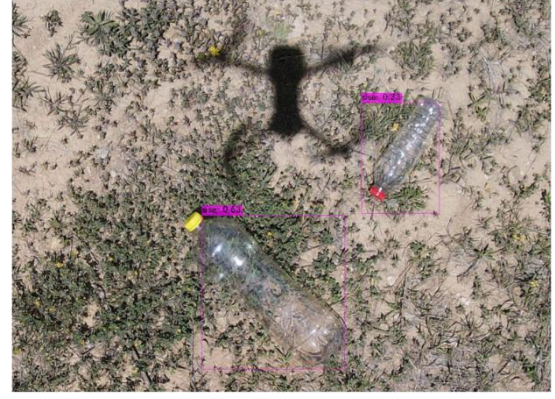
(f)

**Şekil 6.22.** Tek nesnenin bulunduğu farklı ortamlarda cam şişe tespiti

Şekil 6.22’de farklı ortamlarda bulunan cam şişelerin nesne tespit test sonuç görüntüleri verilmiştir. Çevreye atılmış cam şişenin, Şekil 6.22.a’da %90, Şekil 6.22.b’de %85, Şekil 6.22.c’de %58, Şekil 6.22.d’de %82, Şekil 6.22.e’de %52, Şekil 6.22.f’de ise %46 doğruluk oranı ile tespiti gerçekleştirilmiştir. Görüntüler incelendiğinde tek nesnenin olduğu tüm görüntülerde nesne tespitinin gerçekleştiği görülmektedir.



(a)



(b)



(c)



(d)



(e)



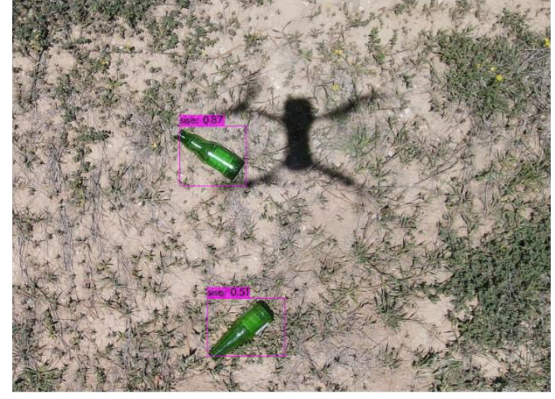
(f)

**Şekil 6.23.** İki nesnenin bulunduğu farklı ortamlarda pet şişe tespiti

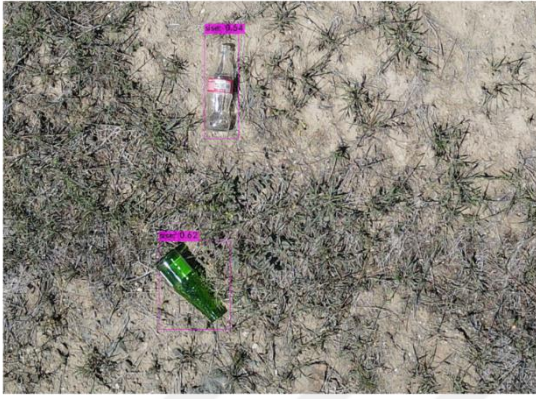
Şekil 6.23’de farklı ortamlarda bulunan pet şişelerin nesne tespit test sonuç görüntüleri verilmiştir. Çevreye atılmış pet şişelerin, Şekil 6.23.a’da %75 ve %64, Şekil 6.23.b’de %63 ve %33, Şekil 6.23.c’de sadece bir pet şişenin %86, Şekil 6.23.d’de sadece bir pet şişenin %94, Şekil 6.23.e’de sadece bir pet şişenin %54, Şekil 6.23.f’de ise sadece bir pet şişenin %95 doğruluk oranı ile tespiti gerçekleştirilmiştir. Görüntüler incelendiğinde Şekil 6.23.c, Şekil 6.23.d, Şekil 6.23.e ve Şekil 6.23.f’de bulunan diğer nesnenin tespitinin ise gerçekleştirilemediği görülmektedir.



(a)



(b)



(c)



(d)



(e)



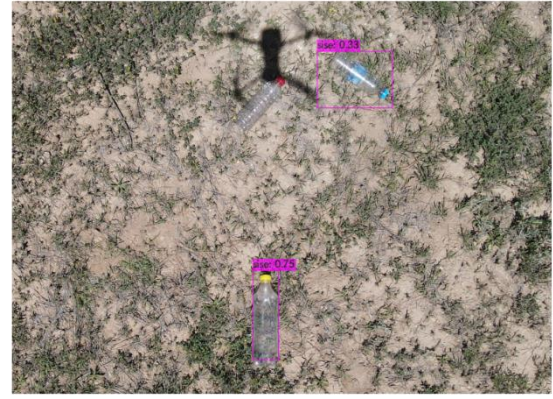
(f)

**Şekil 6.24.** İki nesnenin bulunduğu farklı ortamlarda cam şişe tespiti

Şekil 6.24'de farklı ortamlarda bulunan cam şişelerin nesne tespit test sonuç görüntüleri verilmiştir. Çevreye atılmış cam şişelerin, Şekil 6.24.a'da %38 ve %34, Şekil 6.24.b'de %87 ve %51, Şekil 6.24.c'de %64 ve %62, Şekil 6.24.d'de %93 ve %37, Şekil 6.24.e'de %87 ve %38, Şekil 6.24.f'de ise sadece bir cam şişenin %39 doğruluk oranı ile tespiti gerçekleştirilmiştir. Görüntüler incelendiğinde Şekil 6.24.f'de bulunan diğer nesnenin tespitinin ise gerçekleştirilemediği görülmektedir.



(a)



(b)



(c)



(d)



(e)



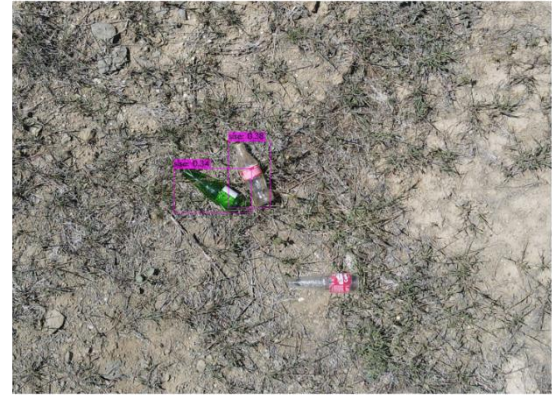
(f)

**Şekil 6.25.** Üç nesnenin bulunduğu farklı ortamlarda pet şişe tespiti

Şekil 6.25’de farklı ortamlarda bulunan pet şişelerin nesne tespit test sonuç görüntüleri verilmiştir. Çevreye atılmış pet şişelerden, Şekil 6.25.a’da iki pet şişenin %66 ve %35, Şekil 6.25.b’de iki pet şişenin %75 ve %33, Şekil 6.25.c’de iki pet şişenin %91 ve %80, Şekil 6.25.d’de iki pet şişenin %45 ve %38, Şekil 6.25.e’de bir pet şişenin %66, Şekil 6.25.f’de ise sadece bir pet şişenin %84 doğruluk oranı ile tespiti gerçekleştirilmiştir. Görüntüler incelendiğinde diğer nesnelerin tespitinin ise gerçekleştirilemediği görülmektedir.



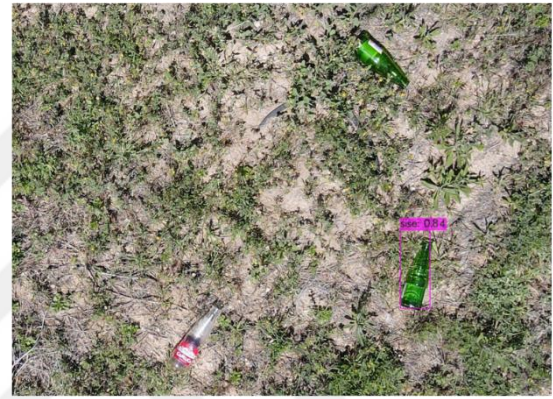
(a)



(b)



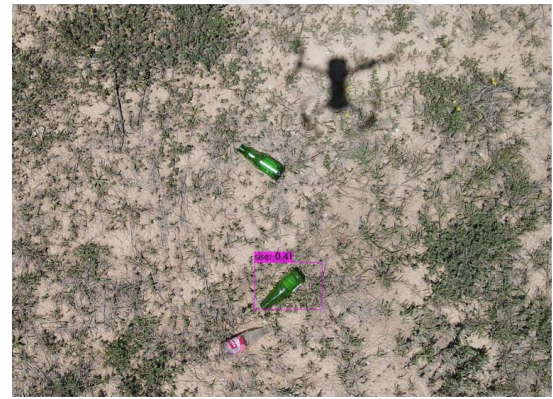
(c)



(d)



(e)



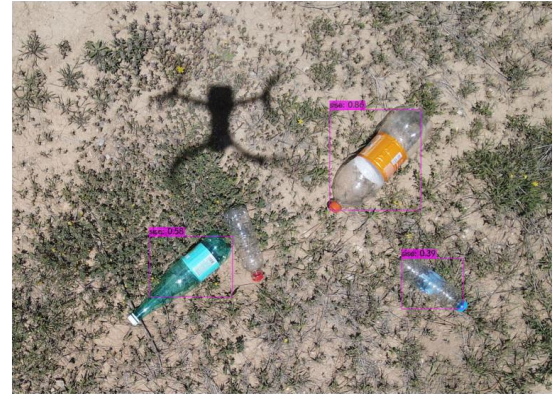
(f)

**Şekil 6.26.** Üç nesnenin bulunduğu farklı ortamlarda cam şişe tespiti

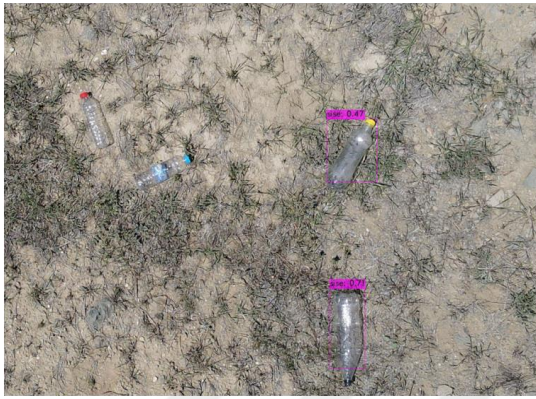
Şekil 6.26'da farklı ortamlarda bulunan pet ve cam şişelerin nesne tespit test sonuç görüntüleri verilmiştir. Çevreye atılmış cam ve pet şişelerden, Şekil 6.26.a'da iki cam şişenin %69 ve %49, Şekil 6.26.b'de iki cam şişenin %38 ve %34, Şekil 6.26.c'de iki cam şişenin %51 ve %36, Şekil 6.26.d'de bir cam şişenin %84, 6.26.e'de bir cam şişenin %83, 6.26.f'de ise bir cam şişenin %41 doğruluk oranı ile tespiti gerçekleştirilmiştir. Görüntüler incelendiğinde diğer nesnelere tespitinin ise gerçekleştirilemediği görülmektedir.



(a)



(b)



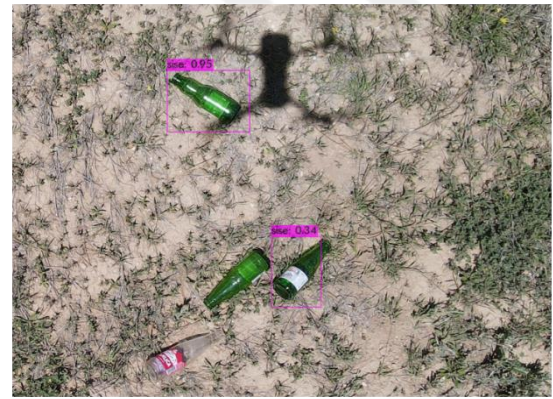
(c)



(d)



(e)



(f)

**Şekil 6.27.** Dört nesnenin bulunduğu farklı ortamlarda şişe tespiti

Şekil 6.27’de farklı ortamlarda bulunan pet ve cam şişelerin nesne tespit test sonuç görüntüleri verilmiştir. Çevreye atılmış cam ve pet şişelerden, Şekil 6.27.a’da üç pet şişenin %55, %44 ve %37, Şekil 6.27.b’de üç pet şişenin %86, %58 ve %39, Şekil 6.27.c’de iki pet şişenin %71 ve %47, Şekil 6.27.d’de iki pet şişenin %87 ve %68, 6.27.e’de iki cam şişenin %71 ve %48, 6.27.f’de ise iki cam şişenin %95 ve %34 doğruluk oranı ile tespiti gerçekleştirilmiştir. Görüntüler incelendiğinde diğer nesnelerin tespitinin ise gerçekleştirilemediği görülmektedir.



(a)



(b)



(c)



(d)



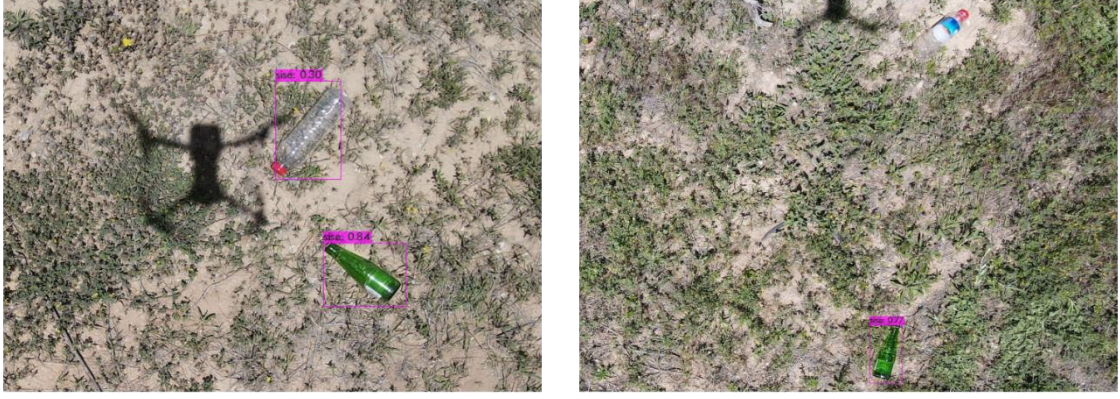
(e)



(f)

**Şekil 6.28.** Beş ve daha fazla nesnenin bulunduğu farklı ortamlarda şişe tespiti

Şekil 6.28'de farklı ortamlarda bulunan pet ve cam şişelerin nesne tespit test sonuç görüntüleri verilmiştir. Çevreye atılmış cam ve pet şişelerden, Şekil 6.28.a'da iki şişenin %84, %59, Şekil 6.28.b'de üç şişenin %70, %65 ve %42, Şekil 6.28.c'de iki cam şişenin %88 ve %34, Şekil 6.28.d'de üç cam şişenin %66, %63 ve %52, 6.28.e'de dört şişenin %58, %55, %43 ve %40, 6.28.f'de ise üç plastik şişenin %56, %52 ve %49 doğruluk oranı ile tespiti gerçekleştirilmiştir. Görüntüler incelendiğinde diğer nesnelerin tespitinin ise gerçekleştirilemediği görülmektedir.



(a)

(b)

**Şekil 6.29.** İki farklı nesnenin bulunduğu farklı ortamlarda şişe tespiti

Şekil 6.29'da farklı ortamlarda bulunan pet ve cam şişelerin nesne tespit test sonuç görüntüleri verilmiştir. Çevreye atılmış cam ve pet şişelerden, Şekil 6.29.a'da iki şişenin %84 ve %30, Şekil 6.29.b'de ise bir cam şişenin %77 doğruluk oranı ile tespiti gerçekleştirilmiştir. Görüntüler incelendiğinde 6.29.b'de diğer nesnenin tespitinin ise gerçekleştirilemediği görülmektedir.

Nesne tespit test sonuçları incelendiğinde İHA ile çekilmiş 50 adet görüntüde bulunan 143 nesnenin 84 tanesi tespit edilerek % 58.74 oranında tespit başarısı sağlanmıştır. Tek bir nesnenin bulunduğu görüntülerde bu oran %100 dür. Ayrıca çekilen görüntülerin tamamına yakınında çok büyük başarı oranıyla görüntüdeki nesnenin şişe olduğu tespit edilmiştir. Bu sonuçlar incelenerek nesne tespitini belirleyen en önemli etkenlerin nesnenin yapısı, görüntünün çekildiği yerin yapısı, nesnenin boyutu ve görüntüde bulunan nesne sayıları olduğu görülmüştür. Bu etkenler değiştirilerek daha başarılı sonuçlar elde edilebilir. Tez konusu belirlerken ortaya konulan hedef olan ekosistem için zararlı pet ve cam şişelerin doğal ortamda tespiti, büyük başarı oranı ile gerçekleştirilmiştir. Görüntüler üzerinde gerçekleştirilen nesne tespit test sonuçları tablo 6.1'de ayrıntılı olarak verilmiştir.

**Tablo 6.1.** Nesne tespit test sonuçları

Görüntü Şekil No	Nesne Sayısı		Tespit Edilen Nesne Sayısı		Tespit Edilen Nesnelerin Ortalama Doğruluk Oranı %	Nesne Tespit Başarısı %
	Pet	Cam	Pet	Cam		
Şekil 6.21.a	1	0	1	0	79	100
Şekil 6.21.b	1	0	1	0	90	100
Şekil 6.21.c	1	0	1	0	80	100
Şekil 6.21.d	1	0	1	0	75	100
Şekil 6.21.e	1	0	1	0	67	100
Şekil 6.21.f	1	0	1	0	61	100
Şekil 6.22.a	0	1	0	0	90	100
Şekil 6.22.b	0	1	0	1	85	100
Şekil 6.22.c	0	1	0	1	58	100
Şekil 6.22.d	0	1	0	1	82	100
Şekil 6.22.e	0	1	0	1	52	100
Şekil 6.22.f	0	1	0	1	46	100
Şekil 6.23.a	2	0	2	0	69,50	100
Şekil 6.23.b	2	0	2	0	48	100
Şekil 6.23.c	2	0	1	0	86	50
Şekil 6.23.d	2	0	1	0	94	50
Şekil 6.23.e	2	0	2	0	54	100
Şekil 6.23.f	2	0	1	0	95	50
Şekil 6.24.a	0	2	0	2	36	100
Şekil 6.24.b	0	2	0	2	69	100
Şekil 6.24.c	0	2	0	2	63	100
Şekil 6.24.d	0	2	0	2	65	100
Şekil 6.24.e	0	2	0	2	62,50	100
Şekil 6.24.f	0	2	0	1	39	50
Şekil 6.25.a	3	0	2	0	55,50	66,66
Şekil 6.25.b	3	0	2	0	54	66,66
Şekil 6.25.c	3	0	2	0	85,50	66,66
Şekil 6.25.d	3	0	2	0	41,50	66,66
Şekil 6.25.e	3	0	1	0	66	33,33
Şekil 6.25.f	3	0	1	0	84	33,33
Şekil 6.26.a	0	3	0	2	54	66,66
Şekil 6.26.b	0	3	0	2	36	66,66
Şekil 6.26.c	1	2	0	2	43,50	66,66
Şekil 6.26.d	0	3	0	1	84	33,33
Şekil 6.26.e	1	2	0	1	83	33,33
Şekil 6.26.f	0	3	0	1	41	33,33
Şekil 6.27.a	4	0	3	0	44,66	75
Şekil 6.27.b	4	0	3	0	61	75
Şekil 6.27.c	4	0	2	0	59	50
Şekil 6.27.d	3	1	2	0	77,50	50
Şekil 6.27.e	2	2	0	2	59,50	50
Şekil 6.27.f	0	4	0	2	64,50	50
Şekil 6.28.a	2	3	0	3	60,33	60
Şekil 6.28.b	3	2	2	1	59	60
Şekil 6.28.c	0	5	0	2	61	40
Şekil 6.28.d	1	4	1	1	71,50	40
Şekil 6.28.e	5	4	1	3	49	44,44
Şekil 6.28.f	8	6	3	0	52,33	21,42
Şekil 6.29.a	1	1	1	1	57	100
Şekil 6.29.b	1	1	0	1	77	50
<b>Toplam</b>	<b>76</b>	<b>67</b>	<b>43</b>	<b>41</b>		

## 7. SONUÇLAR VE ÖNERİLER

Bu tez çalışmasında gerçek zamanlı nesne tespit aracı olan YOLOv3 algoritması kullanılarak ekosistemin korunması için çevreye birçok açıdan zarar veren pet ve cam şişelerin tespit işlemi gerçekleştirilmiş ve başarılı sonuçlar elde edilmiştir. Belirlenen coğrafi bölgede İHA ile çekilen görüntüler üzerinde yapılan tespitler YOLOv3 algoritmasının pet ve cam şişe tespitinde kullanılabilirliğini desteklemiştir. Yapılan değerlendirme sonuçlarında 50 adet görüntü dosyasında bulunan 143 pet ve cam şişe nesnesinin yaklaşık %60'ı tespit edilmiştir. Bu oranın yeni geliştiren algoritmalar kullanılarak gelecekte yapılan çalışmalarda daha yukarılara çıkarılabileceği öngörülmektedir.

Derin öğrenmede eğitim yapılabilmesi için gerekli hız ve kalite günümüz Grafik İşlemci birimlerinde mevcuttur. Bu yüzden ne kadar çok görüntü dosyası kullanılabilirse nesne tespit başarısı da o kadar artacaktır. Sosyal medyanın oldukça yaygın olarak kullanıldığı günümüzde bu görüntü dosyalarına ulaşım da kolaylaşmıştır. Bu nedenle istenilen sayıda görüntü dosyalarına kolaylıkla erişilebilmektedir. Özellikle nesne tespiti için hazırlanmış farklı kategorilerde hazır veri setlerine de internet üzerinden ulaşılabilir.

Tez çalışmasında kullanılan görüntüler Fırat Üniversitesi Teknik Bilimler Meslek Yüksekokulu kampüs alanında belirlenen coğrafi bölgede önceden belirlenen yükseklikten İHA kullanılarak çekilmiştir. Gelecekte farklı coğrafi bölgelerde farklı zemin yapılarında nesne tespitinin yapılacağı çalışmalarda bu tez çalışmasından elde edilen sonuçlardan faydalanılabilir. Ayrıca bu tez çalışması ilerleyen zamanlarda geliştirilerek, İHA ile gerçek zamanlı olarak nesne tespit uygulamaları gerçekleştirilebilir. Böylece ekosisteme zarar veren nesnelere daha hızlı tespit edilecek ve çevreye verdiği zararlar en aza indirgenecektir.

## KAYNAKLAR

- [1] Türkmen, B. A., (2020). Cam Ambalaj Üretimine Çevresel Sürdürülebilirliğinin Değerlendirilmesi, *Bilecik Şeyh Edebalı Üniversitesi Fen Bilimleri Dergisi*, 7(2), ss. 1026-1037.
- [2] Ekici, E. (2022). *Deep Learning Based Fruit And Vegetable Recognition For Android Pos Devices*, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi, Lisansüstü Eğitim Enstitüsü.
- [3] Kabadayı, A., Uysal, M. (2019). İnsansız Hava Aracı ile Elde Edilen Verilerden Binaların Tespiti, *Türkiye İnsansız Hava Araçları Dergisi*, 1(1) , ss. 8-14.
- [4] Yanık, A., Serdar Güzel, M., Yanık, M. ve Bostancı, E. (2021). Machine Learning Based Early Fire Detection System Using a Low-Cost Drone, *arXiv preprint*, arXiv:2101.09362.
- [5] Mutlu, F. (2018). *İnsansız Hava Aracı Kullanılarak Anadolu Yaban Koyunlarının Popülasyonunun Belirlenmesi ve Yoğunluk Haritalarının Üretilmesi*, Yüksek Lisans Tezi, Necmettin Erbakan Üniversitesi, Fen Bilimleri Enstitüsü.
- [6] Yakar, M., Mırdan, Ö. (2017). Tarihi Eserlerin İnsansız Hava Aracı ile Modellenmesinde Karşılaşılan Sorunlar, *Geomatik Dergisi*, 2(3), ss. 118-125.
- [7] Demiröz, H. İ. (2020). *Toprak Yüzey Pürüzlülüğünün Ölçülmesinde Uzaktan Algılamanın Kullanımı*, Yüksek Lisans Tezi, Ankara Üniversitesi, Fen Bilimleri Enstitüsü.
- [8] Körez, A., Barışçı, N. (2018). İnsansız Hava Aracı (İHA) Görüntülerindeki Yayıların Faster R-CNN Algoritması ile Otomatik Tespiti, *2018 2nd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*. IEEE.
- [9] Kim, B., Yuvaraj, N., Sri Pretha, K.R., Arun Pandian, R. (2021). Surface Crack Detection Using Deep Learning with Shallow CNN Architecture for Enhanced Computation, *Neural Computing and Applications*, 33.15, pp. 9289-9305.
- [10] Şanlıyüksel Yücel, D., Ali Yücel, D. (2017). Terk Edilmiş Kömür Ocaklarında Oluşan Maden Göllerinin Hidrokimyasal Özelliklerinin Belirlenmesi ve İnsansız Hava Aracı ile Üç Boyutlu Modellenmesi, *Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi*, 23.6, ss. 780-791.
- [11] Doğan, M. (2019). *Bitkilerde Örülen Hastalıkların Derin Öğrenme Yöntemleriyle Tespiti ve Sınıflandırılması*, Yüksek Lisans Tezi, Yalova Üniversitesi, Fen Bilimleri Enstitüsü.
- [12] Solak, S., Altınışık U. (2018). Görüntü İşleme Teknikleri ve Kümeleme Yöntemleri Kullanılarak Fındık Meyvesinin Tespit ve Sınıflandırılması, *Sakarya University Journal of Science*, 22, ss. 56-65.
- [13] Altun, M., Türker, M. (2020). Çok Yüksek Çözünürlüklü Renkli İHA Görüntülerinden Kentsel Alanlarda Araç Tespiti, *Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi*, 26.2, ss. 371-384.
- [14] Murat, S. (2021). *İnsansız Hava Aracı Görüntülerinden Derin Öğrenme Yöntemleriyle Nesne Tanıma*, Yüksek Lisans Tezi, Maltepe Üniversitesi, Lisansüstü Eğitim Enstitüsü.
- [15] Şen, M. (2018). *Afet Yönetimi İçin İHA Görüntülerinin Kullanımı ve Karapınar Obruğu Örneği*, Yüksek Lisans Tezi, Necmettin Erbakan Üniversitesi, Fen Bilimleri Enstitüsü.
- [16] Gerdan, D. (2020). *Görüntü İşleme Teknikleri Kullanılarak Bazı Meyvelerin Sınıflandırılması*, Yüksek Lisans Tezi, Ankara Üniversitesi, Fen Bilimleri Enstitüsü.
- [17] Öztürk, M. (2018). *Makine Öğrenmesi ve Görüntü İşleme Tekniklerini Kullanarak Drone ile Yaprak Sınıflandırma*, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü.
- [18] Murat Kaymak, A., Nevzat Örnek, M. ve Kahramanlı, H. (2019). Görüntü İşleme Teknolojilerinin Elma Bahçelerine Yönelik Kullanım Örneği, *Uluborlu Mesleki Bilimler Dergisi*, 2.1, ss. 17-26.
- [19] Burgaz, M. (2020). *Derin Öğrenme Algoritmaları Kullanarak İnsansız Hava Araçları İle Silah Tespiti*, Yüksek Lisans Tezi, Batman Üniversitesi, Fen Bilimleri Enstitüsü.
- [20] Bozkurt, H. (2020). *İnsansız Hava Aracı Görüntüleri Yardımıyla Zeytin Ağaçlarında Taç Genişliğinin Belirlenmesi Üzerine Bir Araştırma*, Yüksek Lisans Tezi, Çukurova Üniversitesi, Fen Bilimleri Enstitüsü.

- [21] Cihan Ceylan, M. ve Uysal, M. (2020). İnsansız Hava Aracı ile Elde Edilen Veriler Yardımıyla Ağaç Çıkarımı, *Türkiye Fotogrametri Dergisi*, 3.1, ss. 15-21.
- [22] Yekta Güzel, Y. (2021). *Derin Öğrenme ile Yeşil Alanların Çıkarımı*, Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü.
- [23] Albayrak, E., Yayla, R. ve Yüzgeç, U. (2021). Mask R-CNN ile İHA Görüntülerinden Araç Tespiti, *International Symposium of Scientific Research and Innovative Studies*, 3.1, ss. 326-335.
- [24] Ekiz, A. (2021). *A Study on Detection of Watermelon Fruits by Analysing Images Obtained With an Unmanned Aerial Vehicle (Uav)*, Doktora Tezi, Çukurova Üniversitesi, Doğa ve Uygulamalı Bilimler Enstitüsü.
- [25] Eyili, M. (2021). *Marul Bahçelerinde Görüntü İşleme Tekniği Kullanılarak Yabancı Ot Popülasyonlarının Tespiti*, Yüksek Lisans Tezi, Ondokuz Mayıs Üniversitesi, Lisansüstü Eğitim Enstitüsü.
- [26] Mücahit Arvas, M. (2022). *Derin Öğrenme Algoritmaları Kullanılarak Döner Kanatlı İnsansız Hava Aracı Görüntülerinden Haşhaş, Kenevir ve Tütün Bitkilerinin Tespiti*, Yüksek Lisans Tezi, Van Yüzüncü Yıl Üniversitesi, Fen Bilimleri Enstitüsü.
- [27] Kayan, A. ve Küçük, A. (2020). Plastik Kirliliğin Çevresel Zararları ve Çözüm Önerileri, *Ankara Hacı Bayram Veli Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, 22.2, ss. 403-427.
- [28] Atılgan Türkmen, B. (2020). Cam Ambalaj Üretiminin Çevresel Sürdürülebilirliğinin Değerlendirilmesi, *Bilecik Şeyh Edebali Üniversitesi Fen Bilimleri Dergisi*, 7.2, ss. 1026-1037.
- [29] Yüce, A.E. ve Kılıç, M. (2014). PVC ve PET Atıkların Seçimli Flotasyonu, *Çukurova Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi*, 29.2, ss. 79-94.
- [30] Karakaş, A. (2010). *Katı Atıkların Ekonomik Analizi (Sakarya Büyükşehir Belediyesi Örneği)*, Doktora Tezi, Sakarya Üniversitesi, Sosyal Bilimler Enstitüsü.
- [31] Mentеше, S. Ve Koca, S. (2021). Marmara Bölgesi'nde Katı Atık Durumunun Değerlendirilmesi, *Turkish Studies-Social Sciences*, 16.4, ss. 1533-1552.
- [32] Türkiye Cumhuriyeti Çevre, Şehircilik ve İklim Değişikliği Bakanlığı, Ambalaj Bülteni, (2022), <https://webdosya.csb.gov.tr/db/cygm/icerikler/ambalajbulten-2020-20220226105845.pdf>
- [33] Kahveci, M. ve Can, N. (2017). İnsansız Hava Araçları: Tarihçesi, Tanımı, Dünyada ve Türkiye'deki Yasal Durumu, *Selçuk Üniversitesi Mühendislik, Bilim ve Teknoloji Dergisi*, 5.4, ss. 511-535.
- [34] Pakkan, B. vermiş, M. (2010). İnsansız Hava Araçlarının Genetik Algoritma Yöntemiyle Çoklu Hedeflere Planlanması, *Journal of Aeronautics & Space Technologies/Havacılık ve Uzay Teknolojileri Dergisi*, 4.3, ss. 77-84.
- [35] Bilgehan Makineci, H. (2016). İnsansız Hava Araçları Lidar Etkileşimi, *Geomatik*, 1.1, ss.19-23.
- [36] Akgül, M., Demir, M. ve Emin Akay, A. (2016). İnsansız Hava Araçları ile Yüksek Hassasiyette Sayısal Yükseklik Modeli Üretimi ve Ormancılıkta Kullanım Olanakları, *Journal of the Faculty of Forestry Istanbul University*, 66.1, ss. 104-118.
- [37] Dà-Jiāng Innovations Science and Technology Co. Ltd, *DJI Mavic Air 2 Drone*, (2022), <https://www.dji.com/mavic-air-2?site=brandsite&from=nav>.
- [38] Başaran, G. ve Çağıl, G. (2021). Koruyucu Gözlük Kullanımının Görüntü İşleme Yöntemiyle Tespit Edilmesi, *El-Cezeri*, 9.1, ss. 86-95.
- [39] Çalışkan, D. (2022). *Derin Öğrenme Yöntemleri ile Şüpheli Davranış Tespiti*, Yüksek Lisans Tezi, Marmara Üniversitesi, Fen Bilimleri Enstitüsü.
- [40] Evren, E. (2020). *Otomotiv Sektöründe Saha Çalışanlarının Dijital Dönüşüm Kapsamında Güvenlik Yeleği Ve Baretlarının Yapay Sinir Ağı Yöntemi İle Algılanması*, Yüksek Lisans Tezi, Sakarya Üniversitesi, Fen Bilimleri Enstitüsü.
- [41] İnkaya, M. F. (2021). *Derin Öğrenme Algoritmaları Kullanarak Güvenli Çocuk Oyun Alanlarında Karşılaştırmalı Tehlikeli Nesne Tespiti Yapılması*, Yüksek Lisans Tezi, Bursa Teknik Üniversitesi, Fen Bilimleri Enstitüsü.

- [42] Çağlı, G. ve Yıldırım, B. (2020). Bir Montaj Parçasının Derin Öğrenme ve Görüntü İşleme ile Tespiti, *Journal of Intelligent Systems: Theory and Applications*, 3.2, ss. 31-37.
- [43] Thuan, D. (2021). *Evolution of Yolo Algorithm and Yolov5: The State-of-the-Art Object Detention Algorithm*, Master's Thesis, Oulu University of Applied Sciences.
- [44] Ketua, T. (2020). *Facial Expression Detection and Recognition System with Yolo Algorithm and Convolutional Neural Network*, Master's Thesis, Tarumanagara University.
- [45] Azimjonov, J. (2021). *Kavşak ve Karayollar için Gerçek Zamanlı Görü Tabanlı Trafik Akış Bilgisi Hesaplama Sistemlerinin Geliştirilmesi*, Doktora Tezi, Sakarya Üniversitesi, Fen Bilimleri Enstitüsü.
- [46] Nathanel Liunanda, C., Rostianingsih, S. ve Nathania Purbowo, A. (2020). Implementasi Algoritma YOLO pada Aplikasi Pendeteksi Senjata Tajam di Android, *Jurnal Infra*, 8.2, ss. 235-241.
- [47] Amwin, A. (2021). *Vehicle-Based Detection and Classification You Only Look Once (Yolo) Algorithm*, Master's Thesis, Indonesian Islamic University.
- [48] Küçük, K. (2021). *Automatic Diagnosis from Panoramic Dental X-Rays Using Deep Learning*, Yüksek Lisans Tezi, Akdeniz Üniversitesi, Fen Bilimleri Enstitüsü.
- [49] Mao, Q., Sun, H., Liu, Y. ve Jia, R. (2019). Mini-YOLOv3: Real-Time Object Detector for Embedded Applications, *Ieee Access*, 7, ss. 133529-133538.
- [50] Havuç, E., Alpak, Ş., Çakırel, G. ve Kadir Baran, M. (2021). Derin Öğrenme Vasıtasıyla Masa Tenisi Topu Takibi, *Avrupa Bilim ve Teknoloji Dergisi* 27, ss. 629-635.

# ÖZGEÇMİŞ

Serkan ÇELİK

---

[Redacted]

[Redacted] [Redacted]  
[Redacted] [Redacted]  
[Redacted] [Redacted]  
[Redacted] [Redacted]  
[Redacted] [Redacted]  
[Redacted] [Redacted]  
[Redacted] [Redacted]

---

[Redacted] [Redacted]  
[Redacted] [Redacted]

---

[Redacted] [Redacted]  
[Redacted] [Redacted]  
[Redacted] [Redacted]  
[Redacted] [Redacted]  
[Redacted] [Redacted]  
[Redacted] [Redacted]

---

[Redacted]  
[Redacted] [Redacted]  
[Redacted] [Redacted]

---

[Redacted] [Redacted]  
[Redacted] [Redacted]  
[Redacted] [Redacted]