

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

**İNSANSIZ HAVA ARAÇLARI VASITASIYLA
KARASAL GÖRÜNTÜLER ÜZERİNDEN NESNE TESPİTİ**

YÜKSEK LİSANS TEZİ

Selçuk ÇİFTÇİ

Bilgisayar ve Bilişim Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği Bilim Dalı

OCAK 2025

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

**İNSANSIZ HAVA ARAÇLARI VASITASIYLA
KARASAL GÖRÜNTÜLER ÜZERİNDEN NESNE TESPİTİ**

YÜKSEK LİSANS TEZİ

Selçuk ÇİFTÇİ

Bilgisayar ve Bilişim Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği Bilim Dalı

Tez Danışmanı: Dr.Öğr. Üyesi Fatih ÇALLI

OCAK 2025

Selçuk ÇİFTÇİ tarafından hazırlanan “İnsansız Hava Araçları Vasıtasıyla Karasal Görüntüler Üzerinden Nesne Tespiti” adlı tez çalışması 20.01.2025 tarihinde aşağıdaki jüri tarafından oy birliği ile Sakarya Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar ve Bilişim Mühendisliği Anabilim Dalı Bilgisayar Mühendisliği Bilim Dalı’nda Yüksek Lisans tezi olarak kabul edilmiştir.

Tez Jürisi

Jüri Başkanı :	Dr.Öğr. Üyesi Fatih ÇALLI (Danışman)
	Sakarya Üniversitesi
Jüri Üyesi :	Dr.Öğr. Üyesi Selman HIZAL
	Sakarya Uygulamalı Bilimler Üniversitesi
Jüri Üyesi :	Dr.Öğr. Üyesi Muhammed KOTAN
	Sakarya Üniversitesi



ETİK İLKE VE KURALLARA UYGUNLUK BEYANNAMESİ

Sakarya Üniversitesi Fen Bilimleri Enstitüsü Lisansüstü Eğitim-Öğretim Yönetmeliğine ve Yükseköğretim Kurumları Bilimsel Araştırma ve Yayın Etiği Yönergesine uygun olarak hazırlamış olduğum “İnsansız Hava Araçları Vasıtasıyla Karasal Görüntüler Üzerinden Nesne Tespiti” başlıklı tezin bana ait, özgün bir çalışma olduğunu; çalışmamın tüm aşamalarında yukarıda belirtilen yönetmelik ve yönergeye uygun davrandığımı, tezin içerdiği yenilik ve sonuçları başka bir yerden almadığımı, tezde kullandığım eserleri usulüne göre kaynak olarak gösterdiğimi, bu tezi başka bir bilim kuruluna akademik amaç ve unvan almak amacıyla vermediğimi ve 20.04.2016 tarihli Resmi Gazete’de yayımlanan Lisansüstü Eğitim ve Öğretim Yönetmeliğinin 9/2 ve 22/2 maddeleri gereğince Sakarya Üniversitesi’nin abonesi olduğu intihal yazılım programı kullanılarak Enstitü tarafından belirlenmiş ölçütlere uygun rapor alındığını, çalışmamla ilgili yaptığım bu beyana aykırı bir durumun ortaya çıkması halinde doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi beyan ederim.

(08/01/2025)

Selçuk ÇİFTÇİ





Eşime ve kızlarıma...



TEŞEKKÜR

Yüksek lisans eğitimim boyunca sabırla bana destek olan değerli danışman hocam Dr.Öğr. Üyesi Fatih ÇALLI'ya, yardımlarını esirgemeyen Prof.Dr. Devrim AKGÜN ve Prof.Dr. Eyyüp GÜLBANDILAR hocama, canım eşime ve güzeller güzeli kızlarıma...



Selçuk ÇİFTÇİ



İÇİNDEKİLER

Sayfa

ETİK İLKE VE KURALLARA UYGUNLUK BEYANNAMESİ	v
TEŞEKKÜR	ix
İÇİNDEKİLER	xi
KISALTMALAR	xiii
TABLO LİSTESİ	xv
ŞEKİL LİSTESİ	xvii
ÖZET	xix
SUMMARY	xxi
1. GİRİŞ	1
1.1. Araştırmanın Gerekçesi	2
1.2. Tezin Amaç ve Hedefleri	3
1.3. Tezin Kapsamı	3
2. LİTERATÜR TARAMASI	5
2.1. Yapay Zekâ (YZ)	5
2.2. Yapay Sinir Ağları (YSA)	5
2.3. Makine Öğrenmesi	7
2.4. Derin Öğrenme	8
2.4.1. Derin Öğrenme Yöntemleri	10
2.4.1.1. Evrişimsel Sinir Ağları (CNN)	10
2.4.1.2. Tekrarlayan Sinir Ağları (RNN)	12
2.4.1.3. Transformer Modelleri	13
2.4.1.4. Üretici Çekişmeli Ağlar (GAN)	13
2.5. Nesne Tespit Modellerinin İncelenmesi	13
2.5.1. R-CNN (Region-based CNN)	14
2.5.2. Fast R-CNN	15
2.5.3. Faster R-CNN	15
2.5.4. Mask R-CNN	16
2.5.5. You Only Look Once (YOLO)	17
2.6. İHA Görüntülerinin Analizi Kapsamında Yapılan Çalışmaların İncelenmesi	20
3. MATERYAL VE YÖNTEM	27
3.1. Veri Seti Toplama	27
3.2. Veri Seti Ön İşleme	27
3.3. Veri Seti Etiketleme	29
3.4. YOLOv8 ile Veri Setinin Eğitilmesi	31
3.5. Veri Setinin Artırılması	37
3.6. YOLOv8 ile Artırılan Veri Setinin Eğitilmesi	40
4. SONUÇ VE ÖNERİLER	45
4.1. Sonuçların Değerlendirilmesi	45
4.2. Öneriler	46
KAYNAKLAR	47
ÖZGEÇMİŞ	51



KISALTMALAR

C2F	: Cross Stage Partial with Fusion
CNN	: Convolutional Neural Network (Evrişimsel Sinir Ağları)
GAN	: Generative Adversarial Network (Üretici Çekişmeli Ağlar)
GFLOP	: Giga Floating Point Operations Per Second
GPU	: Graphics Processing Unit (Grafik İşlemci Birimi)
GRU	: Gated Recurrent Unit
İHA	: İnsansız Hava Aracı
IoU	: Intersection over Union
LSTM	: Long Short-Term Memory (Uzun Kısa Süreli Bellek)
mAP	: Mean Average Precision (Ortalama Doğruluk Metriği)
NLP	: Natural Language Processing (Doğal Dil İşleme)
NMS	: None Maximum Suppression
R-CNN	: Region-based Convolutional Neural Network
RNN	: Recurrent Neural Network (Tekrarlayan Sinir Ağları)
RoI	: Regions of Interest
RPN	: Region Proposal Network
SPPF	: Spatial Pyramid Pooling - Fast
SSD	: Single Shot MultiBox Detector
SVM	: Support Vector Machines
YOLO	: You Only Look Once
YSA	: Yapay Sinir Ağı
YZ	: Yapay Zekâ



TABLO LİSTESİ

Sayfa

Tablo 2.1. Makine Öğrenmesi ile Derin Öğrenme Arasındaki Farklar.....	9
Tablo 2.2. R-CNN Modellerinin Özellikleri.	16
Tablo 3.1. YOLOv8 Katman Mimarisi.	32
Tablo 3.2. Modelin Performans Değerlendirmesi.	33
Tablo 3.3. YOLOv8 Veri Artırma Yöntemleri.....	38
Tablo 3.4. Artırılmış Veri Seti Modelinin Performans Değerlendirmesi.	41



ŞEKİL LİSTESİ

Sayfa

Şekil 1.1. Bayraktar TB2, Anka, Akıncı İHA [3], [4].	1
Şekil 2.1. Yapay Sinir Ağı Katmanları.	6
Şekil 2.2. Yapay Zekâ - Makine Öğrenmesi - Derin Öğrenme.	8
Şekil 2.3. Makine Öğrenmesi - Derin Öğrenme.	9
Şekil 2.4. Evrişimsel Sinir Ağları (CNN).	10
Şekil 2.5. Evrişim Katmanı.	11
Şekil 2.6. Maksimum ve Ortalama Havuzlama.	11
Şekil 2.7. Düzleştirme İşlemi.	12
Şekil 2.8. Tam Bağlantılı Katman.	12
Şekil 2.9. R-CNN Modeli Algoritması [21].	15
Şekil 2.10. YOLO Versiyonları Performans Karşılaştırması [26].	18
Şekil 2.11. YOLOv8 Mimarisi [27].	19
Şekil 3.1. CapCut Programı ile Karelerin Alınması.	28
Şekil 3.2. Fotor Uygulaması ile Fotoğrafların Düzenlenmesi.	28
Şekil 3.3. Veri Seti Örnek Görseller.	29
Şekil 3.4. labelImg 1.8.1 ile Veri Etiketleme.	30
Şekil 3.5. Etiketlenmiş Veri Seti Örnek Görseller.	30
Şekil 3.6. Google Colaboratory.	31
Şekil 3.7. F1 Skoru-Güven Seviyesi Eğrisi.	35
Şekil 3.8. Eğitim ve Doğrulama Süreçlerine Ait Kayıp Değerleri.	35
Şekil 3.9. Karışıklık Matrisi (Confusion Matrix).	36
Şekil 3.10. Modelimizin Örnek Tahminleri.	37
Şekil 3.11. Artırılmış Veri Seti Örnek Görseller.	40
Şekil 3.12. F1 Skoru-Güven Seviyesi Eğrisi.	41
Şekil 3.13. Eğitim ve Doğrulama Süreçlerine Ait Kayıp Değerleri.	42
Şekil 3.14. Artırılmış Veri Seti Modelinin Karışıklık Matrisi.	43
Şekil 3.15. Artırılmış Veri Seti Modeli Örnek Tahminleri.	43



İNSANSIZ HAVA ARAÇLARI VASITASIYLA KARASAL GÖRÜNTÜLER ÜZERİNDEN NESNE TESPİTİ

ÖZET

Son yıllarda yapay zekâ teknolojilerindeki hızlı ilerlemeler, insan yaşamını önemli ölçüde etkilerken, özellikle askeri alanda kullanılan İnsansız Hava Araçları (İHA) sistemlerinin ortaya çıkışını ve gelişimini de beraberinde getirmiştir. Bu tez çalışmasında, bir yapay zekâ uygulaması olan derin öğrenme yöntemlerinden faydalanılarak İHA'lar tarafından elde edilen görüntülerden nesne tespiti üzerine odaklanılmıştır. Çalışma kapsamında, günümüz teknolojisinin ulaştığı iki önemli alanın (İHA sistemleri ve yapay zekâ) entegrasyonu ele alınmıştır.

Tez çalışmasının en dikkat çekici yönü, kullanılan veri setinin oluşturulma sürecidir. Veri seti, açık kaynaklı platformlar olan Youtube, Twitter ve Telegram üzerinden detaylı bir tarama yapılarak elde edilen gerçek hareket ortamı görüntülerinden oluşturulmuştur. Bu özelliğiyle, geliştirilen sistemin gerçek operasyonel şartlara kolaylıkla uyarlanabilir olması sağlanmıştır. Çalışma, öncelikli olarak askeri kullanım amacı taşımakla birlikte, sivil uygulamalarda da kullanılabilir bir karar destek mekanizması sunmaktadır. Ek olarak, ilgili sistemler ile iletişim sağlanamadığı durumlarda bile, sistemin önceden belirlenmiş parametreler doğrultusunda otonom hareket edebilmesi sağlanabilecektir. Bu yönüyle tez çalışması, gelecekte yapılacak araştırmalar için bir yol haritası niteliği taşımaktadır.

Çalışmada görüntü işleme alanında başarısı kanıtlanmış derin öğrenme yöntemlerinden biri olan Evrişimsel Sinir Ağları (Convolutional Neural Networks, CNN) tabanlı You Look Only Once (YOLO) modeli kullanılmıştır. Tez çalışması kapsamında, YOLOv8 versiyonu kullanılmış olup en çok bilinen ve uygulanan versiyon olarak öne çıkmaktadır. Araştırmada, dört sınıfa (kamyon, tank, hava savunma sistemleri, top bataryası) ayrılmış bir veri seti hazırlanmıştır. Bu veri seti, %80 eğitim verisi, %10 doğrulama verisi ve %10 test verisi olarak ayrıştırılmış ve YOLOv8 modeli kullanılarak analiz edilmiştir. Çalışmanın ilerleyen aşamalarında, veri artırma yöntemleri uygulanarak veri seti genişletilmiş ve bu genişletilmiş veri setiyle elde edilen sonuçlar başlangıçtaki analizlerle karşılaştırılmıştır.

İlk model eğitimi, precision (doğruluk) %70.6, recall (duyarlılık) %62.1 ve mAP@50 %74.7 değerleriyle tatmin edici bir başlangıç performansı sunmuştur. Ancak mAP@50-95 değeri %57.1 olarak nispeten düşük kalmıştır. Artırılmış veri seti ile yapılan ikinci eğitimde ise tüm bu metriklerde kayda değer iyileşmeler sağlanmıştır. Precision %86.5, recall %85.5 ve mAP@50 %96.6 gibi oldukça yüksek değerlere ulaşılmıştır. Daha zorlu değerlendirme ölçütü olan mAP@50-95 değeri ise %62.9 ile ilk eğitime göre önemli bir artış göstermiştir.

Elde edilen bulgular, sahadan doğrudan toplanan verilerin artırılmasının, sistemin başarı oranı ve tanıyabileceği sınıf sayısının genişletilmesi için kritik önem taşıdığını göstermektedir. Halihazırda kamyon, tank, hava savunma sistemleri ve top bataryası sınıfları üzerinde çalışan sistemin, sahadan daha fazla görüntü elde edilmesi

durumunda, diğ er nesne sınıflarını da y uksek bařarı oranıyla tanıyabileceđi anlařılmıştır. Bu bađlamda, sistemin nesne tespiti ve sınıflandırma bařarısının artırılması i in veri toplama s ure lerinin sistematik bir řekilde geniřletilmesi gerektiđi sonucuna ulařılmıştır.



OBJECT DETECTION ON TERRESTRIAL IMAGERY USING UNMANNED AERIAL VEHICLES

SUMMARY

In recent years, rapid advances in artificial intelligence technologies have significantly impacted human life and brought about the emergence and development of Unmanned Aerial Vehicles (UAV) systems, especially in the military field. This thesis focuses on object detection from images obtained by UAVs by utilizing deep learning methods, an artificial intelligence application. Within the scope of the study, the integration of two important areas that today's technology has reached (UAV systems and artificial intelligence) is discussed.

Although the study is primarily intended for military use, it also provides a decision support mechanism that can be used in civilian applications. In addition, the system will be able to act autonomously in line with predetermined parameters, even in cases where communication with the relevant systems is not possible. In this respect, the thesis is a roadmap for future research.

The most remarkable aspect of the thesis is the process of creating the data set used. Data collection and dataset creation is one of the cornerstones of machine and deep learning projects. This process involves collecting, organizing and processing the data needed to train, validate and test the model. A good dataset plays a vital role in increasing the accuracy and generalization ability of the model. The dataset is composed of real operational environment images obtained through a detailed search on open-source platforms such as Youtube, Twitter and Telegram. With this feature, the developed system is easily adaptable to real operational conditions.

Data preprocessing is a set of methods applied to bring data into a suitable form for analysis or modeling purposes or to eliminate potential obstacles. The videos collected in a file that can be used in a file are divided into frames with the CapCut program, and the images that can be used in the data set are saved as photo frames in jpg format, and the noise and unnecessary images on the photos are cleaned by removing them with the Fotor application without disrupting the integrity of the data.

For deep learning models that work with image classification, each image must be labeled with a class label. Obtaining accurate results from the studies depends on the correct and meticulous labeling of the data. In the dataset labeling step of our study, the bounding box method was applied to all data using the labelImg 1.8.1 program.

The Colaboratory system, a Google service, was used for all training and detection processes with the prepared dataset. Google Colab is a popular cloud-based tool for machine learning, deep learning, data analytics and other Python-based projects. Google Colab works as an online Jupyter Notebook and allows users to write code, run it, create visualizations and annotate it. With this service, the duration of the results obtained can be shortened by selecting high-performance hardware for the GPU feature, which is an important tool especially in visual-based training. With the free

version, trainings can be made by allowing up to 12 hours of connection, and high-performance hardware such as Tesla T4 GPU can be used.

In the training phase of the study, the You Look Only Once (YOLO) model based on Convolutional Neural Networks (CNN), one of the proven deep learning methods in image processing, was used. Within the scope of the thesis study, YOLOv8 version was used and it stands out as the most widely known and applied version. In the research, a dataset divided into four classes (truck, tank, air defense systems, artillery battery) was prepared. This dataset was divided into 80% training data, 10% validation data and 10% test data and analyzed using the YOLOv8 model.

In the later stages of the study, the data set was expanded by applying data augmentation methods and the results obtained with this expanded data set were compared with the initial analysis. Data augmentation is a technique used to improve the performance of machine learning and deep learning models. This method involves creating new data samples by manipulating the existing data set or applying various transformations. Data augmentation methods play an important role in reducing the need for data sets for deep learning models and avoiding problems such as overfitting. Data augmentation aims to reduce overfitting by artificially diversifying the data set so that the model generalizes better to new data, to overcome the problem of insufficient data as it can be difficult to obtain enough labeled data in most applications, and to increase the learning capacity and performance of the model with a more diverse and larger data set. The techniques used on the images can be summarized as rotation and mirroring, which involves rotating the images at certain angles or mirroring them on the horizontal/vertical axis; brightness and contrast adjustment, which involves changing the brightness or contrast levels of the images; cutting, which involves cutting or removing certain regions of the images; and noise addition, which involves adding random noise pixels to the images to create more realistic data.

The object detection performance of the applied models was evaluated with metrics such as precision, recall, mAP@50 (mean average precision) and mAP@50-95. mAP is a measure summarizing the success of the model in making accurate object detection. Precision is defined as the ratio of correctly detected objects to total detected objects, while mAP is the average value of this accuracy. mAP@50 refers to the case where the model uses a threshold of 50% to detect objects, i.e. the “Intersection over Union” (IoU) threshold of 50% is used to determine whether the model detected an object correctly. mAP@50-95 refers to the calculation of mAP for cases where the IoU threshold ranges from 50% to 95%. This is an assessment of how accurately the model detects with more stringent thresholds (IoU 50%-95%). That is, the accuracy of the model is measured at different levels of IoU thresholds ranging from 50% to 95%.

In the initial model training, precision was 70.6%, recall was 62.1% and mAP@50 was 74.7%, giving a satisfactory initial performance. However, the mAP@50-95 value remained relatively low at 57.1%.

The second model training with the augmented dataset resulted in significant improvements in all these metrics. Precision 86.5%, recall 85.5% and mAP@50% 96.6%, which are quite high values. The more challenging evaluation metric, mAP@50-95, showed a significant increase compared to the first training with 62.9%.

The findings show that increasing the data collected directly from the field is critical for the success rate of the system and for expanding the number of classes it can recognize. The system, which currently works on truck, tank, air defense systems and

artillery battery classes, will be able to recognize other object classes with a high success rate if more images are obtained from the field. In this context, it was concluded that the data collection processes should be systematically expanded in order to increase the object detection and classification success of the system.





1. GİRİŞ

Günümüzde teknoloji, insan hayatını şekillendiren en önemli faktörlerden biri haline gelmiştir. Bu teknolojik ilerlemelerin bir sonucu olarak, İnsansız Hava Araçları (İHA) gibi otonom sistemler, bir dönüm noktası oluşturmuş ve birçok sektörde devrim yaratmıştır. Askeri ve sivil hayattaki kullanım alanlarıyla günümüz teknolojisinin vazgeçilmez parçalarından biri haline gelmiştir.

Ancak İHA'ların kökenleri, sanıldığından çok daha eskilere dayanmaktadır. İlk İHA prototipleri, 19. yüzyılda uzaktan kumandalı uçan hedefler olarak ortaya çıkmıştır. Bu deneyler, o dönemde daha çok askeri amaçlarla gerçekleştirilmiştir. İkinci Dünya Savaşı, İHA teknolojilerinde önemli bir dönüm noktası olmuştur. Almanya ve diğer ülkeler, savaş sırasında füze ve bomba taşıyabilen İHA'lar geliştirerek hava savunma sistemlerini test etmek ve düşman hedeflerine saldırmak için kullanmışlardır. Soğuk savaş döneminde ise İHA'lar, özellikle casusluk ve keşif görevlerinde kullanılmaya başlanmıştır. Bu dönemde, yüksek irtifada uzun süre kalabilen ve düşman topraklarının derinliklerine sızabilen İHA'lar geliştirilmiştir [1], [2]. Şekil 1.1'de ülkemizde son dönemde geliştirilen ve en yaygın kullanılan İHA'lara örnekler verilmiştir.



Şekil 1.1. Bayraktar TB2, Anka, Akıncı İHA [3], [4].

Günümüzde İHA'lar, yüksek çözünürlüklü kameralar, radarlar, mühimmatlar ve diğer sensörlerle donatılmış olarak kullanılmaktadır. Bu sensörler vasıtasıyla İHA'lar atış, veri toplama, izleme ve keşif gibi önemli görevlerde kullanılmaktadır. İHA'lar tarafından toplanan bu veriler, genellikle büyük, karmaşık ve dinamik veri setlerini temsil eder. Bu verilerin etkili bir şekilde analizi ile anlamlı bilgilerin çıkarılmasını sağlamak büyük öneme sahiptir. Derin öğrenme yöntemlerinin (deep learning) gelişmesiyle, İHA'lardan elde edilen verilerin analizi ve nesne tespiti gibi görevlerde kullanılabilirliği artmıştır. İşte bu noktada derin öğrenme yöntemleri, özellikle de evrimsel sinir ağları (CNN) ve tekrarlayan sinir ağları (RNN) gibi yöntemler, İHA'lar tarafından sağlanan verilerin analizi için güçlü araçlar olarak ortaya çıkmaktadır.

Evrimsel sinir ağları, görüntü verilerinde nesne tespiti, sınıflandırma ve alan tanıma gibi görevlerde başarıyla kullanılmaktadır. İHA'lar tarafından elde edilen görüntü verileri, CNN yöntemini kullanan modeller ile eğitilerek nesne tespiti ve izleme gibi görevlerde kullanılabilir. Tekrarlayan sinir ağları ise zaman serileri gibi sıralı verilerin analizi için etkili bir şekilde kullanılabilir. Bu yöntem, İHA'lar tarafından sağlanan verilerin zaman içindeki değişikliklerini anlamak için idealdir.

İHA'lar, ulaşılması zor veya tehlikeli alanlarda görev yapabilirken, derin öğrenme yöntemleri büyük veri setlerinde karmaşık durumları tanıyabilir, böylece nesne tespiti ve sınıflandırma gibi görevlerde yüksek doğruluk oranlarına ulaşabilir. İHA'lar ve derin öğrenme yöntemlerinin bu şekilde entegre edilmesi, birçok alanda faydalar sağlayabilir. İHA'lar tarafından toplanan görüntü verileri, CNN yöntemini kullanan modeller ile analiz edilerek istenilen tüm faktörlerin izlenmesinde kullanılabilir.

Bu çalışma, İHA'lar tarafından sağlanan görüntülerin derin öğrenme yöntemleriyle nasıl analiz edilebileceğini ve bu analizlerin askeri alanda nasıl kullanılabileceğini ele alacaktır.

1.1. Araştırmanın Gerekçesi

İHA'lardan elde edilen görüntüler, derin öğrenme yöntemleri ile işlenerek birçok alanda kullanılabilir. Çalışmamızda İHA'ların ilk çıkış noktası olan askeri versiyonlarından elde edilen görüntüler kullanılarak harekât alanında ihtiyaç olduğu durumlarda karar verici mekanizmalara yardımcı olacak değerlendirmenin otomatik olarak yapılması veya karar vericiden bağımsız (İHA ile iletişim olmadığı) nesneleri

tanıyarak önceden belirlenmiş nesnelere İHA'nın yönelmesini sağlayacak sistemlerin geliştirilmesini sağlamaktır.

1.2. Tezin Amaç ve Hedefleri

Bu tezin temel amacı, araştırılan süreçte daha önce hiç yapılmadığı tespit edilen ve kullanıcılara açık ortamda sağlanan İHA görüntülerinin ayrıntılı bir şekilde işlenerek hazır veri seti haline getirilmesi ve derin öğrenme yöntemleri ile harekât ortamında karşılaşılabilecek hedefler için hedef tespitine uygun olup olmadığının araştırılması hedeflenmektedir. Bu kapsamda, günümüzün en çok kullanılan nesne tanıma modellerinden biri olan You Only Look Once (YOLO) modeli üzerinde çalışmalar yapılacak olup çeşitli denemeler ile nesne tespiti uygulamasında performansı değerlendirilecektir.

1.3. Tezin Kapsamı

Çalışma, harekât alanında karşılaşılabilecek ve hedef olabilecek ikmal kamyonlarını, tank ve benzeri zırhlı sistemlerini, araç üzeri monteli hava savunma sistemlerini ve araç arkası çekilerek taşınan ve kurulan top bataryası sistemlerini çeşitli denemeler ile derin öğrenme uygulaması kullanılarak tespit edilmesini kapsamaktadır.



2. LİTERATÜR TARAMASI

2.1. Yapay Zekâ (YZ)

Yapay Zekâ (YZ), makine öğrenmesi kullanıp kullanmamasına bakılmaksızın, tahmin veya karar alma süreçlerini gerçekleştiren teknolojilerin genel adını ifade etmektedir. YZ, makine öğrenmesi ya da derin öğrenme yöntemlerine başvurulmadan da işlev gösterebilen algoritmalar olarak değerlendirilebilir. Makine öğrenmesinin geliştirilmesinden önce, YZ sistemleri, mantıksal ve matematiksel işlemlerin tamamının programcılar tarafından açık bir şekilde kodlandığı bir yaklaşıma dayanmaktaydı [5]. Günümüzde ise, YZ bilgisayarların ve diğer makinelerin, insanlar tarafından gerçekleştirilen, zekâ gerektiren görevleri yerine getirmesini sağlayan bir teknoloji ve bilim dalıdır. YZ'nin temel amacı, insan zekasının belirli yönlerini (öğrenme, problem çözme, akıl yürütme, dil anlama, görsel algı gibi) taklit eden sistemler geliştirmektir. Bu sistemler, verilerden öğrenebilir, mantıklı çıkarımlar yapabilir ve insan benzeri kararlar alabilir.

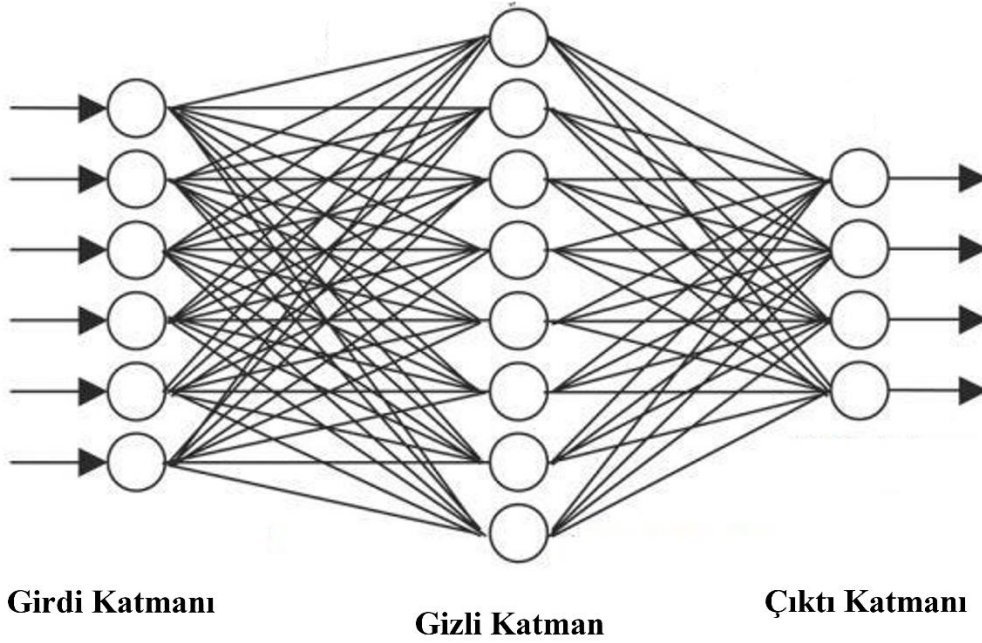
2.2. Yapay Sinir Ağları (YSA)

Yapay Sinir Ağları (YSA), insan beyninin bilgi işleme yöntemlerinden esinlenerek geliştirilmiş bir yapay zekâ modelidir. YSA, temel olarak çok sayıda işlem biriminden (nöron) oluşan bir ağ yapısı üzerinden bilgi öğrenme, modelleme ve genelleme yeteneklerini sağlar. Bu yöntemler, özellikle karmaşık ve doğrusal olmayan problemlerin çözümünde etkili bir şekilde kullanılmaktadır [6]. Günümüzde, görüntü işleme, doğal dil işleme, biyomedikal veri analizi ve finans gibi birçok alanda yaygın bir şekilde kullanılmaktadır.

YSA'lar, bir giriş katmanı, bir veya daha fazla gizli katman ve bir çıkış katmanından oluşur. Her bir katman, kendisine özgü ağırlıklar ve aktivasyon fonksiyonları kullanarak bilgiyi işler [7]. Bu yapı, girdileri işleyerek karmaşık bir çıktı elde etmeye olanak tanır.

Yapay sinir ağlarının genel yapısı, Şekil 2.1'de gösterilmiş olup, temel olarak girdi katmanı, ara (gizli) katmanlar ve çıktı katmanı olmak üzere üç ana katmandan oluşur.

Her bir katman, yapay sinir ağıının farklı işlevlerini yerine getiren ve ağıın genel performansını etkileyen bir yapıya sahiptir.



Şekil 2.1. Yapay Sinir Ağı Katmanları.

Giriş katmanı, yapay sinir ağı ile dış dünya arasındaki bağlantıyı sağlayan birimdir. Bu katman, dış dünyadan gelen verilerin ilk olarak işlendiği bölümdür. Giriş katmanında, gelen veri miktarına eşdeğer sayıda nöron yer alır. Ancak, bu nöronlar genellikle veriler üzerinde herhangi bir işlem yapmaksızın bilgiyi doğrudan alt katmanlara iletir. Bu nedenle, giriş katmanı sadece bir veri taşıyıcı işlevi görür ve ağıın hesaplama sürecine aktif bir katkıda bulunmaz.

Ara (gizli) katmanlar, giriş katmanından alınan bilgilerin işlendiği ve ağıın öğrenme sürecinin gerçekleştiği birimlerdir. Bu katmanlar, yapay sinir ağlarının karmaşık verileri öğrenme ve genelleştirme yeteneğini belirleyen kritik bölümlerdir. Ara katmanların sayısı ve bu katmanlardaki nöron sayısı, ağdan ağa farklılık gösterebilir. Bazı yapay sinir ağlarında ara katman bulunmazken, daha karmaşık yapılara sahip ağlarda birden fazla ara katman bulunabilir. Ara katmanlardaki nöron sayısı, giriş ve çıkış katmanlarındaki nöron sayısından bağımsızdır ve bu sayı katmanlar arasında değişkenlik gösterebilir. Ara katmanların ve bu katmanlardaki nöron sayısının artması, ağıın hesaplama karmaşıklığını ve işlem süresini artırır. Ancak, bu durum aynı

zamanda ağıın daha karmaşık problemlerin çözümünde etkin bir şekilde kullanılabilmesini sağlar.

Çıkış katmanı, ara katmanlardan aldığı bilgileri işleyerek yapay sinir ağının çıktılarını üreten bölümdür. Bu katmandan elde edilen çıktı, doğrudan dış dünyaya aktarılır. Geri beslemeli ağlarda ise bu çıktı, ağıın öğrenme sürecinin bir parçası olarak yeni ağırlıkların hesaplanmasında kullanılır. Çıkış katmanı, yapay sinir ağının temel amacını yerine getiren ve kullanıcının ihtiyaç duyduğu sonuçları sağlayan birim olarak işlev görür.

YSA'ların en büyük avantajı, karmaşık veri setlerindeki doğrusal olmayan ilişkileri öğrenme kapasitesidir. Ancak, büyük veri setlerine ve yüksek işlem gücüne ihtiyaç duymaları, aşırı öğrenme riski gibi sınırlamaları da bulunmaktadır [8].

Derin öğrenme mimarileri (örneğin, Convolutional Neural Networks ve Recurrent Neural Networks), YSA'nın bir alt kümesi olarak, daha büyük ve karmaşık problemlerin çözümünde ön plana çıkmaktadır. Ayrıca, kuantum hesaplama ile birleştiğinde, YSA'ların performansının daha da artacağı öngörülmektedir.

Yapay Sinir Ağları, modern yapay zekâ ve makine öğrenimi uygulamalarının temel taşlarından biridir. Gelişen teknolojilerle birlikte, bu modellerin daha verimli, ölçeklenebilir ve insan kararlarına daha yakın sonuçlar üretebileceği bir geleceğe doğru ilerlenmektedir

2.3. Makine Öğrenmesi

Teknolojinin gelişmesiyle birlikte bilgisayarların deneyimlerden öğrenerek, görevleri otomatik olarak yerine getirmesini sağlayan, bir yapay zekâ (YZ) dalı olan makine öğrenmesi ortaya çıkmıştır. Bu alan, istatistik, matematik ve bilgisayar bilimlerine dayanan disiplinler arası bir süreçle gelişmiştir. Makine öğrenmesinin temelleri, 20. yüzyılda ortaya atılmıştır. Alan Turing, 1950'de yayımladığı "Computing Machinery and Intelligence" adlı makalesinde, makinelerin öğrenip öğrenemeyeceğini tartıştı ve bu alana ilham verdi [9]. 1958'de Frank Rosenblatt, Perceptron algoritmasını geliştirerek yapay sinir ağlarının temelini attı [10]. Ancak bu dönemin bilgisayarlarının sınırlı işlem gücü ve öğrenme algoritmalarının eksiklikleri nedeniyle ilerleme sınırlıydı. Bu nedenle bilgisayar sistemleri gelişene kadar bu alan bir duraklama dönemine sahne oldu.

1990'larda internetin yaygınlaşması, veri miktarını artırdı ve büyük veri çağını başlattı. Bu dönemde makine öğrenmesi, veri madenciliği, spam filtreleme ve öneri sistemleri gibi alanlar yaygınlaşmaya başladı [11].

2.4. Derin Öğrenme

Derin öğrenme ise, makine öğrenmesinin bir alt dalıdır ve temel olarak yapay sinir ağları kullanılarak daha karmaşık ve soyut ilişkileri öğrenmeye odaklanır. Bu sistem, insan beyninin bilgi işleme modeline benzer bir şekilde çalışır. Derin öğrenme, çok sayıda katmandan oluşan yapay sinir ağlarını kullanarak özellikleri otomatik olarak öğrenir. Yapay Zekâ, Makine Öğrenmesi ve Derin Öğrenmenin birbirlerini kapsamaları Şekil 2.2'de gösterilmiştir.



Şekil 2.2. Yapay Zekâ - Makine Öğrenmesi - Derin Öğrenme.

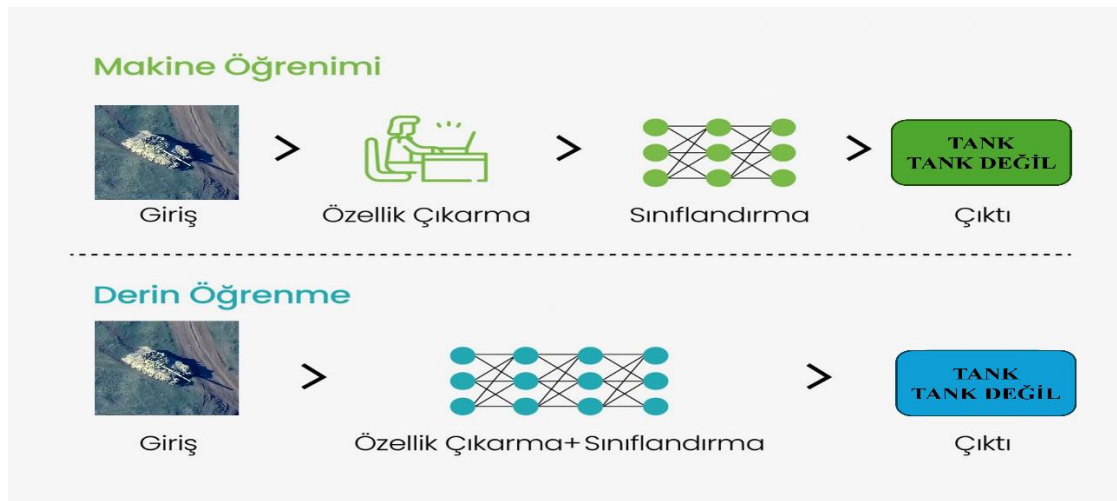
2000'li yıllarda, işlem gücündeki artış ve büyük veri setlerinin erişilebilir hale gelmesi ile makine öğrenmesinin alt dalı olan derin öğrenme gündeme geldi. Geoffrey Hinton ve ekibi, 2006'da derin sinir ağlarını eğitmek için katman katman ön eğitim (layer-wise pretraining) yöntemini önerdi [12]. Ayrıca, Grafik işlem birimleri (GPU)'ların kullanımı bu dönemde hız kazandı. Grafik işlem birimleri, büyük miktarda veriyle karmaşık ağların eğitilmesini hızlandırdı.

2012 yılı, derin öğrenmenin dönüm noktalarından biridir. Geoffrey Hinton'ın ekibi tarafından geliştirilen AlexNet, ImageNet Görüntü Tanıma Yarışması'nda ezici bir üstünlükle birinci oldu [13]. AlexNet, evrimsel sinir ağlarını (CNN) kullanarak görüntü işleme alanında çığır açtı. Bu başarı, Doğal Dil İşleme (NLP), Ses Tanıma, Otonom Araçlar gibi alanlara ilham verdi.

Tablo 2.1. Makine Öğrenmesi ile Derin Öğrenme Arasındaki Farklar.

Özellik	Makine Öğrenmesi	Derin Öğrenme
Kapsam	Geniş bir disiplin	Makine öğrenmesinin bir alt dalı
Veri Miktarı	Daha az veri ile çalışabilir.	Büyük veri setlerine ihtiyaç duyar.
Model Karmaşıklığı	Daha basit algoritmalar.	Çok katmanlı ve karmaşık modeller.
Özellik Seçimi	Manuel olarak yapılabilir.	Özellikleri otomatik öğrenir.
Hesaplama Gücü	Daha az işlem gücü yeterlidir.	Yüksek işlem gücü (GPU/TPU) gereklidir.

Tablo 2.1’ de görüleceği üzere makine öğrenmesi, derin öğrenmeyi de kapsayan geniş bir alan olup, istatistiksel yöntemlerden sinir ağlarına kadar birçok farklı yöntemi içerir. Derin öğrenme ise, makine öğrenmesinin daha özel bir alt dalı olarak, karmaşık veri ilişkilerini öğrenmek için geliştirilmiş ileri düzey bir yöntemdir. Şekil 2.3’ de derin öğrenmenin, makine öğrenmesinin başlangıcı değil, daha gelişmiş bir uygulaması olduğu gösterilmiştir.



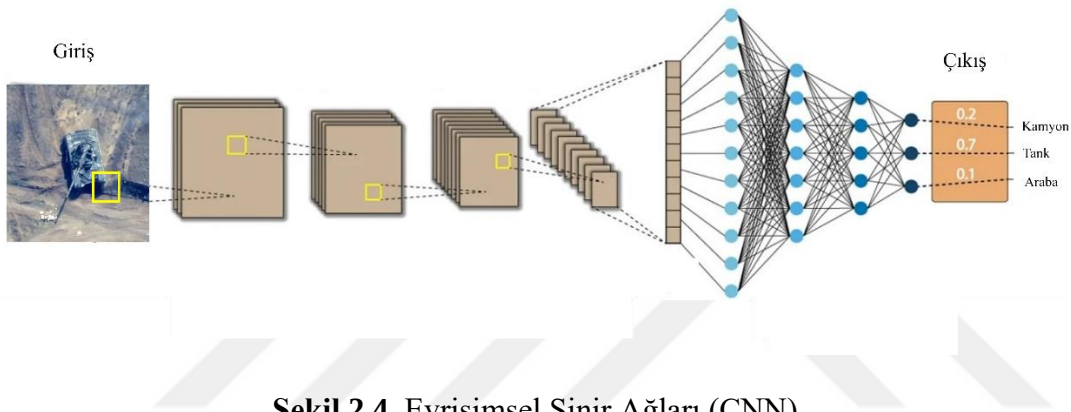
Şekil 2.3. Makine Öğrenmesi - Derin Öğrenme.

Derin öğrenme yöntemleri genel olarak dört ana başlık altında toplanmaktadır.

2.4.1. Derin Öğrenme Yöntemleri

2.4.1.1. Evrişimsel Sinir Ağları (CNN)

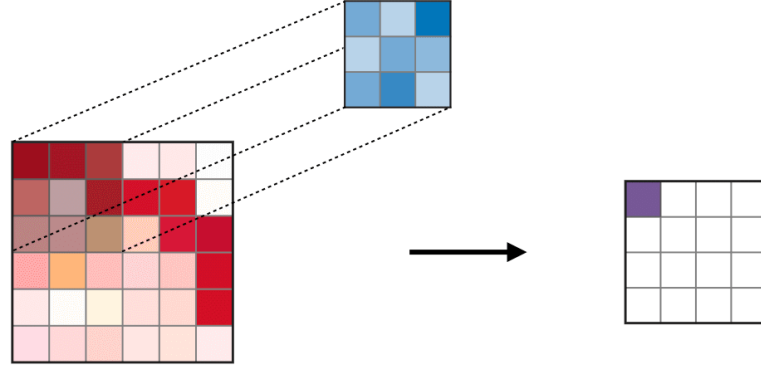
Evrişimsel Sinir Ağları (CNN), özellikle görüntü işleme ve görsel veri analizi alanlarında kullanılan derin öğrenme yöntemidir. Bu ağlar, görüntülerdeki yerel özellikleri çıkarmak için evrişim katmanları ve filtreler kullanır. CNN'ler, her katmanda öğrenilen özelliklerin giderek daha karmaşık hale gelmesini sağlar, bu da daha soyut ve anlamlı temsiller oluşturur (Şekil 2.4).



Bu özellik, CNN'leri görsel verilerle ilgili görevler için ideal hale getirir. CNN'ler, görüntü sınıflandırma, nesne algılama ve görüntü segmentasyonu gibi görevlerde yaygın olarak kullanılmaktadır. Örneğin, R-CNN, Fast R-CNN, Faster R-CNN, Mask R-CNN ve YOLO gibi modeller görsel veriler üzerinde oldukça başarılıdır [14].

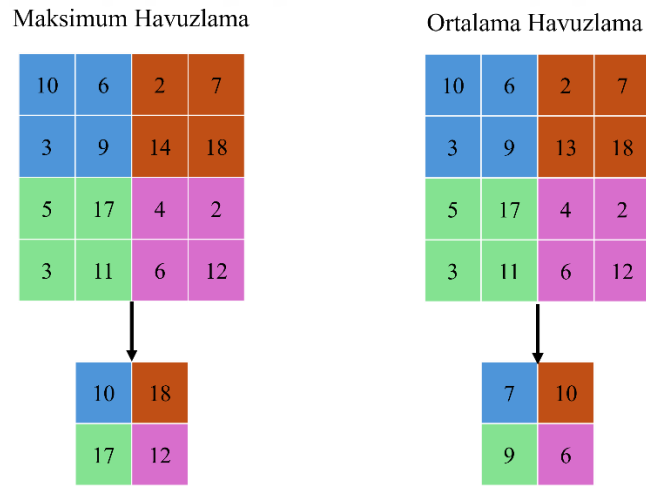
Evrişimsel Sinir Ağları, üç katmanlı bir yapıya sahiptir;

Birinci katman evrişim katmanı olup görüntünün ilk olarak ele alındığı katmandır. Bu katmanda küçük bir filtre görsel üzerinde gezerek Şekil 2.5'deki gibi özelliklerinin ayrıştırılmasını sağlar. Görseller tek bir özellikten bulundurmadiğından birden fazla filtre uygulanabilir. Çok sayıda evrişimli katmanın CNN yöntemlerinde bulunduğu sonucu ortaya çıkar.



Şekil 2.5. Evrişim Katmanı.

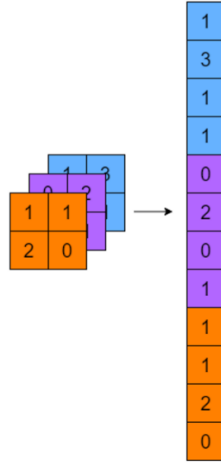
İkinci katman ise havuzlama katmanıdır. Evrişimsel Sinir Ağları (CNN) yönteminde birbirini izleyen birçok evrişim katmanı arasında bu katman bulunur. Havuzlama katmanı, CNN yönteminin yapı taşıdır. Görevi, yapay sinir ağında bulunan parametre ve hesaplamayı azaltmak için görselin katmanlarını ve boyutunu azaltmaktır. Gereksiz katmanlar dikkate alınmaz ve önemli noktalara odaklanılır. İki havuzlama yöntemi bulunmaktadır. Maksimum havuzlama yöntemi geçerli matrisin maksimum değerini seçerken ortalama havuzlama yöntemi ise matris değerlerinin ortalamasını alır [15] (Şekil 2.6).



Şekil 2.6. Maksimum ve Ortalama Havuzlama.

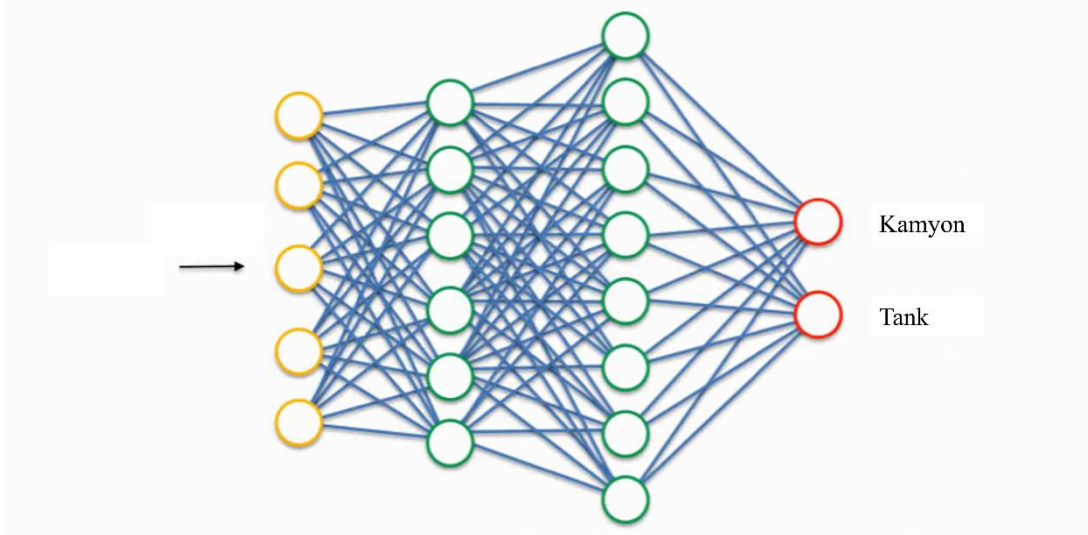
Üçüncü katman, tam bağlantılı katmandır. Bu katmanda Evrişimsel Sinir Ağlarının son adımı gerçekleşir. Veriler sinir ağlarında bir değerler dizisi biçiminde tek boyutlu olarak alındığından, bu katmanda evrişimli ve havuzlama katmandan geçen veriler kullanılır ve diziler tek boyutlu olarak düzleştirme işlemi uygulanır. Şekil 2.7’da

gösterilen düzleştirme işleminden sonra elde edilen değerler sinir ağı girdisi olarak kullanılarak sinir ağları ile öğrenme işlemi uygulanır.



Şekil 2.7. Düzleştirme İşlemi.

Tam bağlantılı katman, CNN'lerin kritik bileşenlerinden biridir ve doğru şekilde kullanıldığında modelin performansını önemli ölçüde artırabilir. Bu katman, ağıın öğrendiği özellikleri daha soyut ve anlamlı hale getirir, bu sayede sınıflandırma ve diğer görevlerde yüksek doğruluk oranlarına ulaşılabilir (Şekil 2.8).



Şekil 2.8. Tam Bağlantılı Katman.

2.4.1.2. Tekrarlayan Sinir Ağları (RNN)

Tekrarlayan Sinir Ağları (RNN), sıralı verilerdeki zaman bağımlılıklarını modellemek için kullanılan ağlardır. Bu ağlar, geçmiş adımlarda elde edilen bilgileri saklayarak sonraki adımlarda kullanır ve sıralı veri setlerinde verimli bir şekilde tahmin yapar.

Ancak, RNN'ler uzun bağımlılıkları öğrenme konusunda zorluklar yaşar ve bu sorunu çözmek için LSTM (Uzun Kısa Süreli Bellek) ve GRU (Gated Recurrent Unit) gibi gelişmiş modeller geliştirilmiştir.

RNN ve türevleri, doğal dil işleme, zaman serisi tahminleri ve ses tanıma gibi alanlarda yaygın olarak kullanılmaktadır. LSTM ve GRU, metin sınıflandırma ve makine çevirisi gibi görevlerde etkilidir [16].

2.4.1.3. Transformer Modelleri

Transformer modelleri, özellikle uzun bağımlılıkları öğrenmede verimli olan bir yapıdır. 2017 yılında Vaswani ve ekibi tarafından tanıtılan Attention is All You Need adlı makale ile büyük bir popülerlik kazanmıştır [17]. Transformer, dikkat mekanizması kullanarak sıralı verilerdeki ilişkileri öğrenir. RNN'lerin yerine kullanılabilen bu yöntem, daha hızlı ve paralel işleme yeteneğine sahiptir.

Transformer, makine çevirisi, metin oluşturma ve soru-cevap sistemleri gibi doğal dil işleme görevlerinde büyük başarılar elde etmiştir. BERT ve GPT gibi modeller, dil anlama ve metin üretme görevlerinde devrim yaratmıştır [18].

2.4.1.4. Üretici Çekişmeli Ağlar (GAN)

Üretici Çekişmeli Ağlar (GAN), iki sinir ağının birbirine rakip olduğu bir yapıyı kullanır. Bu ağlar, veri üretmek için birbirine karşı yarışan bir üretici ve bir ayıracıdan oluşur. Üretici ağ gerçekçi veriler üretirken, ayıracı ağ bu verilerin gerçek mi yoksa sahte mi olduğunu anlamaya çalışır. GAN'lar, bu rekabetçi süreç sayesinde çok gerçekçi veriler üretebilir.

GAN'lar, görüntü üretimi, sahte veri oluşturma ve artırılmış gerçeklik gibi alanlarda yaygın olarak kullanılmaktadır. Ayrıca, eğitim veri seti eksik olduğunda veriyi çoğaltmak için de kullanılır [19].

2.5. Nesne Tespit Modellerinin İncelenmesi

Nesne tespiti (object detection), bir görüntü veya video içerisindeki nesnelerin yerinin belirlenmesi ve sınıflandırılmasını içeren bir bilgisayarlı görü problemidir. Bu alandaki gelişmeler, derin öğrenme yöntemlerinin kullanımıyla büyük bir ivme kazanmıştır. Derin öğrenme tabanlı nesne tespiti, özellikle Evrişimsel Sinir Ağları (Convolutional Neural Networks, CNN) sayesinde yüksek doğruluk ve hız ile gerçekleştirilebilmektedir.

Nesne tespit modelleri, genellikle iki ana adıma sahiptir. Bunlar, yer tespiti (Localization) ve sınıflandırma (Classification) adımlarıdır. Yer tespiti adımında görüntüdeki nesnelerin koordinatlarını belirlemek için sınırlayıcı kutular (bounding boxes) oluşturulur. Sınıflandırmada ise tespit edilen nesneler önceden tanımlanmış kategorilere ayrılır. Aşağıda en çok kullanılan ve tez çalışmamızda kullanılan nesne tespit modeli anlatılmıştır.

2.5.1. R-CNN (Region-based CNN)

Girshick ve arkadaşları tarafından 2014 yılında tanıtılan R-CNN, nesne tespiti için ilk derin öğrenme temelli yaklaşımlardan biridir. R-CNN modelinde ilk olarak her bir görüntüdeki nesnelerin özellikleri, CNN kullanılarak çıkarılır. Ancak önce, bu görüntüdeki bölgesel adaylar belirlenmelidir. Bu süreç, Selective Search adı verilen bir algoritma ile yapılır [20].

Selective Search, görüntüdeki nesnelerin bulunduğu yerleri tahmin etmek için kullanılan bir bölgesel öneri algoritmasıdır. Bu algoritma, görüntüyü çeşitli bölgeler (regions of interest - RoI) olarak böler ve bu bölgelerdeki anlamlı nesne adaylarını belirler. Bu aday bölgeler, daha sonra CNN'e gönderilir.

Selective Search tarafından üretilen aday bölgeler, görüntüdeki potansiyel nesnelerin yerlerini temsil eder. Bu adaylar, çeşitli renk, doku ve benzerlik özelliklerine göre gruplanmış bölgelerden oluşur. Bu aşamada genellikle 2000 kadar öneri üretilir.

Aday bölgeler, önceden eğitilmiş bir Evrişimsel Sinir Ağına verilir. CNN, her bir önerilen bölgeden özellikler çıkarır. Bu özellikler, genellikle daha düşük çözünürlükte ve daha anlamlı temsil edilen vektörlerdir.

Çıkarılan özellikler, her bir bölgenin bir nesneye ait olup olmadığını belirlemek için kullanılır. Bunun için, Support Vector Machines (SVM) gibi bir sınıflandırıcı kullanılır. SVM, her bir bölgeyi belirli bir nesne sınıfına veya arka planda olup olmadığına göre sınıflandırır.

Ayrıca, R-CNN modeli, her bir nesne bölgesinin daha hassas konumunu bulmak için sınırlayıcı kutu (bounding box) regresyonu kullanır. Bu regresyon, nesnenin bulunduğu kutunun konumunu (x, y, genişlik, yükseklik) daha doğru bir şekilde belirlemek için kullanılır.

(RPN) kullanarak tamamen CNN tabanlı hale getirmesidir. Bu yenilik, bölgesel öneriler için kullanılan Selective Search gibi yavaş ve CPU tabanlı algoritmaları ortadan kaldırarak, işlemi tamamen GPU üzerinde hızlı bir şekilde gerçekleştirmeyi mümkün kılmıştır [23].

Avantajı, Faster R-CNN, Selective Search gibi yavaş ve CPU tabanlı algoritmaları ortadan kaldırarak Region Proposal Network (RPN) sayesinde nesne önerisi işlemi çok daha hızlı hale gelir ve tamamen GPU üzerinde çalışır. Bu sayede hem nesne tespitinde hem de sınıflandırmada yüksek doğruluk sağlar. Faster R-CNN, hem nesne önerisini hem de sınıflandırmayı aynı ağ üzerinde uçtan uca öğrenir. Bu, daha tutarlı ve hızlı bir eğitim süreci sağlar. Farklı CNN mimarileri (ResNet, VGG vb.) kullanılarak doğruluğu artırılabilir. Dezavantajı ise Faster R-CNN, çok hızlı olsa da, gerçek zamanlı uygulamalar için yeterince hızlı değildir.

2.5.4. Mask R-CNN

Faster R-CNN'in bir uzantısı olan Mask R-CNN, He ve arkadaşları tarafından geliştirilmiştir. Nesne tespiti ve sınıflandırma gibi görevlerin ötesine geçerek, her nesne için piksel seviyesinde segmentasyon (instance segmentation) yapabilen bir derin öğrenme algoritmasıdır. Faster R-CNN algoritmasının bir genişlemesi olan Mask R-CNN, nesnelerin yalnızca sınırlayıcı kutularını değil, aynı zamanda her bir nesnenin piksel bazında maskesini (şeklini) de tahmin eder [24]. İncelenen dört R-CNN modelinin özellikleri Tablo 2.2' de sunulmuştur.

Tablo 2.2. R-CNN Modellerinin Özellikleri.

Özellik	R-CNN	Fast R-CNN	Faster R-CNN	Mask R-CNN
Özellik	Her bölge için CNN üzerinden	Tüm görüntüden tek seferde	Tüm görüntüden tek seferde	Tüm görüntüden tek seferde
Çıkartma	özellik çıkartma	özellik haritası çıkartma	özellik haritası çıkartma	özellik haritası çıkartma
Eğitim	Parçalı	Uçtan uca	Uçtan uca	Uçtan uca
Hız	Çok yavaş	Daha hızlı	Çok daha hızlı	Yavaş
Gerçek Zamanlı	Hayır	Hayır	Kısmen	Hayır

2.5.5. You Only Look Once (YOLO)

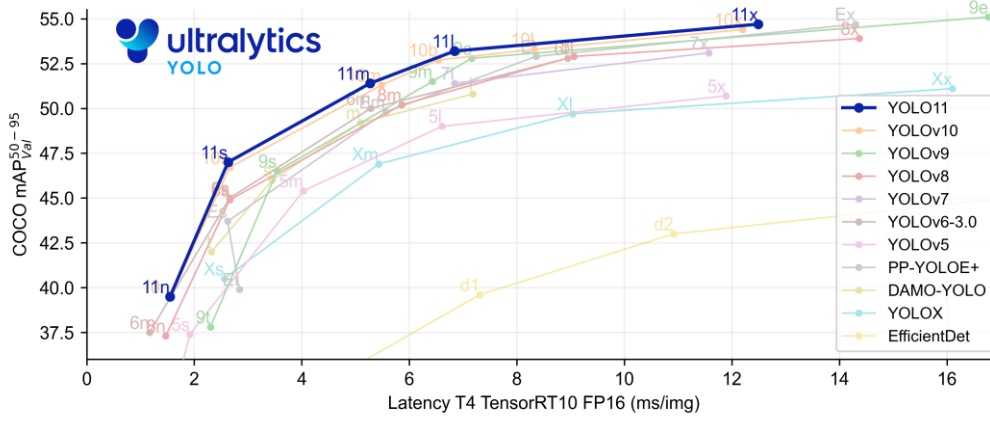
YOLO (You Only Look Once), bu tez çalışması kapsamında kullanılan model olup Evrişimsel Sinir Ağları (CNN) yöntemini kullanılan bir modeldir. YOLO, nesne algılama (object detection) problemlerine yönelik hızlı ve etkili bir yaklaşım sunar. Evrişimsel sinir ağlarının temel prensiplerini kullanarak bir görüntüdeki nesneleri algılar ve sınıflandırır.

YOLO, 2015 yılında Joseph Redmon ve arkadaşları tarafından geliştirilmiştir [25]. YOLO'nun en önemli amacı, nesneleri hem hızlı hem de doğru bir şekilde tespit etmektir. YOLO, bir görüntüyü tek bir ağ geçişiyle işler. Algoritmanın temel prensibi bir görüntüyü parçalara ayırmadan tüm görüntüyü bir bütün olarak işlemesi ve görüntüleri bir ızgaraya bölerek her bir hücrede nesne olup olmadığını tahmin etmesi olarak öne çıkmaktadır.

YOLO'nun diğer modellere göre, gerçek zamanlı nesne tespiti için yeterince hızlı olması, hem büyük hem de küçük nesneler üzerinde oldukça iyi performans göstermesi, bir sinir ağı kullanarak görüntüden doğrudan nesne tespiti yapması ve düşük maliyet özellikleri ile öne çıkmaktadır. 2015 yılında geliştirilmesine başlanan YOLO modeli zaman içerisinde çeşitli sürümleri yayınlanarak geliştirilmeye devam etmiştir.

Özellikle trafik analizi (araçları, yayaları ve trafik işaretlerini tanımlamak), güvenlik (insan algılama, yüz tanıma ve tehdit tespiti), e-ticaret (ürünlerin otomatik kategorize edilmesi) alanlarında yoğun olarak kullanılmaktadır.

Tez kapsamında, YOLOv8 versiyonu kullanılacak olup en çok bilinen ve uygulanan versiyon olarak öne çıkmaktadır. Şekil 2.10'da mevcut YOLO versiyonlarının örnek veri setinde doğruluk ve gecikme süreleri karşılaştırma tablosu verilmiştir.



Şekil 2.10. YOLO Versiyonları Performans Karşılaştırması [26].

YOLOv8, yüksek doğruluk, hız ve çok yönlülük sunmaktadır. Bu model, YOLOv5 üzerine inşa edilmiştir ancak optimizasyonlar ve yeni modüllerle performansı ileri taşınmıştır.

YOLOv8 model mimarisinin, daha modüler bir yapıya sahip olması sayesinde özelleştirilmeye açık hale getirilmiştir. Backbone, neck ve head olmak üzere üç temel bileşenden oluşan YOLOv8, bir görüntüyü tek bir geçişte analiz ederek nesneleri tespit eder. Bu süreçte, görüntü bir grid'e bölünür ve her bir hücre için bir tahmin yapılır. Non-maximum suppression (NMS) algoritması sayesinde ise aynı nesneye ait çoklu tahminler birleştirilerek en doğru sonuç elde edilir [27].

Giriş görüntüsünden anlamlı özellikleri çıkarmak Backbone bileşeninde gerçekleşirken çok ölçekli özellik haritalarının birleştirilip, güçlendirilmesi Neck bileşeninde gerçekleşir. Nihai tespitler (nesne sınıfı, konumu ve opsiyonel olarak segmentasyon) ise Head bileşeninde yapılır. YOLOv8'in backbone bölümünde yer alan ana bileşenler şunlardır:

Conv Blokları (Convolution): Görüntü üzerinde evrişim işlemleri yaparak temel özellikleri çıkarır. Bu bloklar, sıralı olarak yerleştirilmiştir ve evrişim filtreleriyle farklı seviyelerdeki detayları öğrenir.

C2F Blokları: CSPNet tabanlı C2F (Cross Stage Partial with Fusion) modülü kullanılır. Bu yapı, özellik haritalarının kısmen bölünerek işlenmesini sağlar. Bu sayede hem verimlilik artar hem de hesaplama maliyeti düşer.

SPPF (Spatial Pyramid Pooling - Fast): Bu blok, giriş özellik haritalarını farklı ölçeklerde havuzlayarak çoklu ölçekli bilgi öğrenimini sağlar. SPPF, YOLOv8'in daha

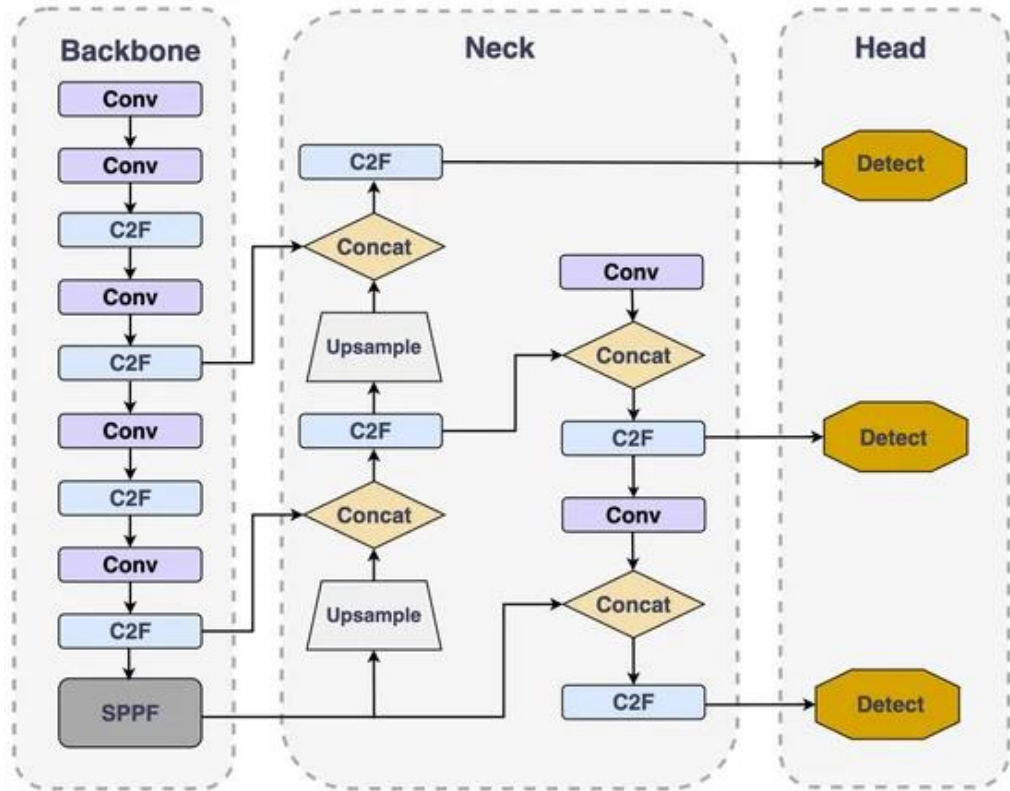
hızlı ve verimli çalışmasını destekleyen bir havuzlama yöntemidir. Backbone, derinleştikçe daha karmaşık ve büyük ölçekli özellikleri öğrenir ve bunları bir sonraki aşamaya (Neck) iletir. Neck bölümünde ise aşağıdaki bileşenler bulunur:

C2F: Backbone’da kullanılan C2F modülleri burada da kullanılır. Bu yapı, özelliklerin daha verimli bir şekilde birleştirilmesini sağlar.

Upsample: Daha küçük ölçekli özellik haritalarını büyötmek için kullanılır. Bu, özellikle küçük nesneleri tespit ederken faydalıdır.

Concat (Concatenation): Farklı katmanlardan gelen özellik haritalarını birleştirir. Bu sayede hem düşük seviyeli (detaylı) hem de yüksek seviyeli (soyut) bilgiler aynı anda kullanılabilir. Neck bölümün temel amacı, özellikleri çoklu ölçekli bir şekilde birleştirerek nesne tespit performansını artırmaktır.

Head bölümü ise, nihai çıktıların üretildiği bölümdür. Neck tarafından üretilen çok ölçekli özellik haritaları burada işlenir ve her bir hücre için nesne sınıfı, sınırlayıcı kutu (konum ve boyut) ve güven skoru tahmin edilir. YOLOv8’in bu kısmı, optimize edilmiş bir head yapısıyla hızlı ve doğru tahminler yapacak şekilde tasarlanmıştır. Şekil 2.11’da YOLOv8 mimarisi gösterilmiştir.



Şekil 2.11. YOLOv8 Mimarisi [27].

Performans olarak YOLOv8, ortalama doğruluk metriği (mAP) açısından önemli bir iyileştirme sunar. Örneğin, YOLOv8'in mAP@0.5 değeri %52.5 olarak rapor edilmiştir, bu da YOLOv5'in %50.5'ine kıyasla daha iyidir. Hız ve verimlilik açısından ise YOLOv8, görüntü başına yalnızca 25 ms gibi kısa bir sürede işlem yapabilir [28].

YOLOv8'in en önemli avantajlarından biri, gerçek zamanlı uygulamalar için ideal olan yüksek hızı ve doğruluğu bir arada sunmasıdır. Bu sayede, güvenlik sistemleri, otonom araçlar, tıbbi görüntüleme ve endüstriyel otomasyon gibi birçok alanda kullanılmaktadır. Açık kaynak olması, modelin sürekli olarak geliştirilmesini sağlarken, kullanıcı dostu arayüzü sayesinde de eğitimi ve kullanımı oldukça kolaydır.

2.6. İHA Görüntülerinin Analizi Kapsamında Yapılan Çalışmaların İncelenmesi

İnsansız Hava Araçları (İHA) tarafından elde edilen görüntülerin analizi, son yıllarda hem akademik hem de endüstriyel alanlarda büyük bir ilgi görmüş ve özellikle görüntü işleme, yapay zekâ ve uzaktan algılama gibi disiplinlerde önemli araştırmalara konu olmuştur.

Bu kapsamda 2015 yılında Kim ve Chervonenkins [29] tarafından gerçekleştirilen çalışmada, İHA kullanılarak yol trafiğinin izlenmesine yönelik bir yaklaşım geliştirilmiştir. Çalışmada, özellikle derin öğrenme yöntemleri kapsamında yapay sinir ağları ve makine öğrenimi tekniklerinden yararlanılarak görüntü segmentasyonuna dayalı bir araç tespit yöntemi önerilmiştir. Araştırma, trafik durumlarının analiz edilmesi amacıyla araç takibi ve hız tespiti gibi sorunlara çözümler sunmayı hedeflemiştir. Önerilen yöntem, yönetim ve tespit süreçlerinin otomasyonu aracılığıyla, İHA'ların karayolu trafiğinin izlenmesindeki etkinliği artırdığı ve bu alanda önemli katkılar sağladığı vurgulanmıştır.

2018 yılında Benjdira ve ark. [30] tarafından yayımlanan çalışmada İHA'ların yüksek manevra kabiliyetleri ile farklı irtifa ve konumlardaki alanları kapsama kapasiteleri sayesinde gözetim ve trafik izleme alanında giderek daha yaygın bir şekilde kullanıldığı vurgulanmıştır. Çalışmada, trafik izleme amacıyla araçları doğru bir şekilde tespit etme ve gerçek zamanlı olarak sayma işlemlerinin karşılaşılan en büyük zorluklardan biri olduğu belirtilmiştir. Son yıllarda bilgisayarla görme alanında gerçek zamanlı sınıflandırma ve tanıma işlemleri için çeşitli derin öğrenme tekniklerinin, özellikle evrişimli sinir ağlarına (CNN) dayalı yöntemlerin önerildiği ifade edilmiştir. Ancak, bu yöntemlerin performanslarının kullanım senaryolarına bağlı olarak

değişkenlik gösterdiği, araştırma kapsamında, Faster R-CNN ve YOLOv3 olmak üzere iki CNN modelinin, hava görüntülerinden araç tespiti bağlamındaki performansı incelenmiştir. Bu modeller, İHA'lerden elde edilen geniş kapsamlı bir araç veri kümesi üzerinde eğitilmiş ve test edilmiştir. Sonuçlar, YOLOv3'ün hem hassasiyet hem de işlem süresi açısından Faster R-CNN'e kıyasla üstün bir performans sergilediğini, ancak hassasiyet açısından karşılaştırılabilir sonuçlar elde ettiğini ortaya koymuştur.

2018 yılında Toraman [31] tarafından, derin öğrenme yardımıyla özellik çıkarılarak İHA görüntülerinden yaya tespiti üzerinde çalışılmış, İnsansız Hava Araçları (İHA) tarafından elde edilen yaya, bisikletli, araç, ağaç ve sokak lambası gibi nesnelerin görüntüleri, belirlenen boyutlara ölçeklendirilerek CNN tabanlı modeller olan AlexNet ve VGG16'ya giriş verisi olarak kullanılmıştır. Bu modeller aracılığıyla görüntülerden özellik çıkarımı gerçekleştirilmiştir. Elde edilen özellikler, Destek Vektör Makineleri (SVM) kullanılarak sınıflandırılmış ve bu sınıflandırma işlemi sonucunda yaya ile diğer nesnelerin ayrımı yapılmıştır. Çalışma kapsamında, AlexNet ve VGG16 modellerinin performansları karşılaştırılarak her iki modelin sınıflandırma başarımları analiz edilmiştir.

2019 yılında Şin ve Kadioğlu [32] tarafından “İnsansız Hava Aracı (İHA) ve Görüntü İşleme Teknikleri Kullanılarak Yabancı Ot Tespitinin Yapılması” konusu üzerinde çalışılmış, İHA'lar ile görüntü işleme teknikleri kombine kullanılarak yabancı otların tespit edilmesi, takibi, teşhisi ve yoğunluklarının belirlenerek mücadele edilmesi konusuna değinilmiştir.

2021 yılında ise Şahin ve Yılmaz [33] tarafından “Hacim Hesaplarında İnsansız Hava Aracı (İHA) Verilerinin Kullanılabilirliğinin Araştırılması” konusu üzerinde durulmuş, Harita Mühendisliğinin İnşaat ve Madencilik sektöründeki en önemli iş kollarından birisinin hacim hesabı olduğu, özellikle erişilmesi zor bölgelerde hacim hesaplarının bölgenin havadan çekilen resimleri yardımıyla yapılmasının uygun olduğu değerlendirilmiştir. Çalışma kapsamında SİLO'ların (tarımsal depolama tesisi) hacimlerinin hesaplanması için İHA'lar ile görüntüleri alınmış, elde edilen veriler 3 ayrı modelleme yazılımında değerlendirilerek hacim hesabı yapılmıştır. Kataloglarından hacmi bilinen SİLO'ların hacimleri ile elde edilen hacim hesapları karşılaştırılmış, 3 modelde elde edilen doğruluk oran ortalamasının %98 olduğu tespit edilmiştir. Aynı yıl Aksoy ve ark. [34] tarafından “İnsansız Hava Aracı ile Görüntü İşleme ve Yapay Zekâ Teknikleri Kullanılarak Yangın Tespiti” konusundan bir

alışma yapılmıřtır. Son yıllarda orman ve řehir yangınlarının artmasının dnya iin bir tehdit haline geldiđi, ekolojik ve ekonomik aıdan byk kayıplara sebep olduđu, bu dođrultuda yařamın gerekliliđi aısından nemli ekosistemlerden olan ormanların ve dođanın korunması nem arz ettiđi belirtilmiřtir. Yangının erken tespiti orman ve řehir yangınları ile mcadelede byk kayıpların nne getiđi, teknolojinin geliřimi ile birlikte yangınların erken tespit edilebilmesi amacıyla insansız hava araları (İHA), yapay zekâ ve grnt iřleme tekniklerinden yararlanıldıđı anlatılmıřtır. alıřmada, grnt iřleme ve yapay zekâ tekniklerinden yararlanılarak grntlerde ateř tespiti yapılması ve yangının konumunun belirlenmesi zerine alıřılmıřtır. Elde edilen sonularda mimarinin karmařıklık matrisine gre performansı deđerlendirilerek, %96 dođruluk, %98 duyarlılık, %89 zgllk ve %96 kesinlik deđerleri bulunarak, yangın tespiti alanında bařarı sađlanabileceđi belirtilmiřtir.

2023 yılında Silva ve ark. [35] tarafından İHA grntlerinin derin đrenme yntemleri ile iřlenerek kara yolları zerinde bulunan hasarların tespiti konusunda alıřma yapılmıřtır. zetle, kara yolları altyapısının sađlam ve gvenilir olmasının ulařım aısından kritik neme sahip olduđu, hasarlı yol verilerinin manuel toplanmasının emek yođun ve maliyetli olduđu, bu sebeple İHA'lar ve yapay zekâ teknolojilerini kullanarak yol hasarı tespitinin verimliliđini ve dođruluđu zerinde durulmuřtur. alıřmada nesne tespiti ve lokalizasyonu iin YOLOv4, YOLOv5 ve YOLOv7 olmak zere  model kullanılmıřtır. İHA yol grntleri olarak in ve İspanya yol veri seti olan RDD2022 veri seti kullanılmıř, YOLOv4 srm iin mAP@50 deđer %26.8, YOLOv5 srm iin mAP@50 deđer %59.9, YOLOv7 srm iin mAP@50 deđer %73.2 olarak hesaplanmıřtır. Bu sonular otomatik yol hasarı tespiti ve asfaltlama iin İHA'ların ve derin đrenmenin kullanılma potansiyeli gsterdiđi, gelecek arařtırmalara yol gsterici olacađı belirtilmiřtir.

2023 yılında Kumbasar [36] tarafından yayımlanan tez alıřmasında, İHA'lar ile elde edilen grntler kullanılarak arkeolojik alanların tespitine ynelik bir arařtırma yapılmıřtır. alıřmada, arkeolojik alanların belirlenmesinde manuel yzey arařtırmalarına olan ihtiyaı ortadan kaldıracabilecek bir yntem olarak, İHA grntleriyle entegre edilen derin đrenme tekniklerinin potansiyeli incelenmiřtir. Arařtırma sonuları, İHA verilerinin kullanımıyla arkeolojik alanların tespitinde

yüksek doğruluk oranlarına ulaşılabileceğini ve bu yaklaşımın geleneksel yöntemlere etkili bir alternatif sunduğu belirtilmiştir.

2023 yılında Çobanoğlu [37] tarafından yayınlanan tez çalışmasında, uydu görüntüleri kullanılarak askeri uçak tiplerinin tespiti ve sınıflandırılmasına yönelik bir yöntem geliştirilmiştir. Çalışmada, derin öğrenme temelli yaklaşımlar benimsenmiş ve transfer öğrenme yöntemi ile birlikte VGG16 derin öğrenme mimarisi kullanılmıştır. Askeri savaş uçaklarının tespiti için gerekli veri seti, MTARSI platformundan alınan uydu görüntüleri temel alınarak oluşturulmuştur. Bu veri seti, eğitim ve test süreçleri için 5 farklı uçak tipini içerecek şekilde hazırlanmıştır. Elde edilen sonuçlar, önerilen modelin askeri uçak tiplerinin sınıflandırılmasında %97.92, tespitinde ise %99 oranında yüksek bir doğruluk sağladığını ortaya koymuştur. Bu çalışma, uydu görüntülerinden askeri nesnelerin analizi ve sınıflandırılması alanında önemli bir katkı sunmaktadır.

2024 yılında Türker [38] tarafından yayınlanan tez çalışmasında, derin öğrenme teknikleri kullanılarak uydu görüntülerinden gemi tespiti ve sınıflandırılmasına yönelik bir araştırma yapılmıştır. Çalışmada, bilgisayarla görme alanındaki nesne tespiti problemleri için sıklıkla tercih edilen You Only Look Once (YOLO) tabanlı mimariler kullanılmıştır. Bu mimariler, transfer öğrenme yöntemi uygulanarak optik uydu görüntüleriyle eğitilmiş ve gemileri tespit edip sınıflandırma yetenekleri geliştirilmiştir. Eğitim sonrasında, mimarilerin performansları doğruluk, duyarlılık, mAP ve F1 skor ölçütleri üzerinden değerlendirilmiştir. Test sonuçları, YOLOv5l modelinin %98,4 doğruluk, %98,4 duyarlılık, %99,1 mAP ve %98,4 F1 skoru ile diğer modeller arasında en yüksek performansı gösterdiğini ortaya koymuştur. Çalışma, özellikle daha büyük modellerin gemi tespiti görevlerinde daha üstün bir performans sergilediğini göstermiştir. Çalışmada, YOLO tabanlı mimarilerin uydu görüntülerinden gemi tespiti ve sınıflandırılmasında etkili ve umut vadeden bir çözüm sunduğu tespit edilmiştir.

2024 yılında Çimen [39] tarafından yapılan çalışmada, derin öğrenme tabanlı görüntü sınıflandırma tekniklerinden GoogleNet ve ResNet18 modelleri kullanılarak hava araçlarının sınıflandırılmasına yönelik performans analizi yapılmıştır. Araştırma kapsamında, yolcu uçağı, askeri uçak, helikopter, drone ve roket olmak üzere 5 farklı sınıfa ait yaklaşık 5000 hava aracı görüntüsünden oluşan hazır bir veri seti kullanılmıştır. Bu veri seti ile modellerin eğitim ve test süreçleri gerçekleştirilmiş, elde

edilen sonuçlar analiz edilmiştir. Çalışmada, eğitim verisinin miktarının artırıldığı durumlarda, genel olarak sınıflandırma başarısının anlamlı bir şekilde yükseldiği gözlemlenmiştir. Araştırma bulguları, derin öğrenme modellerinin hava aracı sınıflandırma görevlerinde etkin bir şekilde kullanılabileceğini ve yeterli miktarda veri ile daha yüksek doğruluk oranlarına ulaşılabilirliğini ortaya koymuştur.

2024 yılında Doğan [40] tarafından gerçekleştirilen tez çalışmasında, derin öğrenme yöntemleri kullanılarak İHA'ların tespitiye yönelik bir araştırma yürütülmüştür. Çalışmada, günümüzde İHA'ların erişilebilirliğinin oldukça kolaylaştığı ve bu araçların askeri operasyonlar, eğlence, spor, kargo taşımacılığı gibi birçok alanda yaygın bir şekilde kullanıldığı ifade edilmiştir. Bununla birlikte, İHA'ların kaçakçılık, casusluk ve terör faaliyetleri gibi yasa dışı amaçlarla da kullanılabileceğine dikkat çekilmiştir. Kamu güvenliği ve gizliliğine ilişkin endişeleri gidermek amacıyla, yetkisiz ve yasa dışı İHA'ların tespitiye kritik öneme sahip olduğu vurgulanmış ve optik yaklaşıma dayalı bir tespit yöntemi geliştirilmiştir. Araştırma kapsamında, derin öğrenme tabanlı YOLO mimarisinin v8l ve v9c modelleri kullanılarak iki farklı veri setiyle model eğitimi gerçekleştirilmiştir. Elde edilen sonuçlar, eğitilen modeller arasında en yüksek tespit başarısının %99,2 oranıyla elde edildiğini ortaya koymuştur. Ayrıca, yasa dışı uçuş simülasyonları gerçekleştirilmiş ve önerilen yöntemin İHA'ları yüksek doğrulukla tespit ettiği gösterilmiştir. Bu çalışmanın, İHA'ların güvenlik ve gizlilik açısından oluşturduğu tehditlere karşı etkili bir çözüm sunduğu belirtilmiştir.

2024 yılında Çetinkaya [41] tarafından bir önceki çalışmaya benzer bir çalışma yapılmış, İHA'ların kötü niyetli bireyler veya terör örgütleri tarafından saldırılarda kullanılma potansiyeline dikkat çekilmiştir. Bu tür saldırıların önceden tespit edilip engellenmesinin zorluğu vurgulanmış ve bu durumun ciddi can ve mal kayıplarına yol açabileceği belirtilmiştir. Mevcut önleme yöntemlerinden biri olan GPS sinyallerinin karıştırılması, meskûn mahal bölgelerinde günlük yaşamı olumsuz etkileyebileceği için etkili bir çözüm olarak değerlendirilmemiştir. Bunun yerine, İHA'ların öncelikle tespit edilmesi ve ardından bertaraf edici veya engelleyici sistemlerin devreye sokulması gerektiği ifade edilmiştir. Çalışma, meskûn mahal alanlarında İHA tespitiye zorluğuna dikkat çekmiştir. Geleneksel radar ve algılama sistemlerinin genellikle sabit bir konumda konuşlandırıldığı ve yayın yaptığı için maniaların oluşturduğu engellerin arkasını algılayamadığı belirtilmiştir. Bu bağlamda, meskûn mahal alanlarında uçan İHA'ların başka bir İHA tarafından izlenmesi ve derin

öğrenme tabanlı YOLO (You Only Look Once) ağ mimarisi ile tespit edilmesi hedeflenmiştir. Bu amaç doğrultusunda, meskûn mahal alanlarında İHA görüntüleri toplanarak bir veri tabanı oluşturulmuştur. Toplanan görüntüler, YoloLabelv1 yazılımı kullanılarak etiketlenmiş ve YOLOv5 ile YOLOv8 derin öğrenme ağ mimarileri eğitilmiştir. Eğitim sonrası, bu modeller kullanılarak dronların otomatik olarak başarılı bir şekilde tespit edildiği gösterilmiştir. Çalışma, meskûn mahal alanlarda İHA tespiti için derin öğrenme yöntemlerinin etkili bir çözüm sunduğunu ortaya koymuştur.





3. MATERYAL VE YÖNTEM

Bu bölümde, İHA'lerden elde edilen görüntülerin sınıflandırılması için gerçekleştirilecek olan tüm çalışmalar ve analizler ayrıntılı bir şekilde sunulmuştur.

3.1. Veri Seti Toplama

Veri toplama ve veri seti oluşturma, makine ve derin öğrenme projelerinin temel taşlarından biridir. Bu süreç, modelin eğitimi, doğrulanması ve test edilmesi için gerekli olan verilerin toplanması, düzenlenmesi ve işlenmesini içerir. İyi bir veri seti, modelin doğruluğunu ve genelleme yeteneğini artırmada hayati bir rol oynar.

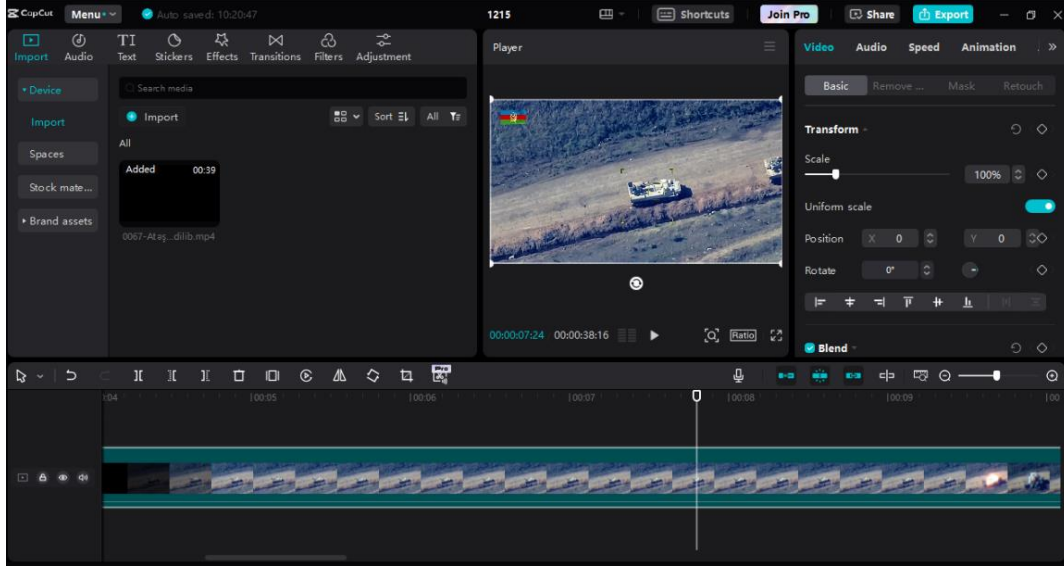
Bu çalışmada, açık kaynaklarda sunulan ve İHA sistemlerinden elde edilen gerçek görüntüler kullanılarak veri seti oluşturulmuş, hazır veri setlerinden yararlanılmamış, yapılan detaylı araştırmalar sonucu veri setinin daha önce aşağıda detayları anlatılacağı şekilde oluşturulmadığı anlaşılmıştır.

Veriler, internet üzerinde insanlara açık olarak sunulan Youtube, Twitter ve Telegram platformları taranarak video formatında bilgisayara indirilmiş, içerisinden kullanılabilecek özellikte olanlar ayıklanarak ayrı bir dosya içerisinde toplanmıştır.

3.2. Veri Seti Ön İşleme

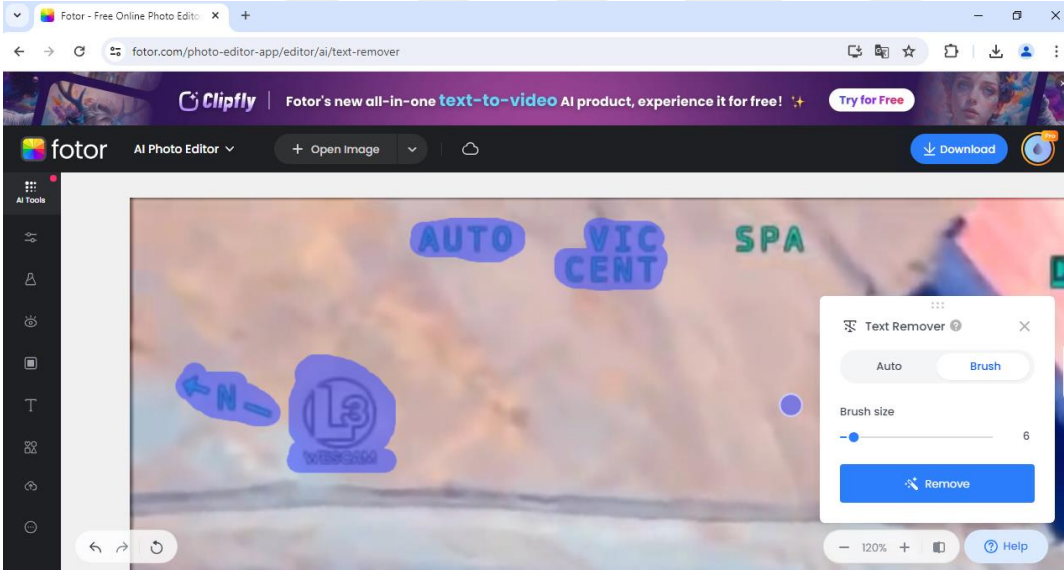
Ön işleme, verilerin analiz veya modelleme amacı doğrultusunda uygun bir forma getirilmesi ya da potansiyel engellerin ortadan kaldırılması için uygulanan yöntemler bütünüdür [42]. Ön işlem adımları veri, durum, isterler, ortam gibi birçok etmene göre süreç içerisinde değişebilen yöntemlerdir. Eksik, tutarsız ve gürültülü veriler bulunması ön işleme adımlarının nedenleri arasındadır.

Bir dosya içerisinde kullanılabilecek özellikte olarak toplanan videolar CapCut [43] programı ile karelere ayrılarak veri setinde kullanılabilecek görüntüler fotoğraf karesi olarak jpg formatında kaydedilmiştir (Şekil 3.1).



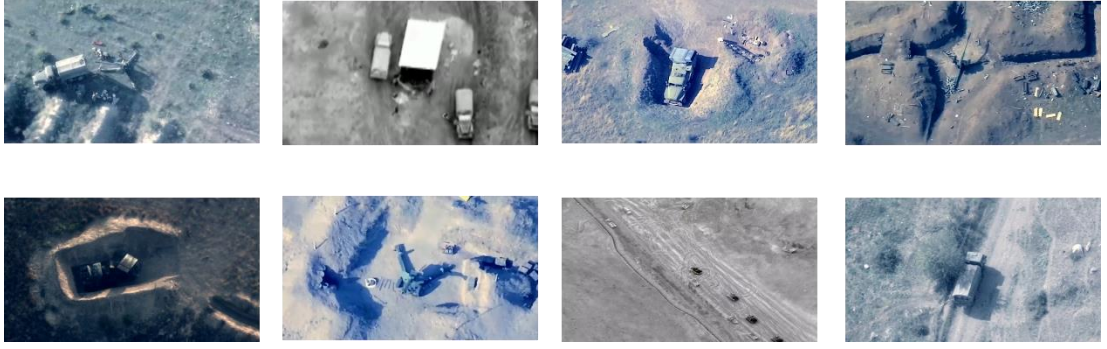
Şekil 3.1. CapCut Programı ile Karelerin Alınması.

Veri setimizin devam eden ön işleme aşamasında fotoğrafların üzerindeki gürültüler ayrıştırılmış, gereksiz imgeler Fotor [44] uygulaması vasıtasıyla verilerin bütünlüğü bozulmadan kaldırılarak temizlenmiştir (Şekil 3.2).



Şekil 3.2. Fotor Uygulaması ile Fotoğrafların Düzenlenmesi.

Veri setindeki kamyon, tank, hava savunma sistemi ve top bataryalarına ait örnek görseller Şekil 3.3’de gösterilmiştir. Netice olarak farklı durum ve şartları barındıran, farklı sınıflara ait veri seti oluşturulmuştur.



Şekil 3.3. Veri Seti Örnek Görseller.

3.3. Veri Seti Etiketleme

Görüntü sınıflandırma ile çalışan derin öğrenme modelleri için her görüntünün sınıf etiketi ile etiketlenmesi gerekir. Çalışmalardan doğru sonuçlar elde edilebilmesi verilerin etiketlenme sürecinin doğru ve titizlikle yapılmasına bağlıdır.

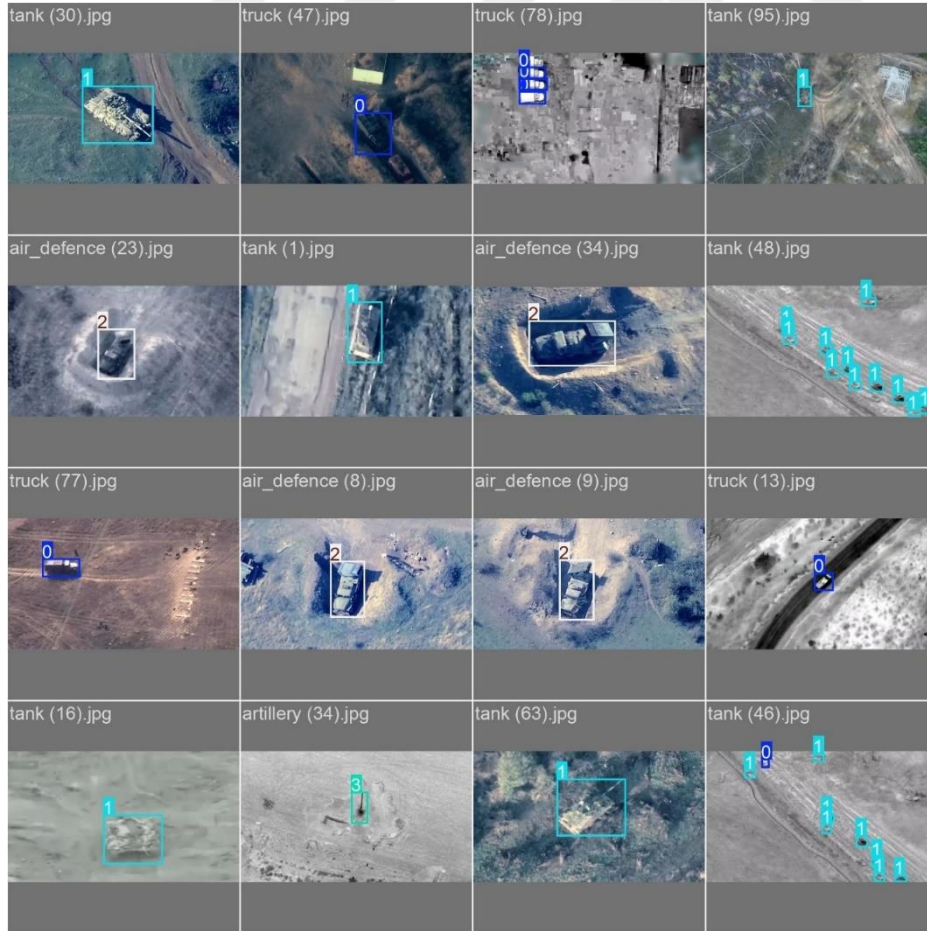
Veri seti etiketlemede sınırlayıcı kutu (bounding box), anlamsal segmentasyon, örnek segmentasyonu, anahtar nokta anotasyonu, çokgen anotasyonu, 3D sınırlayıcı kutu, lidar anotasyonu, poz tahmini anotasyonu gibi yöntemler bulunmaktadır. Çalışmamızda sınırlayıcı kutu yöntemi kullanılmış olup sınırlayıcı kutu, bir nesneyi görüntüde en kolay şekilde tanımlamanın yoludur. Nesneyi çevreleyen dikdörtgen çizilir ve bu da modellere nesnenin konumunu ve boyutunu anlamakta yardımcı olur. Bu yöntem, nesne tespiti ve sınıflandırma modellerinde yaygın olarak kullanılır. Yaygın olarak YOLO, Faster R-CNN, Single Shot MultiBox Detector (SSD) modellerinde tercih edilir.

Sınırlayıcı kutu yöntemi, labelImg 1.8.1 [45] programı ile kullanılarak tüm verilere uygulanmıştır. Toplanan veri seti, truck, tank, air_defence, artillery olarak 4 başlıkta etiketleme işlemi yapılmıştır (Şekil 3.4).



Şekil 3.4. labelImg 1.8.1 ile Veri Etiketleme.

Veri seti incelendiğinde yaklaşık 400 veri üzerinde belirlenen 4 başlık altında 1000'e yakın sınırlayıcı kutu etiketlemesi yapılarak veri seti hazır hale getirilmiştir. Etiketlenmiş veri setine örnek görseller Şekil 3.5'de gösterilmiştir.

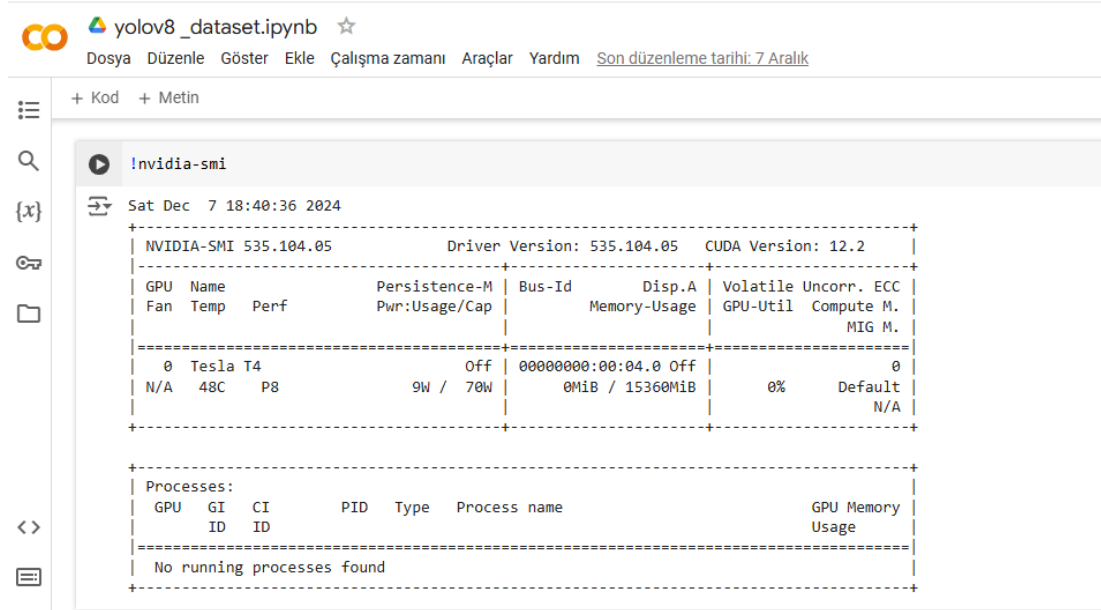


Şekil 3.5. Etiketlenmiş Veri Seti Örnek Görseller.

YOLOv8 ile eğitime başlamadan önce veri setindeki görseller %80 eğitim verisi, %10 doğrulama verisi, %10 ise test verisi olarak rastgele seçilmiştir.

3.4. YOLOv8 ile Veri Setinin Eğitilmesi

Hazırlanan veri seti ile tüm eğitim ve tespit işlemleri için Google hizmeti olan Colaboratory sistemi kullanılmıştır. Google Colab, makine öğrenimi, derin öğrenme, veri analitiği ve diğer Python tabanlı projeler için popüler bir bulut tabanlı araçtır [46]. Google Colab, çevrimiçi bir Jupyter Notebook olarak çalışır ve kullanıcıların kod yazmasını, çalıştırmasını, görselleştirmeler oluşturmalarını ve açıklamalar eklemesini sağlar. Tüm çalışmalar Google Drive ile entegre bir şekilde saklanır, böylece herhangi bir cihazdan kolaylıkla erişim mümkündür. İlgili hizmet ile özellikle görsel tabanlı eğitimlerde önemli bir araç olan GPU özelliği için yüksek performanslı donanımlar seçilerek elde edilen sonuçların süresi kısaltılabilmektedir. Ücretsiz versiyonu ile 12 saate kadar bağlantı izni verilerek eğitimler yaptırılabilen, Tesla T4 GPU gibi yüksek performanslı donanımlar kullanılabilmektedir (Şekil 3.6).



```
!nvidia-smi

Sat Dec 7 18:40:36 2024

+-----+
| NVIDIA-SMI 535.104.05                 Driver Version: 535.104.05   CUDA Version: 12.2   |
+-----+-----+
| GPU  Name            Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap|   Memory-Usage | GPU-Util  Compute M. |
|=====================================+=====+
| 0 Tesla T4               Off      | 00000000:00:04:0 Off |                    0 |
| N/A  48C    P8             9W / 70W   | 0MiB / 15360MiB |      0%   Default   |
+-----+-----+

+-----+
| Processes: |
| GPU   GI   CI        PID   Type   Process name                        GPU Memory |
| ID    ID   ID          |              |           | Usage |
+-----+-----+
| No running processes found |
+-----+
```

Şekil 3.6. Google Colaboratory.

Google Colab üzerinde YOLOv8 modelini çalıştırmak için Ultralytics kütüphanesinden YOLO uygulaması yüklenmiştir. Modelin eğitim sürecinde, tüm veri kümesinin model tarafından kaç kez işleneceğini belirten epoch sayısı 200 olarak belirlenmiş ve bu doğrultuda eğitim gerçekleştirilmiştir.

Hazırlamış olduğumuz modelimizde YOLOv8'in katman mimarisini ve her bir katmanın işlevlerini gösteren bilgiler Tablo 3.1'de gösterilmiş olup ilgili mimari tezimizin ikinci bölümünde ayrıntılı olarak anlatılmıştır.

Tablo 3.1. YOLOv8 Katman Mimarisi.

	from	n	param	module	arguments
0	-1	1	1392	ultralytics.nn.modules.conv.Conv	[3, 48, 3, 2]
1	-1	1	41664	ultralytics.nn.modules.conv.Conv	[48, 96, 3, 2]
2	-1	2	111360	ultralytics.nn.modules.block.C2f	[96, 96, 2, True]
3	-1	1	166272	ultralytics.nn.modules.conv.Conv	[96, 192, 3, 2]
4	-1	4	813312	ultralytics.nn.modules.block.C2f	[192, 192, 2, True]
5	-1	1	664320	ultralytics.nn.modules.conv.Conv	[192, 384, 3, 2]
6	-1	4	3248640	ultralytics.nn.modules.block.C2f	[384, 384, 4, True]
7	-1	1	1991808	ultralytics.nn.modules.conv.Conv	[384, 576, 3, 2]
8	-1	2	3985920	ultralytics.nn.modules.block.C2f	[576, 576, 2, True]
9	-1	1	831168	ultralytics.nn.modules.block.SPPF	[576, 576, 5]
10	-1	1	0	torch.nn.modules.upsampling.Upsa	[None, 2, 'nearest']
11	[-1, 6]	1	0	ultralytics.nn.modules.conv.Concat	[1]
12	-1	2	1993728	ultralytics.nn.modules.block.C2f	[960, 384, 2]
13	-1	1	0	torch.nn.modules.upsampling.Upsa	[None, 2, 'nearest']
14	[-1, 4]	1	0	ultralytics.nn.modules.conv.Concat	[1]
15	-1	2	517632	ultralytics.nn.modules.block.C2f	[576, 192, 2]
16	-1	1	332160	ultralytics.nn.modules.conv.Conv	[192, 192, 3, 2]
17	[-1, 12]	1	0	ultralytics.nn.modules.conv.Concat	[1]
18	-1	2	1846272	ultralytics.nn.modules.block.C2f	[576, 384, 2]
19	-1	1	1327872	ultralytics.nn.modules.conv.Conv	[384, 384, 3, 2]
20	[-1, 9]	1	0	ultralytics.nn.modules.conv.Concat	[1]
21	-1	2	4207104	ultralytics.nn.modules.block.C2f	[960, 576, 2]
22	[15,18,21]	1	3778012	ultralytics.nn.modules.head.Detect	[4, [192, 384, 576]]

Oluşturulan modelin 200 epoch süresince gerçekleştirilen eğitim süreci sonucunda elde edilen performans değerlendirmesi, Tablo 3.2’de sunulmuştur.

Tablo 3.2. Modelin Performans Değerlendirmesi.

Class	Images	Instances	Precision (P)	Recall (R)	mAP@ 50	mAP@50 -95
all	38	56	0.706	0.621	0.747	0.571
Truck	10	17	0.607	0.412	0.613	0.459
Tank	14	24	0.919	0.474	0.721	0.487
Air Defence	5	5	0.405	0.8	0.736	0.621
Artillery	10	10	0.895	0.8	0.918	0.715

Bu çalışmada kullanılan nesne tespiti modeli, 25,858,636 parametre ile yapılandırılmıştır. Modelin hesaplama karmaşıklığı 79.1 GFLOP (Giga Floating Point Operations Per Second) olarak belirlenmiştir.

Eğitimin ardından, modelin nesne tespiti performansı, precision (doğruluk), recall (duyarlılık), mAP@50 (mean average precision), mAP@50-95 ve F1 skoru gibi metriklerle değerlendirilmiştir [47].

Precision (doğruluk), modelin doğru tespit ettiği nesnelerin, toplam yaptığı tespitlere oranıdır (Denklem 3.1). True Positives (TP), doğru tespit edilen nesneleri, False Positives (FP) ise yanlış bir şekilde tespit edilen (aslında olmayan) nesneleri ifade etmektedir.

$$\text{Precision(Doğruluk)} = \frac{\text{True Positives(TP)}}{\text{True Positives(TP)} + \text{False Positives(FP)}} \quad (3.1)$$

Recall (duyarlılık), denklem 3.2’de gösterilmiş olup modelin doğru bir şekilde tespit ettiği nesnelerin, toplam doğru olması gereken nesnelere oranını ifade etmektedir. False Negatives (FN), tespit edilemeyen nesneleri (aslında var olan nesneler) göstermekte olup recall (duyarlılık), eksik tespitleri (False Negatives, FN) azaltmayı hedefler.

$$\text{Recall(Duyarlılık)} = \frac{\text{True Positives(TP)}}{\text{True Positives(TP)} + \text{False Negatives (FN)}} \quad (3.2)$$

mAP@50 ifadesi, modelin nesneleri tespit etmek için %50'lik bir eşik kullandığı durumu ifade eder; yani, modelin tespit ettiği nesnenin doğru olup olmadığını belirlemek için "Intersection over Union" (IoU) eşik değeri %50 olarak kabul edilir.

IoU denklem 3.3'de gösterilmiş olup modelin tespit ettiği bir nesne kutusunun (bounding box) ne kadar doğru olduğunu ölçmek için kullanılır. Gerçek kutu (ground truth) ile tahmin edilen kutunun kesişim alanının, birleşim alanına oranı olarak tanımlanır. IoU, 0 ile 1 arasında bir değer olup 1 değerine yaklaştıkça tespit başarısının arttığını ifade eder.

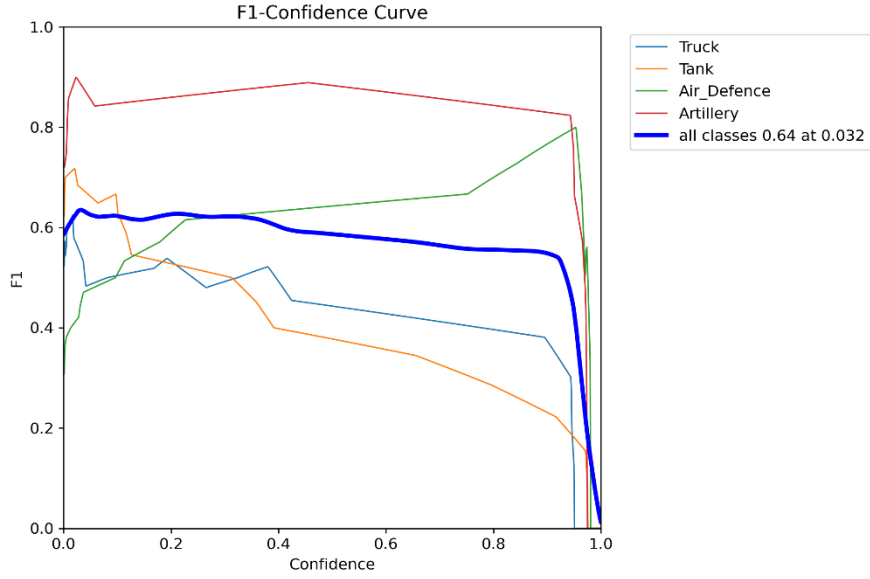
$$IoU = \frac{\text{Prediction Box} \cap \text{Ground Truth Box}}{\text{Prediction Box} \cup \text{Ground Truth Box}} \quad (3.3)$$

mAP@50-95, mAP'in, IoU eşik değeri %50'den başlayıp, %95'e kadar olan aralıkta değişen durumlar için hesaplanmasını ifade eder. Bu, modelin daha sıkı eşiklerle (IoU %50-%95) ne kadar doğru tespit yaptığını ölçen bir değerlendirmedir. Yani, IoU eşik değeri 50% ile 95% arasında değişen farklı düzeylerde modelin doğruluğu ölçülür.

F1 Skoru ise, bir modelin performansını hem Precision (doğruluk) hem de Recall (duyarlılık) metriklerini birleştirerek özetleyen bir ölçüttür. Özellikle dengesiz veri kümelerinde ve Precision ile Recall arasında bir denge kurmak gerektiğinde kullanışlıdır. F1 skoru, bu iki metriğin harmonik ortalamasını alır ve denklem 3.4'de gösterildiği şekilde tanımlanır.

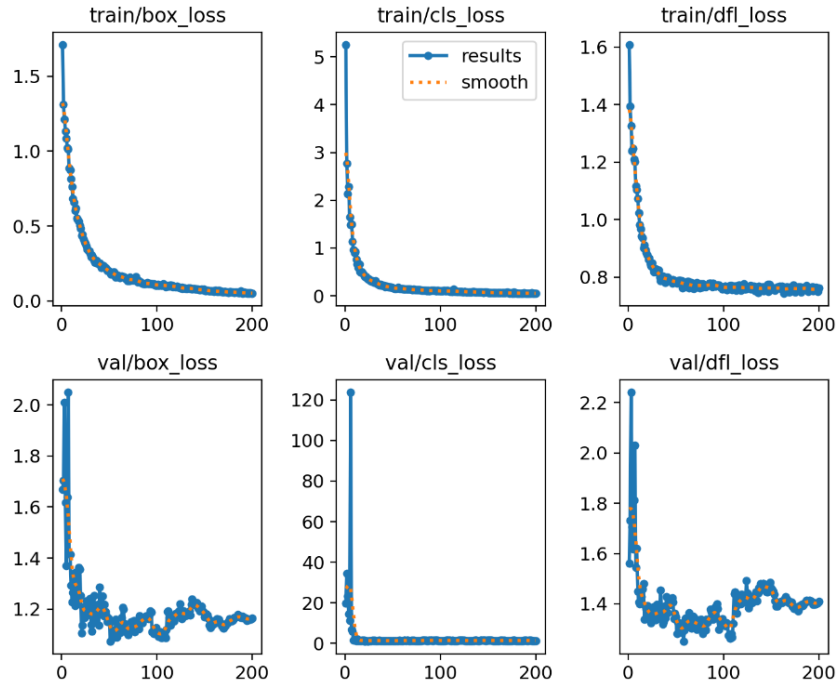
$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (3.4)$$

Modelin genel performansı incelendiğinde, precision (doğruluk) değeri 0.706, recall (duyarlılık) değeri 0.621 olarak hesaplandığı görülmüştür. Bu, modelin doğru tespit ettiği nesnelerin oranının çok yüksek olmadığını, bazı nesnelerin model tarafından gözden kaçırıldığını göstermektedir. mAP@50 değeri ise 0.747 olarak bulunmuş, bu parametre modelin genel doğruluğunun nispeten tatmin edici olduğunu ortaya koymaktadır. Öte yandan, mAP@50-95 değeri 0.571 olarak hesaplanmış, bu da modelin daha sıkı eşiklerde nesne tespitinde oldukça zorluklar yaşadığını göstermektedir. Tüm sınıflar için F1 skoru ve güven seviyesi arasındaki ilişki grafiği ise Şekil 3.7' de sunulmuş olup 0.03 güven seviyesinde F1 skoru 0.64 olmuştur.



Şekil 3.7. F1 Skoru-Güven Seviyesi Eğrisi.

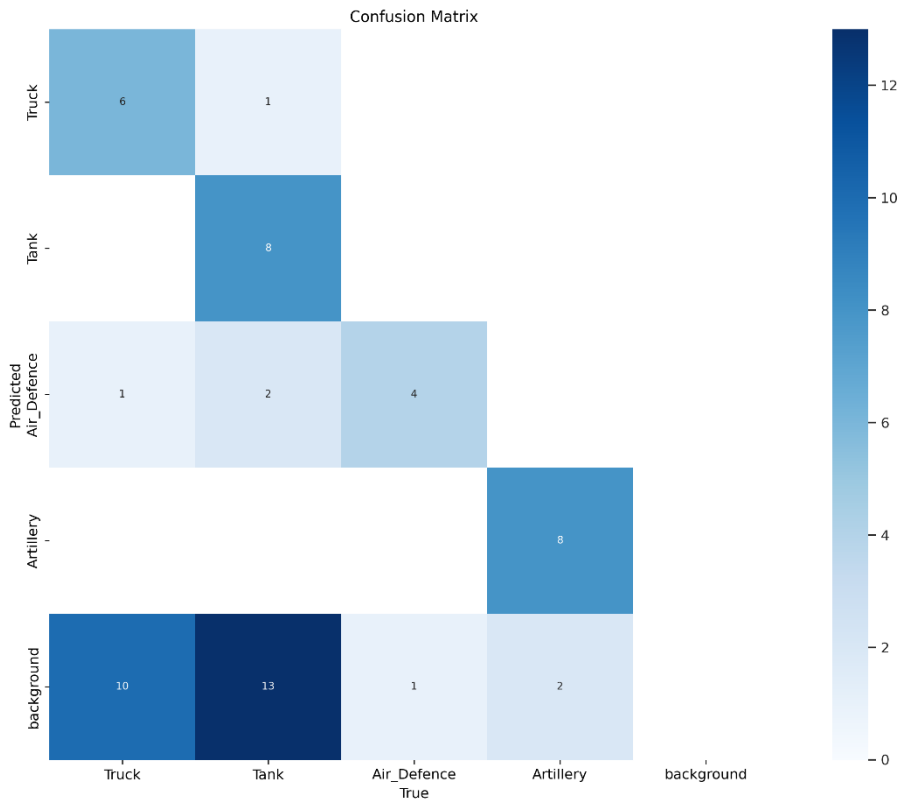
Şekil 3.8’de ise eğitilen modele ait eğitim ve doğrulama sürecine ait kayıp değerleri (loss) görselleştirilmiştir. Eğitim ve doğrulama adımlarına göre kutu kaybı yani modelin sınır kutusu tahminlerindeki kaybı (box_loss), sınıflandırma kaybı (classification_loss), modelin tahmin edilen ve gerçek sınır kutuları arasındaki dağılım farkı (distribution focal loss) (dfl_loss) değerleri gösterilmiştir.



Şekil 3.8. Eğitim ve Doğrulama Süreçlerine Ait Kayıp Değerleri.

Genel olarak, modelin performansı farklı sınıflar arasında değişkenlik göstermektedir. Artillery sınıfında başarılı bir performans sergilerken, truck sınıfında ciddi iyileştirmelere ihtiyaç duyulmaktadır.

Modelimizi karışıklık matrisi (confusion matrix), diğer bir ifade ile modelimizin tahmin ettiği sonuçlarla gerçek sonuçlar arasındaki ilişkiyi görsel olarak gösteren ve modelin doğru ve yanlış sınıflandırmalarını anlamamıza yardımcı olan karışıklık matrisi Şekil 3.9'de gösterilmiştir.



Şekil 3.9. Karışıklık Matrisi (Confusion Matrix).

Eğitilmiş modelin, doğrulama aşamasında gerçekleştirdiği tahminlerin bir kısmı, Şekil 3.10'de görsel olarak sunulmuştur. Bu görsel, modelin veri setindeki örnekler üzerinde yaptığı sınıflandırma ve tespit işlemlerinin sonuçlarını yansıtmaktadır. Şekil 3.10'de yer alan tahminler, modelin nesne tespiti veya sınıflandırma performansını gözler önüne sererken, aynı zamanda modelin doğruluk oranları, hata türleri ve genel başarısını değerlendirme açısından önemli bir referans noktası oluşturmaktadır. Görselde, modelin doğru tespit ettiği nesneler ile yanlış sınıflandırdığı ya da gözden kaçırdığı nesneler arasındaki farklar açık bir şekilde görülebilir.



Şekil 3.10. Modelimizin Örnek Tahminleri.

3.5. Veri Setinin Artırılması

Veri artırma (data augmentation), makine öğrenimi ve derin öğrenme modellerinin performansını artırmak amacıyla kullanılan bir tekniktir. Bu yöntem, mevcut veri setini manipüle ederek veya çeşitli dönüşümler uygulayarak yeni veri örnekleri oluşturmayı içerir. Veri artırma yöntemleri, derin öğrenme modellerinin veri setlerine olan ihtiyacını azaltmada ve aşırı uydurma gibi problemlerin önünü kesmede önemli bir rol oynar [48].

Veri artırmada, veri setine yapay olarak çeşitlilik kazandırılarak modelin yeni verilere daha iyi genelleşmesi ile aşırı uydurmanın azaltılması, çoğu uygulamada yeterli sayıda etiketli veri elde etmek zor olabileceği için yetersiz veri sorununun giderilmesi, daha çeşitli ve geniş bir veri seti ile modelin öğrenme kapasitesinin ve performansının artırılması amaçlanır. Görseller üzerinde kullanılan teknikler; döndürme ve yansıtma ile görsellerin belirli açılarda döndürülmesi veya yatay/dikey ekseninde yansıtılması, parlaklık ve kontrast ayarlama ile görsellerin parlaklık veya kontrast seviyelerinin değiştirilmesi, kesme ile görsellerin belirli bölgelerinin kesilmesi veya çıkarılması,

gürültü ekleme ile görsellere rastgele gürültü pikselleri ekleyerek daha gerçekçi veri oluşturulması olarak özetlenebilir.

YOLOv8 modeline, modelin performansını artırmak ve genelleme yeteneğini geliştirmek amacıyla, Tablo 3.3'te detaylandırılan çeşitli veri artırma yöntemleri entegre edilmiştir. Bu yöntemler, veri setine farklılık ve çeşitlilik kazandırarak modelin farklı koşullara uyum sağlayabilmesini hedeflemektedir. Uygulanan teknikler sayesinde modelin hem eğitim sürecinde daha etkili öğrenmesi hem de gerçek dünyadaki çeşitli durumlara karşı daha dayanıklı hale gelmesi amaçlanmıştır.

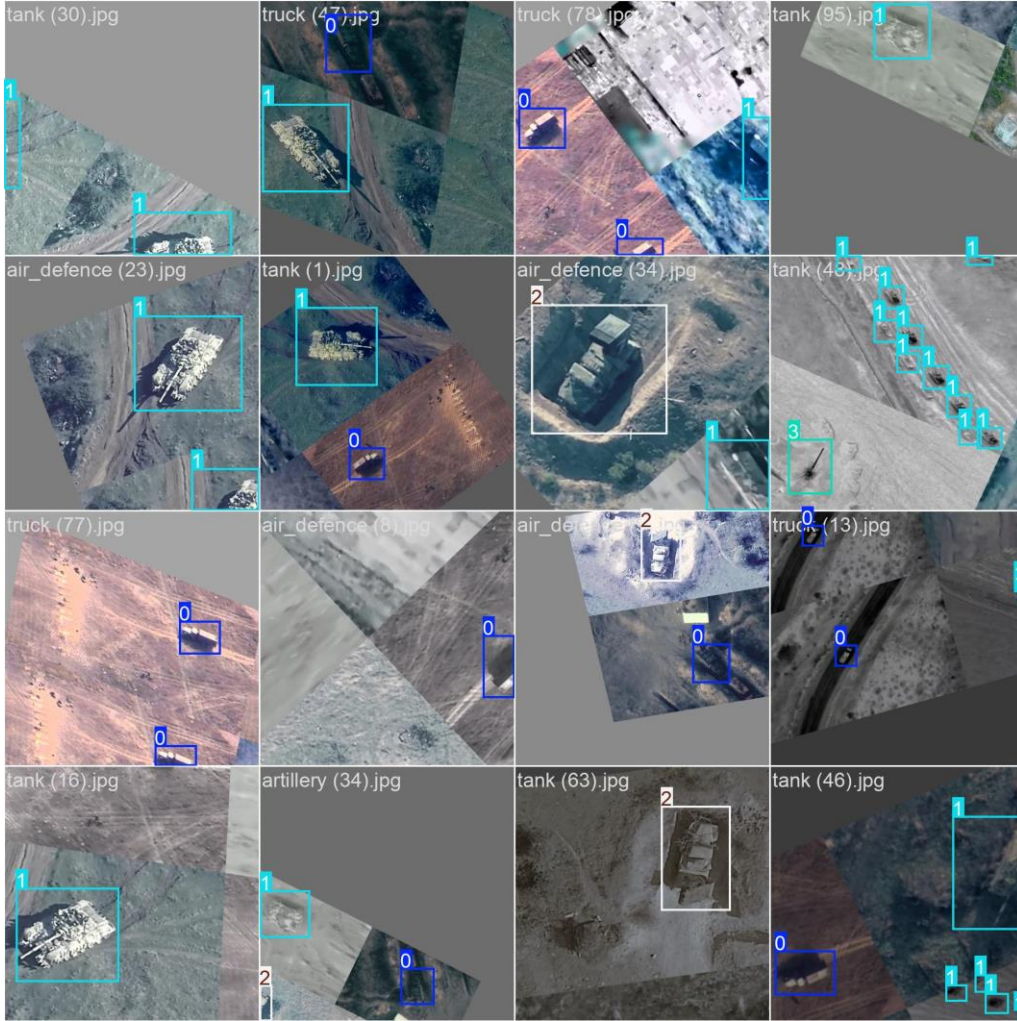
Tablo 3.3. YOLOv8 Veri Artırma Yöntemleri.

İşlem	Veri Türü	Varsayılan Değer	Aralık Değer	Açıklama
hsv_h	float	0.015	0.0 - 1.0	Görüntünün renk tonunu ayarlar, farklı aydınlatma koşullarında modelin genelleme yeteneğini artırır.
hsv_s	float	0.7	0.0 - 1.0	Görüntüdeki renklerin doygunluğunu değiştirir, farklı çevresel koşulları simüle eder.
hsv_v	float	0.4	0.0 - 1.0	Görüntü parlaklığını değiştirir, modelin çeşitli ışık koşullarında performansını iyileştirir.
degrees	float	0.0	-180 - +180	Görüntüyü belirli bir açı aralığında rastgele döndürerek, farklı yönelimlerde nesne tanımayı geliştirir.
translate	float	0.1	0.0 - 1.0	Görüntüyü yatay ve dikey olarak kaydırır, kısmen görünen nesnelerin algılanmasını öğretir.
scale	float	0.5	≥ 0.0	Görüntüyü ölçeklendirerek nesnelerin farklı mesafelerdeki görünümünü simüle eder.

Tablo 3.3. (Devamı) YOLOv8 Veri Artırma Yöntemleri.

İşlem	Veri Türü	Varsayılan Değer	Aralık Değer	Açıklama
shear	float	0.0	-180 - +180	Görüntüyü belirli bir açıyla kaydırarak farklı açılardan nesne görünümünü taklit eder.
flipud	float	0.0	0.0 - 1.0	Görüntüyü yukarıdan aşağıya çevirir, veri çeşitliliğini artırır.
fliplr	float	0.5	0.0 - 1.0	Görüntüyü yatay olarak çevirir, simetrik nesneleri tanıma kabiliyetini artırır.
bgr	float	0.0	0.0 - 1.0	RGB ve BGR kanallarını değiştirerek modelin yanlış kanal sıralamalarına dayanıklılığını artırır.
mosaic	float	1.0	0.0 - 1.0	Dört görüntüyü birleştirerek farklı sahne kompozisyonlarını ve nesne etkileşimlerini simüle eder.
mixup	float	0.0	0.0 - 1.0	İki görüntüyü ve etiketlerini birleştirir, modelin genel özellikleri öğrenmesini destekler.
copy_paste	float	0.0	0.0 - 1.0	Nesneleri farklı görüntülere yapıştırır, nesne örtüşme durumlarını öğrenmeye yardımcı olur (segmentasyon etiketleri gerekir).
erasing	float	0.4	0.0 - 0.9	Görüntünün bir bölümünü rastgele silerek modelin daha az belirgin özelliklere odaklanmasını sağlar.
crop_fraction	float	1.0	0.1 - 1.0	Görüntüyü belirli bir oranla kırpar, nesnenin merkezi özelliklerini vurgular ve ölçek adaptasyonu sağlar.

Bu kapsamda, mevcut veri seti, Tablo 3.3'te açıklanan veri artırma yöntemlerinden belirli tekniklerin uygulanması yoluyla genişletilmiş ve daha zengin bir yapıya kavuşturulmuştur. Uygulanan yöntemler, veri setindeki çeşitliliği artırarak hem modelin eğitim sürecine katkıda bulunmuş hem de farklı senaryolara karşı daha etkili bir performans göstermesine olanak tanımıştır. Bu işlemler sonucunda elde edilen nihai veri setinden seçilen örnekler ise Şekil 3.11'da görselleştirilmiş ve sunulmuştur.



Şekil 3.11. Artırılmış Veri Seti Örnek Görseller.

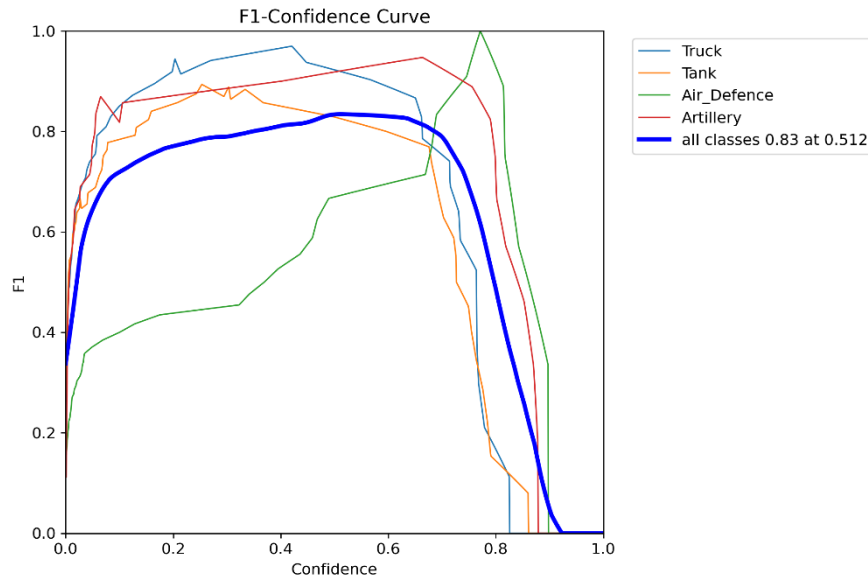
3.6. YOLOv8 ile Artırılan Veri Setinin Eğitilmesi

Yukarıda belirtilen veri artırma yöntemlerinin uygulanmasıyla oluşturulan genişletilmiş veri seti, ilk eğitimde kullanılan koşullar ve parametreler birebir korunarak yeniden eğitime tabi tutulmuştur. Bu eğitim sonucunda elde edilen performans değerlendirme sonuçları, Tablo 3.4'te sunulmuş olup artırılmış veriler ile eğitilen modelin genel doğruluk ve duyarlılık değerleri oldukça yüksektir.

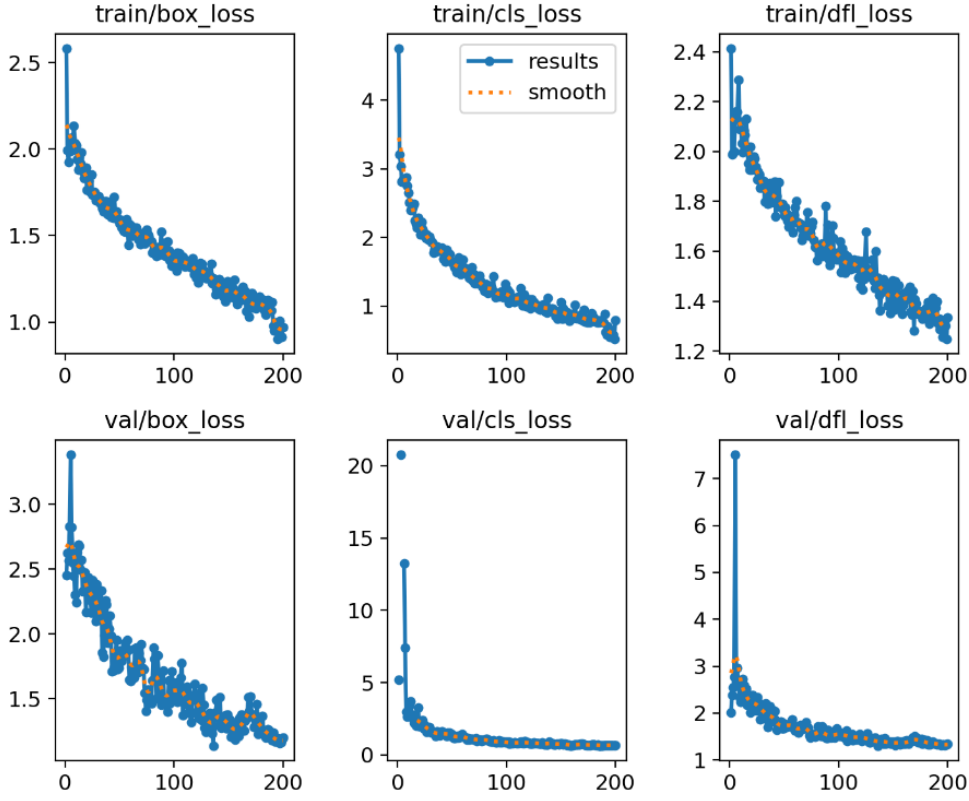
Tablo 3.4. Artırılmış Veri Seti Modelinin Performans Değerlendirmesi.

Class	Images	Instances	Precision (P)	Recall (R)	mAP@50	mAP@50-95
all	38	56	0.865	0.855	0.966	0.629
Truck	10	17	1	0.833	0.987	0.693
Tank	14	24	1	0.688	0.908	0.62
Air Defence	5	5	0.505	1	0.995	0.534
Artillery	10	10	0.956	0.9	0.972	0.666

Precision (doğruluk) %86.5 ve Recall (duyarlılık) %85.5 olarak hesaplanmıştır. Bu durum, modelin hem doğru pozitif tespitlerde güçlü olduğunu hem de büyük bir çoğunlukla gerçek pozitifleri yakalayabildiğini göstermektedir. Ortalama hassasiyet (mAP@50) değeri %96.6 gibi oldukça yüksek bir seviyededir. Bu, modelin IoU eşik değeri 0.5'te oldukça iyi bir doğruluk sergilediğini ifade eder. Ancak, daha zorlayıcı bir ölçüt olan mAP@50-95 değeri %62.9 olup, bu metrikte bir miktar düşüş gözlenmiştir. Bu, modelin daha sıkı eşiklerde performansının kısmen sınırlı olabileceğini göstermektedir. Tüm sınıflar için F1 skoru ve güven seviyesi arasındaki ilişki grafiği ise Şekil 3.12' de sunulmuş olup 0.51 güven seviyesinde F1 skoru 0.83 olmuştur.

**Şekil 3.12.** F1 Skoru-Güven Seviyesi Eğrisi.

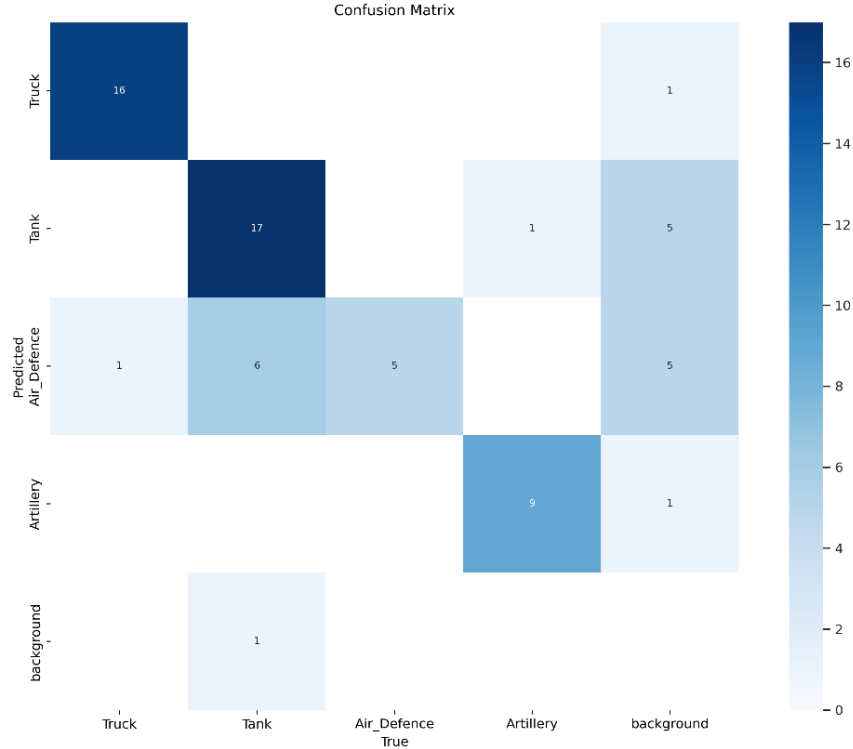
Şekil 3.13’de ise eğitilen modele ait eğitim ve doğrulama sürecine ait kayıp değerleri (loss) görselleştirilmiştir. Eğitim ve doğrulama adımlarına göre kutu kaybı yani modelin sınır kutusu (bounding box) tahminlerindeki kaybı (box_loss), sınıflandırma kaybı (classification_loss), modelin tahmin edilen ve gerçek sınır kutuları arasındaki dağılım farkı (distribution focal loss) (dfl_loss) değerleri gösterilmiştir.



Şekil 3.13. Eğitim ve Doğrulama Süreçlerine Ait Kayıp Değerleri.

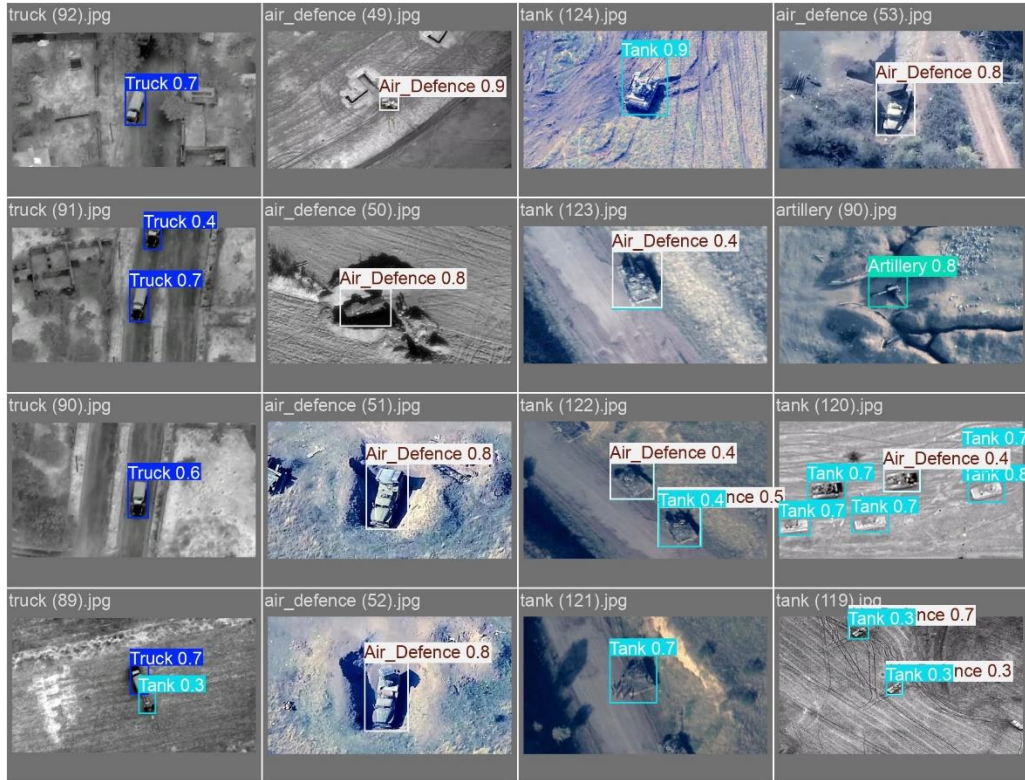
Bu sonuçlar, modelin farklı sınıflar arasında iyi bir doğruluk ve duyarlılık dengesine sahip olduğunu göstermektedir. Truck ve artillery sınıflarında modelin hem Precision hem de Recall değerleri oldukça yüksekken, air defence sınıfında Precision düşüklüğü dikkat çekmektedir. Özellikle, mAP@50'nin tüm sınıflarda yüksek olması, modelin düşük IoU eşiklerinde güçlü bir performans sergilediğini, ancak mAP@50-95 metriklerindeki düşüşler, modelin daha karmaşık tespitler için iyileştirilmesi gerektiğini göstermektedir.

Artırılmış veriler ile eğitilen modelimizin karışıklık matrisi (confusion matrix) ise Şekil 3.14’de sunulmuştur.



Şekil 3.14. Artırılmış Veri Seti Modelinin Karışıklık Matrisi.

Artırılmış veriler ile eğitilen modelin doğrulama aşamasında gerçekleştirdiği tahminlerin bir kısmı, Şekil 3.15'de görsel olarak sunulmuştur.



Şekil 3.15. Artırılmış Veri Seti Modeli Örnek Tahminleri.



4. SONUÇ VE ÖNERİLER

4.1. Sonuçların Değerlendirilmesi

Açık kaynaklardan temin edilerek oluşturulan veri setinin başlangıç hali ile bu veri setine veri artırımı uygulanmış versiyonu üzerinde gerçekleştirilen eğitimlerin genel performans metrikleri karşılaştırıldığında, anlamlı ve dikkat çekici iyileşmelerin elde edildiği görülmüştür.

İlk model eğitimi, precision (doğruluk) %70.6, recall (duyarlılık) %62.1 ve mAP@50 %74.7 değerleriyle tatmin edici bir başlangıç performansı sunmuştur. Ancak mAP@50-95 değeri %57.1 olarak nispeten düşük kalmıştır.

Artırılmış veri seti ile yapılan ikinci eğitimde ise tüm bu metriklerde kayda değer iyileşmeler sağlanmıştır. Precision %86.5, recall %85.5 ve mAP@50 %96.6 gibi oldukça yüksek değerlere ulaşılmıştır. Daha zorlu değerlendirme ölçütü olan mAP@50-95 değeri ise %62.9 ile ilk eğitime göre önemli bir artış göstermiştir. Bu sonuçlar, veri setinin güçlendirilmesi stratejisi ile modelin doğruluk ve duyarlılık kayda değer iyileştirme göstereceğini ve yapılacak yeni çalışmalar ile belirlenen yöntemin harekât sahasında rahat bir şekilde kullanılabileceği gerçeğini ortaya koymaktadır.

Ancak, bu teknolojinin başarı oranı, çevresel koşullardan önemli ölçüde etkilenebilir. Özellikle hava koşullarının kapalı olduğu durumlarda, İHA'ların görüntü işleme yeteneklerinde belirgin sınırlamalar gözlemlenmektedir. Kapalı hava koşulları, bulut örtüsünün artışı, düşük ışık seviyeleri gibi durumlar İHA kameralarından alınan görüntülerin kontrastını ve keskinliğini düşürmekte, ayrıca algılanabilir detay seviyesini sınırlandırmaktadır. Derin öğrenme algoritmalarının yüksek doğruluk oranlarına ulaşabilmesi, genellikle görüntülerin yeterli aydınlatma ve netlik seviyesine sahip olmasını gerektirir. Bununla birlikte, kapalı hava koşullarında ışığın yetersizliği, nesnelerin renk ve şekil özelliklerinin doğru bir şekilde algılanmasını zorlaştırır. Bu kapsamda, İHA'ların kapalı hava koşullarında derin öğrenme tabanlı nesne tespiti dayalı uygulamalarda etkinliği azalmaktadır.

4.2. Öneriler

Elde edilen sonuçların daha başarılı bir performans sergileyebilmesi için veri setinin çeşitlendirilmesi ve boyutunun artırılması kritik bir önem taşımaktadır. Özellikle, insansız hava araçlarından (İHA) elde edilen görüntülerin, bu çalışmaya özgü olarak yalnızca hedef sınıflara odaklanacak şekilde hazırlanması ve sembolojilerden arındırılarak yapılandırılması, modelin doğruluk ve duyarlılık metriklerini artıracaktır. Bu bağlamda, daha homojen ve kaliteli bir veri seti, modelin hedef sınıfları daha etkili bir şekilde tanımasını sağlayacaktır.

Ayrıca, veri çeşitliliği ve sayısındaki artışın, sistemin tanıyabileceği sınıf sayısını genişletmek açısından da önemli bir katkı sağlayacağı öngörülmektedir. Özellikle saha verilerinin artırılması ve gerçek dünya koşullarını daha iyi temsil eden görüntülerin modele dahil edilmesi, modelin kapasitesini güçlendirecektir. Bu durum, hem sınıf çeşitliliği hem de farklı senaryolara uyum sağlayabilme yeteneğini geliştirmek için kritik bir adımdır.

Sonuç olarak, bu tür çalışmaların, hem askeri hem de sivil uygulamalar için İHA'ların kullanımında otoritelere karar verme süreçlerinde önemli kolaylıklar sunacağı değerlendirilmektedir. Özellikle, İHA'ların otonom olarak görev yaptığı durumlarda, geliştirilen modellerin veri çeşitliliği ile güçlendirilmesi, bu araçların bağımsız ve doğru karar alabilme yeteneklerini artıracaktır. Bu bağlamda, gelecekte yapılacak çalışmalarda sahadan toplanan veri miktarının artırılması ve bu verilerin sistematik bir şekilde hazırlanması, model performansı ve kullanım alanlarının genişletilmesi açısından önem arz etmektedir.

KAYNAKLAR

- [1] Türk Hava Kurumu, “Türk Hava Kurumu - İHA”. Erişim: 09 Aralık 2024. [Çevrimiçi]. Erişim adresi: <https://www.thk.org.tr/iha>
- [2] Vikipedi, “İnsansız Hava Aracı”. Erişim: 09 Aralık 2024. [Çevrimiçi]. Erişim adresi: https://tr.wikipedia.org/wiki/%C4%B0nsans%C4%B1z_hava_arac%C4%B1
- [3] “Baykar Teknoloji”. Erişim: 18 Aralık 2024. [Çevrimiçi]. Erişim adresi: <https://baykartech.com/tr/>
- [4] “TUSAŞ”. Erişim: 18 Aralık 2024. [Çevrimiçi]. Erişim adresi: <https://www.tusas.com/>
- [5] STM, “Derin Farklar Yapay Zekâ, Makine Öğrenmesi ve Derin Öğrenme”.
- [6] W. S. McCulloch ve W. Pitts, “A Logical Calculus of the Ideas Immanent in Nervous Activity”, 1990.
- [7] J. Schmidhuber, “Deep Learning in Neural Networks: An Overview”, Nis. 2014, doi: 10.1016/j.neunet.2014.09.003.
- [8] C. M. Bishop, *Pattern Recognition and Machine Learning*. 2006.
- [9] A. Turing, “Computing Machinery and Intelligence”, *Mind*, c. 49, ss. 433-460, 1950.
- [10] F. Rosenblatt, “The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain”, *Psychol Rev*, c. 65, sy 6, ss. 386-408, 1958.
- [11] T. Mitchell, *Machine Learning*. 1997.
- [12] G. E. Hinton, S. Osindero, ve Y.-W. Teh, “A Fast Learning Algorithm for Deep Belief Nets”.
- [13] A. Krizhevsky, I. Sutskever, ve G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks”, [Çevrimiçi]. Erişim adresi: <http://code.google.com/p/cuda-convnet/>
- [14] K. He, X. Zhang, S. Ren, ve J. Sun, “Deep Residual Learning for Image Recognition”, *Microsoft Research*, Ara. 2015, [Çevrimiçi]. Erişim adresi: <http://arxiv.org/abs/1512.03385>
- [15] A. Zafar vd., “A Comparison of Pooling Methods for Convolutional Neural Networks”, 01 Eylül 2022, *MDPI*. doi: 10.3390/app12178643.
- [16] S. Hochreiter ve J. Schmidhuber, “Long Short-Term Memory”, *Neural Comput*, c. 9, sy 8, ss. 1735-1780, Kas. 1997, doi: 10.1162/neco.1997.9.8.1735.
- [17] A. Vaswani vd., “Attention Is All You Need”, Haz. 2017, [Çevrimiçi]. Erişim adresi: <http://arxiv.org/abs/1706.03762>

- [18] J. Devlin, M.-W. Chang, K. Lee, ve K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”, May. 2019, [Çevrimiçi]. Erişim adresi: <http://arxiv.org/abs/1810.04805>
- [19] I. J. Goodfellow vd., “Generative Adversarial Nets”. [Çevrimiçi]. Erişim adresi: <http://www.github.com/goodfeli/adversarial>
- [20] R. Girshick, J. Donahue, T. Darrell, ve J. Malik, “Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation”, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, ss. 580-587, 2014.
- [21] “The Fundamental Guide to Faster R-CNN”. Erişim: 18 Aralık 2024. [Çevrimiçi]. Erişim adresi: <https://viso.ai/deep-learning/faster-r-cnn-2/>
- [22] R. Girshick, “Fast R-CNN”, *IEEE International Conference on Computer Vision (ICCV)*, ss. 1440-1448, 2015, [Çevrimiçi]. Erişim adresi: <http://arxiv.org/abs/1504.08083>
- [23] S. Ren, K. He, R. Girshick, ve J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, *Advances in Neural Information Processing Systems (NIPS)*, ss. 91-99, 2015, [Çevrimiçi]. Erişim adresi: <http://arxiv.org/abs/1506.01497>
- [24] K. He, G. Gkioxari, P. Dollár, ve R. Girshick, “Mask R-CNN”, *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, ss. 2961-2969, 2017, [Çevrimiçi]. Erişim adresi: <http://arxiv.org/abs/1703.06870>
- [25] J. Redmon, S. Divvala, R. Girshick, ve A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection”, Haz. 2015, [Çevrimiçi]. Erişim adresi: <http://arxiv.org/abs/1506.02640>
- [26] Ultralytics, “YOLOv11 - Ultralytics YOLO Docs”. Erişim: 22 Ocak 2025. [Çevrimiçi]. Erişim adresi: <https://docs.ultralytics.com/tr/models/yolo11/>
- [27] G. Yao, S. Zhu, L. Zhang, ve M. Qi, “HP-YOLOv8: High-Precision Small Object Detection Algorithm for Remote Sensing Images”, *Sensors*, c. 24, sy 15, Ağu. 2024, doi: 10.3390/s24154858.
- [28] C. Gonçalves, “Introduction to YOLOv8”. Erişim: 09 Aralık 2024. [Çevrimiçi]. Erişim adresi: <https://medium.com/@camilolgon/introduction-to-yolov8-8ce08589d95d>
- [29] N. V. Kim ve M. A. Chervonenkis, “Situation control of unmanned aerial vehicles for road traffic monitoring”, *Mod Appl Sci*, c. 9, sy 5, ss. 1-13, 2015, doi: 10.5539/mas.v9n5p1.
- [30] B. Benjdira, T. Khursheed, A. Koubaa, A. Ammar, ve K. Ouni, “Car Detection using Unmanned Aerial Vehicles: Comparison between Faster R-CNN and YOLOv3”, Ara. 2018, [Çevrimiçi]. Erişim adresi: <http://arxiv.org/abs/1812.10968>
- [31] S. Toraman, “Derin Öğrenme ile İnsansız Hava Aracı Görüntülerinden Yaya Tespiti”, *Journal of Aviation*, c. 2, sy 2, ss. 64-69, Ara. 2018, doi: 10.30518/jav.450913.

- [32] B. Şin, İ. Kadioğlu, T. Gaziosmanpaşa, Ü. Ziraat, F. Bitki, ve K. Bölümü, “İnsansız Hava Aracı (İHA) ve Görüntü İşleme Teknikleri Kullanılarak Yabancı Ot Tespitinin Yapılması”, 2019. [Çevrimiçi]. Erişim adresi: <https://dergipark.org.tr/tjws>
- [33] V. Şahin ve H. M. Yılmaz, “Hacim Hesaplarında İnsansız Hava Aracı (İHA) Verilerinin Kullanılabilirliğinin Araştırılması”, *Türkiye İnsansız Hava Araçları Dergisi*, c. 3, sy 2, ss. 36-48, Ara. 2021, doi: 10.51534/tiha.955271.
- [34] B. Aksoy, K. Korucu, Ö. Çalışkan, Ş. Osmanbey, ve H. D. Halis, “İnsansız Hava Aracı ile Görüntü İşleme ve Yapay Zekâ Teknikleri Kullanılarak Yangın Tespiti: Örnek Bir Uygulama”, *Düzce Üniversitesi Bilim ve Teknoloji Dergisi*, c. 9, sy 6, ss. 112-122, Ara. 2021, doi: 10.29130/dubited.1016195.
- [35] L. A. Silva, V. R. Q. Leithardt, V. F. L. Batista, G. Villarrubia Gonzalez, ve J. F. De Paz Santana, “Automated Road Damage Detection Using UAV Images and Deep Learning Techniques”, *IEEE Access*, c. 11, ss. 62918-62931, 2023, doi: 10.1109/ACCESS.2023.3287770.
- [36] D. Kumbasar, “Arkeolojik Alanlardaki İHA Görüntülerinin Derin Öğrenme Yardımıyla Anlamsal Segmentasyonu”, 2023.
- [37] T. Çobanoğlu, “Derin Öğrenmeye Dayalı Yapay Zekâ Tekniği ile Askeri Uçak Tiplerinin Tespit Edilmesi ve Sınıflandırılması”, 2023.
- [38] M. S. Türker, “Derin Öğrenme Yöntemleri ile Uydu Görüntülerinden Gemilerin Tespiti ve Sınıflandırılması”, 2024.
- [39] U. Çimen, “Hava Araçlarının Sınıflandırılmasında Derin Öğrenme Tekniklerinin Performans Analizi”, 2024.
- [40] S. Doğan, “Derin Öğrenme Yöntemleri ile İnsansız Hava Araçlarının Tespit Edilmesi”, 2024.
- [41] V. K. Çetinkaya, “Derin Öğrenme Temelli Havadan Havaya İnsansız Hava Aracı Tespiti”, 2024.
- [42] R. Daş, B. Polat, ve G. Tuna, “Derin Öğrenme ile Resim ve Videolarda Nesnelerin Tanınması ve Takibi”, 2019.
- [43] “CapCut”. Erişim: 15 Aralık 2024. [Çevrimiçi]. Erişim adresi: <https://www.capcut.com/tr-tr/>
- [44] “Fotor”. Erişim: 15 Aralık 2024. [Çevrimiçi]. Erişim adresi: <https://www.fotor.com/>
- [45] “GitHub - HumanSignal/labelImg”. Erişim: 15 Aralık 2024. [Çevrimiçi]. Erişim adresi: <https://github.com/HumanSignal/labelImg>
- [46] “Google Colaboratory”. Erişim: 16 Aralık 2024. [Çevrimiçi]. Erişim adresi: <https://colab.research.google.com/>
- [47] D. Shah, “Mean Average Precision (mAP) Explained: Everything You Need to Know”. Erişim: 20 Ocak 2025. [Çevrimiçi]. Erişim adresi: <https://www.v7labs.com/blog/mean-average-precision>
- [48] C. Shorten ve T. M. Khoshgoftaar, “A survey on Image Data Augmentation for Deep Learning”, *J Big Data*, c. 6, sy 1, Ara. 2019, doi: 10.1186/s40537-019-0197-0.



ÖZGEÇMİŞ

Ad-Soyad : Selçuk ÇİFTÇİ

TEZDEN TÜRETİLEN ESERLER:

- Çiftçi, S. ve Çallı, F. 2025. AI-Driven Object Detection for Military UAV Applications, Sakarya University Journal of Computer and Information Sciences, (In Review)