

T.C.
AKDENİZ ÜNİVERSİTESİ



UYDU GÖRÜNTÜLERİNDEN UÇAK TİPİ TESPİTİNDE GENETİK
ALGORİTMA TABANLI HİPERPARAMETRE OPTİMİZASYONUNUN YOLO
MODELİNE ETKİSİNİN ARAŞTIRILMASI

Mustafa Candoğdu ÇUTUR

FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ
ANABİLİM DALI

YÜKSEK LİSANS TEZİ

EYLÜL 2025
ANTALYA

T.C.
AKDENİZ ÜNİVERSİTESİ



UYDU GÖRÜNTÜLERİNDEN UÇAK TİPİ TESPİTİNDE GENETİK
ALGORİTMA TABANLI HİPERPARAMETRE OPTİMİZASYONUNUN YOLO
MODELİNE ETKİSİNİN ARAŞTIRILMASI

Mustafa Candoğdu ÇUTUR

FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ
ANABİLİM DALI

YÜKSEK LİSANS TEZİ

EYLÜL 2025
ANTALYA

T.C.
AKDENİZ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

UYDU GÖRÜNTÜLERİNDEN UÇAK TİPİ TESPİTİNDE GENETİK
ALGORİTMA TABANLI HİPERPARAMETRE OPTİMİZASYONUNUN YOLO
MODELİNE ETKİSİNİN ARAŞTIRILMASI

Mustafa Candoğdu ÇUTUR

ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ
ANABİLİM DALI

YÜKSEK LİSANS TEZİ

EYLÜL 2025

T.C.
AKDENİZ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

UYDU GÖRÜNTÜLERİNDEN UÇAK TİPİ TESPİTİNDE GENETİK
ALGORİTMA TABANLI HİPERPARAMETRE OPTİMİZASYONUNUN YOLO
MODELİNE ETKİSİNİN ARAŞTIRILMASI

Mustafa Candoğdu ÇUTUR

ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ
ANABİLİM DALI

YÜKSEK LİSANS TEZİ

Bu tez 19/09/2025 tarihinde jüri tarafından oybirliği ile kabul edilmiştir.

Prof. Dr. Övünç POLAT (Danışman)

[imza]

Prof. Dr. Süleyman BİLGİN

[imza]

Doç. Dr. Serdar GÜNDOĞDU

[imza]

ÖZET

UYDU GÖRÜNTÜLERİNDEN UÇAK TİPİ TESPİTİNDE GENETİK ALGORİTMA TABANLI HİPERPARAMETRE OPTİMİZASYONUNUN YOLO MODELİNE ETKİSİNİN ARAŞTIRILMASI

Mustafa Candoğdu ÇUTUR

Yüksek Lisans Tezi, Elektrik Elektronik Mühendisliği Anabilim Dalı

Danışman: Prof. Dr. Övünç POLAT

Eylül 2025, 53 sayfa

Tez çalışmasında, uydu görüntülerinde yer alan hava araçlarının tespitinde kullanılan derin öğrenme tabanlı bir yöntem ele alınmış ve farklı eğitim stratejilerinin model performansı üzerindeki etkileri incelenmiştir. Nesnelerin uydu görüntülerinden yüksek doğrulukla tanınması, hem sivil hem de askerî alanlarda önem taşıyan güncel bir araştırma konusudur. Bu amaçla çalışmada YOLOv5 mimarisi tercih edilmiş ve görsel olarak benzer özellikler taşıyan ancak farklı görevlerde kullanılan üç farklı uçak tipi ele alınmıştır. Bu sınıflar, askeri savaş uçağı, sivil yolcu uçağı ve askeri kargo uçağından oluşmaktadır. Çalışmada kullanılan veri seti, literatürde yaygın olarak kullanılan hazır veri setleri yerine Google Earth Pro aracılığıyla dünyanın çeşitli bölgelerinde bulunan farklı havaalanlarından manuel olarak toplanan uydu görüntülerinden oluşturulmuş ve YOLO formatında etiketlenmiştir.

Eğitim süreci üç farklı deneysel çalışma kapsamında yürütülmüştür. İlk aşamada, herhangi bir değişiklik yapılmadan hazır tanımlı hiperparametreler kullanılmış ve bu bölüm sonraki çalışmalar için başlangıç noktası olarak değerlendirilmiştir. İkinci aşamada, modelin genelleme yeteneğini geliştirmek amacıyla çeşitli veri artırımı tekniklerinden yararlanılmıştır. Son aşamada ise veri artırımı uygulamalarına ek olarak genetik algoritma tabanlı hiperparametre optimizasyonu gerçekleştirilmiştir. Veri setinin sınırlı sayıda örnek içermesi nedeniyle mekânsal ve fotometrik dönüşümlerden yararlanılarak çeşitlilik artırılmış, bu sayede modelin genelleme performansı güçlendirilmiştir. Genetik algoritma tabanlı hiperparametre optimizasyonu ise öğrenme sürecini daha verimli hale getirmiştir.

Uygulanan bu üç deneysel çalışmanın karşılaştırılması sonucunda, en yüksek performans genetik algoritma tabanlı hiperparametre optimizasyonu ve veri artırımı yöntemlerinin birlikte kullanıldığı senaryoda elde edilmiştir. Bu senaryoda mAP@0.5 değeri %75.5'ten %89.9'a yükselmiş (+14.4 puan). Benzer şekilde, mAP@0.5:0.95 değeri %45.3'ten %60.2'ye çıkarak (+14.9 puan). Kesinlik (precision) değeri %71.6'dan %85.3'e yükselmiş (+13.7 puan), duyarlılık (recall) değeri %70.5'ten %83.6'ya yükselmiş (+13.1

puan) ve F1-skoru (F1-skor) değeri %70.6'dan %84.4'e yükselmiş (+13.8 puan). Elde edilen bulgular, modelin öğrenmesini zorlaştıracak ölçüde yetersiz sayıda örnek görüntü içeren veri kümeleri üzerinde gerçekleştirilen nesne tespiti çalışmalarında veri artırımı ile hiperparametre optimizasyonunun birlikte kullanılmasının model performansı açısından önemli katkılar sağladığını göstermektedir. Bu nedenle uygulanan bu yaklaşım, hem sivil hem de askerî uygulamalarda kullanılabilecek bir seçenek olarak değerlendirilmektedir.

ANAHTAR KELİMELEER: Derin öğrenme, Genetik algoritma, Hiperparametre optimizasyonu, Nesne tespiti, Uydu görüntüleri, Veri artırımı, Yapay zeka, YOLOv5

JÜRİ: Prof. Dr. Övünç POLAT

Prof. Dr. Süleyman BİLGİN

Doç. Dr. Serdar GÜNDOĞDU

ABSTRACT

INVESTIGATION OF THE EFFECT OF GENETIC ALGORITHM-BASED HYPERPARAMETER OPTIMIZATION ON THE YOLO MODEL FOR AIRCRAFT TYPE DETECTION FROM SATELLITE IMAGES

Mustafa Candoğdu ÇUTUR

MSc Thesis in Electrics and Electronics Engineering

Supervisor: Prof. Dr. Övünç POLAT

September 2025, 53 pages

In this thesis, a deep learning–based approach was employed for the detection of aircraft in satellite imagery, and the effects of different training strategies on model performance were investigated. The accurate detection of objects from satellite images constitutes a current research topic of significance in both civilian and military domains. For this purpose, the YOLOv5 architecture was adopted, focusing on three types of aircraft that share visually similar features but serve different functions. These classes consist of military fighter aircraft, civilian passenger aircraft, and military cargo aircraft. Unlike commonly used benchmark datasets in the literature, the dataset utilized in this study was created by manually collecting satellite images of various airports worldwide through Google Earth Pro and labeling them in YOLO format.

The training process was conducted through three experimental studies. In the first stage, predefined hyperparameters were employed without modification, serving as a baseline for subsequent experiments. In the second stage, various data augmentation techniques were applied to enhance the model’s generalization capability. In the final stage, genetic algorithm–based hyperparameter optimization was implemented alongside data augmentation. Since the dataset contained a limited number of samples, spatial and photometric transformations were employed to increase diversity, thereby strengthening the generalization ability of the model. Furthermore, genetic algorithm–based hyperparameter optimization contributed to a more efficient training process.

The comparison of the three experimental studies revealed that the highest performance was achieved in the scenario where genetic algorithm–based hyperparameter optimization was combined with data augmentation. In this case, the mAP@0.5 increased from 75.5% to 89.9% (+14.4 points), while the mAP@0.5:0.95 improved from 45.3% to 60.2% (+14.9 points). Precision rose from 71.6% to 85.3% (+13.7 points), recall from 70.5% to 83.6% (+13.1 points), and F1-skor from 70.6% to 84.4% (+13.8 points). The findings demonstrate that, in object detection tasks conducted on datasets with limited

sample sizes that challenge the learning process, the combined use of data augmentation and hyperparameter optimization provides significant improvements in model performance. Therefore, the proposed approach is regarded as a viable option for both civilian and military applications.

KEYWORDS: Artificial intelligence, Data augmentation, Deep learning, Genetic algorithm, Hyperparameter optimization, Object detection, Satellite images, YOLOv5

COMMITTEE: Prof. Dr. Övünç POLAT

Prof. Dr. Süleyman BİLGİN

Assoc. Prof. Dr. Serdar GÜNDOĞDU



ÖNSÖZ

Bu tez kapsamında uydu görüntülerinden nesne tespiti gerçekleştirebilen bir derin öğrenme modeli uygulanmış ve modelin başarımını artırmak amacıyla genetik algoritma tabanlı hiperparametre optimizasyonu yöntemlerinden yararlanılmıştır. Ayrıca modelin eğitim sürecinde destekleyici bir yaklaşım olarak veri artırımı tekniklerinden de faydalanılmıştır. Yöntem seçiminden değerlendirme sürecine kadar her aşamada karşılaşılan zorluklar, teknik bilgi birikiminin yanı sıra sabır, kararlılık ve dikkatli çalışma disiplini-
min gelişmesine de katkı sağlamıştır.

Çalışmanın her aşamasında bilgi ve tecrübeleriyle yol gösteren, desteğini hiçbir zaman esirgemeyen değerli danışmanım Sayın Prof. Dr. Övünç Polat’a teşekkür ederim.

Bu süreç boyunca yanımda olan, sabrını ve sevgisini her zaman hissettiren eşim Tansu Çutur’a ve ilham kaynağım olan kızım Öykü Çutur’a en içten teşekkürlerimi sunarım. Onların varlığı bu çalışmayı tamamlamamdaki en büyük dayanak olmuştur.

Bu çalışmanın nesne tespiti ve yapay zekâ alanındaki araştırmalara katkı sağlama-sını ve ileride yapılacak çalışmalara faydalı bir kaynak oluşturmasını temenni ederim.

İÇİNDEKİLER

ÖZET	i
ABSTRACT	iii
ÖNSÖZ	v
AKADEMİK BEYAN	viii
SİMGELER VE KISALTMALAR	ix
ŞEKİLLER DİZİNİ	x
ÇİZELGELER DİZİNİ	xii
1. GİRİŞ	1
2. KAYNAK TARAMASI	2
2.1. Uydu Görüntülerinde Nesne Tespiti	2
2.2. Makine Öğrenmesi Temelli Yaklaşımlar	3
2.2.1. Denetimli Öğrenme (Supervised Learning, SL)	4
2.2.2. Denetimsiz Öğrenme (Unsupervised Learning, UL)	4
2.2.3. Yarı Denetimli Öğrenme (Semi-Supervised Learning, SSL)	5
2.2.4. Pekiştirmeli Öğrenme (Reinforcement Learning, RL)	5
2.3. Derin Öğrenme ve CNN Tabanlı Yöntemler	6
2.3.1. Derin Öğrenmenin Temelleri	6
2.3.2. Konvolüsyonel Sinir Ağları (CNN)	7
2.3.3. Uydu Görüntülerinde CNN Uygulamaları	7
2.4. YOLO Algoritmaları	8
2.4.1. YOLO v3	8
2.4.2. YOLO v5	10
2.5. Genetik Algoritmalar ile Hiperparametre Optimizasyonu	13
2.6. Nesne Tespitinde Kullanılan Diğer Derin Öğrenme Modelleri	15
2.6.1. Faster R-CNN (Faster Region-Based CNN)	15
2.6.2. RetinaNet (Focal Loss Based One-Stage Object Detector)	16
2.6.3. SSD (Single Shot MultiBox Detector)	17
2.6.4. YOLOv5'in Diğer Mimari Yapılarla Karşılaştırılması	18
3. MATERYAL ve METOT	19
3.1. Veri Setinin Hazırlanması	19
3.1.1. Veri Toplama ve Ön İşleme	19
3.1.2. Etiketleme Formatı (YOLO)	20
3.1.3. Veri Artırımı	21
3.2. Kullanılan Yazılım ve Donanım Ortamı	24
3.2.1. Donanım Özellikleri	24
3.2.2. Yazılım Altyapısı ve Kütüphaneler	24
3.3. Model Eğitim Süreci	26
3.3.1. YOLOv5 Mimarisinin Yapılandırılması	26
3.3.2. Genetik Algoritma ile Hiperparametre Optimizasyonu	27
3.3.3. Eğitim ve Doğrulama	30

3.4.	Performans Değerlendirme Kriterleri	32
3.4.1.	mAP (mean Average Precision) Hesaplaması	33
4.	BULGULAR	34
4.1.	Eğitim Aşamaları ve Sonuçları	34
4.1.1.	YOLOv5 Varsayılan Ayarlar ile Model Eğitimi	34
4.1.2.	YOLOv5 ile Veri Artırımı Uygulanan Eğitim	36
4.1.3.	YOLOv5 ile Veri Artırımı ve Optimizasyon Uygulanan Eğitim	38
4.2.	Deney Sonuçlarının Karşılaştırılması	40
4.2.1.	Sınıf Bazında Metrik Sonuçlarının Karşılaştırılması	41
5.	TARTIŞMA	43
5.1.	Varsayılan Parametrelerin Sınırlılıkları	43
5.2.	Veri Artırımı ile Sağlanan Katkılar	43
5.3.	Genetik Algoritmanın Model Performansına Katkısı	43
5.4.	Sonuçların Literatür ile Karşılaştırılması	44
5.5.	Uygulama Sürecinde Karşılaşılan Sınırlılıklar	45
5.6.	Sonuçların Genellenebilirliği Açısından Değerlendirme	45
6.	SONUÇLAR	47
7.	KAYNAKLAR	49
ÖZGEÇMİŞ		

AKADEMİK BEYAN

Yüksek Lisans Tezi olarak sunduğum “UYDU GÖRÜNTÜLERİNDEN UÇAK TİPİ TESPİTİNDE GENETİK ALGORİTMA TABANLI HİPERPARAMETRE OPTİMİZASYONUNUN YOLO MODELİNE ETKİSİNİN ARAŞTIRILMASI” adlı bu çalışmanın, akademik kurallar ve etik değerlere uygun olarak yazıldığını belirtir, bu tez çalışmasında bana ait olmayan tüm bilgilerin kaynağını gösterdiğimi beyan ederim.

19/09/2025

Mustafa Candoğdu ÇUTUR

İMZA

SİMGELER VE KISALTMALAR

Kısaltmalar

AI	: Artificial Intelligence
AP	: Average Precision
AWS	: Amazon Web Services
BCE	: Binary Cross Entropy
Bi-FPN	: Bi-Directional Feature Pyramid Network
CA	: Coordinate Attention
CE	: Cross Entropy
CIoU	: Complete Intersection over Union
CNN	: Convolutional Neural Network
CPU	: Central Processing Unit
CSP	: Cross Stage Partial
CUDA	: Compute Unified Device Architecture
cuDNN	: CUDA Deep Neural Network library
CV	: Cross Validation
DA	: Data Augmentation
DBSCAN	: Density-Based Spatial Clustering of Applications with Noise
DIOR	: Dataset for Object Detection in Optical Remote Sensing Images
DOTA	: Dataset for Object Detection in Aerial Images
Faster R-CNN	: Faster Region-Based Convolutional Neural Network
FL	: Focal Loss
FN	: False Negative
FP	: False Positive
FPN	: Feature Pyramid Network
FPS	: Frames Per Second
GA	: Genetic Algorithm
GAN	: Generative Adversarial Network
GMM	: Gaussian Mixture Model
GPU	: Graphics Processing Unit
IoU	: Intersection over Union
mAP	: Mean Average Precision

MXNet	: Mobile eXecution Network
NMS	: Non-Maximum Suppression
NumPy	: Numerical Python
NVMe	: Non-Volatile Memory Express
OneCycleLR	: One-Cycle Learning Rate Policy
OS	: Operating System
PANet	: Path Aggregation Network
PNG	: Portable Network Graphics
RAM	: Random Access Memory
ReLU	: Rectified Linear Unit
ResNet	: Residual Network
RL	: Reinforcement Learning
RoI	: Region of Interest
RPN	: Region Proposal Network
R-CNN	: Region-Based Convolutional Neural Network
RSOD	: Remote Sensing Object Detection
SEB-YOLO	: Small-object Enhanced Bi-FPN YOLO
SGD	: Stochastic Gradient Descent
SIoU	: Scylla Intersection over Union
SiLU	: Sigmoid Linear Unit
SL	: Supervised Learning
SPDF-Conv	: Spatial Pyramid Dilated Convolution
SPPF	: Spatial Pyramid Pooling Fast
SSD	: Single Shot MultiBox Detector
SSL	: Semi-Supervised Learning
TP	: True Positive
UL	: Unsupervised Learning
VGG-16	: Visual Geometry Group 16-layer CNN
VHR	: Very High Resolution
VS Code	: Visual Studio Code
YOLO	: You Only Look Once
YOLOv3	: You Only Look Once version 3
YOLOv5	: You Only Look Once version 5

ŞEKİLLER DİZİNİ

Şekil 2.1.	a) Basit sinir ağı; b) derin sinir ağı mimarisi (İbrahim vd. 2021) . .	6
Şekil 2.2.	YOLOv3 mimarisi ve Darknet-53 omurga yapısı (Mao vd. 2019) .	9
Şekil 2.3.	Darknet-53 yapısı (Redmon ve Farhadi 2018)	9
Şekil 2.4.	YOLOv5 mimari yapısı (Jiang vd. 2023)	10
Şekil 2.5.	Genetik algoritma ile hiperparametre optimizasyonu akış şeması .	14
Şekil 2.6.	Faster R-CNN blok diyagramı (Chen vd. 2022).	15
Şekil 2.7.	SSD mimarisinin VGG-16 tabanlı blok diyagramı (Liu vd. 2016). .	17
Şekil 3.1.	Veri setinden seçilmiş örnek uydu görüntüleri	19
Şekil 3.2.	LabelImg arayüzünde etiketlenmiş bir uydu görüntüsü örneği . . .	20
Şekil 3.3.	Veri artırımı ile oluşturulan örnek görüntüler	23
Şekil 4.1.	Deney 1 için nesne tespit çıktılarına ait örnek görseller	36
Şekil 4.2.	Deney 2 için nesne tespit çıktılarına ait örnek görseller	38
Şekil 4.3.	Deney 3 için nesne tespit çıktılarına ait örnek görseller	40

ÇİZELGELER DİZİNİ

Çizelge 2.1.	Uçak tespitinde model performansları (El Ghazouali vd. 2024) .	18
Çizelge 3.1.	YOLO formatında etiket yapısı	20
Çizelge 3.2.	Sınıf etiketleri ve karşılıkları	21
Çizelge 3.3.	YOLOv5 varsayılan veri artırımı parametreleri	22
Çizelge 3.4.	Albumentations kütüphanesi veri artırımı parametreleri	22
Çizelge 3.5.	Veri artırımı öncesi ve sonrası görüntü sayıları	23
Çizelge 3.6.	Donanım özellikleri	24
Çizelge 3.7.	YOLOv5 başlangıç hiperparametreleri	27
Çizelge 3.8.	GA hiperparametre arama uzayı	28
Çizelge 3.9.	Genetik algoritma konfigürasyon parametreleri	29
Çizelge 3.10.	GA ile optimize edilmiş hiperparametreler	30
Çizelge 3.11.	5 katlı çapraz doğrulama iterasyonları	31
Çizelge 3.12.	Performans değerlendirme metrikleri	32
Çizelge 4.1.	Deney 1 için ortalama metrik sonuçları	35
Çizelge 4.2.	Deney 1 için sınıf bazında ortalama metrik sonuçları	35
Çizelge 4.3.	Deney 2 için ortalama metrik sonuçları	37
Çizelge 4.4.	Deney 2 için sınıf bazında ortalama metrik sonuçları	37
Çizelge 4.5.	Deney 3 için ortalama metrik sonuçları	39
Çizelge 4.6.	Deney 3 için sınıf bazında ortalama metrik sonuçları	39
Çizelge 4.7.	Genel ortalama metrik değerleri	41
Çizelge 4.8.	Sınıf bazında ortalama metrik değerleri	42

1. GİRİŞ

Son yıllarda uzaktan algılama teknolojilerindeki gelişmeler yeryüzüne dair daha ayrıntılı verilerin elde edilmesini sağlamıştır. Yüksek çözünürlüklü uydu görüntüleri savunma, çevre izleme, şehir planlaması ve hava sahası yönetimi gibi alanlarda yaygın olarak kullanılmaktadır. Ancak bu görüntüler geniş alanları kapsadığı ve çok sayıda nesne içerdiği için geleneksel analiz yöntemleri nesne tespiti ve sınıflandırmada yeterli doğruluk sağlayamamaktadır. Bu durum, uydu görüntülerinde nesnelerin otomatik olarak tanımlanmasına yönelik yöntemlerin önemini artırmaktadır.

Bu tez çalışmasında uydu görüntülerinde yer alan hava araçlarının otomatik olarak tespit edilmesi amacıyla derin öğrenme tabanlı YOLOv5 mimarisi kullanılmıştır. Modelin performansını artırmak için veri artırımı teknikleri ve genetik algoritma tabanlı hiperparametre optimizasyonu birlikte ele alınmış ve üç farklı eğitim yaklaşımı karşılaştırılmıştır. İlk aşamada YOLOv5, varsayılan hiperparametre ayarları kullanılarak ve herhangi bir veri artırımı yapılmadan eğitilmiştir. Bu aşama modelin veri seti üzerinde hangi seviyede performans gösterebildiğini ortaya koymak için başlangıç noktası olarak değerlendirilmiştir. İkinci aşamada aynı veri setine mekânsal ve fotometrik dönüşümleri içeren veri artırımı yöntemleri uygulanmış, böylece sınırlı örnek sayısının modelin genelleme kabiliyeti üzerindeki etkisi azaltılmaya çalışılmıştır. Son aşamada veri artırımı yöntemleri ile birlikte öğrenme oranı, momentum ve ağırlık çürümesi gibi hiperparametreler genetik algoritma kullanılarak optimize edilmiştir. Bu aşama hiperparametre ayarlarının model performansı üzerindeki etkilerinin değerlendirilmesini sağlamıştır.

Çalışmada kullanılan veri seti, mevcut açık kaynaklı hazır veri setleri yerine Google Earth Pro üzerinden özgün olarak toplanan uydu görüntülerinden oluşturulmuştur. Görseller askeri kargo uçağı, sivil yolcu uçağı ve askeri savaş uçağı olmak üzere üç sınıfa kapsayacak şekilde seçilmiş ve YOLO formatında etiketlenmiştir. Bu sınıflar arasındaki görsel benzerlikler, modelin ayırım gücünü test etmek açısından özellikle tercih edilmiştir.

Tüm çalışma süreci açık kaynak çerçevesinde yürütülmüş, dış kaynaklı içerikler uygun biçimde atıf yapılarak kullanılmıştır. Oluşturulan veri seti herhangi bir ülkenin güvenlik veya savunma politikalarını ihlal etmeyecek şekilde seçilmiş ve yalnızca akademik araştırma amacıyla kullanılmıştır.

2. KAYNAK TARAMASI

2.1. Uydu Görüntülerinde Nesne Tespiti

Günümüzde uydu görüntüleri şehir planlaması, tarımsal izleme, afet yönetimi, çevre gözlemleri ve savunma gibi pek çok alanda etkin biçimde kullanılmaktadır. Yüksek çözünürlükleri ve geniş alanları kapsama özellikleri sayesinde bu görüntüler analiz süreçlerinde değerli birer kaynak olarak öne çıkmaktadır. Ancak bu tür veriler üzerinde nesne tespiti gerçekleştirmek her zaman kolay değildir. Düşük kontrast seviyeleri, karmaşık zemin yapıları, farklı açılardan görülen nesneler ve atmosfer koşulları nesne tespitini önemli ölçüde zorlaştırmaktadır. Bu sebeplerle geleneksel yöntemlerin yetersiz kaldığı durumlarda derin öğrenme tabanlı yaklaşımların daha etkili sonuçlar sunduğu görülmektedir.

Konvolüsyonel sinir ağlarının (CNN) nesne tespiti alanındaki etkisini inceleyen bir çalışmada uydu görüntüleri üzerinde geliştirilmiş farklı mimariler karşılaştırılmıştır. Bu doğrultuda Inception-V2 tabanlı Faster R-CNN ile SSD algoritmaları değerlendirilmiş; doğruluk oranı ve işlem süresi gibi kriterlere göre performans analizi yapılmıştır. Değerlendirme sonuçları Faster R-CNN'nin doğruluk açısından daha başarılı olduğunu, buna karşın SSD algoritmasının daha kısa sürede sonuç ürettiğini ortaya koymuştur (Mansour vd. 2019).

YOLOv3 mimarisi temel alınarak yürütülen bir çalışmada uydu görüntülerinde yer alan nesnelerin daha esnek ve hassas biçimde tespit edilebilmesi amacıyla yeni bir çapa yaklaşımı geliştirilmiştir. Bu yöntemde sınırlayıcı kutuların boyutları %30–40 oranında artırılmış ve modelin genel doğruluk oranında kayda değer bir artış sağlanmıştır. Yapılan testlerde bu iyileştirmenin hassasiyet üzerinde olumsuz bir etkisi olmadığı belirlenmiş; yöntem, uçak, araç, tenis kortu ve köprü gibi nesneleri içeren NWPU VHR-10 veri seti üzerinde başarılı şekilde uygulanmıştır (Laban vd. 2019).

Farklı çevresel koşullar altında araç tespiti amacıyla gerçekleştirilen bir çalışmada Faster R-CNN ve YOLO algoritmalarının performansları detaylı biçimde karşılaştırılmıştır. Uydu görüntüleri üzerinde yapılan deneyler sonucunda Faster R-CNN'nin doğruluk bakımından güçlü sonuçlar verdiği, YOLOv3 algoritmasının ise özellikle havadan çekilen görüntülerde yüksek başarıyla çalıştığı rapor edilmiştir. Her iki yapının da farklı avantajlara sahip olduğu ve belirli senaryolara göre tercih edilebileceği belirtilmiştir (Maity vd. 2021).

Faster R-CNN ve SSD algoritmalarının yüksek çözünürlüklü uydu görüntülerindeki performanslarını inceleyen bir diğer çalışmada bu iki yöntem eğitim sürecine dâhil edilerek karşılaştırmalı analiz gerçekleştirilmiştir. Elde edilen sonuçlar Faster R-CNN'nin karmaşık sahnelerde daha yüksek doğruluk sağladığını, SSD algoritmasının ise daha sade ortamlarda hızlı ve etkili tespitler gerçekleştirdiğini göstermiştir. Her iki algoritmanın da

uçak gibi belirgin nesnelerin tespiti açısından uydu görüntülerinde uygulanabilir yapılar sunduğu ifade edilmiştir (Qifang vd. 2018).

RetinaNet tabanlı bir yaklaşımın, uydu görüntülerinde segmentasyon ve nesne tespiti görevlerini eşzamanlı olarak gerçekleştirecek şekilde tasarlandığı bir çalışmada önerilen yöntem *Balanced Mode* ve *Recall Mode* olmak üzere iki farklı senaryoda test edilmiştir. *Balanced Mode* senaryosunda segmentasyon için duyarlılık (recall) ve kesinlik (precision) değerleri sırasıyla 0.91 ve 0.78, nesne tespiti için ise 0.87 ve 0.75 olarak elde edilmiştir. Önerilen yöntem aynı senaryoda 0.95 doğruluk ve 0.88 kesinlik değerleri ile her iki görevde de daha yüksek performans sergilemiştir. *Recall Mode* senaryosunda ise segmentasyon için 0.95 doğruluk ve 0.50 kesinlik, nesne tespiti için 0.95 doğruluk ve 0.37 kesinlik rapor edilmiştir. Bu modda önerilen yaklaşım 0.96 doğruluk ve 0.84 kesinlik değerleriyle en yüksek sonuçları elde etmiştir. Elde edilen bulgular yöntemin hem dengeli hem de geri çağırma odaklı değerlendirme koşullarında olumlu performans gösterdiğini ortaya koymaktadır (Grosgeorge vd. 2020).

YOLOv5 mimarisi temel alınarak geliştirilen daha hafif ağ yapılarının değerlendirildiği bir çalışmada bu modellerin nesne tespiti performansları DIOR ve DOTA gibi yaygın uydu görüntüsü veri kümeleri üzerinde test edilmiştir. Klasik YOLOv5s sürümüyle kıyaslandığında geliştirilen modellerin işlem süresi açısından daha avantajlı olduğu ve ortalama doğruluk oranında (%mAP) %3.3 ile %6.7 arasında artış sağladığı rapor edilmiştir. Bu bulgular önerilen yapıların büyük ölçekli verilerle çalışan uygulamalarda daha verimli kullanılabileceğini ortaya koymaktadır (Wang vd. 2023).

Küçük ve düşük çözünürlüklü nesnelerin daha isabetli biçimde tespit edilmesini hedefleyen bir çalışmada YOLOv5 temelli bir modele süper çözünürlük katmanı entegre edilmiştir. NWPU VHR-10 veri seti kullanılarak gerçekleştirilen bu çalışmada süper çözünürlük ile desteklenen yapının özellikle küçük nesnelerin yoğun olduğu sahnelerde tespit başarımını önemli ölçüde artırdığı gözlemlenmiştir. Bu yaklaşım düşük çözünürlüklü ve karmaşık yapıya sahip uydu görüntülerinde nesne tespiti performansını artırmak için etkili bir ön işleme yöntemi olarak değerlendirilmiştir (Chen vd. 2024).

2.2. Makine Öğrenmesi Temelli Yaklaşımlar

Geleneksel görüntü işleme yöntemleri, özellikle yüksek çözünürlüklü ve geniş alanları kapsayan verilerin analizinde çoğu zaman sınırlı başarı göstermektedir. Çevresel koşullardaki değişkenlik, ışık şiddetindeki dalgalanmalar ve sahne yapılarının karmaşıklığı, bu yöntemlerin performansını olumsuz etkileyen başlıca faktörlerdir. Son yıllarda, bu kısıtların aşılması amacıyla makine öğrenmesi ve özellikle derin öğrenme tabanlı yaklaşımlar ön plana çıkmıştır. Konvolüsyonel Sinir Ağları (CNN), Tekrarlayan Sinir Ağları (RNN), Üretici Çekişmeli Ağlar (GAN) ve Transformer tabanlı mimariler; sınıflandırma,

nesne tespiti ve semantik segmentasyon gibi temel görevlerde hem doğruluk hem de işlem verimliliği açısından önemli iyileşmeler sağlamaktadır.

Makine öğrenmesi ve derin öğrenme tabanlı mimarilerin yüksek çözünürlüklü uydu görüntülerinde ölçeklenebilirlik, gerçek zamanlı analiz ve adaptif öğrenme yetenekleri bakımından sundukları avantajlar, literatürde kapsamlı biçimde incelenmiştir. Yapılan değerlendirmeler, bu sistemlerin yalnızca doğruluk oranlarını artırmakla kalmayıp, aynı zamanda veri işleme sürelerini kısaltarak kaynak kullanımını da optimize edebildiğini ortaya koymaktadır (Shoaib vd. 2025).

2.2.1. Denetimli Öğrenme (Supervised Learning, SL)

Denetimli öğrenme, makine öğrenmesinin temel yaklaşımlarından biridir ve etiketli veri kümeleri kullanılarak modelin giriş-çıkış ilişkisini öğrenmesine dayanır. Bu yöntemde her eğitim örneği, belirli giriş özellikleri ile buna karşılık gelen doğru çıktı etiketinden oluşur. Model, bu örneklerden yola çıkarak yeni ve daha önce karşılaşmadığı veriler üzerinde doğru tahminler yapmayı hedefler. Denetimli öğrenme, özellikle sınıflandırma (örneğin, hastalık tanısı) ve regresyon (örneğin, enerji tüketimi tahmini) problemlerinde yaygın olarak kullanılmaktadır. Model başarımı, problem türüne bağlı olarak doğruluk (*accuracy*), kesinlik (*precision*), duyarlılık (*recall*) veya hata ölçümleri gibi çeşitli istatistiksel metriklerle değerlendirilebilir. Sağlık, finans, üretim ve güvenlik gibi birçok alanda geniş çaplı uygulama örnekleri bulunan bu yaklaşım, veri ile öğrenme süreci arasında doğrudan bir bağ kurması sayesinde yüksek tahmin performansı sağlayabilmektedir (Liu ve Wu 2012).

2.2.2. Denetimsiz Öğrenme (Unsupervised Learning, UL)

Denetimsiz öğrenme, etiketlenmemiş veri kümelerinde yer alan yapısal örüntülerin ve benzerliklerin keşfedilmesini amaçlayan makine öğrenmesi yaklaşımıdır. Özellikle uzaktan algılama alanında, geniş ölçekli verilerden anlamlı bilgi çıkarımı için yaygın olarak kullanılmaktadır. K-means, Gaussian karışım modelleri (GMM) ve yoğunluk temelli kümeleme (DBSCAN) gibi yöntemler, arazi örtüsü sınıflandırması, değişim tespiti ve nesne gruplama gibi görevlerde sıklıkla tercih edilmektedir. Bununla birlikte, hiperspektral görüntülerin yüksek boyutluluğu ve karmaşık spektral dağılımı, geleneksel yöntemlerin doğruluk ve genellenebilirlik açısından sınırlı kalmasına yol açabilmektedir.

Bu sınırlamaları aşmaya yönelik olarak yapılan bir çalışmada, hiperspektral görüntüler için difüzyon tabanlı benzerlik ölçümleri ile hacim maksimize eden kümeleme stratejilerini bir araya getiren özgün bir denetimsiz yöntem geliştirilmiştir. Bu yöntemde, veri dağılımının geometrik özellikleri daha etkili biçimde modellenmiş ve kümeleme başarımı artırılmıştır. Özellikle karmaşık arazi örtüsü tiplerinin ayrımında, klasik K-means

ve GMM yöntemlerine kıyasla daha yüksek doğruluk elde edilmiştir. Böylece, yüksek boyutlu uzaktan algılama verilerinde denetimsiz öğrenmenin etkinliğini artıran güncel ve etkili bir yöntem ortaya konulmuştur (Polk vd. 2023).

2.2.3. Yarı Denetimli Öğrenme (Semi-Supervised Learning, SSL)

Yarı denetimli öğrenme, hem etiketli hem de etiketsiz verileri birlikte kullanarak model performansını artırmayı amaçlayan, denetimli ve denetimsiz öğrenme yaklaşımlarının birleşiminden oluşan bir makine öğrenmesi yöntemidir. Bu yaklaşım, özellikle etiketli veri toplamanın maliyetli veya zaman alıcı olduğu durumlarda önemli avantajlar sunar. SSL'in başarısı, etiketsiz veriden elde edilen bilginin modelin karar sınırlarını geliştirecek şekilde etkin biçimde kullanılmasına bağlıdır.

Görüntü, metin ve ses veri kümelerinde 16 farklı derin SSL algoritmasının karşılaştırmalı olarak incelendiği bir araştırma, FreeMatch, SimMatch ve SoftMatch gibi yöntemlerin gürültülü veri ortamlarında dahi düşük hata oranlarıyla öne çıktığını rapor etmiştir. Bu bulgular, SSL'in yalnızca veri yetersizliği sorununu azaltmakla kalmayıp, aynı zamanda farklı veri türlerinde ve zorlayıcı koşullarda da yüksek genellenebilirlik sağlayabileceğini göstermektedir (Fredriksson ve Zhang 2025).

2.2.4. Pekiştirmeli Öğrenme (Reinforcement Learning, RL)

Pekiştirmeli öğrenme, bir ajanın çevresiyle sürekli etkileşim kurarak, aldığı geri bildirimleri kullanıp davranışlarını iyileştirdiği bir makine öğrenmesi yaklaşımıdır. Bu yöntemde ajanın amacı, verilen görev hedeflerini en iyi şekilde gerçekleştirecek eylem stratejisini öğrenmektir. RL'in en temel unsurlarından biri ödüdür; ödül, ajanın istenen davranışlara yönelmesini sağlayan ve uzun vadede kümülatif ödülün en üst düzeye çıkarılmasına rehberlik eden sinyaldir. Derin pekiştirmeli öğrenme (Deep RL) ise, RL'in derin öğrenme teknikleri ile birleşmesiyle, yüksek boyutlu ve karmaşık durum-aksiyon uzaylarında başarılı eylem stratejisi öğrenimini mümkün kılar.

Literatürdeki bir çalışmada, derin RL alanındaki ödül modelleme (reward modeling) kavramı; ödül kaynakları (insan tarafından sağlanan veya yapay zekâ tarafından üretilen), mekanizmalar (dışsal ve içsel ödüller) ve öğrenme paradigmaları (gösterimlerden öğrenme, hedeflerden öğrenme, tercihlerden öğrenme) kapsamında sistematik bir çerçevede incelenmiştir. Bulgularda, ödül modellemenin hem akademik hem de endüstriyel uygulamalarda kritik bir rol oynadığı ve doğru tasarlanmış ödül modellerinin etmenin hedeflerle daha yüksek uyum sağlamasına katkıda bulunduğu vurgulanmıştır (Yu vd. 2025).

2.3. Derin Öğrenme ve CNN Tabanlı Yöntemler

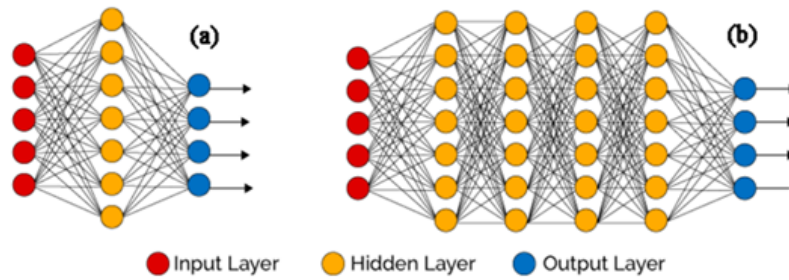
2.3.1. Derin Öğrenmenin Temelleri

Derin öğrenme, makine öğrenmesinin bir alt alanı olarak, verilerdeki karmaşık yapıları çok katmanlı yapay sinir ağlarıyla tanıyıp anlamlandırmayı amaçlar. Klasik yöntemlerden farkı, özellikleri doğrudan veriden çıkarabilmesidir. Özellikle büyük ve zengin veri kümelerinde, insan müdahalesi olmadan örüntüleri öğrenebilme yeteneği, onu birçok uygulama alanında öne çıkarmıştır.

Derin öğrenme modelleri, bilgiyi katmanlar boyunca işleyerek giderek daha soyut temsillere dönüştürür. İlk katmanlar kenar ve doku gibi basit özellikleri öğrenirken, ara katmanlar nesne bileşenlerini ve konumlarını ayırt eder. Son katmanlar ise sınıflandırma veya nesne tespiti gibi hedeflenen çıktıları üretir. Bu yapı, klasik yöntemlerle elde edilmesi zor karmaşık desenlerin otomatik olarak öğrenilmesini sağlar (Goodfellow vd. 2016).

Derin öğrenmenin geniş bir kullanım alanına ulaşmasında yalnızca yöntemsel gelişmeler değil, aynı zamanda teknolojik ilerlemeler de belirleyici olmuştur. Daha büyük veri kümelerine erişim imkânının artması, grafik işlem birimlerinin (GPU) sunduğu yüksek hesaplama gücü ve açık kaynak kodlu yazılımların yaygınlaşması, bu alanın gelişimini hızlandırmıştır. Derin öğrenme, görüntü işleme, konuşma tanıma ve doğal dil işleme gibi karmaşık veri tiplerinde, özellikle görsel veriye dayalı uygulamalarda geleneksel yöntemlerden daha yüksek başarı sağlamıştır (Goodfellow vd. 2016).

Şekil 2.1’de, yapay sinir ağlarının katman yapısı basit ve derin mimari örnekleri üzerinden karşılaştırmalı olarak gösterilmektedir.



Şekil 2.1. a) Basit sinir ağı; b) derin sinir ağı mimarisi (İbrahim vd. 2021)

Basit sinir ağı: giriş katmanı (kırmızı), tek bir gizli katman (sarı) ve çıkış katmanından (mavi) oluşan basit sinir ağı yer almaktadır. Derin sinir ağı mimarisi ise, giriş ve çıkış katmanları arasında birden fazla gizli katman içeren derin sinir ağı görülmektedir. Derin mimariler, çok katmanlı yapıları sayesinde veriden daha karmaşık ve soyut özellikler çıkarabilme yeteneğine sahiptir.

2.3.2. Konvolüsyonel Sinir Ağları (CNN)

Konvolüsyonel Sinir Ağları (CNN), özellikle görsel veriler üzerinde işlem yapmak üzere tasarlanmış derin öğrenme mimarileridir. Bu ağlar, görüntüdeki şekil, kenar, doku ve renk geçişleri gibi öğeleri bulundukları yerle birlikte değerlendirebildiği için, görsel içeriklerin tanınmasında yüksek performans göstermektedir. Bu özellikleriyle klasik yapay sinir ağlarından ayrılır ve görsel veriyle çalışan pek çok uygulamada yaygın olarak tercih edilmektedir (Gu vd. 2018).

Bir CNN modeli genellikle üç temel bileşenden oluşur: konvolüsyon katmanları, havuzlama (pooling) katmanları ve tam bağlantılı (fully connected) katmanlar. Konvolüsyon katmanları, görüntüdeki kenar, doku ve renk geçişi gibi temel özellikleri algılamak için filtreler kullanır. Havuzlama katmanları, bu özellikleri daha küçük boyutlara indirerek hem işlem yükünü azaltır hem de modelin, görüntüdeki küçük konumsal değişikliklere ve ölçek farklılıklarına karşı daha kararlı çalışmasını sağlar. Tam bağlantılı katmanlar ise önceki katmanlardan gelen bilgileri bütünleştirerek sınıflandırma veya tahmin görevlerini gerçekleştirir (Gu vd. 2018).

CNN tabanlı yaklaşımlar; görüntü sınıflandırma, nesne tespiti, semantik segmentasyon ve yüz tanıma gibi farklı görevlerde yüksek doğruluk oranlarına ulaşmıştır. Bu başarı yalnızca mimarinin yapısından değil, aynı zamanda eğitim verisinin çeşitliliği ve uygun hiperparametre seçiminin sağladığı katkılardan da kaynaklanmaktadır. Özellikle yüksek çözünürlüklü ve görsel çeşitliliği fazla olan uydu görüntülerinde, CNN tabanlı yöntemler, karmaşık arka plan koşullarına rağmen küçük nesneleri doğru şekilde tespit edebilme kapasitesiyle öne çıkmaktadır (Zhao vd. 2019).

2.3.3. Uydu Görüntülerinde CNN Uygulamaları

Konvolüsyonel Sinir Ağları (CNN), görüntü işleme alanındaki başarısının yanı sıra, uzaktan algılama ve özellikle uydu görüntülerinin analizi gibi alanlarda da yaygın olarak kullanılmaktadır. Uydu görüntüleri genellikle yüksek çözünürlüklü olup, farklı spektral bantlar içerir ve çok sayıda nesneyi barındırabilir. Bu çeşitlilik, veri işleme sürecinde derin öğrenme tabanlı modellere olan ihtiyacı artırmaktadır.

CNN yapıları, uydu görüntülerinde hem arazi örtüsü sınıflandırması gibi genel analizlerde hem de küçük nesnelerin tespiti gibi ayrıntılı görevlerde etkili sonuçlar vermektedir. Özellikle nesnelerin farklı boyut, konum ve dönüş açılarına sahip olduğu durumlarda, CNN'lerin çok katmanlı yapısı bu çeşitliliğe uyum sağlayarak tespit doğruluğunu artırabilmektedir (Cheng vd. 2016). Ayrıca, veri artırımı yöntemleri ve önceden eğitilmiş ağların yeniden kullanımı (transfer öğrenme) sayesinde, sınırlı etiketli veriye sahip durumlarda dahi tatmin edici performanslar elde edilebilmektedir. DOTA gibi geniş ölçekli

veri kümeleri üzerinde yapılan çalışmalar, CNN tabanlı yaklaşımların yüksek çözünürlüklü hava görüntülerinde, farklı ölçekteki nesnelerin tespitinde başarılı sonuçlar verdiğini göstermiştir (Xia vd. 2018).

2.4. YOLO Algoritmaları

Derin öğrenme tabanlı nesne tespiti yaklaşımları arasında gerçek zamanlı çalışabilme yeteneğiyle öne çıkan YOLO (You Only Look Once) algoritmaları, nesnelerin konumunu ve sınıfını tek bir ileri besleme adımıyla tahmin edecek şekilde tasarlanmıştır (Redmon vd. 2016). Bu yaklaşımda görüntü, eşit boyutlu hücrelere bölünür ve her hücre için nesne olup olmadığı, varsa sınır kutusu (bounding box) koordinatları ve sınıf tahmini yapılır. Böylece, tüm görüntü tek seferde işlenir ve klasik iki aşamalı yöntemlere kıyasla analiz süresi önemli ölçüde kısalır.

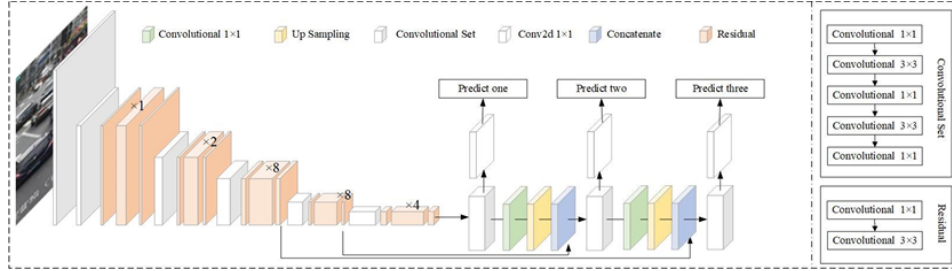
YOLO mimarisi, hız ve doğruluk dengesini optimize eden çeşitli teknikler ve iyileştirmelerle geliştirilmiştir (Bochkovskiy vd. 2020). Mimari düzenlemeler, veri artırma yöntemleri ve optimizasyon teknikleri sayesinde, özellikle büyük veri kümelerinde hem yüksek doğruluk hem de gerçek zamanlı işleme hızları elde edilebilmektedir. Bu özellikler, YOLO ailesini gerçek zamanlı nesne tespiti gerektiren pek çok uygulamada yaygın olarak tercih edilen bir yöntem hâline getirmiştir.

2.4.1. YOLO v3

YOLOv3'ün temelinde yer alan Darknet-53 ağı, artık bağlantılar (residual connections) sayesinde bilgi kaybı olmaksızın daha derin katmanlardan öğrenmeyi sürdürebilecek biçimde tasarlanmıştır. Bu sayede, özellikle benzer sınıflar arasındaki ince farkların ayırt edilmesi gereken karmaşık senaryolarda daha başarılı sonuçlar elde edilmiştir. YOLOv3, farklı boyutlardaki nesneleri tespit edebilmek için üç ölçekte çıktı üretir; bu çok ölçekli tahmin mekanizması küçük ve büyük nesnelerin tutarlı biçimde algılanmasını sağlamıştır. Model, konum kestirimi için sabit kutu oranlarını (anchor boxes) kullanmış ve bu oranlar K-means algoritması ile belirlenmiştir; böylece farklı nesne boyutlarına uyum sağlanarak konum tahmin doğruluğu artırılmıştır (Redmon ve Farhadi 2018).

Mimari, evrimsel katmanların yanı sıra Batch Normalization ve Leaky ReLU aktivasyon fonksiyonlarını kullanır. Klasik havuzlama yöntemleri yerine, stride değeri 2 olan evrişim katmanları ile alt örnekleme yapılır; böylece görüntüdeki bilgi daha fazla korunur. Üç ölçekli tahmin yapısında, giriş görüntüsünün $1/32$, $1/16$ ve $1/8$ oranına indirgenmiş hâlleri üzerinde çıktılar üretilir. Özellikle $1/8$ ölçekli çıktı katmanı, küçük nesnelerin daha doğru bir şekilde tespit edilmesinde kritik rol oynar (Redmon ve Farhadi 2018).

Şekil 2.2’de, ağ mimarisinde evrişimsel katman grupları (convolutional set) arasına yerleştirilen artık bağlantılar gösterilmiştir.



Şekil 2.2. YOLOv3 mimarisi ve Darknet-53 omurga yapısı (Mao vd. 2019)

Ağ yapısı içerisinde sıralı olarak yer alan *Conv2d 1x1* katmanlarının, tahminlerin oluşturulmasında kullanıldığı görülmektedir. Bu katmanlar, önceki katmanlardan gelen bilgileri sadeleştirerek son katmanda yapılacak tahminlere temel oluşturur. Bu yapı, modelin farklı katmanlardan elde ettiği bilgileri bir araya getirerek kullanmasını sağlar.

Şekil 2.3’te, YOLOv3 mimarisinde kullanılan Darknet-53 omurgasının katman yapısı gösterilmiştir.

	Type	Filters	Size	Output
1x	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
2x	Convolutional	128	3 × 3 / 2	64 × 64
	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
8x	Convolutional	256	3 × 3 / 2	32 × 32
	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
8x	Convolutional	512	3 × 3 / 2	16 × 16
	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
4x	Convolutional	1024	3 × 3 / 2	8 × 8
	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Şekil 2.3. Darknet-53 yapısı (Redmon ve Farhadi 2018)

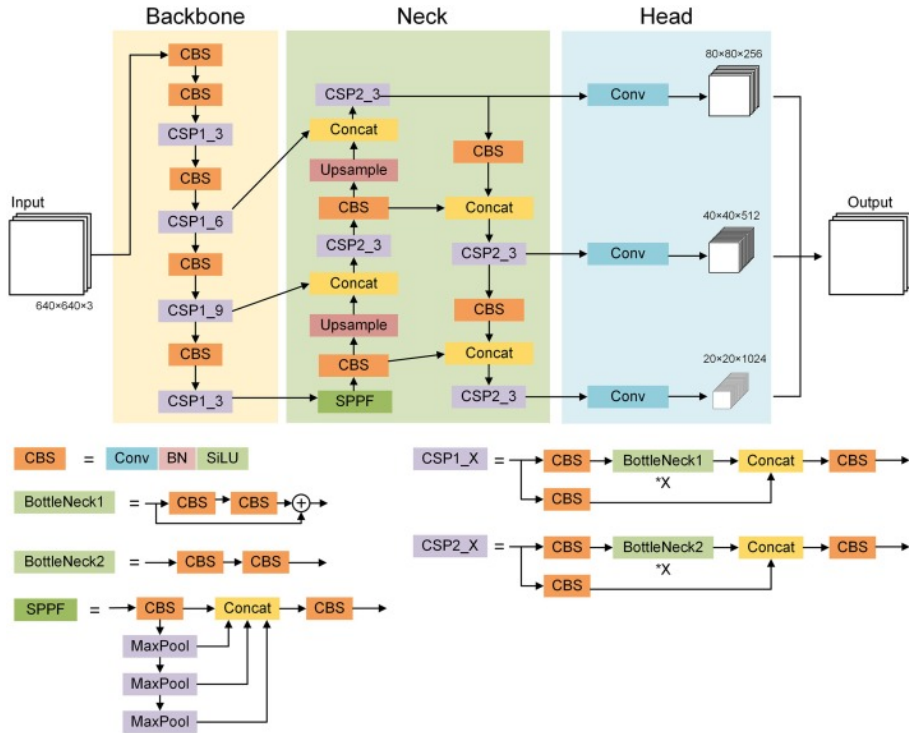
Evrişimsel katmanların filtre sayıları, çekirdek boyutları ve her aşamadaki çıkış çözünürlükleri görselleştirilmiştir. Ayrıca, residual bloklar ile ağın derinliği arttırılırken öğrenme kayıplarının önüne geçilmesi amaçlanmaktadır. Stride değeri 2 olan evrişim katmanları, giriş görüntüsünün boyutunu kademeli olarak küçültürken, ağın daha karmaşık özellikleri öğrenmesine yardımcı olur. Ağın sonundaki bu yapı, giriş görüntüsünün ait olduğu sınıfın belirlenmesi için gerekli olan son çıktıyı üretir.

2.4.2. YOLO v5

PyTorch tabanlı olarak geliştirilen YOLOv5 model mimarisi, farklı donanım kapasitelerine uygun olacak şekilde yapılandırılmıştır. Temelinde CSPDarknet adı verilen bir omurga yer almaktadır. CSP (Cross Stage Partial) mimarisi, Öğrenme sırasında yapılan güncellemelerin derin katmanlara kadar etkili şekilde ulaşmasını sağlar. Verinin bir kısmını işlemeye alıp bir kısmını doğrudan ileten CSP yapısı, ağır daha hafif çalışmasını sağlarken öğrenme sırasında bilgi kaybını da önlemeye yardımcı olur. PANet (Path Aggregation Network), alt katmanların sağladığı ayrıntılı bilgiler ile üst katmanların genelleştirilmiş temsilleri birlikte kullanılarak, modelin hem küçük hem de büyük nesneleri tanıma yeteneği artırılır. Giriş görüntüsünü farklı çözünürlüklerde değerlendiren çok ölçekli yapı, uydu görüntülerinde sık karşılaşılan küçük nesnelerin algılanmasında modelin doğruluğunu artırmaktadır.

YOLOv5, yüksek çözünürlüklü uydu görüntülerinden oluşan veri kümelerinde zaman ve doğruluk açısından etkili sonuçlar ortaya koymuştur. Özellikle *DOTA*, *NWPU VHR-10* ve *xView* gibi çok sayıda sınıf içeren ve görsel açıdan karmaşık veri setleri üzerinde yapılan deneylerde, bu mimarinin gerçek zamanlı nesne tespiti görevleri için etkili bir seçenek olduğu değerlendirilmiştir (Jocher vd. 2023; Bochkovskiy vd. 2020).

YOLOv5 mimarisi, üç temel bileşenden oluşmaktadır: *Backbone*, *Neck* ve *Head*. Bu mimarinin genel yapısı Şekil 2.4'te gösterilmiştir.



Şekil 2.4. YOLOv5 mimari yapısı (Jiang vd. 2023)

Backbone (omurga) kısmı, girdi olarak verilen görüntü üzerinden temel özneliklerin (feature) çıkarılmasını sağlar. Bu aşamada *CSPDarknet* yapısı kullanılır. Cross Stage Partial (CSP) blokları, parametre verimliliğini artırırken gereksiz tekrarları azaltarak daha hızlı ve dengeli bir öğrenme sağlar. Backbone katmanlarında, ilk aşamalarda düşük seviyeli özellikler (kenar, renk geçişi), ilerleyen katmanlarda ise yüksek seviyeli soyut özellikler (şekil, yapı) elde edilir. Bu bölüm sonunda *P3* (80×80), *P4* (40×40) ve *P5* (20×20) olmak üzere üç farklı ölçekli öznelik haritası çıkarılır.

Neck (boyun) kısmı, Backbone'dan elde edilen çok ölçekli özellik haritalarını birleştirerek zenginleştirir. Bu kısımda *PANet* (*Path Aggregation Network*) yapısı kullanılır. PANet, hem yukarıdan aşağıya hem de aşağıdan yukarıya bilgi akışı sağlayarak küçük nesne tespiti için daha güçlü temsiller oluşturur. Ayrıca, *SPPF* (*Spatial Pyramid Pooling Fast*) modülü yardımıyla farklı boyutlarda max pooling uygulanarak modelin farklı ölçeklere duyarlılığı artırılır.

Head (çıkıtı) kısmı, üç farklı ölçekten gelen öznelikler üzerinden nihai tahminlerin yapılmasını sağlar. Bu bölümde her bir öznelik haritası için belirli grid hücreleri tanımlanır ve bu hücrelerden her biri için: sınıf tahmini (class prediction), sınır kutusu koordinatları (bounding box regression), güven değeri (confidence score) üzerinden tahminler yapılır. Elde edilen tahminler, *Non-Maximum Suppression* (NMS) algoritması ile filtrelenerek son tespitler oluşturulur. *P3* katmanı küçük nesnelere, *P4* orta büyüklükteki nesnelere ve *P5* ise büyük nesnelere karşılık gelecek şekilde modelin çok ölçekli nesne tespiti yeteneği güçlendirilmiştir.

Veri setinin sınırlı sayıda örnek içermesi, modelin her bir öğrenme biriminden elde ettiği bilginin önemini daha da artırmaktadır. Özellikle derin sinir ağı içerisindeki öznelik haritaları (feature maps), az sayıda veriden en fazla anlamın çıkarılabilmesini mümkün kılan temel yapılarıdır. Bu yaklaşım ile, YOLOv5 mimarisinde yer alan CSP (Cross Stage Partial) blokları, bilgiyi kaybetmeden daha verimli öznelik temsilleri oluşturulmasına katkı sağlamaktadır. CSP blokları içerisinde yer alan aktivasyon fonksiyonu olarak kullanılan SiLU (Sigmoid Linear Unit), sınırlı veri ile öğrenme sürecinde modelin daha dengeli ve yumuşak geçişli kararlar almasını desteklemektedir. SiLU fonksiyonu, giriş değerlerine göre negatif değerleri sıfıra yakın, pozitif değerleri ise bire yakın değerlere çeker; ancak ReLU gibi keskin sınırlar getirmez. Bu sayede, özellikle sınırlı sayıda örnek içeren veri kümelerinde, her bir öznelik katkısı korunmuş olur.

SiLU aktivasyon fonksiyonu aşağıdaki şekilde tanımlanır:

$$\text{SiLU}(x) = x \cdot \sigma(x) = \frac{x}{1 + e^{-x}} \quad (2.1)$$

Burada $\sigma(x)$ sigmoid fonksiyonunu, x ise giriş değerini temsil etmektedir. Denklem

(2.1)'de verilen bu ifade, YOLOv5 mimarisinde kullanılan aktivasyon fonksiyonunun matematiksel tanımıdır.

YOLOv5 modelinde kullanılan kayıp fonksiyonu, kutu regresyonu, nesne varlığı ve sınıf tahmini olmak üzere üç temel bileşenden oluşur. Toplam kayıp fonksiyonu Denklem (2.2)'de gösterildiği üzere şu şekilde tanımlanır:

$$\mathcal{L}_{\text{total}} = \lambda_{\text{box}} \cdot \mathcal{L}_{\text{CIoU}} + \lambda_{\text{obj}} \cdot \mathcal{L}_{\text{obj}} + \lambda_{\text{cls}} \cdot \mathcal{L}_{\text{cls}} \quad (2.2)$$

Burada $\mathcal{L}_{\text{CIoU}}$ CIoU tabanlı konum kaybını, $\mathcal{L}_{\text{obj}}$ nesne varlığına ilişkin Binary Cross Entropy kaybını ve $\mathcal{L}_{\text{cls}}$ ise sınıf tahmini kaybını ifade eder. Her bileşenin önüne gelen λ katsayıları, ilgili kaybın toplam fonksiyondaki ağırlığını belirler.

Konum kaybı olarak kullanılan CIoU fonksiyonu Denklem (2.3)'te ifade edildiği gibi yalnızca kutuların çakışma oranını değil, aynı zamanda merkez uzaklığı ve en-boy oranı farklılıklarını da dikkate alarak daha hassas hizalama sağlar:

$$\mathcal{L}_{\text{CIoU}} = 1 - IoU + \frac{\rho^2(\mathbf{b}, \mathbf{b}_{gt})}{c^2} + \alpha v \quad (2.3)$$

Burada ρ iki kutunun merkezleri arasındaki Öklidyen mesafeyi, c en küçük kapsayıcı kutunun köşegen uzunluğunu, v ise en-boy oranı uyumsuzluk ölçüsünü, α ise bu ölçünün etkisini kontrol eden katsayıyı ifade eder.

Nesne varlığı (objectness) kaybı için kullanılan Binary Cross Entropy fonksiyonu Denklem (2.4)'te verilmiştir:

$$\mathcal{L}_{\text{obj}} = -[y \cdot \log(\hat{y}) + (1 - y) \cdot \log(1 - \hat{y})] \quad (2.4)$$

Burada y gerçek etiket, $\hat{y}$ model tahminidir.

Çok sınıflı sınıflandırma için kullanılan sınıf kaybı formülü Denklem (2.5)'te gösterilmiştir:

$$\mathcal{L}_{\text{cls}} = \sum_{c=1}^C [y_c \cdot \log(\hat{y}_c) + (1 - y_c) \cdot \log(1 - \hat{y}_c)] \quad (2.5)$$

YOLOv5'te kullanılan bu kayıp fonksiyonları, modelin öğrenme sürecini hem hızlandırmakta hem de daha güvenilir tahminler yapılmasını sağlamaktadır. Özellikle konum doğruluğu, nesne varlığı ve sınıf ayrımı gibi bileşenleri ayrı ayrı optimize eden bu yapı, modelin yüksek çözünürlüklü ve karmaşık uydu görüntüleri üzerinde daha hassas ve tutarlı sonuçlar üretmesine olanak tanımaktadır.

2.5. Genetik Algoritmalar ile Hiperparametre Optimizasyonu

Derin öğrenme modellerinin performansı, yalnızca mimarilerine değil, aynı zamanda eğitim sürecinde kullanılan hiperparametrelerin doğru biçimde seçilmesine de bağlıdır. Öğrenme oranı, batch boyutu ve epoch sayısı gibi hiperparametreler, modelin öğrenme davranışlarını ve çıktı kalitesini doğrudan etkileyen unsurlar arasında yer almaktadır. Uygun şekilde ayarlanmayan hiperparametreler, modelin yetersiz öğrenmesine ya da aşırı öğrenerek genelleme kabiliyetini kaybetmesine neden olabilir. Özellikle uydu görüntüleri gibi nesne çeşitliliği yüksek ortamlarda, bu parametrelerin dengeli biçimde seçilmesi kritik bir gereklilik hâline gelmektedir (Feurer ve Hutter 2019).

Kapsamlı arama (grid search) ve rastgele arama (random search) gibi klasik hiperparametre optimizasyon yöntemleri, sınırlı parametre uzaylarında etkili olabilir. Ancak derin öğrenme modellerinin yüksek boyutlu yapılarına uygulandığında bu yöntemler zaman ve kaynak açısından ciddi verimsizlikler yaratabilmektedir. Bu nedenle son yıllarda, daha esnek ve öğrenmeye duyarlı meta-sezgisel yöntemlere olan ilgi artmıştır (Bischl vd. 2023).

Genetik algoritmalar, çözüm arayışında doğadaki evrimsel süreci taklit eden yöntemler arasında öne çıkmaktadır. Her birey, farklı bir hiperparametre kombinasyonunu temsil eder ve belirli bir başarı fonksiyonuna (fitness) göre değerlendirilir. Bu yaklaşım, ilk olarak John Holland tarafından ortaya konulmuş ve evrimsel süreçlerin matematiksel modellenmesi üzerine kurulmuştur (Holland 1975). Genetik algoritmalarda bireyin başarısı, genellikle doğruluk ve kayıp fonksiyonlarıyla ilişkilendirilerek bir uygunluk fonksiyonu şeklinde tanımlanır. Örneğin, Denklem (2.6)'da gösterildiği gibi karma bir ölçüt kullanılabilir:

$$f(x) = \text{Accuracy}(x) - \alpha \cdot \text{Loss}(x) \quad (2.6)$$

Bu denklemde x bireyi temsil ederken, α katsayısı doğruluk ile kayıp arasındaki dengeyi sağlamaktadır. Alternatif olarak, daha sade bir uygunluk fonksiyonu Denklem (2.7)'de verilmiştir:

$$f(x) = \frac{1}{1 + L(x)} \quad (2.7)$$

Burada $L(x)$, bireyin kayıp (loss) değerini ifade eder ve uygunluk değeri, kayıp azaldıkça artmaktadır.

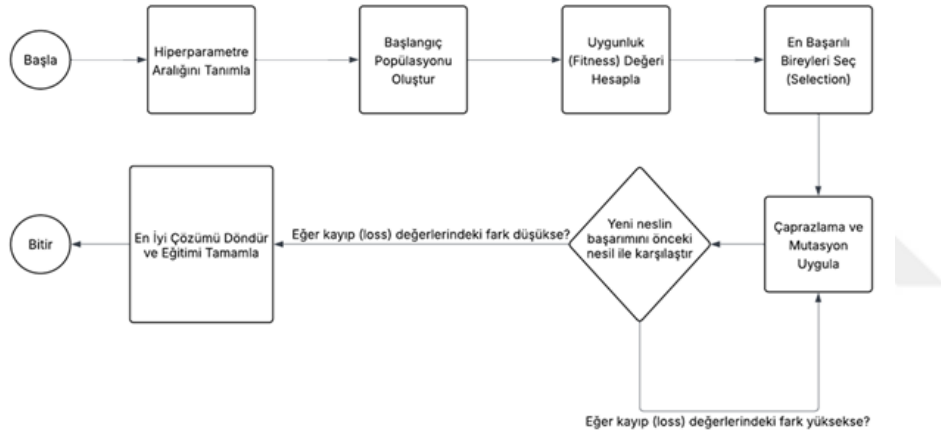
Genetik algoritmaların çalışma mantığında temel adımlar seçim (selection), çaprazlama (crossover) ve mutasyon (mutation) işlemlerinden oluşmaktadır. Bu işlemlerin birleşimiyle yeni bireyler üretilerek popülasyon bir sonraki nesle taşınır. Popülasyonun her bir nesilde nasıl güncellendiği Denklem (2.8)'de genel biçimde ifade edilmektedir:

$$P_{t+1} = \text{Mutation}(\text{Crossover}(\text{Selection}(P_t))) \quad (2.8)$$

Burada P_t , t anındaki mevcut popülasyonu temsil ederken, P_{t+1} bir sonraki nesildeki güncellenmiş popülasyonu ifade etmektedir.

Genetik algoritmaların derin öğrenme modellerindeki hiperparametre ayarları üzerinde etkili sonuçlar sunduğu, son yıllarda birçok çalışmada gösterilmiştir. Ultralytics tarafından geliştirilen YOLOv5 modelinde de, “hyperparameter evolution” olarak adlandırılan bir özellik bulunmaktadır. Bu özellik, kullanıcı tarafından belirlenen aralıklar üzerinden evrimsel prensipleri kullanarak arama yapar ve öğrenme oranı (*learning rate*), momentum, sınıf (*class loss*) ve nesne (*object loss*) kaybı katsayıları gibi parametreler için optimize edilmiş değerler önerir. Sürecin tamamlanmasının ardından, modelin doğruluk (*accuracy*) oranında artış ve eğitim süresinde verimlilik sağlanabildiği belirtilmektedir (Ultralytics 2024).

Genetik algoritmanın hiperparametre optimizasyonundaki temel adımları, Şekil 2.5’te yer alan akış diyagramında gösterilmiştir.



Şekil 2.5. Genetik algoritma ile hiperparametre optimizasyonu akış şeması

Akış şeması incelendiğinde, süreç ilk olarak hiperparametre aralıklarının tanımlanmasıyla başlamakta, ardından bu aralıklardan rastgele bireylerden oluşan bir başlangıç popülasyonu oluşturulmaktadır. Her birey için bir uygunluk (fitness) değeri hesaplanmakta ve en iyi bireyler seçilerek genetik işlemler olan çaprazlama ve mutasyon uygulanmaktadır. Kullanılan uygunluk değeri, modelin doğrulama kümesi üzerindeki toplam kayıp değerine dayalı olarak Denklem (2.9)’da gösterildiği gibi tanımlanmıştır:

$$\text{Fitness} = \frac{1}{L_{\text{val}} + \epsilon} \quad (2.9)$$

Burada L_{val} doğrulama kümesindeki toplam kayıp değerini, ϵ ise sıfıra bölme hatasını engellemek amacıyla kullanılan küçük bir pozitif sabiti temsil etmektedir.

Elde edilen yeni nesil bireylerle yapılan değerlendirmede, kayıp (loss) değerindeki iyileşme belirlenen eşiğin altına düştüğünde süreç sonlandırılmakta; aksi durumda döngü

yeni bireylerle sürdürülmektedir. Bu evrimsel yaklaşımın, modelin doğruluk ve genelleme performansını artırabileceği değerlendirilmektedir; etki düzeyi veri setine ve ayarlara bağlı olarak değişebilmektedir.

2.6. Nesne Tespitinde Kullanılan Diğer Derin Öğrenme Modelleri

2.6.1. Faster R-CNN (Faster Region-Based CNN)

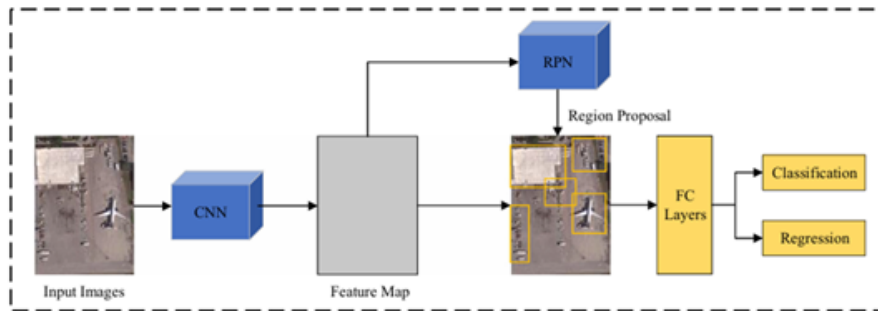
Faster R-CNN (Faster Region-Based CNN), nesne tespiti için geliştirilen en yaygın ve yüksek performanslı derin öğrenme tabanlı mimarilerden biridir. İki aşamalı yapısı, bölge önerisi oluşturma ve bu bölgelerde nesne sınıflandırma adımlarını bütünleşik biçimde gerçekleştirir. Mimari üç bileşenden oluşur: öznetelik çıkarımı için CNN tabanlı backbone (örn. VGG16, ResNet), bölge adaylarını üreten Region Proposal Network (RPN) ve sınıflandırma ile konum iyileştirmesini yapan Fast R-CNN ağı. RPN, özellik haritası üzerinde kayan pencere yöntemiyle farklı ölçek ve en-boy oranlarındaki anchor box'lar üzerinden olası nesne bölgeleri üretir. Bu bölgeler RoI pooling katmanında sabit boyutlu özellik haritalarına dönüştürülür ve ardından sınıflandırma ile kutu regresyonu yapılır (Patki ve Gawade 2024).

Eğitim süreci, sınıflandırma kaybı (L_{cls}) ile sınırlayıcı kutu regresyon kaybını (L_{loc}) birleştiren çok görevli bir kayıp fonksiyonu ile uçtan uca (end-to-end) şekilde optimize edilir. Bu fonksiyon Denklem (2.10)'da gösterildiği gibi tanımlanır:

$$L(p, u, t'', v) = L_{cls}(p, u) + \lambda \cdot [u \geq 1] L_{loc}(t'', v) \quad (2.10)$$

Burada p , tahmin edilen sınıf olasılıkları; u , doğru sınıf etiketini; t'' , tahmin edilen sınırlayıcı kutu koordinatlarını; v , gerçek (ground truth) koordinatlarını; λ , iki kayıp terimi arasındaki denge katsayısını ifade etmektedir. $[u \geq 1]$ ifadesi, konum kaybının yalnızca pozitif örnekler (foreground) için hesaplandığını belirtir. Denklem (2.10)'da verilen bu yapı sayesinde Faster R-CNN, harici bölge öneri algoritmalarına olan ihtiyacı ortadan kaldırarak hem hız hem de doğruluk açısından önemli iyileşmeler sağlamaktadır.

Faster R-CNN mimarisine ait blok diyagramı Şekil 2.6'da gösterilmiştir.



Şekil 2.6. Faster R-CNN blok diyagramı (Chen vd. 2022).

Model, giriş görüntüsünden konvolüsyonel sinir ağı (CNN) ile öznitelik haritası (feature map) çıkarır. Bu harita iki bileşene iletilir. İlk olarak, RPN (Region Proposal Network) bu harita üzerinden nesne içerebilecek alanları belirler. Ardından, hem RPN'nin belirlediği bu alanlar hem de öznitelik haritası tam bağlantılı (fully connected) katmanlara aktarılır. Bu katmanlarda her bölge için sınıflandırma ve konum tahmini yapılır. Modelin iki aşamalı yapısı, özellikle yüksek konum doğruluğu gerektiren çalışmalarda daha etkili sonuçlar sağlar.

2.6.2. RetinaNet (Focal Loss Based One-Stage Object Detector)

RetinaNet, tek aşamalı nesne tespitinde hız ve doğruluk arasında dengeli bir performans sunan güncel ve yüksek verimli bir derin öğrenme mimarisidir. Modelin başarısının temelinde, nesne tespitinde yaygın olarak karşılaşılan sınıf dengesizliği sorununu etkili şekilde çözen *focal loss* fonksiyonu yer alır. Bu fonksiyon, kolay sınıflandırılan örneklerin etkisini azaltarak modelin zor örnekler odaklanmasını sağlar. RetinaNet'in mimarisi, öznitelik çıkarımı için ResNet tabanlı bir omurga ağı (*backbone*) ile çok ölçekli öznitelik temsili sunan *Feature Pyramid Network* (FPN) yapısını bir araya getirir. Model, her konumda nesne varlığını tahmin eden bir sınıflandırma alt ağı ve sınırlayıcı kutuları iyileştiren bir regresyon alt ağı içerir. Varsayılan olarak, farklı ölçek ve en-boy oranlarındaki *anchor* kutularıyla çeşitli boyutlardaki nesneleri algılar. Ancak bu varsayılan *anchor* parametreleri her veri seti için ideal olmayabilir. Bu nedenle, söz konusu çalışmada *differential evolution* (DE) tabanlı bir optimizasyon algoritması önerilmiş; *anchor* ölçekleri, en-boy oranları ve sayıları veri setine uygun şekilde otomatik olarak ayarlanmıştır. Yapılan deneyler, bu optimizasyonun RetinaNet'in hem küçük hem de büyük nesnelerde tespit başarısını anlamlı ölçüde artırdığını ve bazı durumlarda *anchor*'suz yaklaşımları dahi geride bıraktığını göstermektedir (Mohammed vd. 2025).

RetinaNet'in odaklandığı *focal loss* fonksiyonu, standart çapraz entropi kaybını (*cross-entropy loss*) modifiye ederek modelin zor örnekler üzerinde yoğunlaşmasını sağlar. Formül Denklem (2.11)'de verilmiştir:

$$FL(p_t) = -\alpha_t(1 - p_t)^\gamma \log(p_t) \quad (2.11)$$

Burada p_t , doğru sınıfa ait tahmin olasılığını; α_t , sınıf dengesizliğini azaltmak için kullanılan ağırlık katsayısını; γ ise kolay örneklerin etkisini azaltarak zor örneklerin kayıpta daha fazla ağırlık kazanmasını sağlayan odaklama parametresini ifade eder. Denklem (2.11)'de görüldüğü üzere bu yapı, modelin az sayıda görülen sınıfları ve tespit edilmesi güç nesneleri daha etkili şekilde öğrenmesini sağlar.

2.6.3. SSD (Single Shot MultiBox Detector)

SSD (Single Shot MultiBox Detector), nesne tespiti görevini hem sınıflandırma hem de konum belirleme işlemlerini aynı anda gerçekleştirerek hızlı bir şekilde tamamlayan bir mimaridir. Bu iki görevi eş zamanlı olarak yürütebilmesi, özellikle gerçek zamanlı sistemlerde kullanımı mümkün kılmaktadır. Hesaplama açısından fazla kaynak tüketmemesi, onu mobil ya da donanımsal olarak sınırlı sistemlerde de tercih edilebilir hâle getirmiştir (Liu vd. 2016).

Modelin farklı boyutlardaki nesneleri tanıyabilmesi, değişik seviyelerdeki katmanlardan elde edilen özellik haritalarının birlikte değerlendirilmesiyle sağlanır. Bu yöntemde her katman, belirli boyutlardaki nesneler için hem sınıf hem de konum tahmini yapar. Yüksek çözünürlüklü ve yüzeye yakın katmanlar küçük nesnelere odaklanırken, daha derin katmanlar geniş alanları kapsayan büyük nesneler hakkında bilgi üretir. Böylece tek bir model, farklı boyutlardaki nesneleri aynı anda tespit edebilir.

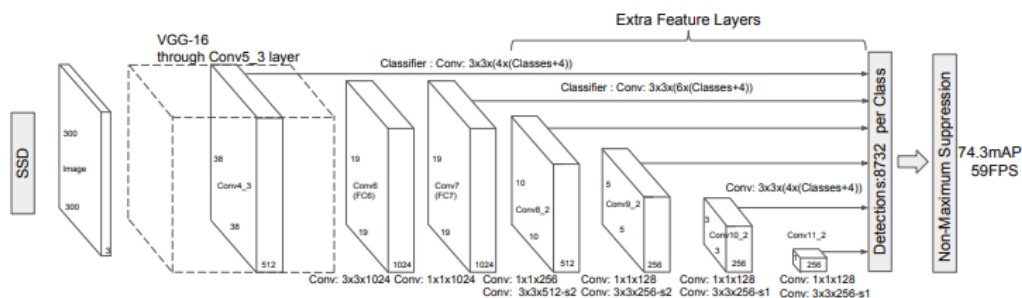
SSD, hız açısından önemli avantajlar sunsa da, özellikle küçük nesneleri tanımada sınırlı performans sergileyebilmektedir. Bu sorunun temel nedenlerinden biri, SSD'nin özellikle yüzeysel (shallow) katmanlardan elde edilen özniteliklerin sınırlı semantik bilgi sunmasıdır. Bu durum, özellikle küçük nesnelerin konumlandırılmasında modelin etkinliğini düşürebilir (Zhang vd. 2019).

SSD'nin başarımını belirleyen unsurlardan biri de kayıp fonksiyonudur. Model, sınıflandırma kaybı (L_{conf}) ve konum kaybı (L_{loc}) olmak üzere iki bileşeni birleştirerek toplam kaybı hesaplar. Bu yapı Denklem (2.12)'de tanımlandığı gibi ifade edilir:

$$L(x, c, l, g) = \frac{1}{N} (L_{\text{conf}}(x, c) + \alpha L_{\text{loc}}(x, l, g)) \quad (2.12)$$

Burada N pozitif eşleşmiş anchor sayısını, L_{conf} çapraz entropi sınıflandırma kaybını, L_{loc} ise smooth L1 konum kaybını ifade eder. α katsayısı bu iki bileşen arasındaki dengeyi sağlar (Liu vd. 2016). Denklem (2.12) ile SSD'nin toplam kayıp fonksiyonu hem sınıflandırma hem de konum hatalarını birlikte optimize edecek şekilde tanımlanmaktadır.

SSD mimarisinin VGG-16 tabanlı blok diyagramı Şekil 2.7’de gösterilmiştir.



Şekil 2.7. SSD mimarisinin VGG-16 tabanlı blok diyagramı (Liu vd. 2016).

SSD modelinde, VGG-16 omurgası ve farklı çözünürlüklerdeki katmanlar sayesinde, farklı boyutlardaki nesneler aynı anda tespit edilebilmektedir. Giriş görüntüsü önce VGG-16 katmanlarında işlenir, ardından elde edilen özellik haritaları (*feature map*) ek katmanlara aktarılır. Bu katmanlar, farklı boyutlardaki nesnelerin hem konumunu hem de sınıf bilgisini tahmin etmeyi sağlar (Liu vd. 2016).

2.6.4. YOLOv5'in Diğer Mimari Yapılarla Karşılaştırılması

Uydu görüntülerinde hava aracı tespitine yönelik farklı derin öğrenme yöntemlerinin başarımı, 2024 yılında yapılan karşılaştırmalı bir çalışmada incelenmiştir. HRPlanesV2 ve GDIT veri kümeleri kullanılarak Faster R-CNN, SSD, RetinaNet ve YOLOv5 modelleri aynı koşullarda değerlendirilmiştir. Sonuçlar, ortalama doğruluk (*mAP*), duyarlılık (*recall*) ve örtüşme oranı (*IoU*) ölçütlerinde YOLOv5'in diğer modellere göre daha iyi performans gösterdiğini ortaya koymuştur. Özellikle karmaşık arka planlı ve küçük nesnelerin yoğun olduğu uydu görüntülerinde YOLOv5'in tespit başarısı öne çıkmıştır (El Ghazouali vd. 2024).

Çizelge 2.1. Uçak tespitinde model performansları (El Ghazouali vd. 2024)

Mimari	Metrik	Eğitim Kümesi	Test Kümesi	Doğrulama Kümesi	Ortalama
YOLOv5	AP	0.968	0.956	0.942	0.955
	Duyarlılık	0.661	0.705	0.770	0.712
	IoU	0.740	0.752	0.742	0.745
SSD	AP	0.594	0.653	0.638	0.628
	Duyarlılık	0.386	0.432	0.496	0.438
	IoU	0.492	0.517	0.516	0.509
Faster R-CNN	AP	0.573	0.714	0.726	0.671
	Duyarlılık	0.422	0.413	0.491	0.442
	IoU	0.473	0.564	0.564	0.533
RetinaNet	AP	0.819	0.830	0.815	0.821
	Duyarlılık	0.566	0.620	0.644	0.610
	IoU	0.647	0.652	0.641	0.647

Çizelge 2.1, uzaktan algılama görüntülerinde uçak tespiti için kullanılan derin öğrenme tabanlı yöntemlerin *AP*, *duyarlılık* ve *IoU* performanslarını göstermektedir. Yapılan değerlendirmede YOLOv5, tüm metriklerde öne çıkmış ve en yüksek başarıyı sağlamıştır (El Ghazouali vd. 2024).

Tez kapsamında model seçimi yapılırken literatürdeki karşılaştırmalı sonuçlar ve uygulama gereksinimleri göz önünde bulundurulmuştur. YOLOv5, dengeli performansı ve gerçek zamanlı çalışmaya uygunluğu nedeniyle tercih edilmiştir.

3. MATERYAL VE METOT

3.1. Veri Setinin Hazırlanması

3.1.1. Veri Toplama ve Ön İşleme

Bu çalışmada kullanılan uydu görüntüleri, Google Earth Pro yazılımı aracılığıyla farklı bölgelerden manuel olarak toplanmıştır. Görseller, çeşitli ülkelerde yer alan hava ulaşım tesislerinden seçilerek, hazır veri setlerine bağlı kalmaksızın özgün bir veri seti oluşturulmuştur. Şekil 3.1’de veri setinden seçilmiş örnek uydu görüntüleri gösterilmektedir.



Şekil 3.1. Veri setinden seçilmiş örnek uydu görüntüleri

Tez kapsamında oluşturulan veri seti; askeri kargo uçağı, sivil yolcu uçağı ve askeri savaş uçağı olmak üzere üç sınıftan oluşmaktadır. Görseller PNG formatında kaydedilmiş ve YOLOv5 ile uyumlu olacak şekilde 832×832 piksel boyutlarına dönüştürülmüştür. Giriş boyutunun 832 piksel olarak seçilmesinin nedeni, küçük nesnelerin daha iyi temsil edilmesini sağlaması ve mevcut donanımına uygun olmasıdır.

Etiketleme işlemleri, her görüntüdeki nesnelerin manuel olarak işaretlenmesiyle gerçekleştirilmiştir. Eğitim sürecinde, modelin genelleme kabiliyetini artırmak amacıyla parlaklık değişimi, döndürme ve yakınlaştırma gibi dönüşümleri içeren veri artırımı yöntemleri kullanılmıştır. Bu sayede mevcut görüntülerden yeni varyasyonlar üretilmiş ve modelin farklı koşullara uyum sağlayabilmesi hedeflenmiştir.

3.1.2. Etiketleme Formatı (YOLO)

Uydu görüntülerinde yer alan sınıfların etiketlenmesi işlemi, açık kaynaklı *LabelImg* yazılımı kullanılarak manuel olarak gerçekleştirilmiştir. Etiketleme sürecinde her bir nesne, dikdörtgen sınırlayıcı kutular (bounding boxes) aracılığıyla işaretlenmiş ve sınıf bilgisiyle birlikte YOLO formatına uygun şekilde .txt uzantılı dosyalara kaydedilmiştir. Her dosya, ilgili görüntüyle aynı isimlendirme yapısını izlemekte olup, her satır bir nesneye karşılık gelmektedir. Bu arayüzün nasıl çalıştığını gösteren örnek bir ekran görüntüsü Şekil 3.2’de gösterilmiştir.



Şekil 3.2. LabelImg arayüzünde etiketlenmiş bir uydu görüntüsü örneği

YOLO formatında kullanılan etiket yapısı Çizelge 3.1’de gösterilmiştir. Bu yapıda nesnelerin merkez koordinatları ile genişlik ve yükseklik değerleri, görüntü boyutlarına göre 0 ile 1 aralığında normalize edilmiştir.

Çizelge 3.1. YOLO formatında etiket yapısı

Alan	Açıklama (TR)	Detaylı Tanım
class_id	Sınıf Kimliği	Nesnenin ait olduğu sınıfın sayısal karşılığı
x_center	Yatay Merkez Konumu	Nesnenin merkezinin yatay ekseninde normalize edilmiş değeri (0–1 arası)
y_center	Dikey Merkez Konumu	Nesnenin merkezinin dikey ekseninde normalize edilmiş değeri (0–1 arası)
width	Genişlik	Nesnenin görüntüye göre normalize edilmiş genişliği (0–1 arası)
height	Yükseklik	Nesnenin görüntüye göre normalize edilmiş yüksekliği (0–1 arası)

Etiketleme işlemi sırasında tüm görseller tek tek incelenmiş ve her bir nesne, sınıfına uygun olarak doğru biçimde işaretlenmiştir. Bu süreçte nesnelerin konum ve boyutları dikkatle belirlenmiş, etiketlerin tutarlılığı ve doğruluğu sürekli olarak kontrol edilmiştir. Böylece modelin eğitiminde hatalı veya eksik etiketlerin neden olabileceği performans kayıpları en aza indirilmiştir. Bu çalışmada kullanılan üç hava aracı sınıfı ve bu sınıflara ait sayısal karşılıklar Çizelge 3.2’de sunulmuştur.

Çizelge 3.2. Sınıf etiketleri ve karşılıkları

Sınıf Etiketi (class_id)	Sınıf Adı	Açıklama
0	Askeri Kargo Uçağı	Askerî malzeme ve personel taşıyan büyük gövdeli uçak
1	Sivil Yolcu Uçağı	Ticari yolcu taşımacılığında kullanılan hava aracı
2	Askeri Savaş Uçağı	Hızlı ve manevra kabiliyeti yüksek askeri hava aracı

3.1.3. Veri Artırımı

Veri artırımı, sınırlı örnek içeren görüntü veri kümelerinde modelin genelleme yeteneğini geliştirmek amacıyla yaygın olarak kullanılan bir yöntemdir. Bu kapsamda eğitim verilerine çeşitli dönüşümler uygulanarak yeni görüntüler oluşturulur. Döndürme, yeniden ölçekleme, nesne kaydırma ve şekil bozma gibi işlemlerle elde edilen yapay örnekler, modelin farklı koşullara uyum kabiliyetini artırmaktadır. Deneysel bulgular, her bir görüntüye iki ya da üç dönüşüm uygulanmasının doğruluk oranını belirgin şekilde yükselttiğini; daha fazla tekrarın ise hesaplama maliyetini artırmasına karşın sınırlı fayda sağladığını göstermektedir. Bu nedenle veri kümesinin 2–3 kat genişletilmesinin yeterli olduğu ve çoğu durumda basit dönüşüm tekniklerinin karmaşık yöntemlere kıyasla daha etkili sonuçlar verdiği belirtilmektedir (Hu vd. 2019).

Modelin genelleme kabiliyetini artırmak amacıyla eğitim sürecinde farklı veri artırımı yöntemlerinden yararlanılmıştır. YOLOv5, renk tonlamaları (hsv değişimleri), ölçekleme, döndürme, kaydırma ve ayna çevirme gibi parametreler aracılığıyla görüntülerdeki renk, ölçek ve yönelim farklılıklarına karşı dayanıklılık kazanmıştır. Bunun yanı sıra Albumentations kütüphanesiyle ek dönüşümler uygulanmış ve bu işlemler, YOLOv5’in yerleşik artırımıyla uyumlu olacak şekilde yapılandırılmıştır. Böylece veri çeşitliliği artırılarak modelin daha sağlam ve genellenebilir özellikler öğrenmesi sağlanmıştır. Parametre seçiminde literatürdeki öneriler ve görseller üzerinde yapılan ön denemeler dikkate alınmış; kullanılan veri artırımı parametreleri sırasıyla Çizelge 3.3 ve Çizelge 3.4’te verilmiştir.

Çizelge 3.3. YOLOv5 varsayılan veri artırımı parametreleri

Parametre	Açıklama	Değer
hsv_h	HSV renk tonu değişimi	0.015
hsv_s	HSV doygunluk değişimi	0.50
hsv_v	HSV parlaklık değişimi	0.35
degrees	Görüntü rotasyonu (°)	6.0
translate	Görüntü kaydırma oranı	0.06
scale	Görüntü ölçekleme oranı	0.15
shear	Görüntü eğrilme derecesi	2.0
perspective	Perspektif bozma oranı	0.0
flipud	Dikey çevirme olasılığı	0.05
fliplr	Yatay çevirme olasılığı	0.50
mosaic	Mosaic veri artırımı olasılığı	0.20
mixup	Mixup veri artırımı olasılığı	0.05
copy_paste	Nesne kopyala-yapıştır olasılığı	0.0

Çizelge 3.4. Albumentations kütüphanesi veri artırımı parametreleri

Dönüşüm	Parametreler	Olasılık (p)
RandomResizedCrop	Ölçek: (0.85–1.0), Oran: (0.90–1.10), 832×832	0.20
HorizontalFlip	—	0.50
VerticalFlip	—	0.10
ShiftScaleRotate	Kaydırma: ± 0.03 , Ölçek: ± 0.05 , Rot: $\pm 4^\circ$	0.25
ImageCompression	Kalite: 70–95	0.10
GaussNoise	Varyans: 5–15	0.08
RandomGamma	Gamma: 85–115	0.10
Blur	Max: 3	0.00
HueSaturationValue	Hue: ± 12 , Sat: ± 12 , Val: ± 8	0.15

Çizelge 3.3 YOLOv5'in varsayılan parametrelerini, Çizelge 3.4 ise Albumentations kütüphanesi ile eklenen dönüşümleri göstermektedir. Bu parametreler eğitim verisini çeşitlendirerek modelin genelleme yeteneğini artırmıştır. Renk dönüşümleri farklı ışık koşullarına uyum sağlarken, ölçekleme, kaydırma ve çevirme gibi geometrik işlemler nesnelerin farklı konum ve boyutlarda öğrenilmesine katkı sağlamıştır. Mosaic ve Mixup sahne çeşitliliğini artırarak aşırı uyumu önlemiş, Albumentations ile eklenen gürültü, bulanıklık ve kontrast değişimleri ise düşük kaliteli görüntülere karşı dayanıklılığı güçlendirmiştir.

Veri seti, 5 katlı çapraz doğrulama yöntemi ile değerlendirilmiş ve veri artırımı yalnızca eğitim kümelerine uygulanmıştır. Doğrulama kümeleri orijinal halleriyle korunmuş, herhangi bir artırma yapılmamıştır. Veri artırımı sonucunda eğitim kümesindeki örnek sayılarındaki değişim Çizelge 3.5'te verilmiştir.

Çizelge 3.5. Veri artırımı öncesi ve sonrası görüntü sayıları

İterasyon	Eğitim Kümesi		Doğrulama Kümesi	
	Öncesi	Sonrası	Öncesi	Sonrası
İterasyon 1	100	434	25	25
İterasyon 2	100	441	25	25
İterasyon 3	100	436	25	25
İterasyon 4	100	439	25	25
İterasyon 5	100	423	25	25
Ortalama	100	434.6	25	25

Not: 5 katlı çapraz doğrulama %80/%20 oranı yalnızca özgün (artırılmamış) veri bölümü için geçerlidir. Veri artırımı yalnızca eğitim kümesine uygulanmış, doğrulama kümesi sabit kalmıştır.

Çizelge 3.5'te veri artırımı öncesinde her iterasyonda eğitim kümesinde 100 ve doğrulama kümesinde 25 görüntü yer aldığı görülmektedir. Veri artırımı yalnızca eğitim kümesine uygulanmış, bu nedenle eğitim kümesindeki görüntü sayısı ortalama 434.6'ya çıkmıştır. Böylece model daha çeşitli örneklerle eğitilmiştir. Doğrulama kümesi ise değiştirilmemiş olup, doğrulama süreci orijinal veriler üzerinden sürdürülerek eğitim performansı nesnel bir biçimde takip edilmiştir.

**Şekil 3.3.** Veri artırımı ile oluşturulan örnek görüntüler

Şekil 3.3'te, döndürme, parlaklık ayarı ve yakınlaştırma gibi farklı veri artırımı yöntemleri sonucunda üretilmiş örnek görüntüler yer almaktadır. Bu yöntemler, modelin gerçek dünyadaki değişken koşullara uyum sağlamasına yardımcı olmuştur.

3.2. Kullanılan Yazılım ve Donanım Ortamı

Modelin eğitim ve test süreci, Python diliyle geliştirilen YOLOv5 mimarisi üzerinden yürütülmüştür. Çalışmalar Visual Studio Code ortamında, sanal Python ortamı içinde gerçekleştirilmiş; eğitimde PyTorch kullanılmış ve GPU desteği için CUDA ile cuDNN yapılandırılmıştır.

3.2.1. Donanım Özellikleri

Modelin eğitim ve doğrulama işlemleri, GPU destekli bir dizüstü bilgisayar ortamında gerçekleştirilmiştir. Bu süreçte kullanılan sistemin temel donanım özellikleri Çizelge 3.6'da gösterilmiştir.

Çizelge 3.6. Donanım özellikleri

Donanım	Özellik	Açıklama
İşlemci	Intel® Core™ i7-13650HX	Çok çekirdekli CPU
Ekran Kartı	NVIDIA® GeForce® RTX 4050 (6 GB GDDR6)	CUDA destekli GPU
RAM	16 GB DDR5	Sistem belleği
Depolama	512 GB NVMe SSD	Hızlı disk
İşletim Sistemi	Windows 11 Home	Kullanılan işletim sistemi
CUDA	11.8	GPU arayüzü
cuDNN	8.6.0	DL kütüphanesi

3.2.2. Yazılım Altyapısı ve Kütüphaneler

Modelin eğitimi, doğrulaması ve performans analizleri çeşitli yazılım araçları ve kütüphaneler kullanılarak gerçekleştirilmiştir. Bu süreçte açık kaynaklı ve yaygın olarak kullanılan yazılım araçlarından yararlanılmıştır.

Visual Studio Code:

Modelin hazırlanması, eğitimi ve test süreçleri Visual Studio Code geliştirme ortamında yürütülmüştür. Terminal erişimi, eklenti desteği ve hata ayıklama gibi özellikler sayesinde kodlama süreci daha verimli hâle gelmiştir (Microsoft 2021).

Python:

Python, geniş kütüphane desteği sayesinde yapay zekâ ve derin öğrenme projelerinde sıkça tercih edilen bir programlama dilidir (Anonim 2024). Çalışmada, model geliştirme ve veri işleme süreçlerinde temel programlama dili olarak kullanılmıştır.

PyTorch:

PyTorch, derin öğrenme modellerinin geliştirilmesi ve eğitimi için kullanılan açık kaynaklı bir kütüphanedir. Python ile uyumlu olması ve işlemlerin kolayca ayarlanabilmesi, model geliştirme sürecinin daha kontrollü yürütülmesini sağlamaktadır. YOLOv5 eğitimi tez kapsamında oluşturulan veri seti ile yapılmış, hiperparametre optimizasyonunda PyTorch altyapısı kullanılmıştır. GPU desteği sayesinde eğitim süreci daha verimli gerçekleştirilmiştir (Paszke vd. 2019).

TensorBoard:

TensorBoard, TensorFlow ekosistemi içerisinde geliştirilen ve model eğitimine ilişkin temel performans değerlerinin izlenmesine olanak tanıyan bir görselleştirme aracıdır. *mAP*, *kayıp (loss)*, *kesinlik (precision)* ve *duyarlılık (recall)* gibi metriklerin zaman içindeki değişimini takip etmeye, eğitim sürecini analiz etmeye ve modelin gelişimini değerlendirmeye yardımcı olmaktadır. Çalışmada, model eğitimi sırasında elde edilen verilerin görsel olarak izlenebilmesi ve karşılaştırılabilmesi amacıyla TensorBoard kullanılmıştır (TensorFlow 2024).

GluonCV (Gluon Computer Vision):

GluonCV, Apache MXNet platformu üzerine kurulmuş açık kaynaklı bir bilgisayarla görme kütüphanesidir. Amazon Web Services (AWS) tarafından desteklenen bu kütüphane, özellikle görüntü işleme uygulamalarında tercih edilmektedir. GluonCV içerisinde YOLO, SSD ve Faster R-CNN gibi yaygın olarak kullanılan nesne tespiti modelleri eğitilmiş olarak verilmektedir. Bu kütüphane, model mimarisinin değiştirilebilmesine ve öğrenme oranı, batch boyutu ya da veri artırma yöntemleri gibi birçok parametre ayarının düzenlenebilmesine imkan sağlamaktadır. Bu sayede GluonCV, sadece hazır modelleri çalıştırmak için değil, aynı zamanda farklı hiperparametre ayarlarını deneyip karşılaştırmak için de kullanılacak bir araç hâline gelmiştir (He vd. 2019).

CUDA ve cuDNN:

PyTorch'un CUDA desteği ile model eğitimi sırasında gerçekleştirilen hesaplamalar GPU üzerinden yapılmıştır. Ayrıca, NVIDIA tarafından geliştirilen cuDNN kütüphanesi sayesinde bu hesaplamalar daha verimli ve hızlı bir şekilde gerçekleştirilmiştir (NVIDIA 2024).

NumPy ve Pandas:

NumPy, Python’da bilimsel hesaplamalar için kullanılan temel bir kütüphanedir. Görüntü verilerinin piksel matrisleri gibi çok boyutlu yapılarda işlenmesini ve hesaplamaların hızlı gerçekleştirilmesini sağlar. Pandas ise veri analizi ve işleme için geliştirilmiş açık kaynaklı bir kütüphanedir. *Series* ve *DataFrame* yapıları sayesinde etiketli veriler üzerinde düzenleme, filtreleme ve dönüştürme işlemleri kolayca yapılabilir. Çalışmada, görüntü verilerinin matris formatında işlenmesi ve sayısal verilerin modellenmesinde NumPy; etiket dosyalarının okunması, düzenlenmesi ve eğitim verisinin hazırlanmasında ise Pandas kullanılmıştır (Harris vd. 2020; McKinney 2010).

Matplotlib ve Seaborn:

Matplotlib, Python dili için geliştirilmiş, çizgi grafiği, histogram, dağılım grafiği ve kutu grafiği gibi çok sayıda görselleştirme türünü destekleyen bir kütüphanedir. Seaborn ise Matplotlib üzerine kurulmuş, istatistiksel veri görselleştirmelerini daha sade ve anlaşılır bir yapıda sunmayı amaçlayan bir kütüphanedir. Pandas veri yapılarıyla uyumlu çalıştığı için, analiz edilen bilgilerin görsel olarak sunulmasını kolaylaştırmaktadır. Çalışmada, eğitim sürecine ait performans metriklerinin görselleştirilmesinde Matplotlib ve Seaborn kütüphanelerinden yararlanılmıştır (Hunter 2007; Waskom 2021).

Albumentations Kütüphanesi:

Albumentations, veri artırımı işlemleri için kullanılan açık kaynaklı bir Python kütüphanesidir. Görüntü sınıflandırma ve nesne tespiti gibi bilgisayarla görme alanındaki uygulamalarda yaygın olarak tercih edilen bu kütüphane; döndürme, yansıtma, bulanıklaştırma, parlaklık-kontrast ayarı ve bozulma gibi çeşitli dönüşümleri desteklemektedir (Buslaev vd. 2020).

3.3. Model Eğitim Süreci**3.3.1. YOLOv5 Mimarisinin Yapılandırılması**

YOLOv5 modeli, omurga (backbone) katmanı olarak CSPDarknet53 mimarisini kullanmaktadır. Bu yapı, derinlik ve verimlilik arasında denge kurarak, farklı seviyelerdeki görsel özelliklerin daha etkili biçimde öğrenilmesini sağlar. PANet (Path Aggregation Network) ve SPPF (Spatial Pyramid Pooling Fast) katmanları, farklı boyutlardaki nesnelerin aynı anda ve daha sağlıklı biçimde algılanabilmesi için, çeşitli ölçeklerde elde edilen özellik haritalarını bir araya getirmektedir. Bu sayede model, hem küçük hem de büyük nesneleri daha doğru bir şekilde tanımlayabilmektedir.

Çalışmada nesne tespiti amacıyla YOLOv5 mimarisi kullanılmıştır. Eğitimde kullanılan görüntüler, başlangıçta 1280×720 piksel çözünürlüğe sahip olmakla birlikte, YOLO mimarisinin gereksinimleri doğrultusunda 832×832 piksele ölçeklendirilmiştir. Daha yüksek çözünürlük seçimi, özellikle küçük nesnelerin daha iyi temsil edilmesi ve modelin genel doğruluk performansının artırılması amacıyla tercih edilmiştir. Modelin çıkış katmanları, bu çalışmada tanımlanan üç sınıfı ayırt edebilecek şekilde düzenlenmiştir. Ağ yapısı, farklı boyutlardaki nesneleri algılayabilmek amacıyla her bir ölçekte nesnelerin konumunu, boyutunu ve hangi sınıfa ait olduklarını ayrı ayrı tahmin etmektedir. Bu tahminler sayesinde, farklı büyüklükteki nesnelerin aynı anda doğru bir şekilde tespit edilebilmesi amaçlanmaktadır.

3.3.2. Genetik Algoritma ile Hiperparametre Optimizasyonu

Bir derin öğrenme modelinin başarısı yalnızca mimarisine değil, aynı zamanda hiperparametre değerlerine de bağlıdır. Doğru ayarlanan hiperparametreler, modelin farklı veri kümelerinde dengeli ve genellenebilir sonuçlar üretmesini sağlar. Yanlış ayarlamalar ise öğrenmenin yetersiz kalmasına ya da aşırı uyuma yol açarak modelin sadece eğitim verisine bağımlı hale gelmesine neden olur. Bu nedenle hiperparametreler, derin öğrenme modellerinin güvenilir ve uyarlanabilir performansı için kritik öneme sahiptir.

YOLOv5 mimarisinde varsayılan olarak tanımlanan ve model eğitiminde başlangıç noktası olarak kullanılan hiperparametre değerleri Çizelge 3.7’de gösterilmiştir. Bu değerler, çalışmada yürütülen hiperparametre optimizasyonu için başlangıç referansı olarak kabul edilmiştir.

Çizelge 3.7. YOLOv5 başlangıç hiperparametreleri

Parametre	Değer	Açıklama
lr0	0.01	Başlangıç öğrenme oranı
lrf	0.10	Son öğrenme oranı çarpanı
momentum	0.937	SGD momentum / Adam β_1
weight_decay	0.0005	Ağırlık cezası (düzenleştirme)
warmup_epochs	3.0	Isınma (warmup) epoch sayısı
warmup_momentum	0.80	Isınmada başlangıç momentum değeri
warmup_bias_lr	0.10	Isınmada <i>bias</i> için başlangıç öğrenme oranı
box	0.05	Kutu (bounding box) kaybı ağırlığı
cls	0.30	Sınıflandırma (class) kaybı ağırlığı
cls_pw	1.00	Sınıflandırma için pozitif ağırlık (BCE)
obj	0.70	Nesne (object) kaybı ağırlığı
obj_pw	1.00	Nesne için pozitif ağırlık (BCE)
iou_t	0.20	IOU eğitim eşiği
anchor_t	4.0	Anchor çarpan eşiği
fl_gamma	0.0	Focal loss γ
anchors	3	Anchor sayısı

Öğrenme oranı, momentum ve ağırlık çarpanı gibi parametreler modelin öğrenme sürecinin hızını ve kararlılığını belirlerken; kutu, nesne ve sınıf kayıplarına ait katsayılar ise farklı hata türleri arasındaki dengeyi sağlamaktadır. Ayrıca *warmup* parametreleri, modelin eğitime daha istikrarlı başlamasına katkı sağlamaktadır. *fl_gamma* ve *anchor_t* gibi değerler ise daha hassas öğrenme süreçlerini düzenlemektedir. Çizelge 3.7’de verilen hiperparametre değerleri eğitim süreci için bir referans noktası oluşturmaktadır. Ancak farklı veri kümeleri ve deneysel koşullarda her zaman en iyi sonuçları vermeyebilir. Bu nedenle çalışmada, modelin daha yüksek doğruluk ve dengeli bir şekilde öğrenmesini sağlamak amacıyla hiperparametrelerin optimizasyonunda genetik algoritma (GA) tabanlı bir yaklaşım kullanılmıştır. Bu yöntemde her birey (kromozom), farklı bir hiperparametre kombinasyonunu temsil etmektedir. Bireylerin başarımı ise doğrulama veri seti üzerinde hesaplanan *mAP@0.5* metriği kullanılarak değerlendirilmiştir. Seçilim, çaprazlama ve mutasyon gibi temel GA işlemleriyle yeni bireyler oluşturulmuştur. Bu evrimsel süreç, en fazla 60 nesil olacak şekilde yürütülmüştür.

Genetik algoritmanın uygulanabilmesi için her bir hiperparametreye ait alt ve üst sınırların tanımlandığı bir arama uzayı oluşturulmuştur. Bu sınırların belirlenmesinde literatürde YOLO ve hiperparametre optimizasyonuna ilişkin önerilen değerler ile bu çalışma kapsamında yürütülen deneysel süreçten elde edilen gözlemler dikkate alınmıştır. Kullanılan hiperparametre arama uzayı Çizelge 3.8’de gösterilmiştir.

Çizelge 3.8. GA hiperparametre arama uzayı

Parametre	Alt Sınır	Üst Sınır
lr0	0.004	0.020
lrf	0.08	0.22
momentum	0.920	0.970
weight_decay	1e-4	1e-3
warmup_epochs	1.0	4.0
warmup_momentum	0.70	0.90
warmup_bias_lr	0.05	0.15
box	0.04	0.12
cls	0.35	0.65
cls_pw	1.0	1.3
obj	0.75	1.15
obj_pw	0.9	1.2
iou_t	0.22	0.32
anchor_t	3.0	5.0
fl_gamma	0.0	2.0
anchors	3.0	5.0

Öğrenme oranı ($lr0$) ve momentum gibi temel optimizasyon parametreleri geniş fakat mantıklı sınırlar içinde tutulmuş, ağırlık çarpanı ($weight_decay$) ve $warmup$ parametreleri ise modelin daha dengeli bir başlangıç yapabilmesi için sınırlandırılmıştır. Kayıp fonksiyonlarına ilişkin katsayılar (box , cls , obj) için de belirli bir aralık tanımlanarak, farklı hata türlerinin öğrenme üzerindeki etkisi optimize edilmeye çalışılmıştır. Ayrıca, fl_gamma ve $anchors$ gibi parametrelerin de arama uzayına dahil edilmesi, modelin nesne tespit performansını farklı veri özelliklerine göre uyarlayabilmesine olanak sağlamaktadır.

Genetik algoritmanın uygulanabilmesi için konfigürasyon parametreleri tanımlanmıştır. Bu parametreler, popülasyon büyüklüğü, mutasyon ve çaprazlama oranları ile elit bireylerin korunma seviyelerini içermektedir. Belirlenen aralıklar, literatürde önerilen genel GA parametre değerleri ve bu çalışma kapsamında yapılan ön denemelerin sonuçları dikkate alınarak seçilmiştir. Örneğin, popülasyon büyüklüğü 18 belirlenmiş ve bu değer her nesilde yeterli çeşitlilik sağlarken hesaplama maliyetini makul seviyede tutmuştur. Mutasyon oranı %12–%20 aralığında bırakılarak çözümlerde hem yeni kombinasyonların keşfi sağlanmış hem de aşırı rastgeleleşmenin önüne geçilmiştir. Çaprazlama oranı sabit 0.85 değerinde tutulmuş, böylece iyi çözümlerden bilgi aktarımı kararlı sağlanmıştır. Elit birey sayısı 2–3 ile sınırlandırılarak başarılı çözümler korunurken popülasyonun çeşitliliği bozulmamıştır. Turnuva seçiminde 3–4 birey kullanılması seçim baskısını orta seviyede tutarak denge sağlamıştır. Kullanılan değerler Çizelge 3.9’da verilmiştir.

Çizelge 3.9. Genetik algoritma konfigürasyon parametreleri

Parametre	Değer	Açıklama
pop_size	18	Popülasyon büyüklüğü
mutation_rate_min	0.12	Minimum mutasyon oranı
mutation_rate_max	0.20	Maksimum mutasyon oranı
crossover_rate_min	0.85	Minimum çaprazlama oranı
crossover_rate_max	0.85	Maksimum çaprazlama oranı
min_elite_size	2	Minimum elit birey sayısı
max_elite_size	3	Maksimum elit birey sayısı
tournament_size_min	3	Minimum turnuva boyutu
tournament_size_max	4	Maksimum turnuva boyutu

Çalışmada sabit bir test kümesi ayrılmış, kalan veriler üzerinde 5 katlı çapraz doğrulama uygulanmıştır. Her iterasyonda doğrulama kümesi farklı olacak şekilde eğitim–doğrulama ayrımı yapılmıştır. Bu nedenle sabit bir hiperparametre seti yerine her iterasyon için GA tabanlı optimizasyon ayrı ayrı uygulanmış ve kendi eğitim–doğrulama dağılımına en uygun hiperparametreler belirlenmiştir. Nihai performans değerlendirmesi tüm modeller için sabit tutulan test kümesi üzerinde yapılmıştır. Çizelge 3.10’da beş iterasyonda GA ile optimize edilen hiperparametreler gösterilmektedir.

Çizelge 3.10. GA ile optimize edilmiş hiperparametreler

Parametre	İterasyon 1	İterasyon 2	İterasyon 3	İterasyon 4	İterasyon 5
lr0	0.01934	0.012627	0.00948	0.00869	0.013001
lrf	0.11046	0.188	0.21372	0.22000	0.15489
momentum	0.92000	0.97000	0.97000	0.95461	0.96495
weight_decay	0.00034	0.00090301	0.00010	0.00025	0.00051422
warmup_epochs	2.3276	1.5633	1.4222	3.4126	2.8124
warmup_momentum	0.8614	0.7723	0.85435	0.7169	0.7000
warmup_bias_lr	0.138	0.085086	0.07634	0.0668	0.05938
box	0.040	0.04000	0.04000	0.060	0.040
cls	0.6204	0.57253	0.63079	0.6183	0.63301
cls_pw	1.0477	1.2891	1.0707	1.146	1.2446
obj	1.1405	0.97271	0.78325	1.0405	0.81819
obj_pw	0.9399	1.1283	1.0986	1.2000	1.0684
iou_t	0.2245	0.2569	0.22000	0.2745	0.3200
anchor_t	4.1177	3.2984	4.2776	3.6177	3.7496
fl_gamma	1.4517	1.555	1.6236	1.4517	1.1243
anchors	4.0254	4.16	3.1039	3.0000	3.1849

Çizelge incelendiğinde, her iterasyon için elde edilen hiperparametre değerlerinin farklılık gösterdiği görülmektedir. Bu farklılıklar, her iterasyonda eğitim ve doğrulama ayrımlarının değişmesinden kaynaklanmaktadır. Bazı iterasyonlarda öğrenme oranı ve momentum gibi optimizasyon parametrelerinde değişiklikler öne çıkarken, diğerlerinde kutu, sınıf ve nesne kaybı ağırlıkları ile kayıp fonksiyonunu dengeleyen katsayılar da farklılıklar gözlenmiştir. Bu sonuç, genetik algoritmanın her iterasyon için en uygun hiperparametreleri ayrı ayrı belirleyebildiğini göstermektedir.

3.3.3. Eğitim ve Doğrulama

Eğitim işlemleri, GPU hızlandırmalı bir donanım ortamında yürütülmüş; *batch size* değeri 16 olarak belirlenmiş ve model toplam 250 *epoch* boyunca eğitilmiştir. Her bir *epoch* sonunda doğrulama kümesi üzerinde modelin başarımı izlenmiş ve başlıca metrikler olan *mAP*, *F1-skor*, *kesinlik* ve *duyarlılık* değerleri hesaplanmıştır.

Bu çalışmada kullanılan veri seti, eğitim (*train*), doğrulama (*validation*) ve test (*test*) olmak üzere üç alt kümeye ayrılmıştır. İlk aşamada toplam 33 görüntü sabit bir test kümesi olarak ayrılmış ve eğitim sürecinden tamamen bağımsız tutulmuştur. Kalan 125 görüntü üzerinde 5 katlı çapraz doğrulama uygulanmış; her yinelemede bu verilerin dördü eğitim, biri doğrulama amacıyla kullanılmıştır. Eğitim sürecinde model, eğitim kümesi üzerinde optimize edilmiş; her *epoch* sonunda doğrulama kümesi kullanılarak modelin ara performansı izlenmiştir. Bu aşamada elde edilen doğrulama sonuçları, aşırı öğrenmeyi (*overfitting*) engellemek ve hiperparametre ayarlarını yönlendirmek amacıyla kullanılmıştır. Eğitim tamamlandıktan sonra, en iyi doğrulama performansına sahip model ağırlıkları

kullanılarak sabit test kümesi üzerinde nihai değerlendirme yapılmıştır. Böylece modelin genelleme kabiliyeti objektif bir şekilde ölçülmüş ve farklı deneysel senaryoların performans karşılaştırmalarında yalnızca test kümesinden elde edilen sonuçlar esas alınmıştır.

5 Katlı Çapraz Doğrulama Yöntemi (5-Fold Cross Validation):

Sınırlı sayıda örnek içeren veri kümelerinde sabit oranlı tek bir eğitim, doğrulama ve test ayrımı, modelin genelleme yeteneğini tam olarak yansıtmayabilir. Bu çalışmada, modelin güvenilir şekilde değerlendirilmesi amacıyla sabit bir test kümesi ayrılmış ve kalan veri üzerinde 5 katlı çapraz doğrulama yöntemi uygulanmıştır. Toplam 158 görüntüden 33 tanesi bağımsız bir test kümesi olarak belirlenmiş ve eğitim sürecinden tamamen ayrı tutulmuştur. Kalan 125 görüntü beş eşit parçaya ayrılmış; her yinelemede bu parçaların dördü (100 görüntü) eğitim, biri (25 görüntü) doğrulama için kullanılmıştır. Doğrulama kümesi, eğitim sırasında modelin ara değerlendirmelerini yapmak ve aşırı öğrenme eğilimini kontrol etmek amacıyla kullanılmıştır. Süreç toplam beş kez tekrarlanmış, böylece her örnek en az bir kez doğrulama sürecine dahil edilmiş, test kümesi ise yalnızca nihai performans değerlendirmesinde kullanılmıştır. Bu yöntem, sonuçların güvenilirliğini artırmış ve modelin genelleme kabiliyetini daha sağlıklı biçimde ortaya koymuştur. İterasyonlara ait eğitim ve doğrulama kümesi dağılımları Çizelge 3.11’de gösterilmiştir.

Çizelge 3.11. 5 katlı çapraz doğrulama iterasyonları

İterasyon	Doğrulama (25 görüntü)	Eğitim (100 görüntü)
İterasyon 1	Fold 1	Fold 2 + Fold 3 + Fold 4 + Fold 5
İterasyon 2	Fold 2	Fold 1 + Fold 3 + Fold 4 + Fold 5
İterasyon 3	Fold 3	Fold 1 + Fold 2 + Fold 4 + Fold 5
İterasyon 4	Fold 4	Fold 1 + Fold 2 + Fold 3 + Fold 5
İterasyon 5	Fold 5	Fold 1 + Fold 2 + Fold 3 + Fold 4

Her iterasyonda model sıfırdan eğitilmiş ve bağımsız test kümesi üzerinde $mAP@0.5$, $mAP@0.5:0.95$, *kesinlik*, *duyarlılık* ve *F1-skor* gibi temel başarı metrikleri kullanılarak değerlendirilmiştir. Böylece her fold, modelin performansını bağımsız bir doğrulama sürecinden geçirmiştir. Beş iterasyondan elde edilen sonuçlar bir araya getirilmiş ve ortalaması alınarak modelin genel başarısı yorumlanmıştır. Bu yöntem, sonuçların yalnızca tek bir veri bölmesine bağlı kalmadan, veri kümesinin tamamına daha dengeli ve güvenilir şekilde genellenmesini sağlamıştır.

3.4. Performans Değerlendirme Kriterleri

Nesne tespiti performansını değerlendirmek için literatürde yaygın kullanılan ve farklı açılardan ölçüm yapmaya olanak tanıyan metrikler tercih edilmiştir. Bu metrikler yalnızca modelin genel doğruluğunu göstermekle kalmaz, aynı zamanda her sınıftaki nesnelerin ne kadar dengeli ve doğru tespit edildiğini de ortaya koyar. Böylece modelin güçlü ve zayıf yönleri daha net anlaşılır. Nesne tespiti, yalnızca doğru pozitiflerin artırılmasına odaklanan basit bir süreç değil; yanlış pozitiflerin (yanlış tespitler) ve yanlış negatiflerin (kaçırılan nesneler) de dikkate alınmasını gerektiren çok boyutlu bir problem-dır. Bu nedenle çalışmada kullanılan kriterler, modelin performansını kapsamlı biçimde değerlendirmeyi amaçlamaktadır. Böylece YOLOv5 ile eğitilen modeller tek bir ölçüte indirgenmeden farklı yönleriyle güvenilir şekilde değerlendirilmiştir.

Çalışmada esas alınan temel metrikler ve kısa açıklamaları Çizelge 3.12’de verilmiştir.

Çizelge 3.12. Performans değerlendirme metrikleri

Metrik	Açıklama
mAP@0.5	IoU eşiği %50 iken hesaplanan ortalama doğruluk değeri.
mAP@0.5:0.95	IoU eşiği %50–%95 arasında 0.05 adımlarla hesaplanan ortalama doğruluk.
Kesinlik (Precision)	Doğru tespitlerin tüm tespitlere oranı.
Duyarlılık (Recall)	Doğru tespitlerin tüm gerçek nesnelere oranı.
F1-Skoru (F1-Score)	Kesinlik ve duyarlılığın harmonik ortalaması.

Metriklerin birlikte değerlendirilmesi, modelin hem *kesinlik* hem de *duyarlılık* açısından dengeli bir performans sergileyip sergilemediğini ortaya koyar. Örneğin, yüksek *kesinlik* ancak düşük *duyarlılık*, çok sayıda nesnenin gözden kaçmasına neden olurken; yüksek *duyarlılık* ancak düşük *kesinlik*, gereksiz tespitlerin artmasına yol açar. Bu durum özellikle güvenlik ve savunma gibi kritik alanlarda kabul edilemez. *F1-skor*, *kesinlik* ve *duyarlılık* arasındaki dengeyi tek bir ölçüde özetlediği için bu açıdan önemlidir.

mAP@0.5 ve *mAP@0.5:0.95* metriklerinin birlikte kullanılması, modelin farklı IoU eşiklerinde ne kadar kararlı çalıştığını gösterir. Böylece hem daha esnek bir eşikte (%50) hem de daha zorlayıcı koşullarda (%95’e kadar) performans değerlendirmesi yapılabilir. Bu iki metriğin birlikte incelenmesi, modelin farklı boyutlardaki nesnelerdeki performansının daha kapsamlı bir şekilde değerlendirilmesine olanak sağlamaktadır.

3.4.1. mAP (mean Average Precision) Hesaplaması

Nesne tespiti başarımını değerlendirmek için temel ölçüt olarak *mAP* kullanılmıştır. Bu metrik, hem doğru sınıf tahminlerini hem de nesnelerin konumsal doğruluğunu dikkate alır. *mAP*, her sınıf için hesaplanan *AP* değerlerinin ortalamasıdır. *AP* ise, bir sınıfa ait tespitlerde *kesinlik* ve *duyarlılık* ilişkisinden elde edilen eğri altı alanı ifade eder.

Precision (*P*) ve *recall* (*R*) metrikleri Denklem (3.1) ve Denklem (3.2)'de tanımlanmıştır:

$$P = \frac{TP}{TP + FP} \quad (3.1)$$

$$R = \frac{TP}{TP + FN} \quad (3.2)$$

Burada *TP* doğru pozitif sayısını, *FP* yanlış pozitif sayısını, *FN* ise yanlış negatif sayısını göstermektedir.

Kesinlik ve *duyarlılık* birlikte değerlendirildiğinde, modelin hem doğruluk hem de kapsama başarısını daha dengeli şekilde ifade eden *F1-skor* Denklem (3.3)'te verilmiştir:

$$F1 = 2 \cdot \frac{\text{Kesinlik} \cdot \text{Duyarlılık}}{\text{Kesinlik} + \text{Duyarlılık}} \quad (3.3)$$

AP, belirli bir *IoU* (*Intersection over Union*) eşliğinde *kesinlik-duyarlılık* eğrisinin altında kalan alanın hesaplanmasıyla elde edilir. Bu tanım Denklem (3.4)'te gösterilmiştir:

$$AP = \int_0^1 p(r) dr \quad (3.4)$$

Tüm sınıflar için hesaplanan *AP* değerlerinin ortalaması alınarak *mAP* değeri elde edilir. Denklem (3.5)'te bu ifade yer almaktadır:

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (3.5)$$

mAP@0.5, *IoU* eşliği 0.5 alındığında hesaplanır; yani tahmin kutusu gerçek kutunun en az yarısını örtüyorsa doğru kabul edilir. Daha kapsamlı olan *mAP@0.5:0.95* ise *IoU* değerini 0.5'ten 0.95'e 0.05'lik adımlarla değiştirerek elde edilen *AP* değerlerinin ortalamasını alır ve modeli farklı hassasiyet düzeylerinde tek bir ölçüt ile değerlendirir. Denklem (3.6)'da bu hesaplama gösterilmiştir:

$$mAP@[0.5:0.95] = \frac{1}{10} \sum_{t=0.5}^{0.95} AP_{IoU=t} \quad (3.6)$$

Burada *t* değişkeni farklı *IoU* eşiklerini, *AP_{IoU=t}* ise bu eşığe karşılık gelen ortalama doğruluğu ifade eder.

4. BULGULAR

4.1. Eğitim Aşamaları ve Sonuçları

Bu bölümde, YOLOv5 mimarisi üç farklı deneysel çalışma kapsamında uygulanarak elde edilen sonuçlar karşılaştırılmıştır. İlk aşamada herhangi bir değişiklik yapılmadan varsayılan hiperparametreler kullanılmış ve bu aşama sonraki çalışmalar için başlangıç noktası olarak değerlendirilmiştir. İkinci aşamada modelin genelleme yeteneğini artırmak amacıyla çeşitli veri artırımı tekniklerinden yararlanılmıştır. Üçüncü aşamada ise veri artırımı uygulamalarına ek olarak genetik algoritma tabanlı hiperparametre optimizasyonu gerçekleştirilmiştir. Her bir aşama, modelin doğruluk, kesinlik ve genelleme performansı üzerindeki etkilerini ortaya koymak amacıyla ayrı ayrı değerlendirilmiş ve sonuçlar karşılaştırmalı olarak analiz edilmiştir. Eğitim ve doğrulama süreçleri 5 katlı çapraz doğrulama yöntemiyle yürütülmüş; her iterasyon sonunda elde edilen en iyi model ağırlıkları sabit tutulan test kümesi üzerinde değerlendirilmiştir. Beş iterasyondan elde edilen test sonuçlarının ortalaması alınarak nihai performans raporlanmış ve modelin genel başarımı güvenilir biçimde ortaya konulmuştur.

- **Deney 1:** YOLOv5 mimarisinde varsayılan olarak tanımlanan ve bu çalışmada başlangıç noktası olarak kullanılan hiperparametreler ile yapılan eğitim.
- **Deney 2:** YOLOv5 mimarisi üzerinde veri artırımı teknikleri uygulanarak gerçekleştirilen eğitim.
- **Deney 3:** YOLOv5 mimarisi, veri artırımı ve genetik algoritma tabanlı hiperparametre optimizasyonu birlikte kullanılarak yapılan eğitim.

4.1.1. YOLOv5 Varsayılan Ayarlar ile Model Eğitimi

Bu deneyde, modelin eğitimi sırasında herhangi bir veri artırma tekniği ya da hiperparametre optimizasyonu kullanılmamıştır. Yalnızca YOLOv5 mimarisinde varsayılan olarak tanımlanan hiperparametreler ile eğitim yapılmış ve bu ayarlar çalışmada başlangıç noktası olarak kabul edilmiştir. Bu yaklaşımın amacı, ilerleyen aşamalarda uygulanacak iyileştirmelerin etkisini karşılaştırabilmek için temel bir performans düzeyi belirlemektir. Eğitim süreci 5 katlı çapraz doğrulama yöntemiyle yürütülmüş; her iterasyonda elde edilen en iyi model ağırlıkları sabit test kümesi üzerinde değerlendirilmiştir. Beş iterasyondan elde edilen test sonuçlarının ortalaması alınarak modelin genel performansını gösteren değerler Çizelge 4.1’de verilmiştir.

Çizelge 4.1. Deney 1 için ortalama metrik sonuçları

Metrik	Ortalama Değer
mAP@0.5	0.755
mAP@0.5:0.95	0.453
Kesinlik	0.716
Duyarlılık	0.705
F1-skör	0.706

Çizelge 4.1 incelendiğinde, mAP@0.5 değerinin 0.755 seviyesinde olduğu görülmektedir. Bu oran modelin temel nesne tespitinde kabul edilebilir bir başarı sağladığını gösterirken, mAP@0.5:0.95 değerinin 0.453 seviyesinde kalması özellikle daha zorlayıcı eşiklerde modelin performansının sınırlı olduğunu ortaya koymaktadır. Kesinlik (0.716) ve duyarlılık (0.705) değerlerinin birbirine yakın olması, modelin yanlış pozitif ve yanlış negatif tahminler arasında dengeli davrandığını göstermektedir. Buna karşılık F1-skörünün 0.706 düzeyinde kalması, modelin genel başarısının orta seviyede olduğunu ve iyileştirmeye açık olduğunu işaret etmektedir. Dolayısıyla bu deney, sonraki aşamalarda uygulanacak veri artırımı ve hiperparametre optimizasyonu gibi yöntemlerin etkisini ölçmek için temel bir referans niteliği taşımaktadır.

Çizelge 4.2, askeri kargo uçağı, sivil yolcu uçağı ve askeri savaş uçağı olmak üzere üç farklı sınıf için gerçekleştirilen beş katlı çapraz doğrulama sonucunda elde edilen ortalama mAP, kesinlik, duyarlılık ve F1-skör metriklerini göstermektedir.

Çizelge 4.2. Deney 1 için sınıf bazında ortalama metrik sonuçları

Metrik	Askeri Kargo Uçağı	Sivil Yolcu Uçağı	Askeri Savaş Uçağı
mAP@0.5	0.856	0.764	0.646
mAP@0.5:0.95	0.576	0.439	0.343
Kesinlik	0.746	0.698	0.689
Duyarlılık	0.834	0.728	0.554
F1-skör	0.774	0.699	0.569

Çizelge 4.2 incelendiğinde, en yüksek başarıya *Askeri Kargo Uçağı* sınıfında ulaşıldığı görülmektedir. Bu sınıfta mAP@0.5 değeri 0.856 ve duyarlılık değeri 0.834 olup, modelin bu sınıfı yüksek doğrulukla ayırt edebildiği anlaşılmaktadır. *Sivil Yolcu Uçağı* sınıfında mAP@0.5 değeri 0.764 olarak hesaplanmış, kesinlik ve duyarlılık değerleri dengeli seyretmiş ancak genel performans kargo uçaklarına kıyasla daha düşük kalmıştır. En zayıf sonuçlar ise *Askeri Savaş Uçağı* sınıfında elde edilmiştir. Bu sınıfta mAP@0.5 değeri 0.646, duyarlılık değeri ise yalnızca 0.554 seviyesinde kalmış, bu da modelin birçok örneği tespit edemediğini göstermektedir. Genel olarak sonuçlar, modelin büyük boyutlu

ve belirgin görsel özelliklere sahip uçakları daha kolay tanıyabildiğini, görsel benzerlikleri yüksek olan veya daha küçük ölçekli nesnelerde ise performansın sınırlı kaldığını ortaya koymaktadır.

Elde edilen model çıktılarının görselleştirilmiş sonuçlarını göstermektedir. Görselde farklı havaalanı sahnelerinden alınan örneklerde üç sınıfa ait tespitler ile modelin tahmin ettiği etiketler ve güven skorları Şekil 4.1’de gösterilmektedir.



Şekil 4.1. Deney 1 için nesne tespit çıktılarına ait örnek görseller

Şekil 4.1 incelendiğinde modelin üç sınıfı da ayırt edebildiği ancak güven skorlarının genel olarak düşük kaldığı görülmektedir. Askeri savaş uçaklarında tahminler çoğunlukla 0.28–0.71 aralığında olup bu sınıfta belirgin bir kararsızlık söz konusudur. Sivil yolcu uçakları 0.35–0.68, askeri kargo uçakları ise 0.45–0.91 aralığında etiketlenmiştir. Apron gibi yoğun sahnelerde kutuların çakıştığı ve bazı nesnelerin eksik ya da düşük güvenle tespit edildiği dikkat çekmektedir. Bu durum, modelin varsayılan hiperparametrelerle eğitildiğinde sınıflar arasındaki ayrımı netleştirmekte yetersiz kaldığını göstermektedir.

4.1.2. YOLOv5 ile Veri Artırımı Uygulanan Eğitim

Bu deneyde YOLOv5 mimarisinde varsayılan hiperparametrelerle yapılan eğitim sürecine ek olarak çeşitli veri artırımı teknikleri uygulanmıştır. Eğitim sürecinde mekânsal (spatial) ve fotometrik (photometric) dönüşümlerden yararlanılmış, bu sayede veri çeşitliliği artırılarak modelin farklı koşullar altında daha güvenilir ve genelleme yeteneği yüksek çıktılar üretmesi hedeflenmiştir. Eğitim süreci 5 katlı çapraz doğrulama yöntemiyle yürütülmüş; her iterasyonda elde edilen en iyi model ağırlıkları sabit test kümesi üzerinde değerlendirilmiştir. Beş iterasyondan elde edilen test sonuçlarının ortalaması alınarak modelin genel performansını gösteren değerler Çizelge 4.3’te verilmiştir.

Çizelge 4.3. Deney 2 için ortalama metrik sonuçları

Metrik	Ortalama Değer
mAP@0.5	0.850
mAP@0.5:0.95	0.553
Kesinlik	0.804
Duyarlılık	0.806
F1-skor	0.804

Çizelge 4.3 incelendiğinde, mAP@0.5 değerinin 0.850 seviyesine yükseldiği görülmektedir. Bu sonuç, Deney 1'e kıyasla belirgin bir artış sağlandığını göstermektedir. mAP@0.5:0.95 değerinin 0.553'e ulaşması, modelin farklı IoU eşiklerinde daha istikrarlı tahminler yapabildiğini ortaya koymaktadır. Kesinlik (0.804) ve duyarlılık (0.806) değerlerinin birbirine çok yakın olması, modelin yanlış pozitif ve yanlış negatif oranları arasında dengeli bir performans gösterdiğini ifade etmektedir. Ayrıca F1-skorunun 0.804'e yükselmesi, genel başarı düzeyinde dikkate değer bir iyileşmeye işaret etmektedir. Bu bulgular, veri artırımı yöntemlerinin modelin genelleme yeteneğini güçlendirdiğini ve sınırlı veri koşullarında performans kayıplarını önemli ölçüde azalttığını göstermektedir.

Çizelge 4.4, askeri kargo uçağı, sivil yolcu uçağı ve askeri savaş uçağı olmak üzere üç farklı sınıf için gerçekleştirilen beş katlı çapraz doğrulama sonucunda elde edilen ortalama mAP, kesinlik, duyarlılık ve F1-skor metriklerini göstermektedir.

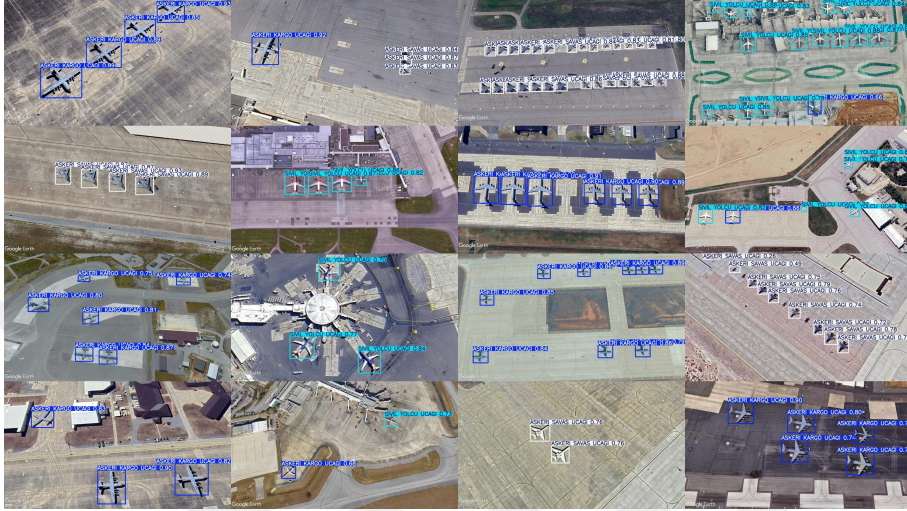
Çizelge 4.4. Deney 2 için sınıf bazında ortalama metrik sonuçları

Metrik	Askeri Kargo Uçağı	Sivil Yolcu Uçağı	Askeri Savaş Uçağı
mAP@0.5	0.920	0.862	0.768
mAP@0.5:0.95	0.657	0.533	0.470
Kesinlik	0.816	0.825	0.772
Duyarlılık	0.891	0.835	0.674
F1-skor	0.844	0.825	0.702

Çizelge 4.4 incelendiğinde, en yüksek başarı yine *Askeri Kargo Uçağı* sınıfında elde edilmiştir. Bu sınıfta mAP@0.5 değeri 0.920'ye, duyarlılık ise 0.891'e ulaşarak modelin bu sınıfı yüksek doğrulukla tespit edebildiğini göstermektedir. *Sivil Yolcu Uçağı* sınıfında da dengeli sonuçlar elde edilmiş; mAP@0.5 değeri 0.862, kesinlik ve duyarlılık değerleri ise birbirine yakın (0.825 ve 0.835) bulunmuştur. *Askeri Savaş Uçağı* sınıfında ise performans önceki deneye göre iyileşmiş, mAP@0.5 değeri 0.768'e yükselmiş ve F1-skoru 0.702 olmuştur. Bununla birlikte duyarlılık değerinin 0.674 seviyesinde kalması, bu sınıfın bazı örneklerinde modelin hâlâ yetersiz kaldığını göstermektedir. Genel olarak, veri artırımı tüm sınıflarda performansı yükseltmiş, özellikle kargo ve yolcu uçaklarında

yüksek doğruluk düzeyleri sağlanırken, savaş uçağı sınıfında da kayda değer bir gelişme ortaya çıkmıştır.

Elde edilen model çıktılarının görselleştirilmiş sonuçlarını göstermektedir. Görselde farklı havaalanı sahnelerinden alınan örneklerde üç sınıfa ait tespitler ile modelin tahmin ettiği etiketler ve güven skorları Şekil 4.2’de gösterilmektedir.



Şekil 4.2. Deney 2 için nesne tespit çıktılarına ait örnek görseller

Şekil 4.2 incelendiğinde modelin önceki aşamaya göre daha yüksek güven skorlarıyla tahminler yaptığı görülmektedir. Askeri kargo uçakları 0.74–0.92 aralığında tutarlı biçimde tespit edilmiş, sivil yolcu uçakları da 0.66–0.86 aralığında daha dengeli sonuçlar vermiştir. Askeri savaş uçaklarında güven skorları 0.49–0.89 arasında değişmekte olup bu sınıfta belirsizlik kısmen sürmektedir. Apron gibi yoğun sahnelerde yanlış veya çakışan kutuların azaldığı, ancak tamamen ortadan kalkmadığı gözlenmiştir. Bu bulgu, veri artırımı yöntemlerinin özellikle yoğun sahnelerdeki belirsizlikleri kısmen azalttığını ortaya koymaktadır.

4.1.3. YOLOv5 ile Veri Artırımı ve Optimizasyon Uygulanan Eğitim

Bu deneyde, veri artırımı tekniklerinin yanı sıra genetik algoritma (GA) tabanlı hiperparametre optimizasyonu uygulanmıştır. GA; öğrenme oranı, momentum ve ağırlık çarpanı gibi kritik hiperparametreleri evrimsel bir arama süreciyle optimize ederek modelin hem eğitim sürecinde daha istikrarlı öğrenmesini hem de tahmin doğruluğunun yükseltilmesini sağlamıştır. Eğitim süreci 5 katlı çapraz doğrulama yöntemiyle yürütülmüş; her fold’da elde edilen en iyi model ağırlıkları sabit test kümesi üzerinde değerlendirilmiştir. Beş fold’dan elde edilen test sonuçlarının ortalaması alınarak modelin genel performansını gösteren değerler Çizelge 4.5’te verilmiştir.

Çizelge 4.5. Deney 3 için ortalama metrik sonuçları

Metrik	Ortalama Değer
mAP@0.5	0.899
mAP@0.5:0.95	0.602
Kesinlik	0.853
Duyarlılık	0.836
F1-skor	0.844

Çizelge 4.5 incelendiğinde, mAP@0.5 değerinin 0.899'a ulaştığı görülmektedir. Bu sonuç, önceki iki deneye kıyasla modelin doğruluk düzeyinde kayda değer bir artış sağlandığını göstermektedir. mAP@0.5:0.95 değerinin 0.602'ye yükselmesi, modelin farklı IoU eşiklerinde daha kararlı tahminler üretebildiğini ortaya koymaktadır. Kesinlik (0.853) ve duyarlılık (0.836) değerleri arasındaki denge, modelin yanlış pozitif ve yanlış negatif tahminleri başarıyla minimize ettiğini göstermektedir. F1-skorunun 0.844'e ulaşması ise genel performansın en yüksek seviyeye taşındığını ortaya koymaktadır. Bu bulgular, veri artırımı ve genetik algoritma tabanlı hiperparametre optimizasyonunun birlikte uygulanmasının, modelin sınırlı veri koşullarında dahi daha istikrarlı ve güvenilir sonuçlar üretmesini sağladığını göstermektedir.

Çizelge 4.6, askeri kargo uçağı, sivil yolcu uçağı ve askeri savaş uçağı olmak üzere üç farklı sınıf için gerçekleştirilen beş katlı çapraz doğrulama sonucunda elde edilen ortalama mAP, kesinlik, duyarlılık ve F1-skor metriklerini göstermektedir.

Çizelge 4.6. Deney 3 için sınıf bazında ortalama metrik sonuçları

Metrik	Askeri Kargo Uçağı	Sivil Yolcu Uçağı	Askeri Savaş Uçağı
mAP@0.5	0.931	0.865	0.798
mAP@0.5:0.95	0.691	0.561	0.471
Kesinlik	0.804	0.786	0.813
Duyarlılık	0.938	0.841	0.717
F1-skor	0.860	0.808	0.742

Çizelge 4.6 incelendiğinde, en yüksek başarı *Askeri Kargo Uçağı* sınıfında elde edilmiştir. Bu sınıfta mAP@0.5 değeri 0.931 ve duyarlılık 0.938 seviyesine ulaşarak modelin bu sınıfı oldukça yüksek doğrulukla tespit edebildiğini göstermektedir. *Sivil Yolcu Uçağı* sınıfında da güçlü sonuçlar elde edilmiş, mAP@0.5 değeri 0.865, F1-skoru ise 0.808 olarak hesaplanmıştır. *Askeri Savaş Uçağı* sınıfı ise en düşük değerlere sahip olmasına rağmen önceki deneylere kıyasla gelişme göstermiş, mAP@0.5 değeri 0.798'e, F1-skoru ise 0.742'ye yükselmiştir. Bu bulgular, veri artırımı ve genetik algoritma tabanlı hiperparametre optimizasyonunun özellikle kargo ve yolcu uçaklarında yüksek doğruluk

düzeylerine ulaşmayı sağladığını, savaş uçaklarında ise modelin daha dengeli ve güvenilir sonuçlar üretebilmesine katkı sunduğunu ortaya koymaktadır.

Elde edilen model çıktılarının görselleştirilmiş sonuçlarını göstermektedir. Görselde farklı havaalanı sahnelerinden alınan örneklerde üç sınıfa ait tespitler ile modelin tahmin ettiği etiketler ve güven skorları Şekil 4.3'te gösterilmektedir.



Şekil 4.3. Deney 3 için nesne tespit çıktılarına ait örnek görseller

Şekil 4.3 incelendiğinde modelin tahmin güveninde önemli bir artış sağlandığı görülmektedir. Askeri kargo uçakları 0.76–0.93, sivil yolcu uçakları 0.64–0.91, askeri savaş uçakları ise 0.74–0.92 aralığında yüksek skorlarla etiketlenmiştir. Özellikle Apron ve terminal bölgelerindeki yoğun sahnelerde tüm uçakların doğru şekilde ve yüksek güvenle tespit edildiği, çakışan kutuların ya da eksik tahminlerin bulunmadığı dikkat çekmektedir. Bu sonuç, genetik algoritma ile optimize edilen parametrelerin veri artırımıyla birlikte kullanıldığında modelin en dengeli ve güvenilir performansa ulaştığını göstermektedir.

4.2. Deney Sonuçlarının Karşılaştırılması

Gerçekleştirilen üç farklı deneysel senaryonun sonuçları karşılaştırmalı olarak sunulmuştur. Karşılaştırmanın amacı, farklı eğitim stratejilerinin model başarımına olan etkisini nesnel ve sayısal veriler üzerinden ortaya koymaktır. Bu kapsamda, beş katlı çapraz doğrulama ile elde edilen ortalama $mAP@0.5$, $mAP@0.5:0.95$, *kesinlik*, *duyarlılık* ve *F1-skor* değerleri değerlendirilmiştir. Elde edilen sonuçlar Çizelge 4.7’de verilmiştir.

Çizelge 4.7. Genel ortalama metrik değerleri

Deney	mAP@0.5	mAP@0.5:0.95	Kesinlik	Duyarlılık	F1-skor
Deney 1	0.755	0.453	0.716	0.705	0.706
Deney 2	0.850	0.553	0.804	0.806	0.804
Deney 3	0.899	0.602	0.853	0.836	0.844

Çizelge 4.7 incelendiğinde, deneyler arasında tüm metriklerde kademeli ve tutarlı artışların gerçekleştiği görülmektedir. Deney 1, modelin başlangıç seviyesini temsil etmekte olup ortalama $mAP@0.5$ 0.755, $mAP@0.5:0.95$ 0.453, *kesinlik* 0.716, *duyarlılık* 0.705 ve *F1-skor* 0.706 değerleri elde edilmiştir. Bu sonuçlar, modelin temel performansının sınırlı ancak dengeli olduğunu göstermektedir. Deney 2’de uygulanan veri artırımı sayesinde tüm metriklerde belirgin iyileşmeler kaydedilmiş; özellikle *kesinlik* (0.804) ve *duyarlılık* (0.806) değerlerinin birbirine yakın olması, modelin hem doğru tespit yapma hem de olası nesneleri yakalama konusunda dengeli bir başarıya ulaştığını ortaya koymaktadır. Deney 3’te ise genetik algoritma ile optimize edilen hiperparametreler kullanılmış ve en yüksek performans değerlerine ulaşılmıştır. Ortalama $mAP@0.5$ 0.899, $mAP@0.5:0.95$ 0.602, *kesinlik* 0.853, *duyarlılık* 0.836 ve *F1-skor* 0.844 olarak hesaplanmıştır. Bu sonuçlar, hiperparametre optimizasyonunun modele ek katkı sağladığını ve özellikle kesinlik ile duyarlılık arasındaki dengeyi koruyarak genel başarıyı artırdığını göstermektedir.

4.2.1. Sınıf Bazında Metrik Sonuçlarının Karşılaştırılması

Farklı deneysel çalışmaların sınıf bazındaki performans değerleri ayrıntılı olarak ele alınmıştır. Çalışmada, üç temel sınıf olan askeri kargo uçağı, sivil yolcu uçağı ve askeri savaş uçağı ayrı ayrı incelenmiştir. Her bir sınıf için elde edilen sonuçların tek tek değerlendirilmesi, modelin her sınıfta ulaştığı başarı düzeyini daha net şekilde ortaya koymak açısından önem taşımaktadır. Böylece modelin yalnızca genel performansı değil, aynı zamanda sınıflar arası farklılıkları da somut verilerle gözlemlenebilmiştir. Bu incelemede, varsayılan hiperparametrelerle eğitilen modelin temel sonuçları başlangıç noktası olarak ele alınmış, daha sonra veri artırımı tekniklerinin katkıları ve genetik algoritma tabanlı hiperparametre optimizasyonunun etkileri sınıf düzeyinde ayrı ayrı değerlendirilmiştir. Böylelikle, uygulanan her yöntemin farklı sınıflardaki performansa olan etkisi açıkça karşılaştırılabilir hale getirilmiştir. Elde edilen sonuçlar Çizelge 4.8’de verilmiştir.

Çizelge 4.8. Sınıf bazında ortalama metrik değerleri

Sınıf	Deney	mAP@0.5	mAP@0.5:0.95	Kesinlik	Duyarlılık	F1-skor
Askeri	Deney 1	0.856	0.576	0.746	0.834	0.774
Kargo	Deney 2	0.920	0.657	0.816	0.891	0.844
Uçağı	Deney 3	0.931	0.691	0.804	0.938	0.860
Sivil	Deney 1	0.764	0.439	0.698	0.728	0.699
Yolcu	Deney 2	0.862	0.533	0.825	0.835	0.825
Uçağı	Deney 3	0.865	0.561	0.786	0.841	0.808
Askeri	Deney 1	0.646	0.343	0.689	0.554	0.569
Savaş	Deney 2	0.768	0.470	0.772	0.674	0.702
Uçağı	Deney 3	0.798	0.471	0.813	0.717	0.742

Askeri Kargo Uçağı sınıfında, ilk deneyden ikinci deneye geçişte tüm metriklerde dikkate değer bir iyileşme gerçekleşmiştir. $mAP@0.5$ değeri 0.856'dan 0.920'ye, $mAP@0.5:0.95$ değeri ise 0.576'dan 0.657'ye yükselmiştir. Üçüncü deneyde bu artış eğilimi sürmüştür, $mAP@0.5$ değeri 0.931'e, F1-skor 0.860'a ulaşmıştır. Özellikle duyarlılık değerinin 0.938'e çıkması, modelin bu sınıfın örneklerini yakalama konusundaki başarısını ortaya koymaktadır.

Sivil Yolcu Uçağı sınıfında, başlangıç senaryosunda elde edilen metrikler görece düşük seviyelerde kalmış, ikinci deneyde ise belirgin iyileşmeler gözlenmiştir. Bu deneyde $mAP@0.5$ değeri 0.764'ten 0.862'ye, F1-skor 0.699'dan 0.825'e yükselmiştir. Üçüncü deneyde $mAP@0.5$ ve $mAP@0.5:0.95$ değerlerindeki artış devam etmiş, duyarlılık metriğinde de yükseliş kaydedilmiştir. Elde edilen sonuçlar, uygulanan yöntemlerin bu sınıfta genel performansa olumlu katkı sağladığını göstermektedir.

Askeri Savaş Uçağı sınıfında, ilk deneyde düşük seviyelerde başlayan performans ikinci deneyle birlikte belirgin biçimde artmıştır. $mAP@0.5$ değeri 0.646'dan 0.768'e, F1-skor ise 0.569'dan 0.702'ye çıkmıştır. Üçüncü deneyde bu gelişim eğilimi korunmuş, $mAP@0.5$ değeri 0.798'e, F1-skor 0.742'ye yükselmiştir. Böylece bu sınıfta da deneyler ilerledikçe modelin başarısının arttığı görülmektedir.

Genel olarak bakıldığında, üç sınıfta da deneylerin ilerlemesiyle birlikte metriklerde düzenli artışlar elde edilmiştir. Özellikle ikinci deneyle birlikte başlayan iyileşmeler, üçüncü deneyde korunmuş ve bazı metriklerde daha ileri seviyelere ulaşılmıştır. Bu sonuçlar, uygulanan yöntemlerin farklı sınıflarda modelin performansını geliştirmede etkili olduğunu ortaya koymaktadır.

5. TARTIŞMA

5.1. Varsayılan Parametrelerin Sınırlılıkları

Varsayılan hiperparametrelerle gerçekleştirilen model eğitimi temel bir referans noktası sunmakla birlikte sınırlı bir performans düzeyi ortaya koymuştur. Özellikle karmaşık arka planların bulunduğu sahnelerde ve küçük boyutlu nesnelerin tespitinde modelin başarısı düşmektedir. Bu durum, varsayılan parametrelerin veri setinin özgün yapısına yeterince uyum sağlayamadığını göstermektedir. Nesne yoğunluğu, sınıf dağılımındaki dengesizlikler ve görüntü çözünürlüğü gibi faktörlere uyum sağlayamayan sabit hiperparametreler modelin genelleme kabiliyetini kısıtlamaktadır.

Ayrıca varsayılan parametrelerle yapılan eğitimde modelin belirli sınıflarda diğerlerine kıyasla daha düşük doğruluk sergilediği gözlenmiştir. Bu da her sınıfın farklı görsel özelliklere sahip olması nedeniyle tek tip parametre ayarlarının tüm sınıflar için aynı derecede etkili olmadığını ortaya koymaktadır. Dolayısıyla bu senaryo ilerleyen aşamalarda veri artırımı ve hiperparametre optimizasyonu gibi yöntemlerin kullanılmasını gerekli kılan bir başlangıç noktası oluşturmıştır.

5.2. Veri Artırımı ile Sağlanan Katkılar

Veri artırımı tekniklerinin uygulanması modelin farklı senaryolara uyum sağlama yeteneğini belirgin biçimde geliştirmiştir. Özellikle mekânsal ve fotometrik dönüşümler sayesinde eğitim verisinin çeşitliliği artmış, model daha önce karşılaşmadığı varyasyonlara karşı dayanıklılık kazanmıştır. Bu durum küçük nesnelerin tespitinde ve karmaşık arka planlarda yapılan sınıflandırmalarda doğruluk oranlarının yükselmesine katkı sağlamıştır.

Ayrıca sınıf dağılımında gözlenen dengesizliklerin kısmen dengelenmesi modelin sınıflar arasında daha tutarlı sonuçlar üretmesine imkân tanımıştır. Veri artırımı yalnızca genel başarı metriklerinde iyileşme sağlamakla kalmamış, aynı zamanda modelin genelleme kabiliyetini de güçlendirmiştir. Böylece varsayılan parametrelerle elde edilen sınırlı performansın ötesine geçilerek daha dengeli ve güvenilir bir performans düzeyi elde edilmiştir.

5.3. Genetik Algoritmanın Model Performansına Katkısı

Genetik algoritma ile gerçekleştirilen hiperparametre optimizasyonu modelin öğrenme sürecini daha dengeli hâle getirerek genel performansta belirgin bir iyileşme sağlamıştır. Evrimsel arama mekanizması sayesinde öğrenme oranı, momentum ve kayıp fonksiyonu ağırlıkları gibi kritik hiperparametreler veri setinin özgün yapısına uyumlu biçimde yeniden düzenlenmiştir. Bu süreç modelin hem daha hızlı yakınsamasına hem de daha kararlı bir hata eğrisi sergilemesine katkıda bulunmuştur.

Ayrıca farklı sınıflarda gözlenen dengesizliklerin azaltılmasında ve küçük nesnelerin tespitinde elde edilen gelişmeler GA'nın yalnızca tekil metriklerde değil, modelin genel genelleme kabiliyetinde de olumlu etkiler yarattığını göstermektedir. Veri artırımı ile sağlanan çeşitliliğin üzerine eklenen bu optimizasyon adımı modelin performansını daha üst düzeye taşıyarak en dengeli ve güvenilir sonuçların elde edilmesine olanak tanımıştır.

5.4. Sonuçların Literatür ile Karşılaştırılması

Literatürde uzaktan algılama görüntülerinde nesne tespiti amacıyla YOLOv5 mimarisi üzerinde çeşitli iyileştirmeler yapılmıştır. SEB-YOLO yönteminde modelin küçük nesnelerden daha fazla bilgi çıkarabilmesi ve benzer nesnelerin ayırt edilebilmesi için bazı yapısal değişiklikler uygulanmıştır. Backbone bölümüne SPD-Conv katmanı eklenmiş, son katmanda dikkat mekanizması içeren bir havuzlama modülü kullanılmış, boyutlar arası bilgi aktarımı için Bi-FPN yapısı tercih edilmiş ve sınıflandırma ile regresyon görevlerini ayırmak amacıyla decoupled head eklenmiştir. Yapılan testler sonucunda NWPU VHR-10 veri setinde %93.5, RSOD veri setinde ise %93.9 mAP değerine ulaşılmış; bu sonuçlar orijinal YOLOv5l modeline göre sırasıyla %4.0 ve %5.3 oranında artışa karşılık gelmiştir (Hui vd. 2024). Bu çalışma geniş veri setlerinde mimari değişikliklerin model doğruluğuna katkı sağlayabileceğini göstermektedir.

KCFS-YOLOv5 yönteminde de benzer şekilde model üzerinde yapısal iyileştirmeler yapılmıştır. Anchor kutularının belirlenmesi için K-means++ algoritması kullanılmış, backbone bölümüne Coordinate Attention modülü eklenmiş, boyutlar arası bilgi kaybını azaltmak için Bi-FPN yapısı entegre edilmiş ve küçük nesnelerin tespiti için tiny object detection head tasarlanmıştır. Ayrıca tahmin kutuları ile gerçek kutular arasındaki uyumu artırmak için SIoU loss fonksiyonu tercih edilmiştir. Bu düzenlemeler NWPU VHR-10, RSOD ve UCAS-AOD-CAR veri setleri üzerinde test edilmiş ve özellikle küçük nesnelerin tespitinde orijinal YOLOv5'e kıyasla daha yüksek doğruluk değerleri elde edilmiştir (Tian vd. 2023).

Uydu görüntülerinden askeri uçak tiplerinin sınıflandırılmasına yönelik bir çalışmada farklı CNN tabanlı mimariler (VGG16, ResNet50, MobileNetV2, DenseNet121, InceptionV3 ve Xception) karşılaştırılmış ve epoch 8 sonrasında en yüksek başarı Xception modeliyle elde edilmiştir. Bu model *doğruluk*, *kesinlik*, *duyarlılık* ve *F1-skor* değerlerinde %99 seviyelerine ulaşırken InceptionV3 yüksek performans sergilemiştir. MobileNetV2 ve ResNet50 orta düzeyde sonuçlar üretmiş, VGG16 ve DenseNet121 ise görece daha düşük doğruluk sağlamıştır. Çalışmada YOLOv5 kullanılmamış, hazır ve geniş kapsamlı veri setleri üzerinden sadece askeri uçak tipleri için mimari farklılıkların sınıflandırma başarısına etkisi incelenmiştir (Özçelik 2024).

Bu tez çalışmasında ise farklı bir veri yapısı üzerinde inceleme yapılmıştır. Literatürde sıkça kullanılan geniş ve hazır veri setleri yerine Google Earth Pro aracılığıyla oluşturulan ve sınırlı sayıda örnek içeren özel bir uydu görüntüsü veri seti kullanılmıştır. Veri sayısının az olması nedeniyle modelin genelleme kabiliyeti sınırlı kalabileceğinden eğitim sürecinde mekânsal ve fotometrik veri artırımı yöntemleri uygulanmıştır. Böylece modelin farklı koşullarda daha dayanıklı olması amaçlanmıştır. Ayrıca hiperparametreler üzerinde durulmuş ve genetik algoritma tabanlı bir optimizasyon yöntemi kullanılmıştır. Varsayılan hiperparametrelerin küçük ölçekli veri setlerinde yeterli sonuç vermemesi ihtimali dikkate alınarak genetik algoritma ile farklı parametre kombinasyonları denenmiş ve bu yolla daha uygun ayarların elde edilmesi hedeflenmiştir.

Sonuç olarak bu çalışmada sınırlı örnekten oluşan bir uydu görüntüsü veri setinde hem veri artırımı hem de hiperparametre optimizasyonunun model performansına etkileri değerlendirilmiştir. Elde edilen bulgular küçük ölçekli veri setleri üzerinde uygulanabilecek yöntemlerin anlaşılmasına katkı sağlamaktadır. Literatürdeki SEB-YOLO ve KCFS-YOLOv5 gibi çalışmalar daha çok geniş veri setlerinde mimari düzenlemeler üzerinden değerlendirilmişken, bu tez çalışması sınırlı veri koşullarında hiperparametre optimizasyonunun sonuçlarını incelemiştir. Böylece farklı veri ölçeklerinde hangi yöntemlerin öne çıkabileceği konusunda karşılaştırma yapma imkânı sunmuştur.

5.5. Uygulama Sürecinde Karşılaşılan Sınırlılıklar

Model eğitimi ve hiperparametre optimizasyonu süreci tek bir GPU üzerinde (RTX 4050) yürütülmüştür. Bu donanım modelin eğitilmesi ve genetik algoritmanın uygulanması için yeterli olmakla birlikte, daha geniş popülasyon büyüklükleri veya daha uzun nesil sayılarının denenmesini sınırlamıştır. Genetik algoritma kapsamında popülasyon büyüklüğü 16–18, nesil sayısı ise en fazla 60 olarak belirlenmiş; bazı durumlarda en iyi sonuçlara bu sınırdan daha erken ulaşılsa da daha kapsamlı bir arama süreci zaman ve işlem maliyeti nedeniyle gerçekleştirilememiştir. Ayrıca çalışmada yalnızca YOLOv5 mimarisi ele alınmış, farklı mimari yapılar ile karşılaştırma yapılmamıştır. Bu nedenle elde edilen bulgular YOLOv5 mimarisi ve kullanılan özel veri seti ile sınırlı olup, farklı yöntem veya koşullarda elde edilebilecek performanslara doğrudan genellenemez.

5.6. Sonuçların Genellenebilirliği Açısından Değerlendirme

Bu çalışmada elde edilen sonuçlar yalnızca oluşturulan özel veri seti ve YOLOv5 mimarisi temelinde değerlendirilmiştir. Görüntülerin sınırlı sayıda olması ve yalnızca üç sınıfa odaklanması sonuçların farklı veri tipleri veya daha geniş sınıf yapıları için doğrudan genellenmesini sınırlandırmaktadır. Ayrıca verilerin tek bir kaynaktan (Google Earth Pro) manuel olarak toplanmış olması, farklı bölgelerden veya farklı görüntüleme koşulla-

rından elde edilecek verilerde benzer performansın sağlanıp sağlanamayacağı konusunda belirsizlik oluşturmaktadır.

Beş katlı çapraz doğrulama ile modelin mevcut veri seti üzerindeki genelleme yeteneği değerlendirilmiştir. Bu yöntem verilerin farklı bölünmelerle hem eğitim hem de doğrulama sürecinde kullanılması sayesinde daha dengeli bir ölçüm imkânı sunmuştur. Ancak elde edilen sonuçlar yalnızca bu veri seti için geçerlidir. Farklı bölgelerden sağlanacak uydu görüntüleri veya alternatif mimarilerle yapılacak çalışmalar bulguların daha geniş koşullarda doğrulanmasına katkı sağlayacaktır.

Genetik algoritma tabanlı hiperparametre optimizasyonu ise prensip olarak farklı veri kümelerine uyarlanabilir bir yöntemdir. Ancak her veri seti kendi özelliklerine sahip olduğundan GA ile elde edilen hiperparametrelerin başka veri kümelerine doğrudan aktarılması mümkün değildir. Bu nedenle farklı senaryolarda benzer bir iyileştirme sağlamak için optimizasyon sürecinin yeniden yürütülmesi gerekecektir.

6. SONUÇLAR

Bu tez çalışmasında, uydu görüntülerinde nesne tespiti amacıyla derin öğrenme tabanlı bir yaklaşım uygulanmış ve YOLOv5 mimarisi temel alınmıştır. Çalışmanın temel hedefi, sınırlı veri koşullarında modelin başarımını artırmak ve bu koşulların neden olduğu genelleme sorunlarını en aza indirmektir. Bu doğrultuda iki yöntem üzerinde durulmuştur. Bunlardan ilki, eğitim verilerinin çeşitliliğini artırmaya yönelik veri artırımı teknikleridir. İkincisi ise öğrenme sürecini daha sistemli bir şekilde yönlendirmeyi amaçlayan genetik algoritma tabanlı hiperparametre optimizasyonudur. Böylece hem verilerin zenginleştirilmesinin hem de parametrelerin düzenli biçimde ayarlanmasının model performansına katkıları birlikte ele alınmıştır.

Elde edilen bulgular, veri artırımı sayesinde modelin daha geniş bir örnek yelpazesinden öğrenme fırsatı bulduğunu göstermiştir. Bu durum, özellikle sınırlı veri setlerinde kritik öneme sahip olan genelleme yeteneğini güçlendirmiştir. Sonuçlarda *kesinlik* ve *duyarlılık* değerlerinde görülen artış, çeşitliliği artırılmış verilerin modelin hem doğru sınıflandırma yapma becerisini hem de daha fazla nesneyi yakalama kapasitesini geliştirdiğini ortaya koymuştur. Böylece veri artırımı, eğitim sürecine yalnızca nicelik açısından değil, içerik açısından da anlamlı bir katkı sağlamıştır.

Genetik algoritma tabanlı hiperparametre optimizasyonu ise modelin başarımını farklı bir yönden geliştirmiştir. Parametrelerin sistematik biçimde ayarlanması, yalnızca rastgele denemelerle ulaşılamayacak uygun değerlerin bulunmasını mümkün kılmıştır. Bu sayede $mAP@0.5$, $mAP@0.5:0.95$, *duyarlılık* ve *F1-skor* gibi temel metriklerde belirgin ilerlemeler elde edilmiştir. Bununla birlikte, bazı sınıflarda *kesinlik* değerlerinde küçük düşüşler görülmüş; bu da hiperparametre optimizasyonunun sınıf bazında farklı etkiler doğurabileceğini göstermiştir. Ancak genel eğilim, optimizasyonun model başarımını güçlendirdiği yönündedir.

İki yöntemin birlikte kullanılması, tek başına sağlanan kazanımları daha da ileriye taşımış ve modelin farklı sınıflarda daha dengeli sonuçlar üretmesine katkıda bulunmuştur. Veri artırımı modelin çeşitlilikle öğrenmesini sağlarken, genetik algoritma bu öğrenme sürecini daha istikrarlı ve verimli hale getirmiştir. Böylece model yalnızca doğruluk düzeyinde yükseliş göstermemiş, aynı zamanda farklı veri senaryolarında da daha tutarlı bir performans sergilemiştir.

Sonuç olarak, bu çalışma sınırlı veri koşullarında nesne tespiti için veri artırımı ve hiperparametre optimizasyonunun birlikte kullanılmasının etkili bir strateji olduğunu ortaya koymuştur. Özellikle genetik algoritma gibi evrimsel bir yaklaşımın hiperparametre arama sürecini hızlandığı ve parametrelerin daha kısa sürede uygun değerlere ulaşmasını sağladığı görülmüştür. Elde edilen yaklaşımın benzer şekilde sınırlı veri ya-

pısına sahip farklı senaryolarda da uygulanabilir olduğu düşünülmektedir. Ayrıca, askeri ve sivil havacılıkta hava araçlarının otomatik olarak tespitinde; gözetleme, güvenlik, hava sahasının takibi ve acil durumlarda karar desteği gibi alanlarda kullanılma potansiyeline sahip olduğu değerlendirilmektedir.



7. KAYNAKLAR

- Anonim. 2024. Octoverse: AI leads Python to top language as the number of global developers surges. GitHub. <https://github.blog/news-insights/octoverse/octoverse-2024/> [Son erişim tarihi: 27.05.2025].
- Bischl, B., Binder, M., Lang, M., Pfisterer, F. and Thomas, J. 2023. Hyperparameter optimization: Foundations, algorithms, best practices and open challenges. *WIREs Data Mining and Knowledge Discovery*, 13(2): e1487.
- Bochkovskiy, A., Wang, C.Y. and Liao, H.Y.M. 2020. YOLOv4: Optimal speed and accuracy of object detection. arXiv preprint, arXiv:2004.10934.
- Buslaev, A., Iglovikov, V.I., Khvedchenya, E., Parinov, A., Druzhinin, M. and Kalinin, A.A. 2020. Albumentations: Fast and flexible image augmentations. *Information*, 11(2): 125.
- Chen, J., Sun, J., Li, Y. and Hou, C. 2022. Object detection in remote sensing images based on deep transfer learning. *Multimedia Tools and Applications*, 81: 12093–12109.
- Chen, X., Liu, Z. and Yang, J. 2024. YOLOv5-based aerial object detection with super resolution. arXiv preprint, arXiv:2401.14661.
- Cheng, G., Zhou, P. and Han, J. 2016. Learning rotation-invariant convolutional neural networks for object detection in VHR optical remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 54(12): 7405–7415.
- El Ghazouali, S., Gucciardi, A., Venturini, F., Venturi, N., Rügsegger, M. and Micellucci, U. 2024. FlightScope: An Experimental Comparative Review of Aircraft Detection Algorithms in Satellite Imagery. <https://arxiv.org/abs/2404.02877> [Erişim tarihi: 3 Ağustos 2025].
- Feurer, M. and Hutter, F. 2019. Hyperparameter optimization. In: Hutter, F., Kotthoff, L. and Vanschoren, J. (Ed.), *Automated Machine Learning: Methods, Systems, Challenges*. Springer, Cham, ss. 3–33.
- Fredriksson, T. and Zhang, Y., 2025. An empirical evaluation of deep semi-supervised learning. *International Journal of Data Science and Analytics*, 15, 233–251. <https://doi.org/10.1007/s41060-024-00713-8>
- GluonCV. 2024. GluonCV Toolkit for Computer Vision on MXNet. GitHub. <https://github.com/dmlc/gluon-cv> [Son erişim tarihi: 27.05.2025].

- Goodfellow, I., Bengio, Y. and Courville, A. 2016. Deep Learning. MIT Press, Cambridge, MA, 775 s.
- Grosgeorge, D., Arbelot, M., Goupilleau, A., Ceillier, T. and Allieux, R. 2020. Concurrent segmentation and object detection CNNs for aircraft detection and identification in satellite images. IEEE International Geoscience and Remote Sensing Symposium (IGARSS), 4378–4381.
- Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J. and Chen, T. 2018. Recent advances in convolutional neural networks. Pattern Recognition, 77: 354–377.
- Harris, C.R., Millman, K.J., van der Walt, S.J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N.J. vd. 2020. Array programming with NumPy. Nature, 585(7825): 357–362.
- He, T., Zhang, Z., Zhang, H., Zhang, Z., Xie, J., Li, M., Wang, C. and Shen, C. 2019. Bag of tricks for image classification with convolutional neural networks. CVPR, ss. 558–567.
- Holland, J.H. 1975. Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor.
- Hu, B., Lei, C., Wang, D., Zhang, S. ve Chen, Z. 2019. A Preliminary Study on Data Augmentation of Deep Learning for Image Classification. arXiv preprint, arXiv:1906.11887. <https://arxiv.org/abs/1906.11887>
- Hui, Y., You, S., Hu, X., Yang, P. ve Zhao, J. 2024. SEB-YOLO: an improved YOLOv5 model for remote sensing small target detection. Sensors, 24(7), 2193, İsviçre.
- Hunter, J.D. 2007. Matplotlib: A 2D graphics environment. Computing in Science & Engineering, 9(3): 90–95.
- Ibrahim, H., Abdo, A., El Kerdawy, A.M. and Sharaf Eldin, A. 2021. Signal detection in pharmacovigilance: A review of informatics-driven approaches. Artificial Intelligence in the Life Sciences, 1: 100005.
- Jiang, T., Frøseth, G. T., and Rønnquist, A. 2023. A YOLOv5-based vision system for autonomous bolt-tightening operations using a collaborative robot. Computers & Structures, 267: 106013.
- Jocher, G., Chaurasia, A., Stoken, A., Borovec, J. vd. 2023. YOLO by Ultralytics. arXiv preprint, arXiv:2304.00501.

- Laban, N., Abdellatif, B., Ebeid, H.M., Shedeed, H.A. and Tolba, M.F. 2019. Convolutional neural network with dilated anchors for object detection in very high resolution satellite images. 14th International Conference on Computer Engineering and Systems (ICCES), ss. 34–39. IEEE.
- Liu, Q. and Wu, Y. 2012. Supervised learning. In: Seel, N.M. (Ed.), Encyclopedia of the Sciences of Learning. Springer, Boston, MA, ss. 3243–3245.
- Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., and Berg, A.C., 2016. SSD: Single shot multibox detector. European Conference on Computer Vision (ECCV), Springer, ss. 21–37.
- Maity, M., Banerjee, S. and Chaudhuri, S.S. 2021. Faster R-CNN and YOLO based vehicle detection: A survey. 5th International Conference on Computing Methodologies and Communication (ICCMC), ss. 1442–1447. IEEE.
- Mansour, A., Hassan, A., Hussein, W.M. and Said, E. 2019. Automated vehicle detection in satellite images using deep learning. IOP Conference Series: Materials Science and Engineering, 610(1): 012027.
- Mao, Q.C., Sun, H.M., Liu, Y.B. and Jia, R.S. 2019. Mini-YOLOv3: Real-time object detector for embedded applications. IEEE Access, DOI: 10.1109/ACCESS.2019.2941547.
- McKinney, W. 2010. Data structures for statistical computing in Python. Proceedings of the 9th Python in Science Conference, ss. 51–56.
- Microsoft. 2021. Visual Studio Code Documentation: Debugging. Microsoft Docs. <https://code.visualstudio.com/docs/debugtest/debugging> [Son erişim tarihi: 27.05.2025].
- Mohammed, A., Ibrahim, H.M. and Omar, N.M. 2025. Optimizing RetinaNet anchors using differential evolution for improved object detection. Scientific Reports, 15: 20101. <https://doi.org/10.1038/s41598-025-02888-x>
- NVIDIA. 2024. NVIDIA cuDNN: GPU-accelerated library for deep neural networks. NVIDIA Developer. <https://developer.nvidia.com/cudnn> [Son erişim tarihi: 27.05.2025].
- Özçelik, O., 2024. Derin öğrenme ile uydu fotoğraflarından askeri uçak tiplerinin tespiti. Yüksek Lisans Dönem Projesi, İzmir Kâtip Çelebi Üniversitesi, Fen Bilimleri Enstitüsü, Yazılım Mühendisliği Anabilim Dalı.

- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L. vd. 2019. PyTorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32: 8024–8035.
- Patki, Y. and Gawade, T., 2024. Object detection using Faster R-CNN. ResearchGate, December 2024. Available at: <https://www.researchgate.net/publication/387275821> [Accessed 9 Aug. 2025].
- Polk, S.L., Chen, K., Sun, W., Crawford, M.M. ve Li, W., 2023. Unsupervised diffusion and volume maximization-based clustering of hyperspectral images. *IEEE Transactions on Geoscience and Remote Sensing*, 61, ss. 1–15.
- Qifang, X., Guoqing, Y. and Pin, L. 2018. Aircraft detection of high-resolution remote sensing image based on Faster R-CNN model and SSD model. *Proceedings of the 2018 International Conference on Image and Graphics Processing*, ss. 133–137.
- Redmon, J., Divvala, S., Girshick, R. and Farhadi, A. 2016. You only look once: Unified, real-time object detection. *CVPR*, ss. 779–788.
- Redmon, J. and Farhadi, A. 2018. YOLOv3: An incremental improvement. *arXiv preprint*, arXiv:1804.02767.
- Shoaib, H.A., Nabil, H.R., Rahman, M.A., Kabir, M.M., Mridha, M.F. and Shin, J., 2025. Advancements and challenges of deep learning architectures for aerial image analysis: A systematic review. *Intelligent Systems with Applications*, 27, 200537.
- TensorFlow, 2024. TensorBoard: TensorFlow’s visualization toolkit. Google. <https://www.tensorflow.org/tensorboard> [Son erişim tarihi: 27.05.2025].
- Tian, Z., Huang, J., Yang, Y. ve Nie, W. 2023. KCFS-YOLOv5: a high-precision detection method for object detection in aerial remote sensing images. *Applied Sciences*, 13(1), 649, İsviçre.
- Ultralytics. 2024. Hyperparameter Evolution for YOLOv5. Ultralytics Documentation. <https://docs.ultralytics.com/guides/hyperparameter-tuning/> [Erişim tarihi: 3 Ağustos 2025].
- Wang, Y., Zhang, H. and Liu, M. 2023. Faster and lightweight: An improved YOLOv5 object detector for remote sensing images. *Remote Sensing*, 15(20): 4974.

- Waskom, M.L. 2021. seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60): 3021.
- Xia, G., Bai, X., Ding, J., Zhu, Z., Belongie, S., Luo, J., Datcu, M., Pelillo, M. and Zhang, L. 2018. DOTA: A large-scale dataset for object detection in aerial images. *CVPR*, ss. 3974–3983.
- Yu, R., Wan, S., Wang, Y., Gao, C.-X., Gan, L., Zhang, Z. and Zhan, D.-C., 2025. Reward models in deep reinforcement learning: A survey. *International Joint Conference on Artificial Intelligence (IJCAI) 2025, Survey Track (to appear)*, arXiv:2506.15421.
- Zhang, H., Hong, X., and Zhu, L., 2019. Detecting small objects in thermal images using single-shot detector. *Journal of Electronic Imaging*, 28(4), 043011.
- Zhao, Z.Q., Zheng, P., Xu, S.T. and Wu, X. 2019. Object detection with deep learning: A review. *IEEE Transactions on Neural Networks and Learning Systems*, 30(11): 3212–3232.

ÖZGEÇMİŞ

Mustafa Candođdu ÇUTUR

ÖĞRENİM BİLGİLERİ

Yüksek Lisans 2022–2025	Akdeniz Üniversitesi Fen Bilimleri Enstitüsü, Elektrik-Elektronik Mühendisliği Anabilim Dalı, Antalya
Lisans 2010–2014	Gazi Üniversitesi Teknoloji Fakültesi, Elektrik-Elektronik Mühendisliği Bölümü, Ankara

MESLEKİ VE İDARİ GÖREVLER

Stajyer Mühendis 08.07.2013–23.08.2013	ASELSAN A.Ş. SST – Elektronik Tasarım Birimi, Ankara
--	--