

**T.C.  
HAKKARİ ÜNİVERSİTESİ  
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ**

**YÜKSEK LİSANS TEZİ  
ELEKTRİK – ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI**

**İNSANSIZ HAVA ARACI (İHA) İLE SÜRÜ ANALİZİ VE YÖNETİMİNDE  
DERİN ÖĞRENME TEKNİKLERİNİN KULLANILMASI**

**Çetin YALÇIN**

**Danışman  
Dr. Öğretim Üyesi Yusuf Yargı BAYDİLLİ**

**HAKKARİ, 2025**

**T.C.**  
**HAKKARI ÜNİVERSİTESİ**  
**LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ MÜDÜRLÜĞÜNE**  
**HAKKARI**

Elektrik – Elektronik Mühendisliği Anabilim Dalı Yüksek Lisans Programı öğrencisi Çetin YALÇIN tarafından hazırlanan İNSANSIZ HAVA ARACI (İHA) İLE SÜRÜ ANALİZİ VE YÖNETİMİNDE DERİN ÖĞRENME TEKNİKLERİNİN KULLANILMASI başlıklı bu tez, 01.08.2025 tarihinde yapılan savunma sonucunda aşağıda isimleri bulunan jüri üyelerince oy birliği ile kabul edilmiştir.

	<b>Ünvanı, Adı Soyadı</b>	<b>Kurumu</b>	<b>İmzası</b>
Başkan:	Doç Dr. Emrullah SONUÇ	Karabük Üniversitesi	
Üye:	Dr. Öğretim Üyesi Erkan YAŞAR	Hakkari Üniversitesi	
Üye:	Dr. Öğretim Üyesi Yusuf Yargı BAYDİLLİ	Hakkari Üniversitesi	

Jüri üyeleri tarafından kabul edilen bu tez, Enstitü Yönetim Kurulunun ..... sayılı kararıyla ..... tarihinde onaylanmıştır.

Prof. Dr. Mehmet ŞAHİN  
Enstitü Müdürü V.

## TEZ BİLDİRİMİ

Bu tezde sunulan tüm bilgi ve sonuçların, bilimsel yöntemlerle yürütülen gerçek deney ve gözlemler çerçevesinde tarafımdan elde edildiğini, çalışmada bana ait olmayan tüm veri, düşünce, sonuç ve bilgilere bilimsel etik kuralların gereği olarak eksiksiz şekilde uygun atıf yaptığımı ve kaynak göstererek belirttiğimi beyan ederim.

.../.../2025

İmza

Çetin YALÇIN

## ÖZET

### İNSANSIZ HAVA ARACI (İHA) İLE SÜRÜ ANALİZİ VE YÖNETİMİNDE DERİN ÖĞRENME TEKNİKLERİNİN KULLANILMASI

Çetin YALÇIN

Yüksek Lisans Tezi, Elektrik – Elektronik Mühendisliği Anabilim Dalı

Tez Danışmanı: Dr. Öğretim Üyesi Yusuf Yargı BAYDİLLİ

2025, 69 Sayfa

Tarım 4.0 devrimiyle birlikte insansız hava araçları (İHA) ve derin öğrenme tekniklerinin entegrasyonu, hayvancılık sektöründe önemli yeniliklere kapı aralamaktadır. Bu tez çalışmasında, İHA görüntüleri ile modern nesne tespiti algoritmaları ve veri artırma stratejileri bir araya getirilerek sürülerin etkin ve doğru biçimde izlenmesi amaçlanmıştır. Batman ili Kozluk ilçesinde SJRC F22S Pro 4K kameralı İHA ile farklı irtifalarda çekilen toplam 131 keçi ve 1531 sığır görüntüsünden oluşan çok ölçekli bir veri seti oluşturulmuştur. Bu veri seti, hem küçük nesne boyutları (32–64 piksel) hem de değişken irtifa koşulları (0.5 m – 30 m) açısından literatürde nadir görülen bir zenginliğe sahiptir.

Çalışma kapsamında tek aşamalı, iki aşamalı ve transformasyona dayalı nesne tespiti algoritmaları ile birlikte YOLOv5, YOLOv8 ve YOLO11 mimarilerinin farklı varyantları detaylı biçimde karşılaştırılmıştır. Toplamda 20’den fazla model yapılandırması incelenmiş olup, bu karşılaştırmada temel modeller arasında Cascade R-CNN öne çıkarken, YOLOv8m hız ve performans açısından tüm modeller arasında üstün bir performans sergilemiştir.

Modern veri artırma tekniklerinin model performansı üzerindeki etkisi detaylı olarak incelenmiş; özellikle orijinal veriyle harmanlanan artırma stratejilerinin doğruluk oranlarını anlamlı düzeyde artırdığı gözlemlenmiştir.

Sonuç olarak, bu çalışma, nesne tespiti algoritmaları ile veri artırma tekniklerinin entegrasyonunun, hayvancılıkta sürü izleme süreçlerini anlamlı düzeyde geliştirdiğini ortaya koymuştur.

**Anahtar Kelimeler:** İnsansız Hava Aracı (İHA), derin öğrenme, sürü analizi, veri artırma, Tarım 4.0, nesne tespiti.

## ABSTRACT

### USING DEEP LEARNING TECHNIQUES IN HERD ANALYSIS AND MANAGEMENT WITH UNMANNED AERIAL VEHICLES (UAVS)

Çetin YALÇIN

Master of Science Thesis, Department of Electric – Electronic Engineering

Supervisor: Assist. Prof. Dr. Yusuf Yargı BAYDİLLİ

2025, 69 Pages

The Agriculture 4.0 revolution, integrating unmanned aerial vehicles (UAVs) and deep learning, fosters transformative innovations in livestock management. This thesis leverages UAV imagery, advanced object detection algorithms, and data augmentation strategies to enhance the efficiency and accuracy of herd monitoring. A multiscale dataset, comprising 131 goat and 1531 cattle images captured at 0.5–30 m altitudes using an SJRC F22S Pro 4K UAV in Kozluk, Batman, was developed. This dataset, distinguished by small object sizes (32–64 pixels) and diverse altitude conditions, offers unique richness in the literature.

Within the scope of the study, single-stage, two-stage, and transformation-based object detection algorithms, along with different variants of the YOLOv5, YOLOv8, and YOLO11 architectures, were compared in detail. More than 20 model configurations in total were examined. In this comparison, Cascade R-CNN stood out among the baseline models, while YOLOv8m demonstrated superior performance in terms of both speed and performance among all models.

The impact of modern data augmentation techniques on model performance was investigated in detail; it was observed that augmentation strategies, particularly those blended with the original data, significantly increased the accuracy rates.

In conclusion, this study demonstrates that the integration of object detection algorithms with data augmentation techniques significantly improves herd monitoring processes in livestock farming.

**Key Words:** Unmanned Aerial Vehicle (UAV), deep learning, herd analysis, data augmentation, Agriculture 4.0, object detection.

## ÖNSÖZ

Bu çalışmanın her aşamasında yanımda olan, bilgi ve deneyimiyle yolumu aydınlatan, beni yönlendiren ve destekleyen çok kıymetli danışmanım Dr. Öğr. Üyesi Yusuf Yargı BAYDİLLİ'ye en derin şükranlarımı sunarım.

Sadece bir akademik rehber değil, aynı zamanda bir ilham kaynağı olarak; karşılaştığım her zorlukta gösterdiği sabır, anlayış ve yol gösterici tutumuyla bu tez çalışmasının şekillenmesinde ve tamamlanmasında büyük rol oynamıştır. Gerek teknik bilgi birikimi gerekse araştırma sürecine olan titiz yaklaşımı sayesinde birçok engeli kolaylıkla aşmam mümkün olmuştur.

Kendisinin akademik birikimi ve vizyonu, bu çalışmanın yalnızca bir tez olmanın ötesine geçmesini sağlamış, bana sadece bu alanda değil, araştırma etiği ve bilimsel düşünce biçimi açısından da büyük kazanımlar sunmuştur. Bu yolculukta onun rehberliği olmasaydı, bu tez bugün sahip olduğu derinlik ve kaliteye ulaşamazdı.

Saygıdeğer hocamın üzerimdeki emekleri asla unutulmayacak olup, kendisine olan minnet borcumu kelimelerle ifade etmek yetersiz kalacaktır. Bilimsel çalışmalarında başarılarının devamını içtenlikle diler, onun öğrencisi olmaktan her zaman onur duyacağımı belirtmek isterim.

Çetin YALÇIN

## İÇİNDEKİLER

TEZ BİLDİRİMİ.....	ii
ÖZET.....	iii
ABSTRACT.....	iv
ÖNSÖZ .....	v
İÇİNDEKİLER .....	vi
SİMGELER VE KISALTMALAR DİZİNİ .....	viii
ŞEKİLLER DİZİNİ.....	x
ÇİZELGELER DİZİNİ .....	xii
1. GİRİŞ .....	1
2. KAYNAK ÖZETLERİ .....	3
3. MATERYAL VE YÖNTEM .....	10
3.1. İHA Özellikleri ve Uçuş Protokolleri .....	10
3.2. Veri Toplama Süreci .....	12
3.3. Video Kare Çıkarma Algoritmaları.....	13
3.4. Anotasyon ve Etiketleme .....	17
3.5. Veri Arttırma Teknikleri .....	18
3.5.1. CutOut Veri Arttırma Tekniği.....	19
3.5.2. CutMix Veri Arttırma Tekniği.....	20
3.5.3. MixUp Veri Arttırma Tekniği.....	21
3.5.4. Mosaic Veri Arttırma Tekniği.....	22
3.6. Nesne Tespit Modelleri .....	22
3.6.1. Tek Aşamalı Nesne Tespit Modelleri .....	23
3.6.2. İki Aşamalı Nesne Tespit Modelleri .....	24
3.7. Eğitim Ortamı ve Donanım.....	24
3.8. Performans Ölçütleri .....	24
4. BULGULAR VE TARTIŞMA .....	26
4.1. Temel Modellerin Orijinal Verilerde Performans Analizi .....	26
4.1.1. Sığır Veri Seti Üzerinde Temel Modellerin Performansı .....	26
4.1.2. Keçi Sürüsü Veri Seti Üzerinde Temel Modellerin Performansı.....	27
4.2. YOLO Varyantlarının Orijinal Verilerde Performans Analizi.....	28

4.2.1. Sığır Veri Seti Üzerinde YOLO Varyantlarının Performansı .....	29
4.2.2. Keçi Sürüsü Veri Seti Üzerinde YOLO Varyantlarının Performansı .....	30
4.3. Orijinal Veri Setlerinde Kıyaslanan Modellerin Genel Performans Analizi .	31
4.4. Veri Arttırma Tekniklerinin Etkisinin İncelenmesi .....	37
4.4.1. Cascade R-CNN ile Veri Arttırma Tekniklerinin Performans Analizi .....	37
4.4.2. YOLOv8m ile Veri Arttırma Tekniklerinin Performans Analizi.....	39
4.5. Veri Arttırmanın Modellerin Genel Performansına Etkisinin İncelenmesi ...	41
4.6. Görsel Performans Analizi .....	44
4.7. Genel Değerlendirme .....	56
5. SONUÇ .....	58
5.1. Sınırlılıklar .....	60
5.2. Gelecek Çalışmalar İçin Öneriler .....	60
KAYNAKLAR .....	62
ÖZGEÇMİŞ .....	69

## SİMGELER VE KISALTMALAR DİZİNİ

### Simgeler

**m:** Metre

**m/s:** Metre/Saniye

**px:** Piksel

### Kısaltmalar

**3B:** 3 Boyutlu

**ABFF:** Attention-Based Feature Fusion (Dikkat Tabanlı Özellik Füzyonu)

**AdamW:** Adaptive Moment Estimation with Weight Decay

**AGFF:** Attention-Guided Feature Fusion (Dikkat Yönlendirmeli Özellik Füzyonu)

**AP:** Average Precision (Ortalama Hassasiyet)

**AP<sub>50</sub>:** Average Precision at IoU=0.50 (IoU=0.50 Eşiğinde Ortalama Hassasiyet)

**AP<sub>s</sub>:** Average Precision for Small Objects (Küçük Nesnelere için Ortalama Hassasiyet)

**cGAN:** Conditional Generative Adversarial Network (Koşullu Üretici Çekişmeli Ağ)

**CNN:** Convolutional Neural Network (Evrişimli Sinir Ağı)

**CT:** Computed Tomography (Bilgisayarlı Tomografi)

**CUDA:** Compute Unified Device Architecture

**cuDNN:** CUDA Deep Neural Network

**CVAT:** Computer Vision Annotation Tool

**DA:** Domain Adaptation (Etki Alanı Uyarlaması)

**DCGAN:** Deep Convolutional Generative Adversarial Network (Derin Evrişimli Üretici Çekişmeli Ağ)

**DETR:** Detection Transformer

**DR:** Domain Randomization (Etki Alanı Rastgeleleştirme)

**FPS:** Frames Per Second

**FPN:** Feature Pyramid Network

**GAN:** Generative Adversarial Network (Üretici Çekişmeli Ağ)

**GIS:** Geographic Information System (Coğrafi Bilgi Sistemi)

**GPS:** Global Positioning System

**GSAM:** Global Spatial Attention Module (Küresel Konumsal Dikkat Modülü)

**İHA:** İnsansız Hava Aracı

**IoT:** Internet of Things (Nesnelerin İnterneti)

**IoU:** Intersection over Union

**LIDAR:** Light Detection and Ranging (Işık Tabanlı Algılama ve Menzil Tayini)

**mAP:** Mean Average Precision (Ortalama Hassasiyet)

**MFL:** Modified Focal Loss (Değiştirilmiş Odak Kaybı)

**NAS:** Neural Architecture Search (Sinirsel Mimari Arama)

**NAM:** Normalization-based Attention Module (Normalleştirme Tabanlı Dikkat Modülü)

**PFAM:** Pixel Feature Attention Module (Piksel Özellik Dikkat Modülü)

**PLF:** Precision Livestock Farming (Hassas Hayvancılık Yönetimi)

**RFID:** Radio-Frequency Identification (Radyo Frekanslı Tanımlama)

**RGB:** Red-Green-Blue (Kırmızı-Yeşil-Mavi)

**RPN:** Region Proposal Network

**SA-FPN:** Scale-Aware Feature Pyramid Network (Ölçek Duyarlı Özellik Piramidi Ağı)

**SAGM:** Scale-Aware Attention-Guided Module (Ölçek Duyarlı Dikkat Yönlendirmeli Modül)

**SOTA:** State-of-the-Art (En Güncel Teknoloji)

**SPHs:** Swin Transformer Prediction Heads (Swin Dönüştürücü Tahmin Başlıkları)

**SSD:** Single Shot MultiBox Detector

**WGAN-GP:** Wasserstein Generative Adversarial Network with Gradient Penalty (Gradyan Cevalı Wasserstein Üretici Çekişmeli Ağ)

**YOLO:** You Only Look Once

## ŞEKİLLER DİZİNİ

Şekil 3.1. Videoların her karesini klasöre kaydeden algoritma .....	14
Şekil 3.2. Hayvan görüntülerinin zaman dilimlerini ayıran algoritma.....	14
Şekil 3.3. Her saniyeden sabit aralıklarla kare çıkaran algoritma.....	15
Şekil 3.4. İdeal karelerin zaman dilimlerini içeren metin dosyalarının organizasyonu .....	16
Şekil 3.5. Times klasöründeki zaman dilimlerine göre kare çıkaran algoritma....	17
Şekil 3.6. Sığır görüntülerine CutOut uygulanması .....	20
Şekil 3.7. Sığır görüntülerine CutMix uygulanması .....	21
Şekil 3.8. Sığır görüntülerine MixUp uygulanması .....	21
Şekil 3.9. Sığır görüntülerine Mosaic uygulanması .....	22
Şekil 3.10. Tek ve iki aşamalı nesne tespit modellerinin karşılaştırılması .....	23
Şekil 4.1. Temel modellerin sığır ve keçi sürüsü veri setindeki recall ve mAP değer karşılaştırması.....	32
Şekil 4.2. YOLOv5 varyantlarının sığır ve keçi sürüsü veri setindeki recall ve mAP değer karşılaştırması .....	33
Şekil 4.3. YOLOv8 varyantlarının sığır ve keçi sürüsü veri setindeki recall ve mAP değer karşılaştırması .....	35
Şekil 4.4. YOLO11 varyantlarının sığır ve keçi sürüsü veri setindeki recall ve mAP değer karşılaştırması .....	36
Şekil 4.5. Veri arttırma teknikleri ile eğitilen Cascade R-CNN modelinin sığır ve keçi sürüsü veri setindeki recall ve mAP değer karşılaştırması. ....	42
Şekil 4.6. Veri arttırma teknikleri ile eğitilen YOLOv8m modelinin sığır ve keçi sürüsü veri setindeki recall ve mAP değer karşılaştırması.....	43
Şekil 4.7. Orijinal ve arttırılmış veri seti ile eğitilen YOLO modellerinin sığır tespit sonuçları (Görsel 1). ....	45
Şekil 4.8. Orijinal ve arttırılmış veri seti ile eğitilen YOLO modellerinin sığır tespit sonuçları (Görsel 2). ....	47
Şekil 4.9. Orijinal ve arttırılmış veri seti ile eğitilen YOLO modellerinin sığır tespit sonuçları (Görsel 3). ....	49

Şekil 4.10. Orijinal ve arttırılmış veri seti ile eğitilen YOLO modellerinin keçi sürüsü tespit sonuçları (Görsel 1).....	51
Şekil 4.11. Orijinal ve arttırılmış veri seti ile eğitilen YOLO modellerinin keçi sürüsü tespit sonuçları (Görsel 2).....	53
Şekil 4.12. Orijinal ve arttırılmış veri seti ile eğitilen YOLO modellerinin keçi sürüsü tespit sonuçları (Görsel 3).....	55



## ÇİZELGELER DİZİNİ

Çizelge 3.1. Veri setinin özellikleri.....	13
Çizelge 3.2. Geleneksel ve modern veri artırma tekniklerinin özellik karşılaştırması .....	19
Çizelge 4.1. Temel modellerin sığır veri seti üzerindeki performans sonuçları ...	27
Çizelge 4.2. Temel modellerin keçi sürüsü veri seti üzerindeki performans sonuçları .....	28
Çizelge 4.3. YOLO varyantlarının sığır veri seti üzerindeki performans sonuçları .....	29
Çizelge 4.4. YOLO varyantlarının keçi sürüsü veri seti üzerindeki performans sonuçları .....	30
Çizelge 4.5. Veri artırma teknikleri ile eğitilen Cascade R-CNN modelinin sığır veri seti üzerindeki performans sonuçları .....	38
Çizelge 4.6. Veri artırma teknikleri ile eğitilen Cascade R-CNN modelinin keçi sürüsü veri seti üzerindeki performans sonuçları.....	39
Çizelge 4.7. Veri artırma teknikleri ile eğitilen YOLOv8m modelinin sığır veri seti üzerindeki performans sonuçları .....	40
Çizelge 4.8. Veri artırma teknikleri ile eğitilen YOLOv8m modelinin keçi sürüsü veri seti üzerindeki performans sonuçları .....	41

## 1. GİRİŞ

Tarım 4.0 paradigması, modern teknolojilerin tarımsal üretime entegrasyonu ile verimlilik ve sürdürülebilirlik hedeflerini dönüştürmektedir (Sevli, 2023). Bu bağlamda, İnsansız Hava Araçları (İHA'lar), özellikle hayvancılık yönetiminde devrim niteliğinde bir çözüm sunmaktadır. İHA'lar, geniş arazilerde manuel denetim süreçlerini otomatikleştirerek, hayvan sağlığı izleme, sürü hareket analizi ve çevresel koşulların değerlendirilmesi gibi kritik görevleri hızlı ve uygun maliyetli bir şekilde gerçekleştirmektedir (Banhazi vd., 2012). Derin öğrenme teknolojileri ise İHA görüntülerinden elde edilen verilerin analizini otomatikleştirerek ve doğruluğu artırarak bu süreçleri daha etkin hale getirmektedir (Aquilani vd., 2022). Örneğin, geleneksel yöntemlerle saatler süren sürü denetimleri, İHA'lar ve derin öğrenme ile dakikalar içinde tamamlanabilmekte, bu da gıda güvenliği, hayvan refahı ve ekonomik sürdürülebilirlik açısından önemli avantajlar sağlamaktadır (Bailey vd., 2021). Bu çalışma, İHA tabanlı sürü analizinde derin öğrenme modellerinin performansını ve modern veri artırma tekniklerinin bu modeller üzerindeki etkisini sistematik bir şekilde incelemeyi amaçlamaktadır.

İHA tabanlı hayvancılık izleme, derin öğrenme modellerinin yüksek doğrulukla çalışması için büyük ve çeşitli veri setlerine ihtiyaç duymaktadır (Bengio, 2009). Ancak, bu alanda karşılaşılan temel zorluklar, model performansını olumsuz etkilemektedir. İlk olarak, karmaşık arka planlar ve değişen aydınlatma koşulları, hayvanların çevreyle karışmasına neden olmakta; özellikle veri setlerinde hayvanların örtüşmeleri ve arka plan benzerlikleri tespiti zorlaştırmaktadır (Wu vd., 2022). İkinci olarak, yüksek irtifadan çekilen görüntülerdeki küçük nesnelere ve düşük irtifadaki büyük nesnelere (sığırlar) farklı ölçeklerdeki tespit ihtiyaçlarını ortaya çıkarmaktadır (Mufford vd., 2022). Üçüncü olarak, veri setlerinin sınırlı boyutu ve çeşitlilik eksikliği, modellerin genelleme yeteneğini kısıtlamaktadır (Schmidhuber, 2015). Geleneksel veri artırma teknikleri (örneğin, basit çevirme, döndürme ve parlaklık ayarı) bu karmaşık görevler için yetersiz kalmaktadır (Alanezi vd., 2022). Nesne tespiti gibi karmaşık görevlerin hem sınıf hem de konum bilgisi gerektiriyor; bu durum, modern veri artırma tekniklerinin (CutOut, CutMix,

MixUp, Mosaic) potansiyelini deęerlendirme gereklilięini ortaya koymaktadır (Jiang vd., 2020). Bu alıřma, deęiřken irtifa kořullarında ve heterojen veri setlerinde derin ęrenme modellerinin performansını artırmak iin modern veri artırma tekniklerinin etkisini analiz etmeyi hedeflemektedir.

Bu alıřmanın temel amacı, İHA tabanlı hayvancılık izlemede derin ęrenme modellerinin performansını deęerlendirmek ve modern veri artırma tekniklerinin bu modeller üzerindeki etkisini sistematik bir řekilde incelemektir. Bu baęlamda,  alt hedef belirlenmiřtir:

1. **Model Performansını Deęerlendirme:** Tek ařamalı ve iki ařamalı derin ęrenme tabanlı nesne tespit modellerinin, Kei srs ve sıęır veri setlerindeki performansını karřılařtırmaktır.
2. **Veri Artırma Tekniklerinin Etkisini İnceleme:** eřitli modern veri artırma tekniklerinin, model performansına olan etkisini analiz etmek ve bu tekniklerin irtifa-nesne leęi baęlamındaki katkılarını belirlemektir.
3. **En Etkili Stratejiyi Belirleme:** Kei srs ve sıęır tespiti iin en uygun veri artırma stratejisini ortaya koyarak, gerek dnya senaryolarında uygulanabilir zmler sunmaktır.

Bu alıřma hem akademik literatre katkı saęlamayı hem de tarımsal uygulamalarda İHA tabanlı sr ynetimini optimize etmeyi amalamaktadır.

## 2. KAYNAK ÖZETLERİ

İnsansız Hava Araçları (İHA'lar) tabanlı sürü analizi ve yönetimi, derin öğrenme tekniklerinin tarımsal uygulamalara entegrasyonu ile son yıllarda önemli bir araştırma alanı haline gelmiştir (Aquilani vd., 2022; Banhazi vd., 2012). Ancak, bu teknolojilerin etkinliğini artırmak için derin öğrenme tabanlı nesne tespiti ve veri artırma teknikleri gibi yenilikçi yaklaşımlar, metodolojik ve uygulamalı zorlukların üstesinden gelmek için kritik öneme sahiptir. Bu yaklaşımlar, özellikle mikroorganizma sayımı gibi karmaşık görüntü analizlerinde kanıtlanmış yöntemlerden (Zhang vd., 2022) esinlenerek, hayvan tespitinde adapte edilebilir. Hassas hayvancılık yönetimi (Precision Livestock Farming - PLF) bağlamında, İHA'lar, otomatik hayvan sayımı, davranış izleme, mera sağlığı değerlendirmesi ve tehdit tespiti gibi işlevleriyle geleneksel yöntemlerin sınırlamalarını aşarak, hayvan refahını artırma, kaynak kullanımını optimize etme ve çevresel etkileri azaltma hedeflerini desteklemektedir (Alanezi vd., 2022; Yousefi vd., 2022; Bailey vd., 2021). Bu bölümde, İHA tabanlı sistemlerin tarım ve hayvancılıktaki rolleri, hayvan tespiti için geliştirilen derin öğrenme modelleri, küçük nesne tespitindeki zorluklar, veri artırma teknikleri ve sentetik veri üretiminin katkıları, mevcut literatür ışığında detaylı bir şekilde ele alınmıştır.

İHA'lar, hassas hayvancılıkta hayvan refahı, verimlilik ve çevresel sürdürülebilirlik açısından benzersiz avantajlar sağlamaktadır. Alanezi vd. (2022), İHA'ların hayvan sayımı, sağlık izleme (termal görüntüleme), davranış analizi (yürüyüş kalıpları) ve mera değerlendirmesi (vegetasyon indeksleme) gibi 27 farklı uygulama alanını sistematik olarak sınıflandırmış; RGB, multispektral ve LIDAR sensörlerinin bu süreçlerdeki etkinliğini vurgulamıştır. Örneğin, Mufford vd. (2022), İHA'ların sığırlarda ısı stresi izlemede etkili olduğunu, besi yerindeki sığırların mera sığırlarına kıyasla istatistiksel olarak anlamlı düzeyde yüksek solunum hızı ( $89.2 \pm 4.2$  vs.  $67.3 \pm 3.8$  soluk/dakika;  $*p* < 0.01$ ) ve deri sıcaklığı ( $35.7^{\circ}\text{C} \pm 0.3$  vs.  $34.6^{\circ}\text{C} \pm 0.4$ ;  $*p* < 0.05$ ) sergilediğini raporlamıştır. Bu tür nicel bulgular, İHA'ların hayvan sağlığını izlemede manuel yöntemlere kıyasla daha nesnel ve tekrarlanabilir veriler sunduğunu göstermektedir. Benzer şekilde, Yousefi vd. (2022), 2016-2021 yılları arasında yayınlanan 75 çalışmayı analiz ederek,

İHA'ların sığır türlerinde %63.4 oranla en çok araştırılan hayvan grubu olduğunu, ancak koyun ve kanatlı hayvanların yetersiz temsil edildiğini belirtmiştir. Bu, çok türlü tespit modellerine olan ihtiyacı ortaya koymaktadır.

İHA'ların yüksek çözünürlüklü RGB, termal ve multispektral görüntüleme yetenekleri, hayvanlara minimum rahatsızlık verirken operatör güvenliğini korumaktadır (Alanezi vd., 2022). Sevli (2023), IoT sensörleri ve İHA tabanlı multispektral görüntüleme entegrasyonunun buğday ve mısır tarlalarında su kullanım verimliliğini %23 artırdığını, rekolte tahmin doğruluğunu %89.5'e çıkardığını ve işgücü maliyetlerini %34 azalttığını göstermiştir. Bu bulgular, İHA'ların operasyonel verimliliği artırarak Tarım 4.0 hedeflerine katkıda bulunduğunu doğrulamaktadır. Ayrıca, Jiang vd. (2020), Faster R-CNN modelinin grup halinde barındırılan keçilerin davranışlarını otomatik olarak tanımda %92.3 mAP elde ettiğini, bu sayede hayvan refahı izlemede manuel gözleme kıyasla daha yüksek hız ve tutarlılık sağladığını raporlamıştır. Benzer şekilde, Qiao vd. (2019), Mask R-CNN'nin İHA görüntülerinde sığır segmentasyonu için uyarlanmasının %90'ın üzerinde mAP ile otomatik izlemeyi mümkün kıldığını vurgulamıştır. Bu çalışmalar, derin öğrenme tabanlı modellerin İHA görüntülerinden hayvan davranışlarını ve sağlık durumlarını analiz etmede kritik bir rol oynadığını göstermektedir.

Derin öğrenme tabanlı nesne tespiti, İHA'ların hassas hayvancılık uygulamalarındaki etkinliğini artırmak için temel bir teknoloji olarak öne çıkmaktadır. Liu vd. (2020), derin öğrenme tabanlı nesne tespit mimarilerini iki aşamalı (bölge teklifi tabanlı) ve tek aşamalı (doğrudan regresyon) yaklaşımlar olarak kapsamlı şekilde sınıflandırmış; iki aşamalı modellerin PASCAL VOC gibi veri setlerinde doğruluk (%73.2 mAP), tek aşamalı modellerin ise gerçek zamanlı performans (20 FPS) avantajı sağladığını, ancak küçük nesnelere ölçek değişimi ve düşük çözünürlük nedeniyle AP değerlerinin MS COCO'da %34.2'ye kadar düştüğünü vurgulamıştır. Bu zorluğa karşı FPN gibi çok ölçekli füzyon tekniklerinin performansı %8.5 artırdığı belirtilmiştir. Zhao vd. (2019), nesne tespiti yöntemlerini iki aşamalı (örn. Faster R-CNN, Cascade R-CNN) ve tek aşamalı (örn. YOLO, SSD) yaklaşımlar olarak sınıflandırmış; Faster R-CNN'nin MS COCO veri setinde %36.2 mAP@0.5:0.95, YOLO ve SSD'nin ise gerçek

zamanlı uygulamalarda hız avantajı sağladığını belirtmiştir. YOLOv3 gibi modeller, otonom araçlarda %70.2 mAP performansı sunarken (Redmon & Farhadi, 2018), küçük nesnelere RetinaNet'e (%28.4 mAP) kıyasla sınırlı kalmıştır (Carranza-García vd., 2021). Ren vd. (2015), Faster R-CNN'nin Region Proposal Network (RPN) ile iki aşamalı nesne tespiti için %73.2 mAP (PASCAL VOC 2007) sunduğunu, ancak küçük nesnelere düşük performans (%49.1 AP) ve hesaplama karmaşıklığı gibi sınırlamalarla karşılaştığını raporlamıştır. Buna karşılık, Cai ve Vasconcelos (2018), Cascade R-CNN'nin kademeli IoU eşikleriyle yanlış pozitifleri azaltarak MS COCO'da %42.8 mAP@0.5:0.95 elde ettiğini, ancak hesaplama yükü ve küçük nesne tespiti gibi zorluklarla karşılaştığını vurgulamıştır.

Tek aşamalı modellerde ise, Bochkovskiy vd. (2020), YOLOv4'ün bag-of-freebies ve bag-of-specials tekniklerini optimize edilmiş anchor tabanlı mimariyle birleştirerek MS COCO'da %43.5 mAP@0.5:0.95 ve 65 FPS hız sunduğunu belirtmiştir. Son dönemde, Deformable DETR gibi dönüştürücü tabanlı modeller, esnek öznitelik örneklemeyle küçük nesnelere %43.8 AP performansı sunarak yeni bir alternatif oluşturmuştur (Zhu vd., 2021). Wang vd. (2023a), YOLOv7'nin trainable bag-of-freebies yaklaşımıyla MS COCO'da %56.8 mAP@0.5:0.95 elde ettiğini, ancak küçük nesnelere ve karmaşık arka planlarda sınırlamalar yaşadığını raporlamıştır. Li vd. (2022), YOLOv6'nın anchor-free mimarisi ve EfficientRep omurgasıyla endüstriyel uygulamalarda %43.1 mAP@0.5:0.95 sunduğunu, ancak küçük nesnelere düşük performans (%26.0 AP<sub>s</sub>) gösterdiğini belirtmiştir. Küçük nesne tespitine yönelik yenilikçi bir yaklaşım olarak, Wang vd. (2023b), YOLOX-X'in dikkat tabanlı özellik füzyon modülü (ABFF) ve optimize edilmiş kayıp fonksiyonu (MFL) ile VisDrone veri setinde %51.2 mAP (küçük nesnelere %41.7 AP) elde ettiğini, ancak 132.5 GFLOPs hesaplama yükünün pratik kullanımı kısıtladığını raporlamıştır.

Uzaktan algılama görüntülerinde küçük nesne tespiti, arka plan karmaşıklığı ve ölçek çeşitliliği nedeniyle özel zorluklar barındırmaktadır. Ying vd. (2019), MA-FPN mimarisinin Global Spatial Attention Module (GSAM) ve Pixel Feature Attention Module (PFAM) ile konum bilgisini ve piksel düzeyinde detayları koruyarak NWPU (%92.4 mAP), RSOD (%88.9 mAP) ve DOTA (%76.3 mAP) veri setlerinde mevcut yöntemleri geride bıraktığını göstermiştir. Gong vd. (2022),

YOLOv5 tabanlı SPH-YOLOv5 mimarisinin Shallow-branch, Swin Transformer Prediction Heads (SPHs) ve Normalization-based Attention Module (NAM) ile NWPU-VHR10'da %98.0 mAP ve DOTA'da %71.6 mAP elde ettiğini, özellikle küçük nesnelere %5.3 AP artışı sağladığını raporlamıştır. Han vd. (2025), SA-FPN mimarisinin Scale-Aware Attention-Guided Module (SAGM) ile çelik şerit yüzey defekti tespitinde %78.4 mAP (NEU-DET) sunduğunu ve geleneksel FPN'den %3.2 mAP üstünlük sağladığını belirtmiştir. Bu dikkat mekanizmaları, tıbbi görüntüleme optik disk segmentasyonunda da %96.2 doğrulukla kanıtlanmıştır (Bhatkalkar vd., 2020). Yang vd. (2023), dikkat temelli özellik füzyonu (AGFF) ile RetinaNet tabanlı uygulamanın MS COCO'da küçük nesnelere %34.2 AP<sub>s</sub> elde ettiğini, kalabalık sahnelerde kaçırma oranını %19.2 azalttığını raporlamıştır. Li vd. (2017), Light-Head R-CNN'in ayrılabilir evrişim ve ince öznetelik haritalarıyla MS COCO'da %41.5 mAP ve 102 FPS hız sunduğunu, küçük nesnelere %34.1 AP<sub>s</sub> ile Faster R-CNN'den üstün olduğunu göstermiştir.

Veri miktarı kısıtlı senaryolarda, veri artırma teknikleri derin öğrenme modellerinin genelleme yeteneğini artırmada kritik bir rol oynamaktadır. Kaur vd. (2021), veri artırma tekniklerini geometrik (Mosaic, CutMix), fotometrik (renk değişimleri) ve GAN-tabanlı yöntemler olarak sınıflandırmış; CutMix ve Mosaic'in performansı önemli ölçüde iyileştirdiğini, ancak GAN tabanlı yöntemlerin mod çökmesi ve yüksek hesaplama maliyetleri gibi sınırlamalarla karşılaştığını belirtmiştir. Yun vd. (2019), CutMix yönteminin eğitim görüntüleri arasında kesilen ve yapıştırılan yamalar için temel gerçek etiketlerin yama alanlarıyla orantılı olarak karıştırılması prensibiyle ImageNet'te %21.40 top-1 hata oranı sunduğunu ve Mixup ile Cutout'u geride bıraktığını raporlamıştır. Zhang vd. (2018), mixup yönteminin örnekler arası doğrusal davranış benimsemesini sağlayarak CIFAR-10, CIFAR-100 ve ImageNet-2012'de genelleme hatalarında iyileşmeler sağladığını, bozuk etiketlerle eğitimde ezberlemeyi azalttığını göstermiştir. DeVries ve Taylor (2017), Cutout tekniğinin girdideki kare bölgeleri rastgele maskeleyerek WideResNet ile CIFAR-10'da %2.56, CIFAR-100'de %15.20 test hatası elde ettiğini ve mevcut artırma teknikleriyle sinerji sağladığını belirtmiştir.

Uzaktan algılama görüntülerinde nesnelere seyrek dağılımı, standart Mosaic yönteminin hedef içermeyen örnekler üretmesine yol açmaktadır. Zhao vd. (2021), toplulaştırılmış-mosaic (aggregated-mosaic) yöntemiyle VEDAI ve NWPU VHR-10 veri kümelerinde sırasıyla 5.48 ve 2.72 puanlık mAP artışları sağlamıştır. Zhang vd. (2024), Select-Mosaic stratejisinin hedef yoğunluğu hesaplama ve maske alanı optimizasyonu ile AI-TOD'da %2.17, VisDrone2019'da %0.37 AP<sub>50</sub> artışı sunduğunu göstermiştir. Wong vd. (2016), data-space artırmanın (döndürme, ölçekleme, çevirme) feature-space yöntemlerinden daha etkili olduğunu, MNIST veri kümesinde model performansını iyileştirdiğini raporlamıştır.

Generatif modeller, sentetik veri üretiminde önemli bir rol oynamaktadır. Goodfellow vd. (2014), GAN'ların minimax iki-oyunculu çerçeve ile veri dağılımını öğrendiğini teorik olarak ispatlamış; ancak mod çökmesi ve eğitim kararsızlığı gibi zorlukları vurgulamıştır. Beyin MR sentezinde, GAN tabanlı görüntüler Visual Turing Test'te %57.5 doğrulukla gerçeklerden ayırt edilememiş, ancak lokal anatomik tutarsızlıklar sınırlayıcı olmuştur (Han vd., 2018). Mirza ve Osindero (2014), koşullu GAN'ların veri üretimini kontrol edilebilir hale getirdiğini, MNIST'te belirli rakam sınıfları üreterek bu alandaki ilk temeli attığını göstermiştir. Radford vd. (2016), DCGAN mimarisinin strided konvolüsyonlar ve BatchNorm ile LSUN veri kümesinde yüksek kaliteli 64×64 görüntüler ürettiğini, CIFAR-10'da %82.8 doğruluk sunduğunu raporlamıştır. Gulrajani vd. (2017), WGAN-GP'nin gradient penalty terimiyle CIFAR-10'da 8.42 Inception Score ve 40.2 FID elde ederek standart WGAN ve DCGAN'ı geride bıraktığını belirtmiştir. Tremblay vd. (2018), etki alanı rastgeleleştirme (DR) ile VKITTI tabanlı sentetik veriyle eğitilen SSD ve R-FCN modellerinin %10.2 ve %6.9 daha yüksek AP@0.5 sunduğunu göstermiştir.

Tıbbi görüntü sentezinde, Frid-Adar vd. (2018), GAN tabanlı sentetik veriyle karaciğer lezyonlu CT sınıflandırmasında duyarlılığı %78.6'dan %85.7'ye, özgüllüğü %88.4'ten %92.4'e yükselttiğini göstermiştir. Singhal vd. (2023), etki alanı uyarlamasının (DA) görsel, dil ve çoklu ortam veri alanlarında dağılım tabanlı, çekişmeli öğrenme ve sözde etiketleme yöntemleriyle ele alındığını; ancak veri eksikliği ve metrik standartların yokluğunun pratik engeller oluşturduğunu belirtmiştir. Gui vd. (2021), GAN literatürünü algoritmalar, teori ve uygulamalar

ekseninde sentezleyerek mod çökmesi ve eğitim kararsızlığı gibi sorunlara dikkat çekmiştir.

Disiplinlerarası aktarım, İHA tabanlı hassas hayvancılık uygulamalarının geliştirilmesinde önemli bir rol oynamaktadır. Scott vd. (2024), vahşi yaşam izlemeden aktarılabilen termal drone görüntüleme (%89±4 doğruluk), 3B mesh rekonstrüksiyonu ve LIDAR-RFID füzyonu gibi teknikleri hayvancılık izlemeye uyarlamının potansiyelini vurgulamıştır. Bu kapsamda, mikroorganizma sayımı için geliştirilen çok ölçekli öznelik füzyon teknikleri (Zhang vd., 2022), hayvan yoğunluğu yüksek sahnelerde uyarlanabilir. Ancak, hayvancılık izleme çalışmalarının %92'sinin kontrollü koşullarda yürütülmesi, gerçek dünya senaryolarında validasyon eksikliğini ortaya koymaktadır. Ramachandran ve Sangaiyah (2021), İHA tabanlı nesne tespiti ve takibini geleneksel ve derin öğrenme tabanlı yöntemler olarak sınıflandırmış; UAVDT, VisDrone, UAV123 ve DOTA veri setlerinde mAP, IoU ve F1-score metriklerini detaylandırmış; denetimsiz öğrenme ve onboard işlem güvenliği gibi araştırma boşluklarını vurgulamıştır. Wu vd. (2022), TDFA modelinin VisDrone-VID'de %27.27 AP, YOWO modelinin Okutama-Action'da %87.18 AP<sub>50</sub> sunduğunu raporlamış; küçük nesne tespiti, arka plan gürültüsü ve gömülü işlem sınırlamalarını gelecek araştırma öncelikleri olarak tanımlamıştır.

Küçük nesne tespiti, İHA görüntülerinde arka plan karmaşıklığı ve ölçek çeşitliliği nedeniyle temel bir zorluk olarak öne çıkmaktadır. Lin vd. (2017), Focal Loss fonksiyonunun sınıf dengesizliğini adaptif olarak yöneterek RetinaNet ile MS COCO'da %39.1 AP sunduğunu, tek aşamalı detektörlerin hızını korurken iki aşamalı yöntemleri geride bıraktığını göstermiştir. Sinirsel mimari arama (NAS) yöntemleri, otomatik model optimizasyonu ile ölçeklenebilir çözümler sunabilir (Zoph vd., 2018). Carranza-García vd. (2021), YOLOv3'ün KITTI'de %70.2 mAP ve 20 FPS hız sunduğunu, ancak küçük nesnelere RetinaNet'in %28.4 mAP ile üstün olduğunu raporlamıştır. Korkmaz vd. (2024), YOLOv8'in özel bir çiftlik hayvanı tehdit algılama veri setinde %92.3 mAP ile YOLOv5 ve Faster R-CNN'ye üstünlük sağladığını, ancak sınırlı veri seti boyutu ve çevresel değişkenler (örn. Türkiye'deki tozlu koşullar) nedeniyle kısıtlamalar yaşadığını belirtmiştir.

İHA tabanlı hassas hayvancılık uygulamaları, düzenleyici kısıtlamalar, sınırlı uçuş süresi ( $\leq 25$  dakika), hava koşullarına bağımlılık, sensör sınırlamaları, gürültü hassasiyeti ve yüksek başlangıç maliyetleri gibi zorluklarla karşı karşıyadır (Ramachandran & Sangaiah, 2021; Aquilani vd., 2022). Derin öğrenme modellerinde küçük nesne tespiti (örn.  $AP_s \leq 34.2$ ), karmaşık arka planlar ve sınırlı veri seti boyutları genelleme yeteneğini kısıtlamaktadır (Li vd., 2022; Wang vd., 2023a). Tıbbi görüntü analizinde kullanılan çoklu çözünürlüklü dikkat ağları (Shalini & Gopi, 2024), hayvan segmentasyonunda benzer karmaşıklıkları aşmak için örnek alınabilir. Denetimsiz öğrenme eksikliği, onboard işlem güvenliği ve gerçek dünya koşullarında validasyon eksikliği, araştırma boşlukları olarak öne çıkmaktadır (Ramachandran & Sangaiah, 2021; Wu vd., 2022). Gelecek çalışmalar, çoklu sensör füzyonu, dinamik aydınlatma adaptasyon modelleri, model sıkıştırma, denetimsiz öğrenme ve çoklu tür tespiti gibi alanlara odaklanmalıdır. Ayrıca, sentetik veri üretimi ve etki alanı uyarlaması, veri kıtlığı sorununu çözmek için umut verici yaklaşımlar sunmaktadır (Singhal vd., 2023; Tremblay vd., 2018).

İHA tabanlı hassas hayvancılık, derin öğrenme tabanlı nesne tespiti (YOLO serisi, Cascade R-CNN) ve veri artırma teknikleri (CutMix, Select-Mosaic) ile desteklenerek Tarım 4.0 hedeflerine önemli katkılar sağlamaktadır. Bu modeller, hayvan sayımı, sağlık izleme ve davranış analizi gibi görevlerde %90'a varan mAP performansı sunarken; veri artırma teknikleri, sınırlı veri setleriyle genelleme yeteneğini artırmaktadır. Disiplinlerarası aktarımlar (vahşi yaşam izleme, tıbbi görüntüleme), özellikle dikkat mekanizmaları ve sentetik veri üretimi alanlarında kritik avantajlar sağlar. Ancak, küçük nesne tespiti, karmaşık arka planlar ve çevresel sınırlamalar (özellikle Türkiye'deki tozlu koşullar; Korkmaz vd., 2024) gibi zorluklar, metodolojik ve uygulamalı yenilikler gerektirmektedir. Gelecek çalışmalar, çoklu sensör füzyonu, denetimsiz öğrenme ve model sıkıştırma gibi yenilikçi yaklaşımlarla bu alandaki sınırlamaları aşarak, İHA tabanlı hassas hayvancılığı daha erişilebilir ve sürdürülebilir bir çözüm haline getirebilir.

### 3. MATERYAL VE YÖNTEM

Bu bölümde, çalışmada kullanılan metodoloji ayrıntılı bir şekilde sunulacaktır. Çalışmada, İHA'lar ile toplanan veriler, sürü davranışlarını ve yönetimini analiz etmek amacıyla ileri derin öğrenme teknikleriyle entegre edilmiştir. Yaklaşım, sınırlı veri, karmaşık arazi koşulları ve gerçek zamanlı işlem gereksinimleri gibi zorlukları ele alacak şekilde tasarlanmış olup, sağlam, ölçeklenebilir ve tekrarlanabilir bir çözüm geliştirilmesini hedeflemektedir.

Veri toplama, çeşitli sürü senaryolarını yakalamak için İHA'lar ile havadan görüntülerin sistematik olarak toplanmasını içeren temel bir aşamadır. Sonraki bölümde, kullanılan İHA'nın teknik özelliklerini, uçuş protokollerini, toplanan veri türlerini ve veri toplama sürecinde uygulanan etik hususları ayrıntılı bir şekilde açıklanmaktadır.

#### 3.1. İHA Özellikleri ve Uçuş Protokolleri

Çalışmada, hayvan popülasyonlarının doğal davranışlarını bozmadan yüksek çözünürlüklü görüntüleme yapabilmek amacıyla SJRC F22 S2 Pro+ 4K Kameralı Drone kullanılmıştır. İHA'nın teknik özellikleri şunlardır:

##### 1. Kamera Sistemleri:

- 4K Ultra HD Çözünürlük (3840×2160 piksel) ve 1/2,5 inç CMOS sensör, detaylı görüntü yakalama kapasitesi sağlamıştır.
- f/2.8 diyafram açıklığı ve 120° geniş görüş alanı (FOV), farklı ışık koşullarında (özellikle sabah ve öğle saatlerinde) net görüntü elde edilmesini kolaylaştırmıştır.
- 4x dijital yakınlaştırma özelliğiyle uzak mesafelerdeki hayvanların davranışsal detayları rahatlıkla kaydedilmiştir.

##### 2. Stabilizasyon Teknolojisi:

- 3 eksenli mekanik gimbal ve elektronik görüntü sabitleme (EIS) kombinasyonu hem fiziksel titreşimleri hem de hareket kaynaklı bulanıklığı minimize etmiştir. Bu sistem, özellikle düşük irtifalı uçuşlarda kritik öneme sahip olmuştur.

### 3. Uçuş Performansı:

- Maksimum 10 m/s (36 km/s) rüzgâr direnci sayesinde Güneydoğu Anadolu Bölgesi'nde yaygın olan hafif-orta şiddetteki rüzgârlarda stabilite korunmuştur.
- 11.1V 3500mAh batarya ile 35 dakikalık uçuş süresi, geniş alan taramaları için yeterli operasyonel esneklik sağlamıştır.

### 4. Otonom Özellikler:

- GPS/GLONASS/BDS üçlü konumlandırma sistemi, hassas navigasyon ve sabit nokta çevresinde dairesel uçuş imkânı sunmuştur.
- Engelden kaçınma özelliği karmaşık arazi yapısına sahip bölgelerde güvenliği artırmıştır.

Çalışmamızda hayvanların doğal davranışlarını bozmamak ve etik çerçevede veri toplamak için aşağıdaki protokoller uygulanmıştır:

#### 1. Uçuş Parametreleri:

- İrtifa: Hedef hayvan türüne ve arazi koşullarına bağlı olarak 50cm-30 metre arasında dinamik olarak ayarlanmıştır. Örneğin, sığırlar gibi daha az hareketli türler için düşük 50cm -10m (nadiren 30m) keçiler gibi hassas türler için yüksek irtifa (5-30 m) tercih edilmiştir.
- Hız: İHA'nın varsayılan hızı (10 m/s) kullanılarak ani manevralardan kaçınılmış ve hayvanların stres tepkisi minimize edilmiştir.

#### 2. Çekim Stratejisi:

Minimum Mesafe: Etik gereklilikler doğrultusunda sığırlar için 50 cm, keçiler için 10 metre güvenli mesafe korunmuştur.

#### 3. Çevresel ve Zamanlama Koşulları:

- Işık: Veriler, sabah 06:30 – 08:00 (yumuşak ışık), öğlen 12:00 – 14:00 (yüksek kontrast) ve ikindi-akşam arası (16:00 – 18:00) saatlerinde toplanmıştır.
- Bölge: Batman ili Kozluk ilçesindeki hayvanların doğal yaşam alanları olan meralar ve otlaklarda gerçekleştirilen çekimlerde,

İHA'nın termal kamera eksikliği nedeniyle gece operasyonları yapılmamıştır.

#### 4. Veri Tutarlılığı:

- Veri tutarlılığını sağlamak için manuel çekimlerde standart prosedürler uygulanmıştır. Operatörler, önceden tanımlanmış çekim parametrelerini (irtifa, hız, çekim açıları) titizlikle takip ederek veri setinde tutarlılık sağlamıştır. Her uçuş öncesinde, İHA'nın kamera ayarları (örneğin, 4K çözünürlük, 30 FPS) ve çevresel koşullar (örneğin, ışık yoğunluğu) kontrol edilerek en ideal görüntüler alınmaya çalışılmıştır. Ayrıca, aynı sürü senaryoları için birden fazla çekim yapılarak veri tekrarlanabilirliği artırılmış ve operatör kaynaklı hatalar sistematik olarak değerlendirilip düzeltilmiştir.

#### 3.2. Veri Toplama Süreci

Bu çalışmada kullanılan veri seti, Batman ili Kozluk ilçesindeki dağlık ve bozkır alanlarda, hayvanların doğal yaşam ve otlama ortamlarında toplanmıştır. Veriler, 4K çözünürlüklü (3840×2160) İHA ile 30 FPS'de, her biri maksimum 5 dakika süreli 45 video kaydından elde edilmiştir. Çekimler, günün farklı saatlerinde (sabah 06:30-08:00, öğlen 12:00-14:00, ikindi 16:00-18:00) gerçekleştirilmiş, böylece ışık ve çevresel koşullardaki çeşitlilik sağlanmıştır. Çekimlerde, sığır (sığır) ve küçükbaş (keçi) hayvanların İHA'ya tepkileri gözlemlenmiştir. Özellikle keçi sürülerinin yakın çekimlerde (5 m'den az) İHA'ya karşı hassas davranarak kaçtığı tespit edilmiş, bu da ideal görüntülerin toplanmasını zorlaştırmıştır. Bu sorunu çözmek için İHA irtifası optimize edilmiştir: sığırlar için 0,5-30 m, keçiler için 10-30 m aralığında çekimler yapılmıştır. Elde edilen videolardan, özel algoritmalar kullanılarak 1531 sığır ve 131 keçi görüntüsü çıkarılmıştır (Çizelge 3.1). Veri seti, %70 eğitim ve %30 test olacak şekilde bölünmüştür. Buna göre, Keçi sürüsü için 91 eğitim, 40 test görüntüsü elde edilmiştir. Görüntülerde birden fazla sürüleme hareketi mevcut olduğundan keçi sürüsü etiketli örnek sayısı 90 ve 46 olarak oluşmuştur. Sığır görüntülerinde ise eğitim setinde 1071, test setinde ise 460 veri yer almaktadır. Aynı şekilde, bir görüntüde birden fazla sığır yer alabilmesi nedeniyle, eğitim setinde 7648, test setinde ise 3286 sığır örneği bulunmaktadır.

Çizelge 3.1. Veri setinin özellikleri

Veri Seti	Çözünürlük	İHA Yüksekliği	Görüntü Sayısı		Örnek Sayısı	
			Eğitim	Test	Eğitim	Test
<b>Keçi Sürüsü</b>	3840 × 2160	3m – 10m	91	40	90	46
<b>İnek</b>		50 cm – 30m	1071	460	7648	3286

İHA’lardan elde edilen 45 adet 4K video, model eğitimi ve veri setini zenginleştirmek için ideal görüntülere ayrıştırılmıştır. Bu süreçte, video karelerini çıkarmak ve veri setini hazırlamak için çeşitli algoritmalar geliştirilmiştir.

### 3.3. Video Kare Çıkarma Algoritmaları

İlk algoritma (Şekil 3.1), videoların her karesini okuyarak bir klasöre kaydetmiştir. Ancak bu yöntem, fazla veri üretmesi, hayvan olmayan kareler içermesi, yüksek disk alanı kullanımı ve uzun işlem süresi nedeniyle uygun bulunmamıştır.

```

import cv2
vidcap = cv2.VideoCapture('myData/data.mp4')
success, image = vidcap.read()
count = 0
images = "myData/images/"
while success:
    cv2.imwrite(f"{images}/frame{count}.jpg", image)
    success, image = vidcap.read()
    count += 1

```

Şekil 3.1. Videoların her karesini klasöre kaydeden algoritma

İkinci algoritma (Şekil 3.2), hayvan görüntülerinin bulunduğu zaman dilimlerini ayırmaya odaklanmıştır. Video adı ve zaman dilimleri giriş olarak alınmış, yalnızca hayvan görüntülerini içeren bölümler ayrı video dosyaları olarak kaydedilmiştir. Böylece, hayvan olmayan kareler sorunu çözülmüştür.

```

import os
from moviepy.video.io.VideoFileClip import VideoFileClip

def cut_videos(input_paths, output_dir, time_intervals):
    if not os.path.exists(output_dir):
        os.makedirs(output_dir)

    for i, input_path in enumerate(input_paths):
        video = VideoFileClip(input_path)
        start_time, end_time = time_intervals[i]
        video_subclip = video.subclipped(start_time, end_time)

        base_name = os.path.basename(input_path)
        output_path = os.path.join(output_dir, base_name)

        # Eğer aynı isimde dosya varsa numaralandır
        count = 1
        while os.path.exists(output_path):
            name, ext = os.path.splitext(base_name)
            output_path = os.path.join(output_dir, f"{name}_{count}{ext}")
            count += 1

        video_subclip.write_videofile(output_path, codec="libx264")

# Kullanım örneği
input_paths = ["Video/keci.mp4", "Video/keci.mp4"] # Girdi video dosyalarının yolları
output_dir = "Çikti" # Çıktı video dosyalarının kaydedileceği klasör
time_intervals = [("00:00:07", "00:00:24"), ("00:00:36", "00:01:10")] # Her video için başlangıç ve bitiş süreleri (saat:dakika:saniye)

cut_videos(input_paths, output_dir, time_intervals)

```

Şekil 3.2. Hayvan görüntülerinin zaman dilimlerini ayıran algoritma

Üçüncü algoritma (Şekil 3.3), işlenmiş videoların her saniyesinden sabit aralıklarla 5 kare çıkararak “Images” klasörüne kaydetmiştir. Örneğin, 10 saniyelik bir videodan yaklaşık 50 kare elde edilmiştir. Bu yöntem, veri miktarını optimize etmiş ve disk kullanımını azaltmıştır.

```

import cv2

def extract_frames(video_path, output_dir, fps=5):
    # Video dosyasının adını ve uzantısını ayır
    video_name, _ = os.path.splitext(os.path.basename(video_path))
    # Çıktı klasörünü oluştur
    video_output_dir = os.path.join(output_dir, video_name)
    os.makedirs(video_output_dir, exist_ok=True)
    # Videoyu yükle
    vidcap = cv2.VideoCapture(video_path)
    success, image = vidcap.read()
    count = 0
    frame_count = 0
    interval = int(vidcap.get(cv2.CAP_PROP_FPS) / fps)

    print(f"İşlenen video: {video_name}")

    while success:
        if frame_count % interval == 0:
            frame_path = os.path.join(video_output_dir, f"{video_name}_frame_{count:04d}.jpg")
            cv2.imwrite(frame_path, image)
            print(f"Kaydedilen kare: {frame_path}")
            count += 1
        success, image = vidcap.read()
        frame_count += 1
    vidcap.release()
    print(f"{video_name} videosunun işlenmesi tamamlandı.")

def process_videos(input_dir, output_dir, fps=5):
    # Videos klasöründeki tüm video dosyalarını al
    video_files = [f for f in os.listdir(input_dir) if f.endswith(('.mp4', '.avi', '.mov', '.mkv'))]
    total_videos = len(video_files)
    processed_videos = 0

    # Her video dosyasını işleyerek karelere ayır
    for video_file in video_files:
        video_path = os.path.join(input_dir, video_file)
        extract_frames(video_path, output_dir, fps)
        processed_videos += 1
    print(f"Toplam video: {total_videos}, İşlenen video: {processed_videos}")

# Kullanım örneği
input_dir = "Videos" # Girdi video dosyalarının bulunduğu klasör
output_dir = "Images" # Çıktı resim dosyalarının kaydedileceği klasör
process_videos(input_dir, output_dir)

```

Şekil 3.3. Her saniyeden sabit aralıklarla kare çıkaran algoritma

Ancak, otomatik kare çıkarma algoritmalarının ideal görüntüleri seçmede yetersiz olduğu fark edilmiştir. Bu nedenle, kısmen manuel bir yöntem geliştirilmiştir. “Times” adında bir klasör oluşturulmuş ve her video için ideal karelerin zaman dilimlerini içeren metin dosyaları hazırlanmıştır (Şekil 3.4). Videolar taranarak, hayvanların net görüldüğü, bulanık olmayan ve uygun açılı karelerin zaman dilimleri kaydedilmiştir.

Ad	Değiştirme tarihi	Tür	Boyut	
0070101-006059.txt	1.01.2025 17:45	Text Document	1 KB	00:01
20240802-185650.txt	1.01.2025 19:37	Text Document	1 KB	00:06
20240805-175338.txt	1.01.2025 19:09	Text Document	1 KB	00:12
20240805-179932.txt	1.01.2025 19:21	Text Document	1 KB	00:14
20240805-180308.txt	2.01.2025 09:30	Text Document	1 KB	00:23
20240805-180509.txt	2.01.2025 09:30	Text Document	1 KB	00:25
20240805-181009.txt	2.01.2025 09:12	Text Document	1 KB	00:34
20240807-063423.txt	2.01.2025 09:52	Text Document	1 KB	00:36
20240807-065019.txt	2.01.2025 10:21	Text Document	1 KB	00:38
20240807-063019.txt	2.01.2025 09:50	Text Document	1 KB	00:39
20240807-070019.txt	2.01.2025 10:42	Text Document	1 KB	00:41
20240807-070210.txt	2.01.2025 10:46	Text Document	1 KB	00:42
20240808-134538.txt	2.01.2025 10:55	Text Document	1 KB	00:45
20240808-132058.txt	2.01.2025 11:01	Text Document	1 KB	00:59
20240812-185222.txt	2.01.2025 20:33	Text Document	1 KB	01:03
20240812-185559.txt	2.01.2025 20:45	Text Document	1 KB	01:05
20240813-181448.txt	2.01.2025 21:07	Text Document	1 KB	01:06
20240813-184225.txt	2.01.2025 21:34	Text Document	1 KB	01:07
20240815-183136.txt	2.01.2025 21:56	Text Document	1 KB	01:09
20240815-185638.txt	2.01.2025 21:31	Text Document	1 KB	01:14
20240821-181401.txt	2.01.2025 22:33	Text Document	1 KB	01:17
20240821-182541.txt	2.01.2025 22:39	Text Document	1 KB	01:19
20240821-182750.txt	2.01.2025 22:20	Text Document	1 KB	01:21
20240821-183412.txt	3.01.2025 00:19	Text Document	1 KB	01:25
20240821-184847.txt	3.01.2025 00:26	Text Document	1 KB	01:27
20240821-185326.txt	2.01.2025 23:07	Text Document	1 KB	01:28
20240830-165403.txt	2.01.2025 23:25	Text Document	1 KB	01:29
20240830-170123.txt	3.01.2025 01:13	Text Document	1 KB	01:29
20240830-171520.txt	3.01.2025 00:50	Text Document	1 KB	01:32
20240901-171636.txt	3.01.2025 01:08	Text Document	1 KB	01:36

Şekil 3.4. İdeal karelerin zaman dilimlerini içeren metin dosyalarının organizasyonu

Son olarak, beşinci algoritma (Şekil 3.5) bu metin dosyalarını okuyarak belirtilen zaman dilimlerindeki kareleri çıkarmış ve “Images” klasörüne kaydetmiştir.

Bu süreç sonucunda, keçi sürüleri için 131, sığır hayvanlar için 1531 görüntüden oluşan bir veri seti elde edilmiştir. Veri seti, %70 eğitim ve %30 test olacak şekilde bölünmüştür.

```

[ ]: import os
import cv2

def extract_frames(video_path, output_dir, times, fps=5):
    # Video dosyasının adını ve uzantısını ayır
    video_name, _ = os.path.splitext(os.path.basename(video_path))

    # Çıktı klasörünü oluştur
    video_output_dir = os.path.join(output_dir, video_name)
    os.makedirs(video_output_dir, exist_ok=True)

    # Videoyu yükle
    vidcap = cv2.VideoCapture(video_path)
    success, image = vidcap.read()

    print(f"İşlenen video: {video_name}")

    for time_str in times:
        if time_str.strip(): # Boş satırları atla
            minutes, seconds = map(int, time_str.split(':'))
            total_seconds = minutes * 60 + seconds
            for i in range(fps):
                time = total_seconds + i / fps
                vidcap.set(cv2.CAP_PROP_POS_MSEC, time * 1000)
                success, image = vidcap.read()
                if success:
                    frame_path = os.path.join(video_output_dir, f"{video_name}_sec_{total_seconds:04d}_frame_{i:02d}.jpg")
                    cv2.imwrite(frame_path, image)
                    print(f"Kaydedilen kare: {frame_path}")
                    # Kaydedilen resmi ekrana yazdır
                    #cv2.imshow(f"{video_name}_sec_{total_seconds:04d}_frame_{i:02d}", image)
                    cv2.waitKey(500) # 500 milisaniye bekle
                    cv2.destroyAllWindows()
                else:
                    print(f"Frame alınamadı: {video_name} saniye {total_seconds}, frame {i}")

    vidcap.release()
    print(f"{video_name} videosunun işlenmesi tamamlandı.")
def process_videos(input_dir, output_dir, fps=5):
    # Girdi klasöründeki tüm txt dosyalarını al
    txt_files = [f for f in os.listdir(input_dir) if f.endswith('.txt')]

    for txt_file in txt_files:
        video_name = os.path.splitext(txt_file)[0]
        video_path = os.path.join(input_dir, f"{video_name}.mp4")

        if os.path.exists(video_path):
            # txt dosyasını oku ve zamanları al
            with open(os.path.join(input_dir, txt_file), 'r') as file:
                times = [line.strip() for line in file.readlines()]

            # Videoyu işleyerek karelere ayır
            extract_frames(video_path, output_dir, times, fps)
        else:
            print(f"Video dosyası bulunamadı: {video_path}")

# Kullanım örneği
input_dir = "tmpVideos" # Girdi video ve txt dosyalarının bulunduğu klasör
output_dir = "tmpImages" # Çıktı resim dosyalarının kaydedileceği klasör
process_videos(input_dir, output_dir)

```

Şekil 3.5. Times klasöründeki zaman dilimlerine göre kare çıkararak algoritma

### 3.4. Anotasyon ve Etiketleme

Anotasyon ve etiketleme, ham görüntü verilerine anlamlı bilgiler ekleyerek makine öğrenimi modellerinin eğitimi için gerekli olan sınıflandırma ve yapılandırılmış veri oluşturma sürecini kapsar. Bu çalışmada, İHA'lerden elde edilen görüntüler, sığır (sığır) ve küçükbaş (keçi sürüsü) hayvanları sınıflandırmak amacıyla etiketlenmiştir. Anotasyon süreci, nesne algılama modellerinin (örneğin, YOLO) doğru bir şekilde eğitilmesi için sınırlayıcı kutular (bounding boxes) çizilerek gerçekleştirilmiştir (Jiang vd., 2020).

Veri seti, daha önce belirtildiği üzere, 1531 sığır ve 131 keçi görüntüsünden oluşmaktadır. Bu görüntülerin tamamı, nesne algılama için YOLO formatında

etiketlenmiştir. Etiketleme işlemi, sırasıyla manuel, yarı otomatik ve otomatik araçlar kullanılarak tamamlanmıştır.

**CVAT (Computer Vision Annotation Tool):** CVAT (CVAT.ai), interaktif bir arayüz sunarak etiketleme sürecini hızlandırmış ve birden fazla görüntünün eş zamanlı anotasyonuna olanak sağlamıştır. CVAT ile etiketlenen görüntüler, YOLO formatında kaydedilmiştir.

**Özel AI Destekli Araç:** Etiketleme sürecini daha da optimize etmek amacıyla, çalışmaya özgü bir yapay zekâ destekli etiketleme aracı kullanılmıştır (roboflow.com). Bu araç, önceden eğitilmiş bir YOLOv5 (Gong vd., 2022) modelini kullanarak görüntülerdeki hayvanları otomatik olarak tespit etmiş ve sınırlayıcı kutular önermiştir. Operatörler, bu önerileri manuel olarak kontrol ederek etiketlerin doğruluğunu doğrulamıştır. Bu yöntem, etiketleme süresini yaklaşık %60 oranında azaltmış ve veri setinin hazırlanmasını önemli ölçüde hızlandırmıştır.

Etiketleme sürecinde kalite kontrolü, her görüntünün en az iki operatör tarafından bağımsız olarak incelenmesiyle sağlanmıştır. Hatalı veya belirsiz etiketler düzeltilmiş, bulanık veya hayvanların net görünmediği görüntüler veri setinden çıkarılmıştır. Sonuç olarak, yüksek kaliteli ve doğru etiketlenmiş bir veri seti elde edilmiştir, bu da modellerin eğitiminde yüksek doğruluk oranlarına katkı sağlamıştır.

### 3.5. Veri Arttırma Teknikleri

Veri artırma, derin öğrenme modellerinin genelleştirme yeteneğini artırmak ve sınıf dengesizliği gibi sorunları hafifletmek için kritik bir yöntemdir (Bhatkalkar vd., 2020). Geleneksel veri artırma teknikleri (örneğin, görüntü çevirme veya parlaklık ayarı), nesne tespiti gibi karmaşık görevlerde yetersiz kalmaktadır, çünkü nesne tespiti hem sınıf hem de konum bilgisi gerektirir (Ramachandran & Sangaiah, 2021; Wu vd., 2022). Bu çalışmada, CutOut (DeVries & Taylor, 2017), CutMix (Yun vd., 2019), MixUp (H. Zhang vd., 2018) ve Mosaic (H. Zhang vd., 2024; B. Zhao vd., 2021) gibi modern veri artırma teknikleri uygulanarak, örtüşmeler, ölçek farklılıkları ve karmaşık arka planlar simüle edilmiş ve modellerin sağlamlığı artırılmıştır. Çizelge 3.2, geleneksel ve modern veri artırma tekniklerini karşılaştırmaktadır.

Çizelge 3.2. Geleneksel ve modern veri artırma tekniklerinin özellik karşılaştırması

Özellik / Kriter	Geleneksel Teknikler	Modern Teknikler
<b>Tanım</b>	Basit görüntü işleme ile veri çeşitlendirme	Öğrenen/üreten sistemlerle yapay veri üretimi
<b>Yöntemler</b>	Döndürme, kırpma, çevirme, yakınlaştırma, gürültü ekleme	CutOut, CutMix, MixUp, Mosaic
<b>Uygulama Zorluğu</b>	Düşük – Kolay uygulanabilir	Orta-Yüksek – Model eğitimi ve hesaplama gerektirir
<b>Gerçekçiliği</b>	Gerçek görüntülere oldukça benzer	Bazen daha gerçekçi, bazen abartılı/sentetik olabilir
<b>Veri Çeşitliliği Katkısı</b>	Orta düzey	Yüksek – Özellikle az veri durumlarında avantajlı
<b>Model Performansına Etkisi</b>	Ortalama %5–10 iyileştirme	%10–25 arası doğruluk artışı gözlemlenebilir
<b>Overfitting Azaltma Yetisi</b>	Kısmen	Güçlü – Daha zengin ve karmaşık veri sağlar
<b>Kullanım Alanları</b>	Görüntü sınıflandırma, basit nesne tespiti	Karmaşık görevler (NLP, segmentasyon, drone görüntüleri vb.)
<b>İHA Tabanlı Hayvancılıkta Kullanım</b>	Sınırlı – Görseli bozma riski var	Daha etkili – Çeşitlilik ve düşük veri sorunu çözülür

### 3.5.1. CutOut Veri Artırma Tekniği

CutOut tekniği, görüntünün rastgele bölümlerini maskeleyerek örtüşme senaryolarını simüle eder (Şekil 3.6). Örneğin, bir sığırın %20'lik bir bölümü rastgele maskelenerek modelin kalan özelliklere odaklanması sağlanmıştır. Bu yöntem, modelin eksik bilgiye rağmen doğru tahminler yapabilmesini sağlar ve İHA görüntülerinde hayvanların ağaçlar arasında kaldığı durumlar gibi gerçek dünya senaryolarında sağlamlığı artırır



Şekil 3.6. Sığır görüntülerine CutOut uygulanması

### 3.5.2. CutMix Veri Arttırma Tekniği

CutMix, iki farklı görüntüyü ve onların sınırlayıcı kutularını birleştirir (Şekil 3.7). Örneğin, bir sığır görüntüsünün bir kısmı başka bir sığır görüntüsüne entegre edilmiş ve sınırlayıcı kutular buna göre ayarlanmıştır. Bu teknik, modeli karmaşık görsel bağlamlara maruz bırakarak aşırı uyumu önler ve farklı arka planlardan özellik çıkarımını teşvik eder. CutMix, özellikle İHA görüntülerinde değişen arazi koşullarıyla başa çıkmada etkili olmuştur.



Şekil 3.7. Sığır görüntülerine CutMix uygulanması

### 3.5.3. MixUp Veri Arttırma Tekniği

MixUp, iki görüntüyü ve sınırlayıcı kutularını doğrusal olarak harmanlar (Şekil 3.8). Örneğin, iki sığır görüntüsü 0,6 : 0,4 oranında birleştirilerek yeni bir görüntü oluşturulmuştur. Bu yöntem, modelin gürültülü verilere karşı duyarlılığını azaltır ve yumuşak etiketler tahmin etmesini sağlar.



Şekil 3.8. Sığır görüntülerine MixUp uygulanması

### 3.5.4. Mosaic Veri Arttırma Tekniđi

Mosaic tekniđi, drt farklı grntyy tek bir grntde birleřtirir ve veri eřitliliđini artırır (řekil 3.9). rneđin, sıđır grntleri farklı leklerde birleřtirilerek modelin kk nesnelere ve karmařık arka planları tanınması teřvik edilmiřtir. Bu ynem, İHA tabanlı uzaktan algılamada kk nesne tespitini iyileřtirmeyi ve eđitim srecini hızlandırmayı amalar.

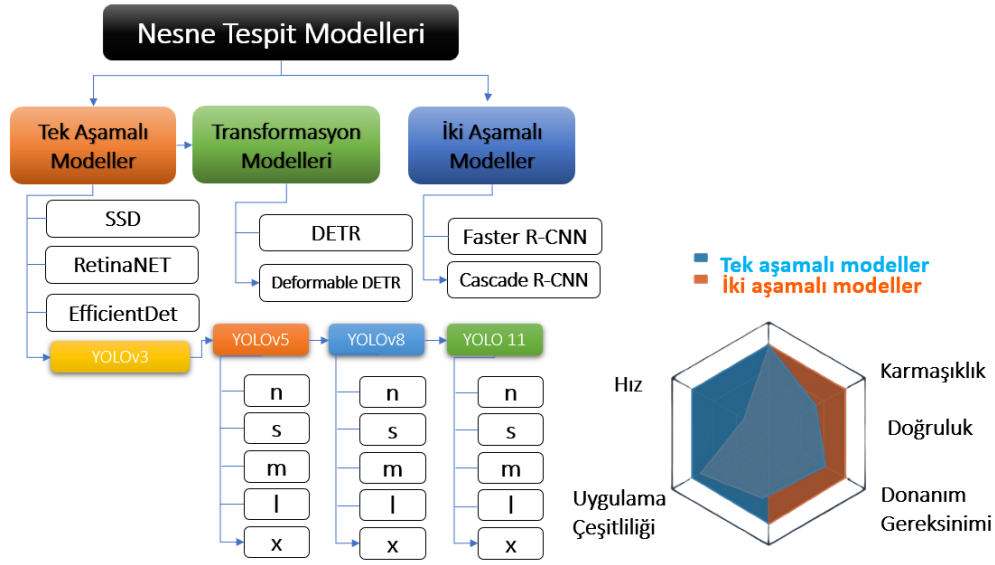


řekil 3.9. Sıđır grntlerine Mosaic uygulanması

Bu teknikler, veri setini 1531 sıđır ve 131 kei grntsy yaklaşık 4 katına ıkartmıř ve zellikle kei grntlerinin az olduđu sınıf dengesizliđi sorununu hafifletmiřtir. Veri artırma iřlemleri, nesne algılama modellerinin genelleme kapasitesini artırarak daha sađlam ve gvenilir sonular elde edilmesini sađlamıřtır (Wong vd., 2016).

### 3.6. Nesne Tespit Modelleri

Bu alıřmada, insansız hava aracı (İHA) grntlerinden elde edilen hayvan srsy verilerinin analizi iin hem tek ařamalı hem de iki ařamalı nesne tespiti temelli derin đrenme modelleri kullanılmıřtır. alıřmamızda hem tek ařamalı hem de iki ařamalı nesne tespit modelleri deđerlendirilmiřtir. Tek ařamalı modeller daha hızlı eđitim ve test sađlarken, iki ařamalı modeller daha yksek dođruluk vermektedir (řekil 3.10).



Şekil 3.10. Tek ve iki aşamalı nesne tespit modellerinin karşılaştırılması

### 3.6.1. Tek Aşamalı Nesne Tespit Modelleri

Tek aşamalı modeller, nesne sınıfını ve sınırlayıcı kutuları tek bir adımda tahmin ederek düşük gecikmeli çıkarım süreleri sağlar. Bu özellik, İHA tabanlı gerçek zamanlı hayvan tespit uygulamaları için uygundur. Çalışmada kullanılan tek aşamalı modeller şunlardır:

- **SSD (Single Shot MultiBox Detector):** Çoklu özellik haritaları ile farklı ölçeklerde nesne tespiti yapar. Hızlıdır, ancak karmaşık arka planlarda doğruluk açısından sınırlamaları vardır (W. Liu vd., 2016).
- **RetinaNet:** Focal loss fonksiyonu ile küçük ve nadir nesnelerin tespitinde etkilidir (Lin vd., 2020).
- **YOLOv3:** Tamamen evrimsel mimarisi ile hızlı ve verimlidir, ancak küçük nesnelere sınırlamalar gösterir (Redmon & Farhadi, 2018).
- **EfficientDet:** Bi-directional Feature Pyramid Network (BiFPN) ile hız ve doğruluk arasında denge sunar.
- **YOLOv5:** Hafif mimarisi ve esnek konfigürasyonları (nano, small, medium, large) ile farklı donanımlara uyar. Çeşitli varyantları çalışmamızda ayrı ayrı eğitilmiştir (Gong vd., 2022).
- **YOLOv8:** Gelişmiş özellik haritası füzyonu ile yüksek doğruluk sağlar. Çeşitli varyantları (n, s, m, l) çalışmamızda ayrı ayrı eğitilmiştir (C.-Y. Wang vd., 2023).

- **YOLO11**: YOLO serisinin en yeni sürümü olarak, dikkat mekanizmaları ve otomatik hiperparametre optimizasyonu ile öne çıkar. İHA görüntülerinde küçük ve yoğun nesnelerin tespitinde avantaj sağlar. Çeşitli varyantları (n, s, m, l) çalışmamızda ayrı ayrı eğitilmiştir (X. Wang vd., 2023).

### 3.6.2. İki Aşamalı Nesne Tespit Modelleri

İki aşamalı modeller, önce bölge önerileri üretir, ardından sınıflandırır ve hizalar. Daha yüksek doğruluk sunmalarına rağmen, çıkarım süreleri uzundur. Çalışmada kullanılan iki aşamalı modeller şunlardır:

- **Faster R-CNN**: Region Proposal Network (RPN) ile verimli bölge önerileri sunar, ancak gerçek zamanlı uygulamalarda yavaştır (Ren vd., 2015).
- **Cascade R-CNN**: Çok aşamalı tespit ile bölge önerilerini iyileştirir ve yüksek doğruluk sağlar (Cai & Vasconcelos, 2018).
- **DETR (Detection Transformer)**: Öz-dikkat mekanizması ile nesne ilişkilerini çıkarır, ancak eğitim süreci karmaşıktır (Zoph vd., 2018).
- **Deformable DETR**: Deformable dikkat modülleri ile küçük ve tıkanık nesnelere iyileşme sağlar (Zhu vd., 2021).

### 3.7. Eğitim Ortamı ve Donanım

Model eğitimi, Python 3.12.5 ve PyTorch 2.5.0+cu124 kütüphanesi kullanılarak gerçekleştirilmiştir. Eğitim süreci, CUDA destekli NVIDIA GeForce RTX 4060 Laptop GPU (8 GB VRAM) üzerinde yürütülmüştür. Eğitimler, 50 epoch boyunca,  $640 \times 640$  piksel görüntü boyutu ve 16'lık yığınlar (batch size) ile gerçekleştirilmiştir. Optimizasyon algoritması olarak AdamW (lr = 0.002, momentum = 0.9) kullanılmış, ağırlık düzenlemesi (weight\_decay = 0.0005) ile optimizasyon yapılmıştır [50].

### 3.8. Performans Ölçütleri

Nesne tespiti, bilgisayarla görü alanında görüntü veya video içerisindeki nesnelerin hem yerini (konumunu) hem de sınıf etiketini doğru bir şekilde belirlemeyi amaçlayan temel görevlerden biridir. Nesne tespit modellerinde sadece nesnelere bulma yeteneğini değil, aynı zamanda bu nesnelere ne kadar doğru sınıflandırdığı ve kaç tanesini kaçırmadan tespit ettiğini anlamakta kritik bir öneme sahiptir. Bu bağlamda en yaygın kullanılan ölçütler Duyarlılık (Recall) ve mean

Average Precision (mAP)'dır. Bu metrikler, modelin güçlü ve zayıf yönlerini belirlemek ve farklı model yapılarını karşılaştırmak açısından önemli bir role sahiptir.

- *Recall*, modelin gerçekten var olan nesnelere ne kadar iyi ulaşabildiğini ölçer. Yani, tüm gerçek nesnelere arasından kaç tanesinin doğru şekilde tespit edildiğini gösterir (Eşitlik (3.1)).

$$Recall = \frac{TP}{FN + TP} \quad (3.1)$$

Bu formülde TP (Gerçek Pozitif) değeri doğru şekilde tespit edilen nesnelere sayısını, FN (Yanlış Negatif) değeri ise modelin kaçırdığı (tespit edemediği) gerçek nesnelere sayısınıdır.

- *mAP* ise, nesne tespitinde hem Doğruluk (Precision) hem de Duyarlılık (Recall) dikkate alınarak hesaplanan bir performans ölçütüdür. Her sınıf için Average Precision (AP) değeri hesaplanır ve tüm sınıfların ortalaması alınarak mAP bulunur (Eşitlik (3.2)).

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (3.2)$$

Bu formülde N toplam sınıf sayısını,  $AP_i$  ise i. Sınıfın ortalama kesinlik (Average Precision) değeridir. mAP değerinin ölçeklendirmesinde iki adet skala kullanılır:

- *mAP@50*: Intersection over Union (IoU) eşik değeri %50 alınarak hesaplanan mAP değerini belirtir.
- *mAP@[.5:.95]*: IoU değeri %50'den %95'e kadar (0.5, 0.55, ..., 0.95) adım adım hesaplanır ve ortalaması alınır.

Bu metrikler modelin performans değerlendirmesinde yaygın bir şekilde kullanılmaktadır.

## 4. BULGULAR VE TARTIŞMA

Çalışmamızda üç eğitim stratejisi geliştirilmiştir:

- Çalışmanın ilk aşamasında tüm modeller kullanılarak orijinal veri setlerinde eğitim ve testler gerçekleştirilmiştir. Bu sayede modellerin sınırlı verideki performansını incelemek mümkün olmuştur.
- Sonraki aşamada ise eğitim sadece arttırılmış verilerle gerçekleştirilmiştir. Bu sayede modern veri arttırma tekniklerinin tek başına sağladığı katkı sorgulanmıştır.
- Son aşamada ise modeller, orijinal + arttırılmış veri stratejisiyle eğitilmiştir. Böylece hem verilerin sayılarının fazlaşmasının hem de tekniklerin yaptığı katkılar bütüncül bir şekilde değerlendirilebilmiştir.

### 4.1. Temel Modellerin Orijinal Verilerde Performans Analizi

Temel modeller, sığır ve keçi Sürüsü veri setlerinde farklı performanslar sergilemiştir. Sığır veri seti, 1531 görüntüden oluşmakta olup daha geniş, çeşitlilik açısından zengin ve çoğunlukla düşük irtifadan (50 cm-5 m, nadiren 30 m) çekilmiş net görüntüler içermektedir. Buna karşın, keçi sürüsü veri seti yalnızca 131 görüntüden oluşmakta, yüksek irtifadan (10-30 m) çekilen görüntüler nedeniyle daha düşük çözünürlüklü ve karmaşık arka planlara sahip bir yapı sergilemektedir. Bu yapısal farklılıklar, modellerin performansını doğrudan etkilemiş; özellikle küçük nesne ölçeği ve örtüşme sorunları, keçi Sürüsü veri setinde tespit doğruluğunu sınırlayan temel faktörler olarak öne çıkmıştır.

#### 4.1.1. Sığır Veri Seti Üzerinde Temel Modellerin Performansı

Sığır veri setinde, temel modeller genellikle yüksek recall ve mAP değerleri elde etmiştir. Özellikle iki aşamalı modeller (örneğin, Cascade R-CNN, Faster R-CNN) ve Focal Loss tabanlı RetinaNET, yüksek doğruluklarıyla dikkat çekmiştir. Ancak, tek aşamalı modeller arasında SSD'nin performansı daha düşük kalmıştır. Çizelge 4.1, sığır veri setindeki sonuçları özetlemektedir.

Çizelge 4.1. Temel modellerin sığır veri seti üzerindeki performans sonuçları

<b>Model</b>	<b>Recall</b>	<b>mAP</b>
<b>SSD</b>	0,830	0,701
<b>RetinaNET</b>	0,971	0,886
<b>YOLOv3</b>	0,941	0,886
<b>DETR</b>	0,957	0,891
<b>Faster R-CNN</b>	0,938	0,892
<b>Deformable DETR</b>	0,918	0,894
<b>EfficientDet</b>	0,900	0,896
<b>Cascade R-CNN</b>	0,914	0,898

Çizelge 4.1'e göre, Cascade R-CNN, Sığır veri setinde en yüksek mAP değerine (0,898) ulaşırken, RetinaNET en yüksek recall değerini (0,971) elde etmiştir. SSD, hem recall (0,830) hem de mAP (0,701) açısından en düşük performansı göstermiştir. Bu durum, SSD'nin karmaşık arka planlarda ve küçük nesnelere sınırlamalarını ortaya koymaktadır. YOLOv3 ve DETR, sırasıyla 0,886 ve 0,891 mAP değerleriyle dengeli bir performans sergilemiştir. EfficientDet ve Deformable DETR, sırasıyla 0,896 ve 0,894 mAP değerleriyle Cascade R-CNN'e yakın sonuçlar vermiştir, ancak recall açısından biraz daha düşük kalmıştır.

#### **4.1.2. Keçi Sürüsü Veri Seti Üzerinde Temel Modellerin Performansı**

Keçi sürüsü veri setinde, modellerin performansı sığır tespitine kıyasla daha düşük olmuştur. Bunun temel nedeni, veri setinin küçük boyutu ve keçilerin daha küçük ve yoğun nesnelere görüntülerde yer almasıdır. Çizelge 4.2, keçi sürüsü veri setindeki sonuçları özetlemektedir.

Çizelge 4.2. Temel modellerin keçi sürüsü veri seti üzerindeki performans sonuçları

<b>Model</b>	<b>Recall</b>	<b>mAP</b>
<b>SSD</b>	0,696	0,566
<b>RetinaNET</b>	0,848	0,570
<b>YOLOv3</b>	0,696	0,585
<b>DETR</b>	0,783	0,586
<b>Faster R-CNN</b>	0,804	0,603
<b>Deformable DETR</b>	0,783	0,607
<b>EfficientDet</b>	0,696	0,616
<b>Cascade R-CNN</b>	0,739	0,674

Çizelge 4.2’den görüldüğü üzere, Cascade R-CNN keçi sürüsü veri setinde en yüksek mAP değerine (0,674) ulaşırken, RetinaNET en yüksek recall değerini (0,848) elde etmiştir. SSD ve YOLOv3, sırasıyla 0,66 ve 0,585 mAP değerleriyle en düşük performansı göstermiştir. Deformable DETR ve EfficientDet, orta seviyede bir performans sergileyerek sırasıyla 0,607 ve 0,616 mAP değerlerine ulaşmıştır. Keçi sürüsü veri setindeki düşük performans, veri setinin sınırlı boyutu ve keçilerin küçük nesne boyutları nedeniyle ortaya çıkmıştır.

#### **4.2. YOLO Varyantlarının Orijinal Verilerde Performans Analizi**

YOLO serisi (YOLOv5, YOLOv8, YOLO11), gerçek zamanlı uygulamalar için optimize edilmiş tek aşamalı modeller olarak çalışmada test edilmiştir. Nano, small, medium ve large varyantları, hesaplama karmaşıklığı ve doğruluk arasında farklı dengeler sunar. Aşağıda, sığır ve keçi sürüsü veri setlerindeki performansları ayrı ayrı analiz edilmiştir.

#### 4.2.1. Sığır Veri Seti Üzerinde YOLO Varyantlarının Performansı

Sığır veri setinde, YOLO varyantları genel olarak yüksek recall ve mAP50 değerleri elde etmiştir. YOLOv8 ve YOLO11, YOLOv5'e kıyasla daha gelişmiş özellik haritası füzyonu ve dikkat mekanizmaları sayesinde daha iyi performans göstermiştir. Çizelge 4.3, sığır veri setindeki YOLO varyantlarının sonuçlarını özetlemektedir.

Çizelge 4.3. YOLO varyantlarının sığır veri seti üzerindeki performans sonuçları

Model	Recall	mAP50	mAP50-95
yolov5n	0,794	0,872	0,591
yolov5s	0,841	0,908	0,635
yolov5su	0,832	0,909	0,633
yolov5m	0,858	0,914	0,655
yolov5mu	0,858	0,914	0,655
yolov5lu	0,862	0,918	0,663
yolov8n	0,814	0,883	0,606
yolov8s	0,841	0,911	0,642
yolov8m	0,857	0,923	0,663
yolov8l	0,860	0,923	0,664
yolo11n	0,785	0,875	0,595
yolo11s	0,847	0,911	0,644
yolo11m	0,862	0,917	0,661
yolo11l	0,846	0,913	0,656

Çizelge 4.3'e göre, YOLOv8m ve YOLOv8l, sığır veri setinde en yüksek mAP50 değerine (0.923) ulaşmıştır. YOLOv5lu ve YOLO11m, 0,862 recall ile en yüksek geri çağırma değerlerini elde etmiştir. YOLOv5n, 0,872 mAP50 ile en düşük performansı göstermiştir, bu da hafif modellerin sığır gibi büyük veri setlerinde sınırlı kaldığını göstermektedir. mAP50-95 değerlerine bakıldığında,

YOLOv8l (0,664) ve YOLOv5lu (0,663) en iyi sonuçları vermiştir, bu da bu modellerin farklı IoU eşiklerinde daha kararlı olduğunu göstermektedir. YOLOv8m, hem recall (0,857) hem de mAP50 (0,923) açısından dengeli bir performans sergileyerek sığır veri setinde öne çıkmıştır.

#### 4.2.2. Keçi Sürüsü Veri Seti Üzerinde YOLO Varyantlarının Performansı

Keçi sürüsü veri setinde, YOLO varyantlarının performansı sığır tespitine kıyasla daha düşük kalmıştır. Bunun temel nedeni yine, keçilerin küçük nesne boyutlarına sahip olması ve veri setinin sınırlı boyutudur. Çizelge 4.4, Keçi Sürüsü veri setindeki YOLO varyantlarının sonuçlarını göstermektedir.

Çizelge 4.4. YOLO varyantlarının keçi sürüsü veri seti üzerindeki performans sonuçları

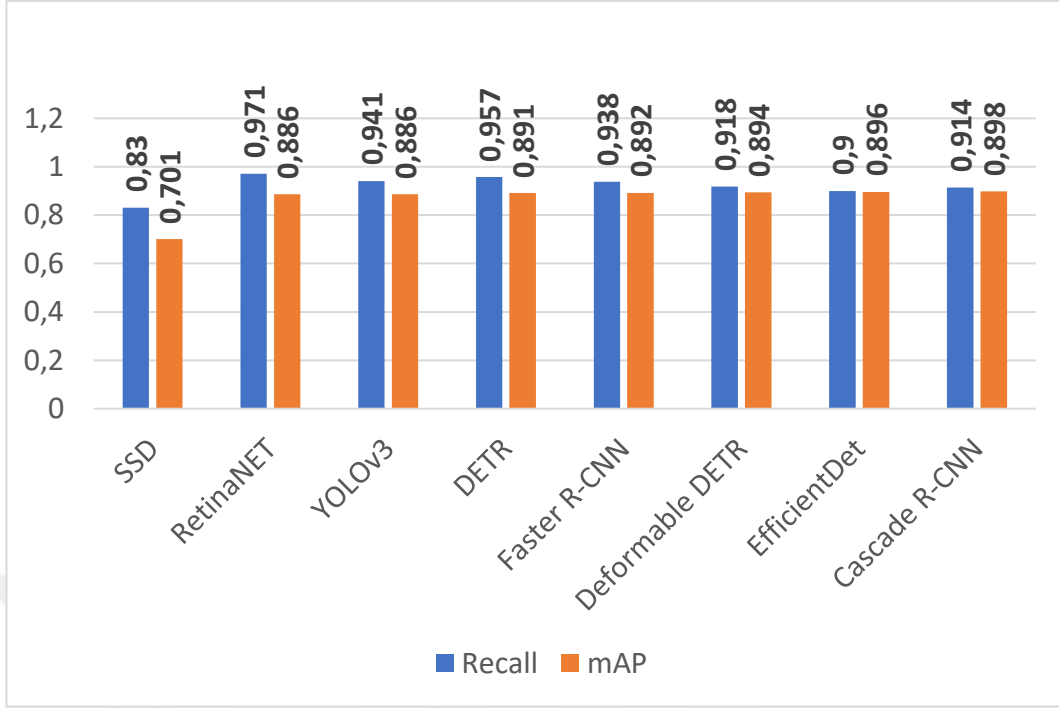
Model	Recall	mAP50	mAP50-95
yolov5n	0,580	0,573	0,352
yolov5s	0,630	0,705	0,356
yolov5m	0,500	0,569	0,330
yolov5l	0,457	0,509	0,279
yolov8n	0,609	0,683	0,368
yolov8s	0,646	0,692	0,332
yolov8m	0,630	0,708	0,360
yolov8l	0,587	0,654	0,392
yolo11n	0,521	0,588	0,337
yolo11s	0,577	0,659	0,407
yolo11m	0,561	0,578	0,341
yolo11l	0,487	0,559	0,287

Çizelge 4.4'ten görüldüğü üzere, YOLOv8m, keçi sürüsü tespitinde en yüksek mAP50 değerine (0,708) ulaşırken, YOLOv8s en yüksek recall değerini (0,646) elde etmiştir. YOLOv5l, 0,509 mAP50 ile en düşük performansı göstermiştir. YOLO11 varyantları, YOLOv8'e kıyasla daha düşük performans sergilemiş, özellikle YOLO11l 0,559 mAP50 ile sınırlı kalmıştır. mAP50-95

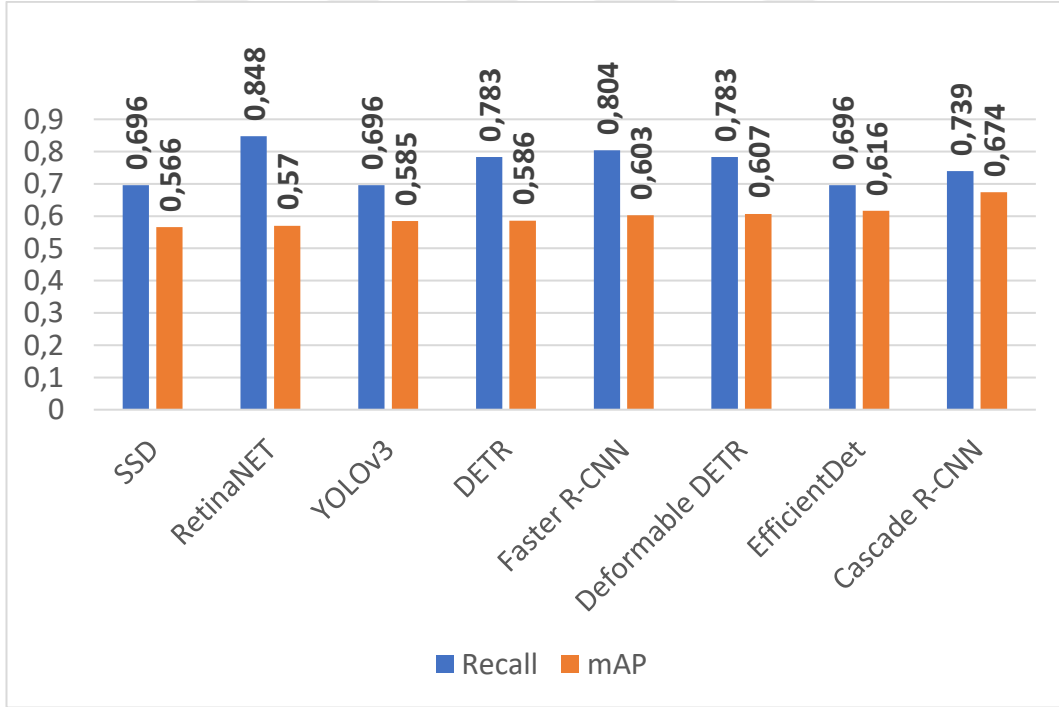
değerleri incelendiğinde, YOLO11s (0,407) ve YOLOv8l (0,392) diğer varyantlara göre daha iyi sonuçlar vermiştir, ancak genel olarak keçi sürüsündeki düşük mAP50-95 değerleri, veri setinin zorluklarını yansıtmaktadır. Bu sonuçlar, YOLOv8'in keçi sürüsü veri setinde daha iyi genelleme yaptığını göstermektedir.

### **4.3. Orijinal Veri Setlerinde Kıyaslanan Modellerin Genel Performans Analizi**

Şekil 4.1 (a) sığır veri setinde RetinaNet ve Cascade R-CNN modellerinin yüksek recall ve mAP değerleriyle diğer modellere kıyasla üstünlük sağladığını göstermektedir. Şekil 4.1 (b) ise keçi sürüsü veri setinde temel modellerin performansını analiz etmektedir. Sonuçlara göre temel modeller arasında Cascade R-CNN'in en yüksek performansı sergilediğini net bir şekilde görülmektedir. Ancak, SSD modeli her iki veri setinde de düşük performansı dikkat çekmiştir. Tüm temel modellerin sığır veri setinde nispeten yüksek mAP ve recall değerleri elde ettiğini, buna karşın keçi sürüsü veri setinde tüm modellerin performansının belirgin bir şekilde düşük kaldığı görülmektedir. Genel olarak, her iki veri setinde Cascade R-CNN tutarlı bir şekilde en iyi performansı sergilerken, SSD en düşük performansı göstermiştir. Bu farklılıklar, veri setlerinin yapısal özelliklerinden (örneğin, görüntü sayısı, irtifa ve nesne ölçeği) kaynaklanmaktadır.

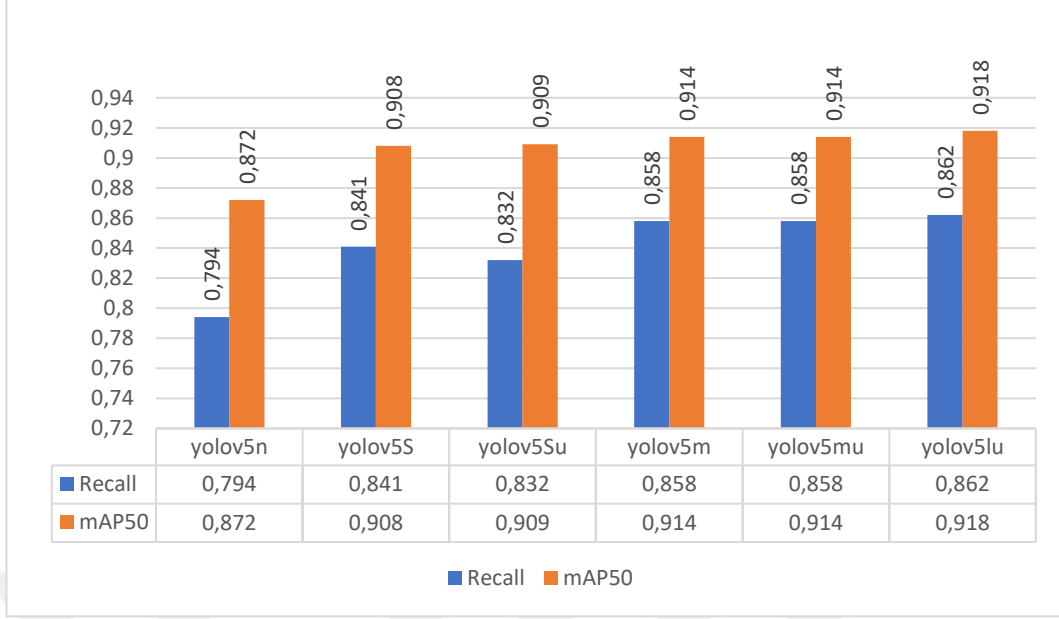


a) Temel modellerin sığır veri setindeki recall ve mAP değer karşılaştırması

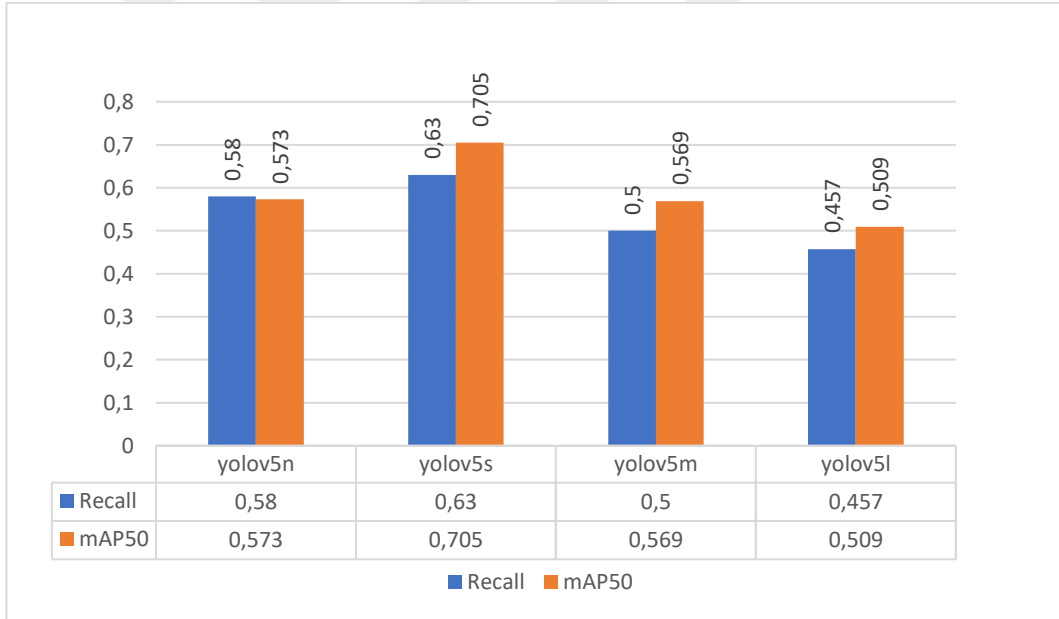


b) Temel modellerin keçi sürüsü veri setindeki recall ve mAP değer karşılaştırması

Şekil 4.1. Temel modellerin sığır ve keçi sürüsü veri setindeki recall ve mAP değer karşılaştırması



a) YOLOv5 varyantlarının sığır veri setindeki recall ve mAP değer karşılaştırması



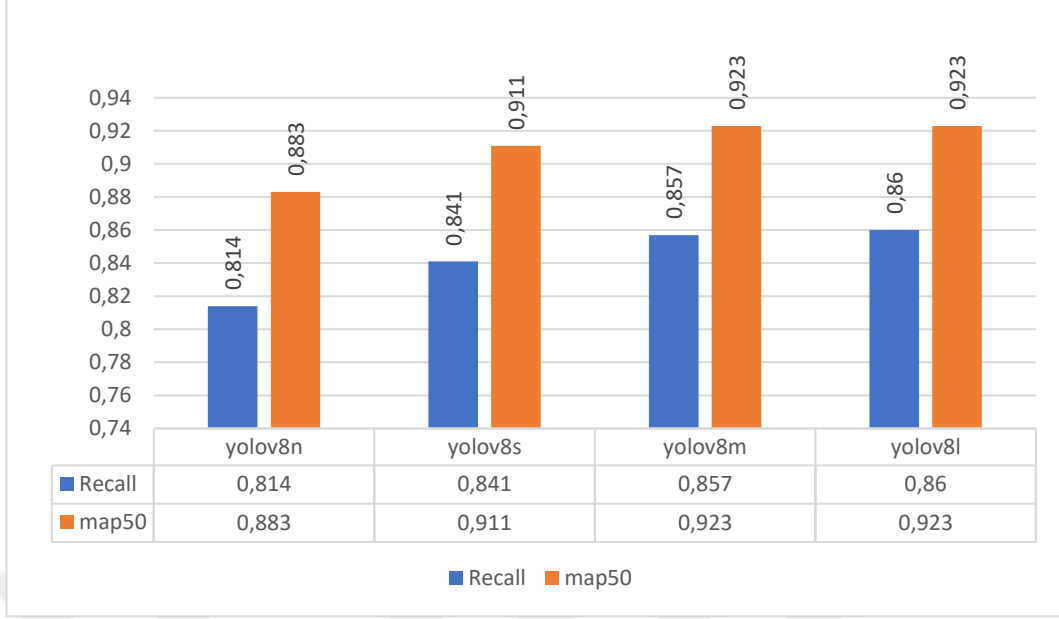
b) YOLOv5 varyantlarının keçi sürüsü veri setindeki recall ve mAP değer karşılaştırması

Şekil 4.2. YOLOv5 varyantlarının sığır ve keçi sürüsü veri setindeki recall ve mAP değer karşılaştırması

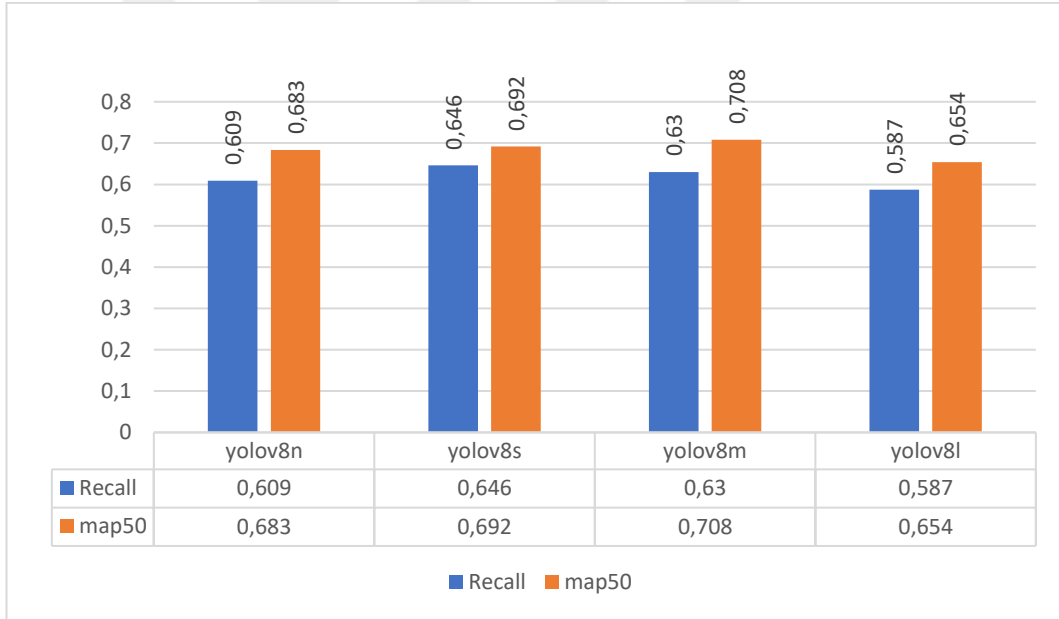
YOLOv5 algoritmasının varyantları (nano, small, medium, large), sığır ve keçi sürüsü veri setlerinde performans açısından Şekil 4.2’de değerlendirilmiştir. Sığır veri setinde (Şekil 4.2 (a)), YOLOv5l varyantı, 0,918 mAP50 ve 0,862 recall değerleriyle en üstün performansı sergilemiştir. Buna karşılık, YOLOv5n varyantı,

0,872 mAP50 ile en düşük performansı göstermiştir. Keçi sürüsü veri setinde ise YOLOv5s varyantı, 0,705 mAP50 ile en yüksek performansı elde ederken, YOLOv5l varyantı 0,509 mAP50 ile en düşük performansı sergilemiştir. Bu sonuçlar, YOLOv5 varyantlarının sığır veri setinde daha yüksek doğruluk ve geri çağırma oranları sunduğunu, ancak keçi sürüsü veri setinde küçük nesne ölçeği (32-64 px) ve yüksek irtifa (10-30 m) koşullarından kaynaklanan zorluklar nedeniyle performansın azaldığını ortaya koymaktadır.

YOLOv8 algoritmasının varyantları (nano, small, medium, large), sığır ve keçi sürüsü veri setlerindeki performansı incelendiğinde ise, sığır veri setinde (Şekil 4.3 (a)), YOLOv8m ve YOLOv8l varyantlarının, 0,923 mAP50 değeriyle en yüksek başarıyı elde ettiği görülmüştür. Öte yandan, YOLOv8n varyantı, 0,883 mAP50 ile en düşük performansı sergilemiştir. Keçi Sürüsü veri setinde ise YOLOv8m varyantı, 0,708 mAP50 ile en iyi sonucu sunarken, YOLOv8l varyantı 0,654 mAP50 ile daha düşük bir performans göstermiştir. Bu bulgular, YOLOv8 varyantlarının genel olarak YOLOv5'e kıyasla daha üstün bir performans sergilediğini ve özellikle YOLOv8m'in her iki veri setinde dengeli bir doğruluk sunduğunu ortaya koymaktadır.

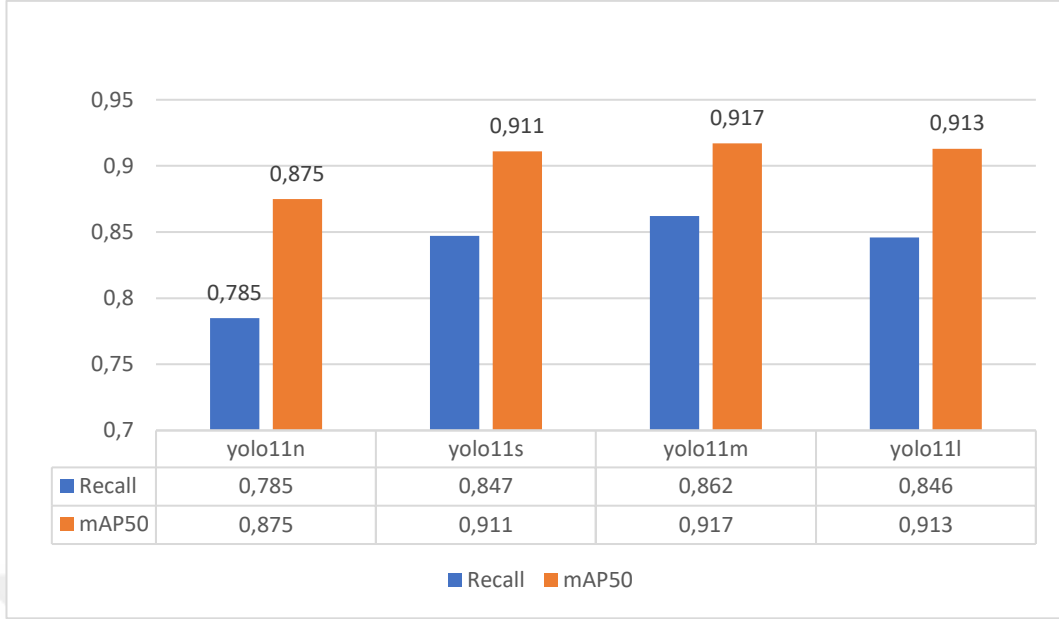


a) YOLOv8 varyantlarının sığır veri setindeki recall ve mAP değer karşılaştırması

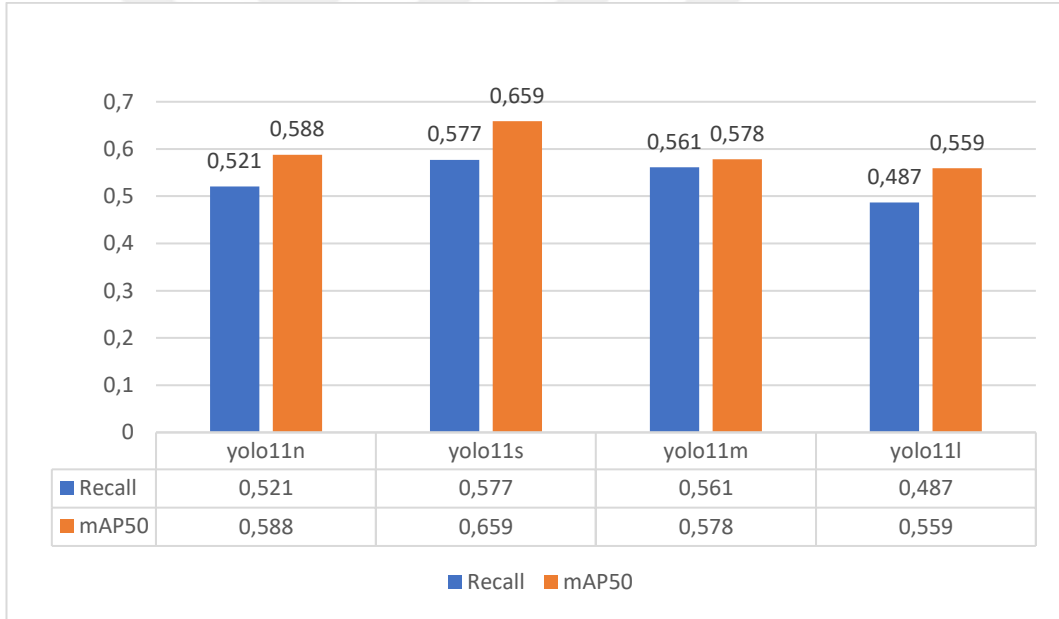


b) YOLOv8 varyantlarının keçi sürüsü veri setindeki recall ve mAP değer karşılaştırması

Şekil 4.3. YOLOv8 varyantlarının sığır ve keçi sürüsü veri setindeki recall ve mAP değer karşılaştırması



a) YOLO11 varyantlarının sığır veri setindeki recall ve mAP değer karşılaştırması



b) YOLO11 varyantlarının keçi sürüsü veri setindeki recall ve mAP değer karşılaştırması

Şekil 4.4. YOLO11 varyantlarının sığır ve keçi sürüsü veri setindeki recall ve mAP değer karşılaştırması

Sığır veri setinde (Şekil 4.4 (a)), YOLO11m varyantı, 0,917 mAP50 ve 0,862 recall değerleriyle en üstün performansı sergilemiştir. Buna karşın, YOLO11n varyantı, 0,875 mAP50 ile en düşük performansı göstermiştir. Keçi Sürüsü veri setinde ise YOLO11s varyantı, 0,659 mAP50 ile en yüksek performansı

sunarken, YOLO111 varyantı 0,559 mAP50 ile en düşük performansı sergilemiştir (Şekil 4.4 (b)). Bu bulgular, YOLO11 varyantlarının YOLOv8'e kıyasla Keçi Sürüsü veri setinde daha düşük bir performans sergilediğini, ancak Sığır veri setinde rekabetçi sonuçlar elde ettiğini ortaya koymaktadır.

YOLO varyantlarının genel performansı değerlendirildiğinde, sığır veri setinde YOLOv8m ve YOLOv8l varyantları, 0,923 mAP50 değeriyle en yüksek performansı sergileyerek diğer varyantlara üstünlük sağlamıştır. Keçi sürüsü veri setinde ise YOLOv8m, 0,708 mAP50 ile en iyi performansı sunmuştur. Öte yandan, YOLOv5l ve YOLO111 varyantları, keçi sürüsü veri setinde sırasıyla 0,509 mAP50 ve 0,559 mAP50 ile en düşük performansı sergilemiş; bu durum, küçük nesne ölçeği ve sınırlı veri seti boyutu karşısında zorluk yaşadıklarını ortaya koymaktadır. Özellikle net olmayan bir görsel kaliteye sahip olması, örtüşme sorunlarının (keçilerin iç içe geçtiği durumlar) yoğunluğu modellerin tespit doğruluğunu olumsuz yönde etkilemiştir. Genel olarak, YOLOv8'in hem sığır hem de keçi sürüsü veri setlerinde diğer modellere kıyasla daha iyi bir genelleme yeteneği sergilediğini ve dengeli bir performans sunduğunu göstermektedir. Bu üstünlük, YOLOv8'in gelişmiş mimari optimizasyonlarının değişken irtifa ve nesne ölçeği dinamiklerine adapte olabildiğini net bir şekilde ortaya koymaktadır.

#### **4.4. Veri Arttırma Tekniklerinin Etkisinin İncelenmesi**

Bu bölümde, Cascade R-CNN ve YOLOv8m modelleri üzerinde uygulanan veri artırma tekniklerinin (CutOut, CutMix, MixUp, Mosaic) sığır ve keçi Sürüsü veri setleri üzerindeki etkileri sistematik bir şekilde analiz edilmiştir. Veri artırma teknikleri, veri setlerinin çeşitliliğini artırmayı ve modellerin genelleme yeteneğini geliştirmeyi hedeflemektedir; bu, özellikle sınırlı boyutlu veya zorlu yapısal özelliklere sahip veri setlerinde model performansını optimize etmek için kritik bir yaklaşımdır. Performans değerlendirmesi, recall ve mAP metrikleri üzerinden, AdamW optimize edici kullanılarak 50 epoch boyunca gerçekleştirilen eğitimler üzerinden gerçekleştirilmiştir.

##### **4.4.1. Cascade R-CNN ile Veri Arttırma Tekniklerinin Performans Analizi**

Sığır veri setinde, veri artırma teknikleri Cascade R-CNN'in performansını farklı oranlarda etkilemiştir. Özellikle, orijinal veri ile veri artırma tekniklerinin kombinasyonları (Orijinal + [Teknik]), yalnızca tek bir tekniğin kullanıldığı

durumlara (Sadece [Teknik]) kıyasla daha yüksek performans göstermiştir. Aşağıdaki Çizelge 4.5, sığır veri setindeki sonuçları özetlemektedir.

Çizelge 4.5. Veri artırma teknikleri ile eğitilen Cascade R-CNN modelinin sığır veri seti üzerindeki performans sonuçları

<b>Veri Artırma</b>	<b>Recall</b>	<b>mAP</b>
<b>Orijinal Veri Seti</b>	0.914	0.898
<b>Sadece CutOut</b>	0.912	0.897
<b>Sadece CutMix</b>	0.900	0.809
<b>Sadece MixUp</b>	0.897	0.806
<b>Sadece Mosaic</b>	0.870	0.807
<b>Orijinal + CutOut</b>	0.904	0.897
<b>Orijinal + CutMix</b>	0.935	0.899
<b>Orijinal + MixUp</b>	0.967	0.899
<b>Orijinal + Mosaic</b>	0.918	0.901

Çizelgede de görüldüğü üzere, Orijinal + MixUp kombinasyonu, sığır veri setinde en yüksek recall değerine (0,967) ulaşarak modelin nesne tespit yeteneğini önemli ölçüde artırmıştır. Buna karşılık, Orijinal + Mosaic kombinasyonu, en yüksek mAP değerini (0,901) elde ederek genelleme performansında üstünlük sağlamıştır. Ancak, Sadece CutMix ve Sadece MixUp teknikleri, sırasıyla 0,809 ve 0,806 mAP ile en düşük performansı sergilemiştir. Orijinal veri setinin performansı (mAP 0,898) ile karşılaştırıldığında, Orijinal + CutMix ve Orijinal + MixUp kombinasyonları hem recall hem de mAP metriklerinde belirgin bir iyileşme sağlamış; bu durum, veri artırma tekniklerinin sığır veri setinde modelin başarımını artırmada etkili olduğunu ortaya koymuştur. Özellikle, orijinal verinin zenginliğini korurken ek veri çeşitliliği sağlayan kombinasyonların, sadece veri artırma ile oluşturulan tekniklere nazaran üstünlük sağladığı gözlemlenmiştir.

Keçi sürüsü veri setinde, veri artırma tekniklerinin Cascade R-CNN üzerindeki etkisi daha karmaşık bir tablo sunmuştur. Bu veri setinin sınırlı boyutu ve düşük çeşitliliği gibi yapısal zorluklar, veri artırma tekniklerinin performans

üzerindeki etkisini kısıtlamıştır. Aşağıdaki Çizelge 4.6, keçi sürüsü veri setindeki sonuçları özetlemektedir:

Çizelge 4.6. Veri artırma teknikleri ile eğitilen Cascade R-CNN modelinin keçi sürüsü veri seti üzerindeki performans sonuçları

<b>Veri Artırma</b>	<b>Recall</b>	<b>mAP</b>
<b>Orijinal Veri Seti</b>	0.739	0.674
<b>Sadece CutOut</b>	0.696	0.625
<b>Sadece CutMix</b>	0.804	0.549
<b>Sadece MixUp</b>	0.696	0.551
<b>Sadece Mosaic</b>	0.804	0.551
<b>Orijinal + CutOut</b>	0.739	0.682
<b>Orijinal + CutMix</b>	0.826	0.698
<b>Orijinal + MixUp</b>	0.935	0.733
<b>Orijinal + Mosaic</b>	0.891	0.751

Çizelgeye göre, Orijinal + Mosaic kombinasyonu, mAP 0,751 ile keçi sürüsü veri setinde en yüksek genelleme performansını sergilerken, Orijinal + MixUp 0.935 ile en iyi recall değerine ulaşmıştır. Buna karşın, Sadece CutMix ve Sadece MixUp kullanımı, sırasıyla mAP 0,549 ve 0,551 ile en düşük performansı göstermiştir. Orijinal veri setinin performansı (mAP 0,674) ile karşılaştırıldığında, orijinal veri ile kombinasyonların (örneğin, Orijinal + MixUp ve Orijinal + Mosaic) modelin performansını önemli ölçüde iyileştirmiştir. Ancak, sadece veri artırma ile elde edilen veri setlerinin düşük performansı, keçi sürüsü veri setinin yapısal zorluklarının bu tekniklerin etkinliğini azalttığını göstermektedir. Bu durum, veri artırma tekniklerinin başarısının veri setinin özelliklerine bağlı olarak değişebileceğini ortaya koymaktadır.

#### **4.4.2. YOLOv8m ile Veri Arttırma Tekniklerinin Performans Analizi**

Bu bölümde, YOLOv8m modelinin sığır ve keçi sürüsü veri setleri üzerindeki performansı, veri artırma teknikleri kullanılarak analiz edilmiştir. CutMix, CutOut, MixUp ve Mosaic teknikleri, orijinal veri setine ek olarak

uygulanmış ve modelin genelleme yeteneği üzerindeki etkileri değerlendirilmiştir. Performans, recall, mAP50 ve mAP50-95 metrikleri üzerinden ölçülmüş; modeller, AdamW optimize edici ile 50 epoch boyunca eğitilmiştir.

Sığır veri setinde, veri artırma teknikleri YOLOv8m'in performansını genel olarak iyileştirmiştir. Özellikle CutMix ve Mosaic teknikleri ile verinin artırılması, orijinal veriye kıyasla daha yüksek recall ve mAP50 değerleri sağlamıştır. Performans sonuçları, aşağıda yer alan Çizelge 4.7 ile özetlenmiştir:

Çizelge 4.7. Veri artırma teknikleri ile eğitilen YOLOv8m modelinin sığır veri seti üzerindeki performans sonuçları

Veri Artırma	Recall	mAP50	mAP50-95
<b>Orijinal + CutMix</b>	0.873	0.929	0.670
<b>Orijinal + CutOut</b>	0.868	0.930	0.669
<b>Orijinal + MixUp</b>	0.856	0.925	0.671
<b>Orijinal + Mosaic</b>	0.873	0.927	0.674

Çizelge 4.7'ye göre, CutOut tekniği en yüksek mAP50 değerine (0,930) ulaşarak modelin nesne tespiti doğruluğunu artırmada öne çıkmıştır. CutMix ve Mosaic, 0,873 recall ile en yüksek geri çağırma oranlarını elde etmiştir. Buna karşılık, MixUp tekniği 0,925 mAP50 ile diğer yöntemlere göre daha düşük bir performans sergilemiştir. Orijinal YOLOv8m modelinin mAP50 değeri (0,923) baz alındığında, veri artırma teknikleri (özellikle CutOut ve CutMix) ile yaklaşık %0,5-0,7'lik bir iyileşme sağlanmıştır. mAP50-95 metrikleri (örneğin, Mosaic için 0,674) ise modelin daha geniş IoU aralıklarında da tutarlı bir performans sergilediğini göstermektedir. Bu sonuçlar, Sığır veri setinin büyük hacmi ve çeşitliliği sayesinde veri artırma tekniklerinin genelleme yeteneğini artırdığını ortaya koymaktadır.

Keçi sürüsü veri setinde ise veri artırma teknikleri, YOLOv8m'in performansını iyileştirmede sınırlı bir etki göstermiştir. Performans sonuçları, Çizelge 4.8 ile özetlenmiştir:

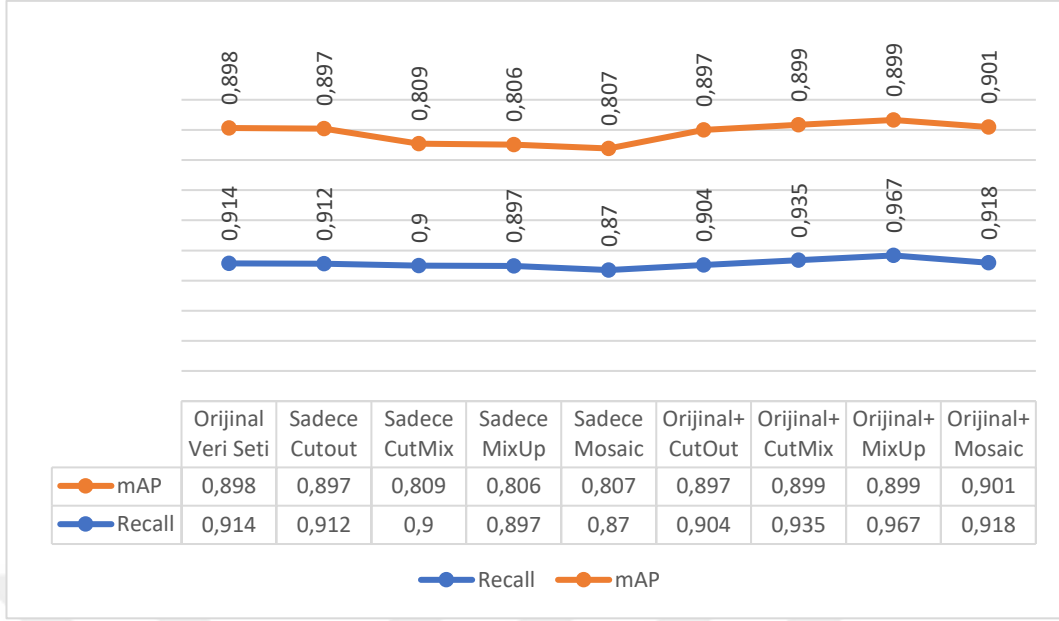
Çizelge 4.8. Veri artırma teknikleri ile eğitilen YOLOv8m modelinin keçi sürüsü veri seti üzerindeki performans sonuçları

Veri Artırma	Recall	mAP50	mAP50-95
<b>Orijinal + CutMix</b>	0.673	0.780	0.601
<b>Orijinal + CutOut</b>	0.630	0.689	0.425
<b>Orijinal + MixUp</b>	0.448	0.471	0.217
<b>Orijinal + Mosaic</b>	0.543	0.613	0.348

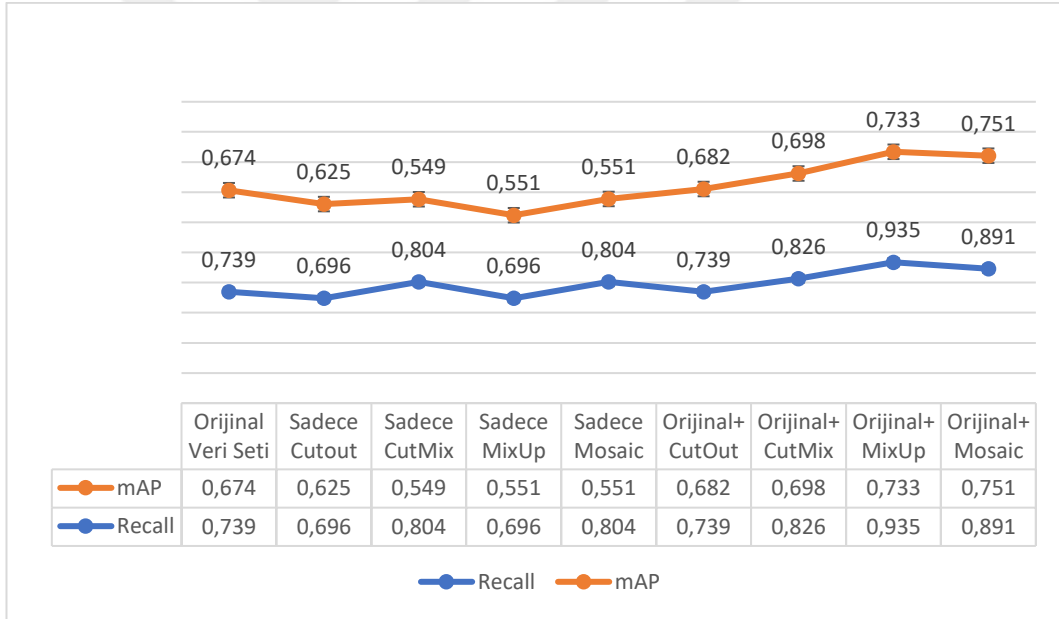
#### 4.5. Veri Arttırmanın Modellerin Genel Performansına Etkisinin İncelenmesi

Bu bölümde veri arttırmanın modeller üzerindeki performans artışı detaylıca incelenmiştir. Şekil 4.5, Cascade R-CNN için sığır ve keçi sürüsü veri setleri üzerindeki performansları, Şekil 4.6 ise YOLOv8m için aynı tekniklerin etkisini göstermektedir.

Veri arttırma teknikleri, Cascade R-CNN ve YOLOv8m modellerinde farklı etkiler yaratmıştır. Sığır veri setinde, Orijinal + MixUp (Cascade R-CNN) ve Orijinal + Cutout (YOLOv8m) en iyi sonuçları vermiş, recall ve mAP değerlerinde %5-10'luk bir artış sağlamıştır. Keçi sürüsü veri setinde ise, küçük veri seti ve nesne boyutları nedeniyle tekniklerin etkisi sınırlı kalmış, ancak Orijinal + MixUp (Cascade R-CNN) ve Orijinal + Cutmix (YOLOv8m) en iyi performansı göstermiştir. Bu sonuçlar, veri arttırmanın büyük ve homojen veri setlerinde daha etkili olduğunu, ancak küçük ve heterojen veri setlerinde dikkatli bir teknik seçimi gerektirdiğini göstermektedir.



a) Veri arttırmanın sığır veri setindeki etkisinin recall ve mAP değer karşılaştırması

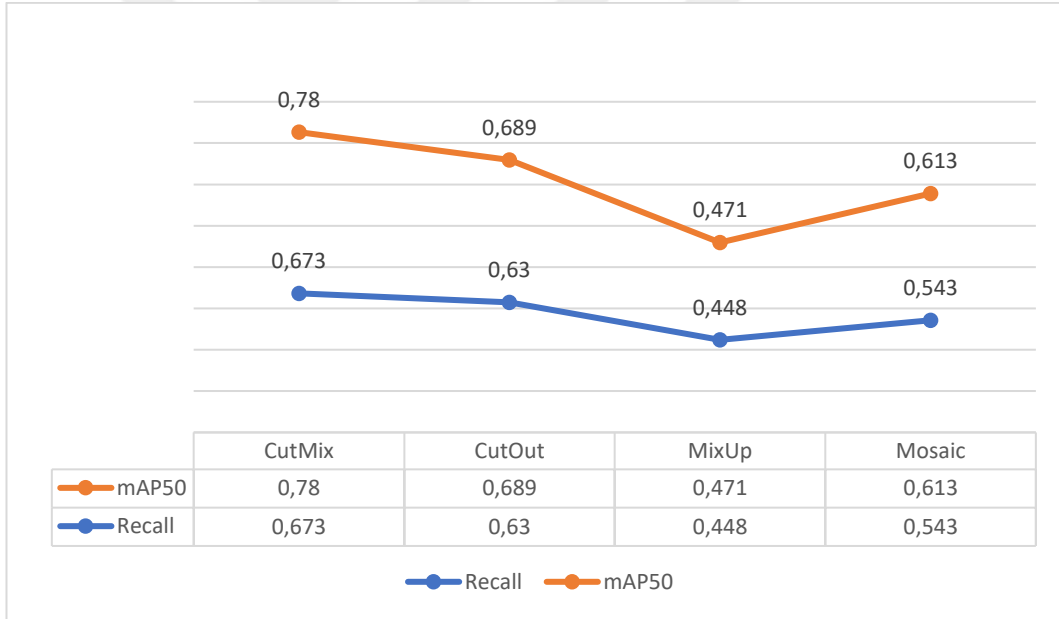


b) Veri arttırmanın keçi sürüsü veri setindeki recall ve mAP değer karşılaştırması

Şekil 4.5. Veri arttırma teknikleri ile eğitilen Cascade R-CNN modelinin sığır ve keçi sürüsü veri setindeki recall ve mAP değer karşılaştırması.



a) Veri arttırmanın sığır veri setindeki etkisinin recall ve mAP değer karşılaştırması



b) Veri arttırmanın keçi sürüsü veri setindeki recall ve mAP değer karşılaştırması

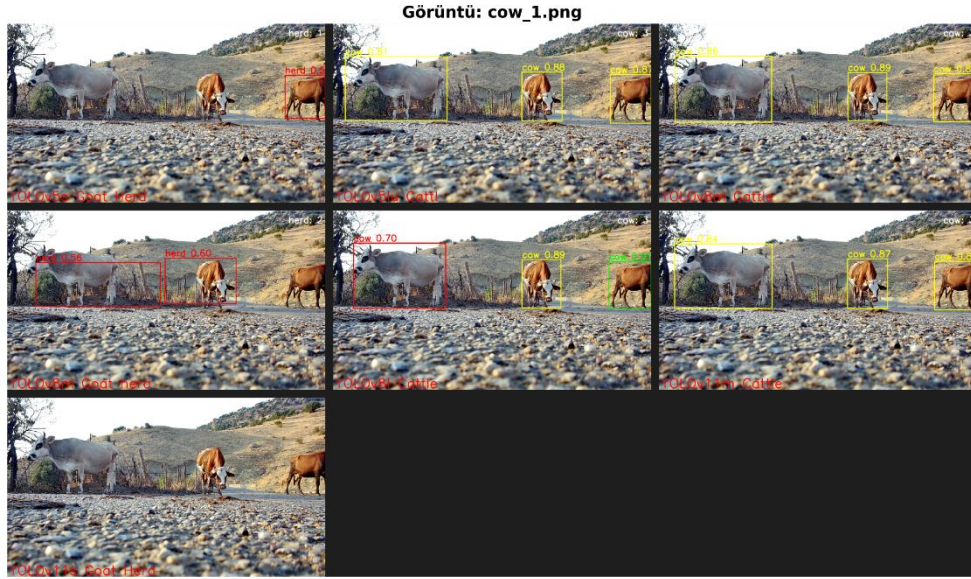
Şekil 4.6. Veri arttırma teknikleri ile eğitilen YOLOv8m modelinin sığır ve keçi sürüsü veri setindeki recall ve mAP değer karşılaştırması

#### 4.6. Görsel Performans Analizi

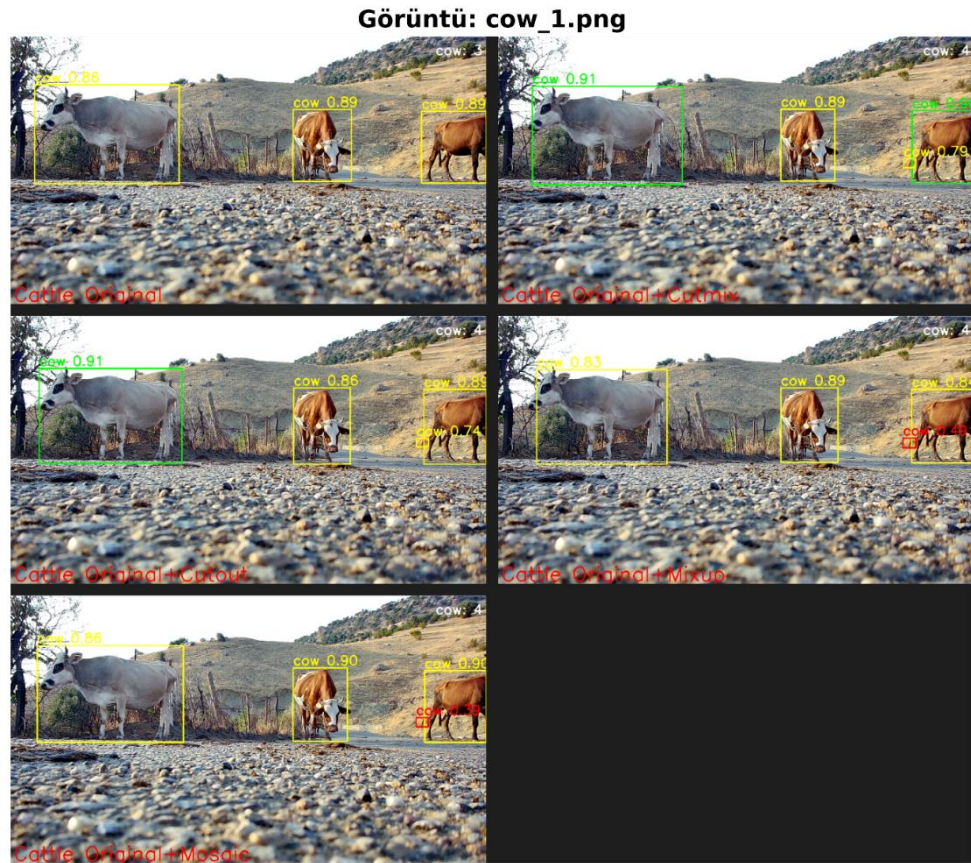
Bu bölümde, YOLO serisinin varyantları ile veri artırma teknikleri kullanılarak eğitilmiş en başarılı YOLO modelinin, İHA görüntüleri üzerindeki nesne tespit performansı görsel olarak incelenmiştir. Görsel analizler, sığır veri setinden seçilen üç görüntü ve keçi sürüsü veri setinden seçilen üç görüntü üzerinde gerçekleştirilmiştir. Analiz kapsamında, iki ana model grubu değerlendirilmiştir:

1. Orijinal veri setlerinde YOLO varyantları arasında sığır veri setinde en yüksek performansı sergileyen modeller:
  - YOLOv5l (mAP50: 0.918, recall: 0.862),
  - YOLOv8m (mAP50: 0.923, recall: 0.857),
  - YOLOv8l (mAP50: 0.923, recall: 0.860),
  - YOLO11m (mAP50: 0.917, recall: 0.862).
2. Keçi sürüsü veri setinde ise en iyi sonuçları elde eden modeller:
  - YOLOv5s (mAP50: 0.705, recall: 0.630),
  - YOLOv8m (mAP50: 0.708, recall: 0.630),
  - YOLO11s (mAP50: 0.659, recall: 0.577).
3. Veri artırma teknikleriyle eğitilmiş (Cutmix, Cutout, Mixup ve Mosaic) YOLOV8m modelleri.

Tespit sonuçları, her iki veri seti için tek bir görsel üzerinde karşılaştırmalı olarak görselleştirilmiştir. Analizlerde, nesne tespit performansları sınırlayıcı kutular, nesne etiketleri ve güven skorları ile görselleştirilerek detaylı bir şekilde incelenmiştir.



a) En başarılı YOLO modellerinin tespit sonuçları

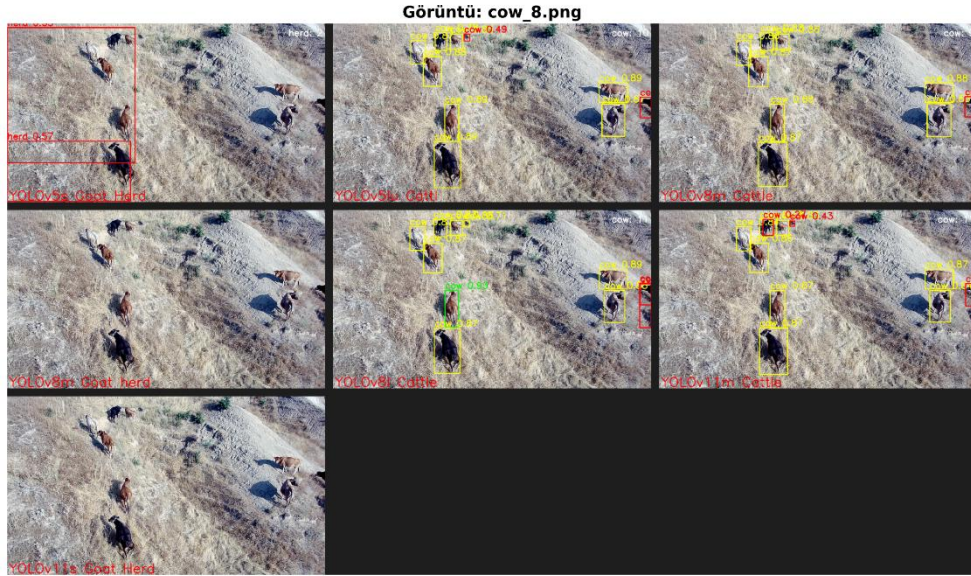


b) Veri artırma teknikleriyle eğitilen modellerin tespit sonuçları

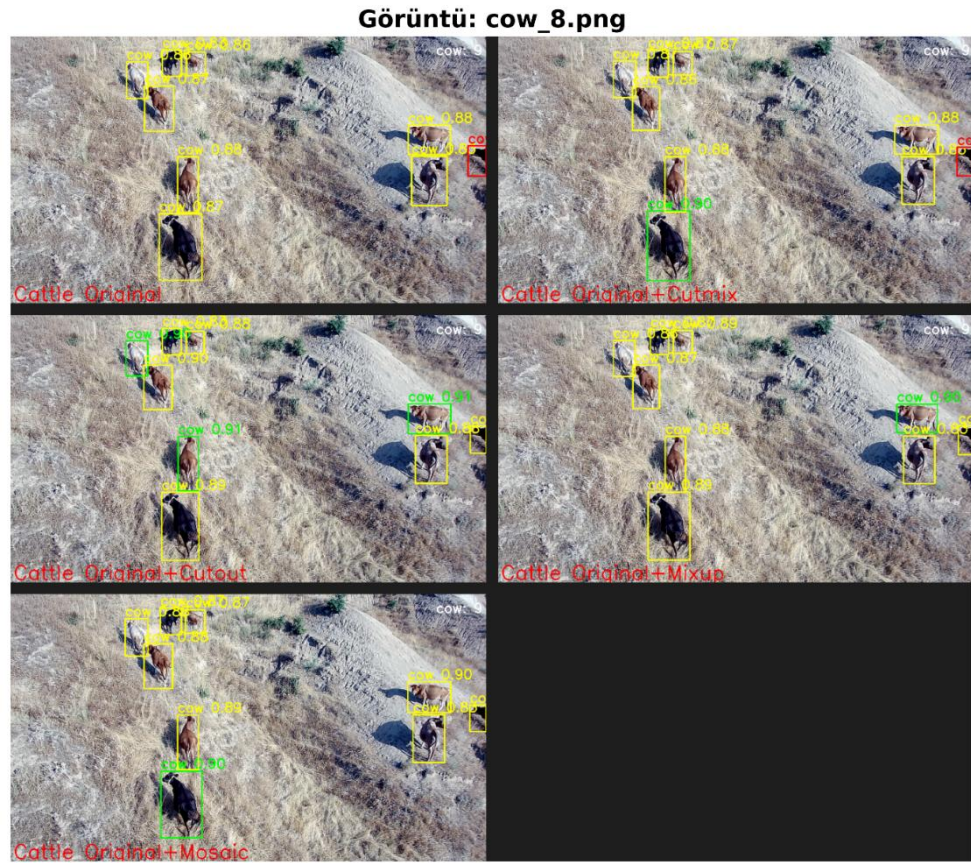
Şekil 4.7. Orijinal ve artırılmış veri seti ile eğitilen YOLO modellerinin sığır tespit sonuçları (Görsel 1).

Şekil 4.7 (a)'da yedi farklı YOLO modeli ile bir İHA görüntüsü üzerinde yapılan nesne tespit sonuçları sunulmuştur. Görselde, üçü ön planda belirgin, biri ise arka planda ve arka planla karışmış olmak üzere toplam dört sığır bulunmaktadır. %90 üzeri güven skoru yeşil, %70–90 arası sarı ve %70 altı kırmızı renklerle gösterilmiştir. YOLOv5l, YOLOv8m, YOLOv8l ve YOLO11m modelleri yalnızca ön plandaki üç sığırı algılayabilmiş, arka plandaki dördüncü sığırı tespit edememiştir. Bu modeller arasında, YOLOv8m; 0,86, 0,89 ve 0,89 güven skorlarıyla en yüksek ve dengeli performansı göstermiştir. Diğer modellerin ortalama skorları ise YOLOv5lu'da 0,85, YOLOv8l'de 0,83 ve YOLO11m'de 0,86 civarındadır. Bu bulgular, YOLOv8m'in sığır nesne tespiti için en kararlı ve başarılı model olduğunu göstermektedir.

Şekil 4.7 (b)'de ise, YOLOv8m modelinin sadece orijinal veriyle değil, aynı zamanda çeşitli veri artırma teknikleriyle eğitilmiş versiyonlarının performansı karşılaştırılmıştır. Görseldeki dört sığırın tamamı, veri artırma uygulanmış modeller tarafından başarıyla tespit edilmiştir. Bu, orijinal modele göre belirgin bir iyileşme sağlamaktadır. CutMix tekniği, 0,91, 0,89, 0,90 ve 0,79 skorlarıyla en yüksek ortalama başarıya (0.8725) ulaşmıştır. CutOut tekniği de dördüncü sığırı daha düşük bir skorla (0.74) algılamış olsa da, genel ortalama (0.8500) bakımından başarılıdır. Buna karşılık, Mixup (0.49) ve Mosaic (0.39) yöntemleri, arka planla karışan dördüncü sığırı düşük güvenle tespit etmiş ve genel ortalamaları 0.7625'te kalmıştır. Elde edilen sonuçlar, veri artırma yöntemlerinin tespit kabiliyetini geliştirdiğini; özellikle Cutmix'in zorlayıcı nesnelere tanıma konusunda öne çıktığını ortaya koymaktadır.



a) En başarılı YOLO modellerinin tespit sonuçları

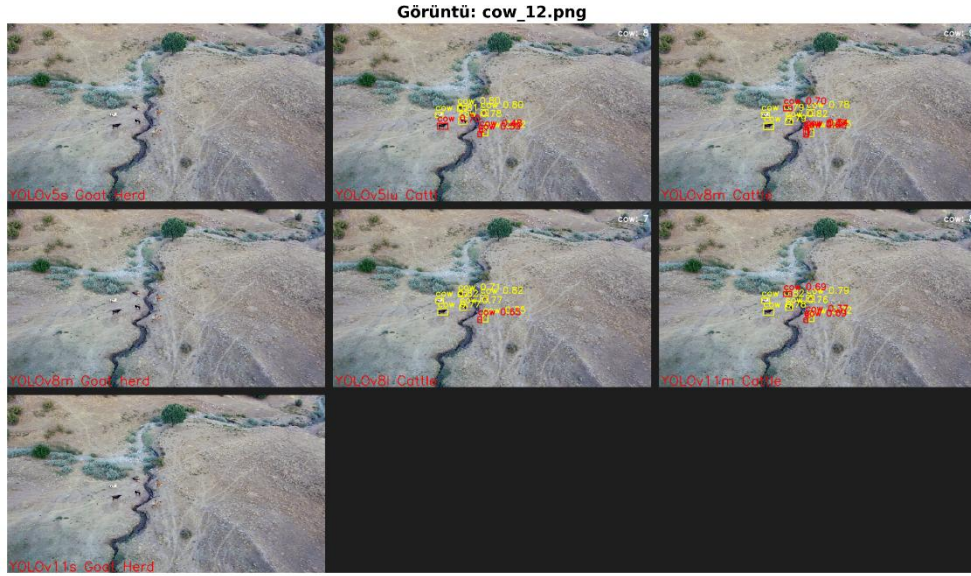


b) Veri artırma teknikleriyle eğitilen modellerin tespit sonuçları

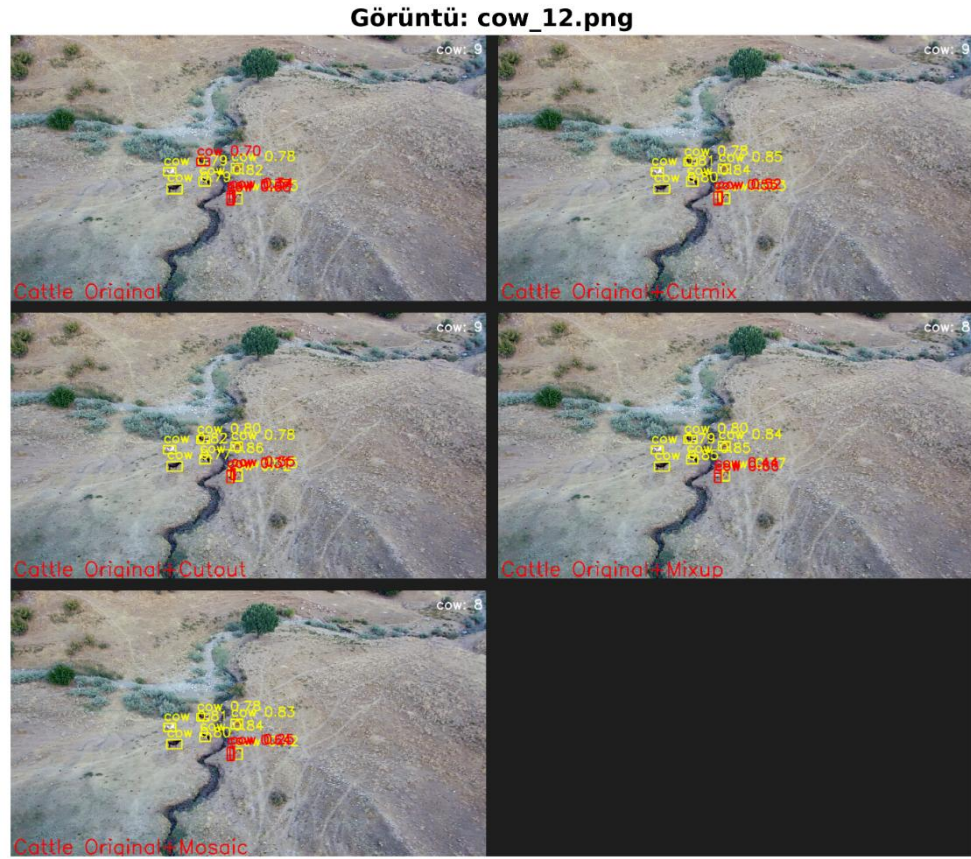
Şekil 4.8. Orijinal ve artırılmış veri seti ile eğitilen YOLO modellerinin sığır tespit sonuçları (Görsel 2).

Şekil 4.8 (a) ise, yine yedi YOLO modelinin “cow\_8” adlı İHA görüntüsü üzerindeki nesne tespit performansını karşılaştırmaktadır. Görüntüde toplam dokuz sığır yer almakta ve sığırların renkleri zeminle oldukça benzer olduğu için tespit görevi zorlaşmaktadır. Analiz sonucunda, YOLOv5l modeli dokuz doğru tespit yanı sıra bir FP (yanlış pozitif) üreterek toplam on nesne algılamıştır. Buna karşılık, YOLOv8m modeli hem dokuz sığırı eksiksiz şekilde tespit etmiş hem de yanlış pozitif üretmeyerek en temiz sonucu sunmuştur. YOLOv8l ve YOLO11m modelleri ise ikişer yanlış pozitif ile toplam on bir nesne tespit ederek fazla algılama eğilimi göstermiştir. Sığır veri setiyle eğitilmiş modellerin çoğunda tespit skorları %90’ın altında kalmıştır; sadece YOLOv8l bir sığır için 0,91 skorla bu eşiği aşmıştır. Ortalama güven skorları; YOLOv8m Sığır için 0,85–0,90, YOLOv5lu için 0,78–0,90, YOLOv8l için 0,71–0,91 ve YOLO11m için 0,72–0,88 aralığında değişmiştir. Bu sonuçlar, YOLOv8m sığır görüntüsü için yanlış pozitif üretmeden en dengeli ve güvenilir sonuçları sunduğunu; YOLOv8l ve YOLO11m’in ise fazla nesne algılama eğiliminde olduğunu göstermektedir.

Şekil 4.8 (b) ise, aynı sığır görüntüsü üzerinde, YOLOv8m modelinin farklı veri artırma teknikleriyle (CutMix, CutOut, MixUp, Mosaic) eğitilmiş sürümlerinin performansı değerlendirilmiştir. Dokuz sığırın tamamı, tüm veri artırma yöntemleriyle eğitilen modeller tarafından doğru şekilde tespit edilmiş ve herhangi bir yanlış pozitif oluşmamıştır. Bu durum, veri artırmanın tespit başarısını belirgin biçimde artırdığını göstermektedir. Orijinal model 0,87–0,90 aralığında skorlar üretmiş; ancak hiçbir nesne için %90’ı aşamamıştır. CutMix, benzer aralıkta skorlar vermiş ve bir sığırda 0,90 skorla iyi bir sonuç sağlamıştır. CutOut ise 0,88–0,92 aralığında tespitler yapmış ve üç sığır için 0,90 üzeri (0,90, 0,91, 0,92) skorlarla en yüksek başarıyı göstermiştir. MixUp ve Mosaic yöntemleri de sırasıyla birer sığırda %90’lık skora ulaşmış olsa da genel ortalama performansları CutMix ve özellikle CutOut’un gerisinde kalmıştır. Bu bulgular, veri artırma tekniklerinin özellikle zorlayıcı görüntü koşullarında model başarımını güçlendirdiğini ve CutOut’un tutarlı sonuçları verdiğini ortaya koymaktadır.



a) En başarılı YOLO modellerinin tespit sonuçları

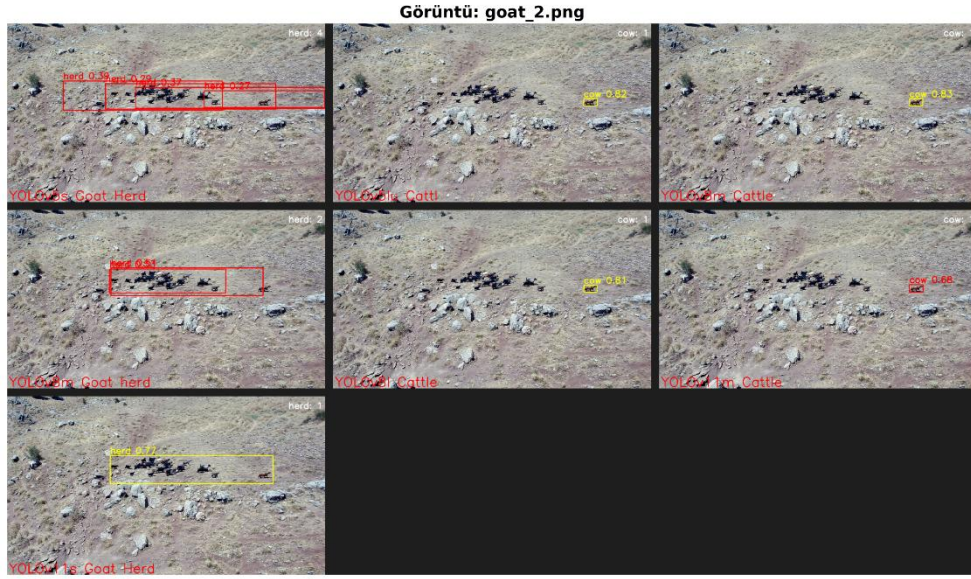


b) Veri artırma teknikleriyle eğitilen modellerin tespit sonuçları

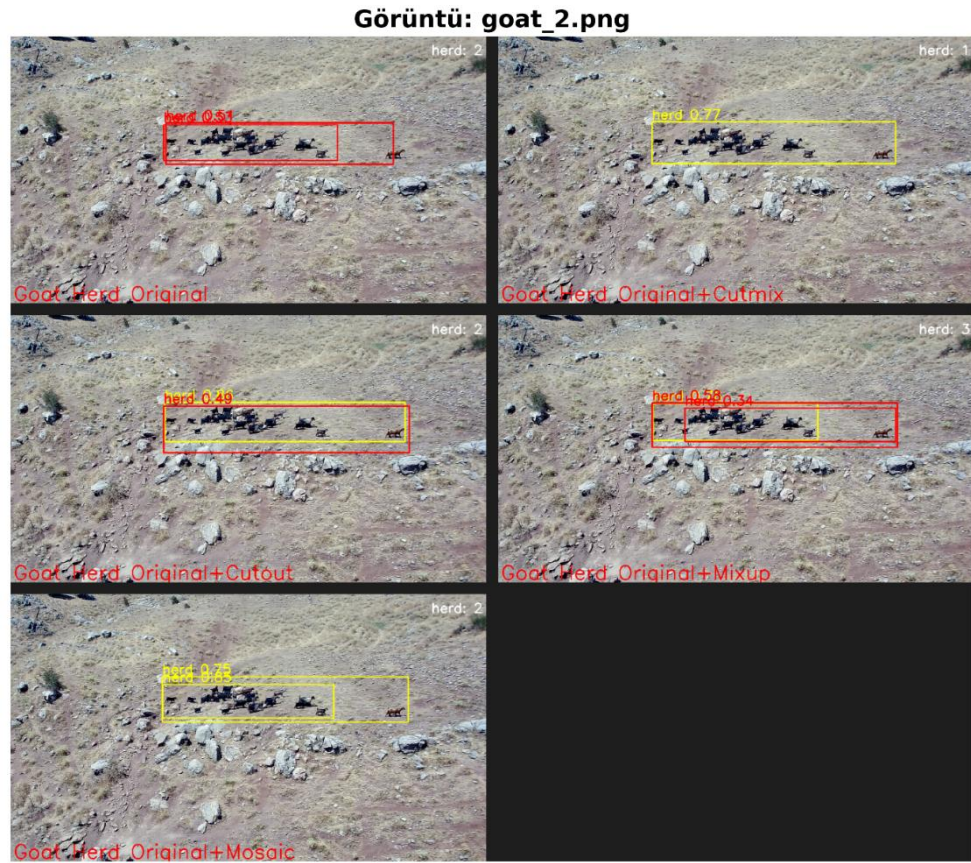
Şekil 4.9. Orijinal ve artırılmış veri seti ile eğitilen YOLO modellerinin sığır tespit sonuçları (Görsel 3).

Şekil 4.9 (a), yedi YOLO modelinin “cow\_12” adlı yüksek irtifa İHA görüntüsü üzerindeki nesne tespit performansını göstermektedir. Görselde yer alan yedi sığır, zeminle benzer renklerde olup insan gözüyle ayırt edilmesi zor bir sahne sunmaktadır. Analiz sonuçlarına göre, YOLOv5l modeli yedi doğru tespit yanı sıra bir FP ile sekiz nesne algılamıştır. YOLOv8m modeli ise iki FP ile toplam dokuz nesne tespit etmiş; buna karşın YOLOv8l, tüm gerçek sığırları eksiksiz ve yanlış pozitif üretmeden algılamıştır. YOLO11m modeli sekiz nesne algılamış ve bir FP üretmiştir. Güven skorları açısından incelendiğinde, YOLOv8l tüm sığırları %75 üzeri skorlarla tespit etmesine rağmen bir sığırda 0,65 değeriyle %70 altına düşmüştür. Diğer yandan, YOLOv5lu, YOLOv8m ve YOLO11m modellerinde üçer sığırda %70’in altında skorlar gözlemlenmiştir. Bu bulgular, YOLOv8l’nin özellikle zorlu koşullarda FP üretmeden en istikrarlı ve yüksek doğruluğa sahip tespitleri gerçekleştirdiğini, YOLOv8m’in fazla algılama eğiliminde olduğunu, diğer modellerin ise düşük skorlu tespitlerde yoğunlaştığını göstermektedir.

Şekil 4.9 (b)’de ise YOLOv8m modelinin orijinal veriyle ve farklı veri artırma teknikleriyle eğitilmiş sürümleri karşılaştırılmıştır. Orijinal YOLOv8m sığır görüntüsü üzerinde yedi gerçek sığır ile birlikte toplam dokuz nesne tespit etmiş, iki FP üretmiş ve üç nesnede %70’in altında skorlar vermiştir. CutMix ve CutOut ile eğitilen modeller, aynı sayıda doğru ve yanlış tespit üretmesine rağmen düşük skorlu nesne sayısını ikiye indirerek genel doğruluğu artırmıştır. Öte yandan, MixUp ve Mosaic modelleri sekizer nesne tespit etmiş, yalnızca birer yanlış pozitif üretmiş ve iki sığırda 0,65–0,69 aralığında düşük skorlar gözlemlenmiştir. Diğer tespitlerde ise skorlar orijinal modele kıyasla yükselmiştir. Bu sonuçlar, veri artırma yöntemlerinin genel başarıyı artırdığını; CutMix ve CutOut’un skor iyileştirmesi sağlarken yanlış pozitif sayısını sabit tuttuğunu, MixUp ve Mosaic’in ise FP sayısını düşürüp aynı zamanda skorları da iyileştirdiğini göstermektedir. Özellikle MixUp ve Mosaic’in, bu tür zorlu görsellerde FP oranını azaltmada avantaj sağladığı söylenebilir.



a) En başarılı YOLO modellerinin tespit sonuçları



b) Veri artırma teknikleriyle eğitilen modellerin tespit sonuçları

Şekil 4.10. Orijinal ve artırılmış veri seti ile eğitilen YOLO modellerinin keçi sürüsü tespit sonuçları (Görsel 1).

Şekil 4.10 (a), yedi farklı YOLO modelinin “goat\_2” adlı İHA görüntüsü üzerindeki keçi sürüsü tespit performansını karşılaştırmaktadır. Görselde, sürünün büyük kısmı bir arada bulunurken bir keçi sürüden ayrılmıştır. YOLOv5s, dört sürü tespit etmiş ancak bunların her biri düşük güven skorlarıyla (%70'in oldukça altında) kaydedilmiş ve keçi bulunmayan bölgeleri de sürü olarak algılayarak birden fazla FP üretmiştir. YOLOv8m modeli, iki sürü tespit etmiş; biri ana sürü (0,51), diğeri ayrılan keçiyi de içeren ikinci bir sürü (0,53) olmak üzere düşük güven seviyelerinde kalmıştır. Öte yandan, YOLO11s modeli yalnızca tek bir sürü tespit etmiş, bu sürü tüm keçileri kapsamış ve 0,77 güven skoru ile en yüksek ve doğru sonucu sunmuştur. Bu bulgular, YOLO11s modelinin keçi sürüsü veri setinde en başarılı tespit performansına sahip olduğunu, YOLOv5s'in ise hatalı sınıflandırmalarla başarısız sonuçlar verdiğini ortaya koymaktadır.

Şekil 4.10 (b)'de, aynı görüntüde YOLOv8m modelinin farklı veri artırma teknikleriyle eğitilmiş sürülerinin performansı analiz edilmiştir. Orijinal modelin iki sürü tespiti (0,51 ve 0,53) ayrı ayrı sürüleri göstermesine rağmen, birinin FP olarak değerlendirildiği görülmüştür. CutMix ile eğitilen model, tüm keçileri tek bir sürü olarak başarıyla tespit etmiş ve 0,77 güven skoru ile dikkate değer bir gelişme sağlamıştır. CutOut tekniğiyle eğitilen model ise iki sürü (0,86 ve 0,49) algılamış, ancak düşük skorlu ikinci sürüyle false positive üretmiştir. MixUp modeli, üç sürü tespit etmiş; ana sürüyü 0,76, ayrılan keçiyi 0,58 ve ilave bir sürüyü 0,34 skorla algılamış, bu durum sınıflandırma hatası üretse de bazı skorların yükseldiğini göstermektedir. Mosaic ile eğitilen model, iki sürü algılamış ve 0,75 ile 0,85 arasında güven skorları sunarak orijinal modele kıyasla daha başarılı olmuştur. Bu veriler, veri artırma tekniklerinin genel tespit performansını artırdığını; özellikle CutMix'in tutarlı tek sürü algısı ve yüksek skorla en başarılı sonucu verdiğini, CutOut'un kısmi başarısını, MixUp'un karışık ancak bazı kazanımlar sağladığını ve Mosaic'in dengeli bir iyileştirme sunduğunu göstermektedir.



a) En başarılı YOLO modellerinin tespit sonuçları

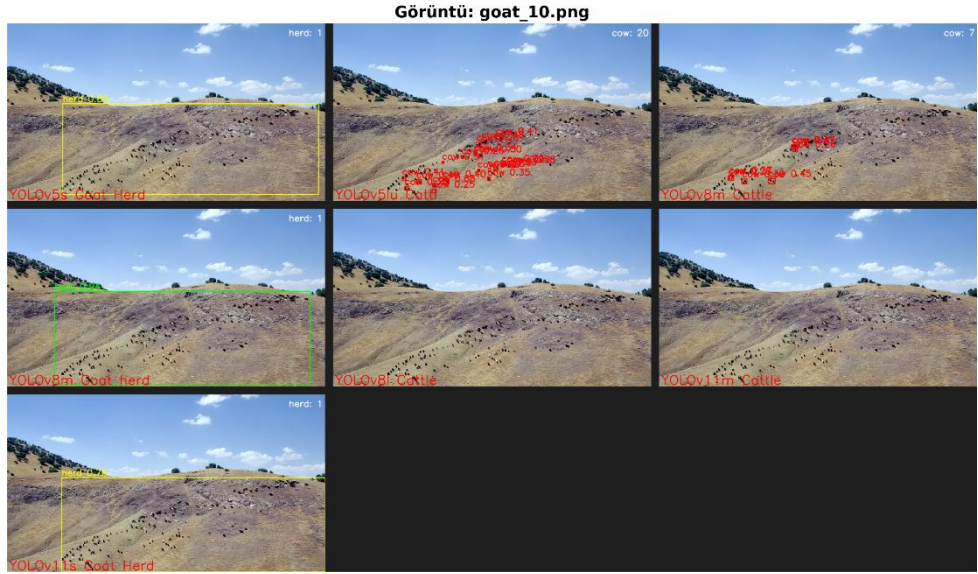


b) Veri artırma teknikleriyle eğitilen modellerin tespit sonuçları

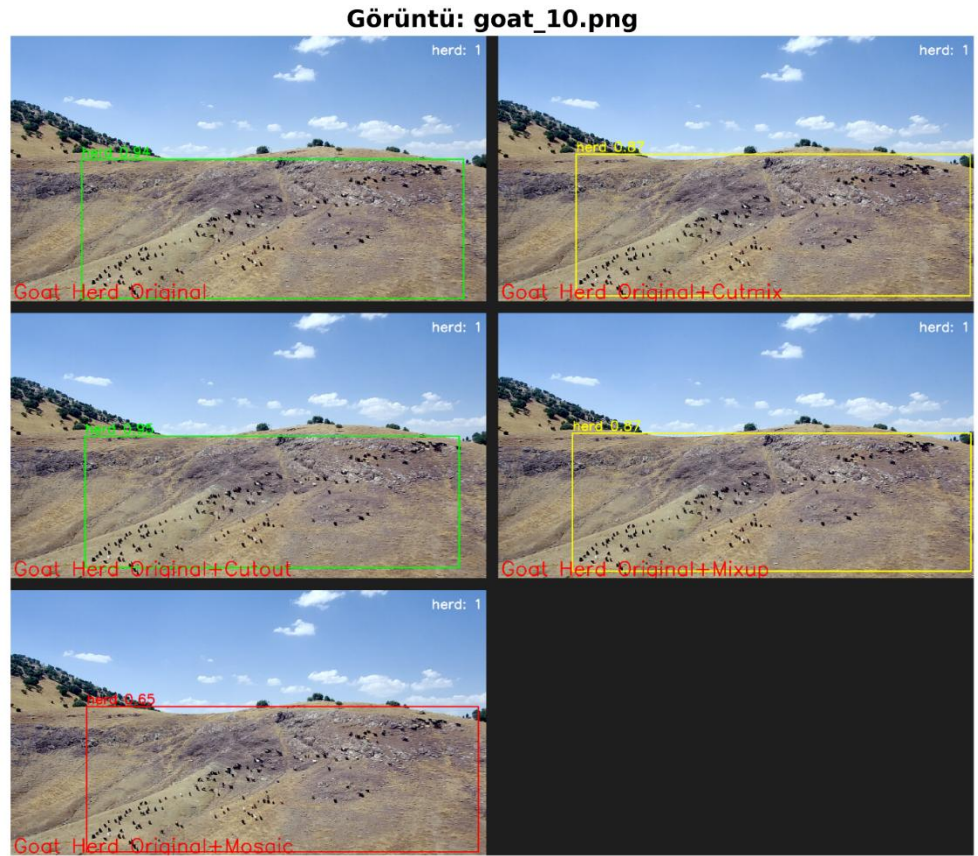
Şekil 4.11. Orijinal ve artırılmış veri seti ile eğitilen YOLO modellerinin keçi sürüsü tespit sonuçları (Görsel 2).

Şekil 4.11 (a), yedi YOLO modelinin “goat\_7” adlı İHA görüntüsü üzerindeki keçi sürüsü tespit performansını göstermektedir. Görüntüde, ana sürüye ek olarak yakın mesafede yer alan tek bir keçi daha bulunmaktadır. YOLOv5s, ana sürüyü 0,91 güven skoru ile başarılı şekilde tespit etmiş, ancak sürüden ayrılan keçiyi dahil etmemiştir. Benzer şekilde, YOLO11s de yalnızca ana sürüyü 0,89’luk güvenle algılamış ve ayrılan keçiyi bağımsız bir nesne olarak değerlendirmemiştir. Öte yandan, YOLOv8m modeli iki sürü tespit etmiş; ana sürüyü 0,81 güven skoru ile, sürüden ayrılan keçiyi ise 0,44 gibi düşük bir skorla sınıflandırmıştır. Bu bulgular, YOLOv5s ve YOLO11s modellerinin yüksek doğruluk ve güven ile ana sürüyü başarılı şekilde algıladığını, YOLOv8m’in ise tüm keçileri kapsamakla birlikte daha düşük skorlarla sınırlı kaldığını göstermektedir.

Şekil 4.11 (b), YOLOv8m modelinin veri artırma teknikleriyle (CutMix, CutOut, MixUp, Mosaic) eğitilmiş sürülerinin “goat\_7” görüntüsü üzerindeki performansı değerlendirilmiştir. Orijinal modelin iki ayrı sürü algılamasının aksine, veri artırma ile eğitilmiş tüm modeller yalnızca tek bir sürü tespit etmiş ve tüm keçileri (ana sürü + ayrılan keçi) doğru şekilde bu sürüye dahil etmiştir. Elde edilen güven skorları CutMix ve Mosaic için 0,95, MixUp için 0,91 ve CutOut için 0,86 olarak ölçülmüştür. Bu sonuçlar, veri artırma tekniklerinin genel tespit doğruluğunu anlamlı düzeyde artırdığını ortaya koymaktadır. Özellikle CutMix ve Mosaic, yüksek skorları ile en etkili yöntemler olurken, MixUp da güçlü bir katkı sağlamıştır. CutOut ise görece olarak daha düşük bir skorla sınırlı fakat olumlu bir iyileştirme sunmuştur. Genel olarak, bu tekniklerin YOLOv8m modelinin eksik kalan yönlerini dengelediği ve tek sürü altında tüm nesnelere yüksek güvenle algılamasını sağladığı sonucuna varılmıştır.



a) En başarılı YOLO modellerinin tespit sonuçları



b) Veri artırma teknikleriyle eğitilen modellerin tespit sonuçları

Şekil 4.12. Orijinal ve artırılmış veri seti ile eğitilen YOLO modellerinin keçi sürüsü tespit sonuçları (Görsel 3).

Şekil 4.12 (a), yedi YOLO modelinin “goat\_10” adlı yüksek irtifadan çekilmiş İHA görüntüsü üzerindeki keçi sürüsü tespit performansını sunmaktadır. Görselde, geniş bir araziye yayılmış dağınık bir sürü yer almaktadır. Keçi sürüsü veri setiyle eğitilmiş üç model (YOLOv5s, YOLOv8m ve YOLO11s) üzerinde yapılan analizler, model davranışlarını karşılaştırmalı olarak değerlendirmektedir. YOLOv5s, sürüyü 0,89 güven skoru ile doğru şekilde tespit etmiş ve herhangi bir FP üretmemiştir. YOLOv8m ise sürünün tamamını 0,94 gibi yüksek bir skorla tespit ederek en iyi performansı göstermiştir. Öte yandan, YOLO11s modeli sürüyü 0,78 güven skoru ile tanımlamış ancak sınırlayıcı kutunun keçi olmayan alanları da kapsadığı ve bu nedenle bir FP ürettiği gözlemlenmiştir. Bu bulgular, yüksek irtifadan alınan görüntülerde YOLOv8m’in en başarılı sonuçları sunduğunu, YOLOv5s’in etkili bir alternatif olduğunu ve YOLO11s’in yanlış pozitif eğilimi nedeniyle daha düşük başarı sağladığını ortaya koymaktadır.

Şekil 4.12 (b), aynı görüntü üzerinde YOLOv8m modelinin veri artırma teknikleriyle eğitilmiş versiyonlarının performansını incelemektedir. Orijinal model, sürüyü 0,94 güven skoru ile doğru biçimde algılamış ve FP üretmemiştir. CutMix modeli, sürüyü 0,87 güven skoru ile tespit etmiş; ancak sınırlayıcı kutunun keçi bulunmayan alanları da kapsaması nedeniyle yanlış pozitif üretmiştir. CutOut, 0,95 güven skoru ile en yüksek başarıyı sağlamış ve FP içermeyen en dengeli performansı göstermiştir. MixUp modeli de CutMix’e benzer şekilde 0,87 skorla sürüyü tanımlamış ancak geniş sınırlayıcı kutu nedeniyle hatalı bölge kapsamış ve FP üretmiştir. Mosaic tekniği ise yalnızca 0,65’lik düşük bir güven skoruyla tespit yapmış ve kutu genişliği nedeniyle yanlış pozitif üretmiştir. Bu sonuçlar, veri artırma tekniklerinin orijinal modele kıyasla karışık sonuçlar verdiğini; CutOut’un yüksek irtifalı ve dağınık sürü görüntülerinde en uygun yöntem olduğunu, CutMix ve MixUp’un sınırlı katkı sağladığını ve Mosaic’in düşük performans sergilediğini göstermektedir.

#### **4.7. Genel Değerlendirme**

Bu bölümde, YOLO ailesine ait çeşitli model varyantlarının İHA görüntülerindeki nesne tespit performansı, sığır ve keçi Sürüsü veri setlerine ait toplam on iki görsel (Şekil 4.7–Şekil 4.12) üzerinden değerlendirilmiştir. Sığır veri setinde YOLOv8m modeli, belirgin nesnelere yüksek başarı göstermiş (0,86–0,90

arası güven skorları) ve FP üretmeden istikrarlı bir performans sergilemiştir (Şekil 4.7 (a), Şekil 4.8 (a)). Ancak, zorlu arka plan koşullarında (Şekil 4.7 (a)) ve yüksek irtifadan çekilen görüntülerde (Şekil 4.9 (a)) sınırlamalar gözlenmiştir. YOLOv8l modeli, daha geniş skor aralığında (0,65–0,91) tutarsızlık göstermekle birlikte, FP üretmeden doğru tespitler yapabilmiş, fakat fazla algılama eğilimleri de sergilemiştir. YOLOv5l ve YOLO11m modelleri ise genel olarak 0,78–0,90 arasında değişen dengeli fakat eksik tespitlerle sınırlı kalmış, bazı örneklerde FP üretmiştir. Veri artırma teknikleriyle eğitilen modeller, özellikle CutMix (ör. 0,79–0,91 arası; Şekil 4.7 (b), Şekil 4.8 (b)) ve CutOut (ör. 0,88–0,92 arası; Şekil 4.8 (b)) ile anlamlı performans artışı sağlamıştır. Buna karşın MixUp ve Mosaic, bazı durumlarda düşük skorlarla (ör. 0,49, 0,39) ve sınırlı FP azaltımıyla daha zayıf sonuçlar sunmuştur (Şekil 4.9 (b)).

Keçi sürüsü veri setinde ise YOLOv8m modeli, özellikle yüksek irtifalı ve dağınık sürü sahnelerinde en yüksek başarıyı (0,94; Şekil 4.12 (a)) elde etmiş; YOLO11s ise 0,77–0,89 aralığında güven skorlarıyla başarılı sonuçlar vermekle birlikte geniş sınırlayıcı kutular nedeniyle FP üretmiştir. YOLOv5s modeli ise çok düşükten (0,27) yükseğe (0,91) kadar geniş bir aralıkta tespit yapmış ve özellikle düşük güvenli, hatalı sınıflandırmalarla dikkat çekmiştir (Şekil 4.10 (a)). Veri artırma teknikleriyle eğitilen modeller, sürüleri tek birim olarak algılamada genel olarak başarılı olmuş; CutMix (Şekil 4.10 (b) ve Şekil 4.11 (b)'de 0,77 ve 0,95) ve Mosaic (Şekil 4.11 (b)'de 0,95) modelleri yüksek skorlarla sağlam sonuçlar vermiştir. CutOut yöntemi ise 0,95'lik güven skoru (Şekil 4.12 (b)) ile en tutarlı ve hatasız performansı sunmuştur. MixUp ise 0,87 skora rağmen FP eğilimi göstermiştir. Tüm bu bulgular, YOLOv8m'in her iki veri setinde genel başarı açısından öne çıktığını; CutMix ve CutOut tekniklerinin modelin genelleme yeteneğini artırmada etkili olduğunu; MixUp ve Mosaic'in ise bazı durumlarda FP üretme riskini artırabildiğini göstermektedir. Sonuç olarak, model başarımı büyük ölçüde görüntülerin yapısal zorluklarına, nesne dağılımına ve kullanılan veri artırma yöntemine bağlı olduğu görülmüştür.

## 5. SONUÇ

Bu tez çalışmasında, İHA teknolojisinin hayvancılıkta sürü takibi ve hayvan izleme uygulamalarında nasıl kullanılabileceğini araştırılmıştır. Özellikle derin öğrenme temelli nesne tespiti algoritmalarının İHA görüntülerinde ne kadar başarılı çalıştığı, farklı mimarilerin ve veri artırma yöntemlerinin bu başarıyı nasıl etkilediği detaylı bir şekilde incelenmiştir:

- Sığır veri seti, genellikle düşük irtifada (50 cm – 5 m) ve büyük nesnelere içeren görüntülerden oluştuğu için, birçok model bu veri setinde yüksek doğruluk göstermiştir. Cascade R-CNN ve EfficientDet gibi modeller, çok aşamalı tespit mekanizmaları sayesinde yüksek mAP ve recall değerlerine ulaşmıştır. Bu durum, büyük nesnelere tespitinde detaylı analiz yapan mimarilerin avantaj sağladığını göstermektedir. Öte yandan, keçi sürüsü veri seti daha yüksek irtifadan (10 – 30 m) ve küçük nesnelere (32–64 piksel) oluştuğu için tespit görevinde daha zorlu koşullar sunmuş ve modellerin başarımında ciddi farklılıklar gözlenmiştir.
- YOLO ailesinden YOLOv8m modeli, her iki veri setinde de oldukça dengeli ve başarılı bir performans sergilemiştir. YOLOv8m'nin çapasız (anchor-free) yapısı, daha hızlı ve esnek öğrenme imkânı sağlarken; özellikle küçük nesnelere başarıyı da artırmıştır. Yüksek irtifadan çekilen keçi görüntülerinde dahi YOLOv8m, diğer varyantlara göre daha istikrarlı sonuçlar vermiştir. Bu modelin hem sığır hem de keçi sürüsü veri setinde gösterdiği başarı, bu modeli tarımsal uygulamalar adına önemli bir model haline getirmiştir.
- Veri artırma tekniklerinin etkisi de önemli bir tartışma konusu olmuştur. Özellikle Orijinal + Cutmix ve Orijinal + Mosaic gibi hibrit yaklaşımlar, küçük nesnelere yer aldığı ve yüksek irtifalı sahnelerde gözle görülür şekilde başarıyı artırmıştır. Bu teknikler, nesne çeşitliliğini artırarak modelin genelleme yeteneğini güçlendirmiştir. Ancak sadece tek bir artırma yöntemine dayalı veri setleri (örneğin, yalnızca Cutmix veya MixUp kullanılmış senaryolar) çoğu zaman ya orijinal model başarısının altında

kalmış ya da FP üretiminde artışa neden olmuştur. Bu bulgu, veri artırma stratejilerinin bağlama özgü ve dikkatli biçimde tasarlanması gerektiğini göstermektedir. CutOut tekniği özellikle sığır veri setinde, yani büyük nesnelere olduğu koşullarda daha başarılı sonuçlar üretmiştir. Bu teknik, nesne sınırlarını maskeleyerek modelin farklı varyasyonlara alışmasını sağlarken, doğal yapıyı fazla bozmaması nedeniyle tespit kararlılığı da korumuştur. Diğer yandan, Mosaic gibi çoklu görüntü birleştirme teknikleri, bazı durumlarda model kararsızlığına yol açmış ve düşük güven skorlarıyla sonuçlanmıştır.

- YOLO11m gibi daha yeni ve gelişmiş dikkat mekanizmaları barındıran modellerin, yüksek irtifa ve küçük nesne içeren keçi görüntülerinde zayıf performans sergilemesi dikkat çekicidir. Bu durum, mimari olarak karmaşık olan modellerin her zaman pratikte daha başarılı olamayabileceğini, gerçek saha verileriyle yapılan bağlamsal testlerin model seçiminde ne kadar önemli olduğunu ortaya koymuştur.
- Görsel analizler de sayısal sonuçları desteklemektedir. Örneğin, bazı sahnelerde yalnızca artırılmış modeller küçük ve arka planda kaybolmuş hayvanları tespit edebilmiş, orijinal modellerin bu nesnelere atladığı gözlemlenmiştir. Bu durum, veri artırmanın sadece istatistiksel değil, görsel doğruluğu da iyileştirdiğini ortaya koymaktadır.

Sonuç olarak, bu tez çalışmasında, nesne tespit başarısının yalnızca modele değil, verinin kalitesine, nesne boyutuna, çekim irtifasına ve uygulanan veri artırma yöntemlerine de bağlı olduğu görülmüştür. Gerçek sahalarda kullanılacak bir İHA tabanlı sürü takip sistemi tasarlanırken, bu faktörlerin tamamının göz önüne alınması gerekir. Bu çalışma, derin öğrenme modellerinin tarımsal uygulamalarda hangi koşullarda daha iyi çalıştığını ortaya koyarak uygulamaya dönük sağlam bir temel sunmaktadır.

## 5.1. Sınırlılıklar

Bu tez çalışmasında hem modellerin hem de farklı hayvan türlerinin İHA ile analizinde kapsamlı bir uygulaması yapılmasına rağmen bazı sınırlamaları bulunmaktadır:

- **Veri Seti Boyutu ve Dengesizliği:** Keçi sürüsü veri seti yalnızca 131 görüntüden oluşmaktadır. Derin öğrenme algoritmaları daha büyük ve çeşitli veri setleriyle daha etkili çalıştığı için, bu kısıtlı veri genelleme yeteneğini sınırlamış olabileceği düşünülmektedir.
- **Spektral Veri Eksikliği:** Tüm görüntüler RGB (Red–Green–Blue) kamerayla çekilmiştir. Termal ya da multispektral sensörlerle alınan veriler bulunmadığından gece koşullarında ya da düşük ışıklı senaryolarda model performansı test edilememiştir.
- **İrtifa Kısıtlaması:** Hayvanların rahatsız olmaması amacıyla görüntüler yüksekten (10–30 m) alınmıştır. Ancak bu, nesnelerin küçülmesine ve bazı detayların kaybolmasına neden olmuştur. Düşük irtifalı örneklerin azlığı, karşılaştırmalı analizleri sınırlamıştır.
- **Tek Cihaz Kullanımı:** Çalışmamızda, tüm görüntüler yalnızca SJRC F22S Pro 4K kameralı bir İHA'dan elde edilmiştir. Farklı sensörlere ve çözünürlüklere sahip cihazların performansa etkisi bu çalışmada incelenememiştir.
- **Coğrafi Kapsam Darlığı:** Tüm veriler Batman ili Kozluk ilçesinde toplanmıştır. Farklı coğrafi ve iklimsel koşullar altında modellerin nasıl davranacağı da analiz edilmelidir.

## 5.2. Gelecek Çalışmalar İçin Öneriler

Bu çalışmanın sonuçları, gelecekte yapılacak araştırmalara temel oluşturabilir. Diğer yandan gelecek çalışmalarda aşağıdaki önerilen modellerin daha kapsamlı ve başarılı sonuçlar üretmesine destek olacaktır:

- **İrtifa ve Nesne Boyutuna Duyarlı Modeller:** Model mimarileri, görüntülerin çekildiği irtifayı ve nesne boyutunu dikkate alarak

geliştirilmeli, bu parametreleri doğrudan analiz eden “ölçek duyarlı” modeller tasarlanabilir.

- **Adaptif Veri Artırma Yaklaşımları:** Her veri artırma yöntemi her görüntü türü için uygun değildir. Gelecek çalışmalarda, görüntü irtifasına ve ortam karmaşıklığına göre dinamik olarak en uygun artırma tekniğini seçen adaptif sistemler üzerine odaklanmalıdır.
- **Veri Setinin Büyütülmesi ve Çeşitlendirilmesi:** Farklı bölgelerden, farklı hava ve ışık koşullarında görüntüler toplanmalı; farklı İHA modelleriyle yapılan çekimler karşılaştırılmalı, sentetik veri üretimi (örneğin GAN’lar, Difüzyon modelleri ile) ile veri çeşitliliği artırılmalıdır.
- **YOLO11 ve Benzeri Gelişmiş Modellerin Optimizasyonu:** Bu modeller, küçük nesnelere için daha uygun çapa (anchor) boyutları ve dikkat mekanizmalarıyla yeniden yapılandırılmalıdır.
- **Hibrit Model Tasarımı ve Transfer Öğrenme:** Cascade R-CNN’in detaylı tespiti ile YOLOv8m’nin hızını birleştiren hibrit mimariler geliştirilebilir. Ayrıca, daha büyük veri setlerinde eğitilmiş modeller, transfer öğrenme ile küçük tarımsal veri setlerine uyarlanarak daha başarılı sonuçlar elde edilebilir.

## KAYNAKLAR

- Alanezi, M. A., Shahriar, M. S., Hasan, Md. B., Ahmed, S., Sha'aban, Y. A., & Bouchekara, H. R. E. H. (2022). Livestock management with unmanned aerial vehicles: A review. *IEEE Access*, *10*, 45001-45028. <https://doi.org/10.1109/ACCESS.2022.3168295>
- Aquilani, C., Confessore, A., Bozzi, R., Sirtori, F., & Pugliese, C. (2022). Review: Precision livestock farming technologies in pasture-based livestock systems. *Animal*, *16*(1), 100429. <https://doi.org/10.1016/j.animal.2021.100429>
- Bailey, D. W., Trotter, M. G., Tobin, C., & Thomas, M. G. (2021). Opportunities to apply precision livestock management on rangelands. *Frontiers in Sustainable Food Systems*, *5*. <https://doi.org/10.3389/fsufs.2021.611915>
- Banhazi, T. M., Lehr, H., Black, J. L., Crabtree, H., Schofield, P., Tscharke, M., & Berckmans, D. (2012). Precision livestock farming: An international review of scientific and commercial aspects. *International Journal of Agricultural and Biological Engineering*, *5*(3), 1-9. <https://doi.org/10.25165/ijabe.v5i3.599>
- Bengio, Y. (2009). Learning deep architectures for AI. *Foundations and Trends® in Machine Learning*, *2*(1), 1-127. <https://doi.org/10.1561/22000000006>
- Bhatkalkar, B. J., Reddy, D. R., Prabhu, S., & Bhandary, S. V. (2020). Improving the performance of convolutional neural network for the segmentation of optic disc in fundus images using attention gates and conditional random fields. *IEEE Access*, *8*, 29299-29310. <https://doi.org/10.1109/ACCESS.2020.2972318>
- Bochkovski, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). YOLOv4: Optimal speed and accuracy of object detection. *arXiv:2004.10934*. <https://doi.org/10.48550/arXiv.2004.10934>
- Cai, Z., & Vasconcelos, N. (2018). Cascade R-CNN: Delving into high quality object detection. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 6154-6162.

[https://openaccess.thecvf.com/content\\_cvpr\\_2018/html/Cai\\_Cascade\\_R-CNN\\_Delving\\_CVPR\\_2018\\_paper.html](https://openaccess.thecvf.com/content_cvpr_2018/html/Cai_Cascade_R-CNN_Delving_CVPR_2018_paper.html)

- Carranza-García, M., Torres-Mateo, J., Lara-Benítez, P., & García-Gutiérrez, J. (2021). On the performance of one-stage and two-stage object detectors in autonomous vehicles using camera data. *Remote Sensing*, *13*(1), 89. <https://doi.org/10.3390/rs13010089>
- DeVries, T., & Taylor, G. W. (2017). Improved regularization of convolutional neural networks with cutout. *arXiv:1708.04552*. <https://doi.org/10.48550/arXiv.1708.04552>
- Frid-Adar, M., Diamant, I., Klang, E., Amitai, M., Goldberger, J., & Greenspan, H. (2018). GAN-Based Synthetic Medical Image Augmentation for Increased CNN Performance in Liver Lesion Classification. *Neurocomputing*, *321*, 321-331. <https://arxiv.org/abs/1803.01229>
- Gong, H., Mu, T., Li, Q., Dai, H., Li, C., He, Z., Wang, W., Han, F., Tuniyazi, A., Li, H., Lang, X., Li, Z., & Wang, B. (2022). Swin-transformer-enabled YOLOv5 with attention mechanism for small object detection on satellite images. *Remote Sensing*, *14*(12), 2861. <https://doi.org/10.3390/rs14122861>
- Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative Adversarial Networks. *Advances in Neural Information Processing Systems*, *27*, 2672-2680. <https://arxiv.org/abs/1406.2661>
- Gui, J., Sun, Z., Wen, Y., Tao, D., & Ye, J. (2021). A Review on Generative Adversarial Networks: Algorithms, Theory, and Applications. *IEEE Transactions on Knowledge and Data Engineering*, *35*(4), 3313-3332. <https://ieeexplore.ieee.org/document/9350683>
- Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., & Courville, A. (2017). Improved Training of Wasserstein GANs. *Advances in Neural Information Processing Systems*, *30*, 5767-5777. <https://arxiv.org/abs/1704.00028>
- Han, C., Hayashi, H., Rundo, L., et al. (2018). GAN-Based Synthetic Brain MR Image Generation. *IEEE 15th International Symposium on Biomedical Imaging (ISBI)*, 734-738. <https://ieeexplore.ieee.org/document/8363678>

- Han, L., Li, N., Li, J., Gao, B., & Niu, D. (2025). SA-FPN: Scale-aware attention-guided feature pyramid network for small object detection on surface defect detection of steel strips. *Measurement*, *249*, 117019. <https://doi.org/10.1016/j.measurement.2025.117019>
- Jiang, M., Rao, Y., Zhang, J., & Shen, Y. (2020). Automatic behavior recognition of group-housed goats using deep learning. *Computers and Electronics in Agriculture*, *177*, 105706. <https://doi.org/10.1016/j.compag.2020.105706>
- Kaur, P., Khehra, B. S., & Mavi, Er. B. S. (2021). Data augmentation for object detection: A review. *2021 IEEE International Midwest Symposium on Circuits and Systems (MWSCAS)*, 537-543. <https://doi.org/10.1109/MWSCAS47672.2021.9531849>
- Korkmaz, A., Agdas, M. T., Kosunalp, S., Iliev, T., & Stoyanov, I. (2024). Detection of threats to farm animals using deep learning models: A comparative study. *Applied Sciences*, *14*(14), 6098. <https://doi.org/10.3390/app14146098>
- Li, C., Li, L., Jiang, H., Weng, K., Geng, Y., Li, L., Ke, Z., Li, Q., Cheng, M., Nie, W., Li, Y., Zhang, B., Liang, Y., Zhou, L., Xu, X., Chu, X., Wei, X., & Wei, X. (2022). YOLOv6: A single-stage object detection framework for industrial applications. *arXiv:2209.02976*. <https://doi.org/10.48550/arXiv.2209.02976>
- Li, Z., Peng, C., Yu, G., Zhang, X., Deng, Y., & Sun, J. (2017). Light-head R-CNN: In defense of two-stage object detector. *arXiv:1711.07264*. <https://doi.org/10.48550/arXiv.1711.07264>
- Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. *2017 IEEE International Conference on Computer Vision (ICCV)*, 2999-3007. <https://doi.org/10.1109/ICCV.2017.324>
- Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2020). Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *42*(2), 318-327. <https://doi.org/10.1109/TPAMI.2018.2858826>
- Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X., & Pietikäinen, M. (2020). Deep learning for generic object detection: A survey. *International*

- Journal of Computer Vision*, 128(2), 261-318.  
<https://doi.org/10.1007/s11263-019-01247-4>
- Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). SSD: Single shot multibox detector. İçinde B. Leibe, J. Matas, N. Sebe, & M. Welling (Ed.), *ECCV: European Conference on Computer Vision 2016* (ss. 21-37). Springer International Publishing.  
[https://doi.org/10.1007/978-3-319-46448-0\\_2](https://doi.org/10.1007/978-3-319-46448-0_2)
- Mirza, M., & Osindero, S. (2014). Conditional Generative Adversarial Nets. arXiv preprint. <https://arxiv.org/abs/1411.1784>
- Mufford, J. T., Reudink, M. W., Rakobowchuk, M., Carlyle, C. N., & Church, J. S. (2022). Using unmanned aerial vehicles to record behavioral and physiological indicators of heat stress in cattle on feedlot and pasture. *Canadian Journal of Animal Science*, 102(1), 1-8.  
<https://doi.org/10.1139/cjas-2020-0125>
- Qiao, Y., Truman, M., & Sukkarieh, S. (2019). Cattle segmentation and contour extraction based on Mask R-CNN for precision livestock farming. *Computers and Electronics in Agriculture*, 165, 104958.  
<https://doi.org/10.1016/j.compag.2019.104958>
- Radford, A., Metz, L., & Chintala, S. (2016). Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. International Conference on Learning Representations (ICLR).  
<https://arxiv.org/abs/1511.06434>
- Ramachandran, A., & Sangaiah, A. K. (2021). A review on object detection in unmanned aerial vehicle surveillance. *International Journal of Cognitive Computing in Engineering*, 2, 215-228.  
<https://doi.org/10.1016/j.ijcce.2021.11.005>
- Redmon, J., & Farhadi, A. (2018). YOLOv3: An incremental improvement. *arXiv:1804.02767*. <https://doi.org/10.48550/arXiv.1804.02767>
- Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems* 28, 28, 91-99.

<https://proceedings.neurips.cc/paper/2015/hash/14bfa6bb14875e45bba028a21ed38046-Abstract.html>

- Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, 61(2015), 85-117. <https://doi.org/10.1016/j.neunet.2014.09.003>
- Scott, S. D., Abbas, Z. J., Ellid, F., Dykhne, E.-H., Islam, M. M., Ayad, W., Kacmorova, K., Tulpan, D., & Gong, M. (2024). Systematic literature review of vision-based approaches to outdoor livestock monitoring with lessons from wildlife studies. *arXiv:2410.05041*. <https://doi.org/10.48550/arXiv.2410.05041>
- Sevli, O. (2023). Tarım 4.0 ölçeğinde bir dijital tarım uygulaması: Çiftlik izleme ve yönetim sistemi. *International Journal of Sustainable Engineering and Technology*, 7(2), 105-116.
- Shalini, R., & Gopi, V. P. (2024). Multiresolution cascaded attention U-Net for localization and segmentation of optic disc and fovea in fundus images. *Scientific Reports*, 14(1), 23107. <https://doi.org/10.1038/s41598-024-73493-7>
- Singhal, P., Walambe, R., Ramanna, S., & Kotecha, K. (2023). Domain Adaptation: Challenges, methods, datasets, and applications. *IEEE Access*, 11, 6973-7020. <https://doi.org/10.1109/ACCESS.2023.3237025>
- Tremblay, J., Prakash, A., Acuna, D., Brophy, M., Jampani, V., Anil, C., To, T., Cameracci, E., Boochoon, S., & Birchfield, S. (2018). Training deep networks with synthetic data: Bridging the reality gap by domain randomization. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 969-977. [https://openaccess.thecvf.com/content\\_cvpr\\_2018\\_workshops/w14/html/Tremblay\\_Training\\_Deep\\_Networks\\_CVPR\\_2018\\_paper.html](https://openaccess.thecvf.com/content_cvpr_2018_workshops/w14/html/Tremblay_Training_Deep_Networks_CVPR_2018_paper.html)
- Wang, C.-Y., Bochkovskiy, A., & Liao, H.-Y. M. (2023). YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 7464-7475. [https://openaccess.thecvf.com/content/CVPR2023/html/Wang\\_YOLOv7\\_](https://openaccess.thecvf.com/content/CVPR2023/html/Wang_YOLOv7_)

Trainable\_Bag-of-Freebies\_Sets\_New\_State-of-the-Art\_for\_Real-Time\_Object\_Detectors\_CVPR\_2023\_paper.html

- Wang, X., He, N., Hong, C., Wang, Q., & Chen, M. (2023). Improved YOLOX-X based UAV aerial photography object detection algorithm. *Image and Vision Computing*, 135, 104697. <https://doi.org/10.1016/j.imavis.2023.104697>
- Wong, S. C., Gatt, A., Stamatescu, V., & McDonnell, M. D. (2016). Understanding data augmentation for classification: When to warp? *2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, 1-6. <https://doi.org/10.1109/DICTA.2016.7797091>
- Wu, X., Li, W., Hong, D., Tao, R., & Du, Q. (2022). Deep learning for unmanned aerial vehicle-based object detection and tracking: A survey. *IEEE Geoscience and Remote Sensing Magazine*, 10(1), 91-124. <https://doi.org/10.1109/MGRS.2021.3115137>
- Yang, J., Liu, X., & Liu, Z. (2023). Attention-guided feature fusion for small object detection. *2023 IEEE International Conference on Imaging Systems and Techniques (IST)*, 1-6. <https://doi.org/10.1109/IST59124.2023.10355735>
- Ying, X., Wang, Q., Li, X., Yu, M., Jiang, H., Gao, J., Liu, Z., & Yu, R. (2019). Multi-attention object detection model in remote sensing images based on multi-scale. *IEEE Access*, 7, 94508-94519. <https://doi.org/10.1109/ACCESS.2019.2928522>
- Yousefi, D. B. M., Rafie, A. S. M., Al-Haddad, S. A. R., & Azrad, S. (2022). A systematic literature review on the use of deep learning in precision livestock detection and localization using unmanned aerial vehicles. *IEEE Access*, 10, 80071-80091. <https://doi.org/10.1109/ACCESS.2022.3194507>
- Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J., & Yoo, Y. (2019). CutMix: Regularization strategy to train strong classifiers with localizable features. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 6023-6032. [https://openaccess.thecvf.com/content\\_ICCV\\_2019/html/Yun\\_CutMix\\_Regularization\\_Strategy\\_to\\_Train\\_Strong\\_Classifiers\\_With\\_Localizable\\_Features\\_ICCV\\_2019\\_paper.html](https://openaccess.thecvf.com/content_ICCV_2019/html/Yun_CutMix_Regularization_Strategy_to_Train_Strong_Classifiers_With_Localizable_Features_ICCV_2019_paper.html)

- Zhang, H., Cisse, M., Dauphin, Y. N., & Lopez-Paz, D. (2018). mixup: Beyond empirical Risk minimization. *arXiv:1710.09412*.  
<https://doi.org/10.48550/arXiv.1710.09412>
- Zhang, H., Zhang, S., & Zou, R. (2024). Select-Mosaic: Data augmentation method for dense small object scenes. *arXiv:2406.05412*.  
<https://doi.org/10.48550/arXiv.2406.05412>
- Zhang, J., Li, C., Rahaman, M. M., Yao, Y., Ma, P., Zhang, J., Zhao, X., Jiang, T., & Grzegorzec, M. (2022). A comprehensive review of image analysis methods for microorganism counting: From classical image processing to deep learning approaches. *Artificial Intelligence Review*, 55(4), 2875-2944.  
<https://doi.org/10.1007/s10462-021-10082-4>
- Zhao, B., Wu, Y., Guan, X., Gao, L., & Zhang, B. (2021). An improved aggregated-mosaic method for the sparse object detection of remote sensing imagery. *Remote Sensing*, 13(13), 2602. <https://doi.org/10.3390/rs13132602>
- Zhao, Z.-Q., Zheng, P., Xu, S.-T., & Wu, X. (2019). Object detection with deep learning: A review. *IEEE Transactions on Neural Networks and Learning Systems*, 30(11), 3212-3232.  
<https://doi.org/10.1109/TNNLS.2018.2876865>
- Zhu, X., Su, W., Lu, L., Li, B., Wang, X., & Dai, J. (2021). Deformable DETR: Deformable transformers for end-to-end object detection. *arXiv:2010.04159*. <https://doi.org/10.48550/arXiv.2010.04159>
- Zoph, B., Vasudevan, V., Shlens, J., & Le, Q. V. (2018). Learning transferable architectures for scalable image recognition. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 8697-8710.  
<https://doi.org/10.1109/CVPR.2018.00907>

## **ÖZGEÇMİŞ**

### **KİŞİSEL BİLGİLER**

Adı Soyadı : Çetin YALÇIN

Doğum Yeri ve Tarihi :

### **EĞİTİM DURUMU**

Lisans Öğrenimi : Kocaeli Üniversitesi Bilgisayar Mühendisliği

Yüksek Lisans Öğrenimi : Hakkari Üniversitesi Elektrik-Elektronik Mühendisliği

Yabancı Diller : İngilizce

### **BİLİMSEL FAALİYETLERİ**

A) Bildiriler