



**EVRIŞİMSEL SİNİR AĞLARI İLE İHA TESPİTİ: YAPAY ZEKÂ
TEKNOLOJİLERİNİN SİVİL İHA GÜVENLİĞİ İÇİN ROLÜ**

YÜKSEK LİSANS TEZİ

Ömer TÜRKMEN

Danışman
Prof. Dr. Yüksel OĞUZ

ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

Ekim 2024

AFYON KOCATEPE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

EVRIŞİMSEL SINIR AĞLARI İLE İHA TESPİTİ: YAPAY ZEKÂ
TEKNOLOJİLERİNİN SİVİL İHA GÜVENLİĞİ İÇİN ROLÜ

Ömer TÜRKMEN

Danışman
Prof. Dr. Yüksel OĞUZ

ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

Ekim 2024

TEZ ONAY SAYFASI

Ömer TÜRKMEN tarafından hazırlanan “Tez Onay Sayfası” adlı tez çalışması lisansüstü eğitim ve öğretim yönetmeliğinin ilgili maddeleri uyarınca 25/10/2024 tarihinde aşağıdaki jüri tarafından **oy birliği** ile Afyon Kocatepe Üniversitesi Fen Bilimleri Enstitüsü **Elektrik-Elektronik Mühendisliği Anabilim Dalı’nda YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Danışman : Prof. Dr. Yüksel OĞUZ

Başkan : Prof. Dr. Yüksel OĞUZ
Afyon Kocatepe Üniversitesi, Teknoloji Fakültesi

Üye : Doç. Dr. İsmail KOYUNCU
Afyon Kocatepe Üniversitesi, Teknoloji Fakültesi

Üye : Doç. Dr. Ferzan KATIRCIOĞLU
Düzce Üniversitesi, Mühendislik Fakültesi

Afyon Kocatepe Üniversitesi
Fen Bilimleri Enstitüsü Yönetim Kurulu’nun
..... /..... /..... tarih ve
..... sayılı kararıyla onaylanmıştır.

.....
Prof. Dr. Bekir YALÇIN
Enstitü Müdürü

BİLİMSEL ETİK BİLDİRİM SAYFASI
Afyon Kocatepe Üniversitesi

Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmada;

- Tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- Görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,
- Başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
- Atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,
- Kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- Ve bu tezin herhangi bir bölümünü bu üniversite veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı

beyan ederim.

25 / 10 / 2024

Ömer TÜRKMEN

ÖZET

Yüksek Lisans Tezi

EVRIŞİMSEL SİNİR AĞLARI İLE İHA TESPİTİ: YAPAY ZEKÂ TEKNOLOJİLERİNİN SİVİL İHA GÜVENLİĞİ İÇİN ROLÜ

Ömer TÜRKMEN

Afyon Kocatepe Üniversitesi

Fen Bilimleri Enstitüsü

Elektrik-Elektronik Mühendisliği Anabilim Dalı

Danışman: Prof. Dr. Yüksel OĞUZ

Bu araştırmada, son zamanlarda sivil insansız hava araçlarının (İHA'ların) kullanımının giderek artmasından dolayı savunma ve güvenlik açısından önemli bir endişe kaynağı haline gelen İHA'ların derin öğrenme ile tespit işlemi gerçekleştirilmiştir. Sabit kanatlı ve döner kanatlı sivil İHA'ların çeşitli görsellerinden oluşan bir veri seti oluşturulmuştur. Daha sonra veri setindeki bu görseller etiketlenerek derin öğrenme modellerine uygun formata dönüştürülmüştür. Literatürde sıkça kullanılan YOLOv5, YOLOv7, YOLOv8 ve güncel olan YOLOv10 derin öğrenme modelleri kullanılarak eğitimler gerçekleştirilmiştir. Eğitimler sonucunda YOLOv8 modelinin %97 başarı oranıyla diğer modellere göre daha iyi performans gösterdiği ortaya konulmuştur. PyQt5 kütüphanesi ve Qt Designer kullanılarak Windows masaüstü uygulaması geliştirilmiştir. Geliştirilen bu uygulama ile en iyi başarı oranına sahip model birleştirilerek sivil İHA tespiti gerçekleştirilebilen bir sistem tasarlanmıştır. Yapılan çalışma ile sivil İHA'ların tespit edilmesi için derin öğrenme modellerinin etkin bir şekilde kullanılabileceği gösterilmektedir. Elde edilen sonuçlar, güvenlik açısından önemli olan sivil hava araçlarının tespitinde kullanılabilecek etkili bir yöntem sunmaktadır.

2024, ix + 59 sayfa

Anahtar Kelimeler: Derin öğrenme, YOLO, İHA, Nesne Tanıma, Görüntü İşleme

ABSTRACT

M.Sc. Thesis

UAV DETECTION WITH CONVOLUTIONAL NEURAL NETWORKS: THE ROLE OF ARTIFICIAL INTELLIGENCE TECHNOLOGIES FOR CIVIL UAV SAFETY

Ömer TÜRKMEN

Afyon Kocatepe University

Graduate School of Natural and Applied Sciences

Department of Electrical and Electronics Engineering

Supervisor: Prof. Dr. Yüksel OĞUZ

In this research, the detection of UAV, which have recently become an important source of concern in terms of defense and security due to the increasing use of civilian unmanned aerial vehicles, has been carried out with deep learning. A dataset consisting of various images of fixed-wing and rotary-wing civilian UAVs was created. These images were then labeled and converted into a format suitable for deep learning models. Trainings were conducted using YOLOv5, YOLOv7, YOLOv8 and the most recent YOLOv10 deep learning models that are frequently used in the literature. As a result of the trainings, it was revealed that the YOLOv8 model performed better than the other models with a 97% success rate. A Windows application was developed using PyQt5 library and Qt Designer. With this application, a system that can detect civilian UAVs was designed by combining the model with the best success rate. This study shows that deep learning models can be used effectively to detect civilian drones. The results obtained provide an effective method that can be used in the detection of civilian aerial vehicles that are important for security.

2024, ix + 59 pages

Keywords: Deep learning, YOLO, UAV, Object Detection, Image Processing

TEŞEKKÜR

Bu araştırmanın konusu, deneysel çalışmaların yönlendirilmesi, sonuçların değerlendirilmesi ve yazımı aşamasında yapmış olduğu büyük katkılarından dolayı tez danışmanım Sayın Prof. Dr. Yüksel OĞUZ'a, lisans eğitimimden itibaren beni hep destekleyen, bu çalışmam kapsamında araştırma ve yazım süresince yardımlarını esirgemeyen hocalarım Sayın Öğr. Gör. Hasan AKKOÇ'a, Sayın Dr. Öğr. Üyesi Ertuğrul ERGÜN'e ve Sayın Doç. Dr. Uçman ERGÜN'e, tüm bu süreçte yanımda olup, her konuda desteğini esirgemeyen, öneri ve eleştirileriyle yardımlarını gördüğüm Sayın Arş. Gör. İsmail KAYADİBİ'ne, hocalarıma ve arkadaşlarıma teşekkür ederim.

Bu araştırma boyunca maddi ve manevi desteklerini esirgemeyen aileme sonsuz teşekkür ederim.

Ömer TÜRKMEN
Afyonkarahisar 2024

İÇİNDEKİLER DİZİNİ

	Sayfa
ÖZET	i
ABSTRACT	ii
TEŞEKKÜR	iii
İÇİNDEKİLER DİZİNİ.....	iv
SİMGELER ve KISALTMALAR DİZİNİ	vi
ŞEKİLLER DİZİNİ	vii
ÇİZELGELER DİZİNİ.....	viii
RESİMLER DİZİNİ	ix
1. GİRİŞ.....	1
2. LİTERATÜR BİLGİLERİ	4
3. MATERYAL ve METOT	8
3.1 Yapay Zeka.....	8
3.1.1 Evrişimsel Sinir Ağı (CNN)	9
3.1.1.1 Giriş Katmanı	10
3.1.1.2 Konvolüsyon Katmanı	11
3.1.1.3 Aktivasyon Katmanları	12
3.1.1.4 Havuzlama Katmanları	13
3.1.1.5 Tam Bağlantılı Katmanlar	14
3.1.1.6 Çıkış Katmanı	15
3.1.2 CNN Mimari Türleri.....	16
3.2 SAHI (Dilimleme Destekli Hiper İnfersan) Algoritması	23
3.3 Veri Seti.....	24
3.4 Yapay Zeka ile Eğitim.....	25
3.5 PyQt5.....	26
3.6 Qt Designer.....	27
3.7 PyQt5 ve Qt Designer ile Masaüstü Uygulamasının Geliştirilmesi	28
4. BULGULAR	31
4.1 Performans Metrikleri	31
4.2 Model Test Sonuçları	33
4.2.1 En Başarılı Modelin Test Sonuçları.....	34
4.2.2 SAHI Algoritması ile Test Sonuçları.....	35
4.2.3 PyQt ve Qt Designer ile Geliştirilen Uygulamada Test Sonuçları	37

5. TARTIřMA ve SONUÇ	42
6. KAYNAKLAR.....	45



SİMGELER ve KISALTMALAR DİZİNİ

Simgeler

%	Yüzde
~	Yaklaşık

Kısaltmalar

AI	Yapay Zeka
API	Uygulama Programlama Arayüzü
CNN	Konvolüsyonel Sinir Ağı
FN	Yanlış Negatif
FP	Yanlış Pozitif
FPN	Özellik Piramidi Ağı
FPS	Saniye Başına Kare Sayısı
GPU	Grafik İşlem Birimi
GUI	Grafiksel Kullanıcı Arayüzü
İHA	İnsansız Hava Aracı
JPG	Birleşik Fotoğraf Uzmanları Grubu
M	Milyon
mAP	Ortalama Hassasiyet
MFnet	MultiFeatureNet
MFNet-FA	MultiFeatureNet-Feature Attention
MNIST	Karışık Ulusal Standartlar ve Teknoloji Enstitüsü
NMS	Maksimum Olmayanları Bastırma
ReLU	Düzeltilici Doğrusal Birim
ResNet	Kalıntı Ağ
RF	Radyo Frekansı
RTD-Net	Gerçek Zamanlı Nesne Tespit Ağı
SAHI	Dilimleme Destekli Hiper İnfersans
SSD	Tek Atışlı Çoklu Kutu Dedektörü
TN	Doğru Negatif
TP	Doğru Pozitif
VGG	Görsel Geometri Grubu
XML	Genişletilebilir İşaretleme Dili
YOLO	Yalnızca Bir Kez Bak

ŞEKİLLER DİZİNİ

	Sayfa
Şekil 1.1 Tez çalışması kapsamında oluşturulan genel mimarinin yapısı	3
Şekil 3.1 CNN mimari yapısı	9
Şekil 3.2 Konvolüsyon katmanı yapısı	12
Şekil 3.3 Havuzlama katmanı yapısı	14
Şekil 3.4 LeNet-5 mimarisi (İnt. Kyn. 2)	16
Şekil 3.5 AlexNet mimari yapısı	17
Şekil 3.6 VGGNet mimari yapısı	17
Şekil 3.7 GoogLeNet mimari yapısı	18
Şekil 3.8 Resnet mimari yapısı	18
Şekil 3.9 DenseNet yoğun katman yapıları (İnt. Kyn. 2)	19
Şekil 3.10 Mobilenet mimari yapısı	19
Şekil 3.11 EfficientNet mimari yapısı	20
Şekil 3.12 YOLO mimarisi yapısı	21
Şekil 3.13 YOLO mimarisi ile tespit adımları	22
Şekil 3.14 Eğitim işlemi sonucunda elde edilen parametreler	26
Şekil 3.15 Qt Designer arayüzü	28
Şekil 3.16 Geliştirilen uygulamanın giriş arayüzü	29
Şekil 3.17 Geliştirilen uygulamanın tespit arayüzü	29

ÇİZELGELER DİZİNİ

	Sayfa
Çizelge 3.1 YOLO derin öğrenme mimarilerinin versiyon farklılıkları	23
Çizelge 4.1 Tez çalışmasında karşılaştırılan mimarilerin eğitim sonuçları.....	33
Çizelge 5.1 Derin öğrenme modelleri ile İHA tespit çalışmalarının karşılaştırılması ...	43



RESİMLER DİZİNİ

	Sayfa
Resim 3.1 SAHI algoritması ile tespit örneği karşılaştırması	24
Resim 3.2 Sivil İHA tespiti amacı ile oluşturulan veri seti	25
Resim 3.3 Roboflow web sitesi ile Sivil İHA tespiti için oluşturulan veri setindeki görsellerin etiketleme işlemi	25
Resim 4.1 YOLOv8 derin öğrenme sonucunda elde edilen ağırlık dosyasının görseller üzerindeki test görüntüleri	34
Resim 4.2 Sürü İHA'lara ait test görseli	36
Resim 4.3 SAHI algoritması eklenmeden YOLOv8_S modelinde test işlemi	36
Resim 4.4 SAHI algoritması eklenerek gerçekleştirilen test işlemi	37
Resim 4.5 Masaüstü uygulamasında döner kanat formdaki İHA'ya ait test sonucu 1 ..	38
Resim 4.6 Masaüstü uygulamasında döner kanat formdaki İHA'ya ait test sonucu 2 ..	38
Resim 4.7 Masaüstü uygulamasında döner kanat formdaki İHA'ya ait test sonucu 3 ..	39
Resim 4.8 Masaüstü uygulamasında sabit kanat formdaki İHA'ya ait test sonucu 1	39
Resim 4.9 Masaüstü uygulamasında sabit kanat formdaki İHA'ya ait test sonucu 2	40
Resim 4.10 Masaüstü uygulamasında sabit kanat formdaki İHA'ya ait test sonucu 3 ..	40
Resim 4.11 Masaüstü uygulamasında sivil İHA olmadan gerçekleştirilen test sonucu.	41

1. GİRİŞ

İnsanoğlu tarih boyunca sürekli olarak kendini geliştirerek bilimde önemli ilerlemeler kaydetmiştir. Bu ilerlemeler, çeşitli alanlarda ortaya çıkan teknolojik başarılarla birlikte gelmiştir. Örneğin, elektrik-elektronik, yapay zekâ ve insansız araç teknolojisi gibi alanlarda önemli adımlar atılmıştır (Kahveci ve Can 2017, Tang vd. 2024, Wan vd. 2024). Bu teknolojik ilerlemelerden biri de insansız hava araçlarıdır (İHA) (Rabiu vd. 2024). İHA'lar, insansız veya uzaktan pilotlu olarak kullanılabilen, genellikle havada bazen de su içerisinde uçuş yapabilen araçlardır. Genellikle uzaktan kumanda veya önceden programlanmış yollarla kontrol edilebilen bu araçlar, çeşitli alanlarda yaygın olarak kullanılmaktadır. İHA'ların çok yönlü kullanım amacı, savunma sanayinden sağlığa, eğitimden turizme kadar geniş bir yelpazede yer almaktadır (Özer ve Türkmen 2024, Peng vd. 2024). Savunma sanayisinde, İHA'lar özellikle askeri operasyonlar esnasında ya da öncesinde keşiflerde ve kamikaze saldırılarda kullanılmaktadır (Kunertova 2023). Sivil uygulamalarda ise tarım, madencilik, ormancılık, arama-kurtarma operasyonları ve medya uygulamaları gibi çeşitli alanlarda etkili bir şekilde kullanılmaktadır (Su vd. 2023). Ayrıca, İHA'lar havadan gözetim ve haritalama, bilimsel araştırmalar ve hava durumu tahminleri gibi birçok alanda da önemli roller üstlenmektedirler (Prabhu vd. 2024). Bu sayede, insansız hava araçları, modern teknolojinin bir ürünü olarak, insanlığın farklı ihtiyaçlarına cevap veren çok yönlü araçlar haline gelmiştir.

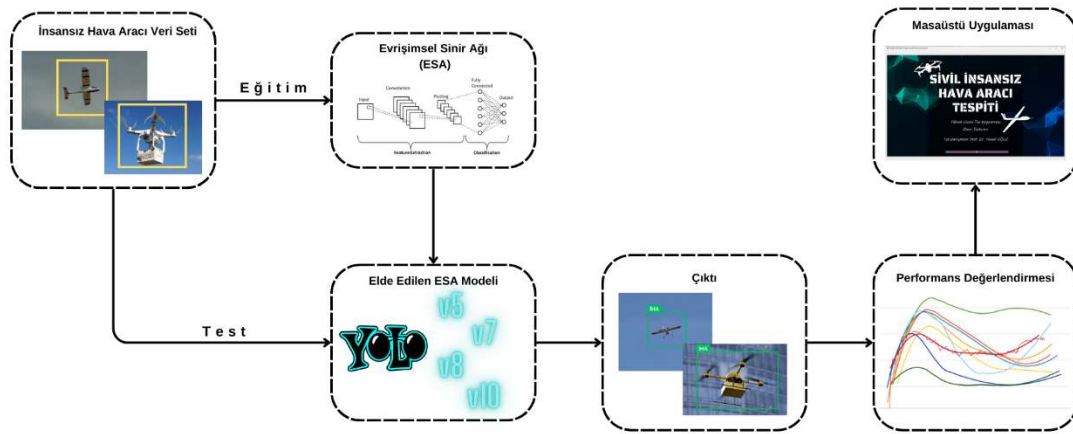
Sivil insansız hava araçlarının, başlangıçta barışçıl amaçlar için tasarlanmış olmalarına rağmen, zaman içinde sivil kullanımdan askeri operasyonlara kadar roller üstlenebilme kabiliyetleri, uluslararası alanlarda bazı soruları gündeme getirmiştir. Özellikle sivil İHA'ların, gelişmiş teknolojileri ve gözetleme yetenekleri nedeniyle, savaş alanlarında casusluk veya istihbarat toplama amaçlarıyla kullanılma potansiyeli bulunmaktadır (Chávez ve Swed, 2023). Bu durum, sivil hava araçlarının askeri operasyonlarda kullanımına dair tartışmalara yol açmıştır. Ayrıca, bazı ülkeler sivil İHA'ları, askeri kapasitelerini artırmak ve stratejik avantaj elde etmek amacıyla kullanmaya başlamıştır. Bu durum, sivil hava araçlarının sadece barışçıl amaçlarla kullanılması gerektiği düşüncesine meydan okuyan bir gelişmedir. Askeri operasyonlarda kullanılan sivil İHA'lar, uluslararası ilişkiler ve güvenlik politikaları açısından yeni sorunlar ortaya çıkarmaktadır.

İHA'lar üzerine yerleştirilebilecek patlayıcılar konusu da, güvenlik ve ulusal savunma açısından önemli bir meseledir. Bu tür bir kullanım, terör saldırıları, askeri operasyonlar (Udenau vd. 2016) veya diğer tehlikeli senaryolar için potansiyel bir tehdit oluşturmaktadır. Bu bağlamda, İHA'lar üzerindeki patlayıcıların kullanımıyla ilgili çeşitli güvenlik endişeleri bulunmaktadır. Bu endişelerden birisi, terörist grupların veya düşman devletlerin İHA'ları kullanarak saldırılarını gerçekleştirmesi olasılığıdır. İHA'lar, küçük boyutları ve geniş kullanım alanları nedeniyle potansiyel saldırgan olmak için iyi bir seçenek olmaktadır (King 2024). Bu durum, sivil yerleşim alanları, askeri üsler veya altyapı tesisleri gibi hassas bölgelerde büyük zararlara neden olabilir. Ayrıca, en önemlisi İHA'lar üzerine yerleştirilecek patlayıcıların kullanımı, mevcut hava savunma sistemlerini aşma potansiyeli taşımaktadır. Gelişmiş manevra kabiliyetleri sayesinde, bu tür İHA'lar, geleneksel hava savunma sistemlerine karşı daha zorlu bir tehdit oluşturmaktadır. Bu durum, savunma ve güvenlik politikalarının gözden geçirilmesini ve uygun önlemlerin alınmasını gerektirmektedir. Elektronik karıştırma sistemleri, radar sistemleri ve saldırı tespit teknolojileri gibi savunma mekanizmaları, bu tür tehditlere karşı daha etkili bir koruma sağlamak amacıyla geliştirilmektedir. Sonuç olarak, İHA'lar üzerine yerleştirilebilecek patlayıcı olasılığına karşı etkili önlemler almak hem sivil hem de askeri alanlarda potansiyel zararları minimum seviyeye indirmek için önemlidir.

Yapay zeka'nın İHA'lar tespiti konusundaki rolü, modern güvenlik paradigmasında önemli bir dönüşüme yol açmıştır (Rostami vd. 2024). Yapay zekâ, büyük veri setlerini analiz etme ve karmaşık yapıları çözme yetenekleriyle, İHA'ların tespitini ve sınıflandırılmasını hızlı, etkili ve güvenilir bir şekilde gerçekleştirmektedir (Al-Iqubaydhi vd. 2024). Bu sayede, insanların sürekli gözlem yapma zorunluluğunu ortadan kaldırarak, savunma sistemlerine hızlı ve anlık bir aksiyon imkânı sunmaktadır. Derin öğrenme algoritmaları sayesinde İHA'ların sınıflandırılması ve davranışlarının tespit edilmesi daha yüksek hassasiyet oranlarıyla sağlanmaktadır.

Yapay zekâ, kameralardan elde edilen verileri analiz ederek İHA'ları belirli özelliklerine göre sınıflandırabilmekte ve böylece dost-düşman sınıflandırması yapabilmektedir. Bu özellik, karmaşık ve dinamik hava sahalarında İHA'ları etkili bir şekilde izleme ve yönetme kapasitesine sahip savunma sistemlerinin oluşturulmasını mümkün kılmaktadır.

Ayrıca, yapay zeka destekli tespit sistemleri, sürekli olarak güncellenen veri setleri üzerinden öğrenme yeteneği sayesinde zaman içinde değişen tehditlere kolaylıkla uyum sağlamaktadır. Bu adaptasyon yeteneği, özellikle hava savunma sistemlerinde ortaya çıkabilecek yeni ve olası tehditlere karşı daha etkili bir şekilde savunma yapabilmesini sağlayacaktır. Sonuç olarak, yapay zekâ, İHA'ların tespiti ve sınıflandırılması konusunda modern güvenlik uygulamalarında kritik bir rol oynamaktadır. Derin öğrenme modelleri ve sürekli gelişen algoritmalar, savunma sistemlerini daha akıllı, esnek ve dinamik hale getirerek güvenliğini arttırmada önemli bir araçtır. Bu bağlamda, yapay zekâ teknolojilerinin geliştirilmesi ve hali hazırda kullanılan teknolojilere entegrasyonu, gelecekteki güvenlik tehditlerine karşı daha etkili bir savunma sağlamak adına hayati bir öneme sahiptir.



Şekil 1.1 Tez çalışması kapsamında oluşturulan genel mimarinin yapısı

Bu çalışmada, sivil İHA'ların tespiti için evrişimsel sinir ağına dayanan Şekil 1.1'deki gibi bir tespit sistemi geliştirilmiştir. Bu problem için çeşitli açık kaynak veri setleri bulunmaktadır (İnt. Kyn. 1). Ayrıca veri setlerine çeşitlilik kazandırmak için veri artırma yöntemleri uygulanmıştır. Güncel İHA çeşitlerine ilişkin görüntüler de toplanarak mevcut veri setleri ile birlikte karma bir veri seti oluşturulmuştur. Oluşturulan veri seti üzerinde farklı evrişimsel sinir ağı yöntemleri kullanılarak yapay zeka modelleri eğitilmiş, eğitimin ardından test edilmiş ve modellerin performansı karşılaştırmalı olarak sunulmuştur. Daha sonra belirlenen model İHA tespit sisteminde kullanılarak hem sabit kanat hem de döner kanat formundaki İHA'ların tespit işlemi gerçekleştirilmiştir.

2. LİTERATÜR BİLGİLERİ

Literatürdeki çalışmalar incelendiği zaman farklı çalışmalar görülmektedir. Bu çalışmalara ait örneklerle aşağıda yer verilmiştir.

Aydin ve Singha (2023) yapmış oldukları çalışmada YOLO (Yalnızca Bir Kez Bak) v5'in YOLOv4'e kıyasla drone tespitiindeki performansını karşılaştırmışlardır. Aynı veri kümesi ve parametreleri kullanarak adil bir karşılaştırma yapmışlardır. Orijinal YOLOv5 modelini drone ve kuş olmak üzere iki sınıfa özelleştirmişlerdir. Hiperparametreleri değiştirerek tespit doğruluğunu arttırmışlar ve transfer öğrenimi ile de eğitim işlemini hızlandırmışlardır. Test veri kümesi üzerinde değerlendirme metriklerini hesaplayarak modelin performansını değerlendirmişlerdir. YOLOv5 önceki modele göre daha yüksek performans göstermiştir. Ayrıca iki farklı drone modeli üzerinde tespit hızını test etmişlerdir.

Ashraf vd. (2021) gerçekleştirdikleri çalışmada drone görüntüleri üzerinden it dalaşı yapan droneleri tespit etmişlerdir. Ucuz kameralar ve uzamsal-zamansal ipuçlarını kullanarak diğer uçan dronelerden drone tespiti için iki aşamalı bir yaklaşım sunmuşlardır. Nesne tespiti tabanlı yöntemler yerine, piksel ve renk kanalı bazlı doğru drone tespiti için segmentasyon tabanlı bir yaklaşım kullanmışlardır. Görünüm bilgisini kullanmanın yanı sıra hareket bilgisinden de faydalanmışlardır. ResNet mimarisi kullanılarak yapılan projede %92 oranında başarımla elde etmişlerdir.

Khan vd. (2023) çalışmalarında MultiFeatureNet (MFNet) adlı bir çözüm önermişlerdir. MFNet, özellik temsiliyi geliştirmek için en yoğun özellik haritalarını yakalayan bir yöntemi kullanmaktadır. Ayrıca, giriş özellik haritalarının farklı kanallarını ağırlıklandıran MultiFeatureNet-Feature Attention (MFNet-FA) tekniğini sunmuşlardır. MFNet ve MFNet-FA'nın küçük (S), orta (M) ve büyük (L) versiyonları, farklı ölçeklerdeki tespit gereksinimlerini karşılamak için geliştirilmiştir. Bu çözümler önemli performans iyileştirmeleri sağlamaktadır. MFNet-M, optimum kuş tespiti için %99,8 hassasiyet elde ederken, MFNet-L, İHA tespiti için %97,2 hassasiyet puanı elde etmiştir. MFNet-FA-S hızlı, gerçek zamanlı ve çoklu çıkarım için uygun bir seçenek olarak sunmuşlardır.

Sun vd. (2020) Deeper-SSD (Single Shot MultiBox Detector) derin öğrenme modelini kullanarak İHA tespit çalışması gerçekleştirmişlerdir. Küçük nesneleri tespit etmek için Deeper SSD sinir ağının yukarı örnekleme ve aşağı örnekleme yöntemlerini kullanmışlardır. Deeper SSD, daha derin evrimsel katmanlar ekleyerek ve giriş görüntüsünün boyutunu büyütürken küçük nesne algılama hassasiyetini artırmaktadır. Ayrıca, eşzamanlı olarak yukarı örnekleme ve aşağı örnekleme işlemlerini kullanarak, sınıflandırma performansını ve nesne konumunun hassasiyetini arttırmışlardır. Deneyler, Deeper SSD'nin küçük drone'lar ve diğer küçük nesnelerin tespitinde etkili olduğunu göstermektedir. Ayrıca, önerilen yöntemlerin diğer nesne algılama sinir ağlarına da uyarlanabileceği ve genişletilebileceği belirtilmektedir.

Cheng vd. (2023) çalışmalarında İHA tespiti için YOLOv4 tabanlı Fast-YOLOv4 modelini geliştirmişlerdir. MobileNetV3 ile de hızlandırma parametreleri kullanmışlardır. Geliştirmiş oldukları sistemi Jetson Nano kartı ile entegre ederek deneme işlemleri gerçekleştirmişlerdir. Deneysel sonuçlar Fast-YOLOv4'ün mAP (ortalama hassasiyet) ve FPS (saniye başına kare sayısı) değerlerinin iş istasyonunda sırasıyla %90,62 ve 54 f/s'ye ulaştığı görülmektedir. NVIDIA Jetson Nano platformunda, Fast-YOLOv4'ün FPS'si normal YOLOv4'ün 2,5 katı çıkmıştır. Bu gelişmiş model performansı, gerçek zamanlı algılama gereksinimlerini karşılamakta ve bu nedenle teorik öneme ve uygulama değerine sahip olduğu ortaya çıkmıştır.

AlKhonaini vd. (2024) çalışmalarında, hiyerarşik bir pekiştirmeli öğrenme tekniği kullanarak RF (Radyo Frekansı) sinyallerine dayalı olarak izinsiz giren İHA'ları tespit etmek ve tanımlamak için yeni bir yaklaşım önermişlerdir. Genel doğruluğu artırmak için entropi düzenleme terimine sahip Reinforce algoritmasını kullanarak bir İHA'yı hiyerarşik olarak birden fazla, farklı şekilde eğitim işlemi gerçekleştirmişlerdir. Araştırma, izinsiz giren İHA'ları tespit etmek için RF sinyallerinden çıkarılan özelliklerin kullanılmasına odaklanmakta ve bu da daha az keşfedilmiş bir İHA tespit yaklaşımını araştırarak takviyeli öğrenme alanına katkıda bulunmaktadır. Değerlendirme sonucunda elde edilen bulgular, önerilen yaklaşımın RF tabanlı doğru tespit ve tanımlama konusunda bulgular %99,7 oranında tespit doğruluğu olduğunu göstermektedir.

Dadboud vd. (2021) çalışmalarında drone ve kuş tespiti ve sınıflandırma sorununun üstesinden gelmek için YOLOV5'i temel alan bir görsel drone dedektörü kullanmışlardır. Küçük nesnelerin ve karmaşık arka planların sayısını artırmak için havadan havaya çekilmiş açık kaynak bir veri seti kullanmışlardır. İki aşamalı dedektörlerin doğruluğunu test etmek ve geliştirmek için hem Faster R-CNN (Konvolüsyonel Sinir Ağı) hem de FPN (Özellik Piramidi Ağı) yöntemlerini içeren bir model eğitmişlerdir. Çalışmalarında YOLOV5 daha hızlı R-CNN+FPN modelini de geride bırakarak daha iyi sonuç vermiştir.

Delleji vd. (2020) yapmış oldukları çalışmalarında, dronelerin ciddi tehlikelere yol açabilmesi nedeniyle belirli güvenlik endişelerine neden olduğunu belirtmişlerdir. Hassas konumları ve kısıtlı alanları korumak için bu yazıda, derin öğrenme tabanlı sınıflandırma ve yerelleştirme görevlerini entegre eden bir drone tespit yöntemi önermişlerdir. Diğer modellere göre daha hızlı ve doğru tespit ettiği için YOLOv3 mimarisini tercih etmişlerdir. Hiperparametrelerde ayarlamalar yaparak YOLOv3 mimarisinde İHA eğitimleri gerçekleştirmişlerdir. Tehdit oluşturma potansiyeline sahip ve Tunus hava sahasında uçabilecek İHA'lara ait görüntülerden çok sınıflı bir veri seti oluşturmuşlardır.

Björklund (2018) çalışmasında küçük droneleri tespit etmek ve sınıflandırmak ve aynı zamanda kuşlardan droneleri ayırt etmek için Mikro-Doppler radar geliştirmiştir. Bu yöntem sayesinde, küçük dronların güvenilir bir şekilde tespit edilebileceği ve dronelerin türlerinin sınıflandırılabilceği sonuçları elde edilmiştir. Mikro İHA'ların ve kuşların radar ölçümlerini kullanmıştır. Sınıflandırma ve tespit işleminde yüksek doğruluk elde etmiştir.

Valaboju vd. (2023) gerçekleştirmiş oldukları çalışmada kötü niyetli drone faaliyetlerine karşı yardımcı olmak ve teslimat drone'ları nedeniyle gereksiz paniğe ve maddi kayıplara yol açmamak amacıyla, droneleri tespit edebilen, gerçek zamanlı bir bilgisayarlı görme sistemi önermişlerdir. Evrişimli Sinir Ağı (CNN), ve YOLOv5 mimarisi kullanmışlardır. Ordu, gözetleme ve teslimat droneleri olarak 3 ayrı kategoride tespit işlemi gerçekleştirmişlerdir.

Zhai vd. (2023) yapmış oldukları çalışmada küçük İHA hedeflerinin tespit edilmesinin

zorluğu sorununu ele almışlardır. Görüntü işleme teknikleri, kenar tespit algoritması ve İHA görüntü hedeflerini doğru bir şekilde tespit edebilen gelişmiş bir YOLOv8 algılama modeli önermişlerdir. Önermiş oldukları yöntem ile hassasiyet, duyarlılık ve ortalama hassasiyeti sırasıyla %11,9, %15,2 ve %9 oranında iyileştirmişlerdir. YOLOv8s derin öğrenme mimarisi ile kendi önermiş oldukları mimariyi karşılaştırmışlardır. Kendi önermiş oldukları mimaride 0,953 mAP değerini elde etmişlerdir.

Ye vd. (2023) çalışmalarında İHA görüntüleri için bir gerçek zamanlı nesne algılama sistemi olan RTD-Net (Gerçek Zamanlı Nesne Tespit Ağı) önermişlerdir. Önerilen yöntem, İHA'lara ait veri setinde %86,4 mAP değerinde tespit doğruluğuna ulaşmıştır. Ayrıca, NVIDIA Jetson TX2 gömülü cihazında ise %86,0 mAP algılama doğruluğuna ve 33,4 kare/sn algılama hızına ulaşmıştır. Ayrıca diğer deneysel sonuçlarda, RTD-Net'in NVIDIA GTX1080Ti üzerinde %86,4 mAP algılama doğruluğu ve 312 kare/s algılama hızı elde ettiğini göstermişlerdir.

Zhang vd. (2023) gerçekleştirdikleri çalışmada İHA'ların havadan görüntüleri üzerinde uygulanmak üzere geliştirilmiş YOLOv5 tabanlı bir nesne algılama ve tanıma yöntemi önermişlerdir. Daha belirgin kenar özelliklerine sahip olması amacı ile Gabor fonksiyonu kullanılmıştır. Daha sonra YOLOv5 mimarisine modelin daha iyi sonuçlar vermesi amacıyla koordinat mekanizması eklemişlerdir. Mimari, havadan çekilen İHA'lara ait görüntülerin tespitinde başarılı sonuçlar elde etmiştir.

Yapılan literatürdeki çalışmalara bakıldığında drone biçimindeki İHA'lara yönelik çalışmaların ağırlıklı olarak yapıldığı görülmüştür. Ancak günümüzde uçak formundaki İHA'lara patlayıcı harp başlıkları entegre edilerek aktif olarak savunma ya da saldırı amacı ile kullanılmaktadır. Bu yüzden tespit edilmesi de önem arz etmektedir. Yapılan çalışmanın YOLOv8 mimari ile entegre sistem geliştirilerek savunma sanayisinde kullanılabilecek yer istasyonları açısından öncülük edeceği düşünülmektedir.

3. MATERYAL ve METOT

Gerçekleştirilen çalışmada sivil İHA'ları tespit etmek için CNN tabanlı YOLO mimarileri kullanılmıştır. Farklı epok değerlerinde gerçekleştirilen eğitimlerde kullanılmak üzere hem sabit kanat hem de döner kanat İHA'lara ait görüntüleri içeren veri seti oluşturulmuştur. Veri setinde hem internet ortamında açık kaynak olarak sunulan İHA'lara ait görüntülere, açık kaynak veri setlerine ve tarafımızca oluşturulmuş İHA'lara ait görüntüler yer almaktadır. Eğitimler sonrasında yapılan testler ve karşılaştırmalı sonuçlar incelendiğinde YOLOv8 mimarisi ile en iyi başarı oranı elde edilmiştir.

3.1 Yapay Zeka

Yapay zeka, bilgisayar sistemlerinin insan zekasına benzer yeteneklere sahip olmasını amaçlayan bir mühendislik ve bilim dalıdır (Huang vd. 2022). Bu alan, bilgisayarların; insanlar gibi öğrenme, algılama, problem çözme, dil anlama, planlama ve karar verme gibi zekâ özelliklerini taklit etmeye çalışmaktadır. Yapay zeka, bilgisayar sistemlerine karmaşık görevleri gerçekleştirme ve öğrenme yetenekleri kazandırmayı hedeflemekte ve birçok alt dala ayrılmaktadır. Makine zekâsı (sembolik yapay zekâ), yapay sinir ağları (sibernetik yapay zekâ), doğal dil işleme, görüntü işleme, konuşma sentezi (yapay konuşma) ve konuşma anlama (konuşma analizi) bu alt dallardan bazılarıdır.

Yapay zekanın güçlü kullanıldığı alanlardan biri de nesne tespiti uygulamalarıdır (Nandhini ve Thinakaran 2023). Günlük yaşamda, özellikle karmaşık veya yoğun ortamlarda, hareketli nesnelerin veya platformlardaki nesnelerin doğru şekilde tespit edilmesi önem arz etmektedir. Tıp, askeri, endüstriyel, tarım ve mühendislik gibi alanlarda, bir ortamdaki veya görüntüdeki nesnelerin doğru şekilde tanımlanması gerekebilmektedir (Li vd. 2022, Mogili ve Deepak 2018, Özer ve Türkmen 2023). Nesne tespiti için çeşitli hızlı ve yüksek başarı oranına sahip algoritmalar geliştirilmiş olup, bu çalışmalar devam etmektedir (Ye vd. 2023).

Derin öğrenme algoritmaları, nesne tespiti uygulamalarında önemli başarılar elde etmiştir. Bu yöntem, tespit işlemlerini otomatikleştirmekte ve yüksek doğruluk oranları sağlamaktadır. Bu da derin öğrenmeye olan ilgiyi arttırmaktadır. Görsel tanıma

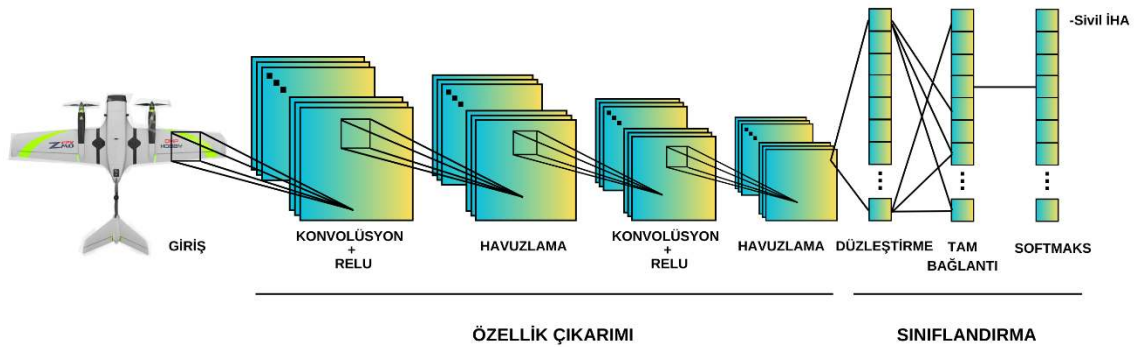
uygulamalarındaki büyük gelişmeler, derin öğrenme modellerinin insanlardan daha üstün performans sergileyebileceğini göstermiştir. Derin öğrenme, nesne tanıma ve tespiti gibi görevlerde önemli bir araç haline gelmiş ve çeşitli endüstrilerde aktif olarak kullanılmaktadır.

Derin öğrenme, genellikle çok katmanlı yapay sinir ağları kullanarak büyük veri setleri üzerinde eğitim işlemi gerçekleştirmektedir. Bu veri setleri, modelin karmaşık desenleri öğrenmesini sağlamak ve insan gözüyle zor algılanabilen özellikleri çıkarmaktadır. Derin öğrenme, görüntü veya nesne tanıma, doğal dil işleme ve oyun stratejileri gibi karmaşık görevlerde yüksek başarı elde etme potansiyeline sahiptir.

Son yıllarda, özellikle görsel görevlerde başarıyla uygulanan Evrişimsel Sinir Ağları (CNN) bu alandaki en etkili modellerden biri olmuştur. CNN'ler, konvolüsyon katmanları aracılığıyla görüntü özelliklerini çıkarma yetenekleri sayesinde nesne tanıma, yüz tanıma ve benzeri görevlerde oldukça etkin bir şekilde yer almaktadır. Derin öğrenme, görüntü işleme dışında ses tanıma, doğal dil işleme ve özerk araçlar gibi birçok farklı alanda da kullanılmaktadır. Bu teknik, karmaşık veri setlerindeki desenleri anlamak ve genelleme yeteneği ile bilgi işleme konusunda önemli ilerlemeler sağlamıştır.

3.1.1 Evrişimsel Sinir Ağı (CNN)

Evrişimsel Sinir Ağı (CNN) ifade kısaca, "Convolutional Neural Network" yani "Evrişimsel Sinir Ağı" ifadesinin kısaltmasıdır. Farklı alanda kullanılsa da genellikle görüntü işleme gibi işlemlerde aktif olarak kullanılan sinir ağı modelidir. Nesne tespiti; İHA, uçak, askeri araç, bitki, hayvan vb. gibi objeleri içeren görüntüler ve pikseller üzerindeki işlemlerde kullanılmaktadır.



Şekil 3.1 CNN mimari yapısı

Ağ, Şekil 3.1’de görüldüğü üzere konvolüsyon işlemleri ve katmanlar arası havuzlama işlemleri gibi özel mimari özellikleri ile öne çıkmaktadır. Temel amacı, karmaşık görsel desenleri öğrenip tanıma yeteneği kazanarak veri analizi ve sınıflandırma yapmaktır.

CNN yapısı genellikle birkaç temel katmandan oluşmaktadır:

1. Giriş Katmanı: CNN'nin ilk katmanı olan giriş katmanı, işlenecek olan veriyi almaktadır. Bu genellikle bir görüntü olup, piksel değerlerini içermektedir.
2. Konvolüsyon Katmanları: Konvolüsyon katmanları, belirli özellikleri vurgulamak için giriş verisine filtreler uygulamaktadır. Bu filtreler, örneğin kenarlar veya renk değişimleri gibi görsel desenlere tepki vermektedir. Her katman, öncekilerden daha karmaşık özellikleri öğrenerek ilerlemektedir.
3. Aktivasyon Katmanları: Konvolüsyon işlemlerinden sonra aktivasyon fonksiyonları uygulanmaktadır. ReLU (Düzeltilici Doğrusal Birim) yaygın olarak kullanılmaktadır. Bu adım, ağın öğrenme yeteneğini arttırmaktadır.
4. Havuzlama Katmanları: Havuzlama katmanları, konvolüsyon çıktılarını özetlemekte ve boyutlarını küçültmektedir. Maksimum havuzlama veya ortalama havuzlama gibi yöntemler kullanılarak önemli özellikler korunmaktadır.
5. Tam Bağlantılı Katmanlar: Tam bağlantılı katmanlar, önceki katmanlardan gelen özellikleri kullanarak sınıflandırma yapmaktadır. Bu katmanlar, nihai sınıflandırma kararlarını almaktadır.
6. Çıkış Katmanı: CNN'nin son katmanı olan çıkış katmanı, sınıflandırma sonuçlarını üretmektedir. Softmaks fonksiyonu genellikle çoklu sınıf problemleri için kullanılmaktadır.

Bu katmanlar arasındaki etkileşim, özellikle konvolüsyon ve havuzlama katmanları sayesinde, ağın hiyerarşik özellikleri öğrenmesine imkan tanımaktadır. CNN yapısı, özellikle görüntü tanıma ve sınıflandırma görevlerinde oldukça etkili olmaktadır.

3.1.1.1 Giriş Katmanı

CNN'nin giriş katmanı, ağa veri girişi yapılacak olan ilk katmandır. Bu katman, genellikle bir görüntüden oluşan giriş verisini alarak işleme başlamaktadır. Görüntü, piksel değerlerinden oluşan bir matristir. Her bir piksel, giriş verisinin bir özelliğini temsil etmektedir.

Örneğin, renkli görüntüde her piksel üç renk kanalına (kırmızı, yeşil, mavi) ait değerleri içermektedir. Gri tonlu bir görüntüde ise her piksel sadece tek bir renk değeri içermektedir. Bu piksel değerleri, CNN'nin işleyeceği temel veriyi oluşturmaktadır.

Giriş katmanı, bu piksel değerlerini alıp işlemeye başlamadan önce genellikle normalizasyon veya ölçeklendirme gibi ön işlemler uygulanmaktadır. Bu işlem de, ağı daha etkili ve istikrarlı bir şekilde öğrenmesine yardımcı olabilmektedir.

Giriş katmanı, daha sonra konvolüsyon katmanlarına geçişi sağlamaktadır. Bu katmanlar, görüntü üzerinde özellikleri vurgulamak ve hiyerarşik öğrenme gerçekleştirmek amacıyla konvolüsyon (filtre uygulama) işlemlerini gerçekleştirmektedir. Bu sayede ağ, giriş verisinden karmaşık özellikleri öğrenmeye başlamaktadır.

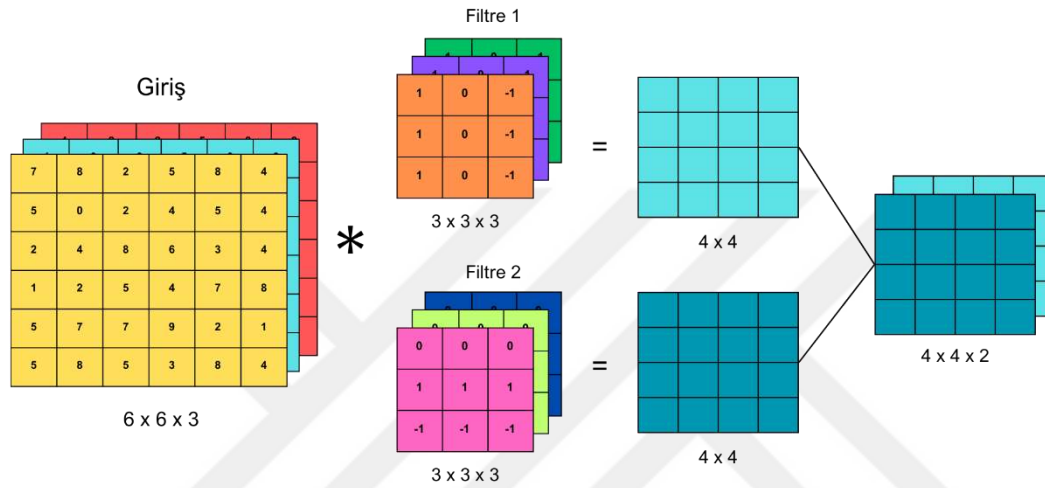
3.1.1.2 Konvolüsyon Katmanı

Konvolüsyon Katmanları, CNN yapısının temel bileşenlerinden biridir ve özellikle görüntü işleme görevlerinde yaygın olarak kullanılmaktadır. Bu katmanlar, giriş verisindeki özellikleri belirlemek ve bu özellikleri vurgulamak amacıyla konvolüsyon işlemlerini gerçekleştirmektedir. Konvolüsyon katmanlarının temel özelliklerinden biri, filtre (kernel) adı verilen küçük matrislerin kullanılmasıdır. Bu filtreler, görüntü üzerindeki belirli özellikleri, örneğin kenarlar, köşeler veya renk değişimlerini tespit etmek amacıyla işlev görmektedir. Filtreler, görüntü üzerinde belirli bir bölgeyi tarayarak bu bölgenin özelliklerini vurgulamaktadır.

Konvolüsyon işlemi sırasında, Şekil 3.2'deki gibi filtre giriş verisi üzerinde kaydırılarak uygulanmakta ve bu işlem, filtre ile giriş verisi arasındaki çarpımları alıp bu çarpımları toplamayı içermektedir. Bu sayede, özellik haritası adı verilen yeni bir matris elde edilmektedir. Ayrıca, striding (kayma) olarak adlandırılan bir parametre, filtrenin giriş verisi üzerinde ne kadar mesafe kaydırılacağını belirlemektedir. Striding, konvolüsyon işlemi sırasında adım sayısını ve dolayısıyla işlem yoğunluğunu etkileyen önemli bir unsurdur. Padding (kenar dolgu) ise, giriş verisinin kenarlarında eksiklikleri önlemek amacıyla eklenen yapay piksellerdir. Bu yöntem, filtrelerin kenar piksellerini de işleyerek giriş verisi üzerinde hareket etmelerini sağlamaktadır.

Konvolüsyon işleminin ardından, her elemana bir aktivasyon fonksiyonu uygulanmaktadır; bu fonksiyon genellikle ReLU (Rectified Linear Unit) olarak

seçilmektedir. ReLU fonksiyonu, negatif değerleri sıfıra eşitleyerek pozitif değerleri olduğu gibi bırakmakta ve bu şekilde işlem yapmaktadır. Bu süreçler, konvolüsyon katmanlarının temel yapı taşlarını oluşturmaktadır. Birden çok konvolüsyon katmanının bir araya gelmesi, ağın giderek daha karmaşık ve soyut özellikleri öğrenmesine olanak tanımakta ve bu katmanlar, özellikle görüntü tanıma ve nesne tespiti gibi görevlerde etkili sonuçlar elde etmek için kullanılmaktadır.



Şekil 3.2 Konvolüsyon katmanı yapısı

3.1.1.3 Aktivasyon Katmanları

CNN içinde yer alan aktivasyon katmanları, genellikle konvolüsyon işlemleri sonrasında elde edilen çıktı üzerinde uygulanan lineer olmayan dönüşümleri sağlamaktadır. Bu katmanlar, ağın öğrenme yeteneğini ve karmaşıklığını artırarak daha geniş bir özellik kümesini temsil etmesine yardımcı olmaktadır.

En sık kullanılan aktivasyon fonksiyonu ReLU'dur. ReLU, negatif değerleri sıfıra eşitleyerek ve pozitif değerleri değiştirmeden bırakarak lineer olmayan bir dönüşüm uygulamaktadır. ReLU'nun matematiksel formülü ise aşağıdaki denklem (3.1) de görüldüğü gibidir:

$$f(x) = \max(0, x) \quad (3.1)$$

Bu, negatif deęerleri sıfıra eřitledięi ve pozitif deęerleri deęiřtirmeden bıraktıęı anlamına gelmektedir. Aktivasyon katmanları, sadece ReLU ile sınırlı deęildir. Dięer yaygın olarak kullanılan aktivasyon fonksiyonları arasında řunlar yer almaktadır:

Sigmoid Fonksiyonu ise denklem (3.2) de g r lmektedir.

$$f(x) = 1 / (1 + \exp(-x)) \quad (3.2)$$

Bu fonksiyon,  ıkıřı [0, 1] aralıęına sıkıřtırarak genellikle ikili sınıflandırma problemlerinde kullanılmaktadır.

Denklemleri 3.3 de verilen bir dięer aktivasyon fonksiyonu ise tanh'dir.

$$f(x) = (\exp(x) - \exp(-x)) / (\exp(x) + \exp(-x)) \quad (3.3)$$

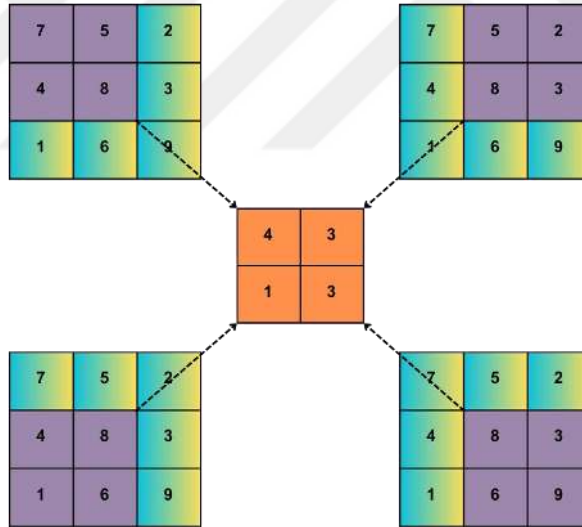
Denklemlerde de g r ld ę  gibi Sigmoid fonksiyonuna benzer ancak  ıkıřı [-1, 1] aralıęında deęer almaktadır. Softmaks fonksiyonu ise genellikle  ıkıř katmanında kullanılmaktadır. Sınıflandırma problemlerinde  oklu sınıflar arasında olasılıkları temsil ederek kullanım saęlamaktadır. Aktivasyon katmanları, aęın her bir  zellik haritasının daha karmařık ve genelleme yetenekleriyle dolu hale gelmesine yardımcı olmaktadır. Bu, aęın daha y ksek d zeyde soyut  zellikleri  ęrenmesine ve genelleme yeteneęini artırmasına olanak tanımaktadır.

3.1.1.4 Havuzlama Katmanları

CNN i indeki havuzlama katmanları,  zellik haritasındaki boyutu k   ltmek ve  nemli  zellikleri vurgulamak amacıyla kullanılan katmanlardır. Havuzlama katmanları, aęın  ęrenme s recini hızlandırmak, hesaplama maliyetini azaltmak ve ařırı  ęrenmeyi engellemek gibi avantajlar saęlamaktadır. Havuzlama iřlemi genellikle herhangi bir b lgedeki maksimum deęeri (maksimum havuzlama) veya b lgedeki deęerlerin ortalamasını (ortalama havuzlama) olarak ger ekleřtirilmektedir. Maksimum havuzlama daha yaygın olarak tercih edilmektedir.   nk  belirli  zelliklerin vurgulanmasında daha etkilidir.

Havuzlama katmanları, sinir aęlarında  zellik haritalarının boyutunu k   lterek hesaplama maliyetini azaltan  nemli bir bileřen olarak iřlev g rmektedir. 2x2 boyutunda bir havuzlama iřlemi uygulandıęında,  zellik haritasının her 2x2 b lgesinden tek bir

değer alınmakta ve bu sayede boyut yarıya indirilmektedir. Havuzlama işlemi, çeşitli türlerde gerçekleştirilebilmektedir; bunlar arasında maksimum havuzlama ve ortalama havuzlama yer almaktadır. Maksimum havuzlama, bölgedeki en büyük değeri seçerek belirli özelliklerin vurgulanmasına katkıda bulunmakta ve bu nedenle genellikle daha sık tercih edilmektedir. Ortalama havuzlama ise bölgedeki değerlerin ortalamasını almaktadır. Ayrıca, havuzlama işlemi sırasında kayma (stride) adı verilen bir parametre kullanılarak işlem yapılacak alan belirli aralıklarla kaydırılmaktadır. Yüksek kayma, boyutu daha fazla küçültse de bazı önemli bilgilerin kaybedilmesine neden olabilmektedir. Havuzlama katmanları, genellikle konvolüsyon katmanlarından sonra gelmekte ve özellik haritasındaki önemli özellikleri koruyarak boyutu küçültmektedir. Bu sayede ağ, daha küçük ve anlamlı özelliklere odaklanırken hesaplama maliyetini azaltmakta ve genel verimliliği artırmaktadır. Havuzlama katmanının çalışma yapısı ise Şekil 3.3’de yer almaktadır.



Şekil 3.3 Havuzlama katmanı yapısı

3.1.1.5 Tam Bağlantılı Katmanlar

Tam Bağlantılı Katmanlar (Fully Connected Layers), CNN yapısının bir parçası olup, genellikle ağın sınıflandırma veya regresyon gibi nihai çıktılarını üretmekle sorumludur. Bu katmanlar, önceki konvolüsyon ve havuzlama katmanlarından gelen özellikleri kullanarak genel bir bağlam oluşturmaktadır. Tam bağlantılı katmanların temel

özelliklerinden biri düzleştirme (flatten) işlemidir. Önceki katmanlardan elde edilen özellik haritaları genellikle 2D matrisler biçiminde olup, tam bağlantılı katmanlarla uyumlu olabilmeleri için bu matrisler düzleştirilmekte, yani tek boyutlu vektörlere dönüştürülmektedir. Düzleştirilmiş bu özellik vektörleri, tam bağlantılı katmanlarda her bir nörona bağlanmaktadır. Bu nöronlar, ağırlıklı girişleri almakta, bir aktivasyon fonksiyonu uygulamakta ve ardından çıkışı üretmektedir. Tam bağlantılı katmanlar, genellikle sınıflandırma veya regresyon problemlerine yönelik sonuçlar üretmektedir; örneğin, softmax aktivasyon fonksiyonu kullanılarak çoklu sınıflandırma problemleri çözülebilmektedir. Bu katmanlar, önceki katmanlardan gelen özellikleri bir araya getirerek daha yüksek düzeyde soyut özelliklerin öğrenilmesine olanak tanımaktadır. Ancak, tam bağlantılı katmanlar genellikle ağırlık parametre sayısını artırarak hesaplama maliyetini yükseltmektedir. Bu sebeple, genellikle konvolüsyon ve havuzlama katmanlarından sonra gelmektedirler.

3.1.1.6 Çıkış Katmanı

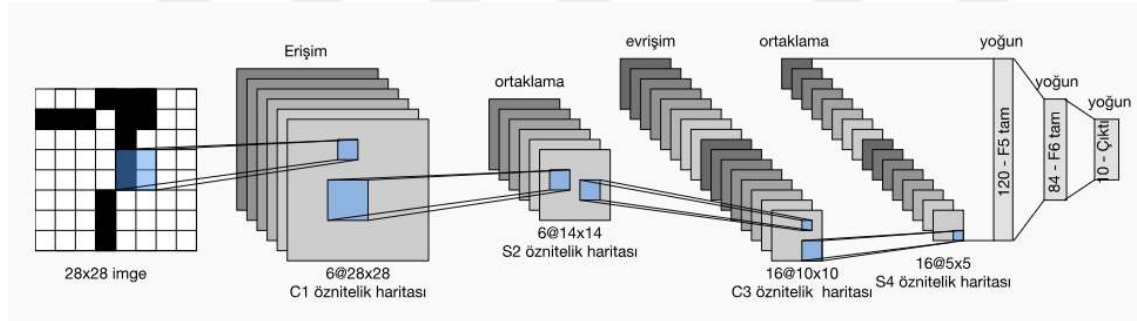
CNN'nin çıkış katmanı, ağırlık nihai çıktıları üreten ve genellikle belirli bir görevle ilgili sonuçları sağlayan katman olarak işlev görmektedir. Çıkış katmanının yapısı, çözülmek istenen probleme göre değişiklik göstermektedir. Çıkış katmanının temel özelliklerinden biri, kullanılan aktivasyon fonksiyonudur. Bu fonksiyon, problemin türüne bağlı olarak farklılık göstermektedir. Örneğin, ikili sınıflandırma problemlerinde sigmoid fonksiyonu yaygın olarak tercih edilmekte olup, bu fonksiyon çıkışı $[0, 1]$ Değerlerinden birini alarak temsil ettiği sınıfı göstermektedir. Çoklu sınıflandırma problemlerinde ise softmax fonksiyonu kullanılmakta ve bu fonksiyon, çıkış vektöründeki değerleri normalize ederek toplam olasılığın 1 olmasını sağlamaktadır.

Çıkış katmanında kullanılan bir diğer önemli bileşen kayıp fonksiyonudur. Kayıp fonksiyonu, ağırlık ürettiği çıkışlar ile gerçek etiketler arasındaki farkı ölçmektedir. İkili sınıflandırma için "binary crossentropy" kayıp fonksiyonu yaygın olarak kullanılmakta, çoklu sınıflandırma problemlerinde ise "categorical crossentropy" tercih edilmektedir. Çıkış katmanının nasıl yorumlanacağı ise çözülen probleme bağlı olarak farklılık göstermektedir. Örneğin, ikili sınıflandırma problemlerinde çıkış katmanı genellikle bir sayıdan oluşmakta ve bu sayı belirli bir eşik değeri üzerinde olup olmadığına göre karar verilmektedir. Çoklu sınıflandırma problemlerinde ise her bir sınıfa ait olasılıkları içeren

bir vektör elde edilmektedir. CNN, eğer bir regresyon problemini çözmek amacıyla kullanılıyorsa, çıkış katmanı genellikle doğrudan tahmin edilen değeri içermektedir. CNN'nin çıkış katmanı, ağıın öğrendiği özellikleri ve bilgileri, çözölen görevin gereksinimlerine uygun bir biçimde ifade etmektedir. Bu katman, aynı zamanda ağıın eğitim süreci boyunca optimize edildiğı ve performansının değeriendirildiğı yerdir.

3.1.2 CNN Mimari Türleri

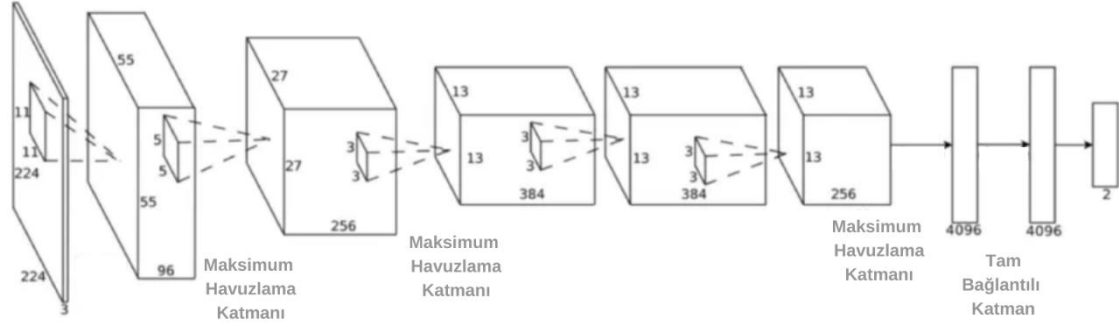
LeNet-5, 1998 yılında Yann LeCun ve meslektaşları tarafından geliştirilen erken dönem CNN mimarilerinden biridir. Özellikle el yazısı rakam tanıma amacıyla tasarlanan LeNet-5, MNIST (Karışık Ulusal Standartlar ve Teknoloji Enstitüsü) veri seti üzerinde büyük başarı elde etmiştir. Yapısal olarak, Şekil 3.4'deki gibi bir dizi konvolüsyon (convolutional) ve alt örnekleme (subsampling/pooling) katmanlarından oluşmaktadır. Tipik olarak, LeNet-5 iki konvolüsyon katmanı, iki alt örnekleme katmanı ve iki tam bağlantılı katmandan meydana gelmektedir. Basitliğı ve etkinliğı, LeNet-5'in modern CNN mimarilerinin temel taşlarından biri olarak kabul edilmesini sağlamıştır.



Şekil 3.4 LeNet-5 mimarisi (İnt. Kyn. 2)

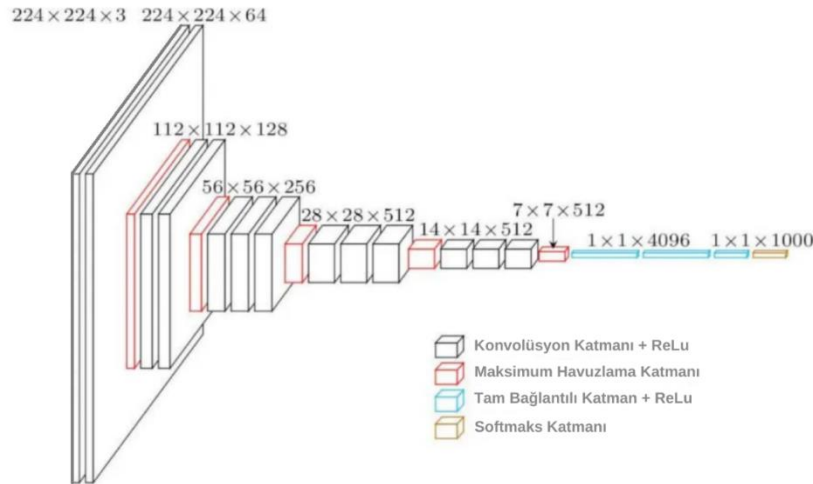
AlexNet, 2012 yılında Alex Krizhevsky, Ilya Sutskever ve Geoffrey Hinton tarafından geliştirilen görüntü sınıflandırma alanında devrim niteliğinde bir ilerleme kaydetmiştir. Özellikle ImageNet yarışmasında elde ettiği büyük başarı ile tanınan AlexNet, Şekil 3.5'te de göröldüğü üzere beş konvolüsyonel katman ve üç tam bağlantılı katmandan oluşan derin ve geniş bir mimariye sahiptir. ReLU aktivasyon fonksiyonu ve dropout tekniğini kullanarak, eğitim sürecini hızlandırmış ve aşırı öğrenmeyi (overfitting) azaltmıştır. Bu mimari, GPU (Grafik İşlem Birimi) kullanımı ile büyük veri kümeleri

üzerinde derin ağların eğitime öncülük ederek alanında çığır açmıştır.



Şekil 3.5 AlexNet mimari yapısı

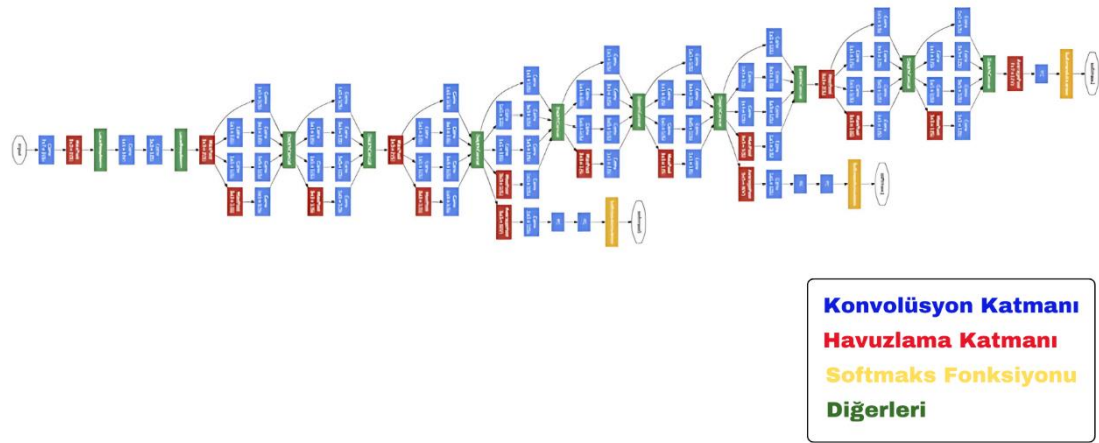
VGGNet, 2014 yılında Oxford Üniversitesi Visual Geometry Group (VGG) tarafından geliştirilen sabit boyutlu filtrelerin derinlemesine kullanımı ile tanınmaktadır. VGGNet, Şekil 3.3.6’da yer aldığı gibi 16 (VGG16) veya 19 (VGG19) katmandan oluşmakta olup, her bir katmanda 3x3 boyutlu küçük filtreler kullanarak ağı derinliğini artırmaktadır. Bu yaklaşım, daha ince ve daha derin ağların oluşturulmasını sağlayarak modelin daha karmaşık özellikleri öğrenmesine olanak tanımaktadır. VGGNet’in basitliği ve yüksek doğruluğu, onu görüntü sınıflandırma görevlerinde yaygın olarak tercih edilen bir mimari haline getirmiştir.



Şekil 3.6 VGGNet mimari yapısı

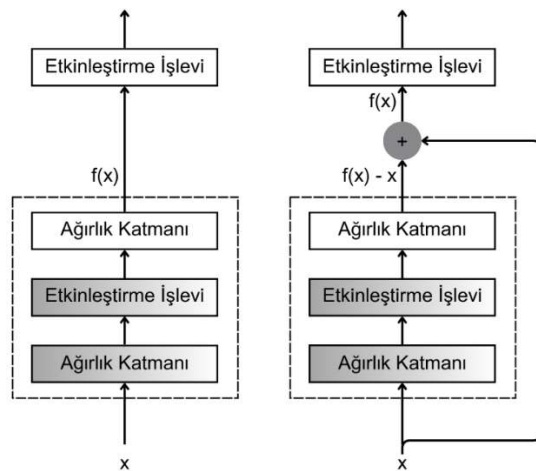
GoogLeNet, Google tarafından 2014 yılında geliştirilmiştir. Bu modüller, farklı boyutlardaki filtreleri bir araya getirerek aynı katmanda çeşitli ölçeklerde bilgi çıkarımı

sağlamaktadır. GoogLeNet, derin bir yapı sunarken parametre sayısını azaltarak verimliliği artırmaktadır. Şekil 3.7’de yer alan ve 22 katmandan oluşan bu mimari, özellikle ImageNet yarışmasındaki başarısıyla dikkat çekmiştir. GoogLeNet’in yenilikçi Inception modülleri, derin öğrenme modellerinde esneklik ve verimlilik sağlamaktadır.



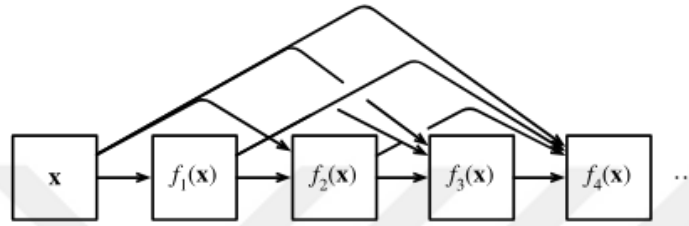
Şekil 3.7 GoogLeNet mimari yapısı

ResNet (Kalıntı Ağ), 2015 yılında Microsoft Research tarafından geliştirilmiştir. ResNet mimarisi, her katmanda giriş bilgilerini bypass ederek öğrenmeyi hızlandırmakta ve gradyan kaybolma problemini azaltmaktadır. ResNet, 50, 101 ve 152 katmanlı versiyonları ile tanınmakta ve derinliği artırarak model performansını iyileştirmektedir. Bu mimari, derin ağların performansını önemli ölçüde artırmış ve daha karmaşık modellerin geliştirilmesine olanak sağlamıştır. Şekil 3.8’de bu mimarinin yapısı yer almaktadır.



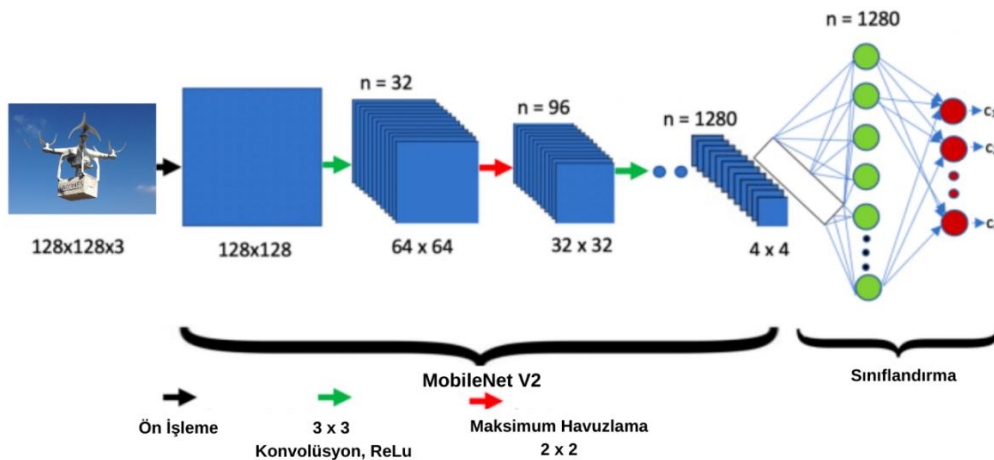
Şekil 3.8 Resnet mimari yapısı

DenseNet, 2016 yılında Gao Huang ve ekibi tarafından geliştirilen "dense blokları" kullanarak Şekil 3.9'daki gibi her katmanı önceki tüm katmanlara bağlamaktadır. Bu yapı, daha verimli özellik kullanımını ve gradyan yayılımını desteklemekte, aynı zamanda daha az parametre ile daha iyi performans elde etmektedir. DenseNet, özellikle görüntü sınıflandırma görevlerinde etkinliği kanıtlanmış bir mimaridir. Bu mimari, özelliklerin yeniden kullanımını artırarak modelin genel verimliliğini ve performansını optimize etmektedir.



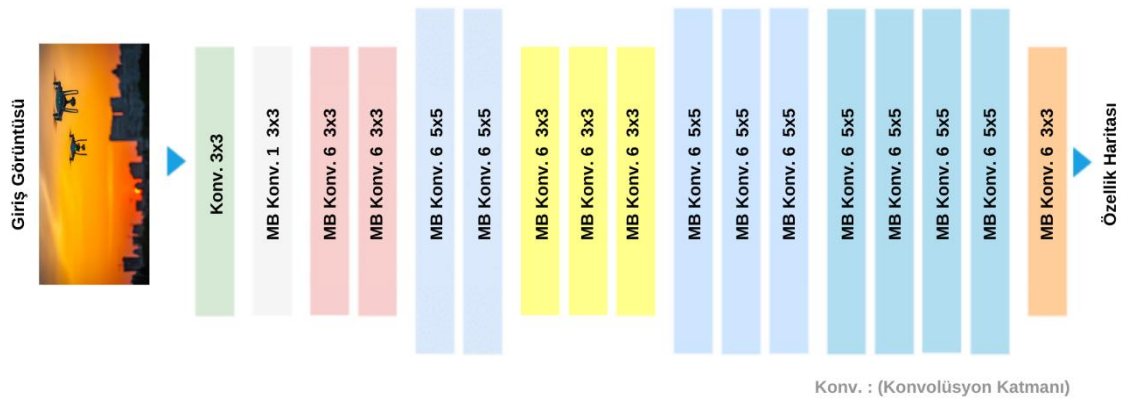
Şekil 3.9 DenseNet yoğun katman yapıları (İnt. Kyn. 2)

MobileNet, Google tarafından 2017 yılında geliştirilen mobil ve gömülü cihazlar için optimize edilmiş bir CNN mimarisidir. MobileNet, derinlik ayırık konvolüsyonlar (depthwise separable convolutions) kullanarak model boyutunu ve hesaplama gereksinimlerini önemli ölçüde azaltmaktadır. Bu özellik, düşük güç tüketimi ve hızlı işlem kapasitesi gerektiren cihazlarda yüksek performans sağlamaktadır. MobileNet, verimlilik ve taşınabilirlik açısından ön plana çıkmakta ve çeşitli mobil uygulamalarda yaygın olarak kullanılmaktadır. Şekil 3.10'da MobileNet mimarisinin katman yapısı görülmektedir.



Şekil 3.10 Mobilenet mimari yapısı

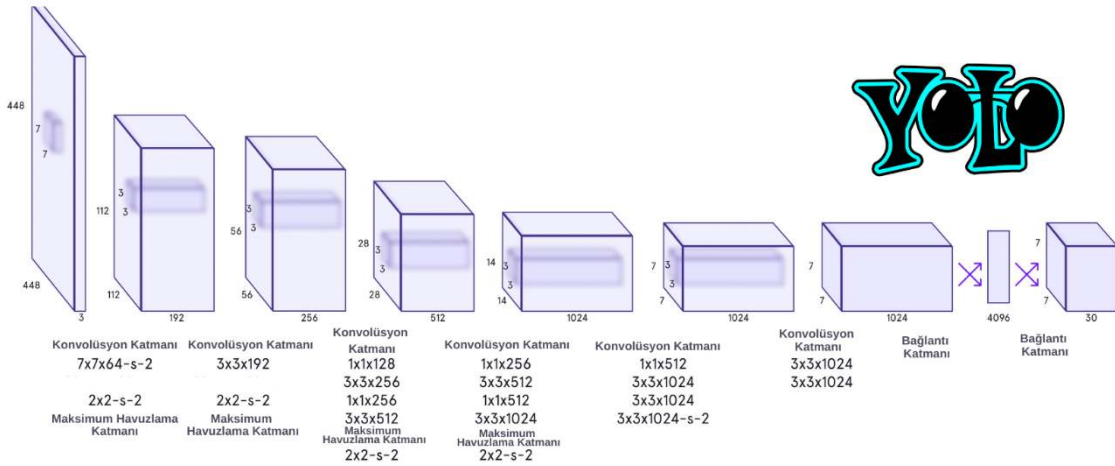
EfficientNet, Google tarafından 2019 yılında geliştirilerek model ölçekleme yöntemlerini kullanarak verimliliği artırmayı hedeflemektedir. Şekil 3.11’de mimarinin genel katman yapısı yer almaktadır. EfficientNet, genişlik, derinlik ve çözünürlük ölçeklemeleri ile farklı versiyonlar sunmaktadır. Bu ölçekleme yöntemleri, modelin daha dengeli ve optimize edilmiş bir şekilde büyümesini sağlamaktadır. EfficientNet, daha az parametre ile daha yüksek performans elde ederek, doğruluk ve verimlilik açısından önemli bir ilerleme kaydetmiştir.



Şekil 3.11 EfficientNet mimari yapısı

YOLO, nesne tespiti ve sınıflandırma işlemlerini tek bir ağda birleştirerek gerçek zamanlı nesne tespiti için kullanılan bir derin öğrenme mimarisidir (İnt. Kyn. 3) . YOLO, bir görüntüyü $S \times S$ boyutlarında ızgaralara (grid) ayırmaktadır. Her bir ızgarayı, bir nesnenin sınırlayıcı kutusunu (bounding box) ve bu nesnenin sınıfını tahmin etmek için bir sinir ağı aracılığıyla işlemektedir. Bu işlem, tek seferde gerçekleştirilmektedir. Bu sayede YOLO, diğer nesne tespiti algoritmalarına kıyasla daha hızlı sonuçlar üretmektedir. YOLO mimarisi, Darknet adı verilen bir CNN modelini temel almaktadır. Bu model, 53 katmandan oluşmakta ve ImageNet veri kümesi üzerinde önceden eğitilmiştir. YOLO, nesne tespiti için bu modele 53 ek katman ekleyerek toplamda 106 katmanlı bir ağ oluşturmaktadır. YOLO hız ve doğruluk açısından diğer mimarilere kıyasla öne çıkmaktadır. Saniyede 45 kareye kadar görüntü işleyebilirken aynı zamanda yüksek doğruluk oranları sağlamaktadır. Kullanım alanları arasında görüntü işleme, robotik görüş, artırılmış gerçeklik vb. alanlar bulunmaktadır. YOLO, fotoğraf ve videolardaki nesneleri tespit etmek, robotların çevrelerini algılamasına yardımcı olmak ve artırılmış

gerçeklik uygulamalarında sanal nesneleri gerçek dünyaya entegre etmek için kullanılabilir. Araştırmacılar tarafından sürekli olarak geliştirilen YOLO'nun yeni versiyonları, küçük nesneleri tespit etme ve genelleme yeteneğinde daha iyi performans göstermektedir. Gelecekte, YOLO'nun nesne tespiti ve sınıflandırma alanında en yaygın kullanılan algoritmalarından biri olması beklenmektedir.

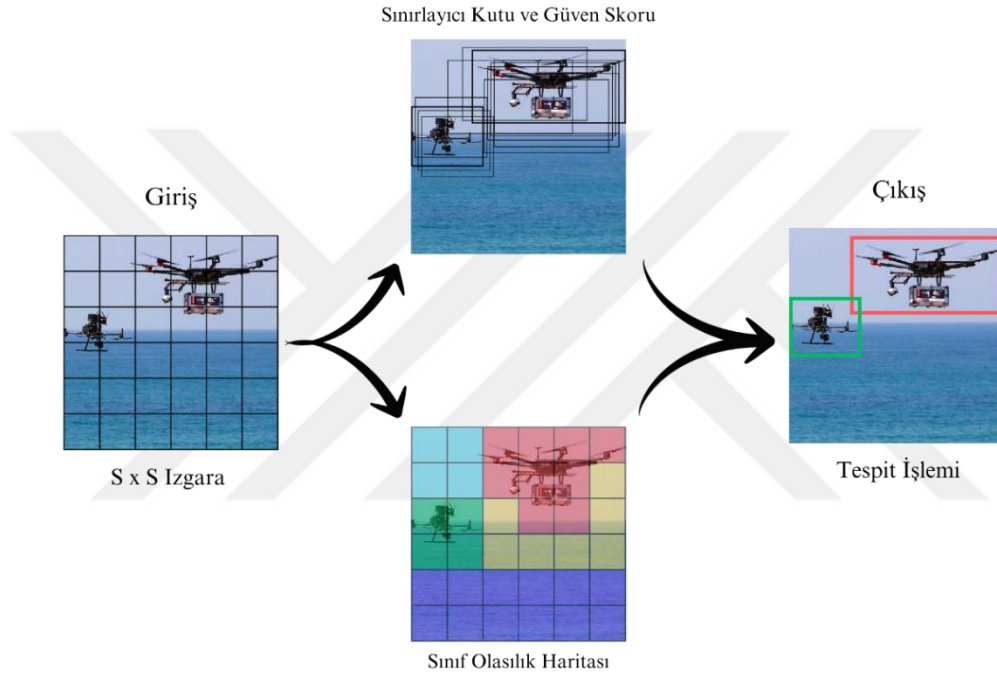


Şekil 3.12 YOLO mimarisi yapısı

YOLO mimarisinin katman yapısı ise Şekil 3.12’de de yer aldığı gibi şu şekildedir: İlk aşamada, YOLO'nun giriş olarak aldığı görüntü belirli bir boyutta (genellikle 416x416 veya 608x608 piksel) yeniden boyutlandırılır. Bu, hesaplamaların ve tahminlerin standart bir boyutta gerçekleştirilmesini sağlamaktadır. Ağ yapısı, CNN tabanlıdır ve genellikle bir dizi konvolüsyonel (convolutional) ve havuzlama (pooling) katmanından oluşmaktadır. Konvolüsyonel katmanlar, görüntüden özellikleri çıkararak ve filtreler kullanarak kenarları, dokuları ve daha yüksek seviyeli özellikleri algılamaktadır. Havuzlama katmanları, özellik haritalarının boyutunu azaltarak hesaplamaları hızlandırmakta ve yerel özelliklerini korumaktadır. Tam bağlantılı katmanlar ise genellikle ağın sonlarında bulunarak sınıflandırma işlemlerini gerçekleştirmektedir.

Giriş aşamasında, Şekil 3.13’de yer aldığı şekilde YOLO görüntüyü SxS boyutlarında bir ızgaraya bölmektedir. Her bir ızgara hücresi, belirli sayıda öngörude bulunmaktadır. Bu öngörüler sınırlayıcı kutu koordinatlarını (nesnenin konumunu ve boyutunu belirten x, y, genişlik ve yükseklik değerleri) ve sınıf olasılıklarını (belirli bir hücrede hangi nesnenin bulunduğunu gösteren olasılıklar) içermektedir. Her hücrede, nesne olup olmadığını

belirten bir "F1-Skor yani confidence skor (güven puanı)" hesaplanmaktadır. Bu puan, bounding box'ın doğruluğunu ve nesnenin o hücrede bulunma olasılığını ifade etmektedir. Aynı zamanda kayıp fonksiyonu ise, YOLO'nun ağı eğitirken; bounding box kayıpları (tahmin edilen ve gerçek bounding box koordinatları arasındaki fark), confidence kayıpları (tahmin edilen confidence skor ile gerçek değer arasındaki fark) ve sınıflandırma kayıpları (tahmin edilen sınıf olasılıkları ile gerçek sınıf etiketleri arasındaki fark) minimize etmeye çalıştığı bileşenleri içermektedir.



Şekil 3.13 YOLO mimarisi ile tespit adımları

Tahmin ve Non-Maximum Suppression (NMS) aşamasında, YOLO her bir hücre için birden fazla bounding box ve sınıf etiketi öngörmektedir. Ancak, bazı bounding box'lar aynı nesneyi temsil edebilmektedir. Bu durumlarda, NMS algoritması kullanılarak yalnızca en yüksek confidence skor'a sahip bounding box'lar seçilerek diğerleri elenmektedir.

YOLO mimarisi sürekli olarak güncellemekte ve yeni versiyonları çıkmaktadır. Bu versiyonlarda farklı katman yapıları ve farklı parametreler yer almaktadır. Aşağıdaki Çizelge 3.1'de bu değişiklikler görülmektedir.

Çizelge 3.1 YOLO derin öğrenme mimarilerinin versiyon farklılıkları

	YOLOv5	YOLOv7	YOLOv8	YOLOv10
Çıkış Tarihi	2020	2022	2023	2024
Geliştirici	Ultralytics	WongKinYiu	Ultralytics	Ultralytics
Model Boyutları	Küçük, Orta, Büyük, X-Large	Küçük, Orta, Büyük, X-Large	Nano, Küçük, Orta, Büyük	Küçük, Orta, Büyük, X-Large
Parametre Sayısı	~7M - ~76M	~7M - ~100M	~3M - ~87M	~3M - ~90M
Katman Sayısı	225-365	244-490	195-390	205-400
Ana Yapı	Kısmi Çapraz Aşama	Geniştirilmiş Verimli Katman Toplama Ağı	C3 Dönüştürücü	C3-L Dönüştürücü
Doğruluk (mAP)	İyi	Çok İyi	Yüksek doğruluk	Yüksek doğruluk
Ağ Çıkışı	Tek çıkış	Çoklu çıkış	Optimize edilmiş çıkış	Optimize edilmiş çoklu çıkış

3.2 SAHI (Dilimleme Destekli Hiper İnfers) Algoritması

SAHI (Sliding Window and Hierarchical Inference) algoritması, büyük ve yüksek çözünürlüklü görüntülerde nesne tespiti gerçekleştirmek için geliştirilmiş bir yöntemdir (Akyon vd. 2021). Bu algoritma, görüntüleri daha küçük ve yönetilebilir pencerelere bölmekte ve her bir pencerede nesne tespiti uygulamaktadır. Kaydırma penceresi yaklaşımı ile, her bir küçük bölümde ayrıntılı tespitler yaparak, tüm görüntüdeki nesnelerin konumlarını ve etiketlerini daha doğru bir şekilde belirlemektedir. Hiyerarşik çıkarım sürecinde ise, küçük pencerelerde elde edilen tespitleri bir araya getirerek ve genel bir değerlendirme yaparak büyük ölçekli görüntülerde daha kapsamlı ve hassas sonuçlar sağlamaktadır. SAHI algoritması, bu yöntemleri kullanarak yüksek çözünürlüklü görüntülerdeki hesaplama yükünü azaltmakta ve detayları daha iyi yakalamaktadır. Ayrıca, esnekliği sayesinde farklı çözünürlüklerdeki ve boyutlardaki görüntülerle etkili bir şekilde çalışabilmektedir. Bu özellikleri, özellikle tıbbi görüntüleme ve uydu görüntüleri gibi karmaşık veri setlerinde SAHI'yi etkili bir nesne tespiti aracı olarak kullanma imkânı sağlamaktadır. Dilimlenmiş çıkarım yöntemi ile çalışan SAHI algoritması, büyük veya yüksek çözünürlüklü bir görüntünün daha küçük parçalara (dilimlere) bölünmesini, bu dilimler üzerinde nesne tespiti yapılmasını ve

ardından orijinal görüntü üzerindeki nesne konumlarının yeniden yapılandırılmasını ifade eden bir tekniktir. Bu yaklaşım, özellikle hesaplama kaynaklarının sınırlı olduğu durumlarda veya aşırı yüksek çözünürlüklü görüntülerle çalışırken bellek sorunlarını önlemekte büyük önem taşımaktadır. Dilimlenmiş çıkarımın başlıca avantajları arasında azaltılmış hesaplama yükü, korunmuş algılama kalitesi ve geliştirilmiş ölçeklenebilirlik bulunmaktadır. Resim 3.1’de A ve B görsellerinde görüldüğü gibi daha küçük dilimlerin işlenmesi, daha hızlı işleme ve daha az bellek tüketimi gerektirerek düşük kaliteli donanımlarda daha verimli çalışmayı sağlamaktadır. Her dilim bağımsız olarak ele alındığı için, dilimlerin yeterince büyük olması durumunda nesne algılama kalitesinde herhangi bir azalma meydana gelmemektedir. Uzak mesafede yer alacak olan sivil İHA’lar için bu durum düşünülerek uzak mesafedeyken dahi tespit işlemi gerçekleştirmek hedeflenmiştir.

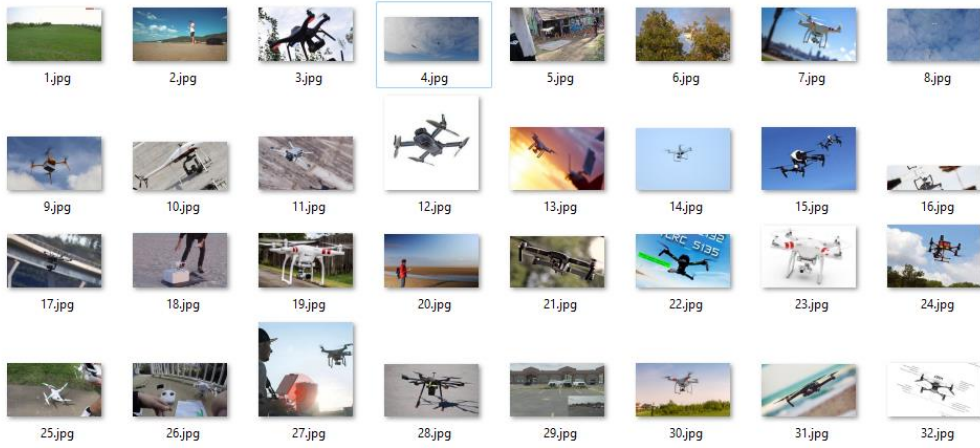


Resim 3.1 SAHI algoritması ile tespit örneği karşılaştırması (İnt. Kyn. 4)

3.3 Veri Seti

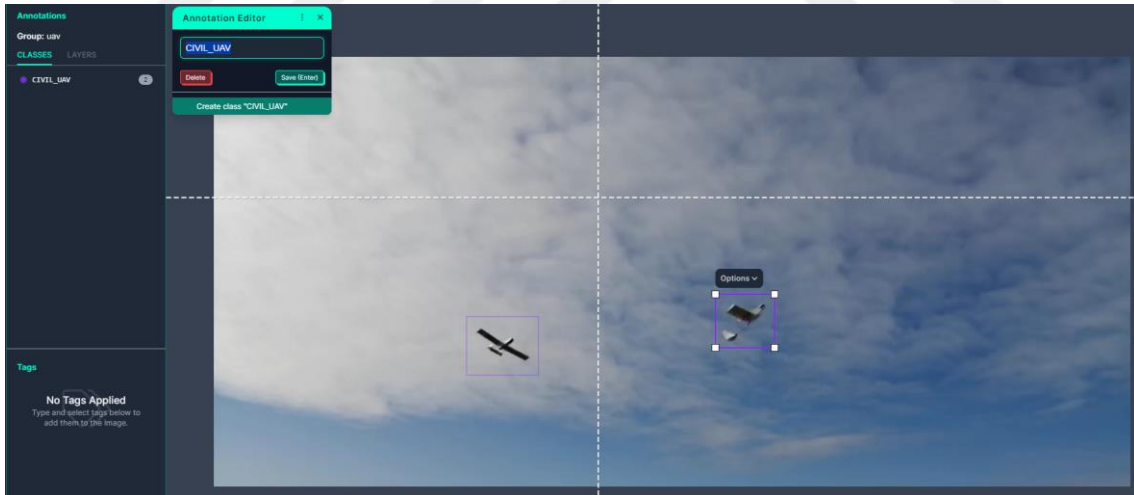
Eğitim aşamasından önce sivil İHA tespiti için gerekli veri setinin oluşturulma işlemi gerçekleştirilmiştir. İnternet üzerinde çeşitli platformlarda bulunan halka açık veri setleri taranarak kullanılacak YOLO mimarisi için uygunluğu kontrol edilmiştir. Açık kaynak olarak sunulan veri setlerine tarafımıza ait İHA görüntüleri de dahil edilerek veri setinin içeriği genişletilmiştir. Toplamda 2710 görsel ile oluşturularak Resim 3.2’de yer alan hem sabit kanat hem de döner kanat formundaki İHA’ların görüntüleri ile veri seti oluşturulmuştur.

Oluşturulan veri setindeki farklı görüntü formatına sahip görsellerin uzantıları düzenlenerek hepsi “JPG (Birleşik Fotoğraf Uzmanları Grubu)” formatına çevrilmiştir.



Resim 3.2 Sivil İHA tespiti amacı ile oluşturulan veri seti

JPG formattaki görseller Resim 3.3’deki “Roboflow” web sitesi ile etiketlenerek yapay zeka eğitimi için hazır hale getirilmiştir. Roboflow içerisinde etiketlenerek oluşturulan veri seti, diğer derin öğrenme modellerine adapte etmesi daha rahat olduğu için özellikle tercih edilmiştir. Tüm veri seti “CIVIL_UAV” olarak tek sınıf bulunacak şekilde etiketlenmiştir.



Resim 3.3 Roboflow web sitesi ile Sivil İHA tespiti için oluşturulan veri setindeki görselleri etiketleme işlemi

3.4 Yapay Zeka ile Eğitim

Oluşturulan veri seti eğitilecek derin öğrenme mimarisi için uygun hale getirilmiştir. Veri seti içerisine 3 ayrı klasör oluşturulmuştur: eğitim (train), test ve doğrulama (validation). Test klasörü içerisinde eğitimden sonra test aşamasında kullanılacak test görüntüleri ve

etiket dosyaları yer almaktadır. Eğitim klasörü içerisinde eğitim işlemi sırasında kullanılacak görüntüleri ve etiket dosyaları yer almaktadır. Doğrulama klasöründe ise eğitim işlemi sırasında her epoktan sonra gerçekleştirilecek doğrulama işlemi için etiket dosyaları yer almaktadır. Eğitimler Google Colab ve ASUS Tuf F17 dizüstü bilgisayarında farklı derin öğrenme modelleri ve farklı epok değerlerinde en optimum başarı oranını elde etmek için gerçekleştirilmiştir. Yapılan eğitimler sonucunda Şekil 3.14'deki çıktılar elde edilmiştir.

```
100 epochs completed in 0.324 hours.
Optimizer stripped from runs/train/yolov5s_100epoch_results/weights/last.pt, 14.3MB
Optimizer stripped from runs/train/yolov5s_100epoch_results/weights/best.pt, 14.3MB

Validating runs/train/yolov5s_100epoch_results/weights/best.pt...
Fusing layers...
YOLOv5s summary: 157 layers, 7012822 parameters, 0 gradients, 15.8 GFLOPs
      Class      Images  Instances      P      R      mAP50  mAP50-95: 100% 9/9 [00:03<00:00, 2.80it/s]
      all         281         289      0.966      0.92      0.958      0.623
Results saved to runs/train/yolov5s_100epoch_results
CPU times: user 17.1 s, sys: 2.53 s, total: 19.7 s
Wall time: 20min 2s
```

```
150 epochs completed in 0.706 hours.
Optimizer stripped from runs/train/yolov5l_150epoch_results/weights/last.pt, 92.7MB
Optimizer stripped from runs/train/yolov5l_150epoch_results/weights/best.pt, 92.7MB

Validating runs/train/yolov5l_150epoch_results/weights/best.pt...
Fusing layers...
YOLOv5l summary: 267 layers, 46108278 parameters, 0 gradients, 107.6 GFLOPs
      Class      Images  Instances      P      R      mAP50  mAP50-95: 100% 9/9 [00:03<00:00, 2.63it/s]
      all         281         289      0.976      0.927      0.964      0.646
Results saved to runs/train/yolov5l_150epoch_results
CPU times: user 31.7 s, sys: 4.84 s, total: 36.5 s
Wall time: 43min 3s
```

Şekil 3.14 Eğitim işlemi sonucunda elde edilen parametreler

3.5 PyQt5

PyQt5, Python programlama dili ile masaüstü uygulamaları geliştirmek amacıyla kullanılan bir grafik kullanıcı arayüzü kütüphanesidir. C++ tabanlı bir GUI (Grafiksel Kullanıcı Arayüzü) yani olan Qt'nin Python için sağlanmış bir bağlayıcısı olarak tanımlanmaktadır. PyQt5, çoklu platform desteği sunması ve geniş widget kütüphanesi ile zengin kullanıcı arayüzlerine sahip uygulamalar geliştirilmesine olanak tanımaktadır. Aynı zamanda Qt Designer aracı sayesinde görsel tasarıma dayalı arayüz oluşturma imkânı sunarak, yazılım geliştirme sürecini hızlandırmaktadır. PyQt5, masaüstü uygulamaları, veri görselleştirme araçları, grafiksel uygulamalar ve ağ tabanlı iletişim yazılımlarında sıklıkla kullanılmaktadır. Bunun yanı sıra, bilimsel simülasyonlar ve mühendislik uygulamalarında da tercih edilmektedir. Ayrıca, Qt Designer aracı ile

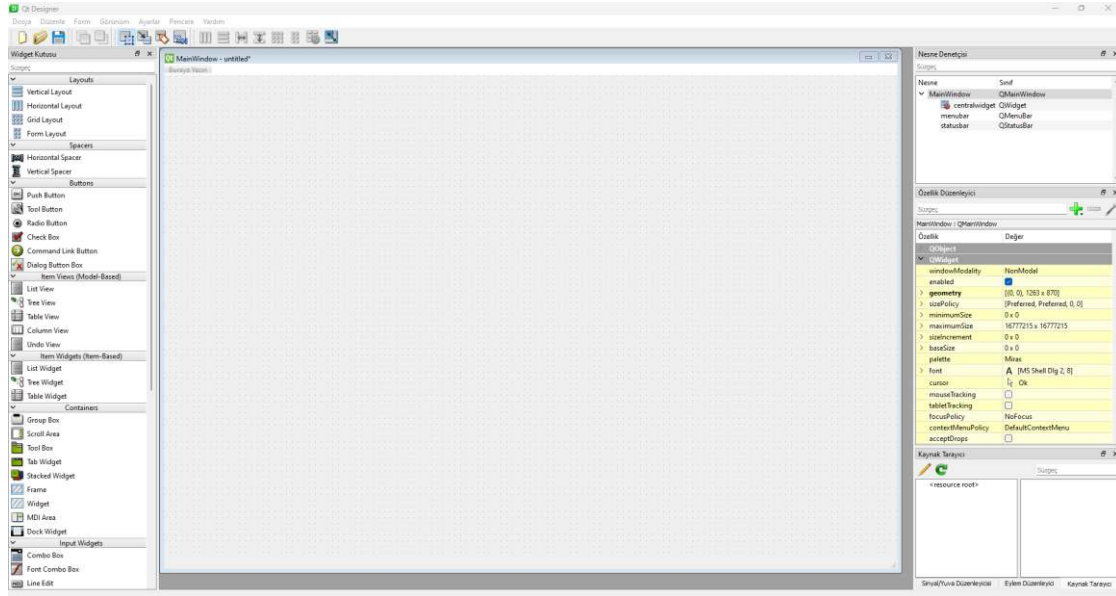
sürükle-bırak yöntemi kullanarak arayüz tasarımlarını daha hızlı ve kolay bir şekilde oluşturmak mümkündür. PyQt5'in modüler yapısı, farklı görevler için QtCore, QtGui, QtWidgets, QtNetwork ve QtMultimedia gibi çeşitli modülleri içermektedir. PyQt5, masaüstü uygulamaları, veritabanı yönetim araçları, grafiksel uygulamalar, oyunlar, simülasyon ve bilimsel uygulamalar ile ağ tabanlı iletişim sistemleri geliştirmek için geniş bir kullanım alanına sahiptir. Özellikle çoklu platform desteği, geniş widget seti ve gelişmiş API (Uygulama Programlama Arayüzü) desteği, PyQt5'i profesyonel yazılım geliştirme projelerinde tercih edilen bir araç haline getirmiştir. Günümüzde AI (Yapay Zeka) araçları için PyQt5'i kullanan çeşitli arayüzler geliştirilmiştir.

3.6 Qt Designer

Qt Designer, Qt framework'ü kullanarak grafiksel kullanıcı arayüzleri tasarlamak için kullanılan bir görsel tasarım aracıdır. Kullanıcı arayüzlerini kod yazmaya gerek kalmadan sürükle-bırak yöntemiyle oluşturmaya olanak tanımaktadır. Bu araç, özellikle PyQt ve PySide gibi Python bağlayıcıları ile birlikte kullanıldığında, arayüzlerin hızlı ve verimli bir şekilde tasarlanmasını sağlamaktadır.

Qt Designer, görsel arayüz tasarımı için kullanıcıya sunduğu çeşitli olanaklarla, özellikle programcılarının her bileşeni manuel olarak kodlamasına gerek bırakmadan, widget'ları (butonlar, etiketler, metin kutuları vb.) sürükle-bırak yöntemiyle hızlı bir şekilde arayüz Qt Designer ile oluşturabilmektedir. Bu özellik, geliştirme sürecini hızlandırırken hataları da en aza indirmektedir. Qt Designer, tasarlanan arayüzleri XML (Genişletilebilir İşaretleme Dili) tabanlı ".ui" dosyaları şeklinde kaydetmekte ve bu dosyalar PyQt veya PySide gibi Python kütüphaneleri aracılığıyla Python koduna dönüştürülerek, uygulamalarda kullanılabilir. Arayüzdeki widget'ların yerleşim düzenlerinin ve hiyerarşilerinin kolayca düzenlenmesine imkan tanıyan Qt Designer, aynı zamanda farklı ekran çözünürlüklerine uyum sağlayacak şekilde dinamik olarak boyutlandırılabilir arayüzler oluşturulmasını da desteklemektedir. Ayrıca, Qt'nin temelini oluşturan sinyal ve slot mekanizması, kullanıcı etkileşimlerini görsel olarak yönetmeyi mümkün kılarak Şekil 3.15.'deki arayüz ile, buton gibi bileşenlere belirli işlevlerin atanmasını sağlamaktadır. Qt Designer ile oluşturulan ".ui" dosyaları, hem Python hem de C++ projeleri ile entegre edilebilmektedir. PyQt gibi kütüphaneler sayesinde bu dosyalar ya

doğrudan çalıştırılabilir ya da Python koduna dönüştürülebilmektedir. Bu özellikler, Qt Designer'ı modern yazılım geliştirme süreçlerinde önemli bir araç haline getirmektedir.



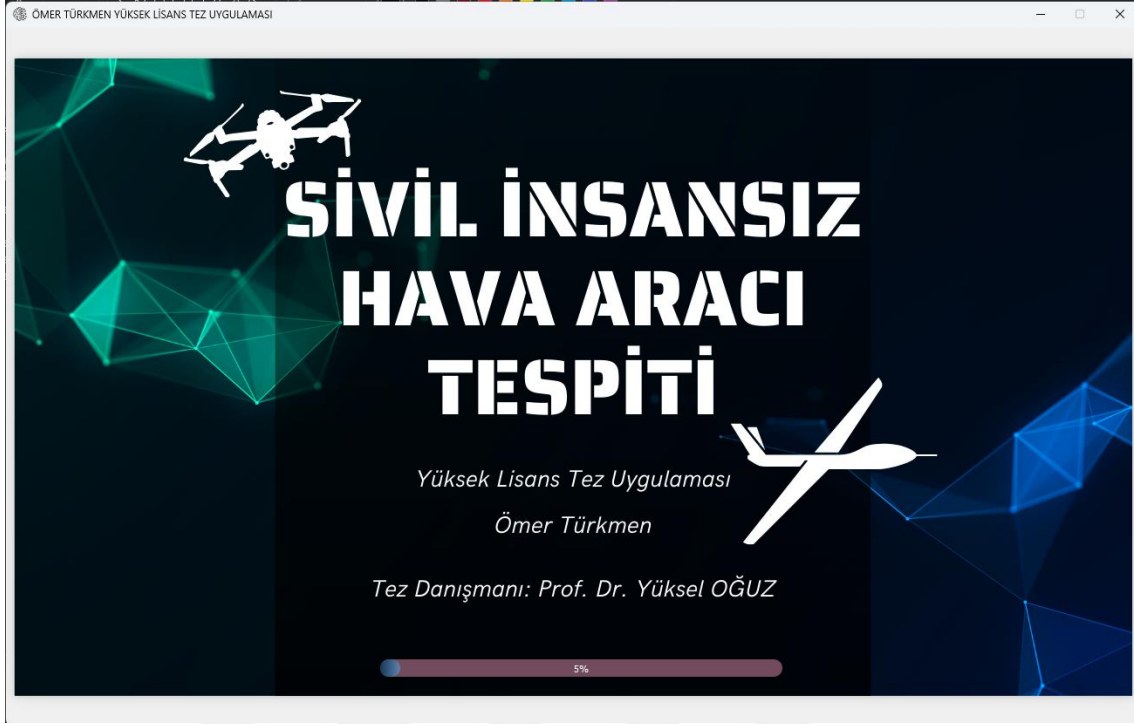
Şekil 3.15 Qt Designer arayüzü

Qt Designer, hem ticari hem de açık kaynak projelerde yaygın olarak kullanılmaktadır. Özellikle masaüstü yazılımlarının kullanıcı arayüzü tasarımında tercih edilmekte ve birçok sektörde, örneğin mühendislik, simülasyon, finans ve eğitim yazılımlarında kullanılabilmektedir. Sonuç olarak, Qt Designer, grafik kullanıcı arayüzü geliştirmeyi hızlı ve kolay hale getirdiği için özellikle PyQt gibi Python bağlayıcılarıyla birlikte kullanıldığında güçlü ve esnek bir araç olarak öne çıkmaktadır.

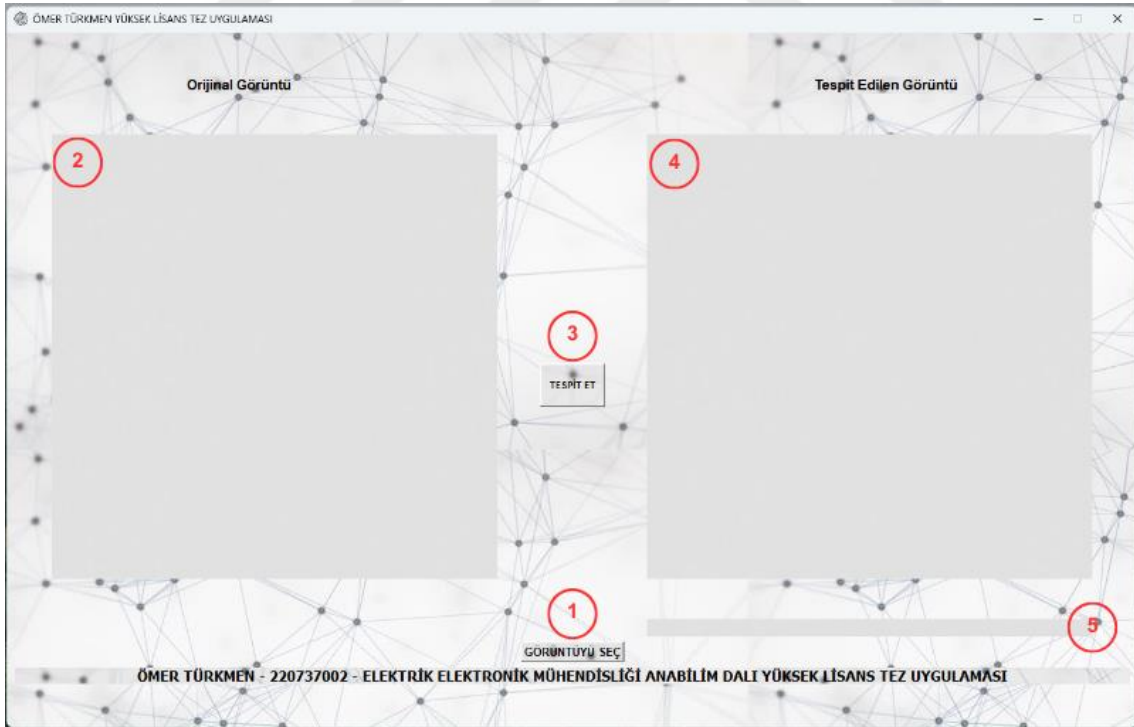
3.7 PyQt5 ve Qt Designer ile Masaüstü Uygulamasının Geliştirilmesi

Bu tez çalışmasında sivil İHA tespiti için önerilen yöntemin bir uygulamada entegrasyonunu göstermek için PyQt5 kütüphanesi ve Qt Designer kullanılarak bir masaüstü uygulaması geliştirilmiştir. Bu uygulama ise bir giriş ve tespit arayüzü olmak üzere iki arayüzden oluşmaktadır.

Geliştirilen arayüzlere ilişkin ekran görüntüleri de Şekil 3.16 ve Şekil 3.17'de gösterilmiştir.



Şekil 3.16 Geliştirilen uygulamanın giriş arayüzü



Şekil 3.17 Geliştirilen uygulamanın tespit arayüzü

Şekil 3.16 ve 3.17’de görüldüğü gibi, öncelikle geliştirilen uygulama açıldığına Şekil 3.16’daki ekran ön plana çıkmakta ve bu ekranda, uygulamanın içinde kullanılan AI

araçları ve gerekli diğer yazılım kütüphaneleri yüklenene kadar kullanıcı kısa süre bekletilmektedir. Sonrasında, kullanıcının karşısına Şekil 3.17’de görülen tespit arayüzü karşısına çıkmaktadır. Bir kullanıcı tarafından Şekil 3.17’de gösterilen tespit arayüzünün kullanımı için temel beş adım bulunmakta ve bu adımlarda gerçekleştirilen işlemlere ilişkin detaylar da aşağıda sırasıyla verilmiştir:

- 1- Bu buton ile kullanıcıdan İHA tespit edilmesini istediği görseli seçme işlemi yapılması sağlanmaktadır. Kullanıcı hangi görsel üzerinde işlem yapmak isterse o görsel seçimini yaparak seçimini tamamlamaktadır.
- 2- İlk adımda görsel seçim işlemi yapıldıktan sonra görsel bu alanda gösterilmektedir. Burada kullanıcının hem istediği görseli doğru seçip seçmediğini hem de tespit öncesi ve sonrası arasındaki farkı görebilmesi amaçlanmıştır..
- 3- Bu adımda kullanıcı butona bastığı zaman daha önceden uygulama içerisine kurulumu gerçekleştirilen derin öğrenme modeli ve bu modelin ağırlık dosyası ile tespit işlemini tespit işlemi gerçekleştirilmektedir.
- 4- Tespit butonuna basıldıktan sonra uygulama eğer sivil İHA tespit işlemi gerçekleştirirse bu alanda tespit edilmiş hali gösterilecektir. Eğer tespit işlemi gerçekleştirmezse de bu alanda tespit edemediği için görselin aynısı gösterilecektir.
- 5- Tespit işlemi sonucunda göre bu alanda tespit edildiği ya da tespit edilmediği bilgisi yer almaktadır.

4. BULGULAR

Yapılan çalışmada eğitim ve test sonuçları incelenmiş, YOLOv5, YOLOv7, YOLOv8 ve YOLOv10 mimarileri farklı epok ve modellerle eğitilerek test edilmiştir. Bu eğitim süreçleri sonucunda kesinlik, duyarlılık ve güven skoru değerleri elde edilmiştir. YOLOv5 mimarisi için EK 1, EK 2 ve EK 3'te kesinlik/güven skoru, duyarlılık/güven skoru ve genel güven skoru grafikleri sunulmuştur. YOLOv7'ye ait sonuçlar EK 4'te yer alırken, YOLOv8 için EK 5, EK 6 ve EK 7'de aynı metriklerin grafikleri bulunmaktadır. YOLOv10 mimarisine ait kesinlik, duyarlılık ve güven skoru grafikleri ise EK 8, EK 9 ve EK 10'da gösterilmiştir.

4.1 Performans Metrikleri

Performans metrikleri, bir sistemin, algoritmanın, uygulamanın veya modelin performansını ölçmek ve değerlendirmek için kullanılan çeşitli nicel göstergelerdir. Farklı alanlarda kullanılan performans metrikleri, değerlendirilen sisteme veya amaca göre değişiklik gösterebilmektedir. Sınıflandırma modellerinde kullanılan dört temel metrik vardır:

TP (True Positive, Doğru Pozitif): Gerçekten pozitif olan örneklerin model tarafından doğru bir şekilde pozitif olarak tahmin edilmesi durumudur.

TN (True Negative, Doğru Negatif): Gerçekten negatif olan örneklerin model tarafından doğru bir şekilde negatif olarak tahmin edilmesi durumudur.

FP (False Positive, Yanlış Pozitif): Gerçekten negatif olan örneklerin model tarafından yanlış bir şekilde pozitif olarak tahmin edilmesi durumudur.

FN (False Negative, Yanlış Negatif): Gerçekten pozitif olan örneklerin model tarafından yanlış bir şekilde negatif olarak tahmin edilmesi durumudur.

Bu dört metrik, sınıflandırma modelinin performansını değerlendirmede önemli rol oynar ve çeşitli performans ölçütlerinin hesaplanmasında kullanılır.

Doğruluk (Accuracy): Sınıflandırma problemlerinde, doğru olarak tahmin edilen örneklerin toplam örnek sayısına oranıdır. Doğruluk değerine ait denklem (4.1) aşağıda yer almaktadır.

$$Doğruluk = \frac{TP + TN}{TP + TN + FP + FN} \quad (4.1)$$

Hassasiyet (Precision): Denklem (4.2) de görüldüğü gibi pozitif olarak tahmin edilen örneklerin gerçekten pozitif olan örneklerine oranıdır. Yani, doğru pozitiflerin (TP) toplam pozitif tahminlerine (TP + FP) oranıdır. TP, FP ve FN, sırasıyla doğru pozitifler, yanlış pozitifler ve yanlış negatiflerdir.

$$Hassasiyet = \frac{TP}{TP + FP} \quad (4.2)$$

Duyarlılık (Recall): Gerçekten pozitif olan örneklerin doğru bir şekilde tahmin edilme oranıdır. TP'nin (doğru pozitifler) gerçek pozitiflerine (TP + FN) oranıdır. FN yanlış negatiflerdir. Aşağıdaki duyarlılığa ait denklem (4.3) yer almaktadır.

$$Duyarlılık = \frac{TP}{TP + FN} \quad (4.3)$$

Başarı Oranı (F1 Skoru): Hassasiyet ve duyarlılığın harmonik ortalamasıdır. Dengesiz sınıflandırma problemlerinde, F1 skoru genellikle doğruluk ölçümünden daha iyi bir performans göstergesi olabilir. Aşağıdaki denklem (4.4) de başarı oranı görülmektedir.

$$Başarı Oranı = 2 \times \frac{Hassasiyet \times Duyarlılık}{Hassasiyet + Duyarlılık} \quad (4.4)$$

YOLO derin öğrenme mimarileri ile farklı epok değerlerinde eğitimler gerçekleştirilmiştir. En optimum değerlerin 100 ve 150 epok da olduğu gözlemlenerek tüm eğitimlerde bu epok değerleri seçilmiştir. Eğitimler sonucunda elde edilen hassasiyet, duyarlılık ve başarı oranı tablosu aşağıda Çizelge 4.1'de görülmektedir.

Çizelge 4.1 incelendiği zaman farklı epok değerlerinde ve farklı derin öğrenme mimarilerinde eğitimler gerçekleştiği görülmektedir. YOLOv8_S modelinde 150 epokta %97,3 ile en iyi başarı oranı elde edilmiştir.

Çizelge 4.1 Tez çalışmasında karşılaştırılan mimarilerin eğitim sonuçları

Veri seti	Mimari	Epok	Hassasiyet	Duyarlılık	Başarı Oranı
Civil Uav Dataset	YOLOv5_N	100	0,985	0,896	0,938
		150	0,939	0,908	0,923
	YOLOv5_S	100	0,966	0,920	0,942
		150	0,961	0,946	0,953
	YOLOv5_M	100	0,971	0,931	0,951
		150	0,96	0,945	0,952
	YOLOv5_L	100	0,957	0,920	0,938
		150	0,976	0,927	0,951
	YOLOv5_X	100	0,949	0,938	0,943
		150	0,972	0,949	0,960
	YOLOv7	100	0,958	0,922	0,940
		150	0,971	0,93	0,950
	YOLOv8_N	100	0,986	0,950	0,968
		150	0,969	0,975	0,972
	YOLOv8_S	100	0,975	0,948	0,961
		150	0,988	0,958	0,973
	YOLOv8_M	100	0,979	0,960	0,969
		150	0,981	0,952	0,966
	YOLOv8_L	100	0,974	0,965	0,969
		150	0,976	0,967	0,971
	YOLOv8_X	100	0,973	0,955	0,964
		150	0,980	0,961	0,970
	YOLOv10_N	100	0,924	0,882	0,903
		150	0,935	0,945	0,940
	YOLOv10_S	100	0,939	0,931	0,935
		150	0,964	0,927	0,945
	YOLOv10_M	100	0,938	0,920	0,926
		150	0,961	0,943	0,952
	YOLOv10_L	100	0,973	0,900	0,935
		150	0,955	0,920	0,937

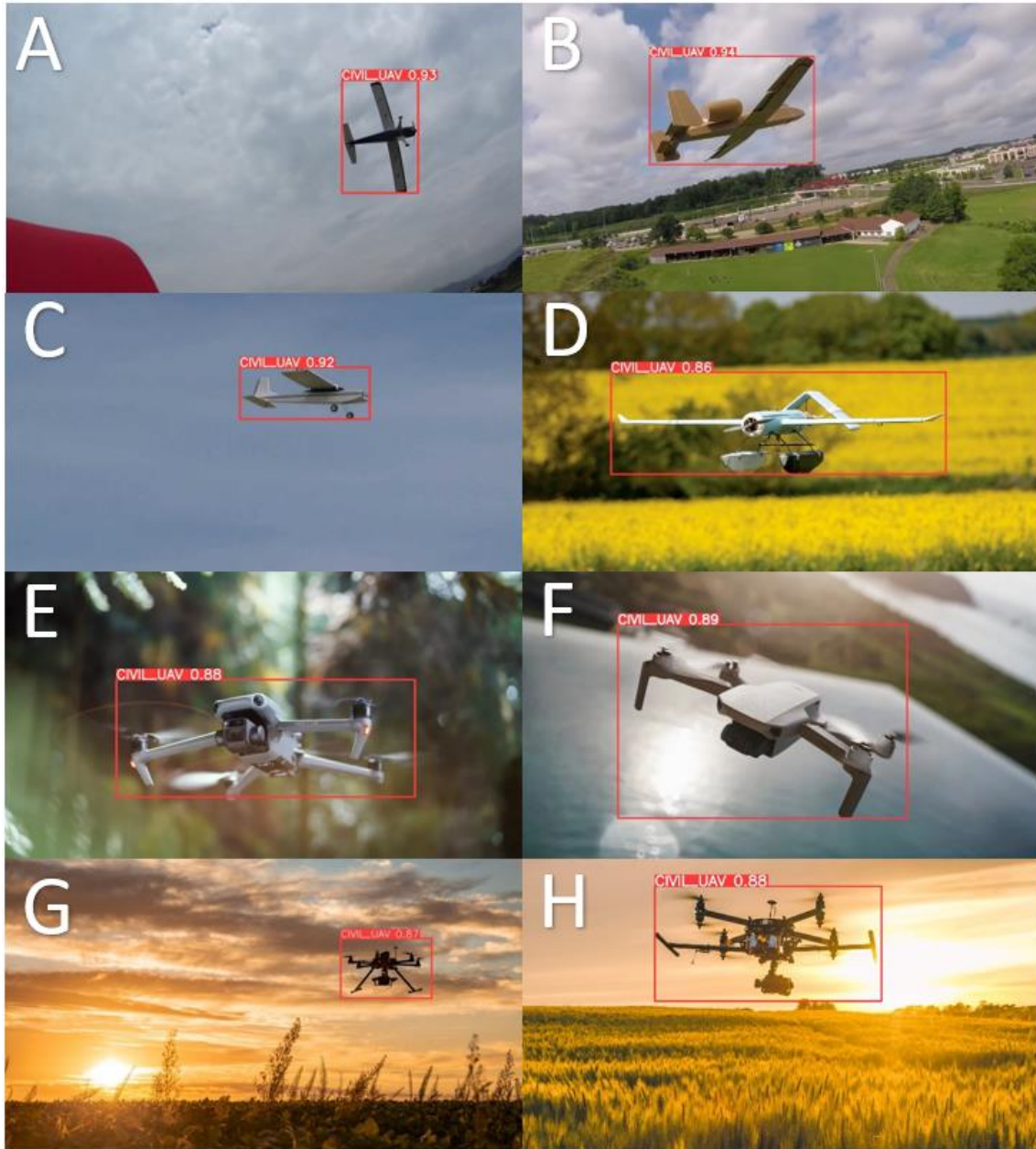
4.2 Model Test Sonuçları

Gerçekleştirilen eğitimler sonucunda birçok farklı test işlemleri gerçekleştirilmiştir. İlk aşamada model test olarak ayrılan veriseti kullanılarak sonuçları incelenmiştir. İkinci aşamada modele SAHI algoritması eklenerek sürü sivil İHA'lar ile tespit işlemi gerçekleştirilmiştir. Son aşamada ise geliştirilen uygulama ile uçuş esnasında çekilen hem

sabit kanat hem de döner kanat İHA'lara ait görüntüler ile derin öğrenme modeli ile test işlemleri gerçekleştirilmiştir.

4.2.1 En Başarılı Modelin Test Sonuçları

Eğitimler sonucunda en başarılı olan YOLOv8_S modeli farklı sivil İHA görselleri ile test edilmiştir.



Resim 4.1 YOLOv8 derin öğrenme sonucunda elde edilen ağırlık dosyasının görseller üzerindeki test görüntüleri

Test edilen görsellerin sonuçları yukarıdaki Resim 4.1’de de belirtildiği gibi görülmektedir. (A), (B), (C) ve (D) fotoğraflarında sabit kanatlı İHA’lara yer verilerek derin öğrenme modeli test edilmiştir. Bu test görsellerinde uçuş esnasında çekilen görseller kullanılarak gerçeğe en yakın sonuçların elde edilmesi amaçlanmıştır. (A) görselinde İHA manevra yaparken çekilen görüntüde İHA’nın alt kısmının çıkmasına rağmen derin öğrenme modeli %93 başarı oranına sahip tespit işlemi gerçekleştirmiştir. (B), (C) ve (D) görsellerinde uçuş esnasında çekilen görseller kullanılmıştır. Farklı tip İHA’lara ait görseller ile gerçekleştirilen test sonucunda YOLOv8 derin öğrenme modeli sırası ile %94, %92 ve %86 başarı oranına sahip sabit kanat sivil İHA’ları başarılı bir şekilde bularak tespit etmiştir. (E), (F), (G) ve (H) fotoğraflarında ise döner kanat formundaki görseller ile test işlemi gerçekleştirilmiştir. (E) test görselinde ormanlık alanda uçuş işlemini gerçekleştiren sivil İHA ile tespit işlemi yapılmıştır. Test sonucunda %88 doğruluk oranında tespit gerçekleşmiştir. (F) görselinde su üzerinde uçarken çekilen görselde ise %89 başarı oranı elde edilmiştir. (G) ve (H) görselleri gün batımında uçuşlarını gerçekleştiren sivil İHA görüntülerinden oluşmaktadır. Farklı formdaki sivil İHA’lar sırası ile %87 ve %88 doğrulukta tespit edilmiştir. Test görsellerinden elde edilen sonuçlar doğrultusunda modelin hem sabit kanat (uçak) hem de döner kanat (drone) formundaki İHA’ları başarılı bir şekilde tespit ederek eğitimin başarılı olduğunu doğrulanmıştır.

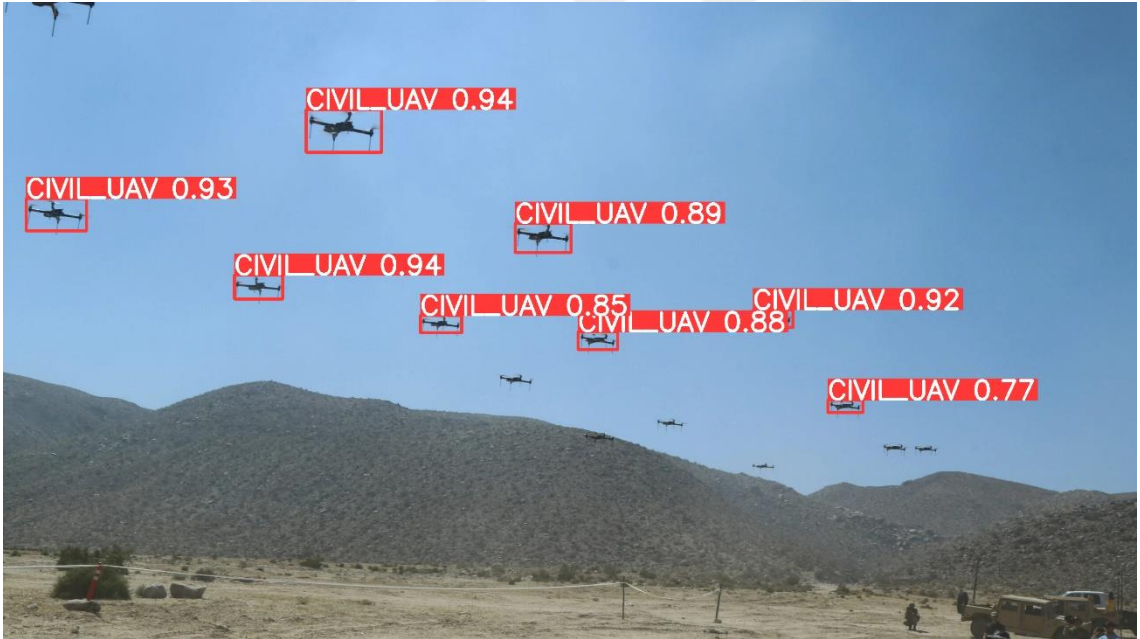
4.2.2 SAHI Algoritması ile Test Sonuçları

Test işlemlerinin ardından, en başarılı derin öğrenme mimarisi olarak belirlenen YOLOv8_S modeline SAHI (Slicing Aided Hyper Inference) algoritması eklenerek model yeniden test edilmiştir. Bu çalışmada, SAHI algoritması, YOLOv8_S modelinin doğruluğunu küçük nesne tespitinde daha da artırmak amacıyla entegre edilmiş ve uzaktan görülen İHA sürülerinin belirlenmesi için önemli bir katkı sağlamıştır.

Test sürecinde, internet ortamında yer alan sürü İHA’lara ait çeşitli görsellerden yararlanılmış olup, bunlar Resim 4.2’de örneklendiği gibi farklı açılardan ve çeşitli ölçeklerde elde edilen İHA görüntülerini içermektedir. Bu görseller, SAHI destekli YOLOv8_S modelinin küçük nesneler üzerindeki performansını değerlendirmek amacıyla kullanılmıştır.



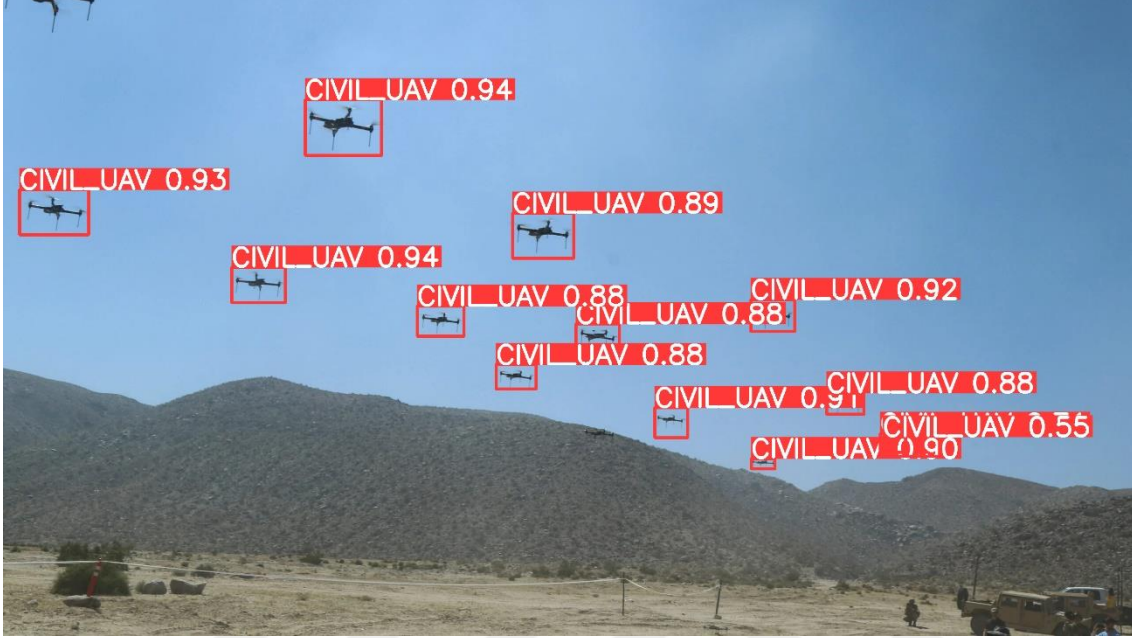
Resim 4.2 Sürü İHA'lara ait test görseli



Resim 4.3 SAHI algoritması eklenmeden YOLOv8_S modelinde test işlemi

Resim 4.3'te en başarılı sonuç veren YOLOv8_S modelinde test edilmiştir. Test işlemi sırasında ilk önce modele SAHI algoritması eklenmeden modelin kendisi test edilmiştir. Genel olarak görsel incelendiği zaman başarımlarının %77 ile %94 arasında değişkenlik gösterdiği gözlemlenmiştir. Aynı zamanda modelin diğer İHA'lara nispeten

daha uzak İHA'ları tespit edemediği de görülmektedir. Aynı görsel, aynı derin öğrenme modeline SAHI algoritması eklenerek test edilmiştir. Bu test işlemine ait sonuç aşağıdaki Resim 4.4' te yer almaktadır.

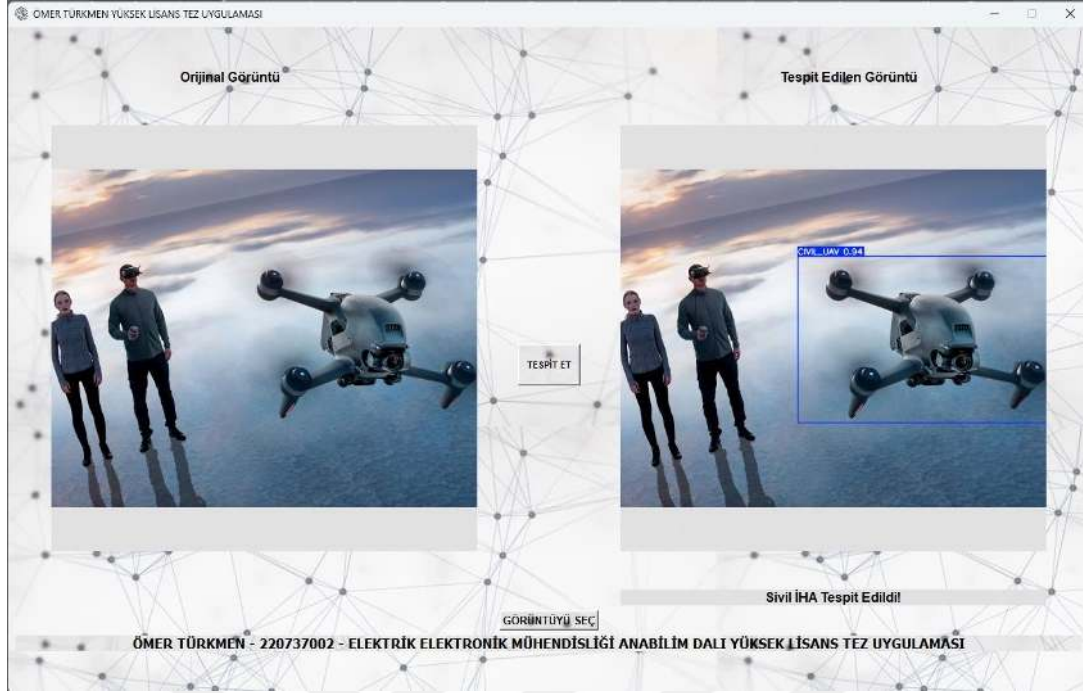


Resim 4.4 SAHI algoritması eklenerek gerçekleştirilen test işlemi

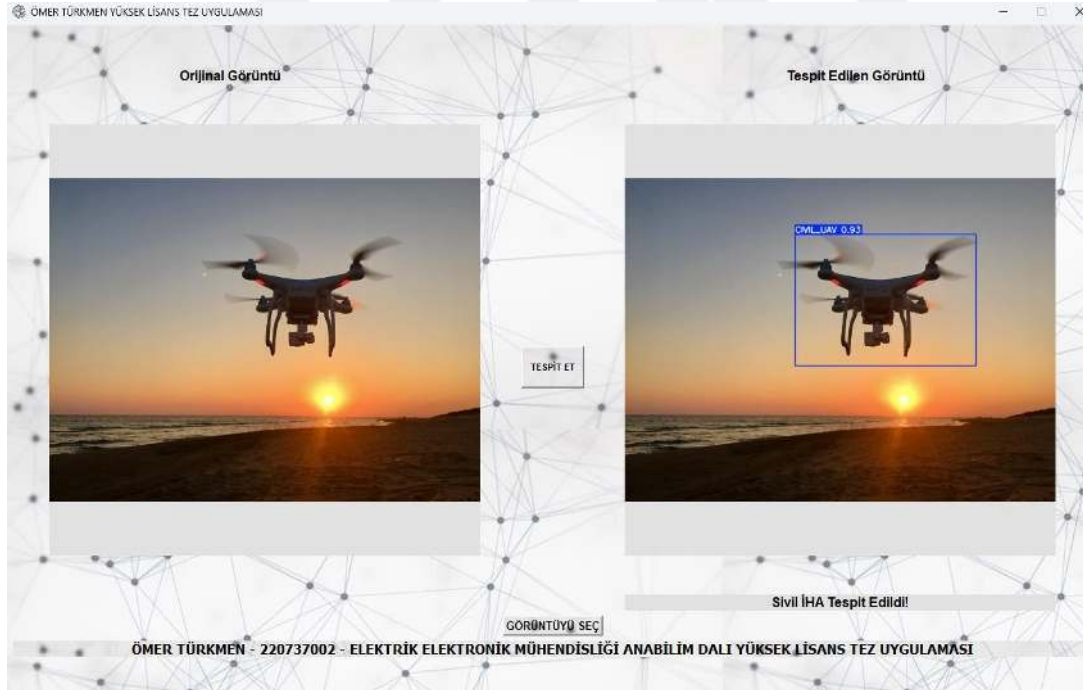
Resim 4.4'te SAHI algoritması YOLOv8_S modeline eklenerek test işlemi gerçekleştirilmiştir. Test işlemi incelendiği zaman algoritmanın modele SAHI algoritmasının eklenmemiş haline göre daha başarılı sonuç verdiği görülmüştür. SAHI algoritması önceki test işleminde biraz daha uzakta kalan İHA'ları da başarılı bir şekilde tespit etmiştir. Aynı zamanda önceki test işleminde başarılı bir şekilde tespit edilen İHA'ların da daha yüksek başarı oranı ile yeniden tespit edildiği de görülmüştür. İlerleyen çalışmalarda SAHI algoritmasının daha da geliştirilerek yer istasyonları gibi sistemlere entegre edilerek de daha hassas bir tespit işlemlerinin gerçekleştirilmesi de ön görülmektedir.

4.2.3 PyQt ve Qt Designer ile Geliştirilen Uygulamada Test Sonuçları

Tasarlanan programda en iyi sonucu veren derin öğrenme modeli olan YOLOv8_S ile sivil İHA görüntüleri test edilmiştir. Test sonuçları aşağıdaki resimlerde yer almaktadır.

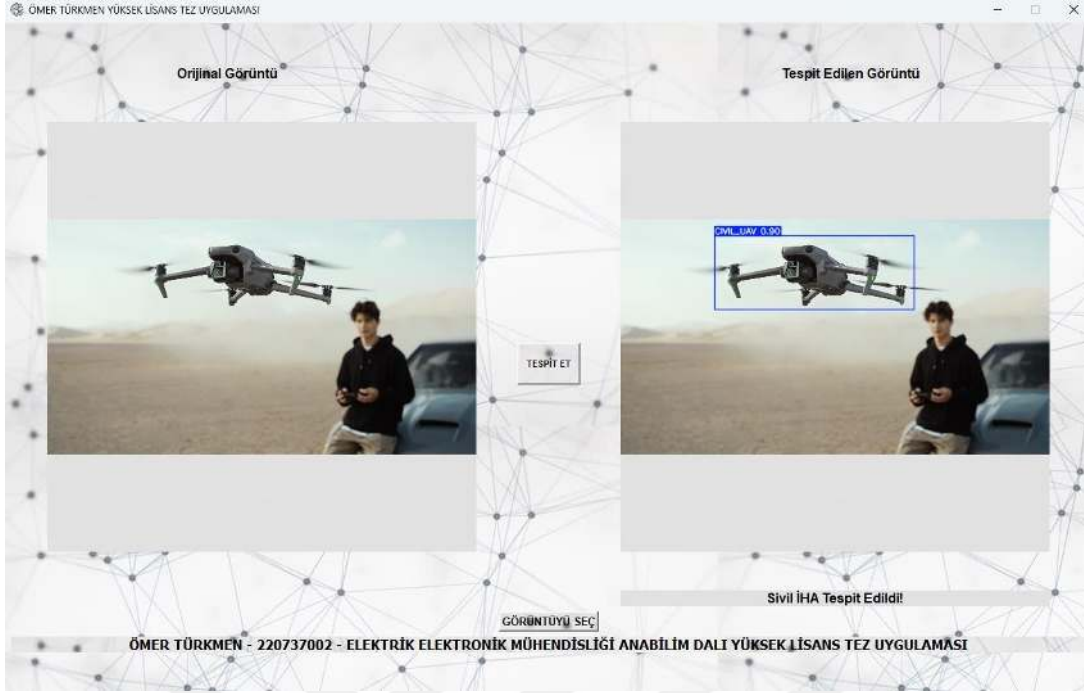


Resim 4.5 Masaüstü uygulamasında döner kanat formdaki İHA'ya ait test sonucu 1

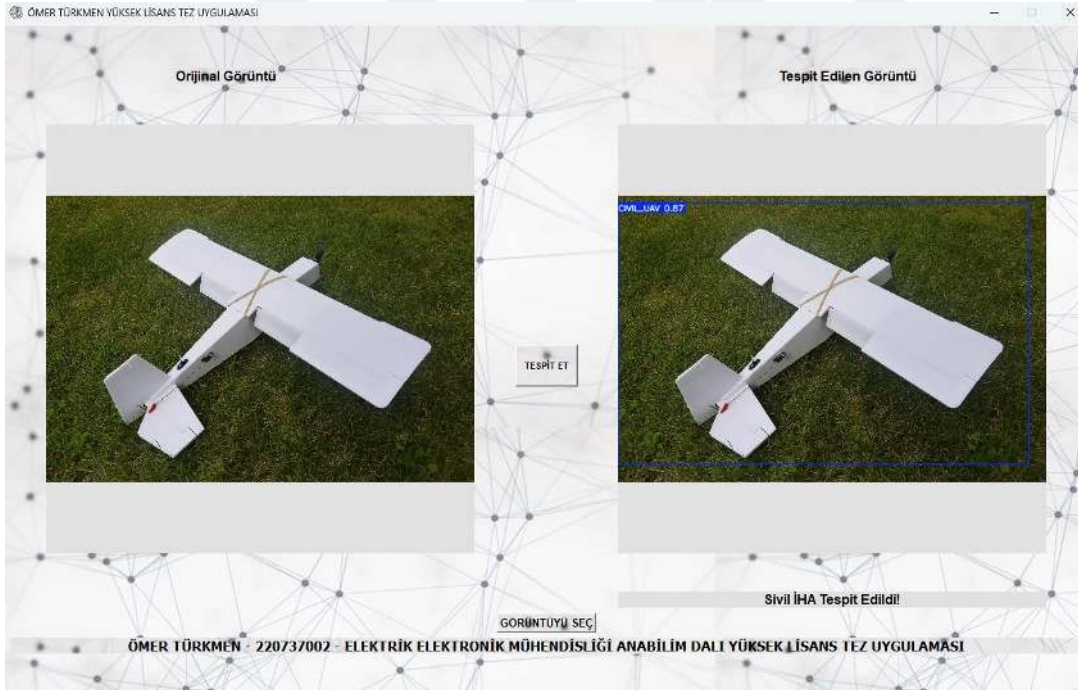


Resim 4.6 Masaüstü uygulamasında döner kanat formdaki İHA'ya ait test sonucu 2

Resim 4.5, 4.6 ve 4.7 başlıklarında uçuş esnasındaki döner kanat formda sivil İHA'lara ait görseller ile geliştirilen uygulama ile test işlemi gerçekleştirilmiştir.

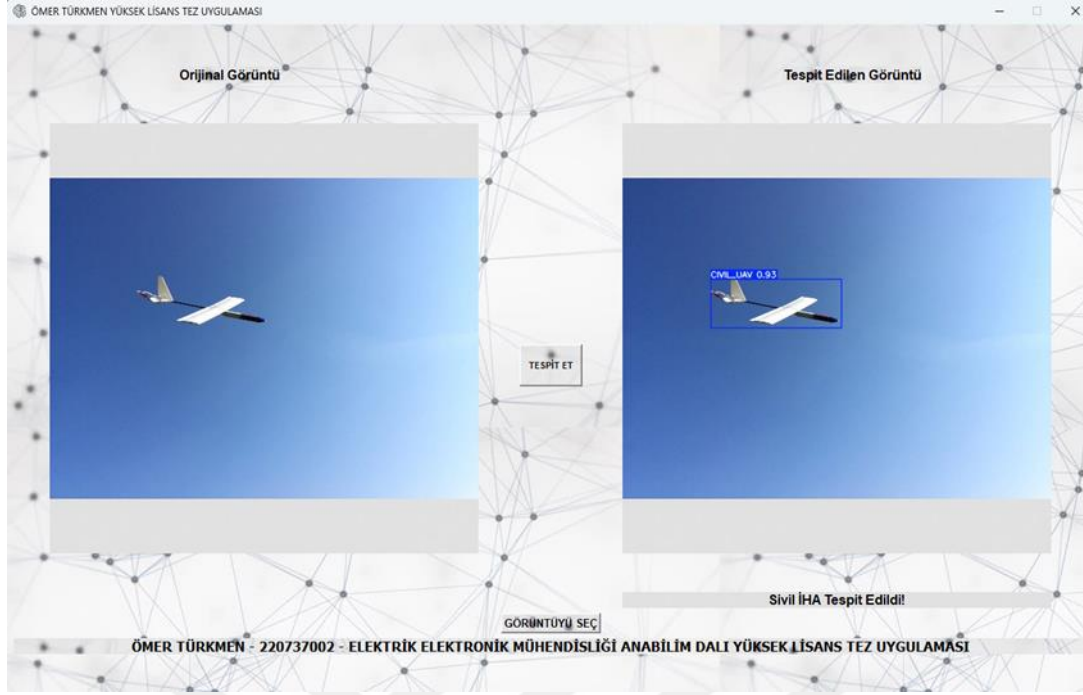


Resim 4.7 Masaüstü uygulamasında döner kanat formdaki İHA'ya ait test sonucu 3

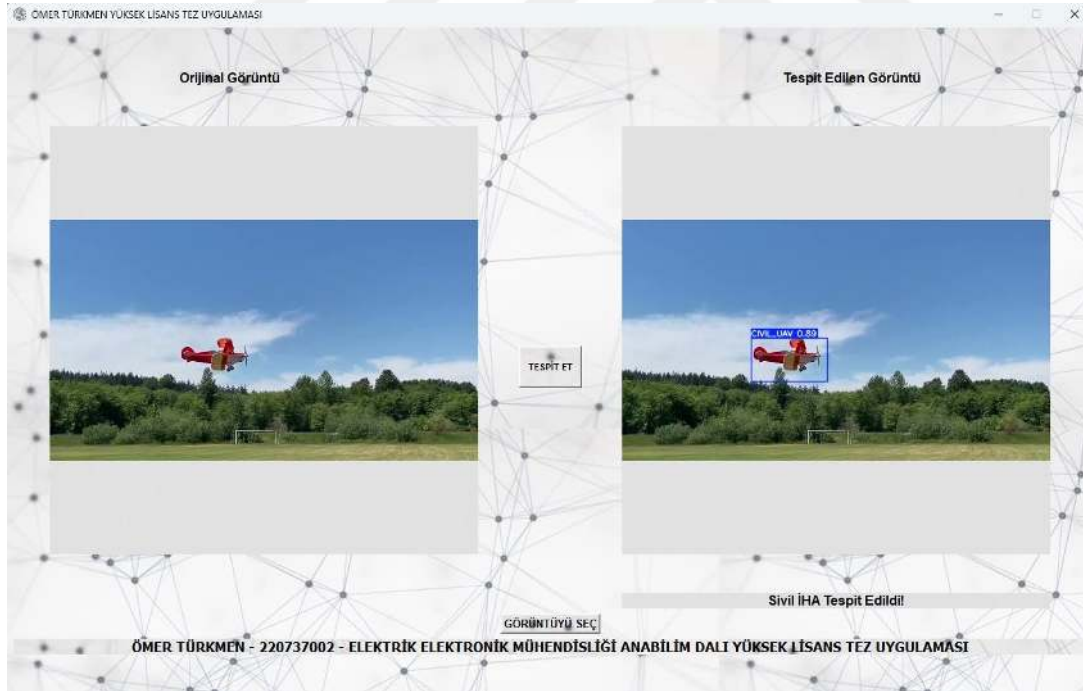


Resim 4.8 Masaüstü uygulamasında sabit kanat formdaki İHA'ya ait test sonucu 1

Resim 4.8, 4.9 ve 4.10'da ise uçuş yaparken görüntülenmiş sabit kanat formundaki sivil İHA'lara ait görseller ile denemeler gerçekleştirilmiştir

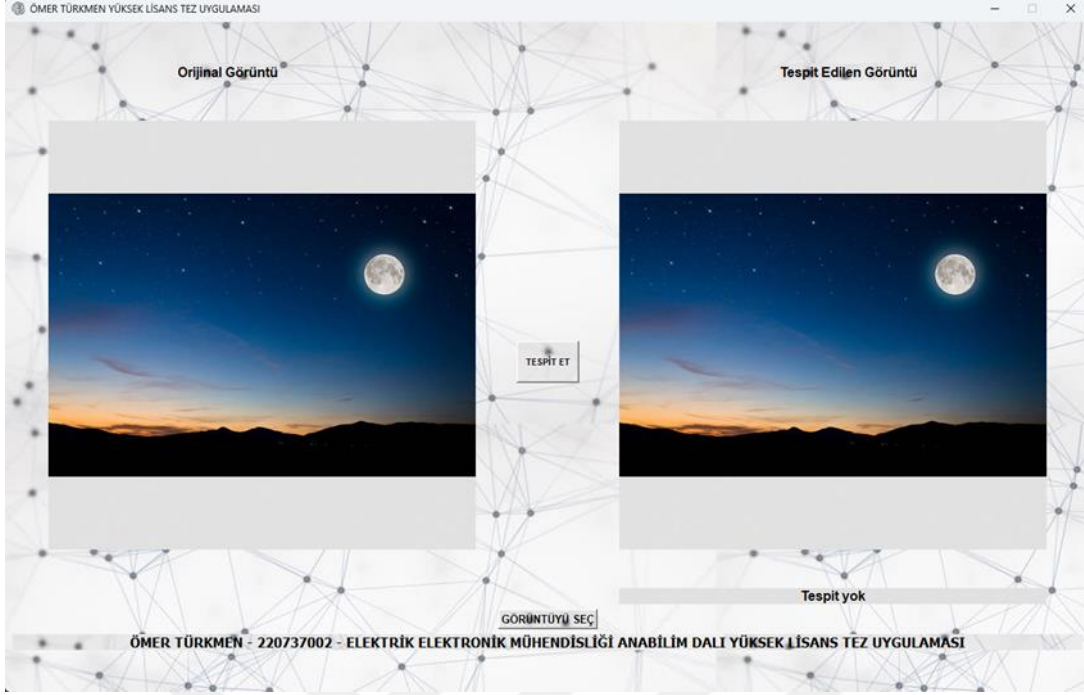


Resim 4.9 Masaüstü uygulamasında sabit kanat formdaki İHA'ya ait test sonucu 2



Resim 4.10 Masaüstü uygulamasında sabit kanat formdaki İHA'ya ait test sonucu 3

Resim 4.8, 4.9 ve 4.10'da ise uçuş yaparken görüntülenmiş sabit kanat formundaki sivil İHA'lara ait görseller ile denemeler gerçekleştirilmiştir.



Resim 4.11 Masaüstü uygulamasında sivil İHA olmadan gerçekleştirilen test sonucu

Döner kanat formundaki İHA'ları sırası ile %94, %91 ve %90 F-skor değerinde başarılı bir şekilde tespit etmiştir. Sabit kanat formundaki İHA'lara ait görüntülerde ise sırası ile %87, %93 ve %89 oranında başarı oranına sahip bir şekilde tespit edilmiştir. Resim 4.11'de ise herhangi bir şekilde sivil İHA bulunmayan gökyüzü görseli ile test edilerek tespit işleminin gerçekleşip gerçekleşmeyeceği denenmiştir. Deneme sonucunda herhangi bir şekilde tespit gerçekleşmemiş aynı zamanda da “Tespit yok” şeklinde uyarı da verilmiştir.

5. TARTIŞMA ve SONUÇ

Gerçekleştirilen bu tez çalışmasında, sivil alanda kullanılan sivil İHA'ların tespiti için YOLO derin öğrenme modelleri kullanılarak eğitimler yapılmıştır. Eğitim için kullanılan veri seti internet ortamındaki sivil İHA'ların görüntüleri, halka açık veri setleri ve tarafımıza ait olan İHA görüntüleri birleştirilerek oluşturulmuştur. Literatürde geçen son YOLO algoritmaları baz alınarak YOLOv5, YOLOv7, YOLOv8 ve YOLOv10 modelleri tercih edilmiştir. Eğitimler sonucunda en iyi başarı oranına sahip YOLOv8_S modeli 150 epokta %97 başarı oranı elde etmiştir. Aynı zamanda eğitim sonucunda elde edilen çıktılar kullanılarak masaüstü uygulaması geliştirilmiştir. PyQt5 ve Qt designer ile oluşturulan uygulamada modelin ağırlık dosyası ile tespit işlemi gerçekleştirilmektedir. Bu şekilde kullanıcı görseller üzerinden sivil İHA'ları bilgisayar üzerinde tespit edebilmektedir.

YOLOv8_S modelinin sivil İHA tespiti konusunda en iyi sonucu vermesi, modelin optimize edilmiş mimarisi ve yenilikçi teknik özellikleriyle doğrudan ilişkilidir. İlk olarak, YOLOv8'de kullanılan C3 dönüşüm yapısı, sahne içindeki nesnelerin ayrıntılı şekilde analiz edilmesini sağlamaktadır. Bu mimari yenilik, özellikle küçük ve uzamsal açıdan karmaşık nesnelerin, örneğin İHA'ların, tespitini daha başarılı hale getirmektedir. Ayrıca, YOLOv8_S modelinin parametre sayısının daha az olması, eğitim sürecinde daha hızlı öğrenme sağlamanın yanı sıra küçük nesneleri daha etkili şekilde algılamasına olanak tanıyarak hareketli ve nispeten küçük boyuttaki İHA'ların tespitinde önemli bir avantaj sağlamaktadır.

Buna ek olarak, YOLOv8'de geliştirilen yeni özellik çıkarım katmanları, doğruluk ile hızlı çıkarım arasındaki dengeyi optimize etmektedir. Bu özellik sayesinde model, farklı açılardan ve hızlarda görülen İHA'ları daha yüksek doğrulukla tespit edebilmektedir. Modelin kompakt yapısı, çıkarım işlemlerinde hızı artırarak eğitim sürecinde daha fazla parametrenin etkin şekilde optimize edilmesine imkân tanıyarak daha kararlı sonuçlar elde edilmesini sağlamaktadır. Sonuç olarak, bu teknik avantajlar, YOLOv8_s modelinin sivil İHA tespiti gibi uygulamalarda diğer modellere göre daha üstün performans göstermesini sağlamaktadır.

Çizelge 5.1 Derin öğrenme modelleri ile İHA tespit çalışmalarının karşılaştırılması

	Tespit	Yöntem	Başarı Oranı
Tez çalışması	İHA (Sabit ve döner kanat)	YOLOv8	%97
Aydin ve Singha (2023)	İHA ve Kuş	YOLOv5	%89
Ashraf vd. (2021)	İHA	ResNet50	%92
Khan vd. (2023)	İHA ve Kuş	MFNet	%96
Sun vd. (2020)	İHA	Deeper-SSD	%0,90 mAP
Cheng vd. (2023)	İHA	Fast-YOLOv4	%90,62 mAP

Çizelge 5.1’de görüldüğü gibi literatürdeki çalışmalar gerçekleştirilen tez çalışması kapsamında karşılaştırılmış ve sonuçları verilmiştir. Aydin ve Singha (2023) yapmış oldukları çalışmada YOLOv5 modelini kullanarak YOLOv4 ile daha önce yapmış oldukları çalışma ile karşılaştırma yapmışlardır. İHA ve kuş olarak belirledikleri iki sınıfı tespit etmişlerdir. Daha kaliteli bir sonuç olması için hiperparametreler üzerinde değişiklik yapmışlardır. Eğitim sonucunda %89 değerine sahip f-skor elde etmişlerdir. Aynı zamanda DJI droneler ile çekilmiş görüntüler üzerinde algılama hızını tespit etmişlerdir. Ashraf vd. (2021) ise çalışmalarında ResNet50 mimarisi kullanarak iki aşamalı segmentasyon işlemi ile İHA tespit işlemi gerçekleştirmişlerdir. ResNet50 modeli ile segmentasyon işlemi ardından olası tahminlere dayalı olarak piksel tespit işlemi gerçekleştirmişlerdir. Çalışma sonucunda %92 oranında başarı elde etmişlerdir. Khan vd. (2023) gerçekleştirdikleri çalışmada YOLOv5 mimarisine yerine MFNet mimarisini önermişlerdir. 5100 fotoğraftan oluşan veri setinde İHA ve kuş tespiti gerçekleştirmişlerdir. Önerdikleri modelde %96 değerinde başarı elde etmişlerdir. Sun vd. (2020) Deeper-SSD modeli ile küçük nesneleri tespit etmek için çalışma gerçekleştirmişlerdir. Standart SSD mimarisini özelleştirerek daha küçük nesneleri tespit etmeyi amaçlamışlardır. Standart SSD mimarisindeki mAP %88,93 değerini Deeper-SSD ile %90 seviyesine ulaştırmışlardır. Geliştirdikleri mimari ile de küçük boyuttaki droneleri tespit etmişlerdir. Cheng vd. (2023) geliştirdikleri mimaride standart YOLOv4 derin öğrenme mimarisini MobileNetV3 modelini kullanarak Fast-YOLOv4 modeli elde etmişlerdir. Hızlı İHA tespiti için geliştirilen sistemde 90,62 mAP değeri elde edilerek İHA tespiti gerçekleştirilmiştir.

Literatürdeki çalışmalar incelendiğinde genel olarak drone formundaki İHA’lara yönelik

alıřmaların mevcut olduėu grlmektedir. Fakat son zamanlarda kullanılan sabit kanat formundaki harp bařlıklı İHA'lar da aktif olarak savunma ya da saldırı amacı ile kullanılmaktadır. Bu yzden tespit edilmesi de nem arz etmektedir. Ayrıca literatrdeki nerilerin mimarilerinin bařarım oranı alıřmada gerekleřtirilen YOLOv8_S modelindeki bařarıdan alt seviyede kalmaktadır. Aynı zamanda literatrdeki alıřmalarda masast uygulamaları mevcut deėildir.

Bu tez alıřmanın son zamanlarda sivil İHA'ların kullanım oranının artması ile doėabilecek tehditlere karřı alt yapı oluřturması planlanmaktadır. Artan kullanım oranı ile birlikte doėabilecek zaafiyetler gz nne alındıėı zaman savunma alanında kayıpların en aza indirilmesi nemli bir ama olmaktadır. Aynı zamanda bu tez alıřmasının gelecekte mimari ile entegre sistem geliřtirilerek savunma sanayinde kullanılabilecek yer istasyonları aısından nclk edeceėi dřnlmektedir.

6. KAYNAKLAR

- Akyon F C, Cengiz C, Altınuc S O, Cavusoglu D, Sahin K, Eryuksel O, 2021, SAHI: A lightweight vision library for performing large scale object detection and instance segmentation, Zenodo, <https://doi.org/10.5281/zenodo.5718950>
- AlKhonaini A, Sheltami T, Mahmoud A, Imam M, 2024, UAV Detection Using Reinforcement Learning, *Sensors*, 24(6), <https://doi.org/10.3390/s24061870>
- Al-IQubaydhi N, Alenezi A, Alanazi T, Senyor A, Alanezi N, Alotaibi B, Alotaibi M, Razaque A, Hariri S, 2024, Deep learning for unmanned aerial vehicles detection: A review *Computer Science Review* (C. 51), Elsevier, <https://doi.org/10.1016/j.cosrev.2023.100614>
- Ashraf M W, Sultani W, Shah M, 2021, Dogfight: Detecting Drones from Drones Videos
- Aydin B, Singha S, 2023, Drone Detection Using YOLOv5 *Eng* 4(1), 416-433, <https://doi.org/10.3390/eng4010025>
- Björklund S, 2018, Target Detection and Classification of Small Drones by Boosting on Radar Micro-Doppler, 2018 15th European Radar Conference (EuRAD), 182-185, <https://doi.org/10.23919/EuRAD.2018.8546569>
- Chávez K, Swed O, 2023, Emulating underdogs: Tactical drones in the Russia-Ukraine war, *Contemporary Security Policy*, 44(4), 592-605, <https://doi.org/10.1080/13523260.2023.2257964>
- Cheng Q, Wang H, Zhu B, Shi Y, Xie B, 2023, A Real-Time UAV Target Detection Algorithm Based on Edge Computing, *Drones*, 7(2), <https://doi.org/10.3390/drones7020095>
- Dadboud F, Patel V, Mehta V, Bolic M, Mantegh I, 2021, Single-Stage UAV Detection and Classification with YOLOV5: Mosaic Data Augmentation and PANet, AVSS 2021- 17th IEEE International Conference on Advanced Video and Signal-Based Surveillance, <https://doi.org/10.1109/AVSS52988.2021.9663841>
- Delleji T, Fekih H, Chtourou Z, 2020, Deep Learning-based approach for detection and classification of Micro/Mini drones, *Proceedings of the International Conference*

- on Advanced Systems and Emergent Technologies, IC_ASET 2020, 332-337, https://doi.org/10.1109/IC_ASET49463.2020.9318281
- Huang L, Liu G, Wang Y, Yuan H, Chen T, 2022, Fire detection in video surveillances using convolutional neural networks and wavelet transform, *Engineering Applications of Artificial Intelligence*, 110, 104737, <https://doi.org/10.1016/j.engappai.2022.104737>
- Kahveci M, Can N, 2017, İnsansız Hava Araçları: Tarihçesi, Tanımı, Dünyada ve Türkiye'deki Yasal Durumu, *Selcuk University Journal of Engineering, Science and Technology*, 5(4), 511-535, <https://doi.org/10.15317/scitech.2017.109>
- Khan M U, Dil M, Alam M Z, Orakazi F A, Almasoud A M, Kaleem Z, Yuen C, 2023, SafeSpace MFNet: Precise and Efficient MultiFeature Drone Detection Network, *IEEE Transactions on Vehicular Technology*, <https://doi.org/10.1109/TVT.2023.3323313>
- King A, 2024, Robot wars: Autonomous drone swarms and the battlefield of the future, *Journal of Strategic Studies*, 1-29, <https://doi.org/10.1080/01402390.2024.2302585>
- Kunertova D, 2023, The war in Ukraine shows the game-changing effect of drones depends on the game, *Bulletin of the Atomic Scientists*, 79(2), 95-102, <https://doi.org/10.1080/00963402.2023.2178180>
- Rabiu L, Ahmad A, Gohari A, 2024, Advancements of Unmanned Aerial Vehicle Technology in the Realm of Applied Sciences and Engineering: A Review, *Journal of Advanced Research in Applied Sciences and Engineering Technology*, 40(2), 74-95, <https://doi.org/10.37934/araset.40.2.7495>
- Li G, Fu L, Gao C, Fang W, Zhao G, Shi F, Dhupia J, Zhao K, Li R, Cui Y, 2022, Multi-class detection of kiwifruit flower and its distribution identification in orchard based on YOLOv5l and Euclidean distance, *Computers and Electronics in Agriculture*, 201, 107342, <https://doi.org/10.1016/j.compag.2022.107342>
- Mogili U R, Deepak B B V L, 2018, Review on Application of Drone Systems in Precision Agriculture *Procedia Computer Science*, 133, 502-509, <https://doi.org/10.1016/j.procs.2018.07.063>

- Nandhini T J, Thinakaran K, 2023, Object Detection Algorithm Based on Multi-Scaled Convolutional Neural Networks, 2023 3rd International conference on Artificial Intelligence and Signal Processing (AISP), 1-5, <https://doi.org/10.1109/AISP57993.2023.10134980>
- Özer T, Türkmen Ö, 2023, Low-cost AI-based solar panel detection drone design and implementation for solar power systems, *Robotic Intelligence and Automation*, 43(6), 605-624, <https://doi.org/10.1108/RIA-03-2023-0022>
- Özer T, Türkmen Ö, 2024, An approach based on deep learning methods to detect the condition of solar panels in solar power plants, *Computers and Electrical Engineering*, 116, <https://doi.org/10.1016/j.compeleceng.2024.109143>
- Peng S, Huang F, Qiang X, Chen S, He W, Ma L, 2024, Research on a near real-time regional change detection system of UAV remote sensing images based on embedded technology, *Multimedia Tools and Applications*, <https://doi.org/10.1007/s11042-024-18402-w>
- Prabhu A, Liu X, Spasojevic I, Wu Y, Shao Y, Ong D, Lei J, Green P C, Chaudhari P, Kumar V, 2024, UAVs for forestry: Metric-semantic mapping and diameter estimation with autonomous aerial robots, *Mechanical Systems and Signal Processing*, 208, 111050, <https://doi.org/10.1016/j.ymssp.2023.111050>
- Rostami M, Farajollahi A, Parvin H, 2024, Deep learning-based face detection and recognition on drones, *Journal of Ambient Intelligence and Humanized Computing*, 15(1), 373-387, <https://doi.org/10.1007/s12652-022-03897-8>
- Su J, Zhu X, Li S, Chen W H, 2023, AI meets UAVs: A survey on AI empowered UAV perception systems for precision agriculture, *Neurocomputing*, 518, 242-270, <https://doi.org/https://doi.org/10.1016/j.neucom.2022.11.020>
- Sun H, Geng W, Shen J, Liu N, Liang D, Zhou H, 2020, Deeper SSD: Simultaneous up-sampling and down-sampling for drone detection, *KSII Transactions on Internet and Information Systems*, 14(12), 4795-4815, <https://doi.org/10.3837/tiis.2020.12.010>
- Tang P, Li J, Sun H, 2024, A Review of Electric UAV Visual Detection and Navigation Technologies for Emergency Rescue Missions. *Sustainability*, 16(5)

<https://doi.org/10.3390/su16052105>

Udenau G, Dobrescu A, Oltean M, 2016, Unmanned Aerial Vehicle In Military Operations, Scientific Research And Education In The Air Force, 18(1), 199-206, <https://doi.org/10.19062/2247-3173.2016.18.1.26>

Valaboju R, Vaishnavi, Harshitha C, Kallam A R, Babu B S, 2023, Drone Detection and Classification using Computer Vision, 7th International Conference on Trends in Electronics and Informatics, ICOEI 2023 - Proceedings, 1320-1328, <https://doi.org/10.1109/ICOEI56765.2023.10125737>

Wan L, Moan T, Gao Z, Shi W, 2024, A review on the technical development of combined wind and wave energy conversion systems. Energy, 130885, <https://doi.org/https://doi.org/10.1016/j.energy.2024.130885>

Ye T, Qin W, Zhao Z, Gao X, Deng X, Ouyang Y, 2023, Real-Time Object Detection Network in UAV-Vision Based on CNN and Transformer, IEEE Transactions on Instrumentation and Measurement, 72, <https://doi.org/10.1109/TIM.2023.3241825>

Zhai X, Huang Z, Li T, Liu H, Wang S, 2023, YOLO-Drone: An Optimized YOLOv8 Network for Tiny UAV Object Detection, Electronics (Switzerland), 12(17), <https://doi.org/10.3390/electronics12173664>

Zhang H, Shao F, He X, Zhang Z, Cai Y, Bi S, 2023, Research on Object Detection and Recognition Method for UAV Aerial Images Based on Improved YOLOv5, Drones, 7(6). <https://doi.org/10.3390/drones7060402>

İnternet Kaynakları

1- <https://www.kaggle.com/datasets/diyorbek0024/drone-detect-data>, 28.05.2024

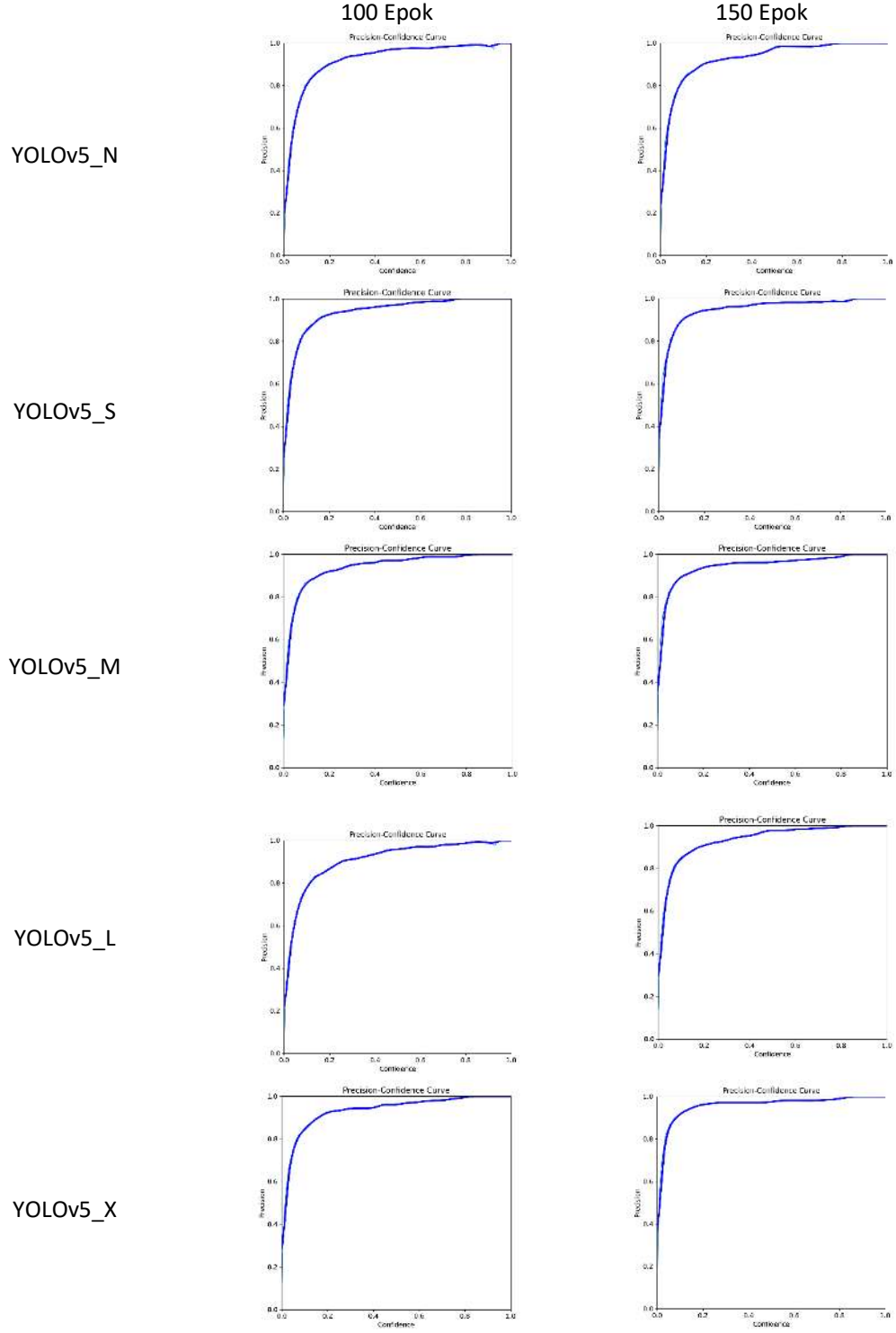
2- <https://tr.d2l.ai/index.html#> , 12.2021

3- <https://docs.ultralytics.com/tr>, 24.09.2024

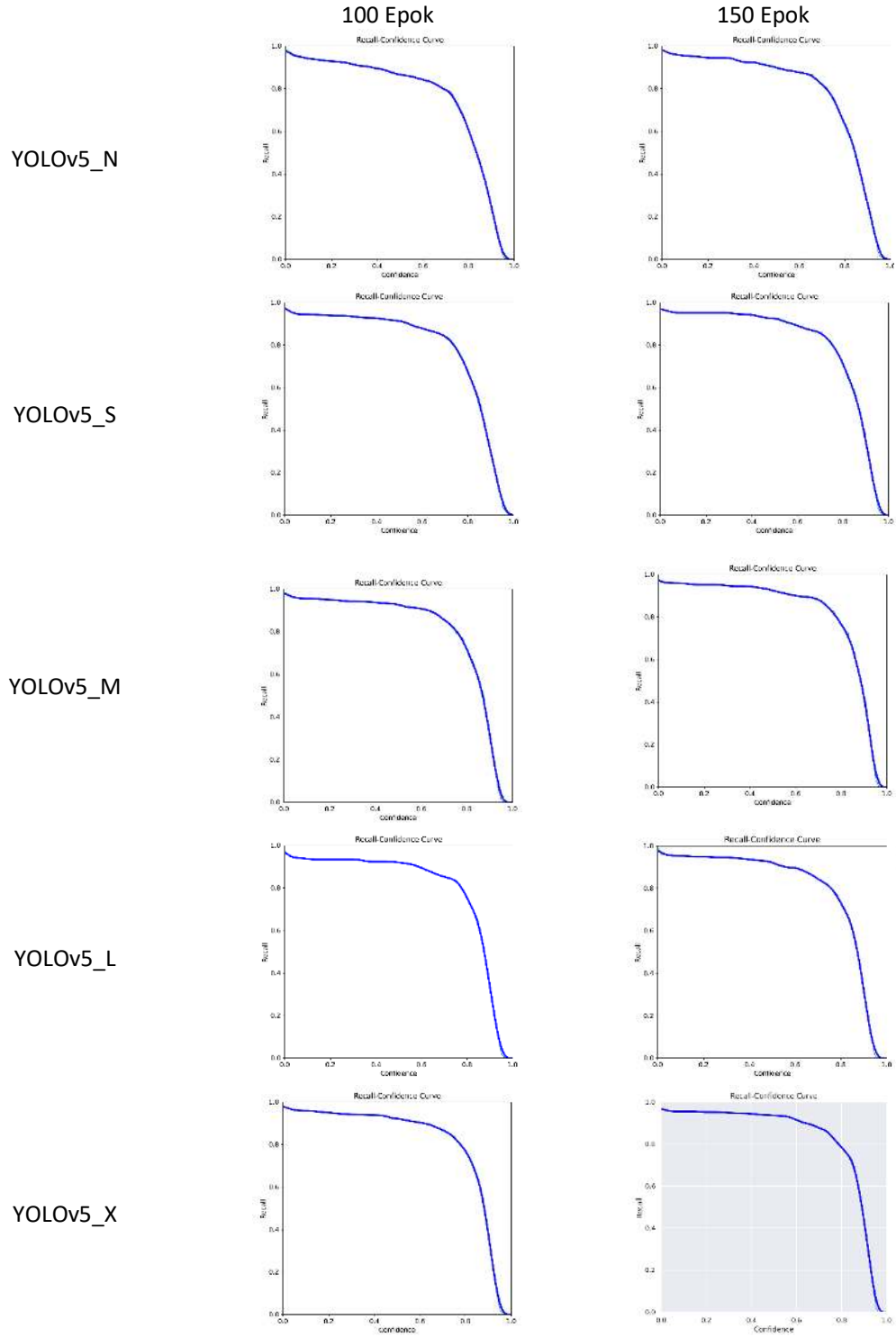
4- <https://docs.ultralytics.com/tr/guides/sahi-tiled-inference/>, 01.01.2024

EKLER

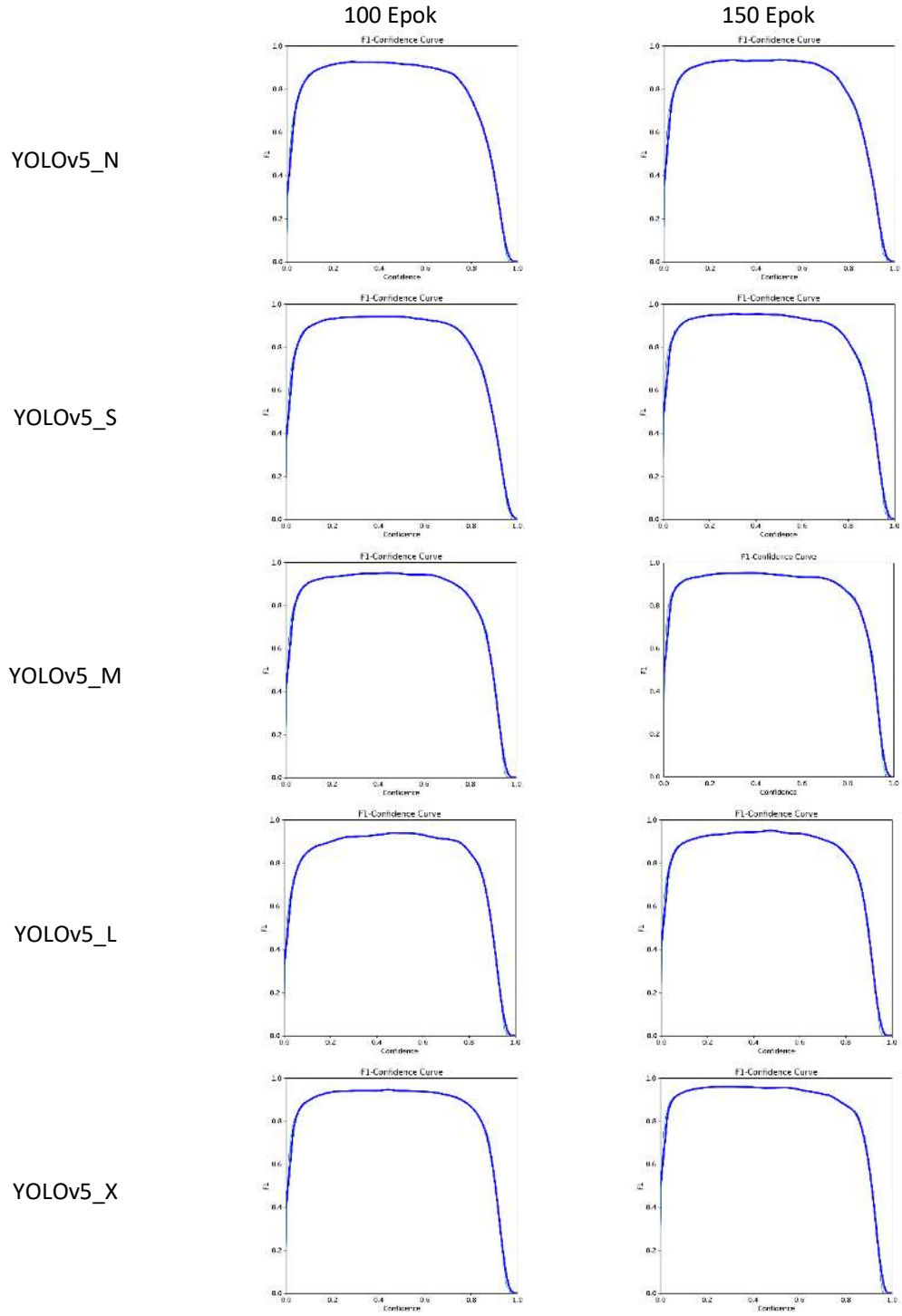
EK 1. YOLOv5 Eğitimleri Sonucunda Elde Edilen Hassasiyet/Başarı Oranı Grafikleri



EK 2. YOLOv5 Eğitimleri Sonucunda Elde Edilen Duyarlılık/Başarı Oranı Grafikleri



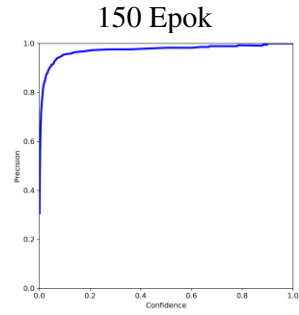
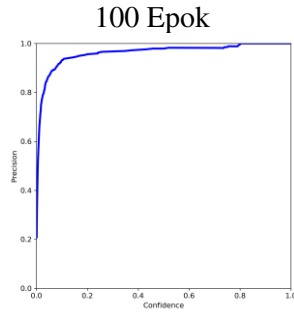
EK 3. YOLOv5 Eğitimleri Sonucunda Elde Edilen Başarı Oranı Grafikleri



EK 4. YOLOv7 Eğitimleri Sonucunda Elde Edilen Hassasiyet/Başarı Oranı, Duyarlılık/Başarı Oranı ve Başarı Oranı Grafikleri

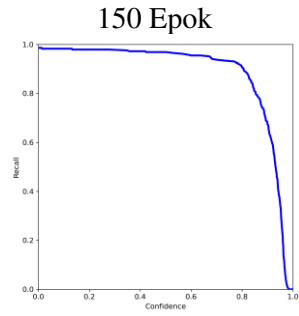
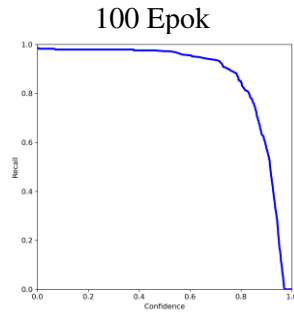
Hassasiyet/Başarı Oranı

YOLOv7



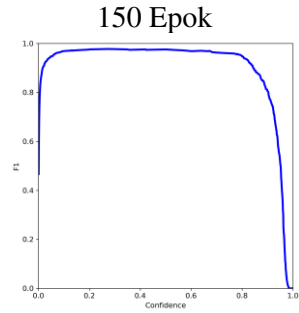
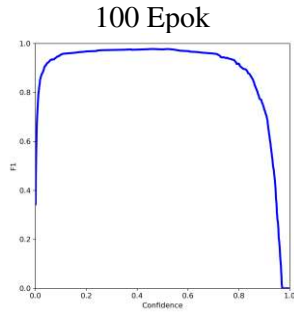
Duyarlılık/Başarı Oranı

YOLOv7

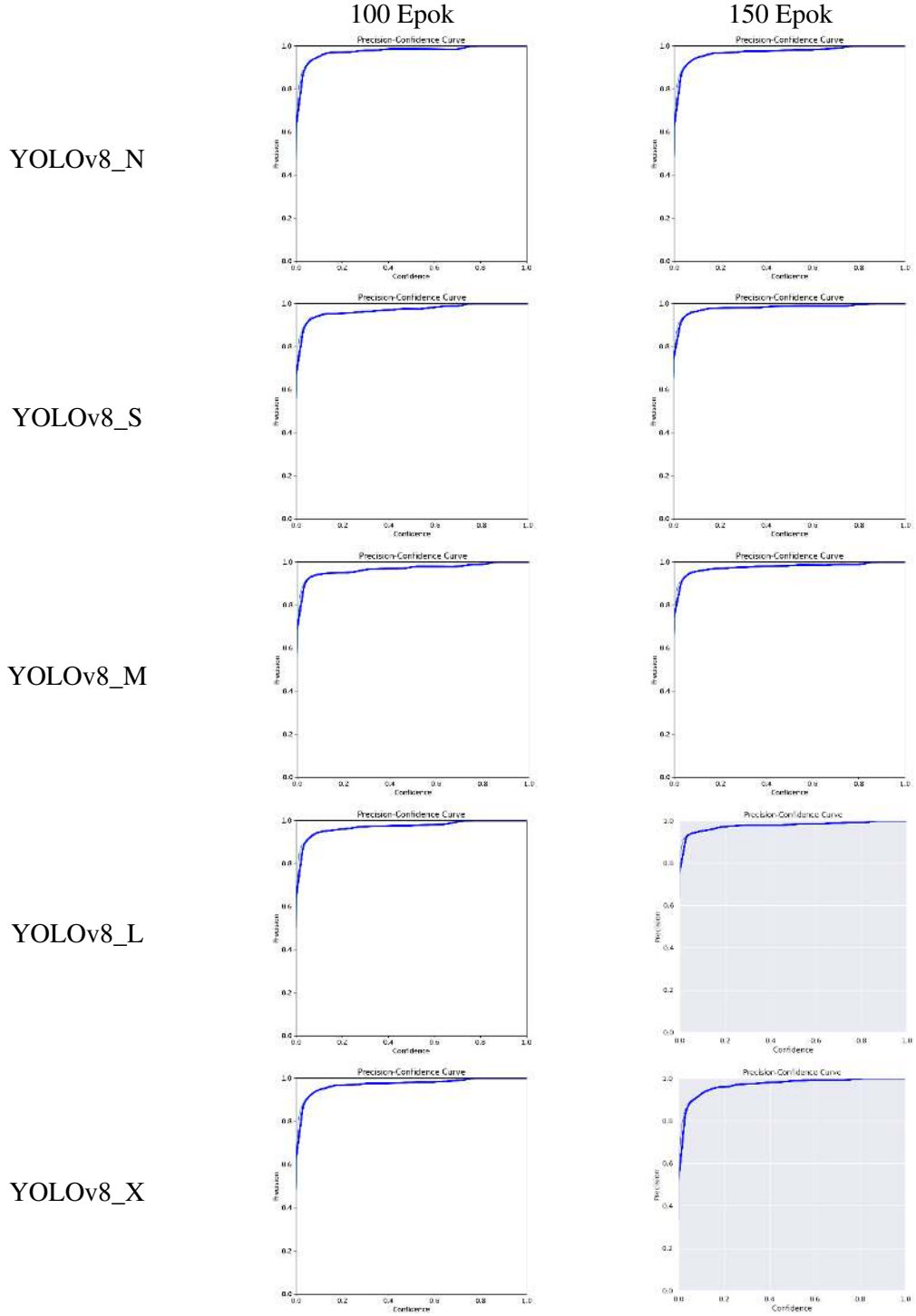


Başarı Oranı

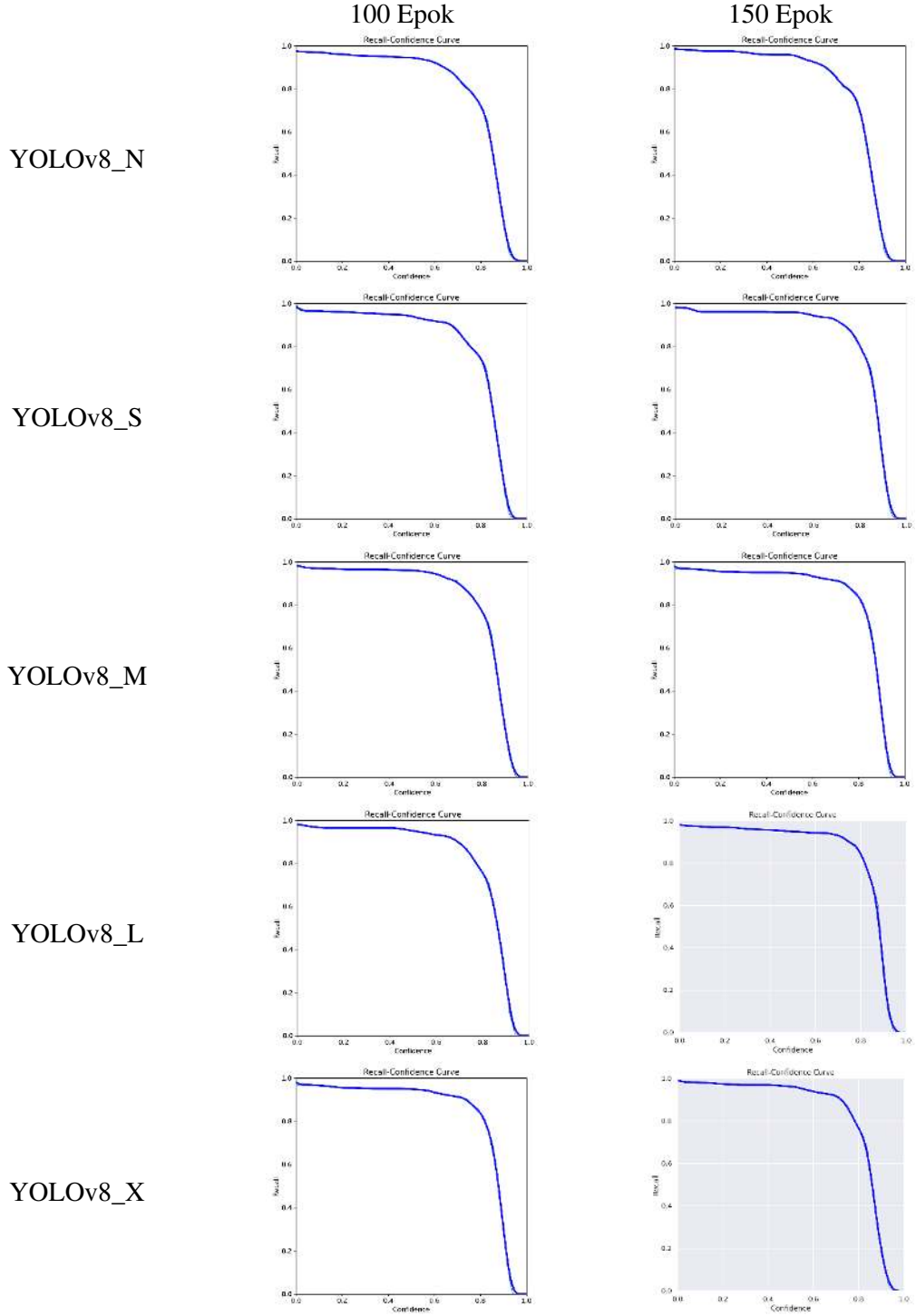
YOLOv7



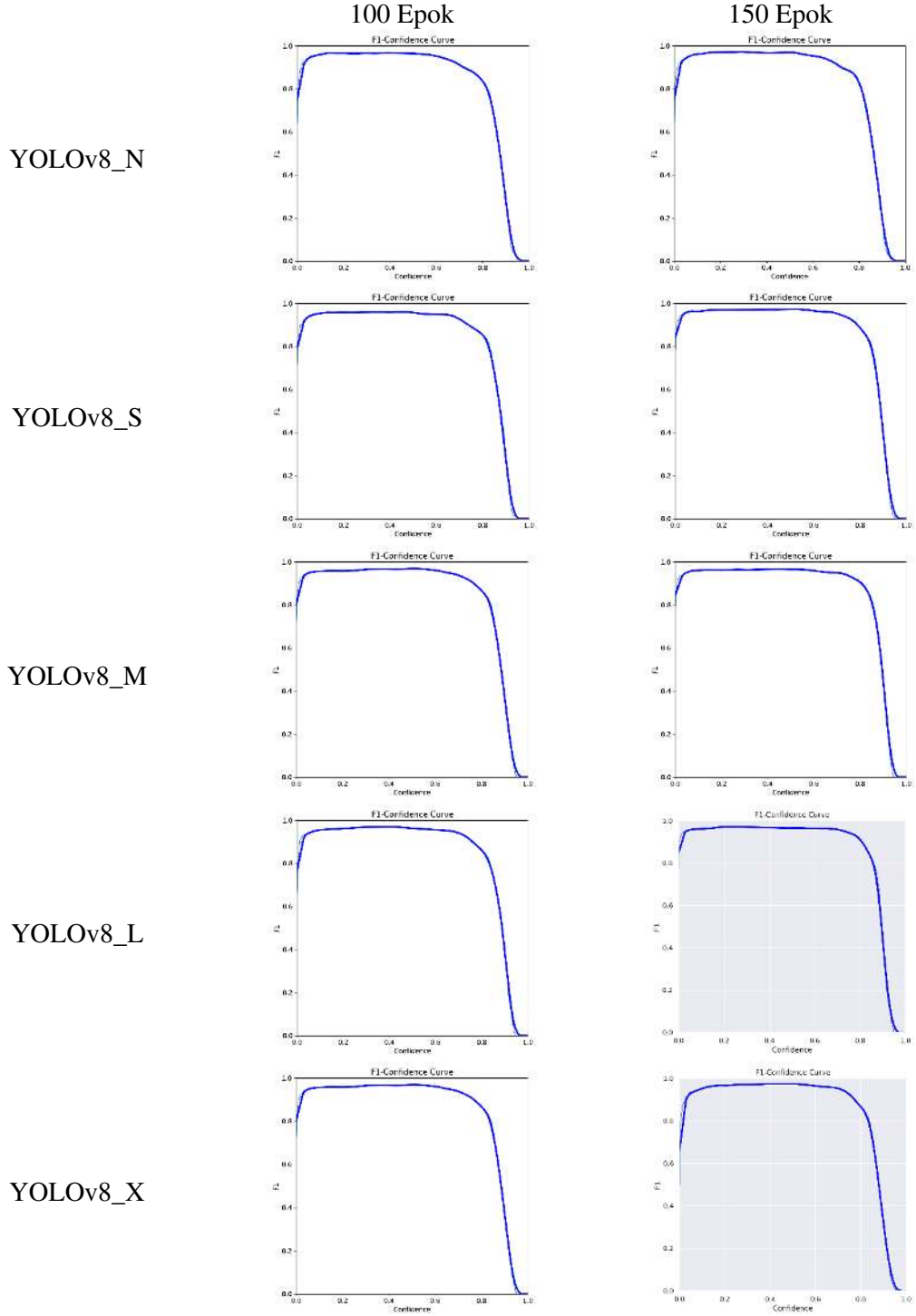
EK 5. YOLOv8 Eğitimleri Sonucunda Elde Edilen Hassasiyet/Başarı Oranı Grafikleri



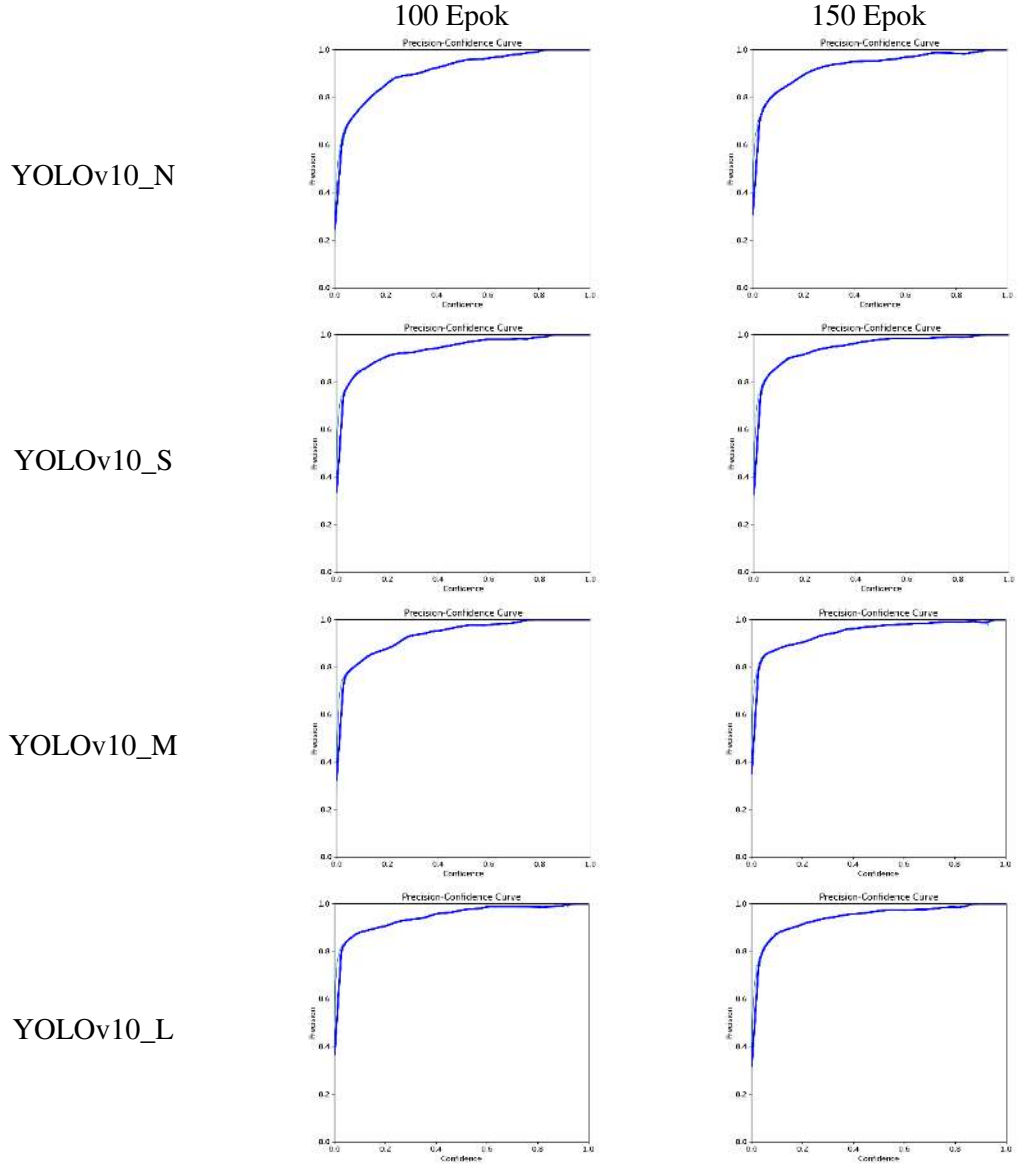
EK 6. YOLOv8 Eğitimleri Sonucunda Elde Edilen Duyarlılık/Başarı Oranı Grafikleri



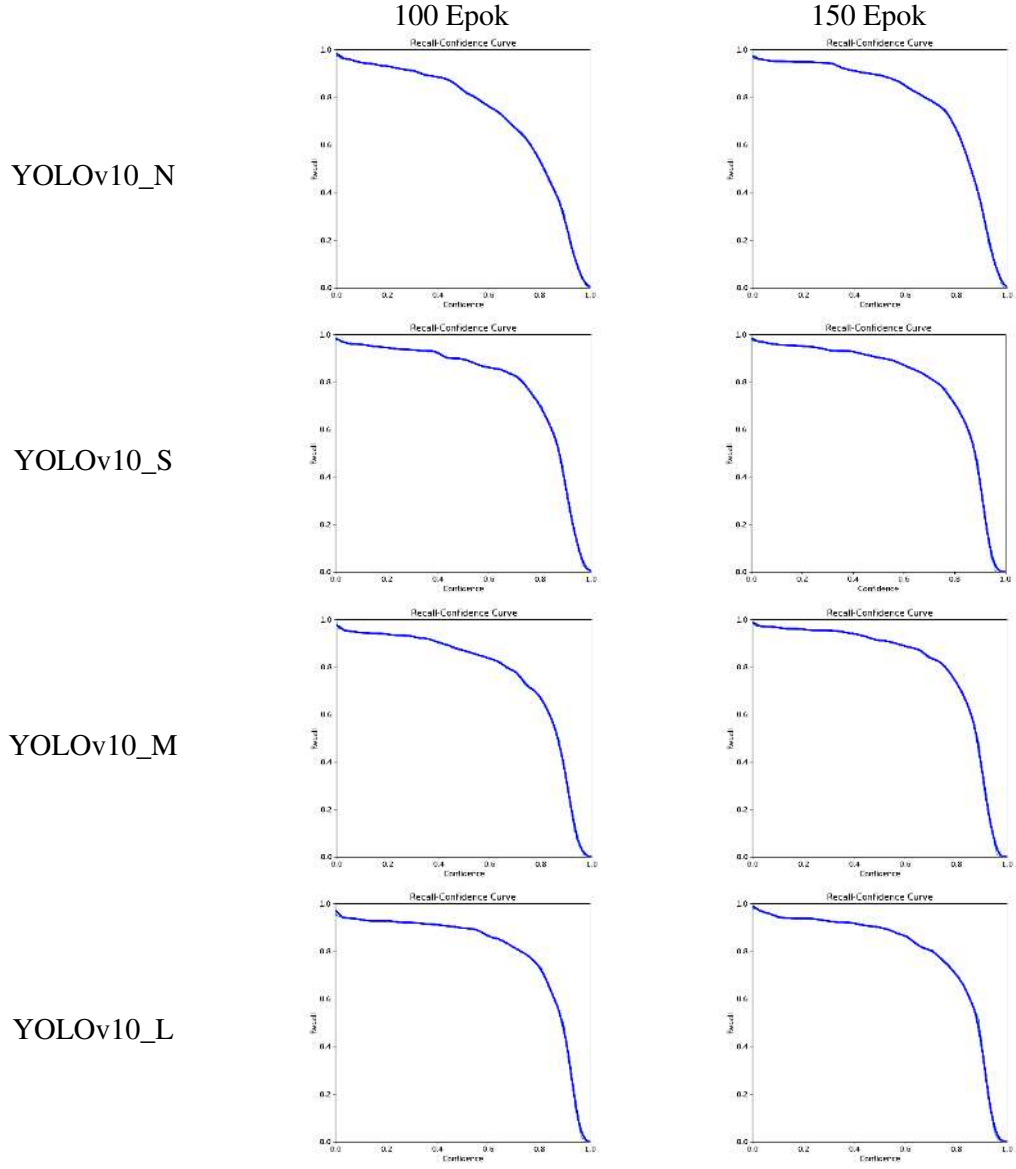
EK 7. YOLOv8 Eğitimleri Sonucunda Elde Edilen Başarı Oranı Grafikleri



EK 8. YOLOv10 Eğitimleri Sonucunda Elde Edilen Hassasiyet/Başarı Oranı Grafikleri



EK 9. YOLOv10 Eğitimleri Sonucunda Elde Edilen Duyarlılık/Başarı Oranı Grafikleri



EK 10. YOLOv10 Eğitimleri Sonucunda Elde Edilen Başarı Oranı Grafikleri

