

**T.C.
ERCIYES ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
MEKATRONİK MÜHENDİSLİĞİ ANABİLİM DALI**

**MEYVE BAHÇELERİNDE DERİN ÖĞRENME YÖNTEMİ
İLE REKOLTE TESPİTİ İÇİN UÇAN ROBOTİK SİSTEM
TASARIMI DENEYSEL UYGULAMASI**

**Hazırlayan
Burak ULU**

**Danışman
Prof. Dr. Şahin YILDIRIM**

Doktora Tezi

**Haziran 2023
KAYSERİ**

**T.C.
ERCIYES ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
MEKATRONİK MÜHENDİSLİĞİ ANABİLİM DALI**

**MEYVE BAHÇELERİNDE DERİN ÖĞRENME YÖNTEMİ
İLE REKOLTE TESPİTİ İÇİN UÇAN ROBOTİK SİSTEM
TASARIMI DENEYSEL UYGULAMASI
(Doktora Tezi)**

**Hazırlayan
Burak ULU**

**Danışman
Prof. Dr. Şahin YILDIRIM**

**Haziran 2023
KAYSERİ**

BİLİMSEL ETİĞE UYGUNLUK

Bu çalışmadaki tüm bilgilerin, akademik ve etik kurallara uygun bir şekilde elde edildiğini beyan ederim. Aynı zamanda bu kural ve davranışların gerektirdiği gibi, bu çalışmanın özünde olmayan tüm materyal ve sonuçları tam olarak aktardığımı ve referans gösterdiğimi belirtirim.

Burak ULU

İmza

“Meyve Bahelerinde Derin ğrenme Yöntemi İle Rekolte Tespiti İin Uan Robotik Sistem Tasarımı Deneysel Uygulaması” adlı Doktora tezi, Erciyes Üniversitesi Lisansüstü Tez Önerisi ve Tez Yazma Yönergesi’ ne uygun olarak hazırlanmıştır.

Hazırlayan

Burak ULU

Danışman

Prof. Dr. Şahin YILDIRIM

Mekatronik Mühendisliği A.B.D. Başkanı

Prof. Dr. Şahin YILDIRIM

TEŞEKKÜR

“Meyve Bahçelerinde Derin Öğrenme Yöntemi İle Rekolte Tespiti İçin Uçan Robotik Sistem Tasarımı Deneysel Uygulaması” adlı tez çalışmasının seçiminde, yürütülmesinde, sonuçlandırılmasında ve sonuçların değerlendirilmesinde destek ve yardımlarını esirgemeyen değerli hocam sayın Prof. Dr. Şahin YILDIRIM’a teşekkür ederim.

Tez çalışması boyunca gösterdiği anlayış, özveri ve verdiği tüm desteklerinden dolayı eşim Banu ULU’ya ve biricik oğluma, yine hayatım boyunca desteğini esirgemeyen ve her zaman yanımda olan annem ve babam başta olmak üzere tüm aileme, ayrıca saha çalışmalarında desteklerini ve önerilerini esirgemeyen Dr. Öğr. Üyesi Ömer Faruk ERGİN ve Arş. Gör. Mehmet Safa BİNGÖL’e teşekkür ederim.

Ayrıca bu tez çalışmasına maddi destek olmak amacı ile 2211-C Öncelikli Alanlar Bursu kapsamında tezi destekleyen Türkiye Bilimsel ve Teknik Araştırma Kurumu’na (TUBİTAK) teşekkür ederim.

Burak ULU

Haziran 2023, KAYSERİ

MEYVE BAHÇELERİNDE DERİN ÖĞRENME YÖNTEMİ İLE REKOLTE TESPİTİ İÇİN UÇAN ROBOTİK SİSTEM TASARIMI DENEYSEL UYGULAMASI

Burak ULU

Erciyes Üniversitesi, Fen Bilimleri Enstitüsü
Doktora Tezi, Haziran 2023
Danışman: Prof. Dr. Şahin YILDIRIM

ÖZET

Son yıllarda nüfus artışı ve iklim değişikliği gibi etkenler ile ortaya çıkan ve içinde bulunduğumuz yüzyılda insanlığın temel sorunları arasında gösterilen gıdaya erişim, tarımda verimliliğin artırılması, hassas ve doğru tarım uygulamalarına geçiş gibi konular oldukça önem arz etmektedir. Bu uygulamalı tez çalışmasında, tarımda verimliliği kontrol etmek için önemli bir konu olan rekoltenin belirlenmesini ve mahsulün durum analizinin gerçekleştirilmesini sağlayacak hava aracının tasarımı ve imalatı, yapay zeka modelinin oluşturulması ve deneysel olarak uygulanması amaçlanmıştır. Bu çalışma için tasarlanıp üretilen dört rotorlu hava aracı, kamera ve sensör sistemi ile donatılmasıyla birlikte derin öğrenme yöntemiyle elde edilen görüntüleri analiz etme özelliğine de sahiptir. Bu sebeple, çalışmada uçan robotik sistem (URS) olarak isimlendirilmiştir. Robotik sistemden elde edilen görüntüler üzerinde evrişimli sinir ağ (ESA) yöntemi kullanılarak iki farklı mimari yaklaşımı ile elmaların tespit edilmesi gerçekleştirilmiştir. Farklı yaklaşımlar ile derin öğrenme modellerinin performansları kıyaslanarak elmaların tespit edilme doğruluğunu artırmak için eğitimde kullanılan hiperparametrelerin optimize edilmesini sağlayan özgün bir yöntem uygulanmıştır. Ayrıca çalışmanın temel amacı olan rekoltenin belirlenmesi için de elmaların sayılma performansını artıran yeni bir yaklaşım önerilmiştir. Sonuçlar son derece umut verici olup uygulanan tüm yöntemler performans açısından karşılaştırmalı olarak tez içerisinde sunulmuştur.

Anahtar Kelimeler: Mini İnsansız Hava Aracı, Derin Öğrenme, Robotik, Tarımda Teknoloji.

EXPERIMENTAL APPLICATION OF FLYING ROBOTIC SYSTEM DESIGN FOR DETECTION OF YIELD WITH DEEP LEARNING METHOD IN ORCHARDS

Burak ULU

Erciyes University, Graduate School of Natural and Applied Sciences

PhD Thesis, June 2023

Supervisor: Prof. Dr. Şahin YILDIRIM

ABSTRACT

Nowadays, issues such as access to food, increasing productivity in agriculture, and transition to sensitive and correct agricultural practices, which have emerged with factors such as population growth and climate change, and which are among the main problems of humanity in the current century, and are of great importance. This applied thesis aims to design and manufacture the micro aerial vehicle (MAV), which will enable the determination of the yield, which is an important issue to control productivity in agriculture, and the analysis of the crop situation, the creation of an artificial intelligence model and its experimental application. The quadrotor MAV designed and produced for this study, equipped with a camera and sensor system, also could analyze images obtained by deep learning method. For this reason, it is named the flying robotic system (FRS) in the study. Apples are detected with two different architectural approaches by using the convolutional neural network (CNN) method on the images obtained from the robotic system. By comparing the performances of deep learning models with different approaches, a unique method was applied to optimize the hyperparameters used in training in order to increase apple detection accuracy, a new approach has been proposed to increase the counting performance of apples to determine the yield, which is the main purpose of the study. The results are very promising and all the applied methods are presented in the thesis in terms of performance.

Keywords: Micro Aerial Vehicle (MAV), Deep Learning, Robotic, Agricultural Technologies.

İÇİNDEKİLER

MEYVE BAHÇELERİNDE DERİN ÖĞRENME YÖNTEMİ İLE REKOLTE TESPİTİ İÇİN UÇAN ROBOTİK SİSTEM TASARIMI DENEYSEL UYGULAMASI

BİLİMSEL ETİĞE UYGUNLUK	ii
YÖNERGEYE UYGUNLUK.....	iii
KABUL VE ONAY	iv
TEŞEKKÜR.....	v
ÖZET.....	vi
ABSTRACT.....	vii
İÇİNDEKİLER	viii
KISALTMALAR	xi
TABLolar LİSTESİ.....	xvi
ŞEKİLLER LİSTESİ	xviii
GİRİŞ	1

1. BÖLÜM

LİTERATÜR ÇALIŞMASI

1.1. Giriş	3
1.2. Yapılan Çalışmalar	3

2. BÖLÜM

DENEYSEL SİSTEM

2.1. İnsansız Hava Araçları	8
2.1.1. Sabit Kanatlı İnsansız Hava Araçları	9
2.1.2. Döner Kanatlı İnsansız Hava Araçları.....	10
2.1.3. Dikey İniş-Kalkış Yapabilen İnsansız Hava Araçları (DİHA).....	11
2.2. Deneysel Çalışmalar için Tasarlanan Sistem.....	12
2.2.1. Dört Rotorlu Uçan Robotik Sistemin Tasarımı.....	12
2.2.1.1. Dört Rotorlu Hava Araçlarının Kinematiki	19
2.2.1.2. Dört Rotorlu Hava Araçlarının Dinamiki	21
2.2.1.3. Dört Rotorlu Hava Aracının Hareket Denklemleri	21

2.2.2. Dört Rotorlu Uçan Robotik Sistemin (URS) Temel Elemanlarının İmalatı.....	22
2.2.2.1. Karbon Fiber URS Çerçeve İmalatı.....	23
2.2.2.2. Bağlantı Elemanlarının Üç Boyutlu Yazıcıda Üretim Safhaları	24
2.2.2.3. Uçan Robotik Sistemin Tasarım ve Üretimi.....	25
2.2.3. Sistemin Genel Donanımsal Yapısı.....	26
2.2.4. Sistemin Yazılımlarının Geliştirilmesi Süreci	33
2.2.4.1. Uçuş Planlama ve Simülasyon Programı	34
2.2.4.1. Python Programlama Dili ve TensorFlow Kütüphanesi	35

3. BÖLÜM

YÖNTEMLER

3.1. Çalışma Ortamının Tanıtılması.....	36
3.2. Görüntü İşleme ve Bilgisayarlı Görü Yöntemi.....	38
3.2.1. Opencv Kütüphanesi ve Görüntü İşleme Temelleri	40
3.2.2. Nesne Tanıma	42
3.2.3. Derin Öğrenme Tabanlı Nesne Tanıma	44
3.2.4. Evrişimsel Sinir Ağları	44
3.3. Elmaların Sayılması İşlem Adımları	51
3.3.1. Veri Setini Belirleme Metodu.....	51
3.3.2. Evrişimli Sinir Ağ (ESA) Modeli Eğitimi	53
3.3.3. Nesne Tespiti.....	56
3.3.4. Nesne Takibi ve Sayımı Metodu	57

4. BÖLÜM

BULGULAR

4.1. Simülasyon Çalışmaları.....	60
4.1.1. Otonom Uçuş Yörünge Takibi için Simülasyon Çalışmaları	60
4.1.2. Kontrol Parametrelerinin Optimizasyonu için Simülasyon Çalışmaları....	61
4.1.3. Konumlandırma ve Haritalama Simülasyon Çalışmaları.....	72
4.2. Deneysel Çalışmalar.....	73
4.2.1. Otonom Uçuş ile Yörünge Takibi için Deneysel Çalışmalar	73
4.2.2. Kontrol Parametrelerinin Optimizasyonu İçin Deneysel Çalışmalar	75
4.2.3. Konumlandırma ve Haritalama Deneysel Çalışmaları	75
4.2.4. Derin Öğrenme ile Elma Tespiti Çalışmaları	76

4.2.4.1. MS COCO Veri Seti ile Eğitilmiş Hazır SM SA Modeli ile Elma Tespiti.....	77
4.2.4.2. MS COCO Veri Seti ile Eğitilmiş Hazır FRC SA Modeli ile Elma Tespiti.....	79
4.2.4.3. Özgün Veri Seti ile Eğitilen SM SA Modeli ile Elma Tespiti	80
4.2.4.4. Özgün Veri Seti ile Eğitilen FRC SA Modeli ile Elma Tespiti.....	83
4.2.4.5. Hiperparametreleri Optimize Edilen SM SA Modeli ile Elma Tespiti .	86
4.2.4.6. Hiperparametreleri Optimize Edilen FRC SA Modeli ile Elma Tespiti.....	92
4.2.5. Elma Sayma Çalışmaları	97
4.2.6. Hastalıklı Elmaların Tespiti.....	101

5. BÖLÜM

SONUÇLAR VE ÖNERİLER

KAYNAKÇA	106
ÖZGEÇMİŞ.....	114

KISALTMALAR

(t)	: Hareketli görüntüdeki (t) anını ifade eden zaman değişkeni
$(t-1)$	: Hareketli görüntüdeki $(t-1)$ anını ifade eden zaman değişkeni
$C_{şarj}$	: Deşarj katsayısı
$C_{batarya}$	: Pilin birim zamandaki taşınan elektrik yükü
$D_ş$	: Motorun şaft çapı
D_m	: Motorun çapı
F_D	: Sürüklenme kuvvetleri
F_g	: Yerçekimi kuvvetleri
G_m	: Motorun ağırlığı
I_{max}	: Motorun çektiği maksimum sürekli akım
K_V	: Motor hız sabiti
K_t	: Motor tork sabiti
$L_ş$	: Motorun şaft uzunluğu
L_m	: Motorun uzunluğu
N_{tespit}	: Derin öğrenme modeliyle tespit edilen elma sayısı
N_{toplam}	: Hesaplanan toplam elma sayısı
P_b	: Belirtilen noktanın boylam derecesi
P_e	: Belirtilen noktanın enlem derecesi
P_m	: Motorun tükettiği maksimum sürekli güç
R_{AB}	: Hava aracının B konumunun A konumuna göre dönüşüm matrisi
R_x	: X ekseninde dönme matrisi
R_y	: Y ekseninde dönme matrisi
R_z	: Z ekseninde dönme matrisi
T_i	: i. motorun ürettiği itme kuvveti
T_i	: i. motorun ürettiği itme kuvveti
T_{zB}	: B noktasında z eksenindeki toplam itme kuvveti
V_{range}	: Voltaj aralığı
i_{motor}	: Uçuş sırasında motorun çektiği miliamper cinsinden ortalama akım değeri

k_d	: Sürüklenme katsayısı
$k_{sınıf}^i$	: i. ağacın sınıfına göre belirlenen katsayı değeri
$r_{pervane}$	: Pervane yarıçapı
$t_{uçuş}$	: İHA'nın uçuş süresi
x_1	: Nesnenin (t-1) anındaki x eksenindeki konumu
x_2	: Nesnenin (t) anındaki x eksenindeki konumu
y_1	: Nesnenin (t-1) anındaki y eksenindeki konumu
y_2	: Nesnenin (t) anındaki y eksenindeki konumu
α	: Öğrenme oranı
Ah	: Pilin amper saat cinsinden toplam kapasitesi
API	: Application Programming Interface, uygulama programlama arayüzü
b	: Bias değeri
b	: İtme katsayısı
BSD	: Berkeley Software Distribution, , Berkeley Üniversitesi'nin kaynak kodu dağıtımını olan, AT&T'nin UNIX'i için bir eklentiler zinciridir.
calib3d	: Kamera kalibrasyonu ve üç boyutlu yeniden boyutlandırma
CNC	: Computer Numerical Control (Bilgisayarlı Sayısal Kontrol)
CNN	: Convolutional Neural Network
core	: Çekirdek işlev modülü
CSV	: Comma Separated Values, virgülle ayrılmış değerler
d	: Nesnenin t ve (t-1) zamanındaki ağırlık merkezlerinin birbirlerine öklid mesafesi
DiHA	: Dikey kalkış yapabilen İnsansız Hava Aracı
DVM	: Nesneleri tek bir görüntüde sınıflandırmak için kullanılan sınıfa özgü doğrusal Destek Vektör Makinaları
dwg	: Drawing (Çizim), iki ve üç boyutlu tasarım verilerini depolamak için kullanılan ikili dosya biçimi
e(t)	: Hata fonksiyonu
EHK	: Elektronik Hız Kontrolcü
EKVH	: Eşanlı Konumlandırma ve Haritalama, hiç bilinmeyen bir ortamda lokasyonun ve ortamın tanınması için kullanılan algoritmadır
Epoch	: Baştan sona tüm verilerin ağ tarafından bir kere işlenmesine 1 epoch denir

ESA	: Evrişimsel Sinir Ağı
F	: Hava aracının kütle merkezine etki eden toplam kuvvet
F	: Motorun sürücü frekansı
F1 Skoru	: Derin öğrenmede eğitim başarısını gösteren bir başarı metriği
Faster RCNN	: ESA tabanlı nesne tespiti için kullanılan algoritma
FCN	: Fully Convolutional Network (Tam Evrişim Ağı)
feature2d	: İki boyutlu özellikler çerçevesi modülü
FFBPN	: Feed-Forward Backpropagation Network, ileri beslemeli geri yayılım sinir ağı
FPN	: Feature Pyramid Network, ESA algoritması
FPS	: Frame Per Seconds, videolarda saniyede geçen kare sayısını ifade eder
FRC	: Faster-RCNN, kullanılan ESA yönteminin kısaltması
g	: Yerçekimi ivmesi
Gazebo	: ROS üç boyutlu görselleştirme platformu
GHz	: Gigahertz, bir işlemcinin saat veri iletim hızını ifade eder
GPU	: Graphics Processing Unit, grafik işlemci ünitesi
GUI	: Graphical User Interface, grafiksel kullanıcı arayüzü
İHA	: İnsansız Hava Aracı
imgproc	: Görüntü işleme modülü
$J(\theta)$	: Kayıp fonksiyonu
Kd	: Türev kazanç parametresi
Ki	: İntegral kazanç parametresi
Kp	: Oransal kazanç parametresi
Lidar	: Lazer imaging detection and ranging, lazer darbeleriyle uzaklığı algılamaya yarayan teknoloji
logsig	: Aktivasyon fonksiyonu
m	: Momentum
MavLink	: Uçuş kontrolcü ile haberleşmede kullanılan protokol
MS COCO	: Microsoft Common Object in Context, 120.000 görüntüden oluşan ve literatürde sıkça kullanılan bir veri kümesidir
MSE	: Mean Square Error
OpenCV	: Gerçek zamanlı bilgisayar görüşü uygulamalarında kullanılan açık kaynaklı kütüphane

OverFeat	: Evrişimsel sinir ağ mimarisi
PID	: Proportional Integral Derivative, Oransal İntegral Türevsel denetleyici
pureIn	: Aktivasyon fonksiyonu
PWM	: Pulse Width Modulation, darbe genişlik modülasyonu
RAM	: Random Accesss Memory, rastgele erişimli hafıza
RBNN	: Radial Basis Neural Network, radyal esaslı sinir ağı
RCNN	: Region Convolutional Neural Network, ESA algoritması
ReLU	: Rectified Linear Unit, doğrultulmuş linear birim (aktivasyon fonksiyonu)
ROS	: Robot Operating System, robotlar için geliştirilen işletim sistemi
RPN	: Regional Proposal Networks, bölge öneri ağı
RVIZ	: ROS Visualization
SA	: Sinir Ağı
SM	: SSD-MobileNet, kullanılan ESA yönteminin kısaltması
SSD	: Single Shot multibox Detection, ESA algoritması
SSD-MobileNet	: Single Shot multibox Detection
SY	: Saat yönü
SYT	: Saat yönünün tersi
tansig	: Aktivasyon fonksiyonu
TF-Record	: Tensorflow'un kendi ikili depolama formatı
TÜBİTAK	: Türkiye Bilimsel ve Teknolojik Araştırma Kurumu
UI	: User Interface, kullanıcı arayüzü
URS	: Uçan Robotik Sistem
v	: Hücre sayısı
videoio	: Video I/O modülü
W	: Ağırlık parametresi
x	: Bağımsız girdi değişkeni
XML	: Extensible Markup Language, genişletilebilir işaretleme dili
y	: Çıkış değeri
YKİ	: Yer Kontrol İstasyonu
YOLO	: You Only Look Once, ESA algoritması
YSA	: Yapay Sinir Ağları
z	: Girdiye ait skoru belirlemede kullanılan bağımlı değişken

η	: Epoch sayısı
τ	: Hava aracının kütle merkezine etki eden toplam tork
B	: Batch boyutu, parametre güncellemesinin gerçekleştiği ağı verilen alt örneklerin sayısıdır.
f	: Filtre boyutu
k	: Rekolte hesaplamada kullanılan düzeltme katsayısı
m	: Hava aracının kütlesi
θ	: Kayıp fonksiyonunda ağırlıklara karşılık gelen parametrelerdir
$\sigma(x)$	: Aktivasyon fonksiyonu
$\omega \times mV$	: Merkezci kuvvetleri tanımlar

TABLOLAR LİSTESİ

Tablo 2.1.	Sistemde bulunan temel bileşenlerin ağırlık değerleri.....	16
Tablo 2.2.	KDE Direct 2315XF-885 fırçasız doğru akım motoruna ait temel parametreler.	28
Tablo 3.1.	Temel nesne tanıma algoritmaları.....	42
Tablo 4.1.	Uçuş Modlarının kısa açıklamaları.	62
Tablo 4.2.	YSA modelleri için eğitim parametreleri.....	64
Tablo 4.3.	Simülasyonda otonom uçuş için kullanılan Kp, Ki ve Kd parametreleri. .	65
Tablo 4.4.	FFBPN sinir ağ modellerinin MSE hata değerleri.....	67
Tablo 4.5.	Farklı spread değerleriyle eğitilen RBNN modellerinin MSE hata değerleri.	68
Tablo 4.6.	FFBPN sinir ağ modellerinin MSE hata değerleri.....	69
Tablo 4.7.	Farklı spread değerleriyle eğitilen RBNN modellerinin MSE hata değerleri.	70
Tablo 4.8.	İstenilen yörünge için tahminlenen Kp, Ki ve Kd parametreleri.(a) Yuvarlanma kontrol modeli için, (b) yunuslama kontrol modeli için elde edilen değerler.....	71
Tablo 4.9.	MS COCO ile eğitilen SSD-Mobilenet sinir ağının eğitim parametreleri.	78
Tablo 4.10.	MS COCO veri setiyle eğitilen SSD-Mobilenet modelinin gerçekleştirilen elma tespit çalışmasına ait sonuçları.	78
Tablo 4.11.	MS COCO ile eğitilen FRC modelinde kullanılan eğitim parametreleri..	79
Tablo 4.12.	MS COCO veri setiyle eğitilen FRC modelinin gerçekleştirilen elma tespit çalışmasına ait sonuçları.	80
Tablo 4.13.	SSD-Mobilenet derin sinir ağının özgün veri setiyle eğitilmesinde kullanılan hiperparametre setleri.	80
Tablo 4.14.	Sinir ağ modellerinin elma tespit çalışmasına ait performans sonuçları. ..	83
Tablo 4.15.	Faster-RCNN derin sinir ağının özgün veri setiyle eğitilmesinde kullanılan hiperparametre setleri.	83
Tablo 4.16.	Sinir ağ modellerinin elma tespit çalışmasına ait performans sonuçları. ..	86
Tablo 4.17.	YSA modelinin eğitilmesi için kullanılan yirmi farklı SSD-Mobilenet modeline ait hiperparametreler.	88

Tablo 4.18. Hiperparametre tahminleyen FFBNP YSA modeli için eğitim parametreleri.	90
Tablo 4.19. FFBNP YSA modeli ile istenilen başarı için, girilen ve tahminlenen parametre değerleri.	91
Tablo 4.20. Optimizasyon edilen SSD-MobileNet derin sinir ağı modeli ile elma tespit çalışmasına ait performans sonuçları.	92
Tablo 4.21. YSA modelinin eğitilmesi için kullanılan yirmi farklı Faster R-CNN modeline ait hiperparametreler.	93
Tablo 4.22. Hiperparametre tahminleyen FFBNP YSA modeli için eğitim parametreleri.	95
Tablo 4.23. İstenilen başarı değeri için eğitilen FFBNP YSA modelinde girilen ve tahminlenen parametre değerleri.	96
Tablo 4.24. Optimizasyon edilen Faster-RCNN derin sinir ağı modeli ile elma tespit çalışmasına ait performans sonuçları.	97
Tablo 4.25. Uygulanan metotların rekolte belirleme başarılarının kıyaslamalı gösterimi.	100

ŞEKİLLER LİSTESİ

Şekil 2.1.	Kanat yapılarına göre farklı tipte İnsansız Hava Araçlarının yapısı.....	9
Şekil 2.2.	Sabit kanatlı İHA yapısının görünümü	10
Şekil 2.3.	Döner Kanatlı İHA yapısının görünümü	11
Şekil 2.4.	Dikey iniş-kalkış yapabilen sabit kanatlı İHA yapısının görünümü.....	11
Şekil 2.5.	"X" konfigürasyonuna sahip bir dört rotorlu İHA'nın şematik görünümü.	12
Şekil 2.6.	URS'de kullanılan (a) saat yönünün tersine (SYT) ve (b) saat yönünde (SY) dönmesi için tasarlanan pervaneler.	13
Şekil 2.7.	(a) Gövde imalatında kullanılan karbon fiber plaka ve (b) motorları taşıyan kolların imalatında kullanılan karbon fiber kare profilin görünümü.	13
Şekil 2.8.	URS parçalarının üç boyutlu katı model çizimlerinin görünümü.....	14
Şekil 2.9.	URS montajının üç boyutlu tasarım görüntüsü.....	15
Şekil 2.10.	URS'nin boyutsal bir şekilde yapısının görüntüsü.	15
Şekil 2.11.	Dört rotorlu bir İHA'nın temel kuvvet ve moment parametrelerinin gösterimi.	16
Şekil 2.12.	Tasarlanan URS'nin mekanik yapısı için uygulanan gerilim analizinin görüntüsü.	18
Şekil 2.13.	Tasarlanan URS'nin mekanik yapısı için gerçekleştirilen toplam deformasyon analizinin görüntüsü.....	19
Şekil 2.14.	Dört rotorlu bir İHA'nın hareket eksenlerinin gösterimi.....	19
Şekil 2.15.	Dört rotora sahip İHA'nın hareket eksenlerinin şematik gösterimi.....	20
Şekil 2.16.	Kesilecek parçaların CNC tezgah için oluşturulan dwg formatı.	23
Şekil 2.17.	Üretilen karbon fiber parçalar.....	24
Şekil 2.18.	Kare karbon fiber profilden üretilen kol yapısının URS'ye montajı.	24
Şekil 2.19.	Tasarlanıp 3 boyutlu yazıcıda üretilen aparatlar.....	25
Şekil 2.20.	URS ve Temel Elemanları.	26
Şekil 2.21.	Dört rotorlu uçan robotik sistemin donanımsal şeması.	27
Şekil 2.22.	URS sistemde kullanılan fırçasız doğru akım motoru.	27
Şekil 2.23.	Deneyisel sistemde kullanılan EHK'nın bağlantı şekli.	28
Şekil 2.24.	Pixhawk Uçuş kontrolcüsü ve bağlantılarının gösterimi.	29

Şekil 2.25.	URS için tercih edilen lityum polimer batarya.	30
Şekil 2.26.	Hokuyo URG-04LX lidar sensör.	31
Şekil 2.27.	EKVH metoduyla oluşturulan elma bahçesinde bir koridorun haritası	31
Şekil 2.28.	Aksiyon kamerası.	32
Şekil 2.29.	Odroid XU4 mini bilgisayar.	32
Şekil 2.30.	Uçuş kontrolcü ile harici kontrolcü haberleşmesinin şematik gösterimi ...	33
Şekil 2.31.	Mission Planner uçuş planlama arayüzü görüntüsü.	34
Şekil 2.32.	Dronekit uçuş izleme ekranı.	35
Şekil 3.1.	Yeşilhisar'da bulunan elma bahçesinin URS ile havadan görünümü.	37
Şekil 3.2.	Bahçe içerisinde kullanılan koridorun gösterimi.	37
Şekil 3.3.	Görüntü işleme akış şeması.	38
Şekil 3.4.	Görüntü işleme teknikleri ile işlenmiş dijital görüntüler.	39
Şekil 3.5.	Geleneksel nesne tanıma adımları.	43
Şekil 3.6.	Evrişim işlemi.	45
Şekil 3.7.	Adım değeri.	46
Şekil 3.8.	Dolgulama işlemi.	46
Şekil 3.9.	Havuzlama katmanı.	47
Şekil 3.10.	Tam bağlantı katmanı.	48
Şekil 3.11.	CNN modeli.	49
Şekil 3.12.	Derin öğrenme yöntemiyle nesne sayma genel metodolojisi.	51
Şekil 3.13.	Veri seti için çekilen kırmızı elma görüntüsü örnekleri.	52
Şekil 3.14.	Etiketleme işlemi için kullanılan LabelImg programının arayüzü.	52
Şekil 3.15.	COCO veri seti (soldaki) ve özgün veri setiyle (sağdaki) eğitilen sinir ağ model ile gerçekleştirilen elma tespiti.	54
Şekil 3.16.	Basit yapılı örnek bir sinir ağ modeli.	54
Şekil 3.17.	Nesne tespiti için kullanılan evrişimli sinir ağ yapısı.	57
Şekil 3.18.	SSD (soldaki) ve F-RCNN (sağdaki) eğitilen sinir ağ model ile gerçekleştirilen elma tespiti.	57
Şekil 3.19.	Uygulanan nesne takibi yönteminin şematik gösterimi.	58
Şekil 3.20.	Deney sahasında elma sayma uygulamasının şematik gösterimi.	59
Şekil 4.1.	Simülasyon ekranında URS'nin otonom yörünge takibi.	61
Şekil 4.2.	URS'nin otonom uçuşunu üç boyutlu olarak görselleştiren Gazebo Programı.	61

Şekil 4.3.	(a) Yuvarlanma hareketi için, (b) yunuslama hareketi için, (c) sapma hareketi için gösterilen genel kontrol şeması.....	63
Şekil 4.4.	Kp, Ki ve Kd parametrelerinin belirlenmesi için eğitilen YSA modelleri: (a) İleri Beslemeli Geri Yayılım Sinir Ağı (FFBPN), (b) Radyal Esaslı Sinir Ağı (RBNN).	64
Şekil 4.5.	Yuvarlanma (a) ve Yunuslama (b) kontrol modellerindeki Kp, Ki ve Kd parametre değişimleri ile yörünge takibinin gösterimi.	66
Şekil 4.6.	En başarılı FFBPN modeline ait performans grafiği.	67
Şekil 4.7.	En başarılı FFBPN modeline ait regresyon grafikleri.	67
Şekil 4.8.	Spread= 0,001 için RBNN eğitim performansı grafiği.....	68
Şekil 4.9.	En başarılı FFBPN modeline ait performans grafiği.	69
Şekil 4.10.	En başarılı FFBPN modeline ait regresyon grafikleri.	69
Şekil 4.11.	Spread= 0,001 için RBNN eğitim performansı grafiği.....	70
Şekil 4.12.	Tahminlenen Kp, Ki ve Kd parametreleri ile simülasyonda gerçekleştirilen otonom uçuşun, referans yörüngeyle karşılaştırılması.	71
Şekil 4.13.	Gazebo simülasyon ortamında EKVH uygulamasının gösterimi.	72
Şekil 4.14.	Deneyisel çalışma için belirlenen noktaların harita üzerinde gösterilmesi.	73
Şekil 4.15.	Hedef noktaları tanımlayan program satırları.	74
Şekil 4.16.	Otonom uçuşta izlenen yörüngeyi uydu haritası üzerinde görünümü.	74
Şekil 4.17.	Kontrol parametrelerinin optimizasyonunun deneyisel yörünge kontrolü üzerine etkisinin gösterimi.....	75
Şekil 4.18.	(a) Uydu sinyalleri ile, (b) EKVH tekniği ile oluşturulan yörüngelerin gösterimi.	76
Şekil 4.19.	MS COCO veri seti kullanılarak eğitilen SSD-MobileNet derin sinir ağı modelinin deneyisel elma tespiti çalışmasına ait görüntüleri.	78
Şekil 4.20.	MS COCO veri seti kullanılarak eğitilen FRC modeline ait deneyisel elma tespiti çalışmasına ait görüntüler.	79
Şekil 4.21.	(a) Öğrenme Oranı-İterasyon grafiği, (b) Kayıp-İterasyon grafiği, (c) Saniyede İşlenen Adım Sayısı-Toplam İterasyon grafiği.	81
Şekil 4.22.	Özgün veri setiyle eğitilen SSD-MobileNet sinir ağı modelinin deneyisel elma tespiti çalışmasına ait görüntüleri. (a) model1 ile, (b) model2 ile ve (c) model3.	82

Şekil 4.23.	(a) Öğrenme Oranı-İterasyon grafiği, (b) Kayıp-İterasyon grafiği, (c) Saniyede İşlenen Adım Sayısı-Toplam İterasyon grafiği.	84
Şekil 4.24.	Özgün veri setiyle eğitilen Faster- RCNN sinir ağ modelinin deneysel elma tespiti çalışmasına ait görüntüleri. (a) model1 ile, (b) model2 ile ve (c) model3.	85
Şekil 4.25.	Kayıp fonksiyonu-ağırlık ilişkisinin grafiksel gösterimi.	87
Şekil 4.26.	(a) Öğrenme Oranı-İterasyon grafiği, (b) Kayıp-İterasyon grafiği.	89
Şekil 4.27.	Hiperparametre tahminlemek amacıyla oluşturulan FFBPN YSA modelinin şematik gösterimi.	89
Şekil 4.28.	YSA modelin eğitim sonucundaki; (a) performans grafiği, (b) regresyon grafiği.	90
Şekil 4.29.	Optimize edilen hiperparametre değerlerine göre eğitilen SSD-MobileNet sinir ağ modelinin (a) Öğrenme Oranı-İterasyon grafiği, (b) Kayıp-İterasyon grafiği.	91
Şekil 4.30.	(a) Özgün veri setiyle eğitilen model ile, (b) hiperparametreleri optimize edilerek özgün veri setiyle eğitilen modelin, elma tespit performansları kıyaslanması.	92
Şekil 4.31.	(a) Öğrenme Oranı-İterasyon grafiği, (b) Kayıp-İterasyon grafiği.	94
Şekil 4.32.	Hiperparametre tahminlemek amacıyla oluşturulan FFBPN YSA modelinin şematik gösterimi.	94
Şekil 4.33.	FFBPN YSA modelinin eğitimi sonucundaki; (a) performans grafiği, (b) regresyon grafiği.	95
Şekil 4.34.	Optimize edilen öğrenme oranı ile eğitilen FRC sinir ağ modelinin; (a) Öğrenme Oranı-İterasyon grafiği, (b) Kayıp-İterasyon grafiği.	96
Şekil 4.35.	(a) Özgün veri setiyle eğitilen model ile, (b) hiperparametreleri optimize edilerek özgün veri setiyle eğitilen modelin, elma tespit performansları kıyaslanması.	97
Şekil 4.36.	Tespit edilen elmaların geliştirilen yazılım ile sayılmasının gösterimi.	98
Şekil 4.37.	Ağaçların sınıflandırılmasını esas alan önerilen yaklaşımın temel hesaplama mantığının gösterimi.	100
Şekil 4.38.	Geliştirilen nesne tespit sistemi ile yapısal bozukluğu olan elmaların tespit edilmesine yönelik gerçekleştirilen uygulamalara ait görüntüler. .	101

GİRİŞ

İnsan nüfusunun özellikle son yüzyıl içerisinde %290 [1] gibi çok yüksek oranda büyümesi dünyada kaynakların kullanımı noktasında sorunları da beraberinde getirmektedir. Bu sorunların en önemlilerinden bir tanesinin de tarımsal kaynakların doğru yönetilmesi olduğu şüphesizdir. Özellikle, günümüzde dünya çapındaki birlikler, ülkeler ve toplumsal kuruluşlar doğru tarım uygulamaları, sürdürülebilir tarım ve tarımsal kaynakların doğru kullanılmasına yönelik önemli kararlar almakta ve bu alandaki çalışmalara ciddi miktarda fonlar aktarmaktadırlar. Örneğin, Avrupa Komisyonu, Avrupa 2020 stratejisine dahil ettiği üç öncelikten biri olan ‘Sürdürülebilir büyüme: kaynakları daha verimli kullanan, daha yeşil ve daha rekabetçi bir ekonomiyi teşvik etmek [2] maddesiyle tarımın çoklu fayda sağladığını kabul etmektedir. Ülkemizde de 11. Kalkınma Planları [3] ve TÜBİTAK (Türkiye Bilimsel ve Teknolojik Araştırma Kurumu) öncelikli alanlar [4] kapsamında tarımsal verimlilik ve tarımda teknolojik gelişmelere yönelik önemli teşvikler açıklanmıştır. Tarımsal verimliliği artırmada önemli alt konulardan bir tanesi; mahsulün hastalık, verimlilik, rekolte bakımından doğru analiz edilmesidir. Ülkemizde bu tip incelemeler geleneksel olarak çoğunlukla çiftçinin gözlem ve tecrübelerine dayandırılarak yapılmaktadır. Dünya geneline bakıldığında ise çoğunlukla bu durum pek değişmemekle birlikte özellikle araştırma geliştirme faaliyetleri bakımından gelişmiş ülkelerde son dönemde birtakım teknolojilerin kullanılmaya başlandığı literatür çalışmalarında tespit edilmiştir. Özellikle bilgisayarlı görüş ile bahçelerin durum analizi, toprak verimlilik analizi gibi konular yaygın olarak araştırılmaktadır.

Bu deneysel tez çalışmasında; geliştirilen dört rotorlu insansız hava aracı ile belirlenen bir elma bahçesindeki dar koridorlarda otonom olarak yörünge takip etmesi, elmaların yapay zeka destekli görsel olarak incelenmesi amaçlanmaktadır (Geliştirilen hava aracından, yapay zeka destekli karar verebilen yapısı sebebiyle bundan sonraki bölümde uçan robotik sistem (URS) olarak bahsedilecektir). Bu çalışma ile büyük çaptaki, ticari

üretim gerçekleştirilen meyve bahçelerinde rekoltenin belirlenmesi ve hastalıklı meyvelerin erken tespiti otomatik olarak gerçekleştirilmiş olacaktır. Bu doğrultuda, dar bir alanda otonom olarak yörünge takibi gerçekleştirecek olan uçan robotik sistemin K_p , K_i ve K_d (K_p : Oransal (Proportional) kazanç parametresi), K_i : İntegral (Integral) kazanç parametresi) ve K_d : Türev (Derivative) kazanç parametresi) parametrelerinin doğru belirlenmesi ve bu doğrultuda uçuş planlamasının uygun yapılması beklenmektedir. Çalışmanın en kritik aşamalarından bir tanesi de üzeri ince bir tente ile kaplı elma bahçesinde, elmaları görsel olarak doğru tespit edebilecek bir derin öğrenme modelinin geliştirilmesi ve tespit edilen elmaların artırimsal olarak doğru sayılması şeklinde belirlenmiştir. Bu deneysel çalışma dört temel aşamada ele alınmıştır;

- Uçan Robotik Sistemin Tasarımı ve İmalatı
- Uçuş Yörüngesinin Planlanması
- Yapay Zeka Modelinin Geliştirilmesi ve Nesne Tespiti
- Nesneleri Doğru Sayabilecek Algoritmanın Oluşturulup Kodlanması

Tez çalışması 5 bölümden oluşmaktadır. 1. bölümde, literatürdeki çalışmalara yer verilmiş olup, tez çalışmasıyla ilgili konularda son yıllarda gerçekleştirilen bilimsel çalışmalara atıfta bulunulmuştur. 2. bölümde insansız hava araçlarıyla ilgili genel bilgilendirme yapılmış ve deneysel çalışma için özel tasarlanan ve imalatı gerçekleştirilen döner kanatlı robotik sistemin tanıtımı gerçekleştirilmiştir. Sistemin matematiksel modeli, hareket denklemleri, tasarım-imalat süreci, donanım-yazılım yapısından bahsedilmiştir. 3. bölümde uygulanan metotlardan bahsedilerek, meyvelerin tespiti ve sayılması için kullanılacak yapay zeka yöntemleri hakkında bilgi verilmiştir. Deneysel çalışma için uygulanan evrişimli sinir ağ yaklaşımları karşılaştırmalı olarak anlatılmış olup nesne tespiti ve kümülatif sayılmasına yönelik uygulanan yöntemler aktarılmıştır. Deneysel çalışmalar uygulanan yöntemlerin performansları 4. bölümde ele alınmıştır. Son olarak da, 5. bölümde sonuç ve öneriler değerlendirilmiştir.

1. BÖLÜM

LİTERATÜR ÇALIŞMASI

1.1. Giriş

Son yıllarda insansız hava araçları ile görsel denetim konusunda; askeri casusluktan [5], kargo taşımacılığına [6], yapı denetimlerinden [7], arama kurtarma operasyonlarına [8] kadar birçok alanda çalışmalar yapılmaktadır. İnsansız hava araçları (İHA), aktüatörler ile birlikte fiziksel bir model, operatörün araç durumunu izlemesini ve gerektiğinde aracı yönetmesini sağlayan yer kontrol istasyonu (YKİ), aviyonikler, yüksek teknolojik sensörler ve bunlar arasında iletişim kolaylığı sağlayan gelişmiş temel bileşenlerden oluşmaktadır. Özellikle mini İHA sınıfına giren döner kanatlı araçlar, sağladıkları hareket kabiliyeti ve gözetleme yetenekleri sayesinde robotik çalışmalarda çok yaygın olarak kullanılmaktadır. Tarım alanında gerçekleştirilen teknolojik uygulamalarda da bu tarz uçan robotik sistemlerin kullanımının yaygınlaştığı gözlemlenmektedir. Tarım alanlarının ilaçlanması, verimlilik analizi, arazi haritalama ve rekolte tespiti gibi konularda İHA'lar ile gerçekleştirilen çalışmalar önemli oranda birçok problemi çözmektedir. Otomasyona dayalı çözümler için son derece uygun olan düzenli endüstriyel meyve bahçelerinde, bilgisayarlı görü ve otonom sistemler ile verimi artırmaya dayalı çalışmalar yapmak mümkün hale gelmiştir. Özellikle dünyada önemli bir yeri olan ve ülkemizin de 3. büyük üreticisi olduğu elma meyvesi ile ilgili gerçekleştirilecek teknolojik alt yapı çalışmaları sürdürülebilir tarım politikalarına, ekonomiye ve gıdaya erişim kolaylığına önemli katkı sağlayacağı kaçınılmazdır.

1.2. Yapılan Çalışmalar

Tarımsal verimliliğin artırılması ve doğru tarım uygulamaları ile ilgili mahsulün incelenmesi ve durumunun analiz edilmesine dayalı yapılan araştırmalardan; özellikle bilgisayarlı görü konusunun alt alanı olan görsel denetim alanında ortaya konan

alıřmalar gze arpmaktadır [9-13]. Grnt iřleme yntemi kullanılarak gerekleřtirilen bu alıřmalar genellikle simlasyon zerinde yapılan uygulamalara dayalı olduėu ve saha alıřmalarında test edilmedikleri ya da bařarısız sonular verdiėi grlmektedir. nk, hareketli bir sistem zerinde deėiřken ıřık ve grnt kalitesi ile elde edilen grntler kullanılarak gerekleřtirilen grnt iřleme uygulamalarında sonuların doėruluėu pek de bařarılı bulunmamaktadır. Tabi ki bu alanda yapılan iyileřtirmeler, uygulanan farklı yntemler ile birtakım geliřmeler getirdiėi de ortadadır. Fakat bu uygulamalı tez alıřmadaki sonulardan da grleceėi zere bir derin ėrenme yapay zeka modeliyle tahminleme sonucu elde edilen ıktılar nesne tespiti ve sınıflandırılması konusunda nemli lde umut vadetmektedir. Literatrde de, yapay zeka destekli bilgisayarlı gr yntemiyle nesne tespiti ve sınıflandırılmasına ynelik bařarılı uygulamalar gze arpmaktadır [14-17].

Yapay zeka destekli bilgisayarlı gr konusunda sinir aėlarında gerekleřtirilen nemli geliřmeler, zellikle derin ėrenme metodunun geliřmesi, nesne tanıma ve hedef takibi konularında bařarılı sonular vermektedir. Katherine ve Karen [18], hava aracı ile elde ettikleri grntler zerinden bitki eřitlerini sınıflandırmaya ynelik derin ėrenme destekli bir bilgisayarlı gr alıřması ile istilacı trlerin belirlenmesini gerekleřtirmiřlerdir. Elde ettikleri kesinlik deėerleri dikkate alınarak umut verici bir alıřma gerekleřtirdikleri gzlenmiřtir.

Meyvelerle ilgili yapay zeka kullanılarak, aėlardaki elma ve portakal sayısının tahmin edildiėi Steven ve arkadaşlarının gerekleřtirmiř olduėu alıřmada [19], meyvelerin statik grntler zerinden sayılması iřlemi gerekleřtirilmiřtir. Nesneyi doėru tespit etmenin nemli bir problem olması gibi sayma iřleminin kmlatif olarak doėru gerekleřtirilmesi de son derece nemli bir problemidir. Tam Evriřim aėı (FCN: Fully Convolutional Network) tarafından tespit edilen her kk yuvarlak ktle iindeki meyve sayısını saymak iin ikinci bir sinir aėı ve doėrusal bir regresyon kullanarak sayma problemine yaklařılmıřtır. Bu yaklařımda sayma iřlemi de tahminleme yntemiyle belirlenmiř olup herhangi bir nesne takip veya sayma algoritması uygulanmamıřtır.

Arruda ve arkadaşları [14], yksek yoėunluklu grntlerde nesneleri saymak ve bulmak iin bir ESA yaklařımı nermiřlerdir. nerilen yntemin, aėlar ve arabaları ieren iki farklı veri seti kullanılarak performansı deėerlendirilmiřtir.

Goldman ve arkadaşları [20], yoğun şekilde konumlandırılmış çok sayıda nesneyi içeren görüntülerle başa çıkmak için sınırlayıcı kutu kullanan ESA tabanlı bir tespit yöntemi önermişlerdir.

Saetchnikov ve arkadaşları [21], önceden eğitilmiş veri kümelerinin sınırlı sayıdaki görüntülerde benzer nesneleri tespit etmesi için farklı derin sinir ağlarının etkinliğine ilişkin karşılaştırmalı bir çalışma ortaya koymuşlardır. Redmon ve Farhadi ise [22] gerçek zamanlı nesne tespiti konusunda YOLO algoritmasının başarısını inceleyen bir çalışma gerçekleştirmişlerdir.

Kenta ve arkadaşları [23], yürüyerek kaydettikleri video görüntüsü üzerinden ESA metodu kullanarak elma ve armutları tespit etmeyi başarmışlardır. Ardışık video karelerindeki aynı meyveleri ayırt etmek için kalman filtresi kullandıkları gözlenmiştir. Sayma işlemini %10 hata oranı ile gerçekleştirmişlerdir. Bu çalışma gerçek zamanlı olarak uygulanmamış olmasına rağmen elde edilen sonuçlar umut verici olmuştur.

Quan ve arkadaşları [16] ise, mahsul aralarında sıra içerisinde bulunan yabancı otları ayıklamak için tekerlekli mobil robotik bir sistem geliştirmişlerdir. Özel olarak geliştirdikleri kesici takımların, mahsule zarar vermemesi için derin öğrenme metodundan faydalanılarak temizleme bölgeleri oluşturulmasına dayalı bir ayıklama modeli önermişlerdir. Deneysel sonuçlara göre, %85.91 temizleme oranına karşın %1.17 mahsul yaralanması sonucu elde etmeyi başarmışlardır.

Tanmay ve arkadaşları [24], derin öğrenme metodunu kullanarak insansız hava aracı üzerinden elde edilen görüntüler ile bitki örtüsü tahminleme çalışması gerçekleştirmişlerdir. Önerdikleri ESA mimarisi ile bitki örtüsü sınıfını hava görüntüsünden bölümlere ayırmak için küresel özellikleri tam olarak çıkarılmasına odaklanılmıştır. Spesifik olarak belirli bir uygulama için geliştirdikleri sinir ağ modeli bitki örtüsü tespiti ve yol hattı çıkarımı konusunda başarılı sonuçlar ürettiği görülmektedir.

Lee ve arkadaşları [25], görüntü işleme teknolojisi kullanarak nesne tanıma işleminin gerçekleştirilmesini insansız hava aracının üzerinde uygulamak yerine verileri uzaktaki bir bulut sistemine taşıyarak robotun ağırlığını önemli ölçüde azaltmayı hedeflemişlerdir. Amazon Web Services gibi ticari bilgi işlem bulutları, neredeyse sınırsız bilgi işlem

kaynaklarına isteğe bağlı erişime izin verme avantajına sahiptir. Bu özellik, navigasyon ve kontrol işlemlerinde dron uygulamaları için kullanışlıdır, ancak bilinmeyen bir nesne tespit edildiğinde veya aktif nesne arama ve izleme sırasında yoğun hesaplamaların kısa sürede yapılması gerekmektedir. Bulut sistemi ile, tanıma algoritması olarak uygulanan bir ESA yöntemi olan Faster R-CNN'leri, gerçek zamanlı olarak yüzlerce nesne türünü tespit etmek için kullanılmıştır. Tabi ki, bulut üzerinden tanıma işlemi, iletişim gecikmelerinden dolayı beklenmeyen gecikmelere neden olabilmektedir. Bu nedenle, gerçek zamanlı navigasyon ve kararlılık için gerekli olan düşük seviye özellik eşleştirmesinin yanı sıra, ilgilenilen nesnelere karşılık gelebilecek bir görüntünün bölgelerini hızlı bir şekilde tanımlayan bir triyaj adımı da dahil olmak üzere bazı görsel işlemler yerel olarak korunmaktadır. Gerçekleştirdikleri çalışma ile düşük maliyetli bir hava aracının yüzlerce nesneyi neredeyse gerçek zamanlı olarak tespit etmesini sağlayan karma bir yaklaşım önermişlerdir. Düşük maliyetli Parrot AR Drone 2.0'ı donanım platformu olarak kullanılan çalışmada, iç mekan ortamında arama yapan nesneler senaryosunda doğruluk, tanıma süresi ve gecikmeleri ölçen deneyler yapılmıştır.

Bu tez çalışmasında, geliştirilen uçan robotik sistem elma bahçesi içerisinde hedeflenen görevi yerine getirebilmesi için dar bir koridorda otonom olarak yörünge takibi gerçekleştirmesi gerekmektedir. Bu sebeple, çalışmanın temel konusu olan derin öğrenme temelli nesne tespiti ve sayma çalışmasının yanında dört rotorlu insansız hava araçlarının uçuş dinamiği ve kontrolü konusunda da literatür incelemesi gerçekleştirilmiş olup bu problem için de bir çözüm önerilmiştir.

Teng ve arkadaşları [26], gerçekleştirdikleri çalışmada bilinmeyen değişken yüklere sahip bir dört rotorlu insansız hava aracı (İHA) sistemi için uygulanabilir denge ve konum kontrolü problemini incelemişlerdir. Geri yayılım sinir ağlarına dayalı olarak, bilinmeyen değişken yüklere sahip dört rotorlu İHA için uyarlanabilir bir PID denetçisi önermişlerdir. Sinir ağı kullanarak, K_p , K_i ve K_d parametrelerini çevrimiçi olarak ayarlamaya dayalı yöntem ile değişken yük sorununa çözüm önermişlerdir.

Sabir ve Alia [27], dört rotorlu İHA'nın verilen yörüngeyi izlemesi amacıyla bir PID denetleyici tasarlamışlardır. Bu denetleyicinin performansı doğrusal olmayan model için sonuçlar analiz edilmiştir.

Bu deneysel tez çalışmasında, endüstriyel çapta üretim gerçekleştirilen bir elma bahçesinden görüntü verileri toplanarak SSD-MobileNet ve Faster R-CNN mimarilerine dayalı derin sinir ağ modeli geliştirilmiştir. Bu sinir ağ modeliyle yine bu tez çalışması için özgün olarak geliştirilen URS'den elde edilen görüntüler üzerinde bahçedeki elmaların tespiti çalışması gerçekleştirilmiştir. Elmaların doğru sayılması konusunda nesne takibi ve sayma yöntemlerinden faydalanılarak uygulanan metotların performansları karşılaştırılmıştır. Ayrıca dar ağaç koridorlarında otonom yörünge takip edecek şekilde geliştirilen hava aracının K_p , K_i ve K_d parametreleri dar alanda daha stabil hareket için geliştirilen sinir ağı modeliyle optimal değerleri tahminlenerek ayarlanmıştır. Sinir ağ modelinin eğitimi uçuş simülasyonları ile elde edilen veriler ile gerçekleştirilmiş olup sonuçlar gerçek ortamda test edilmiştir.

2. BÖLÜM

DENEYSEL SİSTEM

Bu bölümde insansız hava araçları hakkında genel bilgiler, deneysel çalışmalara destek amacıyla geliştirilen uçan robotik sistemin tasarım ve imalat aşamaları, sistemin genel donanımsal ve yazılımsal yapısı hakkında detaylı incelenmesi gerçekleştirilmiştir.

2.1. İnsansız Hava Araçları (İHA)

Son yıllarda, İHA'lar gözetleme, keşif, haritalama, fotoğrafçılık, arama ve kurtarma gibi hem sivil hem de askeri alanlarda kullanılmaktadır. Bu uygulama alanlarında, İHA'ların kullanımının sağladığı önemli avantajlar mevcuttur. Bunlardan bazıları; İHA'lar nitelikli bir pilot gerektirmediği için hayati riski yoktur, maliyeti çok düşüktür, çok uzun ömürlüdür, önceden programlanmış görevleri YKİ ile haberleşme bağlantıları kopsa dahi otonom olarak gerçekleştirebilme özelliğine sahiptir [28]. Şekil 2.1'de gösterildiği gibi farklı amaçlar için kullanılmak üzere çeşitli İHA tipleri geliştirilmiştir. Sabit kanatlı İHA, tek rotorlu helikopterler, döner kanatlı İHA, çok rotorlu İHA, dikey kalkış yapabilen sabit kanatlı İHA, keşif balonları, kuş benzeri hareketli kanatlı İHA'lar vb.



Şekil 2.1. Kanat yapılarına göre farklı tipte İnsansız Hava Araçlarının yapısı.

İHA'lar kanat tipleri bakımından temelde 3 farklı kategoriye ayrılırlar. Bunlar; sabit kanatlı İHA, döner kanatlı İHA ve dikey kalkış yapabilen sabit kanatlı İHA'lardır (DİHA).

2.1.1. Sabit Kanatlı İnsansız Hava Araçları

Sabit kanatlı İHA'lar, aracın ileri hava hızının neden olduğu kaldırma kuvvetini kullanarak uçuşu mümkün kılan önceden belirlenmiş bir kanat profiline sahiptir. İHA'nın eksen kontrolü, kanat üzerine inşa edilmiş yapıların (aileronlar, elevatörler ve rudder) açılma hareketlerine bağlıdır. Bu yapılar, İHA'nın yunuslama, yuvarlanma ve yalpalama hareketlerinin yapılmasını sağlarlar. Elevatör yunuslama hareketini kontrol eder,

aileronlar yuvarlanmayı kontrol eder ve rudder sapmayı kontrol eder. Şekil 2.2, sabit kanatlı bir İHA örneğini göstermektedir.



Şekil 2.2. Sabit kanatlı İHA yapısının görünümü [29].

2.1.2. Döner Kanatlı İnsansız Hava Araçları

Döner kanatlı İHA'lar, daha düşük maliyetli olması, daha kolay kalkış ve iniş imkanı, sabit kanatlı İHA'lara göre daha hassas manevra kabiliyeti ve havada asılı kalma imkanı sağlar. Bununla birlikte, sabit kanatlı İHA'lar daha basit bir yapı, daha az karmaşık bakım, verimli aerodinamik ve daha uzun uçuş süresi ve menzil avantajı sağlar. Döner kanatlı İHA'larda 3 kg'ın üzerindeki taşıma yükleri yaygındır [30-32]. Sabit kanatlı İHA'lar daha çok; geniş alanlarda, ormanların veya doğrusal altyapının (karayolları veya demiryolları) izlenmesinde kullanılır. Öte yandan, döner kanatlı İHA'lar, altyapı denetimi (köprüler, barajlar veya yel değirmenleri), küçük alanların izlenmesi, kargo taşıma ve püskürtme gibi uygulamalara odaklanmıştır. Kullanılan rotor sayısına göre taşıma kapasitesi artmakta ve kullanım alanları farklılık göstermektedir. Bunlardan yaygın kullanılanı ise mikro hava araçları sınıfına giren dört rotorlu İHA'lardır.

Dört rotorlu İHA'lar, küçük boyutları, havada asılı kalabilme, dikey kalkış ve iniş yapabilme ve diğer hava araçlarına göre düşük maliyet gibi farklı özellikleri sebebiyle çeşitli uygulamalar için tercih edilmektedirler. Engebeli araziler veya su yüzeyleri gibi erişimi zor alanlarda kullanılabilmesi, elektrik hatlarının tepeleri [33], hava fotoğrafçılığı [34], atmosferik veri alma [35] ve trafik izleme [36] gibi çalışmalarda bu tip İHA'lar avantaj sağlamaktadır. Ayrıca sürü İHA ve işbirlikçi dronlar ile ilgili son dönemde yapılan çalışmalar [37-41] yaygın hale gelmiştir.

Bununla birlikte, dört rotorlu İHA'ların uçuş dinamiği, bozucu etkilerden güçlü bir şekilde etkilenir. Örneğin, rüzgar, itme ve tork büyüklüğünü ve dört rotorlu İHA'ya etki eden sürüklenme kuvvetini etkiler, sinyal karışıklıkları İHA'nın geç tepki vermesine ve yörüngeyi kaçırmasına sebebiyet verebilir. Ayrıca iyi belirlenemeyen kontrol parametreleri beklenmedik uçuş tepkilerine yol açabilirler. Bu nedenle, bu olumsuz etkenlere karşı çözüm üretmek başarılı ve güvenilir otonom uçuşlar için kritik öneme sahiptir. Şekil 2.3'te örnek olarak bir dört rotorlu mini İHA gösterilmektedir.



Şekil 2.3. Döner Kanatlı İHA yapısının görünümü [42].

2.1.3. Dikey İniş-Kalkış Yapabilen İnsansız Hava Araçları (DİHA)

Dikey iniş-kalkış yapabilme kabiliyetli insansız hava araçları, yapısal olarak döner kanatlı ve sabit kanatlı iki temel uçuş sisteminin hibrit olarak bir araya gelmesinden oluşmaktadır. Bu sistemlerin en önemli avantajları, döner kanatlı hava araçlarının piste ihtiyaç duymayan iniş-kalkış kolaylığı ve sabit kanatlı İHA'ların uzun süre havada kalma özelliklerinin birleştirilmesidir. Ayrıca, döner kanatlı İHA'ların taşıma kapasitelerinin sınırlı olması problemi de bu sistemlerde çözüme kavuşturulmuştur. Tüm özellikler değerlendirildiğinde DİHA, kolayca iniş-kalkış gerçekleştirebilen, yük taşıma kapasitesi ve uçuş süresi bakımından avantajlı bir İHA olarak tanımlanmaktadır.



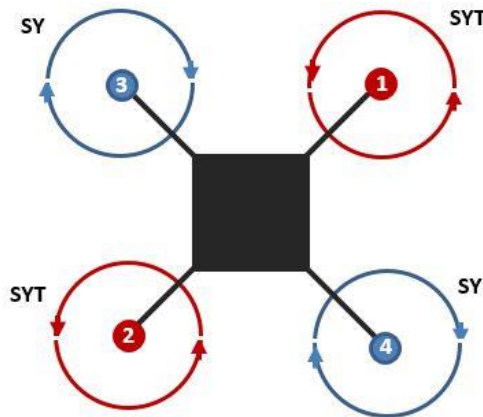
Şekil 2.4. Dikey iniş-kalkış yapabilen sabit kanatlı İHA yapısının görünümü [43].

2.2. Deneysel Çalışmalar için Tasarlanan Sistem

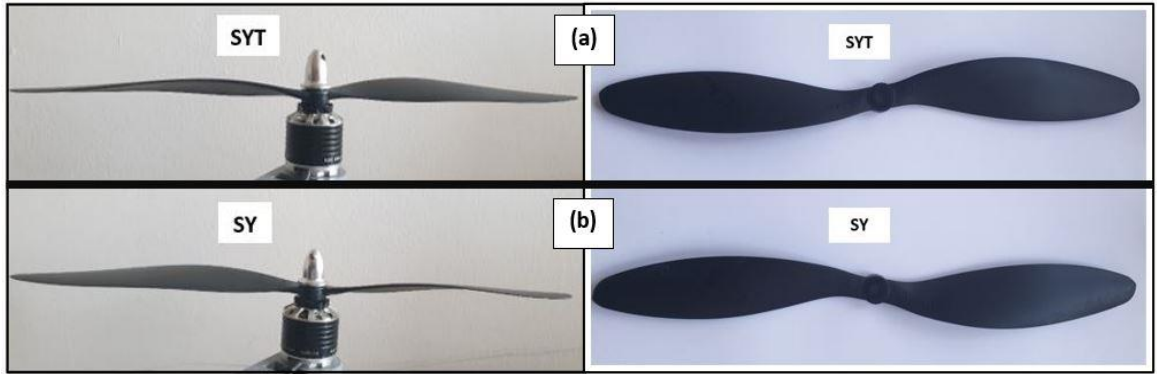
Bu bölümde, geliştirilen uçan robotik sistemin tasarım-imalat süreci, donanımsal yapısı, uçuş planlama programı, sistemler arası haberleşme, kullanılan yazılım ve platformlar detaylı olarak anlatılmıştır. İlk bölümde robotun mekanik yapısı ve hareket denklemleri hakkında bilgi verilmiştir. İkinci bölümde, robotik sistemin imalatı anlatılmıştır. 3. bölümde donanımsal yapı tanıtılmış olup performans testlerinden bahsedilmiştir. 4. bölümde ise yazılım geliştirme süreçleri anlatılmıştır.

2.2.1. Dört Rotorlu Uçan Robotik Sistemin Tasarımı

Gerçekleştirilecek deneysel çalışma için gerekli donanımları taşıyabilecek, küçük boyutlu bir hava aracı tasarımı üzerine çalışılmıştır. Öncelikle dört rotorlu olarak tasarlanan sistemin Şekil 2.5'te gösterilen "X" konfigürasyonuna [44] uygun olan pervaneler belirlenmiştir. Şekil 2.6'da, saat yönünde (SY) dönmesi için tasarlanan pervane tipi ile saat yönünün tersine (SYT) dönmesi için tasarlanan pervane tipi karşılaştırmalı olarak gösterilmiştir. Aracın irtifasını ve açısız konumunu değiştirmek için motorların hızları bu konfigürasyona göre değiştirilmektedir. Daha sonra, çalışmada kullanılması gerekli olan sensörler, aviyonik sistemler belirlenmiştir. Bunların yerleşimi, boyutları ve ağırlıkları hassasiyetle hesaplanmıştır. Donanımsal yapıyı stabil bir şekilde taşıyacak çerçevenin sağlam bir yapı olabilmesi için uygun tasarım, malzeme seçimlerini de kapsayan analizler ile gerçekleştirilmiştir. Bu süreçte, özellikle ağırlık hesabı büyük önem taşımaktadır. Çünkü URS'yi hareket ettirecek itki sisteminin belirlenmesi için yapılacak hesaplamalarda taşınması istenen yük en önemli kriterlerden bir tanesi olacaktır.

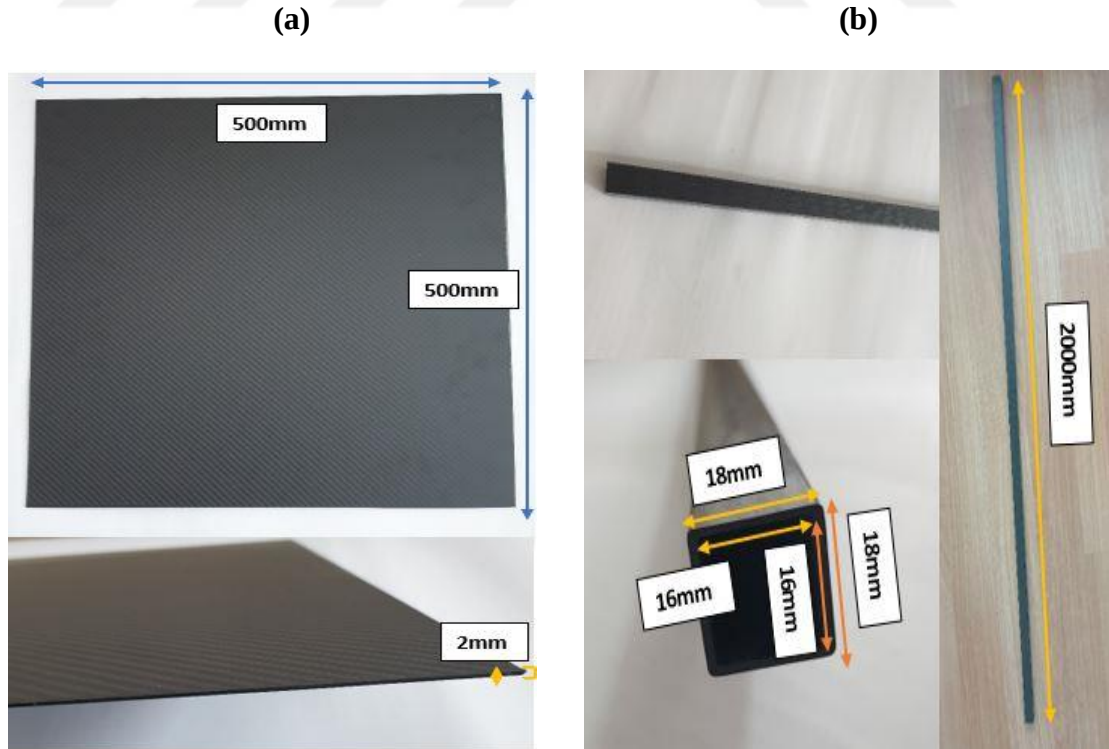


Şekil 2.5. "X" konfigürasyonuna sahip bir dört rotorlu İHA'nın şematik görünümü.



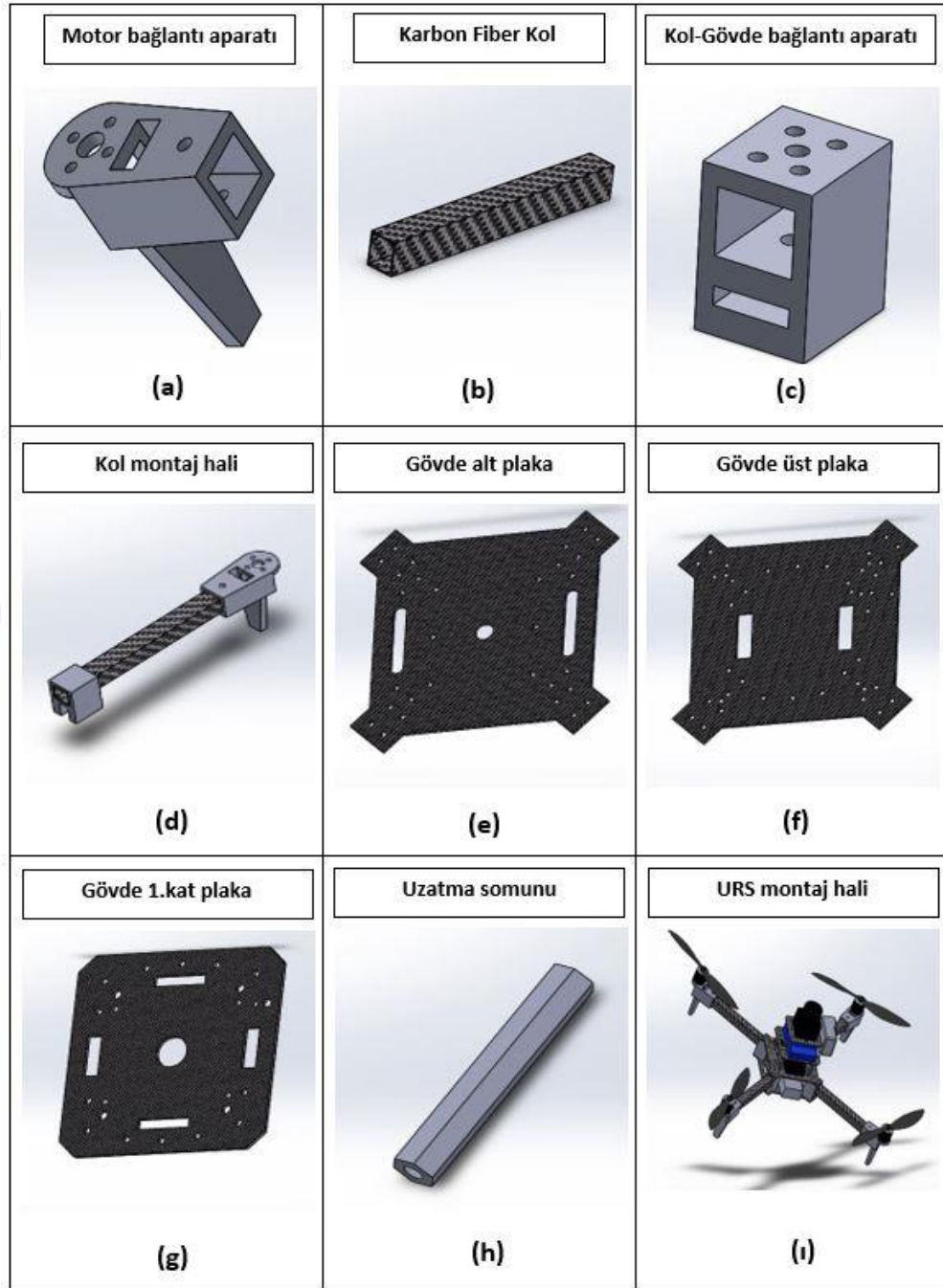
Şekil 2.6. URS’de kullanılan (a) saat yönünün tersine (SYT) ve (b) saat yönünde (SY) dönmesi için tasarlanan pervaneler.

URS çerçevesinin imalatı için gövdenin ve motorların monte edileceği kolların karbon fiber malzemeden olmasına karar verilmiştir. Bu sayede hem dayanıklı hem de hafif bir yapı elde etmek mümkün olmuştur. Gövde için Şekil 2.7 (a)’da gösterilen 500mmx500mmx2mm ebatlarında karbon fiber plaka, CNC makinada işlenmiştir. Hava aracının motorları tutan kolları da Şekil 2.7b’de gösterilen karbon fiber malzemeden, 18mmx16mmx1mm ebatlarında kare profil kesilerek üretilmiştir.



Şekil 2.7. (a) Gövde imalatında kullanılan karbon fiber plaka ve (b) motorları taşıyan kolların imalatında kullanılan karbon fiber kare profilin görünümü.

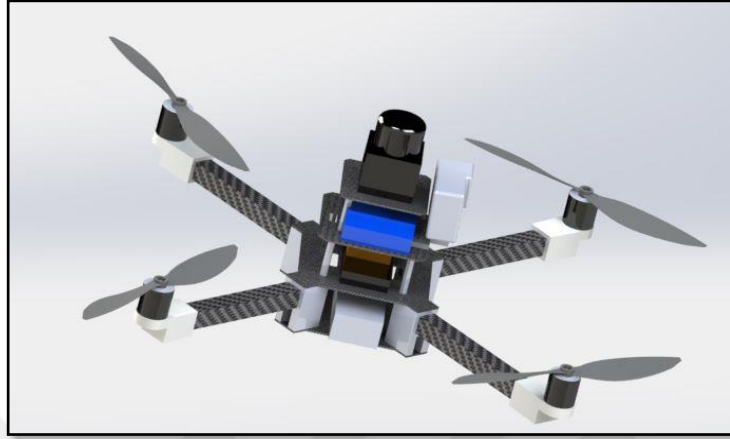
İmalat süreci başlamadan önce Solidworks programında tek tek tasarlanan parçaların (Şekil 2.8), montajı gerçekleştirilerek, gözlemlenen hatalar giderilmiş ve sürekli olarak geliştirilmiştir. Ayrıca ANSYS programında gerilme ve deformasyonlara yönelik analizler de gerçekleştirilerek tasarım gerçekleştirilmiştir.



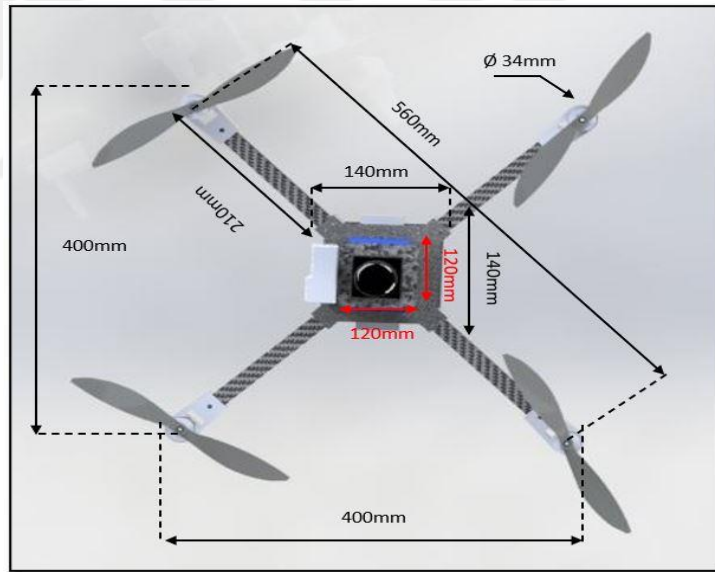
Şekil 2.8. URS parçalarının üç boyutlu katı model çizimlerinin görünümü.

Nihai tasarımın montajı Şekil 2.9’da gösterilmektedir. Ayrıca Şekil 2.10’da tasarıma ait boyutsal özellikler belirtilmiştir. Çerçeve uzunluğu 560 mm, gövdenin dış ölçüleri

140x140 mm, pervane çapı 279,4 mm, toplam yükseklik iniş takımı ile birlikte 420 mm'dir.



Şekil 2.9. URS montajının üç boyutlu tasarım görüntüsü.



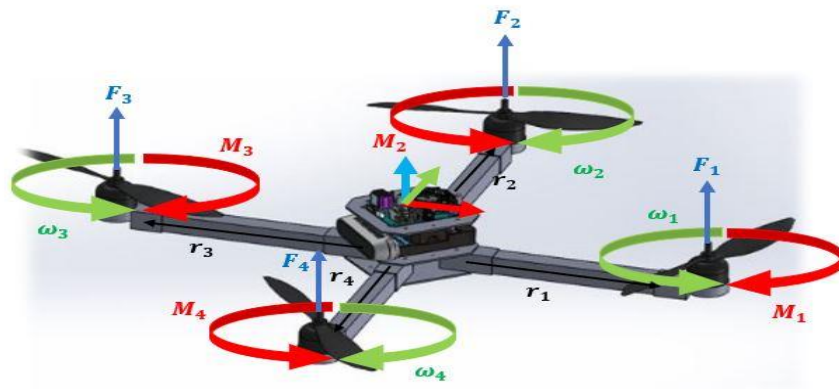
Şekil 2.10. URS'nin boyutsal bir şekilde yapısının görüntüsü.

Tasarımın genel hatları ve kullanılacak ekipmanlar belirlendikten sonra uçan robotik sistemin dolu ağırlığı 1800 gram olarak tartılmıştır. Sistemdeki bileşenlerin ağırlıkları detaylı olarak Tablo 2.1'de verilmiştir. Bundan sonraki süreçte sistemi hareket ettirecek motorun, motor sürücüsünün ve bataryanın belirlenmesi için hesaplamalar yapılmıştır. Bu deneysel çalışma için tasarlanan döner kanatlı İHA'da, bu tarz sistemlerde en uygun eyleyici olan fırçasız doğru akım motorları tercih edilmiştir. Bunun en önemli sebebi daha hafif, kompakt yapıda olması ve optimum enerji ile çok yüksek hızlar elde edilebiliyor

olmasıdır. Fakat buna karşı taşıyabildiği yük miktarı sınırlıdır. Bu sebeple küçük boyutlu İHA'lar için kullanımı son derece uygundur. Tasarlanan URS, dört rotorlu olduğundan hava aracının yerden kalkış yapması için önsel olarak belirlenen değerlere göre her bir motorun sağlayacağı kuvvet, sistemin ağırlığının 1/4'ünden daha fazla olması gerekmektedir. Tabi ki bu hesaplanan minimum şart olup stabil bir uçuş için yine önsel belirlemelere göre motorun %80 verimde bile bu değer yaklaşık %10 daha üzerinde olması beklenmektedir. Şekil 2.11'de dört rotorlu bir sistemin motor dönüş ve moment yönleri gösterilmiştir ve denklem (2.2) ve (2.4)'te kuvvet, moment eşitliği verilmiştir.

Tablo 2.1. Sistemde bulunan temel bileşenlerin ağırlık değerleri.

Sistem Bileşeni	Ağırlık
Gövde-Çerçeve	600 gr.
İniş Takımı	180 gr.
Motorlar (4 adet)	256 gr.
EHK (4 adet)	104 gr.
Batarya	240 gr.
Harici Kontrolör	55 gr.
Kamera	165 gr.
Lidar Sensör	140 gr.
Diğer Bileşenler	60 gr.



Şekil 2.11. Dört rotorlu bir İHA'nın temel kuvvet ve moment parametrelerinin gösterimi.

$$F_i = k_F \omega_i^2 \quad (2.1)$$

$$F = F_1 + F_2 + F_3 + F_4 - m g a_3 \quad (2.2)$$

$$M_i = k_M \omega_i^2 \quad (2.3)$$

$$M = r_1xF_1 + r_2xF_2 + r_3xF_3 + r_4xF_4 + M_1 + M_2 + M_3 + M_4 \quad (2.4)$$

Eşitliklerdeki, F_i : i. motorun pervaneyi döndürmesi sonucu oluşan kuvveti, M_i ise dönme momentlerini ifade etmektedir. k_F : kuvvet sabiti, k_M : moment sabiti, ω_i ise açıl dönme hızlarıdır.

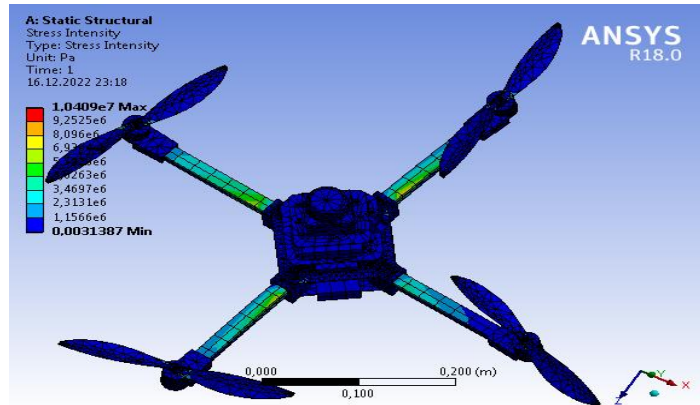
Bu denklem eşitliğine göre her bir motorun, itki sisteminin ağırlığını da sisteme ekleyerek, maksimum verimde en az 6 N'luk bir kuvvet üretmesi beklenmektedir. Buna göre katalog değerlerine bakılarak fırçasız doğru akım motoru seçimi gerçekleştirilmiştir. Seçilen motorun ağırlığı 65 gramdır. Maksimum 24 amper akım çekmektedir ki bu değer EHK (Elektronik Hız Kontrolcü) motor sürücü seçimi noktasında önemli bir parametredir. Bu tarz motorlar yüksek akım çektikleri için EHK motor sürücüler ve lityum polimer batarya gibi bu akımı güvenli bir şekilde sağlayacak ekipmanlar seçilmelidir. EHK seçerken motorun çektiği maksimum akımdan daha yüksek akımda çalışan özellikte olmasına dikkat edilmelidir. Burada %25 motorun akımından daha yüksek bir emniyet oranı belirlenmiş olup 30 amperin üzerinde EHK modelleri araştırılmıştır. Piyasadaki modeller incelendiğinde uygun aralıkta 30 ve 40 amperlik modellerin olduğu belirlenmiş olup bu çalışmadaki deneysel sistem için 40 amper değerinde bir EHK motor sürücü seçilmiştir. Seçilen motor sürücünün ağırlığı 25 gramdır. Son olarak da gerekli itkiyi sağlayacak voltaj değerini sağlayan, istenilen uçuş süresini karşılayabilecek kapasiteli ve 40 amper EHK ile uyumlu lityum polimer bataryanın belirlenmesi gerçekleştirilmiştir. Bu tarz bataryalarda kritik 3 parametre vardır. Bunlar; hücre sayısı (v), toplam kapasite (Ah) ve deşarj katsayısı (c) değeridir. Her bir hücre sayısı 3.7 voltluk bir gerilim sağlamaktadır. Bu çalışmada tasarlanan sistem için belirlenen motorun istenilen değerlere ulaşabilmesi için hesaplamalara göre 14,8 voltluk gerileme sahip bir lityum polimer bataryaya ihtiyaç duyulmaktadır. Bu sebeple bataryanın 4 hücre olması gerektiği belirlenmiştir. Ayrıca 40 amper değerinde bir EHK motor sürücüsü kullanılacağından c değeri 40 ve üzeri olacak şekilde seçim yapılmıştır. Son olarak sistemin deneysel çalışmalar için en az 5 dakika havada kalması yeterli olacağı düşünüldüğünden yerde geçirilecek hazırlık aşamaları da hesaba katılarak emniyetli olarak maksimum güçte 10 dakika çalışabilecek kapasitede bir batarya seçilmiştir. Bunun için denklem (2.5)'teki hesaba göre en az 3000 mAh kapasiteye sahip 240 gram ağırlığında bir batarya belirlenmiştir. Tabi ki çok daha güçlü bir motor, yüksek akım değerli bir motor sürücüsü ya da çok daha büyük kapasiteli bir batarya seçimi

yapılabilirdi. Fakat görüldüğü gibi bu üç bileşen birbirleriyle ve toplam ağırlık ile ilişkili olduğu için birinin büyümesi diğerinin de büyümesine ve sistemin toplam ağırlığının artmasına sebebiyet vermektedir, bu da tekrar motorun ve diğer bileşenlerin büyümesi anlamına gelmektedir. Burada önemli olan planlanan işlemi yerine getiren optimum bileşenlerin en az ağırlık ile belirlenmesidir ki bu çalışma kapsamında gerçekleştirilen hesaplamalar tam olarak bunu sağlamıştır. Sistemin, yeni eklenen motor, motor sürücü ve batarya ile birlikte ağırlığı 1800 gram olarak hesaplanmıştır.

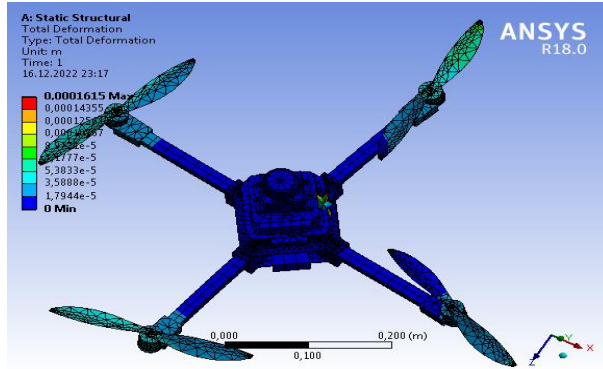
$$C_{batarya} = i_{motor} \times t_{uçuş} \quad (2.5)$$

$C_{batarya}$: pilin birim zamandaki taşınan elektrik yükünü ifade eder, mAh (miliamper saat) birimindedir. I_{motor} : uçuş sırasında motorun çektiği mA (miliAmper) cinsinden ortalama akım değeridir. $t_{uçuş}$: İHA'nın uçuş süresini tanımlar, birimi saattir.

Bu deneysel çalışma için tasarlanan dört rotorlu uçan robotik istemin imalat aşamalarına geçilmeden önce son olarak ANSYS programında gerilme ve deformasyon analizleri gerçekleştirilmiştir. Denklem (2.1)'deki hesaplamalara göre elde edilen kuvvet değerleri ve malzeme seçimleri gerçeğe bire bir uygun olarak ayarlandıktan sonra elde edilen nihai tasarıma ait Şekil 2.12'de uygulanan gerilim analizi ve 2.13'te toplam deformasyon analizi sonuçları gösterilmektedir.



Şekil 2.12. Tasarlanan URS'nin mekanik yapısı için uygulanan gerilim analizinin görüntüsü.

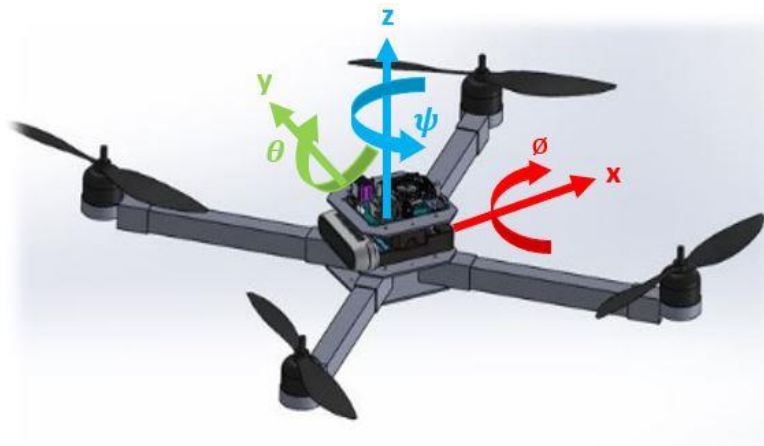


Şekil 2.13. Tasarlanan URS'nin mekanik yapısı için gerçekleştirilen toplam deformasyon analizinin görüntüsü.

Analiz sonuçlarından da görüleceği gibi kuvvete bağlı gerilme deformasyonlar en fazla motor ile gövde arasındaki bağlantı noktalarında gerçekleşmektedir. Fakat geliştirilen sistem için beklenen gerilme ve deformasyonlar problem çıkartacak boyutlarda değildir. Sistem son derece güvenli kriterlerde tasarlanmıştır.

2.2.1.1. Dört Rotorlu Hava Araçlarının Kinematikği

Kinematik, nesnelerin devinimleriyle ilgilenen bir hareket bilimidir [45]. Üzerine etki eden kuvvetleri dikkate almadan parçacıkların veya cisimlerin hareketini tanımlamak için kullanılmaktadır. Dört rotorlu hava araçlarında hareketi tanımlayan altı serbestlik derecesi bulunmaktadır. Bunlardan üç tanesi Şekil 2.14'te gösterilen (x,y,z) eksenleri boyunca lineer hareketi tanımlamaktadır. Diğer üç serbestlik derecesi ise yine aynı şekil üzerinde, gövdenin ortasında tanımlı koordinat eksen takımının her bir eksen etrafında (ϕ , θ , ψ) açıları ile gerçekleştirilen açısal hareketini göstermektedir.



Şekil 2.14. Dört rotorlu bir İHA'nın hareket eksenlerinin gösterimi.

Dört rotorlu hava aracının dönme hareketleri sonucu mevcut konumunu hesaplamak için dönüşüm matrisleri kullanılmaktadır. Hava aracının merkezine yerleştirilen koordinat eksen takımı referans alındığında, x eksenini etrafındaki dönme matris formunda denklem (2.6)'daki gibi tanımlanır:

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix} \quad (2.6)$$

Y eksenini etrafındaki dönme denklem (2.7)'deki gibi tanımlanır:

$$R_y = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \quad (2.7)$$

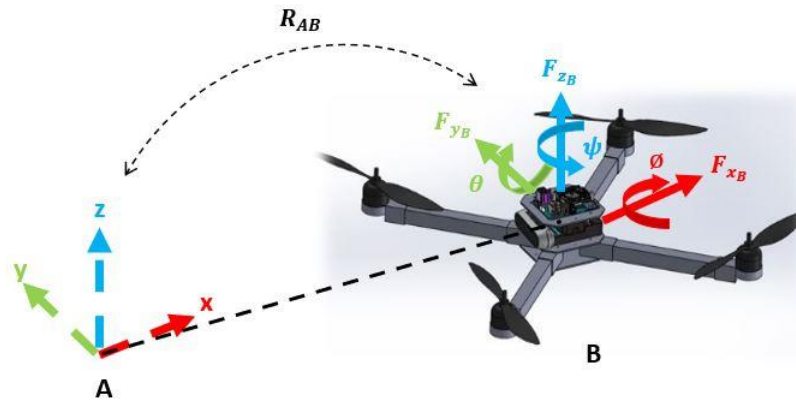
Z eksenini etrafındaki dönme denklem (2.8)'deki gibi tanımlanır:

$$R_z = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.8)$$

Hava aracının Şekil 2.15'teki B konumunun A konumuna göre, dönüşüm matrisi aşağıda gösterilen denklem (2.9)'a göre hesaplanmakta ve (2.10)'daki gibi elde edilmektedir:

$$R_{AB} = R_x R_y R_z \quad (2.9)$$

$$R_{AB} = \begin{bmatrix} \cos \psi \cos \theta & -\sin \psi \cos \phi + \cos \psi \sin \theta \sin \phi & \sin \psi \sin \phi + \cos \psi \sin \theta \cos \phi \\ \sin \psi \cos \theta & \cos \psi \cos \phi + \sin \psi \sin \theta \sin \phi & -\cos \psi \sin \phi + \sin \psi \sin \theta \cos \phi \\ -\sin \theta & \cos \theta \sin \phi & \cos \phi \cos \theta \end{bmatrix} \quad (2.10)$$



Şekil 2.15. Dört rotora sahip İHA'nın hareket eksenlerinin şematik gösterimi.

2.2.1.2. Dört Rotorlu Hava Araçlarının Dinamiği

Dinamik, sistemin hareketine etki eden kuvvetlerin ve torkların etkilerini araştıran mekaniğin bir dalıdır. Dört rotorlu bir hava aracının dinamiği genel olarak uzayda öteleme ve dönme hareketi esnasında meydana gelen etkileri kapsamaktadır. Newton-Euler, Lagrange, Gibbs-Appell ve Jourdain gibi çeşitli yaklaşımlar rijit cisim dinamiklerini elde etmek için kullanılmaktadır [46].

Newton-Euler metodu, bu deneysel çalışma için geliştirilen türdeki mini hava araçlarının dinamiğini ifade etmek için kullanılmaktadır. Dönen rijit URS gövdesinin dinamiği Newton-Euler denklemleriyle aşağıdaki denklem (2.11)'deki gibi tanımlanmaktadır [27]:

$$\begin{bmatrix} F \\ \tau \end{bmatrix} = \begin{bmatrix} mI_{3 \times 3} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \dot{V} \\ \dot{\omega} \end{bmatrix} + \begin{bmatrix} \omega \times mV \\ \omega \times I\omega \end{bmatrix} \quad (2.11)$$

burada F ve τ sırasıyla kütle merkezine etki eden toplam kuvveti ve torku, m ise cismin kütleini ifade etmektedir. $I_{3 \times 3}$ birim matris, I kütle merkezine göre atalet momenti, ω cismin açısal hızı ve V kütle merkezinin doğrusal hızıdır.

2.2.1.3. Dört Rotorlu Hava Aracının Hareket Denklemleri

Geliştirilen dört rotorlu robotik sistem için öteleme hareketinin dinamiği Newton'un ikinci hareket yasasına göre aşağıdaki gibi ifade edilir:

$$F = m\dot{V} + \omega \times mV \quad (2.12)$$

Denklemin ikinci kısmı ($\omega \times mV$) merkezci kuvvetleri tanımlamaktadır. Denklemin çözümünün kolaylaştırılması için merkezci ve coriolis kuvvetleri ihmal edilmektedir. Geliştirilen sistem için lineer olmayan denklemin, ihmal edilen kuvvetlerden sonraki hali aşağıdaki gibi ifade edilmektedir:

$$m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} \quad (2.13)$$

Gövde çerçevesi üzerinde araca etki eden toplam itme kuvveti aşağıdaki gibidir:

$$T = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \sum_{i=1}^4 T_i \end{bmatrix} \quad (2.14)$$

$$T_{zB} = \sum_{i=1}^4 T_i = b \sum_{i=1}^4 (\omega_i)^2 \quad (2.15)$$

denklem (2.15)'te T_{zB} : B noktasında z eksenindeki toplam itme kuvvetini, T_i : i. motorun ürettiği itme kuvvetini, b : itme katsayısı, ω_i : i. pervanede oluşan açısal hız olup pervane hızının $r_{pervane}$ yarıçapına bölünmesine eşittir. URS'nin A noktasından B noktasına gerçekleştirdiği yönelim için, gövdesine etki eden bütün kuvvetler dikkate alınarak ortaya çıkan hareket denklemleri aşağıdaki gibidir:

$$F_E = R_{AB} T_B + F_g + F_D \quad (2.16)$$

$$m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = R_{AB} \begin{bmatrix} T_{xB} \\ T_{yB} \\ T_{zB} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + \begin{bmatrix} -k_{dx}\dot{x} \\ -k_{dy}\dot{y} \\ -k_{dz}\dot{z} \end{bmatrix} \quad (2.17)$$

$$F_x = T_{zB}(\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi) - k_{dx}\dot{x} \quad (2.18)$$

$$F_y = T_{zB}(\cos\phi\sin\theta\sin\psi + \sin\phi\cos\psi) - k_{dy}\dot{y} \quad (2.19)$$

$$F_z = T_{zB}(\cos\phi\cos\theta + \sin\phi\sin\psi) - mg - k_{dz}\dot{z} \quad (2.20)$$

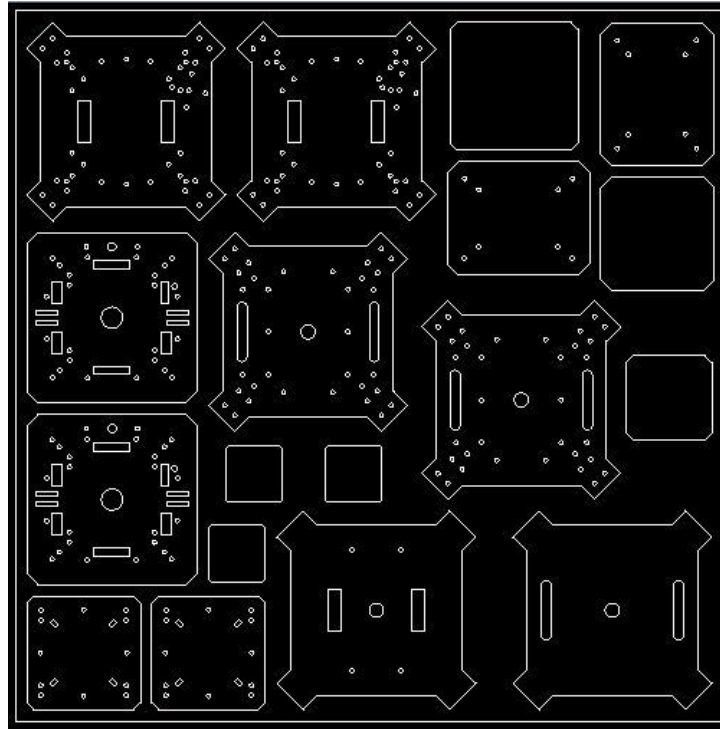
Eşitliklerde verilen, R_{AB} : B noktasının A noktasına göre döndürme matrisi, F_g : yerçekimi kuvvetleri, F_D : sürüklenme kuvvetleri, m : kütle, g : yerçekimi ivmesi, k_d : sürüklenme katsayısıdır.

2.2.2. Dört Rotorlu Uçan Robotik Sistemin (URS) Temel Elemanlarının İmalatı

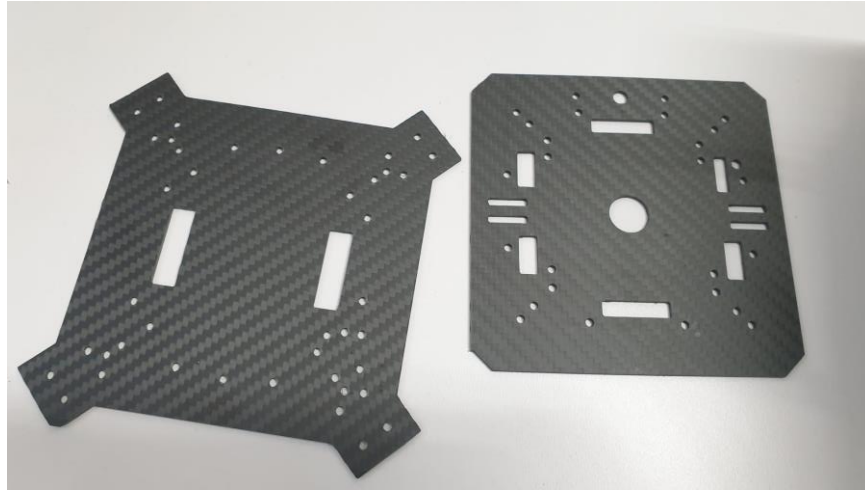
Gerekli hesaplamalar ve donanımsal elemanların seçimi gerçekleştirildikten sonra URS'nin tasarım süreci tamamlanmıştır. Bundan sonraki süreçte tasarım dosyaları uygun formatlara dönüştürülerek imalata hazır hale getirilmiştir. Bu bölümde, tasarlanan parçaların imalat süreçleri ile ilgili detaylı bilgiler verilecek olup üretim aşamasından sonraki montaj ve performans testleri konusunda bilgiler verilecektir.

2.2.2.1. Karbon Fiber URS Çerçeve İmalatı

Deneysel çalışma için geliştirilen URS katmanlı yapı olarak tasarlanmıştır. Ana gövde bloğunu oluşturan iki plaka arasına motor sürücüler ve batarya yerleştirilecek şekilde tasarlanmıştır. Bu sebeple bu ilk katman güç katmanı olarak isimlendirilmiştir. Ana yapının üzerine metal uzatma somunları ile yeni bir plaka montajı gerçekleştirilerek ikinci bir katman daha oluşturulmuştur. Bu katmana da uçuş kontrolcü monte edilmiştir. Sonraki katmanlar kamera, sensör, harici bilgisayar için tasarlanmış olup gerçekleştirilecek uygulamaya göre bu katmanlar sökölüp takılabilir yapıda olacak şekilde imal edilmişlerdir. Ayrıca her bir plaka üzerine yerleştirilecek donanımlar planlanarak uygun konumlara delikler açılmıştır. Buradaki amaç URS üzerindeki bütün donanımların rijit bir yapıda montajının gerçekleştirilmesi ve uçuş dinamiğini bozacak herhangi bir yapının önüne geçilmesi olmuştur. Şekil 2.16'da tasarlanan plakaların imalatı için 2 boyutlu tek bir düzlemde .dwg formatına dönüştürülmüş hali gösterilmektedir. Tasarlanan plakaların karbon fiber plaka üzerinde parça israfı en az olacak şekilde imalatı planlanmış ve CNC tezgahta hassas bir şekilde kesilmeleri için uygun formata dönüştürülmüştür. Kesim sonucu elde edilen ürünlere ait örnek Şekil 2.17'de gösterilmiştir.

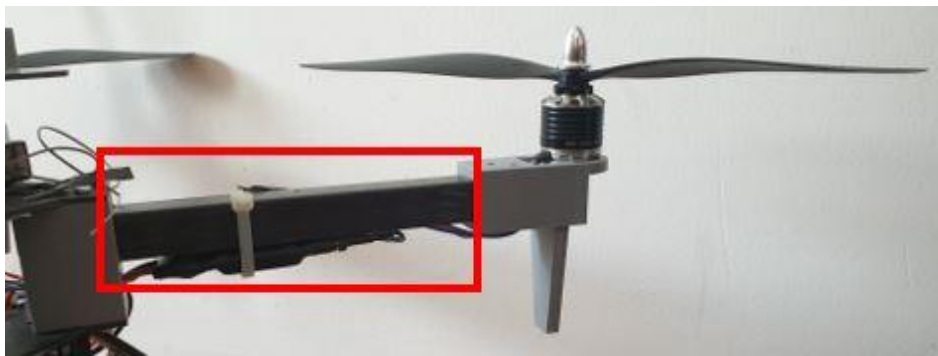


Şekil 2.16. Kesilecek parçaların CNC tezgah için oluşturulan dwg formatı.



Şekil 2.17. Üretilen karbon fiber parçalar.

URS'nin çerçeve imalatının ikinci bölümü ise gövde ile motorları birbirine bağlayan 4 adet kol yapısının üretimini kapsamaktadır. Bunun için yine karbon fiber malzemeden üretilen kare profil içi boş çubuklar tercih edilmiştir. Malzeme dayanıklılığı, hafifliği ve motor sürücülere yataklık yapabilecek profile olması belirleyici kriterler olmuştur. Hesaplanan ölçülere göre bu profiller, Erciyes Üniversitesi Havacılık ve Uzay Fakültesi Laboratuvarında karbon fiber malzemeye zarar vermeden kesim yapabilecek kesici uç ve uygun devirde çalışan kesici ile işlenilmiştir. Kesilen profil parçaları Şekil 2.18'de gösterildiği gibi URS'ye monte edilerek kol yapılarını oluşturmuştur.

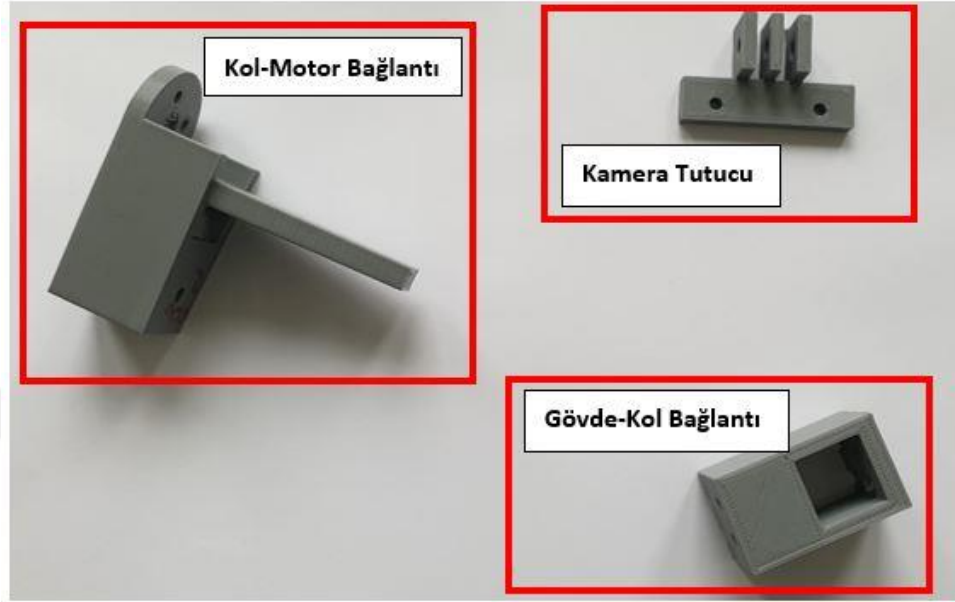


Şekil 2.18. Kare karbon fiber profilden üretilen kol yapısının URS'ye montajı.

2.2.2.2. Bağlantı Elemanlarının Üç Boyutlu Yazıcıda Üretim Safhaları

Geliştirilen sistemin mekanik ana çerçevesinin tamamlanması için gövde, kollar ve motorların birbirlerine rijit bir şekilde monte edilmesi gerekmektedir. Bunun için motor bağlantı deliklerine ve boyutlarına uygun karbon fiber profile sıkı geçmeli bir motor

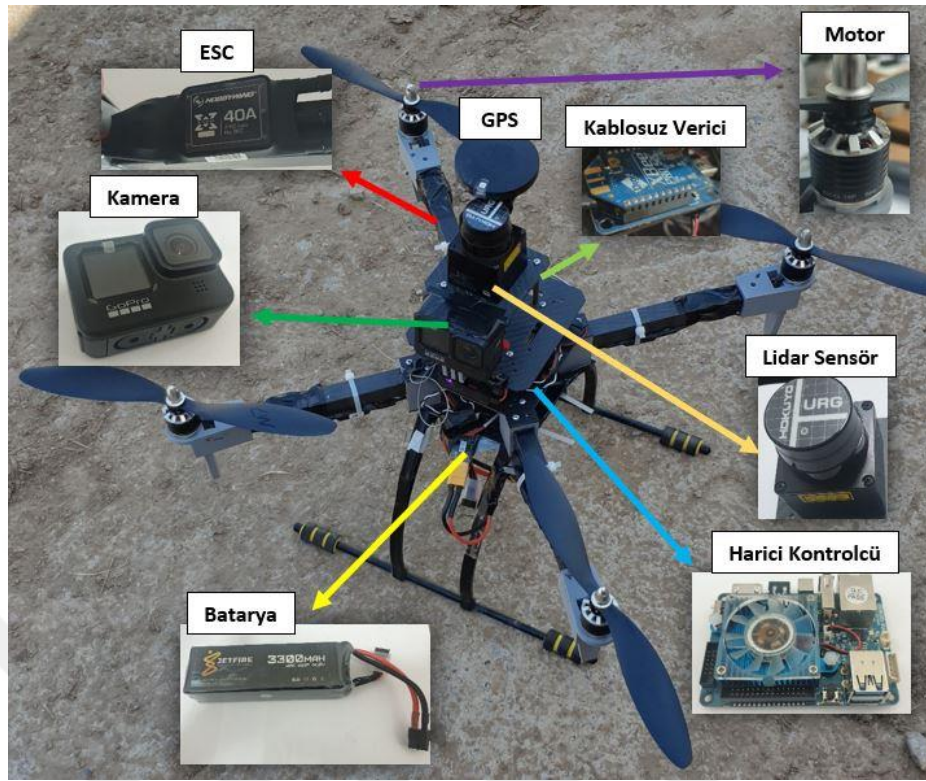
yatağı tasarlanmıştır. Ayrıca alt ve üst karbon fiber plakadan oluşan gövdenin birleştirilip kare profil kollar ile rijit montajı için de bağlantı elemanları tasarlanıp, Şekil 2.19’da gösterildiği gibi ABS plastik malzeme kullanılarak üç boyutlu yazıcıda üretilmiştir.



Şekil 2.19. Tasarlanıp 3 boyutlu yazıcıda üretilen aparatlar.

2.2.2.3. Uçan Robotik Sistemin Tasarım ve Üretimi

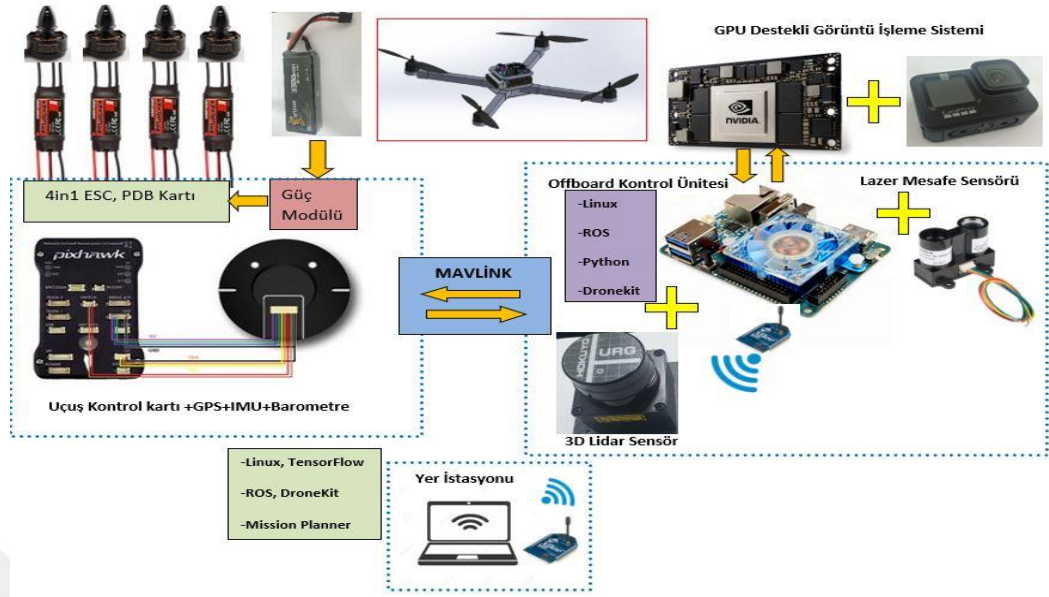
Mekanik parçaların tasarım ve imalatı tamamlandıktan sonra planlanan düzeyde montajları gerçekleştirilmiştir. Bağlantı aparatları kullanılarak gerçekleştirilen birleştirme işleminde sıkı geçme ve M3 metal vidalar kullanılarak vidalama yöntemi uygulanmıştır. Ayrıca robotik sisteme eklenen karbon fiber malzemeden üretilen katman plakaları metal uzatma somunları ve vidalar kullanılarak sisteme monte edilmiştir. Daha sonra tasarımda belirlenen yerlere donanımlar sabitlenerek sistem hazır hale getirilmiştir. Geliştirilen URS nihai montaj hali Şekil 2.20’de gösterilmiştir.



Şekil 2.20. URS ve Temel Elemanları.

2.2.3. Sistemin Genel Donanımsal Yapısı

Geliştirilen dört rotorlu robotik sistemin genel yapısı Şekil 2.21’de gösterilmektedir. Şekildeki sol blok, robotik sistemin otonom veya kumanda kontrollü hareketinin gerçekleştirilmesi için kullanılan itki sistemi ve aviyonikleri kapsamaktadır. Şekildeki sağ blok ise dört rotorlu uçan sisteme robotik özelliği kazandıran alt birimleri ifade etmektedir. Mini bilgisayar, sensörler ve kamera sisteminden oluşan bu alt birim uçuş kontrol sistemini yazılan kodlara göre harici olarak manipüle etmektedir.



Şekil 2.21. Dört rotorlu uçan robotik sistemin donanımsal şeması.

URS'nin temel elemanları daha detaylı olarak açıklanmaktadır:

a. Fırçasız Elektrik Motoru

Gerçekleştirilen hesaplamalar neticesinde minimum 6 N'luk bir kuvvet üretecek fırçasız bir doğru akım motoru kullanılmasına karar verilmiştir. Performans verilerinin güvenilirliğiyle bilinen ve sağlam malzemeden üretim gerçekleştiren KDE Direct firmasından uygun değerleri optimum aralıkta sağlayan Şekil 2.22'de gösterilen 2315XF-885 model fırçasız doğru akım motoru [47] deneysel sisteme entegre edilmek için Tablo 2.2'de örnekleri verilen katalog verilerine göre seçilmiştir.



Şekil 2.22. URS sistemde kullanılan fırçasız doğru akım motoru.

Tablo 2.2. KDE Direct 2315XF-885 fırçasız doğru akım motoruna ait temel parametreler.

Parametre	Sembol	Değer
Motor Hız Sabiti	Kv	885 RPM/V
Motor Tork Sabiti	Kt	0.0108 Nm/A
Maksimum Sürekli Akım	Imax	24 A
Maksimum Sürekli Güç	Pm	355 W
Voltaj Aralığı	Vrange	11.1 V-26.1 V
Şaft Çapı	Dş	Φ 4 mm
Şaft Uzunluğu	Lş	Φ 3.5 mm
Motor Çapı	Dm	Φ 28.3 mm
Motor Uzunluğu	Lm	28.5 mm
Motor Ağırlığı	Gm	64 gr.
Motor Sürücü Frekansı	F	16-32 kHz

b. Motor Sürücüsü

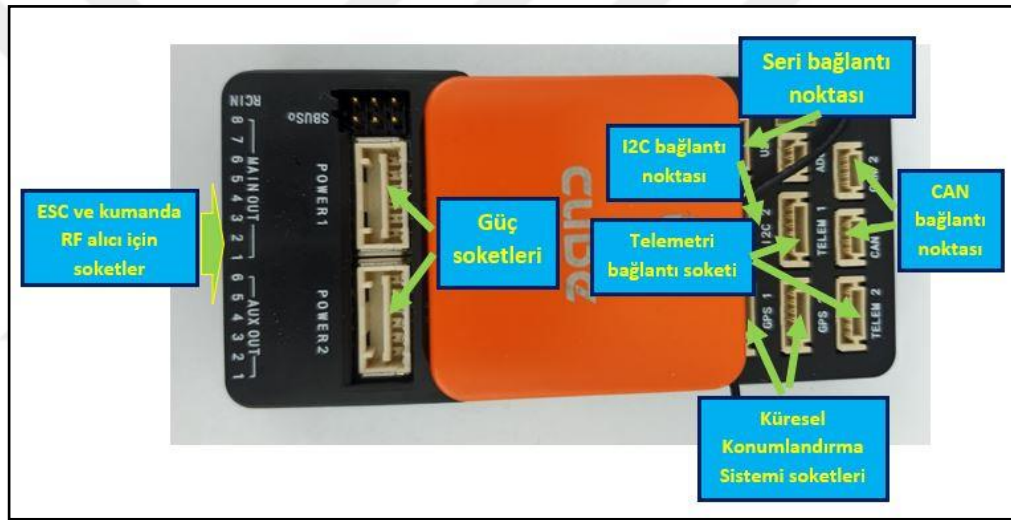
Tasarlanan URS'nin her bir motorunun hızını kontrol etmek için, uçuş kontrol cihazından PWM sinyallerini alan ve motorları sürmek için bunları analog sinyallere dönüştüren toplam dört adet Şekil 2.23'te gösterilen Hobbywing marka EHK devresi kullanılmıştır . Maksimum 24 amper akım çeken motorlar için 40 amper değerinde EHK belirlenmiştir.



Şekil 2.23. Deneysel sistemde kullanılan EHK'nın bağlantı şekli.

c. Uçuş Kontrolörü

Hava araçlarının kullanım amaçlarına göre farklılık gösterebilen çeşitli uçuş kontrolörleri mevcuttur. Genel olarak uçuş kontrolörleri iki gruba ayrılır: açık ve kapalı kaynaklı uçuş kontrolcüler. Açık kaynaklı uçuş kontrolcüler, kullanıcının uçuş kontrolcüsünün hem donanımına hem de yazılımına değişiklik yapmasına imkan sağlarken, kapalı kaynaklı kontrolcülerde böyle bir durum söz konusu değildir. Bu deneysel çalışmada Şekil 2.24'te gösterilen açık kaynaklı bir kontrolör olan Pixhawk Cube Orange kullanılmıştır. Pixhawk uçuş kontrolcüler ile gerçekleştirilen çok sayıda çalışma bulunmakta [48] olup çalışma kararlılığı açısından son derece güvenilirdir.



Şekil 2.24. Pixhawk Uçuş kontrolcüsü ve bağlantılarının gösterimi.

d. Batarya

Geliştirilen uçan robotik sistemin enerjisi lityum polimer batarya tarafından sağlanmaktadır. Şekil 2.25'te gösterildiği gibi batarya olarak Jetfire marka 3300 mAh (miliamper x saat) kapasiteli 4 hücreli bir lityum polimer pil seçilmiştir. Pilin sürekli deşarj oranı 40c değerini desteklemektedir. Bu değer, denklem (2.21) kullanılarak (2.22)'deki gibi pilin kapasite değeri ile çarpıldığında tahrik sistemi tarafından çekilebilecek maksimum akım değeri hesaplanmaktadır. Tercih edilen pil için bu değer 132 amperdir. Ancak geliştirilen sistem üzerindeki tahrik sistemi asla böyle bir güce ihtiyaç duymayacaktır. Tasarlanan sistem için deşarj katsayısı (c) değeri, EHK'nın akım değeri olan 40 amperden yüksek olması yeterlidir.

$$I_{max} = C_{batarya} \times c_{şarj} \quad (2.21)$$

$$I_{max} = 3300 \times 40 = 132 \text{ amper} \quad (2.22)$$



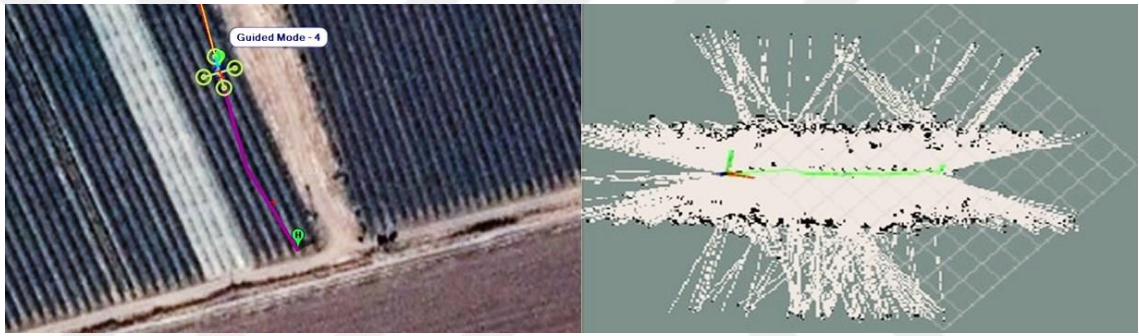
Şekil 2.25. URS için tercih edilen lityum polimer batarya.

e. Lidar Sensör

Ortamın haritasının çıkarılması ve robotun anlık konumunun hesaplanabilmesi için URS'nin üzerine entegre edilerek Hokuyo URG-04LX lidar sensör kullanılmıştır (Şekil 2.26). Ortamın haritasının çıkarılmasında ve robotun anlık konumunun tespit edilmesinde Eşanlı Konumlandırma ve Haritalandırma (EKVH) metodundan faydalanılmıştır. Lidar sensör her 100 milisaniyede 1 tam tur dönmek suretiyle sürekli olarak etrafı taramaktadır. Tarama çözünürlüğü 0,36° olup 30 mm'lik hata payı ile çalışmaktadır. Sensör saniyede 10.000 lazer ışını göndererek çevresindeki engellere çarpıp yansıyan ışınları algılamaktadır [49]. Yansıma sonucu tespit edilen noktalar EKVH yöntemiyle ROS (Robot Operating System) platformunda RVIZ (ROS Visualization) görselleştirme arayüzü kullanılarak veri birleştirme ile harita oluşturulmaktadır. Bu süreç haritalama boyunca sürekli olarak devam etmektedir ve robotik sistemin konumu değiştiğinde yeni gözlemler bir öncekini referans alarak engeller güncellenmekte ve haritanın doğruluğu artırılmaktadır. Bu şekilde elma bahçesinde URS dar koridorlarda hareketini gerçekleştirirken Şekil 2.27'deki gibi ortamın haritasını ve robotun takip ettiği yörüngeyi anlık olarak oluşturulmaktadır.



Şekil 2.26. Hokuyo URG-04LX lidar sensör.



Şekil 2.27. EKVH metoduyla oluşturulan elma bahçesinde bir koridorun haritası [50].

f. Kamera Sistemi

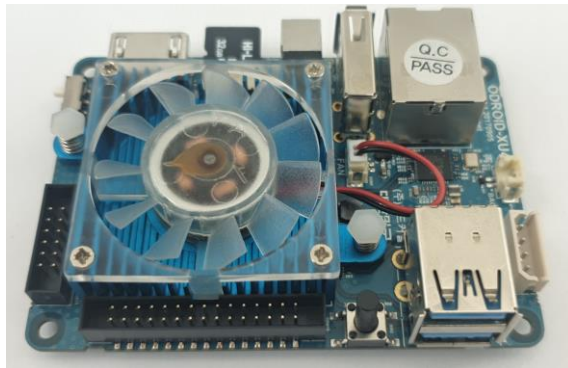
Gerçekleştirilen deneyler için en önemli alt sistemlerden bir tanesi kamera sistemidir. Bu sistem temel olarak bir adet Şekil 2.28'deki GoPro Hero 9 marka aksiyon kamera ve URS'de aynı zamanda harici kontrolü gerçekleştiren kontrolörden oluşmaktadır. Aksiyon kamera tercih edilmesinin sebebi üzerinde tahminleme işlemi uygulanacak görüntülerin harekete bağlı olarak oluşabilecek bozulmaları engellemek ve yüksek çözünürlükte görüntü elde edebilmektir. Ayrıca bu tip kameralar tasarım olarak da mobil sistemler için hem ağırlık hem de ergonomi açısından son derece uygundur. Tercih edilen kamera 1080p çözünürlükte 240 fps çerçeve hızında çekim yapabilmekte ve kablosuz olarak haberleşme imkanı sunmaktadır.



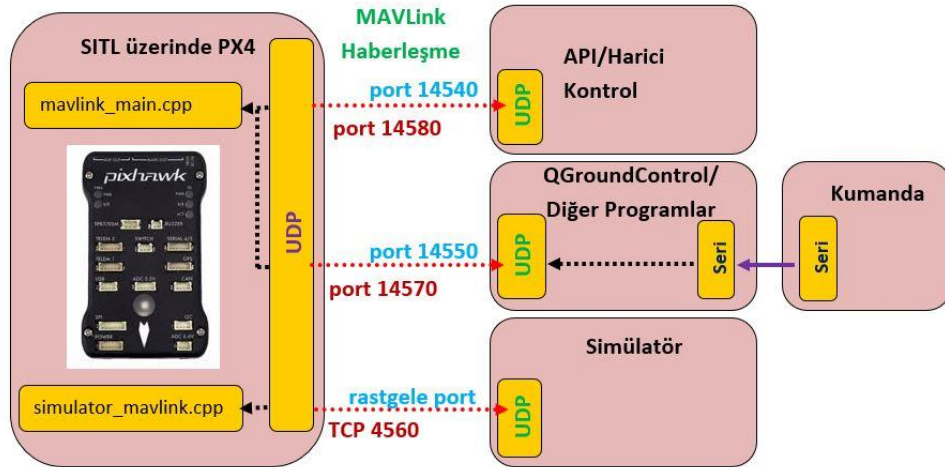
Şekil 2.28. Aksiyon kamerası.

g. Harici Kontrolör

Tasarlanan uçan robotik sistemin uçuş kontrolcüsü Pixhawk Cube Orange donanımı; motorların yönetilmesi, sensör haberleşmesi, uçuş bilgilerinin edinilmesi ve otonom uçuşun gerçekleştirilmesi başta olmak üzere uçuşla ilgili sistemin işletilmesini sağlamaktadır. Bununla birlikte, bu çalışma kapsamında kullanılacak kamera sistemi ve geliştirilen yazılımların URS ile haberleşmesi ve sistemi yönlendirmesi için harici bir kontrolcüye daha ihtiyaç duyulmaktadır. Bu çalışmada, harici kontrolcü olarak Şekil 2.29'daki Odroid XU4 mini bilgisayar modülü olarak isimlendirilen bir kontrolcü kullanılmıştır. Modülün 4 çekirdekli Cortex-A15 2.0 GHz ve 4 çekirdekli Cortex-A7 1.4 GHz, toplam 8 çekirdekli işlemcisi ve 2 GB RAM kapasitesi bulunmaktadır [51]. Bu kontrolcü üzerinde geliştirilen yazılımlar çalıştırılarak MavLink protokolü kullanılarak Şekil 2.30'da şematik olarak gösterildiği gibi Pixhawk ile haberleşmenin sağlanması suretiyle uçuş sistemine komut gönderme ve sistem ile ilgili gerekli parametrelerin alınması sağlanmaktadır.



Şekil 2.29. Odroid XU4 mini bilgisayar.



Şekil 2.30. Uçuş kontrolcü ile harici kontrolcü haberleşmesinin şematik gösterimi [52].

2.2.4. Sistemin Yazılımlarının Geliştirilmesi Süreci

Bu çalışmada, dron sistemi ile farklı yükseklikte ve elma sayısına sahip ağaçlardan oluşan elma bahçesinde; rekolteyi tespit edebilmek için görüntü işleme tekniği ile çalışmalar gerçekleştirilmiştir. Bu hava aracının en önemli özelliği harici bir kontrolcü üzerinden çalıştırılan özgün bir yazılım tarafından yönetilebiliyor olmasıdır. Ayrıca hava aracı üzerindeki kameradan elde edilen görüntülerden, bir yapay zeka tekniği olan derin öğrenme yaklaşımı kullanılarak nesne tanıma kabiliyeti geliştirilmiştir. Bu özelliklerinden dolayı bu hava aracı uçan robotik sistem (URS) olarak isimlendirilmiştir.

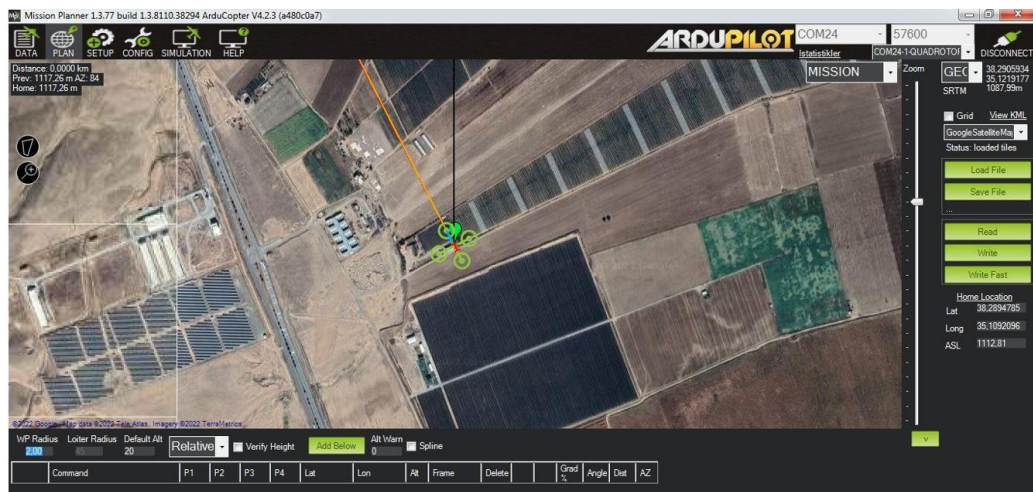
Geliştirilen bu URS üzerinde bulunan Pixhawk Orange Cube uçuş kontrolcü içerisinde kullanılan temel yazılım Ardupilot yazılımıdır. Bu yazılım sayesinde uçuş kontrolcü üzerinden EHK, dolayısıyla motorlar denetlenebilmekte, yer istasyonu ve harici kontrolcüler ile MavLink protokolü kullanılarak haberleşilebilmekte ve Pixhawk'ın giriş çıkışlarına bağlı sensör vb. elemanlar ile veri alışverişi yapılabilmektedir. Ayrıca bu yazılım ve MavLink protokolü kullanılarak URS'ye komut gönderme ve uçuş ile ilgili gerekli verileri izlemek amacıyla yer istasyonu üzerinde bir uçuş planlama yazılımı kullanılmaktadır. Bu çalışmada Mission Planner açık kaynak kodlu uçuş planlama programından faydalanılmış olup çoğunlukla bu amaç için Dronekit kütüphanesi kullanılarak Python programlama diliyle oluşturulan özgün bir yazılım kullanılmıştır. Ayrıca geliştirilen bu yazılım Mission Planner ve Gazebo üç boyutlu simülasyon ortamıyla da haberleştirilerek bu programların üstün oldukları fonksiyonlarından da faydalanılmıştır. Bu doğrultuda deneysel çalışmalardan önce Python ile oluşturulan görev

yazılımları, Mission Planner ve Gazebo ortamlarında simülasyon olarak denenmiş olup optimum uçuş parametreleri belirlenmiştir.

Otonom uçuş sistemiyle ilgili yazılımların geliştirilme süreçlerine ek olarak bu deneysel çalışma için derin öğrenme modelinin geliştirilmesi, nesne tespiti ve nesne sayma uygulamaları için de Python dilinde özgün yazılımların geliştirilmesi de gerçekleştirilmiştir. Görüntüler üzerinden nesne tespit edilmesi için geliştirilen ESA modeli için Tensorflow kütüphanesinden ve nesne sayma uygulama programlama arayüzünden (API) faydalanılmıştır. Ayrıca video üzerinden nesnenin tespit edilmesi ve kümülatif olarak doğru bir şekilde elmaların sayılması için Python programlama dili ev Opencv kütüphanesi kullanılarak program geliştirilmiştir.

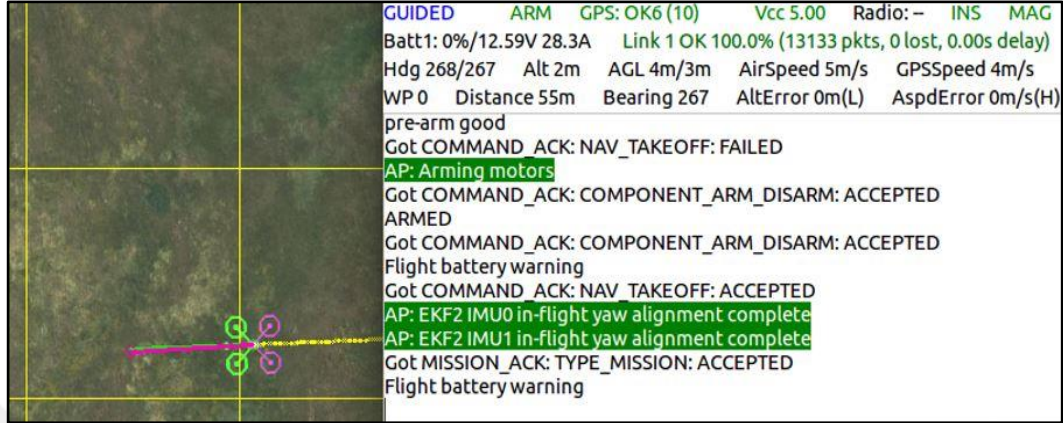
2.2.4.1. Uçuş Planlama ve Simülasyon Programı

Uçuş planlama programları, İHA'lar ile haberleşerek uçuş için belirlenen komutların yürütülmesini gerçekleştiren, uçuş ve hava aracı ile ilgili bilgilerin görüntülenebildiği kullanıcı arayüzlü programlardır. Bu çalışmada uçuş programları Dronekit kütüphanesi kullanılarak Python dilinde yazılmıştır ve otonom uçuş süreci bu şekilde yürütülmüştür. Fakat daha kapsamlı bir arayüz sunan Şekil 2.31'de gösterilen Mission Planner uçuş planlama programı da, bu çalışmalar esnasında eş zamanlı olarak kullanılarak URS'nin uçuşuyla ilgili bilgiler daha verimli bir şekilde gözlenmiştir. Ayrıca URS'nin Kp, Ki ve Kd kontrol parametreleri de yine Mission Planner programı üzerinden değiştirilerek optimize edilmişlerdir.



Şekil 2.31. Mission Planner uçuş planlama arayüzü görüntüsü.

Şekil 2.32’de Dronekit kütüphanesi kullanılarak yürütülen otonom uçuşun yine bu kütüphanede açık kaynaklı olarak sunulan komut ve harita ekranı gösterilmiştir.



Şekil 2.32. Dronekit uçuş izleme ekranı.

2.2.4.1. Python Programlama Dili ve TensorFlow Kütüphanesi

Python programlama dili 1990 yılında Guido van Rossum tarafından geliştirilmeye başlanmıştır [53]. Python, nesne yönelimli ve etkileşimli yüksek seviyeli programlama dillerindendir. Basit söz dizimi sayesinde akılda kalıcı ve hızlı program geliştirmeye uygun bir yapısı mevcuttur. Özellikle hızlı prototip üretilmesi amaçlanan çalışmalarda sıklıkla tercih edilmektedir. Bu deneysel çalışma kapsamında da tüm bu özellikleri sayesinde ve derin öğrenme kütüphanelerinin sunmuş olduğu zengin içeriklerden dolayı tercih sebebi olmuştur.

TensorFlow, derin sinir ağlarının eğitilmesinde kullanılan açık kaynak kodlu bir yazılım kütüphanesidir. İlk olarak Google Brain ekibinde çalışan araştırmacılar tarafından makine öğrenme ve derin öğrenme araştırmalarını yürütmek amacıyla geliştirilmiştir. Bu alandaki güncel gelişmelerin ve uygulamaların kullanılabildiği son derece zengin ve kullanışlı bir kütüphane yapısı sunmaktadır. Bu çalışmada da, elmaların video üzerinden tespit edilmesini sağlayan derin sinir ağ modelinin eğitilmesi ve elmaların ayrı ayrı tespit edilmesini gerçekleştiren Python programlarında bu kütüphaneden faydalanılmıştır.

3. BÖLÜM

YÖNTEMLER

Bu çalışma kapsamında gerçekleştirilen deneylerin büyük bir bölümü elma bahçesinde belirlenen sıra boyunca gerçek zamanlı olarak gerçekleştirilmiştir. Bunun haricinde ön hazırlık aşamasında üç boyutlu ve iki boyutlu simülasyon platformlarında denemeler de gerçekleştirilmiştir. Otonom olarak belirlenen görevi yerine getirmeye hazır hale getirilen URS'nin meyve bahçesinden elde ettiği veriler üzerinden geliştirilen sinir ağ yapısıyla gerçekleştirilen analizler için birtakım yöntemler uygulanmıştır. Aşağıdaki alt başlıklarda öncelikle deneysel ortam hakkında bilgiler verilmiş olup sonrasında bu yöntemler detaylı olarak anlatılmıştır.

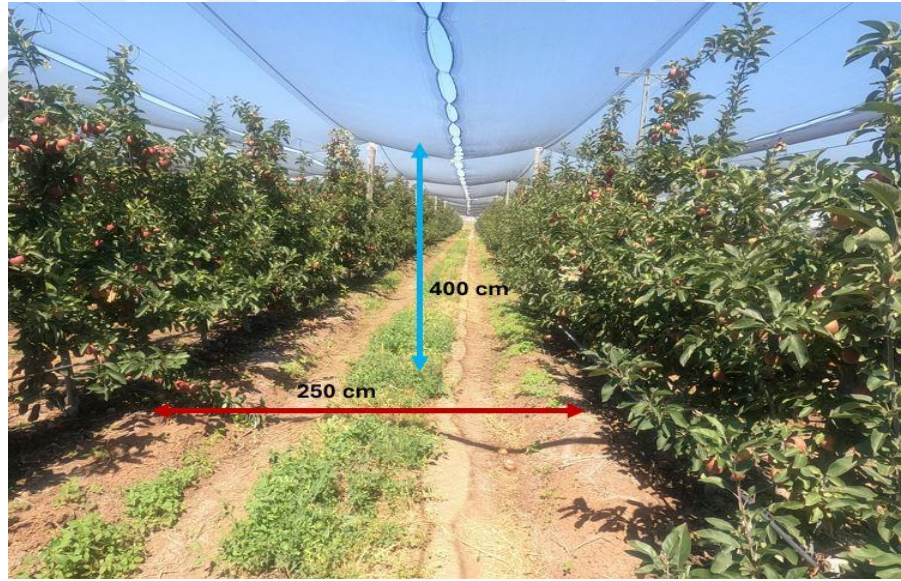
3.1. Çalışma Ortamının Tanıtılması

Bu çalışmanın gerçek zamanlı deneylerinin gerçekleştirildiği elma bahçesi, Kayseri Erciyes Üniversitesi'ne yaklaşık olarak 57 kilometre mesafede Yeşilhisar ilçesinde bulunmaktadır. Bölge toprak bakımından verimli olmasının yanında elma üretimi için son derece önemli olan gece-gündüz arası sıcaklık farkı bakımından da son derece uygun bir iklime sahiptir. Deneysel çalışmaların gerçekleştirildiği, Şekil 3.1'de URS ile havadan çekilen görüntüsü verilen elma bahçesi 350 dönüm büyüklüğündedir. Üzeri ince bir tente ile kaplı olmasının sebebi meyveleri hem don olayından hem de güneş ışınlarının dik gelmesini engelleyerek yanma durumundan korumaktır. Bu tente görüntü işleme çalışmalarında ışığın farklı kırılması sebebiyle sorunlar oluşturmakla beraber uydu sinyallerinin de zayıflamasına sebebiyet vermektedir. Bu tez çalışmasında derin öğrenme tabanlı nesne tespit yöntemi ile görüntü işleme sorunlarına çözüm üretilmiştir. Ayrıca olası sinyal sorunları için de yörünge karşılaştırması amacıyla kullanılan EKVH metodu gelecek çalışmalar için bir çözüm olarak önerilmiştir.



Şekil 3.1. Yeşilhisar’da bulunan elma bahçesinin URS ile havadan görünümü.

Uçan robotik sistemin otonom yörünge takibini gerçekleştirdiği deneysel elma sıra boyu Şekil 3.2’de gösterildiği gibi üzeri tente ile örtülü olup dallar arası 250 santimetre genişliğinde ve 400 santimetre yüksekliğindedir.



Şekil 3.2. Bahçe içerisinde kullanılan koridorun gösterimi.

Çalışmaların gerçekleştirildiği elma bahçesinde toplam 90.000 adet elma ağacı bulunmaktadır. Bunların 71.000 adeti deneysel çalışmada kullanılan kırmızı tip elma ağaçlarıdır. Bu bahçeden tam verimde 1700 ton elma hasadı yapılması olasıdır. Son dört yılın verilerine bakıldığında (2019, 2020, 2021, 2022) sırasıyla 1200 ton, 1600 ton, 1100 ton ve 600 ton elma hasadı gerçekleştirilmiştir. Bu verilerin son derece değişkenlik göstermesi, rekolte tahmininin teknolojik yöntemler ile doğru gerçekleştirilmesinin

önemini göstermektedir. Geçmiş yılların hasat rakamları ile görsel inceleme sonucu gerçekleştirilecek tecrübeye dayalı geleneksel tahminler üreticileri önemli ölçüde zarara uğratmaktadır.

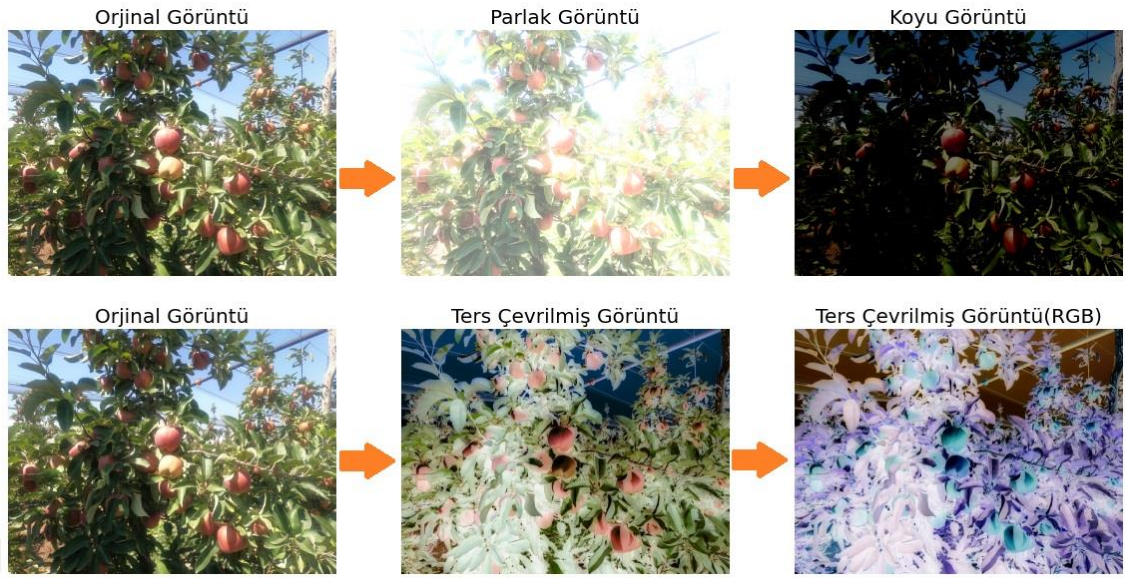
3.2. Görüntü İşleme ve Bilgisayarlı Görü Yöntemi

Görüntü işleme, bir görüntü üzerinden daha gelişmiş bir görüntü elde etmek veya görüntüden bazı yararlı bilgiler çıkarmak için geliştirilmiş bir yöntemidir. Daha özel manada, girişin bir görüntü olduğu ve çıkışın görüntü veya o görüntüyle ilişkili özellikler olabileceği bir sinyal işleme türüdür [54]. Günümüzde, hızla gelişen görüntü işleme teknolojisi, mühendislik ve bilgisayar bilimleri disiplinlerinde de temel araştırma alanını oluşturmaktadır. Görüntü işleme Şekil 3.3'te verildiği gibi; görüntü elde etme araçları aracılığıyla görüntünün içe aktarılması, görüntüyü analiz etme ve düzenleme ve görüntü analizine dayalı olarak sonucun değiştirilebileceği görüntü veya rapor çıktısı olmak üzere 3 temel adımdan oluşmaktadır.



Şekil 3.3. Görüntü işleme akış şeması.

Görüntü işlemede, görüntü analistleri analog ve dijital görüntü işleme olarak iki yöntem kullanırlar. Sayısal görüntü işlemede sayısal görüntüler, dijital görüntü işlemede de dijital görüntüler bilgisayarlar vasıtasıyla işlenmektedir. Şekil 3.4'te görüntü işleme teknikleri ile işlenmiş iki farklı görüntü verilmiştir.



Şekil 3.4. Görüntü işleme teknikleri ile işlenmiş dijital görüntüler.

Görüntü işlemenin farklı temel alanlarında, görüntülerin parçalara ayrılması, sınıflandırılması, boyutları ve tanınması gibi farklı zorlukları vardır. Özellikle görüntülerin sınıflandırılması ve tanınması alanında, ölçek değişimi, bakış noktası değişimi, aydınlatma, nesne algılama ve nesne yerleştirme gibi zorluklar bulunmaktadır. Bu zorlukların aşılması için farklı teknikler kullanılmaktadır. Bilgisayarla görü bu tekniklerden birisidir ve nesne tanıma alanında görüntü işlemenin merkezinde bilgisayarlı görü bulunmaktadır.

Bilgisayarla görü, bilgisayarların video kameralardan alınan sinyaller üzerinde işlemler gerçekleştirerek dijital görüntülerin anlamlı bir şekilde görünmesine ve algılanmasına yardımcı olacak bazı tekniklerin geliştirilmesini amaçlayan bir mühendislik disiplini [55]. Görme işlemi insanlar ve diğer canlılar için zahmetsiz bir yetenek olmasına rağmen, bilgisayar ortamında görme işleminin yapılması için oldukça kompleks bir dizi işlem gerekmektedir. Bu kapsamda bilgisayarlı görme anlamsal bölütleme, sınıflandırma ve yerleştirme, nesne algılama ve örnek bölütleme olmak üzere 4 ana alana ayrılmaktadır [56].

Anlamsal bölütlemede, bir görüntüdeki her piksele bir sınıf etiketi atanmaktadır. Ancak anlamsal bölütlemede, aynı sınıfın örnekleri arasında ayrım yapılamamaktadır. Sınıflandırma ve yerleştirme alanında, bir görüntüdeki belirli bir nesnenin sınıfı tahmin edilmeye ve sınıfı tahmin edilen nesnenin etrafına sınırlayıcı kutu çizilmeye

çalışılmaktadır. Ancak gerçek dünyada çoğu görüntü bu şekilde tek bir nesneden değil farklı şekillerde ve boyutlarda birden çok nesneden oluşmaktadır. Nesne algılama ise, her giriş görüntüsü için farklı sayıda nesneyi tahmin etmeyi amaçlamaktadır. Böylece her nesnenin bir örneği ve uzaysal kapsamı tahmin edilebilmektedir. Böylece sınırlayıcı kutular vasıtasıyla görüntülerdeki her nesnenin yeri ve konumu tahmin edilmiş olmaktadır. Örnek bölütleme alanı ise tüm bu alanların birleşimi gibi düşünülerek, tek tek tüm nesnelerin bir sınırlayıcı kutu ile sınıflandırıldığı ve yerleştirildiği nesne algılama ve her pikselin verilen sınıflarda sınıflandırıldığı anlamsal bölütlemenin birleşmesinden oluşmaktadır. Yani, bir görüntünün aynı sınıfa ait tüm nesneleri tek bir sınıf etiketi altında birleştirilmektedir.

Günümüzde bilgisayarlı görü disiplini ile farklı alanlarda çok sayıda çalışma gerçekleştirilmektedir. Bu alanlardan bazıları;

- Otomatik yüz tanıma ve ifadelerin yorumlanması,
- Otonom araçların görsel rehberliği, çarpışmadan kaçınma ve araç takibi,
- Otomatik tıbbi görüntü analizi,
- Robotik üretimde parçaların manipülasyonu, derecelendirilmesi ve montajı,
- Tarımsal robotlarda, ürünlerin görsel olarak sınıflandırılması ve toplanması,
- Akıllı ofislerde, biyometrik tabanlı görsel tanımlama ve görsel olarak donatılmış robotik yardımcılar olarak karşımıza çıkmaktadır.

Tez çalışması kapsamında, tarımsal alanda ürünlerin görsel olarak sınıflandırılması ve sayılması alanında çalışma gerçekleştirilmiştir. Bu çalışma için bilgisayarlı görüde örnek bölütleme ile nesne algılama tekniğinden yararlanılmıştır.

3.2.1. OpenCV Kütüphanesi ve Görüntü İşleme Temelleri

OpenCV, açık kaynak kodlu bilgisayarlı görü uygulamaları için Intel tarafından geliştirilmiş, görüntü işleme kütüphanesidir [57-59]. C programlama dili ile geliştirilmeye başlanmış OpenCV kütüphanesi daha sonraki dönemde C++ ile geliştirilmeye devam edilmiştir. Projelerde ücretsiz olarak kullanılabilmesine izin verilen BSD (Berkeley Software Distribution) lisansına sahiptir ve ayrıca açık kaynak kodlu olarak geliştirilmektedir. OpenCV, platform bağımsız olmasından dolayı istenilen bilgisayar dili ve platformunda geliştirilebilmektedir. Hem açık kaynak kodlu, hem de

platform bağımsız olmasından dolayı, OpenCV kütüphanesi oldukça fazla kaynağa, topluluğa ve teknik dokümana sahiptir.

OpenCV anlaşılması ve geliştirilmesi kolay modüler bir yapıya sahiptir. Bu modüller aşağıda tanımlanmıştır.

- Çekirdek İşlev (core) modülü; çok boyutlu diziler matris modülü ve diğer tüm modüller tarafından kullanılan temel veri yapılarını tanımlayan kompakt bir modüldür.
- Görüntü İşleme (imgproc) modülü; doğrusal ve doğrusal olmayan görüntü filtreleme, geometrik görüntü dönüştürmeleri (yeniden boyutlandırma, perspektif çarpıtma, genel tablo tabanlı yeniden eşleme, renk alanı dönüştürme, histogramlar vb.) içeren bir görüntü işleme modülüdür.
- Video Analiz (video) modülü; hareket tahmini, arka plan çıkarma ve nesne izleme algoritmalarını içeren bir video analiz modülüdür.
- Kamera Kalibrasyonu ve 3B Yeniden Yapılandırma (calib3d) modülü; temel çoklu geometri algoritmaları, tek ve stereo kamera kalibrasyonu, nesne poz tahmini, stereo yazışma algoritmaları ve 3B yeniden yapılandırma öğeleri modülüdür.
- 2B Özellikler Çerçevesi (feature2d) modülü; göze çarpan özellik algılayıcılar, tanımlayıcılar ve tanımlayıcı eşleştiriciler modülüdür.
- Nesne Algılama (objdetect) modülü; önceden tanımlanmış sınıfların (örneğin, yüzler, gözler, kupalar, insanlar, arabalar vb.) nesnelerinin ve örneklerinin algılanması modülüdür.
- Yüksek seviyeli GUI (Graphical User Interface) modülü; basit UI (User Interface) yetenekleri için kullanımı kolay bir arayüz modülüdür.
- Video I/O (videoio) modülü; video yakalama ve video codec bileşenleri için kullanımı kolay bir arayüz modülüdür.

OpenCV, verilen bu modüler yapısı sayesinde özellikle görüntü işleme alanında en sık ve popüler kullanılan kütüphane olma özelliğine sahiptir. OpenCV içerisinde 2500'den fazla algoritma, makine öğrenmesi ve görüntü işlemeye yönelik olarak kullanılmaktadır. Görüntü işleme uygulamaları arasında; yüz, hareket ve nesne tanıma, nesne sınıflandırma, görüntü karşılaştırma gibi geniş kullanım alanı bulunmaktadır. Günümüzde, Google

tarafından yürütülen cadde ve sokakları haritalama projesi Street View ve NASA tarafından Mars yüzeyini tanımlamak ve görüntülemek için Mars'a gönderilen Curiosity keşif aracında OpenCV kütüphanesi kullanılmaktadır.

Bu tez çalışması kapsamında nesne tanımlama modülü için OpenCV kütüphanesinden yararlanılmıştır. Böylece videodan elde edilen elma görüntüleri OpenCV kütüphanesi sayesinde tanımlanabilmektedir.

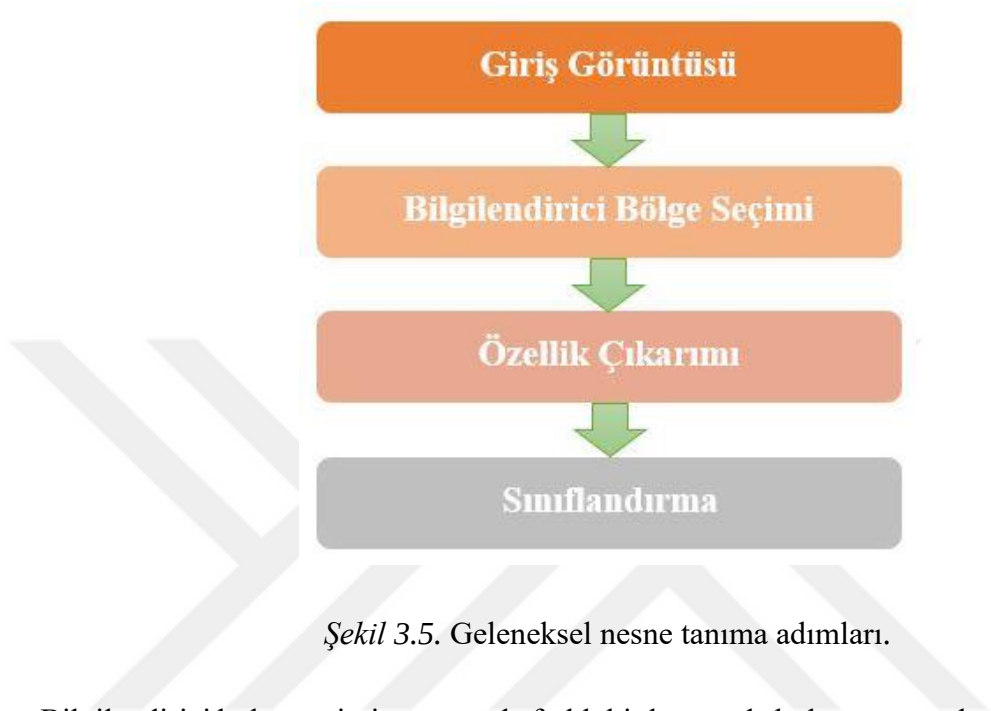
3.2.2. Nesne Tanıma

Nesne algılama, nesnenin yerinin tespiti ile sınıflandırmasının bir kombinasyonu olarak, görüntünün tam ve doğru bir şekilde belirlenmesini ve tanımlanmasını sağlayan önemli bir bilgisayarla görme alanıdır [55,60,61]. Genel olarak nesne algılama işleminde, nesnenin görüntüdeki yerini tam olarak belirtmek ve nesneyi bir etiketle sınıflandırmak için dikdörtgen bir sınırlayıcı kutu kullanılmaktadır. Nesne tanıma algoritmaları, geleneksel yöntemlere dayalı, manuel özellik çıkarımı ve derin öğrenme tabanına dayalı, öğrenerek özellik çıkarımı olmak üzere iki farklı sınıflandırmaya sahiptir. Tablo 3.1'de nesne tanıma algoritmaların şematik olarak sınıflandırılması verilmiştir.

Tablo 3.1. Temel nesne tanıma algoritmaları.

1-Manuel Özellik Çıkarımı	Oxford-MKL	
	DPM	
	NLPR-HOGLBP	
	Selective Search	
2-Öğrenerek Özellik Çıkarımı	Tek Aşamalı Nesne Tanıma	OverFeat YOLO YOLOv2 SSD
	İki Aşamalı Nesne Tanıma	RCNN SPPNet Fast RCNN Faster RCNN R-FCN FPN Mask RCNN

Geleneksel nesne tanıma algoritmaları, nesne tespiti için manuel özellik çıkarma ve ardından basit eğitilebilir mimariler kullanılmaktadır. Şekil 3.5'te bu adımlar sırasıyla verilmiştir.



Şekil 3.5. Geleneksel nesne tanıma adımları.

Bilgilendirici bölge seçimi, görüntüde farklı bir konumda bulunan nesnelerin değişken en boy oranları veya boyutları ile seçilmesi işlemidir. Tüm görüntünün taranması için, çok ölçekli kayan pencere yöntemi kullanılmaktadır. Kayan pencere yapısı yüksek bilgi işlem karmaşıklığına sahip ve ayrıca bölge seçimi, görüntüdeki bir nesnenin olası tüm konumlarında birden çok pencere üretilmesini sağlayan hesaplamalı süreçlerdir.

Özellik çıkarma, nesnenin görsel özelliklerinin çıkarılmasını sağlayan, nesne tanıma için en temel adımdır. Aydınlatma şartları, görünüm ve arka planın farklılıkları nedeniyle her türden nesneyi mükemmel bir şekilde tanımlayan sağlam bir özellik tanımlayıcısının manuel olarak tasarlanması oldukça karmaşıktır.

Sınıflandırma, hedef nesneyi uygun sınıfa yerleştirme işlemidir. Görsel nesne tanıma için temsilin daha bilgilendirici, semantik ve hiyerarşik hale getirilmesine ihtiyaç duyulmaktadır.

Tüm bu işlevler karmaşık ve zaman alıcıdır. Son yıllarda geliştirilen derin evrişimli sinir ağları ve GPU'ların yüksek bilgi işlem güçleri sayesinde nesne algılama teknikleri de oldukça hızlı bir şekilde gelişmiştir ve güvenlik izleme, drone sahne analizi, ulaşım

gözetimi, otonom sürüş, robotik görme, insan davranış analizi, yüz tanıma, tıbbi teşhis gibi hem akademi hem de gerçek dünya uygulamalarında kapsamlı bir araştırma alanını oluşturmuştur. Derin evrişimli sinir ağları, düşük seviyeden yüksek seviyeye kadar çok sağlam olan görüntülerin özelliklerini öğrenebilmektedir. Bu sayede nesne tanıma alanında oldukça performanslı sonuçlar elde edilebilmektedir.

3.2.3. Derin Öğrenme Tabanlı Nesne Tanıma

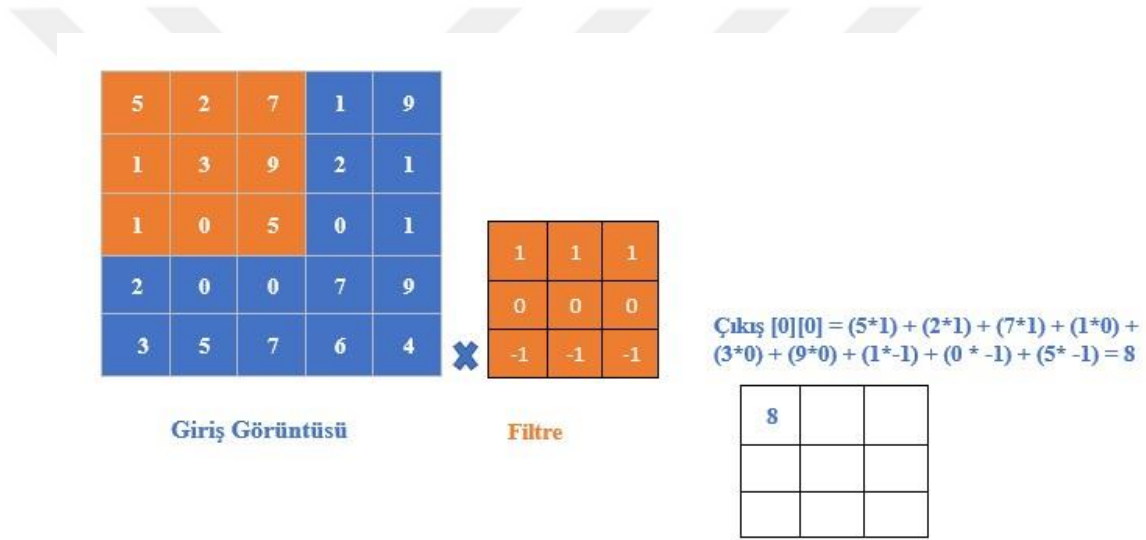
Derin öğrenme yöntemlerine dayalı olarak gelişen öğrenerek özellik çıkarımı tek aşamalı ve iki aşamalı nesne tanıma algoritmaları olmak üzere ikiye ayrılmıştır. Nesne tanıma, nesne konumunun bulunması ve nesnelerin sınıflandırması olmak üzere iki kısımda gerçekleştirilebilmektedir. Tek aşamalı nesne tanıma algoritmaları konumlandırmanın ve sınıflandırmanın tek aşamada yapıldığı nesne algılama mimarileridir. Tek aşamalı nesne algılama, bir aşamada bir nesnenin sınıf olasılıklarını ve konum koordinatlarını doğrudan üretebilmektedir. Böylece yüksek algılama hızına sahip mimariler oluşturulabilmektedir. Ancak algılama doğruluğu düşüktür [62]. OverFeat [63], YOLO serisi [22,64,65], SSD [66], DSSD [67], FSSD [68] ve DSOD [69] tek aşamalı nesne algılama mimarileridir. İki aşamalı nesne tanıma algoritmaları nesne konumunun bulunması ve nesnelerin sınıflandırılması olmak üzere iki kısım içermektedir. İlk kısımda bölge önerisi belirli bir çerçeveye göre oluşturulur, ikinci kısımda ise bu bölge sınıflandırılır. İki aşamalı nesne tanıma yönteminin en önemli özelliği, yüksek algılama doğruluğu kazandırmasıdır. Ancak, iki aşamadan oluştuğu için oldukça yavaş algılama hızına sahiptir. RCNN [70], SPPNet [71], Fast RCNN [72], Faster RCNN [73], RFCN [74], FPN ve Mask RCNN [75] iki aşamalı nesne algılama mimarisi örnekleridir.

3.2.4. Evrişimsel Sinir Ağları (ESA)

ESA, canlıların etrafındaki nesneleri görme ve algılama süreçlerinden esinlenilen yapay sinir ağı örneğidir [76-78]. ESA bu süreci canlıların biyolojik arka planındaki gibi yürütmektedir. Öncelikle, düşük seviyeli öznitelik haritası çıkartılır ve öğrenilir. Ardından üst düzey özellikleri ayıklayarak algılama sürecinin gerçekleşmesi sağlanır. ESA'yı yapay sinir ağlarından ayıran temel nokta öncelikle nesnenin özelliklerinin belirleniyor olmasıdır. Bu sayede hesaplama maliyeti düşürülürken daha iyi sonuçların elde edilmesi sağlanmaktadır.

Genel olarak ESA yapısı 3 temel katmanı içermektedir. Evrişim katmanı ESA'nın ilk katmanıdır; matematiksel bir süreç olarak evrişim işleminin gerçekleştirildiği bu katmanda öznitelik çıkartımı yapılmaktadır. İkinci katman havuz katmanıdır; bu katmanda alt örnekleme işlemi yapılmaktadır. Son katman ise geleneksel yapay sinir ağı görevlerinin uygulandığı tam bağlantı katmanıdır.

Evrişim katmanı, ESA tekniğinde öznitelik çıkarımının gerçekleştirildiği ana bileşendir. Bu katmanda kullanılan filtre ile ana görüntü arasında eleman bazında çarpma işlemi gerçekleştirilir. Şekil 3.6'da örnek üzerinde evrişim işlemi gösterilmiştir. Filtreler giriş görüntüsündeki tüm piksellerin üzerinden geçerek evrişim işlemini tamamlarlar.



Şekil 3.6. Evrişim işlemi.

Kullanılan filtreler vasıtasıyla özellik çıkarımı gerçekleştirilir ve böylece özellik haritası elde edilir. Farklı filtreler kullanılarak farklı ve önemli özelliklerin çıkartılması sağlanabilir.

Evrişim işlemi gerçekleştirilirken, ardışık iki filtre konumlandırılması arasındaki mesafe adım olarak adlandırılmaktadır. Genelde adım değeri bir olarak seçilmektedir. Ancak bazı durumlarda evrişim işleminin çıktısını aşağı örnekleme ve daha az hesaplama işlemi gerçekleştirmek için adım değeri iki ya da üzeri de seçilebilmektedir. Şekil 3.7'de adım örneği verilmiştir. Adım değeri Şekil 3.7(a)'da 1 olarak belirlenmişken, Şekil 3.7(b)'de 2 olarak belirlenmiştir.

5	2	7	1	9
1	3	9	2	1
1	0	5	0	1
2	0	0	7	9
3	5	7	6	4

(a)

5	2	7	1	9
1	3	9	2	1
1	0	5	0	1
2	0	0	7	9
3	5	7	6	4

(b)

Şekil 3.7. Adım değeri.

Evrişim işlemi sırasında görüntü üzerinde filtre, sol üst köşeden başlanarak önce yatay sırada ardından dikey sırada ilerletilir. Bu tarama işlemi sırasında filtrenin boyutuna ve özelliklerine bağlı olarak, görüntünün merkezindeki özelliklere odaklanılmaktadır. Ayrıca filtre boyutuna göre, giriş görüntüsü eleman bazında çarpım işleminden dolayı küçülmektedir. Görüntünün kenar ve köşelerindeki özelliklerinde özellik haritasında bulunması ve giriş boyutunun çıkış boyutu ile aynı olması için dolgulama işlemi kullanılmaktadır. Dolgulama işleminde genel olarak giriş görüntüsünün tüm kenarlarına sıfır değeri eklenerek giriş boyutunun değişmemesi ve kenardaki özelliklerinde özellik haritasına eklenmesi sağlanmaktadır. Şekil 3.8’de dolgulama işlemi bir örnek üzerinde verilmiştir.

0	0	0	0	0	0	0
0	5	2	7	1	9	0
0	1	3	9	2	1	0
0	1	0	5	0	1	0
0	2	0	0	7	9	0
0	3	5	7	6	4	0
0	0	0	0	0	0	0

Şekil 3.8. Dolgulama işlemi.

Görüntü boyutunun evrişim işlemi sonrasında hesaplanmasını sağlayan denklem, denklem (2.23)’te verilmiştir.

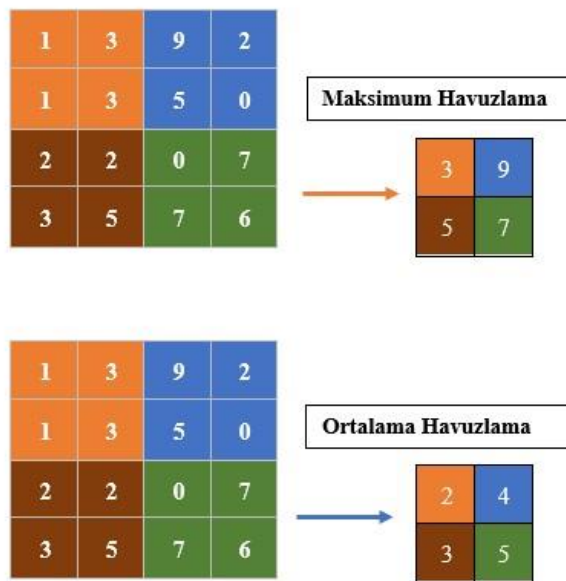
$$\frac{n+2p-f}{s} + 1 \quad (2.23)$$

Burada, n giriş görüntüsünün boyutunu, p dolgulama katmanını, f filtre boyutunu ve s adım değerini temsil etmektedir. Örneğin bu formüle göre, adım değeri 2, dolgulama değeri 1 ve filtre boyutu 3*3 olan, 5*5 boyutlu bir giriş görüntüsünün çıkış boyutu;

$$\frac{5+2*1-3}{2} + 1 = 3 \text{ (3*3) olmalıdır.}$$

Evrişim katmanındaki, filtre, adım ve dolgulama değerleri ESA'da kullanılan hiper parametre değerleridir.

ESA'da evrişim katmanından sonra genel olarak havuzlama katmanı gelmektedir. Bu katmanda maksimum havuzlama ve ortalama havuzlama gibi yöntemler kullanılmaktadır. Maksimum havuzlama işlemi, görüntünün belirli boyutta küçük parçalara bölünmesini ve her parçadaki en büyük değerin alınmasını içermektedir. Aynı şekilde ortalama havuzlama yönteminde de her parçadaki değerlerin ortalaması alınarak ortalama değerin belirlenmesini içermektedir. ESA'da uygulamalarının genelinde maksimum havuzlama işlemi kullanılmaktadır. Şekil 3.9'da ESA'nın havuzlama katmanında gerçekleştirilen işlemler verilmiştir. Burada maksimum havuzlama ve ortalama havuzlama işlemi gerçekleştirilmiştir.

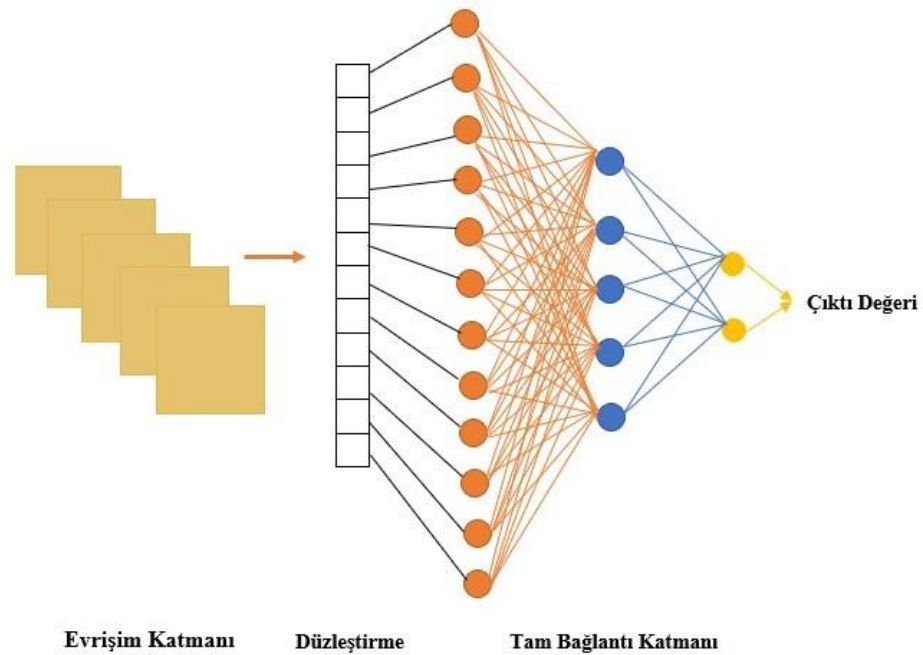


Şekil 3.9. Havuzlama katmanı.

Havuzlama katmanının uygulanmasının amacı, sonraki katmanlardaki karmaşıklığın azaltılması, önemli piksellerin vurgulanması ve aşırı öğrenmenin kontrol edilmesidir. Havuzlama katmanında hiper parametre yoktur. Bu sebeple genel olarak ESA’da evrişim ve havuzlama katmanı bir katman olarak değerlendirilmektedir.

Havuzlama katmanından sonra elde edilen özelliklere aktivasyon fonksiyonu uygulanmaktadır. Aktivasyon işlevi, bir giriş değerinden bir çıkış değerinin üretilmesini sağlar. Aktivasyon fonksiyonunun seçilmesi, modelin performansını önemli derecede etkilemektedir. ESA’da da standart sinir ağlarındaki; sigmoid, tanh, ReLU, leaky ReLU, Maxout gibi aktivasyon fonksiyonları kullanılmaktadır.

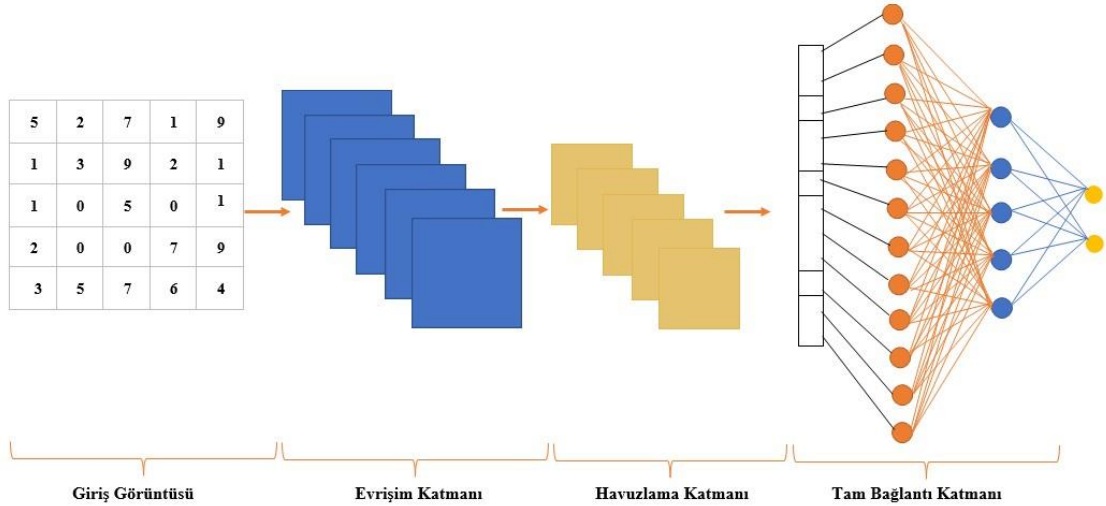
ESA’da tam bağlantı katmanı, en önemli ve en son katmandır. Bu katmanın amacı, ESA’nın ileri yönde beslenmesidir. Tam bağlantı katmanında, daha önceki gerçekleştirilen evrişim, havuzlama ve aktivasyon fonksiyonlarının çıktısının alınarak bir vektör olarak düzleştirilmesi gerçekleştirilir. Bu katmanda her düğüm, önceki ve sonraki düğümlere doğrudan bağlıdır. Bu sebeple bu katman, standart sinir ağına benzerlik göstermektedir. Ayrıca, bu katmanda aktivasyon fonksiyonu diğer katmanlarda kullanılan aktivasyon fonksiyonlarından biraz farklıdır ve bu katman, sınıf sayısı ile aynı sayıda çıktı düğümüne sahiptir. Şekil 3.10’da ESA’daki tam bağlantı katmanı verilmiştir.



Şekil 3.10. Tam bağlantı katmanı.

Standart sinir ağına benzer olarak, ESA'da de hiper parametreler mevcuttur. Her bir katmanın filtre sayısı, kullanılan filtrelerin boyutu, adım kaydırma ve dolgu değerleri, havuzlama boyutu ve türü gibi değerler ESA'da model başarısını etkileyen hiper parametrelerdir. Bu değerlerin boyutları genel olarak sezgisel ya da denenerek elde edilmiştir. Şekil 3.11'de tüm katmanlarıyla bir ESA modeli verilmiştir.

Tüm başarılı sinir ağı modellerinde olduğu gibi, ESA'da de geri yayılım algoritması kullanılmaktadır. Denetimli algoritma örneklerindeki gibi, giriş ve çıkış örnekleri kullanılarak hata değeri hesaplanır. Hata değeri elde ettiğimiz sonuç ile elde etmeyi umduğumuz sonuç arasındaki farktır. Ağ bu şekilde hata oranını azaltmak için güncellenen parametreler ile yeniden eğitilir. Böylece sistem eğitim verilerini öğrenene kadar bu şekilde işlemlere devam edilmektedir.



Şekil 3.11. ESA modeli.

Bu tez çalışma kapsamında, SSD-MobileNet ve Faster-RCNN olmak üzere iki farklı ESA modeli üzerinde çalışma yürütülmüştür. Bu kapsamda ESA modellerinden SSD, SSD-MobileNet, RCNN ve Faster-RCNN detaylı olarak verilecektir.

SSD, tek bir sinir ağı ile görüntülerdeki nesneleri algılamak için tasarlanmıştır [66]. Yöntem, sınırlayıcı kutuların, özellik haritalarına göre farklı en boy oranları ve ölçekler üzerinden bir dizi varsayılan kutuya ayrılmasına dayanmaktadır. Tahmin zamanında ağ, varsayılan kutulardaki, nesne kategorilerine göre puanlar üretir ve nesne şekline daha iyi uyum sağlaması için kutudaki ayarlamaları gerçekleştirir. Ek olarak, ağ, çeşitli

boyutlardaki nesneleri doğal olarak işlemek için farklı çözünürlüklere sahip birden çok özellik haritasından gelen tahminleri birleştirir. SSD, özelliklerin yeniden örneklenmesi gibi aşamaları ortadan kaldırdığı ve tüm hesaplamayı tek bir ağda gerçekleştirdiği için uygulanması açısından daha basittir. Diğer tek aşamalı yöntemlerle karşılaştırıldığında, SSD, daha küçük bir giriş görüntüsü boyutunda bile çok daha iyi bir doğruluğa sahiptir.

Mobilenet, mobil ve gömülü görüş uygulamaları için iyi organize edilmiş model sınıfıdır. Bu model sınıfı, derin sinir ağları oluşturmak için derinlikle ayrılabilir basitleştirilmiş bir mimariye dayalıdır. Standart evrişimi, derin evrişimi ve nokta evrişimi olarak bilinen 1×1 evrişim olarak ayrıştırır. Derin evrişim, her giriş kanalına bir filtre uygular. Daha sonra nokta evrişimi, derinlik evrişiminin çıktılarını birleştirmek için 1×1 evrişim oluşturur. Standart evrişim, girdileri yeni bir çıktı kümesinde hem filtrelemek hem de birleştirmek için bir adıma sahiptir. Ancak derinlikle ayrılabilir evrişim, filtreleme için ayrı bir katman ve birleştirme için ayrı bir katman olmak üzere standart evrişimi iki katmana ayırır. Bu ayırma işlemi, hesaplamayı ve model boyutunu büyük ölçüde azaltır. Mobilenet-SSD ise, Mobilenet ile SSD uygulamalarının birleştirilmesini içermektedir.

RCNN, iki aşamalı bir ESA modelidir [70]. R-CNN dedektörü dört modülden oluşmaktadır. İlk modülde, seçici arama algoritması ile yaklaşık 2000 kategoriden bağımsız bölge önerileri üretilir. İkinci modülde, ESA kullanılarak, her bölge önerisinden 4096 boyutlu bir özellik vektörü çıkartılır. Tam bağlı katmanda, sabit uzunlukta giriş vektörlerine ihtiyaç duyulduğundan, bölge önerisi özellikleri aynı boyuta sahip olmalıdır. ESA'da giriş boyutu olarak 227×227 piksel kullanılır. Özellik çıkarma ağı, beş evrişim katmanından ve iki tam bağlantılı katmandan oluşur. Üçüncü modülde, nesneleri tek bir görüntüde sınıflandırmak için sınıfa özgü doğrusal destek vektör makinaları (DVM) kullanılmaktadır. DVM, büyük veri kümelerinde ön eğitim ve hızlı yakınsama özelliklerinden dolayı iyi bir eğitim yöntemidir. Son modül ise, sınırlayıcı kutu tahmini için oluşturulmuştur. RCNN, nesne algılamada başarılı sonuçlar elde etmesine rağmen, dezavantajları da vardır. İlk olarak, her görüntü, depolama alanını çok fazla kullanan yaklaşık 2000 bölge önerisiyle parçalanmaktadır. İkinci olarak ise, her bölge önerisi özelliğinin ayrı olarak çıkarılması, oldukça fazla bilgi işlem gücü gerektirmektedir. Bu dezavantajların giderilmesi için farklı çalışmalar gerçekleştirilmiştir. Bu çalışmalardan biri de RCNN'in daha hızlı olmasını sağlayan Faster-RCNN yöntemidir.

Faster-RCNN, iki aşamalı ESA mimarisin hız kazandırmıştır. Faster-RCNN, seçici arama algoritmasının görüntüdeki tüm bölge önerilerini araması ve bunları özellik haritalarında eşleştirmesi fikri yerine, çok çeşitli ölçekler ve en boy oranları ile bölge önerilerini verimli bir şekilde tahmin etmek için tamamen evrişimli bir ağ olan bölge öneri ağı (Regional Proposal Networks:RPN) kullanmaktadır. RPN, teklif sayısını azaltır ve bölgesel tekliflerin kalitesini iyileştirir, böylece nesne algılama ağının konum doğruluğu ve hızını oldukça artırır.

Deneyisel çalışmalarda kullanılan Mobilenet-SSD ve Faster-RCNN evrişim ağları mimarileri performans metrikleri açısından 4. bölümde detaylı olarak karşılaştırılmıştır.

3.3. Elmaların Sayılması İşlem Adımları

Elma bahçesinde otonom olarak uçan robotik sistem ile rekoltenin belirlenmesi amacıyla gerçekleştirilen çalışmada, bu bölüme kadar URS'nin üretilmesi, otonom uçuşu, görüntü işleme ve yapay zeka teknikleri hakkında bilgiler verilmiştir. Bu bölümde ise artık geliştirilen yapay zeka modeli ve uygulanan elma sayma yöntemlerinden bahsedilecektir. Derin öğrenme tabanlı elma sayma çalışmasının genel uygulama metodolojisi Şekil 3.12'de belirtilmiştir.



Şekil 3.12. Derin öğrenme yöntemiyle nesne sayma genel metodolojisi.

3.3.1. Veri Setinin Oluşturulması

Başarılı sonuçlar üreten bir nesne algılama modelinin oluşturulması için titizlikle yapılan veri toplama süreci son derece önemlidir. Süreci otomatikleştirmenin birçok yolu olmasına rağmen, geliştirilecek bu özgün sinir ağ modelinde eğitim veri setinin kalitesini sağlamak için veri toplama robotik sistem üzerinden manuel olarak gerçekleştirilmiştir. Gerçekleştirilen deneyisel uygulamanın şartlarına göre GoPro kamera ile 5184x3888 piksel boyutlarında 1000 adet Şekil 3.13'teki gibi kırmızı tip elma görüntüsü elde edilmiştir.



Şekil 3.13. Veri seti için çekilen kırmızı elma görüntüsü örnekleri.

Gerçekleştirilen bu deneysel çalışma için görüntünün değerlendirilip o görüntüde elma olup olmadığını tahmin eden bir model tam anlamıyla amaca uygun olmayacaktır. Bu sebeple sinir ağ modeline görüntünün olduğu gibi girdi olarak tanıtılması yeterli değildir. Bu çalışmada görüntü üzerindeki bütün elmaların belirlenip sayılması amaçlanmaktadır. Bu sebeple, elde edilen görüntülerin etiketlenmesi gerekmektedir. Görüntü etiketleme, görüntü üzerindeki nesnelerin sınırlayıcı kutular içerisine alınıp bir isim ile etiketlenmesi işlemini kapsamaktadır. Daha sonra bu etiketlenen nesne görüntüleri XML (Extensible Markup Language) dosyasına kaydeden bir açık kaynaklı python tabanlı uygulama olan LabelImg yazılımı kullanılarak geliştirilen ağ modeli için veri setine dönüştürülmüştür. Şekil 3.14'te, resimlerin etiketlenmesi işlemi gösterilmektedir.



Şekil 3.14. Etiketleme işlemi için kullanılan LabelImg programının arayüzü.

LabelImg programından oluşturulan çıktı, elle etiketlenen görüntülerin sınırlayıcı kutu koordinatlarını ve etiket isimlerini içeren bir XML dosyasıdır. Ancak sinir ağı modelleri,

XML dosyalarını ayrıştırılmaz ve TF-Record adı verilen bir formata dönüştürülmesi gerekmektedir. Dolayısıyla bir sonraki adımda, XML dosyasının CSV çıktısına dönüştürülmesi ve ardından CSV çıktısının TF-Records'a son olarak dönüştürülmesi gerekmektedir.

Bu deneysel çalışma için geliştirilen ESA modelinin eğitiminde; 800x600 çözünürlüğe dönüştürülmüş 100 adet seçilmiş resim üzerinden 3548'i eğitim için 729'u test için toplam 4277 etiketli elma görüntüsü kullanılmıştır.

3.3.2. Evrişimli Sinir Ağ (ESA) Modeli Eğitimi

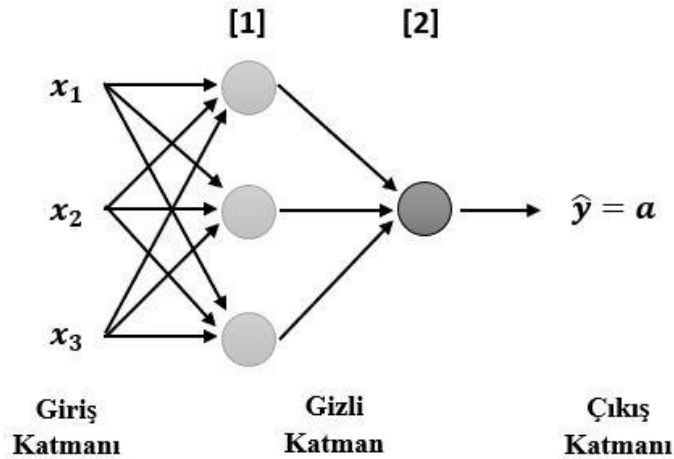
Derin öğrenme metotları içerisinde görüntü üzerinden tahminleme işlemi için en uygun metot olan ESA modeli için farklı özellikte mimariler geliştirilmiştir. Bu mimariler arasından robotik uygulamalar için küçük boyutu ve hızı ile avantaj sağlayan SSD-MobileNet mimarisi ve doğruluk oranı yüksek olan Faster R-CNN mimarileri bu çalışma kapsamında kullanılmıştır.

İncelenen birçok çalışmada, farklı mimarilerdeki ESA modellerinin kullanıma açık MS COCO (Microsoft Common Objects in Context) isimli veri setleriyle eğitilmiş hazır ağırlık dosyalarının kullanıldığı gözlenmiştir. Bu çalışmada kullanılan TensorFlow kütüphanesi için de bu şekilde oluşturulmuş çok sayıda hazır ESA model dosyaları paylaşılmış olup araştırmacılar bu modeller ile yaptıkları çalışma sonuçlarını kıyaslama imkanı bulmaktadırlar. Fakat bu tez çalışmasında, URS üzerinden kısıtlı koşullarda ve dinamik görüntüler üzerinde gerçek zamanlı bir uygulama gerçekleştirilmesinden dolayı bu hazır modellerin yetersiz kalacağı düşünülmüştür. Bu sebeple, çalışmaya uygun veri seti özgün olarak oluşturulmuş olup sinir ağ modeli bu özgün veri setiyle eğitilmiştir ve hazır MS COCO veri setiyle de eğitilen ikinci model ile hem SSD-MobileNet mimarisi için hem de Faster R-CNN mimarisi için sonuçlar kıyaslanmıştır. Şekil 3.15, deneysel çalışmada Faster R-CNN mimarisi kullanılarak MS COCO veri setiyle eğitilen (soldaki) ve özgün veri setiyle eğitilen (sağdaki) sinir ağ modeli ile gerçekleştirilen uygulamaya ait sonuçları göstermektedir. Daha detaylı kıyaslamalar sonuçlar bölümünde verilmiştir.



Şekil 3.15. COCO veri seti (soldaki) ve özgün veri setiyle (sağdaki) eğitilen sinir ağ model ile gerçekleştirilen elma tespiti.

Derin sinir ağ modeli eğitimi için mimari seçimi ve uygun veri seti kullanımı ile birlikte eğitim için kullanılacak hiperparametrelerin de doğru belirlenmesi çok önemli bir konudur. Bu hiperparametrelerin en önemlileri şunlardır: katman sayısı, öğrenme oranı, epoch sayısı, batch boyutu, gizli katmandaki nöron sayısı, momentum, aktivasyon fonksiyonu. Bu parametreler, sinir ağ modelinin yapısını geliştiren ağırlık ve bias parametrelerinin güncellenmesinde önemli rol oynamaktadır ve sinir ağ yapısının başarı yüzdesini artırmak için doğru belirlenmelidir. Şekil 3.16’da basit yapıli örnek bir sinir ağ yapısı gösterilmiştir. Buradaki ağırlık ve bias değerleri kullanılarak denklem (2.28)’e göre kayıp fonksiyonu oluşturulmaktadır. Daha sonra geriye doğru türev alma işlemi ile kayıp değerini düşürmek için parametrelerin güncellenmesi gerçekleştirilmektedir. Bu işlem belirlenen iterasyon boyunca devam etmektedir.



Şekil 3.16. Basit yapıli örnek bir sinir ağ modeli.

$$z^{[1]} = W^{[1]}x + b^{[1]} \quad (2.24)$$

$$a^{[1]} = \sigma(z^{[1]}) \quad (2.25)$$

$$z^{[2]} = W^{[2]}a^{[1]} + b^{[2]} \quad (2.26)$$

$$a^{[2]} = \sigma(z^{[2]}) \quad (2.27)$$

$$L(a^{[2]}, y) \quad (2.28)$$

Burada W: ağırlık parametresi, x: bağımsız değişken (girdi), b: bias değeri, z: girdiye ait skoru belirlemede kullanılan bağımlı değişken, L: Loss (kayıp) fonksiyonunu vermektedir.

Öğrenme oranı (α): Sinir ağ modelinin parametrelerini güncelleme hızını tanımlamaktadır. Güncelleme, geriye doğru türev alınan türevin öğrenme oranı ile çarpılmasına dayanmaktadır. Düşük öğrenme oranı, daha sorunsuz bir yaklaşım gibi görünse de öğrenme süresini çok yavaşlatmaktadır. Öğrenme oranı yüksek seçildiğinde de öğrenme hızı artar ancak bu sefer de istenilen sonuca yakınsama mümkün olmayabilir. Öğrenme oranı sabit bir sayı olarak belirlenebileceği gibi değişken olarak da belirlenebilir. En ideal yaklaşım başlangıçta yüksek, eğitim tur sayısı arttıkça azalan bir değer olarak belirlenmesidir. Başlangıç değerinin belirlenmesi için optimizasyon çalışması yapılmıştır.

Epoch sayısı (η): Sinir ağ modeli eğitilirken veriler belirli sayıda parçalar halinde eğitime dahil edilmektedir. Belirlenen veri parçası eğitilir, model test edilir ve geriye yayılım ile parametreler güncellenir. Daha sonra yeni veri parçası ile model tekrar eğitilir ve parametreler tekrar güncellenir. Bu işlemler her bir eğitim adımı için tekrarlanarak en uygun parametrelerin belirlenmesi hedeflenmektedir. Bu eğitim adımlarının her biri “epoch” olarak isimlendirilir. Başlangıçta epoch değeri ne kadar belirlenmiş ise model o kadar sayıda adımda eğitimini tamamlamaktadır.

Batch Boyutu (B): Veri setinde bulunan tüm verileri aynı anda işleme dahil etmek zaman ve bellek açısından maliyetli bir işlemdir. Bu çalışma kapsamında da kullanılan sistem bunun için kesinlikle uygun değildir. Çünkü eğitimin her bir adımında gerçekleştirilen hesaplamalar çok zaman almakta, işlemci ve bellek üzerinde önemli bir kullanım alanına

ihtiyaç duymaktadır. Bu sebeple yüzlerce veriyi aynı anda işleme almak çok mümkün değildir. Genellikle bu değer, sistemin performansına CPU ve GPU işlemci kullanımına bağlı olarak 2, 4, 8, 16, 32 değerlerinden birisi olarak tercih edilmektedir. Bu çalışma kapsamında batch boyutu 4 olarak belirlenmiştir. Daha büyük seçilebilen batch boyutu doğrusallaştırmayı azaltacak ve hata oranı daha düşük olabilecektir.

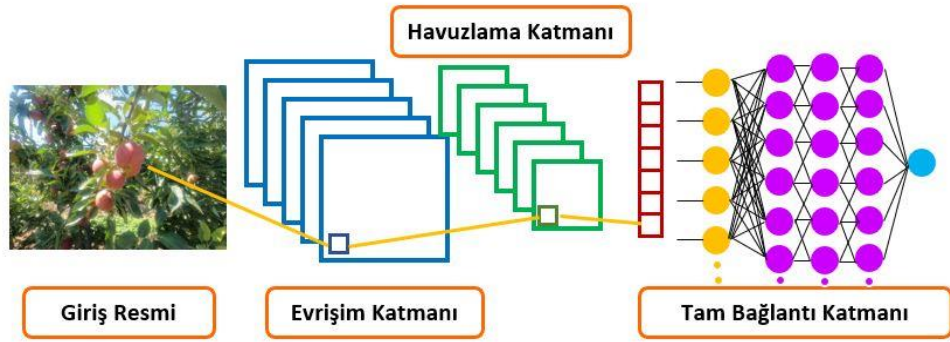
Katman sayısı: Giriş katmanı, çıkış katmanı ve gizli katmanı kapsamaktadır. Giriş katmanı giriş verilerini, çıkış katmanı da çıktı verilerini ifade ederken gizli katman matematiksel işlemlerin yapıldığı katmandır. Derin öğrenmedeki derinliği burada katman sayısının ikiden fazla olmasını ifade etmektedir. Katman sayısının ve katmanlardaki nöron sayılarının belirlenmesi model açısından son derece önemlidir. Bu sayılar belirlenirken sinir ağ modeli bütün olarak ele alınmalıdır.

Momentum (m): Eğitimde önceki adımlardaki değerler ile bir sonraki adım üzerine yönlendirme yapılmasını sağlamaktadır. Eğitim esnasında hata oranının daha kararlı bir şekilde azalmasına yardımcı olabilmektedir.

Aktivasyon fonksiyonu ($\sigma(x)$): Eğitim için kullanılan optimizasyon algoritması belirlenir. Burada hangi algoritma eğitilecek model için daha hızlı ve başarılı sonuç veriyorsa tercih edilir. Bu tez çalışmasında ReLU aktivasyon fonksiyonu tercih edilmiştir.

3.3.3. Nesne Tespiti

Bu tez çalışmasında, özgün olarak oluşturulan veri seti ile bir derin öğrenme modeli geliştirilmiştir. Bu modelin nesne tespiti gerçekleştirebilmesi için Python dilinde bir nesne tespit yazılımı geliştirilmiştir. Bu yazılım geliştirilirken TensorFlow kütüphanesinden ve nesne tanıma programlama arayüzünden faydalanılmıştır. Bu arayüzü kullanmak için gerekli paketler yüklenmiştir. Belirlenen optimum parametrelere göre SSD-MobileNet ve Faster R-CNN mimarileri dikkate alınarak özgün veri setiyle yeni sinir ağ modelleri eğitilmiştir. Sonuç olarak da geliştirilen farklı koşullardaki sinir ağ modellerinin başarı performansları karşılaştırmalı olarak incelenmiştir. Şekil 2.49'da bir sinir ağ modeli ile nesne tespiti aşamaları şematik olarak gösterilmiştir.



Şekil 3.17. Nesne tespiti için kullanılan evrişimli sinir ağı yapısı.

SSD model mimarisi kullanılarak tek aşamalı ve Faster-RCNN model mimarisiyle çift aşamalı olarak yapılan örnek bir nesne algılama işlemi Şekil 2.50’de gösterilmiştir.



Şekil 3.18. SSD (soldaki) ve F-RCNN (sağdaki) eğitilen sinir ağı model ile gerçekleştirilen elma tespiti.

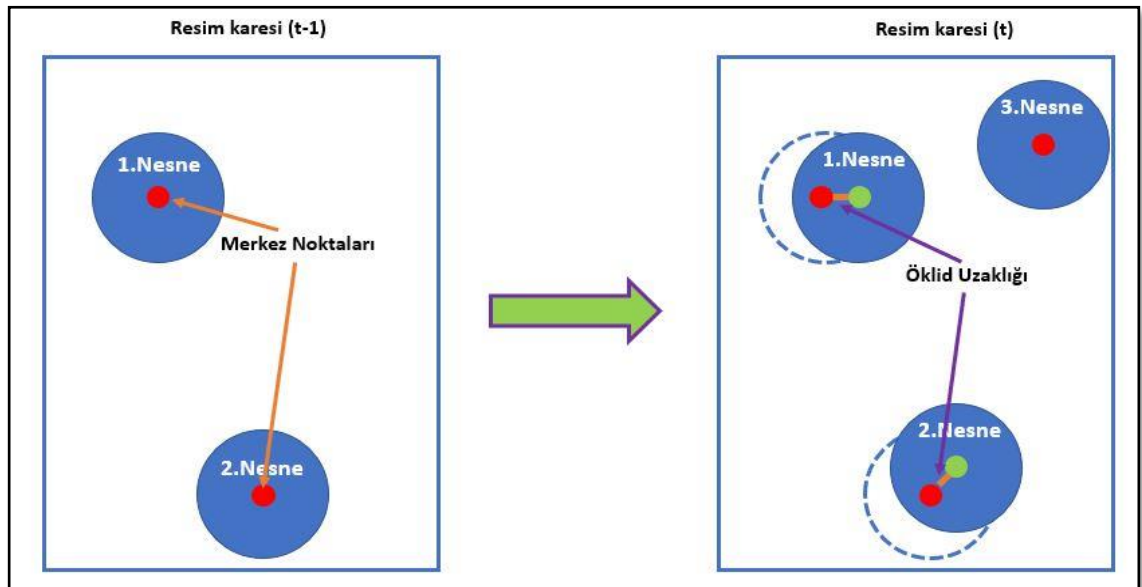
3.3.4. Nesne Takibi ve Sayımı Metodu

Meyve bahçesinin rekoltesinin belirlenmesinde elmaların tespit edilmesinden sonra kümülatif olarak doğru sayılması en önemli problemlerden biridir. Aynı zamanda bu çalışmada uygulanan metotların sonucudur. Buradaki temel problem; fotoğraf karelerinden oluşan video kaydı üzerinde elmalar sayılırken aynı elmanın birden fazla sayılıyor olmasıdır. Bunun üstesinden gelmek için farklı yöntemler denenebilir. Bu çalışmada, değişen karelerde aynı elmaları tekrar tekrar saymamak ve fotoğraf karesine yeni eklenen elmaları saymak için Opencv kütüphanesinden nesne takip yöntemi kullanılmıştır.

Şekil 3.19’da genel çalışma mantığı şematik olarak gösterilen bu yöntemde; ilk olarak nesneler (t-1) zamanında algılanır ve sınırlayıcı çerçeveye yerleştirilerek kimlik atanır. Yine (t-1) zamanında, oluşturulan sınırlayıcı çerçevenin ağırlık merkezinin konumu tespit edilir. Daha sonra sınırlayıcı çerçevenin (t) zamanındaki ağırlık merkezinin konumu belirlenir. Son olarak, (t-1) ve (t) zamanlarında hesaplanan ağırlık merkezi konumları arasındaki öklid mesafeleri denklem (2.29) kullanılarak hesaplanır. Geliştirilen nesne sayma yazılımında belirlenen eşik değerin altında bir değer hesaplanırsa tespit edilen nesneler aynı nesne olarak kabul edilir ve daha önce atanan kimlik aynen kalır. Fakat hesaplanan öklid mesafesi eşik değerin üzerinde olursa (t) anında tespit edilen nesne yeni bir nesne olarak kabul edilir ve yeni bir kimlik atanır.

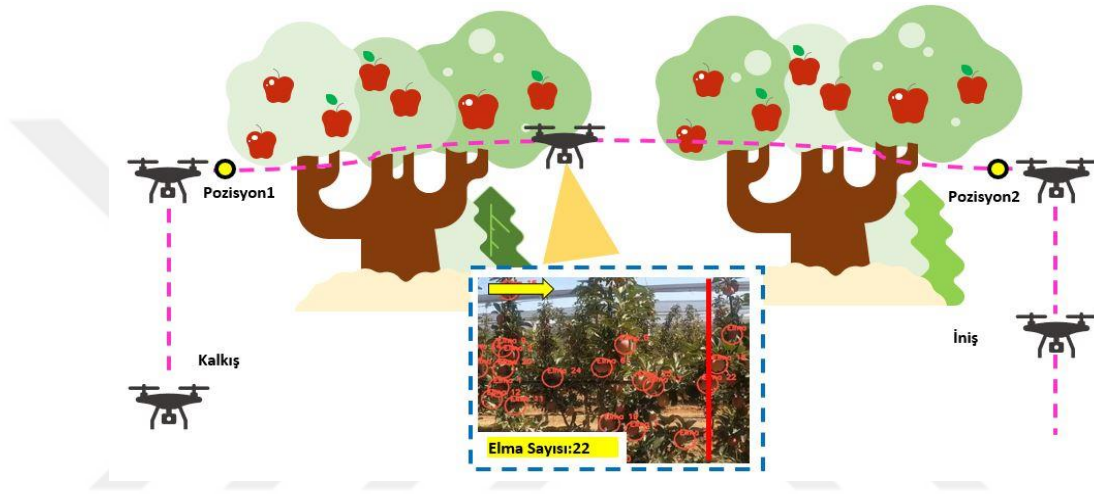
$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (2.29)$$

Bu denklemde; d: algılanan nesnenin (t) ve (t-1) zamanındaki ağırlık merkezlerinin birbirlerine öklid mesafesidir, x_2 : (t) anında nesnenin x koordinat eksenine göre konumudur, x_1 : (t-1) anında nesnenin x koordinat eksenine göre konumudur, y_2 : (t) anında nesnenin y koordinat eksenine göre konumudur, y_1 : (t-1) anında nesnenin y koordinat eksenine göre konumudur.



Şekil 3.19. Uygulanan nesne takibi yönteminin şematik gösterimi.

Bu nesne takip yöntemiyle aynı nesne tekrar tekrar sayılmadan elmaların rekoltesinin tespiti noktasında doğru bir sayma işlemi gerçekleştirmek mümkün olmuştur. Son olarak da ayrıık olarak sayılan elmaların URS'nin takip ettiği yörünge boyunca kümülatif olarak sayılması ve toplam elma sayısının tespit edilmesi için otopark otomasyon sistemine benzer bir metot uygulanmıştır. Şekil 3.20'de URS'nin otonom olarak izlediği yörünge şematik olarak gösterilmiştir ve kameradan elde edilen görüntü üzerinde uygulanan işlem görsel olarak sunulmuştur.



Şekil 3.20. Deney sahasında elma sayma uygulamasının şematik gösterimi.

Bu şematik gösterimde de anlatıldığı gibi kameradan elde edilen görüntü soldan sağa doğru akmaktadır. Bu görüntü karesi üzerine belirlenen koordinata kırmızı renkli sanal dikey bir çizgi yerleştirilmiştir. Yapay sinir ağ modeli çalıştırılarak tespit edilen ve takip işlemi uygulanan elmalar bu sanal çizginin solundan sağına geçtiği tespit edildiği anda sayılmaktadır. Kameradan elde edilen görüntüler kesildiğinde program kapanmaktadır ve bir değişken içerisinde tutulan son değer sayılan toplam elma sayısını vermektedir.

4. BÖLÜM

BULGULAR

Bu bölümde, geliştirilen URS ile gerçekleştirilen simülasyonlar ve deneysel çalışmalar anlatılacak olup sonuçlar incelenecektir. Bu çalışmalar, temelde deneysel olarak gerçekleştirilen derin öğrenme ile elmaların tespiti ve sayılmasına odaklanmış olsa da, robotun konum tespiti ve yörünge takibi, PID kontrol parametreleri K_p , K_i ve K_d 'nin en optimum değerde belirlenmesi ve otonom uçuş ile ilgili testleri de kapsamaktadır. Robotik sistem ile ilgili uygulamalar, bu bölümde simülasyon çalışmaları ve deneysel çalışmalar olmak üzere iki ayrı başlık altında anlatılmıştır. Deneysel çalışmaların son bölümünde “Derin Öğrenme ile Elma Tespiti Çalışmaları”, “Elma Sayma Çalışmaları” ve “Hastalıklı Elma Tespiti” başlıkları altında uygulanan evrişimli sinir ağ modellerinin, farklı durumlardaki performanslarının incelenmesi ve önerilen yeni elma sayma yaklaşımlarının incelenmesi gerçekleştirilmiştir. Son olarak da elmaların sayılmasında uygulanan yöntemler karşılaştırılmıştır.

4.1. Simülasyon Çalışmaları

Teorik çalışmalar ve hesaplamalar sonucunda belirlenen parametreler deneysel çalışmada kullanılmadan önce simülasyon ortamında test edilmiştir. Bu bölüm üç ayrı aşamada gerçekleştirilmiştir. Bu aşamalar sırasıyla; “Otonom Uçuş Yörünge Takibi için Simülasyon Çalışmaları”, “Kontrol Parametrelerinin Optimizasyonu için Simülasyon Çalışmaları” ve “Konumlandırma ve Haritalama için Simülasyon Çalışmaları” olmak üzere farklı başlıklar altında ayrıntılı olarak anlatılmıştır.

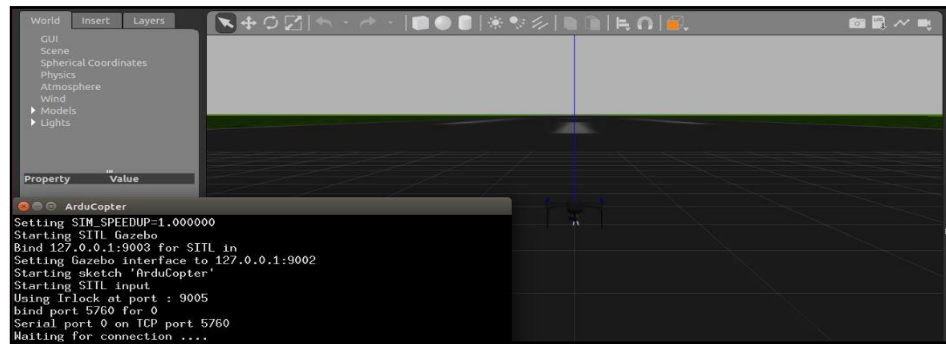
4.1.1. Otonom Uçuş Yörünge Takibi için Simülasyon Çalışmaları

Robotik sistemin tasarım süreci devam ederken otonom olarak planlanan görevi yerine getirmesi ile ilgili kodlama çalışmaları gerçekleştirilmiştir. Çalışmalarda; DroneKit

kütüphanesi kullanılarak Python dilinde yazılan kod parçaları kullanılmıştır. Tamamlanan kod parçalarının test edilmesi amacıyla iki boyutlu ve üç boyutlu simülasyon ortamları hazırlanmıştır. Bunun için harita üzerinde robotik sistemin izlediği yörüngeyi görselleştiren Dronekit kütüphanesindeki paketlerden ve uçuş planlama programı olan Mission Planner programından faydalanılmıştır. Ayrıca Gazebo ortamında da üç boyutlu olarak otonom uçuş sırasında robotun hareketleri gözlenmiştir. Şekil 4.1’de URS’nin otonom olarak harita üzerinde uçuş simülasyonunda izlediği yörünge ve Şekil 4.2’de eş zamanlı olarak URS’nin hareketinin izlendiği Gazebo ortamının görüntüsü verilmiştir. Bu çalışmalarda; Gazebo ortamında, geliştirilen sisteme tasarım olarak benzer özelliklerde olan standart dört rotorlu bir hava aracı modeli kullanılmıştır.



Şekil 4.1. Simülasyon ekranında URS’nin otonom yörünge takibi.



Şekil 4.2. URS’nin otonom uçuşunu üç boyutlu olarak görselleştiren Gazebo Programı.

4.1.2. Kontrol Parametrelerinin Optimizasyonu için Simülasyon Çalışmaları

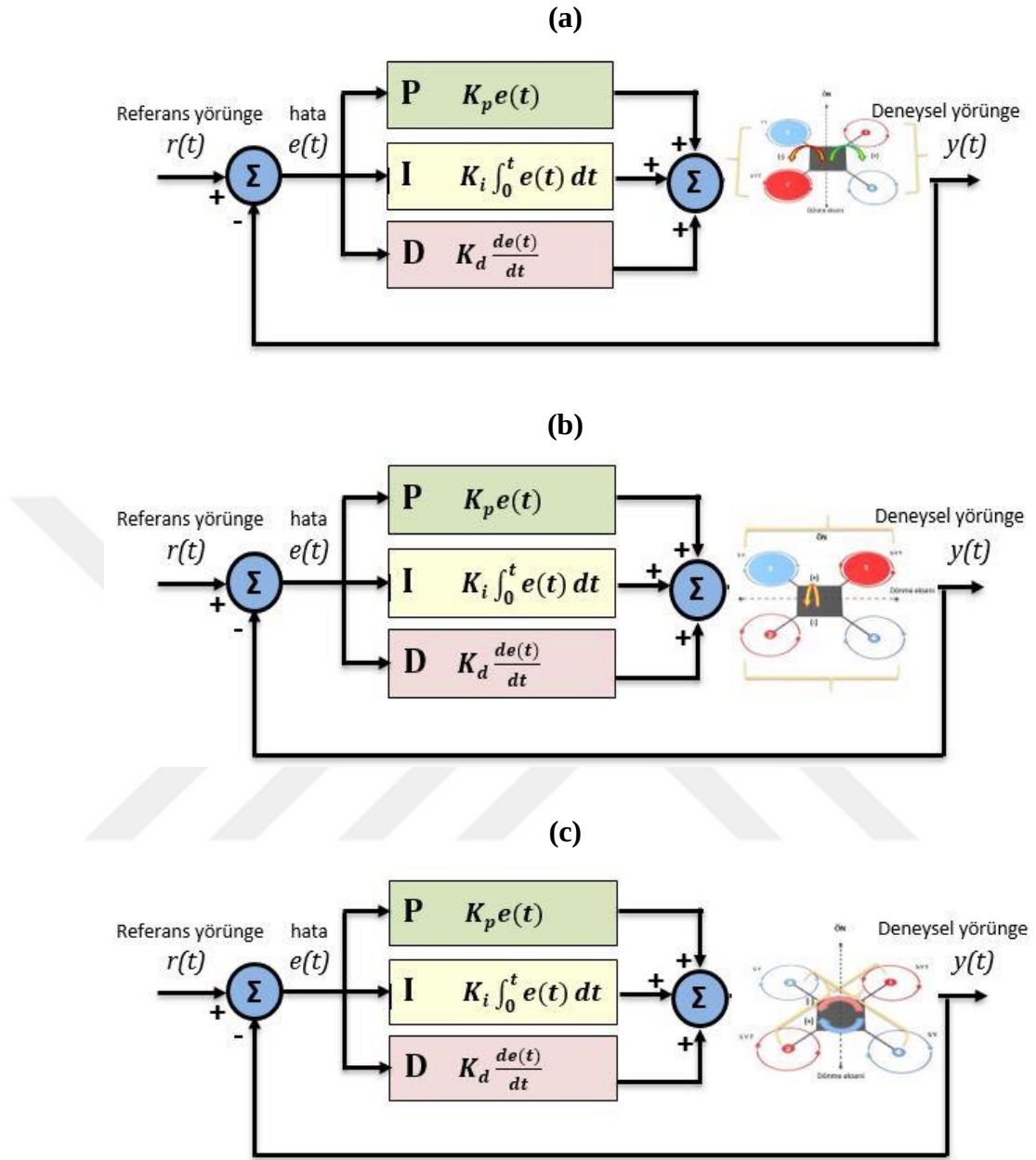
Hava araçları, belirli modlara göre uçuşlarını gerçekleştirmektedirler. Bu modlar kısaca Tablo 4.1’de açıklanmıştır. Gerçekleştirilen çalışmada uçan robotik sistemin tam otonom olarak görevi yerine getirebilmesi için Auto ve Guided modları kullanılmıştır. Uçuş

programının kontrolcüye yüklendiği durumlarda Auto mod, Dronekit ile yazılan program üzerinden harici olarak yönetildiği durumlarda ise Guided mod kullanılmıştır.

Tablo 4.1. Uçuş Modlarının kısa açıklamaları.

Uçuş Modu	Açıklaması
STABILIZE	Uçuş pilot kontrolünde gerçekleşmektedir.
AUTO	Hava araç, belirlenen konum ve irtifada istenilen görevi otonom olarak gerçekleştirir.
GUIDED	MAVLink protokolüyle harici bir kontrolcü üzerinden kontrol gerçekleştirilir.
LOITER	Uçuş pilot kontrolünde gerçekleşmektedir. Hava aracı mevcut konumu, rotayı ve irtifayı korur.
ALTITUDE	Uçuş pilot kontrolünde gerçekleşmektedir. Hava aracı irtifasını korur.
POSHOLD	Uçuş pilot kontrolünde gerçekleşmektedir. Hava aracı pozisyonunu korur.

Tercih edilen her iki modda da, hava aracının otonom olarak kararlı bir şekilde belirlenen yörüngeyi takip edebilmesi için bir kontrolcü tarafından konum hatalarının sürekli olarak denetlenerek en aza indirilmesi gerekmektedir. Bu sistem; yuvarlanma, yunuslama ve sapma hareketleri için üç farklı PID kontrol yapısı ile kontrol edilmektedir. Şekil 4.3'te URS'nin PID kontrol modeli sırasıyla yuvarlanma, yunuslama ve sapma hareketleri için şematik olarak gösterilmiştir. Uçuş öncesinde her bir kontrol modeli için K_p , K_i ve K_d parametrelerinin ayarlanması gerekmektedir. Bu ayarlama tecrübeye dayalı olarak deneme yanılma yöntemi ile yapılabileceği gibi bu tip hava araçları için geliştirilen bir mod olan Autotune modunda otonom olarak da ayarlanabilmektedir. Uçuş için uygun ve yeterli bir alan olması durumunda otomatik olarak PID kontrolcü kazanç parametrelerinin ayarlanması daha etkili bir sonuç vermektedir. Çünkü bu modda, hava aracı belirlenen bir algoritmaya göre arıların dansına benzer şekilde geniş alana yayılı bir yörüngede otonom olarak hareket etmektedir. Bu hareketi boyunca da K_p , K_i ve K_d parametrelerinin ayarlamasını en optimum şekilde gerçekleştirmektedir. Ayrıca, çoğunlukla bu yöntem ile parametrelerin belirlenmesi ortamdaki bütün bozucu etkenler dikkate alınarak deneysel olarak belirlendiği için de iyi sonuçlar vermektedir.



Şekil 4.3. (a) Yuvarlanma hareketi için, (b) yunuslama hareketi için, (c) sapma hareketi için gösterilen genel kontrol şeması.

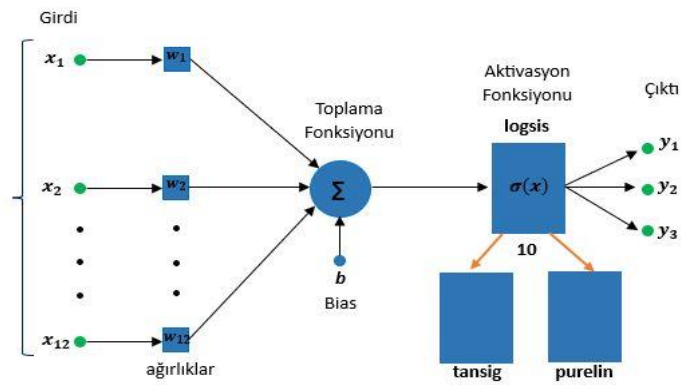
Fakat bu çalışmada URS'nin ağaçların arasında dar ve üzeri örtülü bir ortamda hareketini gerçekleştirecek olması sebebiyle Autotune modunun burada uygulanabilmesini mümkün kılmamıştır. Bu sebeple K_p , K_i ve K_d parametrelerinin belirlenebilmesi için bir yapay sinir ağ (YSA) modeli geliştirilmiştir. İleri beslemeli geri yayılım sinir ağı (FFBPN: Feed-Forward Backpropagation Network) ve radyal esaslı sinir ağı (RBNN: Radial Basis Neural Network) tipi iki farklı YSA modeli ile eğitimler, farklı algoritmalar ve aktivasyon fonksiyonları kullanılarak performans karşılaştırmalı olarak gerçekleştirilmiştir. YSA

modelinde kullanılan eğitim parametreleri Tablo 4.2’de verilmiştir. Kp, Ki ve Kd parametrelerinin belirlenmesi amacıyla eğitilen YSA modellerinin şematik gösterimi sırasıyla FFBN ve RBNN modelleri için Şekil 4.4’te sunulmuştur. FFBN şematik gösteriminde örnek olarak logsig aktivasyon fonksiyonlu model kullanılmış olup, tansig ve purelin fonksiyonlarının şematik gösterimleri de eklenmiştir.

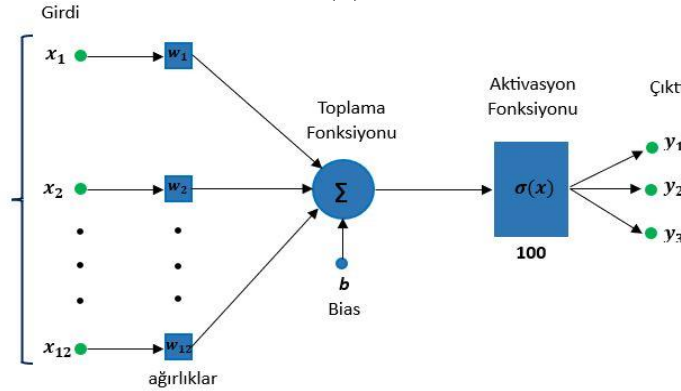
Tablo 4.2. YSA modelleri için eğitim parametreleri.

Performans Fonksiyonu	MSE
Maksimum İterasyon Sayısı (Epochs)	1000
Gizli Katman Nöron Sayısı	10
Minimum Gradient	1e-06
Sonuç Gösterme İterasyon Sayısı	25

(a)



(b)

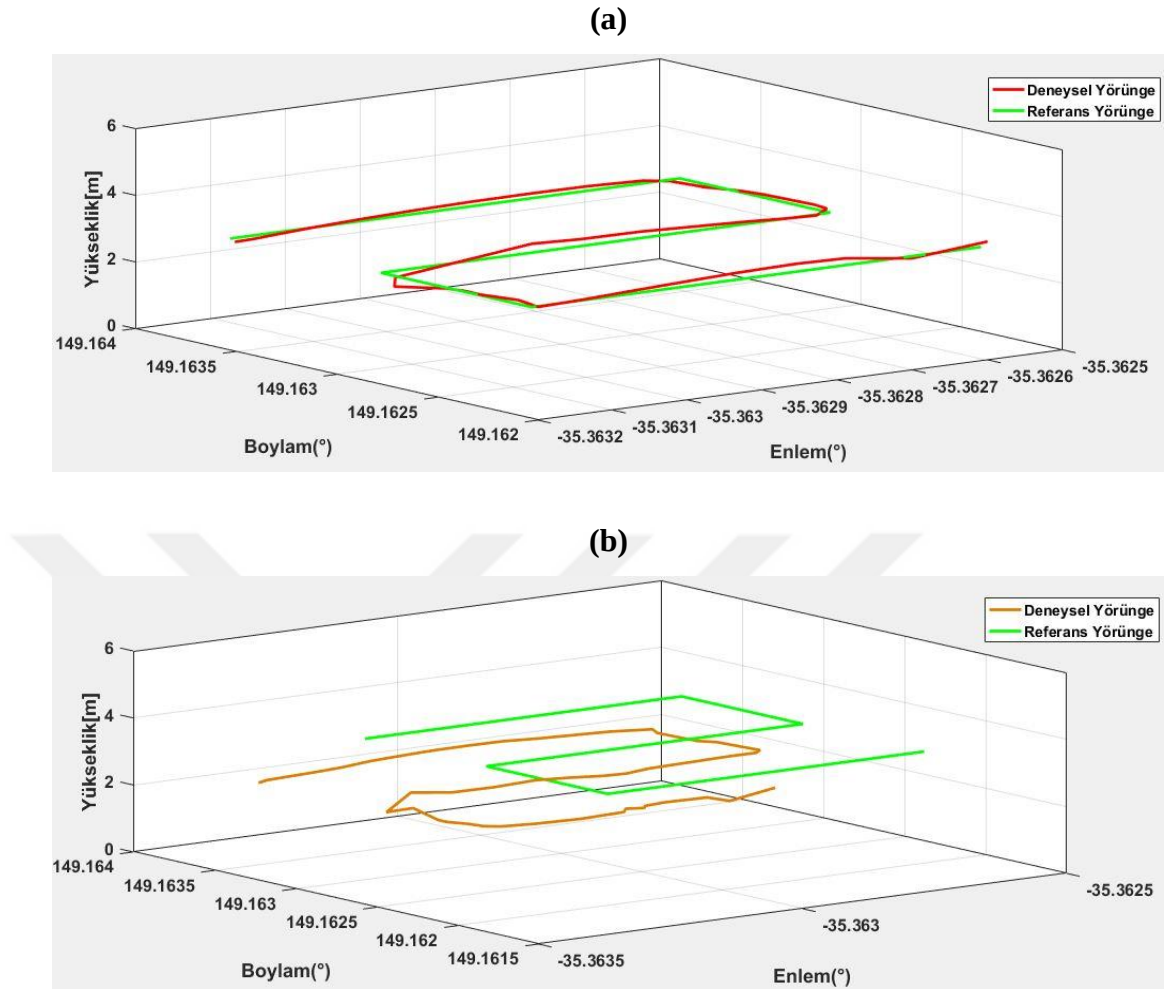


Şekil 4.4. Kp, Ki ve Kd parametrelerinin belirlenmesi için eğitilen YSA modelleri: (a) İleri Beslemeli Geri Yayılım Sinir Ağı (FFBN), (b) Radyal Esaslı Sinir Ağı (RBNN).

Simülasyon ortamında, deneysel çalışmanın gerçekleştirileceği ağaç dizilimine ait sıra boyu referans alınarak bir yörünge belirlenmiştir. Bu yörünge yuvarlanma ve yunuslama kontrol modelleri için Tablo 4.3'te bir bölümü gösterilen ayrı ayrı belirlenen 100'er farklı K_p , K_i ve K_d parametrelerinden oluşan veri setleri ile otonom olarak takip edilmiştir. Gerçekleştirilen simülasyon uçuşlarında sapma kontrol modeli için bir optimizasyona ihtiyaç olmadığı görülmüş olup bütün uçuşlarda başlangıç değerleri kullanılmıştır. Bu yöntemde, iki farklı kontrol modelindeki parametre değişimlerine göre toplam 200 uçuş simülasyonu gerçekleştirilmiştir. Her bir simülasyonda ayarlanan K_p , K_i ve K_d parametreleri YSA modelinin eğitimindeki örnekler için çıkış değerleri olarak kullanılmıştır. Bu değerler program aracılığıyla rastgele belirlenmiş olup yuvarlanma ve yunuslama kontrol modellerinden bir tanesinin K_p , K_i ve K_d parametreleri değiştirilirken diğerleri sabit tutulmuştur. Bu şekilde, her bir hareket tipi için en optimum K_p , K_i ve K_d kontrol parametresinin tespit edilmesi amaçlanmıştır. Şekil 4.5'te sırasıyla yuvarlanma ve yunuslama kontrol modellerindeki değişimler ile örnek uçuş simülasyonları gösterilmiştir.

Tablo 4.3. Simülasyonda otonom uçuş için kullanılan K_p , K_i ve K_d parametreleri.

Simülasyon No	YUVARLANMA Kontrol Modeli			YUNUSLAMA Kontrol Modeli			SAPMA Kontrol Modeli		
	K_p	K_i	K_d	K_p	K_i	K_d	K_p	K_i	K_d
1	0.4804	3.7621	0.0618	0.15	0.10	0.004	0.20	0.02	0.002
2	1.1861	3.5162	0.0292	0.15	0.10	0.004	0.20	0.02	0.002
3	2.2328	2.2456	0.0668	0.15	0.10	0.004	0.20	0.02	0.002
4	3.8343	2.5277	0.0702	0.15	0.10	0.004	0.20	0.02	0.002
.	.	.	.	0.15	0.10	0.004	0.20	0.02	0.002
.	.	.	.	0.15	0.10	0.004	0.20	0.02	0.002
.	.	.	.	0.15	0.10	0.004	0.20	0.02	0.002
99	2.1613	0.3786	1.1124	0.15	0.10	0.004	0.20	0.02	0.002
100	0.7460	2.1344	0.7664	0.15	0.10	0.004	0.20	0.02	0.002
101	0.15	0.10	0.004	0.4804	3.7621	0.0618	0.20	0.02	0.002
.	0.15	0.10	0.004	.	.	.	0.20	0.02	0.002
.	0.15	0.10	0.004	.	.	.	0.20	0.02	0.002
.	0.15	0.10	0.004	.	.	.	0.20	0.02	0.002
198	0.15	0.10	0.004	1.3137	0.2655	1.6420	0.20	0.02	0.002
199	0.15	0.10	0.004	2.1613	0.3786	1.1124	0.20	0.02	0.002
200	0.15	0.10	0.004	0.7460	2.1344	0.7664	0.20	0.02	0.002



Şekil 4.5. Yuvarlanma (a) ve Yunuslama (b) kontrol modellerindeki K_p , K_i ve K_d parametre değişimleri ile yörünge takibinin gösterimi.

Uçuş sırasında 2 saniye aralıklarla kaydedilen pozisyon verileri içerisinde belirlenen köşe noktalarının enlem ve boylam değerleri (12 adet) YSA modelinin eğitiminde girdi olarak kullanılmıştır. YSA modelinin çıktısı ise K_p , K_i ve K_d parametreleridir. YSA modellerinin performansları Yuvarlanma ve Yunuslama kontrol modelleri için detaylı olarak aşağıdaki alt başlıklarda incelenmiştir.

4.1.2.1. Yuvarlanma Kontrol Modelinde PID Parametrelerinin Tahminlenmesi

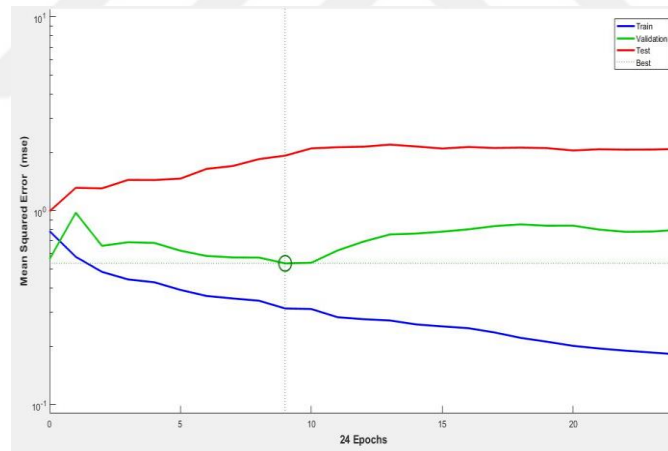
a. İleri Beslemeli Geri Yayılım Sinir Ağı (FFBPN)

Yuvarlanma kontrol modeli için FFBPN sinir ağı modeli Levenberg-Marquart ve Scaled Conjugate Gradient eğitim algoritmaları ve üç farklı aktivasyon fonksiyonu (logsig, pureln, tansig) kullanılarak eğitilmiştir. Sinir ağı modelinin 12 giriş, 10 nöron ve 3 çıkışı

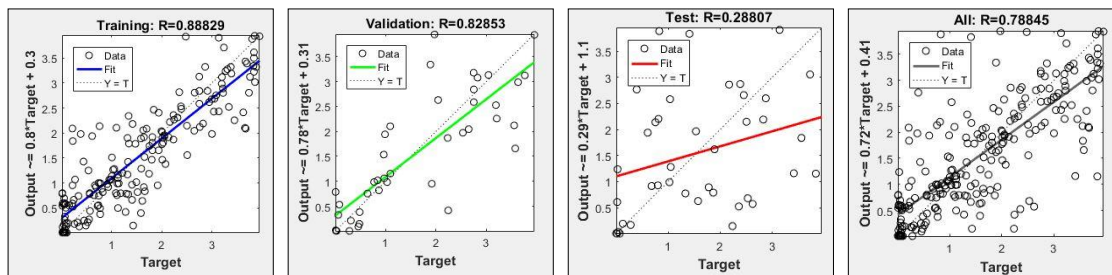
bulunmaktadır. FFBPN sinir ağı modellerinin MSE hata değerleri Tablo 4.4'te verilmiştir. Şekil 4.6'da ve Şekil 4.7'de sırasıyla en başarılı FFBPN sinir ağı modellerinin performans ve regresyon grafiği gösterilmektedir.

Tablo 4.4. FFBPN sinir ağı modellerinin MSE hata değerleri.

YSA Tipi	Eğitim Algoritması	Aktivasyon Fonksiyonu	Nöron Sayısı			MSE
			Giriş Katmanı	Gizli Katman	Çıkış Katmanı	
FFBPN	Levenberg-Marquart	logsig	12	10	3	2.2721
FFBPN	Levenberg-Marquart	purelin	12	10	3	0.5735
FFBPN	Levenberg-Marquart	tansig	12	10	3	0.5355
FFBPN	Scaled Conjugate Gradient	logsig	12	10	3	1.3957
FFBPN	Scaled Conjugate Gradient	purelin	12	10	3	0.9996
FFBPN	Scaled Conjugate Gradient	tansig	12	10	3	1.2463



Şekil 4.6. En başarılı FFBPN modeline ait performans grafiği.



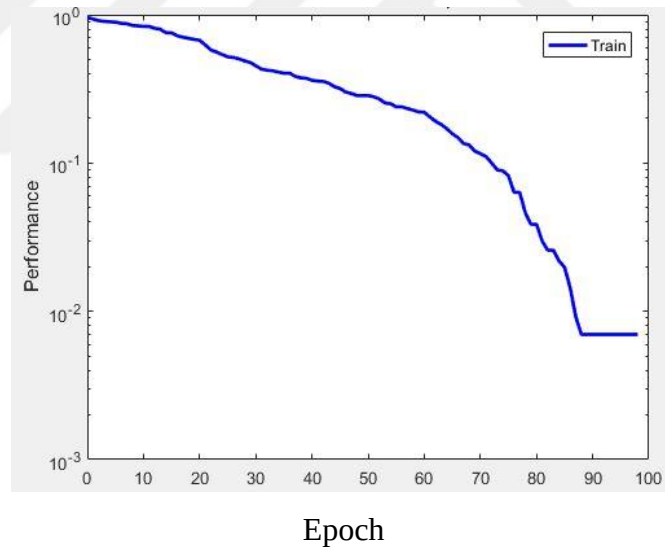
Şekil 4.7. En başarılı FFBPN modeline ait regresyon grafikleri.

b. Radyal Esaslı Sinir Ağı (RBNN)

RBNN sinir ağı modeli dört farklı spread değeri kullanılarak eğitilmiştir. Sinir ağı modelinin 12 giriş, 100 nöron ve 3 çıkışı bulunmaktadır. Farklı spread değerleriyle eğitilen RBNN sinir ağı modellerinin MSE hata değerleri Tablo 4.5'te verilmiştir. Şekil 4.8'de ise RBNN eğitim performans grafiği gösterilmiştir.

Tablo 4.5. Farklı spread değerleriyle eğitilen RBNN modellerinin MSE hata değerleri.

YSA Tipi	Spread	MSE
RBNN	0,001	0.0069
RBNN	0,01	0.1086
RBNN	0,1	0.5962
RBNN	1	0.7832



Şekil 4.8. Spread= 0,001 için RBNN eğitim performansı grafiği.

4.1.2.2. Yunuslama Kontrol Modelinde PID Parametrelerinin Tahminlenmesi

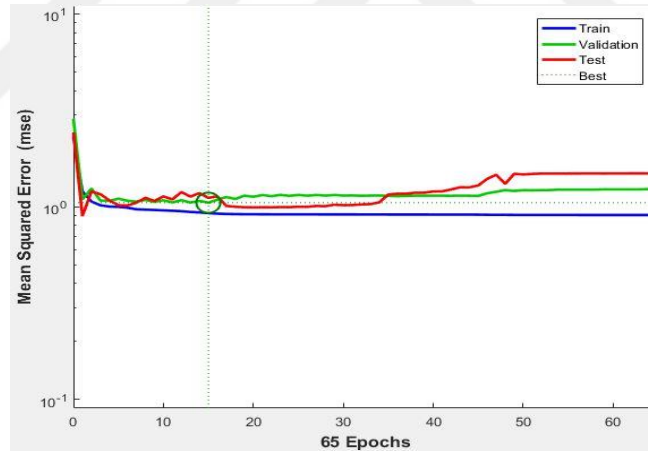
a. İleri Beslemeli Geri Yayılım Sinir Ağı (FFBPN)

Yunuslama kontrol modeli için FFBPN sinir ağı modeli yine Levenberg-Marquart ve Scaled Conjugate Gradient eğitim algoritmaları ve üç farklı aktivasyon fonksiyonu (logsig, pureln, tansig) kullanılarak eğitilmiştir. Sinir ağı modelinin aynı şekilde 12 giriş, 10 nöron ve 3 çıkışı bulunmaktadır. FFBPN sinir ağı modellerinin MSE hata değerleri

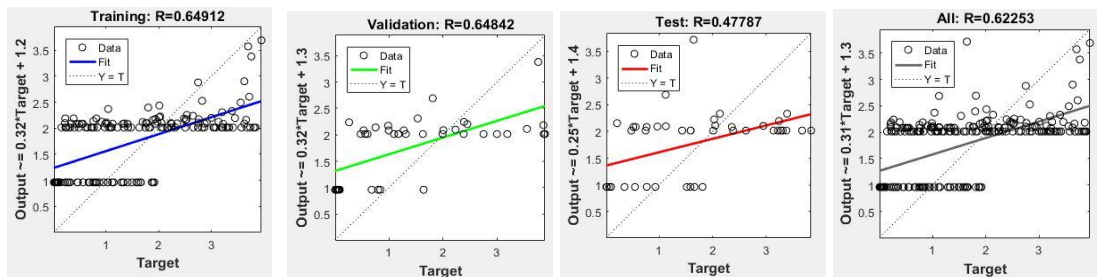
Tablo 4.6’da verilmiştir. Şekil 4.9 ve Şekil 4.10’da ise sırasıyla en başarılı FFBNP sinir ağı modellerinin performans grafiği ve regresyon grafikleri gösterilmektedir.

Tablo 4.6. FFBNP sinir ağı modellerinin MSE hata değerleri.

YSA Tipi	Eğitim Algoritması	Aktivasyon Fonksiyonu	Nöron Sayısı			MSE
			Giriş Katmanı	Gizli Katman	Çıkış Katmanı	
FFBNP	Levenberg-Marquart	logsig	12	10	3	2.4632
FFBNP	Levenberg-Marquart	purelin	12	10	3	1.1244
FFBNP	Levenberg-Marquart	tansig	12	10	3	1.0480
FFBNP	Scaled Conjugate Gradient	logsig	12	10	3	1.6291
FFBNP	Scaled Conjugate Gradient	purelin	12	10	3	1.3977
FFBNP	Scaled Conjugate Gradient	tansig	12	10	3	1.5788



Şekil 4.9. En başarılı FFBNP modeline ait performans grafiği.



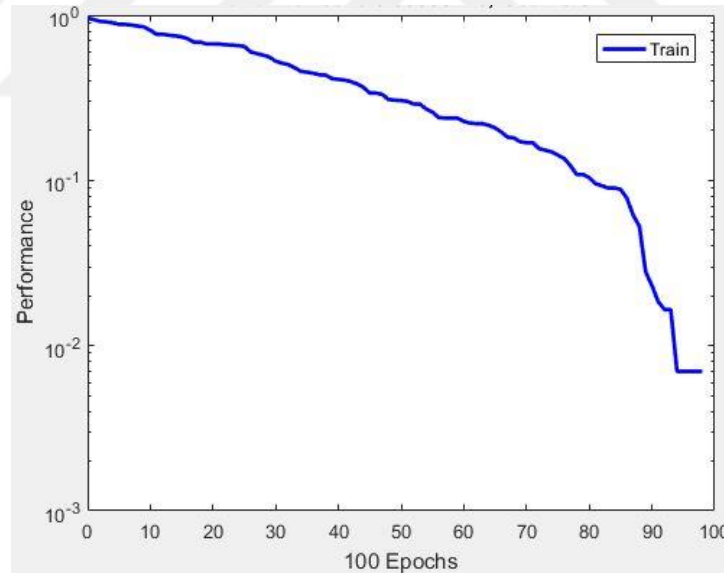
Şekil 4.10. En başarılı FFBNP modeline ait regresyon grafikleri.

b. Radyal Esaslı Sinir Ağı (RBNN)

RBNN sinir ağı modeli yuvarlanma modelinde olduğu dört farklı spread değeri kullanılarak eğitilmiştir. Sinir ağı modelinin 12 giriş, 100 nöron ve 3 çıkışı bulunmaktadır. Farklı spread değerleriyle eğitilen RBNN sinir ağı modellerinin MSE hata değerleri Tablo 4.7’de verilmiştir. Şekil 4.11’de ise RBNN sinir ağı modelinin performans grafiği gösterilmiştir.

Tablo 4.7. Farklı spread değerleriyle eğitilen RBNN modellerinin MSE hata değerleri.

YSA Tipi	Spread	MSE
RBNN	0,001	0.0069
RBNN	0,01	0.1466
RBNN	0,1	0.6861
RBNN	1	0.8912



Şekil 4.11. Spread= 0,001 için RBNN eğitim performansı grafiği.

Tablo 4.8’de istenilen yörüngenin tahminlenmesi için YSA modelinde girdi olarak kullanılan değerler ve çıktı olarak tahminlenen K_p , K_i ve K_d parametreleri sırasıyla Yuvarlanma ve Yunuslama modelleri için gösterilmiştir.

Tablo 4.8. İstenilen yörünge için tahminlenen K_p , K_i ve K_d parametreleri.(a) Yuvarlanma kontrol modeli için, (b) yunuslama kontrol modeli için elde edilen değerler.

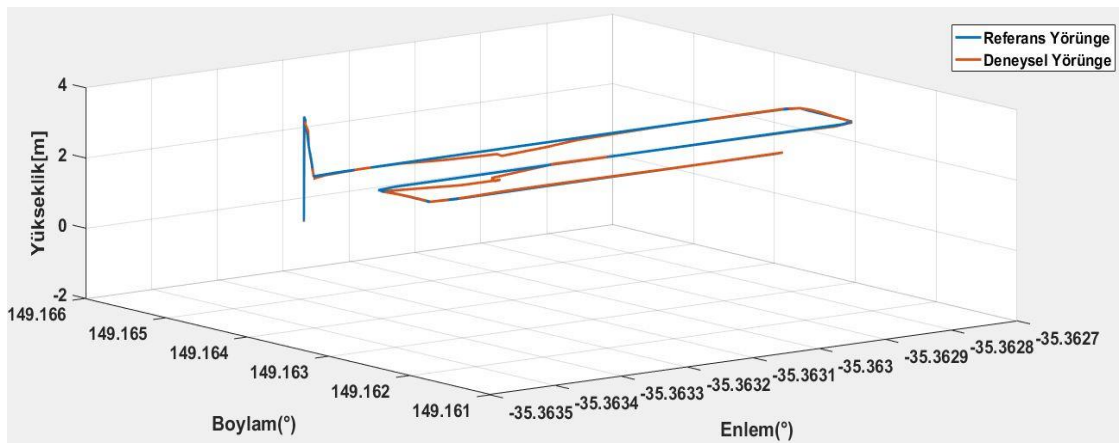
(a)

Girdi						Çıktı		
P_{1b}	P_{2b}	P_{3b}	P_{4b}	P_{5b}	P_{6b}	P_{FFBPN}	I_{FFBPN}	D_{FFBPN}
149.16368	149.16368	149.16293	149.16293	149.16219	149.16219	3.3265	0.9081	0.002
P_{1e}	P_{2e}	P_{3e}	P_{4e}	P_{5e}	P_{6e}	P_{RBNN}	I_{RBNN}	D_{RBNN}
-35.36315	-35.36255	-35.36255	-35.36315	-35.36315	-35.36255	3.8853	0.2118	0.004

(b)

Girdi						Çıktı		
P_{1b}	P_{2b}	P_{3b}	P_{4b}	P_{5b}	P_{6b}	P_{FFBPN}	I_{FFBPN}	D_{FFBPN}
149.16368	149.16368	149.16293	149.16293	149.16219	149.16219	1.248	0.2963	0.0213
P_{1e}	P_{2e}	P_{3e}	P_{4e}	P_{5e}	P_{6e}	P_{RBNN}	I_{RBNN}	D_{RBNN}
-35.36315	-35.36255	-35.36255	-35.36315	-35.36315	-35.36255	3.146	0.523	0.004

En iyi performanslı RBNN YSA modeli ile tahminlenen K_p , K_i ve K_d parametreleri ile gerçekleştirilen otonom simülasyon uçuşunun referans yörünge ile karşılaştırılması Şekil 4.12’de gösterilmiştir.

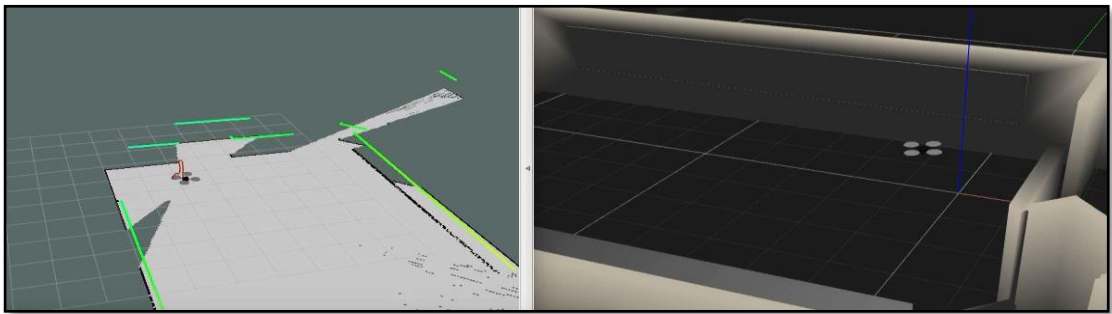


Şekil 4.12. Tahminlenen K_p , K_i ve K_d parametreleri ile simülasyonda gerçekleştirilen otonom uçuşun, referans yörüngeyle karşılaştırılması.

4.1.3. Konumlandırma ve Haritalama Simülasyon Çalışmaları

Robotik sistemin, dar koridorlardan oluşan ve üzeri örtülü elma bahçesinde otonom olarak yörünge takibi gerçekleştirebilmesi için en uygun yörünge ve bu yörüngeyi takip ederken kullanılacak en uygun PID kontrolcü parametrelerinin optimum değerlerde belirlenmesi için EKVH yönteminden de faydalanılmıştır. Bu yöntemin kullanılmasında iki önemli amaç mevcuttur. Birincisi; dar ve üzeri kapalı alanda uydu sinyallerinden elde edilen konum verilerine göre robotun yörüngesi üzerinde inceleme yapılmasının ve parametre ayarlamasının gerçekleştirilmesinin tam olarak verimli olmamasıdır. Bu sebeple lidar sensörden elde edilen veriler EKVH tekniği ile birleştirilmiş ve bir ROS bileşeni olan Rviz görselleştirme arayüzü üzerinde ortamın haritası oluşturulmuştur. Böylece robotun izlediği yörünge bu harita üzerinde çizdirilmiştir. Bu sayede uygulanan K_p , K_i ve K_d parametrelerine göre robotik sistemin yörünge hatalarının farklı metotlarla incelenmesi mümkün olmuştur. İkincisi ise; EKVH tekniği ile elde edilen sonuçların uygulanabilirliğinin kanıtlanması ile birlikte sonraki çalışmalar için uydu sinyallerinden bağımsız olarak sadece EKVH tekniği ile elde edilen veriler kullanarak robotun otonom yörünge takibinin gerçekleştirilebileceği gözlemlenmiştir. EKVH ile ilgili böyle bir çalışmanın gerçekleştirilmesi yeni bir araştırma alanı oluşturacağından buradan elde edilen bulgular sonraki çalışmalarda kullanılmak üzere kaydedilmiştir. Çalışma kapsamında esas olarak derin öğrenme ile elma tespitinin gerçekleştirilmesi ve elmaların sayılması problemine odaklanılmıştır.

Gazebo platformu kullanılarak, deneysel çalışma gerçekleştirilmeden önce robotik sisteme benzer yapıda tasarlanan dört rotorlu bir dron sistemi EKVH yöntemiyle oluşturulan sanal ortamda test edilmiştir. Şekil 4.13'te oluşturulan sanal ortamdaki robotun izlediği yörünge verilmiştir.



Şekil 4.13. Gazebo simülasyon ortamında EKVH uygulamasının gösterimi.

ROS ve Gazebo platformlarının robotik çalışmalar için sunduğu önemli özelliklerden bir tanesi gerçekte kullanılan ekipmanların özelliklerini çok iyi yansıtan kütüphanelerin sunulmasıdır. Bu önemli özellik sayesinde, deneysel çalışmada kullanılan Hokuyo lidar sensörünün kütüphanesi simülasyon çalışmalarında kullanılmıştır. Bunun sonucunda, simülasyonda elde edilen başarılı gözlemler sahada da elde edilebilmiştir.

4.2. Deneysel Çalışmalar

4.2.1. Otonom Uçuş ile Yörünge Takibi için Deneysel Çalışmalar

Deneysel çalışmalar elma bahçesinde belirlenen bir koridorda gerçekleştirilmiştir. Öncelikle URS'nin takip etmesi istenilen yörünge üzerinde belirlenen ara noktalar uydudan alınan sinyaller ile harita üzerinde işaretlenmiştir. Şekil 4.14'te deneysel çalışma için belirlenen noktaların harita üzerinde gösterildiği görsel verilmiştir.



Şekil 4.14. Deneysel çalışma için belirlenen noktaların harita üzerinde gösterilmesi.

URS'nin ulaşması için işaretlenen bu ara noktaların koordinat bilgileri Python dili ile yazılan uçuş programında Şekil 4.15'te belirtilen komut satırına girilmiştir.

```

def gorev_ekle():
    global komut
    komut = urs.commands

    komut.clear ()
    time.sleep(1)

    #KALKIS
    komut.add(Command(0, 0, 0, mavutil.mavlink.MAV_FRAME_GLOBAL_RELATIVE_ALT, mavutil.mavlink.MAV_CMD_NAV_TAKEOFF, 0, 0, 0, 0, 0, 0, 0, 0, 0, 10))

    #HEDEF
    komut.add(Command(0, 0, 0, mavutil.mavlink.MAV_FRAME_GLOBAL_RELATIVE_ALT, mavutil.mavlink.MAV_CMD_NAV_WAYPOINT, 0, 0, 0, 0, 0, -35.3634010, 149.1645098, 10))
    komut.add(Command(0, 0, 0, mavutil.mavlink.MAV_FRAME_GLOBAL_RELATIVE_ALT, mavutil.mavlink.MAV_CMD_NAV_WAYPOINT, 0, 0, 0, 0, 0, -35.3625610, 149.1643918, 10))
    komut.add(Command(0, 0, 0, mavutil.mavlink.MAV_FRAME_GLOBAL_RELATIVE_ALT, mavutil.mavlink.MAV_CMD_NAV_WAYPOINT, 0, 0, 0, 0, 0, -35.3625785, 149.1636729, 10))
    komut.add(Command(0, 0, 0, mavutil.mavlink.MAV_FRAME_GLOBAL_RELATIVE_ALT, mavutil.mavlink.MAV_CMD_NAV_WAYPOINT, 0, 0, 0, 0, 0, -35.3634097, 149.1638124, 10))
    komut.add(Command(0, 0, 0, mavutil.mavlink.MAV_FRAME_GLOBAL_RELATIVE_ALT, mavutil.mavlink.MAV_CMD_NAV_WAYPOINT, 0, 0, 0, 0, 0, -35.3634447, 149.1630721, 10))
    komut.add(Command(0, 0, 0, mavutil.mavlink.MAV_FRAME_GLOBAL_RELATIVE_ALT, mavutil.mavlink.MAV_CMD_NAV_WAYPOINT, 0, 0, 0, 0, 0, -35.3625960, 149.1630077, 10))

    #RTL
    komut.add(Command(0, 0, 0, mavutil.mavlink.MAV_FRAME_GLOBAL_RELATIVE_ALT, mavutil.mavlink.MAV_CMD_NAV_RETURN_TO_LAUNCH, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0))

    #DOGRULAMA
    komut.add(Command(0, 0, 0, mavutil.mavlink.MAV_FRAME_GLOBAL_RELATIVE_ALT, mavutil.mavlink.MAV_CMD_NAV_RETURN_TO_LAUNCH, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0))

    komut.upload()
    print("Komutlar Yukleniyor...")

```

Şekil 4.15. Hedef noktaları tanımlayan program satırları.

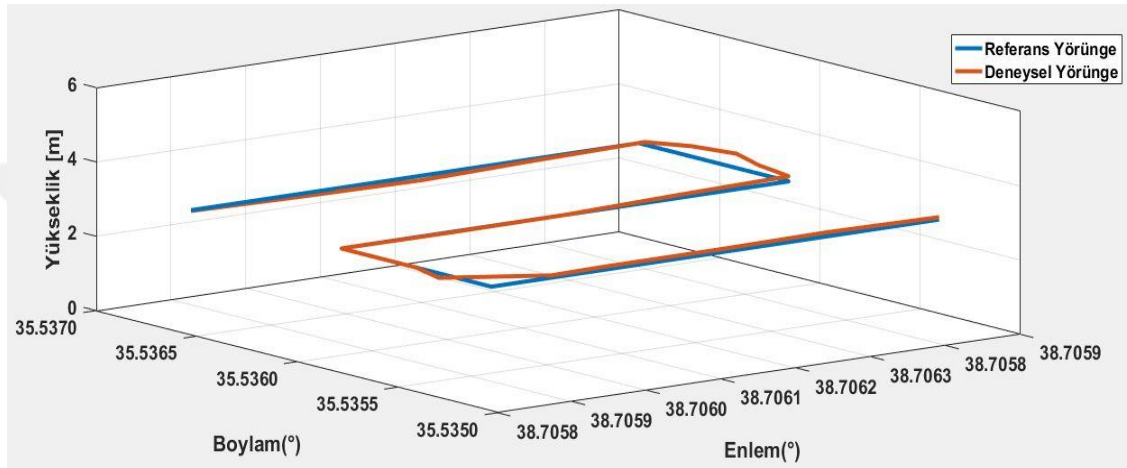
Şekil 4.15'te verilen kod parçasında, komut.add komutuyla robotik sistemin otonom olarak ulaşması için belirlenen noktaların tanımlanması gerçekleştirilmiştir. Ayrıca; #KALKIS etiketiyle belirtilen bölümdeki komut.add komutuyla da URS'nin başlangıç pozisyonundan 10 metre yükseklikte belirlenen noktaya kalkış yapması tanımlanmış, #HEDEF etiketiyle belirtilen bölümdeki komut.add komutuyla ise uçuş esnasında URS'nin uğrayacağı ara noktalar kodlanmış ve son olarak #RTL etiketiyle belirtilen bölümdeki komut.add komutuyla da URS'nin başlangıç noktasına dönmesini sağlayan komut tanımlanmıştır. Sonuç olarak yazılan uçuş programı çalıştırıldığında 2.4 GHz radyo frekansıyla bilgisayar üzerinden iletilen komutlar vasıtasıyla hava aracının görevini yerine getirmesi sağlanmıştır. Uydu sinyalleriyle elde edilen konum verileri aracılığıyla otonom uçuş sonucunda sistemin izlediği yörünge Şekil 4.16'da gösterilmiştir.



Şekil 4.16. Otonom uçuşta izlenen yörünge uydur haritası üzerinde görünümü.

4.2.2. Kontrol Parametrelerinin Optimizasyonu İçin Deneysel Çalışmalar

Bu bölümde, daha önce YSA yöntemi ile simülasyon çalışmalarından elde edilen K_p , K_i ve K_d parametreleri ile kontrolcünün Erciyes Üniversitesi kampüsünde bulunan çim sahada belirlenen deneysel ortamdaki performansı değerlendirilmiştir. Şekil 4.17’de optimize edilen K_p , K_i ve K_d parametreleri ile gerçekleştirilen deneysel uçuşun belirlenen referans yörüngeyi takip başarısı gösterilmiştir.



Şekil 4.17. Kontrol parametrelerinin optimizasyonunun deneysel yörünge kontrolü üzerine etkisinin gösterimi.

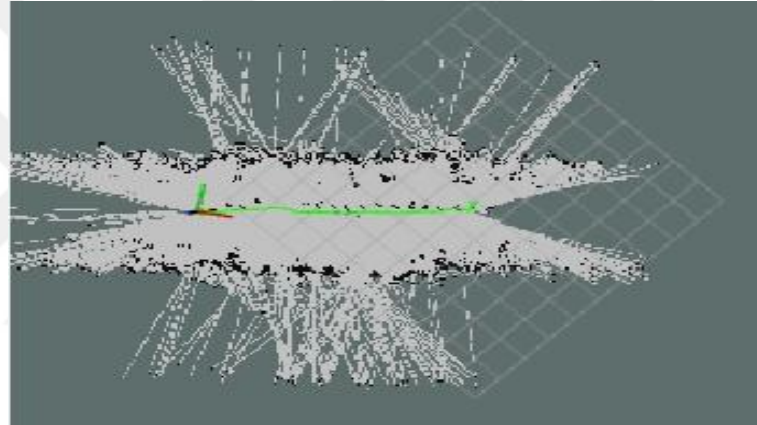
4.2.3. Konumlandırma ve Haritalama Deneysel Çalışmaları

Simülasyon ortamında lidar sensör ile ortamın haritasının çıkarılması ve hava aracının yörüngesinin tespit edilmesi önceki bölümlerde anlatılmıştır. Bu çalışmalarda geliştirilen kodlar, deneysel çalışmaların gerçekleştirildiği elma bahçesindeki otonom uçuş sırasında gerçek zamanlı olarak da test edilmiştir. Şekil 4.18’de URS’nin uydu sinyalleriyle elde ettiği pozisyon verileriyle uydu haritası üzerinde gösterilen yörüngesi ve EKVH tekniği ile elde edilen verilerle oluşturulan yörüngesi karşılaştırmalı olarak verilmiştir. Uydu sinyalleriyle oluşturulan yörüngedeki hatalar bir sonraki ağaç koridoruna kadar 2 metreyi bulurken, EKVH tekniği ile oluşturulan yörüngedeki hataların koridor içerisinde 20 santimetre ile sınırlı kaldığı görülmüştür.

(a)



(b)



Şekil 4.18. (a) Uydu sinyalleri ile, (b) EKVH tekniği ile oluşturulan yörüngelerin gösterimi.

4.2.4. Derin Öğrenme ile Elma Tespiti Çalışmaları

Deney için belirlenen elma bahçesi koridorunda URS ile otonom olarak gerçek zamanlı takip edilen yörüngeden hareketli görüntüler elde edilmiştir. Bu görüntüler, geliştirilen derin sinir ağ modelleri ile elmaların tespiti amacıyla kullanılmıştır. Çalışmada, SSD-MobileNet ve Faster R-CNN mimarileri esas alınarak geliştirilen sinir ağ modelleri kullanılmış ve aşağıda belirtilen altı farklı durum için performansları analiz edilmiştir:

1. MS COCO veri seti ile eğitilmiş hazır SSD-MobileNet (SM) sinir ağ (SA) modeli ile elma tespiti.
2. MS COCO veri seti ile eğitilmiş hazır Faster R-CNN (FRC) sinir ağ (SA) modeli ile elma tespiti.

3. Özgün veri seti ile eğitilen SSD-Mobilenet (SM) sinir ağ (SA) modeli ile elma tespiti.
4. Özgün veri seti ile eğitilen Faster R-CNN (FRC) sinir ağ (SA) modeli ile elma tespiti.
5. Hiperparametreleri optimize edilen SSD-Mobilenet (SM) sinir ağ (SA) modeli ile elma tespiti.
6. Hiperparametreleri optimize edilen Faster R-CNN (FRC) sinir ağ (SA) modeli ile elma tespiti.

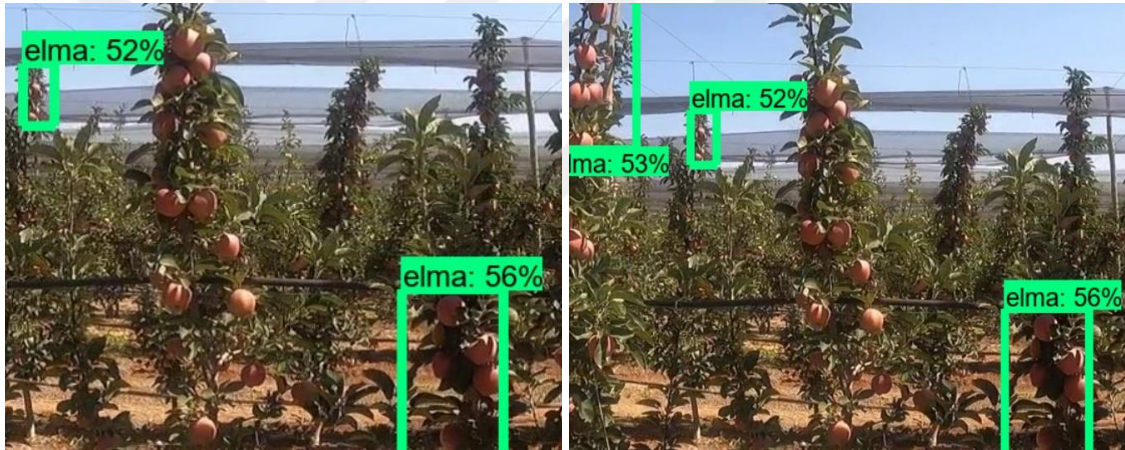
4.2.4.1. MS COCO Veri Seti ile Eğitilmiş Hazır SM SA Modeli ile Elma Tespiti

MS COCO, bilgisayarlı görmede çokça kullanılan bir veri kümesidir. MS COCO veri seti, Microsoft tarafından yayınlanan büyük ölçekli bir nesne algılama, görüntü segmentasyonu ve altyazı oluşturma veri setidir [79]. Makine öğrenimi ve bilgisayarla görme çalışmalarında sıklıkla kullanılmakta olup önerilen metotlar için performans kıyaslamasında referans olarak kullanılmaktadır. Bu veri seti içerisinde çok sayıda objeye ait etiketlenmiş görüntüler mevcuttur. MS COCO veri seti bazı çalışmalar için oldukça uygun olmasına rağmen, bu çalışmadaki gibi zorlayıcı şartlar altında gerçekleştirilen spesifik çalışmalarda yetersiz kalabilmektedir. Gerçekleştirilen tez çalışmasında hareketli sistem ile elde edilen görüntüler üzerinde nesne tespiti üzerinde çalışılması, MS COCO gibi standart veri setleriyle eğitilen hazır modeller için zorlayıcı olabilmektedir. Ayrıca elma ağaçlarında karanlıkta kalan veya yaprakların arkasında kalan elmaların tespit edilmesi neredeyse imkansız hale gelebilmektedir. Bu konudaki sıkıntıların giderilmesi için çalışmaya MS COCO veri setiyle eğitilen hazır derin sinir ağ modelleriyle nesne tespiti çalışmaları kullanılarak başlanmıştır. Bu bölümde MS COCO veri setiyle eğitilen evrişimli bir derin sinir ağ modeli olan SSD-Mobilenet ile gerçekleştirilen elma tespiti çalışmasının performansı incelenmiştir. Tablo 4.9’da bu çalışmada kullanılan modele ait eğitim parametreleri verilmiştir.

Tablo 4.9. MS COCO ile eğitilen SSD-Mobilenet sinir ağının eğitim parametreleri.

Parametre	Değeri
Sınıf Sayısı	90
Aktivasyon Fonksiyonu	RELU
Öğrenme Oranı	0.133-0.8
İterasyon Sayısı	50000
Batch Boyutu	512

Şekil 4.19’da MS COCO veri seti kullanılarak eğitilen SSD-Mobilenet derin sinir ağ modelinin deneysel elma tespiti çalışmasına ait görüntüleri verilmiştir.



Şekil 4.19. MS COCO veri seti kullanılarak eğitilen SSD-Mobilenet derin sinir ağ modelinin deneysel elma tespiti çalışmasına ait görüntüleri.

Şekil 4.19’den de anlaşılacağı gibi hazır SSD-Mobilenet modeliyle gerçekleştirilen çalışmada elmaların tespit edilme doğruluk oranı son derece düşüktür, Tablo 4.10’da ilgili sonuçlar yer almaktadır.

Tablo 4.10. MS COCO veri setiyle eğitilen SSD-Mobilenet modelinin gerçekleştirilen elma tespit çalışmasına ait sonuçları.

Sayılan Elma	Yanlış Sayılan Elma	Toplam Elma Sayısı	Doğruluk Oranı
12	6	150	%8

4.2.4.2. MS COCO Veri Seti ile Eğitilmiş Hazır FRC SA Modeli ile Elma Tespiti

Bu bölümde MS COCO veri setiyle eğitilen evrişimli bir sinir ağı modeli olan Faster R-CNN ile gerçekleştirilen elma tespiti çalışmasının performansı incelenmiştir. Tablo 4.11’de FRC modelinde kullanılan eğitim parametreleri verilmiştir.

Tablo 4.11. MS COCO ile eğitilen FRC modelinde kullanılan eğitim parametreleri.

Parametre	Değeri
Sınıf Sayısı	90
Aktivasyon Fonksiyonu	RELU
Öğrenme Oranı	0.133-0.8
İterasyon Sayısı	50000
Batch Boyutu	512

MS COCO veri seti kullanılarak eğitilen FRC modeline ait deneysel elma tespiti çalışması görüntüleri Şekil 4.20’de verilmiştir.



Şekil 4.20. MS COCO veri seti kullanılarak eğitilen FRC modeline ait deneysel elma tespiti çalışmasına ait görüntüler.

Şekil 4.20’den de anlaşılacağı gibi hazır Faster-RCNN modeliyle gerçekleştirilen çalışmada elmaların tespit edilme doğruluk oranı hazır SSD-MobileNet modeline göre daha iyi olsa da yine de oldukça düşüktür ve Tablo 4.12’de ilgili sonuçlar yer almaktadır.

Tablo 4.12. MS COCO veri setiyle eğitilen FRC modelinin gerçekleştirilen elma tespit çalışmasına ait sonuçları.

Sayılan Elma	Yanlış Sayılan Elma	Toplam Elma Sayısı	Doğruluk Oranı
27	8	150	%18

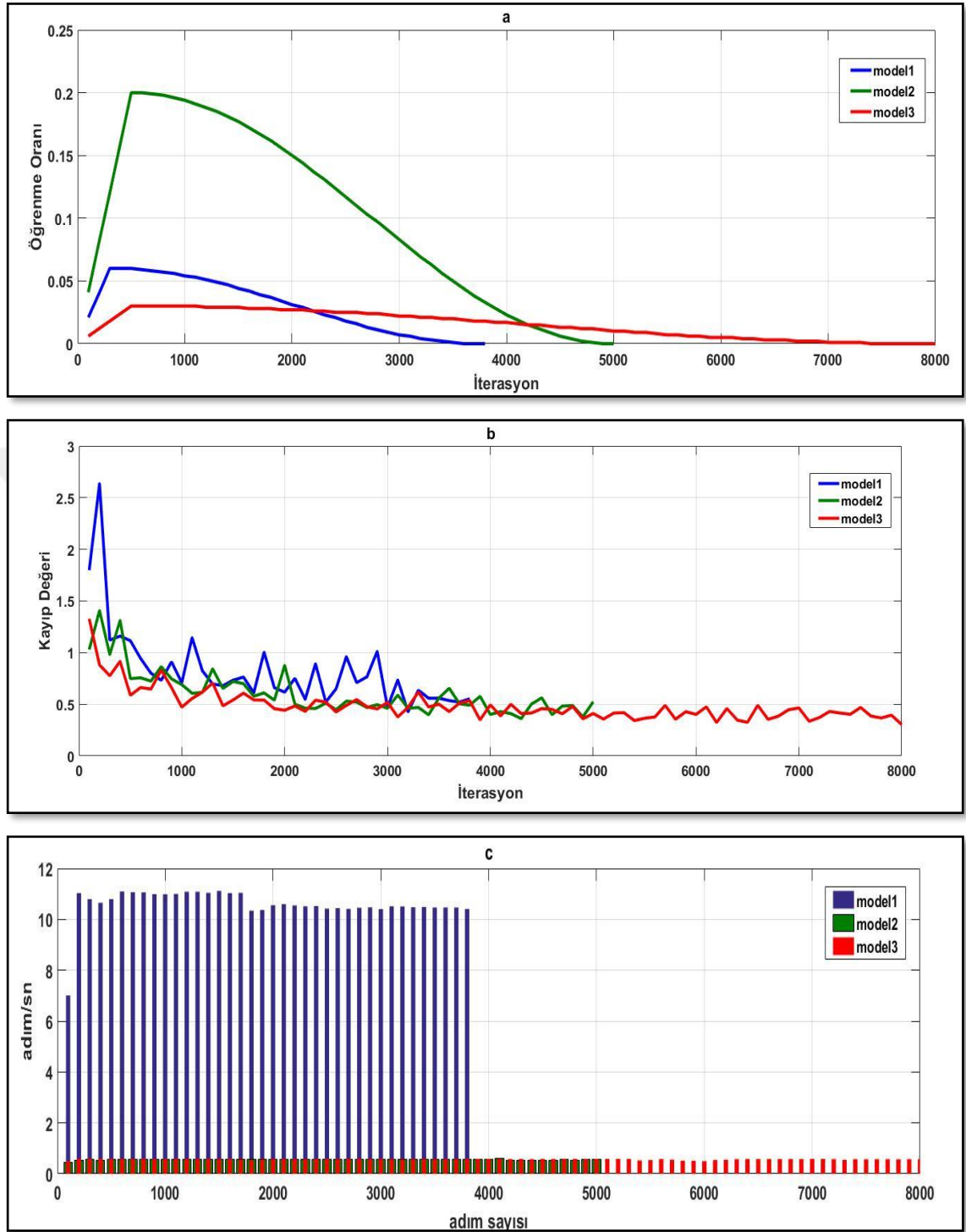
4.2.4.3. Özgün Veri Seti ile Eğitilen SM SA Modeli ile Elma Tespiti

MS COCO veri setiyle eğitilen hazır derin sinir ağ modelleri ile gerçekleştirilen çalışmalar, elmaların tespit edilmesinde istenilen doğruluk oranı seviyelerini yakalayamamıştır. Bu sebeple, deneyin gerçekleştiği elma bahçesinden URS ile elde edilen görüntülerin 100 tanesi seçilerek etiketleme işlemi uygulanmıştır. Her bir görüntü üzerinde bulunan elmalar tek tek dikdörtgen sınırlayıcı kutular içerisine alınarak etiketleme işlemi yapılmıştır. Böylece, görüntü üzerindeki tüm elmaların ayrı ayrı derin sinir ağ modeline öğretilmesi sağlanmıştır. Sonuç olarak 4000 adet etiketlenmiş elma görüntüsü derin sinir ağ modelini eğitmek için .record formatındaki veri seti haline dönüştürülmüştür. Tablo 4.13'te, özgün veri seti ile SSD-Mobilenet derin sinir ağ modelinin eğitiminde kullanılan üç farklı hiperparametre seti verilmiştir.

Tablo 4.13. SSD-Mobilenet derin sinir ağının özgün veri setiyle eğitiminde kullanılan hiperparametre setleri.

Parametre	1.Model Değeri	2.Model Değeri	3.Model Değeri
Sınıf Sayısı	1	1	1
Aktivasyon Fonksiyonu	RELU	RELU	RELU
Öğrenme Oranı	0.001-0.06	0.001-0.2	0.0001-0.03
İterasyon Sayısı	3800	5000	8000
Batch Boyutu	2	8	8

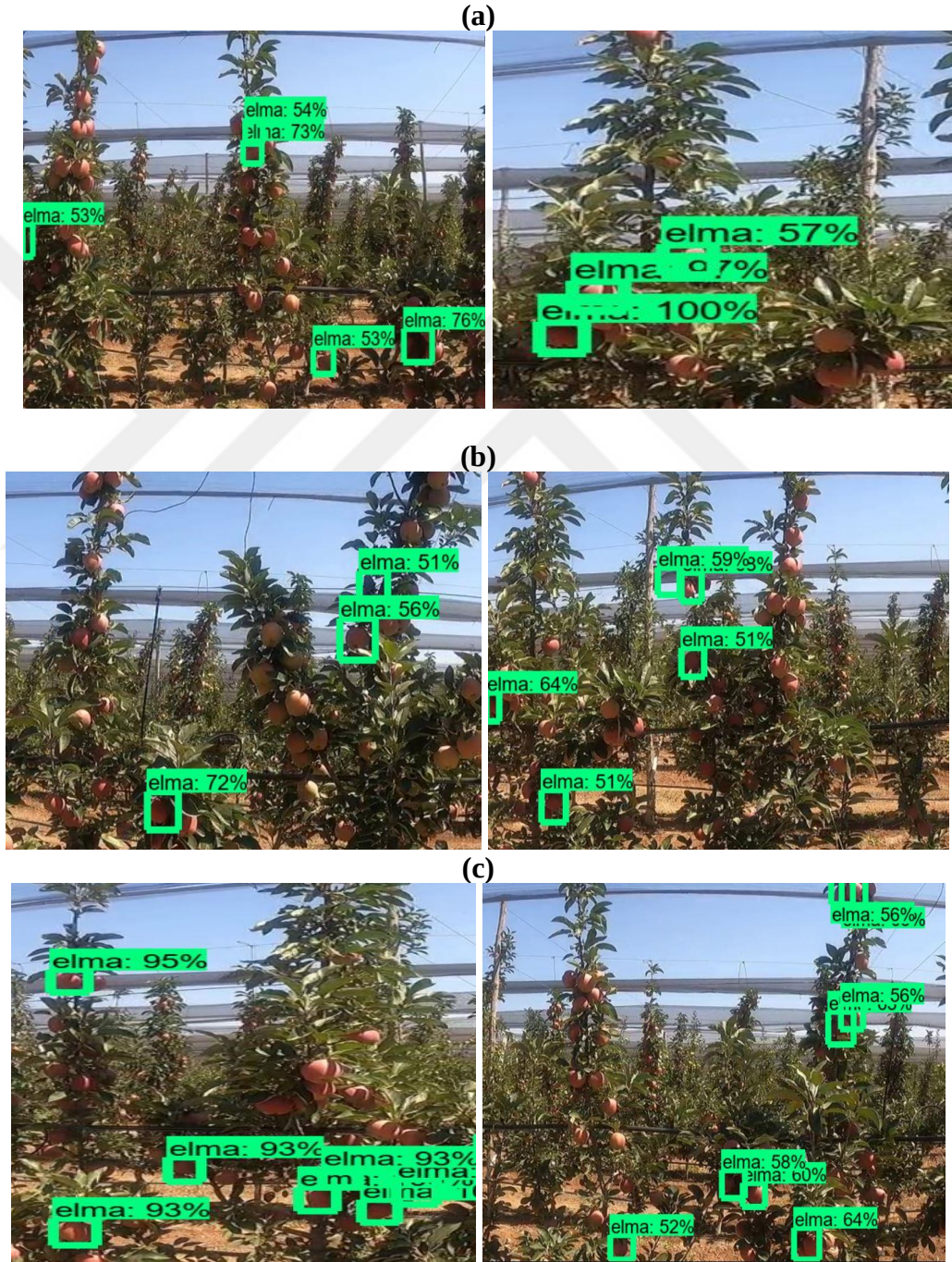
Üç farklı modelin eğitimi sonucunda elde edilen performans grafikleri Şekil 4.21'de sırasıyla öğrenme oranı-iterasyon grafiği, kayıp değeri-iterasyon grafiği ve saniyede işlenen adım sayısı- toplam iterasyon grafiği için gösterilmiştir.



Şekil 4.21. (a) Öğrenme Oranı-İterasyon grafiği, (b) Kayıp-İterasyon grafiği, (c) Saniyede İşlenen Adım Sayısı-Toplam İterasyon grafiği.

Şekil 4.21. (a) ile verilen grafikte üç farklı model ile belirlenen hiperparametrelere göre öğrenme oranının değişimi gösterilmiştir. Şekil 4.21. (b)'de verilen grafik ile modellerin kayıp fonksiyonu değerleri verilmiştir; bu kısımda öğrenme oranı ile kayıp fonksiyonu arasındaki ilişki gösterilmiştir. En iyi sonuç, 0.3 kayıp değeri ile model3 üzerinde elde

edilmiştir. Model2 0.52 kayıp değeri üretirken, model1 ise 0.55 kayıp değeri üretmiştir. Son grafik olan Şekil 4.21. (c), modelin işleme hızını gösteren saniyede ilerlenen adım sayısını belirtmektedir. Şekil 4.22’de özgün veri setiyle eğitilen SSD-Mobilenet sinir ağ modelinin deneysel elma tespiti çalışmalarına ait görüntüleri sırasıyla model1, model2, model3 için verilmiştir.



Şekil 4.22. Özgün veri setiyle eğitilen SSD-Mobilenet sinir ağ modelinin deneysel elma tespiti çalışmasına ait görüntüleri. (a) model1 ile, (b) model2 ile ve (c) model3.

Tablo 4.14’te üç farklı model ile gerçekleştirilen SSD-MobileNet derin sinir ağ modeli ile elma tespit çalışmasına ait performans metrikleri sunulmuştur. Üç farklı model ile video görüntüleri üzerinde gerçekleştirilen elma tespit çalışmalarına ait sonuçlar incelendiğinde daha düşük kayıp değeri elde edilen model3 ile daha yüksek doğrulukta elma tespit edildiği görülmektedir.

Tablo 4.14. Sinir ağ modellerinin elma tespit çalışmasına ait performans sonuçları.

Model	Sayılan Elma	Yanlış Sayılan Elma	Toplam Elma Sayısı	Doğruluk Oranı	F1 Skoru	Hassasiyet
model3	34	2	150	%23	0.43	0.46
model2	29	1	150	%19	0.39	0.42
model1	27	2	150	%18	0.37	0.40

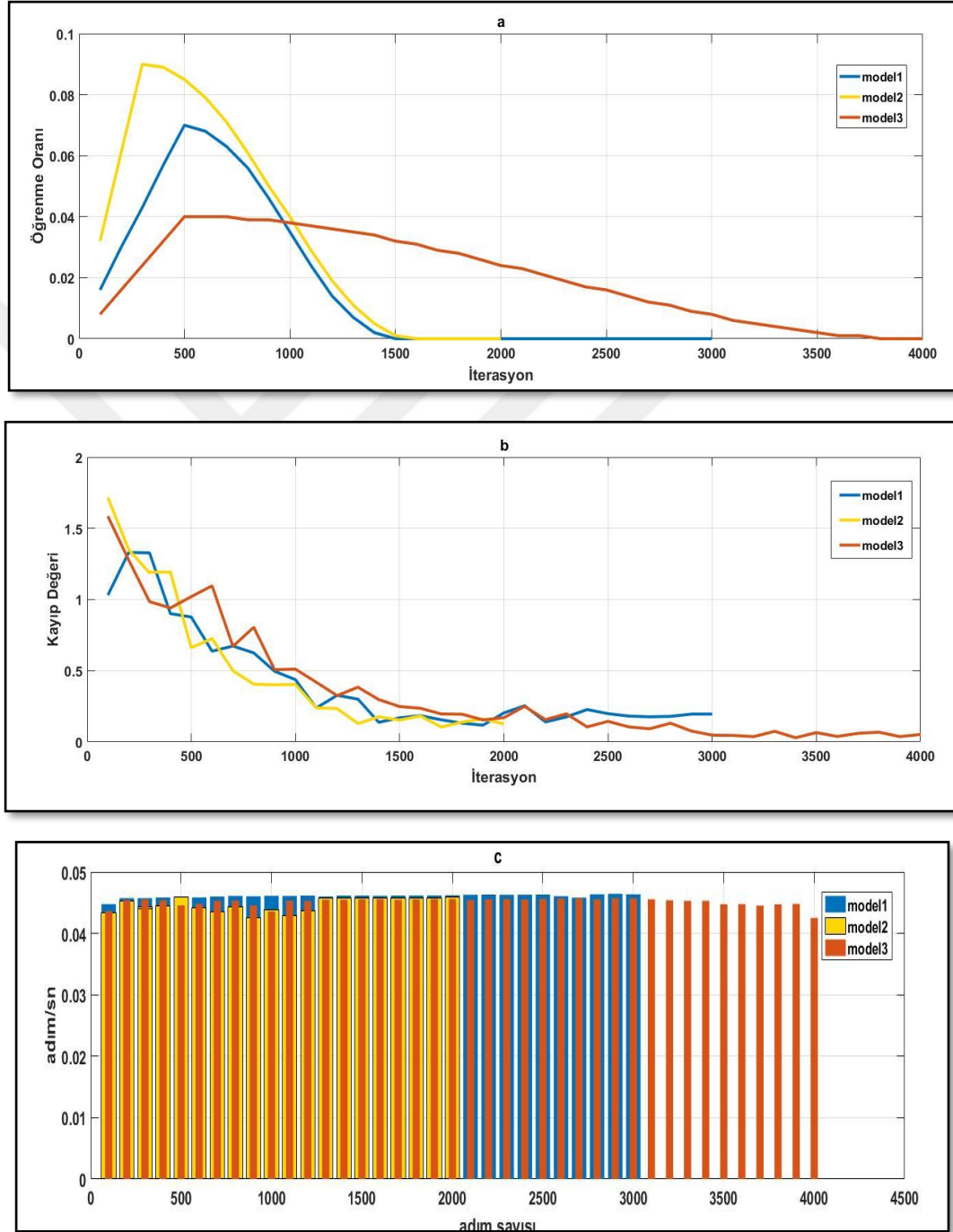
4.2.4.4. Özgün Veri Seti ile Eğitilen FRC SA Modeli ile Elma Tespiti

Özgün veri setiyle eğitilen SSD-MobileNet mimarili sinir ağ modelleri ile gerçekleştirilen çalışma sonuçları istenilen seviyelere çıkartılamamıştır. En iyi denemede elde edilen %23 doğruluk oranı rekolte tespit etmek için geliştirilen sistemde yeterli bulunmamaktadır. Deneysel çalışmalarda kullanılan ikinci mimari ise SSD-MobileNet’e göre daha yavaş fakat doğruluk oranı daha yüksek olan Faster-RCNN mimarisidir. Tablo 4.15’te, oluşturulan özgün veri setiyle Faster-RCNN sinir ağ modelinin eğitilmesi için kullanılan üç farklı parametre seti verilmiştir.

Tablo 4.15. Faster-RCNN derin sinir ağının özgün veri setiyle eğitilmesinde kullanılan hiperparametre setleri.

Parametre	1.Model Değeri	2.Model Değeri	3.Model Değeri
Sınıf Sayısı	1	1	1
Aktivasyon Fonksiyonu	RELU	RELU	RELU
Öğrenme Oranı	0.03-0.7	0.03-0.09	0.0001-0.04
İterasyon Sayısı	3000	2000	4000
Batch Boyutu	2	4	4

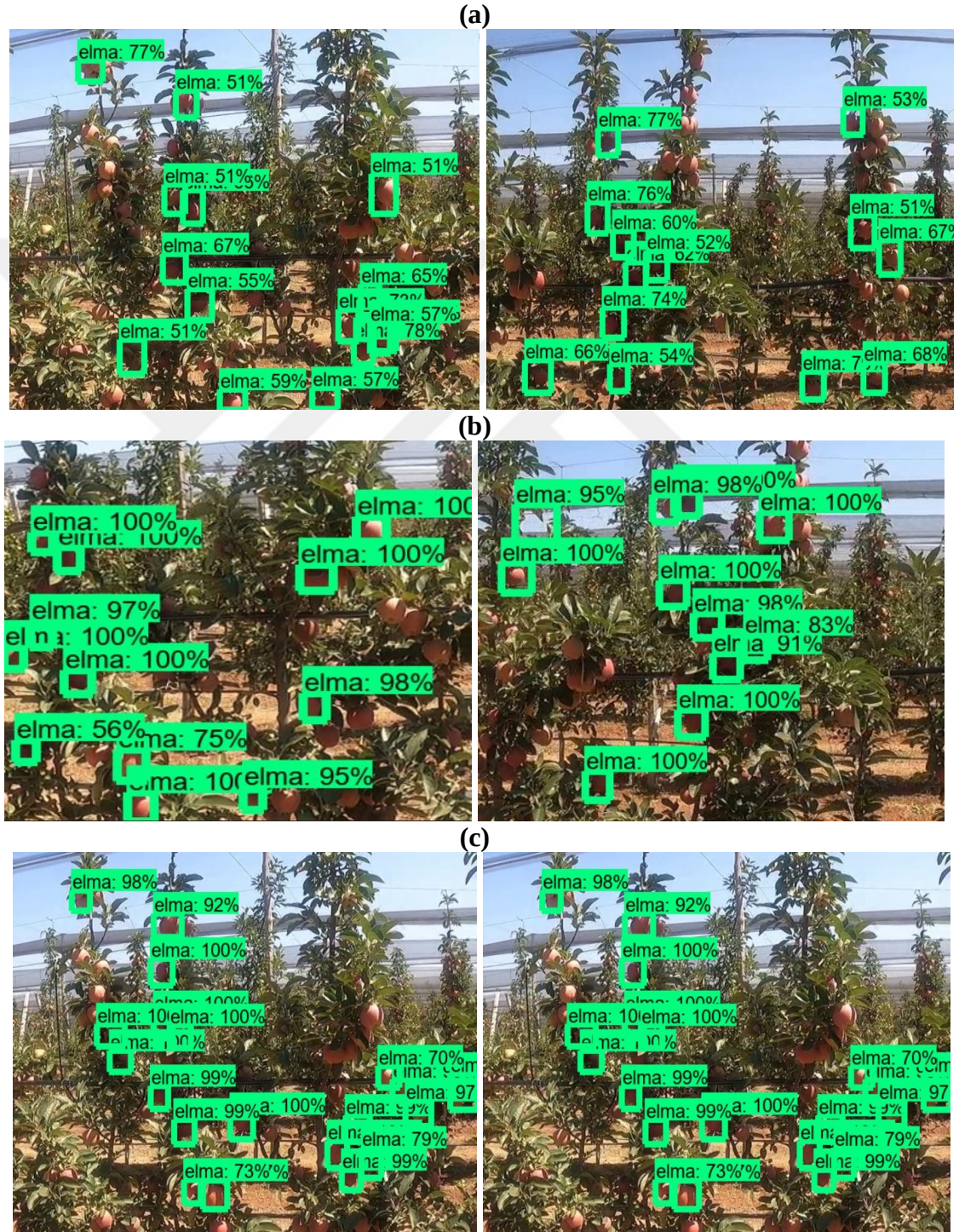
Üç farklı modelin eğitimi sonucunda elde edilen performans grafikleri Şekil 4.23'te sırasıyla öğrenme oranı-iterasyon grafiği, kayıp değeri-iterasyon grafiği ve saniyede işlenen adım sayısı-toplam iterasyon grafiği için gösterilmiştir.



Şekil 4.23. (a) Öğrenme Oranı-İterasyon grafiği, (b) Kayıp-İterasyon grafiği, (c) Saniyede İşlenen Adım Sayısı-Toplam İterasyon grafiği.

Şekil 4.23'te verilen grafik üç farklı modelin öğrenme oranı, kayıp fonksiyonları değişimlerini ve modelin eğitim sürecindeki işlem hızını göstermektedir. En iyi

performans, 0.052 kayıp değeri ile model3 üzerinden elde edilmiştir. Model2 0.126 kayıp değeri üretirken, model1 ise 0.195 kayıp değeri üretmiştir. Şekil 4.24'te de özgün veri setiyle eğitilen Faster-RCNN sinir ağ modelinin deneysel elma tespiti çalışmalarına ait görüntüleri sırasıyla üç farklı model için verilmiştir.



Şekil 4.24. Özgün veri setiyle eğitilen Faster- RCNN sinir ağ modelinin deneysel elma tespiti çalışmasına ait görüntüleri. (a) model1 ile, (b) model2 ile ve (c) model3.

Tablo 4.16.'da özgün olarak oluşturulan üç farklı model ile gerçekleştirilen elma tespit çalışmasına ait performans metrikleri sunulmuştur. Üç farklı model ile video görüntüleri üzerinde gerçekleştirilen elma tespit çalışmasına ait sonuçlar incelendiğinde daha düşük kayıp değeri elde edilen model3 ile daha yüksek doğrulukta elma tespit edildiği görülmektedir.

Tablo 4.16. Sinir ağ modellerinin elma tespit çalışmasına ait performans sonuçları.

Model	Sayılan Elma	Yanlış Sayılan Elma	Toplam Elma Sayısı	Doğruluk Oranı	F1 Skoru	Hassasiyet
model3	84	0	150	%56	0.69	0.72
model2	71	1	150	%47	0.66	0.69
model1	56	0	150	%37	0.64	0.67

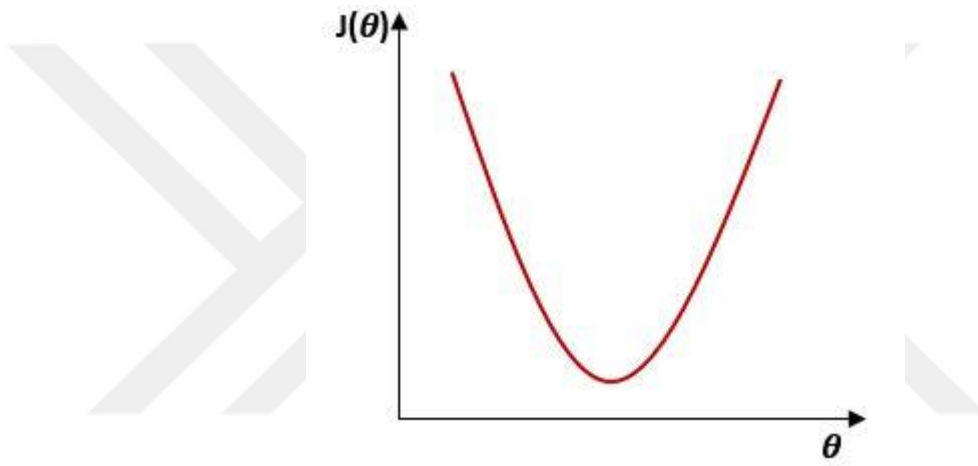
4.2.4.5. Hiperparametreleri Optimize Edilen SM SA Modeli ile Elma Tespiti

Hiperparametreler, eğitim aşamasında belirlenen ve makine öğrenmesi modellerindeki performansı etkileyen parametrelerdir. Bu nedenle, doğru hiperparametrelerin belirlenmesi modelin daha iyi performans göstermesi açısından son derece önemlidir. Geleneksel olarak hiperparametreler deneme yanılma yöntemiyle belirlenebildiği gibi, rastgele arama ve ızgara arama yöntemleriyle de daha sistematik bir hiperparametre optimizasyonu çalışmaları yapılarak bu parametreler belirlenebilmektedir. Fakat, bu yöntemlerin kullanılması oldukça fazla zaman almaktadır. Hiperparametreler ile ilgili yapılan çalışmalarda sonucu en çok etkileyen parametrenin öğrenme oranı olduğu gözlemlenmiştir. Bu sebeple gerçekleştirilen optimizasyon çalışması temel olarak öğrenme oranı üzerine yoğunlaştırılmıştır. Öğrenme oranı, makine öğrenme modelinin güncelleme adımlarında kullanılan bir parametredir. Öğrenme oranının artması, güncelleme adımlarının daha büyük olmasına neden olur. Bu, modelin daha hızlı öğrenmesini sağlarken aynı zamanda aşırı öğrenmeye ve kayıp değerinin yüksek olmasına sebebiyet verebilir. Öğrenme oranının düşürülmesi ise daha küçük güncelleme adımları oluşturur ve modelin daha yavaş öğrenmesine neden olur. Bu durum aşırı öğrenmeyi engelleyip kayıp değerinin azaltılmasını sağlayabilir. Fakat bu durumda da öğrenme çok yavaş ilerleyecektir. Sonuç olarak, öğrenme oranı ile kayıp değeri arasındaki ilişki karmaşıktır ve modelin ya da veri setinin özelliğine göre değişiklik gösterebilmektedir. Genellikle öğrenme oranı sabit bir değer almak yerine optimum

değeri bulmak için önce düşük değerden başlatıp kademeli olarak artırmaya dayalı stratejiler tercih edilmektedir. Şekil 4.25'te kayıp fonksiyonu – ağırlık grafiği ilişkisi verilmiştir. Denklem (4.1)'e göre oluşturulan grafikte her bir adımda ağırlık değeri güncellenerek yeni kayıp değeri hesaplanmaktadır:

$$\theta = \theta - \alpha \frac{\delta}{\delta \theta} J(\theta) \quad (4.1)$$

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2 \quad (4.2)$$



Şekil 4.25. Kayıp fonksiyonu-ağırlık ilişkisinin grafiksel gösterimi.

Burada, $J(\theta)$ kayıp fonksiyonunu, θ ağırlık değişkenini, α öğrenme oranını, $h_{\theta}(x^{(i)})$ tahminlenen değerini ve $y^{(i)}$ ise gerçek değeri ifade etmektedir.

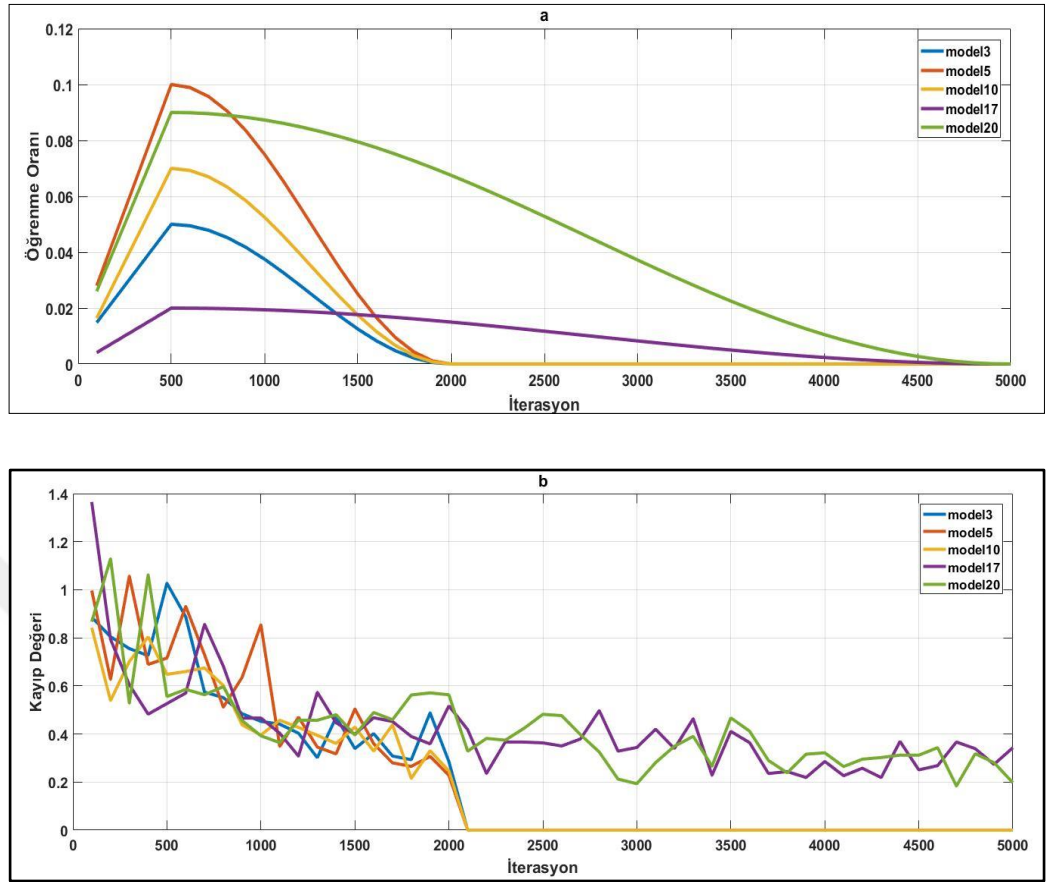
Hiperparametre optimizasyonu çalışması için önerilen yeni yaklaşımda, rastgele öğrenme parametreleri belirlenerek iki farklı iterasyon değeri ile toplam altmış farklı SSD-MobileNet derin sinir ağı modeli eğitilmiştir. Tablo 4.17'de altmış farklı modelin seçilen farklı hiperparametreler ile eğitilen yirmi ayrı modelin özellikleri verilmiştir. Hiperparametre olarak; başlangıç ve maksimum öğrenme oranları, batch boyutu ve iterasyon sayıları ele alınmıştır. Derin sinir ağ modellerinin eğitimi sonucunda elde edilen kayıp değerleri ve iterasyon sayıları giriş olarak kullanılarak hiperparametre optimizasyonu için yeni bir YSA modeli eğitilmiştir. Eğitim de sinir ağ modelinin çıkışı olarak da her bir modeli eğitirken kullanılan öğrenme oranı başlangıç ve maksimum

değerleri kullanılmıştır. Sonuç olarak hedeflenen kayıp değeri için optimum hiperparametrelerin tahminlenmesi amaçlanmaktadır.

Tablo 4.17. YSA modelinin eğitilmesi için kullanılan yirmi farklı SSD-MobileNet modeline ait hiperparametreler.

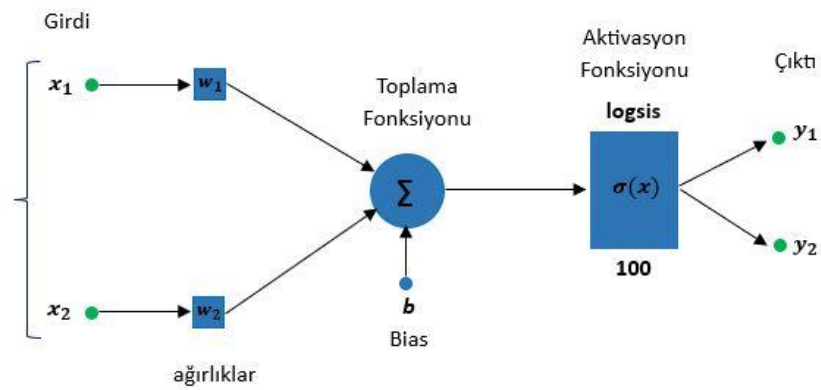
Parametre	Başlangıç Öğrenme Oranı (L0)	Maks. Öğrenme Oranı (Lr)	Batch Boyutu	İterasyon Sayısı	Kayıp Değeri
1. Eğitilen Model	0.001	0.01	8	2000	0.38
2. Eğitilen Model	0.005	0.01	8	2000	0.35
3. Eğitilen Model	0.006	0.05	8	2000	0.28
4. Eğitilen Model	0.008	0.09	8	2000	0.44
5. Eğitilen Model	0.01	0.1	8	2000	0.22
6. Eğitilen Model	0.05	0.5	8	2000	3
7. Eğitilen Model	0.1	0.9	8	2000	1.27
8. Eğitilen Model	0.04	0.16	8	2000	0.52
9. Eğitilen Model	0.0001	0.004	8	2000	0.48
10. Eğitilen Model	0.003	0.07	8	2000	0.24
11. Eğitilen Model	0.02	0.06	8	5000	0.36
12. Eğitilen Model	0.016	0.08	8	5000	0.29
13. Eğitilen Model	0.02	0.1	8	5000	0.49
14. Eğitilen Model	0.04	0.2	8	5000	0.48
15. Eğitilen Model	0.006	0.05	8	5000	0.36
16. Eğitilen Model	0.006	0.03	8	5000	0.31
17. Eğitilen Model	0.0001	0.02	8	5000	0.27
18. Eğitilen Model	0.001	0.01	8	5000	0.40
19. Eğitilen Model	0.005	0.05	8	5000	0.28
20. Eğitilen Model	0.01	0.09	8	5000	0.19

Tablo 4.17’de verilen hiperparametreler ile gerçekleştirilen farklı modellere ait eğitimler sonucunda en düşük kayıp değerine sahip beş modelin performans grafikleri Şekil 4.26’da sırasıyla öğrenme oranı-iterasyon grafiği ve kayıp değeri-iterasyon grafiği için gösterilmiştir.



Şekil 4.26. (a) Öğrenme Oranı-İterasyon grafiği, (b) Kayıp-İterasyon grafiği.

Eğitimlerde kullanılan hiperparametre değerleri ile Şekil 4.27’de şematik olarak gösterilen iki giriş (kayıp değeri ve iterasyon sayısı) iki çıkışlı (öğrenme oranı maks. ve min. değerleri) FFBNP YSA modeli, Scaled Conjugate Gradient algoritması kullanılarak eğitilmiştir.



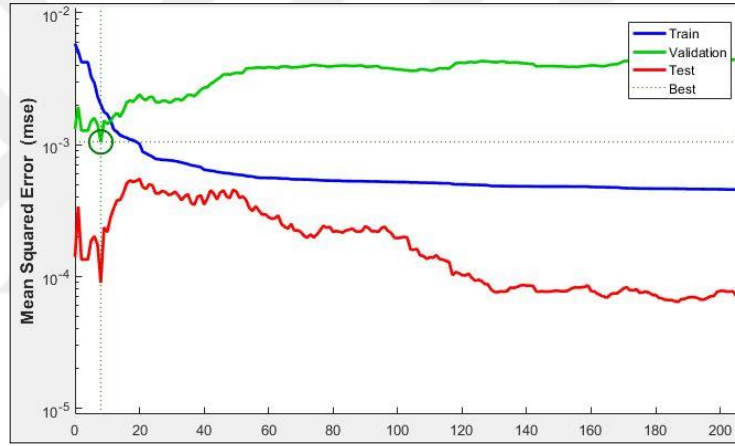
Şekil 4.27. Hiperparametre tahminlemek amacıyla oluşturulan FFBNP YSA modelinin şematik gösterimi.

Eğitim parametreleri Tablo 4.18’de verilmiştir. Şekil 4.28’de ise FFBNP YSA modelinin eğitim sonucundaki performans grafiği ve regresyon grafiği verilmiştir.

Tablo 4.18. Hiperparametre tahminleyen FFBNP YSA modeli için eğitim parametreleri.

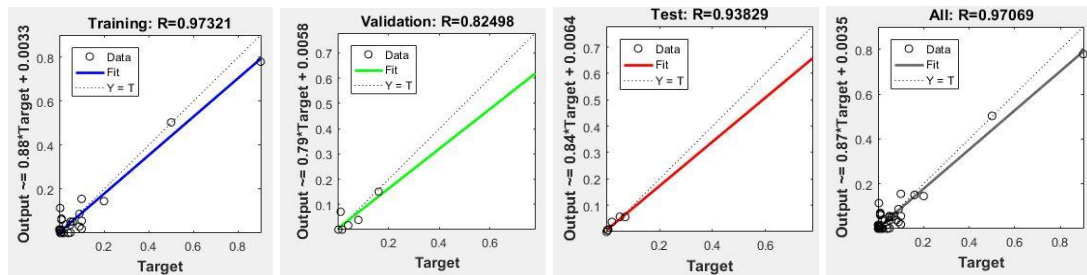
Performans Fonksiyonu	MSE
Maksimum İterasyon Sayısı (Epochs)	1000
Gizli Katman Nöron Sayısı (H)	100
Minimum Gradient	1e-06
Sonuç Gösterme İterasyon Sayısı	25

(a)



Epoch

(b)



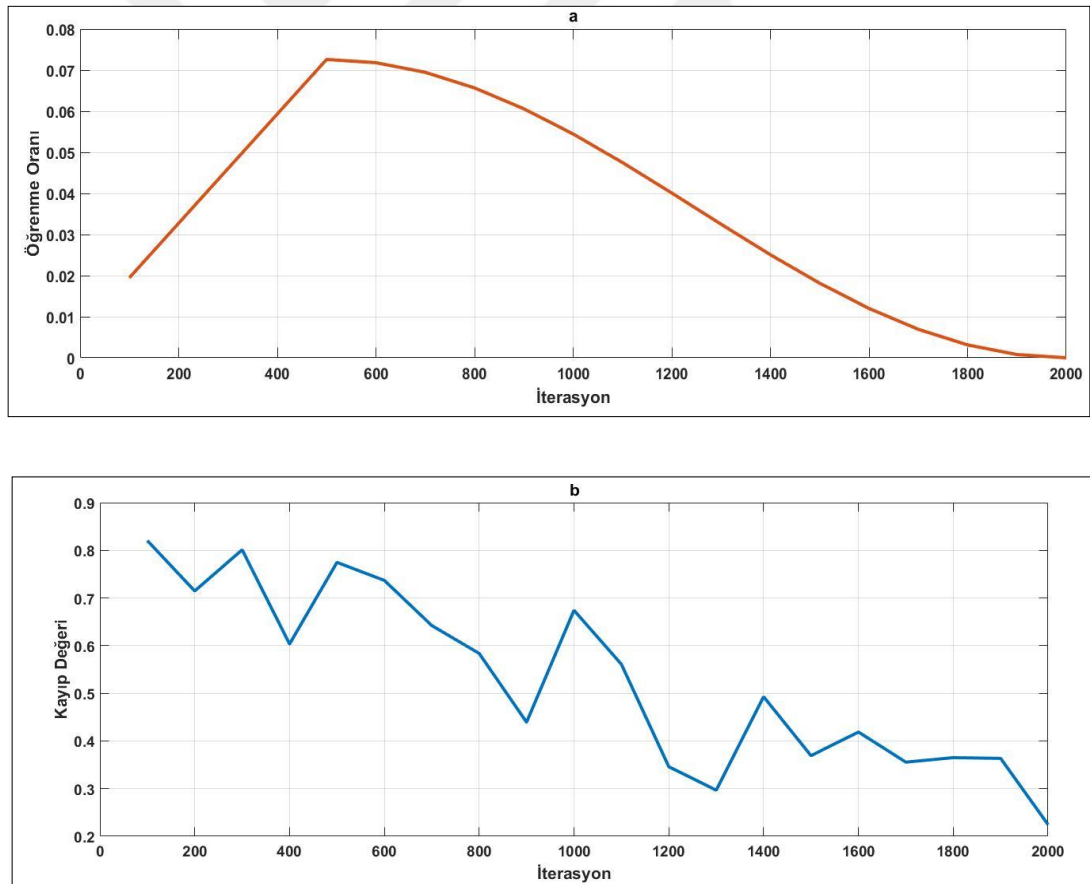
Şekil 4.28. YSA modelin eğitim sonucundaki; (a) performans grafiği, (b) regresyon grafiği.

Eğitilen YSA modeline, optimum hiperparametre değerlerini belirlemek için ideal değer olarak belirlenen kayıp ve iterasyon değerleri girdi olarak verilerek öğrenme oranı parametreleri tahminlenmiştir. Girilen ve tahminlenen parametre değerleri Tablo 4.19’da sunulmuştur.

Tablo 4.19. FFBNP YSA modeli ile istenilen başarı için, girilen ve tahminlenen parametre değerleri.

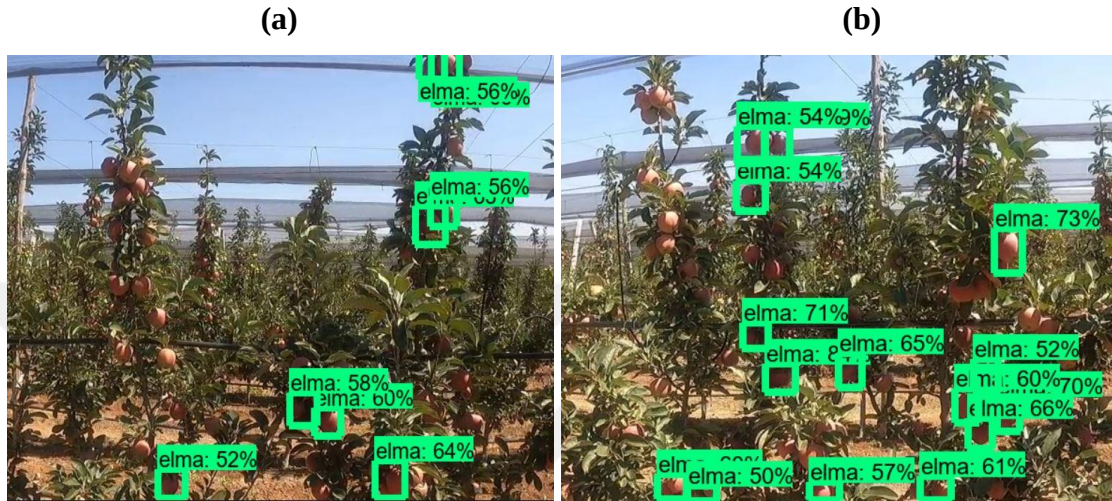
Girdi		Çıktı	
Kayıp Değeri	İterasyon Sayısı	Öğrenme Oranı- Başlangıç Değeri	Öğrenme Oranı- Maks. Değer
0.01	2000	0.0062	0.0726

Önerilen FFBNP YSA tabanlı yaklaşım, 2000 iterasyonda 0.01 kayıp değerine ulaşmak için öğrenme oranının tahminlenmesini gerçekleştirmiştir. Elde edilen öğrenme oranı SSD-MobileNet derin sinir ağ modelinin yeniden eğitiminde kullanılmıştır. Eğitim sonucunda elde edilen performans grafikleri Şekil 4.29’da sırasıyla öğrenme oranı-iterasyon grafiği ve kayıp değeri-iterasyon grafiği olarak gösterilmiştir.



Şekil 4.29. Optimize edilen hiperparametre değerlerine göre eğitilen SSD-MobileNet sinir ağ modelinin (a) Öğrenme Oranı-İterasyon grafiği, (b) Kayıp-İterasyon grafiği.

Şekil 4.30’da sırasıyla özgün veri setiyle eğitilen SSD-MobileNet derin sinir ağ modeli ile hiperparametrelerin optimize edildiği bu yeni yaklaşım ile eğitilen SSD-MobileNet derin sinir ağ modellerinin video üzerinde elma tespit performansları görsel üzerinde kıyaslanmıştır.



Şekil 4.30. (a) Özgün veri setiyle eğitilen model ile, (b) hiperparametreleri optimize edilerek özgün veri setiyle eğitilen modelin, elma tespit performansları kıyaslanması.

Tablo 4.20’de optimize edilen hiperparametre değerleriyle eğitilen derin sinir ağ modeliyle elma tespitine yönelik performans metrikler sunulmuştur.

Tablo 4.20. Optimizasyon edilen SSD-MobileNet derin sinir ağ modeli ile elma tespit çalışmasına ait performans sonuçları.

Sayılan Elma	Yanlış Sayılan Elma	Toplam Elma Sayısı	Doğruluk Oranı	F1 Skoru	Kayıp Değeri
60	1	150	%43	0.72	0.22

4.2.4.6. Hiperparametreleri Optimize Edilen FRC SA Modeli ile Elma Tespiti

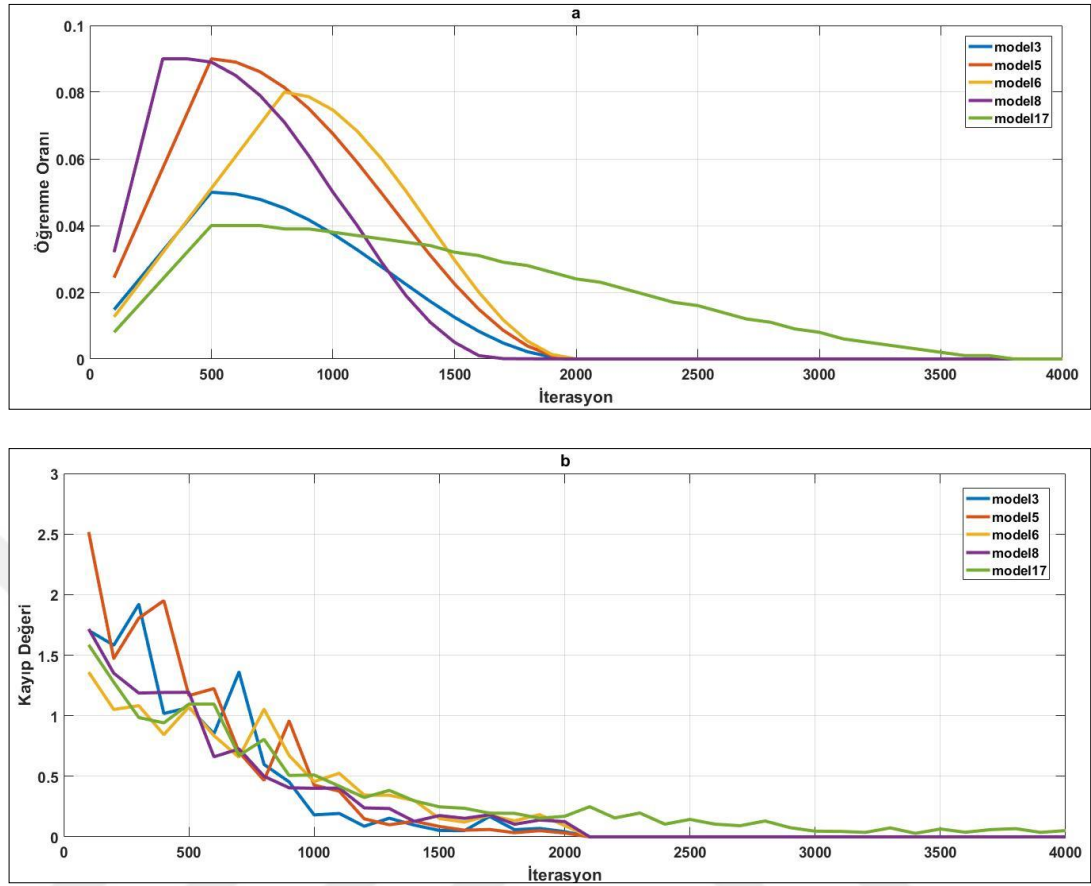
Tablo 4.21’de farklı hiperparametreler ile eğitilen kırk ayrı Faster-RCNN modelinden seçilen yirmi tanesinin özellikleri verilmiştir. Hiperparametre olarak başlangıç ve maksimum öğrenme oranları, batch boyutu ve iterasyon sayıları ele alınmıştır. Derin sinir ağ modellerinin eğitimi sonucunda elde edilen kayıp değerleri ve iterasyon sayısı giriş olarak kullanılarak yapay sinir ağ modeli eğitilmiştir. Eğitim sırasında sinir ağ modelinin çıkışı olarak da her bir derin sinir ağ modelinin eğitiminde kullanılan öğrenme oranı

başlangıç ve maksimum değerleri kullanılmıştır. Sonuç olarak hedeflenen kayıp değeri için optimum hiperparametrelerin tahminlenmesi amaçlanmıştır.

Tablo 4.21. YSA modelinin eğitilmesi için kullanılan yirmi farklı Faster R-CNN modeline ait hiperparametreler.

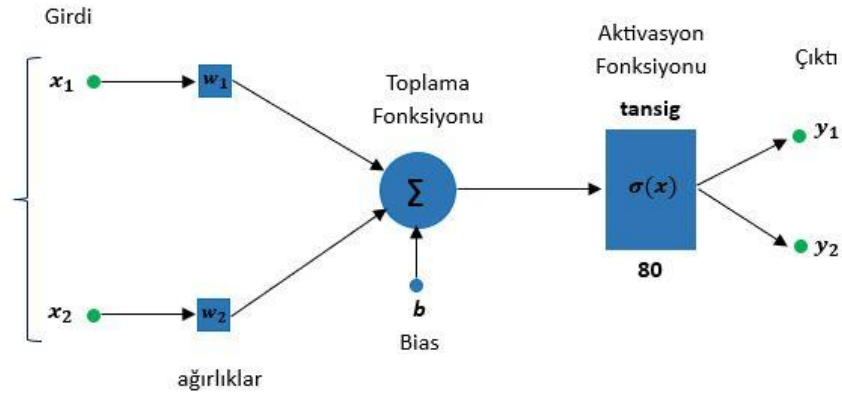
Parametre	Başlangıç Öğrenme Oranı (L0)	Maks. Öğrenme Oranı (Lr)	Batch Boyutu	İterasyon Sayısı	Kayıp Değeri
1. Eğitilen Model	0.001	0.01	8	2000	0.138
2. Eğitilen Model	0.005	0.01	8	2000	0.135
3. Eğitilen Model	0.006	0.05	8	2000	0.043
4. Eğitilen Model	0.04	0.16	8	2000	0.25
5. Eğitilen Model	0.008	0.09	8	2000	0.032
6. Eğitilen Model	0.003	0.08	8	2000	0.093
7. Eğitilen Model	0.003	0.07	8	2000	0.19
8. Eğitilen Model	0.003	0.09	8	2000	0.126
9. Eğitilen Model	0.02	0.04	8	2000	0.162
10. Eğitilen Model	0.02	0.06	8	4000	0.158
11. Eğitilen Model	0.016	0.08	8	4000	0.141
12. Eğitilen Model	0.02	0.1	8	4000	0.149
13. Eğitilen Model	0.04	0.2	8	4000	0.148
14. Eğitilen Model	0.006	0.05	8	4000	0.136
15. Eğitilen Model	0.006	0.03	8	4000	0.141
16. Eğitilen Model	0.001	0.06	8	4000	0.135
17. Eğitilen Model	0.0001	0.04	8	4000	0.052
18. Eğitilen Model	0.005	0.05	8	4000	0.137
19. Eğitilen Model	0.01	0.09	8	4000	0.153
20. Eğitilen Model	0.003	0.07	8	4000	0.128

Tablo 4.21’de verilen hiperparametreler ile gerçekleştirilen farklı modellere ait eğitimler sonucunda elde edilen en optimum değere sahip beş modelin performans grafikleri Şekil 4.31’de sırasıyla öğrenme oranı-iterasyon grafiği ve kayıp değeri-iterasyon grafiği olarak verilmiştir.



Şekil 4.31. (a) Öğrenme Oranı-İterasyon grafiği, (b) Kayıp-İterasyon grafiği.

Eğitimlerde kullanılan hiperparametre değerleri ile Şekil 4.32’de şematik olarak gösterilen iki giriş (kayıp değeri ve iterasyon sayısı) iki çıkışlı (öğrenme oranı maks. ve min. değerleri) FFBPN YSA modeli, Scaled Conjugate Gradient algoritması kullanılarak eğitilmiştir.

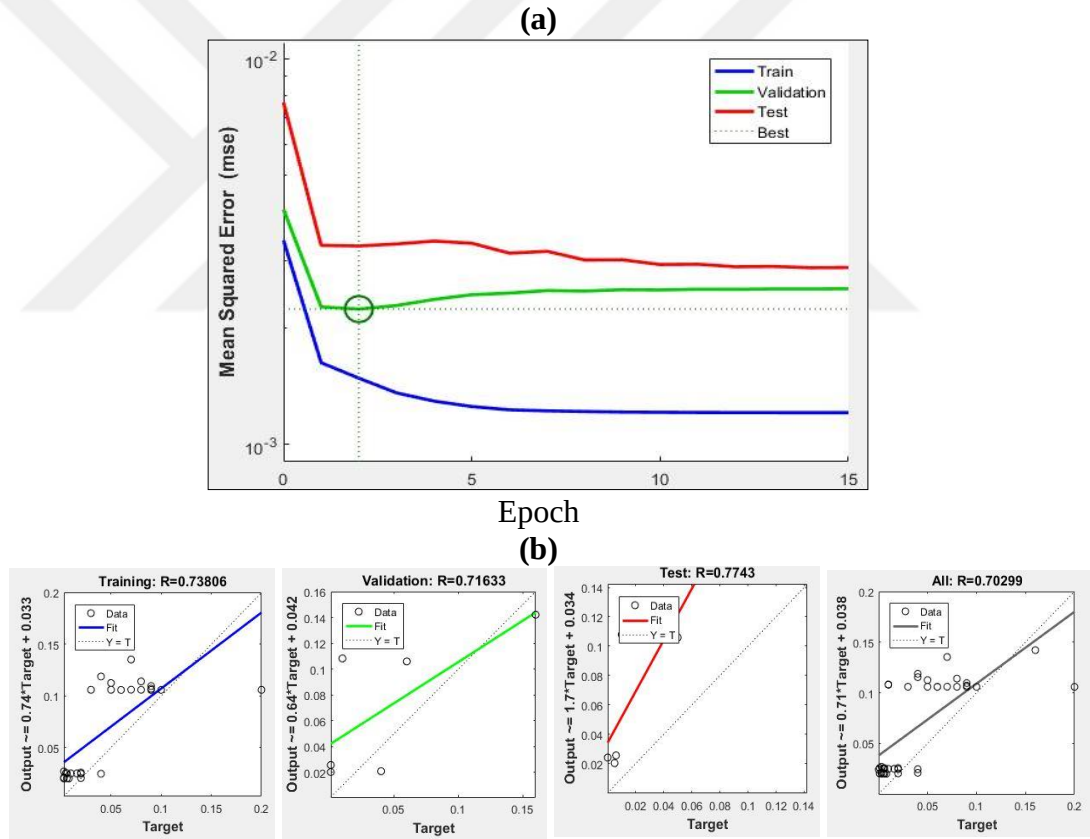


Şekil 4.32. Hiperparametre tahminlemek amacıyla oluşturulan FFBPN YSA modelinin şematik gösterimi.

FFBPN YSA modeli için kullanılan eğitim parametreleri Tablo 4.22’de verilmiştir. FFBPN YSA modelinin eğitilmesinden sonra elde edilen performans ve regresyon grafikleri Şekil 4.33’te verilmiştir.

Tablo 4.22. Hiperparametre tahminleyen FFBPN YSA modeli için eğitim parametreleri.

Performans Fonksiyonu	MSE
Maksimum İterasyon Sayısı (Epochs)	750
Gizli Katman Nöron Sayısı (H)	80
Minimum Gradient	1e-06
Sonuç Gösterme İterasyon Sayısı	25



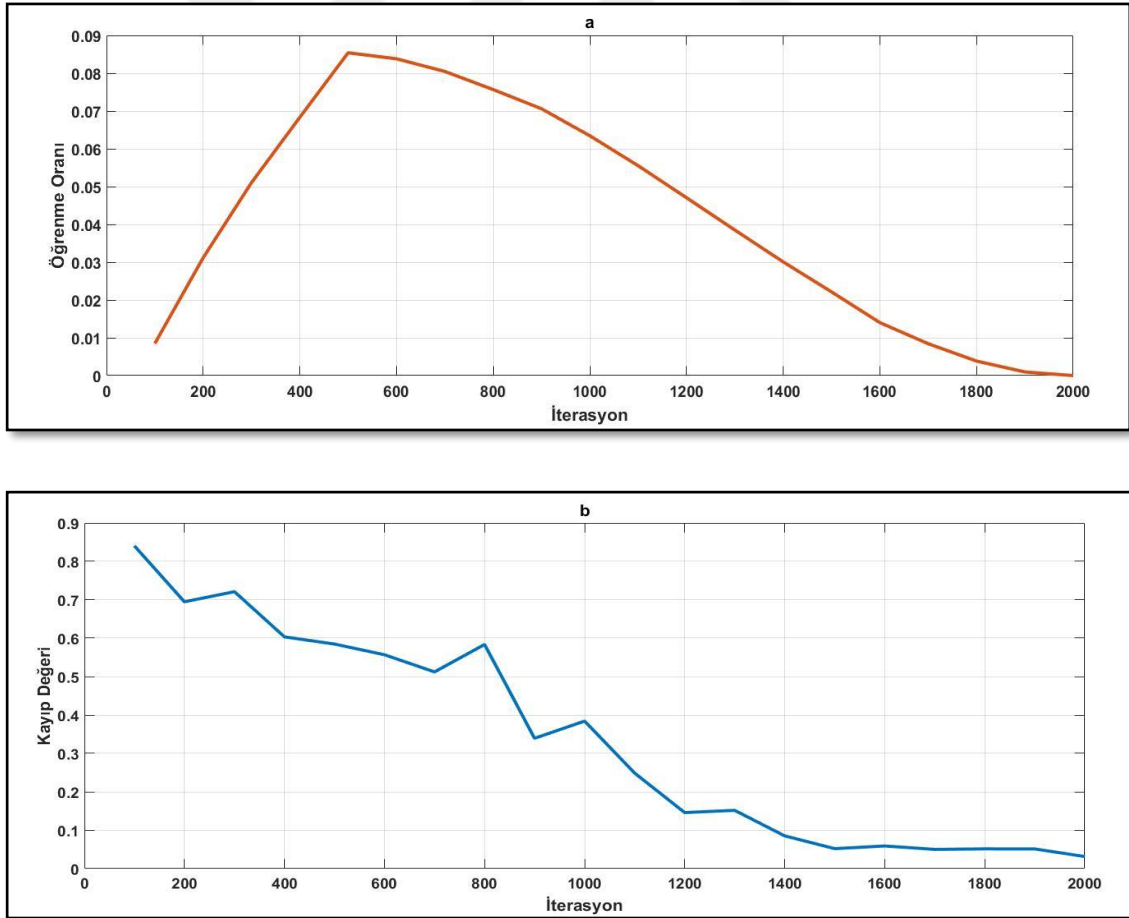
Şekil 4.33. FFBPN YSA modelinin eğitimi sonucundaki; (a) performans grafiği, (b) regresyon grafiği.

Eğitilen FFBPN YSA modelinde, optimum hiperparametre değerlerini belirlemek amacıyla ideal değer olarak belirlenen kayıp ve iterasyon değerleri girdi olarak verilmiş ve öğrenme oranı parametreleri tahminlenmiştir. Girilen ve tahminlenen parametre değerleri Tablo 4.23’te sunulmuştur.

Tablo 4.23. İstenilen başarı değeri için eğitilen FFBPN YSA modelinde girilen ve tahminlenen parametre değerleri.

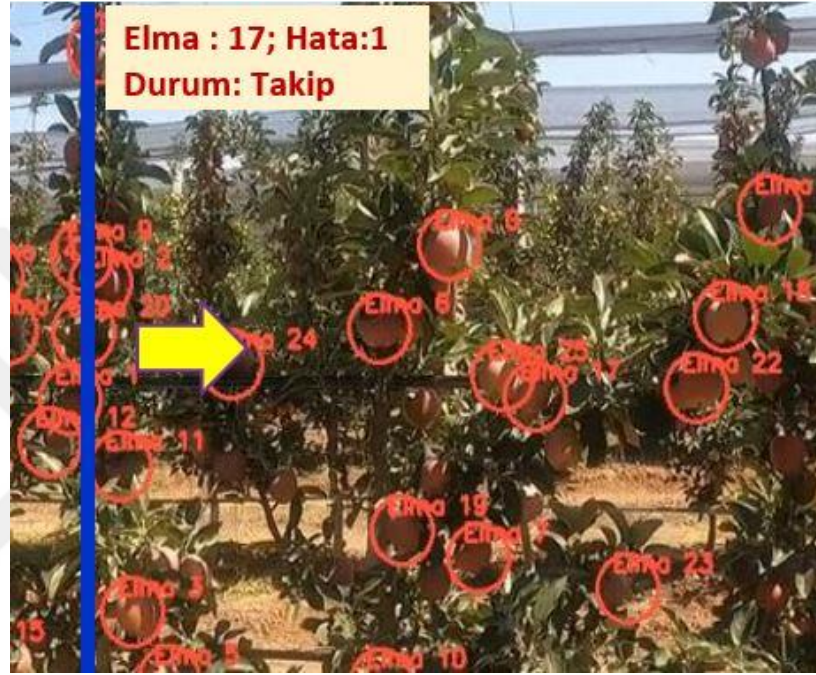
Girdi		Çıktı	
Kayıp Değeri	İterasyon Sayısı	Öğrenme Oranı-Başlangıç Değeri	Öğrenme Oranı-Maks. Değer
0.01	2000	0.0027	0.0854

Önerilen FFBPN YSA tabanlı yaklaşım, 2000 iterasyonda 0.01 kayıp değerine ulaşmak için öğrenme oranının tahminlenmesini gerçekleştirmiştir. Optimum öğrenme oranı değeri elde edilmiştir. Bu değer Faster-RCNN derin sinir ağ modelinin yeniden eğitiminde kullanılmıştır. Eğitim sonucunda elde edilen performans grafikleri Şekil 4.34'te sırasıyla öğrenme oranı-iterasyon grafiği ve kayıp değeri-iterasyon grafiği olarak verilmiştir.



Şekil 4.34. Optimize edilen öğrenme oranı ile eğitilen FRC sinir ağ modelinin; (a) Öğrenme Oranı-İterasyon grafiği, (b) Kayıp-İterasyon grafiği.

hesaplanan değere eklenmesi gerekmektedir. Bu amaçla opencv kütüphanesinde bulunan nesne takip algoritması kullanılmıştır. Değişen her bir karede bu algoritmaya göre elmalar takip edilmiş ve görüntüye yeni eklenen elmalar tespit edilmiştir. Şekil 4.36'da gösterildiği gibi ekranın soluna yerleştirilen sanal dikey çizgiyi geçen elmaların sayıcı tarafından sayılması şeklinde mantıksal bir yapı oluşturulmuştur.



Şekil 4.36. Tespit edilen elmaların geliştirilen yazılım ile sayılmasının gösterimi.

Sonuçta, deneysel olarak gözlemlenen 10 ağaçta toplam 150 elma sayılması gerekirken; daha iyi sonuç veren Faster-RCNN ile eğitilen model kullanılarak 84, modelin hiperparametreleri optimize edilince 112 elma sayılabilmektedir. Hazır modeller ile gerçekleştirilen çalışmalarda 10-25 elma arasında sayım yapıldığı ve dinamik bir görüntü üzerinde deneysel olarak bu işlemin gerçekleştirildiği düşünüldüğünde geliştirilen derin sinir ağ modelleri ile başarılı bir sonuç elde edildiği ortadadır. Ayrıca farklı sayıda ağaç ile gerçekleştirilen sayma işleminde hata oranları birbirine oldukça yakındır. Fakat rekoltenin tahmini için gerçek uygulamalarda %75'lik bir doğruluk oranı tam olarak yeterli olmamaktadır. Bunun için elde edilen bu belirli bir metoda dayalı sayma işleminde tahminlenen sonuçlar iki farklı yöntem ile ele alınmıştır. Gözlemlenen önemli bir durum; farklı sayılarda ağaç ile gerçekleştirilen sayma işleminde hata oranı hemen hemen hep aynı çıkmaktadır. Bu sebeple ilk yöntemde, elde edilen sonuç aşağıda oluşturulan

denklem (4.3) ile belirli bir katsayıyla çarpılarak rekoltenin elde edilmesine dayanmaktadır.

$$N_{toplaml} = N_{tespit} \times k \quad (4.3)$$

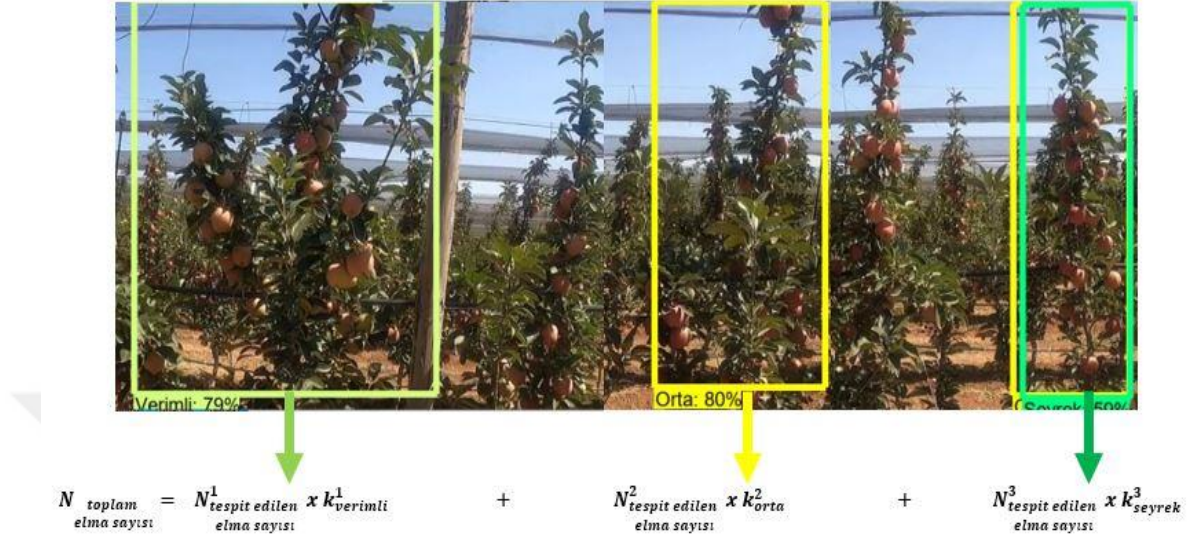
Burada; $N_{toplaml}$ hesaplanan toplam elma sayısını, N_{tespit} derin öğrenme modeli destekli geliştirilen elma sayma yazılımı ile tespit edilen elma sayısını, k ise sonucu düzeltmek için kullanılan bir katsayıyı temsil etmektedir. Bu çalışma için $k=1,25$ olarak belirlenmiştir. Denklem (4.3)'teki hesaplama göre en optimum parametreler ile gerçekleştirilen elma sayma uygulamasında tespit edilen 112 elma belirlenen katsayı ile çarpıldığında 140 elma sayısı belirlenmiş olacaktır. Bu da tekrarlanan deneylerde rekoltenin ortalama %7 hata ile hesaplanabildiğini göstermektedir.

Rekoltenin belirlenmesi için sunulan ikinci yöntemde ise yeni ve daha güvenilir bir yaklaşım önerilmektedir. Elde edilen verilere göre ağaçlardaki elmaların sayısı ağaç yapısına bağlı olarak birbirine yakın değerler içermektedir. Örneğin; bir ağaçta hiç elma yokken diğer ağaç eğer sağlıklı görünüyorsa 20-25 elmaya sahiptir. Ya da yaprak yoğunluğu daha az olan bir ağaçta 15 civarı elma bulunmaktadır. Buna göre, ağaçlar üç farklı kategoriye ayrılmıştır. Bu kategoriler; verimli, orta verimli ve seyrek olmak üzere ağaç üzerinde bulunan elmaların durumunu ifade etmektedir. Bir ağaçtaki elma sayısı 10'dan az ise seyrek, 10 ile 18 arası orta verimli ve 18'den fazla ise verimli olarak kabul edilmektedir. Çalışılan koridorda bir ağaçtaki maksimum elma sayısı 25'tir. Bu yöntemde ağaçtaki elmalar derin ağ modeli tarafından sayıldıktan sonra birinci yöntemdeki gibi sabit bir katsayı ile değil de tahminleyici tarafından belirlenen verimlilik kategorisine göre bir katsayıyla çarpılmaktadır. Bu şekilde elde edilen sonuçlar göre %97'ye varan doğruluk oranları elde edilmiştir. Denklem (4.4)'te bu yaklaşım için üretilen formül verilmiştir:

$$N_{toplaml} = \sum_{i=1}^n N_{tespit}^i \times k_{sınıf}^i \quad (4.4)$$

Burada; $N_{toplaml}$ hesaplanan toplam elma sayısını, N_{tespit}^i derin öğrenme modeli destekli geliştirilen elma sayma yazılımı ile i . ağaçta tespit edilen elma sayısını, $k_{sınıf}^i$ ise i . ağacın sınıfına göre belirlenen katsayı değerini temsil etmektedir. Seyrek sınıf ağaç için $k=1,1$, orta verimli sınıftaki ağaç için $k=1,2$, verimli sınıf olarak belirlenen ağaç için ise $k=1,4$

olarak belirlenmiştir. Şekil 4.37’de, önerilen sayma yönteminin temel hesaplama mantığı görsel olarak sunulmuştur.



Şekil 4.37. Ağaçların sınıflandırılmasını esas alan önerilen yaklaşımın temel hesaplama mantığının gösterimi.

Aşağıda sunulan Tablo 4.25’de çalışma boyunca uygulanan tüm metotlar ile gerçekleştirilen elma sayma uygulamalarının performansları kıyaslanmıştır.

Tablo 4.25. Uygulanan metotların rekolte belirleme başarılarının kıyaslamalı gösterimi.

Uygulanan Yöntem	Sayılan Elma	Yanlış Sayılan Elma	Toplam Elma Sayısı	Doğruluk Oranı
MS COCO ile eğitilen SM SA modeli ile elma sayımı	12	6	150	%8
MS COCO ile eğitilen FRC SA modeli ile elma sayımı	27	4	150	%18
Özgün veri seti ile eğitilen SM SA modeli ile elma sayımı	34	2	150	%23
Özgün veri seti ile eğitilen FRC SA modeli ile elma sayımı	84	0	150	%56
Optimize edilen SM SA modeli ile elma sayımı	60	1	150	%43
Optimize edilen FRC SA modeli ile elma sayımı	112	0	150	%75
Önerilen; elma yoğunluğunu da esas alan yaklaşım ile elma sayımı- FRC SA	145	0	150	%97

4.2.6. Hastalıklı Elmaların Tespiti

Rekolte belirleme çalışmaları için geliştirilen robotik sistem ve nesne tespitinin gerçekleştirildiği derin sinir ağı modeli ile elma bahçesinde hastalık tespit çalışmaları gerçekleştirilmiştir. Hastalıklı elmaların görüntüleri kullanılarak oluşturulan veri seti ile Faster-RCNN tabanlı yeni bir derin sinir ağı modeli eğitilmiştir. Gerçekleştirilen uygulama sonucunda, robotik sistem ile elde edilen görüntü üzerinde tespit edilen hastalıklı elmalar Şekil 4.38’de gösterilmiştir.



Şekil 4.38. Geliştirilen nesne tespit sistemi ile yapısal bozukluğu olan elmaların tespit edilmesine yönelik gerçekleştirilen uygulamalara ait görüntüler.

Sonuç olarak kısıtlı veri ile gerçekleştirilen hastalıklı elma belirleme çalışma neticesinde %75 doğrulukta hastalık tespiti başarısına ulaşılmıştır. Bu alanda gerçekleştirilecek iyileştirmeler ile hastalıkların erken teşhisi ve hassas ilaçlama yapılması mümkün olacaktır.

5. BÖLÜM

SONUÇLAR VE ÖNERİLER

Bu tez çalışmasında, gelişen endüstriyel uygulamaların desteği ile özellikle meyve bahçelerinde rekolte belirlenmesinde manuel sayma işlemlerindeki büyük problemleri çözmeye yönelik yapay zeka ve bilgisayarlı görü yöntemleri ile simülasyon ve deneysel çalışmalar gerçekleştirilmiştir.

Gerçekleştirilen çalışma; uçan robotik sistem tasarımı ve imalatı, otonom yörünge takibi, elma tespiti için derin sinir ağ modelleri geliştirilmesi ve elmaların sayılması olmak üzere dört ana başlık üzerinde yürütülmüştür. Tasarlanan, üretilen ve test edilen URS yapay zeka destekli görüntü işleme sistemi, kontrol ünitesi ve sensörlerden oluşmaktadır. Deneysel çalışmalara başlanmadan önce Rviz ve Gazebo ortamlarında robotik sistemin davranışı incelenmiştir. Lidar sensör ile EKVH yöntemi kullanılarak robotik sistemin izlediği yörünge tespit edilip kontrol parametrelerinin ayarlanması gerçekleştirilmiştir.

Deneysel çalışmalara ilk olarak Erciyes Üniversitesi Spor Bilimleri Fakültesi'nin önünde bulunan çim sahada başlanmıştır ve ilk otonom uçuş testleri bu sahada gerçekleştirilmiştir. Daha sonra 2021 yılı Temmuz ayı itibarıyla Kayseri'nin Yeşilhisar ilçesinde bulunan elma bahçesinde düzenli deneysel çalışmalara başlanmıştır ve 2022 yılı Kasım ayına kadar bu çalışmalar devam etmiştir. Bu süreçte robotik sistem ile bahçeden veriler toplanarak derin öğrenme yöntemiyle elmaları tespit eden modelin eğitimi için veri seti oluşturulmuştur. Bu veri seti kullanılarak, SSD-MobileNet ve Faster-RCNN ESA modelleri eğitilmiş ve bu modellerin performansları kıyaslanmıştır. Ayrıca özgün olarak oluşturulan veri seti kullanılarak eğitilen ESA modelleri ile, MS COCO veri setiyle eğitilen hazır ESA modellerinin performansları da karşılaştırmalı olarak incelenmiştir. Neticede elde edilen sonuçların, başarı oranını %18'lerden %75'lere önemli oranda artırdığı ve gelecek çalışmalar adına umut verici olduğu görülmüştür.

Ancak elde edilen rekolte verilerindeki hata oranının, gerçek uygulamalar için otomatik sayma sistemi üzerindeki amaçlanan performans değerlerini tam olarak sağlamadığı görülmüştür. Bu doğrultuda, yeni bir yaklaşım önerilmiştir. Bu yaklaşıma göre ikinci bir derin sinir ağ modeli geliştirilerek ağaçların elma durumlarına göre seyrek, orta verimli ve verimli olarak sınıflandırılması gerçekleştirilmiştir. Böylece, ağaçta sayılan elmaların ağacın sınıfına göre bir düzeltme katsayısıyla çarpılarak doğruluk oranının %97'lere kadar artırılabilceği gösterilmiştir.

Son olarak, aynı adımlar takip edilerek geliştirilen yeni bir sinir ağı modeli ile hastalıklı elmaların belirlenmesi üzerine bir deneysel çalışma daha gerçekleştirilmiştir. Kısıtlı veriler ile gerçekleştirilen bu uygulama ile amaç sonraki çalışmalar için otonom bir sistem ile hastalıkların erken teşhis edilebilmesi ve ağaca özgü hassas ilaçlamanın gerçekleştirilmesinin mümkün olduğunu göstermekti. Sonuçlar gelecek çalışmalar için teşvik edici bulunmuştur.

Bu çalışmalar ile elde edilen neticeleri listeleyecek olursak:

- Simülasyon sonuçları, YSA ile belirlenen PID kontrol kazanç parametreleri (K_p , K_i ve K_d) değerlerinin kullanılarak gerçekleştirilen uçuşlarda yörünge takibinin daha yüksek doğrulukta olduğunu göstermiştir.
- ESA ile gerçekleştirilen nesne tespiti uygulamalarında; eğitimde kullanılan elma veri seti, hiperparametreler ve tercih edilen ESA mimarisi sinir ağ modelinin başarı performansını önemli ölçüde etkilemektedir. Deneysel çalışmalar elma tespiti için oluşturulan özgün veri setinin başarı oranını beş kata kadar artırdığı ispat etmiştir.
- Deneysel çalışmalar, iki farklı ESA yaklaşımı kullanılarak gerçekleştirilmiştir. Bunlar; SSD-MobileNet ve Faster-RCNN yaklaşımlarıdır. Farklı hiperparametreler kullanılarak gerçekleştirilen eğitim sonucundaki performans verileri bu çalışma için Faster-RCNN yaklaşımının çok daha yüksek doğrulukta olduğunu göstermektedir. Faster-RCNN, SSD-MobileNet'e göre daha yavaş olmasına rağmen yüksek doğrulukta çalışmaktadır. Hız ile ilgili ortaya çıkan dezavantaj, görüntülerin işlenip rekolte belirlenmesi aşamasının yer istasyonunda gerçekleştirilmesi ile ortadan kaldırılmıştır.

- Geliştirilen ESA modellerinin elmaları tespit etmedeki başarısı deneysel çalışmalar süresince sürekli olarak geliştirilmeye çalışılmıştır. Bu amaçla, önemli bir etken olan eğitimde kullanılan hiperparametrelerin optimizasyonu için de yeni bir yaklaşım önerilmiştir. Önerilen yaklaşıma göre; YSA ile hiperparametrelerin optimizasyonu gerçekleştirilmiştir. Bu YSA modeli ile optimum öğrenme oranı belirlenen iterasyon sayısı ve kayıp değerine göre tahminlenmiştir. Bu optimum öğrenme oranı ile eğitilen Faster-RCNN derin sinir ağ modelinin başarı performansını %33 arttırdığı belirlenmiştir.
- Derin sinir ağ modeliyle gerçekleştirilen rekolte belirleme çalışması; zorlayıcı şartlar altında robotik bir sistem kullanılarak hareketli görüntüler üzerinde elmaların sayılabileceğini gösteren önemli bir çalışma olmuştur. Fakat optimum performanstaki derin sinir ağ modelinin performansının %75’ler seviyesinde kalması sebebiyle gerçek bir uygulamada kullanılabilmesi için birtakım geliştirmelere ihtiyaç duyulduğu ortaya çıkmıştır. Bu sebeple, ağaçları elma yoğunluklarına göre sınıflandıran ikinci bir derin sinir ağ modeli eğitilmiştir ve her iki model ile elde edilen sonuçların matematiksel bir ifadeyle rekolteyi hesapladığı yeni bir yaklaşım önerilmiştir. Bu sayede, rekoltenin doğruluk oranı %97’lere çıkarılmıştır.

Sonuç olarak URS tasarımı, önerilen kontrolör parametresi belirleme yaklaşımı, özgün veri setiyle gerçek ortamda gerçekleştirilen elma tespit çalışması, hiperparametre belirleme yaklaşımı ve önerilen elma sayma yaklaşımı ile yapay zeka destekli robotik bir sistemin otonom olarak rekolte belirlemesi detaylı olarak ortaya konmuştur. Önerilen kontrolör parametresi belirleme yaklaşımının, daha dengeli bir yörünge ortaya koyduğu görülmüştür. Ayrıca yapay sinir ağı ile hiperparametrelerin optimal bir şekilde seçilebileceği ve bu konuda da RBNN yapay sinir ağının oldukça başarılı olduğu ortaya çıkmıştır. Gelecek çalışmalar için yeni bir sayma yöntemi önerilmiş olup, ESA yaklaşımında nesne belirleme ve sayma uygulamaları için farklı çalışma alanları uygulamaya geçirilecektir.

Bu çalışma, çiftçiler için önemli bir sorun olan ve uzun süreler alan rekolte belirlenmesi işleminin otomatik olarak robotik bir sistem ile gerçekleştirilebileceğini

ispat etmiştir. Geliştirme sürecinin devam etmesiyle bu çalışmanın ticari ürüne dönüştürölme potansiyeli bulunmaktadır.



KAYNAKÇA

1. Worldometers. <https://www.worldometers.info/world-population/world-population-by-year/>, (Eriřim tarihi: Ağustos 2022).
2. COM(2010). 2020. European Commission A strategy for smart, sustainable and inclusive growth Communication from the Commission. Brussels, Belgium.
3. 11. Kalkınma Planı. https://www.sbb.gov.tr/wp-content/uploads/2022/07/On_Birinci_Kalkinma_Plani-2019-2023.pdf, (Eriřim tarihi: Ağustos 2022).
4. TUBİTAK öncelikli alanlar. https://tubitak.gov.tr/sites/default/files/18842/ar-ge_ve_yenilik_konu_basliklari_4.08.22.pdf, (Eriřim tarihi: Ağustos 2022).
5. Dani, R.R., Kubde, R.R., Sadhu, M.A., 2021. A Comprehensive Study of Various Techniques for Hostile Drone Detection and Their Classification. *2021 6th International Conference on Communication and Electronics Systems (ICCES)*, p. 1538–43.
6. Perez-Grau, F., Martinez-de Dios, J.R., L. Paneque, J., Acevedo, J., Torres-González, A., Viguria, A., et al., 2021. Introducing autonomous aerial robots in industrial manufacturing. **Journal of Manufacturing Systems** **60**: 312–24, Doi: 10.1016/j.jmsy.2021.06.008.
7. Levine, N.M., Spencer, B.F., 2022. Post-Earthquake Building Evaluation Using UAVs: A BIM-Based Digital Twin Framework. **Sensors** **22**(3), Doi: 10.3390/s22030873.
8. Sambolek, S., Ivasic-Kos, M., 2021. Automatic Person Detection in Search and Rescue Operations Using Deep CNN Detectors. **IEEE Access** **9**: 37905–22, Doi: 10.1109/ACCESS.2021.3063681.
9. Debats, S.R., Luo, D., Estes, L.D., Fuchs, T.J., Caylor, K.K., 2016. A generalized computer vision approach to mapping crop fields in heterogeneous agricultural landscapes. **Remote Sensing of Environment** **179**: 210–21, Doi: <https://doi.org/10.1016/j.rse.2016.03.010>.
10. Dairath, M.H., Akram, M.W., Mehmood, M.A., Sarwar, H.U., Akram, M.Z., Omar, M.M., et al., 2023. Computer vision-based prototype robotic picking cum grading

- system for fruits. **Smart Agricultural Technology** 4: 100210, Doi: <https://doi.org/10.1016/j.atech.2023.100210>.
11. Donmez, C., Villi, O., Berberoglu, S., Cilek, A., 2021. Computer vision-based citrus tree detection in a cultivated environment using UAV imagery. **Computers and Electronics in Agriculture** 187: 106273, Doi: <https://doi.org/10.1016/j.compag.2021.106273>.
 12. Mojumdar, M.U., Chakraborty, N.R., 2020. A Computer Vision Technique To Detect Scab on Malabar Nightshade. *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, p. 1–4.
 13. Zárate, V., Cáceres, D., 2022. Fruit Detection and Classification Using Computer Vision Techniques. *2022 8th International Engineering, Sciences and Technology Conference (IESTEC)*, p. 665–72.
 14. dos Santos de Arruda, M., Osco, L.P., Acosta, P.R., Gonçalves, D.N., Marcato Junior, J., Ramos, A.P.M., et al., 2022. Counting and locating high-density objects using convolutional neural network. **Expert Systems with Applications** 195: 116555, Doi: <https://doi.org/10.1016/j.eswa.2022.116555>.
 15. He, L., Wu, F., Du, X., Zhang, G., 2022. Cascade-SORT: A robust fruit counting approach using multiple features cascade matching. **Computers and Electronics in Agriculture** 200: 107223, Doi: <https://doi.org/10.1016/j.compag.2022.107223>.
 16. Quan, L., Jiang, W., Li, H., Li, H., Wang, Q., Chen, L., 2022. Intelligent intra-row robotic weeding system combining deep learning technology with a targeted weeding mode. **Biosystems Engineering** 216: 13–31, Doi: <https://doi.org/10.1016/j.biosystemseng.2022.01.019>.
 17. Grijalva, I., Spiesman, B.J., McCornack, B., 2023. Computer vision model for sorghum aphid detection using deep learning. **Journal of Agriculture and Food Research** 13: 100652, Doi: <https://doi.org/10.1016/j.jafr.2023.100652>.
 18. James, K., Bradshaw, K., 2020. Detecting plant species in the field with deep learning and drone technology. **Methods in Ecology and Evolution** 11(11): 1509–19, Doi: <https://doi.org/10.1111/2041-210X.13473>.

19. Chen, S.W., Shivakumar, S.S., Dcunha, S., Das, J., Okon, E., Qu, C., et al., 2017. Counting Apples and Oranges With Deep Learning: A Data-Driven Approach. **IEEE Robotics and Automation Letters** 2(2): 781–8, Doi: 10.1109/LRA.2017.2651944.
20. Goldman, E., Herzig, R., Eisenschtat, A., Goldberger, J., Hassner, T., 2019. Precise Detection in Densely Packed Scenes. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, p. 5222–31.
21. Saetchnikov, I. V., Tcherniavskaia, E.A., Skakun, V. V., 2021. Object Detection for Unmanned Aerial Vehicle Camera via Convolutional Neural Networks. **IEEE Journal on Miniaturization for Air and Space Systems** 2(2): 98–103, Doi: 10.1109/JMASS.2020.3040976.
22. Redmon, J., Farhadi, A., 2017. YOLO9000: Better, Faster, Stronger. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, p. 6517–25.
23. Itakura, K., Narita, Y., Noaki, S., Hosoi, F., 2021. Automatic pear and apple detection by videos using deep learning and a Kalman filter. **OSA Continuum** 4(5): 1688–95, Doi: 10.1364/OSAC.424583.
24. Behera, T.K., Bakshi, S., Sa, P.K., 2022. Vegetation Extraction from UAV-based Aerial Images through Deep Learning. **Computers and Electronics in Agriculture** 198: 107094, Doi: <https://doi.org/10.1016/j.compag.2022.107094>.
25. Lee, J., Wang, J., Crandall, D., Šabanović, S., Fox, G., 2017. Real-Time, Cloud-Based Object Detection for Unmanned Aerial Vehicles. *2017 First IEEE International Conference on Robotic Computing (IRC)*, p. 36–43.
26. Teng, Y.-F., Hu, B., Liu, Z.-W., Huang, J., Guan, Z.-H., 2017. Adaptive neural network control for quadrotor unmanned aerial vehicles. *2017 11th Asian Control Conference (ASCC)*, p. 988–92.
27. Abdelhay, S., Zakriti, A., 2019. Modeling of a Quadcopter Trajectory Tracking System Using PID Controller. **Procedia Manufacturing** 32: 564–71, Doi: <https://doi.org/10.1016/j.promfg.2019.02.253>.

28. Gupta, S., Ghonge, M., Jawandhiya, P., 2013. Review of Unmanned Aircraft System (UAS). **International Journal of Advanced Research in Computer Engineering & Technology** 9, Doi: 10.2139/ssrn.3451039.
29. Baykar Bayraktar TB2. https://tr.wikipedia.org/wiki/Baykar_TB2, (Erişim tarihi: Ağustos 2022).
30. Darvishpoor, S., Roshanian, J., Raissi, A., Hassanalian, M., 2020. Configurations, flight mechanisms, and applications of unmanned aerial systems: **A review. Progress in Aerospace Sciences** 121: 100694, Doi: <https://doi.org/10.1016/j.paerosci.2020.100694>.
31. Hassanalian, M., Abdelkefi, A., 2017. Classifications, applications, and design challenges of drones: **A review. Progress in Aerospace Sciences** 91: 99–131, Doi: <https://doi.org/10.1016/j.paerosci.2017.04.003>.
32. Ahmed., Mohanta, J., Keshari, A., Yadav, P., 2022. Recent Advances in Unmanned Aerial Vehicles: **A Review. Arabian Journal for Science and Engineering** 47: 1–22, Doi: 10.1007/s13369-022-06738-0.
33. Mn, A., Tom Scaria, C., Rose Thomas, S., Menon, S.S., Joy, S., Chacko, G., 2022. Development of a Robotic system for Power Transmission Line Inspection. *2022 IEEE 19th India Council International Conference (INDICON)*, p. 1–6.
34. Chen, W.-Y., Hu, J.-K., 2019. Research on Drone's Aerial Photography Aided Learning System Based on Deep Learning. *2019 IEEE International Conference on Consumer Electronics - Taiwan (ICCE-TW)*, p. 1–2.
35. Ma, R., Li, X., Sun, M., Kuang, Z., 2018. Experiment of Meteorological Disaster Monitoring on Unmanned Aerial Vehicle. *2018 7th International Conference on Agro-geoinformatics (Agro-geoinformatics)*, p. 1–6.
36. Dahle, O.H., Rydberg, J., Dullweber, M., Peinecke, N., Bechina, A.A.A., 2022. A proposal for a common metric for drone traffic density. *2022 International Conference on Unmanned Aircraft Systems (ICUAS)*, p. 64–72.
37. Zhou, J., Liu, K., Lü, Y., Chen, L., 2022. Complex network–based pinning control of drone swarm. **IFAC-PapersOnLine** 55(3): 207–12, Doi: <https://doi.org/10.1016/j.ifacol.2022.05.036>.

38. Najm Abdullah, M., Dagher, K.E., Sulaiman Wali, S., 2023. Improved airborne computer system strategy for swarm drones flying based on skybrush suite and inspired technique. Measurement: **Sensors** 27: 100766, Doi: <https://doi.org/10.1016/j.measen.2023.100766>.
39. Bautista, O., Akkaya, K., Uluagac, A.S., 2020. Customized novel routing metrics for wireless mesh-based swarm-of-drones applications. **Internet of Things** 11: 100265, Doi: <https://doi.org/10.1016/j.iot.2020.100265>.
40. Zhang, J., Li, Y., 2023. Collaborative vehicle-drone distribution network optimization for perishable products in the epidemic situation. **Computers & Operations Research** 149: 106039, Doi: <https://doi.org/10.1016/j.cor.2022.106039>.
41. Rave, A., Fontaine, P., Kuhn, H., 2023. Drone location and vehicle fleet planning with trucks and aerial drones. **European Journal of Operational Research** 308(1): 113–30, Doi: <https://doi.org/10.1016/j.ejor.2022.10.015>.
42. Ware house any where. <https://www.warehouseanywhere.com/resources/supply-chain-automation/>, (Eriřim tarihi: Ağustos 2022).
43. Kellermann, R., Flying high? Unmanned aircraft and the future of transportation. <https://il.boell.org/en/2022/02/24/flying-high-unmanned-aircraft-and-future-transportation>, (Eriřim tarihi: Ağustos 2022).
44. Tarhan, F.A., 2014. Modelling, Control and Implementation of an Unmanned Vertical Take-off and Landing Aircraft. Istanbul Technical University, 2014.
45. Bingöl, Dr.Z., Küçük, Dr.S., 2009. Robot kinematięi. İstanbul: Birsen yayınevi.
46. Jazar, R.N., 2011. Advanced Dynamics: Rigid Body, Multibody, and Aerospace Applications. Wiley.
47. KDE Direct 2315XF. <https://www.kdedirect.com/products/kde2315xf-885>, (Eriřim tarihi: Ağustos 2022).
48. Kumar, R., Wells, J.Z., Jhavar, D., Ranjan, K., Kumar, M., 2022. Prototype Development and Flight Controller Implementation of the Sliding-Arm Quadcopter. **IFAC-PapersOnLine** 55(37): 542–7, Doi: <https://doi.org/10.1016/j.ifacol.2022.11.239>.

49. Hokuyo., URG-04LX-UG01. <https://www.hokuyo-aut.jp/search/single.php?serial=166> , (Eriřim tarihi: Aęustos 2022).
50. Yıldırım řahin and Ulu, B., 2023. Investigation of the SLAM Performance of the Multirotor MAV Developed for the Analysis of Orchards. In: Cioboață, D.D., editor. *International Conference on Reliable Systems Engineering (ICoRSE) - 2022*, Cham: Springer International Publishing p. 247–54.
51. Odroid XU4. <https://wiki.odroid.com/odroid-xu4/odroid-xu4>, (Eriřim tarihi: Aęustos 2022).
52. Yıldırım, ř., ULU, B., 2021. Tarımsal verimlilięi arttırması için geliştirilen uçan robotik sisteme gazebo ortamında ekvh uygulanması. Dicle Üniversitesi, Mühendislik Fakóltesi.
53. Python. <https://tr.wikipedia.org/wiki/Python>, (Eriřim tarihi: Aęustos 2022).
54. Krishna, R., 2017. Computer Vision Foundations and Applications. Stanford University.
55. Dikici, B., 2021. Object Detection Using Convolutional Neural Network. Gaziantep University, 2021.
56. Ilioudi, A., Dabiri, A., Wolf, B.J., De Schutter, B., 2022. Deep Learning for Object Detection and Segmentation in Videos: Toward an Integration With Domain Knowledge. **IEEE Access** **10**: 34562–76, Doi: 10.1109/ACCESS.2022.3162827.
57. OpenCV. <https://opencv.org/>, (Eriřim tarihi: Aęustos 2022).
58. Piřkin, M., 2020. OpenCV ile Görüntü İşleme, (Eriřim tarihi: Aęustos 2022).
59. Smith, M., 2020. Introduction to OpenCV, (Eriřim tarihi: Aęustos 2022).
60. Aziz, L., Haji Salam, Md.S. Bin., Sheikh, U.U., Ayub, S., 2020. Exploring Deep Learning-Based Architecture, Strategies, Applications and Current Trends in Generic Object Detection: A Comprehensive Review. **IEEE Access** **8**: 170461–95, Doi: 10.1109/ACCESS.2020.3021508.
61. Jiao, L., Zhang, F., Liu, F., Yang, S., Li, L., Feng, Z., et al., 2019. A Survey of Deep Learning-Based Object Detection. **IEEE Access** **7**: 128837–68, Doi: 10.1109/ACCESS.2019.2939201.

62. Xiao, Y., Tian, Z., Yu, J., Zhang, Y., Liu, S., Du, S., et al., 2020. A Review of Object Detection Based on Deep Learning. **Multimedia Tools Appl.** **79**(33–34): 23729–91, Doi: 10.1007/s11042-020-08976-6.
63. Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y., 2014. OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks.
64. Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You Only Look Once: Unified, Real-Time Object Detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, p. 779–88.
65. Redmon, J., Farhadi, A., 2018. YOLOv3: An Incremental Improvement.
66. Liu Wei and Anguelov, D. and E.D. and S.C. and R.S. and F.C.-Y. and B.A.C., 2016. SSD: Single Shot MultiBox Detector. In: Leibe Bastian and Matas, J. and S.N. and W.M., editor. *Computer Vision – ECCV 2016*, Cham: Springer International Publishing p. 21–37.
67. Fu, C.-Y., Liu, W., Ranga, A., Tyagi, A., Berg, A.C., 2017. DSSD : Deconvolutional Single Shot Detector.
68. Li, Z., Zhou, F., 2018. FSSD: Feature Fusion Single Shot Multibox Detector.
69. Shen, Z., Liu, Z., Li, J., Jiang, Y.-G., Chen, Y., Xue, X., 2017. DSOD: Learning Deeply Supervised Object Detectors from Scratch. p. 1937–45.
70. Girshick, R., Donahue, J., Darrell, T., Malik, J., 2014. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, p. 580–7.
71. He Kaiming and Zhang, X. and R.S. and S.J., 2014. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. In: Fleet David and Pajdla, T. and S.B. and T.T., editor. *Computer Vision – ECCV 2014*, Cham: Springer International Publishing p. 346–61.
72. Girshick, R., 2015. Fast R-CNN. *2015 IEEE International Conference on Computer Vision (ICCV)*, p. 1440–8.
73. Ren, S., He, K., Girshick, R., Sun, J., 2016. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. p. 1–10.

74. Dai, J., Li, Y., He, K., Sun, J., 2016. R-FCN: Object Detection via Region-Based Fully Convolutional Networks. Proceedings of the 30th International Conference on Neural Information Processing Systems, Red Hook, NY, USA: Curran Associates Inc. p. 379–87.
75. He, K., Gkioxari, G., Dollár, P., Girshick, R., 2017. Mask R-CNN. *2017 IEEE International Conference on Computer Vision (ICCV)*, p. 2980–8.
76. Kang, J., Tariq, S., Oh, H., Woo, S.S., 2022. A Survey of Deep Learning-Based Object Detection Methods and Datasets for Overhead Imagery. **IEEE Access** **10**: 20118–34, Doi: 10.1109/ACCESS.2022.3149052.
77. Jiao, L., Zhang, R., Liu, F., Yang, S., Hou, B., Li, L., et al., 2022. New Generation Deep Learning for Video Object Detection: A Survey. **IEEE Transactions on Neural Networks and Learning Systems** **33**(8): 3195–215, Doi: 10.1109/TNNLS.2021.3053249.
78. Turay, T., Vladimirova, T., 2022. Toward Performing Image Classification and Object Detection With Convolutional Neural Networks in Autonomous Driving Systems: **A Survey. IEEE Access** **10**: 14076–119, Doi: 10.1109/ACCESS.2022.3147495.
79. Lin, T., Maire, M., Belongie, S., Hays., J., Perona, P., Ramanan, D., Dollar, P., Zitnick, L. 2015. Microsoft COCO: Common Objects in Context. 2014 European Conference on Computer Vision (ECCV) p. 740–755.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı: Burak ULU
Uyruğu: Türkiye (T.C)

EĞİTİM

Derece	Kurum	Mezuniyet Tarihi
Yüksek Lisans	Erciyes Üniversitesi, Fen Bilimleri Enstitüsü	2017
Lisans	Erciyes Üniversitesi Mekatronik Mühendisliği	2014
Lise	Ordu Anadolu Lisesi, Ordu	2008

İŞ DENEYİMLERİ

Yıl	Kurum	Görev
2014-Halen	Erciyes Üniversitesi	9 Yıl

YABANCI DİL

İngilizce

YAYINLAR

SCI, SSCI, AHCI İndekslerine Giren Dergilerde Yayımlanan Makaleler

1. Ulu, B., Yıldırım, Ş. (2023). Detection of Apple Density to Improve CNN-Based Yield Estimation Results with Flying Robotic System. *Measurement, (İnceleme)*, ss 1-13.

2. Yıldırım, Ş., Ulu, B. (2022). Deep Learning Based Apples Counting for Yield Forecast using Flying Robotic System. *Sensor, (Basım aşamasında)*, ss 1-13.

Diğer Dergilerde Yayımlanan Makaleler

1. Yıldırım, Ş., Ulu, B. (2017). Use Of Artificial Neural Networks For The Correction Of Faults On The Detected Weld Seam Image. *Journal of Current Issue*. Ss 260-269.

Hakemli Kongre/Sempozyum Bildiri Kitaplarında Yer Alan Yayınlar

1. Yıldırım, Ş., Ulu, B. (2017). Use Of Artificial Neural Networks For The Correction Of Faults On The Detected Weld Seam Image. *Journal of Current Contruction Issue*, ss 260-269.
2. Yıldırım, Ş., Ulu, B. (2023). Examination of the Effect of Hyperparameters on Object Detection in the Orchard using Deep Learning. *International Conference on Reliable Systems Engineering, ICoRSE*, ss 1-6.
3. Yıldırım, Ş., Ulu, B. (2022). Investigation of the SLAM Performance of the Multirotor MAV Developed for the Analysis of Orchards. *International Conference on Reliable Systems Engineering, ICoRSE*, 534, ss 247-254.
4. Yıldırım, Ş., Ulu, B. (2022). Dört Rotorlu Uçan Robotik Sistem Üzerinden Sinir Ağı Kullanılarak Nesne Tespit Edilmesi. *Otomatik Kontrol Ulusal Kongresi*, ss 1-4.
5. Yıldırım, Ş., Ulu, B. (2019). Tarımsal Verimliliği Arttırması için Geliştirilen Uçan Robotik Sisteme Gazebo Ortamında EKVH Uygulanması. *Ulusal Makine Teorisi Sempozyumu*, ss 1-5.