

T.C.  
İSTANBUL GEDİK ÜNİVERSİTESİ  
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ



DERİN ÖĞRENME TABANLI SES VE GÖRÜNTÜ İŞLEME  
TEKNOLOJİLERİNE SAHİP OTONOM İNSANSIZ HAVA  
ARACI

DOKTORA TEZİ

Ertuğrul KIRAÇ

Savunma Teknolojileri Ana Bilim Dalı  
Savunma Teknolojileri Doktora Programı

OCAK 2024  
İSTANBUL

T.C.  
İSTANBUL GEDİK ÜNİVERSİTESİ  
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ



DERİN ÖĞRENME TABANLI SES VE GÖRÜNTÜ İŞLEME  
TEKNOLOJİLERİNE SAHİP OTONOM İNSANSIZ HAVA  
ARACI

DOKTORA TEZİ

Ertuğrul KIRAÇ  
(144138001)

Savunma Teknolojileri Ana Bilim Dalı  
Savunma Teknolojileri Doktora Programı

Tez Danışmanı: Prof. Dr. Sunullah ÖZBEK

İstanbul 2024



**T.C.**  
**İSTANBUL GEDİK ÜNİVERSİTESİ**  
**Lisansüstü Eğitim Enstitüsü Müdürlüğü**

**Jüri Tez Onay Formu**

03.01.2024

**LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ MÜDÜRLÜĞÜ**

Bu çalışma 03.01.2024 tarihinde aşağıdaki jüri tarafından Savunma Teknolojisi Anabilim Dalı, Savunma Teknolojileri (Doktora) Programı, Doktora Tezi olarak kabul edilmiştir.

**TEZ JÜRİSİ**

**Prof. Dr. Sunullah ÖZBEK**

Danışman

İstanbul Gedik Üniversitesi

**Prof. Dr. Feriha ERFAN KUYUMCU**

Üye (İmza)

İstanbul Gedik Üniversitesi

**Prof. Dr. Tarık BAYKARA**

Üye (İmza)

İstanbul Doğu Üniversitesi

**Prof. Dr. Ethem Soner ÇELİKKOL**

Üye (İmza)

İstanbul Beykent Üniversitesi

**Doç. Dr. Haydar ŞAHİN**

Üye (İmza)

İstanbul Gedik Üniversitesi

## YEMİN METNİ

Doktora tezi olarak sunduđum “Derin Öğrenme Tabanlı Ses ve Görüntü İşleme Teknolojilerine Sahip Otonom İnsansız Hava Aracı” adlı çalışmanın, tezin proje safhasından sonuçlanmasına kadarki bütün süreçlerde bilimsel ahlak ve geleneklere aykırı düşecek bir yardıma başvurulmaksızın yazıldığını ve yararlandığım eserlerin Bibliyografya’da gösterilenlerden oluştuđunu, bunlara atıf yapılarak yararlanılmış olduğunu belirtir ve onurumla beyan ederim (03/01/2024).

Ertuđrul KIRAÇ



## ÖNSÖZ

Bu çalışmanın her aşamasında değerli bilgilerini ve desteklerini aktaran çok değerli hocam Prof. Dr. Sunullah ÖZBEK'e şükranlarımı sunarım.

Ayrıca tez süresi boyunca yol gösteren ve manevi desteklerini hiç esirgemeyen sayın Prof. Dr. Tarık BAYKARA ve Prof. Dr. Feriha ERFAN KUYUMCU hocalarıma, ayrıca tez jüri üyelerimden Prof. Dr. Ethem Soner ÇELİKKOL ve Doç. Dr. Haydar ŞAHİN hocalarıma katkılarından dolayı teşekkür ederim.

Doktora süresince her zaman destek olan, sabır ve hoşgörü gösteren değerli eşim Esra KIRAÇ ve oğlum Ömer İhsan KIRAÇ'a teşekkür ederim.

Ocak 2024

Ertuğrul KIRAÇ

---

# İÇİNDEKİLER

## Sayfa

ÖNSÖZ.....	iv
İÇİNDEKİLER .....	v
KISALTMALAR .....	viii
ÇİZELGE LİSTESİ.....	ix
ŞEKİL LİSTESİ.....	x
ÖZET.....	xiv
ABSTRACT .....	xv
<b>1. GİRİŞ .....</b>	<b>1</b>
1.1 İnsansız Hava Araçlarının Kullanım Alanları .....	6
1.2 İnsansız Hava Araçlarının Kategorileri ve Türkiye’deki Sınıflandırılması .....	7
1.3 Literatür Araştırması .....	8
1.4 Tezin Amacı .....	12
<b>2. İNSANSIZ HAVA ARACI TASARIMI .....</b>	<b>14</b>
2.1 Hexacopter İnsansız Hava Aracı Çalışma Mekanizması .....	15
2.2 Hexacopter İnsansız Hava Aracının Matematiksel Modeli.....	16
2.3 Hexacopter İnsansız Hava Aracında Kullanılan Donanımlar .....	20
2.3.1 Elektriksel donanımlar.....	20
2.3.1.1 Dış rotorlu (outrunner) fırçasız DC motor .....	20
2.3.1.2 Güç dağıtım kartı .....	22
2.3.1.3 Lipo pil.....	22
2.3.2 Elektronik donanımlar .....	23
2.3.2.1 Otopilot uçuş kartı .....	23
2.3.2.2 Elektronik hız denetleyicisi .....	24
2.3.2.3 Radyo alıcı ve verici .....	24
2.3.2.4 Telemetry .....	25
2.3.2.5 GPS Modülü .....	26
2.3.2.6 Aksiyon kamera .....	26
2.3.2.7 Gimbal .....	27
2.3.2.8 Pil göstergesi.....	28
2.3.2.9 Video verici.....	28
2.3.3 Mekanik donanımlar.....	28
2.3.3.1 Gövde.....	29
2.3.3.2 İniş takımı .....	29
2.3.3.3 Pervane.....	30
2.4 İHA Uçuş Süresi Hesaplamaları .....	30
2.4.1 İHA Tahmini gövde ağırlığı .....	30
2.4.2 İHA İtki sistemi ve azami uçuş süresi hesaplamaları .....	31
2.5 Aviyonik Sistem .....	36
2.5.1 Planlı uçuş kontrolü .....	36
2.5.2 Otomatik kalkış ve iniş .....	37
2.5.3 Acil durumda eve dönüş modu.....	38

2.5.4 Eş zamanlı görüntü ve telemetri verileri aktarımı .....	38
<b>3. METODOLOJİ .....</b>	<b>39</b>
3.1 Yapay Sinir Ağı.....	39
3.2 Makine Öğrenmesi .....	42
3.3 Derin Öğrenme .....	42
3.3.1 Evrimsel sinir ağı .....	44
3.3.2 Hiperparametrelerin derin sinir ağı üzerindeki etkisi .....	44
3.4 Bilgisayarlı Görü .....	47
3.4.1 Görüntü işleme .....	48
3.4.2 Görüntü ile nesne algılama ve sınıflandırma .....	49
3.4.3 Nesne sınıflandırmada performans metrikleri .....	50
3.4.3.1 Karışıklık matrisi (Confusion matrix).....	50
3.4.3.2 Doğruluk (Accuracy) .....	52
3.4.3.3 Hassasiyet (Precision).....	52
3.4.3.4 Geri çağırma (Recall).....	53
3.4.3.5 F1 Skoru (F1 Score).....	53
3.4.3.6 Ortalama Hassasiyet (Average Precision-AP) ve Ortalama Ortalama Hassasiyet (Mean Average Precision-MAP) .....	53
3.4.4 Görüntü ile Nesne Algılama ve Sınıflandırmada Kullanılan Derin Öğrenme Yapısı ve Teknolojileri .....	54
3.4.4.1 MobileNet mimari yapısı .....	54
3.4.4.2 Tek atışlı çoklu kutu dedektörü .....	54
3.4.4.3 Coco veri seti .....	54
3.4.4.4 SSD Tabanlı MobilnetV2 sinir ağı mimarisi ile nesne algılama .....	55
3.4.4.5 TensorFlow ve tensorflow lite .....	56
3.4.5 SSD MobilNetV2 sinir ağının transfer öğrenme yöntemi ile yeniden eğitilmesi ve özelleştirilmesi .....	57
3.4.5.1 Veri setinin hazırlanması .....	58
3.4.5.2 SSD MobilNetV2 sinir ağı modelinin özelleştirilmesi (Hiperparametre optimizasyonu) ve Raspberry Pi'de kullanılacak TFLite modeline dönüştürülmesi.....	61
3.5 Ses İşleme.....	76
3.5.1 Ses sınıflandırması.....	76
3.5.2 Derin öğrenme tabanlı YAMNET ses sınıflandırma modeli.....	77
3.5.2.1 Ses verisi ön işleme hazırlığı .....	77
3.5.2.2 Mel spektrogramlarını çıkarma.....	78
3.5.3 Ses Verisinde Bulunan ve Ses Sınıflandırma Sonucunu Olumsuz Etkileyen Gürültülerin Tespiti ve Bu Gürültülerin Filtreleme Yöntemi.....	81
3.5.3.1 Ses sınıflandırmasını olumsuz etkileyen gürültüler.....	82
3.5.3.2 İnsansız hava aracı motorlarının ürettiği gürültülerin ses analizleri... 82	
3.5.3.2.1 Hexacopter dış rotorlu fırçasız DC motorlarının (pervanesiz) hover modundaki roll, pitch ve yaw uçuş hareketlerinde ürettiği seslerin spektrogramları .....	84
3.5.3.1.2 İnsansız hava aracının motor-pervane ikililerinin ürettiği gürültülerin ses analizleri .....	89
3.5.3.1.3 Hexacopter dış rotorlu fırçasız DC motor-pervane ikilisinin hover modundaki roll, pitch ve yaw uçuş hareketlerinde çıkardığı seslerin spektrogramları .....	90
3.5.3.2 Ses sınıflandırmasını olumsuz etkileyen gürültüleri filtreleme yöntemi .....	94

3.5.4 YAMNet Ses sınıflandırma modelinin transfer öğrenme yöntemi ile yeniden eğitilmesi ve özelleştirilmesi.....	97
3.5.4.1 Veri setinin hazırlanması .....	97
3.5.4.2 Otomatik veri ön işleme hazırlığı .....	98
3.5.4.3 Modelin transfer öğrenme yöntemi ile yeniden eğitilmesi ve özelleştirilmesi .....	100
3.6 İnsansız Hava Aracı Platformu Gömülü Sistem Tasarımı ve Yazılımı .....	110
3.6.1 İnsansız Hava Aracı Platformu Gömülü Sistem Donanımları .....	112
3.6.1.1 Raspberry Pi 4 model B .....	113
3.6.1.2 Raspberry Pi kamera modülü V2.....	114
3.6.1.3 Sixfab 4G/LTE hücresel modem kiti .....	115
3.6.1.4 Coral edge TPU .....	116
3.6.1.5 Harici USB ses kartı ve mikrofon.....	116
3.6.2 İnsansız hava aracı platformu gömülü sistem yazılım mimarisi .....	117
3.6.2.1 Sistem yazılımı bileşenleri.....	117
3.6.2.1.1 Python programlama dili.....	118
3.6.2.1.2 Yer kontrol istasyonu yazılımı – mission planner .....	119
3.6.2.1.3 MAVLink haberleşme protokolü .....	121
3.6.2.2 Otonom Karar Verme Yapısı .....	123
<b>4. İHA UÇUŞ TESTLERİ SONUÇLARI .....</b>	<b>125</b>
<b>5. TARTIŞMA .....</b>	<b>137</b>
<b>6. SONUÇ VE ÖNERİLER.....</b>	<b>140</b>
<b>KAYNAKLAR .....</b>	<b>142</b>
<b>ÖZGEÇMİŞ.....</b>	<b>163</b>

## KISALTMALAR

<b>İHA</b>	: İnsansız Hava Aracı
<b>UAV</b>	: Unmanned Aerial Vehicle
<b>CNN</b>	: Conventional Neural Network
<b>COCO</b>	: Common Objects in Context
<b>GCS</b>	: Ground Control System
<b>MAVLink</b>	: Micro Air Vehicle Link
<b>VTOL</b>	: Vertical Takeo Off and Landing
<b>SSD</b>	: Single Shot Multibox Detector
<b>ESC</b>	: Electronic Speed Controller
<b>Lipo</b>	: Lityum Polimer
<b>Mbps</b>	: Megabits per second
<b>AP</b>	: Average Precision
<b>mAP</b>	: Mean Average Precision
<b>SHGM</b>	: Sivil Havacılık Genel Müdürlüğü
<b>SGD(M)</b>	: Stochastic Gradient Descent (with Momentum)
<b>YAMNet</b>	: Yet Another Mobile Network
<b>ReLU</b>	: Rectified Linear Units
<b>ADAM</b>	: Adaptive Moment Estimation

## ÇİZELGE LİSTESİ

### Sayfa

Çizelge 1.1: SHGM İHA Sınıflandırması.....	8
Çizelge 2.1: İHA Bileşenleri Ortalama Ağırlığı ve Toplam Platform Ağırlığı .....	31
Çizelge 3.1: SSD MobilNetV2 Modelinin Hiperparametre Değerleri .....	62
Çizelge 3.2: ReLU, Sigmoid ve Tanh Fonksiyonlarının Denklemleri .....	63
Çizelge 3.3: Adım Sayısına Bağlı Öğrenme Hızı Değerleri.....	67
Çizelge 3.4: Adım Sayısına Bağlı Toplam Kayıp Değerleri .....	68
Çizelge 3.5: Adım Sayısına Bağlı Classification_Loss Ve Localization_Loss Değerleri.....	69
Çizelge 3.6: İnsansız Hava Aracı Motorlarının Gürültü Ölçüm Detayı .....	83
Çizelge 3.7: İnsansız Hava Aracı Motorlarının Uçuş Hareketlerindeki Hız Değişimleri .....	86
Çizelge 3.8: YAMNet Modelinin Hiperparametre Yapılandırma Değerleri.....	101
Çizelge 4.1: Nesne Sınıflandırma ve Doğruluk Yüzdesi.....	125
Çizelge 5.1: Çalışmadaki Yenilikler.....	138

## ŞEKİL LİSTESİ

### Sayfa

<b>Şekil 1.1:</b> Yapay Zekâ, Makine Öğrenmesi, Derin Öğrenme, Bilgisayarlı Görü ve Evrışimsel Sinir Ağları Arasındaki İlişki .....	2
<b>Şekil 1.2:</b> Yapılarına Göre İHA kategorileri.....	7
<b>Şekil 2.1:</b> Hexacopter Yapısı .....	15
<b>Şekil 2.2:</b> F550 Hexacopter Tasarımı .....	16
<b>Şekil 2.3:</b> F550 Hexacopter'in x,y ve z Eksenleri .....	17
<b>Şekil 2.4:</b> F550 Hexacopter'in Roll, Pitch ve Yaw Eksenleri .....	17
<b>Şekil 2.5:</b> Hexacopter Uçuş Kontrol Kartı ve GPS Modülü Yönü .....	18
<b>Şekil 2.6:</b> Tipik bir Dış Rotorlu BLDC Motoru .....	21
<b>Şekil 2.7:</b> SunnySky X2216 880Rpm/Volt Dış Rotorlu Fırçasız DC Motor.....	22
<b>Şekil 2.8:</b> Holybro Pixhawk 4 Güç Modülü (PM07).....	22
<b>Şekil 2.9:</b> Leopard Power 7000 mAh 40C 14.8 V-4S Lipo Pil .....	23
<b>Şekil 2.10:</b> Holybro Pixhawk 4 Uçuş Kontrol Kartı.....	24
<b>Şekil 2.11:</b> Racerstar RS40A V2 40A BLHELI_S BB2 OPTO 2-5S ESC .....	24
<b>Şekil 2.12:</b> Radiolink AT10 RC Verici ve R12DS Alıcı .....	25
<b>Şekil 2.13:</b> Holybro Telemetry Radyo.....	25
<b>Şekil 2.14:</b> GCS ile İHA arasındaki Çift Yönlü Veri Akışı.....	26
<b>Şekil 2.15:</b> Holybro Pixhawk 4 GPS Modül.....	26
<b>Şekil 2.16:</b> Sjpgcam Sj6 Legend 4K Aksiyon Kamerası.....	27
<b>Şekil 2.17:</b> Tarot 3DIII Metal Gimbal (TL3T01) .....	27
<b>Şekil 2.18:</b> Lipo Pil Göstergesi .....	28
<b>Şekil 2.19:</b> Video Verici .....	28
<b>Şekil 2.20:</b> F550 Gövde .....	29
<b>Şekil 2.21:</b> F550 İniş Takımı .....	30
<b>Şekil 2.22:</b> 1045 CW ve CCW Pervane Çifti .....	30
<b>Şekil 2.23:</b> İHA Platformunu Oluşturan Uçuş Kontrol Sistemi ve Gömülü Sistem Donanımları .....	32
<b>Şekil 2.24:</b> SunnySky X2216 880Rpm/Volt Dış Rotorlu Fırçasız DC Motor Thrust Değerleri .....	33
<b>Şekil 2.25:</b> Mission Planer GCS Planlı Uçuş Modu (Auto Mode) – Waypoints.....	37
<b>Şekil 3.1:</b> Biyolojik Nöron Modeli .....	39
<b>Şekil 3.2:</b> İki Gizli Katmanı Olan Bir Sinir Ağı Örneği .....	40
<b>Şekil 3.3:</b> Sinir Ağı Modeli İle Gerçekleştirilen Görüntü Algılama Prensipleri.....	40
<b>Şekil 3.4:</b> Basit Yapay Nöron Modeli.....	41
<b>Şekil 3.5:</b> Temel bir Evrışimli Sinir Ağı (CNN) Mimarisi.....	44
<b>Şekil 3.6:</b> Yaygın Olarak Kullanılan Üç Aktivasyon Fonksiyonu: (a) Sigmoid, (b) Tanh ve (c) ReLU .....	45
<b>Şekil 3.7:</b> AH-64 Apache Helikopterinin Etiketlenmesi .....	48
<b>Şekil 3.8:</b> Algılanan Nesnenin Sınırlayıcı Dörtgen Kutu İle Tespit Edilmesi ve Sınıflandırılması .....	49

<b>Şekil 3.9:</b> İkili Sınıflandırma Problemi İçin Karışıklık Matrisi.....	50
<b>Şekil 3.10:</b> İkili Sınıflandırma Problemi İçin Örnek Karışıklık Matrisi.....	51
<b>Şekil 3.11:</b> MobilNet-SSD ile Nesne Algılama ve Sınıflandırma.....	55
<b>Şekil 3.12:</b> TensorFlow Çekirdeği İle Oluşturulan Bir Modelin TensorFlow Lite Modeline Dönüştürülmesinin İş Akışı.....	57
<b>Şekil 3.13:</b> TensorFlow Çekirdeği İle Oluşturulan Bir Modeli Tensorflow Lite Formatına Dönüştüren Python Kodu.....	57
<b>Şekil 3.14:</b> AH-64 Apache Helikopterine ait Görüntüler.....	59
<b>Şekil 3.15:</b> AH-64 Apache Helikopterinin Etiketlenmesi.....	59
<b>Şekil 3.16:</b> PASCAL VOC Formatında XML Dosyası.....	60
<b>Şekil 3.17:</b> AH-64 Apache Helikopterinin Görüntüleri ve Her Bir Görüntünün Etiketini İçeren XML Dosyaları.....	60
<b>Şekil 3.18:</b> SSD MobilNetV2 Modelinin Eğitim Sonundaki Kayıp ve Öğrenme Hızı Değerleri.....	62
<b>Şekil 3.19:</b> Genel Bir Aktivasyon Fonksiyonun Yapısı.....	63
<b>Şekil 3.20:</b> Yaygın Olarak Kullanılan Üç Aktivasyon Fonksiyonu: (a) ReLU, (b) Sigmoid ve (c) Tanh.....	63
<b>Şekil 3.21:</b> SSD MobilNetV2 Modelinin Eğitim Sırasında Adım Sayısına Göre Öğrenme Hızı Değerleri Grafiği.....	67
<b>Şekil 3.22:</b> SSD MobilNetV2 Modelinin Eğitim Sırasında Adım Sayısına Göre Toplam Kayıp Değerleri Grafiği.....	68
<b>Şekil 3.23:</b> SSD MobilNetV2 Modelinin Eğitim Sırasında Adım Sayısına Göre (a)Sınıflandırma Kayıp Değerleri Grafiği ile (b)Yerleştirme Kayıp Değerleri Grafiği.....	69
<b>Şekil 3.24:</b> Edge TPU Modelini Oluşturmaya Yönelik Temel Süreç.....	70
<b>Şekil 3.25:</b> SSD MobilNetV2 Modelinin Eğitim Sonundaki Test Sonucu-1.....	71
<b>Şekil 3.26:</b> SSD MobilNetV2 Modelinin Eğitim Sonundaki Test Sonucu-2.....	71
<b>Şekil 3.27:</b> SSD MobilNetV2 Modelinin Eğitim Sonundaki Test Sonucu-3.....	72
<b>Şekil 3.28:</b> SSD MobilNetV2 Modelinin Eğitim Sonundaki Test Sonucu-4.....	72
<b>Şekil 3.29:</b> SSD MobilNetV2 Modelinin Eğitim Sonundaki Test Sonucu-5.....	73
<b>Şekil 3.30:</b> SSD MobilNetV2 Modelinin Eğitim Sonundaki Test Sonucu-6.....	73
<b>Şekil 3.31:</b> SSD MobilNetV2 Modelinin Eğitim Sonundaki Test Sonucu-7.....	74
<b>Şekil 3.32:</b> SSD MobilNetV2 Modelinin Eğitim Sonundaki Test Sonucu-8.....	74
<b>Şekil 3.33:</b> SSD MobilNetV2 Modelinin Eğitim Sonundaki Test Sonucu-9.....	75
<b>Şekil 3.34:</b> SSD MobilNetV2 Modelinin Eğitim Sonundaki Test Sonucu-10.....	75
<b>Şekil 3.35:</b> YAMNet Ses Özellik Çıkarım ve Tahmin Süreci.....	77
<b>Şekil 3.36:</b> Genel bir Kısa Süreli Fourier Dönüşümü (STFT).....	79
<b>Şekil 3.37:</b> Hann(Hanning) Penceresi.....	80
<b>Şekil 3.38:</b> Hexacopter ve Altı Adet Pervanesiz Motorları.....	83
<b>Şekil 3.39:</b> Hexacopter'deki Gömülü Sistem Mikrofonu.....	84
<b>Şekil 3.40:</b> Hexacopter'deki Motorların Numaraları ve Dönüş Yönleri.....	85
<b>Şekil 3.41:</b> Hexacopter'deki Motorların Numaraları, Dönüş Yönleri ve Dönüş Hızları.....	85
<b>Şekil 3.42:</b> Hexacopter Rc Kumanda Joystick İşlevi.....	86
<b>Şekil 3.43:</b> Hexacopter Hover Modunda Uçarken Altı Motorun Ürettiği Gürültü Spektrogramı.....	87
<b>Şekil 3.44:</b> Hexacopter Roll Left Hareketindeki Altı Motorun Ürettiği Gürültü Spektrogramı.....	87
<b>Şekil 3.45:</b> Hexacopter Roll Right Hareketindeki Altı Motorun Ürettiği Gürültü Spektrogramı.....	87

<b>Şekil 3.46:</b> Hexacopter Yaw Left Hareketindeki Altı Motorun Gürültü Spektrogramı .....	88
<b>Şekil 3.47:</b> Hexacopter Yaw Right Hareketindeki Altı Motorun Ürettiği Gürültü Spektrogramı .....	88
<b>Şekil 3.48:</b> Hexacopter Pitch Forward Hareketindeki Altı Motorun Ürettiği Gürültü Spektrogramı .....	89
<b>Şekil 3.49:</b> Hexacopter Pitch Back Hareketindeki Altı Motorun Ürettiği Gürültü Spektrogramı .....	89
<b>Şekil 3.50:</b> Hexacopter ve Altı Adet Pervaneli Motoru.....	90
<b>Şekil 3.51:</b> Hexacopter Eksen Hareketleri ve Motor Hızları.....	91
<b>Şekil 3.52:</b> Hexacopter Hover Modu Altı adet Motor-Pervane İkिलisinin Ürettiği Gürültü Spektrogramı .....	92
<b>Şekil 3.53:</b> Hexacopter Roll Left Hareketindeki Altı adet Motor-Pervane İkिलisinin Ürettiği Gürültü Spektrogramı.....	92
<b>Şekil 3.54:</b> Hexacopter Roll Right Hareketindeki Altı adet Motor-Pervane İkिलisinin Ürettiği Gürültü Spektrogramı.....	92
<b>Şekil 3.55:</b> Hexacopter Yaw Left Hareketindeki Altı adet Motor-Pervane İkिलisinin Gürültü Spektrogramı .....	93
<b>Şekil 3.56:</b> Hexacopter Yaw Right Hareketindeki Altı adet Motor-Pervane İkिलisinin Gürültü Spektrogramı .....	93
<b>Şekil 3.57:</b> Hexacopter Pitch Forward Hareketindeki Altı adet Motor-Pervane İkिलisinin Ürettiği Gürültü Spektrogramı .....	93
<b>Şekil 3.58:</b> Hexacopter Pitch Back Hareketindeki Altı adet Motor-Pervane İkिलisinin Ürettiği Gürültü Spektrogramı.....	94
<b>Şekil 3.59:</b> Low-Pass Butterworth Filtresi .....	95
<b>Şekil 3.60:</b> High-Pass Butterworth Filtresi.....	95
<b>Şekil 3.61:</b> Band-Pass Butterworth Filtresi .....	96
<b>Şekil 3.62:</b> Band-Stop Butterworth Filtresi .....	96
<b>Şekil 3.63:</b> ESC-50 Çevresel Ses Sınıflandırması Veri Setinin Sınıfları .....	97
<b>Şekil 3.64:</b> ESC-50 Çevresel Ses Sınıflandırması Veri Setinin Helicopter Sınıfı.....	98
<b>Şekil 3.65:</b> YAMNet Modelinin Girdi Gereksinimine Göre Ses Sinyalinin Mel Spektrogramını çıkarma adımları .....	99
<b>Şekil 3.66:</b> YAMNet Modelinin Bazı Sınıfları.....	100
<b>Şekil 3.67:</b> YAMNet Ses Sınıflandırma Modelinin Mimarisi .....	101
<b>Şekil 3.68:</b> YAMNet Ses Sınıflandırma Modelinin Transfer Öğrenme Yöntemi İle Eğitilmesi.....	101
<b>Şekil 3.69:</b> (a) Momentumsuz SGD (b) Momentumlu SGD .....	102
<b>Şekil 3.70:</b> Adam Algoritmasını Uygulayan Optimizer ve Argümanlar .....	106
<b>Şekil 3.71:</b> Adam Optimizer Compile Etme .....	106
<b>Şekil 3.72:</b> MNIST Görüntülerinde Çok Katmanlı Sinir Ağlarının Eğitimi .....	106
<b>Şekil 3.73:</b> YAMNet Ses Sınıflandırma Modelinin Test Sonucu-1 .....	107
<b>Şekil 3.74:</b> YAMNet Ses Sınıflandırma Modelinin Test Sonucu-2 .....	108
<b>Şekil 3.75:</b> YAMNet Ses Sınıflandırma Modelinin Test Sonucu-3 .....	108
<b>Şekil 3.76:</b> YAMNet Ses Sınıflandırma Modelinin Test Sonucu-4 .....	109
<b>Şekil 3.77:</b> YAMNet Ses Sınıflandırma Modelinin Test Sonucu-5 .....	109
<b>Şekil 3.78:</b> YAMNet Ses Sınıflandırma Modelinin Test Sonucu-6 .....	110
<b>Şekil 3.79:</b> Önerilen Gömülü Sistem İHA .....	111
<b>Şekil 3.80:</b> İnsansız Hava Aracı Platformu ve Yer Kontrol İstasyonu .....	113
<b>Şekil 3.81:</b> Raspberry Pi 4 Model B .....	114
<b>Şekil 3.82:</b> Raspberry Pi Kamera Modülü V2 .....	114

<b>Şekil 3.83:</b> Sixfab 4G/LTE Modem Kit.....	115
<b>Şekil 3.84:</b> Sixfab 4G/LTE Modem Web Arayüzü .....	115
<b>Şekil 3.85:</b> Google Coral Edge TPU.....	116
<b>Şekil 3.86:</b> Harici USB Ses Kartı .....	116
<b>Şekil 3.87:</b> Harici USB Ses Kartı aracılığıyla Raspberry Pi'ye Mikrofon Entegrasyonu .....	117
<b>Şekil 3.88:</b> İHA ve Yer Kontrol Merkezi İletişim Bağlantıları .....	120
<b>Şekil 3.89:</b> Mission Planner Yer Kontrol İstasyonu Yazılımı Arayüzü .....	120
<b>Şekil 3.90:</b> Raspberry Pi 4-Pixhawk 4 ve Pixhawk 4-GCS arasındaki MAVLink İletişimi .....	122
<b>Şekil 3.91:</b> Raspberry Pi'den Pixhawk 4 Uçuş Kontrol Kartına Komut Gönderme	122
<b>Şekil 3.92:</b> Gömülü Sistem Yazılım Mimarisi.....	123
<b>Şekil 4.1:</b> SSD MobileNetV2 Modeli İle Aynı Anda Dört Nesnenin Tespit Edilmesi ve Sınıflandırılması .....	126
<b>Şekil 4.2:</b> SSD MobileNetV2 Modeli İle Aynı Karede İki Farklı Sınıfa Ait Nesnelerin Tespit Edilmesi.....	126
<b>Şekil 4.3:</b> SSD MobileNetV2 Modeli İle İnsan ve Araba Tespiti .....	127
<b>Şekil 4.4:</b> SSD MobileNetV2 Modeli İle Araba Tespiti.....	127
<b>Şekil 4.5:</b> SSD MobileNetV2 Modeli İle Yoğun Güneş Işınları Altında Nesne Tespiti .....	128
<b>Şekil 4.6:</b> SSD MobileNetV2 Modeli İle Aynı Anda Üç İnsan Tespiti.....	128
<b>Şekil 4.7:</b> SSD MobileNetV2 Modeli ile İnsan ve Araba Tespiti.....	129
<b>Şekil 4.8:</b> SSD MobileNetV2 Modeli İle Eğimli Kamera Açısı İle Nesne Tespiti .	129
<b>Şekil 4.9:</b> YAMNet Ses Sınıflandırma Modelinin Deneysel Test Çalışma Ortamı	130
<b>Şekil 4.10:</b> Hexacopter'in Kalkışını Engellemek İçin İniş Takımı Ayaklarına Ağırlık Konulması.....	130
<b>Şekil 4.11:</b> Hexacopter'in Hover Modunda Belirli Mesafeden Helikopter Sesini Tanımasını Sağlayacak Test Ortamı.....	131
<b>Şekil 4.12:</b> Uçuş Hareket Komutlarının Kumandadan Gönderilmesi .....	132
<b>Şekil 4.13:</b> Hexacopter Hover Modunda Ses Sinyalinin (İHA ve Helikopter Sesi) Spektrogramı ve Ses Sınıflandırma Modelinin Tahmini.....	132
<b>Şekil 4.14:</b> Hexacopter Roll Left Hareketinde Ses Sinyalinin (İHA ve Helikopter Sesi) Spektrogramı ve Ses Sınıflandırma Modelinin Tahmini.....	133
<b>Şekil 4.15:</b> Hexacopter Roll Right Hareketinde Ses Sinyalinin (İHA ve Helikopter Sesi) Spektrogramı ve Ses Sınıflandırma Modelinin Tahmini.....	133
<b>Şekil 4.16:</b> Hexacopter Yaw Left Hareketinde Ses Sinyalinin (İHA ve Helikopter Sesi) Spektrogramı ve Ses Sınıflandırma Modelinin Tahmini.....	134
<b>Şekil 4.17:</b> Hexacopter Yaw Right Hareketinde Ses Sinyalinin (İHA ve Helikopter Sesi) Spektrogramı ve Ses Sınıflandırma Modelinin Tahmini.....	134
<b>Şekil 4.18:</b> Hexacopter Pitch Forward Hareketinde Ses Sinyalinin (İHA ve Helikopter Sesi) Spektrogramı ve Ses Sınıflandırma Modelinin Tahmini .....	135
<b>Şekil 4.19:</b> Hexacopter Pitch Back Hareketinde Ses Sinyalinin (İHA ve Helikopter Sesi) Spektrogramı ve Ses Sınıflandırma Modelinin Tahmini.....	135

# DERİN ÖĞRENME TABANLI SES VE GÖRÜNTÜ İŞLEME TEKONOLİLERİNE SAHİP OTONOM İNSANSIZ HAVA ARACI

## ÖZET

Bu çalışma, derin öğrenme tabanlı çevresel ses sınıflandırma teknolojisi ve bilgisayarlı görü tekniklerini bir araya getirerek gerçek zamanlı nesne algılama ve sınıflandırma işlemlerini gerçekleştirebilen ve bu özellikleri sayesinde otonom karar verebilen bir İnsansız Hava Aracı (İHA) platformunu tanıtmayı amaçlamaktadır. İHA'nın aynı anda ses ve görüntü işleme yeteneklerini kullanabilmesi, iki farklı veri kaynağından bilgi toplama ve analiz yapma olanağını sunar. Bu, İHA'nın hem çevresel sesleri dinleyebilme hem de çevresindeki nesnelere gözlemleme yeteneğini içerir. Ses işleme, platformun çevresel sesleri algılayarak ses tabanlı nesne tanıma ve sınıflandırma yapmasına olanak tanırken, görüntü işleme ise uçuş sırasında çevresindeki nesnelere görsel olarak algılayarak nesne tanıma ve sınıflandırma işlemlerini gerçekleştirebilir. Bu da İHA'nın tanımlı görevleri otomatik olarak gerçekleştirebilmesine olanak tanır.

Daha önceki bilimsel çalışmalar incelendiğinde nesne algılama ve sınıflandırmada, görüntü ve ses işleme teknolojisinin farklı alanlarda değişik amaçlar için ayrı ayrı kullanıldığı görülmektedir. İHA'ya otonom karar verme yeteneği kazandırmak ve gerçek zamanlı algılamanın yanı sıra kaydedilen görüntülerden ve seslerden nesne algılama ve sınıflandırma yapabilmek için gömülü bir yapay zekâ modülü gereklidir. İHA'nın, gömülü sistemler aracılığıyla hem ses hem de görüntü işleme teknolojilerini aynı anda kullanabilme yeteneğine sahip olması, nesne algılama ve sınıflandırma kapasitesini büyük ölçüde artırır ve ona önemli bir avantaj sağlar. Bu, İHA'nın çeşitli görevlerde daha etkili ve güvenilir bir şekilde kullanılmasına olanak tanır.

Önerilen yaklaşımda, görüntü ve ses işleme, İHA üzerine Raspberry Pi 4, kamera ve mikrofon monte edilerek gömülü gerçekleştirildi. Ayrıca uzaktan müdahale imkânı sağlanması için Raspberry Pi ile uyumlu 4G/LTE modem kit ve nesne algılama hızını arttırması için Coral Edge TPU yardımcı işlemcisi kullanıldı. Görüntü işleme için TensorFlow Kütüphanesi ve SSD MobilNetV2 evrişimsel sinir ağı modeli kullanıldı. Test uçuşları sırasında, araba için %96.3 ve insan için %96.2 doğruluk elde edildi. Ses sinyali işleme için derin öğrenme modeli olarak MobileNetV1 evrişimsel sinir ağı üzerine kurulu YAMNet derin öğrenme modeli seçildi. Ses sınıflandırması için yapılan uçuş testlerinde helikopter sesi kullanıldı ve modelin sesi doğru sınıflandırdığı gözlemlendi.

**Anahtar Kelimeler:** *İnsansız hava aracı, Bilgisayarlı görü, Derin öğrenme, Görüntü işleme, Nesne algılama, Evrişimsel sinir ağı, Raspberry pi, Ses Sınıflandırması.*

# AUTONOMOUS UNMANNED AERIAL VEHICLE WITH DEEP LEARNING-BASED AUDIO AND VISUAL PROCESSING TECHNOLOGIES

## ABSTRACT

This study aims to introduce an Unmanned Aerial Vehicle (UAV) platform capable of real-time object detection and classification by combining deep learning-based environmental sound classification technology with computer vision techniques. The UAV's ability to simultaneously utilize audio and visual processing capabilities allows it to gather information and analyze data from two different sources. This includes the UAV's capacity to both listen to environmental sounds and observe surrounding objects. Sound processing enables the platform to detect and classify objects based on sound, while image processing allows it to visually perceive and classify objects during flight. This enables the UAV to autonomously perform defined tasks.

Previous scientific studies indicate that in object detection and classification, image and sound processing technologies have been separately used for different purposes. To endow the UAV with autonomous decision-making abilities and enable real-time detection from recorded images and sounds, an embedded artificial intelligence module is required. The UAV's capability to simultaneously use both sound and image processing technologies through embedded systems significantly enhances its object detection and classification capacity, providing it with a significant advantage. This allows the UAV to be used more effectively and reliably in various missions.

In the proposed approach, image and sound processing were embedded onto the UAV using Raspberry Pi 4, a camera, and a microphone. Additionally, a 4G/LTE modem kit compatible with Raspberry Pi was utilized to enable remote intervention, and the Coral Edge TPU auxiliary processor was used to increase object detection speed. For image processing, the TensorFlow Library and the SSD MobilNetV2 convolutional neural network model were used. During test flights, accuracy rates of 96.3% for cars and 96.2% for humans were achieved. For sound signal processing, the YAMNet deep learning model built on the MobileNetV1 convolutional neural network was chosen. Helicopter sound was used during flight tests, and the model correctly classified the sound.

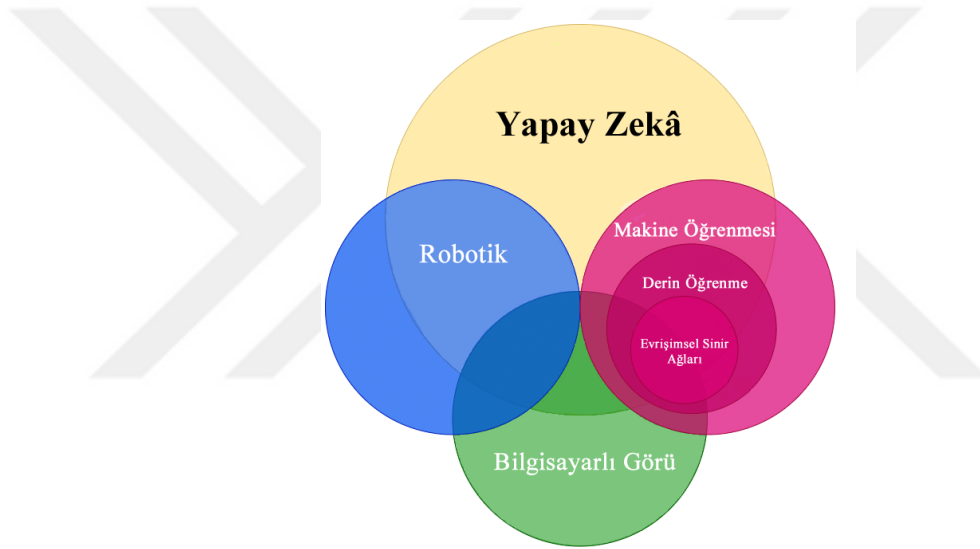
**Keywords:** *Unmanned aerial vehicle, Computer vision, Deep learning, image processing, Object detection, Conventional neural network, Rapsberry pi, Sound Classification.*

## 1. GİRİŞ

İnsansız Hava Aracı, bir yer kontrol merkezi tarafından kontrol edilebilen pilotsuz bir hava aracıdır. Kökeni askeri kullanımla ilişkilendirilse de [1] İHA'lar son yıllarda, bilimsel araştırma [2], lojistik [3], tarım [4], belgesel çekimi [5], itfaiye [6], sağlık [7] ve diğer birçok farklı alanda yaygın olarak kullanılmaktadır. İHA uygulamaları yaygınlaştıkça, güvenlik ve operasyonel verimliliği sağlamak, tanımlı görevleri yerine getirmek için ileri düzeyde otonomi gerekmektedir. İdeal olarak, otonom bir İHA, güvenli navigasyon için öncelikle sensörlere, mikroişlemcilere ve daha da önemlisi yerleşik yapay zekâyâ sahip olmalıdır. Mevcut sivil ve askeri insansız hava araçları, uçuş görevlerini yerine getirmek için uçuş kontrol sistemi ile birlikte bazı sensörler kullanılmaktadırlar [8] ve sınırlı yerleşik yapay zekâyâ sahiptirler. Çoğu durumda, İHA'lar, uçuş operasyonu için küresel konumlandırma sistemi (Global Positioning System - GPS), engel tespiti ve çarpışma önleme için sensörler kullanılmaktadırlar [9, 10]. Bu tip İHA'lar, operasyonel otonomiye ulaşmak için konum, hız ve irtifa verilerinin değiştirildiği algoritmalara sahip uçuş kontrol sistemlerini kullanırlar [11]. Bu tür sistemleri barındıran İHA'ların karmaşık görevleri yerine getirmedeki kabiliyetleri sınırlıdır. İHA'ların karar verme konusunda otonom olabilmesi ve uçuş planlaması dışında belirli görevleri icra edebilmesi ancak sistemin bilgisayarla görmede (bilgisayarlı görü) sırasıyla algılama ve sınıflandırma olarak adlandırılan nesne görme ve kavrama gibi ikili işlevleri yerine getirebilmesiyle mümkündür [12-15]. Bu görevler insanlar için doğal olsa da, makinelerin kendi başlarına gerçekleştirmesi için soyut ve karmaşıktır. Bu sebeple, otonom İHA uygulamalarının birçoğunu gerçekleştirmek için yapay zekâ alanındaki bilgisayarlı görü teknolojisi hayati bir role sahiptir [16].

Genel anlamda yapay zekâ, bir makine veya sistem tarafından herhangi bir insan davranışının taklit edilmesini temsil eder. Makine öğrenmesi, yapay zekânın bir alt sınıfıdır [17]. Günümüzde makine öğrenmesindeki gelişmeler, makine öğrenmesinin bir alt sınıfı olan bilgisayarlı görü alanında giderek daha güçlü yöntemlerin geliştirilmesini sağlamıştır ve bu alandaki ilerlemeleri

kolaylaştırmaktadır. Bilgisayarlı görü, insan görüş sisteminin yürüttüğü, görüntülerde nesne ayırt etme, yüz tanıma ve hatta mimik tanıma vb. görevleri taklit etmeye çalışır ki tüm bunlar yine makine öğrenmesinin bir alt sınıfı olan derin öğrenme mimarisindeki yapay sinir ağlarının kullanılması ve eğitilmesiyle mümkündür. Tüm bu alanlar ve teknolojiler birbiriyle iç içedir. Şekil 1.1’de yapay zekâ teknolojilerinin birbiriyle olan bağlantıları gösterilmiştir. Nesne algılama, görüntülerde yer alan nesnelerin belirlenmesi için kullanılan bir bilgisayarlı görü tekniğidir. Nesne algılama, görüntü verilerinde nesnelerin tespit edilmesi ve sınıflandırılmasıdır. Tespit edilen nesnenin özelliklerine bakılarak nesnenin önceden tanımlanmış sınıflardan hangisine ait olduğu çıkarılır, bu işlem nesne sınıflandırma olarak adlandırılır. Nesne tespiti ve sınıflandırma, yapay zekâ sistemlerinin en temel uygulamalarındandır.



**Şekil 1.1:** Yapay Zekâ, Makine Öğrenmesi, Derin Öğrenme, Bilgisayarlı Görü ve Evrişimsel Sinir Ağları Arasındaki İlişki

Yapay zekâ ve makine öğrenmesinin otonom İHA operasyonlarıyla entegrasyonunda karşılaşılan sorunlardan en önemlisi, nesne algılama ve sınırlandırma görevlerinin işlem hızı ve doğruluğu nedeniyle gerçek zamanlı veya gerçek zamanlıya yakın bir şekilde yürütülememesidir. Bu çalışmanın amaçlarından biri, görüntü ve ses işleme teknikleri ile ham verilerden nesnelere gerçek zamanlı olarak tespit etmek ve sınıflandırmak için evrişimsel sinir ağı algoritmalarını kullanan derin öğrenme tabanlı bir yazılımın geliştirilmesidir.

Derin öğrenme, Grafik İşleme Birimlerinin [18] (Graphics Processing Units-GPUs) ve Tensor İşleme Birimlerinin [19] (Tensor Processing Units-TPUs) gelişimine paralel olarak son araştırmalarda görüntü içeriğinde nesne tespiti ve insan

algılamayı gerçekleştirmek için kullanılmaya başlanmıştır. Derin öğrenme, belirli bir tipik nesneyi veya nesnelere gerçek zamanlı olarak tespit edebilmesi için ağı eğitim örneklerine dayalı olarak eğitmemizi sağlar. Derin öğrenme mimarisinde girdi olarak kullanılan görüntülerdeki desenleri tespit etmek için özelleştirilmiş sinir ağları oluşturulmuştur, bu sinir ağları evrimsel sinir ağları olarak adlandırılır ve adını görüntüde yer alan örneğin insan vücudu, arabalar, binalar, hayvanlar ve diğerleri gibi karmaşık özelliklerin belirli filtreler ile evrim işlemi gerçekleştirilmesinden almaktadır. Günümüzde yapılan çalışmalarda insansız hava aracı ile elde edilen görüntülerin işlenmesi için farklı yöntemler kullanılmaktadır. Yöntemlerin birisinde, kaydedilen görüntüler veya sesler wi-fi ile fiili işlemlerin gerçekleştiği bir yer kontrol istasyonuna gönderilir [20], diğer yöntemde ise kaydedilen görüntüler veya sesler internet aracılığıyla bulut tabanlı (sanal) bir merkeze gönderilir [21] ve her iki yöntemde görüntü ve ses verileri uzak sistemlerde işlenerek sonuçları İHA'ya aktarılır. Bu tür sistemler ne yazık ki hem yavaş kalmaktadır hem de çok geniş alanlarda kullanılamamaktadır.

Bu çalışmada önerilen ve uygulanan yöntem, görüntü işlemeyi İHA üzerinde gömülü olarak yapma fikridir. Bunun için İHA üzerine yerleştirilecek bir mini bilgisayara gereksinim vardır, böylece kameradan elde edilen video ve görüntü bilgileri doğrudan bu mini bilgisayar içine beslenir ve görüntü işleme gerçekleştirilir. Bu çalışmada kullanılacak derin öğrenme modeli ve kütüphanesine uygun olarak Raspberry Pi mini bilgisayarın) Raspberry Pi 4 B Model verisyonu seçilmiştir. Raspberry Pi 4'ün derin öğrenme modellerini çalıştırılmasını sağlayan şey, IoT (Nesnelerin İnterneti) için TensorFlow uçtan uca açık kaynaklı makine öğrenmesi platformunun (TensorFlow Lite) gelişidir [22].

Daha önce de bahsedildiği gibi şu anda otonom İHA operasyonunun karşılaştığı sorunlardan biri, algılama ve sınıflandırma işlemlerini gerçek zamanlı veya gerçek zamanlıya yakın olarak yürütememektir. Bu sorunu çözebilecek, askeri ve sivil uygulamalarda otonom İHA operasyonlarını yürütmek için kullanılacak nesne algılama ve sınıflandırma görevlerini gerçekleştirmek için yüksek performanslı tek atış çoklu kutu dedektörü (Single Shot Multibox Detector-SSD) nesne algılama yöntemini ve evrimsel sinir ağı mimarisini içeren SSD MobilNetV2 derin öğrenme modeli seçilmiştir. Derin öğrenme tabanlı evrimsel sinir ağı mimarisi, İHA'nın yakın çevresinde yer alan nesne bilgilerini insan müdahalesi olmadan makineler

tarafından yorumlanabilecek soyut bilgilere dönüştürmesini sağlar. Mevcut bilgilere dayanarak, makineler gerçek zamanlı karar verme işlemini gerçekleştirebilir. Bir İHA'nın uçuş kontrol sistemine, derin öğrenme tabanlı evrişimsel sinir ağını içeren görüntü işleme teknolojisinin entegre edilmesi, İHA'nın otonom karar verme kabiliyetini ve uçuş güvenliğini önemli ölçüde artırabilir. Diğer makine öğrenmesi yöntemleri ile karşılaştırıldığında evrişimsel sinir ağı algoritmalarının ana avantajı, nesnelere hesaplama açısından daha hızlı ve üstün performansla gerçek zamanlı olarak algılayıp sınıflandırabilmeleridir. Bu çalışmada kullanılan evrişimsel sinir ağı algoritması, derin öğrenme algoritmaları ile gelişmiş TPU teknolojisinin [23] birleşimine dayanmaktadır.

İnsansız hava aracı, yerleşik bilgisayar ve görüntü işleme yeteneklerine sahip olduğu için kamera tarafından yakalanan görüntü verilerini merkezi bir bilgi işlem motoruna ihtiyaç duymadan işleyebilir, bu verilere dayanarak nesneyi tespit edebilir ve daha önceden kendisine aktarılmış bilgilere göre otonom karar verebilir. Bilgisayarlı görü teknolojisine sahip bu İHA platformu hakkında daha fazla ayrıntı vermeden önce, yapay zekâ, makine öğrenmesi, derin öğrenme, sinir ağları ve ilişkili diğer konular hakkında bazı temel bilgilere yer verilecektir.

Bu çalışmada, İHA kamerası tarafından sağlanan görüntü ve video verilerinden gerçek zamanlı nesne tespit ve sınıflandırma işlemlerini yapabilen ve bu işlemlerin sonucunda elde edilen verilere göre İHA'na otonom karar verme ve hareket etme kabiliyeti sağlayan gömülü yazılım geliştirildi. Nesne tespit ve sınıflandırma görevleri için TensorFlow Kütüphanesi ve SSD MobilNetV2 evrişimsel sinir ağı mimarisi kullanıldı.

Bu model, iki farklı görev için ayrı ayrı sıfırdan eğitildi ve özelleştirildi. Transfer öğrenme yöntemi ile iki ayrı düzlemde eğitilen derin öğrenme modeli, daha sonra Edge TPU yardımcı işlemcisi ile Raspberry Pi'de çalışacak şekilde "TensorFlow Lite" formatına dönüştürüldü. İHA platformunun kullanım alanı, günümüzde sıklıkla ismini drone olarak duyuran İHA'ların askeri alanda çok tercih olması sebebiyle askeri alan olarak hedeflenerek tasarlandı. Nesne olarak askeri helikopterin, görüntü ve ses verisi ile İHA platformu tarafından tespit edilebileceği hedeflendi. Ancak, uçuş sahasında askeri helikopterin olmasının mümkün olmayacağı bilindiğinden görüntü ile nesne tespit ve sınıflandırma yöntemi, uçuş sahasında daha kolay bulunabilir insan ve araba nesnelere üzerinden test edildi. Bu

iki nesne için bilgisayarlı görü alanında yapılan çalışmalarda transfer öğrenme yöntemi ile modelin eğitim süreci bilindiğinden ve daha kolay olduğundan, daha önce çalışılmamış nesne olarak askeri helikopter seçildi. Bu sebeple, bu çalışmada, sinir ağı modelinin transfer öğrenme yöntemi ile yeniden eğitim süreci ve özelleştirilmesi bu nesne üzerinden anlatıldı.

Bu çalışmanın yenilikçi yönü, derin öğrenme temelli bilgisayarlı görü tekniklerini ve yine derin öğrenme tabanlı çevresel ses sınıflandırma teknolojisini bir araya getirerek gerçek zamanlı nesne algılama ve sınıflandırma işlemlerini gerçekleştirebilen bir İnsansız Hava Aracı (İHA) platformunu tanıtmayı amaçlamasıdır. Bu platform, çevresel seslerin ve görüntülerin işlenmesi ve anlamlandırılması konusunda gelişmiş yeteneklere sahiptir, bu da onun tanımlı görevleri otomatik olarak gerçekleştirebilmesine olanak tanır.

İHA platformu, çevresel sesleri analiz etmek ve sınıflandırmak için derin öğrenme teknolojisini kullanır. Bu, platformun uçuş sırasında çevresel sesleri algılayarak, tanımlı ses kalıplarını tespit etmesini ve bu seslere ilişkin bilgi toplamasını sağlar. Bu sayede İHA, çeşitli görevlerin gereksinimlerine uygun olarak ses sinyallerini yorumlayabilir ve bu bilgiyi daha fazla anlamlandırmak için görüntü analizi ile birleştirebilir.

İHA platformu, aynı zamanda, bilgisayarlı görü teknikleri ile gerçek zamanlı nesne algılama ve sınıflandırma yeteneklerine sahiptir. İHA'nın kamerası, uçuş sırasında görüntülerin sürekli olarak kaydedilmesini ve bu görüntülerin derin öğrenme algoritmaları ile işlenmesini sağlar. Bu, platformun uçuş alanındaki nesnelere tespit etmesini, bu nesnelere sınıflandırmasını ve izlemesini sağlar.

Sonuç olarak, bu İHA platformu, çevresel seslerin ve görüntülerin işlenmesi için derin öğrenme ve bilgisayarlı görü tekniklerini bir araya getirerek çeşitli otomatik görevleri gerçekleştirebilir. Bu, araştırma, güvenlik, arama kurtarma ve bilhassa askeri alan dâhil diğer birçok uygulama alanında kullanılabilir. İHA, yetenekleri sayesinde, insan müdahalesine gerek duymadan karmaşık görevleri gerçekleştirebilme kapasitesine sahiptir, bu da çeşitli alanlarda verimliliği ve etkinliği artırabilir.

## 1.1 İnsansız Hava Araçlarının Kullanım Alanları

Son yıllarda, İHA'ların kullanım alanları önemli ölçüde genişlemiştir. İstihbarat, uzaktan algılama, harita oluşturma, arama ve kurtarma operasyonları, yaban hayatın izlenmesi, çevresel kirlilik tespiti, meteorolojik araştırmalar, tarımsal uygulamalar, reklam ve film prodüksiyonu, doğal afet izleme ve arkeolojik çalışmalar gibi birçok farklı alanda kullanılmaktadır [24]. İHA'ların bir kısım kullanım alanları aşağıda listelenmektedir [25, 26].

### - *Harita Mühendisliği Uygulamaları*

- 3 Boyutlu şehir modelleme
- Ortofoto harita üretimi
- Kentsel dönüşüm alanlarının ölçümü
- Maden sahalarının ölçümü
- Arkeolojik çalışma alanlarının ölçümü vb.

### - *Askeri Uygulamalar*

- Sınırların kontrolü ve korunması
- Mayın arama ve imha
- Kaçakçılık ile mücadele
- Özel operasyonlar ve psikolojik harekât
- Gözetleme ve keşif vb.

### - *Endüstriyel Uygulamalar*

- Altyapı izleme ve denetleme
- Petrol hattı izleme ve denetleme
- Yüksek gerilim hattı izleme ve denetleme
- Rüzgâr türbinleri izleme ve denetleme
- Güneş paneli izleme ve denetleme vb.

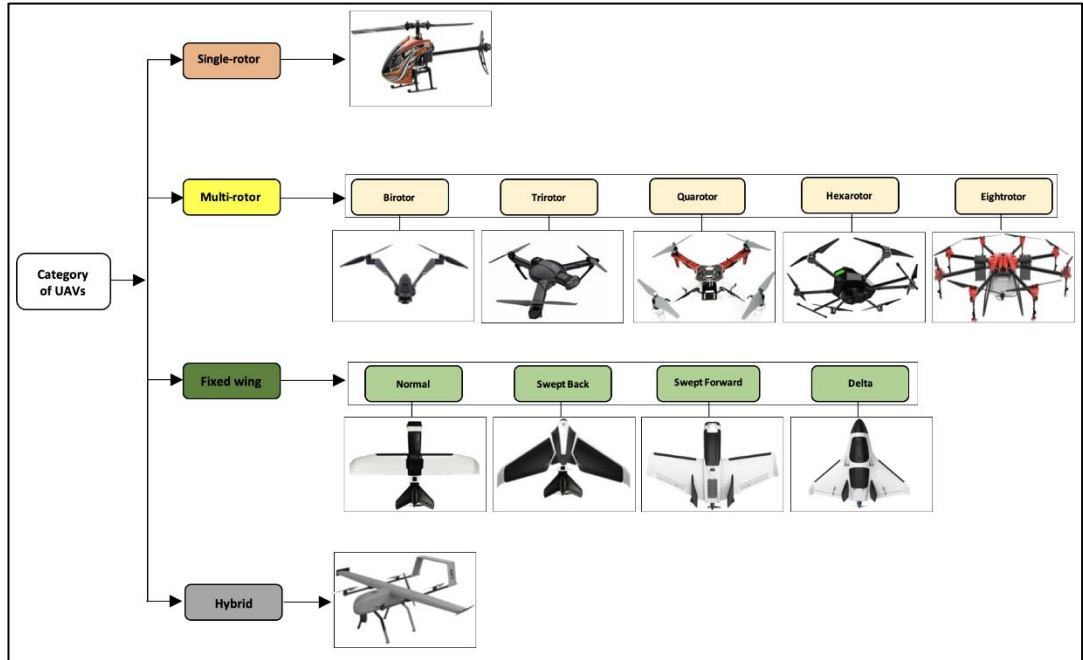
### - *Tarım Uygulamaları*

- Hastalık teşhisi

- Zirai ilaçlama
- Ağaç ve bitki gelişimi izleme
- Rekolte tahmini
- Bitki ve ağaç sulama planlaması vb.
- *Lojistik Uygulamaları*
- Sipariş teslimat
- Sağlık malzemesi taşıma
- Taksi uygulamaları

## 1.2 İnsansız Hava Araçlarının Kategorileri ve Türkiye'deki Sınıflandırılması

Son yıllarda, İnsansız Hava Araçları üzerindeki artan ilgi, şekil ve boyut açısından farklı yapılar ve bileşenlerle donatılmış çeşitli İHA türlerinin ortaya çıkmasına yol açmıştır. İHA'lar, Şekil 1.2'de gösterilen dört türe ayrılmıştır: tek rotorlu (single-rotor), çok rotorlu (multi-rotor), sabit kanatlı (fixed wing) ve hibrit (hybrid) [27].



**Şekil 1.2:** Yapılarına Göre İHA Kategorileri

**Kaynak:** (Adoni, W.Y.H., et al, 2023)

Sivil Havacılık Genel Müdürlüğü (SHGM), 2016 yılı SHT-İHA Talimatnamesi [28] Madde 5' göre İHA'lar azami kalkış ağırlıkları referans alınarak dört ayrı sınıfa ayrılır:

- a) İHA0: Azami kalkış ağırlığı 500 gr (dâhil) – 4kg aralığında olan İHA'lar,
- b) İHA1: Azami kalkış ağırlığı 4 kg (dâhil) – 25 kg aralığında olan İHA'lar,
- c) İHA2: Azami kalkış ağırlığı 25 kg (dâhil) – 150 kg aralığında olan İHA'lar,
- d) İHA3: Azami kalkış ağırlığı 150 kg (dâhil) ve daha fazla olan İHA'lar, olarak sınıflandırılmaktadır.

**Çizelge 1.1: SHGM İHA Sınıflandırması**

Sınıf	Kütle
İHA0	500 g (dâhil) – 4 kg aralığı
İHA1	4 kg (dâhil) – 25 kg aralığı
İHA2	25 kg (dâhil) – 150 kg aralığı
İHA3	150 kg (dâhil) ve daha fazla

İHA pilotunun, bu sınıflandırmaya göre sertifika sahibi olması gerekir.

### 1.3 Literatür Araştırması

Bu tez, derin öğrenme tabanlı ses sinyali ve görüntü işleme teknolojileri ile nesne tespit edebilen bir İnsansız Hava Aracı (İHA) tasarımı geliştirmeyi amaçlamaktadır. Bu İHA, ses ve görüntü verilerini analiz edebilme yeteneğine sahip olacak ve bu analizler sonucunda nesnelere tespit edebilecektir. Bu yetenekler, İHA'nın çevresini algılamasına ve anlamasına olanak tanırken, aynı zamanda otonom kararlar almasını da mümkün kılacaktır.

İHA tasarımının ana odak noktası, derin sinir ağı modellerini belirli bir nesne veya nesnelere tespiti için yeniden eğitmek ve modellemek, bu modelleri kullanarak ses ve görüntü verilerini işleyebilen ve nesnelere tanıyabilen yapay zekâ modülü oluşturmaktır. Bu modül, İHA'nın çevresini sürekli olarak tarayacak ve potansiyel tehditleri, hedefleri veya diğer nesnelere tanımlayacaktır. Bunun sonucunda, İHA kendi başına kararlar alabilecek veya belirli görevleri yerine getirebilecektir. Bu, İHA'nın otonom bir şekilde hareket etmesini ve görevlerini daha etkili bir şekilde yerine getirmesini sağlayacaktır.

Bu çerçevede konu ile ilgili detaylı literatür taraması yapıldı, görüntü ve ses sinyali işleme teknolojilerinin hangi amaçla hangi alanlarda kullanıldığı incelendi.

Yamazaki ve arkadaşları [29] çalışmalarında, deprem bölgelerinde enkaz altında kalan afetzedelerin tespiti amacıyla İnsansız Hava Aracı (İHA) üzerinde ses işleme teknolojisini kullanmayı hedeflemiştir. Bu çalışmada, ses tanıma işlemi için açık kaynaklı bir yazılım olan Julius kullanılmıştır.

Önerdikleri yöntemde, İHA üzerinde bulunan bir hoparlör, enkaz altındaki afetzedelerin dikkatini çekmek ve tepki vermelerini sağlamak için ses üretir. Bu sesi duyan afetzedelerin hangi kelimeleri kullanarak yardım çağırdığını belirlemek için bir veri tabanı kullanılır. Afetzedelerin yardım için hangi kelime veya kelimeleri kullandığı, aynı zamanda afetzedelerin konum bilgisi ile eşleştirilerek bu bilgiler metin formatında Raspberry Pi üzerindeki SD kartta saklanır. İHA kontrol merkezine döndüğünde, kurtarma ekipleri bu verileri toplar ve kurtarma çalışmalarını bu bilgilere göre yönlendirirler. Yamazaki ve arkadaşları bu yöntemde, afetzedelerin sesleri üzerinden yardım çağrılarını tespit etmeye ve afetzedelerin konum bilgilerini kaydetmeye dayalı bir yenilikçi yaklaşım sunmuşlardır. Bu çalışmanın, afetzedelere daha hızlı ve etkili bir yardım sağlayacağını amaçlamışlardır.

Carlos ve arkadaşlarının [30] çalışmasında, gerçek zamanlı ses olaylarını sınıflandırabilen, tespit edilen olayların konumunu ve ilişkilendirilmiş bilgilerini güncelleyebilen bir sistem tanıtılmıştır. Bu çalışmada, gerçek zamanlı ses sınıflandırması için önceden eğitilmiş YAMNet modelini kullandıklarını belirtmişlerdir. Bu sistemin, engelli bireylerin yardımına yönelik farklı ortamlarda uygulanabileceğini vurgulamışlardır.

Kumar ve arkadaşları [31], Parkinson hastalığının erken hastalık tespiti için invaziv olmayan bir yöntem olarak konuşma sinyallerinin kullanılabileceğini, manuel konuşma sinyali analizinin yorucu ve hata yapmaya açık olabileceğini vurgulamaktadır. Bu nedenle, prosedürü hızlandırmak ve doğruluğu artırmak için derin öğrenme temelli yöntemler önermişlerdir. Veri kümesinden özellikler çıkarmak için hesaplama açısından verimli gördükleri ses sınıflandırma modeli YAMNet'i kullandıklarını belirtmişlerdir. Konuşma sinyallerinin analizi kullanılarak, Parkinson hastalığının erken tanı ve yönetimi için hassas ve etkili bir araç oluşturulmaya çalışılmıştır.

Tena ve arkadaşları [32] yaptıkları çalışmada, COVID-19'un kolayca tespiti için ses sınıflandırmadan yararlanmışlardır. COVID-19 ve COVID-19 olmayan hastalara ait öksürüklerin olduğu veri kümesini ve MobilNetV1 evrimsel sinir ağı mimarisi üzerine kurulu YAMNet ses sınıflandırma modelini kullanarak özellik çıkarımı yapmış ve bu sayede COVID-19'lu hastaları tespit etmişlerdir.

Roy ve Satija [33] çalışmalarında, Kronik Obstrüktif Akciğer Hastalıklarının (KOA) ciddiyetinin tespiti için akciğer seslerinin zaman-frekans mel spektrogram gösteriminden ve YAMNet tabanlı transfer öğrenme modelinden yararlanmışlardır.

Gupta ve arkadaşları [34], İnsansız hava aracı sistemi yardımıyla askeri aracın belirli bir görüntü veya video karesinden tespit edilmesi ve tanınması, üzerinde çalışmışlardır. Askeri kamyonlar, tanklar, uçaklar, helikopterler ile sivil araç ve uçak sınıflarına sahip 6772 görüntüden oluşan veri setini kullanarak Quantize SSD MobilenetV2 modelini eğiterek havadan elde ettikleri görüntüler ile yerde bulunan bu araçları tespit edebilmişlerdir.

Tighkhorshid ve arkadaşları [35] çalışmalarında, Covid-19 gibi solunum yolu hastalıklarının yayılmasını önlemek amacıyla görüntü işlemeyi kullanarak insanların yüz maskesi takma koşullarını üç sınıfa ayıran bir sistem geliştirmiştir: yüz maskesi olmayanlar, yüz maskesini doğru takanlar ve yüz maskesini yanlış takanlar. Bu amaçla, SSD MobileNetV2 sinir ağı kullanılmış ve mümkün olan en iyi doğruluk için model üzerinde çeşitli hiperparametre değişiklikleri yapmışlardır. Önerilen sinir ağını, bir Raspberry Pi 3 cihazı üzerinde uygulamışlardır.

Latina ve arkadaşları [36], cam şişelerin geri dönüşüme uygun olup olmadığını belirlemek amacıyla, cam şişelerinin makro ve mikro boyuttaki genel kusurlarını tespit etmek için SSD MobileNetV2 modelini kullanan, derin öğrenmeye dayalı bir yöntem sunmuşlardır. Transfer öğrenimi ve veri artırma yöntemini kullanarak sistemin %98,07'ye varan genel sistem doğruluğuyla cam şişe kusurlarını tespit etmişlerdir.

Mohammed ve arkadaşları [37] çalışmalarında, RF sinyallerini ses sinyallerine dönüştürmüş ve YAMNet ses sınıflandırma modelini kullanarak bu ses sinyalleri üzerinden drone model türlerini belirlemişlerdir.

Fairuzi ve Zülkifli [38], havalimanı pistinde uçağa zarar verebilecek ve kazaya sebep olabilecek yabancı nesnelere gerçek zamanlı olarak algılayabilen bir

güvenlik sistemi geliřtirmişlerdir. SSD ve YOLO'nun, yüksek FPS ve doğruluk performansları nedeniyle gerçek zamanlı tespit için en sık kullanılan yöntemler olduğunu belirterek SSD MobileNetV2 ve YOLOV4 modellerini kullanmış ve iki modeli kıyaslamışlardır.

Junior ve Suharjito [39] çalışmalarında, palm yağı olgunluk seviyesini tespit etmek için nesne tespiti ve video veri kümeleri üzerine kurulan gerçek zamanlı bir yaklaşıma dayalı bir yöntem önermişlerdir. Derin öğrenme modeli olarak YOLOV4 sinir ağı mimarisini kullanmışlardır. YOLOV4 modelinin sonuçlarını doğrulamak için SSD MobileNetV2 FPN ve EfficientDet-D0 karşılaştırmasını yapmışlardır.

Literatürde eksik olduğu değerlendirilen iki durum bu tez çalışmasının kapsamını oluşturmaktadır. İnsansız hava aracı ile yapılan nesne tanıma çalışmalarında seçilen modelin veri setleri üzerinden eğitildiği, belirli bir göreve özel bir modelleme çalışmasının yapılmadığı görülmektedir. Ayrıca, ses ile nesne sınıflandırma çalışmalarının sadece İHA'ya özgü değil genel anlamda kısıtlı olduğu, daha çok tıptaki hastalıkların teşhisi [40-44] ve yer sistemleri ile drone tespiti çalışmalarında [45-49] kullanıldığı görülmektedir.

Bu çalışmada, İHA ile nesne algılama ve sınıflandırmada, derin öğrenme tabanlı görüntü ve ses işleme teknolojilerini aynı anda kullanabilen gömülü bir sistem tasarlandı. Burada amaç nesne tespitinde iki farklı veri kaynağını kullanarak nesneyi en doğru şekilde tespit edebilmektir ve bu da görülen en önemli eksikti. İHA, hem görüntü hem de ses veri üzerinden nesne tespit ve sınıflandırma görevlerini yürütebileceğinden, uçuş koşullarının görüntü verisini elde etmeye imkân vermediği durumlarda ses işleme ve uçuş koşullarının ses versini elde etmeye imkân vermediği durumlarda görüntü işleme teknoloji ile görevler icra edilebilecektir. Diğer eksik durum ise, görüntü ile nesne tespit çalışmalarında, İHA sadece havadan nesne tespit amaçlı kullanılmakta, bu kapsamda veriler ya uçuş sonrası yerdeki merkezde işlenmekte ya da İHA havadan merkeze anlık bilgi göndermektedir.

Bu çalışmada, ses ve görüntü ile nesne tespit ve sınıflandırmanın bir sonucu olarak gömülü sistem, İHA'nın kontrol merkezine göreve özgü komutları gönderebilmekte ve İHA'ya otonom karar verme yeteneği kazandırmaktadır.

#### 1.4 Tezin Amacı

Bu tezin amacı, derin öğrenme tabanlı ses ve görüntü işleme teknolojilerini kullanarak, nesne tanıma ve sınıflandırma yeteneklerini hem ayrı ayrı hem de birbirini doğrulayıcı bir şekilde aynı anda gerçekleştirebilen ve bu yeteneklerin bir sonucu olarak daha önce tanımlanmış görevleri otonom olarak yerine getirebilen gömülü bir sistem tasarlamak ve bu sistemin uygulandığı bir İnsansız Hava Aracı (İHA) üretmektir.

Günümüzde, insansız hava araçları, gözetleme, keşif, sağlık, lojistik ve tarım gibi birçok farklı uygulama alanında kullanılmaktadır. Ancak, İHA'ların yeteneklerini artırmak ve daha karmaşık görevleri yerine getirebilmelerini sağlamak için yeni teknolojilere ihtiyaç vardır. Bu tez, derin öğrenme tabanlı ses ve görüntü işleme teknolojilerinin kullanılmasıyla İHA'ların yeteneklerini geliştirmeyi amaçlamaktadır. Bu amaç doğrultusunda, tez kapsamında şu ana hedefler belirlenmiştir:

**Ses Sinyali İşleme Teknolojisi:** İHA'ya ses sınıflandırma yetenekleri kazandırmak için derin öğrenme yöntemlerini kullanmak ve bu yöntemleri özelleştirmek. Bu sayede İHA, ses kaynaklarını veya nesnelere tanımlayabilir ve sınıflandırabilir.

**Görüntü İşleme Teknolojisi:** İHA'ya görüntü tanıma ve sınıflandırma yetenekleri kazandırmak için derin öğrenme ağlarını kullanmak ve ağları göreve özgü özelleştirmek. İHA, görüntüleri analiz edebilir, nesnelere tanımlayabilir ve bu nesnelere özelliklerini sınıflandırabilir.

**Çapraz Doğrulama:** Ses ve görüntü işleme yeteneklerini bir araya getirerek daha güvenilir ve kesin sonuçlar elde etmek. Ses ve görüntü verilerini birbirini doğrulayan şekilde kullanarak İHA'nın çevresini daha iyi anlamasını sağlamak.

**Otonom Görev Yürütme:** İHA'nın bu yeteneklerini kullanarak daha önce tanımlanmış görevleri otonom olarak gerçekleştirebilme kabiliyetini geliştirmek. İHA, insan müdahalesi olmadan karmaşık görevleri yerine getirebilir, örneğin arama kurtarma operasyonlarında kayıp kişileri bulabilir veya muharebe alanında tanımlı bir askeri hava-kara araçlarını otomatik olarak tarayabilir.

Bu tez, hem savunma hem de sivil uygulamalarda İHA'ların işlevselliğini artırmak için önemli bir katkı sağlamayı amaçlamaktadır. Derin öğrenme tabanlı ses ve görüntü işleme teknolojilerinin kullanılması, İHA'ların daha etkili, verimli ve otonom bir şekilde çalışmalarını sağlayarak çeşitli endüstriler için önemli faydalar sunabilir.



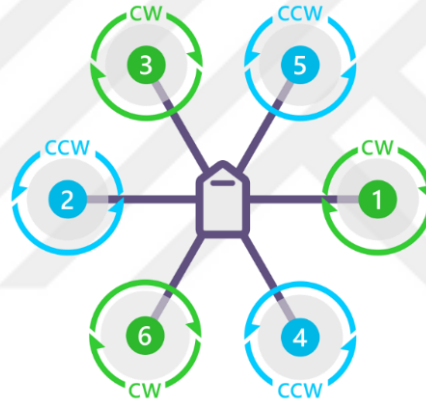
## 2. İNSANSIZ HAVA ARACI TASARIMI

Teknolojik ilerlemeler paralelinde, sabit kanatlı ve döner kanatlı insansız hava araçlarının (İHA) kullanım alanları genişlemekte ve bu araçların yetenekleri de önemli ölçüde artmaktadır [50]. İHA'lar, hızlı üretim, düşük maliyet, tehlikeli görevlerde kullanılma [51], insanların erişemeyeceği durumlarda görev yapma [52] ve geleneksel uçaklara kıyasla pratik donanım güncellemeleri gibi avantajları sayesinde artık günlük yaşamın ayrılmaz bir parçası haline gelmiştir. Özellikle askeri alanlarda, mevcut otonom sistemlerin bilgi ve kapasite sınırlarını zorlayarak üstün donanım ve yazılım özelliklerine sahip İHA'lar geliştirilmektedir [53]. Ayrıca, İHA'lara kolay erişim, araştırmacıların ve farklı alanlardaki profesyonellerin daha geniş bir perspektifle katkı sağlamasına imkân tanımış, böylece daha yenilikçi uygulamaların ortaya çıkmasına olanak sağlamıştır [54]. Bu artan talep nedeniyle, İHA'ların yazılım ve donanımları görevlerine özgü olarak düzenli olarak güncellenmekte ve daha fazla otonom yetenek kazandırılmaktadır [55]. Bu insansız sistemler, son zamanlarda yapay zekâ alanının popüler çıktılarından biri haline gelmektedir.

Döner kanatlı sistemler, yüksek manevra kabiliyetleri sayesinde sabit kanatlı İHA'lara göre önemli avantajlar sunar. Sahip oldukları manevra yeteneği, onları, son derece dar uçuş alanlarında bile kullanılabilme olanağı sunarak birçok potansiyel uygulamanın gerçekleştirilmesini de mümkün kılar [56]. Dört rotorlu (motorlu) İHA'lar, döner kanatlı sistemlerin en yaygın olanları olabilir, ancak altı rotorlu İHA'lar (Hexacopter) gibi başka önemli varyantlar da popülerliğini sürdürmektedir. Altı rotorlu insansız sistemler, ana gövdenin etrafında eşit aralıklarla yerleştirilmiş altı motor tarafından kontrol edilir. Altı rotorlu İHA'ları özel kılan önemli avantajlardan biri, yüksek irtifada ve hatta zorlu hava koşullarında uçabilme yetenekleridir [57]. Güçlü rotorları sayesinde, yoğun ve zorlu hava koşullarına maruz kaldığında bile uçuşa devam edilebilirler ve kontrol altında tutulabilirler [58]. Bir Hexacopterde altı motor kullanılması, sadece manevra kabiliyetini artırmakla kalmaz, aynı zamanda bir rotor arızalandığında bile uçuşu sürdürebilmesini sağlar

[59]. Bu yetenekler, bir insansız hava aracında aranan taktik özelliklerdir [60]. Bu tür taktik özelliklere ve bilhassa havada istikrarlı bir şekilde kalabilme kabiliyetine sahip Hexacopter, gözetim ve istihbarat amaçları için geleneksel modellere alternatif bir platform olarak düşünülebilir [61]. Bu sistemler, ayrıca ağır yükler taşıyabilir ve geniş bir sensör yelpazesi ile donatılabilirler. Günümüzde, hexacopterlar, yazılım ve donanım gelişmeleri ile daha fazla kullanım alanı bulmaktadır [62].

Bu çalışmada yer alan insansız hava aracını diğerlerinden ayıran özellik nesne tespitinde derin öğrenme tabanlı görüntü ve ses işleme teknolojilerini aynı anda kullanabilmesidir. Amaca uygun olması için kolay programlanabilir açık kaynak kodlu donanımlara sahip İHA'ya gereksinim duyuldu. Bu sebeple hazır bir İHA almak yerine Şekil 2.1'deki X-konfigürasyonuna sahip Hexacopter İHA sıfırdan tasarlanmıştır.



**HEXA X**

**Şekil 2.1:** Hexacopter Yapısı

**Kaynak:** (ardupilot.org, 2023)

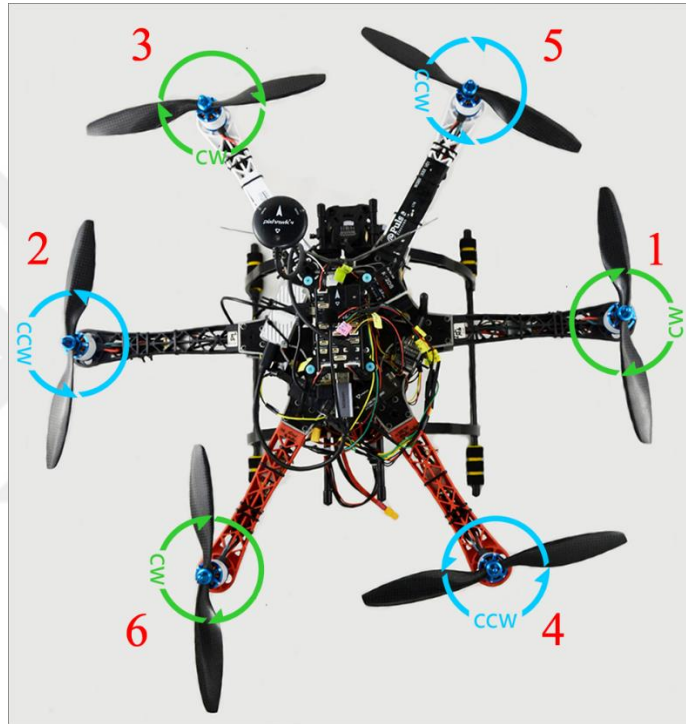
## 2.1 Hexacopter İnsansız Hava Aracı Çalışma Mekanizması

Hexacopter, altı motorlu (rotorlu) bir insansız hava aracıdır. Şekil 2.2'de görüldüğü gibi, Hexacopter'de, üç adet saat yönünde (Clockwise-CW) dönen motor-pervane ve üç adet saat yönünün tersine (Counterclockwise-CCW) dönen motor-pervane kullanılır. Bu motorların dönme hızı ve dönme yönü, Hexacopter'in kalkış-iniş, ileri-geri (pitch-lateral axis), sağ-sol (roll-longitudinal axis) ve tam dönüş (yaw-vertical axis) hareketini sağlar. Altı motor, uçuş kontrol kartı üzerinden gelen komutlara göre farklı itki oluşturur ve hexacopter, bu motorların hızına uygun şekilde asılı kalır veya hareket eder. Askı durumunda, tüm pervaneler yaklaşık olarak aynı hızda çalıştırılır

ve bu nedenle motorlarda yaklaşık olarak aynı itki üretilir. Tüm pervaneler yerçekimi kuvveti üzerinde net dönmeyen bir moment ürettiğinden, hexacofter sıfır rüzgâr (sürükleyici kuvvet) koşullarında dengesini korur.

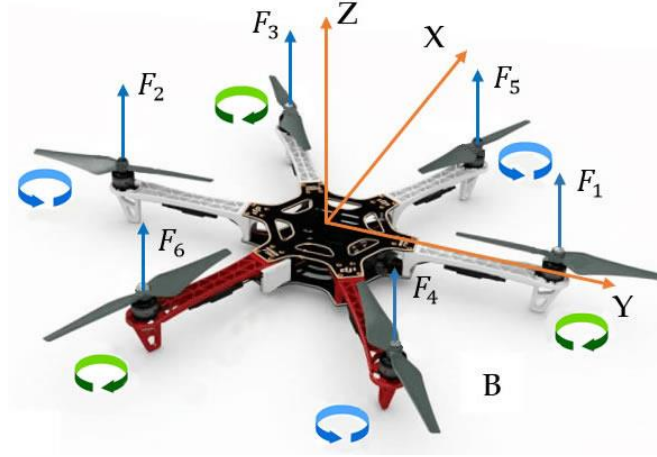
## 2.2 Hexacofter İnsansız Hava Aracının Matematiksel Modeli

Bu çalışmada tasarlanan Hexacofter, Şekil 2.2’de görüldüğü gibi X-konfigürasyonuna sahip ve ticari model adı F550 olarak geçen Frame (Gövde-Şase) üzerinde geliştirildi.



Şekil 2.2: F550 Hexacofter Tasarımı

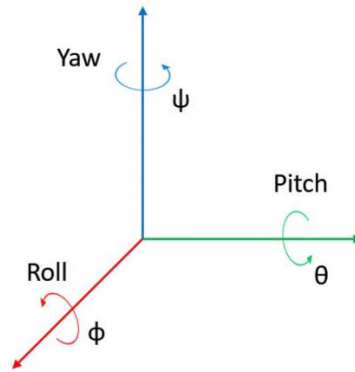
Hexacofter’in konumu, kendi koordinat sisteminin yer-merkezli koordinat sistemine göre nasıl yönlendiğini ifade eder. Bu, Hexacofter’in kendi x, y ve z eksenini etrafında nasıl döndüğünü temsil eder (Şekil 2.3). Bu durumda, sağ el kuralı kullanılarak, Hexacofter’in üç temel hareketi olan roll, pitch ve yaw elde edilir.



**Şekil 2.3:** F550 Hexacopter'in x,y ve z Eksenleri

**Roll (Sağa ve sola hareket):** Hexacopter'in, x ekseninde dönüşünü ifade eder. Hexacopter'in x ekseninde dönmeye hareketi, sağa veya sola yatmasını temsil eder. Bunun gerçekleştirilmesi için 1, 4 ve 5 numaralı motorların hızının artırılması/azaltılması ve aynı anda 2, 3 ve 6 numaralı motorların hızının artırılması/azaltılması gerekir. Bu manevra sırasında x ekseninde dönme momenti oluşturulur ve böylece bir açısal ivme meydana gelir. Roll hareketinin dönüş açısı  $\phi$  ile gösterilir ve rad/s cinsinden ölçülür (Şekil 2.4).

**Pitch (İleri veya geri hareket):** Hexacopter'in y ekseninde dönüşünü ifade eder. Hexacopter'in y ekseninde dönmeye hareketi, öne veya arkaya eğilmesini temsil eder. Bunun gerçekleştirilmesi için 3 ve 5 numaralı motorların hızının artırılması/azaltılması ve aynı anda 4 ve 6 numaralı motorların hızının artırılması/azaltılması gerekir. Her ne kadar y ekseninde, 1 ve 2 numaralı motorların konumuna denk geldiği görülüyor olsa da, bu motorlar pitch hareketini etkilemez. Pitch açısı  $\theta$  ile gösterilir ve rad/s cinsinden ölçülür (Şekil 2.4).



**Şekil 2.4:** F550 Hexacopter'in Roll, Pitch ve Yaw Eksenleri

**Kaynak:** (Perez, A., et al, 2019)

**Yaw (z-ekseni etrafındaki dönme hareketi):** Hexacopter'in z eksenini etrafındaki dönüşünü ifade eder. Hexacopter'in z-ekseni etrafındaki dönme hareketi saat yönünde veya saat yönünün tersine dönmesini temsil eder. Bu hareket sırasında her motor, dönüş sırasında z eksenini etrafında dönme momenti oluşturur. Bu nedenle, bu moment motorun dönüş yönünün tersi yönüne yönlendirilir. Eğer motor saat yönünde dönüyorsa, z eksenini etrafında bir trigonometrik dönüş oluşturur. Bu dönme hareketi, 1, 3 ve 6 nolu motorların hızını azaltarak/artırarak ve aynı anda 2, 4 ve 5 nolu motorların hızını artırarak/azaltarak gerçekleştirilir. Dönme hareketinin açısı  $\psi$  ile gösterilir ve rad/s cinsinden ölçülür (Şekil 2.2).

Yer merkezli koordinat sistemi (Earth-centered inertial-ECI coordinate system), sabit bir referans sistemidir. Hexacopter'in izleyeceği rotayı tanımlarken, Hexacopter'in hareketini sabit bir referansa göre ifade etmek için bu koordinat sistemi kullanılır. Hexacopter'in çerçevesiyle ilişkilendirilen koordinat sistemi, Hexacopter'e monte edilen uçuş kontrol kartı ve gps modülü ile aynı yönde hizalanır (Şekil 2.3).

Bu düzenlemeye göre, x eksenini ileri yönde, y eksenini sağa yönlendirilir ve z eksenini yukarıya doğru, x ve y eksenleri tarafından belirlenen düzleme dik bir şekilde yönlendirilir.



**Şekil 2.5:** Hexacopter Uçuş Kontrol Kartı ve GPS Modülü Yönü

Hexacopter'in çerçevesi ile yerleşik referans sistemi arasındaki dönme hareketlerini ifade etmek için en yaygın dönüşüm yöntemlerinden biri olan dönüş matrisi kullanılır. Hexacopter'in üç eksen etrafında döndüğü durumda, dönüş matrisi (Rotation matrix–Rot), her bir dönme eksenini için bir tane olmak üzere üç dönüş matrisinin ( $R(\phi)$ ,  $R(\theta)$ ,  $R(\psi)$ ) bir serisinden oluşur. Bu yöntem, Hexacopter'in hareketlerini doğru bir şekilde ifade etmek ve kontrol etmek için kullanılır. Yaw (z-axis) eksenindeki dönüş matrisi  $R(\psi)$ , Denklem (2.1)'deki gibi yazılmaktadır [65].

Yaw rotation matrix (z-axis):

$$R(\psi) = \begin{vmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{vmatrix} \quad (2.1)$$

Pitch (y-axis) eksenindeki dönüş matrisi  $R(\theta)$ , Denklem (2.2)'deki gibi yazılmaktadır [65].

Pitch rotation matrix (y-axis):

$$R(\theta) = \begin{vmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{vmatrix} \quad (2.2)$$

Roll (x-axis) eksenindeki dönüş matrisi  $R(\phi)$ , Denklem (2.3)'teki gibi yazılmaktadır [65].

Roll rotation matrix (x-axis):

$$R(\phi) = \begin{vmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{vmatrix} \quad (2.3)$$

Yukarıda gösterilen sırayla üç dönüş matrisi ile Hexacopter'in yer merkezli çerçeve dönüş matrisi olan Denklem (2.4) elde edilir [65]:

$$\text{Rot} = R(\phi)R(\theta)R(\psi) =$$

$$\begin{vmatrix} \cos\psi\cos\theta & \cos\psi\sin\theta\sin\phi - \sin\psi\cos\phi & \cos\psi\sin\theta\cos\phi + \sin\psi\sin\phi \\ \sin\psi\cos\theta & \sin\psi\sin\theta\sin\phi + \cos\psi\cos\phi & \sin\psi\sin\theta\cos\phi - \cos\psi\sin\phi \\ -\sin\theta & \cos\theta\sin\phi & \cos\theta\cos\phi \end{vmatrix} \quad (2.4)$$

Rot, bir ortogonal matristir, yani ters matrisi kendi transpozuna eşittir [65]:

$$(Rot)^{-1} = (Rot)^T \quad (2.5)$$

Altı motorlu Hexacopter, simetrik yapıya sahip rijit bir katı olarak kabul edilir ve ağırlık merkezi Hexacopter'in merkezinde bulunur. İHA üzerine etki eden iki ana kuvvet vardır. Bunlar, yerçekimi kuvveti ( $g$ ) ve pervanelerin dönmesiyle oluşan itki kuvvetidir. Aynı zamanda, Hexacopter üzerinde olumsuz bir yönde etki eden ve ileri veya yukarı hareketi engellemeye çalışan sürtünme kuvveti veya atmosferik hava ile sürtünme de bulunur. Yerçekimi kuvvetinin, her zaman  $z$  ekseninde aşağı yönde bir yöne sahip olacağı [13] ve Denklem (2.6)'daki [65] gibi ifade edilebileceği söylenebilir:

$$F_g = Rot \begin{vmatrix} 0 \\ 0 \\ -mg \end{vmatrix} = \begin{vmatrix} mgsin\theta \\ -mgsin\theta cos\phi \\ -mgcos\theta cos\phi \end{vmatrix} \quad (2.6)$$

### 2.3 Hexacopter İnsansız Hava Aracında Kullanılan Donanımlar

İHA tasarımında kullanılan donanımlar; elektrik, elektronik ve mekanik donanımlar başlıkları altında detaylandırılmaktadır.

#### 2.3.1 Elektriksel donanımlar

Bu bölümde İHA'nın elektriksel donanımları detaylandırılmaktadır.

##### 2.3.1.1 Dış rotorlu (outrunner) fırçasız DC motor

Dış rotorlu fırçasız DC motorlar (Brushless DC Motor – BLDC Motor), yüksek güç yoğunluğuna, yüksek güvenilirliğe, üstün verimliliğe, düşük gürültü seviyelerine ve basit bir tasarıma sahip oldukları için geniş bir uygulama alanına sahiptirler [66]. Bu özellikleri sayesinde, dış rotorlu fırçasız DC motorlar birçok endüstriyel sektörde, insansız hava araçlarında, askeri ekipmanlarda, otomotivde ve ev gereçlerinde yaygın olarak kullanılmaktadır [67, 68]. İnsansız hava araçlarında kullanılan tipik bir dış rotorlu BLDC motoru Şekil 2.6'da gösterilmektedir.



**Şekil 2.6:** Tipik bir Dış Rotorlu BLDC Motoru

**Kaynak:** (sunnyskyusa.com, 2023)

Dış rotorlu fırçasız DC motor, stator ve rotor adı verilen iki ana bileşenden oluşmaktadır. Dış rotorlu fırçasız DC motorların rotorları, genellikle mıknatısların bulunduğu dış kabuk (rotor) veya kısım olarak tasarlanır [70]. Motorun statoru ile ilişkilendirilerek, rotorun kutup sayısı belirlenir. Kutup sayısı arttıkça motorun üretebileceği moment seviyesi de artar. Ancak, yüksek moment genellikle motorun dönüş hızının düşmesine yol açar [71].

İnsansız hava araçlarında kullanılan dış rotorlu fırçasız DC motorları (Brushless DC-BLDC motor), temel olarak rotor ve stator arasındaki manyetik alanların etkileşimiyle çalışır. Özellikle, bu prensibi daha iyi anlamak için dış rotorlu fırçasız DC motorlarına iyice odaklanmak gerekir. Dış rotorlu BLDC motorlarda, rotorun manyetik alanı genellikle kalıcı mıknatıslardan gelir. Bu kalıcı mıknatıslar, rotorun iç yapısına entegre edilmiştir ve dönerken bir manyetik alan oluşturur. Stator ise genellikle akım taşıyan sarılmış bakır tellerden oluşur. Statorun bu sarımları, dış bir kaynaktan sağlanan elektrik akımı ile manyetik bir alan oluşturur.

Rotor ve stator arasındaki bu manyetik alanlar etkileşime girer ve rotorun manyetik alanı, statorun manyetik alanıyla etkileşime geçer. Bu etkileşim sonucunda, rotor manyetik alanı stator manyetik alanı tarafından itilir veya çekilir, bu da rotorun dönmesine neden olur. Bu dönüş hareketi, elektrik enerjisini mekanik enerjiye dönüştürür ve böylelikle insansız hava araçlarının hareketini sağlar. Özetle, rotor ve stator arasındaki manyetik alan etkileşimi, dış rotorlu BLDC elektrik motorlarının temel çalışma prensibini oluşturur.

Dış rotolu fırçasız DC motor üzerindeki harf ve rakam içeren ibareler, motor hakkında kısmi bilgi içermektedir. Örneğin, bu çalışmadaki İHA'nın tasarımında kullanılan SunnySky markasına ait X2216-10 880Rpm/Volt dış rotorlu fırçasız DC

motoru ele alırsak (Şekil 2.7) ; X2216 ibaresindeki 22 sayısı stator çapının 22mm olduğunu, 16 sayısı ise stator yüksekliğinin 16mm olduğunu belirtmektedir. Motor üzerindeki 880 değeri ise motorun yüksüz halinde (pervane yokken) 1 volt için 1 dakikada attığı devir sayısıdır (rpm). Örneğin, 880Rpm/Volt değerine sahip bir motora, yüksüz durumda 3.7 volt verildiğinde, motorun bir dakikada  $880 \times 3.7 = 3256$  devirle döneceği anlamına gelir.

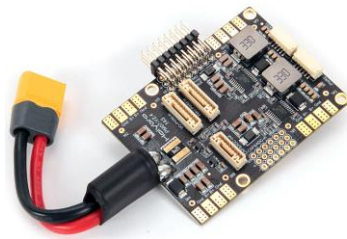


**Şekil 2.7:** SunnySky X2216 880Rpm/Volt Dış Rotorlu Fırçasız DC Motor

**Kaynak:** (sunnyskyusa.com, 2023)

### 2.3.1.2 Güç dağıtım kartı

Güç yönetim kartı (Power management board) (Şekil 2.8), güç dağıtım kartının yanı sıra güç modülü amacına da hizmet eder. Pixhawk 4 uçuş kontrol kartı ve ESC'lere güç sağlamanın yanı sıra, motorlara sağlanan pil voltajı ve akımı hakkında uçuş kontrol kartına bilgi gönderir.



**Şekil 2.8:** Holybro Pixhawk 4 Güç Modülü (PM07)

**Kaynak:** (holybro.com, 2023)

### 2.3.1.3 Lipo pil

Lityum polimer (Lipo) bataryalar, insansız hava araçlarında yaygın olarak kullanılmaktadır. Lipo pillerde voltaj, sahip oldukları hücre sayısına göre belirlenir. Her lipo hücresi 3.7V (nominal voltaj) olarak derecelendirilmiştir. Şekil 2.9'da görüldüğü üzere 4S ( 4 cells connected in Series) bir Lipo pil,  $4 \times 3.7 = 14.8V$ 'dur.

Lipo piller, dış rotorlu fırçasız DC motorlar için yeterli miktarda akım üretebilir. Lipo piller, ayrıca “C” derecelerine göre de derecelendirilir. Lipo pilin C derecesi, pilin güvenli bir şekilde ne kadar hızlı deşarj edilebileceğini ifade eder. Daha yüksek C derecesine sahip bir pil daha fazla enerji sağlar ve bu da daha yüksek performans anlamına gelir. Lipo bataryalarda, pilin maksimum kapasitenin %80’i kullanılmalıdır, bu oran aşıldığında pilin ömrü kısalmaktadır [73].



**Şekil 2.9:** Leopard Power 7000 mAh 40C 14.8 V-4S Lipo Pil

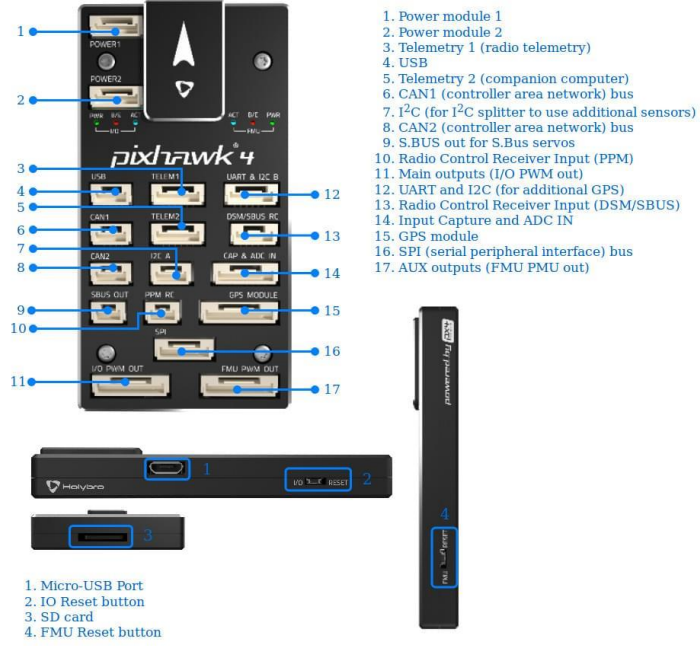
**Kaynak:** (www.fldepo.com, 2023)

### 2.3.2 Elektronik donanımlar

Bu bölümde İHA'nın elektronik donanımları detaylandırılmaktadır.

#### 2.3.2.1 Otopilot uçuş kartı

Otopilot uçuş kartı, İnsansız Hava Aracının kontrolünü sağlamak için kullanılan küçük bir otomatik pilot bilgisayarıdır [75] (Şekil 2.10). Otopilot uçuş kartı, İHA'nın uçuş parametrelerini hesaplamak, sensör verilerini almak, rotayı belirlemek ve hedeflenen konumları izlemek için bir dizi algoritma ve yazılım içerir. Otopilot uçuş kartı, belirlenmiş parametrelere göre İHA'nın otonom bir şekilde uçmasını sağlamak için tasarlanmıştır. Ayrıca, yer kontrol istasyonu tarafından belirlenen talimatları takip etmek üzere de programlanabilirler. Bu tip otopilot kartlar, uçuşun güvenli ve istikrarlı olmasını sağlamak için jiroskop, manyetometre, ivmeölçer ve daha birçok sensörle donatılmıştır. Adından anlaşılacağı üzere ivmeölçer sensörü, ivmeyi ölçmek için kullanılırken, jiroskop sensörü ise açılma hızı ölçmek için kullanılmaktadır. Manyometre sensörü ise yön bulma amaçlı kullanılmaktadır. Sensörler, İHA'nın otonom olarak hareket etmesini sağlayan temel bileşenlerdir.



**Şekil 2.10:** Holybro Pixhawk 4 Uçuş Kontrol Kartı

**Kaynak:** ([https://docs.px4.io/v1.12/en/flight\\_controller/pixhawk4.html](https://docs.px4.io/v1.12/en/flight_controller/pixhawk4.html), 2023)

### 2.3.2.2 Elektronik hız denetleyicisi

Elektronik hız denetleyicileri (Electronic Speed Controllers-ESC'ler), uçuş kontrol kartından Darbe Genişlik Modülasyonu (Pulse Width Modulation-PWM) sinyalini alarak motorların hızını kontrol etmektedir [73]. Şekil 2.11'de görüldüğü üzere ESC'nin verebileceği bir akım değeri vardır. Tercih edilen motora uygun ESC seçilmelidir. Yani, kullanılacak ESC'nin, motorun çalışma sırasında ulaşabileceği en yüksek akım ihtiyacını karşılaması gerekir. Örneğin, 50A akım çekebilen bir motor için 40A akım verebilen ESC tercih edilemez.



**Şekil 2.11:** Racerstar RS40A V2 40A BLHELI\_S BB2 OPTO 2-5S ESC

**Kaynak:** ([rcmumbai.com](http://rcmumbai.com), 2023)

### 2.3.2.3 Radyo alıcı ve verici

Radyo verici, İHA'na kontrol sinyallerini göndermek için kullanılırken, alıcı ise İHA'nın bu sinyalleri alıp işlemesini sağlar. Bu cihazlar, İHA'ların uzaktan kumanda edilmesi için önemli bir role sahiptir. İHA için verici ve alıcı seçilirken

çalışma menzili dışında dikkat edilmesi gereken diğer bir nokta ise kanal sayısıdır. Şekil 2.12’de görülen Radiolink AT10 model alıcı ve verici, 12 kanala sahiptir. Daha fazla kanal sayısı, İHA’nın farklı operasyonel görevleri için kullanıcı tarafından daha fazla komut atanmasına olanak sağlar. İHA için radyo alıcı ve verici seçimi yapılırken bu durum gözönünde bulundurulmalıdır.



**Şekil 2.12:** Radiolink AT10 RC Verici ve R12DS Alıcı

**Kaynak:** (www.team-ncrc.com, 2023)

### 2.3.2.4 Telemetri

Kablosuz iletişim teknolojisi, tarım, sağlık, mühendislik, askeri ve havacılık gibi çeşitli alanlarda yaygın olarak kullanılmaktadır. Genellikle, cihazların kontrolü ve veri izlemesi için kullanılan iletişim teknolojisi, bu alanlarda olduğu gibi İHA teknolojisinde de önemli bir role sahiptir. Kablosuz iletişimin gelişmelerinden biri, İHA’lar tarafından kullanılan telemetri sistemidir (Şekil 2.13).

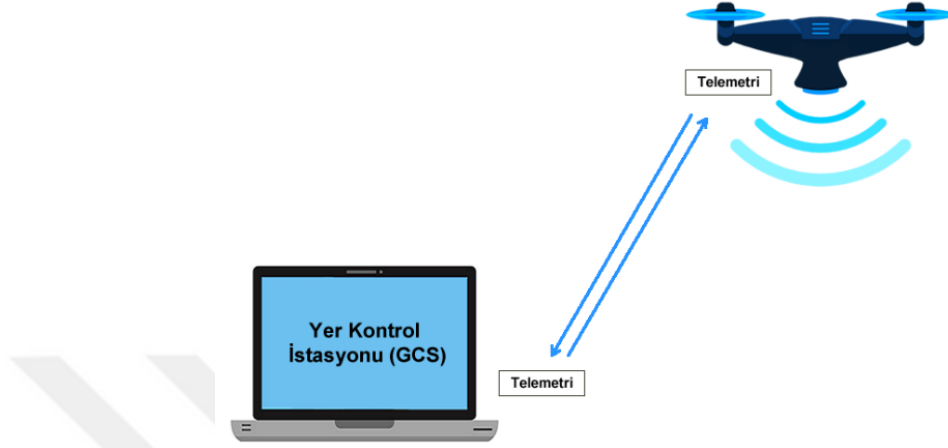


**Şekil 2.13:** Holybro Telemetri Radyo

**Kaynak:** (holybro.com, 2023)

Bu sistem, uzaktaki alanlara veri iletimini sağlamak için kullanılır ve özellikle zorlu veya ulaşılması güç koşullarda çalışmak için ideal bir çözümdür [80].

Telemetri, İHA ile yer kontrol istasyonu (Ground Control Station - GCS) arasında çift yönlü veri iletişimi için kullanılır. İHA'ya ait anlık uçuş verilerinin yer kontrol istasyonuna ve yer kontrol istasyonundan gönderilen görev komutlarının İHA'ya aktarılmasında rol oynar (Şekil 2.14).



**Şekil 2.14:** GCS ile İHA arasındaki Çift Yönlü Veri Akışı

### 2.3.2.5 GPS Modülü

GPS (Global Positioning System), dünya çapında konum belirleme ve zaman tespiti sağlama yeteneğine sahip bir uydu tabanlı navigasyon sistemidir [81]. GPS modülü, GPS sinyallerini almak ve konum bilgilerini belirlemek için kullanılır (Şekil 2.15). GPS modülü, İHA'lar, araç navigasyon sistemleri ve akıllı telefonlar gibi cihazlar için konum tespiti sağlar.



**Şekil 2.15:** Holybro Pixhawk 4 GPS Modül

**Kaynak:** (www.urhanshop.com, 2023)

### 2.3.2.6 Aksiyon kamera

İHA'larda görüntü elde etmek için hafif kameraların kullanılması gereklidir çünkü genellikle İHA'ların taşıma kapasitesi sınırlıdır. Bu nedenle, İHA sistemlerinde özellikle normal lens tiplerine sahip kameralar tercih edilir. Ancak,

aksiyon kameralarının mevcut olması (Şekil 2.16), İHA projelerinde yeni ve farklı bakış açıları sunmaktadır [83]. Gimbal ile birlikte kullanılıyor olabilmeleri çok geniş açıda görüntü almaya olanak sağladığından 4K aksiyon kameraları oldukça faydalı olmaktadır. FPV modundaki çekimleri sayesinde İHA'dan anlık görüntü alma imkânı sunmaktadır.



**Şekil 2.16:** Sjcam Sj6 Legend 4K Aksiyon Kamerası

**Kaynak:** (www.hepsiburada.com, 2023)

### 2.3.2.7 Gimbal

Gimbal, genellikle kameralar, sensörler veya diğer ekipmanların sabitlenmesi ve stabilizasyonu için kullanılan bir araçtır. Gimbal, üç eksenli bir sistem olarak tasarlanmıştır (Şekil 2.17) ve ekipmanın istenen yönlere dönmesine veya eğilmesine olanak tanırken, aynı zamanda istenmeyen titreşimleri ve hareketleri absorbe ederek sabitliği sağlar. Gimbal, insansız hava araçlarında sıklıkla kullanılmaktadır. Kameranın istenen bir nesneyi izlemesini veya istikrarlı bir şekilde görüntülemesini sağlar. Gimbal, pan (yatay), tilt (dikey) ve roll (yatay ekseninde dönme) hareketleri sağlayarak çok yönlü bir kullanım sağlar.



**Şekil 2.17:** Tarot 3DIII Metal Gimbal (TL3T01)

**Kaynak:** (/robu.in, 2023)

### 2.3.2.8 Pil göstergesi

Pil göstergesi, 1'den 8 hücreye kadar Li-Po, HV-LiPo, Li-Ion veya Li-Fe tipi pillerin voltajını ve dengesini izlemek için kullanılan bir cihazdır (Şekil 2.18). Örneğin, 3 hücreli bir pilin hem toplam voltaj değerini hem de her bir hücrenin voltaj değerini ayrı ayrı gösterebilir. Kullanıcı tarafından belirli bir düşük voltaj eşiği belirlenirse, gösterge düşük gerilim eşiğine ulaştığında, bunu bildirmek için yüksek sesli bir alarm sesi çıkarmaktadır. İHA, herhangi bir sebepten ötürü düştüğünde, şayet ormanlık veya engebeli bir araziye düşmüşse, pil seviyesi azalınca sesli uyarı vereceğinden İHA'yı bulmak kolaylaşabilir.



**Şekil 2.18:** Lipo Pil Göstergesi

**Kaynak:** (www.amazon.com, 2023)

### 2.3.2.9 Video verici

İHA kamerasından (bağlı bir kamera ünitesinden) yer kontrol istasyonuna video/ses göndermek için kullanılan bir vericidir (Şekil 2.19). RHPC (Right Hand Circular Polarization) antenlerinin, yaklaşık 4 km ye kadar video gönderebilme menzili vardır.



**Şekil 2.19:** Video Verici

**Kaynak:** (/abra-electronics.com, 2023)

### 2.3.3 Mekanik donanımlar

Bu bölümde İHA'nın mekanik donanımları detaylandırılmaktadır.

### 2.3.3.1 Gvde

Gvde (frame - Őasi), ok rotorlu İnsanız Hava Aralarının iskeletidir (Őekil 2.20). Bir İHA'nın kurulumu genellikle motor-pervane montajı, gvde, faydalı yk, uuŐ kontrol cihazı, ESC'ler, pil ve sensrler dâhil olmak zere mekanik ve elektronik bileŐenleri ierir. Motor-pervane montajı, pil tarafından saęlanan gc kullanarak gereken itme kuvvetini retir. Bu itme kuvveti, İHA Őasisi zerinde gerilmeler oluŐturur, nk İHA'nın tm aęırlıęını Őasi taŐımaktadır. İHA, havada hareket ettięinde eŐitli kuvvetlere maruz kalır ve bu kuvvetlerin iinde srkleme ve kaldırma kuvvetleri baskındır. Daha byk bir İHA Őasisi iin, İHA'nın kollarında farklı kesitler kullanılır. Bu kesitler, İHA Őasisi zerinde hem aerodinamik hem de yapısal bir etkiye sahiptir. İHA gvdesi, İHA'nın tm aęırlıęını taŐıma sorumluluęuna sahiptir ve aynı zamanda gvdenin aerodinamik olarak verimli olması olduka nemlidir [88].



Őekil 2.20: F550 Gvde

Kaynak: (www.motorobot.com, 2023)

### 2.3.3.2 İniŐ takımı

İHA'nın duruŐ pozisyonunu geniŐleterek daha istikrarlı iniŐ ve kalkıŐ yapmasına olanak saęlar. Ayrıca İHA'nın gvdesi altında yer alan Lipo batarya, gimbal ve kamera gibi donanımların iniŐ sırasında veya olası dŐŐlerde zarar grmesini engeller. Őekil 2.21'de grlen iniŐ takımı, F550 gvde ile uyumlu bir iniŐ takımıdır.



**Şekil 2.21:** F550 İniş Takımı

**Kaynak:** (www.motorobot.com, 2023)

### 2.3.3.3 Pervane

Pervaneler, İHA'ların kalkması için itme kuvveti üretir. Çaplarına ve eğim açılarına bağlı olarak sınıflandırılırlar ve çap ile eğim açısının çarpımı şeklinde temsil edilirler. Örneğin, Şekil 2.22'deki 1045'lik (10x4,5) bir pervane ele alırsa, "10" değeri inç cinsinden pervanenin uzunluğunu yani pervanenin oluşturduğu sanal dairenin çapını ifade ederken, "4,5" değeri inç cinsinden pervanenin tek bir dönüşü başına hareket miktarını ifade etmektedir. Motor momentine karşı koymak için hexacopter, 3 saat yönünde (CW) ve 3 saat yönünün tersine (CCW) dönen toplam 6 adet pervaneye ihtiyaç duyar [73].



**Şekil 2.22:** 1045 CW ve CCW Pervane Çifti

**Kaynak:** (www.dronmarket.com, 2023)

## 2.4 İHA Uçuş Süresi Hesaplamaları

### 2.4.1 İHA Tahmini gövde ağırlığı

İHA platformunu oluşturan tüm parçaların ortalama ağırlıkları, internet ortamından elde edilerek Çizelge 2.1'deki gibi İHA'nın ortalama toplam ağırlığı gram cinsinden hesaplandı.

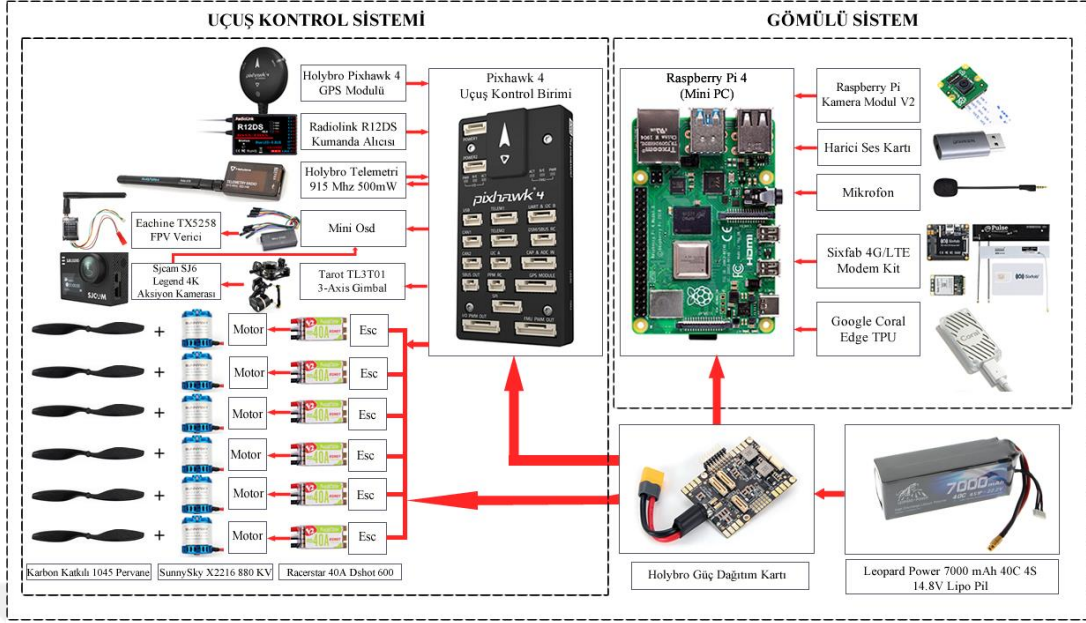
**Çizelge 2.1: İHA Bileşenleri Ortalama Ağırlığı ve Toplam Platform Ağırlığı**

No.	Parça	Ort. Ağırlık (gr)	Top. ağırlık (gr)
1	Frame	620 gr	3180 gr
2	İniş takımı	230 gr	
3	Güç dağıtım kartı	45 gr	
4	6 x Motor	405 gr	
5	6 x ESC	60gr	
6	6 x Pervane	80 gr	
7	Lipo Pil	735 gr	
8	Uçuş kartı	35 gr	
9	Raspberry Pi 4	50 gr	
10	Sixfab Modem Kit	120 gr	
11	Coral Edge TPU	80 gr	
12	Mini OSD	30 gr	
13	FPV Verici	15 gr	
14	1 x ESC (Raspi)	10 gr	
15	Telemetri	20 gr	
16	Gimbal	370 gr	
17	Aksiyon Kamera	85 gr	
18	Radyo Verici	15 gr	
19	Titreşim Önleyici	20 gr	
20	GPS Modül	40 gr	
21	Raspi Kamera Modül	15 gr	
22	Mikrofon	20 gr	
23	Diğer (Vida, Kablo vb.)	80 gr	

#### **2.4.2 İHA İtki sistemi ve azami uçuş süresi hesaplamaları**

İHA'nın görev gereksinimleri doğrultusunda Şekil 2.23'deki gibi uçuş kontrol sistemini ve gömülü sistemi oluşturan donanımlar seçildi. Bir önceki bölümde detaylandırılan donanımlar ve ilerleyen bölümlerde ele alınacak gömülü sistem donanımları dâhil, İHA platformunu oluşturan tüm montaj malzemeleri de hesaplama katılarak Çizelge 2.1'de görüldüğü üzere İHA'nın tahmini toplam ağırlığı 3180 gr olarak hesaplandı. İHA'nın başarılı bir şekilde uçabilmesi için itki sistemi, İHA'nın kendi ağırlığından daha fazla kuvvet üretmelidir. Bu itki kuvveti, İHA üzerinde bulunan 6 adet motor-pervane ikilisi tarafından sağlanmaktadır.

## İNSANSIZ HAVA ARACI PLATFORMU



**Şekil 2.23:** İHA Platformunu Oluşturan Uçuş Kontrol Sistemi ve Gömülü Sistem Donanımları

Tahmini ağırlığı 3180 gr olarak hesaplanan İHA'nın havada askıda kalabilmesi için her bir motor-pervane ikilisinin yaklaşık olarak 530 gram itki kuvveti (Thrust) üretmesi gerekmektedir. Sonuç olarak, toplam itkinin İHA'nın ağırlığına oranı en az 1 olmalıdır. Bu orana "TWR" (Thrust to Weight Ratio - İtme / Ağırlık Oranı) denir ve İHA'nın güvenli bir şekilde uçabilmesi için dikkate alınması gereken önemli bir parametredir [58]. TWR'nin en az 1 olması, İHA'nın yerçekimi etkisiyle düşmemesi ve istenilen uçuş performansını sağlaması için gereklidir. TWR (Thrust to Weight Ratio) oranı, bir İHA'nın agresif manevralara ne kadar uygun olduğunu, hızını ve uçuş süresini etkileyen bir göstergedir. TWR değeri, döner kanatlı İHA'lar için genellikle minimum 2 olarak hedeflenir. Çünkü, hem uçuş daha istikrarlı olur hem de yüksek manevra kabiliyeti elde edilir. TWR değerinin 2 olarak hedeflenmesi, Hexacopter'deki (6 rotorlu) her bir motor-pervane ikilisinin yaklaşık olarak 0,33 TWR üretebilmesi gerektiği anlamına gelir.

TWR değerinin hesaplanabilmesi için, İHA'nın itki sisteminin üretebileceği maksimum itki kuvveti bilinmelidir. Bu bilgi, kullanılan motorların üreticisi tarafından yayınlanan thrust (gram-force-gf) tablosu ile elde edilir. Thrust tablosu Şekil 2.24'deki gibi belirli bir motorun belirli batarya ve pervane kombinasyonlarıyla ne kadar itki kuvveti üretebileceğini tanımlar.

Prop(inch)	Voltage(V)	Amps(A)	Thrust(gf)	Watts(W)	Efficiency(g/W)	RPM	Load temperature in 100% throttle
APC1047	11.1	0.7	100	7.77	12.87	2664	100°C 1min25s
		1.8	200	19.98	10.01	3654	
		3	300	33.3	9.01	4328	
		4.4	400	48.84	8.19	4933	
		6	500	66.6	7.51	5405	
		7.55	600	83.805	7.16	5881	
		9.6	700	106.56	6.57	6300	
		11.6	800	128.76	6.21	6683	
		13.75	900	152.625	5.90	6988	
		16.2	1000	179.82	5.56	7364	
	18.2	1105	202.02	5.47	7718		
	14.8	0.6	100	8.88	11.26	2664	
		1.3	200	19.24	10.40	3598	
		2.3	300	34.04	8.81	4350	
		3.4	400	50.32	7.95	4945	
		4.7	500	69.56	7.19	5501	
		6	600	88.8	6.76	5853	
		7.6	700	112.48	6.22	6255	
		9.05	800	133.94	5.97	6700	
		10.9	900	161.32	5.58	7026	
		12.8	1000	189.44	5.28	7403	
		16.95	1200	250.86	4.78	8003	
22.1		1400	327.08	4.28	8625		
27.2	1650	402.56	4.10	9247			

**Şekil 2.24:** SunnySky X2216 880Rpm/Volt Dış Rotorlu Fırçasız DC Motor Thrust Değerleri

**Kaynak:** (sunnyskyusa.com, 2023)

Şekil 2.24 incelendiğinde, İHA’da dört hücreli (4S-14.8V) bir batarya ve 1047 pervane ile bir motorun üretebileceği maksimum itki kuvvetinin 1650 gram olduğu ve bu itki kuvvetini üretirken 27,2A akım çektiği görülmektedir. İHA’da kullanılan 1045 pervane iken firma tarafından açıklanan dokümanda 1047 pervane ile yapılan test verileri yer almaktadır. 1047’lik pervane çok yakın bir değer olduğu için hesaplamada büyük sapmalara neden olmayacağı düşünüldü ve bu veriler kullanıldı. İHA’da 6 adet motor yer aldığından ve bir motorun, 4S pil ve 1047 pervane ile maksimum 1650 gram itki kuvveti üreteceği varsayılarak İHA’nın itki sisteminin maksimum 9900 gram itki kuvveti üretebileceği hesap edildi.

Denklem (2.7) [92] kullanılarak altı adet motor tarafından üretilen toplam itki kuvveti, İHA’nın toplam ağırlığına bölünmesiyle TWR değeri; 3,11 olarak hesaplandı.

$$TWR = \frac{T}{W} = \frac{9900}{3180} = 3,11 \quad (2.7)$$

Bu değerin 2'den büyük olması önemlidir çünkü 2'nin altındaki değerlerde, İHA'ların manevra kabiliyetini önemli ölçüde kaybettiği uçuş deneylerinde görülmektedir. İHA'nın tahmini uçuş süresini hesaplamak için İHA üzerinde bulunan donanımların güç tüketimi ve bataryanın kapasitesi hesaplanmalıdır. Tahmini uçuş süresi, İHA'nın kendisini havada tutabilmesini sağlayacak durumdaki itme kuvveti dikkate alınarak hesaplanır. Yukarıda hesaplandığı üzere her bir motorun 530 gram itki kuvveti üretmesi durumunda İHA havada askıda kalabilecektir. Motora ait veri dokümanı incelendiğinde; bir motorun 4,7A akım ile 500 gram ve 6A akım ile 600 gram itki kuvveti ürettiği görülmektedir. 530 gram itki kuvveti için her bir motorun yaklaşık olarak 5,1A akım çekeceği tahmini yapılabilir.

İHA üzerinde motorlar dışında akım çeken başka donanımlar da bulunmaktadır. Bu donanımlardan oldukça düşük akım çekenlerin uçuş süresine sağlayacağı negatif katkı, hesaplanan tahmini süre (+ veya -) içerisinde göz ardı edilebilir. Pixhawk 4 uçuş kontrol kartı yaklaşık 5V 2A güç gereksinime ihtiyaç duymaktadır. Bunun yanı sıra, Raspberry Pi 4 Model B ve Sixfab 4G/LTE modem kit için ihtiyaç duyulan değerler her biri için 5V 3A'dir.

Coral Edge TPU hızlandırıcının ortalama 5V 0,5A güce ihtiyacı vardır. Ayrıca Tarot TL3T01 gimbal 12V 0,7A güç ile çalışmaktadır. Sonuç olarak, İHA'nın havada askıda kalabilmesi için 6 motor ( $6 \times 5.1A = 30.6A$ ) ve diğer donanımların bataryadan yaklaşık olarak anlık 40A akım talep edeceği öngörülmektedir.

Uçuş süresinin hesaplanmasında askıdaki uçuş durumu (hover uçuş modu) göz önüne alınırken, bataryanın uyumluluğu için maksimum değer hesaplaması da göz önüne alınır. İHA itki sistemi maksimum güçte çalışırken her bir motor 27,2A akım çekmektedir. Motorların, İHA askı konumunda iken talep edeceği maksimum akım değeri yeniden hesaplandığında ve diğer donanımların çekeceği akım da işin içine katıldığında bataryadan maksimum 172A akım talep edileceği öngörülmektedir.

İki durum için de talep edilecek akım değerleri hesaplandıktan sonra kullanılan Lipo batarya değerlerinin incelenmesi gerekmektedir. Kullanılan batarya 7000 mAh ve 40 C değerindedir. Burada C değerinin birimi  $h^{-1}$  türündedir ve Lipo pilin deşarj kapasitesini ifade eder. C kapasite değeri ile akım değerinin çarpılması durumunda, bataryanın anlık olarak verebileceği maksimum akım değeri Denklem (2.8) [93] ile hesaplanabilmektedir.

$$\text{Maksimum Anlık Akım} = Ah \times C = 7Ah \times 40C = 280A, C=h^{-1} \quad (2.8)$$

Hesaplama neticesinde İHA üzerinde kullanılan bataryanın anlık olarak maksimum 280A değerinde akım verebileceği anlaşılmaktadır. Bu durumda, İHA'nın talep edebileceği maksimum akım değerinden (172A) yüksek olduğu için bataryanın bu koşullar altında güvenli kullanıma uygun olduğu sonucuna varılır.

Uçuş süresinin hesaplanması için bataryadan alınabilecek toplam enerji, batarya kapasitesiyle nominal voltajın çarpılması sonucunda bulunur. Elde edilen sonuç Wh cinsinden batarya kapasitesini belirtir. Bir Lityum Polimer bataryanın tüm kapasitesi kullanıldığında batarya hücrelerinin yapısı tekrar kullanılmayacak şekilde bozulur. Bundan dolayı, acil durumlar dışında, bataryaların tüm kapasitesi kullanılmaz. Genellikle, bu değer %80 ile sınırlandırılır. Ancak her iki durum için İHA'nın kaç dakika uçuş yapabileceğini bilmek faydalıdır. Aşağıda bataryanın tüm kapasitesiyle kullanıldığı duruma “tüm enerji” ve %80 kapasiteyle kullanıldığı duruma “sürdürülebilir enerji” adı verilerek hesaplamalar yapıldı.

Bataryanın pil kapasitesi, mAh cinsinden verildiğinden;

$$\text{Elektriksel Güç(P)} = \text{Akım(I)} \times \text{Voltaj (V)} \quad (2.9)$$

ve

$$\text{Elektrik Enerjisi(E, Wh)} = \text{Elektrik Gücü(P)} \times \text{Zaman(t)} \quad (2.10)$$

Denklem (2.9)[93] ve Denklem (2.10)[93] kullanılarak,

$$\text{Tüm Enerji} = Ah \times V = 7Ah \times 14,8V = 103,6 Wh$$

$$\text{Sürdürülebilir Enerji} = Ah \times V = \frac{7Ah \times 14,8V \times 80}{100} = 82,88 Wh$$

olarak hesaplanır.

Bataryanın tüm enerjisi hesaplandığında 103,6Wh değerine ve %80 kapasite ile kullanılması durumunda ise enerjisinin 82,88Wh değerine sahip olduğu görülmektedir. Yukarıda yapılan hesaplamalarda İHA'nın askıda kalabilmesi için 14,8V gerilimde 40A akıma ihtiyaç duyacağı hesaplandı. Bataryanın çalışma gerilimi ile İHA'nın ihtiyaç duyduğu toplam akım değeri çarpıldığında platformun 592W'lık bir güce gereksinimi olduğu ortaya çıkmaktadır. Uçuş süresini hesaplamak için bataryanın enerji değeri ihtiyaç duyulan güç değerine bölünebilir. Hesaplanan

değerler 1 saatlik değerler olduğu için uçuş sürelerini hesaplariken dakikaya çevirmek için 60 katsayısı ile çarpıldı.

$$\text{Tüm Uçuş Süresi (dakika)} = \frac{Wh}{W} \times 60 = \frac{103,6Wh}{592W} \times 60 = 10,5 \text{ dk} \quad (2.11)$$

$$\text{Sürdürülebilir Uçuş (dakika)} = \frac{Wh}{W} \times 60 \times \frac{80}{100} = \frac{103,6Wh}{592W} \times 60 \times \frac{80}{100} = 8,4 \text{ dk} \quad (2.12)$$

Yapılan hesaplama sonuçları; insansız hava aracının, batarya kapasitesinin %80'ini kullanması durumunda 8,4 dakika, %100'ünü kullanması durumunda 10,5 dakika askıda kalabileceğini göstermektedir.

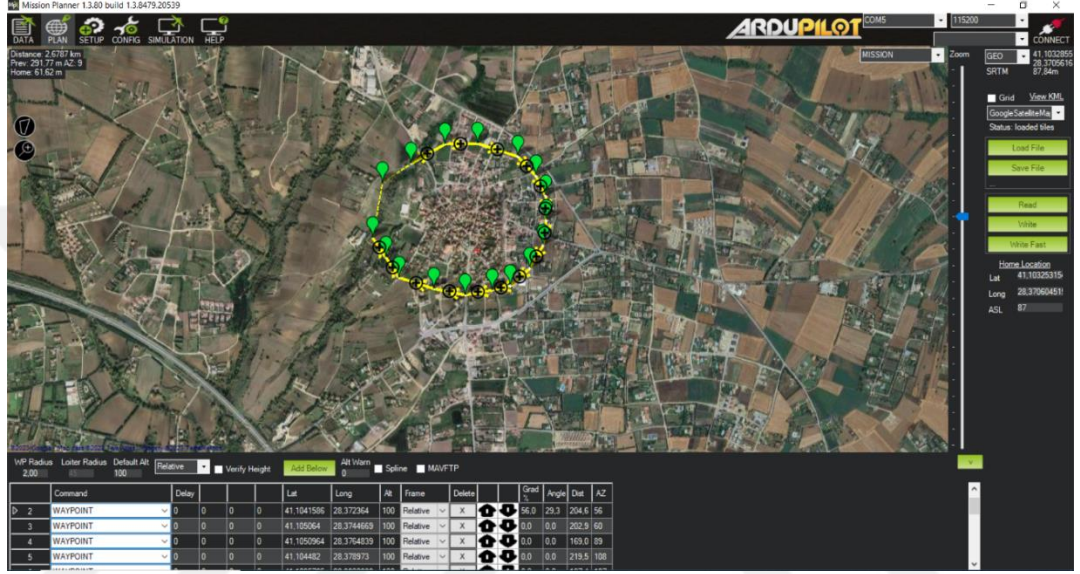
## 2.5 Aviyonik Sistem

Bu çalışmada, insansız hava aracı olarak Hexacopter sıfırdan geliştirildi ve İHA'nın aviyonik sistemi amaca uygun olarak tasarlandı. Pixhawk 4 otopilot uçuş kartı, aviyonik sistemin temelini oluşturmaktadır. Yapay zekâ destekli gömülü sisteme sahip İHA, nesne tespit ve sınıflandırma görevlerini yürütebilir. Gömülü sistem donanımları tarafından yakalanan görüntü ve ses verileri, Raspberry Pi üzerinde çalışan derin öğrenme algoritmaları ile işlenmektedir. Ayrıca, uçuş telemetri verileri, yer istasyonuna iletilmektedir. İHA üzerinde bulunan 4G/LTE modem kit sayesinde iletişimdeki mesafe sorununa kalıcı çözüm sağlandığı gibi, diğer taraftan İHA'nın uçuş yönetim bilgisayarına, Raspberry Pi'ye uzaktan komut gönderebilme yeteneği de kazandırıldı. İHA, istenilen önemli bilgileri ve yakalanan görüntüleri, anlık rapor olarak mail gönderebilmektedir. Hexacopter'in sahip olduğu aviyonik sistem tasarımı ile yer kontrol istasyonuna anlık görüntü gönderebilir ve veri transferi gibi farklı görevler gerçekleştirebilir. İHA, otonom uçuş yeteneğine sahiptir.

### 2.5.1 Planlı uçuş kontrolü

İnsansız Hava Aracı, bir İHA pilotu tarafından uzaktan kontrol edilerek uçurulabildiği gibi, yer kontrol istasyonu aracılığıyla belirlenen waypointler (ara noktalar) üzerinden planlı bir şekilde otomatik uçuş gerçekleştirebilir. Waypoint'ler, bir İHA'nın belirli bir noktadan belirlenen son pozisyona ulaşmak için geçmesi gereken ara konumları temsil eder. Tabi ki ufak sapmalar olabileceğini de dikkate almak gereklidir. Şekil 2.25'de yer kontrol istasyonu (GCS) yazılımında waypointleri belirlenmiş bir uçuş planı görülmektedir. Belirlenen uçuş planı, GCS üzerinden

kablolu veya kablosuz telemetri aracılığıyla İHA uçuş kontrol kartına aktarılır. Otonom uçuş özelliği olmayan İHA'lar yerden bir operatör tarafından yönlendirilirken, otonom uçuş yapabilen İHA'lar insan yardımı olmadan görevlerini gerçekleştirirler. Ayrıca planlı uçuşlarda da İHA'lar veri toplayabilirler ve bu verileri yer kontrol istasyonuna iletebilirler. Görev planlama ve komuta kontrol operasyonları, İHA'lardan alınan verileri saklayan ve merkezi olarak yöneten GCS'de gerçekleştirilir.



**Şekil 2.25:** Mission Planer GCS Planlı Uçuş Modu (Auto Mode) – Waypoints

### 2.5.2 Otomatik kalkış ve iniş

Otomatik kalkış ve iniş yeteneklerine sahip İHA'lar; genellikle, askeri, keşif ve gözetleme, tarım, haritalama ve diğer birçok uygulama alanında kullanılmaktadır. Otomatik kalkış ve iniş, İHA'nın belirli bir görevi gerçekleştirmek için kendisini hedef bölgeye getirmesini ve ardından güvenli bir şekilde kalkış ve iniş yapmasını sağlar. Bu özellikler, İHA'ların operatör müdahalesi olmadan daha karmaşık görevleri gerçekleştirebilmesine olanak tanır. Bu özellik, planlı uçuş kontrolü sırasında belirlenebilmektedir.

Otomatik kalkış ve iniş özellikleri, İHA'ların daha karmaşık görevleri gerçekleştirmesine ve operasyonel verimliliğini artırmasına yardımcı olur. Ancak, bu sistemlerin güvenlik ve hassasiyeti önemli olduğundan, tasarım ve uygulama süreçleri dikkatlice ele alınmalıdır.

### **2.5.3 Acil durumda eve dönüş modu**

İHA'nın operasyonel bir görevi yürütmesi sırasında acil durum veya durumlar meydana gelebilir ve İHA'nın kalkış yaptığı noktaya geri dönmesi istenebilir. Bu, İHA'nın Return To Launch (Başlatma Moduna Dönüş) modudur. Bu özellik kullanılırken İHA'nın başlangıç konumunun, o andaki doğru GPS kalkış konumu olup olmadığına dikkat edilmelidir. Aksi halde başka bir konuma iniş yapılabilir. Bu sebeple, daha önceki uçuşlarda bu özellik kullanılmış ise GPS konumunun yeniden belirlenmesi gerekir. Hangi koşullara bağlı olarak başlangıç noktasına döneceği ve başlangıç konumu, yer kontrol istasyon yazılımı ile belirlenmektedir. Ayrıca başlangıç konumu, İHA'nın kalkışı esnasında uzaktan kumandanın belirli komutlarıyla da ayarlanabilmektedir. Bataryanın belirli bir seviye altına inmesi durumu veya İHA'nın bir engel ile karşılaşması durumu, acil durum modunun koşulları olarak belirlenebilir. Böylesi bir durumda, İHA'nın otomatik geri dönmesi sağlanabilir. Bulunduğu konumdan başlangıç konumuna dönerken dönüş yüksekliği (rakım bilgisi), başlangıç noktasına döneceği yatay hızı gibi birçok parametre uçuş öncesinde görev planlayıcısı yazılımı (Mission Planer) ile ayarlanabilir.

### **2.5.4 Eş zamanlı görüntü ve telemetri verileri aktarımı**

İHA, aksiyon kamera ve video verici sayesinde yer istasyonuna anlık görüntü aktarımı sağlamaktadır. Ayrıca, radyo modülü ile telemetri verileri de yine anlık olarak yer kontrol istasyonuna gönderilmektedir.

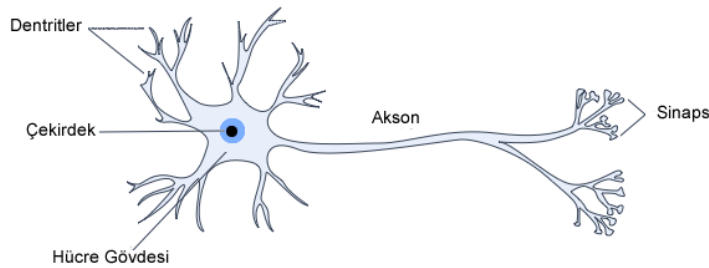
### 3. METODOLOJİ

İHA ile nesne tespit ve sınıflandırma uygulamaları, daha çok ticari olarak satılan hazır insansız hava araçları üzerinde gerekli modifikasyonlar sağlanarak yapılmaktadır [94, 95]. Bu sistemlerde donanım ve yazılım müdahalesi kısıtlıdır. Bu sebeple, bu çalışmada kullanılan İHA platformu açık kaynak kodlu yazılımlara sahip donanımlar ile sıfırdan tasarlanmıştır. Bu bölümde, İHA platformunun gömülü yapay zekâ modülünün geliştirilmesi sırasında kullanılan yapay zekâ teknolojileri, derin öğrenme tabanlı bilgisayarlı görü ve ses sınıflandırma teknolojileri ile nesne algılama ve sınıflandırma yöntemleri, gömülü sistem tasarımı ve otonomi için geliştirilen yazılım mimarisi detaylandırılmaktadır.

#### 3.1 Yapay Sinir Ağı

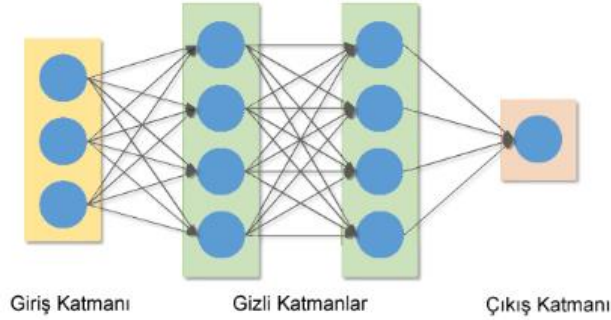
Yapay zekâ, insan zihninin doğasında olan görevleri yerine getirme yeteneğine sahip olduğu için son yıllarda oldukça popüler hale gelmiştir. Günümüzde makine zekâsı olarak da adlandırılan yapay zekâ, sinir ağları kullanılarak uygulanmaktadır [96]. Sinir ağları, farklı görevleri yerine getirmek üzere eğitilebilen özel yazılım modelleridir. Görüntülerin sınıflandırılması, konuşma tanıma, metinlerin çevrilmesi ve otonom araçların kontrolü gibi karmaşık görevler için kullanılırlar. Çeşitli sinir ağı türleri vardır, ancak görüntü algılama ve bilgisayarla görme için en yaygın kullanılanları evrimsel sinir ağlarıdır [97].

Sinir ağı, insan sinir sisteminden esinlenen ve tıpkı biyolojik nöronlar gibi bilgi işleme yeteneğine sahip bir yapay zekâ sistemidir [98] (Şekil 3.1).



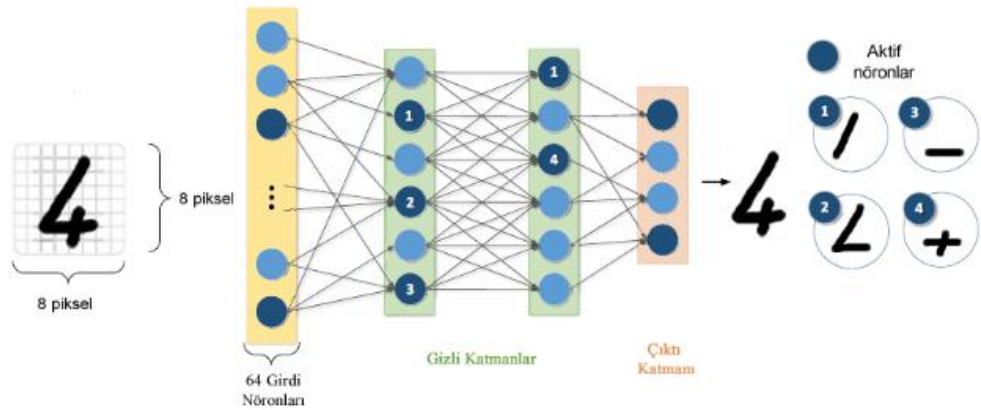
Şekil 3.1: Biyolojik Nöron Modeli

Optimizasyon, paralel hesaplama, matris cebiri ve sinyal işleme gibi çeşitli problem alanlarında başarıyla uygulanmaktadır. Sinir ağı, öncelikle nöronlara ve bağlantı linklerine ayrılır. Nöronlar, sinaps adı verilen bağlantı linkleri ile birbirine bağlanan zekâ işleme birimleridir. Geleneksel bir sinir ağı, Şekil 3.2’de gösterildiği gibi giriş katmanı, gizli katmanlar ve birbirine bağlı nöronların çıkış katmanından oluşurlar [97]. Bir ağın giriş ve çıkış katmanı arasında ne kadar çok gizli katmanı varsa, sinir ağı o kadar derindir.



**Şekil 3.2:** İki Gizli Katmanı Olan Bir Sinir Ağı Örneği

Sinir ağı modeli ile gerçekleştirilen bir görüntü algılama prensibi Şekil 3.3’te gösterilmektedir. Bu örnek, sinir ağının sekiz piksele sekiz piksel (8x8) çözünürlüğe sahip belirli bir görüntüden bir rakamı nasıl tanıyabileceğini ortaya koymaktadır. Tüm görüntü pikselleri, giriş nöronları olarak modele aktarılır. Bağlantılara atanan ağırlıklara (weights) bağlı olarak, farklı gizli katman setleri etkinleştirilebilir. Etkinleştirilen nöronun tekliği çıktıyı tanımlar. Bu tür karmaşık görevleri yerine getirmeden önce her bir sinir ağının eğitilmesi gerekir. Eğitim, kayıp fonksiyonunun minimize edildiği nöronlar arasındaki bağlantıların ağırlıklarını bulma süreci olarak tanımlanabilir.



**Şekil 3.3:** Sinir Ağı Modeli İle Gerçekleştirilen Görüntü Algılama Prensibi

Sinir ağı modelinde giriş nöronları eğitim için kullanılacak verileri temsil eder. Yukarıdaki örnekte olduğu gibi, girdi bir görüntü ise, girdi nöronları her bir piksel için değerleri temsil edebilir. Orta katmanlarda gizlenmiş nöronlar genellikle matematiksel hesaplamaları gerçekleştirir. Nöronlar arasındaki bağlantılar, girdi değerinin önemini belirleyen ağırlıklarla (weights) parametrelendirilir [97]. Giriş nöronları, girdiyi alır ve bağlantılar aracılığıyla işlenmek üzere gizli katmana gönderir. İşlenen bu hücreler daha sonra işlenmek üzere diğer gizli katmanlara (varsa), aksi takdirde çıktı katmanına gönderilir. Girdi sayısal bir değer, dize veya bir görüntü olabilir [99].

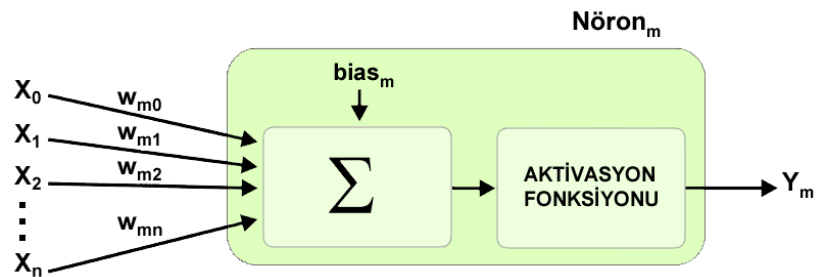
Temel bir yapay nöron, dört farklı öğeden oluşur (Şekil 3.4):

**Girdiler:** Girdiler, sensörlerden, özelliklerden veya diğer nöronlardan gelebilen sayısal değerlerdir.

**Ağırlıklar (Weights):** Her bir girdi için bir ağırlık bulunur. Bu ağırlıklar, nöronun hangi girdilerin önemli olduğuna ve hangilerinin önemli olmadığına karar vermesine yardımcı olmak için girdi değerini çarpar [100].

**Önyargılar (Biases):** Önyargılar, girdilerden bağımsız olarak nöronun çıktısını düzenlemek için kullanılır [100].

**Aktivasyon fonksiyonu:** Nöronun çıktısını üretmek için kullanılır. Aktivasyon fonksiyonları genellikle girdiler ile çıktılar arasındaki doğrusal olmayan ilişkileri öğrenmesini sağlamak için kullanılan non-linear fonksiyonlardır [101].



Şekil 3.4: Basit Yapay Nöron Modeli

Aktivasyon fonksiyonları genellikle doğrusal olmayan fonksiyonlardır [102]. Aktivasyon fonksiyonları, sinir ağlarının temel yapı taşlarından biridir çünkü nöronların çıktılarını düzenler ve genellikle ateşlenip ateşlenmemesi gerektiğine karar verir [103]. Bu fonksiyonlar, giriş sinyallerini alır ve onlara doğrusal olmayan bir dönüşüm uygular. Bu, sinir ağlarının basit doğrusal ilişkilerden ziyade daha

karmaşık ve esnek ilişkiler öğrenmesini sağlar. Yani, aktivasyon fonksiyonları, sinir ağlarının esneklik kazanmasına ve veri içindeki karmaşık desenleri ve özellikleri daha iyi anlamasına yardımcı olur. Bu doğrusal olmayan dönüşüm, sinir ağlarının daha geniş bir veri yelpazesini öğrenmesine ve modelin daha iyi performans göstermesine olanak tanır. Bazı yaygın aktivasyon fonksiyonları şunlardır [101]:

**Sigmoid fonksiyon:** Genellikle, sınıflandırma problemleri için yaygın olarak kullanılır.

**Hiperbolik tanjant fonksiyonu (tanh):** Sigmoid fonksiyonunun ölçeklendirilmiş bir versiyonu, ancak daha güçlü bir gradyana sahiptir.

**Rektifiye Edilmiş Doğrusal Birim fonksiyonu (ReLU):** En popüler aktivasyon fonksiyonlarından biri, varsayılan yaklaşım olarak kabul edilir. Diğer ikisine göre hesaplama açısından daha verimlidir.

### 3.2 Makine Öğrenmesi

Makine öğrenmesi, örüntü tanıma teorisinden ve bilgisayarların programlama olmadan belirli görevleri yerine getirebileceği teorisinden doğmuştur [104]. Makine öğrenmesi, otomatik olarak bir analiz modeli oluşturabilen bir veri analizi yöntemidir [105]. Aynı zamanda, yazılım uygulamalarının net bir programlama olmadan sonuçları daha doğru bir şekilde tahmin etmesini sağlayan bir algoritmadır. Makine öğrenmesi algoritmaları, büyük ve karmaşık verileri analiz edebilen ve hızlı bir şekilde doğru sonuçlar sağlayabilen çok geniş bir aralıkta bile hızlı ve otomatik olarak modeller oluşturabilir [106].

### 3.3 Derin Öğrenme

Derin öğrenme, bilgisayarların tıpkı insanlar gibi örneklerden öğrenmesine olanak sağlayan ve makinelerin insan müdahalesi olmadan deneysel öğrenme yoluyla beceri kazanabileceği bir yapay zekâ yöntemidir. Derin öğrenme modellerinin çoğu sinir ağı mimarisini kullanmaktadır, bu nedenle derin öğrenme modelleri genellikle derin sinir ağları olarak adlandırılır [107]. Derin öğrenme, makine öğrenmesinin bir alt kümesi olmasına [108] rağmen, makine öğrenmesinin nispeten benzersiz bir şeklidir. Örnek olarak görüntü sınıflandırması ele alındığında, derin öğrenme işlemi sürecinde manuel müdahale gerekmez ve algoritma görüntüden

özellikleri otomatik olarak çıkarır. Geleneksel algoritmalarla karşılaştırıldığında, derin öğrenmenin avantajı, eğitim için mevcut veri miktarı arttıkça, algoritmanın sonuçlarının sürekli olarak iyileştirilebilmesidir [106].

Derin öğrenme, belirli görevleri yerine getiren algoritmalarından farklı olarak, öğrenme verilerine bağlı olarak kendini geliştirebilen bir makine öğrenimi yöntemine sahiptir. Derin öğrenme yöntemlerinin en önemli özelliği, özellik temsillerini otomatik olarak öğrenebilmesi ve böylece çok fazla zaman kaybını önleyebilmesidir. Geleneksel makine öğrenmesi, bir giriş ve bir çıkış katmanından oluşan, giriş ve çıkış katmanları arasında birden fazla gizli katman bulunmayan sıg ağlara dayanır. Derin öğrenme ise girdi ve çıktı katmanları da dâhil olmak üzere bir ağda üçten fazla katman bulunduğunda nitelendirilir. Dolayısıyla bir ağın giriş ve çıkış katmanı arasındaki gizli katman sayısı arttıkça ağ daha da derinleşmektedir. Genel olarak, iki veya daha fazla gizli katmana sahip herhangi bir sinir ağı, derin sinir ağı olarak adlandırılır. Günümüzde kullanılan bazı popüler derin sinir ağı algoritmaları, Evrimsel Sinir Ağı (Convolutional Neural Network-CNN), Tekrarlayan Sinir Ağı (Recurrent Neural Network-RNN), Kısıtlı Boltzmann Makinesi (Restricted Boltzmann Machine-RBM) ve Otomatik Kodlayıcı (Autoencoder) algoritmalarıdır [109].

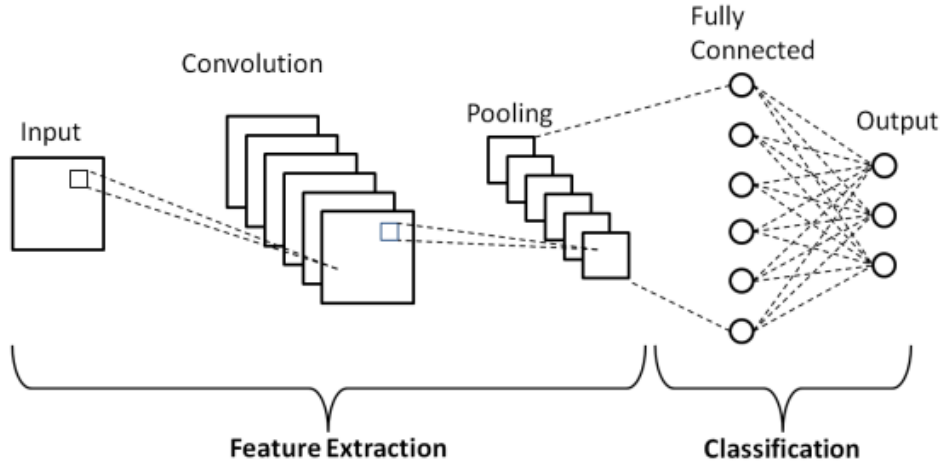
Bir derin öğrenme modelinin başarısı, modelin eğitiminden önce belirlenen ve eğitim sürecinde kullanılan hiperparametreler ile yakından ilişkilidir [110]. Modeli eğitmeden önce hiperparametrelerin belirlenmesi gerekir. Belirlenen bu hiperparametreler, modelin eğitim süreci boyunca değişmez. Derin öğrenme algoritmalarıyla çalışırken karşılaşılan en büyük zorluk, hiperparametre değerlerini ayarlamak ve kontrol etmektir [111]. Buna teknik olarak hiperparametre ayarlama veya hiperparametre optimizasyonu denir. Hiperparametreler derin öğrenme algoritmalarının birçok yönünü kontrol eder. İşte bunlardan bazıları:

- Algoritmayı çalıştırmanın süresine ve hesaplama maliyetine karar verebilirler.
- Sinir ağı modelinin yapısını tanımlayabilirler
- Modelin tahmin doğruluğunu ve genelleme yeteneğini etkilerler.

Başka bir deyişle hiperparametreler, sinir ağı modellerinin davranışını ve yapısını kontrol eder.

### 3.3.1 Evrişimsel sinir ağı

Evrişimsel sinir ağı (CNN), bilgisayarlı görü uygulamalarında kullanılan derin öğrenme ağıdır ve son yıllarda farklı araştırmacılar tarafından incelenmiştir [112]. Evrişimsel sinir ağı, görüntü sınıflandırma, görüntü segmentasyonu, nesne tanıma ve yüz tanıma ile ilgili görevler için kullanılan popüler bir mimaridir. Evrişimli bir sinir ağı, bir girdi katmanı (input layer) ve bir çıktı katmanı (output layer), girdi ve çıktı katmanı arasında birden fazla evrişim (convolution) ve havuzlama (pooling) katmanı içerir (Şekil 3.5). Evrişim katmanları, görüntüden özniteliklerin çıkarılmasını (features extraction) sağlar. Bu öznitelik çıktılarına, “öznitelik haritaları” denir [113]. Havuzlama katmanları, öznitelik haritalarını küçültür ve özelliklerin öğrenimini kolaylaştırır. Daha sonra tam bağlantılı katmanlar (fully connected layers), bu öznitelik haritalarını kullanarak nihai çıktı sınıfını üretir (classification).



Şekil 3.5: Temel bir Evrişimli Sinir Ağı (CNN) Mimarisi

Kaynak: (Phung ve diğ., 2018)

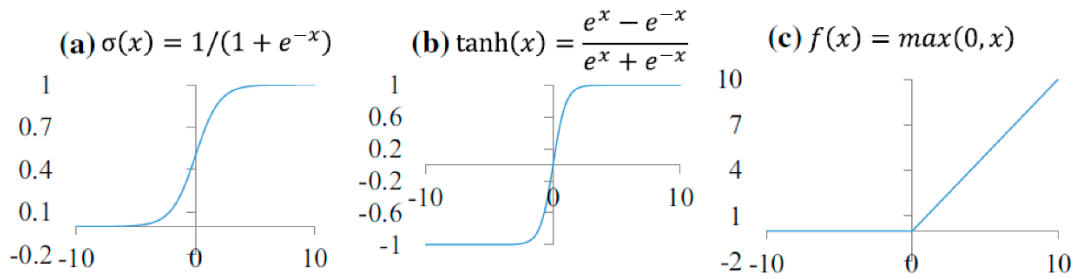
### 3.3.2 Hiperparametrelerin derin sinir ağı üzerindeki etkisi

Derin sinir ağları gibi sınıflandırma algoritmaları, kendileri tarafından oluşturulan modellerin tahmin performansını önemli ölçüde etkileyebilecek bir dizi hiperparametreye sahiptir [115]. Bir sinir ağının hiperparametreleri, eğitim sırasında ağın mimarisini ve davranışını belirleyen değişkenlerdir. Derin öğrenme algoritmalarını çalıştırmadan önce hiperparametrelerin ayarlanması gerekir. Bu ayarlama, katman sayısını, her katmandaki nöron sayısını, epok(epoch) sayısını, aktivasyon fonksiyonlarını, öğrenme hızını (learning rate), parti boyutunu (batch

size), optimize edici (optimizer) seçimini, kayıp fonksiyonunu (loss function) , sınıflandırma fonksiyonunu ve diğer hiperparametreleri belirlemeyi içerir. Hiperparametrelerin ayarlanması, sinir ağının performansını optimize etmek için çok önemlidir. Bu hiperparametreler, kullanıcı tarafından manuel yapılandırabilir veya en iyi tahmin performansı için bazı hiperparametre optimizasyon teknikleri kullanılabilir [116].

**Nöron Sayısı:** Ayarlanabilir ilk hiperparametre, her bir gizli katmandaki nöronların sayısıdır. Nöron sayısı, çözümün karmaşıklığına göre ayarlanmalıdır. Karmaşık bir seviyeye sahip olan görevlerde tahmin doğruluğunun yüksek olması için model daha fazla nörona ihtiyaç duyar [117].

**Aktivasyon Fonksiyonu (Activation Function):** Model eğitiminde önemli diğer bir hiperparametredir. Bir sinir ağında, giriş verileri (girdiler) giriş katmanına, ardından gizli katmanlara ve son çıkış katmanına işlenir. Bir katmandan diğer katmana geçen giriş değerleri, aktivasyon fonksiyonuna göre değişmeye devam eder. Aktivasyon fonksiyonu, bir katmanın giriş değerlerinin çıkış değerlerine nasıl hesaplanacağına karar verir [118]. Bir katmanın çıkış değerleri daha sonra tekrar giriş değerleri olarak bir sonraki katmana aktarılır. Yani, her katman önceki katmanın çıktı değerlerini girdi olarak alır ve kendi çıktı değerlerini hesapladıktan sonra, bir sonraki katmana aktarır. Bu, iteratif olarak devam eder. Katmanlarda yer alan ve girdinin nasıl hesaplanacağını belirleyen aktivasyon fonksiyonları, modelin öğrenme yeteneğini büyük ölçüde etkileyebilir. Derin sinir ağlarında kullanılan farklı aktivasyon fonksiyonları vardır (Şekil 3.6).



**Şekil 3.6:** Yaygın Olarak Kullanılan Üç Aktivasyon Fonksiyonu: (a) Sigmoid, (b) Tanh ve (c) ReLU

**Kaynak:** (Teng ve diğ., 2020)

Her aktivasyon fonksiyonunun, giriş değerlerini hesaplamak için kendi formülü (grafiki) vardır. Örneğin, ReLU (Rectified Linear Unit) ve Sigmoid fonksiyonları aynı model için farklı sonuçlar verebilir.

**Öğrenme Hızı (Öğrenme Oranı – Learning Rate):** Modelin optimize olmasını sağlayan hiperparametrelerden biri öğrenme hızıdır. Öğrenme hızı, her eğitim iterasyonunda modelin güncelleme adımlarının ne kadar büyük veya küçük olacağını belirler [120]. Başka bir deyişle, bir modelin minimum kayıp fonksiyonuna ulaşması için gereken adım boyutunu kontrol eder. Yüksek bir öğrenme hızı, hızlı bir öğrenme sağlayabilir, ancak aşırı uyum sorununa yol açabilir. Düşük bir öğrenme hızı, minimum kayıp fonksiyonunu bulma olasılığını artırır [121].

**Epok Sayısı (Number of Epoch):** Bir veri kümesinin tamamının sinir ağı modelinden geçme sayısına epoch (epok-dönem) denir [122]. Çok az sayıda dönem, sinir ağının öğrenme potansiyelini kullanamamasına yol açabilir ve yeterince öğrenmemesi nedeniyle eğitim yetersiz uyumla sonuçlanabilir. Fazla epok, modelin eğitim verilerini çok iyi tahmin edebildiği ancak yeni görüntü verilerini yeterince iyi tahmin edemediği aşırı uyumla sonuçlanacaktır. Optimum sonucu elde etmek için dönem sayısı ayarlanmalıdır.

**Parti Boyutu (Batch Size):** Bir defada sinir ağına iletilen örneklerin sayısıdır [123]. Eğitim verilerini küçük gruplar halinde işlemek için kullanılır. Batch size, bellek ve hesaplama verimliliği üzerinde etkili olabilir. Büyük parti boyutları daha hızlı eğitim süresi sağlayabilir, ancak daha fazla bellek gerektirir.

**Adım Sayısı (Number of Step):** Epok sayısı gibi “adım sayısı” da modeli eğitmeyle ilgili bir parametredir ve genellikle epok sayısı ile karıştırılan bir terimdir. Tipik bir derin öğrenme eğitim döngüsünde, eğitim veri kümesi küçük gruplar (batch size –parti boyutu) halinde işlenir. Her küçük grup, sinir ağında ileri ve geri yayılımı tamamlar. İleri ve geri yayılım, derin öğrenme modelinin eğitilmesinin temelini oluşturur. İleri yayılım, modelin tahminlerini hesaplar, geri yayılım ise bu tahminlerin gerçek değerlerle ne kadar uyumsuz olduğunu değerlendirir ve ağın parametrelerini bu uyumsuzluğu azaltmaya yönlendirir. Bir küçük grup işlendikten sonra, modelin parametreleri güncellenir.

Başka bir ifadeyle, bir epoch boyunca, modelin parametreleri (ağırlıklar) eğitim verilerine dayalı olarak kaybı veya hatayı azaltmak için güncellenir. Modelin eğitimi sırasında yapılan güncelleme sayısı, “adım sayısı” sayısı olarak ifade edilir.

Adım sayısı, eğitim veri kümesinin boyutuna, batch size ve epoch sayısına bağlı olarak hesaplanır. Hesaplama, Denklem (4.1) [124] ile şu şekilde yapılabilir:

$$\text{Her bir Epoch için Adım Sayısı (step number)} = \frac{\text{Eğitim Örnek Sayısı}}{\text{Batch Size}} \quad (3.1)$$

Epok, adım sayısı ve batch size arasındaki ilişkiyi gösteren bir örnek:

Elinizde modeli eğitmek için 1600 adet görsele sahip bir veri setiniz varsa ve mini batch size = 4 olarak belirlenirse;

Bu, 1 adımda 4 adet görselin eğitim için işleneceği ve işlendikten sonra model parametrelerinin güncelleneceği anlamına gelir. Tüm veri setinin işlenmesi,  $1600/4 = 400$  adımda tamamlanacaktır. Model tüm veri setini işlediğinde bir Epoch'u tamamladığını söyleyebiliriz. Modelin eğitiminden önce ayarlanan epok sayısı 10 olarak belirlenmiş ise  $10 \times 400 = 4000$  adımda tüm eğitim tamamlanmış olur.

Özetlemek gerekirse, bir epoch, eğitim verileri üzerinde tam bir geçişi temsil eder ve adım sayısı, her epoch içindeki güncellemeleri ifade eder, bu güncellemeler model parametrelerinin nasıl güncellendiğini gösterir. Eğitimdeki toplam adım sayısı, epoch sayısı, batch size ve eğitim veri kümesinin boyutuna bağlıdır.

**Optimizasyon Algoritmaları (Optimizer – Optimize Edici):** Optimizasyon algoritmaları, eğitim sırasında ağı kullandığı parametreleri en iyi şekilde optimize ederek modelin tahmin doğruluğunu arttırmaya yarayan algoritmalardır. Stochastic Gradient Descent (SGD), Momentum, Adam, RMSProp gibi optimizasyon algoritmaları, en çok tercih edilen algoritmalardır [125].

**Model Mimarisi:** Katman sayısı, katmanların derinliği, nöron sayısı ve katman türleri (evrişimli, tam bağlantılı, vs.) modelin kapasitesini ve yeteneklerini etkiler.

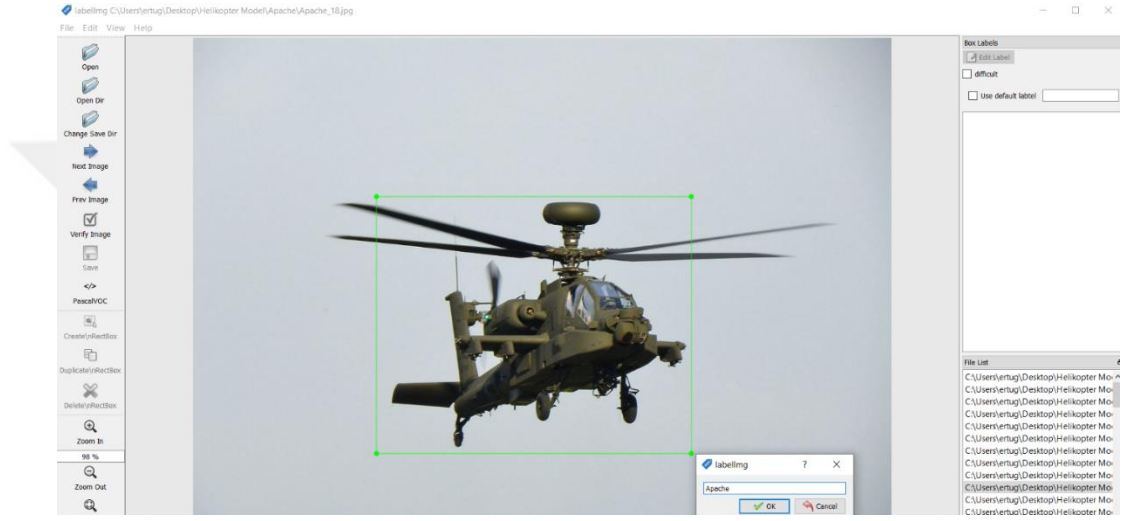
Bu hiperparametreler, modelin performansını büyük ölçüde etkileyebilir ve bu nedenle genellikle deneme yanılma yöntemiyle optimal değerleri bulmak gerekir.

### 3.4 Bilgisayarlı Görü

Günümüzde, İnsansız Hava Aracı platformlarında bilgisayarla görme teknikleri uygulanmaktadır [126-133]. Bilgisayarlı görü algoritmalarının geliştirilmesi ve bu algoritmaların hatalarının azaltılması, İHA üzerine monte edilen sensörler ve mikro denetleyiciler, İHA'ların karmaşık uygulamalarda kullanılabilmesine olanak sağlamıştır. İHA üzerinde yer alan kamera tarafından görüntüler ve videolar kayıt altına alındıktan sonra, bu ham veriler görüntü işleme

teknikleri ve algoritmaları ile analiz edilerek İHA'nın tutumu ve konumunu değiştirmek dâhil olmak üzere birçok aksiyon gerçekleştirilebilir [16].

Nesne algılama ve sınıflandırma işlevini uygulamak için öncelikle algılanacak nesneyi tanıyacak olan sinir ağını eğitmek gerekir. Derin sinir ağı mimarisine sahip algoritmaların eğitimi birkaç adımdan oluşmaktadır. Nesne algılama algoritmalarını kullanmak için öncelikle girdi verilerini tanımlamamız gerekir. Veri kümesi olarak da bilinen girdi verisi, tespit edilecek nesnenin bulunduğu, işaretlendiği ve etiketlendiği bir dizi görüntüdür (Şekil 3.7).



**Şekil 3.7:** AH-64 Apache Helikopterinin Etiketlenmesi

Veri kümesi eğitim ve test için olmak üzere iki kısma ayrılmalıdır. Bu, aşırı uyum olarak adlandırılan durumdan kaçınmak için temel gerekliliktir. Eğitimden sonra sinir ağı modeli test edilir. Şayet, bu testler eğitim verileriyle yapılırsa model çok iyi performans gösterir. Ancak, yeni görüntülerde o kadar da iyi performans gösteremez. Bu nedenle, sinir ağını eğitirken aynı verileri hem eğitim hem de test için tekrar kullanmak iyi bir fikir değildir, çünkü modeli değerlendirmek için modelin eğitim için kullanılmayan test verileri üzerinde nasıl çalışacağını bilmek gerekir. Veri kümesinin iyi bir şekilde ayrılması, görüntülerin yaklaşık %75'inin eğitim ve %25'inin test için kullanılması ile mümkün olabilir [97].

### 3.4.1 Görüntü işleme

Görüntü işleme, görüntülerin değiştirilmesi ve analiz edilmesi için kullanılan bir yöntemdir. Bu işlemde, girdi bir görüntüdür ve çıktı olarak işlenmiş bir görüntü elde edilir. Görüntü işleme, genellikle bir görüntüdeki özellikleri belirlemek veya piksellerin özelliklerini kullanmak suretiyle gerçekleştirilir [134]. Görüntü işlemenin

temel amaçlarından biri, görüntü kalitesini iyileştirmektir. Bu, görüntü iyileştirme olarak adlandırılır ve genellikle görüntülere özel özellikler eklenerek yapılır. Ayrıca, görüntülerde oluşan kalite kaybına neden olan gürültüyü kaldırarak görüntüleri onarmak da başka bir amaçtır.

Görüntü işleme, görüntülerin daha iyi anlaşılmasını sağlayan bir teknolojidir. Bu süreç, birçok farklı uygulama alanında görüntülerin analizinden iyileştirilmesine kadar birçok farklı amaç için kullanılabilir.

### 3.4.2 Görüntü ile nesne algılama ve sınıflandırma

Nesne algılama, verilen bir görüntü içindeki nesnenin sınıfını ve konumunu tanıyan bir bilgisayarla görme görevidir. Nesne algılama, fotometrik veya geometrik özellikleri eğitim veri tabanındaki hedef nesnenin özellikleriyle eşleşen parçaları saptamak için bir görüntünün her bir parçasını arayarak gerçekleştirilir. Bu, bir nesne şablonunun farklı konumlarda, ölçeklerde ve rotasyonlarda bir görüntü boyunca taranmasıyla gerçekleştirilebilir ve şablon ile görüntü arasındaki benzerlik yeterince yüksekse bir algılama bildirilir. Algılanan nesne, belirli bir sınıf tanımlayıcısına sahip bir dörtgen sınırlayıcı kutu ile açıklanır (Şekil 3.8).



**Şekil 3.8:** Algılanan Nesnenin Sınırlayıcı Dörtgen Kutu İle Tespit Edilmesi ve Sınıfı

Son çalışmalar, nesne tespitini gerçekleştirmek için derin öğrenmeyi kullanmıştır. Nesne tespiti için evrimsel sinir ağı, bu tür derin öğrenme yöntemlerinin temsilcisidir [135].

### 3.4.3 Nesne sınıflandırmada performans metrikleri

Nesne sınıflandırmada, derin öğrenme modelinin yapısı ve eğitim süreci modelin performansını doğrudan etkiler. Modelin sınıflandırmadaki tahmin doğruluğunu ölçmek için test veri kümesi kullanılır ve modelin performansını ortaya çıkaracak birkaç ölçüm değeri hesaplanır. Karışıklık matrisi (confusion matrix), doğruluk (accuracy), hassasiyet (precision), geri çağırma (recall), F1-skoru (F1-score) ve ortalama Ortalama hassasiyet (mean Average precision – mAp) metrikleri, sınıflandırma doğrulukları açısından performans değerlendirmesi sağlayan ve yaygın olarak kullanılan değerlendirme ölçekleridir [136-138].

#### 3.4.3.1 Karışıklık matrisi (Confusion matrix)

Karışıklık matrisi (Confusion matrix), bir nesne sınıflandırma modelinin performansını ölçmek için kullanılır. Sınıflandırma sonucunda elde edilen, doğru pozitifler (True Positives), doğru negatifler (True Negatives), yanlış pozitifler (False Positives) ve yanlış negatiflerin (False Negatives) sayısını içerir. İkili sınıflandırma problemleri için karışıklık matrisi, Şekil 3.9'daki gibi 2x2 matris şeklinde gösterilir.

		PREDICTIVE VALUES	
		POSITIVE	NEGATIVE
ACTUAL VALUES	POSITIVE	TP	TN
	NEGATIVE	FP	FN

**Şekil 3.9:** İkili Sınıflandırma Problemi İçin Karışıklık Matrisi

Karışıklık matrisleri, tahmin edilen ve gerçek değerlerden elde edilen sayıları temsil eder ve sınıflandırma problemindeki tahmin sonuçlarının bir özetini verir. Karışıklık matrisinde, tahmin edilen değerler Positive/Negative (Pozitif/Negatif), gerçek değerler ise True/False (Doğru/Yanlış) olarak tanımlanmaktadır.

**TP (True Positive):** Model pozitif bir tahminde bulunur ve bu gerçekte doğrudur.

**FP (False Positive):** Model negatif bir tahminde bulunur ama gerçekte doğrudur.





**FN (False Negative):** Model negatif bir tahminde bulunur ve bu gerçekte yanlıştır.

**TN (True Negative):** Model negatif bir tahminde bulunur ama bu gerçekte doğrudur.

Karışıklık matrisini, daha iyi anlaşılması için bir örnek üzerinden açıklayalım. Örneğimizdeki nesne sınıflandırma modelinin görevi, “kedi” ve “köpek” sınıfından oluşan test veri setinde yer alan her bir görsel için “kedi” veya “kedi değil” şeklinde tahminde bulunmasıdır. Böyle bir ikili sınıflandırma probleminin karışıklık matrisi Şekil 3.10’daki görünecektir.

**Pozitif tahmin:** “Sen bir kedisin”

**Negatif tahmin:** “Sen bir kedi değilsin”

		PREDICTIVE VALUES	
		POSITIVE (CAT)	NEGATIVE (DOG)
ACTUAL VALUES	POSITIVE (CAT)	<b>TRUE POSITIVE</b> You are a cat  <b>A</b>	<b>FALSE NEGATIVE</b> You are not a cat  <b>B</b> TYPE II ERROR
	NEGATIVE (DOG)	<b>FALSE POSITIVE</b> You are a cat  <b>C</b> TYPE I ERROR	<b>TRUE NEGATIVE</b> You are not a cat  <b>D</b>

**Şekil 3.10:** İkili Sınıflandırma Problemi İçin Örnek Karışıklık Matrisi

Bu örneğin; TP, TN, FP ve FN terimlerini belirleyelim:

TP (True Positive): Model, hayvanın kedi olduğunu tahmin eder (pozitif yönde tahmin) ve gerçekte de öyledir (doğru sonuç).

TN (True Negative): Model, hayvanın kedi olmadığını tahmin eder (negatif yönde tahmin) ve gerçekte de öyledir (doğru sonuç)(o bir köpek).

FP (False Positive)(Type 1 Error): Model, hayvanın kedi olduğunu tahmin eder (pozitif yönde tahmin) ama gerçekte öyle değildir (yanlış sonuç)(o bir köpek).

Pozitif bir sonucun tahmin edildiği ancak gerçekte yanlış bir sonucun olduğu durum, yanlış pozitif olarak adlandırılır. Bu senaryo, Tip 1 Hatası olarak bilinir. Kötü öngörüdeki bir iyilik gibi.

FN (False Negative)(Type 2 Error): Model, hayvanın kedi olmadığını tahmin eder (negatif yönde tahmin) ama gerçekte öyle değildir (yanlış sonuç)(o bir kedi).

Negatif bir sonucun tahmin edildiği ancak gerçekte doğru bir sonucun olduğu durum, yanlış negatif olarak adlandırılır. Bu senaryo, Tip 2 Hatası olarak bilinir.

#### 3.4.3.2 Doğruluk (Accuracy)

Doğruluk ölçütü, bir sınıflandırma modelinin doğru tahmin ettiği örneklerin, toplam örnek sayısına oranıdır. Yani, Denklem (3.2)'de gösterildiği gibi doğru sınıflandırılan örneklerin tüm örnekler içindeki oranını ifade eder [139].

$$\text{Accuracy} = \frac{\text{Number of Correctly Classified Samples}}{\text{Total Number of Samples}} = \frac{TP+TN}{TP+TN+FP+FN} \quad (3.2)$$

Doğruluk, nesne sınıflandırması için yaygın olarak kullanılan bir ölçüttür.

#### 3.4.3.3 Hassasiyet (Precision)

Hassasiyet ölçüğü, pozitif tahminlerin doğruluğunu ölçer. Bu ölçek, Denklem (3.3)'te gösterildiği gibi doğru pozitif tahminlerin, toplam doğru pozitif tahminler ve yanlış pozitif tahminlerin toplamına bölünmesiyle elde edilir [139].

$$\text{Precision} = \frac{TP}{TP+FP} \quad (3.3)$$

#### 3.4.3.4 Geri çağırma (Recall)

Geri çağırma metriği, toplam pozitif örneklerin içinde doğru pozitif örneklerin oranını ölçer. Denklem (3.4)'te gösterildiği gibi modelin gerçekte pozitif olanların ne kadarını doğru bir şekilde tespit ettiğini gösterir [139].

$$\text{Recall} = \frac{TP}{TP+FN} \quad (3.4)$$

#### 3.4.3.5 F1 Skoru (F1 Score)

İki veya daha fazla sınıflı sınıflandırma modellerinde bir değerlendirme ölçütü olan F1-skoru, hassasiyet ve geri çağırma ölçütlerini tek bir formülde birleştirerek model performansının daha iyi analiz edilmesini sağlar. F1-skoru, hassasiyet ve geri çağırmanın harmonik ortalaması olarak [140] Denklem (3.5) ile hesaplanır [139].

$$\text{F1 - score} = 2 * \frac{\text{Precision.Recall}}{\text{Precision+Recall}} \quad (3.5)$$

#### 3.4.3.6 Ortalama Hassasiyet (Average Precision-AP) ve Ortalama Ortalama Hassasiyet (Mean Average Precision-MAP)

Ortalama Hassasiyet (Average Precision-AP)

AP, hassasiyet-geri çağırma (precision-recall) eğrisi altındaki alanın ortalama değerini ifade eder [141]. AP, bu eğri altındaki alanı Denklem (3.6) ile hesaplayarak bir sınıfın doğruluğunu ölçer [140].

$$AP = \int_0^1 p(r)dr \quad (3.6)$$

Ortalama Ortalama Hassasiyet (Mean Average Precision-MAP)

Değerlendirme metriği mAP, her bir sınıfın ortalama hassasiyetinin (AP - Average Precision ) ortalamasıdır [142]. Kısacası, mAP, bir modelin sınıflandırma doğruluğunu değerlendirmek için kullanılır. mAP değeri ne kadar yüksekse, sınıflandırma dedektörün performansı o kadar iyi olur [143]. Ortalama Ortalama Hassasiyet (mAP) metriği, Denklem (3.7) kullanılarak hesaplanır [140].

$$mAP = \frac{\sum_{i=1}^N AP_i}{N} \quad (3.7)$$

### **3.4.4 Görüntü ile Nesne Algılama ve Sınıflandırmada Kullanılan Derin Öğrenme Yapısı ve Teknolojileri**

Bu bölümde, görüntü ile nesne sınıflandırma tekniğinde kullanılan derin sinir ağının yapısal içeriği detaylandırılmaktadır.

#### **3.4.4.1 MobileNet mimari yapısı**

MobileNet, mobil platformlarda verimliliği artırmak amacıyla tasarlanmış optimize edilmiş modellerin bir kümesidir [144]. MobileNet, sinir ağı alanına kazandırılmış çok önemli bir katkıdır, çünkü sinir ağının performansını optimize etme ihtiyacından doğar ve modelin diğer özellikleri yerine verimliliğini önceliklendirir. Bu verimliliği elde etmek için, model daha önce tanımlanan şemalara uygun bir şekilde tekrar parçalanır ve evrişim katmanlarına bazı önemli değişiklikler getirir. MobileNet yapısı, geleneksel evrişim katmanını "derinlik bazlı" bir evrişim ve "nokta bazlı" bir evrişim ile değiştirmeyi önerir. Derinlik bazlı evrişim ile giriş katmanının her kanalında yalnızca bir filtreleme işlemi gerçekleştirilir. Nokta bazlı evrişim, 1x1 evrişimdir.

#### **3.4.4.2 Tek atışlı çoklu kutu dedektörü**

Tek atışlı çoklu kutu dedektörü (Single Shot Multibox Detector-SSD), temsilci bölge önerme ağı olmayan ve sınır kutularını ve sınıfları doğrudan özellik haritalarından tek seferde tahmin eden bir mimaridir [145]. SSD ile görüntülerdeki nesnelere sadece bir derin sinir ağıyla tespit edilebilir. SSD, görüntünün özellik haritasına bağlı olarak tespit ettiği nesneyi sınırlayıcı kutu içine alır. Diğer tek aşamalı yöntemlerle karşılaştırıldığında SSD, daha küçük bir giriş görüntüsü boyutuyla bile çok daha iyi bir doğruluğa sahiptir [146].

#### **3.4.4.3 Coco veri seti**

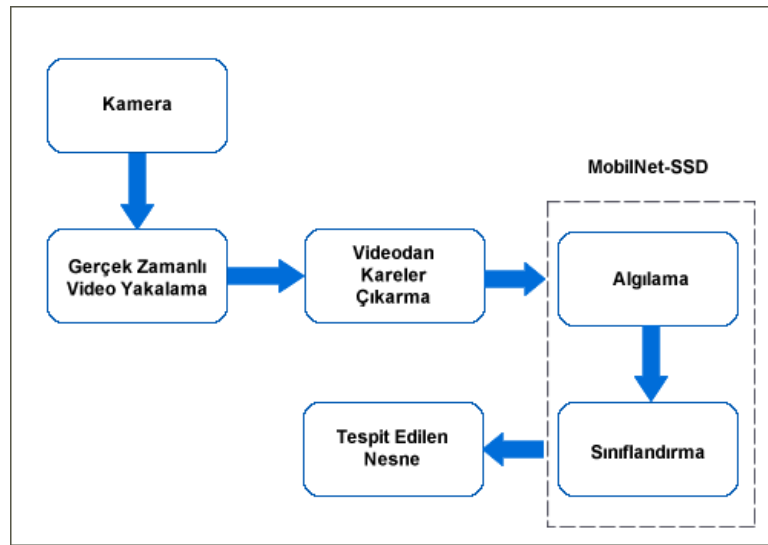
Microsoft Common Objects in COntext (MS COCO) veri kümesi, 82'si 5.000'den fazla etiketli örneğe sahip olmak üzere toplamda 91 ortak nesne kategorisi içerir ve veri kümesi toplamda 328.000 görüntüde 2.500.000 etiketli örneğe sahiptir. Popüler ImageNet veri kümesinin aksine, COCO daha az kategoriye ancak kategori başına daha fazla örneğe sahiptir. Kategori başına düşen örnek sayısı çok fazla olduğundan hassas 2D lokalizasyonu yapabilen ayrıntılı nesne modellerinin öğrenilmesine yardımcı olabilir. Veri kümesi ayrıca kategori başına düşen örnek

sayısı bakımından PASCAL VOC ve SUN veri kümelerinden önemli ölçüde daha büyüktür. Ek olarak, COCO veri seti ile diğerleri arasındaki kritik bir fark, bağlamsal bilgilerin öğrenilmesine yardımcı olabilecek görüntü başına etiketli örnek sayısıdır. MS COCO, ImageNet (3.0) ve PASCAL (2.3) ile karşılaştırıldığında görüntü başına çok daha fazla nesne örneği (7.7) içermektedir [147].

#### 3.4.4.4 SSD Tabanlı MobilNetV2 sinir ağı mimarisi ile nesne algılama

SSD MobileNetV2, tek atışlı çoklu kutu dedektörü (Single Shot Multibox Detector-SSD) [146] ve MobilNetV2 [148] derin öğrenme algoritmalarının birleşiminden oluşan derin öğrenme modelidir. SSD MobileNetV2, gerçek zamanlı olarak birden fazla nesneyi veya çoklu nesnelere tespit etmek için tasarlanmıştır ve MS COCO veri seti [147] üzerinde önceden eğitilmiş bir modeldir. Bu model, hafif özelliklere sahip mobil cihazlarda ve Raspberry Pi gibi mini bilgisayarlarda kullanılan en verimli ve doğru modeldir [149]. SSD MobileNetV2, TensorFlow ile geliştirilebilen ve optimize edilebilen bir derin sinir ağı mimarisidir.

SSD MobileNetV2, evrişimli sinir ağı (Convolutional Neural Network - CNN) mimarilerinden biridir. Evrişimli sinir ağı, nesnelere tanıma ve tespit etme amacıyla insan sinir ağlarını benimseyerek yapay sinir ağlarının geliştirilmesi için bilgisayar biliminde kullanılan bir yöntemdir [150]. SSD MobileNetV2, derinlik tabanlı evrişim ve noktasal evrişim kullanır [151]. Şekil 3.11’de, MobilNet-SSD ile nesne algılama ve sınıflandırma sürecinin aşamaları gösterilmektedir.



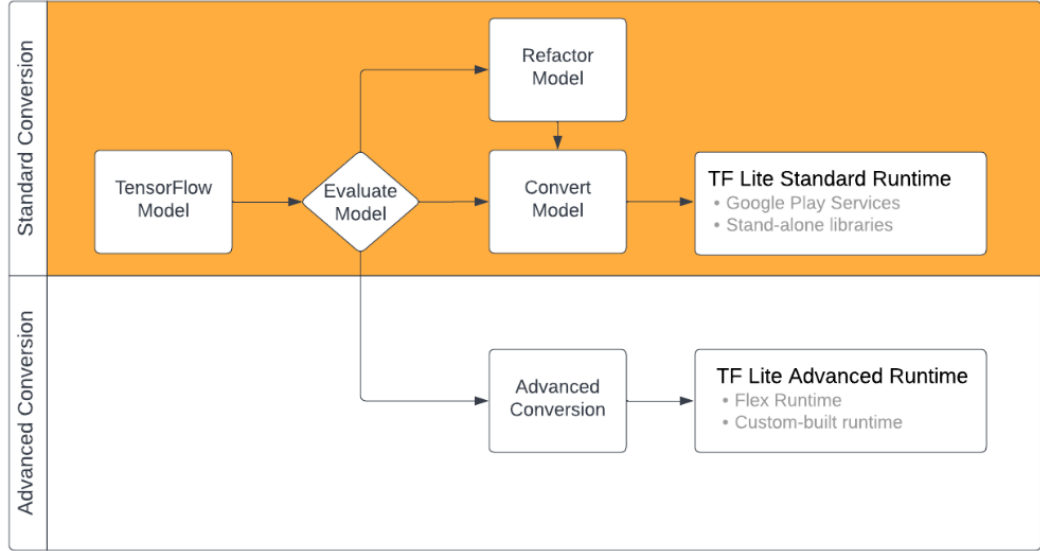
Şekil 3.11: MobilNet-SSD ile Nesne Algılama ve Sınıflandırma

### 3.4.4.5 TensorFlow ve tensorflow lite

TensorFlow, Google tarafından geliştirilmiş, makine öğrenmesi algoritmalarını ve derin öğrenme uygulamalarını yürütmek için kullanılan açık kaynaklı bir yazılım kütüphanesidir. Esnek mimarisi, geliştiricilerin birden fazla platformda hesaplamalar yapmasına olanak tanır. TensorFlow, yüksek hesaplama verimliliği, yüksek esneklik, güçlü taşınabilirlik, çoklu dil desteği ve optimize edilmiş performans özelliklerine sahiptir [106]. Şu anda TensorFlow, bilgisayarlarla görme, metin işleme, ses olayı sınıflandırma, konuşma tanıma, robotik ve görüntü tanıma gibi birçok makine öğrenmesi ve derin öğrenme alanında yaygın olarak kullanılmaktadır [152-159]. TensorFlow tabanlı sinir ağı, verileri temsil etmek için Tensör kullanır, hesaplama grafiği ile sinir ağını oluşturur, hesaplama grafiğini çizim ile yürütür, hat üzerindeki ağırlıkları (parametreleri) optimize eder ve modeli elde eder [160].

TensorFlow sadece Windows, Linux ve mac OS üzerinde değil, aynı zamanda iOS ve Android gibi mobil cihazlarda da çalışabilmektedir. TensorFlow kullanılarak ifade edilen bir hesaplama, telefonlar ve tabletler gibi mobil cihazlardan yüzlerce makineden ve GPU kartları gibi binlerce hesaplama cihazından oluşan büyük ölçekli dağıtılmış sistemlere kadar çok çeşitli heterojen sistemlerde çok az değişiklikle veya hiç değişiklik yapılmadan yürütülebilir [161].

TensorFlow çekirdeği ile oluşturulan bir model, daha küçük ve daha verimli bir makine öğrenmesi modeli olan TensorFlow Lite biçimine dönüştürülebilir [162]. TensorFlow Lite, Raspberry Pi mikroişlemcileri ve mobil cihazlar gibi kaynak kısıtlaması olan cihazlar için tasarlanmış açık kaynaklı bir derin öğrenme çerçevesidir. TensorFlow Lite, TensorFlow modellerinin mobil, gömülü ve IoT cihazlarında çalıştırılmasını mümkün kılar. Düşük gecikme süresi ve küçük ikili boyut ile cihaz üzerinde makine öğrenmesi çıkarımı sağlar. Bu nedenle, gecikme süresini iyileştirir ve güç tüketimi azalır. TensorFlow modeli, Şekil 3.12'deki iş akışına uygun olarak TensorFlow Lite model formatına dönüştürülür.



**Şekil 3.12:** TensorFlow Çekirdeği İle Oluşturulan Bir Modelin TensorFlow Lite Modeline Dönüştürülmesinin İş Akışı

**Kaynak:** (www.tensorflow.org, 2023)

Bir TensorFlow modeli, TensorFlow Lite dönüştürücü Python API (tf.lite.TFLiteConverter) ile TensorFlow Lite modeline dönüştürebilir. TensorFlow çekirdeği ile oluşturulan bir model, “SavedModel” formatı kullanılarak depolanır.

**tf.lite.TFLiteConverter.from\_saved\_model()**, SavedModel’i TensorFlow Lite dönüştürür. Aşağıdaki örnek kod (Şekil 3.13), SavedModel’in TensorFlow Lite modeline nasıl dönüştürüleceğini gösterir [162].

```

import tensorflow as tf

# Convert the model
converter = tf.lite.TFLiteConverter.from_saved_model(saved_model_dir) # path to the SavedModel directory
tflite_model = converter.convert()

# Save the model.
with open('model.tflite', 'wb') as f:
    f.write(tflite_model)

```

**Şekil 3.13:** TensorFlow Çekirdeği İle Oluşturulan Bir Modeli Tensorflow Lite Formatına Dönüştüren Python Kodu

**Kaynak:** (www.tensorflow.org, 2023)

### 3.4.5 SSD MobilNetV2 sinir ağının transfer öğrenme yöntemi ile yeniden eğitilmesi ve özelleştirilmesi

Bir nesne sınıflandırma işlemi için kullanılan sinir ağı modelinin başarısı, genellikle model parametreleri ve hiperparametrelere bağlıdır. Model parametreleri, ağırlıklar (weights) ve önyargılar (biases) gibi özelliklerdir. Hiperparametreler ise

eđitim s¼recini y¼nlendiren ve model eđitiminden ¼nce belirlenmesi gereken ¼zelliklerdir. Hiperparametrelerin, verilere ve g¼reve uygun Őekilde belirlenmesi, modelin baŐarısını ¼nemli ¼l¼de arttırır.

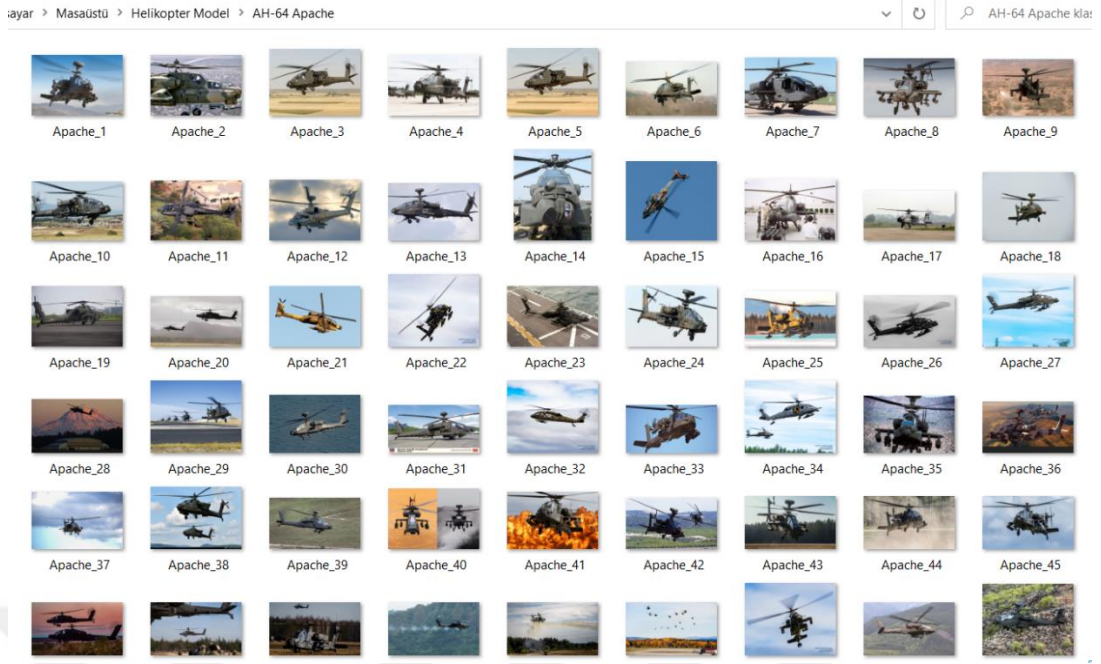
Sinir ađı modelinin belirli bir sınıfı tanıma kabiliyeti, bu sınıf i¼in kullanılan etiketlenmiŐ g¼r¼nt¼ verilerinin sayısı ile dođru orantılıdır. Bu, modelin bu sınıfı ne kadar iyi ¼ğrenebildiđini ve tanıyabildiđini g¼sterir. Daha fazla etiketlenmiŐ veri, genellikle daha iyi sınıflandırma sonu¼larına yol a¼ar. Bilgisayarlı g¼rmede, daha ¼nceden eđitilmiŐ bir derin sinir ađı modelinin, belirli bir hedef i¼in uyarlanması transfer ¼ğrenme y¼ntemi ile ger¼ekleŐtirilmektedir.

Transfer ¼ğrenme, eđitilmiŐ bir modelin elde ettiđi bilgileri baŐka bir veri k¼mesini ¼ğrenmek i¼in kullanma y¼ntemidir. Bu teknik, mevcut bir modelin bilgi birikimini yeni bir ¼ğrenme g¼revine aktararak, hedef alandaki ¼ğrenmeyi iyileŐtirmeyi hedefler [163].

İHA'ya g¼r¼nt¼ aracılıđıyla belirli bir nesnenin tespit edilmesi g¼revi atandıđı d¼Ő¼n¼lerek SSD MobilnetV2 evriŐimsel sinir ađı modeli, transfer ¼ğrenme y¼ntemi ile yeniden eđitildi ve hiperparametre ayarlarıyla g¼reve uygun olarak optimize edildi. İHA'nın nesne tanıma ve sınıflandırmada belirli bir g¼revi vardır: AH-64 Apache askeri helikopterini tespit etmek. Eđitim s¼recinin aŐamaları, aŐađıda alt baŐlıklar halinde detaylandırıldı.

#### **3.4.5.1 Veri setinin hazırlanması**

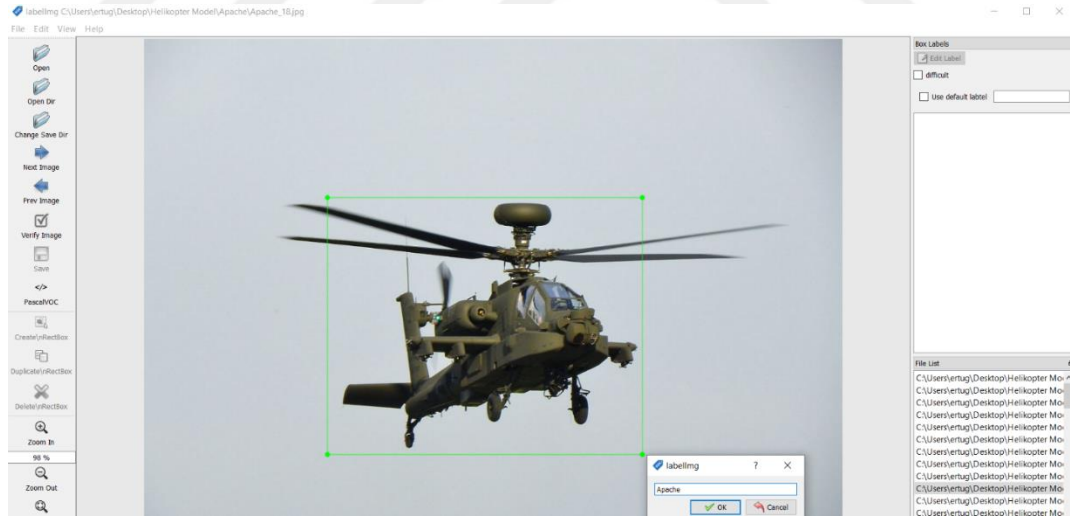
SSD MobilNetV2 sinir ađının eđitiminde; eđitim, dođrulama ve test verileri olarak AH-64 Apache askeri helikopterinin internet ortamından elde edilen, farklı konum mesafelerine ve farklı perspektif a¼ılarına sahip toplamda 436 g¼r¼nt¼s¼ kullanıldı. Bu g¼r¼nt¼lerin bir kısmı Őekil 3.14'deki g¼r¼lmektedir.



**Şekil 3.14:** AH-64 Apache Helikopterine ait Görüntüler

**Kaynak:** ([https://en.wikipedia.org/w/index.php?title=Boeing\\_AH](https://en.wikipedia.org/w/index.php?title=Boeing_AH), 2023)

Bu görüntülerin her biri, açık kaynak etiketleme yazılımı olan LabelImg kullanılarak "Apache" sınıfı olarak etiketlendi (Şekil 3.15).



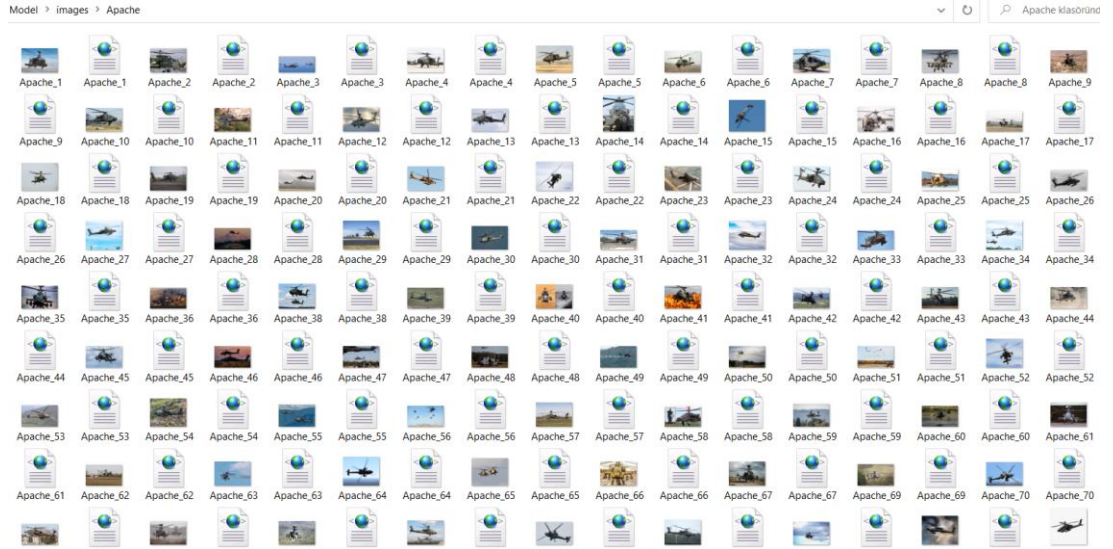
**Şekil 3.15:** AH-64 Apache Helikopterinin Etiketlenmesi

LabelImg yazılımı, her bir etiketleme işleminde; görüntüde işaretlenmiş olan nesnenin piksel konumunu ve nesnenin sınıfını içeren, PASCAL VOC formatında, görüntü ile aynı ismi taşıyan XML dosyası oluşturur (Şekil 3.16).

```
Apache_18.xml
1 <annotation>
2   <folder>Apache</folder>
3   <filename>Apache_18.jpg</filename>
4   <path>C:\Users\vertug\Desktop\Apache\Apache_18.jpg</path>
5   <source>
6     <database>Unknown</database>
7   </source>
8   <size>
9     <width>1280</width>
10    <height>960</height>
11    <depth>3</depth>
12  </size>
13  <segmented>0</segmented>
14  <object>
15    <name>Apache</name>
16    <pose>Unspecified</pose>
17    <truncated>0</truncated>
18    <difficult>0</difficult>
19    <bndbox>
20      <xmin>362</xmin>
21      <ymin>293</ymin>
22      <xmax>911</xmax>
23      <ymax>765</ymax>
24    </bndbox>
25  </object>
26 </annotation>
27
```

Şekil 3.16: PASCAL VOC Formatında XML Dosyası

Veri setimizde yer alan 436 görüntünün etiketlenmesi sonucunda 436 adet XML dosyası elde edildi (Şekil 3.17).



Şekil 3.17: AH-64 Apache Helikopterin Görüntüleri ve Her Bir Görüntünün Etiketini İçeren XML Dosyaları

Veri seti olarak kullanılacak görüntüler ve her bir görüntüye ait XML formatındaki dosyalar derin öğrenme modelinin oluşturulması sırasında kullanılmaktadır. Model, görüntüde işaretlenmiş nesnenin sınıf ve pozisyon bilgisini XML dosyasından elde eder [165].

### **3.4.5.2 SSD MobilNetV2 sinir ağı modelinin özelleştirilmesi (Hiperparametre optimizasyonu) ve Raspberry Pi’de kullanılacak TFLite modeline dönüştürülmesi**

Belirli bir nesne tespit görevi için optimal model oluşturmak gerekir. Bu, sinir ağı modelinin doğruluğunu artıracak şekilde hiperparametre ayarlamasını gerektirir. Hiperparametrelerin optimizasyonu, deneme yanılma esasına göre uygulanan yoğun bir süreçtir ve genellikle belirli bir veri kümesi ve model eşleştirmesi için gerçekleştirilir ve yalnızca o belirli eşleştirme için en uygun ayarların elde edilmesiyle sonuçlanır [166].

Bu çalışmada, SSD MobilNetV2 derin öğrenme modeli kullanıldı ve belirli bir nesne olarak AH-64 Apache askeri helikopterini tanıması için transfer öğrenme yöntemi ile yeniden eğitilmesi amaçlandı. Bu kapsamda, veri seti, sinir ağı mimarisi, sınıflandırma sayısı ve görev tanımı göz önüne alınarak öngörülen model hiperparametre ayarları, eğitim öncesinde belirlendi. Eğitim ve test süreci Google Colab[167] NVIDIA Tesla A100 GPU makinesinde gerçekleştirildi. Eğitim süreci sırasında, Tensorboard görselleştirme aracı kullanıldı. TensorBoard, TensorFlow Kütüphanesi kullanılarak oluşturulan makine öğrenimi modellerinin eğitim ve değerlendirmesi ile ilgili çeşitli yönleri görselleştirmek için kullanılan bir araçtır. Tensorboard kullanmanın asıl amacı eğitim süreci, model mimarisi, hiperparametre konfigürasyonunu gözlemlemek ve çeşitli metrikler hakkında içgörüler sağlamaktır [168]. İlk öngörülen hiperparametre değerlerinin model üzerindeki etkileri, eğitim süreci sırasında Tensorboard aracılığıyla analiz edildi. Her defasında analiz sonuçlarına göre yeni değerler ile modelin doğru tahminlerde bulunabilmesi için gerekli olan hiperparametre ayarları saptanmaya çalışıldı.

Sonuç olarak, adım sayısının 40.000, aktivasyon fonksiyonun ReLU ve diğer hiperparametrelerin Çizelge 3.1’deki değerlerde ayarlanması durumunda, Şekil 3.18’deki görüldüğü gibi kayıp değerlerinin oldukça düşük olduğu ve modelin AH-64 Apache askeri helikopterini tanıma doğruluğunun bu parametreler ile arttığı belirlendi.

**Çizelge 3.1:** SSD MobilNetV2 Modelinin Hiperparametre Değerleri

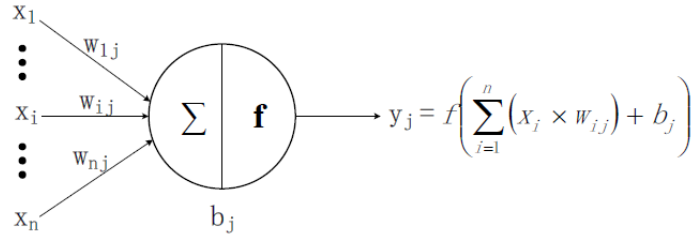
Parametre	Değer
Activation Function	ReLU
Classifier	Softmax
Optimizer	Stochastic Gradient Descent (SGD) + Momentum
Batch Size	16
Number of Step	40000

```
I1013 19:01:18.672041 134284502258304 model_lib_v2.py:705] Step 39900 per-step time 0.303s
INFO:tensorflow:{'Loss/classification_loss': 0.024091847,
'loss/localization_loss': 0.0023098686,
'loss/regularization_loss': 0.045754,
'loss/total_loss': 0.072155714,
'learning_rate': 0.008097499}
I1013 19:01:18.672388 134284502258304 model_lib_v2.py:708] {'Loss/classification_loss': 0.024091847,
'loss/localization_loss': 0.0023098686,
'loss/regularization_loss': 0.045754,
'loss/total_loss': 0.072155714,
'learning_rate': 0.008097499}
INFO:tensorflow:Step 40000 per-step time 0.306s
I1013 19:01:49.223052 134284502258304 model_lib_v2.py:705] Step 40000 per-step time 0.306s
INFO:tensorflow:{'Loss/classification_loss': 0.011748167,
'loss/localization_loss': 0.001472075,
'loss/regularization_loss': 0.045725882,
'loss/total_loss': 0.058946125,
'learning_rate': 0.007943453}
I1013 19:01:49.223366 134284502258304 model_lib_v2.py:708] {'Loss/classification_loss': 0.011748167,
'loss/localization_loss': 0.001472075,
'loss/regularization_loss': 0.045725882,
'loss/total_loss': 0.058946125,
'learning_rate': 0.007943453}
```

**Şekil 3.18:** SSD MobilNetV2 Modelinin Eğitim Sonundaki Kayıp ve Öğrenme Hızı Değerleri

Modelin özelleştirilmesi sırasında belirlenen adım sayısı (dolaylı olarak epok sayısı), bir dizi denemenin sonucunda sinir ağının performansına bakılarak belirlenmektedir. Ayrıca sinir ağında bulunan ve ağın hiperparametrelerinden biri olan aktivasyon fonksiyonunun da, model görevine uygun belirlenmesi önemli bir noktadır. Bu modelde kullanılan aktivasyon fonksiyonu ReLU'nun, modelin yüksek doğruluktaki tahmin başarısında önemli bir payı vardır. Nasıl bir payı olduğunu anlamak için genel anlamda aktivasyon fonksiyonunun sinir ağındaki rolüne ve ReLU'nun katkısını matematiksel olarak incelemek gerekir.

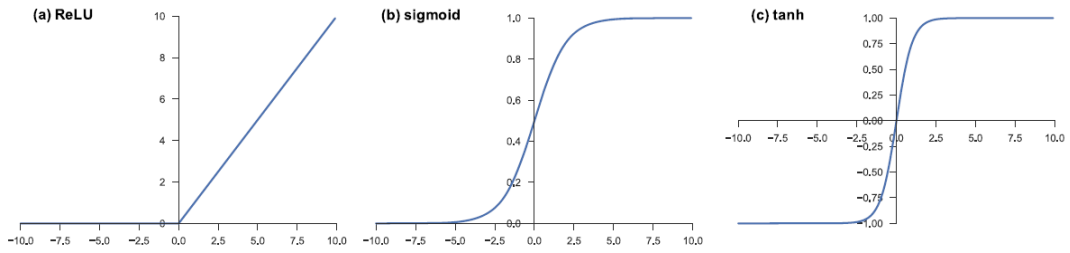
Bir sinir ağı içinde her nöron, önceki katmandaki nöronların çıkış değerini girdi olarak kabul eder ve işlenmiş değeri bir sonraki katmana ileterek bilgiyi aktarır. Çok katmanlı bir sinir ağı içinde, girdiler "aktivasyon fonksiyonu" tarafından işlenir. Şekil 3.19'da genel bir aktivasyon fonksiyonunun yapısı gösterilmektedir.



**Şekil 3.19:** Genel Bir Aktivasyon Fonksiyonunun Yapısı

**Kaynak:** (Li ve diğ., 2021)

Şekil 3.19'daki,  $x_i$  giriş özelliğini temsil eder;  $n$  adet özellik aynı anda nöron  $j$ 'ye giriş yapar;  $w_{ij}$ , giriş özelliği  $x_i$  ile nöron  $j$  arasındaki bağlantının ağırlık değerini temsil eder;  $b_j$ , nöron  $j$ 'nin içsel durumunu temsil eder, yani bias değeridir ve  $y_j$ , nöron  $j$ 'nin çıkışını ifade eder.  $f(\cdot)$  aktivasyon fonksiyonudur, bu sigmoid fonksiyonu,  $\tanh(x)$  fonksiyonu, düzeltilmiş doğrusal birimler (Rectified Linear Unit-ReLU)) gibi yaygın olarak kullanılan üç aktivasyon fonksiyonundan biri olabilir (Şekil 3.20) (Çizelge 3.2).



**Şekil 3.20:** Yaygın Olarak Kullanılan Üç Aktivasyon Fonksiyonu: (a) ReLU, (b) Sigmoid ve (c) Tanh

**Kaynak:** (Yamashita, 2018)

**Çizelge 3.2:** ReLU, Sigmoid ve Tanh Fonksiyonlarının Denklemleri

Fonksiyon	Denklem	Türevi
ReLU	$f(x) = \max(0, x)$	$df(x)/dx = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases}$
Sigmoid	$f(x) = \frac{1}{1 + e^{-x}}$	$\frac{df(x)}{dx} = \frac{e^x}{(1 + e^x)^2}$
Tanh	$f(x) = \frac{\sinh(x)}{\cosh(x)} = \frac{e^x - e^{-x}}{e^x + e^{-x}}$	$\frac{df(x)}{dx} = 1 - f(x)^2$

**Kaynak:** (Kılıçarslan ve diğ., 2021)

Düzeltilmiş Doğrusal Birimler (Rectified Linear Unit - ReLU), derin öğrenme modellerinde en yaygın kullanılan aktivasyon fonksiyondur. ReLU fonksiyonu, girdi

olarak herhangi negatif bir deęer aldığında çıktı olarak 0'ı döndürür, herhangi bir pozitif x deęeri için aynı deęeri geri döndürür [120]. Bu sebeple, Denklem (3.8)[170] veya Denklem (3.9) [171] ile şu şekilde:

$$f(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases} \quad (3.8)$$

veya

$$f(x) = \max(0, x) \quad (3.9)$$

ifade edilebilir.

Birçok derin öğrenme modelinde ve özellikle evrişimli sinir ağları (CNN'ler) gibi görüntü işleme görevlerinde yaygın olarak kullanılan aktivasyon fonksiyonu ReLU'nun, CNN'lerdeki önemini daha ayrıntılı bir şekilde açıklayan bazı ana nedenler:

**Hızlandırılmış Eğitim:** ReLU, gradyanların yakınsamasını hızlandırır. Diğer aktivasyon fonksiyonları, özellikle sigmoid veya tanh gibi, belirli deęer aralıklarında gradyanları çok küçük yapma eğilimindedir. Bu, eğitimin yavaş ilerlemesine neden olabilir, çünkü gradyanların ağırlıkların güncellenmesine katkı sağlaması gerekir. Ancak ReLU, pozitif girişler için gradyanı 1'e ayarlar ve bu, gradyan tabanlı optimizasyon algoritmalarının daha hızlı yakınsamasına yol açar [172].

**Kaybolan Gradyan Problemini Önler:** Sigmoid veya tanh gibi bazı aktivasyon fonksiyonları, derin sinir ağlarında "kaybolan gradyan" sorununa yol açabilir. Bu, derin katmanlarda gradyanların çok küçük hale gelmesi veya türevlerin çok küçük olması durumunda gerçekleşir. ReLU, pozitif girişler için gradyanı 1 olduğu için, bu tür problemlerin olasılığını azaltır.

**Ağın Hesaplama Karmaşıklığını Azaltır:** ReLU, negatif girişler için 0 çıktı üretir. Bu, birçok nöronun pasif hale gelmesine neden olur, yani belirli bir katmandaki yalnızca bazı nöronlar aktif hale gelir. Bu, ağın hesaplama karmaşıklığını azaltır ve ağı daha etkili hale getirir. Ayrıca düşük ağırlıklarla pasif hale gelen nöronlar, ağın düzensizliklerle başa çıkmasına yardımcı olabilir.

**Öğrenme Kapasitesini Artırır:** ReLU, özellikle büyük veri kümeleri üzerinde eğitildiğinde, daha fazla öğrenme kapasitesi sunar. Negatif deęerlerin pasif

hale gelmesi ve pozitif deęerlerin doęrusal olarak aktarılması, daha fazla örüntü ve karmaşıklık yakalama yeteneęi saęlar.

Sonuç olarak, ReLU aktivasyon fonksiyonu, CNN’lerde eęitimi hızlandırmaya, kaybolan gradyan sorununu önlemeye [170] , hesaplama karmaşıklığını azaltmaya ve daha fazla öğrenme kapasitesi sunmaya yardımcı olur. Bu nedenle, derin öğrenme ve özellikle görüntü işleme uygulamalarında yaygın bir tercih haline gelmiştir.

Model eęitimi öncesinde belirlenen dięer bir hiperparametre ise optimizasyon algoritmasıdır (optimizer-optimize edici). Sinir aęı mimarilerinin eęitilmesi sırasında karşılaşılan ortak sorun, kısa sürede en yüksek tahmin doęruluęuna ulaşmaktır [173]. Sinir aęlarının doęruluęunun artırılması için, veri setini artırma, katman sayısını artırma veya azaltma, nöron sayısını deęiştirme, farklı sinir aęı türleri ekleme (örneğin, evrişimsel sinir aęına, uzun-kısa vadeli bellek (LSTM) katmanları ekleme) ve benzeri birçok yaklaşımlar bulunmaktadır. Ancak bu yaklaşımlar, tüm modeller için genelleştirilmemiştir [173].

Optimize edici, sinir aęlarının eęitimini geliştirmek ve sinir aęının kayıp fonksiyonu optimize etmek için kullanılır ve sinir aęlarının doęruluęunun artırılması için kullanılan genel yöntemdir. Kayıp fonksiyonu, bir makine öğrenimi modelinin eęitilmesi sırasında gerçek ve tahmin edilen deęerler arasındaki farkı ölçen bir metriktir [139]. Modelin ne kadar iyi veya kötü performans gösterdięini belirlemeye yardımcı olur. Kayıp fonksiyonu, modelin parametrelerini (aęırlıklar, ön yargılar vb.) optimize etmek için kullanılır.

Model eęitimi için “Stochastic Gradient Descent (SGD) + Momentum” optimize edici kullanıldı. Stokastik gradyan iniři (SGD), makine öğrenimi ve özellikle derin öğrenme modelleri için yaygın olarak kullanılan bir optimizasyon algoritmasıdır [174]. Bu algoritma, eęitim sürecini hızlandırmak için tercih edilir. Her bir adımda, SGD mevcut eęitim örneęinin gradyanını hesaplar ve model parametrelerini bu gradyan yardımıyla günceller. Bu sayede tüm veri seti yerine rastgele örneklerle eęitim yapılabilir, bu da hesaplama açısından daha verimli olabilir. Bu yöntem, minimuma yakınsamak için gradyan iniřini kullanırken ani dalgalanmaları azaltarak veya atlayarak daha hızlı yakınsama saęlayabilir. Momentum, optimizasyon sürecinin yakınsama hızını ve istikrarını artırmak için

SGD'ye sonradan eklenmiştir [175]. Önceki gradyanların yönünü ve hızını takip ederek daha tutarlı ve istikrarlı güncellemeler sağlar.

Sinir ağı eğitilirken "öğrenme hızı" (learning rate) olarak adlandırılan hiperparametre değeri oldukça önemlidir. Öğrenme hızı, bir sinir ağının ağırlık güncellemelerinin ne kadar büyük veya küçük olacağını kontrol eden bir faktördür. Ağırlık güncellemeleri, ağı eğitimi verilerine daha iyi uyarlanabilmesi için yapılan değişikliklerdir.

Öğrenme hızı, eğitim sırasında her adımda ağı ağırlıklarının ne kadar değiştirileceğini belirler. Eğer öğrenme hızı çok yüksekse, ağı hızlı bir şekilde eğitim verilerine uyum sağlayabilir, ancak bu durum ağı aşırı uyumlu hale getirme riskini artırabilir. Eğer öğrenme hızı çok düşükse, ağı eğitimi daha istikrarlı olabilir, ancak daha uzun sürebilir.

Genel olarak, öğrenme hızı  $\alpha$  karakteriyle gösterilir.  $\alpha$  değeri 0 ile 1 arasında tanımlanır. Öğrenme hızı azalmasının matematiksel denklemi [176]:

$$\alpha = \frac{1}{1+(\text{decay rate} \cdot \text{epoch number})} \alpha_0 \quad (3.10)$$

şeklinde Denklem (3.10) ile ifade edilir. Denklem (3.10)'da:

**$\alpha$  (Alpha):** Optimizasyon sırasında adım büyüklüğünü belirleyen öğrenme oranı,

**$\alpha_0$  (Alpha naught):** Eğitimin başında belirlenen başlangıç öğrenme oranı,

**Epok numarası (Epoch number):** Eğitim sırasındaki geçerli epoch (dönem) sayısıdır ve genellikle 1'den başlayıp her bir veri seti üzerinden tam geçişte bir artar (+1),

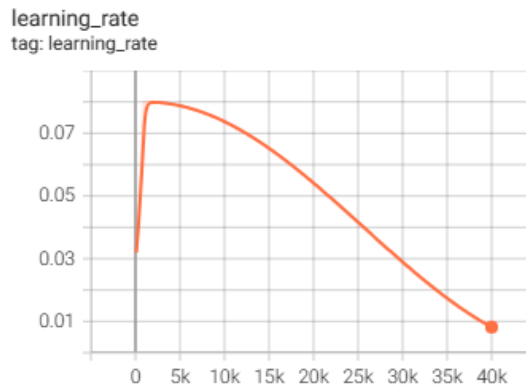
**Decay rate (Azalma hızı):** Öğrenme oranının azaldığı hızı belirleyen bir parametredir.

Optimal öğrenme hızı, deneme yoluyla belirlenir ve denklem (3.10)'a göre epoch sayısı değiştirilerek optimal öğrenme hızı elde edilebilir. Genel olarak, öğrenme hızı, sinir ağlarının eğitimini kontrol etmek ve istenmeyen sonuçları önlemek için önemli bir hiperparametredir. İyi bir öğrenme hızı, ağı hızlı ve etkili bir şekilde öğrenmesine yardımcı olabilir. Çizelge 3.3, altı farklı adım sayısındaki

öğrenme hızının değerlerini ve Şekil 3.21'deki grafik ise tüm adımlardaki öğrenme hızının değerlerini yansıtmaktadır.

**Çizelge 3.3:** Adım Sayısına Bağlı Öğrenme Hızı Değerleri

Adım Sayısı	Öğrenme Hızı
1.000	0.08
5.000	0.078691795
10.000	0.07352352
20.000	0.0538146
30.000	0.028618898
40.000	0.007943453



**Şekil 3.21:** SSD MobilNetV2 Modelinin Eğitim Sırasında Adım Sayısına Göre Öğrenme Hızı Değerleri Grafiği

Şekil 3.21'deki öğrenme hızı değerlerine bakıldığında, adım sayısı arttığında öğrenme hızının azaldığı görülmektedir. Bu da AH-64 Apache askeri helikopterini tanıması için eğitilen ağı, eğitim süresince istikrarlı olduğunun göstergesidir.

Bir sinir ağı modeli, ağırlıklar (weights) ve önyargılar (biases) gibi parametrelere sahiptir. Bu parametreler, girdi verilerini çıkışa dönüştüren matematiksel işlemleri tanımlar. Modelin performansını ölçmek için bir kayıp fonksiyonu kullanılır. Bu fonksiyon, modelin tahminlerinin gerçek etiketlerden ne kadar uzak olduğunu ölçer [177]. Eğer tahminler mükemmelse (yani kayıp sıfırsa), model hiçbir hata yapmamış demektir. Model, eğitim verileri üzerinde iteratif (bir önceki katmanın çıktısı bir sonraki katmanın girdisidir) olarak eğitilir. Bu süreç, girdi verilerini modele vererek ve ardından modelin tahminlerini gerçek etiketlerle karşılaştırarak gerçekleşir.

Eğitim sırasında modelin ağırlıkları ve önyargıları güncellenir. Bu güncellemeler, kayıp fonksiyonunun değerini azaltmaya yönelik yapılır. Yani,

model, hata yaptığı örneklerde ağırlıklarını ve önyargılarını ayarlar. Eğitim süreci birçok iterasyon (tekrar) gerektirebilir. Model, veri kümesi üzerinde birden fazla geçiş yaparak ağırlık ve önyargıları iyileştirir. Kayıp fonksiyonu, modelin ne kadar başarılı olduğunu ölçen önemli bir ölçüttür ve bu fonksiyonun değerini en aza indirme, modelin iyileştirilmesi için ana hedeflerden biridir. Şekil 3.22, Apache AH-64 askeri helikopterin tespiti için yeniden eğitilen SSD MobilNetV2 sinir ağı modelinin, 0-40.000 adım sayısına karşılık toplam kayıp (total\_loss) değerlerinin değişimini göstermektedir.



**Şekil 3.22:** SSD MobilNetV2 Modelinin Eğitim Sırasında Adım Sayısına Göre Toplam Kayıp Değerleri Grafiği

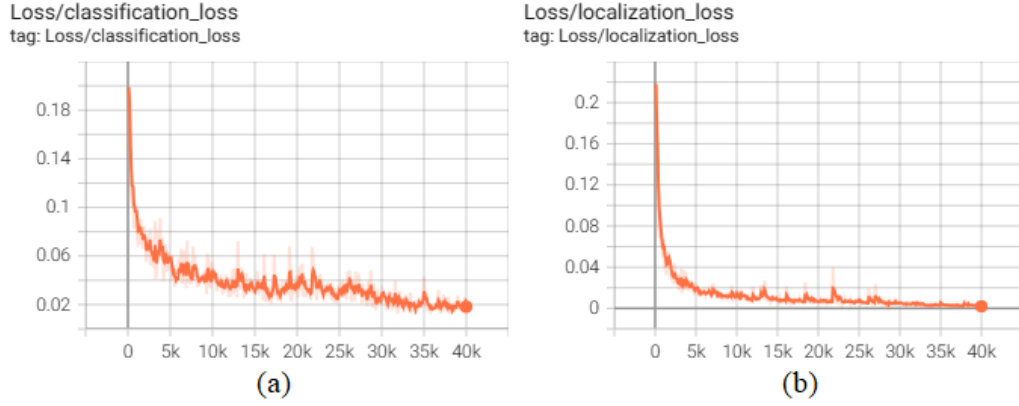
Grafik üzerinde ilk 100 adım ve sonrasında her 5000 lik adıma karşılık gelen “total\_loss” (toplam kayıp) değerleri Çizelge 3.4’de yer almaktadır. Bu veriler ışığında modelin başarılı olduğu söylenebilir.

**Çizelge 3.4:** Adım Sayısına Bağlı Toplam Kayıp Değerleri

Adım Sayısı (num_steps)	Toplam kayıp (total_loss)
100	0.57075876
5.000	0.20305446
10.000	0.1641489
15.000	0.11425168
20.000	0.110265195
25.000	0.10409389
30.000	0.093730286
35.000	0.099287495
40.000	0.058946125

Bu çalışmada olduğu gibi modelin özellikle belirli bir nesneyi sınıflandırması isteniyorsa, “total\_loss”, “learning\_rate” dışında “classification\_loss” ve “localization\_loss” değerleri ve bunlara ait grafikler de oldukça önemlidir.

Şekil 3.23’da, SSD MobilNetV2 modelinin, Apache AH-64 askeri helikopteri sınıflandırma eğitimine ait sırasıyla sınıflandırma (classification\_loss) ve yerleştirme (localization\_loss) kayıp değerlerinin adım sayısına göre değişim grafikleri görülmektedir.



**Şekil 3.23:** SSD MobilNetV2 Modelinin Eğitim Sırasında Adım Sayısına Göre (a)Sınıflandırma Kayıp Değerleri Grafiği ile (b)Yerleştirme Kayıp Değerleri Grafiği

Grafiklere bakıldığında kayıp değerlerinin adım sayısı arttıkça azaldığı ve eğitimin sonunda sıfıra yaklaştığı görülmektedir. Çizelge 3.5, 40.000 adım sonundaki “classification\_loss” ve “localization\_loss” değerlerini göstermektedir.

**Çizelge 3.5:** Adım Sayısına Bağlı Classification\_Loss Ve Localization\_Loss Değerleri

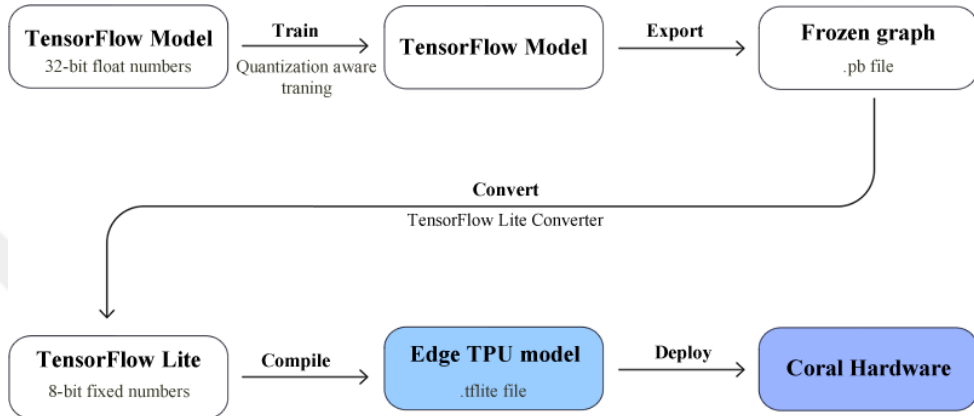
	Adım Sayısı	Kayıp Değeri
classification_loss	40.000	0.011748167
localization_loss	40.000	0.001472075

Bu değerler, transfer öğrenme yöntemi ile eğitilen SSD MobilNetV2 modelinin, AH-64 Apache askeri helikopterini sınıflandırmadaki başarısını işaret etmektedir.

SSD MobilNetV2 modeli, tüm süreç sonunda TensorFlow Lite (TFLite) formatına dönüştürüldü. TensorFlow Lite, TensorFlow’un mobil ve gömülü cihazlar için tasarlanmış hafif bir sürümüdür. TensorFlow Lite formatı, modelin hafif ve hızlı çalışmasını sağlar, bu da sınırlı işlem kapasitesine sahip İHA’nın nesne tespit görevini etkin bir şekilde gerçekleştirmesini sağlar. TensorFlow Lite modelleri, 32 bitlik parametre verilerini 8 bitlik gösterimlere (Edge TPU için gerekli olan) dönüştüren niceleme yoluyla daha da küçük ve daha verimli hale getirilebilir. Bir model doğrudan TensorFlow Lite ile eğitilemez. Öncelikle TensorFlow modeli

eğitilir ve ardından TensorFlow Lite dönüştürme adımları kullanılarak eğitilmiş olan model bir TensorFlow dosyasından (bir. pb dosya gibi) bir TensorFlow Lite dosyasına (.tflite) dönüştürülür.

Şekil 3.24, Edge TPU ile uyumlu bir model oluşturmaya yönelik temel süreci göstermektedir. Model oluşturma sürecinde standart TensorFlow araçları kullanılır. Bir TensorFlow Lite modeli, Edge TPU derleyicisi kullanılarak Edge TPU ile uyumlu (edgetpu.tflite) bir dosya oluşturulur.



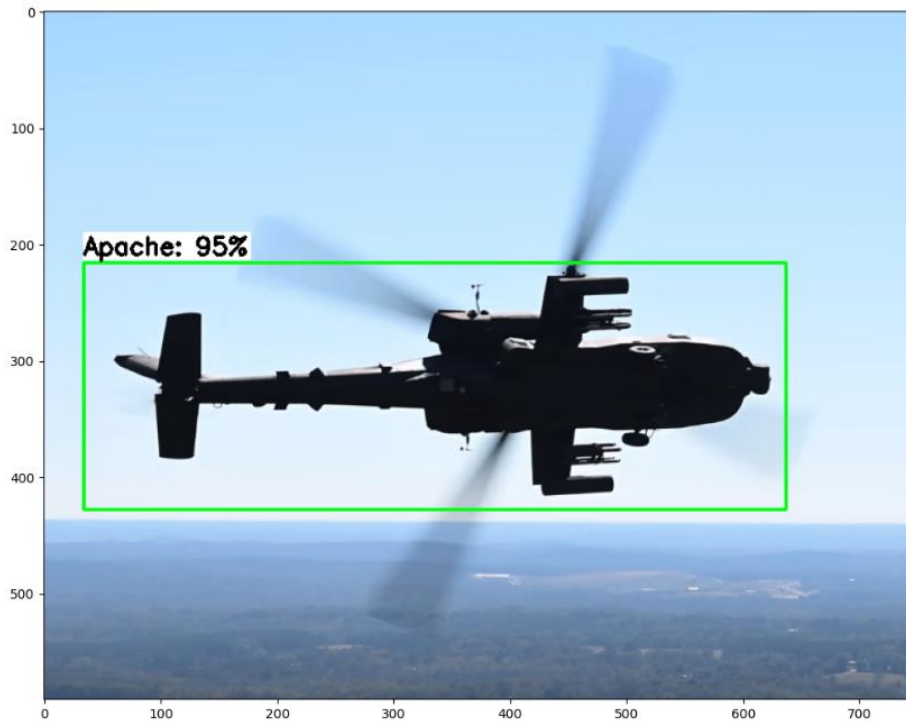
**Şekil 3.24:** Edge TPU Modelini Oluşturmaya Yönelik Temel Süreç

Eğitim sonunda SSD MobilNetV2 modelinin, Coral Hardware (Edge TPU yardımcı işlemcisi) ile çalışacak şekilde Edge TPU. tflite modeli oluşturuldu. Böylece, İnsansız hava aracı platformundaki gömülü sistemde kullanılabilir hızlı bir model elde edildi.

Sonuç olarak, öncelikle SSD MobilNetV2 modeli, AH-64 Apache askeri helikopterini sınıflandırabilmesi için transfer öğrenme yöntemi kullanılarak ve model parametreleri göreve özgü özelleştirilerek eğitildi ve daha sonra TensorFlow Lite modeli oluşturuldu. Modelin nesne sınıflandırmadaki başarısı eğitimde kullanılan veri miktarı ile yakından ilişkilidir. Daha çok görsel veri daha yüksek doğruluk anlamına gelmektedir. Model eğitimi sonunda, eğitim süresince hiç kullanılmamış AH-64 Apache askeri helikopter görüntüleri ile yapılan testlerde modelin Apache askeri helikopteri, yüksek doğrulukta tespit etme yeteneğine sahip olduğu görülmektedir (Şekil 3.25-3.34). Şekil 3.25 ve Şekil 3.26'daki görüntülere bakıldığında SSD MobilNetV2 sinir ağı modelinin farklı görüntü açısına sahip Apache helikopterlerini yüksek doğrulukla tespit edebildiği görülmektedir.

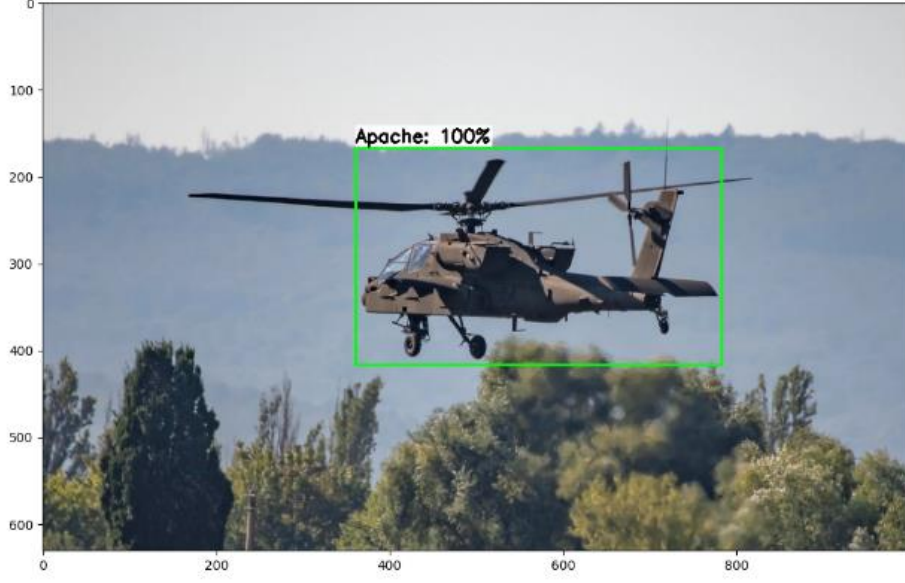


**Şekil 3.25:** SSD MobilNetV2 Modelinin Eğitim Sonundaki Test Sonucu-1



**Şekil 3.26:** SSD MobilNetV2 Modelinin Eğitim Sonundaki Test Sonucu-2

Şekil 3.27 ve 3.28’de görüldüğü üzere SSD MobilNetV2 sinir ağı modelinin test görüntülerinde %100 ve %99 gibi yüksek doğrulukla Apache helikopterlerini tespit ederek sınıflandırabildiği görülmektedir.



**Şekil 3.27:** SSD MobilNetV2 Modelinin Eğitim Sonundaki Test Sonucu-3



**Şekil 3.28:** SSD MobilNetV2 Modelinin Eğitim Sonundaki Test Sonucu-4

Şekil 3.29 ve Şekil 3.30, SSD MobilNetV2 sinir ağı modelinin Apache helikopterini yan açıdan tespit edip sınıflandırabildiğini göstermektedir.

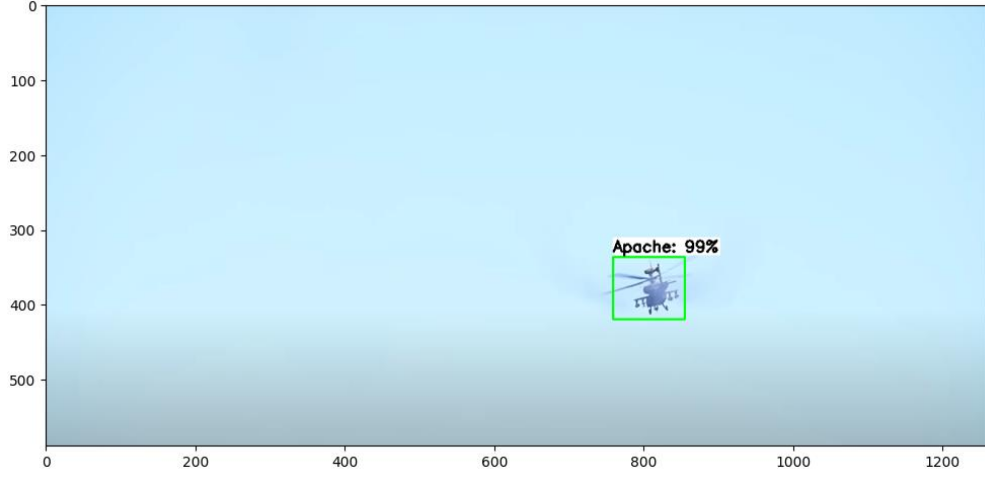


**Şekil 3.29:** SSD MobilNetV2 Modelinin Eğitim Sonundaki Test Sonucu-5



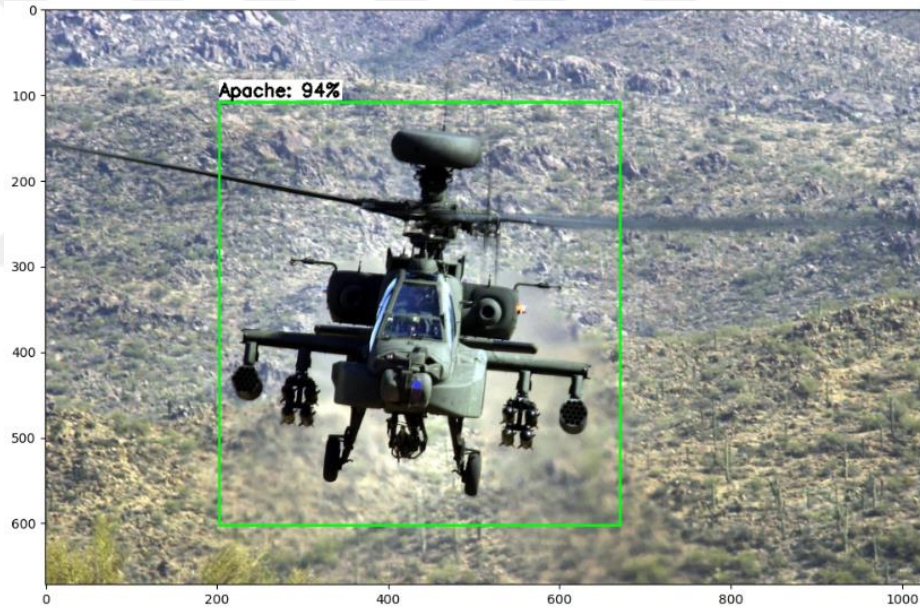
**Şekil 3.30:** SSD MobilNetV2 Modelinin Eğitim Sonundaki Test Sonucu-6

Model, test görüntüsünde yer alan uzaktaki bir Apache helikopterini %99 gibi bir doğruluk değeri ile tespit edebilmiştir (Şekil 3.31).



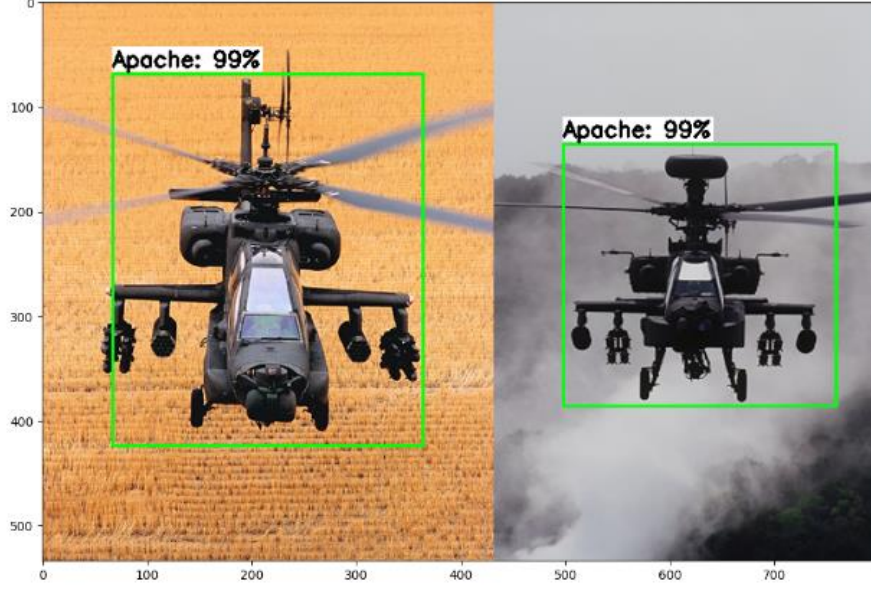
**Şekil 3.31:** SSD MobilNetV2 Modelinin Eğitim Sonundaki Test Sonucu-7

SSD MobilNetV2 derin öğrenme modelinin test görüntülerindeki en düşük doğruluk değeri %94 olarak çıkmıştır (Şekil 3.32).

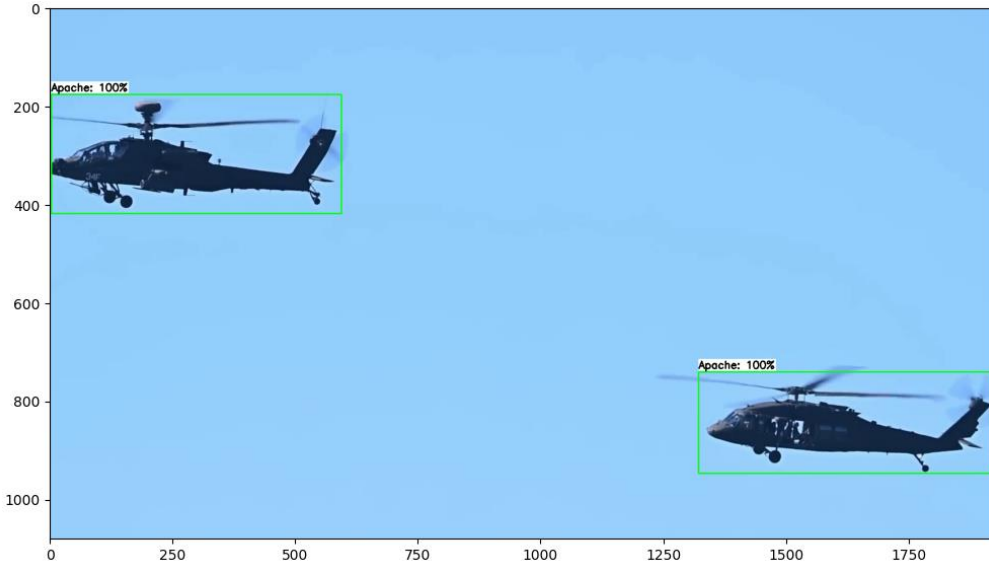


**Şekil 3.32:** SSD MobilNetV2 Modelinin Eğitim Sonundaki Test Sonucu-8

Şekil 3.33 ve Şekil 3.34'de görüldüğü üzere SSD MobilNetV2 sinir ağı modeli, tek bir görüntü karesindeki iki adet Apache helikopterini tespit ederek sınıflandırabilmektedir.



**Şekil 3.33:** SSD MobilNetV2 Modelinin Eğitim Sonundaki Test Sonucu-9



**Şekil 3.34:** SSD MobilNetV2 Modelinin Eğitim Sonundaki Test Sonucu-10

Sinir ağı modelinin eğitim performansı, eğitim sırasında ve sonrasındaki “classification\_loss” (sınıflandırma kaybı), “localization\_loss” (yerelleştirme kaybı), “total\_loss” (toplam kayıp), “learning\_rate” (öğrenme hızı) değerlerine ait grafikler analiz edilerek değerlendirilmektedir. Modelin eğitimi sırasında tüm grafikler analiz edilerek hiperparametreler ayarlandı. Yukarıda ayrı ayrı analiz edilen eğitim sonrasına ait tüm grafik değerlendirmeleri, modelin sınıflandırmadaki başarısını kanıtlamaktadır.

Model, hem gerçek zamanlı hem de uçuş sırasında İHA kamerası tarafından kaydedilen video görüntülerinden AH-64 Apache askeri helikopterlerini yüksek doğrulukla tespit edebilir. İHA'nın görüntü ile nesne tanıma modeli aynı yaklaşımla tespit edilmesi istenen başka nesnelere için de özelleştirilebilir. Böylece, İHA askeri operasyonlar veya güvenlik uygulamaları gibi çeşitli alanlarda kullanılabilir.

### 3.5 Ses İşleme

Derin öğrenme alanındaki gelişmeler, ses sinyalinin derin öğrenme yöntemleri ile işlenebilmesine olanak sağlamıştır [178]. Konuşma tanıma, müzik bilgisi alma ve çevresel seslerin sınıflandırılması gibi birçok konuda derin öğrenme teknolojilerinden yararlanılmaktadır [179-184]. Seslerin sınıflandırılması, genellikle ses verilerinden özelliklerin çıkarılması sürecini içermektedir [185-189].

Ses verisini, verimli bir şekilde işleyebilmek ve sese ait özellikleri öğrenmek için evrimsel sinir ağları kullanılmaktadır [190-193]. Bu bölümde, ses sınıflandırması ve derin öğrenme tabanlı ses sınıflandırıcısı YAMNet modeli hakkında bilgi verildikten sonra İHA'ya ses sınıflandırma ile nesne tespit yeteneği kazandırılmasına yönelik adımlar anlatılmaktadır.

#### 3.5.1 Ses sınıflandırması

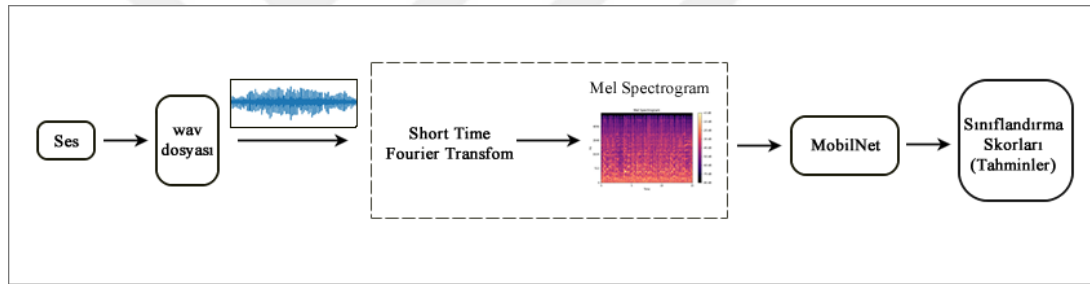
Ses sınıflandırılmasının temeli, ses verilerinden özelliklerin çıkarılmasına dayanmaktadır. Ses özelliklerinin çıkarılması, ses kaynaklarını tespit etmek için gerçekleştirilir. Ses sinyalleri genellikle işlenmek üzere iki boyutlu zaman-frekans temsillerine dönüştürülür. Son dönemlerde, ses verisini verimli bir şekilde işleyebilmek ve yüksek doğruluk seviyeleriyle ses sınıflandırması yapabilmek için evrimsel sinir ağları kullanılmaktadır [194]. Derin öğrenme modellerinin Raspberry Pi gibi kaynakları sınırlı gömülü cihazlara entegrasyonu zorlu bir süreçtir. Bu tür cihazlarda işlem hızını artırmak için kullanılan Tensor İşleme Birimleri (TPU'lar), daha iyi performans elde etmek için büyük bir önem taşır.

Bu çalışma, önceden eğitilmiş ses sınıflandırma modeli YAMNet'in belirli bir görev için "transfer öğrenme" yöntemi ile eğitilmesini, özellikle Raspberry Pi (RPI) ve TPU gibi kaynak sınırlı cihazlarda daha verimli ve hızlı çalışabilen TensorFlow Lite modeline dönüştürülmesini hedeflemektedir.

### 3.5.2 Derin öğrenme tabanlı YAMNET ses sınıflandırma modeli

YAMNet, ses özelliklerini çıkarmak ve ses sınıflandırması yapmak için Google TensorFlow ekibi tarafından geliştirilen [195], MobileNetV1 derinlemesine ayrılabilir evrişim mimarisini içeren ve AudioSet ontolojisi [144] tarafından tanımlanan ses sınıflarını kullanarak 521 farklı ses olayını tahmin eden, önceden eğitilmiş bir sinir ağı modelidir. Model,  $[-1.0, +1.0]$  aralığında tek kanallı (mono), 16 kHz örnekler olarak temsil edilen, isteğe bağlı uzunlukta bir dalga biçimi içeren 1-D float32 Tensör veya NumPy dizisini, girdi olarak kabul eder [196].

YAMNet modelinin içinde özellik çıkarma katmanı yer almaktadır. Bu özellik çıkarma katmanı, ses verilerini spektrogramlara dönüştürür. Ardından, bu spektrogramlar MobileNet'e iletilir. Son olarak model, sınıflandırma skorlarını çıktı olarak verir. YAMNet'in özellik çıkarım ve ses sınıflandırma süreci aşağıdaki Şekil 3.35'de gösterilmektedir.



Şekil 3.35: YAMNet Ses Özellik Çıkarım ve Tahmin Süreci

Bu model, özel olarak ses özellikleri çıkarmak için uyarlanmış ve önceden eğitilmiştir, bu da onu ses sınıflandırma ve ses analizi gibi görevler için güçlü bir araç haline getirmektedir. YAMNet ile ses özelliklerini çıkarabilme süreci bir dizi matematiksel işlem ve dönüşüm sonunda gerçekleşmektedir. Modelin transfer öğrenme yöntemi ile eğitilmesi ve özelleştirilmesi için ses sinyallerinin özellik çıkarım sürecinin iyi bilinmesi gerekir. Bu süreç, aşağıda alt başlıklar halinde detaylandırılmaktadır.

#### 3.5.2.1 Ses verisi ön işleme hazırlığı

Sinir ağını eğitmek için kullanılan ses sinyalinin, YAMNet modelinin giriş formatına uygun olması gerekir. Bu sebeple, girdi olarak kullanılacak ses sinyali uygun formata dönüştürülmelidir. Ses sinyali, 16kHz örnekleme hızına sahip değilse

16kHz örnekleme hızına örnekleme ve stereo formatındaysa mono formatına (tek kanallı) dönüştürmek gerekir.

### 3.5.2.2 Mel spektrogramlarını çıkarma

Spektrogram, bir ses sinyalinin frekans spektrumunun zamana göre değişkenliğinin görsel temsilidir [197, 198] ve genellikle seslerin spektrogramları ayırt edici özellikler içerir [199]. YAMNet modeli için girdi olarak kullanılan özellik haritaları, seslerin zaman-frekans temsillerinden katsayılar çıkartılarak oluşturulur. Sesin frekans içeriği insanlar tarafından doğrusal olarak algılanmaz.

İnsanların sesleri ayırt etme yeteneği, düşük frekans aralıklarında (100Hz-200Hz, fark 100Hz ama iki katı) yüksek frekansta olduğundan daha hassas olup (10000Hz-10100Hz, fark 100Hz ama yüzde biri kadar), iki ses arasında tutarlı bir frekans farkı olduğunu varsayarsak, yüksek frekansta sesler arasındaki benzerlik daha fazla olabilir. Bu nedenle, insanlar çok fazla çaba harcamadan farklı sesleri kolaylıkla ayırt edebilirler. Mel spektrogramları, insanın ses algısının doğrusal olmayan yapısını göstermek için kullanılabilir. YAMNet modeli, makinelerin, tıpkı insanlar gibi sesleri ayırt edebilmesi için özellik haritası olarak log-mel spektrogramları üzerinde çalışır ve bunları ses analizinde kullanılır [200].

Mel ölçeği, insan işitmesinin frekans algısının logaritmik doğasını temsil etmek için kullanılan bir ölçüdür. Mel spektrogramı ise Mel ölçeği üzerindeki frekansların zamana bağlı değişimini görselleştiren bir grafik veya görseldir. Geleneksel spektrogramlarda frekanslar doğrudan Hertz cinsinden gösterilirken, Mel spektrogramı frekansları Mel ölçeği üzerinde temsil eder. Bu, insan işitmesine daha uygun bir şekilde frekans bileşenlerini görselleştirmeye olanak tanır ve ses işleme alanında yaygın olarak kullanılır.

Mel ölçeği ve frekans arasındaki bağlantı Denklem (3.11)'de gösterilmektedir [201]:

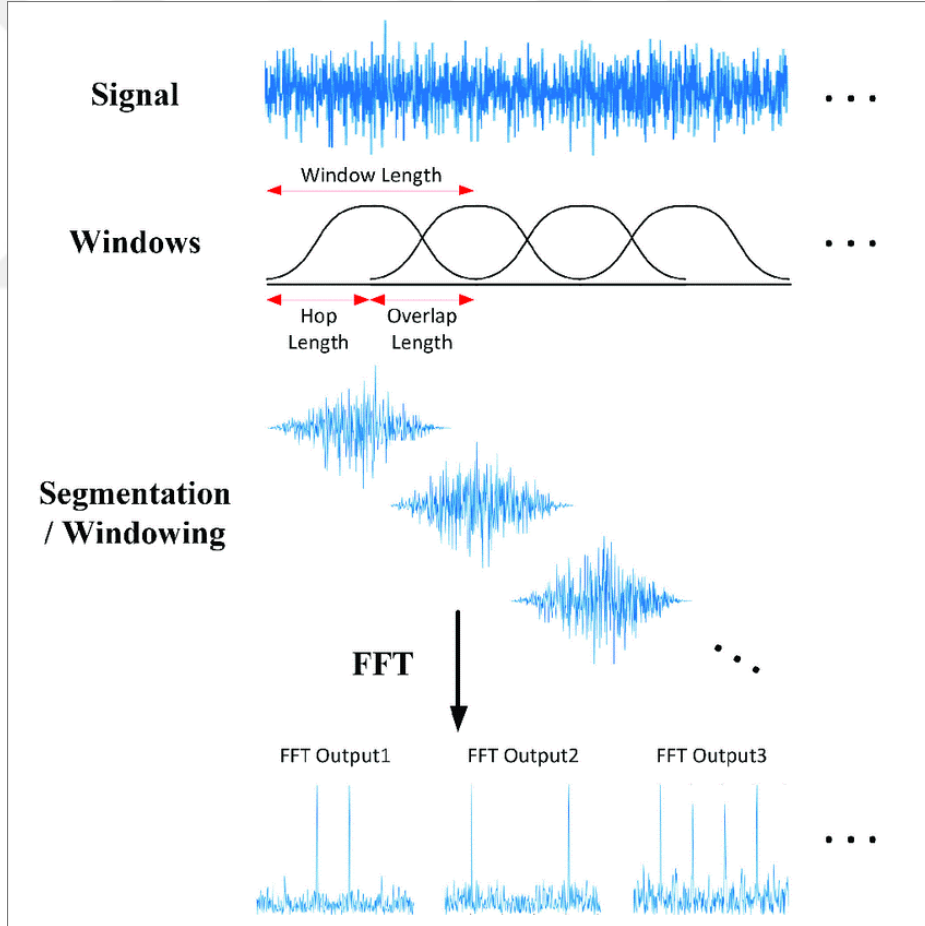
$$f_{mel} = \begin{cases} f & , f \leq 1000 \text{ Hz} \\ 2595 \log_{10}(1 + \frac{f}{700}), & f > 1000 \text{ Hz} \end{cases} \quad (3.11)$$

burada  $f_{mel}$ , mel frekansıdır ve  $f$ , doğrusal frekanstır.

Mel spektrogramını ön işlenmiş ses segmentinden elde etmek için ses sinyaline kısa zamanlı Fourier dönüşümü (STFT) uygulanması gerekmektedir [202].

Kısa Zamanlı Fourier Dönüşümü (STFT), sürekli sinyaller için bir zaman-frekans analiz tekniğidir, yani sinyali zaman-frekans alanına dönüştürür [203, 204]. Başka bir deyişle, STFT, uzun bir sinyalin yalnızca kısa süreli bir bölümünü dikkate alır ve Fourier dönüşümünü hesaplar [205, 206]. Yani, sinyali bir pencere fonksiyonuyla çarparak zaman alanı giriş sinyalini birkaç ayrı veya örtüşen çerçeveye böler ve ardından her çerçeveye hızlı Fourier dönüşümünü (FFT) uygular [178, 207].

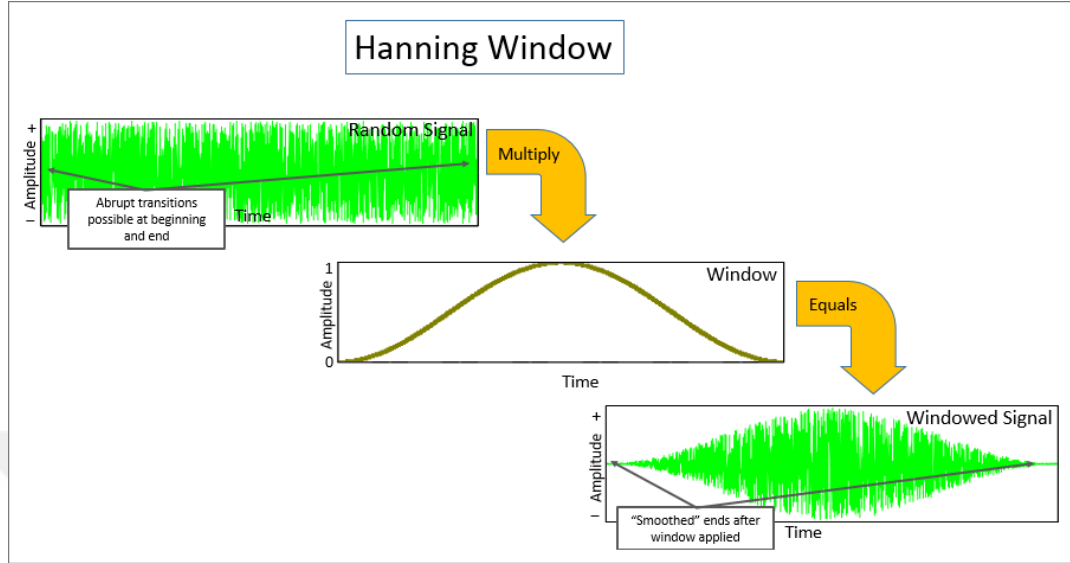
Fourier dönüşümleri pencereyi hareket ettirirken gerçekleştirildiğinden, bu teknik bir sinyalin zaman içindeki frekans içeriği değişikliklerini ölçebilir [208]. Şekil 3.36'da görüldüğü üzere her bölüme eşit büyüklükte bir pencere fonksiyonu ( $w(t)$ ) seçilir ve sinyalin başına yerleştirilir. Hop boyutuna (hop length) bağlı olarak pencere sinyal boyunca kayacak ve güncellenmiş dalga verileriyle yeni pencere için Fourier dönüşümünü tekrarlayacaktır [209].



**Şekil 3.36:** Genel bir Kısa Süreli Fourier Dönüşümü (STFT)

**Kaynak:** (Jeon ve diğ., 2020)

Şekil 3.37'e bakıldığında, STFT'nin esasen Fourier dönüşümü ile pencere fonksiyonu çarpımı olduğu göz önüne alındığında, STFT'ye pencereli Fourier dönüşümü veya zamana bağlı Fourier dönüşümü de denilebilir [211].



**Şekil 3.37:** Hann(Hanning) Penceresi

**Kaynak:** (community.sw.siemens.com, 2023)

Bu özellikleri nedeniyle STFT, radar sistemleri ve ses sinyali işleme sistemleri gibi zaman içinde frekans ölçümü gerektiren çeşitli alanlarda yaygın olarak kullanılmaktadır. Denklem (3.12), STFT'nin sürekli versiyonunu temsil eder ve matematiksel olarak şu şekilde yazılır [213]:

$$\text{STFT}\{x(t)\}(\tau, \omega) \equiv X(\tau, \omega) = \int_{-\infty}^{\infty} x(t)w(t - \tau)e^{-i\omega t}dt \quad (3.12)$$

burada  $w(\tau)$  pencere fonksiyonudur, genellikle Hann penceresi veya Gaussian penceresi kullanılır,  $x(t)$  dönüşüme uğrayacak sinyaldir,  $\omega$  frekansıdır.  $X(\tau, \omega)$ , sinyalin zaman ve frekansa göre fazını ve büyüklüğünü temsil etmektedir ve  $x(t)w(t-\tau)$ 'in Fourier dönüşümüdür. Denklemde (3.12)'de yer alan  $\tau$ , STFT'nin faz sonucundaki herhangi bir sıçrama süreksizliğini bastırmak için sıklıkla zaman eksenini ( $\tau$ ) ve frekans eksenini ( $\omega$ ) boyunca faz açma işlemi kullanılır [211]. Operasyonel gürültü ve titreşim ölçümleri yapılırken genellikle Hanning penceresi kullanılır. Birçok operasyonel sinyal, Şekil 3.37'de gösterildiği gibi doğası gereği rastgeledir. Hanning penceresi sıfır değerinde başlar ve sıfır değerinde biter (Şekil 3.37). Pencerenin ortasında bir değeri vardır. 0 ile 1 arasındaki bu kademeli geçiş, ölçülen sinyalin pencere ile çarpılması sırasında genliklerde yumuşak bir değişim sağlar ve bu da spektral sızıntının azaltılmasına yardımcı olur [212].

YAMNet modelinde, bir spektrogram, 25 ms'lik bir pencere boyutuna, 10 ms'lik bir pencere sıçramasına ve periyodik bir Hann penceresine sahip Kısa Zamanlı Fourier Dönüşümünün büyüklükleri kullanılarak hesaplanır. Rastgele sinyallere örnek olarak, motor çalışırken oluşan sesi verilebilir. STFT, sinyali frekans alanına dönüştürür ve ardından büyüklüğünü hesaplar. Daha sonra, büyüklük spektrogramına bir filtre bankası uygulanır ve farklı frekans bantlarındaki enerji hesaplanır. Son olarak, filtre bankası enerjilerinin logaritması alınarak mel spektrogramı elde edilir.

**YAMNet modelini kullanma:** Mel spektrogramı elde edildikten sonra, YAMNet modeline özellik çıkarmak için girdi olarak kullanılır. Model, bir dizi evrişimli ve havuzlama katmanından sonra tam bağlantılı katmanlar içerir. YAMNet modeli, Mel spektrogramını işler ve her ses segmenti için sabit uzunlukta bir özellik vektörü üretir. Bu özellik vektörü ses içeriğinin yüksek düzeyde temsilini yakalar.

**Son işleme:** YAMNet modelinden elde edilen özellik vektörlerine, ihtiyaca bağlı olarak ek işlemler uygulanabilir. Örneğin, normalleştirme, boyut indirgeme veya uygulamanın belirli gereksinimleri için başka işlemleri içerebilir.

**Eğitim veya çıkarım:** Çıkarılan özellik vektörleri, ses sınıflandırma veya ses analizi gibi çeşitli görevler için kullanılabilir.

### **3.5.3 Ses Verisinde Bulunan ve Ses Sınıflandırma Sonucunu Olumsuz Etkileyen Gürültülerin Tespiti ve Bu Gürültülerin Filtreleme Yöntemi**

Çevresel ses sınıflandırmasında, mikrofon ile elde edilen ses verisinde birçok nesne sesi veya gürültü olabilir. Bu sebeple, ses ile nesne tespit ve sınıflandırmada tahmin sonuçlarını olumsuz etkileyecek gürültülerin önce tespit edilmesi ve ardından filtrelenmesi gerekmektedir. Bu çalışmada, İHA üzerindeki bir ses algılayıcı (mikrofon) aracılığıyla elde edilen ses bilgileri kullanılarak, çevredeki nesnelerin tespit edilmesi hedeflenmektedir. Bu tür bir uygulama, ses tabanlı nesne tespiti olarak da adlandırılabilir. İHA, gömülü sistem üzerine monte edilmiş mikrofon aracılığıyla çeşitli ses kaynaklarından gelen sesleri algılar ve bu ses verilerini analiz eder. Nesne sınıflandırma modeli, nesnelerin çıkardığı sesleri tanıyarak, İHA'nın çevresindeki nesnelere algılamasına ve sınıflandırmasına yardımcı olabilir. Söz konusu ses sınıflandırma modeli, İHA üzerinde çalıştığından, sınıflandırmayı olumsuz etkileyen gürültülerin tespit edilmesi ve uygun filtreleme yöntemleri ile bu

gürültülerin filtrelenmesi, modelin verimliliğini ve performansını artıracaktır. Aşağıda, öncelikle gürültü kaynaklarının tespiti ve analiz yapıldıktan sonra gürültü sesine uygulanabilir filtreleme yöntemi hakkında bilgi verilmektedir.

### 3.5.3.1 Ses sınıflandırmasını olumsuz etkileyen gürültüler

Bir Hexacopter İHA ile ses sınıflandırmasındaki en önem gürültü kaynağı Hexacopter'in kendi sesidir. Hexacopter'in ürettiği sesin kaynağı, genellikle dış rotorlu fırçasız DC motorlarından ve pervanelerinden gelir [214]. İşte Hexacopter'in ürettiği sesin temel kaynakları:

**Motorlar:** Çok rotorlu İHA'ların çoğunda dış rotorlu fırçasız DC motorları kullanılır. Dış rotorlu BLDC motorlarda, gürültü kaynağının daha çok elektromanyetik ses olduğuna yönelik birçok çalışma bulunmaktadır [215-221]. Elektromanyetik sesin ana nedeninin, bir motorun statörüne veya rotoruna etki eden radyal kuvvet olduğu bildirilmektedir [222].

**Pervaneler:** Pervaneler, İHA'nın havada kalabilmesi için döner. Pervanelerin dönme hızı, genellikle RPM (devir/dakika) cinsinden ölçülür. Hızlı dönen pervaneler, daha yüksek frekansta bir gürültü üretebilir. Gürültü kaynakları genellikle kanat yüzeyinde aerodinamik olarak üretilir ve genel olarak kalınlık ve yükleme gürültüsü olarak sınıflandırılabilir.

Hexacopter'in ürettiği ses, motorların çalışma hızına ve pervanelerin dönme hızına bağlı olarak değişiklik gösterebilir. Motorların ürettiği elektromanyetik gürültü ve motor-pervane ikilisinin ürettiği gürültü aşağıda ayrı ayrı analiz edilmekte ve ardından bu sesleri filtreleme yöntemi detaylandırılmaktadır.

### 3.5.3.2 İnsansız hava aracı motorlarının ürettiği gürültülerin ses analizleri

Hexacopter'deki motorların ürettiği gürültü, manyetik alan içerisinde dönen rotorların oluşturduğu gürültüdür. Bu kapsamda motorların ürettiği elektromanyetik gürültünün tespiti için Hexacopter'de bulunan 6 adet motor pervanesiz çalıştırıldı (Şekil 3.38).



**Şekil 3.38:** Hexacopter ve Altı Adet Pervanesiz Motorları

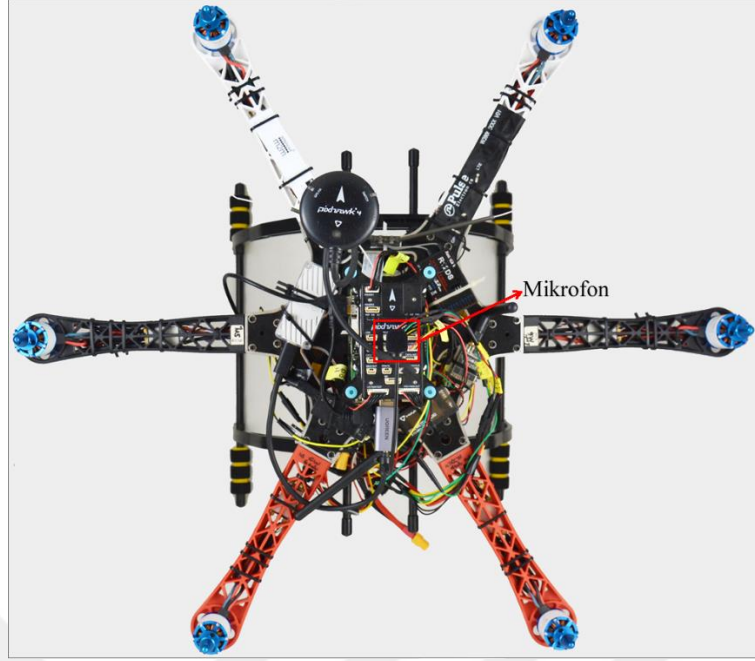
Ayrıca motorların dönme hızına bağlı olarak üretilen ses farklı olacağından hexacopter'in kumandası aracılığıyla hover modundaki (hexacopter'in belirli bir hedef noktada beklediği uçuş modu) ve hover modundaki roll, pitch ve yaw uçuş hareketlerine ait motor sesleri kaydedilerek ses frekans analizleri yapıldı. Ölçüm detayı, Çizelge 3.6'da gösterilmektedir.

**Çizelge 3.6:** İnsansız Hava Aracı Motorlarının Gürültü Ölçüm Detayı

Dış Rotorlu Fırçasız DC Motor Modeli	Yük Taşıma Durumu	Gürültü Kaynağı
SunnySky X2216 880Rpm/V	Yüksüz	Elektromanyetik (Pervane Yok)

### **Ölçüm kurulumu**

Yalnızca elektromanyetik olarak uyarılmış gürültüyü kaydetmek için 6 adet motordan pervaneler çıkarıldı. Motor sesleri, yerdeki Hexacopter'in tam merkezinde, her bir motordan 27,50 cm uzaklıkta yer alan mikروفon tarafından kaydedildi (Şekil 3.39).

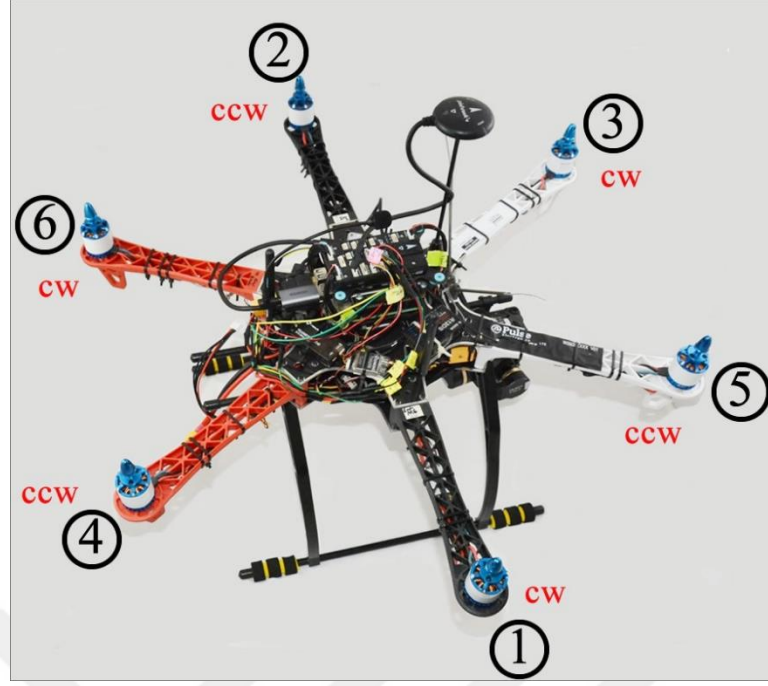


**Şekil 3.39:** Hexacopter'deki Gömülü Sistem Mikrofonu

Motorlar, hover modundaki (bu hexacopter için kumanda gaz kolu orta komunda iken) devir sayısına (rpm) kadar hızlandırıldı. Aerodinamik yük ihmal edildi.

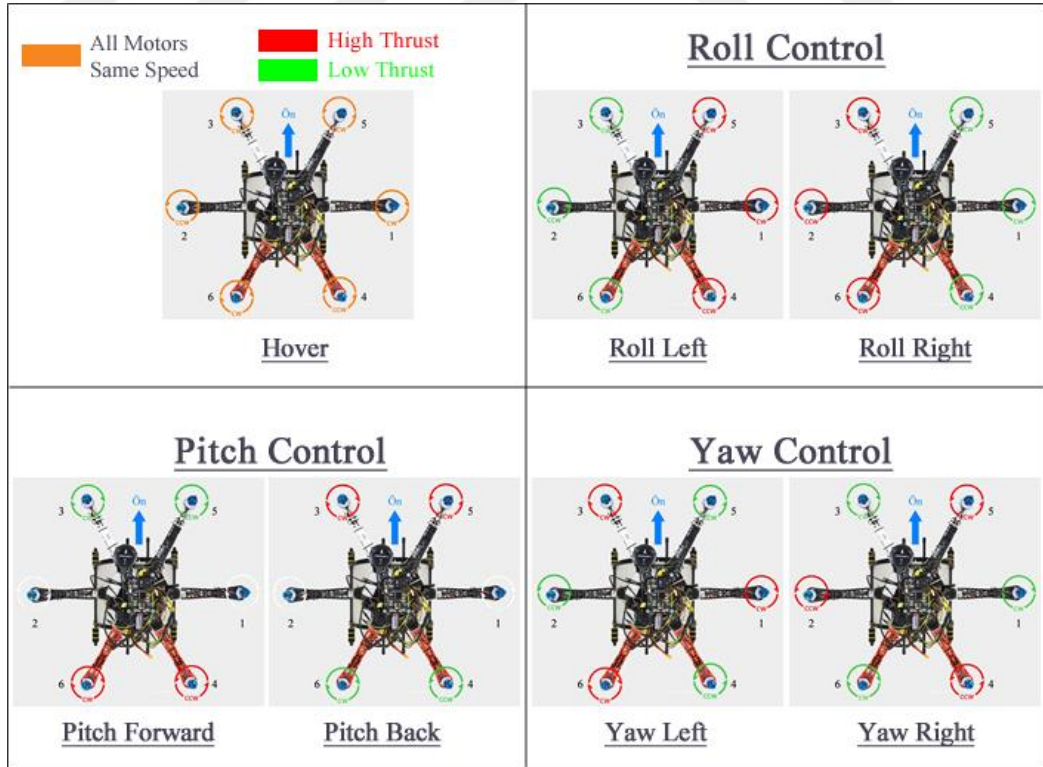
#### **3.5.3.2.1 Hexacopter dış rotorlu fırçasız DC motorlarının (pervanesiz) hover modundaki roll, pitch ve yaw uçuş hareketlerinde ürettiği seslerin spektrogramları**

Hexacopter'in Roll, Pitch ve Yaw eksenlerindeki uçuş hareketlerine göre motor hızları farklılık göstermekte ve buna bağlı olarak 6 adet motor tarafından üretilen ses sinyalleri değişmektedir. Şekil 3.40, Hexacopter'in motor numaralarını ve motorların dönüş yönlerini (cw:clockwise, ccw:counterclockwise) göstermektedir.



Şekil 3.40: Hexacopter'deki Motorların Numaraları ve Dönüş Yönleri

Hexacopter'in roll, pitch ve yaw uçuş hareketlerinde, 6 adet motordan hangilerinin hızlarının arttığı veya azaldığı, Şekil 3.41'deki motor numaralarına göre Çizelge 3.7'de gösterilmektedir.

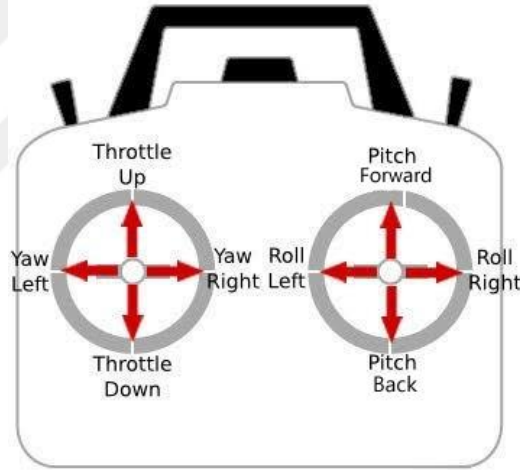


Şekil 3.41: Hexacopter'deki Motorların Numaraları, Dönüş Yönleri ve Dönüş Hızları

**Çizelge 3.7:** İnsansız Hava Aracı Motorlarının Uçuş Hareketlerindeki Hız Değişimleri

	<b>Hızlı Dönen Motorlar</b>	<b>Yavaş Dönen Motorlar</b>
Roll Right	2, 3, 6	1, 4, 5
Roll Left	1, 4, 5	2, 3, 6
Pitch Down	4, 6	3, 5
Pitch Up	3, 5	4, 6
Yaw Right	2, 4, 5	1, 3, 6
Yaw Left	1, 3, 6	2, 4, 5

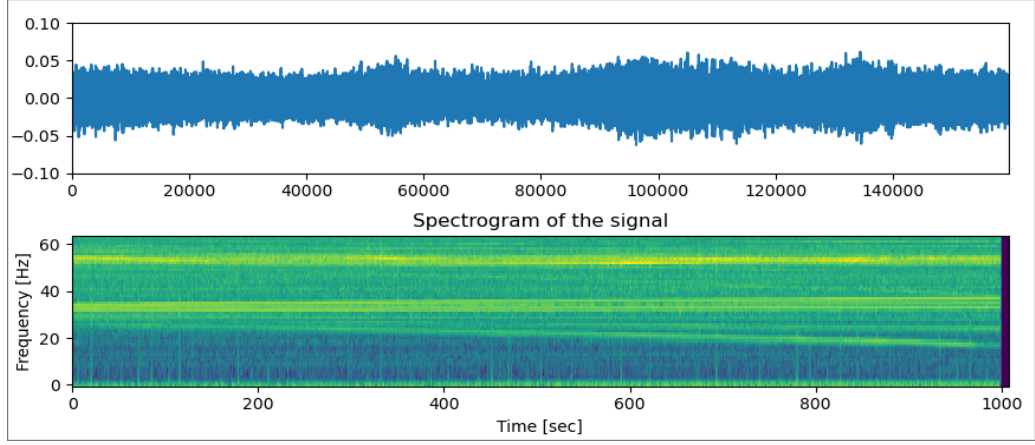
Hexacopter'in hover modundaki (havada asılı konumda, tüm motorlar eşit hızla dönüyor) ve hover modunda iken RC kumandadan gönderilen Roll Left, Roll Right, Pitch Forward (veya Pitch Up), Pitch Back (veya Pitch Down), Yaw Left ve Yaw Right uçuş hareket komutlarına (Şekil 3.42) göre 6 adet motor tarafından üretilen ses sinyallerine ait spektogramlar, aşağıda Şekil 3.43 ile Şekil 3.49 arasında gösterilmektedir.



**Şekil 3.42:** Hexacopter Rc Kumanda Joystick İşlevi

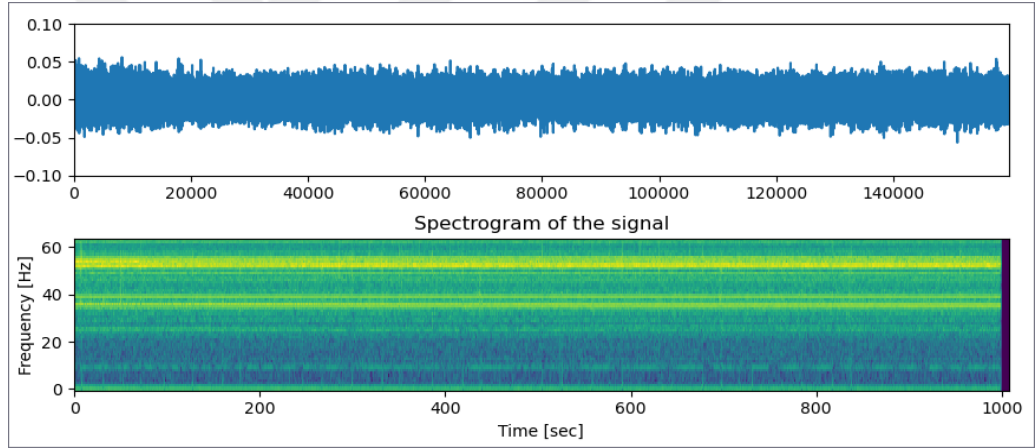
**Kaynak:** (Louros ve diğ., 2018)

Hexacopter, hover modunda uçarken 6 adet dış rotorlu fırçasız DC motor tarafından üretilen gürültüye ait spektrogram Şekil 3.43'de görülmektedir.

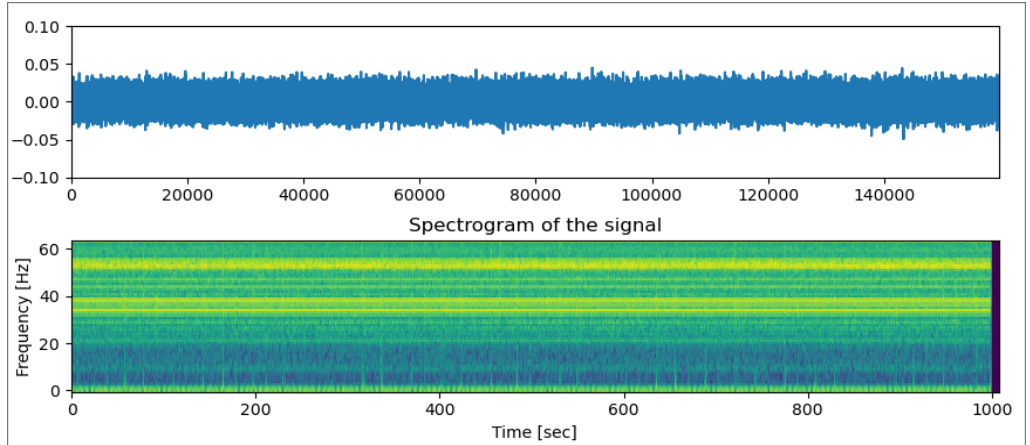


**Şekil 3.43:** Hexacopter Hover Modunda Uçarken Altı Motorun Ürettiği Gürültü Spektrogramı

Hexacopter, hover modunda uçarken kumandadan gönderilen Roll Left ve Roll Right uçuş hareketlerinde 6 adet motor tarafından üretilen gürültüye ait spektrogramlar Şekil 3.44 ve 3.45’de görülmektedir.

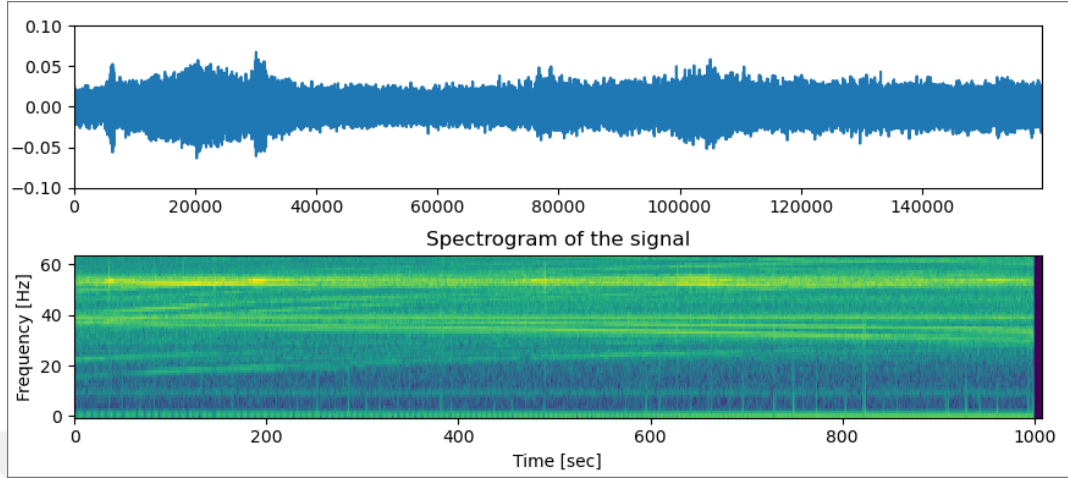


**Şekil 3.44:** Hexacopter Roll Left Hareketindeki Altı Motorun Ürettiği Gürültü Spektrogramı

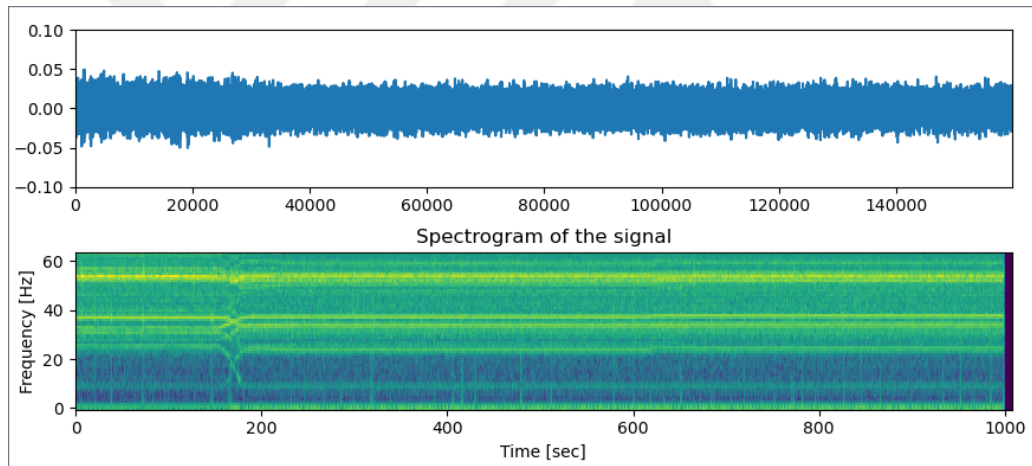


**Şekil 3.45:** Hexacopter Roll Right Hareketindeki Altı Motorun Ürettiği Gürültü Spektrogramı

Hexacofter, hover modunda uçarken kumandadan gönderilen Yaw Left ve Yaw Right uçuş hareketlerinde 6 adet motor tarafından üretilen gürültüye ait spektrogramlar Şekil 3.46 ve 3.47’de görülmektedir.

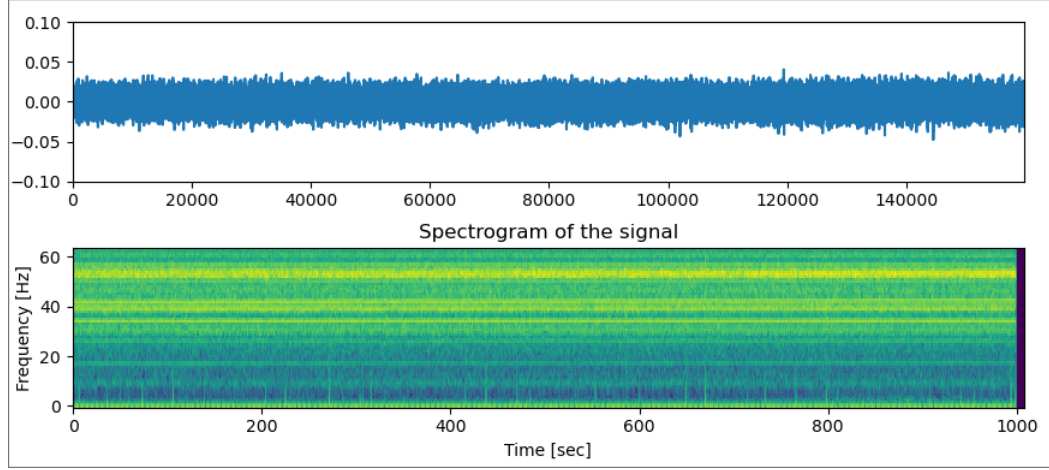


Şekil 3.46: Hexacofter Yaw Left Hareketindeki Altı Motorun Gürültü Spektrogramı

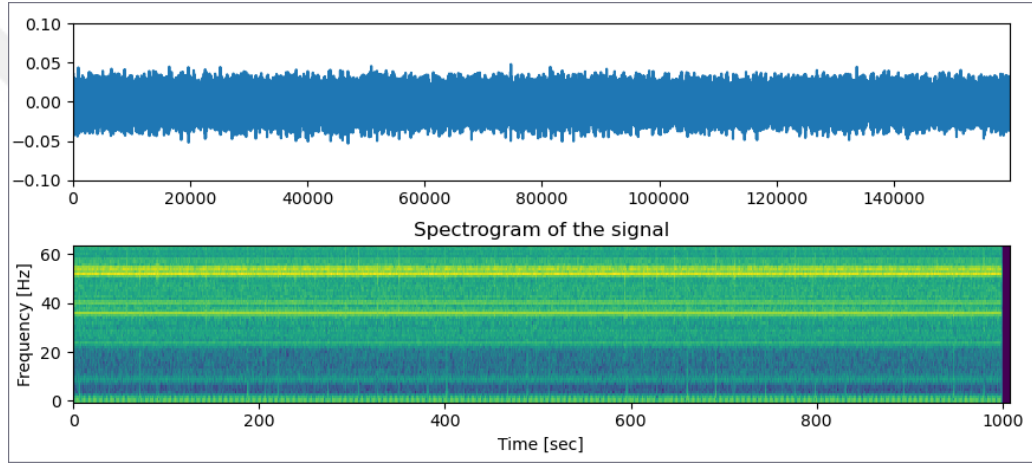


Şekil 3.47: Hexacofter Yaw Right Hareketindeki Altı Motorun Ürettiği Gürültü Spektrogramı

Hexacofter, hover modunda uçarken kumandadan gönderilen Pitch Forward ve Pitch Back uçuş hareketlerinde 6 adet motor tarafından üretilen gürültüye ait spektrogramlar Şekil 3.48 ve 3.49’da görülmektedir.



**Şekil 3.48:** Hexacopter Pitch Forward Hareketindeki Altı Motorun Ürettiği Gürültü Spektrogramı



**Şekil 3.49:** Hexacopter Pitch Back Hareketindeki Altı Motorun Ürettiği Gürültü Spektrogramı

Bu spektrogramlar, Hexacopter'deki dış rotorlu BLDC motorların elektromanyetik olarak uyarılmış gürültüsünü göstermektedir. Bu deneyler İHA'ların (İnsansız Hava Araçları) sesinin bir kısmının elektromanyetik gürültüden kaynaklandığını göstermektedir.

### 3.5.3.1.2 İnsansız hava aracının motor-pervane ikililerinin ürettiği gürültülerin ses analizleri

Literatürde, çok rotorlu İHA'lardaki pervanelerin gürültüsünü incelemek için yapılmış birçok çalışma bulunmaktadır [224-228]. Araştırmacılar tarafından yapılan deneysel akustik ölçümlerde, rotor-gövde etkileşiminin, kanat geçiş frekansı (BPF) harmoniklerinde önemli ton gürültüsü ürettiği tespit edilmiştir [229]. Ayrıca, pervaneler ve çevredeki türbülanslı akışlar arasındaki etkileşim nedeniyle geniş bant gürültüsünün üretildiğini vurgulamışlardır [230]. Hexacopter'deki 6 adet motor-


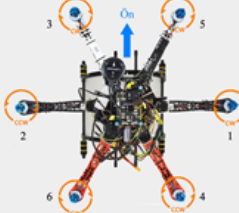
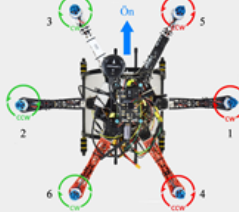
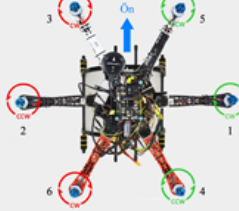
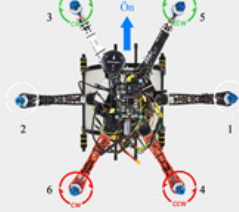
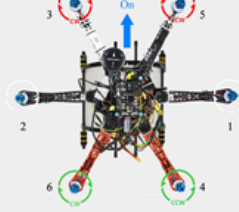
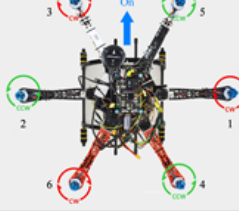
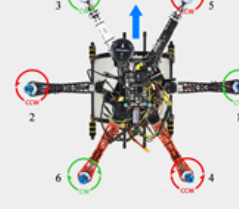
pervane ikilisinin ürettiği ses sinyallerini kaydetmek için Hexacopter pervaneleri Şekil 3.50'deki gibi pervane yönleri doğru olacak şekilde monte edildi. Sonrasında, Hexacopter'in iniş takımlarına yeterli ağırlıklar koyularak çalıştırıldı. Hover uçuş modu ve hover uçuş modundaki roll, pitch ve yaw uçuş hareketlerine göre ses kayıtları alındı.



Şekil 3.50: Hexacopter ve Altı Adet Pervaneli Motoru

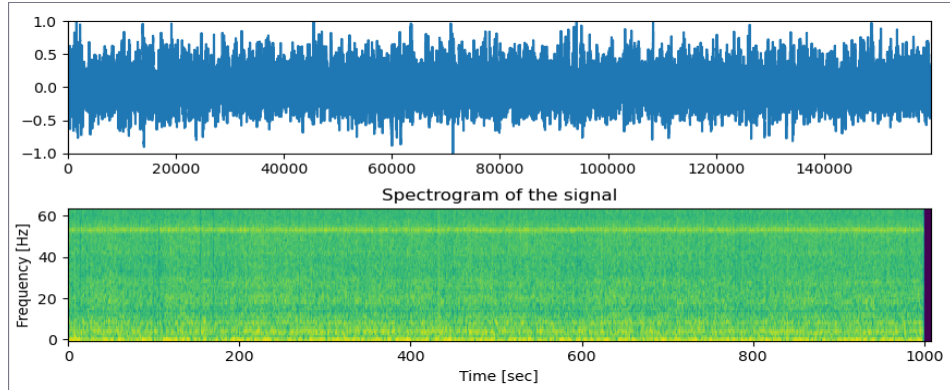
### 3.5.3.1.3 Hexacopter dış rotorlu fırçasız DC motor-pervane ikilisinin hover modundaki roll, pitch ve yaw uçuş hareketlerinde çıkardığı seslerin spektrogramları

Bir önceki bölümde motorların çıkardığı sesler Hexacopter'in merkezindeki mikروفon aracılığıyla kaydedilmiş ve ses frekans analizleri yapılmıştı. Bu bölümde, işlemlerin aynısı motorlara pervaneler takılı iken gerçekleştirildi. Şekil 3.51'de görüldüğü üzere Hexacopter'in hover modunda (havada asılı konumda) ve hover modunda iken RC kumandadan gönderilen Roll Left, Roll Right, Pitch Forward (veya Pitch Up), Pitch Back (veya Pitch Up), Yaw Left ve Yaw Right komutlarına (uçuş hareketlerine) göre 6 adet motor-pervane ikilisi tarafından üretilen ses sinyallerine ait spektrogramlar, aşağıda Şekil 3.52 ile Şekil 3.58.

Movements	
Hover	
Roll Left	
Roll Right	
Pitch Forward	
Pitch Back	
Yaw Left	
Yaw Right	

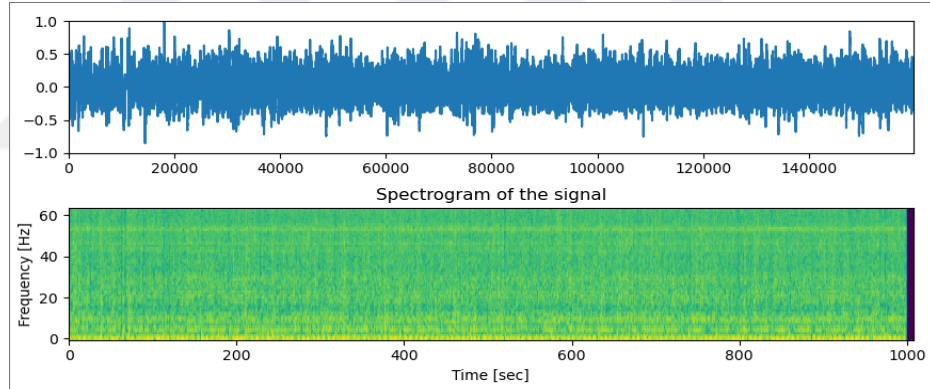
Şekil 3.51: Hexacopter Eksen Hareketleri ve Motor Hızları

Hexacopter, hover modunda uçarken 6 adet motor-pervane ikilisi tarafından üretilen gürültüye ait spektrogram Şekil 3.52’de görülmektedir.

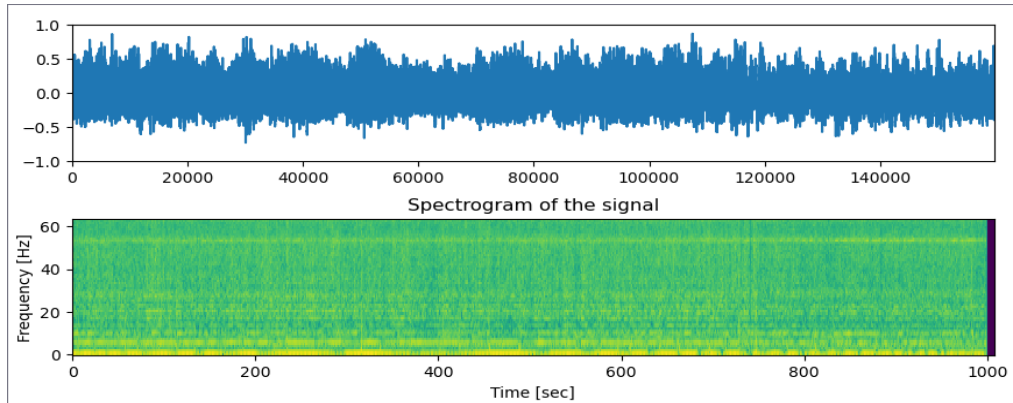


**Şekil 3.52:** Hexacopter Hover Modu Altı adet Motor-Pervane İkilisinin Ürettiği Gürültü Spektrogramı

Hexacopter, hover modunda uçarken kumandadan gönderilen Roll Left ve Roll Right uçuş hareketlerinde 6 adet motor-pervane ikilisi tarafından üretilen gürültüye ait spektrogramlar Şekil 3.53 ve 3.54’da görülmektedir.

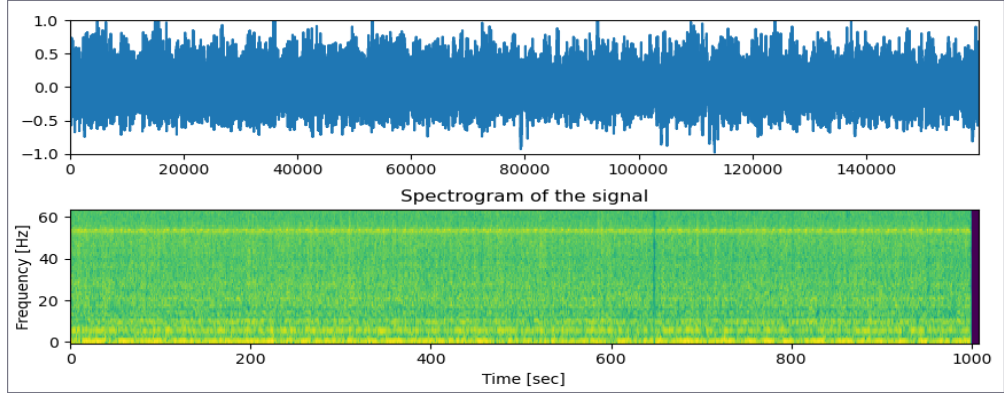


**Şekil 3.53:** Hexacopter Roll Left Hareketindeki Altı adet Motor-Pervane İkilisinin Ürettiği Gürültü Spektrogramı

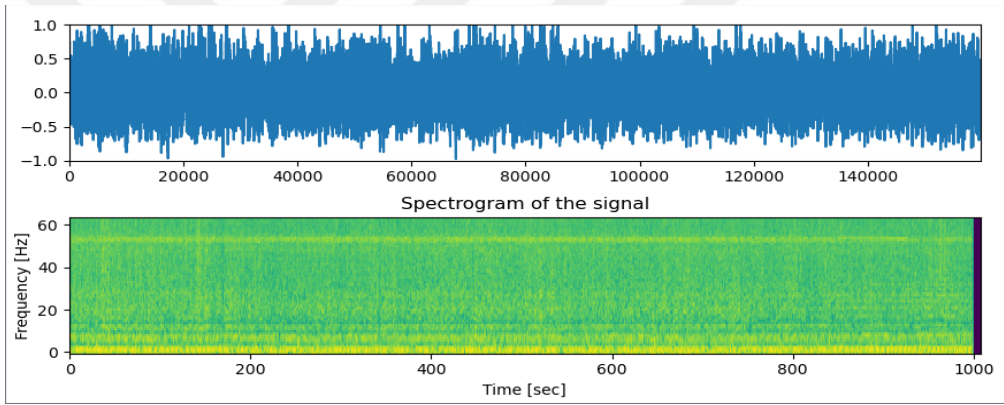


**Şekil 3.54:** Hexacopter Roll Right Hareketindeki Altı adet Motor-Pervane İkilisinin Ürettiği Gürültü Spektrogramı

Hexacopter, hover modunda uçarken kumandadan gönderilen Yaw Left ve Yaw Right uçuş hareketlerinde 6 adet motor-pervane ikilisi tarafından üretilen gürültüye ait spektrogramlar Şekil 3.55 ve 3.56'da görülmektedir.

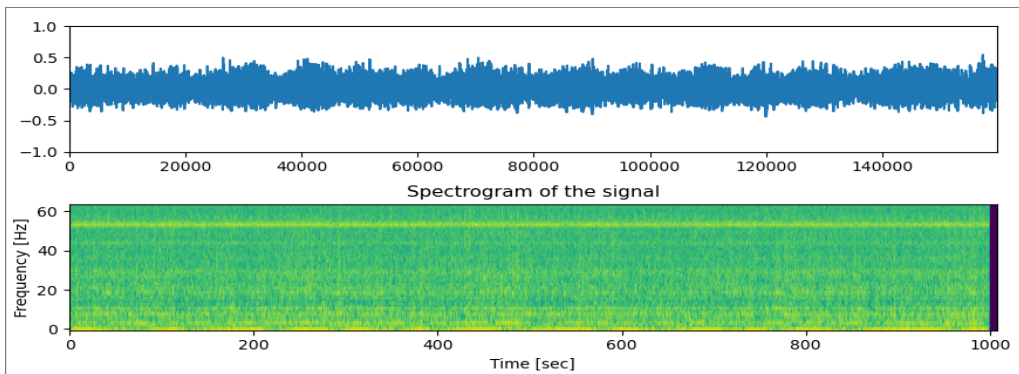


**Şekil 3.55:** Hexacopter Yaw Left Hareketindeki Altı adet Motor-Pervane İkilisinin Gürültü Spektrogramı

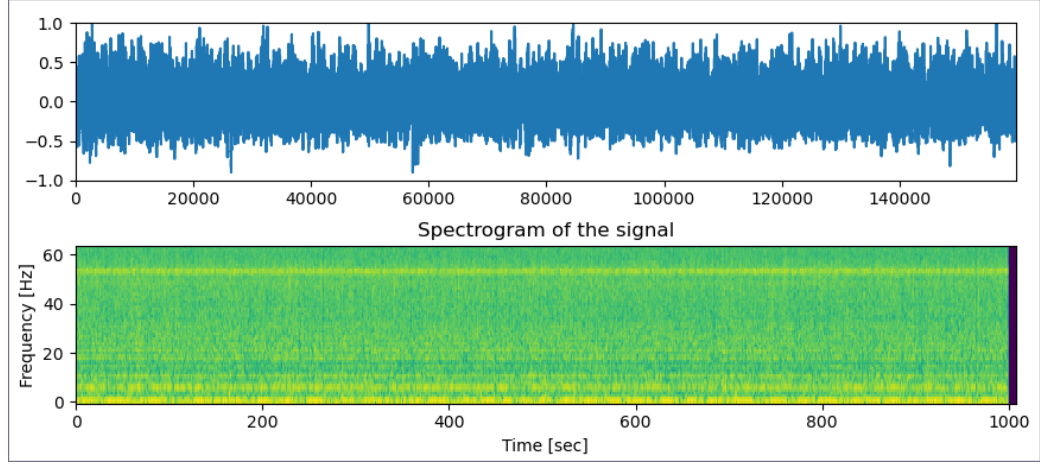


**Şekil 3.56:** Hexacopter Yaw Right Hareketindeki Altı adet Motor-Pervane İkilisinin Gürültü Spektrogramı

Hexacopter, hover modunda uçarken kumandadan gönderilen Pitch Forward ve Pitch Back uçuş hareketlerinde 6 adet motor-pervane ikilisi tarafından üretilen gürültüye ait spektrogramlar Şekil 3.57 ve 3.58'de görülmektedir.



**Şekil 3.57:** Hexacopter Pitch Forward Hareketindeki Altı adet Motor-Pervane İkilisinin Ürettiği Gürültü Spektrogramı



**Şekil 3.58:** Hexacopter Pitch Back Hareketindeki Altı adet Motor-Pervane İkilisinin Ürettiği Gürültü Spektrogramı

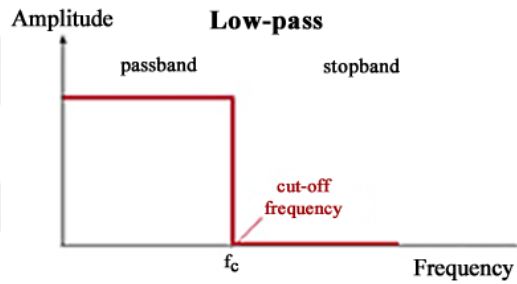
### 3.5.3.2 Ses sınıflandırmasını olumsuz etkileyen gürültüleri filtreleme yöntemi

Çevresel ses sınıflandırmasında, belirli bir nesnenin doğru sınıflandırılabilmesi için ses sinyalindeki gürültülerin ve farklı seslerin filtrenmesi gerekir. Filtreleme süreci, istenmeyen gürültü ve diğer bileşenleri sinyalden uzaklaştırmak ve istenen sinyali elde etmek için kullanılan bir süreçtir. Başka bir deyişle, sinyal işleme alanında filtreler, bir sinyaldeki istenmeyen frekansları ortadan kaldırmak veya istenen frekansları yükseltmek için kullanılan temel araçlardır. Filtreler kullanılarak, orijinal sinyalden istenen genlik, faz ve frekans elde edilebilir. Dijital filtreler, birçok alanda analog filtrelere kıyasla gürültü sinyallerini tespit etme ve filtreleme konusunda daha tercih edilir olduğundan daha yaygın olarak kullanılır [231].

Dijital filtreler, bir sinyaldeki belirli frekansları kaldırmak veya yükseltmek için kullanılan algoritmalarıdır. Dijital filtreler, ses işleme, görüntü işleme ve veri analizi dâhil olmak üzere çeşitli alanlarda kullanılır. Filtrelerin temel işlevi, istenmeyen frekansları ortadan kaldırmak ve istenen frekansları korumaktır. Böylece bir sinyalden gerekli bilginin verimli bir şekilde çıkarılması sağlanır. Sinyal işleme, günümüz iletişim sistemlerinin çok önemli bir yönüdür ve filtreler, sinyal işlemede hayati bir rol oynar. Sinyal işlemede filtrelerin kullanılması, sinyali bozabilecek ve gerekli bilgilerin çıkarılmasını zorlaştırabilecek gürültü ve parazit ortadan kaldırılmasına yardımcı olur. En sık kullanılan filtrelerden biri Butterworth filtresidir. Python-Scipy Kütüphanesi, ses sinyaline dijital Butterworth filtrelerini (`scipy.signal.butter`) uygulama imkânı vermektedir .

**Butterworth Filtre Türleri:** Filtreler, sinyalleri belirli frekans aralıklarına göre işler. En yaygın filtre türleri arasında düşük geçiş, yüksek geçiş, bant geçiren ve bant durdurucu filtreler bulunur. Düşük geçiş filtre, düşük frekanslı sinyallerin geçmesine izin verirken, yüksek geçiş filtre, daha yüksek frekanslı sinyallerin geçmesine izin verir. Bant geçiren filtreler, belirli bir frekans aralığının geçmesini sağlar. Bant durdurucu filtreler, belirli bir frekans aralığındaki sinyalleri ortadan kaldırır.

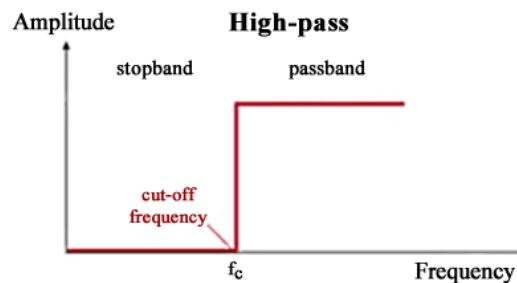
**Düşük Geçiş Butterworth Filtresi (Low-pass Butterworth Filter):** Adından da anlaşılacağı gibi bu filtre, belirli bir kesme frekansından (cut-off frequency) daha düşük frekanslardaki sinyalleri geçirirken, daha yüksek frekanslardaki sinyalleri keser veya zayıflatır (Şekil 3.59). Yani, düşük frekans bileşenlerini korurken yüksek frekansları bastırır.



**Şekil 3.59:** Low-Pass Butterworth Filtresi

**Kaynak:** (/www.open.edu, 2023)

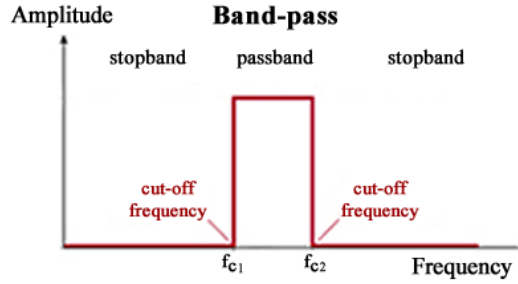
**Yüksek Geçiş Butterworth Filtresi (High-pass Butterworth Filter):** Bu filtre, belirli bir kesme frekansından (cut-off frequency) daha yüksek frekanslardaki sinyalleri geçirirken, düşük frekanslardaki sinyalleri keser (Şekil 3.60). Yani, yüksek frekans bileşenlerini korurken düşük frekansları bastırır.



**Şekil 3.60:** High-Pass Butterworth Filtresi

**Kaynak:** (/www.open.edu, 2023)

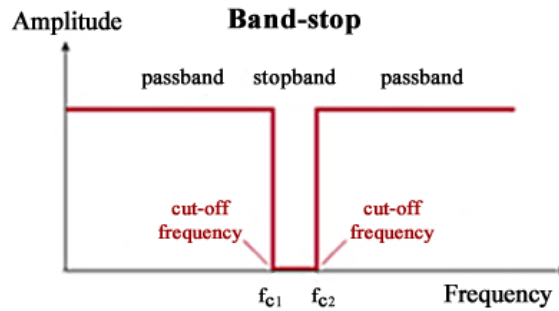
**Bant Geçiren Butterworth Filtresi (Band-pass Butterworth Filter):** Bu filtre, belirli bir alt ve üst kesme frekansı arasındaki banttaki frekansları geçirirken, bu bant dışındaki frekansları bastırır (Şekil 3.61).



**Şekil 3.61:** Band-Pass Butterworth Filtresi

**Kaynak:** (/www.open.edu, 2023)

**Bant Durduran Butterworth Filtresi (Band-stop Butterworth Filter):** Bu filtre, belirli bir alt ve üst kesme frekansı arasındaki banttaki frekansları bastırırken, bu bant içindeki frekansları geçirir (Şekil 3.62).



**Şekil 3.62:** Band-Stop Butterworth Filtresi

**Kaynak:** (/www.open.edu, 2023)

Sonuç olarak, filtreler sinyal işlemede hayati bir rol oynamaktadır ve elektronik, ses işleme, görüntü işleme ve veri analizi gibi çeşitli alanlarda kullanımları yaygındır. Farklı filtre türlerini ve uygulamalarını anlamak, verimli sinyal işleme sistemlerinin tasarlanması ve uygulanmasında önemlidir. Butterworth filtreleri, dijital sinyal işleme, ses işleme ve görüntü işleme dâhil olmak üzere çeşitli uygulamalarda yaygın olarak kullanılmaktadır [233, 234]. Bu çalışmada, dijital Butterworth filtreleri uygulanarak motor-pervane sesleri filtrelendi.

### 3.5.4 YAMNet Ses sınıflandırma modelinin transfer öğrenme yöntemi ile yeniden eğitilmesi ve özelleştirilmesi

TensorFlow-YAMNet, ses olaylarını tanımak için özellik çıkarma katmanına sahip ve MobilNetV1 derin sinir ağı içeren oldukça kapsamlı bir ses sınıflandırıcısıdır [235]. YAMNet, transfer öğrenimi yöntemi ile yeniden eğitilebilen ve sahip olduğu model parametreleri üzerinde değişiklikler yapmamıza izin vererek göreve özel ses sınıflandırıcısının hızlıca oluşturulmasına olanak tanıyan bir modeldir [196]. Bu çalışmada, derin öğrenme tabanlı YAMNet ses sınıflandırma modeli, helikopter nesnesini yüksek doğrulukta sınıflandırabilmesi için transfer öğrenme yöntemi ile yeniden eğitildi ve model parametreleri üzerinden göreve özgü özelleştirildi.

#### 3.5.4.1 Veri setinin hazırlanması

YAMNet modelinin transfer öğrenme yöntemi ile yeniden eğitilmesi sırasında eğitim için Environment Sounds Classifications ESC-50 (Çevresel Ses Sınıflandırması Veri Seti) veri seti kullanıldı. ESC-50 veri seti, 5 ana kategoride 50 anlamsal sınıfa ayrılmış ve 2000 etiketli çevresel ses olaylarından oluşan (örnekten) bir koleksiyondur [236, 237]. ESC-50 veri setindeki 50 adet sınıfın her birinde “.wav” formatında 40’ar adet 5 sn’lik kayıtlar bulunmaktadır. Bu sınıflardan bazıları Şekil 3.63’de görülmektedir.

Animals	Natural soundscapes & water sounds	Human, non-speech sounds	Interior/domestic sounds	Exterior/urban noises
Dog	Rain	Crying baby	Door knock	Helicopter
Rooster	Sea waves	Sneezing	Mouse click	Chainsaw
Pig	Crackling fire	Clapping	Keyboard typing	Siren
Cow	Crickets	Breathing	Door, wood creaks	Car horn
Frog	Chirping birds	Coughing	Can opening	Engine
Cat	Water drops	Footsteps	Washing machine	Train
Hen	Wind	Laughing	Vacuum cleaner	Church bells
Insects (flying)	Pouring water	Brushing teeth	Clock alarm	Airplane
Sheep	Toilet flush	Snoring	Clock tick	Fireworks
Crow	Thunderstorm	Drinking, sipping	Glass breaking	Hand saw

**Şekil 3.63:** ESC-50 Çevresel Ses Sınıflandırması Veri Setinin Sınıfları

**Kaynak:** (/github.com, 2023)

YAMNet ses sınıflandırıcısını göreve özel eğitmek için ESC-50 Çevresel Ses Sınıflandırması Veri Setinin helikopter sınıfına ait 40 ses dosyası kullanıldı (Şekil 3.64).

ESC-50 / meta / esc50.csv															
Preview Code Blame 2001 lines (2001 loc) · 91.5 KB															
1	filename	fold	target	category	esc10	src_file	take		filename	fold	target	category	esc10	src_file	take
57	1-172649-A-40.wav	1	40	helicopter	True	172649	A	1029	3-154926-B-40.wav	3	40	helicopter	True	154926	B
58	1-172649-B-40.wav	1	40	helicopter	True	172649	B	1172	3-68630-A-40.wav	3	40	helicopter	True	68630	A
59	1-172649-C-40.wav	1	40	helicopter	True	172649	C	1173	3-68630-B-40.wav	3	40	helicopter	True	68630	B
60	1-172649-D-40.wav	1	40	helicopter	True	172649	D	1174	3-68630-C-40.wav	3	40	helicopter	True	68630	C
61	1-172649-E-40.wav	1	40	helicopter	True	172649	E	1232	4-125929-A-40.wav	4	40	helicopter	True	125929	A
62	1-172649-F-40.wav	1	40	helicopter	True	172649	F	1311	4-161579-A-40.wav	4	40	helicopter	True	161579	A
74	1-181071-A-40.wav	1	40	helicopter	True	181071	A	1312	4-161579-B-40.wav	4	40	helicopter	True	161579	B
75	1-181071-B-40.wav	1	40	helicopter	True	181071	B	1383	4-175000-A-40.wav	4	40	helicopter	True	175000	A
604	2-188822-A-40.wav	2	40	helicopter	True	188822	A	1384	4-175000-B-40.wav	4	40	helicopter	True	175000	B
605	2-188822-B-40.wav	2	40	helicopter	True	188822	B	1385	4-175000-C-40.wav	4	40	helicopter	True	175000	C
606	2-188822-C-40.wav	2	40	helicopter	True	188822	C	1487	4-193480-A-40.wav	4	40	helicopter	True	193480	A
607	2-188822-D-40.wav	2	40	helicopter	True	188822	D	1488	4-193480-B-40.wav	4	40	helicopter	True	193480	B
631	2-37806-A-40.wav	2	40	helicopter	True	37806	A	1654	5-177957-A-40.wav	5	40	helicopter	True	177957	A
632	2-37806-B-40.wav	2	40	helicopter	True	37806	B	1655	5-177957-B-40.wav	5	40	helicopter	True	177957	B
633	2-37806-C-40.wav	2	40	helicopter	True	37806	C	1656	5-177957-C-40.wav	5	40	helicopter	True	177957	C
634	2-37806-D-40.wav	2	40	helicopter	True	37806	D	1657	5-177957-D-40.wav	5	40	helicopter	True	177957	D
994	3-150979-A-40.wav	3	40	helicopter	True	150979	A	1658	5-177957-E-40.wav	5	40	helicopter	True	177957	E
995	3-150979-B-40.wav	3	40	helicopter	True	150979	B	1700	5-191131-A-40.wav	5	40	helicopter	True	191131	A
996	3-150979-C-40.wav	3	40	helicopter	True	150979	C	1773	5-205898-A-40.wav	5	40	helicopter	True	205898	A
1028	3-154926-A-40.wav	3	40	helicopter	True	154926	A	1854	5-220955-A-40.wav	5	40	helicopter	True	220955	A

Şekil 3.64: ESC-50 Çevresel Ses Sınıflandırması Veri Setinin Helicopter Sınıfı

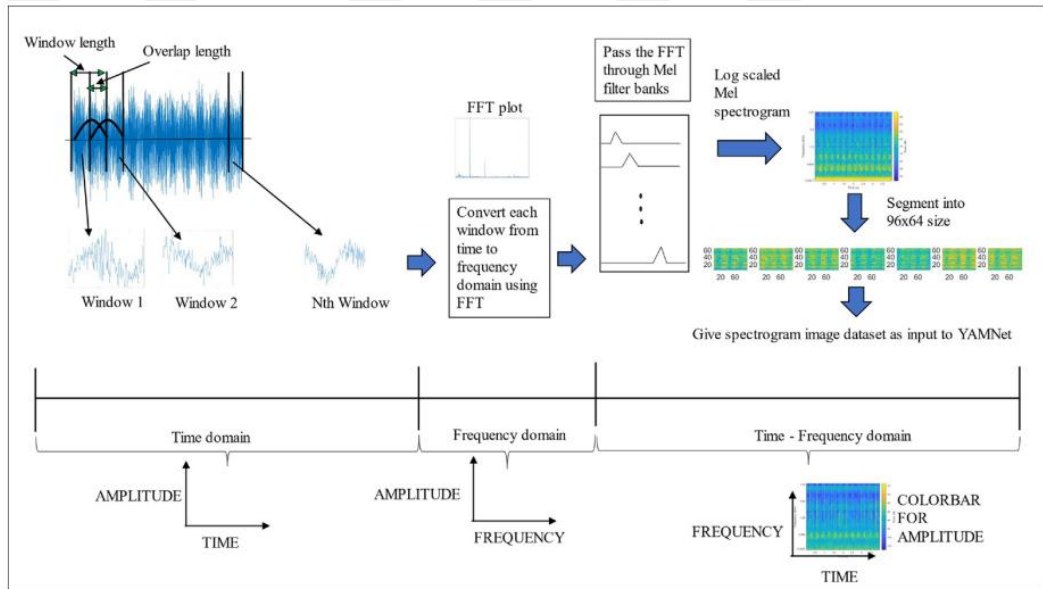
### 3.5.4.2 Otomatik veri ön işleme hazırlığı

Modelin yeniden eğitilmesi ve özelleştirilmesi için model yapısının ve özellik çıkarım sürecinin iyi anlaşılması gerekir. YAMNet modelinde ses sinyalinin girdi olarak kullanılabilmesi için bazı veri ön işleme adımlarına gerekessim vardır. Veri ön işleme, YAMNet mimarisine beslemek için Mel spektrum katsayılarının çıkarılmasını içerir (Şekil 3.65), çünkü bu katsayılar giriş özellik haritasını oluşturur. Veri ön işleme adımları şu şekildedir [201, 238-240]:

1. Ses sinyali, mono sinyal değilse öncelikle mono bir sinyale dönüştürülür ve ardından 16kHz'e (16000Hz) yeniden örneklenir.
2. Bir spektrogram, 25 ms'lik (0.025 sn) bir pencere boyutuna, 10 ms'lik (0.010sn) bir pencere sığmasına ve periyodik bir Hann penceresine sahip

Kısa Zamanlı Fourier Dönüşümü'nün (STFT) büyüklükleri kullanılarak hesaplanır.

3. Bir mel spektrogramı, spektrogramın 125-7500 Hz aralığını kapsayan 64 mel bininin (kutuya) eşlenmesiyle hesaplanır.
4. Logaritmik Mel spektrogramı,  $\log(\text{mel-spektrum} + 0.001)$  uygulanarak hesaplanır. Burada  $\log\_offset$  parametresi kullanılır ve değeri 0.001 alınır. Burada amaç sıfırın logaritmasını almaktan kaçınmaktır.
5. Bu özellikler daha sonra 0,96 saniyelik %50 örtüşen örnekler halinde çerçevlenir; burada her örnek, 64 Mel bandı ve her biri 10 ms'lik 96 çerçeveyi kapsamaktadır. Sonuç olarak, 96x64 boyutunda logaritmik bir Mel Spektrogramı elde edilir ve YAMNet modeline özellik çıkarma için iletilir.



**Şekil 3.65:** YAMNet Modelinin Girdi Gereksinimine Göre Ses Sinyalinin Mel Spektrogramını çıkarma adımları

**Kaynak:** (Patil ve Wani, 2023)

İHA'nın çevresindeki sesleri doğru bir şekilde sınıflandırabilmesi için bir çevresel ses sınıflandırıcı oluşturuldu. YAMNet'ten elde edilen özellikler eğitim sırasında modele girdi olarak verildi. ESC-50 veri seti kullanılarak bir ortam sesi sınıflandırıcısı oluşturuldu; bu sınıflandırıcı, helikopter nesnesini doğru bir şekilde sınıflandırabilir.

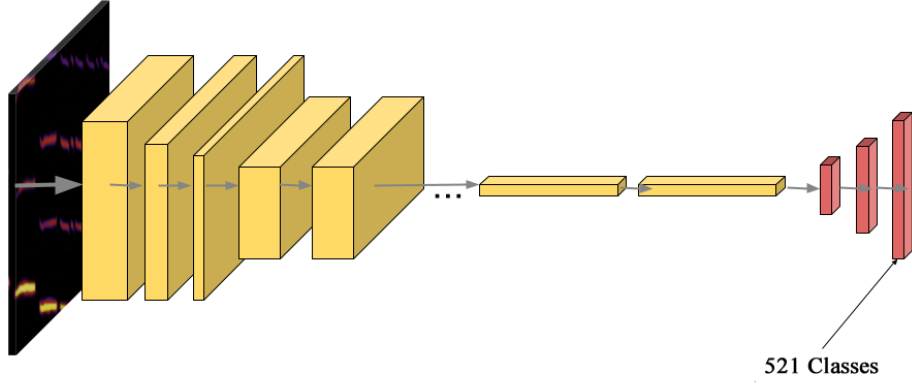
### 3.5.4.3 Modelin transfer öğrenme yöntemi ile yeniden eğitilmesi ve özelleştirilmesi

YAMNet, 521 ses olayını tahmin eden bir ses sınıflandırma modelidir [241]. YAMNet'in helikopter sınıfı bulunmaktadır (Şekil 3.66) ancak genel bir model olduğu için sınıflandırmadaki tahmin doğruluğu özel bir görev için düşük kalabilir.

index	mid	display_name
0	/m/09x0r	Speech
1	/m/0ygtg	Child speech, kid speaking
2	/m/01h8n0	Conversation
3	/m/02qldy	Narration, monologue
4	/m/0261r1	Babbling
5	/m/0brhx	Speech synthesizer
6	/m/07p6fty	Shout
7	/m/07q4ntr	Bellow
8	/m/07rvj3x	Whoop
9	/m/07sr1lc	Yell
10	/t/dd00135	Children shouting
11	/m/03qc9zr	Screaming
12	/m/02rbxlg	Whispering
13	/m/01j3sz	Laughter
14	/t/dd00001	Baby laughter
15	/m/07r660_	Giggle
16	/m/07s04w4	Snicker
17	/m/07sq110	Belly laugh
18	/m/07rgt08	Chuckle, chortle
.....		
318	/m/012n7d	Ambulance (siren)
319	/m/012ndj	Fire engine, fire truck (siren)
320	/m/04_sv	Motorcycle
321	/m/0btp2	Traffic noise, roadway noise
322	/m/06d_3	Rail transport
323	/m/07jdr	Train
324	/m/04zmvq	Train whistle
325	/m/0284vy3	Train horn
326	/m/01g50p	Railroad car, train wagon
327	/t/dd00048	Train wheels squealing
328	/m/0195fx	Subway, metro, underground
329	/m/0k5j	Aircraft
330	/m/014yck	Aircraft engine
331	/m/04229	Jet engine
332	/m/02l6bg	Propeller, airscrew
333	/m/09ct_	Helicopter
334	/m/0cmf2	Fixed-wing aircraft, airplane
335	/m/0199g	Bicycle
336	/m/06_fw	Skateboard

Şekil 3.66: YAMNet Modelinin Bazı Sınıfları

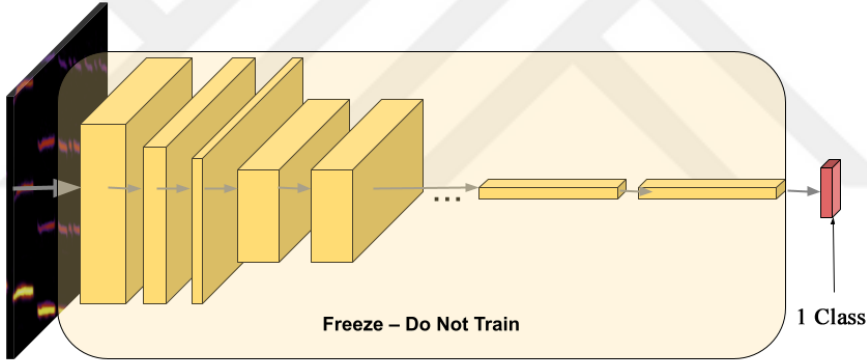
Bu çalışmada, İHA'nın özel görevi helikopter nesnesini, ses verisi üzerinden tespit etmektir. Daha yüksek bir tahmin doğruluğu için YAMNet ses sınıflandırma modelinin ESC-50 veri setinde yer alan helikopter ses verileri ile transfer öğrenme yöntemiyle yeniden eğitilmesi gerekir. YAMNet modeli, bir dizi evrişimli katmandan ve ardından 521 çıktıya göre sınıflandırmayı gerçekleştiren yoğun katmanlardan oluşur (Şekil 3.67).



**Şekil 3.67:** YAMNet Ses Sınıflandırma Modelinin Mimarisi

**Kaynak:** (github.com, 2023)

YAMNet modelinde, önceden öğrenilmiş katmanlarını alabilir ve eğitilemeyecekleri şekilde dondurabiliriz. Ardından eğitmek istediğimiz katmanları eğitebilir ve göreve özel sınıflandırıcıyı (helikopter sınıflandırıcısı) Şekil 3.68'deki gibi ekleyebiliriz.



**Şekil 3.68:** YAMNet Ses Sınıflandırma Modelinin Transfer Öğrenme Yöntemi İle Eğitilmesi

**Kaynak:** (github.com, 2023)

YAMNet ses sınıflandırma modeli, Çizelge 3.8'deki Hiperparametre değerleri ile yeniden eğitildi ve modelin helikopter nesnesini sınıflandırmadaki tahmin doğruluğu artırıldı.

**Çizelge 3.8:** YAMNet Modelinin Hiperparametre Yapılandırma Değerleri

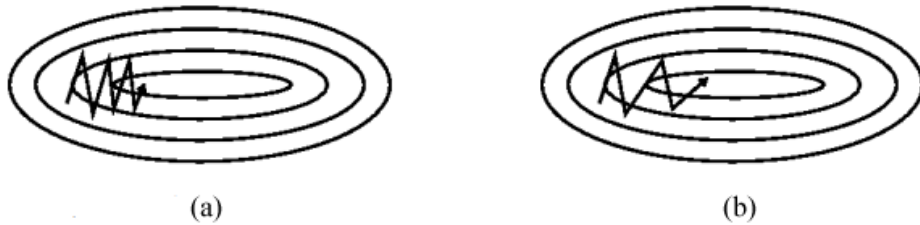
HiperParametre	Değer
Epoch	30
Activation Function	ReLU
Classifier	Softmax
Optimizer	Adam
Batch Size	64

Bölüm 3.4.5.2’de SSD MobilNetV2 sinir ağı modelinin tranfer öğrenme yöntemi ile yeniden eğitilmesi sırasında aktivasyon fonksiyonu ReLu, adım sayısı, batch size ve optimize edici gibi hiperparametreler hakkında detaylı bilgiler verildiğinden bu bölümde tekrar anlatılmamaktadır. Görüntü verileri üzerinden nesne sınıflandırması yapabilen SSD MobilNetV2 modelinin eğitimi sırasında SGDM (SGD+Momentum) optimizier tercih edilirken, ses verileri üzerinden nesne sınıflandırması yapabilen YAMNet modelinin eğitimi sırasında Adam optimizier tercih edildi.

Adaptive Moment Estimation (Adam) optimizier, sinir ağlarını eğitmek için kullanılan popüler bir optimizasyon algoritmasıdır. Adam optimizasyonu, birinci dereceden ve ikinci dereceden momentlerin uyarlanabilir tahminine dayanan stokastik bir gradyan iniş yöntemidir [243] ve SGDM [244] algoritması ile RMSprop [245] algoritmasının bir birleşimidir. Adam optimizasyon algoritmasını anlayabilmek için Momentumlu SGD ve RMSprop optimizasyon algoritmalarına değinmek gerekir.

#### **Momentum Optimizer (Momentum Optimize Edici)**

Stokastik gradyan iniş (SGD) [246], optimum noktaya ilerlerken çok fazla salınım yapar [246], bu da ilerlemeyi yavaşlatır. Momentum, salınımı azaltır, böylelikle yakınsamayı hızlandırır. Şekil 3.69 (a)’da momentumsuz SGD’nin, yerel optimuma doğru giderken çok fazla salınım yaparak ilerlediği görülmektedir. Momentum, Şekil 3.69 (b)’de görüldüğü gibi SGD’nin daha az salınım yapmasını sağlayan ve yerel optimuma daha hızlı ilerlemesine yardımcı olan bir yöntemdir [247].



**Şekil 3.69:** (a) Momentumsuz SGD (b) Momentumlu SGD

**Kaynak:** (Ruder, 2016)

Momentumlu SGD’de, mevcut gradyan yerine  $(\partial L/\partial W_t)$  momentumlu gradyan kullanılmaktadır [246]. Denklem (3.13)’te [246] görüldüğü gibi,

momentumlu gradyan ( $m_t$ ), öğrenme oranı ( $\alpha$ ) ile çarpılarak mevcut ağırlık ( $w_t$ ) güncellenmektedir.

$$w_{t+1} = w_t - \alpha m_t \quad (3.13)$$

Burada  $m_t$  değeri, Denklem (3.14) ile hesaplanmaktadır [246].

$$m_t = \beta m_{t-1} + (1 - \beta) \left[ \frac{\partial L}{\partial w_t} \right] \quad (3.14)$$

Denklem (3.13) ve Denklem (3.14)'te kullanılan parametreler [246, 247]:

$m_t$  = aggregate of gradients at time t [current] (initially,  $m_t = 0$ )

$m_{t-1}$  = aggregate of gradients at time t-1 [previous]

$w_t$  = weights at time t (zaman t'deki ağırlıklar)

$w_{t+1}$  = weights at time t+1 (zaman t+1'deki ağırlıklar)

$\alpha$  = learning rate (öğrenme oranı)

$\partial L / \partial w_t$  = gradient value at time t ("g<sub>t</sub>" in some equations)

$\partial L$  = derivative of Loss Function

$\partial w_t$  = derivative of weights at time t

$\beta$  = moving average parameter (const, 0.9), şeklindedir.

$\beta$ , momentum faktörü, 0 ile 1 arasında bir değer alır ve genellikle  $\beta$  için 0.9 değeri kullanılır [247].

### **Root Mean Square Propagation (RMSP) Optimizer (RMSP Optimize Edici)**

RMSprop, duruma bağlı olarak öğrenme oranını belirleyen AdaGrad algoritmasını geliştirmeye çalışan adaptif bir öğrenme algoritmasıdır [248]. Öğrenme oranları adaptif olduğundan, gerçek öğrenme oranı parametrelere bağlı olarak belirlenir [249]. AdaGrad'da olduğu gibi gradyanların kümülatif toplamını almak yerine, 'üstel ağırlıklı ortalamasını' alır [248, 249]. Ağırlıklar, Denklem (3.15)'deki parametrelere göre güncellenir [246].

$$w_{t+1} = w_t - \frac{\alpha}{(v_t + \epsilon)^{1/2}} \left[ \frac{\partial L}{\partial w_t} \right] \quad (3.15)$$

Denklem (3.15)'deki  $v_t$  değeri, Denklem (3.16)[246] ile hesaplanmaktadır.

$$v_t = \beta v_{t-1} + (1 - \beta) \left[ \frac{\partial L}{\partial w_t} \right]^2 \quad (3.16)$$

Denklem (3.15) ve Denklem (3.16)'te kullanılan parametreler [246, 247]:

$w_t$  = weights at time t (zaman t'deki ağırlıklar)

$w_{t+1}$  = weights at time t+1 (zaman t+1'deki ağırlıklar)

$\alpha$  = learning rate (öğrenme oranı)

$\partial L$  = derivative of Loss Function (kayıp fonksiyonun türevi)

$\partial w_t$  = derivative of weights at time t (zaman t'deki ağırlıkların türevi)

$\partial L / \partial w_t$  = gradient value at time t ("g<sub>t</sub>" symbol in some equations)

$v_t$  = sum of square of past gradients [i.e sum ( $\partial L / \partial w_{t-1}$ )] (initially,  $v_t = 0$ )  
(geçmiş gradyanların karelerinin toplamı)

$\beta$  = moving average parameter (const, 0.9)

$\varepsilon$  = A small positive constant ( $10^{-8}$ ), şeklindedir.

### **Adam Optimizer (Adam Optimize Edici)**

Adam (Adaptive Moment Estimation - Uyarlanabilir Moment Tahmini), momentumlu stokastik gradyan inişinin (SGDM) ve RMSprop'un iyi özelliklerinin birleştirildiği bir optimizasyon algoritmasıdır [248]. Yukarıdaki iki yöntemde yer alan denklemler; Denklem (3.13), Denklem (3.14), Denklem (3.15) ve Denklem (3.16) kullanılarak, Denklem (3.17)[247] ve Denklem (3.18)[247] elde edilir:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (3.17)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \quad (3.18)$$

Denklem (3.17) ve Denklem (3.18)'de kullanılan parametreler [246, 247]:

$m_t$  = gradyanların birinci moment (ortalama) tahmini

$v_t$  = gradyanların ikinci moment (merkezi olmayan varyansı) tahmini

$g_t = \frac{\partial L}{\partial w_t}$  = gradient value at time t (zaman t'deki gradyan değeri)

$\partial L$  = derivative of Loss Function (kayıp fonksiyonun türevi)

$\partial w_t$  = derivative of weights at time t (zaman t'deki ağırlıkların türevi)

$\varepsilon$  = A small positive constant ( $10^{-7}$  or  $10^{-8}$ ) to avoid 'division by 0' error

$\beta_1$  &  $\beta_2$  = decay rates of average of gradients in the above two methods. ( $\beta_1 = 0.9$  &  $\beta_2 = 0.999$ )

$\alpha$  =step size or learning rate (0.001) (step sayısı veya öğrenme oranı), şeklindedir.

Momentlerin beklenti değerleri ile gradyan değerinin birbirine eşit olması gerektiğinden birinci ve ikinci momentlerin ortalama değeri, Denklem (3.19)[247] ve Denklem (3.20)[247] ile hesaplanmaktadır:

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t} \quad (3.19)$$

$$\hat{v}_t = \frac{v_t}{1-\beta_2^t} \quad (3.20)$$

Adam, tüm bu bilgileri kullanarak, RMSprop'ta kullanılan denkleme oldukça benzeyen Denklem (3.21)'i [246, 247] kullanarak ağırlıkları günceller:

$$w_{t+1} = w_t - \alpha \left( \frac{\hat{m}_t}{\sqrt{\hat{v}_t + \varepsilon}} \right) \quad (3.21)$$

burada kullanılan parametreler [246, 247],

$w_t$  = weights at time t

$w_{t+1}$  = weights at time t+1

$\alpha$  =step size or learning rate (0.001)

$\varepsilon$  = A small positive constant ( $10^{-7}$  or  $10^{-8}$ ) to avoid 'division by 0' error, şeklindedir.

Adam optimizasyonu, hesaplama açısından verimli olması, daha az belleğe gereksinim duyması ve büyük miktarda veri veya parametrelere sahip problemler için uygun olmasıyla bilinir. Adam optimizasyonu, hareketli hedeflere ve oldukça gürültülü veya seyrek gradyanlara sahip problemlerde etkilidir [250]. Şekil 3.70, YAMNet modelinin eğitimi sırasında Adam algoritmasını uygulayan optimize edicidir.

```

tf.keras.optimizers.Adam(
    learning_rate=0.001,
    beta_1=0.9,
    beta_2=0.999,
    epsilon=1e-07,
    amsgrad=False,
    weight_decay=None,
    clipnorm=None,
    clipvalue=None,
    global_clipnorm=None,
    use_ema=False,
    ema_momentum=0.99,
    ema_overwrite_frequency=None,
    jit_compile=True,
    name='Adam',
    **kwargs
)

```

**Şekil 3.70:** Adam Algoritmasını Uygulayan Optimizer ve Argümanlar

**Kaynak:** (www.tensorflow.org, 2023)

TensorFlow ile Şekil 3.71’de görünen Python kodu ile Compile edilir:

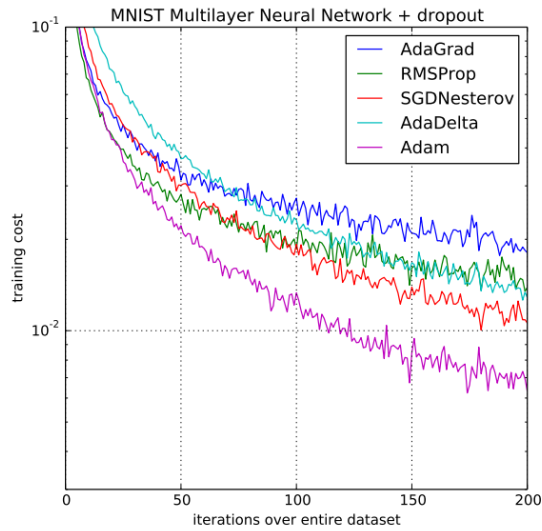
```

my_model.compile(loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
                 optimizer="adam",
                 metrics=['accuracy'])

```

**Şekil 3.71:** Adam Optimizer Compile Etme

Adam optimize ediciyi tasarlayan bilim insanları, büyük modeller ve veri kümeleri kullanarak Adam’ın pratik derin öğrenme problemlerini verimli bir şekilde çözebildiğini göstermişlerdir (Şekil 3.72) [250].

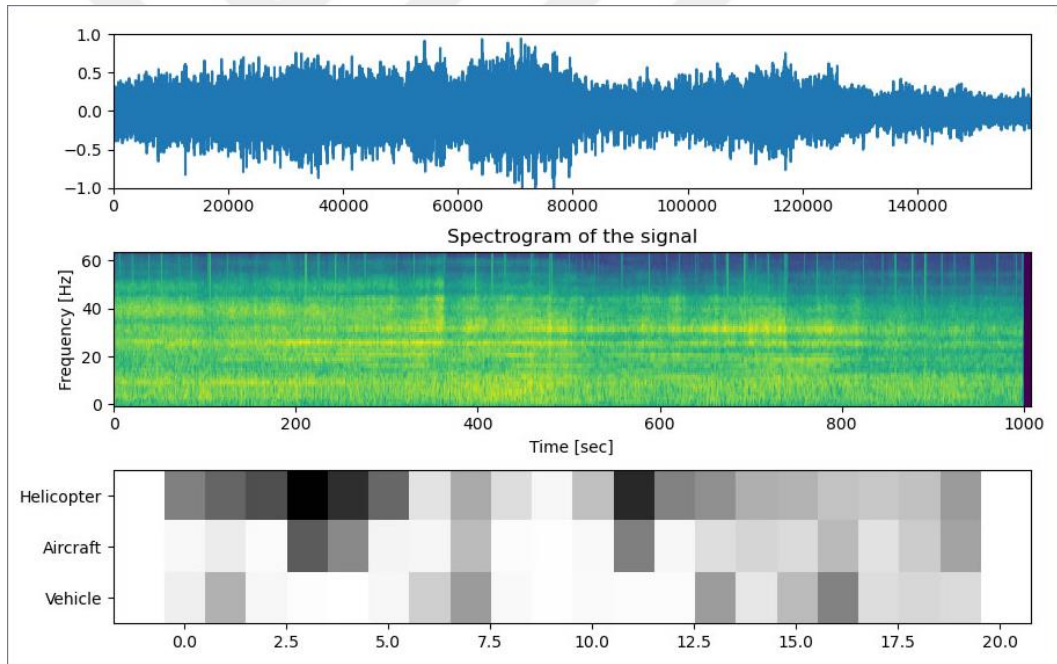


**Şekil 3.72:** MNIST Görüntülerinde Çok Katmanlı Sinir Ağlarının Eğitimi

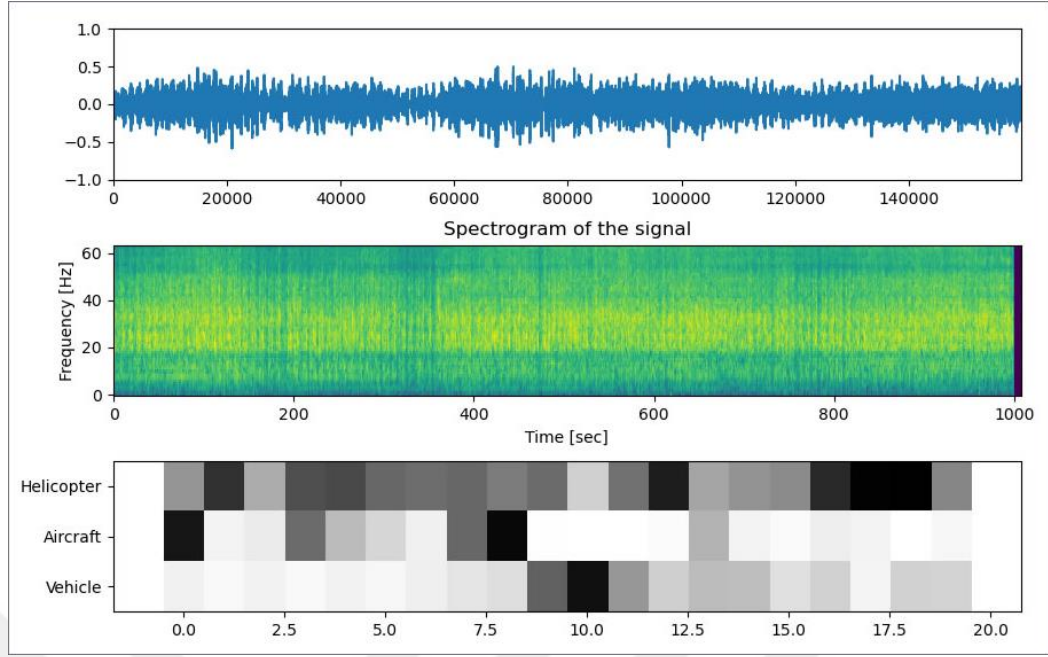
**Kaynak:** (Kingma, 2014)

Bu sebeple, Adam optimizasyon algoritması, adaptif öğrenme oranları ve çeşitli senaryolarda etkinliği nedeniyle derin öğrenme modellerini eğitmek için yaygın olarak kullanılır. Doğru optimizasyon algoritmasını belirlemek, belirli bir problem ve veri kümesine bağlıdır. Belirli bir görev için hangi optimizasyon algoritmasının en iyi performansı gösterdiğini belirlemek için deneyler yapmak gerekebilir. YAMNet modeli, belirli bir görevde kullanıldığı için hiperparametre ayarları tüm bu detaylar dikkate alınarak belirlendi ve uygulandı.

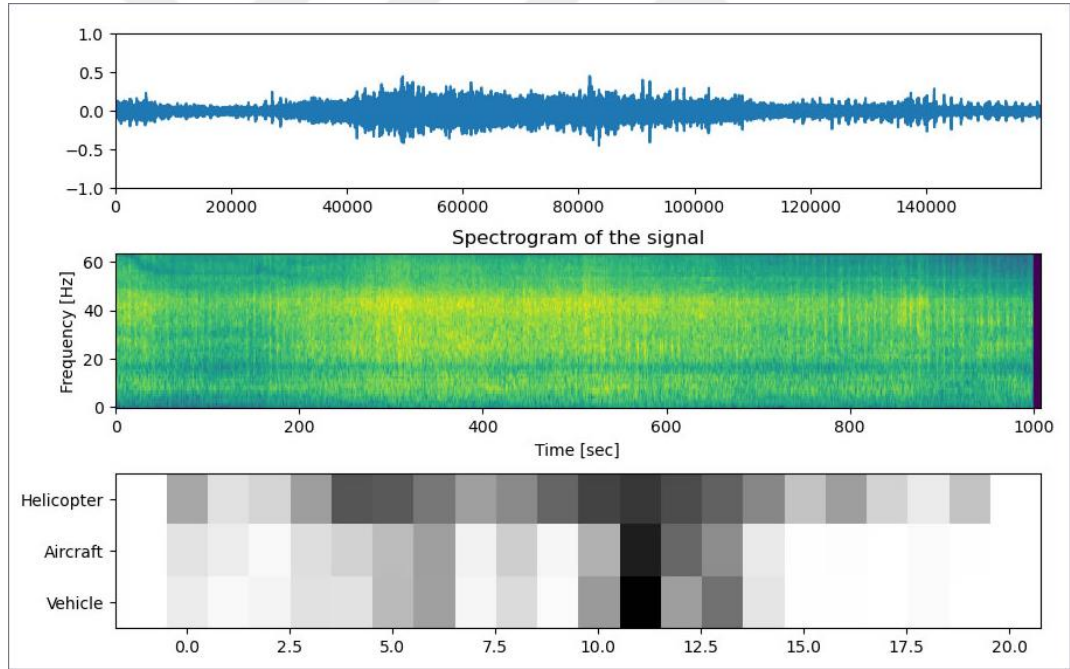
YAMNet ses sınıflandırma modeli, belirli bir görev için özelleştirildikten sonra transfer öğrenme yöntemi ile yeniden eğitildi. Ardından, Raspberry Pi'de kullanılacak şekilde TensorFlow Lite modeline dönüştürüldü. Daha sonra model, test verileri ile test edildi. Aşağıda, Şekil 3.73, Şekil 3.74, Şekil 3.75, Şekil 3.76, Şekil 3.77 ve Şekil 3.78'deki görüntüler YAMNet ses sınıflandırma modelinin eğitim sonrası test sonuçlarına ait çıktılarıdır.



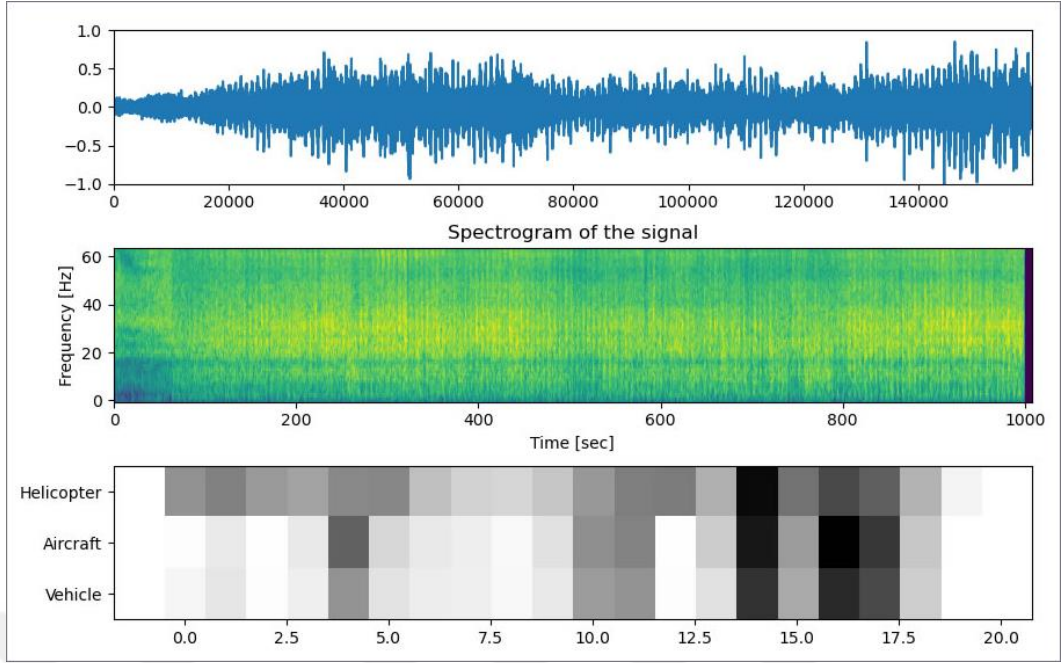
**Şekil 3.73:** YAMNet Ses Sınıflandırma Modelinin Test Sonucu-1



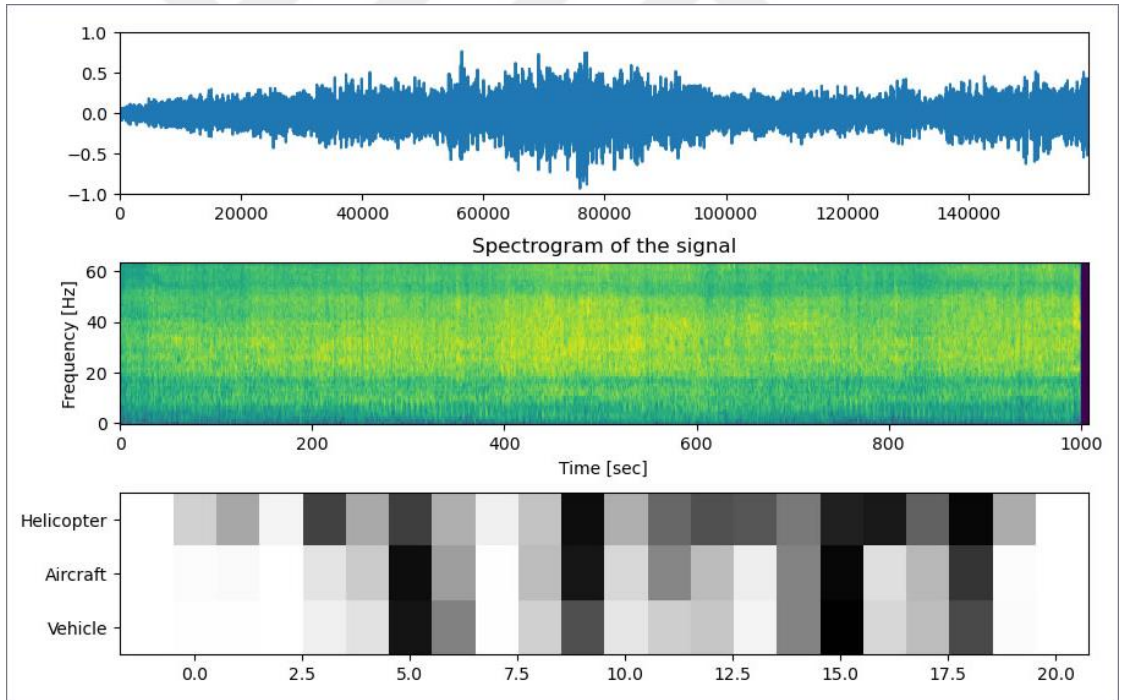
Şekil 3.74: YAMNet Ses Sınıflandırma Modelinin Test Sonucu-2



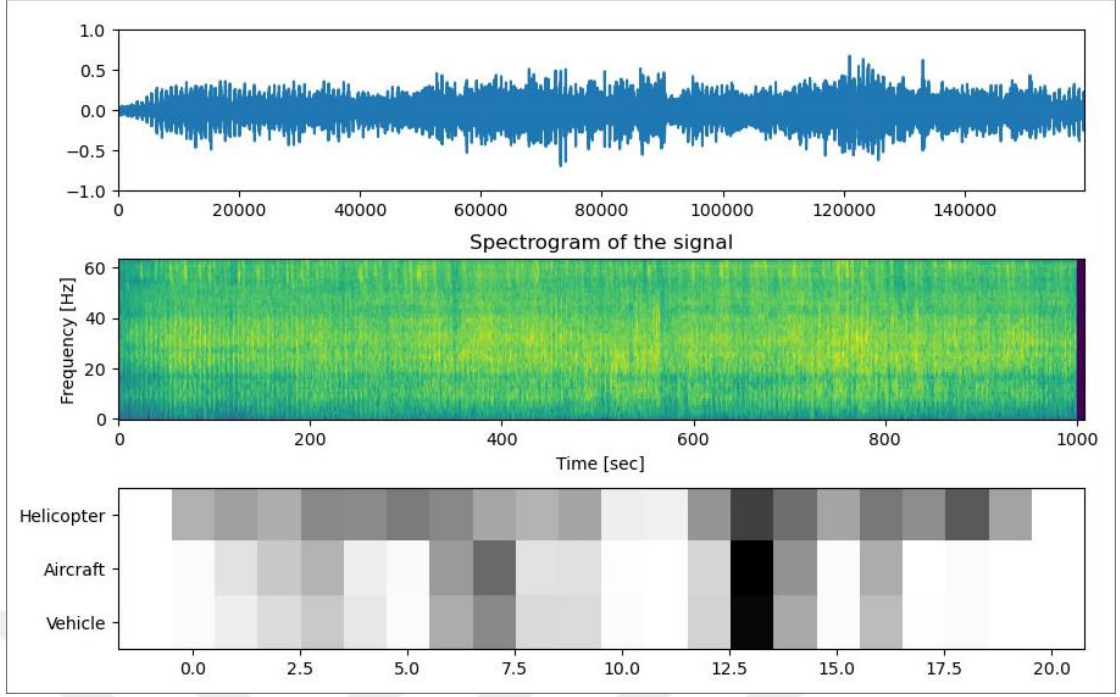
Şekil 3.75: YAMNet Ses Sınıflandırma Modelinin Test Sonucu-3



Şekil 3.76: YAMNet Ses Sınıflandırma Modelinin Test Sonucu-4



Şekil 3.77: YAMNet Ses Sınıflandırma Modelinin Test Sonucu-5



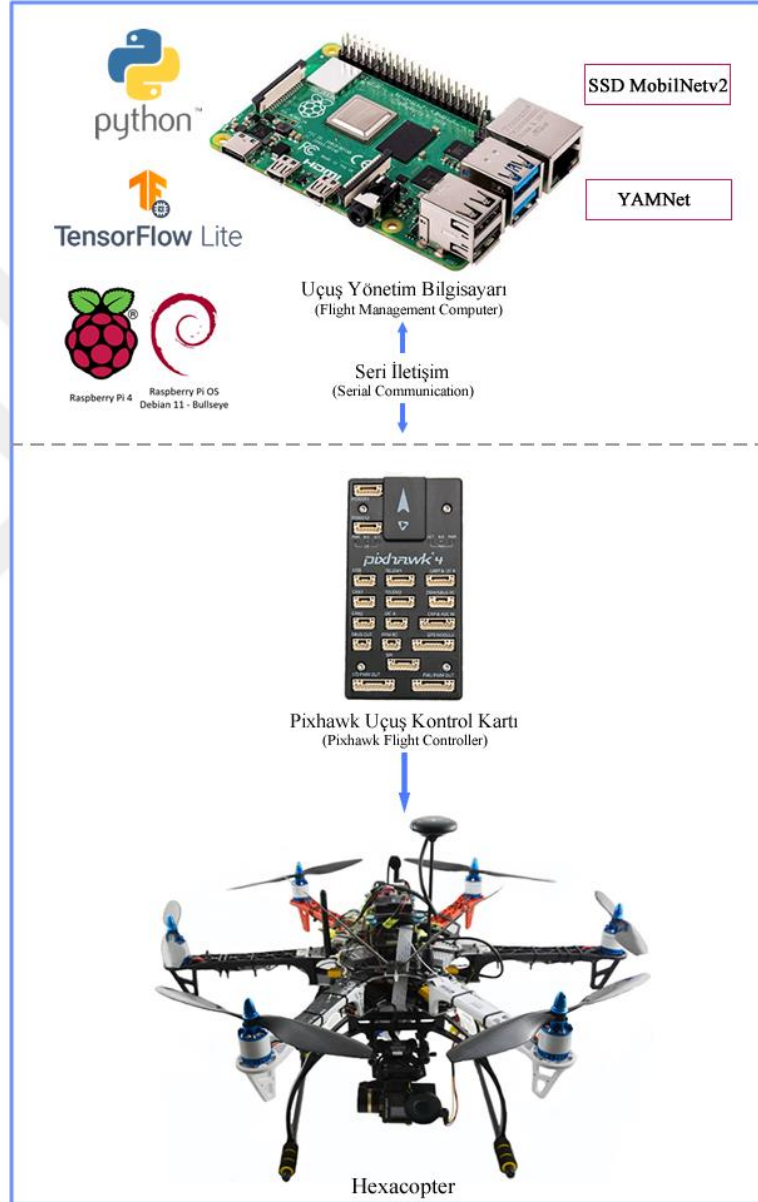
**Şekil 3.78:** YAMNet Ses Sınıflandırma Modelinin Test Sonucu-6

### 3.6 İnsansız Hava Aracı Platformu Gömülü Sistem Tasarımı ve Yazılımı

İHA ile görüntü tabanlı nesne tespit ve sınıflandırma uygulamaları, daha çok ticari olarak satılan hazır İHA'lar üzerinde gerekli modifikasyonlar sağlanarak yapılmaktadır [21, 252, 253]. Bu sistemlerde, donanım ve yazılım müdahalesi kısıtlıdır. Bu nedenle, bu çalışmada kullanılan İHA platformu açık kaynak kodlu yazılımlara sahip donanımlar ile sıfırdan tasarlandı. Bu bölümde, bilgisayarlı görü yöntemlerinden nesne tespit ve sınıflandırma görevlerini yapabilen, Edge TPU yardımcı işlemcisi ile uyumlu çalışan, SSD MobilNETV2 derin öğrenme modelinin ve ses verisi üzerinden nesne tespit ve sınıflandırma görevlerini yapabilen YAMNet derin öğrenme modelinin kullanıldığı gömülü sistem mimarisi, yazılımı ve otonom karar verme süreci detaylandırılmaktadır.

İHA ile nesne tespit ve sınıflandırma ve otonom karar verme süreci üç aşamadan meydana gelmektedir. İlk aşama, görüntü ve ses verilerinin elde edilmesidir. Görüntü ve ses işleme süreçleri senkronize çalışmaktadır. Uçuş sırasında, Raspberry Pi kamerası tarafından görüntü verileri elde edilirken, mikrofon tarafından da ses verileri elde edilmektedir. İkinci aşama, görüntü ve ses verilerinin analiz edilmesidir. Görüntü ve ses verileri gömülü yapay zekâ sistemi tarafından gerçek zamanlı olarak işlenir. Son olarak, görüntü ve ses işleme sonucu elde edilen

veriler (analiz sonuçları) ile sistemde belirli görevler için daha önceden girilmiş verilerin örtüşmesiyle otonom karar verme süreci başlar. Tüm süreç saniyeler içinde gerçekleştirilir ve bu da anında görev yürütmeye sonuçlanır. Sürecin can alıcı kısımları ikinci ve üçüncü aşamadır. Sistemin çevredeki nesnelere gerçek zamanlı olarak algılaması ve sınıflandırması, sonrasında ise tanımlı görevi gerçekleştirecek komutları uçuş kontrol kartına iletmesi gerekmektedir (Şekil 3.79).



**Şekil 3.79:** Önerilen Gömülü Sistem İHA

Bu sistemde, görüntü verisi üzerinden belirli bir nesneyi tespit etmesi ve sınıflandırması için transfer öğrenme yöntemi ile yeniden eğitilen ve özelleştirilen, derin öğrenme tabanlı SSD MobileNetV2 evrişimsel sinir ağı mimarisi, bir görüntüde

veya karede bulunan birden fazla nesneyi algılayabilir. Model, her bir nesnenin konumunu tespit eder ve sınırlayıcı kutu ile nesnenin adını çıktı olarak verir.

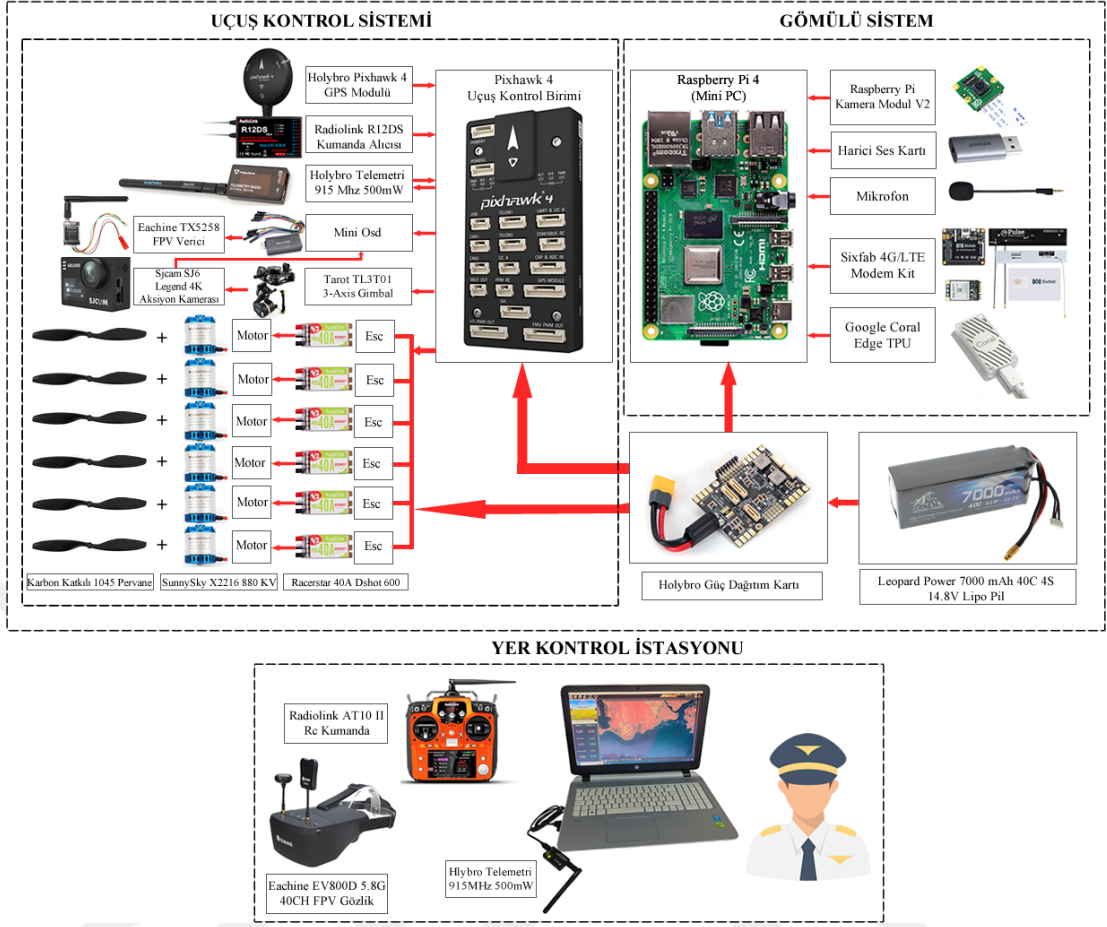
MS COCO veri kümesi üzerinde önceden eğitilmiş SSD MobileNetV2 derin sinir ağı 90 farklı nesneyi sınıflandırabilme kapasitesine sahiptir. Bu sinir ağı modeli, transfer öğrenme yöntemi ile yeniden eğitildikten ve özelleştirildikten sonra kullanılmak üzere Raspberry Pi'ye yüklendi. Yine bu sistemde, ses verisi üzerinden belirli bir nesneyi tespit etmesi ve sınıflandırması için transfer öğrenme yöntemi ile yeniden eğitilen ve özelleştirilen, derin öğrenme tabanlı YAMNet modeli, sesin ait olduğu nesneyi sınıflandırabilme kapasitesine sahiptir. Bu derin öğrenme modeli de kullanılmak üzere Raspberry Pi'ye yüklendi.

İHA'nın nesne algılama ve sınıflandırma sonucu tanımlı görevleri yapabilmesi için birçok Python betiği kodlandı. Betiklerin, İHA ile sekronize çalışabilmesi için tüm sisteme güç verildiğinde İHA ile birlikte çalışmaya başlayan Raspberry Pi 4'ün açılışında otomatik olarak başlaması gerekmektedir. Bunun gerçekleşmesi için, betiklerin çalışacağı sanal ortamı oluşturan ve betikleri başlatan bir bash betiği oluşturuldu. Raspberry Pi 4, kendisine güç geldiğinde ve açıldığında bash betiği çalışacak şekilde yapılandırıldı. Python programlama dili kullanılarak yazılan betikler, derin öğrenme tekniklerini içerir. Görüntü ve ses verileri, önceden eğitilmiş modeller veya özel modeller aracılığıyla analiz edilir.

### **3.6.1 İnsansız Hava Aracı Platformu Gömülü Sistem Donanımları**

İHA'ya kazandırılan gömülü sistem donanım teknolojilerinin birbiriyle uyumlu çalışması ve aralarındaki haberleşmenin sorunsuz olması önem arz etmektedir. Bu sebeple, gömülü sistem tasarımında Linux işletim sistemine sahip ve Python programlama dili ile yazılan yazılım betiklerinin çalıştırılmasına olanak tanıyan donanımlar tercih edildi. Ses ve görüntü işleme süreçlerinin İHA'da yerleşik olarak gerçekleştirilmesi için Şekil 3.80'de gösterildiği gibi gömülü bir sisteme ihtiyaç vardır. Sistem için gerekli olan temel donanımlar aşağıda anlatılmaktadır.

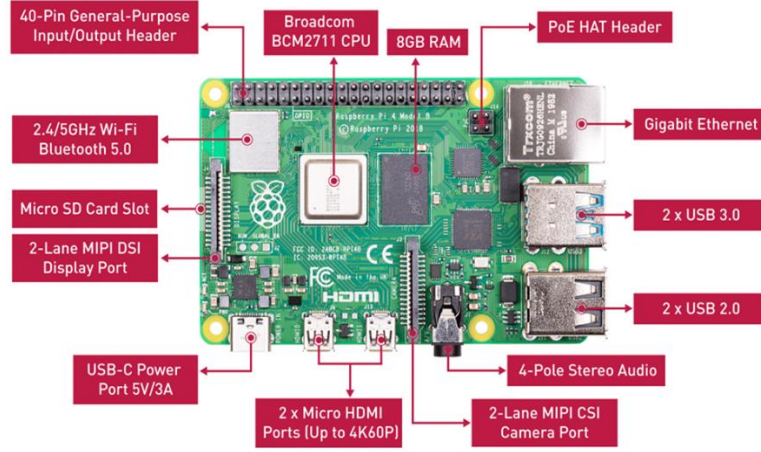
## İNSANSIZ HAVA ARACI PLATFORMU



Şekil 3.80: İnsansız Hava Aracı Platformu ve Yer Kontrol İstasyonu

### 3.6.1.1 Raspberry Pi 4 model B

Bu çalışmada, Raspberry Pi 4 Model B mini bilgisayar kullanıldı (Şekil 3.81). Raspberry Pi, Virtual Network Computing (VNC) olarak adlandırılan sunucu tarafından internet üzerinden gönderilen komutları alan wi-fi özellikli bir mikroişlemcidir. İHA üzerinde yer alan Raspberry Pi mikroişlemcisini uzaktan kontrol etmek ve anlık bilgi akışı sağlamak için Raspberry Pi için özel geliştirilmiş 4G/LTE internet alt yapısı kullanan bir modül entegre edildi. Böylelikle sağlıklı bir internet bağlantısıyla Raspberry Pi cihazına kolaylıkla erişim imkânı ve anlık iletişim yöntemi sağlandı. Raspberry Pi kartına SSD MobileNetV2 ve YAMNet modelleri için TensorFlow Kütüphanesi yüklendi.



**Şekil 3.81:** Raspberry Pi 4 Model B

**Kaynak:** (www.seeedstudio.com, 2023)

### 3.6.1.2 Raspberry Pi kamera modülü V2

Raspberry Pi Kamera Modülü V2, Raspberry Pi Vakfı tarafından piyasaya sürülen, 8 megapiksel Sony IMX219 görüntü sensörüne sahip ve üzerinde sabit odaklı bir lens içeren, özel tasarlanmış bir kamera modülüdür (Şekil 3.82).



**Şekil 3.82:** Raspberry Pi Kamera Modülü V2

**Kaynak:** (makotoltd.com, 2023)

Kamera, 3280 x 2464 piksel statik görüntüler çekebilme yeteneğine sahiptir ve ayrıca 1080p30 (1080 lines of vertical resolution / 1080 horizontal scan line, 30 progressive frames per second), 720p60 ve 640x480p90 videoyu destekler. Raspberry Pi üzerindeki CSI portuna bir şerit kablosu aracılığıyla bağlanır (Şekil 6.4). Picamera Python Kütüphanesi, Raspberry Pi kamera modülüne bir Python arayüzü sağlamaktadır. Bu çalışmada, “Picamera” kütüphanesi, görüntü ile nesne tespit işlemi için ilgili Python betiğinde kullanıldı ve kameraya erişim sağlandı.

### 3.6.1.3 Sixfab 4G/LTE hücresel modem kiti

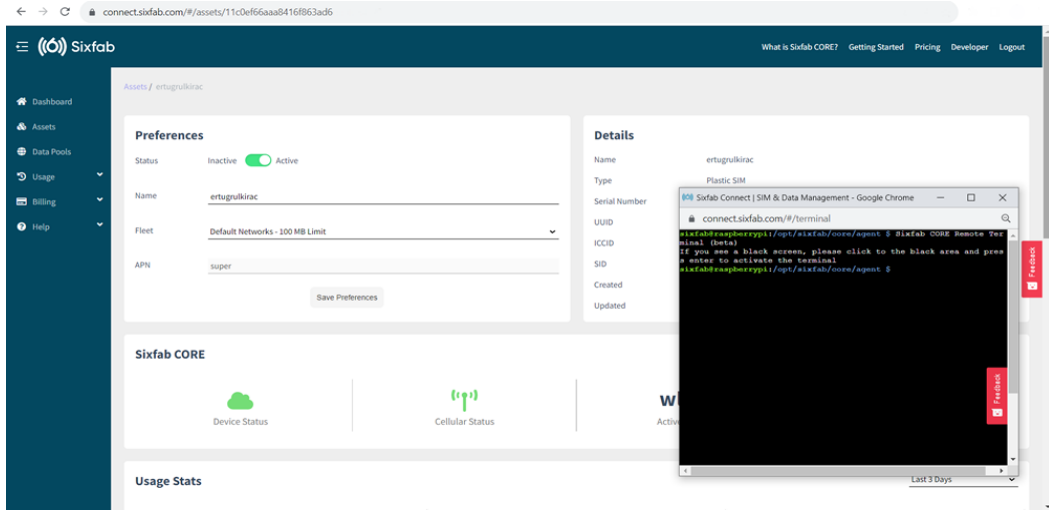
Sixfab 4G/LTE hücresel modem kiti, Raspberry Pi tabanlı IoT projelerine mobil LTE bağlantısı almak için tasarlanmıştır ve Raspberry Pi 4 ile uyumludur (Şekil 3.83).



Şekil 3. 83: Sixfab 4G/LTE Modem Kit

Kaynak: (sixfab.com, 2023)

Bu kit, İHA'nın sağlıklı ve istikrarlı bir mobil ağ bağlantısına sahip olması için platformunun bir parçası olarak kullanıldı. Böylece, telemetri ve wifi gibi mesafe sınırlı veri aktarım bağlantıları yerine her mesafede yer kontrol merkezine veri akışı sağlayacak bir mobil ağ alt yapısı oluşturuldu. Asıl önemli nokta ise sixfab connection ile web ara yüzü üzerinden Raspberry Pi'ye (device) bağlanıp terminal penceresi açmayı sağlayarak Raspberry Pi'de komut yazmamıza ve dolayısıyla İHA'yı kontrol etmemize olanak sağlamaktadır (Şekil 3.84).



Şekil 3.84: Sixfab 4G/LTE Modem Web Arayüzü

Kaynak: (connect.sixfab.com, 2023)

### 3.6.1.4 Coral edge TPU

Coral Edge TPU, yüksek performanslı makine öğrenimi çıkarımı sağlayan ve TensorFlow Lite modellerine güç tasarrufu ve işlem hızı kazandıran özel entegre bir devredir (Şekil 3.85). Birincil nesil Edge TPU, evrimsel sinir ağları gibi derin sinir ağlarını yürütme yeteneğine sahiptir. TensorFlow Lite'ı destekler, bu sebeple sıfırdan modeller oluşturmaya gerek yoktur. TensorFlow Lite modelleri, Edge TPU'da çalışacak şekilde derlenebilir [258]. Örneğin, mobil ve gömülü cihazlarda kullanılan TensorFlow Lite tabanlı SSD MobileNetV2 gibi bir derin sinir ağı modelini neredeyse 400 FPS'de güç verimli bir şekilde çalıştırabilir. Raspberry Pi gibi Debian Linux bir sisteme USB aracılığıyla bağlanabilir.



**Şekil 3.85:** Google Coral Edge TPU

**Kaynak:** (coral.ai, 2023)

### 3.6.1.5 Harici USB ses kartı ve mikrofon

Raspberry Pi mini bilgisayarda çalışan çevresel ses sınıflandırma modelinin girdisi olan ses verisini elde etmek için Raspberry Pi ile uyumlu, mikrofon girişi için TRRS bağlantı noktasına sahip harici USB ses kartı kullanıldı (Şekil 3.86).



**Şekil 3.86:** Harici USB Ses Kartı

**Kaynak:** (www.teknostore.com, 2023)

Bu USB ses kartı üzerinden Şekil 3.87'de görüldüğü gibi Raspberry Pi'ye mikrofon entegre edildi.



**Şekil 3.87:** Harici USB Ses Kartı aracılığıyla Raspberry Pi'ye Mikrofon Entegrasyonu

### 3.6.2 İnsansız hava aracı platformu gömülü sistem yazılım mimarisi

İnsansız Hava Aracının gerçek zamanlı nesne tespit ve sınıflandırma görevlerini yapabilmesi ve otonom karar verme kabiliyeti kazanabilmesi için gömülü bir sistem tasarlandı. İHA platformu üzerinde yer alan bu sistem, Raspberry Pi mikroişlemci üzerine kurulu ve Raspberry ile çalışan ek donanımlara sahip, derin öğrenme tabanlı görüntü ve ses işleme teknolojilerinin kullanıldığı, SQLite veritabanı bulunan, Python programlama dili ile yazılan betiklerden oluşan ve MAVlink iletişim protokolü ile uçuş kontrol kartına komut gönderebilen bir yapıdadır. Bu kısımda gömülü sistem yazılım mimari ve otonom karar verme süreci detaylandırılmaktadır.

#### 3.6.2.1 Sistem yazılımı bileşenleri

İHA platformunda, gerek uçuş kontrol sistemi gerek ise gömülü sistem içerisindeki her donanımın açık kaynak kodlu olması önem arz etmektedir. Sistem parçaları arasındaki çalışma ve iletişim uyumu, kararlı ve sorunsuz bir işleyiş için kaçınılmazdır. Yazılım mimarisi göz önünde bulundurularak donanımlar belirlendi. Python programlama dili ve Python kütüphaneleri kullanılarak Raspberry Pi'de yapay zekâ işlemlerini gerçekleştirecek betikler kodlandı. Raspberry Pi ile Pixhawk arasındaki iletişim için MAVLink iletişim protokolü kullanıldı. Gerekliğinde geliştirme yapmak için C# nesne tabanlı programlama dili ile yazılmış Mission

Planer yer kontrol istasyonu tercih edildi. Bu bileşenler, aşağıda başlıklar altında detaylandırılmaktadır.

### 3.6.2.1.1 Python programlama dili

Python'da, sıfırdan kod yazmanıza gerek kalmadan sinir ağları oluşturmanıza olanak tanıyan Keras, Tensorflow, Pytorch ve diğerleri gibi farklı açık kaynak kodlu derin öğrenme kütüphaneleri bulunmaktadır. Numpy kütüphanesinin dâhil edilmesinden kaynaklanan dizilerin kullanım kolaylığı, Librosa kütüphanesi sayesinde ses sinyalleriyle çalışmanın doğrudan ve basit yolu, Pandas ile veri analizlerini hızlılığı ve Matplotlib ile bu verileri görselleştirmenin kolaylığı Python'un sinir ağı projeleri alanında en önemli yazılım dillerden biri olmasının nedenidir. Bu çalışmanın yazılım süreci, Python 3.9.0 sürümünde gerçekleştirildi, bu sürüm <https://www.python.org/> adresindeki özel web sitesinden indirilebilir. İndirildikten ve kurulduktan sonra, paket yöneticisi PIP sisteme dâhil edildi. Python betiklerinin kodlanması sırasında kullanılan birçok Python Kütüphanesi bulunmaktadır, bu kütüphanelerden önemli olanlarının birkaçının kullanım amaçları aşağıda açıklanmaktadır:

**Numpy:** Numpy, dizilerle çalışmak için kullanılan ve aynı zamanda doğrusal cebir, Fourier dönüşümü ve matrisler alanında çalışmaya yönelik işlemlere de sahip bir Python kütüphanesidir. Numpy için sayısal Python ifadesi kullanılabilir. Numpy, Python'daki tüm veri bilimi ve makine öğrenmesinin temelini oluşturmaktadır [261]. Sinir ağı modelleriyle çalışırken üst düzey işlevsellik için Numpy matematiksel Python Kütüphanesi kullanılır.

**Librosa:** Librosa, ses sinyallerini işlemek ve analiz etmek için kullanılan bir Python kütüphanesidir. Librosa kütüphanesi sayesinde ses sinyallerinin spektrogramları ve dalga formları görüntülenebilir ve bu sayede ses sinyallerinden bilgi çıkarılabilir [262]. Librosa kütüphanesi, konuşma ve ses olayı tanıma gibi çalışma alanlarında etkin biçimde kullanılmaktadır.

**Scipy:** SciPy, diferansiyel denklemler, integrasyon, enterpolasyon, doğrusal cebir, optimizasyon, seyrek matrisler gibi matematik ve mühendislik alanlarındaki çeşitli hesaplama problemlerini çözmek için NumPy üzerine kurulmuş bir Python kütüphanesidir [263]. Scipy, ses işleme ve görüntü işleme gibi alanlarda, özellikle

ses sinyalinin filtrelenmesi ve spektral analiz gibi görevlerde yaygın olarak kullanılmaktadır [264, 265].

**Matplotlib:** Matplotlib, verileri görselleştirmek ve dizilerin 2 boyutlu grafiklerini oluşturmak için kullanılan en yaygın Python kütüphanesidir [266, 267]. Matplotlib, sayısal bir matematik kütüphanesi olan NumPy ile çalışacak şekilde tasarlanmıştır ve Python için bir grup bilimsel hesaplama aracı olan SciPy yığınının temel bir parçasıdır.

**Pandas:** Pandas, veri analizine yönelik bir Python kütüphanesidir [268]. Pandas, veri görselleştirme için kullanılan Matplotlib ve matematiksel işlemler için kullanılan NumPy gibi iki temel Python Kütüphanesi üzerine kurulmuştur. Pandas, aslında Matplotlib ve NumPy gibi kitaplıkların bir tür sarmalayıcı işlevini görür. Bu sayede, daha az kod yazarak Matplotlib ve NumPy kütüphanelerinin birçok metoduna erişebilmenizi sağlar.

**Pyaudio:** Pyaudio, Linux, Microsoft Windows ve macOS gibi çeşitli platformlarda ses kaydetmek ve oynatmak için kullanılan bir Python kütüphanesidir [269].

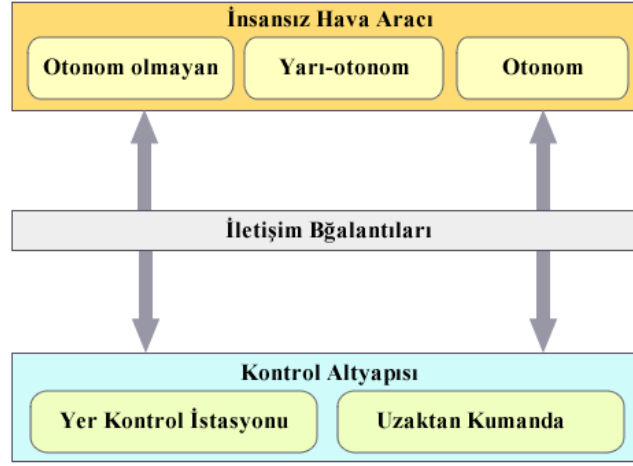
**Jupyter Notebook:** Jupyter notebook, Python kodunu çalıştırmak ve yürütülebilir kodun çıktılarını ve açıklamalarını kolayca paylaşmak için kullanılabilen bir web uygulamasına dayanmaktadır [270].

**Tensorboard:** TensorBoard, TensorFlow ile birlikte sağlanan ve derin sinir ağı modellerinin deneylerine yönelik kullanılan bir görselleştirme aracıdır [271]. TensorBoard, kayıp ve doğruluk gibi eğitim ölçümlerinin izlenmesine, model grafiğinin görselleştirilmesine, histogramların görüntülenmesine ve çok daha fazlasına olanak tanır. Bir sinir ağını optimize etmek için genellikle onu ölçebilmeniz gerekir. TensorBoard, iş akışı sırasında ihtiyaç duyulan ölçümleri ve görselleştirmeleri sağlamaktadır.

### 3.6.2.1.2 Yer kontrol istasyonu yazılımı – mission planner

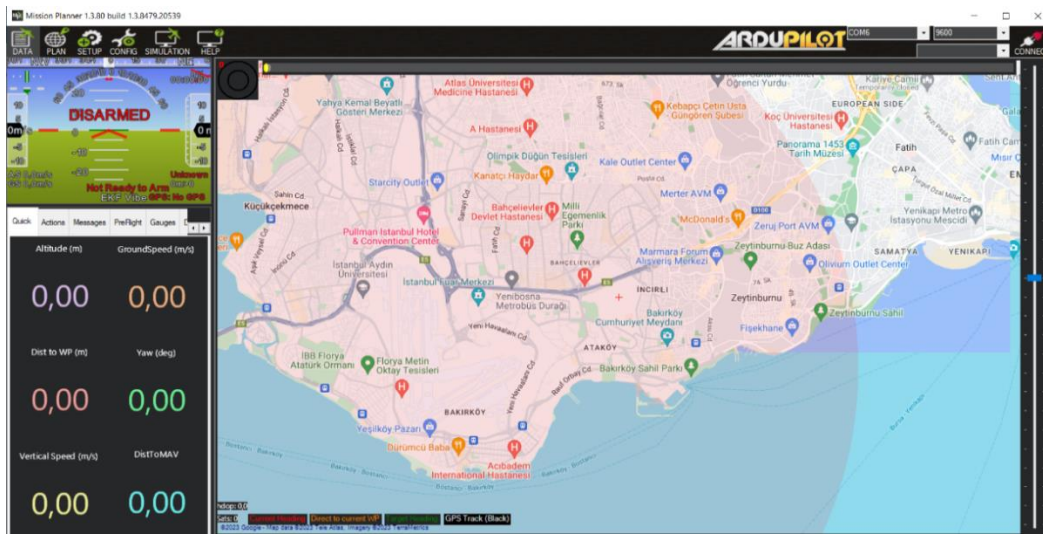
Bir yer kontrol istasyonu (Ground Control Station-GCS), insan tarafından uzaktan kumandalı ile yönetilen İHA'ların kontrolü için gerekli unsurları sunan komuta merkezidir. Bir GCS, uçuş planlaması ve uçuş izlemenin yanı sıra İHA-çevre etkileşimlerinin gerçek zamanlı işlenmesi için tasarlanmıştır. Bir GCS, görevlerin

planlanmasını gerçekleştiren, verilerin izlenmesine olanak sağlayan bir yazılım ara yüzüne ve İHA'larla telemetri aracılığıyla bağlantı kurabilen bir bağlantıya sahiptir [27]. GCS ile İHA arasında çift yönlü veri akışı bulunur. Bu veri akışı MAVLink protokolü üzerinden gerçekleşir. GCS'den İHA'ya görev komutları gönderilirken İHA'dan GCS'ye uçuş verileri aktarılır (Şekil 3.88).



Şekil 3.88: İHA ve Yer Kontrol Merkezi İletişim Bağlantıları

Mission Planner, yer kontrol istasyonu yazılımları arasında en çok tercih edilen İHA kontrol yazılımıdır [272]. Bu yazılımla; İHA komponentlerinin kalibrasyonları yapılabilir, birçok uçuş modu seçilebilir ve otonom uçuş planlanabilir, İHA'ya komut gönderilebilir, İHA'dan hız, konum, durum verileri anlık alınabilir, İHA'nın uçuşu izlenebilir, ayrıca çok detaylı uçuş log kayıtlarına da erişilebilir. Şekil 3.89, Mission Planner programının kullanıcı arayüzünü göstermektedir.

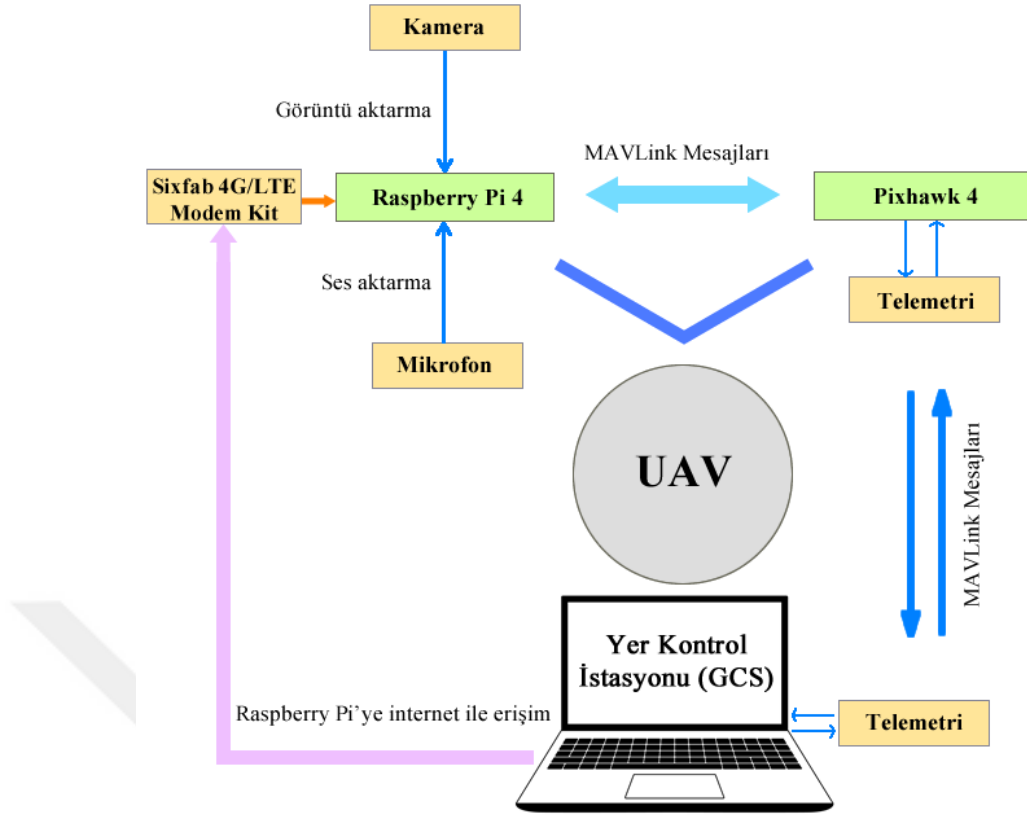


Şekil 3.89: Mission Planner Yer Kontrol İstasyonu Yazılımı Arayüzü

Bu çalışmada, Mission Planner'ın tercih edilmesinde birkaç önemli faktör vardır. Bunlardan en önemlisi, Mission Planer yazılımının C# ile kodlanmış ve projenin web ortamında geliştirilmeye açık olmasıdır. Böylelikle, projenin gidişine uygun olarak yenilikler eklenebilir. Diğer faktörler ise kullanıcı ara yüzünün kolay ve kullanışlı olması, ayrıca Mission Planer yazılımının bu konu ile ilgilenen yazılımcılar tarafından devamlı geliştirildiği için güncelliğini her daim koruyor olmasıdır.

### **3.6.2.1.3 MAVLink haberleşme protokolü**

İnsansız Hava Araçları (İHA'lar), uçuşlarını iki farklı yöntemle gerçekleştirirler. Ya bir yer kontrol istasyonu (GCS) tarafından uzaktan kontrol edilir ya da önceden programlanmış görevlere otomatik olarak (auto mode - otonom uçuş) yönlendirilirler. İHA'lar, belirli bir bölgeyi haritalamak, gözetlemek veya veri toplamak gibi görevleri yerine getirmesi için kullanılabilir. Uzaktan kontrol edildiğinde, İHA ile GCS arasındaki iletişim, bir iletişim protokolü aracılığıyla kurulur. Micro Hava Aracı bağlantısı (MAVLink), İHA'lar ile GCS'ler arasındaki iletişim için en yaygın kullanılan protokollerden biridir [273]. MAVLink, otomatik pilot ile GCS arasında çift yönlü veri alışverişi için geliştirilmiştir, esnek ve açık kaynaklı bir iletişim protokolüdür. GCS, İHA'ya komutlar gönderip kontrol sağlarken, İHA ise telemetri ve durum bilgisi verilerini GCS'ye gönderir. İHA'nın durumu, pozisyonu, hızı ve diğer önemli bilgilerini içeren tüm verileri taşımak için farklı mesaj türleri vardır. HEARTBEAT mesajı, bu iletişimin temel bir parçasıdır çünkü İHA'nın güvenilir bir şekilde çalıştığını ve bağlantısının sürdüğünü doğrular. Bu sayede yer kontrol istasyonu, İHA'nın durumunu izleyebilir ve gerektiğinde müdahale edebilir. MAVLink, Ardupilot ve PX4 dâhil birkaç otomatik pilot sistemleri tarafından kullanılmaktadır. Ardupilot ve PX4 açık kaynak kodlu otomatik pilot sistemleri, sabit kanatlı uçaklardan çeşitli döner kanatlı platformlara kadar her türlü İHA'yı kontrol etmek üzere tasarlanmıştır. Bu sistemler, İHA'ların güvenli ve etkili bir şekilde uçmasını sağlamak için geniş bir işlevselliğe sahiptirler [274]. Bu çalışmada MAVlink protokolü, Şekil 3.90'da gösterildiği gibi hem Pixhawk 4 uçuş kontrol kartı ile GCS arasında hem de Raspberry ile Pixhawk 4 arasındaki iletişim için kullanıldı.



**Şekil 3.90:** Rasperry Pi 4-Pixhawk 4 ve Pixhawk 4-GCS arasındaki MAVLink İletişimi

Python betikleri, MAVlink aracılığıyla Rasperry Pi'den Pixhawk uçuş kontrol kartına komutlar göndererek İHA'nın hareketini kontrol edebilmektedir (Şekil 3.91).

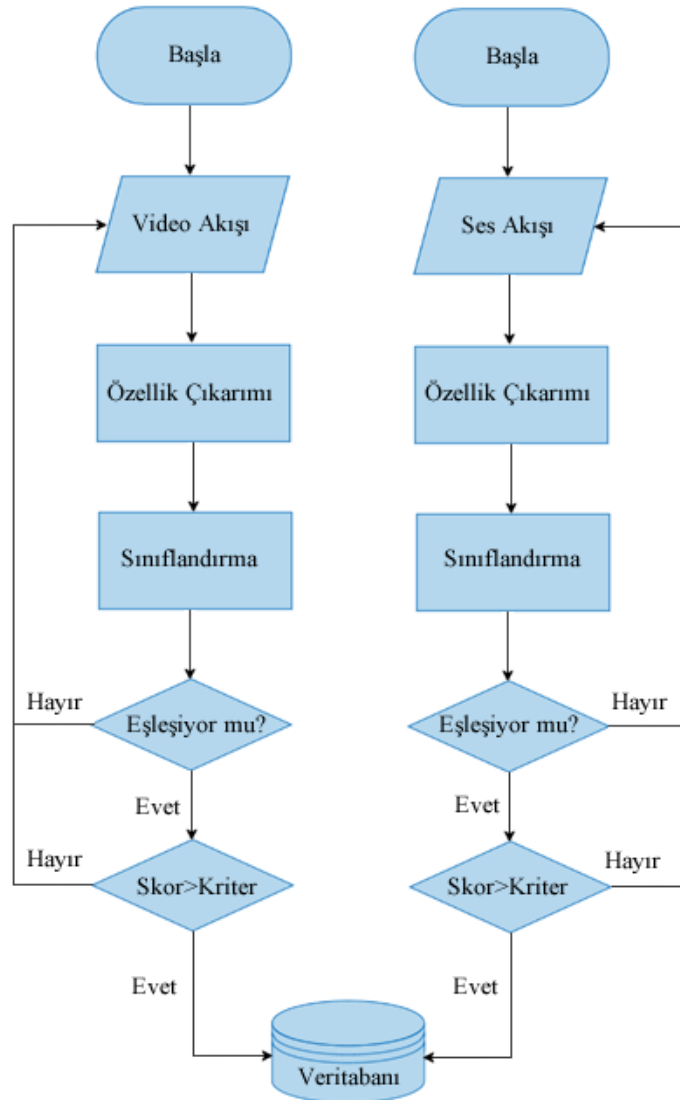
```

pi@raspberrypi:~$ mavproxy.py --master=/dev/ttyAMA0
Connect /dev/ttyAMA0 source_system=255
Log Directory:
Telemetry log: mav.tlog
Waiting for heartbeat from /dev/ttyAMA0
MAV>
MAV> online system 1
STABILIZE> Mode STABILIZE
fence present
AP: ArduCopter V4.3.2 (fc0f70e7)
AP: ChibiOS:
AP: Pixhawk4 00320042
AP: RCOut: PWM:1-16
AP: IMU0: fast sampling enabled 0.0kHz/2.0kHz
AP: Frame: HEXA/X
Received 955 parameters (ftp)
Saved 955 parameters to mav.parm
STABILIZE> no
  
```

**Şekil 3.91:** Rasperry Pi'den Pixhawk 4 Uçuş Kontrol Kartına Komut Gönderme

### 3.6.2.2 Otonom Karar Verme Yapısı

İHA gömülü sistem yazılımı, Python programlama dili ile yazılan ve birbiri ile sekronize çalışan birçok betikten meydana gelmektedir. Kodlanan tüm yazılım Raspberry Pi'ye yüklenmiştir. Derin öğrenme algoritmalarını içeren görüntü ve ses işleme yazılımları, gömülü sistem donanımları aracılığıyla İHA'nın çevresini analiz eder. Bu analiz sonucunda elde edilen bilgiler (nesne sınıfı, sınıflandırma skoru, tarih-saat vb. bilgiler) veri tabanına kaydedilir ve karar verme algoritmalarına iletilir. Otonom karar verme süreci, belirli hedeflere ulaşma veya belirlenen görevleri gerçekleştirme gibi hedeflere yönelik olabilir.



Şekil 3.92: Gömülü Sistem Yazılım Mimarisi

Şekil 3.92’de görüldüğü üzere nesne tespit ve sınıflandırması için literatürde ayrı ayrı kullanılan görüntü ve ses işleme teknolojileri, bu çalışmadaki İHA

platformunda aynı anda kullanıldı ve İHA'nın nesne sınıflandırma kabiliyetinin çok daha güçlü seviyeye ulaşması sağlandı. Raspberry Pi'ye güç verilmesi ile birlikte ön yüklemeye çalışmaya başlayan, Python programlama dili ile yazılmış birçok program (script) bulunmaktadır. Bunlardan *objectClassImg.py* programı video akışı üzerinde anlık olarak nesne tespit ve sınıflandırması için çalışırken, *objectClassSnd.py* ses akışı üzerinden nesne tespit ve sınıflandırması için çalışmaktadır. Belirli bir nesnenin veya nesnelerin tespiti için modellenen bu yazılımlar, sistemde tanımlı nesneyi daha önceden belirlenen nesne sınıflandırmasına ait doğruluk değerleri (skor) üzerinde tespit etmeleri durumunda bunu veri tabanına anlık olarak kaydeder. *Mission.py* programı ise veri tabanından devamlı sorgu çekerek ikili tespitin aynı süre dilimi içerisinde (gecikme aralığı dahil edilerek) sağlanması durumunda görev komutlarını uçuş kontrol kartına gönderir ve böylelikle süreç tamamlanır. Derin öğrenme mimarilerine sahip görüntü ve ses ile nesne sınıflandırma teknolojileri ayrı ayrı da kullanılabilir. Bu durumda *Mission.py* programı, tek bir koşulun sağlanıp sağlanmadığına bakar ve ona göre uçuş kontrol kartına komut gönderir. Tüm bu süreç içerisinde sekronize çalışan *sendReport.py* betiği, tüm analiz bilgilerini kullanılan ağ bağlantısı üzerinden yer kontrol merkezine göndermektedir. Bu sayede gerektiğinde görev koşulları ve görev tanımı uzaktan müdahale ile güncellenebilmektedir.

#### 4. İHA UÇUŞ TESTLERİ SONUÇLARI

Yapılan uçuş testleri, SSD MobilNet V2 evrişimsel sinir ağının, İHA tarafından sağlanan video akışından gerçek zamanlı olarak farklı nesne sınıflarını Çizelge 4.1’de gösterildiği gibi, insan tespiti %96.2 ve araba tespiti %96.3, doğruluk yüzdeleri ile tespit edip tanımlayabildiğini göstermiştir.

**Çizelge 4.1:** Nesne Sınıflandırma ve Doğruluk Yüzdesi

Nesne (Object)	Doğruluk Yüzdesi (%)	Başarı
Araba (Car)	96.3	✓
İnsan (Person)	96.2	✓

Şekil 4.1 ve Şekil 4.8 arasındaki görüntüler, İHA kamera tarafından sağlanan video beslemesinden evrişimsel sinir ağı algoritmasının nesne algılama senaryoları üzerindeki testlerinden elde edilen sonuçları göstermektedir. Sinir ağı mimarisine sahip derin öğrenme algoritmasının "araba" ve "insan" gibi iki farklı sınıfa nesneyi tespit edip tanıyabildiği görülmektedir (Şekil 4.1 ve Şekil 4.2). Ayrıca sinir ağı, bir görüntüdeki dört nesneyi de doğru bir şekilde tespit edip "insan" ve "araba" kategorisi olarak sınıflandırabilmiştir (Şekil 4.3 ve Şekil 4.4).

Diğer taraftan, Şekil 4.3, Şekil 4.4 ve Şekil 4.5’te görüldüğü üzere derin öğrenme modeli, güneş ışınlarının yoğun olduğu ve direkt kameraya vurduğu bir ortamda görüntü açısındaki nesnelerin tamamını sınıfları ile birlikte doğru tahmin etmiştir. Şekil 4.6 ve Şekil 4.7’ye bakıldığında, görüntünün arka tarafındaki iki insanın birbirine çok yakın mesafe olduğu ve öndeki insanın arkada insanın görüntü açısını kapattığı görülmektedir.

Model, bu kısıtlayıcı duruma rağmen arkadaki ikinci insanı da sınırlayıcı kutu ile tespit edebilmiştir. Ayrıca, Şekil 4.8’deki görüntüde görüldüğü üzere İHA’nın dönüş hareketi sırasında görüş açısı eğimli dahi olsa modelin, nesnelere doğru tespit edip sınırlayıcı kutularla belirlediği görülmektedir. Elde edilen nesne tanıma ve sınıflandırma doğruluğu seviyesine dayanarak bu yaklaşımın hem ticari hem de askeri uygulamalarda kullanılabileceği söylenebilir.



**Şekil 4.1:** SSD MobileNetV2 Modeli İle Aynı Anda Dört Nesnenin Tespit Edilmesi ve Sınıflandırılması



**Şekil 4.2:** SSD MobileNetV2 Modeli İle Aynı Karede İki Farklı Sınıfa Ait Nesnelerin Tespit Edilmesi

SSD MobileNetV2 derin öğrenme modeli, yoğun güneş ışınları altında Şekil 4.3'te görüldüğü gibi tanımlı iki farklı sınıfa (insan ve araba) ait dört nesneyi tespit ederek sınırlayıcı kutu içine almıştır.



**Şekil 4.3:** SSD MobileNetV2 Modeli İle İnsan ve Araba Tespiti

Şekil 4.4’de görüldüğü üzere, SSD MobileNetV2 derin öğrenme modeli, yoğun güneş ışınları altında araba nesnesini %98 doğrulukla tespit edebilmiştir.



**Şekil 4.4:** SSD MobileNetV2 Modeli İle Araba Tespiti



**Şekil 4.5:** SSD MobileNetV2 Modeli İle Yoğun Güneş Işınları Altında Nesne Tespiti

Şekil 4.6 ve Şekil 4.7’de görüldüğü üzere tanımlı, SSD MobileNetV2 modeli, birbirine çok yakın olan iki insanı ayrı ayrı tespit ederek sınırlayıcı kutu ile içine almıştır.



**Şekil 4.6:** SSD MobileNetV2 Modeli İle Aynı Anda Üç İnsan Tespiti



**Şekil 4.7:** SSD MobileNetV2 Modeli ile İnsan ve Araba Tespiti

Ayrıca, SSD MobilNetV2 derin sinir ağı modeli, İHA'nın dönüş hareketi sırasında kamera açısı eğimliken bile tanımlı iki sınıfa ait dört nesneyi tespit edip sınıflandırabilmiştir.



**Şekil 4.8:** SSD MobileNetV2 Modeli İle Eğimli Kamera Açısı İle Nesne Tespiti

Derin öğrenme tabanlı çevresel ses sınıflandırma modeli YAMNet, ses verisi ile helikopter sınıfına ait nesnelere tespit edebilmesi için transfer öğrenme yöntemi eğitildi. Eğitim için ESC-50 veri setindeki helikopter sınıfına ait ses verileri kullanıldı. Gerek yasal mevzuatlar gerekse İHA'nın uçuş testleri sırasında uçuş sahası içinde helikopterin varlığı sağlanamayacağı için sinir ağı modelinin testi,

gerçek uçuş ortamını yansıtacak şekilde kapalı bir spor salonunda gerçekleştirildi (Şekil 4.9).



**Şekil 4.9:** YAMNet Ses Sınıflandırma Modelinin Deneysel Test Çalışma Ortamı

İnsansız hava aracının kalkışını engellemek için tüm motorların kaldırabileceği maksimum yük miktarını aşacak şekilde ( 6 motor, %100 kapasite ile  $6 \times 1650\text{gr} = 9.9\text{ kg}$  yük kaldırabilir, Hexacopter'in kendi ağırlığı yaklaşık 4kg dır) Hexacopter'in iniş takımının ayaklarına toplamda 7 kg (iniş takımının sol ayağına 3kg, sağ ayağına 4kg (2+2)) ağırlık konuldu ve motorlar çalıştırıldı (Şekil 4.10).



**Şekil 4.10:** Hexacopter'in Kalkışını Engellemek İçin İniş Takımı Ayaklarına Ağırlık Konulması

Helikopter sesi, cep telefondan açılarak bluetooth ile hopörlere aktarıldı. Spor salonunun uzunluğu göz önüne alınarak Hexacopter sabit kalmak üzere, hopörler 20mt uzaklığa götürüldü (Şekil 4.11).



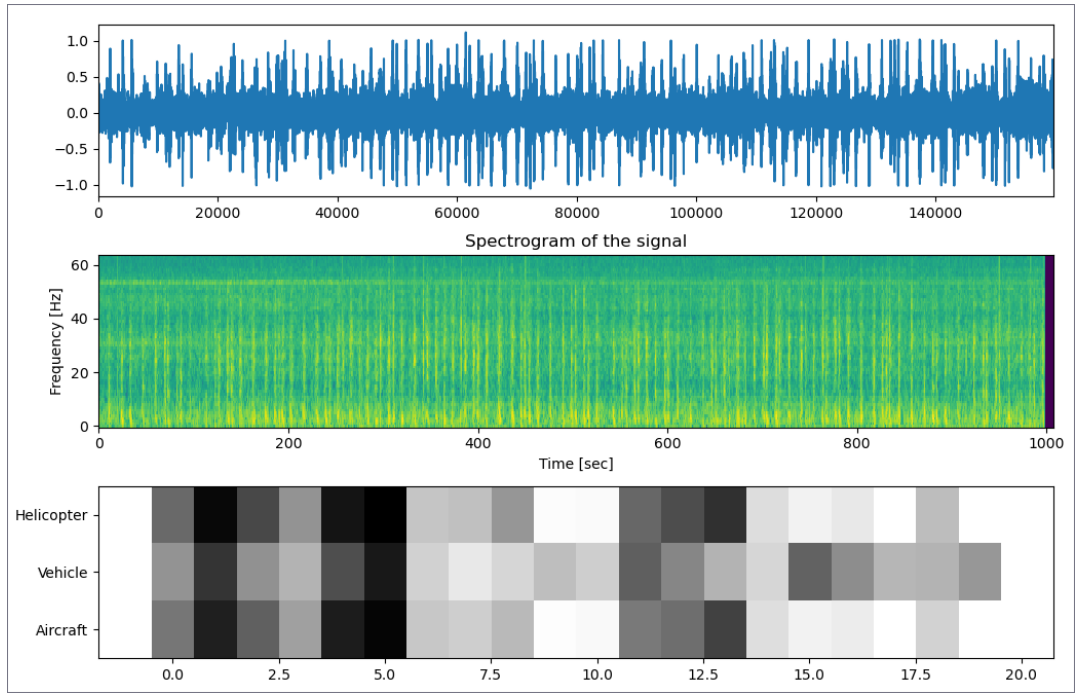
**Şekil 4.11:** Hexacopter'in Hover Modunda Belirli Mesafeden Helikopter Sesini Tanımasını Sağlayacak Test Ortamı

Ses sınıflandırma modelinin helikopter sesini tanıma test çalışmalarında; hover modunda ve hover modundaki roll left, roll right, pitch forward, pitch back, yaw left ve yaw right uçuş hareketlerinde helikopter sesini tanıyıp tanımadığı ayrı ayrı test edildi (Şekil 4.12).



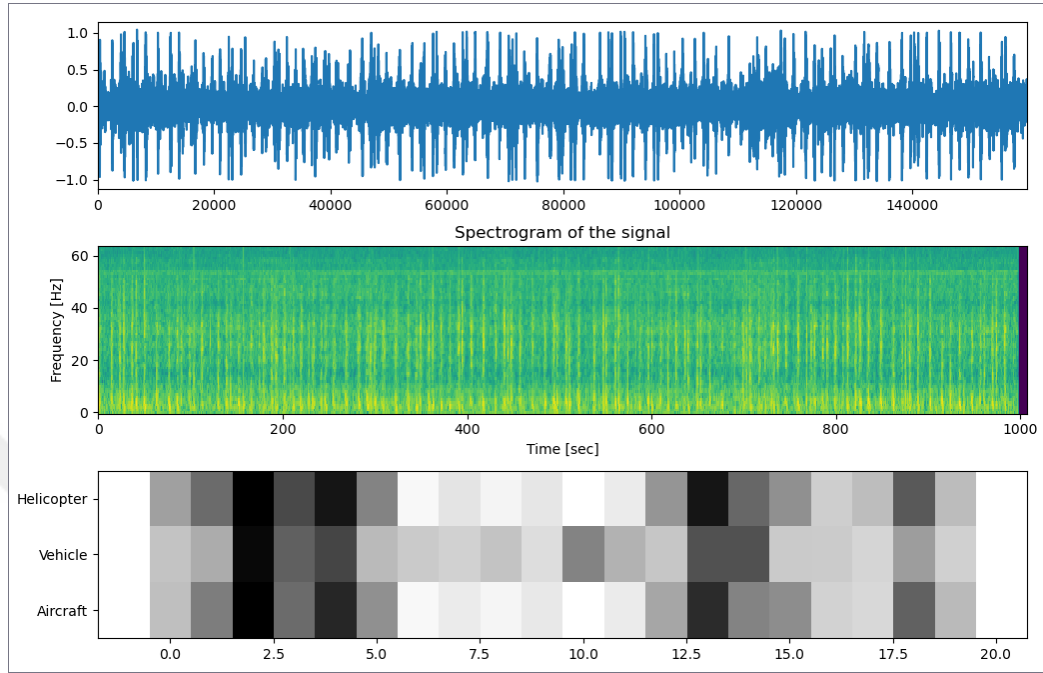
**Şekil 4.12:** Uçuş Hareket Komutlarının Kumandadan Gönderilmesi

Her durum için çıktı sonuçları aşağıda gösterilmektedir. Şekil 4.14, ses sınıflandırma modeli YAMNet'in, Hexacopter hover uçuş modun uçarken helikopter sesini tanıyabildiğini göstermektedir.

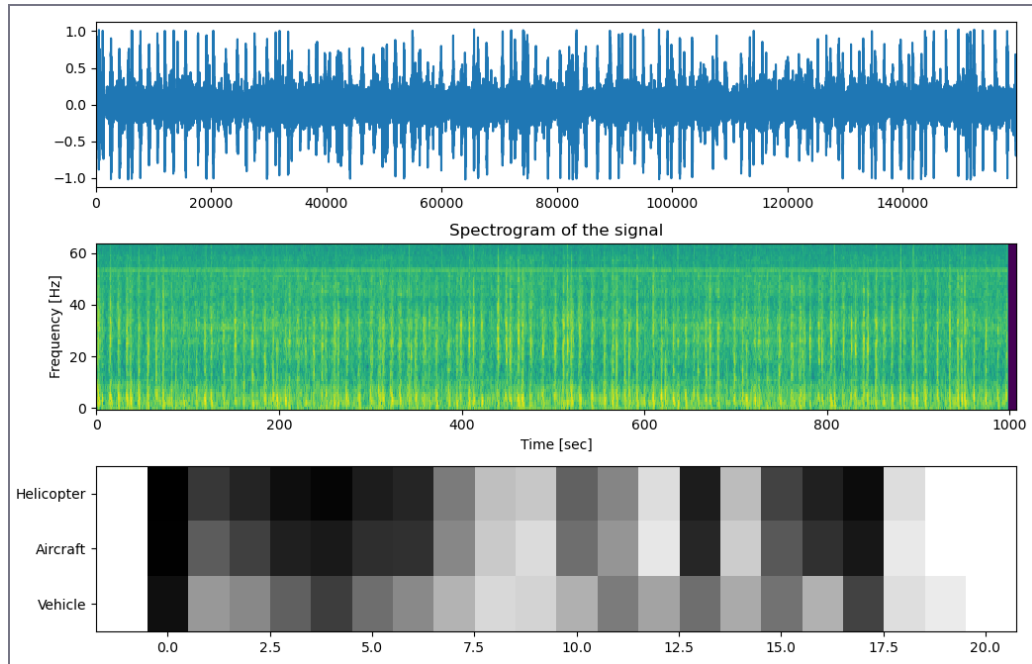


**Şekil 4.13:** Hexacopter Hover Modunda Ses Sinyalinin (İHA ve Helikopter Sesi) Spektrogramı ve Ses Sınıflandırma Modelinin Tahmini

Şekil 4.14 ve Şekil 4.15 görüldüğü üzere ses sınıflandırma modeli YAMNet, Hexacopter Roll Left ve Roll Right hareketi yaparken helikopter sesini tanıyabilmiştir.

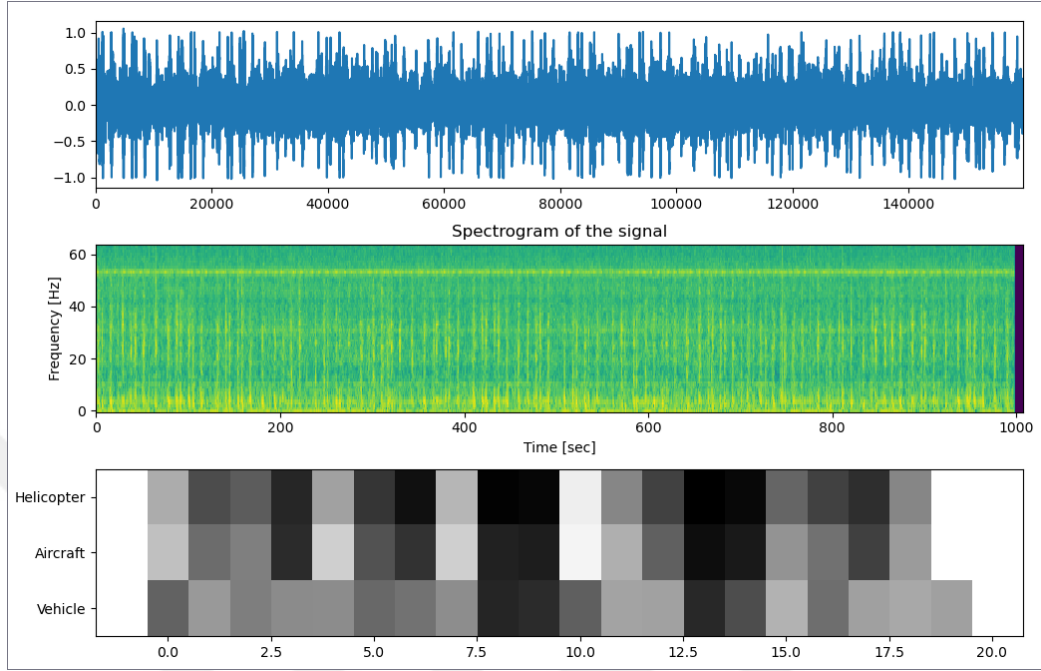


Şekil 4.14: Hexacopter Roll Left Hareketinde Ses Sinyalinin (İHA ve Helikopter Sesi) Spektrogramı ve Ses Sınıflandırma Modelinin Tahmini

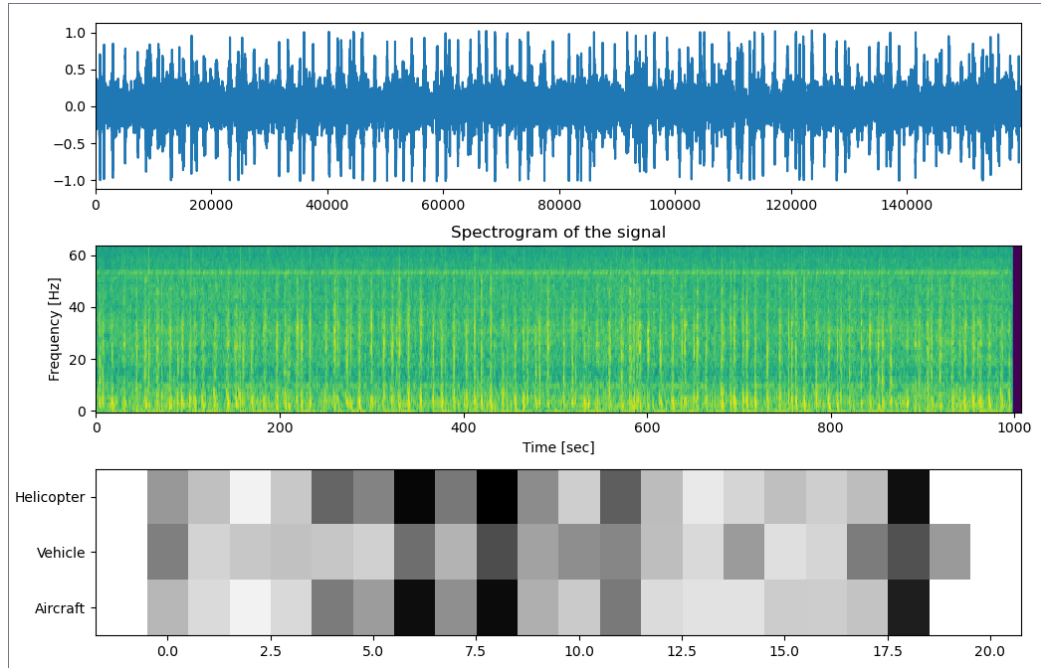


Şekil 4.15: Hexacopter Roll Right Hareketinde Ses Sinyalinin (İHA ve Helikopter Sesi) Spektrogramı ve Ses Sınıflandırma Modelinin Tahmini

Şekil 4.16 ve Şekil 4.17 görüldüğü üzere ses sınıflandırma modeli YAMNet, Hexacopter Yaw Left ve Yaw Right hareketi yaparken helikopter sesini tanıyabilmiştir.

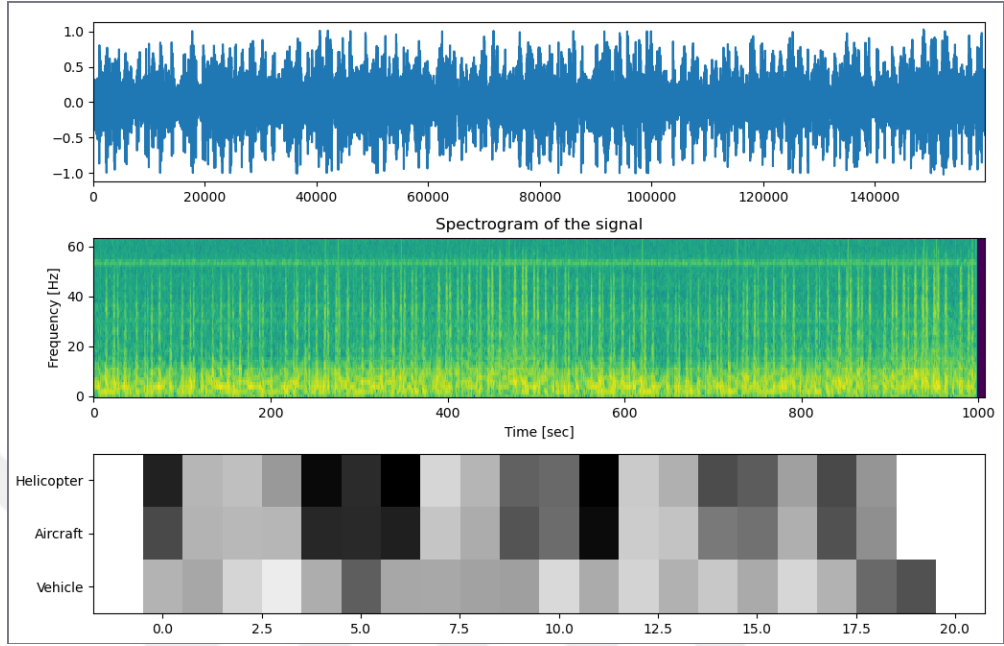


Şekil 4.16: Hexacopter Yaw Left Hareketinde Ses Sinyalinin (İHA ve Helikopter Sesi) Spektrogramı ve Ses Sınıflandırma Modelinin Tahmini

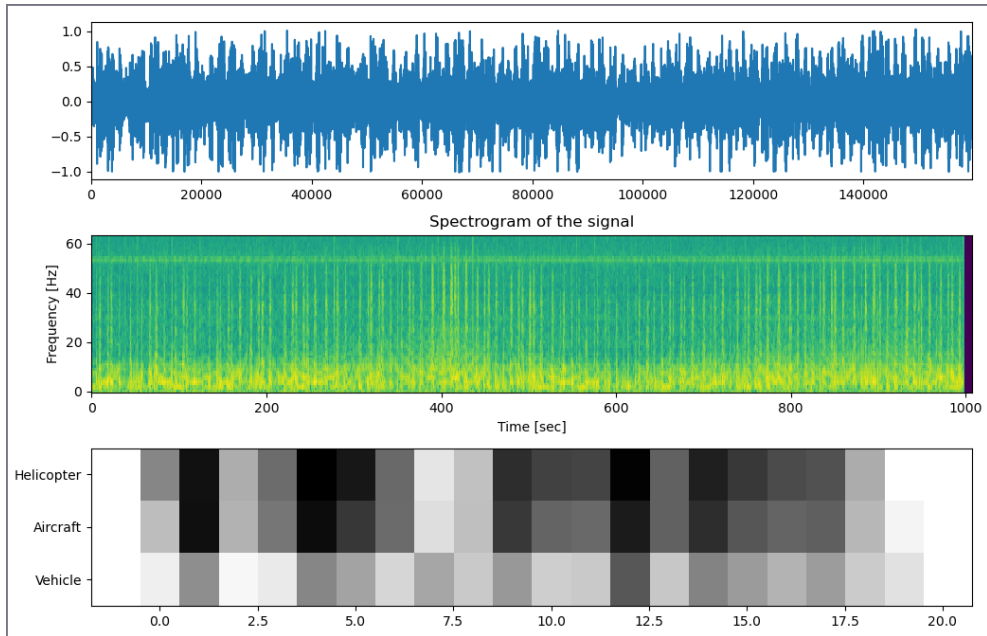


Şekil 4.17: Hexacopter Yaw Right Hareketinde Ses Sinyalinin (İHA ve Helikopter Sesi) Spektrogramı ve Ses Sınıflandırma Modelinin Tahmini

Şekil 4.18 ve Şekil 4.19 görüldüğü üzere ses sınıflandırma modeli YAMNet, Hexacopter, Pitch Forward ve Pitch Back hareketi yaparken helikopter sesini tanıyabilmiştir.



**Şekil 4.18:** Hexacopter Pitch Forward Hareketinde Ses Sinyalinin (İHA ve Helikopter Sesi) Spektrogramı ve Ses Sınıflandırma Modelinin Tahmini



**Şekil 4.19:** Hexacopter Pitch Back Hareketinde Ses Sinyalinin (İHA ve Helikopter Sesi) Spektrogramı ve Ses Sınıflandırma Modelinin Tahmini

YAMNet modelinin çıktı sonucunda en yüksek sınıflandırma skoruna sahip 3 sınıf listelenmektedir. Bu, yazılım tarafında belirlenmektedir. Deneysel sonuçlara

göre YAMNet ses sınıflandırma modelinin helikopter sesini tüm uçuş hareketlerinde tespit edebildiği görülmektedir.



## 5. TARTIŞMA

Bu çalışmaya başlamadan önce ve çalışmanın başladığı andan itibaren literatürde nesne tespit ve sınıflandırma için kullanılan derin öğrenme modelleri ve bu modellerin hangi alanlarda kullanıldığına dair çok kapsamlı araştırmalar yapıldı. Özellikle de görüntü ile nesne tespit ve sınıflandırma yöntemlerinin İHA platformlarındaki kullanım alanları ve amaçları incelendi. Tarım [275, 276], arama kurtarma [277-279], coğrafi bilgi sistemleri [280, 281], şehircilik [282], enerji nakil hatlarının izlenmesi [283, 284], askeri [34, 285] ve daha birçok alanda İHA ile bilgisayarlı görü yöntemlerinden faydalandığı görüldü. Derin öğrenme modelleri ile ses verisi üzerinden nesne tespit ve sınıflandırma uygulamalarının kısıtlı ve daha çok tıp alanında hastalıkların tanısını koymada bir yardımcı araç olarak kullanılabileceğine dair çalışmalara rastlandı [286-293].

Bu çalışma, nesne tespit ve sınıflandırma için görüntü ve ses verisini aynı anda kullanabilmeyi amaçlamıştır. Çünkü böyle bir sistem, nesne tespit ve sınıflandırmada iki önemli yeteneğe sahip olabilmektedir. Bunlardan birincisi, sistem görüntü verilerine sahip olmadığı koşullarda nesne tespit ve sınıflandırma görevini ses verisi üzerinde yürütebilir ve tam tersi durumda, yani ses verisinin yetersiz veya sorunlu olduğu durumlarda görüntü verisi üzerinden görevini icra edebilir. İkinci ise sistemin çapraz doğrulama yeteneğine sahip olmasıdır. Yüksek doğruluk gerektiren hassas görevler için her iki nesne tespit ve sınıflandırma yöntemi kullanılarak çapraz doğrulama sağlanabilmektedir. Bu kapsamda, bu çalışma literatüre katkı sağlamaktadır.

Derin öğrenme teknolojisine sahip bu sistem, İHA platformunda gömülü sistem olarak kullanıldı ve testleri yapıldı. Ses sınıflandırma modeli, özel bir görev olarak helikopter nesnesini tespit edebilmesi için tasarlandı. Ancak, İHA'nın uçuşu için yasal mevzuat böyle bir deneysel çalışmaya izin vermediğinden, test çalışmaları kapalı bir spor salonunda simüle edildi. Diğer taraftan, İHA platformunun uygulama alanı askeri alan olarak hedeflendiğinden ve ses verisi ile nesne tespiti için helikopter seçildiğinden görüntü ile nesne tespit ve sınıflandırma modelinin deneysel

çalışmaları, insan ve araba gibi uçuş sahasında bulunabilir nesnelere üzerinden gerçekleştirildi. SSD MobilNetV2 modeli, iki farklı görev için ayrı olarak transfer öğrenme yöntemi ile yeniden eğitildi ve özelleştirildi. İlk olarak modelin, insan ve araba nesnelere tanıması amaçlandı. Ardından yine aynı sinir ağı yapısı, belirli görev olarak Apache askeri helikopteri tanıyacak şekilde sıfırdan yapılandırıldı. İkinci uygulama, kendi özelinde çalışılmamış bir alan olarak kabul edilerek çalışmadaki modelin eğitim ve özelleştirme süreci bu ikinci uygulama (askeri helikopterin tanınması) üzerinden detaylandırıldı.

Uçuş testlerinden elde edilen veriler, görüntü üzerinden yapılan nesne tespiti için önemli kanıtlar sunar. Bu veriler, sistemin gerçek dünya uygulamalarında başarılı olduğunu ve farklı uçuş koşullarında güvenilir bir şekilde nesne tespiti yapabildiğini gösterir. Benzer şekilde, ses verisi üzerinden yapılan nesne tespiti deneylerinin sonuçları da çalışmanın amacına ulaştığını kanıtlar niteliktedir. Bu veriler, sistemin sadece görüntü değil, aynı zamanda ses analizi yoluyla da nesne tespitinde başarılı olduğunu gösterir. Sonuç olarak, deneysel sonuçlara bakarak, bu çalışmanın, hem görüntü hem de ses verileri üzerinden nesne tespiti yapabilen bir sistem geliştirme hedefini başarıyla gerçekleştirdiğini göstermektedir. Bu uygulama alanları açısından geniş bir potansiyel sunar ve hem askeri hem de teknoloji alanında önemli bir adım olabilir.

Bu çalışmanın, daha önceki çalışmalardan farklı olarak getirdiği yenilikler Çizelge 5.1’de özet olarak detaylandırılmaktadır.

**Çizelge 5.1: Çalışmadaki Yenilikler**

Yenilikler
1- Derin öğrenme tabanlı ses ve görüntü işleme teknolojisinin bir arada kullanılması. Bu sayede: A- Görüntü verisinin sağlıklı alınmayacağı koşullarda (sisli hava, gece karanlığı, çok fazla güneş ışığının kamerayı etkilediği durumda) ses verisi ile nesne tespit ve sınıflandırma görevi icra edilebilir. B- Ses verisinin sağlıklı alınmayacağı aşırı gürültülü ortamlarda görüntü verisi ile görev icra edilebilir. C- Her iki teknoloji ile çapraz doğrulama sağlanabilir.
2- Ses sınıflandırma modelinin tahmin doğruluğunu olumsuz yönde etkileyecek İHA motorlarının ve pervanelerinin ürettiği sesler, uçuş hareketlerine göre ayrı ayrı incelenerek modelleme yapıldı ve dijital filtreleme analizlere göre değişken yapıda tasarlandı.

**Çizelge 5.1: (Devamı) Çalışmadaki Yenilikler**

<b>Yenilikler</b>
3- Ses ve görüntü işleme teknolojisine sahip gömülü sistem ile İnsansız Hava Aracının uçuş kontrol kartı arasında MAVLink iletişim protokolü üzerinden iletişim sağlandı. Gömülü sistemdeki Raspberry Pi'nin ses ve görüntü işleme görevlerine paralel olarak Pixhawk uçuş kontrol kartını yönetmesi sağlandı. İHA'lar için otonom kelimesi daha çok yer kontrol istasyonu aracılığıyla uçuş kontrol kartına yüklenen planlı uçuş için kullanılır. Bu çalışmada bahsedilen ise gömülü sistemin görev icrasına göre farklı senaryolarda farkı işlemleri yürütebilme yeteneğidir.
4- Gömülü sistemin ana unsuru Raspberry Pi'ye, Six 4G/LTE modem kit entegre edilerek sürekli internet erişimi sağlandı. Böylelikle, modem kit arayüzü ile Raspberry Pi'ye müdahale edebilme imkânı sağlandı. Ayrıca, Raspberry uçuş sırasında istenilen her bilgiyi rapor olarak gönderebilme yeteneğine sahip oldu.

Yukarıda ayrı ayrı anlatılan tüm teknolojik uygulamaları bir arada bulunduran bu çalışmanın, literatüre katkı sunduğu düşünülmektedir.

## 6. SONUÇ VE ÖNERİLER

Bu çalışmada, İHA'ya otonom karar verme yeteneği kazandıracak nesne algılama ve sınıflandırma görevleri için derin öğrenme tabanlı evrimsel sinir ağı mimarilerine sahip SSD MobilNetV2 modeli ile görüntü verileri ve YAMNet modeli ile ses verileri üzerinden gerçek zamanlı algılama yapabilen gömülü sistem ve İHA platformu tanıtıldı. Çalışma süresince her iki model, farklı hiperparametre değerleri kullanılarak transfer öğrenme yöntemi ile eğitildi ve modellerin nesne sınıflandırma başarısı ayrı ayrı analiz edildi. Her bir modelin hiperparametre değerlerinin etkisinin optimal düzeye ulaşılması sağlandı.

DeneySEL sonuçlara göre, görüntü ile nesne tespitinde araba için %96,3 ve insan için %96,2 olarak bulunan doğruluk değerleri, bu çalışmanın uygulanabilirliğini kanıtlamaktadır. Bu değerler, İHA tarafından kullanılan kameranın çözünürlük kalitesi ve veri setinin büyüklüğü ile doğrudan ilişkilidir.

Transfer öğrenme yöntemi ile yeniden eğitilen ve özelleştirilen, ardından TensorFlow Lite formatına dönüştürülen SSD MobileNetV2 derin öğrenme modelinin, Raspberry Pi gibi sınırlı kapasiteye sahip bir cihazda, Edge TPU yardımcı işlemci ile hızlı ve iyi tespit doğruluğuna sahip bir model olduğu gözlemlenmiştir.

Derin öğrenme tabanlı ses sınıflandırma modeli YAMNet, helikopter nesnesini ses verisi üzerinden doğru bir şekilde sınıflandırması için sinir ağı mimarisi göreve özel optimize edildi ve transfer öğrenme yöntemi ile yeniden eğitildi. Sistem real-time analiz yapabilecek şekilde tasarlandı çünkü sahada bir mikrofon kullanarak sesi yakalamak ve bulutta analiz etmek her zaman mümkün olmayabilir. Bulutta analiz etme yöntemi, üçüncü bir tarafın olası müdahalesine açık olacağından gizlilik endişelerini artırabilir ve aynı zamanda bu yöntem önemli bir gecikmeye de neden olur. Bu sebeple, gömülü sistem olarak çalışması büyük avantaj sağlamaktadır.

Ses sınıflandırma modelinde hedef görev olarak helikopter sesi seçilmesinin iki ana nedeni bulunmaktadır. Birincisi, drone olarak tabir edilen İHA'ların günümüzde özellikle askeri alanda kullanımının çok yaygın olmasıdır. İkincisi ise,

ses sınıflandırma modelinin eğitimi için gerekli veri setini oluştururken gerekli ses verilerinin, askeri nesne olarak helikopter olarak seçilmesi durumunda daha ulaşılabilir olmasıdır. Ancak, bu durum beraberinde bir dezavantaj getirmiştir. Ses sınıflandırma modelinin sahadaki deneysel testleri için helikopter gerektirdiğinden, bu ortam kapalı bir spor salonunda simüle edildi ve testler gerçekleştirildi. Elde edilen deneysel sonuçlar ses sınıflandırma modelinin İHA'larda kullanılabileceğini göstermektedir.

Sistem verimliliğini artırmak için derin öğrenme modellerinde daha fazla optimizasyon yapılabilir ve sistemin daha iyi donanımlarla (örneğin yüksek çözünürlüğe sahip bir kamera ve daha hassas ses algılayabilen bir mikrofon kullanılması) daha iyi performanslı olması sağlanabilir. Sistemin başarılı bir şekilde çalışabilmesi için derin öğrenme modellerinin doğru bir şekilde eğitilmesi ve sürekli olarak güncellenmesi önemlidir.

Sistemin nesne tespiti ve sınıflandırması için kullanılan mevcut platformları geliştirebileceği düşünülmektedir. İHA ile nesne ve insan tespit çalışmasına ait ar-ge çalışmaları ve geliştirmeler hâlâ devam etmektedir.

## KAYNAKLAR

- [1] K. Saastamoinen ve T. Taipale, “Short study of unmanned cargo multicopters with simulation”, *Procedia Computer Science*, vol. 176, pp. 3217-3224, 2020.
- [2] F. Nex ve F. Remondino, “UAV for 3D mapping applications: A review”, *Applied Geomatics*, vol. 6(1), pp. 1-15, 2014.
- [3] Y. Li, M. Liu ve D. Jiang “Application of Unmanned Aerial Vehicles in Logistics: A Literature Review”, *Sustainability*, vol. 14(21), pp. 1-18, 2022.
- [4] A. Bouguettaya, H. Zarzour, A. Kechida ve A. M. Taberkit, “Deep learning techniques to classify agricultural crops through UAV imagery: a review”, *Neural Computing and Applications*, vol. 34(12), pp. 9511-9536, 2022.
- [5] N. W. Firmansyah, F. W. Arizal ve J. A. Sudarmanto, “Use of FPV Drones for Sports Documentaries”, *ICADECS International Conference on Art, Design, Education and Cultural Studies (ICADECS)*, 2021, pp. 368-376.
- [6] M. A. Akhloufi, A. Couturier ve N. A. Castro , “Unmanned Aerial Vehicles for Wildland Fires: Sensing, Perception, Cooperation and Assistance”, *Drones*, vol. 5(1), p. 15, 2021.
- [7] C. A. Thiels, J. M. Aho, S. P. Zietlow ve D. H. Jenkins, “Use of unmanned aerial vehicles for medical product transport”, *Air Medical Journal*, vol. 34(2), pp. 104-108, 2015.
- [8] G. Greco, C. Lucianaz, S. Bertoldo ve M. Allegretti, “A solution for monitoring operations in harsh environment: A RFID reader for small UAV”, 2015 International Conference on Electromagnetics in Advanced Applications (ICEAA), 2015, pp. 859-862.
- [9] A. Moffatt, E. Platt, B. Mondragon, A. Kwok, D. Uryeu ve S. Bhandari, “Obstacle detection and avoidance system for small UAVs using a LiDAR”, 2020 International Conference on Unmanned Aircraft Systems (ICUAS), 2020, pp. 633-640.
- [10] E. Schnipke, S. Reidling, J. Meiring, W. Jeffers, M. Hashemi, R. Tan, A. Nemati ve M. Kumar, “Autonomous Navigation of UAV through GPS-Denied Indoor Environment with Obstacles”, *AIAA Infotech at Aerospace*, p. 0715, 2015.
- [11] C. Goerzen, Z. Kong ve B. Mettle, “Survey of Motion Planning Algorithms from the Perspective of Autonomous UAV Guidance”, *Journal of Intelligent and Robotic Systems*, vol. 57(1), pp. 65-100, 2010.
- [12] A. Al-Kaff, D. Martín, F. García, A. Escalera ve J. M. Armingol , “Survey of computer vision algorithms and applications for unmanned aerial vehicles”, *Expert Systems with Applications*, vol. 92, pp. 447-463, 2018.

- [13] D. Cazzato, C. Cimarelli, J. L. Sanchez-Lopez, H. Voos ve M. Leo , “A survey of computer vision methods for 2d object detection from unmanned aerial vehicles”, *Journal of Imaging*, vol. 6(8), pp. 78, 2020.
- [14] Y. Lu, Z. Xue, G. Xia ve L. Zhang, “A survey on vision-based UAV navigation”, *Geo-spatial Information Science*, vol. 21(1), pp. 21-32, 2018.
- [15] H. Choi, M. Geeves, B. Alsalam ve F. Gonzalez, “Open source computer-vision based guidance system for UAVs on-board decision making”, 2016 IEEE Aerospace Conference, 2016, pp. 633-640.
- [16] Y. Akbari, N. Almaadeed, S. Al-maadeed ve O. Elharrouss , “Applications, databases and open computer vision research from drone videos and images: a survey”, *Artificial Intelligence Review*, vol. 54(5), pp. 3887-3938, 2021.
- [17] O. Seveli ve V.G. Başer, “Covid-19 salgınına yönelik zaman serisi verileri ile Prophet model kullanarak makine öğrenmesi temelli vaka tahminlemesi”, *Avrupa Bilim ve Teknoloji Dergisi*, vol.19, pp. 827-835, 2020.
- [18] M. Madijagan ve S. Raj, “Chapter 1 - Parallel Computing, Graphics Processing Unit (GPU) and New Hardware for Deep Learning in Computational Intelligence Research”, In: A.K. Sangaiah (ed) *Deep Learning and Parallel Computing Environment for Bioengineering Systems*, Academic Press, 2019, pp. 1-15.
- [19] N. P. Jouppi, C. Young, N. Patil, D. Patterson, G. Agrawal, R. Bajwa, ..., D. H. Yoon , “In-Datacenter Performance Analysis of a Tensor Processing Unit”, *44th Annual International Symposium on Computer Architecture (ISCA) 2017*, 2017, pp. 1-12.
- [20] J. Choi, Z. Jin ve S. Im, “Implementation of an UAV Real-Time Wireless Communication System Using Wi-Fi”, 2020 International Conference on Information and Communication Technology Convergence (ICTC), 2020, pp. 1855-1859.
- [21] J. Lee, J. Wang, D. Crandall, S. Šabanović ve G. Fox, “Real-Time, Cloud-Based Object Detection for Unmanned Aerial Vehicles”, 2017 First IEEE International Conference on Robotic Computing (IRC), 2017, pp. 36-43.
- [22] Z. Domozi, D. Stojcsics, A. Benhamida, M. Kozlovsky ve Andras Molnar, “Real time object detection for aerial search and rescue missions for missing persons”, 2020 IEEE 15th International Conference of System of Systems Engineering (SoSE), 2020, pp. 519-524.
- [23] M. Radovic, O. Adarkwa ve Q. Wang, “Object Recognition in Aerial Images Using Convolutional Neural Networks”, *Journal of Imaging*, vol. 3(2), pp. 21-30, 2017.
- [24] O. Villi ve M. Yakar, “İnsansız Hava Araçlarının Kullanım Alanları ve Sensör Tipleri”manned Aerial Vehicles, *Türkiye İnsansız Hava Araçları Dergisi*, vol. 4, pp. 73-100, 2022.
- [25] H.N. Lashari, H.M. Ali ve S.U.R. Massan, “Applications of unmanned aerial vehicles: a review”, *3C Tecnologi*, Special Issue on 5th International Conference on Green Computing and Engineering Technologies, pp. 85-105, 2019.

- [26] B. Erdil, “İnsansız Hava Araçlarının Kullanım Alanları ile Bu Araçların Türkiye’nin Yurtdışı Operasyonlarındaki Yeri ve Önemi”, *Journal of Regional Studies*, vol. 5(2), pp. 581 – 607, 2021.
- [27] W.Y.H. Adoni, S. Lorenz, J.S. Fareedh, R. Gloaguen ve M. Bussmann, “Investigation of Autonomous Multi-UAV Systems for Target Detection in Distributed Environment: Current Developments and Open Challenges”, *Drones*, vol. 7(4), pp. 263, 2023.
- [28] Sivil Havacılık Genel Müdürlüğü, “İnsansız hava aracı sistemleri talimatı sht-ih”  
[https://web.shgm.gov.tr/documents/sivilhavacilik/files/mevzuat/sektorel/talimatlar/2020/SHT-IHA\\_Rev-04.pdf](https://web.shgm.gov.tr/documents/sivilhavacilik/files/mevzuat/sektorel/talimatlar/2020/SHT-IHA_Rev-04.pdf) (erişim 13.06.2023).
- [29] Y. Yamazaki, M. Tamaki, C. Premachandra, C. J. Perera, S. Sumathipala ve B. H. Sudantha “Victim Detection Using UAV with On-board Voice Recognition System”, 2019 Third IEEE International Conference on Robotic Computing (IRC), 2019, pp. 555-559.
- [30] C. M. Mesa-Cantillo, I. Alonso-González, M. A. Quintana-Suárez, C. Ley-Bosch, C. Ramírez-Casañas, J. J. Sánchez-Medina ve D. Sanchez-Rodriguez, “A Sound Events Detection and Localization System based on YAMNet Model and BLE Beacons”, The Nineteenth International Conference on Wireless and Mobile Communications (ICWMC), 2023, pp. 1-5.
- [31] S. A. Kumar, S. Sasikala, K. B. Arthiya, J. Sathika ve V. Karishma, “Parkinson’s Speech Detection Using YAMNet”, 2023 2nd International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA), 2023, pp. 1-5.
- [32] A. Tena, F. Clarià ve F. Solsona, “Automated detection of COVID-19 cough”, *Biomedical Signal Processing and Control*, vol. 71, pp. 103175, 2022.
- [33] A. Roy ve U. Satija, “A Novel Melspectrogram Snippet Representation Learning Framework for Severity Detection of Chronic Obstructive Pulmonary Diseases”, *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1-11, 2023.
- [34] P. Gupta, B.Pareek, G. Singal ve D. V. Rao, “Edge device based Military Vehicle Detection and Classification from UAV”, *Multimedia Tools and Applications*, vol. 81(14), pp. 19813-19834, 2021.
- [35] A. Tighkhorshid, Y. Borhani, J. Khoramdell ve E. Najafi , “Facial Mask Wearing Condition Detection Using SSD MobileNetV2”, 2022 12th International Conference on Computer and Knowledge Engineering (ICCKE), 2022, pp. 440-445.
- [36] M. A. E. Latina, J. Van Russel R. Dela Cruz ve F. D. Delos Santos, “Empty Glass Bottle Defect Detection Based on Deep Learning with CNN Using SSD MobileNetV2 Model”, 2022 IEEE 14th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM), 2022, pp. 1-6.
- [37] K. K. Mohammed, E. I.Abd El-Latif, N. E. El-Sayad, A. Darwish ve A. E. Hassanien, “Radio Frequency Fingerprint-based Drone Identification and

Classification using Mel Spectrograms and Pre-trained YAMNet Neural”, *Internet of Things*, vol. 23, p. 100879, 2023.

- [38] M.R. Fairuzi ve F.Y. Zulkifli, “Performance Analysis of YOLOv4 and SSD Mobilenet V2 for Foreign Object Debris (FOD) Detection at Airport Runway Using Custom Dataset”, 2021 17th International Conference on Quality in Research (QIR): International Symposium on Electrical and Computer Engineering, 2021, pp. 11-16.
- [39] F.A. Junior ve S. Suharjito, “Video based oil palm ripeness detection model using deep learning”, *Heliyon*, vol. 9(1), p. e13036, 2023.
- [40] N. Sobahi, O. Atila, E. Deniz, A. Sengur ve U. R. Acharya, “Explainable COVID-19 Detection Using Fractal Dimension and Vision Transformer with Grad-CAM on Cough Sounds”, *Biocybernetics and Biomedical Engineering*, vol. 42(3), pp. 1066-1080, 2022.
- [41] Y. Zhu, X. Liang, J.A. Batsis ve R.M. Roth, “Exploring Deep Transfer Learning Techniques for Alzheimer’s Dementia Detection”, *Frontiers in Computer Science*, vol. 3, p. 624683, 2021.
- [42] M. G. Campana, F. Delmastro, E. Pagani, “Transfer Learning for the Efficient Detection of COVID-19 from Smartphone Audio Data”, *Pervasive and Mobile Computing*, vol. 89, p. 101754, 2023.
- [43] T.K. Nguyen ve K. Nguyen, “Multi-modal approach for COVID-19 detection using coughs and self-reported symptoms”, *Journal of Intelligent & Fuzzy Systems*, vol. 44(3), pp. 3501-3513, 2023.
- [44] Z. S. Syed, M. S. S. Syed, M. Lech ve E. Pirogova, “Automated recognition of Alzheimer’s dementia using bag-of-deep-features and model ensembling”, *IEEE Access*, vol. 9, pp. 88377-88390, 2021.
- [45] J. Mariscal-Harana, V. Alarcón, F. González, J. J. Calvente, , F. J. Pérez-Grau, A. Viguria ve A. Ollero, “Audio-Based Aircraft Detection System for Safe RPAS BVLOS Operations”, *Electronics*, vol. 9, p. 2076, 2020.
- [46] M.F. Al-Sa’d, A. Al-Ali, A. Mohamed, T. Khattab ve A. Erbad, “RF-based drone detection and identification using deep learning approaches: An initiative towards a large open source drone database”, *Future Generation Computer Systems*, vol. 100, pp. 86-97, 2019.
- [47] S. Jeon, J. W. Shin, Y. J. Lee, W. H. Kim, Y. Kwon ve H. Y. Yang, “Empirical study of drone sound detection in real-life environment with deep neural networks”, 2017 25th European Signal Processing Conference (EUSIPCO), 2017, pp. 1858-1862.
- [48] D. Utebayeva, M. Alduraibi, L. Ilibayeva ve Y. Temirgaliyev, “Stacked BiLSTM - CNN for Multiple label UAV sound classification”, 2020 Fourth IEEE International Conference on Robotic Computing (IRC), 2020, pp. 470-474.
- [49] S. Al-Emadi, A. Al-Ali, A. Mohammad ve A. Al-Ali, “Audio Based Drone Detection and Identification using Deep Learning”, 2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC), 2019, pp. 459-464.

- [50] Z. Ameli, Y. Aremanda, W.A. Friess ve E.N. Landis, “Impact of UAV Hardware Options on Bridge Inspection Mission Capabilities” *Drones*, vol. 6(3), p. 64, 2022.
- [51] Z. Ziyang, X. Dongjing ve G. Chen, “Cooperative search-attack mission planning for multi-UAV based on intelligent self-organized algorithm”, *Aerospace Science and Technology*, vol. 77, pp. 402-411, 2018.
- [52] J. Galvez-Serna, F. V. Alvarez, S. Brar, J. Sandino, D. Flannery ve L. F. Gonzalez , “UAV4PE: An Open-Source Framework to Plan UAV Autonomous Missions for Planetary Exploration”, *Drones*, vol. 6(12), p. 391, 2022.
- [53] K. Nguyen ve C. Ha, “Development of Hardware-in-the-Loop Simulation Based on Gazebo and Pixhawk for Unmanned Aerial Vehicles”, *International Journal of Aeronautical and Space Sciences*, vol. 19, pp. 238-249, 2018.
- [54] W. Stecz ve K. Gromada, “UAV Mission Planning with SAR Application”, *Sensors*, vol. 20(4), p. 1080, 2020.
- [55] T. Elmokadem ve A.V. Savkin, “Towards Fully Autonomous UAVs: A Survey”, *Sensors*, vol. 21(18), p. 6223, 2021.
- [56] H. Kim, L. Mokdad ve J. Ben-Othman, "Designing UAV Surveillance Frameworks for Smart City and Extensive Ocean with Differential Perspectives", *IEEE Communications Magazine*, vol. 56(4), pp. 98-104, 2018.
- [57] M.A. Fadhil, “Technology of propulsion system for unmanes combat aerial vehicle (UCAV) – a review, *Jurnal Teknologi Daya Gerak*, vol. 4(2), p. 74-96, 2021.
- [58] S.S, Dol, “Aerodynamic optimization of unmanned aerial vehicle for offshore search and rescue (SAR) operation”, *IOP Conference Series: Materials Science and Engineering*, vol. 715(1), p. 012015, 2020.
- [59] M. McKay, R. Niemiec ve F. Gandhi, “Analysis of Classical and Alternate Hexacopter Configurations with Single Rotor Failure”, *Journal of Aircraft*, vol. 55, pp. 1-8, 2018.
- [60] S. Luo, Z. Zhang, S. Wang, S. Zhang, J. Dai, X. Bu ve J. An, “Network for hypersonic UCAV swarms”, *Science China Information Sciences*, vol. 63(4), p. 140311, 2020.
- [61] Chang, K.-H. ve S.-K. Hung, Design and Implementation of a Tether-Powered Hexacopter for Long Endurance Missions. *Applied Sciences*, vol. 11(24), pp. 11887-11900, 2021.
- [62] N. P. Nguyen, N. X. Mung, Sung Kyung Hong, “Actuator Fault Detection and Fault-Tolerant Control for Hexacopter”, *Sensors*, vol. 19(1), pp. 4721-4741, 2019.
- [63] ArduPilot, “Motor order diagrams.” <https://ardupilot.org/copter/docs/connect-escs-and-motors.html> (erişim 17.05.2023).
- [64] P. Burggräf, A.R. Pérez Martínez, H. Roth ve J. Wagner, “Quadrotors in factory applications: design and implementation of the quadrotor’s P-PID cascade control system Modeling and implementation”, *SN Applied Sciences*, vol. 1, pp. 722-739, 2019.

- [65] M.-A. Stamate “Constructive-functional improvement of hexacopter drones to ensure dynamic stability”, Doctoral Thesis, University POLITEHNICA of Bucharest, Bucharest, 2023.
- [66] B. Divakar, R.S.R.K. Naidu, G. Divya, S. Yugandhar, G. M. Deepak ve D. Sindhusha, “A Review On Brushless Dc Motor Control Techniques”, *Journal of Pharmaceutical Negative Results*, vol. 13(7), pp. 6821-6828, 2023.
- [67] D. Mohanraj, R. Arul David, R. Verma, K. Sathiyasekar, A. B. Barnawi, B. Chokkalingam ve L. Mihet-Popa "A Review of BLDC Motor: State of Art, Advanced Control Techniques, and Applications", *IEEE Access*, vol. 10, pp. 54833-54869, 2022.
- [68] D. L. Gabriel, J. Meyer ve F. du Plessis, "Brushless DC motor characterisation and selection for a fixed wing UAV," *IEEE Africon '11*, 2011, pp. 1-6.
- [69] SunnySky USA, “SunnySky X Series V3 X2216 V3 Brushless Motors Short Shaft Version.” <https://sunnyskyusa.com/collections/x-v3-motors/products/sunnysky-x2216> (erişim 15.07.2023).
- [70] P. Yedamale, “Brushless DC (BLDC) motor Fundamentals”, *Microchip Technology Inc*, vol. 20(1), pp. 3-15, 2003.
- [71] Ö. Alaca, R. Selbaş ve M. Türkkalesi, Fırçasız Motor Sürücülerinin Enerji Verimliliği. *Uluslararası Sürdürülebilir Mühendislik ve Teknoloji Dergisi*, vol. 1(6), p. 1-9, 2021.
- [72] Holybro, “PM07 Power Module.” <https://holybro.com/products/pixhawk-4-power-module-pm07> (erişim 05.05.2023).
- [73] M. F. Ahmed, M. N. Zafar ve J. C. Mohanta, "Modeling and Analysis of Quadcopter F450 Frame," 2020 International Conference on Contemporary Computing and Applications (IC3A), 2020, pp. 196-201.
- [74] F1 Depo, “4s 7000mah 40C Lipo Batarya 14.8V Pil.” <https://www.f1depo.com/urun/4s-7000mah-40c-lipo-batarya-14-8v-pil> (erişim 15.10.2023).
- [75] R. Andoga, M. Schreiner, T. Moravec, L. Fözö ve M. Schrötter, "Automatic decision making process in a small unmanned airplane," 2018 IEEE 18th International Symposium on Computational Intelligence and Informatics (CINTI), 2018, pp. 000301-000306.
- [76] PX4 Autopilot, “Pixhawk 4.” [https://docs.px4.io/v1.12/en/flight\\_controller/pixhawk4.html](https://docs.px4.io/v1.12/en/flight_controller/pixhawk4.html) (erişim 18.06.2023).
- [77] Banggood, “RACERSTAR RS40A V2 40A BLHELI\_S BB2 OPTO 2-5S BRUSHLESS ESC.” [https://tr-m.banggood.com/Racerstar-RS40A-V2-40A-BLHELI\\_S-BB2-OPTO-2-5S-Brushless-ESC-Dshot-600-Ready-for-RC-Drone-FPV-Racing-p-1137033.html](https://tr-m.banggood.com/Racerstar-RS40A-V2-40A-BLHELI_S-BB2-OPTO-2-5S-Brushless-ESC-Dshot-600-Ready-for-RC-Drone-FPV-Racing-p-1137033.html) (erişim 21.07.2023)
- [78] Team NCRC, “AT10II 2.4G Uçak Kumandası 12CH Kumanda ve R12DS Alicili.” <https://www.team-ncrc.com/magaza/urun/at10ii-24g-ucak-kumandasi-12ch-kumanda-ve-r12ds-alicili-turuncu-renk> (erişim 18.04.2023).
- [79] Holybro, “Sik Telemetry Radio V3.” <https://holybro.com/products/sik-telemetry-radio-v3> (erişim 12.05.2023).

- [80] T.Nasution, I. Siregar ve M. Yasir, "UAV telemetry communications using ZigBee protocol", *Journal of Physics: Conference Series*, 2017, vol. 914, p. 012001.
- [81] N.Ashby, "Relativity in the Global Positioning System", *Living Reviews in Relativity*, vol. 6(1), pp. 41-47, 2003.
- [82] Aliexpress, "Holybro Pixhawk 4 M8n Gps Modülü." <https://tr.aliexpress.com/i/33031659488.html> (erişim 10.07.2023).
- [83] H. Hastedt, T. Ekkel ve T. Luhmann, "Evaluation of the Quality of Action Cameras with Wide-Angle Lenses in Uav Photogrammetry", *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLI-B1, pp. 851-859, 2016.
- [84] Hepsiburada, "Sjcam Sj6 Legend 4K Aksiyon Kamerası." <https://www.hepsiburada.com/sjcam-sj6-legend-4k-aksiyon-kamerasi-siyah-pm-HBC000017AV5Z> (erişim 11.10.2023).
- [85] Robuin, "Tarot GoPro Gimbal T-3D V Metal Brushless Gimbal." <https://robu.in/product/tarot-gopro-gimbal-t-3d-v-metal-brushless-gimbal-3-axis/> (erişim 15.09.2023).
- [86] Amazon, "Lipo Battery Voltage Tester Low Voltage Buzzer." <https://www.amazon.com/Wishiot-Battery-Voltage-Checker-Indicator/dp/B089W64GKR?th=1> (erişim 11.10.2023).
- [87] ABRA Electronics, "Eachine TX5258 5.8GHz, 72CH Switchable FPV Transmitter." <https://abra-electronics.com/quadcopters/first-person-video/d-e-t-01.html> (erişim 24.09.2023)
- [88] P.A.Nagmote, P.S. Kadu ve S.D. Munghate, "Analysis of Drone Frame", *Journal of Emerging Technologies and Innovative Research (JETIR)*, vol. 7(6), pp. 783-789, 2020.
- [89] Motorobit, "F550 Hexacopter Frame Gövdesi." <https://www.motorobit.com/f550-hexacopter-frame-govdesi> (erişim 19.07.2023).
- [90] Motorobit, "F450 ve F550 Drone İniş Takımı." <https://www.motorobit.com/f450-ve-f550-drone-inis-takimi> (erişim 20.05.2023).
- [91] Dronemarket, "1045 Karbon CW/CCW Karbon Pervane Seti." <https://www.dronemarket.com/urun/1045-karbon-katkili-cw-ccw-pervane-seti-carbon-propeller-kit> (erişim 10.06.2023).
- [92] R. Citroni, F. Di Paolo ve P. Livreri, "A Novel Energy Harvester for Powering Small UAVs: Performance Analysis, Model Validation and Flight Results", *Sensors*, vol. 19(8), pp. 1771-1792, 2019.
- [93] F.D.P. Koffi, "Ground target tracking unmanned aerial vehicle (UAV)", Master's Thesis, Aydın Adnan Menderes University, Aydın, 2022.
- [94] E. Castelão Tetila, B. Brandoli Machado, N. A. Belete, D. A. Guimarães ve H. Pistori, "Identification of Soybean Foliar Diseases Using Unmanned Aerial Vehicle Images", *IEEE Geoscience and Remote Sensing Letters*, vol. 14(12), pp. 2190-2194, 2017.
- [95] Y. Liu, Q. Wang, H. Hu ve Y. He, "A Novel Real-Time Moving Target Tracking and Path Planning System for a Quadrotor UAV in Unknown

- Unstructured Outdoor Scenes”, *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49(11), pp. 2362-2372, 2019.
- [96] R. Y. Choi, A. S. Coyner, J. Kalpathy-Cramer, M. F. Chiang ve J. P. Campbell, “Introduction to machine learning, neural networks, and deep learning”, *Translational vision science & technology*, vol. 9(2), pp. 2-14, 2020.
- [97] D. Kinaneva, G. Hristov, J. Raychev ve P. Zahariev, "Early Forest Fire Detection Using Drones and Artificial Intelligence," 2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2019, pp. 1060-1065.
- [98] M. M. Mijwil, A. Esen ve A. Alsaadi, “Overview of neural networks”, *Computer Engineering Techniques Department*, vol. 1, pp. 1-3, 2019.
- [99] U. Afzal ve T. Mahmood, "Using predictive analytics to forecast drone attacks in Pakistan," 2013 5th International Conference on Information and Communication Technologies, 2013, pp. 1-6.
- [100] A. Zayegh ve N. Al Bassam, “Neural network principles and applications”, In: V. Asadpour (ed) *Digital systems*, Intechopen, 2018, pp. 115-131.
- [101] S.Sharma, S. Sharma ve A. Athaiya, “Activation functions in neural networks”, *International Journal of Engineering Applied Sciences and Technology*, vol. 6(12), pp. 310-316, 2020.
- [102] R.M. Tharsanee, R.S. Soundariya, A. S. Kumar, M. Karthiga ve S. Sountharajan, “7 - Deep convolutional neural network–based image classification for COVID-19 diagnosis”, In: U. Kose, D. Gupta, V. H. Costa de Albuquerque, A. Khanna (eds), *Data Science for COVID-19*, Academic Press, 2021, pp. 117-145.
- [103] D. Nie ve X. Wen, “Chapter 6 - Convolutional neural networks for image synthesis”, In: N. Burgos ve D. Svoboda (eds), *Biomedical Image Synthesis and Simulation*, Academic Press, 2022, pp. 91-104.
- [104] I. El Naqa ve M.J. Murphy, “What Is Machine Learning?”, In: I. El Naqa, R. Li, M. Murphy (eds), *Machine Learning in Radiation Oncology*, Springer, 2015, pp. 3-11.
- [105] C. Janiesch, P. Zschech ve K. Heinrich, “Machine learning and deep learning”, *Electronic Markets*, vol. 31(3), pp. 685-695. 2021
- [106] W. Xu, “Efficient Distributed Image Recognition Algorithm of Deep Learning Framework TensorFlow”, *Journal of Physics: Conference Series*, vol. 2066(1), p. 012070, 2021.
- [107] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu ve F. E. Alsaadi, “A survey of deep neural network architectures and their applications”, *Neurocomputing*, vol. 234, pp. 11-26, 2017.
- [108] R. Zhao, R. Yan, Z. Chen, K. Mao, P. Wang ve R. X. Gao, “Deep learning and its applications to machine health monitoring”, *Mechanical Systems and Signal Processing*, vol. 115, pp. 213-237, 2019.
- [109] H.Y. Khdir, W.M. Jasim ve S.A. Aliesawi, “Deep Learning Algorithms based Voiceprint Recognition System in Noisy Environment”, *Journal of Physics: Conference Series*, vol. 1804(1), p. 012042, 2021.

- [110] J. -Y. Kim ve S. -B. Cho, "Evolutionary Optimization of Hyperparameters in Deep Learning Models," 2019 IEEE Congress on Evolutionary Computation (CEC), 2019, pp. 831-837.
- [111] R. Andonie, "Hyperparameter optimization in learning systems", *Journal of Membrane Computing*, vol. 1(4), p. 279-291, 2019.
- [112] Y. Quiñonez, C. Lizarraga, J. Peraza ve O. Zatarain, "Image recognition in UAV videos using convolutional neural networks", *IET Software*, vol. 14, pp. 176-181, 2020.
- [113] V. H. Phung ve E. J. Rhee, "A High-Accuracy Model Average Ensemble of Convolutional Neural Networks for Classification of Cloud Image Patches on Small Datasets", *Applied Sciences*, vol. 9, p. 4500, 2019.
- [114] V.H. Phung ve E.J. Rhee, "A deep learning approach for classification of cloud image patches on small datasets", *Journal of Information and Communication Convergence Engineering*, vol. 16: p. 173-178, 2018.
- [115] R. G. Mantovani, A. L. D. Rossi, J. Vanschoren, B. Bischl ve A. C. P. L. F. Carvalho, "To tune or not to tune: Recommending when to adjust SVM hyperparameters via meta-learning", 2015 International Joint Conference on Neural Networks (IJCNN), 2015, pp. 1-8.
- [116] P. Probst, B. Bischl ve A.-L. Boulesteix, "Tunability: Importance of Hyperparameters of Machine Learning Algorithms", *Journal of Machine Learning Research*, vol. 20, pp. 1-32, 2018.
- [117] S. Nikbakht, C. Anitescu ve T. Rabczuk, "Optimizing the neural network hyperparameters utilizing genetic algorithm", *Journal of Zhejiang University-Science A*, vol. 22(6), pp. 407-426, 2021.
- [118] T. Szandała, "Review and Comparison of Commonly Used Activation Functions for Deep Neural Networks", In: A. Bhoi, P. Mallick, C. M. Liu, V. Balas (eds) *Bio-inspired Neurocomputing*, Springer, 2021, pp. 203-224.
- [119] Z. Teng, S. Teng, J. Zhang, G. Chen ve F. Cui "Structural Damage Detection Based on Real-Time Vibration Signal and Convolutional Neural Network", *Applied Sciences*, vol. 10, p. 4720, 2020.
- [120] L. Alzubaidi, J. Zhang, , A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M.A Fadhel, M. Al-Amidie ve L. Farhan, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions", *Journal of Big Data*, vol. 8(1), p. 53, 2021
- [121] I. Cardoza, J. P. García-Vázquez, A. Díaz-Ramírez ve V. Quintero-Rosas, "Convolutional Neural Networks Hyperparameter Tuning for Classifying Firearms on Images", *Applied Artificial Intelligence*, vol. 36(1), p. 2058165, 2022.
- [122] E. Bochinski, T. Senst ve T. Sikora, "Hyper-parameter optimization for convolutional neural network committees based on evolutionary algorithms," 2017 IEEE International Conference on Image Processing (ICIP), 2017, pp. 3924-3928.
- [123] M. Feurer ve F. Hutter, "Hyperparameter Optimization", In: F. Hutter, L. Kotthoff, J. Vanschoren (eds), *Automated Machine Learning*, Springer, 2019, p. 3-33.

- [124] A.L. Duca, "What are steps, epochs, and batch size in Deep Learning." <https://medium.com/syntaxerrorpub/what-are-steps-epochs-and-batch-size-in-deep-learning-5c942539a5f8> (erişim 11.08.2023)
- [125] Z. Li, F. Liu, W. Yang, S. Peng ve J. Zhou, "A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects," in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33(12), pp. 6999-7019, 2022.
- [126] S.S. Moumgiakmas, G.G. Samatas ve G.A. Papakostas, "Computer vision for fire detection on UAVs-From software to hardware", *Future Internet*, vol. 13(8), p. 200, 2021.
- [127] L.M. Belmonte, R. Morales ve A. Fernández-Caballero, "Computer vision in autonomous unmanned aerial vehicles-a systematic mapping study", *Applied Sciences*, vol. 9(15), p. 3196, 2019.
- [128] A. Bouguettaya, H. Zarzour, A. Kechida ve A. M. Taberkit, "A review on early wildfire detection from unmanned aerial vehicles using deep learning-based computer vision algorithms", *Signal Processing*, vol. 190, p. 108309. 2022
- [129] A. Bouguettaya, H. Zarzour, A. Kechida ve A. M. Taberkit, "Vehicle detection from UAV imagery with deep learning: A review", *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33(11), p. 6047-6067, 2021.
- [130] D.C Tsouros, S. Bibi ve P.G. Sarigiannidis, A review on UAV-based applications for precision agriculture. *Information*, vol. 10(11), p. 349, 2019.
- [131] K. P. Seng, L. -M. Ang, L. M. Schmidtke ve S. Y. Rogiers, "Computer Vision and Machine Learning for Viticulture Technology," in *IEEE Access*, vol. 6, pp. 67494-67510, 2018.
- [132] A. Khaloo, D. Lattanzi, A. Jachimowicz ve Charles Devaney, "Utilizing UAV and 3D computer vision for visual inspection of a large gravity dam", *Frontiers in Built Environment*, vol. 4, p. 31, 2018.
- [133] C. Donmez, O. Villi, S. Berberoglu ve A. Cilek, "Computer vision-based citrus tree detection in a cultivated environment using UAV imagery", *Computers and Electronics in Agriculture*, vol. 187, p. 106273, 2021.
- [134] Evan, M. Wulandari ve E. Syamsudin, "Recognition of Pedestrian Traffic Light using Tensorflow and SSD MobileNet V2", *IOP Conference Series: Materials Science and Engineering*, vol. 1007(1), p. 012022, 2020.
- [135] C. Li, X. Sun ve J. Cai, "Intelligent Mobile Drone System Based on Real-Time Object Detection", *Journal on Artificial Intelligence*, vol. 1(1): p. 1-8, 2019.
- [136] K. He, G. Gkioxari, P. Dollár ve R. Girshick, "Mask R-CNN", 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2980-2988.
- [137] J. Redmon ve A. Farhadi, "Yolov3: An incremental improvement", 2018, arXiv:1804.02767.
- [138] Z. Cai ve N. Vasconcelos, "Cascade R-CNN: Delving Into High Quality Object Detection", 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 6154-6162.
- [139] J. Terven, D. M. Cordova-Esparza, A. Ramirez-Pedraza ve E. A. Chavez-Urbiola, "Loss Functions and Metrics in Deep Learning", 2023, arXiv:2307.02694.

- [140] M. A. Arserim ve A. Usta, “İnsansız hava aracından çekilen videolar kullanılarak derin öğrenme yaklaşımı ile nesne tespiti”, *Dicle University Journal of Engineering*, vol. 14(1), pp. 9-15, 2023.
- [141] A. Koirala, K. B. Walsh, Z. Wang ve C. McCarthy, “Deep learning – Method overview and review of use for fruit detection and yield estimation”, *Computers and Electronics in Agriculture*, vol. 162, p. 219-234, 2019.
- [142] H. Zhu, H. Wei, B. Li, X. Yuan ve N. Kehtarnavaz, “A Review of Video Object Detection: Datasets, Metrics and Methods”, *Applied Sciences*, vol. 10(21), p. 7834, 2020.
- [143] R. Padilla, S. L. Netto ve E. A. B. da Silva, "A Survey on Performance Metrics for Object-Detection Algorithms," 2020 International Conference on Systems, Signals and Image Processing (IWSSIP), 2020, pp. 237-242.
- [144] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto ve H. Adam, “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications”, 2017, arXiv:1704.04861.
- [145] M. Konaiteet, P. A. Owolawi, T. Mapayi, V. Malele, K. Odeyemi, G. Aiyetoro ve J. S. Ojo, “Smart Hat for the blind with Real-Time Object Detection using Raspberry Pi and TensorFlow Lite”, *icARTi '21: International Conference on Artificial Intelligence and its Applications*, 2021, pp. 1-6.
- [146] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu ve A. C. Berg, “Ssd: Single shot multibox detector”, *Computer Vision–ECCV 2016*, 2016, pp. 21-37.
- [147] T. -Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. ve C. L. Zitnick, “Microsoft COCO: Common Objects in Context”, *Computer Vision – ECCV 2014*, 2014, pp. 740-755.
- [148] K. Dong, C. Zhou, Y. Ruan ve Y. Li, "MobileNetV2 Model for Image Classification," 2020 2nd International Conference on Information Technology and Computer Application (ITCA), 2020, pp. 476-480.
- [149] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov ve L. -C. Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks”, 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 4510-4520.
- [150] K. O’Shea ve R. Nash, “An introduction to convolutional neural networks”, 2015, arXiv:1511.08458.
- [151] M. Xin ve Y. Wang, “Research on image classification model based on deep convolution neural network”, *EURASIP Journal on Image and Video Processing*, pp. 1-13, 2019.
- [152] T. Ganegedara, *Natural Language Processing with TensorFlow: Teach language to machines using Python’s deep learning library*, Packt Publishing Ltd, 2018.
- [153] A. Sener, B. Bergen ve M. Togacar, “Fault Detection from Images of Railroad Lines Using the Deep Learning Model Built with the Tensorflow Library”, *Turkish Journal of Science and Technology*, vol. 17(1), pp. 47-53, 2022.
- [154] O. Kembuan, G. Caren Rorimpandey ve S. Milian Tompunu Tengker, "Convolutional Neural Network (CNN) for Image Classification of Indonesia

- Sign Language Using Tensorflow," 2020 2nd International Conference on Cybernetics and Intelligent System (ICORIS), 2020, pp. 1-5.
- [155] T. V. Janahiraman ve M. S. M. Subuhan, "Traffic Light Detection Using Tensorflow Object Detection Framework", 2019 IEEE 9th International Conference on System Engineering and Technology (ICSET), 2019, pp. 108-113.
- [156] S. Srivastava, A. Gangwar, R. Mishra ve Sudhakar Singh, "Sign language recognition system using TensorFlow object detection API", 2021, arXiv:2201.01486.
- [157] F. Al-Azooa, A.M. Taqia ve M. Milanovab, "Human related-health actions detection using Android Camera based on TensorFlow Object Detection API", *International Journal of Advanced Computer Science and Applications*, vol. 9(10), 2018.
- [158] R. Phadnis, J. Mishra ve S. Bendale, "Objects Talk - Object Detection and Pattern Tracking Using TensorFlow," 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), 2018, pp. 1216-1219.
- [159] D. Galvez, G. Damos, J. Ciro, J. F. Cerón, K. Achorn, A. Gopi, ..., V. J. Reddi, "The People's Speech: A Large-Scale Diverse English Speech Recognition Dataset for Commercial Usage", 2021, arXiv:2111.09344.
- [160] Bai, J. ve J. Fei, "Research and Implementation of Handwritten Numbers Recognition System Based on Neural Network and Tensorflow Framework", *Journal of Physics: Conference Series*, vol. 1576(1), p. 012029, 2020.
- [161] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, ..., X. Zheng, "TensorFlow : Large-Scale Machine Learning on Heterogeneous Distributed Systems", 2015, arXiv:1603.04467.
- [162] TensorFlow, "Model conversion overview." <https://www.tensorflow.org/lite/models/convert?hl=en> (erişim 17.07.2023).
- [163] J. A. L. Marques, F. N. B. Gois, J. P. do Vale Madeiro, T. Li ve S. J. Fong, Chapter 4 - Artificial neural network-based approaches for computer aided disease diagnosis and treatment, In: A. K. Bhoi, V. H. C. de Albuquerque, P. N. Srinivasu, G. Marques (eds) *Cognitive and Soft Computing Techniques for the Analysis of Healthcare Data*, Academic Press, 2022, pp. 79-99.
- [164] Wikipedia, "Boeing AH-64 Apache", [https://en.wikipedia.org/w/index.php?title=Boeing\\_AH-64\\_Apache&oldid=1186132996](https://en.wikipedia.org/w/index.php?title=Boeing_AH-64_Apache&oldid=1186132996) (erişim 24.10.2023).
- [165] D. Xiao, F. Shan, Z. Li, B. T. Le, X. Liu ve X. Li, "A Target Detection Model Based on Improved Tiny-Yolov3 Under the Environment of Mining Truck", *IEEE Access*, vol. 7, pp. 123757-123764, 2019.
- [166] J. Bergstra, D. Yamins ve D. Cox, "Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures", *Proceedings of Machine Learning Research*, vol.28(1) , pp. 115-123, 2013.

- [167] E. Bisong, "Google Colaboratory", In: E. Bisong (ed) *Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners*, Apress: Berkeley, 2019, pp. 59-64.
- [168] J.I. Tamir, S.X. Yu, ve M. Lustig, "DeepInPy: Deep Inverse Problems in Python", ISMRM Workshop on Data Sampling and Image Reconstruction, 2020.
- [169] R. Yamashita, M. Nishio, R. K. Gian Do ve K. Togashi, "Convolutional neural networks: an overview and application in radiology. Insights into Imaging", vol. 9(4), pp. 611-629, 2018.
- [170] S. Kılıçarslan, K. Adem ve M. Celik, "An overview of the activation functions used in deep learning algorithms", *Journal of New Results in Science*, vol. 10, pp. 75-88, 2021.
- [171] D.C. Marcu ve C. Grava, "The impact of activation functions on training and performance of a deep neural network", 2021 16th International Conference on Engineering of Modern Electric Systems (EMES), 2021, pp. 1-4.
- [172] B. Ding, H. Qian ve J. Zhou, "Activation functions and their characteristics in deep neural networks", 2018 Chinese Control And Decision Conference (CCDC), 2018, pp. 1836-1841.
- [173] R. Abdulkadirov, P. Lyakhov ve N. Nagornov, "Survey of Optimization Algorithms in Modern Neural Networks", *Mathematics*, vol. 11(11), p. 2466, 2023.
- [174] C. Fjellström ve K. Nyström, "Deep learning, stochastic gradient descent and diffusion maps", *Journal of Computational Mathematics and Data Science*, vol. 4, p. 100054, 2022.
- [175] J. Fu, B. Wang, H. Zhang, Z. Zhang, W. Chen ve N. Zheng, "When and Why Momentum Accelerates SGD: An Empirical Study", 2023, arXiv:2306.09000.
- [176] Great Learning Team, "Understanding Learning Rate in Machine Learning.", <https://www.mygreatlearning.com/blog/understanding-learning-rate-in-machine-learning/> (erişim 12.07.2023)
- [177] H. Zhao, O. Gallo, I. Frosio ve J. Kautz, "Loss Functions for Image Restoration With Neural Networks," *IEEE Transactions on Computational Imaging*, vol. 3, no. 1, pp. 47-57, 2017.
- [178] H. Purwins, B. Li, T. Virtanen, J. Schlüter, S. -Y. Chang ve T. Sainath, "Deep Learning for Audio Signal Processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13(2), pp. 206-219, 2019.
- [179] S Shuvaev, H. Giaffar ve A. A. Koulakov, "Representations of sound in deep learning of audio features from music", 2017, arXiv:1712.02898.
- [180] W. Chen, Q. Sun, X. Chen ve G. Xie, "Deep learning methods for heart sounds classification: A systematic review", *Entropy*, vol. 23(6), p. 667, 2021.
- [181] Y. Yang ve Y. Yue, English speech sound improvement system based on deep learning from signal processing to semantic recognition. *International Journal of Speech Technology*, vol. 23, pp. 505-515, 2020.
- [182] W. Xiong, X. Xu, L. Chen ve J. Yang, "Sound-based construction activity monitoring with deep learning", *Buildings*, vol. 12(11), p. 1947, 2022.

- [183] A. Mehrish, N. Majumder, R. Bharadwaj, R. Mihalcea ve S. Poria, "A review of deep learning techniques for speech processing", *Information Fusion*, p. 101869, 2023.
- [184] T. N. Sainath, R. J. Weiss, K. W. Wilson, B. Li, A. Narayanan, E. Variiani, ..., C. Kim, "Multichannel Signal Processing With Deep Neural Networks for Automatic Speech Recognition," in *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25(5), pp. 965-979, 2017.
- [185] A. Bose ve B. Tripathy, "Deep learning for audio signal classification", In: S. Bhattacharyya, V. Snasel, A. E. Hassanien, S. Saha, B. K. Tripathy (eds) *Deep learning research and applications*, De Gruyter, 2020, p. 105-136.
- [186] Dhiraj, R. Biswas ve N. Ghattamaraju, "An effective analysis of deep learning based approaches for audio based feature extraction and its visualization", *Multimedia Tool and Applications*, vol. 78, pp. 23949-23972, 2019.
- [187] A. Aggarwal, A. Srivastava, A. Agarwal, N. Chahal, D. Singh, A. A. Alnuaim, A. Alhadlaq, H. -N Lee, "Two-way feature extraction for speech emotion recognition using deep learning", *Sensors*, vol. 22(6), p. 2378, 2022.
- [188] E. Babae, N. Anuar, A. Wahid, S. Band ve A. Chronopoulos, "An overview of audio event detection methods from feature extraction to classification", *Applied Artificial Intelligence*, vol. 31(9-10), pp. 661-714, 2017.
- [189] D. Wu, "An Audio Classification Approach Based on Machine Learning," 2019 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS), 2019, pp. 626-629.
- [190] T. Grill ve J. Schlüter, "Two convolutional neural networks for bird detection in audio signals," 2017 25th European Signal Processing Conference (EUSIPCO), 2017, pp. 1764-1768.
- [191] S. Becker, J. Vielhaben, M. Ackermann, K. -R. Müller, S. Lapuschkin ve W. Samek, "Interpreting and explaining deep neural networks for classification of audio signals", 2018, arXiv:1807.03418.
- [192] H.-H. Wang, J.-M. Liu, M. You ve G.-Z. Li, "Audio signals encoding for cough classification using convolutional neural networks: A comparative study," 2015 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), 2015, pp. 442-445.
- [193] S. Abdoli, P. Cardinal ve A.L. Koerich, "End-to-end environmental sound classification using a 1D convolutional neural network", *Expert Systems with Applications*, vol. 136, pp. 252-263, 2019.
- [194] K. J. Piczak, "Environmental sound classification with convolutional neural networks," 2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP), 2015, pp. 1-6.
- [195] I. Kuzminykh, D. Shevchuk, S. Shiaeles ve B. Ghita, "Audio Interval Retrieval Using Convolutional Neural Networks", *Internet of Things, Smart Spaces, and Next Generation Networks and Systems*, pp. 229-240, 2020.
- [196] Google, "TensorFlow YAMNet Model for Audio Event Classification." <https://tfhub.dev/google/yamnet/1> (erişim 23.10.2023).

- [197] Natsiou, A. ve S. O’Leary, “Audio representations for deep learning in sound synthesis: A review”, 2021 IEEE/ACS 18th International Conference on Computer Systems and Applications (AICCSA), 2021, p. 1-8.
- [198] G. Z. Felipe, Y. Maldonado, G. d. Costa ve L. G. Helal, "Acoustic scene classification using spectrograms", 2017 36th International Conference of the Chilean Computer Science Society (SCCC), 2017, pp. 1-7.
- [199] T.L. Nwe, T.H. Dat ve B. Ma, “Convolutional neural network with multi-task learning scheme for acoustic scene classification”, 2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), 2017, pp. 1347-1350.
- [200] Y. Zeng, H. Mao, D. Peng ve Z. Yi, “Spectrogram based multi-task audio classification”, *Multimedia Tools and Applications*, vol. 78, pp. 3705–3722, 2019.
- [201] E. Brusa, C. Delprete ve L.G. Di Maggio, “Deep Transfer Learning for Machine Diagnosis: From Sound and Music Recognition to Bearing Fault Detection”, *Applied Sciences*, vol. 11(24), p. 11663, 2021.
- [202] A. Ahmed, Y. Serrestou, K. Raouf ve J. -F. Diouris, “Empirical Mode Decomposition-Based Feature Extraction for Environmental Sound Classification”, *Sensors*, vol. 22(20), p. 7717, 2022.
- [203] I. Papadimitriou, A. Vafeiadis, A. Lalas, K. Votis ve D. Tzovaras, “Audio-Based Event Detection at Different SNR Settings Using Two-Dimensional Spectrogram Magnitude Representations”, *Electronics*, Vol. 9(10), p. 1593. 2020.
- [204] F. Zhang, L. Zhang, H. Chen ve J. Xie, “Bird Species Identification Using Spectrogram Based on Multi-Channel Fusion of DCNNs”, *Entropy*, vol. 23(11), p. 1507, 2021.
- [205] T. Yin, “Music Track Recommendation Using Deep-CNN and Mel Spectrograms”, *Mobile Networks and Applications*, vol.28, pp. 1-8, 2023.
- [206] Y. Kumar, S. Gupta ve W. Singh, “A novel deep transfer learning models for recognition of birds sounds in different environment”, *Soft Computing*, vol. 26(3), p. 1003-1023, 2022.
- [207] Z. K. Abdul ve A. K. Al-Talabani, "Mel Frequency Cepstral Coefficient and its Applications: A Review," in *IEEE Access*, vol. 10, pp. 122136-122158, 2022.
- [208] T. Toshniwal, P. Tandon ve N. P. "Music Genre Recognition Using Short Time Fourier Transform and CNN," 2022 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 2022, pp. 1-4.
- [209] M. Ashouri, F.F.D. Silva ve C.L. Bak, “Application of short-time Fourier transform for harmonic-based protection of meshed VSC-MTDC grids”, *The Journal of Engineering*, vol. 2019(16), p. 1439-1443, 2019.
- [210] H. Jeon, Y. Jung, S. Lee ve Y. Jung , “Area-Efficient Short-Time Fourier Transform Processor for Time–Frequency Analysis of Non-Stationary Signals”, *Applied Sciences*, vol. 10, p. 7208, 2020.
- [211] Wikipedia, “Short-time Fourier transform.” [https://en.wikipedia.org/wiki/Short-time\\_Fourier\\_transform](https://en.wikipedia.org/wiki/Short-time_Fourier_transform) (erişim 31.10.2023).

- [212] Siemens, "Window Types: Hanning, Flattop, Uniform, Tukey, and Exponential." <https://community.sw.siemens.com/s/article/window-types-hanning-flattop-uniform-tukey-and-exponential> (erişim 31.10.2023).
- [213] Wikipedia, "W. Short-time Fourier transform." [https://en.wikipedia.org/w/index.php?title=Short-time\\_Fourier\\_transform&oldid=1115436538](https://en.wikipedia.org/w/index.php?title=Short-time_Fourier_transform&oldid=1115436538) (erişim 20.11.2023).
- [214] H. Xu, D. Kong, Y. Qian ve X. Tang, "Motor noise reduction of unmanned aerial vehicles", *Applied Acoustics*, vol. 198, p. 108979, 2022.
- [215] I. Ibrahim ve D. A. Lowther, "A Study of the Relationship between Acoustic Noise and Torque Pulsation in Permanent Magnet Synchronous Motors," 2022 IEEE 20th Biennial Conference on Electromagnetic Field Computation (CEFC), 2022, pp. 1-2.
- [216] A. Saxena ve B. G. Fernandes, "Noise and cogging torque reduction in brushless DC ceiling fan," 2015 18th International Conference on Electrical Machines and Systems (ICEMS), 2015, pp. 1334-1338.
- [217] K. Xia, Z. Li, J. Lu, B. Dong ve C. Bi, "Acoustic noise of brushless DC motors induced by electromagnetic torque ripple", *Journal of Power Electronics*, vol. 17, pp. 963-971, 2017.
- [218] J. Le Besnerais, Q. Souron ve E. Devillers, "Analysis of the electromagnetic acoustic noise and vibrations of a high-speed brushless DC motor," 8th IET International Conference on Power Electronics, Machines and Drives (PEMD 2016), UK, 2016, pp. 1-10.
- [219] R. M. Pindoriya, A. K. Mishra, B. S. Rajpurohit ve R. Kumar, "An Analysis of Vibration and Acoustic Noise of BLDC Motor Drive," 2018 IEEE Power & Energy Society General Meeting (PESGM), 2018, pp. 1-5.
- [220] K.-S. Kim, C.-M. Lee, G.-Y. Hwang ve S.-M. Hwang, Effect of the number of poles on the acoustic noise from BLDC motors. *Journal of Mechanical Science and Technology*, vol. 25(2), pp. 273-277, 2011.
- [221] L. Jianjun, X. Yongxiang ve Z. Jibin, "A study on the reduction of vibration and acoustic noise for brushless DC motor," 2008 International Conference on Electrical Machines and Systems, Wuhan, 2008, pp. 561-563.
- [222] T.-S. Jeong, G.-W. Cho ve G.-T. Kim, "The Design of Radial Magnetic Force Equilibrium for Reduction of Vibration and Noise in IPM Type BLDC Motor", *The Transactions of The Korean Institute of Electrical Engineers*, vol. 62(11), pp.1535-1540, 2013.
- [223] A. Louros, M. Papoutsidakis, A. Chatzopoulos ve C. Drosos, Design of an Innovative Flight Controller for Quadcopter Robust Handling, *Journal of Multidisciplinary Engineering Science and Technology (JMEST)*, p. 2458-9403, 2018.
- [224] H. Bu, H. Wu, C. Bertin, Y. Fang ve S. Zhong "Aerodynamic and acoustic measurements of dual small-scale propellers", *Journal of Sound and Vibration*, vol. 511, pp. 116330, 2021.
- [225] R. S. McKay, M. J. Kingan, S. T. Go ve R. Jung, "Experimental and analytical investigation of contra-rotating multi-rotor UAV propeller noise", *Applied Acoustics*, vol. 177, p. 107850, 2021.

- [226] C. Ramos-Romero, N. Green, A. J. Torija ve J. Asensio, “On-field noise measurements and acoustic characterisation of multi-rotor small unmanned aerial systems”, *Aerospace Science and Technology*, vol. 141, p. 108537, 2023.
- [227] G. Sinibaldi ve L. Marino, “Experimental analysis on the noise of propellers for small UAV”, *Applied Acoustics*, vol. 74(1), p. 79-88, 2013.
- [228] H. Wu, H. Jiang, P. Zhou, S. Zhong, X. Zhang, G. Zhou ve B. Chen, “On identifying the deterministic components of propeller noise”, *Aerospace Science and Technology*, vol. 130, p. 107948, 2022.
- [229] R.J. Fattah, W. Chen, H. Wu ve Y. Wu, “Noise measurements of generic small-scale propellers”, 25th AIAA/CEAS Aeroacoustics Conference, 2019, p.2498.
- [230] S. Glegg ve W. Devenport, “Open rotor noise”, in”, In: S. Glegg and W. Devenport (eds) *Aeroacoustics of Low Mach Number Flows*, Academic Press, 2017, pp. 399-436.
- [231] P. Podder, M. Hasan, R. Islam ve M. Sayeed, “Design and Implementation of Butterworth, Chebyshev-I and Elliptic Filter for Speech Signal Analysis”, *International Journal of Computer Applications*, vol. 98(7), p. 12–18, 2014.
- [232] The Open University, “Designing a digital filter in the frequency domain.” <https://www.open.edu/openlearn/science-maths-technology/electronic-applications/content-section-3.6> (erişim 10.12.2023).
- [233] M. Turnip, A. Dharma, Andrian, A. Afriansyah, A. Oktarino ve A. Turnip, “Integration of FIR and Butterworth Algorithm for Real-Time Extraction of Recorded ECG Signals”, In: E. Joeliante, A. Turnip, A. W. Widyotriatmo (eds) *Cyber Physical, Computer and Automation System: A Study of New Technologies*, Springer International Publishing, 2021, pp. 49-58.
- [234] A. A. Fime, M. Ashikuzzaman ve A. Aziz, “Audio signal based danger detection using signal processing and deep learning”, *Expert Systems with Applications*, vol. 237, p.121646, 2023.
- [235] E. Tsalera, A. Papadakis ve M. Samarakou, “Comparison of Pre-Trained CNNs for Audio Classification Using Transfer Learning”, *Journal of Sensor and Actuator Networks*, vol. 10(4), p. 72, 2021.
- [236] K. J. Piczak, “ESC: Dataset for Environmental Sound Classification”, *Proceedings of the 23rd ACM International Conference on Multimedia*, 2015, pp. 1015-1018.
- [237] Github, “ESC-50: Dataset for Environmental Sound Classification.”, <https://github.com/karolpiczak/ESC-50> (erişim 10.10.2023).
- [238] S. Patil ve K. Wani, “Gear fault detection using noise analysis and machine learning algorithm with YAMNet pretrained network”, *Materials Today: Proceedings*, vol. 72, pp. 1322-1327. 2023
- [239] Github, “YAMNet”, <https://github.com/tensorflow/models/tree/master/research/audioset/yamnet> (erişim 12.12.2023).
- [240] TensorFlow, “ Transfer learning with YAMNet for environmental sound classification”,

[https://www.tensorflow.org/tutorials/audio/transfer\\_learning\\_audio?hl=en](https://www.tensorflow.org/tutorials/audio/transfer_learning_audio?hl=en)  
(erişim 10.12.2023).

- [241] B.L. Giordano, M. Esposito, G. Valente ve E. Formisano, “Intermediate acoustic-to-semantic representations link behavioral and neural responses to natural sounds”, *Nature Neuroscience*, vol. 26(4), p. 664-672, 2023.
- [242] Github, Customizing an Audio Classification model, [https://github.com/googlecode/odml-pathways/blob/main/audio\\_classification/doc/transfer\\_learning.md](https://github.com/googlecode/odml-pathways/blob/main/audio_classification/doc/transfer_learning.md) (erişim 23.07.2023).
- [243] Keras, “Adam Optimizer.” <https://keras.io/api/optimizers/adam/> (erişim 10.11.2023).
- [244] K. Zeng, J. Liu, Z. Jiang ve D. Xu, “A Scaling Transition Method from SGDM to SGD with 2ExpLR Strategy”, *Applied Sciences*, vol. 12(23), p. 12023, 2022.
- [245] T. Tieleman ve G. Hinton, “Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude”, *COURSERA: Neural networks for machine learning*, vol. 4(2), pp. 26-31, 2012.
- [246] E. Seyyarer, F. Ayata, T. Uçkan ve A. Karci, “Derin öğrenmede kullanılan optimizasyon algoritmalarının uygulanması ve kıyaslanması”, *Computer Science*, vol. 5(2), pp. 90-98, 2020.
- [247] S. Ruder, “An overview of gradient descent optimization algorithms”, 2016, arXiv:1609.04747.
- [248] GeeksforGeeks, “Adam Optimizer in Tensorflow”, <https://www.geeksforgeeks.org/adam-optimizer-in-tensorflow/> (erişim 13.06.2023).
- [249] R. Zaheer ve H. Shaziya, "A Study of the Optimization Algorithms in Deep Learning," 2019 Third International Conference on Inventive Systems and Control (ICISC), 2019, pp. 536-539.
- [250] D.P. Kingma ve J. Ba, “Adam: A Method for Stochastic Optimization”, 2014, arXiv:1412.6980.
- [251] TensorFlow, “Adam Optimizer”, [https://www.tensorflow.org/api\\_docs/python/tf/keras/optimizers/Adam](https://www.tensorflow.org/api_docs/python/tf/keras/optimizers/Adam) (erişim 11.09.2023).
- [252] B. Wu, A. Liang, H. Zhang, T. Zhu, Z. Zou, D. Yang, W. Tang, J. Li ve J. Su, “Application of conventional UAV-based high-throughput object detection to the early diagnosis of pine wilt disease by deep learning”, *Forest Ecology and Management*, vol. 486, p. 118986, 2021.
- [253] K. Bhujbal ve S. Barahate, “Custom Object detection Based on Regional Convolutional Neural Network & YOLOv3 With DJI Tello Programmable Drone”, 7th International Conference on Innovations and Research in Technology and Engineering (ICIRTE), 2022, pp. 1-9.
- [254] Seedstudio, “Raspberry Pi 4 Model B.”, <https://www.seedstudio.com/blog/2020/05/28/meet-the-brand-new-raspberry-pi-4-8gb-ram/> (erişim 18.05.2023).

- [255] Makotold, “Raspberry Pi Kamera Modülü V2.”, <https://makotold.com/product/raspberry-pi-kamera-camera-modulu-v2/> (erişim 20.05.2023).
- [256] Sixfab, “Sixfab 4G/LTE Cellular Modem Kit for Raspberry Pi.” <https://sixfab.com/product/raspberry-pi-4g-lte-modem-kit/> (erişim 12.06.2023).
- [257] Sixfab, “Sixfab Bağlantısı | SIM ve Veri Yönetimi.”, <https://connect.sixfab.com/#/login> (erişim 12.06.2023).
- [258] A. M. Kist ve Y. Sun, “Deep Learning on Edge TPUs”, 2021, arXiv:2108.13732.
- [259] Coral, USB Accelerator, <https://coral.ai/products/accelerator> (erişim 10.05.2023).
- [260] Teknostore, “Harici USB Ses Kartı.”, <https://www.teknostore.com/ugreen-usb-to-3-5mm-trrs-harici-ses-karti-1828> (erişim 19.06.2023).
- [261] J. Unpingco, “Numpy”, *Python Programming for Data Analysis*, Springer, 2021, pp. 103-126.
- [262] K. Nikhil, K. Drutesh, A. K. R. Manchuri, B. H. Kiran, V. S. Teja, R. Manjula, “Speech Emotion Recognition Using Librosa”, *International Journal of Scientific Development and Research*, vol. 8(5), pp. 1119-1127, 2023.
- [263] A. Danial, *NumPy and SciPy, in Python for MATLAB Development: Extend MATLAB with 300,000+ Modules from the Python Package Index*, Apress, 2022.
- [264] A. Rayhan ve R. Kinzler, “Advancing Scientific Computing with Python’s SciPy Library”, 2023.
- [265] A. Bizzego, A. Battisti, G. Gabrieli, G. Esposito ve C. Furlanello, “pyphysio: A physiological signal processing library for data science approaches in physiology”, *SoftwareX*, vol. 10: p. 100287, 2019.
- [266] J. Hunt, “Introduction to Matplotlib”, *Advanced Guide to Python 3 Programming*, Springer International Publishing, 2019, pp. 35-42.
- [267] A. Pajankar, *Hands-on Matplotlib: Learn Plotting and Visualizations with Python 3*, Apress, 2022.
- [268] P. Boris, *Pandas in Action*, Manning, 2021.
- [269] H. Pham, “PyAudio.”, <https://pypi.org/project/PyAudio/> (erişim 12.12.2023).
- [270] S. Samuel ve D. Mietchen, “Computational reproducibility of Jupyter notebooks from biomedical publications”, 2023, arXiv:2308.07333.
- [271] S. -C. Huang ve T. -H. Le, “Advanced TensorBoard”, *Principles and Labs for Deep Learning*, Academic Press, 2021, pp. 169-200.
- [272] F. A. KAZAN ve H. SOLAK, “Improvement of ultrasonic sensor-based obstacle avoidance system in drones”, *International Journal of Aeronautics and Astronautics*, vol. 4, no. 1, pp. 9–35, 2023.
- [273] J. Li, Y. Zhou ve L. Lamont, "Communication architectures and protocols for networking unmanned aerial vehicles," 2013 IEEE Globecom Workshops (GC Wkshps), 2013, pp. 1415-1420.

- [274] A. Allouch, O. Cheikhrouhou, A. Koubâa, M. Khalgui and T. Abbes, "MAVSec: Securing the MAVLink Protocol for Ardupilot/PX4 Unmanned Aerial Systems," 2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC), 2019, pp. 621-628.
- [275] H. Tian, T. Wang, Y. Liu, X. Qiao ve Y. Li, "Computer vision technology in agricultural automation—A review", *Information Processing in Agriculture*, vol. 7(1), pp. 1-19, 2020.
- [276] M. Ouhami, A. Hafiane, Y. Es-Saady, M. El Hajji ve R. Canals, "Computer Vision, IoT and Data Fusion for Crop Disease Detection Using Machine Learning: A Survey and Ongoing Research", *Remote Sensing*, vol. 13(13), p. 2486, 2021.
- [277] M. Mittal, R. Mohan, W. Burgard ve Abhinav Valada, "Vision-based autonomous UAV navigation and landing for urban search and rescue, The International Symposium of Robotics Research, 2019, pp. 575–592.
- [278] B. Mishra, D. Garg, P. Narang ve V. Mishra, "Drone-surveillance for search and rescue in natural disaster", *Computer Communications*, vol. 156, p. 1-10, 2020.
- [279] E. Lygouras, N. Santavas, A. Taitzoglou, K. Tarchanidis, A. Mitropoulos ve A. Gasteratos, "Unsupervised human detection with an embedded vision system on a fully autonomous UAV for search and rescue operations", *Sensors*, vol. 19(16), p. 3542, 2019.
- [280] S. Kentsch, M. Cabezas, L. Tomhave, J. Groß, B. Burkhard, M. L. Lopez Caceres, K. Waki ve Y. Diez, "Analysis of UAV-acquired wetland orthomosaics using GIS, computer vision, computational topology and deep learning", *Sensors*, vol. 21(2), p. 471, 2021.
- [281] S. Micklethwaite, D. Turner, Y. Vasuki, P. Kovesei, Holden, E. -J. Holden ve A. Lucieer, "Mapping from an Armchair: Rapid, high-resolution mapping using UAV and computer vision technology", *Proceedings of Structural Geology and Resources*, vol. 2012, p. 130-133, 2012.
- [282] K. Luo, X. Kong, J. Zhang, J. Hu, J. Li ve H. Tang, "Computer vision-based bridge inspection and monitoring: A review", *Sensors*, vol. 23(18), p. 7863, 2023.
- [283] F. Mirallès, N. Pouliot ve S. Montambault, "State-of-the-art review of computer vision for the management of power transmission lines," *Proceedings of the 2014 3rd International Conference on Applied Robotics for the Power Industry*, 2014, pp. 1-6.
- [284] Jenssen, R. ve D. Roverso, Automatic autonomous vision-based power line inspection: A review of current status and the potential role of deep learning", *International Journal of Electrical Power & Energy Systems*, vol. 99, pp. 107-120, 2018.
- [285] H. Liu, Y. Yu, S. Liu ve W. Wang, "A Military Object Detection Model of UAV Reconnaissance Image and Feature Visualization", *Applied Sciences*, vol. 12(23), p. 12236, 2022.

- [286] Y.H. Lin ve O.N.N. Fernando, "Animal Hunt: AI-Based Animal Sound Recognition Application", "International Conference on Human-Computer Interaction", 2023, pp. 465–474.
- [287] M. Lamrini, M.Y. Chkouri ve A. Touhafi, "Evaluating the Performance of Pre-Trained Convolutional Neural Network for Audio Classification on Embedded Systems for Anomaly Detection in Smart Cities", *Sensors*, vol. 23(13), p. 6227, 2023.
- [288] E. Grooby, C. Sitaula, D. Fattahi, R. Sameni, K. Tan, L. Zhou, ..., F. Marzbanrad, "Real-Time Multi-Level Neonatal Heart and Lung Sound Quality Assessment for Telehealth Applications," in *IEEE Access*, vol. 10, pp. 10934-10948, 2022.
- [289] L.G. Di Maggio, "Intelligent Fault Diagnosis of Industrial Bearings Using Transfer Learning and CNNs Pre-Trained for Audio Classification", *Sensors*, vol. 23(1), p. 211, 2023.
- [290] A. Maity, A. Pathak ve G. Saha, "Transfer learning based heart valve disease classification from Phonocardiogram signal", *Biomedical Signal Processing and Control*, vol. 85, p. 104805, 2023.
- [291] M. Boulares, T. Alafif ve A. Barnawi, "Transfer Learning Benchmark for Cardiovascular Disease Recognition," *IEEE Access*, vol. 8, pp. 109475-109491.
- [292] Y. Anwar, S. M. Mullan, O. Chipara, A. M. Segre and P. Polgreen, "Audio-Based Cough Detection in Clinic Waiting Rooms," 2022 IEEE 10th International Conference on Healthcare Informatics (ICHI), 2022, pp. 182-191.
- [293] U. Ranasinghe, S. Abeyrathne, L. Samaranayake, T. Weerakoon, N. Harischandra ve G. Dissanayake, "Enhanced Frequency Domain Analysis for Detecting Wild Elephants in Asia using Acoustics", IEEE 17th International Conference on Industrial and Information Systems (ICIIS), 2023, pp. 140-145.

## ÖZGEÇMİŞ

Ertuğrul KIRAÇ

### ÖĞRENİM DURUMU:

- **Lisans** : 2010, Bolu Abant İzzet Baysal Üniversitesi, Fizik
- **Y. Lisans** : 2010-2014 İstanbul Yıldız Teknik Üniversitesi, Fizik
- **Doktora** : 2014-2024 İstanbul Gedik Üniversitesi Savunma Teknolojileri Ana Bilim Dalı-Savunma Teknolojileri Programı

### MESLEKİ DENEYİM:

- Yazılım Geliştirme Uzmanı, Spintek İletişim A.Ş (2012-2013)
- Yazılım Uzmanı – Arge, Mobilera Bilişim ve İletişim A.Ş (2013- 2014)
- Yazılım Uzmanı – Arge, Soft Bilgi İşlem ve Danışmanlık A.Ş (2014 – 2015)
- Yazılım Uzmanı – Arge, Kozmo Shop Kozmetik (2016 – 2023)