



**BİLGİSAYARLI GÖRÜ İLE OTONOM HEDEF
TAKİBİ YAPABİLEN İHA'NIN
GELİŞTİRİLMESİ**

Muhammed Yavuz BAYRAK

Danışman: Dr. Öğr. Üyesi Levent BAYINDIR

Yüksek Lisans Tezi

Bilgisayar Mühendisliği Ana Bilim Dalı

2022

(Her hakkı saklıdır.)

T.C.
ATATÜRK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANA BİLİM DALI

**BİLGİSAYARLI GÖRÜ İLE OTONOM HEDEF TAKİBİ YAPABİLEN İHA'NIN
GELİŞTİRİLMESİ**

(Development of a UAV Capable of Autonomous Target Tracking with Computer Vision)

YÜKSEK LİSANS TEZİ

Muhammed Yavuz BAYRAK

Danışman: Dr. Öğr. Üyesi Levent BAYINDIR

Erzurum
Aralık, 2022

KABUL VE ONAY TUTANAĐI

Muhammed Yavuz BAYRAK tarafından hazırlanan “BİLGİSAYARLI GÖRÜ İLE OTONOM HEDEF TAKİBİ YAPABİLEN İHA’NIN GELİŞTİRİLMESİ” başlıklı çalışması 26 / 12 / 2022 tarihinde yapılan tez savunma sınavı sonucunda başarılı bulunarak jürimiz tarafından Bilgisayar Mühendisliği Ana Bilim Dalı, Bilgisayar Mühendisliği Bilim Dalında, yüksek lisans tezi olarak kabul edilmiştir.

Jüri Başkanı:	Dr. Öğr. Üyesi Faruk Baturalp Günay <i>Atatürk Üniversitesi</i>	Aslı Islak İmzalıdır
Danışman:	Dr. Öğr. Üyesi Levent Bayındır <i>Atatürk Üniversitesi</i>	Aslı Islak İmzalıdır
Jüri Üyesi:	Işıl Karabey Aksakallı <i>Erzurum Teknik Üniversitesi</i>	Aslı Islak İmzalıdır

Enstitü Yönetim
Kurulunun/....../.... tarih
ve sayılı kararı.

Bu tezin Atatürk Üniversitesi Lisansüstü Eğitim ve Öğretim Yönetmeliği’nin ilgili maddelerinde belirtilen şartları yerine getirdiğini onaylarım.

Prof. Dr. Saltuk Buğrahan CEYHUN

Enstitü Müdürü

Aslı Islak İmzalıdır

Not: Bu tezde kullanılan özgün ve başka kaynaklardan yapılan bildiriş, çizelge, şekil ve fotoğrafların kaynak olarak kullanımı, 5846 sayılı Fikir ve Sanat Eserleri Kanunundaki hükümlere tabidir.

ETİK BİLDİRİM VE İNTİHAL BEYAN FORMU

Yüksek Lisans Tezi olarak Dr. Öğr. Üyesi Levent BAYINDIR danışmanlığında sunulan “BİLGİSAYARLI GÖRÜ İLE OTONOM HEDEF TAKİBİ YAPABİLEN İHA’NIN GELİŞTİRİLMESİ” başlıklı çalışmanın tarafımızdan bilimsel etik ilkelere uyularak yazıldığını, yararlanılan eserlerin kaynakçada gösterildiğini, Fen Bilimleri Enstitüsü tarafından belirlenmiş olan Turnitin Programı benzerlik oranlarının aşılmadığını ve aşağıdaki oranlarda olduğunu beyan ederiz.

Tez Bölümleri	Tezin Benzerlik Oranı (%)	Maksimum Oran (%)
Giriş	0	30
Kuramsal Temeller	0	30
Materyal ve Yöntem	1	35
Bulgular	0	20
Tartışma	0	20
Tezin Geneli	0	25

Not: Yedi kelimeye kadar benzerlikler ile Başlık, Kaynakça, İçindekiler, Teşekkür, Dizin ve Ekler kısımları tarama dışı bırakılabilir. Yukarıdaki azami benzerlik oranları yanında tek bir kaynaktan olan benzerlik oranlarının %5'den büyük olmaması gerekir.

Beyan edilen bilgilerin doğru olduğunu, aksi halde doğacak hukuki sorumlulukları kabul ve beyan ederiz.

Tez Yazarı (Öğrenci)	Tez Danışmanı
Muhammed Yavuz Bayrak	Dr. Öğr. Üyesi Levent BAYINDIR
26.12.2022	26.12.2022
İmza: Aslı Islak İmzalıdır	İmza: Aslı Islak İmzalıdır

* Tez ile ilgili YÖKTEZ’de yayınlamasına ilişkin bir engelleme var ise aşağıdaki alanı doldurunuz.

☐ Tezle ilgili patent başvurusu yapılması / patent alma sürecinin devam etmesi sebebiyle Enstitü Yönetim Kurulunun/....../.... tarih ve sayılı kararı ile teze erişim 2 (iki) yıl süreyle engellenmiştir.

☐ Enstitü Yönetim Kurulunun/....../.... tarih ve sayılı kararı ile teze erişim 6 (altı) ay süreyle engellenmiştir.

TEŞEKKÜR

Yüksek lisans öğrenimim boyunca bilgi ve tecrübeleri ile bana yol gösteren tez danışmanım ve hocam Sayın Dr. Öğr. Üyesi Levent Bayındır'a, hayatım boyunca desteklerini benden asla esirgemeyen ve lisansüstü eğitimimi tamamlamam konusunda en büyük motivasyon kaynağım olan aileme, lisansüstü öğrenimim ile ilgili gösterdikleri yardım ve anlayışlarından dolayı tüm çalışma arkadaşlarıma, destekleri ve sağladığı çalışma ortamından dolayı Adem Çamlıkaya'ya, çalışmaya ilham kaynağı olan Teknofest organizatörlerine, Teknofest kapsamındaki Savaşan İHA yarışmasına birlikte katıldığımız ve tez kapsamında İHA'nın üretimine büyük katkı sağlamış olan dATA öğrenci kulübü üyelerine, yazım kurallarının denetimi ve düzenlenmesinde yardımcı olan Coşkun Irmak, Ömer Faruk Yıldırım ve Selami Erdem'e saygı ve sevgilerimle sonsuz teşekkür ve minnetlerimi sunarım.

Muhammed Yavuz BAYRAK

ÖZET

YÜKSEK LİSANS TEZİ

BİLGİSAYARLI GÖRÜ İLE OTONOM HEDEF TAKİBİ YAPABİLEN İHA'NIN GELİŞTİRİLMESİ

Muhammed Yavuz BAYRAK

Danışman: Dr. Öğr. Üyesi Levent BAYINDIR

Amaç: Çalışmada nesne tespiti ve takibi yapabilen otonom bir İHA'nın geliştirilmesi amaçlanmıştır. Bahsi geçen kabiliyetler, düşük maliyetli donanımlarda tümleşik bir yazılımla sağlanmaya çalışılmıştır. Nesne tespiti ve takibinin düşük maliyetli donanımlarla ve sadece derin öğrenme metotlarıyla yapılması durumunda karşılaştığı performans problemlerinin gerçek zamanlı takip algoritmalarıyla iyileştirilmesi hedeflenmiştir. Bu kapsamda YOLO ve takip algoritmalarının birlikte gösterdiği performans ve doğruluk değerleri incelenmiştir.

Yöntem: Amaçlanan İHA'nın üretimi için mekanik, elektronik ve yazılımsal tasarımlar yapılmıştır. İHA gövdesi için 3B yazıcıyla üretilen gövde iyileştirmeleri tasarlanıp üretilmiştir. Elektronik gereksinimler için iki bağımsız güç dağıtım sistemi kullanılmıştır. Elektronik ve mekanik tasarımın uçuş süresine etkileri açıklanmıştır. İHA'nın nesne tespiti ve takibi yapabilmesi için YOLO ile takip algoritmalarının birlikte çalıştırılması sağlanmıştır. İHA'da tespit, takip, otonomi ve haberleşme gibi yazılımsal isteklerin karşılandığı yazılım çerçevesi modülleriyle açıklanmıştır.

Bulgular: Geliştirdiğimiz 3261 gram ağırlığındaki İHA, 23,19 dakikaya kadar uçuş yapabilmektedir. İHA'nın nesne tespiti için kullandığı YOLO modeli; %86 f-score, %84,17 mAP ve %64,65 IoU değerlerinde doğruluk gösterirken 2 FPS performans göstermiştir. İncelenen takip algoritmalarında en yüksek performansı KCF, en yüksek doğruluğu CSRT göstermiştir. YOLO ve takip algoritmalarının birlikte çalıştırılması durumunda geliştirilen metotla en yüksek doğruluk %56,47 IoU ve 20,99 FPS değerleriyle MEDIANFLOW algoritmasından alınmıştır.

Sonuç: Çalışmada nesne tespiti ve takibi yapabilen otonom bir İHA düşük maliyetli donanımlar ile tasarlanmış ve geliştirilmiştir. Uçuş süresinin artırılması için bağımsız güç dağıtım sistemleri kullanılmış, gövde üzerinde pek çok iyileştirme yapılmıştır. Düşük maliyetli donanım üzerinde derin öğrenmeyle nesne tespiti yapılması durumunda ortaya çıkan performans problemlerine karşın gerçek zamanlı takip algoritmaları kullanılmış, çeşitli algoritmaların performans ve doğruluğa etkileri analiz edilmiştir.

Anahtar Kelimeler: İnsansız Hava Aracı, Nesne Tespiti, Nesne Takibi, Derin Öğrenme, Bilgisayarlı Görü

Aralık 2022, 85 sayfa

ABSTRACT

MASTER'S THESIS

DEVELOPMENT OF A UAV CAPABLE OF AUTONOMOUS TARGET TRACKING WITH COMPUTER VISION

Muhammed Yavuz BAYRAK

Supervisor: Asst. Prof. Dr. Levent BAYINDIR

Purpose: In this study, it is aimed to develop a UAV capable of detecting and tracking objects autonomously. The mentioned capabilities have been tried to be provided with integrated software on low-cost hardware. It is aimed to improve the performance problems encountered with real-time tracking algorithms when object detection and tracking are done with low-cost hardware and only deep learning methods. In this context, the performance and accuracy values of YOLO and tracking algorithms together were examined.

Method: Mechanical, electronic and software designs were made to produce the intended UAV. For the fuselage, 3D-printed upgrades designed and manufactured. Two independent power distribution systems are used for electronic requirements. The effects of electronic and mechanical design on flight time are explained. For the UAV to detect and track objects, YOLO and tracking algorithms are operated together. It is explained with software framework modules that software requirements such as detection, tracking, autonomy, and communication in the UAV are met.

Findings: The developed UAV, weighing 3261 grams, can fly up to 23.19 minutes. The YOLO model used by the UAV for object detection; showed an accuracy of 86% f-score, 84.17% mAP, and 64.65% IoU, while performing 2 FPS. In the examined tracking algorithms, KCF showed the highest performance and CSRT showed the highest accuracy. In case YOLO and tracking algorithms are run together, the highest accuracy with the developed method was obtained from the MEDIANFLOW algorithm with 56.47% IoU and 20.99 FPS.

Results: In this study, an autonomous UAV capable of object detection and tracking is designed and developed with low-cost hardware. Independent power distribution systems were used to increase the flight time, and many upgrades were made to the fuselage. Because of the performance problems that arise in the case of object detection with deep learning on low-cost hardware, real-time tracking algorithms are used and the effects of various algorithms on performance and accuracy are analyzed.

Keywords: Unmanned Aerial Vehicle, Object Detection, Object Tracking, Deep Learning, Computer Vision

December 2022, 85 pages

İÇİNDEKİLER

KABUL VE ONAY TUTANAĞI.....	i
ETİK BİLDİRİM VE İNTİHAL BEYAN FORMU	ii
TEŞEKKÜR	iii
ÖZET	iv
ABSTRACT	v
İÇİNDEKİLER.....	vi
TABLolar DİZİNİ.....	viii
ŞEKİLLER DİZİNİ	ix
KISALTMALAR VE SİMGELER DİZİNİ	xii
GİRİŞ.....	1
KURAMSAL TEMELLER.....	4
İnsansız Hava Araçları ve İHA Sınıfları	4
Nesne Tespiti ve YOLO	5
Nesne Takibi.....	6
İHA’larla Yapılan Bazı Nesne Tespiti ve Takibi Çalışmaları.....	9
MATERYAL VE METOT.....	14
İHA’nın Mekanik, Yapısal ve Aviyonik Tasarımı	14
İHA’da Kullanılan Malzemeler ve Tahmini Gövde Ağırlığı	15
İHA İtki Sistemi ve Azami Uçuş Süresi Hesaplamaları	31
İHA’da Haberleşme ve Yer Kontrol İstasyonu.....	35
İHA ve Yer İstasyonu İçin Geliştirilen Yazılımların Mimarileri	38
İHA Yazılım Mimarisi.....	39
Yer Kontrol İstasyonu Yazılım Mimarisi	45
Nesne Tespiti ve Nesne Takibi	49
Nesne Tespiti İçin Derin Öğrenme	49
Veri Kümeleri	50
Nesne Takibi İçin Takip Algoritmaları.....	51
Nesne Tespiti ve Takibinin Birlikte Çalıştırılması	52
Doğruluk ve Performans Metrikleri.....	53
ARAŞTIRMA BULGULARI ve TARTIŞMA	57
İHA’nın Yapısal, Mekanik ve Aviyonik Çıktıları	57
Nesne Tespiti	62

Nesne Takibi.....	68
Nesne Tespiti ve Takibinin Birlikte Çalıştırılması.....	74
SONUÇLAR VE ÖNERİLER	80
KAYNAKLAR.....	82
ÖZGEÇMİŞ.....	85



TABLÖLAR DİZİNİ

Tablo 1. SHGM İHA Sınıfları (SHGM 2020).	4
Tablo 2. Kavramsal Tasarımda Ortaya Çıkan Tahmini Ağırlık Tablosu.....	30
Tablo 3. YOLO v4 Tiny modelinin eğitilmek istenen veri kümesine ilişkin config dosyasında düzenlenen değerler.	62
Tablo 4. Test ortamına ilişkin parametreler.	63
Tablo 5. Veri kümesinde yapılan test neticesinde elde edilen; TP, FP ve FN değerlerinin 6000 iterasyon boyunca aldığı değerler gösterilmiştir.	65
Tablo 6. Veri kümesinde yapılan test neticesinde elde edilen precision, recall ve f-score değerleri.	65
Tablo 7. Veri kümesinde yapılan test neticesinde elde edilen AP, mAP ve IoU değerleri. Tabloda her bin iterasyonda alınan değerler gösterilmiştir.	66
Tablo 8. 713 frame boyunca YOLO v4 Tiny ile nesne tespiti yapılması durumunda Raspberry Pi 4 Model B'nin sıcaklık değerleri.....	67
Tablo 9. FPS Skorları.....	68
Tablo 10. 1841 görüntü üzerinde yapılan takip testi neticesinde elde edilen anlık ve ortalama IoU değerleri.....	70
Tablo 11. 1841 görüntü üzerinde yapılan takip testi neticesinde takibin sürekliliğine ilişkin elde edilen veriler.	71
Tablo 12. 1841 görüntü üzerinde yapılan takip testi neticesinde elde edilen anlık ve ortalama FPS değerleri.	72
Tablo 13. 1841 görüntü üzerinde yapılan takip testi neticesinde elde edilen işlemci sıcaklığı verileri.....	73
Tablo 14. 5 ve 30 YFT aralığında farklı takip algoritmaları ile elde edilen ortalama FPS değerleri.	75
Tablo 15. 35 ve 250 YFT aralığında farklı takip algoritmaları ile elde edilen ortalama FPS değerleri.	75
Tablo 16. 5 ve 30 YFT aralığında farklı takip algoritmaları ile elde edilen IoU skorları.....	76
Tablo 17. 35 ve 250 YFT aralığında farklı takip algoritmaları ile elde edilen IoU skorları....	76

ŞEKİLLER DİZİNİ

Şekil 1. Kavramsal İHA tasarımındaki aviyonik birimler ve birbirleri ile bağlantılarının genel diyagramı.....	15
Şekil 2. F450 gövdenin 3B tasarımı (sol üstte), F450 gövdenin fotoğrafı (sağ üstte), gövde için tasarlanan iyileştirmeler (sol altta) ve çalışma neticesinde üretilen İHA'nın fotoğrafı (sağ altta) gösterilmiştir.....	16
Şekil 3. Quad X quadcopter gövdesinin (solda) ve octo quad X8 gövdesinin (sağda) temsili görselleri ve motor dönüş yönleri. (Anonymous, 2016).....	17
Şekil 4. Motor sayısını arttırmak için tasarlanmış parçalar.....	17
Şekil 5. İHA gövdesindeki montaj alanının arttırılması için kullanılan levhalar. Solda gövde alt katında kullanılan levha, ortada gövde üst katında işlem birimi için kullanılan levha ve sağda kamera modülünün üstünde kullanılan ızgara gösterilmiştir.	18
Şekil 6. Kamera ve sigortaların montajlandığı levhaların 3B tasarımları ve çalışma neticesindeki fotoğrafları.....	18
Şekil 7. İniş takımında kullanılan parçaların tasarımı (solda) ve iniş takımının montajlanmış hali (sağda).	19
Şekil 8. F450 gövdesi üzerinde kullanılmak üzere üretilen parçaların bir arada gösterimi.	19
Şekil 9. Pixhawk 2.4.8 uçuş kontrol kartı (solda) ve Raspberry Pi 4 (sağda) fotoğrafları.....	20
Şekil 10. İHA gövdesindeki GPS modülünün ve modül altındaki RF kalkanının fotoğrafları.	21
Şekil 11. Raspberry Pi kamera modülü ve esnek kablosunun fotoğrafı.....	21
Şekil 12. Gövdede kullanılan buzzer (solda), safety switch (ortada) ve güç modülünün (sağda) fotoğrafları.	22
Şekil 13. Telemetri alıcı ve vericisinin antenleriyle birlikte fotoğrafı.	22
Şekil 14. LoRa modülü (solda) ve PL2303 USB-TTL dönüştürücü (sağda) fotoğrafları.....	23
Şekil 15. FPV alıcı ve vericisinin bağlantı kablolarıyla birlikte fotoğrafı.	23
Şekil 16. EasyCAP (solda) ve HDMI2AV Dönüştürücü (sağda) fotoğrafları.	24
Şekil 17. AT9S RC vericinin (solda) ve R9DS RC alıcının (sağda) fotoğrafları.....	24
Şekil 18. SunnySky X2212-13 980 KV BLDC motorun fotoğrafı.....	25
Şekil 19. 1045 pervane çifti fotoğrafı.....	26
Şekil 20. İHA'da kullanılmış 30 A ESC'nin fotoğrafı.....	26
Şekil 21. JetFire 11.1 V 8000 mAh 70 C 3S1P LiPo bataryanın fotoğrafı.	27

Şekil 22. F450 bottom board (solda) ve 80 A sigorta (sağda) fotoğrafları.	28
Şekil 23. Matek PDB (Power distribution board) ve çalışmada kullanılmak üzere hazırlanmış halinin fotoğrafları.	28
Şekil 24. Güç dağıtım kartının altına yerleştirildiği levha ve montajlanışının fotoğrafları.	29
Şekil 25. 3DR güç modülünün fotoğrafı.	29
Şekil 26. Askı halindeki İHA üzerine etki eden itki ve ağırlık kuvvetlerinin gösterimi. T (Thrust) itki kuvvetini; W, ağırlığı göstermektedir. Askı durumunda itki, ağırlığa eşittir.	31
Şekil 27. SunnySky X2212-13 980 KV BLDC motorun thrust tablosu. Soldan sağa doğru ilk sütunda kullanılan pervane, ikinci sütunda batarya gerilimi (V), üçüncü sütunda anlık akım (A), dördüncü sütunda üretilen itki kuvveti (gr), beşinci sütunda güç tüketimi (Wh), altıncı sütunda motor verimliliği (gr/Wh) ve yedinci sütunda motor sıcaklığı (C°) gösterilmiştir.	32
Şekil 28. Telemetri (solda), LoRa (ortada) ve RC832 video alıcısı (sağda) fotoğrafları.	36
Şekil 29. Kavramsal tasarımda yer kontrol istasyonu ile kullanılacak donanımların ve birbirleri ile bağlantılarının genel diyagramı.	37
Şekil 30. Mission Planner'ın video üzerine eklediği OSD bileşenleri (solda) ve Mission Planner'ın ekran görüntüsü (sağda).	38
Şekil 31. Yazılım tasarımındaki modülleri gösteren genel diyagram.	38
Şekil 32. Throttle, yaw, pitch ve roll manevralarının temsili görselleri (Anonymous, 2020).	40
Şekil 33. İHA takip manevrası çalışmasından. yer kontrol istasyonunda İHA'nın sağa doğru yaptığı manevra (solda) ve İHA'nın manevra esnasında aldığı görüntü (sağda) gösterilmiştir.	42
Şekil 34. Geliştirilen yer istasyonu arayüzünün (görsel üzerinde bazı açıklamalarla) gösterimi.	47
Şekil 35. Tespit ve takip işlemlerinin birlikte çalıştırılması ve YFT değeri.	53
Şekil 36. IoU metriğinin hesaplanmasında kullanılan alanların temsili gösterimi.	55
Şekil 37. İHA ağırlık ölçümünden alınan sonucun fotoğrafı.	57
Şekil 38. 3B çıktıların ağırlıklarının ölçülmesi (solda) ve üretilmiş parçalar (sağda).	58
Şekil 39. İHA gövdesindeki tüm aviyonik bileşenleri ve aralarındaki bağlantıları gösteren diyagram. Çizimde batarya gerilimi 12 V kabul edilmiştir. 12 V ile çalışamayan bileşenlerin gerilim ve akım değerleri kendi çalışma gerilimleri dikkate alınarak yazılmıştır. Motor ve ESC'ler arasında kullanılan konnektörler gösterilmemiştir.	61
Şekil 40. Training neticesinde üretilen Loss ve mAP grafiği.	62
Şekil 41. Test sonucunda elde edilen sonuçlar.	64
Şekil 42. Test esnasında alınmış bir görüntü.	66

Şekil 43. BOOSTING, CSRT, KCF, MEDIANFLOW, MIL ve MOSSE isimli takip algoritmalarının ortalama IoU değerlerinin zaman içerisinde değişimi gösterilmiştir.....	69
Şekil 44. BOOSTING algoritmasının YOLO ile farklı YFT değerlerinde ürettiği FPS (solda) ve IoU (sağda) değerleri.	77
Şekil 45. CSRT algoritmasının YOLO ile farklı YFT değerlerinde ürettiği FPS (solda) ve IoU (sağda) değerleri.	77
Şekil 46. KCF algoritmasının YOLO ile farklı YFT değerlerinde ürettiği FPS (solda) ve IoU (sağda) değerleri.	78
Şekil 47. MEDIANFLOW algoritmasının YOLO ile farklı YFT değerlerinde ürettiği FPS (solda) ve IoU (sağda) değerleri.	78
Şekil 48. MIL algoritmasının YOLO ile farklı YFT değerlerinde ürettiği FPS (solda) ve IoU (sağda) değerleri.	79
Şekil 49. MOSSE algoritmasının YOLO ile farklı YFT değerlerinde ürettiği FPS (solda) ve IoU (sağda) değerleri.	79

KISALTMALAR VE SİMGELER DİZİNİ

3B	: Üç Boyutlu
AV	: Audio Visual
AP	: Average Precision
API	: Application Programming Interface
BEC	: Battery Eliminator Circuit
BLDC	: Brushless DC (Fırçasız Doğru Akım)
CNN	: Convolutional Neural Networks (Evrişimli Sinir Ağları)
DAL	: Data Access Layer
DC	: Direct Current (Doğru Akım)
ESC	: Electronic Speed Controller
FC	: Flight Controller (Uçuş Kontrolcüsü)
FN	: False Negative
FOV	: Field of View
FP	: False Positive
FPS	: Frames per Second
FPV	: First Person View
GPS	: Global Positioning System
HDMI	: High Definition Multimedia Interface
HTTP	: Hyper Text Transfer Protocol
IMU	: Inertial Measurement Unit
IoU	: Intersection over Union
İHA	: İnsansız Hava Aracı
LiPo	: Lityum Polimer
LoRa	: Long Range
MAV	: Micro Aerial Vehicle
mAP	: Mean Average Precision
MPA	: Multi-Purpose Aircraft
MVC	: Model View Controller
NMS	: Non Maximal Suppression
OSD	: On Screen Display
PCB	: Printed Circuit Board (Baskılı Devre Kartı)
PDB	: Power Distribution Board (Güç Dağıtım Kartı)
PLA	: Polilaktik Asit
PPM	: Pulse Position Modulation
PWM	: Pulse Width Modulation
RC	: Radio Control / Radio Controlled
RF	: Radyo Frekans
RPA	: Remotely Piloted Aircraft
RPAS	: Remotely Piloted Aircraft System
RPM	: Revolutions per Minute

RPV	: Remotely Piloted Vehicle
RTL	: Return To Launch
SBUS	: Serial Bus
SPA	: Special Purpose Aircraft
SPF	: Seconds per Frame
TN	: True Negative
TP	: True Positive
TWR	: Thrust to Weight Ratio
UAS	: Unmanned Aircraft System
UAV	: Unmanned Aerial Vehicle
UCAV	: Unmanned Combat Aerial Vehicle
UI	: User Interface (Kullanıcı Arayüzü)
UMA	: Unmanned Aircraft
USB	: Universal Serial Bus
USDOD	: United States Department of Defense
UX	: User Experience
YFT	: YOLO Frame Threshold
YOLO	: You Only Look Once
YKi	: Yer Kontrol İstasyonu

GİRİŞ

İnsansız hava araçları (İHA), birden fazla mühendislik alanının çalışma ve birikimleri neticesinde ortaya çıkan, uzaktan veya otonom olarak kontrol edilebilen robotik sistemlerdir. İHA'ların tanımı itibariyle temel gereksinimleri sağlaması ve dünya sahnesine günümüzde bildiğimiz anlamda çıkması uzun yıllar süren teknolojik ilerlemenin bir sonucudur. Bu süreçte faydalanılan mühendislik dallarına aerodinamik, aviyonik, bilgisayar, elektrik, elektronik, haberleşme, havacılık, makine, mekatronik ve yazılım örnek olarak gösterilebilir.

Günümüzde elektronik pek çok bileşenin geçmişe göre çok daha ucuz ve harcanabilir olması, açık kaynaklı yazılım kültüründeki ilerlemeler ve bilgi ve birikimin daha erişilebilir olması, İHA'lar gibi birden fazla mühendislik disiplinin çıktısı olan karmaşık sistemlerin üretimini oldukça kolaylaştırmıştır. Bu durumun doğal bir sonucu olarak sektörde ve akademik çalışmalarda pek çok probleme İHA'lar ile çözüm üretilmeye çalışılmaktadır. Hem ülkemizde hem de dünyada İHA'lar ile üretilen söz konusu çözümler, daha çok savunma sanayindeki uygulamaları ile tanınmış olsalar da arama kurtarma çalışmalarından haritalamaya; sinematografiden tarıma kadar pek çok farklı sektörde kendilerine uygulama alanı bulmaktadırlar.

Önerilen ve uygulanan çözümler incelendiğinde, genellikle sabit veya hareketli bir kameraya sahip olan İHA'dan alınan görüntülere nesne tespiti ve takibi için derin öğrenme metotlarının ve takip algoritmalarının uygulandığı sıkça görülmüştür. Bu çalışmalarda kullanılan İHA'ların azımsanamayacak bir bölümünü düşük maliyetli sistemler oluşturmaktadır. Nesne tespiti ve takibi için düşük maliyetli ve tümleşik sistemlerde sadece derin öğrenme metotlarının kullanılması performans açısından verimsiz bir sistemin ortaya çıkmasına sebep olmaktadır. Nesne tespitinin akabinde tamamen takip algoritmalarının kullanılması ise doğruluk problemi oluşturmaktadır. Bu sebeple günümüzde düşük maliyetli ve tümleşik sistemlerde derin öğrenme ve takip algoritmalarının doğruluk ve performans özelliklerini bir araya getirmeyi hedefleyen bazı çalışmalar yapılmaktadır. Diğer bir deyişle; güncel çalışmalarda derin öğrenme metotlarının yüksek işlem karmaşıklığıyla sebep olabildiği performans problemleri, takip algoritmaları gibi doğruluk açısından zayıf ancak yüksek performanslı çözümlerle giderilmeye çalışılmaktadır. Bu çalışmada da derin öğrenme ve takip algoritmaları birlikte kullanılarak birbirlerinin performans ve doğruluk değerleri nasıl etkiledikleri irdelenmiştir. Çalışmada sadece derin öğrenme ile nesne tespiti yapılmamış

ancak takip algoritmalarının zaman zaman doğrulanması için derin öğrenme metotlarının tetiklendiği bir çerçeve kullanılmıştır.

Derin öğrenme metotları ve takip algoritmalarının birlikte kullanıldığı İHA uygulamaları incelendiğinde, sıkça açık kaynaklı İHA konfigürasyonlarının kullanıldığı ve bu konfigürasyonlar üzerinde kullanılan işlem birimlerinin ve elektronik bileşenlerin açıklandığı görülmüştür. Bildiğimiz kadarıyla bu uygulamalarda, yapay zeka ve bilgisayarlı görü metotları detaylı bir şekilde açıklanmakta ancak metotları içeren genel yazılımın mimarisi detaylı bir şekilde açıklanmamaktadır. Bu durum yapılmış çalışmaların farklı araştırmacılar tarafından yeniden denenmesi durumunda doğru test ortamını oluşturmalarını zorlaştırmaktadır. Buna karşın örnek bir yazılım mimarisinin sunulmasında fayda görülmüştür. Geliştirilen yazılımın modülleri ve bu modüllerin işlevleri açıklanarak performans metrikleri değerlendirilirken toplam işlem yükünün de göz önüne alınması sağlanmıştır. Ayrıca bu çalışmayı test etmek ve uygulamak isteyen diğer araştırmacıların daha doğru bir test ortamı hazırlamalarına yardımcı olunması amaçlanmıştır.

Mevcut çalışmalardaki çözümlerin büyük bölümünde, nesne tespiti ve takibi için üretilen İHA'ların havadan karaya sistemler oldukları görülmüştür. Yani İHA uçar haldeyken yerdeki hareketli veya sabit hedefleri tespit ve/veya takip etmektedir. Havacılık ve savunma sanayindeki gelişmeler ise gelecekte havadan havaya çözümlerin sayısının artacağını göstermektedir. Ülkemizde gerçekleştirilen ve oldukça popüler bir organizasyon olan Teknofest Uzay ve Havacılık Festivali'nin en bilindik yarışmalarından bazılarının konusu havadan havaya nesne tespiti ve takibini kapsadığından bu probleme ilişkin çözüm ihtiyacı da ülkemizde popülerliğini oldukça artırmıştır. Bu sebepten ikincil bir fayda sağlamak amacıyla nesne tespit ve takibinde yapılacak testler için havadan havaya İHA tespiti ve takibi üzerine çalışılmasına karar verilmiştir. Yine bu kapsamda tasarlanan ve üretilen İHA'nın yarışma şartlarına uygun olması hedeflenmiştir. Yani İHA'nın uçuş süresi, ağırlığı, kamerasının sabit olması gibi isterler, yarışma şartnamesinde tanımlanan şartlara uygun olarak tasarlanmış ve üretilmiştir. Ayrıca bu çalışmada İHA'nın hem tasarım faaliyetleri neticesinde hesaplanan özellikleri hem de üretim neticesinde ortaya çıkan özellikleri karşılaştırmalı bir şekilde incelenmiştir.

Sonuç olarak bu tez çalışmasında düşük maliyetli donanımlar ile bir İHA'nın üretilmesi ve nesne tespiti ve takibinde performans ve doğruluk açısından bahsi geçen faydaları sağlayacak şekilde derin öğrenme metotları ve takip algoritmalarının birlikte çalıştırılması amaçlanmıştır. Ayrıca üretilen İHA'nın otonomi kabiliyetleri üzerine de çalışılarak otonom manevralar ile fiziksel nesne takibinin geliştirilen nesne tespit ve takip

metotlarıyla birlikte alıřmasını saęlayacak yazılımın mimarisi aıklanmıřtır. zetle; nesne tespiti ve takibi yapabilen, dřk maliyetli, tmleřik ve otonomi kabiliyetlerine sahip bir dner kanatlı İHA geliřtirilmiř; geliřtirilme srecine dair mekanik, elektronik/aviyonik ve yazılımsal gereksinimlerin nasıl giderildięi aıklanmıř, alıřma neticesinde retilen bulgular temel bilgiler ile sunulurak tartıřılmıřtır.



KURAMSAL TEMELLER

İnsansız Hava Araçları ve İHA Sınıfları

İnsansız hava araçları; insan operatör taşımayan, uçuş için aerodinamik kuvvetleri kullanan, otonom veya uzaktan kumada ile kontrol edilebilen, harcanabilir veya geri kazanılabilir ve faydalı yük taşıyabilen araçlardır (USDOD 2005).

İnsansız hava araçları günümüzde popüler araçlar olsa da tarihleri birinci dünya savaşı yıllarına kadar uzanmaktadır. Erken insansız hava araçlarının tarihi, kullanılan teknoloji göz önüne alındığında havadan atımı yapılabilen torpidoların dünya sahnesine çıkmasıyla başlamaktadır (Keane and Carr 2013). Ancak günümüzde bilindiği anlamda UAV (İHA) terimi 1990'lı yıllarda kullanılmaya başlanmıştır. Ayrıca bu yıllarda UAV terimi, RPV (Remotely Piloted Vehicle) teriminin yerini almaya başlamıştır (Newcome 2004). Günümüzde daha çok UAV terimi kullanılsa da RPV, RPA, UMA, UAV, UAS ve RPAS terimleri de aynı konseptteki araçları ve sistemleri ifade etmek için kullanılabilmektedir (Cuerno-Rejado *et al.* 2016).

Çeşitli İHA modellerinin tanımlanmasında birden fazla farklı sınıflandırma metodu uygulanmaktadır. Genel olarak İHA'lar, performans özellikleri (ağırlık, uçuş süresi, menzil, irtifa vb.) ve görev unsurları (UCAV, VTOL, Aerial Delivery/Resupply, SPA, MPA vb.) göz önüne alınarak sınıflandırılabilir (Arjomandi *et al.* 2006). Türkiye Cumhuriyeti'nde Sivil Havacılık Genel Müdürlüğü (SHGM) ise İHA'ları azami kalkış ağırlıklarına göre dört farklı grupta sınıflandırılmaktadır (SHGM 2020).

Tablo 1. SHGM İHA Sınıfları (SHGM 2020).

İHA Sınıfı	Azami Kalkış Ağırlığı
İHA0	500 gr (dahil) – 4 kg arası
İHA1	4 kg (dahil) – 25 kg arası
İHA2	25 kg (dahil) – 150kg arası
İHA3	150 kg (dahil) ve daha fazlası

Günümüzde İHA0 ve İHA1 sınıflarındaki İHA'lar akademik çalışmalarda sıklıkla kullanılmaktadır. Bu çalışmalar incelendiğinde İHA'ların nesne tespiti ve nesne takibi gibi kabiliyetlerle donatıldığı sıklıkla görülmektedir. Ayrıca söz konusu İHA'ların pek çok farklı

uygulama alanında kendilerini gösterdikleri de açıkça görülebilmektedir. Aşağıdaki bölümlerde nesne tespiti, nesne takibi ve İHA'ların örnek kullanımlarına ilişkin bazı çalışmalar paylaşılmıştır.

Nesne Tespiti ve YOLO

Nesne tespiti için kullanılan pek çok metot bulunmaktadır. Bu metotlara R-CNN, SSD (Single Shot MultiBox Detector) ve Fast R-CNN örnek olarak gösterilebilir. Söz konusu metotların başarımında zorlandıkları konu ise performansları olmuştur. Nesne tespitinde performans, genellikle bir saniyede işlenilen görüntü sayısı ile ölçülmektedir. Bu kapsamda, özellikle gerçek zamanlı nesne tespiti yapabilmek için sunulan CNN (Convolutional Neural Network) modelleri yeterli performansı sağlamakta güçlük yaşayabilmektedir.

YOLO metotları, özellikle yüksek performans için hazırlanmış olan CNN modellerini içermektedir. YOLO, 2016 yılında GoogLeNet'den esinlenerek ortaya çıkarılmış bir nesne tespiti metodudur. Yıllar içerisinde YOLO, pek çok araştırmacı tarafından ele alınmış ve geliştirilmiştir. YOLO üzerine çalışan bazı araştırmacılar, yeni YOLO versiyonlarını duyurmuştur. Bazı araştırmacılar da çalışmalarını bu versiyonların varyasyonları olarak sunmuştur. Neticede zaman içerisinde YOLO adı altında pek çok versiyon ve varyasyon oluşmuştur. Ortaya çıkan bu versiyon ve varyasyonlar, diğer nesne tespit metotlarına göre temel bir fark üzerinde durmaktadır. Bu fark; diğer nesne tespit metotlarında görüntü belirgin bölgelere ayrılmaktayken, YOLO'nun tüm görüntüyü tek bir parça halinde evrişim katmanlarında işlemesidir (Redmon *et al.* 2016). Aşağıda söz konusu YOLO versiyonlarına ilişkin literatür paylaşılmıştır.

Redmon *et al.* (2016) tarafından yapılan çalışmada GoogLeNet'den esinlenerek ortaya çıkarılmış yeni bir nesne tespiti metodu olan YOLO ileri sürülmüştür. İleri sürülen metot, birleşik mimari kullanılarak diğer nesne tespiti metotlarının aksine görüntüyü bir bütün halinde tek bir sinir ağından geçirerek sınırlayıcı kutuları ve sınıfları tahmin eder. Bu metotla YOLO, görselleri 45 FPS ile gerçek zamanlı olarak işlemeyi başarmıştır. Ayrıca YOLO'nun Fast YOLO adında daha hafif bir versiyonu da çıkarılmıştır. Çıkarılan bu versiyon ile 155 FPS başarılmıştır. Çalışmada karşılaştırılan diğer gerçek zamanlı nesne tespiti algoritmalarına göre YOLO çok daha yüksek bir mAP (%63,4) değeri göstermiştir. Fast YOLO da aynı şekilde diğer nesne tespit algoritmalarına göre yüksek mAP (%52,7) değerleri başarmıştır. Performans ve doğruluk verileri birlikte değerlendirildiğinde, gerçek zamanlı olmayan tespit metotlarına göre doğruluğunun iyi, FPS değerinin ise en iyisi olmuştur. Karşılaştırıldığı ve çok daha yüksek performans sergilediği diğer metotlar DPM ve R-CNN metotlarıdır. Ayrıca

bu testler VOC2007 Picasso ve People-Art veri kümelerinde gerçekleştirilmiştir. En yüksek AP değerleri YOLO ile elde edilmiştir.

Redmon *and* Farhadi (2017) tarafından yapılan çalışmada YOLO9000 adında yeni bir nesne tespiti sistemi sunulmuştur. Bu sistem 9000'den fazla kategorideki nesneleri tespit edebilmiştir. Çalışmada daha önce ileri sürülen YOLO metodu geliştirilmiş ve YOLO v2 adıyla yayınlanmıştır. YOLO v2; yapılan testlerin ilkinde 67 FPS ile çalışırken %76,8 mAP, bir diğerinde 40 FPS ile çalışırken %78,6 mAP değerlerini başarmıştır. YOLO v2, çeşitli R-CNN ve SSD metotları ile karşılaştırılmıştır. Ayrıca önceki YOLO versiyonundan alınan sonuçlar da karşılaştırmaya dahil edilmiştir. YOLO v2 çeşitli çözünürlüklerde farklı FPS ve mAP değerlerini başarmıştır. 208 x 208 çözünürlükte yapılan nesne tespitinde %69,0 mAP sağlarken 91 FPS skoru üretmiştir. 544 x 544 çözünürlükteyse %78,6 mAP ve 40 FPS değer üretmiştir. YOLO'nun ilk versiyonunda üretilen mAP ve FPS değerlerinde ise 448 x 448 çözünürlüğünde %63,4 mAP ve 45 FPS üretilmiştir.

Redmon *and* Farhadi (2018) tarafından yapılan çalışmada YOLO algoritmasının üçüncü versiyonu ileri sürülmüştür. YOLO v3, YOLO'nun büyük değişikliklerden ziyade çok sayıda küçük değişikliğin yapılarak iyileştirildiği bir versiyonudur. YOLO v2'ye göre daha büyük ama doğruluk açısından daha iyi bir versiyon olarak ortaya çıkmıştır. SSD metodu ile (320 x 320 çözünürlükte) benzer doğruluk sonuçları üretmektedir ancak performansı çok daha yüksektir. YOLO v3 doğruluk açısından muadilleri ile benzer değerler üretirken çok daha yüksek performans gösterebilmektedir.

Bochkovskiy *et al.* (2020) tarafından yapılan çalışmada YOLO v4 ileri sürülmüştür. YOLO v4, evrimsel sinir ağlarında doğruluğu arttırdığı söylenen pek çok metodun test edilmesi ve birlikte kullanılması ile ortaya çıkmıştır. Söz konusu metod ve özellikler büyük veri kümeleri üzerinde test edilerek alınan sonuçlar değerlendirilmiştir. Neticede ortaya çıkan YOLO versiyonu, MS COCO veri kümesi üzerinde test edilerek diğer pek çok nesne tespiti metodu ile karşılaştırılmıştır. %43,5 AP, %65,7 AP@50 doğruluk değerlerini MS COCO veri kümesinde üretmiştir. Bu üretilen değerler TESLA V100 ile test edilmiş ve yaklaşık 65 FPS değerinde performans gösterilmiştir. Bu FPS değeri, metodun gerçek zamanlı çalışabildiğini göstermektedir.

Nesne Takibi

Nesne takibi, genellikle video görüntülerinde belirlenmiş bir alanı veya noktayı videonun diğer görüntülerinde takip edilebilmesini nitelemektedir. Bu bölümde nesne takibi için kullanılan bazı metotlara ilişkin literatür paylaşılmıştır. Üzerinde durulan metotlar genel

itibariyle OpenCV gibi kütüphanelerle entegre edilmiş ve kullanımı görece kolay metotlardır. Bu çalışma kapsamında incelenen metotların pek çoğu üzerinde çalışılmış ve incelenmiş olsa da altı adet metot bu çalışma kapsamında test edilmiştir. Test edilen metotlar ve test sonuçları ilgili bölümde açıklanacaktır.

Grabnel *et al.* (2006) tarafından yapılan çalışmada günümüzde BOOSTING adıyla bilinen metot ileri sürülmüştür. Takipçi, bir sınıflandırıcı olarak tanımlanmıştır. Görüntü üzerinde takip edilecek hedefi kapsayan dörtgen belirlendikten sonra dört aşamada takip işlemi gerçekleştirilmiştir. İlk aşamada takipçi, görüntü üzerinde takip edilen dörtgeni ve dörtgen etrafında bir arama bölgesini belirlemektedir. İkinci aşamada, sınıflandırıcı muhtemel pozisyonlarda araması yapılan dörtgeni işlemektedir. Üçüncü aşamada, önceki işlemin sonuçları kullanılarak kararlılık haritası oluşturulmaktadır. Son olarak kararlılık haritasından faydalanılarak sınıflandırıcı güncellenmekte ve sonraki görüntü üzerinde takip edilen bölgeyi kapsayan dörtgenin konumu tahmin edilmektedir. Çalışmada ileri sürülen gerçek zamanlı takip metodunun 1,6 GHz işlemciye ve 512 MB RAM'e sahip sıradan bir bilgisayar kullanılarak test edildiği ve testlerde 20 FPS başarıldığı açıklanmıştır.

Babenko *et al.* (2009) tarafından yapılan çalışmada; günümüzde MIL adıyla bilinen, tespit ile takip modeline uygun, gerçek zamanlı bir takipçi önerilmiştir. Çalışmada BOOSTING framework temel alınmıştır. Görüntü üzerinde takip edilmesi istenen bölge belirlendikten sonra sınıflandırıcı, söz konusu bölgenin farklı pozisyonlarda birden fazla örneğini üretmektedir. Dolayısıyla görüntü üzerinde takip edilen nesnenin engellenmesi gibi durumlarda takibin doğruluğu korunmaya çalışılmıştır. Ortaya konan MILTrack algoritması, videolar üzerinde bazı takipçilerle karşılaştırılmış ve doğruluk açısından daha iyi sonuçlar üretmiştir.

Bolme *et al.* (2010) tarafından yapılan çalışmada, günümüzde MOSSE adıyla bilinen takip metodu ileri sürmüştür. Yapılan çalışmada yeni bir korelasyon filtresi sunulmuştur. Sunulan yeni korelasyon filtresiyle çalışan takipçi, 2,4 GHz Core 2 Duo işlemciyle 669 FPS değerinde çalışarak yüksek performans göstermiştir. Geliştirilen takipçi, ölçek ve deformasyon gibi durumlara karşı dayanıklıdır. Ayrıca pek çok takip algoritmasına göre daha basit işlemleri barındıracak şekilde geliştirildiği ileri sürülmüştür. ASEF ve UMACE filtreleri gibi daha çok çevrimdışı eğitilen ve nesne tespiti ve tanımlamasında kullanılan metotların filtrelerinden faydalanılmıştır.

Kalal *et al.* (2010) tarafından yapılan çalışmada, günümüzde MEDIANFLOW olarak bilinen takip metodu ortaya konmuştur. Bu metotta takip işlemi, belirgin bir noktanın takip edilmesi şeklinde açıklanmıştır. Video üzerindeki görüntü akışlarını ileri ve geri yönlü olarak

kullanarak ForwardBackward isimli bir hata yakalama metodu kullanılmıştır. Basitçe, takip edilen noktanın sıradaki video görüntüsü üzerinde nerede olduğu tahmin edilmiştir. Daha sonra tahmin edilen noktanın önceki görüntüdeki yeri tekrar tahmin edilerek iki tahmin işleminin çıktıları birbirleri ile karşılaştırılmıştır. İki noktanın arasında çok büyük fark olması, takibin hatalı olduğunu göstermektedir. Dolayısı ile çalışmada hataya dayalı bir takipçi önerilmiştir. Çalışmada önerilen takipçinin yüksek performans ve doğrulukla çalıştığı yapılan test ve karşılaştırmalarla ileri sürülmüştür.

Kalal *et al.* (2011) tarafından yapılan çalışmada, TLD adıyla bilinen metot ileri sürülmüştür. Bu metot takip, öğrenme ve tespit olarak özetlenmiştir. Metot, içerisinde P-N learning adında yeni bir öğrenme metodunu kullanmaktadır. Takipçi, her bir görüntüde takip ettiği nesnenin örneklerini kullanarak P-N learning ile kendini eğitmektedir. P-N learning ile nesne tespit performansının artırılması amaçlanmıştır. Bu durum; uzun vadeli takiplerde, takip edilen nesnenin farklı görünümelerini takipçinin öğrenmesini sağlamaktadır. Ayrıca gerçek zaman algısı da korunmaktadır. P-N learning, experts adı verilen iki bölümü içermektedir. Bunlar tespitin doğru olduğu durum için T expert ve tespitin hatalı olduğu durum için N expert olarak açıklanmıştır. MIL de dahil olmak üzere bazı metotlar ile karşılaştırılarak söz konusu metodun ürettiği recall, precision ve f-score gibi doğruluğa ilişkin değerler sunulmuştur.

Henriques *et al.* (2014) tarafından yapılan çalışmada, günümüzde KCF olarak bilinen takip algoritması ileri sürülmüştür. İleri sürülen metot, lineer regresyon ve yeni bir korelasyon filtresi kullanmaktadır. Çalışma kapsamında MOSSE, MIL ve TLD algoritmaları da dahil olmak üzere diğer takipçiler ile karşılaştırılmıştır. Karşılaştırma için 50 videodan oluşan bir veri kümesi kullanılmıştır. Karşılaştırmada ortalama doğruluk ve FPS değerleri üzerinde durulmuştur. Neticede KCF algoritmasının yüksek doğruluk ve rekabet edebilir düzeyde performans sergilediği gösterilmiştir.

Held *et al.* (2016) tarafından yapılan çalışmada, GOTURN adı ile bilinen nesne takip metodu ileri sürülmüştür. GOTURN metodu bildiğimiz kadarıyla ve çalışmada iddia edildiği üzere ilk yapay sinir ağı ile nesne takibi metodudur. Çalışmada yapay sinir ağı önceden video veri kümeleri ile eğitilmiş ancak sınıf verileri belirtilmemiştir. Dolayısı ile GOTURN metodu, başlangıç ağırlıklarını içeren eğitilmiş bir model ile kullanılmaktadır. Söz konusu model ile yeni nesneler takip edildiğinde 100 FPS gibi yüksek bir performans gösterildiği ileri sürülmüştür. 100 FPS üretmek için Nvidia GTX Titan X GPU kullanılmıştır. CPU ile yapılan performans testinde ise 2,7 FPS üretilmiştir. Bunun sebebi, gerçek zamanlı olarak nitelendirilebilecek eğitim sürecidir.

Lukezix *et al.* (2017) tarafından yapılan çalışmada, günümüzde CSRT adıyla bilinen CSR-DCF metodu ileri sürülmüştür. Metodun geliştirilmesinde özellikle kısa vadeli takip işlemi üzerinde durulmuştur. Kısa vadeli takiplerde takipçi, görüntü üzerindeki takip edilecek alanla alakalı yalnızca bir örneğe sahip olarak takibe başlamaktadır. Ayrıca görüntülerde takip edilen nesne deformasyona uğrayabilmektedir. Bu deformasyonlara nesnenin hızlı hareket etmesi, kameranın hareketli olması veya arka plan gürültüsü örnek olarak gösterilebilir. Uzun vadeli bir takip durumunda takipçi farklı örnekler ile eğitilebilirken kısa vadede yeterli örneğin üretilmesi her zaman mümkün olmamaktadır. Bu durum kısa vadeli takibi daha zor hale getirmektedir. Yüksek doğruluk ve performans değerleri gösteren pek çok takipçi daha önce ileri sürülse de bahsi geçen durumlarla baş edebilmek oldukça zorlu bir görevdir. Bu çalışmada ileri sürülen CSR-DCF metodu, oldukça yüksek doğruluk değerleri üretebilmiş ve CPU üzerinde gerçek zamanlı çalıştırılabilmektedir. Elde edilen sonuçlar KCF metodu da dahil olmak üzere pek çok metod ile karşılaştırılmıştır. Karşılaştırmada ortalama doğruluk, FPS ve ortalama hata verileri sunulmuştur.

İHA'larla Yapılan Bazı Nesne Tespiti ve Takibi Çalışmaları

Günümüzde İHA'lar pek çok farklı probleme çözüm olarak hem sektörde hem de akademik çalışmalarda ileri sürülebilmektedir. İHA'ların kullanım alanlarının yaygınlaştığı ve havadan gözlem yeteneği sebebiyle nesne tespiti ve takibi uygulamalarında sıkça kullanıldığı görülmektedir. Bu kapsamda İHA'ların trafik ve atık kontrolden, hayvancılık ve tarıma kadar birbirlerinden oldukça farklı sektörlerde kendilerine yer edindikleri ileri sürülebilir. Bu durum yapılan literatür taramalarında da kendisini göstermektedir. Bu kapsamda İHA'lar kullanılarak nesne tespiti ve/veya takibi konularında yapılmış bazı örnek çalışmalar bu bölümde paragraflar halinde sunulmuştur.

Chen *and* Miao (2020) tarafından yapılan çalışmada, doğal afetler neticesinde elektrik dağıtım sistemlerinde yaşanan hasarların daha etkili ve verimli bir şekilde tespiti amaçlanmıştır. Bunun için bir İHA ile elektrik direkleri ve hasar durumları tespit edilmeye çalışılmıştır. Çalışmada elektrik direklerinin kayıp/hasar durumunu ve sayımını hızlı bir şekilde değerlendirmek için yaklaşık %90 doğruluk ile çalışan bir uygulama önerilmiştir. Enerji sektörüne yönelik olarak üretilen bu çözümde nesne tespiti için YOLO v3'ten faydalanılmıştır.

Alam *and* Valles (2021) tarafından yapılan çalışmada, yola saçılmış enkaz parçalarının tespit edilmesi üzerine çalışılmıştır. Trafik kazaları gibi durumların neticesinde ortaya çıkan enkazın temizlenmesi için gerekli iş yükünü daha iyi optimize etmek için İHA'ların

kullanılabileceği ve bu sayede daha kısa sürede enkazın temizlenebileceği ileri sürülmüştür. Bu sebeple bir İHA ile yola saçılmış enkazların tespiti için SSD ve Faster R-CNN metotları ile nesne tespiti yapılmıştır.

Tzelepi *and* Tefas (2017) tarafından yapılan çalışmada, İHA'lar ile insan topluluklarının ve kalabalık alanların tespiti üzerine çalışılmıştır. Kalabalık insan topluluklarının tespiti için birden fazla evrişimli sinir ağı modeli üretilmiş ve eğitim aşamasına yönelik farklı yaklaşımlar karşılaştırılmıştır. Yapılan karşılaştırma neticesinde elde edilen en iyi doğruluk oranı %95,32 olmuştur. Bu doğruluk oranını elde etmek için kalabalık ve kalabalık olmayan alanların havadan alınmış görüntüleri ile yapay sinir ağı eğitilmiştir. Kullanılan veri kümesinde toplam 5254 görüntü bulunmaktadır. Bu veri kümesinin %22'si test için ayrılmıştır. Çalışma kapsamında, kalabalık alanları niteleyen bir ısı haritası üretilmiştir. Uçuş yapılan alanların haritalarına ek bir gösterge olarak bahsi geçen ısı haritasının eklenmesinin uçuş yapılan bölgenin incelenmesinde ve uçuşa yasak bölgelerin belirlenmesinde fayda sağlayacağı ileri sürülmüştür.

Somaldo *et al.* (2020) tarafından yapılan çalışmada, Covid-19 salgınına karşı sosyal mesafe kuralının denetimi üzerine çalışılmıştır. Covid-19 salgınının yayılmasını kolaylaştırmamak için sosyal mesafe kurallarının ihlalini tespit eden bir İHA önerilmiştir. İHA; YOLOv3 ile insan, insan topluluğu ve kural ihlallerini tespit edebilmektedir. Çalışma neticesinde insan topluluklarının %90 doğruluk ile tespiti başarılmıştır.

Peng *et al.* (2018) tarafından yapılan çalışmada, otopark alanlarının durumunu tespiti için İHA'lar ile bir çözüm ileri sürülmüştür. Sıradan bir otoparkta kısıtlı görüş açısı sebebiyle sabit kameralar yerine havadan alınan görüntülerin daha iyi sonuçlar üreteceği belirtilmiştir. Özetle çalışma, havadan aldığı görüntüler ile park yerlerini ve bu park yerlerinin dolu/boş durumlarını tanımlayabilen İHA tabanlı bir uygulama önermektedir.

Al-Thani *et al.* (2020) tarafından yapılan çalışmada, küçükbaş hayvancılık ile alakalı İHA tabanlı bir çözüm önerilmiştir. Yapılan çalışmada küçük baş hayvanların açık alanda tespiti ve sayımı için İHA kullanılmıştır. Tespit metodu olarak YOLO tercih edilmiştir.

Qin *et al.* (2021) tarafından yapılan çalışmada, engebeli tarım arazilerinde İHA ile ilaçlama yapmanın getirdiği zorluklara değinilmiştir. Çalışmada YOLO'dan esinlenilerek özel bir evrişimli sinir ağı modeli üzerine çalışılmıştır. Bu model ile engebeli tarım arazilerinde ağaçların tespiti ve pestisit püskürtme işlemi için İHA tabanlı bir çözüm önerilmiştir. Hazırlanan CNN modeline ag-YOLO adı verilmiştir. Bu model kullanılarak 36,5 FPS ve %92 F1-Score başarılmıştır.

Gupta and Verma (2022) tarafından yapılan çalışmada, trafik gözetimi üzerinde İHA tabanlı bir çözüm önerilmektedir. Trafikte yaşanan problemlerin doğru gözetim ile kolayca önlenebilecek bazı ölümcül kazalara yol açabildiği ileri sürülmüştür. İHA ile gözetim ve araç tespiti üzerine yapılan çalışmada farklı CNN metotları üzerinde durulmuş ve YOLO v4 ile minimum %88 mAP değeri başarılmıştır.

Vohra et al. (2022) tarafından yapılan çalışmada, İHA ve uydu fotoğrafları kullanılarak trafik gözetimi üzerine çalışılmıştır. Çalışmada araba, otobüs, kamyon gibi nesnelerin havadan alınan görüntülerde tespit edilmesi için YOLO v4 kullanılmıştır. Çalışmada trafik ışıklarının hala zaman ayarlı tetikleyiciler ile kontrol edildiği, bunun İHA'lar aracılığıyla havadan alınan görüntülerden ve yapay zekadan faydalanılarak çok daha verimli bir şekilde kontrol edilebileceği ileri sürülmüştür.

Sadykova et al. (2019) tarafından yapılan çalışmada, elektrik iletim hatlarında bulunan izolatörlerin tespiti üzerine İHA tabanlı bir çözüm önerilmiştir. Yüksek gerilim hatlarında izolatörlerin manuel olarak kontrol edilmesi hem yüksek maliyetli hem de yüksek riskli bir işidir. İzolatörlerin durumlarının insan gücü yerine İHA'lar ile tespit edilmesinin daha faydalı olacağının ileri sürüldüğü çalışmada, YOLO v2 ile izolatör tespiti üzerine çalışılmıştır.

Hosseiny et al. (2020) tarafından yapılan çalışmada, tarım arazilerinin İHA ile alınan görüntüleri üzerinde çalışılmıştır. Tarım arazilerinin bakımı ve haritalanması gibi ihtiyaçların giderilmesi için İHA'ların kullanılması bir çözüm olarak sunulmuştur. İHA görüntülerinde ekinlerin tespiti üzerinde durulan çalışmada R-CNN ve ResNet-101 tabanlı bir metot ile %90,9 doğruluk başarılmıştır.

Goyal et al. (2020) tarafından yapılan çalışmada, orman yangınlarının erken tespiti üzerine İHA tabanlı bir çözüm önerilmiştir. Orman yangınlarının erken tespiti ile can ve mal kayıplarının önüne geçilebilmek için YOLO tabanlı bir yapay sinir ağı modeli eğitilmiştir. Yapılan testlerde yaklaşık %90 doğruluk başarılmıştır.

Jiang et al. (2022) tarafından yapılan çalışmada, İHA ile termal infrared görüntüler üzerine çalışılmıştır. Görüntüler üzerinde araba ve insan gibi modellerin tespiti için YOLO v3, YOLO v4 ve YOLO v5 evrimsel sinir ağı modelleri eğitilmiş ve karşılaştırılmıştır. Karşılaştırmada söz konusu YOLO versiyonlarının farklı varyasyonlarına da yer verilmiştir. Çalışmada İHA'lar ve uzaktan kontrol edilebilen sistemlerde nesne tespitinin oldukça sık kullanıldığından bahsedilerek en büyük zorluklardan birinin veri kümesi oluşturulması olduğundan bahsedilmiştir. Çalışma kapsamında gündüz ve gece şartlarında termal infrared görüntülerin birbirlerinden farkları üzerinde durulmuştur. Çalışmada modellerin eğitimi için

veriler toplanmış ve ön işlem den geçirilerek 9216 görselden oluşan bir veri kümesi oluşturulmuştur. Veri kümesindeki 1360 görsel test için, geri kalanlarsa modellerin eğitimi için kullanılmıştır. Çalışma neticesinde karşılaştırılan YOLO versiyon ve varyasyonları arasında YOLO v3 en yüksek ortalama doğruluk değerini göstermiştir. En yüksek performans ise YOLO v4 Tiny ve YOLO v3 Tiny modelleriyle elde edilmiştir. Çalışmada elde edilen en yüksek mAP değeri %88,69 olmuştur. En yüksek FPS değeri ise 50 olarak ölçülmüştür.

Liao *and* Juang (2022) tarafından yapılan çalışmada, atık yönetimi konusu üzerine İHA'lar kullanılarak bir çözüm ileri sürülmüştür. Çalışmada özellikle sahil kesimlerinde ve sığ sulardaki atıkların tespiti üzerine durulmuş ve atık tespiti için YOLO metodlarından faydalanılmıştır. Atıkların tespit edilmesi durumunda uçuş yapılan bölgede atık durumunu niteleyen bir ısı haritası üretilmesi de sağlanmıştır. Yapılan uygulamada İHA nesne tespiti ve ısı haritalarının oluşturulması sorumluluğunu uzaktaki bir sunucuya devretmektedir. Örnek senaryo olarak sunulan çözümde İHA tarama yapması için belirlenen rotada uçuş yapmakta ve önceden belirlenmiş rota üzerindeki noktalarda fotoğraf alarak bu fotoğrafı sunucuya göndermektedir. Sunucu tarafında atık tespiti işlemi gerçekleştirilmekte ve atık tespit edilirse bu durum haritalanmaktadır. İHA uçuş rotasındaki son noktaya ulaştığında ise kalkış yaptığı noktaya dönerek iniş yapmaktadır. Çalışmada ileri sürülen çözümün atık yönetiminde gerekli iş gücünün azaltılması, verimliliğin artırılması ve idari makamlara gerekli istatistiki bilgilerin sağlanması açısından faydalı olacağı ileri sürülmüştür.

Petso *et al.* (2021) tarafından yapılan çalışmada, İHA ile hayvan ve hayvan sürülerinin tespiti üzerine çalışılmıştır. Çalışmada farklı yüksekliklerden alınan görüntülerin nesne tespiti üzerine olan etkisi incelenmiştir. Nesne tespiti için YOLO v3 ve YOLO v4 yapay sinir ağı mimarilerinden faydalanılmıştır. YOLO versiyonları üzerinde yapılan özelleştirmeler neticesinde daha yüksek performans gösterecek şekilde model mimarilerinin iyileştiği görülmektedir. Çalışmada, YOLO modellerinin eğitilebilmesi için bir veri kümesi toplanmıştır. Veri kümesinin toplanabilmesi için 8585 hektarlık bir arazide İHA ile uçuşlar düzenlenmiştir. Söz konusu uçuşlar 15 metreyle 130 metre arasında değişen irtifalarda gerçekleştirilmiştir. Veri kümesinde zürafa, zebra, gergedan ve antilop olmak üzere 4 hayvan türü üzerinde durulmuştur. Bu türlerin farklı uçuş irtifalarında İHA'nın sebep olduğu gürültü gibi sebeplerden etkilendikleri görülmüş ve türlerin tepki gösterdikleri irtifalar belirlenmiştir. Vahşi hayatın gözlemlenebilmesi açısından bir örnek teşkil eden çalışmada 90 metre irtifadan uçuş yapılması durumunda canlıların İHA'ya tepki göstermediği ortaya çıkmıştır. Bununla birlikte 30 metre irtifadan yapılan uçuşlarda tüm türlerin İHA'ya tepki gösterdiği belirtilmiştir. Neticede İHA'lar ile vahşi hayatın gözlemlenmesi durumunda, özellikle sürü

halindeki türlerin tespitinde, yüksek irtifa kaynaklı düşük doğruluk değerlerinin üretildiği ortaya çıkmıştır.

Pham et al. (2021) tarafından yapılan çalışmada, otonom İHA yarışları üzerinde durulmuştur. İHA'nın yarış ortamında içerisinden geçmesi beklenen çerçevelerin tespit edilebilmesi için GateNet adındaki yapay sinir ağı modeli ileri sürülmüştür. Özellikle yüksek hızda yapılması beklenen uçuşlarda söz konusu çerçevelerin yüksek performansla tespit edilebilmesi elzemdir. Çalışmada ileri sürülen GateNet modeli çeşitli çalışmalar ile kıyaslanmış ve kıyaslanan çalışmalara göre 57 FPS gibi bir başarımla çok daha yüksek bir performans gösterdiği ileri sürülmüştür. Yapılan testlerde bahsi geçen çerçevelerin İHA'ya olan uzaklığı, görüntüdeki çerçeve sayısı, çerçevelerin tamamının mı yoksa yalnızca bir bölümünün mü görüldüğü gibi farklı durumlar da ele alınmıştır.



MATERYAL VE METOT

Çalışma kapsamında geliştirilen İHA'nın hedef tespiti ve takibi görevlerini yerine getirebilmesi için yazılımsal gereksinimlerinin yanında mekanik, yapısal ve elektronik/aviyonik gereksinimlerin de giderilmesi gerekmektedir.

Bu bölümde mekanik, yapısal, aviyonik ve yazılımsal açılardan İHA'nın görev gereksinimlerine uygun inşası için yapılan tasarım üç ana başlık altında gruplandırılarak sunulmuş ve gerçekleştirilen İHA konfigürasyonu açıklanmıştır: İlk başlık altında İHA'nın mekanik, aviyonik ve yapısal olarak tasarımı, ikinci başlık altında İHA ve yer istasyonu için geliştirilmiş yazılımların mimarisi, üçüncü başlık altında derin öğrenme ile nesne tespiti ve takibi konuları gerekli temel bilgiler ile açıklanmıştır.

İHA'nın Mekanik, Yapısal ve Aviyonik Tasarımı

İHA0 ve İHA1 sınıflarındaki 20 kg'dan daha az azami kalkış ağırlığına sahip görece küçük denebilecek döner kanatlı İHA'ların tasarımında izlenen metotlar, pek çok serbest İHA topluluğunda kullanılan oldukça yaygın yöntemlerdir. Söz konusu tasarım yöntemlerinin detaylarından bahsetmeden önce bu yöntemlerde kuvvet birimleri olarak kütle birimlerinin kullanıldığını belirtmek gerekir. Bunun sebebi tasarım aşamasında hesaplanacak kuvvetlerin yerçekimi ivmesiyle olan ilişkilerinin formüllerde sadeleştirilmesidir. Daha basitçe ifade etmek gerekirse; kuvvetlerin hesaplanmasında, yer çekimi ivmesi ile kütlenin çarpılması gereksinimi serbest İHA topluluklarında kullanılmamaktadır. Bu durum o kadar yaygınlaşmıştır ki İHA'larda kullanılan motorların üreticileri dahi teknik dokümanlarda itki kuvvetini kütle birimleriyle belirtmektedir. Dolayısı ile motorun üreteceği itki kuvveti newton cinsinden değil gram cinsinden açıklanmaktadır. Bu çalışmada da ilgili hesaplamalarda bu geleneğe uygun davranılmış ve itki kuvveti gibi kuvvet birimleri, kütle birimleri ile gösterilmiştir.

Bahsi geçen yaygın tasarım metotlarında, öncelikle görev isterleri için gerekli malzemelerin seçimi yapılmaktadır. Görev gereksinimlerini karşılayan malzemeler belirlendikten sonra bu malzemelerin üretici dokümanlarına bakılarak toplam ağırlık tahmin edilir. İHA'nın tahmini ağırlığında istenen uçuş süresini elde etmek için itki sistemine ilişkin malzemelerde, gerekiyorsa, değişiklik yapılır. Bu değişiklik neticesinde tahmini ağırlık değişeceğinden yeniden hesaplamalar yapılır ve istenen değerler sağlanıncaya kadar farklı

malzemeler ile bu işlem tekrar edilir. Dolayısı ile söz konusu sınıflardaki İHA’larda tasarımın iteratif bir optimizasyon süreci olduğu söylenebilir.

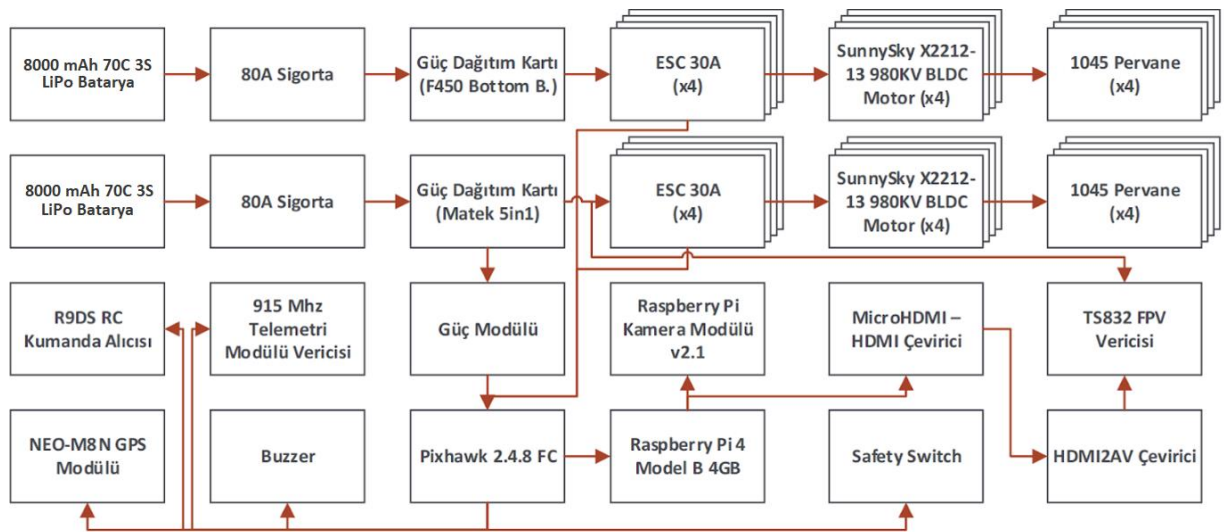
İHA’da Kullanılan Malzemeler ve Tahmini Gövde Ağırlığı

Çalışma kapsamında geliştirilen İHA’nın malzemelerinin belirlenmesi için öncelikle yerine getireceği görev isterleri düşünülmelidir. Seçilen malzemelerin söz konusu görev isterlerini yerine getirecek kabiliyetleri sağlaması gerekmektedir.

Bu çalışmada İHA’nın görevi, düşük maliyetli işlem birimi üzerinde hedef tespiti ve takibi işlemlerini tümleşik bir şekilde yerine getirmesidir. Ayrıca yer birimleri (yer kontrol istasyonu ve pilot) ile sürekli bir şekilde haberleşmeli, gerçek zamanlı olarak görüntü ve veri akışı sağlamalıdır. Tüm bunların test edilmesi için yeterli süre uçuş yapabilmesi ve başarısız testlere karşın İHA’nın harcanabilir olması gerekmektedir.

İHA’nın harcanabilir olması, muhtemel kaza veya kırım durumlarında İHA’nın onarımının yüksek maliyete sebep olmaması ile sağlanmaya çalışılmıştır. Bu sebepten çalışma kapsamında malzeme seçimi yapılırken az sayıda maliyeti yüksek malzeme kullanmak yerine çok sayıda ucuz malzeme kullanılmasına ve malzemelerin yedeklerinin de temin edilmesine karar verilmiştir. Bu sayede bir kaza veya kırım durumunda malzemelerden kayıp olanların yeri hızlıca doldurulabilecek ve söz konusu durumlar yüksek maliyete sebep olmayacaktır.

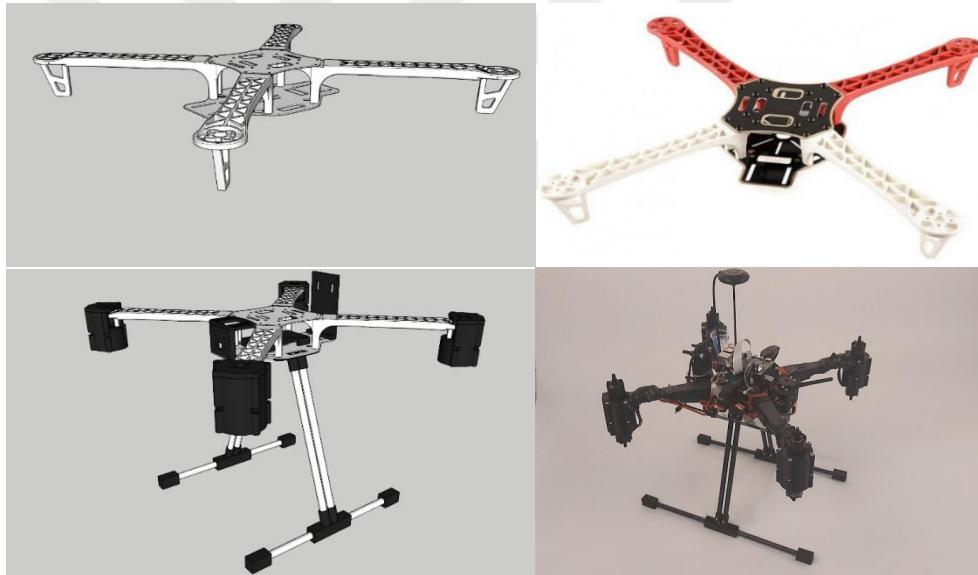
Söz konusu görev isterlerini yerine getirebilmesi için İHA üzerinde kullanılan malzemeler (Şekil 1), bu bölümde gruplandırılarak ayrı başlıklar altında açıklanmış ve seçimlerinde göz önüne alınan kriterler temel bilgiler ile sunulmuştur.



Şekil 1. Kavramsal İHA tasarımındaki aviyonik birimler ve birbirleri ile bağlantılarının genel diyagramı

Gövde Seçimi ve Gövde İyileştirmeleri

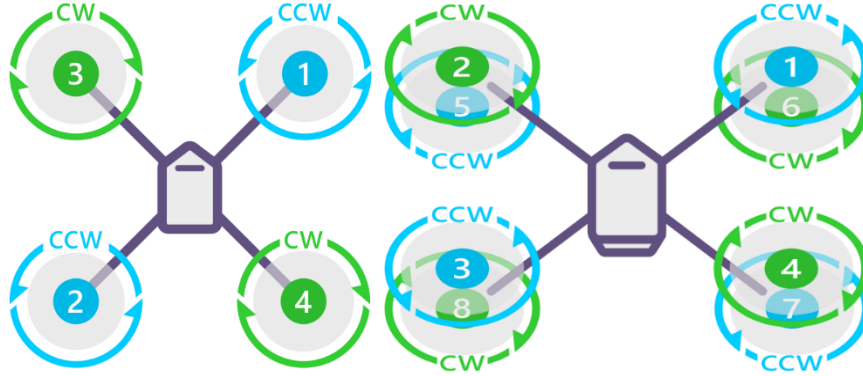
Döner kanatlı İHA'ların malzeme seçiminde öncelik, genellikle tüm platformun üzerine kurulacağı gövdeye verilmektedir. İHA gövdeleri motor sayıları ve yapısal biçimlerine göre farklılık göstermektedir. Ayrıca bu gövdelerin üretildikleri malzemeler ve üretim metotları gövde maliyetini değiştirmektedir. Çalışma kapsamında geliştirilen İHA için DJI Flame Wheel marka F450 gövde tercih edilmiştir. F450 gövde, plastik enjeksiyon metodu ile naylon plastikten üretilen ve gövde merkezinde hafif bir malzemeyle mukavemeti sağlamak için PCB (Printed Circuit Board) kullanan bir gövdedir. Ayrıca söz konusu PCB'lerden birisi olan bottom board, güç dağıtım kartı olarak da kullanılmaktadır. Bu gövdenin tercih edilmesindeki en büyük sebep maliyeti düşürmek ve kaza veya kırım durumlarında kaybedilecek malzemenin yüksek maliyete sebep olmamasıdır. Şekil 2'de bu çalışmada kullanılan gövde ve gövde iyileştirmelerinin tasarım ve fotoğrafları gösterilmiştir.



Şekil 2. F450 gövdenin 3B (3 Boyutlu) tasarımı (sol üstte), F450 gövdenin fotoğrafı (sağ üstte), gövde için tasarlanan iyileştirmeler (sol altta) ve çalışma neticesinde üretilen İHA'nın fotoğrafı (sağ altta) gösterilmiştir.

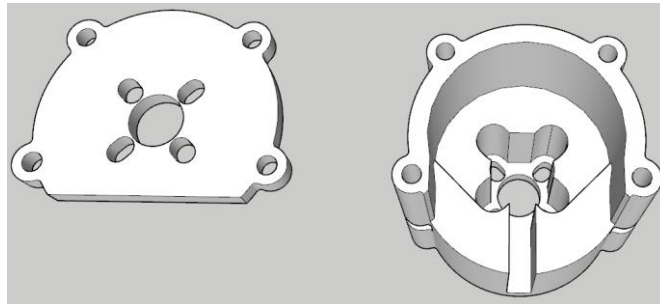
F450 gövde İHA gövdesinin merkezini oluştursa da görev isterlerini karşılaması için iyileştirilmiştir. F450 gövde bir quad X quadcopter gövdesidir. Yani gövde, dört motor ile çalışacak şekilde tasarlanmıştır. Görev isterlerini karşılayacak aviyonik bileşenlerin ağırlıklarının yalnızca dört motor ile taşınması durumunda motor verimliliğinin düşük ve uçuş süresinin çok kısa olacağı yapılan hesaplamalar neticesinde anlaşılmıştır. Bu sebepten F450 gövde iyileştirmelerinden ilki, motor sayısını iki katına çıkaracak bazı yapısal modifikasyonlar olmuştur. Bu durumda F450 gövdesi, quad X biçimli bir quadcopter

gövdesinden octo quad X8 gövdesine sahip bir octocoptere yükseltilmiştir (Şekil 3). Bu sayede gövdeye bağlı kol sayısı arttırılmadan motor sayısı arttırılmış ve motor başına düşen gövde ağırlığı azaldığından motor verimliliğinde ve uçuş süresinde artış sağlanmıştır.



Şekil 3. Quad X quadcopter gövdesinin (solda) ve octo quad X8 gövdesinin (sağda) temsili görselleri ve motor dönüş yönleri. (Anonymous, 2016)

F450 gövdesinde motor sayısını iki katına çıkarmak için her bir kola ikinci bir motorun montajlanabileceği parçaların üretilmesi gerekmektedir. Kollarda ikinci motorlara gerekli montaj alanını sağlayabilmek için iki parçadan oluşan bir iyileştirme tasarlanmıştır (Şekil 4). Bu parçalardan kol üstüne monte edilecek olan birinci parça, motor ile kol arasına sıkıştırılacak şekilde tasarlanmıştır. Bu sayede motorun kola montajı aynı zamanda ikinci motorun monte edileceği parçayı gövdeye bağlamak için gerekli bağlantı noktalarını da oluşturmuştur. Kol altında konumlandırılacak motor ikinci ve ilkinе göre daha büyük olan parçaya monte edilir. Son olarak motor alt ve üst bölümlerindeki tasarlanmış ve üretilmiş parçalar birbirlerine 4 adet m3 vida ile bağlanır.



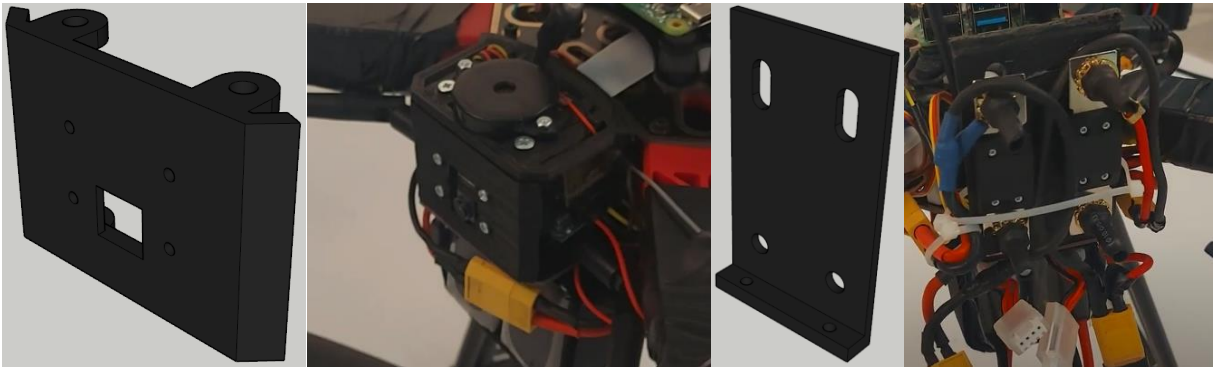
Şekil 4. Motor sayısını arttırmak için tasarlanmış parçalar.

Gövde içerisinde güç dağıtım kartını yalıtırmak ve daha fazla montaj alanı elde etmek için uçuş kontrol kartının altına yerleştirilmek üzere bir levha tasarlanmış ve üretilmiştir. Bu levha sayesinde gövdenin alt katına diğer elektronik bileşenlerden soyutlanmış bir güç dağıtım kartı daha yerleştirilmiştir. Ayrıca daha fazla montaj alanının sağlanması ve gövde üzerinde işlem biriminin montajlanabilmesi için titreşim önleyici bir levha kullanılmıştır. İşlem birimi için kullanılan levha üretilmemiş olup temin edilmiştir. Montaj alanının artırılması için son olarak İHA'nın ön tarafında bulunan kamera modülünü taşıyacak levhanın üzerine montajlanmak üzere bir ızgara tasarlanıp üretilmiştir. Şekil 5'de bahsi geçen levhalar sırasıyla gösterilmiştir.



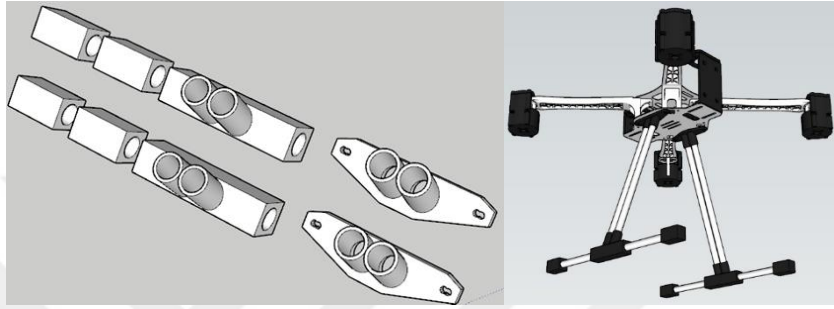
Şekil 5. İHA gövdesindeki montaj alanının artırılması için kullanılan levhalar. Solda gövde alt katında kullanılan levha, ortada gövde üst katında işlem birimi için kullanılan levha ve sağda kamera modülünün üstünde kullanılan ızgara gösterilmiştir.

İHA'nın ön tarafında sabit bir kamera modülü olması planlanmıştır. Bu kameranın gövde üzerinde istenen yere tam olarak yerleştirilmesi için, dikey durmak üzere F450 gövdeye uygun bir levha tasarlanmış ve üretilmiştir. İHA'nın arka tarafında sigortaların yerleştirilebilmesi için de bir levha tasarlanmış ve üretilmiştir. Sigortalar için üretilen bu levha İHA'nın arka tarafında kablo karmaşasını önlemek için uygun bir cep oluşturmaktadır. Ayrıca önceden bahsi geçmiş olan ızgara söz konusu kamera modülü levhası üzerine monte edilmektedir. Şekil 6'da kamera ve sigortaların montajlanması için kullanılan levhaların tasarım ve fotoğrafları paylaşılmıştır.



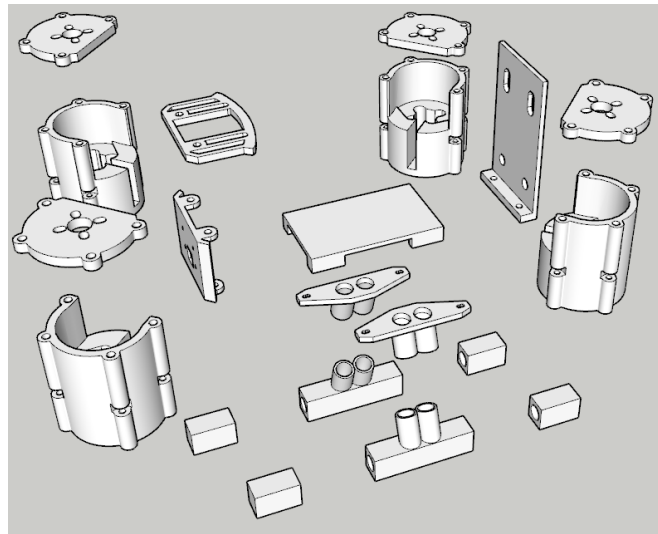
Şekil 6. Kamera ve sigortaların montajlandığı levhaların 3B tasarımları ve çalışma neticesindeki fotoğrafları.

Son olarak İHA için dayanıklı bir iniş takımı üretilmiştir. Bataryalar İHA'nın altında sarkar şekilde konumlandırılacağından yeterli alanı sağlayacak bir tasarıma ihtiyaç duyulmuştur. Bu ihtiyacı karşılayacak iniş takımı temin edilemediğinden üretilmesine karar verilmiştir. Mukavemeti sağlayabilmek için karbonfiber borular kullanılmıştır. Karbonfiber boruların birbirlerine ve gövdeye bağlanması için plastik dirsekler tasarlanmış ve üretilmiştir. Ayrıca iniş takımının ayak uçlarına dengeyi sağlamak için küçük parçalar da üretilerek eklenmiştir (Şekil 7).



Şekil 7. İniş takımında kullanılan parçaların tasarımı (solda) ve iniş takımının montajlanmış hali (sağda).

Sonuç olarak İHA gövdesinin çekirdeğini F450 gövde oluştursa da motor verimliliğini arttırmak, uçuş süresini arttırmak, gövde üzerindeki bileşenlerin yerleştirilecekleri alanları sabitlemek, bazı elektronik bileşenlerin arasında yalıtımı sağlamak ve iniş takımı ihtiyacını gidermek için pek çok özgün parça tasarlanmış ve üretilmiştir (Şekil 8).



Şekil 8. F450 gövdesi üzerinde kullanılmak üzere üretilen parçaların bir arada gösterimi.

Bu parçalar 3B yazıcı vasıtası ile PLA (Polilaktik Asit) malzemeden üretilmiştir. Üretimde 0,4 mm nozzle, 0,3 mm katman kalınlığı ve %20 doluluk değerlerinin kullanılması uygun görülmüştür. Bu değerlerin tercih edilmesinin sebebi PLA malzeme ile üretilecek parçaların yeterince dayanıklı ve mümkün olduğunca hafif olmasını sağlamaktır.

İşlem Birimleri, Kamera ve Çevre Birimleri

İHA üzerinde iki adet işlem birimi bulunmaktadır. Bu işlem birimlerinden ilki mikrodenetleyici tabanlı uçuş kontrol kartı olan Pixhawk 2.4.8'dir. Diğer işlem birimi ise yoldaş bilgisayar (companion computer) olarak Raspberry Pi 4'tür. Şekil 9'da işlem birimlerinin fotoğrafları paylaşılmıştır.



Şekil 9. Pixhawk 2.4.8 uçuş kontrol kartı (solda) ve Raspberry Pi 4 (sağda) fotoğrafları.

Pixhawk uçuş kontrol kartı üzerinde açık kaynak kodlu ArduCopter aygıt yazılımı (firmware) bulunur. Ayrıca uçuş kontrol kartı, üzerinde dahili IMU (Inertial Measurement Unit) algılayıcılarını barındırır. Bu dahili algılayıcılar ve yazılım ile İHA'nın uçuş için gerekli temel fonksiyonlarını kontrol eder. Aynı zamanda telemetri verilerini seri portlarından yayınlamaktadır. Yoldaş bilgisayar ise çalışma kapsamında geliştirilen yazılımı barındırır. Yoldaş bilgisayar, uçuş kontrol kartından aldığı telemetri verilerine ve ürettiği sonuçlara göre uçuş kontrol kartına komutlar gönderir. İlerleyen bölümlerde geliştirilen yazılımın mimarisi daha detaylı bir şekilde açıklanmıştır.

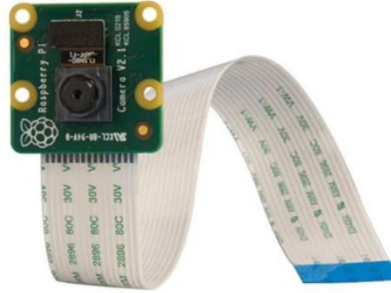
İşlem birimlerinin görev isterlerini karşılayabilmesi için bazı algılayıcılara sahip olması gerekmektedir. Öncelikle İHA'nın çevresel farkındalığında en büyük ihtiyacı mevcut konumunu belirleyebilmesidir. Bunun için gövde üzerine bir NEO M8N GPS (Global Positioning System) modülü yerleştirilmiştir. Bu sayede İHA'nın yaptığı manevralarda dahili IMU algılayıcılarıyla birlikte GPS modülünü de kullanarak kendi konumunu daha doğru bir şekilde belirleyebilmesi sağlanmıştır. Ayrıca GPS modülü sayesinde İHA, önceden belirlenmiş bir koordinata gidebilme, kalkış yaptığı noktaya dönerek otonom iniş yapabilme

ve GeoFence gibi kabiliyetler edinmiştir. GPS modülü radyo frekans (RF) gürültüsüne karşı oldukça hassastır ve gürültü sebebiyle hatalı sonuçlar üretebilmektedir. Bu sebepten GPS'in alt kısmı, İHA üzerindeki diğer bileşenlerin sebep olduğu gürültüden daha az etkilenmesi için alüminyum ile kaplanmıştır (Şekil 10).



Şekil 10. İHA gövdesindeki GPS modülünün ve modül altındaki RF kalkanının fotoğrafları.

İHA'nın üzerinde görü işlemleri için bir adet Raspberry Pi kamera modülü (Şekil 11) kullanılmıştır. Bu modül hem hafif hem de montajı kolay olduğundan dolayı tercih edilmiştir. Dezavantajı ise kameranın işlem birimine bağlantısı için kullanılan esnek kablonun çok fazla RF gürültüsüne sebep olmasıdır. Bunun için GPS modülünde yapıldığı gibi bu kablonun etrafı da gürültüyü önlemek için alüminyumla kaplanmıştır.



Şekil 11. Raspberry Pi kamera modülü ve esnek kablosunun fotoğrafı.

Pixhawk uçuş kontrol kartına bataryadan, doğru gerilim ve akım değerlerinde, enerji verilmesi ve batarya durumunun uçuş kontrolcüsü tarafından izlenebilmesi için bir adet 3DR güç modülü kullanılmıştır. Güç modülü, bataryadan aldığı enerjiyi 5V gerilim ve 2A akım değerlerinde uçuş kontrol kartına iletmekle yükümlüdür. Batarya ile arasına sigorta dışında üçüncü bir birim yerleştirilmez. Batarya emniyeti için oldukça önemli bir bileşendir ve batarya seviyesine göre tetiklenecek işlem ve manevraların ayarlanmasında kullanılır.

Son olarak Pixhawk uçuş kontrol kartının çevre birimleri olarak bir adet buzzer ve bir adet safety switch gövdeye yerleştirilmiştir. Uçuş kontrol kartı İHA gövdesinin alt katında yer alacağından üzerinde bulunan LED'in rengi ve ritmine bakılarak uyarıların anlaşılabilmesi söz konusu olabilir. Bu sebeple buzzer kullanılmış ve LED ile verilen sinyallere benzer şekilde sesli sinyaller alınması sağlanmıştır. Safety switch ise uçuş kontrol kartının kontrolsüz bir şekilde motorları çalıştırmasının önüne geçmek için kullanılmıştır. Safety switch üzerinde bulunan düğmeye basılarak İHA arm veya disarm moduna getirilir. Şekil 12'de bahsi geçen bileşenlerin fotoğrafları sunulmuştur.



Şekil 12. Gövdede kullanılan buzzer (solda), safety switch (ortada) ve güç modülünün (sağda) fotoğrafları.

Haberleşme Birimleri

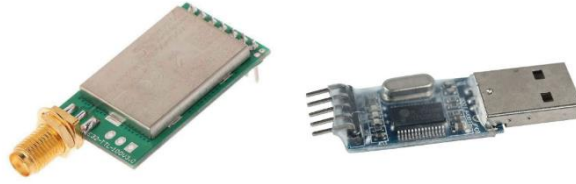
İHA durumunun anlık olarak takip edilebilmesi için farklı görevleri üstlenen birden çok haberleşme birimi kullanılmıştır. Söz konusu haberleşme birimleri; alıcı ve verici olarak biri yer biriminde, diğeri İHA üzerinde olmak üzere iki parça olarak kullanılmaktadır.

Kullanılan haberleşme birimlerinden ilki 915 MHz frekansında çift taraflı olarak uçuş kontrol kartıyla yer kontrol istasyonu arasında kablosuz iletişim sağlayan telemetridir (Şekil 13). Telemetri sayesinde İHA anlık durumunu yer istasyonuna bildirmekte ve yer istasyonu üzerinden İHA'nın kontrolü sağlanabilmektedir.



Şekil 13. Telemetri alıcı ve vericisinin antenleriyle birlikte fotoğrafı.

Telemetri ile benzer bir şekilde kullanılan diğ er bir haberleşme birimi ise LoRa (Long Range) mod ul udur (Şekil 14). LoRa mod ul u, 433 MHz frekans bandında half duplex olarak yer kontrol istasyonu ile yoldaş bilgisayar arasında veri alışverişine olanak sağlamaktadır. İHA'nın  zellikle hedef tespit ve takibine ilişkin mevcut durumunu yer kontrol istasyonuna iletmesi gerekmektedir. U uş kontrol kartı  zerinde bulunan telemetri, aygıt yazılımında değışiklik yapılmasından ka ınmak amacıyla, bu iřlem i in kullanılmamış bunun yerine yeni bir haberleşme biriminin dođrudan yoldaş bilgisayara bađlanması uygun g r lm şt r. LoRa mod ul unun yer istasyonuna ve iřlem birimine bađlantısının sađlanabilmesi i in PL2303 USB-TTL d n şt r c  kullanılmıřtır (Şekil 14).



Şekil 14. LoRa mod ul u (solda) ve PL2303 USB-TTL d n şt r c  (sađda) fotođrafları.

İHA'nın  zerinde bulunan yoldaş bilgisayarın kamera mod ul unden alınan g r nt y  anlık olarak yer istasyonuna iletebilmesi i in 5.8 GHz frekans bandında video aktarımı yapan TS832 AV (Audio Visual) vericisi İHA  zerine yerleştirilmiřtir. Yer kontrol istasyonunda ise RC832 AV alıcısı ile g r nt  alınmıřtır. S z konusu alıcı ve vericiler, aynı zamanda FPV (First person view) sistemi olarak da bilinmektedir (Şekil 15). Ayrıca menzili diğ er haberleşme birimlerine g re  ok daha kısadır.



Şekil 15. FPV alıcı ve vericisinin bađlantı kablolarıyla birlikte fotođrafi.

TS832 video vericisi, AV formatında veri ile çalışmaktadır. İHA üzerinde yoldaş bilgisayarın HDMI (High Definition Multimedia Interface) portundan alınan görüntünün AV formatına dönüştürülmesi için HDMI2AV dönüştürücü kullanılmıştır. Yer kontrol istasyonunda ise RC832 ile yakalanan AV formatındaki görüntü EasyCAP ile yer kontrol istasyonuna seri porttan iletilmiştir. Dolayısı ile yer kontrol istasyonu olarak kullanılan bilgisayarda, İHA üzerinden alınan görüntünün yakalanması için bağlanan donanım, herhangi bir web kamerası gibi davranmıştır. Şekil 16'da görüntünün yer istasyonuna iletilmesinde kullanılan dönüştürücülerin fotoğrafları paylaşılmıştır.



Şekil 16. EasyCAP (solda) ve HDMI2AV Dönüştürücü (sağda) fotoğrafları.

Son olarak pilot ile uçuş kontrol kartı arasında iletişimi sağlamak için AT9S RC (Remote Control) kumanda vericisi ve R9DS RC kumanda alıcısı kullanılmıştır. Bu bileşenler, pilota ihtiyaç duyulması halinde, İHA'nın elle kontrolü için kullanılmaktadır. R9DS isimli uzaktan kumanda alıcısı, İHA gövdesine yerleştirilmiştir. Söz konusu RC kumanda alıcısı, pilotun kullandığı AT9S kumanda ile 2.4 GHz frekans bandında haberleşmektedir. Bu haberleşmede; 9 kanal üzerinden alınan PWM (Pulse Width Modulation) sinyalleri, R9DS tarafından SBUS (Serial Bus) sinyali olarak uçuş kontrol kartına iletilmektedir. Dolayısı ile burada pilotun 9 farklı kontrol ile İHA'yı yönettiği söylenebilir (Şekil 17).



Şekil 17. AT9S RC vericinin (solda) ve R9DS RC alıcının (sağda) fotoğrafları.

Sonuç olarak haberleşme için İHA ve yer birimleri arasında dört farklı haberleşme aygıtı kullanılmıştır. Her bir aygıt İHA üzerindeki farklı bir haberleşme gereksinimini gidermektedir. Yoldaş bilgisayar ile yer kontrol istasyonu arasındaki iletişim, İHA üzerinden alınan videonun yer kontrol istasyonuna iletilmesi, yer kontrol istasyonu ile uçuş kontrol kartı arasındaki iletişim ve son olarak pilot ile uçuş kontrol kartı arasındaki iletişim için farklı aygıtlar kullanılmıştır. Söz konusu haberleşme birimleri 433 MHz, 915 MHz, 2.4 GHz ve 5.8 GHz bantlarını kullanmaktadırlar.

Güç ve İtki Sistemi Birimleri

İHA'nın görev isterlerini karşılayacak ekipmanlarla birlikte kendi ağırlığını kaldırabilmesi ve görevlerin tamamlanabilmesi için yeterince uzun süre havada kalabilmesi gerekmektedir. Bunun için İHA'nın motorlarının verimli bir şekilde ihtiyaç duyulan süre boyunca yeterli itki kuvvetini üretebilmesi gerekmektedir. İHA üzerinde bulunan itki sistemi bu görevi üstlenmektedir. Güç sisteminin amacı ise itki sistemine ve diğer bileşenlere gerekli enerjiyi uygun değerlerde aktarabilmesidir. Dolayısı ile güç sistemi ve itki sistemi birbirlerine bağımlı sistemlerdir ve birlikte değerlendirilirler. Bu sistemlerin malzeme seçiminde olduğu gibi tasarımsal hesaplamaları da birlikte yapılmalıdır.

İHA'da itki kuvvetini üretebilmek için 8 adet fırçasız DC (BLDC) motor kullanılmıştır. BLDC motor olarak SunnySky X2212-13 980 KV motorlar (Şekil 18) tercih edilmiştir. Burada X2212-13 ibaresi motorun üretebileceği azami itki kuvveti ile alakalı fikir vermektedir. Bir BLDC motor; stator ve rotor olmak üzere iki bölümden oluşmaktadır. X2212-13 ibaresindeki 22 sayısı stator çapının 22 mm olduğunu, 12 sayısı ise stator yüksekliğinin 12 mm olduğunu belirtmektedir. Benzer boyutlarda olan BLDC motorlar, genellikle benzer itki kuvveti üretmektedirler. Bu sebeple kullanılacak motorun itki kuvvetinin ne olduğunu anlamak için farklı markaların benzer boyutlardaki motorları da incelenebilir.



Şekil 18. SunnySky X2212-13 980 KV BLDC motorun fotoğrafı

İtke sisteminde en bilindik bileşen motor olsa da itki sistemi batarya, ESC (Electronic Speed Controller), motor ve pervane dörtlüsünden oluşmaktadır. Çalışma kapsamında 1045 naylon pervaneler (Şekil 19) kullanılmıştır. 1045 ibaresi, motorlarda olduğu gibi pervanenin boyut ve şeklini açıklamaktadır. 1045 ibaresindeki 10 sayısı pervanenin uçtan uca uzunluğunun 10 inç olduğunu göstermektedir. 45 sayısı ise pervanenin eğimini göstermektedir. 45 sayısı arttıkça pervanenin eğimi ve tork artarken RPM (Revolutions Per Minute) düşmektedir.



Şekil 19. 1045 pervane çifti fotoğrafı.

Motora ihtiyaç duyacağı akımı iletmek için ESC kullanılmaktadır. ESC, ağırlığı ve üzerinden geçirebileceği maksimum akım değeri dikkate alınarak seçilir. Çalışma kapsamında 30 A değerinde ESC'lerin (Şekil 20) kullanılması uygun bulunmuştur. Kullanılan motorların maksimum talep edebileceği akım miktarı 15 A olarak üretici firma tarafından yayınlanmıştır. Bununla birlikte bazı çok agresif manevralarda bu akım değeri peak denen zirve değer üzerine çıkabilmektedir. Bu sebepten İHA'larda ESC'nin akım değeri genellikle motorun talep edebileceği maksimum değer üzerinde seçilmektedir.



Şekil 20. İHA'da kullanılmış 30 A ESC'nin fotoğrafı.

ESC'ler ihtiyaç duydukları enerjiyi doğrudan bataryadan almaktadırlar. İHA'larda yüksek akım değerlerini verilebilecek şekilde özelleştirilmiş olan LiPo (Lityum Polimer) bataryalar sıkça tercih edilmektedir. Bataryalar genellikle İHA gövdesindeki en ağır

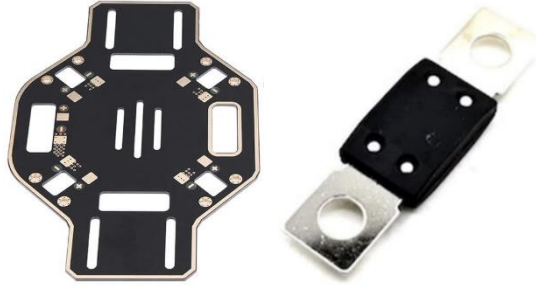
birimlerdir. Bu sebepten batarya seçiminde, bataryanın ağırlığının kapasitesi ile oranlanarak seçilmesi gerekmektedir. Çalışma kapsamında JetFire 8000mAh 70 C 3S1P 11,1 V LiPo batarya (Şekil 21) kullanılmıştır. 8000 mAh, bataryanın kapasitesini; 70 C, anlık alınabilecek maksimum akımı; 3S1P, bataryanın kaç hücreden oluştuğunu ve hücrelerin arasındaki bağlantı tipini; 11.1 V, bataryanın nominal voltajını göstermektedir. İtki sistemine ilişkin bu sayısal değerlerin neleri ifade ettiği ve ilişkili hesaplamalara nasıl etki ettikleri ilerleyen bölümlerde daha detaylı bir şekilde açıklanmıştır.



Şekil 21. JetFire 11.1 V 8000 mAh 70 C 3S1P LiPo bataryanın fotoğrafı.

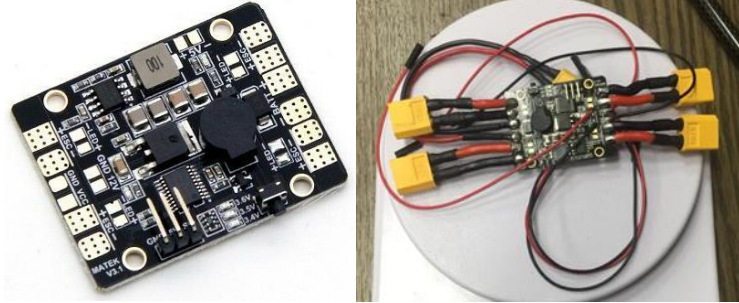
Çalışma kapsamında geliştirilen İHA'nın özgün olmasını sağlayan en önemli özelliklerinden birisi de birbirinden bağımsız iki ayrı güç dağıtım sistemine sahip olmasıdır. İki ayrı güç dağıtım sisteminin, LiPo bataryaların paralel olarak bağlanmaları gereksinimini ortadan kaldırarak bağlantının yapılması esnasında doğan iş güvenliği riskini ortadan kaldırmıştır. Ayrıca uçuş bazı bileşenlere güç verirken bataryalar birbirinin yedeği şeklinde davranmaktadır. Daha fazla güç kapasitesinin sağlanması daha fazla motorun gövdeye bağlanması için olanak sağlamakta ve motorların daha verimli çalışmasına fırsat vermektedir. Bildiğimiz kadarıyla çalışma neticesinde ortaya çıkan İHA sınıfında birden fazla bağımsız güç dağıtım sistemi kullanılan başka bir örneğe rastlanmamıştır.

Geliştirilen İHA'da iki adet LiPo batarya ve güç dağıtım kartı kullanılmıştır. İlk güç dağıtım kartı F450 gövdenin bottom board olarak bilinen güç dağıtım kartıdır (Şekil 22). F450 bottom board, dört adet motor için ESC çıkışları ve batarya için bir giriş barındırır. Burada batarya girişinden alınan güç herhangi bir ara devre elemanına uğramadan doğrudan ESC'lere dağıtılır. Geliştirilen İHA'da ise batarya girişinden önce bir adet 80 A sigorta (Şekil 22) güvenlik gerekçesiyle kullanılmıştır. Bu güç sistemi ile gövdenin alt katındaki dört motora güç sağlanmıştır.



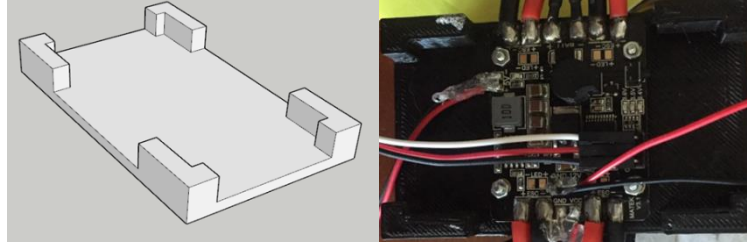
Şekil 22. F450 bottom board (solda) ve 80 A sigorta (sağda) fotoğrafları.

İkinci güç dağıtım sistemi için Matek marka güç dağıtım kartı (Şekil 23) kullanılmıştır. Bu güç dağıtım kartı F450 bottom board gibi dört motor için ESC çıkışları ve batarya için bir giriş sağlamaktadır. Bununla birlikte üzerinde 5 V ve 12 V BEC (Battery Eliminator Circuit) devreleri de barındırmaktadır. Kullanılan bataryaların nominal voltajı 11,1 V olduğundan dolayı 12 V BEC batarya gerilimine eşit olmaktadır. Ayrıca 12 V BEC devresi kullanılırken 5 V BEC’den alınan akım değerinin düştüğü fark edilmiştir. Bu sebepten 12 V BEC devresinin kullanılması gereksiz görülmüştür. 5 V BEC çıkışı ise yoldaş bilgisayar için gerekli olan enerjiyi doğru gerilim değeri ile sağlamak için kullanılmıştır. Burada alınan maksimum akım değeri 3 A olduğundan Raspberry Pi’nin çalışması için yeterli akım sağlanmıştır.



Şekil 23. Matek PDB (Power distribution board) ve çalışmada kullanılmak üzere hazırlanmış halinin fotoğrafları.

Matek güç dağıtım kartı üzerinde ayrıca V_{cc} çıkışı da bulunmaktadır. V_{cc} çıkışına video vericisi olan TS832 bağlanmıştır. 12 V BEC yerine V_{cc} çıkışı kullanılarak hem 5 V BEC çıkışında akımın azalması engellenmiş hem de TS832 için gerekli enerji güç dağıtım kartı üzerinden sağlanmıştır. Matek güç dağıtım kartı gövde üzerindeki montaj alanının verimli kullanılabilmesi ve ağırlık merkezinin gövde merkezinden sapmaması amacıyla erişilmesi zor bir alana yerleştirilmiştir (Şekil 24). Bu sebepten montaj sonrası erişimi kolaylaştırmak için PDB üzerindeki ESC çıkışlarına XT60 konnektörler bağlanmıştır.



Şekil 24. Güç dağıtım kartının altına yerleştirildiği levha ve montajlanışının fotoğrafları.

Batarya ile Matek güç dağıtım kartı arasına, F450 bottom board'da olduğu gibi, 80 A sigorta yerleştirilmiştir. Ayrıca sigortadan sonra 3DR güç modülü de sigorta ile PDB arasına yerleştirilmiştir. 3DR güç modülü (Şekil 25), güç dağıtım sisteminden Pixhawk uçuş kontrol kartına istenen değerlerde güç verilmesini sağlar.

Sonuç olarak iki eşdeğer batarya ile kullanılan İHA'nın yalnızca güç modülüne bağlı olan bataryasının durumu gözlemlenmektedir. Burada durumu gözlemlenen bataryanın gözlemlenmeyen bataryaya göre daha erken biteceği şekilde güç, iki sisteme dağıtılmıştır. Güç dağıtım sistemlerinin tüketeceği enerji miktarlarına ilişkin hesaplamalar, ilerleyen bölümlerde daha detaylı bir şekilde açıklanmıştır.



Şekil 25. 3DR güç modülünün fotoğrafı.

Tahmini Gövde Ağırlığı

İHA tasarımında öncelikle görev isterlerini karşılayacak malzemelerin belirleneceği sonrasında ise tahmini ağırlığın hesaplanacağı önceki bölümlerde belirtilmiştir. Ayrıca İHA üzerinde kullanılacak malzemeler de önceki bölümlerde tanıtılmış ve kullanılması uygun bulunan malzemeler açıklanmıştır. Bu bölümde ise İHA'nın tahmini ağırlığına ilişkin kavramsal tasarım açıklanmıştır. Kavramsal çalışmadan sonra üretilen İHA'nın gerçekleşen ağırlığı ise bu tezin ilgili bölümünde açıklanacaktır.

İşbu çalışma kapsamındaki benzer, görece küçük denilebilecek İHA'larda tasarımın iteratif bir optimizasyon süreci olarak nitelendirilebileceği daha önce söylenmişti. Dolayısı ile seçilen malzemeler görev isterlerini karşılayıncaya ve hedeflenen uçuş süresini yakalayıncaya kadar defalarca değiştirilebilir. Örneğin; bu çalışma kapsamında önce dört motorlu bir

quadcopter konfigürasyonu üzerinde durulmuştur. Ancak seçilen malzemeler ile tahminlere dayalı hesaplamalar neticesinde uçuş süresinin çok az olacağı öngörülmüş ve motor sayılarını artıracak şekilde değişiklik yapılmıştır. Bu sayede motor verimliliği artırılmış ve kabul edilebilir düzeyde uçuş süresi değerleri elde edilmiştir.

Tahmini ağırlık, bahsi geçen hesaplamalarda en çok kullanılan değerlerden birisidir. Tahmini ağırlık belirlenirken seçilmiş malzemelerin üreticileri tarafından yayınlanan dokümanlar incelenir ve verilen ağırlık değeri doğru kabul edilir. Üretici tarafından yayınlanmış veri bulunmadığı durumlarda ise üçüncü kaynaklara başvurulabilir veya benzer özelliklerdeki diğer malzemelerin ağırlıkları araştırılabilir. Malzemelerin ağırlığına ilişkin bulunan veriler dışında vida, kablo, bant gibi diğer ağırlıklar için de bir tahminde bulunulup toplam ağırlığa eklenir. İşbu çalışmanın kavramsal tasarımıyla ortaya çıkan tahmini ağırlık aşağıdaki tabloda gösterilmiştir.

Tablo 2. Kavramsal Tasarımda Ortaya Çıkan Tahmini Ağırlık Tablosu.

İsim	Adet	Birim Ağırlığı*	Toplam Ağırlık*
JetFire 11.1V 70C 8000mAh LiPo Batarya	2	584	1168
30A ESC	8	25	200
Matek 5in1 PDB	1	50	50
3DR Güç Modülü	1	20	20
Telemetry Modülü Vericisi	1	20	20
SunnySky X2212-13 ii 980kv BLDC Motor	8	56	448
Raspberry Pi 4 Model B (4GB)	1	70	70
Raspberry Pi Kamera Modülü v2.1	1	5	5
Raspberry Pi Alüminyum Soğutucu Seti	1	5	5
GPS Modülü	1	30	30
Cırt Cırt Kayış	2	5	10
Pixhawk 2.4.8 Uçuş Kontrol Kart	1	40	40
R9DS RC Alıcısı	1	10	10
TS832 FPV Vericisi	1	35	35
F450 İHA Gövdesi	1	300	300
İniş Takımı	1	150	150
Pervane (1045)	8	10	80
Sigorta (80A)	2	10	20
HDMI2AV Çevirici	1	10	10
Diğer (Vida, somun, buzzer, 3B çıktılar, vs.)	1	379	379
Tüm Malzemeler	-	-	3050

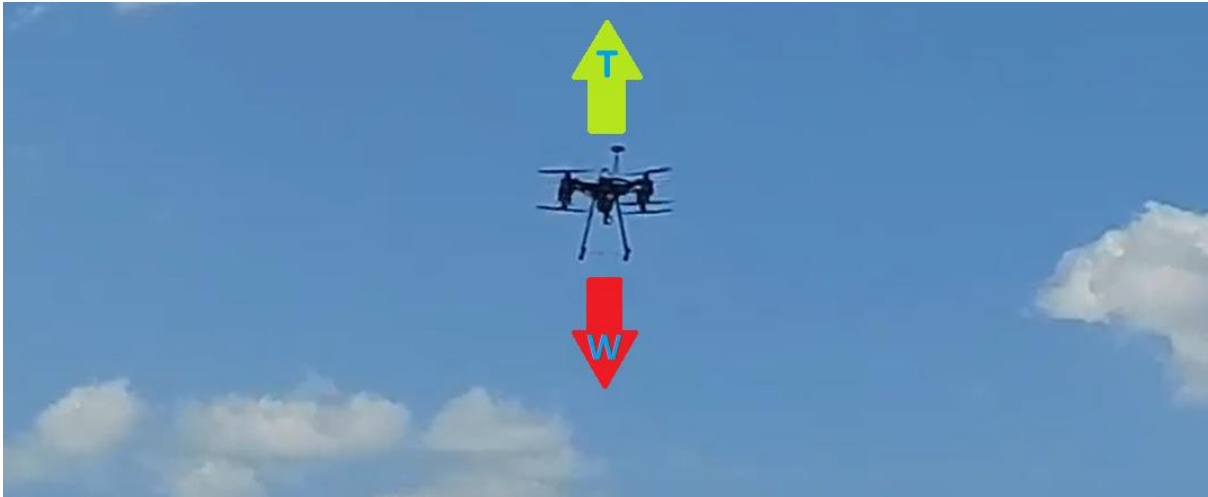
* Gram

Sonuç olarak tasarım aşamasında beklenen gövde ağırlığı 3050 gram olarak tahmin edilmiştir. Bu değer gerçekleşen İHA'nın ağırlığı ile alakalı fikir vermekte ve İHA'nın kabiliyetlerine ilişkin hesaplamalarda temel bir değer olarak kullanılmaktadır.

İHA'nın tahmini ağırlığı belirlendikten sonra motorların ürettiği itki kuvveti ile kıyaslanıp bataryaların kesin koşullarda kaç dakika boyunca enerji verebileceği hesaplanır. Yapılan hesaplamaların neticesiyle görev isterleri uyumsuz ise itki sistemi malzemelerinde değişiklik yapılabilir. Tahmini ağırlığın kullanıldığı ve bahsi geçen hesaplamalar sonraki bölümde açıklanmıştır.

İHA İtki Sistemi ve Azami Uçuş Süresi Hesaplamaları

İHA'nın tasarımında görev isterlerini karşılayan malzemelerin seçimi neticesinde tahmini İHA ağırlığı önceki bölümlerde hesaplanmıştır. İHA'nın uçuş kabiliyetini sağlaması için itki sisteminin İHA ağırlığından daha fazla kuvvet üretmesi gerekmektedir. İtki kuvveti (thrust), İHA üzerindeki 8 motor tarafından üretilmektedir. İHA tahmini ağırlığı ise 3050 gr olarak hesaplanmıştır. Bu durumda İHA'nın askıda kalabilmesi için her bir motorun yaklaşık 380 gr itki kuvveti üretmesi gerekmektedir. Sonuç olarak toplam itkinin ağırlığa oranının (Şekil 26) en az 1 olması gerekir. Bu değere TWR (Thrust to Weight Ratio) denir.



Şekil 26. Askı halindeki İHA üzerine etki eden itki ve ağırlık kuvvetlerinin gösterimi. T (Thrust) itki kuvvetini; W, ağırlığı göstermektedir. Askı durumunda itki, ağırlığa eşittir.

TWR oranı bir İHA'nın agresif manevralara yatkınlığı, hızı ve uçuş süresi ile alakalı fikir vermektedir. TWR değeri tasarımda varılan bir sonuç olmakla birlikte aynı zamanda bir hedeftir. Sıradan döner kanatlı İHA'ların TWR değeri genellikle 2 ile 4 arasında değişmektedir. Yarış gibi sebeplerle özelleştirilmiş İHA'larda ise TWR değerinin 4 ile 11

arasında olması gayet normaldir. İşbu çalışma kapsamında gerçekleştirilen İHA’da tasarım öncesi TWR hedefi minimum 2 olarak hedeflenmiştir.

TWR değerinin hesaplanması için İHA itki sisteminin üreteceği maksimum itki kuvvetinin bilinmesi gerekmektedir. Bunun için kullanılan motorların üreticisi tarafından yayınlanan thrust tablosu birincil kaynaktır. Thrust tablosu (Şekil 27), bir motorun belli batarya ve pervane kombinasyonlarıyla ne kadar itki kuvveti üretebileceğini tanımlar.

APC1047	11.1	0.7	100	7.77	12.87001287	77° 全油门过载
		1.8	200	19.98	10.01001001	
		3	300	33.3	9.009009009	
		4.6	400	51.06	7.833920877	
		6.2	500	68.82	7.265329846	
		8.4	600	93.24	6.435006435	
		10.8	700	119.88	5.839172506	
		13	800	144.3	5.544005544	
		15.5	910	172.05	5.289160128	
APC1038	11.1	0.8	100	8.88	11.26126126	53°
		1.8	200	19.98	10.01001001	
		3.2	300	35.52	8.445945946	
		4.6	400	51.06	7.833920877	
		6.3	500	69.93	7.15000715	
		8.3	600	92.13	6.512536633	
		10.5	700	116.55	6.006006006	
		13.2	800	146.52	5.46000546	
		15	890	166.5	5.345345345	

Şekil 27. SunnySky X2212-13 980 KV BLDC motorun thrust tablosu. Soldan sağa doğru ilk sütunda kullanılan pervane, ikinci sütunda batarya gerilimi (V), üçüncü sütunda anlık akım (A), dördüncü sütunda üretilen itki kuvveti (gr), beşinci sütunda güç tüketimi (Wh), altıncı sütunda motor verimliliği (gr/Wh) ve yedinci sütunda motor sıcaklığı (C°) gösterilmiştir.

İHA’da kullanılacak motorlarda üç hücreli bir batarya ve 1047 pervane ile bir motorun üretebileceği maksimum itki kuvvetinin 910 gram olduğu anlaşılmaktadır. 910 gram itki kuvveti üretirken motorun 15,5 A akıma ihtiyaç duyduğu da resmi dokümanlarda açıklanmaktadır. İHA’da 1045 pervane kullanılacağı önceki bölümlerde açıklanmıştı. Üretici tarafından verilen dokümanda 1045 pervane kullanılmadığından en yakın değerdeki pervane ile üretilen itki kuvveti doğru kabul edilmiştir. Gövde üzerinde 8 motor olacağı ve bir motorun maksimum 910 gram itki kuvveti üreteceği varsayıldığından toplam maksimum itki kuvveti 7280 gram olarak tahmin edilmiştir.

$$TWR = \frac{T}{W} = \frac{7280 \text{ gr}}{3050 \text{ gr}} = 2,39$$

Toplam maksimum itki kuvveti ile tahmini İHA ağırlığı oranlandığında beklenen TWR değerinin 2,39 olduğu hesaplanmıştır. Bu, daha önce bahsi geçen hedeflenen TWR değerinden büyüktür. Yani hedeflenen TWR sağlanmıştır. Ancak yalnızca TWR hedefinin yakalanması, istenen uçuş süresinin de yakalandığı anlamına gelmez. Bu durumda İHA üzerinde tahmini güç tüketimi ve bataryaların kapasitesi ile tahmini uçuş süresi hesaplanmalıdır.

Tahmini uçuş süresi, İHA'nın askı durumu dikkate alınarak hesaplanır. Askı durumu, İHA'nın kendisini havada hareketsiz tutmaya yetecek kadar itki kuvveti ürettiği durumdur. Her bir motorun yaklaşık 380 gram itki kuvveti üretmesi, çalışma kapsamında geliştirilen İHA için askı durumunu sağlamaktadır. Üretici tarafından yayınlanan doküman incelendiğinde bir motorun 3 A akım ile 300 gram, 4,6 A akım ile 400 gram itki kuvveti ürettiği görülmektedir. BLDC motorların devirleri arttıkça verimliliklerinin düşeceği de dikkate alındığında, 380 gram itki kuvveti için her bir motorun 4 A akım çekeceği tahmin edilmiştir.

Daha önceki bölümlerde İHA'da iki bağımsız güç dağıtım sistemi olduğu ve bu güç dağıtım sistemlerindeki bataryalardan birisinin diğerinden önce biteceğinin varsayıldığından bahsedilmişti. Dolayısıyla İHA'nın uçuş süresinin hesaplanmasında erken bitecek olan batarya ve bu bataryadan beslenen dört motor dikkate alınmıştır.

Bahsi geçen güç dağıtım sisteminde motorlar için toplam 11,1 V gerilim ve 16 A akım gerekmektedir. Görev birimlerinden video vericisi olan TS832, 600 mW güce ihtiyaç duymaktadır ve 11,1 V gerilim ile çalışabilmektedir. Bu durumda TS832 yaklaşık 0,05 A akım tüketmektedir. Pixhawk uçuş kontrol kartı yaklaşık 5 V 2 A güç gereksinime ihtiyaç duyar. Bununla birlikte Raspberry için 5 V 3 A güç sağlanmaktadır. Pixhawk uçuş kontrol kartı ve Raspberry Pi için ihtiyaç duyulan akım değerlerinin bataryanın gerilim değerindeki eş değeri 2,25 A olarak hesaplanmıştır. Sonuç olarak askı durumundaki İHA'nın erken bitecek olan bataryadan anlık 18,3 A akım talep edeceği öngörülmüştür.

Uçuş süresinin hesaplanmasında askı durumu göz önüne alınırken, bataryanın uyumluluğu için maksimum değer göz önüne alınır. İHA itki sistemi maksimum güçte çalışırken her bir motor 15,5 A akım çekmektedir. Askı durumu için yapılan hesaplama motorların talep edeceği maksimum akım değeri için yeniden yapıldığında bataryadan maksimum 64,3 A akım talep edileceği öngörülmüştür.

İki durum için de talep edilecek akım değerleri hesaplandıktan sonra kullanılan bataryanın değerlerinin incelenmesi gerekmektedir. Kullanılan batarya 8000 mAh ve 70 C

değerindedir. Burada C değerinin birimi h^{-1} türündedir. Dolayısıyla kapasitenin C değeri ile çarpılması durumunda, bataryanın anlık olarak verebileceği maksimum akım değeri hesaplanabilmektedir.

$$\text{Maksimum Anlık Akım} = AhC = 8Ah \cdot 70C = 560 A, \quad C = h^{-1}$$

Hesaplama neticesinde İHA üzerinde kullanılan bataryaların her birinin anlık olarak maksimum 560 A değerinde akım verebileceği anlaşılmaktadır. Bu durumda, İHA'nın talep edebileceği maksimum akım değerinden (64,3 A) yüksek olduğu için bataryanın 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. Burada belirtmek gerekir ki; bir LiPo bataryanın tüm kapasitesi kullanıldığında batarya hücreleri tekrar kullanılamayacak şekilde bozunuma uğrar. Bu sebepten, acil durumlar dışında, bataryaların tüm kapasitesi kullanılmaz. Genellikle bu değer %80 ile sınırlandırılır. Ancak iki durumda da kaç dakika uçuş yapılacağını bilmek faydalıdır. Aşağıda bataryanın tüm kapasitesiyle kullanıldığı duruma tüm enerji, %80 kapasiteyle kullanıldığı duruma sürdürülebilir enerji denerek ilişkili hesaplamalar yapılmıştır.

$$\text{Tüm Enerji} = AhV = 8Ah \cdot 11,1V = 88,8 Wh$$

$$\text{Sürdürülebilir Enerji} = AhV = \frac{8Ah \cdot 11,1V \cdot 80}{100} = 71,04 Wh$$

Bataryanın tüm enerjisi hesaplandığında 88,8 Wh olduğu hesaplanmıştır. %80 kapasite ile kullanılması durumunda ise batarya enerjisinin 71,04 Wh olduğu hesaplanmıştır. Önceki hesaplamalarda askı durumunda İHA'nın 11,1 V gerilimde 18,3 A akıma ihtiyaç duyacağı hesaplanmıştır. Bu durumda, çalışma gerilimi ile çarpılarak motorun ihtiyaç duyacağı gücün 203,13 W olduğu hesaplanmıştır. Uçuş süresini hesaplamak için bataryanın enerji değeri ihtiyaç duyulan güç değerine bölünebilir.

$$\text{Tüm Uçuş Süresi (dakika)} = \frac{Wh}{W} \cdot 60 = \frac{88,8 Wh}{203,13 W} \cdot 60 = 26,23 dakika$$

$$\text{Sürdürülebilir Uçuş (dakika)} = \frac{Wh}{W} \cdot 48 = \frac{88,8 Wh}{203,13 W} \cdot 48 = 20,98 dakika$$

Yapılan hesaplamalar neticesinde; İHA'nın batarya kapasitesinin %80'ini kullanması durumunda 20,98 dakika, %100'ünü kullanması durumunda 26,23 dakika askıda kalabileceği hesaplanmıştır.

İHA'da Haberleşme ve Yer Kontrol İstasyonu

Çalışma kapsamında geliştirilen İHA'nın iki adet yer birimi mevcuttur: Bunlardan ilki yer kontrol istasyonu, diğeri ise pilottur. Yer kontrol istasyonu, İHA'nın mevcut durumunun izlenmesi ve gerekirse İHA'ya komutlar gönderilmesi için kullanılan yazılım veya yazılımlardır. Yer kontrol istasyonu; uçuş öncesi kontrol, algılayıcı kalibrasyonu ve telemetri gibi ihtiyaçların giderilmesi için kullanılmaktadır.

Çalışma kapsamında, yer kontrol istasyonu olarak Mission Planner adındaki açık kaynak kodlu yazılım kullanılmaktadır. Ayrıca Mission Planner'ın yetersiz kaldığı bazı durumlar için yer istasyonunda çalışmak üzere ikinci bir yazılım daha geliştirilmiştir. Bu bölümde İHA ile yer birimleri arasında haberleşme için kullanılan görev bileşenleri ve yer istasyonunda hangi görev bileşenlerinin ne sebeple kullanıldığı açıklanmıştır.

Yer birimlerinden pilot, İHA ile haberleşmek ve İHA'ya komut göndermek için AT9S RC kumandayı kullanmaktadır. Bu kumanda dokuz kanallı bir kumandadır. Bu kanallardan her birisi İHA üzerindeki alıcı için bir adet PWM sinyali taşımaktadır. Burada her bir PWM sinyali bir sayısal değer içerir. Bu sayısal değerler kumanda üzerinde bulunan kontrollerin ürettiği değerlerdir. Pilot bu değerlerin dördü ile İHA'nın manevralarını kontrol eder. İki kanal uçuş modunu belirlemek ve bu uçuş modları arasında değişiklik yapmak için kullanılır. Kalan üç kanal ise isteğe göre özelleştirilebilir.

PWM sinyalleri İHA üzerinde bulunan alıcı tarafından toplanır. Toplanan PWM sinyalleri, uçuş kontrol kartına iletmek üzere dönüştürülür. Pixhawk 2.4.8 uçuş kontrol kartı PPM (Pulse Position Modulation) veya SBUS tipindeki sinyalleri kabul etmektedir. Bu sinyallerin özellikleri ise PWM sinyalinin aksine üzerlerinde birden fazla sayısal değeri taşıyabilmeleridir. İHA üzerinde alıcı olarak kullanılan R9DS, aldığı PWM sinyallerini SBUS veya PPM sinyaline dönüştürebilmektedir. Bu sayede dönüştürücü donanıma gerek kalmadan ve kablo karmaşası olmaksızın kumanda sinyallerinin İHA'ya iletilmesini sağlar. İHA ile pilot arasındaki bu iletişim 2.4 GHz bandında gerçekleşir ve uçtan uca şifrelidir. Bu iletişim İHA ile yer birimleri arasındaki en tutarlı iletişimidir ve gürültüye karşı dayanıklı haberleşme donanımları kullanılmaktadır.

Pilot, İHA üzerinde en fazla kontrole sahip yer birimidir. İHA'yı otonom veya manuel modlara alabilir. İHA manuel moddayken elle kontrolü üstlenir. Ayrıca görev esnasında İHA'ya otonom iniş komutu gönderebilir. Sonuç olarak görev esnasında İHA'ya verilen komutlar pilot tarafından uygulanır. Pilot ile İHA arasındaki iletişim tek taraflıdır; yani

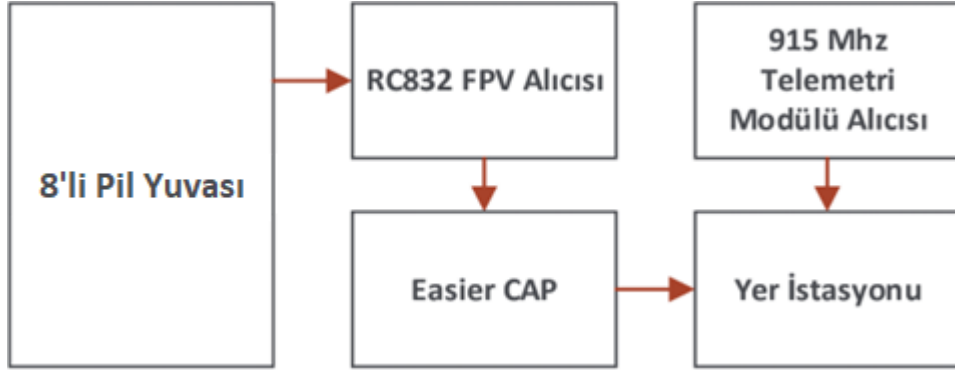
telemetri verileri pilota iletilmez, yalnızca pilot tarafından İHA'ya komut iletilir. İHA tarafından yer birimlerine olan veri akışı, yer kontrol istasyonuna yapılmaktadır.

İHA'dan yer kontrol istasyonuna veri akışı için üç farklı frekansta çalışan donanımlar (Şekil 28) kullanılmaktadır: Bunlardan ilki LoRa, ikincisi telemetri, sonuncusu ise video alıcısıdır. Mission Planner bahsi geçen donanımlardan telemetriyi kullanmaktadır. Telemetri sayesinde doğrudan uçuş kontrol kartından gelen MAVLink (Micro Aerial Vehicle Link) mesajlarını görselleştirmekte ve yine MAVLink mesajları ile uçuş kontrol kartına komut göndermektedir. Geliştirilen yazılım ise LoRa ve video alıcısından gelen verileri görselleştirmektedir. LoRa modülü, yer kontrol istasyonu ile İHA üzerindeki yoldaş bilgisayar arasında veri alışverişini sağlarken; video alıcısı, yoldaş bilgisayarın masaüstü görüntüsünü yer kontrol istasyonuna aktarmaktadır.



Şekil 28. Telemetri (solda), LoRa (ortada) ve RC832 video alıcısı (sağda) fotoğrafları.

Yer kontrol istasyonu ile İHA arasındaki iletişim; LoRa ile 433 MHz, telemetri ile 915 MHz ve RC832 ile 5.8 GHz frekans bantlarında gerçekleşmektedir. Farklı frekanslarda olmalarının sebebi sinyallerdeki karışıklığın önüne geçilmesinin istenmesidir. Yer kontrol istasyonu, RC832 ve LoRa donanımlarından veriyi alabilmek için bazı dönüştürücü donanımlar kullanır. LoRa modülünün yer kontrol istasyonuna bağlantısı için USB-TTL seri dönüştürücü kullanılmıştır. Bununla birlikte video alıcısı olan RC832'nin yakaladığı görüntünün USB portundan yer kontrol istasyonuna aktarılabilmesi için EasyCAP adındaki bir dönüştürücü kullanılmıştır. Bu sayede telemetri, LoRa ve video alıcısının yer kontrol istasyonuna bağlanması için üç USB portu kullanılmıştır. RC832'nin çalışabilmesi için gerekli 12V gerilimin üretilmesi amacıyla pil kutuları ile bir düzenek de (Şekil 29) hazırlanmıştır.



Şekil 29. Kavramsal tasarımı yer kontrol istasyonu ile kullanılacak donanımların ve birbirleri ile bağlantılarının genel diyagramı.

Yer kontrol istasyonunda Mission Planner dışında bir yazılım geliştirilmesi gereksiniminin doğduğundan bahsedilmiştir. Bunun sebebi, Mission Planner tarafından giderilemeyen ihtiyaçlardır. Bunlardan ilki LoRa modülünden alınan verilerin yer kontrol istasyonuna görüntülenememesidir. Özel mesajların yer kontrol istasyonuna telemetri aracılığı ile iletilmesi için uçuş kontrol kartı üzerindeki ArduCopter yazılımında değişiklik yapılması gerekmektedir. Çalışma kapsamında geliştirmelerin yoldaş bilgisayar ile sınırlı kalması istendiğinden ve doğrudan yoldaş bilgisayar ile iletişim sağlanması gerekebileceğinden LoRa modülü ile bir iletişim yolu da sisteme eklenmiştir. LoRa modülü ile gönderilen mesajların operatör tarafından kolay bir şekilde takip edilebilmesi ve anlamlı hale getirilebilmesi için ikincil yazılım geliştirilmiştir.

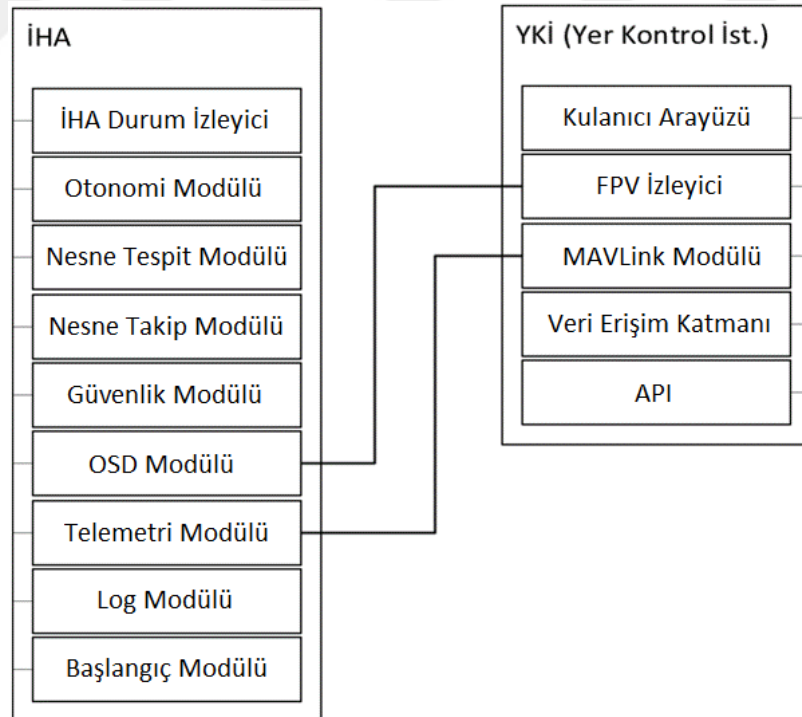
Mission Planner dışında bir yazılım geliştirilmesinin diğer sebebi ise video kayıtlarında yaşanan problemlerdir. İHA'da yer kontrol istasyonuna gönderilen video yayını Mission Planner tarafından görüntülenebilmektedir. Mission Planner ayrıca gönderilen videonun kaydedilmesini de sağlamaktadır. Ancak gelen görüntü üzerine varsayılan olarak bazı uçuş verilerini işlemekte (Şekil 30) ve kaydı bu şekilde değiştirerek yapmaktadır. Söz konusu görüntü üzerine işlenen şekil ve verilerin kaldırılmasında ise kayıt alamamaktadır. Bu hata sebebiyle video aktarımının sağlıklı bir şekilde yakalanması, operatöre gösteriminin yapılması ve hatasız bir şekilde kaydedilmesi için geliştirme yapılmasına ihtiyaç duyulmuştur.



Şekil 30. Mission Planner'ın video üzerine eklediği OSD (On Screen Display) bileşenleri (solda) ve Mission Planner'ın ekran görüntüsü (sağda).

İHA ve Yer İstasyonu İçin Geliştirilen Yazılımların Mimarileri

Çalışmada yapılan yazılım geliştirme faaliyetleri iki bölümde incelenebilir. Bunlardan ilki İHA üzerinde bulunan Raspberry Pi üzerinde çalışacak yazılım, diğeri ise yer kontrol istasyonunda yapılan geliştirmelerdir. Yazılım tasarımında söz konusu iki bölümde de modüler bir mimari üzerinde durulmuştur. Söz konusu modüller (Şekil 31) yazılımın yerine getirmesi gereken görevlerin modüllere ayrılmasıyla belirlenmiştir.



Şekil 31. Yazılım tasarımındaki modülleri gösteren genel diyagram.

Bu bölümde İHA ve yer kontrol istasyonu için geliştirilen yazılımların mimarileri, içerdikleri modüllerin işlevleriyle açıklanmıştır.

İHA Yazılım Mimarisi

İHA yazılımı olarak isimlendirilen ve bu bölümde açıklanmış olan yazılım, İHA'nın gövdesi üzerinde bulunan Raspberry Pi üzerinde tümleşik olarak çalışan bir program komut dosyasıdır. Geliştirilen yazılım, görevlerine göre farklılaştırılmış modüllerden oluşmaktadır. Her bir modül, içerisinde İHA'nın kullanması için gerekli olabilecek temel veya karmaşık işlevsellikleri barındırır.

Modüllerin programlanması için Python programlama dili kullanılmıştır. Modüllerin programlanmasının ardından kullanılmayan işlevsellikler sadeleştirilmiş ve tek bir program komut dosyası oluşturulmuştur. Sadeleştirme neticesinde; asenkron işlemler, eşzamanlılık ve IO işlemleri için komut dosyası içerisindeki işlevler üç ayrı thread ile çalıştırılmıştır.

Aşağıda her bir modülün işlevi açıklanarak mimari hakkında bilgi verilmiştir. Sonuç olarak üretilen program komut dosyasıysa bu tezin ilgili bölümünde açıklanacaktır.

İHA Durum İzleyici Modülü ve İşlevi

İHA üzerinde uçuş kontrol kartı ile yoldaş bilgisayar arasında haberleşmenin sağlanması, otonominin sağlanması için gereklidir. Bu iki işlem birimi arasındaki iletişim, seri iletişim protokolü ve MAVLink mesajları ile sağlanmaktadır. Ayrıca otonominin sağlanması için söz konusu iletişimin çift taraflı gerçekleşmesi gerekmektedir.

İHA durum izleyici modülü, gerçek zamanlı olarak uçuş kontrol kartının telemetri bilgilerini dinlemekle sorumlu olan modüldür. Uçuş kontrol kartını anlık olarak dinlemek ve talep edilmesi durumunda diğer modüllere uçuş kontrol kartından aldığı veriyi servis etmekle görevlidir.

Tasarlanan mimaride ve kullanılan malzemelerde İHA'nın bazı otonomi özellikleri doğrudan pilot tarafından tetiklenebilmektedir. Kullanılan kumanda üzerinde 9 kanal bulunduğu, bunların dördünün uçuş kontrolleri, ikisinin uçuş modları ve üçünün özelleştirilebileceği daha önce açıklanmıştı. Bahsi geçen iki kanal ile kontrol edilen uçuş modlarından bazıları İHA'nın otonomi özelliklerinin kontrolü için de kullanılmıştır. Örneğin İHA'nın otonom moddan pilot kontrolüne geçmesi için GUIDED uçuş modundan çıkması gerekmektedir. Bu durumda, İHA durum izleyici modülü uçuş modunu sürekli dinlemektedir ve bu veriyi otonomi modülü gibi modüller ile paylaşmaktadır. Ayrıca uçuş modu dışındaki bazı özelleştirilmiş kanal değerlerini de benzer sebeplerle dinlemektedir.

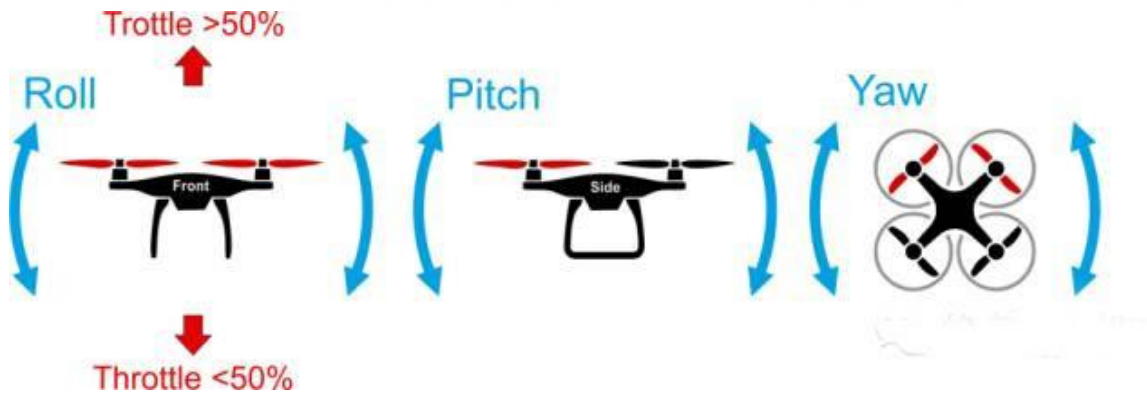
İHA durum izleyici modülü, uçuş kontrol kartından GPS bilgilerini de gerçek zamanlı olarak dinlemektedir. GPS bilgilerinin gerçek zamanlı olarak dinlemesinin sebebi uçuş izni

alınan bölge içerisinde kalınmasını sağlamak ve kalkış yapılan noktanın GPS koordinatlarını kaydetmektir. Uçuş modu, kanal değerleri ve GPS koordinatları dışında İHA durum izleyici GPS saati bilgilerini de dinlemektedir. Bunun sebebi yoldaş bilgisayar üzerinde tutulan logların ve diğer verilerin zaman etiketinin uçuş kontrol kartı ile eşleşmesinin sağlanmasıdır.

Bu değerler dışında da bazı veriler uçuş kontrol kartından talep edilmektedir. İHA durum izleyici modülü, bu verileri uçuş kontrol kartından okuyup servis etmektedir. Her bir veri için ayrı bir fonksiyona sahiptir. Örneğin batarya durumu için gerekli fonksiyon, batarya durumuna ihtiyaç duyan modül tarafından çağrılır. Dolayısı ile İHA durum izleyici modülünün çevresel farkındalık için en önemli modül olduğu söylenebilir.

Otonomi Modülü ve İşlevi

Otonomi modülü İHA'nın otonom manevralarının kontrolünden sorumlu modüldür. Tüm otonom manevra fonksiyonlarını barındırır ve bu fonksiyonlar ihtiyaç duyulması halinde diğer modüller tarafından çağrılır. Döner kanatlı bir İHA, 4 farklı temel manevraya sahiptir. Bu manevralardan üçü basitçe İHA'nın x, y ve z eksenlerinde pozitif ve negatif yönlü hareketini sağlar (Şekil 32). Son temel manevra ise xy düzleminde İHA'nın açısal dönüşünü sağlar.



Şekil 32. Throttle, yaw, pitch ve roll manevralarının temsili görselleri (Anonymous, 2020).

Otonomi modülü söz konusu temel manevraların her birisi için bir fonksiyon içermektedir. Söz konusu temel manevra fonksiyonları hareketin ilgili ekseninde pozitif veya negatif yönde ne kadar süreceğini gösterecek parametreler ile çağrılırlar. Otonomi modülü içerisinde temel manevralar dışında önceden programlanmış manevralar da bulunmaktadır. Bunlardan ilki belirlenmiş bir GPS koordinatına irtifa değiştirmeden uçuşu sağlayan fonksiyondur.

Önceden programlanmış manevralardan diğer ikisi otonom kalkış ve otonom iniş fonksiyonlarıdır. Otonom kalkış hedeflenen irtifayı parametre olarak alır ve hedeflenen irtifaya ulaşınca kadar irtifasını arttırmaya devam eder. İlk irtifa değeri operatör tarafından uçuş öncesi kontrollerde girilir. Otonom kalkışın bir başka varyasyonu olarak söz konusu temel otonom kalkış fonksiyonunu kullanan, ancak kalkıştan sonra uçuş alanının merkezine otonom olarak seyahat eden bir fonksiyon daha programlanmıştır. Bu fonksiyonun doğru bir şekilde çalışabilmesi için bahsi geçen otonom kalkış fonksiyonu ve GPS noktasına uçuş fonksiyonları birleştirilmiştir.

Otonom iniş, daha önce kaydedilmiş olan İHA'nın kalkış yaptığı GPS noktasına irtifa değiştirmeden uçuşunu ve daha sonra irtifa bilgisi minimum değerine ulaşınca kadar irtifasını azaltması ile sağlanmıştır. Bu manevrayı sağlayan RTL (Return To Launch) adında bir uçuş modu ArduCopter içerisinde zaten bulunsun da özellikle sert hava koşullarında kalkış yaptığı noktanın GPS koordinatlarının kontrolünde problem yaşandığından otonomi modülünde alternatif programlanmıştır.

İHA'nın herhangi bir hedefi tespit etmediği durumda çalışması ve çevre taraması yapması için bir otonom manevra programlanmıştır. Bu manevrada, İHA 360 derece boyunca saat yönünde dönerken aynı zamanda irtifasını da değiştirmektedir. Bu sayede kullandığı kameranın FOV (Field of view) kısıtlarından görece kurtulmuş olur. Bu manevra otonom temel manevralardan yüksekliğin değiştirilmesini sağlayan manevrayla açısal dönüşü sağlayan manevranın birlikte kullanılmasıyla geliştirilmiştir. Özetle bu manevrayla hem yatay hem de dikey ekseninde FOV kısıtları olmaksızın hedef tespiti için çevre taraması hedeflenmiştir.

Son olarak otonom takibin gerçekleştirilmesi için yukarıda bahsi geçen manevraların çoğunu kullanan bir manevra da programlanmıştır. Bu manevra nesne tespitinden sorumlu modülün hedef tespiti durumunda tetiklenmektedir. Takip manevrası, parametre olarak diğer fonksiyonlarda olduğu gibi yön ve mesafe bilgileri almamaktadır. Bunun yerine tespit edilen hedefi içerisine alan en küçük dörtgenin (Bounding box) görüntü üzerindeki genişliği, yüksekliği ve koordinatlarını almaktadır. Bahsi geçen dörtgenin, görüntünün merkezine olan uzaklığı ve açısına göre temel manevraya karar vermektedir (Şekil 33). Örneğin tespit edilen hedef görüntünün merkezinden aşağıda ise İHA alçalacak, yukarıda ise İHA yükselecektir.



Şekil 33. İHA takip manevrası çalışmasından. yer kontrol istasyonunda İHA'nın sağa doğru yaptığı manevra (solda) ve İHA'nın manevra esnasında aldığı görüntü (sağda) gösterilmiştir.

Nesne Tespit Modülü ve İşlevi

İHA gövdesindeki kameradan alınan görüntü üzerinde hedefin tespit edilmesinden sorumlu olan modüldür. İHA'da derin öğrenme metotları yalnızca bu modülde kullanılmıştır. Modül girdi olarak görüntü alır ve sonuçta hedefi içerisine alan en küçük dikdörtgenin görüntü üzerindeki koordinatlarını üretir. Modülün bu işlevi yerine getirebilmesi için YOLO (You Only Look Once) kullanılmıştır.

Modül, düşük maliyetli bir donanım üzerinde çalıştığından FPS (Frames per second) değerinin düşük olacağı tahmin edilmiştir. Bu sebepten modülün fonksiyonları sadece iki amaçla çağrılır. Bunlardan ilki hedef tespiti yokken hedefin tespit edilmesi, ikincisi ise takip sırasında takip edilen hedefin doğrulanmasıdır.

Derin öğrenme metotları her görüntü üzerinde çalışmaması durumunda hedefin takip edilmesi için takip algoritmalarına başvurulur. Yani tespit sonrasında modül, nesne takibinden sorumlu olan modülü tetikler. Çalışma aralığının belirlenmesi ve başlangıç ayarları gibi bilgiler sonraki bölümlerde daha detaylı bir şekilde açıklanacaktır.

Nesne Takip Modülü ve İşlevi

İHA üzerindeki kameradan alınan her bir görüntünün derin öğrenme yöntemleriyle işlenmesi, Raspberry Pi gibi düşük maliyetli bir donanımın işlem kapasitesi için ağır bir yük oluşturmaktadır. Her bir görüntü üzerinde YOLO algoritmasının çalışması durumunda FPS değeri oldukça düşmekte ve gerçek zaman algısı tamamen yok olmaktadır. Bu durumla baş edebilmek için daha az işlem maliyeti olan takip algoritmalarının kullanılması uygun görülmüştür.

Takip algoritmalarının eksikliği ise düşük doğruluk ile çalışmalarıdır. Dolayısı ile derin öğrenme metotlarının doğruluğu ile takip algoritmalarının performansının birlikte çalışabilir hale getirilmesi ihtiyacı doğmuştur. Söz konusu metotlar iki farklı modül içerisinde kullanılmıştır. Bu modüllerden ilki daha önce bahsi geçen nesne tespit modülüydü ve derin öğrenme metodunu kapsamaktaydı. Diğer modül ise nesne takip modülüdür. Nesne takip modülü ise CSRT, KCF, MOSSE gibi takip algoritmalarıyla yapılan nesne takibi işlevselliklerini kapsamaktadır.

Nesne tespit modülü, herhangi bir hedefi tespit etmesi durumunda nesne takip modülünü tetikler. Tetiklenen fonksiyon parametre olarak hedefin kapsayan en küçük dikdörtgenin koordinatlarını alır. Operatör tarafından uçuş öncesinde belirlenmiş olan takip algoritmasıyla belirlenmiş görüntü adedince görüntü üzerinde hedefi takip eder. Belirlenmiş görüntü adedine ulaştığında ise takibin doğruluğunu sağlamak için nesne tespit modülünü tetikler. Bu sayede takip esnasındaki doğruluk probleminin yüksek FPS değerleri ile giderilmesi amaçlanır.

Nesne takip modülü ayrıca otonomi modülünü de tetikleyerek görüntü üzerindeki takibin İHA manevraları için kullanılmasını da sağlar. Ayrıca nesne takip modülü, OSD modülünün hedefi kapsayan dikdörtgeni görüntüye işlemesi ve otonom takip manevralarının sürdürülmesi için takip durumunu bildiren global değişkenin değerini düzenler. Takibin kaybedilmesi durumunda ise global değişkenlerle nesne tespit modülünü ve diğer ilişkili modülleri bilgilendirir.

Güvenlik Modülü ve İşlevi

Güvenlik modülü, istenmeyen durumlar karşısında bazı eylemlerin tetiklenmesi için tasarlanmıştır. ArduCopter, bazı güvenlik amaçlı işlevsellikleri zaten sunmaktadır. Ancak ArduCopter'in sunduğu failsafe sistemi, riskli bulunan bazı manevraları içerdiğinden geofence gibi bazı güvenlik önlemleri, geliştirilen güvenlik modülü içerisinde yeniden programlanmıştır.

Güvenlik modülü, İHA durum izleyici modülü tarafından servis edilen GPS koordinatları ve uçuş modu verilerine oldukça bağımlıdır. Güvenlik modülünde uçuş yapılan hava sahasının dikdörtgen şeklinde olacağı varsayılmıştır. Bu sebepten yazılım içerisinde uçuş alanının sınırlarının belirlenmesi için dört GPS koordinatının bilinmesi gerekmektedir. Bu GPS koordinatları dörtgen şeklindeki uçuş alanının köşelerini belirtir. Bu veriler, uçuş öncesinde operatör tarafından girilmektedir. Ayrıca kalkış esnasında operatörün girdiği veriler ile uçuş alanının merkez noktası hesaplanarak kaydedilir.

Yeniden programlanan geofence fonksiyonuyla İHA uçuş alanının sınırına yaklaştığında, güvenlik modülü ilişkili bir fonksiyonu tetikleyerek İHA'yı uçuş alanının merkezine irtifa değişmeden yeniden yönlendirir. Bu yönlendirmenin ardından otonomi modülünün bir işlevi olan tarama manevrasına başlanır. Ayrıca otonomi modülünde acil manevralar olarak isimlendirilen bazı manevraları da güvenlik modülü tetikleyebilmektedir.

OSD Modülü ve İşlevi

OSD (On Screen Display) modülü, elektronik bir bileşen olan OSD bileşeninin görevini yazılımsal olarak üstlenmektedir. Geliştirilen İHA'da yer istasyonuna video aktarımı yoldaş bilgisayarın masaüstünün gerçek zamanlı yayınlanması ile başlar. OSD modülü ise masaüstünde istenen görüntünün elde edilmesinden sorumludur.

OSD modülü Raspberry Pi kamera modülünden alınan gerçek zamanlı görüntü üzerine İHA durum izleyici modülünden edindiği uçuş modu bilgisini yazılı olarak işler. Bu sayede telemetri modülü dışında temel bazı bilgiler de yer birimlerine görüntü vasıtasıyla iletilmiş olur. Bunun dışında OSD modülü hedefin içerisinde tutulması istenen ve hedefi kapsayan dörtgenleri görüntü üzerine işler. Geliştirme esnasında FPS skorları gibi test verileri de ihtiyaca göre OSD modülü tarafından görüntüye işlenebilir.

Telemetri Modülü ve İşlevi

İHA'nın yer kontrol istasyonuna gerçek zamanlı veri göndermesi için uçuş kontrol kartına bağlı 915 MHz bandında çalışan bir telemetri modülü kullanılmaktadır. Telemetri verilerinin yer kontrol istasyonundan dinlenmesi için ise Mission Planner adında bir açık kaynak kodlu yer istasyonu kullanılmaktadır. İHA ve yer istasyonu arasındaki bu veriler MAVLink protokolüne uygun içeriklerden oluşmaktadır. Ancak çalışma kapsamında bazı özel mesajların yer kontrol istasyonuna gönderilmesi ihtiyacı doğmuştur. Bu ihtiyacı karşılamak için ilk akla gelen çözüm uçuş kontrol kartı üzerinde bulunan ArduCopter isimli aygıt yazılımı üzerinde özelleştirilme yapılması olmuştur. Burada söz konusu MAVLink mesajlarının içeriklerinin değiştirilmesi hedeflenmiştir. Ancak daha sonra yazılım geliştirme faaliyetlerinin yoldaş bilgisayar ile sınırlandırılmasına karar verilmiştir. Özel mesajların yer kontrol istasyonuna gönderilmesi ihtiyacını karşılamak ve yalnızca yoldaş bilgisayar üzerinde geliştirme yapmak amacıyla yoldaş bilgisayara, 433 MHz bandında çalışan bir LoRa modülü bağlanmıştır.

Telemetri modülü, hedef tespiti ve takibine ilişkin özel mesajları LoRa modülünü kullanarak yer kontrol istasyonuna göndermekten sorumlu olan modüldür. LoRa modülüne

veriyi göndermek için seri iletişim protokolü kullanılmaktadır. Dolayısı ile Telemetry modülü seri protokol ile yer kontrol istasyonuna gönderilmek istenen mesajları yazan ve bu işlemi asenkron olarak gerçekleştiren modüldür.

Log Modülü ve İşlevi

İHA üzerindeki yoldaş bilgisayarda yerel log dosyaları oluşturmaktan ve bu dosyaların içeriğini belirlemekten sorumlu modüldür. Log dosyaları çalışma kapsamında bazı testlerin yapılması esnasında kullanılmıştır. Farklı mesaj tiplerini, mesaj içeriklerini ve İHA durum izleyici tarafından sağlanan GPS saatini kullanır. Info, alert, danger gibi etiketler ile log dosyasındaki mesajları etiketler. Bu modül İHA üzerinde IO işlemleri yapan üç modülden birisidir. Modül, geliştirme sırasında kullanılsa da çalışma çıktısı olan sadeleştirilmiş program komut dosyası içerisinde kullanılmamıştır.

Başlangıç Modülü ve İşlevi

Başlangıç modülü, diğer modüllerden önce çalışan ve diğer modüllerin çalışması için gerekli şartların sağlandığını doğrulamak için global değişkenleri kontrol eden modüldür. Ayrıca bu modül, operatör tarafından uçuş öncesinde yapılan ayarlamaları içeren konfigürasyon dosyasını okumak ve global değişkenlere ilk değerlerini atamaktan sorumludur. Başlangıç modülü gerekli kontrolleri yaptıktan sonra İHA'yı arm moduna geçirerek uçuş izni verebilen yegane modüldür.

Yer Kontrol İstasyonu Yazılım Mimarisi

Çalışma kapsamında yer kontrol istasyonunda kullanılan Mission Planner, algılayıcıların kalibre edilmesinden uçuş verilerinin anlık olarak takip edilmesine kadar pek çok ihtiyacı gidermiştir. Ancak Mission Planner tarafından giderilememiş bazı ihtiyaçlar sebebiyle yer kontrol istasyonunda ikinci bir yazılımın geliştirilmesi ihtiyacı doğmuştur.

Bu ihtiyaçlardan ilki Mission Planner'ın varsayılan OSD çizimleri devre dışı bırakıldığında, İHA'dan alınan görüntüyü kaydedememesidir. Bu problem Mission Planner içerisinde bulunan bir hata olduğundan ve çalışmanın yapıldığı tarihlerde bu hata henüz giderilmediğinden çözümü için iki seçenek olduğu fark edilmiştir. Bunlardan ilki İHA üzerinde ve yer istasyonunda bulunan video alıcı verici donanımlarının değiştirilmesidir. İkincisi ise yer istasyonunda bu gereksinimi giderecek ikincil bir yazılımın Mission Planner'la birlikte çalıştırılmasıdır. Aviyonik konfigürasyonu düşük maliyette tutmak için yazılım geliştirme ile problemin çözümüne karar verilmiştir.

Mission Planner ile giderilemeyen diğer bir ihtiyaç ise özel MAVLink mesajlarının yer istasyonuna gönderilmesi ihtiyacıdır. Uçuş kontrol kartı tarafından kullanılan ArduCopter, içerisinde yaygın olarak kullanılan pek çok MAVLink mesajını barındırmaktadır. Ancak hedef tespiti ve takibi durumlarında yer istasyonunu bilgilendirecek bir MAVLink mesajı bulunmamaktadır. Bunun için ArduCopter üzerinde değişiklik yapılarak yeni MAVLink mesajlarının sisteme eklenmesi gerekmektedir. ArduCopter’de yapılacak değişiklikler için kesin bir çözüm bulunamadığından ve geliştirme faaliyetlerinin yoldaş bilgisayarla sınırlandırılması kararlaştırıldığından ArduCopter üzerinde değişiklik yapılmamıştır. Bunun yerine İHA üzerinde doğrudan uçuş kontrol kartı ile haberleşebilmesi için bir LoRa modülü dahil edilmiş ve yer istasyonunda LoRa mesajlarını okuyan bir yazılım geliştirilmesine karar verilmiştir.

Yukarıda bahsi geçen FPV (First person view) ve LoRa problemleri sebebiyle geliştirilmesine karar verilen programcıkların, doğabilecek diğer ihtiyaçlar da dikkate alınarak, bir yer istasyonu projesi kapsamında geliştirilmesine karar verilmiştir. Bu sayede ileride doğabilecek diğer gereksinimler için çok fazla sayıda programcık çalıştırmak yerine tek bir yer istasyonu projesi çalıştırılarak operatörün iş yükünün azaltılması amaçlanmıştır.

Sonuç olarak yer istasyonunda doğan gereksinimlerin giderilmesi için geliştirilen programcıklar tek çatı altında toplanmış ve bir yer istasyonu projesine başlanmıştır. Kullanıcı deneyimi açısından Mission Planner arayüzüne benzer bir arayüz tasarlanmış ancak ileride doğabilecek ihtiyaçlar düşünülerek çoğu alan boş bırakılmıştır.

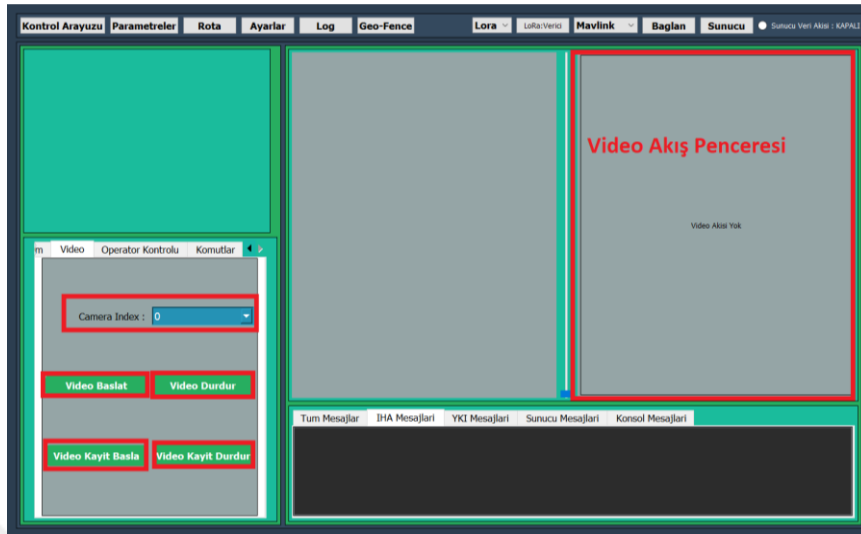
Geliştirilen yer kontrol istasyonunda yaygın olarak bilinen MVC (Model View Controller) şablonundan esinlenilmiş ve gerekli modüller bu şablon içerisine entegre edilmiştir. Sonuçta ortaya çıkan yer kontrol istasyonunda altı adet modül oluşturulmuştur. Bu modüller ve modüllere ilişkin bilgiler aşağıda ayrı başlıklar altında açıklanmıştır.

Kullanıcı Arayüzü ve İşlevi

Kullanıcı Arayüzü, arayüz için gerekli geliştirmeleri barındıran modüldür. PyQt5 kütüphanesi ile geliştirilen arayüzde kullanıcı deneyimi (UX) düşünülerek Mission Planner arayüzüne benzer bir tasarım ortaya çıkarılmıştır. Tasarımın büyük bir bölümü ihtiyaç halinde kullanılmak üzere boş bırakılmış yalnızca daha önce bahsi geçen ihtiyaçları karşılayacak bölümler için geliştirme yapılmıştır (Şekil 34).

Arayüzde FPV izleyici isimli modülden alınan görüntülerin izlenebileceği bir çerçeve ve ekranın alt kısmında LoRa modülünden gelen mesajları canlı olarak izleyebilmek için gerekli bir konsol tasarlanmıştır. Ayrıca bu modüllerin çalışmaya başlaması veya

durdurulması için gerekli kontroller ve bazı gerekli ayarlar için de tasarım ve programlama yapılmıştır.



Şekil 34. Geliştirilen yer istasyonu arayüzünün (görsel üzerinde bazı açıklamalarla) gösterimi.

FPV İzleyici Modülü ve İşlevi

FPV izleyici modülü, yer kontrol istasyonunda gerçek zamanlı video yayınından alınan görüntüyü kullanıcı arayüzüne iletmek ve ihtiyaç halinde kaydetmekten sorumlu modüldür. Kullanılan FPV donanımı (RC832 ve TS832) ve dönüştürücü (EasyCAP) sayesinde İHA'dan gelen görüntülerin içeri aktarılması bir web kamerasının açılması işlemi ile aynı metotları kullanır. Yani donanımı yer istasyonuna USB portundan bağlayıp OpenCV kütüphanesiyle kamera açıldığında İHA'dan gelen görüntülere erişilmiş olur. Alınan görüntüler kullanıcı arayüzü modülüne gönderilir ve kaydediliyorsa yine OpenCV kütüphanesi aracılığıyla kayıt işlemi gerçekleştirilir.

MAVLink Modülü ve İşlevi

MAVLink modülü, İHA'dan yer kontrol istasyonuna gönderilen mesajların LoRa ile yakalanmasından ve bu mesajları arayüze aktarmaktan sorumlu olan modüldür. LoRa ile haberleşme half duplex olarak gerçekleştirilir. Bu durumda MAVLink modülünde mesaj alma ve mesaj gönderme için iki işlevsellik bulunmaktadır. Alıcı pozisyonda olan ve yer istasyonuna bağlı olan LoRa donanımı bir seri dönüştürücü vasıtasıyla aldığı mesajları USB port üzerinden yer istasyonuna iletmektedir. Seri haberleşme ile gelen veri paketlerinin anlamlı mesajlara dönüştürülebilmesi için serial kütüphanesi kullanılmıştır. Ayrıca yakalanan veri paketlerini sorunsuz şekilde yazılımın içine aktarılması için LoRa modülüne ait bir

sürücü kurulmuştur. Gelen veri paketleri dinlenerek anlamlı mesajlara dönüştürölüp kullanıcı arayüzü modölüne iletilmiştir.

Kontrolcü / Veri Erişim Katmanı ve İşlevi

Kontrolcü; FPV izleyici ve MAVLink modülleri de dahil olmak üzere yer istasyonunda bulunan genel işlevsel özellikleri barındıran bir üst modüldür. Ayrıca kontrolcü katmanı yazılım geliştirme sürecinde test için kullanılan geçici verileri barındıran ve bunları test esnasında kullanan modüldür. Doğru çalışma için başlangıç değışkenlerinin tanımlandığı “config.ini” dosyasına bağımlıdır. Söz konusu dosyadan okuma yaparak yer istasyonunun hatasız bir şekilde ve doğru başlangıç değeriyle çalışmaya başlamasını sağlar. Barındırdığı işlevselliklerde kullandığı veri modellerini model katmanından alan kontrolcü modülü, giden ve gelen verileri bu modellere atayarak veri erişim katmanı (Data Access Layer) gibi görev yapmaktadır.

API Modülü ve İşlevi

API modülü İHA’dan gelen ve MAVlink modülü tarafından yakalanan mesajların uzak bir sunucuya gönderilmesinin istenmesi durumunda kullanılmak üzere geliştirilmiştir. MAVLink modülü gelen mesajları yakalayıp kullanıcı arayüzü modölüne göndermektedir ancak gelen mesajları kaydetmemektedir. Bunun yerine ihtiyaç duyulması halinde gelen mesajları uzak bir sunucuya göndermesi için API (Application Programming Interface) modülü geliştirilmiştir. Bu sayede kayıt veya istatistik işlemleri istenmesi halinde bir uzak sunucuda gerçekleştirilebilir hale getirilmiştir. Gönderilen verilerin modeli yer istasyonundaki model katmanında olduğundan uzaktaki sunucu gelecek olan verinin modelini ve barındırdığı bilgileri uygun şekilde değerlendirebilecektir. API modülü HTTP (Hyper Text Transfer Protocol) taleplerini iletebilmek için url bilgisini içerisinde barındırmaktadır. Özetle API modülü, HTTP taleplerini ve JSON formatındaki verileri uzak bir sunucuya göndermekten sorumlu olan görece basit bir modüldür.

Model Katmanı

Model katmanı transfer edilen veya yer istasyonu içerisinde kullanılan verilerin modellerini barındıran bir sözlük olarak tasarlanmıştır. Alışlagelmiş MVC şablonunda olduğu gibi veritabanı mimarisini yansıtmak yerine gönderilen ve alınan verilerin doğru ve tutarlı olmasını sağlamak, veri modellerinin alt işlevlerini barındırmak ve gerekmesi halinde veri modelleri arasında dönüşüm için gerekli işlevleri sağlamak amacıyla geliştirilmiştir. Model katmanı yazılım mimarisinde bir modül olarak değil bir sözlük gibi değerlendirilmiştir.

Nesne Tespiti ve Nesne Takibi

İHA'nın en önemli görev isteri hedef tespiti ve takibidir. Düşük maliyetli donanım ile bu görevin gerçek zamanlı olarak gerçekleştirilmesinde bazı zorluklar ortaya çıkmaktadır. Pek çok çalışmada olduğu gibi nesnenin tespit edilebilmesi için bir evrişimli sinir ağı metodu olan YOLO kullanılmıştır. Düşük maliyetli bir donanım olan Raspberry Pi üzerinde YOLO ile nesne tespiti yapıldığında performans problemleri ortaya çıkmakta ve FPS çok düşük olduğundan gerçek zaman algısı bozulmaktadır. Bu sebeple YOLO kullanılarak kameradan gelen her bir görüntü (frame) üzerinde nesne tespiti yapılmaması gerektiğine karar verilmiştir.

YOLO'nun tüm görüntülerde nesneyi tespit etmemesi, tespit sonrası takibi imkansız kılmaktadır. Bu sebeple YOLO'nun kullanılmadığı görüntülerde takip algoritmalarının kullanılmasına karar verilmiştir. Takip algoritmaları performans açısından YOLO'dan çok daha iyi sonuçlar ortaya koysa da doğruluk problemleri vardır. Yani tespit edilen nesnenin uzun süre takip algoritmaları ile izlenmesi durumunda hatalı takip yapılması olağandır. Dolayısı ile takip algoritmaları doğrulamaya ihtiyaç duymaktadır.

Performans ve doğruluk problemlerinin çözülebilmesi için YOLO ve takip algoritmalarının birbirlerinin eksiklerini tamamlayacak şekilde sıralı çalıştırılmasına karar verilmiştir. Önerilen çözüme göre bir nesne tespit edilinceye kadar YOLO çalışacak, nesne tespit edildiğinde ise takip algoritması tetiklenecektir. Takip algoritması bir süre nesneyi görüntü üzerinde takip ettikten sonra takibin doğruluğunu sağlamak için YOLO algoritmasını çalıştıracaktır. Bu çözüme ilişkin çalışma, aşağıda başlıklar altında açıklanmıştır.

Nesne Tespiti İçin Derin Öğrenme

YOLO (You Only Look Once), GoogLeNet'ten esinlenilerek 2016 yılında nesne tespiti için ileri sürülmüş bir CNN metodudur. YOLO görseli tek parça halinde sinir ağında işlediğinden Faster R-CNN gibi metotlara göre daha yüksek performans üretmeyi başarmıştır. YOLO'nun 2016 yılındaki ilk versiyonundan sonra pek çok ileri versiyonu çıkmıştır. Ayrıca çıkarılmış her YOLO versiyonunun farklılaştırılmış varyantları da ortaya çıkmıştır. YOLO versiyonları ve varyantları ile performans ve doğruluk arttırılmaya çalışılmıştır. Örneğin YOLO v4 bir versiyon olarak ortaya çıkarken, YOLO v4 Tiny daha düşük işlem maliyeti ile nesne tespiti yapabilmeyi sağlayan bir varyant olarak ortaya çıkmıştır.

İşbu çalışmada, YOLO v4 Tiny ve Darknet kullanılarak nesne tespiti yapılmıştır. Bunun sebebi YOLO v4 Tiny'nin düşük maliyetli sistemler için özelleştirilmiş ve kolay uygulanabilir olmasıdır. YOLO v4 Tiny ile YOLO v4 arasındaki fark evrişim ve YOLO katmanlarının sayılarının düşürülmüş olmasıdır. Bu sayede YOLO v4 Tiny, YOLO v4'e göre

daha yüksek performans ve daha az doğruluk göstermektedir. Diğer taraftan YOLO v4 Tiny, OpenCV kütüphanesi ile yüksek entegrasyona sahiptir. Yani tespit işlemi için geliştirme faaliyetleri bazı YOLO versiyonlarına göre çok daha kolaydır.

YOLO'nun İHA yazılımında kullanılabilmesi için OpenCV kütüphanesi kullanılmıştır. İHA yazılımı önceden eğitilmiş bir YOLO modelini kullanarak nesneyi tespit eder. Dolayısı ile training işlemi için gerekli kodları içerisinde barındırmaz. Bunun yerine YOLO versiyonuna ait bir config dosyası ve önceden eğitilmiş modelin ağırlık değerlerinin bulunduğu weights dosyasının sağlanması yeterlidir.

İHA yazılımı için training işlemi, Google Colab aracılığı ile yapılmıştır. Training neticesinde üretilen model ağırlıklarını içeren weights dosyası ise İHA yazılımı içerisine konmuştur. Tespit edilecek nesne yalnızca döner kanatlı İHA olacağından tespit için gerekli konfigürasyon ayarlamalarının pek çoğunda varsayılan değerler kullanılmıştır. Training esnasında döner kanatlı İHA'ların YOLO formatında etiketlenmiş fotoğraflarından oluşan bir veri kümesine ihtiyaç duyulmuştur. Bu ihtiyacı gidermek için yapılan araştırma sonraki başlıkta açıklanmıştır.

Veri Kümeleri

Nesne tespiti için derin öğrenme metodunda kullanılmak üzere tespit edilecek nesnenin etiketlenmiş görsellerinden oluşan bir veri kümesine ihtiyaç duyulmuştur. Daha önce çalışma kapsamında tespit edilecek olan nesnenin diğer döner kanatlı İHA'lar olacağı açıklanmıştı. Döner kanatlı İHA fotoğraflarından oluşan bir veri kümesinin bulunabilmesi için yapılan çalışma neticesinde dört adet kullanılabilir veri kümesi bulunmuştur. Söz konusu veri kümeleri bu bölümde açıklanmıştır.

Bahsi geçen veri kümelerinden ilki Det-Fly isimindeki veri kümesidir. Det-Fly veri kümesi, farklı arka plan, görüş açısı ve ışıklarda yüksek çözünürlüklü İHA fotoğraflarından oluşmaktadır. Her bir fotoğrafta uçar haldeki bir adet döner kanatlı İHA etiketlenmiştir. Veri kümesi toplam 13261 fotoğraftan oluşmaktadır. (Zheng *et al.* 2021)

Kullanılabilir bulunan diğer bir veri kümesi ise MIDGARD adındaki veri kümesidir. İçerisinde farklı ortamlarda çekilmiş döner kanatlı İHA fotoğrafları bulunmaktadır. Uçuşun yapıldığı mekana (iç, dış, tepe, şehir içi vs.), ışık durumuna ve kamera görüş alanına (FOV) göre bölümlere ayrılmıştır. Fotoğraflar görece düşük çözünürlüklüdür ve fotoğraflar video görüntülerinden oluşmaktadır. (Walter *et al.* 2020)

Kullanılabilir bulunan diğer veri kümesi amatör bir veri kümesidir. Söz konusu veri kümesi internetten bulunmuş döner kanatlı İHA görsellerinden oluşmaktadır. Görseller İHA fotoğraflarıyla birlikte bazı İHA çizimlerini de barındırmaktadır. Veri kümesi, kullanılmadan önce ön işlem gerektirmektedir. (Aksoy vd. 2019)

Kullanılabilir bulunan son veri kümesi ise yine internetten bulunmuş görsellerden oluşmaktadır. Aksoy vd. (2019) tarafından hazırlanan veri kümesiyle benzer niteliktedir. Ön işlem olmaksızın doğrudan kullanılamaz. (Özel 2019)

Aksoy vd. (2019) ve Özel (2019) tarafından sunulan veri kümeleri bu çalışma kapsamında amatör veri kümeleri olarak nitelendirilmiştir. Bu veri kümeleri internetten bulunan görsellerden oluştuklarından görsel çözünürlükleri sabitlenmemiştir. Ayrıca tüm veri etiketlemelerinin tamamı doğru bir şekilde yapılmamıştır. İlgili ön işlemler yapıldıktan sonra bu veri kümeleri bu çalışma kapsamında kullanılabilir olarak sınıflandırılmıştır.

Zheng *et al.* (2021) ve Walter *et al.* (2020) tarafından sunulan veri kümeleri ise etiket verilerinin YOLO formatına dönüştürülmesi neticesinde kullanılabilmektedir. Bu veri kümelerinden MIDGARD, videolardan alınmış görüntüler şeklinde olduğundan ve kamera görüş açısında değişiklik yaptığından dolayı tercih edilmemiştir. Sonuç olarak veri kümelerinden 13261 fotoğraftan oluşan Det-Fly'nin kullanılmasına karar verilmiştir. Ayrıca training süresini kısaltarak test süreçlerine daha kısa bir sürede geçiş yapmak ve testlerin doğru metrikleri ürettiğinden emin olmak amacıyla bir süreliğine amatör veri kümelerindeki görüntülerden faydalanılmıştır.

Nesne tespiti ve model eğitimi dışında, takip algoritmalarının test edilmesi için 1841 ardışık video görüntüsünden veri kümesi oluşturulmuştur. Söz konusu veri kümesinin elde edildiği videoda bir döner kanatlı İHA ekran dışına çıkmayacak şekilde takip edilmektedir. Veri kümesinin oluşturulması için videonun her bir görüntüsü etiketlenerek teste hazır hale getirilmiştir. Takip algoritmaları genel itibarı ile düşük doğruluk ve yüksek performans gösteren metotlar olduğundan 1841 görüntüden oluşan bir veri kümesi yeterli test şartlarını sağlamıştır. Yani son görüntüye ulaşmadan takip algoritmalarının davranışlarının analiz edilebilmesine yönelik ihtiyaç duyulan sonuçlar elde edilebilmiştir. Ayrıca nesne tespitinde performansın ölçülebilmesi için 713 görüntüden oluşan bir videonun kullanıldığı da belirtilmelidir.

Nesne Takibi İçin Takip Algoritmaları

Takip algoritmaları, yapı itibarıyla gerçek zamanlı olarak videolar üzerinde çalışmak üzere ileri sürülen çözümler olduğundan derin öğrenme metotlarına göre daha yüksek

performans sergilemektedirler. İşbu çalışmada İHA yazılımı içerisine altı adet takip algoritması eklenmiştir. Operatör uçuş öncesinde hangi takip algoritmasının kullanılacağını belirleyebilir. Operatör tarafından belirlenmemesi durumunda varsayılan bir takip algoritması kurulur. İHA yazılımı içerisinde kullanılan takip algoritmaları: KCF, BOOSTING, MIL, MEDIANFLOW, MOSSE ve CSRT olmak üzere altı adettir. Varsayılan algoritma ise MIL'dir.

Takip algoritmalarının her biri test edilmiş ve özellikle doğruluk değerleri incelenmiştir. Takip algoritmalarının doğruluklarının anlaşılabilmesi için her bir takip algoritması, uçuş videosu üzerinde çalıştırılarak IoU (Intersection Over Union) değerleri hesaplanmıştır. Ancak alınan sonuçlar algoritmaların davranışlarını tam olarak niteleyememiştir. Bunun sebebi algoritmaların davranışlarındaki farklılıklardır. Örneğin BOOSTING algoritması yüksek FPS üretmektedir ancak İHA uzaklaştığında bounding box küçülmemekte veya yaklaştığında büyümektedir. Dolayısı ile BOOSTING algoritmasının kullanılması durumunda YOLO'ya daha sık başvurulması ihtiyacı doğacaktır. CSRT algoritması ise BOOSTING'in bu zafiyetine sahip olmasa da performansı daha düşüktür.

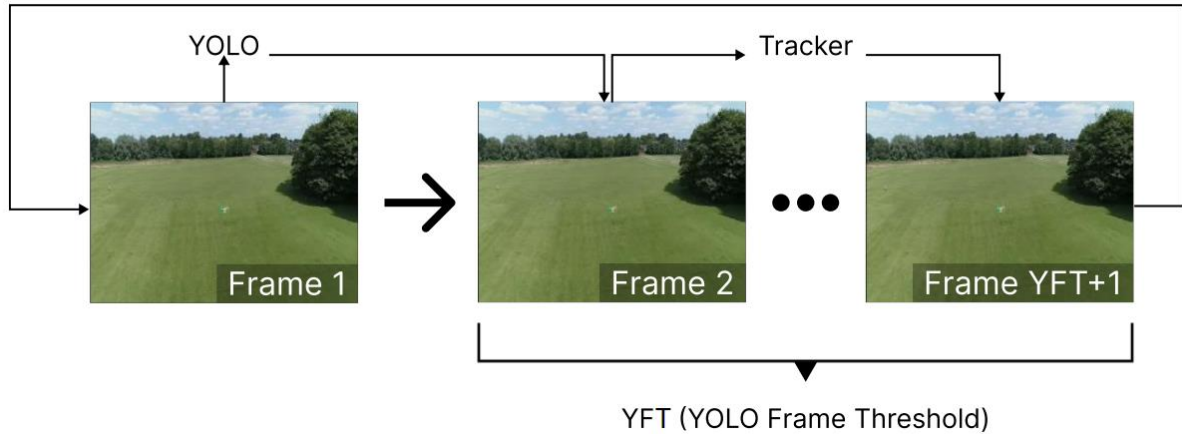
Sonuç olarak tek bir takip algoritmasının kullanılması yerine, yazılım içerisine incelenen takip algoritmalarının tamamı eklenmiş ve karar operatöre bırakılmıştır. Operatörün takip algoritmasını belirlemesi için uzaktan bağlantıyla İHA üzerinde bulunan ini uzantılı bir dosyayı düzenlemesi gerekmektedir.

Nesne Tespiti ve Takibinin Birlikte Çalıştırılması

İşbu çalışmada; nesne tespiti ve takibi modülleri, birbirlerini periyodik olarak tetikleyen modüllerdir. Birlikte nasıl çalıştıklarını anlamak için üzerinde çalışılan kullanım senaryosunu açıklamak gerekir. Varsayılan kullanıma göre İHA, otonom kalkış yaptıktan sonra nesne tespiti ve takibi için temel ayarlamaları yapar ve nesne tespiti modülünü başlatır. Kalkış tamamlandığında İHA, uçuş yapacağı hava sahasının merkezine doğru otonom uçuşa başlar ve merkezde tarama manevrasını başlatır. Bu esnada nesne tespiti modülü herhangi bir hedef tespit edilinceye kadar her bir görüntü üzerinde YOLO algoritmasını çalıştırır. Nesne tespiti modülü bir hedefi tespit ettiğinde ise nesne takibi modülünü tetikleyerek tespit edilen hedefin görüntü üzerinde önceden belirlenmiş takip algoritması ile izlenmesini sağlar. Nesne takibi modülü önceden belirlenmiş görüntü adedince nesneyi izlemeye devam eder. Söz konusu görüntü adedine erişince yeniden nesne tespiti modülünü tetikler ve sıradaki görüntü üzerinde nesne tespiti modülünü çalıştırır. Nesne tespiti modülü gelen görüntü üzerinde hedefi yeniden tespit ederse takip algoritmasını yeniden tetikler ve süreç kendini tekrar eder. Eğer bir

hedef tespit edilmezse YOLO, yeniden hedef tespit edilinceye kadar gelen görüntüler üzerinde çalışmaya devam eder.

Burada iki modülün birbirlerini tetiklemeleri global değişkenler vasıtasıyla gerçekleşir. Bu global değişkenlerin ilk değerleri, Başlangıç modülü tarafından config dosyasından okunarak atanır. Takip algoritmasının kaç görüntü adedince çalışacağını belirten değişken, YOLO Frame Threshold (YFT) olarak isimlendirilmiş değişkendir. Bu değişkenin varsayılan değeri 15 olarak belirlenmiştir. Ancak bu değer operatör tarafından uçuş öncesinde değiştirilebilmektedir. Yani varsayılan ayarlarla tespitten sonra 15 görüntü boyunca takip algoritması tespit edilen nesneyi izlemekte, 16'ncı görüntüde ise tekrar nesne tespiti modülü çalıştırılmaktadır. Tespit ve takip durumlarını belirten flag değişkenleri ve tespit edilen nesneyi kapsayan en küçük dikdörtgenin görüntü üzerindeki koordinatlarını belirten değişkenler de iki modülün doğru bir şekilde çalışmasını sağlamak için kullanılmıştır. Şekil 35'te YFT değeri ile tespit ve takip metotlarının çalışma şeklini gösteren bir temsili görsel sunulmuştur.



Şekil 35. Tespit ve takip işlemlerinin birlikte çalıştırılması ve YFT değeri.

Doğruluk ve Performans Metrikleri

Nesne tespiti ve nesne takibinin değerlendirilmesinde kullanılacak olan metriklerin anlaşılması ilerleyen bölümlerde yapılan testler ve bu testlerin sonuçlarını anlamlandırmak için oldukça önemlidir. Bu noktada değerlendirmede kullanılan metriklerin genel olarak performans ve doğruluk metrikleri olarak ikiye bölünebileceği söylenebilir. Performans metrikleri metodun hızını FPS cinsinden göstermektedir. Doğruluk metrikleri ise yapılan tespitin ve takibin doğru ve tutarlı olmasını ifade etmektedir. Bu bölümde nesne tespiti ve takibinde kullanılan metotların test edilmesi ve test sonuçlarının anlamlandırılması için gerekli metrikler açıklanmıştır.

Nesne tespiti için kullanılan YOLO v4 Tiny metodu, pek çok metrik ile doğruluğunu ölçümleyebilmektedir. Bu metriklerden görece daha basit olanlar TP (True Positive), FP (False Positive), FN (False Negative), TN (True Negative) metrikleridir. Test esnasında eğitilmiş olan model, eğitildiği veri kümesinden oluşturulmuş daha küçük bir veri kümesi ile test edilir. Bu test esnasında görüntülerde ilişkili nesneyi tespit etmesi istenir ve doğru bir şekilde tespit işleminin gerçekleştirilip gerçekleştirilmediği kontrol edilir. Bu kapsamda TP, görüntüler üzerinde bulunan nesnelerin eğitilen model ile kaç kez doğru bir şekilde tespit edildiğini; FP, üzerinde tespit edilecek nesne olmayan görüntülerde kaç kez nesne tespit edildiğini gösterir. Benzer şekilde FN, görüntü üzerinde tespit edilmesi beklenen bir nesne varken modelin kaç kez bu nesneyi tespit edemediğini belirtir. Son olarak TN ise görüntü üzerinde nesne yokken modelin nesne tespit etmemesi durumunun kaç kez gerçekleştiğini gösterir. Anlaşıldığı üzere TP ve TN metrikleri modelin yaptığı doğru tahminleri, FN ve FP metrikleri modelin yaptığı hatalı tahminleri belirtir. Sayış oldukları söylenebilecek bu 4 metrik daha kompleks metriklerin hesaplanmasında sıkça kullanılmaktadır.

Precision ve recall metrikleri, nesne tespiti için modelin doğruluğunu anlamak için kullanılan ve bahsi geçen 4 metrikten türetilmektedir. Precision, görüntü üzerinde yapılan tespitlerin ne kadarının doğru tespit olduğunu belirten bir metriktir. Tespit yapılmayan görüntülerin doğru veya yanlış olmasına bakılmaksızın bu metriğin hesaplanmasında kullanılmadığını belirtmek gerekir.

$$Precision = \frac{TP}{TP + FP}$$

Recall isimli metrik, eğitilmiş modelin görüntü üzerinde tespit etmesi beklenen nesnelerin kaçını tespit ettiğini belirtmektedir. Recall, TP metriğinin TP ve FN metriklerine bölümüyle hesaplanır. Görüntü üzerinde nesne olmayan görüntülerin sayısı bu hesaplamanın dışında tutulmaktadır.

$$Recall = \frac{TP}{TP + FN}$$

Bahsi geçen doğruluk metriklerinin dışında en çok kullanılan ve genel olarak bir modelin doğruluğunu niteleyen metrikler de bulunmaktadır. Bunlar f-score, AP (Average Precision) ve mAP (Mean Average Precision) metrikleridir. Söz konusu metrikler de dolaylı olarak TP, TN, FN ve FP metriklerine bağlı olarak hesaplanmaktadır. Bu metriklerden f-score, daha önce bahsi geçen precision ve recall metriklerinin harmonik ortalaması alınarak hesaplanır.

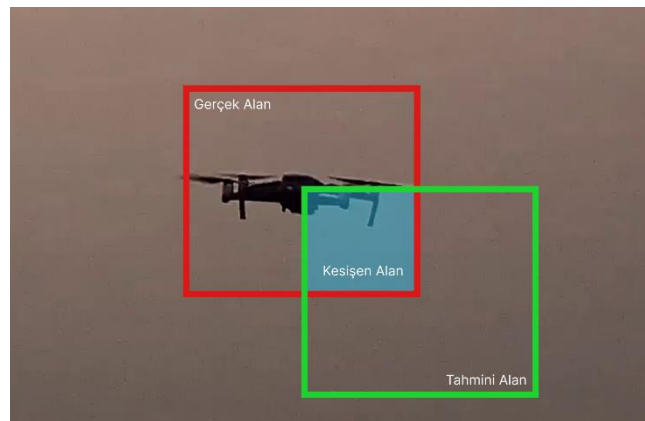
$$F - Score = \frac{2 * Precision * Recall}{Precision + Recall}$$

Bir modelin doğruluğunun değerlendirilmesinde F-score ile birlikte AP ve mAP metriklerinin de kullanılmaktadır. AP değeri precision ve recall değerlerinin kesişimlerinin ortalaması ile hesaplanmaktadır. mAP değeri ise modelin tespit etmesi gereken her bir farklı nesnenin/sınıfın AP değerlerinin ortalamasından ibarettir. Eğitilen modelde tek bir sınıf olması durumunda AP değeri mAP değerine eşit olmaktadır.

$$AP = \frac{1}{11} \sum_{k \in \{0,0.1,...,1\}} Precision(Recall_k)$$

$$mAP = \frac{1}{n} \sum_{i=1}^{i=n} AP_i , \quad n = \text{nesne sayısı}, \quad AP_i = i. \text{ nesnenin AP değeri}$$

Bahsi geçen metrikler genel itibari ile eğitilmiş bir modelin doğruluğunun hesaplanmasında kullanılmaktadır. Bununla birlikte hem nesne tespit hem de nesne takibi metotlarında doğruluğun hesaplanabilmesi için kullanılan IoU (Intersection over Union) adında bir metrik daha bulunmaktadır. IoU metriği yalnızca tespitin yapılması durumunda değil, görüntü üzerinde tespit edilen nesnenin gerçekte ne kadar örtüştüğünü de göstermektedir. Bunun için görüntü üzerinde bulunan nesneyi belirtmek için kullanılan alan ve modelin tahmini neticesinde nesnenin içerisinde olduğu varsayılan alan kullanılmaktadır. Tahmini alan ile gerçek alanın kesişimlerinin birleşimlerine oranı IoU metriğini göstermektedir. Burada bahsi geçen alanlar şekil 36'da örnek bir görüntü üzerinde gösterilmiştir.



Şekil 36. IoU metriğinin hesaplanmasında kullanılan alanların temsili gösterimi.

$$IoU = \frac{|Tahmini Alan \cap Gerçek Alan|}{|Tahmini Alan \cup Gerçek Alan|}$$

Doğruluk metrikleri dışında performans metriği olarak FPS (Frames per Secons) kullanılmaktadır. Bu durumda belirlenmiş sabit bir donanım üzerinde bir takip veya tespit metodunun bir saniyede kaç görüntüyü işleyebildiği test edilen metodun performansını göstermektedir. Ayrıca bazı durumlarda FPS metriğinin tersi olan SPF (Seconds per Frame) de kullanılabilir. SPF ise bir görüntünün kaç saniyede işlendiğini niteler.

Bahsi geçen doğruluk ve performans metrikleri bu çalışma kapsamında elde edilen bulguları ölçümlemek için kullanılmıştır. Ayrıca bu çalışma kapsamında nesne tespiti ve nesne takibi için birlikte çalışan bir çerçeve oluşturulduğundan ortalama IoU ve ortalama FPS değerlerinin hesaplanabilmesi için de bazı işlemler yapılmıştır. Ortalama IoU ve ortalama FPS metrikleri özellikle birlikte çalışma esnasında bir takip algoritmalarının YOLO ile birlikte çalışmaları durumunda birbirleri ile kıyaslanabilmeleri için hesaplanmıştır. Aşağıda ortalama IoU ve ortalama FPS metriklerinin hesaplanması için kullanılan formüller paylaşılmıştır.

$$FPS_{MEAN} = \frac{(YFT + 1) * FPS_{YOLO} * FPS_{TRACKER}}{FPS_{TRACKER} + (YFT * FPS_{YOLO})}$$

$$IoU_{MEAN} = \frac{IoU_{YOLO} + (YFT * IoU_{YOLO} * IoU_{TRACKER})}{YFT + 1}$$

ARAŞTIRMA BULGULARI ve TARTIŞMA

İşbu tez çalışmasında; temel otonomi kabiliyetlerine sahip düşük maliyetli donanım ile tümleşik olarak nesne tespiti ve takibi yapabilen bir döner kanatlı İHA'nın geliştirilmesi amaçlanmıştır. Yapısal, mekanik, aviyonik ve yazılımsal açılardan yapılan çalışmanın bulgu ve çıktıları bu bölümde paylaşılmıştır. Paylaşılan bulgular, bölümlere ayrılarak her biri ayrı başlık altında sunulmuştur.

İHA'nın Yapısal, Mekanik ve Aviyonik Çıktıları

Çalışma kapsamında üretilen İHA'nın tasarım aşamasında, 3050 gram ağırlığında olacağı tahmin edilmiştir. Bu tahmini ağırlık için üreticiler tarafından yayınlanan resmi dokümanlar öncelikli kaynak olarak kullanılmıştır. Bunun dışında vida, somun, bant, kablo gibi sarf malzemelerin ağırlıkları için tahmini bir ağırlık, son toplama eklenmiştir. Neticede kavramsal olarak üretilecek İHA'nın tahmini ağırlığı ortaya çıkmıştır. Tahmini ağırlık ile gerçekleştirilecek İHA'nın özellikleri hakkında hesaplamalar yapılmıştır.

İHA'nın üretiminin ardından gerçekleşen ağırlık ile bu hesaplamaların tekrar edilmesi durumunda üretilen sistemin özellikleri ortaya çıkacaktır. 3 gram hata payına sahip bir hassas teraziyle İHA'nın ağırlığı ölçüldüğünde, gerçekleşen ağırlığın 3261 gram olduğu ölçülmüştür (Şekil 37). Dolayısıyla tasarım ile üretim arasında 211 gram fark ortaya çıkmıştır.



Şekil 37. İHA ağırlık ölçümünden alınan sonucun fotoğrafı.

Ortaya çıkan ağırlık farkı araştırılırken öncelikle bir üreticiden temin edilmeyen elde yapılmış parçalar üzerinde durulmuştur. Burada en büyük farkı oluşturan birimlerin 3B yazıcı vasıtasıyla PLA malzemeden üretilen parçalar (Şekil 38) olduğu ortaya çıkmıştır. Bu parçaların ağırlığı diğer sarf malzemelerle birlikte tahmini bir değer olarak kavramsal tahmini ağırlığa eklenmiştir. Üretimleri yapıldıktan sonra yapılan ölçümlerde parçaların toplam ağırlıklarının 287 gram olduğu anlaşılmıştır.



Şekil 38. 3B çıktılarının ağırlıklarının ölçülmesi (solda) ve üretilmiş parçalar (sağda).

Tasarım aşamasında tüm sarf malzemeler, 3B çıktılar ve bazı küçük elektroniklerin toplamda 350 gram ağırlığında olacağı tahmin edilmişti. Yalnızca 3B çıktılarının 287 gram olması bu tahminin tutarlı olmadığını göstermektedir. Neticede tasarım aşamasında 350 gram olarak belirtilen bu kalemin gerçekleşen ağırlığının 561 gram olduğu ve 3B çıktılar dışında diğer malzemelerin 274 gram ağırlığında olduğu söylenebilir.

Gövdede montaj için çift taraflı bant ve kablolar dışında 102 adet vida ve 79 adet somun kullanılmıştır. Bununla birlikte 62 parça konnektör ile aviyonik konfigürasyon daha modüler hale getirilmiştir. Hem sarf malzemeler hem de önceki bölümlerde gösterilmiş olan tabloda yer verilmeyen LoRa modülü, buzzer, safety switch gibi aviyonik bileşenler söz konusu 274 gramlık ağırlığı oluşturmuşlardır. Gerçekleşen ağırlık ile tahmini ağırlık arasındaki fark İHA'nın uçuş süresi gibi performans özelliklerinin de değişmesine sebep olmuştur.

Gerçekleşen ağırlık ile TWR değeri yeniden hesaplandığında sonucun 2,23 olduğu ortaya çıkmıştı. Tasarım aşamasında ortaya çıkan TWR değeri 2,39'du. Gerçekleşen ve tasarlanan TWR değeri arasında 0,16 fark olduğu görülmektedir. Ancak tasarım aşamasında hedeflenen TWR değeri, gerçekleşen İHA'da da yakalanmıştır. Bu durumda gerçekleşen ağırlığın TWR hedefi açısından kabul edilebilir olduğu söylenebilir.

$$TWR = \frac{T}{W} = \frac{7280 \text{ gr}}{3261 \text{ gr}} = 2,23$$

Gerçekleşen ağırlığın 3261 gram olması, motor başına düşen gövde ağırlığının 402 gram olduğunu göstermektedir. Bu durumda İHA'nın her bir motoru askı durumunda 4,6 A akım talep etmektedir. Yani üretilen İHA'nın askı durumunda kalabilmesi için her bir motor tasarım aşamasındaki tahmine göre 0,6 A daha fazla akım talep edecektir. Toplamda ise üretilen İHA'nın tasarımda öngörülenden 4,8 A daha fazla akıma ihtiyaç duyacağı görülmektedir.

Uçuş süresi hesaplamalarında, kullanılan iki bataryadan ilk bitecek olan dikkate alındığından, uçuş süresi hesaplaması 4 motor üzerinden gerçekleştirilmiştir. Askı durumunda her motor 4,6 A akım çekeceğinden ilgili güç dağıtım sisteminde motorlar 18,4 A akım talep etmektedir. Bunun dışında diğer bileşenler için batarya gerilime eşit olmak kaydıyla 2,3 A akım gerekeceği önceki bölümlerde hesaplanmıştır. Dolayısı ile İHA askıdayken ilk tükenen batarya 11,1 V nominal gerilimde 20,7 A akım verecektir. Bu şartlarda harcanan güç 229,77 W olacaktır.

İlk tükenen bataryanın askı durumu için harcayacağı güç dikkate alınarak uçuş süresi yeniden hesaplanmış ve gerçekleşen uçuş süresi ortaya çıkarılmıştır. Burada hatırlatmak gerekir ki uçuş süreleri; tüm uçuş süresi ve sürdürülebilir uçuş süresi olarak ikiye ayrılmıştır. Sürdürülebilir uçuş süresi, bataryanın %80 kapasitesinin harcanması durumunu; tüm uçuş süresi ise bataryanın %100 kapasitesinin kullanılması durumunu göstermektedir. Kullanılan bataryalar 11,1 V nominal gerilimde ve 8000 mAh kapasiteye sahip olduğundan batarya enerjisi 88,8 Wh olarak hesaplanmıştır. Batarya enerjisinin talep edilen güce bölünmesi durumunda saat cinsinden bataryanın enerjisinin tükenmesi için gerekli zaman hesaplanabilir.

$$\text{Tüm Uçuş Süresi (dakika)} = \frac{Wh}{W} 60 = \frac{88,8 \text{ Wh}}{229,77 \text{ W}} 60 = 23,19 \text{ dakika}$$

$$\text{Sürdürülebilir Uçuş (dakika)} = \frac{Wh}{W} 48 = \frac{88,8 \text{ Wh}}{229,77 \text{ W}} 48 = 18,55 \text{ dakika}$$

Yapılan hesaplamalar neticesinde İHA'nın askı durumunda, tüm batarya kapasitesini kullanarak 23,19 dakika uçuş yapabileceği; bataryasının %80 kapasitesini kullandığında ise 18,55 dakika uçuş yapabileceği hesaplanmıştır. Motorların tam güç ile çalışması durumunda her bir motorun 15,5 A akım talep edeceği önceki bölümlerde belirtilmişti. Benzer şekilde motorların tam güçle çalıştığı durum için hesaplamalar yapıldığında bataryadan 713,73 W güç talep edileceği görülmektedir.

$$\text{Tüm Uçuş Süresi (dakika)} = \frac{Wh}{W} 60 = \frac{88,8 Wh}{713,73 W} 60 = 7,47 \text{ dakika}$$

$$\text{Sürdürülebilir Uçuş (dakika)} = \frac{Wh}{W} 48 = \frac{88,8 Wh}{713,73 W} 48 = 5,97 \text{ dakika}$$

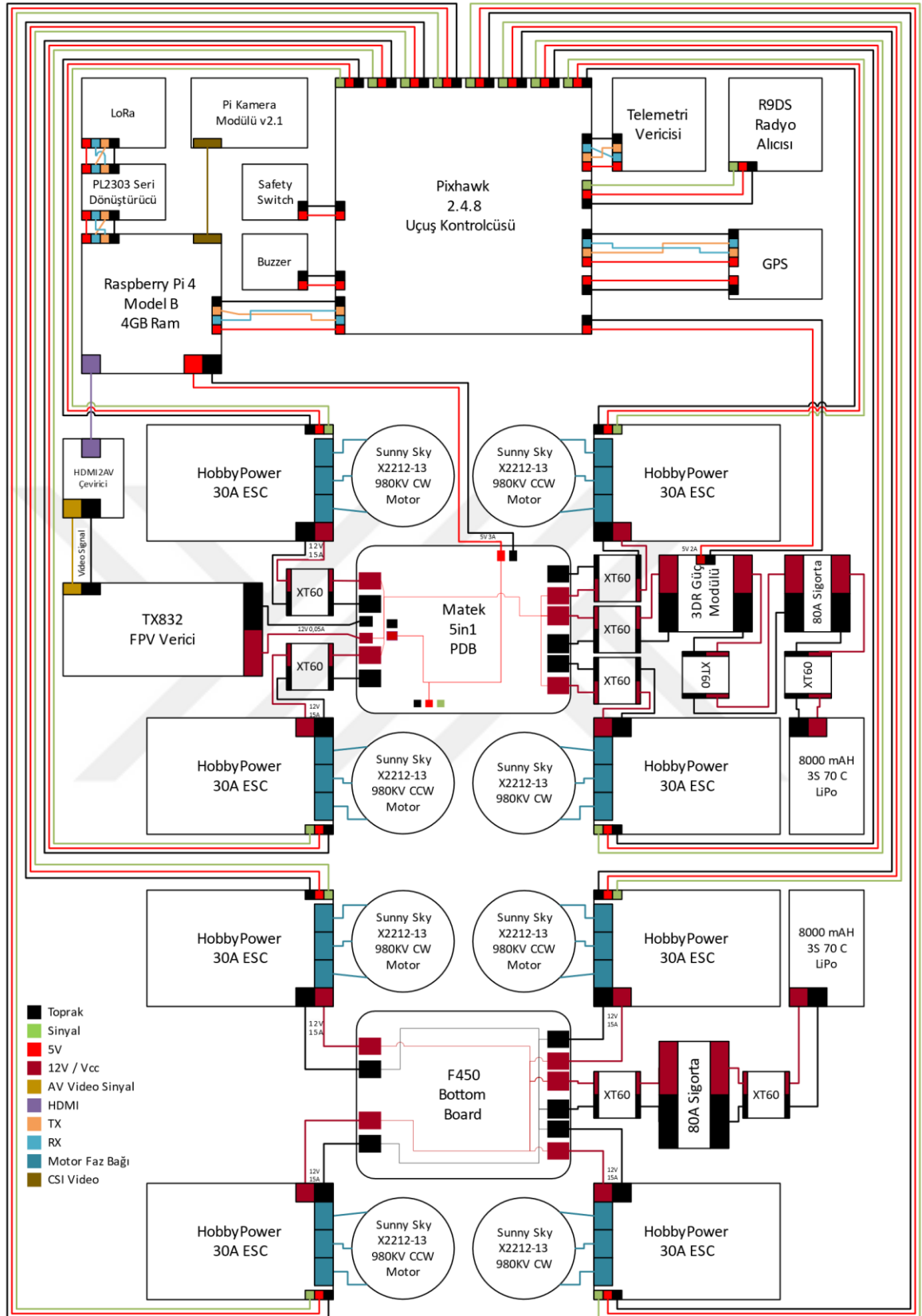
Sonuçta motorların tüm güçle çalışması ve bataryanın tüm kapasitesinin kullanılması durumunda 7,47 dakika uçuş yapılabilir. Bataryanın %80 kapasite ile kullanılması durumunda ise 5,97 dakika uçuş yapılabilir. Bu durumda, karma uçuş süresinin motorlar tam güçle çalışırken gerçekleşen uçuş süresiyle askı durumundaki uçuş süresi arasında olacağı söylenebilir. Yani bataryanın tam kapasitesinin kullanılması durumunda karma uçuş süresi 7,47 dakika ile 23,19 dakika arasında olacaktır. Sürdürülebilir karma uçuş süresi ise 5,97 dakika ile 18,55 dakika arasındadır. Yapılan uçuş testlerinde de uçuş sürelerinin bu değerler arasında olduğu doğrulanmıştır.

İHA üzerinde kullanılan iki bağımsız güç dağıtım sistemi olduğundan önceki bölümlerde bahsedilmiştir. Yukarıdaki hesaplamalar yüksek güç tüketimine sahip güç dağıtım sistemine ilişkin hesaplamalardır. İkinci güç dağıtım sisteminde ise motorlar dışında akım talep eden elektronik bileşen bulunmamaktadır. Bu sebeple ikinci güç dağıtım sistemi motorların maksimum güçle çalışması durumunda 62 A, askı durumundaysa 18,4 A akım talep etmektedir. Neticede güç dağıtım sistemleri toplamda anlık olarak en fazla 126,3 A akım talep etmektedir. Askı durumundaysa güç dağıtım sistemlerinden 39,1 A akım talep edilmektedir.

Son olarak gerçekleşen ağırlık beklenenin üzerinde olduğundan motorların verimliliklerinde düşüş yaşanmıştır. Bir motorun verimliliği, ürettiği itki kuvvetinin talep ettiği güce oranı ile ölçülmektedir. Daha önce İHA gövdesindeki tek bir motorun sağlaması gereken itki kuvvetinin 402 gram olacağından ve 4,6 A akım talep edeceğinden bahsedilmiştir. Bu durumda tek bir motor, askı durumunu sağlamak için 51,06 W güç tüketecektir. Dolayısıyla İHA, askı durumundayken her bir motor 7,87 g/W verimlilik ile çalışacaktır.

$$\text{Motor Verimliliği} = \frac{g}{W} = \frac{402 g}{51,06 W} = 7,87 g/W$$

İHA'nın elektronik konfigürasyonuna ait kavramsal tasarım önceki bölümlerde gösterilmiştir. Kavramsal tasarımda detaylı bağlantılar ve kullanılan tüm bileşenler olmaksızın varsayımsal çalışma gösterilmiştir. Çalışma neticesinde üretilen İHA'nın tüm aviyonik bileşenlerinin bağlantıları aşağıdaki görselde sunulmuştur (Şekil 39).



Şekil 39. İHA gövdesindeki tüm aviyonik bileşenleri ve aralarındaki bağlantıları gösteren diyagram. Çizimde batarya gerilimi 12 V kabul edilmiştir. 12 V ile çalışmayan bileşenlerin gerilim ve akım değerleri kendi çalışma gerilimleri dikkate alınarak yazılmıştır. Motor ve ESC’ler arasında kullanılan konnektörler gösterilmemiştir.

Nesne Tespiti

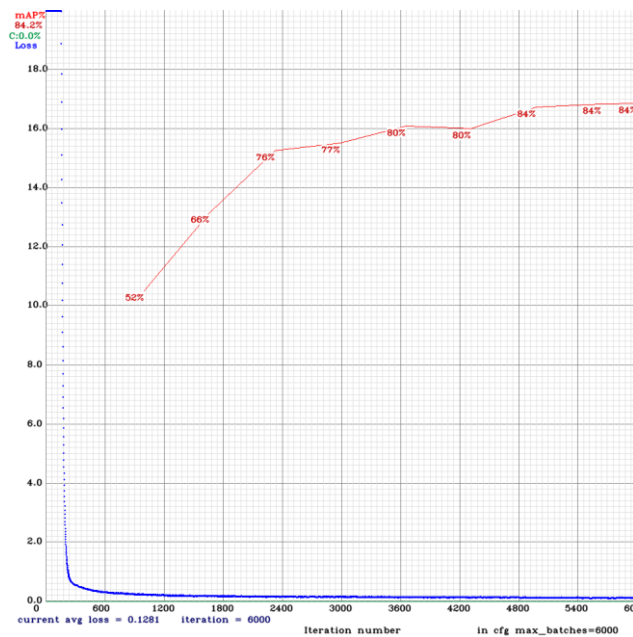
Nesne tespiti için YOLO v4 Tiny kullanılmıştır. Eğitim için kullanılan veri kümesi ise Det-Fly isimli veri kümesidir. İlişkili ön işlemler gerçekleştirildikten sonra veri kümesinin %80'i eğitim, %20'si test için ayrılmıştır. Veri kümesinde İHA nesnesi, drone ismiyle etiketlenmiştir. YOLO modelinin eğitimi için gerekli değerlerin atanmasında, varsayılan değerler veri kümesi için hesaplanıp uygulanmıştır. Söz konusu değerler aşağıdaki tabloda sıralanmıştır.

Tablo 3. YOLO v4 Tiny modelinin eğitilmek istenen veri kümesine ilişkin config dosyasında düzenlenen değerler.

İsim	Değer
batch	64
max_batches	6000
subdivision	16
steps	4800, 5400
classes	1
filters*	18

* filters değeri iki YOLO katmanına atanmıştır.

Eğitim için Google Colab kullanılmıştır. 6000 iterasyonda tamamlanan eğitim yaklaşık 12 saat sürmüştür. Eğitim esnasında mAP ve Loss değerlerinin hesaplanması için gerekli parametreler kullanılmıştır. Sonuç olarak Loss ve mAP değerlerini gösteren bir grafik eğitim işleminden çıktı olarak alınmıştır (Şekil 40). Loss değeri her iterasyonda hesaplanırken mAP değeri periyodik olarak hesaplanmıştır. İlişkili grafik aşağıda paylaşılmıştır.



Şekil 40. Training neticesinde üretilen Loss ve mAP grafiği.

Eğitim neticesinde elde edilen ağırlık değerleri ile model test edilmiştir. Yapılan testler neticesinde mAP ve loss dışındaki değerleri de üretmek için her bininci iterasyon için test tekrar edilmiştir. Dolayısı ile 1000, 2000, 3000, 4000, 5000 ve 6000 numaralı iterasyonlarda üretilmiş ağırlık değerleri ile testler tekrar edilmiş ve eğitim boyunca söz konusu değerlerin nasıl değiştiği ortaya konmuştur.

Test için 2656 görsel kullanılmıştır. Bu görsellerden 2654 tanesi YOLO tarafından özgün doğru olarak sayılmıştır. Dolayısı ile test sonucunda ortaya çıkacak TP ve FN değerlerinin toplamının 2654 olması beklenecektir.

Test sonucunu doğrudan etkileyecek olan `conf_thresh` değeri 0,25 olarak atanmıştır. Bu değer YOLO algoritmasında NMS (Non Maximum Suppression) aşamasında kullanılan bir parametredir ve tespit edilen objelerden güven skoru 0,25 altında olanların temizlenmesini sağlamaktadır.

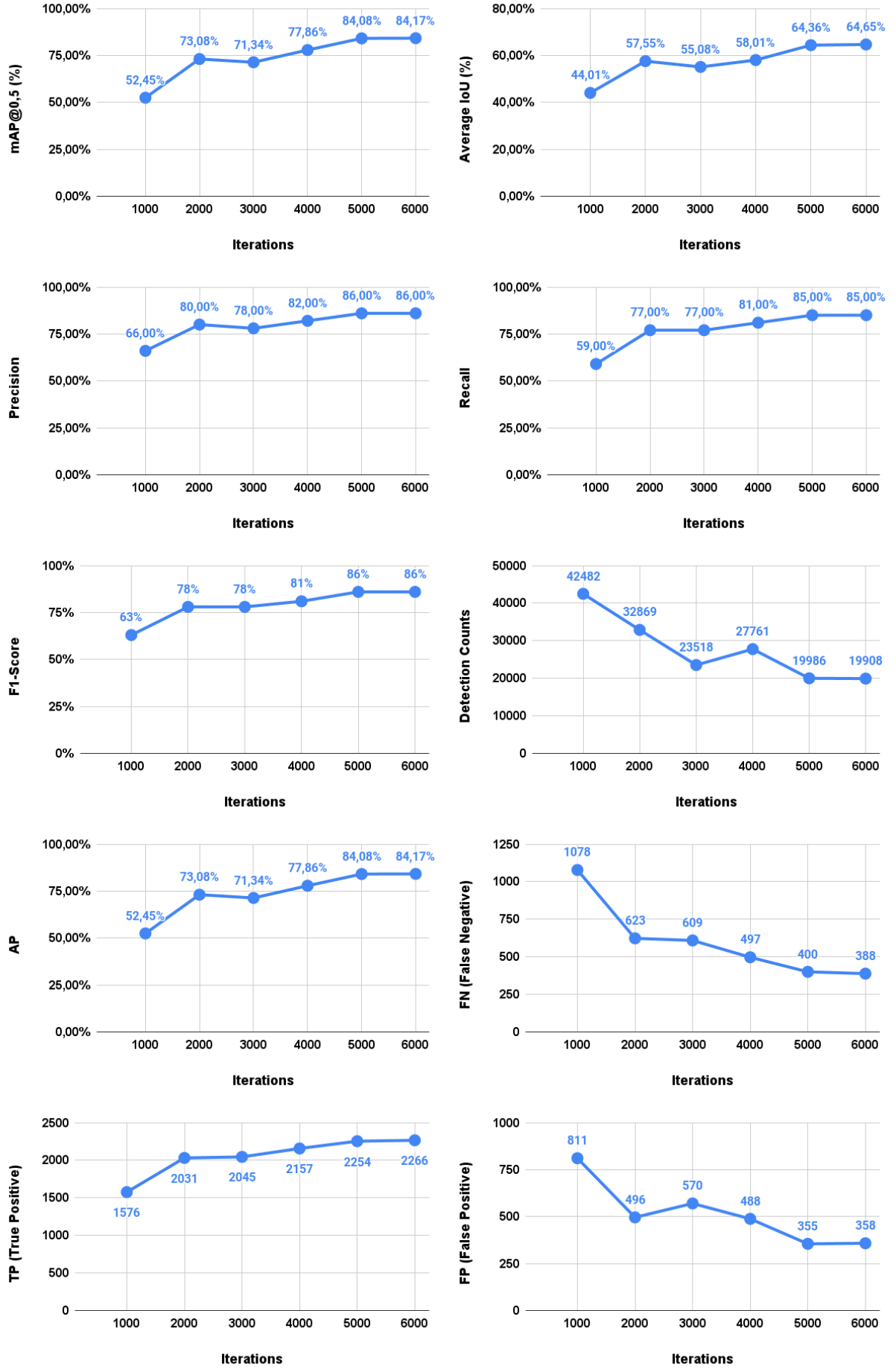
Test sonucunu doğrudan etkileyecek olan diğer parametre ise `IoU_threshold` değeridir. Bu değer IoU değeri 0,5'in altında olan tespitlerin 0'ın üzerinde bir IoU değeri olsa bile FP olarak sayılmasını sağlamaktadır. Yani tespit olsa dahi yeterli IoU üretilmediğinden tespit yokmuş gibi davranılmasını sağlamaktadır.

`IoU_threshold` ve `conf_thresh` değerleri doğrudan etkileyecektir. Bunların dışındaki parametreler daha ziyade test ortamını belirtmek amacıyla açıklanmıştır. Söz konusu değerler için özellikle bir değişiklik yapılmamış olup varsayılan değerlere sadık kalınmaya çalışılmıştır. Bahsi geçen parametreler aşağıdaki tabloda sıralanmıştır.

Tablo 4. Test ortamına ilişkin parametreler.

Parametre Adı	Değeri
<code>image_count</code>	2656
<code>unique_truth_count</code>	2654
<code>class_id</code>	0
<code>Name</code>	Drone
<code>conf_thresh</code>	0,25
<code>IoU_threshold</code>	0,50

Test neticesinde alınan sonuçlar aşağıdaki görselde (Şekil 41) grafiklerle gösterilmiştir. Devamında ise her bir değerin neyi ifade ettiği açıklanmıştır.



Şekil 41. Test sonucunda elde edilen sonuçlar.

Test neticesinde elde edilen TP, FP, FN metriklerinin değerleri çıktı olarak alınmıştır. Burada belirtmek gerekir ki FP değeri; yalnızca görüntü üzerinde İHA olmayan bir alanda İHA tespit edilmesini değil, IoU threshold parametresi sebebiyle IoU değerinin %50'den küçük olması durumları da kapsamaktadır. Test neticesinde TN metriğine ilişkin sonuç üretilmemesinin sebebi TN metriğinin precision, recall ve f-score gibi metriklerin hesaplanmasında kullanılmamasıdır. Bahsi geçen TP, FP ve FN metriklerine göre test sonuçları tablo 5'te gösterilmiştir.

Tablo 5. Veri kümesinde yapılan test neticesinde elde edilen; TP, FP ve FN değerlerinin 6000 iterasyon boyunca aldığı değerler gösterilmiştir.

İterasyon	TP*	FP*	FN
1000	1576	811	1078
2000	2031	496	623
3000	2045	570	609
4000	2157	488	497
5000	2254	355	400
6000	2266	358	388

* mAP değerlerinin doğrulamasında kullanılmışlardır.

Test neticesinde TP, TN, FP ve FN metriklerinden faydalanılarak hesaplanan recall, precision ve f-score metrikleri tablo 6'da listelenmiştir. Burada f-score değeri, precision ve recall değerlerinden türetildiği için bu metriklerle birlikte listelenmiştir.

Tablo 6. Veri kümesinde yapılan test neticesinde elde edilen precision, recall ve f-score değerleri.

İterasyon	Recall	Precision	F-Score
1000	0,59	0,66	0,63
2000	0,77	0,80	0,78
3000	0,77	0,78	0,78
4000	0,81	0,82	0,81
5000	0,85	0,86	0,86
6000	0,85	0,86	0,86

Eğitilen modelin doğruluğunun anlaşılmasında f-score ile birlikte AP ve mAP metrikleri de göz önüne alınmalıdır. Yapılan testte yalnızca tek bir sınıf bulunduğundan dolayı AP değerleri mAP değerleriyle eşit çıkmıştır. Tablo 7'de bahsi geçen metriklerin değerleri her bininci iterasyon için gösterilmiştir.

Tablo 7. Veri kümesinde yapılan test neticesinde elde edilen AP, mAP ve IoU değerleri. Tabloda her bin iterasyonda alınan değerler gösterilmiştir.

İterasyon	AP	mAP@0,5	IoU
1000	%52,45	0,524513	0,44
2000	%73,08	0,730826	0,58
3000	%71,34	0,713445	0,55
4000	%77,86	0,778607	0,58
5000	%84,08	0,840799	0,64
6000	%84,17	0,841708	0,65

Sonuç olarak eğitilen model %84,17 doğruluk ile nesne tespiti yapabilmektedir. YOLO v4 Tiny ve Darknet kullanılarak eğitilen modelin f-score değeri ise %86 olarak ölçülmüştür. Bu değerler modelin doğruluğu hakkında bilgi verse de performansı için gerekli bilgileri sunmamaktadır. Performansın ölçümü için FPS değerlerinin hesaplanması gerekmektedir.

Nesne tespiti testlerinde FPS değerlerinin ölçülmesi için önceden belirlenmiş bir video kullanılmıştır (Şekil 42). Kullanılan video 30 FPS ile kaydedilmiş ve toplamda 713 görüntüden oluşmaktadır. Bu test için OpenCV kütüphanesi kullanılmıştır. OpenCV kütüphanesinin dnn adı altında toplanmış işlevselliklerinden faydalanılmıştır. Test görüntü çözünürlüğünün değiştirilmesiyle 3 farklı şekilde gerçekleştirilmiştir. Test için çalışma kapsamında geliştirilen İHA'nın üzerinde bulunan Raspberry Pi 4 Model B (4 GB RAM) kullanılmıştır. Testte kullanılan confidence değeri ise 0,5 olarak belirlenmiştir. Yani %50 ihtimalden daha düşük olan tahminler dikkate alınmamıştır.



Şekil 42. Test esnasında alınmış bir görüntü.

Test esnasında değiştirilen değer, nesne tespiti için kullanılacak görüntünün yeniden boyutlandırılmasını sağlamak için kullanılmaktadır. Bu değer bu bölümde görüntü çözünürlüğü olarak isimlendirilecektir. Görüntü çözünürlüğünün varsayılan değeri 416 piksel genişlik ve 416 piksel yüksekliktir (416x416). Düşük maliyetli donanım üzerinde performansı arttırmak için bu değer iki kez azaltılmıştır. Dolayısı ile 416x416, 312x312 ve 209x209 değerleri için nesne tespiti performansı test edilmiştir.

Her bir testin öncesinde ve sonrasında Raspberry Pi 4 Model B'nin (4 GB) işlemci sıcaklığı ölçülmüştür. Ölçüm neticesinde videodaki her bir görüntü için sıcaklıktaki artış değeri hesaplanmıştır. İşlemci sıcaklığındaki artış performansı etkilediğinden performans açısından önemli bir ölçüt olarak görülmüştür. Testlerin başlangıcında 70,6 °C sıcaklığın oluşması beklenmiş veya başka görevler verilerek bu sıcaklığa ulaşılması sağlanmıştır. Başlangıç değeri sağlandıktan sonra 713 görüntüden oluşan video üzerinde eğitilen model çalıştırılmıştır. Test bittikten hemen sonra tekrar sıcaklık ölçümü alınmıştır. Aşağıdaki tabloda ilgili sıcaklık bilgileri listelenmiştir.

Tablo 8. 713 frame boyunca YOLO v4 Tiny ile nesne tespiti yapılması durumunda Raspberry Pi 4 Model B'nin sıcaklık değerleri.

Görüntü Çözünürlüğü	İlk CPU Sıcaklığı*	Son CPU Sıcaklığı*	Görüntü Başına Sıcaklık Artışı*
209 x 209 Piksel	70,6	79,4	0,012
312 x 312 Piksel	70,6	79,9	0,013
416 x 416 Piksel	70,6	80,3	0,014

* Santigrat Derece (°C)

Görüntü çözünürlüğü arttıkça işlem yükü arttığından ve görüntüyü işlemek için daha fazla zaman harcandığından dolayı farklı çözünürlüklerde yapılan testlerin farklı sürelerde tamamlandığının göz ardı edilmemesi gerekmektedir. Hem süre hem de işlem maliyeti açısından çözünürlük artışı kendisini sıcaklık artışı olarak da göstermektedir.

Test süresince her bir görüntü üzerinde nesne tespiti yapılmıştır. Nesne tespiti yapılmadan önce ve nesne tespiti sonuçlandığında zaman bilgisi alınarak bir görüntü üzerinde nesne tespiti yapmak için gerekli süre hesaplanmıştır. Bir görüntü için gerekli zaman bilgisi ile bir saniye içerisinde aynı işlemin kaç kez yapılabileceği hesaplanarak SPF (Seconds per frame) verisi FPS (Frames per second) verisine çevrilmiştir. Bu değer her bir görüntü için hesaplanmıştır ve anlık FPS olarak isimlendirilmiştir. Anlık FPS verilerinin ortalaması alınarak her bir test için ortalama FPS bilgisi üretilmiştir. Ayrıca anlık FPS verilerinden ilk,

son, en yüksek ve en düşük değerler de çalışma kapsamında değerlendirilmiştir. Söz konusu FPS değerleri aşağıdaki tablo 9’da listelenmiştir.

Tablo 9. FPS Skorları.

Görüntü Çözünürlüğü (Piksel x Piksel)	İlk Anlık FPS*	Son Anlık FPS*	En Düşük FPS	En Yüksek FPS	Ortalama FPS
209 x 209	1,902	1,796	1,387	2,712	1,996
312 x 312	1,105	0,797	0,689	1,362	0,895
416 x 416	0,716	0,474	0,424	0,819	0,514

* İlk ve son anlık FPS değerleri video görüntülerinden ilkinin ve sonuncusunun işlenmesi için gerekli sürenin FPS cinsinden ifadesidir.

FPS skorlarından ilk ve son anlık FPS verileri incelendiğinde çözünürlük değeri fark etmeksizin bir düşüş yaşandığı görülmektedir. Bu durum sıcaklık artışıyla birlikte incelendiğinde işlemci sıcaklığının performansa etkisi olarak yorumlanabilir. Bununla birlikte en yüksek ve en düşük FPS değerleri incelendiğinde FPS değerlerinin değişiklik gösterdiği aralık anlaşılmaktadır. Test neticesinde üretilen en önemli veri ise ortalama FPS bilgisidir. Farklı çözünürlüklerde ortalama FPS değerinin 3,88 kat artış gösterdiği görülmektedir.

Sonuç olarak, düşük maliyetli donanım üzerinde maksimum ortalama FPS değeri 209x209 piksel çözünürlükte 1,996 FPS olarak ölçülmüştür. Bu bir görüntü üzerinde nesne tespiti yapmanın yaklaşık olarak 0,5 saniye süreceği bilgisini vermektedir.

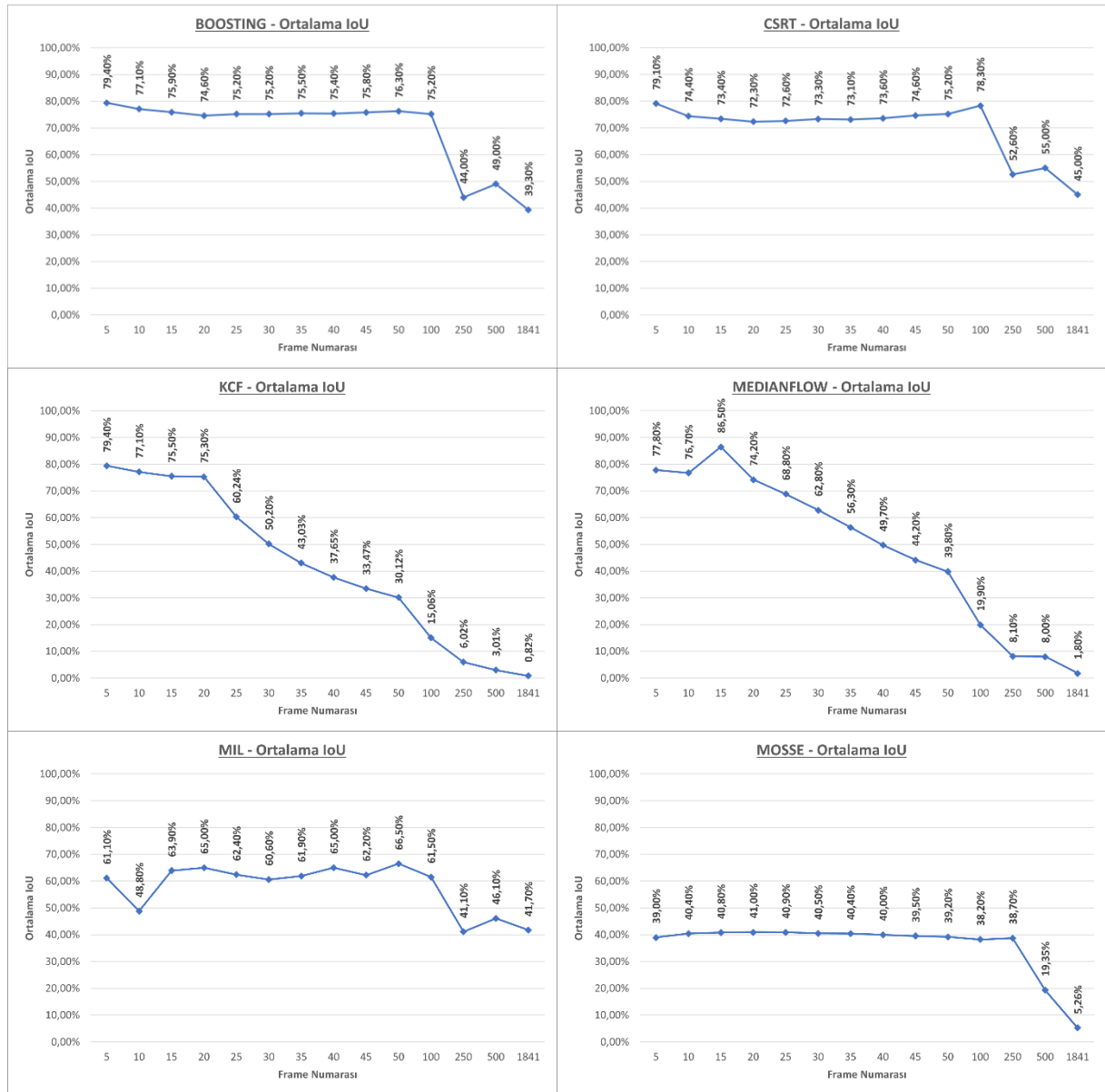
Nesne Takibi

Nesne tespiti ve takibi için yapay sinir ağı tabanlı metotların ve takip algoritmalarının birlikte çalıştırılacağı önceki bölümlerde açıklanmıştı. Ayrıca önceki bölümlerde kullanılacak takip algoritmasına uçuş öncesinde operatör tarafından karar verileceği de açıklanmıştı. Bunun sebebi farklı takip algoritmalarının birbirlerinden farklı avantaj ve dezavantajlar sergilemeleri olmuştur. Bu bölümde, takip algoritmalarına ilişkin yapılan testlerden alınan sonuçlar sunulmuş ve takip algoritmalarının davranışları açıklanmıştır.

Önceki bölümlerde takip algoritmalarının test edilmesi için 1841 görüntüden oluşan bir veri kümesi kullanılacağından bahsedilmiştir. Söz konusu veri kümesi üzerinde döner kanatlı bir İHA ekran dışına çıkmayacak şekilde takip edilmektedir. Görüntülerdeki İHA’yi çevreleyen en küçük dikdörtgen verileri test esnasında bir dosya içerisinde okunmuştur. Testte önceden kaydedilmiş olan dikdörtgen ile takip algoritmasının ürettiği dikdörtgenin IoU değerleri hesaplanmıştır. IoU hesaplamaları sırayla 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 100,

250, 500 ve 1841 görüntü için tekrar edilmiştir. Bu sayede takip algoritmasının ne kadar süreyle doğru bir takip yapabileceğine ilişkin verilerin üretilmesi amaçlanmıştır.

IoU değerinin hesaplanmasında gerçek ve tahmini alanların genişliği, yüksekliği ve koordinatları birbirine eşit olduğunda IoU değeri 1 olarak hesaplanmaktadır. Alanlar kesişmediğindeyse bu değer 0 olarak hesaplanmaktadır. Dolayısı ile takip algoritmasının takibi sürdürdüğü müddetçe 0 ile 1 arasında bir IoU değeri üretmesi beklenmektedir. Takip algoritması, takibi kaybettiğindeyse IoU değeri 0 olacaktır. Görüntülerde hesaplanan IoU verilerinin ortalaması ise takip algoritmasının doğruluğu hakkında en önemli bilgilerden birisidir. Şekil 43’de takip algoritmalarının ortalama IoU değerleri sunulmuştur.



Şekil 43. BOOSTING, CSRT, KCF, MEDIANFLOW, MIL ve MOSSE isimli takip algoritmalarının ortalama IoU değerlerinin zaman içerisinde değişimi gösterilmiştir.

Yapılan testlerde ve test esnasındaki gözlem neticesinde en doğru takibin CSRT algoritması ile yapıldığı görülmüştür. En az doğruluğa sahip takibin ise KCF algoritması ile elde edildiği görülmüştür. Ancak takip algoritmalarının daha doğru değerlendirilebilmesi için çeşitli verilere ihtiyaç duyulmaktadır.

Test esnasında yapılan gözlem neticesinde yalnızca ortalama IoU değeri ile takip algoritmalarının doğruluklarının değerlendirilemeyeceği anlaşılmıştır. Bu sebepten takip algoritmasının ilk ve son görüntülerdeki IoU değerleri, maksimum ve minimum IoU değerleri, takibin hangi görüntülerde başarısız olduğu bilgileri de doğruluğa ilişkin sonuçlara dahil edilmiştir. Bu sayede takip algoritmalarının davranışlarına ilişkin daha doğru bir değerlendirme yapılabilmesi mümkün olmuştur. Aşağıdaki tabloda anlık ve ortalama IoU değerleri birlikte sunulmuştur.

Tablo 10. 1841 görüntü üzerinde yapılan takip testi neticesinde elde edilen anlık ve ortalama IoU değerleri.

Algoritma	İlk Anlık IoU*	Son Anlık IoU*	En Düşük Anlık IoU*	En Yüksek Anlık IoU*	Ortalama IoU
BOOSTING	0,889	0	0	0,894	0,393
CSRT	0,804	0,551	0,121	0,894	0,450
KCF	0,889	0	0	0,889	0,008
MEDIANFLOW	0,814	0	0	0,814	0,018
MIL	0,7	0,551	0,09	0,809	0,417
MOSSE	0,415	0	0	0,457	0,053

* Anlık IoU değerleri tek bir görüntü üzerinde yapılan IoU hesaplamasını nicelemektedir.

Yapılan test neticesinde üretilen anlık ve ortalama IoU değerleri birlikte incelendiğinde CSRT ve MIL algoritmalarının diğer algoritmalarla göre daha doğru sonuçlar ürettiği anlaşılmaktadır. Ayrıca bu iki algoritma sıfır IoU değeri üretmediğinden test boyunca takibi devam ettirdikleri anlaşılmaktadır. Diğer algoritmaların minimum IoU değerlerinde sıfır değeri görüldüğünden dolayı bir noktada takibi kaybettikleri anlaşılmaktadır. Ayrıca bu algoritmaların son anlık IoU değeri de sıfır olduğundan videodaki belli bir görüntüden itibaren takibi tamamen kaybettikleri ve telafi edemedikleri anlaşılmaktadır. Bu durumda takibin koptuğu görüntülerin tespiti ve algoritmaların takibin kopması durumundaki davranışları için gerekli verilere ihtiyaç duyulmaktadır. Aşağıdaki tabloda bahsi geçen veriler sunulmuştur.

Tablo 11. 1841 görüntü üzerinde yapılan takip testi neticesinde takibin sürekliliğine ilişkin elde edilen veriler.

Algoritma	Sıfır IoU Değerli Görüntüler	Tracker Hatası Alınan Görüntüler	Telafi Edilen Tracker Hataları
BOOSTING	1493...1841	-	-
CSRT	-	-	-
KCF	20...1841	-	-
MEDIANFLOW	39...1841	185, 186, 203, 204, 253...1841	-
MIL	-	-	-
MOSSE	2, 53, 54, 58, 113...1841	2, 53, 54, 58, 113...1841	2, 53, 54, 58

* Sayısal ifadeler kaçınıcı görüntü olduğunu ifade etmektedir. Görüntü numaraları aralarına virgül konarak ayrılmıştır. Üç nokta ile aralıklar belirtilmiştir.

Takip algoritmalarının takibi kaybettikleri noktalar dikkate alındığında, takibi en hızlı şekilde kaybeden algoritmanın MOSSE algoritması olduğu görülmektedir. Takip başladıktan sonra alınan ikinci görüntüde MOSSE takibi kaybetmiştir. Ancak MOSSE algoritması söz konusu takip kaybını tespit ederek bunu telafi edebilmiştir. MOSSE algoritması sonraki görüntülerde de takibi dört kez kaybetmiş, bunlardan üçünü telafi edebilmiştir. Telafi edilemeyecek şekilde takibi 113'üncü görüntüde kaybeden MOSSE bu noktadan sonra da takibin devam etmediğine dair hata mesajlarını sürdürmüştür. Bu açıdan bakıldığında test edilen algoritmalarda takibin kaybedilmesi durumunda bunu telafi edebilen tek algoritma MOSSE olmuştur.

MOSSE algoritmasından sonra takibi en hızlı kaybeden algoritma KCF olmuştur. KCF algoritması 20'nci görüntüde takibi kaybetmiştir. Takibi kaybettikten sonra herhangi bir hata mesajı vermeyen algoritma, testin devamında sıfır IoU değeri üretmeye devam etmiştir. Bu durum KCF algoritmasının neden en düşük ortalama IoU değerine sahip olduğunu açıklamaktadır.

MOSSE ve KCF'den sonra takibi en hızlı kaybeden algoritma ise MEDIANFLOW algoritması olmuştur. MEDIANFLOW algoritması 39'uncu görüntüden itibaren takibi kaybetmiş ve telafi edememiştir. Ancak takibi kaybettiğine dair sonraki görüntülerde hata mesajları üretebilmiştir. Gözlem neticesinde anlaşılmıştır ki takibi kaybettiğine dair hatayı ürettikten sonra takibi sürdürmek üzere telafi etmeye çalışmıştır. Ancak takibin kaybedildiğini geç fark ettiğinden dolayı telafi etmek için yaptığı girişim doğru bir çıktı üretememiştir. Bu sebeple MEDIANFLOW algoritması, KCF algoritmasından sonra en düşük ortalama IoU değerini üretmiştir.

BOOSTING algoritması, takibi 1493'üncü görüntüde kaybetmiştir. Dolayısı ile sıfır değerindeki IoU değerlerini üretmeye bahsi geçen üç algoritmadan daha sonra başlamıştır. Bu durum diğerlerine göre daha yüksek bir IoU değeri üretmesini sağlamıştır. Ancak gözlem neticesinde BOOSTING algoritmasının görüntü üzerinde takip ettiği alanın genişlik ve yüksekliğini değiştirmedeği anlaşılmıştır. Bu durum yakınlaşma ve uzaklaşma durumlarında takibin kolayca kaybedilebileceğini göstermektedir. Ayrıca BOOSTING algoritması takibi kaybettiğinde herhangi bir hata üretmemiş veya telafi etmeye çalışmamıştır.

CSRT ve MIL algoritmaları ise tüm test boyunca takibi kaybetmemişlerdir. CSRT algoritması takip boyunca daha doğru bir şekilde yüksek IoU değerleri üreterek takibi devam ettirmiştir. Bu durum takip algoritmaları içerisinde CSRT'yi en yüksek doğruluk değerine sahip algoritma olarak öne çıkarmıştır. MIL algoritması ise takibin başlangıcında bazı düşük IoU değerleri üretmiştir. Bu sebeple CSRT'den sonra en yüksek doğruluğa sahip algoritma olsa da kullanımının riskli olduğu söylenebilir.

Takip algoritmalarının yalnızca doğrulukları değil aynı zamanda performansları da oldukça önemlidir. Takip algoritmalarının kullanımındaki asıl neden YOLO'nun sebep olduğu düşük FPS değerini yükseltmek olduğundan takip algoritmalarının FPS skorları da test edilmiştir. Test neticesinde üretilen FPS değerleri aşağıdaki tabloda sunulmuştur.

Tablo 12. 1841 görüntü üzerinde yapılan takip testi neticesinde elde edilen anlık ve ortalama FPS değerleri.

Algoritma	İlk Anlık FPS	Son Anlık FPS	En Düşük Anlık FPS	En Yüksek Anlık FPS	Ortalama FPS
BOOSTING	24,451	21,316	12,372	24,456	22,359
CSRT	7,446	6,022	3,933	7,446	5,829
KCF	169,154	231,478	68,21	251,933	225,205
MEDIANFLOW	71,809	59,486	16,889	125,754	70,085
MIL	5,23	5,208	4,022	5,462	4,946
MOSSE*	25+	25+	25+	25+	25+

* MOSSE algoritması video üzerinde test edilirken çok tutarsız FPS değerleri üretmiştir. Bu sebepten FPS verilerinde MOSSE algoritmalarından alınan sonuçlar 25 FPS olarak sabitlemiştir.

Test neticesinde alınan FPS verileri incelendiğinde KCF algoritmasının en yüksek FPS değerine sahip olduğu görülmektedir. KCF algoritmasından sonra en yüksek performans ile çalışan algoritma ise MEDIANFLOW olmuştur. Söz konusu algoritmalar en yüksek performansı gösterecekleri de en düşük doğruluğa sahip algoritmalar da bunlardır. Ayrıca bu

algoritmalar kısa sürede takibi kaybetmişlerdir. Dolayısı ile KCF ve MEDIANFLOW algoritmaları kısa süre kullanılmak üzere YOLO ile birlikte çalıştırıldıklarında FPS skorunda ciddi bir artışa sebep olabileceklerdir.

BOOSTING algoritması KCF ve MEDIANFLOW algoritmalarından çok daha uzun süre takibi devam ettirebilmiştir. FPS skoru ise söz konusu iki algoritmadan çok daha düşüktür. Ayrıca takip esnasında BOOSTING algoritması takip dörtgeninin yalnızca koordinatlarını değiştirmiştir. Dolayısı ile takip edilen nesnenin yaklaşması veya uzaklaşması durumlarında takibi kaybetme ihtimali artmaktadır.

MIL ve CSRT algoritmaları en düşük FPS değerlerini üreten takip algoritmaları olmuştur. MIL ve CSRT algoritmaları aynı zamanda en yüksek doğruluğa sahip algoritmalarlardır. Dolayısı ile şu ana kadar incelenen algoritmalarda genel olarak doğruluk arttıkça performansın düştüğü söylenebilir.

MOSSE algoritması video üzerinde yapılan testte çok tutarsız ve fazlasıyla yüksek FPS değerleri (4334 FPS gibi) üretmiştir. Bu sebeple MOSSE algoritmasının FPS değerleri benzer çalışmalarda alınan değerlerle uygun olacak şekilde tahmini bir değere sabitlenmiştir. Gözlem neticesinde MOSSE algoritmasının performans açısından BOOSTING algoritması gibi davrandığı ancak hataları giderebilmesi sebebiyle daha kısa süreli kullanımda BOOSTING algoritmasının yerine tercih edilebileceği sonucuna varılmıştır.

Takip algoritmaları üzerine yapılan testlerde pek çok algoritmanın ilk ve son FPS değerleri arasında düşüş olduğu gözlemlenebilmektedir. Kullanılan donanım göz önüne alındığında (Raspberry Pi 4) işlemci sıcaklığının performansı ciddi bir şekilde etkileyebileceği de aşıkardır. Bu sebeple yapılan her bir testin başlangıcında ve sonunda işlemci sıcaklığı ölçülmüştür. Aşağıdaki tabloda söz konusu sıcaklık değerleri sunulmuştur.

Tablo 13. 1841 görüntü üzerinde yapılan takip testi neticesinde elde edilen işlemci sıcaklığı verileri.

Algoritma	İlk CPU Sıcaklığı*	Son CPU Sıcaklığı*	Görüntü Başına Sıcaklıktaki Değişim
BOOSTING	72,062	81,802	0,00529
CSRT	78,393	83,75	0,00291
KCF	71,549	80,828	0,00504
MEDIANFLOW	74,984	83,263	0,00450
MIL	75,958	82,776	0,00370
MOSSE	66,705	78,880	0,00661

* Santigrat Derece (°C)

Nesne Tespiti ve Takibinin Birlikte Çalıştırılması

Nesne takibi ve tespitine ilişkin FPS ve IoU değerleri önceki bölümlerde açıklanmıştır. Elde edilen bu bulgulardan faydalanılarak video üzerinde tespit ve takip algoritmalarının birlikte çalıştırılması durumunda üretebilecekleri FPS ve IoU skorları bu bölümde sunulacak ve açıklanacaktır.

Çalışma kapsamında geliştirilen İHA'nın tespit ve takip algoritmalarının sıralı bir şekilde birbirlerini tetikleyecekleri önceki bölümlerde açıklanmıştır. Nesne tespit edildikten sonra nesne tespit modülü, nesne takip modülünü tetikleyecek ve YFT (YOLO Frame Threshold) adedince takip algoritması çalışacaktır. YFT değeri ve kullanılacak olan takip algoritması uçuş öncesinde operatör tarafından belirlenecektir.

Test edilen takip algoritmaları incelendiğinde zaman içerisinde doğruluklarında düşüş yaşandığı görülmektedir. Yani doğruluğun yüksek olması için takip algoritmasının mümkün olduğunca kısa süre çalıştırılması gerektiği söylenebilir. Ancak takip algoritmalarının alternatifi olan YOLO ise çok düşük bir FPS değerine sahiptir. Yani FPS değerinin yüksek olması için takip algoritmalarının mümkün olduğunca uzun süre çalıştırılması gerektiği söylenebilir. Dolayısı ile FPS ve doğruluk arasında bir optimizasyona ihtiyaç duyulduğu ortaya çıkmaktadır.

Söz konusu optimizasyon problemini çözmek için YOLO ve takip algoritmalarına ilişkin önceki bölümlerde sunulmuş verilerden faydalanılarak yeniden hesaplamalar yapılmıştır. Yapılan ilk hesaplamada YOLO ve takip algoritmasının birlikte çalıştırılması durumunda üretecekleri ortalama FPS değeri hesaplanmıştır. Takip algoritmasının ve YOLO'nun ürettiği ortalama FPS verileri bilindiğinden birlikte kullanıldıklarında elde edilecek FPS verisi hesaplanabilmektedir. Bu hesaplamanın yapılabilmesi için aşağıdaki formül kullanılmıştır.

$$FPS_{MEAN} = \frac{(YFT + 1) * FPS_{YOLO} * FPS_{TRACKER}}{FPS_{TRACKER} + (YFT * FPS_{YOLO})}$$

Ortalama FPS hesaplaması her bir takip algoritması için yapılmıştır. Ayrıca her bir takip algoritmasında 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 100 ve 250 YFT değerlerinde hesaplamalar tekrar edilmiştir. Alınan ortalama FPS değerleri aşağıdaki tablolarda sunulmuştur.

Tablo 14. 5 ve 30 YFT aralığında farklı takip algoritmaları ile elde edilen ortalama FPS değerleri.

Algoritma	FPS YFT=5	FPS YFT=10	FPS YFT=15	FPS YFT=20	FPS YFT=25	FPS YFT=30
BOOSTING	8,343	11,993	14,217	15,574	16,759	17,636
CSRT	4,788	5,391	5,776	5,841	5,885	5,992
KCF	11,254	19,507	27,198	34,236	40,774	47,247
MEDIANFLOW	10,194	16,500	20,987	24,990	28,105	30,762
MIL	3,974	4,329	4,539	4,577	4,666	4,791
MOSSE*	8,559	12,209	14,532	16,141	17,322	18,225

* MOSSE algoritması video üzerinde yapılan testlerde çok tutarsız sonuçlar ürettiğinden FPS değeri daha önce tahmini bir beklenen değere eşitlenmiştir. Söz konusu değer ilgili bölümde sunulmuştur.

Tablo 15. 35 ve 250 YFT aralığında farklı takip algoritmaları ile elde edilen ortalama FPS değerleri.

Algoritma	FPS YFT=35	FPS YFT=40	FPS YFT=45	FPS YFT=50	FPS YFT=100	FPS YFT=250
BOOSTING	18,187	18,714	19,281	19,767	21,590	22,819
CSRT	5,935	6,141	6,081	6,199	6,338	6,333
KCF	53,322	58,366	64,142	68,164	105,478	153,539
MEDIANFLOW	32,993	35,235	36,504	37,703	46,121	52,398
MIL	4,739	4,769	4,782	4,847	4,879	4,892
MOSSE*	18,937	19,514	19,991	20,392	22,439	23,902

* MOSSE algoritması video üzerinde yapılan testlerde çok tutarsız sonuçlar ürettiğinden FPS değeri daha önce tahmini bir beklenen değere eşitlenmiştir. Söz konusu değer ilgili bölümde sunulmuştur.

Ortalama IoU değerleri, YOLO ve takip algoritmalarının ortalama IoU değerleri bilindiğinden ortalamaları alınarak hesaplanabilir. Burada takip algoritmasının IoU değeri, YOLO'nun IoU değeri ile oranlanarak hesaplanmıştır. Bunun sebebi takip algoritmalarında yapılan testlerde ilk tespitin %100 IoU değeri ile başlamasıdır. Tespit ve takip algoritmalarının birlikte kullanılması durumunda ise ilk IoU değerinin YOLO'nun IoU değerine eşit olacağı kabul edilmiştir. Söz konusu hesaplama için aşağıdaki formül kullanılmış, alınan sonuçlar tablo 16 ve tablo 17'de sunulmuştur.

$$IoU_{MEAN} = \frac{IoU_{YOLO} + (YFT * IoU_{YOLO} * IoU_{TRACKER})}{YFT + 1}$$

Tablo 16. 5 ve 30 YFT aralığında farklı takip algoritmaları ile elde edilen IoU skorları.

Algoritma	IoU YFT=5	IoU YFT=10	IoU YFT=15	IoU YFT=20	IoU YFT=25	IoU YFT=30
BOOSTING	%53,552	%51,191	%50,043	%49,011	%49,233	%49,134
CSRT	%53,390	%49,604	%48,528	%47,595	%47,617	%47,945
KCF	%53,552	%51,191	%49,801	%49,442	%39,934	%33,493
MEDIANFLOW	%52,690	%50,956	%56,468	%48,765	%45,255	%41,376
MIL	%43,693	%34,558	%42,770	%43,100	%41,277	%40,000
MOSSE	%31,786	%29,621	%28,769	%28,323	%27,911	%27,424

* YOLO, %64,65 ortalama IoU değerine sahiptir.

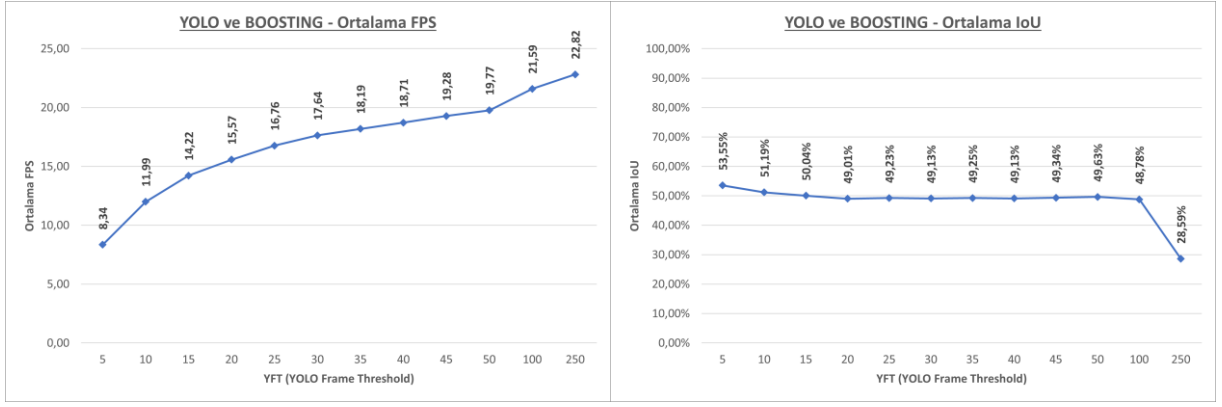
Tablo 17. 35 ve 250 YFT aralığında farklı takip algoritmaları ile elde edilen IoU skorları.

Algoritma	IoU YFT=35	IoU YFT=40	IoU YFT=45	IoU YFT=50	IoU YFT=100	IoU YFT=250
BOOSTING	%49,251	%49,134	%49,345	%49,628	%48,776	%28,590
CSRT	%47,742	%47,999	%48,586	%48,931	%50,760	%34,128
KCF	%28,841	%25,324	%22,571	%20,358	%10,280	%4,134
MEDIANFLOW	%37,183	%32,924	%29,360	%26,494	%13,378	%5,473
MIL	%40,703	%42,574	%40,744	%43,417	%40,006	%26,723
MOSSE	%27,189	%26,806	%26,387	%26,114	%25,092	%25,117

* YOLO, %64,65 ortalama IoU değerine sahiptir.

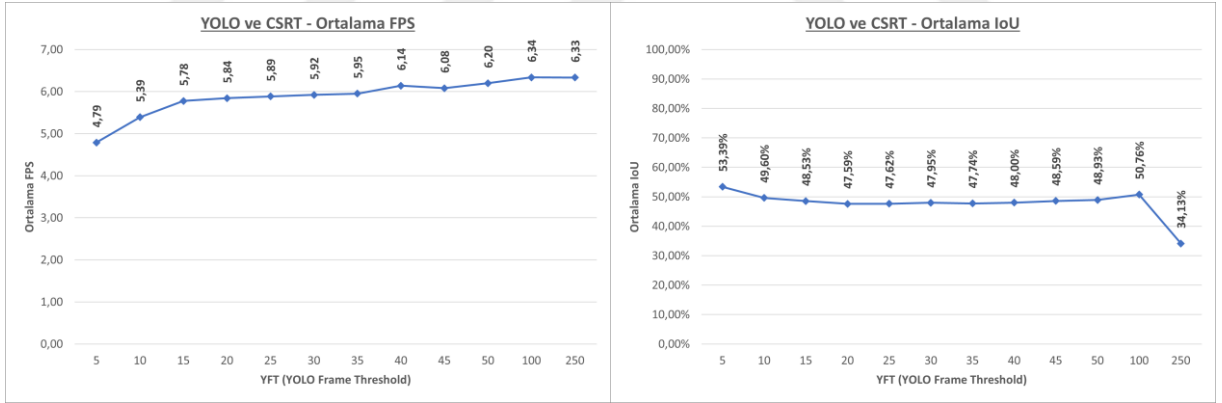
En yüksek IoU değeri, YFT değeri 15 iken MEDIANFLOW algoritmasıyla %56,47 olarak hesaplanmıştır. Bu değer yakalandığında üretilen FPS değeri ise 20,99 olarak hesaplanmıştır. En yüksek FPS değeri, YFT değeri 250 iken KCF algoritması ile 153,54 olarak hesaplanmıştır. Bu değer yakalandığında IoU değeri %4,14 olarak hesaplanmıştır. Aşağıda tüm takip algoritmalarının IoU ve FPS hesaplamalarından alınan sonuçlar grafikler ile sunulmuş ve açıklanmıştır.

BOOSTING algoritmasının en yüksek ortalama IoU değeri %53,55 olarak hesaplanmıştır. Bu IoU değeri, YFT değeri 5 iken hesaplanmıştır. BOOSTING algoritmasının en yüksek ortalama FPS değeri ise 22,82 olarak hesaplanmıştır. En yüksek FPS değeri ise YFT değeri 250 iken hesaplanmıştır. En yüksek FPS değerinin alındığı durumda IoU değerinin %28,59 olduğu hesaplanmıştır. Benzer şekilde en yüksek IoU değerinin alındığı durumda ise FPS değerinin 8,34 olduğu hesaplanmıştır. BOOSTING algoritmasının farklı YFT değerlerindeki FPS ve IoU değerleri şekil 44'te gösterilmiştir.



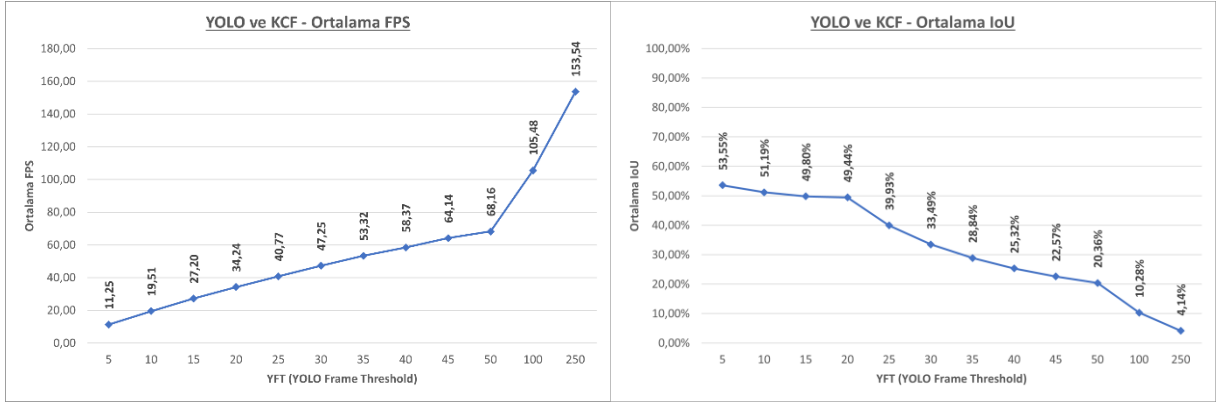
Şekil 44. BOOSTING algoritmasının YOLO ile farklı YFT değerlerinde ürettiği FPS (solda) ve IoU (sağda) değerleri.

CSRT algoritmasının en yüksek ortalama IoU değeri %53,39 olarak hesaplanmıştır. Bu IoU değeri, YFT değeri 5 iken hesaplanmıştır. CSRT algoritmasının en yüksek ortalama FPS değeri ise 6,34 olarak hesaplanmıştır. En yüksek FPS değeri ise YFT değeri 100 iken hesaplanmıştır. En yüksek FPS değerinin alındığı durumda IoU değerinin %50,76 olduğu hesaplanmıştır. Benzer şekilde en yüksek IoU değerinin alındığı durumda ise FPS değerinin 4,79 olduğu hesaplanmıştır. CSRT algoritmasının farklı YFT değerlerindeki FPS ve IoU değerleri şekil 45'te gösterilmiştir.



Şekil 45. CSRT algoritmasının YOLO ile farklı YFT değerlerinde ürettiği FPS (solda) ve IoU (sağda) değerleri.

KCF algoritmasının en yüksek ortalama IoU değeri %53,55 olarak hesaplanmıştır. Bu IoU değeri, YFT değeri 5 iken hesaplanmıştır. KCF algoritmasının en yüksek ortalama FPS değeri ise 153,54 olarak hesaplanmıştır. En yüksek FPS değeri ise YFT değeri 250 iken hesaplanmıştır. En yüksek FPS değerinin alındığı durumda IoU değerinin %4,14 olduğu hesaplanmıştır. Benzer şekilde en yüksek IoU değerinin alındığı durumda ise FPS değerinin 11,25 olduğu hesaplanmıştır. KCF algoritmasının farklı YFT değerlerindeki FPS ve IoU değerleri şekil 46'da gösterilmiştir.



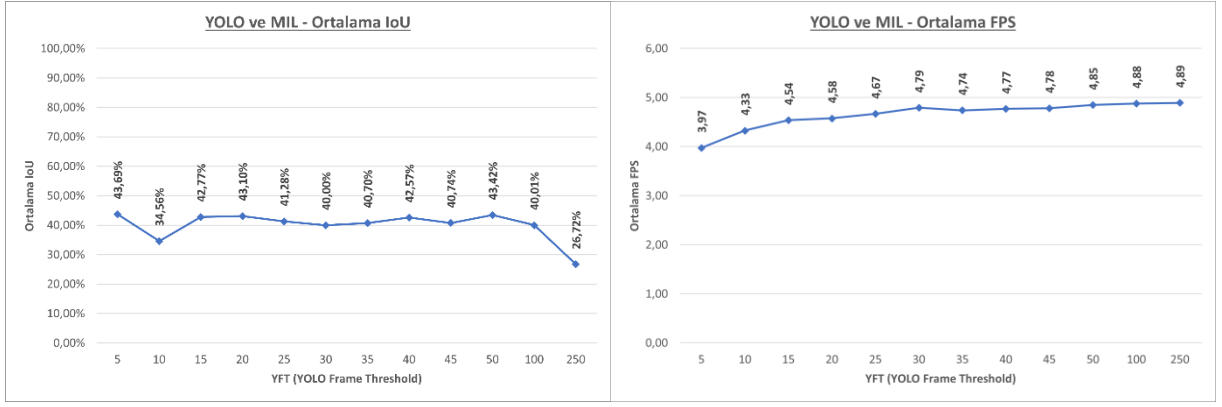
Şekil 46. KCF algoritmasının YOLO ile farklı YFT değerlerinde ürettiği FPS (solda) ve IoU (sağda) değerleri.

MEDIANFLOW algoritmasının en yüksek ortalama IoU değeri %56,47 olarak hesaplanmıştır. Bu IoU değeri, YFT değeri 15 iken hesaplanmıştır. MEDIANFLOW algoritmasının en yüksek ortalama FPS değeri ise 52,4 olarak hesaplanmıştır. En yüksek FPS değeri ise YFT değeri 250 iken hesaplanmıştır. En yüksek FPS değerinin alındığı durumda IoU değerinin %5,47 olduğu hesaplanmıştır. Benzer şekilde en yüksek IoU değerinin alındığı durumda ise FPS değerinin 20,99 olduğu hesaplanmıştır. MEDIANFLOW algoritmasının farklı YFT değerlerindeki FPS ve IoU değerleri şekil 47’de gösterilmiştir.



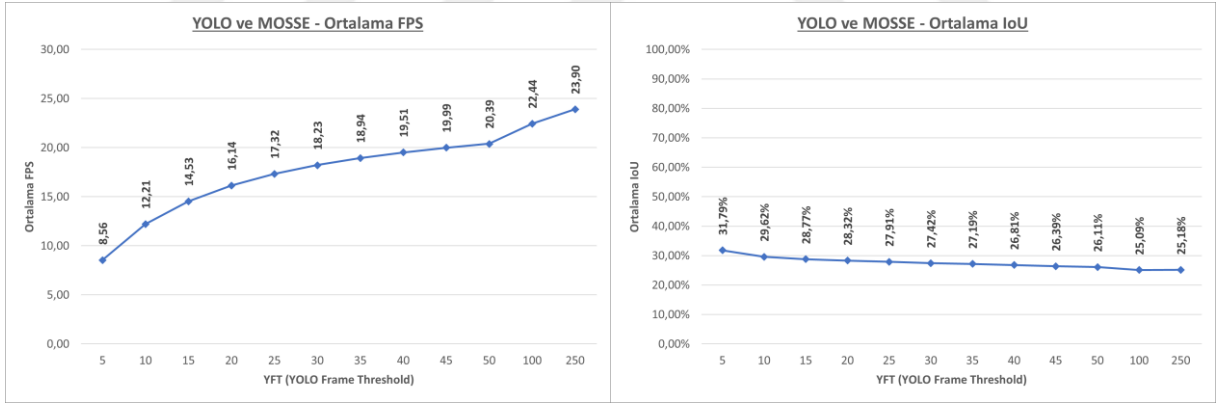
Şekil 47. MEDIANFLOW algoritmasının YOLO ile farklı YFT değerlerinde ürettiği FPS (solda) ve IoU (sağda) değerleri.

MIL algoritmasının en yüksek ortalama IoU değeri %43,69 olarak hesaplanmıştır. Bu IoU değeri, YFT değeri 5 iken hesaplanmıştır. MIL algoritmasının en yüksek ortalama FPS değeri ise 4,89 olarak hesaplanmıştır. En yüksek FPS değeri ise YFT değeri 250 iken hesaplanmıştır. En yüksek FPS değerinin alındığı durumda IoU değerinin %26,72 olduğu hesaplanmıştır. Benzer şekilde en yüksek IoU değerinin alındığı durumda ise FPS değerinin 3,97 olduğu hesaplanmıştır. MIL algoritmasının farklı YFT değerlerindeki FPS ve IoU değerleri şekil 48’de gösterilmiştir.



Şekil 48. MIL algoritmasının YOLO ile farklı YFT değerlerinde ürettiği FPS (solda) ve IoU (sağda) değerleri.

MOSSE algoritmasının en yüksek ortalama IoU değeri %31,79 olarak hesaplanmıştır. Bu IoU değeri, YFT değeri 5 iken hesaplanmıştır. MOSSE algoritmasının en yüksek ortalama FPS değeri ise 23,9 olarak hesaplanmıştır. En yüksek FPS değeri ise YFT değeri 250 iken hesaplanmıştır. En yüksek FPS değerinin alındığı durumda IoU değerinin %25,18 olduğu hesaplanmıştır. Benzer şekilde en yüksek IoU değerinin alındığı durumda ise FPS değerinin 8,56 olduğu hesaplanmıştır. MOSSE algoritmasının farklı YFT değerlerindeki FPS ve IoU değerleri şekil 49’da gösterilmiştir.



Şekil 49. MOSSE algoritmasının YOLO ile farklı YFT değerlerinde ürettiği FPS (solda) ve IoU (sağda) değerleri.

Video üzerinde yapılan testlerden alınan sonuçlar gerçek zamanlı olarak Raspberry Pi kamera modülünden alınan video akışı üzerinde test edilmemiştir. Bunun sebebi yukarıda bahsi geçen değerlerin üretilmesi için yapılan testlerin uçuş süresinden çok daha uzun sürmesidir. Uçuşlar arası uzun süre geçeceğinden ve pilotun birebir aynı uçuşu tekrar etmesi mümkün olmadığından deney ortamını sabitlemek amacıyla testler video üzerinde yapılmıştır.

SONUÇLAR VE ÖNERİLER

Bu çalışmada otonom kabiliyetlere sahip bir döner kanatlı insansız hava aracı tasarlanmış ve üretilmiştir. Geliştirilen İHA'da düşük maliyetli malzemeler kullanılmış ve görev isterlerinin karşılanması için kullanılan malzemeler iyileştirilmiştir. Yapılan iyileştirmeler için gövde mekaniğinde yapısal modifikasyonlar gerçekleştirilmiş ve bu modifikasyonlar için kullanılacak parçalar özgün bir şekilde tasarlanarak 3B yazıcı vasıtasıyla üretilmiştir. Yapısal modifikasyonlar neticesinde bir quadcopter gövdesi octoquadX8 gövdeye dönüştürülmüştür.

İHA gövdesinde kullanılan elektronik bileşenler açıklanarak İHA tasarımına ilişkin hesaplamalar yapılmış ve tasarım aşamasında yapılan hesaplamalar ile üretim neticesinde alınan sonuçlar karşılaştırılmıştır. Çalışma kapsamında geliştirilen İHA'yı eşsiz kılan özellik ise birden fazla bağımsız güç dağıtım sistemine sahip olmasıdır. Bildiğimiz kadarıyla, görece küçük denebilecek İHA'larda, kullanılan malzemelerle iki bağımsız güç dağıtım sistemine sahip başka bir İHA ile karşılaşılmamıştır. Ayrıca İHA üzerinde çalışmak üzere önerilen yazılım mimarisi, modülleri açıklanarak sunulmuştur.

Çalışma kapsamında odaklanılan asıl konu ise düşük maliyetli bir işlem birimi üzerinde tümleşik hedef tespiti ve takibinin yapılmasıdır. Bunun için gövde üzerinde bulunan Raspberry Pi 4 Model B (4GB RAM) ile nesne tespiti ve takibinin yapılması amaçlanmıştır. Düşük maliyetli bir donanım üzerinde derin öğrenme metotlarını kullanmak performans problemlerine sebep olduğundan, derin öğrenmeyle takip algoritmaları birlikte çalıştırılmıştır.

Derin öğrenmede YOLO v4 Tiny kullanılmıştır. YOLO, Det-Fly isimli veri kümesiyle eğitilmiştir. Eğitim sürecinde üretilen değerlerin nasıl değiştiği periyodik olarak gösterilmiştir. Bununla birlikte eğitilen modelin düşük maliyetli donanım üzerinde farklı çözünürlüklerde ürettiği FPS skorları test edilmiştir. Bu sayede YOLO v4 Tiny modelinin Raspberry Pi üzerinde çalıştırılması durumunda üretebileceği performans ve doğruluk değerlerine ilişkin bulgular ortaya çıkarılmıştır.

Derin öğrenme metodunda olduğu gibi farklı takip algoritmalarının FPS ve IoU hesaplamaları da Raspberry Pi üzerinde test edilmiştir. Yapılan testler ve gözlemler neticesinde takip algoritmalarının davranışları açıklanmıştır. Testlerde incelenen takip algoritmaları: CSRT, BOOSTING, KCF, MOSSE, MIL ve MEDIANFLOW olmak üzere 6 adettir.

Takip algoritmaları ile derin öğrenmenin birlikte çalıştırılması için önerilen yöntemde tespit ve takip metotlarının birbirlerini periyodik ve sıralı olarak tetiklemeleri önerilmiştir. Yapılan çalışmada söz konusu periyotların uzunlukları değiştirildiğinde sistemin performansı ve doğruluğunun nasıl değiştiği ortaya konmuştur. Bu sayede farklı takip algoritmalarının YOLO v4 Tiny ile çalışmaları durumunda her birinin dezavantaj ve avantajları ortaya konabilmiştir.

Sonraki çalışmalarda iletişim ekipmanlarının menzilini arttırmaya yönelik olarak yapılacak çalışmalar ve İHA gövdesindeki RF gürültüsünün önlenmesi için daha iyi koruyucu donanımların üretimi veya temin edilmesi uçuş testlerinde operatör tarafından yapılan takibin daha iyi bir seviyeye gelmesini sağlayacaktır.

Dört motor taşımak üzere üretilmiş quadcopter gövdesi için yapılan motor sayısının arttırılmasına yönelik gövde iyileştirmelerinin; altı motor taşımak üzere üretilmiş hexacopter gövdesine uygulanması, motor verimliliklerinde ve uçuş süresinde artışa sebep olacaktır. Ayrıca motorlar dışındaki elektronik bileşenler için yeni bir bağımsız güç kaynağının sisteme dahil edilmesi durumunda geliştirilecek sistem hem özgün olacak hem de uçuş süresinde artış görülecektir.

Nesne tespiti ve takibine ilişkin algoritmaların performanslarının arttırılması için işletim sistemi tarafından varsayılan olarak aktif olan masaüstü birimleri pasif hale getirilebilir ve masaüstü görüntülerini HDMI portuna göndermek için gerekli geliştirme işlemleri yapılabilir. Bu durumda Raspberry Pi üzerinde masaüstü çevre birimleri çalışmayacağından performansta artış görüleceği tahmin edilmektedir. Ayrıca Raspberry Pi için daha iyi soğutucu ekipman kullanılması durumunda performansın artacağı da tahmin edilmektedir.

Son olarak nesne takibinde takip edilecek ortamın arka plan gürültüsünün tespiti ile kullanılacak takip metoduna karar veren bir modelin geliştirilmesinin operatör iş yükünü azaltacağı ve daha bütüncül bir çözüm sağlayacağı öngörülmektedir.

KAYNAKLAR

- Aksoy, M. Ç., Orak A. S., Özkan, H. M., Selimoğlu, B., 2019. Drone Dataset: Amateur Unmanned Air Vehicle Detection. Mendeley Data, V4, doi: 10.17632/zcsj2g2m4c.4
- Alam, H., & Valles, D. (2021, October). Debris Object Detection Caused by Vehicle Accidents Using UAV and Deep Learning Techniques. In 2021 IEEE 12th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON) (pp. 1034-1039). IEEE.
- Al-Thani, N., Albuainain, A., Alnaimi, F., & Zorba, N. (2020, June). Drones for sheep livestock monitoring. In 2020 IEEE 20th Mediterranean Electrotechnical Conference (MELECON) (pp. 672-676).
- Anonymous., 2016. Connect ESCs and Motors. Ardupilot, <https://ardupilot.org/copter/docs/connect-escs-and-motors.html> (15.07.2022).
- Anonymous., 2020. Popular science px4 introduction. FlyDragon, <http://www.dronefromchina.com/new/px4-introduction.html> (15.07.2022).
- Arjomandi, M., Agostino, S., Mammone, M., Nelson, M., & Zhou, T. (2006). Classification of unmanned aerial vehicles. Report for Mechanical Engineering class, University of Adelaide, Adelaide, Australia, 1-48.
- Babenko, B., Yang, M. H., Belongie, S., 2009. Visual tracking with online multiple instance learning. IEEE Conference on computer vision and Pattern Recognition (pp. 983-990).
- Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M., 2020. YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv 2020, arXiv:2004.10934.
- Bolme, D. S., Beveridge, J. R., Draper, B. A., Lui, Y. M., 2010. Visual object tracking using adaptive correlation filters. IEEE computer society conference on computer vision and pattern recognition (pp. 2544-2550).
- Chen, B., & Miao, X., 2020. Distribution line pole detection and counting based on YOLO using UAV inspection line video. Journal of Electrical Engineering & Technology, 15(1), 441-448.
- Cuerno-Rejado, C., Garcia-Hernandez, L., Sanchez-Carmona, A., Carrio, A., Sanchez Lopez, J. L., & Campoy, P. (2016). Historical evolution of the unmanned aerial vehicles to the present. Dyna, 91(3), 282-288.
- Grabner, H., Grabner, M., Bischof, H., 2006. Real-time tracking via on-line boosting. In Bmvc (Vol. 1, No. 5, p. 6).
- Gupta, H., & Verma, O. P. (2022). Monitoring and surveillance of urban road traffic using low altitude drone images: a deep learning approach. Multimedia Tools and Applications, 81(14), 19683-19703.
- Goyal, S., Shagill, M. D., Kaur, A., Vohra, H., & Singh, A. (2020). A yolo based technique for early forest fire detection. Int. J. Innov. Technol. Explor. Eng.(IJITEE) Vol, 9, 1357-1362.
- Held, D., Thrun, S., Savarese, S., 2016. Learning to track at 100 fps with deep regression networks. In European conference on computer vision (pp. 749-765).

- Henriques, J. F., Caseiro, R., Martins, P., Batista, J., 2014. High-speed tracking with kernelized correlation filters. *IEEE transactions on pattern analysis and machine intelligence*, 37(3), 583-596.
- Hosseiny, B., Rastiveis, H., & Homayouni, S. (2020). An Automated Framework for Plant Detection Based on Deep Simulated Learning from Drone Imagery. *Remote Sensing*, 12(21), 3521.
- Jiang, C., Ren, H., Ye, X., Zhu, J., Zeng, H., Nan, Y., Min, S., Ren, X., Huo, H. (2022). Object detection from UAV thermal infrared images and videos using YOLO models. *International Journal of Applied Earth Observation and Geoinformation*, 112, 102912.
- Kalal, Z., Mikolajczyk, K., Matas, J., 2010. Forward-backward error: Automatic detection of tracking failures. *20th International Conference on Pattern Recognition* (pp. 2756-2759).
- Kalal, Z., Mikolajczyk, K., Matas, J., 2011. Tracking-learning-detection. *IEEE transactions on pattern analysis and machine intelligence*, 34(7), 1409-1422.
- Keane, J. F., & Carr, S. S. (2013). A brief history of early unmanned aircraft. *Johns Hopkins APL Technical Digest*, 32(3), 558-571.
- Liao, Y. H., & Juang, J. G. (2022). Real-Time UAV Trash Monitoring System. *Applied Sciences*, 12(4), 1838.
- Lukezic, A., Vojir, T., Cehovin Zajc, L., Matas, J., Kristan, M., 2017. Discriminative correlation filter with channel and spatial reliability. *IEEE conference on computer vision and pattern recognition* (pp. 6309-6318).
- Newcome, L. R. (2004). *Unmanned aviation: a brief history of unmanned aerial vehicles*. Aiaa.
- Özel M., 2019. Drone Dataset (UAV). Kaggle, <https://www.kaggle.com/datasets/dasmehdixtr/drone-dataset-uav> (21.07.2022).
- Peng, C. F., Hsieh, J. W., Leu, S. W., Chuang, C. H., 2018. Drone-based vacant parking space detection. In *2018 32nd International Conference on Advanced Information Networking and Applications Workshops (WAINA)* (pp. 618-622).
- Petso, T., Jamisola, R. S., Mpoeleng, D., & Mmereki, W. (2021, October). Individual Animal and Herd Identification Using Custom YOLO v3 and v4 with Images Taken from a UAV Camera at Different Altitudes. In *2021 IEEE 6th International Conference on Signal and Image Processing (ICSIP)* (pp. 33-39). IEEE.
- Pham, H. X., Bozcan, I., Sarabakha, A., Haddadin, S., & Kayacan, E. (2021, September). GateNet: An Efficient Deep Neural Network Architecture for Gate Perception Using Fish-Eye Camera in Autonomous Drone Racing. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 4176-4183). IEEE.
- Qin, Z., Wang, W., Dammer, K. H., Guo, L., Cao, Z., 2021. Ag-YOLO: A Real-Time Low-Cost Detector for Precise Spraying With Case Study of Palms. *Frontiers in Plant Science*, 12.
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A., 2016. You only look once: unified, real-time object detection. *IEEE conference on computer vision and pattern recognition* (pp. 779-788).
- Redmon, J., Farhadi, A., 2017. YOLO9000: better, faster, stronger. *IEEE conference on computer vision and pattern recognition* (pp. 7263-7271).

- Redmon, J., Farhadi, A., 2018. YOLOv3: An Incremental Improvement. *IEEE Trans. Pattern Anal.* 15:1125–1131.
- Sadykova, D., Pernebayaeva, D., Bagheri, M., & James, A. (2019). IN-YOLO: Real-time detection of outdoor high voltage insulators using UAV imaging. *IEEE Transactions on Power Delivery*, 35(3), 1599-1601.
- SHGM., 2020. İnsansız Hava Aracı Sistemleri Talimatı (SHT-İHA), https://web.shgm.gov.tr/documents/sivilhavacilik/files/mevzuat/sektorel/talimatlar/2020/SHT-IHA_Rev-04.pdf (13.04.2022).
- Somaldo, P., Ferdiansyah, F. A., Jati, G., Jatmiko, W., 2020. Developing smart COVID-19 social distancing surveillance drone using YOLO implemented in robot operating system simulation environment. In 2020 IEEE 8th R10 humanitarian technology conference (R10-HTC) (pp. 1-6).
- Tzelepi M., & Tefas A., 2017. Human crowd detection for drone flight safety using convolutional neural networks. In 2017 25th European Signal Processing Conference (EUSIPCO) (pp. 743-747). IEEE.
- USDOD., 2005. Department of Defence Dictionary of Military and Associated Terms, <https://apps.dtic.mil/sti/pdfs/ADA439918.pdf> (12.04.2022).
- Vohra D.S., Garg P.K., Ghosh S., 2022. Real-time vehicle detection for traffic monitoring by applying a deep learning algorithm over images acquired from satellite and drone. *International Journal of Intelligent Unmanned Systems*, (ahead-of-print).
- Walter V., Vrba M., Saska M., 2020. On training datasets for machine learning-based visual relative localization of micro-scale UAVs. *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 10674-10680, doi: 10.1109/ICRA40945.2020.9196947.
- Zheng, Y., Chen, Z., Lv, D., Li, Z., Lan, Z., & Zhao, S., 2021. Air-to-air visual detection of micro-uavs: An experimental evaluation of deep learning. *IEEE Robotics and Automation Letters*, 6(2), 1020-1027.

ÖZGEÇMİŞ

Kişisel Bilgiler	
Adı Soyadı:	Muhammed Yavuz BAYRAK
Doğum tarihi:	
Doğum Yeri:	
Uyruğu:	
Adres:	
Tel:	
E-mail:	
Eğitim	
Lise:	Aziziye Anadolu Lisesi (2013)
Lisans:	Sakarya Üniversitesi, Bilgisayar Müh. Bölümü (2018)
Yüksek lisans:	Atatürk Üniversitesi, Bilgisayar Müh. A.B.D. (2022)
Yabancı Dil Bilgisi	
İngilizce:	İyi
Projeler ve Yayınlar	
1. BAP Projesi 2020 - Uçuşa ve Nesne Taşıma Özelliklerine Sahip İnsansız Hava Aracı" ve "Akıllı İlaç Kutusu" 2. BAP Projesi 2021 - "Eğitim Amaçlı Otonom Araç Kiti" ve "Hedef Tespiti ve Takibi Yapabilen Döner Kanatlı İnsansız Hava Aracı"	