

# | YODA İHA Yazılım Birimi - 1. Hafta Raporu

*Not: Kritik Tasarım Raporu (KTR) yarışmacıların projeleri hakkında hazırladıkları detaylı bir rapordur. Ancak Uluslararası Savaşan İHA Yarışması'nda bu raporlar paylaşılmamaktadır.*

## | 1.1 Savaşan İHA Yarışması Kritik Tasarım Raporu İncelemeleri:

Otonomi, yapay zeka ve yazılım alanlarına yoğunlaşmış bir inceleme:

### | GÖK-TEK SİHA Takımı

#### | Uçuş Kontrol Kartları:

**Tercih Edilen Kart:** Projede, başlangıçta planlanan "Pixhawk 2.4.8" yerine "Pixhawk The Orange Cube" uçuş kontrol kartı tercih edilmiştir. (s. 12)

##### **Tercih Sebepleri:**

**Performans:** Yapılan karşılaştırmalarda, özellikle uçuş performansı bakımından rakiplerine göre daha üstün olduğu için seçilmiştir. (s. 15)

**Donanım Gücü:** Rakiplerine kıyasla daha fazla RAM'e (3 adet) ve daha fazla barometreye (2 adet) sahip olması gibi donanımsal üstünlükleri bulunmaktadır. (s. 15)

**Bütçe:** Proje sürecinde maddi kaynaklarda yaşanan olumlu değişimler, daha güçlü ve gelişmiş bir kart olan Pixhawk The Orange Cube'e geçilmesine olanak tanımıştır. (s. 12)

**Teknik Özellikler:** Kart, STM32H753 işlemciye sahiptir ve ağırlığı 76 gramdır. (s. 15)

#### | Otopilot Yazılımı:

**Kullanılan Yazılım:** Otopilot yazılımı olarak, Pixhawk The Orange Cube kartı ile tam uyumlu çalışan "Ardupilot" kullanılmıştır. (s. 30)

##### **İşlevi ve Entegrasyon:**

**İletişim Protokolü:** Otopilot ile görev bilgisayarı (Nvidia Jetson Xavier NX) arasındaki haberleşme, hava araçları için özel olarak geliştirilmiş olan MAVLink protokolü ile sağlanmaktadır. (s. 29)

**Yazılım Mimarisi:** Otopilot ile görev bilgisayarı arasındaki veri alışverişi, Robot Operating System (ROS) yazılımı ve bu yazılımla birlikte çalışan MAVROS paketi aracılığıyla yönetilmektedir. Bu yapı, sanal bir ağ üzerinden otopilota hedef yönelimleri gibi komutların aktarılmasını sağlar. (s. 23)

**Yer İstasyonu Uyumu:** Seçilen Ardupilot yazılımı, aynı zamanda yer istasyonu yazılımı olan "Mission Planner" ile de tam uyumlu çalışarak test, simülasyon ve anlık veri takibi gibi işlemleri kolaylaştırmaktadır. (s. 30)

#### | Nesne Tespit Algoritmaları:

**Kullanılan Algoritma:** Nesne tespiti için temel olarak "YOLOv3" (You Only Look Once) algoritması kullanılmıştır. Ayrıca, sistemin hızını artırmak amacıyla "YOLOv3-Tiny" modeli de eğitilerek test edilmiştir. (s. 20, s.23)

#### **Tercih Edilme Sebepleri ve Kullanım Amacı:**

**Hibrit Yapıdaki Rolü:** YOLOv3, tek başına sürekli takip için değil, geliştirilen hibrit sistemin bir parçası olarak kullanılır. Temel görevleri şunlardır:

1. Takip edilecek hedefi ilk karede tespit etmek. (s. 21)
2. Takip algoritması (GOTURN) hedefi kaybettiğinde yeniden tespit yaparak sistemi kurtarmak (kötü durum kurtarıcısı). (s. 21)

**Hız ve Optimizasyon:** YOLOv3'ün standart hali gömülü sistemlerde yavaş kalabildiği için C dilinde yazılmış orijinal kütüphaneleri Python üzerinden çağrılarak hız optimizasyonu yapılmıştır. Ayrıca daha hızlı bir alternatif olarak YOLOv3-Tiny modeli de geliştirilmiştir. (s. 21, 23)

**Veri Seti:** Algoritmanın eğitimi için standart veri setleri yerine, takımın kendi özel problemine yönelik olarak yaklaşık 7500 İHA görüntüsünden oluşan özgün bir veri seti oluşturulmuştur. Bu veri setinin yaklaşık 2000 görüntüsü YOLOv3'ün eğitimi için kullanılmıştır. (s. 21)

## **I Nesne Takibi:**

**Kullanılan Algoritma:** Nesne takibi için temel olarak "GOTURN" (Generic Object Tracking Using Regression Networks) algoritması kullanılmıştır. (s. 20)

#### **Tercih Edilme Sebepleri:**

**Yüksek Hız:** GOTURN, test esnasında 100 fps gibi çok yüksek hızlarda çalışabilme yeteneğine sahiptir, bu da onu gerçek zamanlı uygulamalar için ideal kılar. (s. 21)

**Çevrimdışı Eğitim:** Birçok takip algoritmasının aksine, çalışma anında (çevrimiçi) eğitim gerektirmez. Basit bir ileri beslemeli ağ kullanarak nesne hareketi ile görünüm arasındaki genel ilişkiyi öğrenir. (s. 21)

**Genelleme Yeteneği:** Eğitim setinde bulunmayan yeni ve farklı nesneleri de başarıyla takip edebilir. (s. 21)

**Geliştirilen Hibrit Yöntem (GHT):** Projenin en yenilikçi kısmı, YOLOv3 ve GOTURN'ün zayıf ve güçlü yönlerini birleştiren "Geliştirilen Hibrit Takipçi (GHT)" sistemidir. (s. 21)

**İşleyişi:** Bu sistemde GOTURN, hedefin konumunu aldıktan sonra yüksek hızda sürekli takibi gerçekleştirir. YOLOv3 ise sadece ilk tespitte ve GOTURN'ün hedefi kaybettiği anlarda devreye girerek sistemi kurtarır. (s. 21)

**Avantajları:** GHT, GOTURN'den hız, tek hedefe kilitlenme ve sürekli sonuç üretme avantajlarını alırken; YOLO'dan ise kötü durumları kurtarma ve hedefin en-boy oranını dinamik olarak güncelleme yeteneklerini alarak robust (sağlam) bir takip sistemi oluşturur. (s. 21)

## **I BOREAS Takımı**

### **I Uçuş Kontrol Kartları:**

**Tercih Edilen Kart:** Projede yapılan karşılaştırmalar sonucunda "Pixhawk Cube Orange" otonom kartı tercih edilmiştir. (s. 9)

### **Tercih Sebepleri:**

**Açık Donanım ve Yazılım:** Açık donanım ve yazılım desteği sunması. (s. 9)

**Haberleşme Yeteneği:** Çevresel birimler ve tümleşik bilgisayarlarla haberleşme sağlayabilmesi. (s. 9)

**Donanım Üstünlüğü:** Pixhawk 4 modeli ile yapılan kıyaslamada işlemci (32 bit ARM STM32H753) ve diğer sensör (Jiroskop, Manyometre) özellikleriyle daha üstün bulunmuştur. (s. 9)

**Teknik Özellikler:** Kart, 32 bit ARM STM32H753 Cortex-M7 işlemciye ve MS-5611 barometreye sahiptir. (s. 9)

## **I Otopilot Yazılımı:**

**Kullanılan Yazılım:** Otopilot yazılımı olarak "ArduPlane" (Mission Planner arayüzü ekran görüntüsünden teyit edilmiştir) kullanılmaktadır. (s. 29)

### **İşlevi ve Entegrasyon:**

**İletişim Protokolü:** Otonom uçuş kartı (Pixhawk) ile yapay zeka bilgisayarı (Xavier) arasındaki haberleşme, UART protokolü üzerinden MAVLink kullanılarak yapılmaktadır. (s. 20, 24)

**Yazılım Mimarisi:** Geliştirilen takip yazılımı, Python dilinde yazılmıştır ve tespit edilen hedefin konumuna göre MAVLink protokolü aracılığıyla otopilota yönelim komutları gönderir. (s. 20)

**Yer İstasyonu Uyumu:** Otopilot, telemetri verilerini yer istasyonunda çalışan ve açık kaynaklı olan "Mission Planner" arayüzüne gönderir. (s. 25)

## **I Nesne Tespit Algoritmaları:**

**Kullanılan Algoritma:** Rakip İHA tespiti için "YOLO-V4" modeli kullanılmasına karar verilmiştir. (s. 18)

### **Tercih Edilme Sebepleri ve Kullanım Amacı:**

**Performans Kıyaslaması:** YOLO-V4, yapılan araştırmalarda SSD ve Faster RCNN gibi diğer modellerle kıyaslanmıştır. Grafıklara göre en yüksek doğruluk değerine sahip olması ve hızının da kabul edilebilir seviyede olması sebebiyle ideal tercih olarak belirlenmiştir. (s. 17, 18)

**Çalışma Hızı:** Sistemin İHA havadayken kısıtlı imkanlarda hızlı çalışabilmesi gerektiğinden, çalışma hızı öncelikli bir parametre olarak belirlenmiştir. (s. 17)

**Veri Seti:** Sanal ortamdan ve gerçek hayattan alınan görüntüler ile 13,500 fotoğraflık bir veri seti hazırlanmıştır. (s. 372) Bu fotoğraflar, Python dilinde yazılan bir program ile etiketlenmiştir. (s. 18)

**Model Başarısı:** Çeşitli konfigürasyonlarla eğitilen modelin, test verileri üzerindeki doğruluk değeri %97,34 olarak hesaplanmıştır. (s. 18)

## **I Nesne Takibi:**

**Kullanılan Algoritma:** Takım, ayrı bir nesne takip algoritması (GOTURN gibi) kullanmak yerine, tespit tabanlı bir takip sistemi geliştirmiştir. (s. 20)

### **İşleyişi:**

**Konum Hesaplama:** Yazılım, YOLO-V4'ten gelen hedef İHA'nın görüntü içerisindeki piksel koordinatlarını alır. (s. 20)

**Açısal Hata Tespiti:** Kameranın 78 derecelik görüş açısı (FOV) kullanılarak, hedefin merkezden ne kadar uzakta olduğu açısal bir değere (hata fonksiyonu) dönüştürülür. (s. 20)

**PID Kontrolü:** Hesaplanan bu açısal hata, bir PID algoritmasına girdi olarak verilir. PID kontrolcüsü, bu hatayı sıfırlamak için gerekli olan kontrol sinyalini üretir. (s. 20)

**Yönlendirme:** PID'den çıkan değer, uçağın kontrol yüzeylerini (örneğin eleron, rudder) hareket ettiren özel bir "yönelim fonksiyonuna" aktarılır. Bu döngü sayesinde uçak, hedefi sürekli olarak kamera görüntüsünün merkezinde tutmaya çalışır. (s. 20)

## **KOUSTECH Takımı**

### **Uçuş Kontrol Kartları:**

**Tercih Edilen Kart:** Projede "Pixhawk The Cube Orange" modeli tercih edilmiştir. (s. 13)

#### **Tercih Sebepleri:**

**Açık Kaynak Yapı:** Açık kaynak kodlu ve STM32 tabanlı bir yapıya sahiptir. (s. 13)

**Sensör Çeşitliliği:** İçerisinde MPU9250 jiroskop/ivmeölçer, MS5611 barometre, LSM303D (s. 13) ivmeölçer/pusula ve L3GD20 jiroskop gibi çeşitli ve yetenekli sensörleri barındırmaktadır. (s. 13)

**Teknik Özellikler:** Kart, MPU9250, MS5611, LSM303D ve L3GD20 sensörlerini içermektedir. (s. 13)

### **Otopilot Yazılımı:**

**Kullanılan Yazılım:** Ekibin geçmiş yıllardaki tecrübesine dayanarak geliştirme sürecini hızlandırmak amacıyla otopilot yazılımı olarak "ArduPilot" tercih edilmiştir. (s. 13)

#### **İşlevi ve Entegrasyon:**

**İletişim Protokolü:** Otopilot ile haberleşmede Python'un stabil MAVLink kütüphanesi (pymavlink) kullanılmaktadır. (s. 23, 24)

**Yazılım Mimarisi:** ArduPilot yazılımı, uçuşu PID, TECS ve NAVL1 olmak üzere üç farklı kontrol yapısıyla sağlamaktadır. Açık kaynak kodlu olması, yarışma isterlerine göre koda müdahale etme avantajı sunmaktadır. (s. 13)

**Yer İstasyonu Uyumu:** Uçuş parametrelerinin ayarlanması ve takibi için açık kaynaklı olan "Mission Planner" arayüzü kullanılmaktadır. (s. 40)

### **Nesne Tespit Algoritmaları:**

**Kullanılan Algoritma:** Hava aracı tespiti için "YOLOv4" algoritması kullanılmıştır. (s. 26)

#### **Tercih Edilme Sebepleri ve Kullanım Amacı:**

**Performans:** YOLO, hızlı ve tek seferde nesne tespiti yapabilmesi nedeniyle sıklıkla tercih edilmektedir. (s. 26)

**Doğruluk:** Jetson Nano üzerinde yapılan bir çalışmada, YOLOv4'ün daha hızlı olan YOLOv4-tiny modeline göre daha yüksek güven skoruna sahip olduğu görülmüştür. Tespit işlemi sürekli olmayacağı için yüksek güven skoru, FPS'e göre daha önemli bir ölçüt olarak kabul edilmiştir. (s. 26)

**Model Başarısı:** Yapılan araştırmalara göre YOLOv4 modelinin, test edilen videolarda %75.6 ile %94.2 arasında değişen ortalama güven skorlarına ulaştığı belirtilmiştir. (s. 26)

## I Nesne Takibi:

**Kullanılan Algoritma:** Takip algoritmalarının avantaj ve dezavantajları değerlendirilerek, farklı algoritmaları birleştiren hibrit bir takip algoritması geliştirilmiştir. Bu hibrit sistem Kalman Filtresi, ORB ve SIFT algoritmalarını içerir. (s. 29)

### İşleyişi:

**Tahmin:** Hedefin sınırlayıcı kutusuna Kalman filtresi uygulanarak bir sonraki görüntüdeki tahmini yolu çıkarılır. (s. 29)

**Yakın Mesafe Takibi (0–20m):** Bu tahmini yol üzerinde bölgesel kırpma işlemi uygulanarak, 0–20 metre aralığında hızı nedeniyle ORB algoritması ile takip gerçekleştirilir. (s. 29)

**Uzak Mesafe Takibi (>20m):** Mesafe 20 metrenin üzerine çıktığında ise daha yüksek kesinliğe sahip olan SIFT algoritması ile takibe devam edilir. (s. 28, 29)

**Periyodik Yeniden Tespit:** Takip doğruluğunu korumak için her on görüntüde bir tespit işlemi (YOLOv4 ile) yeniden gerçekleştirilir ve bulunan hedef, takip algoritması için yeni referans görüntüsü olarak kullanılır.

**Tercih Edilme Sebepleri:** ORB algoritmasının çok hızlı olmasına rağmen 15–20 metreden sonra başarısız olması, SIFT'in ise daha uzak mesafelerde yüksek kesinlik sunmasına karşın yavaş kalması nedeniyle her iki algoritmanın da güçlü yönlerini farklı mesafelerde kullanmak hedeflenmiştir. Kalman filtresi ise bu iki algoritmanın çalışacağı alanı tahmin ederek performansı artırır. (s. 28, 29)

## I Sonuç ve Değerlendirme: Projeler Arası Teknolojik Yakınsama

İncelenen üç farklı takımın (GÖK-TEK SİHA, BOREAS ve KOUSTECH) kritik tasarım raporları, projelerin detaylarında özgün yaklaşımlar barındırsa da temel otonomi ve yapay zeka mimarilerinde şaşırtıcı bir teknolojik yakınsama olduğunu ortaya koymaktadır. Bağımsız olarak geliştirilen bu üç projenin de hemen hemen aynı temel bileşenleri ve algoritmaları tercih etmesi, bu teknolojilerin Savaşan İHA gibi yarışmalar için bir endüstri standardı haline geldiğinin güçlü bir göstergesidir.

Bu ortak teknolojiyi şu temel bileşenlerden oluşmaktadır:

1. **Uçuş Kontrol Kartı:** Her üç takım da otonom uçuşun beyni olarak yüksek performanslı ve yedekli sensör yapısına sahip Pixhawk Cube Orange modelini tercih etmiştir.
2. **Otopilot Yazılımı:** Uçuş kontrol kartını yöneten yazılım platformu olarak, geniş topluluk desteği ve esnekliği ile bilinen Ardupilot açık kaynak kodlu otopilot yazılımı ortak bir seçim olmuştur.
3. **Görev Bilgisayarı:** Yüksek hesaplama gücü gerektiren yapay zeka ve görüntü işleme görevleri için her üç takım da Nvidia Jetson ailesinden bir görev bilgisayarı (Jetson Nano

veya daha güçlü olan Xavier serisi) kullanmıştır.

4. **Nesne Tespit Algoritması:** Rakip İHA'ların gerçek zamanlı tespiti gibi kritik bir görev için tüm takımlar, hız ve doğruluk dengesiyle öne çıkan YOLO (You Only Look Once) algoritmasının farklı versiyonlarını (YOLOv3, YOLOv4) temel almıştır.
5. **Yer Kontrol İstasyonu:** Temel uçuş takibi ve manuel müdahaleler için standartlaşmış bir arayüz olan Mission Planner yazılımı, tüm takımların ortak kullandığı bir diğer bileşendir.

Bu durum, tesadüfi bir seçimden ziyade, söz konusu donanım ve yazılım kombinasyonunun; performans, güvenilirlik, geniş dokümantasyon imkanı ve açık kaynak ekosisteminin sunduğu geliştirme kolaylığı gibi avantajlar sayesinde bu alanda en uygun ve verimli çözüm olarak kabul edildiğini göstermektedir. Takımlar, temel teknoloji yığınına sıfırdan geliştirmek yerine, bu kanıtlanmış platform üzerine kendi özgün takip algoritmalarını (GÖK-TEK'in hibrit GHT'si gibi), görev yönetim sistemlerini ve optimizasyonlarını inşa ederek rekabette avantaj sağlamaya odaklanmaktadır.

## | 2. İHA Yazılımı Temel Kavramları

### | Yer Kontrol İstasyonu (YKİ)

Yer Kontrol İstasyonu (YKİ), İHA'nın yerdeki operatör tarafından izlendiği, yönetildiği ve görevlerinin planlandığı yazılım arayüzüdür. Operatör ile hava aracı arasında bir köprü görevi görür.

- **Amacı ve İşlevi:** Temel amacı, pilota veya görev operatörüne anlık uçuş verilerini (konum, hız, irtifa, pil durumu vb.) sunmak, otonom görevler için rotalar (waypoint'ler) planlamak ve "Eve Dön" gibi acil durum komutlarını göndermektir. Ayrıca, hava aracından gelen canlı kamera görüntüsünün izlenmesini sağlar.

### | Pixhawk Uçuş Kontrol Kartı

Pixhawk, İHA'nın temel uçuş fonksiyonlarını yöneten, adeta aracın "beyni" olarak kabul edilen merkezi donanım birimidir.

- **Amacı ve İşlevi:** Üzerindeki sensörlerden (ivmeölçer, jiroskop, barometre vb.) gelen verileri işleyerek hava aracının dengede kalmasını (stabilizasyon) sağlar. Otopilot yazılımından aldığı komutları motorlara ve servolara ileterek aracın hareketini yönetir.

### | Ardupilot ve PX4 Otopilot Yazılımı

Bu yazılımlar, Pixhawk gibi uçuş kontrol kartları üzerinde çalışan ve donanıma ne yapacağını söyleyen "işletim sistemleridir".

- **Amacı ve İşlevi:** Hava aracının nasıl uçacağını belirleyen temel mantığı ve algoritmaları (PID kontrolcüler gibi) içerirler. "Otonom Görev", "Havada Sabit Kalma (Loiter)" veya "Kalkış Noktasına Geri Dön (RTL)" gibi farklı uçuş modlarını tanımlarlar. **Ardupilot**

esnekliđi ve geniş topluluđu ile öne çıkarken, **PX4** genellikle daha standart ve ticari odaklı projelerde tercih edilir.

## | Görev Bilgisayarı – Nvidia Jetson Modelleri

Görev bilgisayarı, uçuş kontrol kartının işlem gücünün yetersiz kaldığı yapay zeka ve karmaşık görüntü işleme gibi ileri düzey görevleri yürüten güçlü bir yardımcı işlemcidir.

- **Amacı ve İşlevi:** Temel amacı, kameralardan gelen görüntüleri gerçek zamanlı olarak analiz etmektir. Örneğin, **YOLO** gibi yapay zeka modellerini çalıştırarak rakip bir İHA'yı tespit eder ve takip eder. Bu stratejik bilgiyi işledikten sonra, temel uçuşu yöneten Pixhawk'a "hedefi takip et" gibi daha basit, üst düzey komutlar gönderir. Nvidia Jetson serisi, bu tür yapay zeka hesaplamaları için optimize edilmiş güçlü GPU'ları sayesinde bu görev için endüstri standardı haline gelmiştir.