

# TUNGA SAYE

## 1. Sistem Tanımı

Yarışmanın ilk 10 dakikası Savaşan İHA görevine ayrılmıştır. Bu görevde İHA, üzerindeki Jetson cihazına bağlı kamera ve YOLOv5x nesne tanıma algoritması kullanarak rakip İHA'ları tespit etmek zorundadır. Rakip İHA, görüntüde en az %8 alan kapladığında, özel bir takip algoritması ile hedefe kilitlenme başlar ve bu kilitlenmenin otomatik olarak en az 4 saniye sürdürülmesi gereklidir.

Uçuş süresinin kalan 5 dakikası ise Kamikaze görevidir. Bu aşamada İHA'nın yerde bulunan 2 x 2 metrelük hedefleri tespit etmesi beklenir. Kamikaze görevi başarılı olursa, İHA kalan sürede tekrar savaşan İHA moduna dönerek rakip hedeflere kilitlenmeye çalışır.

İHA, uçuşa elden atılarak başlar ve hemen otomatik moda geçer. Kalkış ve inişler, kumandadaki Auto ve RTL (Return To Launch) modları ile otomatik olarak gerçekleştirilir. Uçuş sırasında hız, konum ve yönelim gibi telemetri verileri RFD 868X sistemi ile, canlı görüntüler ise Jetson'dan başlayan bir aktarım zinciri ile yer kontrol istasyonuna ulaştırılır ve Mission Planner arayüzünden takip edilir. Sistem, güvenlik amacıyla kumanda veya yer istasyonu ile bağlantının kesilmesi durumunda uçuşu otomatik olarak sonlandıracak şekilde tasarlanmıştır.

## 2. Detaylı Tasarım Özeti

### 2.1. Uçuş Kontrol Kartı: Pixhawk 2 Cube

İşlemci: FPU ile 32 bit ARM Cortex M4 çekirdekli

Ram: 168 Mhz / 256 KB

Yazılım: ArduPilot/Autopilot

Sensörler: Üçlü IMU (ivme, jireskop ve manyetometre)

sensör seti

Barometre: İki adet MS5611

Güç: Otomatik yük devretme özellikli yedek güç kaynağı

Bağlantı Portları: 14x PWM servo çıkışlı S Bus servo çıkışlı,

Analog/PWM RSSI girişi, 5 adet genel amaçlı seri

port (UART), 2 adet I2C bağlantı noktası, 2 adet CAN Bus

bağlantı noktası

Boyu ve Ağırlık: 50 x 81,5 x 15,5 mm/38g



Raspberry Pi 4 kartı ile uyumlu olan İHA'nın otomatik uçuş, kalkış ve iniş görevlerinin yapılması ve desteklenmesini sağlayan Pixhawk 2 Cube kartı seçilmiş. Üzerinde bulunan üçlü sensör seti (ivmeölçer, jiroskop ve manyetometre) ve ek barometreleri sayesinde hava aracının rüzgar ve hava basıncı gibi dış etkenlerden etkilenmesini minimuma indirip stabil bir şekilde uçuşun gerçekleştirilmesini sağlıyor. İçerisinde bulunan otopilot yazılımı olan "ArduPilot" açık kaynak kodlu olması sayesinde geliştirilmeye açık olması da seçiminde önemli rol oynadı.

## **2.2 Görüntü İşleme Bilgisayarı: NVIDIA Jetson Xavier NX Geliştirici Kiti**

GPU: NVIDIA Volta mimarisi 384 NVIDIA CUDA çekirdekli ve 48 Tensor çekirdekli

CPU: 6 çekirdekli NVIDIA Carmel ARMv8.2 64-bit CPU 6MB L2 + 4MB L3

Görüş Hızlandırıcı: 7 Yönlü VLIW Görüntü İşlemcisi

Bellek: 8GB 128 bit LPDDR4x 51,2GB/sn

Video Kodlama: 2x 4K 30fps | 6x 1080p 60fps | 14x 1080p 30fps  
(H.265)



Video Kod Çözme: 2x 4K 60 fps | 4x 4K 30 fps | 12x 1080p 60fps | 32x 1080p 30 fps (H.265)  
2x 4K 30 fps | 6x 1080p 60 fps | 16x 1080p 30fps (H.264)

Depolama: 128 GB SSD Hafıza Kartı

Bağlantı: Gigabit Ethernet, M.2 Key E (WiFi)

Bağlantı Portları: HDMI ve ekran bağlantı noktası, 4x USB 3.1, USB 2.0 Mikro-B, GPIO, I2C, 12S, SPI  
UART

Güç: 19V

Boyut ve Ağırlık: 103 x 90,5 x 34,66 mm/172 g

Kameradan alınan görüntülerin YOLOv5x algoritmasıyla GPU destekli bir şekilde işlenip hedeflerin tespitinde en önemli rolü olan, özellikle yapay zekâ uygulamaları için geliştirilen ve paralel olarak birden fazla sinir ağını çalıştırabilen “Jetson Xavier NX Geliştirme Kiti” seçilmiş.

### **2.2. RASPBERRY PI 4:**

İşlemci: 4 Çekirdekli, Cortex-A72 64-bit 1.5GHz

Ram: 8 GB LPDDR4-3200

Video ve Ses: 4K 60 fps destekli

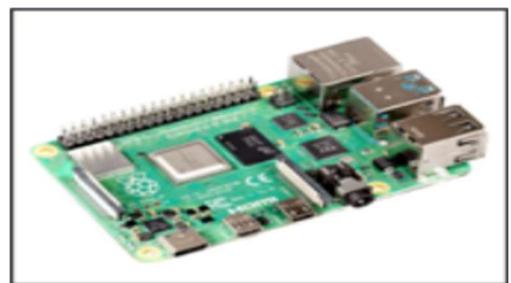
Depolama: 128 GB SSD Hafıza Kartı

Bağlantı: Gigabit Ethernet, 2.4/5.0GHz (WiFi), 5.0 Bluetooth

Bağlantı Portları: HDMI ve ekran bağlantı noktası,  
2xUSB 2.0, 2x USB 3.0

Güç: 5V, 3A (PoE, desteği)

Boyut ve Ağırlık: 103 x 90.5 x 34.66 mm/46 g



Ön Tasarım Raporu sonrasında “Pixhawk 2 Cube” ile “Jetson Xavier NX” kartlarının birbirlerine uyumlu olmadığını tespit etmişler, bu sorunun giderilmesi ve görüntü işleme bilgisayar ile uçuş kontrol kartı arasındaki haberleşmeyi sağlaması için “Raspberry Pi 4” kartı seçilmiş. Bir bilgisayara ait birçok özelliği sahip olup dahili Wi-Fi desteği, Python dil desteği gibi özellikleriyle muadillerine göre öne çıkmaktadır.

### 3. Otonom Sistemler

#### 3.1 Otonom Kilitlenme:

Takım otonom kilitleme görevinin temelde 3 aşamada gerçekleştirilmesini kararlaştırmış. Bunların ilki İHA'nın önüne sabitlenmiş olan kamera görüntülerini Jetson Kartı yardımıyla işlenerek rakip İHA'ların otonom tespitinin yapılması, ikincisi otonom takibe başlanması ve en son olarak 4 saniye boyunca takibi gerçekleşen başarılı vuruş paketin yarışma sunucusuna gönderilmesi.

##### 3.1.1 Otonom Tespit

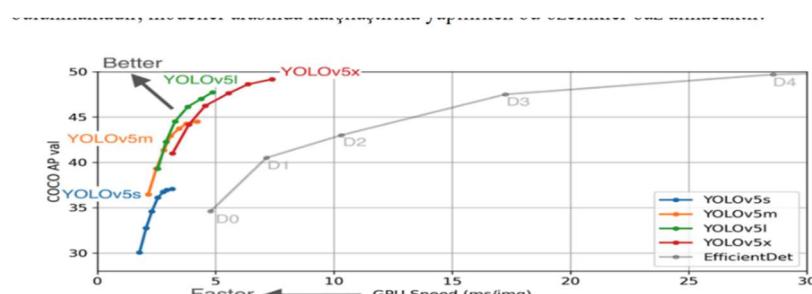
Takım, İHA üzerindeki Jetson Xavier NX bilgisayarında uçuş sırasında kameradan gelen görüntüleri işleyerek rakip İHA'ları tespit etmek için görüntü işleme algoritmaları araştırmış. Görüntü işleme için hızlı ve yaygın kullanılan OpenCV kütüphanesi etmişler ve C++ kullanmışlardır. Tespit doğruluğu ve işlem hızı (FPS) en önemli iki kriter olarak belirlenmiş. Farklı algoritmalar araştırmak yerine, YOLO (You Only Look Once) algoritmasının çeşitli versiyonlarını kullanmışlardır.

YOLO Algoritması gerçek zamanlı nesne tespiti yapabilen, GPU ile çalışabilen derin öğrenme tabanlı en gelişmiş algoritmaların biridir, en önemli özelliklerinde biri de görüntünden alınan her kareyi tek seferde işleyerek ve herhangi bir vakit kaybına sebep olmadan nesne tespiti gerçekleştirebilmesidir. YOLO algoritmasının farklı versiyonları arasında YOLOv5 geçmiş versiyonlarına göre FPS değeri konusunda çok az bir fark ile geride kalmış olmasına rağmen çok daha yüksek doğrulukta tespit gerçekleştirebiliyor. Bu FPS değeri yarışma için yeterli olduğundan göz ardı edilerek tespit doğruluk oranı daha yüksek olan YOLOv5 kullanılmış. YOLOv5 eski versiyonlarında mimari yapı olarak Darknet53 (YOLOv3) ve CSPdarknet53 (YOLOv4) kullanırken bu versiyonda CSPdarknet53 yapısının yanında Focus yapısını da kullanmaktadır.

#### KARŞILAŞTIRMA:

- YOLOv3 0,73 puan ile hassasiyet konusunda 0,69 puanı olan YOLOv4 ve 0,707 puana sahip YOLOv5 modellerini geride bırakmaktadır.
- YOLOv5 geri çağırma konusunda 0,611 puan ile 0,41 puanı olan YOLOv3 ve 0,57 puanı olan YOLOv4 modellerine üstünlük sağlamaktadır.
- YOLOv5 hassasiyet ve geri çağrıma değerlerin oranı ile elde edilen F1 puanı konusunda 0,655 puan ile 0,53 puanı olan YOLOv3 ve 0,63 puana sahip YOLOv4 modellerine üstünlük sağlamaktadır.
- YOLOv5 ortalama hassasiyet (mAP) değeri konusunda 0,633 puan ile 0,46 puanı olan YOLOv3 ve 0,607 puana sahip YOLOv4 modelleri geride bırakmaktadır.
- Son olarak ise FPS değer konusunda YOLOv3 63,7 FPS ile YOLOv4 (59) ve YOLOv5 (58,82) modellerine üstünlük sağlamaktadır.

YOLOv5 versiyonun kendi içinde farklı modelleri vardır, bu modeller arasında temelde ortalama hassasiyet (mAP), gecikme süreleri, FPS değer farkları ve boyut farkları bulunmaktadır, modeller arasında karşılaştırma yapılırken bu özellikler baz alınarak yapılacak.



İlk başta modellerin ortalama hassasiyet değerlerini inceleyince YOLOv5x modellerin diğerlerine göre daha yüksek ortalama hassasiyet değerine sahip ancak gecikme süresi konusunda bu modelin diğer modellerin gerisinde kalıyor ancak YOLOv5x model diğerlerine göre çok daha yüksek FPS değerlerine sahiptir. Son olarak modellerin boyutları karşılaştırıldığında YOLOv5x modeli diğer modellere kıyasla hafızada daha fazla alan kaplamaktadır.

## **SONUÇ:**

	<b>YOLOv5 s</b>	<b>YOLOv5m</b>	<b>YOLOv5l</b>	<b>YOLOv5x</b>
<b>Ortalama Hassasiyet (mAP)</b>	37,2	44,5	48,2	50,4
<b>Gecikme Süreleri (ms)</b>	2,0	2,7	3,8	6,1
<b>FPS Değeri</b>	17,0	51,3	115,4	218,8
<b>Boyut (MB)</b>	14	41	90	168

Ortalama hassasiyet ve FPS değeri kıyaslamasında YOLOv5x modelin daha verimli ve iyi sonuçlar vermesi ve yarışma isterlerini karşılayabilmesi bu modelin gecikme süresinin fazla olmasının göz ardı edilmesini sağlamış, sonuç olarak nesne tespiti için YOLOv5x modelin kullanılması uygun görülmüş.

### **3.1.2. Otonom Takip:**

Otonom kilitlenme sisteminin son aşaması otonom hedef takibidir. Otonom hedef takibi yapacak yazılımın en önemli özellikleri çalışma hızı ve hedefleri doğru şekilde takip edebilmesidir. Bu konuda yapılan araştırmalar sonucunda KCF, GOTURN, CSRT, MOSSE, siamMaske-E gibi birçok farklı algoritma var ama takım bu algoritmaların yetersiz olduğunu düşünüp takip algoritmasının kendileri tarafından geliştirilmesine karar vermiş.

İHA görevde yarışma günü, belirli bir irtifada rastgele dolaşarak başlayacak. Algoritma hedef İHA'ları tespit edecek; tespit edilince takip moduna geçilecek. Kameradan gelen görüntüler koordinat sistemine dönüştürülp hedefin etrafına bir “kilit” kutusu konacak ve merkez noktası belirlenecek. Görüntüdeki piksel farkları kullanılarak bu merkezin geçmiş iki konumu ile şu anki konumu arasındaki fark hesaplanacak; böylece hedefin bir sonraki hareketi tahmin edilecek. Sadece anlık konuma bakmak yerine hareket tahmini yapıldığı için takip daha güvenilir olacak. Ayrıca pozisyon değişiminden ivme, oran ve yön gibi bilgiler de çıkarılıp takipte kullanılacak.

Takibin gerçekleştiği ana yöntem İHA'nın merkez noktası ile görüntü alanın merkez noktası arasındaki fark hesaplanarak iki merkez noktanın eşitlenmesi için uçuş kontrol kartına yunuslama, yuvarlanma ve yalpalanma açıları hesaplanarak gönderilecektir. Takip işlemi sırasında hedefin kaybolduğu durumdan en son göründüğü konuma doğru 3 saniye boyunca hareket devam edip hedef bulunmadığı takdirde takip sonlandırılacak. Hedefe uzaklık ise kilitlenme kutucuğun yatay uzunluğunun görüntü erkanına oranı hesaplanarak İHA'nın yavaşlanması hızlanması için uçuş kontrol kartına komut gönderilecek. Hedefe olan mesafe, hedefin etrafına konulan kilit kutucuğunun yatay uzunluğunun ekran genişliğine oranı hesaplanarak bulunacak bu orana göre İHA'nın hızı azaltılıp artırılmak üzere uçuş kontrol kartına komut gönderilecek.

Katılımcılardan beklenen, hedef İHA'nın görüntüsünün kamerada en az %5'luk bir alanı kaplayacak şekilde takip edilmesidir. Eğer hedef İHA aniden uzaklaşıp kilitlenme süresi (4 saniye) dolmadan görüntüde %5'ten az görünürse, takip yine de devam edecek fakat hedef görüntüsü kameranın görüş alanının en az %8'i, en fazla %10'u olacak şekilde hedefe göre hız ve konum ayarlamaları yapılacak. 4 saniye boyunca hedef İHA'ya kilitlendikten sonra İHA'nın takibi bırakılacak ve tekrar rastgele olarak yeni hedef aramasına başlanılacaktır. Aynı İHA'ya tekrar kilitlenmemesi için ise bir sonraki kilitlenme gerçekleşene kadar kilitlenen İHA'nın takım numarası kaydedilecek.

### **3.2 Kamikaze Görev:**

Savaşan İHA yarışmasında ilk kez bu sene girmiş olan kamikaze görevinde amaç, yerdeki 2x2 metre boyutuna sahip bir kare kodu okuyabilmek. Bu görevin temelde iki zorluğu bulunmaktadır. Bunlardan ilki İHA'nın dalış sırasında yere çakılmaması ikincisi ise kare kodun okunabilmesi.

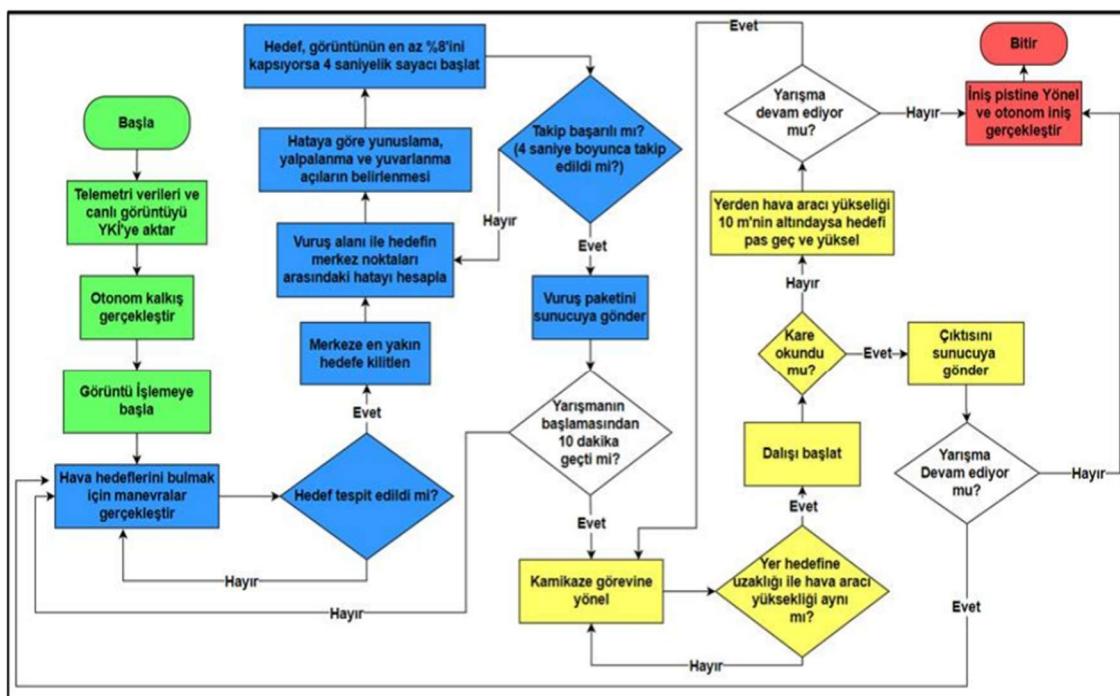
Bu zorluklarla başa çıkılabilmesi için öncelikle İHA'nın irtifa verilerinin en fazla birkaç cm'lik bir sapma ile hesaplanması gerekmektedir. Bunun için hava aracı üzerinde bulunan 1 cm'lik sapma payına sahip RTK GNSS modülün yerden yükseklik verileri kullanılmış, böylece İHA'nın dalış sırasında irtifası 40 metrenin altına düşüğü vakit kare kodunu pas geçip geri yükselmesi için komut verilecek. 40 metrenin sınır olarak seçilmesindeki sebep ise uçuş sırasında 30 metrenin altına inmek yasak olduğundan dolayı 40 metrede geri yükselme komutu alan İHA 30 metrelük sınırı ihlal etmeden rahat bir şekilde yükselp hedefi pas geçebilir. Bu sert manevra sırasında kanatların zarar görmeyeceğinden emin olmak için kanat dayanım testi gerçekleştirilmiş. Dalış sırasında yer hedefinin etrafına 45° açıları yerleştirilmiş plakaların kare kodu kapatmaması için dalış açısı 45° olarak belirlenmiş. Dalışın başlatıldığı noktası, İHA irtifası ile hedefe x ekseni üzerindeki yatay mesafenin eşit olduğu an olarak seçildi böylece plaka engellerine takılmadan kare kodun okunabileceği en uzun dalış mesafesi ve süresi sağlanmış olur. Karel kodun okunması için "OpenCV" kütüphanesinin kullanılması uygun görülmüş, böylece farklı bir yazılım kullanarak görüntü işleme kartına ek olarak binecek yükün önüne geçirilmiş.

### 3.3 Görev Algoritması:

İHA otonom bir şekilde kalkışını gerçekleştirdikten sonra güvenli bir yüksekliği ulaşana kadar yükselmeye devam edecek. Ardından yarışma alanı koordinatlarını kullanarak sınırlarının en az 20 metre öncesi kırmızı alan olarak belirleniyor ve bu alana girildiğinde İHA'nın manevra yaparak geri dönmesi için komut veriliyor. İlk başta yarışma sırasında hedeflere yönelmesi için iki yol düşünülmüş. İlk strateji rakip İHA'ların sunucuya gönderdiği telemetri içinden konum bilgilerini okuyup, İHA'ların yoğunlaştığı bölgeye yönelip oradaki hedeflere kilitlenmeye çalışmak. İkinci strateji kamerada belirli süre boyunca rakip bulunamazsa rastgele manevralar (arama manevraları) yaparak çevrede hedef aramak ve bulunca kilitlenmek. Birinci yöntem rakip İHA'ların sunucuya göndereceği verilerin doğruluğun belirsiz olması İHA'ların verilerini sunucuya gönderilmesi ve diğer YKI'lere dağıtılması, oradan algoritma tarafından hesaplanıp ve ona göre hedeflere yönelmesi gibi birçok aşamalardan geçmesi sırasında yaşanacak gecikmelerden dolayı sıkıntılı. İkinci yöntem daha mantıklı.

Kamera görüntü alanına girilen rakip İHA'lar YOLOv5x algoritması sayesinde fark edilip kare bir kutucuk içine alınması ve görüntü alanının (0,0) noktası olan merkez konumuna en yakın İHA hedef olarak seçiliyor ve hemen ardından takip algoritması çalıştırılıyor.

Vuruşun başarılı olduğundan emin olduktan sonra vuruş paketi sunucuya gönderilip diğer hedeflere yönelmesi komutu veriliyor. Yarışmanın ilk 10 dakikası hava hedeflerine yönelikken son 5 dakika boyunca kamikaze görevi için ayrılacak kamikaze görevi başarılı olmasa halinde ise tekrar savaşan İHA görevine yönelmesi komutu otonom şekilde başlatacak. Yarışma süresi sonlandığında ise kalkış yapıldığı yere otonom olarak yönelip iniş gerçekleştirilecek.



## 4.Kullanıcı Arayüz Tasarımı



Mission Planner Arayüzü

Kullanıcı arayüz tasarımı olarak hem açık kaynak kodlu olması hem de Pixhawk uçuş kontrol kartı ile uyumlu olması sebebi ile Mission Planner programı kullanılmış.

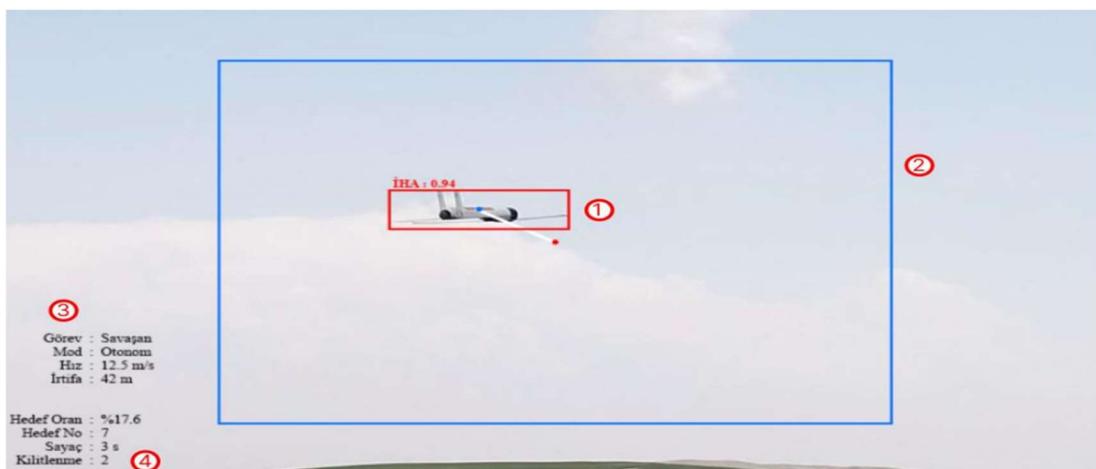
### 4.1. Mission Planner'ın Avantajları ve Kullanımı

- 1- Mission Planner gelişmiş bir simülasyon sistemine sahiptir. Simülasyon sayesinde gerçek uçuş yapılmadan, gerçeğe yakın seviyede bir uçuş simüle edilir ve problemler için erkenden önlem alınır.
- 2- Mission Planner açık kaynaklı bir program olduğundan kaynak kodunun herkes tarafından görülebilir olması güvenlik açılarının hızlıca tespit edilip düzeltilemesine ve yazılımın daha esnek bir şekilde geliştirilmesine olanak tanır. Ayrıca, geniş bir geliştirici topluluğu tarafından desteklenmesi sayesinde sürekli güncellenir ve geliştirilir.
- 3- İHA tarafından toplanan anlık veriler (hız, yükseklik, yalpalanma açısı, yuvarlanma açısı, yunuslama açısı, konum vb.) arayüzde gösterilir. Uçağa anlık komutlar verilebilir.
- 4- Google Haritalar, Yandex Haritalar, Bing Haritalar gibi çeşitli harita servisleri desteklediğinden dolayı, olası bir harita sorununda diğer haritalar kullanılabilir.
- 5- Mission Planner uçuş sırasında tuttuğu kapsamlı analizlerini, log dosyalarını ve araca ait alt sistemlerin parametrelerini uçuş sonrasında kullanıcı arayüzü ile göstermektedir. Mission Planner programının ilk arayüzü 4 temel bölgeden oluşmaktadır.



1. Bölge: Menülerin bulunduğu alan. Buradan Uçuş Verileri, Uçuş Planı, Kurulum, Ayarlar, Simülasyon ve Yardım gibi pencerelere geçilir.
2. Bölge: Bu bölgede İHA hakkında hava hızı, irtifa, dönüş hızı gibi çeşitli parametreler bulunuyor.
3. Bölge: Bu bölgede İHA'ya çeşitli komutlar verebilir, İHA'nın durumu hakkında bilgiler yer alır.
4. Bölge: Bu bölgede İHA'ya çeşitli komutlar verebilir, İHA'nın durumu hakkında bilgiler yer alır.

## 6.2 Görüntü İşleme Arayüzü:



1. Bölüm: Kilitlenme dörtgenini gösteriyor.
2. Bölüm: Hedef vuruş alanı gösteriyor.
3. Bölüm: İHA hakkında bilgileri içeriyor.
4. Bölüm: Hedef ve görev hakkında bilgiler içeriyor.