

# **ELEKTRİK – ELEKTRONİK MÜHENDİSLİĞİ**

## **BİTİRME ÇALIŞMASI**

**İnsansız Hava Araçlarında Otonom Nesne Tespiti ve Hizalanmaya  
Yönelik Takip**

**DANIŞMAN  
Doç. Dr. Gökçen ÇETİNEL**

**HAZIRLAYAN  
Ömer Faruk USTA**

**HAZİRAN 2024**

T.C.  
SAKARYA ÜNİVERSİTESİ  
MÜHENDİSLİK FAKÜLTESİ

**İnsansız Hava Araçlarında Otonom Nesne Tespiti ve Hizalanmaya  
Yönelik Takip**

**BİTİRME ÇALIŞMASI**

**Ömer Faruk USTA**

**G190100007**

**DANIŞMAN**

**Doç. Dr. Gökçen ÇETİNEL**

**Bu çalışma .. / .. /20... tarihinde aşağıdaki jüri tarafından Oybirliği ile kabul edilmiştir.**

.....  
**Jüri Başkanı**

.....  
**Üye**

.....  
**Üye**

## **TEŞEKKÜR**

Bu çalışmanın gerçekleştirilmesi sırasında bilgi birikimi ve desteğini esirgemeyen Sayın Doç. Dr. Gökçen ÇETİNEL'e teşekkürlerimi sunarım.

# İÇİNDEKİLER

TEŞEKKÜR.....	i
BEYAN.....	iv
ÖZET.....	v
SİMGE VE KISALTMALAR LİSTESİ.....	vi
ŞEKİLLER LİSTESİ.....	vii
TABLolar LİSTESİ.....	viii
DENKLEM LİSTESİ.....	viii
<b>BÖLÜM 1. GİRİŞ .....</b>	<b>1</b>
1.1. İnsansız Hava Aracı .....	1
1.1.1. İnsansız Hava Araçlarının Askeri Alanda Kullanılması.....	2
1.2. Yapay Zekâ .....	3
1.2.1. Makine Öğrenmesi.....	3
1.2.2. Derin Öğrenme .....	4
1.3. Görüntü İşleme .....	4
1.3.1. Görüntü İşleme Teknolojisinin İnsansız Hava Araçlarında Kullanımı.....	5
1.4. Otonom Sistemler.....	5
1.5. Problemin Tanımı.....	7
1.6. Çalışmanın Amacı.....	7
1.7. Çalışmanın Kapsamı .....	8
1.8. Literatür Özeti.....	9
1.9. Çalışmanın Gerçekçi Kısıtlar Açısından Analizi.....	10
<b>BÖLÜM 2. MATEMATİKSEL YÖNTEM ve TASARIM .....</b>	<b>12</b>
2.1. Yöntem Hakkında Genel Bilgi.....	12
2.1.1. Nesne Tespiti .....	12
2.1.1.1. Tekli Vuruş Dedektörü .....	13
2.1.1.2. Bölgesel Bazlı Yapay Sinir Ağları .....	13
2.1.1.3. Hızlı Bölgesel Bazlı Yapay Sinir Ağları.....	13
2.1.1.4. Sadece Bir Kez Bak Algoritması.....	14
2.1.1.5. Algoritmaların Kıyası .....	14
2.1.2. Nesne Takibi .....	18
2.1.2.1. Dönüşe Git Algoritması (GOTURN) .....	19
2.1.2.2. Derin Basit Çevrimiçi ve Gerçek Zamanlı Takip Algoritması (DeepSort).....	19
2.1.3. İstatistiksel Öğrenme .....	20
2.1.3.1. Bayes Teoremi .....	21

2.1.3.2. Parçacık Filtresi .....	22
2.1.4. Kalman Filtresi .....	22
2.2. Tasarım (Yöntemin Problemin Çözümünde Kullanımı).....	25
2.2.1. Yapay Zekâ Eğitimi ve Sistemin Tasarlanması.....	25
2.2.2. Sistem Tespit Mimarisinin Geliştirilmesi.....	28
2.2.3. Sistem Takip Mimarisinin Geliştirilmesi.....	29
<b>BÖLÜM 3. UYGULAMA ÇALIŞMALARI.....</b>	<b>31</b>
3.1. Uygulamada Kullanılan Araç ve Gereçler.....	31
3.2. Uygulamanın Gerçekleştirilme Aşamaları.....	32
3.3. Uygulama Sonuçları ve Yorumlanması .....	39
<b>BÖLÜM 4. SONUÇLAR VE ÖNERİLER .....</b>	<b>41</b>
4.1. Sonuçlar .....	41
4.2. Öneriler .....	42
4.3. Sonuçların Sağlık, Çevre ve Güvenlik Açısından Analizi .....	42
<b>BÖLÜM 5. KAYNAKLAR .....</b>	<b>43</b>
<b>BÖLÜM 6. ÖZ GEÇMİŞ.....</b>	<b>44</b>

## BEYAN

Bitirme çalışması içindeki tüm verilerin akademik kurallar çerçevesinde tarafımdan elde edildiğini, görsel ve yazılı tüm bilgi ve sonuçların akademik ve etik kurallara uygun şekilde sunulduğunu, kullanılan verilerde herhangi bir tahrifat yapılmadığını, başkalarının eserlerinden yararlanılması durumunda bilimsel normlara uygun olarak atıfta bulunulduğunu, çalışmada yer alan verilerin bu üniversite veya başka bir üniversitede herhangi bir tez çalışmasında kullanılmadığını beyan ederim.

İmza :

Ömer Faruk USTA

..... / ..... / 2024

## ÖZET

İnsansız Hava Araçları günümüzde oldukça yaygın faaliyet göstermekte olan sistematik yapılardır. Özellik, askeri alanda her geçen gün önemi artan bu sistematik yapıların, zamanla otonomi özelliğinin artırılması ve savaş kabiliyetlerinin yükseltilmesi, teknoloji geliştiren ülkeler özelinde gereklilik haline gelmiştir.

Bu sistemler, genellikle Hava-Kara muharebe senaryoları üzerine kurgulanmış olsa da son zamanlarda insansız savaş uçağı kapsamında çalışmalar gün yüzüne çıkmış ve bu çalışmaların değerleri artmıştır. Yüksek teknoloji geliştiren ülkeler tarafından, askeri yatırımlar bu yönde çoğalım göstermektedir.

Çalışma boyunca, bu sistemlerin sadece Hava-Kara özelinde kalmadan, askeri teknik için büyük önem arz eden Hava-Hava muharebe boyutuna taşınması ve bunu tam otonomiye sahip şekilde gerçekleştirmesi için araştırma yürütülmüş, sistematik incelemeler yapılarak farklı bir bakış açısı ortaya konmuştur. İnsansız Hava Aracı'nın, rakip veya düşman denilebilecek başka bir İnsansız Hava Aracına karşı, saldırı, kaçış ve hareket manevralarını kapsayan, it dalaşı olarak adlandırılabilir görev mekanizması üzerine çalışılmıştır.

Hava-Hava muharebe gereğı, saldırı öncesinde hareketli olan başka bir nesnenin tanınması, tanınan nesneye gerekli yaklaşım algoritmasının gerçekleşmesi, takip işlemi yapıldıktan sonra hizalanma yöntemiyle sistemin saldırıya hazır hale getirilmesi, çalışmanın temelini oluşturmaktadır.

Bu çalışmada, çalışmanın amaçlarından, kapsamlarından ve literatür taraması sonucu elde edilen benzer çalışma faaliyetlerinden söz edilmiş, proje boyunca ne yapılmak istendiğı detaylarıyla ortaya konulmuştur.

Çalışmanın genelinde, kullanılabilir potansiyeli bulunan tüm algoritmalar ve yazılımlar incelenerek sistematik yapı oluşturulmuş, sistematik yapıyı kapsayan temel kod yapısı ve matematiksel kapsam üzerinde durulmuştur. Uygulamanın işleyişi, uygulama adımları, kıstaslar ve bunun dahilindeki sonuçlar detaylı olarak verilmiştir.

## SİMGE VE KISALTMALAR LİSTESİ

<b>DÖ</b>	: Derin Öğrenme
<b>İHA</b>	: İnsansız Hava Aracı
<b>KF</b>	: Kalman Filtresi
<b>MÖ</b>	: Makine Öğrenmesi
<b>SİHA</b>	: Silahlı İnsansız Hava Aracı
<b>YZ</b>	: Yapay Zekâ



## ŞEKİLLER LİSTESİ

Şekil 1. 1 Bayraktar Akıncı İnsansız Hava Aracı .....	2
Şekil 1. 2 YZ, MÖ ve DÖ kapsama grafiği.....	3
Şekil 1. 3 Yapay Sinir Ağları .....	4
Şekil 2. 1 Sadece Bir Kez Bak Model Kıyaslama Tablosu.....	15
Şekil 2. 2 Algoritma Çalışma Prensibi .....	16
Şekil 2. 3 Yolo Algoritması Model Yapısı.....	17
Şekil 2. 4 Sadece Bir Kez Bak Algoritma Modelleri Kıyas Tablosu.....	17
Şekil 2. 5 Orta (Medium) ve Küçük (Small) Paketlerinin Kıyaslanması.....	18
Şekil 2. 6 Kalman Filtresi Modellemesi.....	23
Şekil 2. 7 Kalman Filtresi Gerçek Zamanlı Uygulaması .....	25
Şekil 2. 8 Veri Seti Etiketleme İşlemi .....	26
Şekil 2. 9 Doğrulama Setinde Ölçülen Modelin Doğruluk Grafiği .....	27
Şekil 2. 10 Sistem Mimarisi Akış Diyagramı .....	29
Şekil 2. 11 Nesne Tespit Yazılımı ve Hedef Çerçevesi Eklentisi .....	29
Şekil 3. 1 Aşama 1 - Hava Aracı Tespiti.....	33
Şekil 3. 2 Aşama 2 – Hedef Kilitlenme Çerçevesi.....	33
Şekil 3. 3 Aşama 3 – Hedef Kilitlenme Vektörü .....	34
Şekil 3. 4 Aşama 4 – Hedef – Hedef Vuruş Alanı Kontrolü.....	35
Şekil 3. 5 Aşama 5 - Rakip Koordinat ve Mesafe Ölçümünün Kontrolü .....	36
Şekil 3. 6 Kalman Filtresinin Uygulamaya Eklenmesi .....	36
Şekil 3. 7 Kalman Başlangıç Koşulu.....	37
Şekil 3. 8 Kalman Filtresi Hata Kontrol Grafiği .....	38
Şekil 3. 9 Hata Kontrol Paneli.....	39
Şekil 3. 10 Güneşli Hava Test Uygulama Sonucu .....	40
Şekil 3. 11 Bulutlu Hava Test Uygulama Sonucu.....	40
Şekil 3. 12 Yağmurlu Hava Uygulama Sonucu .....	41

## **TABLolar LİSTESİ**

Tablo 2. 2 İstatistiksel Öğrenme ve Makine Öğrenmesi.....	21
---	----

## **DENKLEM LİSTESİ**

Bayes Teoremi	Denklem 2. 1.....	21
Kalman Filtresi 1. Tahmin Adımı	Denklem 2. 2 .....	23
Kalman Filtresi 2. Tahmin Adımı	Denklem 2. 3 .....	23
Kalman Filtresi Güncelleme Adımı 1	Denklem 2. 4.....	24
Kalman Filtresi Güncelleme Adımı 2	Denklem 2. 5.....	24
Kalman Filtresi Güncelleme Adımı 3	Denklem 2. 6 .....	24
Öklid Uzaklığı	Denklem 3.1.....	36
Tahmin-Gerçek Mesafeler Üzerine Öklid Uygulanması	Denklem 3.2.....	39

## BÖLÜM 1. GİRİŞ

İnsansız Hava Araçları, faaliyet alanları zamanla gelişmekte ve birçok farklı sektörde adından söz ettirmektedir. Zamanla gelişen ve birçok alanda aktif olarak kullanılan bu sistemlerde, etkinlik bakımından otonomi çok önemli bir yere sahiptir. Bu bağlamda, özellikle savaş üzerine kurgulanmış insansız hava araçları özelinde, yüksek otonomi, zaruriyet durumundadır.

Bu bölüm altında, çalışma boyunca yapılacak araştırma ve geliştirilecek olan sistemlerin kapsayacağı tüm birimler incelenecektir.

### 1.1.İnsansız Hava Aracı

Günümüzde teknolojinin hızla ilerlemesiyle birlikte, insansız hava araçları dünya genelinde dikkat çeken ve hızla yayılan bir teknoloji haline gelmiştir. İnsansız Hava Araçları, insan olmadan uzaktan kumanda veya önceden belirlenmiş bir program doğrultusunda uçuş yapabilen, genellikle dronlar olarak da adlandırılan hava araçlarıdır. Bu teknolojinin tarihçesi, savaş amaçlarından ticari ve kişisel kullanıma kadar geniş bir yelpazede ilerlemiştir.

İnsansız hava araçlarının tarihi, 19. yüzyılın sonlarına dayanmaktadır. İlk İnsansız Hava Araçları, askeri keşif ve gözetleme amaçları için geliştirilmiş olup, I. Dünya Savaşı sırasında deneysel olarak kullanılmıştır. Ancak, teknolojinin sınırlılıkları ve kontrol edilebilirlik sorunları nedeniyle o dönemde yaygın bir kullanımı söz konusu değildi.

II. Dünya Savaşı ve sonrasında teknolojik gelişmelerle birlikte, İnsansız Hava Araçları üzerine yapılan çalışmalar artmış ve bu araçlar modern savaş alanlarında önemli bir yer edinmeye başlamıştır. 21. yüzyılda ise ticari amaçlar için de kullanılmaya başlanmış, tarım, madencilik, haber medyası ve eğlence gibi farklı sektörlerde kullanım alanları bulmuştur.

Gün geçtikçe, İnsansız Hava Araçları alanında birçok yeni çalışma yapılmaktadır, çalışmaların yanı sıra kullanım alanlarının da yaygınlaşması, bu bağlamda düzenleyici unsurlar gelmesine önyak olmuştur. Ülkemizde Sivil Havacılık Genel Müdürlüğü

(SHGM) tarafından, sınıflandırmaya alınan İnsansız Hava Araçları, sınıfları kapsamında belli şartlara tabi tutulmuşlardır. Aynı şekilde, İnsansız Hava Aracı pilotluğu da bu sınıflar kapsamında belirlenmektedir.

- İHA 0: Azami Kalkış Ağırlığı 500 g – 4kg aralığında olanlar,
- İHA 1: Azami Kalkış Ağırlığı 4 kg- 25kg aralığında olanlar,
- İHA2: Azami Kalkış Ağırlığı 25kg- 150 kg aralığında olanlar,
- İHA 3: Azami Kalkış Ağırlığı 150kg ve daha fazla olanlar

Şeklinde sınıflandırma mevcuttur. Bu sınıflandırma SHT-İHA talimatnamesinin 5.maddesinde mevcuttur. (SHT-İHA,2016)

### **1.1.1. İnsansız Hava Araçlarının Askeri Alanda Kullanılması**

İnsansız Hava Araçlarının en yaygın kullanım alanlarından biridir. Keşif, gözetleme, bombalama ve hedef tespit gibi görevlerde etkin olarak kullanılabilirler. Ayrıca versiyonuna göre Savaşan İnsansız Hava Aracı, Kamikaze İnsansız Hava Aracı gibi farklı fonksiyonları da mevcuttur.

İnsansız Hava Araçlarının bu alanda kullanılması sivil kayıplarının en aza indirmesinde de etkili olmaktadır.

İnsansız Hava Araçlarının Askeri kullanımı Türkiye dahil olmak üzere pek çok ülkenin hem savunma birimlerinde hem istihbarat birimlerinde yaygındır. Şekil 1’de Askeri alanda kullanıma ait bir İnsansız Hava Aracı olan Bayraktar AKINCI görülmektedir.



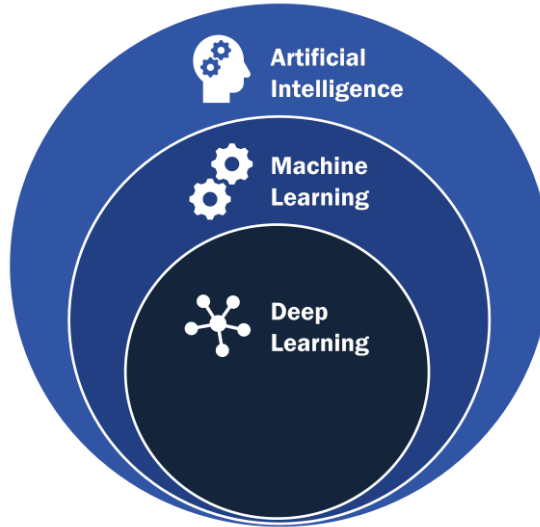
Şekil 1. 1 Bayraktar Akıncı İnsansız Hava Aracı

## 1.2.Yapay Zekâ

Yapay zekâ (YZ), günümüzde hızla gelişen ve birçok endüstriyi etkileyen bir teknoloji dalıdır. Bilgisayarların, problemleri çözmek, öğrenmek ve hatta insan benzeri zeki davranışlar sergilemek için kullanıldığı YZ, birçok sektörde devrim niteliğinde değişikliklere yol açmaktadır. YZ, bilgisayar sistemlerini, genellikle insan benzeri yeteneklerle donatmayı amaçlar. Bu yetenekler arasında öğrenme, problem çözme, dil işleme, görsel algı, ses tanıma ve karar verme gibi bir dizi zekâ özelliği bulunur. Yapay zekâ, veri analizi, örüntü tanıma ve karmaşık algoritmalar kullanarak bu görevleri gerçekleştirebilir.

Birçok alt alana sahip olan Yapay Zekâ sistemleri doğal dil işlemede yazma deneyimini artırma, robotikte karar verme kabiliyetini artırma, askeri ve ulaşım sanayinde otonomiye artırma ve geliştirme gibi amaçlara sahiptir.

Yapay Zekâ, Makine Öğrenmesi ve Derin Öğrenmenin kapsama birimleri, Şekil 2’de verildiği gibidir.



Şekil 1. 2 YZ, MÖ ve DÖ kapsama grafiği

Şekil 1.2’de görüldüğü üzere Derin Öğrenme en temel birimken, Makine Öğrenmesi onu kapsamakta, Yapay Zekâ ise tüm hepsini içine almaktadır.

### 1.2.1. Makine Öğrenmesi

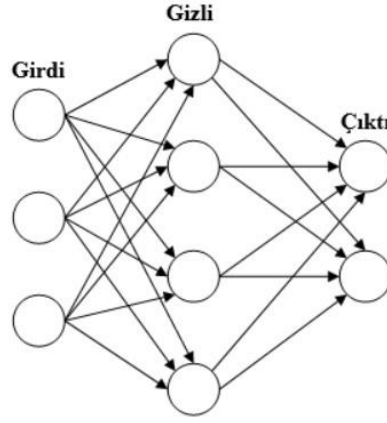
Makine Öğrenmesi, açıkça programlanmadan öğrenilmeyi sağlayan Yapay Zeka dalıdır. Algoritmaların ve istatistiksel modellerin kullanılmasıyla gerçekleştirilir. Sistemlerin veri üzerinden öğrenmesine olanak sağlamaktadır.

Veri Toplama, Özellik Çıkarımı, Model Eğitimi ve Tahmin üzerine dayalı bir sistematığe sahiptir. Finans, Sağlık, Pazarlama, Üretim gibi sektörlerde yaygın olarak kullanılmaktadır.

### 1.2.2. Derin Öğrenme

Derin Öğrenme, çok katmanlı Yapay Sinir Ağlarını (YSA) kullanan bir makine öğrenmesi alt birimidir. Derin Öğrenme modelleri, büyük veri setleri üzerinde çalışarak karmaşık ilişkileri çözebilmektedirler.

Derin Öğrenme, insan beyninin çalışma sistemini taklit etmektedir. Şekil 3’de verilmiş olan model Yapay Sinir Ağlarını örneklemektedir.



Şekil 1. 3 Yapay Sinir Ağları

Derin Öğrenmede her katman, bir önceki katmanın çıktısını alır ve ağırlıkları işleyerek bir sonraki katmana aktarır. Çalışma prensibi katmanlar arası aktarma ile gerçekleşmektedir.

Görüntü İşleme, Doğal Dil İşleme, Otonom Sistemler, Sağlık gibi sektörlerde yaygın olarak kullanılmaktadır.

### 1.3.Görüntü İşleme

Bilgisayarlı görü, bilgisayarların dijital görüntülerden nasıl anlam kazanabileceğiyle ilgilenen bir disiplindir. İnsanın görsel sisteminin yapabildiği yetenekleri, bilgisayarın yapması bilgisayarlı görü olarak tanımlanmaktadır.

Görüntü işleme ise, temelde bir görsel veri setini alarak bu veriyi analiz etme ve çeşitli işlemlere tabi tutma sürecidir. Bu işlemler arasında görüntü düzenleme, özellik çıkarma, desen tanıma, nesne algılama, rekonstrüksiyon ve sınıflandırma gibi bir dizi teknik bulunmaktadır.

Görüntü İşleme, Bilgisayarlı Görünün önemli bir yapıtaşını oluşturmaktadır.

### **1.3.1. Görüntü İşleme Teknolojisinin İnsansız Hava Araçlarında Kullanımı**

Görüntü İşleme Teknolojisi, bölümün başında bahsedildiği şekilde, görüntülerin analiz edilmesi ve işlenmesi ile ilgili yöntemleri içermektedir. İnsansız Hava Araçlarında bu sistem, Askeri, Sivil ve Ticari anlamda kullanılmaktadır.

Askeri alanda; Gözetleme, Keşif, Hedef Tanıma ve Takip, Haritalama, Arama ve Kurtarma gibi özel kullanım alanları mevcuttur.

### **1.4.Otonom Sistemler**

Teknolojinin hızla ilerlemesiyle birlikte, otonom sistemler hayatımızın birçok alanında önemli bir yer edinmekte ve geleceğin teknolojik gelişimlerinde önemli bir rol oynamaktadır. Otonom sistemler, insan müdahalesi olmadan kendi kendine işleyebilen, karar verebilen ve görevlerini yerine getirebilen sistemlerdir. Bu sistemler, birçok alanda kullanılabilir ve farklı düzeylerde otonomi seviyelerine sahiptir.

Otonom sistemlerin temeli, yapay zekâ, makine öğrenimi, sensör teknolojileri ve otomasyonun birleşimiyle oluşturulmuştur. Bu sistemler, çevrelerini algılayabilme, çeşitli durumlara tepki verebilme ve belirlenen hedefleri gerçekleştirme yeteneklerine sahiptir. Otonom sistemlerin gelişimi, özellikle robotik, otonom araçlar, yapay zekâ uygulamaları ve endüstriyel otomasyon gibi alanlarda hızla ilerlemektedir.

Otonom sistemler, genellikle belirli bir otonomi seviyesine sahiptir ve bu seviyeler, insan müdahalesinin ne ölçüde gerektiğini belirler. Yaygın olarak kabul edilen otonomi seviyeleri şunlardır:

- **Sıfır Otonomi (Seviye 0):** Bu seviyede, sistem tamamen insan kontrolünde çalışır ve hiçbir otonom işlevi bulunmaz.
- **Sınırlı Otonomi (Seviye 1-2):** Bu seviyelerde, sistem belirli koşullarda kısmi otonom hareket edebilir. Ancak, genellikle insan kontrolüne ihtiyaç duyar.

- **Yüksek Otonomi (Seviye 3-4):** Bu seviyelerde, sistem belirli görevleri büyük ölçüde otonom olarak gerçekleştirebilir, ancak acil durumlarda veya belirli koşullarda insan müdahalesi gerekebilir.
- **Tam Otonomi (Seviye 5):** Bu seviyede, sistemler her türlü görevi tamamen otonom olarak yerine getirebilir ve insan müdahalesine hiçbir şekilde ihtiyaç duymaz.

Otonom sistemlerin kullanım alanları oldukça geniştir:

- **Otomotiv Sektörü:** Otonom araçlar, sürücüsüz taşıma sistemleri ve otomatik park sistemleri gibi teknolojilerle otomotiv sektöründe önemli bir yer edinmektedir.
- **Endüstriyel Otomasyon:** Fabrikalarda üretim süreçlerinin otomatikleştirilmesi, robotik sistemlerin kullanımıyla verimliliği artırmaktadır.
- **Sağlık Hizmetleri:** Ameliyat robotları, tıbbi görüntüleme sistemleri gibi otonom teknolojiler sağlık sektöründe kullanılmaktadır.
- **Tarım ve Tarım Teknolojileri:** Otonom traktörler, toprak analizi sistemleri gibi uygulamalar tarım alanında etkili olmaktadır.
- **Lojistik ve Dağıtım:** Otonom hava araçları veya robotlarla paket teslimatı gibi lojistik süreçlerde otonom sistemler kullanılmaktadır.

Bu ve benzeri pek çok alanda kullanılan otonomi, gün geçtikçe yeni faaliyet alanlarına açılmakta ve yaygınlaşmaktadır.

Otonomi, teknolojinin geleceğini şekillendiren ve hayatımızı kolaylaştıran önemli bir unsurdur. Gelişen yapay zekâ, sensör teknolojileri ve otomasyon ile birlikte otonom sistemlerin kullanım alanları ve etkinliği sürekli genişlemektedir. Gelecekte, daha fazla sektörde otonom sistemlerin yaygınlaşması ve insan hayatına entegrasyonu beklenmektedir.

Otonom sistemlerin gelişimiyle birlikte, güvenlik, verimlilik ve yaşam kalitesi gibi konularda önemli iyileştirmelerin gerçekleşmesi umulmaktadır. Ancak, bu teknolojilerin kullanımı sırasında etik, yasal ve güvenlik gibi konuların da göz önünde bulundurulması gerektiği unutulmamalıdır.



### **1.5.Problemin Tanımı**

İnsansız Hava Araçlarının askeri bakımdan önemi, tüm Dünya tarafından kabul görmektedir ve bu alanda Dünya ülkeleri tarafından yatırımlar yapılmaktadır. Pek çok konuda, askeri düzeni rahatlatan bu sistemler, saldırı ve savunma fonksiyonlarıyla birlikte orduların havacılık birimlerinin değişilmez parçası olmuşlardır. Sadece Hava Kuvvetleri kapsamında değil Deniz ve Kara Kuvvetleri bünyesinde de İnsansız Hava Aracı-Silahlı İnsansız Hava Aracı envanteri daima genişletilmeye ve geliştirilmeye açık durumdadır.

Bu durum, teknoloji ve askeri anlamda bir rekabet halini de doğurmaktadır. Ülkeler, diğer ülkelere kıyasla teknolojik ve askeri alanda potansiyele sahip olan bu gelişimin öncüsü olmak için çabalamaktadırlar.

Savaşan İnsansız Hava Araçlarının, operasyonel faaliyetler özelinde, sıklıkla hava-kara müdahalelerinde sabit, noktasal hedefler üzerinde kullanılsa da hava-kara müdahalelerinde hareketli hedefler üzerinde kullanımı da mevcuttur. Bu kapsamda pek çok ülke ve pek çok firma tarafından üretilen Hava Araçları hava-kara muharebe özelliğine sahip olsa da hava-hava muharebe özelliğine sahip olmadığı bilinmektedir. Savaş uçakları ve jetlerinin böylesine gelişmiş ve gelişmekte olduğu dönemde, hava-hava muharebe fonksiyonu gitgide önemini arttırmaktadır. Bu yönden incelediğimizde, İnsansız Hava Araçlarına bu fonksiyonun kazandırılması, envanter, maddiyat ve askeri personel kullanımı başta olmak üzere birçok alanda avantajlar sağlamaktadır.

Hava'dan havaya savaşabilme fonksiyonuna sahip, tam otonomi ile donatılmış bir hava aracı sistemi, tüm bu özellikler ele alındığında önem ve yenilik arz etmektedir.

### **1.6.Çalışmanın Amacı**

Çalışma özelinde, insansız hava araçlarında kullanılan akıllı kontrol sistemleri üzerinde nesne tespit ve takip yöntemleri kullanılarak, araç otonomisinin kontrolüne bırakılmış, optimum başarıdaki takip yönteminin bulunması amaçlanmıştır.

Bilindiği gibi İnsansız Hava Araçlarının son dönemde gerek döner kanat gerekse de sabit kanat versiyonları halinde saldırı/savunma amaçlı kullanımları artmaktadır. Sabit hedeflere karşı son derece başarılı örnekleri mevcut olan bu sistemlerin hareketli hedefler üzerinde henüz kesinlik kazanmış bir uygulaması bulunmamaktadır. Özellikle

hava-hava muharebesi bakımından insansız hava araçlarının yetenek kabiliyetleri sınırlı kalmaktadır.

Hava-Hava muhaberesi yapabilme kabiliyetine sahip bir insansız hava aracı, takip yapma yeteneğine de sahip olmak zorundadır. Muharebe esnasında, rakip/düşman hava araçlarının da hareketi göz önüne alındığında, takip sağlanmadığı durumda hedef tespiti yapılsa bile başarı yakalanmayacaktır. Bu özelliklere sahip bir İnsansız Hava Aracını ele aldığımızda, Otonom Kontrol Sistemi'nin oldukça hızlı ve doğru karar alma yapısına sahip olması gerekmektedir. Bu gereklilik, nesne tespit ve takip algoritmaları kullanılarak yerine getirilebilir. Kullanılması gereken bu tespit ve takip algoritmalarında,

- Yüksek hız,
- Yüksek doğruluk,
- Uyumlu Mimari, gibi özellikler aranmaktadır. Çalışma kapsamında isterleri en başarılı şekilde karşılayan ve sistematığe uyumlu olarak çalışabilen algoritma/metot bulunması amaçlanmıştır. Bu algoritmanın çalışma prensibi,
- Kamera girdisinin, işlenmek üzere algoritmaya alınması,
- Girdinin üzerinden, daha önce eğitilmiş veri seti kullanılarak tarama yapılması,
- Girdi üzerinde nesne (rakip/düşman hava aracı) tespiti,
- Tespit edilen nesnenin, gelecek hareketlerinin tahmin edilmesi olarak işlemektedir.

### **1.7.Çalışmanın Kapsamı**

Çalışmanın kapsamında, ilk olarak nesne tespit ve takip algoritmaları için gerekli uyumluluklar göz önüne alınarak matematiksel modelleme, akış diyagramı ve sistem mimarisi oluşturulacaktır. Bu adım, sistem tasarımının temellerini oluşturacak ve demo öncesi yapının kurgulanmasını sağlayacaktır.

Daha sonra, belirlenen algoritmalar, tasarlanan mimariye göre gerçekleştirilerek, koda dökülecek, simülasyon ve demo uygulamaları ile teorik derlemeler, pratik denemelere dönüştürülecektir.

### 1.8.Literatür Özeti

İnsansız Hava Araçları ve Görüntü İşleme uygulamaları üzerinde nesne tespiti ve nesne takibine yönelik çeşitli çalışmalar mevcuttur. Nesne tespiti üzerine yapılan çalışmaları göz önüne alacak olursak, Havadan çekilen görüntülerde nesne tespiti üzerine odaklanan ve Gaikwad ve ark. tarafından yayımlanan bu makale, mevcut literatürü gözden geçirir ve farklı nesne tespiti tekniklerini detaylı bir şekilde karşılaştırır. Bu inceleme, İnsansız Hava Araçlarının nesne tespiti alanındaki mevcut durumu ve gelecekteki potansiyel gelişmeleri değerlendirir. [1]

Kim & Rohan & Rabah (2019) tarafından yayımlanan makelede ise Evrişimli sinir ağları kullanarak İHA'ların gerçek zamanlı nesne takibi araştırılmıştır. Çeşitli Evrişimli Sinir Ağları yapılarını ve algoritmalarını inceleyerek İHA'ların nesneleri takip etme yeteneklerini nasıl iyileştirebileceği de makalede sunulmuştur. [2]

Hava aracı üzerinden alınan görüntülerle takip işlemi üzerine yapılan çalışmalara değinirsek, 2013 senesinde Dang ve ark. Tarafından yapılan çalışmada AR (Arttırılmış Gerçeklik) Drone ile yerdeki objenin takibi yapılmıştır. İnsansız Hava Aracı, kapalı çevrim bir kontrolcü ile kontrol edilmiştir. Yer istasyonundan Kablosuz Bağlantılar (WiFi) kullanılarak kontrolü sağlanan bu hava aracında, anlık olarak alınan görüntü, HSV formatına çevrildikten sonra eşik(threshold) uygulanarak nesne tespiti sağlanır. Tespit edilen obje ortada tutulacak şekilde piksel hesabı yapılarak, hava aracının hizalanması gerçekleştirilmiştir. [3]

Yine 2013 senesinde, Bi ve ark. İnsansız Hava Aracına otonom olarak nesne takibi ve iniş yaptırmışlardır. Görüntünün RGB değerleri üzerine uygulanan eşik(threshold) işlemi ile uygulamalar gerçekleştirilmiştir. Uygulamalardan biri pist üzerine iniş iken, diğeri kırmızı araç takibidir. Bu takip işlemi düz bir zeminde gerçekleştirilmiştir. [4]

Pestana ve ark. Tarafından 2013'te yapılan çalışmada döner kanatlı İnsansız Hava Aracı'nın görmeye dayalı kontrolü sağlanmıştır. GPS kullanılmaya bu sistemde, mekanizma görüntüye bağlı olarak kontrol edilmiştir. Nesne takip algoritması olarak Açık Takip, Öğrenme, Tespit (OpenTrackingLearningDetection – OpenTLD) kullanılmıştır. Çalışma sırasında insan, bitki, araba vb. takibi sağlanmıştır. [5]

Hava Aracı özelinde durulmadan, nesne takibi ile ilgili genel olarak literatür taraması yapıldığında ise Kalman Filtresi ile ilgili birçok çalışma bulunmaktadır.

Weng ve ark. Tarafından 2006 yılında gerçekleştirilen çalışmada Kalman Filtresi kullanılarak hareketli nesne takibi metodu ortaya konulmuştur.[6] Seçilen nesne, görüntüden ayrıştırılarak, RGB özelliklerinden baskın olan renk çıkarılmıştır. Tespit edilen nesnenin hareket tahminleri ise Kalman Filtresi ile elde edilmiştir. Bu çalışma, 2010 senesinde Li ve ark. tarafından yapılmış başka bir çalışmayla ileri seviyeye taşınmıştır [7]. Bu çalışmada ise, takip edilen nesnenin önüne geçen başka bir cismin birbirini engellemesi durumuna karşı takip işleminin devamı sağlanmıştır.

Robert ve ark. bu çalışmalara örnek olarak sunulacak bir çalışmayı 2012 senesinde gerçekleştirmişlerdir. Bayes ağları kullanılarak ormanlık arazide otonom uçuş yapılarak ağaçların haritalandırılması işlemi, bu çalışma kapsamında gerçekleştirilmiştir. [8]

### 1.9.Çalışmanın Gerçekçi Kısıtlar Açısından Analizi

- 1) **Hava Şartları:** Hava Aracının görüntü aldığı esnada, hava şartlarına bağlı olarak potansiyel başarısı değişebilir. Havanın çok güneşli olduğu durumlarda parlamanın fazla olması veya havanın kapalı olduğu günlerde renk bareminin birbirine yakın olması olumsuz sonuç yaratabilmektedir. Bu duruma çözüm olarak, filtreleme işlemleri yapılabilir.
- 2) **Sistem Performans Kriterleri:** Sistemin yoğun bir algoritmayı kullanacağı göz önüne alındığında, grafik işlemci biriminin yüksek performansa sahip bir işlemci olması zaruri önem taşımaktadır. Görüntünün Merkezi İşlemci Biriminde işlenmesi durumunda alınacak olan görüntünün akışı oldukça yavaş olacaktır.
- 3) **Haberleşme Bağlantıları Kriterleri:** Hava Aracı ile Yer İstasyonu arasında anlık olarak veri akışı gerçekleştirileceği için haberleşme bağlantısı kesilmeyecek şekilde ayarlanmalıdır. Çevredeki haberleşme frekansları, kullanılan frekans bastırmamalıdır.

- 4) **Kamera Kalitesi:** Kameradan alınacak olan görüntünün işlenecek olması, görevin gerçekleştirilmesinde başlıca öneme sahiptir. Bu sebeple kameradan gelecek görüntünün gerçek zamanlı ve optimum kalitede olması gereklidir.

## **BÖLÜM 2. MATEMATİKSEL YÖNTEM ve TASARIM**

Bu bölümde çalışmada kullanılan yöntem ya da yöntemlere ait bilgilendirme ve yöntemin çalışmada kullanımına ilişkin literatür araştırmalarına yer verilmiştir. Yöntem veya yöntemlerin problemin çözümünde kullanımı ve tasarım aşamaları da bu bölümde verilmektedir.

### **2.1. Yöntem Hakkında Genel Bilgi**

Bu Bölümde Çalışmanın Kapsamı gereğince, Nesne Tespiti, Nesne Takibi, İstatistiksel Öğrenme ve Kalman Filtresi incelenmiş, çalışma süresince kullanılan yönteme dair bilgiler verilmiştir.

#### **2.1.1. Nesne Tespiti**

Öncelikli olarak nesne tanıma sistemi, tanınacak olan cisme ait görsellerin etiketlenmesi ve Yapay Zekâ eğitime tabii tutulması üzerine gerçekleştirilmeye dayanan bir işlemdir. Yani, kullanılacak fonksiyonlar ne olursa olsun bilgisayarları görü algoritmalarının çalıştırılması, daha önce yapay zekaya verilmiş, eğitilmiş veri setlerine dayalıdır.

Proje dahilindeki başlangıç amacı, otonom hareketler mimarisinin temelinde ilk olarak elde edilen görüntülerin işlenip belirli tanımlamalar yapılması suretiyle hava aracının çeşitli manevralarına izin verecek olan algoritmayı geliştirmek yatmaktadır. Bu algoritmanın temelinde hedef alınan aracın nitelik ve nicelik yönlerinden tespiti bulunmaktadır. Hava aracının bu tespiti otonom yapabilmesi, tespit sürecinde hata payının sıfıra yakın ve tanımlamaların yeterliliğinin ve güvenilirliğinin iyi düzeyde olması açısından optimum değerlendirmenin yapılması başlıca gerekliliklerdendir. Zira nesne tespitindeki hata payının yüksek olması ve buna dayalı olarak nesne tespitinin doğru yapılamaması durumlarında, ilk aşama başarılı olarak gerçekleştirilemeyeceği için ikinci aşama olan takip ve hizalanma aşaması da sekteye uğrayacak ve sistem başarılı bir sonuç elde edemeyecektir.

Nesne tespit algoritmaları; nesnenin anlık lokasyonunun, sınıfının, nitel ve nicel özelliklerinin tespitinde kullanılan ve bunun için de yapay sinir ağlarını kullanan derin öğrenme yazılımlarıdır. İnsansız hava aracı için optimum algoritmanın seçileceği değerlendirme sürecinde ilk olarak algoritma modelleri ele alınmıştır. R-CNN (Region Based Convolutional Neural Networks - Bölgesel Bazlı Yapay Sinir Ağları), Faster R-CNN (Faster Region Based Convolutional Neural Network - Hızlı Bölgesel Bazlı Yapay Sinir Ağları), SSD (Single Shot Detector - Tekli Vuruş Dedektörü), YOLO (You Only Look Once - Sadece Bir Kez Bak) algoritmaları üzerinde durulmuştur.

#### **2.1.1.1. Tekli Vuruş Dedektörü**

Gerçek Zamanlı Nesne Tespiti için tasarlanmış bir algoritmadır. Farklı boyutlarda öneri kutuları kullanarak nesnelerin farklı ölçeklerini ele alır. Her öneri kutusu için birden fazla sınıf tahmini yapan bu algoritma, bir tek geçişte hem nesne konumunu hem de sınıfını belirler. Bu durum, algoritmanın hızlı çalışma mantığını sağlamaktadır. Algoritmanın dezavantajı ise nesne tespitinde muadil algoritmalara kıyasla başarı oranının az olmasıdır.

#### **2.1.1.2. Bölgesel Bazlı Yapay Sinir Ağları**

Nesne Tespiti için bölge tabanlı bir Evrişimsel Sinir Ağı kullanır. Öncelikli olarak görüntü üzerinde potansiyel nesne bölgelerini belirler, her bölge için özel olarak eğitilmiş Evrişimsel Sinir Ağından geçirilir ve özellik haritasını elde eder. Her bölge, özellik haritasındaki özelliklere dayanarak sınıflandırılır ve iyileştirilmiş nesne konumları için regresyon uygulanır. Algoritmanın çalışma mantığı bu şekildedir. Dezavantajı ise alınan görüntü üzerinde birkaç farklı Nesne Tespiti için çok defa bölge önerisi gerekmektedir. Bu, algoritmayı yavaşlatmaktadır.

#### **2.1.1.3. Hızlı Bölgesel Bazlı Yapay Sinir Ağları**

Bölgesel Bazlı Yapay Sinir Ağlarının (2.1.1.2), öneri kutularını oluşturmak için ayrı bir Evrişimsel Sinir Ağı ekleyerek hızı arttırmayı hedefler. Görüntü üzerindeki potansiyel nesne bölgelerini belirlemek için Bölgesel Amaçlı Ağ (RPN) kullanır. Bölgesel Amaçlı Ağ ve nesne sınıflandırma için aynı ağı kullanır bu da hızlı bir eğitim ve tahmin sürecini beraberinde getirir. Algoritmanın dezavantajı ise, alınan görüntü üzerinde birkaç farklı Nesne Tespiti için çok defa bölge önerisi gerekmesidir. 2.1.1.2

başlığı altında belirtilen Bölgesel Bazlı Yapay Sinir Ağları algoritması ile ortak dezavantaja sahiptir.

#### **2.1.1.4. Sadece Bir Kez Bak Algoritması**

Sadece Bir Kez Bak (YOLO) Algoritması, nesneyi sınıflandırmak ve konumlandırmak için tek bir geçiş kullanmaktadır. Görüntü üzerindeki tüm nesnelerin sınıf ve konumlarını aynı anda tahmin eden algoritma, sınıf ve konum tahminleri arasındaki hata için özel kayıp fonksiyonu kullanmaktadır. Sadece Bir Kez Bak Algoritması, gerçek zamanlı uygulamalarda kullanılan, muadil algoritmalara kıyasla hızlı olan bir Nesne Tespit yöntemidir.

Algoritmanın dezavantajı ise birbirine çok yakın ve küçük boyutta olan nesneleri bulmakta zorlanmasıdır.

#### **2.1.1.5. Algoritmaların Kıyası**

**Hız Bakımından Karşılaştırma:** Projenin gerçek zamanlı uygulaması ele alındığında, hava aracının hareketiyle birlikte rakip hava araçlarının da hareketi var olacağından dolayı, sistemin hızlı bir şekilde nesne tespit işlemini gerçekleştirerek, nesne takip işlemine geçmesi gerekmektedir.

**Doğruluk Bakımından Karşılaştırma:** Sistemin ilk aşamasında yaşanacak aksaklıklar, sistemin ikinci aşamasını doğrudan etkileyeceği için nesne tespit algoritmasının doğruluk oranı oldukça önemli bir kriterdir.

**Mimari Karmaşıklık ve Örnek Çeşitliliği Bakımından Karşılaştırma:** Seçilecek olan algoritma mimarilerinin sadeliği, proje için çok önemlidir. Bunun sebebi, sisteme uyarlanabilir olmasından ve kullanılabilirliğinin basit olmasından geçmektedir. Aynı zamanda kullanılacak olan algoritmalarındaki örnek çokluğu, sistemin oluşturulma hızını önemli ölçüde etkileyecektir.

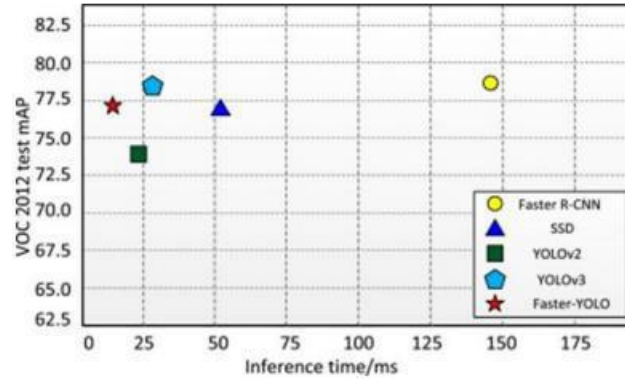
Bu kriterleri ele alarak karşılaştırma yaptığımızda şu özellikler öne çıkmaktadır:

- **Hız:** Sadece Bir Kez Bak, tek bir geçişle çalıştığı için diğerlerine göre genellikle daha hızlıdır.



- **Doğruluk:** Bölgesel Bazlı Yapay Sinir Ağları ve Hızlı Bölgesel Bazlı Yapay Sinir Ağları, özellikle nesne tespiti konusunda daha yüksek doğruluk sunabilir, ancak daha yavaş çalışabilirler.
- **Mimari Karmaşıklığı:** Sadece Bir Kez Bak, basit ve hızlı bir mimariye sahipken, Bölgesel Bazlı Yapay Sinir Ağları ve Hızlı Bölgesel Bazlı Yapay Sinir Ağlar daha karmaşık yapılar içerir.

**Şekil 2.1'deki** karşılaştırma tablosu ve yukarıda detaylıca açıklanmış olan kriterleri ele aldığımızda, Sadece Bir Kez Bak algoritması kıyası yapılan ikame algoritmalara oranla öne çıkmaktadır. Kıyas tablosunda karşılaştırılmaya sunulan algoritmanın v3 modeli olduğunu ele alırsak, güncellenen versiyonların çok daha başarılı sonuç elde edeceği gözükmemektedir.



Şekil 2. 1 Sadece Bir Kez Bak Model Kıyaslama Tablosu

Mimari karmaşıklığının nispeten basit olması ve örnek çokluğunun bulunması büyük bir avantajken, hızıyla da öne çıkan algoritma, diğer algoritmalara göre doğruluk payında biraz geride kalmaktadır. Lakin Sadece Bir Kez Bak Algoritmasının doğrulukta geri kalmasının en büyük sebebi birbirine çok yakın veya küçük boyutta olan objeleri bulmakta zorlanmasından geçer. Proje özelinde hava araçları üzerinde çalışılacağı için aynı ortamda birden çok küçük nesneye tabi tutacağımız bir işlem bulunmamaktadır. Dolayısıyla bu dezavantaj sistemi etkilememektedir.

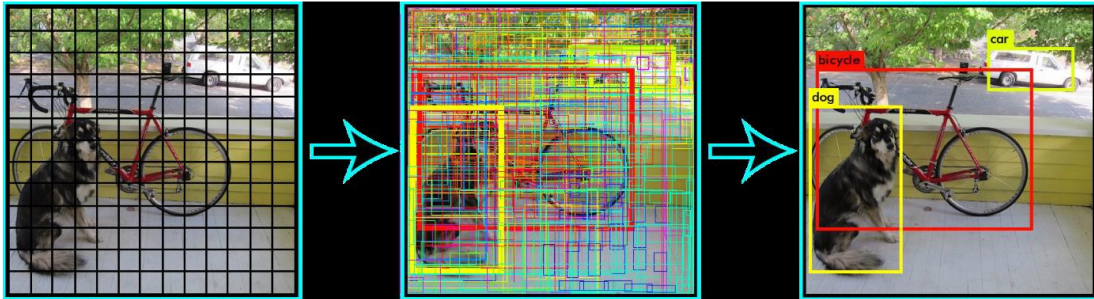
Sadece Bir Kez Bak, Nesne Tespiti görevini bir regresyon sorunu olarak ele almaktadır. Görüntü üzerindeki nesnelerin her biri için sınıf tahminleri arasında korelasyon oluşturur. Bu, bir nesnenin birden fazla sınıfa ait olabileceği durumları ele alır. Görüntüyü, bir ızgara hücrelerine böler ve her hücre, o bölgedeki nesnelerin konumunu ve sınıf tahminlerini içerir.

Görüntü, önceden eğitilmiş bir Evrimsel Sinir Ağı ile işlenir. Bu, görüntüdeki önemli özelliklerin çıkarılarak nesne tahminlerini iyileştirmektedir. Her hücre, o bölgedeki

nesnelerin sınıfını ve konumunu tahmin eder, bu tahminler bir ölçü belirleyici (hassaslık değeri – confidence score) ile birleştirilir. Bu belirleme ile birlikte belirli bir güven eşiğinin altındaki tahminler filtrelendirir. Ardından, aynı nesneyi temsil eden çoklu tahminler arasında yalnızca en güvenilir olanı seçmek için Maksimum Olmayan Bastırma (Non-Maximum Suppression) uygulanır.

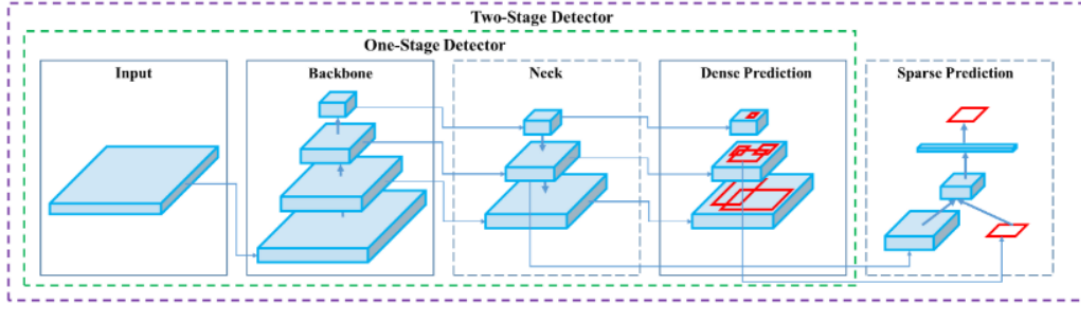
Bu işlemler sonrasında, algoritma, nesne tespiti sonuçlarını verir, bu sonuçlar sınıf etiketleri ve konum bilgilerini içerir.

Hava aracı yüksek hıza sahipken yine yüksek hıza sahip başka bir hava aracının tespiti, takibi, üzerinden görüntü alınıp işlenmesi, gerektiğinde kaçış algoritmasının devreye girmesi ve tüm bunların ani manevra gerektiren durumlar olmaları nedeniyle nesne tespitinin yapılacağı algıtmada hız ön planda olmalıdır, açılımında da belirtildiği üzere tespitin tek seferlik ve hızlı biçimde yapıldığı gerçek zamanlı algoritma modeli olan Sadece Bir Kez Bak algoritmasının diğer modellerden daha hızlı olmasının sebebi görüntü verisinin tamamını tek seferde nöral ağdan geçiriyor olmasıdır. Birçok hava aracının her birinin tespitinde Sadece Bir Kez Bak Algoritmasının otomatik olarak merkezi algılaması, Çevreleyici Kutu (Bounding Box) ve tahmin vektörü oluşturup güven skorunu otomatik olarak belirtme özelliğine sahip olması avantajlı olarak değerlendirilmiştir. Algoritmanın çalışma prensibi **Şekil 2.2**'de verilmiştir.



Şekil 2. 2 Algoritma Çalışma Prensibi

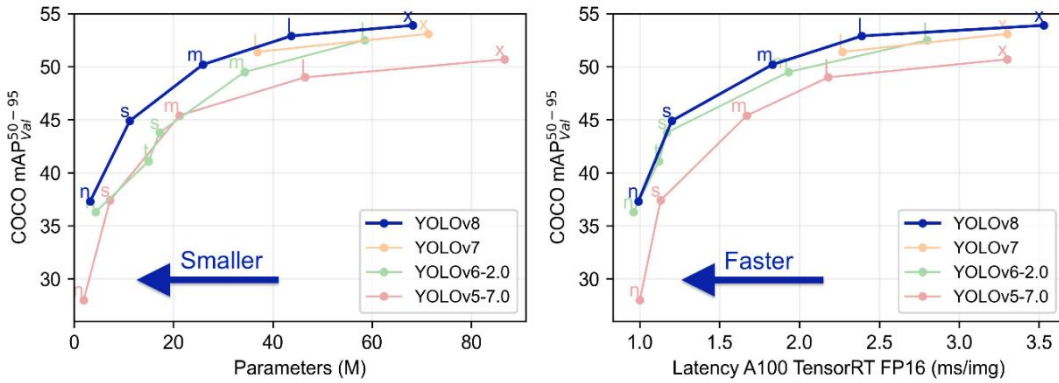
Algoritmanın model yapısı ise **Şekil 2.3**'de verilmiştir.



Şekil 2. 3 Yolo Algoritması Model Yapısı

Sadece Bir Kez Bak Algoritmasının Versiyon 8(v8) modeli, optimize edilmiş bir versiyondur. Önceki versiyonlara kıyasla optimum performans ve hız sağlamaktadır. Model Optimizasyonu bu versiyonun öne çıkmasında etkin olan parçadır. Aynı zamanda bu versiyon düşük kaynak tüketimi ile daha geniş bir kullanıcı kitlesine hitap etmektedir.

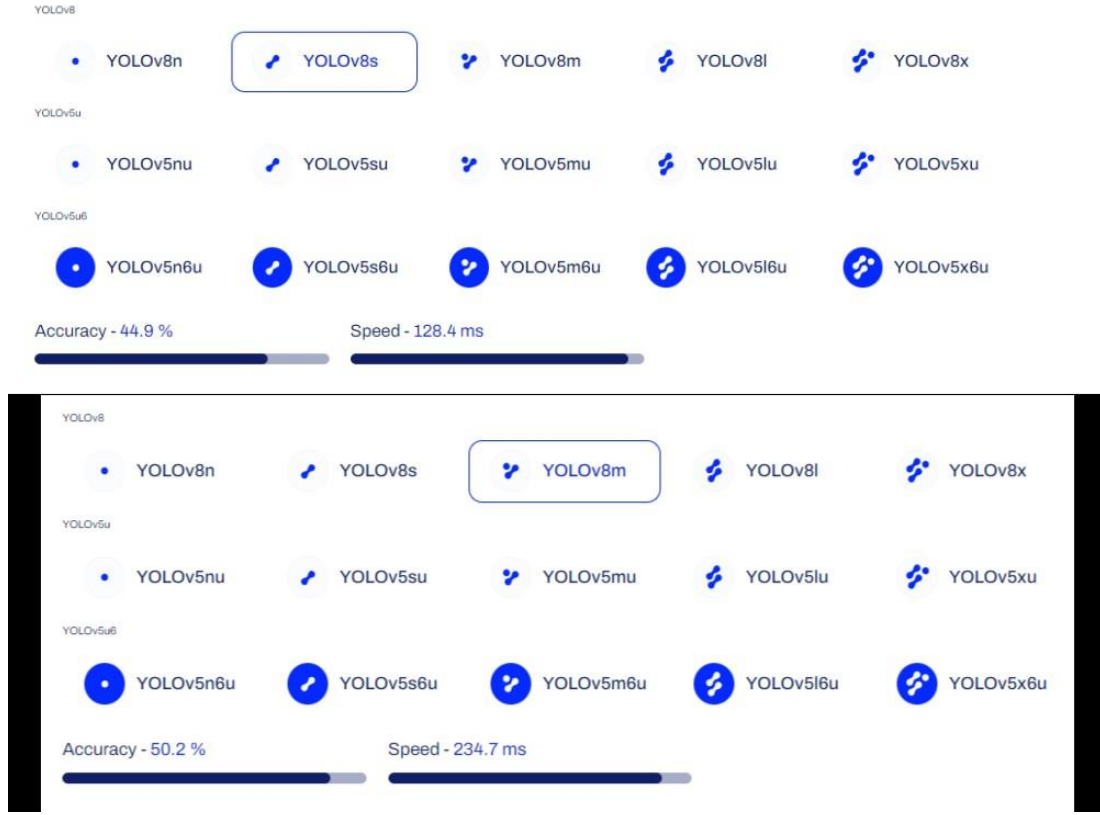
Sadece Bir Kez Bak algoritması modelinin bilgisayarlı görü uygulaması için kullanılan makine öğrenmesi kütüphanesi olan PyTorch implementasyonuna izin vermesi ve yerleşik sürüm olması sebebiyle kullanılacak olan diğer kod birimlerinde uygunluk göstermesi nedeniyle Sadece Bir Kez Bak versiyon 8 (v8) olarak seçilmesine karar verilmiştir.



Şekil 2. 4 Sadece Bir Kez Bak Algoritma Modelleri Kıyas Tablosu

Sadece Bir Kez Bak v8 algoritması içerisinde seçilecek olan model ise **Şekil 2.4'te** verilen tabloya göre seçilmiş ve Küçük (Small) modeli olarak belirlenmiştir. Grafikte verilen değerlerin optimum kesişmesi yaklaşık olarak Sadece Bir Kez Bak v8 Orta (Medium) ve v8 Küçük (Small) modellerine üzerine denk gelmektedir. Bu iki algoritma arasında kıyas yaptığımızda ize, isterler göz önüne alındığında hız faktörünün önemli etkisi ele alınarak Küçük modeli tercih edilmiştir. Bu kıyaslama **Şekil 2.5'te** gösterilmiştir. Aynı şekilde Sadece Bir Kez Bak versiyonlarının kıyası da

şekil üzerinden açıkça gözükmekte ve seçilen algoritma olan v8'in öne çıktığı anlaşılmaktadır.



Şekil 2. 5 Orta (Medium) ve Küçük (Small) Paketlerinin Kıyaslanması

Görüntü İşleme Algoritmasında, nesne tespiti yaptıktan sonra yakaladığımız nesneyi belirli süre boyunca takip edilebilmesi için ayrı bir takip algoritması eklentisi yapılması gerekmektedir. Bu algoritma seçilen nesne tespit algoritması olan Sadece Bir Kez Bak Algoritması ile uyumlu çalışma özelliğine sahip olmalı, aynı zamanda görüntü üzerindeki takip görevini optimum bir şekilde yapmalıdır.

### 2.1.2. Nesne Takibi

Nesne takibi, kullanıcı tarafından belirlenen alanın içerisinde algılanan şeklin, zaman geçtikçe akan karelerden takip edilmesi olayıdır. Şeklin algılanması, daha önce yapay zekâ eğitimi sırasında tanıtılan etiketlerin eğitim sonunda ortaya çıkan ağırlık dosyası çıktılarının, görüntü işleme kod bütünü içerisine entegre edilmesi ile gerçekleşir. Algılanan ve etiketi verilen bu şeklin akan zaman dilimindeki her bir kareler üzerinden takip edilmesi ise takip algoritmasının temelini oluşturmaktadır.

Görüntü İşleme Sistemi'nin hazırlanması sırasında yapılan literatür çalışmalarında, isterler doğrultusunda gerçekleştirilmesi gereken görevlerin sadece nesne tespit algoritmaları ile yapılması sonucunda doğruluk oranında sıkıntılar yaşanabileceği saptanmıştır. Bu sebeple sistem yazılım mimarisinde nesne takibi algoritması kullanılması gerekmektedir.

#### **2.1.2.1. Dönüşe Git Algoritması (GOTURN)**

Dönüşe Git (GOTURN) algoritması, 2016 yılında David Held, Sebastian Thrun ve Silvio Savarese tarafından "Learning to Track at 100 FPS with Deep Regression Networks" adlı makalede tanıtılmıştır. Algoritma, nesne takibini bir regresyon problemi olarak formüle eder. Hedef nesnenin geçmiş konumunu öğrenmek için bir derin öğrenme modeli kullanır ve ardından bu modeli mevcut çerçevedeki nesne konumunu tahmin etmek için kullanır. Algoritma, 2 aşamadan oluşmaktadır.

Eğitim aşamasında nesnenin hareketini öğrenen bir regresyon ağı kullanır. Eğitim verileri, bir referans çerçevesi içinde nesnenin konumunu ve hareketini içerir. Ağ, bu referans çerçevesini giriş olarak alır ve çıkış olarak nesnenin yeni konumunu tahmin eder.

Takip aşamasında ise, eğitilen model, hedef nesnenin geçmiş konumunu anlamak için kullanılır. Mevcut çerçevede nesnenin konumunu tahmin etmek için regresyon ağı kullanılır ve bu bilgi takip işlemi için kullanılır.

#### **2.1.2.2. Derin Basit Çevrimiçi ve Gerçek Zamanlı Takip Algoritması (DeepSort)**

Algoritma, takip edilen nesnelerin kimliklerini korumak için geliştirilmiş olan Basit Çevrimiçi ve Geliştirilmiş Takip (SORT) algoritmasını temel almaktadır ve 5 aşamadan oluşmaktadır.

İlk adım, bir nesnenin bir karede tespit edilmesidir. İkinci adımda her tespit edilen nesne için özellik vektörleri çıkarılır. Üçüncü adım olarak, ilk tespitlerle birlikte bir nesnenin takibini başlatmak için bir izleme numarası atanır. Dördüncü adımda, takip sırasında önceki çerçevelerdeki nesnelerin özellik vektörleri kullanılır ve derin öğrenme modeli ile takip sağlanır. Beşinci adımda ise, aynı nesne, çerçeveler arasında farklı pozisyonlarda bulunsa bile, özellik vektörlerinin benzer olacağından yola çıkılarak nesne kimliklerinin koruma işlemi gerçekleştirilir.

İzleme sırasında, bir nesnenin kaybolması durumunda kayıp filtresi kullanılarak nesnenin konumu tahmin edilir.

### 2.1.3. İstatistiksel Öğrenme

İstatistiksel öğrenme, hipotezler ve veriler üzerine kurulmaktadır. Burada, hipotezler, alanın nasıl çalıştığına dair teorileri belirtirken, veriler kanıt olarak nitelendirilebilir. İstatistiksel öğrenme, genellikle denetimli ve denetimsiz öğrenme olmak üzere iki temel yaklaşıma ayrılmaktadır.

- **Denetimli Öğrenme:** Denetimli öğrenmede, modelin eğitileceği veri setinin daha önce etiketlenmiş olduğu öğrenme türüdür. Bu yaklaşım, girdi verisi ile çıktı verisi arasındaki ilişkiyi öğrenerek, yeni, görülmemiş veriler üzerinde tahminler yapabilen modeller oluşturmayı amaçlar.
- **Denetimsiz Öğrenme:** Denetimsiz öğrenme, etiketlenmemiş veri setleri üzerinde çalışır. Model, bu veri setindeki desenleri algılamaya çalışır ve veri içindeki yapıları anlamaya çalışır. Denetimsiz öğrenme, Denetimli öğrenmeye kıyasla daha zorlu bir süreç olarak bilinmektedir.

İstatistiksel öğrenme, yapay sinir ağları ve derin öğrenme gibi alt dalları içerir.

Yapay sinir ağları, insan beynindeki nöron ağlarına benzer bir şekilde çalışan matematiksel modellerdir. Derin öğrenme ise çok katmanlı yapay sinir ağlarının kullanıldığı bir öğrenme yöntemidir.

Makine öğrenmesi, bilgisayar sistemlerinin belirli görevleri öğrenmelerine ve bu görevleri gerçekleştirmelerine odaklanan bir alanı kapsar.

- **Denetimli ve Denetimsiz Öğrenme:** Makine öğrenmesi hem denetimli hem de denetimsiz öğrenmeyi içerir. Denetimli öğrenme, modelin etiketlenmiş veri setleri üzerinde eğitildiği bir süreçtir. Denetimsiz öğrenme ise etiketlenmemiş veri setleri üzerinde desenleri keşfetmeye odaklanır.
- **Derin Öğrenme:** Makine öğrenmesinde önemli bir alt dal, derin öğrenmedir. Bu, çok katmanlı yapay sinir ağlarını kullanarak karmaşık görevlerde yüksek başarı elde etmeye odaklanır.

Genel olarak bakıldığında İstatistiksel Öğrenme ve Makine Öğrenmesi arasında büyük farklar bulunmamaktadır. İstatistiksel Öğrenme ve Makine Öğrenmesi kavramları Tablo 2.1’de farkları ortaya konularak incelenmiştir.

Tablo 2. 1 İstatistiksel Öğrenme ve Makine Öğrenmesi

Özellikler	İstatistiksel Öğrenme	Makine Öğrenmesi
Veri Türleri ve Eğitim Yaklaşımı	Genellikle etiketlenmiş veri setleri üzerinde odaklanır. Denetimli ve denetimsiz öğrenmeyi içerir, ancak genellikle denetimli öğrenme üzerinde vurgu yapar.	Hem etiketlenmiş hem de etiketlenmemiş veri setleri üzerinde odaklanır. Denetimli, denetimsiz ve güçlendirme öğrenmesi gibi çeşitli öğrenme türlerini içerir.
Yapı ve Modelleme Yaklaşımı	Daha geleneksel istatistiksel yöntemlere dayanabilir. Belirli modeller (regresyon, sınıflandırma) üzerine odaklanabilir.	Daha geniş bir yelpazede model ve algoritma çeşitliliğine sahiptir. Yapay sinir ağları, derin öğrenme gibi karmaşık yapıları içerir.
Kompleks Veri İşleme	Genellikle daha küçük veri setlerinde başarılıdır. Karmaşık yapılar için sınırlı performans gösterebilir.	Büyük veri setlerinde ve karmaşık yapılarla başa çıkmak için daha etkilidir. Derin öğrenme, karmaşık desenleri tanıma konusunda öne çıkar.
Uygulama Alanları	Geleneksel istatistiksel analizlere dayalı uygulama alanlarına odaklanır. Örneğin, ekonomi, epidemiyoloji.	Geniş bir yelpazede uygulama alanlarına sahiptir. Sağlık, finans, görüntü işleme, doğal dil işleme gibi birçok alanda kullanılır.

#### 2.1.3.1. Bayes Teoremi

Bayes Teoremi, bir rassal değişken için olasılık dağılımı içinde koşullu olasılıklar ile marjinal olasılıklar arasındaki ilişkiyi gösterir. Yani bu kuramı formüle edecek olursak: **(Denklem 2.1)**

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Denklem 2. 1

Bu denklemde:

- $P(A|B)$ , B şartı altında A'nın olasılığını temsil eder.
- $P(B|A)$ , A şartı altında B'nin olasılığını temsil eder.
- $P(A)$ , A'nın genel olasılığını temsil eder.
- $P(B)$ , B'nin genel olasılığını temsil eder.

Özetle, Bayes Teoremi; Gözlemlenmiş bir “X” olasılığı üzerinden gözlemlenecek olan “Y” olasılığı hakkındaki inançların ne şekilde güncelleştirebileceğini ortaya çıkartır. [9]

Bayes teoreminden yola çıkarak, istatistiksel öğrenmede bahsi geçen Bayes Ağlarını yorumlamak da mümkündür. Yüz tanıma uygulaması, Bayes Ağlarına örnek olarak gösterilebilir.[10] Uygulama özeline, görüntülerin özellikleri dalgacıkları dönüşümü kullanılarak çıkarılmış ve Bayes ağları ile sınıflandırılmıştır.

### 2.1.3.2. Parçacık Filtresi

Parçacık Filtresi, belirsiz ve karmaşık ortamlarda hareket eden nesnenin durumunu tahmin etmek için kullanılan, olasılık temelli filtreleme yöntemidir. Kalman Filtresine kıyasla, Parçacık Filtresinin işlem yükü daha fazladır.

Parçacık Filtresinin, çalışma prensibi, bir nesnenin durumunu belirli bir zaman aralığında güncellemektir. Öncelikle, önceki durumu belirli bir hareket modeli kullanarak günceller, bu adım, nesnenin beklenen hareketini temsil eder. Daha sonra, gözlemleri kullanarak nesnenin durumunu günceller ve gerçek Dünyadan alınan verilere dayalı olarak nesnenin tahmin durumunu düzeltir.

Parçacık Filtresi, belirli bir durumu temsil etmek için parçacık seti kullanır. Her parçacığın ağırlığı mevcut gözlemlerle uyumlu olup olmadığına göre belirlenir. Uyumlu parçacıklar, daha yüksek ağırlıklarla değerlendirilir.

Ağırlıklandırılmış parçacık seti, nesnenin olası durumlarını temsil eden bir olasılık dağılımını oluşturur. Ağırlıklarına göre parçacıkların tekrar örneklendirilmesi, daha iyi tahminlere odaklanarak filtreleme performansını artırır. Parçacık Filtresinin temel adımları özet olarak bu şekildedir.

### 2.1.4. Kalman Filtresi

Kalman Filtresi, gerçek zamanlı veri analizi ve tahminleme konularında yaygın olarak kullanılan bir matematiksel filtreleme tekniğidir. Bu filtre başta uzay ve havacılık olmak üzere birçok uygulama alanında kullanılmaktadır.

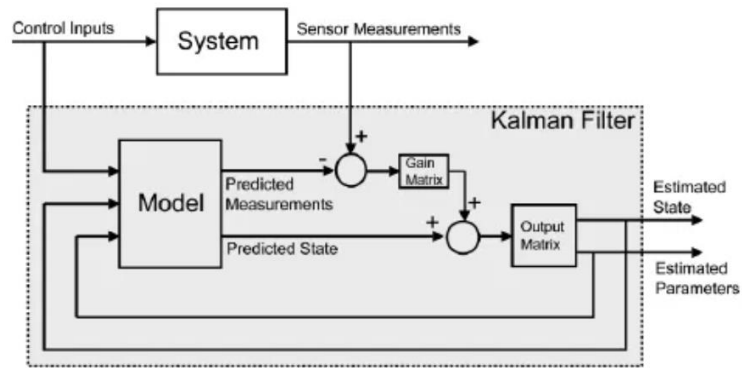


Matematiksel Sistem Teoristi olan Rudolf Kalman tarafından 20.yy'da bulunan algoritma, Kalman'ın adına ithaf edilmiştir. Bu algoritma, Amerikan Apollo Uzay programının temel taşlarından biri olarak bilinmektedir.

Temel olarak, belirsiz ve gürültülü gözlemleri kullanarak bir sistem durumunu tahmin etmeye çalışır. Kalman Filtresi'nin temel prensibi, sistem durumunu tahmin etmek ve ardından gerçek ölçümlerle bu tahmini güncellemektir. Bu süreç, bir döngü içinde tekrarlanır: tahmin yap, ölçüm al, tahmin güncelle.

Kalman Filtresi, doğrusal zaman sürekli bir sistem modelini temsil eden bir dizi diferansiyel denklemle başlar. Ancak, genellikle sistem ve gözlemler arasındaki ilişkiyi ifade etmek için doğrusal zaman sürekli bir sistemden, doğrusal zaman aralıklı bir sistem modeline geçilir. **Şekil 2.6** üzerinde görüldüğü gibi Kalman Filtresinin modellemesi, kapsamlı bir sistematiği içermektedir.

Kalman Filtresi, Bayes temelli bir filtredir. KF Algoritması, gürültülü bir gözlem veri akışı üzerine çalışarak hatayı en aza indirmeye amaçlayarak filtreleme yapar ve sistemin fiziksel karakteristiklerini modelleyerek üretilen gelecek bir sonraki adımın matematiksel tahminine göre hatayı optimize eder [12].



Şekil 2. 6 Kalman Filtresi Modellemesi

Kalman Filtresi, iki temel adımdan oluşur. Bunlar, Tahmin ve Güncellemedir. Bu adımların matematiksel olarak anlatımı şu şekilde yapılabilir:

#### Tahmin Adımı:

$$X_{k|k-1} = F_k \cdot X_{k-1|k-1} + B_k \cdot u_k \quad \text{Denklem 2. 2}$$

$$P_{k|k-1} = F_k \cdot P_{k-1|k-1} \cdot F_k^T + Q_k \quad \text{Denklem 2. 3}$$

- $X_{k|k-1}$ : Tahmini durum vektörü
- $F_k$ : Durum geçiş matrisi
- $X_{k-1|k-1}$ : Önceki durumun tahmini

- $B_k$ : Giriş kontrol matrisi
- $u_k$  : Giriş kontrol vektörü
- $P_k | k - 1$ : Durum tahmini kovaryans matrisi
- $Q_k$  : Sürekli süreç hatası kovaryans matrisi

#### **Güncelleme Adımı:**

$$K_k = P_k | k - 1 \cdot H_{kT} \cdot (H_k \cdot P_k | k - 1 \cdot H_{kT} + R_k)^{-1} \quad \text{Denklem 2. 4}$$

$$X_k | k = X_k | k - 1 + K_k \cdot (z_k - H_k \cdot X_k | k - 1) \quad \text{Denklem 2. 5}$$

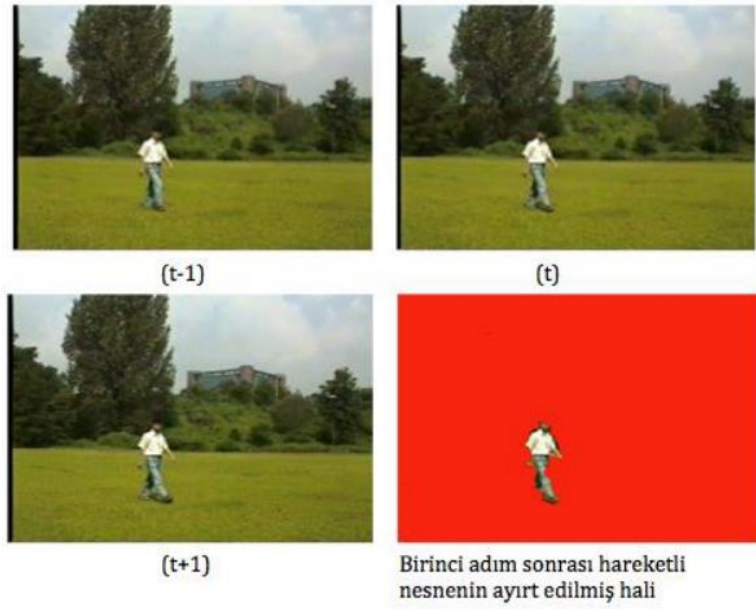
$$P_k | k = (I - K_k \cdot H_k) \cdot P_k | k - 1 \quad \text{Denklem 2. 6}$$

- $K_k$ : Kalman Kazancı
- $H_k$ : Gözlem Matrisi
- $z_k$ : Gerçek Gözlem vektörü
- $R_k$ : Gözlem Hatası Kovaryans Matrisi

Tahmin, mevcut durumu temsil eden bir tahmin yapar ve gelecekteki durumu belirler.

Güncelleme ise, gerçek Dünyadan alınan gözlemlerle mevcut tahmini günceller.

Kalman Filtresinin gücü, matematiksel temeli ve geniş uygulama alanlarındaki esnekliğinden gelmektedir. Kalman Filtresinin gerçek zamanlı uygulamalarda kullanımı **Şekil 2.7'deki** gibi açıklanabilir. Bu uygulama, Shiuh-Ku Weng ve arkadaşları tarafından yapılmıştır [12].



Şekil 2. 7 Kalman Filtresi Gerçek Zamanlı Uygulaması

Gerçekleştirilen uygulama şu temeller üzerinden yapılmıştır.

- Hareketli nesnenin, görüntüden ayırt edilmesi.
- Nesnedeki özelliklerin çıkarılması.
- Görüntüden Renk Tonu, Doygunluk, Değer (HSV) uzayına dönüştürülmesi.
- Kalman Filtresi ile nesnenin takibi.

Öncelikle karenin  $t-1$ ,  $t$  ve  $t+1$  anı seçilerek birbirinden çıkartılmış ve bölge büyütülmesi algoritması ile nesnenin ayırt edilmesi sağlanmıştır. Daha sonrasında ise Kalman Filtresi uygulanarak, nesnenin takibi sağlanmıştır. Bu metot sayesinde nesnenin, başka bir nesnenin arkasına geçmesi veya başka bir şekilde anlık olarak görüntüden kaybolması ihtimaline karşın takibin devam etmesi sağlanmıştır.

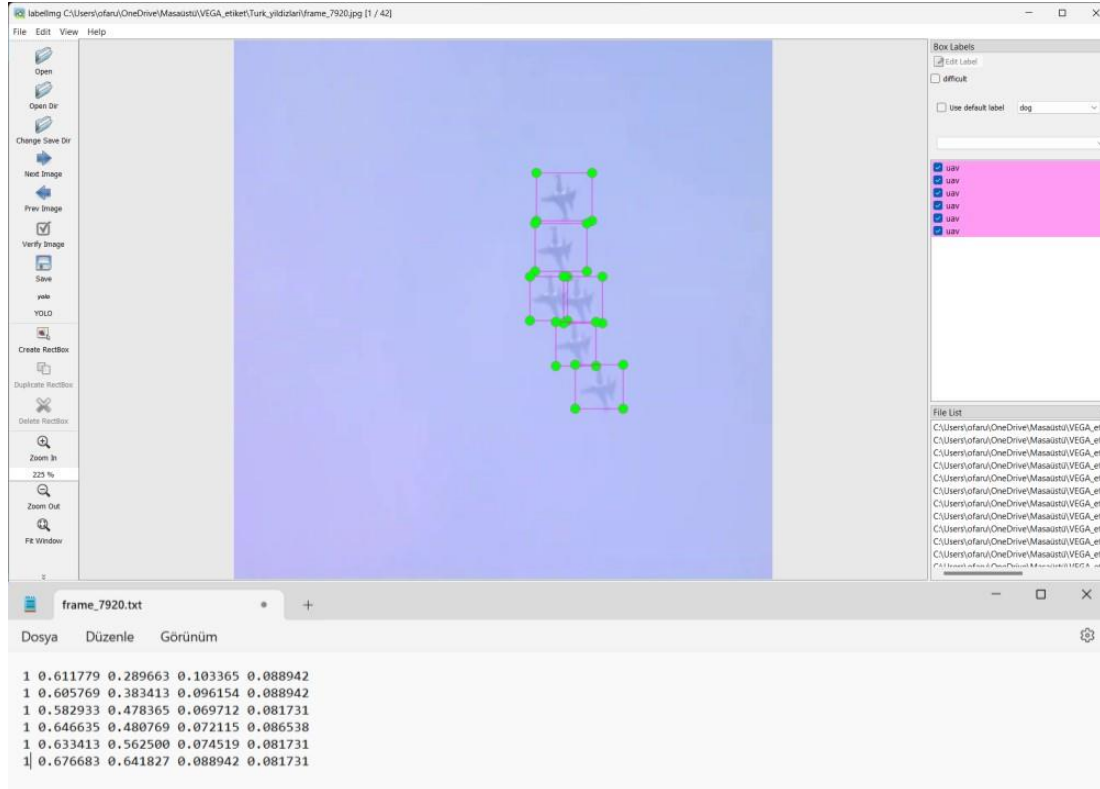
## 2.2. Tasarım (Yöntemin Problemin Çözümünde Kullanımı)

### 2.2.1. Yapay Zekâ Eğitimi ve Sistemin Tasarlanması

Öncelikle, hava araçlarının otonomisi üzerinde çalışılacağı için, 2.1’de bahsi geçen isterler doğrultusunda, kapsamlı bir veri seti oluşturulması gerekmektedir. “Roboflow” ortamında hazırlanmış olan 8080 verili “UAVSON” veri seti ana set olarak ele alınarak, üzerine basit yapıli sistemlerde kullanıma uygun olabilecek ve aynı zamanda veri seti zenginliğini arttıracak olan eklemeler yapılması planlanmıştır. Bu planlama doğrultusunda, Teknofest ve Tübitak yarışmalarının açık kaynak paylaşım verileri incelenerek, veri seti genişletme faaliyetlerine dahil edilmiştir. Bu platformlardan alınan veriler, video olarak alındığından dolayı çok fazla veriyi kapsamaktadır, bu sebeple yararlı ekleme yapmak için videolar üzerinden anlık görüntüler (frame) alınması planlanmıştır. Bu işlem, Python diliyle yazılan kod

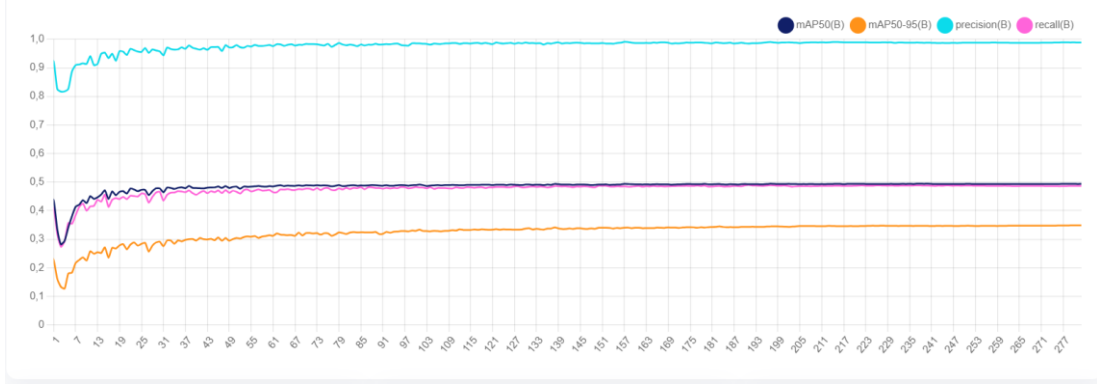
parçasıyla gerçekleştirilmiştir. Görüntü alma işlemi 720 anlık görüntüde 1 görsel alınacak şekilde ayarlanmıştır bu sayede her ortamdan belirli aralıklarla görüntü çekme işlemi gerçekleştirilmiştir.

Alınan bu görsellerin veri setine dahil edilmeden önce etiket ataması yapılması gerektiğinden “LabelImg” ortamında, etiketlenmesi planlanmış ve gerçekleştirilmiştir. Bu sürece ait örnek çalışma **Şekil 2.8’de** gösterilmiştir.



Şekil 2. 8 Veri Seti Etiketleme İşlemi

Görsellerin etiketlenmesinin tamamlanmasının ardından, veri seti eğitim, validasyon ve test olarak 3 parça halinde ele alınmış, eğitime hazır hale getirilmiştir. Hazırlanmış olan veri seti, “Ultralytics” ortamına aktarılarak burada tanımlanmış ve Bölüm 2’de anlatılan kriterlere göre seçilmiş olan Yolo v8 medium paketi ile eğitilmiştir. Eğitim çıktıları **Şekil 2.9’da** görüldüğü gibidir.



Şekil 2. 9 Doğrulama Setinde Ölçülen Modelin Doğruluk Grafiği

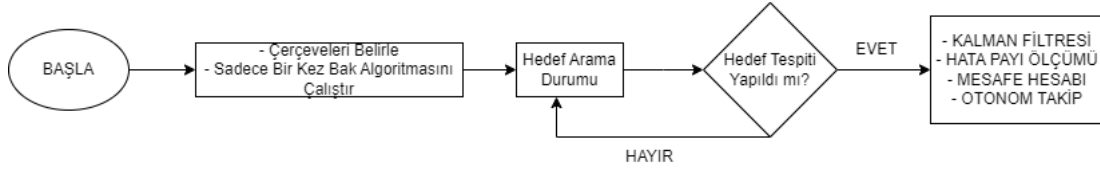
Buradaki veriler şu anlama gelmektedir:

- mAP50(B): “Mean Average Precision at 50 IoU” anlamına gelen bu metrik, nesne algılama modelinin %50 izleme oranında ortalama hassasiyetini ölçmektedir.
- mAP50-95(B): “Mean Average Precision at between 50-95 IoU” anlamındaki bu metrik, %50 ile %90 arasındaki farklı izleme oranları için ortalama hassasiyeti ifade etmektedir. Modelin ne kadar iyi çalıştığını daha kapsamlı bir şekilde gösterir.
- Precision(B): Bu metrik, modelin tahmin ettiği pozitif örneklerin gerçekten pozitif olduğu orandır. Yani, modelin ne kadar doğru tahmin yaptığını göstermektedir.
- Recall(B): Bu metrik, gerçek pozitiflerin, tüm pozitiflere oranını gösterir. Yani, modelin ne kadar gerçek pozitif kaçırmadan tespit yaptığı bu metriktir kontrol edilebilir.

Eğitim, süreci tamamlandıktan sonra “pt” uzantılı ağırlık dosyası elde edilmiştir. Bu dosya, süreç boyunca nesne tespiti işlemlerinde kaynak dosyası olarak kullanılacaktır. Yapay Zekâ (YZ) modelleri, genellikle karmaşık yapıya sahip ve bir dizi matematiksel parametreyi içeren öğrenme süreçlerini gerçekleştiren sistemlerdir. Bu modeller, eğitim sürecinde bir veri kümesi üzerinde iteratif olarak çalışırken, belirli bir görevi öğrenir ve optimize edilmiş ağırlıklar elde eder. Bu ağırlıklar, modelin performansını belirleyen önemli parametrelerdir. Eğitim sürecinin sonunda, modelin öğrenilmiş bilgilerini korumak ve gelecekte kullanılmak üzere, bu optimize edilmiş ağırlıklar bir ağırlık dosyasında saklanır. Ağırlık dosyası, bir modelin mimarisine ve kullanılan derin öğrenme çerçevesine özgü bir formatta bilgiyi içermektedir. Yani, etiketleme

boyunca tanıtılan tüm detaylar, çıktı olarak alınan ağırlık dosyası üzerinde saklanmaktadır.

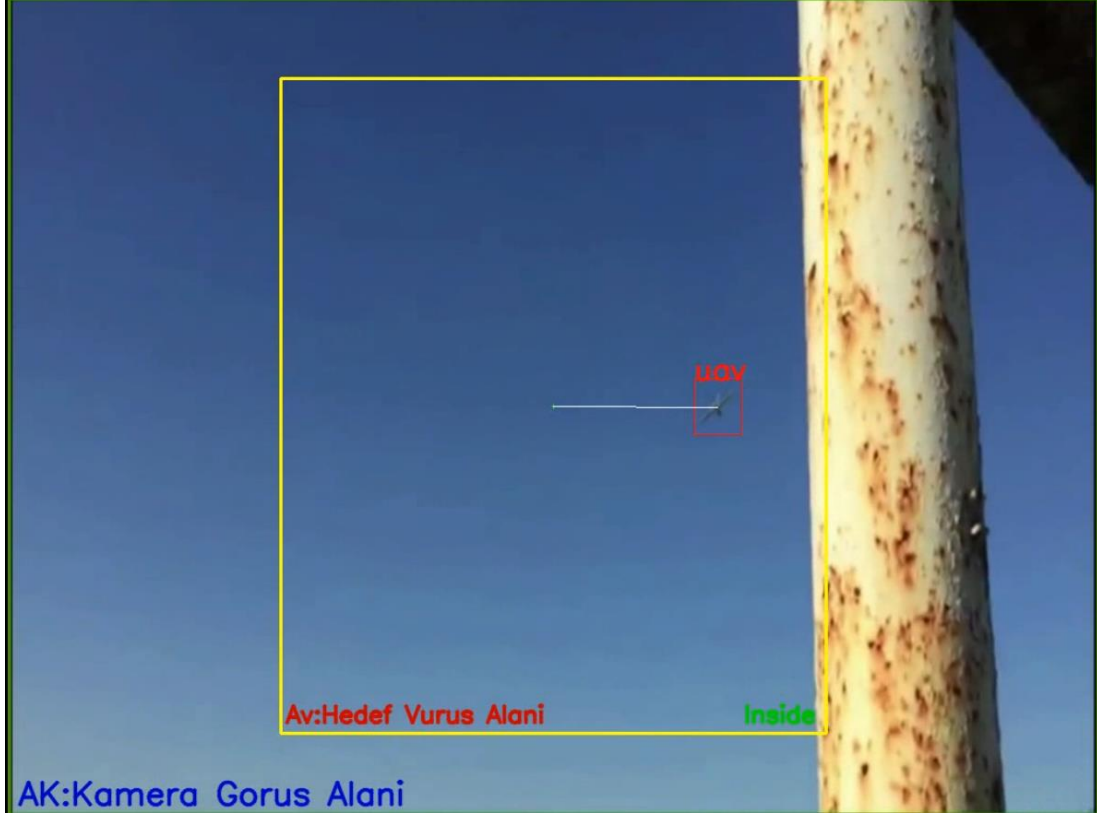
Eğitim işlemi tamamlandıktan sonra, sistem mimarisinin tasarımı olarak akış diyagramı geliştirilmiştir. Sistem algoritmasına ait olan akış diyagramı **Şekil 2.10'da** verilmiştir.



Şekil 2. 10 Sistem Mimarisi Akış Diyagramı

### 2.2.2. Sistem Tespit Mimarisinin Geliştirilmesi

**Şekil 2.10'da** verilen akış diyagramı üzerine geliştirilecek olan yazılım sistem tespit mimarisinde öncelikli olarak nesnenin saptanması ve doğru bir şekilde işaretlenmesi ele alınmıştır. Açık Bilgisayar Görüsü (OPENCV) kütüphanesi ve Sadece Bir Kez Bak (YOLO) algoritması kullanılarak basit bir nesne tanıma yazılımı geliştirilerek ilk aşama tamamlanmıştır. Sonrasında ise hava aracının rakip hava araçlarına saldırı manevrasında kullanabileceği çerçeve eklentisi planlanmış, hedeflenmeyi sağlayacak olan bu eklenti Açık Bilgisayar Görüsü kütüphanesinin özellikleri kullanarak tasarlanmıştır. Bu aşamaya dair çıktı **Şekil 2.11'de** verilmiştir.



Şekil 2. 11 Nesne Tespit Yazılımı ve Hedef Çerçevesi Eklentisi

Bu aşamadaki temel prensip, kameraya bağlanarak, görüntü oluşturulmasından başlamaktadır. Görüntü oluşturulurken ilk olarak, Hedef Vuruş Alanı ve Kamera Görüş Alanı bloğu oluşmakta ve kontrol panelinin merkezinden bir takip dairesi çizilmektedir. Daha sonra döngü içerisinde, nesne tespit işlemi başlatılmakta ve kameranın açısı doğrultusunda rakip/düşman İnsansız Hava Aracı hedefleri aranmaktadır. Hedefin tespit edilmesi durumunda, tespit edilen hava aracı tespit karesi içine alınmaktadır. Bu kare çevreleyici kutu (bounding box) olarak da geçmektedir.

Sistemin aşamaları Bölüm 3.2’de detaylı olarak verilmiştir.

### 2.2.3. Sistem Takip Mimarisinin Geliştirilmesi

Yazılım sistem takip mimarisinin geliştirilmesinde öncelikli olarak, takibin vektörel olarak yapılması sağlanmıştır. Bunun için geliştirilen sistemde, Hedef tespitinin kopmaması durumunda ekran merkezindeki takip çemberinden, hedef hava aracının merkezine (karelemenin merkezine) doğru takip vektörü çizilmektedir. Bu takip vektörü Şekil 2.10’da görülmektedir.

Kalman Filtresi, gürültülü ve kesinlik içermeyen bir veriden, gerçeğe yakın bir tahminde bulunma işlemini gerçekleştiren bir algoritmadır. Yani sistemin önceki durumlarına göre bir sonraki durumu tahmin etmek amacıyla kullanılan Kalman Filtresi, 2.1’de de belirtildiği gibi  $t$  ve  $t-1$  zamanının verilerini ele alarak  $t+1$  zamanındaki hareketi tahmin etmeye çalışır. Burada matematiksel model, “input” dediğimiz giriş değer ve “output” dediğimiz çıkış değeri barındıran bir fonksiyonu ifade etmektedir. Bu fonksiyonun oluşturulup, mümkün olan en yakın modeli tahmin etmek Kalman Filtresinin başlıca prensibidir.

Takip mimarisinde, başlangıç koşulu olarak bir Kalman Matrisi oluşturularak, sistem bunun üzerine inşa edilmiştir.

Sistemin aşamaları Bölüm 3.2’de detaylı olarak verilmiştir



## BÖLÜM 3. UYGULAMA ÇALIŞMALARI

Bu bölümde, geliştirilen sisteme dair aşamalar ve alt bileşenler verilmiştir. Sistemin, kurgusundan tamamlanmasına kadar olan tüm detaylar ve sistem geliştirme süresince kullanılan araçlar aktarılmıştır.

### 3.1. Uygulamada Kullanılan Araç ve Gereçler

Yazılım platformu, Tümlşik Geliştirme Ortamı (IDE) olarak “VSCode” uygulaması kullanılmıştır. VSCode uygulaması indirildikten sonra “Python” eklentisi kurulmuş ve tema ayarları yapılmıştır.

Sonrasında, VSCode üzerinden terminale erişilerek uygulamada kullanılacak olan kütüphaneler indirilmiştir. Bu kütüphaneler Açık Bilgisayar Görüsü (OpenCV), “Numpy”, “Ultralytics”, Bilgisayar Görüsü Alanı (CVZone), Matematik (Math), Matplotlib, Zaman (Time) ve Torch kütüphaneleridir. Bu kütüphanelerin kullanım amacı şu şekildedir:

- **Açık Bilgisayar Görüsü (OpenCV):** Görüntü işleme ve bilgisayar görüşü görevlerini gerçekleştirmek için kullanılan, görüntü üzerinde çeşitli işlemlere imkân tanıyan açık kaynaklı bir kütüphanedir.
- **Numpy:** Sayısal hesaplamalar ve büyük veri kümeleri üzerinde verimli işlemler yapmak için kullanılan güçlü bir Python kütüphanesidir.
- **Ultralytics:** Yüksek performanslı, gerçek zamanlı nesne tespiti ve izleme uygulamaları geliştirmek için kullanılan bir bilgisayar görüşü kütüphanesidir. Sadece Bir Kez Bak Algoritması, sisteme bu kütüphane aracılığıyla eklenebilmektedir.
- **Bilgisayar Görüsü Alanı (CVZone):** Bilgisayar görüşü projeleri için kullanıcı dostu bir API sağlayarak çeşitli görsel işleme görevlerini basitleştiren bir kütüphanedir. Açık Bilgisayar Görüsü Kütüphanesinin üzerine inşa edilerek geliştirilmiştir. Bu kütüphane ile görüntü üzerine basit eklentiler eklemek mümkündür.
- **Matematik (Math):** Temel matematiksel işlemler ve hesaplamalar yapmak için kullanılan standart Python kütüphanesidir.
- **Matplotlib:** Veri görselleştirme ve grafik çizimi için kullanılan kapsamlı ve esnek bir Python kütüphanesidir.

- **Zaman (Time):** Zamanla ilgili işlemleri gerçekleştirmek, zamanı ölçmek ve biçimlendirmek için kullanılan standart Python kütüphanesidir.
- **Torch:** Makine öğrenimi ve derin öğrenme modelleri oluşturmak ve eğitmek için kullanılan güçlü bir açık kaynaklı derin öğrenme kütüphanesidir. Bu kütüphane aracılığıyla, Grafik İşlemci Birimi (GPU) desteği bulunan ana makinelerde yüksek performansta görüntü işleme yazılımları çalıştırılabilmektedir.

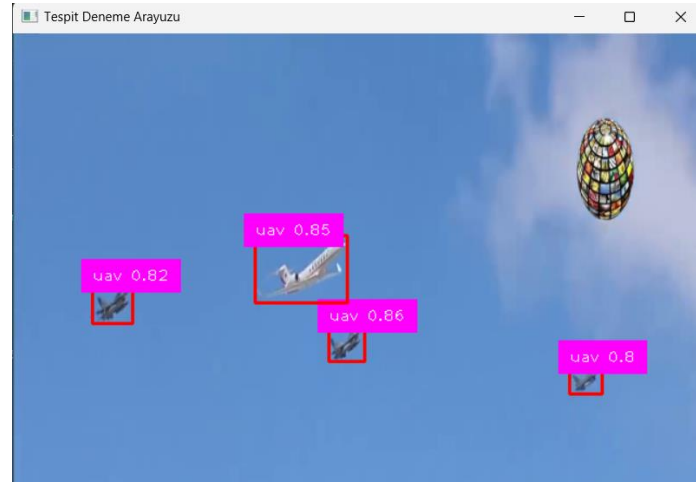
Değişken atamalarının gerçekleştirilmesinin ardından temel algoritma tek bir While döngüsü üzerinden tasarlanmış, bu döngü içerisinde if/else koşul blokları ile görev kontrolleri eklenmiştir.

Yazılım geliştirme aşamasında eğitim aracı olarak Roboflow, Ultralytics ve Colab ortamları kullanılmıştır. Roboflow yapılan etiketlemeler sonrası, Ultralytics üzerinden veri seti düzenlenmiş ve Colab ortamında eğitim tamamlanmıştır. Eğitim çıktıları doğrudan kaydedilme yöntemi ile diske alınarak, VSCode yazılım geliştirme ortamına hazır hale getirilmiştir.

### 3.2. Uygulamanın Gerçekleştirilme Aşamaları

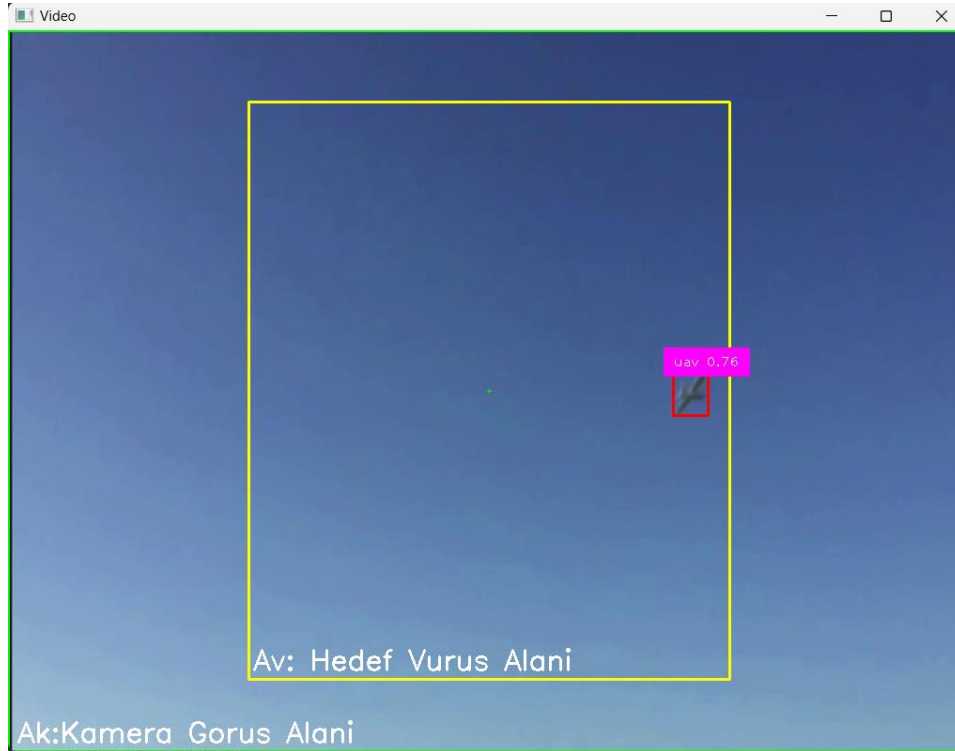
Geliştirilen uygulama, İnsansız Hava Araçlarında otonom tespit ve takip senaryosu üzerine kurgulanmıştır. Hava-Hava muharebe senaryosu ele alındığında, insansız hava araçlarının hava sahasında bulunan rakip/düşman hedefleri tespit ederek, bunlara hizalanması ve bu hava araçlarını hedef haline getirmesi sistemin temel çalışma prensibidir.

Öncelikli olarak Hava Aracı tespit senaryosu üzerine çalışılmış ve YOLO v8s paketi ile geliştirilen algoritma ile optimum başarı oranında yer alan ağırlık dosyası kullanılarak hava aracı tespit yazılımı geliştirilmiştir. Tespit yazılımı, **Şekil 3.1’de** görüldüğü gibi, uçuş videosundan alınan örneklerle tekli ve çoklu hava aracı videosu üzerinden teste tabi tutulmuştur.



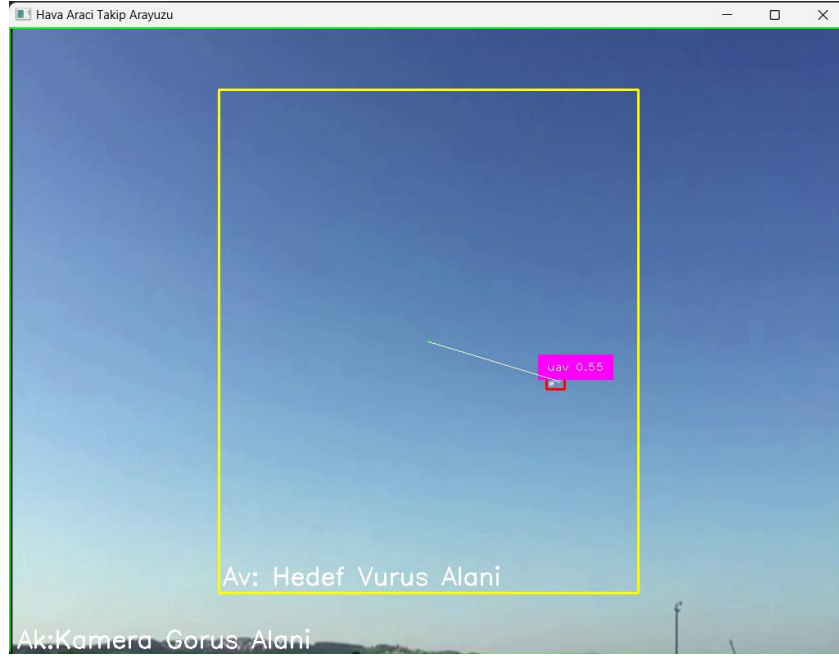
Şekil 3. 1 Aşama 1 - Hava Aracı Tespiti

Sonrasında hava aracının hedeflenme mekanizmasının yazılımsal kontrolü için, hedef tespit arayüzü geliştirilmiştir. Bu aşamada, hedefin saptandıktan sonra, hedefe kilitlenilmesi ve atış pozisyonu alınabilmesi için çerçeveleme algoritması geliştirilerek ana yazılıma eklenmiştir. Daha önce **Şekil 2.11**'de tanıtıldığı gibi kamera görüş alanı ve hedef vuruş alanı tasarlanmış, rakip hava aracının bu şekilde hedeflenmesi kurgulanmıştır. **Şekil 3.2**'de gösterilmiştir.



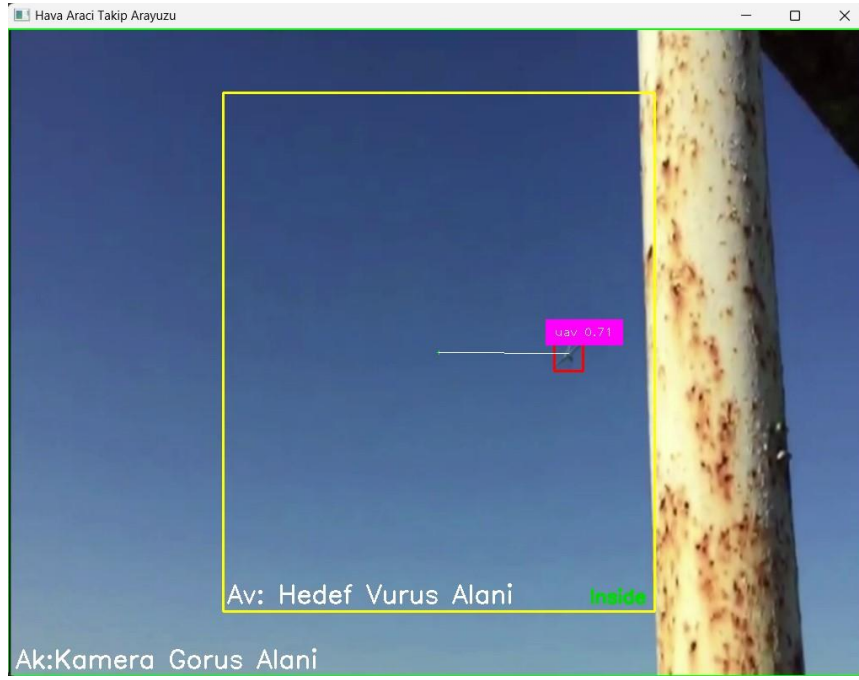
Şekil 3. 2 Aşama 2 – Hedef Kilitlenme Çerçevesi

Tespit ve Hedef Kilitlenme çerçevelerinin geliştirilmesinden sonra, tanınan hava aracının takibini sağlamak amacıyla takip vektörü oluşturulmuştur. Bu takip vektörü, merkez tespit noktasından, tespit edilen rakip hava aracının merkez hedef noktasına yöneltilen bir takip kontrol çizgisidir. Rakip Hava aracı merkez hedef noktasının tespiti ile bu işlem gerçekleştirilmektedir. Tasarlanan bu işlem **Şekil 3.3**'te verilmiştir.



*Şekil 3. 3 Aşama 3 – Hedef Kilitlenme Vektörü*

Takip İşleminin düzenli halde gerçekleştirilmesi için, hedefin hedef vuruş alanında olup olmadığının kontrolü yapılması gerekmektedir. Bu amaçla, tespiti yapılan hava aracının konum tespiti için bir algoritma gerçekleştirilmiş ve ana yazılıma dahil edilmiştir. Bu algoritma, tespit edilen hava aracının sarı renk ile çizdirilmiş iç kare bölgedeki, hedef vuruş alanına göre konumunu tespit ederek bilgi verir.



Şekil 3. 4 Aşama 4 – Hedef – Hedef Vuruş Alanı Kontrolü

Şekil 3.4’te görüldüğü üzere, tespiti yapılan rakip hava aracının bölge kontrolü yapılmış ve içeride (inside) olarak hesaplanan konum ekranın sağ alt bölgesine yazdırılmıştır. Tespiti sağlanan hava aracının dışarıda olması durumunda ise dışarıda (outside) yazdırılmaktadır.

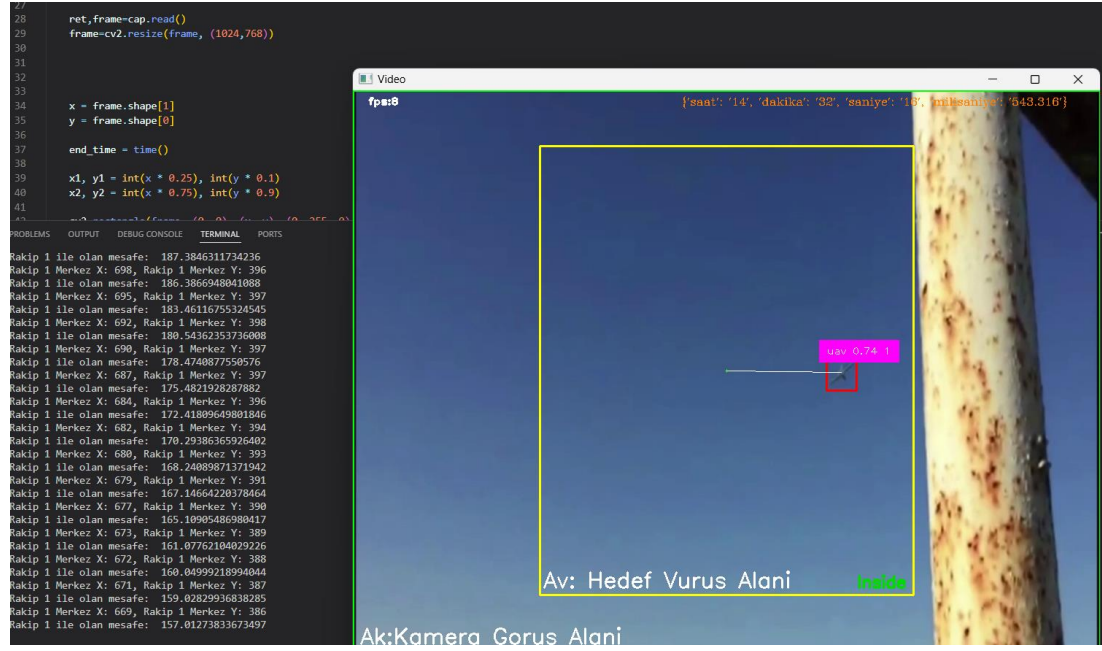
Tespit ve Takibe dayalı yazılımın optimum sonucu elde edebilmesi için, hava aracının rakip/düşman tespiti yaptığı farklı hava araçlarının kameraya göre konumunu bilmesi ve bu konumun gelecekte ne yöne doğru değişebileceğini tahmin etmesi, proje kapsamında elzem bir konudur. Şayet, görüntü üzerindeki konumun bilindiği takdirde, araç önu kameranın sabit olmasının etkisiyle görüş açısı nettir ve bu görüş açısı hava aracının hareketi ile birlikte değişmektedir. Bu durum ele alındığında, kameranın rakip hava aracını kamera ile takip etmesi, mevcut hava aracının rakip hava aracını takip etmesi anlamında gelmektedir.

Sadece Bir Kez Bak Algoritması kullanılarak tespiti yapılan hava aracının merkezine odaklı çevreleyici kutu çizildikten sonra, bu çevreleyici kutunun merkezi esas alınarak rakip hava aracının koordinatlarının alınma işlemi sağlanmıştır. Sonrasında ise rakip konumu ile kamera merkez konumu arasındaki mesafe Öklid Uzaklığı (**Denklem 3.1**) kullanılarak hesaplanmıştır.

$$d(x,y) = \sqrt{(x1 - x2)^2 + (y1 - y2)^2}$$

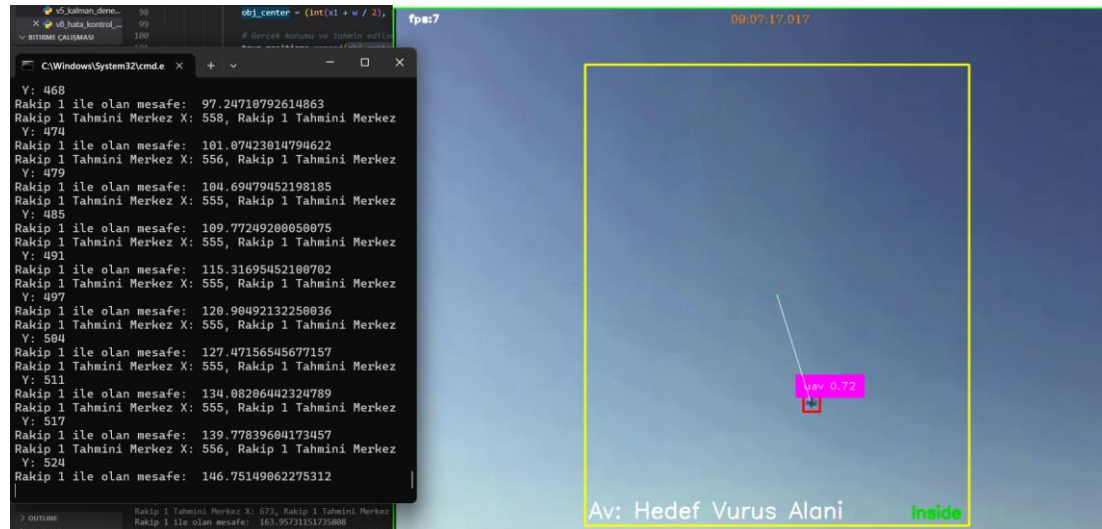
Denklem 3. 1 Öklid Uzaklığı

Bu işlemin ana algoritmaya eklenmesi, Şekil 3.5'te verildiği gibidir



Şekil 3. 5 Aşama 5 - Rakip Koordinat ve Mesafe Ölçümünün Kontrolü

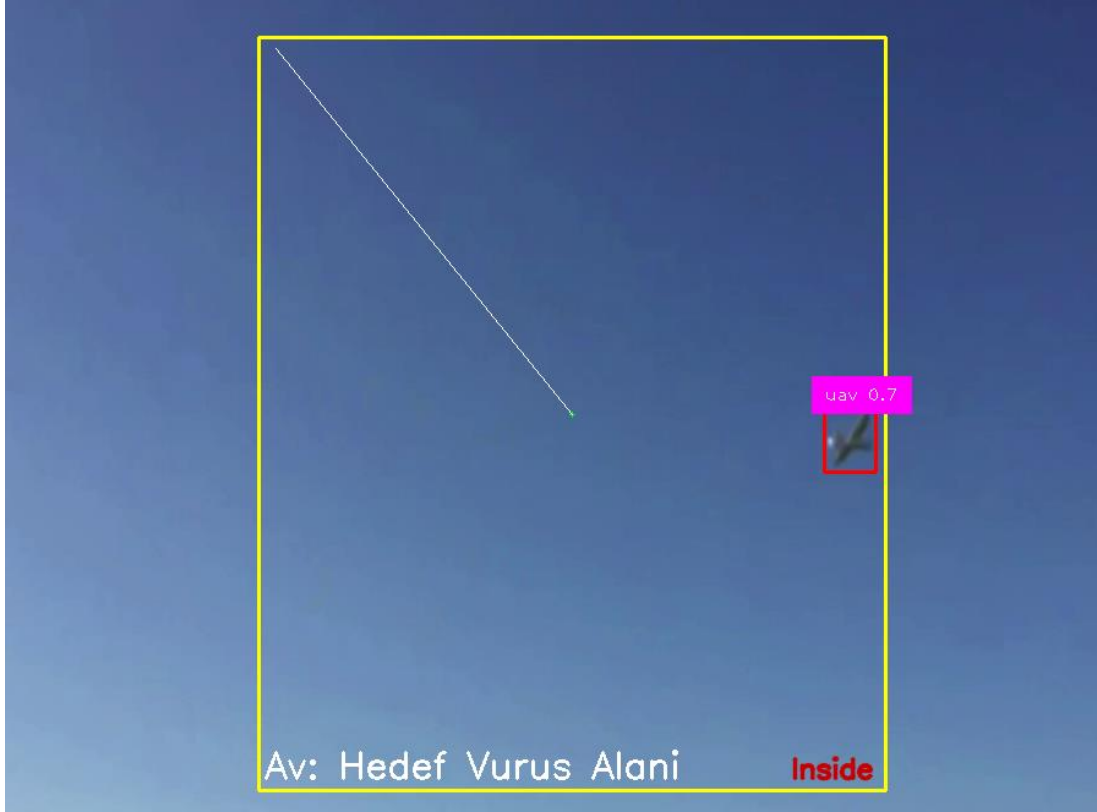
Tespit işlevlerinin geliştirilmesinden sonra, takip işlevinin fonksiyonel hale getirilmesi için öncelikli olarak Kalman Filtresi eklenmiştir. Sonrasında eklenen bu Kalman Filtresine tahmin yaptırılarak, daha önce tespit merkezine çizilmekte olan takip vektörü, bu tahmin eksenine kaydırılmıştır. Yeni durumda mesafe ölçümü ise yeni takip vektörüne göre yani Kalman Tahmini ile Merkez arasına göre hesaplanmaktadır. Bu duruma ait uygulama çalışması Şekilde 3.6'da verilmiştir.



Şekil 3. 6 Kalman Filtresinin Uygulamaya Eklenmesi

**Şekil 3.6'da** görüldüğü gibi ekranın merkezinden tahmin edilen noktaya takip vektörü çizilmektedir. Bu vektörün uzunluğu ve tahmin edilen koordinatlar terminal ekranından takip edilmektedir.

Kalman Filtresi girdi olarak  $t$  ve  $t-1$  sürelerini alarak  $t+1$  süresindeki hareketi tahmin etmeye odaklı bir sistem olduğundan dolayı başlangıç anında sapmalar

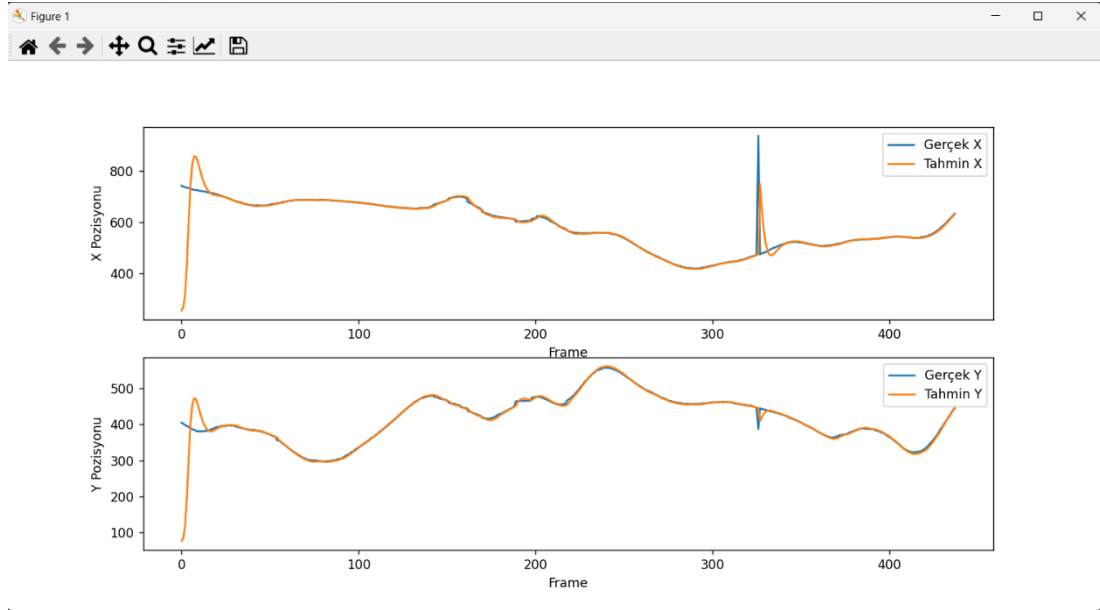


*Şekil 3. 7 Kalman Başlangıç Koşulu*

gerçekleşmektedir. 0 (Sıfır) anında başlangıç verisi olmadığından dolayı tahmin mekanizması sağlıklı çalışmayacağından, sisteme bir başlangıç koşulu atanması gerekmektedir. Bu sebeple başlangıç vektör noktası, Hedef Vuruş Alanının sol üst köşesi olarak ayarlanmıştır. Bu duruma ait görsel **Şekil 3.7'de** verilmiştir.

Başlangıç Koşulunun atanmasıyla birlikte Kalman Tahmin Algoritmasının çalışma prensipleri gerçekleştirilmiştir. Bu aşamadan sonra uygulamanın performansı test

edilmiştir. Teste dair ilk aşama olan hata payının hesaplanması Şekil 3.8’de verilmiştir.



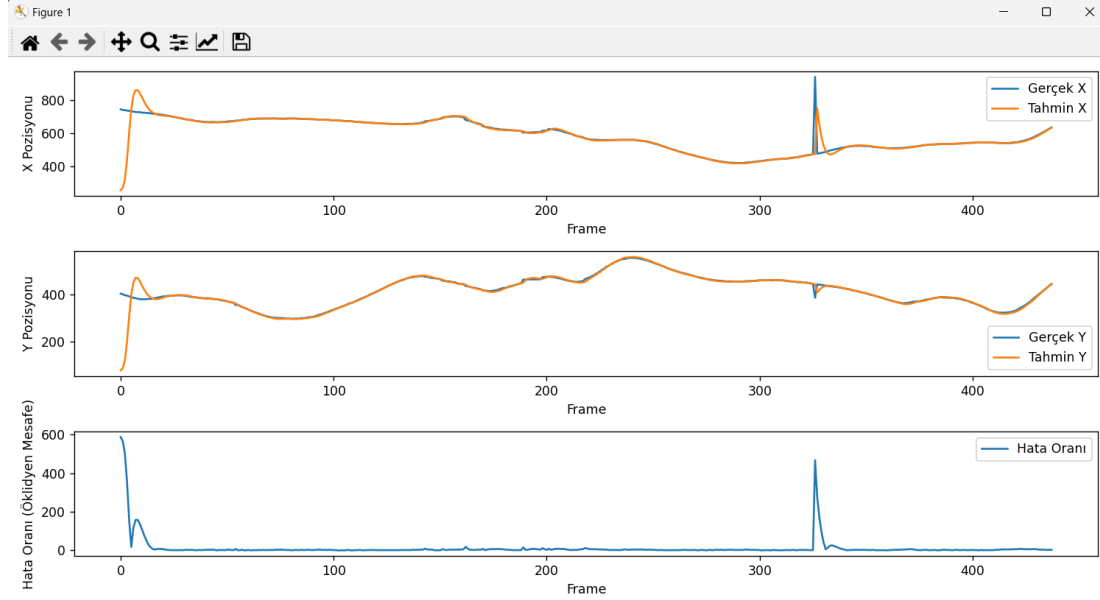
Şekil 3. 8 Kalman Filtresi Hata Kontrol Grafiği

Burada, Kalman Filtresi tarafından tahmin edilen konum değerleri ile takibi yapılan hava aracının gerçek konum bilgileri kıyaslanmış ve grafikleştirilmiştir. Başlangıç koşulunda, başlangıç değeri atandığı için gerçek değer bilinmemekte, Kalman



başlangıç değeri görülmektedir. Sistemin tahminin bozulma gösterdiği pozisyonda ise görüntü sırasında kopma saptanmıştır.

Hata payının daha detaylı ölçülmesi ve bir panel üzerinden takip edilmesi planlanarak hata payına eklemeler yapılması planlanmıştır. Eklentilerle birlikte tasarlanan yeni hata payı paneli **Şekil 3.9**'da gösterilmiştir.



Şekil 3. 9 Hata Kontrol Paneli

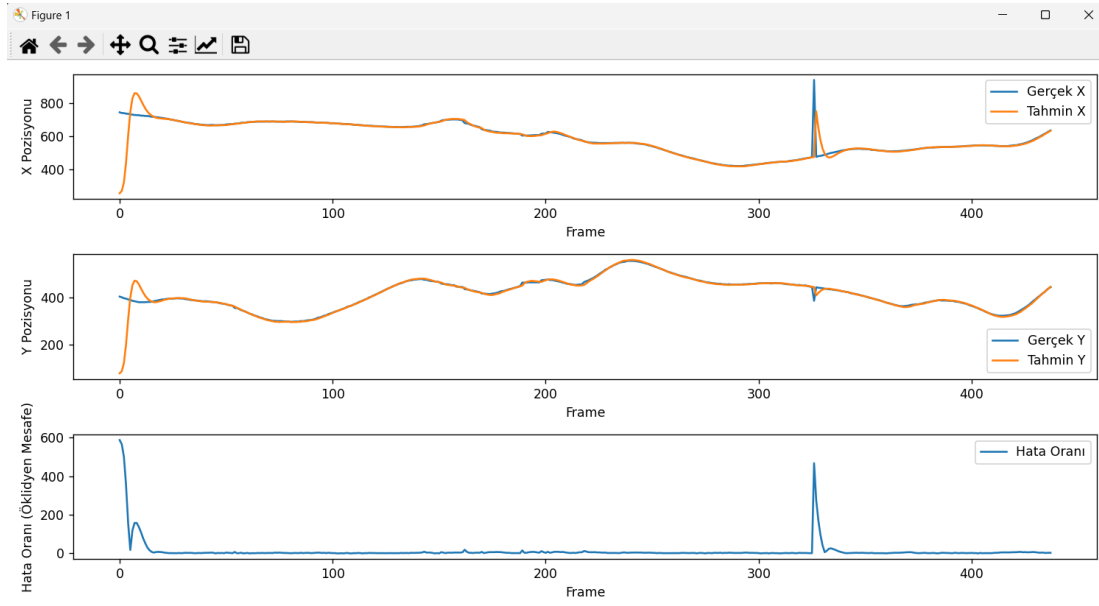
Hata Kontrol Panelinin ilk iki kısmında gerçek ve tahmin edilen değerlerin X ve Y koordinatında konumları kıyaslanmıştır. Üçüncü bölümde ise gerçek konum ve tahmini konum arasındaki hata oranı Öklid Denklemi ile hesaplanmıştır. Hesaplama, **Denklem 3.2**'ye göre yapılmıştır.

$$d(x, y) = \sqrt{(X_{tahmin} - X_{gercek})^2 + (Y_{tahmin} - Y_{gercek})^2} \quad \text{Denklem 3. 2}$$

### 3.3. Uygulama Sonuçları ve Yorumlanması

Uygulamada başarı ölçütleri tahmin edilen konum ile gerçek konumun örtüşmesi ve takibin kesintisiz olarak sağlanmasıdır. Uygulama süresince 1.9'da verilen kısıtlar da ele alınarak çeşitli denemeler yapılmış ve sonuçlar test edilmiştir. Farklı Hava Durumlarına ait denemeler aşağıda verilmiştir.

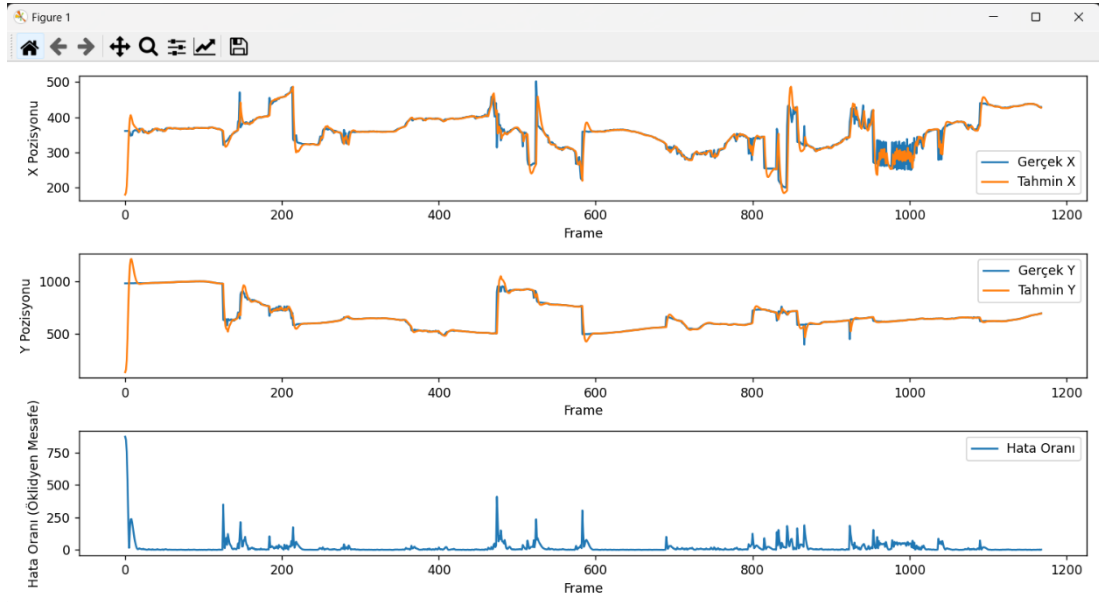
Güneşli ve parlak hava durumuna göre yapılan test **Şekil 3.10**'da verilmiştir.



Şekil 3. 10 Güneşli Hava Test Uygulama Sonucu

Bu test sonucu, başarı kriterlerini yüksek oranda kapsamaktadır. Güneş ışınlarının aşırı parlak olarak vurduğu bazı anlarda kameranın görüntü kalitesi bozulduğundan dolayı anlık parlamalar ile birlikte nesne tespitinde sapmalar yaşanmıştır. Bu da Kalman Filtresinin farklı nesneye odaklanmasına sebep olmuştur.

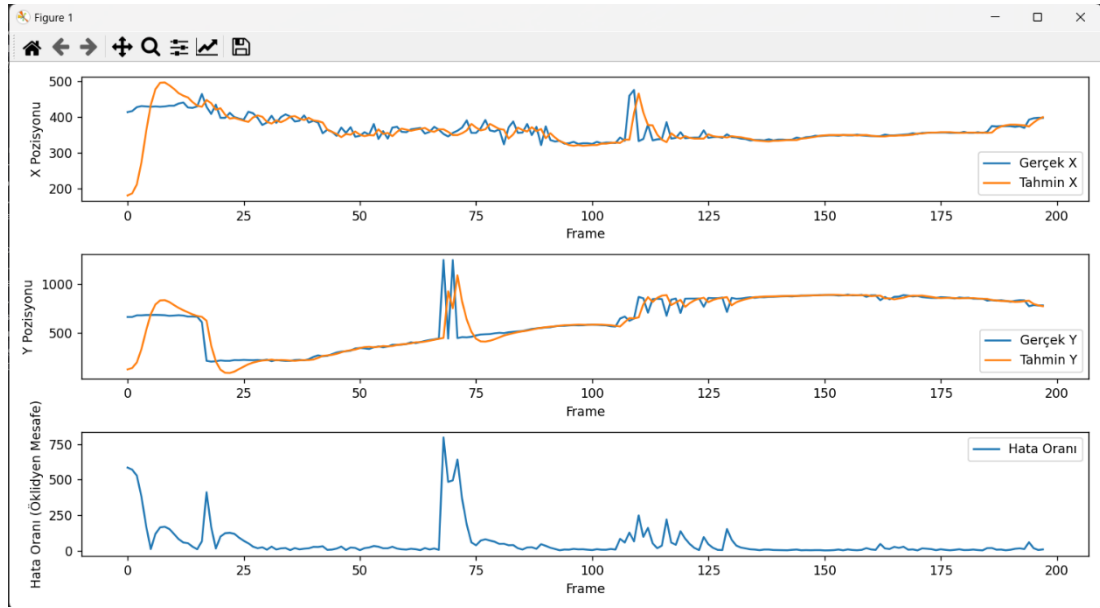
Bulutlu ve parlak olmayan hava şartlarında uygulanan test Şekil 3.11’de verilmiştir.



Şekil 3. 11 Bulutlu Hava Test Uygulama Sonucu

Bu durumda takibin sağlandığı anların hiçbirinde sapma yaşanmamıştır, hava aracının ani manevraları konum değişikliklerinde görünmektedir. Bu manevralara rağmen takip fonksiyonu yüksek hata payı barındırmamaktadır.

Yağmurlu hava şartlarında denenmiş olan uygulama Şekil 3.12’de verilmiştir.



Şekil 3. 12 Yağmurlu Hava Uygulama Sonucu

Bu test yağmurlu havada ve oldukça hareketli bir döner kanat insansız hava aracıyla gerçekleştirilmiştir. X konumunda önemli sapmalar yaşanmamış olsa da Y konumunda sapmalara rastlanmıştır.

Bu bölümünde tüm çalışmanın sonucunda elde edilen bulgular yorumlanarak çeşitli öneriler sunulmuştur. Çalışma sonuçlarının gerçekçi kısıtlar açısından analizi yapılmıştır.

## BÖLÜM 4. SONUÇLAR VE ÖNERİLER

### 4.1. Sonuçlar

- Çalışma boyunca teorik hesaplamalar ile uygulama sonuçları arasında görülen en önemli sorun hava muhalefetinden dolayı yaşanan problemler olmuştur. Parlak olmayan havada sistem tam performansta çalışırken, parlak veya yağmurlu havada sapmalara uğramıştır.
- Teorik çalışmada yüksek görüntü performansı beklenirken, grafik işlemci biriminin olmadığı durumda düşük görüntü performansı alınmıştır. Görüntü, merkezi işlemci birimine alınarak işlenmiş bu da sistemin yavaş çalışmasına neden olmuştur.
- Ağırlık dosyasının temelini oluşturan veri seti hazırlanırken, yarışma ve festival özelinde genel kullanıma uygun hava araçları kullanılmıştır, telif

hakları sebebiyle özel hava araçları eklenememiştir. Bu da sistemin nesne tespit performansına doğrudan etki etmiştir.

#### **4.2. Öneriler**

- Sistemin grafik işlemcide kullanılması, görüntü işleme performansını arttıracığından dolayı sistem için elzemdir. Bu yöntem çalışmanın performansını arttıracaktır.
- Sistemin temelinde kullanılan ağırlık dosyası oluşturulmadan önce hazırlanacak olan veri setinin gerçek dünyada karşılaşılmaması muhtemel düşman hava araçlarına göre kurgulanması durumunda nesne tespit performansı artış gösterecektir.
- Algoritmada nesne tespit işleminden önce Sadece Bir Kez Bak Algoritmasının girişine uygulanacak olan filtreler, hava şartlarında yaşanması muhtemelen olumsuzların önüne geçebilme potansiyeli taşımaktadır. Parlak veya Karanlık durumlar için görüntü filtreleri ile önlem alınabilir.

#### **4.3. Sonuçların Sağlık, Çevre ve Güvenlik Açısından Analizi**

İnsansız Hava Araçlarında hava-hava muharebe başarısının elde edilmesi şüphesiz ki güvenlik açısından önemli bir bağlam içerisinde yer almaktadır. Hava-Hava ve Hava-Kara özelliğine sahip bir insansız hava aracı, savaş uçağı niteliğine sahip olmakta ve bunu pilotsuz olarak gerçekleştirmektedir. Bu açıdan bakıldığında, güvenlik tedbirlerini maksimum seviyede tutmak ve personel güvenliğini sağlamak projenin özellikleri kapsamında yorumlanabilir.

## BÖLÜM 5. KAYNAKLAR

- [1] Ajmera, F., Meshram, S., Nemade, S., & Gaikwad, V. (2021). Survey on object detection in Aerial Imagery. *2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*. <https://doi.org/10.1109/icicv50876.2021.9388517>
- [2] Rohan, A., Rabah, M., & Kim, S. H. (2019). Convolutional Neural Network-Based Real-Time Object Detection and Tracking for Parrot AR Drone 2. *IEEE Access*, 7, 69575–69584. <https://doi.org/10.1109/access.2019.2919332>
- [3] Chen, P., Dang, Y., Liang, R., Zhu, W. ve He, X. J. I. T. o. I. T. S., 2017, Real-time object tracking on a drone with multi-inertial sensing data, 19 (1), 131-139.
- [4] Bi, Y., Duan, H. J. O.-I. J. f. L. ve Optics, E., 2013, Implementation of autonomous visual tracking and landing for a low-cost quadrotor, 124 (18), 3296-3300.
- [5] Pestana, J., Sanchez-Lopez, J. L., Campoy, P. ve Saripalli, S., 2013, Vision based gps-denied object tracking and following for unmanned aerial vehicles, *2013 IEEE international symposium on safety, security, and rescue robotics (SSRR)*, 1-6.
- [6] Weng, S.-K., Kuo, C.-M., Tu, S.-K. J. J. o. V. C. ve Representation, I., 2006, Video object tracking using adaptive Kalman filter, 17 (6), 1190-1208.
- [7] Li, X., Wang, K., Wang, W. ve Li, Y., 2010, A multiple object tracking method using Kalman filter, *The 2010 IEEE international conference on information and automation*, 1862-1866
- [8] Roberts, R., Ta, D., Straub, J., Ok, K., & Dellaert, F. (2012). Saliency detection and model-based tracking: a two part vision system for small robot navigation in forested environment. *Proceedings of SPIE*. <https://doi.org/10.1117/12.919598>
- [9] Mehmet Celalettin Ergene, İNSANSIZ HAVA ARAÇLARINDA AKILLI KONTROL SİSTEMLERİNE YÖNELİK NESNE TANIMA VE TAKİP UYGULAMALARI.
- [10] Schneiderman, H. J. C., 2004, Learning a restricted Bayesian network for object detection, 4, 639-646.
- [11] Cuevas, E. V., Zaldivar, D. ve Rojas, R., 2005, Kalman filter for vision tracking.
- [12] Weng, S.-K., Kuo, C.-M., Tu, S.-K. J. J. o. V. C. ve Representation, I., 2006, Video object tracking using adaptive Kalman filter, 17 (6), 1190-1208

## **BÖLÜM 6. ÖZ GEÇMİŞ**

### **KİŞİSEL BİLGİLER**

**Adı Soyadı:** Ömer Faruk USTA

**Uyruğu:** T.C.

**Doğum Yeri ve Tarihi:** Sakarya - 09.01.2001

**Telefon:** 0537 022 7400

**E-mail:** [ofarukusta1903@gmail.com](mailto:ofarukusta1903@gmail.com)

### **EĞİTİM**

Derece	Bölüm, Okul	Bitirme Yılı
Lise:	Tes-iş Adapazarı Anadolu Lisesi, Sakarya	2019
Üniversite:	Elektrik Elektronik Mühendisliği, Sakarya Üniversitesi	Devam(4.Sınıf)
Üniversite:	Yönetim Bilişim Sistemleri, İstanbul Üniversitesi	Devam(4.Sınıf)

### **İŞ DENEYİMLERİ**

Yıl	Kurum, Görevi
2023-Devam	PNP Elektrik Enerji Otomasyon, Software Developer
2023	Oreltek Teknoloji, Artificial Intelligence Engineer Intern
2022	Federal Mogul Powertrain, Maintenance Engineer Intern

### **UZMANLIK ALANI**

Yapay Zekâ, Görüntü İşleme, Hava Aracı Kontrol Sistemleri.

### **YABANCI DİLLER**

- İngilizce (B2)

## BİTİRME ÇALIŞMASI SON KONTROL FORMU

		EVET	HAYIR
1	Beyan formu çalışmada bulunuyor mu?	<input type="checkbox"/>	<input type="checkbox"/>
2	Özet bölümü en az 150 kelime içermekte mi?	<input type="checkbox"/>	<input type="checkbox"/>
3	Simge ve Kısaltmalar uygun şekilde yazılmış mı?	<input type="checkbox"/>	<input type="checkbox"/>
4	Şekiller listesi formata uygun şekilde hazırlanmış mı?	<input type="checkbox"/>	<input type="checkbox"/>
5	Tablolar listesi formata uygun şekilde hazırlanmış mı?	<input type="checkbox"/>	<input type="checkbox"/>
6	İş paketleri ve zaman çizelgesi çalışmada verilmiş mi?	<input type="checkbox"/>	<input type="checkbox"/>
7	Başarı ölçütlerinden bahsedilmiş mi?	<input type="checkbox"/>	<input type="checkbox"/>
8	Risk yönetimi ve planlaması bulunmakta mı?	<input type="checkbox"/>	<input type="checkbox"/>
9	Sonuçların sağlık, çevre ve güvenlik açısından analizi gerçekleştirilmiş mi?	<input type="checkbox"/>	<input type="checkbox"/>
10	Deney Tasarımı Açıklamaları verilmiş mi?	<input type="checkbox"/>	<input type="checkbox"/>
11	IEEE Etik kurallar onay formu imzalanmış mı?	<input type="checkbox"/>	<input type="checkbox"/>
12	Çalışmanın Ulusal ya da Uluslararası standartlar ilişkisi verilmiş mi?	<input type="checkbox"/>	<input type="checkbox"/>
13	Seminer, Kongre, Konferans veya Teknik Gezi katılım kanıtı mevcut mu?	<input type="checkbox"/>	<input type="checkbox"/>
14	Kaynaklar bölümü formata uygun hazırlanmış mı?	<input type="checkbox"/>	<input type="checkbox"/>
15	Yazım kılavuzuna aykırı durumlar mevcut mu?	<input type="checkbox"/>	<input type="checkbox"/>

Bu çalışma tarafımdan incelenmiş olup kontrol sonuçları yukarıda verildiği gibidir.

İnceleyen :

Tarih : 2024-06-11

İmza :