



KONYA  
TEKNİK ÜNİVERSİTESİ

**T.C.**  
**KONYA TEKNİK ÜNİVERSİTESİ**  
**LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ**



**İNSANSIZ HAVA ARAÇLARINDA AKILLI  
KONTROL SİSTEMLERİNE YÖNELİK NESNE  
TANIMA VE TAKİP UYGULAMALARI**

**Mehmet Celalettin ERGENE**

**YÜKSEK LİSANS TEZİ**

**Elektrik - Elektronik Mühendisliği Anabilim Dalı**

**Temmuz – 2019**  
**KONYA**  
**Her Hakkı Saklıdır**

## TEZ KABUL VE ONAYI

Mehmet Celalettin ERGENE tarafından hazırlanan “İnsansız Hava Araçlarında Akıllı Kontrol Sistemlerine Yönelik Nesne Takip Uygulamaları” adlı tez çalışması 22/07/2019 tarihinde aşağıdaki jüri tarafından oy birliği ile Konya Teknik Üniversitesi Lisansüstü Eğitim Enstitüsü Elektrik - Elektronik Mühendisliği Anabilim Dalı'nda YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

### Jüri Üyeleri

#### Başkan

Dr. Öğr. Üyesi Murat KÖKLÜ


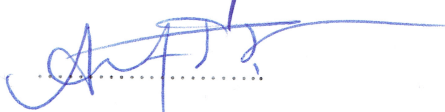

#### Danışman

Dr. Öğr. Üyesi Akif DURDU

#### Üye

Dr. Öğr. Üyesi Hakan TERZİOĞLU

### İmza

  
.....  
  
.....  
  
.....

Yukarıdaki sonucu onaylarım.

Prof. Dr. Hakan KARABÖRK  
Enstitü Müdürü

## TEZ BİLDİRİMİ

Bu tezdeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

## DECLARATION PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Mehmet Celalettin ERGENE

Tarih: 22.07.2019



## ÖZET

### YÜKSEK LİSANS TEZİ

## İNSANSIZ HAVA ARAÇLARINDA AKILLI KONTROL SİSTEMLERİNE YÖNELİK NESNE TAKİP UYGULAMALARI

**Mehmet Celalettin ERGENE**

**Konya Teknik Üniversitesi  
Lisansüstü Eğitim Enstitüsü  
Elektrik - Elektronik Mühendisliği Anabilim Dalı**

**Danışman: Dr. Öğr. Üyesi Akif DURDU**

**2019 – 64 Sayfa**

İnsansız hava araçları günümüzde sıklıkla kullanılmakta olup bunlara olan revaç ve ihtiyaç gün geçtikçe artmaktadır. Hava araçlarının insansız olması ile bu araçlarda otonom kontrol uygulamaları gereksinimi hızla artmaktadır. İnsansız hava araçları çeşitli koşullarda ve çeşitli görevler için kullanılmakta ve bu görevlerde tam otonom uçuş için akıllı kontrol uygulamaları gerekmektedir. Akıllı bir kontrol sisteminde çeşitli makine öğrenmesi, derin öğrenme ve istatistiksel öğrenme metotları gibi kontrol sistemleri geliştirilmiş ve geliştirilmeye devam etmektedir.

Bilindiği üzere insansız hava araçları oldukça hızlı hareket edebilmektedir. Engelden kaçınma veya nesne takibi gibi özelliklere sahip bir insansız hava aracı göz önüne alındığında, kontrolü sağlayan akıllı kontrol sistemi kararları oldukça hızlı ve doğru vermelidir aksi halde hava aracı ya engele çarpacak ya da yanlış karar vererek nesneyi kaybedecektir. Bunun için geliştirilecek nesne takip algoritmasının iki olmazsa olmaz özelliğe sahip olması gerekmektedir. Bu özellikler hız ve doğruluktur. Algoritmanın hızlı olması kullanılacak algoritmanın işlem gücüne ve kullanılacak olan donanımın yeterliliğine bağlıdır. Algoritmanın doğruluğu ise kullanılan metodun sınıflandırma veya takip işlemini ne kadar doğru yaptığına bağlı olacaktır. Bu nedenle, bu çalışmada literatürde halihazırda kullanılan derin öğrenme ve istatistiksel öğrenme metotları kullanılarak performansları karşılaştırılmış ve insansız hava araçlarında akıllı kontrol sistemlerine en uygun nesne takibi metodu bulunmaya çalışılmıştır. Bu çalışmaya yönelik, bir insansız hava aracının 5, 10 ve 20 metre irtifalarda hareketli ve sabit uçuşlarından alınan görüntüler üzerinde Parçacık Filtresi (PF), Kalman Filtresi (KF), Faster R-CNN ve Single Shot Multibox Detector (SSD) yöntemleri kullanılarak nesne tanıma ve nesne takibi uygulamaları gerçekleştirilmiş, sonuçları karşılaştırılmış ve insansız hava araçlarında akıllı kontrol sistemlerine yönelik en uygun nesne takibi yöntemi bulunması amaçlanmıştır.

Performans değerlendirmesi amacıyla her bir yöntemin her uçuş sırasında saniye başına kare (FPS), takip doğruluğu, GPU ve CPU kullanım yüzdeleri kıyaslanmıştır. Sonuçlar göz önüne alındığında, Faster R-CNN en yüksek başarı oranına sahipken bunu sırasıyla SSD, KF ve PF takip etmiştir. FPS oranları kıyaslandığında en iyi sonucu KF vermiş ve sırasıyla PF, SSD ve Faster R-CNN gelmiştir. FPS oranlarında SSD ve PF oldukça yakın çıkmışlardır. Kıyaslamalar sonucunda insansız hava araçları için akıllı kontrol sistemlerinde nesne takibi için kullanılmaya en uygun metot, ortalama 12 FPS ile hızı ve yakın mesafede ortalama %90 başarı oranı göz önüne alınarak SSD olarak görülmüştür. Fakat maliyetin, sistem ağırlığının ve işlem gücünün az olması gerektiği sistemlerde ise KF kullanımının uygun olacağı kararına varılmıştır.

**Anahtar Kelimeler:** Akıllı Kontrol Sistemleri, Bayes Teoremi, Derin Öğrenme, Drone, Faster R-CNN, İnsansız Hava Aracı, Kalman Filtresi, Nesne Takibi, Nesne Tanıma, Parçacık Filtresi, Single Shot Multibox Detector, SSD.

# ABSTRACT

## MS THESIS

### OBJECT TRACKING APPLICATIONS FOR INTELLIGENT CONTROL SYSTEMS IN UNMANNED AERIAL VEHICLES

**Mehmet Celalettin ERGENE**

**Konya Technical University  
Institute of Graduate Studies  
Department of Electrical and Electronics Engineering**

**Advisor: Dr. Akif DURDU**

**2019 – 64 Pages**

Unmanned aerial vehicles are frequently used nowadays and the demand for them is increasing. As the aerial vehicles are becoming unmanned, the need for autonomous control applications increases rapidly. Unmanned aerial vehicles are used in a variety of conditions and for a variety of tasks, requiring intelligent control applications for full autonomous flight. In an intelligent control system, control systems such as various machine learning, deep learning and statistical learning methods have been developed and continue to be developed.

As it is known, unmanned aerial vehicles can move very fast. When an unmanned aircraft with obstacle avoidance or object tracking is considered, the intelligent control system that provides control must make decisions very quickly and accurately, otherwise the aircraft will either hit the obstacle or lose the object by making the wrong decision. For this, the object tracking algorithm to be developed must have two essential features. These features are speed and accuracy. The speed of the algorithm depends on the computing power of the algorithm to be used and the adequacy of the hardware to be used. The accuracy of the algorithm will depend on how accurately the method used makes the classification or track the object. For this reason, in this study, the performance of deep learning and statistical learning methods used in the literature is compared and the most suitable method for object tracking for intelligent control systems in unmanned aerial vehicles is tried to find. For this study Particle Filter (PF), Kalman Filter (KF), Faster R-CNN and Single Shot Multibox Detector (SSD) are used in order to track objects during stable and moving flights on 5, 10, 20 meter altitude. After this the results were compared and the most suitable object tracking method is aimed to find.

For the purpose of performance evaluation, frames per second (FPS), tracking accuracy, GPU and CPU usage percentages of each method during each flight were compared. When the results were considered, Faster R-CNN had the highest success rate, followed by SSD, KF and PF, respectively. When FPS ratios were compared, KF yielded the best results and PF, SSD and Faster R-CNN respectively. SSD and PF were very close in FPS rates. As a result of the comparisons, the most suitable method for tracking objects in smart control systems for unmanned aerial vehicles is observed as SSD by considering 12 FPS speed and average %90 success rate on lower altitude. However, in systems where cost, system weight and computing power should be low, it is concluded that the use of KF will be appropriate.

**Keywords:** Bayesian Theorem, Deep Learning, Drone, Faster R-CNN, Kalman Filter, Object Detection, Object Tracking, Particle Filter, Single Shot Multibox Detector, Smart Control Systems, SSD, Unmanned Aerial Vehicle.

## ÖNSÖZ

Yüksek lisans tez çalışması sırasında ve hayatımın her anında, yardımlarını ve anlayışını esirgemeyerek bana sabırla destek veren ve bu günlere gelmemi borçlu olduğum aileme, tüm bölüm içi ve bölüm dışı arkadaşlarıma, çalışmalarım sırasında bana gerekli maddi ve manevi desteği veren Robosys Mekatronik Sistemleri ve Solter Teknoloji Hizmet ve Danışmanlık Ltd. Şti.'ne teşekkürlerimi sunarım.

Bu yüksek lisans tez çalışmamda değerli katkılarını ve desteklerini esirgemeyen danışmanın Sayın Dr. Öğr. Üyesi Akif DURDU'ya teşekkürlerimi sunuyorum.

Mehmet Celalettin  
ERGENE  
KONYA-2019



## İÇİNDEKİLER

ÖZET.....	iv
ABSTRACT .....	v
ÖNSÖZ.....	vi
İÇİNDEKİLER.....	vii
SİMGELER VE KISALTMALAR.....	ix
1. GİRİŞ .....	1
1.1. Dronelar.....	1
1.2. Akıllı Kontrol Sistemleri .....	3
1.2.1. Kullanıldığı alanlar.....	3
1.3. Tezin Amacı .....	5
1.4. Tezin Önemi.....	6
1.5. Tezin Kapsamı.....	8
2. LİTERATÜR TARAMASI.....	10
2.1. Akıllı Kontrol Sistemleri ile Yapılmış İnsansız Hava Aracı Uygulamaları .....	10
2.2. Nesne Tespiti ve İHA’larda Nesne Takibi Üzerine Yapılmış Uygulamalar .....	11
2.2.1 Kalman Filtresi ve Parçacık Filtresi ile İstatistiksel Öğrenme .....	11
2.2.2. Görüntü işleme ve Makine Öğrenmesi.....	13
2.2.3. Derin Öğrenme .....	14
3. MATERYAL VE YÖNTEM .....	18
3.1. Nesne takibi.....	18
3.2. İstatistiksel Öğrenme.....	20
3.2.1. Bayes Teoremi.....	22
3.2.2. Kalman Filtresi .....	24
3.2.3. Parçacık Filtresi.....	30
3.3. Derin Öğrenme .....	32

3.3.1. Regions with CNN Features (R-CNN).....	34
3.3.2. Fast R-CNN.....	35
3.3.3. Faster R-CNN.....	36
3.3.4. Mask R-CNN.....	37
3.3.5. Single Shot Multibox Detector (SSD).....	39
4. UYGULAMA VE SONUÇLAR.....	41
4.1. Parçacık Filtresi.....	43
4.2. Kalman Filtresi.....	47
4.3. Faster R-CNN.....	51
4.4. Single Shot Multibox Detector (SSD).....	55
5. SONUÇ VE TARTIŞMA.....	61
KAYNAKLAR.....	65
EKLER.....	68
ÖZGEÇMİŞ.....	70



## SİMGELER VE KISALTMALAR

### Simgeler

$X_k$	: Durum Vektörü
$Z_k$	: Ölçüm Vektörü
$\tilde{\omega}_k^i$	: Ağırlık Değeri
$P(z_t x_t)$	: Ölçüm Modeli (Olabilirlik Fonksiyonu)
$\Sigma_k$	: Kovaryans Matrisi

### Kısaltmalar

pdf	: Olasılık Yoğunluk Fonksiyonu (Probability Density Function)
CNN	: Evrimsel Sinir Ağları (Convolutional Neural Networks)
HSV	: Renk Doygunluk Değer (Hue Saturation Value)
RGB	: Kırmızı Yeşil Mavi (Red Green Blue)
SVM	: Destek Vektör Makinaları (Support Vector Machine)
SURF	: Speed up Robust Features
SIFT	: Scale Invariant Features Transform
SSD	: Single Shot Multibox Detector
KF	: Kalman Filter
EKF	: Genişletilmiş Kalman Filtresi (Extended Kalman Filter)
UKF	: Kokusuz Kalman Filtresi (Unscented Kalman Filter)
PF	: Parçacık Filtresi (Particle Filter)
SMC	: Sıralı Monte Carlo (Sequential Monte Carlo)
İHA	: İnsansız Hava Aracı
SHGM	: Sivil Havacılık Genel Müdürlüğü
SHT-İHA	: İnsansız Hava Araçları Talimatnamesi
AKS	: Akıllı Kontrol Sistemleri
CPU	: Merkez İşlemci Ünitesi (Central Processing Unit)
GPU	: Grafik İşlemci Ünitesi (Graphics Processing Unit)
FAST	: Features From Accelerated Segment Test
SHGM	: Sivil Havacılık Genel Müdürlüğü
AKS	: Akıllı Kontrol Sistemleri
IMU	: Atalet Sensörü (Inertial Measurement Unit)
4K	: 4000 piksel yatay çözünürlük
FPS	: Saniye Başına Kare (Frame per Second)

## 1. GİRİŞ

### 1.1. İnsansız Hava Araçları (Dronlar)

“Drone” kelimesi erkek arı anlamını taşıırken Türk Dil Kurumu bu tanımını “uçangöz” olarak belirlemiştir veya genel söylemiyle insansız hava araçları (İHA) olarak geçmektedir. Basitçe insansız hava aracı (SHT-İHA) anlamına gelmektedir. Yani, pilot aracın içinde bulunmamaktadır. Bu tanım baz alındığında, ilk İHA’lar, 1839’da Avusturya askerleri tarafından patlayıcılarla doldurulmuş insansız balonlar olarak kabul edilmektedir. Bu balonları Avusturyalılar Venedik şehrine saldırmak için kullanmışlardır. Bu İHA’ların bazıları başarılı olmuştur, ancak bir kısmı rüzgar ve hatalar sonucu geri dönerek Avusturyalıların kendi hatlarını bombalamasına sebep olmuştur, böylece uygulama ilk zamanlarında yaygın olarak kabul görmemiştir. Ancak kanatlı hava taşıtlarının icadı, insanlı ve insansız araçlar için birçok şeyi değiştirmiştir. İHA’lar 2. Dünya Savaşı süresince gelişim göstermiş hatta Amerika Birleşik Devletleri (ABD) ve Sovyet Sosyalist Cumhuriyetler Birliği (SSCB) arasındaki soğuk savaş sırasında casus dronlar kullanılmıştır. Fakat sonuç olarak İHA’ların profesyonel olarak kullanımı, çoğu alanda olduğu gibi yine savaş amaçlı olmuş ve ilk örneği İsrail tarafından 1982 yılında Suriye ordusuna karşı kullanılmıştır. Bu olaydan sonra İHA’lar kendilerini kanıtlamış ve aslında ne kadar önemli olduklarını göstermişlerdir.

Endüstriyel İHA’ların modern tarihi ise aslında çok uzak olmayan bir tarihe, 2006 yılına dayanmaktadır. Bu araçlar felaket yardımı, sınır gözetimi ve orman yangını mücadelesi için devlet kurumları tarafından, boru hatlarını denetlemek ve tarlalarda böcek ilacı püskürtmek için özel şirketler tarafından kullanılmaya başlanmıştır. Günümüzde ise İHA’ların önemi her sektörde kendini belli etmektedir. Haritacılıktan tarıma, enerji sistemlerinden askeri amaçlı kullanımlara kadar dronlar hayatımızın hemen hemen her yerinde kullanılmaktadırlar. Şekil 1.1’de modern İHA’ların bir örneği gösterilmektedir.



Şekil 1.1. Modern İHA'ılara bir örnek Robosys marka HK-3 model İHA

Gün geçtikçe İHA'lar hakkında birçok çalışma yapılmakta ve bunların kullanımını düzenleyici talimatname ve kanunlar çıkarılmaktadır. Bu çalışmaların ve düzenleyici unsurların en temel amaçlarından bir tanesi de insan hayatını korumak ve uçuş güvenliğini sağlamaktır. Ülkemizde Sivil Havacılık Genel Müdürlüğü (SHGM) 22.12.2016 yılında çıkarmış olduğu İnsansız Hava Araçları Talimatnamesi (SHT-İHA) ile İHA'ların sınıflandırmalarını yaparak insan güvenliği ve hava trafiğinin düzenini sağlamayı amaçlamaktadır. SHT-İHA talimatnamesinin 5. Maddesine göre İHA'lar ağırlıkları baz alınarak şu şekilde sınıflandırılmaktadır (SHT-İHA, 2016):

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

SHT-İHA talimatnamesinde hava aracının, uçuşun ve insanların güvenliğini sağlamak amacıyla kurallar ve kısıtlamaları getirilmiştir bunlara kısaca örnek verilecek olursa (SHT-İHA, 2016):

- MADDE 20 – (1) Uçuşlar esnasında, diğer hava araçları ile ayırmayı sağlamak ve çarpışmayı önlemek İHA pilotunun sorumluluğundadır.
- MADDE 25 – (1) İHA işleticisi ve pilotu, İHA'nın terör saldırılarında veya kamunun güvenliğini tehlikeye düşürecek herhangi bir yasadışı eylemde kullanılmasını önlemek amacıyla İHA sisteminin ve aracının dışarıdan müdahalelere karşı güvenliğini sağlamakla yükümlüdür.

Talimatnamedeki ilgili maddelerin tamamı EK-1’de verilmiştir. Bu gibi kuralların koyulması ve uygulanması insan hayatı için kritik bir önem arz etmektedir. Uçuş güvenliğinin sağlanması için koyulan kurallar ve çıkarılan talimatnamelerin yanında araştırmacılar da bu konu üzerinde yoğun bir biçimde çalışmaktadırlar. Sensörler ve akıllı kontrol sistemleri ise hava aracının dengeli ve güvenli bir uçuş yapabilmesi için hava aracına destek olmaktadır.

## **1.2. Akıllı Kontrol Sistemleri**

Parçaları arasında karşılıklı ilişki, etkileşim, bağlantı ve bağımlılık bulunan tutarlı bir bütün haline getirilmiş nesnelere toplamına sistem denir. Sistemdeki öğelerin arasındaki karşılıklı ilişkinin, etkileşimin ve bağlantının otomatik olarak yapay zeka tekniklerini veya bilgi tabanlı karar mekanizmaları ile otomatik kontrol edildiği sistemlere ise akıllı kontrol sistemleri (AKS) denmektedir. Akıllı kontrol sistemlerinde çeşitli yapay zeka, makine öğrenmesi, istatistiksel öğrenme metotları ve akıllı kontrol sistemleri kullanılmaktadır. Kullanılan metot ve algoritmalara ayrıntılı örnek olarak sinir ağları, bulanık mantık, kalitatif muhakeme, istatistiksel öğrenme, derin öğrenme, genetik algoritmalar gibi çeşitli metotlar gösterilebilir. Bu metotlar İHA’ların otonom olarak kontrol edilmesinde sıklıkla kullanılmaktadır.

### **1.2.1. Kullanıldığı alanlar**

Bir balondan uçan taksilere gelen hikayesiyle İHA’lar günümüzde arama kurtarmadan güvenliğe, savunma sanayinden akıllı şehirlere kadar sivil ve askeri birçok alanda yoğun bir biçimde kullanılmaktadır. Geleceğimizde de İHA’ların daha da çeşitli alanlarda hayatımıza gireceği yadsınamaz bir gerçektir. Akıllı kontrol sistemleri, sensörler, görüntü işleme ve yapay zeka gibi bir çok bileşenin de entegrasyonu ile birlikte dronlar kullanıldığı işlere katma değer kazandırmakta, maliyeti düşürmekte ve işin süresini oldukça belirgin bir şekilde kısaltmaktadır. Sivil hayatta insanlar dronların sadece çekim ve eğlence amaçlı kullanıldığını düşünmesine rağmen endüstriyel alanlarda özellikle de akıllı kontrol sistemlerinin de eklenmesi ve çeşitli kameraların (RGB, termal, multispektral, hiperspektral) faydalı yük olarak entegre edilmesi ile birlikte İHA’ların kullanım alanları şu şekilde oldukça geniş bir yelpazeye ulaşacaktır:

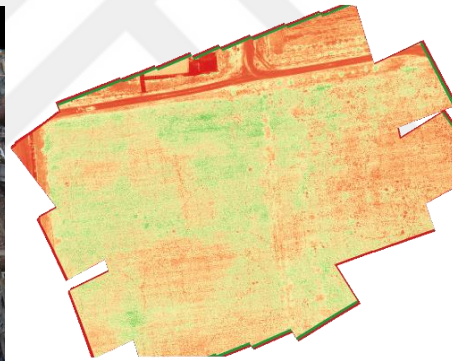
- Haritalama,
- Kargo,
- Toprak hacim analizi,
- Tarımda verimlilik ve hastalık analizi,
- İlaçlama ve sulama dronları,

- Enerji sistemlerinde ve boru hatlarında kaçak tespiti,
- Güneş enerjisi sistemlerinde arıza tespiti,
- Arama kurtarma amaçlı takviye veya gözlem dronları,
- İnşaatlarda süreç analizi,
- Yüze yakın yer altı kaynaklarının incelenmesi,
- Depo ve ambarlarda stok analizi,
- Askeri alanda saldırı, savunma, kitle yönetimi, psikolojik saldırı ve gözlem amaçlı v.b.

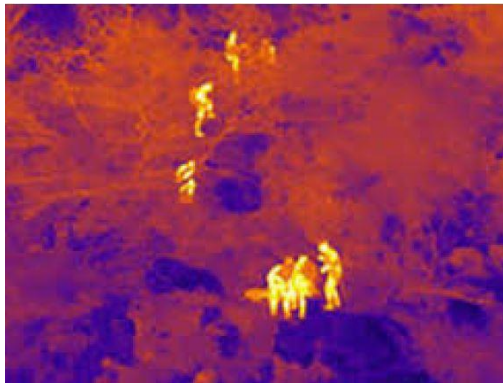
Görüldüğü üzere İHA'lar aslında oldukça geniş bir kullanım alanına sahiptir ve bu alanların çoğunda saf çekimden ziyade yapay zeka ve görüntü işleme barındıran akıllı kontrol sistemleri entegre edilmiş dronlar kullanılmaktadırlar. Şekil 1.2 (a)' da afet durumlarında İHA'lar ile bir modelleme yapılarak değerlendirme örneği, Şekil 1.2 (b)'de multispektral kameralı bir İHA'dan alınmış görüntü örneği, Şekil 1.2 (c)'de termal kameralı bir İHA'nın arama kurtarma amacıyla kullanımına bir örnek ve Şekil 1.2 (d)'de güneş enerjisi sistemlerinin yine termal kameralı bir İHA' ile çekilmiş görüntülerine örnekler verilmiştir.



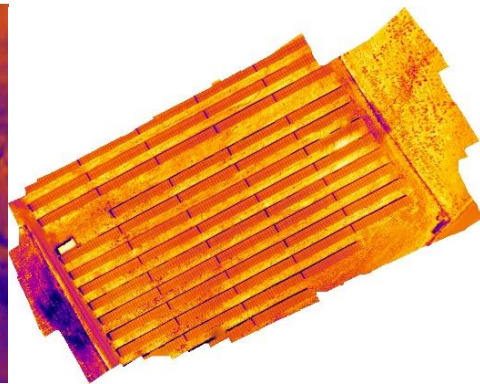
(a) Felaketlerde kullanım



(b) Tarımsal analiz



(c) Arama kurtarma



(d) Güneş enerjisi sistemlerinde analiz

Şekil 1.2. İHA'ların kullanım alanlarına örnekler

### 1.3. Tezin Amacı

Bu tezde dronlarda kullanılan akıllı kontrol sistemlerine yönelik nesne takip yöntemleri kullanılarak başarılı ve bir o kadar da hızlı bir nesne takip yönteminin bulunması amaçlanmıştır. Bu uygulamanın amacı İHA'larda bulunan akıllı kontrol sistemlerinde kullanılmaya en uygun gerçek zamanlı bir nesne takip algoritması bulunmasıdır. Bilindiği üzere İHA'lar oldukça hızlı hareket edebilmektedir. Engelden kaçınma veya nesne takibi özelliğine sahip bir İHA düşünüldüğünde AKS kararları oldukça hızlı ve doğru vermelidir aksi halde hava aracı ya engele çarpacak ya da yanlış karar vererek nesneyi kaybedecektir. Bunun için kullanılacak nesne takip algoritmasının iki olmazsa olmaz özelliğe sahip olması gerekmektedir. Bu özellikler hız ve doğruluktur. Algoritmanın hızlı olması kullanılacak algoritmanın işlem gücüne ve kullanılacak olan donanımın yeterliliğine bağlıdır. Algoritmanın doğruluğu ise kullanılan metodun sınıflandırma veya takip işlemini ne kadar doğru yaptığına bağlı olacaktır. Bu nedenle literatürde halihazırda kullanılan derin öğrenme ve istatistiksel öğrenme metotları kullanılarak performansları karşılaştırılmış ve İHA'larda akıllı kontrol sistemlerine en uygun nesne takibi metodu bulunmaya çalışılmıştır.

Belirtilmiş olan takip algoritmasının istatistiksel öğrenme metotları ile doğru bir şekilde gerçekleştirilmesi için aşağıdaki adımların gerçekleştirilmesi gerekmektedir.

- Girdi olarak nesneyi içeren bir şablon görüntünün algoritmaya verilmesi,
- Bu şablonun ayırt edici özelliklerinin bulunması,
- Nesnenin görüntü üzerinde tespiti,
- Nesnenin hareketinin tahmini.

Yukarıda belirtildiği üzere istatistiksel öğrenme metotları ile nesne takibinin gerçekleştirilmesi için öncelikle takip edilecek nesnenin bir şablon (template) görüntüsünün elde edilmesi gerekmektedir. Daha sonra bu kalıp görüntü ayırt edici özellikleri tespit edilerek nesne görüntü üzerinde tespit edilir. Ayırt edici özellikler nesnenin rengine, şekline veya yönelimine bağlı olabilir. Tespit edilen görüntünün istatistiksel öğrenme metotları ile hareket yönelimi tahmin edilir ve nesne takibi gerçekleştirilir. Bu çalışmada nesnenin ayırt edici özelliklerini çıkarmak için K-Means algoritması kullanılarak nesnenin baskın RGB değeri bulunmuş ve bu özellikleri çıkarılmıştır. Daha sonra istatistiksel olarak Bayes teorisine dayanan Kalman Filtresi (KF) ve Parçacık Filtresi (PF) kullanılarak İHA görüntülerinden nesnenin hareketinin tahmini edilerek nesne takibi gerçekleştirilmiştir.

Belirtilmiş olan takip algoritmasının derin öğrenme metotları ile doğru bir şekilde gerçekleştirilmesi için ise aşağıdaki adımların gerçekleştirilmesi gerekmektedir.

- Takip edilecek nesnenin veri setinin oluşturulması,
- Eğitim ve sınıflandırma işlemi yapacak olan çok boyutlu yapay sinir ağı mimarisinin belirlenmesi,
- Veri setinin eğitimi,
- Oluşturulmuş derin ağ kullanılarak sınıflandırma ve görüntü üzerinde nesnenin lokalizasyonunun yapılması.

Bir nesneyi derin öğrenme metotları kullanılarak takip etmek için bu cismin bulunduğu görüntülerden bir veri seti oluşturulmalıdır. Daha sonra bu görüntülerin eğitiminin yapılacağı çok boyutlu yapay sinir ağı mimarisi belirlenmelidir. Verileri bu mimaride eğitildikten sonra oluşturulmuş derin ağ ile görüntülerde sınıflandırma yapılarak nesnenin görüntü üzerindeki yeri tespit edilir. Bu çalışmada veri seti olarak Microsoft COCO (Common Object in Context) (2014) veri seti kullanılmıştır. Eğitim ve sınıflandırma için ilk olarak Singleshot Multibox Detector (SSD) (2016) yöntemi ile bir evrişimsel ağ mimarisi olan MobileNet (2017) ağ mimarisi kullanılmış. İkinci olarak Faster R-CNN (Ren ve ark., 2015) yöntemi ile Başlangıç (Inception) Res-Net v2 (TensorFlow, 2016) mimarisi kullanılmıştır.

Bu tez kapsamında bir İHA'dan alınmış farklı yükseklikteki (5, 10, 20 metre) görüntülerden cismin oryantasyonunun değişmesine rağmen nesnenin takibinin sağlanması için Bayes teoremi temelli istatistiksel metotlar kullanılmıştır. Aynı görüntülerde derin öğrenme metotları kullanılarak nesne takibi uygulaması yapılmış ve birlikte bir performans karşılaştırması yapılmıştır. Bu kapsamda evrişimsel sinir ağları (CNN), Fast R-CNN, Faster R-CNN, SSD, PF ve KF teorik olarak incelenerek İHA'dan alınan görüntüler üzerinde denenmiştir. Saniye başına kare (FPS), doğruluk oranı, CPU (Central Processing Unit) ve GPU (Graphics Processing Unit) kullanım yüzdeleri karşılaştırılarak gerçek zamanlı uygulamaya en uygun ve bu sayede bir İHA'da kullanılacak en efektif nesne takip uygulamasının bulunması amaçlanmıştır. Bu sayede İHA'lar görevlerini sorunsuzca yerine getirebilecek, can ve mal güvenliği sağlayacak ve uçuş güvenliği arttırılacaktır.

#### 1.4. Tezin Önemi

İHA'lar üzerine yapılan çalışmalar gün geçtikçe ilgi görmektedir. Özellikle İHA'ların otonom olarak kullanılması için son zamanlarda popüler hale gelen yapay zeka ve görüntü

işleme teknikleri içeren uygulamalar pek çok alanda göze çarpmaktadır. Bunun için devlet kurumları ve özel şirketler bu alana yönelik inovatif ve efektif çalışmalar içerisindeyler.

Bu çalışmada İHA'lara yönelik akıllı kontrol sistemlerini geliştirmek için görüntü işleme, istatistiksel öğrenme ve yapay zekaya dayalı nesne takip metotları üzerine uygulamalar gerçekleştirilmiştir. Bu uygulamalarda karşılaşılan zorluklar nesnenin boyut ve oryantasyonunun hareket süresince değişmesi, nesne takibinin gerçek zamanlı bir hava aracının hareketini yönetebilecek kadar hızlı olması gerekmesinden kaynaklanmaktadır. Yapılan uygulamalarda alınan görüntülerin İHA üzerinde veya yer kontrol istasyonunda bulunan bir işlemci (CPU, GPU veya her ikisi de) tarafından değerlendirilmesi esastır. Kullanılan metotlardaki temel sorun ise hızlı olan metotların düşük başarı oranı vermesi, yüksek başarıya sahip olan metotların ise genellikle gerçek zamanlı uygulamalara nazaran yavaş kalmasıdır. İHA'larda kullanılan bu metotlarda mevcut olan sorunlar ise şu şekilde:

- Alınan görüntülerin yüksek titreşime veya gürültüye maruz kalması,
- Hava aracının hareketinden dolayı cismin yönünün değişmesi,
- Takip edilen nesnenin başka bir cismin arkasına geçerek görüşünün yarım veya tam olarak kapatılması,
- Takip sırasında nesne üzerine düşen ışık miktarının ya da açısının değişimi,
- Nesnenin ani ve karmaşık hareketler yapması,
- Hava aracının yüksekliğine bağlı olarak görüntüde nesnenin boyutunun değişmesi olarak sıralanabilir.

Yukarıda belirtilen zorluklar İHA'larda gerçek zamanlı takip uygulamalarının gerçekleştirilmesini zorlaştıran faktörlerdir. Bu tez sırasında bu faktörleri ortadan kaldırmak ya da minimuma indirmek için optimum nesne takip uygulamasının bulunması ve geliştirilmesi amaçlanmıştır.

Gün geçtikçe hayatımızın her yerine giren İHA'ların kullanımının gelecekte de artacağı yadsınamaz bir gerçektir. Gelecekte İHA'ların kendi kendine tam otonom olarak karar vererek bir operatöre ihtiyaç duymadan çalışması beklenmektedir. Yapay zeka teknikleri ve bu teknikleri işleyecek donanımların da her geçen gün gelişmesiyle beraber, bunun çok yakında olması muhtemeldir. Yani yakın gelecekte İHA'lar artık kendi başlarına uçuşa başlayacak, görevini gerçekleştirecek, uçuş sırasında dış etmenlerin neden olduğu zorluklar ile, bulundurduğu sensörler ve akıllı karar mekanizmaları sayesinde kendi başına karar vererek mücadele edecek, görevini tamamladıktan sonra inişini yapacak ve gerekirse kendi şarjını



kendisi edecektir. Yeri gelecek bu hava araçları bir insan-makine ara yüzü sayesinde insanlar ile beraber koordine bir şekilde çalışacaktır. Örneğin halihazırda enerji sistemlerinde kaçak analizi yapan İHA'lar sayesinde arızalar kolayca tespit edilmekte, arızanın yer ve durum bilgisi operatöre gönderilerek sorumlu kişinin bu arızayı tamir etmesi sağlanmaktadır. Bu duruma başka bir örnek ise askeri alanda zorlu arazilerde mühimmat taşıyan, gözlem yapan veya saldırı amaçlı kullanılan İHA'lar gösterilebilir. Bu kapsamda nesne takibi İHA'lar için büyük bir önem taşımaktadır. Bu tezde de İHA'dan alınmış görüntüler kullanılarak mevcut nesne takip uygulamaları arasında bir kıyaslama yapılmış, ve İHA'lara yönelik akıllı sistemlerde kullanılabilecek en uygun nesne takip algoritmasının bulunması hedeflenmiştir.

### 1.5. Tezin Kapsamı

Bu tez kapsamında gerçekleştirilen uygulamalar İHA'lar için akıllı kontrol sistemlerinin geliştirilmesine yöneliktir. Bu konuda nesne takibi önemli bir rol oynamaktadır. Bu kapsamda mevcut nesne takibi uygulamaları baz alınarak performansları kıyaslanmış. Mevcut uygulamalar kullanılarak geliştirilen hibrit algoritma ile geliştirmeler sunulmuş ve alternatifler verilmiş. Gerçek zamanlı olarak bir İHA'da kullanılabilecek en uygun nesne takip algoritması bulunması amaçlanmıştır.

Bu çalışmada ilk olarak farklı yüksekliklerde görüntüsü alınmış bir nesnenin dört farklı mevcut nesne takip yöntemi ile tespit ve takibi yapılmış, FPS, doğruluk oranı, CPU ve GPU kullanım yüzdeleri karşılaştırılarak bir performans analizi yapılmıştır. Son olarak ise mevcut metotların avantajlı yönlerinde faydalanarak bu metotların bir arada kullanılabileceği bir hibrit algoritma önerilmiştir. Bu İHA'lar için en optimum nesne takip algoritmasının bulunarak gerçek zamanlı İHA uygulamalarında kullanılması için çalışılmıştır. İHA'larda akıllı kontrol sistemlerine yönelik bu çalışmalar her aşamada geliştirilmeye açıktır.

Özet olarak İHA sistemleri gün geçtikçe önem kazanmaktadır. İHA'ların kullanım alanları arttıkça insan ve uçuş güvenliği konusunda regülasyonlar artmaktadır. Buna ek olarak araştırmacılar da en güvenli uçuşu sağlamak için çeşitli akıllı kontrol sistemleri geliştirmektedirler. Bu akıllı kontrol sistemlerinin en önemli parçalarından biri de nesne takip algoritmalarıdır. İHA'larda kullanılacak olan nesne takip algoritmalarının yüksek doğruluk oranı ve hızlı tepki süresine ihtiyaçları vardır. Aksi halde İHA başarılı bir nesne takibi gerçekleştiremeyecek veya ani bir manevra gerektirdiğinde kazalara sebebiyet verecektir. Bu tezde dronlarda kullanılan akıllı kontrol sistemlerine yönelik nesne takip yöntemleri

kullanılarak başarılı ve bir o kadar da hızlı bir nesne takip yönteminin bulunması amaçlanmıştır. Bu amaca yönelik literatür taraması hakkında bilgi 2. Bölümde verilmiştir.



## 2. LİTERATÜR TARAMASI

Bu tez çalışmasının amacı olan İHA'larda bulunan akıllı kontrol sistemlerine en uygun nesne takip algoritmasının bulunmasına yönelik birçok farklı kaynak araştırması yapılmıştır. Kullanılması planlanmış yöntemler ve ilgili konular araştırılmıştır. Yöntemlerin kullanıldığı uygulamalar ve hangi yardımcı yöntemlerle beraber kullanıldığı ile alakalı bilgi edinilmiştir.

### 2.1. Akıllı Kontrol Sistemleri ile Yapılmış İnsansız Hava Aracı Uygulamaları

İHA'ların otonom uçuşları konusunda akıllı kontrol sistemlerinin kullanıldığı uygulamalara göz atacak olursak Zhou ve ark. yaptıkları çalışmada bir hava aracı ile ormanlık alanda uçuş yapılmış ve bu uçuş sırasında video çekilmiştir. Bu videodan bir harita birleştirme (map merging) türü olan ortorektifikasyon işlemi yapılmıştır. Genellikle bu işlem fotoğrafların birleştirilmesinde kullanılmasına karşın bu makalede video üzerinde işlem yapılarak bir yenilik getirilmiştir.

Wenzel K. E. ve ark. (2012) yapmış olduğu çalışmada kızılötesi alıcılar kullanılmıştır. Bu makalede bir İHA'nın diğer bir dronu Nintendo Wii kızıl ötesi alıcı ve vericiler ile takip etmesi anlatılmıştır. Çalışmada açık kaynak kodlu İHA'lar olan Hummingbird ve Parrot kullanılmıştır.

Bi ve ark. (2013) bunu bir adım öteye taşıyıp görüntü işleme teknikleri kullanarak bir İHA'ya otonom olarak nesne takibi ve iniş yaptırmıştır. Bu makalede görüntüdeki RGB değerleri kullanılarak basit eşikleme (threshold) işlemleri ile iki farklı uygulama gerçekleştirilmiştir. İlkinde bir helikopter pistine iniş yapılmış, ikinci uygulamada ise kırmızı bir araba düz bir zemin üzerinde takip edilmiştir. PID kontrolcüsü ile dron kontrol edilmiş, stabil uçuş için PID değerleri bulunmuştur. Uçuş ve iniş esnasında GPS kullanılmamış sadece Wii kameradan alınan görüntülerden faydalanılmıştır. Aynı zamanda bir kullanıcı arayüzü hazırlanmış ve iniş, kalkış ve takip gibi hareketler kontrol edilmiştir.

Möckel (2012) doktora tezinde optik akış (optic flow) sensörlerinin tanıtımını yapmış ve İHA uçuşunda bunların kullanılabilceği alanlara değinmiştir. Optik akış sensörleri konusunda ayrıntılı bir şekilde açıklamaya yer verilmiş ve bununla robotların otonom olarak kontrolüne değinilmiştir. İlk olarak optik akış sensörünün ne olduğu içeriği ve yapısı anlatılmış, bu sırada optik akış sensörünün kamerasında bulunan CMOS transistörlerden bahsedilmiştir. Optik akış tanımını ne demektir grafiklerle ayrıntılı bir şekilde anlatılmıştır. Daha sonra bu sensörün arıların gözlerine benzerliğinden bahsedilmiş ve birkaç tane sensörü aynı böcek

gözüne benzer şekilde yerleştirerek bir otonom uçuş uygulaması denenmiştir. Ayrıca uygulama olarak öncelikle bir araba simülasyonu dinamik optik akış sensör ile kontrol edilmiştir.

Pestana ve ark. (2013) yaptıkları çalışmada bir dört motorlu döner kanat dronun GPS'e bağımlı olmaksızın görüntü temelli görmeye dayalı robot kontrolü (visual servoing) metodu ile kontrol edilmiştir. Kullanıcının belirlediği bir obje Parrot Dron tarafından takip edilmiştir. Objeye takibi için açık kaynak kodlu obje takip framework olan OpenTLD kullanılmıştır. Bu algoritma eski olup, şimdilerde Clustering of Static-Adaptive Correspondences for Deformable Object Tracking (CMT) algoritması ile yenilenmiştir. Bu çalışmada uygulama olarak insan, pencere, bitki, araba ve AC makineler takip edilmiştir.

Dang ve ark. (2013) yaptıkları çalışmada düşük bütçeli AR dron ile yerdeki objenin takibi yapılmıştır. İHA kapalı çevrim (closed-loop) PD kontrolcüsü ile kontrol görüntü temelli olarak kontrol edilmiştir. Sabit ve düz bir arka planda RGB'den HSV'ye ve daha sonra threshold uygulanarak obje görüntüde bulunmuştur. İHA bir yer istasyonundan (ground station) Wifi ile kontrol edilmiştir. İHA'da ultrasonik sensörler de kullanılmıştır. Objeye ortada tutulacak şekilde piksel hesabı ile İHA objeye X-Y düzleminde hizalanmıştır.

## **2.2. Nesne Tespiti ve İHA'larda Nesne Takibi Üzerine Yapılmış Uygulamalar**

### **2.2.1. Kalman Filtresi ve Parçacık Filtresi ile İstatistiksel Öğrenme**

İstatistiksel öğrenme metotları kullanılarak İHA'lar üzerinde birçok çalışma yapılmış ve halen yapılmaya devam edilmektedir. İstatistiksel öğrenme metotları İHA'ların kontrolünden nesne tanımasına ve takibine kadar birçok farklı şekilde kullanılmaktadır. Örneğin Tao ve ark. (2002) yaptıkları çalışmada Bayesci tahmin ile nesne takibine değinilmiştir. Teorik bilgi yoğunluklu bu çalışmada araba ve insan takibine ilişkin bir uygulama yapılmıştır. Cisimleri görüntüde bulmak için şekil öncelikli gauss temelli bir algoritma kullanılmıştır ve hava aracından alınan iki boyutlu görüntü ile gerçek zamanlı bir uygulama gerçekleştirilmiştir. Dinamik katman temsiline Bayesci tahmin ile görüntüler takip edilmiştir. Dinamik katman temsili ile çalışılmasının sebebi olarak hem hareket tahmini hem de birden fazla obje ile çalışabilme olanağı sağlamaktadır.

İmamoğlu ve ark. (2010) çalışmalarında dalgacık dönüşümünden yararlanmış iki ayrı uygulama yaparak bunları birbiri ile ilişkilendirmişlerdir. İki uygulamada da Gabor dalgacıkları kullanılmıştır. Gabor dalgacıklarının en önemli özelliklerinden biri farklı ölçek ve oryantasyonlarda analiz imkânı sağlamasıdır. Yapılan çalışmada görüntüdeki nesne arka

plandan Gabor dalgacıkları ile ayrılarak, elde edilen görüntünün istenen nesne sınıfına ait olup olmadığı belirlenmiştir. En yakın komşu (nearest neighbours), en yakın k-komşu (k-nearest neighbours), Bayes teoremi ve yapay sinir ağları ile sınıflandırma yapılarak örüntülerin istatistiksel özellikleri üzerinden başarımları analizi yapılmıştır.

Bouffard ve ark. (2012) yaptıkları çalışmada öğrenmeye dayalı öngörü modeli kontrolü (learning based predictive model control, LBPMC) metodu kullanarak bir İHA'nın uçuş kalitesini ve güvenliğini artırma üzerine çalışmıştır. Yapılan çalışma istatistiksel öğrenme ile kontrol mühendisliğinin güzel bir birleşimini teşkil etmektedir. Bu çalışmaya benzer bir çalışmada ise Roberts ve ark. (2012) İHA'lar üzerine yaptıkları uygulamada Bayes ağları kullanarak ormanlık arazide otonom uçuş sırasında ağaçların haritalanması üzerine bir çalışma yürütmüştür.

İstatistiksel temelli Bayesci tahmine dayanan uygulamalarda Kalman filtresine örnek verilecek olursa, KF ile ilgili bir çok çalışmanın yapıldığı görülecektir. GPS sistemlerinden radarlara kadar bir çok alanda kullanılan KF ile ilgili literatürde yapılmış çalışmalara örnek verilecek olursa aşağıdaki çalışmalar göze çaracaktır.

Weng ve ark. (2006) yaptıkları çalışmada video üzerinden Kalman filtresi kullanarak hareketli nesne takibi metodu sunmuşlardır. Kullanıcı tarafından seçilen hareketli nesne görüntüden ayrıştırılmış ve RGB renk özelliklerinden baskın olan renk çıkarılmıştır. Nesne takibi sırasında ise adaptif Kalman filtresi kullanılarak nesne hareketi yöneliminin tahmini yapılmıştır. Sunulan bu metod, gerçek zamanlı kompleks durumlarda dahi hızlı bir şekilde nesnenin takibini yapabilmekte, cismin ani hareketi, yönünün değişmesi, oryantasyonun değişmesi ve ortam ışığının değişmesi durumunda dahi başarılı olmuştur. Fakat hızlı olmasına karşın bu metod, başarı oranı nezdinde güncel metodlar ile kıyaslandığında yetersiz kalmaktadır. Benzer şekilde Cuevas ve ark. (2005) Kalman filtresinin doğrusal çözümüne ek olarak doğrusal olmayan sistemlerde de kullanılabilir olan genişletilmiş Kalman filtresini (EKF) görsel nesne takibi için sunmuşlardır. EKF, doğrusal olmayan modelleri adım adım doğrusallaştırarak bu modelleri standart Kalman filtresine sokularak hareket takibini gerçekleştirmişlerdir. EKF'de daha önce bahsedildiği üzere doğrusal olmayan modeller çözümlenebilmektedir. Fakat modelin tek gerekliliği türevinin alınabilir olmasıdır. Li ve ark. (2010) yapmış oldukları çalışmada, Weng ve arkadaşlarının 2006 yılında ortaya koydukları video üzerinden Kalman filtresi kullanılarak hareketli nesne takibi metodunu bir üst seviyeye çıkarmış; takip edilecek nesne ya da nesnelerin önüne geçen başka bir cisim sebebiyle birbirini engellemesi (oklüziona uğraması) durumunda dahi hareket eden tek ya da çoklu nesnelerin takibini

gerçekleştirebilmişlerdir. Uygulama sırasında kullanıcı tarafından seçilmiş nesne ya da nesnelerin baskın RGB renk özelliklerinin de yanında cisimlerin geometrik özellikleri de çıkarılmış, daha sonra nesnelere 'Arka Planı Ayrıştırma' işlemi ile Kalman filtresine sokulmuştur. Kalman filtresinin sağlamış olduğu istatistiksel temelli Bayesci yaklaşımı ile hareket tahmininin yapılması sayesinde karşılıklı birbirlerine doğru gelen iki nesne oklüzyona uğramasına rağmen hata olmaksızın takip edilebilmektedir.

KF'ne benzer bir şekilde istatistiksel temelli Bayesci tahmine dayanan en sık kullanılan metotlardan bir diğeri ise Parçacık filtresidir. PF ile ilgili literatürde yapılmış çalışmalara aşağıdaki çalışmalar örnek gösterilebilir.

Şekli, oryantasyonu ve boyutu değişken nesnelerin hızlı gerçek zamanlı nesne takibinin yapılması konusunda kendini kanıtlamış, doğrusal olmayan problemlerin çözümüne odaklı başarılı diğer bir metot ise Parçacık filtresi yöntemidir. Nummiaro ve ark.(2003) yapmış oldukları çalışmalarında renk dağılımlarının entegrasyonunu kenar tabanlı görüntü özelliklerinin kombinasyonu ile birlikte Parçacık filtresine sokmuş; bu sayede cisim yarı oklüzyona uğradığında, yön değişikliğinde veya boyut değişikliğinde nesne takibinin başarılı bir şekilde yapıldığı görülmüştür. Çalışmanın devamında ortalama kayma vektörü (Mean Shift) takip algoritması ve Kalman filtresi ile başarı oranı karşılaştırılmış ve Parçacık filtresinin daha başarılı olduğu sonucuna ulaşılmıştır. Hue ve ark. (2002) yapmış oldukları çalışmalarında Parçacık filtresi kullanarak birden fazla nesnenin takibini sağlamışlardır. Çalışmada klasik Parçacık filtresine ek olarak rassal konum vektörlerinin tahminini Gibbs örnekleyicisi ile yapmıştır. Sun ve ark. (2012) yapmış oldukları çalışmalarında, Parçacık filtresi ile Mean Shift birleştirilerek kullanılmıştır. Sunmuş oldukları yeni yöntemde nesnenin konumu, oryantasyonu ve boyutu eliptik olarak tanımlanarak ve yapılmış olan bu tanımlama ilk kareyi temsil etmektedir. İlk olarak Mean Shift vasıtası ile bir dengeleme vektörü elde edilerek bu vektör bilgisi, durum geçiş modeline dönüştürülmektedir. Buna göre parçacıkların durumu Gauss dağılımı ile belirlenmekte ve her karede güncellenmektedir.

Ek olarak istatistiksel öğrenmenin yanında makine öğrenmesi ve derin öğrenme uygulamaları da İHA'ların otonom uçuşları için sıkça kullanılmaktadır.

### **2.2.2. Görüntü işleme ve Makine Öğrenmesi**

İHA'lara yönelik yapılan çalışmaların başında görüntü işleme ve makine öğrenmesi uygulamaları gelmektedir. İHA'larda görüntü işleme uygulamalarının ilk zamanlarında sıklıkla İHA'dan alınan görüntülerle yangın tespiti yapılması amaçlanmıştır. Merino ve ark. (2005)

çalışmalarında kızıl ötesi görüntüler kullanarak yangın tespiti yapmıştır. RGB renk uzayında işlem yapılmış, resimde alevi temsil eden piksellerin kırmızı değeri daha fazla olduğu için bu şekilde alevin koordinatlarına ulaşılmıştır. Yine Merino ve ark. (2006) yaptıkları çalışmayı geliştirmiş ve bir önceki makalede anlatılan metodun aynısını kullanmış fakat bundan farklı olarak öncelikle bir veri tabanı oluşturulmuş ve alev tespit edilen görüntüler veri tabanı ile karşılaştırılarak kesin bir tespit yapılmıştır.

Yuan ve ark. (2015) yine havadan çekilen İHA görüntülerinden yangın tespiti yapmıştır. İlk başta ön işleme olarak medyan filtresi ve renk dönüşümü yapılmış daha sonra Otsu Threshold metodu uygulanarak görüntü morfolojik operatörlerle düzenlenmiş ve damlacık sayımı (blob counter) ile ateş tespiti yapılmıştır.

Görüntü işleme uygulamalarına diğer bir örnek ise görüntüden özellik çıkarımı yapılarak nesne tespiti yapılmasıdır. Moranduzzo ve ark. (2012) insansız hava aracı görüntülerinden Scale Invariant Feature Transform (SIFT) özelliklerini çıkardıkları görüntüleri bir makine öğrenmesi metodu olan Destek Vektör Makinası (Support Vector Machine, SVM) ile sınıflandırılarak araba tespiti yapmıştır.

Molloy ve ark. (2017) yaptıkları bu çalışmada, bir hava aracı ile başka bir hava aracının tespit edilmesi uygulaması gerçekleştirilmiştir. Bu çalışmadaki amaç, iki tane karşı karşıya gelmekte olan hava araçlarının çarpışmasını engellemektir. Makalede üç metot karşılaştırılmıştır. 1) Görüntünün homografisi kullanılarak background subtraction yapılmış ve uçak tespit edilmiş, 2) SIFT özellikleri çıkarılarak tespit işlemi gerçekleştirilmiş, 3) Speeded up Robust Features (SURF) özellikleri çıkarılarak tespit edilmiştir ve en iyi metot olarak birinci metot bulunmuştur.

Chen ve ark. (2017) yaptıkları çalışmada bir önceki çalışmayı geliştirmeyi amaçlamış ve İHA'dan alınan görüntülerden Features From Accelerated Segment Test (FAST) özellikleri çıkartılarak local difference binary algoritması kullanılarak görüntüler binary olarak temsil edilmiş ve en yakın k-komşu (K-nearest neighbor) algoritması kullanılarak bir nesne tanıma gerçekleştirilmiştir. Daha sonra atalet sensöründen (IMU) alınan veriler ile nesnenin İHA ile takip edilmesi sağlanmıştır.

### 2.2.3. Derin Öğrenme

Krizhevsky ve ark. (2012) görüntü sınıflandırma için evrimsel sinir ağlarını (CNN) kullanarak ImageNet yarışmasında birinciliği elde ettiğinden bu yana saf görüntü işleme yavaş

yavaş arka plana atılmış ve CNN'ler, araştırmacıların odak noktası olmuştur. Girshick ve ark. (2014) yapmış oldukları çalışmalarında CNN'ler ile nesne tespiti üzerindeki öncü çalışmalardan birini gerçekleştirmişlerdir. Metot, aslında basit görünen iki aşamadan oluşmaktadır. Bunlardan ilki girdi olarak bir görüntü verilmesi ve ikinci aşamada çıktı olarak her nesnenin çerçevesiyle bir şekilde etiketlerinin yazılmasıdır. Bu çalışmada standart CNN'e nesnenin görüntü üzerinde daha hızlı bir şekilde aranması için seçici tarama yöntemi (Selective Search) kullanılmıştır. Daha ayrıntılı açıklamak gerekirse seçici tarama; görüntüyü farklı boyutlardaki pencerelerle tarar ve her bir boyutta pencere için bitişik pikselleri doku, renk ve yoğunluklarına göre gruplandırarak nesnelerin aslında bir nesne olup olmadıklarını bulmaya çalışır. Bu seçici tarama metodunun CNN ile birleştirilmiş haline Regions With CNN Features (R-CNN) denmektedir. Basit bir biçimde özetlemek gerekirse, R-CNN şu şekilde uygulanmaktadır (Girshick, 2015):

- 1) Nesnelerin olabileceği yerlerin sunulması (region proposal),
- 2) Sunulmuş olan bu bölgelerin daha önceden eğitilmiş bir yapay sinir ağından geçirilmesi ve son olarak eski bilgilere dayanılarak SVM metodu ile sunulan bölgedeki nesnenin hangi nesne olduğunun bulunması,
- 3) Bölgelerin lineer regresyon modelinden geçirilerek sınıflandırılmış olan nesnenin koordinatlarının daha kesin bir biçimde bulunması.

Günümüzün revaçta konularından olan derin öğrenme de İHA uygulamalarında sıkça kullanılmaktadır. Bunun için Gemert ve ark. (2014) bu konuda ilk çalışmalardan birini gerçekleştirmiş ve evrimsel sinir ağıları kullanarak hayvanların doğada tespit edilmesi ve sayılması hakkında bir çalışma ortaya koymuştur.

Hung ve ark. (2014) çalışmalarında tarladan alınan İHA görüntülerinde bitki sınıflandırılması yapılmıştır. Öncelikle alınan görüntüler mozaikleme ile birleştirilerek orto foto elde edilmiş, daha sonra derin öğrenmenin bir yöntemi olan özellik öğrenimi (feature learning) kullanılarak bitkiler sınıflandırılmıştır.

2015 yılına kadar R-CNN'ler oldukça iyi bir biçimde sonuç veriyorlardı. Ancak oldukça yavaş çalışıyorlardı. Bunun için Girshick ve ark. (2015) var olan metodu bir adım öteye taşıyarak biraz daha hızlı çalışır hale getirmişlerdir. R-CNN metodunda sunulmuş olan bölgelerin her birinin tek tek CNN'den geçirilmesi işlem yükünü oldukça artırmaktaydı (yaklaşık olarak 2000 defa). Araştırmacılar bu durumdan ilham alarak sunulan bölgelerin bir arada toplanıp tek seferde CNN'den geçirilmesi ile eski uygulama yönteminin sonucunda



oluşan işlem yükünü kayda değer ölçüde azaltmışlardır. Bu işleme RoIPool denilmektedir. Önceki uygulamada; (R-CNN) görüntü özelliklerinin çıkarılması (CNN), sınıflandırma (SVM) ve koordinatların kesinleştirilmesi (Regressor) işlemleri için farklı modeller kullanılmasına karşın bu metotta bütün bu işlemler için tek bir yapay sinir ağı kullanılmıştır. Buna da araştırmacılar, R-CNN'in bir gelişmiş versiyonu olduğundan dolayı Fast R-CNN adını vermişleridir. Ren ve ark. (2015) bütün geliştirmeler ve hızlandırmalara rağmen Fast R-CNN'de bir dar boğaz olduğunu fark etmişlerdi. Bu sorunun bölgelerin önerimi olduğunu fark eden araştırmacılar seçici tarama işleminden kurtulmak için yeni bir ağ geliştirmiş ve bunun adını ise Fast R-CNN'in bir üstü olan Faster R-CNN koymuşlardır. Bu çalışmada bütün özellikler Fast R-CNN ile aynı iken sadece bölgelerin önerimi yerine CNN'in evrimsel katmanından çıkarılmış özellikler kullanılarak nesnelere yerinin tahmini CNN tarafından gerçekleştirilmiş ve böylece seçici taramada meydana gelen dar boğaz engellenmiş oldu. He ve ark. (2017) ise nesnelere bulunduğu bölgeler yerine her bir nesnenin piksellerinin tek tek incelenmesi konusunda çalışmalarını yürütmüşlerdir. Mask R-CNN adını verdikleri bu metotta Faster R-CNN'de bulunan nesne bölgelerinin bulunduğu kutucuklar gibi çıktı yerine 1 ve 0'lardan oluşan bir matris verilmektedir (bu matrise binary mask da denmektedir). Bu matris nesnelere bulunduğu yerleri belirtmektedir.

Sonuç olarak standart makine öğrenmesi ve lineer regresyon destekli yapay sinir ağlarından piksel seviyesinde analize kadar gelişen derin öğrenme yöntemleri geliştikçe, İHA'lara da uyumlu hale gelmekte ve kullanımı artmaktadır. Derin öğrenmenin İHA'larda kullanımına modern bir örnek olarak Kaufmann ve ark. (2018) yaptıkları çalışma gösterilebilir. Bu çalışmada CNN'ler ile bir rota planlama sistemi ve kontrol sistemi birleştirilmiştir. CNN, ham görüntüleri doğrudan bir yol rotası ve istenen hız şeklinde temsil edilir. Bu bilgi daha sonra planlayıcı tarafından en kısa, minimum sapmaya sahip yörünge parçaları ve istenen hedefe ulaşmak için kullanılması gereken motor komutlarını üretmek için kullanılır.

Yukarıda bahsedilen uygulamalar incelendiğinde İHA'lar için geliştirilen akıllı kontrol sistemlerinin başarı oranı gün geçtikçe artmaktadır. Nesne takibi göz önüne alındığında ise üretilen yöntemlerin gerçek zamanlı olarak İHA'larda kullanılması için hala hızının ve performansının artırılması gerekmektedir. Şayet yeteri kadar gelişmemiş ve yavaş bir nesne takip algoritmasının kullanılması çok hızlı tepki vermesi gereken bir İHA'nın kaza yapmasına sebep olacaktır.

Görüleceği üzere İHA'lar ve nesne takibi üzerine yapılmış birçok çalışma literatürde bulunmaktadır. Yaygın bir şekilde üzerinde çalışılmasına rağmen literatürde yer alan

alıřmalarda İHA’larda kullanılabilir en uygun nesne takibi metodunun ne olduđuna bir aıklık getirilmemiřtir. Yapılan literatür taraması ışığında, bu tezde İHA’larda akıllı kontrol sistemlerine yönelik en uygun nesne takibi metodunu bulmak için yapılacak olan alıřmalara yönelik materyal ve yöntem bir sonraki bölümde yer verilmiřtir.



### 3. MATERYAL VE YÖNTEM

Bu bölümde yapılan literatür taraması ışığında İHA'larda kullanılan akıllı kontrol sistemlerine yönelik en uygun nesne tanıma ve nesne takibi metodunun bulunması için gerekli materyal ve yönteme yer verilmiştir. Yapılan uygulamalarda istatistiksel metotlar için Python betik dili kullanılmış ve özgün bir yazılım yazılmıştır. Derin öğrenme uygulamaları için yine Python betik dili ile Tensorflow framework kullanılmıştır.

#### 3.1. Nesne takibi

Nesne takibi, basitçe anlatmak gerekirse, hareketli bir veya birçok nesnenin bir video üzerinde rotasının takip edilmesidir. Nesne takibi uzun bir süredir popüler bir araştırma konusu olmuştur. Hatta bu konunun görüntü işleme uygulamalarının en önemli konularından biri olduğunu söylemek yanlış olmayacaktır.

İHA'ların kullanım alanlarına göre uygulamalarda kullanılacak metotlar farklılık göstermektedir. Lakin pek çok uygulamada nesne tanıma ve nesne takibinin üzerine çalışıldığı göze çarpmaktadır. Güvenlik, arama-kurtarma, savunma sanayi odaklı birçok çalışmanın nesne tanıma gerektirmesi de bu konu üzerine yapılan çalışmaların artmasına sebep olmaktadır.

Nesne tanıma, robotların, bilgisayarların, diğer yapay zeka ve bilgisayarlı görüş uygulamalarının çeşitli nesnelere ve varlıklarını tanıma yetenekleri ile ilgili görüntü işleme ve yapay zeka alanını kapsamaktadır. Nesne tanıma, robotların ve bilgisayarların, video ve kamera görüntüleri gibi girdi nesnelere seçip almalarını sağlar. Nesne tanımlama için kullanılan yöntemler arasında 3B (3 boyutlu) modeller, bileşen tanımlama, kenar algılama ve farklı açılardan görünümünün analizi bulunur.

Görsel bir nesne takip sisteminin iki ana bileşeni vardır:

- 1) Hedef temsili ve lokalizasyon,
- 2) Filtreleme ve veri ilişkilendirmesi.

**Hedef Temsili ve Lokalizasyon:** Hedef temsili ve lokalizasyon aslında bir tüme varım yöntemidir. Bu yöntemler, hareketli nesneyi tanımlamak için çeşitli araçlar sunar. Hedef nesneyi başarılı bir şekilde bulmak ve izlemek algoritmanın türüne ve işleyişine bağlıdır. Örneğin, damlacık (blob) takibini kullanmak insan hareketini tanımlamak için kullanışlıdır, çünkü bir kişinin profili dinamik olarak sürekli değişir (Kang ve ark., 2003). Tipik olarak, bu algoritmalar için hesaplama karmaşıklığı düşüktür. Bazı sık kullanılan hedef temsili ve lokalizasyon algoritmalarına örnek olarak:

- Çekirdek (Kernel) temelli takip,
- Kontur takibi verilebilir.

Çekirdek temelli algoritmalar arasında en sık kullanılan metotlardan biri ise Mean Shift olarak gösterilebilir (Comaniciu ve ark., 2000). Battacharyya katsayısı adı verilen benzerlik ölçüsünün maksimize edilmesine dayanan yinelemeli bir yerleştirme prosedürüdür. Kontur takibi ise nesnelerin çevre sınırlarının tespit edilmesine dayanmaktadır. Kontur takibi yöntemleri, önceki kareden başlatılan ilk konturu geçerli karedeki yeni konumuna yinelemeli olarak geliştirir. Kontur takibine yönelik bu yaklaşım, gradyan inişini (gradient decent) kullanarak kontur enerjisini en aza indirerek doğrudan konturu geliştirmeye yöneliktir.

**Filtreleme ve Veri İlişkilendirilmesi:** Filtreleme ve veri ilişkilendirilmesi ise bir tündengelem yöntemidir. Bu yöntem sahne veya obje hakkında önceden bilginin dahil edilmesini, obje dinamikleri ile ilgilenmeyi ve farklı hipotezlerin değerlendirilmesini içerir. Bu yöntemler, karmaşık nesnelerin izlenmesinin yanı sıra, engellerin arkasında hareket eden nesnelerin izlenmesi gibi daha karmaşık koşullarda da nesne takibini mümkün kılar (Black ve ark., 2003). Ek olarak, görüntülerin alındığı kaynak sabit değil de hareketli olduğunda karmaşıklık artar; kamera sisteminin gerekli dinamiklerini ve bant genişliğini azaltmak için görüntü kaynağının sabit olması önerilmektedir. Bunun için İHA'lerde görüntü kompleksitesi oldukça fazla olacaktır. Bu sebeple bu sistemlerde kullanılacak nesne takip uygulamalarının oldukça başarılı ve kompleks durumlarda bile çalışabilir olması gerekmektedir. Bu kompleksiteyi çözebilecek algoritmalar için ise hesaplama karmaşıklığı daha yüksek olacaktır. Bazı sık kullanılan filtreleme algoritmaları aşağıda verilmiştir:

- Kalman Filtresi,
- Parçacık Filtresi.

Kalman filtresi Gauss gürültüsüne maruz lineer fonksiyonlar için optimal özyinelemeli bir Bayesian filtresidir. Gürültü (rastgele varyasyonlar) ve diğer yanlışlıkları içeren zaman içinde gözlemlenen bir dizi ölçüm kullanan bir algoritmadır (Arulampalam ve ark., 2002). Parçacık filtresi ise doğrusal olmayan ve Gauss olmayan süreçlerin altta yatan durum-uzay dağılımını örneklemek için yararlıdır (Maggio ve Cavallaro, 2011). Kalman filtresi ve parçacık Filtresi daha ayrıntılı olarak ilerleyen bölümlerde anlatılacaktır.

### 3.2. İstatistiksel Öğrenme

İstatistiksel öğrenmede temel kavramlar veriler ve hipotezlerdir. Veriler burada “kanıttır” yani, alanı tanımlayan rasgele değişkenlerin bazılarının veya hepsinin örnekleridir. Hipotezler, alanın nasıl çalıştığına dair olasılık teorileridir ve mantıksal teorileri özel bir durum olarak içermektedir. İstatistiksel tahmin metotları temel olarak daha önceden elde edilmiş veriler ışığında, bir ihtimalin gerçekleşme ihtimalini istatistiksel veriler ve belirli bir formüle dayanarak tahmin edilmesi için kullanılır. Genel olarak istatistiksel tahmin metotlarının büyük bir çoğunluğu Thomas Bayes tarafından bulunmuş olan Bayes Teoremi’ne dayanır. Bayes Teoremi bir sonraki bölümde ayrıntılı şekilde anlatılacaktır.

İstatistiksel öğrenme yöntemleri genel olarak **gözetimli (supervised) öğrenme** ve **gözetimsiz (unsupervised) öğrenme** olmak üzere iki ana başlık altında incelenmektedir.

**Gözetimli Öğrenme:** Gözetimli öğrenmede, çıkış değişkeninin ne olacağı daha önceden belirlenmiştir ve girdi değeri ile bu çıkış değişkeni arasında bir ilişki bulunmaya çalışılır. Çıkış değişkeni öğrenme işleminde kullanıcıya yol gösterir (supervising). Gözetimli öğrenme kategorisinde yer alan metotlara Lineer regresyon, lojistik regresyon, genelleştirilmiş ilave modeller (generalized additive models), yükseltme (boosting) ve SVM örnek olarak gösterilebilir.

**Gözetimsiz Öğrenme:** Ancak gözetimsiz öğrenme gözetimli öğrenmeye nispeten daha zorlu bir süreçtir. Bu durumda daha önceden bilinen bir çıkış değişkeni yoktur. Dolayısıyla öğrenme sürecinde modele eğitim sırasında yardımcı olacak bir değişken yoktur. Örneğin kümeleme (clustering) yöntemi gözetimsiz öğrenme kategorisinde yer almaktadır. Sade bir şekilde ifade edilecek olursa gözetimsiz öğrenmede eldeki veriler ışığında bir verinin hangi kategoriye gireceği tahmin edilmeye çalışılır.

**Neden İstatistiksel Öğrenme:** Daha önce de bahsedildiği üzere günümüzde birçok İHA uygulaması yapılmaktadır. Kimi uygulamada istatistiksel öğrenme metotlarına yer verilirken kimi uygulamada da makine öğrenmesi veya derin öğrenme metotları kullanılmaktadır. Hangi metodun nerede ve nasıl kullanılması gerektiğine Vladimir Naumovich Vapnik yazdığı kitapta açıkça anlatmıştır (Vapnik, 1998).

Aslına bakılırsa istatistiksel öğrenme, makine öğrenimi araştırmasında üçüncü ana akımdır, yani makine öğrenmesi uygulamalarında kullanılmaktadır. Makine öğreniminde hesaplamayla sistemin performansını artırmak için deneyim kullanılır. Bilgisayar sistemlerinde

"deneyim" genellikle "veri" biçimindedir. İstatistiksel öğrenmede ise tahmin ve hesaplama için olasılık modelini hesaplama yoluyla elde etmek için verileri kullanılır. İstatistiksel öğrenme ile yapılan temel varsayım, aynı özelliklere sahip olan verilerin istatistiksel olarak düzenli olmasıdır. Bu nedenle, bu veriler olasılıklı yöntemlerle işlenebilir. Örneğin, özellikleri tanımlamak için rassal değişkenler kullanılabilir; Verilerin istatistiksel düzenliliği tanımlanmak için ise olasılıklı dağılımları kullanılabilir.

İstatistiksel öğrenme metotlarının en büyük başarılarından biri destek vektör makineleri (SVM) ve çekirdek (kernel) yöntemleridir. Destek vektörü, VC boyutu ve ampirik risk minimizasyonu kavramı gibi istatistiksel öğrenme teorileri 1960-1970'li yıllarda kurulmuştur. Bununla birlikte, istatistiksel öğrenme 1990'lı yıllara kadar makine öğrenmesinin ana hatları haline gelmemiştir. Bir yandan etkili bir SVM algoritması 1995 yılına kadar önerilmemiş ve belge sınıflandırması üzerinde üstün performans göstermiştir. Öte yandan, bağlantıcılığın (connectivism) sınırlandırılmasının bulunmasından sonra insanlar, dikkatlerini istatistiksel öğrenme tekniklerine çevirmiştir.

İstatistiksel öğrenmenin, bağlantıcılığın "deneme yanılma" yöntemi ile karşılaştırıldığında büyük bir gelişme olduğu aşikardır; ancak aynı zamanda kendi sınırlamaları da bulunmaktadır. İlk olarak, teorik olarak bir kernel haritalaması kullanarak zor bir sorun çözülebilir, ancak uygun bir kernelin nasıl seçileceği tecrübeye bağlıdır. İkincisi, hem istatistiksel öğrenme hem de connectivism, karmaşık veri ve karmaşıklık veri ilişkilerini temsil edemeyen özellik vektörünü kullanır. Özellik vektörü gösterimi, alan bilgisi kullanmayı pek zor bir hale getirir ve öğrenme sonucu genellikle bir "kara kutu" dur. Son olarak, klasik istatistiksel öğrenme teknikleriyle bazı istatistiksel özellikleri korumak veya sorunları basitleştirmek için yapılan bazı varsayımlar vardır; ancak birçok gerçek zamanlı uygulamalarda doğruyu alamazlar.

Aslına bakılırsa istatistiksel öğrenme ve makine öğrenmesi arasındaki çok fazla bir farklılık yoktur. Ancak istatistiksel öğrenmede bulunan öğelerin makine öğrenmesindeki karşılıkları basitçe Çizelge 3.1 ile özetlenebilir.

Çizelge 3.1. İstatistiksel öğrenme ve makine öğrenmesinin karşılaştırılması

İstatistiksel Öğrenme	Makine Öğrenmesi
Tahmin	Öğrenme
Sınıflandırıcı	Hipotez
Veri noktası	Örnek/Durum
Regresyon	Gözetimli Öğrenme
Sınıflandırma	Gözetimli Öğrenme
Ön değişken	Özellik
Tepki	Etiket

### 3.2.1. Bayes Teoremi

Bayes teoremi rassal bir süreç sırasında meydana gelen rastgele bir olay 'A' ile diğer bir rastgele olay 'B' (eğer B için kaybolmamış olasılık var ise) için koşullu olasılıkları ve marjinal olasılıkları arasındaki ilişkidir, yani Bayes teoremi kısaca formüle edilecek olursa (Denklem (3.1)),

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad (3.1)$$

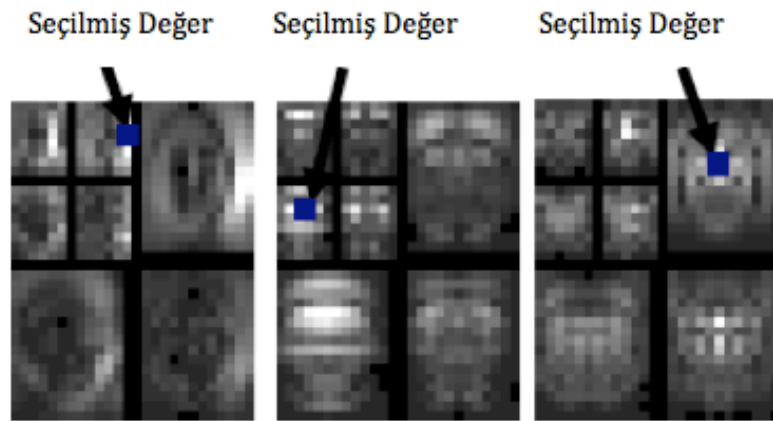
Bu denklemde;

- $P(A)$  değişkeni 'A' için önsel (prior) olasılık veya marjinal olasılık adı verilir. Bu değişken önseldir, çünkü 'B' olayı hakkında önceden herhangi bir bilgiyi içermemektedir.
- $P(A|B)$  değişkeni bilinen 'B' için 'A'nın koşullu olasılığıdır.
- $P(B|A)$  değişkeni bilinen 'A' için 'B'nin koşullu olasılığıdır
- $P(B)$  değişkeni 'B' olayı için önsel (prior) olasılıktır veya 'B'nin marjinal olasılığıdır ve matematiksel olarak normalize eden bir sabittir.

Bu şekilde Bayes teoremi, fazla matematiksel olmadan, basit bir şekilde sezgiye dayanarak şöyle açıklanabilir: Bayes teoremi eğer ‘B’ olasılığı gözlemlenmiş ise, ‘A’ olasılığının gözlemi hakkındaki inançların ne şekilde güncelleştirilebileceğini ortaya çıkarır.

**Nesne Tanıma için Bayes Ağlarının Kullanılması:** İstatistiksel öğrenmenin diğer bir sık kullanılan metodu ise Bayes ağlarıdır. Bayes ağlarını daha iyi açıklayabilmek için bir yüz tanıma uygulamasına örnek gösterilebilir (Schneiderman, 2004). Uygulamada verilen görüntülerin özellikleri dalgacık (wavelet) dönüşümü kullanılarak çıkarılmış ve Bayes ağları kullanılarak bu özellikler sınıflandırılmıştır.

Öncelikle görüntüden özellik çıkarımına tekrardan değinmek gerekir. Bilindiği üzere birçok resim sınıfı, istatistiksel bağımlılığın seyrek yapısına sahiptir. Her değişken, az sayıda diğer değişkenle güçlü istatistiksel bağımlılığa ve geri kalan değişkenlerle ihmal edilebilir bağımlılığa sahiptir. Örneğin, yüzlerin, arabaların ve telefonların geometrik olarak hizalanmış görüntüleri bu özelliği sergilemektedir. Şekil 3.1’de, önden çekilmiş insan yüz görüntülerini temsil eden dalgacık değişkenleri arasındaki karşılıklı bilgiyi göstermektedir. Karşılıklı bilgi, iki değişken arasındaki istatistiksel bağımlılığın gücünü ölçer. Her görüntü, bir okla gösterilen bir seçilen dalgacık değişkeni ile dalgacık dönüşümündeki diğer tüm değişkenler arasındaki karşılıklı bilgi değerlerini gösterir. Her lokasyonda parlaklık, bu lokasyondaki değişken ile seçilen değişken arasındaki karşılıklı bilgiyi gösterir (Schneiderman, 2004).



**Şekil 3.1.** Yüz görüntülerindeki dalgacık değerleri arasındaki karşılıklı bilgi (Schneiderman, 2004)

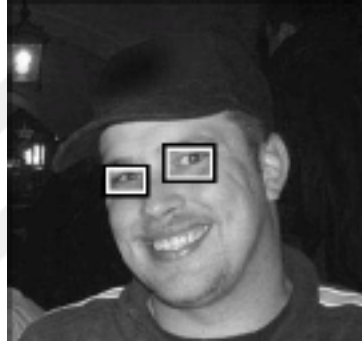
Bununla birlikte, istatistiksel bağımlılık her zaman ayrı "parçalara" ayrıştırılamaz. Bu tarz bir "istatistiksel ayrıştırılamazlık" koşullu bağımsızlık kavramını kullanarak kompakt olarak ifade edilebilir. Böyle bir durumda, olasılık dağılımı, bağımlılıkları paylaşan değişkenlerden oluşan alt kümesindeki şartlandırma ile Bayes kuramı kullanılarak parametreleştirilebilir. Böyle bir



durumda Bayes ağırları kullanılarak değişkenlerin olasılıksal dağılımları sınıflandırılabilir. Böyle bir sınıflandırıcı, logaritmik olasılık oranı testi olarak yazıldığında aşağıdaki genel formül ile ifade edilebilir (Denklem (3.2)):

$$f(X_1, \dots, X_n) = \log \frac{P(X_1, \dots, X_n | \omega_1)}{P(X_1, \dots, X_n | \omega_2)} > \lambda \quad (3.2)$$

Burada X giriş değerleridir. Verilen örnek için giriş, griye dönüştürülmüş görüntü penceresinin dalgacık dönüşümüdür.  $\omega_1$  ve  $\omega_2$  iki sınıfa işaret eder. Nesne algılama sorunu için sınıflar, algılanılması istenen "nesne" ve istenen nesne dışında olan "nesne harici" görüntülerdir. Örneğin,  $\omega_1$ , yüze karşılık gelebilir ve  $\omega_2$ , "yüz olmayan" sınıfa karşılık gelebilir. Bu sınıflayıcı,  $f(X_1, \dots, X_n) > \lambda$  ise, sınıfı  $\omega_1$ 'i seçer. Aksi halde,  $\omega_2$  sınıfını seçer. Verilen örnek için çıktı aşağıdaki şekilde görüldüğü gibi olacaktır (Şekil 3.2).



Şekil 3.2. Bayes ağırları ile yüz tanıma uygulaması (Schneiderman, 2004)

Bu uygulamada Bayes ağırları yüz tanıma için kullanılmış olup, herhangi bir İHA uygulamasında kullanılmaya elverişli olacaktır. Yüz tanıma yerine bir önceki konuda bahsedilen bir aracın tanınması veya başka bir İHA'nın tanınması ve bir önceki metot ile takibi için birlikte kullanılabilir.

### 3.2.2. Kalman Filtresi

İstatistiksel öğrenmenin yoğun olarak kullanıldığı bir diğer metot ise yine Bayes temelli olan Kalman filtresidir. KF, durum uzayı modeli ile gösterilen dinamik bir sistemde, modelin önceki bilgilerinin bilinmesiyle birlikte giriş ve çıkış bilgilerinden sistemin durumlarını önceki olaylardan faydalanarak tahmin edilebilen Bayes temelli bir filtredir. KF, 1960'lardan sonra araçlarda navigasyon amaçlı kullanılarak hayatımıza girmeye başlamasıyla, üstüne çok çeşitli uygulamalarda kullanılmış ve sistemin durumu hakkında optimize edilmiş bir tahmin sağlayabilen bir algoritmadır. Algoritma, gürültülü (ideal olarak Gauss gürültüsü) bir gözlem veri akışı (tipik olarak, sensör ölçümleri) üzerinde gerçek-zamanlı, özyinelemeli çalışarak

hatayı en aza indirmeyi amaçlayarak filtreleme yapar ve sistemin fiziksel karakteristiklerini modelleyerek üretilen gelecek bir sonraki adımın matematiksel tahminine göre hatayı optimize eder (Cuevas, 2005).

Modelin sonraki adımdaki tahmini, gözlem ile karşılaştırılır ve bu fark, Kalman kazancı olarak bilinen bir çarpan ile ölçeklendirilir. Bu kazanç daha sonra devam eden tahminleri iyileştirmek ve optimize etmek için sistem modeline bir girdi olarak geri beslenir. Kazanç, performansı iyileştirmek için istenilen şekilde ayarlanabilir. Yüksek kazanç değerlerinin kullanılması durumunda, filtrenin çıkışında alınan sonuç gözlemleri daha yakından takip eder. Düşük kazanç değerinin kullanılması durumunda ise filtre modelin tahminlerini daha yakından takip eder. KF, bilinmeyen değerlere, tek bir ölçüm yapılan sistemlere veya sadece model tahminlerine bağlı olarak elde edilebilecek tahminlerden daha başarılı tahminler üretmek için kullanılmaktadır.

Her bir zaman adımında, KF, gerçek bilinmeyen değerlerin tahminlerini, değerlerin belirsizlik değerleriyle beraber üretir. Buna belirsizlik (uncertainty) de denmektedir. Bir sonraki adımdaki ölçümün sonucu gözlemlendiğinde, bu tahminler, belirsizlik değeri düşük tahminlerin ağırlık değerleri artırılarak, ağırlıklı ortalama ile güncellenir.

Teorik olarak anlatılacak olursa, KF'nin temel olarak yaptığı iş alttaki kullanılan sistemin doğrusal ve dinamik bir sistem olduğu, ayrıca tüm hataların ve ölçümlerin Gauss dağılımına (sıklıkla çok değişkenli Gauss dağılımı) sahip olduğudur. Yönteme eklentiler de yapılarak genişletilmiş Kalman filtresi (EKF) ve kokusuz Kalman filtresi (UKF) gibi geliştirmeler de yapılmıştır. Altteki sistem modeli Gizli Markov Modeli'ne benzeyen bir Bayes modelidir ancak gizli-değişkenlerin durum uzayı süreklidir ve tüm gizli ve gözlenen değişkenlerin dağılımı Gauss'dur.

Kalman filtresinin iki tane kendine özgü özelliği vardır. Bunlardan biri “sistem-durum modeli” olarak adlandırılan matematiksel model, diğeri ise art arda hesaplanarak bulunan “sonuç” durumudur. Genellikle kalman filtresi “sistem-durum modeli” ve “ölçüm modeli” olarak tanımlanır. Sistem-durum modeli:

$$s(t) = \mathcal{O}(t-1)s(t-1) + w(t) \quad (3.3)$$

ve ölçüm modeli:

$$z(t) = H(t)s(t) + v(t) \quad (3.4)$$

burada  $\ddot{O}(t-1)$  ve  $H(t)$  sırası ile geçiş ve ölçüm matrislerinin durumudur.  $w(t)$  ve  $v(t)$  ise sıfır ortalamaya sahip beyaz Gauss gürültüsüdür.

$$E\{w(k)w^T(l)\} = Q\delta_{kl} \quad (3.5)$$

$$E\{v(k)v^T(l)\} = R\delta_{kl} \quad (3.6)$$

$\delta_{kl}$ , Kronecker delta fonksiyonunu ifade ederken, Q ve R sırası ile  $w(k)$  ve  $v(k)$ 'nin kovaryans matrisleridir. Şimdiki zaman  $t$ 'nin durum vektörü  $s(t)$  önceki tahmin ve yeni ölçüm  $z(t)$  kullanılarak tahmin edilir.

KF uygulamada iki kısma ayrılır; tahmin adımı ve doğrulama adımı. Tahmin adımı halihazırdaki adımı öne çıkarmak ve  $s^-(t)$  durumunun öncül değer tahminini elde etmekten sorumludur. Doğrulama adımın amacı ise geri besleme yapmaktır. Yani diğer bir deyişle öncül değer ölçümünü tahmin aracılığı ile daha gelişmiş bir ardıl değer ( $s^+(t)$ ) elde etmek için kullanılır.  $s^+(t)$  şu şekilde yazılabilir;

$$s^+(t) = s^-(t) + K(t)(z(t) - H(t)^T s^-(t)) \quad (3.7)$$

$K(t)$  burada ağırlık değeridir ve şu şekilde ifade edilebilir,

$$K(t) = P^-(t)H(t)^T (H(t)P^-(t)H(t)^T + R(t))^{-1} = \frac{P^-(t)H(t)^T}{P^-(t)H(t)^T + R(t)} \quad (3.8)$$

yukarıdaki denklemde  $P^-(t)$  öncül tahmin hata kovaryansıdır. Aşağıdaki şekilde ifade edilir,

$$P^-(t) = E[e^-(t)s^-(t)^T] \quad (3.9)$$

burada  $e^-(t) = s(t) - s^-(t)$  öncül tahmin hatasıdır. Ek olarak ardıl tahmin hata kovaryansı  $P^+(t)$  şu şekilde ifade edilir,

$$P^+(t) = E[e^+(t)s^+(t)^T] \quad (3.10)$$

burada  $e^+(t) = s(t) - s^+(t)$  ardıl tahmin hatasıdır. Tahmin adımı ve doğrulama adımı art arda aşağıda gösterildiği şekilde uygulanır. Tahmin adımı:

$$s^-(t) = \ddot{O}(t-1) s^+(t-1) \quad (3.11)$$

$$P^-(t) = \ddot{O}(t-1)P^+(t-1)\ddot{O}(t-1)^T + Q(t-1) \quad (3.12)$$

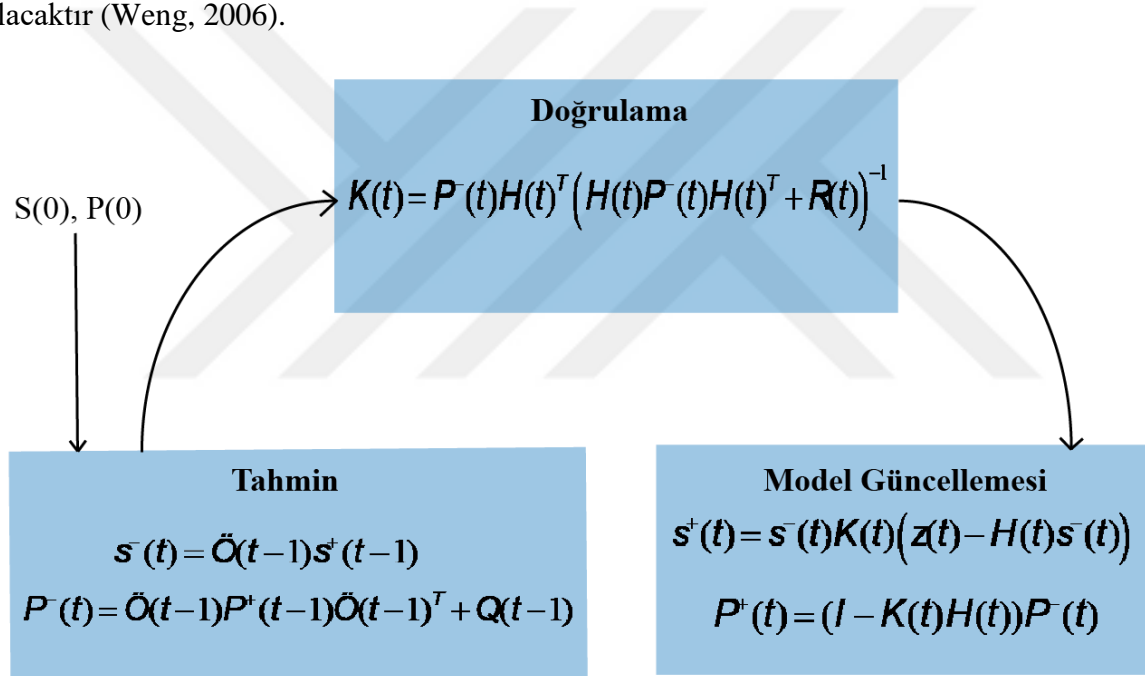
ve doğrulama adımı:

$$K(t) = s^-(t)H(t)^T (H(t)P^-(t)H(t)^T + R(t))^{-1} \quad (3.13)$$

$$s^+(t) = s^-(t)K(t)(z(t) - H(t)s^-(t)) \quad (3.14)$$

$$P^-(t) = (I - K(t)H(t))P^-(t) \quad (3.15)$$

Tahmin-doğrulama döngüsü tekrarlanır. Denklem (3.13)'e bakıldığında ölçme hatası  $R(t)$  ve Kalman kazancı  $K(t)$  ters orantılıdır.  $R(t)$ 'nin küçük olması ile  $K(t)$  kazanı daha ağır tartar. Bu durumda ise ölçüme olan güven daha da artarken tahmin sonucuna olan güven azalır. Diğer bir açıdan öncül tahmin hatası  $P^-(t)$  sifira yaklaştıkça  $K(t)$  kazancı kazanı daha hafif tartar ve bu durumda ölçüme olan güven daha da azalırken tahmin sonucuna olan güven daha da artar. Böylece hangi duruma güvenilebileceğine karar verilebilirse sistem optimum sonuca daha da yaklaşır. Böylece art arda Denklem (3.11)'den (3.15)'ye kadar olan Kalman filtresi döngüsü tekrarlanır. Kalman filtresini özetleyecek olursak blok diyagramı Şekil 3.3'deki gibi olacaktır (Weng, 2006).



Şekil 3.3. Kalman Filtresi blok diyagramı

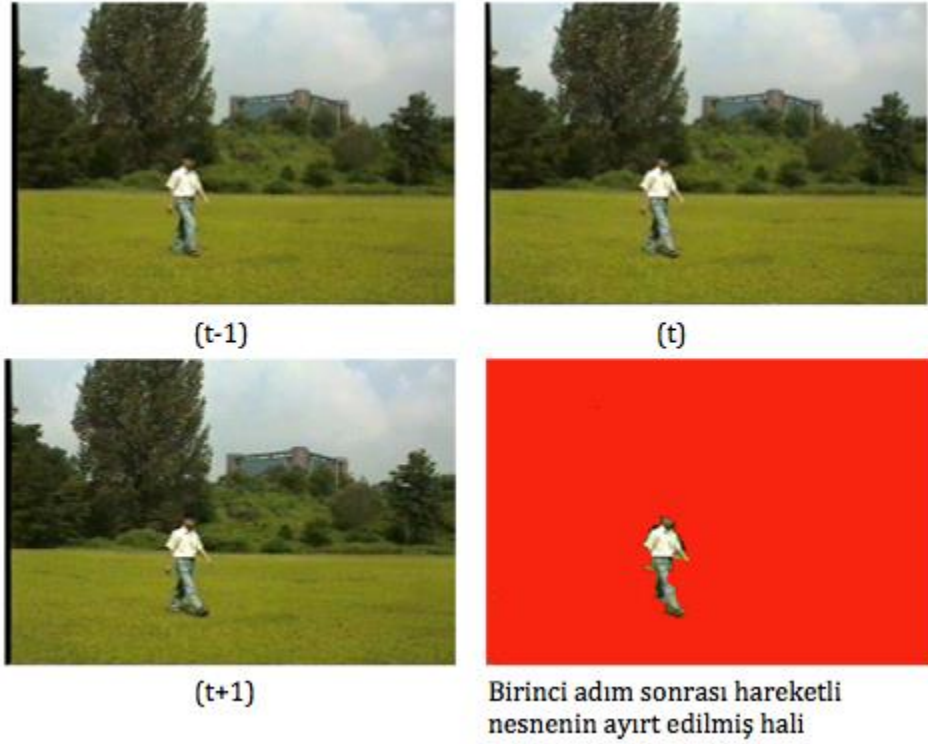
KF'nin kullanıldığı en göze çarpan uygulamalardan birini Shiu-Ku Weng ve arkadaşları yapmıştır (Weng ve ark., 2006). Uygulamada adaptif bir KF kullanılarak video görüntülerinde hareketli nesne takibi yapılmıştır. Takip edilecek nesne bir ara yüz aracılığı ile kullanıcı tarafından seçilmiş ve KF kullanılarak seçilen nesne takip edilmiştir.

Gerçekleştirilen uygulama dört ana adımdan oluşmaktadır;

- 1) Seçilmiş hareket eden nesnenin frame farklılığı ve bölge büyütülmesi (region growing) metotları ile görüntüden ayırt edilmesi,

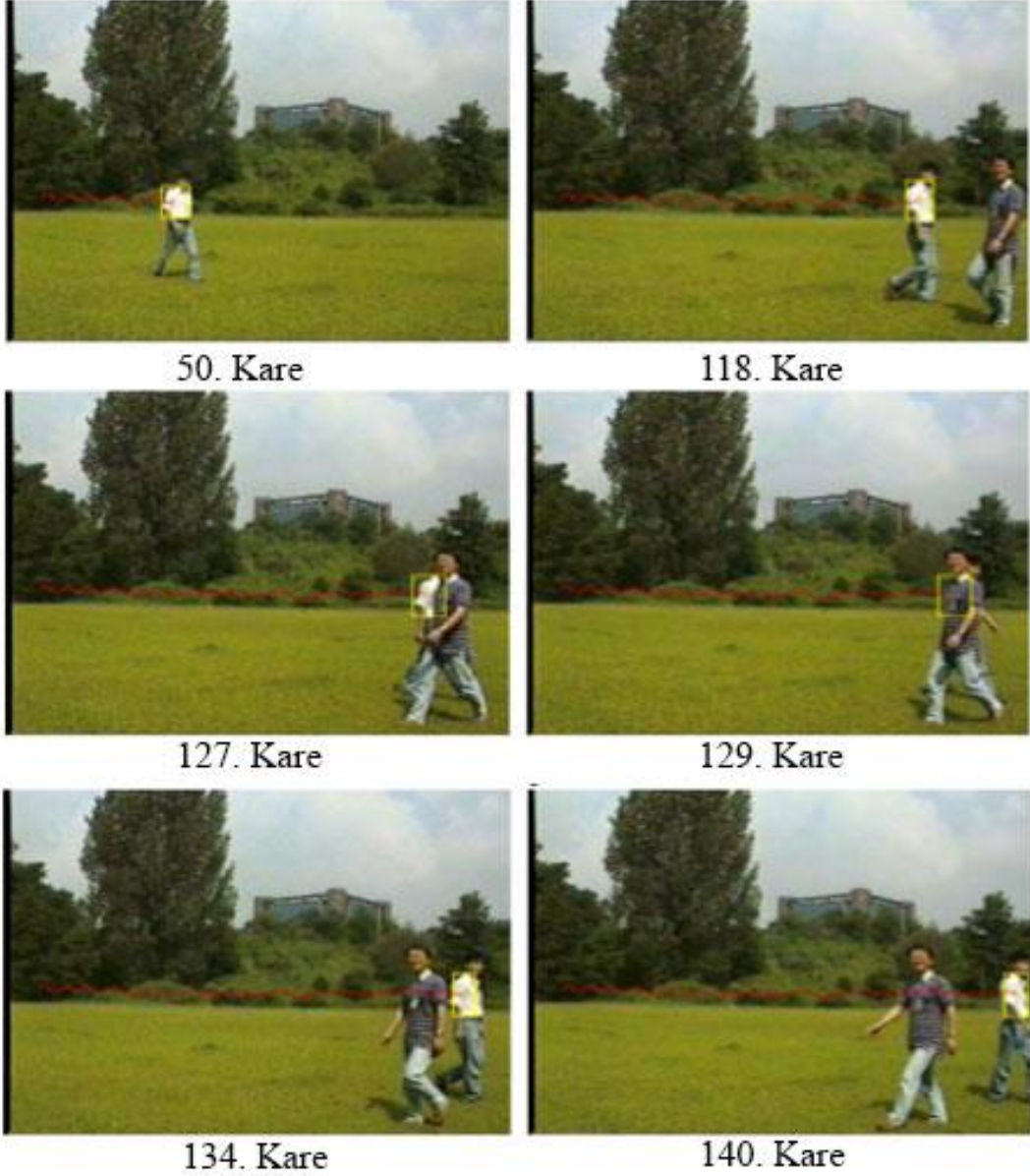
- 2) Nesnedeki özelliklerin çıkarılması,
- 3) Görüntünün HSI uzayına dönüştürülmesi,
- 4) Kalman filtresi ile nesnenin takibi.

İlk adımda art arda 3 kare seçilmiş ( $t-1$ ,  $t$ ,  $t+1$ ), bu kareler birbirinden çıkartılmış ve bölge büyütülmesi algoritması ile nesne görüntüden Şekil 3.4'de görüleceği üzere ayırt edilmiştir.



**Şekil 3.4.** Kare farklılığı ve bölge büyütülmesi (region growing) metotları ile görüntüden ayırt edilmesi  
(Weng ve ark., 2006)

İkinci adımda K-means algoritması kullanılarak hareketli nesnenin baskın RGB değeri bulunmuş ve bu özelliği çıkarılmıştır. Üçüncü adımda ise Kalman filtresi uygulanmış, nesne ayırt edilmiş ve özellikleri çıkarılmış nesneni takibi Şekil 3.5'de görüldüğü gibi gerçekleştirilmiştir.



**Şekil 3.5.** Kalman filtresi kullanılarak hareketli nesne takibi (Weng ve ark., 2006)

Bu metodun en güzel özelliklerinden bir tanesi de nesnenin şeklinin değişmesi (dönmesi, bir kısmının kareden çıkması veya başka bir nesnenin arkasına geçmesi) durumunda dahi KF'nin içerdiği Bayes tahmini ile nesnenin takibine devam etmesidir. Bu uygulamaya geliştirme olarak bir nesne tanıma algoritması eklenerek kullanıcının nesneyi seçmesini elimine ederek uygulamanın tam otonom hale getirilmesi söz konusu olabilir. İHA ile nesne takibine elverişli bir uygulamadır.

### 3.2.3. Parçacık Filtresi

Nesne takibi uygulamalarında kullanılan algoritmanın işlem yükünün yanında değişken ortam koşulları ve takip edilen cisminin hızı, oryantasyonu, şekli, boyutu ve rotasının değişimlerine duyarlı olması gerekmektedir. KF Gauss gürültüsüne sahip ortamlarda lineer bir rota izleyen nesnelere takibinde oldukça başarılıdır. Örneğin düz giden bir arabanın takibinde kullanılan KF oldukça başarılı olacaktır. Bu herhangi lineer bir rotayı takip eden bir cisim için geçerlidir. Bu rota düz bir çizgi, dairesel ya da eliptik olabilir. Ayrıca KF'nin işlem yükü oldukça düşük olacaktır. Fakat lineer olmayan bir rotayı takip eden herhangi bir nesne için bu yöntem yeteri kadar esnek olmayacaktır. Bunun için yine istatistiksel bir temele dayanan PF kullanmak daha faydalı olacaktır. KF'e kıyasla PF'nin işlem yükü daha yüksektir ancak daha esnek bir modeldir. Sistem doğrusal değil veya görüntülerde Gauss gürültüsünden farklı bir gürültü varsa PF oldukça başarılı bir sonuç verecektir. Örneğin düz bir araba değil de sürekli ve hızlıca yön değiştiren bir topun takip edilmesi gerekir ve dış ortam nedeniyle oluşan gürültü sürekli değişiklik gösteriyorsa KF yerine PF kullanılmalıdır.

Parçacık filtreleri veya diğer adıyla Sıralı Monte Carlo (SMC) yöntemleri, sinyal işleme ve Bayes istatistiksel çıkarımında ortaya çıkan filtreleme sorunlarını çözmek için kullanılan bir Monte Carlo algoritmalarıdır. Filtreleme problemi, kısmi gözlemler yapıldığında dinamik sistemlerde iç durumları tahmin etmekten ve dinamik sistemde rasgele bozulmalar bulunduğu için oluşur. Amaç, bazı gürültülü ve kısmi gözlemler göz önüne alındığında, Markov sürecinin durumlarının sonraki adımdaki dağılımlarını hesaplamaktır. "Parçacık Filtresi" terimi ilk olarak Del Moral ve ark. (1996) tarafından ortaya atılmıştır. PF, Gaussian olmayan gürültülü ve/veya kısmi gözlemler verilen bazı rassal işlemlerin arka dağılımını temsil eden bir dizi parçacığı (ayrıca örnekler olarak da adlandırılır) kullanır. Sistemin durum uzayı modeli doğrusal olmayabilir ve nesnenin ilk durum ve gürültü dağılımları herhangi bir şekilde olabilir. PF teknikleri, durum-uzay modeli veya durum dağılımları hakkında varsayımlar gerektirmeden gerekli dağıtımdan örnekler üretmek için iyi kurulmuş bir metodoloji sağlar ((Del Moral ve ark., 1996) (Del Moral, 1998)). Bununla birlikte, bu yöntemler çok yüksek boyutlu sistemlere uygulandığında iyi performans göstermeyebilir.

En temel PF, Bayesian filtrenin doğrudan bir türü olarak görülebilir (Salmond ve ark., 2005). Örneğin ( $k > 0$ ) için  $p(x_{k-1}|Z_{k-1})$  olan olasılık yoğunluk fonksiyonundan (pdf) gelen  $N$  sayıda örneğe sahip olduğu düşünülürse, bu örnekler (parçacıkları)  $\{x_{k-1}^{i*}\}_{i=1}^N$  olarak belirtilecektir (Hue, 2002).

**Tahmin** aşamasındaki basit algoritma her  $k$  zaman diliminde bir dizi örnek oluşturmak için önceki oluşturulan örneklerin her  $k-1$  zaman diliminde sistem modelinden geçirilmesidir (Howard ve ark.).

$$x_k = f_{k-1}(x_{k-1}, v_{k-1}), \quad k > 0 \text{ iken} \quad (3.16)$$

Bu durumda örnekler  $\{x_k^i\}_{i=1}^N$  olduğu durumda,

$$x_k^i = f_{k-1}(x_{k-1}^{i*}, v_{k-1}^i) \quad (3.17)$$

ve  $v_{k-1}^i$  sistem gürültüsünün olasılık yoğunluk fonksiyonundan çizilen (bağımsız) bir örnek olacaktır. Bu önceki pdf  $p(x_{k-1}|Z_{k-1})$ 'den parçacık (örnek) üreten basit bir prosedürdür.

**Güncelleme** aşamasında  $z_k$ 'ya bağlı olarak önceki örnekleri güncellemek için her parçacık için bir  $\tilde{\omega}_k^i$  ağırlık değeri hesaplanır. Bu ağırlık, önceki örneğin değerinde değerlendirilen ölçüm olasılığıdır:  $\tilde{\omega}_k^i = p(z_k|x_k^i)$ . Daha sonra bu ağırlıklar normalize edilerek toplanır (3).

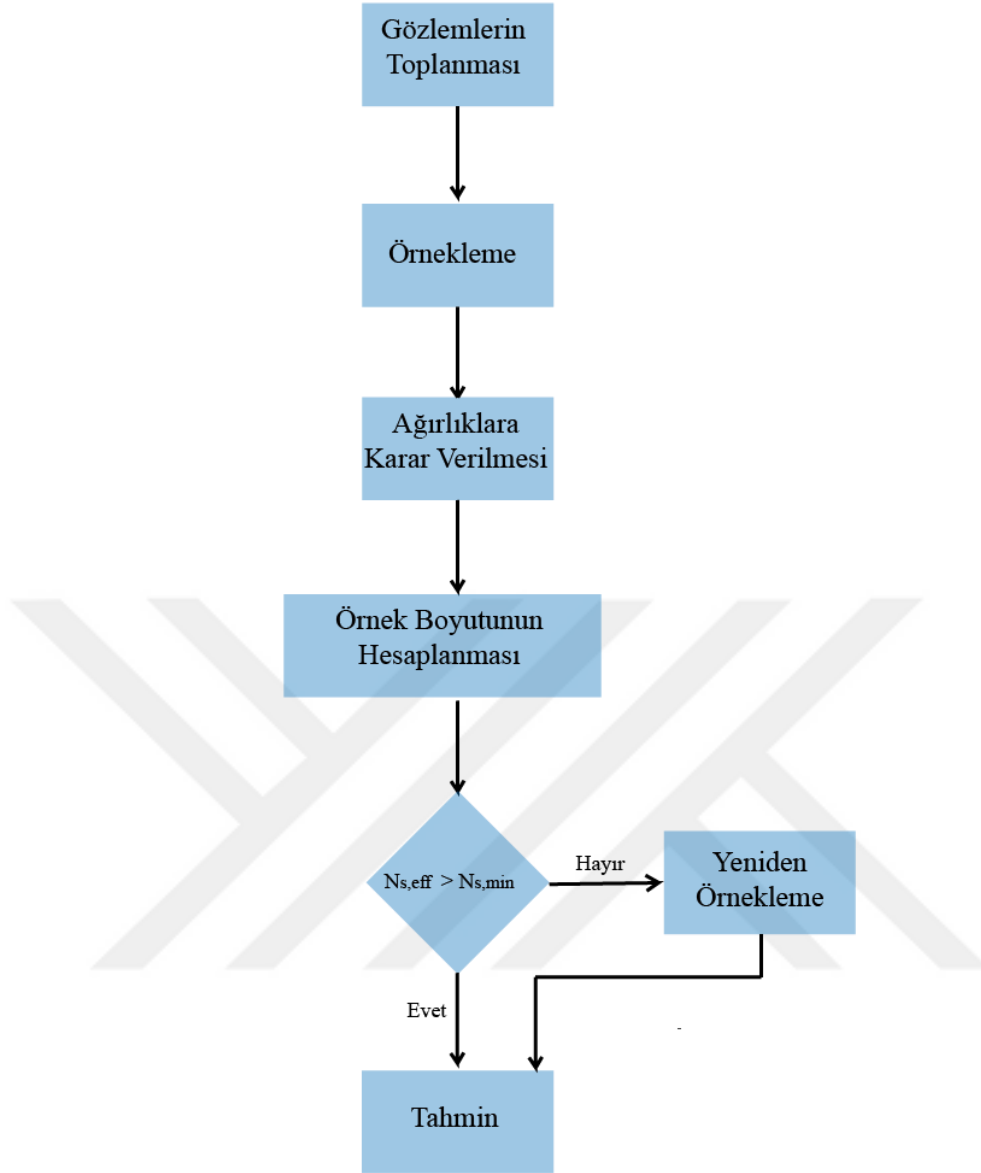
$$\omega_k^i = \tilde{\omega}_k^i / \sum_{j=1}^N \tilde{\omega}_k^j \quad (3.18)$$

Daha sonra bu parçacıklar tekrardan örneklenecektir. Bu örnekleme işlemi ağırlıkları düşük olan parçacıkların kaldırıp yeni bir parçacığın ağırlıklı parçacıklara yakın bir bölgeye konulması ile gerçekleştirilir (4).

$$\text{Parçacıklar} \{x_k^{i*}\}_{i=1}^N \text{ iken her } i, j \text{ değerleri için } Pk\{\omega_k^{i*} = x_k^j\} = \omega_k^j \quad (3.19)$$

Diğer bir deyişle olasılığı normalize edilmiş ağırlığına eşit olan bir set örnek alınır ve bu adım  $N$  defa tekrar edilir. Sonuçta yeni örnek seti  $\{x_k^{i*}\}_{i=1}^N$  oluşur. Sonuç olarak bu yeni parçacık dizisi gerekli pdf  $p(x_k|Z_k)$ 'nin örnekleridir (Salmond ve ark., 2005). Böylece algoritmanın bu döngüsü sonlandırılır. Basitçe özetlenecek olursa PF'de öncelikle gözlemler toplanarak örnekleme işlemi yapılır ( $\{x_{k-1}^{i*}\}_{i=1}^N$ ). Daha sonra ağırlıklara karar verilerek örneklerin boyutu hesaplanır. Eğer örnek değerinin ağırlığı minimumdan büyükse tahmin adımına geçilir aksi takdirde yeniden örnekleme yapılır. PF'nin akış diyagramı Şekil 3.6'da gösterilmiştir.





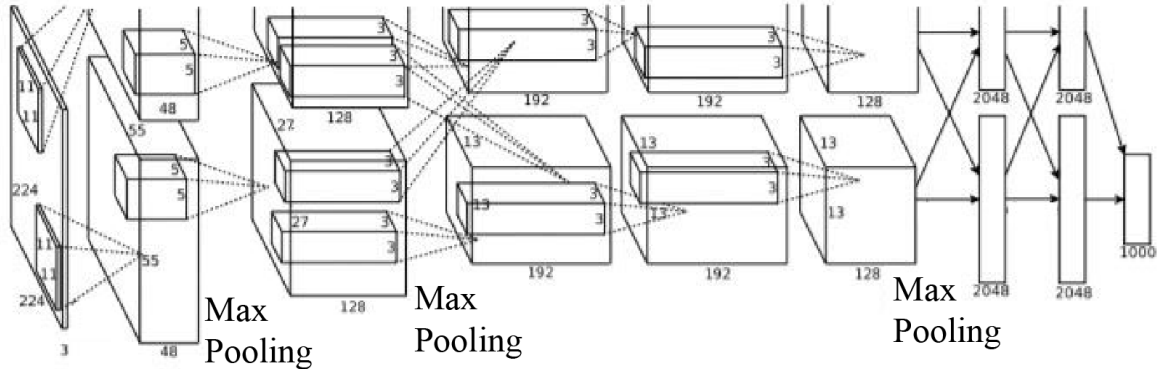
Şekil 3.6. Parçacık Filtresi akış diyagramı

### 3.3. Derin Öğrenme

Derin öğrenme; bilgisayarlara, insanlara doğal gelen şeyleri yapmalarını öğreten bir metot olup, aslında makine öğrenme metotlarının alt alanıdır: örneklerle öğrenim (supervised learning). Derin öğrenme, şoförsüz araçların ardındaki, bir dur işaretini tanımalarını veya bir yayayı bir elektrik direğinden ayırmalarını sağlayan kilit bir teknolojidir. Telefonlar, tabletler, TV'ler ve eller serbest hoparlörler gibi tüketici cihazlarında ses kontrolünün anahtarıdır. Derin öğrenme son zamanlarda ve birçok sebepten dolayı çok dikkat çekmektedir. Derin öğrenme aracılığıyla her geçen gün daha önce mümkün olmayan sonuçlara ulaşılmaktadır. Derin öğrenmede, bir bilgisayar, sınıflandırma görevlerini doğrudan görüntü, metin veya seslerden

yapmayı öğrenir. Derin öğrenme modelleri, bazen insan seviyesindeki performansı aşarak, en gelişmiş doğruluğu elde edebilir. Modeller, çok sayıda etiket içeren çok sayıda etiketli veri ve sinir ağı mimarisi kullanılarak eğitilmiştir. Derin öğrenme bahsedildiği üzere birçok konuda kullanılmaktadır. Bu konuların en önemlilerinden biri de nesne tanıma ve nesne takibidir. Son yıllarda derin öğrenme destekli İHA'lar hızlı bir artış göstermiştir. Derin öğrenmenin İHA'larda kullanılan akıllı kontrol sistemlerine olan faydası yadsınamaz bir gerçektir.

Derin öğrenme ile nesne sınıflandırma ve takibinin yapıldığı halihazırda en yaygın kullanılan metot ise evrimsel sinir ağlarıdır. CNN'lerin kullanımı aslında çok da eskiye dayanmamaktadır. Aslında Krizhevsky ve ark. (2012) görüntü sınıflandırma için CNN'leri kullanarak görüntü işleme camiasında benchmark yarışmalardan biri olan ImageNet yarışmasında birinciliği elde etmesi, bir anda herkesin gözünü CNN'lere çevirmesine sebep olmuştur. Çok geçmeden CNN beklenilene vermiş ve görüntü sınıflandırmada başarı oranı bir insanın yapabileceğinin de üstüne çıkmıştır. Bir CNN yakından incelendiğinde azaltılmış işlem gereksinimleri için tasarlanmış çok katmanlı bir perceptron modeline benzer bir sistemi kullanır (Rosenblatt, 1958). Bir CNN'nin katmanları, bir giriş katmanı, bir çıkış katmanı ve çoklu evrimsel katmanlar, örnekleme (pooling) katları, tamamen bağlı katmanlar (fully connected layer) ve normalizasyon katmanını içeren gizli bir katmandan oluşur. Bir CNN mimarisi aşağıdaki şekilde daha açıkça Şekil 3.7'de gösterilmiştir.



Şekil 3.7. Evrimsel yapay sinir ağı mimarisi (Krizhevsky ve ark., 2012)

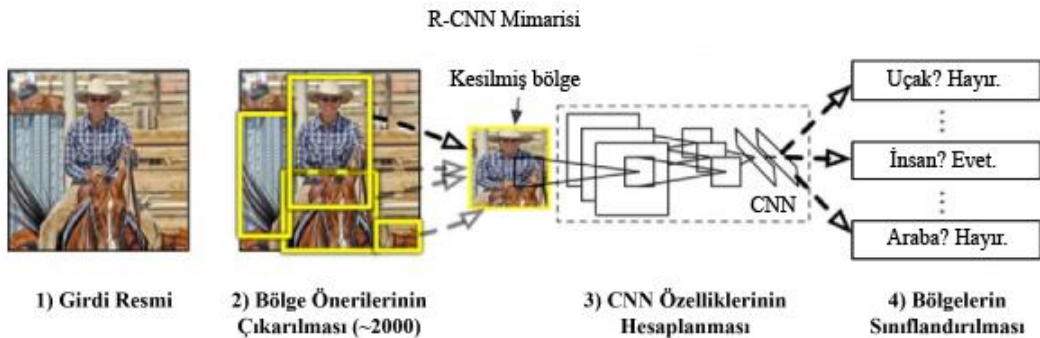
İlk evrimsel katmanda görüntü konvole edilerek parçalara bölünür. Örneğin 32x32x3 boyutunda bir görüntü varsayılırsa 5x5'lik bir pencerenin görüntü üzerinde adım adım gezdirilerek 5x5'lik parçaların ayrıldığı düşünülebilir. Bu görüntü 5x5'lik çekirdek (kernel) ile çarpılır. Tüm değerler matriste eleman eleman çarpılır ve tüm değerlerin toplamı çıkış matrisinin ilgili elemanı olarak kaydedilir. Buna çapraz korelasyon ilişkisi de denir. Özet olarak bu katman özelliklerin saptanması için kullanılır. Örnekleme (pooling) katmanında ağırlık sayısı

azaltılır ve bu ağırlıkların uygunlukları kontrol edilir. Tamamen bağlı katmanlar ise sınıflamada kullanılan standart sinir ağıdır.

### 3.3.1. Regions with CNN Features (R-CNN)

CNN'ler ile nesne tespiti üzerine öncü çalışmalardan birini Girshick ve ark. (2014) gerçekleştirmişlerdir. R-CNN'lerin amacı aslına bir görüntüyü alarak içindeki nesnelere çerçeve (bounding box) denilen kareler içine alarak bunlara hangi cisim olduğu bilgisini etiketlemektir. Sonuç olarak metod, aslında iki aşamadan oluşmaktadır. Bu çalışmada standart CNN'e görüntü üzerinde hangi bölgelerde nesne olabileceğinin aramasını yapması için bir bölge önerisi (region proposal) metodu olan seçici tarama yöntemi (Selective Search) kullanılmıştır. Daha ayrıntılı açıklamak gerekirse seçici tarama; görüntüyü farklı boyutlardaki pencerelerle tarar ve her bir boyutta pencere için bitişik pikselleri doku, renk ve yoğunluklarına göre gruplandırarak nesnelere aslında bir nesne olup olmadıklarını bulmaya çalışır. Yani biraz daha basit anlatmak gerekirse bir grup kutucuk resim üzerinde sunulur ve aslında bu kutucukların birer nesne olup olmadığına bakılır. Bu bölge önerisi (region proposal) metodunun CNN ile birleştirilmiş haline Regions With CNN Features (R-CNN) denmektedir. Adımlarına tek tek değinmek gerekirse, R-CNN Şekil 3.8'deki gibi uygulanmaktadır:

- 1) Bir girdi olarak görüntünün verilmesi,
- 2) Nesnelere olabileceği yerlerin (bounding box) sunulması (region proposal),
- 3) Sunulmuş olan bu bölgelerin daha önceden eğitilmiş CNN'den geçirilmesi ve özelliklerinin çıkarılması,
- 4) Son olarak eski bilgilere dayanılarak SVM metodu ile sunulan bölgedeki nesnenin hangi nesne olduğunun bulunması ve bölgelerin lineer regresyon modelinden geçirilerek sınıflandırılmış olan nesnenin koordinatlarının daha kesin bir biçimde bulunması.

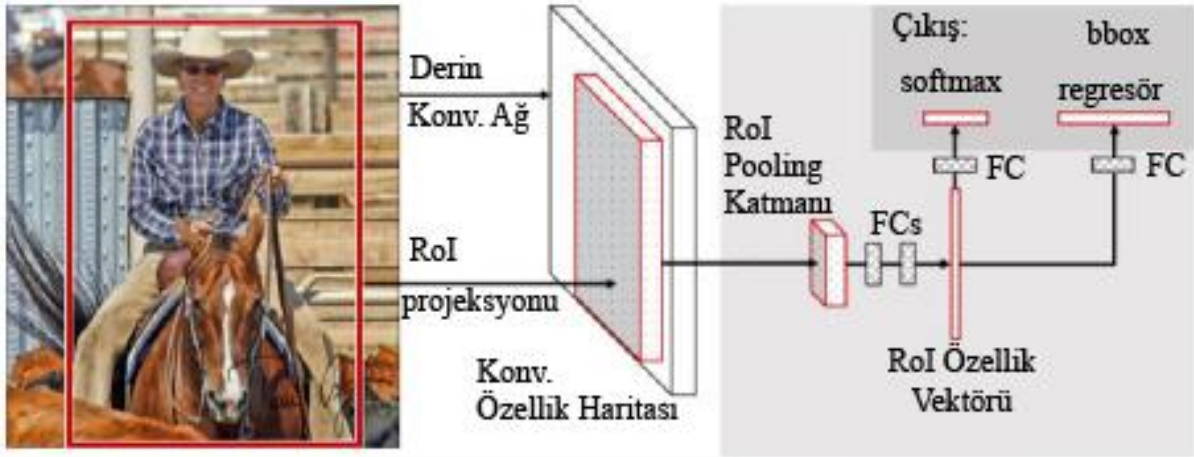


Şekil 3.8. R-CNN çalışma yapısı (Girshick ve ark., 2014)

R-CNN'in geliştirilmesi ile derin öğrenmenin popülaritesi daha da artmıştı. Ancak R-CNN'lerin başarılı sonuçlar vermesine karşın gerçek zamanlı uygulamalarda kullanılması için hızlandırılmasına ihtiyacı vardı.

### 3.3.2. Fast R-CNN

R-CNN'lerin başarısı oldukça etkileyiciydi ancak oldukça yavaş çalışıyorlardı. Bunun sebebi ise R-CNN metodunda bir resim için sunulmuş olan bölgelerin her birinin (yaklaşık 2000 tane) tek tek CNN'den geçirilmesi işlem yükünü oldukça artırmaktaydı ve bu da gerçek zamanlı uygulamaları imkansız hale getirmekteydi. Bunun için Girshick ve ark. (2015) var olan metodu geliştirmek için bir çalışma yürütmüş, bulmuş oldukları yeni yöntem ile bunu bir adım öteye taşıyarak biraz daha hızlı çalışır hale getirmişlerdir. Araştırmacılar Fast R-CNN metodunda sunulan bölgelerin bir arada toplanıp tek seferde CNN'den geçirebileceklerini düşünmüşlerdir. Bu işleme RoIPool denmektedir. Önceki uygulamada; (R-CNN) görüntü özelliklerinin çıkarılması (CNN), sınıflandırma (SVM) ve koordinatların kesinleştirilmesi (Regressor) işlemleri için farklı modeller kullanılmasına karşın bu metotta bütün bu işlemler için tek bir yapay sinir ağı kullanılmıştır. Bu ağda Şekil 3.9'da görüleceği gibi, özellik çıkarımı, sınıflandırma ve regresyon işlemlerinin tamamı tek bir ağ üzerindedir.

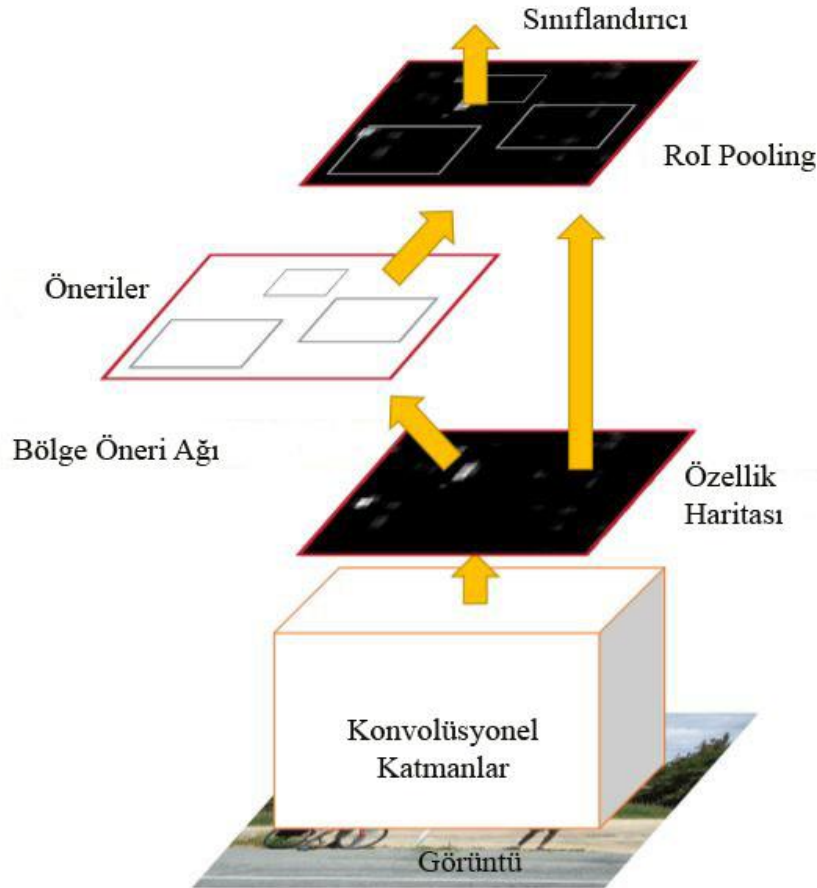


Şekil 3.9. Fast R-CNN mimarisi (Girshick, 2015)

Görüldüğü üzere özellik çıkarım işlemi evrimsel katman ve örnekleme (pooling) katmanında yapıldıktan sonra SVM sınıflandırıcı yerine bir eşiksiz en büyük işlev fonksiyonu katmanının (softmax) konulduğu görülmektedir. Regresyon işlemi ise lineer regresyonla değil çerçeve regresyonu (bbox regressor) katmanında yapılmaktadır. Böylece üç farklı metodun ayrı ayrı kullanılarak işlem gerekliliğini yavaşlatmak yerine bütün işlemler tek bir ağda yapılarak metod hızlandırılmıştır.

### 3.3.3. Faster R-CNN

Bütün geliřtirmeler, hızlandırmalara ve başarısına rağmen Fast R-CNN gerçek zamanlı uygulamalarda halen yetersiz kalmaktaydı. Yavaş işlem yapan bir robot'ta kullanılabilir hale gelmiş olan Fast R-CNN yüksek tepki süresi gerektiren işlemlerde halen yavaş kalmaktaydı. Ren ve ark. (2015) bunun algoritmada oluşan bir dar boğazdan dolayı gerçekleştiğini fark etmişlerdi. Bu bölge öneri metodu olduğunu fark eden arařtırmacılar seçici tarama işleminden kurtulmak için yeni bir ađ geliřtirmeyi düşünmüşlerdi. Bu yeni geliřtirilen ađda bölge öneri metodu yerine nesnenin yerinin tahminini CNN'in evriřimsel katmanından çıkarılmış özellikler kullanılarak yapıyordu. Bu sayede bölge öneri metodu sebebiyle oluşan yavaşlama engellenmiş oldu. Faster R-CNN adı verilen bu yeni derin ađda bütün katmanlar Fast R-CNN ile aynı iken sadece daha önceden özelliklerin çıkarıldığı evriřimsel katmandan örnekleme katmanına geçilirken bu sefer buna paralel olarak bu özellikler Şekil 3.10'da görüleceđi üzere bir bölge öneri ađından geçirilerek nesnelere bulunabileceđi yerler işaretlenmiştir.



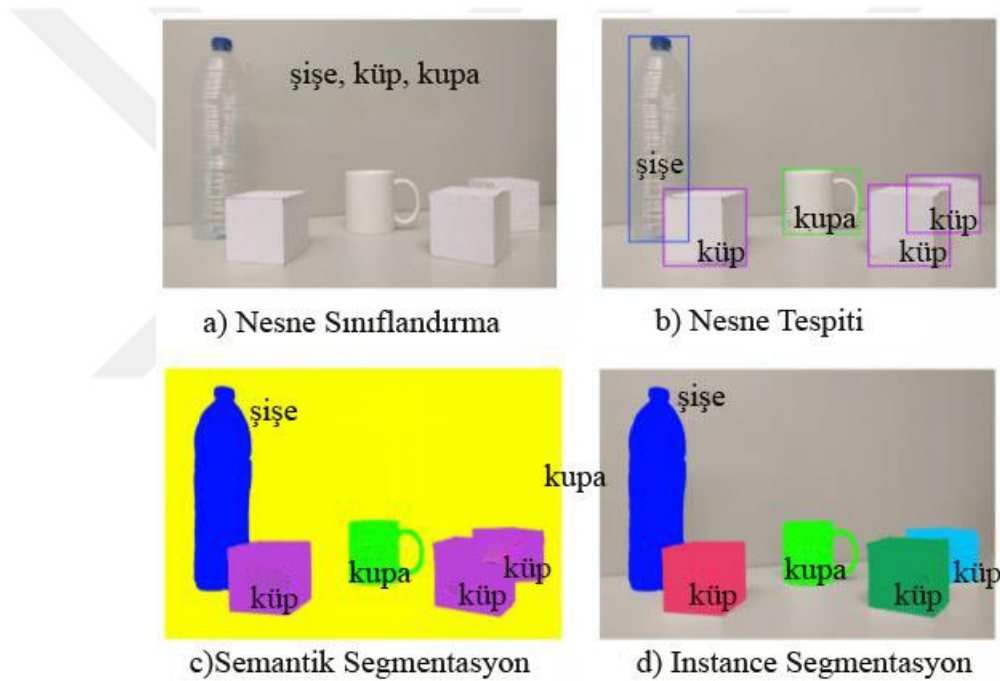
Şekil 3.10. Faster R-CNN'de çıkarılmış özelliklerin region proposal networkten geçirilmesi

Faster R-CNN sayesinde derin öğrenme ile nesne takibi uygulamaları büyük bir başarı elde etmiştir. Bulunduđu yılda gerçek zamanlı nesne takibi uygulamalarına en uygun metot olan

Faster R-CNN günümüzde de başarısını sürdürmektedir ancak performansı güncel metotlarla karşılaştırılmaya devam etmektedir.

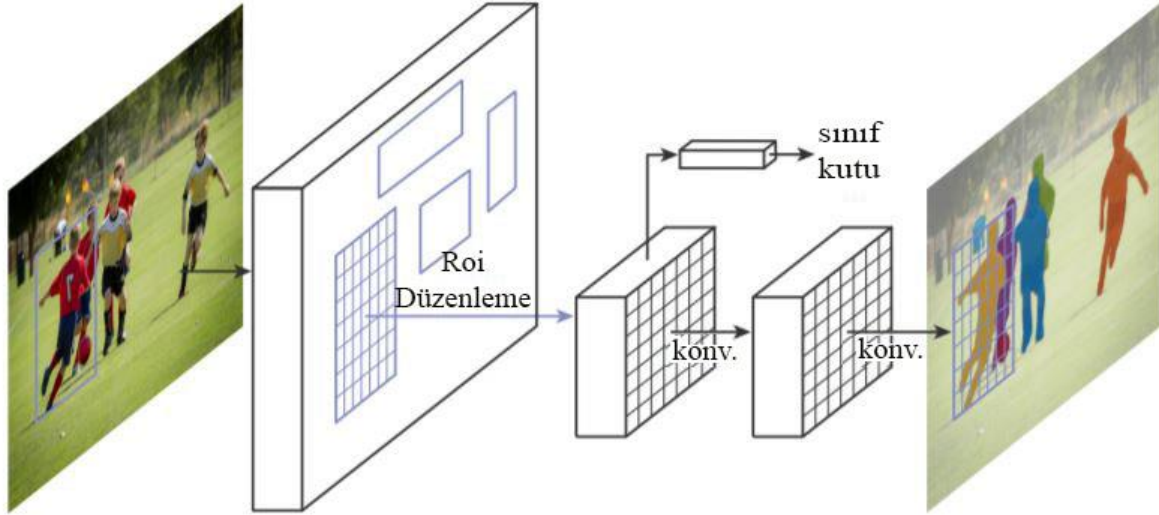
### 3.3.4. Mask R-CNN

R-CNN'lerin hikayesi üstünde bir etiket olan kutucuktan piksel seviyesinde sınıflandırmaya kadar ilerlemiştir. R-CNN üzerine yapılan He ve ark. (2017) nesnelerin bulunduğu bölgeler yerine her bir nesnenin piksellerinin tek tek incelenmesini daha başarılı olacağını ileri sürmüşlerdir. Böylece R-CNN'ler ile hem başarı artmış hem de nesnelerin anlamsal (semantik) ve örnek (instance) segmentasyonu yapılabilir hale gelmiştir. Segmentasyonun ne demek olduğunu ve türlerini daha kolay anlamak için Şekil 3.11'deki görsel örnek gösterilebilir.



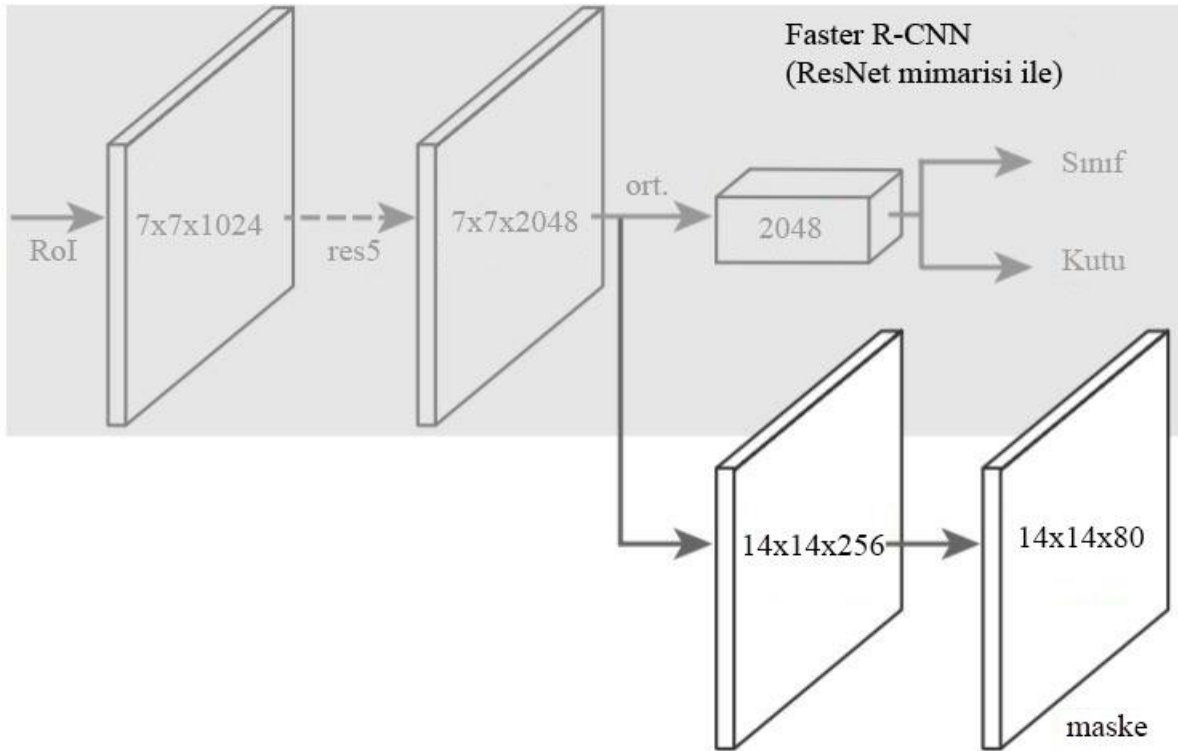
Şekil 3.11. Görüntü sınıflandırma nesne tanıma ve segmentasyona örnek (Garcia-Garcia ve ark., 2017)

Mask R-CNN adını verdikleri bu metotta diğer R-CNN'lerde bulunan nesne bölgelerinin bulunduğu kutucuklar gibi çıktı yerine 1 ve 0'lardan oluşan bir matris vermektedir (bu matrise binary mask da denmektedir). Bu matris nesnelerin bulunduğu yerleri belirtmektedir. Sonuç olarak Mask R-CNN mimarisi Faster R-CNN ile hemen hemen aynı olmasına çıkış katmanında nesne bölgeleri yerine piksel seviyesinde sınıflandırma yaparak hem başarıyı arttırmış hem de cisimlerin segmentasyonunun yapılmasına olanak sağlamıştır. Şekil 3.12'de Mask R-CNN mimarisini basitçe özetlenmiştir.



Şekil 3.12. Mask R-CNN mimarisi (He ve ark., 2017)

Mask R-CNN mimarisini Faster R-CNN'den ayıran özellikleri Şekil 3.13'deki resimde görülebilir.



Şekil 3.13. Faster R-CNN mimarisine piksel seviyesinde sınıflandırmanın eklenmesi (He ve ark., 2017)

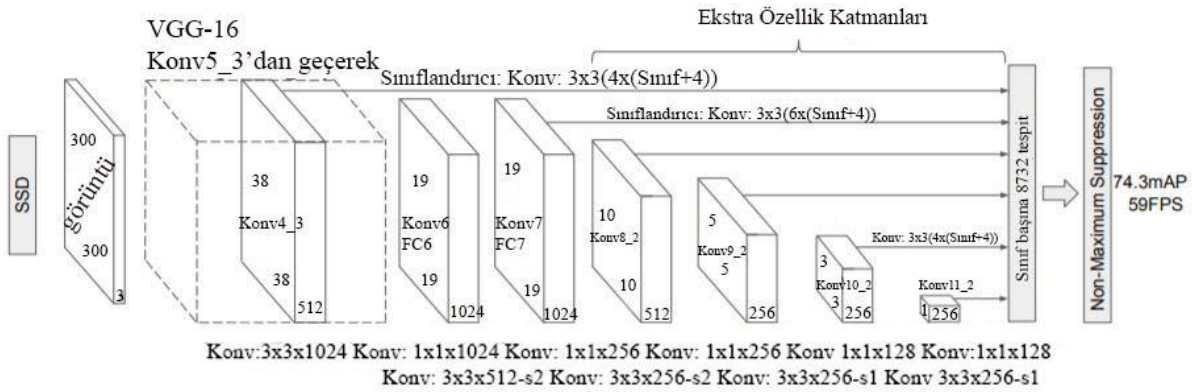
Mask R-CNN'in işlem yükü bakımından çok ağır olacağından dolayı bu tezde İHA uygulamalarında yer verilmemiştir.

### 3.3.5. Single Shot Multibox Detector (SSD)

Daha önce bahsedildiği üzere R-CNN'ler oldukça başarılı fakat hız olarak gerçek zamanlı uygulamalar için yetersiz kalmaktaydı. Liu ve ark. (2016) Single Shot Multibox Detector (SSD) adını vermiş oldukları yeni metod ile gerçek zamanlı nesne tanıma uygulamaları için bir derin öğrenme metodu gerçekleştirmişlerdir. Faster R-CNN, sınır kutuları oluşturmak için bir bölge öneri ağı kullanmaktadır ve bu kutuları nesnelere sınıflandırmak için kullanır. Faster R-CNN nesne tanıma bir teknoloji harikası olarak kabul edilirken, tüm işlem saniyede 7 kare hızında gerçekleşir. Bu FPS oranı gerçek zamanlı bir işlemin ihtiyaç duyduğu şeyin çok altında kalmaktadır. SSD, bölge öneri ağının ihtiyacını ortadan kaldırarak süreci hızlandırmaktadır. Başarı oranındaki düşüşü azaltmak için SSD çok ölçekli özellikler ve çerçeveler (bounding box) dahil olmak üzere birkaç iyileştirme uygular. Bu iyileştirmeler, SSD'nin daha düşük çözünürlüklü görüntüler kullanarak Faster R-CNN'in doğruluğuna yetişmesini sağlar ve bu da hızı daha da yükseltir. SSD ile nesne tanıma işlemi iki adımda gerçekleştirilir:

- 1) Özellik haritalarının çıkarılması,
- 2) Objelerin bulunması için evrimsel filtrenin uygulanması.

Aşağıdaki şekilde SSD'nin mimarisi Şekil 3.14'de görülmektedir.

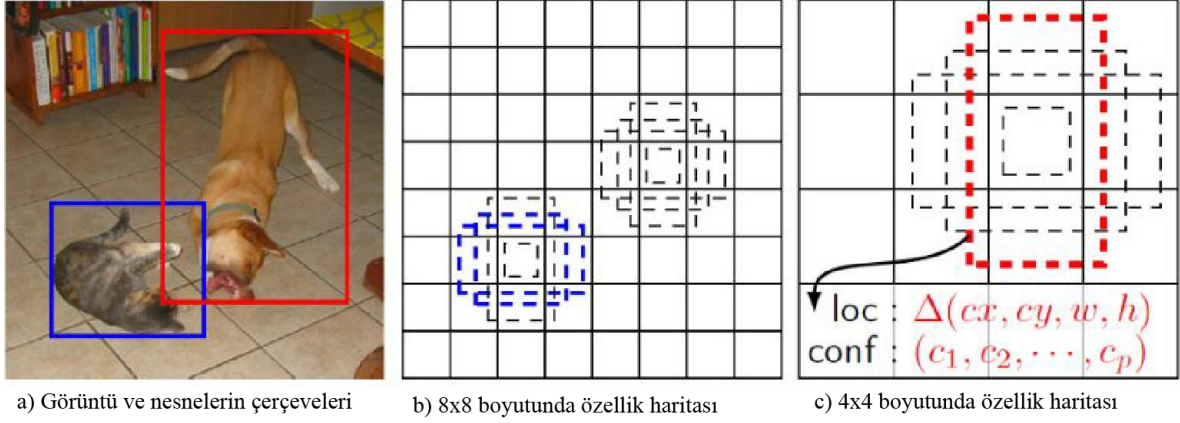


Şekil 3.14. Single Shot Multibox Detector (SSD) mimarisi (Liu ve ark., 2016)

SSD, R-CNN'lerde olduğu gibi parça parça değil de tüm resmi tek bir seferde CNN'den geçirerek nesne tanıma işlemini gerçekleştirir.

**Multibox Dedektörü:** Özellik çıkarımı için gerekli adımlar tamamlandıktan sonra "m x n x p" boyutunda bir özellik katmanı elde edilir. Bu katmanda "m x n" boyutu (nesne bölgelerinin sayısı) "p" ise kanal sayısını ifade eder. Örneğin Şekil 3.15'de 8x8 ve 4x4 boyutlarında iki tane özellik katmanı çıkarılmıştır ve 3x3 boyutunda konvolüsyon uygulanmıştır.





**Şekil 3.15.** Nesnelerin çerçeveleri için pozisyon (loc) ve doğruluk (conf) değerleri (Liu ve ark., 2016)

Her lokasyon için  $k$  sayıda çerçeve bulunmakta. Bu  $k$  sınırlayıcı kutular farklı boyutlara ve en boy oranlarına sahiptir. Çeşitli nesnelere göre uygunluk gösteren çerçevelerde, belki dikey bir dikdörtgen insan için daha uygun olacakken yatay bir dikdörtgen araba ya da bir hayvan için daha uygun olacaktır. Bu çerçevelerin doğruluğu iki kayıp oranı ile doğrulanır. Bunlardan ilki ‘güven kaybı (confidence loss)’ ikincisi ise ‘konum kaybı (location loss)’ oranlarıdır. Güven kaybı, ağın hesaplanan sınırlama kutusunun içinde nesne olup olmadığına ne kadar emin olduğunu ölçer. Kategorik çapraz-entropi bu kaybı hesaplamak için kullanılır. Konum kaybı, ağın öngörülen sınırlayıcı kutularının eğitim setindeki temel gerçeklerden ne kadar uzakta olduğunu ölçer. Burada L2-Norm kullanılır.

Özetlenecek olursa insansız hava araçlarına yönelik akıllı kontrol sistemlerinde kullanılabilecek en uygun nesne takibi metodunun bulunması amacıyla bu bölümde istatistiksel temelli nesne takip metotları olan KF ve PF anlatılmıştır. Yine aynı amaca yönelik derin öğrenme metotlarının tarihçesine ve gelişim sürecine yer verilmiş ve bir sonraki bölümde uygulaması anlatılacak olan Faster R-CNN ve SSD’nin detaylarına değinilmiştir. Bir sonraki bölümde anlatılan materyal ve yöntem ışığında yapılan uygulamalara yer verilmiştir.

#### 4. UYGULAMA VE SONUÇLAR

Bu bölümde tez aşamasında gerçekleştirilen uygulamalardan bahsedilecektir. Uygulama için 4 farklı nesne takip algoritması test edilmiştir. Kullanılan metotlardan 2 tanesi istatistiksel temelli, 2 tanesi ise derin öğrenme algoritmasıdır. Uygulamanın amacı İHA'lar için tasarlanan akıllı kontrol sistemlerine en uygun ve hızlı nesne takip algoritmasının seçilmesidir. Uygulama için test edilen nesne takip yöntemleri şunlardır:

- Parçacık Filtresi (PF),
- Kalman Filtresi (KF),
- Faster R-CNN,
- Single Shot Multibox Detector (SSD).

Uygulamalar sırasında 4K çözünürlükte çekilmiş İHA görüntüleri kullanılmıştır. Çekilen görüntüler nesne takip algoritmalarını kompleks koşullarda test etmek için arka planda mümkün olduğunca düz zeminden kaçınılmış, nesnenin yön değişimleri, yavaş ve hızlı hareketleri durumunda nesne takip yöntemlerinin tepkisi ölçülmüştür. İki farklı koşulda görüntü alınmıştır:

1. İHA sabit bir biçimde görüntü almakta ve nesne yay çizer bir şekilde bir rota takip ederek görüntü sonunda yönünü değiştirmektedir.
2. İHA hareketli bir biçimde önce yavaş bir şekilde hareketine başlayıp sonradan hızlanan bir cisim takip etmektedir. Bu hareket sırasında cisim yine yay çizerken İHA cismin sağından, solundan veya arkasından görüntü almaktadır.

Çekilen görüntüler takip algoritmalarının nesnenin boyutundaki gerçekleşen değişiklikleri de test etmek amacıyla 5 metre , 10 metre ve 20 metre yüksekliklerden yapılmıştır. Bu koşullar altında nesne takip algoritmalarının cismin şu değişimlerine rağmen başarısı ölçülmüştür:

- Nesnenin hafif dönüşlü bir yolda ilerlemesi,
- Nesnenin yönünün değişimi,
- Nesnenin hızının değişimi,
- Nesnenin boyutunun değişimi,
- Nesnenin görüntü kalitesinin azalması.

Nesne takibinde kullanılan videoların örnek görüntüleri Şekil 4.1 ve Şekil 4.2'de verilmiştir.



Şekil 4.1. İHA sabit bir konumda iken çekilmiş görüntüler



Şekil 4.2. İHA hareketli iken çekilmiş görüntüler

Bu koşullarda nesne takip algoritmalarının sonuçlarında şu kriterler kullanılmıştır:

1. Takip başarı oranı,
2. Saniye başına kare sayıları (FPS),
3. Grafik İşlemci Ünitesi (GPU) kullanım yüzdeleri,
4. Merkezi İşlemci Ünitesi (CPU) kullanım yüzdeleri.

Uygulama sırasında kullanılan bilgisayarın donanım bilgileri Çizelge 4.1’de verilmiştir.

Çizelge 4.1. Uygulamada Kullanılan Donanım Özellikleri

Donanım	Özellik
İşlemci (CPU)	Intel® Core™ i7-8750H 4.10GHz
Ekran Kartı (GPU)	NVidia RTX 2060 GDDR6 6GB
Hafıza (RAM)	DDR IV 16GB 2666MHz
Depolama	NVMe PCIe SSD
Chipset	Intel HM370
İşletim Sistemi	Windows 10 Professional

Takip başarı oranı hesaplanırken her bir karede cismin başarılı bir şekilde takip edilip edilmediği gözlenmiş, toplam 10 saniyelik videolarda her bir karedeki başarı hesaplanarak bir başarı yüzdesi hesaplanmıştır. FPS sayıları için işlemcinin her bir saniyede işleyebildiği kare sayısı hesaplanmıştır.

Tez aşamasındaki yapılan uygulamanın amacı insansız hava araçları için gerçekleştirilen akıllı kontrol sistemlerine yönelik en hızlı ve doğru sonucu veren gerçek zamanlı bir nesne takibi yönteminin bulunması ve yapılabilecek geliştirmelerin önerilmesidir. Yapılan uygulamaların teorik bilgileri 3. bölümde anlatılmıştır.

#### 4.1. Parçacık Filtresi

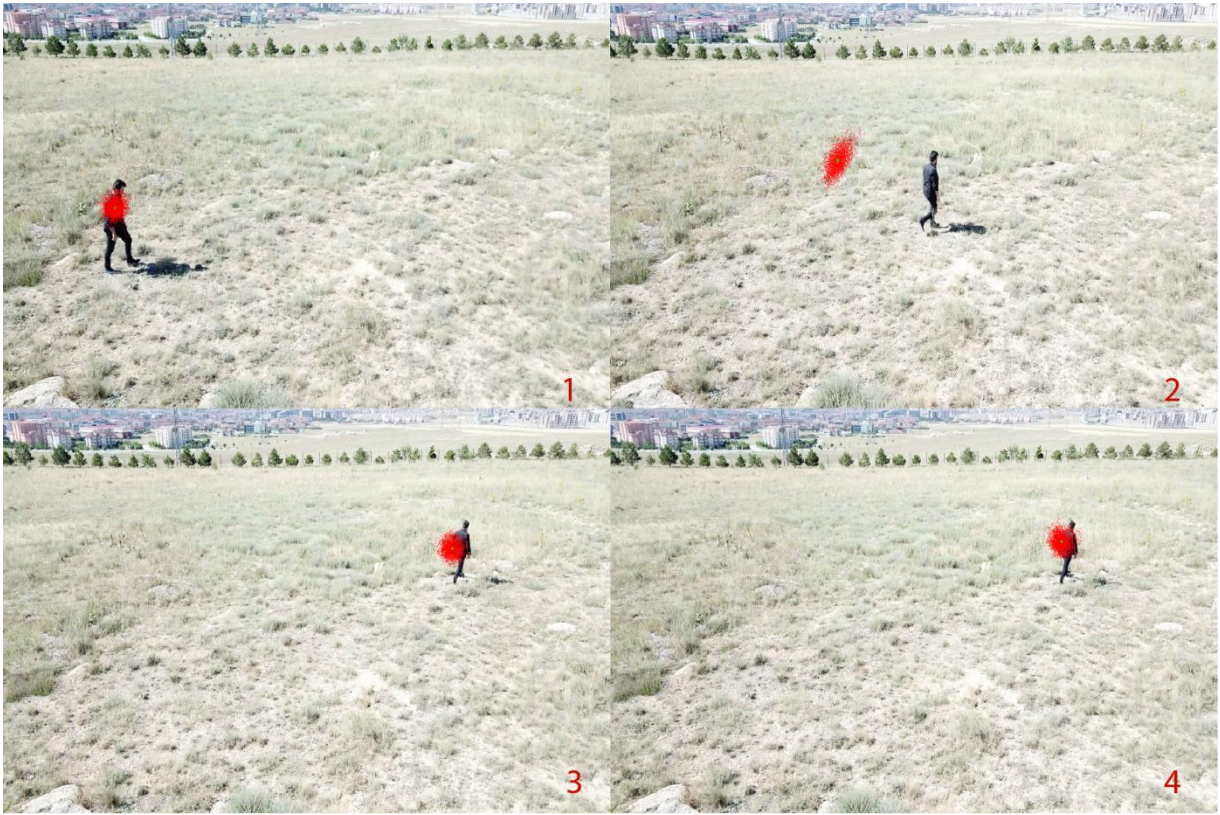
Bayes temelli istatistiksel bir nesne takip metodu olan PF sık kullanılan nesne takip metodlarından olup diğer bir Bayes temelli algoritma olan KF ile sıklıkla karşılaştırılmaktadır. Bu iki metotta da arka planda nesne takibini etkileyecek şekilde karışık renkler olması, ortam ışığının sürekli değişmesi, nesnenin boyutu (kameranın uzaklığı), ve özellikle gerçek zamanlı uygulamalarda hızlı yer değiştiren nesnelere takip konusunda zorluklar çıkartmaktadır. Çekilen 4K çözünürlüklü görüntülerde öncelikle görüntünün boyutu 1920x1080 boyutuna küçültülmüştür aksi takdirde görüntünün işlenmesi oldukça fazla zaman alacaktır. Görüntülerin daha fazla küçültülememesinin sebebi ise 20 metre yükseklikte gerçekleştirilen uçuşlarda cismin çok küçük kalması ve özellik çıkarımı yapılamayacak kadar az özellik barındırmasıdır. Sonraki adımda gürültü ve ışık değişiminin etkisini azaltmak için görüntünün HSV uzayına dönüştürülmesidir. Daha sonra görüntü histogram eşitleme metoduyla netleştirilerek önceden nesneden elde ettiğimiz Şekil 4.3'deki gibi bir şablondan faydalanarak görüntüde maskeleye yapılmış ve görüntü binary (siyah-beyaz) hale getirilmiştir.



Şekil 4.3. Maskeleye yapmak için çıkarılmış şablon görüntü

Son olarak Şekil 3.6'da görüleceği üzere parçacıkların tahmin işlemi başlamıştır. Parçacık sayısının yüksek olması takip doğruluğunu arttıracaktır ama bu durum, işlem yükünü oldukça arttıracaktır. Denklem 3.16'da görüldüğü üzere parçacık sayısının artması matematiksel işlem

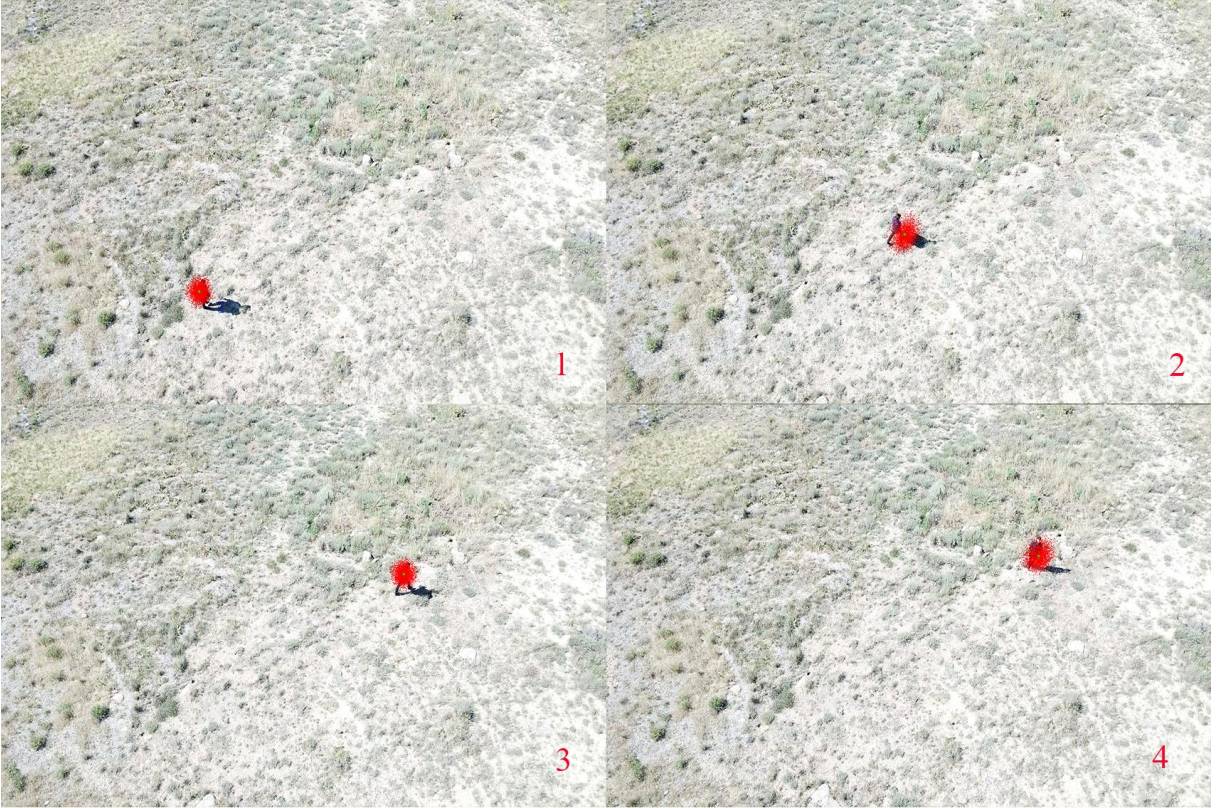
yükünü arttırmaktadır. Bu da algoritmanın yavaş çalışmasına neden olmaktadır. Takip doğruluğu ve hız bakımından çeşitli denemeler yapılarak bu uygulamada en optimum parçacık sayısının 3000 adet olduğu sonucuna varılmıştır. 3000 parçacık seçilmesi durumunda hem algoritma matematiksel olarak ağırlaşmamakta hem de yeterli nene takip doğruluğuna ulaşmaktadır. Parçacık sayısının kısmen çok olmasının sebebi alınan görüntülerin hareketli olması ve arka planın karmaşık olmasıdır. Takip sonucu 5 metre ve 20 metre yükseklikten yapılan sabit uçuşta ve 10 metre yükseklikten yapılan hareketli uçuşta Şekil 4.4, Şekil 4.5, Şekil 4.6'de gösterilmiştir. Nesnenin üzerindeki kırmızı noktalar parçacıkları, yeşil nokta ise cismin tahmin edilen yerini göstermektedir.



Şekil 4.4. Parçacık Filtresi ile nesnenin takibi (5 metre yüksekten sabit uçuş sırasında)



Şekil 4.5. Parçacık Filtresi ile nesnenin takibi (10 metre yüksekten hareketli uçuş sırasında)



Şekil 4.6. Parçacık Filtresi ile nesnenin takibi (20 metre yüksekten hareketli uçuş sırasında)

Resimlerde bazı karelerde takip sırasında hataların olduđu grlmektedir. Tek bir kareyi 5 metre hareketli ve sabit uuřlar sırasında daha yakından inceleyecek olursak Őekil 4.7, Őekil 4.8'de grlecektir.



Őekil 4.7. PF ile 5 metre ykseklikten hareketli uuř sırasında nesne takibi



Őekil 4.7. PF ile 5 metre ykseklikten sabit uuř sırasında nesne takibi

Yukarıda görüldüğü gibi nesne takibi başarılı bir şekilde gerçekleştirilmiştir. Daha önce de bahsedildiği üzere 5, 10 ve 20 metrelik sabit ve hareketli uçuşlar sırasında PF ile nesne takibine yönelik performans sonuçları Çizelge 4.2’de verilmiştir.

**Çizelge 4.2.** Parçacık Filtresi ile nesne takibi sonuçları

Parçacık Filtresi	FPS	Doğruluk %	GPU %	CPU %
<b>5m sabit</b>	6,6	84,21	0	70
<b>10m sabit</b>	9,19	91,67	0	70
<b>20m sabit</b>	6,6	78,85	0	71
<b>5m hareketli</b>	8	89,19	0	63
<b>10m hareketli</b>	8,04	89,86	0	69
<b>20m hareketli</b>	7,1	90	0	70

Çekilen görüntülerin 4K olması nedeni ile algoritmanın FPS oranı nispeten düşük gözlenmiştir. Ancak alınan görüntülerin çözünürlüğünün daha düşük olması durumunda FPS oranı gerçek zamanlı uygulama için uygun olacaktır. Ayrıca başarı oranına rağmen özellikle hareketli uygulamada, arka planın karışık olduğu uygulamalarda ve yüksekte uçuş sırasında nesne boyutunun küçük olması sebebiyle özelliklerinin az olduğu durumlarda takip sırasında Şekil 4.4’de de görüleceği üzere büyük sapmalar meydana geldiği gözlemlenmiştir. Bu sonuçlar bütün metotların sonuçları ile karşılaştırılarak 5. bölümde tekrar değerlendirilecektir.

#### 4.2. Kalman Filtresi

Diğer bir Bayes temelli metot olan KF ile nesne takibi için yapılan ön işlemler oldukça benzerlik göstermektedir. Sonuçların daha verimli karşılaştırılabilmesi için benzer işlemlerin uygulanmasına özen gösterilmiştir. Çekilen 4K görüntüler KF’nin hızının 4K çözünürlükte bir görüntüyü işleyebilecek yeterlilikte olması ve başarı oranını arttırmak için görüntüdeki özelliklerin kaybedilmemesi adına çözünürlüğü düşürülmemiştir. Öncelikle görüntü ışık değişimlerine duyarsız hale getirilmek için HSV uzayına çevrilmiştir. Daha sonra PF’de olduğu gibi görüntü histogram eşitleme metoduyla netleştirilip önceden nesneden elde ettiğimiz bir şablondan faydalanarak görüntüde maskeleye yapılmış ve görüntü binary hale getirilmiştir (Şekil 4.9).





**Şekil 4.9.** Maskeleye yapmak için çıkarılmış şablon görüntü

Son olarak da Şekil 3.3’de verilen Kalman döngüsü başlamıştır (Denklemler (3.3)-(3.15)). Alınan sonuçlar Şekil 4.10. ve Şekil 4.11’da gösterilmiştir.



**Şekil 4.10.** Kalman Filtresi ile 5 metre yükseklikte sabit uçuş sırasında nesne tespiti

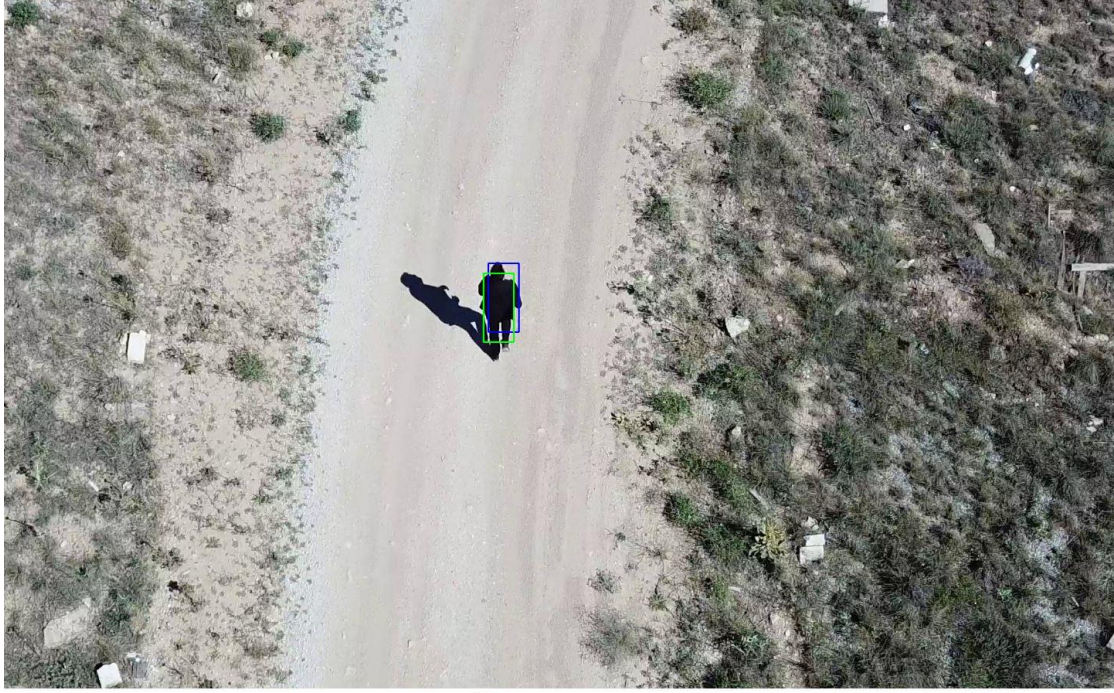


**Şekil 4.11.** Kalman Filtresi ile 10 metre yükseklikte hareketli uçuş sırasında nesne tespiti

Alınan sonuçlar oldukça başarılı olmuştur. Şekil 4.10'da doğruluk oranı oldukça yüksek bir sonuç çıkarken, Şekil 4.11'da çekim yapılan kamera hareketli ve cisim hızlı olmasına rağmen kabul edilebilir bir sonuç elde edilmiştir. Yapılan uçuş sırasında Şekil 4.12 ve Şekil 4.13'de görülen yeşil kare nesnenin bulunduğu yeri gösterirken mavi kare bir sonraki adımın tahminini göstermektedir.



**Şekil 4.12.** KF ile 20 metre yükseklikten sabit uçuş sırasında nesne takibi (yakınlaştırılmış)



**Şekil 4.13.** KF ile 20 metre yükseklikten hareketli uçuş sırasında nesne takibi

Genel olarak KF nesne takibinde başarılı olmasına karşın, hareketli uçuş sırasında Şekil 4.11’de görüleceği üzere yüksek irtifada ve hareketli uçuşlar sırasında bazı sapmalar meydana gelmektedir. Uçuşların performans sonuçları Çizelge 4.3’de gösterilmiştir.

**Çizelge 4.3.** Kalman Filtresi ile yapılan uçuşlarda nesne takibi performansı

Kalman Filtresi	FPS	Doğruluk %	GPU %	CPU %
<b>5m sabit</b>	12,6	99	0	37
<b>10m sabit</b>	12,85	97	0	38
<b>20m sabit</b>	12,87	99	0	36
<b>5m hareketli</b>	12,79	66,67	0	37
<b>10m hareketli</b>	12,74	84,09	0	36
<b>20m hareketli</b>	12,73	92,31	0	36

PF’de olduğu gibi çekilen görüntülerin 4K çözünürlükte olması nedeni ile algoritmanın FPS oranı düşük gözlenmiştir. Ancak yine aynı görüntülerin çözünürlüğünün daha düşük olması

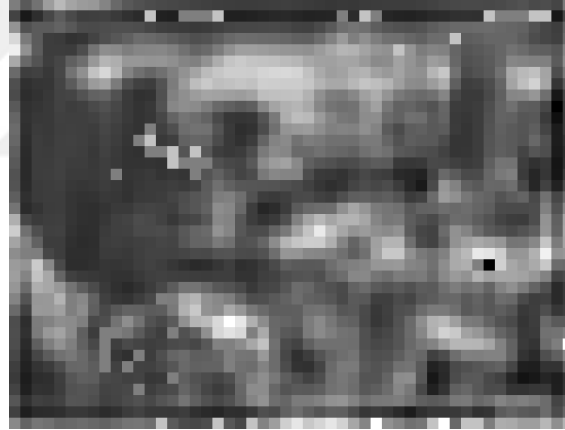
durumunda FPS oranı gerçek zamanlı uygulama için daha da uygun olacaktır. Bu sonuçlar bütün metotların sonuçları ile karşılaştırılarak 5. bölümde tekrar değerlendirilecektir.

### 4.3. Faster R-CNN

Nesne takip uygulamalarında son zamanlarda en popüler yöntemlerden biri derin öğrenmedir. Uygulamanın bu kısmında derin öğrenme metotlarından biri olan Faster R-CNN kullanılmıştır. Faster R-CNN ile nesne takibi yapılırken görüntü şu 3 adımı takip etmektedir:

- 1) Özellik çıkarım ağı,
- 2) Bölge öneri (Region Proposal) ağı,
- 3) Nesne tespit ağı.

Özellik çıkarım ağı genellikle önceden eğitilmiş iyi bilinen bir resim sınıflandırma ağıdır. Bu ağın amacı isminden de anlaşılacağı üzere görüntüden işe yarar özelliklerin çıkarılmasını sağlar. Bu ağın çıktısı, orijinal görüntünün şeklini ve yapısını korur. Şekil 4.14'de bir görüntünün griye çevrilerek çıkarılmış özellikleri görülmektedir.



Şekil 4.14. Bir görüntünün çıkarılmış özelliklerine bir örnek

Bölge öneri ağı aslında 3 evrimsel katmandan oluşan basit bir ağıdır. Bu katmanlardan bir tanesi diğer iki tanesini beslemektedir bu diğer iki katmanın bir tanesi sınıflandırma diğeri de sınır kutucuklarının (bounding box) regresyonu işlevlerini görürler. Bu ağın amacı bu nesne bölgelerinde aslında nesne olup olmadığını tespit etmektir.

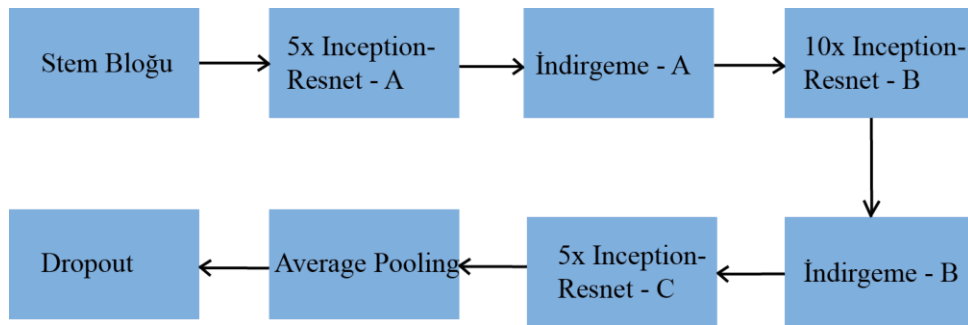
Nesne tespit ağının (diğer bir deyişle RCNN) ağı ise bu iki ağdan gelen sonuçları alarak nesne bölgelerinin yerini kesinleştirmek ve nesnelerin tespitini yapmaktır. Nesne tespit ağı ve bölge öneri ağının her ikisinin de eğitilmesi gerekmektedir. Bu çalışmada Faster R-CNN (Ren ve ark., 2015) yöntemi ile Başlangıç (Inception) Res-Net v2 (TensorFlow, 2016) mimarisi kullanılmıştır. Veri seti olarak Microsoft COCO (Common Object in Context) (2014) veri

setinin insan sınıfı kullanılarak hazır eğitim yapılmış ağ kullanılmıştır. Şekil 4.15’de MS COCO veri setinden bir örnek gösterilmiştir.



Şekil 4.15. MS COCO veri setinin sınıflarından örnekler (Lin ve ark., 2014)

Başlangıç (Inception) ResNet v2 ağ mimarisi kullanılarak 5,10 ve 20 metre yükseklikte hareketli ve sabit uçuş sırasında çekilmiş 4K çözünürlüklü görüntülerde gerçek zamanlı nesne tanıma yapılmıştır. Şekil 4.16’da Başlangıç (Inception) ResNet v2 mimarisinin blok diyagramı görülmektedir.



Şekil 4.16. Başlangıç (Inception) ResNet v2 mimarisinin blok diyagramı

Yapılan uygulamaların sonucunda Faster R-CNN’in oldukça başarılı olduğu görülmüştür. Şekil 4.17’de 5 metre yükseklikten sabit uçuş sırasında yapılmış çekimde nesne tanıma uygulaması görülmektedir. Şekil 4.18’de ise 5 metre yükseklikten hareketli uçuş sırasında yapılmış çekimde nesne tanıma uygulaması görülmektedir.



Şekil 4.17. Faster R-CNN ile 5 metre yükseklikte sabit uçuş sırasında nesne tespiti



Şekil 4.18. Faster R-CNN ile 5 metre yükseklikte hareketli uçuş sırasında nesne tespiti

Daha yakından incelenecek olursa Şekil 4.19’de 20 metrede sabit uçuşta çekilmiş görüntünün büyütülmüş hali gösterilmektedir.



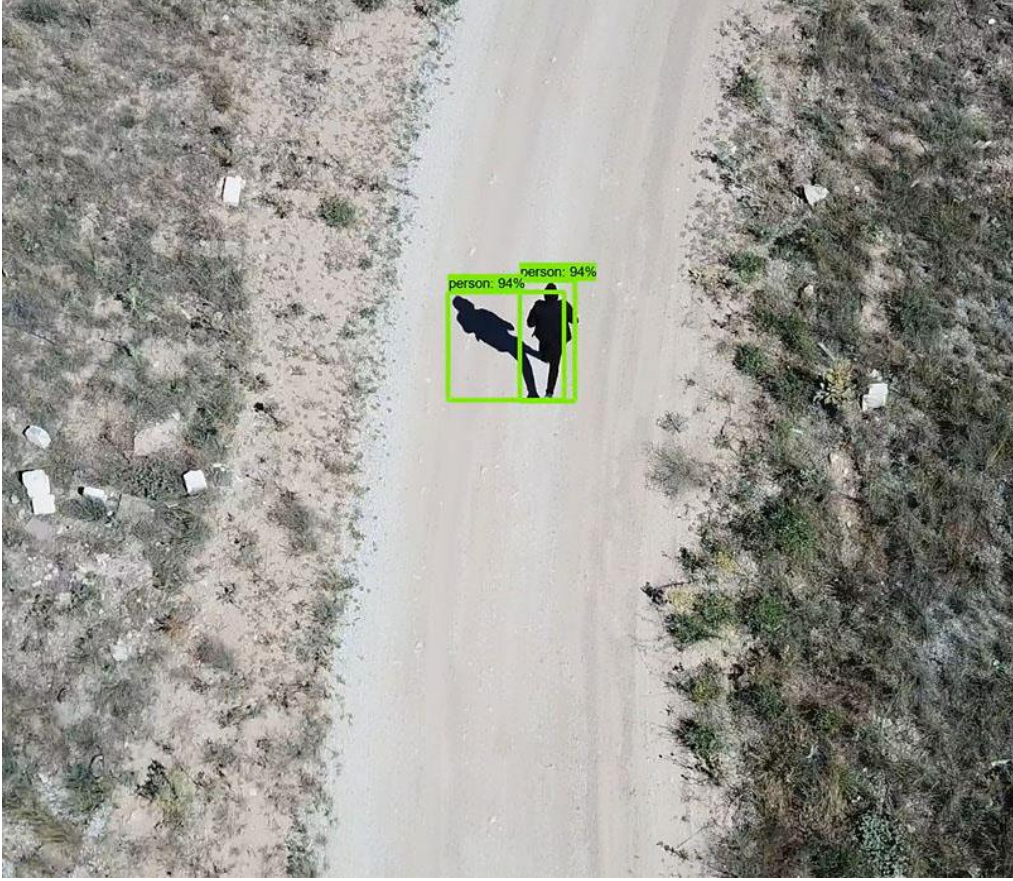
Şekil 4.19. 20 metre yükseklikte uçuştan bir görüntü

Uygulamada yapılan uçuşlar sonucunda nesne takibinin başarısını gösteren performans tablosu Çizelge 4.4’de verilmiştir.

Çizelge 4.4. Faster R-CNN yapılan uçuşlarda nesne takibi performansı

Faster R-CNN	FPS	Doğruluk %	GPU %	CPU %
5m sabit	1,2	100	78	25
10m sabit	1,16	100	78	28
20m sabit	1,16	99,5	75	28
5m hareketli	1,15	100	73	27
10m hareketli	1,16	100	76	26
20m hareketli	1,17	81,25	79	26

Çizelgede görülen sonuçlara bakıldığında Faster R-CNN nesne takibinde oldukça başarılı olmasına rağmen nadir de olsa bazen nesneyi bulamamakta (doğruluk oranı %80'e kadar düşebilmekte) veya nesnenin gölgesiyle kendisini karıştırabilmektedir (Şekil 4.20).



**Şekil 4.20.** Faster R-CNN'in nesne tespiti yaparken nesnenin gölgesini nesne olarak etiketlemesi

Ayrıca yüksekte doğrudan cisme bakıldığında nesne takibinin anlık olarak başarısız olduğu gözlemlenmiştir. Bunun sebebi olarak derin öğrenmede kullanılan verilerin nesnenin tam üstündeyken daha fazla çekilip veri setine eklenerek tekrardan eğitim yapılması gerekmektedir. Buna rağmen FPS oranına ve işlem gücü gerekliliğine bakıldığında Faster R-CNN çok doğru sonuçlar vermesinin yanında oldukça yavaş sonuçlar vermektedir. Bu durum İHA'larda doğrudan kullanım ihtimalini düşürmektedir. Bu sonuçlar bütün metotların sonuçları ile karşılaştırılarak 5. bölümde tekrar değerlendirilecektir.

#### **4.4. Single Shot Multibox Detector (SSD)**

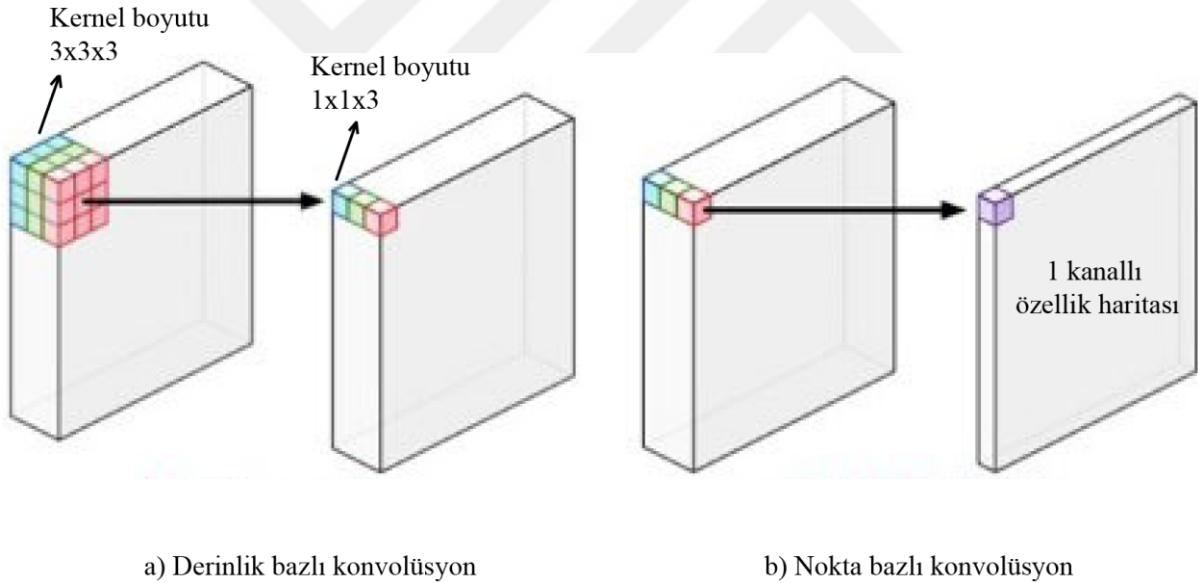
Nesne takip uygulamalarında son zamanlarda en popüler yöntemlerden biri derin öğrenmedir. Uygulamanın bu kısmında nesne takibi konusunda en popüler derin öğrenme metotlarından biri olan SSD kullanılmıştır. SSD ile nesne takibi 2 adımdan oluşmaktadır.



- 1) Nesne tespiti için çok ölçekli özellik haritalarının çıkarılması,
- 2) Nesne tespiti için evrişimsel tahmin yapılması.

Çok ölçekli özellik haritalarının mantığı, Faster R-CNN'deki özellik çıkarım ağına çok benzemektedir. Bu katman, görüntüden işe yarar özelliklerin çıkarılmasını sağlar. Bu ağın çıktısı, orijinal görüntünün şeklini ve yapısını korur. İkinci adımda evrişimsel sınıflandırıcıdan geçirilerek 'güven kaybı' ve 'konum kaybı' kayıp oranları ölçülerek nesnelerin sınırlayıcı çerçeveleri bulunur.

Bu uygulamada eğitim ve sınıflandırma Singleshot Multibox Detector (SSD) (2016) yöntemi ile bir evrişimsel ağ mimarisi olan MobileNet (2017) ağ mimarisi kullanılmıştır. MobileNet ağ mimarisi önce derinlik bazlı konvolüsyon yaparak üç kanallı resimler üzerinde 3 kanallı bir özellik haritası oluşturur daha sonra nokta bazlı konvolüsyon işlemi gerçekleştirerek bu özellik haritasının Şekil 4.21'de görülen tek kanallı özellik haritasına dönüştürülmesini sağlar.



**Şekil 4.21.** MobileNet ağ mimarisiyle görüntünün konvole edilmesiyle özellik haritasının çıkarılması (Lee, 2017)

Veri seti olarak Microsoft COCO (2014) veri setinin insan sınıfı kullanılarak hazır eğitim yapılmış ağ kullanılmıştır. 5,10 ve 20 metre yükseklikte hareketli ve sabit uçuş sırasında çekilmiş 4K çözünürlüklü görüntülerde gerçek zamanlı insan tanıma yapılmıştır. Yapılan uygulamaların sonucunda SSD'nin oldukça başarılı olduğu görülmüştür. Şekil 4.22'de 5 metre yükseklikten sabit uçuş sırasında yapılmış çekimde nesne tanıma uygulaması görülmektedir.

Şekil 4.23’de ise 5 metre yükseklikten hareketli uçuş sırasında yapılmış çekimde nesne tanıma uygulaması görülmektedir.



Şekil 4.22. SSD ile 5 metre yükseklikte sabit uçuş sırasında nesne tespiti



Şekil 4.23. SSD ile 5 metre yükseklikte hareketli uçuş sırasında nesne tespiti

Daha yakından incelenecek olursa Şekil 4.24'de 20 metrede sabit uçuşta çekilmiş görüntünün büyütülmüş hali gösterilmektedir. Şekil 4.25'te ise 10 metreden sabit uçuşta çekilmiş görüntünün büyütülmüş hali gösterilmektedir. Yapılan uçuşlar sırasında, SSD'nin küçük cisimlere karşı duyarlılığının az olduğu görülmüştür. Bu durum 20 metrelik uçuşlar sırasında nesne takibinde aksamalar sonucu gözlemlenmiş, doğruluk oranı genel olarak %90 üzerinde görülürken bazı uygulamalarda %66.7'e kadar düştüğü gözlemlenmiştir. Uygulamada yapılan uçuşlar sonucunda nesne takibinin başarısını gösteren performans tablosu Çizelge 4.5'de verilmiştir.



Şekil 4.24. SSD ile 10 metre yükseklikten sabit uçuş sırasında nesne takibi (yakınlaştırılmış)



Şekil 4.25. SSD ile 10 metre yükseklikten hareketli uçuş sırasında nesne takibi (yakınlaştırılmış)

Çizelge 4.5. SSD ile nesne takibi sonuçları

SSD	FPS	Doğruluk %	GPU %	CPU %
5m sabit	5,5	92,86	12	72
10m sabit	5,35	90,91	14	75
20m sabit	6,9	58,33	14	69
5m hareketli	5,44	96,67	13	75
10m hareketli	6,62	66,7	17	68
20m hareketli	7,38	46,88	20	64

Çizelge görülen sonuçlara bakıldığında Faster R-CNN nesne takibinde oldukça başarılı olmasına rağmen nadir de olsa bazen nesneyi bulamamaktadır. Ayrıca yüksekten doğrudan cisme bakıldığında nesne takibinin anlık olarak başarısız olduğu gözlemlenmiştir. Bunun sebebi olarak derin öğrenmede kullanılan verilerin yetersiz olmasıdır. Nesnenin tam üstündeyken daha fazla fotoğraf çekilip veri setine eklenerek tekrardan eğitim yapılması gerekmektedir. Ancak FPS oranına ve işlem gücü gerekliliğine bakıldığında SSD Faster R-CNN'den başarı olarak daha düşük olmasına rağmen hız konusunda daha üstün olduğu görülmektedir. Bu durum

İHA’larda doğrudan kullanım ihtimalini arttırmaktadır. Bu sonuçlar bütün metotların sonuçları ile karşılaştırılarak 5. bölümde tekrar değerlendirilecektir.



## 5. SONUÇ VE TARTIŞMA

Bu tezin amacı insansız hava araçlarında akıllı kontrol sistemlerine yönelik en uygun nesne takip algoritmasının bulunmasıdır. Bu amaca yönelik gerçekleştirilen uygulamalar sonucunda, insansız hava araçlarında akıllı kontrol sistemlerine yönelik nesne takip uygulamaları için 4 farklı metot kıyaslanmıştır. Bu amaç doğrultusunda 5, 10 ve 20 metre yükseklikte yapılan sabit ve hareketli olmak üzere toplamda 6 uçuştan veri toplanarak ikisi istatistiksel ikisi derin öğrenme metodu olan 4 farklı yöntem kıyaslanmıştır. Bu yöntemler Parçacık Filtresi, Kalman Filtresi, Faster R-CNN ve SSD'dir. Alınan verilere bu metotların uygulanması sonucunda her bir metodun performans tabloları çıkarılmış ve Çizelge 5.1'de gösterilmiştir.

**Çizelge 5.1.** Uygulanan 4 metodun (PF, KF, Faster R-CNN, SSD) performans tablosu

Parçacık Filtresi	FPS	Doğruluk %	GPU %	CPU %	Kalman Filtresi	FPS	Doğruluk %	GPU %	CPU %
<b>5m sabit</b>	6,6	84,21	0	70	<b>5m sabit</b>	12,6	99	0	37
<b>10m sabit</b>	9,19	91,67	0	70	<b>10m sabit</b>	12,85	97	0	38
<b>20m sabit</b>	6,6	78,85	0	71	<b>20m sabit</b>	12,87	99	0	36
<b>5m hareketli</b>	8	89,19	0	63	<b>5m hareketli</b>	12,79	66,67	0	37
<b>10m hareketli</b>	8,04	89,86	0	69	<b>10m hareketli</b>	12,74	84,09	0	36
<b>20m hareketli</b>	7,1	90	0	70	<b>20m hareketli</b>	12,73	92,31	0	36
Faster R-CNN	FPS	Doğruluk %	GPU %	CPU %	SSD	FPS	Doğruluk %	GPU %	CPU %
<b>5m sabit</b>	1,2	100	78	25	<b>5m sabit</b>	5,5	92,86	12	72
<b>10m sabit</b>	1,16	100	78	28	<b>10m sabit</b>	5,35	90,91	14	75
<b>20m sabit</b>	1,16	99,5	75	28	<b>20m sabit</b>	6,9	58,33	14	69
<b>5m hareketli</b>	1,15	100	73	27	<b>5m hareketli</b>	5,44	96,67	13	75
<b>10m hareketli</b>	1,16	100	76	26	<b>10m hareketli</b>	6,62	66,7	17	68
<b>20m hareketli</b>	1,17	81,25	79	26	<b>20m hareketli</b>	7,38	46,88	20	64

Çizelgelerde gösterilen sonuçlar ışığında her bir metot için aşağıdaki sonuçlara varılmıştır.

**Parçacık Filtresi:** Doğruluk oranı kıyaslamasında sonuncu sırada olmasına karşın kabul edilebilir bir doğruluk oranına sahiptir. Ayrıca FPS sayısına bakıldığı zaman ortalama 7.58 ile Kalman Filtresi'nden sonra ikinci sırada yer almaktadır. Fakat yüksekte çekilen görüntülerde cismin küçük olmasından dolayı yeterli özellik cisimden çıkarılamamaktadır. 20 metre yükseklikte çekilen görüntülerde kabul edilebilir bir başarı oranına sahip olmasına karşın daha yüksek bir irtifada alınan görüntüde başarı oranının düşme ihtimali yüksektir. Ayrıca hızlı hareket eden ve sık yön değiştiren nesnelerin takibinde başarı oranının düştüğü gözlemlenmiştir. Ek olarak, arka planın karmaşık olduğu durumlarda da olumsuz sonuçlar vermektedir. İşlem gücü bakımından GPU kullanmayan algoritma CPU'yu ortalama %70 oranında kullanırken bu değer parçacık sayısına bağlı olarak değişim göstermektedir. CPU kullanım miktarına bakıldığında optimize edilmiş düşük maliyetli bir donanıma sahip bir sistemde çalıştırılabilir. Bu nedenle PF basit İHA uygulamalarına uygun olarak görülmüştür. Fakat yüksek bir performans beklenmemesi gerekir. Son kullanıcı ürünü olması muhtemel görülmemiştir.

**Kalman Filtresi:** Doğruluk oranı kıyaslamasında derin öğrenme metotlarının arkasında kalmasına karşın FPS sayısına bakıldığında ortalama 12.7 ile en iyi performansı veren yöntemdir. Fakat Parçacık Filtresi'nde olduğu gibi yüksekte çekilen görüntülerde cismin küçük olmasından dolayı yeterli özellik cisimden çıkarılamamaktadır. Bu nedenle 20 metre yükseklikte çekilen görüntülerde kabul edilebilir bir başarı oranına sahip olmasına karşın daha yüksek bir irtifada alınan görüntüde başarı oranının düşme ihtimali yüksektir. Ayrıca hızlı hareket eden ve sık yön değiştiren nesnelerin takibinde de başarı oranının düştüğü gözlemlenmiştir. İstatistiksel temelli metotların genel sorunlarından biri olan arka planın gürültülü ve karmaşık olması durumunda KF Gauss gürültüsüne benzeyen gürültülere karşı iyi çalışmasına rağmen gerçek hayatta İHA'lardan alınmış görüntülerde sapma yaşayabilmektedir. Fakat her şeye rağmen işlem gücü bakımından GPU gereksinimi olmaması ve CPU kullanımında ortalama %36,6 oranı ile oldukça avantajlı olması sebebiyle düşük maliyeti fiyat-performans ilişkisi bakımından incelendiğinde KF'yi en üst sıraya taşımaktadır. Bu nedenle KF ekstrem olmayan ve çok yüksek hızlarda seyir etmeyen İHA uygulamalarına oldukça uygun görülmüştür.

**Faster R-CNN:** Doğruluk oranı kıyaslamasında diğer bütün yöntemlere üstünlük sağlayan hemen hemen her uygulamada %90 ve üzeri başarı gösteren Faster R-CNN'in en büyük

dezavantajı işlem yükü ve FPS sayısı olmuştur. Ortalama 1,1 ile oldukça düşük FPS oranlarına sahip Faster R-CNN donanım gereksinimi bakımından oldukça maliyetli olacaktır. Gerekli donanımın bir İHA üzerine monte edilebilmesi durumunda dahi sağlamış olduğu FPS oranı bir İHA'nın reaksiyon süresinden oldukça düşük kalmaktadır. Veri setinin yeniden düzenlenerek bir eğitim yapılması durumunda İHA'lar olmasa da Faster R-CNN sabit bir bilgisayar ortamında gözetleme amaçlı kullanıldığında oldukça başarılı olacaktır. Fakat İHA'larda kullanımı için uygun görülmemiştir.

**SSD:** Doğruluk oranı kıyaslamasında 4 metodun içinden ikinci sırada olan SSD işlem yükü (%15 GPU, %70,5 CPU) ve ve FPS (6,19) oranı göz önünde alındığında ise Faster R-CNN'den başarılı bir sonuç vermiştir. Uçuş sırasında nesne takibinde özellikle yüksek irtifalarda anlık olarak nesneyi kaçırmamasına rağmen nesnenin tespitini tekrardan yapabilmesi sayesinde, veri setinin düzenlenip daha fazla veri eklenerek tekrardan eğitilmesi durumunda oldukça başarılı bir yöntem olacaktır. Orta maliyetli bir donanım ile gerekli isterleri karşılayabilen SSD yapılan uygulamalar sonucunda İHA'larda kullanımı en uygun yöntem olduğu görülmüştür.

Gerçekleştirilen uygulamalar sonucu gözlemlere ve yapılan yorumlara dayanılarak insansız hava araçlarında akıllı kontrol sistemlerine yönelik nesne takibi uygulamalarına en uygun metodun SSD olduğu gözlemlenmiştir. Orta maliyetli bir donanım ile yüksek performans gösterebilmektedir. Fakat maliyetin, sistem ağırlığının ve işlem gücünün az olması gerektiği sistemlerde ise KF kullanımı uygun olacaktır.

**Geliştirme ve Gelecek Çalışmalar:** Unutulmamalıdır ki her metodun bir avantajı olduğu gibi dezavantajı da bulunmaktadır. Bu tezin amacına yönelik gelecekte yapılacak çalışmalarla bu tezde uygulaması yapılmış 4 metodun birlikte çalıştırıldığı hibrit bir sistem geliştirilmesi planlanmaktadır. Geliştirilmesi planlanan hibrit algoritmada ise şu adımlar izlenecektir:

- Derin öğrenme metotları ile nesne tespit edilmesi,
- Tespit edilen nesne istatistiksel temelli metoda girdi olarak verilmesi,
- Nesnenin hareketinin tahmini yapılması,
- Belirli aralıklarla nesne tespiti derin öğrenme metodu ile teyit edilerek adımlar tekrardan izlenmesi.

Bu yöntemde derin öğrenme metodu ile tespit edilmiş nesne istatistiksel temelli metoda girdi olarak verilerek nesne tespitinin başarı oranı arttırılması hedeflenmiştir. Bu sayede istatistiksel öğrenmede yapılan özellik çıkarımı ihtiyacı ortadan kaldırılacaktır. Bu durum istenilen irtifada ve karmaşık arka plan olması durumunda dahi nesne takibi yapılmasına olanak



sağlayacaktır. Daha sonra nesnenin takibinin derin öğrenme ile değil de istatistiksel metotlarla yapılarak algoritmanın işlem gücü azaltılarak hızının artırılması hedeflenmektedir. Belirli aralıklarla derin öğrenme ile nesnenin tespitinin teyit edilmesi ile birlikte, nesnenin takibinin kaybedilmesi önlenmesi ve bu sayede sürekli sınıflandırma işleminin yapılarak işlem gücünün azaltılması hedeflenmektedir.

Daha önce bahsedildiği üzere bu tezi çalışmasının amacı insansız hava araçlarında akıllı kontrol sistemlerine yönelik en uygun nesne takip algoritmasının bulunmasıdır. Bu amaca yönelik yapılmış olan bu tez uygulamasının en önemli özelliği gerçek bir İHA'dan alınan görüntüler kullanılarak daha önce yapılmamış bir karşılaştırmanın (KF, PF, Faster R-CNN, SSD) yapılmış olmasıdır. Yapılan uygulamadan edinilen sonuçlar ve önerilen geliştirmeler gelecekte İHA'lara veya farklı alanlara yönelik çalışmalara ilham olacak niteliktedir.

Bu tez uygulaması amacına yönelik oluşturulmuş veriler bu teze atıf yapmak koşuluyla kullanılmaya açık bir şekilde paylaşılmıştır. Verilere aşağıdaki bağlantıdan ulaşılabilir:

<https://drive.google.com/open?id=1Nx5tsNfqtPLnnTie45Mi6ElahnLRfVSc>

## KAYNAKLAR

- Arulampalam, M. S., Maskell, S., Gordon, N. ve Clapp, T. J. I. T. o. s. p., 2002, A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking, 50 (2), 174-188.
- 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.
- Black, J., Ellis, T., Rosin, P. J. P. o. t. I. I. o. V. S., Tracking, P. E. o. ve Surveillance, 2003, A novel method for video tracking performance evaluation, 125-132.
- Bouffard, P., Aswani, A. ve Tomlin, C., 2012, Learning-based model predictive control on a quadrotor: Onboard implementation and experimental results, *2012 IEEE International Conference on Robotics and Automation*, 279-284.
- 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.
- Comaniciu, D., Ramesh, V. ve Meer, P., 2000, Real-time tracking of non-rigid objects using mean shift, *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*, 142-149.
- Cuevas, E. V., Zaldivar, D. ve Rojas, R., 2005, Kalman filter for vision tracking.
- Dang, C.-T., Pham, H.-T., Pham, T.-B. ve Truong, N.-V., 2013, Vision based ground object tracking using AR. Drone quadrotor, *2013 International Conference on Control, Automation and Information Sciences (ICCAIS)*, 146-151.
- Del Moral, P. J. M. p. ve fields, r., 1996, Non-linear filtering: interacting particle resolution, 2 (4), 555-581.
- Del Moral, P. J. T. A. o. A. P., 1998, Measure-valued processes and interacting particle systems. Application to nonlinear filtering problems, 8 (2), 438-495.
- Garcia-Garcia, A., Orts-Escolano, S., Oprea, S., Villena-Martinez, V. ve Garcia-Rodriguez, J. J. a. p. a., 2017, A review on deep learning techniques applied to semantic segmentation.
- Girshick, R., Donahue, J., Darrell, T. ve Malik, J., 2014, Rich feature hierarchies for accurate object detection and semantic segmentation, *Proceedings of the IEEE conference on computer vision and pattern recognition*, 580-587.
- Girshick, R., 2015, Fast r-cnn, *Proceedings of the IEEE international conference on computer vision*, 1440-1448.
- He, K., Gkioxari, G., Dollár, P. ve Girshick, R., 2017, Mask r-cnn, *Proceedings of the IEEE international conference on computer vision*, 2961-2969.
- Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M. ve Adam, H. J. a. p. a., 2017, Mobilenets: Efficient convolutional neural networks for mobile vision applications.
- Hue, C., Le Cadre, J.-P., Pérez, P. J. I. t. o. a. ve systems, e., 2002, Tracking multiple objects with particle filtering, 38 (3), 791-812.
- Hung, C., Xu, Z. ve Sukkarieh, S. J. R. S., 2014, Feature learning based approach for weed classification using high resolution aerial images from a digital camera mounted on a UAV, 6 (12), 12037-12054.
- İMAMOĞLU, N. ve MÜHENDİSLİĞİ, E. V. E., 2010, DÖNERKANAT TİPİNDE BİR İNSANSIZ HAVA ARACIYLA VİDEO TABANLI ÜST DÜZEY İŞLEVLERİN TASARLANMASI.
- Kang, S., Paik, J.-K., Koschan, A., Abidi, B. R. ve Abidi, M. A., 2003, Real-time video tracking using PTZ cameras, *Sixth International Conference on Quality Control by Artificial Vision*, 103-112.
- Kaufmann, E., Loquercio, A., Ranftl, R., Dosovitskiy, A., Koltun, V. ve Scaramuzza, D. J. a. p. a., 2018, Deep drone racing: Learning agile flight in dynamic environments.

- Krizhevsky, A., Sutskever, I. ve Hinton, G. E., 2012, Imagenet classification with deep convolutional neural networks, *Advances in neural information processing systems*, 1097-1105.
- Lee, J., 2017, MobileNet - PR044, <https://www.slideshare.net/JinwonLee9/mobilenet-pr044>:
- 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.
- Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P. ve Zitnick, C. L., 2014, Microsoft coco: Common objects in context, *European conference on computer vision*, 740-755.
- Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y. ve Berg, A. C., 2016, Ssd: Single shot multibox detector, *European conference on computer vision*, 21-37.
- Maggio, E. ve Cavallaro, A., 2011, Video tracking: theory and practice, John Wiley & Sons, p.
- Merino, L., Caballero, F., Martínez-de Dios, J. ve Ollero, A., 2005, Cooperative fire detection using unmanned aerial vehicles, *Proceedings of the 2005 IEEE international conference on robotics and automation*, 1884-1889.
- Merino, L., Caballero, F., Martínez-de Dios, J. R., Ferruz, J. ve Ollero, A. J. J. o. F. R., 2006, A cooperative perception system for multiple UAVs: Application to automatic detection of forest fires, 23 (3-4), 165-184.
- Molloy, T. L., Ford, J. J. ve Mejias, L. J. J. o. F. R., 2017, Detection of aircraft below the horizon for vision-based detect and avoid in unmanned aircraft systems, 34 (7), 1378-1391.
- Moranduzzo, T. ve Melgani, F., 2012, A SIFT-SVM method for detecting cars in UAV images, *2012 IEEE International Geoscience and Remote Sensing Symposium*, 6868-6871.
- Möckel, R., 2012, Bio-inspired optical flow vision sensors for visual guidance of autonomous robots, *ETH Zurich*.
- Nummiaro, K., Koller-Meier, E., Van Gool, L. J. I. ve computing, v., 2003, An adaptive color-based particle filter, 21 (1), 99-110.
- 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.
- Ren, S., He, K., Girshick, R. ve Sun, J., 2015, Faster r-cnn: Towards real-time object detection with region proposal networks, *Advances in neural information processing systems*, 91-99.
- Roberts, R., Ta, D.-N., Straub, J., Ok, K. ve Dellaert, F., 2012, Saliency detection and model-based tracking: a two part vision system for small robot navigation in forested environment, *Unmanned Systems Technology XIV*, 83870S.
- Rosenblatt, F. J. P. r., 1958, The perceptron: a probabilistic model for information storage and organization in the brain, 65 (6), 386.
- Salmond, D., Gordon, N. J. S. s., theory, u. c. m. ve applications, 2005, An introduction to particle filters, 1-19.
- Schneiderman, H. J. C., 2004, Learning a restricted Bayesian network for object detection, 4, 639-646.
- SHT-İHA, 2016, Sivil Havacılık Genel Müdürlüğü İnsansız Hava Aracı Sistemleri Talimatı, Sivil Havacılık Genel Müdürlüğü, Ankara,
- Sun, W., Chen, L., Ren, L., Guo, B. ve Wu, X., 2012, Objects detecting and tracking with a new particle filter, *2012 2nd International Conference on Consumer Electronics, Communications and Networks (CECNet)*, 3340-3343.
- Tao, H., Sawhney, H. S., Kumar, R. J. I. t. o. p. a. ve intelligence, m., 2002, Object tracking with bayesian estimation of dynamic layer representations, 24 (1), 75-89.

- TensorFlow, 2016, Inception-ResNet-v2 Network, TensorFlow, the TensorFlow logo and any related marks are trademarks of Google Inc., [https://github.com/tensorflow/models/blob/master/research/slim/nets/inception\\_resnet\\_v2.py](https://github.com/tensorflow/models/blob/master/research/slim/nets/inception_resnet_v2.py):
- van Gemert, J. C., Verschoor, C. R., Mettes, P., Epema, K., Koh, L. P. ve Wich, S., 2014, Nature conservation drones for automatic localization and counting of animals, *European Conference on Computer Vision*, 255-270.
- Vapnik, V., 1998, *Statistical Learning Theory*/Vapnik V.–NY, John Wiley & Sons, Inc.
- 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.
- Wenzel, K. E., Masselli, A. ve Zell, A., 2012, Visual tracking and following of a quadcopter by another quadcopter, *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 4993-4998.
- Yuan, C., Liu, Z. ve Zhang, Y., 2015, UAV-based forest fire detection and tracking using image processing techniques, *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, 639-643.
- Zhou, G., Li, C. ve Cheng, P., 2005, Unmanned aerial vehicle (UAV) real-time video registration for forest fire monitoring, *Proceedings. 2005 IEEE International Geoscience and Remote Sensing Symposium, 2005. IGARSS'05.*, 1803-1806.

## EKLER

### EK-1 İnsansız Hava Araçları Talimatnamesi (SHT-İHA)

- MADDE 17 – (1) İHA0 ve İHA1 sınıfı İHA'lar için;
  - a) Ticari faaliyetler dışında sportif ve amatör amaçla gerçekleştirilecek uçuşlarda, bu Talimatta belirtilen kurallara uyulmak kaydı ile, 18inci Maddede yer alan bölgeler hariç Genel Müdürlükten uçuş izni alınması gerekmez. Bu sınıflar için mahallin en büyük mülki idare amiri tarafından yasaklanan yer ve zamanlarda uçuş yapmak yasaktır.
  - b) Meskun mahal, kalabalık veya çok kalabalık bölgeler dışında kalan bölgelerde, 18inci ve 19uncu Maddelerde belirtilen gerekliliklere uyulması kaydıyla ticari amaçlı uçuş yapılması halinde uçuş izni alınmasına gerek yoktur.
  - c) Meskun mahal, kalabalık veya çok kalabalık bölgelerde, yapılacak her türlü uçuşlarda ise 5 iş günü öncesinden Genel Müdürlük resmi internet sayfasında yer alan Uçuş İzni Talep Formu ile başvuru yapılır.
  - ç) Başvurulara ilişkin ilgili sivil ve askeri kurumlardan olumlu görüş alınmasını müteakip söz konusu sahaların NOTAM ile tüm hava sahası kullanıcılarına duyurulması sağlanır. Söz konusu taleplere ilişkin, ilgili kurumlardan olumsuz görüş alınması veya Genel Müdürlük tarafından uygun görülmemesi halinde uçuş izni verilmez.
  - d) Uçuş izni alınması halinde; işletici/pilot/şahıs tarafından Türkiye AIP'sinde yer alan hükümlere ve belirtilen diğer kurallara uyulması zorunludur.
- (2) İHA2 ve İHA3 sınıfı İHA'lar için;
  - a) Ticari faaliyetler dışında sportif ve amatör amaçla gerçekleştirilecek İHA uçuşunda gerekli koordinasyonun yapılabilmesi için 10 iş günü öncesinden Genel Müdürlüğe Uçuş İzni Talep Formu ile başvuru yapılır.
  - b) Başvurulara ilişkin ilgili sivil ve askeri kurumlardan olumlu görüş alınmasını müteakip söz konusu sahaların NOTAM ile tüm hava sahası kullanıcılarına duyurulması sağlanır. Söz konusu taleplere ilişkin ilgili kurumlardan olumsuz görüş alınması veya Genel Müdürlük tarafından uygun görülmemesi halinde uçuş izni verilmez.
  - c) Ticari amaçlı uçuşlarda 15 iş günü öncesinden Genel Müdürlüğe Uçuş İzni Talep Formu ile başvuru yapılır. Başvurulara ilişkin ilgili sivil ve askeri kurumlardan olumlu görüş alınmasını müteakip söz konusu sahaların NOTAM ile tüm hava sahası kullanıcılarına duyurulması sağlanır. Söz konusu taleplere ilişkin ilgili kurumlardan olumsuz görüş alınması veya Genel Müdürlük tarafından uygun görülmemesi halinde uçuş izni verilmez.

ç) İHA işletmecisi veya sahibi, başvurunun onaylanmasını takiben uçuşların düzenleneceği NOTAM'lanmış hava sahasından sorumlu ATC ünitesi ile acil durumlar haricinde operasyondan en az bir gün önce uçuşa ilişkin usuller üzerinde mutabakat sağlar. Bu müsaadeler, İHA operasyonları için geçerli olan Standart Harekât Usullerine dayalı olacaktır.

d) Standart Harekât Usulleri; normal usulleri, planlanmamış durumlara ilişkin usulleri, hava aracının acil durumdan kurtarılmasına diğer bir deyişle uçuşun sona erdirilmesine ilişkin usulleri içerecektir.

e) Başvuru dosyası, uçuşun düzenleneceği ayrılmış hava sahasının koordinatlarını, bu sahada icra edilecek uçuşun saatlerini, ayrılmış hava sahasında kullanılacak uçuş seviyesini, acil durum usullerini, İHA ve sistemlerinde kullanılacak frekanslar gibi ilgili bilgileri içerir. Genel Müdürlük ilgili diğer kurumlarla koordinasyon sonucunda uygun bulunması halinde uçuş müsaadesi ve NOTAM yayınlar.

f) Uçuş izni alınması halinde; işletici/pilot/şahıs tarafından Türkiye AIP'sinde yer alan hükümlere ve belirtilen diğer kurallara uyulması zorunludur.

- MADDE 20 – (1) Uçuşlar esnasında, diğer hava araçları ile ayırmayı sağlamak ve çarpışmayı önlemek İHA pilotunun sorumluluğundadır.
- MADDE 25 – (1) İHA işleticisi ve pilotu, İHA'nın terör saldırılarında veya kamunun güvenliğini tehlikeye düşürecek herhangi bir yasadışı eylemde kullanılmasını önlemek amacıyla İHA sisteminin ve aracının dışarıdan müdahalelere karşı güvenliğini sağlamakla yükümlüdür.
- GEÇİCİ MADDE 1 – (1) Bu Talimatın yayımlanmasından önce faaliyette bulunan İHA0 ve İHA1 sahipleri en geç 01/05/2016 tarihine kadar, sahip oldukları İHA'ları Genel Müdürlük tarafından oluşturulan internet tabanlı sisteme kayıt eder. Aksi halde söz konusu İHA'lar ile uçuş yapılmaz.

(2) Bu Talimatın yayımlanmasından önce faaliyette bulunan İHA2 ve İHA3 sahipleri en geç 01/05/2016 tarihine kadar, sahip oldukları İHA'ları Genel Müdürlüğe tescil ettirir. Aksi halde söz konusu İHA'lar ile uçuş yapılmaz.

## ÖZGEÇMİŞ

### KİŞİSEL BİLGİLER

**Adı Soyadı** : Mehmet Celalettin Ergene  
**Uyruğu** : T.C.  
**Doğum Yeri ve Tarihi** : Selçuklu - 01.03.1994  
**Telefon** : 0538 240 51 91  
**Faks** :  
**e-mail** : mcelalettin42@gmail.com

### EĞİTİM

Derece	Adı, İlçe, İl	Bitirme Yılı
Lise	: Karatay TOKİ Anadolu Lisesi, Karatay, Konya	2012
Üniversite	: Selçuk Üniversitesi, Selçuklu, Konya	2016

### İŞ DENEYİMLERİ

Yıl	Kurum	Görevi
2017-...	Solter Teknoloji Hizmet ve Danışmanlık Ltd. Şti.	Co-Founder, CTO

### UZMANLIK ALANI

Yapay Zeka, Görüntü İşleme, Gömülü Sistem Yazılımı, Drone Kontrol Sistemleri.

### YABANCI DİLLER

- **İngilizce** (TOEFL iBT:91, YDS:90, YÖKDİL:92,5)
- **Almanca** (A2)

### ULUSLARARASI KONFERANSLAR

- 1) Mehmet Celalettin Ergene, Akif Durdu, and Halil Cetin. "Imitation and learning of human hand gesture tasks of the 3D printed robotic hand by using artificial neural networks." *2016 8th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*. IEEE, 2016.
- 2) Mehmet Celalettin Ergene and Akif Durdu. "Robotic hand grasping of objects classified by using support vector machine and bag of visual words." *2017 International Artificial Intelligence and Data Processing Symposium (IDAP)*. IEEE, 2017.

**ULUSLARARASI YAYINLAR**

- 1) Akif Durdu, Üzeyir İlbay Bozkurt, and Mehmet Celalettin Ergene. "Design of a Distributed Control System with Fuzzy Logic Controller and PLC in Wireless Sensor Network Based Industrial Environments and Monitoring the System with RFID." *Uluslararası Mühendislik Araştırma ve Geliştirme Dergisi* 11.1: 173-191.
- 2) Akif Durdu, et al. "Image Processing based Task Allocation for Autonomous Multi Rotor Unmanned Aerial Vehicles." *Image* 9.11 (2018).

