



**NESNELERİN İNTERNETİ İÇİN KENAR BİLİŞİM  
TABANLI AKILLI GÖRSEL ALGILAMA ANALİZİ**

**Emrah ŞİMŞEK**

**Yüksek Lisans Tezi**

**Bilgisayar Mühendisliği Anabilim Dalı  
Bilgisayar Mühendisliği Bilim Dalı**

**Dr. Öğr. Üyesi Barış ÖZYER**

**2019**

**Her hakkı saklıdır**

**ATATÜRK ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**YÜKSEK LİSANS TEZİ**

**NESNELERİN İNTERNETİ İÇİN KENAR BİLİŞİM  
TABANLI AKILLI GÖRSEL ALGILAMA ANALİZİ**

**Emrah ŞİMŞEK**

**BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI  
Bilgisayar Mühendisliği Bilim Dalı**

**ERZURUM  
2019**

**Her hakkı saklıdır**



TEZ ONAY FORMU

NESNELERİN İNTERNETİ İÇİN KENAR BİLİŞİM TABANLI AKILLI GÖRSEL ALGILAMA  
ANALİZİ

Dr. Öğretim Üyesi Barış ÖZYER danışmanlığında, Emrah Şimşek tarafından hazırlanan bu çalışma, 17/07/2019 tarihinde aşağıdaki jüri tarafından Bilgisayar Mühendisliği Anabilim Dalı Bilgisayar Mühendisliği Bilim Dalı'nda yüksek lisans tezi olarak oybirliği / oy çokluğu (3/3) ile kabul edilmiştir.

Başkan: Doç. Dr. Eyüp Fahri KESKENLER

Üye : Dr. Öğretim Üyesi Barış ÖZYER

Üye : Dr. Öğretim Üyesi Bilal USANMAZ

İmza :

İmza :

İmza :

Yukarıdaki sonuç;

Enstitü Yönetim Kurulu'nun ~~08/09~~2019 tarih ve 32.../...69.. nolu kararı ile onaylanmıştır.

  
Prof. Dr. Mehmet KARAKAN Y.  
Enstitü Müdürü

Bu çalışma Bilimsel Araştırma Projelerini destekleme programı kapsamında desteklenmiştir.

Proje No: FYL-2018-6623

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

## ÖZET

Yüksek Lisans Tezi

### NESNELERİN İNTERNETİ İÇİN KENAR BİLİŞİM TABANLI AKILLI GÖRSEL ALGILAMA ANALİZİ

Emrah ŞİMŞEK

Atatürk Üniversitesi  
Fen Bilimleri Enstitüsü  
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Dr. Öğretim Üyesi Barış ÖZYER

Günlük yaşamda kullanılan nesnelerin, internete bağlanarak belirli işlevleri yapabilme kabiliyeti kazanması olarak ifade edilen Nesnelerin İnterneti, yapay zekâ, gömülü sistemler, yaygın bilişim ve bulut bilişim gibi birçok farklı teknoloji ile etkileşim halindedir. Bu teknolojilerin birlikte kullanımı, nesnelerin interneti uygulamalarının niteliklerini arttırmaktadır. Günümüzde gerçekleştirilmekte olan akıllı ev, akıllı okul, akıllı araba, akıllı şehir gibi Nesnelerin İnterneti projeleri, bu alanın kısa sürede insan hayatını daha fazla etkileyeceğini göstermektedir. Bu projelerin uygulanması ile birlikte yakın zamanda, kullanılacak nesne sayısının artması, algılama türlerinin ve olanaklarının gelişmesi ve bunlara bağlı olarak sunucu ve iletişim altyapısı ihtiyacının artması beklenmektedir. Görsel algılama, nesnelerin interneti çalışmalarının kapsamını genişletebilecek ve niteliğini arttırabilecek çok sayıda anlamlı veri elde edilmesini sağlayan önemli bir algılama türüdür. Büyük boyutlu veriler elde edilebilen Görsel algılama Nesnelerin İnterneti projeleri için, verilerin işleme, depolanma, iletme zorluklarının yanı sıra, iletişim ve sunucu alt yapılarının etkili kullanılması için uygun veri işleme mimarileri gereksinimi barındırmaktadır. Kenar bilişim, görsel algılama gibi sunucu ve iletişim ihtiyacı fazla olan uygulamalarda, veri iletimi gereksiniminin azaltılması, sunucu ve iletişim altyapısının yükünün hafifletilmesi amacıyla, bulut bilişim ile sunucularda yapılması gereken bazı işlemlerin ağın kenar kısmı olan nesnelere yapılmasını öneren yaklaşımdır. Bu tezde, nesnelerin interneti kapsamında görsel algılama problemi için kullanılacak kenar bilişim tabanlı çözümler araştırılmıştır. Çözümlerde uygulanan yöntemler, işlem, bellek ve veri iletişim gereksinimleri açısından, işlemlerin nesne ve sunucu arasında etkili bir şekilde paylaştırılabilmesi amacıyla oluşturulmuş farklı senaryolar kullanılarak analiz edilmiştir. Senaryoların uygulanabilmesi amacıyla, görsel algılama için kullanılan foto-kapan cihazının fonksiyonları temel alınarak, bilgisayarlı görme, görüntü işleme ve görüntü tanıma işlevleri bulunan bir akıllı nesne tasarımı gerçekleştirilmiştir. Tasarlanan akıllı nesne kullanılarak Atatürk Üniversitesi'nde iç ortamdan ve dış ortamdan hareketli cisimlere ait görüntü kaydının yapıldığı foto-kapan uygulamaları gerçekleştirilmiştir. Kaydedilen görüntüler üzerinde üniversitenin eğitim, yönetim ve güvenlik gibi faaliyetlerinde uygulanabilecek bilgisayarlı görme ve görüntü işleme tabanlı işlevleri içeren kenar bilişim tabanlı veri işleme mimarileri incelenmiştir. Uygulanan veri işleme mimarileri, gerçekleştirilecek akıllı şehir, akıllı kampüs, akıllı derslik gibi projeler için üniversite, belediye, askeriye gibi kurumların dijital dönüşüm projelerinde kullanılması ve katkı sağlaması açısından önem arz etmektedir. Gerçekleştirilen ilk senaryoda standart foto-kapan işlevi uygulanmış olup, diğerlerinde sırasıyla cisim bulunan görüntülerin tespit edilmesi, görüntüdeki cisim konumlarının tespit edilmesi, cisim görüntülerinin özelliklerinin çıkartılması ve cisimlerin tanınması işlevleri bulunmaktadır. Senaryoların işlem, bellek ve iletişim gereksinimleri ile cismin dijital ortamdaki temsiliyeti ve verinin tekrar kullanılabilmesi gibi parametreler göz önüne alındığında, cisimlere ait görüntü parçalarının sunucuya iletildiği üçüncü senaryonun, diğer senaryolara göre daha uygun olduğu sonucu elde edilmiştir.

2019, 204 sayfa

**Anahtar Kelimeler:** Nesnelerin İnterneti, Akıllı Nesne, Foto-kapan, Cisim Tespiti, Cisim Tanıma, Bilgisayarlı Görme, Görüntü Tanıma, Görüntü Özellikleri, Özellik Eşleştirme, Dijital Dönüşüm

## ABSTRACT

MS Thesis

### EDGE COMPUTING BASED INTELLIGENT VISUAL SENSING ANALYSIS FOR INTERNET OF THINGS

Emrah ŞİMŞEK

Atatürk University  
Graduate School of Natural and Applied Sciences  
Department of Computer Engineering

Supervisor: Asst. Prof. Dr. Barış ÖZYER

The Internet of Things, which is expressed as the ability of objects used in daily life to be able to perform certain functions by connecting to the internet, interacts with many different technologies such as artificial intelligence, embedded systems, ubiquitous computing and cloud computing. The combined use of these technologies increases the quality of Internet of Things applications. The Internet of Things projects such as smart home, smart school, smart car, smart city, which are being realized today, show that this area will affect human life more in a short time. With the implementation of these projects, it is expected that the number of objects to be used, the detection types and possibilities will be improved and the requirement for server and communication infrastructure will increase in the near future. Visual perception is an important type of perception that provides a large number of meaningful data that can broaden the scope and enhance the quality of Internet of Things projects. Visual perception with large data acquisition in addition to the difficulties of processing, storing and transmitting data, suitable data processing architectures are required for effective use of communication and server infrastructures for Internet of Things projects. Edge computing is the approach that proposes to perform some operations on the edge of the network, rather than on the servers by using cloud computing. Edge computing is implemented in order to reduce the requirement for data transmission and to alleviate the burden of the server and communication infrastructure. In this thesis, edge computing based solutions, that can be used for the visual perception problem within the Internet of Things, are investigated. The methods applied in the solutions were analyzed in terms of process, memory and data communication requirements by using different scenarios that were created to effectively share the processes between the object and the server. In order to implement the scenarios, a smart object design with computer vision, image processing and pattern recognition functions has been realized based on the functions of the camera-trap device used for visual sensing. By using designed smart object, camera-trap applications were performed at Atatürk University where the image of moving objects from indoor and outdoor environments was recorded. Edge computing based data processing architectures including computer vision and image processing based functions, that can be applied in the university's education, management and security activities, were examined on the recorded images. The applied data processing architectures are important in terms of contributing to the digital transformation projects of institutions such as universities, municipalities and military for projects such as smart city, smart campus and smart classrooms. In the first scenario, the standard camera-trap functions were applied and the subsequent scenarios had the functions of detecting the object in the image, determining the positions of the objects in the image, feature extraction of the object images and recognizing the objects. When the parameters such as the process, memory and communication requirements of the scenarios, the representation of the object in the digital environment and the reuse of the data are taken into consideration, the third scenario in which the image parts of the objects are transmitted to the server is found to be more appropriate than the other scenarios.

**2019, 204 pages**

**Keywords:** Internet of Things, Smart Thing, Camera-trap, Object Detection, Object Recognition, Computer Vision, Pattern Recognition, Image Features, Feature Matching, Digital Transformation

## TEŞEKKÜR

Yüksek Öğretim Kurulunun (YÖK) belirlemiş olduğu öncelikli alanlar kapsamında bulunan tez çalışmasının hazırlanmasında ve akademik çalışmalarında yaptığı katkılardan dolayı danışmanım Dr. Öğretim Üyesi Barış ÖZYER başta olmak üzere, Bilgisayarlı Görü ve Görüntü İşleme konularındaki yardımlarından dolayı Dr. Öğretim Üyesi Gülşah Tümüklü ÖZYER'e, Nesnelerin İnterneti alanındaki katkılarından dolayı Dr. Öğretim Üyesi Bilal USANMAZ'a, tez konusu ile ilgili desteklerinden dolayı Dr. Öğretim Üyesi Levent BAYINDIR ve mesai arkadaşım Arş. Gör. Mustafa Furkan KESKENLER'e teşekkürlerimi sunarım.

Tezimi, eğitim hayatımda hakkı ödenmeyecek emekleri bulunan babam İsmet ŞİMŞEK ve rahmetli annem Asalet ŞİMŞEK'e ithaf ediyorum.

**Emrah ŞİMŞEK**

**Temmuz, 2019**

## İÇİNDEKİLER

ÖZET.....	i
ABSTRACT.....	ii
TEŞEKKÜR.....	iii
SİMGELER ve KISALTMALAR DİZİNİ.....	vii
ŞEKİLLER DİZİNİ.....	viii
ÇİZELGELER DİZİNİ.....	xii
<b>1. GİRİŞ.....</b>	<b>1</b>
<b>2. KAYNAK ÖZETLERİ.....</b>	<b>11</b>
2.1. Nesnelerin İnterneti.....	11
2.1.1. Tanım ve kapsam.....	11
2.1.2. Katmanlar ve elementler.....	15
2.1.3. Gelecek vizyonu.....	18
2.2. Nesnelerin İnternetinde Cisim Tespit Etme ve Tanıma.....	23
2.2.1. Tanım ve kullanım amaçları.....	23
2.2.2. Foto-kapan tanımı ve kullanım amaçları.....	25
2.2.3. Yapılan çalışmalar.....	26
2.3. Bilgisayarlı görü.....	33
2.4. Hareketli cisim tespiti.....	37
2.4.1. Arka plan çıkarma yöntemleri.....	38
2.4.2. Çerçeve farkı yöntemi.....	40
2.4.3. Üç ve dört çerçeve farkı yöntemleri.....	42
2.5. Derin Öğrenme ile Cisim Tespiti.....	43
2.6. Örüntü Tanıma ve Sınıflandırma.....	45
2.6.1. K en yakın komşu algoritması (kNN).....	45
2.6.2. Rastgele orman algoritması (Random Forest).....	47
2.6.3. Destek vektör makineleri (Support Vector Machines).....	48
2.6.4. Evrişimli sinir ağları.....	50
2.7. Öznitelik çıkarma ve öznitelik eşleştirme yöntemleri.....	57
2.7.1. HOG (Eğimlerin yönelim histogramı).....	62

2.7.2. LBP (Yerel ikili örüntü) .....	63
2.7.3. Renk histogramı .....	65
2.7.4. SIFT (Ölçek değişmez öznitelik dönüşümü).....	66
2.7.5. SURF (Hızlandırılmış sağlam öznitelikler).....	67
2.7.6. BRIEF (İkili sağlam bağımsız temel öznitelikler) .....	68
2.7.7. ORB (Oriented fast and rotated brief) .....	68
2.7.8. Öznitelik eşleştirme (Feature matching) .....	69
2.7.9. K en yakın komşu eşleştiricisi (kNN matcher) .....	70
2.7.10. Kaba kuvvet eşleştiricisi (Brute force).....	70
2.8. Ön ve Son İşleme .....	72
2.8.1. Gauss filtresi.....	72
2.8.2. Medyan filtre .....	73
2.8.3. OTSU eşikleme .....	73
2.8.4. Kabarcık algılama.....	74
2.8.5. Histogram eşitleme.....	75
2.8.6. Kayan pencere yöntemi ve bağlı bileşen analizi .....	76
<b>3. MATERYAL ve YÖNTEM.....</b>	<b>78</b>
3.1. Akıllı Nesne Tasarımı ve Veri İşleme Senaryoları .....	78
3.2. Görsel Algılama Analizi.....	87
3.2.1. Arka plan çıkarma ve çerçeve farkı ile hareketli cisim tespiti .....	87
3.2.2. Hareketli cisim tespitinin örüntü tanıma yöntemleri ile iyileştirilmesi.....	99
3.2.3. Yerel öznitelikler ve öznitelik eşleştirme ile cisim tespiti .....	107
3.2.4. YOLO ile insan tespiti.....	121
3.2.5. Akıllı üç çerçeve farkı .....	127
3.2.6. Örüntü Tanıma ve Sınıflandırma.....	143
3.3. Veri Setleri .....	148
<b>4. ARAŞTIRMA BULGULARI ve TARTIŞMA.....</b>	<b>152</b>
4.1. Yazılım Geliştirme Platformu, Gömülü Sistem ve Görsel Uygulama .....	152
4.2. Bulgular .....	157
4.3. Tartışma.....	167
<b>5. SONUÇ .....</b>	<b>170</b>



5.1. Öneriler.....	173
KAYNAKLAR .....	177
ÖZGEÇMİŞ .....	185



## SİMGELER ve KISALTMALAR DİZİNİ

$\Sigma$	Toplam Sembolü
$\otimes$	Mantıksal Ve Operatörü
–	Matematiksel Fark Operatörü

### Kısaltmalar

BOVW	Görsel Kelime Çantası
BOW	Kelime Çantası
BRIEF	İkili Dayanıklı Bağımsız Temel Öznitelikler
cm	santi metre
CNN	Evrışimli sinir ağları
DVM	Destek Vektör Makineleri
FAST	Hızlandırılmış Segment Testi Öznitelikleri
FCN	Tam Bağlı Katmanlar
GPS	Küresel Konumlandırma Sistemi
HOG	Yönlendirilmiş Eğim Histogramı
IoU	Birleşimlerin Kesişimi
IP	İnternet protokol
LBP	Yerel İkili Örüntü
m	metre
ORB	Odaklı FAST ve Döndürülmüş BRIEF
PIL	Python Görüntü Kütüphaneleri
RCNN	Bölge Tabanlı Evrışimli Sinir Ağları
RFID	Radyo Frekansı ile Tanımlama
RGB	Kırmızı Yeşil Mavi
SD	Güvenli Sayısal Hafıza Kartı
SIFT	Ölçek Değişmez Öznitelik Dönüşümü
SURF	Hızlandırılmış Sağlam Öznitelikler
TCP-IP	İletim Kontrol Protokolü – İnternet Protokolü
UDP	Kullanıcı Datagram Protokolü
Wi-Fi	Kablosuz Bağlantı Alanı

## ŞEKİLLER DİZİNİ

Şekil 1.1. Nesnelerin interneti kullanım alanları .....	1
Şekil 1.2. Nesnelerin internetinin ilişkili olduğu alanlar .....	3
Şekil 1.3. Nesnelerin internetinde veri tabanından anlam keşfi mimarisi .....	4
Şekil 1.4. Bulut ve sis/kenar bilişim ilişkisi.....	6
Şekil 1.5. Foto-kapan görüntülerinde cisim bulunma durumu .....	8
Şekil 2.1. Nesnelerin interneti projelerinin pazar payları .....	12
Şekil 2.2. Bakış açılarına göre nesnelerin interneti.....	13
Şekil 2.3. Literatürde önerilen nesnelerin interneti katmanları.....	16
Şekil 2.4. Algılama katmanının işlevleri.....	17
Şekil 2.5. Yerel öznitelikler ile veri tabanından görüntü alma .....	24
Şekil 2.6. Örnek foto-kapan görüntüsü .....	26
Şekil 2.7. Ön işleme katmanı kullanılan uygulama .....	30
Şekil 2.8. Iou değerlerine göre pencere kesişimleri.....	33
Şekil 2.9. Cisim tespit yaklaşımları .....	34
Şekil 2.10. Cisim sınıflandırma .....	35
Şekil 2.11. Çerçeve farkı yöntemi .....	41
Şekil 2.12. Yolo mimarisinde cisim tespiti.....	44
Şekil 2.13. K en yakın komşu algoritması .....	46
Şekil 2.14. Rastgele orman algoritması .....	47
Şekil 2.15. Doğrusal dvm .....	49
Şekil 2.16. Doğrusal olmayan dvm.....	50
Şekil 2.17. Evrişimli sinir ağı mimarisi .....	51
Şekil 2.18. Hog öznitelik dönüşümü.....	63
Şekil 2.19. Lbp algoritması.....	64
Şekil 2.20. Renk histogramı.....	65
Şekil 2.21. Öznitelik eşleştirme tabanlı cisim tespiti.....	70
Şekil 2.22. İşlemlere ait örnek görüntüler.....	73
Şekil 2.23. Kabarcık tespit yöntemleri a) log b) dog c) doh).....	75
Şekil 2.24. Histogram eşitleme işlemi, öncesi (a) ve sonrası (b).....	75

Şekil 2.25. Kayan pencere yaklaşımı .....	76
Şekil 2.26. Görüntüdeki bağlı bileşenlerin tespit edilmesi .....	77
Şekil 3.1. Akıllı nesne tasarımı .....	78
Şekil 3.2. Akıllı nesnede kullanılan elektronik geliştirme kartı.....	79
Şekil 3.3. Hareket algılayıcısı ve bağlantı şeması.....	80
Şekil 3.4. Çalışma için tasarlanan akıllı nesne.....	80
Şekil 3.5. Akıllı nesne işlevleri .....	81
Şekil 3.6. Önerilen sistem modeli ve gerçekleştirilecek işlemler .....	85
Şekil 3.7. Foto-kapan görüntülerinde hareketli cisim tespiti için önerilen yöntem .....	88
Şekil 3.8. Görüntülerdeki hareketli cisimlerin elde edilmesi.....	89
Şekil 3.9. Ardışık görüntülerdeki ışık şiddeti değişimi.....	90
Şekil 3.10. Işık şiddeti farkı olan görüntülerin çerçeve farkı çıktısı .....	90
Şekil 3.11. Çerçeve farkı çıktısı, gauss bulanıklaştırma ve otsu eşikleme .....	90
Şekil 3.12. Kabarcık analizi algoritmasının sonuçları .....	92
Şekil 3.13. Kayan pencere tabanlı konumlandırma .....	92
Şekil 3.14. Sınırlayıcı kutu tespiti.....	93
Şekil 3.15. Önerilen yöntem ile tespit edilen hareketli cisimlere ait görüntüler.....	95
Şekil 3.16. Örnek foto-kapan görüntüleri .....	100
Şekil 3.17. Uygulanan yöntemde ışık şiddeti değişimi içeren sonuçlar .....	101
Şekil 3.18. Uygulanan yöntemde arka plan hareketi içeren sonuçlar .....	102
Şekil 3.19. Uygun olarak etiketlenen sonuçlar .....	102
Şekil 3.20. Klasik yönteme ait çıktıların sınıflandırılması .....	104
Şekil 3.21. Örüntü tabanlı iyileştirme için önerilen sistem modeli.....	104
Şekil 3.22. Öznitelik eşleştirme tabanlı cisim tespiti için önerilen sistem modeli .....	108
Şekil 3.23. Veri setine ait örnek görüntüler .....	109
Şekil 3.24. Cisimlere ait kırılmış ve arka plandan ayrılmış örnek görüntüler .....	109
Şekil 3.25. Öznitelik eşleştirme sonucuna ait örnek görüntü.....	110
Şekil 3.26. Eşik değere göre öznitelik eleme sonrası sınırlayıcı kutu değişimi.....	113
Şekil 3.27. Yüzdellik tabanlı aykırılık tespiti sonrası cisim tespit başarıları .....	114
Şekil 3.28. Yüzdellik tabanlı aykırılık tespiti sonrası sınırlayıcı kutu boyutları.....	115
Şekil 3.29. Yüzdellik tabanlı aykırılık tespiti sonrası eşleşen öznitelik sayıları.....	116
Şekil 3.30. Ortalama tabanlı aykırılık tespiti sonrası cisim tespit başarıları.....	117

<b>Şekil 3.31.</b> Ortalama tabanlı aykırılık tespiti sonrası sınırlayıcı kutu boyutları .....	117
<b>Şekil 3.32.</b> Ortalama tabanlı aykırılık tespiti sonrası eşleşen öznitelik sayıları.....	118
<b>Şekil 3.33.</b> K en yakın komşu öznitelik eleme ile cisim tespit başarıları .....	118
<b>Şekil 3.34.</b> K E en yakın komşu öznitelik eleme ile sınırlayıcı kutu boyutları.....	119
<b>Şekil 3.35.</b> K en yakın komşu öznitelik eleme ile eşleşen öznitelik sayısı.....	120
<b>Şekil 3.36.</b> Cisim tespit başarı oranları ile eşleşen öznitelik sayıları ilişkisi .....	120
<b>Şekil 3.37.</b> Zorluk barındıran foto-kapan görüntüleri.....	121
<b>Şekil 3.38.</b> Yolo mimarisi ile cisim tespiti.....	122
<b>Şekil 3.39.</b> Foto-kapan veri seti .....	123
<b>Şekil 3.40.</b> Tespit edilemeyen 17 insan görüntüsü .....	124
<b>Şekil 3.41.</b> İnsan olarak tespit edilmiş hatalı görüntüler.....	125
<b>Şekil 3.42.</b> Tespit edilememiş 8 insan içeren görüntü .....	126
<b>Şekil 3.43.</b> Hareketli cisim tespiti için önerilen yöntem.....	127
<b>Şekil 3.44.</b> İlk aşama a) çerçeve farkı b) medyan filtre c) genel eşikleme .....	128
<b>Şekil 3.45.</b> Üst üste gelmeyen çıktıyı veren görüntülerin seçilmesi .....	129
<b>Şekil 3.46.</b> Cisim konumları ayrık olan çerçeve farkı çıktılarının seçilmesi .....	130
<b>Şekil 3.47.</b> Ardışık görüntülerdeki çerçeve farklarının değişimi.....	131
<b>Şekil 3.48.</b> İkinci aşama .....	131
<b>Şekil 3.49.</b> Seçilen 1., 2. ve 3. görüntülerdeki hareketli cisimlerin tespit edilmesi.....	132
<b>Şekil 3.50.</b> Diğer görüntülerdeki hareketli cisim tespiti .....	133
<b>Şekil 3.51.</b> Önerilen yöntemde hareketli cisim tespitinin iyileştirilmesi .....	137
<b>Şekil 3.52.</b> Benzer çalışmalara ait sonuçlar .....	138
<b>Şekil 3.53.</b> Önerilen sistemdeki çıktılar .....	138
<b>Şekil 3.54.</b> Önerilen yöntem ile farklı görüntülerde elde edilmiş sonuçlar .....	139
<b>Şekil 3.55.</b> Walk2.mpg'de karşılaşılan problem .....	141
<b>Şekil 3.56.</b> Örüntü tanıma ile sınıflandırma için önerilen sistem modeli .....	144
<b>Şekil 3.57.</b> Örnek foto-kapan görüntüleri .....	149
<b>Şekil 3.58.</b> Hareketli cisim tespiti için kullanılan veri setinden örnek görüntü.....	150
<b>Şekil 3.59.</b> Akıllı foto-kapan ile kaydedilmiş görüntüler .....	151
<b>Şekil 3.60.</b> Videolara ait örnek görüntüler .....	151
<b>Şekil 4.1.</b> Uygulamaya ait görüntü .....	153
<b>Şekil 4.2.</b> Sunucuda kullanılan görsel uygulama .....	154

<b>Şekil 4.3.</b> Sunucuda kullanılan görsel uygulamada özellik belirleme sayfası .....	155
<b>Şekil 4.4.</b> Çalışmadaki iletişim sistemi .....	156
<b>Şekil 4.5.</b> Hareketli cismin sunucuda gerçek zamanlı olarak görüntülenmesi .....	157
<b>Şekil 4.6.</b> İndoor2-49 videosunda seçilen görüntüler .....	167



## ÇİZELGELER DİZİNİ

Çizelge 2.1. İletişim protokollerinin özellikleri.....	18
Çizelge 2.2. Arka plan çıkarma yöntemleri ve performans analizi.....	40
Çizelge 2.3. Öznitelik detektörleri ve özellikleri.....	59
Çizelge 2.4. Öznitelik detektörlerinin değişmezlikleri .....	61
Çizelge 3.1. Akıllı nesne ile hareket algılandığında görüntü kaydetme algoritması .....	83
Çizelge 3.2. Gerçek zamanlı hareketli cisim tespiti için istemci algoritması .....	83
Çizelge 3.3. Gerçek zamanlı hareketli cisim tespiti için sunucu algoritması .....	84
Çizelge 3.4. Tezde uygulanarak performansları analiz edilen senaryolar .....	86
Çizelge 3.5. El ile cisim tespiti yapılan görüntülere ait sonuçlar.....	94
Çizelge 3.6. Birinci veri setinde gauss bulanıklığı ile elde edilen sonuçlar .....	96
Çizelge 3.7. Birinci veri setinde medyan filtre ile elde edilen sonuçlar .....	96
Çizelge 3.8. İkinci veri setinde gauss bulanıklığı ile elde edilen sonuçlar .....	97
Çizelge 3.9. İkinci veri setinde medyan filtre ile elde edilen sonuçlar .....	97
Çizelge 3.10. Gauss bulanıklaştırma hata matrisi .....	98
Çizelge 3.11. Her iki veri setine ait medyan filtre hata matrisi .....	98
Çizelge 3.12. Gauss bulanıklaştırma ve medyan filtre kıyaslaması.....	99
Çizelge 3.13. Klasik yöntemler ile elde edilmiş sonuçlar.....	105
Çizelge 3.14. Önerilen yöntem kullanılarak iyileştirilmiş sonuçlar .....	106
Çizelge 3.15. İyileştirmede kullanılan yöntemlerin sayısı.....	106
Çizelge 3.16. Rastgele orman sınıflandırma işlemi hata matrisi .....	103
Çizelge 3.17. Öznitelik eşleştirme algoritması .....	110
Çizelge 3.18. K en yakın komşu öznitelik seçme ve eleme algoritması.....	111
Çizelge 3.19. Özniteliklerin tanımlayıcı boyutları, çıkarma ve eşleştirme süreleri.....	114
Çizelge 3.20. Iou değerlerine göre tespit sonuçları.....	125
Çizelge 3.21. Veri setinin birinci bölümüne ait hata matrisi .....	125
Çizelge 3.22. Veri setinin ikinci bölümüne ait hata matrisi.....	126
Çizelge 3.23. Ayrık konumlardaki cisimlerin görüntülerinin seçilmesi .....	130
Çizelge 3.24. Hareketli cisim tespitinin iyileştirilmesi için kullanılan algoritma.....	136
Çizelge 3.25. Walk1.mpg hata matrisi.....	140

<b>Çizelge 3.26.</b> Walk2.mpg hata matrisi.....	140
<b>Çizelge 3.27.</b> Walk3.mpg hata matrisi.....	141
<b>Çizelge 3.28.</b> Tarafımızca toplanmış video veri seti ile ilgili bilgiler .....	142
<b>Çizelge 3.29.</b> Video10.mp4 hata matrisi.....	142
<b>Çizelge 3.30.</b> Video1.mp4 hata matrisi.....	143
<b>Çizelge 3.31.</b> Video27.mp4 hata matrisi.....	143
<b>Çizelge 3.32.</b> Öznitelik dönüşüm yöntemlerinde kullanılan parametreler .....	146
<b>Çizelge 3.33.</b> Dengesiz veri setinin sınıflandırma işlemine ait doğruluk oranları.....	146
<b>Çizelge 3.34.</b> Sınıflandırma işlemine ait hata matrisi .....	147
<b>Çizelge 3.35.</b> Dengeli veri setinin sınıflandırma işlemine ait doğruluk oranları.....	147
<b>Çizelge 3.36.</b> Gündüz kaydedilmiş görüntülerin sınıflandırma başarıları .....	148
<b>Çizelge 3.37.</b> Gece kaydedilmiş görüntülerin sınıflandırma başarıları .....	148
<b>Çizelge 3.38.</b> Tarafımızca toplanmış video veri seti ile ilgili bilgiler .....	150
<b>Çizelge 3.39.</b> Veri setine ait görüntü ve cisim sayıları .....	151
<b>Çizelge 4.1.</b> Veri setlerindeki görüntülerin ve cisimlerin özellikleri.....	158
<b>Çizelge 4.2.</b> Hareketli cisim bulunan görüntülerin tespit başarıları (%)......	159
<b>Çizelge 4.3.</b> Hareketli cisim bulunan görüntülerin tespit süreleri (mili saniye) .....	159
<b>Çizelge 4.4.</b> Hareketli cisimlerin sınırlayıcı kutu tespit süreleri (mili saniye) .....	160
<b>Çizelge 4.5.</b> Tespit edilen hareketli cisimlerin sayısı .....	160
<b>Çizelge 4.6.</b> (False pozitive) Hatalı tespit edilen cisimlerin sayısı .....	161
<b>Çizelge 4.7.</b> (False negative) Hatalı tespit edilen cisimlerin sayısı .....	161
<b>Çizelge 4.8.</b> (True pozitive) Doğru tespit edilen cisimlerin sayısı .....	161
<b>Çizelge 4.9.</b> Hassasiyet (precision) değerleri.....	162
<b>Çizelge 4.10.</b> Anımsama (recall) değerleri .....	162
<b>Çizelge 4.11.</b> Hareketli cisimlere ait modellerin bellek ve zaman çizelgesi .....	162
<b>Çizelge 4.12.</b> Hareketli cisimler ile oluşturulan modellerin sınıf içi başarıları (%) .....	163
<b>Çizelge 4.13.</b> Hareketli cisimlerin oluşturulan modellerinin bellekteki boyutları .....	163
<b>Çizelge 4.14.</b> Hareketli cisim görüntülerine ait modellerin bellekteki boyutları .....	164
<b>Çizelge 4.15.</b> 4. Senaryoya ait cisim görüntülerinin öznitelik boyutları (Kbyte).....	164
<b>Çizelge 4.16.</b> 5. Senaryodaki doğrusal dvm sınıflandırma modellerinin boyutları .....	165
<b>Çizelge 4.17.</b> 5. Senaryodaki doğrusal olmayan dvm modellerinin boyutları .....	165
<b>Çizelge 4.18.</b> 5. Senaryodaki knn sınıflandırma modellerinin boyutları (Kbyte).....	165



<b>Çizelge 4.19.</b> 5. senaryodaki rastgele orman sınıflandırma modellerinin boyutları ....	166
<b>Çizelge 4.20.</b> Ön ve son işleme aşamalarının ortalama zaman gereksinimi.....	166
<b>Çizelge 5.1.</b> Senaryoların bellek ve veri iletişim gereksinimi (%) .....	172
<b>Çizelge 5.2.</b> Senaryoların akıllı nesne için çalışma zamanı ihtiyacı (mili saniye) .....	173



## 1. GİRİŞ

Nesnelerin interneti, insanların günlük yaşamını kolaylaştırmak üzere tasarlanan, kameralar, algılayıcılar, iletişim modülleri, mikro denetleyicilerden oluşan internete bağlı nesnelerin oluşturduğu ağ olarak tanımlanmaktadır (Li *et al.* 2015). Nesnelerin interneti, ses, görüntü, hareket, ışık, duman, nem, sıcaklık algılayıcılarının gömülü sistemlerde kullanılarak ev aletleri, araçlar, kapılar, asansörler, klimalar gibi günlük hayatta kullanılan nesnelerin faydalarını arttırmasının yanı sıra, bu cihazların çevrimiçi olarak erişilebilir olması ve internet üzerinden birbirine bağlanması açısından dönüm noktasıdır (Whitmore *et al.* 2015). Şekil 1.1'de görüldüğü üzere nesnelerin interneti ulaşımdan sağlığa, kişisel platformlardan sosyal hayata kadar birçok alanda uygulamalar içermektedir.

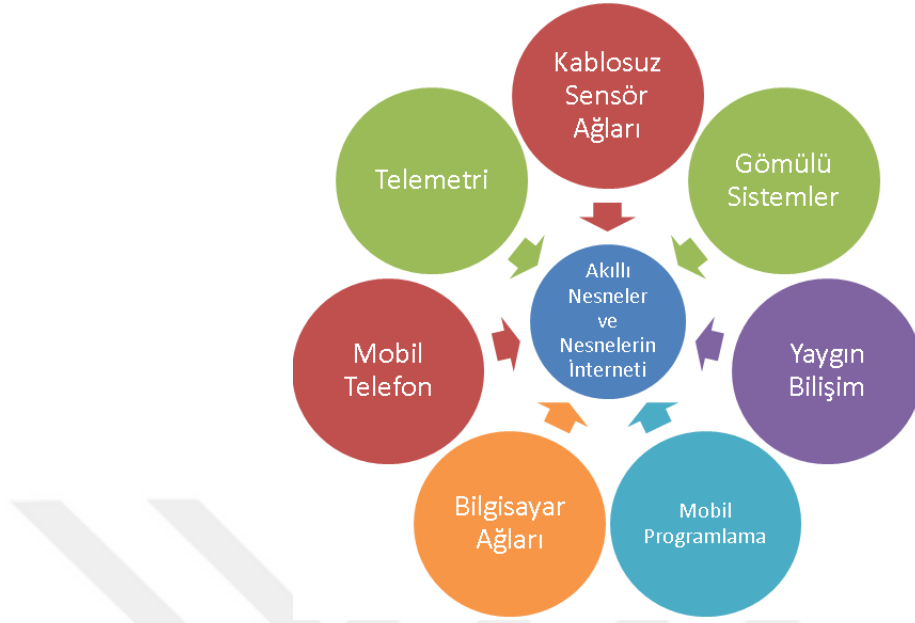


**Şekil 1.1.** Nesnelerin interneti kullanım alanları (Atzori *et al.* 2010)

Nesnelerin internetinin kapsamı, maliyeti azalan gömülü sistemlerin, yaygınlaşan iletişim teknolojilerinin ve yapay zekâ yöntemlerinin kullanılması ile genişlemektedir. Mikro denetleyici, algılayıcılar ve iletişim modülleri gibi gömülü sistem

teknolojilerindeki gelişmeler internet nesnelere daha işlevsel, akıllı ve daha faydalı olmalarının önünü açmaktadır. Bu gelişmeler nesnelere internetin insan hayatını kolaylaştıran en temel teknoloji olmasını, büyük bölgelere ve kitlelere hizmet etmesini sağlamaktadır. Algılayıcı teknolojilerinin üretim maliyetlerinin azalması ve nesnelere internetin potansiyelinin fark edilmesi ile daha önce elektronik gibi görünmeyen gıda, giyim, ev aletleri gibi eşyaların dijital öncelikli internet nesnesi olmasını sağlamaktadır. Buna bağlı olarak 2020’de 50 milyar nesnenin internete bağlı olması tahmin edilmektedir. (Swan, 2012).

İnternetin gerçek zamanlı lokalizasyon ve gömülü algılayıcılar gibi yeni teknolojilerin birleşimi, günlük nesnelere, çevrelerini anlayabilen ve tepki verebilen akıllı nesnelere dönüştürmemizi sağlamaktadır. Bu nesnelere, nesnelere internetin yapı taşı oluşturulmaktadır (Kortuem *et al.* 2009). Akıllı nesne, veri elde etmenin yanı sıra hizmet üretebilen, hizmet kullanabilen, ortak bir hedefe yönelik diğer nesnelere işbirliği yapabilen, bu yeni hizmetleri keşfeden, yeni tanıdık cihazlar edinen, bilgi alışverişi, dış hizmetlere bağlanma, diğer nesnelere işlevlerini kullanan, ortak bir hedef doğrultusunda iş birliği yapan nesne olarak ifade edilmektedir (Atzori *et al.* 2014). Ortam zekâsı terimi, akıllı nesnelere çevrelerini anlamaları, insanlarla etkileşimde bulunmaları ve karar vermeleri anlamına gelmektedir. Akıllı nesnelere dolu bir dünya, iş süreçlerini ve insanların yaşamlarını iyileştirmek için büyük bir umut vaat etmektedir (Whitmore *et al.* 2015). Şekil 1.2’de gösterildiği üzere nesnelere interneti ve akıllı nesne kavramı, her biri yaygın olarak çalışılmakta olan yaygın bilişim, gömülü sistemler, bilgisayar ağları, kablosuz algılayıcı ağları, mobil programlama gibi birçok alanın birleşiminden oluşmaktadır (Vasseur and Dunkels 2010). Bu tanımlamalar doğrultusunda akıllı nesnelere algılama, iletişim ve hizmet işlemlerinde kullanıcıdan bağımsız olarak daha önce belirlenmiş görevleri yerine getirmeleri beklenmektedir.

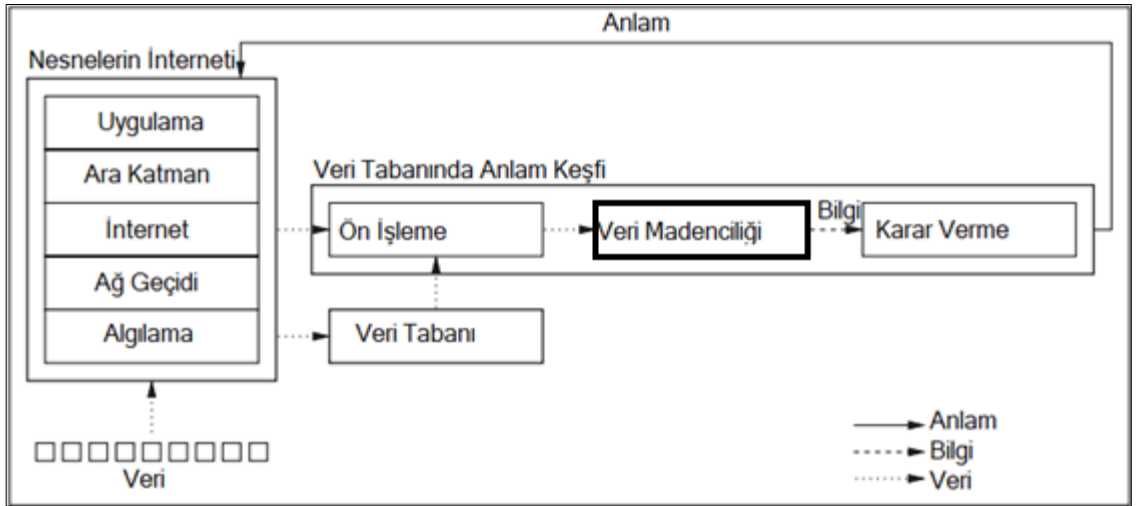


**Şekil 1.2.** Nesnelerin internetinin ilişkili olduğu alanlar (Vasseur and Dunkels 2010)

Nesnelerin interneti temel olarak 3 katmandan oluşmaktadır (Al-Fuqaha *et al.* 2015). Farklı tanımlamalar olmasına rağmen temel olarak bu katmanlar algılama, iletişim, uygulama olarak ifade edilmektedir. Algılama işlemi kısaca fiziksel dünyadaki bilgilerin, dijital dünyaya aktarılması olarak ifade edilmektedir. Algılama katmanında maliyet, boyut, enerji tüketimi, dağıtım, heterojenlik, iletişim ve ağ hususlarına dikkat edilmesi gerekmektedir (Li *et al.* 2015). Akıllı nesnelere ise algılama katmanı, sıradan veri toplayıp gönderme işlevinin yanı sıra, iletişim trafiğinin azaltılması ve sunucu iş yüklerinin azaltılması gibi zaman, bellek ve işlem tasarrufu açısından önemli görevler üstlenmektedir. Akıllı nesne ile ilgili en temel beklenti, aldığı algılayıcı verisini doğrudan merkeze iletmesi yerine, bir veri ön işleme aşaması gerçekleştirmesi ve hatta veriyi işleyerek ham veri yerine bir sonuç verisini merkeze iletmesidir. Bu görevin insan müdahalesi olmadan, diğer nesnelere yardımlaşarak ve minimum kaynak kullanımı ile gerçekleştirmesi gerekmektedir (Lin *et al.* 2017). İletişim katmanı, nesnelere tarafından üretilen verileri güvenli kanallar üzerinden hizmet katmanına aktarmaktadır. Veriler, RFID, 3G, GSM, UMTS, WiFi, Bluetooth, kızılötesi, ZigBee, vb. gibi çeşitli teknolojiler yoluyla aktarılabilmektedir. Ayrıca, bulut bilişim ve veri yönetimi işlemleri gibi diğer işlevler bu katmanda ele alınmaktadır (Yang *et al.* 2011). Hizmet ve arayüz

katmanı, Nesnelerin İnterneti programcılarının belirli bir donanım platformunu dikkate almadan heterojen nesnelere çalışmasını sağlamaktadır. Ayrıca, bu katmanda nesnelere alınan veriler işlenmekte, kararlar alınmakta ve gerekli hizmetler ağ protokolleri üzerinden servise sunulmaktadır (Al-Fuqaha *et al.* 2015).

Nesnelerin, internet üzerinden gönderdiği büyük ölçekli veriler Şekil 1.3'te mimarisi gösterildiği üzere Veri Madenciliği ve Büyük Veri yöntemleri ile işlenebilmektedir (Al-Fuqaha *et al.* 2015; Bin *et al.* 2010; Chen *et al.* 2015; Qin *et al.* 2016; Tsai *et al.* 2014). Çok hızlı büyüyen bu sistemde artan veri trafiğinin ve veri işleme artışının, ağ hizmetini ve alt yapıyı meşgul edeceği görünen bir gerçektir. İletişim teknolojilerindeki gelişmeler artan bant genişliği ve veri iletim hızını kalitesine ek olarak, kenar bilişim tabanlı çözümler ile sunucuların iş yüklerinin ve iletişim gereksinimlerinin büyük oranda azalması beklenmektedir. Ayrıca kenar bilişimin bilgisayarlı görü, büyük veri, makine öğrenmesi gibi yapay zekâ teknolojilerinin uygulanmasını kolaylaştıracağı ve yaygınlaştıracağı anlaşılmaktadır (Yi *et al.* 2015). Bu nedenle nesnelerin internetinin gelecek vizyonunda kenar bilişim tabanlı bir alt yapı kullanılması önerilmektedir (Al-Fuqaha *et al.* 2015; Lin *et al.* 2017; Wang *et al.* 2018; Yi *et al.* 2015).

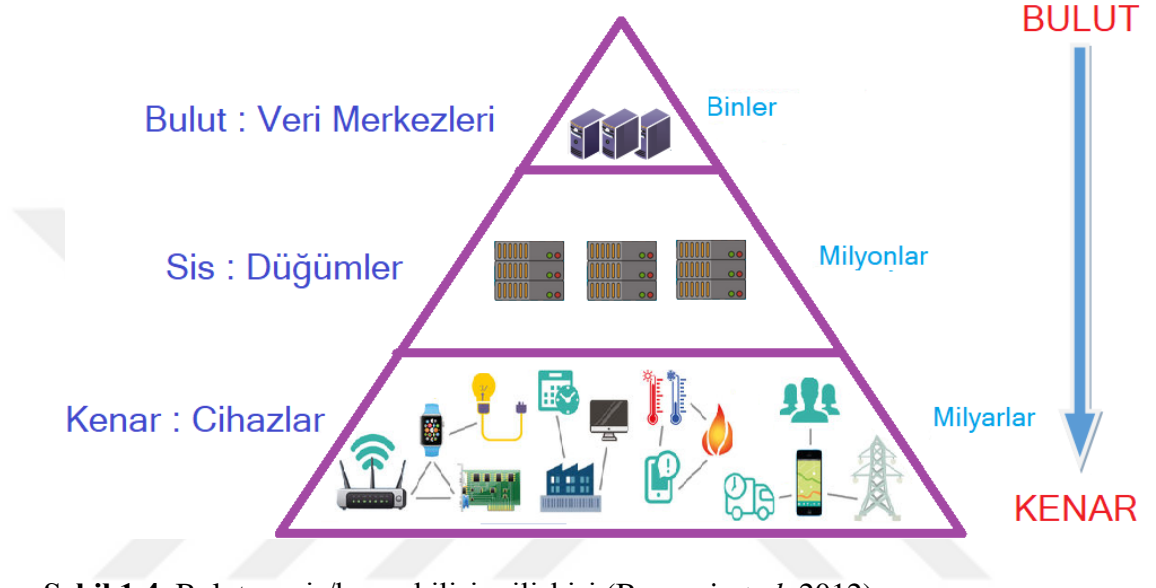


**Şekil 1.3.** Nesnelerin internetinde veri tabanından anlam keşfi mimarisi (Tsai *et al.* 2014)

Kenar bilişimde temel olarak nesnenin problemlere özel daha fazla işlev sahibi olması ve sunucuların görevlerini yerine getirebilmesi hedeflenmektedir (Yi *et al.* 2015). Bu amaçla kullanılacak nesnelere akıllı algılama, iş birliği, kaynak paylaşımı, kendi servislerini genel ağa sunma gibi gelişmiş özelliklerin bulunması ve bu hizmetleri insandan bağımsız olarak, daha özerk bir altyapı ile gerçekleştirmesi gerekmektedir (Atzori *et al.* 2010; J. Lin *et al.* 2017; Vermesan and Friess 2013; Yi *et al.* 2015). Şekil 1.3'te gösterildiği üzere nesnelere, elde ettiği veriler üzerinde uyguladıkları ön işlemlerden yapay zekâ tabanlı yöntemler ile bilgi keşfi veya karar verme işlemleri gerçekleştirilmektedir (Tsai *et al.* 2014). Bu sayede elde ettikleri ham veri, sonuç bilgisine dönüşebilecek faydalı veri haline getirilmektedir. Bilgisayarlı görü, makine öğrenmesi gibi yapay zekâ çalışmalarında kullanılan verilerin büyüklüğü, bu çalışmalardan elde edilecek sonucu doğrudan etkilemektedir. Bu nedenle veriler üzerinde gerçekleştirilecek işlemlerin, kısıtlı veri depolama imkânı olan kenar kısım ve daha fazla işlem yapma ve depolama imkânı olan sunucular arasında etkili bir biçimde paylaştırılmasının faydalı olacağı anlaşılmaktadır (Yi *et al.* 2015). Şekil 1.4'te görüldüğü üzere bulut sisteminden kenar sistemlere doğru geçildikçe görülen sayı artışı, kenar cihazlarda yapılacak işlemlerin bulut sistemlerdeki sunucuların iş yükünü büyük miktarda azaltacağı anlaşılmaktadır (Bonomi *et al.* 2012).

Kenar bilişimde, verilerin sunucuda veya bulutta işlenmesi yerine nesnede veya ağın uç kısmında işlenmesi ve saklanması, bu sayede merkezdeki işlem yükünün azaltılması ve internete gönderilmesi gerekmeyen verilerin nesnede kullanılması ile gereksiz ağ trafiğinin önüne geçilmesi hedeflenmektedir (Lin *et al.* 2017). Günümüzde öne sürülen 5G ve nesnelere interneti paradigması, internet nesnelere ve yapay zekânın birleşimi olarak ifade edilmektedir. Bu tanıma göre akıllı nesnelere, cisim tanıma, doğal dil işleme ve akıllı kontrol gibi zor uygulamaları gerçekleştirmek için yüksek fonksiyonel işlevler içeren cihazlar olması beklenmektedir (Wang *et al.* 2018). Görüntü tabanlı nesnelere interneti uygulamalarında veri miktarının fazla olması sebebiyle kenar bilişim tabanlı çözümler etkili sonuçlar üretmektedir (Wang *et al.* 2018). Görsel algılama günümüzde video kaydı yapan kameralar veya hareket ile tetiklenen foto-kapan cihazları ile uygulanmaktadır. Foto-kapan cihazı doğal ortamda canlı yaşamı

hakkında görüntü toplanmasını sağlayan cihazlardır (Kays *et al.* 2010). Hareket ile tetiklenen bu cihazlar belirli bir zamanda belirli sayıda görüntü kaydetmekte ve GSM teknolojisi üzerinden bu görüntüleri çoklu ortam mesajı veya e-posta olarak gönderebilmektedir (Kays *et al.* 2010).

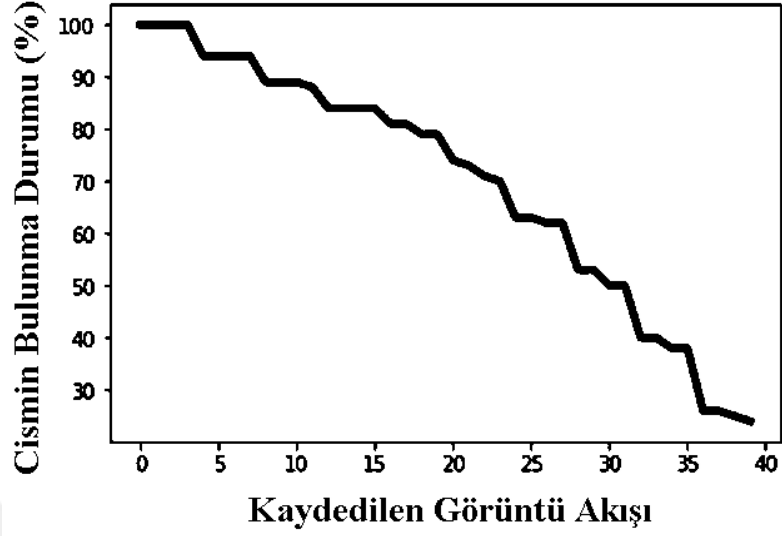


**Şekil 1.4.** Bulut ve sis/kenar bilişim ilişkisi (Bonomi *et al.* 2012).

Standart foto-kapanlar tasarımları gereği bazı kısıtlamalara sahiptir. Görüntüde cisim tespit etme ve tanıma ile ilgili en önemli kısıtları, saniyede en fazla 2 görüntü kaydedebilmesi, kızılötesi flaş kullanıldığında saniyede 1 görüntü kaydedebilmesi, her tetiklemede ayarlanabilir 1 ile 100 arası görüntü kaydedebilmesi ve 0-60 saniye arasında görüntü kaydetme hızına sahip olmasıdır (Rovero *et al.* 2013). Bu kısıtlar düşünüldüğünde hareketli cisme ait yeterli sayı ve pozda görüntü kaydedilmemektedir. Buna bağlı olarak bilgisayarlı görü ve görüntü işleme teknikleri ile görüntüler üzerinde yapılacak çalışmalar için etkili sonuçların alınabileceği yeterli veri elde edilememektedir. Yatay ekseninde görüntü akışının, dikey ekseninde ise görüntülerde cisim bulunma durumunun gösterildiği Şekil 1.5'ten anlaşıldığı üzere, foto-kapanlar ile elde edilen veri setleri incelendiğinde, görüntü sayısı arttıkça, görüntülerde cisim bulunma durumunun azaldığı anlaşılmaktadır. Foto-kapan çalışmalarında kullanılan parametrelerden dolayı kaydedilen görüntülerinin önemli bir kısmının hareketli cisim içermediği görülmektedir (Norouzzadeh *et al.* 2018). Bu duruma tetiklemenin arka plan

hareketinden kaynaklanması veya her tetiklemede belirli sayıda görüntü kaydedilmesi sebep olduğu anlaşılmaktadır. Elde edilen bu görüntülerin gsm protokolü üzerinden e-posta ile gönderilebilmesi için büyük maliyet gerektirmekte ve foto-kapan belleğinin verimsiz kullanımı problemi ortaya çıkmaktadır. Ayrıca foto-kapanlar sabit görüntü kaydetme frekansına sahip olmasından dolayı hareketli cisim, yeterince görüntü alınamadan kamera açısından çıkabilmektedir. Bu gibi problemler foto-kapanın bellek ve güç kaynağının etkili kullanılamamasına neden olmaktadır. Foto-kapanlar ile cisim tespiti yapılan çalışmalarda, kaydedilen görüntülerdeki cisimler insanlar tarafından el ile tespit edilmektedir (Buehler *et al.* 2019; Wang, 2014). Bu çalışmalarda, farklı türdeki cisimlerin görüntülerinin tespit edilmesi, kırılması ve kaydedilmesi zor, zahmetli, zaman alan bir süreçtir ve çok fazla iş yükü gerektirmektedir (Buehler *et al.* 2019; Nguyen *et al.* 2017; Norouzzadeh *et al.* 2018). Ayrıca cisim tespit işleminin performansı, elle yapılan bu tespit etme ve kırma işleminin miktarı ve başarısı ile sınırlanmaktadır. El ile yapılan tespit işleminde veri setinin küçük bir kısmı incelenmekte, tespit edilen türlere ait yetersiz model görüntü kaydedilmekte, bu nedenle bazı cisim türleri tespit edilememektedir (Buehler *et al.* 2019). Bu gibi problemler, bellek ve güç kaynağının etkili kullanılamamasına, daha fazla insan iş gücüne ihtiyaç duyulmasına, daha az ve faydasız verinin elde edilmesine ve bu nedenle yapılan çalışmalardan daha az başarı elde edilmesine neden olmaktadır (Norouzzadeh *et al.* 2018). Foto-kapanlar ile ilgili yapılan anket çalışmasında (Glover *et al.* 2019), önümüzdeki 10-20 yıl içerisinde foto-kapanlarda olması istenen en önemli üçüncü özellik cisim içermeyen görüntülerin elenebilmesi olarak açıklanmış olup, foto-kapanlarda bu dönemle olacağı tahmin edilen ilk işlev ise otomatik cisim tespittir (Glover *et al.* 2019).





**Şekil 1.5.** Foto-kapan görüntülerinde cisim bulunma durumu

Görüntü ve videolarda cisim tespit etme ve tanıma problemleri birçok alanda yaygın olarak çalışılmaktadır (Jain *et al.* 1995). Bu problemler akıllı araba, akıllı şehir, akıllı ev ve güvenlik gibi birçok nesnelerin interneti uygulamasında araç, yaya, eşya, plaka gibi cisimlerin tespiti ve takibi amacıyla çözülmeye çalışılmaktadır (Swan, 2012). Bu problemler genel olarak hareketli cisim tespiti ve örüntü tanıma tabanlı cisim tespit etme olarak ikiye ayrılmaktadır (Jain *et al.* 1995). Hareketli cisim tespiti, sabit kameralardan alınan görüntülerde arka plan modelini veya görüntülerin farklarını kullanarak yapılmaktadır (Cucchiara *et al.* 2003). Literatürde bu amaçla arka plan çıkarma, çerçeve farkı ve optik akış yöntemleri bulunmaktadır. Örüntü tanıma tabanlı yöntemlerde ise cisimlere ait görüntülerin şekil, doku, kenar, köşe veya diğer öznitelik dönüşümü yöntemleri kullanılarak kayan pencereler ile hedef görüntüde tespit edilmesine dayanmaktadır (Lowe, 1999). Literatürde kullanılan öznitelik dönüşüm yöntemleri HOG (Histogram of Oriented Gradients), Renk Histogramı, LBP (Local Binary Pattern) ve öznitelik dönüşümleri olan SIFT (Scale-Invariant Feature Transform), SURF (Speeded-Up Robust Features), BRIEF (Binary Robust Independent Elementary Features) ve ORB (Oriented FAST and Rotated BRIEF) olarak sıralanabilmektedir. Bu yöntemlerin hesaplanma zamanı, başarı oranı ve hassasiyet anlamında birbirlerinden avantaj ve dezavantajları bulunmaktadır (Hanbay and Hüseyin 2017). Ayrıca örüntü tanıma tabanlı yöntemlerdeki aşamalar, yapay sinir ağı modellerinde kullanılarak

evrişimli sinir ağları mimarileri ile derin özniteliklerin elde edilmesi ve öznitelikli bölgeler gibi yaklaşımlarla cisimlerin tespit edilmesi de son yıllarda yaygınlaşmaktadır (Tompson *et al.* 2015).

Yukarıda bahsedilen motivasyon doğrultusunda bu tezde, nesnelerin interneti için kenar bilişim tabanlı cisim tespit etme ve tanıma problemlerinin çözümünde kullanılacak yöntemler analiz edilmiştir. Kaydedilen görüntülerde gerçekleştirilecek işlemler, farklı senaryolar şeklinde kenar veya sunucu arasında paylaştırılmış ve kullanılan yöntemlerin bellek, işlem ve iletişim gereksinimleri, doğruluk performansları araştırılmıştır. Çalışmada hareketli cisim tespiti için literatürde yaygın olarak kullanılan arka plan çıkarma ve çerçeve farkı yöntemleri uygulanmıştır. Öznitelik tabanlı cisim tespiti için öznitelik eşleştirme yöntemi ile birlikte öznitelik dönüşüm yöntemlerinden SIFT, SURF, BRIEF ve ORB kullanılmıştır. Derin öğrenme tabanlı cisim tespiti için gerçek zamanlı uygulamalarda kullanılan YOLO mimarisi uygulanmıştır. Cisim tanıma için ise HOG, Renk Histogramı, Yerel İkili Örüntü, SIFT, SURF öznitelik dönüşümü yöntemleri, k en yakın komşu, destek vektör makineleri ve rastgele orman örüntü tanıma yöntemleri ile birlikte uygulanmıştır.

Çalışmada senaryoların uygulanabilmesi amacıyla görsel algılama için kullanılan foto-kapan cihazının fonksiyonları temel alınarak, bilgisayarlı görme, görüntü işleme ve örüntü tanıma işlevleri bulunan bir akıllı nesne tasarımı gerçekleştirilmiştir. Tasarlanan akıllı nesne kullanılarak Atatürk Üniversitesi'nde iç ortamdan ve dış ortamdan hareketli cisimlere ait görüntü kaydının yapıldığı foto-kapan uygulamaları gerçekleştirilmiştir. Kaydedilen görüntüler üzerinde üniversitenin eğitim, yönetim ve güvenlik gibi faaliyetlerinde uygulanabilecek bilgisayarlı görme ve görüntü işleme tabanlı işlevleri içeren kenar bilişim tabanlı veri işleme mimarileri incelenmiştir. Uygulanan veri işleme mimarileri, gerçekleştirilecek akıllı şehir, akıllı kampüs, akıllı derslik gibi projeler için üniversite, belediye, askeriye gibi kurumların dijital dönüşüm uygulamalarına katkı sağlaması açısından önem arz etmektedir. Akıllı nesne, hareket algılayıcısı ile tetiklendikten sonra görüntü kaydına başlamakta olup, 1.5 saniye içerisinde hareket algılandığı sürece kayda devam etmektedir. Akıllı nesne ile görüntüler gerçek zamanlı

olarak sunucuya TCP üzerinden iletilebilmekte, görüntüler üzerinde otomatik hareketli cisim tespiti yapılabilen ve bilgisayarlı görü ve görüntü işlemeyle ilgili yöntem ve işlemler uygulanabilmektedir.

Yapılan çalışmalarda uygulanan senaryolar ile sunucuların işlem ve iletişim ihtiyaçları büyük miktarda azaldığı anlaşılmaktadır. Ancak sunucuya iletilen veri miktarı azaldıkça cisimlere ait temsiliyetin azaldığı, iletilen verinin yeniden kullanılabilirliğinin veya farklı işlem yapılabilirliğinin olmadığı sonucu elde edilmiştir. Yapılan uygulamalarda görüntülerdeki cisimlere ait parçaların sunucuya iletildiği üç numaralı senaryoda, işlem, bellek ve veri iletişim gereksinimlerinin büyük oranda azalması ve cisimlere ait temsiliyetin korunması nedeniyle en etkili senaryo olduğu anlaşılmaktadır.

Çalışma kapsamında hareketli cisimlerin görüntüdeki konum değişimlerine dayanan görüntü seçme yöntemi önerilmiştir. Yöntem, kamera önünden geçen cisimlerin farklı konumlarda farklı pozlarını elde etme amacı taşımaktadır. Görüntü kaydetme işlemi harekete dayalı olduğu için, görüntü kaydı hareketle birlikte sonlandırılabilen ve cisim içermeyen görüntülerin kaydedilmeyerek gereksiz bellek kullanımının önüne geçilebilmektedir.

Çalışmada hareketli cisim tespitinde karşılaşılan problemlerin çözümü amacıyla hareketli cisim tespitinin iyileştirilmesi için kullanılacak görüntü tanıma tabanlı algoritma önerilmiştir. Yöntem, karmaşık ve hareketli arka plan problemlerinden kaynaklanan sorunları çözmek amacıyla, elde edilen çıktının sınıflandırılmasına dayanmaktadır. Yapılan çalışmalarda foto-kapan görüntülerinde hareketli cisim tespitinin %12-%15 arasında iyileştirilmesi sağlanmıştır.

Hareketli cisim tespitinin minimum bellek ve işlem kullanılarak gerçekleştirilmesi ve hareketli cisimlere ait bütünsel veri elde edilebilmesi amacıyla üç çerçeve farklı yöntemini temel alan yeni bir yaklaşım önerilmiştir. Yaklaşım, üç çerçeve farklı yöntemine herhangi bir yöntem ve işlem kullanılmasına gerek kalmadan, yukarıda bahsedilen görüntü seçme işlemi uygulanmasına dayanmaktadır.

## 2. KAYNAK ÖZETLERİ

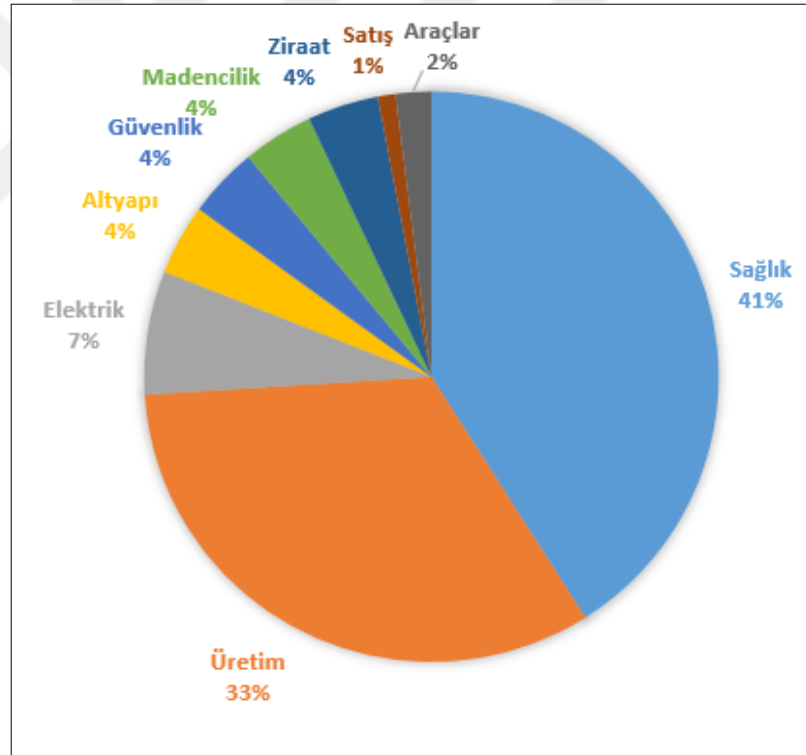
### 2.1. Nesnelerin İnterneti

#### 2.1.1. Tanım ve kapsam

Nesnelerin interneti kablosuz iletişim teknolojilerinde hızla gelişen bir paradigmadır. Bu kavramın temel fikri, etrafımızdaki benzersiz adres şemaları aracılığıyla birbirleriyle etkileşime girebilen ve ortak hedeflere ulaşmak için komşularıyla işbirliği yapabilen çeşitli nesnelerin etrafımızdaki yaygın varlığıdır. Nesnelerin interneti temel gücünü, günlük yaşam ve davranışlarımızı yüksek seviyede etkileyebilecek bir fikir olmasından almaktadır (Atzori *et al.* 2010). Nesnelerin interneti ağını, verilen görevleri yerine getirebilmek için internete üzerinden diğer cihazlarla iletişim kurabilen, eşsiz kimlik, algılama, ağ iletişimi ve işlem yapma kapasitesi olan, günlük hayatta kullanılan nesnelere oluşturmaktadır. Web 2.0'dan sonra Web 3.0 veya anlamsal web olarak adlandırılan, web içeriklerinin yazılımlar tarafından anlam üzere akıllı sistemler geliştirilmiştir. Bu akıllı davranışa sahip sistemler ile algılayıcı ağları ve internetin birleşmesi ile nesnelerin interneti meydana gelmiştir (Whitmore *et al.* 2015).

Nesnelerin interneti, fiziksel nesnelerin görmesini, duymasını, verilen görevleri yerine getirmesini, birbirleri ile iletişime geçerek bilgi paylaşmasını ve karar vermesini sağlamaktadır (Al-Fuqaha *et al.* 2015). İnternet ile birlikte birbirleri ile iletişim kurabilen istemciler, sunucular, yönlendiriciler, zamanla bunlara entegre olan masaüstü ve dizüstü bilgisayarlar, tabletler, akıllı telefonlar geliştirilmiştir. Nesnelerin interneti ise, bu ağa gündelik aygıtlara ait teknolojilerin ses ve görüntü alıcıları, hareket, nem, sıcaklık ve duman detektörü, ev aletlerini bağlamak ve başlangıçta bu şekilde tasarlanmasalar bile onları çevrim içi hale getirebilmektir. Nesnelerin internetinin getirdiği değişim ilk aşamada bu cihazların internete entegre olması ve internet üzerinden erişilebilir olmasıdır (Whitmore *et al.* 2015).

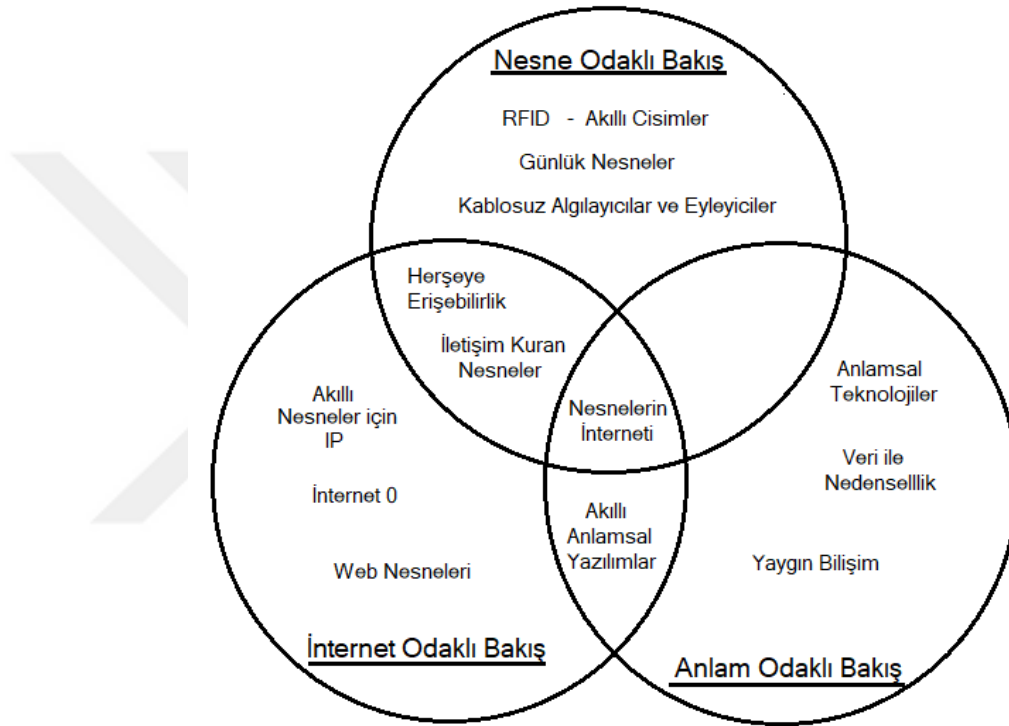
Nesnelerin interneti Şekil 2.1’de görüldüğü üzere insan hayatının hemen hemen her alanında uygulamalara sahiptir. Gömülü sistem, iletişim maliyetlerinin azalması ile yaygınlaşması hızlanan nesnelerin interneti çalışmaları, yapay zekâ teknolojileri desteklendikçe insan hayatının merkezinde yerini almakta ve artık insan hayatını en çok etkileyen teknoloji haline gelmektedir. Literatüre göre nesnelerin interneti bu nesnelerin yaygın bilişim, gömülü sistemler, iletişim teknolojileri, algılayıcı ağları, internet protokolleri ile destekleyerek akıllı nesne olmasını sağlamaktadır. Nesnelerin internetinin insan hayatında önemli rol oynayabileceği acil durum tespiti, taşımacılık, sağlık, endüstri otomasyonu gibi alanlarda kullanılacağına kesin gözüyle bakılmaktadır (Al-Fuqaha *et al.* 2015).



**Şekil 2.1.** Nesnelerin interneti projelerinin pazar payları (Al-Fuqaha *et al.* 2015)

Nesnelerin interneti bakış açılarına göre Şekil 2.2’de görüldüğü üzere internet, nesne ve anlamsal olarak 3 farklı şekilde ele alınmaktadır. Nesne odaklı bakışta kablosuz algılayıcı ağlarından etkilenerek basit bir ifadeyle RFID vb. iletişim teknolojisi içeren

kablosuz nesnelere olarak tanımlanmaktadır. İnternet odaklı bakışta IP'si bulunan internet üzerinden erişilebilen web nesnelere olarak tanımlanmaktadır. Anlamsal olarak ise veri ile anlam ve nedensellik elde edebilen, iletişimden çok elde edilen verilerin anlamlandırılması ile ilgilenen nesnelere olarak tanımlanmaktadır. Literatüre göre ise nesnelere interneti ise bu üç bakışın kesişimi olarak ifade edilmektedir (Atzori *et al.* 2010).



**Şekil 2.2.** Bakış açılarına göre nesnelere interneti (Atzori *et al.* 2010)

Nesnelere interneti yaşam döngüsü boyunca kullanabileceği yararlı bilgileri, akıllı nesnelere vasıtasıyla fiziksel dünyadan elde etmektedir. Nesnelere interneti elde ettiği bu bilgileri kullanarak sağlık, akıllı ev, bina otomasyonu, enerji verimliliği, endüstriyle otomasyon, akıllı ölçüm, akıllı şebeke altyapısı, çevresel izleme ve tahmin, varlık yönetimi, lojistik, araç otomasyonu, akıllı ulaşım, akıllı tarım, akıllı alışveriş uygulamalarında önemli rol oynamaktadır (Mainetti *et al.* 2011). Nesnelere interneti kapsamına göre, gerçek olan her şey sanal hale gelebilmekte, herkes ve her şey internette bulunabilir, adreslenebilir, okunabilir bir karşılığı olmaktadır (Atzori *et al.* 2014). Nesnelere İnternetinin en büyük itici gücü, birçok farklı işlev türü için mevcut

olan düşük maliyetli algılayıcıların olmasıdır. Standart algılayıcıların bazıları hareket, ses, ışık, elektrik potansiyeli, sıcaklık, nem, konum, kalp atış hızı ve kalp atış hızı değişkenliği gibi algılama işlevleri içerebilmektedir (Swan, 2012). Buna ek olarak akıllı nesnelerin gelişmesi ile nesneler yeni hizmetleri keşfetme, yeni tanıdık cihazlar edinme, bilgi alışverişi, dış hizmetlere bağlanma, diğer nesnelerin işlevlerini kullanma, ortak bir hedef doğrultusunda iş birliği yapma gibi işlemlere de sahip olabilecektir (Atzori *et al.* 2014). Bu, karmaşık hizmetleri gerçekleştirmek, işbirliği yapan nesneler arasındaki yoğun etkileşimler sayesinde mümkündür. Bu, yeni hizmetleri keşfedebilecek, yeni tanıdıklar başlatabilecek, bilgi alışverişinde bulunabilecek, harici hizmetlere bağlanabilecek, diğer nesnelerin yeteneklerinden yararlanabilecek ve ortak bir hedef için işbirliği yapabilecek yeni nesil “akıllı nesneler” tasarımını getirmektedir (Atzori *et al.* 2014).

Gömülü sistemler, programlama ve ağ sistemleri alanlarındaki ilerlemeler nesnelerin interneti ile birlikte, milyonlarca farklı tür cihazdan oluşan bir altyapıya öncülük etmektedir. Nesnelerin interneti vizyonuna göre bu cihazlar sadece bilgi aktarmak yerine, bilgi işleme, birebir iletişim ve gelişmiş işbirlikleri kurabileceklerdir. Bu nesnelerin interneti altyapısı çevreye sağlam bir şekilde bütünleşmiş olması gerekmektedir (Yun and Yuxin 2010). Nesnelerin internetinin kapsamlı algılama, güvenilir veri aktarımı ve akıllı işlem yapabilme gibi 3 önemli özelliğe sahip olması gerekmektedir (Yun and Yuxin 2010). Akıllı nesnelerin temel özellikleri şunlardır (Atzori *et al.* 2014):

- Dış sistemlerle birlikte çalışabilme,
- Sosyal ağlarla iletişim,
- Çevre bilinci,
- Çevre ile etkileşim,
- Komşularla söz ve sosyal davranış,
- Kendi sosyal ağlarını kurma,
- İşbirliği ve yardımlaşma ile karmaşık hizmet sunmak için kendi ağlarını kurma.

Nesnelerin interneti, internetin geleceği olarak kabul edilmektedir ve yakın zamanda milyarlarca akıllı nesne içerecek hale gelecektir. Nesneler insan müdahalesi olmadan iletişim kurabilmektedir. Akıllı algılayıcılar, düşük güç tüketen kablosuz iletişim, algılayıcı ağ teknolojilerinin avantajları çok sayıda nesnenin ağ olarak bağlanmasını sağlamaktadır. Nesnelerin internetindeki kritik gereklilik ise ağdaki nesnelerin birbiri ile bağlantılı olmasıdır (Li *et al.* 2015).

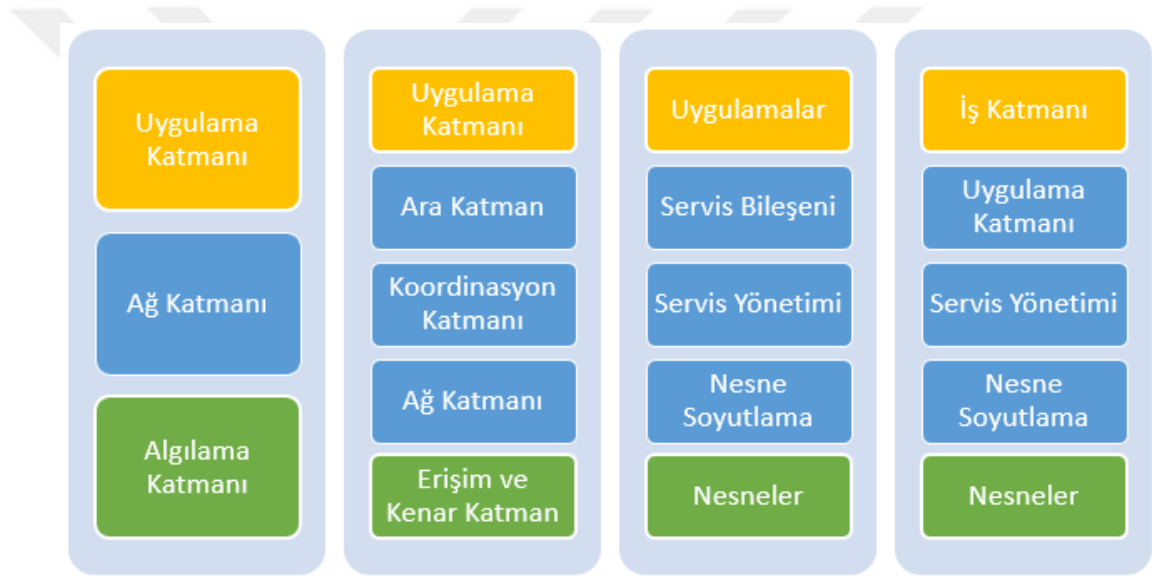
Nesnelerin internetinde gerekli olan kablosuz algılayıcı ağları, IPv6 gibi protokoller desteklenerek ip tabanlı algılayıcı ağları haline gelebilmektedir. Kablosuz algılayıcı ağlarının internet ile birleşmesi nesnelerin internetinde akıllı nesne uygulamalarını mümkün hale getirmektedir. Gelecekte birçok kablolu veya kablosuz heterojen yapıyı birleştirecek ve nesnelerin internetini meydana getirecektir. Algılayıcı teknolojilerinin düşük maliyetli olması, kablosuz algılayıcı ağlarının çoğalmasını, çevre izleme, tarım, sağlık hizmetleri, akıllı binalar gibi pek çok uygulama yapılmasını sağlamıştır. Kapalı veya tescilli sistemlere dayanan algılayıcı ağları dış dünya ile sınırlı iletişim sahiptir. Kablosuz algılayıcı ağları verilerini internete iletmek için bluetooth, zigbee gibi özel ağ teknolojileri kullanmaktadırlar. Özel dönüşümler uygulanmadıkça, standartlar arasında doğrudan dönüşüm bulunmamaktadır (Mainetti *et al.* 2011).

### **2.1.2. Katmanlar ve elementler**

Nesnelerin interneti uygulamalarında 3 teknoloji üzerinde durulmaktadır. Bunlar tanımlama(kimlik), algılama ve iletişimdir. Tanımlama için kullanılabilecek RFID veya IP sistemleri eşsiz kimlik, görüş alanı kısıtı olmadan gerçek zamanlı izlenebilme, etiketlenebilme, tanınabilme imkânı sağlamaktadır. Algılama alanında algılayıcılar ve algılayıcı ağları kullanılabilmektedir. RFID ile işbirliği yapabilen ağlar, ortamdan konum, sıcaklık, durum, hareket gibi bilgileri alarak fiziksel dünya ile dijital dünya arasında köprü işlevi görebilmektedir (Atzori *et al.* 2010). Nesnelerin internetinin mimari yapısı temel olarak 3 katmandan oluşmaktadır, Şekil 2.3'te görüldüğü üzere farklı bakış açılarına göre farklı ara katmanlar da önerilmektedir:

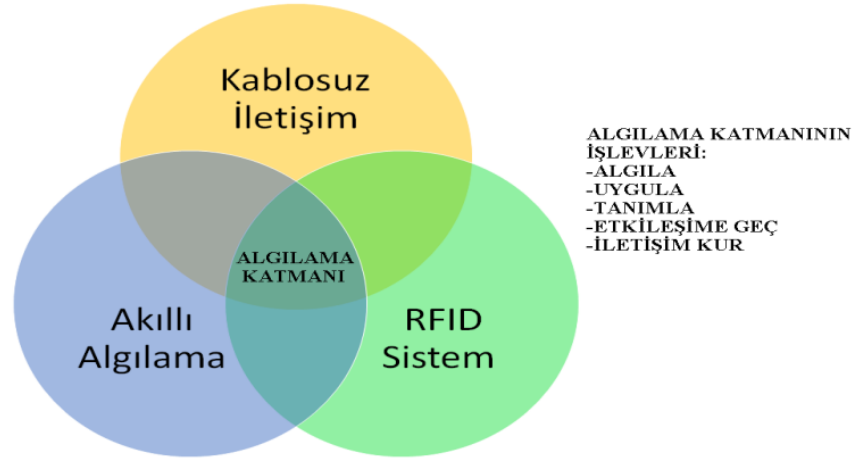


- Algılama Katmanı: RFID, kamera, GPS, algılayıcı çeşitleri, algılayıcı ağları, M2M terminalleri, bilgi toplamak için kullanılacak her şey bu katmanın bileşenidir.
- Ağ Katmanı: İnternete bağlanabilen her türlü iletişim teknolojisini ifade eden katmandır. Ağ katmanı, nesnelerin internetinin evrensel hizmetinin alt yapısını oluşturmaktadır.
- Uygulama Katmanı: Nesnelerin interneti ve ticari uzmanlığın birleşmesi ile akıllı uygulamaların gerçekleştirilmesi, hizmetlerin kullanıcılara ulaştırılması gibi görevler bu katman ile gerçekleştirilmektedir (Yun and Yuxin 2010).



**Şekil 2.3.** Literatürde önerilen nesnelerin interneti katmanları (Al-Fuqaha *et al.* 2015)

Nesnelerin internetinde hali hazırda IPv4 ve IPv6 standartlarında internet iletişim protokollerinden TCP ve UDP üzerinden veri aktarma imkânı bulunmaktadır. Nesnelerin internetinde iletişim protokolü için standart belirleme çalışmaları devam etmektedir (Al-Fuqaha *et al.* 2015). Şekil 2.4'te görüldüğü üzere, algılama katmanı kablosuz iletişim, akıllı algılama ve tanımlama sistemi olan RFID'nin birleşimi olarak ifade edilmektedir.



**Şekil 2.4.** Algılama katmanının işlevleri (Li *et al.* 2015)

Bir internet nesnesinde enerji verimliliği, gerçek zamanlı veri elde etmek için her zaman aktif olması gerekliliği ve buna göre yüksek enerji verimliliği ile daha uzun süre çalışabilmesi özelliklerinin olması gerekmektedir. İletişim katmanında IPv4 ve IPv6 kullanılabileceği gibi, 3G, 4G, Wifi, ZigBee, Wimax gibi ağlar da kullanılabilmektedir (Li *et al.* 2015). Çizelge 2.1’de gösterildiği üzere, kullanılan iletişim teknolojisi veri iletim hızını, frekansını ve veri iletim mesafesini etkilemektedir. Kablosuz algılayıcı ağlarında sıklıkla tercih edilen iletişim teknolojileri, diğer kablosuz iletişim sistemleri ile kıyaslandığında yaygın olmaması, kullanımı uygun olmayan standartlara sahip olması, iletim hızının az olması, aktarım mesafesinin yetersizliği gibi dezavantajları bulunmaktadır. Mevcut eğilim kablosuz algılayıcı ağları ile internet arasında yerel bağlantı sağlanması için wi-fi ağları internet protokolünü(IP) kullanmaktır (Vasseur and Dunkels 2010). Bu şekilde akıllı nesnelere, benzersiz ip adresine sahip olan ve esas olarak açık standartlara dayanarak bir işlevi yerine getirmek için birbirine bağlanabilen nesnelere tanınabilmektedir.

IP’siz çözüm olarak Zigbee, Z-wave, insteon, wavenis örnek gösterilebilirken, IP’li çözümlerde, 32 bit mikro denetleyicilerin performans ilerlemeleri ve yüksek düzeyde optimize edilmiş protokol yığını uygulamalarının kullanılabilirliği, akıllı nesnelere IP bağlantısı eklemenin mümkün olmasını sağlamaktadır. Yakın zamanda 50 milyar nesnesinin internete bağlanacağı öngörüsünde IPv4 protokolünün yetersiz kalacağı ve

ancak IPv6 ile bu kadar nesnenin adreslenebileceği ifade edilmektedir. IPv6 sayesinde her akıllı nesne gerçek ip adresi kullanarak NAT (Network Address Translation), Proxy (Vekil) veya diğer arabirimlere ihtiyaç duymadan IP tabanlı ağlara kolayca bağlanabilecektir (Mainetti *et al.* 2011).

**Çizelge 2.1.** İletişim protokollerinin özellikleri (Li *et al.* 2015)

İletişim Protokolü	Veri Aktarım Hızı	Frekans	Aktarım Mesafesi
RFID (Mikro Dalga)	424 kbps	2.4 Ghz	15 m
ZigBee 3.0	250 kbps	2.4 GHz	10- 100 m
Bluetooth 5.0	2 Mbps	2.4 GHz	50 m
WiFi	450-1300 Mbps	2.4/5.0 GHz	100 m
Wi-Max	70 Mbps	2-11 GHz	50 km

İnternet nesnelere gömülü sistem olarak yeni bir akım başlatan Arduino mikro denetleyicileri, daha gelişmiş özellikler sunan Raspberry Pi ve benzeri donanım ürünleri tercih edilmektedir. Hızla gelişen bu ürünler birçok algılayıcı ve iletişim modülü ile uyumlu çalışabilmektedir. Bu ürünlerden Arduino küçük çaplı uygulamalar için uygun olmaktadır. Raspberry Pi ve benzeri ürünlerin son sürümlerinde Linux işletim sistemleri, dâhili wi-fi, sd kart, Ethernet, usb bağlantı gibi imkânları barındırması nedeniyle nesnelere interneti çalışmalarında sıklıkla tercih edilmektedir. Ayrıca Raspberry pi ve benzeri ürünlerde entegre kamera bulunmakta ve görüntü tabanlı uygulamalarda büyük kolaylık sağlamaktadır.

### 2.1.3. Gelecek vizyonu

Nesnelere internetinin gelecek vizyonunu düşük güç tüketimi, düşük maliyet, işlev çeşitliliği, kolay programlanma gibi özellikler ile, insandan bağımsız karmaşık görevleri iş birliği ile yerine getirebilecek standartlar oluşturmaktadır. Günümüzde düşük maliyetli algılayıcıların ve mikro denetleyicilerin kullanımı yaygınlaşmakta, gömülü sistemlerde programlama kolaylığı ve işlev geliştirme imkânları artmaktadır. Bu

imkanlar nesnelerin internetinin gelecek vizyonunda yapay zekâ işlemleri gibi zor görevleri yerine getirebilmesi ve diğer nesnelere iş birliği yapabilmesinin önünü açmaktadır. Nesnelerin interneti alanı kısa vadede günlük yaşamımıza akıllı nesnelerin girmesini sağlayacaktır. Bu akıllı nesnelerin elde edilen verileri yapay zekâ yöntemleri ile analiz edebilmesi, diğer nesnelere iş birliği yapabilmesi, uygulanmalarda bulut hizmetlerine daha az ihtiyaç duyması beklenmektedir. Ayrıca yapay zekâ teknolojilerini dağıtık olarak iş birliği içinde yürütebilecek mimarilerin geliştirilmesi için çalışmalar yapılmaktadır.

Bulut sistemlerin dezavantajları, kenar bilişim tabanlı çözümlerin faydaları ve açık kaynaklı yazılımların yaygınlaşması, sektörleri nesnelerin interneti uygulamalarında kenar bilişim tabanlı çözümler kullanmaya yöneltmektedir. Sunucuların işlem ve bellek ihtiyacının azalması ve veri trafiğinin azalması için en uygun çözüm, nesnelere elde edilen verilerin bulut sistemlerde veya sunucularda işlenmesi yerine, kenarda veya nesnelere işlenmesi ve nesnelere ihtiyaç duyulan bilgilerin nesnelerin belleklerinde depolanması olarak görülmektedir (Lin *et al.* 2017). Bu sayede, akıllı nesne uygulamalarının yaygınlaşması ve uygulamaların etkisinin artması mümkün olabilecektir. Bu vizyona göre akıllı nesnelerin, cisim tanıma, doğal dil işleme ve akıllı kontrol gibi zor uygulamaları gerçekleştirebilecek yüksek fonksiyonel işlevler içeren cihazlar olması beklenmektedir (Wang *et al.* 2018).

Yerel ve belirli problemler için uygulanabilecek kenar bilişim tabanlı çözümlerin kullanıldığı nesnelerin, geliştirilecek standartlar ile akıllı algılama, iş birliği, kaynak paylaşımı, kendi servislerini genel ağa sunma gibi gelişmiş özelliklere sahip olması ve bu hizmetleri insandan bağımsız olarak, özerk bir altyapı ile gerçekleştirmesi gerekmektedir (Atzori *et al.* 2010; Lin *et al.* 2017; Vermesan and Friess 2013; Yi *et al.* 2015).

Bulut bilişim, sınırlı bellek ve hesaplama imkânına sahip olan akıllı cihazlar ve uygulamalar için esnek kaynak sağlayabilecek uygun bir hesaplama paradigması kabul edilmektedir. Akıllı nesnelerin problemlerin çözümü için diğer nesnelere veya sunucular

ile koordinasyon kurarak iş birliği yapması, hem kenar sistemlerin en iyi performans ile kullanılması, hem de bulut sistemlerin büyük veri tabanları barındırma gibi avantajlarının kullanılmasını sağlamaktadır. Nesnelerin interneti uygulamaları, bulut bilişim ile güçlendirilmesine rağmen, hareket desteği, coğrafi dağıtım, konum duyarlılığı, yüksek gecikme (latency) gibi eksiklikler içermektedir. Bu nedenle hesaplamının bir kısmının ağına uç kısmında yapıldığı, internetin geleceği için yeni uygulama ve hizmetler sağlayabilen kenar bilişimin kullanılması önerilmektedir (Bonomi *et al.* 2012). Sis bilişim tanım olarak, çok sayıda heterojen ve yaygın olan cihazların iletişim kurması, potansiyel olarak bunların arasında iş birliği yapması ve üçüncü tarafların müdahalesi olmadan depolama, işlem yapma görevlerini yerine getirmesi için kullanılan senaryodur. Kenar bilişim ise verilerin kaynağına olabildiğince yakın olarak işlendiği bir mimaridir. Sis ağı internetin uç kısmında bulunduğu için heterojendir. Sis ağının görevi içerdiği her bir bileşeni bağlamaktır. Büyük ölçekli nesnelerin interneti için bu ağı yönetmek, bağlantısını sürdürmek, hizmet sağlamak kolay olmamaktadır (Yi *et al.* 2015).

Sis programlama 3 farklı senaryoda ifade edilmektedir. Bunlar kablosuz algılayıcı ve eyleyici ağları, bağlı araç ve akıllı şebekedir. Nesneden nesneye, nesneden erişim noktasına, erişim noktasından erişim noktasına olmak üzere farklı iletişim türleri bulunmaktadır. Nesnelerin bu tür etkileşimlerle verdiği hizmetlerin kalitesinin ve çeşidinin artırılması amaçlanmaktadır. Akıllı şebeke, sis düğümleri yerleştirilmesinin yanı sıra, düşük gecikme süresi, içerik farkındalığı sağlamaktadır. Bulut ise küresel merkezileştirme sağlamaktadır. Büyük ve analitik uygulamalar için ikisi de gerekmektedir. Sis ağı cihaz ve algılayıcılardan gelen veriyi toplamakta, işlemekte ve işlenmesi için üst katmanlara iletmektedir. Kablosuz algılayıcı ve eyleyici ağları ise, düğümlerin pil ömrünü uzatmak veya enerji hasadını mümkün kılmak için son derece düşük güçte çalışmak üzere tasarlanmaktadır. Bu ağların çevreyi algılamak, basit işlemleri yapmak, verileri statik ağa göndermek gibi görevleri bulunmaktadır. Bu ağlar düşük bant genişliği, düşük enerji, bellek ve işlem gücü gerektirmektedir (Bonomi *et al.* 2012).

Uzun vadede nesnelerin internetinin eğilimi algılama ve internetin birleşimidir. Ağa bağlanılan tüm nesnelerin esnek, akıllı ve gerekli hizmetleri sağlayabilecek özerk bir yapıda olması beklenmektedir (Li *et al.* 2015). Tek algılayıcı veya eyleyiciye aktarılan veri sınırlı olsa da, çok sayıda nesne ve sık etkileşimleri nedeniyle toplam veri miktarı çok büyük olabilmektedir (Mainetti *et al.* 2011). Büyük verilerin işlenmesinin nasıl ele alınacağı gelecek vizyonu için önemli zorluklardandır (Zorzi *et al.* 2010). Nesnelerin interneti ile daha önce elektronik gibi görünmeyen eşyalar, gömülü sistemler ve mikro denetleyiciler ile internete bağlanarak çevrimiçi olacak ve birbirleri ile iletişim kurabileceklerdir. Günümüzde üretilen nesnelerin %5'inin dijital öncelikli olarak piyasaya sunulduğu düşünülmektedir (Swan, 2012).

İnternet ve yakın alan iletişimi, gerçek zamanlı lokalizasyon ve gömülü algılayıcılar gibi yeni teknolojilerin birleşimi, günlük nesnelere çevrelerini anlayabilen ve tepki verebilen akıllı nesnelere dönüştürülmesini sağlamaktadır. Bu nesnelere nesnelerin interneti için yapı taşlarıdır ve yeni bilgi işlem uygulamalarını mümkün kılmaktadır (Kortuem *et al.* 2009).

Akıllı nesnelere, yerel durumları anlamlandırılabilen ve insanla etkileşime imkân veren uygulamaları içermektedir. Akıllı nesnelere kendi aralarında ve dünyada neler gerçekleştiğini algılamakta, kaydetmekte ve diğer nesnelere paylaşmaktadır (Kortuem *et al.* 2009). Bu açıklama doğrultusunda akıllı nesne topolojisi aşağıdaki şekilde ifade edilmektedir:

- Farkındalık: Fiziksel dünyadaki olayları anlama, yorumlama ve tepki verme.
- Temsiliyet: Uygulama ve programlama modeli.
- Etkileşim: Nesne ile kullanıcı arasında giriş, çıkış, kontrol ve geri besleme için iletişim kurma yeteneği (Kortuem *et al.* 2009).

Ayrı ayrı çalışan akıllı nesnelere, bilgi hizmetleri için iyi fırsatlar sunmaktadır. Ancak akıllı nesnenin gücü birden fazla nesnenin ilgili yeteneklerini bağlamak için iş birliği yapmasıdır. Akıllı nesnelere dağıtık mimariye sahiptir. Nesnelere, nesnelerin internetinde

sadece teknik bir altyapı değil, insan merkezli etkileşimi olan bir nesne olarak görülmektedir. Bu nesneyi donanım ve yazılımın ötesinde sosyal yönleri de içerecek şekilde genişletilmesi gerektiği anlamına gelmektedir (Kortuem *et al.* 2009).

Akıllı cihaz olarak görülebilen foto-kapanlar, bulunduğu ortamdaki hareketi algılamakta, görüntü kaydetmekte ve duruma göre internet üzerinden gönderebilmektedir. Foto-kapan cihazları ile ilgili yapılan anket çalışmasında foto-kapanlarda önümüzdeki 10-20 yıl içerisinde olması istenen özellikler içerisinde cisim olmayan görüntülerin filtrelenmesi %54 ile üçüncü önemli konu başlığı olduğu ifade edilmektedir. Bu süreçte otomatik cisim tespitinin, foto-kapanda olması istenen bu özellikler içerisinde ilk olarak gerçekleştirileceği düşünülmektedir (Glover *et al.* 2019). Çalışmadaki anket sonuçlarına göre foto-kapanlara, hızlı tetiklenme (%66), nem direncinin artırılması (%56) ve yanlış pozitiflerin otomatik olarak filtrelenmesi (%54) gibi işlevlerin eklenmesi beklenmektedir. Yapılan çalışmada kullanıcılar foto-kapanlarda 10-20 yıl sonraki süreçlerde görüntü sınıflandırma ve analiz otomasyonu (%54) ve görüntülere uzaktan erişim sağlayacak kablosuz bağlantının (%36) olması öngörmektedirler. Çalışmada foto-kapanların problemleri, görüntü sınıflandırma (%25), hırsızlık (%18), veri depolama ve yönetimi (%17), çevresel etkenler (%12), foto-kapan konumlandırma (%10) ve pil ömrü (%5) olarak sıralanmaktadır. İlgili çalışmada foto-kapanların önümüzdeki 10-20 yılda otomatik nesne tespiti (%54), kablosuz bağlantı ile uzaktan erişim (%36), metodolojik gelişmeler (%24), daha geniş kullanım (%23), düşük maliyet (%21), daha uzun kullanım süresi (%20), görüntü kalitesi (%17), teknoloji entegrasyonu (%13), güvenilirlikte gelişme (%13), gerçek zamanlı görüntü alma (%11) ve foto-kapanların işbirliği yapması (%11) yönünde gelişeceği ifade edilmiştir. Foto-kapan çalışmalarındaki kısıtlar, foto-kapan maliyeti, hırsızlık ve güvenlik eksikliği, kamera-algılayıcı performansı, pil ömrü, görüntü kalitesi, veri analizi olarak sıralanmaktadır. Üçüncü sıradaki kısıt olan kamera-algılayıcı performansının kötü olması, yanlış pozitif ve yanlış negatif görüntülerin sayısının artmasına, pil, bellek ve görüntü analizinde zaman kaybına sebep olmaktadır. En önemli 5. kısıt olan verilerin etkili olarak yönetilmesinin, görüntü sınıflandırmasında kolaylık sağlayacağı ifade edilmiştir (Glover *et al.* 2019).

## 2.2. Nesnelerin İnternetinde Cisim Tespit Etme ve Tanıma

### 2.2.1. Tanım ve kullanım amaçları

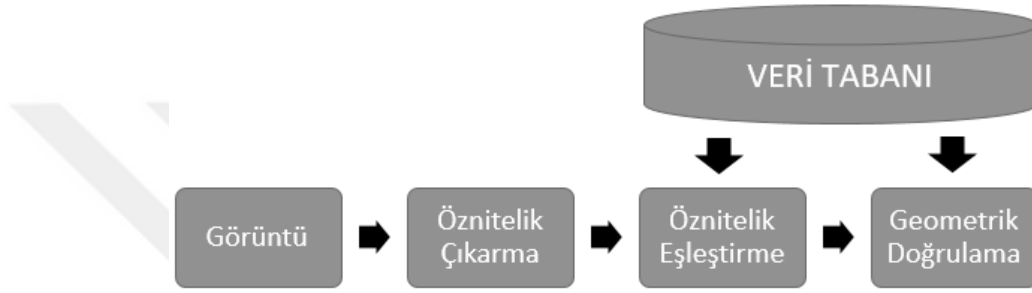
Nesnelerin internetinde cisim tanıma problemi kameralı mobil telefonların kullanılması ile “Mobil Görüntü Alma” ve “Mobil Görsel Arama” kavramları ile birlikte ortaya çıkmıştır. Bu kavramlar mobil telefonlardan elde edilmiş görüntülerin, internet üzerinden veri tabanlarındaki model görüntülerle karşılaştırılarak yapılan görsel tanıma problemini ifade etmektedir. Mobil telefonlar, yüksek çözünürlüklü kameralar, renkli ekranlar, donanım hızlandırmalı grafiklerle donatılmış video işleme cihazlarına dönüşmüştür. Bu problemde en büyük sorun, tanıma işleminin hangi kısmının mobil telefonda, hangi kısmının sunucularda yapılacağı olmuştur. Mobil cihazlarda saniyeler içerisinde görüntü öznitelikleri çıkarılabilmektedir (Girod *et al.* 2011). Verinin nerede işleneceği ile ilgili 3 farklı istemci-sunucu mimarisi bulunmaktadır :

- Görüntü nesneden sunucuya gönderilmekte, görüntü analizi tamamen sunucu üzerinde gerçekleştirilmektedir.
- Görüntü öznitelikleri nesnede çıkarılmakta, bu öznitelikler sunucuya gönderilmekte ve öznitelikler sunucuda işlenmektedir.
- Nesne, cisim tanımlama için tüm veriyi sunucudan indirmekte ve görüntü tanıma işlemini cihazda gerçekleştirilmektedir (Girod *et al.* 2011).

Bu mimarilerin veya kullanılacak bir hibrit mimarinin seçiminde, veri tabanının boyutuna, iletilecek verinin büyüklüğüne ve gecikmeye dikkat edilmelidir. Nesnede veya sunucuda oluşturulacak veri tabanı için, sınıflandırma işleminde ihtiyaç duyulan bellek, hesaplama, güç ve bant genişliği kısıtlamaları dikkate alınmalıdır. İletilen verinin az olması, gecikmenin azalması ve kullanıcı deneyiminin artması için gereklidir (Girod *et al.* 2011).



Şekil 2.5'te mimarisi gösterilen mobil tabanlı cisim tanıma problemleri, nesnelerin internetindeki cisim tespit etme ve tanıma problemine temel oluşturmaktadır. Her ne kadar mobil telefonlar da bir internet nesnesi olarak görülse de, nesnelerin internetinin bütün özelliklerini taşımamaktadır. Mobil telefonlar, bir internet nesnesi olmaktan çok, kişisel ihtiyaçlarda nesnelerin interneti hizmetlerinden faydalanan bir araç olarak ifade edilmektedir.



**Şekil 2.5.** Yerel öznitelikler ile veri tabanından görüntü alma (Girod *et al.* 2011)

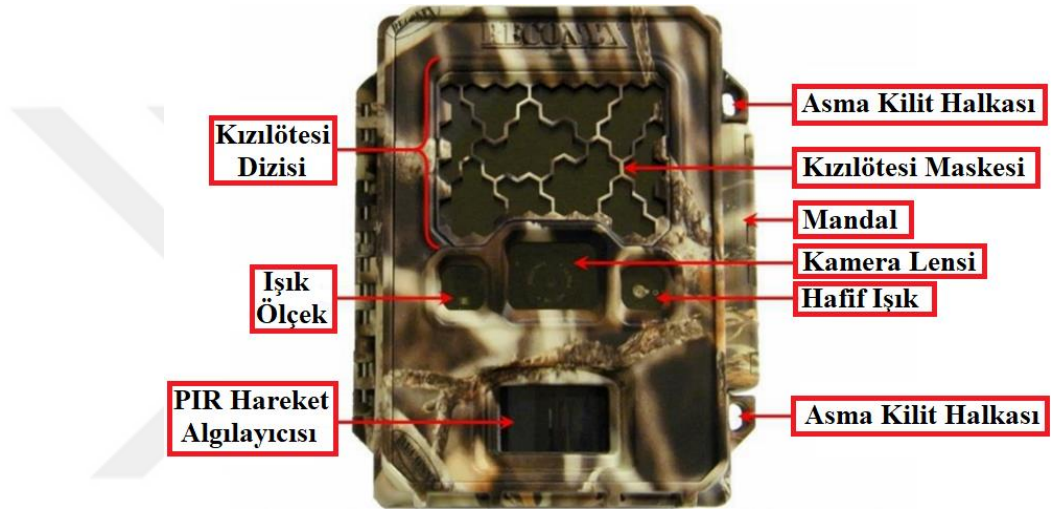
Nesnelerin internetinin gelecek vizyonu kapsamında akıllı nesnelere olması beklenen görevler, işlevler ve fonksiyonlar dikkate alındığında, çok sayıda uygulamada kullanılabilecek cisim tespit etme ve cisim tanıma işlevleri öne çıkmaktadır. Cisim tespit etme ve tanıma yöntemlerinin, bu yöntemlerden önce ve sonra gerçekleştirilecek diğer işlemlerin çok çeşitli ve fazla olmasından dolayı, çalışmanın nesnelerin interneti açısından önemli olduğu anlaşılmaktadır. Bu tezde, görüntü üzerinde cisim tespit etme ve tanıma işlemlerinin uygulandığı, nesnelerin interneti kapsamına uygun olan foto-kapan cihazları tercih edilmiştir. Foto-kapan cihazları genellikle doğal ortamdaki canlıları tespit etmek amacıyla, bunun dışında vahşi hayvan tespiti, güvenlik uygulamaları gibi birçok uygulamada görüntü kaydetmek ve bu görüntüleri iletmek amacıyla sıklıkla kullanılmaktadır. Bu tezde foto-kapan cihazları temel alınarak, akıllı nesne olabilecek bir cihaz tasarımı gerçekleştirilmiştir. Bu akıllı nesnenin cisim tespit etme ve cisim tanıma kenar programlama tabanlı uygulanabilecek çözümler ile buluttaki sunucuların iş yükü ve iletişim trafiği üzerindeki etkileri araştırılmıştır.

### 2.2.2. Foto-kapan tanımı ve kullanım amaçları

Foto-kapanlar doğal ortamda canlı yaşamı hakkında bilgi toplanmasını sağlayan, görüntü kaydı yapan cihazlardır (Kays *et al.* 2010). Şekil 2.6'da örnek bir foto-kapan görüntüsü ve foto-kapanda bulunan diğer modüller gösterilmektedir. Genellikle ormanlık bölgelere yerleştirilen foto kapanların kaydettiği görüntüler çevre bilimciler tarafından doğadaki canlı türlerini ve davranışlarını incelemek amacıyla kullanılmaktadır (Yu *et al.* 2013). Ayrıca foto-kapanlar, bölgede yaşayan canlıların tespit edilmesi, türlerinin ve nüfuslarının belirlenmesi için görüntü kaydının yanı sıra, yerleşim yerlerinde vahşi hayvanların tespit edilmesi amacıyla da kullanılmaktadır. Bu çalışmalarda, görüntülerde bulunan canlıların otomatik olarak tespit edilebilmesi için bilgisayarlı görü ve örüntü tanıma yöntemleri kullanılmaktadır. Bu yöntemler kullanılarak görüntüdeki hayvanlar ve insanlar otomatik olarak tespit edilmeye veya tanınmaya çalışılmaktadır.

Foto-kapan görüntüleri üzerinde yapılan çalışmalarda, cisim tespit etme ve cisim tanıma olmak üzere iki temel problemin çözülmesi hedeflenmektedir. Görüntüde cisim tespiti işlemi, cismin görüntüde bulunması ve eğer var ise görüntüdeki konumunun belirlenmesi olarak ifade edilmektedir (Song *et al.* 2011). Cisim tanıma ise tespit edilen cisimlerin tür veya farklı türde sınıflandırılması anlamına gelmektedir. Cisim tespiti amacıyla yapılan çalışmalarda genellikle cisimlere ait model görüntülerin öznelikleri, foto-kapan görüntülerinde kayan pencere tabanlı yöntemle aranarak cisim tespit işlemi gerçekleştirilmektedir (Andavarapu and Vatsavayi 2017; Yu *et al.* 2013). Bu ve benzeri görüntü tabanlı işlemlerin otomatik olarak gerçekleştirilmesi için literatürde bilgisayarlı görü yöntemleri kullanıldığı çalışmalar bulunmaktadır (Nguyen *et al.* 2017). Foto-kapan görüntülerinde yapılan çalışmalarda, cisim tespit etme ve tanıma işlemlerinin yapılmasını zorlaştıran karmaşık ve hareketli arka plan, ışık yoğunluğu değişimi ve cismin kısmi görünümü gibi önemli problemler bulunmaktadır (Andavarapu and Vatsavayi 2017; Meek *et al.* 2014; Nguyen *et al.* 2017; Norouzzadeh *et al.* 2018; Yu *et al.* 2013).

Bir foto-kapan cihazı temelde kamera, kızılötesi flaş ve hareket algılayıcı ile birlikte kullanılmaktadır ve hareket ile tetiklendikten sonra en az 5 adet olmak üzere, hareket devam ettikçe görüntü kaydetmektedir (Kays *et al.* 2010). Gece kaydedilen görüntülerde canlıların fark edemeyeceği kızıl ötesi flaş kullanılmaktadır. Ayarları değiştirilerek farklı sayıda görüntü kaydetmesi veya sms veya e-posta ile uyarı göndermesi sağlanabilmektedir (Kays *et al.* 2010).



Şekil 2.6. Örnek foto-kapan görüntüsü (Rovero *et al.* 2013)

### 2.2.3. Yapılan çalışmalar

Nesnelerin interneti ile yapılmış cisim tespit etme ve tanıma çalışmalarında yaygın olarak bulut mimarisi kullanılmakta ve öznelik tabanlı örüntü tanıma veya görüntü alma (image retrieval) işlemleri gerçekleştirilmektedir. Bu çalışmalarda sunucularda cisimlere ait model görüntülerinin öznelikleri ile oluşturulmuş veri tabanı kullanılarak, nesnelere elde edilen görüntülerin SIFT, SURF, HOG ve ORB gibi öznelik dönüşümü yöntemleri ile sınıflandırma işlemi veya öznelik eşleştirme ile benzerlik taraması işlemi gerçekleştirilmektedir.

Tyagi'nin 2010 yılında yaptığı tezde SURF öznelik dönüşümü yöntemi ile android platformunda cisim tanıma problemi gerçekleştirilmiştir (Tyagi, 2010). Çalışmada mobil

platformdan elde edilen görüntünün öznitelikleri sunucuya iletilmekte, sunucudaki veri tabanında öznitelik eşleştirme yöntemi ile görüntü etiketlenmektedir. Çalışmada kullanılan veri tabanı, tespit edilen cisimler hakkında bilgi, tespit ile ilgili parametreler ifade edilmemiş olup, dönme değişmezliği, aydınlatma değişimi, bakış açısı, ölçekleme değişmezliği için kullanılan çözümler, cisim tespitinde sunucu ve mobil platformda gerçekleştirilen işlem miktarı ve kullanılan yazılım teknolojileri ifade edilmiştir.

Jain ve Rajankar'nın 2017 yılında yaptıkları çalışmada (Jain and Rajankar 2017), SIFT, SURF ve ORB öznitelik dönüşümü yöntemleri ile gerçek zamanlı cisim tespiti ve tanımlama amaçlanmıştır. Yapılan çalışmada internet nesnesi olarak varsayılan gömülü sistem modülü ile elde edilen görüntülerde, 10 adet cisim için oluşturulmuş model veri tabanı kullanılarak ile öznitelik eşleştirme ile cisim tespiti gerçekleştirilmiştir ve gerçek zamanlı uygulamadaki öznitelik bazlı gecikme süreleri ifade edilmiştir. Yapılan çalışmada cisimden bağımsız olarak, elde edilen görüntünün tamamının öznitelikleri internet nesnesinde çıkarılmakta, öznitelikler internet üzerinden iletilerek veri tabanındaki öznitelikler ile eşleştirme yolu ile karşılaştırılarak model sınıflardan birisi olarak etiketlenmektedir. Kullanılan yöntem akıllı nesne için örnek olsa da kullanılan görüntülerin model ve yapay görüntüler olması, aranan cisimlerin herhangi birinin hedef görüntülerde zaten mevcut olması nedeniyle tespit işlemine gerek olmaması, görüntünün tamamında öznitelik çıkarma ve eşleştirme işleminin uygulanması gibi gerçek zamanlı cisim tespit etme ve tanıma problemi için dezavantajları bulunmaktadır.

Kawano ve Yanai'nin 2013 yılında yaptığı çalışmada (Kawano and Yanai 2013) mobil platformda gerçek zamanlı yiyecek tanıma işlemi gerçekleştirilmiştir. Yapılan çalışmada model olarak belirlenen yiyecek türleri, grafik kesme ve destek vektör makineleri yöntemleri kullanılarak hedef görüntüde gerçek zamanlı tür tespiti yapılmıştır. 50 yiyecek türünün bulunduğu veri tabanında, mobil platformdan elde edilen görüntülerdeki yiyecekler SURF, HOG ve renk histogramı kullanılarak etiketlenmektedir. Çalışmada yiyecek tespit ve tanımlama işlemleri ile sınırlayıcı kutu belirleme gibi işlemlerde harcanan ortalama süreleri ifade edilmiştir.

Akıllı nesnelere ile ilgili yapılmış en kapsamlı öneri Wang ve arkadaşlarının 2018 yılında yaptığı 5G ve nesnelere interneti paradigmasıdır (Wang *et al.* 2018). Bu paradigma internet nesnelere ve zekânın birleşmesi olarak ifade edilmektedir. Akıllı nesnelere, cisim tanıma, doğal dil işleme ve akıllı kontrol gibi zor uygulamaları gerçekleştirmek için yüksek fonksiyonel işlevler içeren cihazlardır. Şu anda, makine öğrenimi ve derin öğrenme bu hedeflere ulaşmak için umut verici yöntemlerdir. Bazı kilit teknolojilere, özellikle de 5G iletişim protokollerine ve veri alışverişi standartlarına dayanarak, İnternet ve istihbaratın kesişim noktası olarak, akıllı İnternet, daha yüksek iletişim kabiliyetine sahiptir. Son olarak, işlem merkezi, nesne işlemcisi ve algılama bölgeleri de dâhil olmak üzere İnternet'i, şeyleri ve zekâyı birleştiren 5G I-Nesnelere interneti kavramına odaklanıyoruz. İşleme merkezi, 5G I-Nesnelere internetinin yapay zekâ yöntemleri ile verilerle ilgilenmesini sağlamayı amaçlamaktadır. İşleme merkezi sürekli olarak çevrimiçi olarak otomatik olarak büyük verileri işler. Nesne işlemci, yalnızca algılama bölgelerini ve işlem merkezini bağlamakla kalmaz, aynı zamanda ham verileri algılama bölgelerinden işler ve bunları işleme merkezine iletir. Daha önemlisi, nesne işlemcisi akıllıca bir işlem yapmak için işlem merkezinden geri bildirim alabilir. Algılama bölgeleri, ham veri algısını gerçekleştirmek için kullanılan devasa algılayıcılardır (Wang *et al.* 2018).

Nesnelere interneti verileri, benzersiz özellikleri dikkate alınarak, hem 5G hem de Nesnelere internetine güven, güvenlik ve gizlilik gibi yeni zorluklar getirmektedir. Aynı zamanda veri depolama ve veri işleme konularında hesaplama karmaşıklığı ve maliyeti üzerinde doğrudan bir etkiye sahiptir. Bu arada, farklı senaryolarda iletimin etkinliği ve kanal kullanımı da zordur. Bu nedenle, büyük verileri işleme ve iletim kanalını optimize etme kapasiteleri, makine öğrenmesi ve derin öğrenme gibi daha akıllı yöntemler gerektirmektedir (Wang *et al.* 2018).

Bulut sistemlerin işlem merkezi akıllı bir hesaplama modülü ve bir yürütme modülü olmak üzere iki bileşen içermektedir. Akıllı bilgi işlem modülü, çoklu bilgi işleme sistemlerinden oluşan bir bilgi işlem merkezi olarak önerilmektedir. Çevrimiçi öğrenmeyi gerçekleştirmek için, bu sistemler bir sonraki bölümde tanıtılan derin bir

öğrenme algoritması kullanmaktadır. Her akıllı bilgi işlem sistemi, sunucu kümelerinin dâhil edildiği ve iç bağlantı ağları aracılığıyla tümleşik olduğu bağımsız bir akıllı işlem birimidir. İşlem merkezi, hesaplama ve karar vermede iyi bir performans sergilemekte ve aynı zamanda, büyük verilerin nesne işlemcisinden akıllıca işlenmesi ve nesne işlemcisine veya veri depolamasına talimatların gönderilmesinden sorumlu olmaktadır. Nesne işlemcisi, baz istasyonunda kurulmakta ve ham verileri işlemek, işlenmiş verileri işlem merkezine iletmek ve daha akıllı stratejiler ve eylemler yapmak için geri bildirim almak üzere, algılama bölgelerini ve işlem merkezini birleştirmeye adanmaktadır (Wang *et al.* 2018).

Nesnelerin internetinde, cisim tespit etme ve cisim tanıma gibi görüntü tabanlı problemlerin çözümü için, düşük güç tüketimine sahip derin öğrenme mimarisi de önerilmektedir. Bu mimaride amaç harici güç kaynağı olan internet nesnelere bu problemlerde kullanacakları modelleri düşük güç harcayarak yapay sinir ağlarını eğitmek suretiyle elde etmesi ve kullanmasıdır. Yapılan çalışmada büyük çaplı eğitimin bulut sisteminde, küçük çaplı eğitimlerin ise nesnelere interneti nesnesinde yapılması önerilmektedir (Venkataramani *et al.* 2016).

2016 yılında yapılan nesnelere interneti-bulut altyapısında dağıtılmış video gözetimi için merkezi olmayan yüz tanıma çalışmasında, nesnelere interneti ağının uç kısmı olan nesnede yüz tanıma için gerekli ön işlem adımlarından olan kenar ve köşe belirleme işlemleri ve Haar özellik tabanlı sınıflandırma ile yüz tespiti gerçekleştirilmiştir. Mimarisi Şekil 2.7’de gösterildiği üzere tespit edilen yüzler bulutta bulunan veri tabanında LBP öznitelikleri kullanılarak eşleştirilerek sınıflandırılmaktadır (Amin *et al.* 2016).



**Şekil 2.7.** Ön işleme katmanı kullanılan uygulama (Amin *et al.* 2016).

Foto-kapanlarla ilgili yapılan çalışmalar incelendiğinde, çevrim içi veya çevrim dışı olarak nesnelerin interneti kapsamında olsa bile, gelecek vizyonu karşılayabilecek akıllı nesne olmaya aday bir çalışma bulunmamaktadır. Yapılan çalışmalarda önerilen sistemler genellikle model cisim görüntüleri kullanıldığından gerçek zamanlı ve büyük çaplı çalışmalara uygun olmadığı anlaşılmaktadır. Foto-kapan yapı itibariyle nesnelerin interneti uygulamasına benzemektedir. Foto-kapanlar hareket ile tetiklendiğinde elde ettiği görüntüleri gsm alt yapısı ile sunuculara ulaştırmaktadır ve bir internet nesnesi gibi görev yapmaktadır. Foto-kapanlarla elde edilen görüntülerin büyük çoğunluğuna hareketli bir cisim bulunmamaktadır ve görüntüler veri tabanında veya foto-kapanların belleklerinden kopyalanarak toplandıktan sonra, bilgisayarlar kullanılan görüntüler üzerinde cisim tespit etme ve tanıma amacıyla gerekli işlemler gerçekleştirilmektedir. Foto-kapanların kontrol edilebilme anlamında çevrim dışı olması, cisim tespit etme ve tanıma çalışmalarının veriler toplandıktan sonra yapılması, foto-kapanların kullanım alanlarını, işlevlerini ve işlevlerinin faydalarını oldukça azaltmaktadır. Ayrıca çalınma gibi diğer problemlerin önüne geçilmesi gibi ihtiyaçları da bulunmaktadır.

Son yıllarda foto-kapanların kullanılması ile hayvanlar üzerinde yapılan tür ve nüfus tespiti, sınıflandırma ve bölgesel çeşitlilik gibi çalışmalar hızla artmaktadır. Harekete

duyarlı foto-kapanlar, sahip oldukları kameralar sayesinde çok çeşitli hayvan türleri ile ilgili detaylı bilgi elde edilmesini sağlamaktadır. Foto-kapanlar, dikkat çekmemesi, düşük güç tüketimine sahip olması, doğru ve çeşitli veriler sunması nedeniyle hareketli nesne izleme için uygundur (Kays *et al.* 2010). Yapılan çalışmalar ile canlıların tespit edilmesi, türlerin tanınması (DeArruda *et al.* 2018; Gray *et al.* 2019; Hoebeke *et al.* 2018; Meek *et al.* 2013; Nguyen *et al.* 2017; Norouzzadeh *et al.* 2018; Willi *et al.* 2019; Yousif *et al.* 2019), vahşi hayvan tanıma (Andavarapu and Vatsavayi 2017), canlıların sayılması işlemleri, elde edilmiş çok sayıda foto-kapandan görüntüsü kullanılarak bilgisayarlı görü ve örüntü tanıma yöntemleri ile otomatikleştirilmeye çalışılmaktadır. Foto-kapanlar ilgili yapılan anket çalışmasında foto-kapanlarda önümüzdeki 10-20 yıl içerisinde olması istenen özellikler içerisinde nesne olmayan görüntülerin filtrelenmesi %54 ile üçüncü önemli konu başlığı olarak ifade edilirken, bu süreçte otomatik nesne tespitinin bu istenen özellikler içerisinde ilk gerçekleştirilecek işlev olduğu düşünülmektedir (Glover *et al.* 2019).

Foto-kapan görüntüleri üzerinde yapılan çalışmalar incelendiğinde, yaygın olarak görüntünün her bir alt parçası, model nesne görüntülerinin öznelikleri kullanılarak arka plan, hayvan veya insan olarak sınıflandırılmaktadır (Buehler *et al.* 2019; Gray *et al.* 2019; B. Wang, 2014; Yousif *et al.* 2019). Bazı derin öğrenme tabanlı yöntemlerin uygulandığı çalışmalarda ise nesnelere ait derin öznelikleri çıkarılabildiği için çok sayıda model nesne görüntüsü kullanılarak çok sayıda hayvan türü tespit edilmeye ve sınıflandırılmaya çalışılmaktadır (Andavarapu and Vatsavayi 2017; DeArruda *et al.* 2018; Norouzzadeh *et al.* 2018; Willi *et al.* 2019; Zhang *et al.* 2016). Model görüntülerin kullanıldığı bazı çalışmalarda öncelikle el ile kırılmış nesne görüntülerinin yerel veya genel öznelikleri çıkarılmakta, nesnelere ait bu öznelikleri sınıflandırma yöntemleri modellenmekte ve daha sonra görüntülerin her alt parçası sınıflandırılarak nesne olup olmadığı tespit edilmektedir (Andavarapu and Vatsavayi 2017; DeArruda *et al.* 2018; Norouzzadeh *et al.* 2018; Willi *et al.* 2019; Zhang *et al.* 2016). Derin öğrenme tabanlı yöntemlerde ise model görüntüler tasarlanmış ağ mimarisinde belirlenen devir sayısınınca eğitilmekte ve nesnelere ait derin öznelikleri ile nesnelere modellenmektedir. Bu nesne modelleri ile görüntülerdeki alt parçalar



sınıflandırılmakta ve nesnelere tespit edilmektedir (Norouzzadeh *et al.* 2018; B. Wang, 2014; Willi *et al.* 2019).

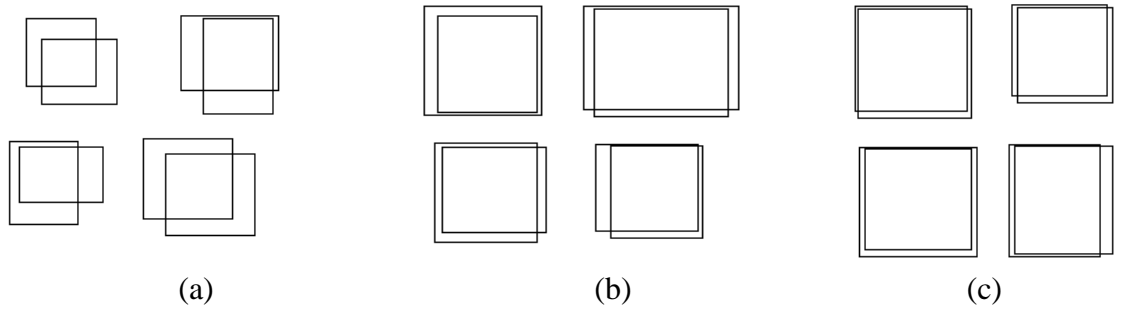
Foto-kapan ile elde edilen görüntülerde karmaşık ve hareketli arka plan olması, ışık şiddeti değişiminin fazla olması ve hareketli nesnenin görüntülerde kısmi olarak bulunması problemi zorlaştırmaktadır (Kulchandani and Dangarwala 2015; Nguyen *et al.* 2017; Wang, 2014; Yu *et al.* 2013). Foto-kapan görüntülerinde yapılan çalışmalarda öncelikli amaç aranan nesnenin görüntüde olup olmadığının tespit edilmesidir. Yu ve arkadaşları, Andavarapu ve arkadaşları ve Buehler ve arkadaşlarının yaptığı (Andavarapu and Vatsavayi, 2017; Buehler *et al.* 2019; Yu *et al.* 2013) çalışmalarda örnek görüntüleri ait öznitelikler kullanılarak, kayan pencere tabanlı yöntemlerle aranan nesne tespit edilmeye çalışılmıştır. Zhang ve arkadaşlarının yaptığı (Zhang *et al.* 2016) çalışmada arka plan ve ön plan olarak belirlenen görüntülerin öznitelikleri ve çok katmanlı grafik kesim yöntemi kullanılarak her görüntü parçasının arka plan veya nesne olduğu tespit edilmiştir. Khorrami ve arkadaşlarının yaptığı (Khorrami *et al.* 2012) çalışmada Sağlam Temel Bileşen Analizi (RPCA-Robust Principal Component Analysis) ile arka planı çıkarılmış görüntülerde yerel entropi ve bağlı bileşen analizi yöntemleri ile nesne tespiti gerçekleştirilmiştir. Yapılan derin öğrenme tabanlı (DeArruda *et al.* 2018; Gray *et al.* 2019; Hoebeke *et al.* 2018; Nguyen *et al.* 2017; Norouzzadeh *et al.* 2018; Willi *et al.* 2019; Yousif *et al.* 2019) çalışmalarda ise tespit işlemi elle kırılmış görüntüler kullanılarak eğitilen sinir ağlarının kullanılması ile görüntülerde nesne tespit işlemi gerçekleştirilmiştir.

Foto-kapan görüntülerinde nesne tespit etme ve tanıma üzerine yapılan çalışmalarda, çoğunlukla kırpma işleminin elle yapıldığı model görüntüler ön bilgi olarak kullanılmaktadır (De Arruda *et al.* 2018; Gray *et al.* 2019; Hoebeke *et al.* 2018; Nguyen *et al.* 2017; Norouzzadeh *et al.* 2018; B. Wang, 2014; Willi *et al.* 2019; Yousif *et al.* 2019; Yu *et al.* 2013). Bu çalışmalarda daha önceden kaydedilmiş foto-kapan görüntülerinden insanlar tarafından tespit edilmiş hayvan görüntüleri el ile kırılmaktadır. Kırılan bu model görüntüler öznitelik çıkarma ve görüntü tanıma yöntemlerinde eğitim verisi olarak kullanılmaktadır (Buehler *et al.* 2019; B. Wang,

2014). Bu nesnelerin otomatik olarak tespit edilmesi için sırasıyla tespit ve konumlandırma aşamalarının uygulanması gerekmektedir. Tespit aşamasında arka plan modeli oluşturulmakta ve her görüntünün arka plan modeli ile olan farktan hareketli nesnenin bulunup bulunmadığı belirlenmektedir. Devamında konumlandırma aşamasında ise hareketli nesnenin görüntü içerisinde bulunduğu konum ve boyutları belirlenmektedir (He *et al.* 2016; Lin *et al.* 2014).

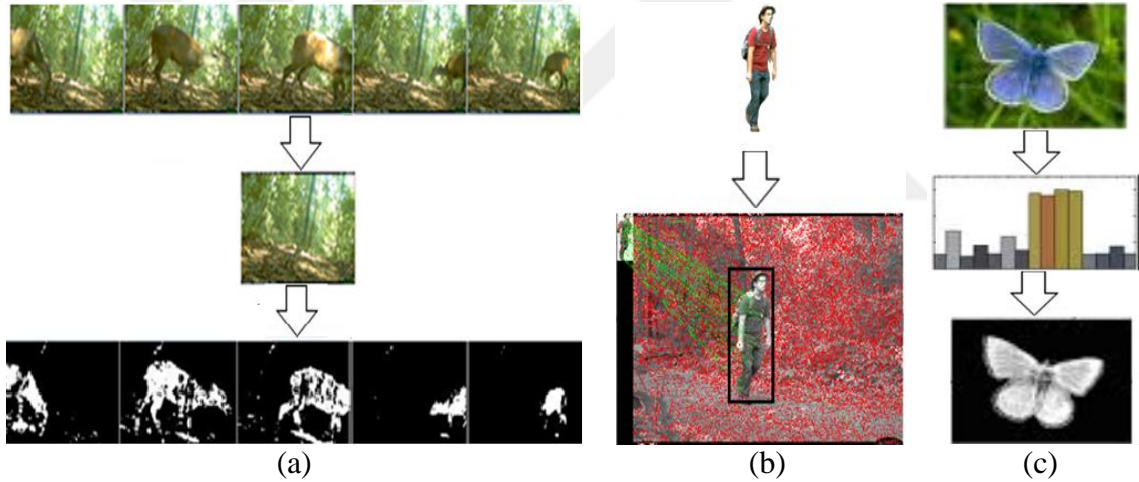
### 2.3. Bilgisayarlı Görü

Bir cisim tanıma sistemi, gerçek dünyadaki bir cismin, bilinen bir cisim modelini kullanarak görüntü içerisinde tespit edilmesidir. Cisim tanıma problemi bilinen cisimlerin modellerine dayanan bir etiketleme problemi olarak tanımlanmaktadır. Cisim tespitinin amacı bir görüntüde cisim olup olmadığını ve eğer varsa görüntüdeki konumunun belirlenmesidir. Geçtiğimiz on yılda bu problem için en baskın yaklaşım kayan pencere yaklaşımı olmuştur (Zitnick and Dollár 2014). Şekil 2.8’de cisim tespitinde başarı metriği olarak kullanılan Birleşimlerin Kesişimi (Intersection over Union) yöntemi gösterilmektedir. Bu yöntemde tahmin edilen cisim bölgesi ile cisme ait gerçek bölgenin kesişimi, bu iki çerçevenin birleşimi ile oranlanmakta ve 0 ile 1 arasında bir değer elde edilmektedir. Elde edilen değer şekilde görüldüğü üzere, 1’e yaklaştıkça daha iyi bir tespit işlemi gerçekleştirildiği anlamına gelmektedir.



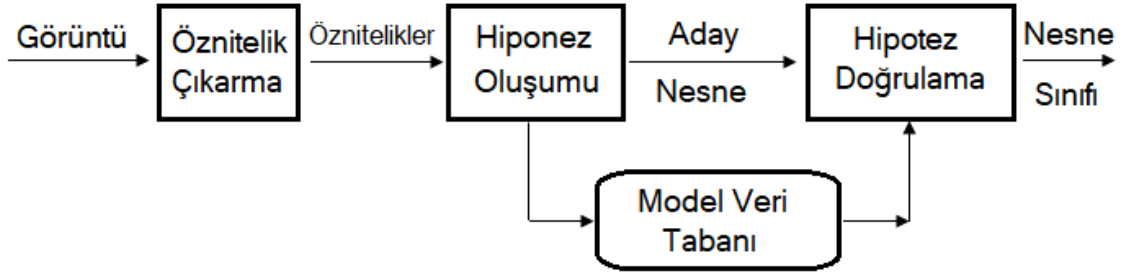
**Şekil 2.8.** IoU değerlerine göre pencere kesişimleri a) 0,5 b) 0,7 c) 0,9 (Zitnick and Dollár 2014)

Cisim tespiti, Şekil 2.9’da görüldüğü üzere bilinen model bir görüntünün öznelikleri veya özellikleri kullanılarak yapılabileceği gibi, hareket veya belirginlik tabanlı yöntemler ile de yapılabilmektedir. Hareketli cisim tespiti, ardışık görüntülerinde sabit olmayan bölgelerin tespit edilmesi işlemidir (Kulchandani and Dangarwala 2015). Temel amaç hareketli ön plan hedeflerinin tespit edilmesi ve takip edilmesidir (Piccardi, 2004). Literatürde bu amaçla arka plan çıkarma, optik akış, çerçeve ve geçici farklar olmak üzere 3 temel yaklaşım bulunmaktadır (Hu *el al.* 2004). Belirginlik tabanlı yöntemlerde cismin arka plandan ayrılması amacıyla renk, histogram, kontrast gibi farklı ve belirgin özellikler kullanılmaktadır. Gerçek hayat görüntülerinin karmaşık olması ve cisimlere ait yeterli belirginlik farkı elde edilememesinden dolayı belirginlik tabanlı yöntemler bu tezde kullanılmamıştır.



**Şekil 2.9.** Cisim tespit yaklaşımları a) hareket tabanlı b) öznelik tabanlı c)belirginlik tabanlı

Öznelik tabanlı tespit işlemi, genellikle model görüntünün öznelikleri, hedef görüntüde farklı boyutlardaki kayan pencerelerde sınıflandırma işlemi veya öznelik eşleşme sayıları kullanılarak gerçekleştirilmektedir. Şekil 2.10’da görüldüğü üzere cisme ait model görüntü sayısı veya öznelikler çoğaltılarak cisme ait hipotez oluşturulmakta, hedef görüntüdeki pencereler bu model veri tabanı ile karşılaştırılarak daha etkili cisim tespiti yapılmaktadır.



**Şekil 2.10.** Cisim sınıflandırma (Jain *et al.* 1995)

Şekil 2.10'da ifade edilen model veri tabanı, tanıma sisteminde cisme ait bilinen tüm modelleri içermektedir. Model veri tabanındaki bilgiler, tanıma için kullanılan yaklaşıma bağlı olmakla birlikte, niteliksel, fonksiyonel veya kesin geometrik yüzey bilgilerine kadar her şey olabilmektedir. Birçok durumda cisimlerin modelleri soyut öznitelik vektörleri olmaktadır. Boyut, renk, şekil yaygın olarak kullanılan bazı temel özniteliklerdir. Öznitelik çıkarma işleminde öznitelik çıkarma operatörleri görüntüye uygulanmakta ve model cisim hipotezlerini oluşturmaya yardımcı olan özniteliklerin anahtar noktaları (keypoint) belirlenmektedir. Hipotez oluşturma aşamasında tespit edilen öznitelikler kullanılarak görüntüdeki mevcut cisimlere olasılık değeri atanmaktadır. Bu olasılık değeri arama alanını azaltmak için kullanılmaktadır. Model veri tabanında değerlendirmeyi kolaylaştırmak için bir çeşit indeksleme şeması da kullanılabilir. Daha sonra en yüksek olasılığa sahip olan sınıf etiketi olarak belirlenmektedir (Jain *et al.* 1995).

Öznitelik tabanlı cisim tanıma işleminde, model cisim temsilinde görüntülerin nasıl ve ne şekilde temsil edileceklerine, hangi özniteliklerin problem çözümde etkili olacağına, uygun olan eşleştirme yönteminin kullanılmasına, hipotez oluşumunda cismin en iyi şekilde temsil edilmesine ve tüm olasılıkların hesaplanmasına, cisim etiketleme işleminin uygun sınıflandırma yöntemi ile gerçekleştirilmesine dikkat edilmelidir (Jain *et al.* 1995). Bu yöntemlerde cisim tanıma karmaşıklığını sahne sabitliği, model görüntü uzayı, model cisim sayısı, hedef görüntüdeki cisim sayısı etkilemektedir. Sahne sabitliği görüntülerin benzer aydınlatma, arka plan, kamera parametresi, bakış açısı koşullarında model olarak tespit edilip edilemeyeceğine bağlıdır. Öznitelik algılayıcılarının

performansı farklı sahne koşullarından önemli ölçüde etkilenmektedir. Görüntü model uzayı, üç boyutlu cisimlerin iki boyutlu olarak temsil edildiği durumlarda, modellerin 2 boyutlu özniteliklerinin farklı perspektifleri temsil etmesi beklenmektedir. Model veri tabanındaki cisim sayısı az ise hipotez oluşturma adımına gerek olmayabilmektedir ve sıralı ve kapsamlı bir eşleştirme işlemi yeterli kabul edilmektedir. Hedef görüntüdeki cisim sayısının fazla olması tıkanmayı arttırmaktadır. Tıkanma (oklüzyon) ciddi bir problemdir ve tıkanma beklenen özniteliklerin yokluğu ve beklenmedik özniteliklerin oluşmasıyla sonuçlanmaktadır (Jain *et al.* 1995).

Görüntüdeki cisimlerin tespit edilmesi ve tanınması, özellikle sınıf içi çeşitlilik, karmaşıklık, tıkanıklık ve poz değişikliklerinin olmasından dolayı bilgisayarlı görüdeki en zorlu problemlerden biridir (Zhang *et al.* 2007). Görüntüde cisim tespit etme veya tanıma amacıyla kullanılabilir temelde iki tip öznitelik çıkarılmaktadır. Bunlar genel ve yerel özniteliklerdir. Genel öznitelikler görüntüyü bir bütün olarak tanımlamakta ve tüm bir nesneyi tek bir vektörle genelleştirmektedir. Genel öznitelikler, alan, çevre uzunluğu, fourier tanımlayıcıları ve sabitlerdir (moments). Genel öznitelikler bölge için noktaları veya bölge sınırındaki noktaları dikkate alarak bir bölge için elde edilmektedir. Her durumda amaç tüm noktaları, konumlarını, yoğunluk özelliklerini ve mekânsal ilişkilerini dikkate alarak tanımlayıcıları bulmaktır (Jain *et al.* 1995). Yerel öznitelikler ise, görüntüde belirlenen birden çok anahtar nokta ve bu anahtar noktaların tanımlayıcılara dönüştürülmesi ile elde edilmektedir. Bu tanımlayıcılara dönüştürme işlemi, öznitelğe tıkanma ve karmaşıklığa karşı dayanıklılık sağlamaktadır. Ancak, her nesne veya görüntü için değişken sayıda öznitelik vektörünün bulunduğu durumları ele almak için özel sınıflandırma algoritmaları gerekebilmektedir (Lisin *et al.* 2005). Görüntüdeki nesnelerin tespit edilmesi veya tanınması için kullanılabilir bir yerel öznitelik, yakın komşuluklarından farklılık ifade eden bir görüntü deseni olarak tanımlanmaktadır. Yerel öznitelikler noktalar, kenarlar veya küçük görüntü yamalarıdır. Tipik olarak, bazı ölçümler yerel bir öznitelik üzerinde bulunan bir bölgeden alınmakta ve daha sonra çeşitli uygulamalar için kullanılabilir tanımlayıcılara dönüştürülmektedir. Bir dizi yerel öznitelik, parçalara ayırma gereği duymadan nesnelere veya sahneleri tanımaya yarayan sağlam bir görüntü temsili olarak

kullanılabildikleri için sıklıkla tercih edilebilmektedir (Tuytelaars and Mikolajczyk 2008).

#### **2.4. Hareketli Cisim Tespiti**

Hareketli cisim tespiti, ardışık görüntülerde sabit olmayan bölgelerin tespit edilmesi işlemidir (Kulchandani and Dangarwala 2015). Temel amaç hareketli ön plan hedeflerinin tespit edilmesidir (Piccardi, 2004). Video izleme, güvenlik, video sıkıştırma, otomatik hedef tespit ve takibi, aktivite tanıma gibi yaygın kullanım alanlarına sahip olmasından dolayı bilgisayarlı görü alanında önemli bir yere sahiptir. Literatürde bu amaçla arka plan çıkarma, optik akış, çerçeve ve geçici farklar olmak üzere 3 temel yönelim bulunmaktadır (Hu *et al.* 2004). Arka plan çıkarma yöntemlerinde amaç, ardışık görüntüleri kullanarak arka plan modelini hesaplamak ve hareketli cisimleri bu arka plan modeli ile her bir görüntünün farkından tespit etmektir. Çerçeve veya geçici fark yöntemlerinde, art arda iki görüntü arasındaki hareketli cismin varlığını tahmin edilmekte ve ardışık görüntüler arasındaki hareketli cisimler algılanmaya çalışılmaktadır. Optik akış yönteminde görüntünün optik akış alanı hesaplanmakta ve cisim hareketi hakkında bilgi elde edilmektedir (Kulchandani and Dangarwala 2015). Optik akış yöntemi işlem yoğunluğu nedeniyle fazla zaman gerektirmekte ve optik akış tabanlı yöntemlerin gerçek zamanlı uygulamalarda donanım ivmesiyle kullanılması gerekmektedir. Arka plan çıkarma yöntemleri diğerlerine göre daha iyi performans göstermekte ve bunlar sabit kameralar tarafından kaydedilen videolar üzerinde hareketli cisim tespiti için başarıyla uygulanmaktadır. Çerçeve farkı yöntemleri basitliği, işlem ve bellek gereksinim azlığı nedeniyle etkili olmaktadır ancak cisimler hakkında yetersiz veri elde edilmesi ve arka plan modelinden kaynaklanan hayalet problemi gibi dezavantajları bulunmaktadır (Dhake and Dalu 2018).

Hareketli cisim tespitinde kullanılan yöntemler ışık şiddeti değişimi ve arka plan hareketinden oldukça fazla etkilenmektedir. Literatürdeki bazı yöntemler bu zorluklara karşı kısmen dayanıklılık göstermektedir. Örneğin gaussların karışımı yöntemi hızlı ışık şiddeti geçişlerinde faydalı olmaktadır (Karasulu and Korukoglu 2013). Arka plana ait

komşu blokların benzer farklılıklar göstermesi gerektiği ifade edilen Görüntü Çeşitlerinin Eş Oluşumu yöntemi (Piccardi, 2004), arka plan hareketine ve ışık şiddeti değişimine kısmen dayanıklılık göstermektedir. Ancak bu yöntemler arka plan hareketi ve ışık şiddeti problemlerine sınırlı bir çözüm sunmaktadır ve her yöntemde probleme özgü parametre kullanılması gerekebilmektedir.

#### 2.4.1. Arka plan çıkarma yöntemleri

Arka plan çıkarma yöntemi, sabit kameralardan elde edilen görüntülerde hareketli nesnelere tespit etmek amacıyla yaygın olarak kullanılmaktadır. Bu yöntemler kullanılarak ardışık görüntüler kullanılarak etkili bir arka plan modeli elde edilmekte ve her yeni görüntüde bu model güncellenmektedir. Arka plan çıkarma, hareketli nesnenin tespit ve takip edilmesi için en güvenilir yöntemlerden birisidir (Piccardi, 2004). Bu arka plan çıkarma yöntemlerinin performansı ışık şiddeti değişimine ve arka plan hareketi nedeniyle azalmaktadır (Karasulu and Korukoglu 2013). Yaklaşımın temel mantığı, aynı konumdaki piksel değerleri kullanarak belirli yöntemlerle ardışık görüntülerdeki hareketsiz veya az hareketli pikselleri tespit ederek arka plan görüntüsü veya arka plan modeli olarak adlandırılan bir referans görüntüsü elde etmektir. Bu yöntemler, arka plan modelini çerçevelerin zamansal dizisinden etkili bir şekilde tahmin etmeye çalışmaktadır (Piccardi, 2004). Arka plan çıkarma amacıyla literatürde, Değişen Gauss Ortalaması (Running Gaussian Average), Geçici Medyan Filtre (Temporal Median Filter), Gaussların Karışımı (Mixture of Gaussians), Çekirdek Yoğunluk Tahmini (Kernel Density Estimation (KDE), Görüntü Çeşitlerinin Eş Oluşumu (Co-Occurrence of Image Variations) ve Öz Değer Arka Planı (Eigen Backgrounds) yöntemleri bulunmaktadır.

Değişen Gauss Ortalama metodu her bir piksel  $(x, y)$  konumundan bağımsız olarak,  $n$  adet pikselin son değerlerini Gauss olasılık yoğunluk fonksiyonu üzerine uydurulmasına dayanmaktadır (Piccardi, 2004; Wren *et al.* 1997). Değişen Gauss Ortalama metoduna göre arka planın her bir pikselinin ortalama değeri  $\mu_t$  Denklem 1 ile hesaplanmaktadır.

$$\mu_t = \alpha I_t + (1 - \alpha) \mu_{t-1} \quad (1)$$

Denklem 1’de  $I_t$ , t anındaki görüntüyü,  $\mu_{t-1}$  pikselin önceki ortalama değerini ve  $\alpha$  ise güncelleme hızı olarak tercih edilebilecek bir ağırlık değerini ifade etmektedir (Piccardi, 2004; Wren *et al.* 1997).

Gaussların karışımı metodunda Gauss değerlerinin karışımı vasıtasıyla t zamanında belirli bir piksel değerinin gözlemlenme olasılığı  $P(x_t)$  Denklem 2 ile hesaplanmaktadır (Piccardi, 2004; Wren *et al.* 1997):

$$P(x_t) = \sum_{i=1}^K \omega_{i,t} \eta(x_t - \mu_{i,t}, \Sigma_{i,t}) \quad (2)$$

Denklem 2’de gösterilen Gaussların karışımı algoritmasında,  $x_t$ , t görüntüsünün piksel değerini,  $K$  karışımındaki toplam dağılım sayısını,  $\omega_{i,t}$  t görüntüsündeki k. dağılımın ağırlığını,  $\mu_{i,t}$ , t görüntüsündeki k. dağılımın ortalamasını,  $\Sigma_{i,t}$  ise k. dağılımın standart sapmasını ifade etmektedir (Nurhadiyatna *et al.* 2013; Piccardi, 2004).

Arka plan doğru ya da reaktif değilse hayalet olarak adlandırılan yanlış cisim tespit sonuçları elde edilmektedir (Cucchiara *et al.* 2003). Arka plan çıkarma ile ilgili aşağıdaki tanımların bilinmesi gerekmektedir:

- MVO(Moving Visual Object): Hareketle oluşan, boş olmayan, nesneye ait bağlı noktalar kümesidir.
- Kapanmamış Arka Plan: Hareket etmeyen görünür sahne noktaları kümesidir.
- Arka Plan(Back Ground): Hesaplanmış arka plan modelini ifade etmektedir.
- Hayalet(Ghost): Arka plan çıkarmada hareketli algılanan ancak hareketli cisme karşılık gelmeyen bağlantılı noktalardır.
- Gölge(Shadow): Cismin gölgesi nedeniyle hareketli sanılan arka plan noktaları hayalet ve gölge olarak ikiye ayrılmaktadır (Cucchiara *et al.* 2003).



Arka plan modelinin oluşturulduğu tahmin çerçeveye göre hesaplanmaktadır. Görüntü içindeki hareketli cisimler, geçerli çerçeve ile mevcut arka plan modeli arasındaki fark (Çerçeve Farkı) ile algılanmaktadır (Cucchiara *et al.* 2003). Arka plan çıkarmada hareketli cismin doğru algılanabilmesi için arka plan modelinin olabildiğince doğru tahmin edilmesi ve yüksek tepkime elde edilmesi amacıyla arka plan modelinin cisim hareketinden dolayı ani değişiklikleri yansıtması gibi iki temel sorunu bulunmaktadır.

En hızlı arka plan çıkarma yöntemi, Çizelge 2.2’de gösterildiği üzere arka plan modelinin güncellenmesinin 1-2 parametre ile bir eşikleme işlemi kullanılarak yapıldığı Değişen Gauss Ortalamadır (Piccardi, 2004).

**Çizelge 2.2.** Arka plan çıkarma yöntemleri ve performans analizi (Piccardi, 2004)

Yöntem	Hız	Bellek	Başarı
Değişen Gauss Ortalama	1	1	Az-Orta
Geçici Medyan Filtre	$n_3$	$n_3$	Az-Orta
Gaussların Karışımı	m	m	Yüksek
Çekirdek Yoğunluk Tahmini	n	n	Yüksek
Ardışık KD Tahmini	M+1	M	Orta-Yüksek
Görüntü Varyasyonlarının Eş Oluşumu	$8n/N^2$	$nK/N^2$	Orta
Eigen Arka Planları	M	n	Orta

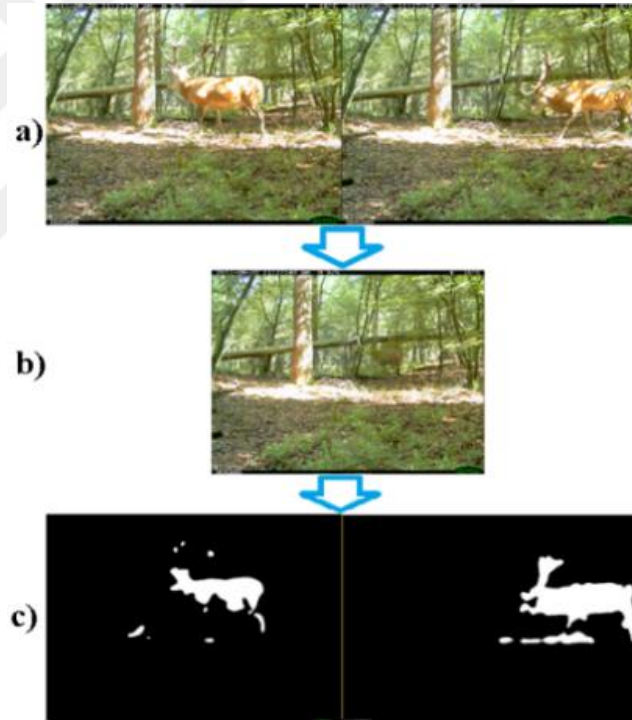
#### 2.4.2. Çerçeve farkı yöntemi

Çerçeve farkı, iki görüntü arasındaki fark kullanılarak hareket eden cismin varlığının tanımlandığı bir yöntemdir ( Jain and Nagel 1979). Çerçeve farkı yöntemi, Denklem 3’te ifade edildiği üzere hareketli nesne tespiti için en basit yöntemlerden biridir. Yöntemde görüntü çıkarma operatörü kullanılarak ilk görüntü çerçevesinden ikinci görüntü çerçeve çıkarılmakta ve hareketli nesneyi arka plandan ayıran çıktı görüntüsü elde edilmektedir (Jain and Nagel 1979). Şekil 2.11’deki örnekte görüldüğü üzere, görüntüler arka plan modeli gibi bir görüntü ile işleme alınmakta, sonuç olarak ise

hareketli bölgelerin beyaz tonlarda, hareketsiz arka plan bölgelerinin siyah tonlarda olduğu ikili görüntü elde edilmektedir. Bu işlemden sonra literatürde genellikle bir eşikleme işlemi uygulanmakta ve gölge ve hayalet gibi hatalar giderilerek hareketli cisme ait daha net bilgiler elde edilmektedir.

$$I_{d(k,k+1)} = |I_{k+1} - I_k| \quad (3)$$

Yukarıdaki Denklem 3'te  $I_k$  ardışık görüntülerdeki  $k$ . görüntüyü ve  $I_{k+1}$  ise bir sonraki  $(k+1)$  inci görüntüyü ifade etmektedir (Jain and Nagel 1979; Singla, 2014).



**Şekil 2.11.** Çerçeve Farkı yöntemi a) örnek görüntü b) arka plan modeli c) çerçeve farkı çıktısı

### 2.4.3. Üç ve dört çerçeve farkı yöntemleri

Sabit kamera görüntülerinde hareket eden cisim tespiti için literatürde iki (R. Jain and Nagel 1979; Singla, 2014), üç (Xiao *et al.* 2010) ve dört (Abdelli and Choi 2017) çerçeve farkı yöntemleri bulunmaktadır. Üç çerçeve farkı metodu, ardışık üç görüntüden elde edilen iki kare farkı çıktısına mantıksal ve (&) operatörü kullanılması dayanmaktadır (Gang *et al.* 2013; Wojcik and Kaminski 2004; Xiao *et al.* 2010; Zhang *et al.* 2012). Üç çerçeve farkı yönteminde, hareketli cisimler hakkında iki çerçeve fark yönteminden daha fazla bilgi alınması amaçlanmaktadır. Üç çerçeve farkı yönteminde uygulanan aşamalar ile ortanca görüntüdeki kenarların elde edildiği belirtilmektedir (Gang *et al.* 2013). Dört çerçeve farkı yöntemi, üç çerçeve farkı yönteminde gölgeleri önlemek için eşik parametresiyle art arda dört görüntünün kullanılmasına dayanmaktadır. Bu üç eğilimin kaçırdığı nokta, çerçeve fark yönteminin sıralı görüntülerde hareketli cismin tümüne uygulanmamasıdır. Çerçeve farkı yönteminde, birleşme veya birliğin hareketli nesnenin kesişiminden farkı, sıralı görüntülerde elde edilebilmektedir. Üç çerçeve fark yönteminde, bu cisimlerin kesişimi elde edilebilmektedir.

Literatürde, üç çerçeve farklılığına dayanan üç geliştirilmiş versiyon bulunmaktadır. Bu yöntemlerde çerçeve farkı çıktılarına morfolojik analiz yöntemi (Zhang *et al.* 2012), kenar birleştirme algoritmaları (Gang *et al.* 2013), optik akış (Han *et al.* 2015) uygulanmaktadır. Zhang'ın çalışmasında (Zhang *et al.* 2012), üç çerçeve farklılığı ve en temel matematiksel morfoloji yöntemleri, erozyon ve genişleme yöntemleri, hareketli cismin kenarlarının etkili bir şekilde temizlenmesi ve doldurulması için kullanılmaktadır. Gang'ın çalışmasında üç çerçeve farkı yöntemine ek olarak (Gang *et al.* 2013), Canny kenar algılama algoritması gürültüleri temizlemek ve hareketli cisimleri algılamak için kullanılmaktadır. Eğilim ve eğilim yönleri, algılanan kenarları birleştirmek için kullanılmaktadır. Gürültüyü temizlemek için, görüntünün %1'inden küçük olan bölgeler silinmektedir. Bu yöntemden dinamik arka planda verim alınamamaktadır (Kulchandani and Dangarwala 2015). Han'ın çalışmasında (Han *et al.* 2015), üç çerçeve farklılığından elde edilen sonuç görüntüsü OTSU eşikleme yöntemi

ile ikili bir görüntüye dönüştürülmektedir. Cisim, Harris köşe algılama algoritması tarafından çıkarılan alanlarda optik akış yöntemi uygulanarak tespit edilmeye çalışılmaktadır. Huang'ın çalışmasında (Huang *et al.* 2007) arka plan modelini oluşturmak ve hareketli bölgeleri tespit etmek için zaman atlamalı olarak seçilmiş görüntülerin kullanılması önerilmektedir. Ortam gürültüsünü etkin bir şekilde temizlemek için ilgi alanlarına iki boyutlu genel eşikler uygulanmıştır. Zhang'ın çalışmasında (Zhang and Zhang 2013), art arda 5 görüntüde üç çerçeve fark yöntemi uygulayarak hareketli cisim tespit etmeye çalışılmaktadır. Arka plan modelinin ortalama 5 görüntü kullanılarak hesaplandığı çalışmada, hareketli cismin dış hat bilgisi elde edilmektedir. Çalışmada birden fazla hareketli cisim tespit etme imkânı bulunmamaktadır (Kulchandani and Dangarwala 2015).

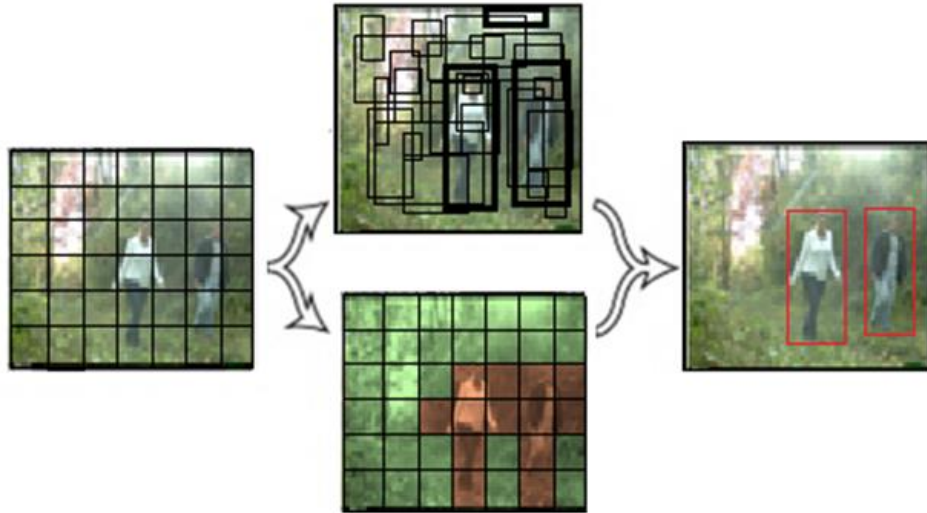
## 2.5. Derin Öğrenme ile Cisim Tespiti

Evrişimli sinir ağlarının kullanımında 3 farklı eğilim bulunmaktadır. Bunlar büyük ağlar ve derin ağlar, çoklu ağlar ve farklı (diverse) ağlardır. Büyük ağlarda temel yaklaşım, ağların derinliğini ve genişliğini arttırarak daha iyi performans elde etmektir. Bu eğilime örnek olarak, GoogleNet ve VGG ağlarında sırasıyla 22 ve 19 katman bulunmaktadır. Bu eğilimde, birbirinden farklı derin ağlar ardışık olarak birbirine bağlanarak kademeli olarak da kullanılmaktadır. Çoklu ağlarda ise temel eğilim, birbirinden bağımsız farklı paralel olarak çalışan derin ağların sonuçlarını birleştirmektir. Buna ilk örnek olarak önerilen çok sütunlu DNN (Multi-Column DNN) modeli, insanlar ile rekabet edebilecek sonuçlara ulaşmıştır (Guo *et al.* 2016) . Farklı ağlarda ise CNN yapısına ek olarak yüzeysel bilgiler içeren yapılarla ve bağlamsal bilgilerle birleştirilerek performans artışı amaçlanmaktadır. DVM ile birleştirilen RCNN metodu bu kategoride bulunmaktadır ve CNN ile Fisher Vector'ün birleştirilmesi görüntü sınıflandırmada doğruluğu arttırmak için kullanılmaktadır (Guo *et al.* 2016).

YOLO (You Only Look Once: Unified, Real-Time Object Detection), GoogleNet mimarisinden esinlenilmiş, görüntüdeki nesnelere ait uzamsal olarak ayrılmış sınırlayıcı kutularının sınıf olasılıklarının regresyon problemi olarak alındığı, tek bir sinir ağında

bu sınıf olasılıklarının tahmin edildiği bir evrişimli sinir ağı mimarisidir (Tompson *et al.* 2015). Cisim tespit işleminin Şekil 2.12’de gösterildiği bu yöntemde standart mimaride saniyede 45 görüntü, hızlı YOLO ile saniyede 155 görüntü gerçek zamanlı işlenebilmektedir. Literatürdeki en gelişmiş yöntemlere göre daha az yerleştirme hatası yapmaktadır ancak arka planın nesne olarak tanımlandığı false pozitive değeri diğerlerine oranla biraz daha yüksektir. YOLO mimarisi, nesnelerin genel temsillerini öğrenmekte, DPM ve R-CNN tespit mimarilerinden daha iyi performans göstermektedir (Tompson *et al.* 2015).

YOLO mimarisinde, 24 adet evrişim katmanı, bu katmanların devamında 2 adet tam bağlı katman bulunmaktadır. Alternatif 1x1 evrişimli katmanlar, önceki katmanlardan gelen özniteliklerin boyutunu azalmaktadır (Tompson *et al.* 2015).



**Şekil 2.12.** Yolo mimarisinde cisim tespiti

Mimarinin son katmanında hem nesnelere ait sınıf olasılıkları, hem de bu sınıflara ait sınırlayıcı kutu koordinatları öngörülmektedir. Son katmanda doğrusal bir aktivasyon fonksiyonu kullanılırken, diğer tüm katmanlarda Denklem 4’te gösterilen sızdıran düzeltilmiş (leaky rectified) doğrusal aktivasyon kullanılmaktadır (Tompson *et al.* 2015):

$$\phi(x) = \begin{cases} x, & \text{if } x > 0 \\ 0.1x, & \text{otherwise} \end{cases} \quad (4)$$

Modelin çıktısında optimizasyonu kolay olan kareler toplamı hatası uygulanmaktadır. Nesne içermeyen tahminlerin elenebilmesi için güven parametresi ve üst üste gelen sınırlayıcı kutuların kabul edilmesi için eşik parametresi olmak üzere 2 parametre kullanılmaktadır. Sınırlayıcı kutular, en yüksek IoU (Intersection over Union) değerine dayanan bir tahmin işlemi kullanılarak etiketlenmektedir. Nesne tespiti için sinir ağı, PASCAL VOC 2007 ve 2012'den eğitim ve doğrulama veri setleri hakkında yaklaşık 135 devir boyunca eğitilmektedir (Tompson *et al.* 2015).

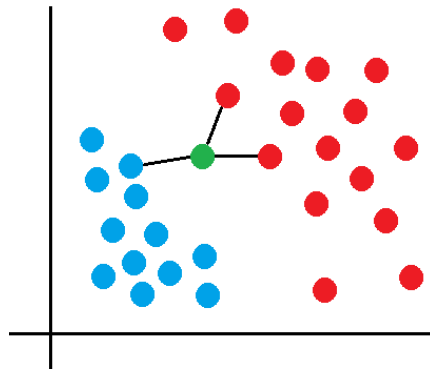
## 2.6. Örüntü Tanıma ve Sınıflandırma

Cisim tespit etme ve tanıma işlemlerinde, örüntü tanıma ve sınıflandırma yöntemleri kullanılmaktadır. Cisimlere ait öznitelikler tespit edildikten sonra yapılması gereken aşamadır. Sınıflandırmada temel fikir, cisimleri öznitelikleri kullanılarak tanımadır. Bu aşamada cisimlere ait hipotezler oluşturulmakta ve bu hipotezler doğrulanmaktadır. Çoğu tanıma yaklaşımında bu iki adım çeşitli miktarda birleştirilerek kullanılmaktadır. Sinir ağı tabanlı yaklaşımlar da bu sınıfa girmektedir (Jain *et al.* 1995). Çalışmada sınıflandırma işleminde K en yakın komşu, destek vektör makineleri, rastgele orman ve yapay sinir ağı tabanlı derin öğrenme yöntemleri kullanılmıştır.

### 2.6.1. K en yakın komşu algoritması (kNN)

K en yakın komşu sınıflandırma işlemi bir örneğin, bir sınıfa olan benzerliğini, her bir sınıfı temsil eden noktalardan uzaklığı hesaplayarak, en yakın sınıfa atanmasıdır. Benzerliklerin hesaplandığı mesafe Öklid veya herhangi bir ağırlıklı öznitelik kombinasyonu olabilmektedir (Jain *et al.* 1995). K En Yakın Komşu algoritması, sınıflandırma amacıyla kullanılan bir denetimli öğrenme yöntemidir. K en yakın komşu yöntemi, Thomas. M. Cover ve Peter. E. Hart tarafından önerilmiştir. Bu algoritma eski, basit ve etkili örüntü sınıflandırma yöntemlerinden biridir ve makine öğrenme

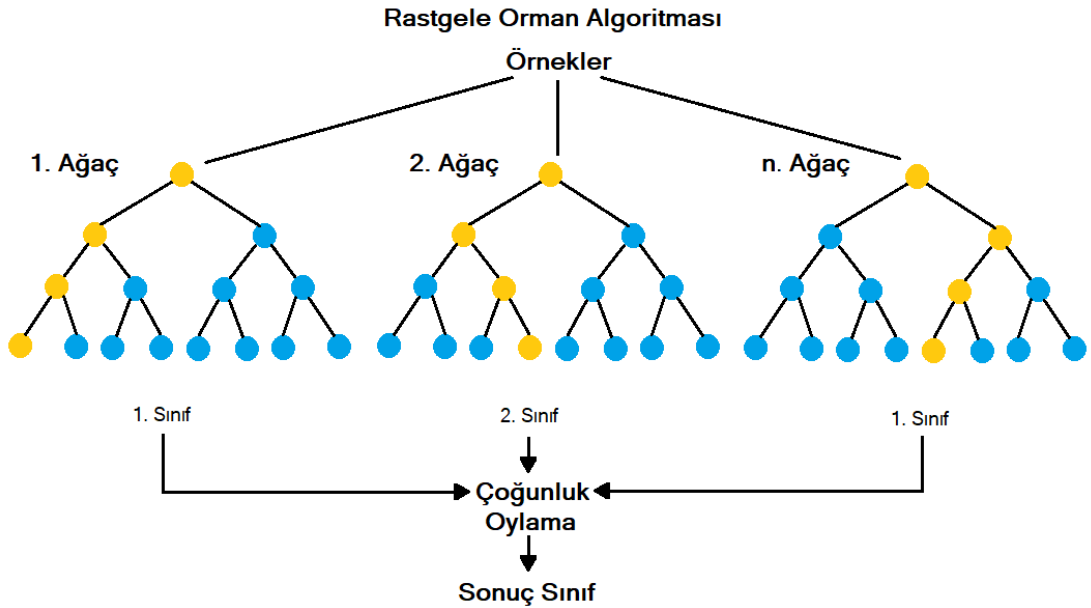
algoritmaları içerisinde en popüleridir. Verilerin sınıflandırılması örüntü tanıma, veri madenciliği, yapay zekâ, istatistik, bilişsel psikoloji, tıp, biyoinformatik gibi birçok alanda kullanılmaktadır bulunmaktadır (Taşcı and Onan 2016). Yöntemde sınıflandırmada kullanılacak verilerin eğitim kümesindeki normal davranış verilerine uzaklıkları belirlenerek, en yakın olduğu tahmin edilen  $k$  verinin ortalamasının alınmasıyla, belirlenen eşik değere göre sınıfları belirlenmektedir. Bu algoritmada en önemli nokta her bir sınıfın özniteliklerinin önceden belirlenmiş olmasıdır (Çalışkan and Soğukpınar 2008). Eğitimin işleminin bulunmaması, gerçekleştirilmesinin kolay, analitik olarak takip edilebilir, yerel bilgilere uyarlanabilir, paralel programlamaya uygun, gürültülü eğitim verilerine karşı dayanıklı olması gibi avantajlar sağlamaktadır. Bu avantajlarının yanı sıra, büyük miktarda bellek alanına ihtiyaç duyması, veri seti ve öznitelik boyutu arttıkça işlem yükünün ve maliyetin önemli miktarda artması, performansın  $k$  komşu sayısı, uzaklık ölçütü ve öznitelik sayısı gibi parametre ve özelliklere bağlı olarak değişmesi gibi dezavantajları da bulunmaktadır (Taşcı and Onan 2016).  $K$  en yakın komşu algoritmasında Şekil 2.13'te gösterildiği üzere sınıflandırma sırasında elde edilen özniteliklerden, sınıflandırılmak istenen yeni verinin daha önceki verilerden  $k$  tanesine yakınlığına bakılmaktadır ve yakınlık mesafelerinin hesaplanmasında genelde öklit mesafesi (euclid distance) kullanılmaktadır (Jain *et al.* 1995).



**Şekil 2.13.**  $K$  en yakın komşu algoritması (Jain *et al.* 1995)

## 2.6.2. Rastgele orman algoritması (Random Forest)

Karar ağaçları algoritmaları sınıflandırma, karar problemleri, kümeleme ve tahmin işlemlerinde kullanılmaktadır. Karar ağacı algoritmalarında kullanılacak verideki kategorik veya sürekli değişkenlerden yola çıkarak birçok farklı karar ağacı oluşturmak mümkündür. Bu algoritmalarda amaç en uygun ağacı oluşturmaktır. Breiman tarafından 1984 yılında karar ağacı temelli ve çok sayıda karar ağacı ile oluşturulan rastgele orman algoritması önerilmiştir (Breiman, 2001). Bu yöntemde Şekil 2.14'te gösterildiği üzere birden çok karar ağacının ortaya koyduğu sonuçlar bir araya getirilerek, bütün ağaçlar adına tek bir karar üretilmektedir. Rastgele orman algoritmasında sonradan gelen veriye ait tahmin yapılabilmesine ek olarak, değişkenlerin önem derecelerinin hesaplanması, model indirgemesi ve aşırı değerlerin tespiti de yapılabilmektedir.



**Şekil 2.14.** Rastgele orman algoritması

Rastgele orman algoritmasında, veri setindeki sınıf değeri kategorik bir değere sahip ise sınıflandırma, sürekli bir değere sahip ise regresyon ağaçları oluşturulmaktadır. Ormanda bulunan her bir karar ağacı, bootstrap yöntemi ile orijinal veri setinden örnek



seçilmesi ve her karar düğümünde tüm değişkenler içinden belirlenen sayıda rastgele değişkenin seçilmesi ile oluşturulmaktadır (Akman *et al.* 2011; Breiman *et al.* 1984).

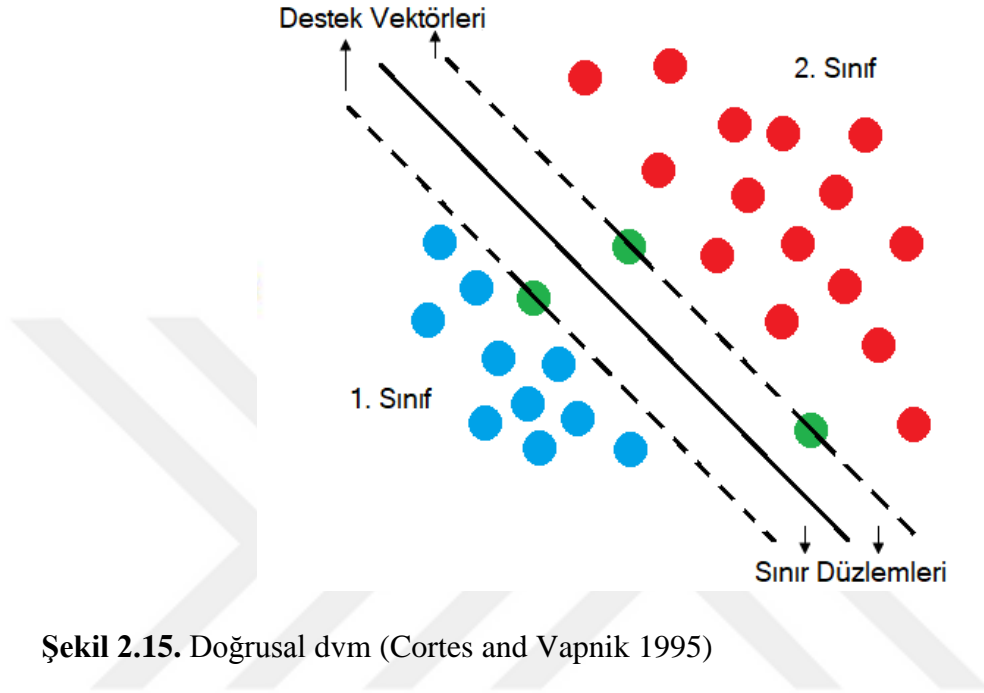
Rastgele orman yönteminde, veri setindeki en iyi özelliklerden seçilen düğümlerin dallara ayrılması yerine, her bir düğümde rastgele alınan özelliklerin en iyisi tercih edilerek tüm düğümler dallara ayrılmaktadır. Her veri grubu asıl veri setinden yer değiştirmeli olarak elde edilmektedir. Rastgele özellik seçimi kullanılarak ağaçlar geliştirilmekte ve budama işlemi uygulanmamaktadır (Breiman *et al.* 1984). Rastgele orman algoritmasının diğer algoritmalara göre daha hızlı ve doğru sonuçlar vermesinin sebebi bu aşamadır (Shi, 2007).

### **2.6.3. Destek vektör makineleri (Support Vector Machines)**

Destek Vektör Makineleri (DVM), örüntü tanıma ve sınıflandırma problemlerinde uygulanmak için Vapnik tarafından 1995 yılında önerilmiştir (Cortes and Vapnik 1995). DVM'in temeli istatistiksel öğrenme teorisi olan Vapnik-Chervonenkis (VC) teorisine dayanmaktadır. Doğrusal DVM, bir eğitim kümesi verildiğinde, farklı sınıflara ait örnekleri birbirinden en iyi şekilde ayıracak doğrusal hiper düzleminin bulunmasına yardımcı olan denetimli öğrenme algoritmaları sınıfına ait makine öğrenimi algoritmasıdır (Cortes and Vapnik 1995).

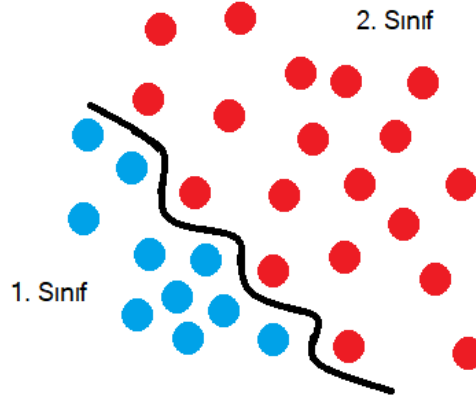
Destek Vektör Makinelerinin en büyük avantajı, sınıflandırma problemini kareli optimizasyon problemi şeklinde çözmesidir. Bu sayede problemin çözümü için uygulanan öğrenme aşamasında işlem sayısı azalmakta ve diğer algoritmalara göre daha hızlı sonuç elde edilmektedir. Yöntem bu faydasından dolayı, özellikle büyük boyutlu veri kümelerinde büyük avantaj sağlamaktadır. Ayrıca optimizasyon temeline dayandığından sınıflandırma performansı, hesaplama karmaşıklığı ve kullanılabilirlik açılarından diğer yöntemlere göre daha etkilidir. Veri setini ayıran düzlemlere ilişkin geometrik ifade Şekil 2.15'te gösterilmektedir. Şekilde görüldüğü gibi farklı sınıflara ait örnekleri birbirinden ayıran birçok doğrusal düzlem bulunabilmektedir. Ancak DVM,

farklı sınıflara ait destek vektörleri arasındaki uzaklığı maksimize eden ayırma hiper düzleminin bulunmasını amaçlamaktadır (Cortes and Vapnik 1995).



Şekil 2.15. Doğrusal dvm (Cortes and Vapnik 1995)

Destek vektörleri, ait olduğu sınıfın sınırını belirlemekte ve ayırma hiper düzlemine paralel bir düzlem üzerinde yer almaktadır. Destek vektörlerinin üzerinde bulunduğu ve kesikli çizgilerle gösterilmiş düzlemlere sınır düzlemleri denmektedir. Sınır düzlemlerinin tam ortasından geçen ve her iki düzleme de eşit uzaklıkta bulunan düzlem ise hiper düzlem olarak ifade edilmektedir (Cortes and Vapnik 1995). Doğrusal olmayan DVM, veri setinin doğrusal bir fonksiyonla tam veya belirli bir hata ile ayrılabilmesi durumunda kullanılan algoritmalarıdır. Gerçek yaşam problemlerinde bir veri setinin hiper düzlem ile doğrusal olarak ayrılması çoğunlukla mümkün değildir. Dolayısıyla sınıfları ayırma işlemi, ayırma eğrisinin tahmin edilmesiyle mümkün olmaktadır. Ancak uygulamada eğrinin tahmin edilmesi oldukça zordur (Cortes and Vapnik 1995). Veri setinin doğrusal ayrılabilme durumunun geometrik ifadesi Şekil 2.16'da gösterilmektedir.



**Şekil 2.16.** Doğrusal olmayan dvm (Cortes and Vapnik 1995)

DVM'ler günümüzde yüz tanıma sistemleri ve ses analizi gibi zor ve karmaşık birçok sınıflandırma probleminde kullanılmaktadır. Avantajları aşağıdaki şekilde listelenebilmektedir:

- Yüksek boyutlu uzaylarda etkilidir,
- Boyut sayısının, örneklem sayısından fazla olduğu durumlarda etkilidir,
- Karar fonksiyonunda bir takım eğitim noktaları kullanılmaktadır, dolayısıyla bellek verimli bir şekilde kullanılmış olmaktadır,
- Karar fonksiyonu için çok farklı çekirdek fonksiyonları kullanılabilir (Akpınar, 2014).

#### **2.6.4. Evrişimli sinir ağları**

Derin öğrenme, hiyerarşik mimarileri kullanarak verideki üst seviye soyutlamaları öğrenmeye çalışan makine öğrenmesi alt alanıdır (Guo *et al.* 2016). Geleneksel makine öğrenme teknikleri, doğal verileri ham formlarında işleme yetenekleriyle sınırlıdır (Girshick *et al.* 2014). Derin öğrenme yöntemleri, her biri temsili daha yüksek ve daha soyut düzeyde bir temsile dönüştüren, basit ama doğrusal olmayan modüller oluşturarak elde edilen çoklu temsil seviyelerine sahip temsil-öğrenme yöntemleridir. Derin öğrenmenin en önemli özelliği, öznetelik öğrenilen katmanların insanlar tarafından

tasarlanmamasıdır, genel amaçlı bir öğrenme prosedürü kullanılarak verilerden öğrenilmektedir. Derin öğrenme kullanımının son yıllardaki hızlı yükselişinin temel nedenleri işlemci kapasitelerinin artması, donanımların maliyetlerinin düşmesi ve makine öğrenmesi algoritmalarındaki gelişmeler olarak sıralanabilmektedir. Son yıllarda bilgisayarlı görü alanında derin öğrenme yoğun bir şekilde çalışılmış ve çok sayıda yaklaşım ortaya çıkmıştır. Bu yaklaşımlar türetildiği temel yöntemlere göre evrişimsel sinir ağları, kısıtlı Boltzmann Makinesi, Otomatik Kodlayıcı ve Seyrek kodlama olarak 4 kategoriden oluşmaktadır (Guo *et al.* 2016).

Evrişimsel Sinir Ağları (Convolutional Neural Networks) çoklu katmanların etkili bir şekilde eğitilerek kullanıldığı en dikkat çeken derin öğrenme yaklaşımlarından biridir (LeCun *et al.* 1998). Genel olarak bir evrişimsel sinir ağı Şekil 2.17’de görüldüğü üzere evrişimsel katmanlar, havuzlama katmanları ve tam bağlı katmanlardan oluşmaktadır. Evrişimsel sinir ağlarında ileri ve geri olmak üzere iki eğitme aşaması bulunmaktadır. İleri aşamanın asıl amacı girdi görüntüsünü, her katmandaki geçerli ağırlık ve eğilim parametreleriyle temsil etmektir. Elde edilen tahmin çıktısı, zarar maliyetini gerçek etiket değerleriyle hesaplamak için kullanılmaktadır. Geri aşama ise, zarar maliyetine göre, geriye doğru her parametrenin gradyanlarını zincir kuralı ile hesaplamaktadır. Tüm parametreler gradyanlara göre güncellenmekte ve bir sonraki ileri hesaplama için hazırlanmaktadır. İleri ve geri aşamaların yeterli yinelenmesinden sonra, ağ öğrenme durdurulmaktadır (Guo *et al.* 2016).



**Şekil 2.17.** Evrişimli sinir ağı mimarisi

Evrişimli sinir ağlarında evrişim, havuzlama ve tam bağlı olmak üzere 3 farklı katman bulunmaktadır. Evrişim katmanlarında, çeşitli çekirdekler kullanılarak görüntünün çeşitli öznitelik haritaları oluşturulmaktadır (Gu *et al.* 2018; Guo *et al.* 2016). Evrişim işleminin 3 ana avantajı bulunmaktadır. Bunlar, aynı öznitelik haritasında ağırlık paylaşma mekanizması sayesinde parametre sayısının azalması, yerel bağlantının komşu pikseller arasındaki ilişkiyi öğrenmesi ve elde edilen bu öznitelikler ile nesnenin konumuna göre değişmezlik kazanılmasıdır (Guo *et al.* 2016). Birinci evrişim katmanlarındaki çekirdekler kenarlar, eğriler gibi düşük seviyeli öznitelikleri tespit etmek için tasarlanırken, daha yüksek katmanlardaki çekirdekler daha soyut öznitelikleri çıkarmaktadır (Gu *et al.* 2018). Öznitelik haritası, girdiyi öğrenilmiş bir çekirdekle birleştirerek ve doğrusal olmayan bir aktivasyon işlevi uygulayarak elde edilmektedir. Matematiksel olarak,  $(i, j)$  konumunun  $m$ . katman  $n$ . öznitelik haritasının öznitelik değeri olan  $z_{i,j,n}^m$  aşağıdaki formül ile hesaplanmaktadır (Gu *et al.* 2018):

$$z_{i,j,n}^m = w_n^{mT} x_{i,j}^m + b_n^m \quad (5)$$

$w_n^m$  ve  $b_n^m$  sırasıyla  $m$ . katmanın  $n$ . filtresinin ağırlık vektörü (weight) ve sapma (bias) terimidir ve  $x_{i,j}^m$  ise  $m$ . katmanın  $(i,j)$  konumunda ortalanmış girdi parçasıdır (Gu *et al.* 2018).  $z_{i,j,n}^m$  'nin öznitelik haritasını oluşturan  $w_n^m$  çekirdeğinin paylaşıldığına dikkat edilmelidir. Böyle bir ağırlık paylaşma mekanizması model karmaşıklığını azaltma ve ağırlık eğitimi kolaylaştırma gibi avantajlar sağlamaktadır (Gu *et al.* 2018).

Aktivasyon fonksiyonu, doğrusal olmayan özniteliklerin algılanabilmesini sağlamaktadır. Evrişimli öznitelik  $z_{i,j,n}^m$  için aktivasyon değeri olan  $a_{i,j,n}^m$  aşağıdaki formülle hesaplanmaktadır:

$$a_{i,j,n}^m = a(z_{i,j,n}^m) \quad (6)$$

Uygulamalarda genellikle sigmoid, tanh ve ReLU aktivasyon fonksiyonları kullanılmaktadır. ReLU (Rectified Linear Unit), doğrultulmuş doğrusal ünite en önemli

aktivasyon fonksiyonlarından biridir. Aşağıdaki formülle hesaplanmaktadır,  $z_{i,j,k}$  k. kanaldaki (i,j) konumundaki aktivasyon girişidir (Gu *et al.* 2018):

$$a_{i,j,k} = \max(z_{i,j,k}, 0) \quad (7)$$

ReLU, negatif kısmı sıfıra doğru çeken, pozitif kısmı tutan parçalı doğrusal bir fonksiyondur. Basit maksimum operatörü olması nedeniyle sigmoid ve tanh aktivasyon fonksiyonlarından daha hızlı çalışmaktadır. Literatürde LReLU, PReLU, RReLU, ELU gibi versiyonları da bulunmaktadır (Gu *et al.* 2018).

Havuzlama katmanı genel olarak, evrişimli bir katmandan sonra gelmekte ve öznitelik haritalarının ve ağ parametrelerinin boyutlarını azaltmak için kullanılmaktadır (Gu *et al.* 2018; Guo *et al.* 2016). Örnek olarak 8x8 boyutlu bir öznitelik haritası, havuzlama katmanına girerek 4x4'lük bir öznitelik haritasına dönüştürülmektedir. Evrişim katmanına benzer şekilde hesaplamalara komşu pikselleri de kattığı için havuzlama katmanları da dönmeye karşı dayanıklıdır. Ortalama ve maksimum havuzlama en sık kullanılan yöntemlerdir. Yapılan çoğu çalışmada, daha hızlı yakınsama, daha etkili değişmez öznitelik seçme ve genelleştirmeyi geliştirmesi sebebiyle maksimum havuzlama yöntemi tercih edilmektedir (Guo *et al.* 2016).  $a_{i,j,n}^m$  öznitelik haritası için havuzlama fonksiyonu aşağıdaki formül uygulanmaktadır:

$$y_{i,j,n}^m = \text{pool}(a_{f,g,n}^m), \forall (f,g) \in R_{ij} \quad (8)$$

$R_{ij}$ , (i,j) konumunun yerel komşuluklarına ifade etmektedir (Gu *et al.* 2018). Havuzlama katmanı olarak Stokastik (Olasılıksal, Tahmini veya Rastgele), Mekânsal Piramit, Deformasyon olmak üzere 3 havuzlama katmanı uygulaması bulunmaktadır. Maksimum havuzlamanın, eğitim setine aşırı uydurulmasına karşı hassas olması ve genelleştirilmesinin zor olması gibi dezavantajları bulunmaktadır. Bu aşırı uydurma sorunu çözmek için her havuzlama bölgesi içindeki aktivasyonu rastgele bir şekilde toplayıp, çok amaçlı bir dağılıma göre seçerek, geleneksel havuzlama operasyonlarını

stokastik bir prosedürle değiştiren bir havuzlama yaklaşımı önerilmiştir (Zeiler and Fergus 2013). Mekânsal piramit havuzlama ise son havuzlama katmanını uzamsal bir piramit havuzlama katmanı ile değiştirerek, sabit boyutlu bir giriş görüntüsü gerektiren evrişimli sinir ağlarında, isteğe bağlı boyutlardaki görüntüler için basit uzunlukta temsiller çıkarılması amacıyla önerilmiştir (He *et al.* 2015). Deformasyonun ele alınması bilgisayarlı görüde nesne tanıma uygulamaları için temel bir zorluktur. Maksimum ve ortalama havuzlama deformasyonunun işlenmesinde yararlıdır ancak nesne parçalarının deformasyon sınırlamalarını ve geometrik modelini öğrenemezler. Bu problemin çözülmesi amacıyla def-havuzlama katmanı kullanılması önerilmiştir (Ouyang *et al.* 2014). Bu havuzlama işleminde görsel modellerin deformasyonu öğrenilerek derin model zenginleştirilmektedir (Guo *et al.* 2016).  $L_p$  Havuzlama, Karma Havuzlama, Olasılıksal Havuzlama, Hayali Havuzlama ve Mekânsal Piramit Havuzlama gibi farklı havuzlama teknikleri de bulunmaktadır (Gu *et al.* 2018).

Tam Bağlı Katmanlar (Fully Connected Layers) son havuzlama katmanının ardından 2 boyutlu öznitelik haritalarının daha fazla öznitelik gösterimi için 1 boyutlu öznitelik vektörüne dönüştüren birkaç tam bağlı katman olarak kullanılmaktadır. Bu katmanlar geleneksel bir sinir ağı gibi performans göstermekte ve bir evrişimli sinir ağındaki parametrelerin yaklaşık % 90'ını içermektedir. Sinir ağını önceden tanımlanmış bir uzunluğa sahip bir vektöre ilerletmemizi sağlamaktadır. Bu katmanların en önemli dezavantajı, pek çok parametre içermeleridir, bu da onların, eğitilmek için büyük bir hesaplama işlemi gerektirmesine neden olmaktadır (Guo *et al.* 2016). Birkaç evrişim ve havuzlama katmanından sonra yüksek seviyeli mantık yürütmeyi amaçlayan bir veya daha fazla sayıda tam bağlı katman kullanılabilir. Bu katmanda önceki katmanın çıktıları mevcut katmandaki bütün nöronlara bağlanmaktadır (Gu *et al.* 2018).

Evrişimli sinir ağlarının çıktı katmanı olan son katmanında, sınıflandırma görevi için genellikle softmax operatörü veya CNN ile birleştirilebilen DVM kullanılmaktadır. Belirli bir görev için en iyi parametreler, bu görevde tanımlanan bir kayıp fonksiyonunun en aza indirilerek elde edilebilmektedir.  $N$  adet istenen girdi-çıkışı ilişkisi  $(x^{(n)}, y^{(n)}); n \in [1, \dots, N]$  için,  $x^{(n)}$ ,  $n$ . girdi verisidir,  $y^{(n)}$  hedef etiketleridir ve  $o^{(n)}$  ise

CNN'in çıkışıdır. Bu ilişkilerin CNN kaybı aşağıdaki formülle hesaplanmaktadır (Gu *et al.* 2018):

$$L = \sum_{n=1}^N l = (\theta; y^{(n)}, o^{(n)}) \quad (9)$$

Kayıp fonksiyonu hesabı için literatürde Hinge, Softmax, Contrastive ve Triplet gibi yöntemler de bulunmaktadır (Gu *et al.* 2018) . Küresel bir optimizasyon problemi olan evrişimli sinir ağı eğitiminde, kayıp fonksiyonunu en aza indirerek en uygun parametrelerin bulunmaktadır. Bu amaçla Olasılıksal Dereceli Azalma (Stochastic gradient descent) yönteminin kullanımı yaygındır (Gu *et al.* 2018).

Derin olmayan öğrenme ile karşılaştırıldığında, derin öğrenme daha soyut bilgileri öğrenmek için derin mimariler inşa etmesi nedeniyle daha avantajlıdır. Bununla birlikte, çok fazla parametre girilmesi, aşırı eğitim problemine neden olmaktadır. Dropout yöntemi, her eğitim aşamasında, karmaşık eş uyarlamaları önlemek ve genelleme yeteneğini arttırmak için öznelik detektörlerinin yarısının atlanmasıdır. Dropconnect ise aktivasyonlar yerine ağırlıkların rasgele düşürüldüğü yöntemdir (Guo *et al.* 2016).

Evrişimli sinir ağı görsel nesne tanıma uygulandığında, ekstra etiketleme maliyetleri getirilmeden daha fazla veri üretmek için veri artırma işlemi kullanılmaktadır. Örnek olarak AlexNet ağına birincisi görüntü dönüşümleri ve yatay yansımalar oluşturma, ikincisi ise RGB kanallarının yoğunluğunun değiştirilmesinden ibaret olan iki veri artırma yöntemi kullanılmıştır (Guo *et al.* 2016).

Ön-eğitim, ağı rastgele ayarlanmış parametreler yerine, önceden eğitilmiş parametrelerle başlatmak demektir. Öğrenme sürecini hızlandırabileceği ve genelleme yeteneğini arttırabileceği avantajları bulunduğundan evrişimli sinir ağlarına dayalı modellerde tercih edilmektedir. Alexnet ağı ile yapılan çalışmalarda eğitilmiş parametreler ve belirli görevler için hassas-ayarlılama yöntemi ile parametrelerin



belirlenmesi diğer benzer modeller olan Clarifai, GoogLeNet ve VGG'ye göre daha iyi performans sağlamaktadır (Guo *et al.* 2016).

Evrişimli sinir ağlarının popüler hale gelmesi, 5 evrişim, 3 tam bağlı katmandan oluşan AlexNet'in ImageNet yarışmasında birinci olması ile başlamıştır. Bu mimarinin, sabit 224x224 piksel boyutlu görüntüler alması ve neden iyi performans gösterdiğinin anlaşılması gibi dezavantajları bulunmaktadır. Bu çalışmadan sonra Clarifai isimli, ara öznitelik katmanları için yeni bir görselleştirme tekniği içeren ve AlexNet'ten daha iyi sonuç veren mimari geliştirilmiştir. Bir başka çalışmada ise sabit görüntü boyutu kısıtını ortadan kaldırmak için yeni bir havuzlama stratejisi olan mekânsal havuzlama stratejisi içeren mimari önerilmiştir (He *et al.* 2015). Daha sonra önerilmiş VGG isimli mimaride ise, daha fazla evrişimli katman eklenerek, tüm evrelerde küçük evrişim filtrelerinden yararlanarak sinir ağının derinliğinin artırılmıştır. Bu gelişmeden sonra 22 katmanı bulunan daha derin bir evrişimli sinir ağı olan GoogLeNet önerilmiştir. Geliştirilen evrişimli sinir ağları model ve uygulamaları, üst düzey sınıflandırma performansları elde edilmesinden sonra, bu modeller ile nesne tespit etme ve anlamsal bölütleme işlemlerinin yapılabilmesi amacıyla yeni yapılar türetilmiştir. En yaygın olanları RCNN(Region based Convolutional Neural Network) olarak ifade edilen bölge tabanlı evrişimli sinir ağları ve FCN(Fully Convolutional Network) tamamen evrişimli ağıdır (Guo *et al.* 2016).

RCNN'nin ana fikri, bölgesel olarak çoklu nesne önerileri oluşturmak, evrişimli sinir ağları kullanarak bu önerilerin özniteliklerini çıkarmak ve bu pencereleri doğrusal SVM ile sınıflandırmaktır (Girshick *et al.* 2014). Öncelikle seçici arama (selective search) kullanılarak görüntüden 2000 adet bölge teklifi çıkarılmaktadır. Bölge tekliflerinin 4096 boyutlu özniteliklerin çıkarılması için 5 evrişim katmanı ve 2 tam bağlı katman uygulanmaktadır. Sonrasında ise bu bölgelere ait öznitelik vektörleri doğrusal SVM kullanılarak her sınıf için puanlanmaktadır. Bölge önerilerinde üst üste binen pencerelerin elenmesi için IoU(Intersection over Union) birleşimlerin kesişimi eşik değeri uygulanmaktadır (Girshick *et al.* 2014).

Bu bölge kullanarak sınıflandırma yaklaşımı sinir ağları tabanlı nesne tespit probleminde genel kullanım olarak uygulanmıştır. Ancak yaklaşımın performansının konumlandırma tahminine dayanması ve çok sayıda öneri üretilmesi ve işlenmesinin zor olması gibi dezavantajları bulunmaktadır (Guo *et al.* 2016) . RCNN, CNN modellerini öznelik çıkarıcı olarak almaktadır ve ağlarda herhangi bir değişiklik yapılmamaktadır. Buna karşılık, FCN, CNN modellerini tamamen evrişimli ağlar olarak yeniden yapılandırmak için bir teknik önermektedir. Bu yeniden yapılandırma tekniği sayesinde CNN ağı görüntü boyutu sınırlandırma kısıtı ortadan kaldırılabilir (Guo *et al.* 2016).

## 2.7. Öznelik Çıkarma ve Öznelik Eşleştirme

Karmaşık gerçek dünya görüntülerinde cisim tespit etme ve cisim tanıma işlemi, görüntüdeki karmaşıklık veya kısmi tıkanıklıktan etkilenmeyen yerel görüntü öznelikleri kullanmayı gerektirmektedir. Yerel özneliğin, tekrarlanabilirlik, ayırt edilebilirlik, yerellik, yeterli miktar, doğruluk ve verimlilik gibi özelliklere sahip olması gerekmektedir (Tuytelaars and Mikolajczyk 2008). Bu öznelikler, aydınlatma değişikliğine, 3B dönüşümlere ve ortak cisim varyasyonlarına karşı değişmezlik sahibi olmalıdır. Öte yandan, öznelikler farklı cisimleri tanımlayabilmesi için ayırt edici olmalıdır (Lowe, 1999). Yerel değişmez öznelikler, yalnızca görüntüleme koşullarındaki değişimlere, tıkanma veya görüntü dağınıklığına rağmen görüntüdeki benzerliklerin bulunmasını sağlamakla kalmamakta, aynı zamanda görüntü içeriği ile ilgili tanımlama yapabilmektedir (Tuytelaars and Mikolajczyk 2008).

Bilgisayar ortamında ve cisim tespitinde, görüntüdeki öznelikler çıkarılarak görüntünün soyutlanması ve görüntüde belirli bir tipte bir özneliğin bulunup bulunmadığının hesaplanarak yerel kararların alınabilmesi için yöntemler bulunmaktadır. Ortaya çıkan öznelikler, genellikle noktalar, sürekli eğriler veya bağlı bölgeler biçiminde görüntü alanının alt kümeleri olmaktadır (Lindeberg, 1998).

Bir öznitelik bir görüntünün "ilginç" bir parçası olarak tanımlanmaktadır ve öznitelikler birçok bilgisayarlı görü algoritması için başlangıç noktası olarak kullanılmaktadır. Öznitelikler sonraki algoritmalar için başlangıç noktası olarak kullanıldığı için, genel algoritma genellikle sadece öznitelik dedektörü kadar iyi olmaktadır. Sonuç olarak, bir öznitelik detektörünün istenen özelliği, aynı görüntünün aynı sahnenin iki veya daha fazla farklı görüntüsünde tespit edilip edilmeyeceğini ifade eden tekrarlanabilirliktir. Öznitelik algılama, düşük seviyeli bir görüntü işleme işlemidir. Yani, genellikle bir görüntü üzerinde ilk işlem olarak gerçekleştirilmekte ve bu pikselde mevcut bir öznitelik olup olmadığını tespit etmek için her pikseli incelenmektedir. Çok sayıda öznitelik detektörü geliştirilmiştir. Bunlar tespit edilen özellik çeşitleri, hesaplama karmaşıklığı ve tekrarlanabilirlik açısından geniş çapta farklılık göstermektedir (Lindeberg, 1998).

Görüntü öznitelikleri kenarlar, köşeler ve ilgi noktaları ile kabarcıklar olabilmektedir. Kenarlar, iki görüntü bölgesi arasında sınır veya kenar bulunan noktalardır. Genel olarak bir kenar, keyfi bir şekle sahip olabilmektedir. Pratikte, kenarlar genellikle görüntüde güçlü bir gradyan büyüklüğüne sahip olan nokta kümeleri olarak tanımlanmaktadır. Bazı algoritmalar genellikle bir kenarın özellikleri üzerinde, şekil, düzgünlük ve indirgeme değeri gibi bazı kısıtlamalar getirmektedir. Yerel olarak kenarlar tek boyutlu bir yapıya sahiptir. Köşeler ve ilgi noktaları terimleri birbirinin yerine kullanılmaktadır ve yerel iki boyutlu bir yapıya sahip olan bir görüntüdeki noktaya benzer özellikler anlamına gelmektedir. Köşeler, kenar algılama işlemi gerçekleştirildikten sonra, kenarlardaki ani değişikliklerin tespit edilmesi ile ortaya çıkmıştır. Daha sonra geliştirilen algoritmalar sayesinde açık kenar tespiti yapmak yerine yüksek eğrilik seviyelerinin belirlenmesi sağlanmıştır. Daha sonra, köşelerin, geleneksel anlamda köşeler olmayan, görüntünün siyah zemin üstünde küçük parlak nokta gibi kısımlarında da tespit edildiği fark edildiğinden bu noktalar ilgi noktaları olarak tanımlanmaktadır. Kabarcıklar, bölgelere göre görüntü yapılarının tamamlayıcı bir tanımını sağlamaktadır. Bununla birlikte, kabarcık tanımlayıcıları sıklıkla tercih edilen bir noktayı içerebilmektedir. Bu da birçok kabarcık detektörünün aynı zamanda ilgi noktası operatörleri olarak kabul edilebileceği anlamına gelmektedir. Kabarcık

detektörleri, bir görüntüdeki alanları bir köşe detektörü tarafından algılanmayacak kadar düzgün algılayabilmektedir. Kabarcıkların algılanması için LoG, DoG ve DoH algoritmaları kullanılmaktadır. Çizelge 2.3'te öznelik detektörlerinin kenar, köşe veya kabarcık algılama durumları ifade edilmektedir.

**Çizelge 2.3.** Öznelik detektörleri ve özellikleri (Canny, 1987; Harris and Stephens 1988; Lindeberg, 2007)

Öznelik Detektörü	Kenar	Köşe	Kabarcık
Canny	X		
Sobel	X		
Kayyali	X		
Harris and Stephens	X	X	
Susan	X	X	
Shi and Tomasi		X	
Seviyeli Eğilim		X	
FAST		X	X
Gaussların Laplacianı		X	X
Gaussların Farkı		X	X
Hessianların Determinantı		X	X
MSER			X
PCBR			X
Gri Ton Kabarcıklar			X

Bu öznelikler bölge sınırları, renk veya doku gibi görüntülerdeki herhangi bir değişikliğe karşılık gelmek zorunda değildir. Ayrıca birden fazla bölge üst üste gelebilmekte ve homojen alanlar gibi görüntünün ilgisiz kısımları açığa çıkabilmektedir (Tuyltaars and Mikolajczyk 2008). Yerel özneliklerin özellikleri şu şekilde listelenebilmektedir:

- Tekrarlanabilirlik: Farklı görüntüleme koşullarında çekilen aynı cismin aynı özneliklere sahip olmasıdır.
- Çeşitlilik: Algılanan özneliklerin yoğunluk desenleri, özneliklerin ayırt edilebileceği, eşleştirilebileceği şekilde çok çeşitli olmalıdır.
- Yerellik: Bu öznelikler tıkanma/oklüzyon olasılığını azaltmak için, farklı görüntüleme koşullarında alınan iki görüntü arasındaki geometrik ve fotometrik deformasyonların olmadığı tek model tahminine izin vermelidir.
- Miktar: cismin daha iyi tespit edilebilmesi için öznelik sayısı yeterince fazla olmalıdır.
- Doğruluk: tespit edilen öznelikler, hem görüntü konumunda hem de ölçek ve olası şekilde göre doğru şekilde yerleştirilmelidir.
- Verimlilik: Tercihen yeni bir görüntüdeki özneliklerin çıkarılması süreci önemlidir. Özneliklerin algılanma süresi zaman açısından kritik öneme sahiptir.

Tekrarlanabilirlik değişmezlik (invariance) ve dayanıklılık (robustness) olarak ikiye ayrılmaktadır. Değişmezlik, büyük deformasyon beklendiğinde, yaklaşım mümkünse matematiksel olarak modellemek ve sonra bu dönüşümden etkilenmeyen öznelik tespiti için yöntem geliştirilmesidir. Dayanıklılık ise, küçük deformasyon durumunda daha az duyarlı öznelik tespitinin yapılmasıdır. Bu küçük deformasyonlar gürültü, ayrıklaştırma etkileri, sıkıştırma etkileri, bulanıklaştırma olabilmektedir. Ayrıca değişmezlik elde etmek için uygulanan matematiksel model de sağlamlığı arttırmaktadır (Tuyltaars and Mikolajczyk 2008).

Öznelik detektörleri değişmezlik seviyesine göre ayırt edilebilir. En fazla değişmezliğe sahip yöntem cazip olabilir ancak bu özneliklerin ayırt etme gücünü azaltmaktadır. Daha fazla parametre daha fazla gürültüye neden olur. Ayrıca öznelik saptama işi daha karmaşık hale gelir ve bu karmaşıklığı ve tekrarlanabilirliği etkiler. Uygulamada ihtiyaç duyulandan daha fazla değişmezlik kullanılmaması temel bir kuraldır. Detektörlerin niteliksel özellikleri, uygulama senaryosuna bağlı olarak özneliklerin bazıları diğerlerinden daha önemlidir. Örnek, kategori düzeyi tanımlamada küçük görünüm değişikliklerine karşı dayanıklılığı önemlidir, 3d modellemede yerleştirme doğruluğu

önemlidir. Büyük miktardaki veride çevrimiçi çalışma için verimlilik önemlidir gibi (Tuytelaars and Mikolajczyk 2008).

Öznitelikler değişmezliklerine göre 4 grupta incelenmektedir. Rotasyon, benzerlik, dönme, perspektif (Rotation, smilarity, affine, perspektive) (Tuytelaars and Mikolajczyk 2008). Çizelge 2.4'te öznitelik detektörlerinin dönme, ölçek ve afin değişmezliklerine göre durumu gösterilmektedir.

**Çizelge 2.4.** Öznitelik detektörlerinin değişmezlikleri (Tuytelaars and Mikolajczyk 2008)

<b>Öznitelik Detektörü</b>	<b>Dönme Değişmezliği</b>	<b>Ölçek Değişmezliği</b>	<b>Afin Değişmezliği</b>
Harris	X		
Hessian	X		
Susan	X		
Harris-Laplace	X	X	
Hessian-Laplace	X	X	
DoG	X	X	
SURF	X	X	
Harris-Affine	X	X	X
Hessian-Affine	X	X	X
Belirgin Bölgeler	X	X	X
Kenar Tabanlı	X	X	X
MSER	X	X	X
Yoğunluk Tabanlı	X	X	X
Superpixel	X	X	X

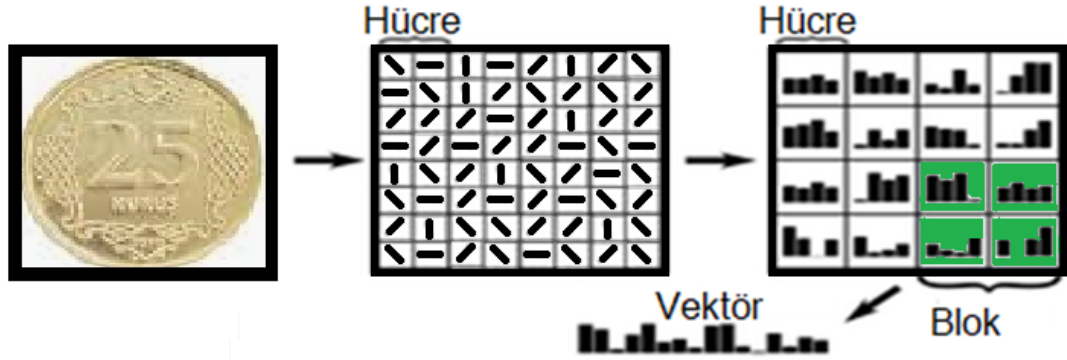
Öznitelik tespit edildikten sonra, özellik etrafındaki yerel bir görüntü yaması çıkarılma işlemine ihtiyaç duyulmaktadır. Bu çıkarma işlemi, oldukça fazla miktarda görüntü işleme aşaması içerebilmektedir. Bir öznitelik tanımlayıcı veya öznitelik vektörü olarak

bilinmektedir (Lindeberg, 1998). Karmaşık gerçek dünya sahnelerinde cisim tanıma, karmaşıklık veya kısmi tıkanıklıktan etkilenmeyen yerel görüntü öznitelikleri gerektirmektedir. Bu öznitelikler, en azından kısmen aydınlatma değişikliklerine, 3D yansıtımlı dönüşümlere ve ortak cisim varyasyonlarına göre değişmez olmalıdır. Öte yandan, öznitelikler birçok alternatif arasında belirli cisimleri tanımlamak için yeterince ayırt edici olmalıdır. Cisim tanıma probleminin zorluğu, büyük ölçüde, bu tür görüntü özelliklerinin bulunmasında başarı eksikliğine bağlıdır.

Aynı sahnenin veya cismin iki görüntüsü arasında noktasal ilişki bulunması görevi, cisim tespit etme, cisim tanıma, görüntü kaydı, kamera kalibrasyonu, görüntü alma gibi birçok bilgisayar görü uygulamasının bir parçasıdır. Ayırık görüntü noktası benzerliğini tespit etmek için 3 aşama uygulanmaktadır. İlk olarak, görüntüdeki farklı konumlarda 'ilgi noktaları', köşeler, lekeler tespit edilmektedir. Ardından, her ilgi noktasının bulunduğu komşuluk değerleri bir öznitelik vektörüyle temsil edilmektedir. Bu tanımlayıcı, ayırt edici ve aynı zamanda gürültü, algılama yer değiştirmeleri ve geometrik ve fotometrik deformasyonlar için sağlam olmalıdır. Son olarak, tanımlayıcı vektörler farklı görüntüler arasında eşleştirilmektedir. Eşleşme, vektörler arasındaki mesafeye dayanmaktadır.

### **2.7.1. HOG (Eğimlerin yönelim histogramı)**

HOG (Histogram of Oriented Gradients) öznitelik dönüşüm yöntemi, Şekil 2.18'de gösterildiği üzere sobel filtrelerinin görüntüye yatay ve dikey uygulanarak, her hücrenin yerel histogramı üzerinden eğim yönü ve kenar yönelimleri belirlendiği bir öznitelik çıkarma yöntemidir.



**Şekil 2.18.** HOG öznitelik dönüşümü (Dalal and Triggs 2005).

Bu özniteliklerde görüntü yerel histogramlar şeklinde tanımlanmaktadır. Bu histogramlar, görüntünün yerel bir parçasındaki gradyanların oryantasyonlarının sayılarını içermektedir. HOG özniteliklerinin çıkarılması için görüntünün gradyanın çıkarılması, tespit edilen konumlar için oryantasyon histogramlarının hesaplanması, oluşturulan konum bölgeleri içindeki histogramların normalize edilmesi gibi işlemler uygulanmaktadır. Gradyan çıkarma görüntünün iki boyutlu filtreler kullanılarak filtrelenmesi ile elde edilmektedir. Birçok yöntemde, gradyan çıkarmada yatay ve dikey filtreler bir arada kullanılması tercih edilmektedir. Bu yöntem uygulanmadan önce görüntü gri seviye haline dönüştürülmektedir. Gri seviye görüntüye ilgili filtre çekirdeği ile evrişim (konvolüsyon) işlemi uygulanmaktadır. Genelde filtre olarak yatay (Denklem 10) ve dikey sobel (Denklem 11) filtreleri tercih edilmektedir. Görüntünün sobel filtreleri ile konvolüsyon işlemine tabi tutulmasından sonra  $I_x$  ve  $I_y$  ile gösterilen görüntüdeki yatay ve dikey kenarlar çıkarılmaktadır (Dalal and Triggs 2005).

$$I_x = I * S_y \quad (10)$$

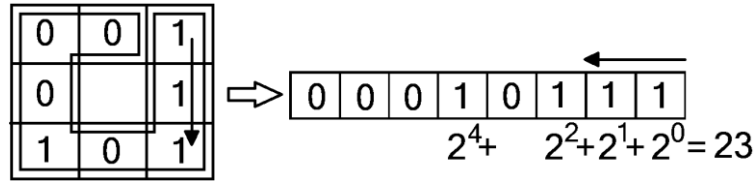
$$I_y = I * S_d \quad (11)$$

### 2.7.2. LBP (Yerel ikili örüntü)

LBP (Local Binary Pattern) yöntemi, bir görüntüdeki bütün pikselleri, eşik değere göre etiketleyen ve ikili sayıya dönüştüren bir örüntü operatörüdür (Ojala *et al.* 2000). LBP



yöntemi uygulanırken Şekil 2.19’da gösterildiği üzere  $r$  yarıçap içerisinde bulunan örnek sayısı  $n$  ve 0-1 dönüşüm sayısı ayarlanması gerekmektedir. Algoritma "uniform" parametresi ile birlikte kullanılarak görüntüden çıkan öznelik sayısı azaltılabilmektedir ve dönmeye karşı dayanıklılık elde edilebilmektedir.



Şekil 2.19. LBP algoritması

LBP yerel örüntüleri temsil etmede kullanılan, parametrik olmayan, ayırt etme gücü yüksek örüntü analizi yöntemidir. LBP bir pikselin komşu pikseller ile arasındaki sıralı ikili karşılaştırmalardan hesaplanmaktadır. LBP işlemi bir görüntüdeki her pikseli, kendisini çevreleyen 3x3'lük komşu bölgesindeki komşu piksellerini, merkezdeki piksel değerine göre eşikleyerek hesaplamaktadır. Eğer komşu pikselin değeri merkezdeki pikselden büyükse veya eşitse komşu piksel 1, küçükse 0 değeri olmaktadır. Böylece komşuluklar için 8-bitlik bir LBP vektörü elde edilmektedir. Bu vektörün onluk sistemdeki değeri ise merkez pikseli çevreleyen 3x3'lük bölgedeki yerel örüntüyü ifade etmektedir. LBP'nin matematiksel gösterimi aşağıdaki Denklem 12 ve Denklem 13'te bulunmaktadır (Tekeli, 2006; Ojala *et al.* 2000):

$$LBP(x) = \sum_{i=1}^8 s(G(x_i) - G(x))2^{i-1} \quad (12)$$

$$s(t) = \begin{cases} 1, & \text{and } t \geq 0, \\ 0, & \text{and } t < 0 \end{cases} \quad (13)$$

Burada  $x$  merkez pikselin konumu,  $x_i$   $i$  indisli komşu pikselin konumu ve  $G(\cdot)$  piksel yoğunluk değeridir. Bu yöntemi uygulayarak piksel yoğunluk değerleri 0 ile 255 değerleri arasında değişen LBP örüntüsü oluşturulmaktadır. Her LBP değeri farklı bir

örüntüyü temsil etmektedir. LBP değeri histogramı ise bir desenin içinde 256 farklı örüntünün her birine ne kadar sıklıkla rastlandığını göstermektedir (Tekeli, 2006).

### 2.7.3. Renk histogramı

Görüntü içeriğini tanımlamak için yaygın bir yöntem, renk histogramlarını kullanmaktır. Bir görüntü için renk histogramı, her rengin piksel sayısını sayarak oluşturulur. Renk özneliği çıkarmak için, RGB uzayından dönüşümü doğrusal olmayan ancak kolayca tersine çevrilebilir olan HSV renk uzayını kullanılmaktadır. HSV renk alanı bir koni olarak görüntülenebilmektedir. Uzun eksen değeri temsil etmekte: beyazdan siyaha doğru gitmektedir. Eksenden gelen mesafe mevcut renk miktarı olan doygunluğu temsil etmektedir. Eksenin çevresindeki açı tonu ifade etmektedir. Ton dairesi, 120 derece ayrılmış kırmızı, yeşil ve maviden oluşmaktadır. 20 derecelik basamaklarda dairesel bir niceleme, tonları, üç ana renk ve sarı, mor ve turkuaz her biri üç alt bölüme temsil edecek şekilde yeterince ayırmaktadır. Doygunluk ve değer, her biri, bu boyutlar boyunca daha büyük algısal tolerans sağlayan üç seviyeye nicelenmektedir (Smith and Chang 1995). Şekil 2.20’de örnek görüntüye ait renk histogramı gösterilmektedir.



Şekil 2.20. Renk histogramı

#### 2.7.4. SIFT (Ölçek değişmez öznitelik dönüşümü)

Ölçek Değişmez Öznitelik Dönüşümü (SIFT – Scale Invariant Feature Transform), 2004 yılında Lowe tarafından geliştirilen bir öznitelik detektörüdür (Lowe, 1999). SIFT detektörü ve tanımlayıcısı nesne tanıma (Lowe, 1999), görüntü birleştirme, görsel haritalama, vb. görsel özellikleri kullanan bir dizi uygulamada oldukça başarılı olduğu kanıtlanmıştır (Rublee *et al.* 2011). SIFT'in nesne tanıma uygulamalarında çok etkili olduğu kanıtlanmış olsa da, büyük bir hesaplama yükü gerektirmesi gibi özellikle gerçek zamanlı uygulamalar için dezavantaja sahiptir (Lowe, 1999; Rublee *et al.* 2011).

Her SIFT özneliği, görüntü dönüştürme, ölçekleme ve rotasyona göre değişmeyen ve aydınlatma ve yerel görüntü deformasyonlarındaki değişikliklere kısmen değişmeyen bir şekilde yerel görüntü ölçümlerinin bir vektörünü temsil etmektedir (Lowe, 2004). SIFT algoritmasının 4 temel adımı vardır: 1) tepe noktası seçimi, 2) anahtar nokta yerelleştirilmesi ve elenmesi, 3) yönlendirme ataması ve 4) anahtar noktaların tanımlanmasıdır. Birinci aşamada, Gaussian'ın Farkı (DoG) kullanılarak bir ölçek uzayı hesaplanmaktadır. Bu aşamada, Gaussian'ların farkının maksimum veya minimumu olan yerleri arayarak ölçek alanındaki anahtar konumları tanımlanmaktadır. İkinci aşama, anahtar noktaların (keypoint) yerelleştirildiği ve adayların düşük karşıtlığa sahip noktalarının, iyi betimleyici özellikleri olmadığı için elendiği işlemdir. Üçüncü aşama, yerel görüntü eğimi temeline dayanan anahtar noktalara bir yönelim değeri atanmasıdır. Yönelim histogramlarına ait en yüksek bir veya bir kaç tepelik öznitelik noktası için yönelim vektörü olarak atanmaktadır. Bu aşamada gradyan yönelimlerinin histogramı kullanılarak dönmeye karşı değişmezlik kazandırılmaktadır. Son aşama ise, görüntü eğim büyüklüğü ve yön belirlenmesine dayanan her bir anahtar noktası için yerel görüntü tanımlayıcısının hesaplanmasıdır. Her bir öznitelik noktası için, etrafındaki 4x4'lük bölgede 8 bölmeli bir histogram elde edilerek 128 (4x4x8) boyutlu bir tanımlayıcı belirlenmektedir. Bu 128 boyutlu vektör, birim uzunluğa dönüştürülerek aydınlatma değişimine karşı kısmen dayanıklılık elde edilmektedir. Ayrıca SIFT öznelikleri, görüntü gradyanı konumlarını bulanıklaştırarak, afin veya 3D görünüm gibi yerel çeşitliliklere kısmi değişmezlik sağlamaktadır (Lowe, 1999).

### 2.7.5. SURF (Hızlandırılmış sağlam öznitelikler)

SURF (Speeded Up Robust Features), Hessian matrisine dayanan bir ölçek-değişmez öznitelik detektörüdür. SIFT'in hızlandırılmış bir yaklaşımı olan SURF tekniği, tespit edilen noktaların kalitesini düşürmeden SIFT'den daha hızlı çalışmaktadır (Bay *et al.* 2006; Karakuş and Karabörk 2014). Konum ve ölçeği seçmek için farklı bir ölçü kullanmak yerine, her ikisi için de Hessian detektörü kullanılmaktadır. Hessian matrisi sayısal bir alanın ikinci dereceden kısmi türevlerinden oluşan kare matristir ve kabaca bir dizi kutu tipi filtrenin kullanılmasına benzemektedir (Tuytelaars and Mikolajczyk 2008). LoG (Laplacian of Gaussian) kullanılarak, ikinci dereceden türevi alınan görüntüde, ölçek uzayında değişmeyen anahtar noktalar elde edilmektedir. SURF'te SIFT'ten farklı olarak, çok işlem gerektiren ikinci dereceden Gauss türevleri yerine, bir kutu filtre kullanılmaktadır ve bu kutu filtredeki hesaplama işlemi, bütünleşik görüntü kullanılarak kolayca yapılabilir (Tuytelaars and Mikolajczyk 2008). Öznitelik tanımlayıcısı, ilgi noktasını tanımlamak için 20x20 piksellik komşu bölgelerin yatay ve dikey yönde Haar dalgacık tepkilerinin dağılımı kullanılmaktadır. Baskın yönelim, 60 derecelik bir kaydırma penceresindeki tüm tepkilerin toplamı hesaplanarak tahmin etmektir. Dalgacık tepkisi, herhangi bir ölçekte çok kolay bir şekilde bütünleşik görüntüler kullanılarak bulunabilmektedir. Bütünleşik görüntü, görüntüdeki piksel değerlerinin dikdörtgen ızgaralar şeklinde hızlı ve etkili olarak toplanabilmesi için kullanılan veri yapısıdır. Ayrıca SURF yönteminde, Laplacian'ın işaretine dayanan eşleşme hızını azaltan ve sağlamlığı arttıran bir indeksleme yöntemi kullanılmaktadır (Bay *et al.* 2006). Bu indekslemede, temel ilgi noktası için Laplacian (Hessian Matrix) izi kullanılmaktadır. Bu değerler algılama sırasında zaten hesaplandığından ek hesaplama maliyeti getirmemektedir. Laplacian'ın işareti, koyu arka plandaki parlak lekeleri ters durumlardan ayırmaktadır. Eşleştirme aşamasında, ilgi noktasının yalnızca aynı türde kontrast özellikleri varsa özellikleri karşılaştırılmaktadır. Bu minimum bilgi, tanımlayıcının performansını düşürmeden daha hızlı eşleştirilmesine olanak tanımaktadır. SURF, bulanıklaştırma ve döndürmeye karşı dayanıklı olmasına rağmen, bakış açısı değişikliği ve aydınlatma değişimine karşı dayanıklılık içermemektedir (Bay *et al.* 2006).

### 2.7.6. BRIEF (İkili sağlam bağımsız temel öznitelikler)

SIFT'in tanımlayıcılar için 128-boyutlu vektör kullanılır ve kayan noktalı sayılar kullanıldığından, temel olarak 512 bayt büyüklüğündedir. Benzer şekilde SURF minimum 256 bayt (64-dim) büyüklüğündedir. Binlerce özellik için böyle bir vektör oluşturmak, özellikle gömülü sistemler için kaynak kısıtlama uygulamaları için uygun olmayan çok fazla bellek gerektirmektedir. Bellek ne kadar büyükse, eşleşme için gereken süre o kadar artmaktadır.

BRIEF (Binary Robust Independent Elementary Features), öznitelik detektörleriyle birleştirilebilen genel amaçlı bir öznitelik tanımlayıcısıdır. Tipik fotometrik ve geometrik görüntü dönüşümlerine karşı dayanıklıdır. BRIEF, SIFT için neredeyse benzer eşleştirme performansına sahip SIFT'den daha az karmaşıklık gerektiren başka bir alternatiftir (Calonder *et al.* 2010). BRIEF yönteminde, yumuşatılmış görüntü yamalarındaki pikseller arasında basit ikili testleri kullanılmaktadır. Performansı, aydınlatma değişimine karşı dayanıklılık, bulanıklık ve perspektif bozulma gibi birçok açıdan SIFT'e benzemektedir. Bununla birlikte, düzlem içi rotasyona karşı dayanıksızdır (Calonder *et al.* 2010; Rublee *et al.* 2011). BRIEF içerisinde bir öznitelik veya anahtar nokta detektörü içermemektedir, yalnızca tespit edilmiş öznitelik veya anahtar noktaların tanımlanması için kullanılmaktadır. Çalışmamızda BRIEF öznitelik tanımlayıcısı, yıldız öznitelik detektörü (star feature detector) ile birlikte kullanılmıştır.

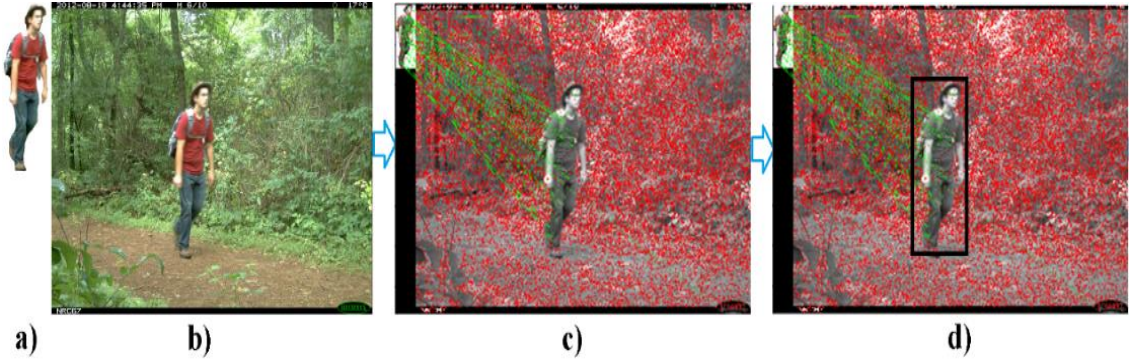
### 2.7.7. ORB (Oriented fast and rotated brief)

ORB (Yönlendirilmiş FAST ve Döndürülmüş BRIEF), FAST (Features from Accelerated Segment Test) anahtar noktası detektörünün ve BRIEF tanımlayıcısının birtakım modifikasyonları ile birleşmesidir (Bay *et al.* 2006; Karami *et al.* 2017; Rublee *et al.* 2011). Her iki teknik de iyi performansları ve düşük işlem maliyeti nedeniyle caziptir (Rublee *et al.* 2011). FAST, gerçek zamanlı uygulamalarda kullanılabilen, diğer köşe detektöründen hızlı olan, gürültüye dayanıklı bir detektördür (Karami *et al.* 2017). SIFT ve SURF anahtar nokta detektörleri bir yönelim operatörü

içermesine rağmen, FAST’de bu işlem bulunmamaktadır (Rublee *et al.* 2011). FAST algoritması yalnızca köşelerin tespit edileceği dairenin yarıçapı bilgisini parametre olarak almakta ve ORB yönteminde FAST ile birlikte tespit edilen köşelerin ölçülendirilebilmesi için Harris Köşe Ölçüsü, köşe yöneliminin belirlenmesi için ise yoğunluk ağırlık merkezi (intensity centroid) yöntemi kullanılmaktadır. Yoğunluk ağırlık merkezi yönteminde standart sabitler kullanılarak köşe yönelimleri tespit edilmektedir. Tespit edilen anahtar noktalar, BRIEF kullanılarak tanımlanmaktadır (Rublee *et al.* 2011) .

### **2.7.8. Öznitelik eşleştirme (Feature matching)**

Sınıflandırma işlemi için önerilen yaklaşımlarda uygulamanın etkin öznitelikleri ve bilgisi kullanılmaktadır. Birçok uygulamada sınıflandırıcı tasarlamak için yeterli veri bulunmamaktadır. Bu durumda modelin bilinmeyen cisim ile doğrudan eşleştirilmesi ve cisim sınıflandırmak için en uygun eşleştirme modelini seçilmektedir. Bu yaklaşımlar her bir modeli sıra ile ele almaktadır. Modelin görüntü bileşenine olan benzerliğini belirlemek için modeli görüntü verisine uyarlamaktadır. Bu genellikle bölümlendirme yapıldıktan sonra gerçekleştirilmektedir (Karami *et al.* 2017). Eşleştirme değeri, aşağıdaki formüle göre benzerlik değerleri ve özniteliklerin ağırlıklarının çarpımının toplamı ile elde edilmektedir. En yüksek benzerlik değeri cismin etiketi olarak atanmaktadır ve Şekil 2.21’de gösterildiği üzere bu etiket ile cisim tespiti gerçekleştirilmektedir. Burada yerel veya genel öznitelikler kullanılmakta ancak öznitelikler arası ilişkiler kullanılmamaktadır (Karami *et al.* 2017).



**Şekil 2.21.** Öznitelik eşleştirme tabanlı cisim tespiti a) örnek cisim b) örnek görüntü c) öznitelik eşleştirme d) cisim konumlandırma ve sınırlayıcı kutu tespiti

### 2.7.9. K en yakın komşu eşleştiricisi (kNN matcher)

FLANN (Fast Library for Approximate Nearest Neighbors), Yaklaşık En Yakın Komşular İçin Hızlı Kütüphane anlamına gelmektedir. Büyük veri kümelerinde en yakın komşu arama ve yüksek boyutlu özellikler için optimize edilmiş bir algoritmalar koleksiyonu içermektedir. Büyük veri kümeleri için Kaba Kuvvet eşleştiriciden daha hızlı çalışmaktadır. En yakın komşu arama için en iyi çalıştığı tespit edilen algoritmaların bir koleksiyonunu ve veri setine bağlı olarak en iyi algoritmayı ve optimum parametreleri otomatik olarak seçmek için bir sistem içermektedir. Hiyerarşik k-ağaçları ve rastgele kd ağaçları olmak üzere iki algoritma bulunmaktadır (Muja and Lowe 2009).

### 2.7.10. Kaba kuvvet eşleştiricisi (Brute force)

Bir anahtar noktası detektörünün en değerli özelliği, farklı görüntüleme koşulları altında anahtar noktalarını güvenilir bir şekilde bulunabilmesi anlamına gelen tekrarlanabilirlik özelliğidir. Ardından, her anahtar noktasının bulunduğu bölge yerel bir öznitelik vektörüyle temsil edilmektedir. Bu tanımlayıcı, ayırt edici ve aynı zamanda, gürültü, algılama hataları ve geometrik ve fotometrik deformasyonlar için dayanıklı olmalıdır. Son olarak, tanımlayıcı vektörler farklı görüntüler arasında Mahalanobis, Manhattan(L1) veya Öklid(L2) (Euclidean) uzaklığının hesaplanması ile

eşleştirilmektedir. Tanımlayıcının boyutunun, eşleştirme işlemi süresine doğrudan bir etkisi vardır ve bu nedenle daha az sayıda boyut beklenmektedir (Bay *et al.* 2006; Karami *et al.* 2017).

Özniteliklerin eşleştirilebilmesi amacıyla, iki görüntü arasındaki özniteliklerin birbirine olan benzerliklerini ölçmek için L1, L2 ve Hamming mesafeleri kullanılmaktadır. Bu mesafelerden L1 mesafesi Denklem 14'te, L2 mesafesi Denklem 15'te, Hamming mesafesi ise Denklem 16'da gösterilmiştir.

Kaba Kuvvet eşleştirme işleminde (Brute Force Matcher) ilk görüntüdeki bir öznitelik tanımlayıcısının, ikinci görüntüdeki tüm özniteliklerle olan mesafesi Denklem (14), Denklem (15) ve Denklem (16)'ya göre hesaplanarak ve en yakın olan eşleşme tespit edilmektedir. Bu eşleştirici iki opsiyonel parametre gerektirmektedir. Birincisi kullanılacak mesafe ölçümünü belirleyen L1, L2 ve Hamming değerleridir. Varsayılan olarak kullanılan L2 SIFT ve SURF yöntemleri için idealdir. ORB, BRIEF, BRISK gibi ikili tabanlı tanımlayıcılar için Hamming mesafesi kullanılmalıdır. İkinci parametre ise, varsayılan olarak kapalı olan çapraz kontrol değişkenidir. Eğer bu özellik açık ise, eşleştirici birinci görüntüdeki öznitelikleri ikinci görüntüdeki en yakın öznitelik ile eşleştirdiği gibi, ikinci görüntüdeki özniteliklerin birinci görüntüdeki en yakın eşleştirme olup olmadığını da kontrol etmektedir. Yani iki görüntüdeki iki öznitelik karşılıklı olarak birbiriyle eşleşmelidir (Karami *et al.* 2017).

$$d(f_a, f_b) = \sum |f_a, f_b| \quad (14)$$

$$d(f_a, f_b) = \sum (f_a - f_b)^2 \quad (15)$$

$$d(f_a, f_b) = \sum XOR(f_a, f_b) \quad (16)$$



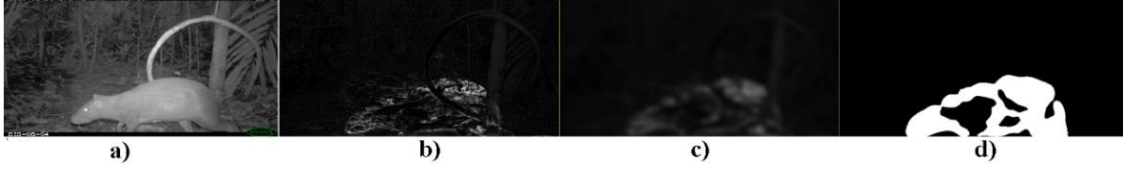
## 2.8. Ön ve Son İşleme

Ön ve son işleme, görsel algılama yöntemlerinde öznelik çıkarma, cisim tespit veya cisim tanıma işlemlerinden önce ve sonra uygulanan, gürültülerin temizlenmesi, aydınlatma değişimlerinin giderilmesi, eşikleme, görüntünün işleme hazır hale getirilmesi, sınırlayıcı kutuların belirlenmesi, küçük cisimlerin elenmesi gibi işlemlerin gerçekleştirildiği aşamadır. Çalışmalarda bu amaçla Gauss filtresi, medyan filtre, Otsu eşikleme, kabarcık algılama, histogram eşitleme, kayan pencere ve bağlı bileşen analizi yöntemleri uygulanmaktadır.

### 2.8.1. Gauss filtresi

Gauss filtresi, görüntü işlemede, yumuşatma işlemi ile görüntünün bulanıklaşması amacıyla kullanılmaktadır. Genellikle görüntü gürültüsünü azaltmak ve ayrıntıları azaltmak için yaygın olarak kullanılan bir yöntemdir. Gauss yumuşatma, farklı ölçeklerdeki görüntü yapılarını geliştirmek için bilgisayarla görü algoritmalarında bir ön işlem aşaması olarak kullanılmaktadır. Matematiksel olarak, bir görüntüye bir Gauss bulanıklığı uygulamak, görüntüyü bir Gauss fonksiyonuyla eşleştirmekle anlamına gelmektedir. Gauss bulanıklığı bir alçak geçiş filtresidir. Gauss bulanıklığı, görüntüdeki her piksele uygulanacak dönüşümü hesaplamak için bir Gauss fonksiyonunu kullanan bir filtre türüdür. Formülde gösterildiği üzere,  $x$  yatay ekseninde orijin mesafedir,  $y$  dikey ekseninde orijin mesafedir ve  $\sigma$  Gauss dağılımının standart sapmasıdır. İki boyutta uygulandığında, bu formül, konturları merkez noktasından bir Gauss dağılımına sahip eş merkezli daireler olan bir yüzey üretmektedir. Bu dağılımdaki değerler, orijinal görüntüye uygulanan bir kıvrım matrisi oluşturmak için kullanılmaktadır. Bu evrişim süreci görsel olarak sağdaki şekilde gösterilmiştir. Her pikselin yeni değeri, bu pikselin komşusunun ağırlıklı ortalaması olarak ayarlanmaktadır. Orijinal piksel değerinin en büyük ağırlığını almakta ve komşu pikseller orijinal piksele olan uzaklıkları arttıkça daha küçük ağırlıklar almaktadır. Bu, sınırları ve kenarları diğer, daha düzgün ve bulanık filtrelerden daha iyi koruyan bir bulanıklığa neden olmaktadır (Shapiro and

Stockman 2001). Şekil 2.22’da örnek görüntü üzerinde gerçekleştirilen bazı işlemler gösterilmektedir.



**Şekil 2.22.** İşlemlere ait örnek görüntüler a) görüntü, b) çerçeve farkı, c) gauss bulanıklık, d) otsu eşikleme

### 2.8.2. Medyan filtre

Medyan filtre, genellikle bir görüntüde gürültüyü temizlemek için tercih edilen doğrusal olmayan bir filtreleme yöntemidir. Bu gürültü temizleme işlemi, daha sonraki aşamaların sonuçlarını iyileştirebilmek için uygulanan bir ön işlemdir. Medyan filtre, gürültüyü çıkarırken kenarları koruduğundan dijital görüntü işlemede çok yaygın bir şekilde kullanılmaktadır, Medyan filtresinin ana fikri, her girişin komşu girişlerin ortanca değeri ile değiştirilmesi suretiyle, giriş ile sinyal girişinden geçirmektir. Komşuların değerini, tüm sinyalin üzerinde, pencere yöntemi ile girişe kaydırmaktadır. Bir boyutlu sinyaller için, en belirgin pencere sadece ilk birkaç önceki ve sonraki girişler iken, görüntüler gibi iki boyutlu sinyaller için, daha karmaşık olan kutu veya çapraz desenler gibi pencere modelleri uygulamak mümkündür. Pencerenin tek sayıda girişe sahip olması durumunda, medyan değeri penceredeki tüm girişler sayısal olarak sıralandıktan sonra sadece orta değer olmaktadır (Huang *et al.* 1979).

### 2.8.3. OTSU eşikleme

OTSU eşikleme, bilgisayarlı görü ve görüntü işlemede, Otsu yöntemi, kümeleme tabanlı görüntü eşleştirmesini otomatik olarak gerçekleştirmek için veya bir gri seviye görüntünün bir ikili görüntüye indirgenmesi için kullanılmaktadır. Algoritma, görüntünün iki modlu histogramını takip eden iki piksel sınıfını içerdiğini

varsayılmakta, daha sonra iki sınıfı ayıran optimum eşiği hesaplanmaktadır (Otsu, 1979).

#### 2.8.4. Kabarcık algılama

Kabarcık algılama (Blob detection) yönteminde, çevredeki bölgelere kıyasla, parlaklık veya renk gibi özelliklerde farklılık gösteren bir dijital görüntüdeki bölgeleri tespit etmek amaçlanmaktadır. Bir kabarcık, bazı özelliklerin sabit veya yaklaşık sabit olduğu bir görüntünün bölgesidir (Lindeberg, 1993). Kabarcık tespiti için kullanılan yöntemler, siyah arka plan üzerindeki beyaz piksellerin bulunduğu bölgeyi ve büyüklüklerini hesaplamak için kullanılmaktadır. Çalışmamızda kabarcık tespiti için LoG (Gaussian of Laplacian), DoG (Difference of Gaussian) ve DoH (Determinant of Hessian) algoritmaları kullanılmıştır. Bu yöntemler arasında LoG en başarılı ama en yavaş çalışan yaklaşımdır. LoG, görüntülerin Laplacian değerini art arda standart sapma ile hesaplamakta ve bir küp içinde toplamaktadır. Bu küpün yerel maksimumu kabarcıkları ifade etmektedir. Daha büyük kabarcıkları tespit etmek, özellikle konvolüsyon sırasında daha büyük çekirdek boyutları gerektirmesi nedeniyle daha yavaş çalışmaktadır. DoG, LoG yaklaşımından daha hızlı çalışan bir yaklaşımdır. Bu yaklaşımda, görüntü artan standart sapmalar ile bulanıklaşmakta ve birbirini izleyen iki bulanık görüntü arasındaki fark bir küp içinde biriktirilmektedir (Suzuki, 1985). DoG, görüntünün birbirini izleyen iki Gauss konvolüsyonunun çıkarılması ile hesaplanmaktadır. DoG haritasındaki yoğunluk maksimum ve minimumları, sırasıyla algılanan parlak ve karanlık bölgeleri temsil etmektedir (Schindelin *et al.* 2012). DoH ise en hızlı çalışan yaklaşımdır. Görüntünün Hessian Determinantının matrisinde maksimum noktalar ile kabarcıkların tespit edilmesine dayanmaktadır. Algılama hızı, uygulamada konvolüsyonlar yerine kutu filtreleri kullandığı için kabarcıkların boyutundan bağımsızdır. Dezavantajı ise 3 pikselden küçük lekelerin doğru olarak algılanmamasıdır (Suzuki, 1985). Bu algoritmalar, çerçeve farkı yönteminden elde edilen sonuca uygun olarak, gri ton görüntülerde kullanılabilir. Kayan pencere tabanlı histogram hesaplamada ise görüntü içerisinde değişen büyüklükte pencereler ile beyaz piksel sayısını maksimum

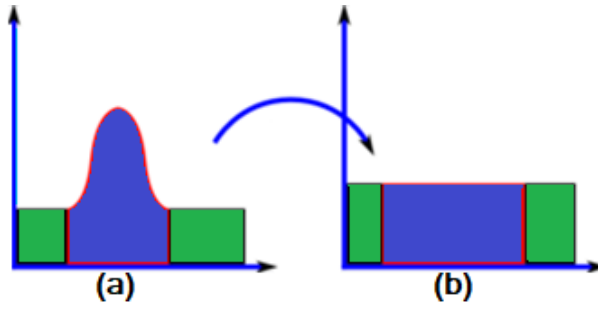
yapan konum belirlenmektedir. Şekil 2.23-a'da LoG, Şekil 2.23-b'de DoG, Şekil 2.23-c'de DoH algoritmalarının çıktıları gösterilmiştir.



Şekil 2.23. Kabarcık tespit yöntemleri a) LoG b) DoG c) DoH)

### 2.8.5. Histogram eşitleme

Histogram Eşitleme (Histogram equalization), Şekil 2.24'te görüldüğü üzere görüntünün histogramını kullanarak kontrast ayarının yapıldığı bir yöntemdir.



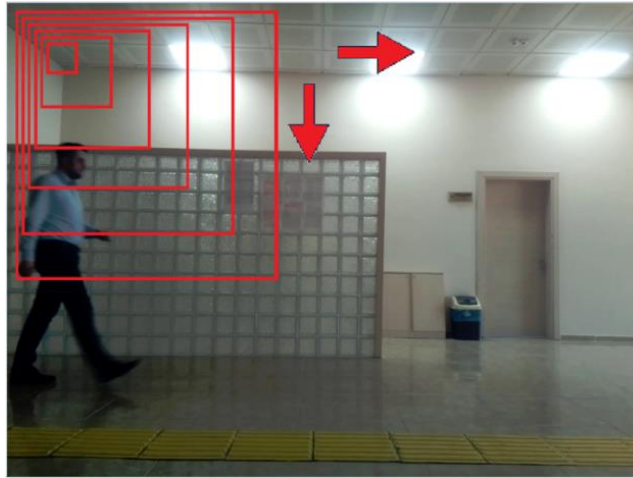
Şekil 2.24. Histogram eşitleme işlemi, öncesi (a) ve sonrası (b)

Bu yöntem özellikle görüntünün kullanılabilir verileri yakın kontrast değerleri ile temsil edildiğinde, genellikle birçok görüntünün genel kontrastını arttırmaktadır. Bu ayarlama sayesinde, yoğunluklar histogram üzerinde daha iyi dağıtılabilmektedir. Bu, daha düşük yerel kontrast alanlarının daha yüksek bir kontrast elde etmesine izin vermektedir. Histogram dengeleme, en sık yoğunluk değerlerini etkin bir şekilde yaymak suretiyle bunu gerçekleştirmektedir. Yöntem, hem aydınlık hem de her ikisi de karanlık olan arka plan ve ön plana sahip resimlerde faydalı olmaktadır. Yöntemin en önemli avantajı,

oldukça basit bir teknik ve tersine çevrilebilir bir operatör olmasıdır. Teoride, histogram eşitleme fonksiyonu biliniyorsa, o zaman orijinal histogram geri kazanılabilmektedir. Hesaplama sayısal olarak fazla işlem gerektirmemektedir. Yöntemin ayırım gözetmeden kullanılması gibi bir dezavantajı bulunmaktadır. Kullanılabilir sinyal azalırken, arka plan gürültüsünün kontrastını artabilmektedir. Histogram eşitleme genellikle fotoğraflarda gerçekçi olmayan etkiler yaratmaktadır. Ancak termal, uydu veya röntgen görüntüleri gibi bilimsel görüntüler, çoğu zaman yanlış renk uygulayacak resimlerin sınıfı için çok faydalı olmaktadır. Aynı zamanda histogram eşitlemesi, düşük renk derinliğine sahip görüntülere uygulandığında istenmeyen etkiler üretebilmektedir (Hum *et al.* 2014).

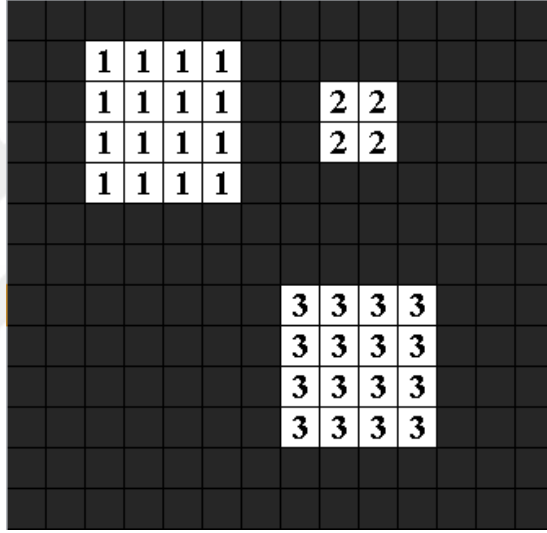
### 2.8.6. Kayan pencere yöntemi ve bağlı bileşen analizi

Cisim tespit etme ve konumlandırma için bir pencereyi görüntü üzerinde çoklu ölçekler şeklinde kaydırmak en yaygın yaklaşımdır. Şekil 2.25'te gösterildiği üzere kayan pencere olarak ifade edilen yöntemde, değişen boyutlardaki bu yerel pencerelerde bulunan görüntüler, aranan cisim veya arka plan şeklinde sınıflandırılarak tespit işlemi gerçekleştirilebilmektedir. Bu yaklaşım, görüntülerde yüz, yaya ve araba gibi cisimleri tespit etmek için başarıyla kullanılmaktadır (Murphy *et al.* 2006).



Şekil 2.25. Kayan pencere yaklaşımı

Bağlı bileşen analizi, Şekil 2.26’da gösterildiği üzere görüntülerde birbiri ile komşuluğu bulunan alt bölge kümelerinin belirli bir sezgisel veri tabanına göre benzersiz bir şekilde etiketlendiği bir grafik teorisi uygulamasıdır (Samet and Tamminen 1988). Bağlı bileşen analizi, ikili giriş görüntüsünün bağlı bileşeniyle ilişkili konumların benzersiz bir etikete sahip olduğu etiketli bir görüntünün reaksiyonu olarak (Dillencourt *et al.* 1992) veya her piksele atanan etiketin, o pikselin ait olduğu bağlı bileşeni benzersiz olarak tanımlayan bir tamsayı olduğu sembolik bir görüntü olarak (Shapiro, 1996) tanımlanabilmektedir.



**Şekil 2.26.** Görüntüdeki bağlı bileşenlerin tespit edilmesi

### 3. MATERYAL ve YÖNTEM

Nesnelerin interneti için kenar bilişim tabanlı akıllı görsel algılama analizi yapılması amacıyla, görsel algılama işlemlerinin gerçekleştirilebileceği bir akıllı nesneye ihtiyaç duyulmaktadır. 3.1 başlığında tez kapsamında tasarlanan akıllı nesneye ait özellikler, kullanılan elektronik malzemeler ve algılayıcılar gösterilmiş olup, analiz için kullanılan senaryolar ifade edilmiştir. 3.2 başlığında ise görsel algılama analizi için kullanılan yöntemler ve bu yöntemlerde elde edilen sonuçlar detaylandırılmıştır. Çalışmada görsel algılama işleminin performansını daha iyi analiz edebilmek amacıyla 3.3 başlığında ifade edilen farklı özelliklere sahip üç farklı veri seti kullanılmıştır. Birinci veri seti foto-kapan cihazları tarafından kaydedilmiş görüntülerden oluşmakta olup, ikinci veri seti hareketli cisim içeren videoları içermektedir. Üçüncü veri seti ise tarafımızca tasarlanan akıllı nesne kullanılarak laboratuvar ortamında kaydedilen videolardan oluşmaktadır.

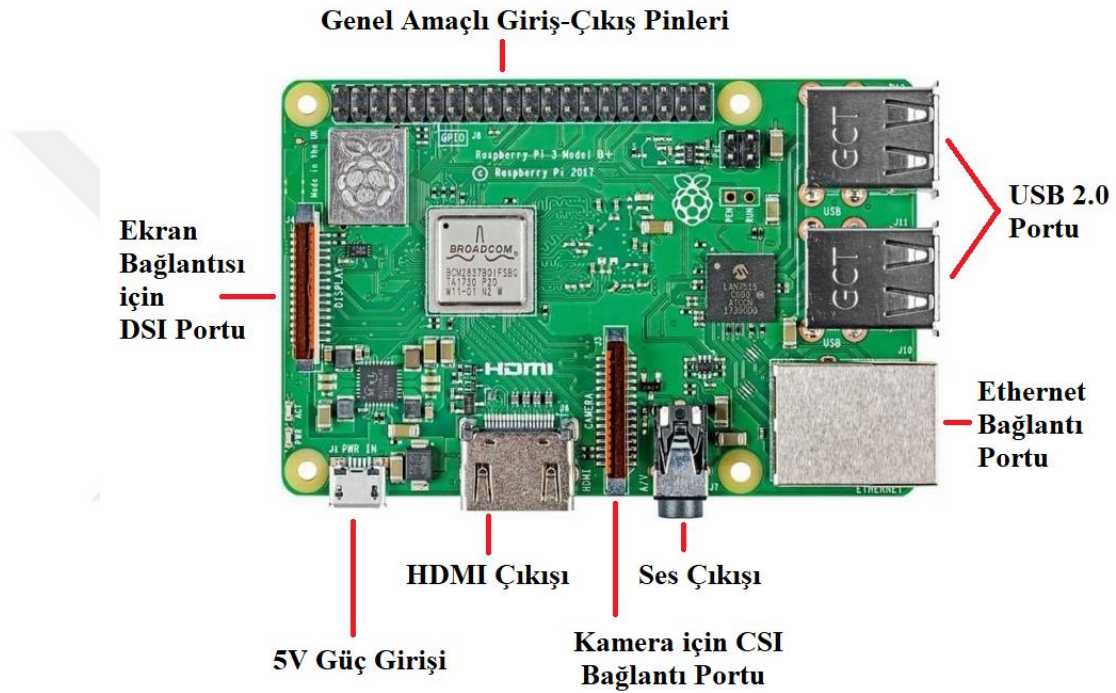
#### 3.1. Akıllı Nesne Tasarımı ve Veri İşleme Senaryoları

Tez kapsamında tasarlanan akıllı nesne, foto-kapan cihazının hareket ile tetiklenidğinde görüntü kaydetme işlevine ek olarak, internete bağlanma, internet üzerinden veri alıp gönderme ve kaydedilen görüntüler üzerinde görüntü işleme ve bilgisayarlı görü yöntemlerine ait işlemleri gerçekleştirme fonksiyonlarına sahiptir. Şekil 3.1'de tasarlanan akıllı nesne gösterilmektedir.



Şekil 3.1. Akıllı nesne tasarımı

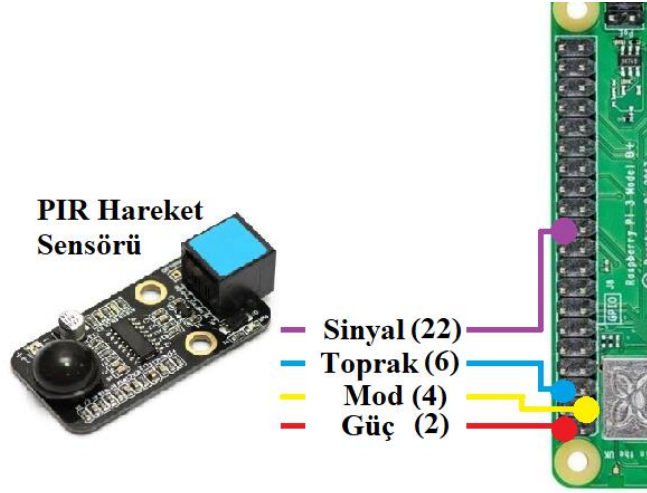
Tasarlanan akıllı nesnede Şekil 3.2’de gösterilen Raspberry Pi 3 model B + kullanılmıştır. Bu geliştirme kartı Cortex-A53 1.4GHz işlemci ve 1GB sdram’e ve 2.4GHz ve 5GHz destekli wi-fi modülüne sahiptir. Geliştirme kartında işletim sistemi olarak Raspian Stretch, sabit disk olarak 16GB harici sd kart kullanılmıştır.



**Şekil 3.2.** Akıllı nesnede kullanılan elektronik geliştirme kartı

Hareket algılama için 6 metre menzile, 120 derece görüş açısına sahip olan PIR hareket algılayıcısı kullanılmış olup, hareket algılayıcısı ve bağlantı şeması Şekil 3.3’te gösterilmiştir. Tasarımı tamamlanmış ve çalışmada kullanılmış akıllı nesne Şekil 3.4’te gösterilmektedir. Şekil 3.5’te ise akıllı nesnenin gerçekleştireceği görsel algılama yaklaşımlarına işlemlerin detayları gösterilmektedir.

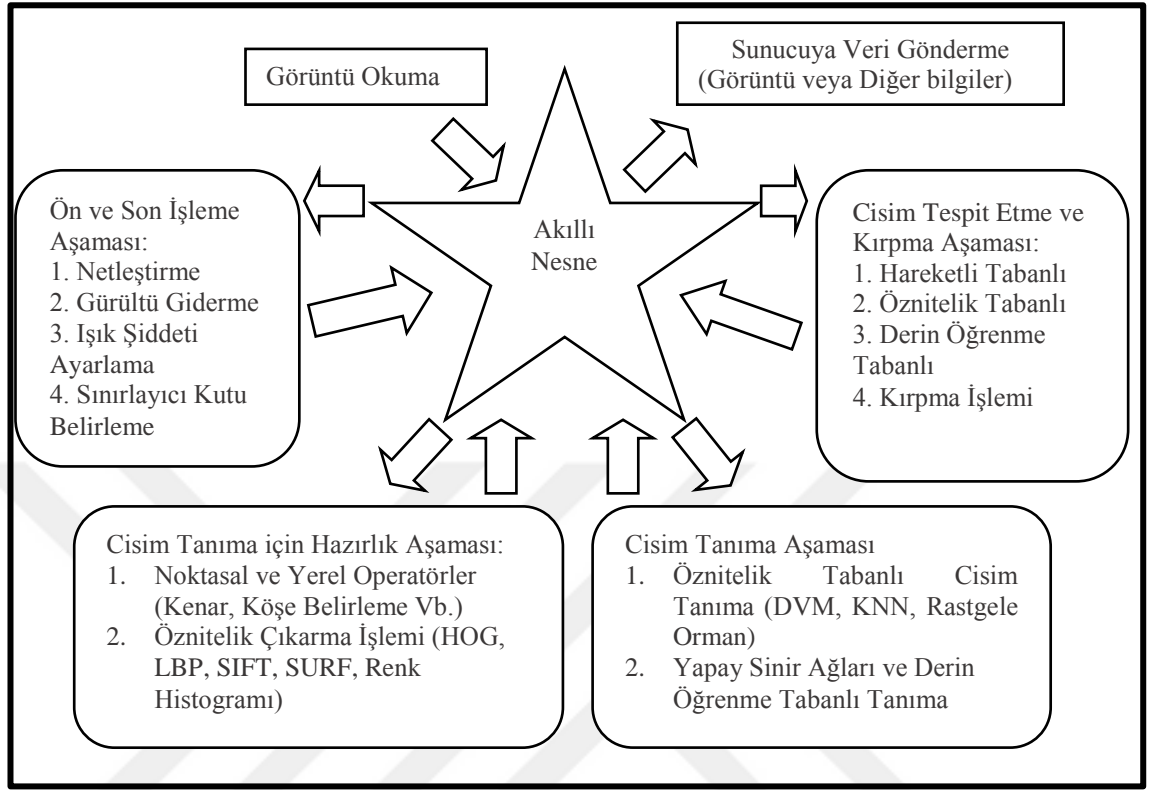




Şekil 3.3. Hareket algılayıcısı ve bağlantı şeması ( bağlantı adı ve pin numarası)



Şekil 3.4. Çalışma için tasarlanan akıllı nesne



**Şekil 3.5.** Akıllı nesne işlevleri

Çalışmada akıllı nesnede, kaydedilen görüntüler üzerinde görüntü işleme, bilgisayarlı görü ve örüntü tanıma işlevleri uygulanacaktır. Bu işlemler ön ve son işleme, öznitelik çıkarma, cisim tespit etme ve cisim tanıma olmak üzere 4 başlığa ayrılabilir. Kullanılacak yöntemler liste halinde aşağıda şekilde ifade edilebilir:

1. Görüntü Ön İşleme Yöntemleri (Kontrast Ayarlama, Netleştirme, Gürültü Giderme, Histogram Çıkarma, Histogram Eşitleme, Diğer Filtre veya Operatörleri Uygulama)
2. Hareketli Cisim Tespit Etme Yöntemleri (Arka Plan Çıkarma, Çerçeve Farkı)
3. Öznitelik Çıkarma Yöntemleri (HOG, LBP Renk Histogramı, SIFT, SURF, BRIEF, ORB)
4. Sınıflandırma Yöntemleri (DVM, Knn, Rastgele Orman, Sinir Ağı Tabanlı Yöntemler)

Çalışmada kenar bilişim tabanlı görsel algılama için cisim tespit etme ve cisim tanıma problemleri üzerinde çalışılmıştır. Uygulama için seçilen foto-kapanların kullanılma amaçları dikkate alındığında, yalnızca örüntü tanıma tabanlı yöntemlerin kullanılması yerine, hareketli cisimlerin tespit edilmesi amacıyla kullanılan ve örüntü tanıma yöntemlerine göre çok daha az bellek ve işlem gerektiren hareketli cisim tespit etme yöntemlerinin de kullanılması uygun görülmüştür. Tasarlanan akıllı nesne, foto-kapana cihazına benzer olarak, hareket algılandığında görüntü kaydetmekte, bu görüntüler üzerinde belirli işlemleri gerçekleştirebilmekte ve internet ağı üzerinden veri gönderip alabilmektedir. Şekil 3.5'teki sistem modelinde, tasarlanan akıllı nesnede görüntüler üzerinde gerçekleştirilecek işlemler senaryolar halinde gösterilmektedir. Literatürdeki diğer çalışmalardan farklı olarak bu çalışma, belirli bir yöntem, yaklaşım veya görsel öznitelik türü ile sınırlandırılmamış olup, birçok yöntem ve farklı öznitelik türünün bellek, iletişim ve işlem gereksiniminin analizini içermektedir. Foto-kapan işlevlerini yerine getirebilmek için akıllı nesnenin Çizelge 3.1'de gösterildiği şekilde hareket algılandığında görüntü kaydetmesi sağlanmıştır. Standart foto-kapanların aksine her saniye ayarlanabilir sayıda görüntü kaydetmek yerine, tespit, tanıma ve analiz işlemlerinin etkili yapılabilmesi için donanımın elverdiği en kaliteli video kaydı yapılmıştır. Foto-kapanlarda kullanılan standart ayarlar olan saniyede 1 görüntü kaydı ve 5 saniyede bir hareket kontrolü ile görüntü kaydının sonlandırılması gibi ayarlar Çizelge 3.1'deki şekilde değiştirilerek uygulanmıştır. Tasarlanan akıllı nesne, saniyede 1200x1600 boyutlarında 32 görüntü kaydetmekte olup, 1.5 saniye süre hareket algılanmadığında kaydı sonlandırmaktadır. Tasarlanan akıllı nesne ile gerçek zamanlı hareketli cisim tespiti yapılması amacıyla Çizelge 3.2 ve Çizelge 3.3'te gösterilen algoritmalar kullanılmıştır. Hareket algılandığında elde edilen görüntülerin Çizelge 3.2'deki şekilde gerçek zamanlı olarak sunucuya gönderilmesi, Çizelge 3.3'teki şekilde sunucuda kaydedilmesi ve sunucu ekranında gösterilmesi sağlanmıştır. Çizelge 3.2'de görüldüğü üzere, hareket algılandığında başlatılan Soket iletişimi ile elde edilen görüntüler sunucuya TCP ile gönderilmekte ve 18 görüntüde 1 kere hareket algılama gerçekleşmediğinde gönderip bitirilmektedir.

**Çizelge 3.1.** Akıllı nesne ile hareket algılandığında görüntü kaydetme algoritması

```
While (True);
```

```
    If (Hareket Algılanırsa);
```

```
        Hareket = Var; Video Kaydı Başlat();
```

```
        While(Hareket Var ise);
```

```
            Hareket = Yok;
```

```
            For (1'den 10'a Kadar);
```

```
                If (Hareket Algılanırsa);
```

```
                    Hareket = Var;
```

```
                    0.15 Saniye Uyu();
```

```
                Video Kaydını Durdur();
```

```
            0.01 SaniyeUyu();
```

**Çizelge 3.2.** Gerçek zamanlı hareketli cisim tespiti için istemci algoritması

```
While (True);
```

```
    If (Hareket Algılanırsa);
```

```
        Hareket = Yok; Sayaç=0;
```

```
        Soket İletişimini Başlat (TCP_IP, TCP_PORT);
```

```
        For( Görüntü = Sıradaki Görüntüyü Al());
```

```
            Görüntüyü Sunucuya Gönder (Görüntü);
```

```
            Sayaç = Sayaç +1;
```

```
            If (Hareket Algılanır ise);
```

```
                Hareket = Var;
```

```
            If ( Sayaç % 18 == 0 ve Hareket Yok ise);
```

```
                Break;
```

```
            If (Sayaç % 18 == 0 ise);
```

```
                Hareket = Yok;
```

```
        Soket İletişimini Durdur();
```

```
    0.001 SaniyeUyu();
```

**Çizelge 3.3.** Gerçek zamanlı hareketli cisim tespiti için sunucu algoritması

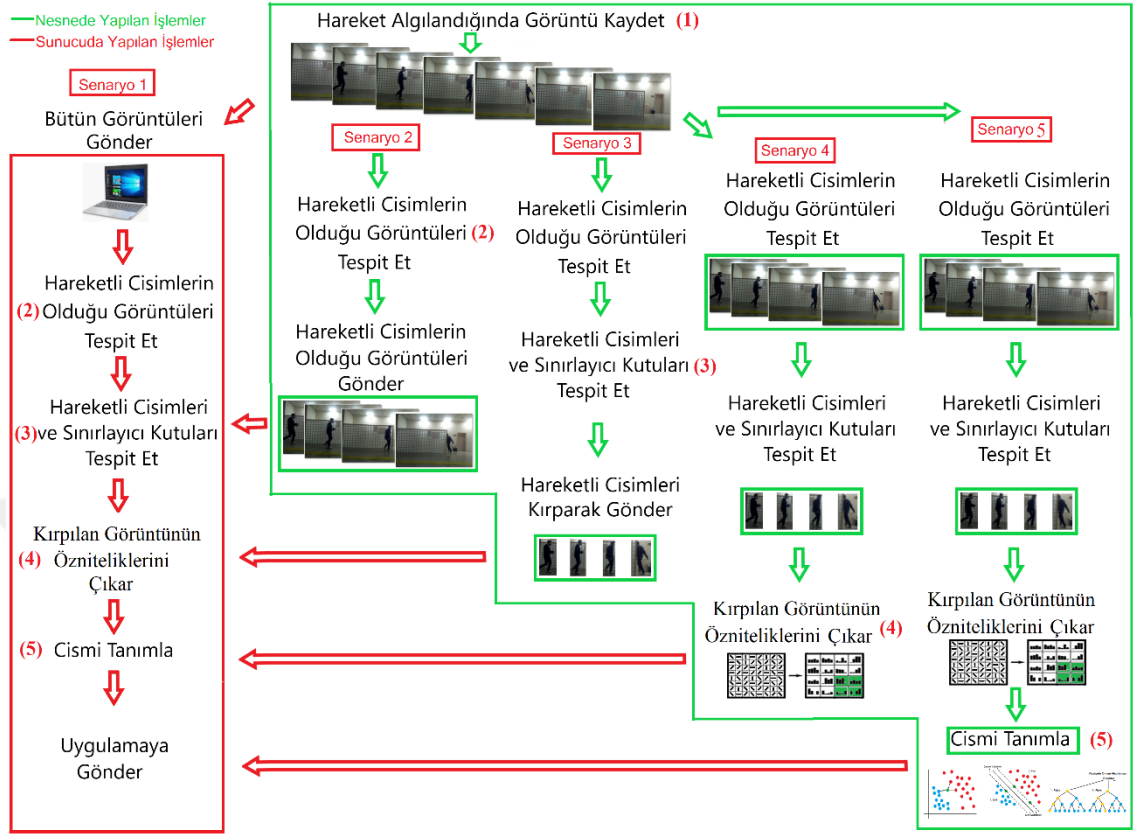
```

While (True);
    Soket İletişimini Başlat (TCP_IP, TCP_PORT);
    Soket Dinle ();
    If (Veri Aktarımı Başlarsa);
        While (Veri Aktarımı Devam Ettiği Sürece);
            Görüntü = Görüntüyü Al();
            Görüntü Ekranta Göster();
            Görüntüyü Kaydet();
        Görüntüleri Video Formatında Kaydet();

```

Tasarlanan akıllı nesne cisim tespit etme ve tanıma çalışması için önerilen sistem modeli Şekil 3.6'da gösterilmektedir. Cisim tespit etme ve tanıma aşamalarının nesne ve sunucu arasındaki farklı senaryolarla gerçekleştirilerek, yöntemlerin, senaryoların ve öznitelik dönüşüm yöntemlerinin, bellek, iletişim, işlem gereksinimleri analiz edilmiştir. Tez kapsamında tasarlanan akıllı nesne, farkındalık ilkesine uygun olarak, çevresindeki hareketliliği algılayarak gerekli tepkiyi verebilmektedir. Temsiliyet ilkesine uygun olarak sunucu tarafından tekil olarak bulunan bir isim ve IP adresi ile temsil edilmekte ve bu akıllı nesne ile elde edilen bilgiler bu temsiliyet altında kaydedilmektedir. Etkileşim ilkesine uygun olarak ise sunucu ve kullanıcı ile etkileşime geçerek, kontrol edilebilmekte, yönlendirilebilmekte ve özelliklere ayarlanabilmektedir. Şekil 3.6'da kırmızı renkte numaralandırma ile gösterilen işlemler sırasıyla;

- 1) Görüntülerin Kaydedilmesi,
- 2) Hareketli Cisim İçeren Görüntülerin Seçilmesi,
- 3) Hareketli Cisim İçeren Görüntülerde Cisim Bölgelerinin Kırpılması,
- 4) Kırpılan Görüntü Parçalarının Özniteliklerinin Çıkarılması ve
- 5) Çıkarılmış Öznitelikler Kullanılarak Hareketli Cismin Sınıflandırılması olarak ifade edilebilmektedir.



**Şekil 3.6.** Önerilen sistem modeli ve gerçekleştirilecek işlemler

Önerilen sistem modelinde cisim tespiti hareketli cisim tespiti ve öznitelik tabanlı tespit olarak iki farklı yaklaşım uygulanmıştır. Hareketli cisim tespiti yaklaşımı kaydedilen görüntülerde yalnızca hareketli olanların tespit edilmesi amacını taşımaktayken, öznitelik tabanlı yöntemler ise belirli cisimlere ait model görüntüler kullanılarak videolarda cisim tespiti yapılmaktadır. Bu tespit etme işlemi, kaydedilen görüntülerde cisim olmayan görüntülerin filtrelenmesi için ilk aşama olarak uygulanmıştır. Çizelge 3.4’de tez kapsamında uygulanan senaryolar, bu senaryolarda gerçekleştirilen işlemler, bu işlemlerin bellek ve işlem gereksinimleri ifade edilmektedir.

**Çizelge 3.4.** Tezde uygulanarak performansları analiz edilen senaryolar

Senaryolar	Gerçekleştirilen İşlemler	Sunucuya Gönderilen Veri Miktarı	Nesnelerin İşlem ve Bellek Gereksinimi
Senaryo 1	1	Bütün Görüntüler	Minimum
Senaryo 2	1, 2	Cisim Bulunan Görüntüler	Az
Senaryo 3	1, 2, 3	Cisimlere ait Kırılmış Görüntü Parçaları	Orta
Senaryo 4	1, 2, 3, 4	Kırılmış Görüntülerin Öznitelikleri	Yüksek
Senaryo 5	1, 2, 3, 4, 5	Sonuç-Cisim Etiketleri	Maksimum

Senaryolarda kullanılan yöntemlerden önce gürültü temizleme, netleştirme, kontrast ayarlama gibi ön işlemler gerçekleştirilmektedir ve bu işlemler cisim tespit etme aşamasına hazırlık olarak ifade edilebilmektedir. Görüntüde cisim tespit etme aşamasında iki farklı yol izlenmiştir. Birinci yolda başlık 3.2.1.'de önerilen yöntemde, arka plan çıkarma ve çerçeve farkı yöntemleri ile cismin tespiti gerçekleştirilmiştir. Bu işlemlerde iki veya daha fazla görüntüye ihtiyaç duyulmaktadır. Başlık 3.2.3.'te önerilen ikinci yolda ise model görüntülerin öznitelikleri kullanılarak öznitelik eşleştirme veya kayan pencere tabanlı yöntem ile cisim tespiti gerçekleştirilmiştir. Birinci yolda yalnızca hareket eden cisimlerin bulunduğu görüntüler yeterli olurken, ikinci yolda görüntüde tespit edilmeye çalışılan cisme ait örnek model görüntülere ihtiyaç duyulmaktadır. Bu ayırım çalışmanın kapsamını ve problemin ele alınışını oldukça değiştirmektedir. İşlem 2 uygulanarak cisim tespitinin gerçekleştirilmesi ile içerisinde bulunan görüntüler seçilerek, diğer görüntüler elenmektedir. Cisim tespit etme aşamasından sonra işlem 3 ile cismin bulunduğu konum belirlenerek bu sınırlayıcı kutuya ait kısım görüntüden kırılmaktadır. Cisim görüntülerinin kırılmasından sonra cisim tanıma işlemi için yapılması gereken diğer bir işlem ise öznitelik çıkarmadır. Bu aşama için uygulanan işlem 4'te kırılmış görüntülerdeki öznitelikler çıkarılmaktadır. Çalışmada HOG, LBP, Renk Histogramı, SIFT, SURF, BRIEF, ORB gibi öznitelik dönüşümü yöntemleri kullanılmıştır. Cisim tanıma işlemi için gerekli son aşama ise işlem 5'te elde edilen özniteliklerin daha önce oluşturulmuş veri tabanı ile kıyaslanması, cismin tanımlanması gerçekleştirilmesidir. Cisim tespitinde kullanılan 2. yol uygulandığı durumlarda ise görüntüde aranacak model cisim görüntüsünün öznitelikleri

örüntü tanıma veya öznitelik eşleştirme yöntemleri kullanıldığından, cismin görüntüdeki varlığı, sınıfı ve bulunduğu konum, sonraki aşamalara gerek kalmadan elde edilebilmektedir.

### 3.2. Görsel Algılama Analizi

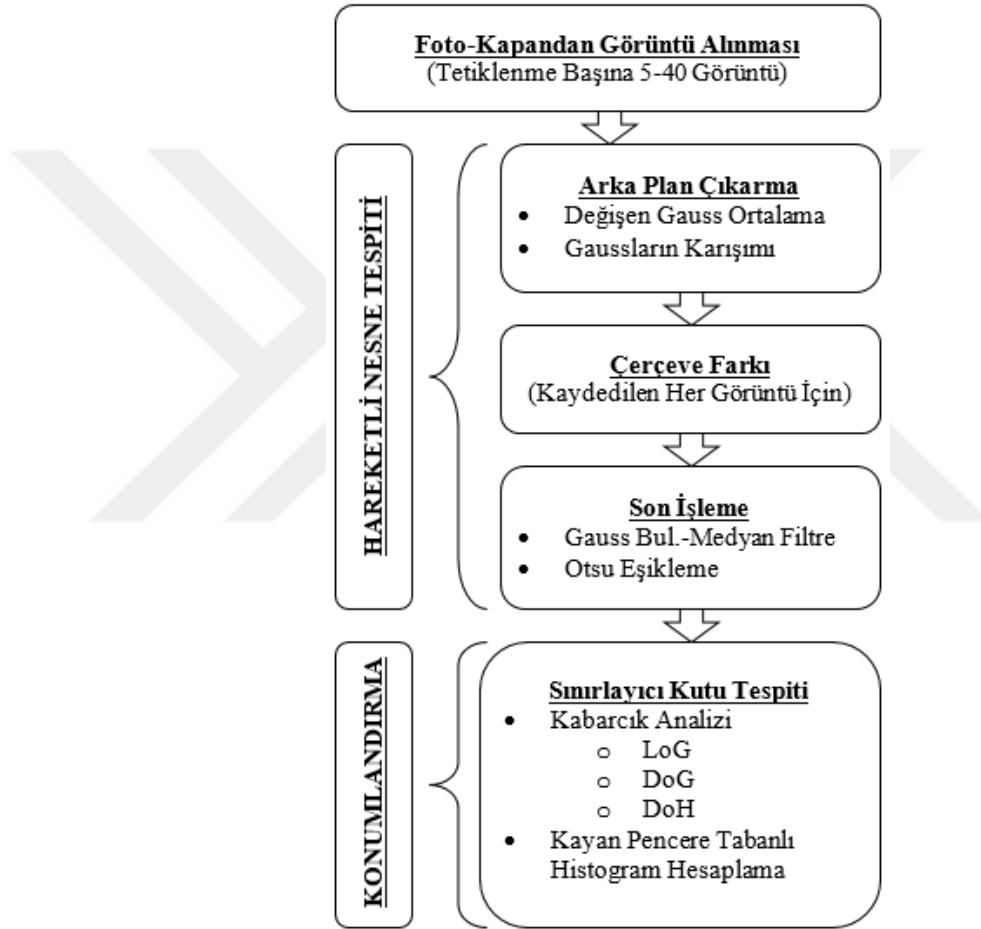
Tasarlanan akıllı nesnede görsel algılama amacıyla önerilen yöntemler bu bölümde alt başlıklar şeklinde ifade edilmiştir. Görsel algılama için kullanılan senaryolara uygun olarak öncelikle hareketli cisim bulunan görüntülerin belirlenmesi amacıyla arka plan çıkarma ve çerçeve farkı yöntemlerini içeren yaklaşım kullanılmıştır. Bu yönteme ek olarak çalışmada tarafımızca önerilen hareketli cisim tespiti iyileştirilmesi ve akıllı üç çerçeve farkı yöntemi olmak üzere iki yeni yaklaşım kullanılmıştır. Bu çalışmaların dışında, hareket içermeyen görüntülerde cisimlerin tespit edilebilmesi amacıyla yerel öznitelikler ve öznitelik eşleştirme ile cisim tespiti, derin öğrenme mimarisi olan YOLO ile cisim tespiti yöntemleri kullanılmıştır. Cisim tanıma işlemi için ise yapay sinir ağı derin öğrenme mimarisinin yanı sıra, HOG, LBP, renk histogramı, SIFT ve SURF öznitelik dönüşümü yöntemleri ile k en yakın komşu, rastgele orman, doğrusal ve doğrusal olmayan destek vektör makineleri örüntü tanıma yöntemleri uygulanmıştır.

#### 3.2.1. Arka plan çıkarma ve çerçeve farkı ile hareketli cisim tespiti

Foto-kapan görüntülerinde hareketli cisimlerin tespiti için önerdiğimiz sistem modeli (Şimşek et al. 2018a) Şekil 3.7'de gösterilmiştir. Foto-kapanlar hareket ile tetiklendiğinde saniyede 1 görüntü olmak üzere toplam en az 5 görüntü yakalamakta ve hareket devam ettiği sürece görüntü kaydetmektedir (Meek et al. 2013; Meek et al. 2014). Veri setlerinde, tetikleme başına 5 ile 40 arasında değişen sayıda kaydedilmiş görüntüler bulunmaktadır. Önerilen sistemin ilk aşamasında görüntü içerisinde hareketli cisimlerin tespiti gerçekleştirmektedir. Yöntemimizde, her tetiklemede elde edilen görüntüler kullanılarak öncelikle bu tetiklenmeye ait arka plan modeli oluşturulmuştur. Bu arka plan modeli çerçeve farkı yönteminde referans görüntü olarak kullanılmıştır.



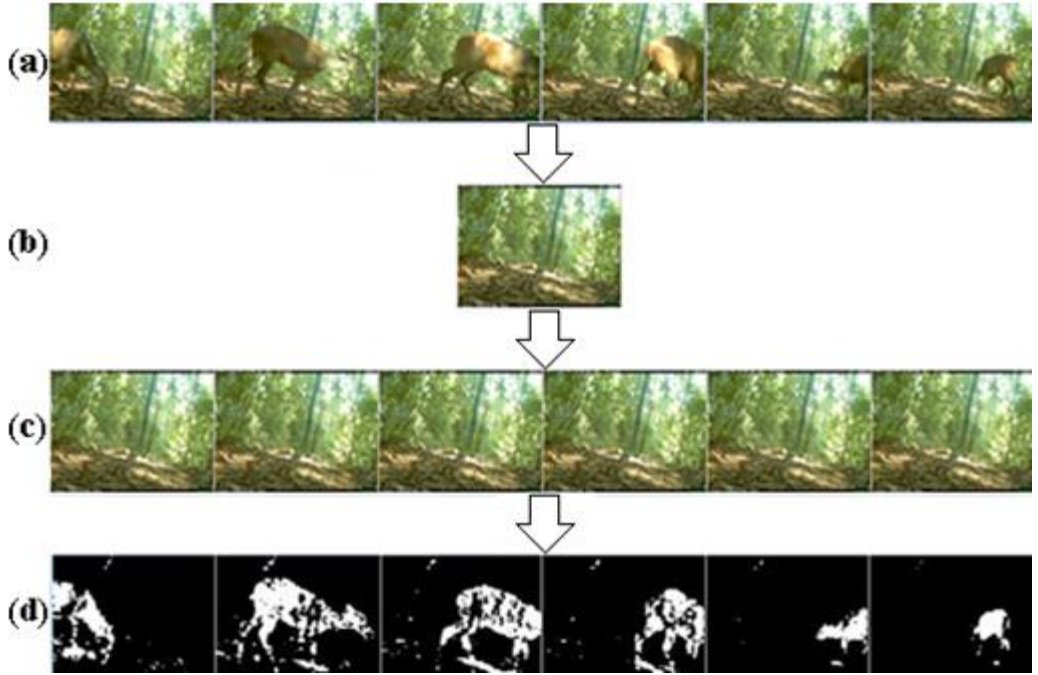
Hareketli cisimlerin bulunduğu görüntüler arka plan modelinden çıkartılarak hareketli cisimler tespit edilmiştir. Elde edilen görüntüler filtreleme ve eşikleme işlemleri ile gürültü ve hatalardan ayrıştırılıp, hareketli cisimler belirgin hale getirilmiştir. Sistemin ikinci aşamasında tespit edilen hareketli cismin görüntü içerisinde konumu ve cismin boyutu belirlenmiştir.



**Şekil 3.7.** Foto-kapan görüntülerinde hareketli cisim tespiti için önerilen yöntem

Önerilen sistem, model cisim görüntülerinin elle kırılarak kullanıldığı literatürdeki diğer çalışmalardan farklı olarak, cisimlere ait ön bilgi kullanılmadan görüntülerin arka plandan hareketli cisimlerin ayırt edilmesine dayanmaktadır. Önerilen sistemde hareketli cisim tespiti için gerekli adımlar Şekil 3.7’de görüleceği üzere iki aşamadan oluşmaktadır. Şekil 3.8’de bu iki işleme ait örnek görüntüler gösterilmektedir. Birinci

aşamada, Şekil 3.8-a ve Şekil 3.8-b’de gösterildiği üzere, her tetiklemede elde edilen görüntüler kullanılarak arka plan modeli oluşturulmaktadır. Arka plan, sabit bir kameradan çekilmiş görüntülerde belirli süre hareketsiz kalan cisimlerden meydana gelmektedir. Bu amaçla çalışmamızda Değişen Gauss Ortalama (Running Gaussian Average) ve Gaussların Karışımı (Gaussian Mixture) yöntemleri kullanılmış ve yöntemlerin başarıları karşılaştırılmıştır. Arka plan modeli elde edildikten sonra her görüntüdeki hareketli cisimlerin tespiti için Şekil 3.8-a ve Şekil 3.8-c’de gösterilen görüntüler üzerinde çerçeve farkı yöntemi uygulanmış ve Şekil 3.8-d çıktıları elde edilmiştir.



**Şekil 3.8.** (a) Tetikleme sonrası kaydedilen foto-kapan görüntüleri (b) bu görüntüler kullanılarak arka plan modelinin oluşturulması (c) 1. aşamadaki görüntüler ile 2. aşamadaki arka plan modelinin çerçeve farkı yönteminde kullanılması (d) görüntülerdeki hareketli cisimlerin elde edilmesi

Şekil 3.9’da ışık şiddeti değişimi olan ardışık görüntüler gösterilmiştir. Işık şiddeti değişimi, Şekil 3.10-a’da görülebileceği üzere, arka plan modelinin hatalı oluşturulmasına ve cisim tespitinin yapılamamasına neden olmaktadır. Bu problem histogram eşitleme yöntemi kullanılarak çözülebilmektedir. Şekil 3.10-b’de ışık şiddeti

değişimi olan görüntülerdeki hataların giderilmesinden sonra elde edilen hareketli cisim çıktısı gösterilmiştir.

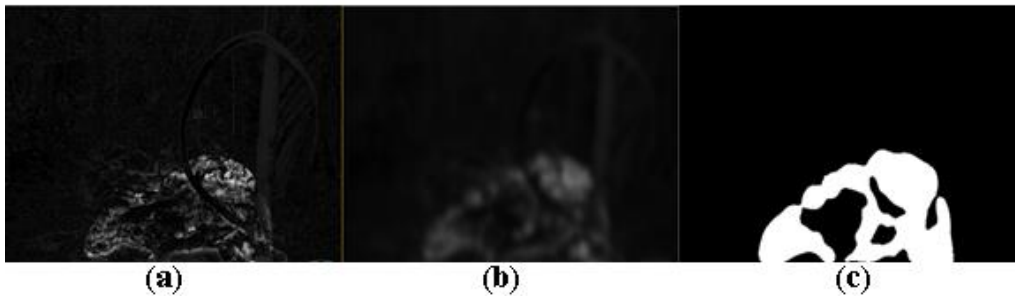


**Şekil 3.9.** Ardışık görüntülerdeki ışık şiddeti değişimi



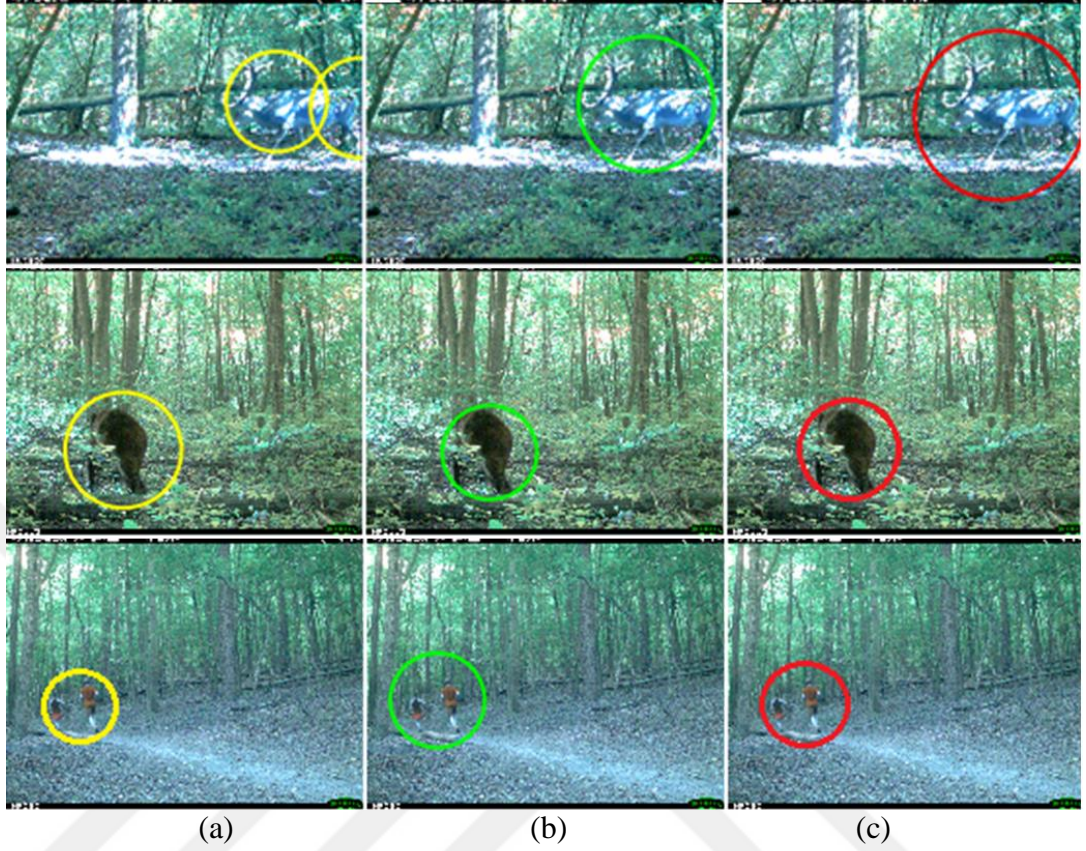
**Şekil 3.10.** Işık şiddeti farkı olan görüntülerin çerçeve farkı çıktısı (a) normal, (b) histogram eşitleme

Ardışık görüntülerdeki hareketli arka plan, çerçeve farkı yönteminden elde edilen çıktı görüntülerinde gürültüye neden olmaktadır. Görüntülerde meydana gelen gürültüleri azaltılmak ve cisimi belirginleştirilmek amacıyla Gauss bulanıklaştırma, Medyan Filtre ve OTSU eşikleme yöntemleri ayrı ayrı uygulanmıştır. Şekil 3.11'de, çerçeve farkı, bulanıklaştırma ve eşikleme işlemlerinden sonraki çıktılar gösterilmektedir.



**Şekil 3.11.** (a) Çerçeve farkı çıktısı (b) gauss bulanıklaştırma (c) otsu eşikleme

Yöntemin ikinci aşamasında, tespit edilen hareketli cisimlerin konumlandırılması amacıyla Kabarcık Analizi (Blob Analysis) ve kayan pencere tabanlı histogram hesaplama yöntemleri uygulanmıştır. Konumlandırma aşamasında hareketli cismin bulunduğu konum ve büyüklüğü hesaplanarak sınırlayıcı kutu (bounding box) tespit edilmektedir. 1. aşamanın sonunda Şekil 3.11-c'de gösterilen ikili (binary) çıktı elde edilmektedir. Kabarcık tespiti için kullanılan yöntemler, siyah arka plan üzerindeki beyaz piksellerin bulunduğu bölgeyi ve büyüklüklerini hesaplamak için kullanılmaktadır. Çalışmamızda kabarcık tespiti için LoG (Gaussian of Laplacian), DoG (Difference of Gaussian) ve DoH (Determinant of Hessian) algoritmaları kullanılmıştır. Bu yöntemler arasında LoG en başarılı ama en yavaş çalışan yaklaşımdır. LoG, görüntülerin Laplacian değerini art arda standart sapma ile hesaplamakta ve bir küp içinde toplamaktadır. Bu küpün yerel maksimumu kabarcıkları ifade etmektedir. Daha büyük kabarcıkları tespit etmek, özellikle konvolüsyon sırasında daha büyük çekirdek boyutları gerektirmesi nedeniyle daha yavaş çalışmaktadır. DoG, LoG yaklaşımından daha hızlı çalışan bir yaklaşımdır. Bu yaklaşımda, görüntü artan standart sapmalar ile bulanıklaşmakta ve birbirini izleyen iki bulanık görüntü arasındaki fark bir küp içinde biriktirilmektedir (Suzuki, 1985). DoG, görüntünün birbirini izleyen iki Gauss konvolüsyonunun çıkarılması ile hesaplanmaktadır. DoG haritasındaki yoğunluk maksimum ve minimumları, sırasıyla algılanan parlak ve karanlık bölgeleri temsil etmektedir (Schindelin *et al.* 2012). DoH ise en hızlı çalışan yaklaşımdır. Görüntünün Hessian Determinantının matrisinde maksimum noktalar ile kabarcıkların tespit edilmesine dayanmaktadır. Algılama hızı, uygulamada konvolüsyonlar yerine kutu filtreleri kullandığı için kabarcıkların boyutundan bağımsızdır. Dezavantajı ise 3 pikselden küçük lekelerin doğru olarak algılanmamasıdır (Suzuki, 1985). Bu algoritmalar, çerçeve farkı yönteminden elde edilen sonuca uygun olarak, gri ton görüntülerde kullanılabilir. Kayan pencere tabanlı histogram hesaplamada ise görüntü içerisinde değişen büyüklükte pencereler ile beyaz piksel sayısını maksimum yapan konum belirlenmektedir. Şekil 3.12-a'da LoG, Şekil 3.12-b'de DoG, Şekil 3.12-c'de DoH algoritmalarının çıktıları, Şekil 3.13'te kayan pencere tabanlı konumlandırma işlemine ait çıktı gösterilmiştir.



**Şekil 3.12.** Kabarcık analizi algoritmasının sonuçları (a) laplacian of gaussian (b) difference of gaussian (c) determinant of hessian



**Şekil 3.13.** Kayan pencere tabanlı konumlandırma

Sınırlayıcı kutu tespitinde başarı ölçümü amacıyla birleşimin kesişimi (IoU-Intersection Over Union) formülü kullanılmaktadır (Zitnick and Dollár 2014). Şekil 3.14'te siyah olan sınırlayıcı kutu cismin bulunduğu bölgeyi, kırmızı ise bizim elde ettiğimiz sınırlayıcı kutuyu temsil etmektedir. Örnek olarak gösterilen bu iki sınırlayıcı kutu için konumlandırma işleminin başarısı, bu iki kutunun kesiştiği bölgenin piksellerinin, iki

kutunun birleşimi olan bölgenin piksellerine oranı üzerinden değerlendirilmektedir. Konumlandırma işleminin başarısı için cisim tespit çalışmalarında birleşimin kesişimi değeri olarak 0,5, 0,7 ve 0,9 kullanılmakta olup (Zitnick and Dollár 2014) çalışmamızda bu değerlere ait konumlandırma sonuçları gösterilmiştir.



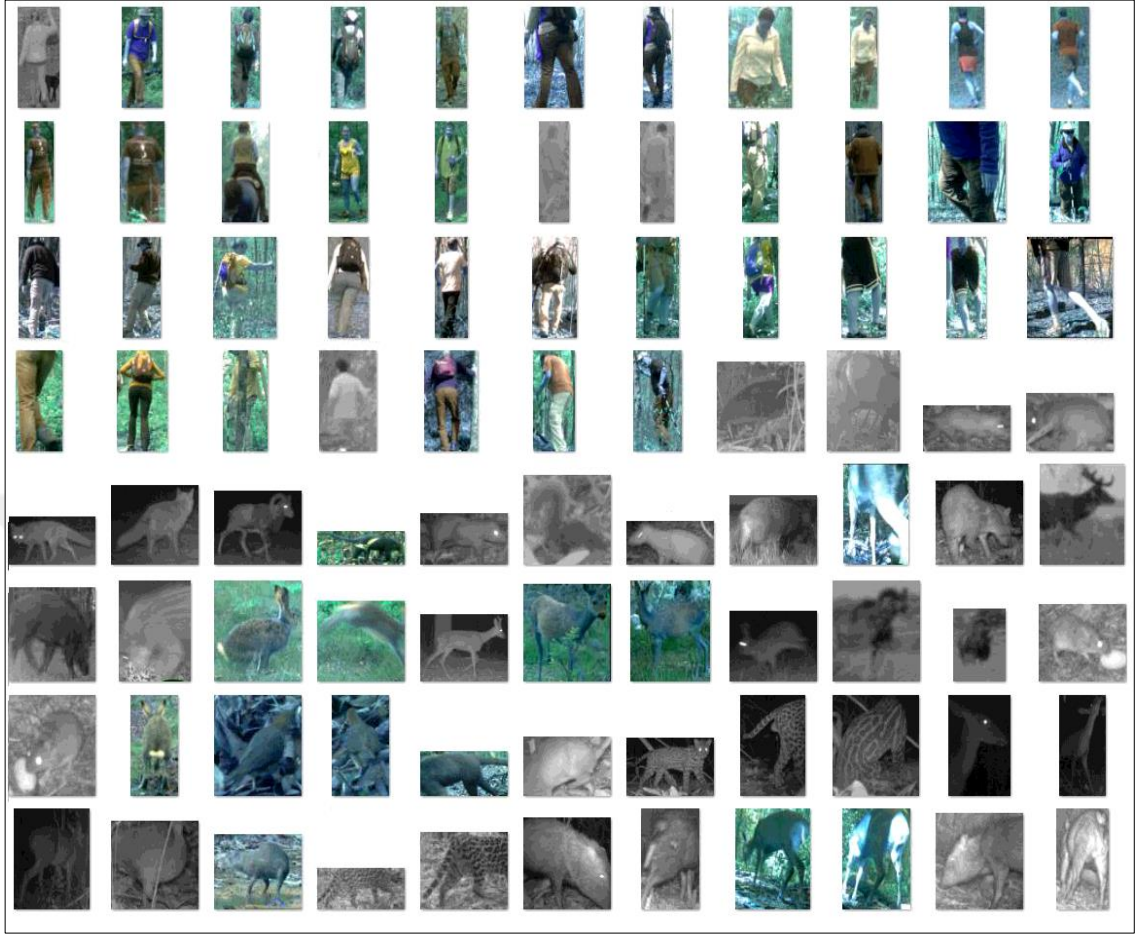
**Şekil 3.14.** Sınırlayıcı kutu tespiti

Çizelge 3.5’te foto-kapan veri setleri üzerinde yapılan çalışmalarda, elle kırılmış cisim görüntülerinin sayısı, veri setindeki cismin sınıf sayısı, çalışmalarda tespit edilen sınıf sayısı ve bu tespit işlemlerine ait başarı oranları gösterilmektedir. Görüldüğü üzere, elle cisim tespiti yapılmış görüntü sayısı arttıkça veya veri setine kıyasla tespit edilmeye çalışılan sınıf sayısı azaldıkça başarı oranı artmaktadır. Sınıf sayısı 2 ve 3 olan çalışmalarda sınıflar arka plan, insan ve hayvan olarak uygulanmaktadır. Sınıf sayısının fazla olduğu çalışmalarda ise derin öğrenme tabanlı yöntemlerle farklı türde hayvanlar tespit edilmeye çalışılmaktadır.

**Çizelge 3.5.** Yapılan çalışmalarda el ile cisim tespiti yapılan görüntü sayıları, sınıf sayıları ve tespit başarı oranları

<b>Çalışma</b>	<b>Elle Kırılan Görüntü Sayısı</b>	<b>Tespit Edilen Sınıf Sayısı</b>	<b>Veri Setindeki Sınıf Sayısı</b>	<b>Başarı Oranı %</b>
Andavarapu ve Vat., 2017	400	8	20	87-97
Zhang vd., 2016	6493	20	23	75-85
Norouzzadeh vd., 2018	1,4 milyon	48	48	92-99
Nguyen vd., 2017	107.000	6	18	92-95
Yu vd., 2013	7196	18	57	74-82
Buehler vd., 2019	1300	2	2	89-90
Willi vd., 2019	141.368	19	55	87-93
Yousif vd., 2019	459.427	3	>50	90-95
Gray vd., 2019	467	2	2	93
Gonçalves vd., 2016	14.547	13	13	86-97
Zhang vd., 2016	6.493	23	23	75-85

Şekil 3.15'te önerilen yöntem kullanılarak veri setlerinde elde edilmiş hareketli cisimlere ait çıktı görüntüleri gösterilmektedir. Tetikleme sonrasında kaydedilen görüntülerde ışık şiddeti değişimi ve arka plan hareketinin olması durumunda arka plan modelinin hatalı oluşturulduğu gözlemlenmiştir.



**Şekil 3.15.** Önerilen yöntem kullanılarak tespit edilen hareketli cisimlere ait görüntüler

Çizelge 3.6 ve Çizelge 3.7'de görüldüğü üzere, birinci veri setinde tespit aşamasında, %77-%82 ve %59-%79 arasında, konumlandırma aşamasında ise %55-%64 ve %41-%60 arasında doğruluk oranları elde edilmiştir. Histogram eşitlemenin kullanılmadığı Gaussların karışımı hariç diğer yöntemlerde, kabarcık analizinin konumlandırma aşamasında daha etkili olduğu anlaşılmaktadır. Cisim tespit aşamasında, Değişen Gauss Ortalama yönteminde Gaussların Karışımı yöntemine kıyasla %2-%5 daha iyi sonuçlar elde edilmiştir. Çizelge 3.6 ve Çizelge 3.7'den anlaşıldığı üzere, Histogram eşitleme işlemi Çizelge 3.6'daki kayan pencere tabanlı yöntem dışında konumlandırma aşamasında doğruluk oranını arttırmıştır.



**Çizelge 3.6.** Birinci veri setinde gauss bulanıklığı ile elde edilen tespit – konumlandırma (0,5-0,7-0,9) sonuçları (%)

Metot	Kayan Pencere Tabanlı Histogram	Kabarcık Analizi
Gauss Ortalama	81-(56-32-19)	<b>82</b> -(60-33-22)
Gaussların Karışımı	77-(58-33-18)	77-(57-32-20)
Gauss Ortalama + Histogram Eşitleme	80-(55-35-22)	81-( <b>64-41-28</b> )
Gaussların Karışımı + Histogram Eşitleme	79-(56-36-21)	79-(63-39-26)

**Çizelge 3.7.** Birinci veri setinde medyan filtre ile elde edilen tespit – konumlandırma (0,5-0,7-0,9) sonuçları (%)

Metot	Kayan Pencere Tabanlı Histogram	Kabarcık Analizi
Gauss Ortalama	<b>79</b> -(47-24-6)	%73-(49-24-15)
Gaussların Karışımı	68-(41-20-5)	%59-(37-18-10)
Gauss Ortalama + Histogram Eşitleme	78-(52-27-8)	%79-( <b>60-39-15</b> )
Gaussların Karışımı + Histogram Eşitleme	77-(51-27-9)	%74-(53-27-11)

Çizelge 3.8 ve Çizelge 3.9'da görüldüğü üzere, ikinci veri setinde tespit aşamasında,%75-%83 ve %72-%81 arasında, konumlandırma aşamasında ise %62-%80 ve %59-%73 arasında doğruluk oranları elde edilmiştir. İkinci veri setinde konumlandırma işleminde birinci veri setine kıyasla daha iyi doğruluk oranları elde edilmiştir. Histogram eşitleme işleminin cisim tespit aşamasında başarıyı %3-%7 arasında düşürdüğü, konumlandırma aşamasında ise başarıyı %4-%16 arasında arttırdığı anlaşılmaktadır. Ayrıca konumlandırma işleminde kabarcık analizinin kayan pencere tabanlı yöntemle göre daha başarılı olduğu gözlemlenmiştir. Çizelge 3.6, Çizelge 3.7, Çizelge 3.8 ve Çizelge 3.9'da görüldüğü üzere daha iyi konumlandırma başarısı anlamına gelen 0,7 ve 0,9 birleşimin kesişimi parametrelerinde cisim konumlandırma işleminde doğruluk oranlarının düşük olduğu anlaşılmaktadır. Bu sonuca arka plan çıkarma işlemindeki yetersizliklerin, arka plan hareketlerinin ve kesin doğruluk

değerlerindeki küçük hataların sebep olduğu düşünülmektedir. Foto-kapan görüntülerinde arka plan hareketi ve ışık şiddeti değişimi problemi fazla olduğu için birleşimin kesişimi 0,5 değeri tercih edilmektedir (Yousif *et al.* 2019; Zhang *et al.* 2016).

**Çizelge 3.8.** İkinci veri setinde gauss bulanıklığı ile elde edilen tespit – konumlandırma (0,5-0,7-0,9) sonuçları (%)

Metot	Kayan Pencere Tabanlı Histogram	Kabarcık Analizi
Gauss Ortalama	<b>83</b> -(71-54-32)	82-(71-53-30)
Gaussların Karışımı	82-(62-49-25)	75-(64-49-23)
Gauss Ortalama + Histogram Eşitleme	76-(78- <b>59-34</b> )	76-( <b>80</b> -58-32)
Gaussların Karışımı + Histogram Eşitleme	79-(78-58-31)	74-(77-55-30)

**Çizelge 3.9.** İkinci veri setinde medyan filtre ile elde edilen tespit – konumlandırma (0,5-0,7-0,9) sonuçları (%)

Metot	Kayan Pencere Tabanlı Histogram	Kabarcık Analizi
Gauss Ortalama	81-(66-50-28)	<b>81</b> -(69-50-24)
Gaussların Karışımı	80-(60-46-20)	74-(59-45-20)
Gauss Ortalama + Histogram Eşitleme	75-(70- <b>56-29</b> )	72-( <b>73</b> -53-26)
Gaussların Karışımı + Histogram Eşitleme	77-(69-54-29)	75-(70-52-27)

Medyan filtre, tuz ve karabiber gürültüsünü gidermede başarılı olsa da, cismin silinmesine veya belirginliğinin azalmasına neden olabilmektedir. Cisim algılama performansını arttırmak için, gürültünün tamamen temizlenmesi yerine yalnızca eşik değerinin altına indirmek yeterli olmaktadır. Çizelge 3.6, Çizelge 3.7, Çizelge 3.8 ve Çizelge 3.9’da gösterildiği gibi, medyan filtre tespit aşamasını olumlu, konumlandırma aşamasını ise olumsuz etkilemiştir.

Çizelge 3.10 ve Çizelge 3.11’de iki veri setindeki arka planın cisim olarak etiketlenme (False Pozitive) ve cismin arka plan olarak etiketlenme (False Negative) oranları verilmiştir. Çizelge 3.10’dan anlaşıldığı gibi, histogram eşitleme, cismin arka plan olarak etiketlenme oranını düşürdüğü böylece konumlandırma işleminde genellikle başarıyı arttırdığı ancak arka planın cisim olma oranı da arttırdığı anlaşılmaktadır. Çizelge 3.11’de görüldüğü üzere, Çizelge 3.12’ye kıyasla medyan filtre gürültülerin temizlenmesinde daha etkili olarak hatalı tespitleri (False Pozitive) azalttığı, buna karşılık cisim varlığı ile ilgili bilgileri silerek cisim tespitini (False Negative) zorlaştırdığı anlaşılmaktadır. Medyan filtre gauss bulanıklığından başarısız olsa da, gauss bulanıklığının başarılı olamadığı bazı görüntüler için yararlı olmaktadır.

**Çizelge 3.10.** Gauss bulanıklaştırma hata matrisi (false negative – false pozitive) (%)

Metot	Gauss Ortalama	Gaussların Karışımı	Gauss Ortalama + Histogram Eşitleme	Gaussların Karışımı + Histogram Eşitleme
Kayan Pencere Tabanlı Histogram	11-10	18-10	1-22	2-23
Kabarcık Analizi	10-12	13-14	2-20	3-21

**Çizelge 3.11.** Her iki veri setine ait medyan filtre hata matrisi (false negative – false pozitive) (%)

Metot	Gauss Ortalama	Gaussların Karışımı	Gauss Ortalama + Histogram Eşitleme	Gaussların Karışımı + Histogram Eşitleme
Kayan Pencere Tabanlı Histogram	16-5	29-10	3-20	8-19
Kabarcık Analizi	27-4	38-8	4-20	13-18

Çizelge 3.12'den anlaşıldığı üzere, tespit ve konumlandırma işlemlerinde medyan filtre, gauss bulanıklığının faydalı olmadığı görüntülerin %2-%8'i için faydalı olmuştur.

**Çizelge 3.12.** Veri setlerindeki gürültülerin giderilmesinde gauss bulanıklaştırmanın başarısız, medyan filtrenin başarılı olduğu görüntülerin oranı (%)

Metot	Gauss Ortalama	Gaussların Karışımı	Gauss Ortalama + Histogram Eşitleme	Gaussların Karışımı + Histogram Eşitleme
Kayan Pencere Tabanlı Histogram	9	3	2	4
Kabarcık Analizi	6	4	2	7

### 3.2.2. Hareketli cisim tespitinin örüntü tanıma yöntemleri ile iyileştirilmesi

Bu çalışmada foto-kapan görüntülerinde arka plan çıkarma ve çerçeve farkı yöntemleri ile hareketli cisim tespitinde karşılaşılan arka plan hareketi ve ışık şiddeti değişimini tespit etmek amacıyla sınıflandırma tabanlı yeni bir sistem önerilmiştir. Çalışmamızda öncelikle değişen gauss ortalama ve çerçeve farkı yöntemi uygulandığında elde edilen sonuç uygun, arka plan hareketi veya ışık şiddeti değişimi olmak üzere 3 sınıfa ayrılmaktadır. Işık şiddeti değişimi olan görüntülerde histogram eşitleme, arka plan hareketi olan görüntülerde farklı arka plan çıkarma yöntemi ve medyan filtre uygulanmıştır. Elde edilen çıktı uygun olarak sınıflandırıldığında ise hareketli cisim tespitindeki sonraki aşama olan hareketli cismin konumunun belirlenmesi işlemine geçilmiştir. Görüntülerin sınıflandırılmasında Yerel İkili Örüntü (LBP) öznitelikleri ile rastgele orman örüntü tanıma algoritması kullanılmıştır.

Çalışmamızda doğal ortamında hayvan türlerini tespit etmek amacıyla yapılan çalışmada (Kays *et al.* 2010) foto-kapanlar kullanılarak elde edilmiş Şekil 3.16'da örnek görüntüleri bulunan bir veri seti kullanılmıştır. Bu veri setinde 100 farklı tetiklemeye ait

1276 görüntü bulunmaktadır. Veri setindeki 947 görüntüde hareketli cisim bulunmaktadır, 329 görüntüde ise hareketli cisim bulunmamaktadır. Veri seti üzerinde hareketli cisim tespiti için önerilen değişen gauss ortalama, çerçeve farkı ve OTSU'nun uygulandığı yöntem (Şimşek *et al.* 2018a) kullanılarak sonuç görüntüleri elde edilmiştir ve tarafımızca el ile seçilerek 3 gruba ayrılmıştır.

Veri setindeki görüntüler hareketli ve karmaşık arka plan ile ışık şiddeti değişimi gibi zorluklar içermektedir. Veri setinde ayrıca, cisimlerin çok sayıda ve parçalı olması, her tetiklemede az sayıda görüntü olması gibi arka plan modelinin hatalı elde edilmesine neden olan zorluklar bulunmaktadır.

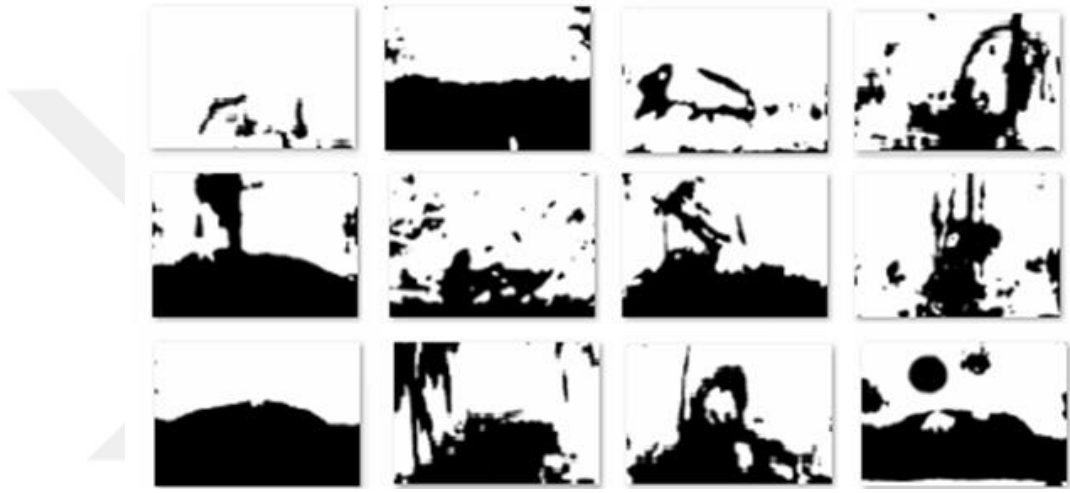


**Şekil 3.16.** Örnek foto-kapan görüntüleri

Arka plan çıkarma, sabit kameralardan alınan görüntülerde hareketli cisimleri tespit için yaygın olarak kullanılan bir yaklaşımdır (Piccardi, 2004). Bu çalışmada en az bellek ve işlem gereksinimi olan Değişen Gauss Ortalaması ile arka plan hareketinin olduğu görüntülerde daha iyi sonuçlar üreten ancak daha fazla bellek ve işlem gereksinimi olan Gaussların Karışımı birlikte kullanılmıştır.

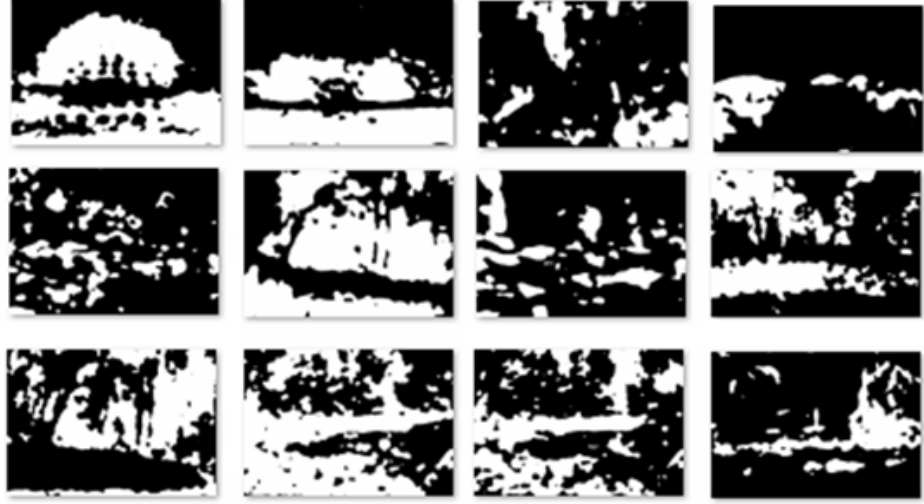
Hareketli cisim tespitinde dinamik ve karmaşık arka plan, ışık değişimi gibi temel zorluklar bulunmaktadır (Karasulu and Korukoglu 2013). Bu zorluklar arka plan

modelinin hatalı olarak elde edilmesine ve dolayısıyla çerçeve farkı ile cisim tespitinde hatalı sonuçlar elde edilmesine neden olmaktadır. Çalışmada kullanılan veri setinde arka plan modeli kullanılarak çerçeve farkı ile elde edilen hatalı sonuçlar Şekil 3.17 ve Şekil 3.18'de, uygun sonuçlar ise Şekil 3.19'da gösterilmektedir. Şekil 3.17, Şekil 3.18 ve Şekil 3.19'da siyah pikseller arka planı, beyaz pikseller ise hareketli cisimleri temsil etmektedir.



**Şekil 3.17.** Uygulanan yöntemde ışık şiddeti değişimi nedeniyle hatalı olarak etiketlenen sonuçlar

Hatalı sonuçlara neden olan arka plan karmaşıklığı veya arka plan hareketi problemi, daha iyi bir arka plan çıkarma yöntemi ve farklı gürültü temizleme filtreleri ile çözülebilmektedir. Görüntüler arasındaki ışık şiddeti değişiminin çözülmesi için ise histogram eşitleme kullanılmaktadır. Hatalı ve uygun olarak etiketlenen çıktı görüntülerinin benzer örüntülere veya özelliklere sahip olduğu Şekil 3.17, Şekil 3.18 ve Şekil 3.19'dan anlaşılabilir.



**Şekil 3.18.** Uygulanan yöntemde arka plan hareketi nedeniyle hatalı olarak etiketlenen sonuçlar

Şekil 3.17’te gösterilen ışık şiddeti değişimi içeren görüntülerde, hareketli cisim olarak belirlenmiş beyaz piksellerin bütüncül ve büyük boyutta olduğu anlaşılmaktadır. Şekil 3.18’de gösterilen arka plan hareketi içeren hatalı görüntülerde ise Şekil 3.17’ye göre kıyasla hareketli cisim olarak etiketlenen beyaz piksellerin genellikle çok sayıda, küçük parçalı ve bölgesel olduğu gözlemlenmektedir. Şekil 3.19’da gösterilen uygun çıktı görüntülerinde ise hareketli cisimler net olarak anlaşılabilir.



**Şekil 3.19.** Uygun olarak etiketlenen sonuçlar

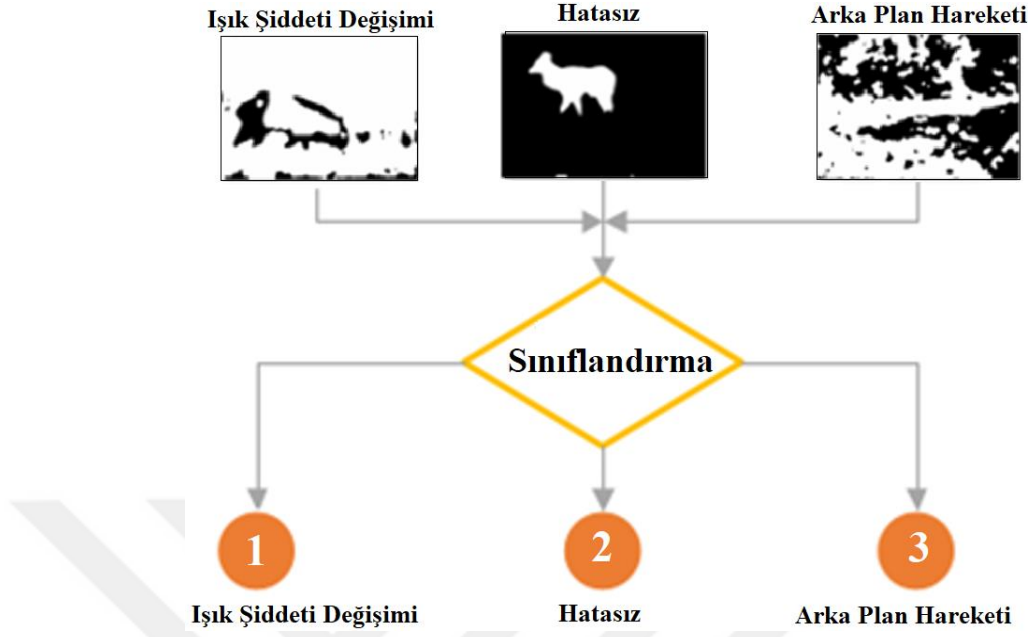
Bu veri setinden elde edilmiş hatalı ve uygun olarak kaydedilen 2046 çıktı görüntüsünden 1343 tanesi uygun, 401 tanesi ışık şiddeti değişimi, 299 tanesi ise arka plan hareketi olmak üzere Şekil 3.20’de gösterildiği üzere üç sınıfa ayrılmıştır. Bu üç gruba ait görüntülerin sınıflandırılması için LBP(Yerel İkili Örüntü) öznelikleri ve Rastgele Orman sınıflandırma algoritması kullanılmıştır. %80 eğitim ve %20 test verisi olarak kullanılan görüntülerde Rastgele Orman sınıflandırma başarısı % 96,33 olarak elde edilmiştir. Rastgele orman algoritmasına ait hata matrisi Çizelge 3.13’te gösterilmiştir.

**Çizelge 3.13.** rastgele orman sınıflandırma işlemi hata matrisi

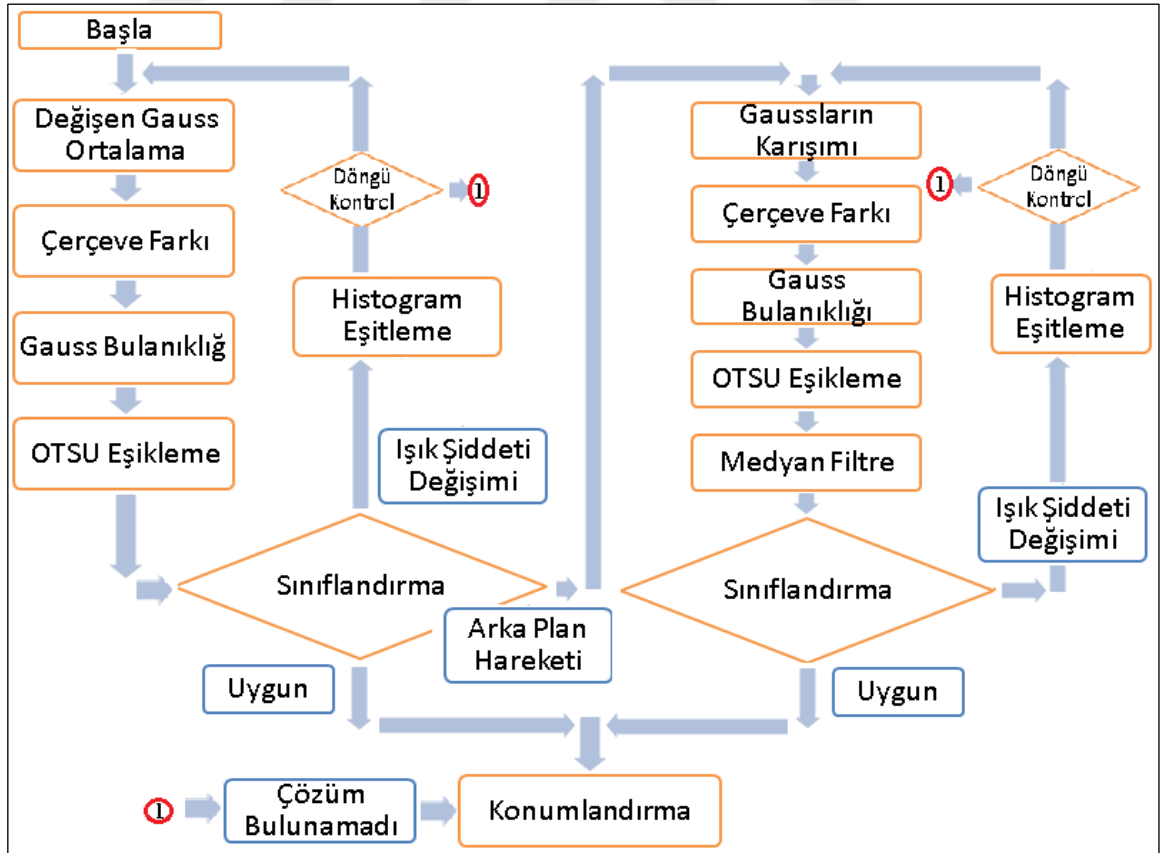
		<b>Gerçek Sınıflar</b>		
		Uygun	Arka Plan Hareketi	Işık Şiddeti Değişimi
<b>Tahmin Edilen Sınıflar</b>	Uygun	259	6	2
	Hareketli Arka Plan	3	50	2
	Işık Şiddeti Değişimi	0	2	85

Şekil 3.21’de gösterilen sistem modelinde olduğu gibi, öncelikle her tetiklemede elde edilen görüntülerde standart olarak Değişen Gauss Ortalama, Çerçeve Farkı, Gauss Bulanıklaştırma ve OTSU eşikleme uygulanmaktadır. Bu yöntemlerin sonuçlarına ait çıktılar rastgele orman algoritması ile uygun, arka plan hareketi veya ışık şiddeti değişimi olarak sınıflandırılmaktadır. Arka plan hareketi olarak sınıflandırılan çıktı için, görüntülere Gaussların Karışımı ve medyan filtre, ışık şiddeti olarak sınıflandırılan çıktıda ise ön işleme olarak histogram eşitleme uygulanmaktadır. Sınıflandırma işlemi bir görüntü için en fazla 2 iterasyon uygulanmaktadır.





Şekil 3.20. Klasik yöntemle elde edilen sonuçların sınıflandırılması



Şekil 3.21. Örüntü tabanlı iyileştirme için önerilen sistem modeli

Uygun olarak sınıflandırılan veya 2 iterasyonda çözüm bulunamayan çıktılar için çözüm bulunamadığı kabul edilip hareketli cismin görüntülerdeki konumlarının bulunması aşamasına geçilmektedir. Bu amaçla kullanılan konumlandırma yöntemi olan kayan pencere ve kabarcık analizi yöntemi uygulanmıştır. Çalışmada hatalı sonuçlar elde edilmesine neden olan problemler önceden belirlenerek hareketli cisim tespit başarısı artırılmıştır.

Yapılan uygulamada standart olarak kullanılan Değişen Gauss Ortalama yönteminde 0,5 parametre değeri, Gauss Bulanıklaştırmada 65 parametre değeri kullanılmıştır. Rastgele Orman algoritması için tahmin parametresi olarak 50, LBP özniteliği için yarıçap değeri olarak 3 piksel uygulanmıştır.

Yapılan çalışmada önerilen yöntem sonuçlarına göre hatalı çıktı elde edilmesi tamamen arka plan modeli doğru oluşturulmasına bağlı olduğu anlaşılmaktadır. Bu veri setinde üzerinde yapılan çalışmada (Şimşek *et al.* 2018a) elde edilen hareketli cisim tespit sonuçları Çizelge 3.14'te gösterilmektedir. Bu çalışmada ön işleme olarak histogram eşitleme, arka plan çıkarma yöntemi olarak Değişen Gauss Ortalama ve Gaussların Karışımı cismin görüntüdeki konumunun belirlenmesi için ise kayan pencere ve kabarcık analiz yöntemleri kullanılmıştır. Yapılan çalışmada arka planın karmaşık, hareketli ve ani ışık değişimi bulunması nedeniyle cisim tespitinin zor olduğu ifade edilmektedir.

**Çizelge 3.14.** Klasik yöntemler ile elde edilmiş sonuçlar (Şimşek *et al.* 2018a)

	<b>Kayan Pencere</b>		<b>Kabarcık Analizi</b>	
	Tespit	Konum Belirleme	Tespit	Konum Belirleme
<b>Değişen Gauss Ortalama</b>	%80,53	<b>%55,41</b>	%80,64	<b>%64,14</b>
<b>Gaussların Karışımı</b>	%79,31	<b>%56,23</b>	%79,00	<b>%63,33</b>

Çizelge 3.15'te önerdiğimiz yöntem ile elde edilmiş sonuçlar gösterilmiştir. Çizelge 3.14 ile kıyaslandığında, aynı veri setinde aynı yöntemler kullanılarak elde edilen en iyi sonuçların %11 ile %16 arasında artırılarak %71-%75 arasında sonuçlar elde edildiği görülmektedir.

**Çizelge 3.15.** Önerilen yöntem kullanılarak iyileştirilmiş sonuçlar

	<b>Kayan Pencere</b>		<b>Kabarcık Analizi</b>	
	Tespit	Konum Belirleme	Tespit	Konum Belirleme
<b>Sınıflandırma Tabanlı İyileştirme</b>	%80,80	<b>%71,01</b>	%80,56	<b>%75,09</b>

Çizelge 3.16'da görüldüğü üzere, görüntülerin %67 - %68'inde standart yöntemlerin uygulandığı, görüntülerin %26 - %27'sinde ışık şiddeti değişimi tespit edilerek histogram eşitleme uygulandığı, görüntülerin %4 - %5'inde arka plan hareketi tespit edilerek gaussların karışım ve medyan filtre uygulandığı anlaşılmaktadır. Görüntülerin yaklaşık % 1'inde ise bu yöntemler ile çözüm bulunamamıştır.

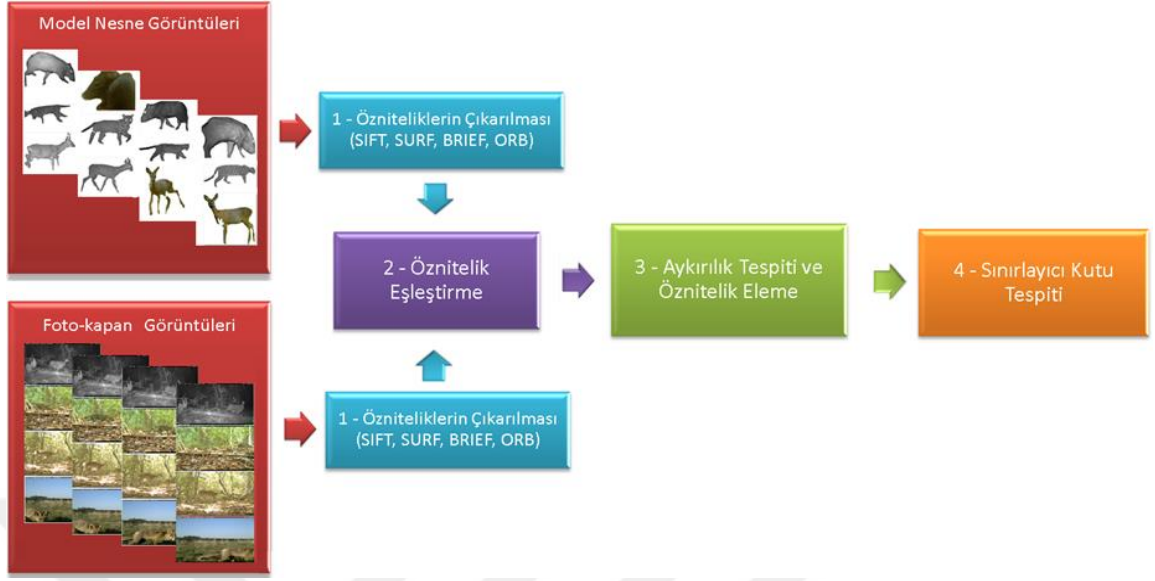
**Çizelge 3.16.** İyileştirmede kullanılan yöntemlerin sayısı

<b>İşlem Numarası</b>	<b>Kayan Pencere</b>		<b>Kabarcık Analizi</b>	
	<b>Çözüm Sayısı</b>	<b>Çözüm Yüzdesi (%)</b>	<b>Çözüm Sayısı</b>	<b>Çözüm Yüzdesi (%)</b>
<b>0</b>	863	67,6	862	67,5
<b>1</b>	344	26,9	356	27,8
<b>2</b>	61	4,7	53	4,2

### 3.2.3. Yerel öznitelikler ve öznitelik eşleştirme ile cisim tespiti

Literatürde yerel öznitelikler ile yapılan çalışmalarda, genellikle yerel öznitelik dönüşümleri birebir eşleştirilmek yerine, bunların istatistikî bilgilerini içeren görsel kelime çantası yaklaşımları tercih edilmektedir. Böyle uygulamalarda görüntüdeki öznitelikler görsel kelime çantası yöntemi ile gruplandırılıp,  $k$  en yakın komşu, destek vektör makineleri ve rastgele orman gibi sınıflandırıcılarda kullanılarak, kayan pencere tabanlı yöntemle, her pencerede ilgili cismin olup olmadığına bakılarak tespit işlemi gerçekleştirilmektedir. Bu çalışmada, kayan pencerelerde sınıflandırma ve öznitelik eşleştirme tabanlı yöntemlerle gerçekleştirilen cisim tespitinden farklı olarak, görüntünün tamamında öznitelik eşleştirme, aykırılık tespiti ve öznitelik eleme yöntemleri ile cisim tespiti gerçekleştirilmiştir.

Foto-kapan görüntülerinde yerel öznitelikler kullanılarak cisim tespit için kullandığımız dört aşamadan oluşan sistem modeli Şekil 3.22'da gösterilmiştir. Öncelikle veri setinde tespit edilecek cisimlere ait model görüntüler, Şekil 3.24'te görüldüğü üzere kırılarak ve arka plandan ayrılarak elde edilmiştir. Birinci aşamada veri setindeki görüntülerin SIFT, SURF, BRIEF ve ORB yöntemleri ile öznitelikleri çıkarılmaktadır. İkinci aşamada görüntülerden elde edilmiş yerel öznitelikler kaba kuvvet (brute-force) ve  $k$  en yakın komşu tabanlı yöntem ile eşleştirilmektedir. Üçüncü aşamada yüzdellik veya ortalama tabanlı aykırılık tespiti ile  $k$  en yakın komşu öznitelik eleme yöntemleri kullanılarak hatalı eşleşmelerin giderilmesi ve elenmesi sağlanmaktadır. Dördüncü aşamada eleme işlemi sonucunda kalan özniteliklerin konum bilgileri kullanılarak tespit edilen cisme ait sınırlayıcı kutular belirlenmektedir.

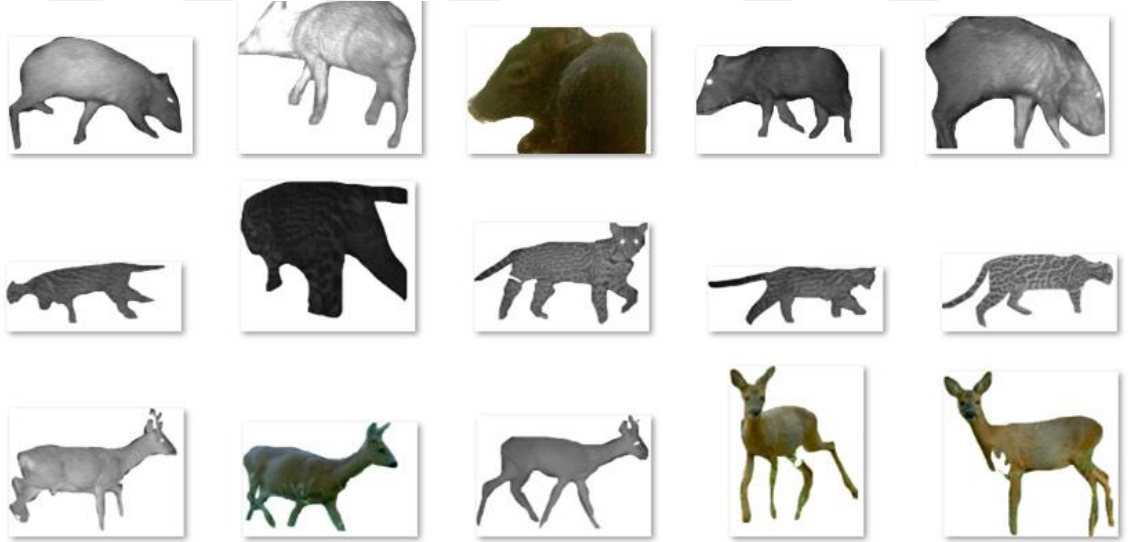


**Şekil 3.22.** Öznitelik eşleştirme tabanlı cisim tespiti için önerilen sistem modeli

Çalışmamızda kullanılan veri setinde 20 farklı hayvan türüne ait 2048x1536 piksel boyutlarında toplam 1016 görüntü bulunmaktadır (Kays *et al.* 2010). Şekil 3.23'te veri setinden rasgele seçilmiş gece ve gündüz kaydedilmiş görüntüler gösterilmektedir. Gece görüntüleri kızıl ötesi flaş kullanılarak kaydedilmiştir. Veri setindeki görüntülerde cisimler farklı ortam, uzaklık ve açılardan elde edilmiş olup tam veya parçalı olabilmektedir. Şekil 3.24'te örnek kırılmış görüntüler gösterilmiştir. Cisim tespit işleminde her tetikleme için 1 model olmak üzere 20 farklı türe ait 79 model cisim görüntüsü kullanılmıştır.



**Şekil 3.23.** Veri setine ait örnek görüntüler



**Şekil 3.24.** Cisimlere ait kırılmış ve arka plandan ayrılmış örnek görüntüler

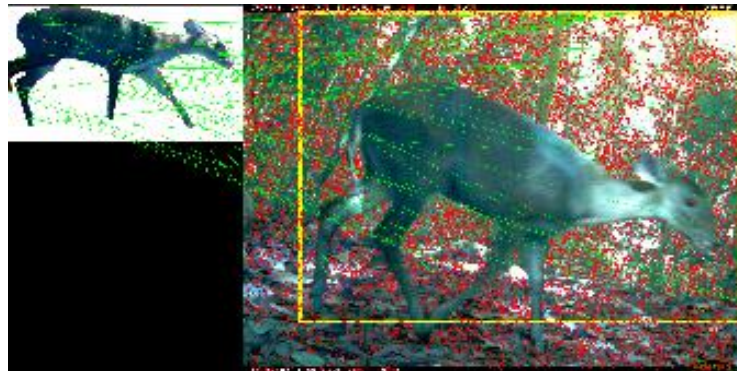
Öznitelik tespiti, görüntü bilgisinin anlamlandırılması ve bu noktada belirtilen tipte bir görüntü özelliğinin olup olmadığını görmek için her bir görüntü noktası için yerel bir karar verme işlemidir. Yerel öznitelikler renk, doku, şekil gibi düşük seviyeli

özniteliklerden farklı olarak görüntüdeki dönme, ölçek, aydınlatma, gürültü ve afin dönüşümü için dayanıklı değildir (Karami *et al.* 2017). Bir dizi yerel öznitelik, cisimleri ve sahneleri tanıyabilmek için bölütleme işlemine ihtiyaç duyulmadan sağlam bir görüntü temsili olarak kullanılabilir (Tuytelaars and Mikolajczyk 2008). Çalışmamızda bu yerel öznitelik dönüşüm yöntemlerinden SIFT, SURF, BRIEF ve ORB Çizelge 3.17’de gösterilen algoritma ile uygulanmıştır.

**Çizelge 3.17.** Öznitelik eşleştirme algoritması

<p><i>Veri Setindeki Her Görüntü İçin:</i></p> <p><i>Bütün Model Cisim Görüntüleri İçin:</i></p> <p><i>Tanımlayıcı 1 = yerel öznitelikleri tanımla (Model Cisim Görüntüsü)</i></p> <p><i>Tanımlayıcı 2 = yerel öznitelikleri tanımla (Foto-kapan Görüntüsü)</i></p> <p><i>Eşleşmeler = kaba kuvvet eşleştirici(Tanımlayıcı 1, Tanımlayıcı 2)</i></p>
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Öznitelik eşleştirme işlemi görüntülerde benzer iki cisim arasında mutlak doğrulukta bir eşleşme ve tespit sağlayamamaktadır. Şekil 3.25’te gösterildiği üzere kırılmış görüntü ile Foto-kapandan elde edilen görüntü eşleşmesinde yeşil renkli pikseller eşleştirilmiş öznitelik noktalarını, kırmızı renkli pikseller ise eşleşmemiş öznitelik noktalarını temsil etmektedir. Görüntüdeki en benzer özniteliklerin eşleştirilmesi işlemi, bazen cisme ait olmayan kısımların hatalı bir biçimde cisme ait olarak etiketlenmesine, bu nedenle tespit işleminin hatalı yapılmasına neden olmaktadır.



**Şekil 3.25.** Öznitelik eşleştirme sonucuna ait örnek görüntü

Şekil 3.25'teki sarı sınırlayıcı kutunun cismin üzerindeki arka plan bölgesini de kapsadığı, yani bazı özniteliklerin hatalı olarak arka plandaki özniteliklerle eşleştiği anlaşılmaktadır. Bu hatalı özniteliklerin giderilmesi amacıyla çalışmamızda kaba kuvvet eşleştirici ile birlikte yüzdellik ve ortalama tabanlı aykırılık tespiti ile k en yakın komşu eşleştirici eşik değeri ile birlikte kullanılarak öznitelik eleme işlemleri uygulanmıştır. Aşağıdaki sahte kodda gösterildiği üzere, tanımlayıcılardan eşleştirilmiş her m ve n eşleşme çiftinin uzaklıkları oranlanmakta ve bu oran belirli bir değerin üzerinde ise eşleşme kabul edilmektedir. Öznitelik seçme ve eleme işlemine ait algoritma Çizelge 3.18'de gösterilmektedir.

**Çizelge 3.18.** K en yakın komşu öznitelik seçme ve eleme algoritması

*Veri Setindeki Her Görüntü İçin:*

*Bütün Model Cisim Görüntüleri İçin:*

*Tanımlayıcı 1 = yerel öznitelikleri tanımla (Model Cisim Görüntüsü)*

*Tanımlayıcı 2 = yerel öznitelikleri tanımla (Foto-kapan Görüntüsü)*

*Eşleşmeler = k en yakın komşu eşleştirici(Tanımlayıcı 1, Tanımlayıcı 2, k=2)*

*Her m, n eşleşmesi için:*

*Eğer m'nin mesafesi < [0.80, 0.85, 0.90, 0.95, 0.99]\*n'nin mesafesi ise*

*Eşleşmeyi kabul et*

*Değil ise*

*Eşleşmeyi ele*

Ortalama tabanlı aykırılık tespitinde kullanılan ortanca mutlak sapmanın (Median Absolute Deviation - MAD) nasıl hesaplandığı Denklem 17'de ifade edilmektedir. Hesaplanan ortanca mutlak sapma değerlerinden, parametrede girilen eşik değerden büyük olanlar aykırılık olarak tespit edilmektedir.

$$MAD = \text{medyan}(|x - \text{medyan}(x)|) \quad (17)$$



Yüzdelik tabanlı aykırılık tespitinde amaç, sıralı bir dizide girilen parametredeki konumlara karşılık gelen elemanlarının değerlerini kullanarak maksimum ve minimum eşik değerleri belirlemektir. Dizide bu minimum ve maksimum değerlerin dışındaki elemanlar aykırılık olarak tespit edilmektedir.

Sınırlayıcı kutu tespiti, aykırılık tespiti veya öznitelik eleme işlemi sonrasında kalan eşleştirilmiş özniteliklerin yerel (x,y) konum bilgilerinin kullanılması ile elde edilmektedir. Eşleşen özniteliklerin cismi temsil ettiği varsayıldığından, bu eşleşen özniteliklerin en küçük ve en büyük (x,y) noktalarının cismi kapsadığı kabul edilmektedir. Sınırlayıcı kutu tespitinde başarı ölçümü amacıyla birleşimlerin kesişimi (IoU-Intersection Over Union) (Zitnick and Dollár 2014) formülü 0,5 parametresi ile kullanılmıştır. İki sınırlayıcı kutu için konumlandırma işleminin başarısı, Denklem 18’de gösterildiği üzere bu iki kutunun kesiştiği bölgenin piksellerinin, iki kutunun birleşimi olan bölgenin piksellerine oranı 0,5 ve üzerinde ise doğru kabul edilmiştir.

$$IoU = \frac{B_P \cap B_G}{B_P \cup B_G} \quad (18)$$

Bu çalışmada SIFT, SURF, BRIEF ve ORB yerel öznitelikleri, 2 farklı öznitelik eşleştirme yönteminde kullanılarak foto-kapan görüntülerinde cisim tespiti gerçekleştirilmiş olup, kaba kuvvet eşleştirici ile elde edilen sonuçlar Şekil 3.27 ve Şekil 3.30’da, k en yakın komşu eşleştirici ile elde edilen sonuçlar ise Şekil 3.33’te gösterilmiştir. Aykırılık tespiti ve öznitelik eleme işlemleri, hatalı eşleşmelerin giderilmesi ve sınırlayıcı kutu tespitinin daha doğru yapılmasını sağlamıştır. Hatalı öznitelik eşleşmelerinin elenmesinden sonra, tespit edilen sınırlayıcı kutuların boyutları azalmış ve daha yüksek tespit başarısı elde edilmiştir. Eşleştirilen öznitelik sayısı ile sınırlayıcı kutu büyüklüğünün ilişkili olduğu ve tespit başarısını etkilediği gözlemlenmiştir. Çizelge 3.19’da öznitelik tanımlayıcılarının boyutları gösterilmektedir. Tanımlayıcıların, özniteliklerin eşleştirilmesinde etkili olduğu ve bu nedenle bu boyutların tespit sonuçlarını arttırdığı anlaşılmaktadır.

Şekil 3.26’te k en yakın komşu eşleştiricide, birinci görüntüdeki öznitelik tanımlayıcısı ile ikinci görüntüde bu öznitelik ile eşleşen en yakın 2 öznitelik tanımlayıcısının mesafelerinin birbirine olan oranlarının 0,99, 0,95, 0,90, 0,85, 0,80 eşik değerlerine göre özniteliklerin elenmesi ve sınırlayıcı kutudaki değişiklikler gösterilmektedir. Bu oran azaldıkça sınırlayıcı kutular gerçek değerlerine yaklaşmakta ve tespit başarısı artmaktadır. Şekil 3.23’de öznitelik eleme işleminde en iyi sonuç SIFT özniteliğinde 0,85 parametresi ile elde edilmiştir. K en yakın komşu eşleştiricisinde, en yakın 2 komşunun elde edilmesi amacıyla  $K=2$  parametresi kullanılmıştır.



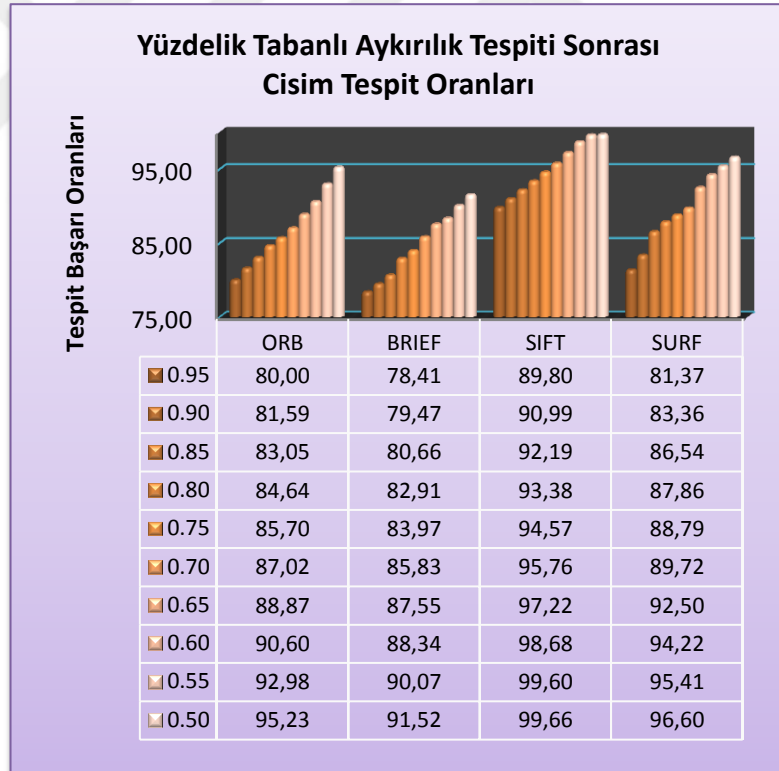
**Şekil 3.26.** k en yakın komşu eşleştirme işlemi ile eşik değere göre öznitelik eleme sonrası sınırlayıcı kutu değişimi

Çalışmamızda kaba kuvvet eşleştirme işlemi, yüzdelik ve ortalama tabanlı aykırılık tespiti yöntemi ile birlikte kullanılmıştır. Ayrıca k en yakın komşunun belirlendiği ve bu

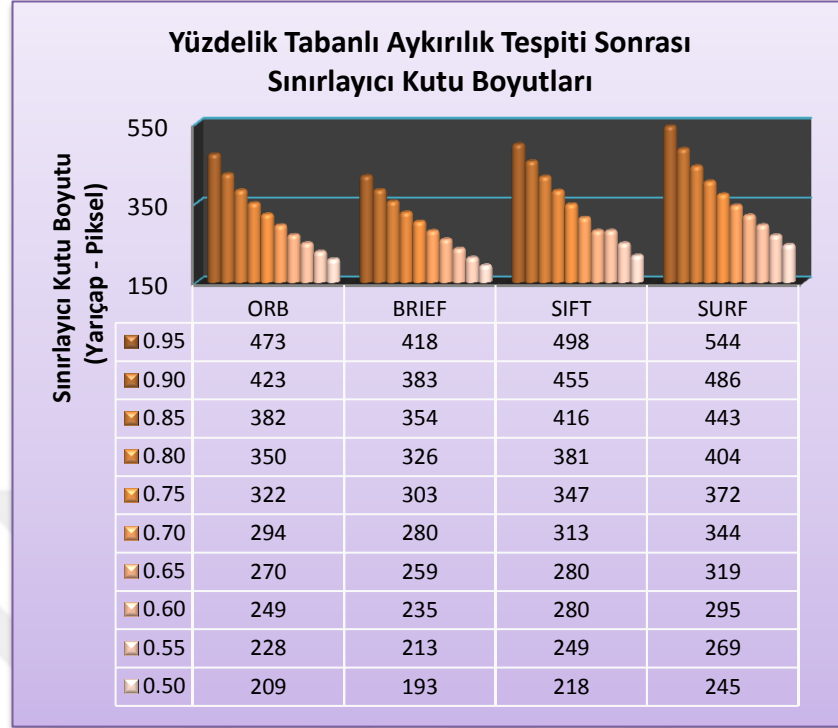
komşular arası mesafenin birbirleri ile oranlanarak öznelik eleme işleminin yapılabildiği k en yakın komşu eşleştirme yöntemi de K=2 parametresi ile de kullanılmıştır.

**Çizelge 3.19.** Özneliklerin tanımlayıcı boyutları, çıkarma ve eşleştirme süreleri

	<b>SIFT</b>	<b>SURF</b>	<b>BRIEF</b>	<b>ORB</b>
<b>Tanımlayıcı Uzunluğu</b>	128	<b>64</b>	256	256
<b>Tanımlayıcı Boyutu (Byte)</b>	512	256	<b>32</b>	<b>32</b>
<b>Ortalama Çıkarma Süresi (1 Öznelik - Mikrosaniye)</b>	2828	398	<b>329</b>	518
<b>Ortalama Eşleştirme Süresi (1 Öznelik - Mikrosaniye)</b>	168	103	<b>11</b>	12

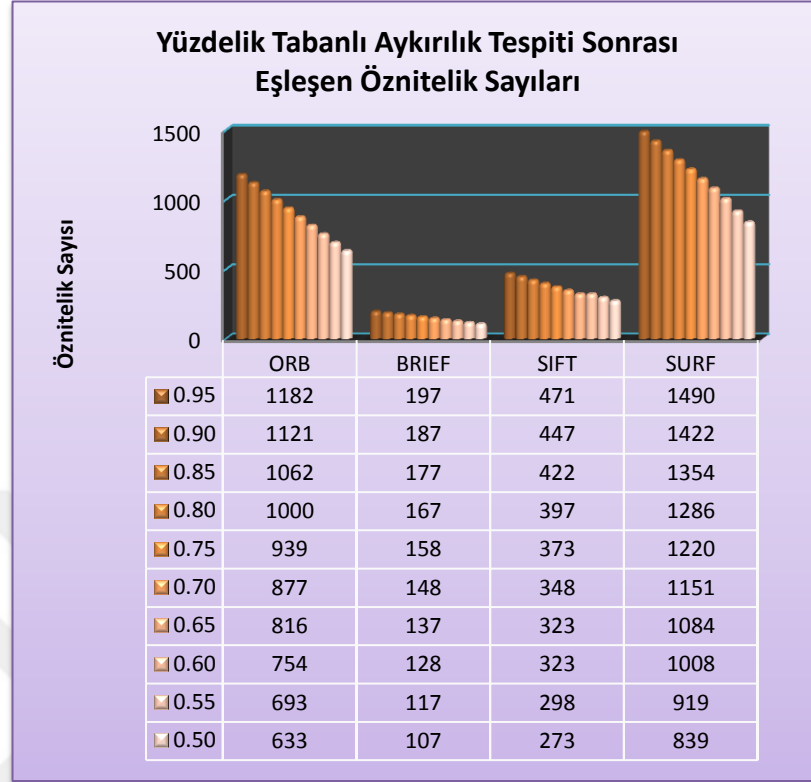


**Şekil 3.27.** Yüzdeleri tabanlı aykırılık tespiti sonrası cism tespiti başarı oranları



**Şekil 3.28.** Yüzdeler tabanlı aykırılık tespiti sonrası sınırlayıcı kutu boyutları

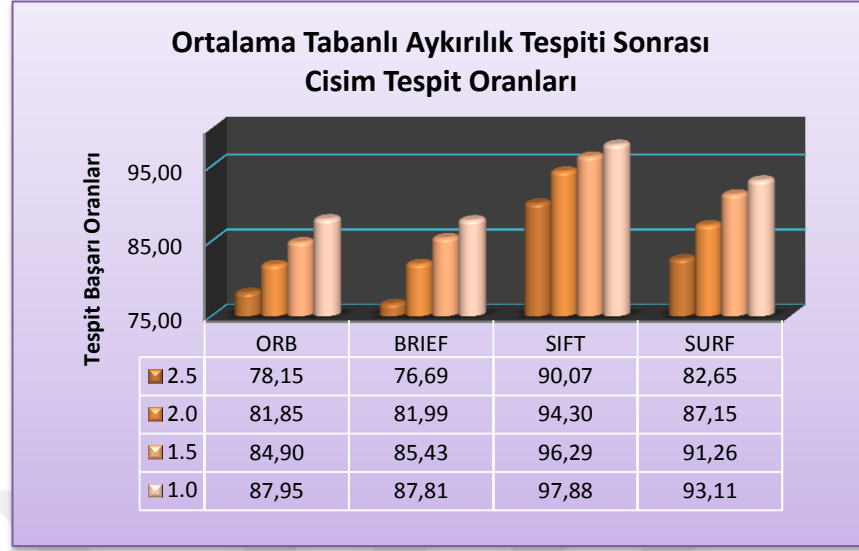
Yüzdeler tabanlı öznelik eleme işleminde, Şekil 3.27, Şekil 3.28 ve Şekil 3.29’da birinci sütunda gösterilen öznelik eleme parametre değeri azaldıkça, Şekil 3.26’daki eşleşen öznelik sayılarının azaldığı, buna bağlı olarak Şekil 3.28’deki sınırlayıcı kutuların gerçek değerlere yaklaştığı ve bunun sonucunda Şekil 3.27’deki cisim tespit başarısında artış elde edildiği anlaşılmaktadır. Şekil 3.27, Şekil 3.28 ve Şekil 3.29’da görüldüğü üzere, yüzdeler tabanlı aykırılık tespiti işleminde, eşleşmelerin yaklaşık yarısının elendiği ve bu durumun cisim tespit başarısını %10-%15 arası arttırdığı anlaşılmaktadır.



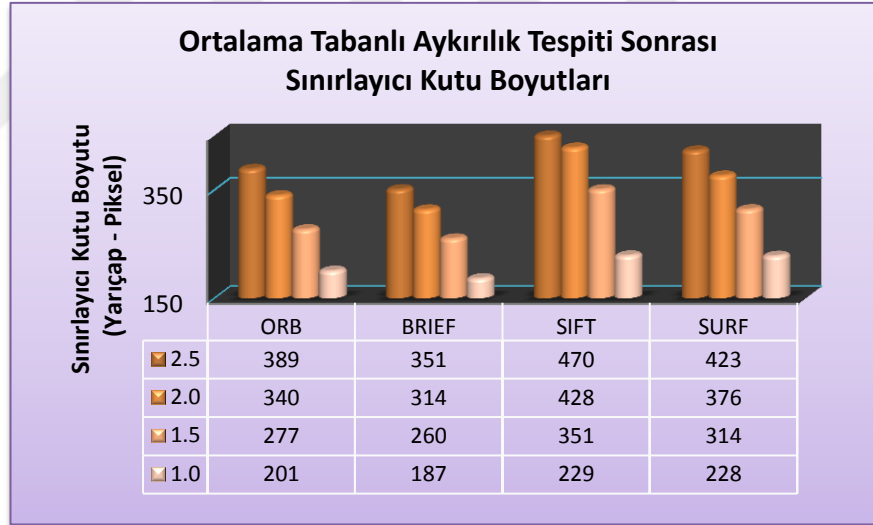
**Şekil 3.29.** Yüzdeler tabanlı aykırılık tespiti sonrası eşleşen öznitelik sayıları

Bütün sonuçlarda en yüksek cisim tespit başarısı bütün öznitelik eleme yöntemlerinde SIFT özniteliği ile elde edilmiş olup, ona en yakın sonuçlar ise SURF özniteliğinde elde edilmiştir. Çizelge 4.12’de görüldüğü üzere, en yüksek boyutlu öznitelik tanımlayıcısı olan SIFT ile en yüksek tespit başarısı elde edilmiştir. Diğer dönüşüm yöntemleri, öznitelik dönüştürme veya eşleştirme işlemi hızlandırmayı veya öznitelikleri daha az boyutta temsil etmeyi amaçladığından daha düşük tespit başarısı elde edilmiştir.

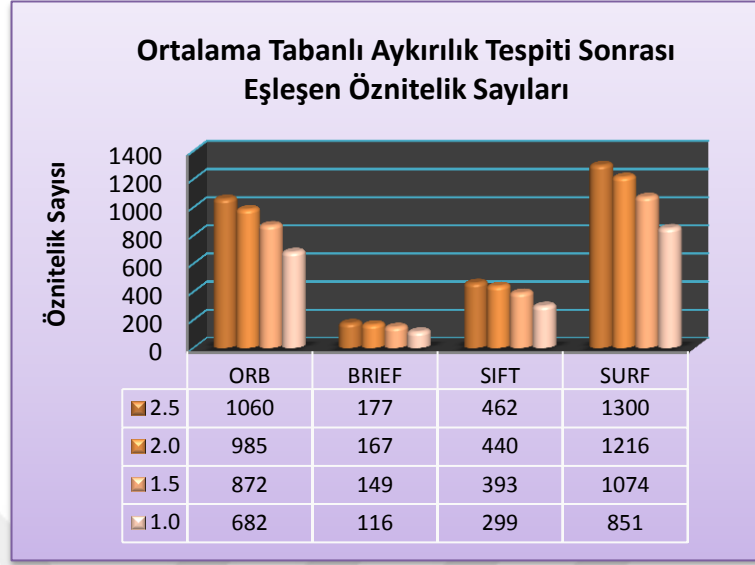
Şekil 3.32’den anlaşıldığı üzere ortalama tabanlı aykırılık tespitinde, öznitelik eşleşmelerinin %35-%40’ı elenmiştir. Yüzdeler tabanlı eleme yöntemine göre daha az öznitelik elenmiş ve buna bağlı olarak Şekil 3.30’daki gösterilen cisim tespit başarısında Şekil 3.27’teki yüzdeler tabanlı yöntemine göre daha düşük tespit başarısı elde edilmiştir. Şekil 3.31’de görüldüğü üzere ortalama tabanlı aykırılık tespitinde elde edilen sınırlayıcı kutuların yüzdeler tabanlı aykırılık tespitinde elde edilenlere göre daha küçük boyutlu olduğu anlaşılmaktadır.



Şekil 3.30. Ortalama tabanlı aykırılık tespiti sonrası cisim tespiti başarı oranları

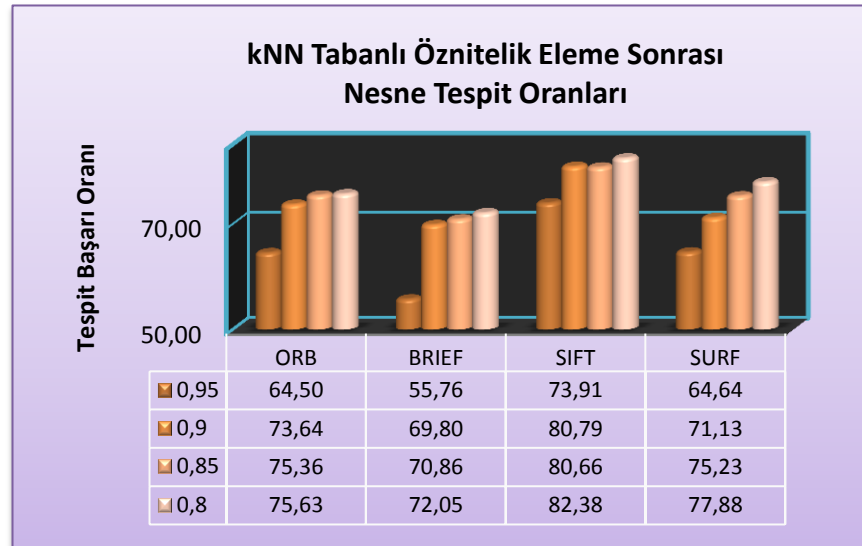


Şekil 3.31. Ortalama tabanlı aykırılık tespiti sonrası sınırlayıcı kutu boyutları

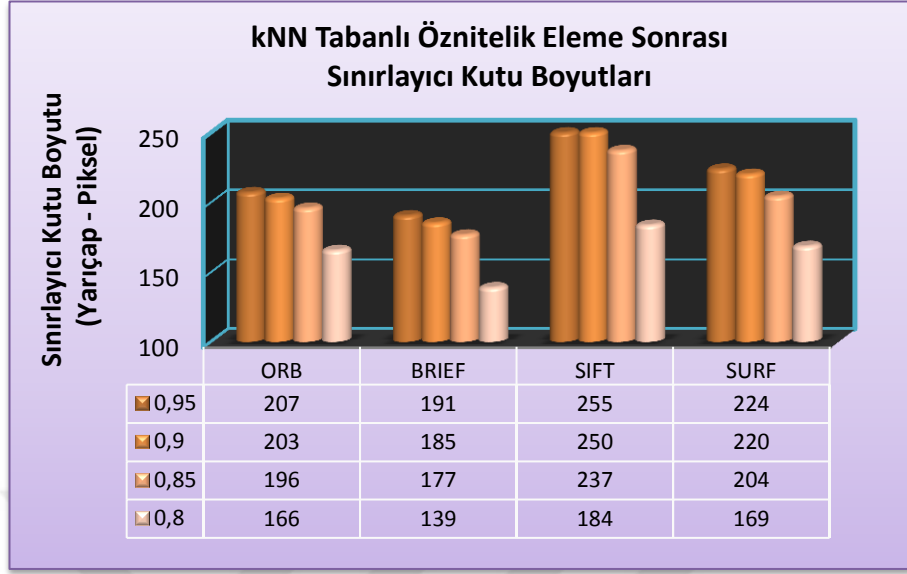


**Şekil 3.32.** Ortalama tabanlı aykırılık tespiti sonrası eşleşen öznitelik sayıları

K en yakın komşu eşleştirme işleminde elde sonuçlar incelendiğinde, Şekil 3.34'teki sınırlayıcı kutu boyutlarının, olması gereken cisim boyutlarından daha küçük olduğu ve bu nedenle düşük tespit başarısı elde edildiği anlaşılmaktadır.



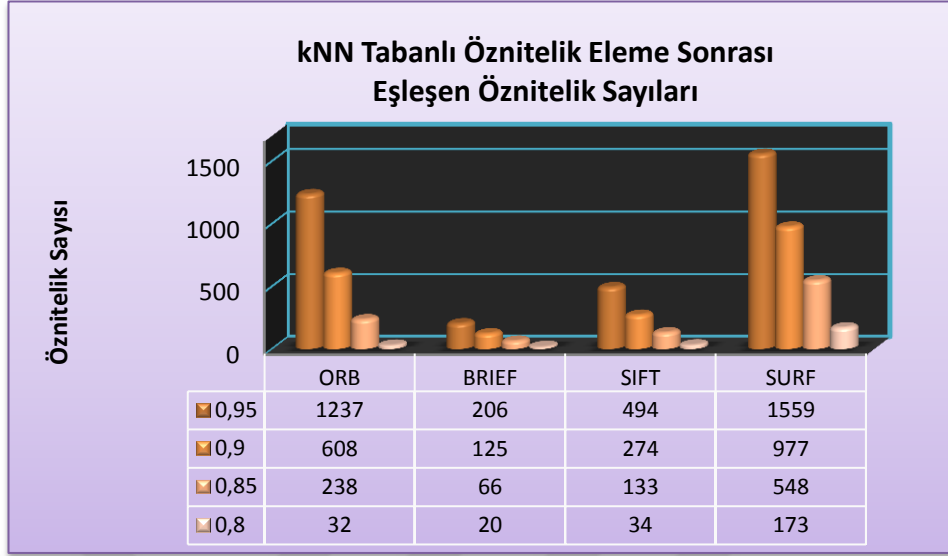
**Şekil 3.33.** k en yakın komşu tabanlı öznitelik eleme sonrası cisim tespiti başarı oranları



**Şekil 3.34.** k en yakın komşu tabanlı öznitelik eleme sonrası sınırlayıcı kutu boyutları

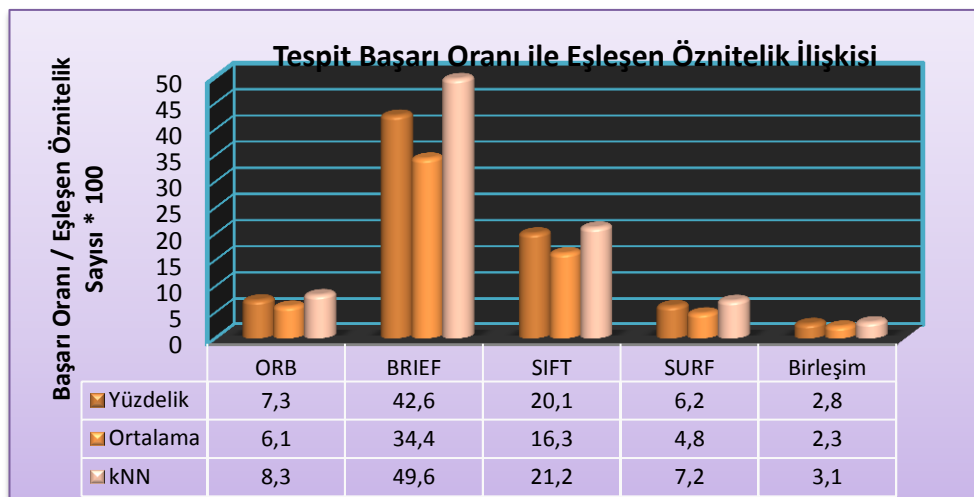
Şekil 3.35'te gösterildiği üzere, K en yakın komşu öznitelik eleme işleminde 1. sütundaki öznitelik yakınlık mesafe parametresinin her 0,05'lik azalmaya karşılık öznitelikler oldukça yüksek oranda elenmişlerdir. Bu işlemde özniteliklerin %40 ile %50'si elenmesine karşılık, Şekil 3.34'teki sınırlayıcı kutularda %7 ile %20 oranlarında küçülme gerçekleşmiş, bunun sonucunda ise Şekil 3.33'teki cisim tespit başarısı %5 ile %17 arasında artmıştır.





**Şekil 3.35.** k en yakın komşu tabanlı öznitelik eleme sonrası eşleşen öznitelik sayısı

Eşleşen öznitelikler sayıları ile cisim tespit başarı oranı karşılaştırıldığında, en iyi sonuç Şekil 3.36'da görüldüğü üzere BRIEF öznitelik tanımlayıcısında elde edilmiştir. Ayrıca k en yakın komşu eşleştirme ve eleme işleminin, kaba kuvvet eşleştiricisi yüzdelik ve ortalama aykırılık tespitine göre, eşleşen öznitelik sayıları ile cisim tespit başarılarına oranına göre daha iyi olduğu Şekil 3.36'dan anlaşılabilir.



**Şekil 3.36.** Cisim tespit başarı oranları ile eşleşen öznitelik sayıları ilişkisi (tespit başarı oranı / eşleşen öznitelik sayısı \* 100)

### 3.2.4. YOLO ile insan tespiti

Foto-kapanlardan elde edilen görüntüler ışık şiddeti yetersizliği, kapanma, ölçekleme değişimi, 3d v afin dönüşümü, parçalı cisim bulunması gibi zorlukları çoğunlukla bulunmaktadır. Şekil 3.37’de gösterildiği üzere bu görüntüler farklı zamanlarda, ortamlarda ve durumlarda kaydedilmektedir.



**Şekil 3.37.** Işık şiddeti yetersizliği, kapanma, ölçekleme değişimi, 3d ve afin dönüşümü, parçalı cisim bulunması zorlukları barındıran foto-kapan görüntüleri

Derin öğrenme tabanlı yapay sinir ağı yöntemleri ile cisimlere ait derin öznitelikleri çıkararak görüntülerin içerdiği birçok zorluğa ve probleme rağmen etkili bir tespit ve sınıflandırma işlemi gerçekleştirilebilmektedir. Derin öğrenme tabanlı evrişimli sinir ağlarında cisim tespiti amacıyla kullanılan OverFeat, Faster R-CNN, ve YOLO mimarileri bulunmaktadır. YOLO (You Only Look Once: Unified, Real-Time Object Detection), GoogleNet mimarisinden esinlenilmiş, görüntüdeki cisimlere ait uzamsal olarak ayrılmış sınırlayıcı kutularının sınıf olasılıklarının regresyon problemi olarak alındığı, tek bir sinir ağında bu sınıf olasılıklarının tahmin edildiği bir evrişimli sinir ağları mimarisidir (Tompson *et al.* 2015). Evrişimsel sinir ağları kategorisinden olan bu mimaride standart olarak saniyede 45 görüntü, hızlı YOLO mimarisinde ise saniyede 155 görüntü gerçek zamanlı işlenebilmektedir. Literatürdeki en gelişmiş yöntemlere göre daha az yerleştirme hatası yapmaktadır (Tompson *et al.* 2015). Bu çalışmada,

Şekil 3.38’de örnek çıktısı gösterilen, konumlandırma işleminin etkili ve hızlı bir şekilde yapıldığı evrişimli sinir ağı mimarisi olan YOLO mimarisi ve PASCAL VOC 2012 veri seti ile 80 sınıf için eğitilmiş ağırlıklar kullanılarak, doğal foto kapan görüntülerinde insan tespiti gerçekleştirilmiştir. Veri setindeki zorluklara karşı YOLO mimarisinin performansı incelenmiş ve yorumlanmıştır.

YOLO mimarisinde, 24 adet evrişim katmanı, bu katmanların devamında 2 adet tam bağlı katman bulunmaktadır. Alternatif 1x1 evrişimli katmanlar, önceki katmanlardan gelen özniteliklerin boyutunu azalmaktadır (Tompson *et al.* 2015).



**Şekil 3.38.** YOLO mimarisi ile cisim tespiti

Mimarinin son katmanında üzere hem cisimlere ait sınıf olasılıkları, hem de bu sınıflara ait sınırlayıcı kutu koordinatları öngörülmektedir. Son katmanda doğrusal bir aktivasyon fonksiyonu kullanılırken, diğer tüm katmanlarda Denklem 19’da gösterilen sızdıran düzeltilmiş (leaky rectified) doğrusal aktivasyon kullanılmaktadır (Tompson *et al.* 2015):

$$\varnothing(x) = \begin{cases} x, & \text{if } x > 0 \\ 0.1x, & \text{otherwise} \end{cases} \quad (19)$$

Cisim içermeyen tahminlerin elenebilmesi için güven parametresi ve üst üste gelen sınırlayıcı kutuların kabul edilmesi için eşik parametresi olmak üzere 2 parametre kullanılmaktadır. Sınırlayıcı kutular, en yüksek IoU (Intersection over Union) değerine dayanan bir tahmin işlemi kullanılarak etiketlenmektedir. Cisim tespiti için sinir ağı, PASCAL VOC 2007 ve 2012'den eğitim ve doğrulama veri setleri hakkında yaklaşık 135 devir boyunca eğitilmektedir (Tompson *et al.* 2015). Çalışmamızda 80 adet cisim için PASCAL VOC 2012 ile eğitilmiş ağırlık değerleri kullanılarak, foto-kapan görüntülerindeki insan tespit etme ve konumlandırma başarısı incelenmiştir.

Foto-kapanlar kullanılarak doğal ortamdan elde edilmiş Şekil 3.37 ve Şekil 3.39'da örnekleri gösterilen veri seti (Kays *et al.* 2010), içerisinde 100 farklı arka plana sahip, gece ve gündüz kaydedilmiş, 2048x1536 boyutlarında toplam 2880 görüntü içermektedir. Veri setindeki görüntülerin 2150'sinde, bu görüntülerde bulunan cisimlere ait tür, konum ve boyut bilgileri kesin referans (ground-truth) olarak bulunmaktadır.



**Şekil 3.39.** Foto-kapan veri seti

Veri setinde kesin referans içerisinde etiketlenen görüntülerin 314'ünde sınırlayıcı kutuları belirlenmiş tam veya parçalı insan görüntüsü bulunmakta olup, 1836'ında insan

bulunmamaktadır. Veri setinde etiketlenmemiş 2 bölümünde ise 730 görüntü bulunmaktadır ve bu görüntüler tarafımızca yalnızca insan olup olmadığı tespit edilerek etiketlenmiştir.

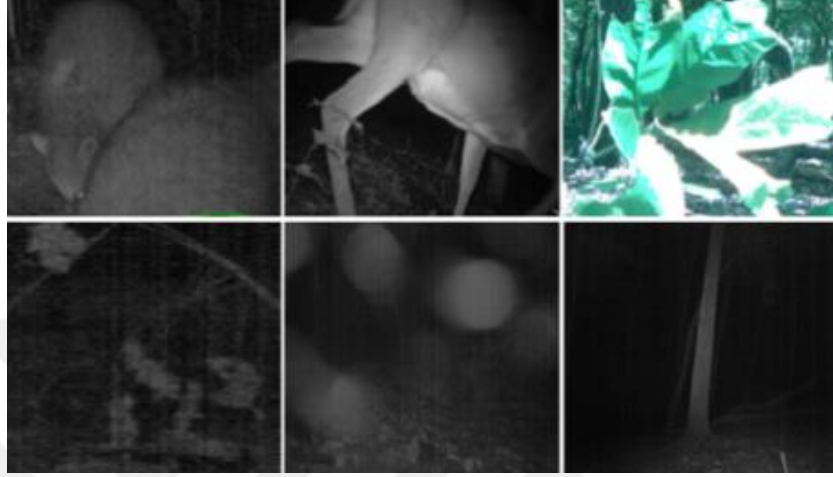
Çalışmada YOLO yöntemine ait güven parametresi olarak 0,5 değeri, eşikleme parametresi olarak 0,3 değeri kullanılmıştır. Veri seti üzerinde yapılan tespit çalışmasında Çizelge 3.20’de görüldüğü üzere YOLO kullanılarak 314 insan içeren görüntünün 297’si tespit edilmiştir. Tespit edilemeyen 17 insan görüntüsü Şekil 3.40’ta görüldüğü üzere parçalı cisim görüntüleri içermektedir ve cisimler ağaçlar bitkiler veya diğer cisimler ile perdelenmiştir. Şekil 3.41’de ise insan tespit edilmiş bitki, hayvan veya arka plana ait görüntüler gösterilmektedir.



**Şekil 3.40.** Tespit edilemeyen 17 insan görüntüsü

Çizelge 3.20 incelendiğinde, IoU yöntemine 0,5 parametresi ile %94,6 gibi oldukça yüksek bir oranda tespit başarısı elde edildiği anlaşılmakta, IoU parametresi arttırıldıkça

tespit başarısının hızla düşmekte olduğu gözlemlenmektedir. Çizelge 3.21'deki hata matrisinden anlaşıldığı üzere hatalı pozitif değeri oldukça düşüktür.



**Şekil 3.41.** İnsan olarak tespit edilmiş hatalı görüntüler

**Çizelge 3.20.** IoU değerlerine göre tespit sonuçları

IoU	Tespit Edilen İnsan Sayısı	Başarı Oranı %
0,5	297	94,6
0,6	291	92,7
0,7	275	87,6

**Çizelge 3.21.** Veri setinin birinci bölümüne ait hata matrisi

		Gerçek Değer		
		Cisim Var	Cisim Yok	Toplam
Tahmin Edilen Değer	Cisim Var	(TP) 297	(FP) 6	303
	Cisim Yok	(FN) 17	(TN) 1830	1847
	Toplam	314	1836	

Çalışmada sınırlayıcı kutu bilgileri bulunan 2150 görüntüde elde edilmiş duyarlılık (Recall) değeri 0,95 iken, kesinlik (Precision) değeri 0,98'dir. Tarafımızca insan var veya yok şeklinde etiketlenen 730 görüntüde ise, Çizelge 3.22'de görüldüğü üzere 219 insan içeren görüntünün 211'i doğru tespit edilmiş olup, 8 tanesi tespit edilememiştir. Tespit edilemeyen insan görüntülerinin gösterildiği Şekil 3.42 incelendiğinde, cisimlerin ağaç ve yapraklar ile büyük oranda kapandığından görülmediği ve cisimlerin parçalı olarak bulunduğu anlaşılmaktadır.



Şekil 3.42. Tespit edilememiş 8 insan içeren görüntü

Çizelge 3.22. Veri setinin ikinci bölümüne ait hata matrisi

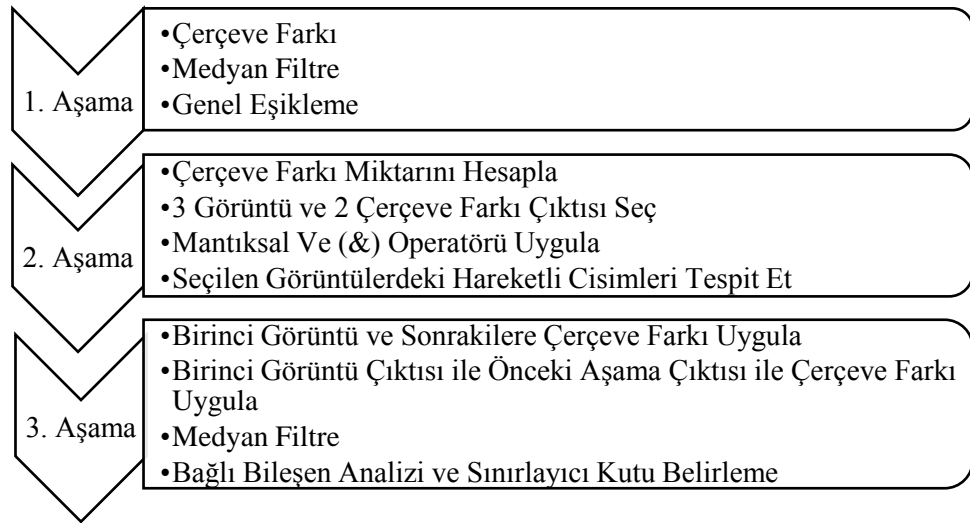
		Gerçek Değer		
		Cisim Var	Cisim Yok	Toplam
Tahmin Edilen Değer	Cisim Var	(TP) 211	(FP) 0	211
	Cisim Yok	(FN) 8	(TN) 511	519
	Toplam	219	511	

730 görüntünün bulunduğu veri setinin 2. bölümünde ise duyarlılık değeri  $211/219 = 0,96$  ve kesinlik değeri  $211/211=1,00$  olarak elde edilmiştir.

### 3.2.5. Akıllı üç çerçeve farkı

Bu çalışmada, üç kare farkı yöntemine dayanan yeni bir hareketli cisim algılama yöntemi önerilmiştir. Çalışmanın katkısı, üç çerçeve farkı yönteminde hareketli cismin bütüne ait sonuç elde edilmesi ve bu üç kare fark yönteminde kullanılacak üç görüntünün seçilmesiyle sağlanmaktadır. Bu yöntemde, iki resim arasındaki yerel farkı en fazla yapan resimler seçilmektedir. Bu katkıyla, ayrık ve örtüşmeyen hareketli cisim bölgelerinin çerçeve farkı çıktısında elde edilmesini sağlamaktadır.

Yöntemimiz, Şekil 3.43'te gösterildiği üzere üç aşamadan oluşmaktadır. İlk aşamada, Şekil 3.44'te ifade edildiği gibi ardışık görüntülere çerçeve farkı, medyan filtresi ve genel eşikleme yöntemleri uygulanmaktadır. Ardışık görüntüler arasındaki farkın yerel maksimum olduğu, çerçeve farkı yönteminde kullanılacak hareketli cisimlerin ayrık ve örtüşmeyen konumlarını içeren görüntüler seçilmektedir. Bu seçim işleminde, ikinci görüntüdeki hareketli cisimler mantıksal ve operatörüyle algılanabilmektedir. Üçüncü aşamada, hareketli cisim bilgisi bulunan görüntü, çerçeve farkı yöntemi uygulanarak bütün görüntülerdeki hareketli cisimler algılanabilmektedir. Hareketli cisimler tespit edildikten sonra, cisimlere ait sınırlayıcı kutuların belirlenmesi amacıyla elde edilen çıkışlara bağlı bileşen analizi uygulanmaktadır.



**Şekil 3.43.** Hareketli cisim tespiti için önerilen yöntem



Önerilen yöntemde hesaplanan iki görüntü arasındaki fark Denklem 20 ile hesaplanmaktadır.  $I_d$  değeri,  $I_k$  görüntünün piksel değerlerini  $I_{k+1}$  görüntüsünden farkının mutlak değeri olmaktadır.

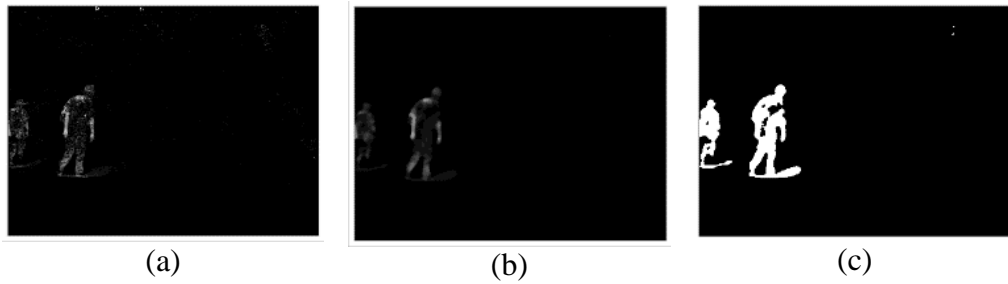
$$I_{d(k,k+1)} = |I_{k+1} - I_k| \quad (20)$$

Hesaplanan  $I_d$  çerçeve farkı çıktısındaki gürültünün giderilmesi amacıyla, Denklem 21'de gösterilen medyan filtre uygulanmaktadır.

$$I_d = \text{median}(I_d) \quad (21)$$

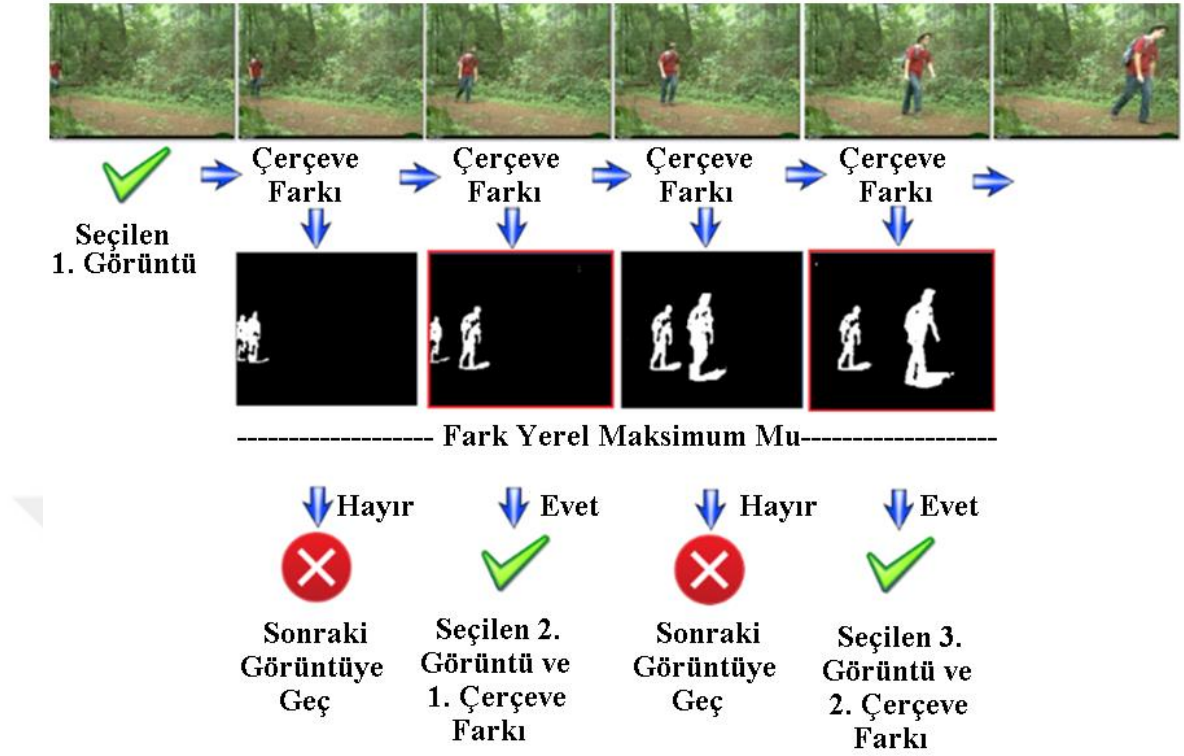
Medyan filtrenin uygulanmasından sonra, çerçeve fark çıktısı, Denklem 22'deki genel eşikleme metodu ile ikili görüntüye dönüştürülmektedir.  $T$ , Denklem 22'deki genel eşik değerini ifade etmektedir.

$$I_f(i,j) = \begin{cases} 0, & \text{and } I_d(i,j) < t \\ 1, & \text{and } I_d(i,j) \geq t \end{cases} \quad (22)$$



**Şekil 3.44.** İlk aşama a) çerçeve farkı b) medyan filtre c) genel eşikleme

Önerilen yöntemin ikinci aşamasında, Şekil 3.45'te gösterildiği gibi, birinci görüntü ile farkı maksimum olan ikinci görüntü, ikinci görüntü ile arasındaki fark maksimum olan üçüncü görüntü seçilmektedir.

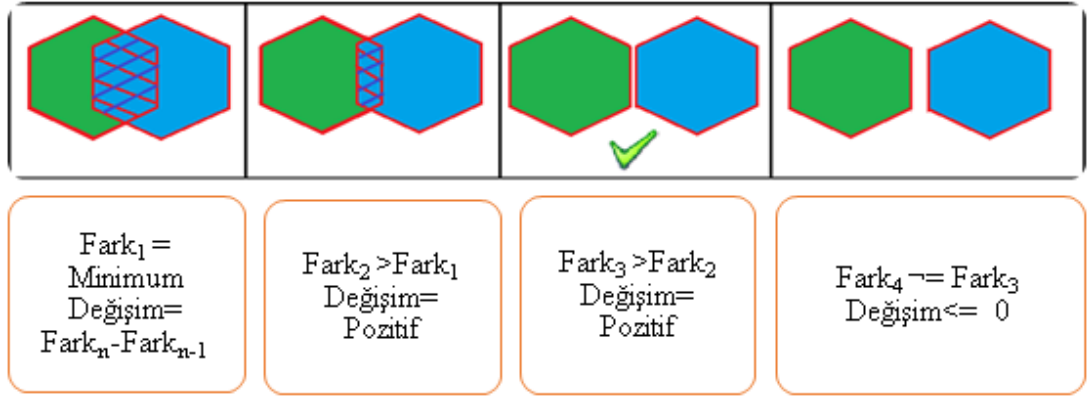


Şekil 3.45. İkinci aşamada üst üste gelmeyen çıktığı veren görüntülerin seçilmesi

Çerçeve farkı çıktılarının fark miktarı Denklem 23 ile hesaplanmaktadır.  $M$  ve  $n$ , Denklem 23'deki görüntünün boyutlarını ifade etmektedir.

$$Diff_1 = \sum_{i=0}^m \sum_{j=0}^n I_{f1}(i,j) \quad (23)$$

Önerilen yöntemde kullanılacak 3 görüntünün seçiminde, çerçeve farkının değişiminin, ilk görüntü ile ardışık her görüntünün fark değerlerinin sıfır veya negatif olması beklenmektedir. Görüntüdeki konumları ayrık olan ve üst üste gelmeyen cisimler içeren görüntülerin seçimi, Şekil 3.46, Şekil 3.47 ve Şekil 3.48'de gösterilmektedir. Şekil 3.46'da görülebileceği gibi, cisimlerin kesişmesi azaldığında fark değerinin miktarı artmaktadır. Cisimler çerçeve farkı çıktısında ayrık olduğunda başka bir ifadeyle üst üste gelmediğinde, fark değeri maksimum olmakta ve önceki çerçeve farkı çıktısındaki değişim 0 veya negatif değer olmaktadır.



**Şekil 3.46.** Cisim konumları ayrıık olan çerçeve farkı çıktılarının seçilmesi

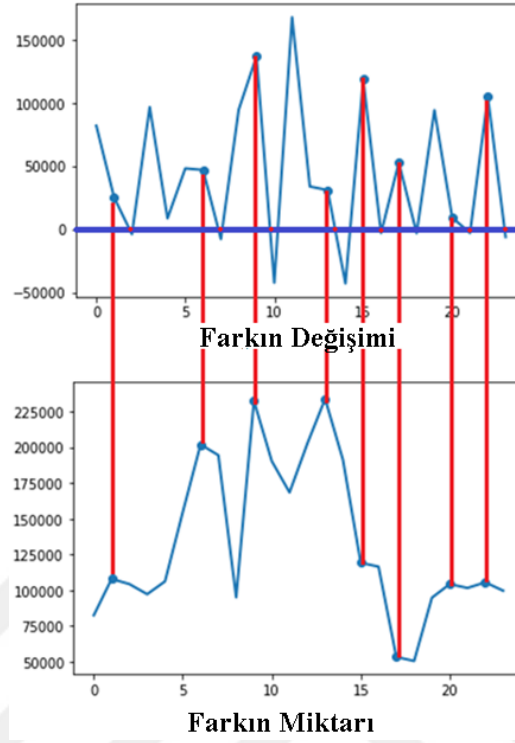
Şekil 3.47’de gösterilen alt görselde, Şekil 3.45’te örneği gösterilen çerçeve farkı çıktılarının beyaz piksel sayısını ifade etmektedir. Bu fark miktarı, hareketli cismin görüntüdeki konumu ayrıklaştıkça artmaktadır. Şekil 3.47’de üst görselde ise ardışık çerçeve farkı çıktılarında, hesaplanan fark miktarının değişimi gösterilmektedir. Bu işleme ait algoritma Çizelge 3.23’te gösterilmektedir.

**Çizelge 3.23.** Ayrıık konumlardaki cisim görüntülerinin seçilmesi

```

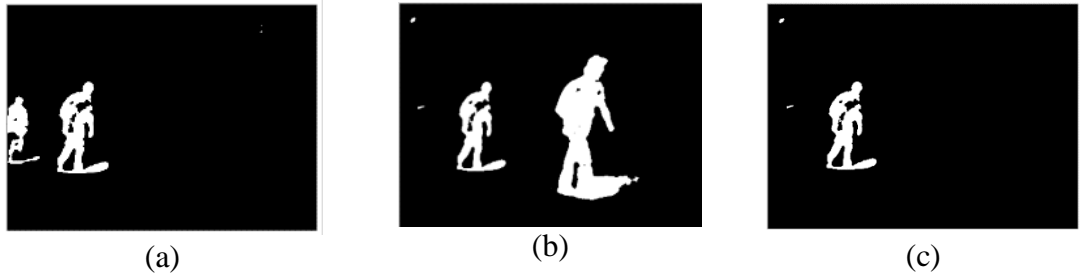
Seçilen Görüntüler [1] = 1. Görüntü;
Seçilen Görüntü Sayısı = 1;
For ( I = 2’den N’e Kadar Bütün Görüntüler İçin):
    Çıktı = Çerçeve Farkı Uygula (Seçilen Son Görüntü, I. Görüntü)
    Çıktı = Medyan Filtre Uygula (Çıktı)
    Çıktı = Eşikleme Uygula (Çıktı)
    Fark = Fark Miktarını Hesapla (Çıktı)
    Değişim = Fark – Önceki Fark;
    If ( Eğer Değişim <= 0 ise)
        Seçilen Görüntü Sayısı = Seçilen Görüntü Sayısı + 1;
        Seçilen Görüntüler [Seçilen Görüntü Sayısı] = (I-1). Görüntü;
    End If
End For

```

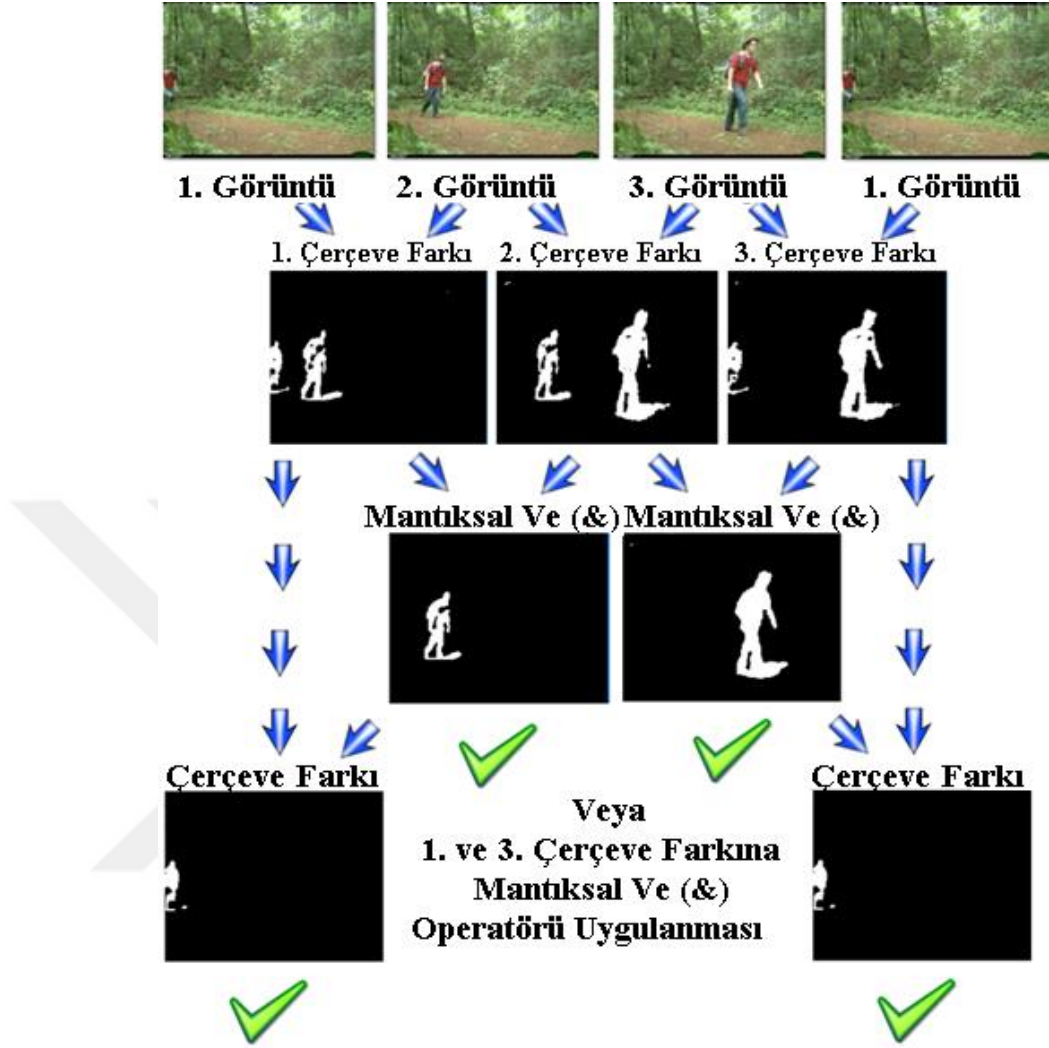


**Şekil 3.47.** Üst) ardışık görüntülerdeki çerçeve farklarının değişimi alt) denklem 4 ile hesaplanan fark miktarı

Şekil 3.48, ikinci görüntüdeki hareketli cisimlerin mantıksal ve operatörü uygulanarak nasıl algılandığını göstermektedir. Şekil 3.49, seçilen görüntülerde hareketli cisimleri tespit etmek için uygulanabilecek diğer olası yöntemleri göstermektedir.



**Şekil 3.48.** İkinci aşama a) seçilen ilk çerçeve farkı b) seçilen ikinci çerçeve farkı c) bu iki görüntüye mantıksal ve (&) uygulanması



Şekil 3.49. Seçilen 1., 2. ve 3. görüntülerdeki hareketli cisimlerin tespit edilmesi

İkinci görüntüdeki hareketli cismi gösteren görüntü  $I_b$ , seçilen görüntülerin çerçeve farkı çıktıları olan  $I_{f1}$  ve  $I_{f2}$  görüntülerine Denklem 24 uygulanarak hesaplanmaktadır. Birinci ve üçüncü görüntülerdeki hareketli cisimleri gösteren  $I_a$  ve  $I_c$ , Denklem 25 ve Denklem 26 kullanılarak hesaplanmaktadır.

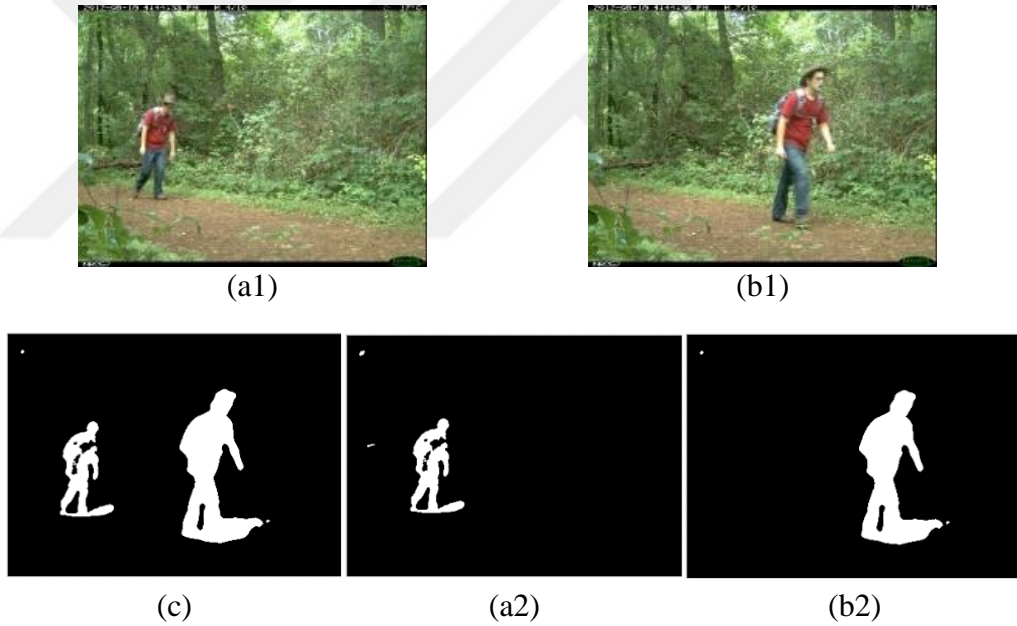
$$I_b = I_{f1} \otimes I_{f2} \quad (24)$$

$$I_a = I_{f1} - I_b \quad (25)$$

$$I_c = I_{f2} - I_b \quad (26)$$

Üçüncü aşamada, diğer görüntülerdeki hareketli cisimler, hareketli cisim bilgisine sahip görüntü ve Denklem 20, Denklem 21, Denklem 22 ve Denklem 25 ile bir sonraki görüntüler kullanılarak, Şekil 3.50'de gösterildiği gibi tespit edilebilir.

Önerilen yöntemin en önemli katkısı, üç çerçeve farkına dayalı yöntemde belirtildiği gibi hareketli cismin kenar bilgileri değil (Gang *et al.* 2013; Wojcik and Kaminski 2004; Zhang *et al.* 2012), görüntülerdeki hareketli cismin tamamına ait bilgiler algılanabilmektedir.



**Şekil 3.50.** Diğer görüntülerdeki hareketli cisim tespiti a1) seçilen ilk görüntü b1) seçilen ikinci görüntü c) a1 ve a2'deki görüntülerin çerçeve farkı çıktısı a2) a1'deki hareketli cisim b2) b1'deki hareketli cisim

$O_a$ ,  $O_b$ ,  $O_c$  Denklem 27'de gösterildiği üzere  $I_a$ ,  $I_b$ ,  $I_c$  görüntülerindeki cisimleri temsil etmekte olup, bu cisimlerin görüntülerdeki konumlarının üst üste gelmediği varsayılırsa, bu cisimlerin kesişimleri boş küme olmaktadır.

$$O_a \cap O_b = \emptyset, O_b \cap O_c = \emptyset, O_a \cap O_c = \emptyset \quad (27)$$

Bu üç görüntünün çerçeve farkı çıktısını ifade eden  $I_{f1}$ ,  $I_{f2}$  Denklem 28'de gösterildiği üzere  $O_a$ ,  $O_b$ ,  $O_c$  ifadelerinin birleşimi ile ifade edilebilmektedir.

$$I_{f1} = O_a \cup O_b, I_{f2} = O_b \cup O_c \quad (28)$$

Üç çerçeve farkında çerçeve farkı çıktıları  $I_{f1}$ ,  $I_{f2}$ 'ye mantıksal ve (&) operatörü Denklem 29'daki şekilde uygulanmaktadır.

$$I_{f1} \cap I_{f2} = (O_a \cup O_b) \cap (O_b \cup O_c) \quad (29)$$

İfadedeki eşitliğin sağ tarafını açarsak Denklem 30 elde edilmektedir.

$$= (O_a \cap O_b) \cup (O_a \cap O_c) \cup (O_b \cap O_b) \cup (O_b \cap O_c) = \emptyset \cup \emptyset \cup O_b \cup \emptyset \quad (30)$$

Denklem 27'deki varsayımlar dikkat alındığında, Denklem 30'daki ifadeden  $I_b$  görüntüsündeki hareketli cisim olan  $O_b$ 'nin elde edilebildiği anlaşılmaktadır.

Çalışmamızda hareketli cisimleri içeren  $I_a$ ,  $I_b$ ,  $I_c$  görüntülerindeki cisimlerin sınırlayıcı kutularını belirlemek için bağlı bileşen analizi kullanılmıştır. Bu işleme ait sözde kod aşağıda verilmiştir. Cisim algılamayı hızlandırmak ve gürültü olarak kabul edilebilecek küçük cisimleri göz ardı etmek için, cisim boyutuna göre 1/35 eşik parametresi ile eleme işlemi gerçekleştirilmiştir.

Önerilen yöntemde gürültülerin temizlenmesi ve cisimlerin bütünleştirilmesi amacıyla medyan filtre ve küçük cisimlerin elenmesi amacıyla eşik değer olmak üzere iki parametre kullanılmaktadır. Medyan filtre parametresi, küçük boyuttaki arka plan hareketlerinin çerçeve farkı çıktısında elenmesini sağlamakta olup, arka plan hareketinin

artması durumunda bu parametrenin artması gerekebilmektedir. Yapılan çalışmalarda parametre optimizasyonu gereken durumlarla karşılaşılmaktadır.

Önerilen yöntemin birçok veri seti ile uyumlu çalışabilmesi amacıyla 3.3.2. başlığında önerilen hareketli cisim tespitin iyileştirilmesi yöntemi kullanılabilir. Önerilen yöntemde oluşturulmuş hatalı ve hatasız çıktı modeli, bu çalışmada iyileştirme amacıyla kullanılmaktadır. Şekil 3.51’de örnek görüntüler ve eşik parametrelerine göre elde edilen tespit sonuçları gösterilmektedir. Verisetindeki (Anonymous, 2019b) arka plan hareketini içeren örnek görüntülere ait elde edilen çıktılarda Şekil 3.51-a1 ve Şekil 3.51-b1’de medyan filtre değeri 29 ile elde edilmiş hareketli cisim tespit sonuçları, Şekil 3.51-a2 ve Şekil 3.51-b2’de ise medyan filtre değeri 51 ile elde edilmiş sonuçları göstermektedir. Şekillerden görüldüğü üzere arka plan hareketinin miktarına göre, ön plandaki cisimlerin tespit edilmesi amacıyla daha yüksek parametre değerleri kullanılabilir. Bu problemin çözülmesi ve adaptif bir çözüm olabilmesi amacıyla Çizelge 3.24’te önerilen algoritma ile bu medyan filtre değeri otomatik olarak optimize edilerek görüntüdeki sahneye göre en uygun parametre değeri belirlenmekte ve Şekil 3.51-a3’te gösterilen sonuç elde edilebilmektedir. Şekil 3.51-a1 ve Şekil 3.51-b3’te ise medyan filtre parametre değişimi ifade edilmektedir. Aşağıdaki sözde koddan anlaşıldığı üzere, arka plan hareketi sahneye göre değişebilmektedir. Videonun bir kısmında arka plan hareketi artarken, sonraki görüntülerde bu hareket azalabilmekte ve hatta bitmektedir. Bu nedenle arka plan hareketi tespit edildiği durumlarda medyan filtre parametresi +2 olarak arttırılmakta iken, eğer 10 döngüde hatasız olarak sınıflandırılma işlemi gerçekleşirse bu parametrenin normalleştirilebilmesi amacıyla -2 olarak azaltılmakta, bu sayede yalnızca videodaki arka plan hareketine göre değil, video içindeki sahnelerle de adaptiflik sağlanmaktadır. Şekil 3.51-b3’te gösterildiği üzere bu parametre değeri duruma göre artıp azalabilmektedir. Optimum parametre 51 iken, arka plan hareketinin maksimuma ulaştığı durumlarda bu parametre 75’ çıkmakta, arka plan hareketi azaldıkça 35’e inerek en iyi hareketli cisim tespitini sağladığı görülmektedir. Uygulamaya ait algoritma Çizelge 3.24’te gösterilmektedir.



**Çizelge 3.24.** Hareketli cisim tespitinin iyileştirilmesi için kullanılan algoritma

```

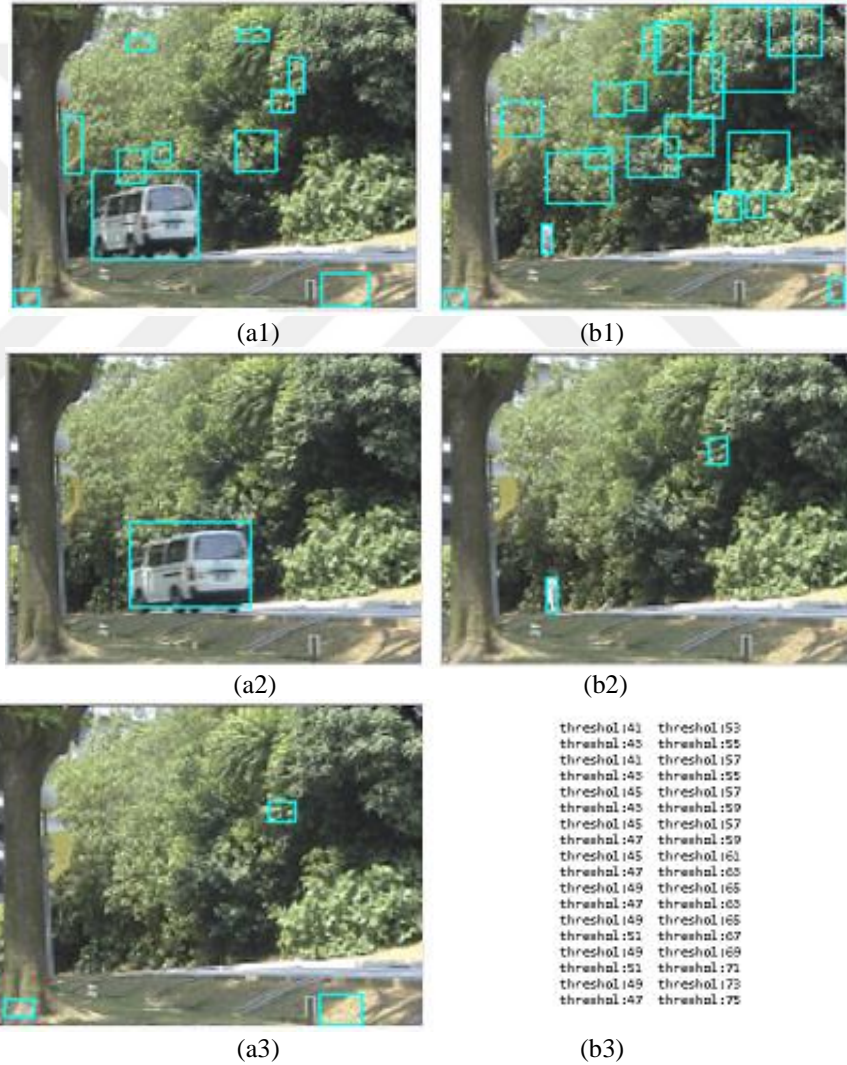
Medyan Filtre Parametresi = 1
For (Her Görüntü için Çerçeve Farkı Çıktısını Elde Et):
  Sınıf=Çerçeve Farkı Çıktısını Sınıflandır()
  If (Eğer Sınıf Hatalı Olarak Etiketlenmiş ise)
    Medyan Filtre Parametresi = Medyan Filtre Parametresi +2
    Cisim Tespitinde Sonraki Aşamayı Uygula()
  Else If (Eğer 10 Döngüdür Sınıf Hatasız Olarak Etiketlenmiş ise)
    Medyan Filtre Parametresi = Medyan Filtre Parametresi -2
    Cisim Tespitinde Sonraki Aşamayı Uygula()
  End If
End For

```

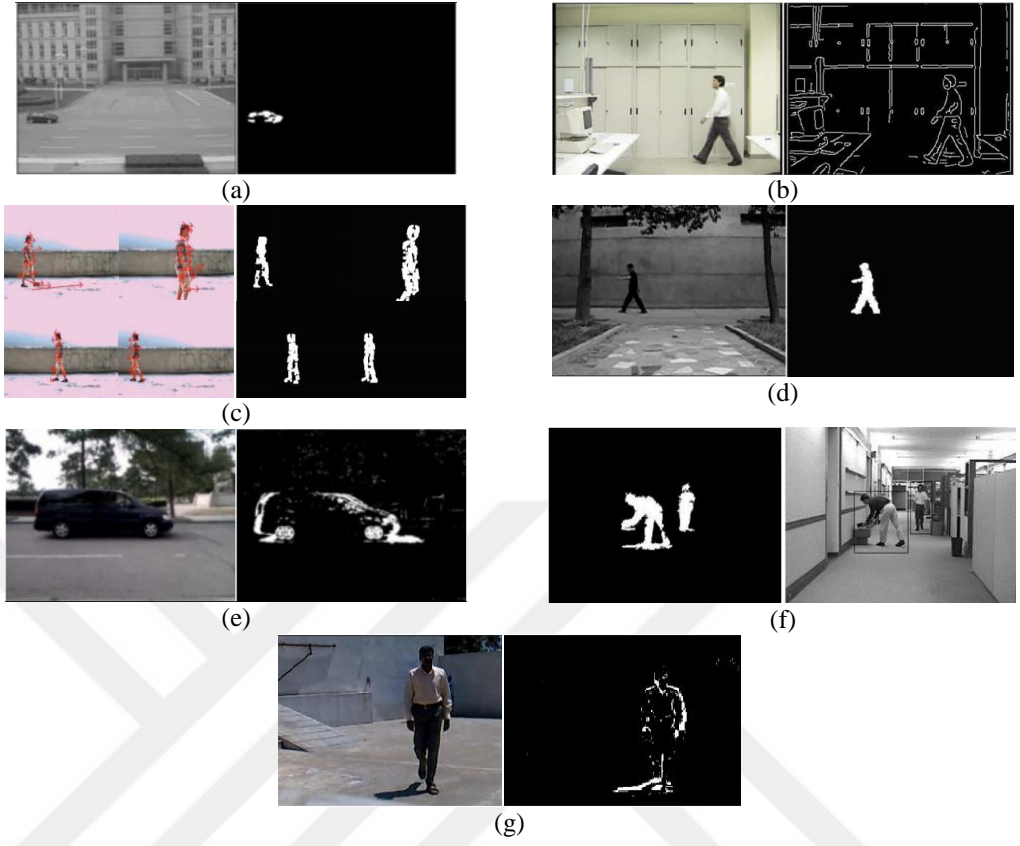
Hareketli cisim tespiti için önerdiğimiz yöntem, genel eşik, medyan filtre ve küçük cisimlerin elenmesi için bir sınırlayıcı kutu boyutu için eşik parametresi olmak üzere 3 parametre gerektirmektedir. Genel eşik parametresi 17 olarak, veri setinde medyan filtre kutusu büyüklüğü olarak görüntü boyutunun 1/50 ve küçük cisimlerin göz ardı edilmesi için 1/35 değerleri kullanılmıştır. Medyan filtre kutusu büyüklüğünün 1/50 oranlı birçok veri setinde başarılı sonuçlar verdiği görülmüştür ve bütün çalışmalarda sabit olarak kullanılmıştır.

Piccardi'nin arka plan çıkarma yöntemlerinin performans analizini içeren çalışmasında (Piccardi, 2004) değişen gauss ortalama yönteminin zaman karmaşıklığı  $O(1)$  olarak ifade edilmiştir. En az karmaşıklığa ve hafıza gereksinimine sahip olan bu yöntem, bir eşikleme ve arka plan modeli güncelleme işlemleriyle ve bir veya iki parametre kullanılarak uygulanmaktadır. Çalışmamızda, bu yönteme benzer şekilde, bir cisim bilgisine bulunan bir görüntü ve cismin görüntüde bulunduğu konumu ifade eden ikili görüntü gerektirmektedir. Önerilen yöntemin iki parametreye ihtiyacı bulunmaktadır ve hareketli cisimleri algılamak için eşikleme işlemi ve fark operatörü kullanılmaktadır. Önerilen diğer yöntemlerden farklı olarak, her görüntüde bir arka plan modelini güncelleme ihtiyacı bulunmamaktadır. Şekil 3.52, benzer çalışmaların sonuçlarını

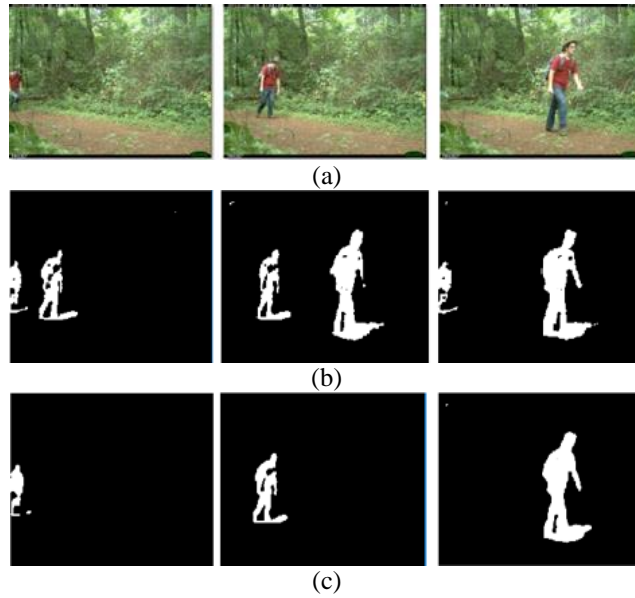
göstermektedir. Bu çalışmalarda çerçeve fark çıktılarının hareketli cisimlerin tespitinde etkin bir şekilde kullanılması amaçlanmaktadır. Şekil 3.53 önerilen yöntemde hareketli cisimlerin tespitindeki aşamaları göstermektedir. Şekil 3.54 ise önerilen yöntem ile farklı veri setlerindeki hareketli cisimlerin sınırlayıcı kutu sonuçlarını göstermektedir. Videolar üzerinde yapılan çalışmada kullanılan veri seti, Edinburgh Üniversitesi'ndeki CAVIAR projesinde kaydedilen bağlantıdan (Anonymous, 2018b) alınmıştır ve sonuçlar Çizelge 3.25, Çizelge 3.26 ve Çizelge 3.27'de gösterilmektedir.



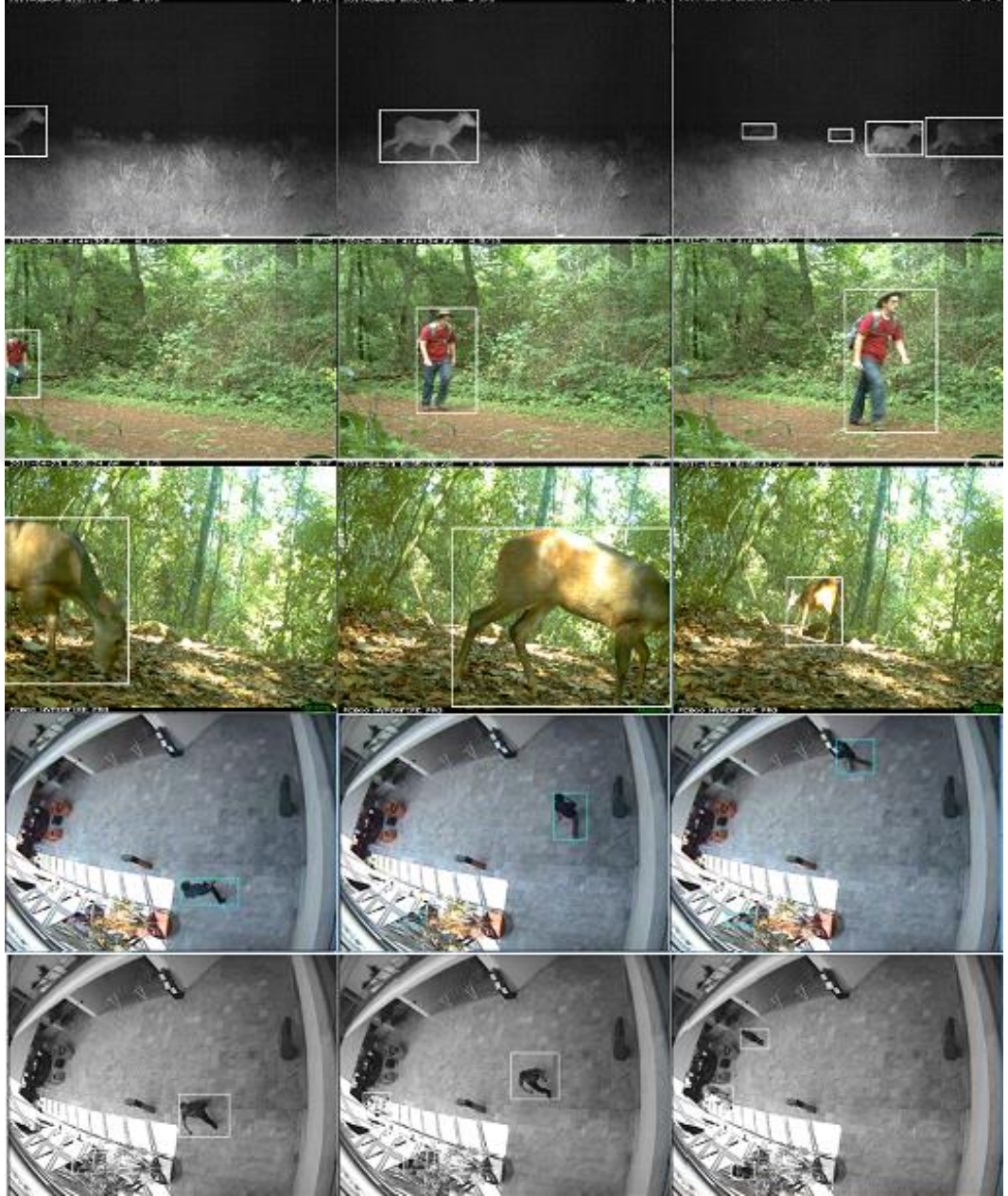
**Şekil 3.51.** Önerilen yöntemde hareketli cisim tespitinin iyileştirilmesi a1-b1) düşük parametrelili sonuçlar a2-b2) yüksek parametrelili sonuç a3) otomatik parametre ayarlanmış görüntü b3) parametre değeri değişimi



**Şekil 3.52.** Benzer çalışmalara ait sonuçlar a) (Zhang *et al.* 2012) b) (Gang *et al.* 2013) c) (Han *et al.* 2015) d) (Huang *et al.* 2007) e) (Zhang and Zhang 2013) f) (Thapa *et al.* 2014) g) (Alex *et al.* 2014)



**Şekil 3.53.** Önerilen sistemdeki çıktılar a) seçilmiş görüntüler b) çerçeve farkı çıktıları (1-2,2-3,1-3) c) seçilen görüntülerde tespit edilen hareketli cisimler



**Şekil 3.54.** Önerilen yöntem ile farklı görüntülerde elde edilmiş sonuçlar

Walk1.mpg videosunda 609 görüntü bulunmaktadır. Önerilen yöntem kullanılarak videonun gerçek kesin değerleri içerisinde bulunmayan 595 cisim tespit edilmiştir. Hata matrisi Çizelge 3.25'te gösterildiği üzere, eri çağırma değeri 0,97, kesinlik değeri 0,98 olarak elde edilmiştir.

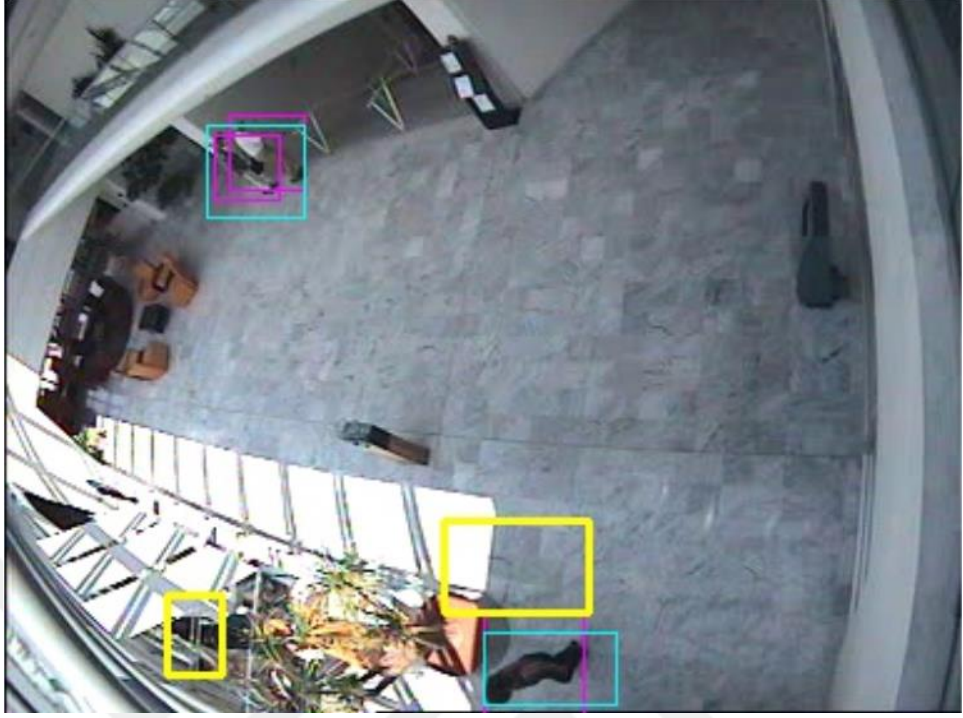
**Çizelge 3.25.** Walk1.mpg hata matrisi

		<b>Gerçek Değerler</b>		
		Pozitif	Negatif	Toplam
<b>Tahmin Edilen Değer</b>	Pozitif	(TP) 1514	(FP) 38	1552
	Negatif	(FN) 45	(TN)	-
	Toplam	1559	-	-

Walk2.mpg videosunda 1040 görüntü bulunmaktadır. Önerilen yöntem kullanılarak videonun gerçek kesin değerleri içerisinde bulunmayan 1037 cisim tespit edilmiştir. Hata matrisi Çizelge 3.26’da gösterildiği üzere, anımsama değeri 0,93, kesinlik değeri 0,72 olarak elde edilmiştir. Şekil 3.55’te gösterildiği gibi, hassasiyet değerinin düşük olmasının nedeni, tamamen ayrı veya cisimlerin üst üste binmediği görüntülerin seçimi olduğu anlaşılmıştır. Birden fazla cisim içeren görüntüler söz konusu olduğunda, görüntünün yerel alt bölgelere bölünmesiyle yöntemin uygulanması, bu tür problemler için bir çözüm olarak uygulanabileceği düşünülmektedir. Şekil 3.55’te, sarı sınırlayıcı kutular, tespit edilip gerçek kesin değerler arasında bulunmayan cisimleri göstermektedir. Sol taraftaki sarı renkli kutu, gerçek gerçeğinde eksik olan cisimi, sağındaki ise yanlış cisim tespit sonucunu göstermektedir. Turkuaz renkli sınırlayıcı kutu, önerilen yöntemde algılanan cisimi ve mor renkli sınırlayıcı kutu, cismin gerçek kesin değerini göstermektedir.

**Çizelge 3.26.** Walk2.mpg hata matrisi

		<b>Gerçek Değerler</b>		
		Pozitif	Negatif	Toplam
<b>Tahmin Edilen Değer</b>	Pozitif	(TP) 2653	(FP) 1016	3669
	Negatif	(FN) 207	(TN)	-
	Toplam	2860	-	-



**Şekil 3.55.** Walk2.mpg'de karşılaşılan problem

Walk3.mpg videoda 1375 görüntü bulunmaktadır. Önerilen yöntem kullanılarak gerçek kesin değerler içerisinde bulunmayan 1125 cisim tespit edilmiştir. Hata matrisi Çizelge 3.27'de gösterildiği üzere, anımsama değeri 0,83, kesinlik değeri 0,98 olarak elde edilmiştir.

**Çizelge 3.27.** Walk3.mpg hata matrisi

		Gerçek Değerler		
		Pozitif	Negatif	Toplam
Tahmin Edilen Değer	Pozitif	(TP) 865	(FP) 18	883
	Negatif	(FN) 174	(TN)	-
	Toplam	1039	-	-

Çizelge 3.28'de tarafımızca laboratuvar ortamında toplanmış foto-kapan video veri setine ait bilgiler ve önerilen yöntemde kullanılan parametreler gösterilmiş olup, bu veri

setindeki videolar üzerinde önerilen yöntem ile cisim tespit işlemine ait sonuçlar Çizelge 3.29, Çizelge 3.30 ve Çizelge 3.31’de gösterilmektedir.

**Çizelge 3.28.** Tarafımızca toplanmış video veri seti ile ilgili bilgiler

Video Adı	Video10.mp4	Video1.mp4	Video27.mp4
Görüntü Sayısı	189	189	285
Cisim Sayısı	141	174	110
Videolarda Kullanılan Parametreler	Eşikleme: 25 Medyan Filtre: 7	Eşikleme: 25 Medyan Filtre: 7	Eşikleme: 17 Medyan Filtre: 11
Küçük Cisim Eleme Oranı	<%12	<%20	<%20
Görüntü Boyutları	1200x1600		

Çizelge 3.29’da gösterilen sonuçlarda 5 adet cisim tespit edilememesi durumu, cismin küçük olmasından kaynaklandığı anlaşılmakta olup, kesinlik 1.00, anımsama değeri 0.96 elde edilmiştir.

**Çizelge 3.29.** Video10.mp4 hata matrisi

		Gerçek Değerler		
		Pozitif	Negatif	Toplam
Tahmin Edilen Değer	Pozitif	(TP) 136	(FP) 0	136
	Negatif	(FN) 5	(TN) 34	-
	Toplam	141	-	-

Çizelge 3.30’da gösterilen sonuçlarda 1 adet hatalı cisim tespit edilme durumu, cismin gölgesinden kaynaklandığı anlaşılmakta olup, kesinlik 0.99, anımsama değeri 0.98 elde edilmiştir.

**Çizelge 3.30.** Video1.mp4 hata matrisi

		Gerçek Değerler		
		Pozitif	Negatif	Toplam
Tahmin Edilen Değer	Pozitif	(TP) 171	(FP) 1	172
	Negatif	(FN) 3	(TN) 15	-
	Toplam	174	-	-

Çizelge 3.31’de gösterilen sonuçlarda 12 adet hatalı cisim tespit edilme durumu, cismin gölgesinden kaynaklandığı anlaşılmakta olup, kesinlik değeri 0.90, anımsama değeri 1.00 elde edilmiştir.

**Çizelge 3.31.** Video27.mp4 hata matrisi

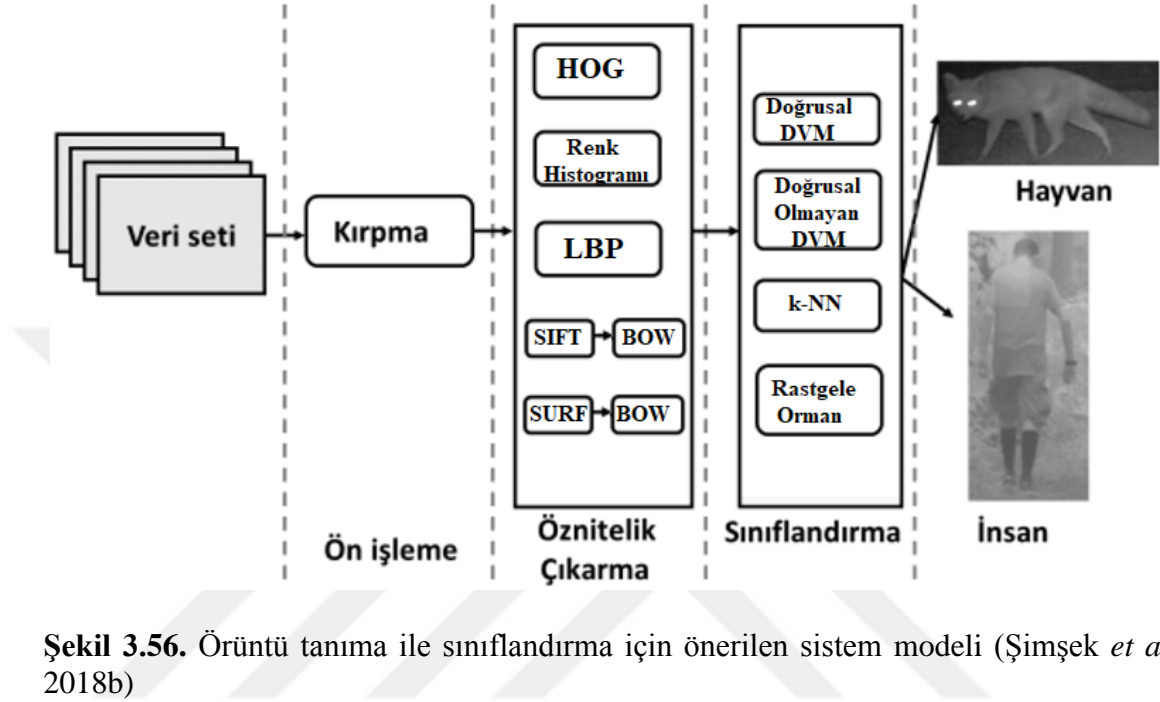
		Gerçek Değerler		
		Pozitif	Negatif	Toplam
Tahmin Edilen Değer	Pozitif	(TP) 110	(FP) 12	122
	Negatif	(FN) 0	(TN)	-
	Toplam	110	-	-

### 3.2.6. Örüntü Tanıma ve Sınıflandırma

Bu çalışmada (Şimşek *et al.* 2018b) insan ve hayvan görüntüleri içeren foto-kapan veri setinde cisim tanıma işlemi gerçekleştirilmiştir. Cisim tanıma için kullanılan sisteme ait akış şeması Şekil 3.56’da gösterilmektedir. Sistem ön işleme, öznelik çıkarma ve sınıflandırma olmak üzere üç ana asamadan oluşmaktadır. Ön işleme aşamasında insan ve hayvanlar ait görüntü parçaları arka plandan ayrılarak kaydedilmiştir. Kullanılan veri setinde görüntü içerisinde hareketli olan cisimlerin bulunduğu konum bilgileri bulunmaktadır. Öznelik çıkarma işleminde, sınıflandırma aşamasında kullanılmak üzere görüntülerin hem ayırt edici hem de değişmez yerel öznelikleri çıkartılmaktadır. Sınıflandırma aşamasında temel nokta gürültüye, ışığa, ölçeklenmeye, ötelemeye ve



dönmeğe karşı dayanıklı, ölçülebilir ve yüksek derecede ayırt edici özniteliklerin elde edilmesidir.



**Şekil 3.56.** Örüntü tanıma ile sınıflandırma için önerilen sistem modeli (Şimşek *et al.* 2018b)

HOG yönteminde görüntüde belirlenen her hücrenin yerel histogramı üzerinden eğim yönleri, kenar yönelimleri belirlenmektedir. Aydınlatma, gölgelendirme vb. durumlarda daha iyi dayanıklılık elde edilmesi için normalizasyon işlemi uygulanmaktadır (Dalal and Triggs 2005). HOG özniteliklerinin çıkarılması için ilk olarak Sobel Filtesi kullanılarak yatay  $I_x$  ve dikey kenarların  $I_y$  belirlenmesi gerekmektedir. Daha sonra öznitelik olarak  $\theta = \arctan I_x I_y$  (eğim) hesabı ile elde edilmektedir. Renk Histogramı, görüntünün rengi en yaygın kullanılan özniteliktir. Renk histogramı çıkarımında literatürde genel olarak HSV renk uzayı kullanılmaktadır. HSV renk modeli üç farklı Hue(H), doyum (S) ve değer (V) niteliğini ifade etmektedir. Hue cisimden yansıyan ışığın dalga boyunu, doyum rengin derinliğini ve değerde rengin parlaklığını temsil etmektedir. Bu çalışmada HSV betimleyicisi olarak H,V,S'in her bir parametrik değerini 16,12,12'lik kutular ayarlayarak 40 boyutlu histogram vektörü çıkarılmıştır. Renk histogramı özniteliği ışık değişimine karşı hassas olmasına rağmen etkili sonuçlar verdiği gözlemlenmiştir (Swain and Ballard 1991). Yerel İkili Desen (LBP),

bir görüntünün piksellerini, her pikseli eşik değere göre etiketleyen ve ikili sayı olarak ifade eden bir desen operatörüdür (Pawar and Belagali 2010). LBP öznitelikleri çıkartılırken  $r$  yarıçap içerisinde alınacak örnek sayısı  $n$  ve 0-1 dönüşüm sayısı ayarlanması gerekmektedir. LBP yöntemi, gri ölçekli değişimlere dayanıklıdır. Yöntem "uniform" parametresi ile birlikte kullanıldığında, görüntüden çıkan öznitelik sayısı azalmakta ve dönmeye karşı dayanıklılık elde etmektedir (Ngoc, 2014). SIFT, bu yöntemde her bir görüntüden elde edilen SIFT tanımlayıcılarından K-ortalamalar yöntemi ile belli sayıdaki görsel kelimeye indirgenmektedir. SIFT yöntemi her 4x4'lük hücre için 8 adet yön histogramı kullanmaktadır. Bu işlem bütün görüntü üzerinde tekrarlanarak kelime torbası yaklaşımı (Bag of Words) ile kelimelerin görüntü histogramı elde edilmektedir. Ölçeklendirmeye ve dönmeye dayanıklı olan bu yöntem, ışık değişimi, bakış açısı farkı, oklüzyon ve karışık arka plana kısmen dayanıklıdır. Ancak SIFT öznitelikleri işlem fazlalığı, karmaşıklığı ve fazla belleğe ihtiyaç duymasından dolayı gerçek zamanlı uygulamaya uygun değildir (Rani *et al.* 2018). Öznitelik çıkarma yöntemlerinde kullanılan parametreler Çizelge 3.32'de gösterilmiştir. Sınıflandırma, hedef sınıf tespitinde bir sınıflandırıcı algoritması kullanılması gereklidir. Literatürde yaygın olan sınıflandırma yöntemlerinden En Yakın Komşu (k-NN), Destek Vektör Makineleri (SVM), Rasgele Orman (RF) çalışmamızda kullanılmıştır. Çalışmamızda sınıflandırma yöntemlerinden KNN algoritmasında komşuluk sayısı 3, Rastgele Orman algoritmasında tahmin operatör sayısı olarak 50, SVM algoritmasında ise çapraz doğrulama parametresi 5 kullanılmıştır.

Çalışmada görüntü tanıma ve sınıflandırma için yapılan çalışmada kullanılan veri seti, rasgele olarak %75 eğitim ve %25 test verisi olarak ikiye ayrılmıştır. Çizelge 3.33'te yöntemimizin veri setindeki tüm görüntülere uygulanmasıyla elde edilen başarı oranları gösterilmiştir. HOG haricinde diğer öznitelikler için sınıflandırmada rasgele orman algoritması ile daha yüksek sonuçların elde edildiği görülmektedir. Ancak HOG öznitelikliğini çıkartırken kullanılan çerçeve büyüklüğü azaldıkça k-NN algoritmasının başarısının diğer yöntemlere göre daha fazla arttığı gözlemlenmiştir. Çizelge 3.34'te bütün yöntemler için hata matrisleri verilmiştir. LBP öznitelikliğini için SVM sınıflandırma yönteminde insan görüntülerinin hiç ayırt edilemediği gözlemlenmiştir. Hata matrisi

incelendiğinde genel olarak insanların hatalı bir şekilde havyan olarak sınıflandırıldığı görülmektedir.

**Çizelge 3.32.** Öznitelik dönüşüm yöntemlerinde kullanılan parametreler

Öznitelik Türü	Öznitelik Boyutu	Parametreler
HOG	2916	Yönlendirme=9 Hücre Başına Piksel=(10,10) Blok Başına Hücre=(2,2)
Renk Histogramı	2304	Histogram Büyüklüğü=(16,12,12) Kanal Sayısı=(0,1,2), Renk Uzayı=HSV Ölçek=(0,180,0,256,0,256)
SIFT	1024	Flann Parametresi=5 Ağaç Bow K-Ortalama Eğitimi=8
LBP	1872	Nokta Sayısı=24, Yarıçap=3 Yöntem=Düzenli

**Çizelge 3.33.** Dengesiz veri setinin sınıflandırma işlemine ait doğruluk oranları (%)

Öznitelik	HOG	Renk Hist.	SIFT	LBP	SURF
Doğrusal DVM	97,2	92,1	84,9	81,7	85,1
Doğ. Olmayan DVM	97,4	89,6	84,7	81,7	85,9
k-NN	92,6	97,5	85,4	91,9	88,6
Rastgele Orman	95,8	98,7	92,8	95,5	95,4
SVM(Çapraz Doğrulama)	93,0	86,7	82,6	84,0	81,7
Derin Öğrenme - CNN	95,2				

**Çizelge 3.34.** Sınıflandırma işlemine ait hata matrisi

Öznitelik	Doğrusal DVM	k-NN	Rastgele Orman	D. O. DVM
HOG	428 4 11 86	431 1 39 59	430 2 22 75	429 3 11 86
Renk His.	432 0 42 55	423 9 4 93	431 1 7 90	405 27 28 69
SIFT	431 3 79 1	399 35 26 54	431 3 24 56	436 0 79 0
LBP	432 0 97 0	414 18 25 72	431 1 22 75	432 0 97 0
SURF	440 4 73 0	408 36 23 50	439 5 29 54	444 0 73 0

Elde edilen sonuçlar 353 insan 1782 havyan görüntüsü bulunduğu dengesiz veri kümesine aittir. Bu dengesizliği gidermek için eşit sayıda (337) insan ve hayvan resimleri rasgele seçilerek yeni bir veri kümesi oluşturulmuş ve elde edilen sonuçlar Çizelge 3.35'te gösterilmiştir. Ayrıca gece ve gündüz çekilmiş fotoğraflar ayrıştırılarak yeni iki veri kümesi oluşturulmuş ve sınıflandırmaya tabi tutulmuştur. Gündüz çekilen görüntülere ait sonuçlar Çizelge 3.36'da, gece görüntülerine ait sonuçlar Çizelge 3.37'de gösterilmiştir. Görüldüğü üzere gece çekilen görüntülerde sınıflandırma başarısı daha yüksektir. Bu durumun nedeninin gece çekimlerinde flaş kullanılması olduğu düşünülmektedir. Flaş kullanılan görüntülerde yakındaki cisimler ışığı daha çok yansıtarak arka plandan daha fazla ayırttığı anlaşılmaktadır.

**Çizelge 3.35.** Dengeli veri setinin sınıflandırma işlemine ait doğruluk oranları (%)

Öznitelik	HOG	Renk His.	SIFT	LBP	SURF
Doğrusal DVM	95,9	91,1	64,5	72,8	70,7
k-NN	88,2	97,0	81,9	84,6	83,8
Rastgele Orman	93,5	97,0	83,7	88,8	86,2
DVM(Çapraz Doğrulama)	87,3	59,1	67,8	71,7	68,0
Doğ. Olmayan DVM	93,5	87,6	83,1	54,5	66,5
Derin Öğrenme - CNN	95,7				

**Çizelge 3.36.** Gündüz kaydedilmiş görüntülerin sınıflandırma işlemine ait doğruluk oranları (%)

Öznitelik	HOG	Renk His.	SIFT	LBP	SURF
Doğrusal DVM	93,3	93,8	73,0	73,6	76,2
k-NN	92,8	93,3	81,8	89,6	86,5
Rastgele Orman	92,2	95,3	80,7	91,2	85,5
DVM(Çapraz Doğrulama)	90,0	74,4	68,7	70,0	67,9
Doğ. Olmayan DVM	94,3	86,5	72,5	75,7	71,0

**Çizelge 3.37.** Gece kaydedilmiş görüntülerin sınıflandırma işlemine ait doğruluk oranları (%)

Öznitelik	HOG	Renk His.	SIFT	LBP	SURF
Doğrusal DVM	98,2	92,3	91,3	92,3	92,6
k-NN	97,9	98,2	96,3	96,7	93,8
Rastgele Orman	97,6	98,8	97,8	99,1	98,5
DVM(Çapraz Doğrulama)	95,4	91,8	91,7	92,1	92,0
Doğ. Olmayan DVM	98,2	92,3	92,9	92,3	92,6

### 3.3. Veri Setleri

Çevrim dışı olarak öznitelik çıkarma yöntemlerinin ve sınıflandırma algoritmalarının sınanması, sınıflandırma işlemine kullanılacak sınıflara ait modellerin çıkarılması amacıyla foto kapanlardan elde edilen, tarafımızca arka plandan ayrılarak kırılan veri seti (Anonymous, 2018a) kullanılmıştır. Kays ve arkadaşlarının yaptığı çalışmada (Kays *et al.* 2010) foto-kapan ile doğal ortamda kaydedilen toplam 2135 adet insan ve hayvan görüntüsü bulunmaktadır. Bu görüntülerden gündüz kaydedilenler renkli, gece kaydedilenler ise kızıl ötesi flaş kullanılarak gri-ton olarak bulunmaktadır. Veri seti hayvan türlerini tespit etmek için oluşturulduğu için dengeli olmayıp, 353 insan 1782 hayvan görüntüsünden meydana gelmektedir. 1782 hayvan içeren görüntü 20 farklı

türde hayvan içermektedir. Şekil 3.57’de kullanılan veri setindeki hayvan ve insan görüntülerinden örnekler verilmiştir.



**Şekil 3.57.** Örnek foto-kapan görüntüleri

Hareketli cisim tespit etme çalışmasında foto-kapanlar kullanılarak doğal ortamdan elde edilen, insan ve hayvan içeren ve iki bölümden oluşan kırılmamış veri seti (Kays *et al.* 2010) kullanılmıştır (Anonymous, 2018a). Veri setinin birinci bölümünde 100 arka plana sahip 1276, ikinci bölümünde ise 215 farklı arka plana sahip 1806 görüntü bulunmaktadır. 1920x1080 boyutlarında olan bu görüntüler gece ve gündüz kaydedilmiştir. Veri setinde, bu görüntülerde bulunan cisimlere ait tür, konum ve büyüklük bilgileri kesin referans (ground-truth) olarak bulunmaktadır. Veri setinin birinci bölümünde 859 görüntüde bir adet hareketli cismin, 88 adet görüntüde ise birde fazla hareketli cismin parçalı veya tam görüntüsü bulunmaktadır. 329 adet görüntüde ise hareketli cisim bulunmamaktadır. Veri setinin ikinci bölümünde ise 771 görüntüde bir adet cisim, 85 görüntüde birden fazla cisim bulunmaktadır. İkinci veri setinde 950 görüntüde ise hareketli cisim bulunmamaktadır.

Hareketli cisim tespiti ile ilgili önerilen yöntemin performans analizinin yapılması amacıyla Edinburgh Üniversitesi'ndeki CAVIAR projesinde kaydedilen bağlantıdan (Anonymous, 2018b) alınan 3 adet video kullanılmıştır. Sabit kameralardan gündüz kaydedilmiş videolardaki görüntü boyutları 384x288 piksel olup, 24 bitlik RGB formatındadır. Görüntülerde birden çok cisim, farklı rotalarda hareket etmektedirler. Birinci videoda 609, ikinci videoda 1040 ve üçüncü videoda 1375 görüntü bulunmakta olup, örnek video görüntüsü Şekil 3.58'de gösterilmektedir.

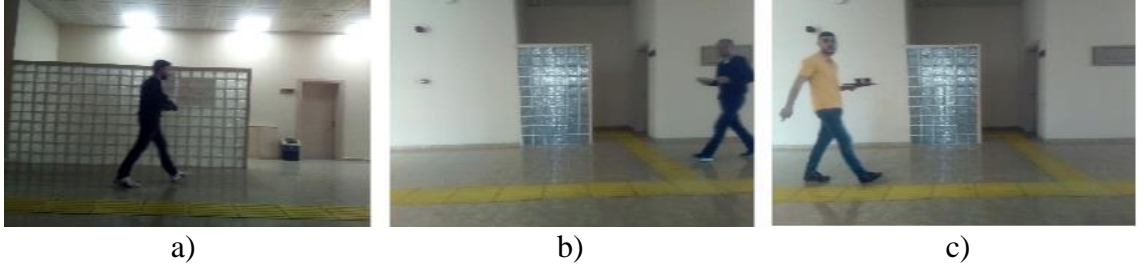


**Şekil 3.58.** Hareketli cisim tespiti için kullanılan veri setinden örnek görüntü görüntüler a) walk1.mpg b) walk2.mpg c) walk3.mpg

Çizelge 3.38'de tarafımızca laboratuvar ortamında toplanmış foto-kapan video veri setine ait bilgiler gösterilmektedir. Bu veri setinde bulunan videolar 3.2.5 başlığında önerilen Akıllı Üç Çerçeve Farkı yönteminde kullanılmış olup, tamamı iç ortamda kaydedilmiştir. Videolar, yöntemin diğer çalışmalar ile farkının etkili bir şekilde ifade edilebilmesi amacıyla, ışık şiddeti değişimi, gölge, arka plan karmaşıklığı veya arka plan hareketi gibi zorlukları en az seviyede içermektedir.

**Çizelge 3.38.** Tarafımızca Toplanmış Video Veri Seti ile İlgili Bilgiler

Video Adı	Video10.mp4	Video1.mp4	Video27.mp4
Görüntü Sayısı	189	189	285
Cisim Sayısı	141	174	110
Görüntü Boyutları	1200x1600		

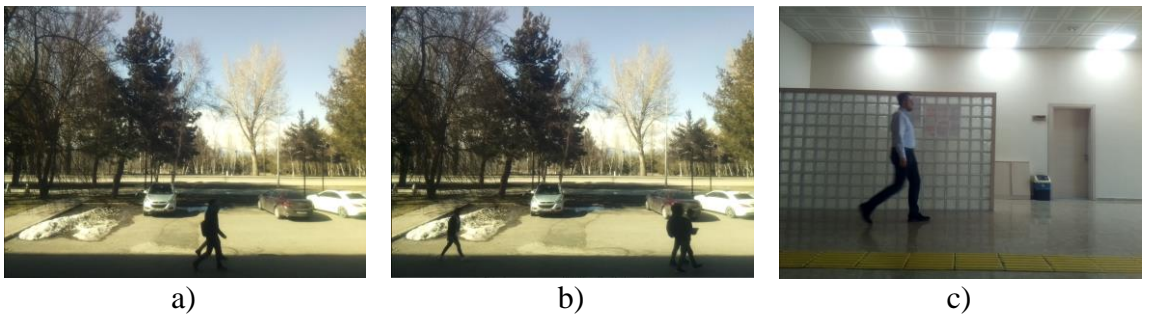


**Şekil 3.59.** Akıllı foto-kapan ile kaydedilmiş örnek görüntüler a) video10.mp4 b)video1.mp4 c)video27.mp4

Senaryolardaki bellek, işlem ve iletişim gereksinimlerinin analiz edilebilmesi amacıyla tez kapsamında tasarlanmış akıllı nesne ile Atatürk Üniversitesi Mühendislik Ek Bina konumunda kaydedilen veri setine (Anonymous, 2019a) ait bilgiler Çizelge 3.39'da ifade edilmekte olup, örnek görüntüler Şekil 3.60'ta gösterilmektedir. Veri seti iç mekân ve dış mekân olmak üzere 6 adet video içermektedir. Videolar, 1600x1200 piksel boyutlarında saniyede 32 görüntü içermekte olup, 24 bit rgb formatındadır.

**Çizelge 3.39.** Veri setine ait görüntü ve cisim sayıları

Video Adı	indoor2-1	indoor2-49	indoor1-20	indoor1-30	outdoor-18	outdoor-30
<b>Görüntü Sayısı</b>	238	190	191	190	238	334
<b>Cisim Bulunan Görüntü Sayısı</b>	108	104	127	158	216	334
<b>Cisim Sayısı</b>	108	104	127	158	210	270



**Şekil 3.60.** Videolara ait örnek görüntüler a)outdoor-18 b)outdoor-30 c)indoor1-30

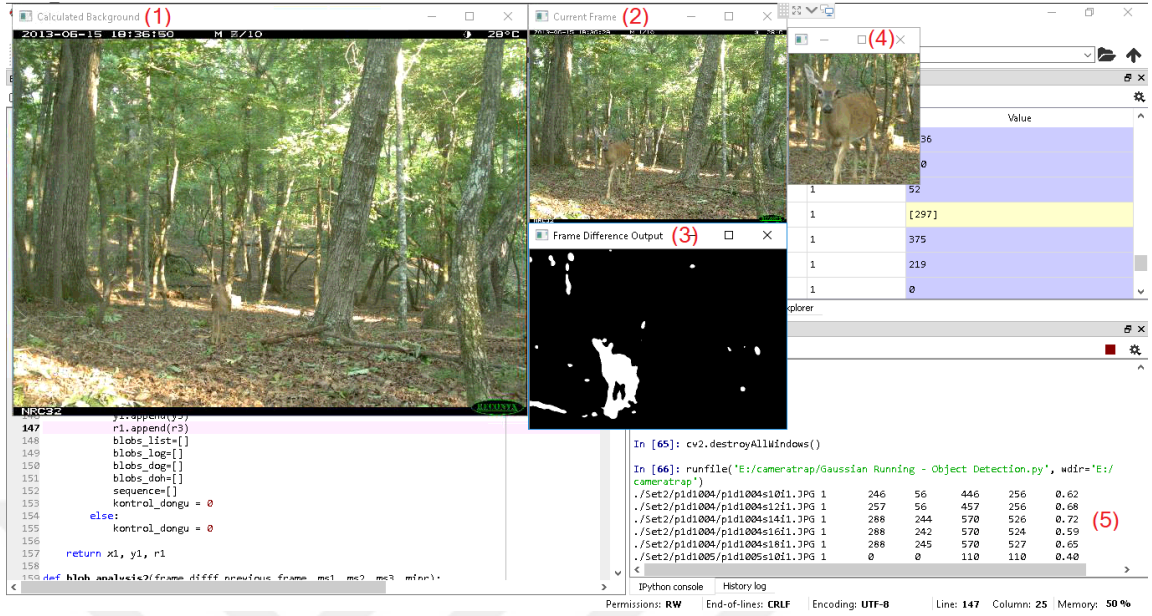


## 4. ARAŞTIRMA BULGULARI ve TARTIŞMA

### 4.1. Yazılım Geliştirme Platformu ve Görsel Uygulama

Çalışmamız, tümleşik python dağıtımı içeren Anaconda programının Spyder alt programı kullanılarak gerçekleştirilmiştir. Arayüze ait örnek görüntü Şekil 4.1’de gösterilmiş olup, Python sürümü olarak 2.7 ve 3.6 kullanılmıştır. Çalışmada önerilen yöntemler ve diğer görüntü işleme aşamaları, bilgisayarlı görü 2 (CV2) ve Python Görüntü Kütüphaneleri (PIL) kullanılarak gerçekleştirilmiştir. Şekil 4.1’deki cisim tespiti örneğinde görüldüğü üzere, 3.2.1. başlığında önerilen yöntemdeki algoritma çalıştırıldığında öncelikle tetiklemede elde edilen görüntüler kullanılarak bir numaralı çıktı olan arka plan modeli elde edilmektedir. Elde edilen arka plan modeli ile iki numara olarak işaretlenmiş her görüntüye çerçeve farkı yöntemi uygulanmakta ve üç numaralı çıktı elde edilmektedir. Üç numaralı çıktı üzerinde sınırlayıcı kutu belirleme işlemi uygulanarak dört numaralı çıktı olan görüntüdeki hareketli cisim elde edilmektedir. Elde edilen sonuca ait dikey ve yatay koordinat bilgileri beş numara ile işaretlenmiş kısımdan görülebileceği üzere metin dosyasına kaydedilmekte ve devamında kesin referans (ground-truth) değerleri ile karşılaştırılmaktadır.

Spyder arayüzünde geliştirilen Python algoritmaları, gömülü sistem modülü olan Raspberry pi üzerinde uygulanmıştır. Üzerinde hareket algılayıcısı, kamera, wi-fi modülü bulunan raspberry pi mikro bilgisayar ile hareket algılandığında video görüntüsü kaydetmektedir. Geliştirdiğimiz algoritmada 1600x1200 piksel boyutlarında saniyede 32 görüntü olmak üzere hareket devam ettiği sürece devam eden kayıt işlemi, hareket bittikten 1,5 saniye sonra durmaktadır.



Şekil 4.1. Uygulamaya ait görüntü

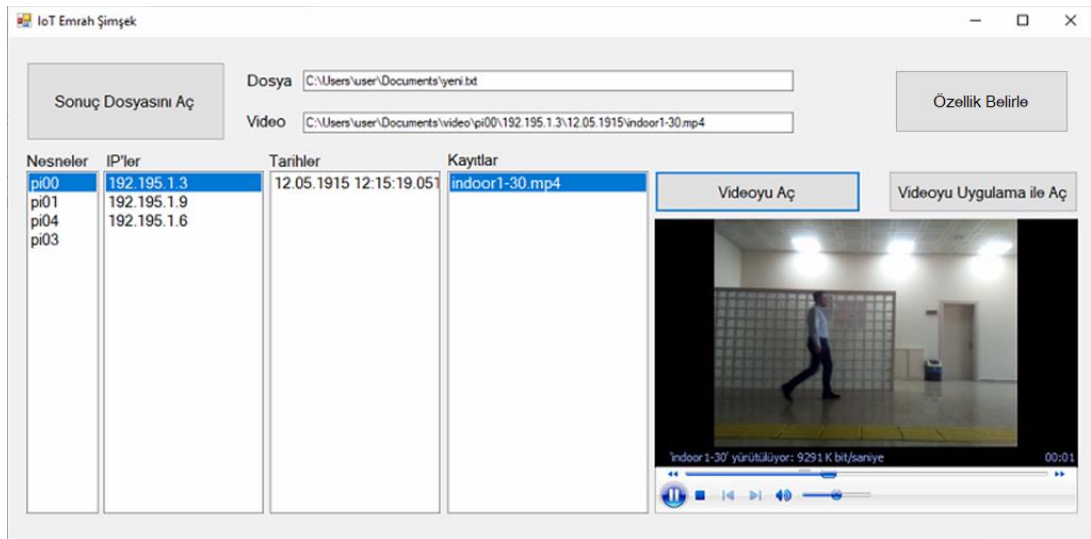
Çalışmada kullanılan Raspberry pi model 3 b'nin özellikleri aşağıda listelenmektedir:

- Quad Core 1.2GHz Broadcom BCM2837 64bit CPU
- 1GB RAM
- BCM43438 Kablosuz Ağ ve Bluetooth
- 40 Adet GPIO Pini
- 4 USB 2 Portu
- HDMI Çıkışı
- Raspberry pi Kamera için CSI Portu
- İşletim Sistemi ve Bellek için Micro SD Portu

Kullanılan raspberry pi model 3 mikro denetleyici, işletim sistemi içerebilmesi, dâhili wifi modülü olması, usb flaş bellek girişleri bulunması gibi nedenlerle tercih edilmiştir. İşletim sistemi olarak Python kodlarının kolayca uygulanabileceği Raspbian Stretch kullanılmıştır. Bu mikro bilgisayar kullanılarak kaydedilen video görüntülerinde, akıllı nesne işlevlerinin her senaryosu ayrı ayrı uygulanarak işlem, veri ve iletişim

gereksinimleri analiz edilebilmektedir. Ayrıca üzerindeki genel amaçlı giriş çıkış pinleri (GPIO) kullanılarak hareket algılayıcısı gibi modüller kolayca entegre edilebilmektedir.

Sunucuda kullanılmak üzere ise Şekil 4.2’de gösterilen görsel program C# dilinde tarafımızca yazılmış olup, akıllı nesnelere gönderilen hareketli cisim görüntülerine ait kayıtlar gösterilmektedir. Geliştirilen görsel uygulamada akıllı nesnelere ilk sütunda “pi00” şeklinde gösterilmektedir. Sisteme entegre edilen her akıllı nesne, sisteme otomatik olarak yeni nesne adıyla kaydedilmektedir. Akıllı nesnelere gelen foto-kapan verileri, her akıllı nesne adına otomatik olarak oluşturulan klasörlerin içerisinde, gönderilen IP ve tarih bilgisine göre oluşturulan alt klasörlerin içerisine kaydedilmektedir. Sırasıyla akıllı nesne adı, IP, tarih klasörlerinin içerisine kaydedilen foto-kapan videoları, ilgili klasördeki video sayısı ve tarih bilgisi kullanılarak isimlendirilmektedir. Görsel uygulamada akıllı nesnelere alınmış videolar isteğe göre Windows medya oynatıcı penceresinde veya uygulama dışındaki bir medya uygulamasında oynatılabilmektedir. Şekil 4.3’te gösterilen görsel uygulamanın ikinci sekmesinde akıllı nesnelere alınan kayıtların özellikleri ve bu özelliklerin ayarlanabileceği menüler gösterilmektedir. Akıllı nesnenin hangi senaryo ile çalışacağı, gönderilecek verilerin özellikleri veya videoların kare sıklığı ve boyutları ayarlanabilmektedir.



Şekil 4.2. Sunucuda kullanılan görsel uygulama

**Özellikler**

**Görüntü Boyutları**

- 1920x1600
- 1600x1200
- 1200x900
- 900x720
- 600x480
- 360x240

**Video için Kayıt Sıklığı**

- 60 frame/second
- 45 frame/second
- 32 frame/second
- 30 frame/second
- 25 frame/second
- 15 frame/second
- 10 frame/second

**Senaryo Seçimi**

- Senaryo 1 - Bütün Görüntüler
- Senaryo 2 - Cisim İçeren Görüntüler
- Senaryo 3 - Cisim Görüntü Parçaları
- Senaryo 4 - Öznitelikler
- Senaryo 5 - Cisim Sınıfları

**Öznitelikler**

- Renk Histogramı
- Yerel İkili Örüntü
- HOG
- SIFT
- SURF
- BRIEF
- ORB

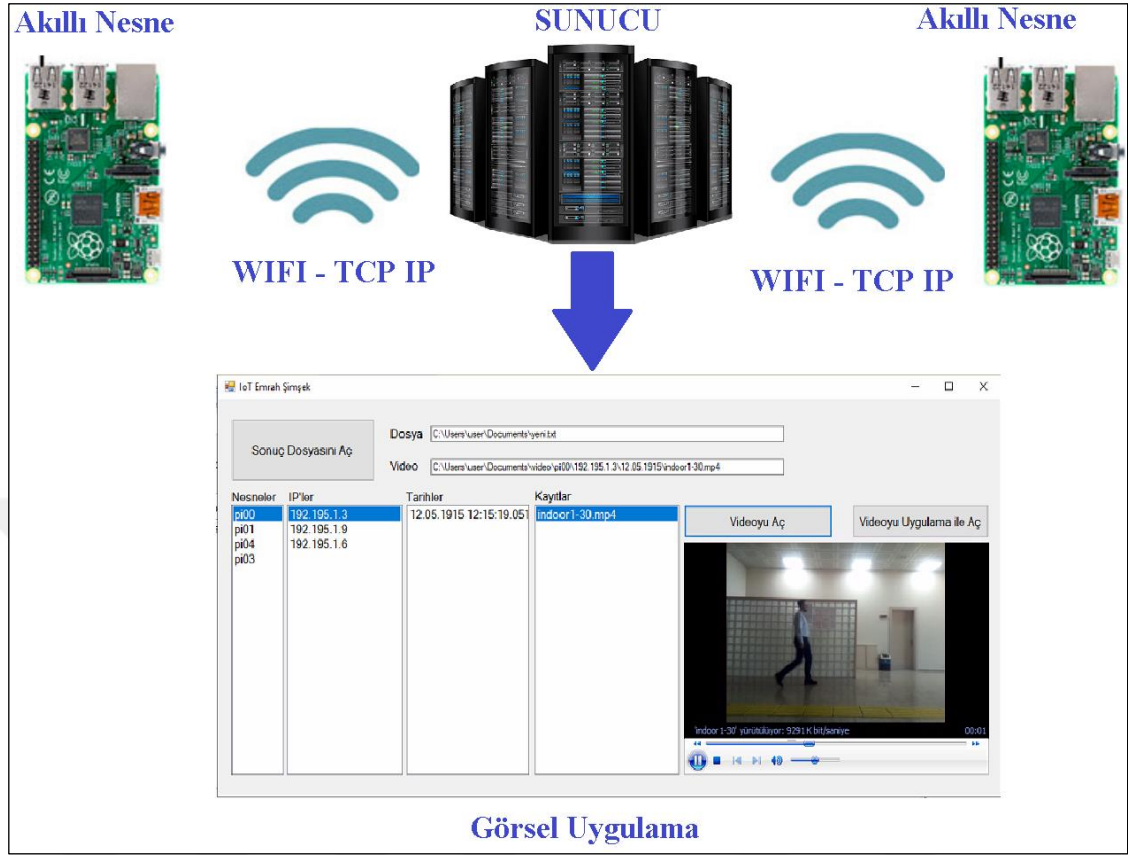
**Sınıflandırma Yöntemleri**

- k En Yakın Komşu
- Rastgele Orman
- Doğrusal Destek Vektör Makineleri
- Doğrusal Olmayan Destek Vektör Makineleri
- Yapay Sinir Ağları (CNN)
- Bölge Tabanlı Evrişimli Sinir Ağları (YOLO)

**Kaydet**

**Şekil 4.3.** Sunucuda kullanılan görsel uygulamada özellik belirleme sayfası

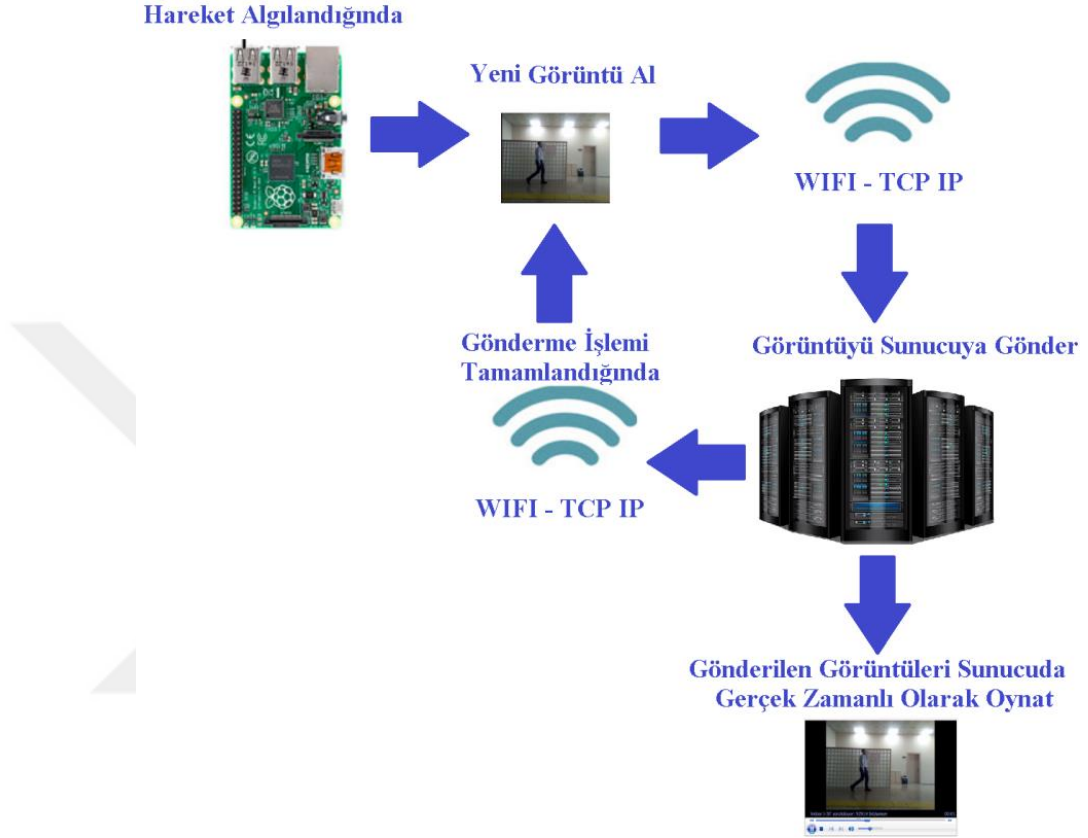
Şekil 4.4'te ise çalışmaya ait iletişim sistemi gösterilmektedir. Çalışmamızda akıllı nesnelere toplanan veriler yerel wi-fi ağı aracılığıyla TCP kullanılarak sunucu bilgisayara iletilmekte ve kaydedilmektedir. Sunucu bilgisayardan akıllı nesnelere gönderilecek veriler de aynı protokol aracılığı ile gönderilmektedir.



**Şekil 4.4.** Çalışmadaki iletişim sistemi

Çalışma kapsamında yapılan akıllı nesnelerin hareket algıladığında gerçek zamanlı video görüntüsünü sunucuya gönderdiği ve ekranda gösterildiği uygulamada, Şekil 4.5'te gösterildiği üzere ardışık video görüntüsünün sıradaki karesi, sunucuya gönderilmektedir. İlgili görüntü karesi sunucuya tamamen iletdikten sonra yeni görüntü karesinin gönderilmesine geçilmektedir. Bu şekilde video görüntülerinin kaydedilmesinden kaynaklanan gecikme bulunmamaktadır. İletişim alt yapısı ve donanımdan kaynaklanan 0,2-0,5 saniye arasında gecikme olabilmektedir. Gönderilecek görüntülerde otomatik cisim tespit etme işlemi de gerçekleştirilebilmekte olup, yapılan bu ek işlemler gecikmeyi veya videoda saniye başına kaydedilen görüntü sayısını azaltmaktadır. Yapılan uygulamada standart 32 kare olarak kaydedilen görüntünün, sunucu bilgisayara saniyede 16-18 kare olarak iletilebildiği anlaşılmaktadır. Bu durumu kullanılan raspberry pi modülünün özelliklerine, gönderilecek görüntünün kalitesine,

wi-fi ağının hızına ve yoğunluğuna, sunucudaki uygulamada gerçekleştirilen işlemlere göre farklılık gösterebilmektedir.



**Şekil 4.5.** Hareketli cismin sunucuda gerçek zamanlı olarak görüntülenmesi

## 4.2. Bulgular

Veri setlerinde görüntülerin cisim içerme durumu ile ilgili yapılan çalışmanın gösterildiği Çizelge 4.1'den anlaşıldığı üzere, tezde kullanılan veri setlerindeki görüntülerin sırasıyla % 25, % 59, % 12'sinde herhangi bir cisim bulunmamaktadır. Çalışmada senaryo 2 olarak ifade edilen bu yaklaşımda, hareketli cisimlerin tespit edilmesi durumunda, veri iletişimde ortalama %40'a yakın kazanç elde edildiği ve sunucuda yapılması gereken önemli bir aşama olan hareketli cisimlerin tespit işleminin yapılmasına gerek kalmadığı anlaşılmaktadır. Ayrıca Çizelge 4.1'de son satırda gösterildiği üzere, cisimlere ait görüntü parçalarının bütün görüntülere oranı ortalama %

4,7 olduğu anlaşılmaktadır. Senaryo 3 olarak ifade edilen hareketli cisimlere ait görüntü parçalarının kırılarak gönderilmesi işleminde sunucularda hareketli cisimlerin tespit edilmesi, konumlarının belirlenmesi ve kırılması işlemleri yapılmasına gerek kalmamasının yanı sıra, iletişim işleminde gereksinimin % 95,3 azalabileceği görülmektedir.

**Çizelge 4.1.** Veri setlerindeki görüntülerin ve cisimlerin özellikleri

	<b>1. Veri Seti</b>	<b>2. Veri Seti</b>	<b>3. Veri Seti</b>
<b>Görüntü Sayısı</b>	1277	2224	3025
<b>Görüntü Boyutları (Piksel)</b>	1920x1080	2048x1536	384x288
<b>Cisim Bulunmayan Görüntü Sayısı</b>	330	1331	379
<b>Cisim Bulunmayan Görüntülerin Bütün Görüntülere Oranı (%)</b>	25,8	59,8	12,5
<b>Görüntülerdeki Cisim Sayısı</b>	1109	1034	5512
<b>Ortalama Cisim Boyutları</b>	315x260	375x458	43x116
<b>Cisim Görüntülerinin Bütün Görüntülere Oranı (%)</b>	3,5	2,5	8,2

Tasarlanan akıllı nesne kullanılarak cisim tespit etme ve cisim tanıma ile ilgili işlevlerin gerçekleştirilmesi amacıyla 3 farklı konumdan 4 adet iç ortam, 2 adet dış ortam videosu kaydedilmiştir. Akıllı nesnede hareketli cisim varlığının tespit edilmesi amacıyla sırasıyla, 2001 yılında önerilmiş gaussların karışımı yöntemi, gaussların karışımının 2004 ve 2006 yıllarında geliştirilmiş bir versiyonu, hareketli cisim tespiti amacıyla tez kapsamında önerilen akıllı üç çerçeve farkı yöntemi ve literatürde en az bellek ve işlem gücü gereksinimi olan değişen gauss ortalama yöntemi uygulanmıştır.

Tasarlanan akıllı nesne ile kaydedilen görüntülerde hareketli cisim tespit sonuçları Çizelge 4.2’de gösterilmektedir. Hareketli cisim varlığı, elde edilen çıktının %10’unda hareket algılanması durumuna göre karar verilmiştir. Çizelge 4.3’te bu hareketli cisim tespitine ait süreler, Çizelge 4.4’te ise bu cisimlerin konumlarının tespit edilmesi ve kırılmasına ait süreler mili saniye cinsinden gösterilmektedir. Tespit edilen hareketli cisimlerin görüntüdeki konumlarının belirlenmesi amacıyla bağlı bileşen analizi yöntemi kullanılmıştır. Çizelgelerde en etkili sonucu veren değerler koyu renkli olarak belirtilmiştir.

**Çizelge 4.2.** Hareketli cisim bulunan görüntülerin tespit başarıları (%)

	indoor2-1	indoor2-49	indoor1-20	indoor1-30	outdoor-18	outdoor-30
Gaussların Karışımı 2001	98,32	<b>100</b>	<b>100</b>	98,42	78,99	85,63
Gaussların Karışımı 2004-6	<b>98,74</b>	98,42	97,91	98,42	96,64	93,41
Önerilen Yöntem	<b>98,74</b>	99,47	98,95	96,84	84,87	<b>98,5</b>
Değişen Gauss Ortalama + Çerçeve Farkı	94,54	95,26	<b>100</b>	<b>98,95</b>	<b>97,48</b>	<b>98,5</b>

**Çizelge 4.3.** Hareketli cisim bulunan görüntülerin tespit süreleri (mili saniye)

Video Adı	indoor2-1	indoor2-49	indoor1-20	indoor1-30	outdoor-18	outdoor-30
Gaussların Karışımı 2001	35,241	28,629	28,166	28,730	35,461	50,510
Gaussların Karışımı 2004-6	31,809	25,977	25,285	24,922	32,763	49,666
Önerilen Yöntem	45,620	31,748	31,560	32,282	42,917	54,978
Değişen Gauss Ortalama + Çerçeve Farkı	<b>29,896</b>	<b>24,854</b>	<b>23,805</b>	<b>23,661</b>	<b>31,711</b>	<b>42,441</b>



**Çizelge 4.4.** Hareketli cisimlerin sınırlayıcı kutu tespit süreleri (mili saniye)

Video Adı	indoor2-1	indoor2-49	indoor1-20	indoor1-30	outdoor-18	outdoor-30
Gaussların Karışımı 2001	41,164	33,510	34,766	35,438	42,359	54,937
Gaussların Karışımı 2004-6	41,283	33,694	34,797	36,565	42,107	61,412
Önerilen Yöntem	52,925	37,113	38,459	42,888	51,791	74,277
Değişen Gauss Ortalama	<b>35,391</b>	<b>29,423</b>	<b>30,863</b>	<b>31,690</b>	<b>39,768</b>	<b>60,912</b>

Çalışmada uygulanan yöntemler ile tespit edilen cisimlere ait görüntü parçaları kırılarak her yöntem için ayrı klasöre kaydedilmiştir. Kaydedilen bu görüntü parçalarının sayıları Çizelge 4.5’te, hatalı olarak tespit edilen cisim sayıları Çizelge 4.6 ve Çizelge 4.7’de, doğru tespit edilen cisim sayıları ise Çizelge 4.8’de gösterilmektedir. “outdoor-18” ve “outdoor-30” videoları diğer videolardan farklı olarak, birden fazla hareketli cisim ve arka plan hareketi içermektedir. Bu videolar dışındaki 4 videoda bir hareketli cisim bulunmaktadır. Cisim tespiti için önerdiğimiz yöntem ile Çizelge 4.9 ve Çizelge 4.10’daki hassasiyette %100 ile tam başarı elde edilmiş olup, anımsamada ise gaussların karışımı 2001 yöntemi ile birlikte en yüksek başarıya sahip olduğu anlaşılmaktadır.

**Çizelge 4.5.** Tespit edilen hareketli cisimlerin sayısı

Video Adı	indoor2-1	indoor2-49	indoor1-20	indoor1-30	outdoor-18	outdoor-30
Gaussların Karışımı 2001	108	104	125	157	166	115
Gaussların Karışımı 2004-6	115	103	121	153	209	270
Önerilen Yöntem	105	104	127	158	145	389
Değişen Gauss Ortalama	94	95	125	156	210	477
Gerçek Cisim Sayıları	108	104	127	158	210	270

Çizelge 4.6. (False Positive) Hatalı tespit edilen cisimlerin sayısı

Video Adı	indoor2-1	indoor2-49	indoor1-20	indoor1-30	outdoor-18	outdoor-30
Gaussların Karışımı 2001	0	0	0	0	0	0
Gaussların Karışımı 2004-6	8	2	0	0	0	0
Önerilen Yöntem	0	0	0	0	0	286
Değişen Gauss Ortalama	0	0	0	0	0	284

Çizelge 4.7. (False Negative) Hatalı tespit edilen cisimlerin sayısı

Video Adı	indoor2-1	indoor2-49	indoor1-20	indoor1-30	outdoor-18	outdoor-30
Gaussların Karışımı 2001	<b>0</b>	0	2	1	44	155
Gaussların Karışımı 2004-6	1	3	6	5	1	<b>0</b>
Önerilen Yöntem	3	<b>0</b>	<b>0</b>	<b>0</b>	65	151
Değişen Gauss Ortalama	14	9	2	2	<b>0</b>	77

Çizelge 4.8. (True Positive) Doğru tespit edilen cisimlerin sayısı

Video Adı	indoor2-1	indoor2-49	indoor1-20	indoor1-30	outdoor-18	outdoor-30
Gaussların Karışımı 2001	<b>108</b>	<b>104</b>	125	157	166	115
Gaussların Karışımı 2004-6	107	<b>104</b>	121	153	209	<b>270</b>
Önerilen Yöntem	105	<b>104</b>	<b>127</b>	<b>158</b>	145	103
Değişen Gauss Ortalama	94	95	125	156	<b>210</b>	193

**Çizelge 4.9.** Hassasiyet (precision) değerleri

Video Adı	indoor 2-1	indoor2-49	indoor1-20	indoor1-30	outdoor-18	outdoor-30	Ortalama Başarı
Gaussların Karışımı 2001	1,00	1,00	1,00	1,00	1,00	1,00	<b>1,00</b>
Gaussların Karışımı 2004-6	0,93	0,98	1,00	1,00	1,00	1,00	0,99
Önerilen Yöntem	1,00	1,00	1,00	1,00	1,00	0,58	0,93
Değişen Gauss Ortalama	1,00	1,00	1,00	1,00	1,00	0,63	0,94

**Çizelge 4.10.** Anımsama (recall) değerleri

Video Adı	indoor 2-1	indoor2-49	indoor1-20	indoor1-30	outdoor-18	outdoor-30	Ortalama Başarı
Gaussların Karışımı 2001	<b>1,00</b>	<b>1,00</b>	0,98	0,99	0,79	0,43	0,87
Gaussların Karışımı 2004-6	0,99	0,97	0,95	0,97	<b>1,00</b>	<b>1,00</b>	<b>0,98</b>
Önerilen Yöntem	0,97	1,00	<b>1,00</b>	<b>1,00</b>	0,69	0,72	0,90
Değişen Gauss Ortalama	0,87	0,91	0,98	0,99	1,00	0,86	0,94

Tasarlanan akıllı nesnede kullanılarak elde edilen görüntülerde, hareketli cisim görüntülerinin özniteliklerinin çıkarılması ile ilgili bellek ve zaman analizleri Çizelge 4.11’de gösterilmektedir. Çizelge 4.11’de görüldüğü üzere en az bellek ve işlem ihtiyacı olan öznitelik dönüşümü yönteminin BRIEF olduğu anlaşılmaktadır.

**Çizelge 4.11.** Hareketli cisimler ile oluşturulan modellerin bellek ve zaman çizelgesi

Öznitelik	Renk Histogramı	HOG	LBP	SIFT	SURF	BRIEF	ORB
Bellek (Kbyte)	6,734	92	292	14,904	5,950	<b>17</b>	341
Zaman (Mili Saniye)	506	7,206	19,659	16,498	4,760	<b>228</b>	1,023

Sınıflandırma işleminde hareketli cisim tespit yöntemleri ile elde edilen cisim görüntülerinin, Çizelge 4.11’de gösterilen öznitelikleri kullanılarak sınıf modelleri oluşturulmuştur. Modeller oluşturulurken 975 adet insan görüntüsü kullanılmış olup, bu sınıf modellerinin sınıf içi sınıflandırma başarıları Çizelge 4.12’de ve bu modellerin bellekteki boyutları Çizelge 4.13’te gösterilmektedir. Bu sonuçlarda BRIEF ve ORB öznitelik dönüşümü yöntemlerinin bulunmamasının sebebi, ikili olarak elde edilen tanımlayıcılara sahip olan yöntemlerde belirtilen sınıflandırma algoritmaları için sınıf modeli oluşturulamamasıdır. SIFT ve SURF öznitelik dönüşümü yöntemlerinde sınıflandırma işlemlerinde kullanılabilecek sınıf modellerinin elde edilmesi amacıyla Görsel Kelime Çantası (Bag of Visual Words) kullanılmaktadır.

**Çizelge 4.12.** Hareketli cisimler ile oluşturulan modellerin sınıf içi başarıları (%)

<b>Öznitelik / Sınıflandırma Algoritmaları</b>	<b>Doğrusal DVM</b>	<b>Doğrusal Olmayan DVM</b>	<b>k-NN</b>	<b>Rastgele Orman</b>
Renk Histogramı	100,00	97,60	100,00	100,00
HOG	99,60	97,60	76,67	97,60
LBP	97,60	97,60	99,60	99,60
SIFT	97,59	97,59	99,47	99,20
SURF	97,45	97,45	99,01	99,29

**Çizelge 4.13.** Hareketli cisimlerin oluşturulan modellerinin bellekteki boyutları (KB)

<b>Öznitelik / Sınıflandırma Algoritmaları</b>	<b>Doğrusal DVM</b>	<b>Doğrusal Olmayan DVM</b>	<b>k-NN</b>	<b>Rastgele Orman</b>
Renk Histogramı	19	254	7,295	<b>33</b>
HOG	24	435	6,384	56
LBP	<b>1</b>	5	62	42
SIFT	<b>1</b>	<b>3</b>	<b>31</b>	57
SURF	<b>1</b>	<b>3</b>	28	56

Tez çalışması sonucunda önerilen bütün senaryolar ve yöntemler kullanılarak bellek, işlem ve iletişim gereksinimleri Çizelge 4.14, Çizelge 4.15, Çizelge 4.16, Çizelge 4.17, Çizelge 4.18 ve Çizelge 4.19’da gösterilmektedir.

**Çizelge 4.14.** Hareketli cisim görüntüleri ile oluşturulan modellerin bellekteki boyutları (KB)

	<b>Gaussların Karışımı 2001</b>	<b>Gaussların Karışımı 2004-2006</b>	<b>Önerilen Yöntem</b>	<b>Değişen Gauss Ortalama</b>
<b>Toplam Görüntü Sayısı</b>	1,381	1,381	1,381	1,381
<b>Toplam Veri (KByte)</b>	542.808	542.808	542.808	542.808
<b>1. Senaryo: Bütün Görüntüler (Kbyte)</b>	542.808	542.808	542.808	542.808
<b>2. Senaryo: Cisim Olan Görüntüler (KByte)</b>	297.767	383.008	386.066	387.595
<b>3. Senaryo: Cisim Görüntü Parçaları (KByte)</b>	1.336	2.053	1.807	1.804

**Çizelge 4.15.** 4. Senaryoya ait cisim görüntülerinin öznelik boyutları (KByte)

	<b>Gaussların Karışımı 2001</b>	<b>Gaussların Karışımı 2004-2006</b>	<b>Önerilen Yöntem</b>	<b>Değişen Gauss Ortalama</b>
Renk Histogramı	5,268	6,600	6,987	7,864
HOG	36	46	48	54
LBP	194	243	257	289
SIFT	11,663	14,613	15,471	17,412
SURF	4,624	5,794	6,134	6,904
BRIEF	6	8	8	9
ORB	239	299	316	356

**Çizelge 4.16.** 5. Senaryodaki doğrusal dvm sınıflandırma modellerinin boyutları (kbyte)

	<b>Gaussların Karışımı 2001</b>	<b>Gaussların Karışımı 2004-2006</b>	<b>Önerilen Yöntem</b>	<b>Değişen Gauss Ortalama</b>
Renk Histogramı	15,53	19,46	20,60	23,18
HOG	19,62	24,58	26,02	29,28
LBP	<b>0,80</b>	<b>1,00</b>	<b>1,06</b>	<b>1,19</b>
SIFT	0,82	1,02	1,08	1,22
SURF	0,81	1,01	1,07	1,21

**Çizelge 4.17.** 5. Senaryodaki doğrusal olmayan dvm sınıflandırma modellerinin boyutları (Kbyte)

	<b>Gaussların Karışımı 2001</b>	<b>Gaussların Karışımı 2004-2006</b>	<b>Önerilen Yöntem</b>	<b>Değişen Gauss Ortalama</b>
Renk Histogramı	207,60	260,10	275,36	309,92
HOG	355,53	445,44	471,59	530,77
LBP	4,09	5,12	5,42	6,10
SIFT	<b>2,45</b>	<b>3,07</b>	<b>3,25</b>	<b>3,66</b>
SURF	2,46	3,08	3,27	3,68

**Çizelge 4.18.** 5. Senaryodaki kNN Sınıflandırma Modellerinin Boyutları (KByte)

	<b>Gaussların Karışımı 2001</b>	<b>Gaussların Karışımı 2004-2006</b>	<b>Önerilen Yöntem</b>	<b>Değişen Gauss Ortalama</b>
Renk Histogramı	5,822	7.295	7,723	8,692
HOG	5,095	6,384	6,759	7,607
LBP	49	62	66	74
SIFT	25	31	33	37
SURF	22	28	30	33

**Çizelge 4.19.** 5. Senaryodaki rastgele orman sınıflandırma modellerinin boyutları (KByte)

	Gaussların Karışımı 2001	Gaussların Karışımı 2004-2006	Önerilen Yöntem	Değişen Gauss Ortalama
Renk Histogramı	26	33	35	39
HOG	45	56	59	67
LBP	34	42	44	50
SIFT	45	57	60	68
SURF	45	56	59	67

Çizelge 4.20’de ise ön ve son işleme aşamalarında uygulanan işlemlerin görüntü başına ortalama zaman gereksinimleri gösterilmektedir.

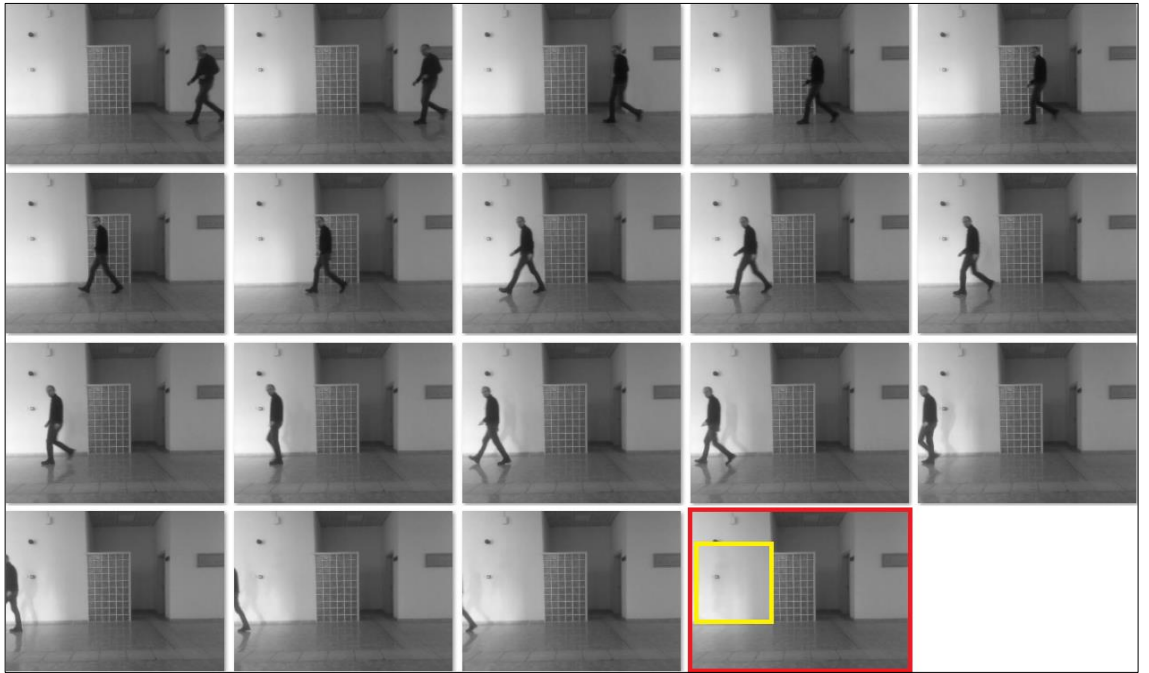
**Çizelge 4.20.** Ön ve son işleme aşamalarının görüntü başına ortalama zaman gereksinimi

	Medyan Filtre	Gauss Bulanıklığı	Eşikleme	Yeniden Boyutlandırma	Bağlı Bileşen Analizi	Çerçeve Farkı Yöntemi	Histogram Eşitleme
Zaman (Mikro Saniye)	7410	3793	64	3196	135	109	1077

Çalışmanın 3.2.5 başlığında Akıllı Üç Çerçeve Farkı yönteminde yeni bir görüntü seçme algoritması önerilmiştir. Algoritma ardışık görüntülerde cisimlerin üst üste gelmeyen konumlara sahip olmayan görüntülerin seçilmesi amacıyla kullanılmaktadır. Önerilen algoritma kullanılarak cismin hareketine ve hareket hızına bağlı olarak görüntüler seçilebilmekte ve kaydedilebilmektedir. Kullanılan algoritma ile hareket veya cisim içermeyen görüntülerin elenmesi, cisimlere ait farklı pozların yakalanması işlemleri gerçekleştirilebilmektedir. Bu yaklaşım ile ilgili veri setleri üzerinde yapılmış çalışmalar Çizelge 4.21’de gösterilmektedir. Algoritmanın foto-kapan veya akıllı nesne cihazlarında uygulanmaya uygun olması ve görüntü kaydetme ile ilgili özgün bir yaklaşım olması açısından özgünlük arz etmektedir. indoor2-49 videosuna ait çıktılarının bulunduğu Şekil 4.6’da sarı alan, kırmızı renk ile işaretlenmiş hatalı seçilen görüntüde hataya sebep olan gölgeyi göstermektedir.

Çizelge 4. 21. Önerilen görüntü seçme algoritması ile elde edilen sonuçlar

Video Adı	indoor2-1	indoor2-49	indoor1-20	indoor1-30	outdoor-18	outdoor-30
Görüntü Sayısı	238	190	191	190	238	334
Cisim Bulunan Görüntü Sayısı	108	104	127	158	216	334
Algoritma ile Seçilen Görüntü Sayısı	31	19	31	39	25	38
Cisim İçermeyen ve Hatalı Olarak Seçilen Görüntü Sayısı	1 (Gölge)	1 (Gölge)	0	0	2 (Arka Plan Hareketi)	0
Elenen Görüntü Sayısı	207	171	160	151	213	296
Elenen Görüntü Oranı	% 87	% 90	% 84	% 79	% 89	% 87



Şekil 4.6. indoor2-49 videosunda seçilen görüntülerden (kırmızı ile belirtilen hatalı)



### 4.3. Tartışma

Nesnelerin interneti alanında yapılan cisim tespit etme ve tanıma çalışmalarında yaygın olarak belirli bir öznitelik çıkarma ve etkili bir sınıflandırma yöntemi kullanılmaktadır. Bu çalışmalarda bulut veya kenar programlama tabanlı çözümler farklı senaryolar şeklinde uygulanmamakta ve bu senaryolara ait iş yükü dağılımı ve bellek, işlem, iletişim gereksinimleri göz ardı edilmektedir. Nesnelerin internetinde gerçekleştirilen cisim tanıma çalışmalarında, veri setleri ile eğitilerek oluşturulan cisim modelleri kullanılmaktadır. Sonuç odaklı sınıflandırma işlemi yapılan birçok uygulamada kullanılan modellerinde cisim sayısının artırılması ve cisim modelinin yeni veriler ile yeniden eğitilmesi mümkün olmamaktadır. Uygulamaların, tespit edilebilecek cisim sayısının sabit olması, eğitimde kullanılan verilerin kısıtlı olması ve arttırılamaması, farklı yöntem ve öznitelik kullanabilme gibi esnekliklerin bulunmaması gibi dezavantajları bulunmaktadır. Bu dezavantajlar yapılan çalışmaların kapsamının dar olmasına, yalnızca belirli bir problemin çözümünde uygulanabilmesi gibi sonuçlara neden olmaktadır. Bu gibi çalışmalar bilimsel katkıdan çok belirli problemlerin geçici olarak çözülmesi amacı taşımaktadır. Özellikle bellek, işlem ve iletişim gereksinimleri açısından analiz içermeyen çalışmaların araştırmacılara kenar veya bulut bilişim tabanlı ver işleme mimarileri hakkında yeterli fayda sağlayamayacağı anlaşılmaktadır.

Geleneksel yöntemlerin dışında, derin öğrenme tabanlı yapay sinir ağı mimarileri, bölge tabanlı tespit etme ve sınıflandırma çalışmalarında yüksek başarı sağlamaları nedeniyle yaygın olarak kullanılmaktadır. Bu mimarilerin nesnelerin interneti çalışmaları açısından, çok fazla eğitim verisi gerektirmesi, eğitim aşamasının fazla zaman alması, model boyutlarının büyük olması gibi dezavantajları bulunmaktadır. Sınırlı işlem ve bellek imkânı olan nesnelere zor bir süreç olan eğitim aşamasının bellek, işlem, zaman gereksinimi ve oluşan büyük boyutlu cisim modellerinin iletim gereksinimi bu yaklaşımın uygulanmasını zorlaştırmaktadır. Bu yöntemlerin dağıtık mimari ile uygulanması için çalışmalar yapılmakta olmasına rağmen, problem farklı senaryolar şeklinde ele alınarak uygulanmamakta veya bellek, işlem ve veri iletişimi gibi konular dikkate alınmamaktadır.

Foto-kapan görüntüleri ile ilgili cisim tespit etme ve tanıma çalışmaları yeterli düzeyde olmasına rağmen, çalışmalarda insanlar tarafından el ile tespit edilmiş veri setleri kullanılarak gerçekleştirilmektedir. Bu çalışmalara ek olarak derin öğrenme tabanlı otomatik tespit ve tanıma uygulamaları yapılmaktadır. Yapılan çalışmalarda uygulanan yöntemler ve gerçekleştirilen diğer işlemler foto-kapan cihazlarında gerçekleştirilmemekte olup, bilgisayar ortamında çevrim dışı olarak kullanılmaktadır. Bu çalışmalarda, internet erişiminin etkili kullanılması, cihazlar arası iletişim, uzaktan kontrol gibi işlevler üzerinde durulmamaktadır. Bu çalışmalarda uygulanan bilgisayarlı görü ve görüntü işleme yöntemlerinin gerçek zamanlı veya dağıtık sistemlerde kullanılması üzerine analizler bulunmamaktadır. Bunlara ek olarak cisim içeren görüntülerin otomatik tespit edilmesi gibi işlevleri barındıran cihazların bulunmaması önemli bir fonksiyonel eksiklik olarak görülmektedir.

## 5. SONUÇ

Bu tezde, Nesnelerin İnterneti için kenar bilişim tabanlı akıllı görsel algılama analizi yapılmıştır. Çalışmada, görüntü tabanlı cisim tespit etme ve tanıma problemlerinin kenar bilişim ile çözülmesinde kullanılacak yöntemler, bellek, işlem ve iletişim gereksinimleri farklı senaryolar halinde incelenmiştir. Kenar ve bulut bilişim teknolojileri için veri işleme mimarilerinin incelendiği bu çalışma, yapılacak benzer çalışmalara rehberlik edilebilmesi amacıyla farklı çözüm yolları içeren beş farklı senaryo kullanılarak gerçekleştirilmiştir. Bu farklı senaryolar kenar ya da bulut bileşenleri arasında yöntemlerin ve işlemlerin paylaşılmasının bellek, işlem ve iletişim gereksinimi açısından karşılaştırmasını kapsamaktadır. Farklı yöntemler içeren senaryoların çeşitliliği, akıllı nesnelerin insandan bağımsız bir şekilde diğer nesnelerle haberleşerek ve yardımlaşarak problem çözebilmeleri fikrine altyapı oluşturmaktadır. Analizlerden elde edilen sonuçlar, problemin çözümünde gerçekleştirilecek işlemlerin sunucu ve akıllı nesneler arasında işlem, bellek ve veri iletişimi açısından etkili şekilde paylaşılması amacıyla kullanılabilir.

Tez çalışmasında, cisim tespit etme ve tanıma probleminde gerçekleştirilmesi gereken arka plan çıkarma, hareketli cisim tespit etme, öznitelik çıkarma, cisim konumlandırma, cisim tanıma gibi işlemler farklı senaryolar şeklinde uygulanmıştır. Bu senaryolarda kullanılan yöntemler ve algoritmalar, işlem, bellek ve başarı performansı açısından analiz edilmiştir. Cisim tespit etme ve tanıma probleminin zorluğu, karmaşıklığı ve farklı türde çözümler içermesi, internet nesnelerinin bellek, işlemci, batarya açısından sınırlı olan gömülü sistemler olması, bilgisayarlı görü ve görüntü tanıma çalışmalarının geniş uygulama alanlarına sahip olması tez çalışmasının temel motivasyonunu oluşturmaktadır.

Tez çalışmasında uygulanan kenar bilişim yaklaşımı, görüntülerde cisim tespit etme ve tanıma probleminin çözümünde kullanılan yöntemlerin bellek gereksiniminin ve işlem karmaşıklığının fazla olması, internet üzerinden veri iletiminin maliyetli olması ve

internet nesnesi sayısının hızla artması gibi nedenlerle, iletişim miktarının ve sunucu iş yükünün azaltılması amacıyla kullanılmaktadır. Bu nedenle çalışma ile hızla yaygınlaşan Nesnelerin İnterneti uygulamalarında kullanılacak kenar bilişim tabanlı çözümlerin, iletişim alt yapısına, sunucu iş yüküne, maliyete olan etkileri en aza indirilebileceği, işlem paylaşımının nesne ve sunucu arasında etkili yapılabileceği ve bu sayede Nesnelerin İnterneti çalışmalarının yaygınlaşabileceği anlaşılmaktadır.

Foto-kapanlar, Nesnelerin İnterneti için cisim tespit etme ve tanıma çalışmalarında kullanılması en uygun cihazlardır. Tez kapsamında, canlıların doğal ortamlarında görüntülenmesi amacıyla kullanılan foto-kapan işlevlerini yerine getirebilecek akıllı nesne tasarımı gerçekleştirilmiştir. Foto-kapanlarda olması istenen yeni işlevler literatürdeki çalışmalar incelenerek tespit edilmiş ve tasarımda kullanılmıştır. Foto-kapanın akıllı internet nesnesi olarak tasarlanması ile gerçek zamanlı olarak internet üzerinden erişilmesi, uzaktan kontrol edilebilmesi, kayıt, görüntü kalitesi, iletişim, ön işleme gibi işlevlerin düzenlenebilmesi, gerçek zamanlı veri aktarabilmesi gibi fonksiyonlar kullanılabilir. Bu fonksiyonların kullanımı hem yapılan çalışmaların başarısının arttıracağı, hem de foto-kapanların birçok farklı problemde kullanılabilmesinin önünü açacağı anlaşılmaktadır. Buna bağlı olarak foto-kapan yaklaşımının, akıllı nesnelerin geliştirilmesi, işlev ve protokollerinin belirlenmesi, diğer nesnelerle yardımlaşma senaryolarının geliştirilmesi için uygun olduğu görülmektedir.

Tez kapsamında, hareketli cisim tespitinde cisimlere ait farklı pozların daha az bellek ve işlem gereksinimi ile elde edilmesini ve örüntü tanıma yöntemi ile hareketli cisim çıktılarının iyileştirilmesini amaçlayan iki yeni yöntem önerilmiştir. Foto-kapanların görüntü kaydetme frekansının ayarlanabilen sabit bir değer olarak kullanılması, hareketli cisimlere ait uygun ve yeterli pozların yakalanamamasına veya benzer pozları içeren gereksiz görüntülerin kaydedilmesine neden olmaktadır. Çalışmada önerilen yöntemle hareketli cismin hareket hızına göre, görüntüdeki farklı konumlarda ve pozlarda cisim görüntüleri elde edilebilmektedir. Ayrıca foto-kapanların birim zamanda videolara kıyasla daha az görüntü kaydedilmesi, hareketli cisim tespitini, gürültü, yetersiz ve hatalı arka plan, gölge gibi problemler açısından zorlaştırmaktadır ve cisim

tespit başarısını düşürmektedir. Çalışmada bu problem tasarlanan nesne ile çözülmüş ve saniyede 32 görüntü karesi kaydedilebilen akıllı nesne ile yüksek tespit başarısı elde edilebildiği anlaşılmıştır.

Çalışmalardaki sonuçlar incelendiğinde, hareketli cisimlerin akıllı nesnelere tespit edilerek cisim bulunan görüntülerin sunucuya gönderilmesi ile yaklaşık %40, cisimlerin görüntülerini içeren görüntü parçalarının kırılarak iletilmesi ile yaklaşık %90 bellek ve veri iletim kazancı elde edilebileceği görülebilmektedir. Sınıflandırma işleminde kullanılacak özniteliklerin çıkarılma işleminin çeşitliliği, karmaşıklığı ve cismin tam olarak temsil edilememesi nedenleriyle, cisimleri içeren görüntü parçalarının sunucuya iletilmesi, öznitelik çıkarma ve sınıflandırma işlemlerinin sunucuda gerçekleştirilmesinin uygun senaryo olduğu anlaşılmaktadır. Ayrıca cisim tespitinin öznitelik eşleştirme, öznitelik tabanlı sınıflandırma veya derin öğrenme tabanlı yöntemler ile gerçekleştirildiği uygulamaların, yeni görüntüler ile cisim modelinin yeniden eğitilememesi, büyük boyutlu cisim modellerinin bütün akıllı nesnelere dağıtımının maliyetli olması, bellek ve işlem gereksinimlerinin fazla olması gibi dezavantajları bulunmaktadır. Bu nedenlerle bu yaklaşımların belirli problemin çözümde gerçekleştirilen özel çalışmalar haricinde uygun olmadığı düşünülmektedir.

**Çizelge 5.1.** Senaryoların bellek ve veri iletişim gereksinimi (%)

	<b>Gaussların Karışımı 2001</b>	<b>Gaussların Karışımı 2004-2006</b>	<b>Önerilen Yöntem</b>	<b>Değişen Gauss Ortalama</b>
Senaryo 1	100,000	100,000	100,000	100,000
Senaryo 2	54,857	70,560	71,124	71,406
Senaryo 3	0,246	0,378	0,333	0,332
Senaryo 4	0,013	0,016	0,017	0,019

**Çizelge 5.2.** Senaryoların akıllı nesne için çalışma zamanı ihtiyacı (mili saniye)

	<b>Gaussların Karışımı 2001</b>	<b>Gaussların Karışımı 2004- 2006</b>	<b>Önerilen Yöntem</b>	<b>Değişen Gauss Ortalama</b>
Senaryo 1	0,00	0,00	0,00	0,00
Senaryo 2	34,46	34,46	39,85	29,39
Senaryo 3	40,36	41,64	49,58	38,01
Senaryo 4	89,30	111,88	118,45	133,31

Çalışmada kullanılan senaryoların dışında tez kapsamında, cisimlerin hareket miktarına, hareket hızına ve görüntü içerisindeki konumlarına bağlı olarak belirli görüntülerin seçilebilmesi amacıyla yeni bir görüntü seçme algoritması önerilmiştir. Yaklaşımına ait sonuçlar incelendiğinde cisim veya hareket bulunmayan görüntüler elenebilmekte, cismin hareketine göre farklı pozlar kaydedilebilmekte ve cismin hareket hızına bağlı olarak yeterli sayıda görüntü alınabilmektedir. Sonuçlar incelendiğinde, cisim içermeyen veya cisimlere ait benzer pozların bulunduğu %85 ile %90 oranlarındaki gereksiz görüntülerin elenebilmektedir. Kullanılan yaklaşım ile cisme ait önemli görüntülerin seçilmesi ve diğerlerine göre daha az öneme sahip pozlar arası ara geçiş görüntülerinin elenmesi ile bellek, işlem ve iletişim gereksinimlerini azaltılabilmektedir.

Bilgisayarlı görü ve görüntü işleme, yaygın olarak çalışma imkânı olan ve kapsamlı araştırma alanlarıdır. Bu alanlara ait çalışmalarda, kullanılan verilerin çok boyutlu olması, farklı türde verilerin kullanılabilmesi, verilerin dönüşüm gerektirebilmesi, verilerin anlamlandırılması için gerekli işlemlerin matematiksel karmaşıklığının olması gibi zorluklar bulunmaktadır. Bu alanlarda görsel algılama işleminin gerçekleştirilebilmesi için literatürde farklı türde yaklaşımlar ve çok sayıda yöntem bulunmaktadır. Görüntülere veya görüntülerdeki cisimlere ait kenar, köşe, belirginlik veya bölge gibi özniteliklerin elde edilebilmesi için kullanılan çok sayıda dönüşüm algoritması bulunmaktadır. Bu algoritmalarından BRIEF ve ORB gibi ikili tanımlayıcı bilgisine sahip olanlar, cisimlerin sınıflandırılabilmesi amacıyla doğrudan

kullanılamayabilmektedir. SIFT, SURF gibi ikil olmayan bazı dönüşüm yöntemleri ise görsel kelime çantası yaklaşımı kullanılarak sınıflandırma yöntemlerinde kullanılabilir. Ayrıca dönüşüm algoritmalarının etkili olabilmesi için dönme, ölçekleme, kapanma, bulanıklık, gürültü, ışık şiddeti değişimi, 3D dönüşümü gibi zorluklara dayanıklı veya değişmez olması gerekmektedir. Bu dayanıklılık veya değişmezliklerin elde edilebilmesi amacıyla görüntüler üzerinde çok sayıda işlem gerçekleştirilmektedir. Bu nedenlerle kullanılan işlemlerin fazlalığı ve karmaşıklığı, işlem ve bellek gereksinimlerini arttırmakta olup, bu uygulamaların gerçek zamanlı yapılmasını zorlaştırmaktadır. Bu öznelik dönüşüm yöntemleri dışında, yapay sinir ağı tabanlı derin öğrenme mimarileri, görüntülerdeki derin özellikleri tespit etme amacıyla kullanılmaktadır. Karmaşık, zor ve uzun bir eğitim işlemi sonunda elde edilen sınıf modelleri kullanılarak, cisim tespiti ve tanıma işlemleri etkili bir şekilde gerçekleştirilebilmektedir. Bu mimarilerin performansını etkileyen en önemli özellikleri, çok sayıda veri ile çok sayıda döngü uygulanarak eğitiliyor olmasıdır. Ayrıca öznelik dönüşümü yöntemlerinde görüntüye ait farklı türde özelliği kullanılabilir. Bu özelliklerin çıkarılması ve dönüştürülmesi hem yapısal olarak farklılık içerebilmektedir, hem de farklı boyutlarda bellek ve işlem gereksinimi bulunmaktadır. Bu nedenle kullanılacak öznelik dönüşümü yönteminin problem için özel olarak seçilmesi gerektiği düşünülmektedir. Doğru öznelik dönüşümü yöntemi seçilmesinin, yapılacak çalışmanın performansını ve etkisini önemli ölçüde belirleyeceği anlaşılmaktadır.

Örüntü tanıma yöntemlerine ait sonuçlardan anlaşıldığı üzere, rastgele orman algoritmasının bütün öznelik dönüşüm yöntemleri için oluşturulmuş sınıf modellerinde, destek vektör makineleri ve k en yakın komşu algoritmalarına kıyasla daha dengeli bellek gereksinimine sahip olduğu anlaşılmaktadır. Sınıflandırma sonuçları incelendiğinde ise bu modellerin sınıflandırma işleminde oldukça başarılı olduğu görülmektedir. Bellek ve işlem gereksinimi ve öznelik türlerine göre performansı dikkate alındığında, rastgele orman algoritmasının kenar bilişim tabanlı uygulamalar için diğer algoritmalara kıyasla daha uygun olduğu düşünülmektedir.

Hareketli cisim tespiti yaklaşımı, öznitelik veya sınıflandırma tabanlı yöntemlerden farklı olarak, ardışık görüntülerdeki piksel değeri değişiminin analiz edilmesine dayanmaktadır. Bu yaklaşımda cisme ait herhangi örnek görüntü, model veya ön bilgi bulunmasına gerek olmamaktadır. Ancak bu yaklaşımda cisim tespitini zorlaştıracak arka plan karmaşıklığı, arka plan hareketi, ışık şiddeti değişimi, cisimlerin sayıca fazla olması ve cismin yavaş hareket etmesi gibi zorluklar bulunmaktadır. Bu yaklaşımda kullanılan yöntemlerin, öznitelik tabanlı yöntemlere oranla daha az işlem ve bellek gereksinimine sahip olması, yöntemlerin gömülü sistemlerde, kenar ürünlerinde veya nesnelere uygulanmasını kolaylaştırmaktadır.

### 5.1. Öneriler

Nesnelerin İnternetinin insanların günlük hayatına etkisi dikkate alındığında, uzaktan kontrol edilebilen, yazılım ve işlev olarak geliştirilebilen, birbirleri ile iletişim kurabilen akıllı nesnelerin tasarlanması beklenmektedir. Bilgisayarlı görü ve örüntü tanıma çalışmaları sayesinde, cisimlerde otomatik olarak tespit edilebilmesi ile Nesnelerin İnterneti çalışmalarının yaygınlaşacağı ve kapsamlarının genişleyeceği düşünülmektedir. Yaygınlaşan ve kapsamı genişleyen nesnelerin interneti çalışmalarının, insan hayatına pozitif etkilerde bulunacağı ve hayat konforunu yükselteceği anlaşılmaktadır. Nesnelerin internet üzerinden erişilebilme ve kontrol edilebilme imkânı olması sayesinde, birçok güvenlik riski ortadan kaldırılabilir, etkili algılama sistemlerinin kullanılması ile doğal afet ve kazaların en az zararla atlatılması sağlanabilecektir. Problem çözümünde uygulanan yöntemlerin geliştirilmesi ile literatüre yapılan katkıların, nesnelerin interneti uygulamaları ile akıllı nesnelerin geliştirilmesinde, nesnelerin interneti için gerekli protokol ve standardizasyon işlemlerinde ve Nesnelerin İnterneti uygulamalarının yaygınlaştırılmasında kullanılması uygun görülmektedir.

Tez kapsamında uygulanan senaryolar ile ilgili dikkat edilmesi gereken önemli hususlar, hareketli cisim tespitinde sunucuya iletilecek verilerin azaltılması ve hareketli cisme veya görüntü sahnelerine ait temsiliyetin arttırılması olarak ifade edilebilmektedir.



Sahne ile ilgili en fazla görüntünün bulunduğu senaryo, sahnede gerçekleşen olayla ve sahnedeki cisimlerle ilgili en fazla bilgiyi içermektedir. Bu durum, görüntülerden elde edilmek istenen sonuca veya çözülmeye çalışılan probleme göre değişiklik gösterebilmektedir. Dördüncü ve beşinci senaryolarda daha az bellek ve iletişim gereksinimi olmasına rağmen, daha fazla anlam ve temsiliyet kaybı gerçekleşmektedir. Hareketli cisimlerin sınıflandırılması için kullanılacak özniteliklerin kullanımı, belirli ve az sayıdaki sınıf bulunan çalışmalar için uygun olmasına rağmen, aktivite tanıma, cisme ait detaylı ve ileri seviyeli sınıflandırma işlemleri içeren problemler için yeterli olamayabilmektedir. Bu nedenle akıllı nesnelerin, sabit işlemler ve özellikler gerçekleştirilmesi yerine, bu işlemlerin kontrol edilebilir, ayarlanabilir ve güncellenebilir olması önerilmektedir.

Tez çalışması kapsamında kaydedilen görüntülerin, foto-kapan cihazlarında kaydedilenlere kıyasla fazla olması, işlenecek verinin, bellek, işlem ve iletişim gereksinimlerinin artmasına neden olmaktadır. Verinin hangi kısmının nerede işleneceği problemi kenar bilişimin önemini arttırmaktadır. Bunlara ek olarak akıllı nesnelere beklenen iş birliği yapma, ağa hizmet verme, yardım alma gibi işlevlerin gerçekleştirilebilmesi için problemlere özgü standartlar ve protokoller geliştirilmesi gerekmektedir.

Tez çalışmasında akıllı nesnelerin enerji tüketimleri analiz edilmemiştir. Akıllı nesne olarak kullanılacak gömülü sistemler, sınırlı enerji kaynağı imkânına sahip olabilmektedir. Görsel algılama için kullanılacak yöntemlerin işlem gereksiniminin fazla olması nedeniyle, akıllı nesne ve sunucu arasında iş paylaşımı yapılması hususuna dikkat edilmesi gerektiği anlaşılmaktadır.

**KAYNAKLAR**

- Abdelli, A., and Choi, H.-J. (2017). *A four-frames differencing technique for moving objects detection in wide area surveillance*. Paper presented at the 2017 IEEE International Conference on Big Data and Smart Computing (BigComp).
- Akman, M., Genç, Y., and Ankarali, H. (2011). Random forests yöntemi ve sağlık alanında bir uygulama. *Turkiye Klinikleri Journal of Biostatistics*, 3(1), 36-48.
- Akpınar, H. (2014). *Data: veri madenciliği veri analizi*: Papatya Yayıncılık Eğitim.
- Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., and Ayyash, M. (2015). Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE communications surveys and tutorials*, 17(4), 2347-2376.
- Alex, D. S., Wahi, A. J. J. o. T., and Technology, A. I. (2014). BSFD: Background Subtraction Frame Difference Algorithm For Moving Object Detection And Extraction. 60(3).
- Amin, A. H. M., Ahmad, N. M., and Ali, A. M. M. (2016). *Decentralized face recognition scheme for distributed video surveillance in IoT-cloud infrastructure*. Paper presented at the 2016 IEEE Region 10 Symposium (TENSymp).
- Andavarapu, N., and Vatsavayi, V. K. (2017). Wild-Animal Recognition in Agriculture Farms Using W-COHOG for Agro-Security. *International Journal of Computational Intelligence Research*, 13(9), 2247-2257.
- Anonymous, 2018a. <http://videonet.ece.missouri.edu/cameratrap/> 12.01.2018
- Anonymous, 2018b. <http://homepages.inf.ed.ac.uk/rbf/CAVIAR/> 22.04.2018
- Anonymous, 2019a. <http://perception.i2r.a-star.edu.sg/> 07.03.2019
- Anonymous, 2019b. <https://drive.google.com/open?id=1YzFYiWvXfcp4IkbQJBLInugKbazV3ROS>
- Atzori, L., Iera, A., and Morabito, G. (2010). The internet of things: A survey. *Computer networks*, 54(15), 2787-2805.
- Atzori, L., Iera, A., and Morabito, G. (2014). From "smart objects" to "social objects": The next evolutionary step of the internet of things. *IEEE Communications Magazine*, 52(1), 97-105.
- Bay, H., Tuytelaars, T., and Van Gool, L. (2006). *Surf: Speeded up robust features*. Paper presented at the European conference on computer vision.
- Bin, S., Yuan, L., and Xiaoyi, W. (2010). *Research on data mining models for the internet of things*. Paper presented at the 2010 International Conference on Image Analysis and Signal Processing.
- Bonomi, F., Milito, R., Zhu, J., and Addepalli, S. (2012). *Fog computing and its role in the internet of things*. Paper presented at the Proceedings of the first edition of the MCC workshop on Mobile cloud computing.
- Breiman, L. (2001). Random forests. *Machine learning*, 45(1), 5-32.
- Breiman, L., Friedman, J., Olshen, R., and Stone, C. (1984). Classification and regression trees. Wadsworth and Brooks. *Cole Statistics/Probability Series*.
- Brili, K.-., and Helen, M.-. (2014). *Analysis and extension of IoT ontologies towards autonomous reasoning of things*. Πανεπιστήμιο Πειραιώς,

- Buehler, P., Carroll, B., Bhatia, A., Gupta, V., and Lee, D. E. (2019). An automated program to find animals and crop photographs for individual recognition. *Ecological Informatics*, 50, 191-196.
- Calonder, M., Lepetit, V., Strecha, C., and Fua, P. (2010). *Brief: Binary robust independent elementary features*. Paper presented at the European conference on computer vision.
- Canny, J. (1987). A computational approach to edge detection. In *Readings in computer vision* (pp. 184-203): Elsevier.
- Chen, F., Deng, P., Wan, J., Zhang, D., Vasilakos, A. V., and Rong, X. (2015). Data mining for the internet of things: literature review and challenges. *International Journal of Distributed Sensor Networks*, 11(8), 431047.
- Cortes, C., and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3), 273-297.
- Cucchiara, R., Grana, C., Piccardi, M., and Prati, A. (2003). Detecting moving objects, ghosts, and shadows in video streams. *IEEE transactions on pattern analysis and machine intelligence*, 25(10), 1337-1342.
- Çalışkan, S. K., and Soğukpınar, İ. (2008). KxKNN: K-Means ve K En Yakın Komşu Yöntemleri İle Ağlarda Nüfuz Tespiti. *EMO Yayınları*, 120-124.
- Dalal, N., and Triggs, B. (2005). *Histograms of oriented gradients for human detection*. Paper presented at the international Conference on computer vision and Pattern Recognition (CVPR'05).
- DeArruda, M. d. S., Spadon, G., Rodrigues, J. F., Gonçalves, W. N., and Machado, B. B. (2018). *Recognition of Endangered Pantanal Animal Species using Deep Learning Methods*. Paper presented at the 2018 International Joint Conference on Neural Networks (IJCNN).
- Dhake, K., and Dalu, S. (2018). An Overview of Methodologies for Moving Object Detection.
- Dillencourt, M. B., Samet, H., and Tamminen, M. (1992). A general approach to connected-component labeling for arbitrary image representations. *Journal of the ACM (JACM)*, 39(2), 253-280.
- Erkin Tekeli, M. Ç., Aytül Erçil. (2006). Yerel İkili Örüntü (LBP) Ve Önsel Şekil Bilgisi Tabanlı Bir Desen Bölütleme Metodu.
- Gang, L., Shangkun, N., Yugan, Y., Guanglei, W., and Siguo, Z. (2013). *An improved moving objects detection algorithm*. Paper presented at the 2013 International Conference on Wavelet Analysis and Pattern Recognition.
- Girod, B., Chandrasekhar, V., Chen, D. M., Cheung, N.-M., Grzeszczuk, R., Reznik, Y., Vedantham, R. (2011). Mobile visual search. *IEEE signal processing magazine*, 28(4), 61-76.
- Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). *Rich feature hierarchies for accurate object detection and semantic segmentation*. Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.
- Glover- Kapfer, P., Soto- Navarro, C. A., and Wearn, O. R. (2019). Camera- trapping version 3.0: current constraints and future priorities for development. *Remote Sensing in Ecology and Conservation*.
- Gray, P. C., Fleishman, A. B., Klein, D. J., McKown, M. W., Bézy, V. S., Lohmann, K. J., and Johnston, D. W. (2019). A convolutional neural network for detecting sea turtles in drone imagery. *Methods in Ecology and Evolution*, 10(3), 345-355.

- Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Cai, J. (2018). Recent advances in convolutional neural networks. *Pattern Recognition*, 77, 354-377.
- Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S., and Lew, M. S. (2016). Deep learning for visual understanding: A review. *Neurocomputing*, 187, 27-48.
- Han, X., Gao, Y., Lu, Z., Zhang, Z., and Niu, D. (2015). *Research on moving object detection algorithm based on improved three frame difference method and optical flow*. Paper presented at the 2015 Fifth International Conference on Instrumentation and Measurement, Computer, Communication and Control (IMCCC).
- Hanbay, K., and Hüseyin, Ü. (2017). Nesne tespiti ve takip metotları: Kapsamlı bir derleme. *Türk Doğa ve Fen Dergisi*, 6(2), 40-49.
- Harris, C. G., and Stephens, M. (1988). *A combined corner and edge detector*. Paper presented at the Alvey vision conference.
- He, K., Zhang, X., Ren, S., and Sun, J. (2015). Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(9), 1904-1916.
- He, Z., Kays, R., Zhang, Z., Ning, G., Huang, C., Han, T. X., . . . McShea, W. (2016). Visual informatics tools for supporting large-scale collaborative wildlife monitoring with citizen scientists. *IEEE Circuits and Systems Magazine*, 16(1), 73-86.
- Hoebeke, L., Stock, M., Van Hoey, S., Casaer, J., and De Baets, B. (2018). *Automated recognition of people and identification of animal species in camera trap images*. Paper presented at the ICEI 2018: 10th International Conference on Ecological Informatics-Translating Ecological Data into Knowledge and Decisions in a Rapidly Changing World.
- Hu, W., Tan, T., Wang, L., and Maybank, S. (2004). A survey on visual surveillance of object motion and behaviors. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 34(3), 334-352.
- Huang, T., Yang, G., and Tang, G. (1979). A fast two-dimensional median filtering algorithm. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 27(1), 13-18.
- Huang, W., Liu, Z., and Pan, W. (2007). *The precise recognition of moving object in complex background*. Paper presented at the Third International Conference on Natural Computation (ICNC 2007).
- Hum, Y. C., Lai, K. W., and Mohamad Salim, M. I. (2014). Multiobjectives bi-histogram equalization for image contrast enhancement. *Complexity*, 20(2), 22-36.
- Jain, R., Kasturi, R., and Schunck, B. G. (1995). *Machine vision* (Vol. 5): McGraw-Hill New York.
- Jain, R., and Nagel, H.-H. (1979). On the analysis of accumulative difference pictures from image sequences of real world scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*(2), 206-214.
- Jain, S. K., and Rajankar, S. O. (2017). Real-Time Object Detection and Recognition Using Internet of Things Paradigm. *International Journal of Image, Graphics and Signal Processing*, 9(1), 18.
- Karakuş, P., and Karabörk, H. (2014). Surf Algoritması Kullanılarak Uzaktan Algılama Görüntülerinin Geometrik Kaydı.

- Karami, E., Prasad, S., and Shehata, M. (2017). Image matching using SIFT, SURF, BRIEF and ORB: performance comparison for distorted images. *arXiv preprint arXiv:1710.02726*.
- Karasulu, B., and Korukoglu, S. (2013). *Performance Evaluation Software: Moving Object Detection and Tracking in Videos*: Springer Science and Business Media.
- Kawano, Y., and Yanai, K. (2013). *Real-time mobile food recognition system*. Paper presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops.
- Kays, R., Tilak, S., Kranstauber, B., Jansen, P. A., Carbone, C., Rowcliffe, M. J., . . . He, Z. (2010). Monitoring wild animal communities with arrays of motion sensitive camera traps. *arXiv preprint arXiv:1009.5718*.
- Khorrani, P., Wang, J., and Huang, T. (2012). *Multiple animal species detection using robust principal component analysis and large displacement optical flow*. Paper presented at the Proceedings of the 21st International Conference on Pattern Recognition (ICPR), Workshop on Visual Observation and Analysis of Animal and Insect Behavior.
- Kortuem, G., Kawsar, F., Sundramoorthy, V., and Fitton, D. (2009). Smart objects as building blocks for the internet of things. *IEEE Internet Computing*, 14(1), 44-51.
- Kulchandani, J. S., and Dangarwala, K. (2015). *Moving object detection: Review of recent research trends*. Paper presented at the 2015 International Conference on Pervasive Computing (ICPC).
- LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324.
- Li, S., Da Xu, L., and Zhao, S. (2015). The internet of things: a survey. *Information Systems Frontiers*, 17(2), 243-259.
- Lin, J., Yu, W., Zhang, N., Yang, X., Zhang, H., and Zhao, W. (2017). A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications. *IEEE Internet of Things Journal*, 4(5), 1125-1142.
- Lin, K.-H., Khorrani, P., Wang, J., Hasegawa-Johnson, M., and Huang, T. S. (2014). *Foreground object detection in highly dynamic scenes using saliency*. Paper presented at the 2014 IEEE International Conference on Image Processing (ICIP).
- Lindeberg, T. (1993). Detecting salient blob-like image structures and their scales with a scale-space primal sketch: A method for focus-of-attention. *International journal of computer vision*, 11(3), 283-318.
- Lindeberg, T. (1998). Feature detection with automatic scale selection. *International journal of computer vision*, 30(2), 79-116.
- Lindeberg, T. (2007). Scale- space. *Wiley Encyclopedia of Computer Science and Engineering*, 2495-2504.
- Lisin, D. A., Mattar, M. A., Blaschko, M. B., Learned-Miller, E. G., and Benfield, M. C. (2005). *Combining local and global image features for object class recognition*. Paper presented at the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)-Workshops.
- Lowe, D. G. (1999). *Object recognition from local scale-invariant features*. Paper presented at the iccv.

- Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2), 91-110.
- Mainetti, L., Patrono, L., and Vilei, A. (2011). *Evolution of wireless sensor networks towards the internet of things: A survey*. Paper presented at the SoftCOM 2011, 19th International Conference on Software, Telecommunications and Computer Networks.
- Meek, P., Ballard, G., Claridge, A., Kays, R., Moseby, K., O'brien, T., . . . Tobler, M. (2014). Recommended guiding principles for reporting on camera trapping research. *Biodiversity and conservation*, 23(9), 2321-2343.
- Meek, P. D., Vernes, K., and Falzon, G. (2013). On the reliability of expert identification of small-medium sized mammals from camera trap photos. *Wildlife Biology in Practice*, 9(2), 1-19.
- Muja, M., and Lowe, D. G. (2009). Fast approximate nearest neighbors with automatic algorithm configuration. *VISAPP (1)*, 2(331-340), 2.
- Murphy, K., Torralba, A., Eaton, D., and Freeman, W. (2006). Object detection and localization using local and global features. In *Toward Category-Level Object Recognition* (pp. 382-400): Springer.
- Ngoc, T. N. (2014). *An efficient LBP-based descriptor for real-time object detection*. Paper presented at the the 2014 Seventh IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA).
- Nguyen, H., Maclagan, S. J., Nguyen, T. D., Nguyen, T., Flemons, P., Andrews, K., . . . Phung, D. (2017). *Animal recognition and identification with deep convolutional neural networks for automated wildlife monitoring*. Paper presented at the 2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA).
- Norouzzadeh, M. S., Nguyen, A., Kosmala, M., Swanson, A., Palmer, M. S., Packer, C., and Clune, J. (2018). Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning. *Proceedings of the National Academy of Sciences*, 115(25), E5716-E5725.
- Nurhadiyatna, A., Jatmiko, W., Hardjono, B., Wibisono, A., Sina, I., and Mursanto, P. (2013). *Background subtraction using gaussian mixture model enhanced by hole filling algorithm (gmmhf)*. Paper presented at the 2013 IEEE International Conference on Systems, Man, and Cybernetics.
- Ojala, T., Pietikäinen, M., and Mäenpää, T. (2000). *Gray scale and rotation invariant texture classification with local binary patterns*. Paper presented at the European Conference on Computer Vision.
- Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE Transactions on systems, man, and cybernetics*, 9(1), 62-66.
- Ouyang, W., Luo, P., Zeng, X., Qiu, S., Tian, Y., Li, H., . . . Qian, C. (2014). Deepid-net: multi-stage and deformable deep convolutional neural networks for object detection. *arXiv preprint arXiv:1409.3505*.
- Pawar, M. P., and Belagali, P. (2010). Image Retrieval Technique Using Local Binary Pattern (LBP). *IEEE Trans. Image Process*, 19(6), 1657-1663.
- Piccardi, M. (2004). *Background subtraction techniques: a review*. Paper presented at the 2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No. 04CH37583).

- Qin, Y., Sheng, Q. Z., Falkner, N. J., Dustdar, S., Wang, H., and Vasilakos, A. V. (2016). When things matter: A survey on data-centric internet of things. *Journal of Network and Computer Applications*, 64, 137-153.
- Rani, R., Kumar, R., and Singh, A. P. (2018). An empirical evaluation of translational and rotational invariance of descriptors and the classification of flower dataset. *Pattern Analysis and Applications*, 21(1), 1-18.
- Rovero, F., Zimmermann, F., Berzi, D., and Meek, P. (2013). " Which camera trap type and how many do I need?" A review of camera features and study designs for a range of wildlife research applications. *Hystrix*, 24(2).
- Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF.
- Samet, H., and Tamminen, M. (1988). Efficient component labeling of images of arbitrary dimension represented by linear bintrees. *IEEE transactions on pattern analysis and machine intelligence*, 10(4), 579-586.
- Schindelin, J., Arganda-Carreras, I., Frise, E., Kaynig, V., Longair, M., Pietzsch, T., . . . Schmid, B. (2012). Fiji: an open-source platform for biological-image analysis. *Nature methods*, 9(7), 676.
- Shapiro, L. G. (1996). Connected component labeling and adjacency graph construction. In *Machine Intelligence and Pattern Recognition* (Vol. 19, pp. 1-30): Elsevier.
- Shapiro, L. G., and Stockman, G. (2001). *Computer Vision: Theory and Applications*. 2001. In: Prentice Hall.
- Shi, H. (2007). *Best-first decision tree learning*. The University of Waikato,
- Singla, N. (2014). Motion detection based on frame difference method. *International Journal of Information and Computation Technology*, 4(15), 1559-1565.
- Smith, J. R., and Chang, S.-F. (1995). *Single color extraction and image query*. Paper presented at the Proceedings., International Conference on Image Processing.
- Song, Z., Chen, Q., Huang, Z., Hua, Y., and Yan, S. (2011). *Contextualizing object detection and classification*. Paper presented at the CVPR 2011.
- Suzuki, S. (1985). Topological structural analysis of digitized binary images by border following. *Computer vision, graphics, and image processing*, 30(1), 32-46.
- Swain, M., and Ballard, D. (1991). Color Indexing International Journal of Computer Vision 7.
- Swan, M. (2012). Sensor mania! the internet of things, wearable computing, objective metrics, and the quantified self 2.0. *Journal of Sensor and Actuator networks*, 1(3), 217-253.
- Şimşek, E., Özyer, B., Özyer, G. T., and Bayındır, L. (2018a). *Animal Detection in Camera-Trap Images*. Paper presented at the ITESDES 2018 Giresun - 1. Uluslararası Teknoloji Bilimleri ve Tasarım Sempozyumu.
- Şimşek, E., Özyer, B., Bayındır, L., & Özyer, G. T. (2018b, May). Human-animal recognition in camera trap images. In 2018 26th Signal Processing and Communications Applications Conference (SIU) (pp. 1-4). IEEE.
- Taşcı, E., and Onan, A. (2016). K-en yakın komşu algoritması parametrelerinin sınıflandırma performansı üzerine etkisinin incelenmesi. *Akademik Bilişim*.
- Thapa, G., Sharma, K., and Ghose, M. (2014). Moving object detection and segmentation using frame differencing and summing technique. 975, 8887.

- Tompson, J., Goroshin, R., Jain, A., LeCun, Y., and Bregler, C. (2015). *Efficient object localization using convolutional networks*. Paper presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
- Tsai, C.-W., Lai, C.-F., Chiang, M.-C., and Yang, L. T. (2014). Data mining for internet of things: A survey. *IEEE communications surveys and tutorials*, 16(1), 77-97.
- Tuytelaars, T., and Mikolajczyk, K. (2008). Local invariant feature detectors: a survey. *Foundations and trends® in computer graphics and vision*, 3(3), 177-280.
- Tyagi, V. K. (2010). *Object recognition on android mobile platform using speeded up robust features*: Florida Atlantic University.
- Vasseur, J.-P., and Dunkels, A. (2010). *Interconnecting smart objects with ip: The next internet*: Morgan Kaufmann.
- Venkataramani, S., Roy, K., and Raghunathan, A. (2016). *Efficient embedded learning for IoT devices*. Paper presented at the 2016 21st Asia and South Pacific Design Automation Conference (ASP-DAC).
- Vermesan, O., and Friess, P. (2013). *Internet of things: converging technologies for smart environments and integrated ecosystems*: River Publishers.
- Wang, B. (2014). *Automatic Animal Species Identification Based on Camera Trapping Data*. University of Alberta,
- Wang, D., Chen, D., Song, B., Guizani, N., Yu, X., and Du, X. (2018). From IoT to 5G I-IoT: The Next Generation IoT-Based Intelligent Algorithms and 5G Technologies. *IEEE Communications Magazine*, 56(10), 114-120.
- Whitmore, A., Agarwal, A., and Da Xu, L. (2015). The Internet of Things—A survey of topics and trends. *Information Systems Frontiers*, 17(2), 261-274.
- Willi, M., Pitman, R. T., Cardoso, A. W., Locke, C., Swanson, A., Boyer, A., . . . Fortson, L. (2019). Identifying animal species in camera trap images using deep learning and citizen science. *Methods in Ecology and Evolution*, 10(1), 80-91.
- Wojcik, G. M., and Kaminski, W. A. J. N. (2004). Liquid state machine built of Hodgkin–Huxley neurons and pattern recognition. 58, 245-251.
- Wren, C. R., Azarbayejani, A., Darrell, T., and Pentland, A. P. (1997). Pfinder: Real-time tracking of the human body. *IEEE transactions on pattern analysis and machine intelligence*, 19(7), 780-785.
- Xiao, J., Cheng, H., Sawhney, H., and Han, F. (2010). *Vehicle detection and tracking in wide field-of-view aerial video*. Paper presented at the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition.
- Yi, S., Li, C., and Li, Q. (2015). *A survey of fog computing: concepts, applications and issues*. Paper presented at the Proceedings of the 2015 workshop on mobile big data.
- Yousif, H., Yuan, J., Kays, R., and He, Z. (2019). Animal Scanner: Software for classifying humans, animals, and empty frames in camera trap images. *Ecology and Evolution*.
- Yu, X., Wang, J., Kays, R., Jansen, P. A., Wang, T., and Huang, T. (2013). Automated identification of animal species in camera trap images. *EURASIP Journal on Image and Video Processing*, 2013(1), 52.
- Yun, M., and Yuxin, B. (2010). *Research on the architecture and key technology of Internet of Things (IoT) applied on smart grid*. Paper presented at the 2010 International Conference on Advances in Energy Engineering.



- Zeiler, M. D., and Fergus, R. (2013). Stochastic pooling for regularization of deep convolutional neural networks. *arXiv preprint arXiv:1301.3557*.
- Zhang, H., and Zhang, H. (2013). *A moving target detection algorithm based on dynamic scenes*. Paper presented at the 2013 8th International Conference on Computer Science and Education.
- Zhang, J., Marszałek, M., Lazebnik, S., and Schmid, C. (2007). Local features and kernels for classification of texture and object categories: A comprehensive study. *International journal of computer vision*, 73(2), 213-238.
- Zhang, Y., Wang, X., and Qu, B. J. P. E. (2012). Three-frame difference algorithm research based on mathematical morphology. 29, 2705-2709.
- Zhang, Z., He, Z., Cao, G., and Cao, W. (2016). Animal detection from highly cluttered natural scenes using spatiotemporal object region proposals and patch verification. *IEEE Transactions on Multimedia*, 18(10), 2079-2092.
- Zitnick, C. L., and Dollár, P. (2014). *Edge boxes: Locating object proposals from edges*. Paper presented at the European conference on computer vision.
- Zorzi, M., Gluhak, A., Lange, S., and Bassi, A. (2010). From today's intranet of things to a future internet of things: a wireless-and mobility-related view. *IEEE Wireless communications*, 17(6), 44-51.

## ÖZGEÇMİŞ

1986 yılı Ankara Altındağ'da doğmuş, ilk ve orta öğrenimini Ankarada tamamlamıştır. Orta öğrenimini 2000-2004 yılları arasında Anafartalar Anadolu Ticaret Meslek Lisesinde bilgi işlem alanında, yüksek öğrenimini ise 2012-2015 yılları arasında Atatürk Üniversitesi Bilgisayar Mühendisliği bölümünde tamamlamıştır. 2007-2017 yılları arasında Atatürk Üniversitesinde idari personel olarak görev yapmış olup, 2006-2007 yılları arasında 1 yıl Bilgisayar Öğretmenliği, 2010-2012 yılları arasında 2 yıl Fizik eğitimi almıştır. 2015 yılında Bilgisayar Mühendisliği Anabilim Dalında Yüksek Lisans eğitimine başlamış olup, evli ve iki erkek çocuk babasıdır.