

**T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**



**ÇOKLU VIDEO GÖRÜNTÜLERİ ÜZERİNDE AKILLI HEDEF
TAKİBİ**

Sevinç AY

Doktora Tezi

YAZILIM MÜHENDİSLİĞİ ANABİLİM DALI

Yazılım Bilim Dalı

TEMMUZ 2023

T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Yazılım Mühendisliği Anabilim Dalı

Doktora Tezi

ÇOKLU VIDEO GÖRÜNTÜLERİ ÜZERİNDE AKILLI HEDEF TAKİBİ

Tez Yazarı

Sevinç AY

Danışman

Prof. Dr. Murat KARABATAK

TEMMUZ 2023

ELAZIĞ

T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Yazılım Mühendisliği Anabilim Dalı

Doktora Tezi

Başlığı:	Çoklu Video Görüntüleri Üzerinde Akıllı Hedef Takibi
Yazarı:	Sevinç AY
İlk Teslim Tarihi:	03.05.2023
Savunma Tarihi:	07.07.2023

TEZ ONAYI

Fırat Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına göre hazırlanan bu tez aşağıda imzaları bulunan jüri üyeleri tarafından değerlendirilmiş ve akademik dinleyicilere açık yapılan savunma sonucunda OYBİRLİĞİ ile kabul edilmiştir.

Danışman:	Prof. Dr. Murat KARABATAK Fırat Üniversitesi, Teknoloji Fakültesi	İmza Onayladım
Başkan:	Prof. Dr. Engin AVCI Fırat Üniversitesi, Teknoloji Fakültesi	Onayladım
Üye:	Doç. Dr. Özal YILDIRIM Fırat Üniversitesi, Teknoloji Fakültesi	Onayladım
Üye:	Prof. Dr. Muhammed Fatih TALU İnönü Üniversitesi, Mühendislik Fakültesi	Onayladım
Üye:	Dr. Öğr. Üyesi Yusuf ÇELİK Munzur Üniversitesi, Mühendislik Fakültesi	Onayladım

Bu tez, Enstitü Yönetim Kurulunun/...../20..... tarihli toplantısında tescillenmiştir.

İmza

Prof. Dr. Burhan ERGEN
Enstitü Müdürü

BEYAN

Fırat Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırladığım “Çoklu Video Görüntüleri Üzerinde Akıllı Hedef Takibi ” Başlıklı Doktora Tezimin içindeki bütün bilgilerin doğru olduğunu, bilgilerin üretilmesi ve sunulmasında bilimsel etik kurallarına uygun davrandığımı, kullandığım bütün kaynakları atıf yaparak belirttiğimi, maddi ve manevi desteği olan tüm kurum/kuruluş ve kişileri belirttiğimi, burada sunduğum veri ve bilgileri unvan almak amacıyla daha önce hiçbir şekilde kullanmadığımı beyan ederim.

07.07.2023

Sevinç AY



ÖNSÖZ

Kameralı gözetim sistemlerinin başta güvenlik olmak üzere birçok uygulamada kullanımı gün geçtikçe yaygınlaşmaktadır. Özellikle trafik uygulamalarında kullanılan bu sistemler üzerinde güvenliğin sağlanması amacıyla şüpheli bir hedefin takibi birçok problemin çözümüne yardımcı olacaktır. Bu tez çalışması kapsamında trafikte kullanılan kameralı gözetim sistemlerinden elde edilen video görüntülerinden hedef tespiti ve takibi yapılmaktadır. Gözetim sistemlerinin otomatik denetlenmesini ve hedefin tespit edilmesini sağlayan bir yöntem sunularak yaşanan birçok problemin önüne geçmek amaçlanmıştır.

Çalışmalarım süresince, değerli görüş ve katkılarıyla beni yönlendiren, fikirlerinden, bilim insanı kişiliğinden ve tecrübesinden çok şey öğrendiğim değerli danışman hocam Sayın Prof. Dr. Murat KARABATAK'a teşekkürü borç bilirim. Tez İzleme Komitemde bulunan ve çalışmam boyunca bilimsel desteklerinden faydalandığım Sayın Prof. Dr. Engin AVCI'ya, bilgilerini, değerli zamanını ve yardımlarını esirgemeyen Sayın Doç. Dr. Özal YILDIRIM'a, desteğini her daim yanımda hissettiğim değerli arkadaşım Sayın Doç. Dr. Derya AVCI'ya, fikirleriyle bana yol gösteren ve bilimsel çalışma disiplini konusunda kendisinden çok şey öğrendiğim Sayın Doç. Dr. Songül KARABATAK'a ve saygıdeğer jüri üyelerine çok teşekkür ederim. Tez çalışma yolculuğuna başlamamda çok büyük katkıları olan, maddi ve manevi desteklerini hiçbir zaman esirgemeyen, evlatları olmaktan gurur duyduğum ve insanlık, eğitimcilik konusunda örnek aldığım sevgili anne ve babama, motivasyon desteğiyle tezimi bitirmemde katkısı büyük olan sevgili kardeşime, beni destekleyen ve her durumda yanımda olduğunu hissettiren sevgili eşime, tezimin son aşamalarında tüm zorlukları birlikte yaşadığım sevgili oğluma çok teşekkür ederim.

Sevinç AY
ELAZIĞ, 2023

İÇİNDEKİLER

Sayfa

ÖNSÖZ	iv
İÇİNDEKİLER	v
ÖZET	vii
ABSTRACT	viii
ŞEKİLLER LİSTESİ	ix
TABLolar LİSTESİ	xi
SİMGELER VE KISALTMALAR	xii
1. GİRİŞ	1
1.1. Problem Durumu	2
1.2. Tezin Amacı	3
1.3. Literatür İncelemesi	4
1.3.1. Nesne Takibi Çalışmaları	5
1.3.2. Nesne Tespiti Çalışmaları	6
1.4. Çalışmanın Katkıları	9
1.5. Tezin Organizasyonu	9
2. TEORİK ARKA PLAN	10
2.1. Nesne Algılama	10
2.1.1. Video İşleme, Nesne Takibi ve Nesne Tespiti	10
2.1.2. Videolarda Nesne Tespiti ve Segmentasyon	11
2.1.3. Hareketli Nesne Algılama	12
2.1.4. Hareketli Nesne Algılama Uygulamaları	13
2.2. Ön Plan ve Arka Plan	13
2.2.1. Ön Plan Tespitine Karşı Arka Plan Çıkarma	15
2.2.2. Arka Plan Çıkarımı Yaklaşımı	15
2.2.3. Temsili Örneklerle Dayalı Arka Plan Çıkarımı Yöntemi	18
2.2.4. Kod Çizelgesi Tabanlı Arka Plan Çıkarımı Yöntemi	19
2.2.5. Diğer Arka Plan Çıkarım Algoritmaları	20
2.2.6. Olasılık Tahminine Dayalı Arka Plan Çıkarımı	20
2.2.7. Gauss Karışım Modeli	20
2.3. Nesne Takibi	24
2.3.1. Özelliğe Dayalı İzleme	25
2.3.2. Kontur Tabanlı İzleme	25
2.3.3. Model Bazlı İzleme	25
2.3.4. Kalman Filtreleme	25
2.3.5. Nokta Takibi	26
2.3.6. Çekirdek Takibi	27
2.3.7. Silüet Takibi	27
2.3.8. Video Görüntülerinde Hedef Takibi	28
2.3.9. Blob (İkili Büyük Nesne, Binary Large Object) Analizi	31
2.4. Derin Öğrenme Yöntemleri Kullanarak Nesne Tespiti	32
2.4.1. Evrişimli Sinir Ağları (ESA) Tanımı	33
2.4.2. Bölge Tabanlı Evrişimli Ağ (B-ESA)	34

2.4.3. Hızlı Bölge Tabanlı Evrişimli Sinir Ağı (Hızlı B-ESA).....	35
2.4.4. Daha Hızlı Bölge Tabanlı Evrişimsel Sinir Ağı (Daha Hızlı B-ESA)	36
2.4.5. YOLO (Yalnızca Bir Kez Bakarsınız).....	39
2.4.6. YOLO sürüm 1	41
2.4.7. YOLO sürüm 2	42
2.5. Derin Öğrenme Modelleri	43
2.5.1. ImageNet	43
2.5.2. AlexNet.....	44
2.5.3. VGGNet.....	46
2.5.4. GoogleNet.....	47
2.5.5. ResNet	50
2.5.6. MobileNet.....	53
3. ÇOKLU VIDEO GÖRÜNTÜLERİNDE ARAÇ TAKİP VE TESPİT SİSTEMİ	56
3.1. Çalışmaya Genel Bakış	56
3.2. Araç Takibi Algoritmasının Tanıtılması	57
3.3. Hedef Aracın Görüntüleneceği Kameraların Tespiti	62
3.3.1. Araç Yönünün Belirlenmesi	63
3.3.2. Aracın Gidiş Yönünde Bulunan Kameraların Tespiti	67
3.3.3. Hedef Aracın Aranacağı Yeni Kameranın Tespit Edilmesi	71
3.3.4. Haversine Formülü	71
3.3.5. Hedef Aracın Yeni Kameraya Göre Yönünün ve Aranacağı Sürenin Bulunması	72
3.4. Araç Tespiti ve Eğitim Süreçleri.....	74
3.5. Deneysel Sonuçlar	74
3.5.1. Eğitim Platformu ve Parametre Ayarları	74
3.5.2. Veri toplama	75
3.5.3. Değerlendirme Metrikleri.....	76
3.5.4. Eğitim Süreci	77
3.5.5. Eğitim sürecinde kayıp fonksiyonu	81
4. SONUÇLAR.....	87
KAYNAKLAR.....	89
ÖZGEÇMİŞ	

ÖZET

Çoklu Video Görüntüleri Üzerinde Akıllı Hedef Takibi

Sevinç AY

Doktora Tezi

FIRAT ÜNİVERSİTESİ
Fen Bilimleri Enstitüsü

Yazılım Mühendisliği Anabilim Dalı

Temmuz 2023, Sayfa: xii + 96

Son yıllarda, teknoloji alanında yaşanan gelişmelerle birlikte kamera, veri depolama cihazları ve yazılım sistemleri bileşenlerinden oluşan görsel gözetim sistemlerinin kullanımı da yaygınlaşmıştır. Görsel gözetim yapmak için kullanılan bu sistemler, nesneler veya insanların hareketlerini, davranışlarını gözlemlenmede yaygın olarak tercih edilmektedirler. Trafikte kullanılan gözetleme sistemlerinde çeşitli algoritmalar sayesinde kalabalıklar arasında şüpheli şahısların ya da araçların anormal davranışlarının tespiti mümkün olmaktadır. Bu hareketlerin doğru bir şekilde tespit edilmesi oluşabilecek birçok tehlikenin önceden tespitine ve engellenmesine yardımcı olmaktadır. Günlük yaşamdaki birçok alanda görsel takip sistemlerinin kullanılma gereksinimi, son yıllarda araştırmacıların nesne takibi konusuna olan ilgisini arttırmıştır. Nesne tespiti ve takibi, görsel gözetim sistemlerinin yorumlanması amacıyla giderek daha yaygın olarak kullanılmaktadır.

Tez çalışması kapsamında bir yol güzergâhında bulunan birden fazla kameradan alınan çoklu video görüntüleri kullanılmıştır. Bu tez çalışmasının ilk aşamasında, şüpheli olarak nitelendirilen bir aracın çoklu trafik video görüntüleri üzerinde geliştirilen araç takip yöntemi kullanarak izlenmesi sağlanmıştır. İkinci aşamasında aracın görüntülenebileceği kameraların belirlenmesi sağlanmıştır. Son aşamada ise derin öğrenme tabanlı nesne tanıma algoritmaları kullanılarak şüpheli aracın tespit edilme işlemi gerçekleştirilmiştir. Tez çalışmasında sırasıyla Bölge Tabanlı Evrişimli Sinir Ağı (B-ESA), Daha Hızlı Bölge Tabanlı Evrişimli Sinir Ağı (Daha Hızlı B-ESA) ve Yalnızca Bir Kez Bakarsınız (YOLO) kullanılmıştır. Deneysel sonuçlardan elde edilen ortalama kesinlik değerlerinin ortalaması (mean Average Precision-mAP) karşılaştırılmış ve Daha Hızlı B-ESA derin öğrenme modelinin ResNet101 omurga ağı ile eğitilmesi ile elde edilen % 89 değerinin en yüksek ortalama olduğu belirtilmiştir.

Anahtar Kelimeler: Derin Öğrenme, Araç Takibi, Araç Tespiti, B-ESA, Daha Hızlı B-ESA, YOLO

ABSTRACT

Intelligent Target Tracking on Multi Video Images

Sevinç AY

Ph.D. Thesis

FIRAT UNIVERSITY
Graduate School of Natural and Applied Sciences

Department of Software Engineering

July 2023, Pages: xii + 96

In recent years, with the developments in the field of technology, the use of visual surveillance systems consisting of camera, data storage devices and software systems has become widespread. These systems, which are used for visual surveillance, are widely preferred in observing the movements and behaviors of objects or people. Thanks to various algorithms in the surveillance systems used in traffic, it is possible to detect the abnormal behavior of suspicious persons or vehicles among the crowds. Accurate detection of these movements helps to detect and prevent many dangers that may occur. The need to use visual tracking systems in many areas of daily life has increased the interest of researchers in object tracking in recent years. Object detection and object tracking is increasingly used for the interpretation of visual surveillance systems.

Within the scope of the thesis study, multiple video images taken from more than one camera located on a road route were used. In the first stage of this thesis, a vehicle, which is considered suspicious, was monitored using the vehicle tracking method developed on multiple traffic video images. In the second stage, it was ensured that the cameras on which the vehicle could be viewed were determined. In the last stage, the suspicious vehicle was detected by using deep learning-based object recognition algorithms. In the thesis study, Region-Based Convolutional Neural Network (CNN), Faster Region-Based Convolutional Neural Network (Faster CNN) and You Only Look Once (YOLO) were used. The mean average precision (mAP) obtained from the experimental results were compared and it was stated that the 89% value obtained by training the Faster CNN deep learning model with the ResNet101 backbone network was the highest mean.

Keywords: Deep Learning, Vechile Tracking, Vechile Detection, Fast RCNN, Faster RCNN, YOLO

ŞEKİLLER LİSTESİ

Sayfa

Şekil 2.1. Kamera hareketi ve kullanım durumlarına göre farklı video türlerinin sınıflandırılması	11
Şekil 2.2. Arka plan çıkarım sürecine ait blok diyagram.....	12
Şekil 2.3. Arka plan çıkarımına dayalı bir genel nesne tanıma sisteminin şeması	18
Şekil 2.4. Bir video çerçevesinde GKM uygulaması (a) GKM uygulamasından önce, (b) GKM uygulamasından sonra.....	24
Şekil 2.5. Nokta takibi	26
Şekil 2.6. Çekirdek takibi dönme hareketi.....	27
Şekil 2.7. Çevre çizgisi oluşturma	27
Şekil 2.8. Hedef takip sisteminin genel yapısı.....	29
Şekil 2.9. Hedef takip yaklaşımları.....	30
Şekil 2.10. Blob analiz modülü.....	31
Şekil 2.11. B-ESA mimarisi	35
Şekil 2.13. Bölge öneri ağı ve bağlantı kutuları	38
Şekil 2.14. YOLO regresyon yaklaşımı.....	40
Şekil 2.15. YOLO katman yapısı.....	41
Şekil 2.16. AlexNet mimarisinin genel yapısı	44
Şekil 2.17. Aktivasyon fonksiyonları	45
Şekil 2.18. AlexNet katmanları.....	46
Şekil 2.19. VGG16 mimarisi	47
Şekil 2.20. GoogleNet mimari yapısı.....	48
Şekil 2.21. GoogleNet başlangıç katmanı.....	48
Şekil 2.22. GoogleNet başlangıç mimarisi	49
Şekil 2.23. Görüntü tanımda derin artık öğrenme.....	50
Şekil 2.24. ResNet artık blok	51
Şekil 2.25. ResNet ilk 34 katmana ait ağ mimarisi ile düz ağı karşılaştırılması	52
Şekil 2.26. Mobilenet v1 temel modeli.....	53
Şekil 2.27. Derinlemesine ayrılabilir evrişim adımı	54
Şekil 2.28. Mobilenet mimarisi.....	55
Şekil 3.1. Araç takip ve tespit sistemine ait işlem adımları	57
Şekil 3.2. (a) Hedefin ilk çerçevede manuel olarak seçilmesi, (b) Hedefin Blob analiz ile takip edilmesi...58	
Şekil 3.3. Arka plan tespiti için morfoloji süreci (a) Ön plan görüntüsü, (b) Gürültü içeren arka plan görüntüsü ve filtre uygulanarak temizlenmiş görüntü	59

Şekil 3.4. Hedef aracın takip işlemi (a) Araç seçimi, (b) Takip işlemi, (c) Takip edilen hedefin engellere karşı dayanıklılığı, (d) Takip edilen hedefin boyut değişimi ve uzaklığa dayanıklılığı.....	60
Şekil 3.5. Hedef araç takibi algoritmasına ait akış diyagramı	61
Şekil 3.6. Kameraların tespit sürecine ait akış diyagramı.....	63
Şekil 3.7. Δy ve Δx değerlerinin eksenler üzerinde gösterimi.....	64
Şekil 3.8. Eğim açısı gösterimi	64
Şekil 3.9. Ekran yön bilgisi içeren koordinat sistemi	65
Şekil 3.10. Takip edilen farklı araçların eğim ve yön bilgileri	66
Şekil 3.10. Bölgedeki kameraların yerleşimi ve id numaraları.....	68
Şekil 3.11. Hedefin tespit edileceği tahmin edilen kameranın aranacağı bölge	68
Şekil 3.12. Haversine formülünün küresel dünya üzerinde gösterimi	71
Şekil 3.13. Hedef aracın farklı kameralarda görüntülenmesi	72
Şekil 3.14. Eğitim süreci grafiği (a) YOLO ile eğitilmiş MobileNet ağı, (b) Daha Hızlı B-ESA ile eğitilmiş Resnet 101 ağı	78
Şekil 3.15. Derin öğrenme performans sonuçlarının karşılaştırılması.....	79
Şekil 3.16. Derin öğrenme modelleri için hassasiyet ve kesinlik eğrileri: (a) B-ESA, (b) AlexNet, (c)VGG16, (d) VGG19	80
Şekil 3.17. Derin öğrenme modelleri için hassasiyet ve geri çağırma eğrileri: (a) ResNet50, (b) Resnet101, (c) GoogleNet, (d) YOLO	81
Şekil 3.18. Eğitim ağındaki kayıp eğrileri (a) AlexNet, (b)Resnet101, (c) Resnet50, (d) YOLO.....	82
Şekil 3.19. Araç tespiti ile tespit edilen araç görüntüleri (a) ResNet50, (b) ResNet101, (c) VGG19, (d) YOLO.....	83

TABLolar LİSTESİ

	Sayfa
Tablo 3.1. Δy ve Δx değerlerinin koordinat ekseninde bulundukları bölgeye göre alacakları değerler	66
Tablo 3.3. Deneysel çalışmalar için kullanılan bilgisayarın donanım özellikleri	74
Tablo 3.4. Derin öğrenme modelleri için parametre ayarları	75
Tablo 3.5. Farklı omurga ağlarının eğitim sürecine etkisi	79
Tablo 3.6. Hedef araç tespit aşamasının benzer çalışmalarla kıyaslanması	84
Tablo 3.7. Nesne takibi alanında yürütülen çalışmaların avantaj ve dezavantajlarının karşılaştırılması.....	85

SİMGELER VE KISALTMALAR

Simgeler

λ	: Etiketlenen hareket alanı
B_s	: Piksel s’de ki arka plan değeri
$\Sigma_{k,t}$	: Kovaryans matrisi
$P(X_t)$	: Gauss olasılık değeri
T	: Eşik değeri
$\mu_{k,t}$	: Gauss fonksiyonunda ortalama değeri
$w_{k,t}$	: Gauss bileşenlerinin ağırlık değeri

Kısaltmalar

AÇ	: Arka Plan Çıkarımı
B-ESA	: Bölge Tabanlı Evrişimli Sinir Ağı
BLOB	: İkili Büyük Besne (Binary Large Object)
BTA	: Bölge Teklif Ağı
BÜK	: Birleşme Üzerinden Kesişim
CNN	: Convolutional Neural Network
Daha Hızlı B-ESA	: Daha Hızlı Bölge Tabanlı Evrişimli Sinir Ağı
DN	: Doğru Negatif
DP	: Doğru Pozitif
DVM	: Destek Vektör Makineleri
ESA	: Evrişimli Sinir Ağı
Fast R-CNN	: Hızlı Bölge Tabanlı Evrişimsel Sinir Ağı
Faster R-CNN	: Daha Hızlı Bölge Tabanlı Evrişimsel Sinir Ağı
GİB	: Grafik İşlem Birimi
GKM	: Gauss Karışım Modeli
Hızlı B-ESA	: Hızlı Bölge Tabanlı Evrişimli Sinir Ağı
İBH	: İlgi Bölgesi Havuzlama
mAP	: Ortalama Kesinlik Değerlerinin Ortalaması (Mean Average Precision)
OICA	: Uluslararası Motorlu Araç Üreticileri Örgütü
ÖSA	: Özyinelemeli Sinir Ağları
RCNN	: Bölge Tabanlı Evrişimli Ağ
REDB	: Düzleştirilmiş (Rektifiye Edilmiş) Doğrusal Birim
SACON	: Örnek Mutabakat (Sample Consensus)
TAD	: Tek Atış Detektörü
YOLO	: Yalnızca Bir Kez Bakarsınız (You Only Look Once) Algoritması
YSA	: Yapay Sinir Ağı
YN	: Yanlış Negatif
YP	: Yanlış Pozitif

1. GİRİŞ

Video kayıt sistemlerinin başta güvenlik uygulamaları olmak üzere hayatımızın birçok alanında yaygın şekilde kullanımı gün geçtikçe artmaktadır. Video kayıt sistemlerinin temel bileşeni olan kameralar ile akıllı güvenlik sistemleri, biyometrik tanıma, medikal görüntüleme ve ürün kalite denetim uygulamaları yapmak mümkün olmaktadır. Temel görüntü algılayıcıları olarak kabul edilen kameralar tarafından yapılan video gözetimi, alışveriş merkezleri, bankalar, trafik yoğunluğu olan bölgeler, kalabalık ortamlar ve sınırlar gibi güvenlik denetimin yüksek olduğu bölgelerde uzun süredir yaygın bir şekilde kullanılmaktadır [1, 2].

Kameralı sistemlerin en yaygın kullanıldığı alanlardan birisi görsel gözetim sistemleridir. Görsel gözetim sistemleri genellikle kameralar, görüntü işleme yazılımları ve veri depolama cihazlarından oluşmaktadır. Kameralar, görüntüleri toplamaktadır ve bu görüntüler görüntü işleme yazılımları tarafından işlenerek kullanılmaktadır. İşlenen veriler daha sonra depolama cihazlarına kaydedilmekte ve gerektiğinde erişilebilir hale getirilmektedir. Bu sistemler, insanlar tarafından yürütülen gözetim görevlerinin yerine geçebilmekte ve zaman, maliyet ve hata oranlarını azaltabilmektedir [3, 4].

Görsel gözetim sistemlerinin trafikte kullanımı araçların giderek daha popüler hale geldiği modern yaşamda, trafik araçlarının takibi ve şehir güvenliğinin sağlanması amacıyla daha fazla ilgilenilen bir alan haline gelmeye başlamıştır [5]. Trafik gözetleme sistemlerinin yaygınlaşması bu sistemler üzerinde araç tespiti ve takibinin gerekliliğini ortaya çıkarmıştır. Özellikle trafik senaryolarında araçların türü ve konumu tüm trafik ortamını büyük ölçüde etkileyen bir husus olabilmektedir. Bu nedenle de trafik araçlarının doğru tespiti kentsel trafik yönetimi için oldukça önemlidir [6]. Trafik gözetleme sistemlerinin yaygınlaşması araç tespiti ve takibi alanında yaşanan problemleri de artırmıştır.

Dünyada ki araç sayısı gün geçtikçe artmaktadır. Uluslararası Motorlu Araç Üreticileri Örgütü (International Organization of Motor Vehicle Manufacturers, OICA), 2021 yılında yaklaşık 71 milyon araç üretildiğini bildirmektedir. Gün geçtikçe artan sayıda aracın yönetilmesi, dünya çapındaki ülkelerin uğraşmak zorunda olduğu en büyük sorunlardan biridir. Klasik araç takip teknikleri ve nesne tanıma algoritmaları, artan veri miktarı ile başa çıkmakta zorlanmaktadır. Bu zorlu görevle başa çıkmak için her geçen gün yeni yöntemler ve algoritmalar ortaya çıkmaktadır [6, 7].

Son yıllarda araç tespiti ve araç takibi alanında çok sayıda yöntem tasarlanmıştır. Bu yöntemlerden derin öğrenme tabanlı olan yaklaşımlar giderek öne çıkmaktadır [8]. Bilgisayarlı görme alanında yaygın olarak kullanılan güçlü öğrenme yeteneklerine sahip Evrimsel Sinir Ağları (ESA) ile araç tanıma teknolojisi büyük ölçüde gelişmiştir. Güçlü özellik çıkarma yeteneğine sahip ESA, insan bilişsel sisteminin işlevsel yeteneklerini yapay olarak çoğaltmak için tasarlandığından,

geleneksel yöntemlere kıyasla çeşitli bilgisayarla görme görevlerinde daha iyi performanslar ortaya çıkarmaktadır. Donanım teknolojilerinde ivme kazanan gelişimler ve özellikle paralel hesaplama için grafik işlem birimi (GİB) kullanımıyla birlikte, birçok derin öğrenme algoritmasının da kullanımı yaygınlaşmaya başlamıştır. Bölge Tabanlı Evrişimli Sinir Ağı (B-ESA), Hızlı Bölge Tabanlı Evrişimli Sinir Ağı (Hızlı B-ESA), Daha Hızlı Bölge Tabanlı Evrişimli Sinir Ağı (Daha Hızlı B-ESA) ve yine nesne tanıma amaçlı Yalnızca Bir kez Bakarsınız (YOLO) vb. birçok derin öğrenme yöntemi araç tespiti amacıyla kullanılmaya başlanmıştır [5, 6, 8].

Daha Hızlı B-ESA algoritması, yüksek hassasiyete ve güçlü tespit oranlarına sahiptir. Karmaşık trafik ortamlarında çok nesneli algılamının yüksek doğruluğunu ve hızını sağlayabildiği ayrıca hareketli küçük nesnelerin tanınmasında gösterdiği başarıdan dolayı birçok araştırmacı tarafından temel alınarak kullanılmaya başlanmıştır. Bu nedenle, bu çalışmada ağırlıklı olarak Daha Hızlı B-ESA ve YOLO tabanlı araç tanıma yöntemleri gibi derin sinir ağlarına odaklanılmıştır [6, 8, 9].

1.1. Problem Durumu

Görsel gözetim sistemlerinde hedef/nesne takibi problemi nesnenin elde edilmesi ve nesnenin takibi olmak üzere iki temel başlık altında incelenmektedir. Video üzerindeki hedef takibinin temel amacı, takip edilen nesnenin pozisyonu, yönü ve hızı gibi zamansal bilgilerini çıkartmaktır. Sonraki aşamada videonun ardışık çerçeveleri kullanılarak videodaki takip edilen hedefler arasında benzerlik ilişkisi aranmaya çalışılmaktadır. Videoda hedef takibi yapılabilmesi için tespit işlemi gerçekleştirilmelidir. Bu nedenle takip ve tespit işlemleri birbiriyle yakından ilişkilidir [1, 2, 10].

Hedef tespiti ve takibi alanında uzun yıllardır sürdürülen çalışmalara rağmen hala bazı problemlerin çözümüne yönelik istenilen aşamaya gelinemediği görülmektedir. Bunların en önemli olanları arasında çoklu hedef takibi, yapılan ölçümlerin hedefler ile eşleştirilememesi, yanlış yorumlamalar sonucu oluşan yeni hedefler veya yeni hedeflerin tespiti ile birlikte eski hedeflerin silinmemesi gibi problemler önemli yer tutmaktadır. Günümüzde hedeflerin örtüşmesi ve çapraz geçişler yapılan çalışmalarda çözilemeyen en önemli problemler olarak gösterilmektedir [8, 9, 10]. Bunların dışında 3 boyutlu gerçek görüntü uzayından 2 boyutlu imge uzayına dönüşümle birlikte oluşan kayıplar, gürültü, hareketli ve karmaşık arka plan, eklentili nesne yapısı ve karmaşık nesne hareketleri, yarı veya tam örtüşme durumları, sahnede gerçekleşen farklı aydınlanma şartları gibi nedenlerden dolayı görüntü işleme yaklaşımlarının başarımını düşürmektedir [11].

Literatürdeki çalışmalar incelendiğinde ölçüm ve hedefin eşleştirilmesi amacıyla uzaklık, ve hedefin büyüklüğü parametre olarak kullanılmaktayken, hedeflerin örtüşmesi amacıyla hedefin rengi ve büyüklüğü parametre olarak kullanılmaktadır [10, 11]. Hedef takibinde iki önemli unsur vardır. Bunlar takip edilen hedef sayısı ve kullanılan kamera sayısıdır. Yapılan çalışmaya bağlı

olarak takip edilecek nesne bir ya da birden çok olabilmektedir. Hedefin durumuna göre sabit ya da hareketli bir kamera ya da birden fazla kamera kullanılabilmektedir. Çoklu kamera özellikle sahnede sınırlı görüş açısı olduğu durumda veya örtüşme problemlerine çözüm olarak tercih edilmektedir. Bu durumda yine kameralar arasındaki veri iletişimi hedef takibinde kolaylık sağlamaktadır. Buna rağmen kamera sayısının artması işlem karmaşıklığını arttırarak başka problemlere yol açmaktadır [11].

Bütün bu sorunlara çözüm getirmek amacıyla birden fazla video görüntüsü üzerinden gürbüz bir hedef takip sistemi, diğer bir tanımla hedef takibini zorlaştıran olumsuz etkenlere karşı dayanıklı ve etkili bir takip sistemi gerçekleştirmek, bu tezin en önemli hedefini oluşturmaktadır. Literatürde incelenen temel hedef tespitinde ve takibinde kullanılan yaklaşımlar incelendiğinde birden fazla yöntemin olduğu görülmektedir. Çalışma kapsamında farklı yöntem ve tekniklerin uygulanarak karşılaştırılması ve iyileştirilmesi gerçekleştirilmiştir.

1.2. Tezin Amacı

Görsel gözetim sistemlerinde kullanılan nesne veya hedef takibi uygulamaları, trafik kontrolü, insan davranışlarının izlenmesi, araç navigasyonu gibi bilgisayarlı görme uygulamalarının temelini oluşturmaktadır. Bu sistemlerde nesne ya da hedefin bulunduğu yerin doğru şekilde tespiti birçok anormal olarak kabul edilen olayın tespit edilmesini kolaylaştırmaktadır [12].

Görsel gözetim sistemlerinde alınan kayıtlar üzerinde yapılan şüpheli arama, kontrol ve izleme işlemlerini kapsayan denetleme işlemi, elle, yarı-otomatik veya tam-otomatik gözetim olmak üzere üç başlık altında incelenebilmektedir. Elle yapılan denetlemede video kayıt sistemlerindeki kayıtlar bir insan tarafından denetlenmektedir. Yarı otomatik denetimde görüntülerde önemli bir hareket olduğu durumlarda kayıt yapan bir kamera bulunmakta ve yapılan kayıtlar daha sonra bir insan tarafından detaylı bir şekilde incelenmektedir. Tam otomatik denetimde insan takibinden bağımsız bir şekilde kontrol, denetim, analiz ve raporlama işlemleri gerçekleşmektedir [1, 2].

Literatürdeki hedef takibi çalışmaları incelendiğinde çoğunlukla kısıtlı bir bölgede bir ya da birden fazla kamera ile yapılan kayıtlar üzerinde bilgisayarlı görü çalışmaları yapılarak belirlenen hedefin takibinin yapılması üzerine yürütüldüğü görülmektedir. Bu tez çalışmasında hedefin bir kamerada tespitinden sonra hedefin ilerleme yönü, hızı ve sahnede ki süresi göz önünde bulundurularak bölgedeki diğer kameralarda hedefin takibine devam edilmesi ve tespit edildiği belirlenen tüm kameralarda görüntülenme süreleriyle birlikte raporlanması amaçlanmaktadır. Takibin gözlemleneceği diğer kameralardan elde edilen kayıt süresinin tamamının değil hedefin olma ihtimalinin olduğu sürenin öncesi ve sonrasında başlayarak oluşturulacak algoritmayla belli

süre aralıklarının izlenmesi sağlanması ve böylelikle hedef takip süresinin kısaltılması hedeflenmiştir.

Birden fazla kamera kaydında bir hedefin yada şüphelinin takibi güvenlik ekiplerine destek olabileceği gibi birçok sorunun çözümünde hem insan gücünden hem de zamandan tasarruf sağlayacaktır. İstanbul'da 2022 yılında İstiklal Caddesi'nde meydana gelen bir bombalı saldırı eyleminde İstanbul Emniyet Müdürlüğü ekipleri tarafından, güvenlik kameralarınca kaydedilen görüntüler üzerinden şüphelinin eşkalini tespit etmek için yoğun çalışma başlatılmıştı. Polis ekipleri, meydana gelen patlamanın hemen ardından harekete geçmiş ve geniş çaplı bir inceleme yapılmıştır. 1200 güvenlik kamerasını inceleyen ekipler sayesinde, olayın failinin kamera görüntülerinden tespit edilerek yakalanması sağlanmıştır [13]. Böyle bir inceleme için onlarca polis memuru görevlendirilip saatlerce kayıtlar incelenmiştir. Bu tez çalışmasında, şüpheli takibi ve hedef arama gerektiren durumlarda video kayıtlarının incelenmesi ve denetlenmesi işleminin otomatik hale getirilmesi ve bu sayede insan gücü ile zamanın etkin kullanımının sağlanması amaçlanmaktadır.

Bu tez çalışması kapsamında birden fazla video görüntüsü üzerinde izlenmesi gereken hedefin belirlenmesi, algılanması ve takip işleminin gerçekleştirilmesi hedeflenmiştir. Hareketli nesnenin gidiş yönü, hızı ve kayıt süresi göz önünde bulundurularak o güzergâhta bulunan diğer kamera kayıtları içerisinden uygun olan belirlenerek takibin devamı sağlanmaya çalışılmıştır. Bu takip işleminin hızlı ve doğru bir şekilde gerçekleştirilebilmesi için mevcut yöntemlerin yanı sıra probleme uygun yeni algoritmaların geliştirilmesi veya mevcut olanların iyileştirilmesi planlanmaktadır.

Tez çalışmasının ana amacı güvenlik amacıyla kullanılan video gözetleme sistemlerinden alınan kayıtların otomatik incelenmesini sağlamak ve daha kısa sürede belirlenen hedefin tespit edilmesini gerçekleştirmektir. Bu çalışmada hedefin özellikle karşılaşılan engellere ve araçların yakınlığı gibi görüntü bozucu koşullara rağmen takibin devam edebilmesi için yeni bir algoritma geliştirilmiştir. Ayrıca hedef aracın bir yolun birden fazla noktasında görüntü alabilen kamerada tespiti sağlanmaya çalışılmıştır.

1.3. Literatür İncelemesi

Literatürde nesne tanıma ve nesne takibi yapılan birçok araştırma mevcuttur. Tez çalışmasının bu bölümünde araç takibi ve araç tespitini de içeren bu çalışmalara dair bir literatür incelemesi sunulmuştur. Bu çalışmalar nesne takibi ve nesne tespiti olmak üzere iki başlık altında incelenmiştir.

1.3.1. Nesne Takibi Çalışmaları

Kiratiratanapruk ve ark. [14], tarafından önerilen çalışmada bir trafik izleme sisteminde nesne takibi için gradyan tabanlı bir ön plan algılama tekniği sunulmuştur. Video gözetim sistemlerinde nesne tanıma ve nesne izlemenin en önemli aşamalar olduğu belirtilen çalışmada nesne tanıma adımına odaklanılmıştır. Video üzerinde hareketli nesne tanımaya yönelik en temel yaklaşım arka plan çıkarma yaklaşımıdır. Arka plan çeşitli tekniklerle belirlenebilmektedir ancak bu tekniklerin birçoğu çeşitli dış etkenlerden kolaylıkla etkilenmektedir. Çalışma kapsamında ışık değişimlerine duyarlı geliştirilmiş kenar algılama tabanlı bir arka plan modelleme yaklaşımı önerilmiştir. Bu yöntem popüler olan arka plan modelleme teknikleri ile karşılaştırılmıştır. Ayrıca literatürdeki yaklaşımlar ile önerilen yaklaşım arasında, aydınlatma değişikliği altında ön plan algılamasının karşılaştırılması yapılmıştır. Çalışmada önerilen teknik ile diğer teknikler arasındaki hesaplama süreleri arasındaki farklar dikkate alınmıştır. Elde edilen sonuçlarda geliştirilen yöntemin, dış ortamlarda ışık değişimlerine ve video kamera sensör karakteristiğine karşı dayanıklı, ön plan nesneleri için yüksek algılama oranlarına sahip olduğu görülmüştür.

Liu ve ark. [15], optik akış tabanlı bir araç takip yöntemi sunmuşlardır. Akıllı ulaşım sistemleri için araç takibinin önemine vurgu yapılan çalışmada araç takibini zorlaştıran ölçek değişimleri, düşük çözünürlüklü veriler ve benzer renklerin karışması gibi sorunlara da değinilmiştir. Çalışma kapsamında ilk olarak aracın görüntüdeki ilk çerçevelerde yönünü belirleyebilmek için optik akış yöntemi kullanılmıştır. Ayrıca örnekleme takip algoritması olarak adlandırılan geliştirilmiş bir Markov zinciri modeli önerilmiştir. Bu yöntem ile ölçek değişikliği problemine çözüm getirilebilmiştir. Özellikle ilk çerçevelerde aracın hareket yönünü belirleyebilmek için kullanılan optik akış yöntemi aynı zamanda aracın hareket hızının da elde edilmesini sağlamıştır. İkinci olarak yeni ve benzer bir aracın sahneye girişinde oluşabilecek problemleri gidermek için mesafe faktörü dikkate alınmaktadır. Daha düşük çözünürlüklü videolarda araç takibi yapabilmek için bir özellik şablonu sunulmuştur. Çalışma kapsamında kullanılan Monte Carlo örnekleme rastgele bir örnekleme yöntemi olduğundan, birkaç deney yaparak farklı parçacık sayıları için en yüksek, en düşük ve ortalama doğruluk oranlarının karşılaştırılması yapılmıştır. Çalışmada benzer renkli araçların yarattığı algılama sorunlarını giderecek bir mesafe faktörü üzerine odaklanılmıştır. Bu soruna ve düşük çözünürlük sorunlarına çözüm sunulabilmiştir. Sonuç olarak geleneksel algoritmalarından daha iyi performansa sahip bir takip algoritması önerilmiştir.

Sushmitha ve ark. [16], trafikte birden fazla aracın takibi üzerine yaptıkları çalışmada yeni bir algoritma sunmuşlardır. Video gözetim sistemleri üzerinde çoklu araba takibi yapılan çalışmada yöntemin, trafikte sıkışıklık problemlerini çözmek, olay mahallinde ki şüpheli tespiti, kaza ve hırsızlık tespiti gibi alanlarda kullanılması amaçlanmıştır. Araçların hareket halinde şeklinin

algılanması ile ilgili sorunun üstesinden gelmek amacıyla video çerçevelerine bir dizi teknik uygulanmıştır. Bu teknikler çerçevelerin dönüştürülmesi, ön işleme, hareket bölümlendirme, özellik çıkarma ve izleme olarak sıralanabilmektedir. Farklı modüllere dayalı olan teknikler bir araya gelerek çoklu araba takibi yapmak için tasarlanmış yöntemin temelini oluşturmaktadır. Videolar sisteme girdi olarak alındıktan sonra bir görüntü dizisine dönüştürülmektedir. RGB olarak alınan görüntünün rengini düzeltmek için griye dönüştürülen görüntü üzerine ön işlemler uygulanmaktadır. Ön plan görüntüsünün elde edilebilmesi amacıyla arka plan çıkarma tekniği kullanılmıştır. Blob analizi kullanılarak, görüntünün bölgesi ile nesnenin sınırlayıcı kutusu belirlenmekte ve ön plan nesnesindeki fazla gürültüler giderilmektedir. Hareket halindeki nesnenin konumu, nesne takibi yardımıyla elde edilmektedir. Video dizisi içerisinde aradışık çerçevelerde hedef nesne takip edilmektedir.

Nguyen [17], tarafından hazırlanan çalışmada MobileNet omurga yapısını kullanan geliştirilmiş Daha Hızlı B-ESA kullanılarak daha hızlı araç tanıma uygulaması yapılmıştır. Çalışmada derin evrişimli sinir ağlarının akıllı ulaşım sistemlerinde gösterdiği başarıya rağmen araç yoğunluğu ve tıkanıklık durumları ile görüntüde aracın kesilmesi durumlarında hala istenilen seviyeye ulaşılamadığının altı çizilmiştir. Çalışmada önce Daha Hızlı B-ESA'de temel evrişim katmanını oluşturmak için MobileNet mimarisi kullanılmış daha sonra, bölge öneri ağından sonra önerilen algoritma ile, yinelenen teklif sorununu çözmek amaçlanmıştır. İlgi alanı havuzlama (İBH) katmanından sonra sınıflandırmak ve tespit edilen her araç için sınırlayıcı kutuyu belirlemek üzere MobileNet mimarisindeki derinlemesine ayrılabilir evrişim yapısı kullanılmıştır. Çalışma kapsamında KITTI araç veri seti üzerinde deneyler yapılmıştır. Sonuç olarak önerilen yaklaşımın hem algılama doğruluğu hem de çıkarım süresi açısından orijinal Daha Hızlı B-ESA'ya kıyasla daha iyi performans elde ettiğini belirtilmektedir. Önerilen yöntemin performansı, orijinal Daha Hızlı B-ESA çerçevesine kıyasla KITTI test setinde % 4 ve kullandıkları diğer test setinde %24.5 oranında iyileştirilmiştir.

1.3.2. Nesne Tespiti Çalışmaları

Tez çalışmasının bir diğer aşamasını oluşturan nesne tespiti alanında her geçen gün yenilenen yöntemlerle oldukça başarılı sonuçlar elde edilebilmektedir. Literatürde bu alanda yapılmış çalışmalar aşağıda sunulmuştur.

Luo ve ark. [6], tarafından yapılan çalışmada trafik sahnesi görüntülerinden araç tespit etmek amacıyla özellik zenginleştirme ile Daha Hızlı B-ESA'ya dayalı bir model sunulmuştur. Karmaşık aydınlatmaya sahip ortamlarda, nesne tanımanın güçleştiği durumlarda yüksek çözünürlüklü görüntüleri önceden işleyerek çözümler sunulmaya çalışılmıştır. Görüntülerdeki ışık değişimi ve gölge etkisini azaltmak için Retinex tabanlı bir düzeltme algoritması kullanılmıştır. Özyinelemeli Sinir Ağları (ÖSA), ile nesnelerin algılanma performansı arttırılmak istenmiştir. Çalışma

kapsamında Daha Hızlı B-ESA optimize edilmiş ve çapraz katman bağlantılarını üreten çok katmanlı bir özellik çıkarma yöntemi sunulmuştur. Önerilen modelde özellik zenginleştirilmesi ile birlikte çok katmanlı özellik bilgisini birleştirerek hedef bilgisi elde edilmektedir. Özellikle küçük hedeflerde hedef tespitinde literatürdeki çalışmalara göre sağlam bir model önerildiği öne sürülmüştür. Deneysel çalışmalarda algılama verimliliğini arttırabilmek için veri kümesine uygun çapalar oluşturmak amacıyla K-ortalama kümeleme algoritması kullanılmıştır. UA-DETRAC veri setindeki dört araç kategorisi için yüksek algılama sonuçları elde edilmiştir. Literatürde kullanılan benzer yöntemlerle karşılaştırıldığında, bu yöntem doğrulukta önceki çalışmalardan yüksek doğruluk elde etmesine rağmen önerilen modelin sınıflandırma sürecinde sadece dört tip araç bulunması da en büyük sınırlılık olarak ortaya çıkmaktadır. Ayrıca gerçek sahneler üzerinde önerilen modelin doğruluk oranı düşmektedir. Bu nedenle gelecekteki araştırmalar için, araç sayısının artırılması gerektiği ve gerçek sahnelerin uygulanmasını kolaylaştırabilmek için algılama hızının artırılması gerektiği çalışmanın sonunda tavsiye edilmiştir.

Maity ve ark. [8], akıllı trafik gözetimi sistemlerinde otomatik hareket eden araçların tespiti üzerine bir çalışma yapmışlardır. Araçların doğru şekilde tespit ve takibinin yapılmasını amaçlayan birçok çalışmayı incelenmişler ve sert hava koşulları ve uygunsuz aydınlatma ortamlarında başarımının düştüğünü belirtmişlerdir. Çalışma kapsamında Daha Hızlı B-ESA ve YOLO tabanlı araç tespit ve takip metodları karşılaştırılmıştır. Literatürdeki çalışmalar incelendikten sonra, bugüne kadar önerilen izleme yöntemlerinin çoğunlukla manuel olması ve yalnızca kamera görüntüsüne bağlı olması nedeniyle izleme yöntemlerinin geliştirilmesi gerektiği sonucuna varıldığı bildirilmiştir. Çalışmada araç tespiti yapmanın yanı sıra, çarpışmaları önlemek amacıyla araç takip etmenin de önemi üzerinde durulmuştur. Ayrıca YOLO sürüm 5 kullanılan bir araç tanıma sistemi tasarlanması ve uydu görüntüleri kullanılarak farklı araç sınıfları arasındaki benzerlikleri ve küçük boyutlu araçları tanıma üzerine de çalışılması tavsiye edilmiştir.

Wang ve ark. [9], tarafından hazırlanan çalışmada geliştirilmiş Daha Hızlı B-ESA kullanılarak trafik işaretleri tanınmıştır. Çalışmada trafik işaretlerini tanıma üzerine yapılan çalışmaların önce ilgi alanlarının belirlendiği ve sonra görüntü analizi uygulandığı ancak bu yaklaşımın eğitim sürecinde yaşattığı sorunlara değinilmiştir. Bu nedenle tek adımlı bir öğrenmeye dayalı bir çözüm önerisi sunulmuştur. Trafik görüntülerinden küçük nesneleri algılayabilmek için önce Daha Hızlı B-ESA kullanılmıştır. Sonraki aşamada yüksek kaliteli bölge önerilerinden, ağı optimize edebilmek için yeni bir yöntem sunulmuştur. Ayrıca, daha ayırt edici özellikler elde etmek amacıyla önerilen ağı omurga yapısı olarak Res2Net kullanılmıştır. Tsinghua-Tencent 100k veri seti üzerinde yapılan deneyler ile önerilen algoritmanın doğruluk ve geri çağırma değerlerinde diğer algoritmalarından yüksek performans elde ettiği bildirilmiştir.

Arinaldi ve ark. [18], tarafından trafik videolarından araç tespit ve sınıflandırılması için Daha Hızlı B-ESA temelli bir trafik analiz sistemi sunulmuştur. Çalışmada bu analiz kapsamında araç

sayımı, araç tipi sınıflandırması, videodan araç hızının tahmini ve şerit kullanımının izlenmesi gibi işlemler yapılmıştır. Dolayısıyla önerilen sistemin temeli trafik videolarında araçların tespiti ve sınıflandırılması işlemleridir. Bu amaçla iki model uygulanmıştır. İlk olarak Gauss Karışım Modeli ile Destek Vektör Makinası birlikte kullanılmıştır. İkinci olarak ise görüntülerdeki nesnelerin tespitinde elde ettiği başarımından dolayı Daha Hızlı B-ESA kullanılmıştır. Deneyler sonucunda, Daha Hızlı B-ESA'nın statik, çakışan veya gece koşullarındaki araçların tespitinde ilk önerilen modelden daha iyi performans gösterdiği belirtilmiştir.

Fan ve ark. [19], Daha Hızlı B-ESA'nın nesne tespiti çalışmalarında yüksek performans elde etmesi sonucunda araç tespiti çalışmalarında performansı daha yakından incelemiştir. Parametre ayarları ve algoritmik modifikasyon yoluyla Daha Hızlı B-ESA'nın araç algılama performansının önemli ölçüde artırıldığı belirtilmiştir. KITTI veri seti üzerinde kapsamlı deneyler gerçekleştirilen çalışmada performansın artırılması için yapılacak uyarlamalara da yer vermişlerdir. Omurga ağı olarak VGG tercih edilmiştir. Daha Hızlı B-ESA üzerinde hem eğitim hem de test ölçek boyutu değerleri farklılaştırılarak, bölge teklif değerleri, yinleme değerleri değiştirilerek, yerelleştirme ve tanıma analizi dahil araç tanıma görevinde çalışma süreleri kıyaslanmıştır. Araştırmacılar çalışmanın Daha Hızlı B-ESA'nın sınırlılıklarını ve farklı veri setlerine uygulanmasını araştıran gelecek araştırmalara yol gösterici olacağını bildirmişlerdir.

Cao ve ark. [20], Daha Hızlı B-ESA'nın küçük nesnelerin algılanmasında elde ettiği başarısı ön plana çıktığı için bu alanda yeni bir algoritma sunmuşlardır. İki aşamalı algılama fikri kullanılarak konumlandırma aşamasında sınırlayıcı kutu regresyonu için gelişmiş bir kayıp işlevi önerilmektedir. Ayrıca ilgi bölgeleri havuzlama (İBH) işlemi için ise doğrusal enterpolasyon kullanılmıştır. Nesne tanıma aşamasında konumlandırma sapması sorunlarını çözmek ve özellik haritasının daha çok bilgi içermesini sağlamak amacıyla çok ölçekli evrişimsel özellik birleştirmesi kullanılmıştır. Elde edilen deneysel sonuçlar, önerilen algoritmanın çözünürlüğü (0, 32) aralığında olan trafik işaretlerinde iyi performans gösterdiğini ve algoritmanın doğruluk oranının % 87'ye ulaştığını göstermektedir. Bu nedenle, algoritmanın küçük nesneleri algılamada başarılı olduğunu belirtmişlerdir.

Arcos-García ve ark. [21], COCO veri seti üzerinde ki trafik işaretleri derin öğrenme yöntemleri ile tanımaya çalışmışlardır. Çalışma sonucunda Daha Hızlı B-ESA'nın Resnet V2 omurga yapısıyla en iyi sonucu verdiği bildirilmiştir. Birden fazla omurga yapısıyla (Resnet V1 50, Resnet V1 101, Başlangıç V2, Başlangıç Resnet V2, Mobilenet V1 ve Darknet-19) daha önce yapılmış çalışmalar incelenmiştir. Bu modellerin değerlendirilmesi ve karşılaştırılması, ortalama kesinlik, bellek tahsisi, çalışma süresi gibi temel ölçümleri içermektedir. Elde edilen bulgular, Daha Hızlı B-ESA'nın Başlangıç Resnet V2'nin en iyi sonucu elde ettiğini, Resnet 101'in ise doğruluk ve yürütme süresi arasındaki en iyi dengeyi sağladığını göstermektedir. YOLO sürüm 2 yüksek doğruluk sonuçları elde etmesi ve Tek Atış Dedektörü MobileNet ise mobil ve gömülü

cihazlar için en hızlı dedektör olması, bellek tüketimi açısından en hızlı ve en hafif model olması açısından çalışmada ayrıca değerlendirilmiştir.

1.4. Çalışmanın Katkıları

- Literatürde yapılan çalışmalar incelendiğinde tek bir kamera kaydında hedef arandığı ya da çoklu kamera kayıtlarında yapılan tespit ve takip işlemlerinde karşılaşılan engel, görüntü bozulmaları gibi tek bir soruna odaklanıldığı görülmektedir. Önerilen yöntemde hedefin bir kamerada tespitinden sonra hedef aracın bölgede geçtiği tespit edilen diğer kameralarda takibine devam edilmesi sağlanmıştır. Ayrıca diğer kameralarda da hedef araç tespiti gerçekleştirilmiştir. Kullanılan algoritma sayesinde bölgedeki tüm kameralar yerine sadece aracın görüntüleneceği belirlenen kameralarda hedef araç aranmaktadır. Ayrıca kamera kaydının tamamının incelenmesi zaman alacağından sadece belirli süresinde hedef tespiti yapılması sağlanmıştır. Bu sayede hem zamandan hem de insan gücünden tasarruf elde edilmiştir.
- Veri toplama aşaması için hedefin seçildiği ilk video görüntüsü kullanılarak hedefin her video çerçevesinde de tespit edilmesi ve sınırlayıcı kutu bilgisi ile kaydedilmesi sağlanarak etiketli eğitim verisi oluşturulmaktadır. Bu yöntem etiketli eğitim verisi oluşturulması amacıyla kullanılan uygulamalar içerisinde yeni bir araç tespit algoritması olarak da kullanılabilir.
- Tez çalışmasının ilk aşaması olarak önerilen araç takip algoritmasının arka plan değişimlerine dayanıklı olduğu ve görüntüyü engelleyecek unsurlara rağmen takibe devam ettiği görülmüştür.
- ESA'lara dayalı modern nesne tanıma algoritmalarından B-ESA, Hızlı B-ESA, Daha Hızlı B-ESA ve YOLO' nun performans değerlendirilmesi yapılmıştır.

1.5. Tezin Organizasyonu

Tez çalışması 4 bölümden oluşmaktadır. Birinci bölümde tezin kapsamı, problemin tanımı, amacı, ortaya konulan çözümler ve yönelim gerekçelerinden bahsedilmiş ve önceki yöntemlere ilişkin bir literatür taraması sunulmuştur. İkinci bölümde kullanılan yöntemlerden arka plan çıkarım yöntemleri, Gauss karışım modelleri, derin öğrenme algoritmalarından Daha Hızlı B-ESA, YOLO gibi algoritmalar hakkında bilgi verilmiştir. Üçüncü bölümde araç takibi için geliştirilen algoritmaya yer verilmiştir, ayrıca birden fazla kamera arasından hedef aracın geçtiği güzergâhta sadece tespit edildiği kameralarda aranmasını sağlayan algoritma ve yöntemlerden bahsedilmiştir ve bu bölümünde araç tespit yöntemi olarak kullanılan derin öğrenme algoritmaları ile deneysel çalışmalar yapılmış ve sonuçları karşılaştırılmıştır. Tez çalışmasının son bölümünde ise tez çalışmasının amaçlarına ne ölçüde ulaşıldığı ve gelecekte yapılabilecek çalışma önerileri bulunmaktadır.

2. TEORİK ARKA PLAN

Tez çalışmasının bu bölümünde; nesne takibi ve nesne tanıma yöntemleri açıklanmıştır. Nesne tanıma yöntemleri ve takip yöntemleri ayrıntılı olarak değerlendirilerek tez çalışması kapsamında kullanılan yöntemler hakkında bilgi verilmiştir.

2.1. Nesne Tanıma

Video üzerinde uygulanan görüntü işleme uygulamalarının aşamalarından olan nesne tanıma, video işleme uygulamalarının en önemli parçası olarak kabul edilmektedir. Video görüntülerinde hareketli nesne tanıma birçok zorluğu da beraberinde barındıran önemli bir süreçtir.

2.1.1. Video İşleme, Nesne Takibi ve Nesne Tespiti

Video, statik görüntüler veya çerçeveler ile ilişkili ses verileri koleksiyonudur. Çerçeve, bir videonun parçası olarak gösterilen tek bir resim veya fotoğraftır. Kesintisiz bir videoyu oluşturmak için birçok tek resim hızlı bir şekilde art arda çalıştırılmaktadır. Bu nedenle, bir video bir dizi statik görüntü veya çerçeveden oluşmaktadır. Bir videoda, kameranın donanım ve yakalama hızına bağlı olarak görüntüler sabit bir hızda (saniyedeki çerçeve sayısı 25–30 veya 60) çekilmektedir. Saniyedeki çerçeve sayısı 30 olan bir kamera, bir video akışı oluşturmak için saniyede 30 fotoğrafın çekildiği anlamına gelir. Bu çerçeveler bir video akışından çıkarılabilmekte ve bir görüntünün işlendiği şekilde işlenebilmektedir. Tüm görüntü işleme işlemleri çıkarılan her video çerçevesi üzerinde gerçekleştirilebilir.

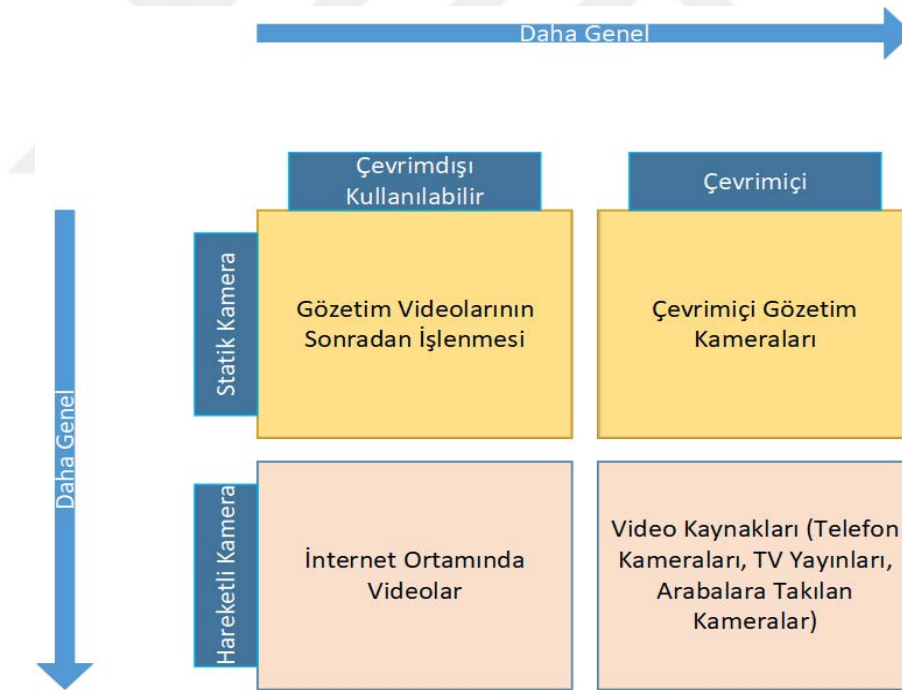
Videolarda nesne tespiti çalışmaları, temel olarak videoyu oluşturan görüntü dizilerinde bir nesnenin varlığının doğrulanmasını ve nesneyi tanımlayabilmek için tam olarak yerinin belirlenmesi aşamalarını içermektedir. Nesne tanıma, aynı zamanda bilgisayarlı görme alanında yaygın bir çalışma alanına sahip olan nesnelerin izlenmesi çalışmaları ile yakından ilişkilidir. Bir video, insan gözü tarafından içeriğin sürekliliğini algılayabilmesi için hızlı bir şekilde ve art arda gösterilen bir dizi resimden (çerçeveden) oluşmaktadır. Bu nedenle bilinen tüm görüntü işleme tekniklerinin bir videodan elde edilen tek tek çerçevelere uygulanabileceği açıktır. Videoda görüntü işleme uygulamaları kullanırken, ardışık iki çerçevenin içeriği genellikle yakından ilişkili olduğundan ve çok fazla anlamlı veri elde edilebildiğinden sıklıkla kullanılmaktadır [22].

Nesne takibi, bir video dizisinde yer alan bir nesnenin varlığının, konumu, boyutu, şekli gibi çeşitli uzamsal ve zamansal değişimlerinin izlenmesiyle ilgilenen önemli bir çalışma alanıdır. Birbirine yakın zaman aralıklarıyla çekilen bir dizi görüntünün birbirini izleyen çerçevelerindeki hedef bölgenin belirlenmesi olarak tanımlanabilmektedir. Nesne tanıma ve nesne takibi yakından ilişkilidir, çünkü takip işlemi genellikle nesnelerin algılanması ve tanınmasıyla başlamaktadır. Bir

video dizinde sonraki çerçevelerde bir nesneyi tekrar tekrar tanımak genellikle takibe yardımcı olmak ve nesnenin varlığını doğrulamak için gereklidir [22].

2.1.2. Videolarda Nesne Tespiti ve Segmentasyon

Videoda nesne tespit ve takip çalışmalarında kullanılan temel iki veri kaynağı bulunmaktadır. Bu kaynaklardan ilki görsel özellikler (örn. renk, doku ve şekil) diğeri ise hareket bilgisi olarak kabul edilmektedir. Önerilen yeni yöntemlerde görsel özelliklerin istatistiksel analizi ile hareket bilgisinin zamansal analizi birleştirilerek sağlam yaklaşımlar ortaya çıkmıştır. Yineden yöntemlerin genel stratejisi belli işlemlerin ardışık olarak yürütülmesi şeklindedir. Bu işlemler önce bir çerçeveyi renk ve doku gibi görsel özelliklere dayalı olarak birkaç bölgeye bölmek ve ardından piksellerin uzamsal komşuluğu gibi belirli kısıtlamalara tabi olarak benzer hareket vektörlerine sahip bölgelerin birleştirilmesi olarak sıralanabilmektedir [22, 23]. Şekil 2.1’de kameraların hareketli ya da sabit olması ve kayıtlarının çevrimdışı kullanılıp kullanılmadığı durumlara dair elde edilen video türlerinin sınıflandırılması gösterilmiştir.



Şekil 2.1. Kamera hareketi ve kullanım durumlarına göre farklı video türlerinin sınıflandırılması [23].

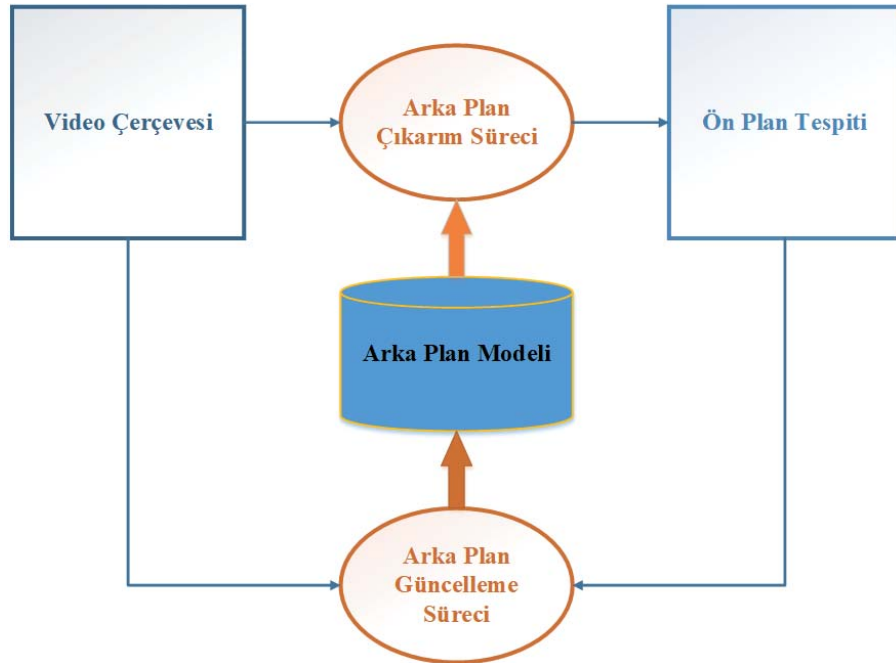
Bir video dizisinden nesne tanımaya odaklanan birçok araştırmacı tarafından gün geçtikçe yeni metodolojiler önerilmektedir. Çoğu araştırmacı, birden fazla tekniği bir arada kullanmaktadır ve mevcut yaklaşımlarda farklı metodolojiler arasında kombinasyonlar ve örtüşmeler bulunmaktadır. Bütün bunlar, tek tip bir sınıflandırma yönteminin bulunmasını oldukça

zorlaştırmaktadır. Bu bölümde, videodan hareket eden nesne tespiti için mevcut olan farklı yaklaşımlar ayrıntılı bir şekilde sınıflandırılmaktadır.

2.1.3. Hareketli Nesne Tespiti

Hareketli nesne tespiti, sınıflandırılması ve ardından tanınması, birbiriyle yakından ilişkili olan bilgisayarla görme alanındaki temel ilgi alanları olarak öne çıkmaktadır. Hareketli nesne tanıma çalışmalarında öncelikli amaç, bir arka plan sahnesinden hareket halinde olan nesneleri algılayarak bir video dizisini analiz etmektir. Sabit bir kamera ile video elde edildiği durumlarda, arka planın statik olduğu varsayılmaktadır [22].

Hareketli nesne tespiti çalışmalarında, yaygın olarak kullanılan üç farklı yaklaşım kullanılmaktadır: optik akış, zamansal fark ve arka plan çıkarma. Arka plan çıkarma yaklaşımları iki önemli adımdan oluşmaktadır. Bu yaklaşımlardan ilki temelde bir referans arka plan görüntüsünün oluşturularak sonrasında güncellenmesi yöntemini, diğeri ise mevcut görüntü ile arka plan modeli arasında en uygun ön plan görüntüsü çıkarma yöntemini kullanmışlardır. Şekil 2.2’de arka plan çıkarma sürecine ait blok diyagram gösterilmektedir. Literatürde ki çalışmalar incelendiğinde günümüze kadar çeşitli araştırmacılar tarafından bir dizi yeni yöntem önerilmiştir. Geliştirilen veya önerilen tüm yöntemler temel olarak arka plan modelinin oluşturulması yönteminde ve modeli güncellemek için kullanılan prosedürde farklılık göstermektedir [24, 25, 26].



Şekil 2.2. Arka plan çıkarma sürecine ait blok diyagram [23]

2.1.4. Hareketli Nesne Tespiti Uygulamaları

Gözetim ve güvenlik sistemlerinin kullanımının yaygınlaşması bu sistemlerde hareket eden nesnelerin videodan algılanmasının gerekliliğini ortaya çıkarmıştır. Özellikle bir güvenlik tehdidi söz konusu ise, terk edilmiş bir nesnenin (örneğin bir şüpheli çanta) izini sürmek gerektiğinde, bir otoparkta bırakılmış aracı bulmak amacıyla ya da güvenlik açısından şüpheli bir hedefi takip etmek gerektiğinde ve benzer durumlar için bu sistemler üzerinde hareketli nesnelerin tespiti oldukça önemlidir.

- İnsan bilgisayar etkileşimi sistemlerinde, bir bilgisayarla insan arasında daha doğal bir iletişim kurabilmek amacıyla insan benzeri şekilde etkileşime girebilen sistemler oluşturmak amacıyla kullanılmaktadır. Özellikle bu sistemlerde nesne tanıma amaçlı, çeşitli vücut bileşenlerini (örneğin hareket tanıma için hareket eden bir el) algılamak ve izlemek için kullanılmaktadır.
- Yürüyüş analizi yapmak da yaygın kullanıldığı alanlardan biridir. Yürüyüş analizi yapılırken bir video akışından hareket eden insan silüetini çıkarmak amaçlanmaktadır [27].
- Hareketli nesne tanımanın videodan insan etkinliği tanıma ve gerçek zamanlı nesne sınıflandırması gibi çeşitli kullanımları alanlarında bulunmaktadır [28, 29].
- Hareketli nesne tanıma, tıbbi görüntü işleme uygulamalarında olduğu gibi sanal ve artırılmış gerçeklik ile robotik vb. alanlarda da uygulama alanı bulmuştur.

2.2. Ön Plan ve Arka Plan

Nesne takibinde ön plan ve arka plan olmak üzere iki önemli kavram öne çıkmaktadır. Ön planın ne demek olduğu neyin ön planı oluşturduğu ile sahne arka planını neyin oluşturduğu sorularını yanıt bulunması, arka plan çıkarımının tanımını yapabilmek için gereklidir. Sabit bir kamera bulunması durumunda, ön plan ve arka planın tanımı açıktır; ön plan nesnesi, görünüm veya hareket açısından durağan arka plana benzemeyen herhangi bir şey olarak kabul edilmektedir. Bu tanım, sahne arka planının ne olduğunu önceden bildiğimiz veya çevrimiçi olarak tahmin edebileceğimiz varsayımına dayanmaktadır. Bu nedenle, sahne arka plan özelliklerinin istatistiksel dağılımını modellemek yeterlidir ve ön planın saptanması esas olarak bu dağılımın aykırı değerlerinin saptanması işlemidir. Sabit bir kamera ve arka plan sahnesi durumunda alternatif bir tanım şöyledir: "dikkate değer ölçüde" hareket eden herhangi bir nesne ön plandır, sabit herhangi bir nesne ise sahne arka planının bir parçasıdır. Yukarıdaki iki tanımın eşdeğer olmadığı ve ayrıştığı noktalar olduğu açıktır.

Serbestçe hareket eden bir kamera söz konusu olduğunda, ön plan ve arka planın tanımını yapmak zorlaşmaktadır. Bu durumda ön plan ve arka plan nedir sorularının cevapları daha belirsiz olmaktadır. Örneğin, hareketli bir kamera tarafından çekilen bir videodan alınan çerçevelerde

bulunan ağaç ön plan olarak kabul edilip edilmemesi önemli bir sorun olarak ortaya çıkmaktadır. Derinlik bilgisini dikkate alıyorsak ve kameraya daha yakın olan nesnelerle ilgileniyorsak, o zaman ağacı ön plan olarak değerlendirmeliyiz. Bununla birlikte, derinliğe güvenmek de belirsizdir. Bu durumda ön plan hakkında karar vermek için kullanabileceğimiz derinlik eşiği ne olacak, derinliğin düzgün bir şekilde değiştiği zemin planı ne olacak belirsizliği ortaya çıkmaktadır. Derinlikten çok hareket ele alınacak olursa bir başka sorun ortaya çıkmaktadır. Çünkü görünen görüntü hareketi (2B hareket) dikkate alındığında, sahnedeki her şey hareket etmektedir. Fiziksel hareket (3B hareket) göz önüne alındığında, ağaç, zemin ve binanın tümü aynı harekete sahip tek bir katı varlık oluşturmaktadır.

Bu arada bir diğer sorun olan şekil/zemin ayrımı sorunu, bilgisayarlı görme alanında fazlasıyla çalışılmıştır. Bilgisayarla görmede algısal gruplama sorunu birçok çalışmada çözülmeye çalışılmıştır. İçinde tek bir görüntü söz konusu olduğunda, şekil/zemin tanımı oldukça belirsiz olabilmektedir. Nesne izleme çalışmalarında bu sorunun önüne geçebilmek adına şekil/zemin belirlemede dikkate alınması gereken en önemli faktörler arasında şunlar yer almaktadır: Tamamen başka bir bölge ile çevrili olduğu görünen bir bölge, boyut, yönlendirme (dikey ve yatay bölgeler daha çok şekil olarak algılanır), kontrast (çevresi ile en büyük kontrastı olan bölge bir figür olarak algılanır), konturun simetrisi, dışbükeyliği ve paralelliği. Bu tür basit görünen özellikler literatürde ve göze çarpan bölge tespitinde yaygın olarak kullanılmaktadır. Belirli nesne şekillerinin bilgisi gibi üst düzey görüntü özellikleri, şekil/zemin işlemede de oldukça önemlidir ve sonucu büyük ölçüde etkilemektedir.

Peterson ve Gibson [30], çalışmalarında, tanıdık şekillerin şekil/zemin işlemedeki rolünün önemi üzerinde durmuşlardır. Bununla birlikte, şekil/zemin işlemenin nesne tanımadan önce gelip gelmediği ve daha sonraki tanıma sürecinin şekil/zemin sürecine geri bildirim verip vermediği veya herhangi bir şekil/zemin kararından önce şekil öncesi tanıma sürecinin var olup olmadığı durumları açık değildir. Her iki durumda da şekil/zemin ayrımının nesne tanıma ile iç içe geçen bir süreç olduğu açıktır. Bu konu, bilgisayarlı görme çalışmalarının yanı sıra nesne tanımda şekil özelliklerinin kullanıldığı birçok çalışma aracılığıyla dikkat çekmiştir.

Dinamik şekil/zemin işleme, tek görüntülü şekil/zemin işlemeye kıyasla çok daha az belirsiz süreçtir. Ortak stratejinin (aynı yön ve hızda hareket eden nesneleri gruplandırma eğilimi), dinamik sahnelerde algısal gruplama ve şekil/zemin ayrımında önemli bir rol oynadığı bu alanda yapılan çalışmalarda açıkça görülmektedir. Ortak strateji, her türlü hareketi ve ayrıca yapısal ve biyolojik hareketi içerecek şekilde genel bir gruplandırma yapılmasıdır [31].

2.2.1. Ön Plan Tespitine Karşı Arka Plan Çıkarma

Son yıllarda, birçok nesne tanıma çalışmalarına özgü "ön plan algılama" algoritması tanıtılmıştır. Yaya dedektörü, yüz dedektörü, araç dedektörü, bisiklet dedektörü, vb. gibi büyük eğitim verileri kullanan özel dedektörler tasarlanmış ve bunların öğrenerek nesneleri algılaması sağlanmaya başlanmıştır. Nesne tanıma amacıyla eğitilmiş bir sınıflandırıcı kullanıldığında, hedef nesnenin görüntüsü, görüntüyü kapsamlı bir şekilde tarayarak ve dedektörü her konum ve ölçekte uygulayarak bulunmaktadır. Bu dedektörler, görüntülerde yayaların ve araçların tespitinde başarılı bir şekilde kullanılmaktadır [32, 33, 34]. Bu algoritmalar, nesne sınırlı görünüm değişkenliğine sahip olduğunda (örneğin yaya tespiti için) en iyi şekilde çalışmaktadır. Ancak bu dedektörleri çeşitli insan pozları, şekil varyasyonları içeren videolarda ki görünüm için eğitmek oldukça zordur. Ayrıca, bu algoritmalar çok sayıda nesneyi algılamak için beklenen doğruluk oranlarını elde edememektedir. Derin evrişimli ağ mimarisindeki son gelişmeler, sınırlı hassasiyetle ve sınırlı kategori sayısı ile dedektörler oluşturmayı sağlamıştır [35]. Alternatif olarak, özellik tabanlı/parça tabanlı dedektörler, nesne pozlarının büyük değişkenliğine ve sınıf içi değişkenliğe rağmen başarılı sonuçlar elde edebilmektedir. Bu tür yaklaşımlarda hedef nesne parçaları tespit edilmektedir ve nesneleri algılamak için bu parçalar üzerinde geometrik kısıtlamalar kullanılmaktadır [36, 37].

Statik kamera videoları söz konusu olduğunda, nesneye özgü dedektörler, nesnelerin tam bölünmesini sağlayamamakla birlikte daha çok nesne ve parçalarının etrafındaki sınırlayıcı kutular ile belirlenmesini sağlamaktadır. Mobil kamera videoları söz konusu olduğunda ise, sabit kamera veya sahne varsaymadıkları için videodaki her çerçevede nesneye özgü ön plan algılama algoritmaları, kullanabildiklerinden doğrudan uygulanabilmektedirler. Ancak yaygın kullanım alanlarına ve sundukları çözümlere rağmen, bu algoritmaların videolarda karşılaşılan sorunlarda etkili olmamalarına neden olan iki temel sınırlama vardır. İlk olarak, bu algoritmalar görüntülerdeki nesneleri algılamak için tasarlanmıştır ve bu nedenle videoda herhangi bir zamansal kısıtlama kullanmazlar. İkincisi, bu algoritmaların performansı gerçek dünya uygulamalarında çoğunlukla kullanışlı olmaktan uzaktır. Hedefler boyut olarak küçük olduğunda veya bir engel arkasında kaldıklarında, performans önemli ölçüde düşmektedir [23].

Son teknoloji yaya algılama algoritmaları incelendiğinde bu tür algoritmaların, hala geliştirilmeye açık yönleri olduğu görülmektedir. Bu algoritmaların farklı nesne kategorileri için eğitilmeleri gerektiğinden, çok sayıda sınıf için algoritmalar oluşturmanın zor olması en önemli sorunlarıdır. Çalışma zamanında, her nesne sınıfı için bir dedektör kullanılması gerekmektedir [38].

2.2.2. Arka Plan Çıkarması Yaklaşımı

Arka plan çıkarması (AÇ) nesne tanımada yaygın kullanım alanına sahiptir. Arka plan çıkarması yaklaşımı bir video dizisini oluşturan her resim çerçevesinde ön plan nesnesinin

algılanması amacıyla piksellerin ön plana ya da arka plana ait olup olmadığı belirlendikten sonra sınıflandırılmasını sağlamaktadır. Nesne tanıma sürecinde, bir sahnenin genellikle arka plan modeli olarak adlandırılan bir modelle temsil edilmesi ile ön plan nesnesinin belirlenmesi amaçlanmaktadır [39].

AÇ, elde edilen görüntüyü belirlenen arka plan modeliyle karşılaştırdıktan sonra, tahmini en doğru ön plan çıkarımı için bir teknik önermektedir. Günümüze kadar, ayrıntılı matematiksel modellerle çeşitli arka plan çıkarma algoritmaları önerilmiştir ve bu algoritmalarından her birinin güçlü ve zayıf yönleri bulunmaktadır [40]. Kullanılan yöntem veya algoritma içerisinde genel yaklaşım, her video çerçevesi için seçilen modelden sapmaları tespit etmektedir. Temel aşamaları, mevcut çerçeveyi bir arka plan görüntüsünden çıkararak ve farkı bir eşikle karşılaştırarak her pikselin ön plana veya arka plana ait olup olmadığı belirlendikten sonra sınıflandırmaktır [40, 41, 42]. Bu temel yöntem daha önceki çalışmalarda çerçeve farkı olarak da adlandırılmaktaydı. Bu işlemlerin bütünü günümüzde arka plan çıkarımı olarak tanımlanmaktadır [43].

Videodan elde edilen, kademeli olarak değişen ya da sabit bir sahnede, ön plan algılama birçok farklı arka plan çıkarımı algoritması ile en uygun şekilde gerçekleştirilebilmektedir [44, 45]. Literatürdeki çalışmalar incelendiğinde bu algoritmalar da en çok istatistiksel modelleme teknikleri ya da bunların varyantları yaygın olarak kullanılmaktadır [46, 47]. Bu yöntemlerin performansını etkileyen birçok etken içerisinde en önemli olanlar dinamik sahne unsurları veya sallanan ağaçlar, kuş sürüleri, dalgalanan su, sis veya duman vb. hareketli arka planlardır [48]. Literatürde kullanılan yöntem ya da algoritmalar da kullanılan ortak arka plan çıkarma teknikleri bulunmaktadır. Bu teknikler Beneath ve ark. [44], çalışmalarında da belirtildiği gibi arka plan çıkarım yöntemleri ilkesi olarak Denklem 2.1’de gösterilmiştir:

$$\lambda_t(s) = \begin{cases} 1, & \text{if } m(I_{s,t}, B_s) > \tau_H \\ 0 & \end{cases} \quad (2.1)$$

λ_t , t zamanındaki etiketlenen hareket alanını ve s ise (x, y) koordinatında uzamsal konumun (hareket maskesi) bir fonksiyonu olarak da adlandırılmaktadır. m mesafeyi belirtmektedir ve piksel s ile B_s 'deki t zamanında aradaki mesafedir. Pixel s’de τ_H bir eşiktir. Çoğu AÇ yöntemi arasındaki temel farklılıklar, AÇ’nin ne kadar iyi modellendiği ve hangi mesafe metriğini kullandığıdır. (örn. Öklit, Manhattan, Mahalanobis vb.) [49, 50].

Cheung ve Kamath tarafından kullanılan bir arka plan çıkarma algoritmasında, dört ana aşama kullanılmaktadır [51]. Bu aşamalar ön işleme, arka plan modellemesi, ön plan algılama ve veri doğrulama olarak sıralanmaktadır. Bu çalışma da ilk adım olan ön işleme aşaması, bir giriş video dizisini sonraki işlem adımlarında kullanılmaya uygun formata dönüştüren bazı basit görüntü işleme görevlerini içermektedir [44]. Arka plan modelleme aşamasında, her video çerçevesinde bir

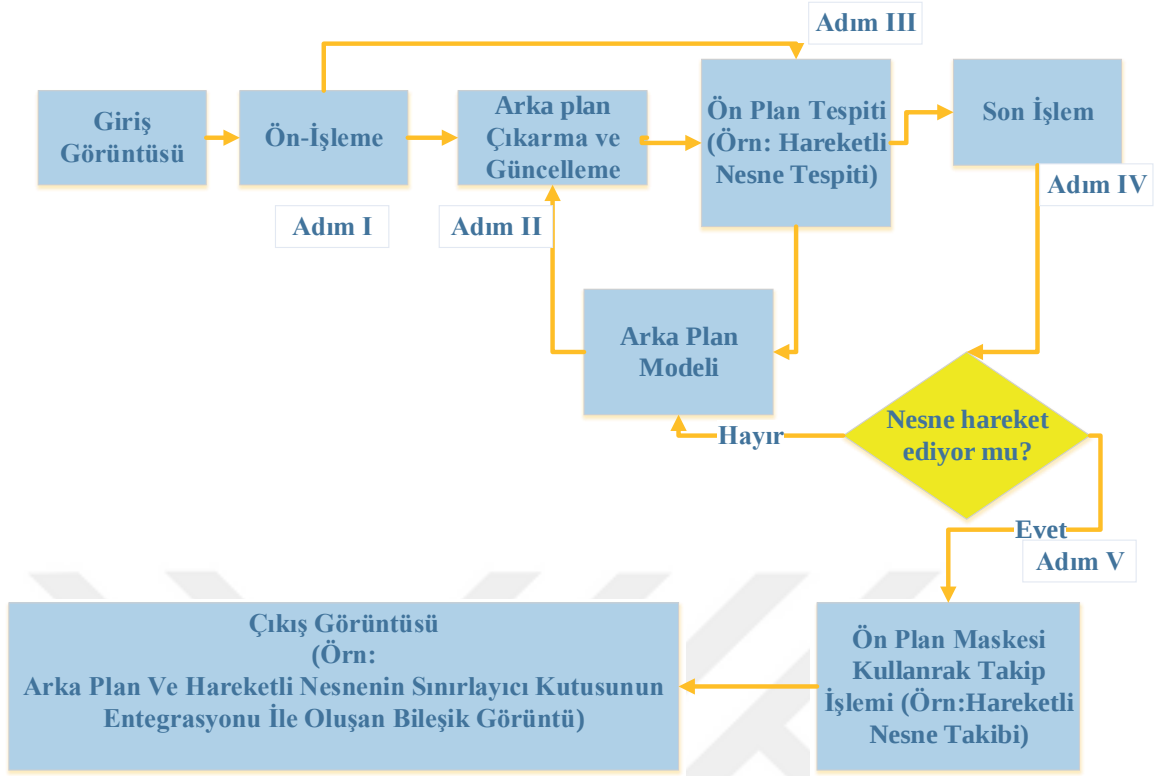
arka plan modeli hesaplama ve güncelleme işlemi gerçekleştirilmektedir. Kullanılan arka plan modeli, tüm arka plan sahnesinin istatistiksel olarak ifade edilmesini sağlamaktadır. Arka plan sahnesi statik veya dinamik olabilmektedir [42]. Ön plan algılama adımında, kullanılan arka plan modeli ile yeterince algılanamayan bazı pikseller, ikili aday ön plan maskesi olarak tanımlanmaktadır [44, 50].

Arka plan modelleme teknikleri iki geniş kategoride sınıflandırılabilir. Bunlar kayıt dışı ve özyinelemeli teknikler olarak sıralanmaktadır. Kayıt dışı teknikler, örnek çerçeve farklılaşması, medyan filtre, doğrusal tahmini filtre ve parametrik olmayan model olarak sıralanabilmektedir. Kayıt dışı tekniklerde temel olarak arka plan tahmini için kayar pencere yaklaşımı kullanılmaktadır. Bu yaklaşımda tahminin doğruluğu için önceki video çerçevelerinin bir deposu saklanmalıdır. Görüntü deposu içindeki her pikselin zamansal varyasyonuna dayanan arka plan görüntüsü tahmin edilerek belirlenmektedir. Bu teknikler, depolanan çerçevelerin dışında geçmişe bağlı olmadıkları için oldukça uyarlanabilir [50].

Özyinelemeli teknikler sıklıkla medyan filtre, kalman filtresi ve gauss karışımı yöntemlerini kullanılmaktadırlar [42]. Özyinelemeli teknikler arka plan tahmini için bir veri deposu kullanmazlar ancak her giriş video çerçevesi için tek bir arka plan modelini tekrarlı olarak güncellemektedirler. Sonuç olarak, daha önceden gelen giriş çerçevelerinin mevcut arka plan modeli üzerinde bir etkisi olabilmektedir. Cheung ve Kamath'ın çalışmasına göre, kayıt dışı tekniklerle karşılaştırıldığında, özyinelemeli teknikler daha az depolama gerektirmektedir [51]. Bununla birlikte, bu tekniklerin arka plan modelindeki herhangi bir hata, tüm sistemi daha az etkilediği için çok daha uzun süre görmezden gelinebilmektedir. Şekil 2.3'te, özyinelemeli arka plan çıkarımına dayanan genel bir nesne izleme sistemi için gösterilmektedir.

Şekil 2.3'te kullanılan arka plan çıkarımı için temel beş adım kullanılmaktadır. Bu aşamalar ön işlem, güncelleme, arka plan çıkarımı, ön plan algılama, ön plan maskesi ile izleme olarak sıralanmaktadır. Güncelleme (arka plan modelleme) adımları, Cheung ve Kamath'ın çalışmasında açıklanan ön işleme ve arka plan çıkarımı adımlarına benzemektedir. [51]. Ön işleme sonrasında ki adımda, bağlı bileşen analizi yoluyla verilen ikili görüntüde bazı düzeltmeler yapılmaktadır. Ön plan maskesi adımıyla (Şekil 2.3'deki adım V) izleme sürecinde, hareketli nesne hakkında geçici bilgiler elde edilmektedir.

Sistemin temel mantığı olarak, hareketli nesne için mekansal ve zamansal tutarlılık ile ilgilenmektedir. Belli sayıda video çerçevesi içeren bir video dizisinde tespit edilmek istenen nesne sabit olarak etiketlendikten sonra, bu hedef nesne modelin bir parçası olarak arka plan modeline gömülerek, arka plan modeli bu yeni bilgilerle hemen güncellenecektir. Nesne hareketli ise, ön plan maskesi olarak izlenebilmektedir. Bu ön plan maskesi genellikle bir ikili (yani siyah beyaz görüntü) görüntüdür. Görüntüde ki beyaz bölgeler genellikle arka plan çıkarma algoritması ile hareketli nesneler olarak ele alınmaktadır.



Şekil 2.3. Arka plan çıkarımına dayalı bir genel nesne tanıma sisteminin şeması

Literatürde, bir arka plan modelinin dinamik sahne varyasyonlarına otomatik olarak uyarlanması amacıyla birçok farklı yaklaşım önerilmektedir. Bu yaklaşımlar farklı özellikleri dikkate alınarak sınıflandırılmaktadır.

Arka plan çıkarma teknikleri kullanılan yöntem olarak genellikle modelleme ve bölümlenme olmak üzere iki farklı yaklaşıma odaklanmaktadır. Buna rağmen literatürde farklı arka plan modellerine göre, arka plan çıkarma teknikleri birkaç kategoride gruplandırılmaktadır. Bunlar arasında en sık kullanılan teknikler olasılık tahmini, bölge doku özellikleri, uzay-zamansal belirlenlik, temsili numune seçimi ve kod sözcükleri olarak sıralanmaktadır [40].

2.2.3. Temsili Örneklerle Dayalı Arka Plan Çıkarma Yöntemi

Temsili örneklerle dayalı arka plan çıkarma yönteminde, arka plan örnekleri toplanarak kullanılmaktadır. Arka plan örnekleri toplama kavramların İngilizce kısaltmalarını içeren Örnek Mutabakat (Sample Consensus -SACON) algoritmasını önermişlerdir. Bu yaklaşımda, arka plan örneklerinin örnek mutabakatı hesaplandıktan sonra piksel başına arka planın istatistiksel bir modeli tahmin edilmektedir [52]. Temel olarak her bir pikselin örnek mutabakatı hesaplandıktan sonra arka plana ait olup olmadığını belirleme işlemi gerçekleştirilmektedir. Bir olasılık dağılım fonksiyonunu doğru bir şekilde tahmin etmek yerine, belirli bir pikselin değerini ön plan veya arka

plan olarak sınıflandırmak için basit “konsensüs (mutabakat)” kavramı kullanılmıştır. Bu yöntem dört aşamalı işlemler dizisinden oluşmaktadır. Bu aşamalar Arka Plan Modeli Oluşturma, Ön Plan Çıkarma, Son İşlem ve Zaman Aşımı Haritası olarak sıralanmaktadır [53, 54].

Geleneksel örnek mutabakat algoritması, farklı arka plan modellerine uyarlanabilir olması, karmaşık arka plan sahnelerinde bile başarılı olması, hızlı bilgi işleme gücü, gürültü ile baş edebilme yeteneğindeki gücü ile öne çıkmaktadır. Kullanılan yöntemde komşu çerçeve farkından olası ön planı çıkarıldığı için çıkartılan hareketli nesnelerde bazı boşluklar bulunmaktadır, bazı boşluklar ise daha sonraki aşamada morfolojik işlemlerle doldurulsa bile bu sorun zayıf yönlerinden biri olarak ortaya çıkmaktadır [55].

Literatürde temsili örneklerle dayalı arka plan çıkarma tekniklerinden kabul edilen ve görsel arka plan çıkarıcı olarak adlandırılan bir arka plan çıkarma tekniği daha sunulmuştur. Bu teknikte her bir arka plan pikseli arka plan örnekleri koleksiyonu tarafından modellenmektedir. Bu teknik, birçok yaygın arka plan çıkarma tekniğine kıyasla hızdaki üstünlüğü ile öne çıkmaktadır. Ancak, dinamik sahneler, aydınlatma değişiklikleri ve kamufle edilmiş engel arkasında kalmış nesnelerle uğraşırken performansı kötüleşmektedir. Tekniğin eksikliklerinin üstesinden gelmek için, Droogenbroeck ve Paquot tarafından iyileştirilmiş yeni bir yöntem önerilmiştir [56]. Bu yöntem ile bazı karmaşık morfolojik işlemler ile bir önceki yöntemler desteklenmiştir. Yeni geliştirilen yöntem, dinamik arka plan sahnelerinde ve kamera titreme durumlarında elde edilen bir veri setinde daha önce geliştirilen tekniğe kıyasla daha iyi performans gösterse de, önceki tekniğin kendine özgü eksikliklerinin birçoğunun üstesinden gelememiştir. Özellikle uzamsal-zamansal bağımlılıkları modellemedeki zayıflığı ve çok modlu arka plan dağılımını tanımlamada karşılaştığı sorunlar giderilememiştir [40].

Temsili örneklerle dayalı yapılan bir diğer çalışmada Li ve ark. [55], tarafından örnek mutabakat algoritması bir yerel ikili model operatörü ile güçlendirilerek yeni bir yöntem önerilmiştir. Bu yöntem ile aydınlatma değişikliklerine dayanıklı bir yöntem oluşturmak ve arka plan noktalarının yanlış değerlendirilmesinin önüne geçebilmek amaçlanmıştır. Deneysel sonuçlar, kullanılan operatörü ve örnek konsensüs modelini bir araya getiren bu yönteminin, boşlukları etkili bir şekilde doldurabildiğini, arka plan noktalarının yanlış değerlendirilmesini önleyebildiğini, aydınlatma değişikliklerine karşı dayanıklı olduğunu ve gürültüyü de ortadan kaldırdığını göstermektedir [55].

2.2.4. Kod Çizelgesi Tabanlı Arka Plan Çıkarımı Yöntemi

Kim ve ark. [57], tarafından piksellere ait örnek bir arka plan değerini içeren kod çizelgesi kullanılmaktadır. Buna göre her pikseldeki arka plan olarak belirlenen örnek değerler, renk benzerliğine göre kendisine karşılık gelen kod çizelgesinde kümelenmektedir. Kod çizelgesi tabanlı arka plan çıkarımı yöntemi hızlı olması ve kısıtlı bellek kullanımı açısından tercih edilmektedir.

Ancak kod çizelgesinin oluşturulması aşamasında tek bir yöntem yada sabit bir strateji önerilememiştir. Guo ve ark.[58], genel detaylardan en ince detaylara kadar ayrıntılı bir ön plan tespiti yöntemi önermişlerdir. Hareket eden nesneyi kabaca konumlandırmak için blok tabanlı bir kod çizelgesinin ve hareketli nesneyi bölümlere ayırmak için piksel tabanlı bir kod çizelgesinin kullanıldığı bir strateji kullanılmıştır. Bu strateji, hesaplama maliyetini etkili bir şekilde azaltmıştır. Ancak karmaşık arka planın hızlı değişimlerini modelleyememiş ve doğru ön plan bölümlemesinde zayıf kalmıştır [40].

2.2.5. Diğer Arka Plan Çıkarım Algoritmaları

Liu ve ark. [59] ile Mahadevan ve Nuno [60], çalışmalarında harekete dayalı algısal gruplama önermişlerdir. Bu sayede belirgin değişiklikleri ve hareketleri tanımlamak mümkün olmuştur. Belirginlik haritası adı verilen yoğun matematiksel hesaplamalar içeren bir tablo oluşturmuşlardır. Bu arka plan çıkarma algoritmaları ile ön plandaki nesneler dinamik arka plandan çıkarılabilmektedir. Ancak büyük miktarda hesaplama gerektirmesi ve nesne kenarlarını doğru tespit etmekte başarılı olamaması bu algoritmaların dezavantajı olmuştur. Literatürdeki diğer çalışmalarda, hareketli nesneleri tanımak için izleme ve bağlama bağlı özellikler farklı arka plan çıkarma algoritmalarıyla birleştirilmiştir [61, 62]. Arka plan çıkarımına dair çalışmalar incelendiğinde doku özellikleriyle arka plan çıkarma işleminin, ön plandaki nesneleri dinamik arka plandan doğru bir şekilde çıkarabildiği görülmektedir. Ayrıca, özellikle dinamik arka planlarda çıkarım sonuçlarının, daha basit bir dokuya sahip olan "sabit" bir arka planla karşılaştırıldığında belirgin derecede başarısız olduğu görülmektedir [41].

2.2.6. Olasılık Tahminine Dayalı Arka Plan Çıkarımı

Karışım modelleri genel olarak karmaşık girdileri kolay ve verimli bir şekilde tanımlayabilen ve onları bilim ve mühendisliğin çeşitli alanlarındaki uygulamalar için uygun hale getiren olasılıksal modeller olarak tanımlanmaktadır. Uygulamada, bu modeller, eğitim ve tahmin için aşırı hesaplama kaynaklarına ihtiyaç duyan Destek Vektör Makineleri (DVM) ve Sinir Ağları gibi karmaşık modellerden daha iyi performans göstermektedirler. En yaygın kullanılan karışım modeli, normal dağılıma dayalı olan Gauss Karışım Modeli (GKM) olarak öne çıkmaktadır. GKM'ler literatürde oldukça araştırılan bir konu olarak popüler bir çalışma alanıdır [63].

2.2.7. Gauss Karışım Modeli

Gauss karışım modeli (GKM), çok değişkenli bir veri kümesinin değerlerini belli bir dağılımın karışımı olarak açıklamayı sağlamaktadır [64]. Gauss karışım modeli, normal dağılım

fonksiyonunun bir karışımı olarak açıklanmaktadır. Bu nedenle de "normal karışım modeli" olarak da adlandırılmaktadır [65, 66].

Gauss karışım modelinin kullanıldığı birçok farklı uygulama alanı mevcuttur. Bu model, aynı zamanda sınıflandırma ve tahmin gibi işlemlerde de kullanılabilir. Özellikle görsel hesaplama görevlerinde yaygın kullanım alanına sahiptir [67].

Arka plan çıkarımında, Gauss karışım modeli veri kümesinin tamamını yada bir bölümünü açıklamak için kullanılmaktadır. Bu model sayesinde, veri kümesi üzerinde yapılan çalışmaların sonuçları hakkında bilgi edinilebilmektedir ve bu sonuçların nasıl ortaya çıktığı anlaşılabilir [23].

Gauss karışım modelinin temeli, bir video dizisinin her çerçevesindeki piksellerin, sonlu Gauss modellerinin ağırlıklı toplamı ile karakterize edilmesine dayanmaktadır. Genellikle, Gauss modelindeki piksel sayısı ne kadar fazlaysa, elde edilen özellik o kadar eksiksizdir. Ancak Gauss modeli sayısı arttıkça hesaplama miktarı artmakta ve karmaşık bir hale gelmektedir. Çünkü elde edilen her yeni görüntü için arka plan modelinin güncellenmesi gerekmektedir. Algoritmanın hızına ve etkinliğine göre, her piksel için belli sayıda Gauss modeli tanımlanmaktadır. Gauss karışım modelinin uygulanma aşamaları üç bölümden oluşmaktadır. Bunlar model tanımı, model güncellemesi ve ön plan tespiti olarak sıralanmaktadır [23].

A. Model Tanımı

Gauss karışım modelinde, bir video çerçevesindeki (veya görüntü dizisindeki) bir pikselin renk değerleri ile ona karşılık gelen piksel işlemini oluşturmaktadır. Denklem 2.2'de bir piksel değerinin renk değerini göstermektedir.

$$\{\{X_1, X_2 \dots \dots \dots, X_t\} = \{I(x, y, i): 1 \leq i \leq t\}\} \quad (2.2)$$

$I(x, y, i)$, piksel (x,y)'nin, i. zamanındaki renk değerini temsil eder. Arka planı bir Gauss karışımıyla modelleme işlemi, piksel işleminin karışık Gauss dağılımını karşıladığı varsayımını dayanmaktadır. Diğer bir ifadeyle bir Gauss karışım modeli, her piksel için 1den K'ya tekil Gauss modelinden oluşmaktadır. Denklem 2.3, Gauss karışım modelini ifade etmektedir.

$$P(X_t) = \sum_{k=1}^k w_{k,t} n(X_t, \mu_{k,t}, \Sigma_{k,t}) \quad (2.3)$$

Burada $w_{k,t}$ k'nıncı Gauss bileşeninin t zamanındaki ağırlığını ifade etmektedir. Bu değer aynı zamanda olasılık yoğunluk fonksiyonu anlamına gelmektedir. $P(X_t)$, t zamanında X piksel

değeri tarafından gözlemlenen olasılığı temsil etmektedir. $\mu_{k,t}$ ortalama değer, $\Sigma_{k,t}$ ise kovaryans matrisidir.

Matris işlemlerinin yoğunluğu tüm süreci ağırlaştırıp hantallaştırdığı için, bu durumdan kaçınmak amacıyla tıpkı RGB renk modelinin kırmızı, yeşil ve mavi bileşenleri gibi X piksel değerinin bileşenlerinin birbirinden bağımsız olduğunu ve aynı kovaryansa sahip olduğunu varsaymak yaygın bir uygulamadır. Bu varsayım sonuçlar üzerinde çok az etkisi olmasına rağmen hesaplamayı hızlandırabilmektedir.

B. Model Güncellemesi

Gauss arka plan modellemesi, modeldeki geçerli piksel değeri ile K Gauss dağılımını eşleştirmektedir. Eğer geçerli piksel değeri K Gauss dağılımıyla eşleşmiyorsa, yeni bir Gauss dağılımı hesaplanmaktadır. Bu amaçla, en küçük ağırlık değeriyle dağılım değiştirilmekte ve yeni dağılım ortalaması geçerli piksel değeri olmaktadır. Sonuç olarak eşleşen bir Gauss dağılımı bulunursa, her dağılım için ağırlık değerleri Denklem 2.4'te ki gibi hesaplanmaktadır..

$$w_{k,t+1} = (1 - \alpha)w_{k,t} + \alpha M_{k,t+1} \quad (2.4)$$

Bu formülde σ öğrenme oranıdır ve değeri (0, 1) arasındadır. Gauss dağılımı için mevcut piksel $M_{k,t+1} = 1$ ile eşleştirilmektedir, aksi takdirde $M_{k,t+1} = 0$ olarak kabul edilmektedir. Gauss dağılımı ağırlık değerini eşleştirirken Gauss dağılımı ağırlık değeri artmaktadır. Geçerli piksel değeriyle eşleşen bir Gauss dağılımı için parametre değerleri Denklem 2.5'te ve 2.6'da gösterilen formüllere göre ayarlanmaktadır.

$$\mu_{k,t+1} = (1 - \alpha)\mu_{k,t} + \rho X_{k,t+1} \quad (2.5)$$

$$\sigma_{k,t+1}^2 = (1 - \alpha)\sigma_{k,t}^2 + \rho(X_{k,t+1} - \mu_{k,t+1})^T(X_{k,t+1} - \mu_{k,t+1}) \quad (2.6)$$

Burada ρ diğer öğrenme oranıdır, değeri Denklem 2.7'de gösterilen formülle hesaplanmaktadır. Eşleşmeyen Gauss dağılımı için parametreleri değişmeden kalır.

$$\rho = \alpha \eta(X_k | \mu_k, \sigma_k) \quad (2.7)$$

C. Ön plan Tespiti

Bir önceki aşama olan model güncelleme aşamasına göre, daha küçük kovaryans ve daha büyük ağırlığa sahip Gauss dağılımının, arka plan piksellerinin dağılımı olma olasılığı daha yüksek kabul edilmektedir. Bu nedenle, belirli bir arka plan modelini belirlemek için, görüntüdeki her piksel için K Gauss dağılımları hesaplanmaktadır. İlk B Gauss dağılımları için arka planın tanımlaması Denklem 2.8’de gösterilmiştir. Yeni pikseller işlenirken, bu piksellerin arka plana dahil olup olmadığı, eşleştiği Gauss bileşeninin ağırlığına göre aşağıdaki şekilde belirlenmektedir:

$$B = \operatorname{argmin}_b \left(\sum_{k=1}^b w_{k,y} > T \right) \quad (2.8)$$

Bu formüle göre, T arka plan modelinin önerilen eşik değeridir. Toplamları T eşik değerini geçen en yüksek B ağırlığa sahip olan bileşenler, arka plan bileşenleri olarak kabul edilmektedir. T değeri küçükse, Gauss karışım modeli tek bir Gauss dağılım modeline dönüşecektir. T değeri büyükse, sallanan yapraklar ve dalgalanan göller gibi karmaşık dinamik bir arka plan pikseli olarak düşünülmektedir. Arka planda açıklanan B Gauss dağılımlarında en az bir Gauss dağılımı geçerli piksel değeriyle eşleşiyorsa, geçerli piksel bir arka plan pikselidir aksi takdirde ön plan pikseli olarak tanımlanmaktadır. Diğer bir ifadeyle elde edilen piksel değeri, bu arka plan bileşenlerinden bir tanesi ile eşleşiyorsa, o pikselin arka plan görüntüsüne ait olduğu belirlenmektedir. Tüm piksellere uygulanacak bu yöntem ile arka plan ve dolayısıyla ön plan nesneleri tespit edilmiş olmaktadır.

Bir görüntü dizisinde ön plan nesnesi olarak kabul edilen bir hareketli nesnenin arka plan çerçeve görüntüsünden ayrılması, en önemli adımdır. Gri tonlamalı veya renkli bir video çerçevesinde bir arka plan modeliyle karşılaştırılarak ön plan tespitinin yapılabilmesi, ön plan algılama sistemi nesnesi kullanılarak yapılır. Ardından belirlenen ön plana yada arka plana ait bireysel piksellere bakılarak etkili bir şekilde bir ön plan maskesi hesaplanmaktadır. GKM ifadesi, piksellerin durumunun bir kareden diğerine değişimine işaret eden K Gauss dağılımlarının bir karışımıdır. GKM’deki her piksel, RGB renk uzayındaki yoğunluğu ile karakterize edilmektedir.

GKM uygulaması ilk aşama olarak renkli görüntüyü ikili bir görüntüye dönüştürerek işleme almaktadır. İkili görüntüde herhangi bir değişime uğramayan pikseller 0 yani siyah, buna karşılık değişime uğrayan pikseller 1 yani beyaz olarak tanımlanmaktadır. Bu sayede, bir video çerçevesinden ön plan nesnesi olarak etiketlenen hareketli nesneler bölümlere ayrılarak ve beyaz olarak örneklendirilebilmektedir. Şekil 2.4’te, bir video çerçevesine GKM uygulanmasının sonucu gösterilmektedir. Sabit olan pikseller siyaha (0) dönüşmüştür ve değişikliğe uğrayan pikseller, ön plandaki nesneyi açıkça belirten 1 (beyaz) olarak tanımlanmıştır. Bu sayede elde edilen çerçevede arabalar ön plan nesnesi olarak algılanmıştır.



Şekil 2.4. Bir video çerçevesinde GKM uygulaması (a) GKM uygulamasından önce, (b) GKM uygulamasından sonra

2.3. Nesne Takibi

Videoda nesne tespiti, bir dizi görüntü çerçevesinde bir nesnenin varlığının doğrulanmasını içermektedir. Video işleme alanında bir diğer önemli konu ise nesnelerin tanınması için konumlandırılmasıdır ve bu işlem nesne izleme olarak tanımlanmaktadır. Literatürde çok çeşitli nesne tanıma ve takip uygulamaları bulunmaktadır. Bu uygulamaların en çok kullanıldığı alanlar ise video gözetimi, görüş tabanlı kontrol, video sıkıştırma, insan-bilgisayar arayüzleri, robotik vb. olarak sıralanabilmektedir [22].

İç ve dış ortamlarda özellikle gözetim amaçlı hareketli nesnelerin video dizilerinde izlenmesi karşılaşılan tüm zorluklara rağmen giderek önem kazanmaktadır. Yeni nesil bilgisayar sistemlerinin ve yüksek kaliteli kameraların geliştirilmesi ve dikkatli izleme gerekliliği, nesne tanıma ve takip algoritmalarının geliştirilmesi ve uygulanması gerekliliğini ortaya çıkarmaktadır. Bu algoritmalarda yeni yaklaşımlar önem kazanmaktadır. Temel işleyiş olarak ön plandaki nesneler birbirini izleyen çerçeveler boyunca takip edilmekte ve hareket eden nesneler algılanmaktadır. Ayrıca, nesnenin davranışı da ardışık olarak tanımlanmakta ve analiz edilmektedir. Literatürde çeşitli nesne izleme yöntemleri gösterilmektedir. Akıllı nesne takibi için mevcut genel yaklaşımlardan bazıları, özellik tabanlı nesne izleme, bölge tabanlı nesne izleme, kontur tabanlı nesne izleme ve model tabanlı nesne izleme olarak sıralanmaktadır. Genel olarak, seçilen takip yaklaşımının belirlenmesi nesne temsiline, izleyici özelliklerine ve görüntü özelliği seçimine bağlıdır. İzleyiciler ve görüntü özellikleri, video ortamına, nesnenin görünümüne, silüetine, nesne sayısına ve gözetleme kamerasının konumu ve hareketine bağlıdır. Takip edilecek nesnenin şekli dikdörtgen veya elips gibi belirli geometrik şekilde ise nesneler, nokta ve ilkel geometrik tabanlı yaklaşımlarla daha iyi temsil edilebilmektedir. İzleme hedefleri iskelet gibi karmaşık yapıda ise parça tabanlı izleme modelleri tercih edilmektedir. Günümüzde yaygın olarak kullanılan nesne izleme yöntemleri aşağıda özetlenmiştir [22].

2.3.1. Özelliğe Dayalı İzleme

Nesne öğelerine dayalı olarak, özellik tabanlı izleme yöntemi, nesne özelliklerini nesneden çıkararak izlemektedir. Ayrıca, elde edilen görüntüler arasındaki öğelerde eşleştirilmektedir.

2.3.2. Kontur Tabanlı İzleme

İzleme konturları, takip edilecek nesnenin şeklini belirlemek için sınırlarını analiz etmekte ve ardından bu sınırları zaman içinde izlemektedir. Bu yöntemde otomatik izleme zorlaşmaktadır. Dokladal ve ark. [68], tarafından etkin bir kontur tabanlı izleme algoritması geliştirilmiştir. Bu algortmada bir görüntü bölümlere ayrılmaktadır ve bunun için gradyan hesaplanmaktadır. Gradyan görüntüyü izlemek için kullanılmaktadır. Chen ve ark. [69], tarafından sunulan kontur modeli sinirsel bulanık ağılara dayanmaktadır. Eğitim ve tanıma için sinirsel bulanık çıkarım ağıları oluşturulmaktadır. Zhou ve ark. [70], tarafından geliştirilen yöntemde partikül filtresi kullanılmaktadır. Renk özelliklerini kontur bilgisi ile entegre etmek önerilen yöntemdir. Kontur tespiti için Sobel operatörü ve silüet benzerliği kullanılmıştır. Lin ve ark. [62], göre renkli bir histogram ve çoklu füzyon yaklaşımı kullanılarak bir nesnenin yeri belirlenebilmektedir. Hareket eden nesneleri izlemek amacıyla konturların çıkarılması için bölge tabanlı bir yöntem kullanılabilir. Özellik birleştirme, parçacık filtreleri için Harris köşeleri kullanılarak gerçekleştirilmektedir. Hu ve ark. [71], göre önerilen yöntemde uyarlanabilir bir silüet kontur evrimi, renk tabanlı kontur evrimi ve Markov tabanlı bir dinamik model kullanılmıştır.

2.3.3. Model Bazlı İzleme

Videoda elde edilen sahne bilgilerini bir modelle eşleştirmek, model tabanlı izleme tekniğinin temel yapısını oluşturmaktadır. Modeller, nesne hakkındaki ön bilgilere dayanarak oluşturulmaktadır. Genellikle eşleştirilen modellere dayalı elle veya bilgisayar görüşü ile elde edilmişlerdir. Ramanan ve ark. [72], vücut parçalarını izleyerek insanları tanımlayan ve takip eden bir 2B model geliştirmiştir. Model, insanların vücut kısımlarını içeren çerçeve setinden öğrendiği gibi, insanların görünüşlerini modellemeyi amaçlamaktadır. Vücut parçaları birbirini takip eden her karede izlenmektedir. Zhao ve ark. [73], çoklu nesneler için yeni bir izleme yöntemi önermişlerdir. Kamera yardımıyla karmaşık bir ortamda böyle bir algoritma ile birkaç nesne izlenebilmektedir. Global ve hareket faktörleri, insanların tam hareketini ayırmak için kullanılabilir. İzleme amacıyla, bu yöntem farklı bölgelere göre 3B modeller ile birleştirilerek geliştirilebilmektedir.

2.3.4. Kalman Filtreleme

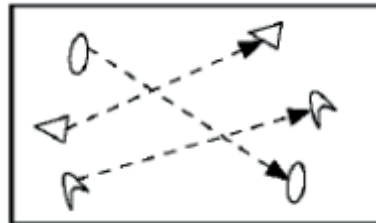
Kalman filtresi, bir analiz aracı ve veri işleme algoritması ile beklenen değerleri hesaplama amacıyla kullanılmaktadır. Nesnenin gerçek, konumunu ve hızını etkili bir şekilde tahmin etmek

için bir dizi denklem ve veri girişi kullanan matematiksel bir süreçtir. Modellenen sistemin kesin yapısı bilinmese bile, modellerin geçmiş ve şimdiki durumların yanı sıra gelecekteki durumları da tahmin edebilmektedir. Chen [74], Kalman filtrelerine dayalı olarak nesneleri izlemek için birleştirici bir yaklaşım önermiştir. Buna göre nesne konumları tahmin edilmektedir ve izleme doğruluğunu arttırabilmek için çekirdek tabanlı bir izleme kullanılmaktadır. Nesne konumu belirlendikten sonra kontur tabanlı bir izleme kullanılmaktadır. Wu ve ark. [75], üç boyutlu bir izleme modeli önermişlerdir. Bu yöntem, karmaşık ortamlarda nesne hareket yönünü çıkarma yeteneğine sahiptir. İzleme doğruluğunu arttıran bu yöntemle geliştirilmiş bir Kalman filtresi kullanılmıştır. Li ve ark. [55], birden fazla teknik içeren birleşik bir nesne izleme yöntemi önermişlerdir. Bu yöntemle hareketli nesnelerin köşeleri ile temsil edilmesi sağlanmıştır. Ardışık çerçevelerde nesnenin köşe noktası değişimlerine bakılarak Kalman filtresinin parametreleri tahmin edilebilmektedir. Kalman filtreleri, temel mantık olarak sonraki her karede olması beklenen noktanın konumunu tahmin etmektir. Komşuları kaydetmekten ise yerel arama sorumludur.

Literatürde farklı kategorilendirilmiş nesne takip yöntemlerine rastlamak mümkündür. Bu yöntemler genellikle nesnenin şeklinin temsil edilmesi, takip edilen nesne sayısı ve nesne hareketinin tahminini içeren yöntemler olarak farklılık göstermektedir. En yaygın nesne takip yöntemi kategorisi içerisinde yer alan takip yöntemleri nokta takibi, çekirdek takibi ve silüet takibi yöntemleri olarak sıralanmaktadır. Bunlar tek başına yada birlikte kullanılabilir [76].

2.3.5. Nokta Takibi

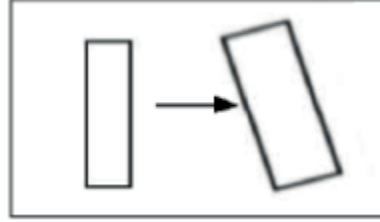
Bir görüntü dizisinde nesneler noktalar ile temsil edilerek takip edilmektedir. Takip edilecek noktalar topluluğu, nesnenin daha önceki durumuna dayanan nesne konumunu ve hareketini kapsamaktadır. Noktalar çoğunlukla nesnenin merkezi yada köşelerinden ibarettir. Nokta takibi yöntemi görüntü dizisinde takip edilecek nesneyi bulmak için ek işlemlere ihtiyaç duymaktadır. Nokta takibi hedefin kaybolduğu, bir başka nesne arkasında kaldığı ve sahneye giriş çıkış olduğu durumlarda yetersiz kalmaktadır. Şekil 2.5'te nokta takibi gösterilmektedir.



Şekil 2.5. Nokta takibi [76]

2.3.6. Çekirdek Takibi

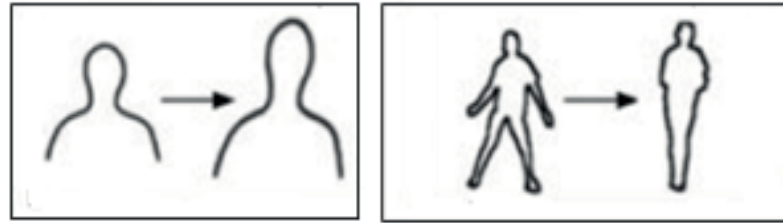
Çekirdek takibi yönteminde, nesnenin geometrik şekil ile temsil edilmektedir. Bu yöntem ile nesne takibi, görüntü dizisinde ki çekirdeğin hareketinin hesaplanması ile yapılmaktadır. Bu hareket genellikle öteleme, dönme veya doğrusal hareket etme şeklinde olabilmektedir. Şekil 2.6’da çekirdek takibi yapılırken dönme hareketi gösterilmektedir.



Şekil 2.6. Çekirdek takibi dönme hareketi [76]

2.3.7. Silüet Takibi

Takip edilecek nesnenin karmaşık bir şekle sahip olması durumunda tercih edilmektedir. Nesnenin sınır çizgilerinin temsil ettiği bölgenin nesne olarak kabul edilmesi şeklinde uygulanmaktadır. İnsan bedeninin temsilde sıklıkla kullanılmaktadır. Bu yöntem karmaşık yapıdaki nesnelerin takibinde başarılı şekil tanımlamaları oluşturduğu için kullanımları uygundur. Şekil 2.7’de silüet takibinde çevre çizgisinin belirlenme aşaması gösterilmektedir.



Şekil 2.7. Çevre çizgisi oluşturma [77]

Silüet tabanlı nesne takibi, görüntü dizisinde ki bir önceki görüntüde elde edilmiş modelin kullanılmasıyla nesne bölgelerini belirlemektedir. Bu model nesne sınır çizgisi içerisinde bulunan renk histogramı, nesne kenarları veya nesne sınır çizgisi bilgilerini kullanmaktadır. Silüet takibi, şekil eşleştirme ve çevre çizgisi takibi şeklinde iki farklı yöntem kullanılmaktadır. Şekil eşleştirme yöntemi, var olan görüntüde ki nesnenin silüetini modellemektedir. Çevre çizgisi yöntemi ise nesnenin görüntü dizisinde ki bir önceki görüntüsüyle o anki görüntüsünü eşleştirerek sınır çizgisini

belirlemektedir. Bu yöntem de durum uzay modelleri ya da enerji fonksiyonlarının minimizasyonu kullanılmaktadır.

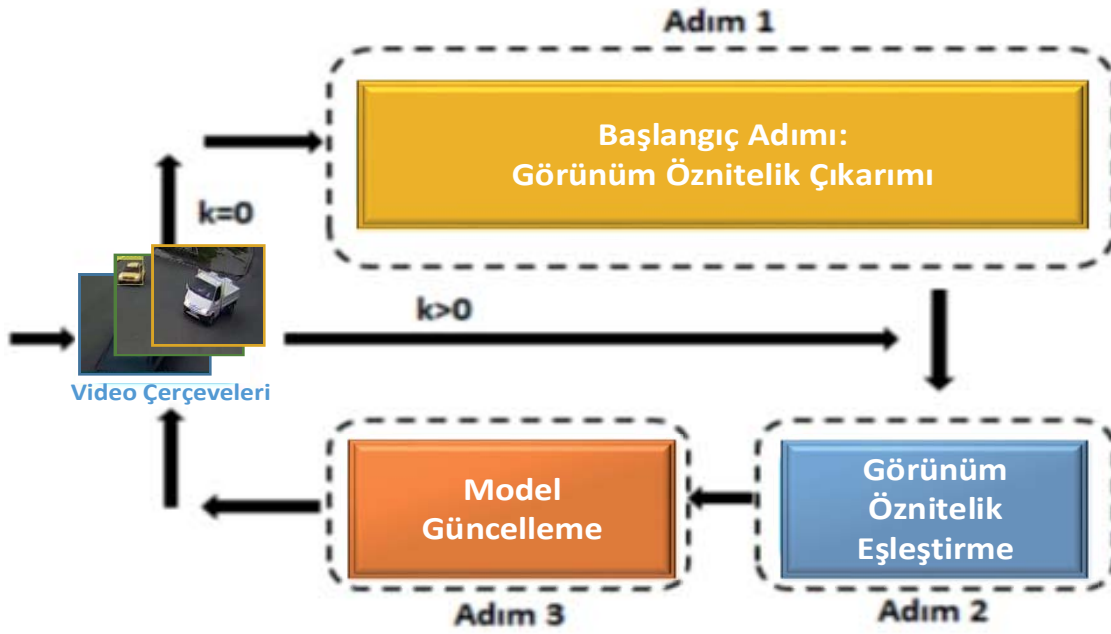
2.3.8. Video Görüntülerinde Hedef Takibi

Videoda görüntü işleme çalışmalarından en önemli aşamalardan biri hedef takibidir. Hedef takibinin önemi, video kayıt sistemlerinin artması ve teknolojinin gelişmesine paralel olarak gün geçtikçe artmaktadır. Hedef takibine, günümüzde birçok alanda karşımıza çıkan görsel takip sistemlerinde, görüntü/video işlemede, bilgisayar-insan etkileşiminde, araç takibi gibi konularda ihtiyaç duyulmaktadır Video görüntüleri üzerinde hedef takibi yapılan çalışmalar incelendiğinde temel olarak iki aşamanın önem kazandığı görülmektedir. Bu aşamalardan ilki nesnenin ilk konumunun sahne içerisindeki tespiti, ikinci aşama ise video çerçeveleri içerisinde elde edilen hedef görüntüler arasında ilişkili verilerin elde edilmesidir. Bu aşamaları birlikte gerçekleştiren nesne takip yöntemleri takip edilecek hedefin bir sonraki durumunu tahmin etmeyi amaçlamaktadır. Bu aşamaları ayrı ayrı ele alan nesne takip yöntemleri ise bölütleme ve nesne yakalama algoritmalarıyla hedefleri belirlemekte ve daha sonra bunlar arasında veri bağı ilişkileri kurmaktadır [78].

Hedef takibi alanında birçok gelişme yaşanmasına rağmen hala çözülemeyen birçok sorun mevcuttur. Hedef takibinde karşılaşılan en büyük sorunlardan biri hedefin zamana göre ani hareket gösterebilmesi olmaktadır. Hedef zamana göre düzgün bir hareket değişimi gösterse bile farklı yönlerde gidebilmektedir. Bununla birlikte, takip edilen hedef nesne görünümü zamanla değişime uğrayabilmektedir. Uygulanan çalışmada hedef görünüm modelinin güncellenmediğinde hedef başarılı bir şekilde takip edilemeyecektir. Hedef takibinde karşılaşılan diğer sorunlar hedef takibinde ışık/aydınlık değişimi, gürültü faktörü, kamera hareketi, kameranın görüş açısı, hedef nesnenin bir başka nesnenin arkasında kalma ve gizlenme durumu, bir başka nesne ile yarı/tam örtüşme durumu, arka plan karmaşıklığı gibi etkenler, hedef takibini zorlaştırmaktadır. Nesnelerin arka plandan ve birbirinden farklı şekillere ve farklı biçimsel özelliklere sahip olması hedef takibinde kolaylık sağlamaktadır.

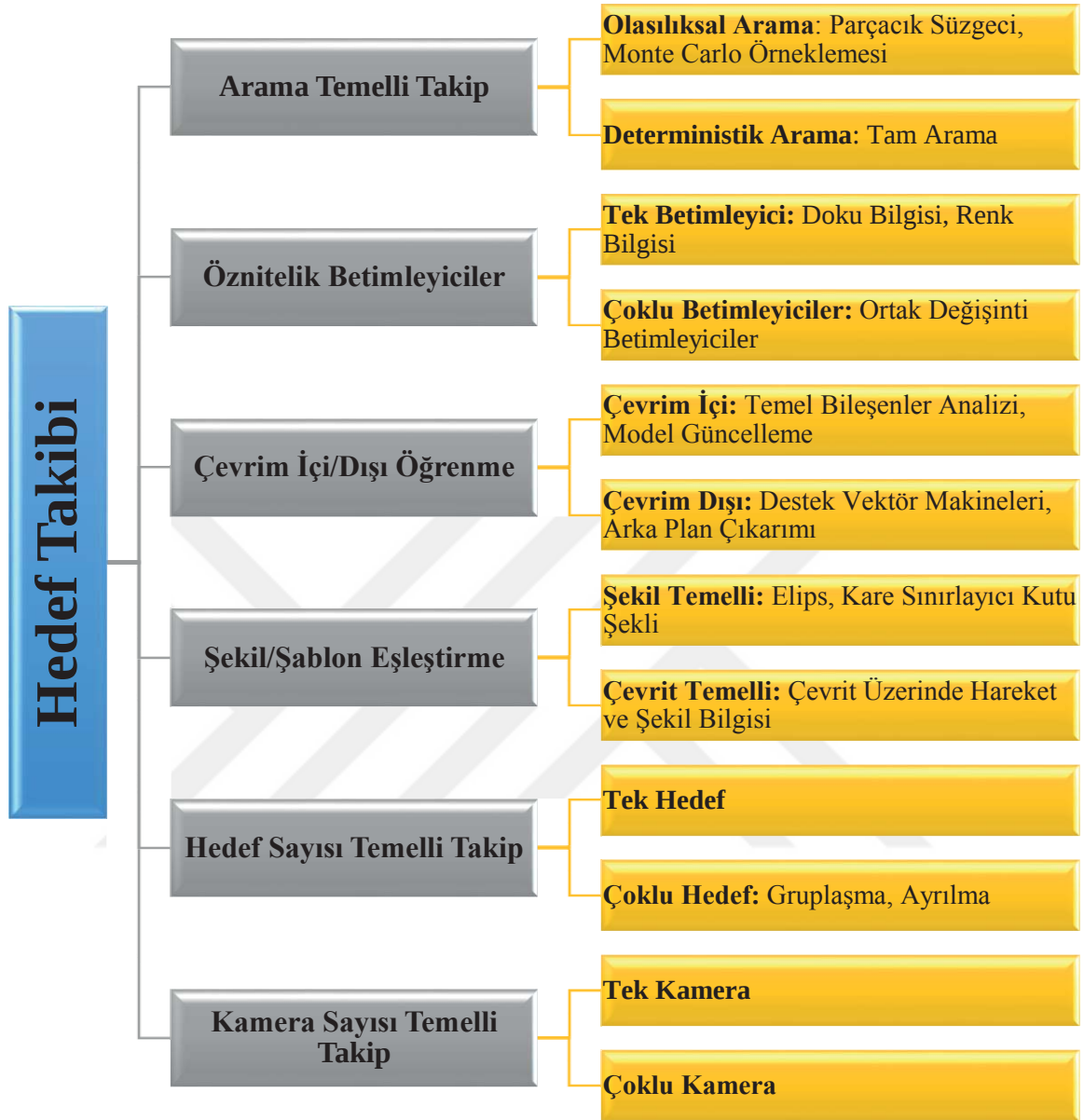
Video görüntüleri üzerinde hedef takibi çalışmalarında yüksek başarımların sağlanabilmesi için bazı gereksinimlere ihtiyaç duyulmaktadır. Bunlardan en önemlisi gürbüz bir hedef takip sistemi tasarlamaktır. Gürbüz bir hedef takip sistemi, takip performansını düşürecek bozucu etkenlere karşı dayanıklı ya da diğer bir deyişle duyarsız olmalıdır. Verimli bir hedef takip sisteminin bir diğer gereksinimi ise uyarlanabilirliktir. Hedef takibinde kullanılan hedef görünüm modeli, zamana göre değişime uğrayan hedef görünümüne paralel bir uyum sağlamalıdır. Gürbüz, uyarlanabilir bir hedef takip sistemi, düşük hesapsal yüke sahip olmalıdır. Diğer bir ifadeyle, takip sisteminin gerçek zamana yakın bir performans sergilemesi gerekmektedir.

Hedef takip sisteminin yapısı, genel olarak Şekil 2.8’de gösterildiği gibi 3 aşamada gösterilebilmektedir. İlk olarak takip edilecek nesneye ait görünüm özellik çıkarımı yapılmaktadır. Takip edilecek nesneler, arka plan çıkarım yaklaşımları gibi yöntemlerle otomatik olarak belirleneceği gibi, el yardımıyla da belirlenebilmektedir. Bir sonraki aşamada hedefe ait görünüm modeli oluşturulmaktadır. İkinci aşamada ise, hedefe ait görünüm modeli, aday hedef görünüm bölgeleri ile eşleştirilerek en uygun aday hedef görünümü belirlenmektedir. En son olarak da, hedef nesnenin görünüm modeli, en uygun aday hedefin görünümüne bağlı olarak güncellenmektedir.



Şekil 2.8. Hedef takip sisteminin genel yapısı [78]

Hedef takibinde nesneye ait karakteristik özelliklerin belirlenmesi, sistemin başarımına doğrudan etki etmektedir. Takip edilecek hedefe ait öznitelik çıkartımı, o nesneye ait özellik tanımlayıcısı olmaktadır. Hedefin renk veya gradyan bilgilerini kullanarak hedefe ait öznitelik çıkarımı yapmak mümkündür. Literatürde hedef takibi yaklaşımları, farklı kategoriler altında değerlendirilmektedir. Bununla birlikte hedef takip yaklaşımlarından en çok kullanılanları Şekil 2.9’da gösterilmiştir. Hedefin hangi yöntemle aranacağı, hedefin hangi karakteristik özelliklerinin betimleneceği, takip edilen hedef sayısı, hedef nesneleri gözlemleyen kamera sayısı, takipte çevrim içi/dışı öğrenme teknikleri, hedef takip yaklaşımlarının başlıca çalışma konularını oluşturmaktadır [34].



Şekil 2.9. Hedef takip yaklaşımları [78]

Literatürde hedef tespiti ve takibi ile ilgili farklı yaklaşımlar kullanılan birçok çalışma yapılmıştır. Kullanılan kamera sayısına bağlı olarak tek bir kamera veya çoklu kamera ile kameraların aktif ya da pasif olmasına göre farklı hedef takip çalışmaları bulunmaktadır. Kullanılan aktif kameraların, sağa sola, yukarı aşağı ve yaklaşma özelliğine sahip olması, aktif kameralar ile yapılan hedef takibinin başarımının artırılması amaçlanmıştır [79, 80]. Çalışmaların durumuna bağlı olarak aktif kameralar yerine pasif kameraların tercih edildiği hedef takip sistemleri günümüzde daha çok kullanılmaktadır [81, 82] .

2.3.9. Blob (Binary Large Object- İkili Büyük Nesne) Analizi

Blob analiz, hedef takip sistemlerinde yaygın olarak kullanılmakla beraber önemli bir yere sahiptir. Blob analiz, bir görüntü dizisinden renk dağılımını kullanarak hedef bölgeleri çıkarmaktadır. Blob, ikili büyük nesne olarak ifade edilmektedir [83].

Özellikle ikili görüntülerde bir çerçeve içerisinde birbirlerine bağlı piksel bölgelerini tanımlamak amacıyla kullanılmaktadır. Görüntü işleme uygulamalarında ikili görüntülerde gürültüden arındırılmış ön plan nesnelerini diğerlerinden ayırarak etiketleyip, öznelik verileri üretmektedir. İlk aşamada görüntüdeki nesneler ayrılır, ikinci aşamada ise hedef nesne bunlar içerisinde belirlenerek işaretlenmektedir. Diğer bir ifadeyle önce Blob'lar ayıklanmakta ve sonra sınıflandırılmaktadır. Şekil 2.10'da Blob analiz modülü gösterilmiştir.



Şekil 2.10. Blob analiz modülü [85]

Blob analiz, görüntüde ki büyük alanları nesne ya da hedef olarak kabul etmektedir. Görüntüde ki küçük alanları ise gürültü kabul ederek değerlendirmeye alınmamasını sağlamaktadır. Büyük ifadesi nesnenin belirli bir boyutta olması anlamında kullanılmaktadır. Kullanım durumuna göre büyüklük değişebilmektedir. Blob analiz kullanılarak video çerçeveleri arasında ki geçişlerde, arka plan çıkarımı ile elde edilen ön plan görüntülerinde oluşan gürültülerin çıkarılması sağlanmaktadır [83, 84].

Bu sayede sadece hedefe ait ön plan görüntüsü dikdörtgen bir sınırlayıcı kutu içerisine alınarak hedef takip işlemi gerçekleştirilmektedir. Blob'ları çerçeveleyen sınırlayıcı kutu oranı, bu kutunun yüksekliğinin genişliğine bölümü olarak ifade edilmektedir. Bu sayede sınırlayıcı kutunun yüksek, uzun vs. gibi boyutunu belirlemeyi sağlamaktadır [84].

Bir Blob'un kompaktlığı diğer bir ifadeyle nesnenin piksellerinin sıklığı ya da seyrekliği Blob alanının sınırlayıcı kutunun alanına oranı olarak tanımlanmaktadır. Blob'ların ayıklanması amacıyla kullanılan bu formül Denklem 2.9'da gösterilmektedir. Bu formül ile örneğin yumruk şeklindeki bir el ile parmakları açık bir el birbirinde ayrılmaktadır.

$$\text{Kompaktlık} = \frac{\text{Blob Bölgesi}}{\text{genişlik} \times \text{yükseklik}} \quad (2.9)$$

Bir nesnenin ağırlık merkezi ya da kütle merkezi nesnenin denge merkezidir. İkili görüntüye sahip bir nesne içinde ortalama x ve y konumlarıdır. Blob'daki piksellerin x koordinatlarının veya y koordinatlarının toplamının toplam piksel sayısına bölünmesiyle x_c, y_c değerleri hesaplanmaktadır. Denklem 2.10' da bulunan formülle hesaplanmaktadır.

$$x_c = \frac{1}{N} \sum_{i=1}^N x_i, \quad y_c = \frac{1}{N} \sum_{i=1}^N y_i \quad (2.10)$$

N, Blob'daki piksel sayısını ifade etmektedir. x_i ve y_i , N pikselin x ve y koordinatlarını tanımlamaktadır.

Görüntü işlemede kullanılan algoritmalar, pikselleri değerlerine göre ayırmakta ve bunları ön plan veya arka plan olmak üzere iki kategoriden birine yerleştirmektedir. Blob analiz ise bu görüntülerde oluşan birbirine bağlı piksellerin oluşturduğu ön planlar hakkında işlem yapmaya yardımcı olmaktadır [85].

Piksellerin bağlı olması, piksellerin komşu olması anlamına gelmektedir. Yaygın olarak kullanılan bağlantı türleri 8'li ve 4'lü bağlantılardır. 8'li bağlantı daha doğru bilgi elde edilmesini sağlamasına rağmen daha az hesaplama ihtiyacı duyulduğundan ve hızlı olduğundan dolayı 4'lü bağlantı sıklıkla tercih edilmektedir [86].

2.4. Derin Öğrenme Yöntemleri Kullanarak Nesne Tespiti

Derin öğrenme insan kontrolüne gerek kalmadan büyük miktardaki veriler üzerinde işlem yapabilme yeteneğine sahip yapay zekanın bir alt alanıdır. Derin öğrenme, ilk ortaya çıktığında ölçeklenebilirlik ve büyük bilgi işlem gücü gerektirmesi gibi birçok etken nedeniyle beklenen seviyede ilgi çekmemiştir. Ancak 2006'dan sonra, iki ana nedenden dolayı modern makine öğrenimi algoritmalarına oranla daha popüler hale gelmeye başlamıştır. Bu nedenler:

- Günden güne işleme için artan veri miktarlarının varlığı
- Üst düzey hesaplama kaynaklarının varlığı

Beynin yapısından esinlenerek geliştirilen çok katmanlı derin öğrenme mimarileri günümüzde geniş bir uygulama alanına sahiptir [87]. Derin öğrenmenin kullanıldığı çeşitli alanlar şöyle sıralanabilir: konuşma tanıma, nesne tanıma, karakter tanıma, izinsiz giriş çıkış algılama, metin sınıflandırma, gen çalışmaları, biyolojik veri işleme, veri madenciliği, video işleme, görüntü işleme, doğal dil işleme, robotik vb.

Derin öğrenme algoritmaları, bu alanlarda elde edilen verileri işleyerek sonuçları daha doğru bir şekilde tahmin edebilir. Derin öğrenmenin en popüler mimarilerinden biri, ESA olarak adlandırılan bir tür evrişimli sinir yapay sinir ağıdır [88].

2.4.1. Evrişimli Sinir Ağları (ESA) Tanımı

Evrişimli sinir ağları, görüntü ve ses gibi işlenmesi gereken veri türleri üzerinde çalışmak için tasarlanmış bir yapay sinir ağları çeşididir. ESA'lar girdi verilerinin evrişim tabanlı bir şekilde işlenmesi için tasarlanmıştır. Bu durum her bir veri ögesinin, bir matris biçiminde ifade edildiği ve matrislerin birbirleriyle çarpılarak işlemlerin gerçekleştirildiği anlamına gelir [89].

ESA 'ların Çalışma Prensibi

ESA'lar, girdi verilerini evrişim adı verilen matris işlemleriyle işlemektedirler. ESA'lar, girdi verilerini (genellikle görüntüler) bölümlere ayırır ve her bir bölümü özel bir matris (filtre) ile işlemektedir. Bu işlem ile bir filtre girdi verisi üzerinde hareket ettirilmektedir. Bu işlem sonucunda, filtrenin çıktısı yeni bir matris olacaktır. Bu matris, daha sonra başka bir filtre için tekrar işleme tabi tutulabilmektedir. Bu filtreler, görüntülerdeki özellikleri çıkarmak için tasarlanmıştır. Örneğin, bir filtre, görüntünün kenarlarını veya köşelerini tespit etmek için kullanılabilir. ESA'lar, her katmanında filtreler kullanır ve her bir katmanın çıkışı, bir sonraki katmanın girdisi olur. Bu süreç, ağı son katmanına kadar devam etmekte ve sonuçta görüntülerdeki desenlerin tanınması ve sınıflandırılması sağlanmaktadır. ESA'lar, özellikle büyük veri kümelerindeki desenleri tanımlama ve sınıflama konusunda son derece başarılıdır. Bu nedenle görüntü tanıma, nesne takibi ve tespiti ile yüz tanıma gibi alanlarda yaygın olarak kullanılmaktadır [90].

ESA'ların Etkililiği

ESA'ların en büyük avantajlarından biri, girdi verilerinin herhangi bir boyutta olabilmesidir. Bu, herhangi bir boyuttaki görüntü veya veri seti için kullanılabilir oldukları anlamına gelmektedir. Ayrıca, ESA'lar evrişim işlemi ile verilerin özelliklerini daha kolay çıkarmaktadırlar. Çıkarılan özellikler, daha sonra sınıflandırma ve tanıma işlemlerinde kullanılmaktadır [91].

ESA'ların, diğer makine öğrenimi tekniklerine göre önemli bazı avantajlara sahiptir:

- 1. Veri Küçültme:** ESA'lar girdi verilerini bölümlere ayırarak, veri boyutunu azaltmaktadır. Bu, özellikle büyük veri kümelerinde veri işlemeyi daha hızlı hale getirmektedir.
- 2. Desen Tanıma:** ESA'lar, filtreleri kullanarak girdi verilerindeki desenleri tanımlayabilmektedir. Bu, özellikle görüntü tanıma ve işleme gibi alanlarda çok yararlıdır.

3. Transfer Öğrenme: Önceden eğitilmiş bir ESA modeli kullanmak, yeni veri kümesi için eğitim süresini azaltabilmektedir. Bu, özellikle sınırlı veri kümelerinde yararlıdır.

4. Paralel İşlem: ESA'lar, büyük veri kümelerindeki işlem yükünü paralel olarak işleyebilmektedir. Bu durum çoklu işlemcili sistemlerde hızlı veri işlemeye olanak tanımaktadır.

ESA'lar geliştirilerek elde edilen, nesne tespiti ve nesne takibi alanında kullanılan derin öğrenme algoritmaları B-ESA (Bölge Tabanlı Evrişimli Sinir Ağı), Hızlı B-ESA, Daha Hızlı B-ESA, Tek Atış Dedektörü (TAD), Maskeli B-ESA, YOLO olarak sıralanabilmektedir.

2.4.2. Bölge Tabanlı Evrişimli Sinir Ağı (B-ESA)

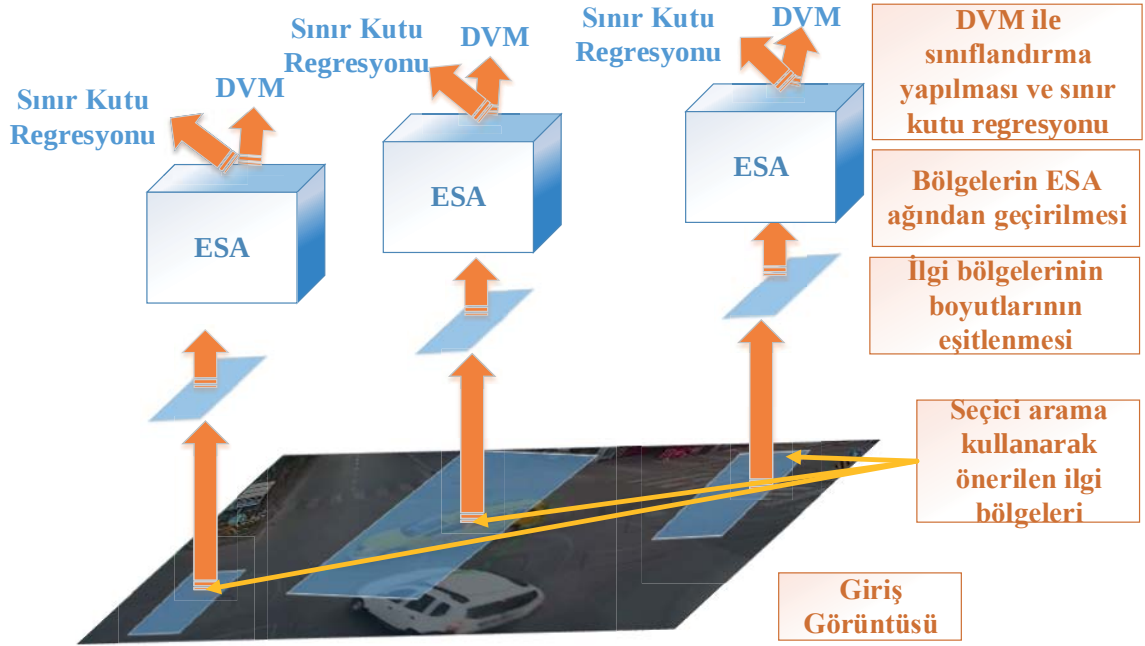
B-ESA mimarisi, nesne tanıma amacıyla kullanılan öncü yöntemlerden biridir. B-ESA'lar nesneleri tanımlayabilmek için nesnelerin etrafında sınırlayıcı kutular belirlemektedir. Her görüntü için farklı sayıda nesne olabileceğinden, çıktı katmanının uzunluğu değişkenlik gösterebilmektedir. Ayrıca her görüntüde nesnelerin boyutları yada konumları birbirinden farklı olabileceğinden, bir görüntüdeki hedef nesneleri tanımlamak ve algılamak için çok sayıda bölge gerekebilmektedir. B-ESA, bu sorunla başa çıkabilmek için “seçici arama” yöntemini kullanmaktadır [8, 92].

B-ESA, ESA'yı nesne tanıma gerçekleştirmek ve 2000 bölge önerisi aday kutusu oluşturmak üzere eğitmek için seçici aramayı kullanmaktadır. Daha sonra seçici arama tarafından oluşturulan nesne önerileri kullanılmaktadır. Her aday kutu daha sonra sabit bir boyuta dönüştürülür ve ESA'ya girdi olarak verilmektedir, bu durum bir özellik çıkarıcı görevi görmekte ve çıktı olarak 4096 boyutlu özellik üretmektedir. Bu özellikler seti, sınıflandırmayı gerçekleştirmek için Destek Vektör Makineleri (DVM) sınıflandırıcısına beslenir. B-ESA, sınıflandırma gerçekleştirmeye ek olarak, her sınırlayıcı kutunun kesinliğini artırma işlemini de gerçekleştirmektedir. B-ESA sistemine genel bakış, Şekil 2.11'de resimsel olarak gösterilmiştir.

B-ESA'nın sınırlılıkları:

B-ESA'nın kullanımını süresince tespit edilmiş bazı sınırlılıklar mevcuttur. Bu sınırlılıklar şöyle sıralanmaktadır:

1. B-ESA'yı gerçek hayat uygulamaları için uygun olmayan hale getiren büyük zaman karmaşıklığı
2. Seçici arama kullanımından dolayı ve her görüntü bölgesi için ESA aracılığıyla elde edilen özelliklerin fazla olması nedeniyle eğitimin yavaş olması
3. Seçici arama algoritmasında içsel öğrenme yeteneğinin olmaması nedeniyle hatalı aday bölge önerileri oluşturma [93].



Şekil 2.11. B-ESA mimarisi [94]

2.4.3. Hızlı Bölge Tabanlı Evrişimli Sinir Ağı (Hızlı B-ESA)

Hızlı B-ESA, B-ESA yönteminin geliştirilmesiyle oluşturulmuştur. B-ESA'dan farklı olarak, Hızlı B-ESA, tüm nesne tanıma ve sınıflandırma işlemlerini tek bir ağda gerçekleştirmektedir. Bu durum, nesne tanıma sürecini hızlandıran bir faktördür. Girshick ve ark. tarafından B-ESA'da meydana gelen büyük zaman karmaşıklığından kaynaklanan sınırlamaların üstesinden gelmek için önerilmiştir [95].

Hızlı B-ESA, beş ana bileşenden oluşmaktadır:

1. Girdi görüntüsü
2. Özellik haritaları oluşturmak için bir özellik çıkarıcı
3. Bölgesel özellik çıkarıcı
4. Tamamen bağlı katmanlar
5. Çıktı katmanı

B-ESA'dan farklı olarak, Hızlı B-ESA, seçici arama yöntemiyle bir görüntüden elde edilen 2000 bölge önerisini tek tek 2000 farklı ESA'ya girdi olarak vermek yerine tüm görüntüyü ESA'ya girdi olarak kullanmaktadır. Bir görüntünün tamamından elde edilen özellik haritasından, bölge teklifleri belirlenir ve bir ilgi bölgesi havuzlama (İBH) katmanı tarafından sabit boyuta indirgenir. Bu havuzlama katmanı ilgi bölgesi havuzlama (İBH) katmanı olarak adlandırılmaktadır. Daha sonra, her nesne önerisi için bölgesel özellik çıkarıcısını kullanarak özellik haritalarından bölgesel özellik vektörlerini çıkarmaktadır. Bu bölgesel özellik vektörleri, tamamen bağlı katmanlar

aracılığıyla sınıflandırma yapmak ve nesne sınırlayıcı kutu koordinatlarının tahmin edilmesi için kullanılmaktadır. Son olarak, çıktı katmanı, her bölgesel öneri için sınıflandırma sonuçlarını ve sınırlayıcı kutu koordinatlarını üretmektedir. Ayrıca Hızlı B-ESA'da bir diğer önemli gelişme ise B-ESA'da kullanılan DVM ve regresörün birleştirilmesidir ve bu sayede performansı arttırılmıştır.

Hızlı B-ESA'nın Avantajları

İlk olarak, Hızlı B-ESA, B-ESA yöntemiyle kıyaslandığında bazı önemli avantajlara sahiptir. tüm nesne tanıma işlemleri tek bir aşda gerçekleştirilmektedir. Bu, nesne tanıma sürecinin hızlandırılmasına yardımcı olmaktadır. İkincisi, Hızlı B-ESA ile tüm nesne önerileri tek bir özellik haritası üzerinde işlenmektedir. Bu, ağı bellek kullanımını azaltmaktadır. Üçüncü olarak, Hızlı B-ESA, daha iyi sınıflandırma sonuçları elde etmektedir.

Hızlı B-ESA'nın Sınırlılıkları:

1. İlgi bölgesi havuzlama katmanının kullanılmasıyla, Hızlı B-ESA'nın zaman karmaşıklığı B-ESA'ya kıyasla bir derece azaltılmıştır. Ancak seçici arama algoritmasının öğrenememe durumundan kaynaklanan hatalı bölge önerileri oluşturma sorunu da Hızlı B-ESA'da mevcuttur. ESA, B-ESA gibi, Hızlı B-ESA'da bölge önerileri seçici arama algoritması kullanılarak tespit edilmektedir [8].

2.4.4. Daha Hızlı Bölge Tabanlı Evrişimsel Sinir Ağı (Daha Hızlı B-ESA)

Daha Hızlı B-ESA, Ross B.Girshick tarafından 2016 yılında geliştirilen bir nesne tanıma algoritmasıdır. Bu algoritma, nesne tanıma işlemi için önceki algoritmalara göre daha hızlı ve daha doğru sonuçlar sağlamaktadır.

Daha Hızlı B-ESA bir aşda özellik çıkarma, teklif çıkarma ve düzeltmeyi kendi içerisinde entegre ederek kullanmaktadır [96]. Genel performans, özellikle algılama hızı açısından büyük ölçüde iyileştirilmiştir. Daha Hızlı B-ESA, bölge önerileri oluşturmak için kullandığı evrişim ağını, önerilen bölge sayısını 2000'den yaklaşık 300'e düşüren nesne tanıma ağı ile paylaşmaktadır [97]. Önerilen bölgenin kalitesi de daha Hızlı B-ESA ile iyileştirilmiştir [93].

Daha Hızlı B-ESA iki aşamalı bir nesne tanıma modelidir ve mimari yapısı şöyledir:

- **Bölge Teklif Ağı (BTA):** Daha Hızlı B-ESA'nın ilk aşaması olarak görüntülerdeki olası nesne konumlarını bulmak için önceden tanımlanmış bölge önerileri üreten bir bölge teklif ağı (BTA) kullanılmaktadır. BTA temelde tam evrişimli bir sinir ağıdır. Girdi olarak bu evrişimli katman herhangi bir boyutta girdiyi almakta ve nesnelere ait olabilecek dikdörtgen şeklindeki bölge teklifini önermektedir. Bu öneriyi, evrişimli katman tarafından oluşturulan öznetelik haritası

- **Hızlı B-ESA:** Daha Hızlı B-ESA'nın ilk aşaması olarak kullanılan Hızlı B-ESA, BTA tarafından üretilen bölge önerilerini almakta ve her bir bölgeyi sınıflandırmak ve konumlandırmak için kullanmaktadır. Daha sonra, her bir bölge önerisinin bir nesne olduğunu ve o nesnenin hangi sınıfa ait olduğunu belirlemek için bir sınıflandırma ve konumlandırma ağı kullanmaktadır [96].

Daha Hızlı B-ESA'nın birçok avantajı bulunmaktadır. İlk olarak, hızlı bir nesne tanıma için tasarlanmıştır. İkinci olarak, tek bir model, nesne konumlandırma ve sınıflandırma için gereken iki ayrı modelin yerini almaktadır. Son olarak, bölge önerisi ağı, diğer nesne tanıma yöntemlerine göre daha az işlemci ve bellek kullanmaktadır [99]. Daha Hızlı B-ESA performans dar boğazına neden olan seçici arama yerine daha kullanışlı olan BTA kullanmaktadır. Şekil 2.12'de Daha Hızlı B-ESA'nın genel mimarisi gösterilmektedir.

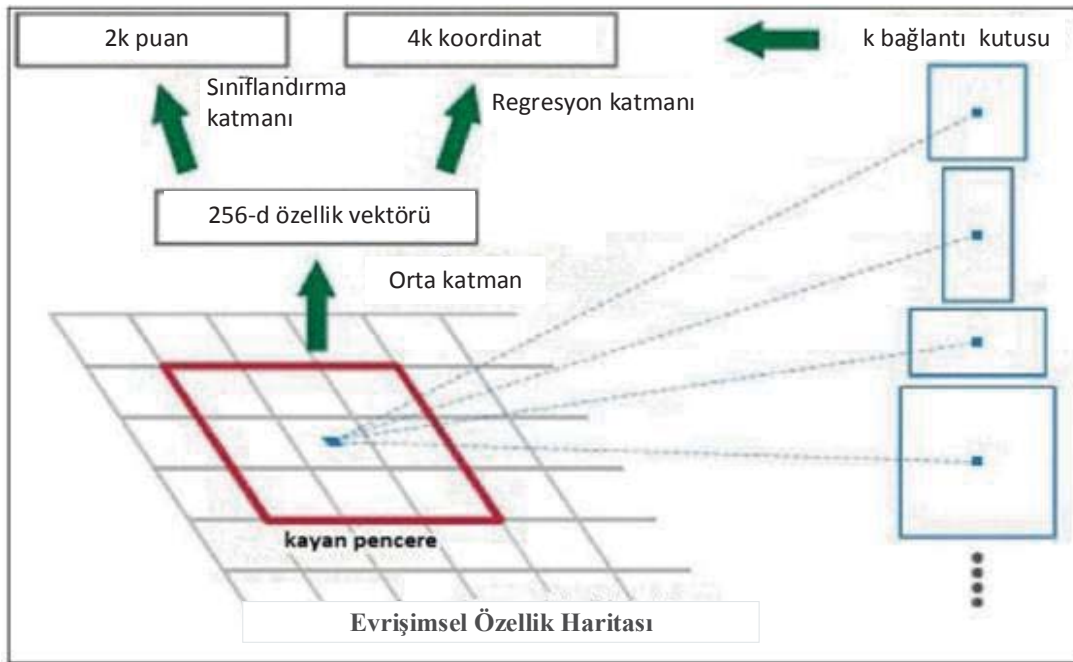


Bağlantı Kutuları

Bağlantı kutuları derin öğrenme nesnesi algılayıcılarının önemli parametreleridir. Daha Hızlı B-ESA'da kullanılan bağlantı kutuları belirli bir yükseliğe ve genişliğe sahip olan sınırlayıcı kutulardır. Bu kutular eğitim verilerindeki nesne boyutlarına göre seçilmektedir ve tespit edilmek istenen nesnenin en boy oranını yakalamak için tanımlanmaktadır. Bağlantı kutuları birden çok ve farklı boyutlardaki nesnelere dair tahmin ve algılama bilgileri içermektedir.

Bağlantı kutularının şekli, ölçeği ve sayısı, dedektörlerin verimliliğini ve doğruluğunu etkilemektedir. Genel olarak 3 ölçek ve 3 en-boy oranı bulunmaktadır. Bu nedenle toplam $K=9$ bağlantı kutusu vardır.

Her bağlantı kutusu görüntü boyunca yerleşmektedir. Ağ çıkışlarının sayısı, bu bağlantı kutularının sayısına eşittir. Ağ, tüm çıktılar için tahminler üretmektedir [100]. Şekil 2.13'te bölge öneri ağının ve bağlantı kutularının yapısı gösterilmektedir.



Şekil 2.13. Bölge öneri ağı ve bağlantı kutuları [96]

Birleşme üzerinden kesişim (BÜK) mesafe metriğini kullanan eğitim verilerinden bağlantı kutuları tahmin edilmektedir. Kutu boyutları arttıkça daha büyük hatalar üreten Öklid mesafe metriğinin aksine, Birleşme üzerinden kesişime dayalı bir mesafe metriği, kutuların boyutuyla değişmemektedir [101]. Ek olarak, bir BÜK mesafe metriği kullanmak, benzer en boy oranlarına ve boyutlarına sahip kutuların birlikte kümelenmesine yol açmaktadır. Bu durumda, verilere uyan bağlantı kutusu tahminleriyle sonuçlanmaktadır. Eğitim verileri ile bağlantı kutuları üzerinde bağlantı sayısının seçilmesi ve ortalama BÜK değerleri elde edilmektedir.

2.4.5. YOLO (Yalnızca Bir Kez Bakarsınız)

YOLO, nesne tespiti alanında son yılların en popüler ve etkili yöntemlerinden biridir. Bu yöntem, nesne tespiti işlemini bir kez yaparak tüm nesneleri aynı anda tespit etme avantajı sunan bir derin öğrenme yöntemidir. YOLO ile tüm görüntü aynı anda işlenerek nesne tespiti işlemi gerçekleştirilmektedir. Diğer nesne tespiti yöntemlerinden farklı olarak, YOLO, tespit edilen nesnelerin sınıfını ve konumunu aynı anda tahmin etmektedir. Bu sayede, tespit süresi ve doğruluğu açısından oldukça etkilidir. YOLO yalnızca bir kez bakmanız yeterli düşüncesinden yola çıktığı için gerçek zamanlı uygulamalarda oldukça başarılı sonuçlar elde edilmesini sağlamıştır [102].

YOLO, ESA mimarisi kullanarak çalışmaktadır. Giriş görüntüsü, bir dizi özellik haritasına dönüştürülmektedir ve her bir özellik haritası, farklı boyutlardaki nesneleri tespit etmek için kullanılmaktadır. Özellik haritaları üzerinde yapılan tespit işlemi sonucunda, her bir nesne için bir sınırlayıcı kutu ve bu kutu içindeki nesnenin sınıfı tahmin edilmektedir. YOLO, bu işlemi tek bir ağ üzerinden gerçekleştirdiği için, diğer yöntemlere göre oldukça hızlıdır [102].

YOLO'nun Kullanım Alanları

YOLO, nesne tespiti alanında birçok farklı alanda kullanılmaktadır. Özellikle gerçek zamanlı uygulamalar başta olmak üzere güvenlik sistemleri, otonom araçlar, robotik, tıp gibi alanlarda yaygın olarak kullanılmaktadır. YOLO ayrıca, insanların nesnelerle etkileşim kurdukları uygulamalarda da kullanılabilir.

YOLO'nun Performansı

YOLO, diğer nesne tespiti yöntemlerine göre oldukça hızlı ve doğru sonuçlar veren bir yöntemdir. YOLO, diğer yöntemlere göre daha az hata oranıyla daha az işlemci gücü kullanarak çalışabilmektedir.

YOLO [103], tanıma ve sınıflandırmayı içeren tek adımlı bir süreç oluşturmak için geliştirilmiştir. Sınırlayıcı kutu ve sınıf tahminleri, giriş görüntüsünün bir değerlendirmesinden sonra yapılır. YOLO'nun en hızlı mimarisinin saniyedeki çerçeve sayısı 45 fps'ye ulaşabilmektedir. Daha küçük bir sürüm olan Tiny-YOLO, GİB'li bir bilgisayarda saniyedeki çerçeve sayısı 244 fps'ye ulaşmaktadır.

YOLO fikri, sınırlayıcı kutu tahminlerinin ve sınıf tahminlerinin aynı anda yapılması bakımından diğer geleneksel sistemlerden farklıdır. Giriş görüntüsü önce bir $S \times S$ ızgarasına bölünmektedir. Ardından, her ızgara hücresinde, her biri bir güven puanına sahip olan sınırlayıcı kutular tanımlanmaktadır [103]. Buradaki güven, bir nesnenin her sınırlayıcı kutuda var olma olasılığını ifade etmektedir.

YOLO'nun Faydaları:

- Saniyedeki çerçeve sayısı gerçek zamandan daha iyidir. Saniyedeki çerçeve sayısı 45 fps (daha büyük ağ) ile 150 fps (daha küçük ağ) arasında ki hızlarda işlemektedir.
- Ağ, görüntüyü daha iyi genelleştirebilmektedir.

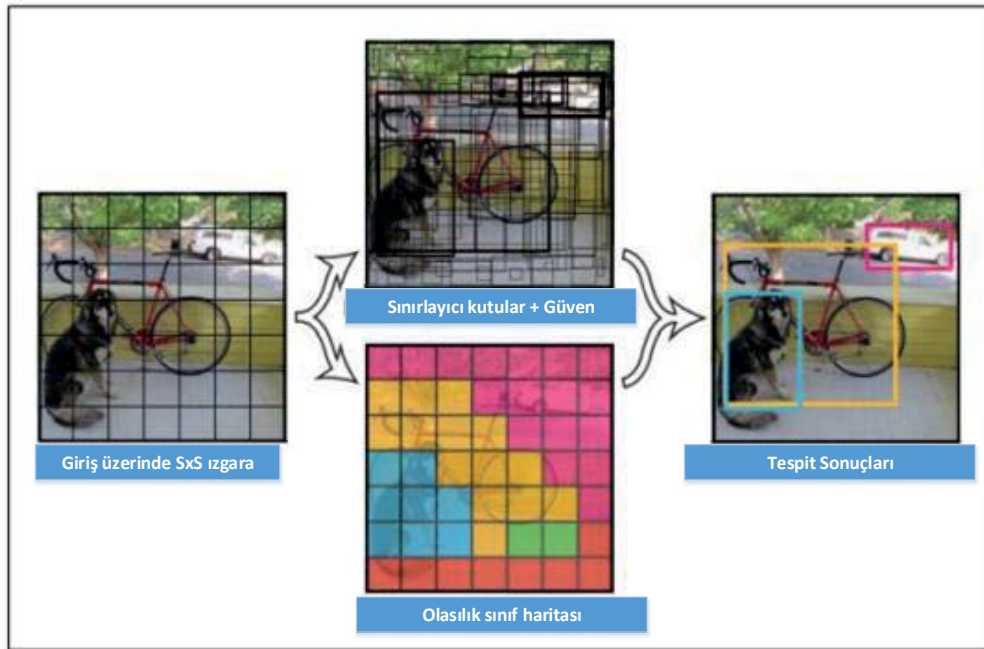
YOLO'nun Dezavantajları:

- Daha Hızlı B-ESA'ya kıyasla nispeten düşük hatırlama ve daha fazla yerleştirme hatasına sahiptir
- Her ızgara yalnızca 2 sınırlayıcı kutu önerebileceğinden, yakın nesneleri tanımakta zorlanmaktadır.
- Küçük nesneleri algılamak için mücadele etmektedir.

YOLO'nun çalışma prensibi

YOLO sinir ağı, aday kutuları çıkarma, özellik çıkarma ve nesne sınıflandırma yöntemlerini bir sinir ağına entegre etmektedir. YOLO sinir ağı, aday kutuları doğrudan görüntülerden çıkarır ve nesneler, tüm görüntü özellikleri aracılığıyla algılanır [104].

YOLO ağında, görüntüler $S \times S$ ızgaralarına bölünür. Aday kutuları, X eksen ve Y eksen üzerinde eşit olarak dağıtılmaktadır. Aday kutuları, nesne algılama özelliğine sahiptir ve her bir aday kutusunda nesnenin varlığının güven oranı tahmin edilmektedir. Güven, görüntülerin nesneyi içerip içermediğini ve nesnenin konumunun doğruluğunu yansıtmaktadır [105]. Şekil 2.14'te YOLO regresyon yaklaşımı, Şekil 2.15'te ise YOLO katman yapısı gösterilmektedir.



Şekil 2.14. YOLO regresyon yaklaşımı [101]

2. YOLO dedektörünün ızgara başına algıladığı maksimum nesne sayısı 1 olduğundan, bir ızgara içinde birden fazla nesne olduğunda hatalı algılama gerçekleştirmektedir.

2.4.7. YOLO sürüm 2

Redmon ve ark. [101], YOLO 9000 olarak da bilinen ve Hızlı B-ESA, Daha Hızlı B-ESA gibi nesne tanıma ağlarından daha verimli aynı zamanda uygun bir süre içinde algılama gerçekleştiren, geliştirilmiş bir YOLO sürümü önermiştir. YOLO dedektörünün bu sürümünde, bir önceki sürümdeki sınırlamaları aşmak için YOLO sürüm 1'in mimarisinde çeşitli değişiklikler gerçekleştirilmiştir.

YOLO sürüm 1'de yapılan bazı önemli mimari değişiklikler şunlardır:

1. Yığın normalleştirme katmanının tanıtılması em önemli değişikliklerden bir tanesidir. Bu katman tüm konvolüsyon katmanından sonra eklenerek, dedektörün performansı artırılmıştır.
2. Eğitim için 224×224 boyutunda görüntüleri kullanan ve test aşamasında boyutlarını 448×448 'e çıkaran YOLO dedektörünün aksine test aşamasında görüntü çözünürlüğündeki ani artış, YOLO dedektörü sürüm 1'in performans verimliliğini düşürmektedir. Bu nedenle, bu dezavantajı ortadan kaldırmak için YOLO sürüm 2'de daha ince bir ayar ile ağ, 10 epok boyunca 448×448 boyutunda ki görüntüler üzerinde eğitilmektedir. Yüksek çözünürlüklü görüntülerde kademeli olarak görüntü boyutunun ayarlanabilmesi, YOLO'da görüntü boyutunda ani artışa bağlı olarak meydana gelen ortalama hassasiyet düşüşünden kaynaklanan sorunu çözmüştür.
3. Bu geliştirilmiş model, YOLO sürümü gibi evrişimli katmanların üstünde yer alan tam bağlı katmanları kullanmayarak ve bunun yerine bağlantı kutularını kullanarak işlem yapmaktadır. Bağlantı kutularının kullanılması, YOLO sürüm 2'nin ortalama hassasiyetini YOLO sürüm 1'e göre azaltsa da, hatırlama değerini artırmaktadır.
4. YOLO sürüm 1, elle sınırlayıcı kutular kullanarak ağın eğitimini gerçekleştirir, ancak öğrenme sürecini daha kolay hale getirmek için sürüm 2'de önerilen mesafe metriğiyle birlikte K-ortalama algoritması kullanılarak oluşturulan sınırlayıcı kutuları kullanarak ağlarının eğitimini gerçekleştirilmiştir.

Modelin belirtilmesi gereken diğer önemli özellikleri şunlardır:

5. **Doğrudan konum tahmini:** Bu özellik, bağlantı kutularının tanıtılması modelin istikrarsızlığını bir dereceye kadar artırdığından, esas olarak yöntemin kararlılığı ile ilgilidir. Bu nedenle modelin kararlılığını artırmak için sınırlayıcı kutuların koordinatları [0 1] içinde sınırlandırılmıştır.

6. **Ayrıntılı özellikler:** Daha Hızlı B-ESA gibi son teknoloji yöntemlerinin çoğu, ağı farklı çözünürlüklere uyarlamak için farklı çözünürlüklere sahip özellikler üzerinde çalışmaktadır. Ancak YOLO sürüm 2'de, ağı farklı çözünürlüklere sahip özellikler üzerinde çalıştırmak yerine ve ağı hem düşük çözünürlüklü hem de yüksek çözünürlüklü özellikleri yerleştirmek yerine bitişik olarak istifleyerek birleştiren bir geçiş katmanı eklenmiştir.
7. **Çok Ölçekli Eğitim:** 448×448 çözünürlüklü görüntüleri kullanarak ağı eğiten YOLO sürüm 1'den farklı olarak, YOLO sürüm 2, ağı farklı çözünürlükteki görüntüleri kullanarak eğitmektedir. Bu ağı, belli bir dönem boyunca belirli bir çözünürlükteki görüntüler üzerinde çalıştırıldıktan sonra görüntülerin çözünürlüğünü rastgele değiştirmektedir. Bu ağın çözünürlük aralığı 302 ile 608 arasında değişmektedir. Ağın bu özelliği, farklı çözünürlüklere uyum sağlamasına ve görüntü çözünürlüklerinden bağımsız olarak verimli çalışmasına yardımcı olmaktadır [8].

2.5. Derin Öğrenme Modelleri

Bu çalışmada görüntü işlemede oldukça yaygın olarak kullanılan AlexNet, VGGNet, GoogLeNet, ResNet, MobileNet derin öğrenme modelleri kullanılmıştır. Kullanılan ağ modelleri alt başlıklar halinde açıklanmıştır.

2.5.1. ImageNet

Görüntü işleme üzerine çalışma yapacak bireylere yardımcı olması amacıyla oluşturulmuş ve yaklaşık 14 Milyon görselden oluşan herkese açık bir hazır veritabanıdır. ImageNet, görüntü sınıflandırma görevlerinde makine öğrenimi modellerini eğitmek için kullanılan büyük bir görüntü veri kümesidir. İlk olarak 2010 yılında Stanford Üniversitesi'ndeki araştırmacılar tarafından yayınlanan bir makalede tanıtılmıştır [106].

Veri seti, diğerleri arasında "köpek", "kedi", "araba" ve "uçak" gibi 1000 farklı sınıfa ait 14 milyondan fazla görüntüden oluşur. Bu görüntüler internetten toplanmıştır ve insan yorumcular tarafından etiketlenmiştir [103].

ImageNet, derin öğrenme modellerinin, özellikle evrimsel sinir ağlarının (ESA'lar) geliştirilmesinde önemli bir rol oynamıştır. ImageNet veri setini kullanarak en doğru görüntü sınıflandırma modellerini geliştirmek için dünyanın dört bir yanından ekiplerin yarıştığı yıllık bir yarışma olan ImageNet Büyük Ölçekli Görsel Tanıma Mücadelesinin (ILSVRC) temeli olarak kullanılmıştır [91].

ImageNet üzerinde eğitilen modellerin başarısı, nesne tanıma, görüntü bölümlendirme ve yüz tanıma dahil olmak üzere çeşitli uygulamalarda önemli ilerlemelere yol açmıştır. Örneğin, 2014

yılında ILSVRC yarışmasını kazanan VGGNet modeli, sonraki birçok görüntü sınıflandırma görevi için temel model olarak yaygın şekilde kullanılmıştır [91].

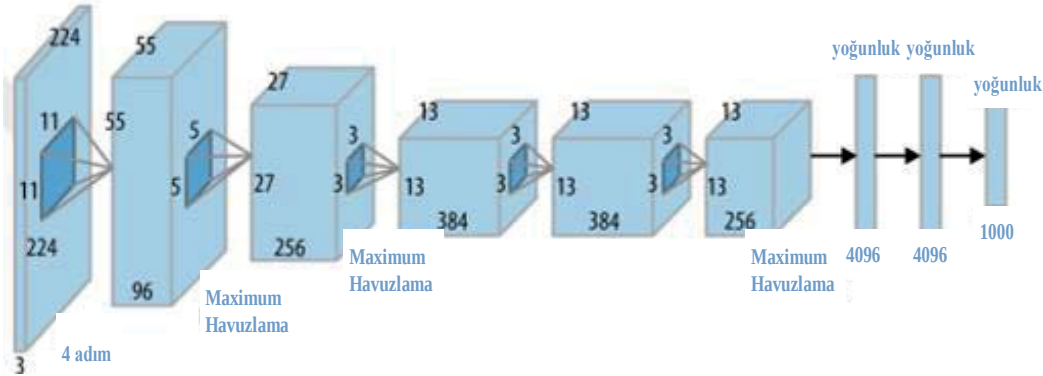
Genel olarak, ImageNet'in bilgisayar görüşü alanında önemli bir etkisi olduğu ve görüntü sınıflandırma görevlerinde birçok makine öğrenimi modelinin başarısına katkıda bulunduğu bilinmektedir.

2.5.2. AlexNet

2012 yılında ImageNet yarışmasında AlexNet önceki tüm rakiplerinden daha iyi başarımlar göstermiştir. Alex Krizhevsky, Ilya Sutskever ve Geoffrey Hinton tarafından geliştirilen bu ağ sınıflandırma doğruluğunu önemli derece arttırmayı başarmış ve hata oranını %15.3 lere kadar azaltmıştır. Mimari olarak evrişim, havuzlama ve tam bağlı katmanlardan oluşmaktadır.

AlexNet mimarisi $227 \times 227 \times 3$ boyutundaki verileri girdi olarak kullanmaktadır ve evrisim katmanında, 3×3 veya 5×5 boyutlu filtreler gezdirilerek, bir sonraki katmana girdi olacak yeni görüntü elde edilmektedir.

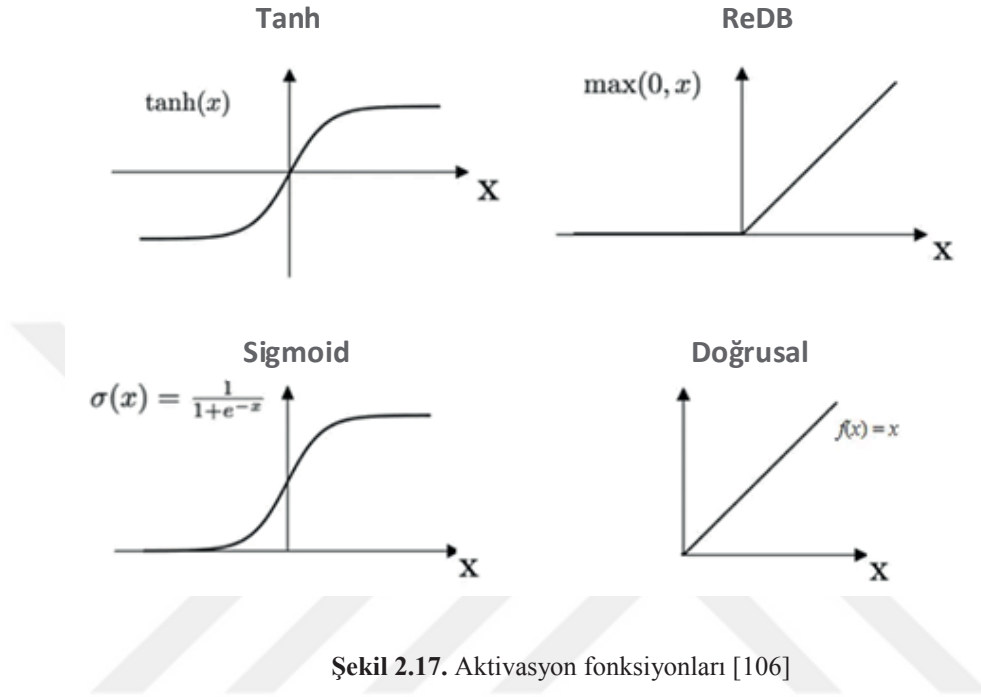
AlexNet, ESA mimarisi olarak 5 evrişim katmanı, 3 tam bağlantılı katman olmak üzere 8 katmanlı yapı kullanmaktadır. Bu ağ, ilk kez, öğrenme yoluyla elde edilen özelliklerin, manuel olarak tasarlanmış özelliklerinden daha verimli olabileceğini ve görüntü işleme uygulamalarında önceki geleneksel kalıpların değişeceğini göstermiştir. AlexNet, ESA modelinin genel yapısı Şekil 2.16'da gösterilmiştir.



Şekil 2.16. AlexNet mimarisinin genel yapısı [107]

AlexNet'in ilk katmanında evrişim penceresinin şekli 11×11 'dir. Nesneyi anlamak için daha büyük bir evrişim penceresi gerekir. İkinci kattaki evrişim penceresi şekli 5×5 'e, ardından 3×3 'e düşürülür. Ek olarak, birinci, ikinci ve beşinci eğrisel katmanlardan sonra ağ, 3×3 pencere şeklinde ve 2'lik uzun bir aralıkla maksimum biriktirme katmanları ekler. Son evrişimli katmandan sonra, 4096 çıkışlı tamamen bağlı iki katman vardır. Bu iki büyük tamamen bağlı katman, model parametrelerinde yaklaşık 1 GB bellek alanı üretir. GİB bellekleri düşünüldüğünde çok önemlidir.

Yaklaşık 60 milyon parametereye sahiptir. AlexNet in hata oranı % 15.3 olarak belirlenmiştir [108, 109]. Alexnet de daha önceki ağlarda kullanılan tanh ve sigmoid fonksiyonları yerine REDB fonksiyonu kullanılmaktadır. Şekil 2.17’de aktivasyon fonksiyonları gösterilmiştir.

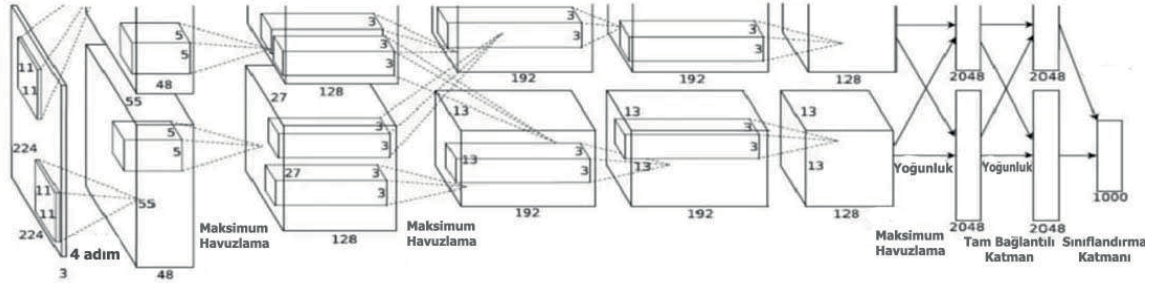


Şekil 2.17. Aktivasyon fonksiyonları [106]

REDB kullanımın, sigmoid fonksiyonuna kıyasla en büyük avantajı, eğitim hızının artması olmuştur. Ayrıca sigmoid kullanımı kaybolan gradyan adı verilen bir problemin ortaya çıkmasına sebep olmaktaydı. Bu problem sigmoid fonksiyonu izlendiğinde ortaya çıkan fonksiyon değeri arttıkça yada azaldıkça türevinin sıfıra yaklaşmasıyla ortaya çıkmaktadır. Bu sorun nedeniyle eğitim esnasında ağırlıkları güncelleştirmek zorlaşmaktadır. Ağırlığın derinliği arttıkça özellikle ilk katmanlarda öğrenme zorlaşmaktadır [110, 111].

Alexnet ayrıca bir diğer problemin daha önüne geçmiştir. Bu ağırlığın kullanımı sayesinde daha önceki mimarilerde yaşanan aşırı öğrenme probleminin önüne geçilebilmiştir. Veri artırma ve bağlantı düşürme kullanılarak eğitim verisinin ezberlenmesi anlamına gelen aşırı öğrenme problemi çözülmektedir ancak kullanılan eğitim iterasyonlarının sayısı artmaktadır. Eğer ağırlığın eğitim doğruluğu %99 olup test doğruluğu %85 olursa ağırlık aşırı öğrenmiştir diyebilmekteyiz. Bir başka deyişle ağırlığın eğitim verisini ezberlediği söylenilebilmektedir [111]. Ağırlığın genel yapısında havuzlama katmanlarında maksimum havuzlama kullanılmaktadır. Softmax ve çıkış katmanı hariç diğer katmanların sonunda REDB aktivasyon fonksiyonu kullanılmaktadır.

AlexNet, geleneksel ağların sınıflandırma etkisini büyük ölçüde geliştiren veri arttırmayı da kullanmaktadır. ResNet gibi daha karmaşık ağ yapılarına kıyasla AlexNet, ağında daha basit ve daha kısa katmanlara sahiptir. Şekil 2.18’de AlexNet katmanlarının yapısı gösterilmiştir.



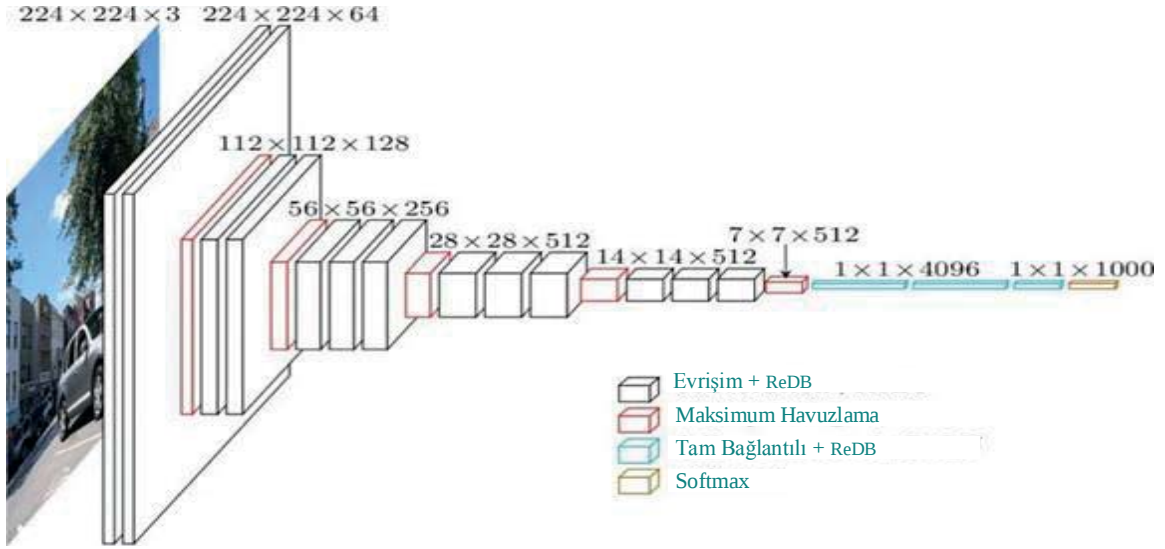
Şekil 2.18. AlexNet katmanları [89]

2.5.3. VGGNet

2014 yılında ImageNet yarışmasında rakiplerini geride bırakan iki mimariden biridir. VGGNet, 2014 yılında Oxford Üniversitesinde görevli akademisyenler tarafından hazırlanan büyük ölçekli görüntü tanıma için evrimsel ağlar ile ilgili bir makalede tanıtılan bir derin öğrenme modelidir [91]. İsmi aynı üniversitede bulunan Görsel Geometri Grubundan almaktadır. ESA’ların derinliğini arttırmak için geliştirilmiş bu model, evrimsel katmanlarında kullanılan parametre sayısını azaltarak eğitim süresini azaltmak amacıyla ortaya çıkmıştır.

VGG ağı küçük boyutlu evrimsel filtrelerden oluşmaktadır. 13 evrimsel katmanı ve 3 tam bağlantılı katmanın birleşimiyle elde edilmektedir. VGG ağı temel mantık olarak, ağı karmaşıklığını artırmamak amaçlı uygun katman derinliği ayarı dikkate alınarak tasarlanmış derin bir evrimsel sinir ağıdır. VGGNet, 3x3 boyutundaki filtreler kullanmaktadır ve bu filtreler aracılığıyla görüntülerin özelliklerini öğrenmektedir. Model 224x224x3’lük giriş verisi kullanmaktadır [95, 112].

VGGNet’in önemli bir diğer özelliği de, tüm katmanlarının aynı boyutta olmasıdır. Bu, modelin daha kolay öğrenmesini ve daha az parametre içermesini sağlamaktadır. Böylece, VGGNet daha az eğitim verisiyle daha iyi performans gösterebilmekte ve aşırı öğrenme riski azaltılmış olmaktadır. VGGNet, günümüzde görüntü tanıma ve sınıflandırma gibi çeşitli uygulamalarda sıklıkla kullanılmaktadır. Örneğin, VGGNet, görüntülerdeki nesneleri tespit etmek ve bunları sınıflandırmak için kullanılabilir. Ayrıca, VGGNet, görüntüleri tamir etme ve tamir edilmiş görüntüler üretme gibi görüntü işleme uygulamalarında da kullanılmaktadır [112]. Şekil 2.19’da VGG ağına mimarisi gösterilmektedir.



Şekil 2.19. VGG16 mimarisi[113]

Vgg16 ve Vgg19 olmak üzere iki mimari öne çıkmıştır. Vgg16, 16 evrişim katmanından Vgg19, 19 evrişimsel katmandan oluştuğu için böyle isimlendirmişlerdir.

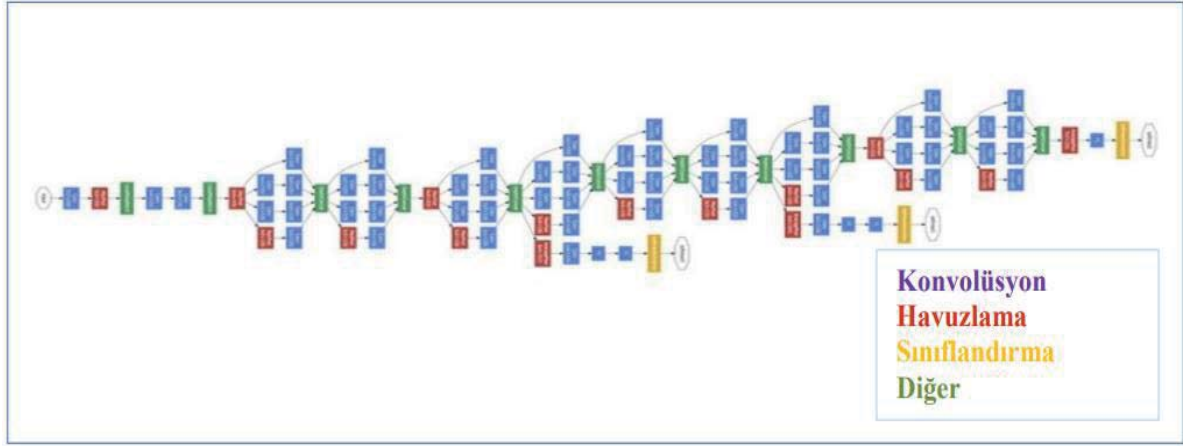
VGG16 modeli ImageNet yarışmasında yaklaşık %92.7 lik bir başarı elde etmiştir. ILSVRC-2012 ve ILSVRC-2013 y %7.0 lık bir hata oranı elde etmiştir. 13 evrişimli katman, ve 3 tam bağlı katmandan oluşmaktadır. Yaklaşık 138 milyon parametreye sahiptir. Mimarisi kendisinden önce geliştirilen modellere göre daha basittir. VGGNet, AlexNet de karşılaşılan yüksek çekirdek boyutlarını azaltarak sabitlemiştir. AlexNet mimarisinde kullanılmakta olan yüksek çekirdek boyutlarını (11x11, 5x5, 3x3) azaltılmıştır ve 3x3 'e sabitlenmiştir. Ancak, bu modelde kullanılan filtre sayısı her katmanda ikiye katlanarak son katmanda 512 ye ulaşmaktadır. Bu yapısı nedeniyle büyük bir ağ olarak kabul edilmektedir. Yaklaşık 533 MB'lık boyutu nedeniyle uygulanması kolay olmasına rağmen eğitilmesi zaman almaktadır.

2.5.4. GoogleNet

GoogleNet, Google tarafından geliştirilen ve imageNet yarışmasının galibi olan bir derin öğrenme modelidir. GoogleNet, 2014 yılında yayınlanan bir makalede tanıtılmıştır.

GoogleNet'in yapısı, birkaç farklı katman türünün bir araya getirilmesinden oluşmaktadır. Özellikle, GoogleNet, çok katmanlı evrişimli sinir ağını ve bunların yanı sıra ayrıca birkaç tam bağlantılı katman da içermektedir. GoogleNet, VGGNet gibi maksimum havuzlama yöntemini de kullanmaktadır. GoogleNet'in en önemli özelliği ise başlangıç adı verilen bir katman türünü kullanmasıdır. Bu katman türü, bir girdi resmini birkaç farklı ölçekte çalıştırarak özelliklerini çıkarmakta ve bu özellikleri birleştirerek resmin ne olduğunu tahmin etmeye çalışmaktadır.

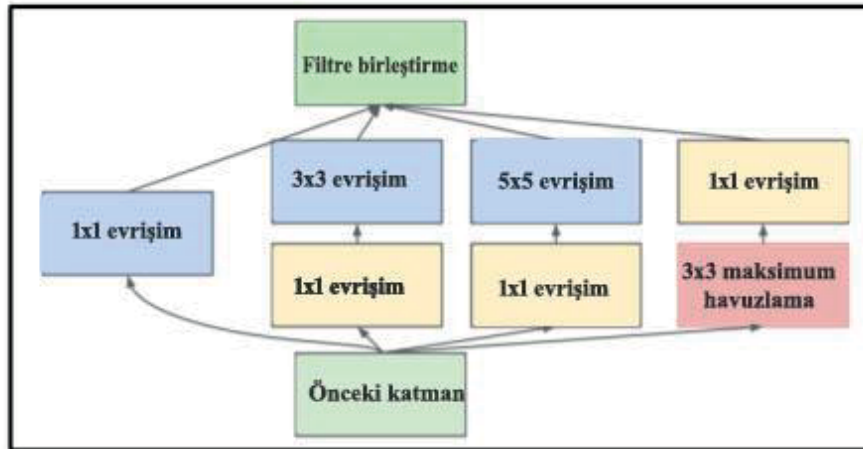
GoogleNet, günümüzde hala popüler olan ve sıkça kullanılan bir model olarak kabul edilmektedir. Özellikle, görüntü tanıma ve sınıflandırma gibi uygulamalarda sıkça kullanılmaktadır. GoogleNet kullanılarak yüz tanıma ve nesnelerin konumlarını tahmin etme vb. uygulamalar da gerçekleştirilmiştir [114]. Şekil 2.20’de GoogleNet’e ait ağ mimarisi gösterilmiştir.



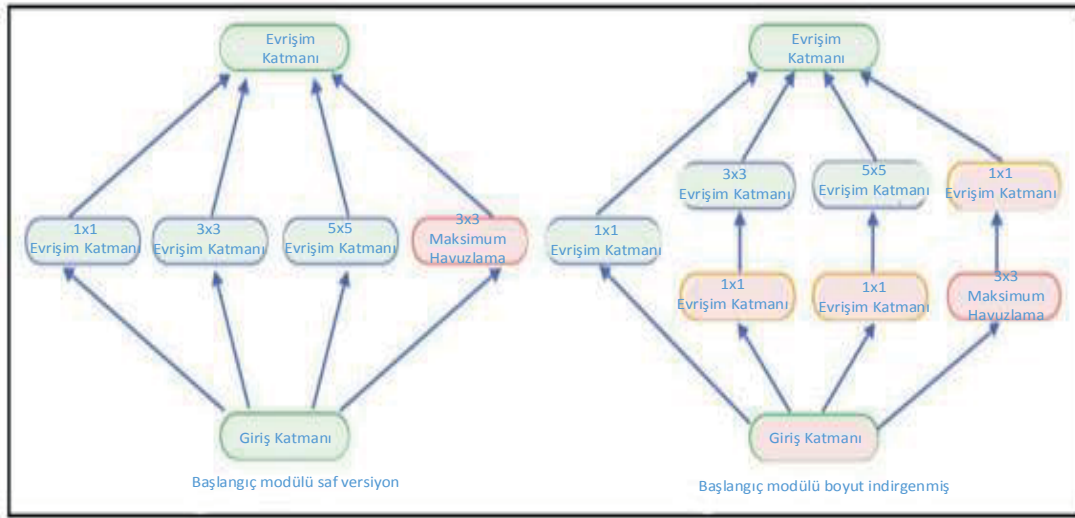
Şekil 2.20. GoogleNet mimari yapısı [114]

GoogLeNet giriş katmanı, 224 x 224 boyutunda görüntüyü girdi olarak kullanmaktadır. Bu ağ tasarımı, düşük donanım kaynaklarıyla bile çalıştırılabilecek şekilde hesaplama verimliliğini arttırmak amaçlanmıştır. GoogleNet kendisinden önceki tüm modellerden daha az yaklaşık 7 milyon ağ parametresine sahip olmasına rağmen 22 katmanlı yapısıyla daha büyük bir ağıdır. GoogleNet kullandığı parametreler açısından AlexNet ve VGG’den daha düşük bir orana sahiptir.

Ağ, 9 başlangıç modülünden oluşmaktadır. Bu nedenle kendinden önceki ağlardan hesaplama açısından daha maliyetli olmasına rağmen hesaplama karmaşıklığı azaltılmıştır. Şekil 2.21’de GoogLeNet başlangıç katmanı, Şekil 2. 22’de başlangıç mimarisi gösterilmektedir.



Şekil 2.21. GoogleNet başlangıç katmanı [114]



Şekil 2.22. GoogleNet başlangıç mimarisi [115]

Kullanılan 7x7'lik filtre yardımıyla girdi görüntüsü küçültülerek görüntünün niteliği korunmaktadır. Girdi görüntü boyutları 8 kata kadar azaltılmaktadır. Buna rağmen elde edilen özellik haritası daha fazla olmaktadır. Boyutun azaltılması katmanlardaki işlem sayısının azalmasına ve bu sayede hesaplama verimliliğinin artmasına neden olmaktadır.

Başlangıçta 7x7 boyutunda filtreler ile 64 öznitelik haritası oluşturulmaktadır. Daha sonra 3x3 filtre kullanılarak maksimum havuzlama katmanı kullanılmaktadır. Sonraki aşamada evrişimsel katman ve maksimum havuzlama katmanları tekrar yer almaktadır. Bir sonraki evrişimsel katman 3x3'lük filtre kullanmaktadır. Böylelikle 192 öznitelik haritası oluşturulmaktadır. İki başlangıç katmanı ve maksimum havuzlama katmanı ile 255 ve 480 öznitelik haritası oluşturulmaktadır. Diğer aşamada beş başlangıç katmanı ve yine maksimum havuzlama katmanı ile 512, 528 ve 832 öznitelik haritası oluşturulmaktadır. Sonraki aşamada 2 tane başlangıç katmanı, ortalama havuzlama katmanı ile 832 ve 1024'lük öznitelik haritaları oluşturulmaktadır. Böylelikle en sonda tam bağlantılı katman ve softmax katmanı yer almaktadır.

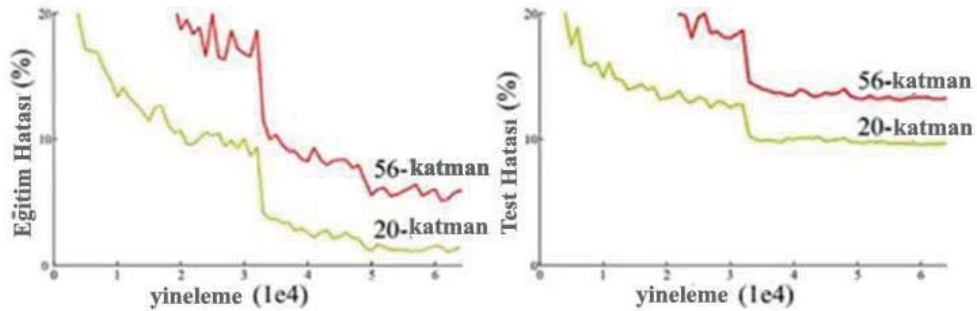
Bu mimarinin öne çıkan özellikleri arasında 1x1 evrişim ve ortalama havuzlama katmanı kullanımı yer almaktadır. 1x1 evrişim ile kullanılan parametere sayısının azaltılarak ağın derinleşmesi amaçlanmıştır. Ortalama havuzlama katmanı ile ise kullanılan tam bağlı katmanların hesaplama maliyetini arttırdığı düşünüldüğünden ağın sonuna eklenen bu yeni katman ile parametre sayısı azaltılmakta ve elde edilen doğruluk oranı artmaktadır. GoogleNet ile elde edilen hata oranı % 6.67 olarak elde edilmiştir.

2.5.5. ResNet

2015 yılında imagenet yarışmasında birinciliği elde eden bu mimari derin artık ağlar ifadesinin kısaltılmasıyla elde edilmiştir. Artık sinir ağı olarak da isimlendirilmektedir. Resnet mimarisi 224x224x3 boyutundaki girdi görüntülerini işlemektedir [116].

ResNet, derin öğrenme modellerinin derinliğinin arttıkça performansının düştüğü "derinlik etkisi" sorununu çözmeyi hedeflemektedir. Daha önce geliştirilen derin öğrenme mimarilerinde aşırı öğrenme problemi engellenebildiği müddetçe ağı derinleşmesinin ağı doğruluk oranını artırdığı düşünülmekteydi. Ancak derinlik arttıkça eğitilmek için etiketlenen veri ile tahmin edilen veri karşılaştırılarak ortaya çıkan ağırlıkları değiştirmek için kullanılan sinyalin önceki katmanlarda giderek azalmasına neden olmaktadır. Bu durum kaybolan gradyan sorununu ortaya çıkarmaktadır. Katman sayısının artması ile gradyanlar o kadar küçülmektedir ki en sonunda sıfırlanmaktadır. Resnet ile sunulan artık bağlantı ile bu küçülmenin etkisi hafifletilmektedir. Bu sorun, derin öğrenme modellerinin parametre sayısının arttıkça, modelin öğrenme yeteneğinin azaldığı ve aşırı öğrenme riskinin arttığı anlamına gelmektedir [116]

Şekil 2.23'te ki grafikte görüldüğü gibi katman sayısının artması ile hem eğitim verisinde hemde test verisinde hata oranının arttığını göstermektedir. Katman sayısının ve derinliğin artmasıyla hata azalacağına doğruluk oranı doyuma ulaşmakta ve ardından hızla düşmektedir. Bu durumda daha derin öğrenme mimarilerinde, optimizasyon sorunu oluştuğunu ve bir bozulma sorunu yaşandığını ortaya çıkarmaktadır.



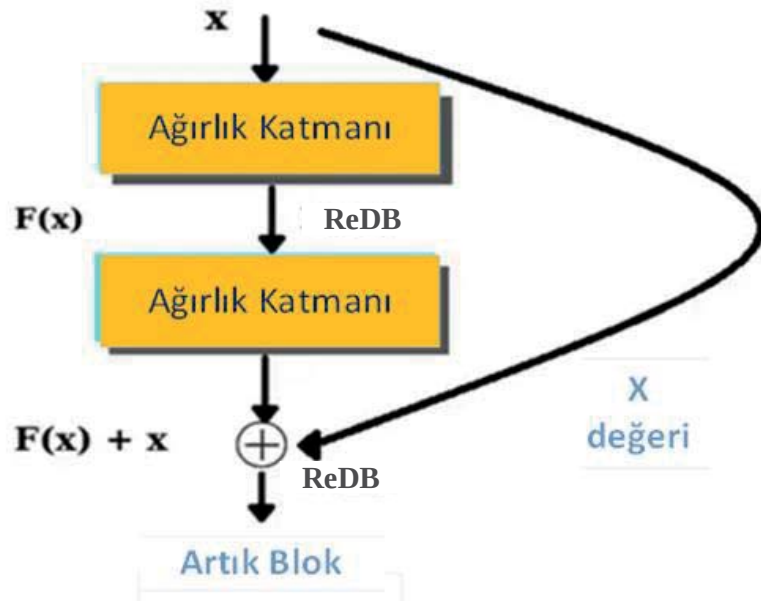
Şekil 2.23. Görüntü tanımda derin artık öğrenme [116]

ResNet, artık bağlantı adı verilen bir yapı kullanarak bu sorununu çözmeyi amaçlamaktadır. Bu yapı, modelin aşırı derecede derin olmasını önleyerek modelin öğrenme yeteneğini arttırmayı hedeflemektedir. Bu yapı, modelin öğrenme sırasında katmanlar arasındaki veri geçişini kolaylaştırarak, modelin derinlik etkisini azaltmayı amaçlamaktadır [118]. Bu amaçla artık ağlarda, bağlantı atlama adı verilen bir yöntem kullanılmaktadır. Kullanılan bu atlama bağlantılarında, aradaki bazı katmanlar atlanarak bir katmanın diğer katmana doğrudan bağlanması sağlanmaktadır.

Bu sayede yeni bir blok yapı oluşmaktadır ve mimarinin geneliylede birleşerek artık ağı oluşturmaktadırlar.

Bu artık bağlantı yada atlama bağlantılarının temel faydası herhangi bir katmanın mimari yapısını ve performansını etkilemeden atlanmasıdır. Bu sayede kaybolan gradyanın neden olduğu problem çok derin bir sinir ağına bile ortaya çıkmamaktadır [117].

Derin artık ağlar, görüntü sınıflandırma ve görüntü bölütlemeye yüksek başarımla göstermektedir. Derin artık ağlar bir filtre bankası gibi çalışmaktadır. Ağı mimarisinin karmaşıklığını azaltmak için küçük boyutlu evrişim filtreleriyle daraltılmıştır. Derin ağlarda, ağı doğruluğunu yükseltmek amacı ön plana alındığında aşırı uyum ve doygunluk sorununa sahip olmaktadır. Ağı yapısı derinleşmeye başladığında ve gizli katman sayısı arttıkça hesaplama değeri artmakta ve ağı yapısı karmaşılaşmaktadır. ResNet bu sorunun önüne geçmek için bağlantıları atlayarak çözmek amacıyla artık ağların yapısını kullanmaktadır. Bu sorunları azaltmak için kullanılan kısayol veya artık ağ yöntemi şekil 2.24'te gösterilmiştir



Şekil 2.24. ResNet artık blok [118]

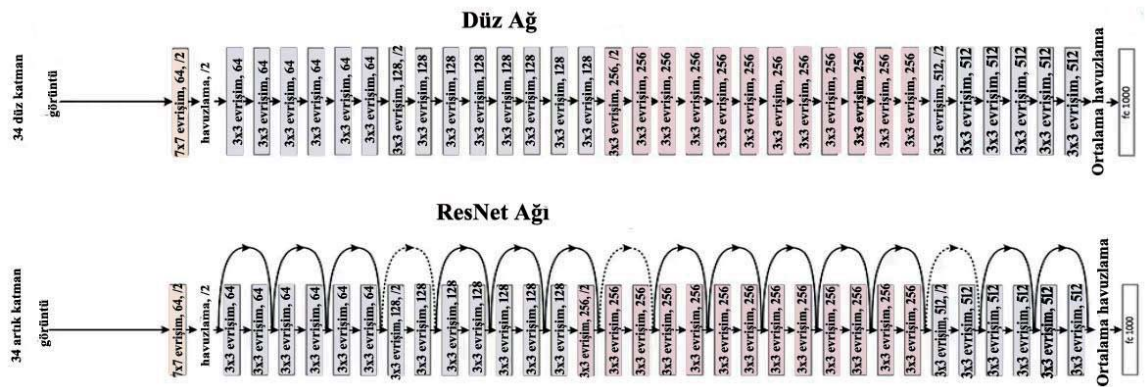
Bu mimari artık bloklardan oluşmaktadır. ResNet artık blok yapısında bulunan x çıkışı artık bağlantı yada atlama bağlantısı olarak kabul edilmektedir. Bu atlama bağlantıları bir yada daha fazla katmanı atlayabilmektedir. Artık bloklar sayesinde giriş verileri daha hızlı aktarılmaktadırlar. Artık blokta, x girişinin evrişimi sonucu bir $F(x)$ sonucu oluşmaktadır. Bu sonuç daha sonra orijinal x girişine eklenmektedir ve $H(x) = F(x) + x$ olarak ifade edilmektedir.

Ağı eğitimi sırasında çıktı olarak elde edilen değer geri yayılım yöntemi ile önceki katmanların ağırlıklarını güncelleştirmek için kullanıldığında başlangıçtaki katmanlarda sıfıra

yakınsama meydana gelir ve ağda öğrenme gerçekleşmez. Bunun önüne geçebilmek amacıyla REDB fonksiyonu kullanılan bir yapı önerilmiştir. Bu stratejinin en büyük avantajı daha önceki mimarilerde mümkün olmayan 150 katmana kadar daha derin ağlar oluşturulabilmesidir [119].

Artık ağ (artık blok), AlexNet'in 2012'de gösterdiği başarıdan sonra en önemli başarılarından birini elde etmiştir [85]. Artık ağın ana temelleri VGG ağlarının temel yapısından esinlenerek ortaya çıkmış olsada, VGG ağlarından daha az sayıda filtreye ve karmaşıklığa sahiptir [87]. Resnet50, 50 katmandan oluştuğu için bu şekilde isimlendirilmiştir. 49 evrimsel katman ve 1 tam bağlantılı katmana sahiptir.

Daha derin özelliklere sahip ResNet daha fazla katmana sahiptir. ResNet'te eğitim hatasını azaltmak için artık bloklar kullanılmakta, derinliği azaltmak için ise eğitim katmanlarında rastgele düşürme yapılmaktadır [119]. Hata eşiği insanın hata eşiğinin altına düşerek %3,57 lere gelmiştir. Resnet50, Resnet101, Resnet152 gibi çeşitleri bulunmaktadır. ResNet ilk 34 katmanlı ağ mimarisi Şekil 2.25'te gösterilmiştir.



Şekil 2.25. ResNet ilk 34 katmana ait ağ mimarisi ile düz ağın karşılaştırılması[118]

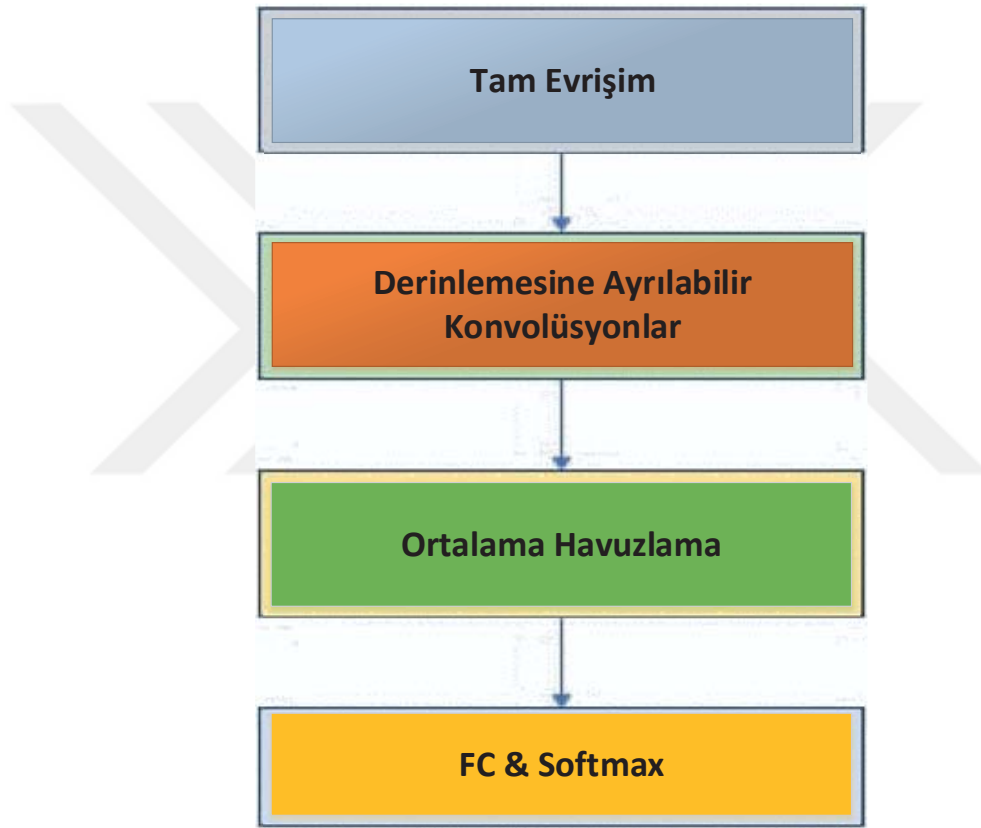
ResNet, 1×1 ve 3×3 evrim katmanlarına sahip bir dizi mimarisine sahiptir. Bu evrim katmanları, ağın katman karmaşıklığının azaltılmasına ve yüksek seviyeli özellik haritaları çıkarılmasına yardımcı olan evrim filtreleri gibi kullanılmaktadır. ResNet ağı çıktı olarak tam bağlı katmanların aktivasyonu olarak kullanılan bir çeşit bilgi bankası kullanmaktadır. Bu ağ, nesnenin tam konumunu gösteren birçok bilgiyi saklayabilmektedir.

Resnet, hata oranı %3.57 olarak elde edilmiştir ve insan hata oranı geçilerek başarımını kanıtlamıştır. ResNet, ImageNet gibi büyük veri kümelerinde kullanılarak, yüksek performans göstermiştir. Özellikle, ResNet-152 adlı modelde, ImageNet veri kümesinde nesne tanıma görevinde, %3.6'dan daha düşük hata oranına sahip bir performans elde edilmiştir [118]. Sonuç olarak, ResNet, derin öğrenme alanında önemli bir adımdır ve derinlik etkisi sorununu çözme amaçlayan yapısı ile yüksek performans göstermiştir.

2.5.6. MobileNet

MobileNet, Google arařtırmacıları tarafından geliştirilmiřtir. Sınırlı kaynakları göz önünde bulundurarak doęruluęu etkin řekilde arttırmak üzere geliştirilmiř öncelikle mobil olan kaynak dostu bir mimaridir [120].

MobileNet adından da anlařılacaęı gibi mobil uygulamalarda kullanılabilmek için tasarlanmıřtır. MobileNet derinlemesine ayrılabilir evriřimler kullanmaktadır. Bu sayede belleęin daha verimli kullanılmasına izin vermektedir. Aynı karmařıklık altında benzer mimarilere oranla oldukça hızlıdır [120]. řekil 2.26'da MobileNet temel modelini içeren adımlar gösterilmektedir.



řekil 2.26. Mobilenet v1 temel modeli [121]

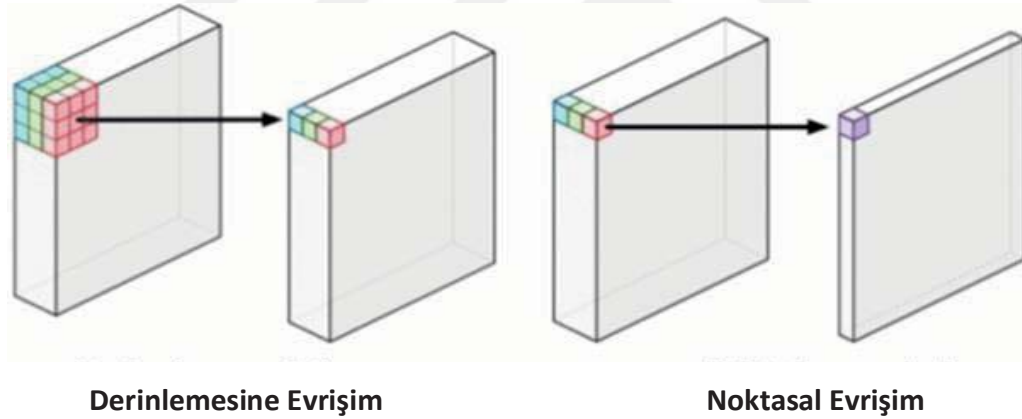
Önceki derin öğrenme modellerinde daha yüksek doęruluk oranı elde etmek amacıyla yapılan büyük deęişiklikler modelleri daha derin ve karmařık hale getirmiřtir. Ortaya çıkan bu derin ve karmařık modeller nedeniyle daha gelişmiř donanımlar kullanılmaya başlandı [118]. Ancak bütün bunlar hesaplama maliyetlerine arttırarak, kaynak kısıtlı mobil ve gömülü platformlarda donanım daęıtımı için verimlilięi düşürmüřtür. Özellikle nesne tanıma, görüntü sınıflandırma içeren gerçek zaman uygulamalarında çıkarım süresi ve doęruluęu oldukça önemlidir. Bu nedenle özellikle gerçek zamanlı uygulamalarda doęruluęu yüksek, daha az bellek karmařıklıęı gerektiren,

daha kısa hesaplama süresine sahip bir model kullanılması önemli bir gereklilik haline dönüşmüştür. Bu nedenle daha az parametre kullanarak hesaplama süresini, kısaltan MobileNet geliştirilmiştir. MobileNet standart evrişim katmanlarını kullanmayıp, derinlemesine ayrılabilen evrişimleri kullanmaktadır [122]. Bu sayede hesaplama maaliyeti önceki mimarilere oranla 8 kat azalmıştır.

MobileNet'in iki hiperparametresi vardır. Bunlar genişlik çarpanı ve çözünürlük çarpanı olarak sıralanmaktadır. Ancak bu parametreleri ağın boyutunu azaltmak için kullandığımızda, doğruluk önemli oranda düşmektedir [118].

Genişlik çarpanı kullanılan filtre sayısını azaltarak ağı her katmanda eşit şekilde basitleştirmektedir. Çözünürlük çarpanı ise görüntü çözünürlüğünü azaltmak için kullanılmaktadır. Genişlik çarpanı ve çözünürlük çarpanı değerleri değiştirilerek daha küçük ve basit modeller elde edilebilmektedir, ancak bu durum doğruluk değerlerinde önemli düşüşe sebep olmaktadır.

MobileNet mimarisinde toplu normalleştirme ve REDB aktivasyon fonksiyonu her evrişim katmanından sonra kullanılmıştır [120]. Şekil 2.27'de derinlemesine ve noktasal evrişim gösterilmektedir.

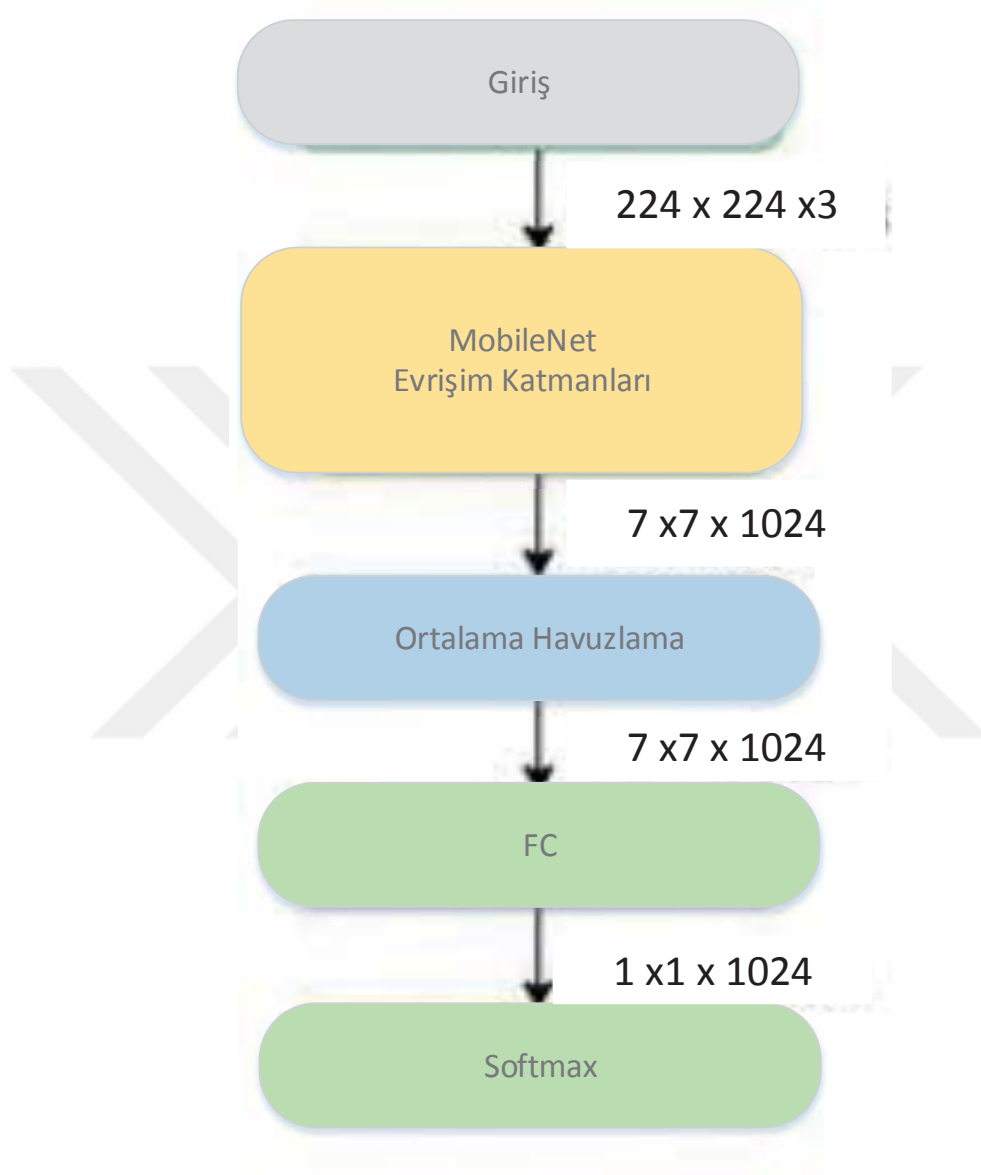


Şekil 2.27. Derinlemesine ayrılabilir evrişim adımı [122]

İlk evrişimde ve ayrıca derinlemesine evrişim katmanlarında basamaklı evrişimler kullanılmaktadır. Bu katmanları ortalama havuzlama katmanı, tam bağlantılı katman ve softmax sınıflandırıcı takip etmektedir. Derinlemesine evrişim ve noktasal evrişim katmanları ile birlikte temel MobileNet'te toplamda 28 katman bulunmaktadır [123].

Derinlemesine evrişim, yeni özellikler oluşturmadan girdiyi filtreleme adımdır. Böylece, noktasal evrişim adı verilen yeni özellikler oluşturma süreci birleştirilmiştir. Son olarak, iki katmanın kombinasyonu derinlemesine ayrılabilir evrişim olarak adlandırılmıştır. Bu model, her giriş kanalı başına tek bir filtre uygulamak için derinlemesine evrişimleri kullanmaktadır. Ardından

derinlemesine katmandan doğrusal bir çıktı kombinasyonu oluşturmak için 1×1 evrişimi (nokta yönünde) kullanmaktadır. Her evrişimden sonra Yığın Normalleştirme ve Düzleştirilmiş Doğrusal Birim (REDB) kullanılmıştır. Şekil 2.28, tüm katmanlarla birlikte MobileNet temel mimarisinin şematik bir diyagramını göstermektedir.



Şekil 2.28. Mobilenet mimarisi [124]

3. ÇOKLU VIDEO GÖRÜNTÜLERİNDE ARAÇ TAKİP VE TESPİT SİSTEMİ

Bu bölümde tez çalışması kapsamında kullanılan yöntemler hakkında genel bilgiler verilmiştir. Çalışmada geliştirilen algoritmalar ve eğitim sonuçları sunulmuştur. Çalışmayı oluşturan hedef takibi, hedef tespiti ve bölgede bulunan kameraların tespiti aşamaları başlıklar halinde incelenmiştir.

3.1. Çalışmaya Genel Bakış

Bu çalışmanın amacı bir hedefin birden farklı video görüntüsünde takibini sağlamaktır. Bunun için güvenlik ve trafik kontrolü amacıyla kullanılan, Elazığ iline ait farklı MOBESE (MOBil Elektronik Sistem Entegrasyonu) kameralarından alınan görüntüler kullanılmıştır.

Bu çalışma üç ana aşamadan oluşmaktadır. Birinci aşamada araç takibi, ikinci aşamada takip edilecek kameraların belirlenmesi, üçüncü aşama ise araç tespiti gerçekleştirilmektedir.

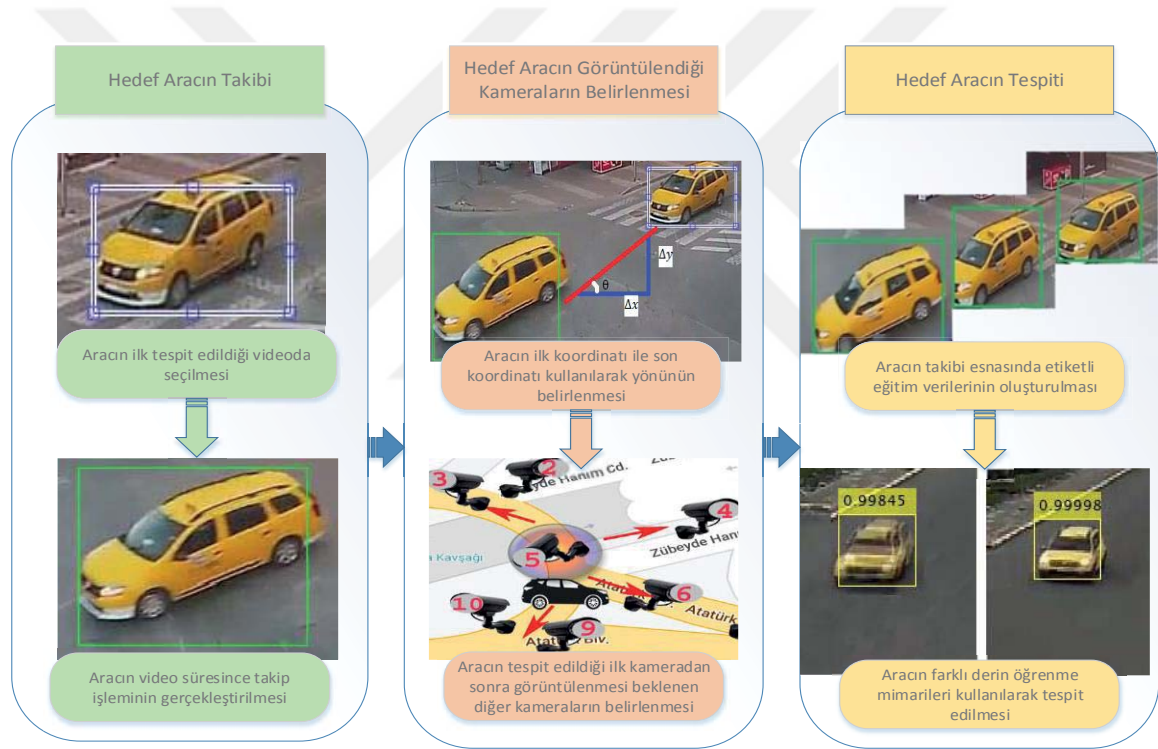
Araç takibi aşamasında, hedef olarak belirlenen araç tespit edildiği ilk kamera kaydında seçilerek kayıt süresince takip edilmektedir. Bu süreçte hedef araca ait bilgiler elde edilmektedir. Ayrıca çalışma kapsamında hedef takibi yaklaşımlarından arka plan çıkarımı yöntemi geliştirilerek yeni bir hedef takip yöntemi sunulmuştur. Bu yöntemle video süresince engellere takılmadan ve hedefi kaybetmeden takip işlemi gerçekleştirilmektedir.

Çalışmanın ilk aşamasında hedef araç ilk videoda seçilerek sınırlayıcı kutu ile çerçevelendirilmiştir. Elde edilen video görüntülerinden ilkinde, seçilen hedef araç geliştirilen takip yöntemi ile video süresi boyunca hem takip edilmiş hemde sınırlayıcı kutu ile takip edilen araç sınırlandırılmıştır. Videoyu oluşturan her çerçeve ayrı ayrı resim olarak kaydedilmiştir. Ayrıca her video çerçevesinde tespit edilen araca ait çeşitli veriler kaydedilmiştir. Bu kayıtlar, sınırlayıcı kutuya ait veriler, aracın eğimi, yönü, hızı, video çerçevesinin kaydedildiği dosya konumu gibi bilgilerden oluşmaktadır. Kayıtlar, hedefin görüntülediği bir sonraki kameranın belirlenmesi ve bu kameradan elde edilen videoda hedefin tespiti amacıyla kullanılmıştır. Özellikle aracın, her video çerçevesinde belirlenen konumuna ait koordinatlar, kullanılacak derin öğrenme yöntemi için eğitim verisi oluşturmak amacıyla kullanılmıştır. Böylece oluşturulan veri setine ait etiketli veri oluşturulması sağlanmıştır.

Çalışmanın ikinci aşamasında hedef aracın birinci kamerada görüntüden çıktıktan sonra tespit edileceği diğer kameraların belirlenmesi amaçlanmıştır. Birden fazla kameranın olduğu bir bölgede tüm kamera kayıtlarında hedefin aranması hem maliyetli hem de fazla zaman isteyen bir işlemdir. Bu nedenle hedefin aranacağı kameraların sınırlandırılması gerekmektedir. Hedef aracın yön bilgisi, ilk tespit edildiği kameranın koordinatları ve bölgedeki kameraların verileri

kullanılarak sadece belirli bölgede kullanılan kameralarda arama çalışması yapılmaktadır. Ayrıca aracın tespit edildiği kameralarda baştan sona tüm video kaydının değil sadece belirli bir süre içerisindeki kayıtların incelenmesi sağlanarak zaman kaybının azaltılması amaçlanmıştır. Her kamerada aracın yeni yönü belirlenerek bir sonraki kameranın tespitinin yolu açılmıştır.

Çalışmanın son aşamasında, aracın tespit edildiği sonraki kameralardan alınan video görüntülerinde aynı hedefi tespit edebilmek amacıyla ilk videodaki hedef araca ait veriler ve sınırlandırıcı kutu bilgileri kullanılmıştır. Hedef aracın tespit edilmesi amacıyla B-ESA, Daha Hızlı B-ESA ve YOLO gibi farklı derin öğrenme mimarileri AlexNet, VGGNet, GoogleNet, ResNet, MobileNet gibi farklı omurga ağlarıyla eğitilmiştir. Performansları kıyaslanarak en iyi performans gösteren ağ sonucu hedef tespit yöntemi olarak seçilmiştir. Şekil 3.1’de çalışma kapsamında takip edilen işlemlere ait adımlar gösterilmektedir.



Şekil 3.1. Araç takip ve tespit sistemine ait işlem adımları

3.2. Araç Takibi Algoritmasının Tanıtılması

Çalışma kapsamında Gauss karışım modeli tabanlı arka plan tespiti kullanılan bir araç takip algoritması önerilmektedir. Gauss karışım modeli, pikselin arka plana ait olup olmadığını belirlemek için kullanılan bir yöntemdir. Bu model arka plan çıkarımı amacıyla kullanılan hem en yaygın modeldir hemde ışık ve koşul değişimlerine karşı dayanıklıdır. Bir Gauss karışım modeli, Gauss bileşen yoğunluklarının ağırlıklı toplamı olan ve Denklem 3.1’de gösterilen parametrik bir olasılık yoğunluk fonksiyonudur [125].

$$p(X|\lambda) = \sum_{k=1}^M w_k N(x|\mu_k, \Sigma_k) \quad (3.1)$$

Bir Gauss karışımı modeli, M bileşeninin ağırlıkları toplamıdır. Gauss dağılımı ile hangi oranda verinin açıklanması gerektiğini belirten tahmini ağırlık değeri w_k ile ifade edilmektedir. Bu ağırlığın, t anındaki karışımda bulunan Gauss dağılımında μ_k ortalama değer ve Σ_k kovaryans matrisidir.

Sabit bir kameradan yakalanan video dizisinde ön plan piksellerini tahmin eden Gauss karışım modeli, arka plan çıkarımı yaparak ön plan nesnelerini vurgulayan bir maske oluşturmaktadır. Hareketli görüntüdeki her pikselin arka plana mı ön plana mı ait olduğunun belirlenmesi amacıyla renkli ya da daha çok gri tonlamalı bir video çerçevesi arka plan modeliyle karşılaştırılmakta ve ön plan maskesi oluşturulmaktadır. Temel olarak arka plan çıkarılarak ön olan nesnelerinin tespit edilmesi sağlanmaktadır. Hesaplanan ön plan maskesi ile takip edilecek hedeflerin çevresinde sınırlayıcı kutular oluşturulmasını sağlayan Blob analiz yöntemidir.

Tez çalışması kapsamında hedef aracın takibi için kullanılacak sınırlayıcı kutuların belirlenerek izlenmesi amacıyla Blob analiz yöntemi kullanılmıştır. Temel olarak Blob bölgesinde bulunan hareketli nesne tanınmaktadır ve bir sınırlayıcı kutu aracılığıyla işaretlenmektedir. Video çerçeveleri arasındaki geçişler ve hedefin görünür olmadığı durumlarda hedef araca dair bilgiler tutularak tekrar sahnede görünür olduğunda takibe devam edilmesi sağlıklı bir takip işleminin en önemli noktasıdır. Bu amaçla geliştirilen algorithmada Blob analiz kullanılmaktadır.

Hedef araç etrafında sınırlayıcı kutu ile belirlendikten sonra video içerisinde görüntüden kayboluncaya kadar takip edilmektedir. Şekil 3.2’de hedefin ilk video çerçevesinde seçilmesi ve diğer video çerçevelerinde takip edilmesi gösterilmektedir.



Şekil 3.2. (a) Hedefin ilk çerçevede manuel olarak seçilmesi, (b) Hedefin Blob analiz ile takip edilmesi

Bu yöntem ile tüm videoyu alıp işlemek yerine arka plan nesnelerinden ayrılmış hareketli ön plan nesnelerini içeren bir başlangıç çerçevesi elde edilmektedir. Ön plan detektörü videonun uzunluğuna göre belli sayıda çerçeveyi Gauss karışım modelini başlatmak için kullanılmaktadır. Bu

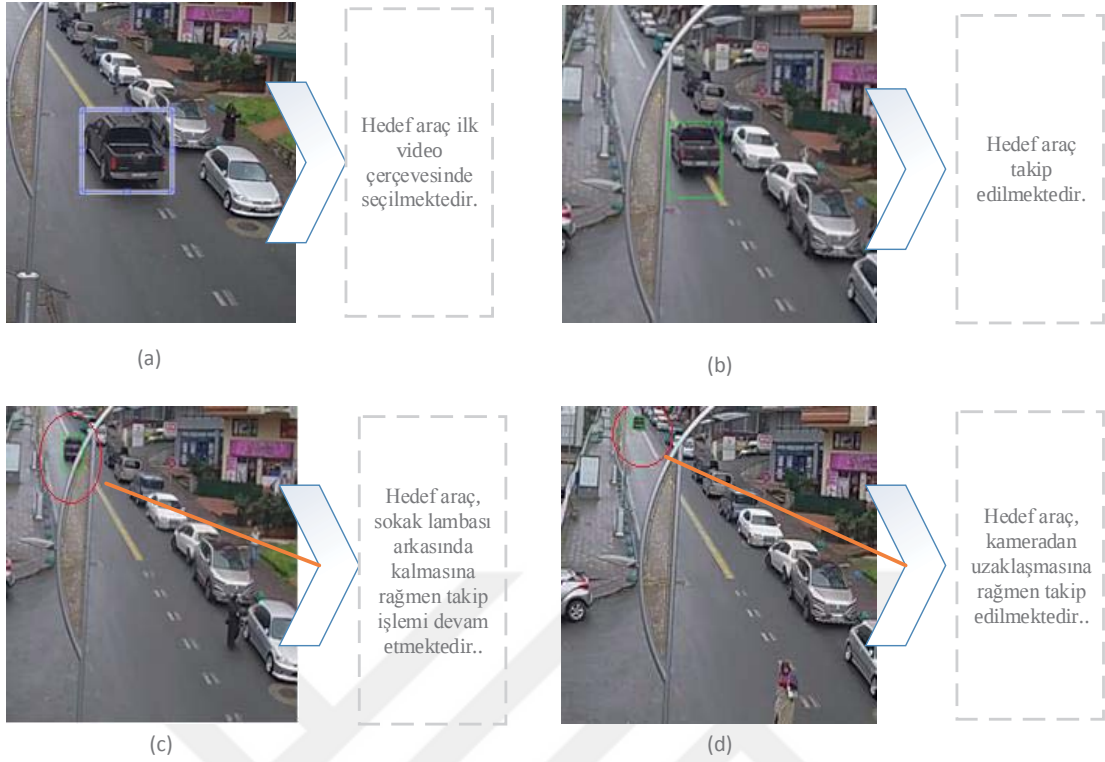
sayede arka plan çerçevesi öğrenilmektedir. Şekil 3.3.(a)'da ön plan detektörü tarafından tespit edilen ön plan görüntüsü gösterilmektedir. Ön plan belirleme işlemi çoğunlukla gürültü içermektedir. Bu nedenle gürültü gidermek ve boşlukları doldurmak için görüntü filtre ve morfolojik açma gibi çeşitli işlemlerden geçirilmiştir. Şekil 3.3.(b)'de gürültüden arındırılmış görüntü ve orijinal hali gösterilmektedir.



Şekil 3.3. Arka plan tespiti için morfoloji süreci (a) Ön plan görüntüsü, (b) Gürültü içeren arka plan görüntüsü ve filtre uygulanarak temizlenmiş görüntü

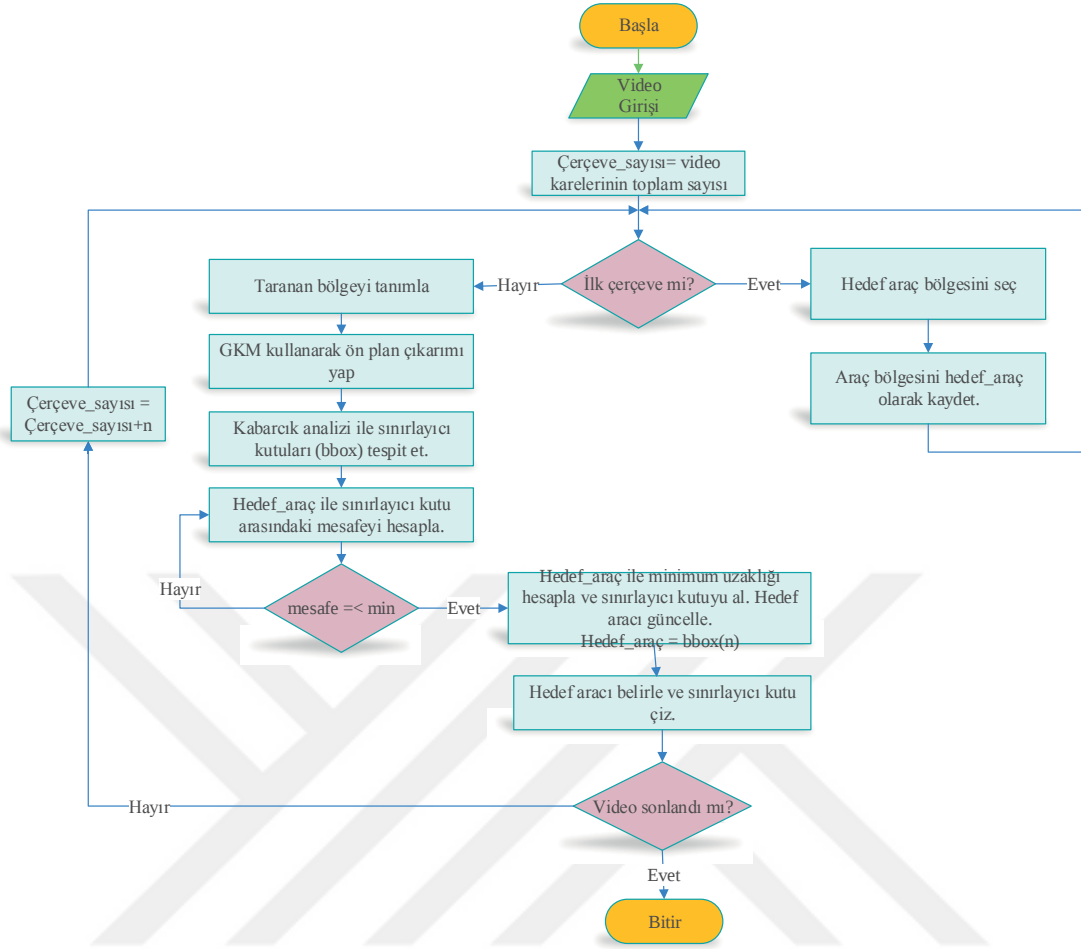
Gauss karışım modeli aynı çerçevede hareketli tüm nesneleri takip etmektedir. Bir diğer ifadeyle arka plan çıkarımı kullanılarak gerçekleştirilen araç takibinde sahnedeki tüm hareketli araçlar tespit edilmektedir. Tez çalışması kapsamında, önerilen takip algoritması ile takip edilecek tek bir araba belirlenerek sınırlayıcı kutu ile çerçevelenmektedir ve takibi gerçekleştirilmektedir.

Araç takibi uzun yıllardır üzerinde yoğun çalışmalar yapılmış bir alan olmasına rağmen hala çeşitli zorluklarla baş etmek durumundadır. Sahnede karşılaşılan engeller, araçların yakınlığı vs. gibi sebeplerle takip işlemi yarım kalmakta ve sağlıklı yürütülememektedir. Bu nedenle tez çalışması kapsamında geliştirilmiş Gauss karışım modeline dayalı arka plan çıkarımı yöntemi sunulmuştur. Bu sayede ilk video çerçevesinde seçilen tek bir görüntünün takibi sağlanmıştır. Ayrıca takip esnasında hedef aracın herhangi bir engelin arkasında kalması ya da başka bir aracın çok yakından geçmesi ve görüntüyü engellemesi gibi zorlayıcı durumlara rağmen aracın takibine devam edildiği görülmüştür. Şekil 3.4'te hedef aracın ilk video görüntüsünde seçilmesi ve takip edilmesi aşamaları ile bir engel ile karşılaşıldığı durumlarda takibe devam edildiği gösterilmektedir.



Şekil 3.4. Hedef aracın takip işlemi (a) Araç seçimi, (b) Takip işlemi, (c) Takip edilen hedefin engellere karşı dayanıklılığı, (d) Takip edilen hedefin boyut değişimi ve uzaklığa dayanıklılığı

Hedef takibini zorlaştıran etkenlerden bazılarında hedefin uğradığı biçimsel değişiklikler, büyüme ve küçülmesi, yön değiştirmesi olarak sıralanabilmektedir. Özellikle trafik senaryolarında araçların aşırı hızlı olması, kameradan aşırı uzaklaşması hedefin kaybedilmesine neden olmaktadır. Önerilen hedef araç takip sisteminin hedefin uzaklaşması durumunda bile takip işlemine devam ettiği, hedefin kameradan uzaklaşmasına rağmen kaybedilmediği görülmüştür. Şekil 3.5'te önerilen hedef takip algoritmasına ait akış diyagramı gösterilmektedir.



Şekil 3.5. Hedef araç takibi algoritmasına ait akış diyagramı

Gauss karışım modeline dayalı araç takip aşamalarına ait sözde kod Algoritma 1’de gösterilmektedir. Sözde kod aşağıda ki gibidir:

Algoritma 1: Araç Takibi

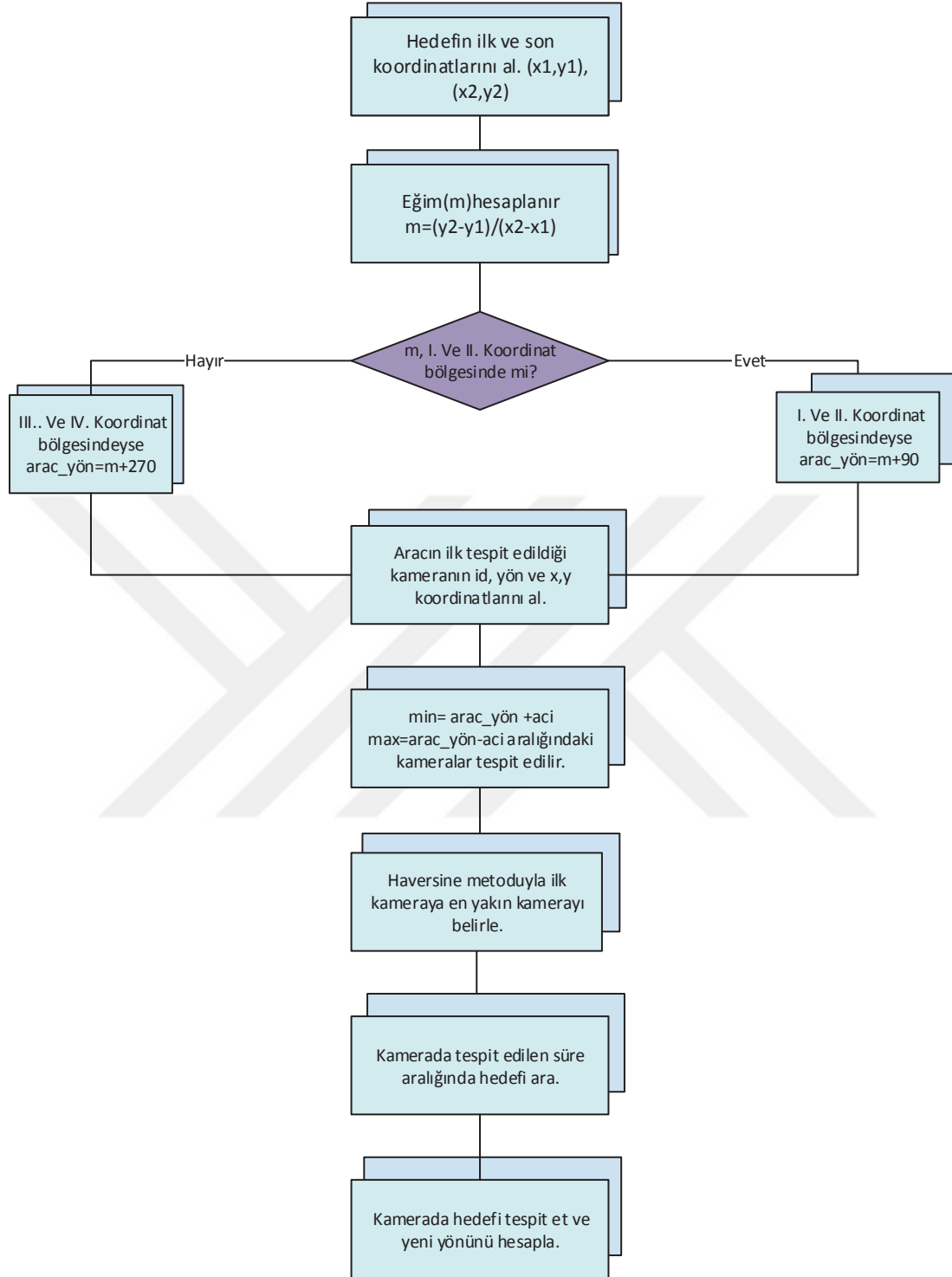
Giriş: Video çerçeveleri $n = \{1, \dots, N\}$, hedef araç

Çıkış: Takip ve tespit edilen araçlar, Sınırlayıcı kutular: video çerçevesindeki sınırlayıcı kutular

```
1: for i= 1 to n //Videodaki çerçevelerin sayısı
2:   if i==1
3:     hedef_arac ← ilgi bölgesi //Çerçeve de ki nesne bölgesi
4:   else
5:     önplan ← Önplandetektörü(n)
6:     filtrelenmiş önplan; //Gürültünün giderilmesi için
        morfolojik filtre
7:     kutular = videodaki her çerçeve (Blob_Analiz, filtrelenmiş
        önplan)
8:   for i= 1 to sınırlayıcı_kutular
9:     D(i) = mesafe (hedef_arac, sınırlayıcı_kutu(i)) //Önerilen
        uzaklık algoritması kullanarak mesafe hesaplama
10:    enkucuk_mesafe ← D dizisi
11:    hedef_arac ← kutular(min_mesafe)
12:    Sınırlayıcı kutu belirle()
13:    Video çerçevelerini kaydet //Hedef aracın bulunduğu her
        video çerçevesinin resim olarak kaydedilmesi
14:    Sınırlayıcı kutu verilerini kaydet
15:  end
16: end if
17: end
18: Sonuçları görüntüle
```

3.3. Hedef Aracın Görüntüleneceği Kameraların Tespiti

Araç takibi uygulamasının bir diğer önemli aşamasında hedef aracın geçtiği kameraların belirlenmesidir. Tez çalışmasının bir önceki aşamasında hedef, tespit edildiği kameralarda görüntüden çıkıncaya kadar takip edilmiştir. Sonraki aşama ise hedefin görüntüleceği düşünülen kameraların belirlenmesi adıımıdır. Çalışmanın ana amaçlarından biri hedef aracın birden fazla kameralarda takip ve tespit edilmesidir. Hedef aracın bir güzergâh boyunca varış noktasına kadar takibini gerçekleştirebilmek için geçtiği yol üzerinde görüntüleceği düşünülen kameraların belirlenmesi gerekmektedir. Bu sayede hedef tüm kameralarda değil sadece belirli kameralarda aranacak ve zamandan tasarruf sağlanacaktır. Kameraların tespiti için oluşturulan algoritma sayesinde kameraların belirlenmesinde ve incelenmesinde insan denetimi ortadan kaldırılmış olacaktır. Bu bölümde belirlenen hedef aracın görüntüleneceği ve takip edileceği kameraların belirlenmesi için kullanılan yöntem açıklanacaktır. Şekil 3.6’da kameraların tespit sürecine dair akış diyagramı sunulmuştur.



Şekil 3.6. Kameraların tespit sürecine ait akış diyagramı

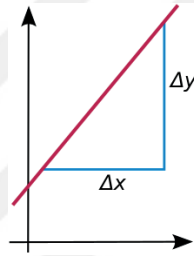
3.3.1. Araç Yönünün Belirlenmesi

Aracın ilk tespit edildiği kamerada öncelikle takibi gerçekleştirilmiş ve eğitim verileri elde edilmiştir. Ayrıca bu takip işlemi sırasında takip edilen hedef aracın yön bilgisi elde edilmektedir. Hedef aracın takibi süresince sınırlayıcı kutu bilgileri de kaydedilmektedir. Tespit edildiği kamera

görüntülerinde ilk video çerçevesinde seçilen hedef aracın sınırlayıcı kutu koordinatları ile takip edilmesi sonucunda belirlenen en son sınırlayıcı kutu koordinatları kullanılarak eğim bilgisi elde edilmektedir. Aracın ilk koordinatları (x_1, y_1) ve son koordinatları (x_2, y_2) olarak kabul edildiğinde iki noktası bilinen doğrunun eğimi formülü kullanılarak aracın eğimi bulunmaktadır. Denklem 3.2’ de eğimin hesaplanmasını gösteren formül gösterilmektedir.

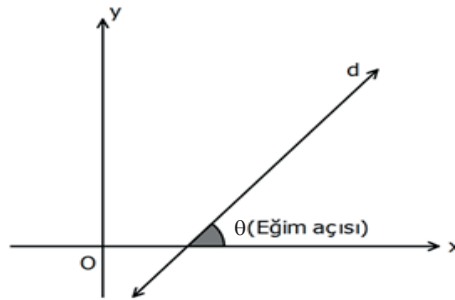
$$m = \frac{y_2 - y_1}{x_2 - x_1} = \frac{\Delta y}{\Delta x} \quad (3.2)$$

Hedefin, koordinat sisteminde yer alan y eksenindeki karşılığını ifade eden değerlerinin farkı Δy , x eksenindeki karşılığını ifade eden değerlerinin farkı Δx olarak ifade edilmektedir. Bu iki ifade kullanılarak eğim hesabı yapılmaktadır. Şekil 3.7’de, Δy ve Δx değerlerinin eksenler üzerinde gösterimi verilmiştir.



Şekil 3.7. Δy ve Δx değerlerinin eksenler üzerinde gösterimi [126]

Eğim bilgisi kullanılarak önce eğim açısı hesaplanmaktadır. Eğim açısı bir doğrunun pozitif x eksenine yaptığı açıdır. Bu doğrunun eğimi, bu eğim açısının tanjant değerine eşittir. Şekil 3.8’de eğim açısı gösterilmektedir.



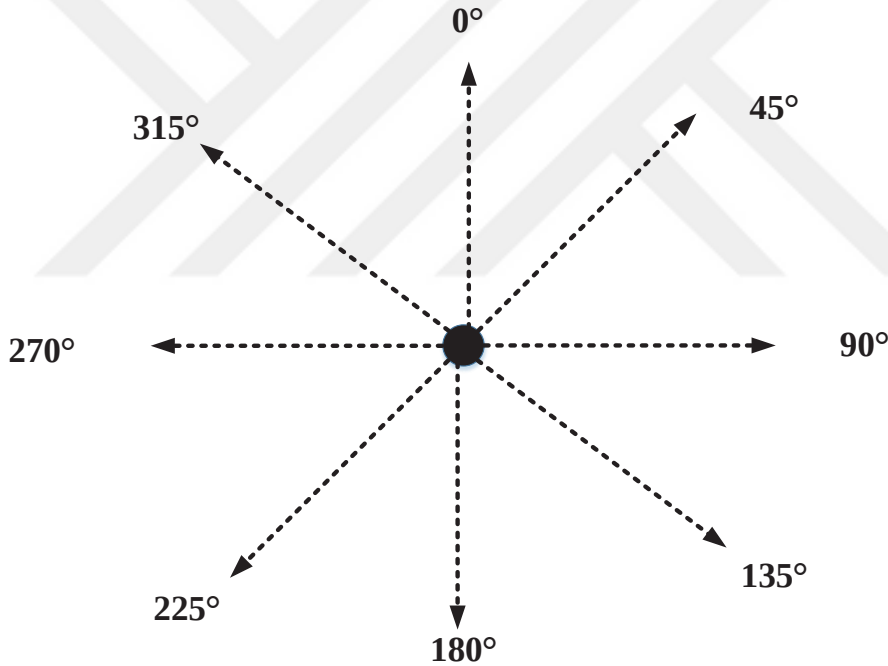
Şekil 3.8. Eğim açısı gösterimi

Bu açı θ olarak kabul edilirse eğim, θ açısının arctan değerine eşittir. Hedef aracın takip ettiği yolu bir doğru olarak kabul ederek eğim açısı hesaplanmaktadır. Hedef aracın eğimi bulunduğundan sonra bulunan eğim açısı ile aracın gidiş yönü elde edilmektedir. Hedef aracın eğim açısı, aracın gerçek açısını bulmak amacıyla kullanılmaktadır. Hedef aracın gerçek açısı ise aracın ilerlediği yönü bulmak için kullanılmaktadır. Denklem 3.3'te eğim ve eğim açısı değerlerinin hesaplanması gösterilmektedir.

$$m = \tan \theta$$

$$\theta = \arctan m \quad (3.3)$$

Aracın yön bilgisi bölgedeki aynı yönde bulunan kameraların tespiti için önemli bir veridir. Ekranın yön bilgisi Şekil 3.9'da gösterildiği şekliyle kabul edilerek aracın yön bilgisi elde edilmektedir.



Şekil 3.9. Ekran yön bilgisi içeren koordinat sistemi

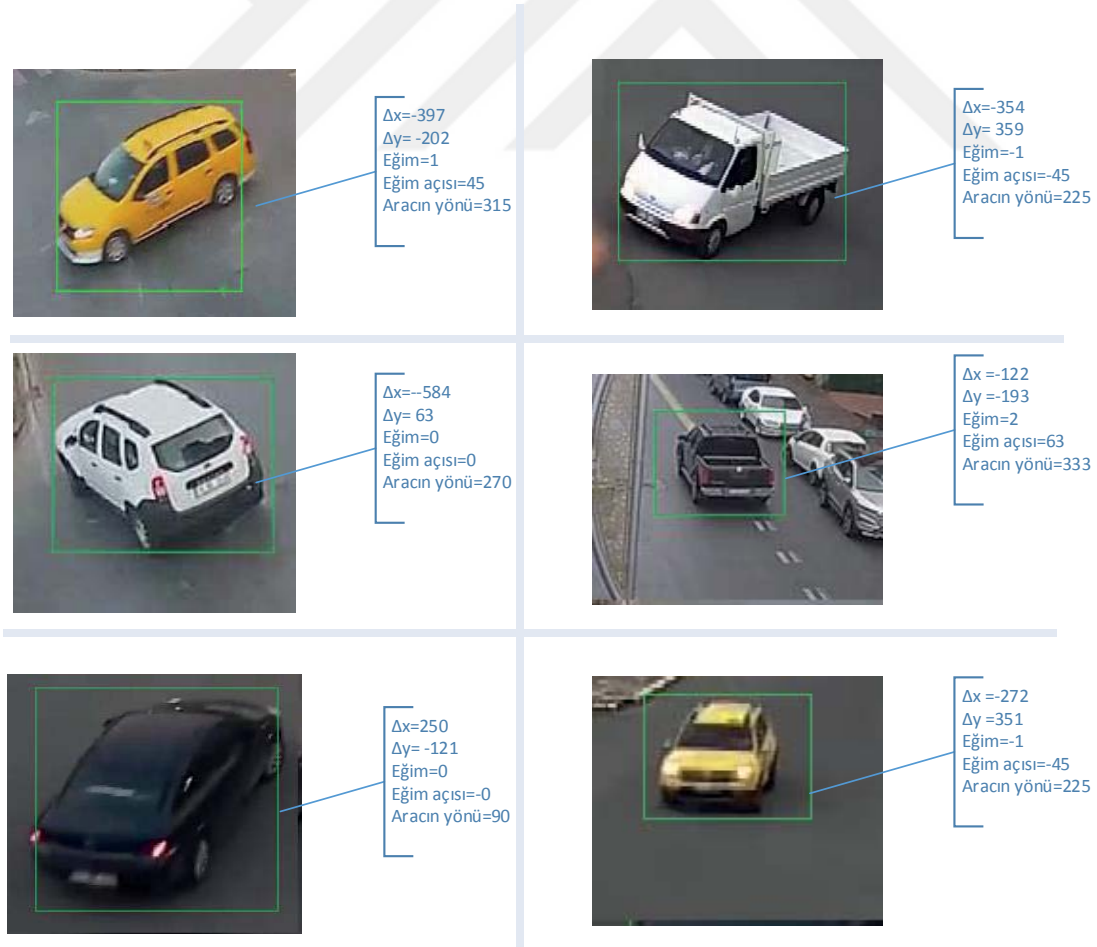
Hedef aracın eğim değeri ve ardından hedef açısı belirlendikten sonra Δy ve Δx değerlerinin Şekil 3.8'de hangi bölgede oldukları belirlenmektedir. Eğer bu değerler koordinat sisteminin 1. ve 2. bölgesinde ise +90, 3. ve 4. bölgesinde ise +270 eklenerek aracın koordinat sistemine göre yönü belirlenmektedir. Ayrıca Δy ve Δx değerlerinin Şekil 3.9'daki koordinat sisteminde bulundukları konuma göre +90 veya +270 eklenme durumları hesaplanmaktadır. Tablo 3.1'de Δy ve Δx değerlerinin pozitif ve negatif olarak aldıkları değerlere göre eklenecek açı

değerleri ile örnek eğim ve eğim açısı değerleri gösterilmektedir. Ayrıca tabloda örnek eğim değeri, örnek değerin arctan'ı olan eğim açısı verilmektedir.

Tablo 3.1. Δy ve Δx değerlerinin koordinat ekseninde bulundukları bölgeye göre alacakları değerler

$\frac{\Delta y}{\Delta x}$	Örnek eğim	Eğim açısı	Eklenecek açı değeri
$\frac{+}{+}$	1	45	+90
$\frac{-}{+}$	-1	-45	+90
$\frac{-}{-}$	1	45	+270
$\frac{+}{-}$	-1	-45	+270

Trafik görüntülerinde takip edilen aracın eğimi ve eğim açısı bulunduktan sonra aracın yönü tespit edilmektedir. Şekil 3.10'da farklı yönlere gitmekte olan araçların eğim, eğim açısı ve yön bilgileri gösterilmektedir.



Şekil 3.10. Takip edilen farklı araçların eğim ve yön bilgileri

Algoritma 2’de hedef aracın yönünün bulunması için önerilen algoritmaya dair sözde kod verilmiştir.

Algoritma 2. Hedefin ilk tespit edildiği kamerada yönünün bulunması

Giriş: Video çerçeveleri $n = \{1, \dots, N\}$, hedef araç

Çıkış: Tespit edilen araç, hedef araca ait sınırlayıcı kutu bilgileri, m : eğim, araç_yön: hedef aracın yönü

```
01. function araç_izleyici()
02.     hedef_bölge_koordinatları←pozisyonunu al()
03.     x1← hedef_bölge_koordinatları(x)
04.     y1← hedef_bölge_koordinatları(y)
05.     for i = 1 to n
06.         bbox ← sınırlayıcı_kutu(n) //son video çerçevesinde takip
            edilen araca ait sınırlayıcı kutu koordinatları alınır.
07.     end
08.     x2 ← sınırlayıcı_kutu(x)
09.     y2 ← sınırlayıcı_kutu(y)
10.     Δy ← (y2-y1)
11.     Δx ← (x2-x1)
12.     m ← Δy/Δx
13.     vehicle_direction=arctan(m)
14.     if (Δy >0) and (Δx >0)
15.         araç_yön← araç_yön +90 eklenmektedir.
16.     elseif (Δy <0) and (Δx >0)
17.         araç_yön ← araç_yön +90 eklenmektedir.
18.     elseif (Δy <0) and (Δx <0)
19.         araç_yön ← araç_yön +270 eklenmektedir.
20.     elseif (Δy >0) and (Δx <0)
21.         araç_yön ← araç_yön +270 eklenmektedir.
22.     end if
23. end
24. end
```

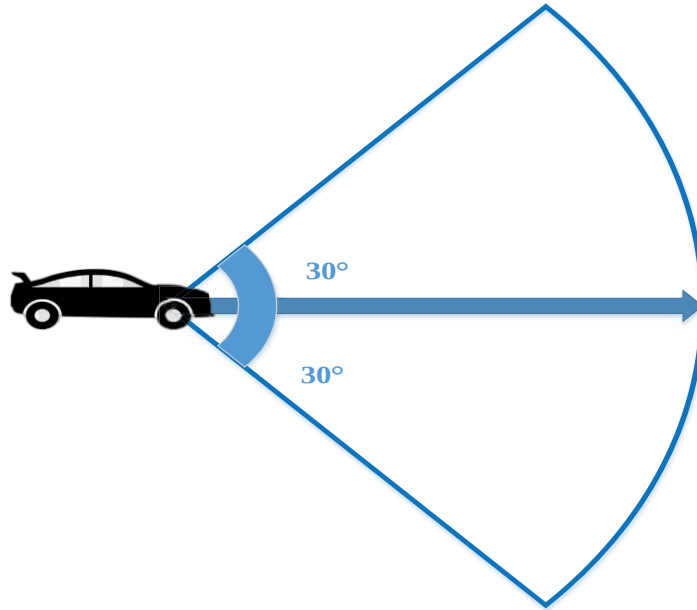
3.3.2. Aracın Gidiş Yönünde Bulunan Kameraların Tespiti

Hedef aracın yönü bulunduktan sonra gidiş yönünde ve belirlenen bir aralıkta ki kameralar tespit edilir. Toplam 40 kameraya ait id, yön ve x, y düzlemindeki coğrafi koordinatları bilinmektedir. Şekil 3.10’da 40 kameranın bölge üzerinde yerleşimi görülmektedir.



Şekil 3.10. Bölgedeki kameraların yerleşimi ve id numaraları

Hedef aracın ilk tespit edildiği kameraya ait bilgiler ve aracın yön bilgisi kullanılarak hedef takibinde hedef aracın görüntülenebileceği yeni kameraların tespit edilmesi sağlanmıştır. Bu amaçla aracın yön bilgisine $+30$ ve -30 açı değerleri eklenerek oluşan aralıktaki kameraların, hedef aracın gözlemlenebileceği aralık olarak belirlenmesi sağlanmaktadır. Böylelikle, bu aralıktaki kameralara ait bilgiler yeni bir diziye aktarılmaktadır. Şekil 3.11’de bir sonraki kameranın aranacağı bölge gösterilmektedir.



Şekil 3.11. Hedefin tespit edileceği tahmin edilen kameranın aranacağı bölge

Bölgedeki tüm kameralara ait veriler elde edilmiştir. Bu veriler daha sonra aracın takip yönüne göre belirlenecek kameraların seçiminde kullanılmıştır. Bu veriler Tablo 3.2’de gösterilmektedir.

Tablo 3.2. Bölgedeki kameraların id, yön ve koordinat bilgileri

Kamera id	Kamera yönü	x koordinatı	y koordinatı
1	60	38.6728574127779	39.1929757298937
2	280	38.6725915405053	39.1942325377230
3	65	38.6723276860107	39.1939643168313
4	270	38.6716282574858	39.1936317229256
5	60	38.6708659982578	39.1927358651472
6	65	38.6702000618502	39.1930684590530
7	60	38.6701665554522	39.1925642037765
8	290	38.6711340463840	39.1924891020279
9	220	38.6718837381583	39.1918775583670
10	290	38.6722439224757	39.1923067118133
11	300	38.6727381259155	39.1917488123331
12	65	38.6729977907624	39.1908422256780
13	300	38.6726334220817	39.1907242084802
14	85	38.6721643470266	39.1905954624464
15	290	38.6719842548943	39.1910514379830
16	90	38.6715337393682	39.1904475782001
17	85	38.6713494574181	39.1908928249005
18	80	38.6711735514777	39.1914024446180
19	85	38.6708552443917	39.1917564962111
20	270	38.6704741116769	39.1917457673750
21	90	38.6704489819760	39.1900076959176
22	95	38.6711065395772	39.1900827977707
23	310	38.6717305854728	39.1902759168215
24	305	38.6717431500994	39.1896590087426
25	95	38.6712782574466	39.1896697375787
26	315	38.6705788186688	39.1894283387652
27	310	38.6711191043133	39.1892834994771
28	270	38.6714415984524	39.1894927117821
29	100	38.6717431500994	39.1882535312061
30	225	38.6709976451051	39.1880818698276
31	135	38.6706625841436	39.1878780219511
32	220	38.6704029108228	39.1887148711724
33	265	38.6702521323355	39.1872450206190
34	260	38.6705034296380	39.1869767997151
35	270	38.6719567484020	39.1876473519749
36	125	38.6719902539622	39.1866334769581
37	310	38.6725975395224	39.1869928929693
38	90	38.6724342011541	39.1876849029014
39	85	38.6729032744409	39.1883661839973
40	80	38.6730205422824	39.1890260074210

Bu bölümde aracın gidiş yönündeki kameraların tespitine ait sözde kod Algoritma 3’de verilmiştir.

Algoritma 3. Bölgedeki kameraların tespiti

Giriş: araç_yön: hedef aracın yön bilgisi, kamera_özellik: hedef aracın tespit dildiği ilk kameraya ait id numarası, yön ve koordinat bilgileri, en yakın kameraların atanacağı boş dizi

Çıkış: yakınkamera: hedef aracın gidiş yönünde bulunan kameralara ait bilgiler, cikti: ilk tespit edilen kameraya en yakın kamera bilgisi

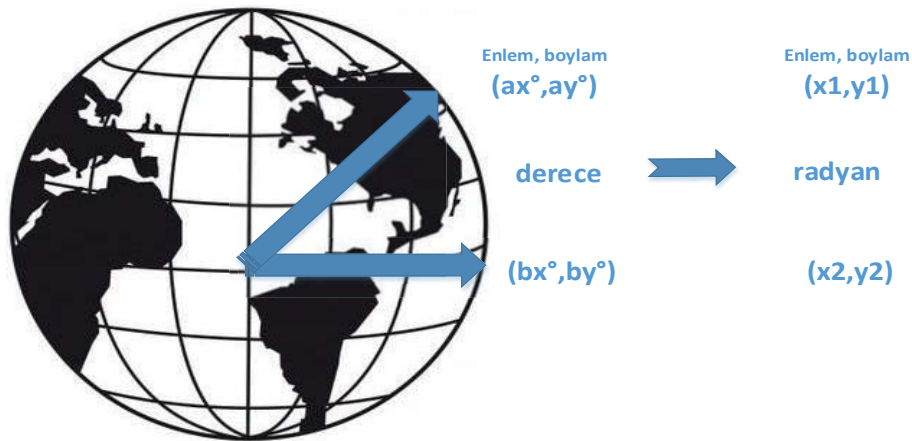
```
01. function kamera_tespiti( araç_yön, kamera_özellik )
02.   yakınkamera=[];
03.   aralık_min= araç_yön, -30
04.   aralık_max= araç_yön +30
05.   kamera_özellik =[1 60 38.672857412777944 39.192975729893696
                        2 280 38.67259154050532 39.19423253772303
                        3 65 38.67232768601066 39.193964316831305
                        4 270 38.671628257485786 39.193631722925566
                        5 60 38.670865998257774 39.192735865147206
                        6 65 38.6702000618502 39.19306845905295
                        7 60 38.67016655545218 39.1925642037765
                        8 290 38.67113404638403 39.19248910202792
                        9 55 38.6718837381583 39.19187755836702
                        10 290 38.67224392247566 39.192306711813266
                        ...]
06.   Satır_sayısı←kamera_özellik matrisinin satır sayısı
07.   for i = 1 to satır_sayısı do
08.     if kamera_özellik _yön> aralık_min & kamera_özellik
        _yön < aralık_max
09.       if araç_yön koordinat ekseninin 1. bölgesinde ise ve
        (araç_yön (x koordinatı)< kamera_özellik (x koordinatı) &
        araç_yön (y koordinatı)< kamera_özellik (y koordinatı)
10.         yakınkamera = kamera_özellik id, yön ve x-y koordinat
        bilgileri atanır.
11.       else if araç_yön koordinat ekseninin 2. bölgesinde ise ve
        (araç_yön (x koordinatı)> kamera_özellik (x koordinatı) &
        araç_yön (y koordinatı)< kamera_özellik (y koordinatı)
12.         yakınkamera = kamera_özellik id, yön ve x-y koordinat
        bilgileri atanır.
13.       else if araç_yön koordinat ekseninin 3. bölgesinde ise
        ve (araç_yön (x koordinatı)> kamera_özellik (x koordinatı) &
        araç_yön (y koordinatı)> kamera_özellik (y koordinatı)
14.         yakınkamera = kamera_özellik id, yön ve x-y koordinat
        bilgileri atanır.
15.       else if araç_yön koordinat ekseninin 4. bölgesinde ise
        ve (araç_yön (x koordinatı)< kamera_özellik (x koordinatı) &
        araç_yön (y koordinatı)> kamera_özellik (y koordinatı)
16.         yakınkamera = kamera_özellik id, yön ve x-y koordinat
        bilgileri atanır.
17.       end if
18.     end
19.   function cikti=enyakınkamera(arac_yön, yakınkamera)
20.     cikti
21.   end
```

3.3.3. Hedef Aracın Aranacağı Yeni Kameranın Tespit Edilmesi

Hedef aracın ilk tespit edildiği kamerada takibinden sonra, bir sonraki kamerada da tespit edilmesi gerekmektedir. Böylelikle belli bir güzergâh boyunca hedefin takibi sağlıklı bir şekilde yürütülmektedir. Hedef aracın izlenmesi sırasında ilk aşamada yönü bulunmaktadır. Sonraki aşamada aracın gidiş yönünde belirlenen aralıktaki görüntülenebileceği kameralar belirlenmektedir. Diğer aşama ise hedef aracın görüntülenebileceği düşünülen yeni kameranın belirlenmesi aşamasıdır. Bölgedeki kameralar içerisinde tespit edildiği bir önceki kameranın yönü, aracın gidiş yönü ve hızı dikkate alınarak yeni kameranın belirlenmesi sağlanmaktadır. Kameralar arasından ilk kameraya en yakın olanın bulunması bu aşamanın ilk adımıdır. Bu nedenle coğrafi koordinatlar arasında ki mesafe dikkate alınarak en yakın kamera bulunmaktadır.

3.3.4. Haversine Formülü

Haversine formülü dünya üzerinde yer alan coğrafi koordinatlar arasında ki mesafeyi ölçmek için kullanılan formüldür [127]. Başlangıç ve bitiş noktaları enlem ve boylam bilgileri olan iki nokta arasında mesafe hesaplamak için kullanılan birçok yöntem arasından en doğru sonucu vermektedir [128]. Coğrafi özelliklerden dağ yükseltileri veya vadi derinliklerini dikkate almayarak iki boyutlu harita üzerinde iki koordinat arasındaki mesafeyi hesaplamak için kullanılan yöntemlerden biridir. Ölçülen mesafe, dünyanın küresel olarak trigonometrik gerçek mesafesidir. Haversine formülü ile aynı zamanda büyük daire mesafesinde ölçülmektedir. Bu nedenle büyük daire mesafesi olarak da adlandırılmaktadır [129]. Bu formül enlem ve boylamı bilinen iki noktanın arasındaki mesafe verilerini metre cinsinden üretmektedir. Haversine formülünün küresel dünya üzerinde gösterimi Şekil 3.12’de gösterilmektedir.



Şekil 3.12. Haversine formülünün küresel dünya üzerinde gösterimi [129]

Özellikle coğrafi bilgi sistemleri üzerindeki uygulamalarda sıklıkla kullanılmaktadır. Çalışma kapsamında hedef aracın tespit edildiği kamera ile bir sonraki kamera arasındaki

mesafenin hesaplanması amacıyla kullanılmaktadır. Denklem 3.4’de Haversine formülü gösterilmektedir.

$$d = 2r \arcsin \left(\sqrt{\sin^2 \left(\frac{\Delta \phi}{2} \right) + \cos(\phi_1) \cos(\phi_2) \sin^2 \left(\frac{\Delta \lambda}{2} \right)} \right) \quad (3.4)$$

D iki coğrafi nokta arasındaki mesafe, ϕ enlem, $\Delta \phi$ iki nokta arasındaki enlemler farkı, λ boylam, $\Delta \lambda$ iki nokta arasındaki boylamlar farkı, r ise Dünya'nın yarıçapıdır ve 6371 (km) olarak kabul edilmektedir.

3.3.5. Hedef Aracın Yeni Kameraya Göre Yönünün ve Aranacağı Sürenin Bulunması

Takip edilen aracın ilk tespit edildiği kamerada takibi gerçekleştirirken sınırlayıcı kutu, yön, hız gibi bilgileri elde edilmektedir. Araç ilk kamera görüntüsünden çıktıktan sonra hangi kameralarda görüntüleneceği tespit edilmelidir. Böylelikle takibin hedefin varış noktasına kadar sürdürülmesi beklenmektedir. Hedefin bir sonraki kamerada tespitinden sonra yön değiştirmesi olası olduğundan ikinci kamerada yönü tekrar belirlenmelidir. Öncelikli olarak kameranın görüş açısına girdikten sonra kameraya göre yönünün belirlenmesi önemlidir. Kameranın kendisine doğru yönelen aracı 180 derece açıyla görmektedir. Aracın tespit edilmesi için bakılacak yeni kamera belirlendikten sonra yapılacak bir diğer önemli adım ise hangi süreler arasında aracın geçtiği belirlenip o aralıkta ki kayıtların incelenmesi aşamasıdır. Bunun için bir önceki kameradan hangi sürede görüntüden çıktığının belirlenmesi önemlidir.

Şekil 3.13’te hedef aracın görüntülendiği tüm kameralarda tespit edilmesi ve kamerada kaydında hangi sürede ilk olarak görüntülendiği gösterilmektedir.



Şekil 3.13. Hedef aracın farklı kameralarda görüntülenmesi

Algoritma 4'te hedef aracın gidiş yönünde tespit edilen birden fazla kamera arasından ilk tespit edildiği kameraya en yakın olanın belirlenmesi, aracın yeni yönünün bulunması ve yeni kamerada aranacağı sürelerin belirlenmesi için izlenen işlem adımları gösterilmiştir.

Algoritma 4. En yakın kameranın belirlenmesi, aracın yeni yönünün bulunması ve hedef aracın kamera kaydında aranacağı sürenin belirlenmesi

Giriş: target_vehicle_direction: hedef aracın yön bilgisi, closecam: aracın gidiş yönündeki kameralar, ϕ_1 = enlem, λ_1 =boylam

Çıkış: d: iki kamera arasındaki mesafe bilgisi, distancecam: aracın gidiş yönündeki tüm kameraların ilk tespit edildiği kameraya uzaklıklarını içeren matris, newcam: aracın tespit edildiği yeni kamera, newdirection: aracın tespit edildiği yeni kameraya göre yönü, newcam_saat: yeni kamerada hedef aracın tespit edileceği süreler

```

01. function enyakinkamera( araç_yön, yakinkamera )
02.  $\phi_1, \lambda_1 \leftarrow$  araç_yön'ün x ve y koordinatları ilk noktanın enlem ve boylamı olarak atanır.
03. mesafematrıs=[];
04. t=1;
05. satır  $\leftarrow$  yakinkamera matrisinin satır sayısı atanır.
06. for i=1 to satır do
07.    $\phi_2, \lambda_2$  yakinkamera matrisinde x ve y koordinat değerleri ikinci noktanın enlem boylamı olarak atanır.
10.    $\Delta\phi \leftarrow \phi_1 - \phi_2$  // enlemler farkı
11.    $\Delta\lambda \leftarrow \lambda_1 - \lambda_2$  //boylamlar farkı
12.   R = 6371 dünyanın yarıçap değeri
13.    $\phi_1 = \phi_1 * \pi/180$  //birinci enleme ait radyan değerini dereceye dönüştürmektedir
14.    $\phi_2 = \phi_2 * \pi/180$  //ikinci enleme ait radyan değerini dereceye dönüştürmektedir.
15.    $\Delta\phi = (\Delta\phi) * \pi/180$ 
16.    $\Delta\lambda = (\Delta\lambda) * \pi/180$ 
17.   a = sin( $\Delta\phi/2$ ) * sin( $\Delta\phi/2$ ) + cos( $\phi_1$ ) * cos( $\phi_2$ ) * sin( $d\Delta\lambda/2$ ) * sin( $\Delta\lambda/2$ );
18.   d = 2 * atan2(sqrt(a), sqrt(1-a));
19.   d = R * d;
20.   kamera_uzaklık  $\leftarrow$  yakinkamera matrisine ait id, yön, x ve y
21.   kamera_uzaklık  $\leftarrow$  d, mesafe matrisinin son sütununa mesafe bilgisi eklenir.
22.   t=t+1;
23. end
24. [a,b]=(min(kamera_uzaklık(:,5))) //matrisin son sütununda ki mesafe bilgilerinin min değeri bulunur.
25. yeni_kamera= kamera_uzaklık (b,1)
26. //Hedef aracın görüntülendiği yeni kamerada yönünü tespit etme
27. yön  $\leftarrow$  kamera_uzaklık matrisinin yön bilgisi alınarak aracın kameraya göre yönü bulunur.
28. yeni_yön  $\leftarrow$  yön+180
29. yeni_yön = yeni_yön -360
30. yeni_yön (i,1)=i;
31. yeni_yön (i,2)=abs((yön +180)-360);
32. // Hedef aracın yeni kamerada tespit edileceği süre
33. bas_saati, bitis_saati  $\leftarrow$  takip edilen kamerada hedefin tespit edildiği süreler
34. kameralararasısüre= kamera_uzaklık
35. yenikamera_saat=bitis_saati+ kameralararasısüre
36. end

```

3.4. Araç Tespiti ve Eğitim Süreçleri

Tez çalışmasının bu aşamasında, ilk aşamada belirlenerek takip edilen aracın diğer video görüntülerinde tespiti sağlanmaktadır. Bu aşamada derin öğrenme yöntemleri ile eğitim yapılarak aracın yeri belirlenmeye çalışılmıştır. Aracın tespit edildiği ilk video görüntüsü süresince takibi sırasında her çerçeve ve sınırlayıcı kutu bilgisi kaydedilmiştir. Aracın geçeceği ikinci trafik videosunda bu bilgiler ışığında gerçekleştirilen eğitim ile aracın yeri farklı derin öğrenme algoritmaları ile tespit edilmiştir. Bu aşamada önce B-ESA, daha sonra ise Daha Hızlı B-ESA ve YOLO derin öğrenme modelleri, AlexNet, VGG16, VGG19, ResNet50, ResNet101, GoogleNet ve MobileNet mimarileri ile eğitilerek karşılaştırılmıştır.

3.5. Deneysel Sonuçlar

Araç tespiti aşamasında araca ait veriler derin öğrenme algoritmalarıyla eğitilmiş ve karşılaştırılmıştır. Çalışma kapsamında en hızlı ve verimli algoritma seçilerek araç tespit ve takip sistemi için kullanılmıştır.

3.5.1. Eğitim Platformu ve Parametre Ayarları

Araç tespiti aşamasında deney platformu olarak Matlab 2021a sürümü kullanılmıştır. Tüm deneyler İşlemci Intel(R) Xeon(R) W-2245 CPU @ 3.90GHz, 3912 Mhz, 8 Çekirdek, 16 Mantıksal işlemci ve NVIDIA Quadro RTX 40000 ekran kartı ile donatılmış bir bilgisayar üzerinde gerçekleştirilmiştir. Kullanılan bilgisayara ait ayrıntılı donanım bilgileri Tablo 3.3'te gösterilmiştir.

Tablo 3.3. Deneysel çalışmalar için kullanılan bilgisayarın donanım özellikleri

Donanım	Özellik
Bilgisayar	İş İstasyonu
Merkezi İşlemci	Intel(R) Xeon(R) W-2245 CPU @ 3.90GHz işlemci
Bellek	32 GB
Sistem Türü	64 bit İşletim sistemi, x64 tabanlı işlemci
Ekran Kartı	NVIDIA Quadro RTX 40000
Sabit Disk	512 GB
İşletim Sistemi	Microsoft Windows 10 pro

Tablo 3.4'te hedef araç tespitinde kullanılan derin öğrenme modellerinden B-ESA, Daha Hızlı B-ESA ve YOLO için kullanılan parametre ayarları gösterilmiştir.

Tablo 3.4. Derin öğrenme modelleri için parametre ayarları

Özellikler	Değerler
Eğitim Tur Sayısı	50
Yineleme	8000
Öğrenme Oranı	0.001
Minimum Parti Boyutu	1
Optimize edici	sgdm
Momentum	0.9

3.5.2. Veri toplama

Video gözlem, farklı hava koşullarında ve günün farklı saatlerinde gerçekleştirilmiştir. Veri toplama adımı Elazığ ilinde MOBESE görüntüleri kullanılmıştır. Ardışık olarak yerleşen ve yolun farklı ya da aynı kısımlarını gören kameralardan alınan video görüntüleri seçilmiştir. Alınan video görüntüleri numaralandırılmıştır. Hedefin tespit edildiği ilk videolarda nesne takip işlemi gerçekleştirilmiştir. Geliştirilen yeni takip algoritması sayesinde video süresince hem hedef araç izlenmiştir hem de her video çerçevesi aracın sınırlayıcı kutu koordinatları ile kaydedilmiştir. 1280x720 çözünürlükte 74 KB görüntülere ayrılan video görüntüsü çerçeveleri çalışma içerisinde oluşturulan klasöre ardışık sayılar ile isimlendirilerek kaydedilmektedir. Bu modelde verilerin % 80'i eğitim ve % 20'si test amacıyla kullanılmıştır.

Etiketli Eğitim Verisi

Etiketli eğitim verisi, nesne tanıma algoritmalarını eğitmek ve test etmek amaçlı kullanılan verileri tanımlayan bir terimdir. Model eğitimi ve testi için gereklidir. Etiketleme, verilerin denetimli bir öğrenme modeli tarafından algılanabilmesi için atama işlemidir. Derin öğrenme modelleri ile nesne tanıma işlemi gerçekleştirilirken daha başarılı sonuçlar elde etmek için büyük miktarda etiketlenmiş veri kullanılmaktadır. Bu çalışma kapsamında hedef takibi işlemi uygulanan ilk video görüntüsünde aynı zamanda etiketlenmiş veride oluşturulmaktadır. Video takip işleminin gerçekleştiği her video çerçevesi kendisine atanan ardışık bir numara ve isim kombinasyonu ile oluşturulan bir klasöre resim olarak kaydedilmektedir. Her çerçevede takip edilen tek bir araç olduğu için ona ait sınırlayıcı kutu koordinatları kaydedilmektedir. Bir dosyanın her satırına video çerçevesine ait dosya yolu ve o çerçevede tespit edilen hedef aracın sınırlayıcı kutu bilgisi kaydedilmektedir. Böylelikle veri etiketleme işlemi otomatik hale getirilerek büyük miktarda verinin kısa sürede etiketlenmesi sağlanmaktadır.

Matlab programı içerisinde etiketleme işlemi için uygulamalar kullanılmaktadır. Bu tez çalışmasında önerilen takip algoritması, etiketlenmiş veriyi manuel olarak oluşturmanın zaman

alacağı durumlarda etiketleme uygulamaları içerisinde de yeni bir takip algoritması olarak kullanılabilir. Bu uygulamalarda ilk video çerçevesinde belirlenen hedef seçilen algoritma ile son çerçeveye kadar sınırlayıcı kutu kullanılarak otomatik olarak etiketlenmektedir. Önerilen algoritma, tüm çerçevelerde hedefin otomatik olarak etiketlenmesini sağlamada oldukça başarılı olmuştur.

3.5.3. Değerlendirme Metrikleri

Hedef tespitinde değerlendirme metriği olarak en yagın kullanılan hassaslık, kesinlik, ortalama kesinlik değerlerinin ortalaması (mAP), zaman metrikleridir. Nesne tanıma görevleri için, belirli bir birleşme üzerinden kesişim (BÜK) eşiği değerini kullanarak hassasiyet ve kesinlik değerleri hesaplanmaktadır. Doğru pozitif (DP), yanlış pozitif (YP), yanlış negatif (YN) ve doğru negatif (DN) değerleri hesaplama amacıyla kullanılmaktadır. Burada DP doğru sınıflandırmayı ifade etmektedir, YP yanlış sınıflandırılmış demektir ve DN bu kategoride kaçırılan örnek olarak ifade edilmektedir.

Hassasiyet

Tüm tahminlerden doğru pozitif (DP) olanların doğru bulunma oranıdır. Denklem 3.5'te hassasiyet oranı hesaplaması gösterilmiştir.

$$\text{Hassasiyet} = \frac{DP}{DP+YN} \quad (3.5)$$

Kesinlik

Tüm pozitif tahminlerden doğru pozitif (DP) olanların bulunma oranıdır. Denklem 3.6'da kesinlik oranı hesaplaması gösterilmiştir.

$$\text{Kesinlik} = \frac{DP}{DP+YP} \quad (3.6)$$

Ortalama kesinlik değerlerinin ortalaması (mean Average Precision-mAP)

Daha Hızlı B-ESA ve YOLO gibi birçok nesne tanıma algoritması, modellerini değerlendirmek için kullanılmaktadır. Ortalama kesinlik değerinin ortalamasını hesaplamadan önce kesinliği ve hassasiyeti hesaplamak gerekmektedir. Bu değer hassasiyet- kesinlik eğrisi altında kalan alan olarak da ifade edilmektedir. Denklem 3.7'de gösterildiği şekilde hesaplanmaktadır.

$$mAP = \frac{1}{n} \sum_{k=1}^{k=n} AP_k \quad (3.7)$$

Birleşme üzerinden kesişim (BÜK)

Birleşme üzerinden kesişim, tahmin edilen sınırlayıcı kutu koordinatları ve etiketli veri sınırlayıcı kutularının kesişme alanı olarak kabul edilmektedir. Kutuların birleşiminin toplam alana bölünmesi ile hesaplanmaktadır. Doğru pozitif (DP), birleşme üzerinden kesişim eşikleri tarafından belirlenmektedir. Bu durum gerçek konumlar ile tahmin edilen konumlar arasındaki BÜK'ün önceden ayarlanmış eşik değerden büyük olması durumunda sınırlayıcı kutunun DP olarak kabul edildiği anlamına gelmektedir. Bu nedenle, Ortalama Kesinlik değeri, BÜK eşik ayarına bağlıdır. Bu çalışmada, algılama sonuçlarını değerlendirmek için birleşme üzerinden kesişim eşik değeri olarak 0,5 belirlenmiştir. BÜK değeri 0,5'ten büyük veya ona eşitse, algılanan sınırlayıcı kutu DP olarak kabul edilmektedir. Denklem 3.8'de formülü gösterilmektedir.

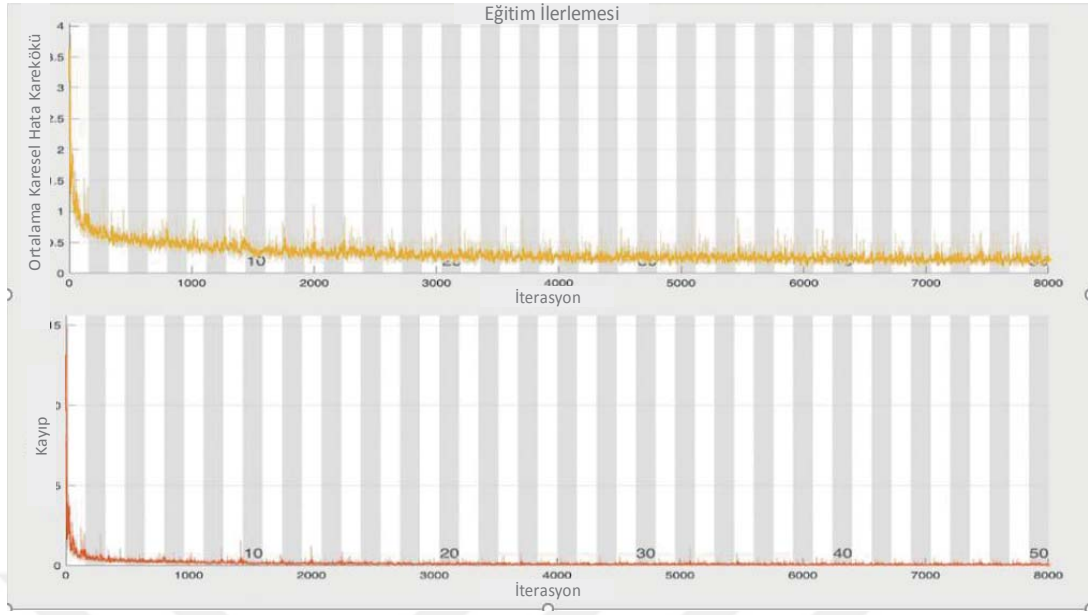
$$\text{Birleşme üzerinden kesişim} = \frac{\text{Kesişim Bölgesi}}{\text{Birleşim Bölgesi}} \quad (3.8)$$

Saniyedeki Çerçeve Sayısı (Frame Per Second-Fps)

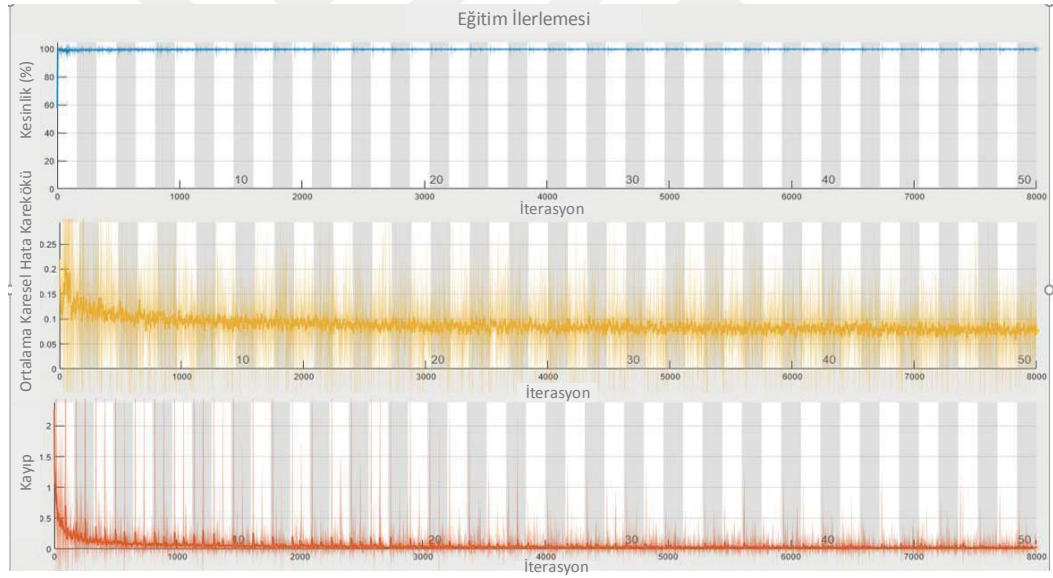
Saniyede algılanan çerçeve sayısını ifade etmektedir ve yalnızca algoritmanın yapısından değil, aynı zamanda deneysel ekipmanın donanım yapılandırmasından da etkilenmektedir.

3.5.4. Eğitim Süreci

Derin öğrenme için ağlar eğitilirken eğitim ilerlemesi grafiğini izleyerek ağın eğitim süreci hakkında ayrıntılı bilgi elde edilebilmektedir. Eğitim süreci sayesinde ağ doğruluğunun gelişim durumu, ne kadar hızlı geliştiği veya ağın eğitim verileri ile ilgili yaptığı kayıp oranı hakkında bilgi alınabilmektedir. Şekil 3.14'te en iyi sonuçları aldığımız Daha Hızlı B-ESA'nın ResNet101 omurga ağı ile eğitilmesi ve YOLO'nun MobileNet omurga ağı ile eğitilmesi aşamasında 8000 yinelemede eğitim süreci görülmektedir.



(a)



(b)

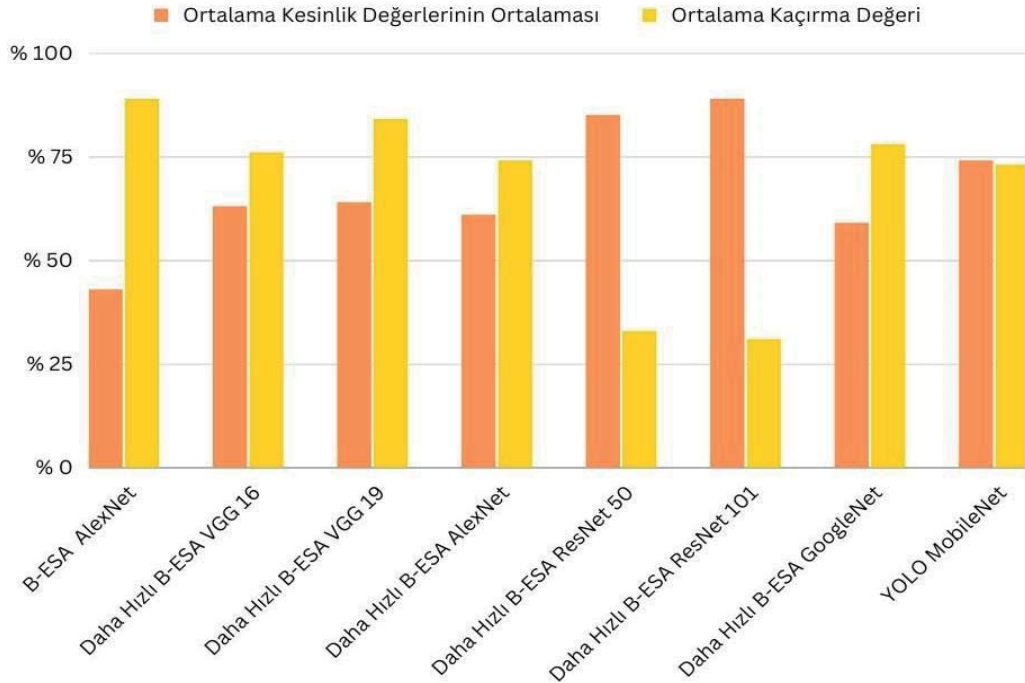
Şekil 3.14. Eğitim süreci grafiği (a) YOLO ile eğitilmiş MobileNet ağı, (b) Daha Hızlı B-ESA ile eğitilmiş Resnet 101 ağı

Çalışma kapsamında farklı nesne tespiti yaklaşımlarını karşılaştırmak için B-ESA, Daha Hızlı B-ESA ve YOLO derin öğrenme mimarilerinin farklı omurga ağı ile eğitilmesi sonucunda elde edilen sonuçlar karşılaştırılmıştır. Bu durumun algılama sonuçları üzerinde etkisi analiz edilmiştir. Tablo 3.5’te bu karşılaştırmalara ait tablo gösterilmektedir.

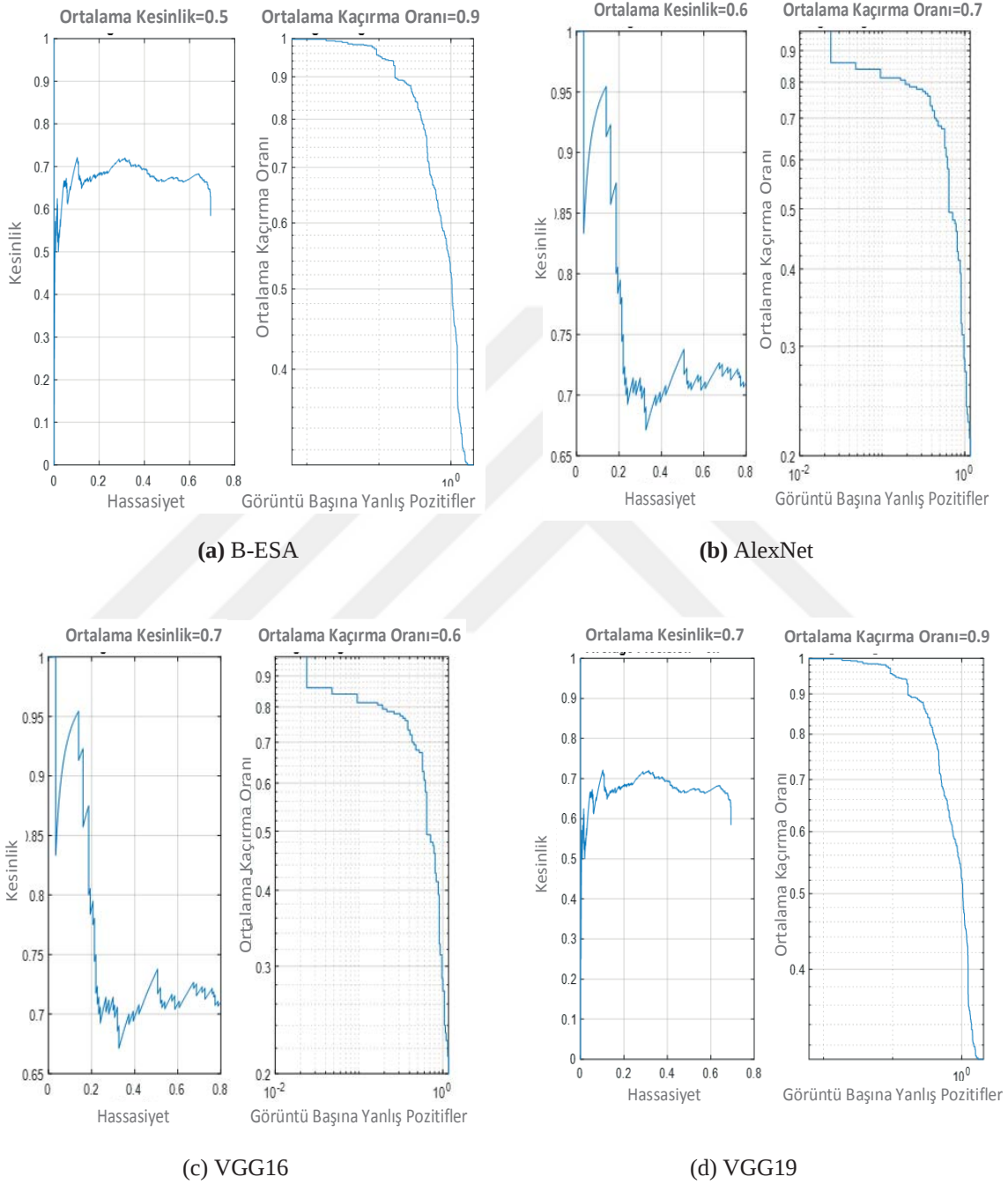
Tablo 3.5. Farklı omurga ağlarının eğitim sürecine etkisi

Derin Öğrenme	Omurga Ağı	Saniyedeki Çerçeve Sayısı (Fps)	Hız	Ortalama Kesinlik Değerlerinin Ortalaması (mAP) (%)	Ortalama Kaçırma Oranı (mMR) (%)
B-ESA	AlexNet	2.4	0.089	42.90	89.44
Daha Hızlı B-ESA	VGG 16	2.26	0.0153	62.04	76.47
Daha Hızlı B-ESA	VGG 19	3.08	0.1582	64.47	84.70
Daha Hızlı B-ESA	AlexNet	2.3	0.0749	61.22	73.56
Daha Hızlı B-ESA	ResNet 50	8.10	0.0023	85.16	32.80
Daha Hızlı B-ESA	ResNet 101	9.24	0.0018	89.20	31.10
Daha Hızlı B-ESA	GoogleNet	3.05	0.0473	59.02	77.87
YOLO	MobileNet	10.04	0.1644	74.20	73.82

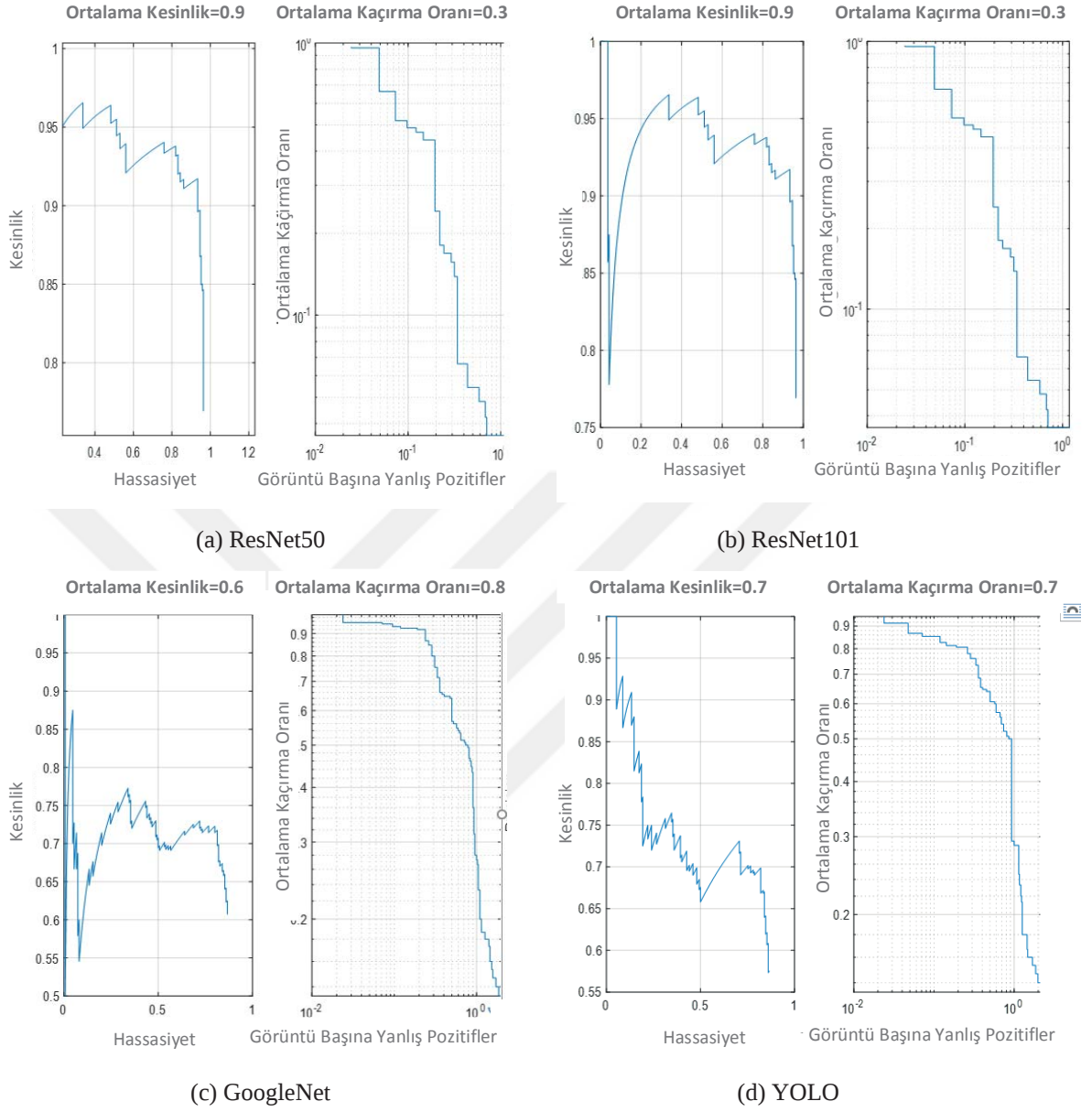
Kullanılan derin öğrenme mimarilerinin farklı omurga ağlarıyla eğitilmesi sonucu Daha Hızlı B-ESA mimarisi ResNet101 ile en yüksek başarıyı elde etmiştir. Şekil 3.15'te derin öğrenme mimarilerinin farklı omurga ağlarıyla eğitilmesi ile elde edilen performans sonuçlarının karşılaştırılması grafikte gösterilmiştir.

**Şekil 3.15.** Derin öğrenme performans sonuçlarının karşılaştırılması

Ortalama Kesinlik Değerlerinin Ortalaması (mAP) ve Ortalama Kaçırma Oranı (mMR) oranları eğitimin performansını göstermek için kullanılmaktadır. Ortalama kaçırma oranı ile hassasiyet ve kesinlik eğrilerinin gösterimi Şekil 3.16 ve Şekil 3.17'deki gibidir.



Şekil 3.16. Derin öğrenme modelleri için hassasiyet ve kesinlik eğrileri: (a) B-ESA, (b) AlexNet, (c) VGG16, (d) VGG19



Şekil 3.17. Derin öğrenme modelleri için hassasiyet ve geri çağırma eğrileri: (a) ResNet50, (b) Resnet101, (c) GoogleNet, (d) YOLO

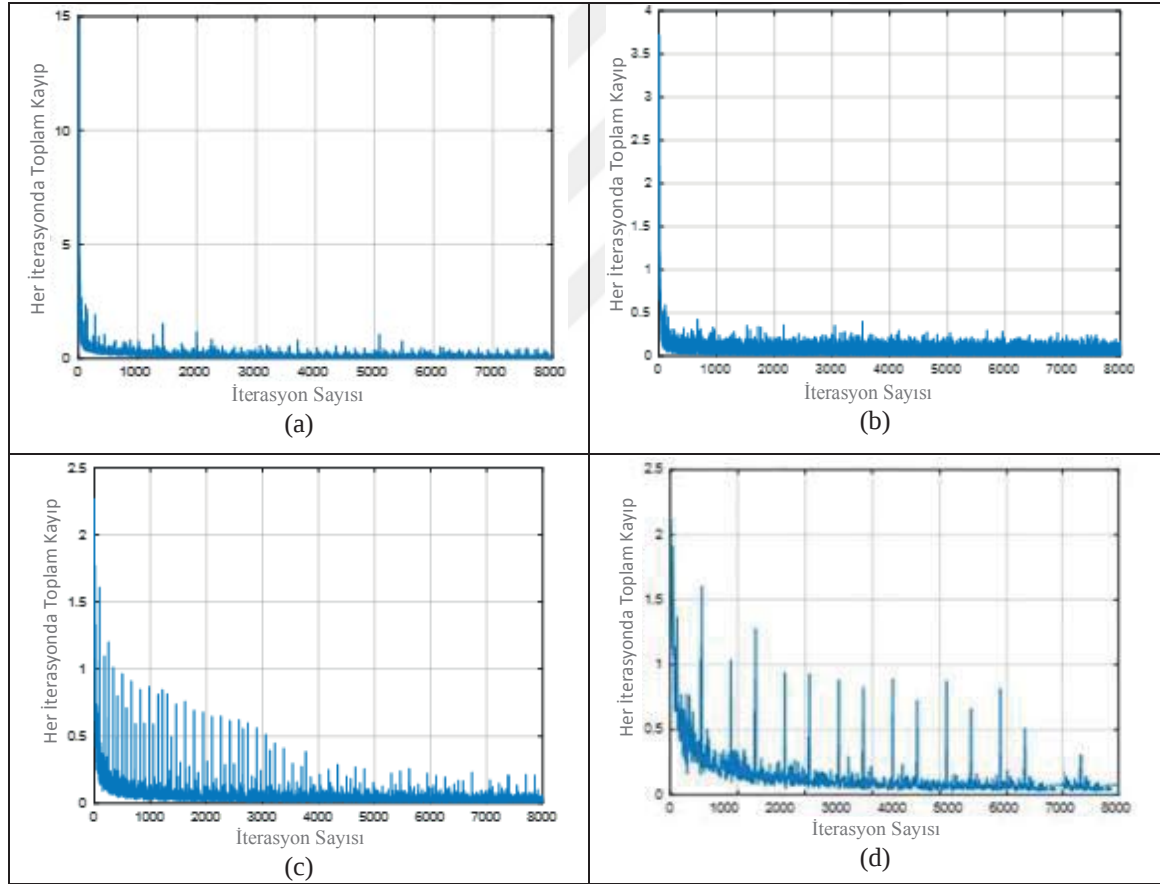
3.5.5. Eğitim sürecinde kayıp fonksiyonu

Derin öğrenmede çoklu görev kaybı bir kayıp fonksiyonu ile ifade edilmektedir. Bir görüntü için kayıp fonksiyonu şu şekilde tanımlanmaktadır:

$$L(\{p_i\}, \{t_i\}) = \frac{1}{N} \sum_i L_{cls}(p_i, p_i^*) + \alpha \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*) \quad (3.9)$$

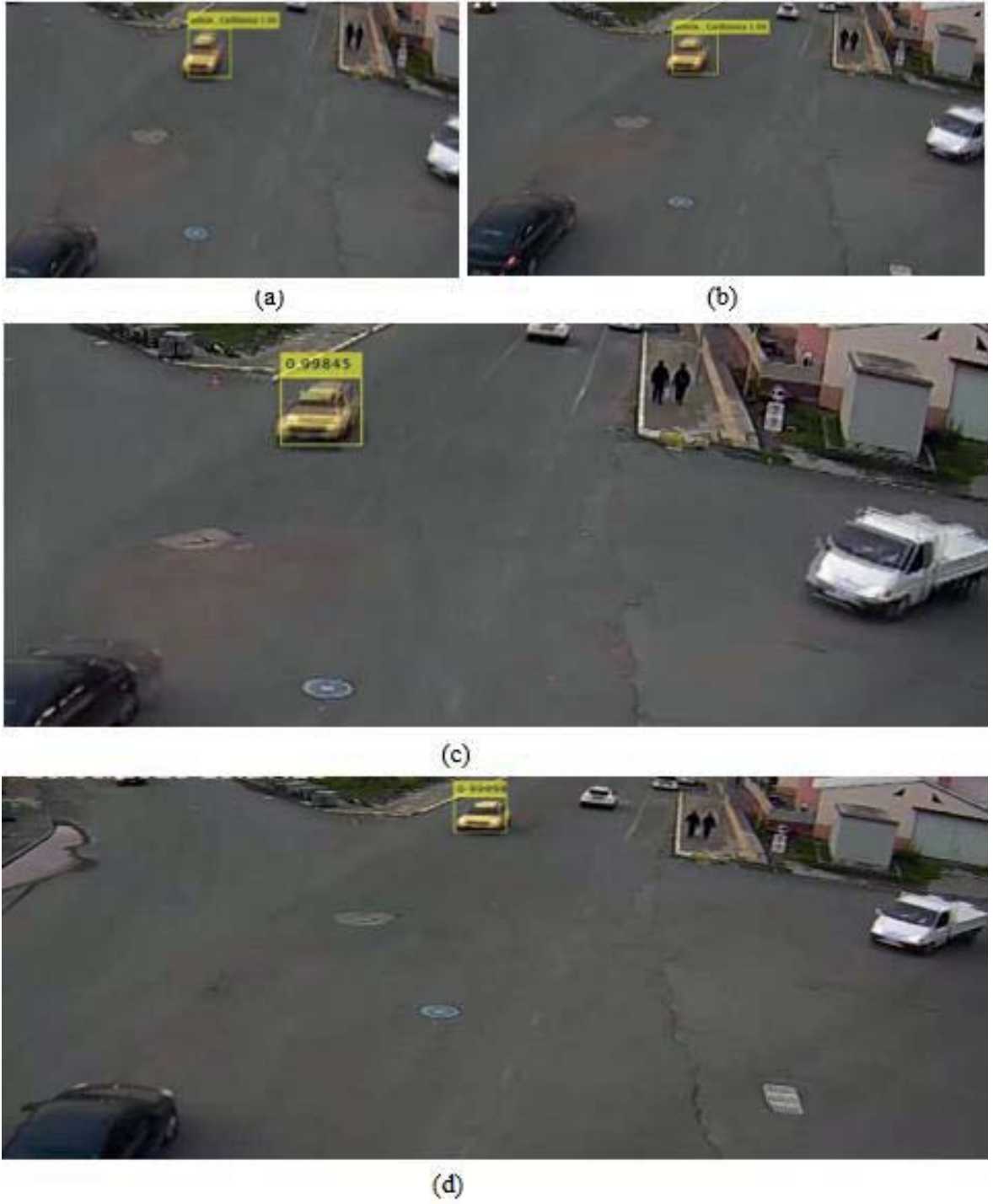
Burada, i , bir iterasyonda alınan veri kümesinde bulunan bir indekstir. p_i , örneğin pozitif olasılık değeridir, L_{cls} ise tüm örneklerin sınıflandırma kaybını ifade etmektedir. Kayıp fonksiyonu, sınıflandırma ve regresyon kaybından oluşmaktadır. Kayıp fonksiyonu, modelin tahmin edilen değeri ile eğitim örneği arasındaki farkı temsil etmektedir. Değer ne kadar küçükse, tahmin örneğinin gerçek örneğe o kadar yakın olduğu ve modelin sağlamlığı o kadar iyi kabul edilmektedir. [36].

Kayıp eğrileri incelendiğinde tüm eğitim sürecinin kayıp değeri sürekli olarak azalmaktadır ve tüm süreçte yukarı ve aşağı titreşimde olumsuz bir eğilim yoktur. Bu durum tüm model eğitiminde ağ parametrelerinin en uygun şekilde seçildiğini göstermektedir. Şekil 3.18’de gösterildiği gibi, eğrinin salınımının ve yakınsamasının ayrıntılarını daha kolay gözlemek için sistemin son 8.000 yinemesinde kayıp fonksiyonunun değişimi gösterilmektedir.



Şekil 3.18. Eğitim ağında kayıp eğrileri (a) AlexNet, (b) Resnet101, (c) Resnet50, (d) YOLO

Video gözetim sistemlerinden elde edilen ilk videolarda takip edilen araç ikinci videoda tespit edilmiştir. Araç tespiti amacıyla farklı derin öğrenme ağları eğitilerek sonuçları karşılaştırılmıştır. Araçların seçilen çerçevelerde %99 ve üzerinde tespit edildiği görülmektedir. Tespit edilen araçlara ait görüntüler Şekil 3.19’da gösterilmiştir.



Şekil 3.19. Araç tespiti ile tespit edilen araç görüntüleri (a) ResNet50, (b) ResNet101, (c) VGG19, (d) YOLO

Tablo 3.6’da nesne tespit çalışmalarından araç tespitine odaklanılmış çalışmalara ait literatür karşılaştırılması gösterilmektedir. Literatürdeki çalışmalar incelendiğinde kullanılan yöntemlerde hazır veri setleri üzerinde işlemlerin yürütüldüğü ve en yüksek başarımın daha çok Daha Hızlı B-ESA ile alındığı görülmektedir. Tez çalışması kapsamında oluşturulan veri seti üzerinde Daha Hızlı B-ESA’nın ResNet101 ile eğitilmesi sonucunda %89’luk başarı elde edilmiştir. Bu alanda yapılan

çalışmalar ile kıyaslandığında nesne tanıma alanında en etkin ve başarılı yöntemlerden biri olarak ön plana çıkmaktadır.

Tablo 3.6. Hedef araç tespit aşamasının benzer çalışmalarla kıyaslanması

Kaynak	Görüntü İşleme Yöntemi	Kullanılan Omurga Yapısı	Veri Seti	Ortalama Kesinlik Değerlerinin Ortalaması (mAP) (%)
[6]	Daha Hızlı B-ESA	Resnet101	UA-DETRAC	%68.25
[17]	Daha Hızlı B-ESA	MobileNet	KITTI	%24.5
[18]	Daha Hızlı B-ESA	ResNet50	MIT Traffic	%69.40
[19]	Daha Hızlı B-ESA	VGG19	KITTI	%95.14
[20]	Daha Hızlı B-ESA	VGG16 ve ResNet50	TT100K	%90
Önerilen yöntem	Daha Hızlı B-ESA	ResNet101	Oluşturulan Veri Seti	%89.20

Önerilen yöntemin özellikle kameralardan uzaklaşmakta olan küçük araçları tespit edecek kadar güçlü olduğu görülmüştür. Ayrıca bulanık, yağmurlu, gece vb. durumlar ile değişken ışık ve aydınlık özelliklerine sahip görüntülerde farklı yönlere giden araçları tespit edebilmektedir. Bununla birlikte kalabalık sahnelerde veya araçların yan yana olduğu durumlarda algılama sonuçları bazen yetersiz olmaktadır. Genel olarak, önerilen yöntem çeşitli boyutlardaki araçlarda oldukça yüksek bir performans göstermektedir. Bu açıdan gelecekteki çalışmalara yol gösterici nitelikte genişletilebileceği düşünülmektedir.

Nesne takibi amacıyla uzun yıllardır sürdürülen ayrıntılı çalışmalar sonucunda ortaya çıkan birden çok yöntem bulunmaktadır. Literatürde nesne takibi alanında yapılan çalışmalardan birçoğunda araç takibi konusunda çalışılmıştır. Hedef araç takibi son yıllarda araştırmacılar tarafından yoğun ilgi gösterilen bir alan olmuştur. Araç takibinde hala çözülmeyen birçok zorlayıcı durumun varlığı hedef araç takibi çalışmalarının dezavantajları olarak ortaya çıkmaktadır. Çalışmalarda odaklanılan bir ya da birden fazla sorun çözüme ulaştırılmış ise çalışmaların literatüre bu yöndeki katkıları avantaj olarak kabul edilmektedir. Nesne takibi amacıyla yürütülmüş çalışmalar arasından birden fazla kamerada araç takip işlemi yapılmış çalışmalar seçilerek incelenmiştir. Tablo 3.7’de nesne takibi amacıyla kullanılan çalışmalardan birden fazla kamerada araç takip işlemi yapılmış çalışmaların avantajlarına ve dezavantajlarına ait karşılaştırma tablosu gösterilmektedir.

Tablo 3.7. Nesne takibi alanında yürütülen çalışmaların avantaj ve dezavantajlarının karşılaştırılması

Kaynak	Yöntem	Avantaj	Dezavantajları
[130]	3B ve renk bilgisi, kenar haritası, nitelik tabanlı model	Araç doğru ve verimli bir şekilde tespit edilmektedir. Hedef aracın birden çok kamerada otomatik izlenmesi ile insan gücünden tasarruf sağlanmıştır. Ayrıca hesaplama yükü azaltılmıştır.	Farklı araçlar elde edilen farklı kamera görünümünde birbirine benzeyebilmekte ve bu nedenle eğitim için fazla veriye ihtiyaç duyulmaktadır.
[131]	Tek Atış Dedektörü	Yerel özelliklere dayalı yöntemlerin kullanımı ile benzersiz görsel ipuçlarının yakalayabilmesi sağlanmıştır. Farklı araçlar arasında ayırım yapılması sağlanmış ve yüksek doğruluk elde edilmiştir.	Yerel özelliklere dayalı yöntemlerde, özelliklerin çıkarılması aşaması hesaplama yükünü önemli ölçüde arttırmaktadır.
[132]	ESA tabanlı özellik çıkarıcı	Tek kameralı ve çok kameralı araç takibi ayrı ayrı yapılmıştır.	Küçük hedeflerde ve birden fazla aracın yakın olduğu durumlarda hedef kaybedilmiştir.
[133]	ESA + GoogleNet	Aydınlanma ve ışık değişimlerine dayanıklı bir yöntem sunulmuştur. Özellik olarak renk ve doku bilgileri birleştirilerek kullanılmıştır.	Görünüşe dayalı özellik çıkarımı kullanılarak hedef araç takibi yapıldığı için araç görünüm değişikliklerinden etkilenmektedir.
[134]	ESA+VGG 16 ResNet50	Uzamsal, zamansal bilgiler dikkate alınmış ve yüksek başarımlar elde edilmiştir. VGG16 ile %61.50, ResNet50 ile %88.92 başarımlar elde edilmiştir.	Hazır veri seti kullanılmış, gerçek dünya problemlerine odaklanılmamış ve gerçek trafik senaryoları üzerinde çalışılmamıştır.

İncelenen hedef aracın takibine yönelik çalışmalarda, plaka tanıma kullanıldığı, ayrıca araç renk ve özelliklerine odaklanarak takip gerçekleştirme yapıldığı görülmektedir. Çalışmalarda, ışık, netlik ve hız değişimi gibi problemlere kesin çözüm getirilemediği gözlemlenmiştir. Tez çalışması kapsamında da hedef tespiti ve takibi yapılırken birçok zorlukla karşılaşmıştır. Çalışmanın avantaj ve dezavantajları şu şekildedir.

Avantajları

Hedef aracın takibi aşamasında önerilen takip yöntemi ile aracın yakın, uzak ya da hızlı olması gibi durumlarda takip işlemi sürdürülebilmektedir. Kullanılan takip yönteminde aracın renk ve doku gibi özelliklerine odaklanılmadığından bu değişimlerden takip süreci çok etkilenmemektedir.

Hedef aracın takibini zorlaştıran etkenlerden, aracın bir engelin arkasında kalması, araçların birbirine çok yakın olmaları, benzer araçların bulunması durumlarında da önerilen takip sistemi ile takip işlemine devam edilmektedir.

Tez çalışması kapsamında hedef takip edilirken her video çerçevesi resim olarak kaydedilmekte ve hedef aracın sınırlayıcı kutu bilgileri ile resim konumları kullanılarak etiketli eğitim verisi oluşturulmaktadır. Derin öğrenme mimarilerinde büyük miktarlarda verinin etiketlenmesi oldukça zaman alıcı bir işlemdir. Bu sayede otomatik bir etiketleme işlemi gerçekleştirilmekte ve etiketleme için gerekli zaman azaltılmaktadır.

Tez çalışması kapsamında birden fazla omurga ağı farklı derin öğrenme mimarileri ile eğitilmiştir. Literatürdeki çalışmalara oranla daha fazla sayıda mimarinin ve omurga ağının sonuçlara etkisi karşılaştırılmıştır.

Dezavantajları

Trafik gözetleme sistemlerinde farklı kameraların çözünürlüğünde değişebilmektedir. Kameranın eski ya da yeni olması, hareketli ya da sabit olması alınan görüntülerde büyük değişimlere sebep olmaktadır. Görüntülerin farklı netlikte oluşu ve aydınlanma değişimleri aracın tanınmasını büyük ölçüde zorlaştırmaktadır.

Trafikte birbirine benzer araçların fazla sayıda olması ve yakın mesafede ilerlemeleri hedef aracın farklı kameralarda tespit edilmesini zorlaştırmaktadır.

Trafik gözetleme sistemlerindeki kameraların konumlarının, yüksekliklerinin ve açılarının farklı olması nedeniyle, aynı aracın farklı kamera görüntülerinde farklı perspektiflere sahip olmasına neden olmaktadır.

Trafik gözetleme sistemlerindeki kameraların gün içinde farklı aydınlatma koşullarından etkilenmesinden dolayı, aynı aracın farklı kameralar tarafından elde edilen renk özellikleri büyük farklılıklar göstermektedir. Ayrıca gece çekilen araç görüntüleri, sokak lambaları ve diğer ışıklar nedeniyle gündüz çekilenlerden oldukça farklıdır.

Hedef aracın ve yol renginin birbirine yakın olmasında özellikle hedef aracın farklı kameralarda tespitini zorlaştırmaktadır.

4. SONUÇLAR

Gelişen teknoloji ile birlikte video gözetim sistemleri olarak adlandırılan kamera kayıt sistemlerinin kullanımı yaygınlaşmış ve en yoğun olarak güvenlik alanında kullanılmaya başlamıştır. Bu sistemlerin izlenmesi ve denetlenmesi güvenlik uygulamalarında oldukça önemlidir. Elle ve yarı otomatik denetimin yaygın olduğu bu sistemlerde, insan denetiminden bağımsız bir denetim hem zamandan hem insan gücünden tasarruf anlamına gelmektedir. Özellikle video gözetim sistemleri üzerinde hedef takibi işlemi, şüpheli nesne kontrolü, olay denetimi, anomali tespiti gibi durumlarda oldukça önemlidir. Ancak artan kamera sayısı ve uzun süreli video kayıtlarının varlığı bu sistemler üzerinde işlem yapmayı zorlu bir süreç haline dönüştürmektedir.

Bu tez çalışmasında bir hedefin birden fazla kamerada takibinin ve tespitinin yapılması amaçlanmıştır. Bu sayede insan denetimi olmadan kameralarda hedef belirlenip takibi bir güzergâh boyunca devam etmektedir. Ayrıca kameralarda kayıtların tamamında değil de sadece görüntüleneceği belirlenen sürelerde aranması sağlanmıştır. Belirli bir bölgede takip edilecek bir hedef aracın geçtiği belirlenen kameralar, geliştirilen algoritma ile saptanmış ve belirlenen kameralarda hedef aracın tespit edilip takip edilme işlemi gerçekleştirilmiştir. Takip işleminde Gauss karışım modelini temel alan arka plan çıkarım yöntemi kullanılmıştır. Önerilen takip yönteminde kullanılan algoritma ile hedefin engellerin arkasında kalmasına veya yan yana geçen araçlara rağmen kaybedilmeden başarılı şekilde takibinin gerçekleştirildiği belirlenmiştir.

Çalışma kapsamında video gözetim sistemlerinde otomatik hedef tanımaya yönelik bir yaklaşım sunulmuştur. Seçilen hedef araç birden fazla video görüntüsünde tespit edilmiştir. Hedefin tespit edilmesi için MOBESE olarak adlandırılan trafik gözetim sistemlerinde kullanılan farklı kameralardan alınan video görüntüleri kullanılmıştır. Hedefin ilk tespit edildiği videoda, kullanılacak derin öğrenme modelinde eğitim verisi oluşturabilmek amacıyla video etiketleme işlemi gerçekleştirilmektedir. Video etiketleme işlemini otomatik ve daha etikili gerçekleştirebilmek için geliştirilmiş araç takip yöntemi kullanılmıştır. Bu yöntem video etiketleme amacıyla kullanılan etiketleme uygulamalarına da uygun yeni bir algoritma olarak kullanılabilir. Elde edilen diğer video görüntülerinde derin öğrenme yöntemleri ile eğitim yapılarak hedef aracın yüksek oranda tespiti sağlanmıştır.

Çalışmanın ilerleyen aşamasında derin öğrenme modelleri üzerinde farklı omurga ağları ile gerçekleştirilen eğitim sonucunda elde edilen ortalama kesinlik değerlerinin ortalaması (mAP) değerleri karşılaştırılmıştır. Eğitimler sonucunda tespit edilen mAP ölçümleri karşılaştırıldığında en iyi sonuç, Daha Hızlı B-ESA'nın ResNet101 omurga ağı üzerinde eğitilmesi ile bulunmuştur. Yakın değerlerin elde edildiği ResNet50 omurga yapısıyla da ikinci en yüksek başarımla elde edilmiştir. YOLO küçük nesneleri tanımda yetersiz kaldığı ve ışık değişimlerine karşı hassas olduğu için uzaklaşan araç görüntülerinde ve ışık değişiminin fazla olduğu MOBESE

görüntülerinde daha yetersiz kalmıştır. Çalışma kapsamında takip edilen hedef araçların video kayıtları süresince uzaklaşması, küçülmesi ve ışık değişimlerinin fazla olması özellikle küçük hedeflerin tanınmasında YOLO'nun zayıf kalması yönündeki literatür çalışmalarını desteklemiştir.

Derin öğrenme modelleri ve kullanılan omurga yapısına göre elde edilen mAP değerlerinin büyükten küçüğe doğru sıralanması şu şekildedir: Daha Hızlı B-ESA ResNet101> Daha Hızlı B-ESA ResNet50> YOLO > Daha Hızlı B-ESA VGG19> Daha Hızlı B-ESA VGG16> Daha Hızlı B-ESA AlexNet> Daha Hızlı B-ESA GoogleNet >B-ESA AlexNet şeklindedir.

Çalışmanın geliştirilerek bir şehrin tüm kameraları ya da birden fazla şehir de bulunan kameralar dahil edilerek daha geniş bir bölgede hedef takibi yapılması daha etkili olacaktır. İlerleyen dönemde çalışmaya sadece güvenlik kameraları değil konut ve iş yeri kameraları da eklenerek kapsamı genişletilebilir. Çalışmada hedef araç takibi üzerinde durulmuştur. Ancak kapsamı genişletilerek hedefin farklılaştığı, örneğin şüpheli araçtan inen şüpheli insanın takibinin devam etmesi gerektiği durumlarda da takibinin gerçekleştirilmesi sağlanabilir. Bu sayede bu alanda yapılacak yeni çalışmalara da yol göstereceği olacağı düşünülmektedir.

KAYNAKLAR

- [1] Yılmaz, M. (2008). *Birden fazla kamera ile çoklu hedef takibi*, Yüksek Lisans Tezi, Orta Doğu Teknik Üniversitesi Fen Bilimleri Enstitüsü , 133s
- [2] Talu, F. (2010). *İnsan hareketlerinin takibinde karşılaşılan problemlerin çözümüne yeni yaklaşımlar*, Doktora Tezi, Fırat Üniversitesi Fen Bilimleri Enstitüsü, 104s
- [3] Al-Karablieh, A., Alsmadi, I., Gharaibeh, K. (2018). A review of visual surveillance systems, *International Journal of Advanced Computer Science and Applications*, 9(10), pp. 126-133
- [4] Chen, L., ve Wang, Z. (2021). An Overview of Visual Surveillance Technology, *International Journal of Advanced Intelligence Paradigms*, 14(1/2), pp. 68-88
- [5] Jeon, H. J. , Pham, Nguyen, V. D., Jeon, J. W. (2018). High-Speed Car Detection Using ResNet-Based Recurrent Rolling Convolution, *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 286-291, doi: 10.1109/SMC.2018.00059
- [6] Luo, J., Fang, H., Shao, F., Zhong, Y., Hua, X. (2021). Multi-Scale Traffic Vehicle Detection Based On Faster R-CNN With NAS Optimization And Feature Enrichment, *Defence Technology*, vol. 17, no. 4, pp. 1542–1554, doi: 10.1016/J.DT.2020.10.006
- [7] Fernández, J., Cañas, J. M., Fernández, V., Paniego, S. (2021). Robust Real-Time Traffic Surveillance with Deep Learning, *Comput. Intell. Neurosci.*, vol. 2021, doi: 10.1155/2021/4632353
- [8] Maity, M., Banerjee, S., Chaudhuri, S. S. (2021). Faster RCNN and YOLO based Vehicle detection: A Survey, *Proc. - 5th Int. Conf. Comput. Methodol. Commun. ICCMC 2021*, pp. 1442–1447, doi: 10.1109/ICCMC51019.2021.9418274
- [9] Wang, F., Li, Y., Wei, Y., Dong, H. (2020). Improved Faster RCNN for Traffic Sign Detection, *2020 IEEE 23rd Int. Conf. Intell. Transp. Syst. ITSC 2020*, doi: 10.1109/ITSC45102.2020.9294270
- [10] Türker, B. (2010). *Çoklu hipotez yöntemi ile çoklu görsel hedef takibi*, Yüksek Lisans Tezi , Orta Doğu Teknik Üniversitesi Fen Bilimleri Enstitüsü , 119s
- [11] Böcekçi, G. (2011). *Video işleme yöntemi ile hassas yer değişimi tespiti*, Doktora Tezi, Marmara Üniversitesi, Fen Bilimleri Enstitüsü, 112 s
- [12] Kalman R. E. (1960). A new approach to linear filtering and prediction problems, *Transaction of the ASME- Journal of Basic Engineering*, 82, pp. 35-45
- [13] TRT Haber (2022, 11 Aralık). Erişim adresi <https://www.trthaber.com/haber/gundem/1200-kamera-kaydi-incelendi-saldirgan-10-saatte-yakalandi-723398.html>
- [14] Kiratiratanapruk, K., Dubey, P., Siddhichai, S.(2005). A gradient-based foreground detection technique for object tracking in a traffic monitoring system, *IEEE Int. Conf. Adv. Video Signal Based Surveill. - Proc. AVSS 2005*, vol. 2005, pp. 377–381, doi: 10.1109/AVSS.2005.1577298
- [15] Liu, Y. , Lu, Y., Shi, Q. , Ding, J. (2013). Optical flow based urban road vehicle tracking, *Proc. - 9th Int. Conf. Comput. Intell. Secur. CIS 2013*, pp. 391–395, doi: 10.1109/CIS.2013.89
- [16] Sushmitha, S., Satheesh, N., Kanchana, V. (2020). Multiple car detection, recognition and tracking in traffic, *2020 Int. Conf. Emerg. Technol. INCET 2020*, doi: 10.1109/INCET49848.2020.9154107
- [17] Nguyen, H. (2019). Improving Faster R-CNN Framework for Fast Vehicle Detection, *Computational Intelligence in Image Processing 2020*, <https://doi.org/10.1155/2019/3808064>
- [18] Arinaldi, A., Pradana, J. A., Gurusanga, A. A. (2018). Detection and classification of vehicles for traffic video analytics, *Procedia Comput. Sci.*, vol. 144, pp. 259–268, doi: 10.1016/J.PROCS.2018.10.527

- [19] Fan, Q., Brown, L., Smith, J. (2016). A closer look at Faster RCNN for vehicle detection, *IEEE Intell. Veh. Symp. Proc.*, vol. 2016-August, pp. 124–129, doi: 10.1109/IVS.2016.7535375
- [20] Cao, C., Wang, B., Zhang, W., Zeng, X., Yan, X., Feng, Z., Liu, Y., Wu, Z., (2019). An Improved Faster RCNN for Small Object Detection, *IEEE Access*, vol. 7, pp. 106838–106846, doi: 10.1109/ACCESS.2019.2932731
- [21] Arcos-García, Á., Álvarez-García, J. A., Soria-Morillo, L. M. (2018). Evaluation of deep neural networks for traffic sign detection systems, *Neurocomputing*, vol. 316, pp. 332–344, doi: 10.1016/J.NEUCOM.2018.08.009
- [22] Shaikh, S. H., Saeed, K., Chaki, N. (2014). Moving Object Detection Using Background Subtraction, *SpringerBriefs in Computer Science*, Springer, Cham, 1, pp. 15–23, doi: 10.1007/978-3-319-07386-6_6
- [23] Elgammal, A. (2015). Background Subtraction: Theory and Practice, Synthesis Lectures on Computer Vision, California, doi: 10.2200/S00613ED1V01Y201411COV006
- [24] Paragios, N., Deriche, R. (2000). Geodesic active contours and level sets for the detection and tracking of moving objects, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22 (3), pp. 266–280
- [25] Fejes, S., Davis, L.S. (1999). Detection of independent motion using directional motion estimation, *Computer Vision and Image Understanding (CVIU)*, Vol. 74 (2), pp. 101–120.
- [26] Piccardi, M. (2004). Background subtraction techniques: A review, *IEEE Int. Conf on Systems, Man and Cybernetics (SMC)*, Vol. 4, pp. 3199–3104
- [27] Shaikh, S.H., Bhunia, S.K., Chaki, N. (2012). On Generation of Silhouette of Moving Objects from Video, *Springer Proceedings of the 4th International Conference on Signal and Image Processing (ICSIP)*, Vol. 1, pp. 213–223
- [28] Chaqueta, J. M., Carmonaa, E.J., Caballerob, A.F. (2013). A survey of video datasets for human action and activity recognition, *Computer Vision and Image Understanding*, Vol. 117 (6), pp. 633–659
- [29] Zhang, L., Li, S.Z., Yuan, X., Xiang, S. (2007). Real-time Object Classification in Video Surveillance Based on Appearance Learning, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8
- [30] Peterson, M. A., Gibson, B. S. (1991). *The initial identification of figure-ground relationships: Contributions from shape recognition processes*, *Bulletin of the Psychonomic Society*, Vol. 29(3), pp. 199–202, doi: 10.3758/BF03342677
- [31] Johansson, G. (1973). *Visual perception of biological motion and a model for its analysis. Perception & psychophysics*, Vol. 14(2), pp. 201–211, doi: 10.3758/BF03212378. 5
- [32] Dalal, N., Triggs, B. (2005). Histograms of oriented gradients for human detection, *In Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, doi: 10.1109/CVPR.2005.177. 5, 10
- [33] Gavrila, D. M. (2000). Pedestrian detection from a moving vehicle, *In Proc. of the 6th European Conference of Computer Vision*, pp. 37–49. Springer, doi: 10.1007/3-540-45053-X_3. 5
- [34] Oren, M. (1997). Constantine Papageorgiou, Pawan Sinha, Edgar Osuna, and Tomaso Poggio. Pedestrian detection using wavelet templates. *In Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 193–199, doi: 10.1109/CVPR.1997.609319. 5

- [35] Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y. (2013). *Overfeat: Integrated recognition, localization and detection using convolutional networks*. *arXiv preprint* 1312.6229, pp. 5, 7
- [36] Felzenszwalb, P. F., Huttenlocher, D. P. (2005) Pictorial structures for object recognition, *International Journal of Computer Vision*, Vol. 61(1), pp. 55–79, doi: 10.1023/B:VISI.0000042934.15159.49. 6
- [37] Felzenszwalb, P. F, Girshick, R. B., McAllester, D., Ramanan. D. (2010). Object detection with discriminatively trained part-based models, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.32(9), pp.1627–1645, doi: 10.1109/TPAMI.2009.167
- [38] Dean, T., Ruzon, M. A., Segal, M., Shlens, J., Vijayanarasimhan, S., Yagnik, J. (2013) Fast, accurate detection of 100,000 object classes on a single machine. In *Proc. of the IEEE Computer Society Conference on Conference on Computer Vision and Pattern Recognition*, doi: 10.1109/CVPR.2013.237
- [39] Streit, R., Angle, R. B., Efe, M. (2021). *Analytic Combinatorics for Multiple Object Tracking*, Springer Cham, Switzerland, 1, pp. 1-221, doi: <https://doi.org/10.1007/978-3-030-61191-0>
- [40] Ge, W., Guo, Z., Dong, Y., Chen, Y. (2016). Dynamic background estimation and complementary learning for pixel-wise foreground/background segmentation, *Pattern Recognition*, 59, pp. 112-125
- [41] Karasulu, B. ve Korukoglu, S. (2012). Moving object detection and tracking by using annealed background subtraction method in videos: Performance optimization, *Expert Systems with Applications*, 39(1), pp. 33-43, ISSN 0957-4174, doi: <https://doi.org/10.1016/j.eswa.2011.06.040>
- [42] Karasulu, B. (2010). Review and evaluation of well-known methods for moving object detection and tracking in videos. *Journal of Aeronautics and Space Technologies*, i4, pp. 11-22
- [43] Nascimento, J. C., Marques, J. S. (2006). Performance evaluation of object detection algorithms for video surveillance, *IEEE Transactions on Multimedia*, pp.761-774
- [44] Benezeth, Y., Jodoin, P. M., Emile, B., Laurent, H., & Rosenberger, C. (2008). Review and evaluation of commonly-implemented background subtraction algorithms. In *Proceedings of 19th international conference on pattern recognition (ICPR 2008)* . pp. 1-4
- [45] Wu, M, Peng, X. (2010). Spatio-temporal context for codebook-based dynamic background subtraction. *AEU Int J Electron Commun* 64(8):739–747. doi:10.1016/ j.aeue. 2009.05.004
- [46] Stauffer C, Grimson, W. (1999) Adaptive background mixture models for real-time tracking. In: *Proceedings of IEEE computer society conference on computer vision and pattern recognition (CVPR)*, pp.246–252
- [47] Zivkovic, Z., Van der Heijden, F. (2006) Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recognit Lett*, 27(7), pp.773–780
- [48] Karasulu, B. ve Korukoglu, S. (2013). *Performance Evaluation Software Moving Object Detection and Tracking in Videos*, SpringerBriefs in Computer Science, pp. 1-75
- [49] Karasulu, B, Korukoglu, S. (2011). A software for performance evaluation and comparison of people detection and tracking methods in video processing. *Multimedia Tool Appl*, 55(3), pp.677–723. doi:10.1007/s11042-010-0591-2
- [50] Carmona, E.J, Martínez-Cantos J, Mira J. (2008). A new video segmentation method of moving objects based on blob-level knowledge. *Pattern Recognit Lett* 29(3):272–285. doi: 10.1016/j.patrec.2007.10.007

- [51] Cheung, SC, Kamath, C. (2004). Robust techniques for background subtraction in urban traffic video. Video communications and image processing, SPIE electronic imaging, San Jose, January, UCRL-JC-153846-ABS, UCRL-CONF-200706
- [52] Wang, H., Suter, D. (2006). Background subtraction based on a robust consensus method, International Conference on Pattern Recognition, IEEE, pp. 223–226
- [53] Wang, H., Suter, D. (2007). A Consensus-based Method for Tracking: Modelling Background Scenario and Foreground Appearance. Elsevier Science Inc
- [54] Yadav, D.K. , Sharma, L., Bharti, S.K. (2015). Moving object detection in real-time visual surveillance using background subtraction technique, International Conference on Hybrid Intelligent Systems, IEEE, pp. 79–84
- [55] Li, Y., Liu, L., Song, J., Zhang, Z., Chen, X. (2018). Combination of local binary pattern operator with sample consensus model for moving objects detection, *Infrared Physics & Technology*, 92, pp. 44-52
- [56] Droogenbroeck, M. V., Paquot, O. (2012). Background subtraction: experiments improvements for ViBe, in Proc. IEEE Workshops on Change Detection (CDW'12) at CVPR'12, pp. 32-37
- [57] Kim, K., Chalidabhongse, T. H., Harwood, D., Davis, L. (2005). Real-time foreground-background segmentation using codebook model,” *Real-Time Imaging*, 11(3), pp. 172-185
- [58] Guo, J. M., Liu, Y. F., Hsia, C.H., Shih, M. H., Hsu, C.S. (2011). Hierarchical Method for Foreground Detection Using Codebook Model,” *IEEE Transactions on Circuits and Systems for Video Technology*, 21(6), pp. 804-815
- [59] Liu, C., Yuen, P. C., Qiu, G. (2009). Object motion detection using information theoretic spatio-temporal saliency,” *Pattern Recognition*, 42(11), pp. 2897-2906
- [60] Mahadevan , V., Nuno, V. (2010). Spatiotemporal Saliency in Dynamic Scenes,” *IEEE Trans. Pattern Anal. Mach. Intell*, 32(1), pp. 171-177
- [61] Liu, X., Lin, L., Yan, S. (2011). Integrating spatio-temporal context with multiview representation for object recognition in visual surveillance,” *IEEE Transactions on Circuits and Systems for Video Technology*, 21(4), pp. 393-407
- [62] Lin, L., Lu, Y., Pan, Y. (2012). Integrating graph partitioning matching for trajectory analysis in video surveillance,” *IEEE Transactions on Image Processing*, 21(12), pp. 4844-4857
- [63] Alimisis, V., Gennis, G., Touloupas, K., Dimas, C., Gourdouparis, M., Sotiriadis, P. P. (2022). Gaussian Mixture Model classifier analog integrated low-power implementation with applications in fault management detection, *Microelectronics Journal*, 126, 105510
- [64] Gu, C., Lu, X., Zhang, C. (2022). Example-based color transfer with Gaussian mixture modeling, *Pattern Recognition*, 129, 108716
- [65] McLachlan G., Peel D. Finite Mixture Models, Wiley Series in Probability and Statistics, John Wiley & Sons, Inc., 299, pp.1-456, doi: <http://dx.doi.org/10.1002/0471721182>
- [66] Murphy, K. P. (2012). Machine learning: a probabilistic perspective. The MIT Press, pp.1104
- [67] Bishop, C. M. (2006). Pattern recognition and machine learning, Information Science and Statistics, 1, pp. 1- 729
- [68] Dokladal, P., Enficiaud, R., Dejnozkova, E. (2004). Contour-based object tracking with gradient-based contour attraction field, *EEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '04)* , pp. 17-20

- [69] Chen, Q., Sun, Q. S., Heng, P. A., Xia, D. S. (2010). Two-Stage Object Tracking Method Based on Kernel and Active Contour, *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 605-609
- [70] Zhou, F., Pu, B., Xiangzhi B. (2011). Particle Filter Based on Color Feature with Contour Information Adaptively Integrated for Object Tracking, *Fourth International Symposium on Computational Intelligence and Design*, pp.359-362
- [71] Hu, W., Zhou, X., Li, W., Luo, W., Zhang, X., Maybank, S. (2013). Active Contour -Based Visual Tracking by Integrating Colors, Shapes, and Motions, *IEEE Transactions on Image Processing*, 22 (5), pp. 1778 – 1792
- [72] Ramanan, D., Forsyth, D., Zisserman, A. (2005). Strike a Pose: Tracking People by Finding Stylized Poses, *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1, pp. 271-278
- [73] Zhao, T., Nevatia, R., Wu, B. (2008). Segmentation and Tracking of Multiple Humans in Crowded Environments, in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(7), pp. 1198-1211, doi: 10.1109/TPAMI.2007.70770
- [74] Chen, T. (2009). Object Tracking Based on Active Contour Model by Neural Fuzzy Network, *IITA International Conference on Control Automation and Systems Engineering*, pp. 570-574
- [75] Wu, X., Mao, X., Chen, L., Compare, A. (2013). Combined Motion and Region-Based 3D Tracking in Active Depth Image Sequence ' Green Computing and Communications (GreenCom), *2013 IEEE and Internet of Things (iThings/CPSCoM), IEEE International Conference on and IEEE Cyber, Physical and Social Computing*, pp. 1734 – 1739
- [76] Cavallaro, A., Steiger, O. ve Ebrahimi, T., (2005), "Tracking video objects in cluttered background", *Circuits and Systems for Video Technology, IEEE Transactions on*, 15(4), pp.575– 584
- [77] Yılmaz, A., Javed, O., Shah, M. (2006). Object tracking: a survey, *ACM Surveys*, 38, 13.
- [78] Akbulut, O., 2014, Video için bölgesel ortak değişim betimleyici tabanlı iyileştirilmiş hedef takibi, Doktora Tezi, Kocaeli Üniversitesi Fen Bilimleri Enstitüsü, 117s
- [79] Bakhtari, A., Benhabib, B. (2007). An active vision system for multitarget surveillance in dynamic environments, *IEEE Transactions on Systems, Man, and Cybernetics*, 37(1), pp.190-198
- [80] Huang, C. M., Fu L. C. (2011). Multitarget visual tracking based effective surveillance with cooperation of multiple active cameras, *IEEE Transactions on Systems, Man, and Cybernetics*, 41(1), pp.234-247
- [81] Song, B., Chowdhury, K. R.(2008) Robust tracking in a camera network: A multi-objective optimization framework, *IEEE Journal of Selected Topics in Signal Processing*, 2(4), pp.582-596
- [82] Wang, H., Liu, C., Zhang, X., Li, Q. (2009). Multi-camera tracking based on information fusion in video surveillance, *2nd International Congress on Image and Signal Processing*, Tianjin, China
- [83] Sönmez, Y., Kutlu, H., Avcı, E. (2016). BLOB Analizi ile Beyaz Kan Hücre Morfolojisi Tespiti, *Tıp Teknolojileri Kongresi*
- [84] Adıgüzel, D. (2018). *Deniz Gözetlemesi İçin Hareketli Hedef Takip Sistemi*, Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü
- [85] Salvi, G. (2012). An automated vehicle counting system based on blob analysis for traffic surveillance. *Proceedings of 2012 International Conference on Image Processing, Computer Vision, and Pattern Recognition*, İtalya
- [86] BLOB Analizi (Video ve Görüntü İşleme), (2022, 10 Aralık). Erişim adresi <http://what-when-how.com/introduction-to-video-and-image-processing/blob-analysis-introduction-to-video-and-image-processing-part-1/>

- [87] Diwan, T., Anirudh, G., Tembhurne, J.V. (2023). Object detection using YOLO: challenges, architectural successors, datasets and applications. *Multimed Tools Appl* 82, pp.9243–9275. <https://doi.org/10.1007/s11042-022-13644-y>
- [88] Tan, F.G., Yüksel, A.S., Aydemir, E. & Ersoy, M. (2021). Derin Öğrenme Teknikleri ile Nesne Tespiti ve Takibi Üzerine Bir İnceleme. *Avrupa Bilim ve Teknoloji Dergisi*, 25, ss.159-171
- [89] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097-1105
- [90] Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. In *European conference on computer vision*, pp. 818-833, Springer, Cham.
- [91] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*
- [92] Srivastava, S., Narayan, S, Mittal, S.(2021).A survey of deep learning techniques for vehicle detection from UAV images, *Journal of Systems Architecture*, 117, doi: <https://doi.org/10.1016/j.sysarc.2021.102152>
- [93] Liu, B., Zhao, W., Sun, Q. (2017). Study of object detection based on Faster R-CNN, *2017 Chinese Automation Congress (CAC)*, Jinan, China, pp. 6233-6236, doi: 10.1109/CAC.2017.8243900
- [94] Sorgun, Ö. (2022). *X-Ray Görüntülerinde Faster R-CNN Kullanılarak Yasaklı Nesne Tespiti*, Yüksek Lisans Tezi, Sakarya Uygulamalı Bilimler Üniversitesi Lisansüstü Eğitim Enstitüsü, 87s.
- [95] Girshick, R., Donahue, J., Darrell, T., Malik, J. (2014). Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587. Columbus, OH, USA: IEEE. <https://doi.org/10.1109/CVPR.2014.81>
- [96] Ren, S., He, K., Girshick, R.B., and Sun, J. (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39, pp.1137-1149
- [97] Salvador, A., Giró-I-Nieto, X., Marqués, F. et al. (2016). Faster R-CNN Features for Instance Search", *IEEE Conference on Computer Vision and Pattern Recognition Workshops*. IEEE Computer Society, pp. 394-401
- [98] Liu, Y. (2018). An Improved Faster R-CNN for Object Detection, *2018 11th International Symposium on Computational Intelligence and Design (ISCID)*, Hangzhou, China, pp. 119-123, doi: 10.1109/ISCID.2018.10128
- [99] Girshick, R. (2015). Fast R-CNN. In *Proceedings of the IEEE international conference on computer vision*, pp. 1440-1448
- [100] Faster R-CNN Explained for Object Detection Tasks, (2022, 15 Kasım). Erişim adresi: <https://blog.paperspace.com/faster-r-cnn-explained-object-detection/>
- [101] Redmon, J., Farhadi, A. (2017). YOLO9000: Better, Faster, Stronger. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6517–25. Honolulu, HI: IEEE, doi: <https://doi.org/10.1109/CVPR.2017.690>
- [102] Huang, R., Pedoeem, J., Chen, C. (2018). YOLO-LITE: A Real-Time Object Detection Algorithm Optimized for Non-GPU Computers," *2018 IEEE International Conference on Big Data (Big Data)*, Seattle, WA, USA, pp. 2503-2510, doi:10.1109/BigData.2018.8621865
- [103] Redmon, J., Farhadi, A. (2018). YOLOv3: An Incremental Improvement. *ArXiv:1804.02767 [Cs]*. <http://arxiv.org/abs/1804.02767>

- [104] Liu, C., Tao, Y., Liang, J., Li, K., Chen, Y. (2018) "Object Detection Based on YOLO Network," *2018 IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC)*, Chongqing, China, pp. 799-803, doi: 10.1109/ITOEC.2018.8740604
- [105] Wei, L. Dragomir, A., Erhan, D., Szegedy, C., Scott, R., Fu, C. (2016). SSD: Single Shot MultiBox Detector, *ECCV 2016. Lecture Notes in Computer Science*, 9905, Springer, Cham. https://doi.org/10.1007/978-3-319-46448-0_2
- [106] Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L. (2010). ImageNet: A large-scale hierarchical image database, 2009 IEEE conference on computer vision and pattern recognition. IEEE, 2009. pp. 248-255.
- [107] Ramsundar, B. and Zadeh, R. B. (2018). TensorFlow for deep learning: from linear regression to reinforcement learning, " O'Reilly Media, Inc."
- [108] Gazioğlu, B. S. (2021). Dermatolojide Melanoma Malignansının Sınıflandırılması, Doktora Tezi, İstanbul Teknik Üniversitesi Lisansüstü Eğitim Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı, 101s.
- [109] Majtnar, T., Bajic, B., Yildirim, S., Hardeberg, J. Y., Lindblad, J., Sladoje, N. (2018). Ensemble of Convolutional Neural Networks for Dermoscopic Images Classification, *CoRR*, abs/1808.05071, <http://arxiv.org/abs/1808.05071>, 1808.05071
- [110] Activation Function ,(2022, 14 Kasım). Erişim adresi: <https://machine-learning.paperspace.com/wiki/activation-function>
- [111] AlexNet, VGGNet, Inception ve ResNet Nedir?, (2022, 14 Kasım). Erişim adresi: <https://frightera.medium.com/alexnet-vggnet-inception-ve-resnet-nedir-bddc7482918b>
- [112] Dong, C., Loy, C. C., He, K., & Tang, X. (2015). Image super-resolution using deep convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(2), pp. 284-296
- [113] VGG16 – Convolutional Network for Classification and Detection, (2022, 14 Kasım). Erişim adresi: <https://neurohive.io/en/popular-networks/vgg16/>
- [114] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Rabinovich, A. (2014). Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-9
- [115] Di, W., Bhardwaj, A, Wei, J. (2018). *Deep Learning Essentials*, Packt Publishing, pp. 1-217
- [116] Kaiming, h., Zhang, X., Ren, S., Sun, J. (2016). Identity Mappings in Deep Residual Networks, *ECCV 2016*, In: *CoRR* abs/1603.05027. URL: <http://arxiv.org/abs/1603.05027> (cited on page 279)
- [117] Shanmugamani, R. (2018). *Deep Learning for Computer Vision Expert techniques to train advanced neural networks using TensorFlow and Keras*, Packt Publishing
- [118] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*,
- [119] Rosebrock, A. (2017). *Deep Learning for Computer Vision with Python*, ,*Published By Pyimagesearch*, pp. 1-332
- [120] Rabano, S. L., Cabatuan, M. K., Sybingco, E., Dadios, E. P., Calilung, E. J. (2018). Common Garbage Classification Using MobileNet, *2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM)*, Baguio City, Philippines, pp. 1-4, doi: 10.1109/HNICEM.2018.8666300
- [121] Howard, A.G, Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., et al. (2017) MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications.

- [122] Sinha, D., El-Sharkawy, M. (2019). Thin MobileNet: An Enhanced MobileNet Architecture, *2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, New York, NY, USA, pp. 0280-0285, doi: 10.1109/UEMCON47517.2019.8993089
- [123] Ioffe, S., Szegedy, C.(2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift, *Proceedings of the 32nd International Conference on Machine Learning, PMLR*, 37, pp.448-456
- [124] Sae-Lim, W., Wettayaprasit, W., Aiyarak, P. (2019). Convolutional Neural Networks Using MobileNet for Skin Lesion Classification, *2019 16th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, Chonburi, Thailand, pp. 242-247, doi: 10.1109/JCSSE.2019.8864155
- [125] Aslam N., and Sharma V. Foreground detection of moving object using Gaussian mixture model, *Proc. 2017 IEEE Int. Conf. Commun. Signal Process. ICCSP 2017*, vol. 2018-January, pp. 1071–1074, Feb. 2018, doi: 10.1109/ICCSP.2017.8286540
- [126] Eğim, (2022, 11 Aralık).). Erişim adresi <https://tr.wikipedia.org/wiki/E%C4%9Fim>
- [127] Monawar, T., Mahmud, S. B., Hira, A. (2017). Anti-theft vehicle tracking and regaining system with automatic police notifying using Haversine formula, *2017 4th International Conference on Advances in Electrical Engineering (ICAEE)*, Dhaka, Bangladesh, pp. 775-779, doi: 10.1109/ICAEE.2017.8255459
- [128] Saputra, K., Nazaruddin, N., Yunardi, D. H., Andriyani, R. (2019). Implementation of Haversine Formula on Location Based Mobile Application in Syiah Kuala University, *2019 IEEE International Conference on Cybernetics and Computational Intelligence (Cybernetics Com)*, Banda Aceh, Indonesia, pp.40-45, doi:10.1109/cyberneticscom. 2019.8875686
- [129] Alam, C.N., Manaf, K., Atmadja, A. R., Aurum, D.K. (2016). Implementation of haversine formula for counting event visitor in the radius based on Android application, *2016 4th International Conference on Cyber and IT Service Management, Bandung, Indonesia*, pp. 1-6, doi: 10.1109/CITSM.2016.7577575
- [130] Han, S.D., Ullah, H. (2019).A survey of advances in vision-based vehicle re-identification, *Computer Vision and Image Understanding*, 182, pp. 50-63, doi: 10.1109/CITSM.2016.7577575
- [131] Zhao, Y., Shen, C.,Wang, H.,Chen, S. (2019).Structural analysis of attributes for vehicle re-identification and retrieval", *IEEE Trans. Intell. Transp. Syst.*, 21(2), pp.723-734
- [132] Wu, C.W., Liu, C.T., Chiang, C.E., Tu, W.C., Chien, S.Y. (2018). Vehicle re-identification with the space-time prior, *Proc. CVPR Workshops*, pp. 121-128
- [133] Liu, X., Liu, W., Mei, T., Ma, H. (2016). A Deep Learning-Based Approach to Progressive Vehicle Re-identification for Urban Surveillance. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds) *Computer Vision – ECCV 2016. ECCV 2016. Lecture Notes in Computer Scienc*, Springer, 9906. pp. 869–884, https://doi.org/10.1007/978-3-319-46475-6_53
- [134] Shen, Y. Xiao, Li, H.,Yi,S., Wang,X. (2017). Learning deep neural networks for vehicle re-id with visual-spatio-temporal path proposals. *In Proceedings of the IEEE international conference on computer vision*, pp. 1900-1909