

T.C.
GEBZE TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

VIDEO GÖZETLEME SİSTEMLERİNDE ALTUZAY
ÖĞRENMESİ YAKLAŞIMI İLE ŞÜPHELİ NESNE TESPİTİ

HASAN HÜSEYİN SÖNMEZ
YÜKSEK LİSANS TEZİ
ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

GEBZE
2015

T.C.
GEBZE TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

VIDEO GÖZETLEME SİSTEMLERİNDE
ALTUZAY ÖĞRENMESİ YAKLAŞIMI İLE
ŞÜPHELİ NESNE TESPİTİ

HASAN HÜSEYİN SÖNMEZ
YÜKSEK LİSANS TEZİ
ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

DANIŞMANI
YRD. DOÇ. DR. ALİ KÖKSAL HOCAOĞLU

GEBZE

2015

T.R.
GEBZE TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

**ABANDONED OBJECT DETECTION IN
VIDEO SURVEILLANCE SYSTEMS VIA
SUBSPACE LEARNING**

HASAN HÜSEYİN SÖNMEZ
A THESIS SUBMITTED FOR THE DEGREE OF
MASTER OF SCIENCE
DEPARTMENT OF ELECTRONIC ENGINEERING

THESIS SUPERVISOR
ASSIST. PROF. DR. ALİ KÖKSAL HOCAOĞLU

GEBZE

2015



YÜKSEK LİSANS JÜRİ ONAY FORMU

GTÜ Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 29/06/2015 tarih ve 2015/41 sayılı kararıyla oluşturulan jüri tarafından 10/07/2015 tarihinde tez savunma sınavı yapılan Hasan Hüseyin SÖNMEZ'in tez çalışması Elektronik Mühendisliği Anabilim Dalında YÜKSEK LİSANS tezi olarak kabul edilmiştir.

JÜRİ

ÜYE

(TEZ DANIŞMANI) : Yrd. Doç. Dr. Ali Köksal HOCAOĞLU

ÜYE

: Doç. Dr. Yusuf Sinan AKGÜL

ÜYE

: Yrd. Doç. Dr. Koray KAYABOL

ONAY

Gebze Teknik Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun

...../...../..... tarih ve/..... sayılı kararı.

İMZA/MÜHÜR

ÖZET

İnsanların yoğun olarak bulundukları yerlerde gerçekleşebilecek terör olaylarına karşı artan güvenlik endişeleri sonucunda, akıllı video gözetleme sistemlerine olan talep büyümektedir. Bu çalışmada, şüpheli/sahipsiz/terk edilmiş nesneleri güvenlik kameralarından elde edilen görüntüleri kullanarak otomatik olarak tespit edecek bir video gözetleme algoritması geliştirilmiştir. Kullanılan yöntemde, ikili ön plan modeli uygulanmıştır. Arka plan ve ön plan nesnelerinin bölütlenmesi GoDec olarak adlandırılan matris ayrıştırması yöntemi $X = L + S$ modeli ile hızlı ve gürbüz sonuç verecek şekilde yapılmaktadır. GoDec algoritması ile nesnelerin hatları korunduğundan, sabit duran insan ve bir valiz birbirinden rahatlıkla ayrılabilir. Bu ön plan modelleri kullanılarak, statik ön plan nesneleri tespit edilmektedir. GoDec algoritması ile ön plan nesnelerinin tespiti, geleneksel yöntemlere göre oldukça iyi yapılmakta ve ön plan nesnelerinin hatları korunabilmektedir. Böylece, farklı türde nesneleri birbirinden ayırmak daha kolay olmaktadır. Bu tez çalışması ile altuzay öğrenmesi yönteminin, şüpheli nesne tespiti problemi üzerinde kullanılması için yeni bir yaklaşım/yapı önerilmiştir. Bu yaklaşım, özellikle arka plan başlangıç değeri sorununa karşı oldukça gürbüz sonuç vermektedir.

Anahtar Kelimeler: Şüpheli Nesne Tespiti, Arka Plan Çıkarımı, Altuzay Öğrenmesi.

SUMMARY

The result of increasing security concerns for the terrorist activities in public places where people gather densely, there has been a growing interest for the smart video surveillance system. In this thesis, using videos obtained from security cameras, an automatic unattended/abandoned object detection algorithm is designed and implemented. Dual foregrounds method is used to detect temporally static regions. A fast and robust matrix decomposition algorithm, named GoDec, as $X = L + S$ is used for foreground/background separation. An evidence score is inferred at each pixel by applying a set of hypotheses on the foreground images. This provides temporal consistency. Segmentation quality of GoDec algorithm is very good and shapes of the objects are well retained. This provides easy discrimination of different objects such as standing person and a luggage. In this study, we propose a novel framework for abandoned object detection via subspace learning. Proposed approach is robust especially against the background initialization errors.

Key Words: Abandoned Object Detection, Background Subtraction, Subspace Learning.

TEŞEKKÜR

Başta, yüksek lisans eğitimimde ve akademik hayatımda desteğini ve yardımlarını hiçbir zaman esirgemeyip bilgisi ile bu çalışmanın oluşmasının yolunu açan danışmanım Yrd. Doç. Dr. Ali Köksal HOCAOĞLU'na,

Çalışma arkadaşım Nagehan ÇAPA'ya,

Teşvik ve fikirlerini her zaman paylaşan değerli arkadaşlarım Yusuf ATMACA ve Teğmen Ali Asım KAPTANLAR'a,

Ve her zaman yanımda duran ve bana destek olan, aileme teşekkürlerimi sunarım.

İÇİNDEKİLER

	<u>Sayfa</u>
ÖZET	v
SUMMARY	vi
TEŞEKKÜR	vii
SİMGELER VE KISALTMALAR DİZİNİ	x
ŞEKİLLER DİZİNİ	xi
TABLolar DİZİNİ	xiii
1. GİRİŞ	1
1.1. Tezin Amacı, Katkısı ve İçeriği	1
1.2. Motivasyon	2
1.3. Tezin Organizasyonu	2
2. LİTERATÜR ÖZETİ	3
2.1. Arka Plan Çıkarımı	8
2.1.1. Arka Plan Modellemesi	9
2.1.1.1. İstatistiksel Arka Plan Modeli	10
2.1.1.2. W^4 Sistemi	12
2.1.1.3. Bulanık Mantık ile Arka Plan Modellemesi	14
2.1.1.4. Altuzay Arka Plan Modeli	17
2.1.2. Arka Plan Güncelleme	20
2.1.2.1. İkili Arka Plan Güncelleme Modeli	21
2.1.2.2. Hareketli Ortalama (Running Average)	25
2.2. Optik Akış Yöntemleri	27
2.3. Performans Değerlendirme Kriterleri ve Veri Setleri	27
2.3.1. Veri Setleri	27
2.3.2. Performans Ölçme Kriterleri	28
3. ŞÜPHELİ NESNE TESPİT ALGORİTMASI	33
3.1. Altuzay Öğrenmesi ile Ön Plan Tespiti	36
3.1.1. Gri Seviyeli Ön Plan Görüntüsü Eşikleme	37
3.2. Statik Nesne Maskesi ve Şüpheli Nesne Hipotezi	39
3.3. Statik Nesne Maskesi	41

3.4. Perdeleme (Occlusion) Engelleme	41
3.5. Alarm Üretme	42
4. DENEYSEL SONUÇLAR VE KARŞILAŞTIRMA METRİĞİ	44
4.1. Nesneye Dayalı Performans Metriği	44
4.2. Deneysel Sonuçlar	45
4.3. GoDec-II ile Mevcut Yöntemlerin Karşılaştırılması	52
4.3.1. Wang ve Liu Tarafından Önerilen Yaklaşım	58
4.3.2. Porikli ve Arkadaşları Tarafından Önerilen Yaklaşım	62
4.3.3. Sigari ve Arkadaşları Tarafından Önerilen Arka Plan Çıkarımı	63
4.4. GoDec-II'nin Avantajları/Dezavantajları	65
5. SONUÇ VE GELECEK ÇALIŞMALAR	67
KAYNAKLAR	68
ÖZGEÇMİŞ	74

SİMGELER ve KISALTMALAR DİZİNİ

<u>Simgeler ve</u>	<u>Açıklamalar</u>
<u>Kısaltmalar</u>	
Σ	: Kovaryans Matrisi
σ^2	: Varyans
μ	: Ortalama
η	: Normal Dağılım
I	: Birim Matris
AVSS	: Advanced Video and Signal-Based Surveillance
B	: Arka Plan (Background) Modeli
EM	: Expectation Maximization
FCH	: Fuzzy Color Histogram
FPS	: Frames Per Second
IIR	: Infinite Impulse Response
MoG	: Gauss Karışımı (<i>Mixture of Gaussian</i>)
OF	: Optical Flow
PDR	: Perturbation Detection Rate
RGB	: Kırmızı, Yeşil, Mavi Renk Uzayı
ROC	: Receiver Operating Characteristics
OTCBVS	: Object Tracking and Classification Beyond the Visible Spectrum
YCbCr	: Luminans, Mavi Renk Farkı, Kırmızı Renk Farkı

ŞEKİLLER DİZİNİ

<u>Sekil No:</u>	<u>Sayfa</u>
2.1: Dedeoğlu'na Göre Nesne Tespit Yöntemleri Sınıflandırması.	4
2.2: Li'ye Göre Nesne Tespit Yöntemleri Ağaç Yapısı.	5
2.3: Literatürdeki Hareketli Nesne Tespiti Yaklaşımları İçin Önerilen Ağaç Yapısı (Sınıflandırma).	8
2.4: W4 Sisteminin Blok Şeması.	13
2.5: UM (Solda) ve UV (Sağda) İçin Birincil Üyelik Fonksiyonları.	16
2.6: Wang ve arkadaşları [12]'de Önerilen Şüpheli Nesne Sistemi Blok Şeması.	21
2.7: $\alpha_{\min}=0.9$ için α - FBGS Diyagramı.	26
3.1: Şüpheli Nesne Tespiti Algoritması Genel Blok Şeması.	35
3.2: Önerilen Şüpheli Nesne Tespit Algoritması (GoDec-II) Blok Şeması.	36
3.3: Örnek Bir Karenin (Kare No: 854) a) Seyrek (Ön Plan) Bileşeni, b) S Bileşeninin Histogramı.	38
3.4: Kare No: 854'ün Otsu İle Eşikleme Sonucu.	38
3.5: Olası Durum Hipotezleri.	40
4.1: Önerilen Yöntemin AVSS_AB_Easy Videosuna Ait Sonuçları.	46
4.2: $F = 3200$, a) Orijinal Sahne, b) AVSS_AB_Easy Statik Nesne Maskesi.	47
4.3: $F = 3200$, a) Aday Şüpheli Nesne Maskesi, b) Tespit Edilen Şüpheli Nesne.	47
4.4: GoDec-II Algoritmasının AVSS_AB_Medium Videosu İçin Performansı.	48
4.5: $F=2224$, a) Orijinal Kare, b) Statik Nesne Maskesi.	49
4.6: a) Aday Şüpheli Nesne Maskesi, b) Tespit Edilen Şüpheli Nesne.	49
4.7: PETS2006-1 Videosu Test Sonucu.	50
4.8: PETS2006 - 1 Videosundan Örnek Bir Kare ($F = 2280$).	50
4.9: $F=2000$ İçin a) Kısa Süreli Ön Plan Maskesi, b) Uzun Süreli Ön Plan Maskesi.	51

4.10:	PETS2006-2 Videosu İçin Test Sonucu.	51
4.11:	CDW-2012 Gece Videosu İçin GoDec-II Test Sonucu.	52
4.12:	F = 2495 İçin a) Statik Nesne Maskesi, b) Tespit Edilen Araç.	52
4.13:	GTU Veri Setinden Örnek Bir Kare a) Statik Nesne Maskesi, b) Tespit Sonucu.	55
4.14:	GTU_01 Videosundan Örnek Görüntü.	55
4.15:	GTU_z23 Videosundan Örnek Bir Kare.	56
4.16:	GTU_z23 a) Kısa Süreli, b) Uzun Süreli Ön Plan Maskeleri.	56
4.17:	AVSS_AB_Easy Videosunun İlk Karesi.	58
4.18:	a) Sık Güncellenen Arka Plan, b) Yavaş Güncellenen Arka Plan.	59
4.19:	Kare No: 1240 İçin Tespit Görüntüsü.	59
4.20:	Wang Tarafından Önerilen Yaklaşımın Nesne Tabanlı Metriğe Göre Tespit Sonucu.	60
4.21:	CDW-2012 Gece Videosu İçin Wang ve Liu Tarafından Önerilen Yöntem, Statik Nesne Maskesi.	61
4.22:	GTU_z23 İçin Wang ve Liu Tarafından Önerilen Yöntem, a) Kısa Süreli, b) Uzun Süreli Ön Plan Maskesi.	61
4.23:	Porikli Tarafından Önerilen İkili Ön Plan Yönteminin AB_Easy Videosunda Sonucu.	62
4.24:	Porikli Tarafından Önerilen İkili Ön Plan Yönteminin AB_Medium Videosunda Sonucu.	63
4.25:	Hareketli Ortalama ve Arka Plan Çıkarımı, PETS2001 Sonucu.	64
4.26:	Bulanık Hareketli Ortalama ve Bulanık Arka Plan Çıkarımı, PETS2001 Sonucu.	64
4.27:	PETS2001 Üzerinde GoDec İle Ön Plan Tespiti Sonucu.	65

TABLÖLAR DİZİNİ

<u>Tablo No:</u>	<u>Sayfa</u>
2.1: Beş Nesne Tespit Algoritmasının Performansı.	30
2.2: Maddalena ve Petrosino Tarafından Uygulanan Performans Kriteri.	31
4.1: GoDec-II Algoritmasının Performans Sonuçları.	54
4.2: Wang ve arkadaşları [12]'de Önerilen Yöntemin Performans Sonuçları.	57
4.3: Porikli [49]'da Önerilen Yöntemin Performans Sonuçları.	57

1. GİRİŞ

İnsanların yoğun olarak bulundukları alışveriş merkezleri, metro istasyonları, duraklar vb. yerlerde gerçekleşebilecek terör olaylarına karşı artan güvenlik endişeleri sonucunda, video gözetleme sistemlerine olan talep büyümektedir. Geleneksel olarak, gözetleme işlemi, operatörler tarafından çok sayıda güvenlik kamerasından alınan veriler üzerinde yapılmaktadır. Eş zamanlı incelenen verinin çok fazla olması, operatör performansını düşürmekte, dolayısıyla güvenlik tehditlerini artırmaktadır. Bu yüzden, otomatik/yarı otomatik gözetleme sistemleri geliştirilerek operatör üzerindeki yükün azaltılması üzerine çalışılmaktadır.

1.1. Tezin Amacı, Katkısı ve İçeriği

Bu çalışmada, toplu taşıma ve kamu alanlarında şüpheli/sahipsiz/terk edilmiş nesneleri güvenlik kameralarından elde edilen görüntüler ile tespit edecek bir video gözetleme algoritması geliştirilmiştir. Şüpheli bir durum tespit edildiğinde, algoritma bir alarm sinyali ile operatörü (kullanıcıyı) uyaracak ve operatörün dikkatini spesifik bir monitöre çekecektir. Bu şekilde, operatörün, tüm monitörleri aynı anda izlemek zorunda kalmaması ve operatör performansının artması beklenmektedir.

Problem, şüpheli nesnenin orijinal sahnede bulunmadığı, kısa bir süre önce getirilmiş olduğu ve bırakıldıktan sonra hareketsiz kaldığı olgusu üzerine kurulmuştur. Arka plan/ön plan bölütlemesi altuzay öğrenmesi (Subspace Learning) yaklaşımı ile gerçekleştirilmektedir. Bu işlem için GoDec algoritması seçilmiştir. Bu algoritma, oldukça hızlı ve iyi sonuç vermektedir. Data, iki farklı hızda işlenerek biri uzun (long term) diğeri kısa (short term) süreli iki ön plan maskesi oluşturulmaktadır. Bu iki ön plan maskesi karşılaştırıldığında terk edilmiş nesneye ait pikseller güven haritasında baskın hale gelmektedir. Güven haritası kullanılarak, zamansal tutarlılık sağlanmaktadır.

Altuzay öğrenme yaklaşımı, literatürde statik nesne tespiti problemi için daha önce kullanılmamıştır. Bu açıdan, bu tez çalışmasında probleme önerilen yaklaşım, yeni bir yaklaşımdır. Altuzay öğrenmesinin arka plan/ön plan bölütlemesindeki hassasiyetinden, şüpheli nesne tespiti probleminde yararlanılmıştır. Bu sayede, nesnelerin dış hatları oldukça iyi muhafaza edilmektedir. Bu da, farklı türdeki ön

plan nesnelerinin (örneğin sabit duran insan ve bir valiz) birbirinden ayırt edilmesini kolaylaştırmaktadır. Böylece yanlış alarm oranı daha da düşürülerek sistemin güvenilirliği artırılabilir.

1.2. Motivasyon

Video görüntüleri kullanarak bir sahnedeki hareketli (veya kısmi hareketli) nesnelerin aktivitelerini anlama, hem bilimsel olarak ilgi çekici bir problem hem de birçok potansiyel uygulamaya sahip verimli bir alandır. Bu yüzden, teknik ekipmanların da gelişmesi ve maliyetlerinin düşmesi ile birlikte, araştırmacıların, bazı ticari şirketlerin ilgisini çeken bir konudur. Bu problem üzerinde çalışmamızdaki motivasyonumuz, hareketli nesne tespit ve sınıflandırma kapasitelerine sahip, iç ve dış mekânlara uyumlu, video gözetleme sistemi üzerinde çalışabilecek bir algoritma gerçekleştirmektir. Nesne takip ve aktivite tanıma yöntemlerinin geliştirilen sisteme entegrasyonu, bu çalışmanın bir sonraki adımını oluşturabilirler.

1.3. Tezin Organizasyonu

Tezin ileriki bölümleri şu şekilde düzenlenmiştir. Bölüm 2’de daha önce yapılmış çalışmaların kısa bir özeti olarak literatür taraması yapılmış ve önerilmiş olan yaklaşımların sınıflandırıldığı bir ağaç yapısı oluşturulmuştur. Bunun yanında konu üzerinde geliştirilen algoritmaların performanslarının nasıl değerlendirildiğiyle alakalı, performans değerlendirme yöntemleri incelenmiş ve geliştirilen yöntemin performansını ölçmek için bir performans metriği belirlenmiştir. Ayrıca, bazı yaygın ön plan nesne tespiti yaklaşımları performansları ile birlikte incelenmiş ve şüpheli nesne tespiti problemi üzerinde test edilmiştir. Şüpheli nesne tespiti problemi için, bu çalışmada önerilen yöntem Bölüm 3’te anlatılmıştır. Bölüm 4’te, geliştirilen sistemin bazı açık veri setleri üzerinde test edilmesiyle elde edilen deneysel sonuçlar sunulmuştur. Bölüm 5’te gelecek çalışmalar için bazı öneriler ile tez sonlandırılmıştır.

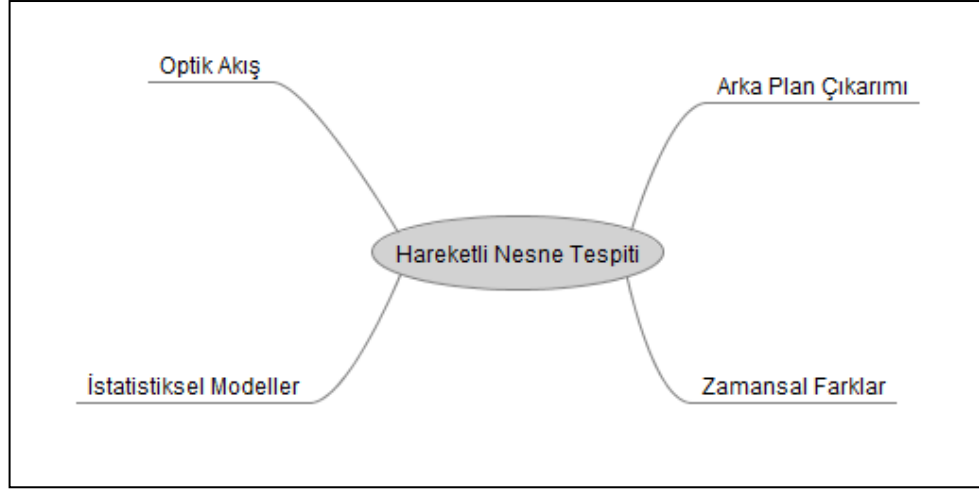
2. LİTERATÜR ÖZETİ

Trafik görüntüleme, şehir merkezleri, meydanlar gibi açık alanlar veya metro, havalimanları gibi kapalı alanlar için geliştirilen farklı uygulamalar, beraberinde aynı probleme farklı çözümlerle yaklaşmayı gerektirmiştir. Bu yüzden, şüpheli nesne tespiti için, son zamanlarda literatürde birçok farklı yöntem önerilmiştir [1]-[4].

Video görüntülerinde hareketli nesneleri tespit ve takip etmek için renk, hareket, kenarlar gibi görüntü öznitelikleri kullanılmaktadır [5]. Ek olarak, hareketli nesneleri tespit etmek için bölütleme yaklaşımından da yararlanılır. Bunları uzamsal bölütleme ve zamansal bölütleme olarak iki başlık altında inceleyebiliriz. Uzamsal bölütleme, sayısal görüntü uzayında yapılan bölütlemeye dayalıdır. Sayısal görüntü bölütleme, bölgesel veya global olarak, bir görüntüyü iki seviyeli veya daha çok seviyeli farklı bölgelere ayırmak için kullanılmaktadır. Zamansal bölütlemeye ise, görüntüyü iki seviyeli yapmak için farklı video karelerinden faydalanılmaktadır. Nesne tespit/takip algoritmaları, genellikle, ardışık iki video karesi arasındaki benzerliği veya farklılığı karakterize eden ölçüye dayanmaktadır [6]. Kareler arası (inter-frame) veya kare içi (intra-frame) mesafe metrikleri genellikle Bhattacharyya [7], Hamming, Mahalanobis, Manhattan veya basit Öklid metrikleridir [8].

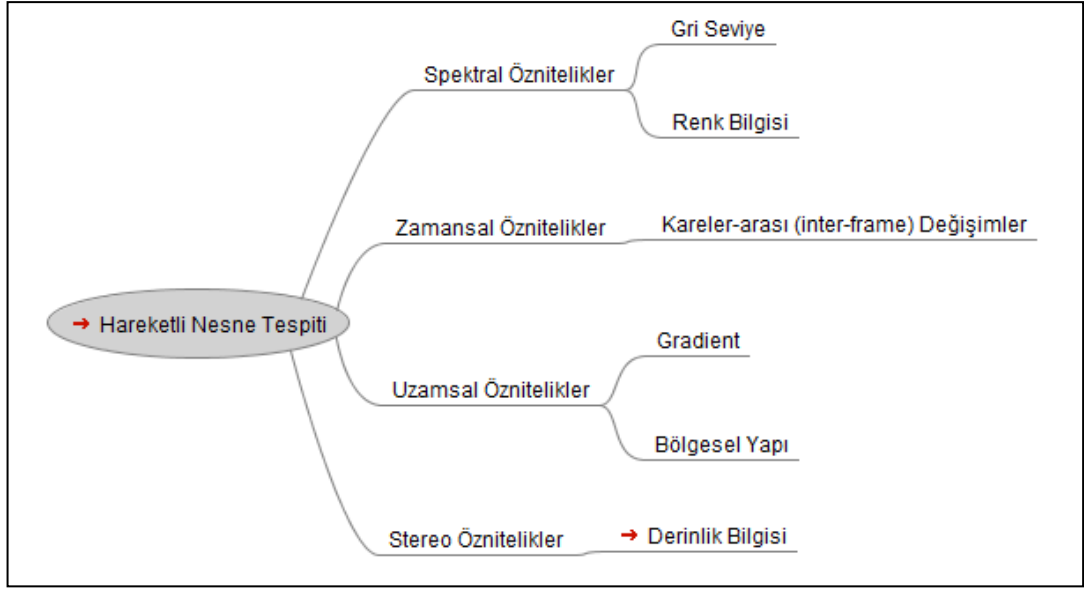
Bu bölümde, şüpheli nesne tespiti problemi için literatürde önerilen en yaygın yöntemler incelenmiştir. Nesne tespiti konusunda yapılmış çalışmalarda yöntemler farklı şekillerde sınıflandırılmıştır. En çok kullanılan yaklaşımlar arka plan çıkarımı, istatistiksel modeller, zamansal fark ve optik akıştır [9]. Arka plan çıkarımı teknikleri, hareketli bölgelerdeki ilgili piksellerin çoğunu çıkarabilmekte, ancak genellikle durağan nesnelerin kaybolması gibi değişimlere veya ani ışık değişimlerine karşı hassastır [10]-[12]. İstatistiksel yöntemler, arka plan çıkarımı tekniklerinden esinlenilerek geliştirilmiştir. Ön plan pikselleri, her pikselin istatistik bilgisi arka plan istatistiği ile karşılaştırılarak bulunmaktadır. Bu yaklaşımlar gürültü ve ışık değişimlerine karşı güvenilir olduklarından daha fazla tercih edilmeye başlanmışlardır [13], [14]. Zamansal fark yaklaşımı, ardışık (iki veya üç) kare arasındaki piksel-piksel fark alınarak hareketli bölgelerin tespit edildiği bir tekniktir. Bu yöntem, dinamik değişimlere oldukça fazla adapte olabilmesine karşı, genellikle tüm ilgili pikselleri, örneğin tek renkli bölgelerde, çıkarma konusunda başarısız olmaktadır [15]. Optik akış, zaman üzerinden hareket kestirimi yaparak, görüntüdeki

hareketli bölgeleri tespit etmeye dayalı bir yöntemdir. Kamera hareket halindeyken dahi tespit yapabilmektedir. Bunun yanında, çoğu optik akış yöntemi hesaplama bakımından çok karmaşıktır ve gerçek zamanlı sistemlerde özelleştirilmiş donanım olmadan kullanılamazlar [9].



Şekil 2.1: Dedeoğlu'na Göre Nesne Tespit Yöntemleri Sınıflandırması.

Li ve arkadaşları [14]'te, ön plan nesne tespiti için literatürde önerilen algoritmaları, kullanılan zamansal, spektral ve uzamsal özniteliklere göre sınıflandırmıştır. Spektral öznitelikler gri seviye veya renk bilgisi, uzamsal öznitelikler görüntünün gradyan ve bölgesel yapısı, zamansal öznitelikler ise piksellerdeki kareler arası değişimler ile özdeşleştirilebilir. Bayona ve arkadaşları [16]'da arka plan çıkarımına dayalı ön plan tespiti için mevcut yaklaşımları temelde, bir veya daha fazla arka plan modeli kullanılmasına göre iki sınıfa ayırmıştır. Daha sonra bu iki sınıf da, kare-kare analiz ve alt örneklemeyle dayalı analiz olmak üzere ikiye ayrılmıştır.



Şekil 2.2: Li'ye Göre Nesne Tespit Yöntemleri Ağaç Yapısı.

Grimson ve arkadaşları [17] ve McKenna ve arkadaşları [18]'de arka planın istatistiksel modellerini, Ohta [19]'da Gauss farklarının minimize edilmesini, Collins ve arkadaşları [20]'de kare (frame) farkları ve istatistiksel arka plan modellerinin bir kombinasyonunu kullanmıştır.

Wang ve arkadaşları [14]'ye göre, geleneksel şüpheli/kayıp nesne tespiti algoritmaları takip temellidir. Bu yöntemlerde, sahnenin kalabalık olmadığı, perdelenmenin en az olduğu ve hareketli nesnelerin etkili bir şekilde yalnızca hareket bilgisi ile başlatılabileceği varsayımı yapılmaktadır. Bunların dışında, son zamanlarda, bazı piksel bazlı tespit yöntemleri de önerilmiştir [3], [4]. Bu yöntemler nesne takibi ile birlikte kullanılabildikleri gibi, tek başına sisteme de uygulanabilmektedirler.

Kim ve arkadaşları [21]'de dinamik dokuya sahip (sallanan ağaç dalları vb.) sahnelerde kullanılabilecek basit ve etkili bir bulanık mantık arka plan çıkarımı yöntemi önermiştir. Arka plan hareketleri tarafından oluşturulan renk değişimleri, bulanık mantık yaklaşımıyla büyük ölçüde sönümlenmektedir. Bu yüzden hiçbir parametre kestirime ihtiyaç duyulmamaktadır. Bölgesel öznitelikler, bulanık (fuzzy) renk histogramından (FCH) elde edilmektedir. Ardından arka plan, güvenilir bir şekilde, bölgesel FCH öznitelikleri arasındaki benzerlikler hesaplanarak, çevrimiçi bir güncelleme mekanizması ile oluşturulmaktadır.

Altuzay öğrenmesi konusu, yakın tarihte bilgisayarla görme alanında üzerinde daha fazla çalışma yapılmaya başlanan bir yaklaşımdır [15]. Bu konuda, şu ana kadar oldukça çeşitli yaklaşımlar öne sürülmüştür [22]. Candes ve arkadaşları [23]’te Robust Principal Component Analysis (RPCA) yaklaşımını önermişlerdir. Lin ve arkadaşları [24]’te Low Rank Recovery (LRR) yaklaşımını önermiştir. Bucak ve Günsel [25]’te artımlı negatif olmayan matris ayrıştırma yöntemi ile arka plan modelleme önermişlerdir.

Matris ayrıştırma yöntemlerinde, her video karesi, yeni bir gözlem olarak sütun vektör formunda ifade edilir. Dolayısıyla, ayrıştırılacak data, piksel sayısı kadar satır, kare sayısı kadar sütunu olan bir matrise dönüştürülür. Ardından, uygulanan yaklaşıma göre, negatif olmayan iki bileşenin çarpımına (Matrix Factorization) veya düşük mertebeli (Low rank) ve Seyrek (Sparse) bileşenlerine ayrıştırılır. Matris ayrıştırma, konveks optimizasyon problemine (principal component pursuit) dönüştürülür ve konveks kısıtlar üzerinden data matrisini bileşenlerinden yeniden oluşturma hatası minimize edilerek çözülür [23]. Bir video dizisinde, durağan arka plan nesneleri bir kareden diğerine birbiri ile hayli ilintili piksel bölgelerine dönüşür, bu da video verisi içinde düşük mertebe (low rank) bir yapıyı işaret eder [26]. Durağan bölgeye (arka plan) ait olmayan pikseller ise veri matrisi içerisinde az sayıda olacağından, seyrek (sparse) yapıyı oluşturur. Matris ayrıştırma yaklaşımları video dizilerinde ön plan nesnelerini neredeyse mükemmel bölütlemelerine rağmen bazı dezavantajları da mevcuttur. Öncelikle, büyük veriler üzerinde çalıştıklarından dolayı, genel olarak yavaşlar ve hesaplama yükleri birden artabilmektedir. Bu durum, video gözetleme sistemlerine entegrasyonlarını zorlaştırmaktadır. Ayrıca, belirli sayıda gözlemin yığılmış olarak işlenmesi gerekmektedir (batch processing). Bu da, hesaplama yükünü artırmaktadır. Bu yüzden, matris ayrıştırma algoritmaları, artımlı ve hesaplama yükünü düşürecek şekilde tasarlanmalıdır. Belirtilen kısıtları sağlayan yaklaşımlar literatürde sunulmuştur [27], [28]. Matris ve tensor ayrıştırma yaklaşımlarının hesaplama zamanı için incelendiği bir çalışma [22]’de verilmiştir. Altuzay öğrenmesi daha detaylı olarak Bölüm 3’te ele alınacaktır.

Video görüntülerinde hareketli nesne tespiti yaklaşımları, literatürde çeşitli şekillerde sınıflandırılmıştır [29], [30]. Bu sınıflandırmalar Şekil 2.1 ve Şekil 2.2’de görülmektedir. Dedeoğlu [29]’da yapılan sınıflandırmada zamansal farklar, arka plan çıkarımı ve istatistiksel yöntemler temelde aynı yaklaşım üzerinden hareket

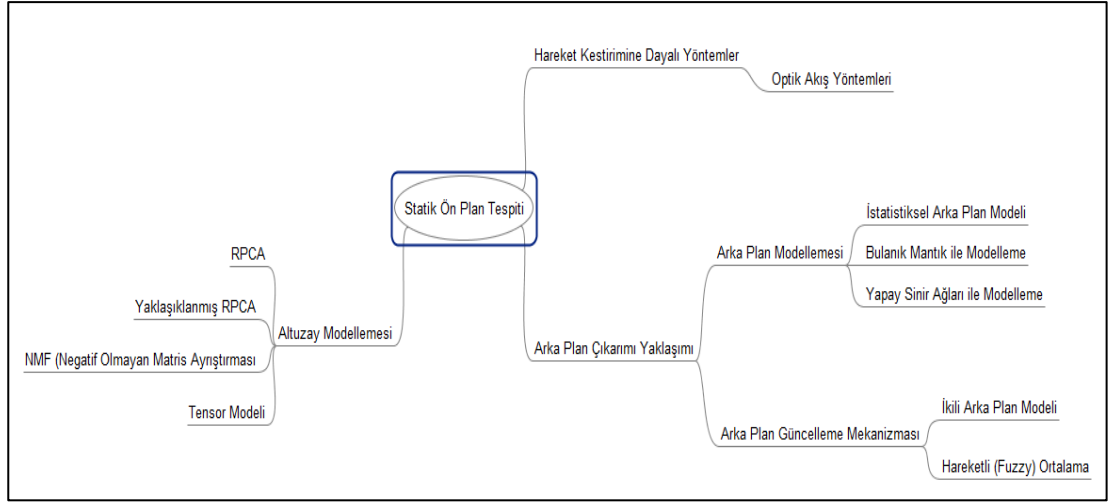
etmektedirler. Bu kategorilerin üçünde de genel mantık, mevcut kare ile bir referans görüntüsünün farkında, belli bir eşik değerinden yüksek piksellerin sahnedeki hareketli nesnelere ait bölgeler olduğu üzerine kurulmuştur [31]-[33]. Dolayısıyla bu üç farklı kategori birlikte arka plan çıkarımı olarak sınıflandırılabilir.

Bu çalışmada, sabit konuma ve görüş açısına sahip tek bir kameradan alınan video görüntülerinde, hareketli nesne tespiti için literatürde önerilen yaklaşımlar üç sınıfta incelenmiştir:

- Hareket kestirimine dayalı yöntemler,
- Ön plan nesne tespiti yöntemleri,
- Altuzay öğrenmesi yöntemleri.

Hareket kestirimine dayalı yöntemler optik akış yöntemleridir ve Optik Akış Yaklaşımları başlığı altında incelenmiştir. Ön plan nesne tespitine dayalı yöntemler ise arka plan çıkarımı uygulanarak sahnedeki hareketli ve kısmi hareketli nesnelerin tespit edildiği yaklaşımlardır. Arka plan çıkarımı yöntemleri de kendi içinde arka plan modellemesi ve kullanılan güncelleme mekanizmasına göre alt başlıklara ayrılmaktadır. Bu yöntemler ve alt sınıfları Şekil 2.3'te görülmektedir. Önerilen bu sınıflandırma yapısına göre, yaygın olarak kullanılan bazı yaklaşımlar bir sonraki bölümde ayrıntılı olarak incelenmiş ve test edilmiştir. Altuzay öğrenmesi, hareketli nesne tespiti problemi için üçüncü temel yaklaşım olarak ele alınabileceği gibi, arka plan modellemesi başlığı altında da incelenebilir.

Arka plan çıkarımı algoritmalarında belirlenmesi gereken bazı parametreler mevcuttur. Bunlar öznitelik tipi ve boyutudur. Öznitelik seçimi algoritmanın ele alacağı kritik durumlara göre değişiklik göstermektedir. Öznitelik boyutu piksel, blok veya küme olarak; öznitelik tipi de renk, ayırıt, stereo, hareket veya yapı (texture) öznitelikleri olarak seçilebilir [34]. Özniteliğin boyutu, gürültüye karşı bağışıklığı ve hassaslığı belirler. Örneğin, piksel tabanlı bir yöntem, piksel bazında bir hassaslık sağlarken blok veya küme temelli yöntemlere göre gürültüye karşı daha az bağışıklık sağlar [15].



Şekil 2.3: Literatürdeki Hareketli Nesne Tespiti Yaklaşımları İçin Önerilen Ağaç Yapısı (Sınıflandırma).

2.1. Arka Plan Çıkarımı

Arka plan çıkarımı, statik sahnelerde hareket segmentasyonu için, özellikle sıkça kullanılan bir tekniktir. Referans olarak bir arka plan görüntüsü oluşturulur ve bu görüntü, sahnedeki statik bölgeleri içerir. Ardından her yeni kareden, bu referans görüntüsü çıkarılır ve belirli bir eşik değerinin üzerindeki pikseller ön plan nesnelerine ait olarak gruplanır. Ön plan görüntüsü oluşturulduktan sonra, açma, kapama, erozyon gibi bir takım morfolojik işlemler uygulanarak gürültüyü azaltmak ve tespit edilen bölgelerde iyileştirme sağlanır. Referans arka plan, yeni görüntülerle belirli kurallara göre güncellenir. Böylece arka planın, sahnede zaman içinde meydana gelen dinamik değişimlere uyum sağlaması amaçlanır. Bu prosedür, genel bir arka planı çıkarma yöntemidir ve özelleşmiş yöntemler, bu yaklaşımı esas alarak, arka planı modelleme ve güncelleme mekanizması üzerinden türetilmektedir.

Video gözetleme konseptinde, arka plan çıkarımı yaklaşımlarında tipik olarak aşağıdaki zorluklarla karşılaşmaktadır [5]:

- **Kademeli ışık değişimleri:** Arka plan modelinin kademeli değişimlere çevrenin görünümüne göre uyum sağlanması gerekmektedir. Örneğin, dış ortamda, ışık yoğunluğu gün içinde değişim göstermektedir. Dolayısıyla, modelin de buna göre adapte olması beklenmektedir.

- Ani ışık değişimleri: Ani gerçekleşen ışık değişimleri genellikle arka plan modelleri tarafından ele alınmazlar ve arka plan modelini oldukça fazla etkilerler. Dolayısıyla bu durum, hatalı pozitif tespitlere neden olmaktadır.
- Hareketli arka plan: Sahnenin bazı bölümleri hareket içeriyor olabilir fakat bu bölümler arka plan olarak değerlendirilmelidir. Bu tür hareketler periyodik veya düzensiz olabilirler (örneğin, sallanan ağaç dalları, trafik ışıkları vs.).
- Kamufraj: İsteyerek veya değil, bazı nesneler görünüm olarak arka plana benzeyebilirler. Bu durum doğru sınıflandırmayı zora sokmaktadır ve gözetleme sistemleri için özellikle önemlidir.
- Gölgeler: Ön plan nesneleri tarafından oluşturulan gölgeler, arka plan çıkarımı ardından gelen süreçleri karmaşıktırmaktadır. Birbiri üzerine gelen gölgeler, örneğin, farklı nesnelerin sınıflandırılıp birbirinden ayırt edilmesini zorlaştırır. Bu yüzden, bu tür bölgelerin düzeltilmesi tercih edilir.
- Önyükleme (Bootstrapping): Eğer data için ön plan nesnelerinden bağımsız başlangıç bilgisi mevcut değilse, arka plan modeli bir önyükleme stratejisi ile başlatılmalıdır.
- Video gürültüsü: Video sinyaline genelde gürültü eklenmektedir. Video gözetleme uygulamaları için arka plan çıkarımı yaklaşımları sensör gürültüsü veya sıkıştırma etkileri gibi etkenlerle bozulmaya uğramış sinyallerle başa çıkabilmelidir.

2.1.1. Arka Plan Modellemesi

Arka plan çıkarımı, video dizilerinde hareketli nesneleri tespit etmek için basit bir yaklaşımdır. Temel fikir, mevcut kare ile arka plan görüntüsünün farkını almak ve her pikseli bir eşik değerine göre arka plan veya ön plan olarak sınıflandırmak üzerine kuruludur. Bağlantılı bileşen analizi ile izlenen morfolojik işlemler uygulanarak, görüntüdeki tüm aktif bölgeler hesaplanır. Pratikte, bazı zorluklar mevcuttur: arka plan görüntüsünün kamera hareketleri yüzünden gürültü ile bozulması ve salınan nesneler, ışık değişimleri, bulutlar, gölgeler gibi. Bu zorlukları aşmak için literatürde bazı yöntemler önerilmiştir [35]. Bouwmans [5] ve [15]'te, literatürde bulunan yaklaşımların oldukça geniş bir incelemesini sunmuştur. Bouwmans [15]'te arka plan modelleme yaklaşımlarını istatistiksel modelleme [13], yapay sinir ağları ile modelleme [36], arka plan kestirimi [37], bulanık mantık ile

modelleme [21] şeklinde sınıflandırmıştır. Her bir sınıf içinde, farklı birçok yöntem mevcuttur.

2.1.1.1. İstatistiksel Arka Plan Modeli

Arka planı modellemek için literatürde önerilen en sık kullanılan yaklaşım, piksel yoğunluklarının zamanla değişen değerlerinin bir Gauss Dağılımı ile modellenmesidir. Wren ve arkadaşları [38]'de tek bir Gauss dağılımı ile modellemeyi önermiştir, fakat bu yaklaşım sallanan ağaçlar, dalgalanan su gibi arka plan hareketleri ile başa çıkamamaktadır. Bu sorunu çözmek için, Gauss Karışımı (Mixture of Gaussian, MOG), modeli Stauffer ve Grimson tarafından önerilmiştir [13]. Bunun dışında, Destek Vektör Makinesi (Support Vector Machine, SVM) [39] gibi daha özelleşmiş istatistiksel yöntemler de önerilmiştir. Bu yöntemlerde hafıza gereksinimi, parametrik olmayan yöntemlere göre daha azdır.

[13]'te, arka plan modelini inşa etmek için ve hareketli nesneleri tespit etmek için K adet Gauss karışımı (K 3 ile 5 arasında) kullanılmıştır. Genelde, ilk Gauss bileşeni kalıcı pikselleri gösterir ve arka planı temsil eder, ikinci bileşen ağaç dallarının salınımı gibi tekrarlayan varyasyonları ve göreceli olarak statik bölgeleri işaret eder, üçüncü bileşen ise ani değişimleri temsil eder. t anında, bir X pikseli için olasılık ifadesi (2.1) numaralı denklemdeki gibidir:

$$P(X_t) = \sum_{i=1}^K w_{i,t} * \eta(X_t, \mu_{i,t}, \Sigma_{i,t}) \quad (2.1)$$

K, Gauss modlarının sayısı, $w_{i,t}$ i'nci Gauss dağılımının t anındaki güncel bir ağırlık kestirimi, $\mu_{i,t}$ i'nci Gauss dağılımının ortalaması, $\Sigma_{i,t}$ kovaryans matrisi ve η ise (2.2) numaralı denklem ile verilen Gauss olasılık yoğunluk fonksiyonudur

$$\eta(X_t, \mu_t, \Sigma_t) = \frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} e^{-\frac{1}{2}(X_t - \mu_t)^T \Sigma^{-1} (X_t - \mu_t)} \quad (2.2)$$

n, X pikselindeki yoğunluğun (intensity) boyutu ve

$$w_{k,t} = (1 - \alpha)w_{k,t-1} + \alpha(M_{k,t}) \quad (2.3)$$

α öğrenme hızıdır ve $[0-1]$ arasında değer almaktadır ve $M_{k,t}$ ise eşleşen model için 1, diğer modeller için 0'dır. RGB renk uzayı için, Kırmızı, yeşil ve mavi piksellerin birbirinden bağımsız ve aynı varyansa sahip oldukları, I birim matris olmak üzere, $\Sigma_{k,t} = \sigma^2_{k,t}I$ varsayılmıştır. Gauss modları, w/σ değerine göre büyükten küçüğe sıralandıktan sonra, ilk b adet dağılım arka planı modellemek için seçilir ve B, arka plan olmak üzere eşitlik (2.4) ile belirlenir

$$B = \arg \min_b (\sum_{k=1}^b w_k > T) \quad (2.4)$$

T, arka plan modelinin minimum parçasıdır. μ ve σ parametreleri, eşleşmeyen dağılımlar için aynı kalırken, yeni gözlemlerle eşleşen dağılımlara ait olanlar (2.5)-(2.7) denklemlerine göre güncellenir:

$$\mu_t = (1 - \rho)\mu_{t-1} + \rho X_t \quad (2.5)$$

$$\sigma^2_t = (1 - \rho)\sigma^2_{t-1} + \rho(X_t - \mu_t)^T(X_t - \mu_t) \quad (2.6)$$

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

Bu yöntemin en belirgin avantajlarından biri, bir nesne arka plan modeline dâhil olduğunda, var olan arka plan modeli yok edilmez. Arka plandaki orijinal renk K'nci en yüksek olasılıklı olana ve yeni bir renk gözlenene kadar karışımında kalır. Bu yüzden, eğer bir nesne arka planın bir parçası olacak kadar uzun süre durağan kalır ve sonra hareket ederse, önceki arka planı tasvir eden dağılım aynı ortalama ve varyans, fakat daha küçük bir w, ağırlık ile hala mevcuttur ve çabukça arka plana tekrar uyum sağlayacaktır.

Gauss karışımı yöntemi; yavaş ışık değişimlerine, karışık arka plandaki periyodik hareketlere, yavaş hareket eden nesnelere, uzun süreli sahne değişimlerine ve kamera gürültülerine karşı dirençlidir. Fakat hızlı ışık değişimlerine uyum sağlayamaz ve gölge etkilerinin üstesinden iyi bir şekilde gelemmez [40].

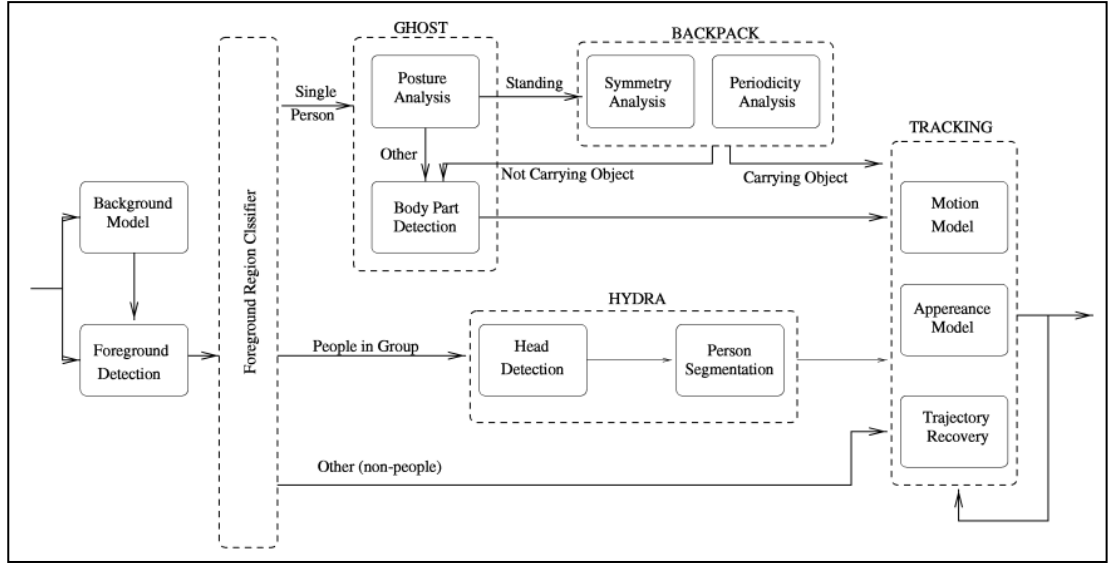
2.1.1.2. W⁴ Sistemi

[31]'de, W⁴ isimli, birden çok insanı tespit ve takip eden, ayrıca hareketlerini izleyen gerçek zamanlı bir video gözetleme sistemi geliştirilmiştir. Bu yöntem, gri seviyeli video görüntüleri veya kızılötesi kamera görüntülerini işlemektedir. W⁴, ayrıca, insanların paket, çanta vb. nesneler taşıyıp taşımadıklarına karar vermekte ve nesneleri silüetlerinden bölütleyebilmektedir. Ardından, bu nesnelerin diğer karelerde de belirlenebilmesi için bir görünüş modeli oluşturmaktadır. W⁴, insanlar ve diğer nesneler arasındaki etkileşimleri; bir nesneyi bırakma, çanta değiş tokuşu veya bir nesneyi bulunduğu yerden alma gibi, tanıyabilmektedir.

Sistemin önemli özellikleri şunlardır:

- İstatistiksel olarak arka planı tamamen durağan olmasa bile öğrenir ve modeller.
- İnsanları, şekil ve periyodik hareket bilgilerini kullanarak diğer nesnelerden ayırt eder.
- Birden çok insanı, birlikte hareket etseler veya birbirleriyle etkileşim halinde olsalar dahi, aynı anda takip edebilir.
- Perdelenmenin ardından tekrar tanımak için her insanın görünüş modelini oluşturur.
- Her insan için, istatistiksel şekil modeli ile 6 temel vücut bölümünü tespit edip takibini yapar.
- Bir kişinin, nesne taşıyıp taşımadığına karar verir ve nesne bölütlenerek değiş tokuş gibi durumlarda takip edebilir.

Şekil 2.4'te verilen blok şema, W⁴ sistem mimarisini göstermektedir [34]. İlk aşamada (tespit), istatistiksel arka plan modeli kullanılarak ön plan pikselleri tespit edilir; ön plan nesnelerine (blob) gruplanırlar ve her gruba ait bölgesel ve global öznitelikler hesaplanır. İkinci aşamada (silüet analizi), her grup, statik şekil ve dinamik periyodiklik analizi kullanılarak önceden belirlenmiş üç sınıftan birine sınıflanır: tek şahıs, grup halindeki insanlar ve diğer nesneler.



Şekil 2.4: W4 Sisteminin Blok Şeması.

W^4 sisteminde kullanılan siluet bazlı tekniklerin genel performansı, ön plan gruplarının ne kadar doğru olarak tespit edildiğine bağlıdır. Eğer ön plan tespit modülü insanları bölütlemeye başarısız olursa, kullanılan siluet bazlı yöntemler açıkça aksayacaktır. Açık hava sahnelerindeki birçok problemin üstesinden gelebilen hızlı bir arka plan çıkarımı algoritması geliştirilmiştir. Diğer yandan, monokromatik görüntülerin kullanıldığı bir yöntem olması nedeniyle, sahne aydınlanmasındaki köklü değişiklikler, büyük tespit hatalarına neden olabilmektedir. Yine de, [13] ve [41]'deki gibi yavaş fakat daha gürbüz arka plan çıkarımı algoritmaları ile değiştirilebilir.

W^4 , bir eğitim süreci boyunca alınan arka plan sıra istatistiğinden oluşturulan bir iki-doruklu dağılım olan arka plan varyasyon modeli kullanır. Arka plan sahnesi, eğitim süresince gözlenen ardışık karelerdeki, her pikselin maksimum yoğunluk farkı $d(x)$, minimum değeri $m(x)$, maksimum değeri $n(x)$ olarak üç değerle modellenir.

Yöntem, arka plan modelini görüş açısı içerisinde yürüyen insanlar, arabalar gibi hareketli ön plan nesneleri olsa dahi öğrenebilmektedir. Yöntem, hareket eden pikselleri arka plan modeli hesabından dışlamaya dayalı iki aşamalı bir metot kullanmaktadır. İlk aşamada, piksel tabanlı zaman üzerinden bir medyan filtresi, giriş videosunun birkaç saniyelik bölümüne (genellikle 20-40 saniye), hareket eden pikselleri durağan piksellerden ayırt etmek için uygulanır. İkinci aşamada, yalnızca bulunan durağan pikseller arka plan başlangıç modelini oluşturmak için işlenir. V , ardışık N video karesinden oluşan bir dizi ve $V^i(x)$, V dizisindeki i 'nci görüntünün x

konumundaki pikselin yoğunluğu olsun. $\sigma(x)$ ve $\lambda(x)$, sırasıyla, V 'deki tüm görüntülerin, x konumundaki piksel yoğunluklarının standart sapması ve medyan değerleridir. x konumundaki bir pikselin arka plan arka plan başlangıç modeli, $[m(x), n(x), d(x)]$, aşağıdaki gibi elde edilir:

$$\begin{bmatrix} m(x) \\ n(x) \\ d(x) \end{bmatrix} = \begin{bmatrix} \min_z \{V^z(x)\} \\ \max_z \{V^z(x)\} \\ \max_z \{|V^z(x) - V^{z-1}(x)|\} \end{bmatrix} \quad (2.8)$$

Burada, $V^z(x)$ durağan pikseller olarak sınıflandırılmıştır ve $|V^z(x) - \lambda(x)| < 2 * \sigma(x)$ 'dir.

2.1.1.3. Bulanık Mantık ile Arka Plan Modellemesi

Arka planın istatistiksel modellemesi literatürde sıklıkla kullanılmaktadır. Modelin çok dorukluluğa uygun olması, kademeli ışık değişimleri ile başa çıkabilmesi en büyük avantajlarından. Bunun yanında, arka planda hareket veya kamera titremesi gibi dinamik arka plan olduğu durumlarda bazı kısıtları mevcuttur. Bunun yanında, model bir eğitim süreci ile oluşturulduğundan, eğitimde kullanılan gözlemlerin gürültülü veya yetersiz olması arka planın doğru modellenememesine neden olabilmektedir. Bu durum da, hatalı tespitlere sebep olmaktadır. Bouwmans ve arkadaşları [42]'de arka planın istatistiksel modeli için MOG yaklaşımını geliştirerek, Tip-2 Fuzzy MOG kullanmayı önermektedir. Tip-2 Fuzzy yöntemi, modeli parametrelerin belirsizliğini de işin içine katarak oluşturmaktadır. OTCBVS benchmark veri seti ile önerilen yöntemin performansı gösterilmiştir. Yöntem, dinamik arka planın varlığında, standart yöntemin başa çıkamadığı durumlarla başa çıkabilmektedir.

Yöntem, özellikle kızılötesi videolarda yüksek performans vermektedir. Bu çalışmada aynı yöntemin gri seviyeli optik kamera görüntülerindeki performansı incelenecektir.

Denklem (2.1)'e, Tip-2 Fuzzy teorisini uygulamak için, Gauss dağılımı parametreleri μ ve σ için net değerler yerine, Tip-2 üyelik fonksiyonları ile tanımlanmış belirsiz ortalama (UM, uncertain mean) veya belirsiz varyans (UV,

uncertain variance) kullanılmaktadır. UM ve UV üyelik fonksiyonları aşağıdaki eşitliklerde verilmiştir.

$$\eta(X_t, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma^2} e^{-\frac{1}{2}\left(\frac{X_t - \mu}{\sigma}\right)^2} \quad (2.9)$$

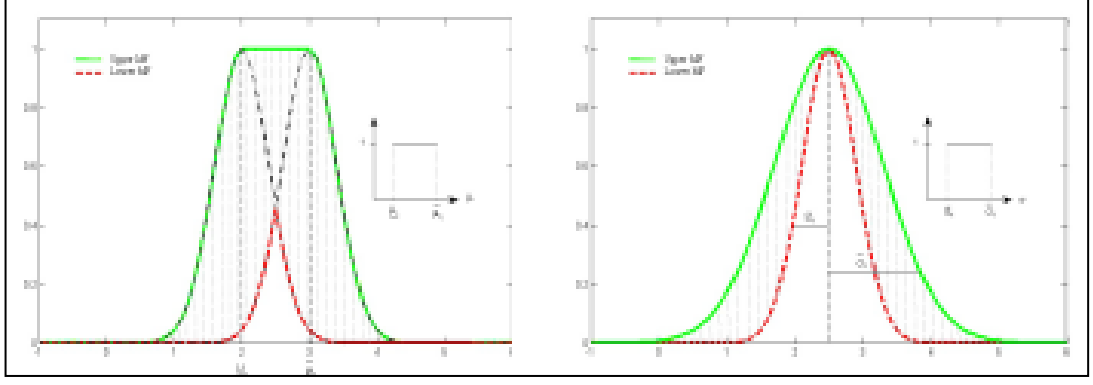
$$\eta(X_t, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma^2} e^{-\frac{1}{2}\left(\frac{X_t - \mu}{\sigma}\right)^2} \quad (2.10)$$

Eşitlik (2.9) ve (2.10) UM ve UV üyelik fonksiyonlarıdır ve $\mu \in [\mu_L, \mu_U]$ ve $\sigma \in [\sigma_L, \sigma_U]$ aralıklarındadır. μ' ve σ' , belirsiz ortalama ve belirsiz standart sapma parametrelerini göstermektedir. Parametrelerin belirsizliği ile ilgili bir ön bilgi olmadığı için, düzgün dağılımlı aralıklar arasında değer almaktadırlar. Bu aralıklar arasında değer alan belirsiz ortalama ve standart sapmaya ait birincil üyelik fonksiyonları Şekil 2.5'de gösterilmiştir. Her iki grafikte, üst üyelik fonksiyonu yeşil çizgi, alt üyelik fonksiyonu kırmızı çizgi ile gösterilmiştir. Yeşil ve kırmızı çizgilerin arasında kalan bölge ise, belirsizlik ayak izi (FOU, footprint of uncertainty) adını almaktadır.

Üst ve alt üyelik fonksiyonları, sırasıyla, belirsiz ortalama için eşitlik (2.11) ve (2.12)'de verilmiştir.

$$h_U(X_t) = \begin{cases} \eta(X_t, \mu_L, \sigma), & X_t < \mu_L \\ 1, & \mu_L \leq X_t \leq \mu_U \\ \eta(X_t, \mu_U, \sigma), & X_t > \mu_U \end{cases} \quad (2.11)$$

$$h_L(X_t) = \begin{cases} \eta(X_t, \mu_U, \sigma), & X_t \leq \frac{\mu_U + \mu_L}{2} \\ \eta(X_t, \mu_L, \sigma), & X_t > \frac{\mu_U + \mu_L}{2} \end{cases} \quad (2.12)$$



Şekil 2.5: UM (Solda) ve UV (Sağda) İçin Birincil Üyelik Fonksiyonları.

Belirsiz standart sapma için alt ve üst birincil üyelik fonksiyonları ise $h_L(X_t) = \eta(X_t, \mu, \sigma_L)$ ve $h_U(X_t) = \eta(X_t, \mu, \sigma_U)$ şeklindedir. μ' ve σ' parametrelerin için alt ve üst sınırlar eşitlik (2.13) ve (2.14)'deki gibi k_m ve k_v çarpanları ile belirlenmektedir.

$$\mu_L = \mu - k_m \sigma, \quad \mu_U = \mu + k_m \sigma, \quad k_m \in [0, 3] \quad (2.13)$$

$$\sigma_L = k_v \sigma, \quad \sigma_U = \frac{1}{k_v} \sigma, \quad k_v \in [0.3, 1] \quad (2.14)$$

Modeli oluşturmak için, σ , μ parametrelerini ve k_m veya k_v çarpanının kestirimi gerekmektedir. k_m ve k_v çarpanları sabit sayılar olarak belirlendiğinden, T2-FMOG için parametre kestirimi üç adımdan oluşmaktadır:

- Adım 1: K , 3 ile 5 arasında seçilir.
- Adım 2: MOG parametreleri, bir EM algoritması ile kestirilir.
- Adım 3: k_m veya k_v çarpanı ile T2-FMOG-UM veya T2-FMOG-UV oluşturulur.

Eğitim aşamasının ardından, arka plan dağılımı, eşitlik (2.4) ile belirlenir.

Yeni bir kare geldiğinde, ön plan tespiti için her piksel için eşleştirme testi yapılmaktadır. Bu test için, alt ve üst birincil üyelik fonksiyonlarının log-likelihood değerleri arasındaki mutlak mesafe, yani $H(X_t) = |\ln(h_L(X_t)) - \ln(h_U(X_t))|$ kullanılmaktadır. Ardından herhangi bir piksel, eşitlik (2.15)'e göre bir Gauss dağılımına atanmaktadır.

$$H(X_t) < k\sigma \quad (2.15)$$

k , sabit eşik değeridir ve deneysel olarak 2.5 şeklinde belirlenmiştir [42]. Eşleştirme testi ile iki durum ortaya çıkmaktadır: (1) piksel, K adet Gauss dağılımından biri ile eşleşmiştir. Bu durumda, eğer eşleşen dağılım eğer arka plan dağılımı ise, piksel arka plana ait denir. (2) K adet Gauss dağılımından hiçbirisi ile eşleşme olmamıştır, bu durumda piksel, ön plan ait olarak sınıflanır. Bu aşamada, ikili bir maske elde edilir. Bu maskede, ön plana ait pikseller 1, arka plan ise 0 değerini almaktadır. Ardından, yeni ön plan tespiti için parametreler [13]'teki gibi güncellenir.

2.1.1.4. Altuzay Arka Plan Modeli

Verilen bir gözetleme videosuna ait video görüntü dizisinden, genellikle arka plandan ayrılan aktivitelerin belirlenmesi gerekmektedir. Video kareleri bir matrisin sütun vektörleri halinde biriktirildiğinde, matrisin düşük mertebe (*low rank*) bileşeni doğal olarak arka plana karşılık gelmektedir. Seyrek (*sparse*) bileşeni ise ön plandaki hareketli nesneleri göstermektedir [43]. Kareler arasındaki korelasyon yüzünden, videolar düşük mertebe modellemesi için doğal problemlerdir. En temel algoritmik görevlerden biri sahnedeki arka plan değişimleri için iyi bir modelin kestirimidir. Fakat ön plan nesnelerinin varlığı sebebiyle bu karmaşık bir işlemdir. Bu tür durumlarda arka plan varyasyonlarını yaklaşık olarak düşük mertebe ile modellemek gayet doğal olacaktır. Ön plan nesneleri, örneğin arabalar, insanlar vs., genellikle görüntünün yalnızca bir kısmında mevcuttur ve bu yüzden seyrek hatalar (*errors*, *outliers*) olarak varsayılabilir.

Arka plan modelleme, video kareleri arasındaki ilişkiyi ortaya çıkarmada, arka plan değişimlerini ve hareketli ön plan nesnelerini modellemede, zorlu bir görevdir. Bir video dizisinde, tüm kareler arasında arka plan bölgeleri birbiri ile ilişkili iken, hareketli nesnelere ait pikseller seyrek ve bağımsız olduğu için *düşük mertebe + seyrek* yapısını sağlamaktadır.

Altuzay öğrenmesinde, düşük mertebe ve seyrek yapı ayrıştırması, matris tamamlama ve sıkıştırılmış algılama konularında çalışılmıştır. Zhou ve arkadaşları [27]'de bir data matrisini düşük mertebe ve seyrek ($X = L + S$) yapısına etkin bir şekilde ayırmak için GoDec (Go Decomposition) ismini verdikleri bir yaklaşım

önermişlerdir. Algoritma, sıralı olarak düşük mertebe yaklaşıklığını $X - S$ 'e ve seyrek yaklaşıklığını da $X - L$ 'ye atamaktadır. Algoritmanın ismi, bu sıralı yapısından dolayı “Go” oyunundan esinlenilerek verilmiştir. Algoritma, ikili rastgele izdüşüm (BRP) ile kayda değer miktarda hızlandırılmaktadır. Zhou tarafından yapılan [27]'deki çalışmada, amaç fonksiyonunun $\|X - L - S\|_F^2$ yerel minimuma yakınsadığı ispatlanmıştır.

Bir matris, sütun uzayı üzerine rastgele izdüşümler ile etkili bir şekilde yaklaşıkklanabilir. $m \times n$ boyutlu yoğun bir X matrisinin r adet BRP'si verilmiş olsun. Bu durumda X matrisinin hızlı bir rank- r yaklaşıklığı eşitlik (2.16)'daki gibidir.

$$L = Y_1(A_2^T Y_1)^{-1} Y_2^T \quad (2.16)$$

Burada A_1 ve A_2 bağımsız Gauss rastgele matrisler olmak üzere $Y_1 = X A_1$ ve $Y_2 = A_2 X^T$, X matrisinin izdüşümleridir. Bu yaklaşım, hesaplama yükü olarak SVD (Singular Value Decomposition)'den çok daha az maliyetlidir.

GoDec, bir iterasyonda sırayla $X - S$ ifadesinin rank- r yaklaşıklığını L 'ye k kardinalliğe sahip $X - L$ ifadesinin seyrek (sparse) yaklaşıklığını S 'ye atamaktadır. L 'nin güncellenmesi $X - S$ 'ye SVT (Singular Value Hard Thresholding) uygulanarak; S 'nin güncellenmesi ise $X - L$ 'ye *entry-wise hard thresholding* uygulanarak gerçekleştirilmektedir. BRP ile düşük mertebe yaklaşıklığı, $X - S$ 'nin rank- r yaklaşıklığını hızlandırmak için kullanılmaktadır.

Hem GoDec hem de RPCA bir X matrisinin düşük mertebe ve seyrek yapısını ayrıştırabilmektedir. Fakat içyapı olarak iki algoritma farklılık göstermektedir. RPCA, $X = L + S$ varsayımını yaparak, X matrisini tam olarak L ve S bileşenlerine ayırmaya çalışır. GoDec ise, X matrisinin G gürültüsünden dolayı tam olarak L ve S bileşenlerine ayıramayacağını varsayarak bunun yerine yaklaşık olarak bu ayrımı yapmaktadır [27].

(2.16)'da L 'nin yaklaşıklık hassasiyetini iyileştirmek için sağ rastgele izdüşüm Y_1 daha iyi bir sol izdüşüm matrisi A_2 , Y_2 de daha iyi A_1 oluşturmak için kullanılmaktadır. Yani, $Y_1 = X A_1$ ardından $A_2 = Y_1$ ile güncellenmekte ve sol rastgele izdüşüm $Y_2 = X^T A_2$ ile hesaplanmakta sonrasında $A_1 = Y_2$ güncellenmektedir. Yeni Y_1 ve Y_2 (2.16)'ya uygulandığında, iyileştirilmiş L yaklaşıklığı BRP ile elde edilmektedir.

X data matrisinin tekil deęerleri yavař azaldığında, (2.16) ifadesi kötü sonuç verebilmektedir. Bu yüzden Zhou ve arkadaşları [27]’de bu durum için [44]’te önerilen bir kuvvet modifikasyonu kullanmıştır. Bu kuvvet modifikasyonunda, X yerine tekil deęerleri daha hızlı azalan $\tilde{X} = (XX^T)^q X$ matrisinin yaklaşıklığı hesaplanmaktadır. Hem X hem de $\tilde{X}$ aynı tekil vektörlere sahiptir. Böylece, $\tilde{X}$ matrisinin BRP’si

$$Y_1 = \tilde{X}A_1, Y_2 = \tilde{X}^T A_2 \quad (2.17)$$

olur. X matrisinin rank- r yaklaşıklığını elde etmek için Y_1 ve Y_2 ’nin QR ayrıştırması hesaplanmaktadır.

$$Y_1 = Q_1 R_1, Y_2 = Q_2 R_2 \quad (2.18)$$

ardından, X matrisinin düşük mertebe yaklaşıklığı eşitlik (2.19)’da verilmiştir:

$$L = (\tilde{L})^{\frac{1}{2q+1}} = Q_1 [R_1 (A_2^T Y_1)^{-1} R_2^T]^{\frac{1}{2q+1}} Q_2^T \quad (2.19)$$

Eşitlik (2.19)’daki güç modifikasyonu bir $r \times r$ matris tersi, bir $r \times r$ matris için SVD hesaplama ve beř adet matris çarpımı gerektirmektedir.

Zhou ve arkadaşları [45]’te BRP ve kuvvet modifikasyonunun yaklaşıklık hatasının deterministik sınırın, ortalama sınır ve sapma sınırının hafif şartlar altında SVD’nin hatasına yaklaştığını göstermiştir.

$$X = L + S, \text{rank}(L) \leq r, \text{card}(S) \leq k \quad (2.20)$$

Eşitlik (2.20) ile verilen yaklaşık “düşük mertebe + seyrek” ayrıştırma problemi, ayrıştırma hatasını minimize ederek řu řekilde çözülebilir:

$$\begin{aligned} \min_{L,S} \|X - L - S\|_F^2 \\ \text{s.t. } \text{rank}(L) \leq r, \text{card}(S) \leq k \end{aligned} \quad (2.21)$$

Zhou ve arkadaşları [27]’de önce naive GoDec algoritmasını önermişlerdir. Buna göre, eşitlik (2.21) ile verilen optimizasyon probleminin çözümü, ařağıdaki iki alt optimizasyon probleminin yakınsama olana dek sıralı çözümlesiyle elde edilebilir:

$$\begin{cases} L_t = \arg \min_{\text{rank}(L) \leq r} \|X - L - S_{t-1}\|_F^2 \\ S_t = \arg \min_{\text{card}(S) \leq k} \|X - L_{t-1} - S\|_F^2 \end{cases} \quad (2.22)$$

(2.22)'deki her iki alt problem de konveks olmayan kısıtlara sahip olsa da, global çözümleri L_t ve S_t mevcuttur.

Özel olarak, (2.22)'deki iki problem, L_t 'yi $X - S_{t-1}$ 'in tekil değer katı eşiklemesi ile güncelleyerek ve S_t 'yi de $X - L_{t-1}$ 'in girdiye dayalı (entry-wise) katı eşiklemesi ile güncelleyerek çözülebilir. Naive GoDec algoritmasındaki en büyük hesaplama yükü L_t 'nin güncellemesindeki $X - S_{t-1}$ 'in tekil değer ayrıştırmasıdır. Özellikle X büyük boyutlarda olduğunda oldukça kullanışsızdır. Bu yüzden, BRP temelli yaklaşıklık kullanılarak hızlı GoDec önerilmiştir.

BRP temelli düşük mertebe yaklaşıklaması etkili ve optimuma yakın olduğu için, [27]'de önerilen naive GoDec algoritmasında SVD yerine BRP kullanılmaktadır. Bu sayede hesaplama yükü hatırı sayılır derecede azaltılmaktadır. Aynı makalede, önerilen GoDec algoritmasının yakınsama özellikleri de ispatlanmıştır.

GoDec algoritması kullanılarak [27]'de elde edilen farklı videolar için arka plan modelleme ve ön plan tespiti sonuçları verilmiştir. Arka plan ve hareketli nesneler net bir şekilde birbirinden ayrılmıştır. Bu işlemde hiçbir detay kaybedilmemiştir. Bu durum, daha ilerideki proses adımlarının performansını da olumlu yönde etkileyecektir.

2.1.2. Arka Plan Güncelleme

Arka plan güncelleme, uzun vadede arka plan modelinin sabit kalamayacağı, bazı değişimlere (kalıcı ışık değişimleri, sahne dışından bazı nesnelerin sahne içinde durması vb.) maruz kalacağı için modelin bu değişimlere uyumlaştırılması anlamına gelmektedir. Genel olarak, arka plan modelini, zaman içinde sahnede meydana gelen kalıcı değişimlere adapte etme işlemine 'arka plan bakımı' denmektedir. Arka plan bakımı ile ilgili üç durum vardır [46]:

- **Bakım Kuralları:** arka planı uyumlaştırmak için kullanılan kurallardır. Bunlar tüm pikselleri aynı kurala göre güncelleme (kör – blind) bakım, pikselleri önceki durumlarını da göz önünde bulundurarak inceleme (seçici bakım) ve pikselleri

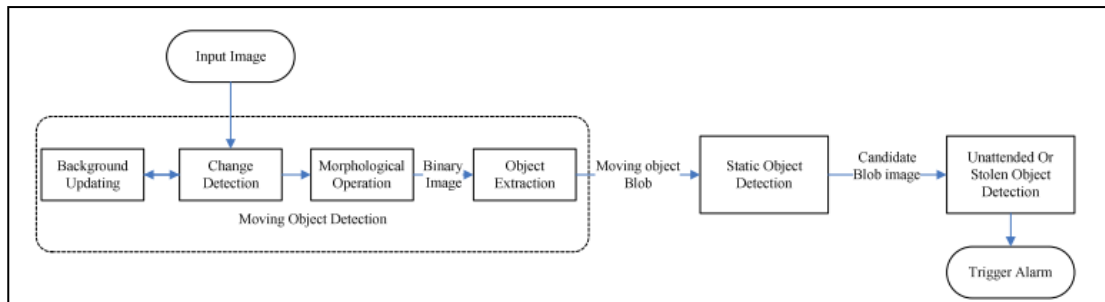
arka plan veya ön plan üyelik derecelerine göre güncellemeye (fuzzy maintenance) dayalıdır.

- Öğrenme Oranı: sahnedeki değişimlere adaptasyon hızını belirler. Sabit öğrenme hızı, piksellerin istatistiksel özelliklerine dayanan istatistikî öğrenme hızı ve fuzzy öğrenme hızı olarak sayılabilir.
- Adaptasyon Oranı: bakımın ne sıklıkla yapılacağını belirler. Öğrenme oranı gibi sabit, istatistikî ve fuzzy olarak seçilebilir.

2.1.2.1. İkili Arka Plan Güncelleme Modeli

Kamu alanlarında terk edilmiş nesneler, sahnede statik (hareketsiz) bulunduklarından, hareketli nesne olarak algılanamazlar ve arka plana dâhil olmalıdırlar. Öte yandan, orijinal sahne arka planına ait olmadıklarından göz ardı edilmemelidirler.

Wang ve arkadaşları [12]'de, şüpheli ve kayıp nesne tespiti için tek kameralı, izleme olmayan (non-tracking), iki arka plan modeli kullanan bir yöntem önermiştir. İki arka planın modeli, başlangıç arka planı farklı hızlarda güncellenerek oluşturulmaktadır. Bu arka plan modellerinden bir tanesi “mevcut arka plan modeli” (BC), diğeri “orijinal arka plan modeli” (BO) olarak adlandırılmıştır. İki arka plan modeli kullanmanın bir avantajı, kullanıcının referans arka plan kareleri arasındaki zaman aralığını farklı ihtiyaçlara ve ortamlara göre ayarlayabilmesidir, dahası, iki arka plan modeli de ortamın değişim hızına göre dinamik olarak; ilki sıklıkla, ikincisi ise daha yavaş bir hızda güncellenmektedir.



Şekil 2.6: Wang ve arkadaşları Tarafından Önerilen Şüpheli Nesne Sistemi Blok Şeması.

Giriş videosunun ilk karesi BC ve BO için başlangıç olarak seçilir. Daha sonra uyarlamalı bir arka plan güncelleme yöntemi uygulanır. t anındaki güncelleme kuralları sırasıyla mevcut arka plan modeli (BC) ve orijinal arka plan modelindeki (BO), (x,y) konumundaki piksel için, (2.23) ve (2.24) numaralı eşitliklerle verilmiştir.

$$BC_t(x,y) = \begin{cases} BC_{t-1}(x,y) & MC_t(x,y) = 1 \wedge StayCount(x,y) < T_0 \\ I_t(x,y) & MC_t(x,y) = 1 \wedge StayCount(x,y) > T_0 \\ (1 - \alpha)BC_{t-1}(x,y) + \alpha I_t(x,y) & MC_t(x,y) = 0 \end{cases} \quad (2.23)$$

$$BC_t(x,y) = \begin{cases} BO_{t-1}(x,y) & MO_t(x,y) = 1 \wedge StayCount(x,y) < T_1 \\ BC_t(x,y) & MO_t(x,y) = 1 \wedge StayCount(x,y) > T_1 \\ (1 - \beta)BO_{t-1}(x,y) + \beta I_t(x,y) & MO_t(x,y) = 0 \end{cases} \quad (2.24)$$

Denklemlerde, $I_t(x,y)$ t anında, mevcut karede (x,y) konumundaki piksel değeridir. $MC_t(x,y)$ ve $MO_t(x,y)$, 0 veya 1 değerini alarak, sırasıyla mevcut ve orijinal arka plan için (x,y) konumundaki pikselin değişip değişmediğini göstermektedir. α ve β arka planların öğrenme hızlarıdır ve $[0-1]$ arasında değer almaktadır. $StayCount(x,y)$, o konumda bulunan pikselin ardışık kaç karede ön plan pikseli olarak tespit edildiğini göstermektedir. Bu değer, T_0 eşik değerinden büyük olduğunda, mevcut arka plan güncellenmektedir, benzer şekilde, T_1 eşik değerinden büyük olduğunda da orijinal arka plan modeli güncellenmektedir. T_0 ve T_1 , kare sayısını gösteren pozitif sayılardır.

Güncelleme kurallarına göre, ön plan hızlı bir şekilde değişse bile, bu arka planı etkilemeyecektir. Fakat ön plan durağansa, zamanla arka plana karışacaktır. Bu yolla, arka planın mantıksal olarak arka plana ait olmayan pikseller tarafından kirletilmesinin önüne geçilmektedir. Bunun yanında, aşağıdaki yorumlar yapılabilir: (FC, mevcut ön plan maskesi, FO orijinal ön plan maskesini göstermektedir)

- $FC_t(x,y) = 1$ ve $FO_t(x,y) = 1$ ise, (x,y) pikseli hiçbir arka plana uymayan bir pikseldir ve sahneye yeni girmiş hareketli nesneyi işaret eder.
- $FC_t(x,y) = 1$ ve $FO_t(x,y) = 0$ ise, önünde nesne bulunmuş olan fakat tekrar orijinal değerine dönen arka plan pikselini işaret etmektedir.

- $FC_t(x,y) = 0$ ve $FO_t(x,y) = 0$ ise, ilgili piksel her iki arka plan modeline de uymakta, yani sahnede hiçbir değişim yok manasına gelmektedir. Bu durumda, yalnızca arka plan güncelleme yapılır.

Özetle, iki arka plan modeli kullanmanın temel avantajı, nesnenin doğru olarak bölütlenebilmesi ve düşük hesaplama yükü ile gerçekleştirilebilmesidir. Dahası, mevcut arka plan ani değişimlere karşı dirençlidir ve sahnedeki değişimlere daha hızlı uyum sağlamaktadır. Tersine, orijinal arka plan mevcut arka plana göre daha karardır. Dolayısıyla, statik nesne tespiti $FC_t(x,y)$ ve $FO_t(x,y)$ arasındaki fark gözlenerek kolayca gerçekleştirilebilir.

$FC_t(x,y) = 1$ ve $FO_t(x,y) = 0$ durumu, tespit için büyük önem taşımaktadır. Bu şart altında, bir (x,y) pikseli, mevcut arka plana çoktan karışmış, fakat orijinal arka plana karışacak kadar uzun süre geçmemiş bir statik nesneye işaret edebilir. Aşağıda, $SO_t(x,y)$ ile gösterilen statik nesne görüntüsünün nasıl hesaplanacağı formülize edilmiştir:

$$SO_t(x,y) = \begin{cases} SO_t(x,y) + 1, & FC_t(x,y) = 0 \wedge FO_t(x,y) = 1 \\ SO_t(x,y) - 1, & FC_t(x,y) \neq 0 \wedge FO_t(x,y) \neq 1 \\ 255, & T < SO_t(x,y) \\ 0, & SO_t(x,y) < 0 \end{cases} \quad (2.25)$$

$SO_t(x,y)$ görüntüsü, tespit sürecinde sahnedeki gürültü ve ani değişimlerin neden olduğu etkiyi azaltabilmektedir. Ayrıca, bir statik pikselin, tespit edilecek aday hedefe atanması için gerekli minimum süreyi de belirlemektedir. Aynı prensipte, $StayCount(x,y)$ ile her piksel için durum istatistik bilgisi toplanır ve T 'den büyük olduğunda, pikselin hedef statik nesneye ait olduğu doğrulanır. T eşik değeri, kare sayısını göstermektedir ve sistemin gürültü karakteristiği ve cevap verebilme yeteneğine göre değişmelidir. Yanlış tespitleri azaltma amacıyla, büyük T değerleri tercih edilir. Bununla birlikte, hedef kaçırmamak için T değeri, T_1 'den büyük olmalıdır.

[12]'deki video gözetleme sisteminde, en önemli uygulamalardan biri terk edilmiş nesneyi, kayıp nesneden ayırt etmektir. Uygulanan yaklaşım, 3 grup pikselden yola çıkmaktadır:

- Durgun (statik) hedef nesne görüntüsü, $SR_t(x,y)$, ilgilenilen bölgeyi (ROI) gösteren sınırlayan kutu ile belirlenmiş şüpheli veya kayıp nesneyi göstermektedir. Tamamen, iki arka plan farkı ile sağlanan bilginin kullanımı ile oluşturulmaktadır.
- Statik hedefin sınırlayan kutu ile kapsanan pikseller ile oluşturulan mevcut arka plan ROI görüntüsü, $CR_t(x,y)$, ilgili bölgelerde mevcut arka planın değişimini gösterir.
- Orijinal arka plan ROI görüntüsü, $OR_t(x,y)$, de durgun hedefin sınırlayıcı kutusu ile kapsanan piksellerden oluşur ve benzer şekilde ilgili bölgelerde orijinal arka planda meydana gelen değişimleri gösterir.

Yukarıda belirtilen $FC_t(x,y) = 0$ ve $FO_t(x,y) = 1$ durumunda, sahnede durgun bir (statik) nesnenin görüldüğü sonucu çıkarılabilir ve ardından bahsi geçen üç görüntünün analizi yapılır. Gözleme dayanarak, bir şüpheli/terk edilmiş nesne olayında, SR_t görüntüsünün CR_t görüntüsü ile aynı ve OR_t görüntüsünden farklı olduğu açıkça görülmektedir. Bu demek olur ki, şüpheli nesne olayında, SR_t görüntüsü, OR_t görüntüsüne göre, CR_t görüntüsü ile daha fazla renk benzerliğine sahiptir. Bu durumun tam aksine, kayıp/çalıntı nesne olayında ise, SR_t görüntüsü ile OR_t görüntüsü arasındaki renk benzerliği, SR_t görüntüsü ile CR_t görüntüsü arasındaki renk benzerliğinden daha fazladır.

Referans ROI'lerle birlikte, [47]'de önerilen Bhattacharyya mesafe metriği benimsenmiştir. Önce, her üç bölge için de, YCbCr renk uzayında luminans kanalı kullanılarak renk histogramları hesaplanır. SR görüntüsünün histogramı, $p = \{p^{(u)}\}_{u=1,...,m}$; CR görüntüsünün histogramı $q = \{q^{(u)}\}_{u=1,...,m}$; ve OR_t görüntüsünün histogramı da $r = \{r^{(u)}\}_{u=1,...,m}$ 'dir. Ardından, Bhattacharyya katsayısı $\rho_0[p,q] = \sum_{u=1}^m \sqrt{p^{(u)} \cdot q^{(u)}}$ ve $\rho_1[p,r] = \sum_{u=1}^m \sqrt{p^{(u)} \cdot r^{(u)}}$ elde edilir. ρ , [0,1] arasında ve m histogram bin sayısı olmak üzere, Bhattacharyya mesafesi $d_0 = \sqrt{1 - \rho_0[p,q]}$ ve $d_1 = \sqrt{1 - \rho_1[p,r]}$ elde edilir. d_k , iki renk histogramı arasındaki mesafedir. Bu değer ne kadar küçükse, renk histogramları o kadar benzerdir. Hesaplamanın ardından, $d_0 - d_1$ farkı karar vermek için kullanılır. Eğer fark, pozitif bir sayıysa, kayıp/çalıntı bir nesne bulunduğu, aksi halde terk edilmiş/şüpheli bir nesne bulunduğu karar verilir.

Bu yöntem ile hiçbir hareket bilgisi veya takip algoritması kullanmaksızın durağan nesneleri tespit edebilen bir sistem geliştirilmiştir. Statik nesneleri tespit etmek için iki arka plan modeli kullanılmıştır. İkili arka plan modeli sayesinde, çabuk ışık değişimlerine, nesne perdelenmesine karşı dirençli ve aynı zamanda gerçek zamanlı video gözetleme sistemlerine uygun, etkili statik nesne tespiti yapılabilmektedir.

2.1.2.2. Hareketli Ortalama (Running Average)

Arka plan modelleme yaklaşımları arasında, Hareketli Ortalama (Running Average) en basit ve hızlı yöntemlerden biridir [10]. Hareketli Ortalama, arka plan başlatıldıktan sonra, her yeni karede aşağıdaki kurala göre güncellenir:

$$B_t = \alpha B_{t-1} + (1 - \alpha)I_t \quad (2.26)$$

Eşiklik (2.26)'da, α , öğrenme hızı $[0,1]$ aralığında olmalıdır. Sinyal ve sistem bakış açısından, eşitlik (2.26) bir sonsuz dürtü cevaplı (IIR) filtredir. Bu yüzden hareketli ortalama, IIR bir sistemdir. Hareketli ortalama, eğer arka plan modeli bozulmaya uğrarsa, gerçek arka plan uzun bir süre boyunca görünmediği takdirde, tekrar düzeltmek oldukça zor olacaktır. Bu yüzden, arka plan güncellemesi isabetli yapılmalıdır.

Standart arka plan çıkarımında bir pikselin arka plana mı yoksa hareketli nesneye mi ait olduğuna karar vermek için katı sınırlayıcı fonksiyon kullanılmaktadır. Sagari ve arkadaşları [10]'da katı bir fonksiyon yerine, lineer olarak doyuma ulaşan (saturating) fonksiyon kullanmayı önermiştir.

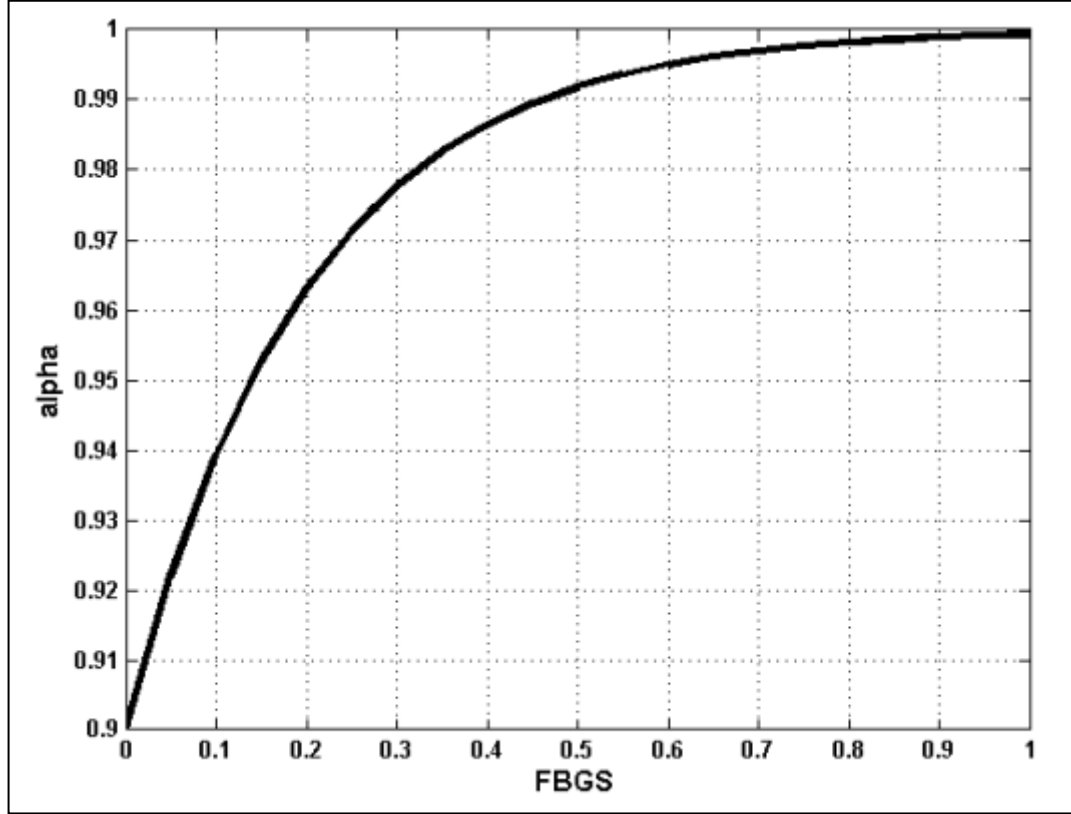
$$FBGS(i, j) = \begin{cases} 1, & \text{if } |I_t(i, j) - B_t(i, j)| > th_s \\ \left| \frac{I_t(i, j) - B_t(i, j)}{th_s} \right|, & \text{otherwise} \end{cases} \quad (2.27)$$

Sonuç, Fuzzy arka plan çıkarımı (FBGS), $[0,1]$ arasında gerçek bir değer alacaktır. I_t , t anındaki kare, B_t ise t anındaki arka plan modeli, th_s eşik değeridir.

Bulanık (Fuzzy) mantık, hareketli ortalama yönteminin performansı artırmak için de kullanılabilir. Fuzzy hareketli ortalama, öğrenme hızı, α , genel bir değer yerine, mevcut arka plan çıkarımına göre her piksel için farklı bir değer almaktadır.

$$\alpha(i, j) = 1 - (1 - \alpha_{min}) * \exp(-5 * FBGS(i, j)) \quad (2.28)$$

Eşitlik (2.28)'de α_{min} öğrenme hızının en küçük değeridir. Eşitlik (2.28) özgün değildir, dolayısıyla farklı uygulamalar için farklı fonksiyonlar kullanılabilir.



Şekil 2.7: $\alpha_{min}= 0.9$ için α - FBGS Diyagramı.

Arka plan güncellemesi, herhangi bir piksel için fuzzy arka plan çıkarımı kullanılarak, eşitlik (2.29)'da verildiği gibi gerçekleştirilebilir.

$$B_t(i, j) = \alpha(i, j)B_{t-1}(i, j) + (1 - \alpha(i, j))I_t(i, j) \quad (2.29)$$

Fuzzy hareketli ortalama ve fuzzy arka plan çıkarımı, klasik yöntemde karşılaşılan arka plan modelinin bozulmasından kaynaklı hatalı tespit oranını düşürmekte ve performansı artırmaktadır. Bunun yanında, klasik hareketli ortalama yöntemine göre, bir miktar daha yavaş çalışmaktadır [10].

2.2. Optik Akış Yöntemleri

Optik akış yöntemleri, sahnedeki hareketli bölgeleri tespit etmek için hareketli nesnelere ait akış vektörlerinin zaman üzerinden hesaplanmasına dayanmaktadır. Bu yaklaşımda kamera hareketliken dahi hareket tespiti yapılabilmektedir. Bunun yanında, birçok optik akış yöntemi hesaplama bakımından karmaşık ve maliyetlidir. Bu yüzden gerçek zamanlı sistemlerde özelleştirilmiş donanım olmadan kullanılamazlar [29].

2.3. Performans Değerlendirme Kriterleri ve Veri Setleri

Durağan nesne tespiti algoritmalarının performanslarını belirlemek ve algoritmaları karşılaştırmak için literatürde kullanılmış farklı yöntemler mevcuttur. Ayrıca, algoritmaların test edilip objektif bir şekilde karşılaştırılabilmesi için herkes tarafından ulaşılabilecek, verilere ihtiyaç duyulmaktadır. Bu ihtiyacı karşılamak için, çeşitli açık (public) veri setleri oluşturulmuştur. Bu bölümde, mevcut açık veri setleri ve performans kriterlerinden bahsedilmiştir.

2.3.1. Veri Setleri

Son yıllarda, video görüntülerinde nesne tespiti algoritmalarının performansını ölçmek ve karşılaştırmak için çeşitli veri setleri oluşturulmuştur. Bu veri setleri konudaki bazı zorlukları ve kesin sonuç (ground truth) görüntülerini vermektedir. Ancak hiçbiri, karşılaşılabilecek zorlukların tamamını bir arada göstermemektedir [15]. Son zamanlarda üretilen veri setleri ise, oldukça geniş kapsamlı gerçekçi videolar ve doğru ground-truth görüntüler ile gerçek dünyadaki mevcut zorlukları dengeli bir biçimde kapsamaktadır. Bu veri setleri, açık olarak ulaşılabilir haldedir. Bu veri setlerine bağlantılar, [48]'ten bulunabilir.

- Wallflower Veri Seti: Bu veri seti, her biri pratikte bir uygulamada karşılaşılabilecek zorlukları (aydınlanma, dinamik arka plan vb) içeren 7 videodan oluşmaktadır. En çok kullanılan veri setlerinden biridir, fakat her bir video için sadece bir ground-truth görüntü bulunmaktadır.

- I2R Veri Seti: Bu set, her biri ışık değişimi veya hareketli arka plan içeren dokuz video görüntüsünden oluşmaktadır. En büyük avantajı, her bir video dizisi için 20 ground-truth görüntü bulunmaktadır. Bu ground-truth görüntüler, kritik bir durum oluştuğunda alınmıştır.
- PETS Veri Setleri: Bu veri setleri, “Performance Evaluation of Tracking and Surveillance” konferansı ile ilgilidir. 2001, 2003 ve 2006 olmak üzere farklı setlerden oluşmaktadır. ground-truth görüntüler, nesneleri çevreleyen kutular şeklinde verilmiştir.
- i-LIDS: Bu veri seti, olay tespiti ve çoklu kamera takibi olarak iki ana kategori içinde farklı senaryolar sunmaktadır. Olay tespiti kategorisinde steril bölge, park edilmiş araç, terk edilmiş nesne, kapı gözetleme gibi senaryolar bulunmaktadır. Bu veri setleri, senaryo kapsamında beklenen tüm sahne yoğunluklarını, hava koşullarını, ışık durumlarını içerecek şekilde oluşturulmuştur.
- BMC 2012 Veri Seti: BMC (Background Models Challenge), arka plan çıkarımı teknikleri konusunda bir çalıştaydır. Bu set, statik bir kameradan alınmış 9 gerçek video içermektedir. Algoritmaların çeşitli koşullardaki güvenilirliğini test etmek için oluşturulmuştur. Dahası, bu veri seti, ön plan tespiti aşamasında karşılaşılan sallanan ağaç dalları, gölgeler, sahnedeki ani ışık değişimleri gibi bazı zorlukların etkisini test etmeye olanak sağlamaktadır.

2.3.2. Performans Ölçme Kriterleri

Bir nesne tespit algoritmasının performansının etkili şekilde ölçülmesi, farklı algoritmalarla objektif olarak karşılaştırma yapabilmek için önemlidir. Bu yüzden literatürde farklı yaklaşımlardan yararlanılmıştır. ROC (Receiver Operating Characteristic) eğrileri, benzerlik ölçüleri, tespit zamanları, precision-recall eğrileri gibi ölçüm yöntemleri mevcuttur.

Porikli ve arkadaşları [49]’da şüpheli nesne tespiti algoritmalarının performansını ölçmek için i-LIDS, PETS, ATC gibi bazı açık veri setleri üzerinde deneyler yapmışlardır. DeneySEL sonuçları, tespit zamanını (doğru ve hatalı tespitler için) ve uyarı vermedeki gecikme zamanını kullanarak değerlendirmişlerdir. Ek olarak, ground-truth olay sayısı ile doğru/hatalı tespit sayılarını da performans ölçüsü olarak kullanmışlardır. Bu sonuçlara göre, oldukça düşük bir yanlış uyarı oranı ile doğru tespit sayısını yüksek tuttuklarını belirtmişlerdir. Önerdikleri yöntemin

hesaplama yükü, piksel bazlı işlemler nedeniyle oldukça düşüktür ve bu sayede, paralel işleme mimarilerine uygundur. GPU üzerinde CUDA programlama ile karşılık gelen CPU implementasyonuna göre görüntüler, 14 kat daha hızlı işlenebildiğini belirtmişlerdir. Örneğin, P4 DualCore 3 GHz bir CPU üzerinde 74 milisaniyede işlenen bir görüntü, CUDA ile 6 milisaniyede işlenebilmektedir [49]. Dolayısıyla önerilen yöntem, gerçek zamanlı sistemlerde etkin olarak kullanılabilir bir yöntemdir.

Li Tian ve arkadaşları [50]'de önerilen yöntemde, arka planı bir Gauss Karışımı ile modellemiştir. Ön plan maskesindeki kaynak bilgisini kullanarak, statik bir ön plan nesnesinin terk edilmiş mi yoksa kayıp nesne mi olduğunu ayırt edebilmektedir. Önerilen yöntem, gerçek zamanlı bir akıllı gözetleme sisteminde kullanılmaktadır ve ayrıca, i-LIDS ve PETS2006 gibi açık veri setleri üzerinde de test edilmiştir. Terk edilmiş nesne tespiti için nicel test sonuçları şüpheli nesne sayısı, doğru tespit (true positive), hatalı tespit (false positive) ve şüpheli nesne olarak tespit edilen durağan insan sayıları üzerinden verilmiştir. Verilen bilgilere göre, hatalı tespit oranı oldukça düşüktür, fakat sahnedeki hareketsiz insanlar şüpheli nesne olarak algılanmaktadır. Ayrıca, algoritma yazarların özel dış mekân veri seti üzerinde de test edilmiştir (Big City on Site Test). Bu veri seti üzerindeki test sonuçlarına göre, 32 terk edilmiş nesne olayı içinde, düşük bir hatalı tespit oranı ile %87.5 tespit oranı elde edilmiştir.

Nascimento ve arkadaşları [35]'te bazı nesne tespiti algoritmalarının performans değerlendirmesini yapmıştır. Değerlendirme için, her algoritma doğru tespit, yanlış tespit ve yanlış alarm oranları ölçüt olarak kullanılmıştır. BBS, temel arka plan çıkarımı yaklaşımıdır. Arka plan ile mevcut görüntü arasındaki mutlak fark, her piksel için eşik değeri ile karşılaştırılır. W^4 , Haritaoğlu ve arkadaşları [31]'de önerdiği algoritmadır. SGM, Wren ve arkadaşları [38]'de önerilen Tek Gauss Modelidir. Dördüncü algoritma, Stauffer ve Grimson'ın Gauss Karışımı Modelidir (MGM) [13]. LOTS, Eckmann ve arkadaşları [51]'de önerdiği yöntemdir.

Tablo 2.1: Beş Nesne Tespit Algoritmasının Performansı.

%	<i>BBS</i>	W^4	<i>SGM</i>	<i>MOG</i>	<i>LOTS</i>
Doğru Tespit	84.3	81.6	86.8	85	91.2
Başarısız Tespit	12.2	9.6	11.5	13.1	8.5
Yanlış Alarm	22.5	8.5	11.3	24.3	0.6
Eşleşen Alan	64.7	50.4	61.9	61.3	78.8

Bunun yanında, Big-O analizi kullanılarak, algoritmaların hesaplama karmaşıklığı da ölçülmüş ve karşılaştırılmıştır. Buna göre, hesaplama bakımından en avantajlı algoritma *BBS* algoritmasıdır. Ardından sırasıyla *LOTS*, W^4 ve *SGM* gelmektedir. Hesaplama karmaşıklığı bakımından en avantajsız algoritma *MGM* olarak belirtilmiştir.

Aggrawal ve arkadaşları [52]’de terk edilmiş nesne olayına ilişkin olayların zamansal akışını bilgisayarla görme yardımıyla incelemiştir. Önerilen yöntem, PETS2006 ve AVSS2007 için i-LIDS veri setleri üzerinde test edilmiştir. Algoritmanın performansı, terk edilmiş nesne olayının başladığı ve uyarının oluşturulduğu zaman açısından *ground-truth* bilgi ile karşılaştırılarak değerlendirilmiştir. Aynı değerlendirme yaklaşımı ile Bayona ve arkadaşları [53]’te, beş farklı arka plan çıkarımı tekniğini incelemiştir. Olayın başlangıç zamanı ve alarm süresi bakımından algoritmalar değerlendirilmiştir. Test için, PETS2006 ve i-LIDS veri setleri kullanılmış, algoritmalar P4 2.8GHz 1GB RAM’e sahip bir CPU üzerinde koşturulmuştur. Ayrıca, algoritmaların *precision-recall* değerleri, ayrı bir performans değerlendirme metriği olarak verilmiştir.

Maddalena ve Petrosino [54] ve [55]’te önerdikleri yöntemin performansını diğer tekniklerle karşılaştırmak için hem işleme hızını hem de tespit hassasiyetini incelemiştir. Nicel sonuçları göstermek için, *precision* (pozitif tahmin) ve *recall* (tespit oranı) değerlerinin yanı sıra, F_1 metriği (*figure-of-merit*, başarımlı ölçüsü) ve benzerlik ölçülerini de kullanmışlardır. F_1 metriği, *precision* ve *recall* değerlerinin ağırlıklı harmonik ortalamasıdır. Bu bahsedilen ölçülerin tamamı [0,1] arasında değer almakta ve 1’e ne kadar yakınsa, o kadar fazla isabet olduğunu söylemektedir.

Tablo 2.2: Maddalena ve Petrosino Tarafından Uygulanan Performans Kriteri.

	Recall	Precision	F-ölçüsü
IR	%86	%90	%89
WT	%98	%97	%98
Seq00	%92	%98	%95

Davis ve arkadaşları [56]’da arka plan ön plan bölütlemesi performansını ölçmek için PDR analizi olarak adlandırılan bir performans ölçüm yöntemi önermişlerdir. Çokça kullanılan ROC analizine göre bazı avantajları vardır. Spesifik olarak, ön plan hedeflerine veya ön plan dağılımı bilgisine ihtiyaç duymamaktadır. Bir arka plan çıkarımı algoritmasının hassaslığını, olası düşük kontrastlı hedefleri arka plana karşı, kontrast bilgisinin bir fonksiyonu olarak, ayrıca modelin hareketli arka planı ne kadar iyi yakaladığına dayanarak ölçer. [56]’da, dört farklı arka plan çıkarımı algoritmasının performanslarının karşılaştırılmasında kullanılmış ve ROC analizine göre iki temel avantaj göze çarpmıştır: (1) ön plan dağılımının bilinmesine bağlı değildir, (2) videoda ön plan nesnelerinin bulunmasını gerektirmez. Bunlara bağlı olarak, PDR analizi, algoritmanın hassaslığı hakkında pratik genel bilgi sağlar.

Kim ve arkadaşları [21]’de önerdikleri yöntemin performansını, beş farklı yöntemle ROC analizi kullanarak değerlendirmişlerdir. Ardından, belirli bir doğru pozitif seviyesindeki yanlış pozitif oranları ile karşılaştırma yapılmıştır. Önerilen yöntem, Visual Studio 2005 ile C++ ortamında Core2Duo 3 GHz PC üzerinde implement edilmiştir. Hesaplama performansı, yaklaşık 37 fps olarak belirlenmiştir. Bu işleme hızında algoritma, gerçek zamanlı uygulamalarda kullanılabilir durumdadır.

Dewan ve arkadaşları [57]’de yapay sinir ağları yaklaşımı ile hareketli nesne tespit ve sınıflandırma yöntemi önermiştir. Arka plandan bağımsız olarak, önerilen kenar tespitine dayalı yöntemin hızlı ve çevre değişimlerine uyumlu olduğunu belirtmişlerdir. Hareketli ayrıtlar tespit edildikten sonra, yapay sinir ağı kullanılarak hareketli nesnelerin sınıflandırılması (insan, araç vb) yapılmıştır. Algoritma, P4 1.5GHz işlemci, 512 MB RAM özelliklerine sahip bir bilgisayarda C++ ile implement edilmiştir. Gri seviyeli görüntülerde, yaklaşık 10 fps işleme hızı ölçülmüştür. Bu yöntem, doğrudan şüpheli nesne tespiti problemine yönelik

olmamakla birlikte, şüpheli nesne tespiti sisteminde tespit edilen nesneleri sınıflandırılmak için kullanılabilir.

Bayona ve arkadaşları [53]'te arka plan çıkarımına dayalı ön plan nesne tespiti algoritmalarını karşılaştırmıştır. PETS2006 ve AVSS2007 için i-LIDS veri setlerinden seçilen videolar üzerinden algoritmalar test edilmiştir. İncelenen yaklaşımlar, hem nesnel hem de öznel olarak karşılaştırılmıştır. Ek olarak, bir ön plan nesnesi eğer 30 saniye boyunca durağan kalırsa, şüpheli nesne olarak algılanıp bir alarm tetiklenecektir. Nesnel ölçü olarak, zaman bakımından, alarm/tespit süresinin hassasiyeti ve isabet/recall oranı kullanılmıştır. Nesnel hesaplama için 5 öznel ölçü seçilmiştir. Bunlar ön plan çıkarımı hassasiyeti, perdelemeye karşı tolerans, gürültü, hesaplama yükü ve parametre ayarlama zorluğudur.

Campos ve arkadaşları [58]'de tespit edilen ön plan nesnelerini terk edilmiş ve kayıp nesne olarak ayırmak için, active contour kullanmayı önermiştir. Önerdikleri yöntemi test etmek için 2 GB RAM'e sahip P4 3 GHz özelliklerine sahip bir bilgisayar üzerinde implementasyonu yapmışlardır. Deneysel sonuçları, PETS2006, PETS2007, CVSG, VISOR, CANDELA gibi açık veri setlerinden seçtikleri bazı videolar üzerinden elde etmişlerdir. Karşılaştırmayı, ROC analizi ve minimum, maksimum ve ortalama hesaplama süreleri üzerinden yapmışlardır. Önerilen yöntem, tek başına bir şüpheli nesne tespit algoritması olmayıp, ön plan nesnesi olarak tespit edilen bölgeleri kayıp/terk edilmiş olarak sınıflandırmaya yarayan bir modül olarak kullanılabilir. Ancak yüksek doğru pozitif oranına rağmen, hesaplama yükü bakımından gerçek zamanlı bir sistemde kullanılmaya uygun bir yöntem olmadığı görülmektedir [58].

Bu tez kapsamında, bir şüpheli nesne tespiti algoritması gerçekleştirilmiştir. Kullanılan algoritmanın işlem süresinin uzun olmaması amaçlanmıştır. İncelenen yaklaşımın, ortalama güçte donanımına sahip bir bilgisayar üzerinde, makul bir hesaplama yüküne sahip olması, temel kriterimizdir. Bir diğer ölçüt ise, tespit/yanlış alarm oranıdır. Sistemin tutarlı ve güvenilir çalışması için, yanlış alarm oranının mümkün olduğunca düşük olması beklenmektedir. Dolayısıyla, ROC eğrileri üzerinden, nicel karşılaştırması yapılarak, yanlış alarm oranını minimum seviyede tutan algoritmalar dikkate alınacaktır.

3. ŞÜPHELİ NESNE TESPİT ALGORİTMASI

Bu çalışmada, terk edilmiş/şüpheli nesne tespiti problemine arka plan çıkarımı, Gauss karışımı modeli gibi genellikle sahnedeki ani veya periyodik değişimlerden fazlaca etkilenen geleneksel yöntemler yerine daha güncel ve son yıllarda üzerinde daha fazla çalışma yapılmaya başlanan [15] altuzay modeli ile bir çözüm sunulmuştur. Bu yaklaşımın seçilmesinin nedeni, hareketli bölgeleri arka plan bölgesinden net bir şekilde ayırırken nesnelere ait detayları da korumasıdır [27].

Yapılan literatür araştırmasında, altuzay modelinin genellikle hareketli ön plan nesnesi tespitinde kullanıldığı çalışmaların mevcut olduğu, fakat şüpheli nesne problemi için kullanılmadığı görülmüştür. Hâlihazırda altuzay modelinin kullanıldığı bir şüpheli nesne tespiti yapısı literatürde bulunmamaktadır. Literatürdeki ikili ön plan yaklaşımı, altuzay öğrenmesi ve her kare için adaptif bir eşikleme yöntemiyle birleştirilerek şüpheli nesne tespiti için yeni bir yaklaşım önerilmiştir. Bu yaklaşımın avantajı, arka plan hareketleri, literatürdeki yaklaşımlara göre, daha kolay bastırılabilir. Ayrıca, nesnelerin genel hatları da korunabildiğinden ön plan bölgelerini iyileştirmek için morfolojik operatörlerin kullanımı gibi ilave işlemlere gerek kalmamaktadır.

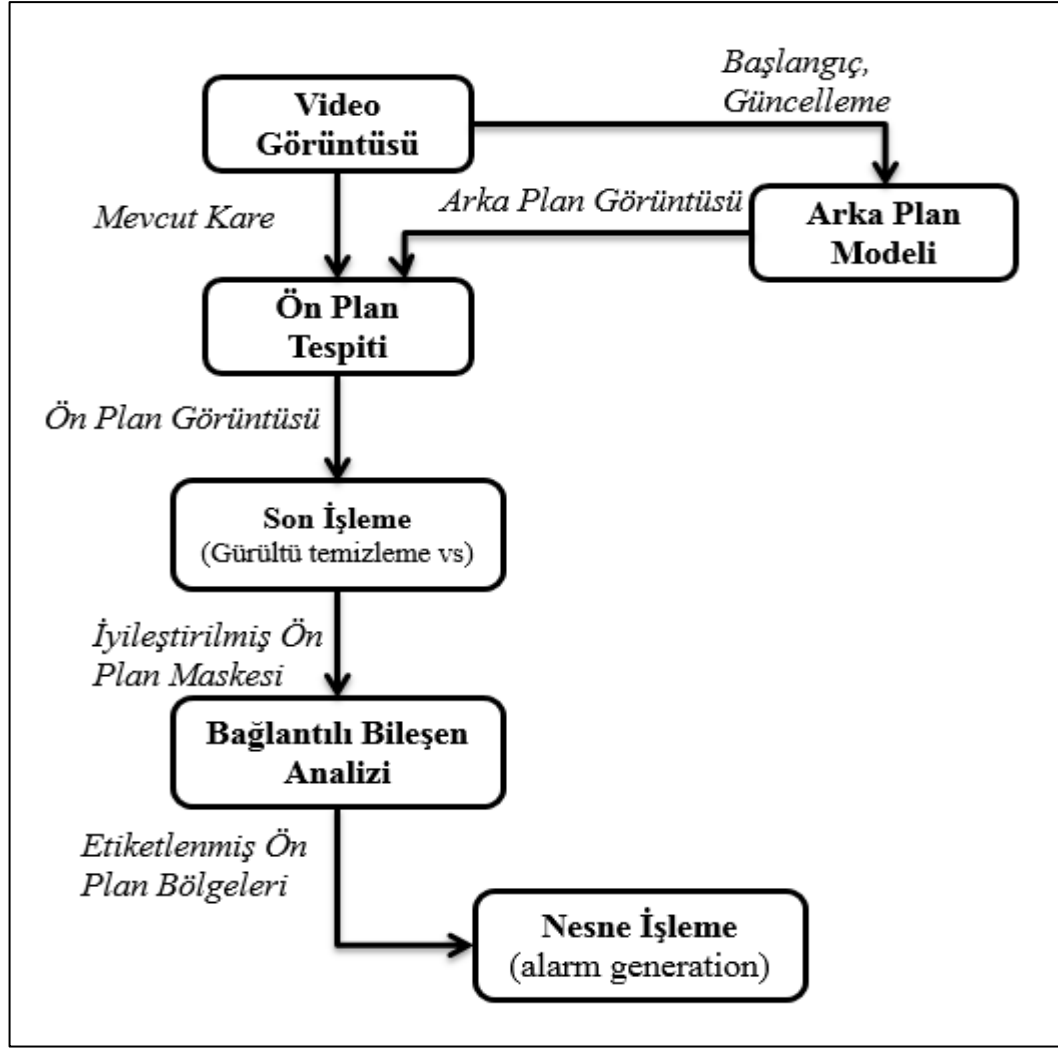
Altuzay yaklaşımında, video dizisinden alınan belirli sayıda kare, yığın olarak işlenmektedir. İşlenecek olan X data matrisi, video dizisinden alınan N adet karenin sütun vektörleri olarak düzenlenmesiyle oluşmaktadır. Örneğin, 720×576 çözünürlükte bir videodan 200 kare işlenecekse, oluşan X matrisinin boyutları 414720×200 olmaktadır. Video işareti, blok olarak işlendiğinden, altuzay modeli artımlı (incremental) proses yapan geleneksel yaklaşımlardan farklılık göstermektedir.

Şüpheli nesne probleminin çözümü için, bu tür nesnelerin video verisinin zaman ve uzamsal istatistiklerini nasıl değiştirdiğinin bilinmesi gerekmektedir [49]. Sistemin yapısı oluşturulurken, problem, şüpheli/terk edilmiş nesnenin başta orijinal sahnede olmadığı, sahneye bir süre sonra getirildiği ve bırakıldıktan sonra belirli bir süre sabit kaldığı düşünülerek tanımlanmıştır. Başka bir deyişle, şüpheli nesne başlangıçta sahnede olmayan, geçici olarak durağan bir nesnedir. Bu demek oluyor ki, statik sahneyi ve hareketli ön planı öğrenerek bir pikselin şüpheli nesneye ait olup olmadığı hipotezi kurulabilir. Herhangi bir andaki video karesinde, işleme hızına

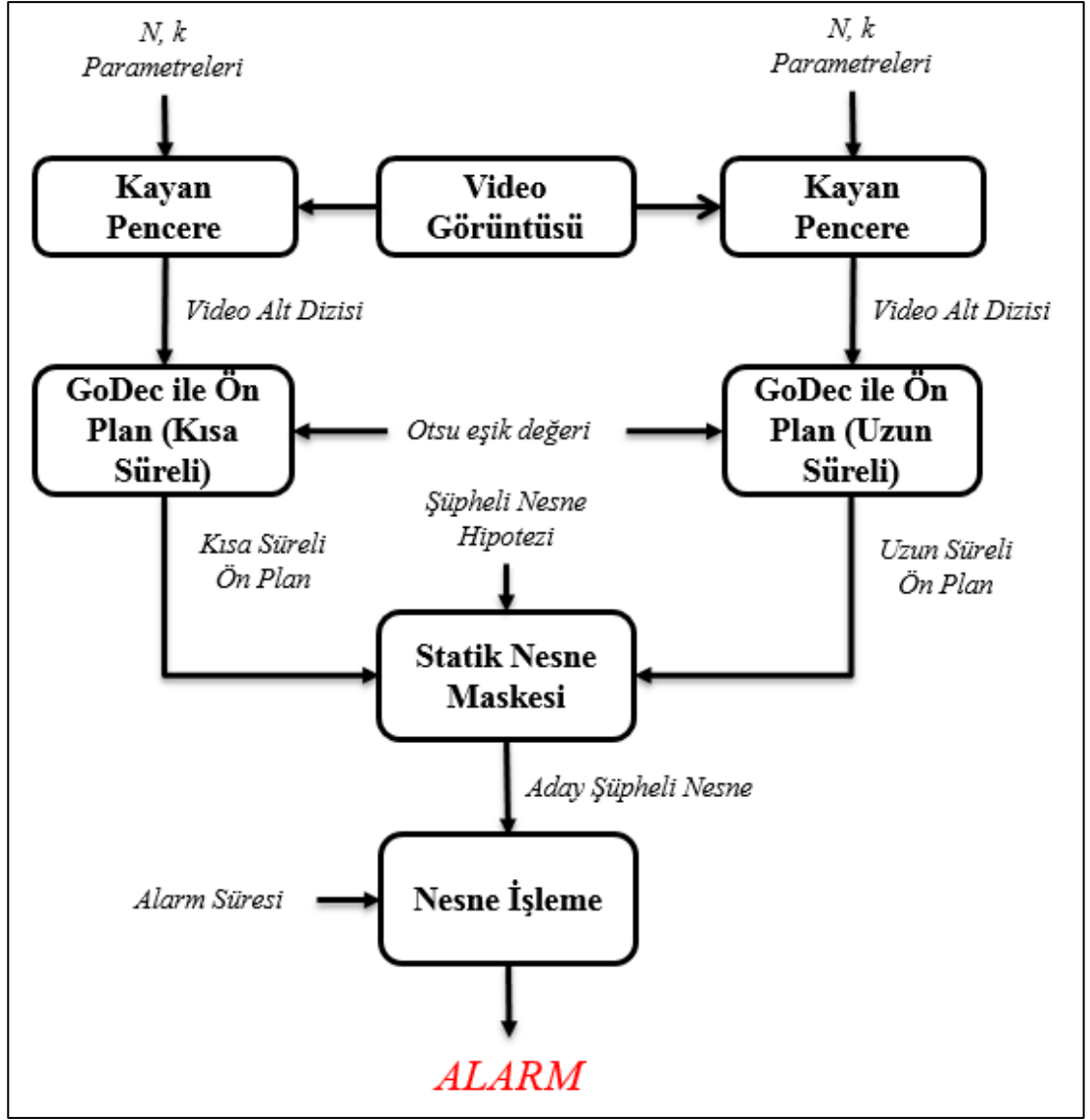
bağlı olarak, geçici olarak statik nesne arka planın bir parçası olarak algılanabilir (düşük işleme hızları) veya hareketli bölgelerle gruplanabilir (yüksek işleme hızları) [3]. Bu yüzden tek bir arka plan modeli geçici olarak durağan pikselleri arka plandan ayırmaya yeterli değildir.

Belirtilen sebeplerden dolayı, farklı işleme hızlarında elde edilmiş iki ön plan maskesi (uzun süreli ve kısa süreli) kullanılarak geçici durağan bölgelerin tespit edilmesi mümkündür. Bu ön plan maskeleri, altuzay öğrenme yaklaşımı ile, spesifik olarak; 2. Bölümde ayrıntılı anlatılan GoDec algoritması ile elde edilmektedir. GoDec algoritmasından yararlanıldığını ve iki ön plan maskesi kullanıldığını vurgulamak üzere, bu çalışmada şüpheli nesne tespiti için önerilen algoritma *GoDec-II* olarak adlandırılacaktır. Burada not olarak düşmek gerekirse, ikiden fazla ön plan modeli kullanılarak tespit haritasında oluşan parçacıklı yapı iyileştirilebilir [49].

Şüpheli nesne tespiti algoritmaları genel olarak Şekil 3.1’de gösterilen akışı izlemektedir. Öncelikle, video görüntüsünden hareketli ön plan nesneleri tespit edilir. Bunun için mevcut yaklaşımlar, Şekil 2.3’te sınıflandırılmıştır. Ardından son işlemede ikili görüntü gürültü temizleme gibi işlemlerden geçer. Sonuç, bağlantılı bileşen analizi ile etiketlenir ve şüpheli nesneye ait olan bölgeler için alarm üretilir.



Şekil 3.1: Şüpheli Nesne Tespiti Algoritması Genel Blok Şeması.



Şekil 3.2: Önerilen Şüpheli Nesne Tespit Algoritması (GoDec-II) Blok Şeması.

3.1. Altuzay Öğrenmesi ile Ön Plan Tespiti

Şekil 3.2’de bu çalışmada önerilen algoritma için genel blok şema görülmektedir. Burada, literatürde şüpheli nesne tespiti için önerilen yaklaşımlardan farklı olarak altuzay modelinin yığın (batch) işlenmesi gerektiğinden videoyu kare-kare işlemek yerine, her adımda belli sayıda video karesi işlenmektedir. Bu yüzden, X data matrisi, kayan bir pencere ile video dizisinden alınan N adet karenin sütun vektörleri halinde düzenlenmesiyle oluşmaktadır. Kayan pencerenin adım aralığı k ve pencere adım genişliği N , algoritmanın temel parametreleridir. Bu parametrelerden N bir iterasyonda video dizisinden kaç adet kare işleneceğini

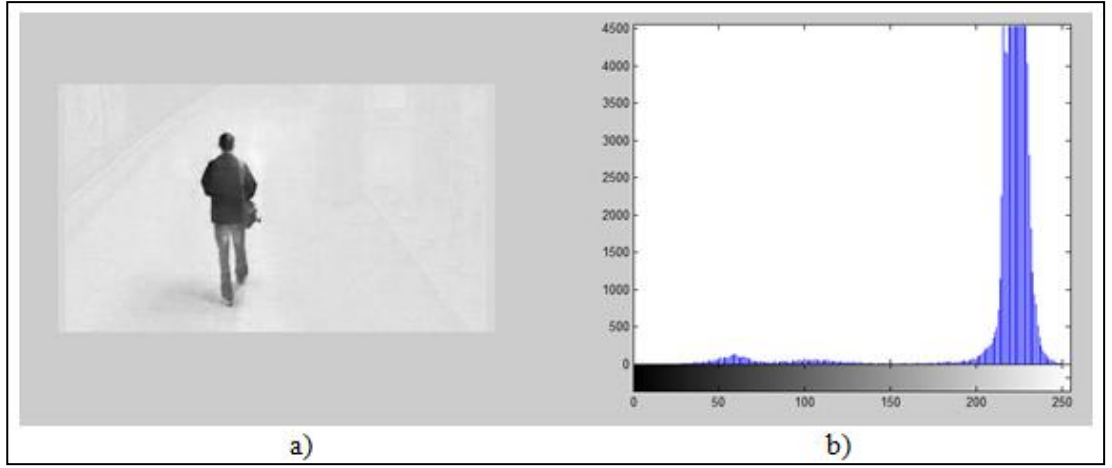
belirlemektedir ve yapılan deneylerde N 'nin yüksek değerleri için “*düşük mertebe+seyrek*” ayrıştırmasının kalitesinin arttığı gözlemlenmiştir. Parametre k , pencerenin her iterasyonda ne kadar kayacağını belirlemektedir. $k = 0$ olduğunda, pencere video dizisi üzerinde üst üste gelmeyen bölgeler alınarak işlenir. $k \neq 0$ için, pencere k kare kadar kayarak yine N adet video karesi işler. Bu N adet karenin son k adet karesi yeni video karelerinden, $N - k$ tanesi de bir önceki iterasyonda işlenen eski video karelerinden oluşmaktadır. Bu sayede, eski video karelerinin yeni modele etki etmesi sağlanmaktadır. k parametresi ile durağan bir ön plan nesnesinin, örneğin terk edilmiş bir nesne, arka plana ne kadar geç karışacağı belirlenmektedir.

k parametresinin sıfırdan farklı değerleri için uzun süreli ön plan maskesi (yani arka planın daha yavaş değiştiği), $k = 0$ için de kısa süreli ön plan maskesi (arka planın daha hızlı değiştiği) elde edilmektedir. İkili ön plan modeli, Porikli tarafından [3] ve [49]'da anlatılmıştır.

3.1.1. Gri Seviyeli Ön Plan Görüntüsü Eşikleme

GoDec algoritması, N ve k parametreleri ile belirlenen video alt dizisinden oluşmuş iki data matrisi X_{Long} ve X_{Short} 'yi *Low rank (L)* ve *Sparse (S)* bileşenlerine ayırmaktadır. S bileşeni gri seviyeli olarak ön planı, hareketli nesneleri göstermektedir. Siyah-beyaz ön plan maskesini elde etmek için S video dizisi Otsu'nun global eşikleme [59] yöntemiyle eşiklenmektedir. Otsu'nun yöntemi, siyah ve beyaz piksellerin histogramını kullanarak sınıf içi varyansını minimum yapan bir eşik değeri belirlemektedir. Böylece, her video karesi için adaptif bir eşik değeri belirlenmektedir.

Şekil 3.3'te i-Lids veri setine ait “AVSS_AB_easy” videosundan örnek bir kare görülmektedir. Şekil 3.2 soldaki görüntüde GoDec algoritması ile ayrıştırılmış S bileşeni ve sağda bu bileşene ait histogram görülmektedir. Histogramdan da açıkça görüldüğü üzere sahnedeki piksellerin çoğunluğu belirli bir renk aralığında (200-250) toplanmıştır. Hareket eden insana ait pikseller ise yaklaşık olarak 50-70 yoğunluk aralığında kalmıştır. Bu sahne için Otsu'nun yöntemi ile eşikleme yapıldığında, eşik değeri 151 ve eşikleme performansı 0.9 olarak bulunmuştur.



Şekil 3.3: Örnek Bir Karenin (Kare No: 854) a) Seyrek (Ön Plan) Bileşeni, b) S Bileşeninin Histogramı.



Şekil 3.4: Kare No: 854'ün Otsu İle Eşikleme Sonucu.

Şekil 3.4'te, Şekil 3.2'de verilen örnek karenin eşikleme sonucu görülmektedir. Bariz bir şekilde Otsu'nun bu tür görüntü üzerindeki yüksek performansı görülmektedir. GoDec ile $L + S$ şeklinde ayrıştırılarak elde edilen ön plan görüntülerinde, hareketli bölgeler kendi piksel yoğunluk değerini korurken, arka plan (*low rank*) bölgesinde kalan piksel yoğunluk değerleri gözle görülür derecede farklı olmaktadır. Böylece, görüntüde iki tepeli bir yoğunluk dağılımı oluşmakta ve bu durum da Otsu ile eşiklemeyi uygun kılmaktadır.

Hem S görüntüsünden, hem de eşiklenmiş görüntüden anlaşıldığı gibi GoDec algoritması ön plan bölgelerini bulurken nesnelerin genel hatlarını gerçek bilgiye (ground truth) yakın tutmaktadır. Bu durum, ileri safhalarda uygulanması muhtemel, örneğin nesne sınıflandırma gibi adımlar için büyük kolaylık sağlamaktadır.

3.2. Statik Nesne Maskesi ve Şüpheli Nesne Hipotezi

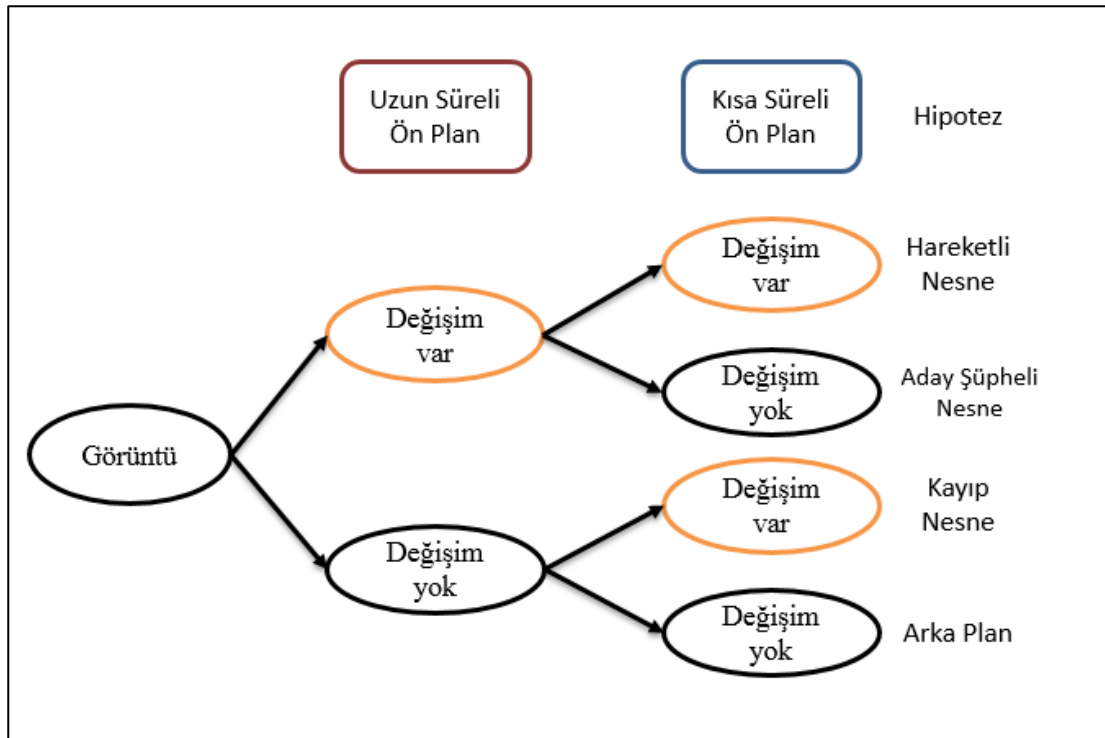
Ön plan tespiti ve eşikleme yapılarak, herhangi bir andaki sahnede mevcut hareketli nesneler beyaz, arka plan bölgesi ise siyah olarak gösterilmektedir. Herhangi bir t anındaki ön plan maskesinde, (x,y) konumundaki pikseli $F_t(x,y)$ ile gösterilsin. $F_t(x,y) = 1$ ise, bu piksel hareketli nesneye aittir.

Durağan ön plan nesnelerinin tespiti için tek ön plan modelinin yeterli olmadığı, iki veya daha fazla model uygulanarak bu tür bölgelerin tespit edilebileceği önceki bölümde açıklanmıştı. Bu yüzden, biri uzun, diğeri kısa süreli değişimleri gösteren iki adet ön plan maskesi hesaplanmaktadır. Herhangi bir t anında, kısa süreli ve uzun süreli ön plan maskelerine ait (x,y) pikseli sırasıyla $F_S(x,y)$ ve $F_L(x,y)$ ile gösterilmektedir. Uzun süreli ön plan maskesi, hareketli nesneler, geçici statik nesneler de dâhil olmak üzere sahnede daha önce var olmayan renk değişimlerini gösterir. Kısa süreli ön plan ise hareketli nesneleri, gürültü vb. özellikleri göstermektedir. Bu iki ön plan maskesinin değerlerine bağlı olarak, şu hipotezler oluşturulabilir:

- $F_L(x,y) = 1$ ve $F_S(x,y) = 1$ ise, yeni video karesi $I_t(x,y)$ iki arka plandan da farklı olduğu için (x,y) pikseli hareketli nesneye ait olabilir.
- $F_L(x,y) = 1$ ve $F_S(x,y) = 0$ ise, piksel geçici olarak statik durumdaki bir nesneye ait olabilir.
- $F_L(x,y) = 0$ ve $F_S(x,y) = 1$ ise, daha önce örtülü (occluded) olan bir arka plan pikseli olabilir.
- $F_L(x,y) = 0$ ve $F_S(x,y) = 0$ ise, arka plan pikselidir.

Kısa süreli arka plan daha hızlı değişmekte ve kendini daha hızlı adapte etmektedir. Buna zıt olarak, uzun süreli arka plan daha yavaş değişmekte ve değişimlere karşı daha dirençlidir.

Bir sahnede, arka plan pikseli geçici olarak değişip tekrar orijinal değerine döndüğünde, uzun süreli ön plan maskesinde değeri sıfır olacaktır. Kısa süreli arka plan ise kendini hızlı bir şekilde uyumlaştıracak ve $F_S(x, y) = 1$ olacaktır. Değişen bir piksel, eğer yeterince uzun süre değerini korursa, kısa süreli ön plandan silinecektir. Bu süre uzun süreli ön planı değiştirecek kadar uzun olmadığında $F_L(x, y) = 1$ olacaktır. Bu durum şüpheli/terk edilmiş nesne içindir [49].



Şekil 3.5: Olası Durum Hipotezleri.

Şekil 3.5'te uzun ve kısa süreli ön plan maskelerine uygulanan hipotezler grafik olarak görülmektedir. Bu çalışmada incelenen problem için, ikinci hipotez, aday şüpheli nesneye ait pikseli işaret etmektedir. Yani bir pikselin, uzun süreli ön plan maskesinde aldığı değer 1, kısa süreli ön plan maskesinde aldığı değer 0 ise bu piksel olası şüpheli nesneye aittir denir.

3.3. Statik Nesne Maskesi

Her video karesinde yapılan tespit sonuçlarını kullanarak bir statik nesne maskesi oluşturulmaktadır. Bu maskenin değeri, orijinal video karesinin boyutlarında, iki numaralı hipotezi sağlayan her piksel için, her bir karede artırılmaktadır.

$$E(x, y) = \begin{cases} E(x, y) + 1, & F_L(x, y) = 1 \wedge F_S(x, y) = 0 \\ E(x, y) - n, & F_L(x, y) \neq 1 \vee F_S(x, y) \neq 0 \\ 0, & E(x, y) \leq 0 \end{cases} \quad (3.1)$$

Statik nesne maskesinin her yeni video karesindeki tespitlerle güncellenmesi eşitlik (3.1)'de verilmiştir. Bu maske, tespit işlemindeki olası gürültüleri de temizlemeye imkân sağlamaktadır. Zamansal tutarlılığı sağlamak amacıyla, E maskesinde herhangi bir piksel değeri belirli bir seviyeye geldiğinde, o piksel şüpheli nesneye ait olarak işaretlenmektedir. Bu eşik seviyesi, sistemin cevap süresine ve gürültü karakteristiğine göre değişiklik gösterebilmektedir. Örneğin tespit aşamasında çokça gürültülü sonuç varsa, eşik seviyesi yüksek seçilebilir. Bu eşik seviyesinin yüksek olması, yanlış alarm oranını düşürmektedir. Fakat öte yandan, alarmin üretilmesi için geçen süreyi de uzatacaktır.

n sabiti, maskenin ne kadar hızlı azalacağını belirlemektedir. Başka bir deyişle, şüpheli nesne olarak işaretlenen bir pikselin arka plana karışması veya orijinal değerine dönmesi durumunda ne olacağını belirler. Eğer nesne yerinden alınır alınmaz alarmin kapatılması isteniyorsa, n değeri yüksek seçilmelidir.

3.4. Perdeleme (Occlusion) Engelleme

Metro, havaalanları, şehir merkezleri gibi kalabalık sahnelerde ilgilenilen bir bölge/nesneye ait pikseller başka nesneler veya etkiler tarafından kısmi veya tam olarak kapatılabilir, perdelenabilir. Bu durum yapılan tespit performansını olumsuz yönde etkileyecektir. Hatta bu tür sebeplerle tespit edilmesi beklenen nesne kaçırılabilir. Bu da sistemin güvenilirliği açısından sorun teşkil etmektedir.

Statik nesne maskesi ile bu perdelenme sorunu kısmi olarak engellenebilmektedir. Şüpheli nesne hipotezini sağlayan piksellerin statik nesne maskesindeki değerini her video karesinde artırarak tutarlılık sağlanmaktadır.

Dolayısıyla, statik nesneye ait bir piksel perdelenmeye maruz kalıp şüpheli nesne hipotezini sağlamadığında, statik nesne maskesindeki değeri azalmaktadır. Bu maskede tamamen kaybolmadığı için perdelenme ortadan kalktığında tespit hala mümkün olmaktadır. Ancak, statik nesne maskesinin azalma sabiti n gürültüye bağışıklık sağlamak için büyük seçildiğinde perdelenmeye maruz kalan piksel çabukça kaybolabilir ve tespit edilmesi gereken bir bölge kaçınılabılır. Bu yüzden güvenlik seviyesinin artması gereken durumlarda n parametresine düşük değer atanabilir, acil müdahale gerektirmeyen ve operatörün hatalı alarmları yönetemeyeceği kadar çok sayıda ekrana bakması gibi durumlarda n parametresi büyük seçilebilir.

Bu yüzden, statik nesne maskesinde eşik değerini aşan şüpheli nesneye ait piksellerin bazı istatistiki bilgileri tutulmaktadır. Statik nesnenin toplam alanındaki değişim oranı, merkez noktasının değişim oranı, tespit edildikten sonra kaybedilen bir nesnenin alarminin kapatılmadan önce geçmesi gereken süre gibi bilgiler işlenerek, kısmi veya tam perdelenmeye karşı bağışıklık, bırakılmış bir nesnenin çeşitli nedenlerle ufak yer değiştirmesi vb. durumlarda tutarlılık sağlanmakta ve yanlış alarm oranı azaltılmaktadır.

Bu adım blok şemada ‘Nesne İşleme’ ile gösterilmiştir.

3.5. Alarm Üretme

Bu çalışmada tasarlanan şüpheli nesne tespit sisteminin, yarı otomatik olması hedeflenmektedir. Yani, güvenlik görüntülerini izlemekle görevli personel, operatörün performansını artırmaya yönelik bir sistem olarak düşünülmüştür. Şüpheli bir olay gerçekleştiğinde sistemin bu olayı tespit ederek, birçok gözetleme monitörü arasından belirli bir tanesine ve o monitörde de olayın gerçekleştiği belirli bölgeye operatörün dikkatini çekmesi beklenmektedir. Bu yüzden tespit edilen olayın ardından uyarıcı bir alarm üretilmesi gerekmektedir.

Nesne İşleme adımından sonra, belirli bir süre eşiğini geçen, örneğin 30 saniye boyunca durağan kalan, piksel bölgelerine bağlantılı bileşen analizi uygulanarak bu bölgeler etiketlenmektedir. Ayrıca, etiketlenmiş her bir bağlantılı bileşeni çevreleyen sınırlar (*Bounding Box*) belirlenmektedir. Bu sınırlar tespit yapıldığı orijinal video karesi üzerinde operatörün dikkatini çekecek şekilde farklı renklerde işaretlenmektedir. Böylece, operatör doğrudan olayın gerçekleştiği bölgeyi

görebilmektedir. Bunun yanında operatörün dikkatinin çok fazla dağıldığı durumların da olabileceğini göz önünde bulundurarak, görsel uyarının yanında sesli bir alarm da üretilebilir.

4. DENEYSEL SONUÇLAR VE KARŞILAŞTIRMA METRİĞİ

Bir nesne tespit algoritmasının performansının etkili bir şekilde ölçülmesi hem başka algoritmalarla objektif karşılaştırma yapılabilmesi açısından, hem de o algoritmanın zayıf yanlarının belirlenmesi açısından önemlidir. Adil bir karşılaştırma için de probleme uygun bir değerlendirme metriği seçilmeli ve ona göre performans ölçümü yapılmalıdır.

Bu çalışmada şüpheli nesne problemi için literatürde en sık kullanılan algoritmalar ve bir ara adım olan arka plan modellemesi ve güncellemesi için önerilen bazı yöntemler incelenmiştir. Ayrıca, bu yöntemler tez kapsamında önerilen yaklaşım ile karşılaştırılmıştır. Karşılaştırma için kullanılabilecek ve literatürde mevcut metrikler 2. Bölümde anlatılmıştır. Bu çalışmada algoritmanın performansını değerlendirmek ve diğer yöntemlerle karşılaştırmak için [60]’ta önerilen metrik kullanılmıştır.

4.1. Nesneye Dayalı Performans Metriği

Nesne tespiti, segmentasyonu, takip gibi bilgisayarla görme uygulamalarında uygulanan algoritmanın performansını görsel sonuçlara bakarak belirlemek hem oldukça zor hem de görecelidir. Dolayısıyla tespitin kalitesini göstermek için tutarlı, sayısal sonuçlara ihtiyaç duyulmaktadır.

Bu çalışmada, literatürde en çok yer edinmiş şüpheli nesne tespiti algoritmalarından bazıları gerçekleştirilmiş ve önerilen algoritma ile aşağıdaki performans metrikleri kullanılarak karşılaştırılmıştır.

Şüpheli nesne tespiti probleminde, bir algoritmanın performansı, ön plan tespiti kalitesinin yanında, farklı yollarla belirlenebilir [60]:

- Gerçek Bilgideki (*Ground Truth*) olay sayısı ile, örneğin video süresince toplamda kaç adet sahipsiz nesne olduğu, algoritmanın tespit ettiği olay sayısına göre pozitif doğru, negatif doğru, pozitif yanlış ve negatif yanlış istatistikleri kullanılabilir.

- Nesneye dayalı, yani tespit edilen ve gerçek bilgidaki nesnenin video dizisi boyunca ömrü ile ölçülebilir. Bir başka deyişle, nesnenin gerçek bilgi ve algoritma sonucunda ne kadar süre ve süreklilikle tespit edildiği incelenebilir.

Problemin tanımına göre, iki metrik de kullanılabilir ve ikisi de kabul edilir sonuç verebilir.

Bu çalışmada, geliştirilen yöntem yarı otomatik, yani operatörün performansını artırmaya, işini kolaylaştırmaya yönelik olduğu için ve algoritma bir nevi izleme algoritması olduğundan, [60]'ta önerilen (1) ve (2) numaralı metrik kullanılmıştır. Bu sayede, hem tespit performansı, hem de izleme (tracking) performansı tarafsız olarak ölçülebilmektedir [60].

4.2. Deneysel Sonuçlar

Tez çalışmasında önerilen GoDec-II olarak adlandırdığımız, GoDec ile İkili Ön Plan Yaklaşımı, 2. Bölümde bahsedilen açık veri setlerinden bazıları üzerinde denenmiştir. Bu videolar üzerinden elde edilen sonuçlar, nesneye dayalı performans metriği ile ölçülmüştür.

Önerilen yöntemin performansını ölçmek için kullanılan videolar, i-Lids 2007 ve PETS2006 veri setinden alınan şüpheli nesne videolarıdır. i-Lids veri setinden, AVSS 2007 konferansı için farklı zorluk seviyelerinde beş video (AVSS_AB_Easy ve AVSS_AB_Medium, Parked Vehicle) ve PETS2006 veri setinden aynı sahnede farklı açılardan çekilmiş iki video üzerinde test edilmiştir. Ek olarak, algoritmanın farklı koşullarda da çalıştığını, örneğin gece ve dış mekânlarda, göstermek için [61]'de sunulan veri setlerinden *'tramStation.mp4'* videosu, Visor [62] veri setinden park edilmiş araç videosu ve PETS2001 veri setinden bir video kullanılmıştır.

Bunların yanında önerilen algoritma, Gebze Teknik Üniversitesi'nde sahipsiz nesne ve park edilmiş araç senaryoları için, iç/dış mekânda, değişen ışık ve sahne koşullarında çekilmiş toplamda 70 dakika ve 65 bin karelik veri üzerinde test edilmiştir.

i-Lids veri setine ait videolarda, algoritmaya uygulanmadan önce, video bilgisinin verildiği ilk birkaç saniyelik bölümler kesilmiş ve işlem yükünü azaltmak için videoların genişlik ve yüksekliği azaltılmıştır. Orijinal halde, 720x576 (WxH) boyutlarında olan videolar, yarı çözünürlüğe dönüştürülerek işlenmiştir.

Bu çalışmada önerilen GoDec-II yaklaşımı için, ‘AVSS_AB_Easy’ videosuna ait tespit sonuçları Şekil 4.1’de grafik olarak gösterilmiştir. Bu videoda toplamda 5222 kare bulunmaktadır.



Şekil 4.1: Önerilen Yöntemin AVSS_AB_Easy Videosuna Ait Sonuçları.

Bu video süresince toplamda bir adet şüpheli nesne olayı gerçekleşmektedir. Bir valiz, sahibi tarafından yaklaşık 1760 numaralı video karesinde sahneye getirilmektedir. Bir süre valizin etrafında bekledikten sonra ayrılmaktadır. Valiz bu pozisyonunda 4630 numaralı video karesine kadar durmakta ve bu andan itibaren sahneden götürülmektedir. Önerilen algoritma, durağan ön plan nesnesini 2230 numaralı karede, yaklaşık 30 saniye sonra tespit etmektedir. Şüpheli nesne sahnede kaldığı süre boyunca alarm üretilmiştir. Gerçekleşen olay, hatasız olarak tespit edilmiştir.

Şekil 4.2 ve Şekil 4.3'te 'AVSS_AB_Easy' videosunda elde edilen sonuçları gösteren örnek bir kare görmek mümkündür. Şekil 4.2'de solda 3200 numaralı video karesi, sağda ise bu kareye ait statik nesne maskesi görülmektedir. Açıkça görüldüğü gibi, sağdaki maske gri seviyelidir. Yoğunluk seviyesi, şüpheli nesne hipotezini daha uzun süre sağlayan pikseller bu maskede daha parlaktır. Geri kalan bölgeler, hareketli nesnelere ait piksellerdir. Bu maske eşiklenerek siyah-beyaz yapılmış hali Şekil 4.3'te soldadır.



Şekil 4.2: $F = 3200$, a) Orijinal Sahne, b) AVSS_AB_Easy Statik Nesne Maskesi.

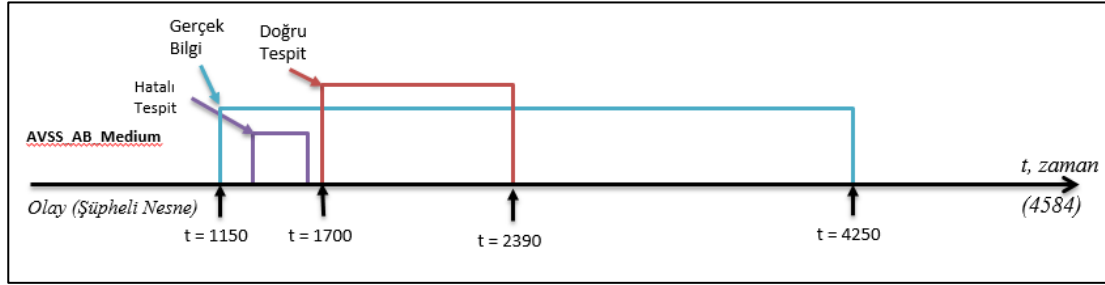
Bu ikili maskede tespit edilen ön plan nesnelerinin istatistikleri, Şekil 3.2’de Nesne İşleme modülünde elde edilerek, şüpheli nesneye ait çerçeve elde edilmektedir. Şekil 4.3’te solda valiz haricinde trenin de bir parçası görülmektedir. Fakat Nesne İşleme modülünde, bu bölge şüpheli nesneye ait gereksinimleri sağlamadığı için, Şekil 4.3’te sağda işaretlenmemiştir. Şüpheli nesneye ait çerçeve, orijinal kare üzerine yerleştirildiğinde, Şekil 4.3 sağdaki görüntü elde edilmiştir.



Şekil 4.3: $F = 3200$, a) Aday Şüpheli Nesne Maskesi, b) Tespit Edilen Şüpheli Nesne.

Test yapılan diğer video, i-Lids setinden ‘AVSS_AB_Medium’ videosudur. Bu videoda sahne, diğerine göre oldukça kalabalıktır ve başlangıçta sahnede bekleyen insanlar mevcuttur. Ayrıca, tespit edilecek nesne, kameradan uzak bir mesafeye koyulan ve nispeten küçük bir çantadır. Çanta, 1150 numaralı karede sahneye bırakılmaktadır. İlk videoda olduğu gibi, çantanın sahibi bir süre etrafında bekledikten sonra sahneyi terk etmekte ve çanta, 4250 numaralı karede tekrar götürülene kadar durmaktadır. Bu süre içerisinde, çanta ile kamera arasından geçen insanlar perdelemeye neden olmaktadır. Şekil 4.4’teki grafikte de görüldüğü gibi,

tespit 1740-2400 kareleri arasında gerçekleşmektedir. Bu videoda, 1210-1440 kareleri arasında durağan halde bir insan tespit edilmektedir. Videodaki mevcut perdeleme sayısına ve sahnenin karmaşıklığına bakarak, bir hatalı tespite rağmen algoritma çantayı yaklaşık 20 saniyede tespit etmekte ve tespit 28 saniye sürmektedir. (1) ve (2) numaralı metriklerle göre, hatalı bir tespite rağmen algoritmanın hala iyi performans verdiğini söylemek mümkündür.



Şekil 4.4: GoDec-II Algoritmasının AVSS_AB_Medium Videosu İçin Performansı.

Şekil 4.5 ve Şekil 4.6’da ‘AVSS_AB_Medium’ videosu için elde edilen sonuçlar görsel olarak verilmiştir. Şekil 4.5 ve Şekil 4.6’daki örnek karelerde, 2224 numaralı kare için önerilen algoritma ile elde edilen sonuçlar gösterilmiştir. Şekil 4.5 solda orijinal sahne görülmektedir. Bu sahnede, arkaya bırakılan çanta, tespit edilecek nesnedir. Aynı şekilde sağda, bu karedeki statik nesne maskesi görülmekte ve bu görüntü eşiklendiğinde, Şekil 4.6 soldaki görüntü elde edilmektedir. Bu sahnede, çanta haricinde farklı bölgeler de tespit edilmiş fakat bu bölgeler Nesne İşleme modülünde elenmiştir.



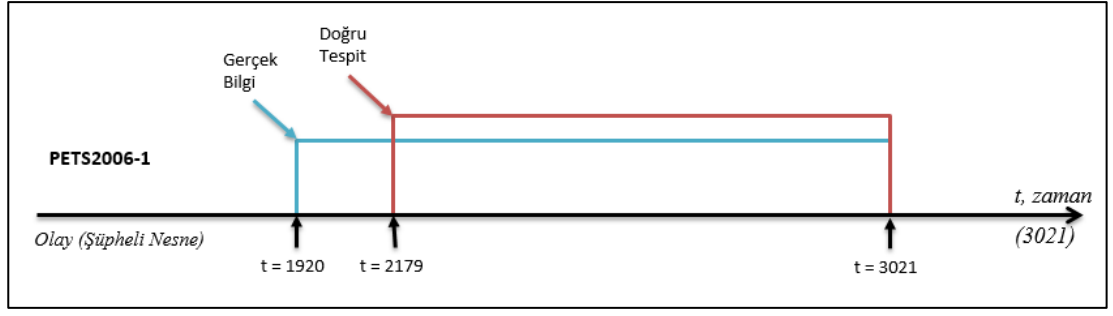
Şekil 4.5: F=2224, a) Orijinal Kare ve b) Statik Nesne Maskesi.



Şekil 4.6: a) Aday Şüpheli Nesne Maskesi, b) Tespit Edilen Şüpheli Nesne.

Geliştirilen algoritma, PETS2006 veri setinden şüpheli nesne senaryosuna ait 2 video üzerinde test edilmiştir. Bu videolar, aynı sahnenin iki farklı açıdan kaydedilmiş halidir. 3000 karelik videolar, orijinalde 720x576 çözünürlüktedir ve i-Lids gibi bu videolar da 360x288, yani yarı çözünürlükte işlenmiştir. Bu videoda da tek bir olay mevcuttur. Yaklaşık 1900 numaralı karede, sahneye bir sırt çantası bırakılmakta ve videonun sonuna kadar sahnede kalmaktadır. GoDec-II algoritmasının ilk video için tespit sonucu Şekil 4.7’de görülmektedir.

Şekil 4.7’de görüldüğü gibi, algoritma çantayı 2179 numaralı karede, yani yaklaşık 10 saniye sonra tespit etmektedir. Yapılan tespit videonun sonuna kadar kalmaktadır.

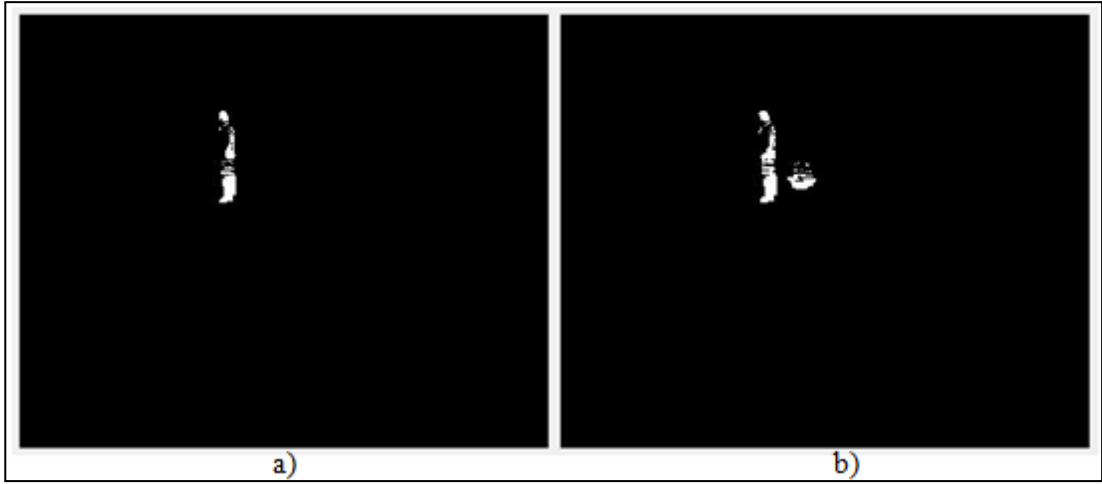


Şekil 4.7: PETS2006-1 Videosu Test Sonucu.



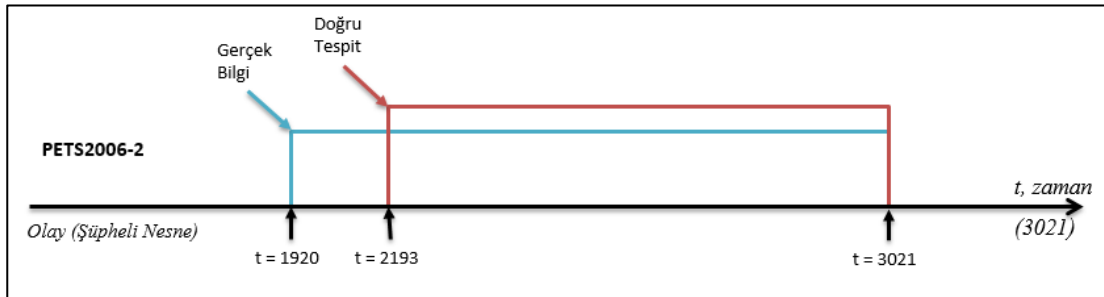
Şekil 4.8: PETS2006 - 1 Videosundan Örnek Bir Kare (F = 2280).

Şekil 4.8’de örnek bir karedeki tespit sonucu gösterilmiştir. Çantanın arkasındaki koyu camlı bölge ile renk karşıtlığı oldukça azdır. Bu yüzden, çantanın üst bölümü tam tespit edilememiştir. Şekil 4.9 ise kısa ve uzun süreli ön plan maskelerinde durağan nesnelerin durumu görülebilmektedir. Örnek kare PETS2006 - 1 videosuna ait 2000 numaralı karedir. Şekil 4.9’da solda kısa süreli ön plan maskesi, sağda ise uzun süreli ön plan maskesi gösterilmiştir.



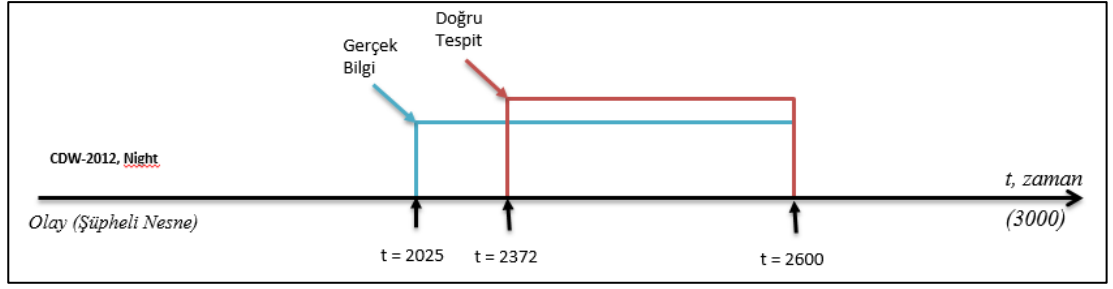
Şekil 4.9: F=2000 İçin Kısa a) Sürelî Ön Plan Maskesi, b) Uzun Sürelî Ön Plan Maskesi.

PETS2006 setinden kullanılan diğer video, aynı sahneyi daha yatay bir açıdan gören kameradan alınmıştır. Diğer kameranın görüş açısına girmeyen bazı bölümler mevcuttur. Şekil 4.10'da GoDec-II algoritmasının ikinci videodaki performansı görülmektedir. Bu videoda da mevcut olay başarılı bir şekilde, yaklaşık 11 saniyelik bir gecikmeyle tespit edilmiştir.



Şekil 4.10: PETS2006-2 Videosu İçin Test Sonucu.

CDW-2012 videosu, GoDec-II algoritmasının gece, dış mekânlarda ve ışık değişimlerinin fazla olduğu ortamlarda da çalışabildiğini göstermiştir. Bu videoda, bir araba, 2025 numaralı karede gelip yol kenarında durmaktadır. GoDec-II algoritması ile araç, 2372 numaralı karede tespit edilmekte ve araç ayrılana kadar alan üretilmektedir. Şekil 4.11'de bu videoya ait test sonucu verilmiştir.



Şekil 4.11: CDW-2012 Gece Videosu İçin GoDec-II Test Sonucu.

CDW-2012 videosu gece, trafiğin işlek olduğu bir saatte çekilmiştir. Yoldan çok sayıda araç geçmekte, bunun yanında yol kenarında da yayalar yürümektedir. Ayrıca, bu sahne için belirtilmesi önemli bir durum da, sahnenin sağ tarafında oldukça belirgin ve değişken bir ışık kaynağı (dinamik arka plan) bulunmaktadır. GoDec-II algoritmasının performansının, bu durumdan etkilenmediği Şekil 4.12'deki örnek görüntülerden görülebilir.



Şekil 4.12: F = 2495 İçin a) Statik Nesne Maskesi, b) Tespit Edilen Araç.

4.3. GoDec-II ile Mevcut Yöntemlerin Karşılaştırılması

Bu çalışmada, literatürde yer edinmiş şüpheli nesne tespiti algoritmalarından bazıları gerçekleştirilmiş ve önerilen algoritma ile nesneye dayalı performans metriği kullanılarak karşılaştırılmıştır.

Gerçeklenen yöntemler arasında Porikli ve arkadaşları [49], Wang ve Liu [12] mevcuttur. Bahsedilen yaklaşımlar doğrudan şüpheli nesne problemi için önerilen yöntemlerdir. Bunların yanında, Sigari ve arkadaşları [10], Wang ve Liu [12]'de önerilen arka plan güncelleme modelleri gerçekleştirilerek, ön plan tespit kalitesine etkisi değerlendirilmiştir.

Tablo 4.2 ve Tablo 4.3'te bu çalışmada önerilen GoDec-II yönteminin ve incelenen Porikli ve arkadaşları [49] ve Wang ve Liu [12] yöntemlerinin veri setleri üzerindeki performans sonuçları verilmiştir. Tablolarda, T videodaki tüm kare sayısını, T_{olay} şüpheli nesnenin bulunduğu toplam kare sayısını, $Olay$ videodaki şüpheli nesne sayısını göstermekte ve bu bilgiler, gerçek bilgiyi (Ground Truth) göstermektedir. DT ve HT , sırasıyla algoritma tarafından doğru tespit edilen olay sayısını ve hatalı tespit (üretilen yanlış alarm) sayısını göstermektedir. T_D doğru tespitin gerçekleştiği kare sayısını, T_H ise hatalı tespitin gerçekleştiği kare sayısını göstermektedir. $Gecikme$ ise alarmın gerçek bilgiye göre ne kadar geç üretildiğini vermektedir.

Tablo 4.1'de GoDec-II yönteminin sonuçları verilmiştir. GoDec-II algoritması, PV_Hard videosu hariç tüm deneylerde tespit etmesi gereken olayı tespit edebilmiştir. Bunun yanında yalnızca bir adet yanlış alarmı kısa bir süre boyunca üretmiştir. CDW-2012 videosunun sonuçlarından, GoDec-II algoritmasının gece ve dış ortamlarda da iyi performans gösterdiği görülmektedir.

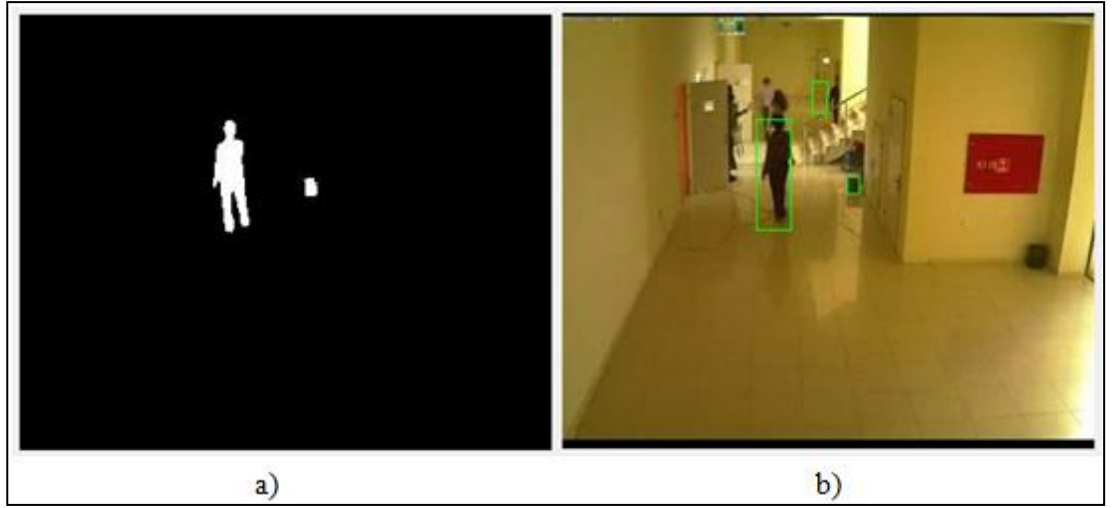
GoDec-II algoritması GTU veri setindeki yaklaşık 64 bin karelik videolarda da test edilmiştir. Bu videolar, sahnenin kalabalık olduğu ve olmadığı, başlangıçta hareketli nesnelerin var olduğu, çeşitli tipte ve sayıda nesnelerin bulunduğu, açık/kapalı alanlarda kaydedilmiştir. Park edilmiş araç senaryoları ile test senaryoları çeşitlendirilmiştir. Algoritmanın bu veri setindeki sonuçları Tablo 4.1'de görülmektedir. Mevcut 14 olaydan, 13'ü tespit edilmiştir. Bunun yanında, toplamda 4 adet hatalı tespit mevcuttur ve bu hatalı tespitlerin süreleri oldukça kısadır.

Tablo 4.1: GoDec-II Algoritmasının Performans Sonuçları.

Gerçek Bilgi (Ground Truth)					Tespit Sonuçları			
Veri Seti	T	T _{olay}	Olay	DT	HT	Gecikme	T _D	T _H
AVSS_Easy	5222	2870	1	1	0	470	2400	0
AVSS_Medium	4584	3100	1	1	1	550	700	230
AVSS_PV_Easy	5040	2200	1	1	0	300	1150	0
AVSS_PV_Med	3496	2100	1	1	0	230	1100	0
AVSS_PV_Hard	4109	2100	1	0	0	-	-	-
PETS2006-1	3021	1101	1	1	0	259	842	0
PETS2006-2	3021	1101	1	1	0	273	828	0
CDW-2012	3000	1050	1	1	0	347	628	0
Visor	3001	1740	1	1	0	343	1397	300
GTU_Dataset	64069	-	14	13	4	-	-	800

Tablo 4.2’de Wang ve arkadaşları [12]’de önerilen yöntemin sonuçları verilmiştir. Bu yöntem, arka planın en başta boş olmasına bağlı olduğundan ve i-Lids videolarında bu durum sağlanmadığından dolayı video boyunca bir hatalı tespiti bulunmaktadır. Yanında (*) işareti bulunan veriler, aynı tespit için kesintili olarak alarm üretildiğini göstermektedir. Ayrıca, CDW-2012 videosunda, bu algoritma bir sonuç vermemiştir.

Şekil 4.13’te GTU veri setine ait bir görüntü gösterilmiştir. Sol tarafta statik nesne maskesi ve sağda ise tespit sonucu verilmiştir. Tespit sonucunda üç bölge işaretlenmiştir. Durağan bir insan, bir nesne ve boş bir bölge görülmektedir. Durağan insan ve nesne statik nesne maskesinde görülmekte fakat boş bölge görülmemektedir. O bölgedeki tespit, statik nesne maskesinde kaybolmuştur. Fakat Nesne İşleme modülünde, küçük hareketlerden ötürü oluşacak tutarsız tespitleri önlemek için, nesne kaybolduktan bir süre sonra tespit silinmektedir. Bu yüzden, statik nesne maskesinde görülmeyişi halde, sonuç görüntüsünde hala belirmektedir.



Şekil 4.13: GTU Veri Setinden Örnek Bir Kare, a) Statik Nesne Maskesi, b) Tespit Sonucu.



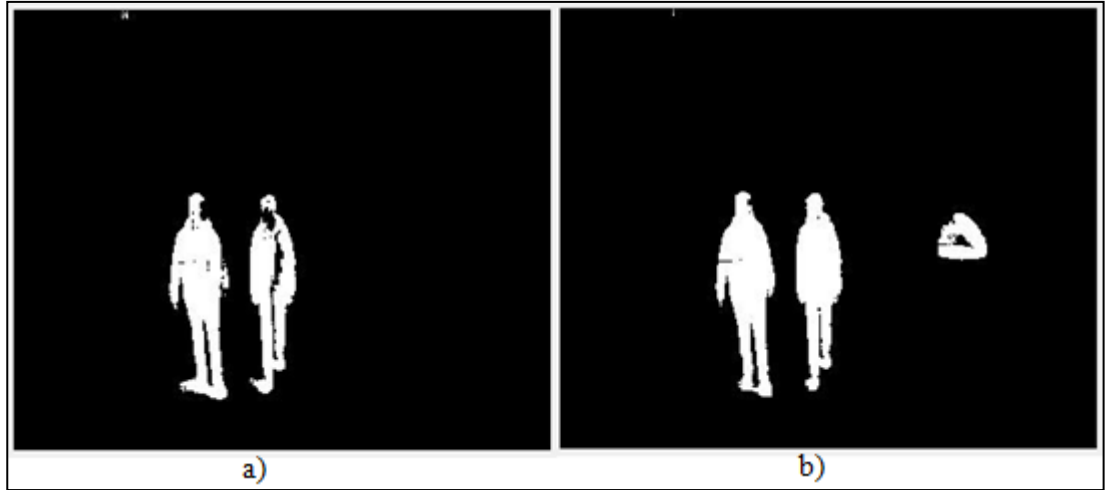
Şekil 4.14: GTU_01 Videosundan Örnek Görüntü.

Şekil 4.14'te GTU veri setine ait bir başka videodan örnek kare verilmiştir. Bu örnekte, sağ tarafta ve sol tarafta birer nesne işaretlenmiştir. Sağ taraftaki nesne, statik nesne maskesinde kaybolmuştur. Şekil 4.13'teki aynı sebepten dolayı, sonuç görüntüsünde henüz tespit kaldırılmamıştır. Bunun yanında, statik nesne maskesinde görünen yavaş yürüyen şahıs, sonuç görüntüsünde işaretlenmemiştir. Bunun sebebi, statik nesne maskesinde henüz alarm eşik değerine ulaşacak kadar kalmamasıdır.



Şekil 4.15: GTU_z23 Videosundan Örnek Bir Kare.

Şekil 4.15'te yine GTU veri setinden farklı bir videoya ait örnek kare verilmiştir. Sol tarafta statik nesne maskesi, sağ tarafta ise tespit sonucu görülmektedir. Bırakılan şemsiye tespit edilmiştir. Sahnede hareket halindeki iki kişi, statik nesne maskesinde görülmemektedir. Şekil 16'da ise aynı sahnede üretilen kısa ve uzun süreli ön plan maskeleri gösterilmiştir. Hareket halindeki bölgeler iki ön plan maskesinde de bulundukları için, şüpheli nesne hipotezini sağlamamaktadırlar.



Şekil 4.16: GTU_z23 a) Kısa Süreli ve b) Uzun Süreli Ön Plan Maskeleri.

Tablo 4.2: Wang ve Liu Tarafından Önerilen Yöntemin Performans Sonuçları.

Gerçek Bilgi (Ground Truth)					Tespit Sonuçları			
Veri Seti	T	T _{olay}	Olay	DT	HT	Gecikme	T _D	T _H
AVSS_Easy	5222	2870	1	1	1	230	2640	4800
AVSS_Medium	4584	3100	1	1	4	200	3100*	4000*
AVSS_PV_Easy	5040	2200	1	1	2	200	2100	300
AVSS_PV_Med	3496	2100	1	0	*	-	-	3496*
AVSS_PV_Hard	4109	2100	1	0	*	-	-	4109*
PETS2006-1	3021	1101	1	1	0	301	800	0
PETS2006-2	3021	1101	1	1	0	321	780	50
CDW-2012	3000	1050	1	1	1	150	290	-
GTU_Dataset	64069	-	14	10	6	-	-	2000

Tablo 4.3'te Porikli ve arkadaşları [49]'da önerilen yöntemin sonuçları verilmiştir. Bu yöntem de yalnızca tek bir yanlış alarm üretmektedir. Fakat bu yanlış alarm, önerilen yönteme göre yaklaşık 4 katı kadar uzun sürmektedir.

Tablolardan da görüldüğü gibi, GoDec-II algoritması diğer iki yönteme göre daha iyi sonuç vermiştir. Tespit edilecek tüm olayları yakalarken, oldukça az yanlış alarm üretmektedir. Yanlış tespitin gerçekleştiği kare sayısı da diğer yöntemlere göre oldukça azdır. Algoritmalar çalıştırılırken, şüpheli nesnenin tespit edilmesi için 200 karelik bir eşik değeri kullanılmıştır. Bir nesne tespit edildiğinde, Şekil 3.2'deki Nesne İşleme modülünde, o tespit 200 kare boyunca gerçekleşirse alarm üretilmeye başlanmakta, daha kısa tespitler için alarm üretilmemektedir.

Tablo 4.3: Porikli Tarafından Önerilen Yöntemin Performans Sonuçları.

Gerçek Bilgi (Ground Truth)					Tespit Sonuçları			
Veri Seti	T	T _{olay}	Olay	DT	HT	Gecikme	T _D	T _H
AVSS_Easy	5222	2870	1	1	0	540	1950	0
AVSS_Medium	4584	3100	1	1	1	1620	1860	870
AVSS_Hard	5060	3250	1	1	2	980	2000	480
AVSS_PV_Med	3496	2100	1	1	1	400	1600	70
PETS2006-1	3021	1101	1	1	0	201	900	0
PETS2006-2	3021	1101	1	1	0	231	870	10

Bu verilere bakarak, GoDec-II algoritmasının, diğer iki yöntemden doğru tespit ve hatalı tespit bakımından üstün olduğu söylenebilir. Ayrıca, alarm üretmedeki gecikme bakımından da en az diğer iki yöntem kadar iyi, bazı videolar için ise daha iyi olduğu görülmektedir.

4.3.1. Wang ve Liu Tarafından Önerilen Yaklaşım

Wang ve Liu [12]'de önerilen yaklaşım, Porikli ve arkadaşları [49] gibi ikili arka plan modelini temel alan yaklaşımdır. Farklı olarak, Porikli [49]'da arka plan modelini Gauss katmanlarının birleşimi olarak modelleyip, arka planı güncellemek için Bayes güncellemesi önermiştir. Wang ise [12]'de iki arka planı da ilk video karesi ile başlatıp, hareketli ortalama ile güncellemektedir. Herhangi bir piksel, ön plan durumunda kesintisiz olarak belirli bir süre kaldıktan sonra tamamen güncellenmektedir. Bu şekilde iki yapıyla Wang, iki farklı hızda örneklenmiş ön plan maskesi kullanarak bir güven haritası oluşturmakta ve bu maske üzerinden şüpheli nesne tespitini yapmaktadır. Yöntemin ayrıntılı teorik incelemesi 2. Bölümde yapılmıştır.



Şekil 4.17: AVSS_AB_Easy Videosunun İlk Karesi.

[12]'deki problem, arka planın başlangıç değerinden kaynaklanmaktadır. Şekil 4.17'de verilen görüntü, 'AVSS_AB_Easy' videosundan alınan ilk kareyi göstermektedir. Şekil 4.18 soldaki görüntü, hızlı güncellenen arka plan, sağdaki ise

yavaş güncellenen arka planın 1240 numaralı karedeki örnekleridir. Arka planın en başta ilk video karesi üzerinden başlatılması, bu karede de arka plana ait olmayan nesnelerin mevcut olması, arka plan modelini bozmaktadır. Asıl arka planın öğrenilme süresi artmaktadır. Bu durum da, hatalı tespitlere sebep olmaktadır. Aynı karede, tespit edilen nesneler Şekil 4.19’da görülmektedir.

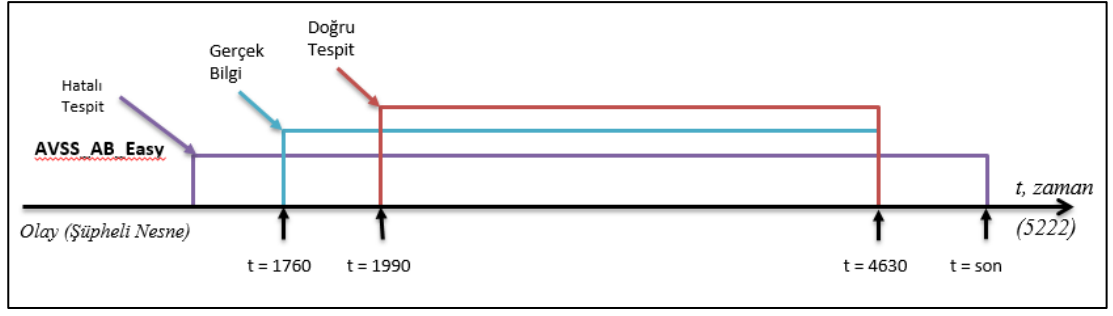


Şekil 4.18: a) Sık Güncellenen Arka Plan, b) Yavaş Güncellenen Arka Plan.



Şekil 4.19: Kare No: 1240 İçin Tespit Görüntüsü.

Şekil 4.19’da sahnenin orta kısmında görülen iki adet çerçeve, videonun ilk karesinde sahnede olan iki insanın arka plan olarak varsayılmasından dolayı oluşmaktadır. Ayrıca, [12]’de önerilen yöntemde tespit perdelemeyen oldukça fazla etkilenmektedir ve arka planın hareketli ortalama ile güncellenmesi, arka plan modelini Sigari ve arkadaşları [10]’da bahsedildiği üzere kirletmektedir.



Şekil 4.20: Wang ve Liu Tarafından Önerilen Yaklaşımın Nesne Tabanlı Metriğe Göre Tespit Sonucu.

Arka planın ilk video karesi ile başlatıldığı yaklaşımda, en başta arka planın tamamen durağan olduğu varsayımı yapılmaktadır. Dolayısıyla, bu varsayım karşılanmadığı takdirde arka plan modeli güncellenirken fazlaca bozulmaya uğramakta ve bu da performansı olumsuz etkilemektedir. Bu durum, öğrenme hızı artırılarak daha çabuk üstesinden gelinebilecek bir problem olarak görülebilir. Ancak öğrenme hızının kısıtlanabildiği durumlar da mevcut olabilecektir. Şekil 4.20’de [12]’de önerilen yöntemin performansı grafiksel olarak gösterilmektedir. Mor renk ile gösterilen alan, yanlış alarm, yani şüpheli nesne olmadığı halde öyle tespit edilen durumu göstermektedir. Tüm video süresince gösterilmesinin sebebi, video boyunca, başlangıç karesindeki hareketli nesnelerin varlığından dolayı, arka plan modelinin bozulması ve sürekli statik bölge tespit edilmesidir.

Bunun dışında, doğru tespit, 2200 numaralı karede başlamakta ve nesne kaldırılana kadar alarm devam etmektedir. Video başlangıç karesi sadece arka planı içerdiğinde, bu yaklaşım daha iyi sonuçlar vermektedir. GoDec-II algoritması ise, aynı video üzerindeki tespiti 2230 numaralı karede, yani 9 saniye daha geç yapmakta fakat hatalı tespit bulunmamaktadır.

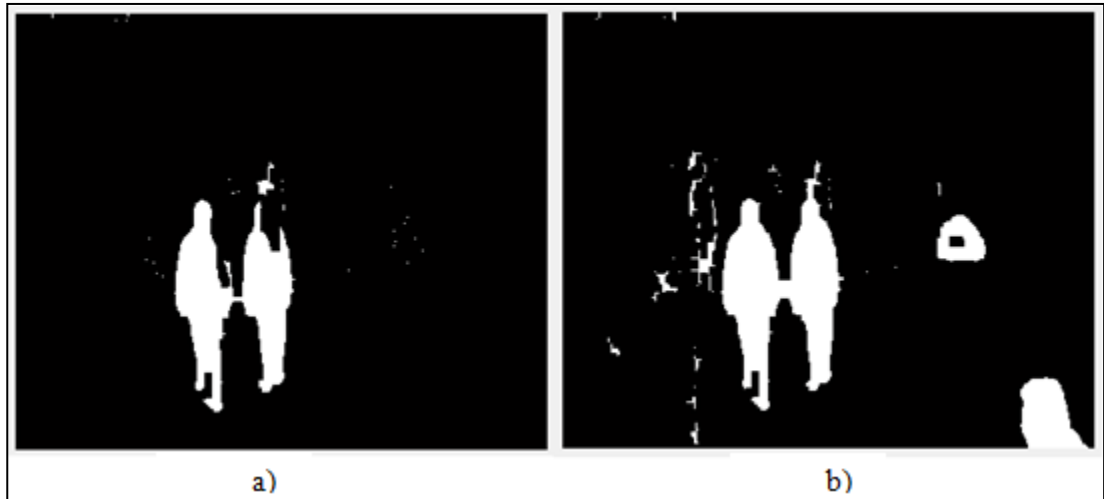
CDW-2012 gece videosunda ise bu algoritma herhangi bir olay tespit edememiştir. Ayrıca, sahnenin sağ tarafında bulunan ışık kaynağından da oldukça etkilenmektedir. Şekil 4.21’de solda bu algoritmanın 2495 numaralı karede verdiği Statik Nesne Maskesi görülmektedir. Şekil 4.21’de sağdaki görüntü ise üretilen alarmları göstermektedir. Şekil 4.21 soldaki görüntü, GoDec-II algoritması için Şekil 4.12 soldaki görüntünün eşdeğeridir. Sahnenin sağ tarafındaki ışık kaynağı hatalı olarak tespit edilmiştir.



Şekil 4.21: CDW-2012 Gece Videosu İçin Wang ve Liu Tarafından Önerilen Yöntem,
a) Statik Nesne Maskesi, b) Sonuç Görüntüsü.

Sahnenin sağındaki ışık kaynağı, bu algorithmada ön plan olarak tespit edilmiştir. Fakat aynı bölge, GoDec-II algoritması ile hatalı olarak tespit edilmemektedir.

Şekil 4.22’de görüldüğü üzere, Wang ve Liu [12]’de önerilen yöntemde kullanılan morfolojik operatörlerden dolayı, nesnelerin detayları kaybolmaktadır. Aynı sahnede GoDec-II algoritmasının sonucu Şekil 4.16’da görülmektedir. İki şekil karşılaştırıldığında, nesnelerin detayları bakımından GoDec-II’nin gözle görülür iyileştirme sağladığı görülmektedir.



Şekil 4.22: GTU_z23 İçin a) Wang ve Liu Tarafından Önerilen Yöntem, b) Kısa Süreli ve Uzun Süreli Ön Plan Maskesi.

4.3.2. Porikli ve Arkadaşları Tarafından Önerilen Yaklaşım

Porikli ve arkadaşları [49]’de şüpheli nesne tespiti, istatistiksel yöntemlerle ele alınmıştır. Modelde, her piksel çok değişkenli Gauss dağılımı ile modellenmektedir. Arka plan parametreleri Bayes yöntemi ile güncellenmekte ve bu güncelleme farklı hızlarda yapılarak biri uzun, diğeri kısa süreli iki arka plan modeli oluşturulmaktadır. Her iterasyonda, Gauss katmanlarından yalnızca bir tanesi, Mahalanobis mesafesine göre modele en yakın katman güncellenmektedir. Böylece, katmanlar arasında en az örtüşme olması sağlanmaktadır.

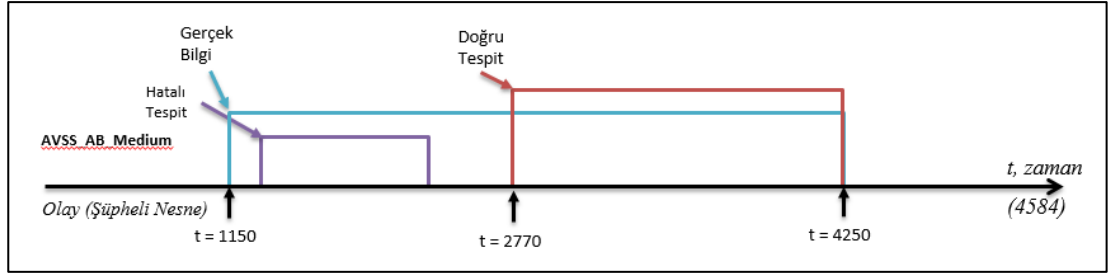
Bu yöntemde elde edilen sonuçların gerek tespit süresi gerekse tespit kalitesi ve yanlış alarm miktarı bakımından literatürdeki diğer yöntemlere göre oldukça iyi olduğu görülmüştür. Nesneleri, oldukça isabetli bölütleyebilmektedir. [49]’deki çalışmada, test için kullanılan videolarda tespit bölgelerinin el ile işaretlendiği, yani tüm sahne yerine sadece işaretlenen bölgelerde tespit yapıldığı belirtilmiştir. Bazı videolarda da başlangıç boş sahneyi içeren birkaç kare ile değiştirilmiştir. Bu şekilde, arka plan modelinin daha iyi öğrenilmesi sağlanmıştır.



Şekil 4.23: Porikli Tarafından Önerilen İkili Ön Plan Yönteminin AB_Easy Videosunda Sonucu.

AVSS_AB_Easy videosu için, GoDec-II algoritması şüpheli nesne tespitini Porikli ve arkadaşları [49]’de önerilen algoritmadan yaklaşık 2 saniye daha erken yapmaktadır.

AVSS_AB_Medium videosu için, GoDec-II, Porikli ve arkadaşları [49]’e göre 22 saniye daha erken tespit etmekte ve hatalı tespit süresi de yaklaşık 25 saniye daha kısadır.



Şekil 4.24: Porikli Tarafından Önerilen İkili Ön Plan Yönteminin AB_Medium Videosunda Sonucu.

Şekil 4.22 ve Şekil 4.23'te [49]'de önerilen yöntemin i-Lids veri setindeki AB_Easy ve AB_Medium videoları üzerindeki sonuçları grafiksel olarak gösterilmiştir.

4.3.3. Sigari ve Arkadaşları Tarafından Önerilen Arka Plan Çıkarımı

Farklı arka plan modelleri ve güncelleme mekanizmaları, şüpheli nesne tespiti algoritmasında kullanıldığında, genel performansı farklı şekilde etkilemektedir. Arka plan modelinin güncellenmesi ve arka plan çıkarımı konusunda en yaygın kullanılan hareketli ortalama yönteminin eksikleri ve buna alternatif olarak Bulanık Hareketli Ortalama ve Bulanık Arka Plan Çıkarımı yöntemi Sigari ve arkadaşları tarafından [10]'da önerilmiştir. Bulanık Arka Plan Çıkarımı geleneksel yöntemle göre daha iyi sonuç vermekte fakat bir miktar daha yavaş çalışmaktadır. Şekil 4.23 ve Şekil 4.24, sırasıyla, normal ve bulanık yaklaşımların PETS2001 veri seti üzerindeki test sonuçları gösterilmiştir. Şekillerden de anlaşıldığı üzere, bulanık yaklaşım gürültüye karşı daha etkilidir ve nesneleri de iyi bir şekilde bölütlemektedir. Kullanılan videoda zaman zaman ani ışık değişimi, parlamalar oluşmaktadır.

Şekil 4.24 ve Şekil 4.25'teki görüntüler, şüpheli nesne tespiti için bir ara adım olan arka plan çıkarımı ve arka planın güncellenmesi için iki farklı yöntemin sonucunu göstermektedir. Şekil 4.26 ise GoDec algoritmasının aynı sahnedeki sonucunu göstermektedir.



Şekil 4.25: Hareketli Ortalama ve Arka Plan Çıkarımı, PETS2001 Sonucu.



Şekil 4.26: Bulanık Hareketli Ortalama ve Bulanık Arka Plan Çıkarımı, PETS2001 Sonucu.



Şekil 4.27: PETS2001 Üzerinde GoDec İle Ön Plan Tespiti Sonucu.

4.4. GoDec-II'nin Avantajları/Dezavantajları

Çalışma kapsamında, literatürde en yaygın kullanılan şüpheli nesne tespiti yöntemleri ile bu çalışmada önerilen yaklaşım, nesne tabanlı performans metriği ile karşılaştırılmıştır.

Şüpheli nesne problemi, farklı çalışmalarda farklı amaçlara hizmet edecek şekilde tanımlanabilmektedir. Bu çalışmada ise, probleme, operatör (kullanıcı) odaklı bir çözüm önerilmiştir. Bu açıdan bakıldığında, önerilen yaklaşım literatürdeki yaygın yöntemler kadar iyi sonuç vermekte ve hatta bazı yönlerden de daha üstün gelmektedir.

Özellikle başlangıç değerleri için, hemen hemen tüm yaklaşımlar görüntülenen sahnenin sadece arka planı yani hareketsiz bölgeleri gösterdiğini varsaymaktadır. Algoritma da arka planı bu varsayım üzerinden öğrenmeye başlamaktadır. Varsayım sağlanmadığında ise incelenen yöntemlerde yanlış alarm oranının arttığı gözlemlenmiştir. Önerilen yaklaşımda ise böyle bir varsayım bulunmadığı için, performans başlangıç değerlerinden yani sahnenin boş olmaması durumundan daha az etkilenmektedir.

Bunun sebebi, yaklaşımın RPCA tabanlı olmasıdır. Altuzay öğrenmesi algoritmaları, örneğin 200 karelik kısa bir süre için dahi, piksellerin baskın durumlarını (arka plan) öğrenebilmektedir. Bu durum da arka plan piksellerinin daha kolay öğrenilmesini sağlamaktadır.

Özellikle Gauss Karışımı yöntemini temel alan ve renk bilgisi kullanan yaklaşımlarda cisimlerin gölgeleri de tespit edildiği için, tespit performansı düşmektedir. Matris ayrıştırması sonucunda ise, Şekil 4.5 ve Şekil 4.6’da görülebileceği üzere, cisimlerin gölgesi tespit performansını etkilememektedir. Ayrıca, cisimlerin genel hatlarını hassaslıkla tespit ettiği için, farklı sınıfta cisimlerin birbirinden ayırt edilmesini de kolaylaştırmaktadır. Bu durum, Şekil 4.3’teki bavul ve Şekil 3.4’teki insan figürlerinden açıkça görülebilmektedir. Wang ve Liu [12]’de önerilen yöntemde ise kullanılan morfolojik işlemler sonucunda tespit edilen bölgelerin hatları bozulmaktadır. Bu da farklı şekilde nesneleri ayırt etmede zorluk çıkarabilmektedir. Şekil 4.5’te görüldüğü gibi, GoDec algoritması, ufak cisimleri dahi detayları kaybetmeden tespit edebilmektedir.

Önerilen yöntem, genel işlem yükü bakımından, standart arka plan çıkarımı yöntemlerinden daha yüksek, GMM gibi yöntemlerden daha düşük işlem gücü gerektirmektedir. Yaklaşımın genel yükü, GoDec algoritması ile ön plan tespitinden kaynaklanmaktadır. Altuzay öğrenmesi yöntemleri, nispeten yeni ele alınan yöntemlerdir.

GoDec ile ön plan tespitinde bu çalışmada Otsu’nun yöntemi ile eşikleme yapılmıştır. Bu yöntem, kullanılan veri setleri için tatmin edici sonuçlar vermektedir. Farklı eşikleme algoritmaları ile algoritmanın performansı daha iyileştirilebilir.

Altuzay yaklaşımı için literatürde henüz artımlı bir yöntem sunulamamıştır. Belirli sayıda video karesinin yığın (batch) olarak biriktirilip işlenmesine dayalı olduğundan, gerçek zamanlı çalışması için yüksek işlem gücü gerektirmektedir. Önerilen yaklaşım video dizilerini, ortalama 10-15 fps hızında (intel Xeon 3.3 GHz işlemci, 16 GB RAM) işleyebilmektedir.

Şüpheli nesnenin ne kadar çabuk tespit edileceği, önerilen yöntem de dâhil, genelde her yöntemde bir parametreye bağlı olarak belirlenebilmektedir. Bu yüzden, adil bir karşılaştırma yapabilmek için, ilk tespit anından itibaren alarm üretmek için 200 karelik bir bekleme süresi kullanılmıştır.

5. SONUÇ VE GELECEK ÇALIŞMALAR

Bu tez çalışmasında bilgisayarla görme konusunda şüpheli nesne tespiti problemi ele alınmış, daha önce bu problemde hiç kullanılmamış olan altuzay öğrenmesi yaklaşımı ile yeni bir yöntem önerilmiştir.

Literatür araştırması sonucunda seçilen bazı yöntemler, kullanılabilecek veri setleri ve farklı yöntemleri karşılaştırmak için performans metrikleri incelenmiştir. Bunun yanında, önerilen yöntem literatürde şüpheli nesne tespiti için geliştirilmiş mevcut algoritmalarla karşılaştırılarak önerilen yöntemin güçlü ve zayıf yönleri belirlenmiştir. Bu yöntemin, video işlemedeki genel sorunlara (perdeleme, ani ışık değişimi) karşı etkili bir çözüm sunduğu deneysel sonuçlarla erişime açık bazı veri setleri üzerinde gösterilmiştir.

Yöntem, arka plan modelinin başlangıç durumundan etkilenmemektedir. Yani, GMM, Hareketli Ortalama gibi yöntemler başlangıçta sahnenin yalnızca durağan bölgeden oluşmasını gerektirirken, altuzay öğrenmesinin etkisinden dolayı GoDec-II yöntemi başlangıçta sahnede bulunan hareketli nesnelerden etkilenmemektedir. Geleneksel yöntemlerde arka plan modelinin oluşturulması esnasında kullanılarak farklı bir yaklaşım uygulanabilir.

Dinamik arka plan bölgeleri de geleneksel yöntemler için başka bir problemidir ve genelde hatalı tespitlere yol açabilmektedir. GoDec-II algoritmasının, dinamik arka plan hareketine karşı, bazı videolar için, etkili sonuç verdiği görülmüştür.

Ayrıca, GoDec-II yaklaşımı hesaplama yükü bakımından makul bir performans vermektedir. Yapılacak iyileştirmeler ile gerçek zamanlı çalışabilecek hale getirilerek bir video gözetleme sistemine entegre edilebilir.

KAYNAKLAR

- [1] Tian Y. L., (2008), “Real Time Detection of Abandoned and Removed Objects in Complex Environments”, International Workshop on Visual Surveillance-VS2008, Marseille, France, 17 Ekim.
- [2] Li Q., Mao Y., Wang Z., Xiang W., (2009), “Robust Real-Time Detection of Abandoned and Removed Objects”, Image and Graphics, 2009. ICIG '09. Fifth International Conference, 156-161, Shanxi, China, 20-23 Eylül.
- [3] Porikli F., (2007), “Detection of Temporarily Static Regions by Processing Video at Different Frame Rates”, Advanced Video and Signal Based Surveillance, AVSS, 236-241, London, UK, 5-7 Eylül.
- [4] Guler S., Farrow M. K., (2006), “Abandoned Object Detection in Crowded Places”, PETS Workshop, 99-106, New York, USA, 18-23 Haziran.
- [5] Web 1, (2014), <https://sites.google.com/site/backgroundsubtraction/contact>, (Erişim Tarihi: 13/03/2015)
- [6] Loza A., Mihaylova L. , Canagarajah N., Bull D., (2006), “Structural Similarity-Based Object Tracking in Video Sequences”, Information Fusion 2006 9th International Conference, 1-6, Florence, Italy, 10-13 Temmuz.
- [7] Thacker N., Aherne F., Rockett P., (1997), “The Bhattacharyya Metric as an Absolute Similarity Measure for Frequency Coded Data”, Kybernetika, 32(4), 1-7.
- [8] Russell S., Norvig P., (2003), “Artificial Intelligence: A Modern Approach, 2nd Edition”, Pearson Education Inc..
- [9] Dedeoğlu Y., (2004) , “Moving Object Detecion, Tracking and Classification for Smart Video Surveillance”, Master's Thesis, Bilkent University.
- [10] Sigari M. H., Mozayani N., Pourreza H. T., (2008), “Fuzzy Running Average and Fuzzy Background Subtraction: Concepts and Applications”, IJCSNS International Journal of Computer Science and Network Security, 8(2), 138-143.
- [11] Zin T. T., Tin P., Toriu T., Hama H., (2012), “A Probability Based Model for Detecting Abandoned Objects in Video Surveillance Systems”, Proceedings of the World Congress on Engineering, London, UK, 4-6 Temmuz.
- [12] Wang W., Liu Z., (2010), “A New Approach For Real Time Detection of Abandoned and Stolen Objects”, Electrical and Control Engineering (ICECE), 2010 International Conference, 128-131, Wuhan, China, 25-27 Haziran.

- [13] Grimson C. ve Stauffer L., (1999), "Adaptive Background Mixture Models for Real-Time Tracking", Proc. CVPR99, 2, 2246 - 2252, Ft. Collins, USA, 23-25 Haziran.
- [14] Li L., Huang W., Gu I. Y.-H., Tian Q., (2004), "Statistical Modeling of Complex Backgrounds for Foreground Object Detection", IEEE Transactions on image processing, 13(11), 1459-1472.
- [15] Bouwmans T., (2014), "Traditional and recent approaches in background modeling for foreground detection: An overview", ScienceDirect, 11, 31-66.
- [16] Bayona A., SanMiguel J. ve Martinez J., (2009), "Comparative Evaluation of Stationary Foreground Object Detection Algorithms Based on Background Subtraction", Advanced Video and Signal Based Surveillance 6th IEEE Conference, 25-30, Genova, Italy, 2-4 Eylül.
- [17] Stauffer C., Eric W. ve Grimson L., (2000), "Learning Patterns of Activity Using Real-Time Tracking", IEEE Transactions Pattern Analysis Machine Intelligence, 22(8), 747-757.
- [18] Gong S. ve McKenna J., (1999), "Tracking Colour Objects Using Adaptive Mixture Models", Image Vision Computing, 17, 225-231.
- [19] Ohta N., (2001), "A Statistical Approach to Background Suppression for Surveillance Systems", Proceedings of IEEE Int. Conference on Computer Vision, 481-486, Vancouver, Canada, 7-14 Temmuz.
- [20] Collins R., Lipton A. ve Kanade T., (1999), "A System for Video Surveillance and Monitoring", Proc. American Nuclear Society (ANS) 8th Int. Topical Meeting on Robotic and Remote Systems, Pittsburgh, USA, 25-29 Nisan.
- [21] Wonjun K. ve Cangick K., (2012), "Background Subtraction for Dynamic Texture Scenes Using Fuzzy Color Histograms", IEEE Signal Processing Letters, 19(3), 127-130.
- [22] Web 2, (2010). <https://github.com/andrewssobral>, (Erişim Tarihi: 10/03/2015).
- [23] Candes E., Li X., Ma Y. ve Wright J., (2011), "Robust Principal Component Analysis?", Journal of the ACM, 3(58), 1-37.
- [24] Lin Z., Chen M. ve Ma Y., (2011), "The Augmented Lagrange Multiplier Method for Exact Recovery of Corrupted Low-Rank Matrices", NIPS 2011, Granada, Spain, 12-17 Aralık.
- [25] Bucak S. S. ve Günsel B., (2009), "Gözetleme Videolarında Artımlı Negatif Olmayan matris Ayırıştırma ile Arkaplan Modelleme", IEEE 15. İsalet İşleme ve İletişim Uygulamaları Kurultayı (SIU), Eskişehir/Türkiye, 11-13 Haziran.

- [26] Kutz J. G. ve Nathan J., (2014), "Dynamic Mode Decomposition for Real-Time Background/Foreground Separation in Video", Computing Research Repository, arXiv:1404.7592.
- [27] Zhou T. ve Tao D., (2011), "GoDec: Randomized Low-rank & Sparse Matrix Decomposition in Noisy Case", Proceedings of the 28th International Conference on Machine Learning, Washington, USA, 28 Haziran - 2 Temmuz.
- [28] Kutz J. G., Nathan J., (2014), "Dynamic Mode Decomposition for Real-Time Background/Foreground Separation in Video", Cornell University.
- [29] Dedeoğlu Y., (2004), "Moving Object Detection, Tracking and Classification for Smart Video Surveillance", Master's Thesis, Bilkent University.
- [30] Li L., Huang W., ve Yu-Hua G. I., (2004), "Statistical Modeling of Complex Backgrounds for Foreground Object Detection", Ieee Transactions On Image Processing, 13(11), 1459-1472.
- [31] Haritaoglu I., Harwood D. ve Davis L., (2000), "W4: real-time surveillance of people and their activities", Pattern Analysis and Machine Intelligence, IEEE Transactions on, 8(22), 809-830.
- [32] Wang L., Hu W. ve Tan T., (2003), "Recent developments in human motion analysis", Pattern Recognition, 36(3), 585-601.
- [33] McIvor A. M., (2000), "Background Subtraction Techniques", Procession of Image and Vision Computing, Proc. of Image and Vision Computing, 4, 3099-3104.
- [34] El Baf F., Bouwmans T. ve Vachon B., (2008), "A Fuzzy Approach to Background Subtraction", Image Processing, 2008. ICIP 2008. 15th IEEE International Conference, 2658-2651, San Diego, USA, 12-15 Ekim.
- [35] Nascimento J. ve Marques J., (2005), "Performance evaluation of object detection algorithms for video surveillance", IEEE Transactions on Multimedia, 8(4), 761-774.
- [36] Culibrk D. ve Marques O., (2007), "Neural Network Approach to Background Modeling for Video Object Segmentation", IEEE Transactions On Neural Networks, 18(6), 1614-1627.
- [37] Toyama K., Krumm J., Brumitt B. ve Meyers B., (1999), "Wallflower: Principles and Practice of Background Maintenance", Computer Vision, 1999. The Proceedings of the 7th IEEE International Conference, 255-261, Kerkyra, Greece, 20-27 Eylül.

- [38] Wren C., Azarbayejani A., Darrell T. ve Pentland A., (1997), “Pfinder: real-time tracking of the human body”, Pattern Analysis and Machine Intelligence, IEEE Transactions, 7(19), 780-785.
- [39] Lin H., Liu T. ve Chuang J., (2002), “A Probabilistic SVM Approach for Background Scene Initialization”, Image Processing. 2002. Proceedings. 2002 International Conference, 893-896, Rochester, USA, 22-25 Eylül.
- [40] Eng H., Wang J., Kam A. ve Yau W., (2004), “Novel Region-based modeling for human detection within high dynamic aquatic environment”, Proc. CVPR, 390–397, Ohio, USA, 27 Haziran - 2 Temmuz.
- [41] Elgammal A., Harwood D. ve Davis L., (2000), “Non-parametric Model for Background Subtraction”, Computer Vision—ECCV 2000, 751-767, Dublin, Ireland, 26 Haziran - 1 Temmuz.
- [42] El Baf F., Bouwmans T. ve Vachon B., (2009), “Fuzzy Statistical Model of Dynamic Backgrounds for Moving Object Detection in Infrared Videos”, Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009. IEEE Computer Society Conference, 60-65, Miami, USA, 20-21 Haziran.
- [43] Candes E. J., Li X., Ma Y. ve Wright J., (2009), “Robust Principal Component Analysis”, Journal of the ACM (JACM), 58(3), 11.
- [44] Roweis S., (1998), “EM Algorithms for PCA and SPCA”, NIPS, 626-632, Denver, USA, 30 Kasım - 5 Aralık.
- [45] Zhou Z., Li X., Wright J., Candes E. J. ve Ma Y., (2010), “Stable Principal Component Pursuit”, Information Theory Proceedings (ISIT), 2010 IEEE International Symposium, 1518-1522, Austin, USA, 13 - 18 Haziran.
- [46] Web 3, (2014), <https://sites.google.com/site/backgroundsubtraction/background-maintenance>.
- [47] Kailath T., (1967), “The Divergence and Bhattacharyya Distance Measures in Signal Selection”, Communication Technology, IEEE Transactions on, 1(15), 52-60.
- [48] Web 4, (2014), <https://sites.google.com/site/backgroundsubtraction/test-sequences>.
- [49] Porikli F., Ivanov Y. ve Haga T., (2008), “Robust Abandoned Object Detection Using Dual Foregrounds”, EURASIP Journal on Advances in Signal Processing, 2008, 30-41.
- [50] Tian Y.-l., Feris R., Hampapur A., (2008), “Real Time Detection of Abandoned and Removed Objects in Complex Environments”, International Workshop on Visual Surveillance-VS2008, Marseille, France, 17 Ekim.

- [51] Boulton T., Michaels R., Gao X. ve Eckmann M., (2001), "Into the Woods: Visual Surveillance of Non-Cooperative Camouflaged Targets", *Multimedia, IEEE Transactions on*, 8(4), 761-774..
- [52] Bhargava M., Chen C., Ryoo M. S. ve Aggarwal J. K., (2009), "Detection of Object Abandonment Using Temporal Logic", *Machine Vision and Applications*, 20, 271-281.
- [53] Bayona A., SanMiguel J. ve Martinez J., (2009), "Comparative Evaluation of Stationary Foreground Object Detection Algorithms Based on Background Subtraction", *Advanced Video and Signal Based Surveillance. AVSS '09. Sixth IEEE International Conference*, 25-30, Genova, Italy, 2-4 Eylül.
- [54] Petrosino L. ve Maddalena A., (2008), "A Self-Organizing Approach to Background Subtraction for Visual Surveillance Applications", *Ieee Transactions On Image Processing*, 17(17), 1168-1177.
- [55] Maddalena L. ve Petrosino A., (2010), "A Fuzzy Spatial Coherence-Based Approach to Background/Foreground Separation for Moving Object Detection", *Neural Computing & Applications*, 19, 179-186.
- [56] Kim K., Chalidabhongse T. H., Harwood D. ve Davis L., 2006, "PDR: Performance Evaluation Method for Foreground-Background Segmentation Algorithms", *EURASIP Journal on Applied Signal Processing*.
- [57] Dewan M. A. E., Hossain M. J. ve Chae O., (2008), "Moving Object Detection and Classification Using Neural Networks", *KES-AMSTA*, 152-161, Incheon., Korea, 27-28 Mart.
- [58] Campos L. C., SanMiguel J. C. ve Martinez J. M., (2011), "Discrimination of Abandoned and Stolen Objects Based on Active Contour", *8th IEEE International Conference on Advanced Video and Signal-Based Surveillance*, 101-106, Klagenfurt, Austria, 30 Ağustos - 2 Eylül.
- [59] Otsu N., (1979), "A Threshold Selection Method from Gray-Level Histograms", *IEEE Transactions on Systems, Man and Cybernetics*, 9(1), 62-66.
- [60] Bashir F. ve Porikli F., (2006), "Performance Evaluation of Object Detection and Tracking Systems", *CVPR 2006*, 7-14, NY, USA, 17-22 Haziran.
- [61] Goyette N., Jodoin P. M., Porikli F., Konrad J. ve Ishwar P., (2012), "changedetection.net: A new change detection benchmark dataset", *IEEE Workshop on Change Detection (CDW-2012) at CVPR 2012*, 1-8, Rhode Island, USA, 16-21 Haziran.
- [62] Vezzani R. ve Cucchiara R., (2010), "Video Surveillance Repository (VISOR): an integrated framework", *Multimedia Tools and Applications*, 50(2), 359-380.

- [63] Porikli F. ve Tüzel Ö., (2006), “Bayesian Background Modeling for Foreground Detection”, ACM Visual Surveillance and Sensor Networks, 55-58, Santa Barbara, USA, 23-27 Ekim.
- [64] Tian Y.-L., Lu M. ve Hampapur A., (2005), “Robust And Efficient Foreground Analysis for Real-Time Video Surveillance”, CVPR 2005, 1182-1187, San Diego, USA, 20-26 Haziran.

ÖZGEÇMİŞ

Hasan Hüseyin SÖNMEZ, 1989 yılında Çankırı’da doğdu. 2008 yılında Kuleli Askeri Lisesi’ni tamamladı ve ardından başladığı Gebze Yüksek Teknoloji Enstitüsü (GYTE), Mühendislik Fakültesi, Elektronik Mühendisliği Bölümü’nü 2013 yılında ikinci olarak tamamladı. Aynı yıl yüksek lisans eğitimine GYTE Elektronik Mühendisliği bölümünde başladı. 2014 yılından bu yana Gebze Teknik Üniversitesi Elektronik Mühendisliği Bölümü’nde araştırma görevlisi olarak Sinyal ve Görüntü İşleme alanında çalışmaktadır.