

**YAYA TESPİTİNDE KULLANILAN FARKLI ÇALIŞMALARIN DENEYSEL  
ANALİZİ**

**ABDULLAH ASIM YILMAZ**

**YÜKSEK LİSANS TEZİ**

**BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**

**TOBB EKONOMİ VE TEKNOLOJİ ÜNİVERSİTESİ**

**FEN BİLİMLERİ ENSTİTÜSÜ**

**MAYIS 2012**

**ANKARA**

Fen Bilimleri Enstitü onayı

---

Prof. Dr. Ünver Kaynak

Müdür

Bu tezin Yüksek Lisans derecesinin tüm gereksinimlerini sağladığını onaylarım.

---

Doç Dr. Erdoğan Dođdu

Anabilim Dalı Başkanı

Abdullah Asım YILMAZ tarafından hazırlanan YAYA TESPİTİNDE KULLANILAN FARKLI ÇALIŞMALARIN DENEYSEL ANALİZİ adlı bu tezin Yüksek Lisans tezi olarak uygun olduğunu onaylarım.

---

Yrd. Doç. Dr. M. Fatih Demirci

Tez Danışmanı

Tez Jüri Üyeleri

Başkan : Yrd. Doç. Dr. Murat ÖZBAYOĞLU

\_\_\_\_\_

Üye : Yrd. Doç. Dr. M. Fatih DEMİRCİ

\_\_\_\_\_

Üye : Yrd. Doç. Dr. Ali Cafer GÜRBÜZ

\_\_\_\_\_

## **TEZ BİLDİRİMİ**

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada orijinal olmayan her türlü kaynağa eksiksiz atıf yapıldığını bildiririm.

Abdullah Asım YILMAZ

**Üniversitesi** : TOBB Ekonomi ve Teknoloji Üniversitesi  
**Enstitüsü** : Fen Bilimleri  
**Anabilim Dalı** : Bilgisayar Mühendisliği  
**Tez Danışmanı** : Yrd. Doç. Dr. M. Fatih Demirci  
**Tez Türü ve Tarihi** : Yüksek Lisans – Mayıs 2012

**ABDULLAH ASIM YILMAZ**

**YAYA TESPİTİNDE KULLANILAN FARKLI ÇALIŞMLARIN DENEYSSEL  
ANALİZİ**

**ÖZET**

Bu tezdeki amaç video çerçeveleri arasındaki farkların analizine dayalı olarak yayaların izlenmesi, yayaların tespiti ve yaya hareketlerini izleyen videonun farklı algoritmalar denenerek oluşturulması ve daha önce yapılan farklı benzer çalışmaların deneysel olarak analizi ve bizim oluşturduğumuz algoritmayla karşılaştırılmasıdır. Bu işlem çekilen çeşitli çekilen videolar aracılığı yapıldı. Projemiz şu adımları izlemektedir; video renkli tonlardan gri tonlara çevrildi, gri tonlarda arka plan hesaplandı, videodaki her bir kareden arka plan çıkarıldı ve çıkan bu sonuçlar normalize edilerek kmeans algoritmasına göre ikili veriye çevrildi, bulunan bu karelerin belli alandan daha küçük alana ve gürültülere sahip olanları elenerek, kalanların içlerindeki boşluklarında morfolojik işlemler kullanılarak doldurulması işlemi ile yayaların videodaki pozisyonları tespit edildi. Buna ek olarak, yaya tespiti için farklı algoritmalar üretilmiş ve test edilmiştir. Ardından, daha önce yapılmış farklı benzer çalışmaların deneysel analizi yapılmış ve bizim algoritmamızla, daha önce yapılmış farklı benzer çalışmaların karşılaştırma işlemi yapılmıştır.

**Anahtar Kelimeler:** Yaya tespiti, Yayaların izlenmesi, Deneysel Analiz ve Karşılaştırma

**University** : TOBB Economics and Technology University  
**Institute** : Institute of Natural and Applied Sciences  
**Science Programme** : Computer Engineering  
**Supervisor** : Associate Professor Yrd. Doç. Dr. M. Fatih Demirci  
**Degree Awarded and Date** : M.Sc. – May 2012

**ABDULLAH ASIM YILMAZ**

**EXPERIMENTAL ANALYSIS OF DIFFERENT STUDIES FOR USING  
PEDESTRIAN DETECTION**

**ABSTRACT**

This thesis aim is monitoring of pedestrians, pedestrian detection, creation of pedestrian movements following the video with trying different algorithms based on difference analysis between video frames and experimental analysis of previous studies and comparison of our algorithm and previous studies. This project is made with image processing techniques for detecting pedestrians. This process was done with some different recording videos. The project includes these steps; converting rgb to grayscale, calculating background with grayscale, subtracting video frame to background for each video frame and kmeans algorithm to make threshold in the frame to convert it black and white frame(binary frame), some of regions to delete the noise and smaller than the obvious region on the frames, and doing morphologic operation for each videos frames. After performing all these steps over frames, the regions with the pedestrians were detected. Moreover, different algorithm is produced and tested for detecting pedestrians. After that, previous studies is done experimental analysis and comparison process with our algorithm.

**Keywords:** Pedestrian Detection, Pedestrian Monitoring, Experimental Analysis and Comparison

## **TEŐEKKÖR**

Çalıőmalarım boyunca deęerli yardım ve katkılarıyla beni yönlendiren hocam Yrd. Doç. Dr. M. Fatih Demirci'ye yine kıymetli tecrübelerinden faydalandığım TOBB Ekonomi ve Teknoloji Üniversitesi Bilgisayar Mühendislięi Bölümü öğretim üyelerine teőekkürü bir borç bilirim.

## İÇİNDEKİLER

ÖZET.....	iv
ABSTRACT.....	v
TEŞEKKÜR.....	vi
İÇİNDEKİLER.....	vii
ŞEKİLLERİN LİSTESİ.....	x
GRAFİKLERİN LİSTESİ.....	xiv
ÇİZELGELERİN LİSTESİ.....	xv
KISALTMALAR .....	xvi
1.GİRİŞ .....	1
1.1. Yaya Tespiti Sistemleri Uygulamaları .....	1
1.2. Tez Kapsamı.....	2
1.3. Tez Taslağı .....	4
2. LİTERATÜR TARAMASI.....	5
3.GÖRÜNTÜ İŞLEME(İMAGE PROCESSİNG) GENEL BİLGİLER .....	6
3.1. Görüntü İşleme (İmage Processing) Nedir, Yararları Nelerdir? .....	8
3.2. Bazı Görüntü İşleme Tanımları .....	10
3.2.1. Dijital Görüntü.....	10
3.2.2. Niteliklendirme .....	11
3.2.3. Görüntü Operasyonları .....	13

3.2.4. Komşuluk Çeşitleri .....	14
3.2.5. Filtreleme .....	15
3.2.6. Histogram.....	17
3.2.7. Maskeleye(Maskeler) .....	18
4. YÖNTEMİN UYGULANMASI.....	20
4.1. Video Okuma .....	21
4.2. Arka Planı Bulma .....	23
4.3. Yaya Pozisyonlarını Tespit Etmek .....	24
4.3.1. Muhtemel Yaya Pozisyonlarının Bulunması .....	24
4.3.2. Hataların Giderilmesi.....	29
4.4. Video Üzerinde Yayaların İşaretlenmesi .....	32
4.5. Yeni Yöntemler(Algoritmalar) Geliştirilmesi ve Bunların Denenmesi .....	33
4.5.1. Geliştirilen Yöntemler(Algoritmalar) .....	34
4.6. Daha Önce Yapılan Farklı Benzer Uygulamaların Bizim Tarafımızdan Araştırılması ve Gerçekleştirilmesi .....	36
4.6.1. Histogram of Oriented Gradients (HOG) .....	36
4.6.2. A Monocular Human Detection System Based on EOH and Oriented LBP Features .....	49
5. DENEYLER.....	59
5.1. Veri Setimiz Özellikleri ve Kullanılan Araçlar .....	59
5.2. Elde Edilen Sonuçların Grafıksel ve Deneysel Karşılaştırılması .....	60
5.3. Karşılaşılan Problemler.....	60



6. SONUÇLAR VE ÖNERİLER.....	69
6.1. Öneriler.....	69
6.2. Sonuç.....	70
KAYNAKLAR.....	71
ÖZGEÇMİŞ.....	74

## ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 1.1 Yaya Tespit Sistemi Akış Şeması.....	3
Şekil 3.1 Görüntü İşleme Tanımı.....	8
Şekil 3.2 Gürültü(Noise) Düzeltme Örneği.....	9
Şekil 3.3 Dijital Görüntü Örneği.....	10
Şekil 3.4 Piksel Gösterimi Örneği.....	11
Şekil 3.5 Niteliklendirme Örneği.....	12
Şekil 3.6 Görüntü İşlemleri.....	13
Şekil 3.7 Komşuluk Örneklemesi.....	14
Şekil 3.8 Filtreleme Örneği.....	15
Şekil 3.9 Filtreleme Matrisi Örneği .....	16
Şekil 3.10 Filtreleme Matrisi Örneği 2.....	16
Şekil 3.11 Histogram Örneği.....	17
Şekil 3.12 Histogram Örneği 2.....	18
Şekil 3.13 Maskeleye.....	18

Şekil 3.19 Maskeleye 2.....	19
Şekil 4.1 İlk Video Görüntüsünden Bir Kare.....	22
Şekil 4.2 İkinci Video Görüntüsünden Bir Kare.....	22
Şekil 4.3 İlk Video Hesaplanmış Arka Plan.....	23
Şekil 4.4 İkinci Video Hesaplanmış Arka Plan.....	23
Şekil 4.5 İlk Video'nun Video Karesiyle-Arka plan Farkı.....	25
Şekil 4.6 İkinci Video'nun Video Karesiyle-Arka plan Farkı.....	25
Şekil 4.7 İlk Video'nun Ölçeklendirilmiş Fark verisi.....	26
Şekil 4.8 İkinci Video'nun Ölçeklendirilmiş Fark verisi.....	27
Şekil 4.9 İlk Video'nun İkili Veriye çevrilmiş Fark Verisi.....	28
Şekil 4.10 İkinci Video'nun İkili Veriye çevrilmiş Fark Verisi.....	28
Şekil 4.11 İlk Video'nun Fazlalıkları Atılmış Karesi.....	29
Şekil 4.12 İkinci Video'nun Fazlalıkları Atılmış Karesi.....	30
Şekil 4.13 İlk Video'nun Morfolojik İşlem Uygulanmış Hali.....	31
Şekil 4.14 İkinci Video'nun Morfolojik İşlem Uygulanmış Hali.....	31
Şekil 4.15 İlk Video'nun Oluşturulmuş Video Görüntüsünden Bir Kare.....	32

Şekil 4.16 İkinci Video'nun Oluşturulmuş Video Görüntüsünden Bir Kare.....	33
Şekil 4.17 HOG Tanımı Resmi 1.....	37
Şekil 4.18 HOG Tanımı Resmi 2.....	37
Şekil 4.19 Renk Normalizasyonu Örnek Resim.....	38
Şekil 4.20 Gamma Doğrulaması Örnek Resim.....	39
Şekil 4.21 Konvolüsyon İşlemi Örneği.....	40
Şekil 4.22 İlk Videonun Kernel Filtresine Yatay Uygulanması Sonucu Oluşan Resim Karesi.....	41
Şekil 4.23 İlk Videonun Kernel Filtresine Dikey Uygulanması Sonucu Oluşan Resim Karesi.....	41
Şekil 4.24 İlk Videonun Kernel Filtresine Hem Yatay Hemde Dikey Uygulanması Sonucu Oluşan Toplam Resim Karesi.....	42
Şekil 4.25 İkinci Videonun Kernel Filtresine Yatay Uygulanması Sonucu Oluşan Resim Karesi.....	42
Şekil 4.26 İkinci Videonun Kernel Filtresine Dikey Uygulanması Sonucu Oluşan Resim Karesi.....	43
Şekil 4.27 İkinci Videonun Kernel Filtresine Hem Yatay Hemde Dikey Uygulanması Sonucu Oluşan Toplam Resim Karesi.....	43
Şekil 4.28 Orientation-Based Histogram.....	44

Şekil 4.29 İlk Videonun Oryantasyon İşlemi Sonrasındaki Ekran Görüntüsü.....	45
Şekil 4.30 İkinci Videonun Oryantasyon İşlemi Sonrasındaki Ekran Görüntüsü.....	46
Şekil 4.31 R-HOG ve C-HOG Şekli.....	47
Şekil 4.32 Sobel Maskesi.....	51
Şekil 4.33 EOH İşlemi Sonrasında Oluşan Ekran Görüntüsü.....	53
Şekil 4.34 EOH İşlemi Sonrasında Oluşan Ekran Görüntüsüne Belli Bir Eşik Değeri Uygulanması Sonucu Oluşan Ekran Görüntüsü.....	53
Şekil 4.35 Piksel Kavis Grafiği Örneği.....	55
Şekil 4.36 LBP İşlemi Sonrasında Oluşan Ekran Görüntüsü.....	56
Şekil 4.37 LBP İşlemi Sonrasında Oluşan Ekran Görüntüsüne Belli Bir Eşik Değeri Uygulanması Sonucu Oluşan Ekran Görüntüsü.....	56
Şekil 5.1 Veri Seti Örneği.....	59

## GRAFİKLERİN LİSTESİ

<b>Grafik</b>	<b>Sayfa</b>
Grafik 5.1 Çalışılan Algoritma İçin Her Bir Videodan Alınan İşlenmiş Resim Karelerindeki FP, FN Değer Karşılaştırması Grafiği.....	62
Grafik 5.2 Çalışılan Algoritma İçin Her Bir Videodan Alınan İşlenmiş Resim Karelerindeki TN Değer Karşılaştırması Grafiği.....	63
Grafik 5.3 Çalışılan Algoritma İçin Her Bir Videodan Alınan İşlenmiş Resim Karelerindeki TP Değer Karşılaştırması Grafiği.....	63
Grafik 5.4 Çalışılan Algoritma İçin Her Bir Videodan Alınan İşlenmiş Resim Karelerindeki Recall, Specity, Precision, Accuracy, F-Measure Değer Karşılaştırması Grafiği.....	65
Grafik 5.5 Çalışılan Algoritma İçin Her Bir Videodan Alınan İşlenmiş Resim Karelerindeki Precision Değer Karşılaştırması Grafiği.....	66
Grafik 5.6 Çalışılan Algoritma İçin Her Bir Videodan Alınan İşlenmiş Resim Karelerindeki Precision Değer Karşılaştırması Grafiği.....	66
Grafik 5.7 Çalışılan Algoritma ile HOG ve EOH+LBP Yöntemlerinin FP Değer Karşılaştırması Grafiği.....	67

## ÇİZELGELERİN LİSTESİ

Çizelge	Sayfa
Çizelge 3.1 RGB Operasyon Sınıflandırması .....	13

## KISALTMALAR

<b>Kısaltmalar</b>	<b>Açıklama</b>
<b>EOH</b>	Köşe Oryantasyon Histogramı - Edge Orientation Histogram
<b>LBP</b>	Lokal Nitelik Tanımlayıcısı - Local Binary Patterns
<b>HOG</b>	Odaklı Gradyanların Histogramı - Histogram of Oriented Gradients
<b>SVM</b>	Destek Vektör Makinesi - Support Vector Machine
<b>TP</b>	Doğru Pozitif – True Positive
<b>TN</b>	Doğru Negatif – True Negative
<b>FP</b>	Yanlış Pozitif – False Positive
<b>FN</b>	Yanlış Negatif – False Negative



## **1.GİRİŞ**

Günümüzde birçok dalda yeni teknolojik gelişmeler olmuştur ve olmaktadır. Bu teknolojik gelişmelerin sonucu olarak, birçok sorun ve sorunlar ortaya çıkmıştır. Bu gibi sorunların tespit edilmesi ile yaşanacak bazı olumsuzluklar, çeşitli yöntemler vasıtasıyla en aza seviyeye düşürülebilir.

Bundan sebeple; teknolojik ve bilimsel gelişmelere paralel düşme olması gerekirken, trafik kazaları hala ülkemizde ve dünyada gündemin en ön sıralarında yer almaktadır. Hava, deniz ve demiryolu trafiğine nazaran karayollarındaki trafik, güncel hayatın önemli bir sorunu olarak devam ede gelmektedir.

Günlük yaşamda trafiğe çıkmayan hiçbir insan düşünülmeceğinden, trafik kazası her insanın başına gelebilecek olağan bir olay olarak nitelendirilmektedir. Bu nedenle bir trafik kazasına karışan veya maruz kalan kişilerin, olaydaki kusur durumları, teknik, hukuki ve bilimsel bir şekilde objektif olarak ortaya konulması gerekir. Bundan sebeple çalışmamız olan yaya tespiti sistemleri çok büyük önem arz etmektedir.

### **1.1. Yaya Tespiti Sistemleri Uygulamaları**

Yaya tespiti sistemleri hakkında çalışmalar yapılmakta ve günden güne yeni çözümler, algoritmalar yeni çalışmalarla birlikte üretilmektedirler. Yaya tespiti sistemleri sadece trafik alanında değil birçok alanda kullanılmaktadırlar. Bunlara örnek olarak; araç sistemleri (kendi kendini yöneten araçlar, sürücü yardım sistemleri), hırsız yakalama sistemleri, insan-robot etkileşimli sistemler, görsel gözetim ve davranış analizi sistemleri, görüntü analizi ve içerik tabanlı görüntü alma sistemleri, maçlarda oyuncu tespitinde kullanılan sistemler, ve bunun gibi birçok sistem örnek verilebilir. Yaya tespiti sistemleri kullanıldıkları sistemlerde büyük önem arz etmekte ve sistemlerin oluşturulup geliştirilmesinde büyük katkılar sağlamaktadırlar.

## 1.2. Tez Kapsamı

Tezimizin hedefi görüntü işleme tekniklerini kullanarak ve video çerçeveleri arasındaki farkların analizine dayalı olarak yayaların izlenmesi, yayaların tespiti ve yaya hareketlerini izleyen videonun oluşturulmasıdır. Bu işlem video kaydı olarak alınmış yayalar (kişiler) üzerinden yapılacaktır. Burada amaç, videolar içerisindeki yayaların buldukları konumların sayısal veriye çevrilmesidir. Ayrıca, daha önce yapılan farklı benzer çalışmaların deneysel olarak analizi ve bizim oluşturduğumuz algoritmalarla performans karşılaştırılması işlemi yapılmıştır.

Yaya tespiti sistemimiz genel olarak 4 ana aşamadan meydana gelmektedir. Bu aşamalar sırasıyla Şekil 1.1 üzerinden anlatılırsa;

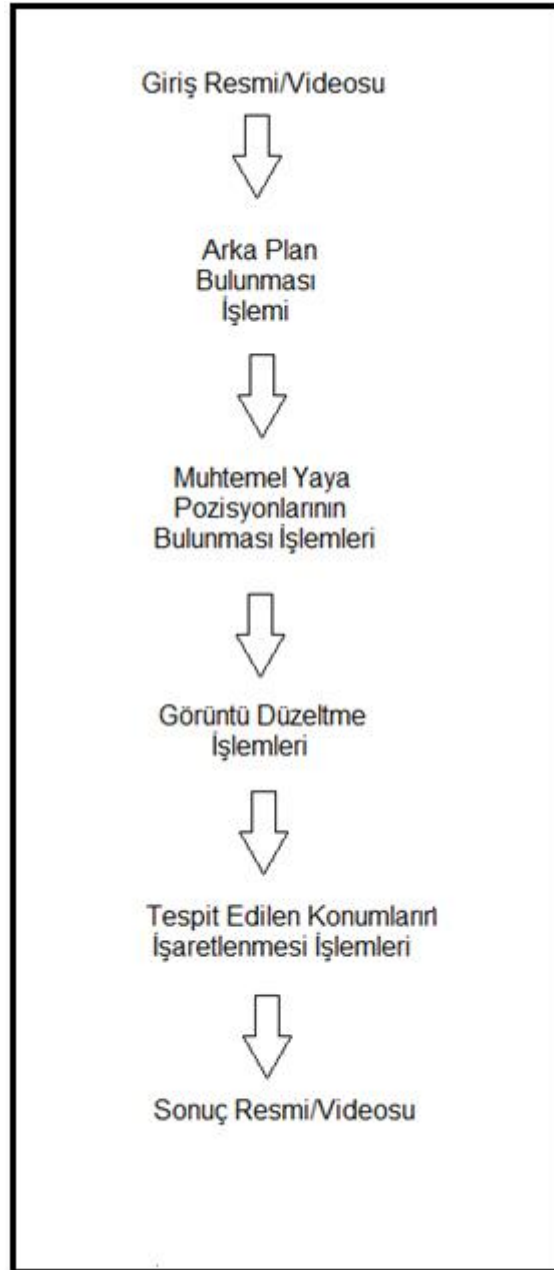
İlk olarak, videoda veya resim karelerinde arkaplan elde etme işlemi yapılır. Arka plan elde etme işleminde ilk adım olarak videodaki her bir kare gri tonlara çevrilir. Ardından gri tonlara çevrilmiş bu kareler toplanır ve videodaki toplam kare sayısına bölünerek videodaki her bir kare için arka plan bulunmaya çalışılır.

İkinci olarak, muhtemel yaya pozisyonlarının bulunması işlemleri gerçekleştirilmektedir. Muhtemel yaya pozisyonlarının bulunması işlemlerinde ise ilk adım olarak gri tonlara çevrilmiş her bir video karesinden ilk aşamada elde edilen arka plan çıkarılır. İkinci adım olarak ilk adımın sonucunda elde edilen video karesindeki değerler ileride daha optimize sonuçlar alınabilmesi için ölçeklendirilir. Ardından üçüncü adımda ise segmentasyon (bölümlendirme) işlemleri gerçekleştirilir.

Üçüncü olarak, görüntü düzeltme işlemleri yapılır. Görüntü düzeltme işlemlerinde ilk adım olarak, video karelerinde oluşan fazlalıklar atılmıştır. Bu işlemi gerçekleştirmek için ise optimize seçilecek bir boyun (değer,parametre) altındaki obje olarak işaretlenmiş kümeler fazlalık olarak kabul edilip elenmiştir. İkinci adım olarak ise

yaya tespiti için tespit edilen yaya görüntüleri içerisindeki boşluklar morfolojik işlemler kullanılarak doldurulmuştur.

Son olarak ise tespit edilen konumların işaretlenmesi işlemleri yapılır. Bu işlem için ilk üç adımın sonucunda tespit edilen obje konumları videodaki her bir kareye işlenir ve işlenmiş video oluşturulur.



**Şekil 1.1 Yaya Tespit Sistemi Akış Şeması**

Tezimizde geliştirilen bu yöntem bazı diğer aygıtlara veya sistemlere uygulanarak kaza problemleri engellenebilir veya azaltılabilir.

### **1.3. Tez Taslağı**

Tezimizin organizasyonu aşağıda anlatıldığı gibidir:

İkinci bölümde, yaya tespit sistemleri bazında daha önce yapılmış çalışmalardan kısa özetler halinde bahsedilecektir.

Üçüncü bölümde, görüntü işleme (image processing) işlemi hakkında genel bilgiler verilecektir ve görüntü işlemede kullanılan bazı terimlerin (işlemlerin) kısa tanımlamaları yapılacaktır.

Dördüncü bölümde, geliştirilen yöntem detaylı bir şekilde anlatılmıştır. Buna ek olarak uygulanan yöntem içerisinde yeni algoritmalar oluşturulup, denenmiştir ve geliştirilen bu algoritmaların anlatımı yapılmıştır. Ayrıca daha önce yapılan farklı benzer çalışmaların gerçekleştirilmesi işlemi yapılmıştır.

Beşinci bölümde, veri setimizin içeriğinin anlatımı ve daha önceki bölümlerde bizim tarafımızdan üretilen algoritmaların sonuçlarıyla, daha önce yapılmış ve bizim tarafımızdan gerçekleştirilmiş farklı benzer çalışmaların sonuçlarının deneysel ve grafiksel analizi yapılacaktır. Ayrıca yöntem geliştirilmesi sırasında karşılaşılan problemlerden bahsedilmiştir.

Altıncı bölüm, sonuç bölümüdür. Bu bölümde tezin üzerine çalışılmaya devam edilecekse eğer hangi problemlere karşı çözüm bulunması konusunda öneriler verilmiştir ve genel sonuç özeti yapılmıştır.

## 2. LİTERATÜR TARAMASI

Bu bölümümüzde yaya tespit sistemleri bazında daha önce yapılmış çalışmalardan kısa özetler halinde bahsedilecektir.

[7] numaralı makalenin amacı var olan resimler üzerinden yaya tespitini gerçekleştirmektir. Bunun için ilk olarak bu çalışmanın algoritması, şekilsel özellikleri öğrenme (shapelet features) ve orta seviyeli(mid-level features) özellikleri kurma işlemleri üzerine tasarlanır. Bu özellikler resim karesinin lokal bir bölgesine odaklanırlar ve düşük seviyeli gradient bilgilerden(low-level gradient information) oluşturulmuşlardır. Bu özellikler sayesinde yaya veya yaya olmayan sınıf ayrımı yapılır. İkinci olarak, adaboost algoritması kullanılarak şekilsel özellikler gradient odaklı cevapların birleşiminden oluşturulur. Son adım olarak ise sonuç sınıflandırıcılarını eğitmek için Adaboost algoritması ikinci kez kullanılarak öğrenilen şekilsel özelliklerin alt kümesi seçilir ve sonuç sınıflandırıcıları eğitilir.

[8] numaralı makale ise genel obje tespit sistemleri üzerine çalışmaktadır ve sistem değişik türlerde obje tespiti için eğitilmiştir. Özellikle de yaya tespiti üzerine odaklanılmıştır. Bu çalışmanın örneklerinden öğrenilen paradigma, el ürünü çözüm ihtiyaçlarımızdan kaçınmamıza izin vermektedir. Bu çalışma diğer çoğu yaya tespit sistemleri gibi bazı şartlara veya temellere dayanmaz. Örnek vermek gerekirse; bu çalışma hareket bilgisine dayanmaz ve görüntü yapısı veya obje sayısı üzerinde tahminler yapmaz. Burada bahsedilen konu, doğruluğu artırmak için işleyen video dizileri üzerinden alınan dinamik bilgilerin avantajlarının artırılmasıdır. Bu çalışma özellikle DomainChrysler test aracının içine entegre edilerek gerçek ve gerçek zamanlı sistemlerde tanımlanmıştır.

[9] numaralı makalede ise gerçek zamanlı tespit için tümleşik sistemler üzerinden hareket eden araçtan yayaların tespit işlemleri tanımlanmıştır. Bu çalışmada birincil sensör olarak stereo vizyon kullanılmıştır ve bu sensörün monoküler vizyon üzerinden birkaç pratik avantajı görülmüştür. Bu avantajlardan ilk olanı resim karesi üzerinde hızlı bir şekilde ilgi alanını tanımlar. İkinci olanı ise tanımladığı alanları

şekilsel bazlı olarak sınıflandırır ve son olanı, yaya tespiti takibini üç boyutlu dünya koordinatlarında yapar. Ek olarak, bu sistemde monoküler görünüm tabanlı algoritmadan (monocular appearance-based algorithm) faydalanılmıştır.

[10] numaralı makalede alışılmışın dışında, değişik resim kareleri üzerinde çok küçük ölçekler üzerinden yaya tespiti yapılmıştır (bu ölçekler 16x20 ve 32x40 arasındadır). Bu çalışmada superpiksel (superpixel) segmentasyonu ile gruplandırılmış lokal resim koleksiyonlarına bağlı olarak yeni, farklı resim öznitelikleri(özellikleri) dizisi oluşturulmuştur. Oluşturulan öznitelik dizileri adaboost algoritması kullanılarak biriktirilmiş ve sınıflandırılmıştır. Daha sonra, ortalama kayma modu tahmini yaklaşımı (mean shift mode estimation approach) kullanılarak biriktirilen pozitif sınıflandırılmış öznitelikler potansiyel hipotez için oylanmıştır. Bu çalışmada sunulan yöntem ilgi noktalarına (interest point) dayalı standart oy yaklaşımları gibi kayan pencere yaklaşımının (sliding window approach) yaygın sınırlamalarının üstesinden gelmiştir. 20000'den fazla resim karesi üzerinde bu yaklaşımın potansiyelinin gözlenmesi ile genişletilmiş testler veri seti üzerinden oluşturulmuştur.

[11] numaralı makalede yaya tespit sistemleri, hareket bilgisiyle resim karesinin yoğunluk bilgisinin bütünleştirilmesi ile tanımlanmıştır. Bu çalışmada tespit stil algoritması (detection style algorithm) kullanılmıştır. Bu algorithmada işlemler ardışık gelen iki video karesi üzerinden bir detektör tarafından taranarak gerçekleştirilir. Detektör adaboost algoritması kullanılarak eğitilir. Ayrıca, detektör yürüyen yayanın tespitinde hem hareket hem de görünüm bilgisinden yararlanarak eğitilir. Eski yaklaşımlarda detektörler ya hareket bilgisine ya da görünüş bilgisine dayalı olarak inşa edilirdi ama bu çalışmada ilk defa her iki bilgide tek bir detektör içerisinde birleştirilmiş ve kullanılmıştır.

[12] numaralı makalede tespit yöntemi, yukarıdan aşağıya tanıma ile aşağıdan yukarıya resim bölümlendirmesinin birleştirilmesiyle geliştirilmiştir. Bu metotta iki ana aşama vardır. Bunlar hipotez oluşturma adımı ve doğrulama adımıdır. İlk olarak, yukarıdan aşağıya hipotez oluşturma adımında şekil içerik özellikleri geliştirilir. Bu

adım sistemin obje ve arka plan bozulmalarına karşı daha dayanıklı olmasını sağlar. İkinci adım olarak, doğrulama adımı içerisinde yukarıdan aşağıya obje hipotezi ile tutarlı olacak şekilde uygulanabilir segmentasyon setleri hesaplanır, ardından doğru pozitif oranını azaltmak için yanlış pozitif budama (False Positive Pruning(FPP)) yöntemi uygulanır. Ek olarak, bu çalışmada tipik olarak doğru pozitif bölgelerin herhangi bir uygulanabilir resim segmentasyonlarıyla uyumlu olmadığı gerçeğinden faydalanılmıştır.

[20] numaralı makaledeki yaya tespit yönteminde, ilk olarak değerlendirme metrikleri geliştirmek amaçlanmıştır ve bu düşüncenin akabinde her bir tespit karesindeki kusurlu ve tam görüntülerde tespit performansını tahmin etmekte başarısız olunabildiği gözlenmiştir. Ardından alınan bu sonuçlar üzerine, var olan yöntemleri tarafsız bir şekilde karşılaştırabilmek adına modern performans ve doğrudan bir genel bir bakış için birkaç tespit sistemleri üzerinde kıyaslama noktaları oluşturulması sağlanmıştır ve bu çerçevede işlemler gerçekleştirilmiştir. Bu çalışmada ek olarak, berkeley bölümlendirme veri kümesi [23], barron [24] ve middlebury [25] optik akış veri kümeleri, middlebury stereo veri kümeleri [26] kullanılmıştır.

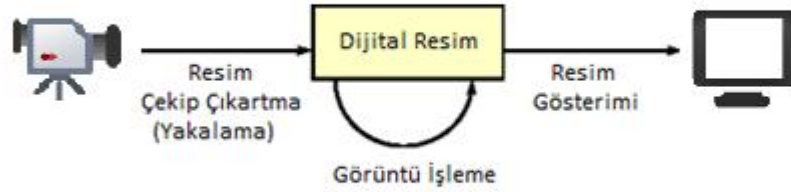
[21] numaralı makalede, kalabalık ve çakışan yaya toplulukları içeren tespit kareleri üzerinde işlemler gerçekleştirilmiştir. Burada birden fazla iterasyon ve değişik kaynaklardan elde edilen kanıtlarla yeni bir algoritma geliştirilmiştir ve bu algoritma kullanılarak tespit işlemleri gerçekleştirilmiştir. Bu metodun ana çekirdeğini, olasılı yukarıdan aşağıya bölümlendirme yöntemi aracılığıyla elde edilen lokal ve global ipuçları oluşturur.

[22] numaralı makalede, sürüş destek sistemleri için tek çerçeveli sınıflandırma yöntemi ve sistem performans düzeyleri kullanımı ile yaya tespit sistemi gerçekleştirilmiştir. Burada fonksiyonel ve mimari açıdan dökümler tanımlanmıştır. Ek olarak, tek çerçeveli sınıflandırma yöntemi sonuçları ve sistem performans düzeyleri rakamları kalan zaman aralıklarında ve normal hava şartlarında ele alınarak oluşturulmuştur.

### 3.GÖRÜNTÜ İŞLEME(İMAGE PROCESSİNG) GENEL BİLGİLER

#### 3.1. Görüntü İşleme (İmage Processing) Nedir, Yararları Nelerdir?

Görüntü işleme, gerçek yaşamdaki görüntülerin kaydedilerek veya ölçülerek sayısal bir resim haline getirilmesi sonucu bir girdi resmi olarak işlenerek, o resmin özelliklerinin ve görüntüsünün amaca uygun bir şekilde değiştirilmesidir. [13,14]

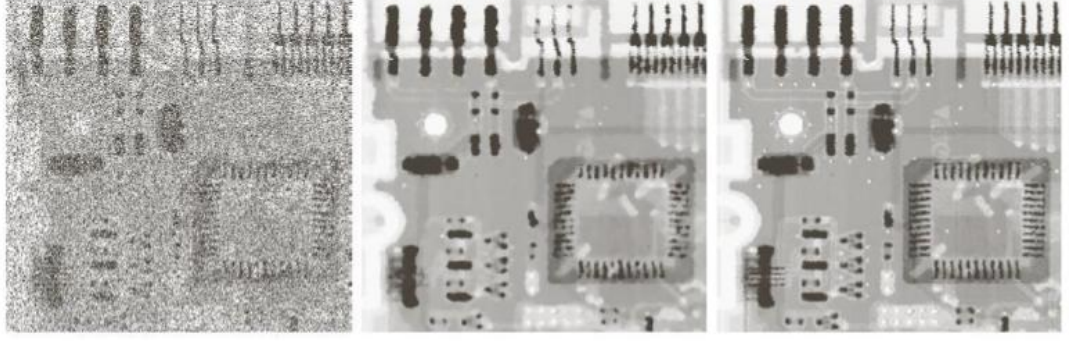


**Şekil 3.1 Görüntü İşleme Tanımı**

Yukarıdaki Şekil 3.1'de, ilk olarak resim karesi video'dan çekip çıkartılıp elde edilir. Ardından elde edilen dijital(sayısal) resim karesi üzerinde çeşitli görüntü işleme işlemleri yapılarak iyileştirilmiş resim karesi elde edilip video'nun üzerine tekrar işlenir ve işlenmiş video çıktısı elde edilir.

Görüntü işleme işlemlerinin resimlerin veya görüntü karelerinin iyileştirilmesi üzerinde, bazı uygulamaların geliştirilmesi noktasında ve bunun gibi birçok konularda yararları vardır. Yararlarından ilk olarak, resimler genellikle üzerinde işlemler yapılmak üzere analog ortamlardan dijital ortamlara geçirildiği için bozukluklar (noise) içerir ve görüntü işleme işlemi bu hataları düzeltmek için kullanılır. Örneğin, bir sonraki sayfadaki Şekil 3.2'deki resimde bir gürültü (noise) düzeltme işlemi yapılmıştır.





**Şekil 3.2 Gürültü(Noise) Düzeltme Örneği (Şekildeki resim [14]'den alınmıştır.)**

Şekil 3.2'de ilk olarak sol taraftaki çok sayıda gürültü içeren girdi resim karesi alınmış ardından çeşitli görüntü işleme teknikleri uygulanarak ilk önce ortadaki (girdi resim karesine 7x7 medyan filtresi uygulanması sonucu oluşan resim karesi), ardından sağ taraftaki(girdi resim karesine uyarlanabilir medyan filtresi(adaptive median filter) uygulanması sonucu oluşan resim karesi) daha az gürültü içeren resim kareleri elde edilmiştir.

Görüntü işleme işleminin yararlarından ikinci olarak, resimler yararlı işlerde kullanılmak üzere işlenir. Bu görüntü işleme teknikleri genellikle insanların hayatına kolaylık getirecek çeşitli uygulamalar geliştirilirken kullanılır. Örneğin bir görüntüdeki obje sayısını saymada, tıp alanında çekilen röntgen resimlerinde, günümüzdeki bilgisayarlarda güvenlik amacıyla kullanılan yüz tanıma sistemlerinde ve bunun gibi işlemlerde görüntü işleme teknikleri kullanılmaktadır.

Bir görüntü veya resim karesi gerçek yaşamda basit 2 değişkenin bir fonksiyonu olarak tanımlanır. Örnek olarak,  $a(x,y)$  gibi bir fonksiyonla ifade edilen bir resimde  $a$  bir şiddet birimi iken  $x$  ve  $y$  değişkenleri ise resmin gerçek koordinatlarıdır. Bir resim karesi, alt resim bileşenlerinden oluşur ve bu alt resimlere bölge denir. Bir görüntü işleme sistemi, resim karesi içerisinde seçilen bölgelere uygun işlemleri yapabilme yeteneğine sahip olmalıdır. Örneğin resim karesinin bir bölgesine

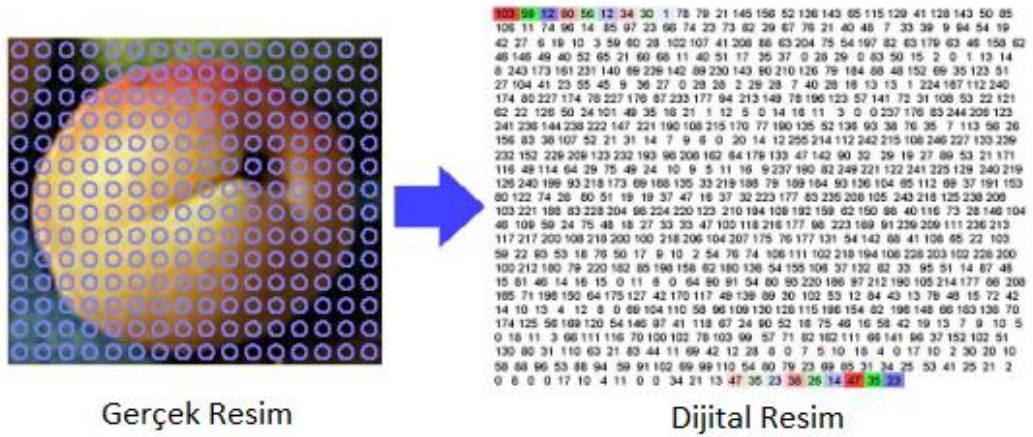
bulanıklık özelliği verilirken, diğer bir bölgesine ise parlaklık özelliği verilebilmelidir. [13]

## 3.2. Bazı Görüntü İşleme Tanımları

### 3.2.1. Dijital Görüntü

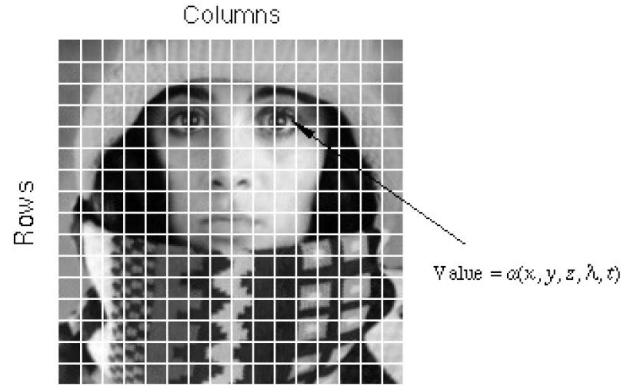
Dijital(sayısal) görüntünün tanımı; iki boyutlu resim karesinin sayısal değerlerden oluşan gösterimidir.

Aşağıdaki Şekil 3.3'deki resim karesinde görüldüğü üzere gerçek bir resim karesinin dijital resim sayısal değerlerinin gösterimi sergilenmiştir.



Şekil 3.3 Dijital Görüntü Örneği(Şekildeki resim [14]'den alınmıştır.)

Dijital görüntü yapısı  $a[x,y]$ ,  $x$  ve  $y$  sayılarında satır ve sütunlardan oluşur. Dijital görüntü içerisindeki satır ve sütunların kesiştiği her bir bölgeye piksel denir. Her bir pikseldeki değer ise derinlik ( $z$ ), renk( $\lambda$ ) ve zamanın( $t$ ) bir fonksiyonudur. Bir sonraki sayfadaki Şekil 3.4'deki resim, yukarıdaki anlatıların şekilsel gösterimidir.[18]



**Şekil 3.4 Piksel Gösterimi Örneği (Şekildeki resim [18]'den alınmıştır..)**

Yukarıdaki Şekil 3.4'deki resimde satır ve sütun sayısı onaltı'dır. Resim içerisindeki her piksele atanan değer ise o pikselin parlaklık değerinin en yakın tamsayıya yuvarlatılmış halidir. [18]

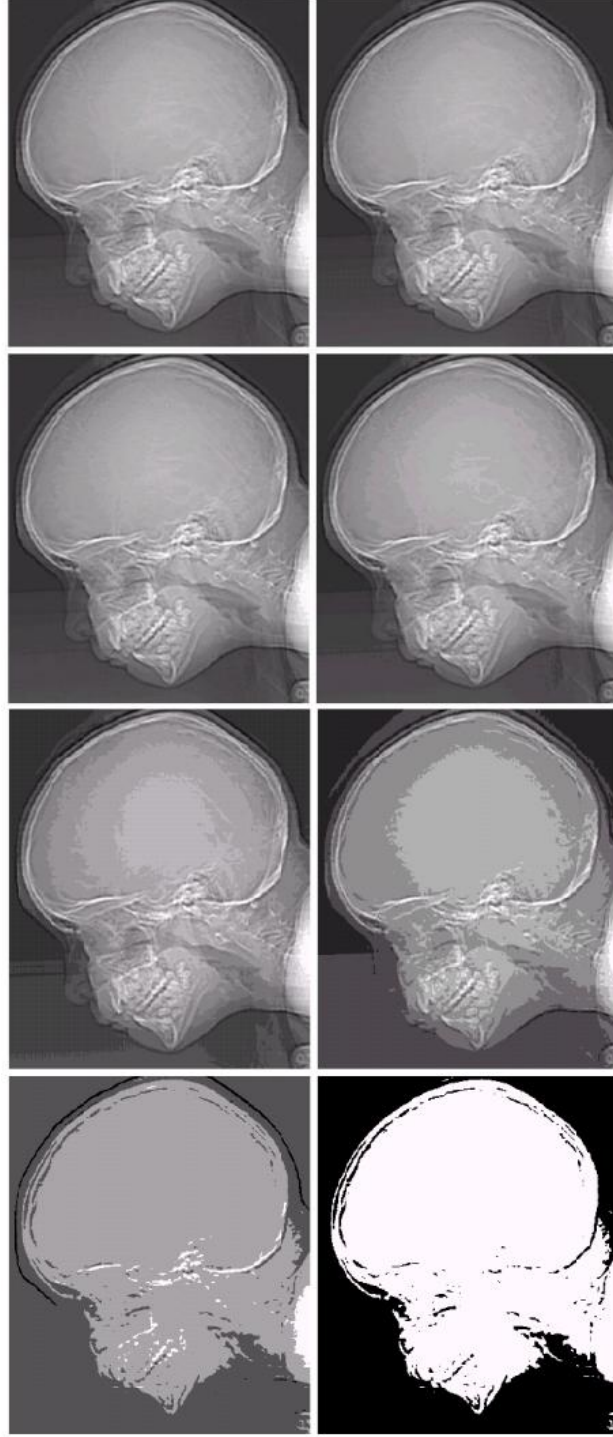
### 3.2.2. Niteliklendirme

Görüntü piksel değerlerinin belirli aralıklarda olması, meydana gelen görüntünün niteliğini tanımlar, ayrıca bu durum görüntünün niteliğiyle değişiklik yapabilmemizi sağlar. Örnek vermek gerekirse 0 beyazı, L-1 de siyahı temsil eder ve bu değerler arasındakiler ise gri tonlarını ifade etmektedir. Burada 3.1 numaralı denklem olmak üzere, b değeri görüntünün 1 pikselini ifade etmek için gereken bit sayısıdır. Örnek vermek gerekirse, eğer görüntüde alınan b değeri 5 ise o görüntüde 32 adet gri ton değeri bulunmaktadır.[13,17]

$$n = 2^b \quad (3.1)$$

Yukarıda anlatılanlara ek olarak, ikili resim(binary image) oluşturulmak istenirse b değeri bir'e eşitlenir (İkili resim sadece 0 ve 1 'lerden oluşur). Bir sonraki sayfadaki Şekil 3.5'deki resim karesinde, n değerlerinin farklı değerler alması sonucu resim karesinin aldığı değerler gözükmemektedir. (Şekil 3.5'deki resimde sol üstten başlamak

üzere sırasıyla n değerleri 256, 128, 64, 32, 16, 8, 4, 2'dir. Burada n değerlerinin artması sonucu gri tonlarının sayısında artma gözlenmektedir.)



**Şekil 3.5 Niteliklendirme Örneği (Şekildeki resim [14]'den alınmıştır.)**

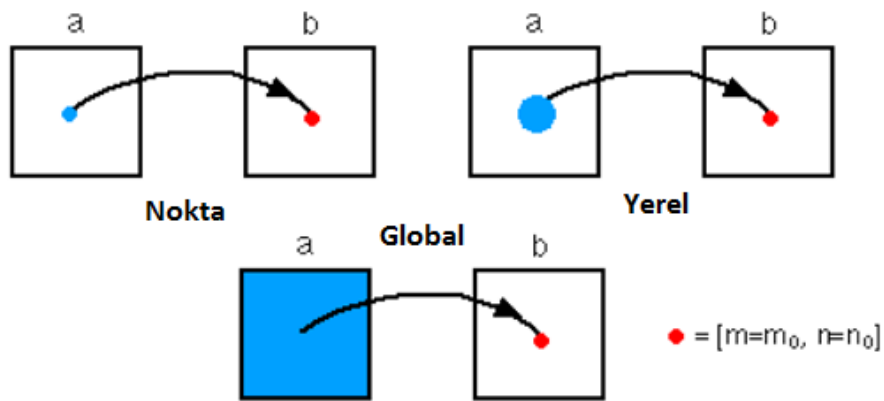
### 3.2.3. Görüntü Operasyonları

Görüntü operasyonları,  $a[x,y]$  gibi bir girdi görüntüsünü veya video karesi görüntüsünü,  $b[x,y]$  gibi farklı bir çıktı görüntüsüne çevirir.[18] Operasyonlar, bir sonraki sayfadaki Çizelge 3.1'deki resimde sınıflandırılmıştır.

**Çizelge 3.1 RGB Operasyon Sınıflandırması**  
(Şekildeki çizelge [18]'den alınmıştır.)

<i>Operasyon</i>	<i>Karakteristik</i>	<i>Piksel Başına Karmaşıklık</i>
Nokta	O koordinatın çıkış değeri sadece giriş değerine bağlı	Sabit
Yerel	O koordinatın çıkış değeri , komşu noktaların değerine bağlı	$P^2$
Global	O koordinatın çıkış değeri, tüm resim elemanlarına bağlı	$N^2$

Yukarıdaki çizelgede gösterilen operasyonlar  $N \times N$  boyutlarında ve  $P \times P$  komşu piksel bölgeleri üzerinden yapılan hesaplamaları içerir. Görüntü işlemleri Şekil 3.6'deki resimde gösterilmiştir.



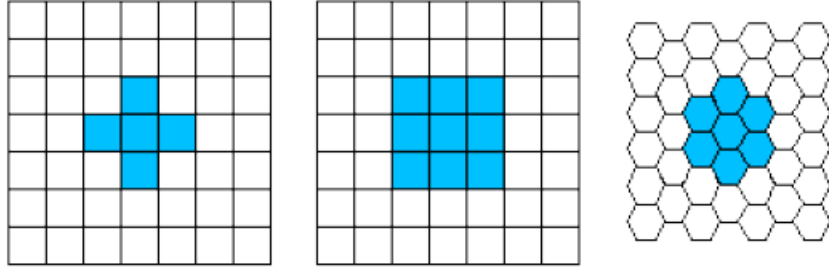
**Şekil 3.6 Görüntü İşlemleri(Şekildeki resim [18]'den alınmıştır.)**

### 3.2.4. Komşuluk Çeşitleri

Komşuluk ilişkileri, sayısal görüntü işlemede çok önemli bir kavramdır ve ancak görüntü örnekleme yapılarak elde edilebilir. Temel örnekleme yöntemleri ise aşağıda anlatıldığı gibidir. [18]

İlk olarak dikdörtgenel örneklemede, resmin üzerinde dikdörtgenel bir ızgara olduğu düşünülür ve işlemler bu durum göz önüne alınarak yapılır. [18]

İkinci olarak, altıgenel örneklemede ise resmin altıgenlerden oluşmuş parçalar içerdiği düşünülür ve işlemler bu bazda ele alınır. [18] Aşağıdaki Şekil 3.7'deki resimde bu iki örnekleme gösterilmiştir. Resimde ilk iki resim dikdörtgenel örnekleme, en sağdaki üçüncü resim ise altıgenel örnekleme göstermektedir. Ayrıca şekilde 4'lü, 8'li ve 6'lı komşuluk ilişkileri gösterilmiştir.



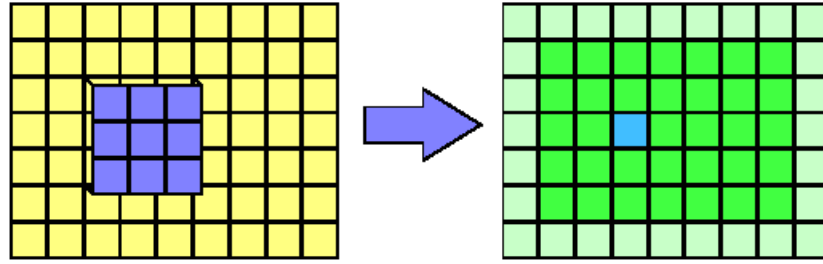
Şekil 3.7 Komşuluk Örnekleme (Şekildeki resim [18]'den alınmıştır.)

Dikdörtgenel örnekleme, altıgenel örnekleme göre donanımsal açıdan daha rahat gerçekleştirilebildiği için altıgenel örnekleme göre daha çok tercih edilir. [18]

Yerel görüntü işleme operasyonları, çıkış piksel değeri olan  $b[x,y]$  değerini, giriş pikseli  $a[x,y]$  değerinin komşularını kullanarak hesaplarlar. [18]

### 3.2.5. Filtreleme

Sayısal bir görüntü üzerinde filtreleme işlemi; resmin üzerindeki her bir piksel değerinin, sanki resmin üzerinde bir filtre varmış gibi davranılıp yeniden hesaplanması işlemidir [13,14]. Aşağıdaki Şekil 3.8'deki resimde filtreleme şekilsel olarak gösterilmiştir.



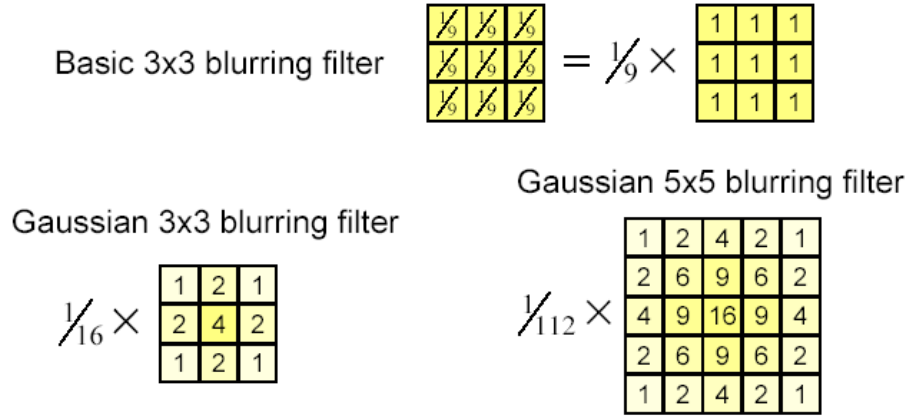
Şekil 3.8 Filtreleme Örneği(Şekildeki resim [13]'den alınmıştır.)

Filtreleme işlemlerinde kullanılan filtreler sayesinde girdi resimlerine yapılan filtreleme işlemleri sonrasında yeni resimler değişik etkiler verilerek elde edilir.

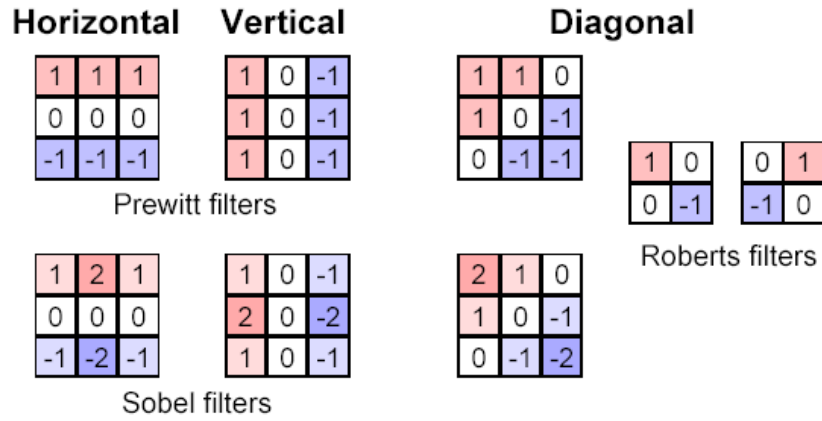
Filtreleme işlemi, genel olarak 3.4 numaralı denklem ile elde edilebilir:

$$f'(x, y) = \sum_{-\infty}^{+\infty} \sum_{-\infty}^{+\infty} h(i, j) * f(x - i, y - j) \quad (3.4)$$

Yukarıdaki formülde h fonksiyonu filtreyi ifade etmektedir. Bu fonksiyonuna değişik filtre değerleri koyulup resim karelerinde değişik filtreleme işlemleri yapılabilir. Bir sonraki sayfadaki Şekil 3.14 ve Şekil 3.15' de resimlerde örnek filtreleme matrisleri gösterilmiştir.



Şekil 3.9 Filtreleme Matrisi Örneği 1



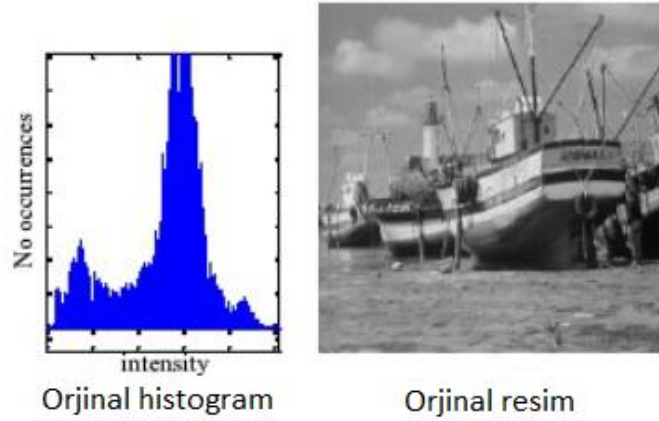
Şekil 3.10 Filtreleme Matrisi Örneği 2

Yukarıdaki Şekil 3.9 ve Şekil 3.10' deki resimlerde gösterilen örnek filtreleme matrisleri için; Şekil 3.9'deki bulanıklık (blurring) filtresi, fotoğraflarda kontrastı azaltmak ve renk geçişlerinde oluşan parazitleri ortadan kaldırmak için kullanılır. Şekil 3.9'deki diğer bir filtre olan gaussian filtresi ise bulanıklık filtresine benzer olarak resim kareleri üzerinde bulanıklaştırma derecesini ayarlamamıza olanak verir. Ardından Şekil 3.10'deki prewitt, sobel, robert filtreleri resim kareleri üzerinde köşe tanıma işlemleri için kullanılır ve tercih edilirler.



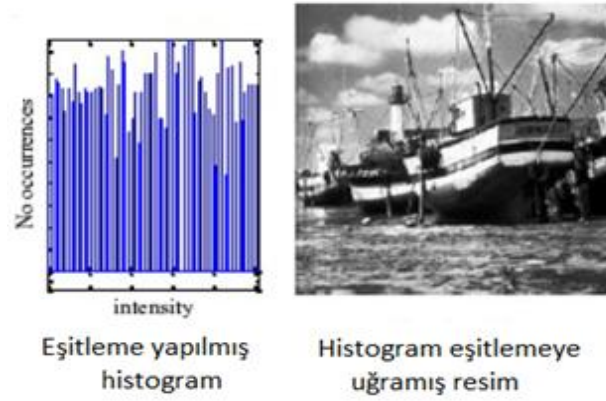
### 3.2.6. Histogram

Histogram, her bir resim karesindeki görüntünün içerdiği piksel değerlerinin ağırlığını belirten grafiksel bir gösterimdir [13,14]. Yani kısacası, resim karesindeki renk tonları değerlerinin yayılımını gösteren bir şekilsel grafikdir. Aşağıdaki Şekil 3.11'deki resimde, sol taraftaki resim sağ taraftaki resim karesinin histogramı'dır.



**Şekil 3.11 Histogram Örneği(Şekildeki resim [13]'den alınmıştır.)**

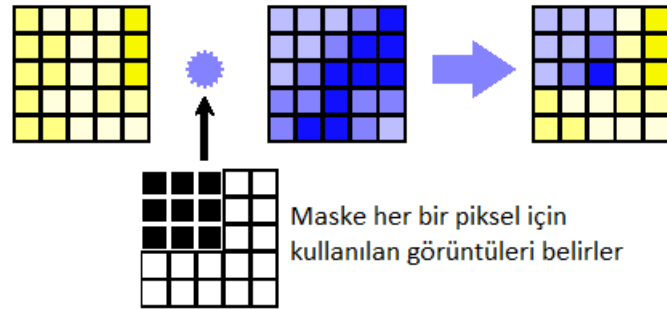
Yukarıdaki Şekil 3.11'deki resimde sağ taraftaki resmin, sol tarafta birçok değer içeren dağınık bir piksel dağılımı vardır. Bu durum resmin netliğinin iyi olmadığını göstermektedir. Bu nedenle daha net bir görüntü için histogramın daha ayırık ve düzgün bir yapıya gelmesi sağlanmalıdır. Bu işleme histogram eşitlemesi (histogram equalisation) denir. Şekil 3.12'deki resim, Şekil 3.11'deki orjinal resmin histogram eşitlemesi işlemine uğradıktan sonraki histogram grafiğini ve resmin son halini göstermektedir.



**Şekil 3.12 Histogram Örneği 2 (Histogram Eşitleme İşlemi)**  
(Şekildeki resim [13]'den alınmıştır.)

### 3.2.7. Maskeleye(Maskeler)

Resim karesinin istenen bölgesinin kesip çıkartılarak, elde edilmesi işlemine maskeleye işlemi denir [13,14]. Aşağıda Şekil 3.13'de resimde maskeleye işlemi şematize edilmiştir ve Şekil 3.14'daki resimde bu durum örneklendirilmiştir.



**Şekil 3.13 Maskeleye(Şekildeki resim [13]'den alınmıştır.)**



Orjinal resim



Maske



Resim ve maskenin  
birleşmesi sonucu  
çıkan resim

**Şekil 3.14 Maskeleme 2(Şekildeki resim [13]'den alınmıştır.)**

Şekil 3.14'daki resimde, ilk olarak resim alınmıştır. Ardından uygulanacak maske seçilmiştir. Son olarak ise maske orjinal resme uygulanmıştır ve en soldaki uçak objesinin elde edildiği resim görüntüsüne ulaşılmıştır.

#### 4. YÖNTEMİN UYGULANMASI

Uygulanan yöntemi adım adım maddelendirerek göstermek gerekirse;

1) Video okunması.

2) Video arka planı bulunması.

3) Videodaki her bir kare için;

3.1) Muhtemel yaya pozisyonlarının bulunması işlemleri yapılması.

3.1.1) Kareden arka planın çıkarılması.

3.1.2) Sonucun 0 ile 1 arasında ölçeklendirilmesi.

3.1.3) Kmeans algoritmasıyla ikiliye veriye çevirme işlemi yapılması.

3.2) Görüntü düzeltme işlemleri yapılması.

3.2.1) Karedeki oluşan fazlalıkların atılması, kırılmaların giderilmesi.

3.2.2) Yaya takibinin sağlanabilmesi için tespit edilen yaya görüntüleri içerisindeki boşluklar doldurulması.

4) Okunan videonun üzerine tespit edilen konumların işaretlenmesi.

5) Yeni yöntemler geliştirilmesi ve bunların denenmesi tekrar birinci adıma dönülmesi işlemi(iterative) ve ardından yeni yöntem geliştirme çalışmaları bitince altıncı adıma geçilmesi.

6) Daha önce yapılmış farklı benzer uygulamaların bizi tarafımızdan gerçekleştirilmesi işlemi.

7) Elde edilen sonuçların grafiksel ve deneysel karşılaştırılması. (Bu kısım deneyler bölümünde anlatılacaktır.)

Yukarıdaki kısımda maddelendirilen yöntemdeki aşamalar ve üretilen algoritmalar Matlab programı kullanılarak gerçekleştirildi. Burada bazı adımlarda algoritmalar veya uygulamalar kendimiz tarafından oluşturulmuştur, bazı adımlarda ise algoritmalar veya uygulamalarda Matlab'ın hazır fonksiyonları kullanılmıştır. Sonuç olarak bu uygulama sürecinin adımları tek tek detaylı olarak aşağıdaki bölümlerdeki gibi gözlenmiştir.

#### **4.1. Video Okuma**

Videoyu işlemek için video ile ilgili elimizde iki yapı olması gerekiyor. Bunlar videonun içerdiği karelerle, videonun yapısal bilgileridir. Yapısal bilgiler; bir videonun kaç kareden oluştuğu, dosyanın boyu, videonun boyutu, sıkıştırma yöntemi ve bunun gibi bilgilerden oluşmaktadır. Bizim kullandığımız yani bizim için gerekli olan yapısal bilgiler videonun kaç kareden oluştuğu ve bu karelerin boyutlarıdır.

Video hakkında yapısal bilgileri öğrenmek için videoyu çalıştırdıktan sonra “aviinfo” fonksiyonu ( örnek olarak “aviinfo SampleVideo.avi “ komutu girilebilir) ve videodaki kareleri okumak için aviread fonksiyonu ( örnek olarak “aviread (‘SampleVideo.avi’) “ komutu girilebilir) kullanılmıştır. Aşağıda Şekil 4.1 ve 4.2’-deki resimlerde iki ayrı videodan okunmuş iki ayrı video karesi görülmektedir. Aşağıdaki adımlar da bu kareler üzerinden anlatılarak gösterilmiştir.



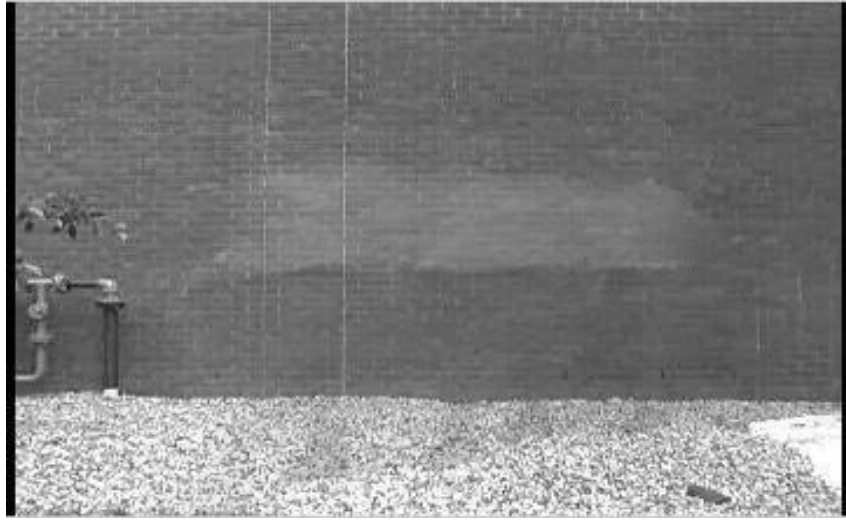
**Şekil 4.1 İlk Video Görüntüsünden Bir Kare**



**Şekil 4.2 İkinci Video Görüntüsünden Bir Kare**

## 4.2. Arka Planı Bulma

Videodaki arka planı bulmak için kırmızı, yeşil, mavi renkte tanımlı olan kareler (videodaki her bir kare ) gri tonlara çevrilmiştir. Ardından gri tonlara çevrilmiş kareler toplanmış ve bu toplam videodaki kare sayısına bölünmüştür. Böylece videolarımızın arka plan resim kareleri gri tonda elde edilmiştir. Aşağıdaki Şekil 4.3 ve Şekil 4.4'deki resimlerde videoların arka planları görülmektedir.



**Şekil 4.3 İlk Video Hesaplanmış Arka Plan**



**Şekil 4.4 İkinci Video Hesaplanmış Arka Plan**

Her bir karenin toplanıp ve çıkan sonucun videodaki kare sayısına bölünerek arka planın bulunmasının bir mantığı vardır. Bu mantık ise yayaların hareketli olduğu bir videoya gerekli işlemleri yaparken bulunan toplam matrisinde her bir piksel pozisyonu için arka planın etkisi yayaların etkisinden fazla olacaktır, bu nedenle bölüm sonucunda bulunan değer arka plana yakın olacaktır düşüncesidir.

Bir önceki paragrafta açıklanan durumun gerçekleşmesi için videonun genelinde yayaların hareket etmesi lazımdır. Eğer ki videonun başlangıcından bitişine kadar bir yaya olduğu yerde sabit durmaktaysa yani hareket etmiyorsa o yayada arka planın bir parçasıymış gibi algılanabilir veya o şekilde yansiyabilir.

### **4.3. Yaya Pozisyonlarını Tespit Etmek**

Bu alt bölümde amacımız yaya pozisyonlarını en az hatayla tespit etmek olacaktır ve açıklanacak adımlar her bir video karesi için uygulanacaktır. Öncelikle bu alt bölümde ilk olarak, her bir kare için muhtemel yaya pozisyonları bulunması işlemi uygulanmıştır. Ardından, çıkan sonuçlar üzerinden düzeltme işlemlerine başlanılmış ve her bir kare düzeltme işlemlerine sokulmuştur.

#### **4.3.1. Muhtemel Yaya Pozisyonlarının Bulunması**

Bu aşamada videodaki hareket eden her şeyi yaya olarak düşünmekteyiz. Bu eylemi fark etmek için her bir karenin renk değerlerini renkli tonlardan gri tonlara çevirip arka plandan farkını buluruz ve bu işlemin sonucunda aşağıdaki Şekil 4.5 ve Şekil 4.6'daki resimlere benzer sonuçlara ulaşılmıştır.





**Şekil 4.5 İlk Video'nun Video Karesiyle-Arka plan Farkı**



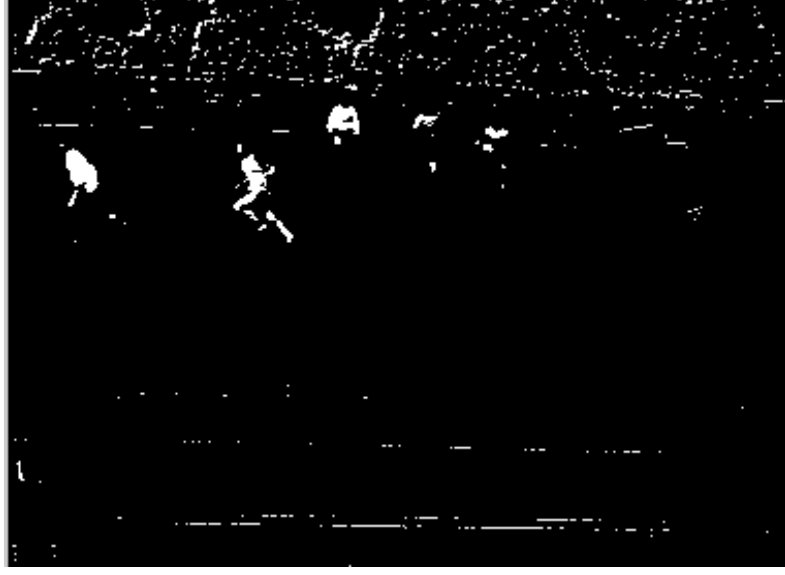
**Şekil 4.6 İkinci Video'nun Video Karesiyle-Arka plan Farkı**

Burada elde edilen arka plan ve kare matrisleri değerleri 0 ile 1 arasındaki değerlerinden oluşmaktadır ( bilindiği üzere matlab programı matrisler (matrix) yani diğer bir deyişle diziler (array) ile çalışır.). 0 siyahı, 1 beyazı, aradaki diğer kalan değerler ise gri renk tonlarını ifade eder. Yukarıdaki işlemler sonucunda

ulařılan matris ierisindeki deęerler -1 ile +1 arasındadır ve bunlar genel olarak -0.5 ile +0.5 aralıęında bulunmaktadırlar. 0 deęerinin altındaki deęerlerin bizim iin bir anlamı yoktur ve bu nedenle bunlar herhangi bir resimdeki(karedeki) 0 deęeri gibi siyah olarak gzkeceklerdir. Bu durumdan dolayı burada iki farklı durumdan biri yapılabilirdi. Bunlardan ilki matristeki 0 deęerinin altında bulunan deęerler sıfır'a eřitlenip sıfır'ın zerindeki deęerlere dokunulmazdı ya da ikinci olarak bu deęerler 0 ile 1 arasında leklenebilirdi. Bu kısımda ikincisi tercih edilmiřtir. Bunun nedeni ikinci yntemde daha az kayıp yařanmasıdır. Sonu olarak, kısacası deęerler 0 ile 1 arasında leklendirilmiřtir.



**řekil 4.7 İlk Video'nun leklendirilmiř Fark verisi**



**Şekil 4.8 İkinci Video'nun Ölçeklendirilmiş Fark verisi**

Video görüntüsünü kullanılan videonun içinde makul sayıda yaya olduğunu düşünmekteyiz. Buradaki makul sözcüğü ile demek istenilen anlam, video içerisindeki yayaların sayısının aşırı fazla sayıda olmaması ve anlık bir kareye bakıldığında arka planı temsil eden piksellerin yayaları temsil eden piksellerden daha fazla olduğunun varsayılmasıdır. Bu nedenle yukarıdaki işlemlere göre oluşan matrise K-means algoritması uygulanarak iki farklı gruba bölünmüştür. Bu gruplardan eleman sayısı daha az olan grubu, yayaları ifade eden piksel olarak düşünülüp değerleri bir ve diğer grubu da arka planı ifade pikseller olarak düşünülüp değerleri sıfır yapılmıştır. Sonuç olarak aşağıdaki Şekil 4.9 ve Şekil 4.10'daki benzer sonuçlara ulaşılmıştır.



**Şekil 4.9 İlk Video'nun İkili Veriye çevrilmiş Fark Verisi**



**Şekil 4.10 İkinci Video'nun İkili Veriye çevrilmiş Fark Verisi**

Yukarıdaki işlemler her bir kareye yapıldığında her bir karede muhtemel yaya pozisyonları 1 ile doldurulmuş olacaktır. Yalnız bunların bir kısmı gürültüdür. Bu gürültüler genellikle video sıkıştırması sonucu oluşmuş bozulmalardan veya dışarıda meydana gelen seslerden oluşmaktadırlar. Bundan sonraki aşamalarda bu hatalar optimize bir şekilde giderilmiştir.

### 4.3.2. Hataların Giderilmesi

Yukarıda bir önceki bölümün son aşamalarında bahsedilen yaya olmayan yerlerin yaya gibi 1 (beyaz) işaretlendiği pozisyonların dışında başka bir takım problemlerle daha karşılaşmıştır. Bu problemler ise yaya olan yerlerde bir takım kırılmalar olması sonucu, yayanın bir parçası değilmiş gibi 0 (siyah) olarak işaretlenmiş konumlarda bulunmaktadır. Bu gibi hatalı olarak işaretlenmiş konular, aşağıda anlatılan yollarla çoğu düzeltilmiştir.

İlk olarak video görüntülerinde oluşan fazlalıkların giderilmesi gerekmektedir. Fazlalıklar çoğunlukla ve videonun kalitesi arttıkça neredeyse her zaman yayalardan küçük olacaktır. Bu nedenle seçilmiş belli bir boyun altında olan 1 ile işaretlenmiş kümeler kir olarak kabul edilmiş ve bu bölgeleri kaldırılmıştır. Seçilen parametrenin belirlenmesi çok kritiktir çünkü eğer video görüntülerinde bu parametre değerini, yaya objesinin temsil ettiği piksellerden daha büyük seçilirse yayayı veya yayanın belli bir takım yerlerini kaybedebiliriz veya tam tersi durumda gerektiğinden küçük bir değer seçilirse bazı gürültüler, fazlalıklar o anki karede kalabilir. Örnek olarak aşağıdaki Şekil 4.11 ve 4.12’ deki resimlerde, Şekil 4.9 ve 4.10’ deki resimlerde gözüken kirler yok olmuştur. Eğer büyük çapta bir obje büyüklüğünde( tabi ki yayanın büyüklüğünden az olmalı) bir gürültü olsaydı o da yöntem tarafından giderilecekti fakat öyle bir durum sonuçlarımızda görülmemektedir.



**Şekil 4.11 İlk Video'nun Fazlalıkları Atılmış Karesi**

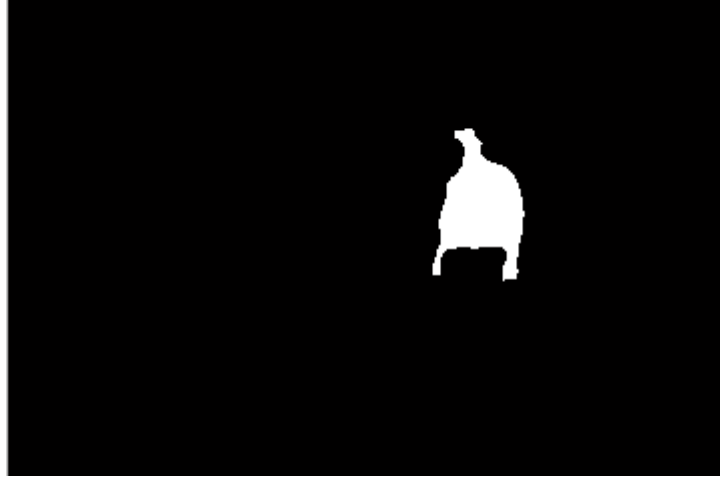


**Şekil 4.12 İkinci Video'nun Fazlalıkları Atılmış Karesi**

Bu yöntem kullanıldığı zaman belli başlı hatalar oluşmaktadır. Bunların ilki, yaya bir önceki paragrafta bahsettiğimiz tipte bir hatanın yakınlarına gelince hata yayanın bir parçasıymış gibi gözükmekte ve bu hatalı bölge yaya ile birlikte 1 olarak işaretlenmektedir.

Başka bir problem ise yayalar videonun biraz uzağında kalırlarsa yani biraz gerilerde var olurlarsa veya belli açılar, belli durumlar meydana geldiğinde yayalar genel boyutlardan küçük gözükmedirler. Bunun sonucu olarak, bu durumlarda gerektiği gibi yaya olarak işaretlenmiş bu konumlar kir olarak algılanıp 0 ile işaretleneceklerdir.

Bu düzeltmelerin ardından çok az nadirde olsa da rastlanan başka bir hatada yayaların vücutlarının belli bölgelerinde boşlukların bulunmasıdır. Bunun nedeni yayanın boş olarak işaretlenmiş bölgesiyle hemen arkasındaki konumun renginin benzer olmasından kaynaklanmaktadır. Bu tipte hataları gidermek için morfolojik işlemler kullanılarak yayanın boş kalan bölgelerinin doldurulmasını sağlanmıştır. Aşağıda Şekil 4.13 ve Şekil 4.14'deki resimlerde bu yöntemle düzeltilmiş hatalar görülmektedir.



**Şekil 4.13 İlk Video'nun Morfolojik İşlem Uygulanmış Hali**



**Şekil 4.14 İkinci Video'nun Morfolojik İşlem Uygulanmış Hali**

Yalnız bu uyguladığımız morfolojik düzeltme işlemi yayalar birbirine yaklaştıkları zaman tek bir insan olarak kabul edebilmekte ve yayaların arasını belli

durumlarda birleştirebilmektedir. Bu durumda sonuçta bir hata olarak karşımıza çıkabilmektedir.

Yukarıdaki uyguladığımız işlemler sonucu yayaların konumları her bir kare için, mümkün olduğunca az hatayla saptanmıştır. Bu işlem sonucu bulunan karelerle, yapılan yoruma göre çeşitli veri paketleri oluşturularak, kazaların oluşmasını engel olacak sistemleri bu yöntem ile uygun hale getirilip, yerleştirilebilir.

#### **4.4. Video Üzerinde Yayaların İşaretlenmesi**

Yapılan işlemler sonucu her bir kare üzerindeki yayaların olduğu konumlar bulunmuştur. Bu pozisyonlarla önceden okuduğumuz girdi videosunun her bir karesi işaretlenmiş ve bu karelerden tekrar video oluşturulmuştur. Ardından, yayaların işaretlendiği videoya ulaşılmıştır. Sonuç olarak; gerekli işlemler uygulanmış, yayalar işaretlenmiş ve yayaların işaretlendiği video kaydı oluşturulmuştur. Aşağıdaki Şekil 4.15 ve Şekil 4.16'daki resimlerde bulunan yeni oluşturulan video görüntüsü kareleri, ilk girdi göstergesi olarak gösterdiğimiz karelerin anlatılan süreçten geçerek işlenmiş halleridir.



**Şekil 4.15 İlk Video'nun Oluşturulmuş Video Görüntüsünden Bir Kare**





**Şekil 4.16 İkinci Video'nun Oluşturulmuş Video Görüntüsünden Bir Kare**

#### **4.5. Yeni Yöntemler(Algoritmalar) Geliştirilmesi ve Bunların Denenmesi**

Bu alt bölümde, dördüncü bölümün başından itibaren anlatılan bölümlerde ( 4.1 - 4.4 bölümleri) anlatılan algorithmadan (yöntem) başka farklı algoritmalar denenmiştir. Bu algoritmalar sırasıyla bu bölümde anlatılmıştır. Bu bölümde her bir algoritma işlemi bulunduktan sonra baştaki bölümler tekrarlanmıştır. (4.1 - 4.4 bölümleri tekrarlanmıştır.)

Sonuç olarak, bu bölümde yukarıda bahsedilen yeni algoritmalar için yapılan işlemler sayesinde yeni algoritmalar oluşturulup, ardından denenmiş ve sonuç olarak diğer denenilen bütün algoritmalarla karşılaştırılmıştır.

#### 4.5.1. Geliştirilen Yöntemler(Algoritmalar)

Bu bölüm içerisinde yukarıdaki bölümlerde anlatılan algoritmanın dışında birkaç algoritma oluşturulup denenmiştir.

Bu bölümde ilk algoritmamızdaki işlemler, her bir video'dan alınan video kareleri üzerinde, bu video karelerinden önceki onaltı, sekiz, dört veya bir kare kullanarak işlemler gerçekleştirilmiştir.

Bu algoritmada ilk olarak, video karesi okunur. Ardından, videodaki her bir video karesi kendinden önceki onaltı, sekiz, dört veya bir video karesinin toplamının onaltı, sekiz, dört veya bir'e bölümünden elde edilen video karesinden çıkarılarak yeni video kareleri elde edilir. Daha sonraki adımlarda ise bu yeni video kareleri üzerinde daha önceki bölümlerde yapılan muhtemel yaya pozisyonları bulunması (4.3.1 bölüm), görüntü düzeltme (4.3.2 bölüm) işlemleri ve okunan video üzerine tespit edilen konumları işaretleme (4.4 bölüm) işlemleri yapılır. Son olarak, elde edilen sonuçlar diğer algoritmalarla karşılaştırılır.

İlk algoritmamızda videodaki her bir video karesi, kendinden önce alınan onaltı, sekiz, dört veya bir video kare değerleri içerisinde alınan en iyi sonuç video karesinin kendinden önce alınan sekiz video karesi değeri olmuştur.

İkinci algoritmamızda işlemler, her bir video'dan alınan video kareleri üzerinde, bu video karelerinden önceki 1/16, 1/8 veya 1/4 video kareleri kullanarak işlemler gerçekleştirilmiştir.

İkinci algoritmamızda ilk olarak, video karesi okunur. İkinci adım olarak, videonun içerik (info) bilgisinden video hakkında saniye başına düşen çerçeve sayısı (fps (frame per second) ) bulunur. Ardından, 4.2 numaralı denklemdeki formülden X sayısı hesaplanır ve bulunan X sayısı eğer ondalıklı ise yuvarlanır.

$$X = \frac{\text{saniye başına düşen çerçeve sayısı(fps)}}{\text{seçilen rakam değeri (16,8,4)}} \quad (4.1)$$

Üçüncü adım olarak, çalışılan video karesinden x kadar önceki video karesi bulunur. Sonra bu bulunan video karesinden x kadar daha önceki video karesi bulunur ve bu işlem seçilen rakam değeri kadar devam eder. Yukarıda anlatılanlara örnek vermek olarak, seçilen rakam değeri sekiz ve saniye başına düşen çerçeve sayısı 15 ise X değerimiz iki elde edilir. Ardından çalışılan video karesi yirminci kare ise algoritmamızda kullanmak üzere onsekiz, onaltı, ondört, oniki, on, sekiz, altı ve dört değerli video kareleri alınır.

Dördüncü adım olarak, bulunan bu video kareleri toplanır ve ardından seçilen rakam değerine bölünerek bir video karesi elde edilir. Sonra elde edilen bu video karesi çalışılan video karesinden çıkarılarak bir sonuç video karesi elde edilir. Daha sonraki adımlarda ise bu elde edilen yeni video kareleri üzerinde daha önceki bölümlerde yapılan muhtemel yaya pozisyonları bulunması (4.3.1 bölüm), görüntü düzeltme (4.3.2 bölüm) ve okunan video üzerine tespit edilen konumları işaretleme (4.4 bölüm) işlemleri yapılır. Son olarak, elde edilen sonuçlar diğer algoritmalarla karşılaştırılır.

Üçüncü algoritmamızda işlemler, her bir video'dan alınan video kareleri üzerinde, bu video karelerinden önceki on video karesi kullanılarak işlemler gerçekleştirilmiştir.

Bu alitmada ilk olarak, video karesi okunur. İkinci olarak, on tane ağırlık rastgele seçilir. Ardından çalışılan video karesinden önceki on video karesi seçilen ağırlıklarla çarpılıp toplanılarak bir video karesi elde edilir. Seçilen ağırlıklar bir ile yüz arasında seçilmiştir. Sonra elde edilen video karesi toplam ağırlık sayısına bölünüp sonuç video karesi elde edilir. Daha sonraki adımlarda ise bu yeni video kareleri üzerinde daha önceki bölümlerde yapılan muhtemel yaya pozisyonları bulunması (4.3.1 bölüm), görüntü düzeltme (4.3.2 bölüm) ve okunan video üzerine tespit edilen

konumları işaretleme (4.4 bölüm) işlemleri yapılır. Son olarak, elde edilen sonuçlar diğer algoritmalarla karşılaştırılır.

#### **4.6. Daha Önce Yapılan Farklı Benzer Uygulamaların Bizim Tarafımızdan Araştırılması ve Gerçekleştirilmesi**

Bu bölümde daha önce yapılan farklı benzer uygulamaların bizim tarafımızdan araştırılması ve gerçekleştirilmesi işlemi yapılacaktır. Bu uygulamalar iki makale üzerinden alınıp gerçekleştirilmiştir.

Bizim tarafımızdan gerçekleştirilen daha önce yapılmış farklı benzer uygulamalar, makaleden yararlanılarak gerçekleştirildiği için yöntemlerde kullanılan ve en optimize sonucu veren spesifik değerler ve algoritmalar genel olarak bizim tarafımızdan denenerek elde edilmeye çalışılmıştır.

##### **4.6.1. Histogram of Oriented Gradients (HOG)**

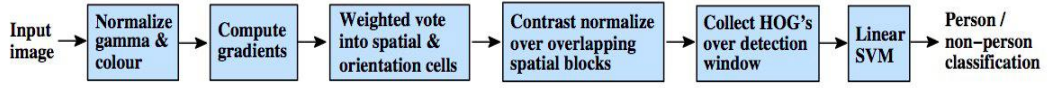
Tanımsal olarak, histogram of gradient (HOG) bilgisayarla görü(computer vision) ve görüntü işleme(image processing) alanlarında bir özellik (öznitelik) tanımlayıcısıdır.

Bu yöntemin bulucuları Navneet Dalal ve Bill Triggs'dir ve 2005 yılında makale haline getirilmiştir.

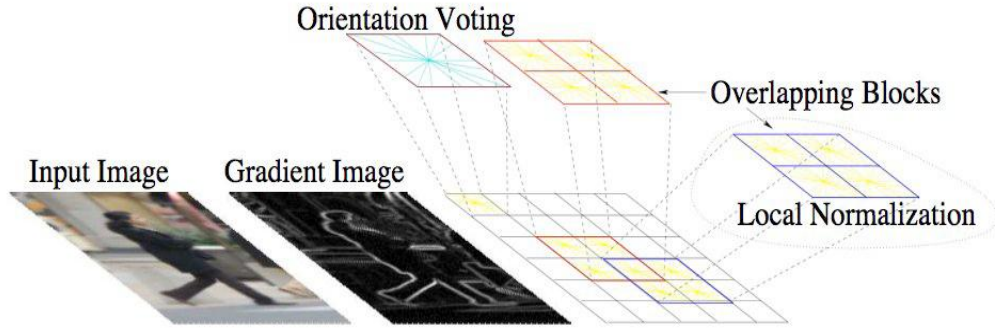
Bu teknik bir resmin veya bir resim karesinin belli bir parçasındaki değişim oryantasyonlarını hesaplar. Bu metod ölçek bağımsız öznitelik dönüşümü (scale invariant feature transform) ve şekil içerik (shape context) gibi bazı öznitelik tanımlama (feature descriptor) yöntemleriyle benzer yönler taşır. Fakat bu teknik düzgün aralıklı hücre ızgaraları(cell grid) arasında çalıştırıldığı için diğer öznitelik tanımlama yöntemlerinden ayrılır.

Histogram of Gradient (HOG)'da yapılan işlemleri aşağıdaki Şekil 4.17 ve Şekil 4.18'de görülen resim üzerinden kısaca anlatılırsa; ilk olarak renk normalizasyonu ve gamma doğrulaması işlemleri yapılır. Ardından konvolüsyon (convolution) metodu

ile gradient değerleri hesaplanmıştır. Sonraki aşamada ise her bir piksel ağırlıklı ortalamaya tutularak oryantasyon işlemi gerçekleştirilir. Daha sonra, çakışan blokların önlenmesi amacıyla normalizasyon işlemi gerçekleştirilir ve son olarak her bir video tespit karesindeki HOG bileşenleri toplanır. Toplanan bu bileşenler daha sonra destek vektör makinesi (support vector machine (SVM)) tanımlayıcısına verilerek eğitilir ve bu işlem sonrasında videoda veya resim karesinde yaya olup olmadığı belirlenir.



**Şekil 4.17 HOG Tanımı Resmi 1 (Şekildeki resim [5]'den alınmıştır.)**



**Şekil 4.18 HOG Tanımı Resmi 2 (Şekildeki resim [5]'den alınmıştır.)**

Yukarıda bu tekniğin genel bir bakışı yapılmıştır. Aşağıdaki bölümlerde burada kısaca bahsedilen terimler detaylandırılarak anlatılacaktır.

#### 4.6.1.1. Renk Normalizasyonu ve Gamma Doğrulaması (Colour Normalization and Gamma correction)

Renk normalizasyonun işleminde RGB ve LAB renk uzayları (RGB ve LAB space) eşitlenir. Renk normalizasyonu işlemindeki amaç videoda ses, ışık, aydınlanma gibi nedenlerden dolayı ortaya çıkan problemlerin en aza indirgenmesidir.

Renk normalizasyonu işlemi iki şekilde yapılabilir. Bu yöntemler histogram eşitleme ve matematiksel hesaplama işlemidir. İlk olarak, histogram eşitlemede yönteminde renk dağılımı yapılır. Bu yöntemde renk histogramında yoğunlukta olan renk tonları değerleri resme yayılır. İkinci olarak, matematiksel hesaplama yöntemi ise her bir video karesindeki kırmızı, mavi, yeşil renk değerlerinin; toplam kırmızı, mavi, yeşil renk değerine oranlanması ile bulunur (  $R = R/(R+G+B)$ ,  $G = G/(R+G+B)$ ,  $B = B/(R+G+B)$  ). Aşağıdaki Şekil 4.19'daki resimde sol tarafta herhangi bir işleme sokulmamış resim karesi bulunmaktadır, sağ tarafta ise bu resmin renk normalizasyonuna uğramış hali görülmektedir.



**Şekil 4.19 Renk Normalizasyonu Örnek Resim**  
(Şekildeki resim [19]'den alınmıştır.)

Gamma doğrulaması işlemi bir videoda veya resim sistemlerinde aydınlanma veya tristimulus (insan gözünün renkleri algılama işlemi) matematiksel olarak simüle

edebilmek gayesiyle yapılan bir takım hesaplar sonucunda elde edilen X, Y ve Z renk koordinat bilgileri) verilerinin şifrelenmesini veya çözümlenmesini sağlar. Aşağıdaki Şekil 4.20'deki resimde sol tarafta gamma değeri yüksek bir gül resmi bulunmaktadır. Burada gamma doğrulaması işlemi yapılarak resimde gamma değeri düşürülmüş ve böylece daha net görüntüler elde edilmiştir. Şekil 4.20'de görüldüğü üzere ortadaki resimde gül resmi biraz daha açılmış fakat sağ taraftaki resimde en net görüntüsüne (ideal gamma değerine) ulaşmıştır.



**Şekil 4.20 Gamma Doğrulaması Örnek Resim**  
(Şekildeki resim [17]'den alınmıştır.)

#### **4.6.1.2. Gradient Çalıştırılması (Computing Gradient)**

Birçok öznitelik tanımlayıcısı ilk adım olarak 4.6.1.1'inci bölümde bahsedilen renk normalizasyonu ve gamma doğrulaması işlemini yapmaktadır. Fakat burada performansa çok az etkisi olduğundan dolayı bu işlem yapılmamıştır.

HOG yönteminde ilk adım olarak gradient değerleri hesaplanmıştır. Gradient değerleri, her bir video karesi için konvolüsyon (convolution) metoduyla yatay ve dikey yönlerde ayrık türev maskeleri (filtreleri) uygulanarak elde edilmiştir. Konvolüsyon (convolution) metodunun kullandığı formül bir sonraki sayfadaki 4.2 numaralı denklemde gösterilmiştir.

$$\begin{aligned}
y[1,1] &= \sum_{j=-\infty}^{\infty} \sum_{i=-\infty}^{\infty} x[i,j].h[1-i,1-j] & (4.2) \\
&= x[0,0].h[1,1] + x[1,0].h[0,1] + x[2,0].h[-1,1] + x[0,1].h[1,0] + \\
&\quad x[1,1].h[0,0] + x[2,1].h[-1,0] + x[0,2].h[1,-1] + x[1,2].h[0,-1] + \\
&\quad x[2,2].h[-1,-1]
\end{aligned}$$

Burada birden çok ayrıık türev maskesi (filtresi) kullanılmıştır. Bu filtrelere sobel filtresi, kernel filtresi ve diagonal filtresi örnek verilebilir. Fakat bu filtreler içerisinde en iyi sonucu kernel filtresi verilmiştir. Aşağıdaki Şekil 4.21'deki resimde, sol taraftaki 3x3 girdi matrisine kernel filtresi ile birlikte konvolüsyon işlemi uygulanmıştır ve en sağ taraftaki 3x3 çıktı matrisi elde edilmiştir.

1	2	3	-1	-2	-1	-13	-20	-17
4	5	6	0	0	0	-18	-24	-18
7	8	9	1	2	1	13	20	17
<b>Girdi Matrisi</b>			<b>Kernel Filtresi</b>			<b>Çıktı Matrisi</b>		

**Şekil 4.21 Konvolüsyon İşlemi Örneği**

Yukarıda anlatılanlar doğrultusunda gradient değerleri hesaplanmıştır ve bu işlem sonucunda oluşan sonuçlar Şekil 4.22, 4.23, 4.24, 4.25, 4.26, 4.27'deki resimlerde görülmektedir.





**Şekil 4.22 İlk Videonun Kernel Filtresine Yatay Uygulanması Sonucu Oluşan Resim Karesi**



**Şekil 4.23 İlk Videonun Kernel Filtresine Dikey Uygulanması Sonucu Oluşan Resim Karesi**



**Şekil 4.24 İlk Videonun Kernel Filtresine Hem Yatay Hemde Dikey Uygulanması Sonucu Oluşan Toplam Resim Karesi**



**Şekil 4.25 İkinci Videonun Kernel Filtresine Yatay Uygulanması Sonucu Oluşan Resim Karesi**



**Şekil 4.26 İkinci Videonun Kernel Filtresine Dikey Uygulanması Sonucu Oluşan Resim Karesi**

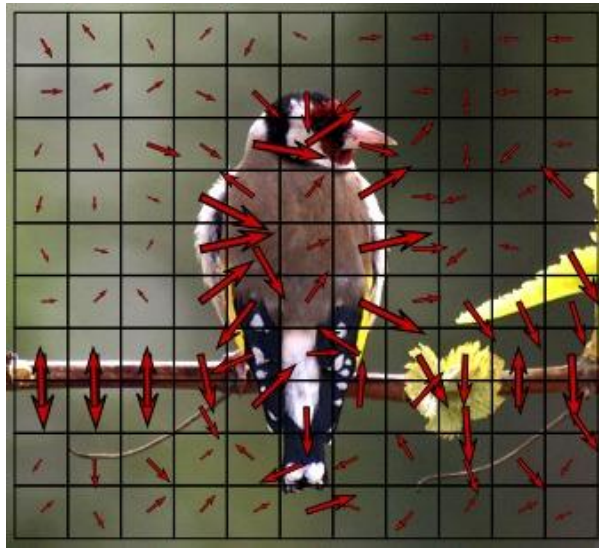


**Şekil 4.27 İkinci Videonun Kernel Filtresine Hem Yatay Hemde Dikey Uygulanması Sonucu Oluşan Toplam Resim Karesi**

Şekil 4.22 ve Şekil 4.25’de, iki videodan alınan resim karelerinin gradient değerlerinin hesaplanması için konvolüsyon işlemi sırasında kernel filtresinin yatay yönde uygulanması sonucu ortaya çıkan sonuç resim karelerini göstermektedir. Ayrıca aynı işlemde Şekil 4.23 ve Şekil 4.26’da, kernel filtresinin dikey uygulanması sonucu ortaya çıkan resim kareleri görülmektedir. Son olarak ise aynı işlemde Şekil 4.24 ve Şekil 4.27’de ise kernel filtresinin hem yatay hem de dikey yönde uygulanması sonucu oluşan toplam resim karesi sonuçları görülmektedir.

#### 4.6.1.3. Orientation Binning

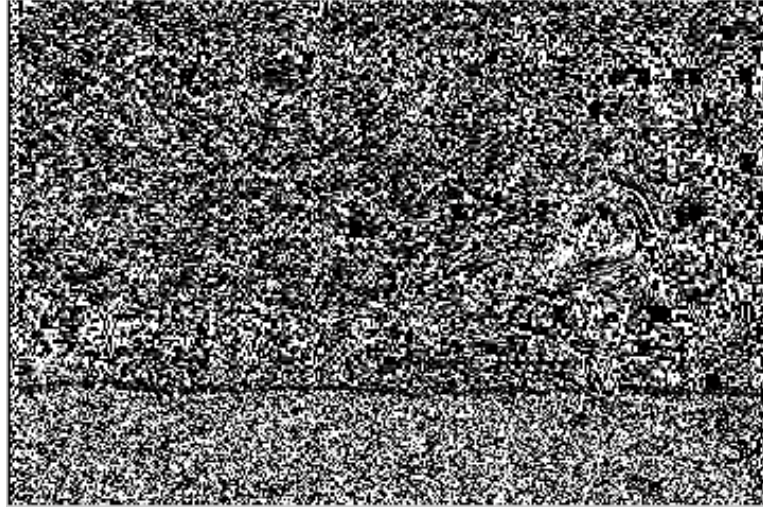
Orientation binning işleminde, hücre histogramları oluşturulmuştur. Hücre histogramlarının oluşturulması için burada hücre dökümü içerisindeki her bir pixel daha önce hesaplanan gradient değerlerine bağlı olarak orientation-based histogram için ağırlıklı oylamaya (weighted vote) tutulmuştur. Aşağıda Şekil 4.28’deki resimde orientation-based histogram için şekillendirilmiş bir örnek gösterilmektedir. Bu resimde her bir hücrenin, çeşitli matematiksel hesaplamalar sonucunda elde edilen yönelimlerini görebilmekteyiz ve görüldüğü üzere kuş varlığının üzerinde oklar yoğunlaşmış görülmektedir; bu yoğunluk bize orada bir objenin var olduğunu belli etmektedir.



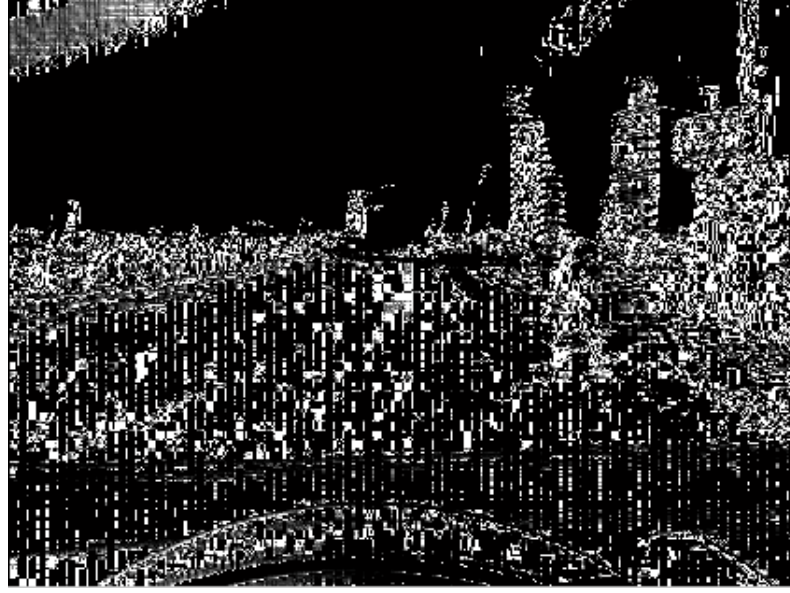
Şekil 4.28 Orientation-Based Histogram (Şekildeki resim [16]’den alınmıştır.)

Bu işlemde oluşturulan hücreler karesel (rectangular) veya dairesel (radial) şekilde olabilir. Ayrıca, histogram kanalı gradient değerlerinin işaretli veya işaretsiz oluşuna göre (signed veya unsigned) 0-180 derece arasına veya 0-360 derece arasına yayılmıştır. Dalal ve Trigs çalışmalarında işaretsiz gradient değerlerinde 9 histogram kanalı ile birlikte en performanslı sonucu alınmıştır [5].

Yukarıda anlatılanlar doğrultusunda oryantasyon işlemi uygulanmıştır ve oryantasyon işlemi sonucunda oluşan sonuçlar Şekil 4.29 ve Şekil 4.30'deki resimde görülmektedir.



**Şekil 4.29 İlk Videonun Oryantasyon İşlemi Sonrasındaki Ekran Görüntüsü**



**Şekil 4.30 İkinci Videonun Oryantasyon İşlemi Sonrasındaki Ekran Görüntüsü**

#### **4.6.1.4. Normalizasyon İşlemi**

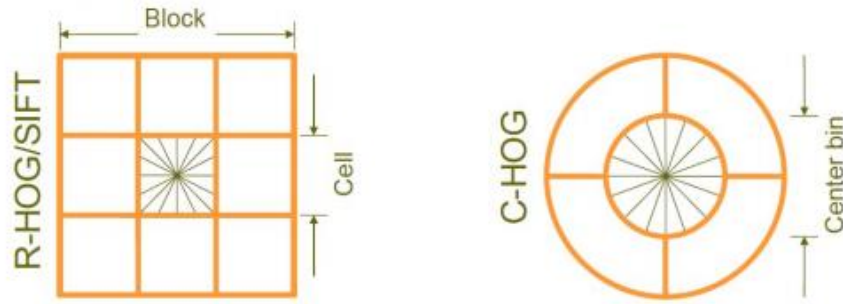
Normalizasyon işlemi, herhangi bir videoda yapılan işlemler sonucunda oluşan problemlerde aydınlanma ve kontrast değişikliklerine karşı hassasiyeti en aza indirmek için yapılan bir işlemdir.

Normalizasyon işlemleri yapılmadan önce hücreler büyük gruplar halinde bir araya getirilirler, bu büyük gruplara mekansal-bağlı bloklar (spatially-connected blocks) denir. Bloklama işleminin yapılmasındaki asıl amaç çakışan blokların önlenmesi amacıyla. Eğer ki çalışmamızda böyle bir sorunun olması demek birden fazla aynı final tanımlayıcısı tanımlamamız demek olacaktır ve bu durum ise bizim için büyük bir problem'dir.

İki tür bloklama geometrisi vardır. Bu bloklama geometrileri R-HOG (Rectangular HOG ) ve C-HOG (Circular HOG)'dur. R-HOG hücrelerin karesel sıralanmasıdır.

Bu bloklama geometrisinin üç parametresi vardır; bunlar blok başına hücre sayısı, hücre başına piksel sayısı, hücre başına kanal(channel) sayısı'dır.

C-HOG (Circular HOG) hücrelerin dairesel sıralanmasıdır. Bu bloklama geometrisinin dört parametresi vardır. Bunlar açısal ve radyal binlerin sayısı, merkezi bin'in yarıçapı, ek radyal bin'in yarıçapı için genişleme faktörü'dür. Aşağıdaki Şekil 4.31'deki resimde R-HOG ve C-HOG'un şekilsel hali görülmektedir.



**Şekil 4.31 R-HOG ve C-HOG Şekli (Şekildeki resim [3]'den alınmıştır.)**

Dalal ve Trigs çalışmalarında dört tane normalizasyon denklemi üzerinde denemeler yapmışlardır. Bunlar L2-norm, L2-hys, L1-norm ve L1-sqrt denklemleri'dir. Bu denklemler aşağıdaki 4.3, 4.4, 4.5 numaralı denklemlerde sırasıyla gösterilmiştir. L2-hys denklemi burada gösterilmemiştir çünkü L2-hys denklemi L1-norm denkleminin kırılması sonucu elde edilen bir denklemdir (L2-hys denkleminde,  $v$  değeri maksimum 0.2 değerini alınmıştır). Bu denklemlerde kullanılan  $v$  değeri, verilen bloktaki bütün histogramları içeren normalize olmayan vektör'dür ve  $\|v\|_r$  terimi ise  $v$  teriminin  $r$  sayısı normalizasyonu ( $r$ -norm) demektir. Burada  $e$  ise önemsiz küçük bir sabit değerdir.

Bu normalizasyon denklemlerinin denenmesi sonucunda L2-norm, L2-hys, L1-norm denklemleri birbirleriyle benzer sonuçlar aldıkları görülürken, L1-norm denklemini diğerlerine nazaran daha az performanslı sonuçlar göstermiştir [5].

$$\text{L2 - norm: } f = \frac{v}{\sqrt{\|v\|_2^2 + e^2}} \quad (4.3)$$

$$\text{L1 - norm: } f = \frac{v}{\|v\|_1 + e} \quad (4.4)$$

$$\text{L1 - sqrt: } f = \sqrt{\frac{v}{\|v\|_1 + e}} \quad (4.5)$$

Yukarıda anlatılanlar doğrultusunda bizim tarafımızdan normalizasyon işlemi, hücreler büyük gruplar halinde bir araya getirildikten sonra 4.3 numaralı normalizasyon denklemi kullanılarak her bir video karesinde her bir hücre için uygulanarak gerçekleştirilmiştir.

#### **4.6.1.5. SVM(Support Vector Machine) Tanımlayıcı (Classifier)**

Son aşama olarak obje hatırlama (object recognition) kısmında HOG kullanarak bazı hatırlama sistemlerinde (recognition system) gözetimli öğrenme'ye bağlı olarak bazı tanımlayıcılar oluşturulmuştur.

Support Vector Machine tanımlayıcısı bir binary (ikili) tanımlayıcıdır, bir karar fonksiyonu görevi yaparak optimal hiper düzlemi arar. Herhangi bir videodan alınmış eğitilmiş resimlerde(training images) belirli objeler bulunur, burada svm tanımlayıcısı bu eğitilmiş resimleri kullanarak herhangi bir video karesinde objenin var olup olmadığına karar verir. (Resim karesinde yaya var yada yok kararı verilir.)



Özetle bu bölüm eğitim tanımlayıcısı bölümüdür, sistemimizin daha düzgün karar verebilmesi için burada eğitilmiş veriler elde edilir ve daha sonra bu veriler test datalarında kullanılıp test datasında objenin var olup olmadığı belirlenir.

Yukarıda anlatılanlar doğrultusunda bizim tarafımızdan Support Vector Machine tanımlayıcısı işlemi, matlab'ın kütüphanesinden svm fonksiyonu kullanılarak yapılmıştır. Burada yapılan işlemlerin detayları aşağıdaki paragrafta açıklanmıştır.

İlk olarak matlab kütüphanesinden kullanılan svm fonksiyonuna veri hazırlanır. Bu verinin içeriği ise HOG yöntemine göre elde ettiğimiz gradient ve oryantasyon değerlerinin birleşiminden oluşan bir vektördür. Bu vektörün ilk sütununa video karesinden elde edilmiş gradient değerleri, ikinci sütununa ise video karesinden elde edilmiş oryantasyon değerleri koyulur. İkinci olarak yeni bir vektör daha tanımlanır ve bu vektör yaya ve yaya değil olarak iki gruba sınıflandırılır. Üçüncü olarak, eğitilmiş setler (training data) ve test setleri oluşturulur. Ardından yukarıda üç aşamada elde edilen veriler kullanılarak svm tanımlayıcısı lineer kernel fonksiyonuyla(linear kernel fonksiyon) birlikte uygulanıp eğitilmiş grup verileri elde edilir. Son aşama olarak ise eğitilmiş grup verileri kullanılarak test setleri sınıflandırılır ve böylece işlem tamamlanmış olur.

#### **4.6.2. A Monocular Human Detection System Based on EOH and Oriented LBP Features**

Yaya tespit sistemlerinin performansları büyük ölçüde öznitelik çıkarımına bağlıdır ve öznitelik çıkarımı çoğu akan çerçeve (sliding window) bazlı insan tespit sistemlerinde ilk adım olarak yapılır.

Bu yöntemde amacımız yerleşik kameradan alınan video karelerinden insan tespiti olduğu için bu sistemde öznitelik çıkarımı büyük önem arz etmektedir.

Yukarıda 4.6.1'inci bölümde bahsettiğimiz yöntem içinde yüksek boyutlu şekil özelliklerinin (high dimensional shape features) doğrusal svm tanımlayıcısı ile

birleşmesi sonucunda resim bazlı insan tespit sistemlerinde başarılı sonuçlar alınmıştır, fakat bu yaklaşımlar genellikle yeterince hızlı değildirler. Bundan sebeple bu çalışma içerisinde yüksek boyutlu şekil özelliklerinin yerine ayırım yapan lokal öznitelik tanımlayıcıları kullanılmıştır. Bu lokal öznitelik tanımlayıcısının adına Oriented LBP (local texture descriptor) denir. Oriented LBP çekip çıkartma işlemini hızlıca yapar. Ardından adaboost algoritması ise önerilen özellikleri ve EOH (Edge Orientation Histogram) özelliklerini alarak insan tespit detektörünü eğitir [6]. Bu yöntemde adaboost algoritması kullanılmıştır fakat burada yöntemin uygulaması bizim tarafımızdan gerçekleştirildiğinde adaboost algoritması yerine svm tanımlayıcısı kullanılmıştır.

Yukarıda bu tekniğin genel bir bakışı yapılmıştır. Aşağıdaki bölümlerde burada kısaca bahsedilen terimler detaylandırılarak üç alt aşamada anlatılacaktır. Üç alt aşama öznitelik havuzu, sınıflandırıcı ve tespit doğrulaması adımlarından oluşmaktadır.

#### **4.6.2.1. Öznitelik Havuzu (Feature Pool)**

Bu çalışmada ilk adım olarak, öznitelik havuzu aşaması gerçekleştirilmiştir. Öznitelik havuzunda yapılan işlemler sırasıyla EOH (Edge Orientation Histogram) ve Oriented LBP (Local Binary Patterns)' dir ve bunların detayları aşağıdaki bölümlerde anlatılmıştır.

##### **4.6.2.1.1. EOH (Edge Orientation Histogram)**

Yukarıda 4.6.1'nci bölümde bahsedilen HOG yönteminin çalışmasından elde edilen öznitelikler ile yaya tespitinde güzel sonuçlar alınmıştır fakat yavaş tespit hızıyla işlemlerimiz gerçekleştirilmiştir. Özetle, HOG yöntemi başarılı sonuçlar vermiştir ama hayli yavaş çalışmaktadır. Bundan sebeple bu sistemde çalışma hızını yükseltmek için EOH (Edge Orientation Histogram) yöntemi tercih edilip kullanılmıştır.

EOH yöntemi de, HOG yöntemi gibi gradient-orientation bazlı özelliştir. İkisi arasındaki fark ise HOG özelliği yüksek boyutlu özellik vektörünü kullanarak tespit çerçevesi için gradient-orientation bilgisi sunarken, EOH birkaç bir boyutlu özellik (1-D) kullanır ve her bir özellik için bir blok zamanda bir oryantasyon (orientation) yapar.

EOH yönteminin uygulamasında, ilk olarak EOH özelliğinde kenar tespiti yapılarak başlanılır. Bundan ötürü her bir nokta (x,y) için gradient değerleri ve gradient oryantasyon değerleri sırasıyla 4.6 ve 4.7'nci denklemler kullanılarak elde edilir. Ardından nokta (x,y)'deki gradient değerleri, 4.8 denklem ve Şekil 4.32'deki resimde gösterilen sobel maskesi uygulanarak elde edilir.

$$\sqrt{G_x(x,y)^2 + G_y(x,y)^2} \quad (4.6)$$

$$\theta(x,y) = \arctan\left(\frac{G_y(x,y)}{G_x(x,y)}\right) \quad (4.7)$$

$$G_x(x,y) = \text{Sobel}_x * I(x,y) \quad (4.8)$$

$$G_y(x,y) = \text{Sobel}_y * I(x,y)$$

-1	0	+1
-2	0	+2
-1	0	+1

Gx

+1	+2	+1
0	0	0
-1	-2	-1

Gy

**Şekil 4.32 Sobel Maskesi**

İkinci adım olarak 4.9' uncu denklemde gösterildiği üzere gradient oryantasyonu k tane (bin) aralığa bölünür ve gradient-orientation histogramı Ek (B<sub>i</sub>)'de her bir blok B<sub>i</sub> için k tane aralık (bin) gözlemlenir.

$$E_k(B_i) = \sum_{(x,y) \in B_i} \psi_k(x,y)$$

$$\psi_k(x,y) = \begin{cases} G(x,y) & \theta(x,y) \in bin_k \\ 0 & \text{aksi takdirde} \end{cases} \quad (4.9)$$

Son adım olarak ise 4.10 denklemde tanımlanan, her bir tek bloktaki K tane EOH özellikleri kümesi tek bir oryantasyonun bütün aralık (bin) değerlerinin toplamına oranıyla bulunur ve böylece işlemimiz tanımlanır ve sonlanmış olur.

$$EOH_{B_i}^k = \frac{E_k(B_i) + \epsilon}{\sum_{j=1}^K E_j(B_i) + \epsilon} \quad (4.10)$$

Yukarıda anlatılanlar doğrultusunda EOH (Edge Orientation Histogram) işlemi uygulanmıştır ve EOH işlemi sonucunda oluşan resim karesi Şekil 4.33'de gösterilmektedir.



**Şekil 4.33 EOH İşlemi Sonrasında Oluşan Ekran Görüntüsü**

Ayrıca buna ek olarak, EOH işlemi sonucunda oluşan Şekil 4.33'deki resim karesindeki her bir piksele yaya tespitinde en optimize sonuçları elde etmek için belli eşik değerleri uygulanmıştır. Burada bizim tarafımızdan denenerek elde edilen belli bir eşik değerinin altındaki piksellerin yaya olmadığı kabul edilmiştir ve bu pikseller elenmiştir. Bu işlemin sonrasında elde edilen ekran görüntüsü Şekil 4.34'de görülmektedir. Burada uygulanan eşik değeri işlemi sonucunda, Şekil 4.33'deki resim karesine oranla en az gürültü oranıyla Şekil 4.34'deki resim karesini elde edilmiştir.



**Şekil 4.34 EOH İşlemi Sonrasında Oluşan Ekran Görüntüsüne Belli Bir Eşik Değeri Uygulanması Sonucu Oluşan Ekran Görüntüsü**

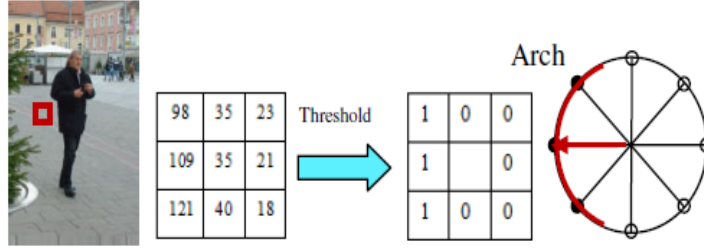
#### 4.6.2.1.2. Oriented LBP (Local Binary Patterns)

Yukarıda 4.6.2.1.1'nci bölümde bahsedilen EOH (Edge Orientation Histogram) özelliği gradient odaklı insan şeklini tanımlayıcı (human shape descriptor) bir özellik iken, LBP (Local Binary Patterns) özelliği lokal nitelik tanımlayıcısı hizmeti sunar.

LBP yönteminin bazı avantajları bulunmaktadır. Bu avantajlara aynı konsept modelini kullanarak gürültülü arkaplanları filtreleme veya çalıştırmada hesaplanabilir verimlilik sağlama durumları örnek verilebilir.

LBP özelliğini hesaplama işlemi için ilk aşamada tespit karesi (detection window) bloklara bölünür ve histogram her bir blok için merkez piksel ile komşu piksellerin yoğunluklarına göre çalıştırılır. Burada, eğer ki komşu piksellerden birinin yoğunluğu merkez pikselden büyükse komşu piksel bir işaretlenmiş, aksi takdirde ise komşu piksel sıfır işaretlenmiştir. Burada bir (beyaz renk) değeri yayanın var olduğu pikseli, sıfır (siyah renk) değeri ise yayanın var olmadığı pikselleri göstermektedir. Ardından bir sonraki aşamada, bütün bloklardan elde edilen LBP modelleri histogramları tespit karesinin niteliğini tanımlamak için sıralanmıştır.

Bu aşamaya kadar LBP özelliğinin düşük boyutlu özniteliklerini hesaplamış bulunmaktayız. Bundan ötürü, LBP özelliğinin yüksek boyutlu özniteliklerini hesaplayabilmek için hücre yapılı LBP özelliği tanımlanmıştır. Hücre yapılı LBP özelliği ile uzun boyutlu ve daha çok anlamlı özellikler elde edilir. Bunun için ilk olarak piksel kavisleri (pixel arch) tanımlanmıştır. Piksel kavisleri (pixel arch) bir pikselin komşularındaki bir bitinin sürekliliğidir. Aşağıda Şekil 4.35'deki resimde gösterilen sol taraftaki resim karesinin, matris değerlerinden alınan 3x3 matris değerine ilk önce LBP özelliği işlemi uygulanmıştır ve içeriği bir veya sıfır değerleri olan 3x3 matris değeri elde edilmiştir. Ardından elde edilen 3x3'lük matris'in piksel kavis (pixel arch) grafiği çizilmiştir ve piksel kavis grafiği resmin sağ tarafında gözlenmektedir.



**Şekil 4.35 Piksel Kavis Grafiği Örneği(Şekildeki resim [6]'den alınmıştır.)**

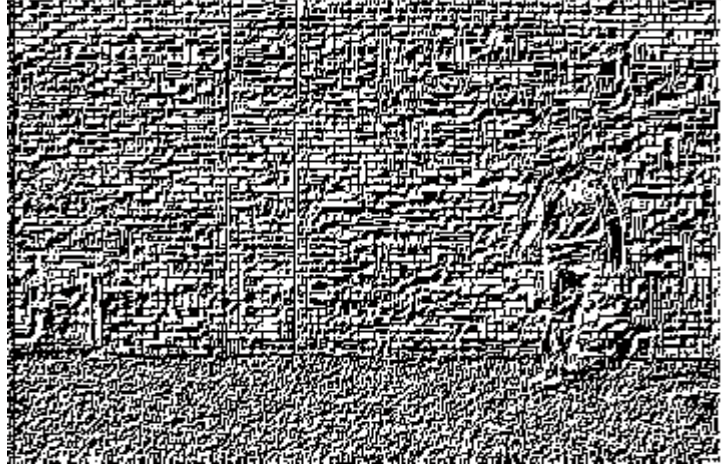
İkinci olarak, piksel yönelimleri (pixel orientation) 0 ile 360 derece arasında  $n$  tane aralığa bölünür. Ardından denklem 4.11'de gösterilen  $F_{i,n}$  oryantasyon histogramı, her bir hücre  $C_i$ 'deki oryantasyon aralıklarının içindeki piksel değerlerinin toplamı ile bulunur.

$$F_{i,n} = \sum_{\substack{(x,y) \in C_i \\ \theta(x,y) \in bin_n}} m(x,y) \quad (4.11)$$

Son adım olarak ise 4.12 denklemde tanımlanan ve EOH yöntemindeki son adıma benzer olarak yapılan her bir bloktaki kurulmuş  $N$  tane Oriented LBP özelliği hesaplanır.

$$O\_LBP_{B_i}^n = \frac{F_{i,n} + \varepsilon}{\sum_{j=1}^N F_{i,j} + \varepsilon} \quad (4.12)$$

Yukarıda anlatılanlar doğrultusunda LBP (Local Binary Patterns) işlemi bizim tarafımızdan uygulanmıştır ve LBP işlemi sonucunda oluşan resim karesi şekil 4.36'de gösterilmektedir.



**Şekil 4.36 LBP İşlemi Sonrasında Oluşan Ekran Görüntüsü**

Ayrıca buna ek olarak, LBP işlemi sonucunda oluşan Şekil 4.36'deki resim karesindeki her bir piksele yaya tespitinde en optimize sonuçları elde etmek için belli eşik değerleri uygulanmıştır. Burada bizim tarafımızdan denenerek elde edilen belli bir eşik değerinin altındaki piksellerin yaya olmadığı kabul edilmiştir ve bu pikseller elenmiştir. Bu işlemin sonrasında elde edilen ekran görüntüsü Şekil 4.37'de görülmektedir. Burada uygulanan eşik değeri işlemi sonucunda, Şekil 4.36'deki resim karesine oranla en az gürültü oranıyla Şekil 4.37'deki resim karesini elde edilmiştir.



**Şekil 4.37 LBP İşlemi Sonrasında Oluşan Ekran Görüntüsüne Belli Bir Eşik Değeri Uygulanması Sonucu Oluşan Ekran Görüntüsü**



#### **4.6.2.2. Sınıflandırıcı (Classifier)**

Bu çalışmada ikinci adım olarak sınıflandırıcı adımı gerçekleştirilmektedir. Bu çalışmada yani [6] numaralı makalede sınıflandırıcı adımında adaboost algoritması kullanılmıştır. Fakat burada bizim tarafımızdan bu adımda HOG işlemi sırasında kullanılan SVM (Support Vector Machine) tanımlayıcısı kullanılarak verilerimiz sınıflandırılmıştır. SVM tanımlayıcısı'nın implementasyon ve tanım detayları HOG (4.6.1. bölüm) işlemi sırasında yapıldığı için burada tekrar aynı işlem üzerinden geçilmeyecektir.

[6] numaralı makalede kullanılan adaboost algoritması (boosting algoritmaları), ağırlıklı oylamaya bağlı olarak oluşturulan zayıf tanımlayıcıları öğrenerek sonuç karar verme mekanizmasını optimize bir şekilde çalıştırır. Her bir adımda seçilen zayıf tanımlayıcılar eğitim hatalarını en aza indirger. Bütün zayıf tanımlayıcıların toplanması sonucu güçlü bir final tanımlayıcısı elde edilir ve sınıflandırma işlemleri bu şekilde tamamlanmış olur.

Boosting algoritmasının, SVM tanımlayıcısı yöntemine karşı birkaç avantajları vardır. İlk olarak yapılar kademeli bir şekilde birleştirildiğinde (cascaded structure) tespit hızı artar ve böylece yayaların olmadığı bölgeler (arkaplanlar) erkenden elenmiş olur. İkinci olarak, bu algorithmada SVM tanımlayıcısı gibi değişik tanımlayıcılar zayıf tanımlayıcı olarak seçilebilir. Son olarak ise, boosting algoritması SVM ile karşılaştırıldığında daha az parametre ayarlaması gerektirir.

#### **4.6.2.3. Tespit Doğrulaması (Detection Validation)**

Bu çalışmada üçüncü ve son adım olarak tespit doğrulaması adımı gerçekleştirilmektedir. Karışık arkaplan, limitli sayıda pozitif veya negatif örneklerin varlığından dolayı bazen yaya tespit sistemlerinde yanlış alarmlar verilebilmektedir. Bu duruma örnek olarak; bir resim karesindeki herhangi bir konumda bir objenin varlığı söz konusu olmadığı halde çalıştığımız sistem bize o konumda bir obje var

diyebiliyorsa bu durum yanlış alarmı bir örnektir. Bundan sebeple, bu gibi durumları önlemek için tespit doğrulaması adımı gerçekleştirilmiştir.

Bu bölümde tespit doğrulamasını gerçekleştirmek için yani obje adaylarının aslında obje olup olmadıklarını öğrenmek için geçici tutarlılık şartı (temporal coherence condition) yapılmıştır. Fakat bu işlem yapılmadan önce 4.13 numaralı denklem kullanılarak resim karesinde belli eşik değerinin altında olan değerler elenmiştir.

$$\text{conf}(x) = \frac{\sum_{i=1}^m \alpha_i h_i(x) + \delta}{\sum_{j=1}^n \alpha_j + \delta} \quad (4.13)$$

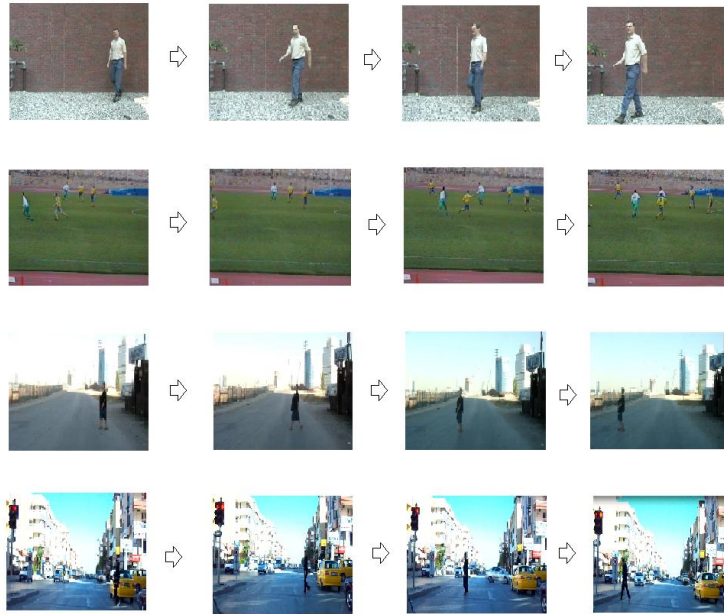
Ardından, geçici tutarlılık şartı yapılmıştır. Bu işlemde; bir çerçevede ilk obje tespit edildiğinde, Kalman filtresi üç kare doğrulama izlemesine başlar. Kalman filtresi bir sonraki çerçevede obje lokasyonunu tahmin eder. Ardından bir sonraki çerçevede bir önceki adımda bulunan obje lokasyonunun bulunduğu yer üst üste bindirilir ve karşılaştırılır. Burada çakışma oranı 0.7 alınmıştır. Eğer ki çakışma oranı en az 0.7 ise burada obje vardır ve pozitif işaretlenmiştir; aksi durumda ise bu konumda obje yoktur ve negatif işaretlenmiştir ve böylece işlemimiz tamamlanmıştır. [6]

## 5. DENEYLER

Bu bölüm içerisinde veri setimizin özellikleri, veri setimizde kullanılan araçlar ve daha önceki bölümlerde bizim tarafımızdan üretilen algoritmaların sonuçlarıyla, daha önce yapılmış ve bizim tarafımızdan gerçekleştirilmiş farklı benzer çalışmaların sonuçlarının deneysel, grafiksel analizi yorumlanarak ve tablolştırılarak anlatılmıştır.

### 5.1. Veri Setimiz Özellikleri ve Kullanılan Araçlar

Veri setimiz içerisindeki video görüntüleri ya bir dijital fotoğraf makinesi tarafından çekilmiş, ya da internetten aynı formata sahip videolar bulunup alınmıştır. Çekim sırasında herhangi bir ayarlama veya sıkıştırma algoritması kullanılmamıştır. Burada 100 tane örnek video üzerinden işlemlerimiz gerçekleştirilmiş ve deneysel karşılaştırmalarımız bu videolar üzerinden yapılmıştır. Aşağıda şekil 5.1'deki resimde veri setimiz içerisindeki 4 tane video'dan ayrı ayrı alınan resim kareleri kesitleri görülmektedir.



Şekil 5.1 Veri Seti Örneği

Videolarımız sabit açıyla çekilmiş olmalıdır yani kullanılacak makine bir sabitleştirici ile sabitlenmiş olmalıdır. Yakınlaştırma, uzaklaştırma veya sağa, sola hareket gibi açı değişiklikleri, işlenecek videoda bulunmamalıdır.

Video görüntü kalitelerinin matlab'ın koyduğu sınırlara uygun olması gereklidir. Videoların her bir pikseli, gri veya indisli karelerden oluşan videolarda 8 bit ile ya da gri veya gerçek renkli karelerden oluşan videolarda 24 bit olmalıdır. (Görüntü işleme aracı olarak Matlab kullanılmıştır. Matlab'ın görüntü işleme araç kutusundan( image processing toolbox ) kullanılmış ve faydalanılmıştır.)

Alınan görüntüleri sıkıştırmak ve video görüntüsünü (bazı videolar için) Matlab ortamındaki koşullara getirebilmek için açık kaynaklı VirtualDub programı kullanılmıştır. (Burada Matlab'da okunabilecek videolar için boyut ve büyüklük sınırlamaları yapılmıştır.)

## **5.2. Elde Edilen Sonuçların Grafikselleştirilmesi ve Deneysel Karşılaştırılması**

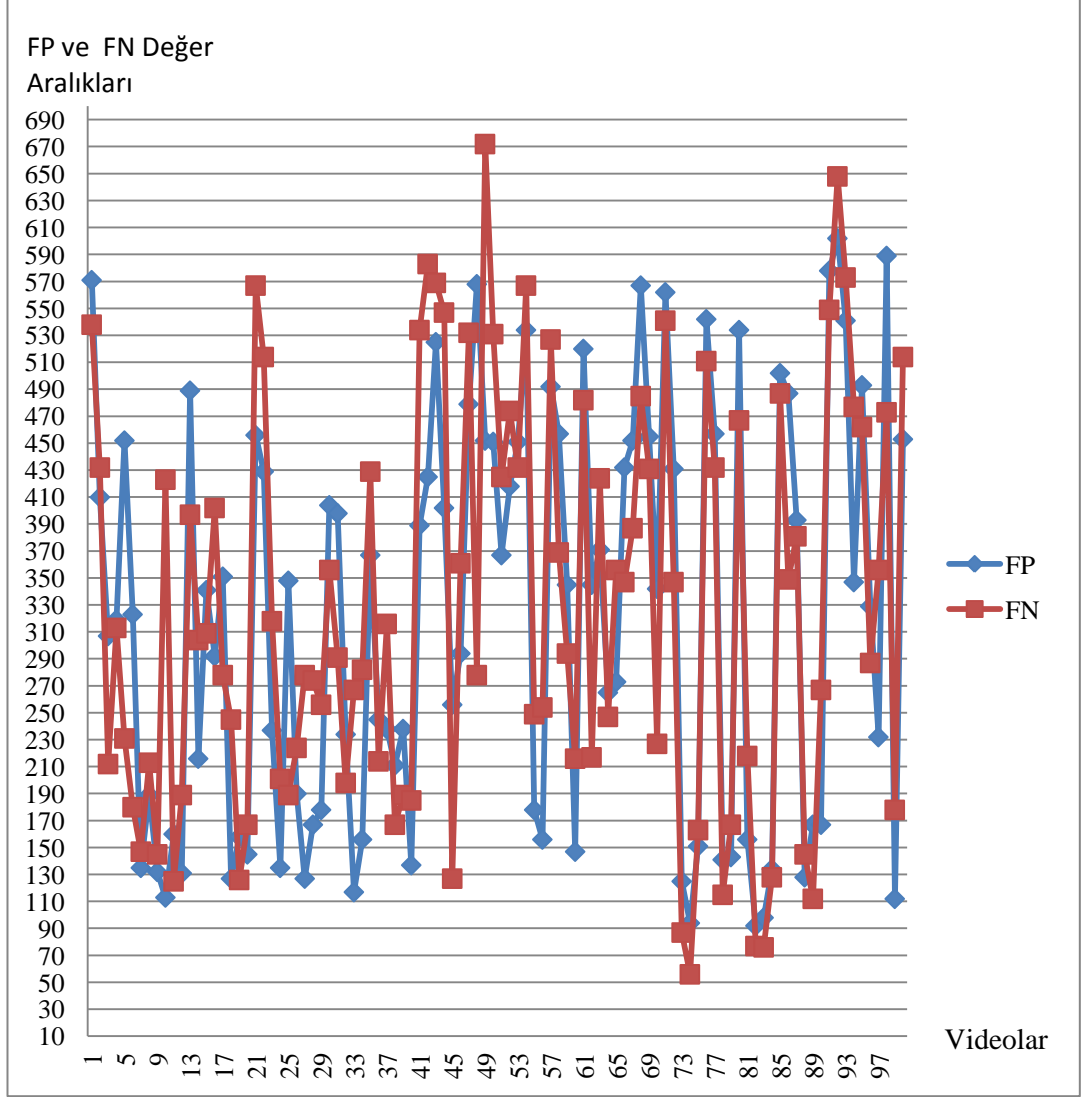
Bu alt bölümde daha önceki bölümlerde bizim tarafımızdan üretilen algoritmaların sonuçlarıyla, daha önce yapılmış ve bizim tarafımızdan gerçekleştirilmiş farklı benzer çalışmaların sonuçlarının deneysel ve grafikselleştirilmesi yapılacaktır. Bu analiz üç aşamada anlatılacaktır. İlk iki adımda çalışılan algoritmanın ürettiği deneysel sonuçlardan bahsedilecektir. Üçüncü ve son adımda ise çalışılan algoritmayla, daha önce yapılmış ve bizim tarafımızdan gerçekleştirilmiş iki yöntem karşılaştırılacaktır.

Karşılaştırma işlemleri için ilk adım olarak algoritmamızın başarısını ölçmek amacıyla yanlış pozitif (False Positive (FP)), yanlış negatif (False Negative (FN)), doğru pozitif (True Positive (TP)) ve doğru negatif (True Negative (TN)) değerleri hesaplanmıştır. Bunun için ilk olarak, veri setimiz içerisindeki her bir video üzerinde gerekli yaya tespit sistem aşamaları gerçekleştirilmiştir. Kısacası her bir videomuzdaki her bir resim karesi üzerinde ürettiğimiz yaya tespit algoritmasını kullanarak yaya olan veya yaya olmayan her bir piksel resim karesi üzerinde tespit edilmiştir. Daha sonra veri setimiz içerisindeki her bir videodan, yaya tespit

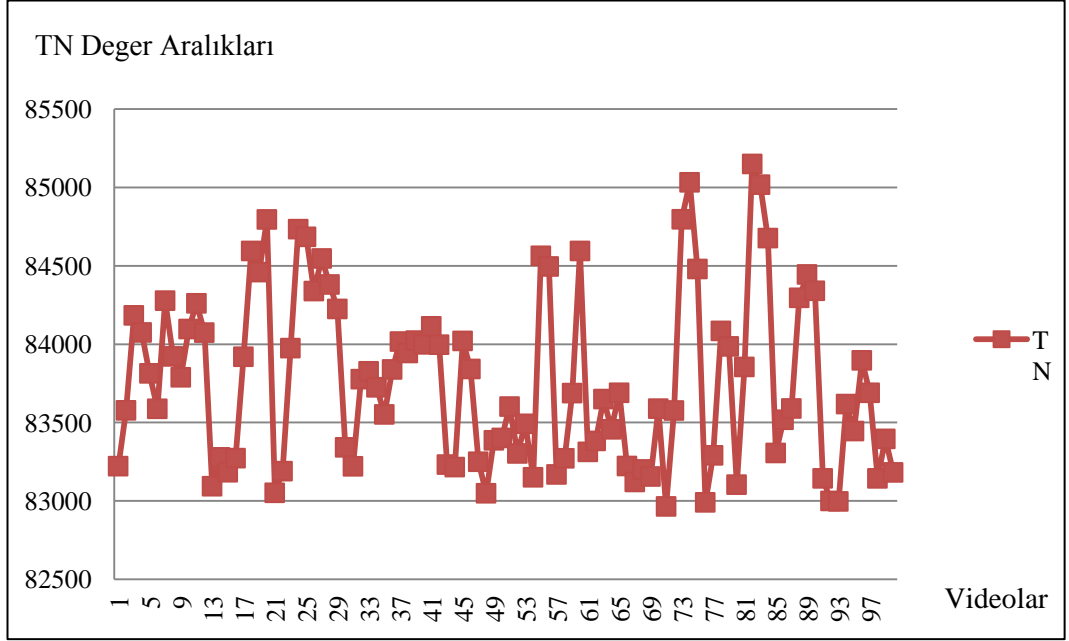
sisteminden geçirilerek işlenmiş bir tespit karesi alınarak o tespit karesi üzerinde yanlış pozitif, yanlış negatif, doğru pozitif ve doğru negatif değerleri hesaplanmıştır. (Burada her bir videodaki aynı numaralı tespit karesi alınarak işlemler gerçekleştirilmiştir)

Bu paragrafta yanlış pozitif, yanlış negatif, doğru pozitif ve doğru negatif değerlerinin bizim değerlendirmelerimizde ne anlamlara geldikleri açıklanmıştır. Yaya tespit sistemimizin çalıştırılması sonucu resim karesinde elde edilen her bir pikseldeki konumda; elde edilen yanlış pozitif değeri, sistemimizin bize o pikseldeki konumda yaya'nın var olduğunu bize söylemesi fakat o pikseldeki konumda aslında yaya objesinin bulunmaması; elde edilen yanlış negatif değeri, sistemimizin bize o konumda yaya'nın var olmadığını bize söylemesi fakat o pikseldeki konumda aslında yaya objesinin bulunması; elde edilen doğru pozitif değeri, sistemimizin bize o konumda yaya'nın var olduğunu bize söylemesi ve o pikseldeki konumda zaten yaya objesinin bulunması; elde edilen doğru negatif değeri, sistemimizin bize o konumda yaya'nın var olmadığını bize söylemesi ve o pikseldeki konumda zaten yaya objesinin bulunmaması anlamına gelmektedir.

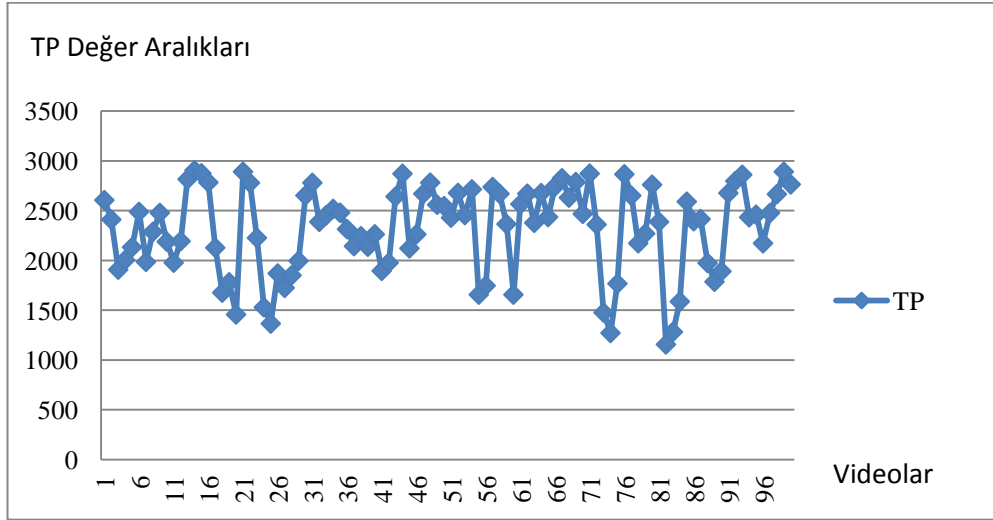
Sonraki sayfalarda gösterilen grafiklerde; grafik 5.1'de yanlış pozitif, yanlış negatif değerleri; grafik 5.2'de doğru pozitif değerleri; grafik 5.3'de doğru negatif değerleri grafiksel olarak gösterilmiştir. Buradaki grafiklerde x eksenleri her bir videodaki işlenen resim karesinden elde edilen FP, FN, TP, TN değer aralıklarını göstermekte iken y eksenleri ise veri setimizdeki ayrı ayrı videolardan yaya tespit sistemi uygulanarak işlenen resim karelerini göstermektedir. (Her bir video için o videodan elde edilen işlenmiş bir tane tespit karesi işleme alınmıştır.)



**Grafik 5.1 Çalışılan Algoritma İçin Her Bir Videodan Alınan İşlenmiş Resim Karelerindeki FP, FN Değer Karşılaştırması Grafiği**



**Grafik 5.2 Çalışılan Algoritma İçin Her Bir Videodan Alınan İşlenmiş Resim Karelerindeki TN Değer Karşılaştırması Grafiği**



**Grafik 5.3 Çalışılan Algoritma İçin Her Bir Videodan Alınan İşlenmiş Resim Karelerindeki TP Değer Karşılaştırması Grafiği**

Karşılaştırma işlemleri için ikinci adım olarak sistem başarılarımızı değerlendirmek için recall, specificity, precision, accuracy, f-measure değerleri hesaplanmıştır. Bu adım için ilk olarak karşılaştırma işlemlerinin ilk adımında (yanlış pozitif, yanlış negatif,

dođru pozitif ve dođru negatif deđerlerinin hesaplanması işlemi) bahsedilen yaya tespit sistem aşamaları gerçekleştirilmiştir. Ardından karşılaştırma işlemlerinin ilk adımında hesaplanan yanlış pozitif, yanlış negatif, dođru pozitif ve dođru negatif deđerleri ve aşağıdaki 5.1, 5.2, 5.3, 5.4, 5.5 numaralı denklemler kullanılarak recall, specifity, precision, accuracy, f-measure deđerleri hesaplanmıştır. Buradaki kavramların anlamları ise ilk olarak accuracy deđeri bize dođruluk oranını vermektedir ve dođru sınıflandırılmış örnek sayısının, toplam örnek sayısına oranı ile bulunur. İkinci deđerimiz olan recall deđeri ise bize sistemimizin duyarlılık oranını göstermektedir ve recall deđeri dođru sınıflandırılmış pozitif örnek sayısının toplam pozitif örnek sayısına oranı ile bulunur. Üçüncü olarak, specifity deđeri dođru tanımlanmış negatif deđerlerin oranını göstermektedir ve specify deđeri dođru sınıflandırılmış negatif örnek sayısının toplam negatif örnek sayısına oranı ile bulunur. Ardından, precision deđeri ise bize sistemimizin kesinlik oranını göstermektedir ve precision deđeri, sınıfı 1 olarak tahminlenmiş dođru pozitif örnek sayısının, sınıfı 1 olarak tahminlenmiş tüm örnek sayısına oranı ile bulunur. Son olarak ise f-measure deđeri yani türkçe anlamıyla f-ölçütü, precision ve recall deđerlerinin birleşiminden oluşan bir ölçüttür ve precision ve recall deđerlerinin harmonik ortalaması ile bulunur.

$$\text{Recall} = \text{TP}/(\text{TP}+\text{FN}) \quad (5.1)$$

$$\text{Specifity} = \text{TN}/(\text{TN}+\text{FP}) \quad (5.2)$$

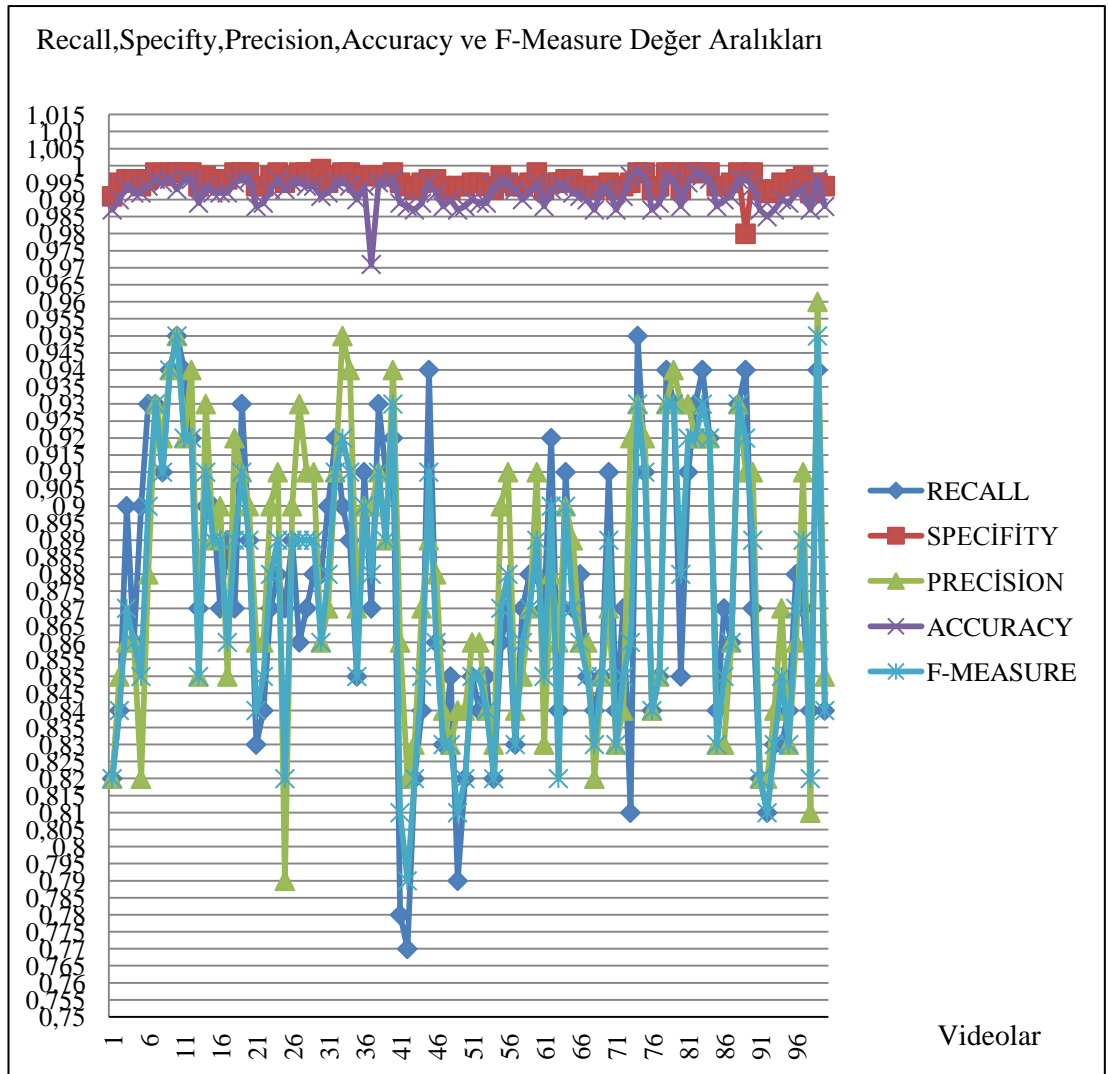
$$\text{Precision} = \text{TP}/(\text{TP}+\text{FP}) \quad (5.3)$$

$$\text{Accuracy} = (\text{TP}+\text{TN}) / (\text{TP}+\text{FP}+\text{FN}+\text{TN}) \quad (5.4)$$

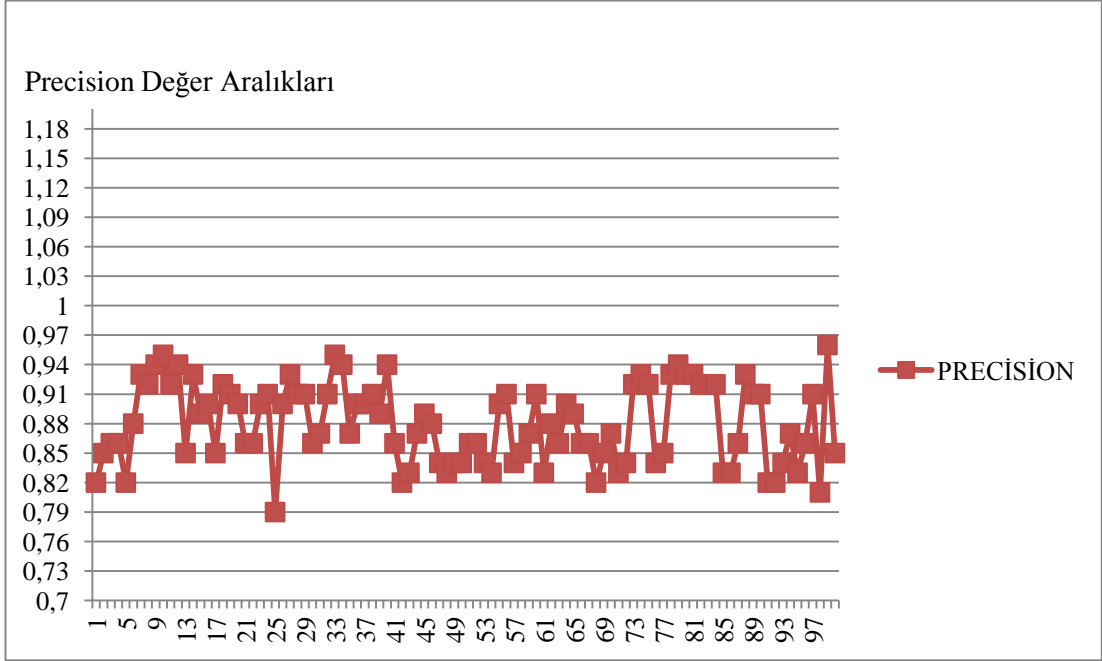
$$\text{F-Measure} = (2 \times \text{Recall} \times \text{Precision}) / (\text{Recall} + \text{Precision}) \quad (5.5)$$



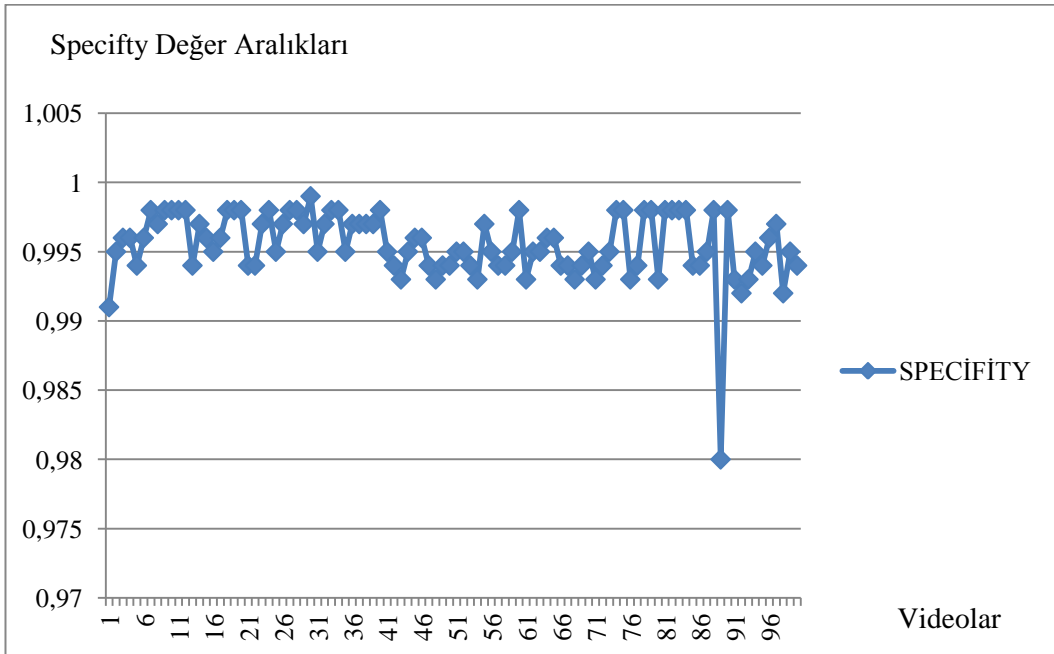
Aşağıda gösterilen grafik 5.4'de recall, specificity, precision, accuracy, f-measure değerleri gösterilmiştir. Ayrıca, grafik 5.4'de gösterilen precision ve specificity grafik değerleri sırasıyla grafik 5.5 ve grafik 5.6'da daha ayrıntılı bir şekilde gösterilmiştir. Buradaki grafiklerde x eksenleri her bir videodaki işlenen resim karesinden elde edilen recall, specificity, precision, accuracy, f-measure değer aralıklarını göstermekte iken y eksenleri ise veri setimizdeki ayrı ayrı videolardan yaya tespit sistemi uygulanarak işlenen resim karelerini göstermektedir.



**Grafik 5.4 Çalışılan Algoritma İçin Her Bir Videodan Alınan İşlenmiş Resim Karelerindeki Recall, Specity, Precision, Accuracy, F-Measure Değer Karşılaştırması Grafiği**

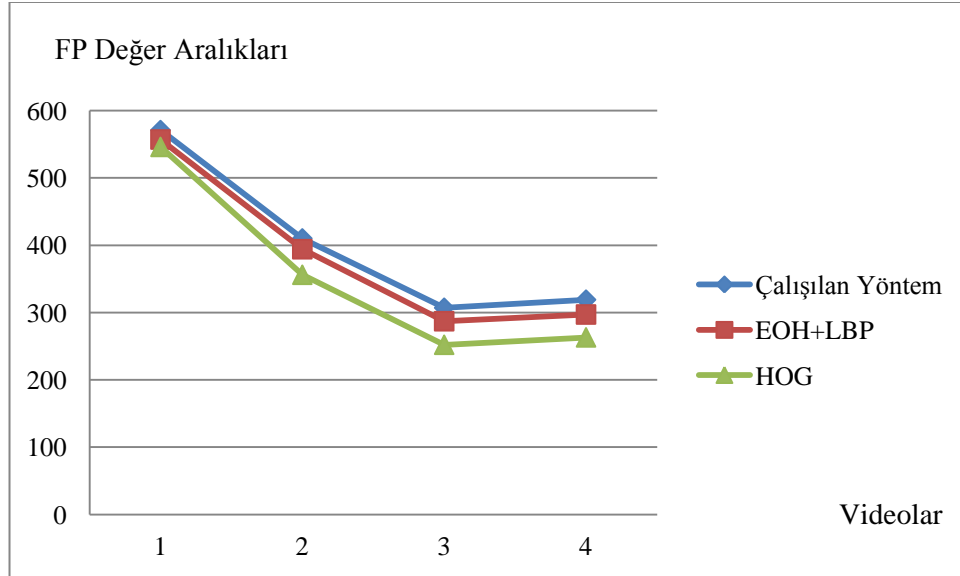


**Grafik 5.5 Çalışılan Algoritma İçin Her Bir Videodan Alınan İşlenmiş Resim Karelerindeki Precision Değer Karşılaştırması Grafiği**



**Grafik 5.6 Çalışılan Algoritma İçin Her Bir Videodan Alınan İşlenmiş Resim Karelerindeki Precision Değer Karşılaştırması Grafiği**

Karşılaştırma işlemlerinin son adımı olarak çalışılan algoritmayla, daha önce yapılmış ve bizim tarafımızdan gerçekleştirilmiş HOG, EOH+LBP yöntemleri karşılaştırılacaktır. Bunun için her bir yöntemin yanlış pozitif değerleri elde edilmiştir ve ardından grafik 5.7 oluşturulmuştur. Bu grafikten çıkarılan sonuç çalışılan yöntem ile diğer iki yöntem arasında benzer sonuçlar alındığıdır. Fakat HOG yönteminin diğer iki yönteme göre daha yüksek tespit oranına sahip olduğu grafikten gözlenmektedir. Burada çalışılan yöntemin, diğer yöntemlere göre tespit yönünden performansız olmasının nedeni çekilen videolardaki çeşitli zorluklara (çeşitli zorluklar kısmına, karşılaşılan problemler (5.3 bölüm) bölümünde detaylıca değinilmiştir.) karşı algoritmamızın veya işleyişimizin yetersiz kalışıdır. Diğer iki yöntem bu tarz çeşitli zor senaryoları karşı daha iyi eğitilmişlerdir ve algoritmaları daha optimize oluşturulmuştur. Fakat çalışılan yöntemde diğer iki yönteme nazaran tespit yönünden daha az performans alınmasına karşın zaman performansı yönünden yüzde on yedi başarı sağlanmıştır. ( Sonuç olarak çalışılan yöntem(geliştirilen) insan tespiti için elverişli bilgi sağlamakta ve yüksek tespit oranına katkıda bulunmaktadır.)



**Grafik 5.7 Çalışılan Yöntem ile HOG ve EOH+LBP Yöntemlerinin FP Değer Karşılaştırması Grafiği**

### 5.3. Karşılaşılan Problemler

Bu alt bölümde sistemimizi geliştirirken karşılaşılan problemlerden bahsedilecektir.

İlk olarak, video'nun aşırı güneşli bir ortamda çekilmesi ve çeşitli objelere (araba camı, araba farları, apartman camları gibi) bu güneş ışınlarının çarpması sonucu oluşan yansımalarından çeşitli hatalar oluşmakta ve bu durumda uyguladığımız yöntemin bazı yerlerinde kayıplar verilebilmektedir, bu gibi sorunlara çözümler getirilmeye çalışılmıştır.

İkinci olarak ise yayaların renklerinin arka planla çok yakın olduğu (arka planla yakın renkte elbise, giysi örnek verilebilir.) konumlarda yayaların bir parçası veya tamamı tespit edilemedi. Fakat morfolojik yöntemlerle bu hatalar çoğunlukla giderilmeye çalışıldı.

Üçüncü olarak, yayaların sonuç olarak oluşan şekli tam olarak bir insan şekli ile örtüşmeyebilmektedir. Dördüncü bölümde gösterilen resimlerden anlaşılacağı üzere yaya tespiti işleminde siyah ile işaretlenmiş yaya objelerinde bazı kayıplar olduğu piksellerin tam insan şekliyle örtüşmeyişinden dolayı gözlenmektedir. Yukarıda bahsedilen tam olarak şeklin örtüşmemesi durumuna hataları yok etme kısmında çeşitli çözümler getirmeyi çalışılmıştır ve uygulanmıştır. (İleriki çalışmalarda, hataları düzeltmenin üzerine gidilmesi iyi sonuçlar verebilir.)

Dördüncü olarak, yayalar videoda buldukları konumdan arkaya doğru gittiklerinde veya belli açılardan algılanamamaktadırlar. Burada bir tercih söz konusudur. Daha çok gürültü olan görüntüler kabul edilmesi durumunda yayaların tespit edilmesine karşı sistemimiz daha duyarlı olacaktır.

Son olarak, yayaların arka arkaya gelmesi veya birbirlerine çok yakın yaklaşmaları sonucunda yayalar tek insan olarak algılanabilmektedirler, bu gibi sorunlara hataları yok etme kısmında çeşitli çözümler getirilmeye çalışılmıştır.

## 6. SONUÇLAR VE ÖNERİLER

Bu bölümde, tezin üzerine çalışılmaya devam edilirse hangi problemlere karşı çözüm bulunması konusunda ve geliştirilmesi gereken kısımlar hakkında öneriler verilmiştir ve genel sonuç özeti yapılmıştır.

### 6.1. Öneriler

Bu tezin üzerine ileride çalışılmaya devam edildiğinde yukarıda bahsedilen problemlere karşı çözüm bulunmaya çalışılacaktır. Bu problemleri çözmek için görüntü işleme teknikleri kullanılabileceği gibi çeşitli başka optimize sonuçlar verecek yöntemlerde uygulanabilir. İleriki aşamalarda geliştirilmesi gereken hususlar ve çözülmesi gereken problemler aşağıda maddelendirilmiştir:

1. Yayaların giyindikleri elbiselerin renkleri arka planla çok yakın olmaması ve bu renklerin tercih edilmemesi gerekir. Bu gibi sorunlardan doğan problemler çeşitli görüntü işleme teknikleri ile düzeltilebilir.
2. Video'nun aşırı güneşli bir ortamda çekilmemesine gayret gösterilmeli veya bu gibi ortamlarda çekilecekse eğer yansımaları engelleyecek çeşitli teknolojik aletlerin kullanılması gerekir.
3. Fazla insan sayısına dikkat edilmeli, optimize bir sonuç alabilmek için az insan sayısı ile çalışılmalıdır.
4. Daha iyi ve optimize bir tespit ve yaya işaretlenmesi için daha iyi matematiksel algoritmalar geliştirilebilir.

## 6.2. Sonuç

Oluşan kazaların önlenmesi için bu yöntem kullanılabilir ve ayrıca bu yöntemle kazalarla ilgili yapılan araştırmalara sayısal veri sağlanabilir. Bu yöntem çeşitli sistemlerle birlikte uygulandığında (örneğin mobese (mobil elektronik sistem entegrasyonu) kameralarını yazılım olarak yüklenildiğinde veya maçlarda ofsayt tespit sistemlerinde kullanılabilir) oluşan kazaların en aza indirgenmesi sağlanabilir, çeşitli rapor analizlerinde efektif sonuçlar görülebilir ve ayrıca birçok çeşitli sorunlara da çözüm getirilebilir.

Yapılan çalışmaların kısaca özeti; gri tonlarda arka planın hesaplanması, ardından videodaki her bir kareden arka planın çıkarılması ve elde edilen sonuçları normalize edip kmeans algoritmasına göre ikili veriye çevirme, akabinde bulunan bu karelerin içerisinde belli alandan küçük alana sahip alanları ( fazlalıkları elemek anlamında ) eleyerek, kalanların iç kısmındaki yani ortasındaki boşluklarında morfolojik işlemler kullanılarak doldurulmasıyla yayaların videodaki pozisyonlarının mümkün olduğunca az hatalı bir şekilde tespit edildi ve bulunan bu konumlar, okunan videonun üzerine işlendi. Son olarak ise daha önce yapılan farklı benzer çalışmaların deneysel olarak analizi ve bizim oluşturduğumuz algoritmalarla performans karşılaştırılması işlemi yapılmıştır.

Sonuç olarak, videodaki yayalar tespit edildi, işaretlendi ve yayaların işaretlendiği videolar oluşturuldu; ayrıca bizim algoritmamızla, daha önce yapılan farklı benzer çalışmaların gerçekleştirimi ve karşılaştırılma işlemleri yapılmıştır. Başarılı bir şekilde video görüntülerinden yaya tespiti ve karşılaştırma işlemlerimiz gerçekleştirilmiştir.

## KAYNAKLAR

[1] Matlab' in Programının Ana Sitesi:

<http://www.mathworks.com/matlabcentral/>

[2] Bill Triggs Yayınları:

<http://lear.inrialpes.fr/people/triggs/pubs/>

[3] Chalmers Publication Library Web Sitesi:

<http://publications.lib.chalmers.se/cpl/>

[4] Matlab Yardım Masası ( Matlab Help Desk )

[5] N. Dalal , B. Triggs : "Histograms of Oriented Gradients for Human Detection.", In: IEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR, 2005

[6] Y. Ma, X. Chen, L. Jin, G. Chen : "A Monocular Human Detection System Based on EOH and Oriented LBP", Advances In Visual Computing Lecture Notes in Computer Science, 2011

[7] P. Sabzmeydani, G. Mori: "Detecting Pedestrians by Learning Shapelet Features.", In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2007

[8] C. Papageorgiou and T. Poggio : "A Trainable Pedestrian Detection system", International Journal of Computer Vision (IJCV), 2000

[9] A. Howard, L. H. Matthies, A. Huertas, M. Bajracharya , A. Rankin : "Detecting Pedestrians with Stereo Vision: Safe Operation of Autonomous Ground Vehicles in Dynamic Environments", 2007

[10] L. Spinello, A. Macho, R. Triebel, R. Siegwart.; "Detecting Pedestrians at Very Small Scales", Autonomous Syst. Lab., ETH Zurich, Zurich, Switzerland , 2009

- [11] P. Viola, M. J. Jones, and D. Snow.; "Detecting Pedestrians Using Patterns of Motion and Appearance. ", Int. Journ. of Comp. Vis., 63(2):153–161, 2005.
- [12] L. Wang, J. Shi, G. Song, I. Shen : "Object Detection Combining Recognition and Segmentation", Eighth Asian Conference on Computer Vision (ACCV), 2007
- [13] Ege Üniversitesi Bilgisayar Grafiği ve Yapay Us Grubu Web Sitesi:  
<http://yzgrafik.ege.edu.tr/>
- [14] R. C. Gonzalez, R. E. Woods, Digital Image Processing (3rd Edition), Prentice Hall, 2008
- [15] Wikipedia Uluslararası Özgür Ansiklopedi Web Sitesi:  
[http:// http://en.wikipedia.org](http://en.wikipedia.org)
- [16] CVC(Computer Vision Center) Web Sitesi:  
<http://www.cvc.uab.cat>
- [17] Photosig Web Sitesi:  
<http://www.photosig.com>
- [18] I. T. Young, J. J. Gerbrands, L. J. van Vliet : " Fundamental of Image Processing",1998
- [19] Roborealm Web Sitesi:  
<http://www.roborealm.com>
- [20] P. Dollar, C. Wojek, B. Schiele, P. Perona: " Pedestrian Detection: A Benchmark", CVPR, 2009
- [21] B. Leibe, E. Seemann, B. Schiele : " Pedestrian Detection in Crowded Scenes ", CVPR, 2005



- [22] A. Shashua, Y. Gdalyahu, G. Hayun : " Pedestrian Detection for Driving Assistance Systems: Single-frame Classification and System Level Performance", Intelligent Vehicle Symposium, IEEE, 2004
- [23] B. Leibe, E. Seemann, and B. Schiele: " Learning to detect natural image boundaries using local brightness, color, and texture cues ", Pattern Analysis and Machine Intelligence, IEEE, 2004,
- [24] J. L. Barron, D. J. Fleet, S. S. Beauchemin, and T. A. Burkitt : " Performance of optical flow techniques ", IJCV, 1994
- [25] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. Black, R. Szeliski.: " A database and eval methodology for optical flow", In ICCV, 2007
- [26] D. Scharstein, R. Szeliski: " A taxonomy and evaluation of dense two-frame stereo correspondence algorithms", IJCV, 2002

## ÖZGEÇMİŞ

### Kişisel Bilgiler

Soyadı, adı : YILMAZ, Abdullah Asım  
Uyruğu : T.C.  
Doğum tarihi ve yeri : 20.05.1988 Ankara  
Medeni hali : Bekar  
Telefon : 0 (312) 287 58 33/1330  
Faks : 0 (312) 287 54 58  
e-mail : aayilmaz@etu.edu.tr

### Eğitim

Derece	Eğitim Birimi	Mezuniyet tarihi
Lisans	Başkent Üniversitesi/Bilgisayar	2010

### İş Deneyimi

Yıl	Yer	Görev
2011- ...	T.C Gıda, Tarım ve Hayvancılık Bakanlığı	Bilgisayar Mühendisi

### Yabancı Dil

İngilizce