T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

VİDEO İŞARETLERİ İÇİNDEKİ İNSANLARIN SAYISININ TESPİTİ

ÇAĞLAR ÇAVDAR

YÜKSEK LİSANS TEZİ
ELEKTRONİK VE HABERLEŞME MÜHENDİSLİĞİ ANABİLİM DALI
HABERLEŞME PROGRAMI

DANIŞMAN
DR. ÖĞR. ÜYESİ BÜLENT BOLAT

İSTANBUL, 2018

T.C.

YILDIZ TEKNİK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

VİDEO İŞARETLERİ İÇERSİNDEKİ İNSANLARIN SAYISININ TESPİTİ

Çağlar ÇAVDAR tarafından hazırlanan tez çalışması 02.11.2018 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Elektronik ve Haberleşme Mühendisliği Anabilim Dalı'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Tez Danışmanı

Dr. Öğr. Üyesi Bülent BOLAT

Yıldız Teknik Üniversitesi

Jüri Üyeleri

Dr. Öğr. Üyesi Bülent Bolat

Yıldız Teknik Üniversitesi

Dr. Öğr. Üyesi Umut Engin AYTEN

Yıldız Teknik Üniversitesi

Dr. Öğr. Üyesi Gökalp TULUM

İstanbul Arel Üniversitesi

ÖNSÖZ

Bu tez çalışması süresince tez danışmanım Dr. Öğr. Üyesi Bülent Bolat hocama, arkadaşım Abdullah Özek'e ve aileme teşekkür ederim.

Kasım, 2018

Çağlar ÇAVDAR

İÇİNDEKİLER

Sayfa

SİMGE LİSTESİ vi

KISALTMA LİSTESİ vii

ŞEKİL LİSTESİ viii

ÇİZELGE LİSTESİ ix

ÖZET x

ABSTRACT xii

BÖLÜM 1

GİRİŞ 1

1.1 Literatür Özeti 1
1.2 Tezin Amacı 3
1.3 Orijinal Katkı 4

BÖLÜM 2

DURAĞAN GÖRÜNTÜLERDE YAYA TESPİTİ 6

2.1 Özniteliklerin Çıkarılması 8
2.1.1 Gamma Doğrulama ve Renk Normalizasyonu 11
2.1.2 Gradyan Hesaplama 12
2.1.3 Histogram Hesaplama 13
2.1.4 Blok Normalleştirme 15
2.2 Sınıflandırıcı Algoritmaları 17
2.2.1 SVM Sınıflandırıcı 17
2.2.2 K en Yakın Komşu Sınıflandırıcı 20
2.2.3 Adaboost Sınıflandırıcı 21
2.3 Geliştirilen Sınıflandırma Algoritması 24

BÖLÜM 3

VİDEO İŞARETLERİNDE İNSAN TESPİTİ 26

3.1 Yaya Tespiti Uygulamaları 27
3.2 Yaya Tespitinde Karşılaşın Zorluklar 29
3.3 Hareket Bölütleme 30
3.3.1 Morfolojik Aşındırma Operasyonu 33
3.3.2 Morfolojik Yayma Operasyonu 34
3.3.3 Morfolojik Açma Operasyonu 35
3.4 Yaya Tespiti 36
3.4.1 Ön Plan İmge Bölgeleri Yöntemi 37
3.4.2 Düşük Seviye Özniteliklerin Çıkarılması Yöntemi 38
3.4.3 Birleştirilmiş Kanal Öznitelikleri Yöntemi 42

BÖLÜM 4

SONUÇ VE ÖNERİLER 45

4.1 Önerilen Yaya Tespit Sistemi ve Uygulaması 45
4.2 Sonuçlar 49

KAYNAKLAR 50

ÖZGEÇMİŞ 55

SİMGE LİSTESİ

$g_x(x,y)$	Gradyan yatay bileşeni
$g_y(x,y)$	Gradyan dikey bileşeni
s	Genlik
θ	Yön
ε	Ağırlık hata oranı
h(.)	Sınıflandırıcı fonksiyonu
e	Sınıflama hata miktarı
w	Örnek ağırlık vektörü
t	İterasyon sayısı
j	Sınıflandırıcı indeksi
i	Örnekleme indeksi
f_{max}	Normalize edilmiş vektörün maximum değeri
f	Normalize edilmiş vektör
v	Normalize edilmemiş vektör
$\in$	Hata miktarı
C	Yeni kanal görüntüsü
Ω	Düşük seviye kayan değişmez fonksiyonu
I	Kanaldan alınan görüntü

KISALTMA LİSTESİ

ACF	Aggregate Channel Features
ALOI	Amsterdam Library of Object Images
HOG	Histogram of Oriented Gradients
ICF	Integral Channel Features
INRIA	Institut National de Recherche en Informatique et en Automatique
KNN	K-Nearest Neighbors
RFID	Radio Frequency Identification
SIFT	Scale Invariant Feature Transform
SVM	Support Vector Machine

ŞEKİL LİSTESİ

Sayfa

Şekil 2.1 Penn-Fudan yaya veri setinden alınmış bir test görüntüsü 7
Şekil 2.2 INRIA pozitif ve negatif örnekler 7
Şekil 2.3 R-HOG, C-HOG ve tek merkezli C-HOG tanımlayıcıları 9
Şekil 2.4 HOG algoritması aşamaları 10
Şekil 2.5 Renk normalizyonu yapılmış görüntü 11
Şekil 2.6 Gamma doğrulama uygulanmış görüntü 12
Şekil 2.7 Gradyan genlik ve açı bilgileri 14
Şekil 2.8 Gradyan yön vektörü 14
Şekil 2.9 HOG tanımlayıcısı için kullanılan hücre ve blok yapısı 15
Şekil 2.10 Yön vektörleri çıkarılmış görüntü 16
Şekil 2.11 HOG ile öznitelik çıkarma ve SVM ile yaya tespiti algoritması 17
Şekil 2.12 SVM genel ağ yapısı 18
Şekil 2.13 Veri setinin SVM ile iki farklı kümeye ayrılması 19
Şekil 2.14 Veri setine gelen yeni üyenin KNN ile sınıflandırılması 21
Şekil 2.15 Veri setinin Adaboost ile sınıflandırılması 24
Şekil 2.16 Çoklu sınıflandırmalı yaya tespiti algoritması 25
Şekil 3.1 Bir disk yapılandırma elemanı ile ikili bir görüntünün aşındırılması 33
Şekil 3.2 3x3 kare yapı elemanı ile ikili görüntünün aşındırılması 33
Şekil 3.3 İkili görüntünün yayılması 34
Şekil 3.4 3x3 kare yapı elemanı ile ikili bir görüntünün yayılması 34
Şekil 3.5 İkili görüntünün açılması 35
Şekil 3.6 3x3 kare yapı elemanı ile ikili bir görüntünün açılması 36
Şekil 3.7 İmge bölgeleri analizinden sonra ön plan maske çıkarımı 38
Şekil 3.8 Hiyerarşik öznitelikler alt pencereler 40
Şekil 3.9 HOG Tanımlayıcısı için kullanılan hücre ve blok yapısı 41
Şekil 3.10 ACF dedektörünün süreçlerin görselleştirilmesi 42
Şekil 3.11 ACF dedektörü algoritması 43
Şekil 3.12 Video görüntüsü üzerinde Matlab ile yaya tespiti 44
Şekil 4.1 SVM, Adaboost, KNN sınıflandırıcılarla elde edilen sonuçlar 46
Şekil 4.2 Önerilen yöntem ile yaya tespiti 46

ÇİZELGE LİSTESİ

Sayfa

Çizelge 2.1 Yön vektörlerinin hesabı için türev maskesi 13
Çizelge 2.2 Adaboost algoritmasının genel akışı 23
Çizelge 4.1 SVM, Adaboost, KNN ve önerilen yöntemin sonuçları 48
Çizelge 4.2 SVM, Adaboost, KNN ve önerilen yöntemin başarımları..... 48
Çizelge 4.3 ALOI veri setinde yaya tanıma oranları ……48

ÖZET

VİDEO İŞARETLERİ İÇİNDEKİ İNSANLARIN SAYISININ TESPİTİ

Çağlar ÇAVDAR

Elektronik ve Haberleşme Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

Tez Danışmanı: Dr. Öğr. Üyesi Bülent BOLAT

Son zamanlarda gelişen teknoloji ile birlikte bilgisayar tabanlı sistemlerin kullanımı gün geçtikçe yaygınlaşmıştır. Nesne algılama, bilgisayar ile görü ve makine öğrenmesinde popüler bir konudur. İnsanların bilgisayarlar ile olan etkileşiminin artması ve bilgisayar tabanlı sistemlerinin insan hayatına etkisi düşünüldükçe, yaya tespit yöntemleri her geçen gün daha da önem kazanmaktadır. Tasarlanan insan sayma sistemleri ise bir kapıdan veya koridordan geçen yayaların sayısını belirlemek için kullanılan cihazlardır. Çoğu zaman bu sistemler, büyük bir binanın girişinde, tren istasyonları, hava alanları veya toplu taşıma araçlarının duraklarında kullanılmaktadır. Böylece izlenilen alan içerisindeki toplam yaya sayısı kaydedilebilir. İnsan sayma sistemleri, pazarlama araştırması anketi yapabilmek ve belirli bir zaman periyodundaki insan yoğunluğunu tespit edebilmek için oldukça önemli sistemlerdir.

Bu tez çalışmasında, farklı sınıflandırıcılar kullanarak video işaretlerinde yaya tespiti yapan bir algoritma sunulmaktadır. Çalışma sırasında görüntünün özniteliklerini çıkarmak için tanımlayıcı olarak yaya tespitinde sıkça kullanılan HOG, sınıflandırıcı olarak ise destek vektör makineleri (SVM), K en yakın komşu (KNN) ve AdaBoost algoritmalarının birleşimiyle çoğunluk oylamasına dayanan özel bir sınıflandırıcı kullanılmıştır. Üç farklı sınıflandırıcı ile birlikte video görüntüsünde hareket kestirimi yapılarak yaya olup olmadığına karar verilmektedir.

Uygulanan yöntemin etkinliğini göstermek için sistem INRIA veri seti kullanılarak eğitilmiştir ve farklı bir veri seti olan Penn Fudan ile test edilmiştir. Önerilen yöntemin doğru pozitif oranını yükseltirken hatalı pozitif oranını düşürdüğü açıkça görülmektedir.

Buradan yola çıkarak çoğunluk oylaması yönteminin sunulan problemin çözümü için daha olumlu sonuçlar ürettiği söylenebilir. Deneysel sonuçlar önerilen yöntemin tek bir sınıflandırıcı ile yapılan yaya tespitine göre daha yüksek doğruluk oranında çalıştığını desteklemektedir.

Anahtar Kelimeler: Yaya tespiti, HOG, destek vektör makineleri, k en yakın komşu algoritması, Adaboost sınıflandırıcı, hareket kestirimi

ABSTRACT

PEOPLE COUNTING ON VIDEO STREAMS

Çağlar ÇAVDAR

Department of Electronics and Communications Engineering

MSc. Thesis

Advisor: Assoc. Prof. Dr. Bülent BOLAT

Recently, the use of computer systems has become widespread with the developing technology. Object detection is a popular topic in computer vision and machine learning. The interaction of people with computers has increased and computer based systems have affected the human life so pedestrian detection methods are gaining more importance every day. Designed human counting systems are devices used to determine the number of pedestrian that pass through a door or corridor. These systems are usually used at the entrance of a large building, at railway stations, at airports, or at stops of public transport. Thus, the total number of people in the monitored area can be recorded. Human counting systems are very important systems for conducting marketing research surveys and for determining the human density in a given time period.

In this thesis, an algorithm which detects pedestrians in video signs using different classifiers is presented. A special classifier based on the majority vote with the combination of support vector machines (SVM), K nearest neighbors (KNN) and AdaBoost algorithms has been used as a descriptor HOG to extract features of image during the this study. Motion estimation is implemented on the video image with three different classifiers to decide whether or not it is pedestrian.

To demonstrate the effectiveness of the method, the system was trained with the INRIA data set and tested with Penn Fudan data set. It is clear that the proposed method decreases the false positive rate while increasing the true positive rate. Thus,

it can be said that the majority voting method produces more positive results for the solution of the problem presented. The experimental results suggest that the method works with higher accuracy than pedestrian detection with a single classifier.

Keywords: Pedestiran detection, HOG, support vector machine, K-nearest neighbors algorithm, AdaBoost, motion detection

BÖLÜM 1

GİRİŞ

1.1 Literatür Özeti

Literatürde, yaya tespiti için birçok yaklaşım ve çalışma bulunmaktadır. Bu bölümde, yaya tespiti üzerine yapılan çalışmalar kısa özetler şeklinde anlatılacaktır. Bir görüntünde yaya tespiti çalışmaları 1985 yıllında ilk defa Tsukiyama ve Shirai tarafından yapılmıştır. Tsukiyama ve Shirai, ardışık iki TV görüntüsünde uzun aralıklarla bir geçitten yürüyen kişilerin tespit edilmesi için bir yöntem geliştirmişlerdir. Yöntem sırasıyla şu adımlardan oluşmaktadır. Görüntü işleme kullanılarak TV görüntülerinden insan olmaya aday olacak alanların çıkarılması, bu alanlarda yer alan kişilerin şekil ve boyutları temel alınarak belirlenmesi ve bir geçide yerleştirilmesi, son olarak ise iki görüntü arasındaki kişilerin konumlarına göre benzerliğinin bulunmasıdır [1].

Oren vd. yayaları tespit etmek için dağınık sahnelerde eğitilebilir nesne algılama mimarisini sunmaktadır. Önceki yaklaşımlardan farklı olarak, herhangi bir el yapımı modele veya harekete dayanmaz ve bu algoritma örneklerden öğrenir. Yayaları tespit etmek için dalgacık tabanlı bir dalgacık dönüşümü tekniği önermektedir [2]. Gavrila ve Philomin, mesafe dönüşümlerine dayanan şekil tabanlı bir nesne algılama yöntemi sunmaktadır ve bu sistemi gerçek zamanlı olarak araçların üzerine yerleştirmeyi hedeflemektedir. Bu yöntem, nesne görüntüleri üzerinde uzaklık dönüşümüyle bir sıradüzeni oluşturulmakta ve bu sıradüzeni görüntü üzerinde eşlemek amacıyla kullanmaktadır. Bölgede aranılan nesnenin olup olmadığına elde edilen uyuşma derecesine göre karar verilmektedir [3].

Papageorgiou ve Poggio tarafından geliştirilen yöntemde nesne tespiti üzerine eğitilebilir bir sistem tasarlamışlardır. Bu sistem ile daha çok yaya tespiti üzerine odaklanmışlardır. Görüntünün özniteliklerini çıkarmak için Haar dalgacık dönüşümü kullanılarak çıkarılan öznitelikler destek vektör makinaları (SVM) ile sınıflandırılıp sistem eğitilmektedir [4]. Ioffe ve Forsyth çalışmasında adayların beden kısımlarını nasıl bulacaklarını kinematik özelliklerden yola çıkarak bir kişinin görünüşü üzerindeki kısıtlamalarla tutarlı parçaların inşa edilmesiyle sağlamışlardır. Bir kişinin makul şekilde modellenmesi için en azından dokuz parçaya ihtiyaç duyduğundan, bu parçalama sonucu meydana gelen karmaşıklık nedeniyle her grubun incelenmesi mümkün değildir. Bu sorun için önerdikleri yaklaşım arama yapılırken kişilere karşılık gelen grupları kabul eden bir sınıflandırıcının öngörülen sürümleri kullanılarak kestirim yapabilmesidir [5]. Viola ve Jones görüntüdeki piksellerin parlaklık ve değişim bilgisinden yararlanarak hızlı bir nesne tespit yöntemi geliştirmişlerdir. Bu yöntemi yüz tespiti yapmak için uyguladıkları dönemde meşhur olmuşlar ve yöntem artık kendi adlarıyla anılmaya başlanmıştır [6]. Bir diğer çalışmalarında ise bir video görüntüsünün iki ardışık çerçevesini hesaba katan Haar ile çıkarılan şekil ve harekete dayalı öznitelikleri Adaboost algoritması ile sınıflandırmışlardır [7]. Bu algoritmada eğitim aşaması yavaş, ancak nesne tespiti oldukça hızlı bir şekilde yapılmaktadır.

Dalal ve Triggs, kenar yönlü sıklık grafiklerine [8] ve ölçeğe göre değişmeyen öznitelik dönüşümlerine göre ölçeklenebilir değişmez özellik dönüşümüne (SIFT) [9] benzeyen bu özniteliklerin yaya tespitinde kullanımını sunmuşlardır. Durağan görüntülerde HOG (Histogram Oriented Gradients) tanımlayıcı ve SVM sınıflandırıcı ile görüntü üzerindeki yayaları tespit etmişleridir [10]. Bu yöntem yaya tespitinde Papageorgiou [4] tarafından önerilen Haar dalgacık modelleme yöntemiyle yapılan yaya tespitine göre daha iyi sonuçlar üretmektedir. 3x3 hücrelerden oluşan blok, 6x6 pikselden oluşan hücre ve 9 ögeden oluşan yön vektörü histogramı kullanarak HOG tanımlayıcı ve sınıflandırma aşamasında ise sadece SVM sınıflandırıcı kullanarak yaya tespiti yapmışlardır. Bu yöntem en çok referans edilen yaya tespit sistemlerindendir. Temel olarak yönlü vektör eğilimlerin sıklık grafiğine dayanmaktadır.

Sabzmeydani ve Mori, pozitif ve negatif veri setlerinde bulunan görüntüler arasındaki yerel şekil farklarından yararlanmayı önermektedir. Şekilsel özellikleri öğrenme ve orta

seviyeli özellikleri çıkarma üzerine çalışmışlardır. Bu özellikleri kullanarak görüntünün yerel bir bölgesine odaklanarak düşük seviyeli gradyan bilgilerini oluşturmuşlardır. Adaboost sınıflandırıcı ile çıkarılan öznitelikleri sınıflandırmışlardır ve tüm bu özellikleri kullanarak görüntünde yaya olup olmadığına karar vermişlerdir [11].

Dollar vd. 2014 yılında yüksek çözünürlüklü görüntü özelliklerini açık bir şekilde hesaplanmak yerine, yakındaki ölçeklerden çıkarım yaparak yaklaşık sonuçlar üretilebileceğini savunmuşlardır. Bu temel öngörü, en son teknolojiye göre doğru ve çok daha hızlı olan nesne algılama algoritmalarını tasarlamalarını sağlamıştır. Bu modern yaya detektöründe iyi örneklenmiş bir görüntü piramidinin her ölçeğinde öznitelikler hesaplanmaktadır. Önerilen ACF (Aggregate Channel Features) oldukça yeni bir detektördür. Yeni önerilen bir yöntem olmasına karşılık yapılan çalışmalarda sıklıkla kullanılmaya başlanmıştır. ACF detektörü tasarlanırken ICF (Integral Channel Features) detektöründen esinlenilmiştir ve bu sebeple kavramsal olarak çoğu açıdan benzerlik göstermektedir [12].

Cai vd. 2016 yılında geliştirilmiş HOG özelliklerine dayalı bir görüntü sınıflandırıcı algoritması önermişlerdir. Bu çalışmada nesne özelliklerinin yerel özniteliklerini tanımlamak için HOG kullanmışlardır ve aynı zamanda algoritmayı güçlendirmek için SVM ve K en yakın komşu (KNN) sınıflandırıcılarını birleştirerek görüntüler sınıflandırmışlardır. Deneysel sonuçlar bu algoritmanın hatırı sayılır bir şekilde iyileştirme yaptığını göstermiştir [13].

1.2 Tezin Amacı

Günümüzde gelişen teknoloji ile birlikte akıllı sistemlerin kullanımı yaygınlaşmıştır. Bilgisayarların nesneleri tanımaları bilgisayar ile görüntü işleme çalışmalarında önemli bir yer tutmaktadır. İnsanların bilgisayarlar ile olan etkileşiminin artması ve bilgisayar tabanlı sistemlerinin insan hayatına etkisi düşünüldükçe, yaya tespit yöntemleri daha da önem kazanmaktadır. Yaya tespiti, literatürde nesne tanıma kategorisinde yer almaktadır.

Kalabalık insan topluluklarına ulaştırılan hizmetlerin belirli bir ölçütün altına düşürülmeden insanlara aktırılması, belli zaman dilimlerinde meydana gelen

yoğunluktan dolayı gün geçtikçe zorlaşmaktadır. Kalabalık yönetiminin önemli bir parçası olan yaya tespit yöntemleri daha da önem kazanmaktadır. Bütün bu problemler göz önüne alınarak yaya tespitinde günümüze kadar birçok çalışma yapılmıştır.

Bu tez çalışmasında, farklı sınıflandırıcın sonuçlarını kullanarak video işaretlerinde yaya tespiti yapan bir sistem önerilmektedir. Çalışma sırasında görüntünün özniteliklerini çıkarmak için tanımlayıcı olarak HOG, sınıflandırıcı olarak ise SVM, KNN ve AdaBoost algoritmalarının çoğunluk oylamasına dayanan özel bir sınıflandırıcı kullanılmıştır. Üç farklı sınıflandırıcı ile birlikte video görüntüsünde hareket kestirimi yapılarak yaya olup olmadığı belirlenmektedir. Uygulanan yöntemin başarısını göstermek için sistem INRIA veri seti kullanılarak eğitilmiştir ve farklı bir veri seti olan Penn Fudan ile test edilmiştir. Deneysel sonuçlar bize gösteriyor ki önerilen yöntemin tek bir sınıflandırıcı kullanılarak yapılan yaya tespitine göre çoğunluk oylamasına dayalı yaya tespit sisteminin daha yüksek doğruluk oranında sonuçlar ürettiğini söylemektedir.

1.3 Orijinal Katkı

Otomatik olarak yapılan yaya tespiti, sayma ve izleme, hatırı sayılır bir sayıda bilgisayar ile görüntü işleme topluluğunun ilgisini çekmiştir. Literatürde şimdiye kadar yapılan yaya algılama tekniklerinin çoğu, az sayıda insanın olduğu kontrollü ortamlarda iyi sonuçlar üretmektedir. Bu karmaşık problemi basitleştiren çeşitli varsayımlara oldukça değerlidir. Bununla birlikte bu tekniklerin performansı, yaya görünümleri, sayıları, yönelimleri, hareketleri, tıkanıklıkları ve ışık koşullarının uygun varsayımları ihlal ettiği ortamlarda sunulduğunda bozulma eğilimindedir.

Literatürde yaya tespiti ve takibi üzerinde şüphesiz birçok çalışma yapılmıştır. Bu tez çalışmasında geliştirilen yöntemle tasarlanan yaya tespit sistemlerinin başarı oranı arttırılmaya çalışılmıştır. Geliştirilen bu yöntemlerden ilerleyen bölümlerde daha detaylı olarak bahsedilecektir. Bu çalışmada literatürde var olan yaya tespit sistemlerini de ele alarak komite oylaması temeline dayalı bir yöntem önerilmektedir. Önerilen yöntemde, yaya tespit doğruluk oranını artırmak için 2x2 hücrelerden oluşan bloklar, 8x8 pikselden oluşan hücreler ve 9 öğeden oluşan yön vektörü histogramı kullanan HOG tanımlayıcısı ile SVM, AdaBoost ve KNN sınıflandırıcıları birlikte kullanılmıştır. Seçilen görüntülerin belirli bir bölgesinde yaya olup olmadığına karar verirken üç

sınıflandırıcıdan en az ikisinin ortak kararı bu bölgede insan var diyorsa ilgili bölgede yaya var olarak kabul edilmektedir. Bu tez çalışmasında, önerilen sınıflandırma yönteminin yüksek doğruluk oranında yaya tespiti yapabildiği açıkça görülmektedir.

BÖLÜM 2

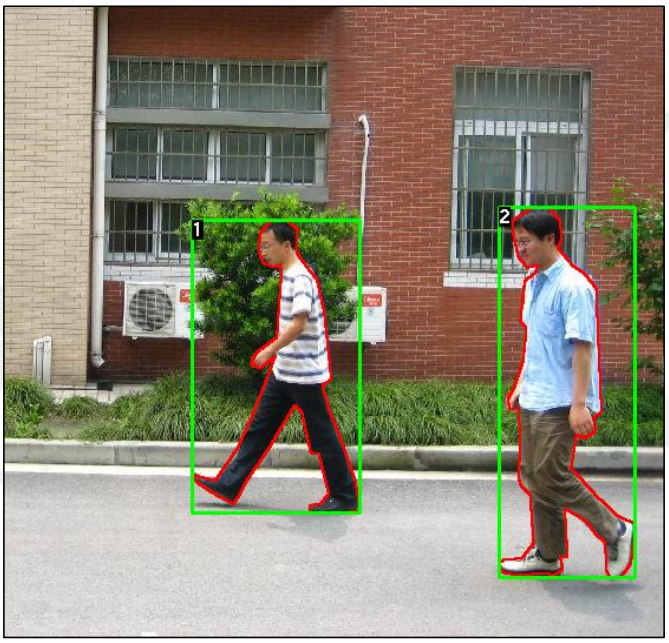
DURAĞAN GÖRÜNTÜLERDE YAYA TESPİTİ

Nesne algılama çok karmaşık ve zor bir problemdir. Bilgisayar ile görüntü çalışmalarında bu sorunlara önerilen çözümlerin çok özel olduğu ve problemlere göre uyarlanmış olması gerektiği iyi bilinmelidir. Herhangi bir görüntüyü her nesne tespit çözümüne uyarlayamayız fakat tüm nesneleri doğru bir şekilde konumlandırması ve sınıflandırması beklenir. Bu algoritmalar, girdi olarak oldukça benzer ve belirli türde görüntüler beklemektedir. Bu tezde, görüntülerde tespit edilmesi gereken yayalar son derece yoğun ve çeşitli biçimlere sahip olmakta, bu da sorunu daha da zor ve çekici hale getirmektedir. Bu nedenle, bu sorun için sağlam bir çözüm elde etme motivasyonu çok önemlidir.

Gerçek zamanlı insan sayısı tahmini, yaya trafiği yönetimi, insan akış yoğunluğu tahmini gibi güvenlik veya insan yönetimi uygulamaları için çok yararlı bir bilgi olabilir. Son zamanlarda görüntü işleme algoritmalarının ve bilgisayarların teknolojisinin ilerlemesi sayesinde insanların izlenmesi ve sayılması için video kameraların kullanımı önemli ölçüde artmaktadır. Bu basit bir problem değildir, günümüzün bilgisayar hızlarıyla bile çözülmesi zor olan bazı durumlar bulunmaktadır. Algoritmanın gerçek zamanlı olarak çalışması gerekir, bu nedenle tespit ve takip için kullanılan yöntemlerin karmaşıklığına sınırlandırmalar getirilebilir. Belki en zor problemlerden birisi de insanların yoğunluğudur. Görüntüdeki görüş alanına giren veya çıkan insanlardan dolayı bu gruptaki tüm insanları ayırt etmek oldukça zordur. Önerilen yöntemler çok doğru ve güvenilir sonuçlar üretebilir ama aynı zamanda çok pahalı olabilmektedir. Bu tez

çalışmasında yüksek başarımda yaya tespiti ve takibi yapabilen basit ve etkili bir yöntem önerilmiştir.

Bu bölümde durağan görüntülerde yaya tespitinin aşamaları ele alınacaktır. Yaya tespiti görüntüde bulunan yayaların çerçeve içerisine alınması işlemidir. Şekil 2.1'de görüldüğü üzere görüntü içeresindeki yayalar çerçeve içerisine alınarak görüntüden ayrılmıştır ve yayalar tespit edilmiştir. Durağan görüntülerde yaya tespitini görüntünün özniteliklerinin çıkarılması ve öznitelikleri çıkarılan görüntülerin sınıflandırılması olmak üzere iki temel kısımda ele alabiliriz.



Şekil 2.1 Penn-Fudan yaya veri setinden alınmış bir test görüntüsü [14]

Pozitif ve negatif veri seti için INRIA'nın açık kaynaklı veri seti kullanılmıştır. Veri seti farklı boyutlarda yaya içeren pozitif ve yaya içermeyen negatif görüntülerden örnekler içermektedir (Şekil 2.2).



Şekil 2.2 INRIA pozitif ve negatif örnekler [19]

2.1 Özniteliklerin Çıkarılması

Görsel bir tanımlayıcı, bir görselin, videonun veya diğer görsel medyaların içeriğinin bir açıklamasıdır. Bu tanımlayıcılar, şekil, renk veya diğer ayırt edici öğeleri karakterize etmek için kullanılmaktadır. Böyle bir görsel tanımlayıcı, Dalal ve Triggs tarafından tanıtılan HOG tanımlayıcısıdır [15]. Siyah beyaz görüntülerde bile HOG'un nesne tespitinde [16] iyi performans gösterdiği görülmüştür. Muhtemelen kısıtlı kaynaklara sahip bir sistemde, daha maliyetli bir RGB görüntüsüne kıyasla sadece giri seviye bir görüntüyü kodlamak için verileri göndermek bir avantajdır.

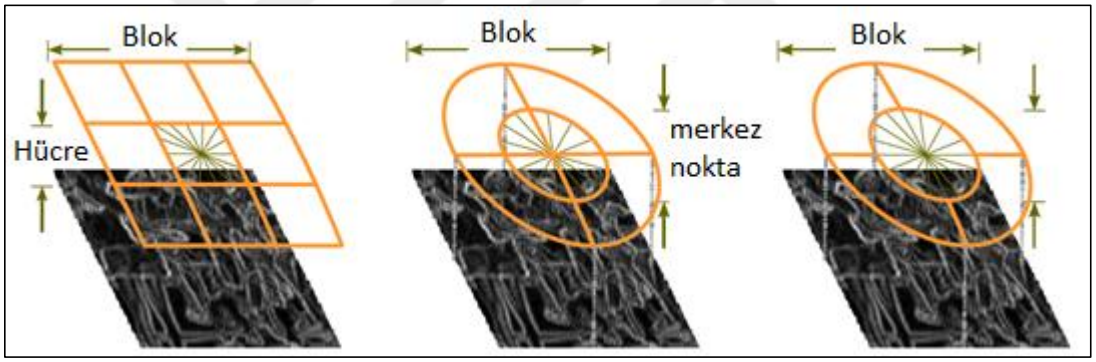
Yönlendirilmiş gradyan tanımlayıcısının histogramının ardındaki temel düşünce, bir görüntü içindeki yerel nesne görünümü ve şeklinin, yoğunluk gradyanlarının veya kenar yönlerinin dağılımı ile açıklanabileceğidir. Görüntü, hücreler olarak adlandırılan küçük bağlantılı bölgelere bölünür ve her hücre içindeki pikseller için, gradyan vektör yönleri histogramı oluşturulur. Tanımlayıcı, bu histogramların birleşimidir. İyileştirilmiş doğruluk için yerel histogramlar, görüntünün daha büyük bir bölgesi boyunca blok olarak adlandırılan yapılarla yoğunluğun bir ölçüsünün hesaplanmasıyla karşıtlıkları normalleştirilebilir ve daha sonra blok içindeki tüm hücreleri normalleştirmek için bu değeri kullanır. Bu normalizasyon, aydınlatma ve gölgelenme değişikliklerine daha iyi uyum sağlar [17].

HOG tanımlayıcısının diğer tanımlayıcılara göre birkaç önemli avantajı vardır. Yerel hücreler üzerinde çalıştığı için, nesne yönelimi dışında geometrik dönüşümler yapıldığında değişmezdir. Bu değişiklikler sadece daha büyük mekânsal bölgelerde ortaya çıkacaktır. Dahası, Dalal ve Triggs'in keşfettiği gibi, kaba mekânsal örnekleme, ince uyum örneklemesi ve güçlü yerel normalizasyon, yayaların bireysel vücut hareketlerinin kabaca dik bir konumda kaldıkları sürece göz ardı edilmesine izin vermektedir. HOG tanımlayıcısı, bu nedenle görüntülerdeki insanların saptaması için oldukça uygun bir tanımlayıcıdır.

Tanımsal olarak, HOG bilgisayarla görü ve görüntü işleme kapsamında kullanılan bir öznitelik tanımlayıcısıdır. Yaya tespiti yapılırken HOG öznitelik çıkarımı yaygın olarak kullanılan yöntemlerdendir. HOG metodu görüntünün öncelikle alt hücrelere daha sonra bloklara bölünüp her bir alt bloktaki yön vektörlerinin açı değerlerini kullanılarak,

histogramlarının hesaplanması temeline dayanmaktadır. HOG yöntemi INRIA'da çalışan iki araştırmacı Dalal ve Triggs tarafından 2005 yılında yaptıkları çalışmada durağan görüntülerde yer alan yayaları tespit etmek amacıyla kullanılmıştır [15].

Işık aydınlatması ve zıtlık etkisini azalatmak için yön vektörü genlikleri normalleştirme operasyonu yapılır. Görüntünün farklı bölgelerindeki yön vektörlerinin normalleştirmek amacıyla hücreler bütünleştirilerek bloklar üretilir. Bloklar üretilirken, genel olarak üst üste eklenerek bloklar arasında kesişen hücreler elde edilir. Temel olarak iki tip blok geometrisi vardır: dikdörtgen şekilli R-HOG blokları ve dairesel C-HOG bloklarıdır. R-HOG blokları genel olarak dikdörtgen şeklindedirler ve üç parametre ile ifade edilir: her bir blokta yer alan hücrelerin sayısı, her bir hücredeki piksel sayısı ve hücre üzerinden hesaplanan histogramın öğe âdeti. Dalal ve Triggs'in yaya tespiti çalışmasında tercih ettiği en iyi sonuç veren değerler şunlardır: 3x3 hücreden oluşan blok, 6x6 pikselden oluşan hücre ve 9 öğeden oluşan yön vektörü histogramıdır [15].

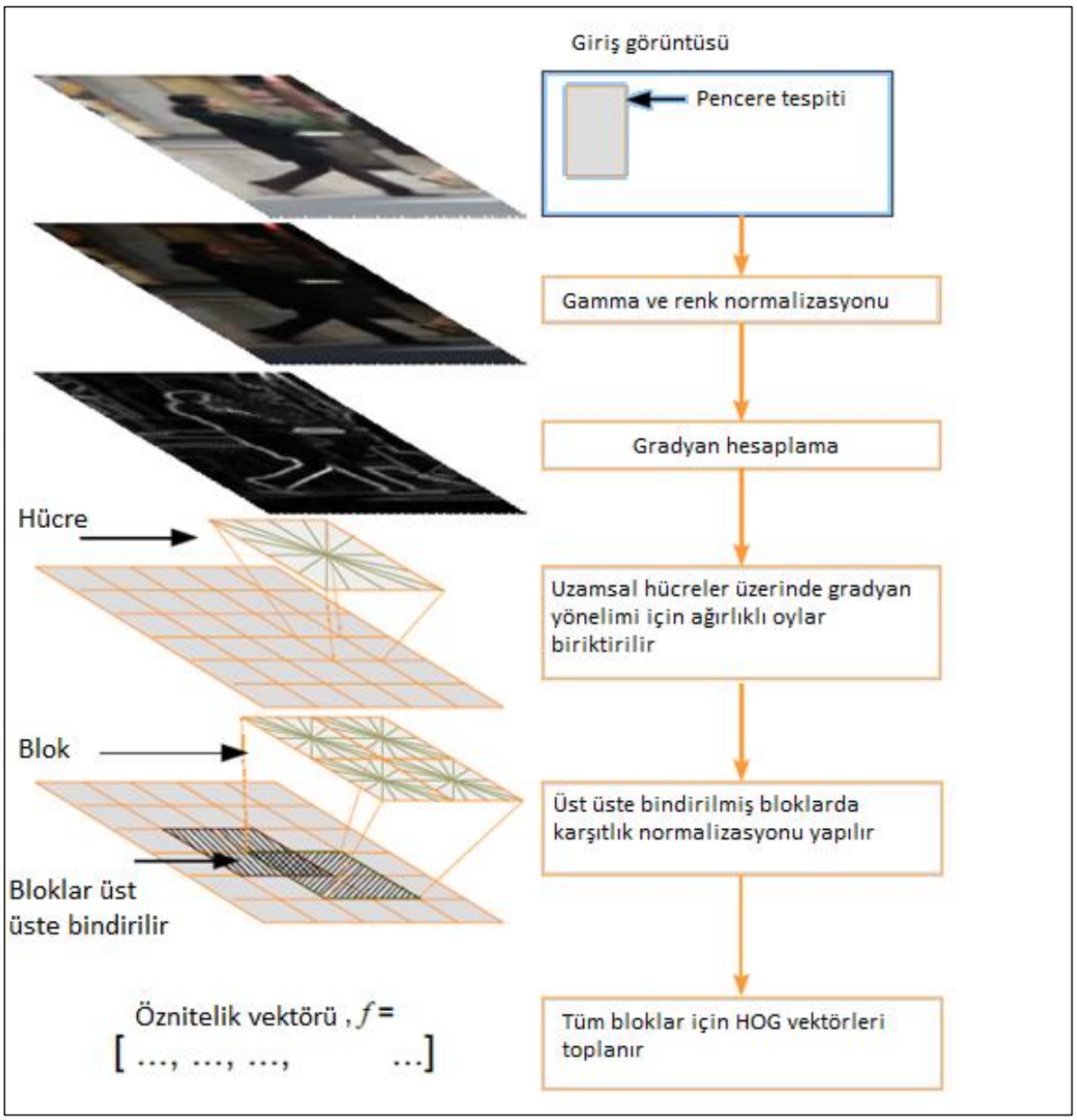


Şekil 2.3 R-HOG, C-HOG ve tek merkezli C-HOG tanımlayıcıları [18]

Şekil 2.3'de R-HOG ve C-HOG tanımlayıcı örnekleri gösterilmiştir. R-HOG tanımlayıcı blokları SIFT blok tanımlayıcılar ile benzerlik gösterse de, tek boyutta hesaplanmalar ve yön düzeltmesi olmadan elde edilen hesaplanmaları yüzünde SIFT'ten ayrıldığı açıkça görülmektedir. SIFT tanımlayıcı ise görüntüdeki genlikten bağımsız esas noktalar yardımı ile yön düzeltmesi yapılarak hesaplanmaktadır.

C-HOG tanımlayıcı blokları iki çeşittir: tek merkezi bir hücreye sahip olanlar ve açısal olarak bölünmüş hücreler. Buna ek olarak C-HOG blokları dört parametre ile ifade edilebilir: açısal veya radyal öğelerin âdeti, merkezi öğenin yarıçapı ve ek açısal öğelerin genişleme faktörünün âdeti. Yapılan bir çalışmada [15] iki yöntemin de birbirine çok yakın sonuçlar ürettiği belirtilmektedir. 4 piksellik bir yarıçapı olan bir hücre ve

genişleme faktörü katsayının iki kat artırılarak seçilmesi ile en iyi performans elde edilmektedir.



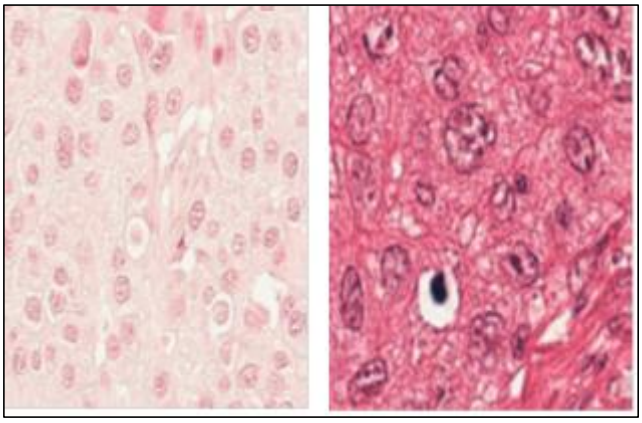
Şekil 2.4 HOG algoritması aşamaları [18]

Şekil 2.4'de HOG algoritmasının aşamaları gösterilmiştir. İlk olarak gamma doğrulaması ve renk normalizasyonu yapılır. Ortaya çıkan gri ölçekli görüntü üzerinden gradyan hesaplanır. Gradyanı alınmış görüntünün yön vektörlerinin genliği ve açısı hesaplanarak histogramı çıkarılır. Her bir hücre için gradyan histogramı oluşturulur. Daha sonra hücre histogramlarının bloklar halinde birbirleri ile normalize edilmesi sağlanır.

2.1.1 Gamma Doğrulama ve Renk Normalizasyonu

Bu bölümde isteğe bağlı olarak yapılacak olan gamma doğrulaması ile gri tonlamalı görüntüler, RGB ve LAB renk boşlukları dâhil olmak üzere birkaç giriş pikseli temsili değerlendirilmektedir. Elimizde renk bilgisi mevcut olduğunda her zaman yardımcı bize olur ve bu bize ek bilgiler sağlar. Örneğin, kişi detektörü için RGB ve LAB renk alanları karşılaştırılabilir sonuçlar verirken, gri tonlama kısıtlaması performansı düşürmektedir [18]. Renk normalizasyonu, renk değerlerini korurken görüntüdeki tüm yoğunluk değerlerini kaldırır. Bu sayede aynı renk pikselleri, gölgeleri veya aydınlatma etkinliklerini kaldırma özelliğine sahiptir. Renk normalizasyonun amacı videodaki aydınlanma, ışık, ses gibi dış etkenlerden dolayı oluşan gürültülerin bastırılmasıdır.

Renk normalizasyonu histogram eşitleme ve matematiksel hesaplama yöntemi olmak üzere 2 şekilde yapılabilir. Histogram eşitleme yönteminde görüntüdeki tüm pikseller, gri ölçekli pikseller (0 - 255 yoğunluğuyla) renk bilgisine sahip değildir. Belirlenen eşik değeri kullanarak görüntüden daha az doygun renk pikselleri temizlenir. Matematiksel hesaplama yöntemi ise her bir görüntü pikselindeki mavi, yeşil, kırmızı renk değerlerinin; toplam kırmızı, mavi, yeşil renk değerine oranlanması ile bulunur (B = B/(R+G+B), G = G/(R+G+B), R = R/(R+G+B)). Aşağıdaki Şekil 2.5'de sol tarafta kaynak bir görüntü, sağ tarafta ise bu kaynak görüntüden renk normalizasyonu uygulanarak elde edilen hedef görüntü yer almaktadır.



Şekil 2.5 Renk normalizyonu yapılmış görüntü [20]

Gamma doğrulaması görüntülerde parlaklık artırma ve aydınlatma işlemi için kullanılan etkili ve basit bir işlemdir. Kısacası bir görüntünün aydınlığını belirleyen çarpandır. Şekil

2.6'da sol tarafta gamma doğrulaması yapılmamış ham bir görüntü, sağ tarafta ise 1/2.2 oranında gamma doğrulaması yapılan görüntü yer almaktadır.



Şekil 2.6 Gamma doğrulama uygulanmış görüntü [21]

Dalal ve Triggs, gamma eşitlemesi ile gri tonlamalı, RGB ve LAB renk uzayları dâhil olmak üzere birkaç giriş pikseli temsilini değerlendirmişlerdir. Bu normalleştirmelerin performans üzerinde sadece mütevazı bir etkisi olduğuna karar vermişlerdir [18].

2.1.2 Gradyan Hesaplama

HOG tanımlayıcı kullanılırken gamma doğrulama ve renk normalizasyonu yapılması zorunu değildir çünkü tanımlayıcının aşamalarından olan normalleştirme bunu göz ardı eder. HOG yönteminde ilk adım olarak gradyant hesaplanmıştır. Gradyant değerleri, her bir görüntü için konvolüsyon yapılarak yatay ve dikey yönlerde ayrık türev maskeleri işlenerek elde edilmiştir.

İsteğe bağlı olarak Gaussian yumuşatmayı kullanarak ve birkaç farklı türev maskeden birini uygulayarak görüntü gradyanlarını hesaplanmaktadır. Bunun soncunda performansın nasıl değiştiği test edilmektedir. Renkli görüntüler (RGB veya LAB alanı) için, her bir renk kanalı için ayrı ayrı gradyanları hesaplanır ve pixelin gradyan vektörü olarak en büyük norma sahip olanı seçilmektedir.

Yön vektörlerini hesaplamak için x ve y eksenlerinde tek boyutlu bir türev maskesi uygulanır. Türev maskesi iki farklı piksel yoğunluk değerinin farkı alınarak hesaplanır. Yön vektörleri hesaplamada en çok Sobel filtresi kullanılmaktadır. Tek boyutlu Sobel filtresiyle yatay ve dikeyde gradyanlar denklem (2.1)'de verilen eşitliklerle hesaplanır.

$$g_x(x,y) = i(x+1,y) - i(x-1,y) \;\; g_y(x,y) = i(x,y+1) - i(x,y-1) \qquad (2.1)$$

Yön vektörlerinin genliği denklem (2.2)'de ve yönü ise denklem (2.3)'de verildiği gibi hesaplanır.

$$Genlik: s = \sqrt{s_x^2 + s_y^2} \quad (2.2)$$

$$Yön: \theta = \tan^{-1}\frac{s_y}{s_x} \quad (2.3)$$

Çizelge 2.1'de verilen türev maskesi kullanılarak x ve y eksenlerinde ayrı ayrı türevler hesaplanır. Genel detektör performansı, gradyanların hesaplanma şekline duyarlıdır ve standart sapma σ = 0'da ortalanmış tek boyutlu [−1, 0, 1] maskelerinin en basit şeması en iyi şekilde çalıştığı görülmüştür. Herhangi bir pürüzsüzleştirme veya herhangi bir tipteki daha büyük maskelerin tercih edilmesi sistem performansı düşürecektir. Bunun en muhtemel nedeni ise küçük detayların önemli olduğudur: görüntüler esasen kenar tabanlı ve düzgünleştirilmiş, kenar karşıtlığı ve dolayısıyla sinyali azaltılmıştır. Burdan çıkarılacak faydalı sonuç ise en uygun görüntü gradyanlarının çok hızlı ve basit bir şekilde hesaplanabilmesidir [18].

Çizelge 2.1 Yön vektörlerinin hesabı için türev maskesi

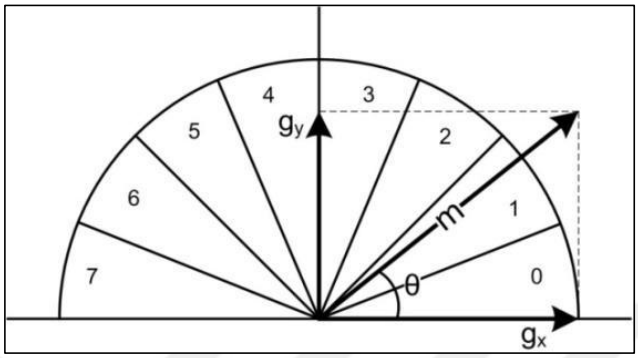
Türev maskesi	
X düzlemi türev maskesi	**Y düzlemi türev maskesi**
-1 0 1	-1 0 1

HOG hesaplanırken görüntü 128x64 piksellik pencerelere bölünür. Kenar bölgelerde gradyan hesaplamak için her bir kenara bir piksel eklenerek 130x66 piksellik eğitim görüntüleri oluşturullmaktadır. Eğitim görüntüleri hücre adı verilen 8x8 piksellik alt parçalara bölünür. Bu hücreler %50 örtüşecek bloklar halinde düzenlenir. Her blok, 2x2 hücre ve 16x16 pikselden oluşur. Böylece, her pencere 8 yatay ve 16 dikey hücre veya 7 yatay ve 15 dikey bloktan oluşur. Her bir hücre için yön vektörleri hesaplanır [22].

2.1.3 Histogram Hesaplama

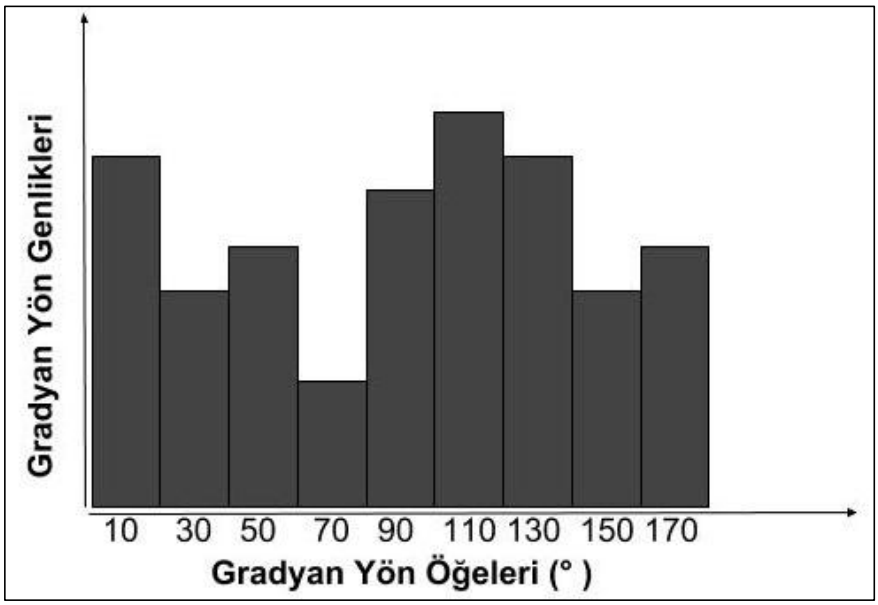
Bu aşamada görüntü bölgelere bölünerek her bölgenin gradyan bilgisi ve histogramı hesaplanır. Bu bölgeler "hücre" olarak adlandırılan, boyutları genellikle 5x5 ya da 8x8

piksel olarak seçilen karelerdir [23], [24]. Seçilen hücre boyutuna göre, 8x8 pikselden oluşan hücrelerin gradyan genliği "s" ve açısı "θ" bilgileri hesaplanır. Histogram 0-180 ya da 0-360 derece arasında hesaplanabilir ve daha iyi sonuçlar elde etmek için oylar ya gradyan büyüklüğü ya da gradyan büyüklüğünün kendisinin bir fonksiyonu kullanılarak yapılabilir.



Şekil 2.7 Gradyan genlik ve açı bilgileri [25]

Görüntünü histogramı hesaplanırken hücre içerisindeki her gradyan büyüklüğü, sahip olduğu açıya göre histogramda belirlenen bölgelere bir metoda göre paylaştırılır. Hesaplamada gradyan büyüklüğü dağıtımında genellikle lineer interpolasyon tercih edilmektedir. Her bir hücrede yer alan pikseller için 0'dan 180 dereceyi kapsayan ve 20 derecelik aralıklardan oluşan yön vektörü histogramı oluşturulur. Şekil 2.8'de gradyan yön vektörleri için örnek bir grafik oluşturulmuştur.

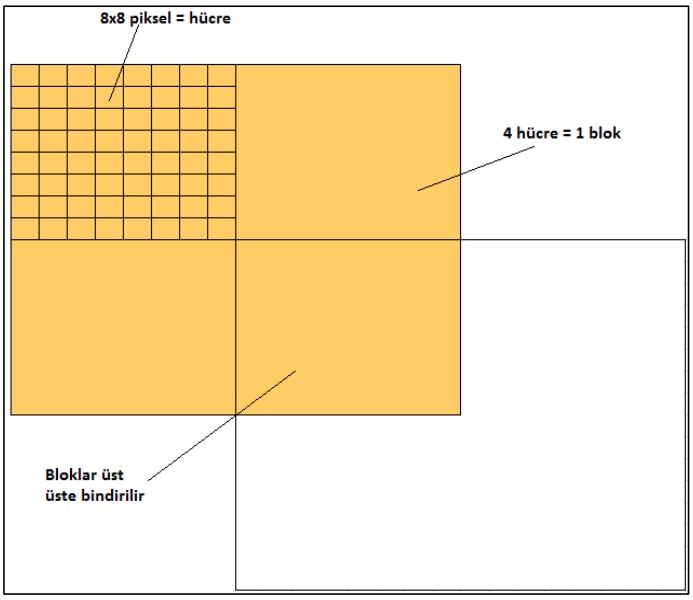


Şekil 2.8 Gradyan yön vektörü

2.1.4 Blok Normalleştirme

Görüntünün farklı bölgelerinden çıkarılan yön vektörlerinin normalleştirmek amacıyla hücreler birleştirilerek bloklar oluşturulur. Oluşturulan bloklar üst üste bindirilir ve bloklar arasında kesişen hücreler bırakılır. Dalal ve Triggs'in üzerine çalışmış olduğu makalede, 3x3 hücreden oluşan blok, 6x6 pikselden oluşan hücre ve 9 öğeden oluşan yön vektörü histogramı kullanmışlardır[18]. Bu çalışmada ise 8x8 pikselden oluşan hücre, 2x2 hücreden oluşan blok ve 9 öğeden oluşan yön vektörü histogramı kullanılmıştır.

Şekil 2.9'da bu çalışmada HOG tanımlayıcısı için kullanılan hücre ve blok yapısı gösterilmiştir. 8x8 piksel alanı kaplayan alana hücre, 2x2 hücrelere ise blok denir. Blokların üst üste bindirilmesiyle bölgesel geçişteki yoğunluk, ışık ve karşıtlık etkilerinin normalleştirmesi sağlanmıştır. Hücre histogramları bloklar halinde gruplandırılarak, blok içerisinde yer alan hücreler birbirleri ile normalize edilmiştir. Blok normalizasyonda üst üste bindirilen bloklar, histogram hücreleri üzerinde oluşturulan pencereler görüntü üzerinde kaydırılırlar. Bu işlem tüm görüntü üzerinde uygulanmaya devam edilir ve böylece sınıflandırıcılara verilecek hale getirilmiş olmaktadır.



Şekil 2.9 HOG tanımlayıcısı için kullanılan hücre ve blok yapısı

Blok normalleştirme işlemi ışık yoğunluğu, gölgeleme ve arka plan gibi faktörlerin etkisini azaltmak için bloklara uygulanır. Normalleştirme sayesinde HOG tanımlayıcı, aydınlatma ve gölgelemeden kaynaklı değişimlere karşı tolerans sağlar. Blok

normalleştirmek için denklem (2.4)'ten denklem (2.7)'e kadar verilen L1-norm, L2-norm L1-hys ve L2-hys yöntemleri mevcuttur. Önerilen algoritmada, denklem (2.6)'da verilen L2-norm yöntemi diğer yöntemlere göre daha iyi sonuç verdiği için tercih edilmiştir [22]. Bu denklemlerde v normalize edilmemiş vektör, ∈ hata miktarını ve f normalize edilmiştir vektör f_{max} ise normalize edilen vektörün maximum değerini göstermektedir.

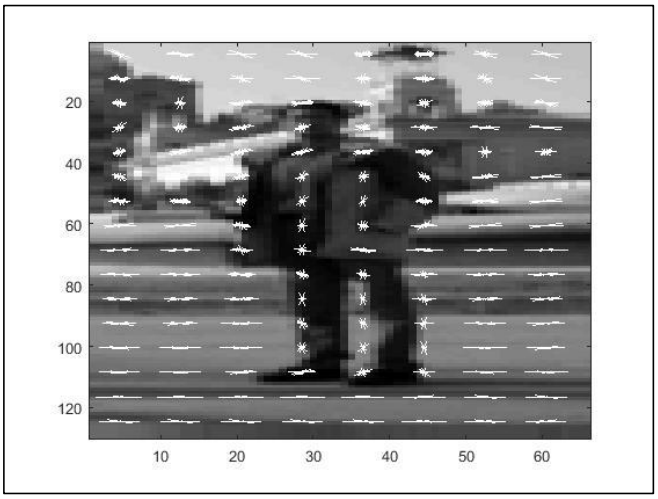
$$L1 - norm: f = \frac{v}{\sqrt{|v| + \epsilon^2}} \tag{2.4}$$

$$L1 - karekök: f = \sqrt{\frac{v}{|v| + \epsilon^2}} \tag{2.5}$$

$$L2 - norm: f = \frac{v}{\sqrt{|v|^2 + \epsilon^2}} \tag{2.6}$$

$$L2 - hys: f = \frac{v}{\sqrt{|v|^2 + \epsilon^2}} \quad f_{max} = 0{,}2 \tag{2.7}$$

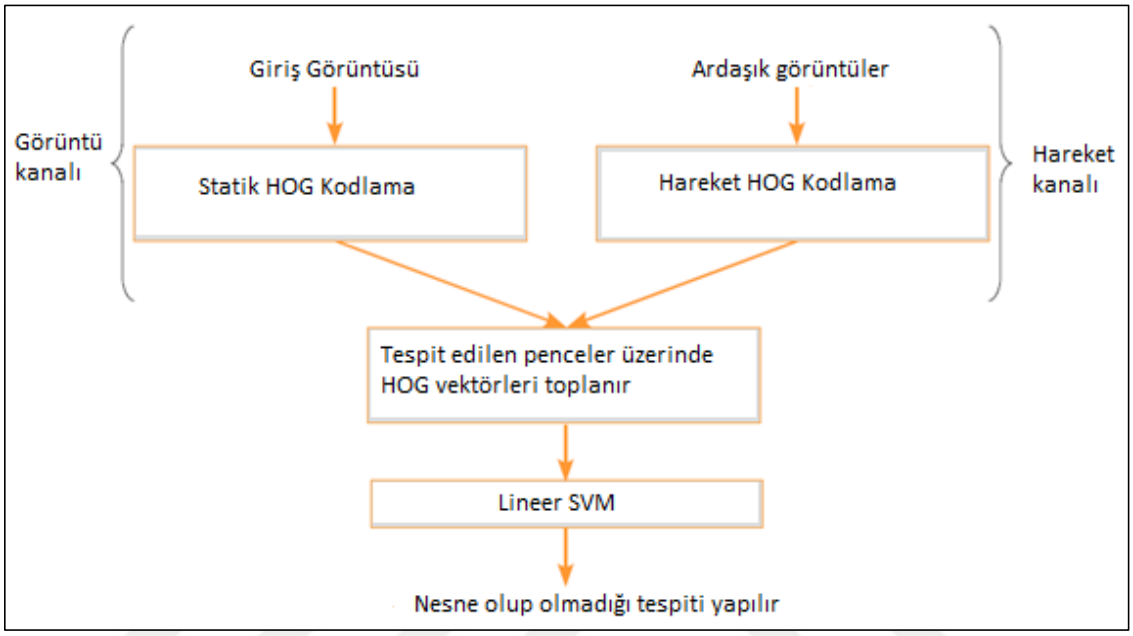
Her pencere 7 yatay ve 15 dikey bloktan oluşur. Bloklarda 4 hücre bulunur ve her hücrede 9 öğeden oluşan yön vektörleri histogramına sahiptir. Görüntünün HOG öznitelik çıkarımı yapılırken her pencere için 3780 (7x15x4x9) boyutunda bir vektör kullanılır. Şekil 2.10'da bir görüntü üzerinde önerilen yöntemdeki HOG tanımlayıcı kullanılarak oluşturulan yön vektörleri gösterilmektedir.



Şekil 2.10 Yön vektörleri çıkarılmış görüntü

Önerilen HOG tanımlayıcı ile insan tespitinin en son aşaması öğrenebilen bir sınıflandırıcıyı eğitmektir. Şekil 2.11'de bir görüntü penceresi üzerinde HOG öznitelik çıkarma uygulanmış ve nesne algılama zincirine genel bir bakış yapılmıştır. Görüntü

yakalama kanalı mevcut görüntüyü kullanır ve statik HOG'ları bu yapı üzerinden hesaplar. Hareket kanalı ise video sekansındaki ardışık görüntülerin üzerinde hareket eden HOG'ları hesaplar. Her iki HOG, algılama penceresi üzerinden toplanır ve birleştirilen vektörler ile nesne olup olmadığı anlayabilen bir sınıflandırıcıyı beslemek için kullanılır.



Şekil 2.11 HOG ile öznitelik çıkarma ve SVM ile yaya tespiti algoritması [18]

2.2 Sınıflandırıcı Algoritmaları

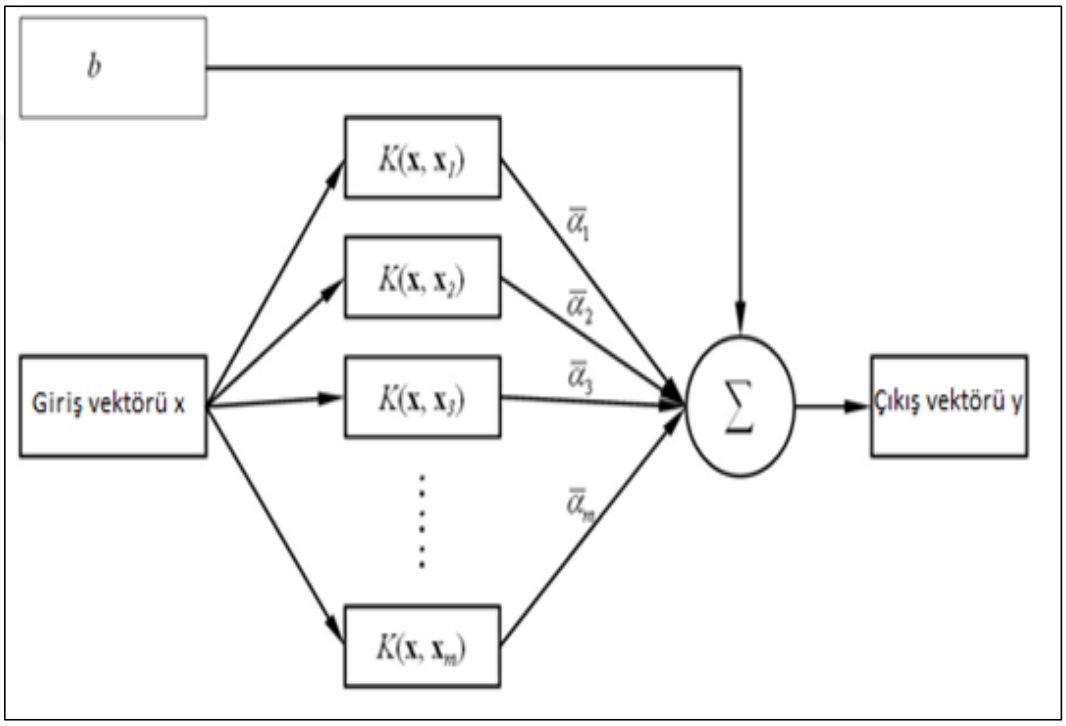
Sınıflandırma, veri kümesinin üzerinde tanımlı olan sınıflar arasında veriyi dağıtma işlemidir. Sınıflandırma için birçok algoritma geliştirilmiştir. Bu algoritmalar temel olarak belirlenen eğitim veri setinden dağılım yöntemini öğrenirler ve bu yönteme göre gelen veriyi doğru sınıfa dağıtmaya çalışırlar. HOG tanımlayıcı ile yaya tespiti algoritmasının en son aşaması ise elde edilen özniteliklerle bir sınıflandırıcıyı eğiterek veri setini doğru bir şekilde sınıflandırmaktır. Bu bölümde yapılan çalışmada kullanılan SVM, Adaboost ve KNN sınıflandırıcıları üzerinde durulacaktır.

2.2.1 SVM Sınıflandırıcı

Destek Vektör Makineleri (SVM) sınıflandırma ve regresyon için çok popüler modellerdir [27]. Makine öğrenmesinde sınıflandırma problemi, belirli bir sınıfa ait verilerden oluşmaktadır. Farklı modeller bu veriler üzerinde küçük bir kısımda

eğitilmekte ve daha sonra kalan örneklerde doğrulanmaktadır. SVM modelleri, aşırı uyuma karşı esneklik sağlar. Yerel minimum değer, yüksek boyutlu alanlardaki etkililik, bellek verimliliği ve çekirdek fonksiyonlarının çeşitliliğinin uyarlanabilmesi gibi sayısız faydalı özellik nedeniyle yaygın ve sıklıkla kullanılmaktadır.

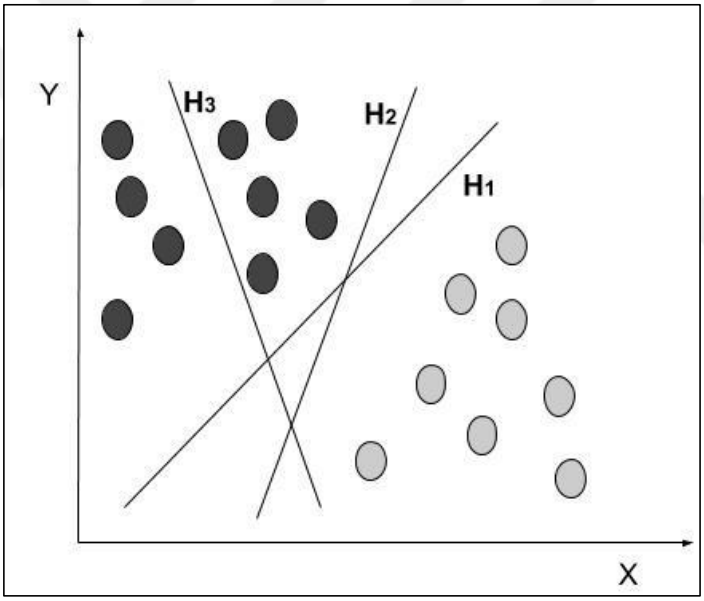
SVM yapısal risk azaltımı prensibine göre çalışan dış bükey iyileştirmesine dayalı makine öğrenmesi algoritmaları arasında yer almaktadır. Bu algoritmada, veri bilgisine ilişkin herhangi bir birleşik dağılım fonksiyonu kullanmaya gerek yoktur çünkü dağılımdan bağımsız olarak çalışan bir öğrenme algoritmasıdır.



Şekil 2.12 SVM genel ağ yapısı

Yukarıda Şekil 2.12'de verilen ağ yapısı incelendiğinde, K(x, xx) çekirdek fonksiyonlarını ve α Lagrange çarpanları görülmektedir. Çekirdek fonksiyonları yardımıyla giriş vektörlerin iç çarpımları hesaplanmaktadır. Lagrange çarpanları ise ağırlık vektörünü ifade etmektedir. SVM'de bir örneğe ilişkin çıktı işareti, girdi vektörlerin iç çarpımları ile Lagrange çarpanlarının bağımsız olarak olasılıklarının toplamına eşittir. SVM'de amaç, sınıfları birbirinden ayıracak en uygun ayırma düzleminin bulunmasıdır. Başka bir deyişle, farklı sınıflara ait olan destek vektörleri arasındaki uzaklığı maksimize hale getirmektir. SVM iki sınıflı ve çok sınıflı sınıflandırma probleminin çözümü için geliştirilmiş makine öğrenmesi algoritmaları arasında yer almaktadır.

SVM sınıflandırıcısı yaya tespitinde sıkça kullanılan sınıflandırma yöntemlerinden birisidir [26]. SVM'de amaç, kümeleri ayıracak düzleminin belirlenmesidir. Bir düzlemde yer alana iki grubu ayırmak için bir sınır çizgisi çekilir. Bu sınıflandırıcıda veri uzayında ayrıştırılan kümeler arasındaki ayrıştırma düzlemi iki sınıfı birbirinden düzgün bir şekilde ayrıştırılır. Yeni gelen veriler ise düzleme dağıtılarak ilgili sınıfı tespit edilir ve o bölgeye yerleştirilir. Şekil 2.13'de verilen 2 farklı sınıfa ait veri noktalarını, H_3 doğru parçası ile doğru bir şekilde ayrılmadığı görülmektedir. H_2 doğru parçası ise veri kümelerini ayırmakta fakat yeterince düzgün olmadığı açıkça görülmektedir. Bu durum SVM sınıflandırıcı için istenen bir durum değildir. H_1 doğru parçası ile ise veri kümeleri oldukça net bir şekilde ayrılıp en iyi çözüm bu şekilde sağlanmaktadır [27]. Sonuç olarak, özellik setlerini en iyi aralık ile ayrıştıran doğru parçası, sınıflandırma için en başarılı olan ayrıştırma çizgisidir.



Şekil 2.13 Veri setinin SVM ile iki farklı kümeye ayrılması

Bu çalışmada, eğitme aşamasında INRIA veri seti tercih edilmiştir [19]. Eğitim aşamasında kullanılan pozitif ve negatif veri setleri için HOG tanımlayıcı elde edildikten sonra, SVM sınıflandırıcısı kullanılarak HOG ile öznitelikleri çıkarılmış görüntülerden model dosyası oluşturulmuştur. Böylece oluşturulan bu model dosyası geliştirilecek çoğunluk oylmasına dayanan sınıflandırıcının modellerinden birisi olacaktır.

2.2.2 K en Yakın Komşu Sınıflandırıcı

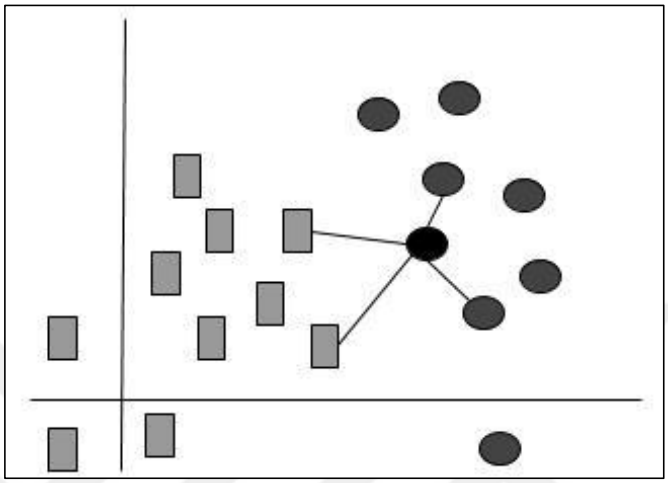
K en yakın komşu (KNN) sınıflandırıcı, örnek tabanlı öğrenmeye dayalı bir sınıflandırma algoritmasıdır. Aynı zamanda KNN algoritması sorgu vektörünün en yakın k komşuluktaki elamanlar ile sınıflandırılmasının bir sonucu olan denetlemeli öğrenme algoritmalarındandır. Bu algoritma ile yeni bir üyeyi sınıflandırabilmek için test vektörü ve eğitim vektörleri kullanılır. Bir sorgu örneği düzenlenir ve bu sorgu noktasına en yakın k tane eğitim noktası oluşturulur. Sınıflandırma ise bu k tane nesnenin en fazla komşuluk sayısına sahip olanı ile yapılmaktadır. K en yakın komşuluk uygulaması yeni gelen sorgu versini sınıflandırmak için kullanılan bir komşuluk sınıflandırma algoritmasıdır.

K en yakın komşuluk algoritması oldukça kolaydır. K en yakın komşulukları bulmak için sorgu örneği ile eğitim veri setleri arasındaki en yakın uzaklıklar belirlenir. Böylece en yakın komşular tespit edildikten sonra bu komşulardan benzerliği en çok olanın kategorisi yeni üyenin kategorisini tahmin etmekte kulanılmaktadır.

K değerinin 1'e eşit olduğu durumlarda giriş vektörü direk olarak en yakın vektörle aynı sınıfında yer almaktadır. K değerinin 1'den farklı olduğu durumlar için genel olarak K en yakın komşu giriş vektörüne olan yakınlık derecesine göre ağırlık vektörü belirlenir. Böylece daha yakın olan komşuların etkisi daha etkili olmaktadır. Ağırlık vektörü belirleme işlemi için mesafenin çarpma işlemine göre tersi oldukça yaygın olarak kullanılmaktadır. Özellikle sınıflar arasındaki dağılımın çarpık olduğu durumlarda ağırlık vektörü belirleme önemli bir rol oynar. Bir diğer yöntemde ise ağırlık vektörü olmadan K en yakın komşunun en fazla sahip olduğu sınıf test vektörünün sınıfı olarak atanır. Bu algoritmanın avantajı basit bir yapıya sahip olduğu için kolay uygulanabilmesini, hesaplama ve işlem sadeliği ve az sayıda öznitelik vektörü olduğu durumlarda güzel sonuçlar üretmesidir [30]. Buna rağmen KNN, her özniteliğe eşit derecede önem göstermesi sebebiyle bazı problemlerin çözümünde başarılı olamaz. Bu sorun normalizasyon ve ağırlık verme gibi operasyonlaral belirli bir ölçüde çözülebilir.

Temel olarak sınıflandırılmak istenen yeni eleman daha önceki bireylerden k tanesinin yakınlığına bakılarak yeni eleman ilgili sınıfa dağıtılır. Örneğin k=4 için yeni gelecek bireyi sınıflandırmak için en yakın 4 tanesi ile ilişki kurulur. Test edilecek olan örnek,

eğitim setinde yer alan örnekler ile karşılaştırılarak benzerliklerine göre sınıflandırılmaktadır [28]. Yapılan çalışmada, KNN algoritmasında örnekler arası yakınlık Öklid uzaklığı ile hesaplanır ve buna göre ilgili sınıfa yerleştirilir. Şekilde 2.14'de yeni gelen üyenin hangi sınıfa nasıl yerleştirildiği gösterilmiştir. Öklid uzaklığı hesabına göre yeni gelen siyah üye koyu gri sınıfa yakınlığından dolayı koyu gri sınıfa yerleştirilmiştir.



Şekil 2.14 Veri setine gelen yeni üyenin KNN ile sınıflandırılması

Veri setlerinden HOG ile çıkarılan öznitelikler KNN sınıflandırıcıya verilir ve en yakın dört komşuya bakılarak eğitim aşaması yapılmaktadır. Bu sınıflandırıcısı ile öznitelikleri çıkırılmış görüntülerden model dosyası oluşturulur.

2.2.3 Adaboost Sınıflandırıcı

Adabost sınıflandırıcısı karar vermede kullanılacak parametre sayısı fazla olduğunda sıkça tercih edilmektedir. Eğitim verileriyle öznitelik sınıflandırıcıların ağırlıklandırılarak birleştirilmesi prensibine dayanır [29]. Boosting zayıf öğreniciyi birleştirilerek daha güçlü bir öğrenici oluşturma işlemidir. Adaboost sınıflandırıcı ise kendinden önce gelen tahmim edici sistemi eksik öğrendiği için daha sonraki gelen verilere daha dikkatli davranarak sınıflandırır. İlk işlem olarak sınıfandırıcı eğitilir ve tüm veri seti üzerinde tahim etme işlemi yapılır. Yanlış sınıflandırılan veri örneklerinin göreli ağırlığı artılır ve yeni bir sınıflandırıcı ortaya çıkar. Bu sınıflandırıcı ile tekrar eğitilir ve yeniden sınıflandırılmaktadır.

Adaboost sınıflandırıcısının birçok çeşiti vardır ancak bunların temel çalışma prensibi, eğitme verileri kullanılarak ortaya çıkarılan öznitelik sınıflandırıcıların ağırlıklandırılarak

birleştirilmesi işlemine dayanmaktadır. Buna göre eğitime başlanırken her örneğe ilk başta eşit ağırlık verilir. Bu ağırlıklarla her bir sınıflandırıcının tüm örnekler için verdiği yanıtlara çarpma işlemi uygulanır. En az toplam hataya sahip olan öznitelik sınıflandırıcı tercih edilir. Seçilen öznitelik sınıflandırıcının hata yaptığı örneklerin ağırlıkları arttırılarak örnek ağırlıkları güncellenir ve güncellenen bu yeni ağırlıklarla her bir öznitelik sınıflandırıcı için yeni ağırlık vektörü bulunur ve en iyi performansı üreten sınıflandırıcı seçilerek ana sınıflandırıcıya eklenir. Bu döngüye hedeflenen başarı oranını sağlayıncaya kadar ya da hata oranı azalmaya devam ettiği sürece boyunca devam edilir. Bu işlem hata matrisi üzerinde her bir örnek için ağırlıkların saklandığı, örnek sayısı n olmak üzere n x 1 boyutunda bir matris oluşturak yapılır.

Daha sonra oluşturulan örnek ağırlık matrisi ile hata matrisinin kolonları tek tek noktasal vektör çarpımı ile çarpılır. Bu çarpımlar sonucunda ortaya çıkan yeni kolonlardan en az hata oranına sahip olan kolon seçilir, bir diğer deyişle bu kolonu oluşturmak için ağırlık kolonunun çarpılmış olduğu kolon ana sınıflandırıcıya eklenecek öznitelik sınıflandırıcı olarak seçilmiş olur. Her bir kolon için hata oranı hm(j) (2.8)'deki gibi hesaplanır. Böylece algoritmada her döngüde yeni bir tane öznitelik sınıflandırıcı seçilmektedir. Belirlenmemiş olan bu öznitelik sınıflandırıcı önceki döngülerde de seçilmiş olabilir ancak aynı sınıflandırıcının sürekli olarak bu şekilde seçilmesi beklenen bir durum değildir. Her döngü sonunda seçilen sınıflandırıcıya sınıflandırıcının hatası ile alakalı olarak (2.9) ile hesaplanan bir katsayı (b) atanır.

$$hm(j) = \sum_{i=0}^{n} X_{ij} \,.\, W_i \tag{2.8}$$

$$b(t) = \mathrm{hm} \,/\, (1 - \mathrm{hm}) \tag{2.9}$$

$$W_i(t+1) = W_i(t) \,.\, b(t)^{\,1-X_{ij}} \tag{2.10}$$

Örneklerin ağırlıkları, o döngüde seçilen en iyi sınıflandırıcının hata oranıyla uygun olacak bir şekilde güncellenir. Denklem (2.10)'da gösterildiği gibi bu güncelleme doğru sınıflandırılan örnek katsayılarının azaltılmasıyla oluşturulur. Çünkü denklem (2.10) ifadesindeki Xij değişkeni hata matrisindeki j'nci sınıflandırıcının i'nci örneğe verdiği yanıtı ifade eder. Önceki bölümlerde de anlatıldığı üzere sadece hata durumunda Xij değişkeni 1'e eşit olmaktadır. Bu nedenle üslü ifadenin üssü hata durumunda 0'a eşit

olur. Sonuçta güncelleme katsayısı 1'e eşit olur ve güncelleme sonucu ağırlıkta değişiklik olmaz.

Çizelge 2.2 Adaboost algoritmasının genel akışı [53]

- Eğitim örnekleri $(X_1,Y_1),\ldots, (X_n,Y_n)$ şeklinde formatlanır. Burada y_i, pozitif örnekler için 1, negatif örnekler içinse 0 olarak set edilir.
- m ve n sırasıyla negatif ve pozitif örneklerin sayısını temsil etmek üzere ağırlık vektörleri $W_{1,i} = \frac{1}{2m}, \frac{1}{2n}$ şeklinde her $y_i \in \{0,1\}$ için ilk değerleri atanır.
- T iterasyon sayısı olmak üzere, her t=1,.,T için :
  1. Ağırlıklar normalize edilir.

$$W_{t,i} \leftarrow \frac{W_{t,i}}{\sum_{j=0}^{n} W_{t,j}}$$

  2. Her bir j özniteliği için, sadece bu j özniteliğini içeren bir h_j sınıflandırıcısı eğitilir. Hata oranı ise W_T ağrılık vektörüne göre hesaplanır.

$$\varepsilon_j = \sum_i w_i \left| h_j(x_i) - y_i \right|$$

  3. En az ε_t hatasına sahip h_t sınıflandırıcısı seçilir.
  4. Ağırlık vektörleri güncellenir. X_i eğer doğru olarak sınıflandırılmışsa $e_i = 0$, aksi takdirde $e_i = 1$ olarak belirlenir. β_t ise $\beta_t = \frac{\varepsilon_t}{1-\varepsilon_t}$ formüyle hesaplanır.

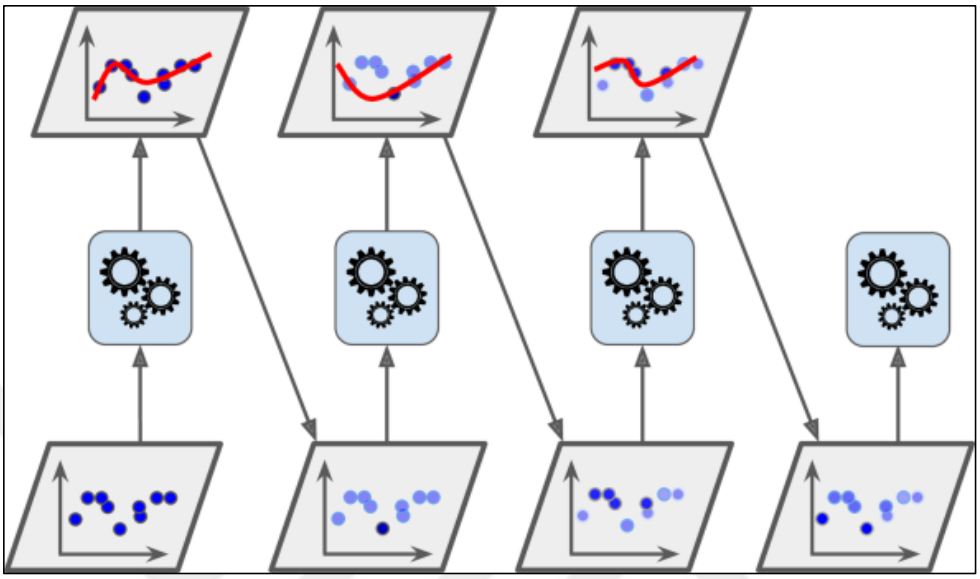
$$W_{t+1,i} = W_{t,i} \cdot \beta_t^{1-e_i}$$

  5. Nihayi olarak ortaya çıkan sınıflandırıcı:

$$h(x) = \begin{cases} 1, (x+a)^n = \sum_{t-1}^{T} a_t h_t(x) > \frac{1}{2} \sum_{t-1}^{T} a_t \\ 0, \qquad\qquad\qquad\qquad\qquad\qquad diğer \end{cases}$$

Burada $a_t = log \frac{1}{1-\beta_t}$ olarak hesaplamalar yapılır.

INRIA veri setindeki pozitif ve negatif görüntüler için öznitelikler çıkarılıp, oluşturulan öznitelikler Adaboost sınıflandırıcı ile eğitilerek model dosyası oluşturulmuştur. Adaboost sınıflandırcı için yaklaşık 100 iterasyon, öğrenici olarak ise karar verme ağacı tercih edilmiştir.

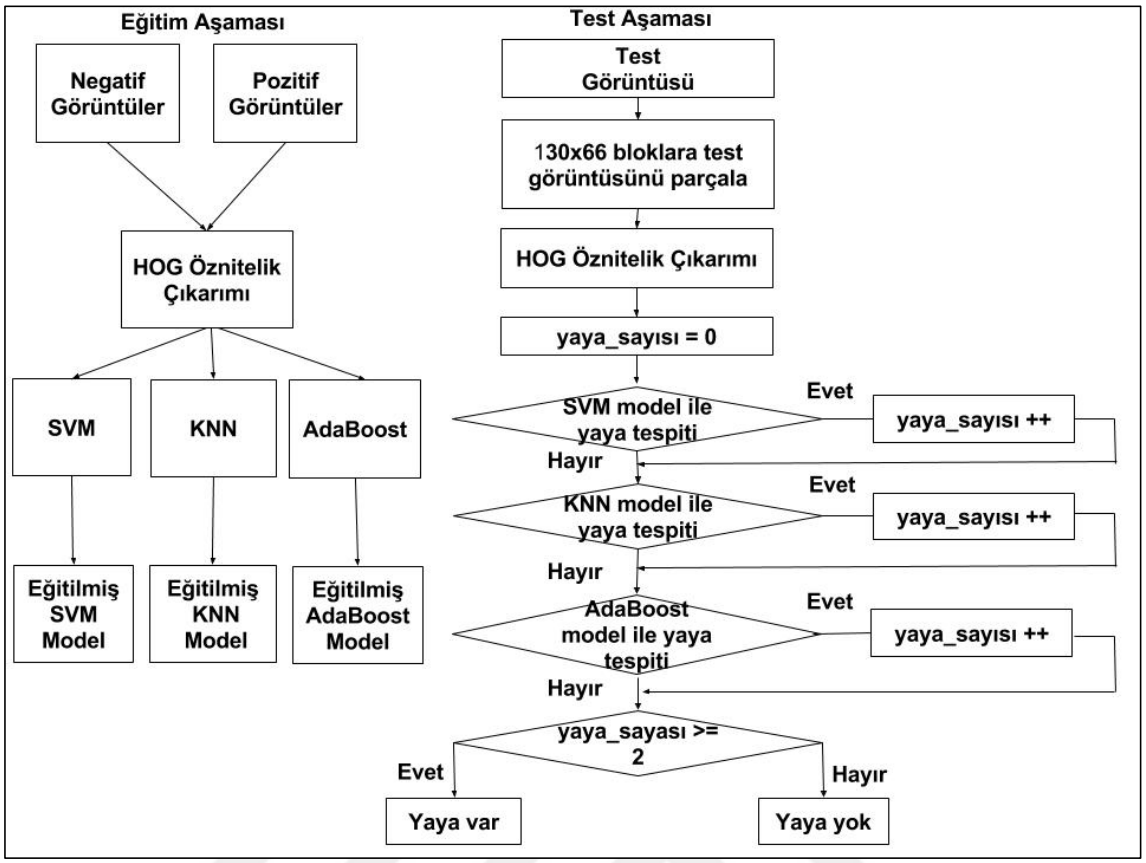


Şekil 2.15 Veri setinin Adaboost ile sınıflandırılması

Yukarıda Şekil 2.15'te elde bulunan bir veri setinin Adaboost algoritması ile sınıflandırılması gösterilmiştir. Bu sınıflandırma işlemi istenen sonuçları üretebilen bir sınıflandırıcı ağırlık vektörü elde edilene kadar tekrarlanmaktadır.

2.3 Geliştirilen Sınıflandırma Algoritması

Bu çalışmada farklı sınıflandırıcılar bir araya getirilerek görüntüde yaya tespiti yaparken hatalı pozitif oranını düşüren çoğunluk oylamasına dayanan bir yaya tespit algoritması geliştirilmiştir. Üretilen sonuçlar göz önüne alındığında önerilen algoritmanın doğru pozitif oranını yükseltirken hatalı pozitif oranını indirdiği açıkca görülmektedir. Buradan yola çıkarak çoğunluk oylaması yönteminin ortaya çıkan problemin çözümü için daha iyi olduğu görülebilir. Üç sınıflandırıcının ortak kararına bakılarak hatalı pozitif sayısının indirildiği görülmektedir ancak bu durumda doğru pozitif oranı da azalmaktadır. Problemin ana yapısı dikkate alındığında bu durumun istenmeyen bir sonuç olduğu açıkça görülmektedir fakat burada hedeflenen yüksek başarım oranında yaya tespiti yapan bir algoritma sunmaktır. Şekil 2.16'da ise geliştirlen sınıflandırma algoritmasının akış diyagramı gösterilmiştir.



Şekil 2.16 Çoklu sınıflandırmalı yaya tespiti diyagramı [55]

Eğitim aşamasında netagif ve pozitif veri setlerin öznitelikleri çıkarılarak SVM, KNN ve Adaboost sınıflandırıcılarına verilir. Her sınıflandırıcı eğitilmiş bir model çıkırarır. Eğitilmiş modellere giriş görüntüsü verilir ve içersinde yaya tespiti yapılır. Her model tespit ettiği yaya sayısı için sayacı bir artırır ve yaya sayısı ikiden fazlaysa bu görüntüde yaya var sonucuna ulaşılır.

BÖLÜM 3

VİDEO İŞARETLERİNDE İNSAN TESPİTİ

Yayaların doğru tespit edilmesi ve takibi, görüntü işleme uygulamaları için gerekli olan iki temel bileşendir. Konu üzerine yapılan çok sayıda potansiyel uygulama nedeniyle, yaya tespiti ve izleme yöntemleri bilgisayarla görü araştırmalarında son derece aktif bir çalışma alanı haline gelmiştir. Bunun sonucu olarak farklı teknikler kullanan yaya bölütleme teknikleri öneren sistemler oldukça popüler bir hale gelmiştir. Bu teknikler geleneksel iki boyutlu bilgisayar görme teknikleri uygulamalarını ve kızılötesi, lazer tarayıcılar, sonar veya radar gibi diğer sensör sistemlerinin kullanılmasını içermektedir. Önerilen yaklaşımların çoğu, belirli varsayımların kabul edilmesine izin veren kısıtlı senaryolarla sunulduğunda iyi sonuçlar üretmektedir. Bu kısıtlayıcı varsayımlar, genel olarak yaya tespitinde karmaşık olan faktörlerin sayısının azaltılması ve böylece bu problemin izlenebilir hale getirilmesi sağlanmıştır. Bu varsayımlara çevresel koşullar, yaya görünüşü, trafik akış yoğunluğu, yaya ve arka plan renk yoğunluğu bilgisi, bir kişinin sahnede var olduğu sürenin uzunluğu, bir kişinin sahneye girmemesi ve hatta sahne içindeki kişilerin sayısı örnek olarak verilebilir. Maalesef, bu varsayımlardan ötürü, kısıtlı ortamlarda uzun süreli videolar için güvenilir sonuçlar elde edilebilen az sayıda yaklaşım vardır [31].

Karmaşık bir ortamda yaya tespiti oldukça zor bir durumdur, çünkü insanlar birbirleriyle etkileşime girerler, gruplar oluştururlar ve beklenmedik şekillerde hareket edebilirler. Bu karmaşık durumlardan ve ortam özelliklerinin değişmesinden etkilenmemek için farklı hareketlerle başa çıkabilen sağlam yöntemlere ihtiyaç duyulmaktadır. Sistem tarafından izlenen ortamdaki değişiklikleri bastırabilmek için,

ortam ve ışık değişiklikleri ile başa çıkabilecek sağlam bir arka plan modeli tasarlanmalıdır.

Bu bölümde sırasıyla video işaretlerinde yaya tespiti uygulamaları, karşılaşılan zorluluklar, hareket segmentasyonu, yaya tespiti ve takibinden bahsedilecektir.

3.1 Yaya Tespiti Uygulamaları

Yaya tespiti ve takibi, çeşitli akıllı sistemler çerçevesinde önemli bir teknoloji olarak görülebilir. Bu teknoloji, bir görüntüde kimin nerede olduğunu ve yapılan eylemlerinin neler olduğunu anlamak için bir anahtardır. Bir uygulamanın çerçevesindeki diğer katmanların bu kişiler hakkındaki bilgileri ortaya çıkarma potansiyeli vardır. Bilgisayar tabanlı çeşitli uygulamalar için doğru yaya algılaması ve takibi bir ön koşuldur. Bu uygulamaların bazılarına bir genel bakış yapılacaktır.

Çevresel zekânın (ambient intelligence) amacı [32], mevcut insanların gereksinimlerini kolaylaştırmak için akıllıca adapte olabilen ortamları tasvir eder. AmI (ambient intelligence), kullanıcılar ve bilgi sistemleri arasında küresel bir arayüz olarak hareket etmek için çevreye sorunsuz bir şekilde bütünleşmiş akıllı cihazlar ve sensörlerden oluşan bir ağ sistemini kullanır [33]. Bu şekilde büyüyen ortamın kontrolü mevcut insan kullanıcılarının davranışına uygun şekilde karşılık vererek eyleme yönelik hale gelir. Bu sayede çeşitli uygulama senaryolarında hem tekil bireyler hem de daha büyük insan toplulukları için birçok fayda vaat etmektedir.

AmI sisteminin gerçekleşmesi için, çeşitli disiplinlerden derlenmiş bir dizi önemli teknoloji gerekmektedir. Bu teknolojilerden göze çarpanlardan bazıları; sensör donanımı, kablosuz ve sabit iletişim sistemleri, yazılım tasarımı, bilgi füzyonu, akıllı ajanlar bunlardan sadece birkaç tanesidir. Buna ek olarak, video görüntülerinde insanların rahat bir şekilde algılanması ve izlenmesi, bir sahnede kimin nerede olduğunu ve bir AmI çerçevesindeki diğer katmanların bu insanlar hakkındaki bilgileri ortaya çıkarmasına nasıl izin verildiğini bildiğinden beri önemli bir teknolojidir. Buna örnek olarak otomatik yaya trafik ışığı sistemini gösterilebilir. Yayaları açıkça tespit etmek gerekli bir ön işlem adımıdır, ancak bir kişi sahnede olduğu için yoldan geçmek istedikleri anlamına gelmez fakat kişi kavşağa doğru yürürse, durur ve beklerse,

muhtemelen insan olduğu varsayılabilir. Bu bilginin elde edilmesi, yaya tespiti ve gerçek dünyadaki kalabalık ortamlarda izlenmesi düşünüldüğünde önemli zorluklar olduğu görülmüştür. RFID etiketleme, görüntülerde bunu belirlemek için olası bir çözümdür ancak görüntüye girinceye kadar insanlarla etkileşimin olmadığı senaryolarda yardımcı olamaz. Diğer örnek AmI uygulamaları, yayaların sayılması ve sınıflandırılması, örneğin otobüs duraklarında bekleyen tekerlekli sandalye kullanıcılarının sayısı, taksi sıraları, asansörler gibi diğer kamu hizmeti uygulamalarını içerir. Böylece gerektiğinde uygun taşıma araçları gönderilebilir. AmI ev ve ofis gibi kamusal alanlarda da diğer gelişmiş hizmetleri öngörmektedir [32]. Bu tür uygulamalar, bir odada bulunan kişi sayısını otomatik olarak özetleyen akıllı toplantı odalarında, otomatik araçlardaki yaya çarpışma uyarı sistemlerine bağlı olarak sıcaklık ve havalandırma seviyelerini otomatik olarak düzenlemede kullanılır.

Bir başka önemli uygulama alanı da güvenlik sistemleridir. Bu tartışmanın amaçları için, AmI ve genel güvenlik uygulamaları arasında bir ayrım yapılır. Bu sınıflandırmada, AmI uygulamalarının tasarımı ile insanların hareketlerini doğrudan etkiledikleri ve sistemin gereksinimlerine uygun özellikler tasarlandığı görülmüştür. Bu eylemler genellikle kullanıcılar için açıktır ve daha fazla son kullanıcının yaşam standardına yardımcı olmak veya geliştirmek için tasarlanmıştır. Ancak, bu çalışmalar kapsamında güvenlik uygulamaları, video görüntülerindeki insanların eylemlerini çevreyi etkilemeyecek veya sistemin tepkisini önleyecek şekilde tanımlanmıştır. Örneğin, bazı güvenlik uygulamalarında, bir kişinin eylemleri güvenlik alarmlarını tetikleyebilir. Bu alarm, güvenlik kapılarının otomatik olarak kilitlenmesiyle sonuçlanırsa, o kişinin hareketleri çevreyi etkilemiştir. Bu açıkça bir AmI uygulamasının özelliğidir. Ancak, AmI uygulamalarından farklı olarak, bu değişiklikler çevreye olan yardımı engellemek için geliştirilmiştir. Güvenlik sistemi, bir olayın insan tarafından doğrulanması için bu tür çevresel değişiklikleri ertelerse, gözetleme ile AmI arasındaki bu örtüşme kaldırılabilir. Bu nedenle, uygulamanın kendisi çevrede bir değişikliğe neden olmaz.

Güvenlik uygulamaları, kalabalık yoğunluk istatistikleri ve tıkanıklık analizi için yayaları saymayı, izlemeyi, akıllı güvenlik gözetleme sistemlerine götüren daha karmaşık analizlere kadar uzanabilir. Yaya sayıları ve akış istatistiki bilgileri, alışveriş merkezleri, spor stadyumları, iş alanları, havaalanları ve yaya yolları geçişleri gibi tesislerin tasarımı

ve işletimi dâhil olmak üzere çeşitli uygulamalarda kullanılmaktadır [34]. Buna ek olarak, istatistiksel bilgiler güvenlik nedenleriyle, örneğin yangın durumunda binanın doluluk oranlarını bilmek için faydalıdır. Bununla birlikte, bu gözetim uygulamaları, yalnızca bir sahne boyunca yaya akışını öngörmeyi gerektirir. Güvenlik uygulamaları için genellikle daha kapsamlı bir gözetim analizi gereklidir. Örneğin, havaalanları ve demiryolları gibi kamusal alanlarda şüpheli davranışları tespit etmede bu analizler önemlidir.

3.2 Yaya Tespitinde Karşılaşın Zorluklar

Kısıtlı koşullar altında yayaların düzgün bir şekilde segmentasyonu ve takibi, onu bilgisayarla görüntü işleme çalışmalarında en zorlu problemlerden biri haline getirmektedir. Bu karmaşık faktörlerin, gerçek dünya senaryolarında yaya tespitinin mümkün olması için bilgisayar ile görme sistemleri tarafından kabul edilmesi ve ele alınması gerekmektedir. Bu tez çalışmasında, mümkün olduğunca bu zorlukların çoğunun üstesinden gelinmeye çalışılmıştır. Bu kısımda yaya tespit ve takip teknikleri uygulanırken karşılaşan zorluklar şunlardır:

- Yaya görünümündeki farklılıklar yayaların yerel ve küresel görünümlerindeki çeşidi, tipi ve giysi tarzlarından kaynaklanabilir [35]. Bu nedenle, bacakların görünümü gibi belirli bir özellik üzerinden yaya tespit etme tekniğine dayanmak, etek giyenler gibi tüm yaya sınıfları için geçerli olmayabilir.
- Bir yaya kamera görüntüsüne göre olası açı ve yönlerde konumlanabilir. Bu zorluklara ek olarak, belirli bir kamera yüksekliği, dönüşü ve yönü için geliştirilen teknikler, ikinci bir kamera pozisyonu için geçerli olmayabilir. Örneğin, literatürde bir baş üstü veya kuş uçuşu görünümü için geçerli olan teknikler, kamera daha eğik bir açıda konumlandırıldığında genellikle geçerli olmamaktadır.
- Yayalar video görüntülerinde kameradan farklı mesafelerde geçmektedir. Kameraya yakın yayaların görünümü, uzak mesafeden geçenlere göre önemli ölçüde farklı olabilir.

- Yaya siluetleri, sırt çantaları, şapkalar, evrak çantaları, el çantaları veya alışveriş çantaları gibi çok sayıda aksesuar ile de gizlenebilir.
- Bir yayanın bir ya da birkaç başka yaya grubunda, özellikle yayaların bir kalabalığın içinde yer alması halinde tespit edilmesini güçleştirmektedir.
- Bir yaya, kameranın görüş alanı içinde sadece kısmen görünebilir. Bu durum yayaların sahneye girip çıktıklarını ya da hiç girmediklerini belirtebilir.
- Yaya tespit sistemleri belirli bir sahnedeki tüm ön plan nesneleri, şekil ve boyut benzer olsa bile yayaları tanımlayamayabilir. Bir sahnede görünebilecek tüm diğer nesnelerden modelleri ayırıp tahmin etmek ve tanımlamak zordur.
- Yaya tespiti, hareketli arka planlar, değişen aydınlatma ve hava koşulları, pencerelere yansımaları, yayalar ve görüntüdeki diğer nesneler tarafından yapılan gölgeler gibi ekstra zorlukları da beraberinde getirir.
- Yaya pozları görüntülerde farklı şekillerde olabilir. Örneğin sevinç, üzüntü, uyku gibi durumlarında insanlar farklı pozlar vermektedir.
- Bir yaya, doğrusal olmayan ve öngörülemeyen bir şekilde hareket edebilir ve bu da zor takibe sebep olabilir. Örneğin, gerçek dünyadaki senaryolarda, belirli bir yaya, beklenmedik bir şekilde yürüyebilir, koşabilir, durabilir veya dönebilir.
- Daha önce izlenmemiş olan yayalar yeni olarak kabul edilmeli ve daha önce izlenen yayalarla karıştırılmamalıdır.

3.3 Hareket Bölütleme

Hareketli nesnelerin tanımlanması, literatürde önerilen çok sayıda yaya algılama algoritmasında temel ve kritik bir adımdır. Bu amaçla, hareketli bölgeleri veya nesneleri geçici bir görüntü sekansından doğru şekilde saptamayı ve segmentlemeyi amaçlayan çeşitli hareket bölütleme teknikleri kullanılır. Belirli bir yaya algılama tekniği için, hangi hareket bölütleme tekniğinin kullanılacağının seçimi, hem algoritmanın özel gereksinimlerine hem de senaryo, kamera hareketi ve çevre koşulları ile ilgili önceden bir dizi varsayımlara bağlıdır. Genel olarak üç ortak hareket analizi yöntemi yaya tespit

algoritmaları ile uygulanır. Bu yöntemler arka plan çıkarma [36,37], zamansal farklılaşma [38,39] ve optik akış [40,41] yöntemleridir.

Bu üç yaklaşımdan, geleneksel arka plan çıkarma en popüler tekniktir. Bununla birlikte, [40, 41] gibi bazı teknikler, optik akış durumunda arka plan çıkarma gibi yöntemine göre daha kullanışlıdır. Bunlara ek olarak, zamansal farklılık, arka plan modelinin önceki bir görüntü çerçevesine karşılık gelecek şekilde ayarlandığı özel bir arka plan çıkarma tekniği olarak düşünülebilir.

Arkaplan çıkarma ve ön plan çıkarma işlemleri genel olarak birlikte kullanımaktadır. Arka plan çıkarma yaklaşımları, statik bir kamera olduğunu ve ön plan nesnelerinin görüntü özelliklerinin renk yoğunluğu veya kenar gradyan bilgisi gibi arka planınkine göre farklı olduğunu varsaymaktadır. Buna ek olarak, aydınlatma koşullarında çok az değişim olacağı veya bir değişiklik meydana geldiğinde kademeli aydınlatma değişikliği olacağı varsayımı yapılmaktadır. Bu tür teknikler genellikle ilgili görüntü özelliklerine göre arka planı modellemektedir.

Ön plan pikselleri, bir giriş görüntüsünden çıkarılan özellikler arka plan modelininkilerden büyük ölçüde farklı olduğunda tespit edilebilir. Yaya tespit uygulamalarında kullanılmak üzere literatürde çeşitli arka plan modelleri önerilmiştir. Bu modeller sırasıyla ön plan nesneleri bulunmayan sahnelerin görüntüsü [42], model eğitme boyunca yer alan her piksel için minimum, maksimum ve en büyük kare arası mutlak farkı [43], geçmiş çerçevelerde bulunan piksellerin medyan ve ortalama değerleri [41], piksellerin varyansının Gauss dağılımın bir karışımı (MoG) ile modellenmesi [44], parametrik olmayan modelleri kullanan [45], ve kod çizelgesi tabanlı arka plan çıkarma modelleridir.

Bu modellerin büyük bir kısmı geleneksel renk yoğunluğu değerlerine uygulanmıştır. Diğer teknikler ise belirli görüntü niteliklerinin değişen aydınlatma koşullarına olan dayanıklılığından yaralanarak yoğunluk değerleri yerine arka plan görüntüsü niteliklerini modellemektedir.

Arka plan tahmini tamamlandıktan sonra, ön planı ve arka planı ayırmak için kullanılan algoritma, ön plan için sadece geçerli arka plan ile video kameranın görüntüsü arasındaki büyük farklılığı temsil eder. Bu süreç, çerçeve farklılaştırma algoritmasına

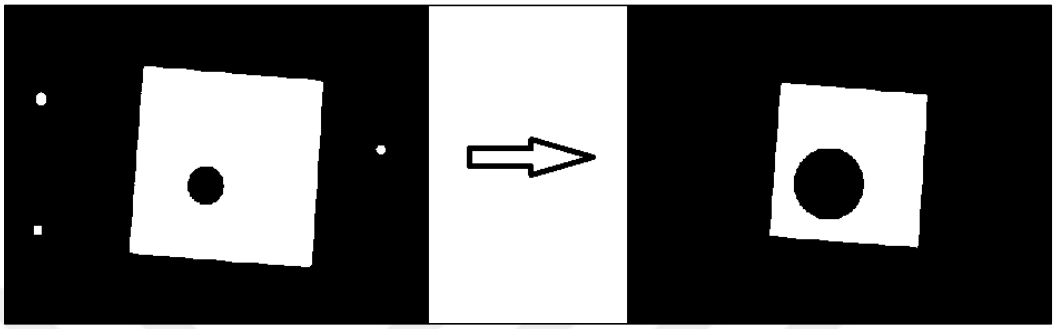
çok yakın fakat tam olarak aynı değildir. Ön plan ve görüntünün arka planını, çerçeve farklılaştırma işlemini kullanarak birbirini izleyen iki kare arasındaki farkı ayırmak dolukça zordur. Bu, ön plandaki nesnelerin kenarlarını vurgulama eğilimine sahip olduğundan, görüntü analizi daha zor olacaktır. Çerçeve farklılaştırma algoritması kullanmamanın başka bir nedeni de, ön plandaki sabit nesneleri algılamanın tamamen etkisiz olmasıdır. Sonuç olarak, insanlar sayma sistemi, ön plandaki nesneleri algılamak için arka plan çıkarma adımını ve hareketi algılamak için çerçeve ayrımını kullanır.

Bu algoritmanın ilk adımı, geçerli arkaplan ile mevcut çerçeve arasındaki mutlak farktan kaynaklanan yeni bir görüntüden arkaplan çıkarma yapmaktır. Arka plan çıkarma görüntüsü, gri ölçekli bir görüntüdür. Böylelikle bölümlenmiş görüntüyü yani ön planın ve arka planın ayırmak için bir ikili görüntüye dönüştürülmesi gerekir. Gri tonlamalı bir görüntüyü 0-255 arası değerleri bir ikili görüntüde dönüştürmek için bellirli bir eşik değerlere göre müdahale edilmelidir. Bu eşikten daha küçük olan tüm piksel değerleri, sahnenin arka planı olarak görüntülenir. Bu durum çoğu zaman, yakın bir değere sahip olan ve hareket eden nesnelerin gölgelerini temsil eden piksellerin bir kısmını ortadan kaldıracak çok sayıda gürültülü piksellerini bastırmaktadır. Aslında gri ölçekli bir görüntüde, bir nesnenin gölgesi çoğu zaman pikseliğin özelliğini ve rengini pek değiştirmez. Yani bu gölge, arka plan çıkarma işleminde küçük bir değere sahiptir.

Ancak bu eşik değer, tüm yanlış pikselleri sahnenin arka planında bulunan pikselleri ortadan kaldırmaya yetmez. Bu algoritmanın ikinci adımı, ikili görüntüde bölümlenmiş olan bir morfoloji açma operasyonudur. Bu morfolojik operasyonun ilgisi, gürültü tarafından oluşturulan küçük nesneleri çıkarmasıdır. Morfoloji açma operasyonu aynı yapı elemanı aşındırma ve yayma işlemi ile matematiksel morfoloji alanında iki temel operatörden oluşur. İlk bakışta, bir aşındırma yapılır ve daha sonra aynı yapı elemanı bir yayma işlemi ile devam eder. Fakat durum aslında böyle değildir. Bir aşındırma operasyonu bazı küçük nesneleri silme yeteneğine sahiptir, bu nedenle bu nesnelerin tekrar ortaya çıkmaması için bir yayma işlemi yapılır, böylece ikili görüntü üzerinde bir morfoloji açılmasının sonucu bu orijinal görüntüden tam olarak ayrıştırılır. Arkaplan çıkarma ile ön plan nesnelerine karşılık gelen imge bölgeleri görüntüleri oluşturulmaktadır. Bu kısımda kullanılan farklı matematiksel morfoloji operasyonlara değenilecektir.

3.3.1 Morfolojik Aşındırma Operasyonu

Arka plan modelleme de görüntülerde kullanılan ilk morfolojik işlem aşındırmadır. Temel bir işlemdir ve başlıca özelliği farklı ön plan bölgelerin sınırlarını aşındırmaktır. Böylece bu ön plan nesneleri küçülecektir. Şekil 3.1'de görüldüğü üzere çok azı tamamen yok olacaktır ve nesnelerdeki delikler daha büyük görünecektir.

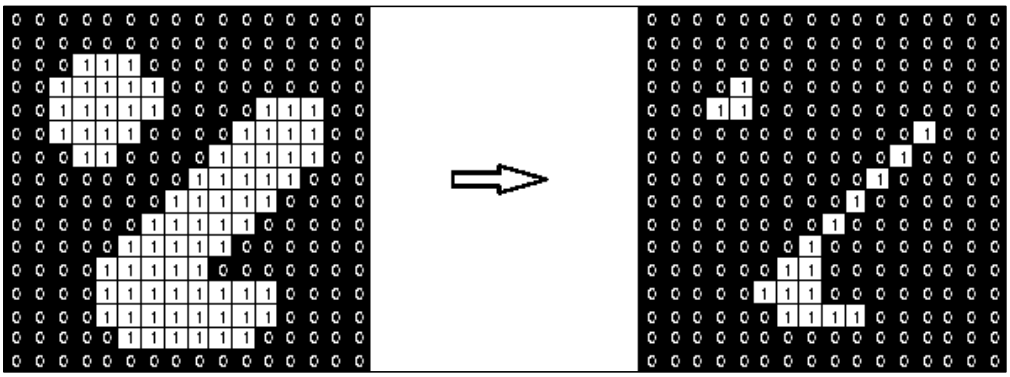


Şekil 3.1 Bir disk yapılandırma elemanı ile ikili bir görüntünün aşındırılması

X, E'nin bir alt kümesi olsun ve B'nin bu kümenin bir öğesini belirtmesine izin verelim. Denklem (3.1)'de morfolojik aşındırmanın tanımı verilmiştir.

$$\in_B (X) = X \ominus B = \{ x \backslash B_x \subset X \} \quad (3.1)$$

Özet olarak, ön yapı nesnesinin tamamen B elemanını içerebilen tüm pikselleri aşınmış nesnede olacaktır. Örneğin, morfolojik merkezi ile geometrik merkezi aynı olan bir 3x3 kare yapı elemanını ele alınmaktadır.



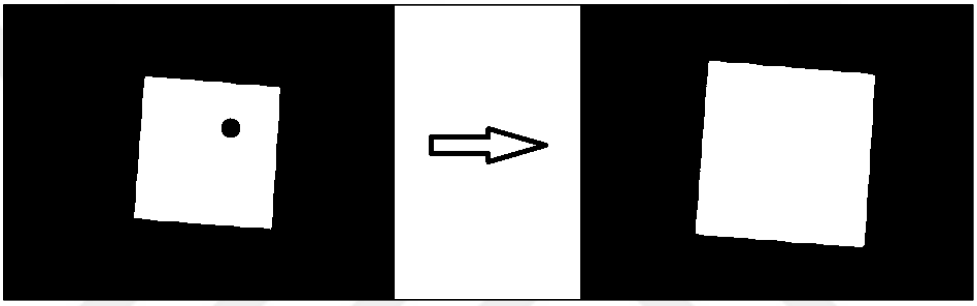
Şekil 3.2 3x3 kare yapı elemanı ile ikili görüntünün aşındırılması

İkili bir aşınmayı hesaplamak için, ön planın tüm pikselleri işleme alınmalıdır. Önplanın her bir pikseli için, algoritma yapı elemanını yapı elemanının merkezi piksel ile eşleştitrir, koyar ve yapı elemanının tamamen ön planda olup olmadığını test eder. Eğer değilse de, geçerli piksel arka plan olarak kabul edilecektir ve tam tersi durumda

mevcut piksel aşındırmaya uğramış ön planda yer alacaktır. Şekil 3.2'de 3x3 kare yapı elemanı ile ikili görüntünün aşındırılması gösterilmiştir.

3.3.2 Morfolojik Yayma Operasyonu

Aşındırma gibi yayma operasyonuda görüntü işlemede morfolojik temel operasyonlarından biridir. Aşındırmanın ardından görüntüye yayma operasyonu uygulanarak açma operasyonun zemini hazırlanır ve en önemli özelliği farklı ön plan bölgelerin sınırlarını genişletmektir. Şekil 3.3'de görüldüğü gibi bu ön plan nesneleri büyüyecek ve nesnelerdeki delikler daha küçük olacaktır fakat çok azı tamamen yok olacaktır.

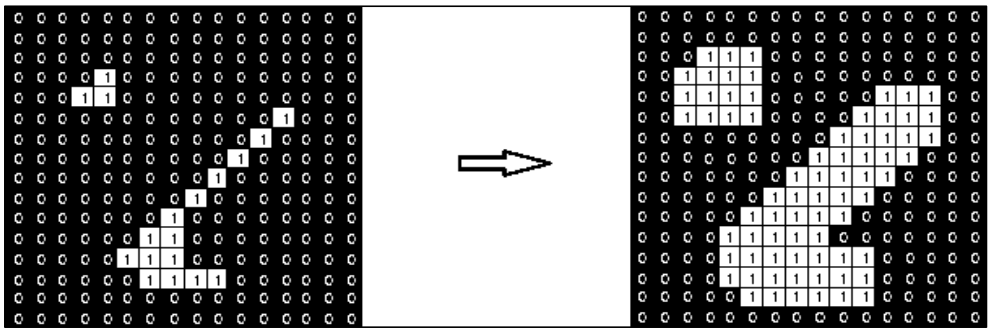


Şekil 3.3 İkili görüntünün yayılması

X, E'nin bir alt kümesi olsun ve B'nin bu kümenin bir öğesini belirtmesine izin verelim. Denklem (3.1)'de morfolojik yayma operasyonun tanımı verilmiştir.

$$\delta_B(X) = X \oplus B = \{ x + b \backslash b \in B, x \in X \} \quad (3.2)$$

Özet olarak, ön plan bölgelerine dokunabilen arka plandaki tüm pikseller, yapı elemanını B üzerine koyarak, genişlemiş nesnede yer alacaktır.

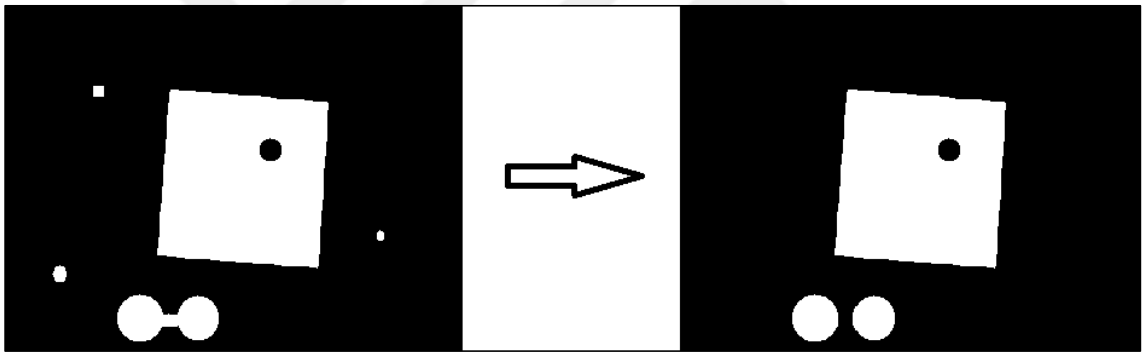


Şekil 3.4 3x3 kare yapı elemanı ile ikili bir görüntünün yayılması

Örneğin, Şekil 3.4'de de görüldüğü üzere morfolojik merkezi geometrik merkezle aynı olan 3x3 kare bir yapı elemanını ele alabiliriz. İkili görüntüde bir yayma hesaplamak için, arka planın tüm pikselleri işleme alınmaldır. Arka planın her bir pikselinde, algoritma yapı elemanını yapı elemanının merkezi piksel ile eşleştitrir, koyar ve yapı elemanının ön planlarını en az bir pikseliyle temas halinde olup olmadığını test eder. Öyleyse, geçerli bir piksel ön plan gibi kabul edilir ve tam tersi durumda eğer değilse, piksel bir arka plan pikseli olarak kalır.

3.3.3 Morfolojik Açma Operasyonu

Morfolojik açma işlemi, iki temel işlem olan aşındırma ve yayma operasyonlarının birleşimidir. Aslında aşındırılan görüntünün yayılmasıdır ve en temel özelliği çok fazla gürültüyü ortadan kaldırmasıdır. Bu işlem, küçük bir katmanla bağlantılı bölümleri kolayca ayıracaktır.

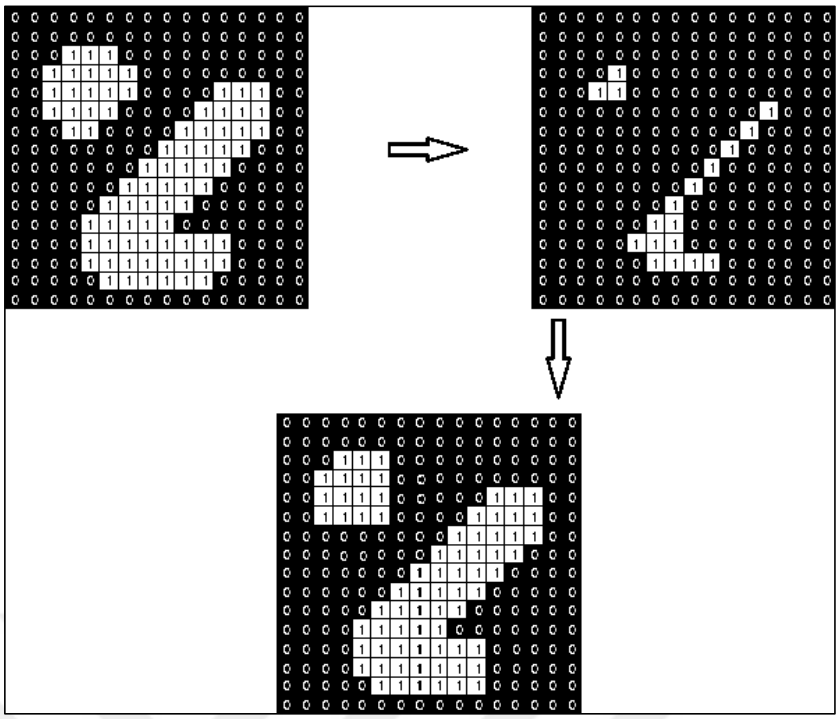


Şekil 3.5 İkili görüntünün açılması

X, E'nin bir alt kümesi olsun ve B'nin bu kümenin bir öğesini belirtmesine izin verelim. Denklem (3.3)'de morfolojik açma operasyonun tanımı verilmiştir.

$$\emptyset_B(X) = \delta_B(\in_B (X)) = (X \ominus B) \oplus B \tag{3.3}$$

Taslak halinde yapılandırma elemanı ile kapsanabilen bütün pikseller, tamamen ön plan bölge içerisinde olacak şekilde korunacaktır. Bununla birlikte, ön plan bölgesinden uzaklaşan bir parçası olmayan yapılandırma elemanı tarafından erişilemeyen tüm ön pikseller aşınacaktır. Aşağıdaki Şekil 3.6'da 3x3 kare yapılandırma elemanı ile bir açma operasyonu örneği gösterilmiştir. Uygulamalarda genişletme ve aşındırma çok çeşitli varyasyonlarda kullanılmaktadır.



Şekil 3.6 3x3 kare yapı elemanı ile ikili bir görüntünün açılması

3.4 Yaya Tespiti

Literatürde birçok yaya tespit tekniği önerilmiştir. Bu tekniklerden bazıları, bir dizi çerçeveden elde edilen zamansal verileri kullanarak yayaları tespit ederken, diğer yaklaşımlar ise tek bir görüntülerde yayaların tanınması konusundaki daha zor problemleri çözmeye çalışmaktadır. Diğer bölümlerde anlatıldığı gibi bu tür teknikleri uygularken birçok zorluk ortaya çıkabilmektedir. Bunlardan bazılarının çevresel koşullar, nesneler arasındaki yoğunluk farklılıklarının eksikliği, belirli bir sahnede insanların yönelimi, pozu, büyüklüğü ve görünüşü gibi olası varyasyonlar olduğunu hatırlayalım. Bu sorunların üstesinden gelmek için, birçok teknik bu olasılıklardan bazılarını görmezden gelir. Örneğin, varsayımlar yapılarak yayaların tanınması [46]; sahnedeki tüm hareketli nesnelerin yaya olarak kabul edilmesi [34]; yayaların sahneye birer birer girmesi varsayımı [47]; yayaların ayaklarının daima görünmesi [48] gibi kabullerdir.

Bu bölümde iki boyutlu tekniklere dayalı tek bir çerçevede yaya algılama yaklaşımlarından bazıları anlatacaktır. Bunlardan ilki olan ön plan imge bölgeleri (foreground blobs) yönteminde, imge bölgeleri arka plan çıkarma işleminden elde edilir ve analitik teknikler kullanılarak incelenir. Bir diğer yöntem olan düşük seviye

öznitelikler yönteminde ise düşük seviyeli özniteliklerin çıkarıldığı ve bir yayanın varlığını belirlemek için desen sınıflandırma teknikleri uygulandığı görülmektedir. Son olarak birleştirilmiş kanal öznitelikleri yönteminden bahsedilecektir. Bu yöntemde algılama çerçevesi kavramsal olarak basittir. ACF detektörü hızlı ve etkili bir kayar pencere detektörüdür.

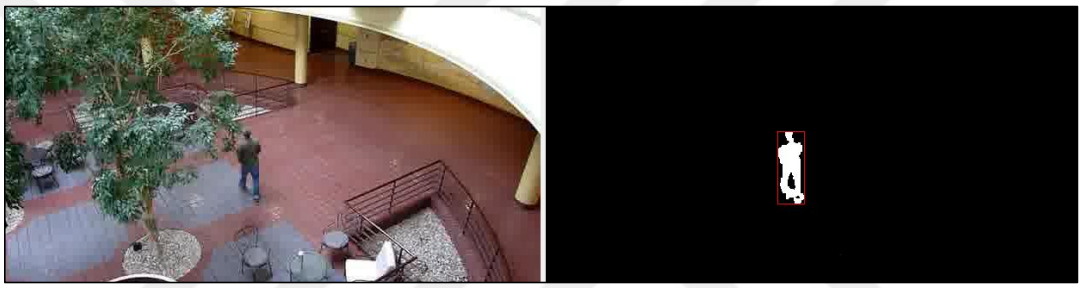
3.4.1 Ön Plan İmge Bölgeleri Yöntemi

[34, 46] gibi birinci kuşak detektörler, arka plan çıkarma işleminden sonra her bir ön plan imge bölgesinin ayrı bir yayayı temsil etmesi de dâhil olmak üzere, olaydaki nesnelerin doğasıyla ilgili çeşitli varsayımlarda bulunur. Bu varsayımlar, kısıtlanmamış durumlarda hataya eğilimli durumların oluşmasına yol açar ve arka plan çıkarma hataları nedeniyle bu durum kişi imge bölgesi ilişkisini bozmaktadır. Bu varsayımlar [49, 50, 40, 43] gibi diğer tekniklerde rahatlatılmıştır. Burada imge bölgeleri önceki karelerden gelen izleme bilgileri aracılığıyla bir veya daha fazla kişi ile ilişkilendirilmiştir. Bu yaklaşımlarda, mevcut nesnelerin herhangi biriyle yeterince örtüşmeyen bir ön plan bölge tespit edilirse, yeni ve takip edilmeyen bir kişiye ait olduğu varsayılır. Ancak bu tekniklerin bu tezin ele alınmasını istediği real-time senaryolarına uygulanamayacağı, birden fazla yayaların aynı anda sahneye girebildiği ve tıkayabildiği aşikârdır.

Bu teknik ile ilgili önemli bir problem, yayaların tek tek arka plan çıkarma işleminden sonra ayrı bloklara ayrılacağı varsayılmaktadır. Buda sistemde tıkanıklığın asla oluşmaması gerektiğini belirten içsel bir kısıtlama olduğu anlamına gelir. Bu gerçek dünyadaki senaryolarda elbette mümkün değildir. Bazı makalelerde bu tıkanıklığı azaltmak için, görüntüleme kamerasının tepeden bakan bir açı ile monte etmek ve yere doğru tutturmak önerilmiştir. Arka plan çıkarma teknikleri, arka plandaki nesneleri bölmek için uygulanır ve ön plan pikselleri, bağlı bileşenler kullanılarak iki boyutlu imge bölgelerine kümelenir. Bu bölgeler daha eğik kamera görüntüleriyle karşılaştırıldığında birden fazla yaya içermesi daha az olası bir durumdur. Buna ek olarak, genel bakış açısı kamera konumlandırması nedeniyle, yayaların boyutunun nispeten tutarlı olması gibi ilave avantajlara sahiptir. Bu yöntem yaya takip ve sayma amaçları için kullanılabilir. Bu bilgiyi kullanarak, iki veya daha fazla kişi tek bir bloğa ayrılmış olsa bile, belirli bir üstten

görünümden bir kişinin pikselindeki ortalama alan bilgisini kullanarak, belirli bir bölgede kaç kişinin yer aldığını kolayca bulabiliriz.

Bu yöntemin avantajları kadar dezavantajları da bulunmaktadır. Öncelikle, kamera açısı genellikle tepeden bakan bir yerleştirme ile kullanıldığından dolayı iç mekân senaryolarına uygulanabilir. Bu bakış açısına göre bir kamera, maksimum yükseklik bir tavan tarafından kısıtlandığı için, genellikle sınırlı bir görüş alanına sahiptir. Bu küçültme oranını azaltmak için bu teknikle çalışan sistemler genellikle görüş alanını genişletmek için daha geniş açılı lensler kullanmaktadır. Bu nedenle, izlenecek alan küçük ve içeride olmadıkça, bu bakış açısına sahip kameraların maalesef avantajı yoktur.



Şekil 3.7 İmge bölgeleri analizinden sonra ön plan maske çıkarımı

Şekil 3.7'de imge bölgeleri (blob) analizinden sonra ön plan maskesinin görselleştirilmiş bir görüntü örneği yer almaktadır.

3.4.2 Düşük Seviye Özniteliklerin Çıkarılması Yöntemi

Başka bir yaklaşım olan düşük seviye öznitelik çıkarımı (low-level features) yöntemi, çeşitli çözünürlüklerdeki pencerelerin farklı çözünürlüklerde kaydırılması, düşük seviyeli özelliklerin çıkarılması ve bir yaya varlığını belirlemek için desen sınıflandırma teknikleri kullanılmasıdır [51]. İstatistiksel şekil modellerinde olduğu gibi, bu tür teknikler çeşitli çözünürlüklerde birden çok pencere kümesinin incelendiği, bir yaya veri kümesinden eğitilir. Tipik olarak aydınlatma değişiminin değişmez olması için tasarlanan özellikler daha sonra bu pencerelerden ayıklanır ve toplanır. SVM'de olduğu gibi yaya sınıfını tanımlayan en iyi öznitelikler bir sınıflandırıcıyı eğitmek için seçilir ve kullanılır. Bu teknikler genellikle hareket bölütleme teknikleri uygulanmadan direkt olarak uygulanabilir. Bununla birlikte, bazı teknikler hatalı pozitiflerin giderilmesi için

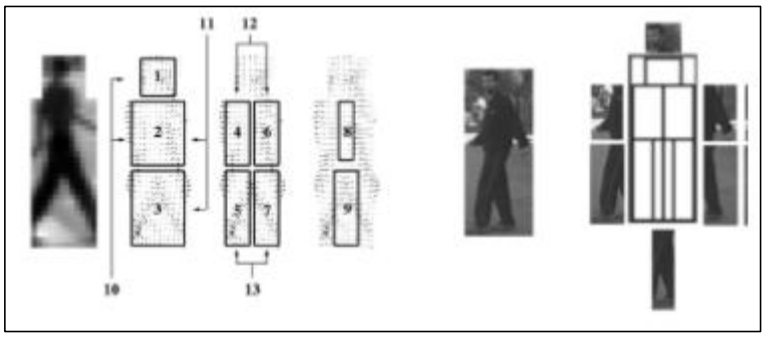
bir adım sonrasında arka plan çıkarma işlemini kullanır. Bu tür teknikler literatürde Haar dalgacıkları yöntemi, normalleştirilmiş Histogram of Oriented Gradient (HOG) tanımlayıcıları yöntemi ve kenar yoğunluk genliği özellikleri yöntemi gibi çeşitli özellikler kullanılarak önerilmektedir.

Sınıflandırıcının eğitimi tamamlandıktan sonra sistem, 128x64 piksel boyutlarındaki pencereleri giriş görüntüsüne kaydırarak, keyfi konumlarda yayaları algılayabilir ve böylece tüm olası yerleri tarayabilir. Her bölgede, yaya sınıfının özelliklerinden manuel olarak seçilen bazı özellikler, pencereden çıkarılmaktadır. Bu özellikler daha sonra yaya olan ve yaya olmayan görüntülerle SVM'i eğitmek için kullanılmaktadır. Ancak, bu teknik bir giriş görüntüsünde yayaları aramak için kullanırken, son derece dikkatli bir arama işlemi yapılmalıdır çünkü birden fazla yayanın bu aramaya yanıt vermesi muhtemeldir.

Belirli bir öğrenme algoritmasının genel doğruluğunu artırıcı olarak bir yöntem uygulanarak geliştirilebilir. Bu genellikle Adaptive Boosting (AdaBoost) algoritması uygulanarak elde edilmektedir. Bu yöntem, yaklaşım olarak birçok yanlış veya zayıf sınıflandırıcının bir araya getirilmesiyle yüksek doğrulukta veya güçlü bir sınıflandırıcının üretilebileceği inancına dayanır. AdaBoost'un ana fikri, bir eğitim setinin her bir örneğini tekrarlı bir çerçevede eğiterek yeniden bir ağırlık tayin etmektir. Başlangıçta her eğitim seti örneğinin ağırlıkları eşittir. Algoritma daha sonra, eğitim örneklerinin sınıflandırmasına göre hatayı en aza indiren zayıf sınıflandırıcıyı seçer. Bu zayıf sınıflandırıcı için, zayıf bir sınıflandırıcı tarafından yanlış sınıflandırılan her bir eğitim örneğinin ağırlıkları arttırılır. Bu nedenle, algoritma kalan zayıf sınıflandırıcıları test ederken, önceki zayıf sınıflandırıcının gözden kaçırdığı örnekleri daha iyi tanımlayan bir sınıflandırıcı seçecektir. Bu işlem eğitim setinin daha zor örnekler karşısında algoritmanın daha fazla odaklanmasını sağlar. Daha önce seçilen ve ağırlıklı zayıf sınıflandırıcılara ek olarak, eğitim örneklerinin sınıflandırmasına göre hatayı en aza indiren bir sonraki en iyi zayıf sınıflandırıcının belirlendiği bu teknik tekrar tekrar yapılmaktadır. Bu tekrarlama işlemi tüm zayıf sınıflandırıcılar kullanıldığında ya da kalan en iyi zayıf sınıflandırıcının hatası belirli bir eşik değerden az olana kadar devam etmektedir. Bu yaklaşımda, her bir zayıf sınıflandırıcının ağırlıkları, daha düşük sınıflandırma hatası elde edilene kadar uygulanır ve eğitim örneklerden daha iyi ağırlığa

sahip olan yeni ağırlık bir çoğunluk oyu ile belirlenir. Bu nedenle, AdaBoost bu tekniği kullanarak hem özellik setini seçmek hem de sınıflandırıcıyı eğitmek için kullanılabilir. AdaBoost, uygulamalarda dâhil olmak üzere yaya algılama algoritmalarında kullanılan bir dizi sınıflandırıcıyı eğitmek için kullanılmıştır.

AdaBoost'u kullanarak eğitim aşamasındaki iyileşmeler ve bu yaklaşımların genel karmaşıklığı, zayıf sınıflandırıcıları kademeli katmanlardan oluşan bir tabaka halinde düzenleyerek azaltılabilir. Bunu saptama işleminin genel biçimi, artan karmaşıklığa sahip çok sayıda sınıflandırıcının katmanındakiyle aynıdır. Basit bir birinci sınıflandırıcıdan gelen olumlu bir sonuç, ikinci bir pozitif sonucun üçüncü bir sınıflandırıcının sonucu ve daha karmaşık bir sınıflandırıcının değerlendirmesini tetikler. Herhangi bir noktada olumsuz bir sonuç alınırsa alt pencerenin hemen reddedilmesine yol açar. Bu nedenle, daha düşük hatalı pozitif oranlar elde etmek için daha karmaşık sınıflandırıcılar çağrılmadan önce, daha basit sınıflandırıcılar alt pencerelerin çoğunluğunu elemek için kullanılır. Bu teknik [7]'de uygulanmakta olup, burada yaya tespiti için bir sınıflandırıcı oluşturulmaktadır ve AdaBoost kaskat düğümlerini eğitmek için kullanılmaktadır. Tasarlanan sistem, sadece düşük çözünürlükte net olarak görünen insan figürleri üzerinde eğitilmiştir ve tıkanmış veya kısmi insan figürlerini tespit etmemektedir.



Şekil 3.8 Hiyerarşik öznitelikler alt pencereler

Benzer bir yaklaşım 3.8'de de uygulanmaktadır. Buradaki önemli fark, arama penceresinin şekilde gösterildiği gibi 9 alt pencereye ayrılması ve HOG tanımlayıcıları kullanılarak eğitilmesidir. Her alt pencere ve dört çift kombinasyon daha sonra hücrelerin içine ayrılır ve işlenir. Her bir yerel bölgeyi bağımsız olarak sınıflandırmak için ayrı bir SVM kullanılır. Her bir ayrı sınıflandırıcının sonuçları daha sonra pencereyi

yaya var ya da yok olarak sınıflandıran nihai bir SVM ile beslenir. Pencerenin bölünmesinin ardındaki fikir, bir SVM'nin sınıflandırılması için bir yaya pozunda çok fazla değişkenliğe sahip parametrenin olmasıdır. Alt pencereler bu nedenle, kümeleri nispeten basit bileşen sınıflandırıcıları tarafından yakalanabilen yönetilebilir parçalara ayırmak için kullanılır. Bir diğer çeşitte ise 6 alt bölgeyi kullanarak benzer bir yaklaşım üstlenilmiştir. Ancak şekil 3.8'de görüldüğü gibi Haar dalgacıkları özellikleri, gradyan temelli özellikler, yoğunluk özellikleri ve doku özellikleri dahil olmak üzere, her bölge için 7 farklı özellik incelenir. Her bölge için en iyi performans özellikleri, daha sonra hiyerarşik SVM sınıflandırması için kullanılmaktadır.

Bu tekniklerin hepsi, özellikle yayaların kameraya yakın olduğu durumlarda, değişen ölçeğe sahip kişileri tespit edememe gibi çeşitli problemlerden dolayı başarılı bir şekilde çalışamamaktadır. Buna ek olarak, SVM pencerede ortalanmış bir yayayı tespit etmek için eğitildiğinden, pozisyonları tam olarak ortalanmış değilse yanlış sınıflandırma meydana gelebilir. Son olarak, SVM sınıflandırıcılarının eğitimi sorunlu olabilir. Pozitif ve negatif örnekler arasındaki oran uygun bir değere ayarlanmalıdır. Eğitim setindeki çok büyük sayıda pozitif örnek, sınıflandırma sırasında yüksek oranda hatalı pozitif tespitlere yol açabilir. Tam tersine, çok sayıda negatif örnek ise hatalı öğrenme üretmektedir. Bu durum eğitim veri setinin büyüklüğünü, hem pozitif hem de negatif örneklerin kalitesi, test verilerinden elde edilen sınıflandırma sonuçları belirlemede büyük bir etkiye sahip olabilir. Negatif örnekler belirsiz nesneleri hesaba katmalı, pozitif örnekler ise çeşitli yayaların pozlarını, yönelimlerini ve görünüşlerini daha iyi temsil etmelidir. Son olarak, test setinin içeriği, aydınlatma koşulları, ortam değişkenlikleri, pozitif/negatif örneklerin oranı, negatif örneklerin kalitesi açısından eğitim setlerininkiyle çok farklı özelliklere sahip olmamalı benzer özellikler içermelidir.



Şekil 3.9 HOG Tanımlayıcısı için kullanılan hücre ve blok yapısı

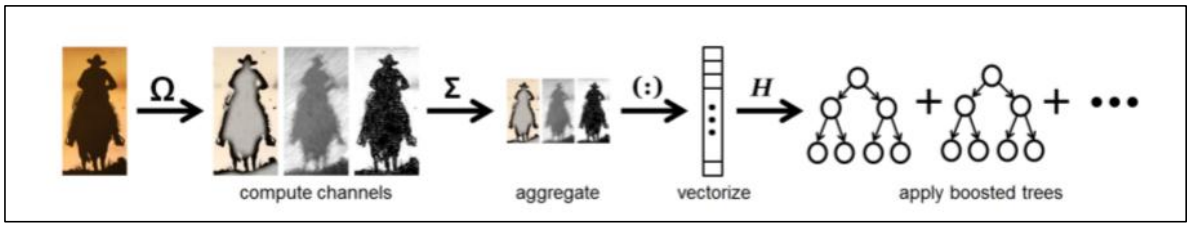
Bu benzerlikteki her teknik, SVM'lerin kullanılmasını gerektirmez. Şekil 3.9'da düşük seviyeli öznitelikleri kullanılarak tespit edilen bir yaya gösterilmektedir.

3.4.3 Birleştirilmiş Kanal Öznitelikleri Yöntemi

Birleştirilmiş kanal öznitelikleri 2014 yılında yayınlanan oldukça yeni bir detektördür [52]. Yeni önerilen bir yöntem olmasına rağmen uygulamalarda sıkça tercih edilmeye başlanmıştır. ACF detektörü, ICF'ye dayanmaktadır ve bu nedenle kavramsal olarak çoğu açıdan aynıdır. Bu metodu basit ve kavramsal olarak özetlemek gerekirse Şekil 3.10'da gösterildiği gibi çeşitli aşamalardan oluşur. İlk kanallar giriş görüntüsünden denklem (3.4)'te belirtilen formülle hesaplanır. Bu denklemde C yeni kanal görüntüsü, Ω düşük seviye kayan değişmez fonksiyonu, I ise kanaldan alınan görüntüdür.

$$C = \Omega\,(I) \tag{3.4}$$

Ayrıca, C kanallarındaki her bir piksel bloğu toplanır ve düzeltilir, böylece daha küçük çözünürlüklü kanallar ortaya çıkar. Daha küçük kanallar daha sonra vektör formuna ve öznitelikler ise birleştirilmiş kanal özniteliklerinde tek bir arama pikseline dönüştüler. Son olarak, bu öznitelikleri karar ağaçlarını eğitmek için bir hızlandırma sürecinde kullanılmaktadır. Oluşturulan modeli kullanarak bu bölgeler işaretlenir ve çıkarılır. Detektör, çok ölçekli bir kayar pencere tekniği ile çalışır.

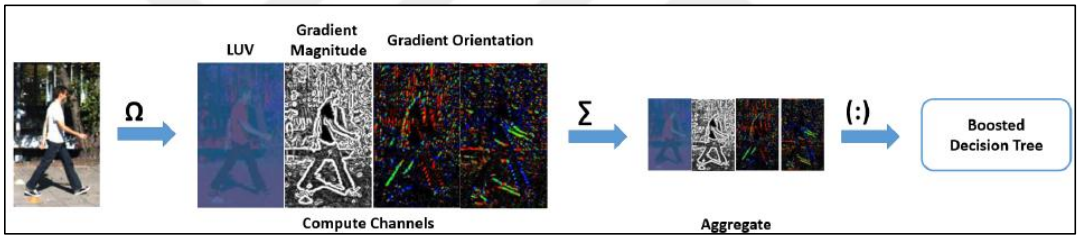


Şekil 3.10 ACF dedektörünün süreçlerin görselleştirilmesi [12]

ACF, üç tip kanalı kullanır: normalize edilmiş gradyan genliği, yönlendirilmiş gradyan histogramı (HOG) ve LUV renk kanallarıdır. Normalize edilmiş gradyan genlik kanalı, görüntünün gradyan kuvvetlerini içerdiğinden, büyük oranda bir bilgi sağlamaktadır. Gradyanlar bir görüntüde çok önemli bilgilerdir, çünkü gradyan noktaları en yoğun şiddetteki değişim yönündedir.

Böylece bu üç özellik kanalının bilgisinden yaralanarak kanallar hesaplanmadan önce, bir Gauss filtresiyle düzeltilmektedir. İşlenmiş bu kanal bilgileri toplanır ve daha düşük

çözünürlükte ACF'ler üretmek için bir Gaussian yumuşatma filtresinden Şekil 3.10'da ∑ ile gösterildiği gibi geçilir. Oluşturulan ACF'ler tek pikselli aramalar tablosu olarak kullanılır. BDF eğitimi için kullanılan bir özellik vektörü üretmek için kanal özellikleri vektör formuna Şekil 3.10'da (:) ile gösterildiği gibidir. Farklı ölçeklerdeki nesneleri tespit etmek için kanal özellikleri piramidi üzerinde çok ölçekli bir kayar pencere yaklaşımı uygulanır. Özellik piramitleri, giriş görüntüsünü ölçeklemek ve her bir ölçeğin kanal özelliklerini ayrı ayrı hesaplamak yerine, ara ölçeklerin kanal özelliklerine yaklaşmak suretiyle daha verimli bir şekilde hesaplanır. 2,048 derinlikteki iki karar ağacını kullanır ve her 128x64 pencerede 128x64x10/16 işlemin sonucundan yaklaşık 5 adet adayın özelliğini kullanarak AdaBoost ile eğitim yaparak birleştirir. Model boyutu, kanal özellikleri piramidinin farklı seviyelerinde kullanılan sürgülü pencerenin boyutunu belirtir. Karar ağaçlarındaki bölünmüş düğümler ise özellik değerleri ve öğrenilen eşikler arasındaki basit karşılaştırmalardır.



Şekil 3.11 ACF dedektörü algoritması

ACF dedektörü algoritması akış diyagramı Şekil 3.11'de gösterilmiştir. Genellikle, özellik vektörünün sadece bir alt kümesi öğrenilen karar ağacı tarafından kullanılır. AdaBoost, her ikisini de gerçekleştirir fakat özellik seçimi ve bölünmüş düğümlerdeki eşikleri öğrenmektedir. Tam bir görüntü algılama çerçevesi kurmak için birden çok ölçek üzerinde uygulanan daha önce belirtilen sürgülü pencere yaklaşımını kullanır.

Video işaretlerinde yaya tespiti için kullanılan PETS, sıklıkla tercih edilen ve literatürde çoğunlukla referans alınan bir veri setidir. PETS veri setinin farklı görüntü işleme tabanlı araştırmalar için birtakım veri kümeleri vardır. Her yıl, PETS belirli bir hedefle sabit veri kümeleri üzerinde bir değerlendirme yapmaktadır. Şekil 3.12'de Matlab ortamında PETS veri seti üzerinde yapılan yaya tespit çalışmasından bir görüntü verilmiştir.

İnsan tespiti için AFC dedektörü tercih edilmiştir ve oldukça başarılı sonuçlar alınmıştır. Sistemin çalışma performansı ve başarım oranı ise tatmin edici düzeydedir.



Şekil 3.12 Video görüntüsü üzerinde Matlab ile yaya tespiti

BÖLÜM 4

SONUÇ VE ÖNERİLER

Bu çalışmada video ve görüntü işaretleri üzerinde aranan yayaları tespit edebilen genel bir yaya tespit sistemi tasarlanmıştır ve tasarlanan bu sistem video işareleri içerisindeki insan sayısını belirlemek için kullanılmıştır. Çalışmanın ikinci kısmında durağan görüntülerde yayaların nasıl tespit edileceği üzerinde durulmuştur. Tanımlayıcı olarak kullanılan HOG yönteminin aşamaları ile SVM, KNN ve Adaboost sınıflandırma algoritmaları anlatılmıştır. Son olarak görüntüdeki insanları geliştirilen yeni yöntem ile tespit edebilen algoritma sunulmuştur.

Üçüncü bölümde yayanın video işaretleri üzerinde tespiti ve takibi için kullanılan yöntemlerden, literatürde yapılan çalışmalardan, yaya tespit uygulamaları ve yaya tespiti sırasında karşılaşılan zorluklardan bahsedilmiştir. Daha sonra video işaretlerinde yaya tespitinin ilk adımı olan hareket kestirimi ve morfolojik görüntü işleme operasyonları açıklanmıştır. Yaya tespit yöntemlerinden olan ön plan imge bölgeleri yöntemi, düşük seviye özniteliklerin çıkarılması yöntemi ve birleştirilmiş kanal öznitelikleri yöntemi ele alınmıştır.

4.1 Önerilen Yaya Tespit Sistemi ve Uygulaması

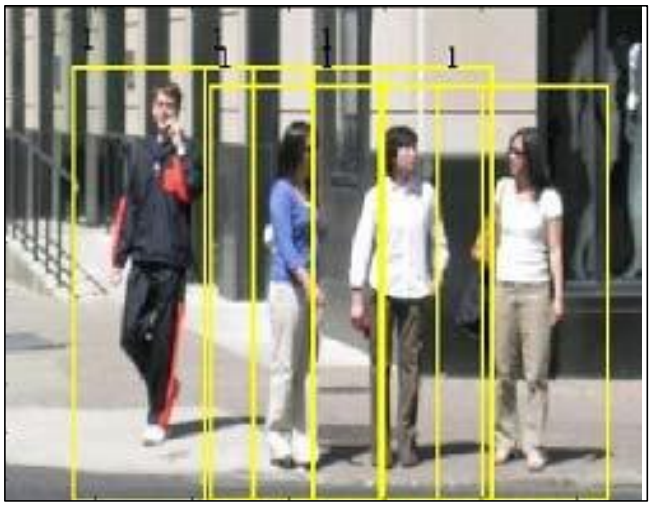
Bu tez çalışmasında yapılan tüm uygulamalar Matlab ortamında geliştirilmiş ve test edilmiştir. INRIA veri kümesi, test veri tabanında bulunan görüntüler içinde en çok bir yaya yer alacak şekilde kesilmiş olduğundan net sonuçlar elde edecek bir test ortamı oluşturmakta yeterli değildir. Bu nedenle test aşamasında Penn-Fudan veri seti [14] kullanılmıştır. Veri setinde toplam 423 yaya bulunan 170 adet görüntü bulunmaktadır. Her görüntüde ise en azından bir adet yaya bulunmaktadır. Görüntüde bulunan

yayaların boyutları 180x390 pikseli geçmemekte ve yayalar ayakta yürümekte ya da sabit durmaktadırlar. Bu veri setinden alınmış olan örnek bir görüntü üzerinde farklı sınıflandırıcı algoritmaları kullanılarak tespit edilen yayalar Şekil 4.1'de gösterilmiştir. Tüm sınıflandırıcılar aynı eğitim veri seti ile eğitilmiş, daha sonra aynı test verisi ile test edilmiştir. Nihai karar alınırken aynı görüntü üzerinde tüm sınıflandırıcıların verdiği sonuçlar kıyaslanarak en azından iki sınıflandırıcının yaya olarak tespit ettiği bölgeler yayanın bulunduğu yer olarak belirlenmiştir.



Şekil 4.1 SVM, Adaboost, KNN sınıflandırıcılarla elde edilen sonuçlar [55]

Örnek görüntü üzerinde SVM ve Adaboost ile 4 tane, KNN ile 5 tane yaya işaretlenmiştir. Bu örnek görüntü üzerinde, SVM ve Adaboost ile yaya sayısı doğru tahmin edilmesine karşılık KNN ile görüntü üzerinde solda bulunan merdiven parmaklıkları da yaya olarak tespit edilmiştir. Önerilen yöntemki algoritma kullanılan üç sınıflandırıcıların sonuçlarına da bakıp hepsinin ortak kararına göre yaya sayısı tahmininde bulunduğu için KNN ile oluşan hatalı pozitif sonuç giderilmiştir. Bu önerilen yöntem sayesinde Şekil 4.2'deki görüldüğü gibi doğru bir şekilde dört tane yaya tespit edilmiştir.



Şekil 4.2 Önerilen yöntem ile yaya tespiti [55]

Penn-Fudan veri seti üzerinden seçilen 50 adet test görüntüsü için örnek görüntüye yapılan işlem adımları sırasıyla uygulanmıştır. Bu uygulamanın sonucunda elde edilen sonuçlar Çizelge 4.1'de gösterilmiştir. Seçilen test veri setinde 195 adet ayakta duran veya yürüyen yaya bulunmaktadır.

Yaya tespit sistemlerinde performans değerlendirmesi yapılırken aşağıdaki parametreler kullanılır [53]:

- Doğru pozitif: Görüntüde doğru şekilde tespit edilen yaya sayıdır.
- Hatalı negatif: Görüntüde olmasına karşılık tespit edilemeyen yaya sayısıdır.
- Hatalı pozitif: Görüntüde yaya olmamasına karşılık yaya olarak tespit edilen pencere sayısıdır.
- Doğru negatif: Görüntüde incelenen pencerede yaya bulunmazken yayanın yok olarak tespit edildiği pencere sayısıdır.

Iskalama oranı, pozitif örneklerin kaç tanesinin kullanılan sınıflandırma yöntemiyle negatif olarak tespit edildiğini ifade etmektedir [54]. Hatalı pozitif oranı ise, negatif örneklerin kaç tanesinin sınıflandırma yöntemiyle pozitif olarak etiketlendiğini ifade etmektedir. Son olarak kıyaslama oluşturması için çoğunluk oylamasının yanı sıra her üç sınıflandırıcının da aynı anda yaya olarak bulduğu çerçevelenmiş pencereler pozitif, diğerleri negatif olarak işaretlenerek bir test çalışması yapılmıştır ve elde edilen bu sonuçlar da Çizelge 4.1 ve 4.2'de gösterilmiştir. Bu yönteme ortak karar sınıflandırması adı verilmiştir. Denklem (4.1) ve (4.2)'de ise ıskalama ve hatalı pozitif oranı formülize edilmiştir.

$$Iskalama\ Oranı\ = \frac{Hatalı\ Negatif}{Hatalı\ Negatif + Doğru\ Pozitif} \tag{4.1}$$

$$Hatalı\ Pozitif\ Oranı\ = \frac{Hatalı\ Pozitif}{Hatalı\ Pozitif + Doğru\ Negaif} \tag{4.2}$$

Iskalama oranı ve hatalı pozitif oranı arasında tam olarak olmasa da yaklaşık bir şekilde doğrusal bir ters orantı olduğu göz önüne alınırsa en iyi sonuç veren yöntemin beklendiği gibi daha başarılı bir sınıflandırıcı olan çoğunluk oylamasına dayanan geliştirelen algoritma olduğu sonucu çıkarılabilir.

Çizelge 4.1 SVM, Adaboost, KNN ve önerilen yöntemin sonuçları

Yöntem	Doğru Pozitif	Hatalı Negatif	Hatalı Pozitif	Doğru Negatif
SVM	132	63	6	77
Adaboost	141	54	21	62
KNN	129	58	14	69
Ortak karar	127	68	2	81
Önerilen Yöntem	149	46	2	81

Çizelge 4.2 SVM, Adaboost, KNN ve önerilen yöntemin başarımları

Yöntemler	Iskalama Oranı	Hatalı Pozitif Oranı
SVM	0,32307	0,07228
Adaboost	0,27692	0,25301
KNN	0,29743	0,16867
Ortak Karar	0,34871	0,02409
Önerilen Yöntem	0,23589	0,02409

Literatürde yapılan başka bir çalışmada ise öznitelik çıkarmak için HOG, sınıflandırıcı olarak ise SVM ve KNN sınıflandırıcıları birleştirelerek görüntüler sınıflandırılmıştır [13]. Amsterdam Library of Object Images (ALOI) veri seti kullanarak 100 tane görüntü üzerinde yaya tespiti tepsiti yapılmıştır elde edilen sonuçlar Çizelge 4.3'de verilmiştir.

Çizelge 4.3 ALOI veri setinde yaya tanıma oranları [13]

Yöntemler	Tanıma Oranı
SVM	%26,83
KNN	%61,10
SVM-KNN	%58,27

Çizelge 4.3'de gösterilen sonuçlarla bu tez çalışmasında geliştirilen algoritma kıyaslandığında çoğunluk oylamasına dayanan yaya tespit algoritmasının daha başırılı olduğu açıkça görülmektedir.

4.2 Sonuçlar

Bu çalışmada farklı sınıflandırıcılar birleştirilerek görüntüde yaya tespiti yaparken hatalı pozitif oranını düşüren çoğunluk oylamasına dayanan özel bir yaya tespit algoritması geliştirilmiştir. Çizelge 4.1 ve 4.2'de özetlenen sonuçlar göz önüne alındığında önerilen yöntemin doğru pozitif oranını yükseltirken hatalı pozitif oranını küçülttüğü açıkca görülmektedir. Buradan yola çıkarak çoğunluk oylaması yönteminin sunulan problemin çözümü için daha başarılı olduğu söylenebilir. Üç sınıflandırıcının ortak kararı alındığında hatalı pozitif sayısının minimize edildiği görülmektedir ancak bu durumda doğru pozitif oranı da küçülmektedir. Problemin yapısı göz önüne alındığında ortaya çıkan bu durumun istenmeyen bir sonuç olduğu aşikârdır [55].

Gelecekte yapılacak çalışmalarda üç sınıflandırıcı ayrı ayrı eğitip öznitelikleri sınıflandırmak yerine bu üç sınıflandırıcının özelliklerini içeren yeni bir öznitelik sınıflandırma yönteminin bulunması amaçlanmaktadır. Ayrıca görüntüdeki yayaların özelliklerini daha iyi gösterecek yeni özniteliklerin oluşturulması da önemli bir adım olarak düşünülmektedir.

KAYNAKLAR

[1] Tsukiyama, T. ve Shirai, Y., (1985). "Detection of the movements of persons from a sparse sequence of TV images", Pattern Recognition, 18: 207-213.

[2] Oren, M., Sinha P., Osuna E. ve Poggio T., (1997). "Pedestrian detection using wavelet templates", Computer Vision and Pattern Recognition, 17-19 Haziran 1997, San Juan.

[3] Gavrila, D.M. ve Philomin, V., (1999). "Real-time object detection for “smart” vehicles ", The Proceedings of the Seventh IEEE International Conference on Computer Vision, 20-27 Eylül 1999, Kerkyra.

[4] Papageorgiou, C. ve Poggio, P., (2000). "A trainable system for object detection", International Journal of Computer Vision, 38: 15-33.

[5] Ioffe, S., ve Forsyth D.,(2001). “Probabilistic methods for finding people”, International Journal of Computer Vision, 43: 45-68.

[6] Viola, P. ve Jones, M., (2001). “Rapid object detection using a boosted cascade of simple features.”, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1:511-518.

[7] Viola, P., Jones, M.J. ve Snow, D., (2003). "Detecting pedestrians using patterns of motion and appearance", The Proceedings of the Ninth IEEE International Conference on Computer Vision, 13-16 Kasım 2003 Nice, France.

[8] Freeman, W. T. ve Roth, M., (1995). "Orientation histograms for hand gesture recognition", International Workshop on Automatic Face and Gesture Recognition, 12:296-301.

[9] Lowe, D.G., (2004). "Distinctive image features from scale-invariant keypoints", International journal of computer vision, 60(2):91-110.

[10] Dalal, N. ve Triggs, B., (2005). "Histograms of Oriented Gradients for Human Detection", IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 25-25 Haziran 2005 San Diego.

[11] Sabzmeydani, P. ve Mori, G., (2007). "Detecting pedestrians by learning shapelet features", IEEE Conference on Computer Vision and Pattern Recognition, 1:1-8.

[12] Dollar, P., Appel, R., Belongie, S. ve Perona, P., (2014). "Fast Feature Pyramids for Object Detection", IEEE Transactions on Pattern Analysis and Machine Intelligence, 8:1532–1545.

[13] Cai, Z., Yu P., Liang Y. ve Lin B., (2016). "SVM-KNN Algorithm for Image Classification Based on Enhanced HOG Feature", The 4th IIAE International Conference on Intelligent Systems and Image Processing, 8-12 Eylül 2016, Kyoto.

[14] Wang, L., Shi, J., Song, G. ve Shen, I., (2007). "Object Detection Combining Recognition and Segmentation", Proceedings of 8th Asian Conference on Computer Vision (ACCV), 1:189-199.

[15] Dalal, N., ve Triggs, B., (2005). "Histograms of oriented gradients for human detection", IEEE Computer Society Conference in Computer Vision and Pattern Recognition, 1:886–893.

[16] Fierro-Radilla, A. ve Perez-Daniel, K., (2014). "An effective visual descriptor based on color and shape features for image retrieval", Human-Inspired Computing and Its Applications, 8856:336–348.

[17] Suard, F. ve Rakotomamonjy, A., (2006). "A. Bensrhair, and A. Broggi. Pedestrian detection using infrared images and histograms of oriented gradients", IEEE Intelligent Vehicles Symposium,206–212.

[18] Dalal, N., (2006). Finding People in Images and Videos, Phd Thesis, Institut National Polytechnique de Grenoble - INPG, France.

[19] Dalal N., (2005). INRIA Person Dataset, http://pascal.inrialpes.fr/data/human/, 21 Şubat 2013.

[20] Khan, A. ve Rajpoot N., (2014). "Nonlinear Mapping Approach to StainNormalization in Digital Histopathology ImagesUsing Image-Specific Color Deconvolution", IEEE Transactions on Biomedical Engineering, Haziran 2014, Pittsburgh.

[21] Cambridge in Color, A Learning Community for Photographers, https://www.cambridgeincolour.com/tutorials/gamma-correction.html, 11 Mart 2016.

[22] Mıhçıoğlu, E., (2016). Pedestrian Detection System for Real-Time Applications, Yüksek Lisans Bitirme Tezi, Hacettepe Üniversitesi Elektrik ve Elektronik Mühendisliği Anabilim Dalı, Ankara.

[23] Bauer, S., Brunsmann, U. ve Schlotterbeck-macht, S., (2009). "FPGA Implementation of a HOG-based Pedestrian Recognition System", MPC-Workshop, Temmuz 2009, Aschaffenburg, Germany.

[24] Kadota, R., Sugano, H., Hiromoto, M., Ochi, H., Miyamoto, R. ve Nakamura, Y., (2009). "Hardware Architecture for HOG Feature Extraction", Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, 12-14 Eylül 2009, Kyoto.

[25] Özcan, A., (2013). Embedded Design and Implemetation of Pedestrian Detection System, Yüksek Lisans Bitirme Tezi, Yıldız Teknik Üniversitesi Elektronik ve Haberleşme Mühendisliği Anabilim Dalı, İstanbul.

[26] Benenson, R., Mathias, M., Timofte, R., ve Gool, L., (2012). "Pedestrian detection at 100 frames per second", Computer Vision and Pattern Recognition, 1:2903 – 2910.

[27] Corinna, C. ve Vladimir, V., (1995). "Support-Vector Networks in Machine Learning", Boston Kluwer Academic Publishers, 1:273-297.

[28] Coomans, D. ve Massart, D.L., (1982). "Alternative k-nearest neighbour rules in supervised pattern recognitition: Part 1. K-nearest neighbour classification by using alternative voting rules", Analytica Chimica Acta, 136: 15-27.

[29] Viola, P. ve Jones, M., (2001). "Rapid object detection using a boosted cascade of simple features", IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1:511-518.

[30] Fort, J. C., Letremy, P. ve Cottrell, M., (2002). "Advantages and drawbacks of the Batch Kohonen algorithm", European Symposium on Artificial Neural Networks, 2:223-230.

[31] Harville, M., (2004). "Stereo person tracking with adaptive plan-view templates of height and occupancy statistics", International Journal of Computer Vision, 22:127–142.

[32] Remagnino, P. ve Foresti G., (2005). "Ambient intelligence: A new multidisciplinary paradigm", IEEE Transactions on Systems, 35:1–6, 2005.

[33] Vall´ee, M.,Ramparany F. ve Vercouter, L., (2005). "A multi-agent system for dynamic service composition in ambient intelligence environments", In International Conference on Pervasive Computing, 1:157–182.

[34] Rourke, A. ve Bell, M., (1994). "An image-processing system for pedestrian data collection", In International Conference on Road Traffic Monitoring and Control, 1:123–126.

[35] Leibe, B., Seemann, E. ve Schiele, B., (2005). "Pedestrian detection in crowded scenes", IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1:878–885.

[36] Cai, Q., ve Aggarwal, J., (1996). "Tracking human motion using multiple cameras", International Conference on Pattern Recognition, 3: 68–72.

[37] Sato, K. ve Aggarwal, J., (2004). "Temporal spatio-velocity transform and its application to tracking and interaction", Computer Vision and Image Understanding, 96:100–128.

[38] Lipton, A., Fujiyoshi H. ve Patil, R., (1998). "Moving target classification and tracking from real-time video", IEEE Workshop on Applications of Computer Vision, 1:8–14.

[39] Madabhushi, A. ve Aggarwal, J., (2000). "Using head movement to recognize human activity", International Conference on Pattern Recognition, 4:698–701.

[40] Maurin, B., Masoud, O. ve Papanikolopoulos, N., (2002). "Camera surveillance of crowded traffic scenes", ITS America Annual Meeting, 1:28–55.

[41] Cucchiara, R., Grana, C., Piccardi, M., Prati, A. ve Sirotti, S., (2001). "Improving shadow suppression in moving object detection with hsv color information", IEEE Intelligent Transportation System Conference, 1:334–339.

[42] Moeslund, T. Ve Granum, E., (2000). "Multiple cues used in model-based human motion capture", IEEE International Conference on Automatic Face and Gesture Recognition, 1:362–367.

[43] Haritaoglu, I., Harwood, D. ve Davis, L., (1998). "W4: Who? when? where? what? a real time system for detecting and tracking people", International Conference on Face and Gesture Recognition, 1:222–227.

[44] Stauffer, C., ve Grimson, W., (1999). "Adaptive background mixture models for real-time tracking", IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1:246–252.

[45] Nanda, H., BenAbdelkader, C. ve Davis, L., (2003). "Modelling pedestrian shapes for outlier detection: A neural net based approach", IEEE Intelligent Vehicle Symposium, 1:428–433.

[46] Ali, T. ve Dagless E (1990). "Vehicle and pedestrian detection and tracking", IEE Colloquium on Image Analysis for Transport Applications, 1:48–54.

[47] Khan, S. ve Shah, M., (2000). "Tracking people in presence of occlusion", Asian Conference on Computer Vision, Ocak 8-11 2000, Taipei.

[48] Niyogi, S. ve Adelson, E., (1994). "Analyzing and recognizing walking figures in xyt", IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1:469–474.

[49] Masoud, O. ve Papanikolopoulos, N., (1997). "A robust real-time multi-level model-based pedestrian tracking system", ITS America Annual Meeting, 1:47–69.

[50] Masoud, O. ve Papanikolopoulos, N., (1997). "Robust pedestrian tracking using a model-based approach", IEEE Conference on Intelligent Transportation Systems, 1:338–343.

[51] Nanda, H. Ve Davis, L., (2002). "Probabilistic template based pedestrian detection in infrared videos", In IEEE Intelligent Vehicle Symposium, 1:15–20.

[52] Dollar, P., Appel, R., Belongie, S., ve Perona, P., (2014). "Fast Feature Pyramids for Object Detection", Transactions on Pattern Analysis and Machine Intelligence, 1:1532-1545.

[53] Tetik, Y.E., (2014). Sayısal Resimlerdeki Yayaların Tespiti, Doktora Bitirme Tezi, Yıldız Teknik Üniversitesi, Elektrik ve Haberleşme Mühendisliği Anabilim Dalı, İstanbul, 2014.

[54] Tetik, Y.E. ve Bolat, B., (2011). "Detection of Pedestrians From Still Images", IEEE SIU 19th Signal Processing and Communication Applications Symposium, 20-22 Nisan 2011, Kemer.

[55] Çavdar, Ç., Özek, A. ve Bolat, B. (2018). "Pedestrian detection with multiple classifiers on still images", IEEE SIU 26th Signal Processing and Communication Applications Symposium, 2-5 Mayıs 2018, Çeşme.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı	:Çağlar ÇAVDAR
Doğum Tarihi ve Yeri	:10.04.1992, Ereğli/KONYA
Yabancı Dili	:İngilizce
E-posta	:caglarcavdar92@gmail.com

ÖĞRENİM DURUMU

Derece	**Alan**	**Okul/Üniversite**	**Mezuniyet Yılı**
Lisans	Elektronik ve Haberleşme Mühendisliği	Kocaeli Üniversitesi	2015
Lise	Fen Bilimleri	Ereğli Atatürk Lisesi	2010

İŞ TECRÜBESİ

Yıl	**Firma/Kurum**	**Görevi**
2016	NETAŞ	Yazılım Tasarım Mühendisi
2015	CARDTEK	Yazılım Mühendisi

YAYINLARI

Bildiri

1. Çavdar, Ç., Özek, A. ve Bolat, B. (2018). "Pedestrian detection with multiple classifiers on still images", IEEE SIU 26th Signal Processing and Communication Applications Symposium, 2-5 Mayıs 2018, Çeşme.