T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

SERT NEGATİF ÖĞRENME İLE YAYALARIN YÜRÜYÜŞ ROTALARININ BELİRLENMESİ

CİHANGİR KOLCU

YÜKSEK LİSANS TEZİ
ELEKTRONİK VE HABERLEŞME MÜHENDİSLİĞİ ANABİLİM DALI
ELEKTRONİK PROGRAMI

DANIŞMAN
DR. ÖĞR. ÜYESİ BÜLENT BOLAT

İSTANBUL, 2018

T.C.

YILDIZ TEKNİK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

SERT NEGATİF ÖĞRENME İLE YAYALARIN YÜRÜYÜŞ ROTALARININ BELİRLENMESİ

Cihangir KOLCU tarafından hazırlanan tez çalışması 22.06.2018 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

T.C. Yıldız Teknik Üniversitesi

Jüri Üyeleri

Dr. Öğr. Üyesi Bülent BOLAT

T.C. Yıldız Teknik Üniversitesi

Dr. Öğr. Üyesi Umut Engin AYTEN

T.C. Yıldız Teknik Üniversitesi

Prof. Dr. Mehmet SAĞBAŞ

T.C. İstanbul Yeni Yüzyıl Üniversitesi

ÖNSÖZ

Öncelikle bana güvenerek bu proje üzerinde çalışmamı sağlayan ve projeyi geliştirme süresince beni yönlendirerek bilgisini, görüşlerini ve deneyimlerini aktaran danışman hocam Sayın Dr. Öğr. Üyesi Bülent BOLAT; hayatım boyunca benim için her türlü fedakarlıkta bulunarak desteklerini eksik etmeyen sevgili annem Nermin KOLCU ve ablalarım Semra, Selda ve Serap KOLCU; rahmetle andığım babam Mustafa KOLCU; yardım ve desteği ile herzaman yanımda olan eşim Sena KOLCU' ya teşekkürlerimi sunarım.

Haziran, 2018

Cihangir KOLCU

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 2
1.2 Tezin Amacı 4
1.3 Hipotez 4

BÖLÜM 2

OTONOM ARAÇ SİSTEMLERİ 5

2.1 Otonom Araçların Tarihçesi 5
2.2 Otonom Araçların Çalışma Prensipleri 6
2.3 Otonom Araçların Avantajları 8
2.4 Otonom Araçların Dezavantajları 8

BÖLÜM 3

ÖRÜNTÜ TANIMA VE ÖZNİTELİK ÇIKARMA YÖNTEMLERİ 10

3.1 Örüntü Tanıma 10
3.1.1 Ön İşlemler 11
3.1.2 Öznitelik Çıkarımı 11
3.1.3 Sınıflandırma 12
3.2 Öznitelik Çıkarma Yöntemleri 14
3.2.1 Düşük Seviyeli Öznitelik Çıkarma Yöntemleri 14

3.2.1.1 Kenar Belirleme Yöntemi 14
3.2.1.2 Köşe Belirleme Yöntemi 14
3.2.1.3 Özel Bölge Belirleme Yöntemi 15
3.2.2 Yüksek Seviyeli Öznitelik Çıkarma Yöntemleri 15
3.2.2.1 Bağımsız Öznitelik Çıkarma Yöntemleri 15
3.2.2.2 Şekil Tabanlı Öznitelik Çıkarma Yöntemleri 17
3.2.2.3 Şekil Eşleştirme Tabanlı Öznitelik Çıkarma Yöntemleri 17
3.2.2.4 Türev Tabanlı Öznitelik Çıkarma Yöntemleri 18

BÖLÜM 4

YÖNLÜ TÜREVLERİN HİSTOGRAMI 20

BÖLÜM 5

DESTEK VEKTÖR MAKİNELERİ 24

5.1 Doğrusal Destek Vektör Makineleri 24
5.2 Doğrusal Olmayan Destek Vektör Makineleri 25
5.3 Literatürde Destek Vektör Makineleri Kullanılan Çalışmalar 27

BÖLÜM 6

GÖRÜNTÜLERDEKİ YAYALARIN TESPİT EDİLMESİ 28

6.1 Arka Plan Çıkarma Tekniği Kullanarak Görüntüdeki Yayaların Tespit Edilmesi 29

BÖLÜM 7

HOG VE DVM YÖNTEMLERİ KULLANARAK YAYALARIN YÜRÜYÜŞ ROTALARININ BELİRLENMESİ 35

7.1 Çalışmada Kullanılan Veri Seti 35
7.2 Sistemin Çalışma Prensibi 35

BÖLÜM 8

SONUÇ VE ÖNERİLER 44

KAYNAKLAR 45

EK-A

SİMÜLASYON SONUÇLARI 51

ÖZGEÇMİŞ 56

SİMGE LİSTESİ

b	Türev yön bölgelerinin açı değeri
B	RGB sinyali mavi renk bileşeni
d	Öklid Uzaklığı
f	Normalize edilmiş blok çıkış vektörü
F_m	Harmonik ortalama
G	RGB sinyali yeşil renk bileşeni
gx(x,y)	Gradyan yatay bileşeni
gy(x,y)	Gradyan dikey bileşeni
I	Görüntü
K	Destek Vektör Makineleri Çekirdek Fonksiyonu
$\boldsymbol{L_D}$	Destek Vektör Makineleri İkincil Form
$\boldsymbol{L_P}$	Destek Vektör Makineleri Birincil Form
P(A)	A için Önsel (Marjinal) Olasılık
P(A|B)	Verilen B için A'nın Koşullu Olasılığı
P(B)	B Olayı İçin Önsel Olasılık
P(B|A)	Verilen A için B'nin Koşullu Olasılığı
R	RGB sinyali kırmızı renk bileşeni
θ	Türevlerin yönelim açıları
xi	Öğrenme Kümesindeki Örnekler
xq	Sınıflandırılmak İstenen Örnek
$\|\|v\|\|_n$	n. dereceden norm
v	Blok çıkış vektörü v
w	Ağırlık
α	HOG büyüklük dağıtım çarpanı
$\boldsymbol{\alpha_i}$	Lagrange Katsayıları

KISALTMA LİSTESİ

BBA	Bağımsız Bileşen Analizi
CoHED	Co-occurence Histograms of Pairs of Edge Orientations and Color Differences
CoHD	Co-occurrence Histograms of Color Differences
CoHOG	Co-occurence HOG
ÇTBA	Çekirdek Temel Bileşenler Analizi
DDA	Doğrusal Diskriminant Analizi
DoG	Difference-of-Gaussian
DoH	Determinant-of-Hessian
DN	Doğru Negatif
DP	Doğru Pozitif
DVM	Destek Vektör Makinası
GLOH	Gradient Location and Orientation Histogram
GPS	Global Positioning System
GRD	Gerçek Referans Değer
HOG	Histogram of Oriented Gradients
KLT	Kanade-Lucas-Tomasi algoritması
LoG	Laplacian-of-Gaussian
MSER	Maksimum Durağan Uç Bölgeler
RGB	Red Green Blue renk uzayı
SD	Sistem Duyarlılığı
SIFT	Scale-Invariant Feature Transform
SK	Sistem Kesinliği
SURF	Speeded Up Robust Features
SVM	Support Vector Machine
TBA	Temel Bileşen Analizi
YN	Yanlış Negatif
YP	Yanlış Pozitif
YSA	Yapay Sinir Ağları

ŞEKİL LİSTESİ

Sayfa

Şekil 3. 1 Örüntü tanıma işlem basamakları.....11
Şekil 4. 1 Türev yön bilgileri.....21
Şekil 4. 2 HOG öznitelik çıkarımında hücre ve blok kullanımı.....22
Şekil 5. 1 Doğrusal DVM.....25
Şekil 5. 2 Doğrusal olmayan DVM.....27
Şekil 6. 1 Sabit arka plan ve çıkartılacak yaya görüntüsü.....29
Şekil 6. 2 Gri seviye arka plan ve çıkartılacak yaya görüntüsü.....29
Şekil 6. 3 Arka plan ve çıkartılacak yayaya ait histogram.....30
Şekil 6. 4 Otsu eşikleme uygulama.....30
Şekil 6. 5 Arka plan çıkartılmış görüntü.....31
Şekil 6. 6 Boşlukları doldurulan ve Medyan filtre uygulanan görüntü.....31
Şekil 6. 7 Aşındırma ve Açma işlemi.....32
Şekil 6. 8 Ön işlem basamakları.....33
Şekil 6. 9 İşlenmiş yürüme siluet dizisi.....34
Şekil 7. 1 Yürüyüş rota belirleme sistemi işlem basamakları.....36
Şekil 7. 2 HOG test giriş görüntüsü ve çıkış öznitelikleri yön haritası.....36
Şekil 7. 3 HOG test görüntüsü ve özniteliklerin yön haritası (8x8 hücre).....37
Şekil 7. 4 HOG test görüntüsü ve özniteliklerin yön haritası (32x32 hücre).....37
Şekil 7. 5 Leave One Out Cross Validation iterasyon örneği.....38
Şekil 7. 6 Uygulama sonucunda elde edilen örnekler.....41

ÇİZELGE LİSTESİ

Sayfa

Çizelge 5. 1 DVM Kullanılarak yapılan çalışmalar ve başarı oranları.....27
Çizelge 7. 1 Performans sonuçları.....42
Çizelge 7. 2 Hipotez test ölçütü.....42
Çizelge 7. 3 Hipotez tet sonuçları..... 43

ÖZET

SERT NEGATİF ÖĞRENME İLE YAYALARIN YÜRÜYÜŞ ROTALARININ BELİRLENMESİ

Cihangir KOLCU

Elektronik ve Haberleşme Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

Tez Danışmanı: Dr. Öğr. Üyesi Bülent BOLAT

Gelişen teknoloji ile birlikte günümüzde otonom araçların kullanımı da hızla artmaktadır. Ancak otonom bir aracın şehir trafiği içerisinde güvenli olarak hareket edebilmesi için başa çıkması gereken bazı durumlar bulunmaktadır. Otonom çalışan bir araç, akan trafik içinde bir sürücü gibi kendi durumunun farkında olmalı, güvenli ve etkili davranış kararları verebilmelidir. Otonom araç yalnızca mevcut durumdaki tüm unsurları tanımlamakla kalmamalı, aynı zamanda geçmiş deneyimlerden elde edilen bilgi ve tecrübeleri kullanarak gelecekteki durumlar için tahminlerde bulunabilmelidir.

Bu çalışmada Yönlü Gradyanların Histogramı (HOG) algoritması ve Destek Vektör Makinesi (SVM) kullanılarak yayaların yürüyüş rotalarının kestirilmesi ile otonom araç sürüşlerinin daha güvenli hale gelmesi hedeflenmektedir. Araç içerisine yerleştirilen kamera üzerinden elde edilen görüntüler yardımıyla yayaların belirlenmesi ve davranışlarının kestirilmesine çalışılmıştır.

Elde edilen video görüntülerinden çıkarılan her çerçeve için yayanın piksel cinsinden konumu ve kameranın özdevinimi bulunduktan sonra, aracın özdeviniminden kaynaklanan konum değişikliği giderilir. Böylece aracın hareketinin, yayanın hareketi üzerindeki etkisi elimine etmeye çalışılır. Özdevinim çıkartıldıktan sonra yayanın konumu bir önceki çerçevede hesaplanan konumdan çıkarılarak yayanın bir çerçeve boyunca ne

kadar hareket ettiği bulunur. Bulunan hareket miktarlarının ortalaması yardımıyla yayanın bir sonraki çerçevede nerede bulunması gerektiği kestirilir.

Çalışmanın sonucunda önerilen yöntemin Daimler Mono Pedestrian Benchmark veri seti üzerinde test edilerek, sunulan probleme başarılı bir çözüm sunduğu görülmüştür. Yaya tespit başarımı DVM'nin Gauss radyal temelli çekirdek fonksiyonu performans sonuçları üzerinden değerlendirildiğinde mevcut yayaların %95.27 oranında tespit edilebildiği görülmüştür. Tespit edilen yayaların rotaları ise %96.69 oranında doğru öngörülmüştür.

Anahtar Kelimeler: Histogram of Oriented Gradients, Destek Vektör Makineleri, Yaya tespiti

ABSTRACT

WALKING PATH DETECTION OF PEDESTRIANS BY USING HARD NEGATIVE LEARNING

Cihangir KOLCU

Department of Electronics and Communications Engineering

MSc. Thesis

Adviser: Assist. Prof. Dr. Bülent BOLAT

Parallel to the developing technology, the use of autonomous vehicles is increasing rapidly. An autonomous vehicle has to be able to move safely within city traffic and there are some situations that need to be addressed. An autonomous vehicle flowing in traffic must be aware of its own situation, behave safe and effective like a driver, and be able to make critical decisions. Autonomous vehicle must use the past experience as well as the current state to predict future states.

In this study, Histogram of Directional Gradients (HOG) algorithm and Support Vector Machine (SVM) are used to predict pedestrians' routes to make the autonomous cars safer.
It is aimed to make their driving safer. A camera located inside the vehicle to capture the video signal. For each frame subtracted from the resulting video images, positions of the pedestrians are calculated and the camera's self-movement is eliminated. After the self-movement is eliminated, the routes of the pedestrians are predicted by using frame by frame position changes. With the help of the average of the movement found, the future positions of the pedestrians are predicted.

The proposed method was applied on Daimler Mono Pedestrian Benchmark dataset. The results are promising regarding to the literature. If Gaussian Kernel SVM is used,

95.27 per cent of the pedestrians were detected. 96.69 per cent of the routes of detected pedestrians were correctly predicted.

Keywords: Histogram of Oriented Gradients, Support Vector Machine, Pedestrian detection

BÖLÜM 1

GİRİŞ

Günümüzde otonom sürüşlü araçların gelişimi hızla ilerlemektedir. Ancak otonom bir aracın şehir trafiği içerisinde hareket edebilmesi için başa çıkması gereken durumlar halen önemli zorluklar içermektedir. Otonom çalışan bir araç, akan trafik içinde bir sürücü gibi kendi durumunun farkında olmalı, güvenli ve etkili davranış kararları verebilmelidir. Otonom araç yalnızca mevcut durumdaki tüm unsurları tanımlamakla kalmamalı, aynı zamanda geçmiş deneyimlerden elde edilen bilgi ve tecrübeleri kullanarak gelecekteki durumlar için tahminlerde bulunabilmelidir.

Otonom sürüş problemi henüz çözülememiş olsa da, bu alanda yapılan çalışmaların bir kısmı sürücü destek sistemleri olarak kullanılmaya başlanmıştır. Sürücü destek sistemleri, dahili sensörlerden gelen verileri kullanarak nesneleri tespit etme ve kısa mesafede konumlarını önceden tahmin etmeye çalışmaktadır. Yaygın olarak kullanılan tahmini modeller; konum, hız ve ivme gibi kinematik değerler arasındaki fiziksel bağıntılara dayanır. Ancak özellikle yayaların geniş bir teorik dinamiği olduğundan, tahmine dayanan Aktif Güvenlik Sistemleri, yalnızca sokağa girmiş olan ya da bunu yapmakta olan yayalara tepki verebilir. Bu gibi reaktif davranış, araçların otonom sürülebilir olması için yeterli değildir. Bu sorun, mevcut durumdaki yayaların niyetini tespit ederek; beklenen eylemler yoluyla gelecek durumlarının daha iyi kavranmasını sağlama ile giderilebilmektedir [1].

Yayaların yürüyüş rotalarının belirlenebilmesi için önceklikle video üzerindeki yürüyen kişilerin tespit edilmesi gerekmektedir. Bu işlem örüntü tanıma algoritmaları ile gerçekleştirilmektedir.

Örüntü tanıma teknolojisi, fizyolojiye veya davranış özelliklerine dayalı olarak insanları ayırt etmekte kullanılır. Bu doğrultuda yürüyen kişinin tespit edilmesi, çeşitli güvenlik sistemleri ve savunma sanayindeki kullanımının yanı sıra hastalık tanısı üzerine yapılan araştırmalardan dolayı da önem kazanmaktadır.

Yürüyüş stiline dayalı örüntü tanıma sistemleri model ve siluet tabanlı olmak üzere iki farklı yaklaşımdan oluşmaktadır. Model tabanlı yaklaşımlar yürüyüş hareketindeki dinamik karakteristikleri matematiksel olarak üretebilmeye dayanırken, siluet tabanlı yaklaşımlar insan vücudunun siluetine dayalı olarak üretilen örüntü yapılarının istatistiksel davranış karakteristiklerine dayanmaktadır [2].

Bu çalışmada yayaların rotalarının belirlenmesi; MATLAB ortamında geliştirilip, Yönlü Türevlerin Histogramı ve Destek Vektör Makineleri kullanılarak sınıflandırılmıştır.

1.1 Literatür Özeti

Yayaların rotalarının belirlenmesine dayalı biyometrik tanıma çalışmaları literatürde model ve siluet tabanlı olmak üzere iki farklı yaklaşımdan oluşmaktadır.

Model tabanlı yaklaşımlar insan vücudundaki eklem noktalarının kinematiği ve adım davranışları gibi yürüyüş hareketindeki dinamik karakteristikleri matematiksel olarak üretebilmeye dayanmaktadır.

Siluet tabanlı yaklaşımlar ise yürüyüş hareketinde insan vücudunun siluetine dayalı olarak üretilen örüntü yapılarının istatistiksel davranış karakteristiklerine dayanmaktadır. Siluet tabanlı yaklaşım da kendi içerisinde durum-uzay metodu ve zaman-uzam metodu olmak üzere ikiye ayrılır.

Durum-uzay metodu hareketsiz vücut pozları dizisinin birleşiminden oluşur ve hareketsiz pozların zaman içerisindeki değişimlerini gözlemleyerek tanımayı gerçekleştirir. Zaman-uzam metodu ise yürüyüş hareketi tarafından sürekli olarak oluşturulan zaman-uzam dağılımını karakterize ederek tanımayı gerçekleştirir. Sonuç olarak; durum-uzay metodu ile statik özellikler, zaman-uzam metodu ile de dinamik özellikler elde edilebilir.

Bu alanda yapılan ilk çalışmada aktif güvenlik sistemleri için yol tahmininin geliştirilmesine odaklanılmıştır. Bu nedenle yayanın hedefinin caddeye girip

girmemesinin tespiti problem olarak kabul edilmiştir. Schneemann ve Heinemann yapmış oldukları çalışmada yaya trafiğinin yol ve çevre ile olan ilişkisini tanımlayan bir özellik vektörü sunmuşlardır. Bağlam tabanlı özellik vektörünü, geçiş niyetine sahip yayaları tespit etmek için bir DVM öğrenme algoritması ile birlikte kullanmışlardır [1].

Bazı çalışmalarda ise farklı sahne öğelerinin varlık ve uzaysal düzeni tarafından verilen bağlamsal bilgiler kullanmıştır. Kooij ve arkadaşları yayaların karşıdan karşıya geçip geçmeyeceğini öngören bir yöntem önermişlerdir [3]. Önerilen yöntemde yayaların konumları stereo kamera ile tespit edilmeye çalışılmıştır. Araç üzerindeki monoküler bir kamera ile çekilen görüntüler üzerinde çalışan Hariyono ve arkadaşları yayanın yörüngesini belirlemek için Kalman filtresi, etkileşen çoklu modeller ve olasılıklı hiyerarşik yörünge uyumunu kullanmışlardır [4]. Yaya yolunun yönünü ardışık çerçeveleri birleştirerek belirlemiş ve her çerçevedeki yayaları Haar özellik tabanlı bir kademeli sınıflandırıcı kullanarak belirlemişlerdir [4]. Jeong ve arkadaşları a kızılötesi (FIR) kameralar kullanarak ani yaya geçişlerini tespit etmeye odaklanmışlardır. Bu çalışmada karşılaşılan en büyük zorluk ise yaz saatlerinde yolun sıcaklığı ile yayaların vücut ısılarının benzer veya daha yüksek olması olarak bildirilmiştir [5]. Schauland ve arkadaşları üç boyutlu doğrusal ötelemeyle değişmez (LSI) süzgeçler kullanarak, farklı hız ve doğrultulara sahip nesnelerin tespit edilmesine çalışılmıştır [6].

Köhler ve arkadaşları kaldırımda ayakta duran ya da yürümeye başlayan yaya olup olmadığına karar vermek için hareket kontur görüntüsü tabanlı türevlerin histogramı benzeri tanımlayıcı (MCHOG) ile destek vektör makinesi (DVM) sınıflandırıcıyı birlikte kullanmıştır [7]. Keller ve Gavrila konum ve optik akış özelliklerini birlikte kullanarak bir yayanın yürümeye devam edip etmediğini belirlemişlerdir [8]. Quintero ve arkadaşları aynı problemi stereo kameralar yardımıyla yayaların eklemlerinin konumlarına dayalı olarak tespit etmeye çalışmıştır [9]. Furuhashi ve Yamada bir yayanın karşıdan karşıya geçmek isteyip istemediğin belirlemek amacıyla kısa video görüntülerinden postur analizi gerçekleştirmiştir [10].

Negri ve Garayalde, yaya takibi için yeni bir yöntem önermiştir [11]. Bu çalışmada her yaya otonom bir ajan olarak düşünülmüş ve bir durum vektörü ile modellenmiştir. Bu ajanlara hedefcik (targlet) adı verilmektedir. İzleme işlemi, Hareket Özellik Uzayı

üzerinde hesaplanan bir kişi dedektörü tarafından başlatılır. Dedektör çıktıları sıra boyunca kendi yörüngelerini elde ederek; her hedefciğin olası rotalarını çıkarır ve bir Kalman süzgeci ile en olası rota belirlenir. Kataoka ve arkadaşları çalışmalarında genişletilmiş eşdizimli yön türevlerinin histogramı (HOG) tanımlayıcısını kullanarak yayaları tespit etmiş ve daha sonra yaya izleme basamağında ise Parçacık Filtresi için optik akış kullanmışlardır [12]. Maki ve arkadaşları kenar yönelimlerinin çift oyla seçilmesi ile elde edilen eşdizimli HOG yönteminden esinlenerek birlikte oluşum akışı (CoF) adında yeni bir hareket özelliği tanımlamışlardır [13]. Birlikte oluşum akışı fikrini, farklı vücut parçalarının bir hareket özelliği içerisinde hareketlerindeki olası tutarlılığı yakalamak olarak nitelendirmişlerdir.

1.2 Tezin Amacı

Yayaların yürüyüş rotalarının kestirilmesi ile otonom araç sürüşlerinin daha güvenli hale gelmesi hedeflenmektedir. Araç içerisine yerleştirilen kamera üzerinden elde edilen görüntülerde; yayaların belirlenmesi ve davranışlarının kestirilmesi ile aracın yayaya çarpma ihtimalini en aza indirgemek çalışmanın ana amaçlarından bir tanesidir.

1.3 Hipotez

Çalışmada yayaların yürüyüş rotalarının belirlenmesi için yönlü türevlerin histogramı (HOG) ve DVM temelli bir yöntem önerilmiştir. Önerilen yöntem Daimler Benchmark veri seti üzerinde test edilmiş ve başarılı sonuçlara ulaşılmıştır.

Çalışmanın dördüncü bölümde HOG yöntemi üzerinde durulmuş, beşinci bölümünde destek vektör makineleri özetlenmiştir. Altıncı bölümde uygulama detayları ve yedinci bölümde ise çalışma ile ilgili sonuçlar yer almaktadır.

BÖLÜM 2

OTONOM ARAÇ SİSTEMLERİ

Otonom araçlar insan müdahalesine gerek duymadan çevresindeki nesneleri tespit etmek için; sahip olduğu çeşitli algılayıcılar yardımı ile bağımsız bir şekilde hareket etme kabiliyetine sahip otomobillerdir.

Otonom araç teknolojilerinin ilk hedefi; insan faktörünün devre dışı bırakılması ile birtakım teknolojilerden faydalanılarak, otomobillerin bağımsız ve güvenli bir şekilde hareket etmesidir. İnsanların araç kullanırken sergiledikleri davranış ve algılama yeteneklerinin çeşitli teknolojiler ile otonom araçlara aktarılması ve bunun sonucunda trafikte oluşabilecek kaza riskinin en aza inmesi beklenmektedir. Bu araçlar hareket ve algılama kabiliyetlerini kullanırken radar, lidar, sensör ve GPS gibi teknolojilerden faydalanarak, daha kısa zamanda etkili ve güvenli kararlar alabilmektedir [14].

2.1 Otonom Araçların Tarihçesi

Bu alanda yapılan ilk çalışmalar 1920 ve 1930 yıllarında gerçekleştirilmiştir. Japon Tsukuba Mekanik Mühendislik laboratuvarlarında 1977 yılında birtakım çalışmalar yapılmıştır. Konu ile ilgili 1984 yıllarında Carnegie Mellon Üniversitenin gerçekleştirmiş olduğu Navlab ve ALV projesi ilklerden sayılmaktadır. Sonraki çalışma ise 1987 tarihinde Mersedes-Benz firmasının Bundeswehr Üniversitesi ile birlikte yürüttüğü EUREKA Prometheus Projesidir. 1987-1995 tarihleri arasında gerçekleştirilen DARPA Otonom Kara Araçları projesi ise Mersedes-Benz ve ABD Savunma Departmanı tarafından gerçekleştirilmiştir [15] [16].

Günümüzde ise başta otomotiv sanayi olmak üzere birçok firma otonom araçların geliştirilmesi konusunda çalışma yapmaktadır. Bu şirketlerden bazıları ve yürüttükleri projeler aşağıdaki gibidir.

- Audi - Piloted Driving
- BMW - Electronic co-pilot system
- Bosch - Autonomous Vehicle
- Ford - Automated Fusion Hybrid
- General Motor - Super Cruise, Chevy EN-V
- Google - Driverless Car Project
- Lexus - Advanced Active Safety Research Vehicle
- Mercedes-Benz - Mercedes-Benz Intelligent Drive
- Nissan - Autonmous Drive
- Tesla - Auto Pilot
- Volkswagen - Temporary Auto Pilot
- Volvo - Drive Me

Bu alanda araç geliştiren ve yolcu taşımacılığı üzerinde faaliyet gösteren UBER firması da kendilerine ait otonom araçların testlerini 2015 yılından itibaren yapmaya başlamışlardır. 2016 yılının sonlarına doğru UBER bu araçları başta San Francisco ve Toronto şehirlerinde olmak üzere test olarak kullanmaya başladı. Google ise otonom araç çalışmalarıyla ön plana çıkmayı başarmış ve bu konudaki ilk ciddi gelişmelerin de öncülerinden oldukları bilinmektedir [18].

2.2 Otonom Araçların Çalışma Prensipleri

Otonom araçları sürücülü araçlardan farklılaştıran en önemli özelliği; sürücünün dikkatini dağıtabilecek; radyo, cep telefonu veya araç içerisindeki kişilerle konuşma gibi durumlarının olmamasıdır. Otonom araçlar verimli olarak çalışabilmeleri için; aracın sistemlerinin yedekli yapıda olması, doğru bilgi için farklı kaynaklar üzerindeki bilgiyi

analiz ederek doğruluğunu test eden sistemler olmalıdır. Bu sistemlerin çalışma prensipleri bilgisayar teknolojileri ile benzer özellik göstermektedir. Otonom araç konusunda araştırma yapan firmalar farklı teknolojiler üzerinde çalışmaktadır. Bir otonom araç üzerinde kullanılabilecek en önemli teknoloji kamera sistemleridir. Otonom araçlar üzerindeki kameralar aracılığı ile çevresini algılamaya başlayıp çevre ile etkileşime geçer. Bu kameralar üzerinden alınan görüntüler, araçta bulunan yazılımlar sayesinde analiz edilerek, araç sistemlerinin doğru davranışlar sergilemesine yardımcı olur. Bu tez kapsamında araç üzerinde bulunan kameralar aracılığı ile alınan görüntüler üzerinde çalışarak, yayaların rotalarının belirlenmesi konusunda yazılım geliştirilmesi amaçlanmıştır.

Otonom araçlar üzerlerinde bulunan kameraların dışında; radar, şerit takip sistemi, Lidar ve GPS gibi sistemleri buluabilmektedir. Bu sistemlerden Radar otonom araç için önemli sayılabilecek diğer bir teknolojidir. Radarlar ile araç çevresindeki nesnelerin hareketleri yakalanarak yine araçta bulunan yazılımlar aracılığı ile bu verileri aracın hareket yeteneği için kullanmaktadır. Radarlar 360 derece açıyla çevrelerini tarama kabiliyetine sahip cihazlardır. Lazer radar sistemi olarak da biline Lidar [19] sistemleri otonom araçlar için bir diğer yardımcı teknolojidir. Lidar sistemleri aracılığı ile 3 boyutlu olarak aracın etrafından olan nesneler algılanarak; araçta bulunan yazılımlar sayesinde analiz edilir. Bu sistem tek başına yeterli olmadığından araç üzerinde bulunan kameralardan da faydalanılmaktadır. Şerit takip sistemleri adından da anlaşılacağı üzere araç üzerinde bulunan kameralar yardımı ile yol yüzeyi, yol kenarları ve şeritler arasındaki farkı algılamak için kullanılmaktadır. Son olarak GPS (Global Positioning System) ile aracın nereye gideceği ve anlık olarak konum bilgisi alınabilmektedir.

Tüm bu sistemler göz önünde bulundurulduğunda otonom araçların gelişimlerinin belirli aşamalarda olması beklenmektedir. Bu aşamaları sıralayacak olursak öncelikle; araçların kendi kendilerine park etmeleri, sonrasında otobanlarda kendi başlarına hareket edebilme kabiliyetlerin geliştirilmesi gerekmektedir. Bu aşamaların başarılı olması halinde araçların şehir merkezlerinde de sürücüsüz bir şekilde hareket edebilmeleri bu problemin en önemli basamaklarından olacaktır. Şehir merkezlerine entegre edilmesi durumunda otonom araçların toplu taşıma araçları şeklinde kullanılacak seviyelere

ulaşması ve hayatın her alanında bu araçların kullanıldığı bir oto-ekosistemin oluşturulması hedefler arasındadır [20].

2.3 Otonom Araçların Avantajları

Otonom araçların hayata geçirilmesindeki en önemli faktör otonom aracın kullanacağı yazılım, donanım ve altyapının dünyanın her yerinde aynı standartta ve uyumda olması zorunluluğudur. Otonom araçlarda standartların sağlanması otonom olmayan araçlardaki standartlardan daha da önemlidir. Bu zorunluluktan dolayı da ülkelerin tamamında trafik sinyalizasyon, denetleme ve kamera gibi trafikle ilgili her alanda standartlar sağlanmış olduğunda insan yaşam kalitesi de olumlu yönde artacaktır.

Otonom araçların daha az hata yapma olasılığından dolayı insan hayatının korunmasında önemli bir rol üstleneceklerdir. Akıllı sürüş teknikleri ve daha az araç ihtiyacı olacağından daha az yakıt tüketimi ve yollarda daha az trafik meydana gelecektir. Kullanılan kamera ve görüş sistemleri sayesinde trafik kazalarında önemli bir role sahip olan kör nokta sorunu ortadan kaldırılacaktır. Trafikte bulunan araç sayısının azalması ile birlikte havaya salınan zararlı gazlar azalacaktır [21]. Araç sürerek zamanın büyük çoğunluğunu yolda harcayan kişilerin bu zamanlarını seyahat sırasında kendilerine ayırarak değerlendirmeleri büyük avantaj sağlayacaktır. Aynı zamanda bedensel engeli bulunan veya yaşı küçük olan kişilerin de otonom araçla seyahat edebilme imkanı sayesinde, ebeveynlerin çocuklarına eşlik etmek zorunluluğu ortadan kalkmış olacaktır [22].

2.4 Otonom Araçların Dezavantajları

Çoğu yeni teknolojide olduğu gibi otonom araçların kullanıma başlanması sonucunda her ne kadar avantajlı yönleri olsa da bir takım istenmeyen sonuçlar meydana gelecektir. Bunların başında, üretim ve servis maliyetlerine bağlı olarak, araç fiyatlarındaki ciddi artışlar olacaktır. Ayrıca bu araçlara üzerinde kullanılacak yazılım ve donanımlar standart araçtakilerden çok daha gelişmiş olacağından; bu durum da maliyetleri ayrıca artıran bir faktör olacaktır.

Her ne kadar otonom araçlarla yapılan testlerde başarılı sonuçlar üretilse de; hava şartlarının trafiğe elverişli olmadığı zamanlarda, özellikle kar ve tipinin olduğu veya internet erişimin olmadığı durumlarda ulaşımda bir takım sorunlar yaşanabilecektir [23].

Google firması otonom araçların yazılımlarının siber saldırıya uğraması, ele geçirilmesi ya da kötü amaçlarla kullanmasının çok zor olduğunu iddia etseler de bu hususu çevre güvenliği açısından dikkat edilmesi gereken önemli bir noktadır. Bu sistemin ele geçirilmesi sonucunda bu araçların uzaktan kumanda edilebilen silah ve bombaların patlatılmasıyla çeşitli suikastların işlenmesi ya da toplumsal katliamların yapılması riski her zaman için var olacaktır [24].

Otonom araçlar şuan için bir takım riskler içerse de yakın gelecekte bu araçlar vasıtası ile seyahat etmek mümkün olacak. Bunun sonucunda insanlar daha kaliteli yaşam standartlarına ulaşacaktır.

BÖLÜM 3

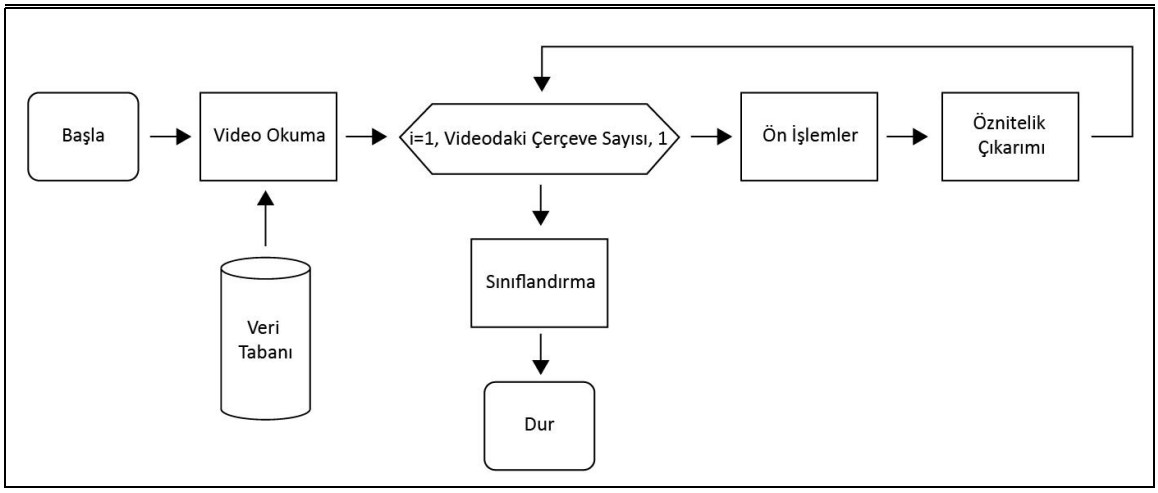
ÖRÜNTÜ TANIMA VE ÖZNİTELİK ÇIKARMA YÖNTEMLERİ

3.1 Örüntü Tanıma

Örüntü; ölçülebilen, gözlenebilen, tekrar edebilen ortak düzenli yapı veya benzerlikleri olan bir örnekler kümesidir. Bir başka ifade ile örüntü, olay veya nesnelerin düzenli bir biçimde birbirini takip ederek gelişmesi olarak tanımlanmaktadır [25].

Örüntü tanıma, aralarında ortak özellik bulunan ve birbirleri ile ilişki kurulabilen işaret sinyallerini veya nesneleri çeşitli yöntemler ile tanımlama veya sınıflandırma işlemidir. Bu doğrultuda örüntü tanımanın amacı; bilinmeyen örüntü sınıflarına belirli bir şekil vermek ve bilinen bir sınıfa ait olan örüntüyü tespit etmektir. Biz canlılar olarak çevremizi duyu organlarımızı kullanarak algılarız. Vücudumuzda bulunan bu sensörler beyindeki algılama için gerekli olan uyarım işlemini gerçekleştirirler [26]. Çevreyi algılamayı sağlayan örüntü tanıma sistemlerinde de, canlılardaki duyu organlarının işlevlerini sağlayan sensörler kullanılmaktadır. Bu sensörler ile elde edilen bilgiler; bilgisayar sistemleri üzerinde işlenerek örüntü tanıma sürecinde sayısal sinyallere dönüştürülür. Kamera gibi cihazlar sayesinde kayıt edilerek elde edilen görüntüler, sinyal veya görüntü işleme teknikleri kullanılarak örüntü tanımaya uygun hale getirilmiş olur.

Örüntü tanıma süreç olarak; önişlem, öznitelik çıkarımı, öznitelik seçimi, öğrenme ve sınıflandırma işlem basamaklarından oluşmaktadır. Örnek olarak video üzerinde gerçekleştirilecek örüntü tanıma sistemi aşağıdaki akış şemasında gösterilen işlem sırasını takip ekmektedir.



Şekil 3. 1 Örüntü tanıma işlem basamakları

3.1.1 Ön İşlemler

Örüntü tanıma sistemlerinde öznitelik çıkarımından önce, sinyal veya görüntünün bir önişlemden geçirilmesi gerekebilir. Kullanılan bütün ön işlemlerin amacı farklı girdiler karşısında aynı sonucu elde etmektir. Her farklı görüntü için benzer özelliklerin elde edilebilmesi için görüntü üzerindeki istenmeyen bütün özellikler temizlenmelidir. Bu işlemler öncelikle görüntünün bilgisayar ortamına aktarılmasıyla birlikte: sayısallaştırma, eşik değeri elde etme, ikili görüntü elde etme, filtre uygulamaları ve morfolojik işlemlerden oluşmaktadır. Elimizdeki veriyi sadeleştirmemizi sağlayan önişlemlerin ardından öznitelikler çıkarılır.

3.1.2 Öznitelik Çıkarımı

Örüntü tanıma yapabilmek için nesneye özgü çeşitli ayırt edici özelliklere ihtiyaç duyulur. Bu özellikler 'öznitelik' olarak adlandırılır ve öznitelikler, vektörlerle ifade edilirler. Görüntü üzerinde işimize yarayacak gerekli bilgi olduğu kadar gereksiz birçok bilgide bulunmaktadır. Bu gibi gereksiz bilgiler kullanılan algoritma üzerinde gereğinden çok daha fazla işlem yüküne ve buna bağlı olarak performans kaybına yol açacaktır. Bu problemin giderilmesi için görüntü üzerindeki gereksiz bilgilerin elenip, ilgili algoritma tarafından kullanılabilecek boyutta ve anlamlı veri kümesine indirgenmesi gerekmektedir.

Veri seti üzerinde kullanılacak olan öznitelik çıkarma yöntemi algoritmanın performansını doğrudan etkileyeceğinden, uygun yöntemin seçilmesine dikkat etmek gerkir. Öznitelik çıkarma yöntemleri kendi içerisinde "düşük seviyeli" ve "yüksek seviyeli" yöntemler olmak üzere ikiye ayrılmaktadır [25]. Bu yöntemler ilerleyen başlık altında detaylı olarak açıklanacaktır.

3.1.3 Sınıflandırma

Görüntü üzerinde önişlem ve öznitelik çıkarımı işlemlerinin ardından sınıflandırma işlemi gerçekleştirilir. Sınıflandırma kısaca, hangi gruptan geldiği bilinmeyen bir verinin hangi gruba dahil edileceğini belirleme işlemi olarak açıklanabilir. Bu işlem için sınıflandırıcı, özellik matrisindeki değerlerden faydalanır ve bu değerlerin sınıflandırıcıya verilmesi sınıflandırma basamağı açısından önemli bir aşamadır. Bu basamakta öncelikle hangi sınıfa ait olduğu önceden bilinen öznitelik vektörlerinin oluşturduğu "veri kümeleri" bir eğitimden geçirilir. Bu eğitim sonucunda, bilinmeyen örüntüyü uygun sınıfa atamakta kullanılan bir karar mekanizması oluşturulur. Bu şekilde gerçekleştirilen sınıflandırma yaklaşımları "eğiticili sınıflandırma" olarak adlandırılır. Herhangi bir eğitim kümesi kullanılmadan gerçekleştirilen sınıflandırma ise "eğiticisiz sınıflandırma" olarak adlandırılır [27], [28].

Sınıflandırma algoritmaları istatistiksel bir karar yöntemi olan Bayes teoremi üzerine kuruludur. Bayes teoremi, rastlantısal bir bilginin incelenmesinde objektif bir bakış açısını esas alır ve bilginin aşamalarına odaklanır. Bu teorem bir rastlantı değişkeni için olasılık dağılımı içinde koşullu olasılıklar ile marjinal olasılıklar arasındaki ilişkiyi gösterir. Olasılık teorisi içinde incelenen bir olay olarak B olayına koşullu bir A olayı için olasılık değeri, A olayına koşullu olarak B olayı için olasılık değerinden farklıdır. Ancak bu iki birbirine ters koşulluluk arasında çok belirli bir ilişki vardır. Bayes teoremi bir stokastik süreç sırasında ortaya çıkan bir rastgele A olayı ile diğer bir rastgele B olayı için koşullu ve marjinal olasılıklar arasındaki ilişkidir.

$$\mathrm{P(A\backslash B)} = \frac{\mathrm{P(B\backslash A)P(A)}}{\mathrm{P(B)}} \tag{3.1}$$

Denklem üzerinde P(A): A için önsel (marjinal) olasılık; P(A|B): verilen B için A'nın koşullu olasılığı; P(B|A): verilen A için B'nin koşullu olasılığı; P(B): B olayı için önsel olasılık olarak ifade edilir [29].

Sınıflandırma algoritmaları arasında ilklerden sayılan bir diğer yöntem ise En Yakın Komşu olarak literatüre geçmiştir. Bu yöntemde sınıflandırılmak istenen örneğe, en yakın örneğin bulunması amaçlanır. Bütün örnekler n-boyutlu uzayda bir noktaya karşı düşürülür ve nesneler arası Öklid uzaklığı hesaplanır. Hesaplama sonucu öğrenilen fonksiyon ayrık değerli veya gerçel değerli olabilir. Ayrık değerli fonksiyonlarda K-Komşu Algoritması, xq örneğine en yakın k öğrenme örneğinde, en çok görülen sınıf değerini verir. Sürekli değerli fonksiyonlarda ise en yakın k öğrenme örneğinin ortalaması alınır. Yöntem kolay uygulanır ancak tanıma oranı çok yüksek değildir. Bunun nedeni olarak algoritmanın sadece mesafeyi temel alıyor olmasıdır ve eğitim setinde ilgili sınıf işareti olmaz ise hata oranının artmasının beklenmesi verilebilir. Sınıflandırma işleminde, öğrenme kümesindeki örneklerin, sınıflandırılmak istenen örneğe olan uzaklıklarına göre ağırlıklar atanır ve sonuçta daha yakın örneklerin ağırlığı daha fazla olur [30], [31].

$$w = \frac{1}{d(x_q, x_i)^2} \tag{3.2}$$

Denklem üzerinde w: ağırlık; d: Öklid uzaklığı; xi: öğrenme kümesindeki örnekler; xq: sınıflandırılmak istenen örnek olarak ifade edilmektedir.

Bayes Teoremi ve En Yakın Komşu sınıflandırıcı geleneksel sınıflandırıcılar olarak adlandırılabilir ve bu algoritmalara ek olarak Çok Değişkenli Gauss Modelleri, Karar Ağacı Sınıflandırıcı ve Fisher'in Doğrusal Sınıflandırıcıları örnek olarak verilebilir.

Bu algoritmalar günümüzde çok fazla tercih edilmemektedir. Günümüzde daha çok akıllı yöntemler olarak adlandırabileceğimiz Yapay Sinir Ağları (YSA) ve Destek Vektör Makinaları gibi sınıflandırıcılar tercih edilir. YSA, insan beyninin çalışma yapısını model alarak; beynin öğrenme, hatırlama genelleme yapma yolu ile yeni bilgiler türetebilme gibi temel işlevlerini gerçekleştirmek üzere geliştirilen mantıksal bir sınıflandırma çeşididir [28], [32]. Bu tez çalışmasında sınıflandırıcı olarak Destek Vektör Makinaları kullanılmıştır. Günümüzde yaygın olarak kullanılan ve başarısını ispatlamış sınıflandırıcı türlerinden biridir. Detaylı olarak Bölüm 5'te açıklanacaktır.

Sonuç olarak örüntü tanıma sistemlerini özetleyecek olursak süreç; veri işleme, kayıt ve test olmak üzere üç ana başlık altında toplanabilir. Veri işlemede ham veriler, ışıklandırma kaynaklı etkilerin giderimi, farklı ölçekteki nesnelerin aynı boya getirilmesi gibi birtakım önişlemlere tabi tutulur. Bir sonraki adımda öznitelik vektörleri çıkartılır ve bu özellik vektörleri gerektiği takdirde sadeleştirilerek kullanılır. Kayıt işlemi ise nesnelerin bir ya da birden fazla zaman diliminde, bir ya da daha fazla özelliklerinin ölçüldükten sonra bu bilgilerin sisteme işlenerek saklanması ve gerektiğinde kullanılması olarak açıklanabilir. Son olarak test işleminde ise öncelikle tanınması istenen nesnenin o andaki ölçümlerinden yola çıkılarak yeni bir kaydı oluşturulur. Daha sonra bu kayıt, önceden elde edilmiş kayıtlarla karşılaştırılarak nesnenin tespit edilmesi amaçlanır.

3.2 Öznitelik Çıkarma Yöntemleri

Öznitelik çıkarımını özetle görüntü üzerindeki anlamlı bilgilerin elde edilmesini sağlayan bir çeşit boyut indirgeme işlemidir. Kullanılacak olan anlamlı bilgilerin elde edilmesi "düşük seviyeli" ve "yüksek seviyeli" yöntemler olmak üzere iki ayrı ana başlık altında toplanmaktadır.

3.2.1 Düşük Seviyeli Öznitelik Çıkarma Yöntemleri

3.2.1.1 Kenar Belirleme Yöntemi

Görüntü üzerindeki piksel noktasındaki değişimlerin yoğun olduğu alanların bulunması kenar belirleme algoritmaları ile gerçekleştirilir. Bu tip algoritmalara Sobel, Canny, Prewitt ve Roberts kenar bulma algoritmaları örnek olarak verilebilir. Bu algoritmalar görüntü işlemenin temellerini oluşturmaktadır [33], [34].

3.2.1.2 Köşe Belirleme Yöntemi

İlk olarak Hans Peter Moravec tarafından kullanılan köşe belirleme yönteminin ana amacı; uygun seçilen bir pencere ile köşe noktaları etrafındaki komşuluklar için her yöndeki parlaklık farkından yararlanarak görüntü üzerindeki köşelerin bulunmasıdır [35]. Sonrasında Förstner; özilinti matrisi olarak bilinen, görüntü üzerinde x-y yönündeki türevlerinin çarpımlarından oluşan iki boyutlu matrisi köşe belirlemek için kullanmışlardır

[36]. Harris ve Stephens ise bu matrisin determinantını ve izini kullanarak farklı bir köşe ölçüsü önermişlerdir [37]. Moravec'in yönteminin yanı sıra Harris ve Stephens kaydırma merkezi etrafında analitik genişletme yaparak, olabilecek tüm küçük kaydırmalara imkan sağlamışlardır. Bu çalışmalar köşe belirleme algoritmalarının temellerini oluşturmaktadır. Günümüzde bu algoritmalar geliştirilerek bölgeyi morfolojik açıdan inceleyerek köşeleri parametrik modellerle ifade eden ve bu parametrelere uygunluğuna göre seçim yapan köşe belirleme yöntemleri de kullanılmaktadır [38].

3.2.1.3 Özel Bölge Belirleme Yöntemi

Görüntü üzerine uygulanan bir takım işlemler sonucunda nesnenin değişmeyen özel bölgelerin belirlenmesi olarak adlandırılır. Bu yöntemin amacı görüntü üzerindeki nesnenin; farklı geometrik ve fotometrik şartlarda elde edilen özel bölgelerinin, birbirleri ile eşleştirilebilmesini sağlamaktır. Özel bölge belirleme yöntemlerine LoG (The Laplacian-of-Gaussian) [39] , DoG (The Difference-of-Gausssian) [39] , DoH (Determinant of Hessian) [40] ve Maksimum Durağan Uç Bölgeler (MSER) [41] algoritmaları örnek olarak verilmektedir [42].

3.2.2 Yüksek Seviyeli Öznitelik Çıkarma Yöntemleri

3.2.2.1 Bağımsız Öznitelik Çıkarma Yöntemleri

Temel Bileşen Analizi (TBA), Bağımsız Bileşen Analizi (BBA), Doğrusal Diskriminant Analizi (DDA) ve Dalgacık Dönüşümü bağımsız öznitelik çıkarma yöntemleri olarak adlandırılmaktadır [25].

Temel Bileşen Analizi (Principal Components Analysis) boyut azaltmak ve görüntüler arasındaki temel farklılıkları çıkarmak için kullanılan bir yöntemdir. TBA karşılıklı ilişkisi olan verinin n-boyutlu uzay yerine daha düşük boyutlu uzayda önemli bilgi kaybı olmadan boyut azaltmanın en uygun yoludur. Azaltılmış alt uzayı kullanarak, veritabanı üzerindeki hesaplamalar daha verimli hale getirilebilir. Üretilen taslaklar genel olarak öz uzaylar olarak adlandırılır. Özvektörler özellik vektörlerinin oluşturulması için kullanılır. Bu özellik vektörleri daha sonra tanıma işleminde kullanılarak gelen örnek görüntünün en çok benzediği eğitim seti elemanı bulunur. TBA'nın üç temel amacı vardır:

- Verilerin boyutunu azaltmak
- Tahmin yapmak
- Veri setini bazı analizler için görüntülemek

TBA'dan türetilen Çekirdek Temel Bileşenler Analizi (ÇTBA) ise, doğrusal olmayan yöntem ile birlikte temel bileşen çıkarımı ve belirlenen görüntülerin pikselleri arasındaki ilişkiyi açıklamaktadır [43], [44].

Bağımsız Bileşen Analizi (Independent Component Analysis) normal dağılıma sahip olmayan çok değişkenli verilerin aralarındaki istatistiksel bağımlılığı en aza indirecek şekilde gösterimini sağlayarak, içindeki saklı bileşenleri bulmaya çalışır. Bu bileşenler verinin içerisindeki önemli bazı özellikleri ifade etmektedir [25].

Doğrusal Diskriminant Analizi (Linear Discriminant Analysis) çok değişkenli bir analiz türüdür. Gruplar arası fark yoktur anlamını taşıyan H_0 hipotezi red edildikten sonra, gruplar arası farkın olduğu sonucuna varılır. Bu farklılığın ana nedenleri Diskriminant Analizi tekniğiyle ortaya çıkartılır. Diskriminant Analizinin iki temel amacı vardır:

- Diskriminant fonksiyonlarını belirleyip, bu fonksiyonlar aracılığıyla gruplar arası ayrıma en fazla etki eden ayırıcı değişkenleri belirlemek,
- Hangi gruptan geldiği bilinmeyen bir birimin hangi gruba dahil edileceğini belirlemektir.

Birinci amaca yönelik olan Diskriminant Analizi, Betimsel Amaçlı Analiz, ikinci amaca yönelik olan Diskriminant Analizi Karar Amaçlı Analiz olarak adlandırılır [45].

Dalgacık Dönüşümü (Wavelet Transform) Fourier dönüşümünün durağan olmayan sinyallerdeki eksiklerini gidermek için geliştirilmiş dönüşüm yöntemidir [46], [47]. Bilindiği üzere Fourier Dönüşümünden sonra sinyalin sonlu veya sonsuz sayıdaki harmoniklerinin sinusoidal bileşenleri (frekans bileşenleri) elde edilebilir fakat harmoniklere ait zamansal bilgi elde edilemez. Bu yüzden Dalgacık Dönüşümü, Fourier Dönüşümünün frekans-zaman hassasiyetini ayarlanabilir yapmak amacıyla kullanılmaktadır [48].

3.2.2.2 Şekil Tabanlı Öznitelik Çıkarma Yöntemleri

Bu yöntemde görüntü üzerindeki geometrik şekillerin belirlenmesi amaçlanır. Şekil tabanlı öznitelik çıkarma yöntemleri arasında en çok kullan Hough dönüşümüdür. Hough dönüşümü ile düzgün doğrular, eğriler, çember ve elipsler, rastgele dağılımlı doğrusal olmayan düzensiz şekiller gibi geometrik cisimler belirlenebilmektedir [25]. Bu dönüşüm bir görüntü üzerine doğru belirleme için uygulandığında elde edilecek yeni görüntü üzerinde doğrusal olmayan nesneler bulunmaz. Hough dönüşümü kenar bilgisi elde edilen gri ölçekli görüntüler üzerinde matematiksel olarak aşağıdaki doğru denklemi ile hesaplanması gerçekleştirilir.

$$a.x_n = y_n - b \tag{3.3}$$

Denklem üzerindeki bazı hesaplamalar sonucunda elde edilen a ve b dik doğrularının sonsuza gitme durumu göz önünde bulundurulursa; kartezyen koordinatlar yerine aşağıdaki denklem ile kutupsal koordinatların kullanılması gerekmektedir [49].

$$q = x.\cos(\theta) + y.sin(\theta) \tag{3.4}$$

3.2.2.3 Şekil Eşleştirme Tabanlı Öznitelik Çıkarma Yöntemleri

Bu yöntemin en önemli özelliği görüntü üzerindeki nesnenin; değişen öteleme, dönme ve ölçekleme gibi durumlar karşısında belirlenmesini sağlamaktır. Bazı şekil eşleştirme tabanlı yöntemler; Ölçekten Bağımsız Öznitelik Dönüşümü (Scale Invariant Feature Transform – SIFT), Hızlandırılmış Gürbüz Öznitelikler (Speeded Up Robust Features – SURF), Gradyan Konum ve Yön Histogramı (Gradient Location and Orientation Histogram – GLOH) olarak bilinmektedir [25].

SIFT algoritmasının en önemli özelliği olarak tanıma işlemi; nesnenin boyutu, açısı ve ortamın ışık koşullarından bağımsız olarak gerçekleştirilebilir. Algoritma hesaplanırken ilk olarak, ölçekli uzaydaki uç değer noktaların tespit edilerek anahtar noktaların bulunması hedeflenir. İkinci aşamada ise yönlerin belirlenmesi ve anahtar noktalar için tanımlayıcı vektörlerin oluşturulması hedeflenir [39].

SURF ise iki boyutlu Haar dalgacıkları toplamını temel alarak, integral görüntüleri kullanan bir şekil eşleştirme tabanlı algoritmadır. İntegral görüntüleri, görüntü üzerinde belirli bir pencere içerisinde kalan piksel değerleri toplamının hızlı bir şekilde

hesaplanmasını gerçekleştirmektedir. SURF algoritmasının temel amacı Haar dalgacıklarını kullanarak; Hessian matrislerini görüntü özniteliklerini belirlemek için yaklaşık determinantları hesaplamaktır [25], [50].

SIFT algoritmasınan benzerliği ile bilinen GLOH yönteminin temelde 2 farklı özelliği bulunmaktadır. Bu özellikler; kullanılan ızgara konumunun değiştirilebilir olmasından dolayı öznitelik vektör boyutunun artırılması ve elde edilen bu yüksek boyutlu öznitelik vektörlerinin TBA yöntemi ile boyutunun indirgenmesidir. Bu bilgiler doğrultusunda GLOH; bir ilgi noktası etrafındaki yerel konuma bağımlı türevlerin yön histogramı olarak tanımlanmaktadır. Aynı zamanda GLOH, SIFT yönteminin sağlamlık ve ayırt edici özelliğini artırmak için tasarlanmış özniteliklerini kullanmaktadır [25], [51].

3.2.2.4 Türev Tabanlı Öznitelik Çıkarma Yöntemleri

Türev tabanlı öznitelik çıkarma yöntemlerine örnek olarak; HOG (Histogram of Oriented Gradients), CoHOG (Co-Occurence Histogram of Oriented Gradients), Color CoHOG (Color Co- Occurence Histogram of Oriented Gradients), CoHED (Co-occurence Histograms of pairs of Edge orientations and color Differences), CoHD (Co-Occurrence Histograms of Color Differences) algoritmaları verilebilir.

HOG algoritması görüntü üzerinde bölgesel olarak piksellerin yön ve büyüklük değerlerini kullanarak histogram oluşturmayı amaçlar. Bu histogramlar, görüntünün yerel bir bölgesindeki türevlerin yönelimlerinin yoğunluğunu içermektedir. Bu yöntem tezde kullanıldığı için Bölüm 4'te detaylı olarak açıklanacaktır.

CoHOG algoritması HOG yönteminin türevi olarak gradyan yönelimleri arasındaki uzaysal ilişkiyi ifade etmektedir. HOG yöntemi görüntüyü tekli gradyan yönelimleri olarak ifade etmektedir. Bunun sonucunda ise ilgili görüntünün dokusu istenildiği gibi ifade edilememektedir. CoHOG ise görüntüyü gradyan çiftleri olarak ifade edebildiğinden; CoHOG öznitelik tanımlayıcıları nesnenin şeklini HOG'dan algoritmasından daha detaylı olarak ifade edebilmektedir [25], [52].

RGB uzayındaki görüntünün gri ölçekli görüntüye dönüştürülmesi sırasında yaşanan bilgi kaybını ortadan kaldırma amacıyla Color-CoHOG yöntemi önerilmiştir. Color-CoHOG yönteminde HOG ve CoHOG'dan farklı olarak öznitelik vektörleri hesaplanırken

görüntünün renk bilgisi de kullanılmaktadır. Öncelikle renkli görüntüdeki kenar yönelimleri hesaplanır; sonrasında ise önplanla arka plan arasındaki fark dikkate alınarak renk eşleşmesinin sonuçları kullanılır [25].

CoHED yöntemi diğerlerinden farklı olarak görüntünün kenar yönelimleri ile kenardaki renk değişimi arasındaki ilişkiye temel alır. CoHD yönteminde ise görüntü içerisinde bulunan bir çizgi üzerindeki üç pikselde meydana gelen renk değişimi kullanılmaktadır. Böylece CoHD ve CoHOG yöntemleri görüntü üzerinde sadece doku bilgisini kullanarak öznitelik çıkarımı yapmaktadır [25].

BÖLÜM 4

YÖNLÜ TÜREVLERİN HİSTOGRAMI

Yönlü türevlerin histogramı (Histogram of Oriented Gradients – HOG), görüntüdeki piksellerin yönelim ve büyüklük değerlerinin karakteristiği olarak tanımlanan gradyan temelli bir öznitelik belirleme yöntemidir. Bu algoritma ilk olarak Dalal ve Shashua tarafından nesne tanıma uygulamaları üzerinde kullanılmıştır [53], [54].

HOG algoritmasının temel amacı bir görüntü üzerinde lokal histogram dizilerini belirlemek olarak adlandırılabilir. Belirli bir bölge için hesaplanan türevlerin, belirlenen yönlerde oluşturduğu yoğunluk dağılımı görüntü üzerinde hücrelerle ifade edilmektedir. Bu türevlerin hesaplanmasında piksel parlaklığı bilgisinden yararlanılır. Bu algoritmada öznitelik belirleme aşağıdaki işlem basamakları üzerinden gerçekleştirilir.

- Türev büyüklük ve yönlerinin bulunması
- Her bir hücre için gradyan histogramlarının oluşturulması
- Hücre histogramlarının bloklar halinde birbirleri ile normalize edilmesi

Genellikle görüntüler RGB renk uzayındadır. RGB renk uzayı İngilizcedeki Red, Green, Blue (Kırmızı, Yeşil, Mavi) kelimelerinin baş harflerinden ismini alan bir renk uzayıdır. Bu üç renk %100 oranında karıştırıldığında beyaz ve %0 oranında karıştırıldığında ise siyah elde edilir. Bir görüntü üzerinde HOG işlem basamaklarının uygulanabilmesi için öncelikle görüntü gri ölçekli yeni bir görüntüye dönüştürülür. Bu işlem yaklaşık olarak aşağıdaki denklem üzerinden gerçekleştirilir [55].

$$Y = 0.299R + 0.587G + 0.114B \tag{4.1}$$

Gri seviye görüntüler, pikselleri belli aralıklar arasındaki parlaklık değerlerini gösteren matrislerden oluşmaktadır ve uint8, uint16, int16, single veya double veri tipinde olabilirler. Örneğin unit8 veri tipinde gri seviye bir görüntü 256 farklı parlaklık değerine sahiptir ve görüntüdeki her bir piksel 0 ile 255 arasında değişen bir tamsayı değer almaktadır.

Gri seviyeye dönüştürülen görüntü küçük boyutlu yerel parçacıklara bölünerek bu parçacıkların türevlerinin yönelimlerinin dağılımı hesaplanır ve bu değerler görüntüdeki nesneleri tanımlamakta kullanılır. Bir görüntünün HOG değerlerinin hesaplanmasında ilk olarak görüntü üzerine yatay ve dikey Sobel filtreleri uygulanarak, yatay ve düşey kenar haritaları I_x ve I_y hesaplanır [56].

$$I_x = I * S_y \tag{4.2}$$

$$I_y = I * S_d \tag{4.3}$$

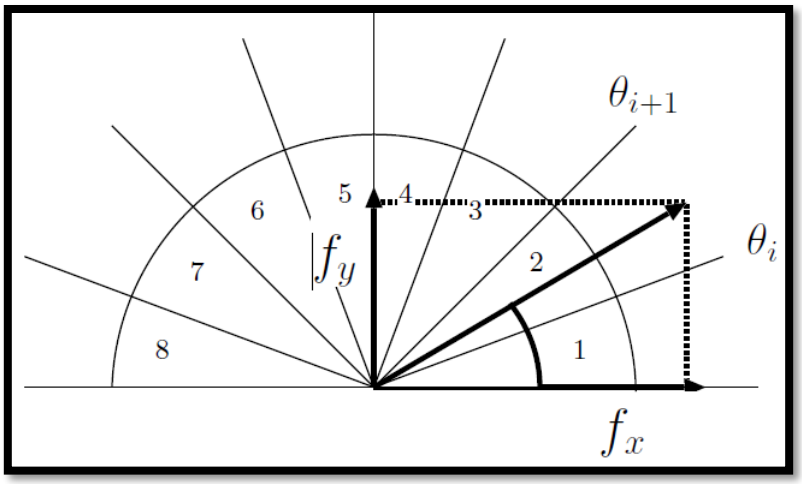
$$I_x(x,y) = i(x+1,y) - i(x-1,y) \tag{4.4}$$

$$I_y(x,y) = i(x,y+1) - i(x,y-1) \tag{4.5}$$

Hesaplanan Ix ve Iy görüntüleri kullanılarak, türev (G) ve bu türevlerin yönelim açıları (θ) aşağıdaki şekilde hesaplanır [57].

$$|G| = \sqrt{I_x^2 + I_y^2} \tag{4.6}$$

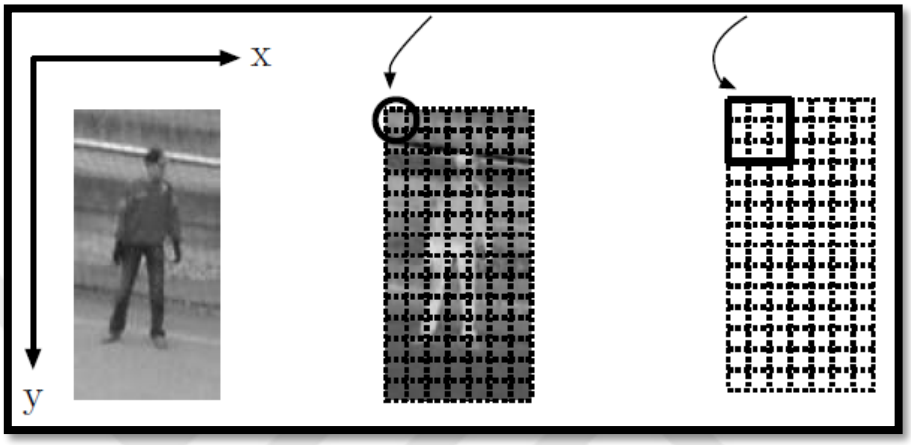
$$\Theta(x,y) = arctan\frac{I_x(x,y)}{I_y(x,y)} \tag{4.7}$$



Şekil 4. 1 Türev yön bilgileri [57]

Bulunan türev (G) ve bu türevlerin yönelim açıları (θ) sonrasında görüntü belirlenen piksel değerlerinde hücrelere ayrılarak histogram değerleri hesaplanır. Herbir hücre belirlenen boyutlar ölçüsünde türev (G) ve yönelim açısı (Q) bilgisini içermektedir. Bu hesaplamada türevlerin yönelim açıları uygulamaya göre 0 – 180 derece ya da 0 – 360 derece arasında eşit bölgelere ayrılır [57].

Giriş Görüntüsü 1 Hücre=5x5 Piksel 1 Blok = 3x3 Hücre



Şekil 4. 2 HOG öznitelik çıkarımında hücre ve blok kullanımı [58]

Histogram hesaplanırken türev büyüklüğü dağıtımında genellikle lineer interpolasyon kullanılmaktadır. Bu yöntem aşağıdaki denklem üzerinden açıklanacak olursa; $h(n+1)$ ve $h(n+1)$ histogramda komşu iki türev yön bilgisi, α büyüklük dağıtım çarpanı, b ise türev yön bölgelerinin açı değeridir [57].

$$h(n) = h(n) + (1-\alpha)m(x,y) \tag{4.8}$$

$$h(n+1) = h(n+1) + \alpha m(x,y) \tag{4.9}$$

$$\alpha = \frac{\theta(x,y) - \theta_n}{\mathrm{b}} \tag{4.10}$$

Elde edilen hücre histogramları bloklar halinde gruplanıp, blok içerisindeki hücreler birbirleri ile normalize edilir. Bazı normalizasyon denklemleri aşağıdaki gibidir.

$$L_1 - norm: f = \frac{v}{(\|v\|_1 + e)} \tag{4.11}$$

$$L_1 - sqrt: f = \sqrt{\frac{v}{(\|v\|_1 + e)}} \tag{4.12}$$

$$L_2 - norm: f = \frac{v}{\sqrt{(\|v\|_2^2 + e^2)}} \quad (4.13)$$

$$L_2 - hys: f = \frac{v}{\sqrt{(\|v\|_2^2 + e^2)}}, f_{max} = 0.2 \quad (4.14)$$

$$\|v\|_n = \sqrt[n]{\sum i |x_i|^n} \quad (4.15)$$

Bu normalizasyon yöntemlerinde histogram değerleri arka arkaya eklenerek v vektörü oluşturularak $\|v\|_n$ ile n. dereceden norm ifade edilmiştir. Sonuçta f vektörü, v vektörünün normalize edilmiş hali olarak elde edilir. Normalizasyon bloğu histogram hücreleri üzerinde gezdirilerek; mxn bir görüntü kxk blok ile normalize edildiğinde ortaya çıkan HOG tanımlayıcısı aşağıdaki denklem üzerinden hesaplanabilir. Burada p türev yön bölgesi sayısı olarak ifade edilmiştir [57].

$$Ht = p.k^2.\big(m - (k-1)\big).(n - (k-1)) \quad (4.16)$$

BÖLÜM 5

DESTEK VEKTÖR MAKİNELERİ

İlk olarak 1963 yılında Vapnik tarafından önerilen Destek Vektör Makineleri (DVM), istatistiksel öğrenme teorisine dayalı bir eğitimli sınıflandırma yöntemidir [58], [59], [60]. Temel olarak üç aşamadan oluşmaktadır:

i. Eğitim hücrelerinin öznitelik vektörü olarak ifade edilmesi,

ii. Öznitelik vektörlerinin çekirdek fonksiyonları kullanılarak öznitelik uzayına eşlenmesi,

iii. Sınıfları en uygun şekilde ayıran n-boyutlu hiper düzlemin oluşturulması.

En uygun hiper düzlemin oluşturulup veri kümesinin birbirinden ayrılması doğrusal ve doğrusal olmayan ayırma şeklinde ikiye ayrılmaktadır [58], [59], [60].

5.1 Doğrusal Destek Vektör Makineleri

Doğrusal Destek Vektör Makineleri, veri kümesini doğrusal olarak ayıran bir sınır olarak nitelendirilir. Doğrusal ayırma birbirini takip eden iki formdan oluşmaktadır.

Birincil form:

$$L_P = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} \alpha_i y_i (x_i \cdot w + b) + \sum_{i=1}^{n} \alpha_i \qquad (5.1)$$

Birincil form L_P denklemiyle ifade edilmektedir fakat w'nun normu bir karekök içerdiğinden optimizasyon problemi oluşturmaktadır.

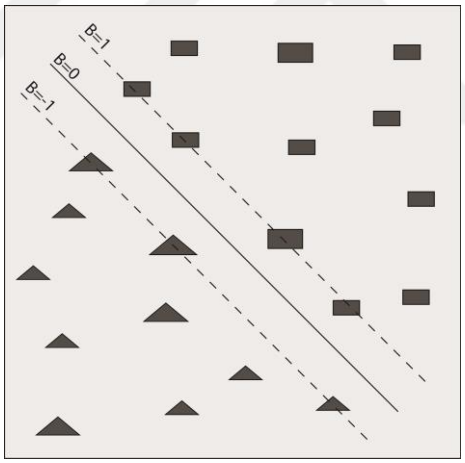
Bu sebepten$\|w\|^2 = \omega \cdot \omega$ ve $\sum_{i=1}^{n} \alpha_i y_i x_i$ kullanmak yerine aşağıdaki ikincil formu kullanmak optimizasyon problemini azaltmaktadır [58], [61].

İkincil form:

$$L_D = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \quad (5.2)$$

L_D denkleminde $i = 1, 2, 3, \dots, n$'e kadar, b minimizasyonundan gelen kısıt ile $\sum_{i=1}^{n} \alpha_i y_i = 0$ ve $\alpha_i \geq 0$ olacak şekilde α_i'ler Lagrange katsayıları olarak ifade edilir. Burada çekirdek $K(x_i, x_j) = (x_i \cdot x_j)$ olarak tanımlanmaktadır. w ise α katsayıları sayesinde $w = \sum_i \alpha_i y_i x_i$ ifadesiyle hesaplanır [58], [59].

Hiper düzlemin oluşturulması işlemi Şekil 2.1 üzerinden açıklanacak olursa, üçgen ve karelerden oluşan bir veri kümesi öncelikle her iki gruba da yakın ve paralel iki sınır çizgisiyle $(B = -1$ ve $B = 1)$ birbirinden ayrılır. Daha sonra bu çizgiler birbirlerine yaklaştırılarak ortak sınır çizgisi $(B = 0)$ üretilir. $B = -1$ ve $B = 1$ çizgileri üzerindeki öğrenme verileri ise destek vektörleri olarak adlandırılmaktadır.



Şekil 5. 1 Doğrusal DVM

5.2 Doğrusal Olmayan Destek Vektör Makineleri

1963 yılında Vladimir N. Vapnik tarafından önerilen orijinal optimum hiper düzlem algoritması doğrusal sınıflandırıcı olduktan sonra 1992 yılında Bernhard E. Boser, Isabelle M. Guyon ve Vapnik maksimum toleranslı hiper düzlemleri, çekirdek fonksiyonuna başvurarak doğrusal olmayan sınıflandırıcılar oluşturmak için bir yol önermişlerdir ve böylece doğrusal olmayan sınıflandırma ortaya çıkmıştır.

Doğrusal olmayan bir eşlemleme ile n-boyutlu veri kümesi, m > n olacak şekilde m-boyutlu yeni bir veri kümesine dönüştürülebilir. Aynı zamanda yüksek boyutta doğrusal sınıflandırma işlemi yapılabilir. Uygun bir dönüşüm ile her zamansal veri bir hiper düzlem ile iki sınıfa ayrılabilir ve sonuçta hiper düzleme en yakın öğrenme verileri destek vektörleri olarak adlandırılır.

Doğrusal olmayan DVM özellik uzayındaki optimum hiper düzlemi oluşturduktan sonra, bilinmeyen bir örneği sınıflandırma işlemini aşağıdaki denklem üzerinden gerçekleştirir [58], [59].

$$f(y) = sgn(\sum_{i=1}^{N} \alpha_i y_i K(x_i, y) + b) \tag{5.3}$$

Bu denklemde $i = 1, 2, 3, \dots, n$'e kadar $\sum_{i=1}^{N} \alpha_i y_i = 0$ şartını sağlayan α_i'ler Lagrange katsayıları olarak ifade edilir ve $\alpha_i \geq 0$ dır. $\{y_i | y_i \epsilon \{-1, +1\}\}_{i=1}^{N}$ olup y_i'ler, eğitim modellerinin $(\{x_i | x_i \epsilon R^N\}_{i=1}^{N})$ sınıf değerleridir. $K(x_i, x)$ özellik uzayında bir iç çarpımı tanımlayan pozitif ve simetrik bir çekirdek fonksiyonu, b ise sınıflandırıcının çıkışına etki eden ön değeri temsil etmektedir. Çekirdek fonksiyonunun da kendi içerisinde farklı türleri vardır [60]. Bunlar şu şekildedir:

- Polinom temelli fonksiyon (homojen):

$$K(x_i, x_j) = (x_i \cdot x_j)^d \tag{5.4}$$

- Polinom temelli fonksiyon (homojen olmayan):

$$K(x_i, x_j) = (x_i \cdot x_j + 1)^d \tag{5.5}$$

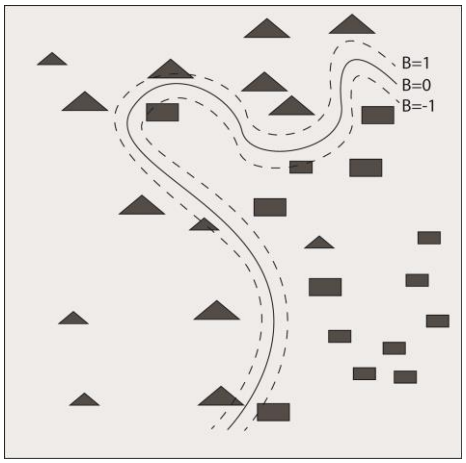
- Gauss radyal temelli fonksiyon:

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2), \gamma > 0 \text{ için} \tag{5.6}$$

- Hiperbolik tanjant:

$$K(x_i, x_j) = \tanh(\kappa x_i \cdot x_j + c), \kappa > 0 \text{ ve } c < 0 \text{ için} \tag{5.7}$$

Sonuç olarak doğrusal ve doğrusal olmayan formlar birbirinden farklı olsa da ayırma işlemlerindeki mantık ikisinde de aynıdır. Öncelikle her iki gruba da yakın ve paralel iki sınır çizgisi $B = -1$ ve $B = 1$ üretilir. Daha sonra bu çizgiler birbirlerine yaklaştırılarak ortak sınır çizgisi $B = 0$ oluşturulur.



Şekil 5. 2 Doğrusal olmayan DVM

5.3 Literatürde Destek Vektör Makineleri Kullanılan Çalışmalar

Yapılan literatür incelemesinde, HOG + DVM kullanılan yayaların yürüyüş rotalarının belirlenmesine ilişkin yapılan çalışmaların sonucunda farklı başarı oranlarına ulaşıldığı görülmüştür. Çizelge 1.1'de varılan sonuçlar karşılaştırılmaktadır. Tabloda kullanılan çeşitli veri tabanları üzerinde farklı başarı oranları elde edildiği görülmektedir. Hangi yöntemin daha başarılı olduğu konusu kullanılan veri tabanının yapısının karmaşıklığı ile birlikte değerlendirilmelidir.

Çizelge 5. 1 DVM Kullanılarak yapılan çalışmalar ve başarı oranları

YÖNTEM ADI	REFERANS NO	VERİ TABANI	BAŞARI ORANI
HOG (MCHOG) + SVM	[1]	-	%89,7
HOG + SVM	[4]	Daimler	%97,50
HOG (MCHOG) + SVM	[7]	-	%99

BÖLÜM 6

GÖRÜNTÜLERDEKİ YAYALARIN TESPİT EDİLMESİ

Sabit olmayan kameralar ile alınan görüntüler üzerinden istenen nesneyi tespit etmek örüntü tanıma alanı için oldukça zor bir problemdir. Sabit kameralarda ise görüntü açısının ve çözünürlüğünün değişmemesinden kaynaklı, görüntü üzerinden istenilen nesnelerin çıkarılması daha kolay bir işlemdir. Bu bölümde CASIA Yürüyüş Veritabanı içerisindeki, her bir kişiye ait 90 derecelik sabit görüş açısından elde edilen videolar üzerinden yayaların elde edilmesi için bazı işlemler gerçekleştirilmiştir. Laboratuvar ortamında oluşturulmuş görüntü üzerinde yayaların tespit edilmesi için uygun olan yöntemin belirlenmesi ve rota belirleme algoritması üzerinde kullanılması amaçlanmıştır.

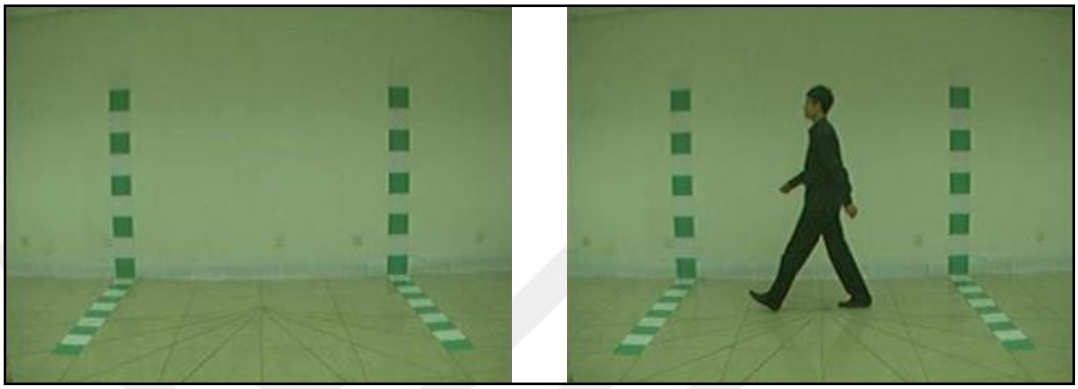
Sabit görüş açısına sahip kameralardan elde edilen görüntüler üzerinde istenilen nesnenin ayrıştırılması için arka plan çıkarma teknikleri kullanılmaktadır. Bu yöntem literatürde yaygın olarak kullanılan ve iyi sonuçlar üreten bir yaklaşımdır. Burada temel amaç kameranın görüş açısındaki hareket etmeyen kısımların alınarak sabit arka plan görüntüsünü elde edilmesi ve sonrasında hareketli nesnelerin bölütlenmesine dayanmaktadır.

Bölütleme işlemi için geliştirilen birçok farklı arka plan ayrıştırma yöntemi bulunmaktadır. Bunların hepsi temelde ardarda iki çerçevenin birbirinden çıkarılması esasına dayanmaktadır. Eigen ve Medyan arka plan çıkarma algoritmaları çerçeveler arası fark işlemini kullanmaktadır [62], [63]. Bu algoritmalarda oluşturulan arka plan belirli zaman aralıkları ile güncellenirken; Gauss karışımı arka plan çıkarma algoritması

ise zamanla değişen çevresel farklılıklara uyum sağlamaktadır [64]. Bu algoritmanın en önemli avantajı işlemlerin pikseller üzerinden yapılmasıdır.

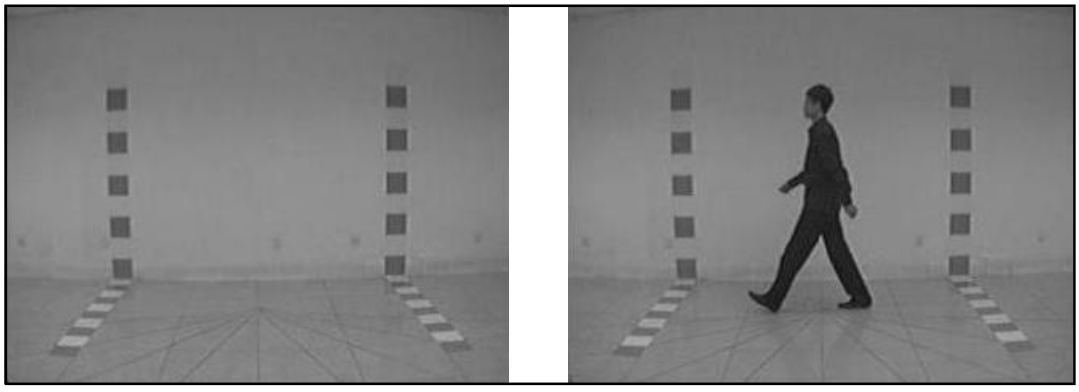
6.1 Arka Plan Çıkarma Tekniği Kullanarak Görüntüdeki Yayaların Tespit Edilmesi

İlk olarak sabit arka plan ve çıkartılacak yayaya ait görüntü CASIA veri tabanına ait bir videodan elde edilir. Elde edilen bu görüntüler RGB renk uzayındadır.



Şekil 6. 1 Sabit arka plan ve çıkartılacak yaya görüntüsü

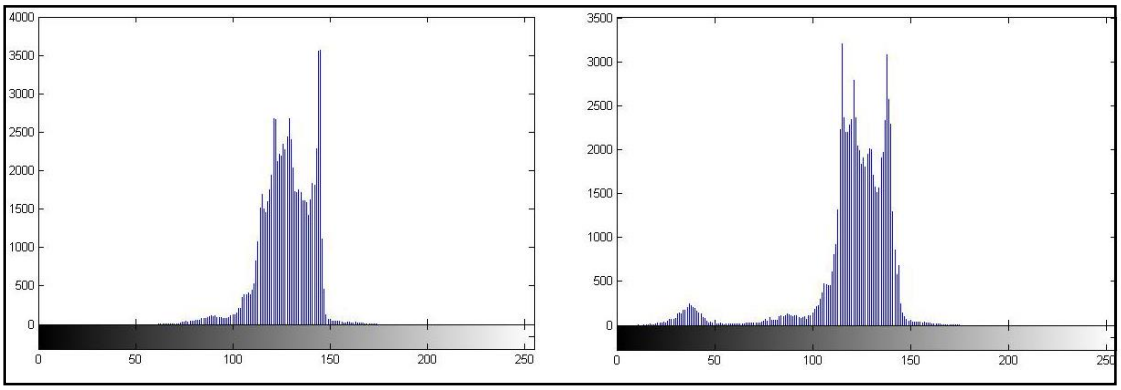
RGB uzayındaki görüntü uygulanacak işlemler açısından uygun olmadığından görüntü ikili (siyah-beyaz) seviyeye indirgemek gerekecektir. Bu işlem için görüntü öncelikle gri seviyeye dönüştürülür. Burada unit8 veri tipi ölçeğinde bir gri seviye dönüşümü gerçekleştirilerek, görüntüdeki her bir piksel 0 ile 255 arasında değişen bir tamsayı aralığında olacaktır.



Şekil 6. 2 Gri seviye arka plan ve çıkartılacak yaya görüntüsü

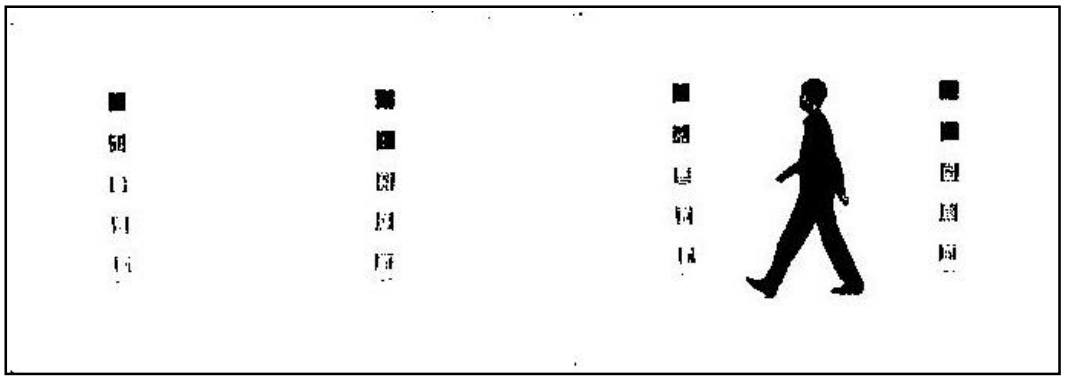
Bu aralık görüntüdeki renk değerlerinin yoğunluğunu gösteren histogram grafiği üzerinde gösterilebilir. Histogramda yatay eksen gri değer aralığını, düşey eksen ise her bir aralıktaki piksel sayısını göstermektedir. Böylelikle görüntünün kontrastına ilişkin

bilgiyi de elde etmek mümkün olmaktadır. Kontrast ise bir görüntüdeki en parlak bölüm ile en karanlık bölüm arasındaki fark veya zıtlık olarak ifade edilebilir.



Şekil 6. 3 Arka plan ve çıkartılacak yayaya ait histogram

Gri seviye görüntüyü ikili görüntü biçime dönüştürmek için histogram üzerinde bir eşik değeri belirlenmelidir. Bu eşik değerin üzerindeki renkler beyaza (1 değerine), altındaki renkler ise siyaha (0 değerine) dönüşmektedir. Ancak tüm görüntüler aynı renk dağılımına sahip olmayabilir ve sabit bir eşik değeri tüm görüntüler üzerinde kabul edilebilir sonuçlar üretemeyebilir. Dolayısıyla eşik değerin, her bir görüntünün renk dağılımına uygun olarak belirlenmesi gerekmektedir. Bu işlem için Otsu eşik belirleme metodu kullanılmaktadır [65]. Yöntem gri seviye görüntüler üzerinde işlem gerçekleştirerek sadece renk seviyelerinin görüntü üzerinde kaç defa bulunduğunu belirler. Bu yüzden önce görüntünün renk histogramı hesaplanır ve tüm işlemler histogram dizisi üzerinde gerçekleştirilir.



Şekil 6. 4 Otsu eşikleme uygulama

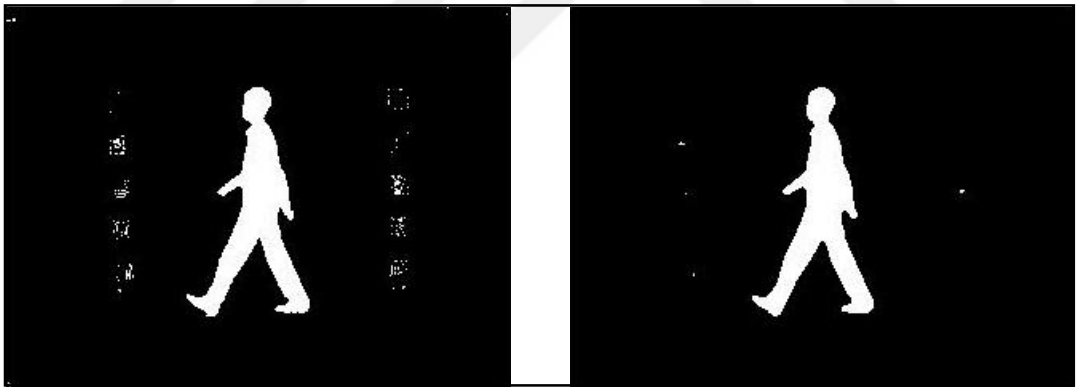
Eşikleme işlemi sonrasında ikili seviyeye indirgenen görüntü 0 ve 1'den oluşan iki parlaklık değerine sahip olur ve yayaya görüntüsü, sabit arka plan görüntüsünden çıkartılarak kişiye ait siluet elde edilir. Bu aşamada bazı istenmeyen bozucu bileşenler de

görüntüye dahil olur. Bu bileşenler gürültü olarak adlandırılır ve etkisini en aza indirgemek için çeşitli filtreler, morfolojik işlemler uygulanır.



Şekil 6. 5 Arka plan çıkartılmış görüntü

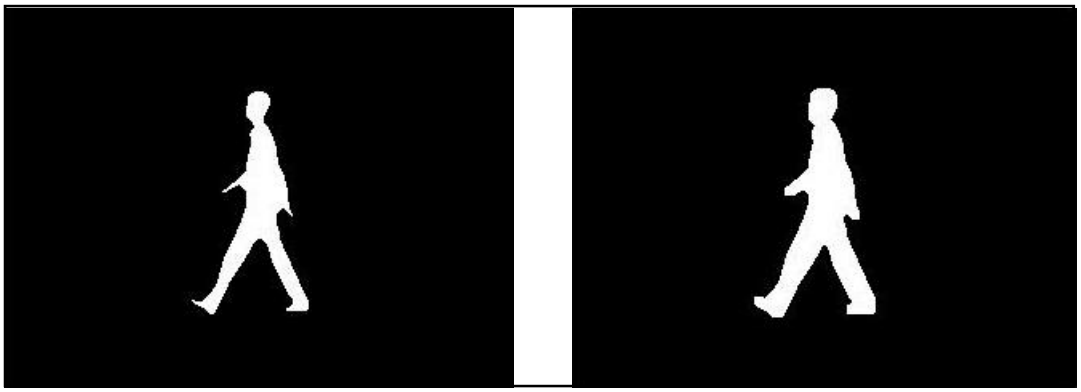
Bu görüntü üzerinde, ilk olarak siluetteki boşlukların tanımlanan çeşitli yapısal elemanlar yardımıyla doldurulup birbirinden ayrılmış olan parçaların tekrar birleştirilmesi işlemi gerçekleştirilir. Daha sonra Medyan filtre kullanılarak birbirlerinden belirgin şekilde ayrılan pikseller temizlenir. Medyan filtre komşu piksellerin kendi içerisinde sıralanarak ortanca değerinin alınması esasına dayanır [65].



Şekil 6. 6 Boşlukları doldurulan ve Medyan filtre uygulanan görüntü

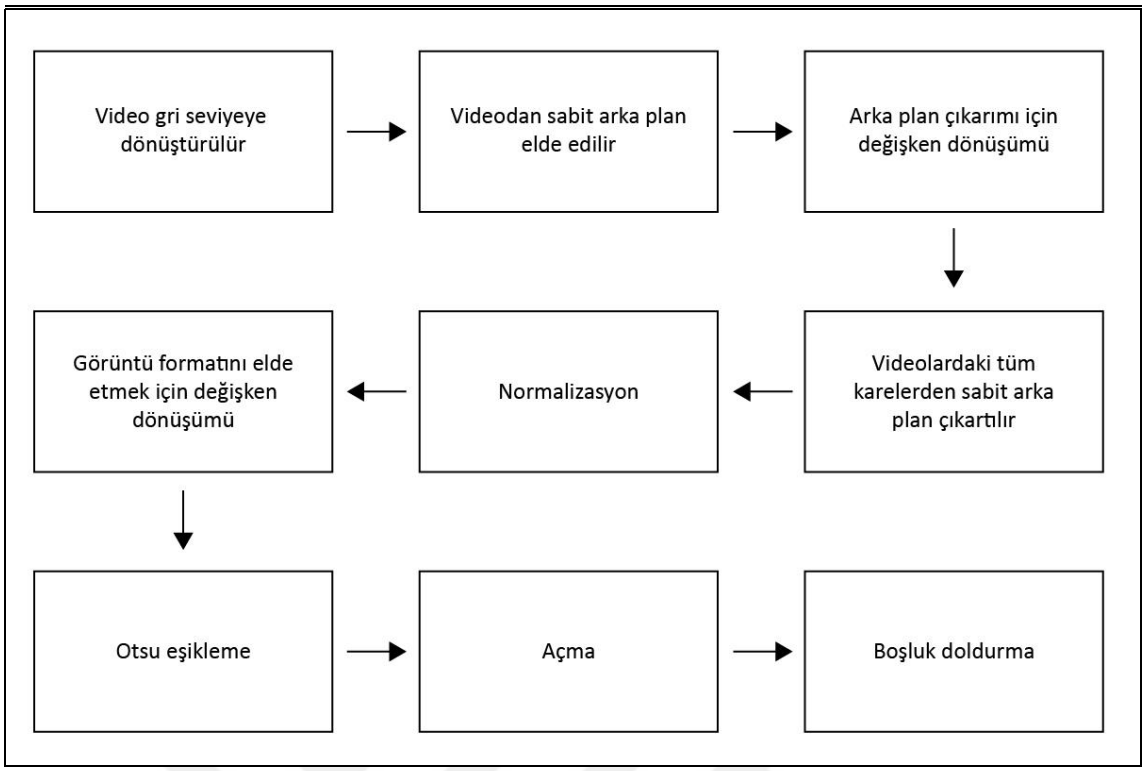
Medyan filtreden geçirilen görüntü bir disk yapısal elemanı kullanılarak aşındırma işlemine maruz bırakılır. Aşındırma işlemi ikili görüntüdeki nesneyi küçültmeye ya da inceltmeye yarayan morfolojik bir işlemdir. Burada yapısal eleman görüntü üzerinde piksel piksel dolaştırılarak merkez pikseli “1” değeri ile karşılaştığında, yapısal eleman içerisindeki piksellerin durumuna bakılır. Eğer yapısal eleman içerisindeki “1” olan piksellerden herhangi biri altında görüntüye ait “0” değeri varsa yapısal elemanın diğer “1”lerin altındakilerle beraber bu piksel “0” a dönüştürülür. Aşındırma işlemi uygulanan

görüntüde istenmeyen kayıplar oluşabilir. Bu durum açma işlemi ile düzeltilebilir. Açma işlemi ikili görüntüdeki nesneyi büyütmeye ya da kalınlaştırmaya yarayan bir morfolojik işlemdir ve aşındırma işleminin tersi olarak düşünülebilir.



Şekil 6. 7 Aşındırma ve Açma işlemi

Açma işleminden sonra görüntü üzerinde eşiklemeden kaynaklı oluşan gürültü giderilmiştir fakat bu işlemler görüntü dizisi üzerine uygulandığında yeterli sonuçlar üretmeyebilir. Tek bir görüntü üzerine uygulanan işlemlerde gürültünün istenilen ölçüde iyileştirildiği fakat bu işlemler görüntü dizisi üzerindeki diğer görüntüler için olumlu sonuçlar vermediği gözlenmiştir. Bunun sebebinin değişken dönüşümü ve normalizasyon işlemlerinden kaynaklı olduğu çalışma sürecinde belirlenmiş ve video üzerine uygulanabilecek ön işlemler Şekil 6. 8'de gösterilen işlem sırasını takip ekmektedir.



Şekil 6. 8 Ön işlem basamakları

MATLAB'de bir görüntü RGB uzayından gri seviye görüntüye indirgendiğinde "uint8" değişken tipi ile ifade edilmektedir. Bu veri tipi 8 bitlik 0 ile 255 arasındaki işaretsiz tamsayılardan oluşmaktadır. Değerleri pozitif olmalıdır ve genelde grafik üzerinden bir işlem yapılacaksa tercih edilmesi gereken bir veri tipidir.

Bu veri tipi matematiksel arka plan çıkarımı için dar bir ölçeğe sahiptir ve bu işlem için veri tipi "double" değişken tipine dönüştürülmesi gerekmektedir. Bu veri tipi daha geniş bir ölçeğe sahip olduğundan arka plan çıkartma işlemi sırasında oluşabilecek gürültü minimum seviyelerdedir.

Arka plan çıkartma işlemi tamamlandıktan sonra Otsu eşik belirleme işlemi için değişken tipinin tekrar "uint8" olarak ifade edilmesi gerekmektedir. Fakat bu değişken dönüşümü öncesinde "double" tipinde olan ve geniş bir ölçeğe sahip olan değerlerin, "uint8" formatında dar bir skalada ifade edilebilmesi için değerlerin 0 ile 255 arasında normalizasyon işleminin yapılması gerekmektedir.

Normalizasyon işlemi sonrasında görüntü tekrar "uint8" değişken tipine dönüştürülüp Otsu eşikleme uygulandığında gürültünün yok denecek kadar az seviyede olduğu

gözlenmiş ve herhangi bir filtreleme işlemine ihtiyaç olmadan sadece açma ve boşluk doldurma işlemleri kullanılarak Şekil 6. 9'da ki siluet dizisi elde edilmiştir.



Şekil 6. 9 İşlenmiş yürüme siluet dizisi

Bu bölümde uygulanan işlemler sonucunda, çalışmada sabit görüş açısına sahip kameralardan elde edilen görüntüler kullanılacak ise bu işlemlerin başarılı sonuçlar üreteceği kararına varılmıştır. Eğer görüntü sabit görüş açısına sahip kameradan elde edilmemiş ise bu yöntemler kullanılmamalıdır. Otonom araçlar üzerindeki kameralardan elde edilen görüntülerde sabit arka plan çıkarımı gerçekleştirilemeyeceğinden; yaya tespiti için farklı algoritmaların kullanılması gerekmektedir. Bu algoritmalar rota belirleme çalışması kapsamında Bölüm 7 içerisinde açıklanacaktır.

BÖLÜM 7

HOG VE DVM YÖNTEMLERİ KULLANARAK YAYALARIN YÜRÜYÜŞ ROTALARININ BELİRLENMESİ

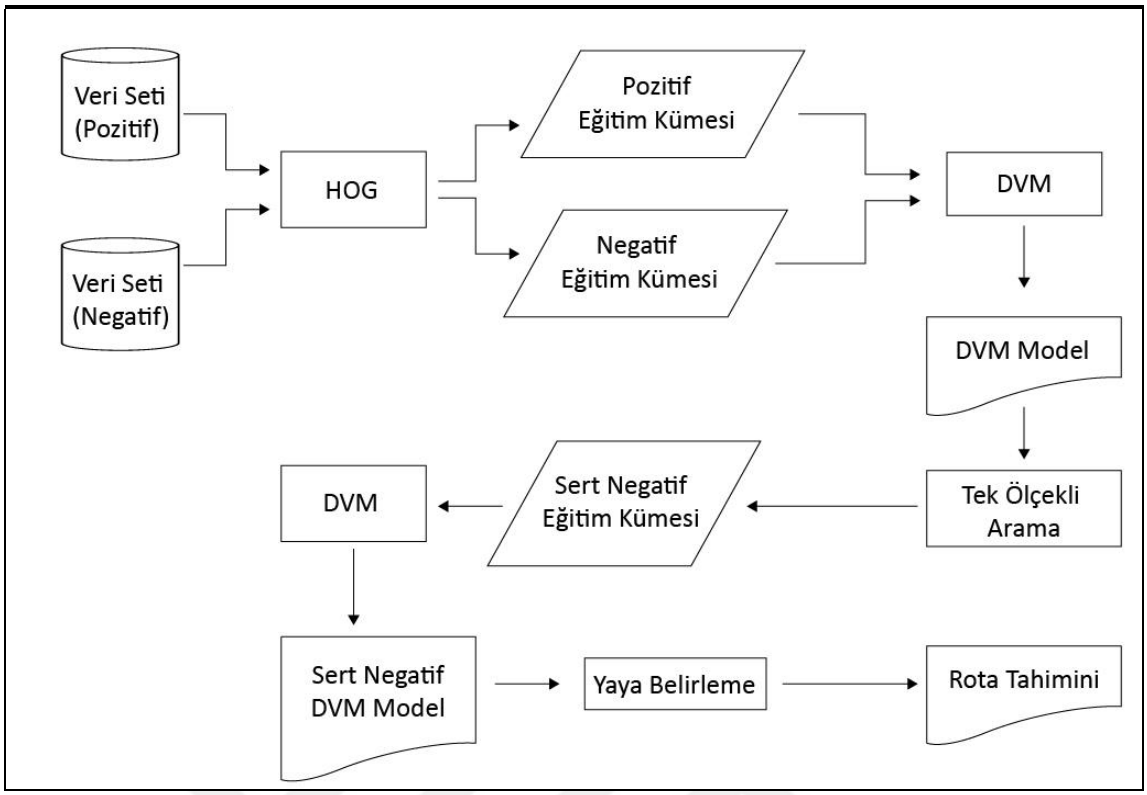
Bu bölümde Yönlü Türevlerin Histogramı algoritması ve Destek Vektör Makineleri sınıflandırıcı kullanarak; araç içerisine yerleştirilen kamera üzerinden elde edilen görüntüler yardımıyla yayaların belirlenmesi ve davranışlarının kestirilmesi konusunda çalışılmıştır. Algoritma ve önerilen sistemin yapısı bu bölümde detaylı olarak açıklanacaktır.

7.1 Çalışmada Kullanılan Veri Seti

Bu çalışmada Daimler Mono Pedestrian Benchmark veri seti [66] kullanılmıştır. Veri setinde eğitme amacıyla etiketlenmiş 15560 pozitif ve 6774 negatif görüntü mevcuttur. Eğitme setindeki görüntülerin boyutları 48x96 pikseldir. Test seti ise bir aracın içerisinden mono bir kamera ile kaydedilmiş VGA çözünürlüğünde 27 dakika uzunluğunda bir videodan oluşmaktadır.

7.2 Sistemin Çalışma Prensibi

Bu çalışmada yayaların yürüyüş rotalarının belirlenmesi için tasarlanan sistem Şekil 7.1'de ki akış şemasında gösterilen işlem basamaklarını takip ekmektedir. Bu işlem basamakları ayrıntılı olarak sırasıyla açıklanacaktır. Çalışma MATLAB ortamında geliştirilmiştir.

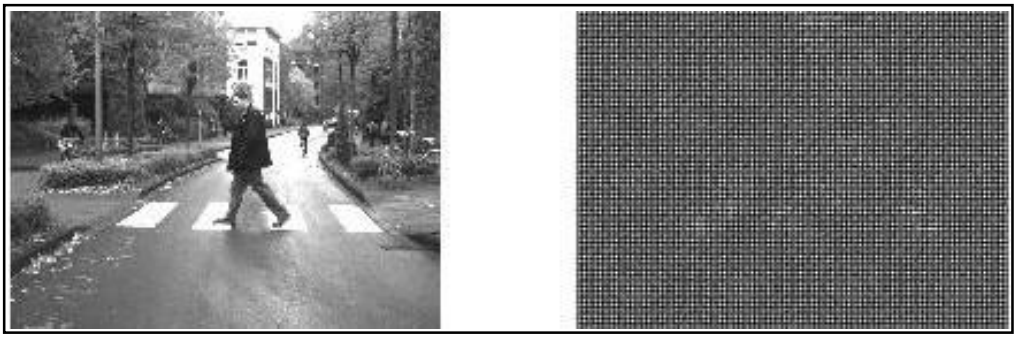


Şekil 7. 1 Yürüyüş rota belirleme sistemi işlem basamakları

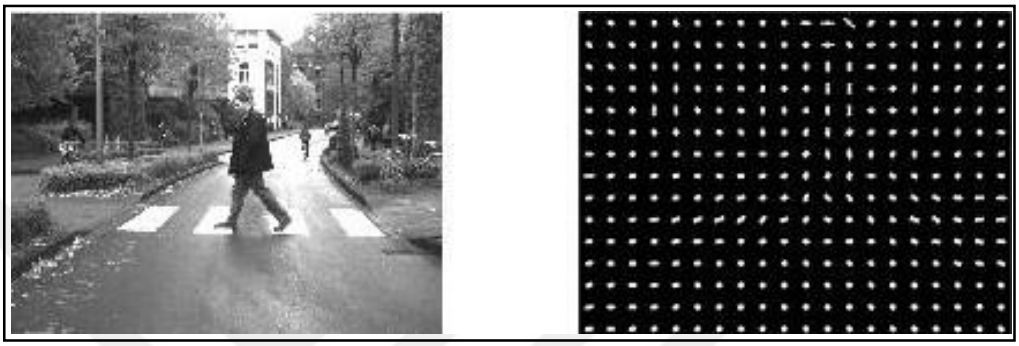
Çalışmada öncelikle Daimler Mono Pedestrian Benchmark veri setine ait eğitim kümesi içerisinde bulunan boyutları 48x96 piksel görüntüler sisteme girdi olarak verilir. Bu görüntüler pozitif ve negatif olmak üzere veri setinde eğitme amacıyla etiketlenmiş iki ayrı grup olarak bulunmaktadır. Verisetinde bulunan görüntü boyutlarına göre HOG boyutu belirlenir. Sonrasında iki ayrı veri seti için HOG algoritması kullanılarak öznitelikler çıkartılır. Veri serinden bulunan bir görüntüye ait HOG çıktıları aşağıdaki gibidir.



Şekil 7. 2 HOG test giriş görüntüsü ve çıkış öznitelikleri yön haritası

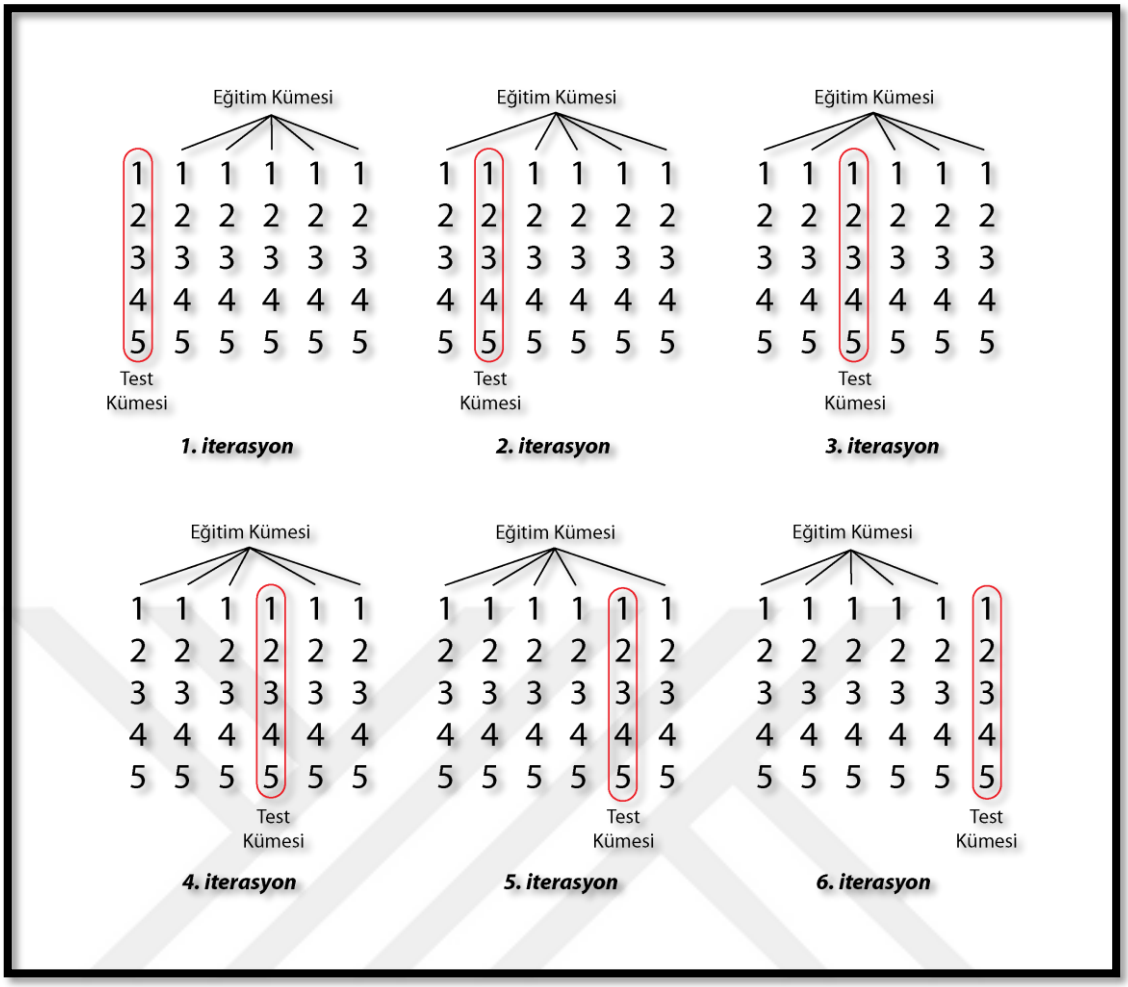


Şekil 7. 3 HOG test görüntüsü ve özniteliklerin yön haritası (8x8 hücre)



Şekil 7. 4 HOG test görüntüsü ve özniteliklerin yön haritası (32x32 hücre)

Elde edilen öznitelikler DVM kullanılarak eğitme işleminden geçirilir. Bu eğitme işleminin sonucunda ilk DVM modeli üretilmiş olur. Örüntü tanıma algoritmalarında veri setinin eğitilmesi işlemi için çeşitli algoritmalar kullanılmaktadır. Bu algoritmalardan bazıları literatürde çapraz doğrulama algoritmaları olarak bilinmektedir. Çapraz doğrulama algoritmada kendi içerisinde K-Fold Cross Validation [59], [67] ve Leave One Out Cross Validation [68], [69] olarak ikiye ayrılmaktadır. K-Fold Validation algoritmasında atanan k değerlerine göre sınıflar eğitim ve test kümelerine ayrılır. Bu algoritma ile bir nesneye ait test görüntüsü 10 ve üzeri olduğu durumlarda verimli sonuçlar elde edilebilir. Eğer test görüntüsü bu değerin altında ise Leave One Out Validation algoritmasına başvurulur. Bu algoritmada eğitim ve test kümeleri iterasyon mantığı ile belirlenmektedir. Örnek olarak Şekil 7.5'de beş farklı nesneye ait altı farklı video örneği içeren veri seti üzerinde iterasyonlar gösterilmiştir. Bu işlemde öncelikle her bir sütunda bir nesneye ait birer video olmak üzere, nesneler alt alta oluşturulan sıfır vektörlere yerleştirilir. İlk olarak 1. sütun test kümesi olarak diğer 5 sütun eğitim kümesi olarak sınıflandırıcıya verilir. Böylece 1. iterasyon işlemi tamamlanır ve bu işlem 6 sütunda bulunan tüm videolar için sırasıyla tekrarlanır. Her seferinde sınıflandırıcıdaki geçmiş değerler temizlenir.



Şekil 7. 5 Leave One Out Cross Validation iterasyon örneği

Daimler Mono Pedestrian Benchmark veri seti çapraz doğrulama algoritmaları için uygun olmadığından bu çalışmada eğitim aşamasında Sert Negatif Madencilik (Hard Negative Mining) tekniği kullanılmıştır [70]. Ayrıca bu yöntem ile modelin hatalı pozitif oranını düşürülmesi amaçlanmıştır.

Sert Negatif Madencilik yöntemine göre pozitif ve negatif veri setleri kullanılarak eğitilen ilk DVM modeli üzerine; test verileri uygulanır ve modelin hatalı olarak pozitif sonu verdiği örnekler tespit edilerek bu örnekler negatif eğitme setine eklenir. Elde edilen yeni veri seti ikinci bir DVM kullanılarak yeni bir eğitim işleminden geçirilir. Böylelikle sistemin hatalı pozitif oranının düşürülerek ikinci bir DVM modeli elde edilmiş olur. Algoritmada bulunacak sert negatif eşik değeri 15000 adet olarak belirlenmiştir. Bu değer ne kadar fazla ise; elde edilecek DVM modeli de bir o kadar iyi olacaktır.

Eğitim işlem basamağında elde edilen DVM modeli, yayaların video üzerinde belirlenmesi ve takip edilmesi işlemi için algoritmanın bir diğer adımına girdi olarak verilir. Burada genel olarak karşılaşılan en büyük problem; performans analizi için algoritmanın ürettiği sonuçlar ile gerçek değerlerin karşılaştırılmasıdır. Bu doğrulama işlemi için veri seti içerisinde bulunan Gerçek Referans Değer (GRD - Ground Truth) bilgisi kullanılacaktır. Bu işlem için algoritmanın o an üzerinde çalıştığı videonun ilk çerçevesini, GRD verisi içerisinde bulup yayaya ait bilgileri oluşturulan ilgili yapı dizisine kaydedilir.

GRD bilgisi dışında algoritmaya veri seti ile ilgili bazı parametrelerin girdi olarak verilmesi gerekir. Bu parametreler algoritmanın görüntüyü 2 boyutlu olarak algılamayıp, 3 boyutlu olarak algılamasını sağlayacaktır. Sistem üzerinde kullanılan parametreler aşağıdaki gibidir.

- Kameranın odak uzklığı = 0.012
- Piksel boyutu (metre) = 10^{-5}
- Kameranın eğim açısı (radian) = 0.043
- Görüntünün Merkezi = [320 240]
- Görüntünün merkezinin kameraya uzaklığı (metre) = 27
- Kameranın yerden yüksekliği = 1.17

Bu işlemlerden sonra özdevinim kestirimi için çerçeveler arasındaki belirlenen noktaların nekadar hareket ettiği bilgisi MATLAB üzerinde hazır olarak bulunan “vision.PointTracker” fonksiyonu ile hesaplanır. Bu fonksiyon Kanade-Lucas-Tomasi (KLT) algoritması kullanılarak oluşturulmuştur [71]. Yayaların tespit edilebilmesi için ise HOG öznitelikleri “vision.PeopleDetector” fonksiyonu algoritmada geliştirilerek kullanılmıştır.

Burada algoritma görüntüyü ölçekleyerek tarama yaptığı için aynı yaya birden fazla bulunabilir. Bunun için birbiriyle örtüşen yaya tespitleri SVM'den çıkan skora göre elenip tek bir kutucuğa dönüştürülmesi gerekmektedir. MATLAB üzerinde hazır fonksiyon olarak bulunan “selectStrongestBbox” komutu bu işlemi gerçekleştirir. Bu işlemde sisteme bir eşik değeri girilmesi gerekmektedir. SVM skoru incelenip bu değer algoritma içerisnde 0.5 olarak belirlenmiştir.

Algoritma tespit ettiği yayaları, GRD ile karşılaştırarak 3 farklı sonuç üretir. Eğer GRD'de yaya var ve algoritma bunu bulamamış ise sonucu "noFound" adlı değişken dizisi üzerinde; tespit etme işlemi doğru ise sonucu "trueDetect" adlı değişken dizisi üzerinde; istenen örtüşme yoksa bunu "falseDetect" adlı değişken dizisi üzerinde sayılır.

Elde edilen ikinci DVM modeli üzerinden yayaların konumları belirlendikten sonra aracın özdevinimi her çerçeveden çıkarılarak aracın hareketinden kaynaklanan hatalar giderilmeye çalışılmıştır. Özdevinim hesaplanırken sadece yatay eksene ait koordinatlar üzerinden işlemler gerçekleştirilir çünkü yayaların yön kestirimi için sadece yatay eksen üzerinden hızlarının hesaplanması yeterli olacaktır. Hesaplanan özdevinimler herbir tespit edilen nesne için farklı olacağından bu işlemin her bir tespit için gerçekleştirilmesi gerekmektedir. Bunun sebebi çerçevenin sağ tarafındaki özdevinim pozitif, sol tarafındaki özdevinim ise negatif olmasından kaynaklıdır.

Tespit edilen yayaların konumu piksel cinsinden belirlendikten sonra peş peşe gelen çerçevelerdeki konumları arasındaki fark elde edilerek yayanın video boyunca kat ettiği rota tespit edilir. Yayanın gideceği yön ise şu andaki çerçevede bulunduğu konum ile daha önceki çerçevelerdeki konumlarının ortasındaki fark yardımı ile kestirilir.

Eğer tespit edilen yayayı çevreleyen kutunun en alt noktası çerçevenin dikey eksenin orta noktası üzerinde ise; bu durum algoritma tarafından "algılanan yayanın arabadan uzakta olduğu" sonucunu üretmesine ve herhangi bir işlem yapmaması gerektiği kararına varılır. Tam tersi senaryo ise yayaların arabaya yakın olması durumu anlamına gelir ve bulunan yayaların hangi yöne gittiği belirlenir. Bu yapının dezavantajı ise sabit olan yayaların sistem tarafından arabaya uzak olarak olarak algılanabilmesi durumudur.

DVM'in yanlış tespitlerini dizi üzerinden silmek için ise tespit edilen yayanın en son hangi çerçeve üzerinde görüldüğüne bakılır. Eğer son iki çerçevede görülmemiş ise algoritma tarafından silinir. Böylece yanlış pozitifler elimine edilmiş olur.

Algoritma üç farklı DVM çekirdek fonksiyonu için çalıştırılarak başarısı test edilmiştir. Testlerde kullanılan DVM fonksiyonları ve parametreleri aşağıdaki gibidir. Tüm çekirdek fonksiyonları için $C = \{10^0, \dots, 10^3\}$ olarak belirlenmiştir.

- Doğrusal DVM fonksiyonu

• Gauss radyal temelli fonksiyon $\gamma = \{10^{-3}, 10^{-4}\}$

• Polinom temelli fonksiyon $d = \{2,3,4\}$

Algoritmanın son adımında tespit edilen yayanın yön tahmini yapıldıktan sonra çıkan 4 boyutlu matris video olarak kaydedilir.

Uygulama sonucunda elde edilen örnek sonuçlar Şekil 7. 6 üzerinden açıklanacak olursa; beyaz dikdörtgenler doğru bulunan yayaları, kırmızı çemberler ise tespit edilemeyen yayaları göstermektedir. Görüntüler detaylı olarak Ek-A bölümünde verilmektedir.



Şekil 7. 6 Uygulama sonucunda elde edilen örnekler

Yapılan çalışma sonucunda önerilen yöntem eğitme verisinde 127 adet yaya olmasına rağmen algoritma tarafından 129 adet hareketli nesne tespit etmiştir. Bulunan nesnelerin 121 adedi yaya iken, kalan 8 adedi farklı nesneler olmasına rağmen yaya olarak işaretlenmiştir. 6 adet yaya ise tespit edilememiştir. Tespit edilen 121 yayanın 117 adedinin rotası doğru olarak öngörülebilmiştir. Çizelge 8. 1 elde edilen sonuçları özetlemektedir.

Çizelge 7. 2 Performans sonuçları

Toplam Yaya	Tespit Edilen Yaya (DP)	Tespit Edilemeyen Yaya (YN)	Doğru Rota Tahmini	Yanlış Rota Tahmini
127	121	6	117	4

Bu sonuçlar Hipotez testi kullanılarak Çizelge 8.2 üzerinden değerlendirilerek algoritmanın başarım oranı hesaplanmıştır. Görüntüde yaya olduğunda algoritmanın bunu yaya olarak algılaması Doğru Pozitif (DP - True Positive); görüntüde yaya olmadığı halde algoritmanın bunu yaya olarak algılaması Yanlış Pozitif (YP - False Positive) olarak temsil edilmiştir.

Çizelge 7. 3 Hipotez test ölçütü

	Yaya Var (Test Sonucu)	**Yaya Var (Test Sonucu)**
Yaya Var (Hipotez)	YN	DP
Yaya Yok (Hipotez)	DN	YP

Bu sonuçlar doğrultusunda algoritmanın başarısı aşağıdaki denklemler üzerinden hesaplanmaktadır [12]. Bu denklemler açıklanacak olursa; Sistem Kesinliği (SK - Precision) hipotezi doğru kabul edilenlerin tüm getirilen örneklere oranı, Sistem Duyarlılığı (SD - Recall) doğru tespit edilen yaya sayısı, F_m ise bunların harmonik ortalaması olarak hesaplanacaktır.

$$SK = \frac{DP}{DP+YP} \tag{7.1}$$

$$SD = \frac{DP}{DP+YN} \tag{7.2}$$

$$F_m = \frac{2\ x\ SD\ x\ SK}{SD + SK} \tag{7.2}$$

$$Doğru\ Rota\ Kestrimi = \frac{\text{Doğru Rota Tahmini}}{\text{Tespit Edilen Yaya}} \tag{7.4}$$

Çizelge 7. 4 Hipotez tet sonuçları

SK	**SD**	F_m
0,937	0,952	0,944

BÖLÜM 8

SONUÇ VE ÖNERİLER

Bu tez çalışmasında Yönlü Türevlerin Histogramı ve Destek Vektör Makinaları yöntemleri kullanılarak yayaların rotalarının belirlenmesi uygulaması gerçekleştirilmiştir. Önerilen yöntemin yaya tespit başarımı DVM'nin Gauss radyal temelli çekirdek fonksiyonu performans sonuçları üzerinden değerlendirildiğinde mevcut yayaların %95.27 oranında tespit edilebildiği görülmüştür. Tespit edilen yayaların rotaları ise %96.69 oranında doğru öngörülmüştür. Bu değerlerden önerilen yöntemin başarılı olduğu sonucuna varılmıştır.

Kaynak olarak kullanılan bilimsel çalışmalarla kıyaslandığında; bu çalışmada önerilen yöntemin otonom sürüşlü araçlara uygulanabilirliği ve geliştirilebilirliği konusunda gelecek araştırmalara referans olma potansiyeline sahip olduğu görülmüştür. Bu çalışmanın devamında, daha karmaşık sahnelerde yayaları tespit edebilen fonksiyonlarının geliştirilmesi ve bunların donanımsal olarak uygulanması amaçlanmaktadır.

KAYNAKLAR

[1] Schneemann, F., ve Heinemann, P., (2016). "Context-based Detection of Pedestrian Crossing Intention for Autonomous Driving in Urban Environments," IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 9-14 Ekim 2016, Korea, 2243-2248.

[2] Li, Z. ve Huang, F., (2006). "Human Gait Tracking Based on Linear Model Fitting", Proceedings of the First International Multi-Symposiums on Computer and Computational Sciences, 283-287.

[3] Kooij, J. F. P., Schneider, N., Flohr, F. ve Gavrila, D. M., (2014). "Context-Based Pedestrian Path Prediction", in Computer Vision ECCV, 618–633.

[4] Hariyono, J., Shahbaz, A. ve Jo, K., (2015). "Estimation of Walking Direction for Pedestrian Path Prediction from Moving Vehicle", IEEE/SICE International Symposium on System Integration (SII), 11-13 Aralık 2015, Meijo, 750-753.

[5] Jeong, M., Ko, B., C. ve Nam, J., (2017). "Early Detection of Sudden Pedestrian Crossing for Safe Driving during Summer Nights", IEEE Journals & Magazines, Transactions on Circuits and Systems for Video Technology, 27: 1368 – 1380.

[6] Schauland, S., Velten, J. ve Kummert, A., (2007). "3D Velocity Filters for The Detection of Moving Objects in Image Sequences", 15th European Signal Processing Conference (EUSIPCO 2007), 3-7 Eylül 2007, Poznan, 326-329.

[7] Köhler, S., Goldhammer, M., Bauer, S., Zecha, S., Doll, K., Brunsmann, U. ve Dietmayer, K., (2013). "Stationary Detection of the Pedestrian's Intention at Intersections", IEEE Intelligent Transportation Systems Magazine, 5(4):87–99.

[8] Keller C. G. ve Gavrila, D. M., (2014). "Will the Pedestrian Cross? A Study on Pedestrian Path Prediction", IEEE Transactions on Intelligent Transportation Systems, 15(2):494–506.

[9] Quintero, R., Almeida, J., Llorca, D. F. ve Sotelo, M. A., (2014). "Pedestrian Path Prediction using Body Language Traits", IEEE Intelligent Vehicle Symposium Dearborn, Michigan, (Iv):317– 323.

[10] Furuhashi, R. ve Yamada, K., (2011). “Estimation of Street Crossing Intention from a Pedestrian s Posture on a Sidewalk Using Multiple Image Frames”, IEEE Pattern Recognition First Asian Conference, Beijing, 17–21.

[11] Negri, P. ve Garayalde, D., (2014). “Concatenating Multiple Trajectories Using Kalman Filter for Pedestrian Tracking”, IEEE Biennial Congress of Argentina (ARGENCON), Buenos Aires, 364-369.

[12] Kataoka, H., Tamura, K., Aoki, Y. ve Matsui, Y., (2012). “Extended CoHOG and Particle Filter by Improved Motion Model for Pedestrian Active Safety,” 38th Annual Conference on IEEE Industrial Electronics Society IECON, 1511-1516.

[13] Maki, A., Seki, A., Watanabe, T. ve Cipolla, R., (2011). “Co-Occurrence Flow for Pedestrian Detection,” 18th IEEE International Conference on Image Processing, 1889-1892.

[14] Yetim S., (2016). “Sürücüsüz Araçlar ve Getirdiği/Getireceği Hukuki Sorunlar”, Ankara Barosu Dergisi, (2016/1), 127-184.

[15] Parent, M., (2015). CityMobil2, Cities Demontsrating Automated Road Passenger Transport, http://www.citymobil2.eu/en/ , 28 Nisan 2018.

[16] Wikipedia, Autonomous Car, https://en.wikipedia.org/wiki/Autonomous_car, 28 Nisan 2018.

[17] Lari, A., Douma, F. ve Onyiah, I., (2015). “Self-Driving Vehicles and Policy Implications: Current Status of Autonomous Vehicle Development and Minnesota Policy Implications”, Minnesota Journal of Law, Science & Technology, 16(2): 735-769.

[18] Kaufman ve Alexander C., (2015). The Tesla Of The Future Could Look A Lot Like Uber Driving Would Never Be The Same, http://www.huffingtonpost.com/entry/teslauber-rival_55d1e72fe4b0ab468d9db3fd?utm_hp_ref=business&kvcommref=mostpopular, 29 Nisan 2018.

[19] Mphippen., (2013). Difference Between Radar and Lidar Explained, http://www.lasertech.com/blogs/Traffic-Safety/post/2013/05/15/Difference-Between-Radar-and-Lidar-Explained.aspx, 29 Nisan 2018.

[20] Fub, P., (2015). Who’s In The Driving Seat?, How The Rise Of Autonomous Vehicles Will Transform The Relationship Between Man And Car, http://www.ey.com/GL/en/Industries/Automotive/EY-whos-in-the-driving-seat, 28 Nisan 2018.

[21] Fub, P., (2015). Gateway Cities Technology Plan for Goods Movement, https://www.yumpu.com/en/document/view/20971775/gateway-cities-its-plan-autonomous-vehicle-reseachsummary/21, 28 Nisan 2018.

[22] Forrest, A. ve Konca, M., (2007). Autonomous Cars and Society, Worcester Polytechnic Institute, Massachusetts.

[23] Nichols, C., (2013). Liability Could Be Roadblock For Driverless Cars, http://www.sandiegouniontribune.com/news/sdut-liability-driverless-car-transovation-google-2013oct30-story.html, 28 Nisan 2018.

[24] Harris, M., (2014). FBI Warns Driverless Cars Could Be Used As 'Lethal Weapons', http://www.theguardian.com/technology/2014/jul/16/google-fbi-driverless-cars-leathal-weapons-autonomous, 28 Nisan 2018.

[25] Alpaslan, N., (2013). Gradyan Tabanlı Heterojen Öznitelik Çıkarma Yöntemlerine Yeni Yaklaşımlar, Yüksek Lisans Tezi, İnönü Üniversitesi Fen Bilimleri Enstitüsü, Malatya.

[26] Paulus, D.W.R ve Hornegger, J., (2003). Applied Pattern Recognition, Wieweg, Germany, 372.

[27] Gunal S., (2001). Örüntü tanıma uygulamalarında alt uzay analiziyle öznitelik seçimi ve sınıflandırma, Doktora tezi, Osmangazi Üniversitesi Fen Bilimleri Enstitüsü, Eskişehir.

[28] Duda, R. O., Hart, P. E. ve Stork, D. G., (2001). "Pattern classification", John Wiley & Sons Inc., USA, 654.

[29] Sun, B., Yan, J. ve Liu, Y., (2010). "Human Gait Recognition by Integrating Motion Feature and Shape Feature", International Conference on Multimedia Technology (ICMT), Ningbo, 1-4.

[30] Liu, Y. ve Wang, X., (2010). "Gait Recognition Based on the Body's Natural Features", 2010 International Conference on Computer Application and System Modeling, Taiyuan, 15:91-93.

[31] Lee, B., Hong, S., Lee, H. ve Kim, E., (2010), "Gait Recognition System Using Decision-Level Fusion", Proceedings of the 5th IEEE Conference on Industrial Electronics and Applications, Taichung, 313-316.

[32] Ceylan R., (2009). Özellik çıkarma teknikleri ve yapay sinir ağları kullanarak bir tele-kardiyoloji sistem tasarımı, Doktora Tezi, Selçuk Üniversitesi Fen Bilimleri Enstitüsü, Konya.

[33] Canny, J., (1986). "A Computational Approach to Edge Detection", IEEE Trans. Patt.Analy. Mach. Intel. 8, 6, 679-698.

[34] J.M.S. Prewitt, (1970). "Object Enhancement and Extraction" in "Picture processing and Psychopictorics", Academic Press.

[35] H.P. Moravec, (1977). "Towards Automatic Visual Obstacle Avoidance (short version)", Proceedings of the 5th International Joint Conference on Artificial Intelligence, MIT, Cambridge, 584.

[36] W. Förstner, (1986). "A Feature Based Correspondence Algorithm for Image Matching", International Archives of Photogrammetry and Remote Sensing, 26:150-166.

[37] Harris, C. G. ve Stephens, M., (1988). "A Combined Corner and Edge Detector", Proc. of Fourth Alley Vision Conference, Manchester, 182-192.

[38] Bastanlar, Y. ve Yardimci, Y., (2007). "Selecting Image Corner Points Using Their Corner Properties", Signal Processing and Communications Applications.

[39] Lowe, D.G., (2004). "Distinctive Image Features from Scale-Invariant Keypoints", International Journal of Computer Vision, 60: 91-110.

[40] Bay, H., Tuytelaars, T. ve Van Gool, L. (2006). "SURF: Speeded Up Robust Features", The Proceedings of the 9th European Conference on Computer Vision, 7-13 May 2006, Graz.

[41] Matas, J., Chum, O., Urban, M. ve Pajdla, T., (2004). "Robust wide-baseline stereo from maximally stable extremal regions", Image and Vision Computing, 22: 761–767.

[42] Güney, M. ve Arıca, N., (2009). "Desen Tabanlı İlgi Bölgesi Tespiti", Naval Science and Engineering, 5: 94-106.

[43] Ekinci, M., ve Aykut, M., (2006). “Yürüyüş Stiline Dayalı Kişilerin Tanınması”, Proceedings of the IEEE 15th Signal Processing and Communications Applications.

[44] Chen, X., Fan, Z., Wang H. ve Li, Z., (2010). “Automatic Gait Recognition Using Kernel Principal Component Analysis”, Proceedings of the International Conference on Biomedical Engineering and Computer Science (ICBECS), Wuhan, 1-4.

[45] Huang, X. ve Boulgouris, N. V., (2010). “Gait Recognition Using Linear Discriminant Analysis with Artifical Walking Conditions”, Proceedings of the 17th IEEE International Conference on Image Processing, Hong Kong, 2461-2464.

[46] Haşiloğlu, A., (1999). Dalgacık Dönüşümü ve Yapay Sinir Ağları ile Döndürmeye Duyarsız Doku Analizi ve Sınıflandırma. Teknik Rapor, Atatürk Üniversitesi, Mühendislik Fakültesi Elektronik ve Haberleşme Mühendisliği Bölümü, Erzurum.

[47] Özden, M., (2005). Ortalama Kayma Algoritmasının Geliştirilerek Görüntü Dizilerinde Hareketli Nesne Takibi ve Görüntü Kesimleme Amaçlı Kullanılması, Yüksek Lisans Tezi, Kırıkkale Üniversitesi Fen Bilimleri Enstitüsü, Kahramanmaraş.

[48] Wei, S., Ning, C. ve Gao, Y., (2010). “Biomimetic Gait Recognition Based on Motion Contours Wavelets Analysis and Mutual Information”, Proceedings of the 3rd International Congress on Image and Signal Processing, Yantai, 404-408.

[49] Safran, M.I. ve Oktem, R., (2007). "A Fast Hough Transform Approximation and Its Application for Barcode Localization", IEEE 15th Signal Processing and Communications Applications, 11-13 Haziran 2007 Eskişehir.

[50] Viola, P. ve Jones, M., (2001). "Rapid object detection using a boosted cascade of simple features", Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 8-14 Kasım 2001, Kauai, Hawaii.

[51] Mikolajczyk, K. ve Schmid, C., (2005). “A performance evaluation of local descriptors”, IEEE Trans. Pattern Anal. Mach. Intell., 1615–1630.

[52] Watanabe, T., Ito, S. ve Yokoi, K., (2009). "Co-occurrence histograms of oriented gradients for pedestrian detection", Third Pacific Rim Symposium on Image and Video Technology, 13-16 Ocak 2009, Tokyo.

[53] Shashua, A., Gdalyahu, Y. ve Hayon, G., (2004), “Pedestrian Detection for Driving Assistance Systems: Single-frame classification and system level performance,” In Proceedings of IEEE Intelligent Vehicles Symposium, Parma, Italy 14-17 Haziran 2004, 1-6.

[54] Dalal, N. ve Triggs, B., (2005). “Histograms of Oriented Gradients For Human Detection”, In C. Schmid, S. Soatto, and C. Tomasi, editors, International Conference on Computer Vision and Pattern Recognition, 2:886–893.

[55] ITU-R BT.601, (1995). Studio encoding parameters of digital television for standard 4:3 and wide-screen 16:9 aspect ratios, International Telecommunication Union.

[56] Dalal, N., (2006). "Finding People in Images and Videos", Master Thesis, Grenoble Institute of Technology Computer Sciences, Applied and Pure Mathematics, Grenoble, France.

[57] 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 Ekim 2009, Kyoto.

[58] Burges, Christopher J.C., (1998). A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery, 2(2):121-167.

[59] Begg, Rezaul K., (2005). “Support Vector Machines for Automated Gait Classification,” IEEE Transactions on Biomedical Engineering, 52(5):828-838.

[60] Yang, Q. ve Tian, Y., (2011). “Gait recognition based on KDA and SVM,” Information Technology and Artificial Intelligence Conference (ITAIC), 2011 6th IEEE Joint International, 1:160-163.

[61] Yan Ma, (2010). “Gait Recognition Using Sparse Representation”, Proceedings of the 2010 International Conference on Wavelet Analysis and Pattern Recognition, Qingdao, 136-139.

[62] Oliver, N.M., Rosario, B. ve Pentland, A.P., (2000). "A Bayesian Computer Vision System for Modeling Human Interactions", IEEE PAMI, 22(8):831-843.

[63] Lo, B.P.L. ve Velastin, S.A., (2001). "Automatic Congestion Detection System for Underground Platforms", Proc. ISIMP 2001, 158-161.

[64] Stauffer, C. ve Grimson, W.E.L., (1999). "Adaptive Background Mixture Models for Real-Time Tracking", CVPR, Fort Collins, CO, USA.

[65] Gonzalez, R. C. ve Woods, R. E., (2007). “Digital Image Processing”, 2nd ed., Beijing: Publishing House of Electronics Industry.

[66] Enzweiler, M. ve Gavrila, D. M. (2009). “Monocular pedestrian detection: survey and experiments”, IEEE Transactions On Pattern Analysis And Machine Intelligence, 31(12):2179-2195.

[67] Wang, F., Wen, S., Wu, C., Zhang, Y. ve Wang, H., (2011). “Gait Recognition Based on the Fast Fourier Transform and SVM”, Control and Decision Conference CCDC, Chinese, 1091-1094.

[68] Su, H. ve Huang, F., (2005). “Human Gait Recognition Based on Motion Analysis”, Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, Guangzhou, 7:4464-4468.

[69] Ye, B. ve Wen, Y., (2007). “Gait Recognition Based on DWT and SVM”, Proceedings of the 2007 International Conference on Wavelet Analysis and Pattern Recognition, Beijing, China, 1382-1387.

[70] Felzenszwalb, Pedro F., Girshick, Ross B., McAllester, D. ve Ramanan, D., (2010). “Object Detection with Discriminatively Trained Part Based Models”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 32(9): 1627-1645.

[71] MathWorks MATLAB R2016a User’s Guide, Image Processing Toolbox, https://www.mathworks.com/products/image.html, 12 Mart 2018.

EK-A

SİMÜLASYON SONUÇLARI



Şekil A. 1 Örnek program çıktısı



Şekil A. 2 Örnek program çıktısı



Şekil A. 3 Örnek program çıktısı



Şekil A. 4 Örnek program çıktısı



Şekil A. 5 Örnek program çıktısı



Şekil A. 6 Örnek program çıktısı



Şekil A. 7 Örnek program çıktısı



Şekil A. 8 Örnek program çıktısı

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Cihangir KOLCU

Doğum Tarihi ve Yeri : 27.11.1987, İstanbul

Yabancı Dili : İngilizce

E-posta : cihangir_kolcu@hotmail.com

ÖĞRENİM DURUMU

Derece	**Alan**	**Okul/Üniversite**	**Mezuniyet Yılı**
Y. Lisans	Elekt. ve Hab. Müh.	Yıldız Teknik Üniversitesi	2018
Lisans	Elekt. ve Hab. Müh.	Haliç Üniversitesi	2012
Lise	Fen Bilimleri	Bahçelievler Lisesi	2005

İŞ TECRÜBESİ

Yıl	**Firma/Kurum**	**Görevi**
2015	Armada Bilgisayar A.Ş.	HPE/Aruba Wireless Engineer

YAYINLARI

Bildiri

1. Kolcu, C. ve Bolat, B., (2018). “Yayaların Yürüyüş Rotalarının Belirlenmesi”, EBBT Elektrik-Elektronik & Bilgisayar & Biyomedikal Mühendislikleri Bilimsel Toplantısı, 18-19 Nisan 2018, İstanbul.

2. Kolcu, C. ve Özen, F., (2011). “Yürüyüş Tanımada Yeni Teknikler” Geleceğin Teknolojileri Sempozyumu, 20-21 Ekim 2011, İstanbul, 121-126.