

GERÇEK ZAMANLI UYGULAMALAR İÇİN YAYA TESPİT SİSTEMİ

PEDESTRIAN DETECTION SYSTEM FOR REAL-TIME APPLICATIONS

MUHAMMED ENİS MIHÇIOĞLU

DOÇ.DR. ALİ ZİYA ALKAR

Tez danışmanı

Hacettepe Üniversitesi

Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliğinin

Elektrik Ve Elektronik Mühendisliği Anabilim Dalı için Öngördüğü

YÜKSEK LİSANS TEZİ olarak hazırlanmıştır.

2016

Muhammed Enis MIHÇIOĞLU' nun hazırladığı '**Gerçek Zamanlı Uygulamalar için Yaya Tespit Sistemi** adlı bu çalışma aşağıdaki jüri tarafından **ELEKTRİK ve ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI**' nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Yrd. Doç. Dr. Uğur TÖREYİN
Başkan

Doç. Dr. Ali Ziya ALKAR
Danışman

Yrd. Doç. Dr. Derya ALTUNAY
Üye

Doç. Dr. Umut SEZEN
Üye

Yrd. Doç. Dr. Mehmet DEMİRER
Üye

Bu Tez Hacettepe Üniversitesi Fen Bilimleri Enstitüsü tarafından **YÜKSEK LİSANS TEZİ** olarak onaylanmıştır.

Prof. Dr. Fatma SEVİN DÜZ

Fen Bilimleri Enstitü Müdürü

Her zaman yanımda olan canım aileme...

ETİK

Hacettepe Üniversitesi Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım tez çalışmada,

- tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,
- başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
- atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,
- kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- ve bu tezin herhangi bir bölümünü bu üniversitede veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı

beyan ederim

MUHAMMED ENİS MIHÇIOĞLU

ÖZET

GERÇEK ZAMANLI UYGULAMALAR İÇİN YAYA TESPİT SİSTEMİ

Muhammed Enis MIHÇIOĞLU

Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü

Tez Danışmanı: Doç.Dr. Ali Ziya Alkar

Şubat 2016, 83 sayfa

Şehir içi trafiğinde farklı uyarı kaynaklarının sürücü dikkatini dağıtmasından ötürü ya da sürücünün anlık dikkatsizliğinden dolayı yaya ölümlerine ve yaralanmalarına neden olan trafik kazaları yaşanmaktadır. Bu sebeple, günümüzde araç kazalarının yol açtığı yaralamaların ve ölümlerin önüne geçmek için otomobiller yaya tespit sistemleri ile üretilmektedir. Ancak, hali hazırda sadece orta ve üst seviye otomobillerde bulunan bu özellikten diğer otomobil sahipleri yararlanamamaktadır. Yaya tespit teknolojisinin diğer araçlarda da kullanılabilir olması, yayaların karıştığı kazaların azaltılmasına önemli oranda katkı sağlayacaktır.

Tez çalışmasında, nesne tespitinde sıkça kullanılan HOG ve Haar yöntemleri birlikte kullanılarak özel bir yaya tespit algoritması geliştirilmiştir. Algoritma geliştirilirken mekânsal ve parçasal bilgiden azami oranda faydalanmak için ön

bilgi olarak alt, üst ve tüm vücut bölümleri için farklı ilgilenilen alan seçimleri kullanılmıştır. Parçasal vücut tespitlerinin birleştirilmesiyle tespit oranı ve tespit hızı gerçek zamanlı sistemler için uygun hale getirilmiştir. Parçasal bilgi birleştirilirken kullanılan skor kıstası ile tespitler arasında önceliklendirme sağlanmıştır. Nihai karar verici olan HOG yöntemi uygulanmadığında dahi skor kıstası karar vermede kullanılabilir. Tasarlanan sistem kısa tepki süresi ile yüksek doğrulukla kullanıcıya sesli ve görsel bir uyarı sağlayarak kaza oluşumunu engellemeyi amaçlamaktadır.

Anahtar Kelimeler: Yaya tespiti, Nesne tespiti, Gerçek Zamanlı Sistemler, HOG, Haar Özellikleri, İlgilenilen Alan, Parçasal Bilgi

ABSTRACT

PEDESTRIAN DETECTION SYSTEM FOR REAL-TIME APPLICATIONS

Muhammed Enis MIHÇIOĞLU

**Master of Science, Department of Electrical and Electronics
Engineering**

Supervisor: Assoc. Prof. Ali Ziya Alkar

February 2016, 83 pages

In modern daily life, traffic accidents happen that involve pedestrians in urban traffic due to different distractions and lack of driver attention. Because of this reason, automobile manufacturers install pedestrian detection and warning systems on newly produced automobiles. However, currently these systems only exist on some new middle and high segment cars. Millions of car owners do not have access to these types of systems. Usage of pedestrian detection and warning system for automobiles can reduce traffic accidents that result in pedestrian injuries and deaths.

In this thesis work, a new algorithm is developed based on HOG (Histogram of Gradients) and Haar like Features. Algorithm uses partial detections for full,

upper and lower body parts and combines them afterwards. Algorithm also uses different ROI's (Region of Interests) for different partial detections in order to utilize partial and spatial information.

Using a score criterion after combining the partial Haar detections provides a prioritization among different detections. Even without the final HOG detection; the algorithm can provide final detection results with high rate of confidence. Hence, algorithm suits to real-time applications in terms of detection rate and detection speed. The designed system aims to alert the user with short reaction time and high accuracy in order to prevent traffic accidents involving pedestrians.

Keywords: Pedestrian detection, Object Recognition, Real-time System, HOG, Haar Features, Region of Interest, Partial Information

TEŐEKKÜR

Deęerli eleőtiri, yardım ve önerileri ile alıőmamın ortaya ıkmasına katkıda bulunan tez danışmanım sayın Do. Dr. Ali Ziya ALKAR' a,
Sabır, hoőęörü ve desteęinin yanı sıra tezimi tamamlamam için beni motive eden deęerli eőim Sabır MIHIOęLU' na.
Desteklerini eksik etmeyen aileme teőekkür ve minnetlerimi sunarım.

İÇİNDEKİLER

	<u>Sayfa</u>
ÖZET	i
ABSTRACT	iii
TEŞEKKÜR	v
İÇİNDEKİLER.....	vi
ÇİZELGELER.....	ix
ŞEKİLLER	x
SİMGELER VE KISALTMALAR	xi
1. GİRİŞ.....	1
2. YAYA TESPİT YÖNTEMLERİ ve YAKLAŞIMLARI	3
2.1 Yaya Tespit Sistemlerinde Kullanılan Farklı Yöntemler	3
2.2 Video Kamera Tabanlı Yaya Tespit Sistemleri	7
2.2.1 Video Kamera Tabanlı Yaya Tespit Sistemlerindeki Yaklaşımlar	7
2.2.2 Video Kamera ile Çarpışma Tahmini ve Yaya Davranışı Analizi	10
2.3 Gerçek Zamanlı Yaya Tespit Yaklaşımı	11
2.3.1 Özellik Çıkarım Yöntemi	11
2.3.2 Özellik Sınıflandırıcı Yöntemi	11
2.3.3 Ön Bilgi Kullanma Yöntemi	12
2.3.4 Basamaklı Sınıflandırıcı Kullanmak	12
2.3.5 Önceliklendirme ve Sınır Koyma	12
2.4 Tez Çalışmasında Kullanılan Yöntemler	12
2.4.1 HOG Tanımlayıcısı	13
2.4.1.1 Teori	13
2.4.1.2 Algoritma	14
2.4.2 Haar Özellikleri Tabanlı Tanımlayıcı	20

2.4.2.1 Teori	20
2.4.2.2 Algoritma	21
3. GERÇEK ZAMANLI YAYA TESPİT ALGORİTMASI	25
3.1 Haar Yöntemi ile Parçasal ve Tüm Vücut Tespiti	26
3.1.1 Alt Vücut Tespiti	27
3.1.2 Üst Vücut Tespiti	28
3.1.3 Tüm Vücut Tespiti	28
3.2 HOG Yöntemi ile Tüm Vücut Tespiti.....	29
3.3 Haar Yöntemi için Sınıflandırıcı Seçimi	30
3.4 İlgilenilen Alan seçimi	34
3.4.1 Ön Bilgi Kullanılarak Yapılan Sabit İlgilenilen Alan Seçimi	35
3.4.2 Ön Bilgi Kullanılmadan Yapılan Dinamik İlgilenilen Alan Seçimi	35
3.5 Farklı Tespitlerin Birleştirilmesi.....	37
3.6 Haar Ön İşleme ile HOG Yönteminde Elde Edilen İyileşme	39
3.7 Skor Kıtası ile Birleştirilen Tespitleri Önceliklendirme	39
4. ÇALIŞMANIN FARKLI PLATFORMLAR ÜZERİNDE GERÇEKLENMESİ	43
4.1 Testlerde Kullanılan Parametreler	43
4.2 Testlerde Kullanılan Veri Seti ve Yaklaşım.....	44
4.3 CPU Tabanlı Sistemdeki Test Sonuçları	45
4.3.1.1 Donanım ve Yazılım Yapısı	45
4.3.1.2 Test Sonuçları	46
4.4 GPU ve CPU Tabanlı Sistemdeki Test Sonuçları.....	50
4.4.1.1 Donanım ve Yazılım Yapısı	50
4.4.1.2 Test Sonuçları	51
4.5 Mobil İşlemci Tabanlı Sistemdeki Test Sonuçları	54
4.5.1.1 Donanım ve Yazılım Yapısı	54

4.5.1.2 Test Sonuları	55
4.6 Test Sonularının Deęerlendirilmesi.....	58
5. SONULAR VE NERİLER	60
KAYNAKLAR.....	62
EKLER	66
EK-1	66
ZGEMİŐ	67

ÇİZELGELER

Tablo 2-1 Yaya Tespit Sistemlerindeki Sensor Sistemlerinin Karşılaştırması [5]....	4
Tablo 2-2 LAZER, RADAR ve Kızılötesi Kamera ile Yaya Tespit Yöntemleri [5]....	5
Tablo 2-3 Video Kamera Temelli Farklı Yaya Tespit Yöntemleri [5]	8
Tablo 2-4 Yön Vektörlerinin Hesabı için Kullanılan Türev Maskeleri	14
Tablo 2-5 Dikdörtgen Alana Sahip Haar Özellik Çekirdekleri [29]	21
Tablo 3-1 Sınıflandırıcı Eğitiminde Kullanılan Parametre Setleri	33
Tablo 3-2 Birleştirilen Tespitler için Kullanılan Skor Tablosu	40
Tablo 4-1 Testlerde Kullanılan Parametre Setleri	43
Tablo 4-2 CPU Tabanlı Sistemdeki Donanım ve Yazılım Yapısı	45
Tablo 4-3 CPU Tabanlı İşlemci için 3 Yöntemin Yaya Tespit Rakamları	48
Tablo 4-4 GPU ve CPU Tabanlı Sistemdeki Donanım ve Yazılım Yapısı	50
Tablo 4-5 GPU/ CPU Tabanlı İşlemciler için 3 Yöntemin Yaya Tespit Rakamları	53
Tablo 4-6 Mobil İşlemci Donanım ve Yazılım Yapısı	54
Tablo 4-7 ARM Tabanlı Mobil İşlemci için 3 Yöntemin Yaya Tespit Rakamları ...	57

ŞEKİLLER

Şekil 2-1 Yön Vektörü Histogramı	15
Şekil 2-2 HOG Tanımlayıcısı için Kullanılan Hücre ve Blok Yapısı.....	17
Şekil 2-3 Bir Veri Setinin DVM ile 2 Farklı Kümeye Ayrılması	20
Şekil 2-4 Tümlşik Görüntü Üzerindeki Taralı Alan	22
Şekil 2-5 Kuvvetlendirilmiş Basamaklı Sınıflandırıcı Sistem Akış Şeması	24
Şekil 3-1 Geliştirilen Yaya Tespit Algoritmasının Genel Akış Şeması	25
Şekil 3-2 Alt, Üst ve Tüm Vücut Kısımlarının Ayrı Tespiti (Görüntü kaynağı: [33])	26
Şekil 3-3 HOG Yönteminde Kullanılan Birleştirilmiş Tespitler (Görüntü kaynağı: [33])	30
Şekil 3-4 Sınıflandırıcı Eğitimi Safhaları	31
Şekil 3-5 Sınıflandırıcı için Kullanılan Örnek Bir Yaya Görüntüsü (Görüntü kaynağı: [38])	32
Şekil 3-6 Örnek Görüntüdeki Yaya Boyutları ve Konumları	33
Şekil 3-7 İlgilenilen Alanın Ön Bilgi ile Sabit Belirlenmesi (Görüntü kaynağı: [33])	35
Şekil 3-8 Dinamik İlgilenilen Alan Seçimi Yaklaşımı	36
Şekil 3-9 Tüm, Alt ve Üst Vücut Tespitleri Histogramı ile Dinamik Alan Seçimi....	37
Şekil 3-10 Aynı Yaya Ait Farklı Tip Tespitlerin Birleştirilmesi.....	38
Şekil 3-11 Tespit Edilen Yayaların Nihai Karar Aşaması Akış Şeması.....	40
Şekil 4-1 CPU Tabanlı Sistemdeki Yaya Tespit Yazılımı Mimari Yapısı	45
Şekil 4-2 SIMD Mimari Yapısı [41].....	46
Şekil 4-3 CPU Tabanlı İşlemci için 3 Yöntemin Karşılaştırılması.....	47
Şekil 4-4 CPU Tabanlı İşlemci için Klasik HOG ve Haar Yöntemi Sonuçları	49
Şekil 4-5 GPU ve CPU Yaya Tespit Yazılımı Mimari Yapısı.....	51
Şekil 4-6 GPU ve CPU Tabanlı İşlemci için 3 Yöntemin Karşılaştırılması	52
Şekil 4-7 ARM İşlemcili Yaya Tespit Yazılımı Mimari Yapısı	55
Şekil 4-8 ARM Tabanlı İşlemci için 3 Yöntemin Karşılaştırılması	56

SİMGELER VE KISALTMALAR

Kısaltmalar

CPU	Merkezi İşlemci Ünitesi (Central Processing Unit)
CVPR	Bilgisayarlı Görüntüleme ve Örüntü Tanıma Konferansı (Conference on Computer Vision and Pattern Recognition)
DVM	Destek Vektör Makinesi
FPGA	Proglamlanabilir Mantık Devreleri (Field Programmable Gate Array)
fps	Saniyede İşlenen Çerçeve Sayısı (Frame per second)
FUL	Full Upper Lower (Tüm Üst Alt)
GPU	Grafiksel İşlemci Ünitesi (Graphical Processing Unit)
HMM	Saklı Markov Modeli (Hidden Markov Model)
HOG	Yön Vektörlerinin Histogramı (Histogram of Gradients)
IA	İlgilenilen Alan
INRIA	Fransız Ulusal Bilgisayar ve Kontrol Mühendisliği Araştırma Enstitüsü (French National Institute for Research in Computer Science and Control)
JNI	Java Doğal Arayüz (Java Native Interface)
SIFT	Boyuttan Bağımsız Özellik Dönüşümü (Scale Invariant Feature Transform)
SIMD	Tek Komut ile Çoklu Veri İşleme (Single Instruction Multiple Data)
SMM	Saklı Markov Modeli
SVM	Destek Vektör Makinesi (Support Vector Machine)

1. GİRİŞ

Her yıl trafik kazaları sebebi ile dünyada 39000 yaya ölmekte ve 430000 yaya yaralanmaktadır. Yayalarla ilgili kazalar, trafikteki en büyük ikinci seviye ölümleri ve yaralanmaları oluşturmaktadır [1]. Yaya güvenliği için araçlarda ve yollarda bazı iyileştirmeler yapılırsa da, kötü sürüş koşullarından (düşük ışık, sis, gölge ve yoğun trafik ortamı), yol şartlarından ya da sürücünün veya yayanın anlık dikkat dağınıklığından dolayı kazalar gerçekleşmektedir.

Araçlar için otomatik yaya tespit, uyarı ve kaza önleme sistemleri, araç üreticilerinin ve teknoloji geliştirenlerin yaya ölümlerinin ve yaralanmalarının önüne geçmek için geliştirdikleri önemli sistemler arasında yerini almaktadır. Bu amaçla her geçen gün artan sayıda olmak üzere üretilen yeni nesil araçlarda bu tip güvenlik sistemlerine yer verilmektedir. Örneğin Mercedes firmasının kaza uyarı ve önleme kapsamında geliştirdiği teknolojiler arasında önleyici fren teknolojisi, sesli sürücü uyarı sistemi, gece görüş sistemi gibi sistemler yer almaktadır [2].

Yaya tespit sistemleri için özel radar sistemleri, lazer menzil bulucular, video ve kızılötesi kameralar ilgi çekici ve gelecek vaat eden çözümler sunmaktadır. Bu sistemlerden her birinin ortamdaki ve uygulama şartlarından kaynaklı eksiklikleri ve kusurları olabilmektedir. Bu yüzden bu sistemlerden ikisi veya üçü bir arada kullanılarak algılayıcı tabanlı birleştirme yöntemleri ile nihai çözüme ulaşılmaktadır.

Temel olarak -ışığın yeterli olduğu şartlarda- insan görme, algılama ve karar verme sistemi taklit edilebilirse, video kameralar yaya tespit sistemleri için doğal bir çözüm olarak kullanılabilir. Yeterli çözünürlüğe sahip video kameralara gerekli ve yeterli işlem yeteneklerinin eklenmesiyle yaya tespit sistemleri oluşturulabilir.

Video kameraların kolayca erişilebilir olması ve mobil cihazlarda yer alması tasarlanacak bir yaya tespit sisteminin yaygınlaştırılması bağlamında radar, lazer, kızılötesi kamera vb. kullanan sisteme sahip çözümlere göre üstünlük sağlamaktadır. Buna ek olarak video kameralar ortama herhangi bir işaret yayımlamadıklarından dolayı ortamdaki elektromanyetik ve termal gürültü ile etkileşime girmeleri de mümkün değildir [3]. Video kameraların ortamdaki elektromanyetik ve termal gürültüden etkilenmemesi radar sistemlerine ve kızılötesi kameraya sahip sistemlere göre önemli bir üstünlük sağlamaktadır [4].

Görüntü tabanlı yaya tespiti belirli sebeplerden dolayı zor bir problemdir. Bu sebepler sırayla şunlardır: yayalar oldukça karmaşık arka planlarda, farklı görüntülerde, pozlarda, boylarda, elbiselerde ve farklı ışık durumlarında yer alabilmektedir. Aynı zamanda yaya kameradan oldukça uzakta olabilir. Bu sebepten yayayı arka plandan ayırt etmek zor olabilir. Görüntü üzerinde peş peşe gelen görüntüler üzerinden yapılacak bir arka plan çıkarma işlemi hareketli ve devinimli kamera yapısından dolayı uygulanamaz ve doğru sonuç vermez. Sonuç olarak gerçek zamanlı tespit koşulu sebebi ile kaba kuvvet yöntemleri hız ve performans açısından beklenen sonuçları vermekten uzaktır [3]. Bütün bu problemleri de göz önüne alarak nesne tespitinde sıkça kullanılan HOG (Histogram of Gradients) ve Haar yöntemleri birlikte kullanılarak özel bir yaya tespit algoritması geliştirilmiştir. Algoritma geliştirilirken mekânsal ve parçasal bilgiden azami faydalanmak için ön bilgi olarak alt, üst ve tüm vücut bölümleri için farklı ilgilenilen alan seçimleri ve yaya tespitleri gerçekleştirilmektedir. Geliştirilen sistem ile tespit hızı ve doğruluğu açısından gerçek zamanlı bir sistemden beklenen sonuçlar elde edilmektedir.

Bu tez çalışmasının organizasyonu şu şekildedir. Bölüm 2 literatür taraması ile birlikte tezde kullanılan yaya tespit metotlarını içermektedir. Bölüm 2.1'de literatürdeki farklı yaya tespit yöntemleri kısaca özetlendikten sonra video kamera tabanlı yaya tespit yönteminin detaylı bir literatür taraması verilmektedir. Bölüm 2.4'te ise tezde kullanılan ve gerçek zamanlı yaya tespit metotları olan HOG ve Haar metotları teori ve algoritma düzeyinde verilmektedir. Bölüm 3'te tez kapsamında geliştirilen gerçek zamanlı yaya tespit algoritması verilmektedir. Bu bağlamda geliştirilen algoritmanın sistem akışı, yöntemler ve sistemin uygulanışı detaylandırılmaktadır. Bölüm 4'te geliştirilen algoritmanın farklı platformlarda uygulanması anlatılmaktadır. Bu kapsamda CPU, GPU ve mobil işlemciye sahip sistemlerde geliştirilen algoritma test edilerek farklı sistemlerdeki yaya tespit hızı ve tespit oranı karşılaştırması yapılmaktadır. Son olarak Bölüm 5'te yapılan tez çalışması ile ilgili sonuçlar ve öneriler verilmektedir.

2. YAYA TESPİT YÖNTEMLERİ ve YAKLAŞIMLARI

Bu bölüm genel (kızılötesi kamera, lazer, radar vs.) ve özel (video kamera) literatür taraması ile birlikte gerçek zamanlı yaya tespit yaklaşımlarını ve tez çalışmasında kullanılan yöntemleri anlatmaktadır. Bu bölüme ait alt bölümlerden ilki olan Bölüm 2.1'de literatürde kullanılan farklı yaya tespit yöntemlerinden bahsedilmektedir. İkinci alt bölüm olan Bölüm 2.2'de bir yaya tespit sistemi olan video kamera tabanlı yaya tespit sistemlerine ait daha derinlikli literatür analizi sunulmaktadır. Bu alt bölümden sonra, Bölüm 2.3 ve 2.4'te sırayla gerçek zamanlı yaya tespit yaklaşımları ve tezde kullanılan yaya tespit yöntemleri (HOG ve Haar) teori ve algoritma düzeyinde verilmektedir.

2.1 Yaya Tespit Sistemlerinde Kullanılan Farklı Yöntemler

Yaya tespit sistemlerinde donanım ve yazılım sistemleri farklı olan yöntemler kullanılmaktadır. Kullanılan sistemlere bağlı olarak farklı algoritma yaklaşımları ön plana çıkmaktadır. Kullanılan donanım/yazılım sistemleri arasındaki temel farklılık olarak kullanılan sensorler ve sensorlerden alınan veriyi işlemek için kullanılan algoritmalar ön plana çıkmaktadır.

Yaya tespit sistemlerinde kullanılan sensorler arasında video kamera, yakın kızılötesi, termal kızılötesi, RADAR ve LAZER tarayıcı sensorleri vardır. Her bir sensorun mesafe çözünürlüğüne, tespit mesafesine, donanım maliyetine ve sensor bilgisinin işlenmesinde kullanılacak algoritmanın karmaşıklığına bağlı olarak farklı kullanım alanları vardır. Farklı sensorler ile geliştirilen sistemlerin farklı ortam, şart ve platformlarda birbirine üstünlükleri ve zayıflıkları vardır.

Yaya tespit sistemlerinin kullanılacağı çevre koşulları (sıcaklık, nem, rüzgâr vs.), ortam aydınlatması ve tespit mesafesi hangi sistemin tercih edileceği konusunda önemli girdiler olarak öne çıkmaktadır. Örneğin ortam sıcaklığının yüksek olduğu bir ortamda kızılötesi kamera etkili bir tespit sonucu üretemeyecektir. Bir diğer örnek de RADAR için verilebilir: ortamdaki elektromanyetik kirlilik yüksek ise RADAR tespit performansı düşecektir [4]. Sonuç olarak, yaya tespit sistemlerinde kullanılacak olan sensor sistemleri ve sensor sistemine bağlı olarak üretilecek yazılım ve donanım sistemi kullanım amacına ve ortamına bağlı olarak değişiklik göstermektedir.

Farklı sensor sistemlerine ait farklı özellikler Tablo 2-1’de verilmektedir. Sensorler arasında karşılaştırılan özellikler: görüş açısı, açısal çözünürlük, tespit mesafesi, tespit çözünürlüğü, aydınlatma, donanım maliyeti ve algoritma karmaşıklığıdır.

Tablo 2-1 Yaya Tespit Sistemlerindeki Sensor Sistemlerinin Karşılaştırması [5]

Sensor Türü	Görüş Açısı	Açısal Çözünürlük	Tespit Mesafesi	Tespit Çözünürlüğü	Aydınlatma	Donanım Maliyeti	Algoritma Karmaşıklığı
Video Kamera	Orta	Orta/Yüksek	Kısa/Orta	Orta	Pasif, ortam ışığı gerekli	Düşük	Yüksek
Geniş Görüş Açılı Video Kamera	Geniş	Düşük/Orta	Kısa	Düşük	Pasif, ortam ışığı gerekli	Orta	Yüksek
Yakın kızılötesi	Orta	Orta/Yüksek	Orta	Orta	Aktif aydınlatma, karanlıkta çalışır	Düşük	Yüksek
Termal kızılötesi	Orta	Düşük/Orta	Kısa/Orta	Düşük	Yayıcı, karanlıkta çalışır	Yüksek	Orta
RADAR	Dar	Düşük	Uzun	Yüksek	Aktif; karanlıkta, yağmurda, siste çalışır	Orta	Düşük
LAZER tarayıcı	Geniş	Orta	Orta	Yüksek	Yayıcı, karanlıkta çalışır	Yüksek	Düşük

Literatürde yukarıdaki tabloda yer alan sensor sistemleri ile ilgili onlarca çalışma vardır. Literatürde yer alan çalışmaların birinde kızıl ötesi kamera kullanan gece görüş sistemleri kullanılmıştır. Gece görüş sisteminde, kızıl ötesi kameradan elde

edilen stereo görüntüden arka plan hareketinden farklı olan hareketli nesnelere tespit edilir. Bu teknikte sıcak nokta olarak gözüken yayaların karakteristik izi yayayı algılamada ve tespitinde kullanılır. Gündüz ise bu sistem etkili bir kullanım alanına sahip değildir [6].

Literatürde yer alan kızılötesi kamera ile yapılan bir diğer yaya tespit çalışmasında, her bir pikselin yoğunluk değeri hedefin yaya olup olmadığının belirlenmesinde kullanılır. Hedef nesnelere, basit yoğunluğa bağlı eşik fonksiyonu kullanılarak görüntüden elde edilir. Arka plana ait olduğu tahmin edilen piksellerin değeri '0' ile sıcaklık yayan nesnelere ait piksellerin değeri '1' ile değiştirilir ve eşik fonksiyonu ile değeri 1 olan piksellerin oluşturduğu bölgelerden yaya tespit edilir [7].

Literatürde LAZER, RADAR ve kızılötesi kamera ile yapılan çalışmaların kullanım amacı, sensor sistemi, yaklaşım ve tanımları Tablo 2-2'de verilmektedir. Çalışmaların birçoğunda şekil, özellik çıkarımı ve hareket tespit tabanlı yaklaşımlar kullanılmaktadır.

Tablo 2-2 LAZER, RADAR ve Kızılötesi Kamera ile Yaya Tespit Yöntemleri [5]

Makale	Amaç	Sensor	Yaklaşım	Tanım
Broggi IV04	Tespit	Uzak kızılötesi	Şekil	Sıcak, simetrik ve belli boyut ve orandaki nesnelere farklı çözünürlükte bulur.
Fang VT04	Tespit	Uzak kızılötesi		Dikey ve yatay izdüşümü yaklaşımı ile şekilden bağımsız tespit yapar.
Meis IV04	Tespit	Uzak kızılötesi	Kafa tespiti	Piksel sınıflandırmada istatistiksel bir yaklaşım ile kafa tespiti yapar.
Xu ITS05	Tespit, izleme	Uzak kızılötesi	Şekil	DVM, Kalman filtre kullanarak yayaları izler. Yol tanıma ile geçerlilik testini yapar.

Liu VT04	Tespit	Uzak kızılötesi	Stereo, hareket	Hareketli nesnelere tespit eder.
Tsuji ITS02	Tespit, Çarpışma önleme	Uzak kızılötesi	Stereo, hareket	Basit kızılötesi tabanlı tespit, izleme ve göreceli hareket vektörlerinin hesaplanması ile tespit yapar.
Fang 03	Tespit karşılaştırma	Görünür, Uzak kızılötesi	Özellik tabanlı	Görünür ve kızılötesi banttaki kameraları karşılaştırır. Çok boyutlu özellik sınıflandırma ile tespit yapar.
Milch	Tespit	RADAR, video kamera	Şekil, Görünürlük zamanı	Yaya listesi RADAR ile elde edilir. Yayalar video kamera üzerinde şekil tabanlı yaklaşım ile onaylanır.
Scheunert	İzleme	Uzak kızılötesi, LAZER tarayıcı	Sıcak nesne tespiti, Görünürlük zamanı	LAZER tarayıcı tespit yapar. Kızılötesi kamera ile parlaklık ve yön ayarı yapılır. Sensor bilgileri birleştirilerek tespit yapılır.

Video kameralara kolay erişim, kurulum ve kullanım sebebi ile video kamera yaya tespit sisteminin yaygınlaştırılması bağlamında RADAR, LAZER, kızılötesi vb. çözümlere göre üstünlük sağlamaktadır. Kızılötesi kameraların ortam sıcaklık değişimlerine duyarlı olması, insan vücut sıcaklığı ile ortam sıcaklığının yakın olduğu yaz günlerinde tespit zorlaşması sebepleri gibi benzer sebeplerden dolayı kızılötesi kamera video kameraya göre bazı dezavantajlar içermektedir. Buna ek olarak video kameralar ortama herhangi bir işaret yayımlamadıklarından ve ortamdaki herhangi bir işaret almadıklarından dolayı video kameraların ortamdaki elektromanyetik ve termal gürültü ile etkileşime girmeleri de mümkün değildir [3]. Bu sebeple video kamera yaya tespit problemi için uygun bir çözüm

olarak ön plana çıkmaktadır. Video kamera ile ilgili detaylı literatür çalışması bir sonraki alt bölümde verilmektedir.

2.2 Video Kamera Tabanlı Yaya Tespit Sistemleri

Bu bölümde video kamera kullanılarak yapılan yaya tespit yaklaşımları ile yaya çarpışma analizleri literatürde yer alan çalışmalar bağlamında ele alınmaktadır. Öncelikle tespiti yaklaşımları anlatılacak; sonraki kısımda ise çarpışma tahmin yaklaşımları ve yaya davranışı/hareketi analizi yaklaşımları ele alınmaktadır.

2.2.1 Video Kamera Tabanlı Yaya Tespit Sistemlerindeki Yaklaşımlar

Bu bölümde, video kamera kullanılarak gerçekleştirilen yaya tespit sistemlerindeki yaklaşımlar anlatılacaktır. Bunun için literatürde yer alan çalışmaların bir listesi sunulacaktır.

Video kamera kullanımında iki temel yöntem esas alınır. Birinci yöntem, tek bir görüntü çerçevesi üzerinde yaya tespit ve teşhisine odaklanır. İkinci yöntemde ise peş peşe gelen görüntü çerçevelerini de kullanarak hareket tabanlı çözümler ile yaya tespit ve teşhisi gerçekleştirir. Genel olarak şekil, hareket, derinlik bilgisi kullanılarak görüntü üzerinden elde edilen özellik parametreleri/setleri herhangi bir sınıflandırıcı vasıtası ile (yapay sinir ağı, destek vektör makinesi veya güçlendirilmiş basamaklı sınıflandırıcı) yaya tespiti yapacak bir tanımlayıcıya dönüştürülür. Elde edilen tanımlayıcı vasıtası ile sistem, farklı gölge, ışık, yoğunluk ve hareket tiplerine göre gerçek ortamda yaya tespit ve teşhisinde kullanılır.

Yaya tespit sistemlerinde algılama safhasında hareket tanıma ve takibi, mesafe temelli yaklaşım ile arka plan çıkarımı ve şekil temelli yöntemler en çok kullanılan yöntemlerdir [8]. Yürüyüş analizi ve şekil modelleri ise tespit safhasında sıkça kullanılmaktadır. Hareket bilgisi, peş peşe gelen çerçevelerdeki olası bölgelerin analizi ile çıkarılmaktadır. Peş peşe gelen iki görüntü çerçevesinde çerçeve üzerinde sabit bir konuma sahip olan yayaları tespit etmek hareket bilgisi çıkarımı ile mümkün olmamaktadır [9]. Bu yöntemde yayaya ait hareket bilgisi üzerinden yapılan yaya tespiti sadece hareket eden yayalar için kullanılmaktadır.

Stereo görüntü tabanlı yaya tespit yöntemi, ön plandaki nesnelere arka plandan ayırmak için kullanılan bir yöntemdir. Bu yöntem ışık, gölge ve üst üste binmelerden az etkilenmektedir. Öncelikle birbirinden belirli mesafe ile ayrı olan iki video kamera ile derinlik bilgisi çıkarılarak ön plandaki nesnelere ortaya çıkarılır.

Yapay sinir ağı ile eğitilen sistem ile nesnelere arasından yaya olanları diğerlerinden ayırırılır [10].

Başka bir tür yaya tespit yöntemi olan şekil tabanlı yaklaşımda, görüntü üzerinde önceden veri tabanına girilen insan şekli/silueti örüntüsü taranır. Bu yaklaşım, sabit olmayan arka plan problemiyle uğraşmadığı gibi yaya hareketine ait bilgiyi çok az kullanır. Bu algoritmada insan şekli ile ilintili ön bilgi kullanılmazsa hesaplamalar çok maliyetli olabilir [11].

İnsan hareketinin periyodik olması yürüyen yayaları tanımlamada, özellikle bakış açısına göre yatay yürüyen yayalarda, önemli bir ipucudur. Sistemin düzgün bir tanımlama yapabilmesi için önce birkaç çerçevenin analiz edilmesi ve güvenilir bir şekilde yayanın görüntü üzerinde takibi gerekmektedir. Bu yöntem, yayanın yatay yürümesine sıkı olarak bağlıdır ve sabit duran yayaların bu yöntemle tanımlanması mümkün değildir [12].

Hareket bilgisi, peş peşe gelen çerçevelerdeki olası bölgelerin analizi ile çıkarılmaktadır. Peş peşe gelen iki görüntü çerçevesinde sabit konuma sahip olan ya da hareketsiz gibi gözükken yayaları tespit etmek hareket bilgisi çıkarımı ile mümkün olmamaktadır [9]. Bu yüzden tek bir görüntü çerçevesi üzerinden yaya tespiti yapmak her durumda (hareketli veya sabit yaya) yaya tespit problemi için genel geçer bir çözüm olarak ön plana çıkmaktadır.

Literatürde yer alan çalışmaların yaklaşım ve tanımları Tablo 2-3'te verilmektedir. Çalışmaların birçoğunda, şekil ve hareket tabanlı yaklaşımlar kullanılmaktadır.

Tablo 2-3 Video Kamera Temelli Farklı Yaya Tespit Yöntemleri [5]

Makale	Yaklaşım	Tanım
Papageorgiou IJCV00	Şekil Sınıflandırma	Yerel yoğunluk farklarını HAAR dalgacık dönüşümü ile ele alır. Sınıflandırma için DVM kullanır.
Abramson IV04	Şekil, Hareket	Şekil ve hareket tabanlı bir algoritma kullanır. Çarpışma öngörüsü parçacık filtreleme ile modellenir.
Hashiyama CSMC03	Hareket, Şekil	Hareketli video kamerada aktif arka plan çıkarımını uygular. Hareket tahmini jiroskop ile yapılır. Yaya algılama için son olarak örüntü tanıma uygulanır.

Viola ICCV03	Hareket, Şekil	Görünüm ve hareket bilgisini tek bir detektörle birleştirir. ADA-BOOST ile şekil ve harekete dayalı çıkarılan özellikleri sınıflandırmada kullanır.
Havasi 04	Şekil, Hareket	Hedef nesnenin yapısal, özellikle bacaklar, özelliklerini kullanarak özellik çıkarımı ve takibi sağlanır. Nesnenin takibi ile çıkan izleri takip ederek yaya tespitini hedefler.
Shashua 04	Şekil, Hareket	Sınıflandırmayı tek bir çerçeve üzerinde yaparken, çoklu çerçeve kullanımı ile yürüyüş örüntülerinin, uzaklık-yakınlık farkının doğrulamasını yapar.
Zhao ITS00	Stereo, Yapay Sinir Ağı	Video görüntüsündeki farklılıkların devamsızlığı ile imgeyi parçalar. Parçala-birleştir yöntemi ile büyüklük/şekil kısıtlamaları ile yayaları oluşturur. Yapay sinir ağı ile yoğunluk yön vektörleri ile yaya tespitini gerçekleştirir.
Gavrila IV04	Stereo, Şekil	Stereo tabanlı derinlik ayrıştırması yaparak CHAMFER karşılaştırması ile şekilleri sınıflandırır. İmge dokusu sınıflandırması yapay sinir ağının sınıflandırmasında kullanılır.
Hilario 05	Şekil, Stereo	Aktif dış hat modeli ile yayaları imgeden ayrıştırır. Stereo aktif dış hat pozisyonunu belirlemede kullanılır çünkü bu yöntem başlangıç noktasına oldukça bağımlıdır.
Gandhi ICIP05	Stereo	Bir çift video kamera ile stereo yöntemi ile yayaları ve diğer nesnelere ayrıştırır ve tespit eder.
Lombardi IV04	Şekil, Hareket, Baş Tespiti	SMM ile görüntüdeki değişimleri modeller. Farklı şartlar altında (araç hızına ve sahne karmaşıklığına göre) ihtiyaç duyulan algoritmaları kullanır.

Tablo 2-3'te video kamera üzerinden elde edilen görüntüden yaya tespiti ve algılamaya yönelik farklı yaklaşımlar verilmektedir. Şekil ve hareket temelli yaklaşımlar çoğunlukta olmakla beraber bu yaklaşımlara ilave olarak performans ve doğruluk artırımı için görüntüde derinlik, yaya yürüyüş şekli de ilave edilmektedir.

Video çerçevesindeki yayayı algıladıktan sonra yayayı takip edip araç ile çarpışma ihtimalini ve senaryosunu hesaplamak için ilave yöntemler ve hesaplamalar yapılmaktadır. Bu yaklaşımlar bir sonraki bölümde ifade edilmektedir.

2.2.2 Video Kamera ile Çarpışma Tahmini ve Yaya Davranışı Analizi

Etkili bir çarpışma önleme sistemi için, yaya algılamayı takiben çarpışma tahmini etkin bir şekilde kullanılmalıdır. Gece görüş sistemi ile yaya tespitini yapan bir sistemde [13], araç ve yayanın birbirlerine göreceli hareketi çarpışma tahmini için kullanılmaktadır. Bu yaklaşımda araç veya yayanın yön ve hızının çarpışma anına kadar değişmeyeceği varsayılmaktadır. Bu model ancak yüksek hızlarda çarpışmaya az bir zaman kala hız değişiminin çarpışma durum ve şartlarını etkilemeyeceği varsayılarak oluşturulmuştur. Ancak, kavşak noktalarında yayaların geçiş anında, yayaların ani manevra yapabilecekleri düşünüldüğünde yaya davranışı ve hareketleri olasılıklar bağlamında modellenmelidir. Monte-Carlo benzeşimleri ile farklı olasılıklar üretilerek olası çarpışma sonuçları ve senaryoları tespit edilerek en muhtemel senaryoya göre çarpışma olup olmayacağı tespit edilmektedir.

Yaya dinamiğinin 4 farklı şekilde modellendiği bir çalışmada; sabit durma, yürüme, yavaş koşma ve koşma esas alınmıştır [14]. Her bir durum için olasılık dağılımı çıkarılarak Monte-Carlo benzeşimleri ile farklı olasılıklar üretilerek olası çarpışma sonuçları tespit edilmektedir.

Parçacık filtreleme yöntemi ile yaya takibi yapılarak çarpışma tahmini yapılmaktadır [15]. Bir adım sonrasını takip etme yerine yaya hareketlerinin uzun vadeli değişimine bağlı olarak tahmin yürütülmeye çalışılır.

Yaya bedeninin yönü genel olarak yayanın gelecekteki hareketinin yönünü belirler. Yapılan bir çalışmada, DVM sınıflandırıcısı kullanılarak yayanın beden yönü üzerinden çıkarılan HAAR dalgacık katsayıları sınıflandırılarak yayanın hareket yönü kestirilmektedir [16].

Sabit kamera düzeninde, yayaların takip ettiği belirli yollar öğrenilerek yayaların gelecekteki pozisyonu için kestirme yapılabilir. Yapılan bir çalışmada yayaların yörüngelerinin olasılık modelleri Bayesyan HMM ile çıkarılarak yayaların gelecekteki pozisyonu için kestirilmektedir [17].

2.3 Gerçek Zamanlı Yaya Tespit Yaklaşımı

Tek bir kamera kullanılarak gerçekleştirilen yaya algılama ve tespiti üzerinde yapılan çalışmalarda belirli kurallar ve kısıtlamalara uyulduğu takdirde sistem hız ve algılama kalitesinden ödün vermeden sistemin gerçek zamanlı olarak hayata geçirilebileceği öngörülmüştür [1].

Gerçek zamanlı sistemler üzerinde yapılan çalışmalarda probleme en iyi yaklaşımın ne olduğu, var olan yaya detektörleri yeterli mi, tespitite yapılan hataların kaynağı ne, en faydalı ilgili bölge analizi gibi sorulara verilen cevaplar çok önemlidir. Sistemin her bir parçasının mümkün olan en iyi biçimde seçilmemesi durumunda, sistemde tespit hızı ve tespit kalitesi/doğruluğu sorunları ile karşılaşmaktadır.

Gerçek zamanlı ve yüksek performanslı bir yaya tespit ve uyarı sisteminin tasarlanması için seçilen yöntemler değişmekle birlikte genel olarak gerçek zamanlı ve iyi tespit sonucu elde etmek için uyulması gereken bazı temel kurallar vardır. Bu kurallar takip eden alt bölümlerde detaylı bir şekilde anlatılmaktadır.

2.3.1 Özellik Çıkarım Yöntemi

Görüntü üzerinde hesaplanması düşük maliyette olan özellikleri çıkarmak hızlı ve iyi bir tespit için önemlidir [18]. Özelliklerin çıkarılması için harcanan süre, tespit için harcanan sürenin önemli bir kısmını kapsamaktadır.

Tümleşik görüntü yöntemi (Bkz. 2.4.2.2.2) ile Haar özellikleri kullanırken tespit hızında önemli bir hızlanma elde edilmiş olur. Çünkü Haar yönteminde, yaya tespitinde bütünleşik görüntüden elde edilen matris kullanılır ve elde edilen matris kullanılırken matris üzerinde çıkarma, toplama gibi basit aritmetik işlemler yapılarak özellikler elde edilir. Bütünleşik görüntü kullanarak dikdörtgenlerin ortalamasını almak [19] veya türevsel yön bilgisini tespit amaçlı kullanmak [20] hızlı özellik çıkarımları için bir örnektir.

Sonuç olarak, özellik çıkarımındaki hesaplama süresini azaltmak, mümkün olduğunca daha düşük karmaşıklığa sahip hesaplamalar yapmak ve hesapları ortaklaştırmak özellik çıkarımında gerekli hızlandırmayı sağlamak için gereklidir.

2.3.2 Özellik Sınıflandırıcı Yöntemi

Belirli bir özellik seti için kullanılan sınıflandırıcı tipi tespit için gerekli hız ve kaliteyi doğrudan etkiler. Genelde hız ve kalite arasında ters orantı vardır [18]. Doğrusal

olmayan sınıflandırıcılar doğrusal sınıflandırıcılara göre daha iyi kaliteye sahip olmakla beraber yavaşlırlar. Doğrusal sınıflandırıcılar ise, kaliteden ödün vermekle birlikte daha yüksek performansa sahiptir. Doğrusal sınıflandırıcı olarak genelde ADABOOST ve SVM kullanılır [21].

2.3.3 Ön Bilgi Kullanma Yöntemi

Görüntü işlemede ön bilgi kullanımı önemlidir. Bu sayede gereksiz işlem yükünden ve hatalı tespitten kaçınılmış olur. Yaya tespiti için tek bir baskın zemin düzleminin bilinmesi ön bilgi olarak hız ve kalite artışında önemli bir etken olarak kullanılmaktadır [22]. Görüntü üzerindeki muhtemel yaya konumlarının önceden bilinmesi de ön bilgi olarak oldukça faydalı olmaktadır. Bu çalışmada, Bölüm 3.4'te detayları verilen ilgilenilen alan seçimi yaklaşımı ön bilgi kullanımına bir örnektir.

2.3.4 Basamaklı Sınıflandırıcı Kullanmak

Sınıflandırıcıyı aşamalara ayırıp daha basit sınıflandırıcıları ön sıraya sokmak hem performansı artırır hem de hatalı tespitlerin en başta fazla hesaplama yapılmadan elenmesini sağlanmış olur. Bu sayede toplam hesaplama zamanı düşerken hatalı tespitler de elenmiş olur [19]. Bölüm 2.4.2.2.3'te basamaklı sınıflandırıcı yapısı daha detaylı bir şekilde ele alınmaktadır.

2.3.5 Önceliklendirme ve Sınır Koyma

Her bir hipotezi denemek yerine en muhtemel olanları seçip, seçilenleri sıraya koymak algılama hızını artırıp önemli bir tespit kaybı olmadan tespit kalitesini devam ettirir [23]. Bu çalışmada, Bölüm 3.6'da verilen skor kriteri ile birleştirilen tespitleri önceliklendirme yöntemi bahsedilen tespit kalitesini artırmaya yönelik bir yaklaşımdır. Yaklaşımın detayları daha sonraki ilgili bölümde detaylı olarak verilmektedir.

2.4 Tez Çalışmasında Kullanılan Yöntemler

Tez çalışmasında alt, üst ve tüm vücut kısımlarını ön tespit için Haar özellikleri tabanlı tanımlayıcı kullanılırken Haar özellikleri yapılan tespitler birleştirilip filtreledikten sonra nihai karar verici olarak HOG tanımlayıcısı kullanılmaktadır. Her iki tanımlayıcı da tek bir görüntü çerçevesi üzerinde çalışmaktadır. Yöntemlerin teori ve algoritma yapısı takip eden alt bölümlerde sunulmaktadır.

2.4.1 HOG Tanımlayıcısı

2.4.1.1 Teori

HOG görüntü işlemede nesne tespiti için kullanılan bir yöntemdir. Temel olarak bir görüntünün alt hücrelere ve bloklara bölünüp her bir alt bloktaki yön vektörlerinin açısal değerinin belirli açı aralıkları kullanılarak histogram ile hesaplanması esasına dayanır.

INRIA'da çalışan iki araştırmacı Dalal ve Triggs 2005'te yayınladıkları çalışmada HOG yöntemini sabit görüntülerde yer alan yayaları tespit etmek için tanımlamıştır [24]. HOG yöntemi yaya tespitinden sonra hayvan ve otomobil tespiti için de kullanılarak HOG yönteminin kullanım alanı genişletilmiştir.

HOG tanımlayıcılarının arka planında yatan esas fikir, herhangi bir objenin şekil ve görünümünün yoğunluk yön vektörlerinin bir diğer ifade ile kenar yönlerinin dağılımı ile ifade edilebileceği fikridir. Bu fikri gerçeklemek için görüntü hücre adı verilen 9x9 veya 16x16 piksellik alt parçalara bölünerek her bir hücreye ait yön vektörleri hesaplanır. Farklı hücrelerdeki yön vektörleri birleştirilerek tanımlayıcı oluşturulur. Tanımlayıcı hesaplanırken ışık yoğunluğu, gölgeleme ve arka plan gibi faktörleri en aza indirmek için her bir hücredeki piksel değerlerinin ortalaması kullanılarak hücrelerden oluşan belirli sayıda bir bloka normalleştirme faktörü uygulanır. Normalleştirme sayesinde, HOG tanımlayıcısı aydınlatma ve gölgelemeden kaynaklı değişimlere karşı tolerans sağlar.

HOG tanımlayıcısı, diğer tanımlayıcılara göre belirli avantajlar içerir. Tanımlayıcı yerel hücreler üzerinden hesaplandığı için nesnenin açısal pozisyonundaki değişim hariç geometrik ve foto-metrik değişimlere karşı iyi sonuç verir. Yani ayakta duran bir insan şekli olduğu müddetçe yayanın anlık hareketi çok da önemli değildir.

HOG tanımlayıcısı bütüncül yapıda bir tanımlayıcıdır. Yani yaya bedeninin tümünü ifade eden bir tanımlayıcıdır. Bu yöntemde temel amaç insan vücudunun tümünü tanımlayan bir tanımlayıcı vektör elde etmek ve bu vektörü farklı boyuttaki yayaları tanımlamada kullanmaktır.




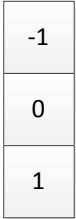
2.4.1.2 Algoritma

2.4.1.2.1 Yön Vektörlerinin Hesaplanması

Görüntü ön işlemede kullanılan birçok özellik tanımlayıcıda dikkat edilmesi gereken iki özellik vardır: renk ve parlaklık. HOG tanımlayıcı kullanılırken ön işleme gerekli değildir. Çünkü tanımlayıcı hesaplanırken kullanılan normleştirme işlemi bunu gereksiz kılar. Bu sebeple HOG yönteminde, görüntü işleminin ilk adımı renk ve parlaklık için ön işleme değil direkt olarak yön vektörlerini hesaplamadır.

Yön vektörlerini hesaplamak için x ve y eksenlerinde tek boyutlu bir türev maskesi uygulanır. Türev maskesi iki farklı piksel yoğunluk değerinin farkı alınarak hesaplanır.

Tablo 2-4 Yön Vektörlerinin Hesabı için Kullanılan Türev Maskeleri

a) x düzleminde tek taraflı türev 	b) x düzleminde iki taraflı türev 
c) y düzleminde tek taraflı türev 	d) y düzleminde iki taraflı türev 

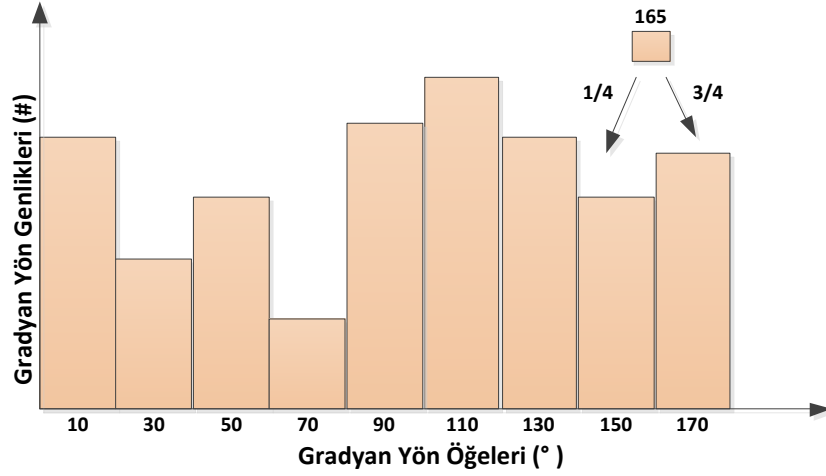
Tablo 2-4'te verilen türev maskeleri esas alınarak x ve y eksenlerinde ayrı ayrı türevler hesaplandıktan sonra yön vektörlerinin genliği ve yönü ise Denklem 2-1'de verildiği üzere hesaplanır.

$$\text{Genlik: } s = \sqrt{s_x^2 + s_y^2}; \text{Yön: } \theta = \arctan\left(\frac{s_y}{s_x}\right) \quad (2-1)$$

Verilen formüle göre x ve y eksenindeki piksel farklarının genliği ve açı değeri hesaplanır. Hesaplanan açı değeri yön vektörü histogramında girdi olarak kullanılır.

2.4.1.2.2 Yön Vektörü Histogramı Oluşturma

Her bir hücrede (9x9) yer alan pikseller için 0'dan 180 dereceyi ya da 0'dan 360 dereceyi kapsayan ve genellikle 20 derecelik aralıklardan oluşan yön vektörü histogramı oluşturulur. Yaya tespiti probleminde 0'dan 180 dereceyi kapsayan bir histogramın problem çözümü için yeterli olduğu tespit edildiğinden her biri 20 derecelik olmak üzere 9 histogram ögesi yeterli olmaktadır [24].



Şekil 2-1 Yön Vektörü Histogramı

Her bir yön ögesine ait genlik hesaplanırken (2-1'de verilen genlik formülü kullanılır. Her bir piksel için hesaplanan genlik değeri 9x9'lık hücre bloklarında yön histogramı hesaplamak için kullanılır. Örneğin, hesaplanan yön değeri 0 ile 20 ya da -20 ile 0 derece arasında ise yön vektörüne ait genlik değeri "10" ile ifade edilen histogram ögesine eklenir. Bu işlem hücre içinde hiç yön vektörü kalmayınca kadar devam eder.

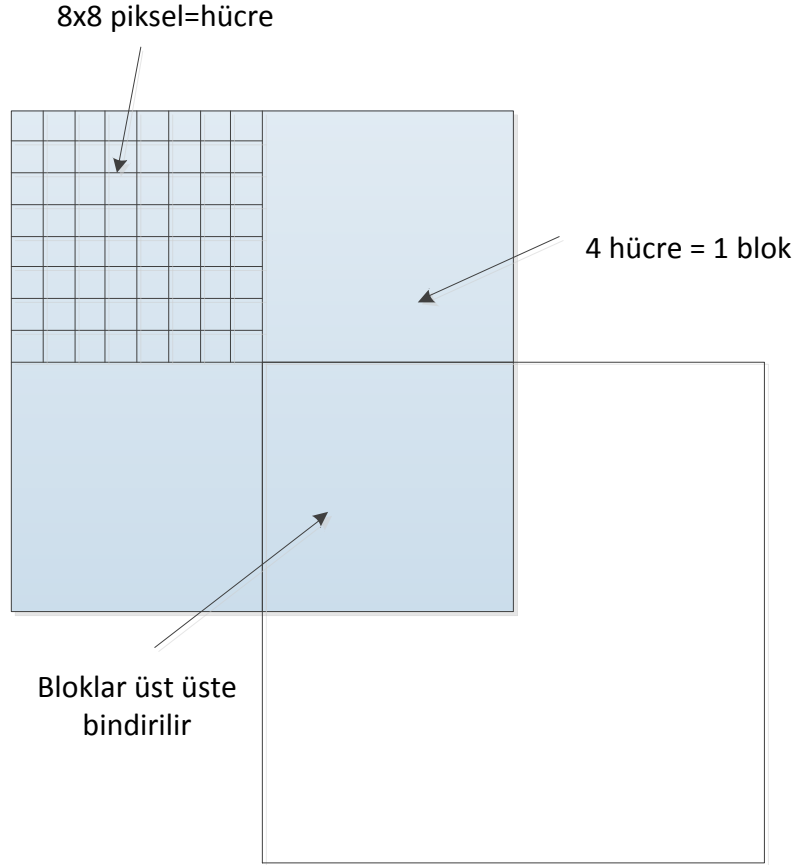
2.4.1.2.3 Tanımlayıcı Bloklar

Işık aydınlatması ve zıtlık etkisini en aza indirmek için yön vektörü genlikleri normalleştirilmelidir. Görüntünün farklı bölgelerindeki yön vektörlerinin normalleştirilmesi için ise hücreler birleştirilerek bloklar oluşturulur. Bloklar

oluřturulurken, bloklar genel olarak üst üste bindirilerek bloklar arasında kesiřen hücreler bırakılır. İki tip blok geometrisi vardır: dikdörtgen řekilli R-HOG blokları ve dairesel C-HOG blokları. R-HOG blokları genel olarak kare řekline sahip olup üç parametre ile ifade edilir: her bir bloktaki hücre adedi, her bir hücredeki piksel adedi ve hücre üzerinden hesaplanan histogramın öęe sayısı. Dalal ve Triggs'in yaya tespiti çalıřmasında kullanılan en uygun deęerler řunlardır: 3x3 hücreden oluřan blok, 6x6 pikselden oluřan hücre ve 9 öęeden oluřan yön vektörü histogramı [24].

R-HOG blokları SIFT tanımlayıcılara benzese de, tek bir boyutta hesaplanmaları ve yön düzeltmesi olmadan hesaplanmaları yönünden SIFT'ten ayrılır. SIFT tanımlayıcı ise görüntüdeki büyüklükten bağımsız temel noktalar yardımı ile yön düzeltmesi yapılarak hesaplanır.

C-HOG blokları iki tiptir: tek ve merkezi bir hücreye sahip olanlarla açısız olarak bölünmüş hücreler. Buna ek olarak C-HOG blokları dört parametre ile ifade edilebilir: açısız veya radyal öęelerin sayısı, merkezi öęenin yarıçapı ve ek açısız öęelerin genişleme faktörünün sayısı. Yapılan bir çalıřmada [24] iki yöntemin de yakın performanslar gösterdięi belirtilmektedir. 4 piksellik bir yarıçapa sahip bir hücre ve genişleme faktörü olarak 2 seçilmesi ile en iyi performans elde edilmektedir.



Şekil 2-2 HOG Tanımlayıcısı için Kullanılan Hücre ve Blok Yapısı

Şekil 2-2’de HOG tanımlayıcısı için kullanılan klasik bir hücre ve blok yapısı görülmektedir. 8x8 piksel alanı kaplayan alana hücre, hücrelerin 2x2 (hücre x hücre) ya da 3x3 gibi kare formunu oluşturduğu nihai yön vektörlerinin atandığı alana blok denir. Blokların üst üste bindirilmesinin sebebi ise bölgesel geçişlerdeki ışık, yoğunluk ve karşıtlık etkilerinin birlikte kullanılması tanımlayıcının farklı bölgeler arası geçişleri normalleştirme yaklaşımı ile ilintilidir.

2.4.1.2.4 Blok Normalleştirilmesi

Blok normalleştirilmesi; ışık yoğunluğu, gölgeleme ve arka plan gibi faktörleri en aza indirmek için her bir hücredeki piksel değerlerinin ortalaması kullanılarak belirli sayıda hücrelerden oluşan bir bloğa uygulanır. Normalleştirme sayesinde tanımlayıcı, aydınlatma ve gölgelemeden kaynaklı değişimlere karşı tolerans sağlar. Blok normalleştirilmesi için üç farklı yöntem kullanılır. Bir bloktaki tüm histogramları içeren normalleştirilmemiş değerleri içeren bir θ vektörü olmak üzere;

ϵ küçük bir sabit olmak kaydı ile 4 farklı normalleştirme faktörü Denklem 2-2'den Denklem 2-4'e kadar verilen 4 denklem ile ifade edilmektedir [25].

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

$$\text{L2 - hys: } v > 0.2 \quad v = 0.2; f = \frac{v}{\sqrt{\|v\|_2^2 + \epsilon^2}} \quad (2-3)$$

$$\text{L1 - norm: } f = \frac{v}{\sqrt{\|v\|_1 + \epsilon^2}} \quad (2-4)$$

$$\text{L1 - karekök: } f = \sqrt{\frac{v}{\|v\|_1 + \epsilon^2}} \quad (2-5)$$

HOG yöntemi kullanılarak yapılan bir çalışmaya göre L2-hys, L2-norm ve L1-karekök normalleştirme yöntemlerinin yakın sonuçlar verdiği gözlemlenmektedir [24]. Yalnız L1-norm yönteminin diğer yöntemlere göre daha kötü sonuç verdiği de ortaya konmaktadır. L1-norm yöntemi diğer yöntemlere göre daha kötü sonuç verse de sonuç olarak dört yöntemin de normalleştirilmemiş veriye göre çok daha iyi sonuç verdiği ortaya konmaktadır.

2.4.1.2.5 DVM (Destek Vektör Makinesi) Sınıflandırıcı

HOG tanımlayıcılar vasıtası ile nesne tespitinin son aşaması gözetimli öğrenmeye sahip olan bir sınıflandırıcıyı eğitmektir. DVM (Destek Vektör Makinesi) sınıflandırıcı bu bağlamda hem performans hem doğruluk anlamında seçilebilecek en iyi sınıflandırıcılardan biridir. DVM sınıflandırıcısı görüntü işlemede yoğun olarak kullanılan, hız ve doğruluk bağlamında iyi sonuçlar veren bir sınıflandırıcı çeşididir [18].

DVM, sınıflandırılacak boyutlar arasındaki ayrışmayı en iyi sağlayacak yüksek boyutlu düzlemi bulmayı sağlayan bir sınıflandırıcıdır. DVM, elde olan ve 2 farklı sınıfa ait olan bir veri setine ait veri noktalarını, 2 sınıftan birine düşecek şekilde olasılık hesabına dayanmayan bir şekilde ayrıştıracak bir modele sahiptir. Bu

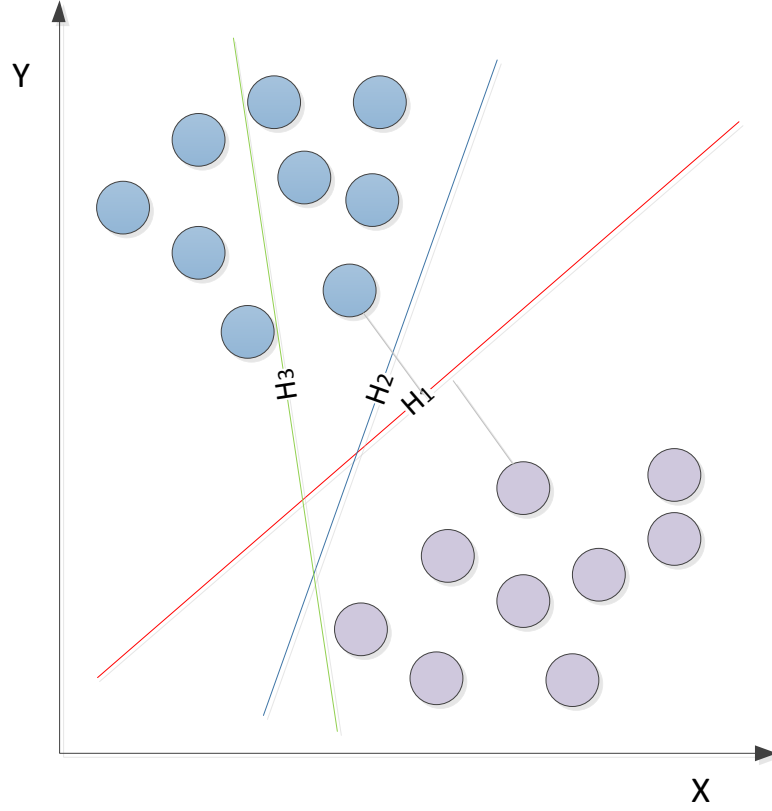
modele göre veri uzayında ayrıştırılan veri grupları arasındaki ayrıştırma düzlemi iki sınıfı birbirinden en uzak noktalara düşecek şekilde bir ayrıştırır. Yeni gelen veriler ise düzlemin hangi tarafına düşmekte ise, düştüğü taraftaki sınıfta yer alır. DVM eğitildikten sonra herhangi bir nesnenin, örneğin yaya, bir görüntü üzerinde var olup olmadığını kararını verebilir.

DVM modeli, bir uzayda yer alan veri noktalarını farklı kategorilere ayırırken ayrılan veri kümelerini bölen çok boyutlu düzlemin veri kümelerine en fazla ve en uygun uzaklıkta olmasını esas alır. Yani, DVM çok boyutlu bir düzlem ya da çok boyutlu düzlemler setini var olan veri seti üzerine kurarak sınıflandırma amacı ile bu düzlemleri kullanır. Bu işlem doğrusal olabildiği gibi özel yöntemlerle doğrusal olmayan sınıflandırma da yapılabilir.

Orijinal DVM algoritması Vladimir N. Vapnik ve Alexey Ya. Chervonenkis tarafından 1963'te geliştirilmiştir. 1992'de Bernhard E. Boser, Isabelle M. Guyon ve Vladimir N. Vapnik doğrusal olmayan sınıflandırıcıların oluşturulması için çekirdek yöntemini en yüksek aralıklı çok boyutlu düzlemler için önermiştir [26].

DVM şu alanlarda kullanılır [27]:

- Metni kategorik olarak ayırma ve sınıflandırılma
- Görüntülerin sınıflandırılması
- Tıpta protein sınıflandırılması
- El yazısı tanıma ve sınıflandırılması



Şekil 2-3 Bir Veri Setinin DVM ile 2 Farklı Kümeye Ayrılması

Şekil 2-3'te verilen 2 farklı sınıfa ait veri noktalarını, H_3 doğru parçası düzgün bir şekilde sınıflandırıp ayırtamazken; H_2 doğru parçası veri kümelerini küçük bir aralıkla ayırır. Bu durum DVM için istenmeyen bir durumdur. Çünkü ayırıştırmanın en iyi olması sınıflandırıcının uzun vadedeki performansını da doğrudan etkilemektedir. DVM için en iyi çözüm olan H_1 doğru parçası ise veri kümelerini en yüksek aralık ile ayırtmaktadır [27]. DVM'de tercih edilen ayırıştırma düzlemi/doğrusu ise bu durumda veri kümelerini en iyi ayırtıran H_1 doğru parçası olmaktadır. Çok boyutlu verileri/özellikleri sınıflandırmada da aynı yaklaşım uygulanmaktadır. Sonuç olarak, özellik setlerinin en iyi aralık ile ayırtıracak çok boyutlu düzlem, sınıflandırma için en iyi olan düzlem olmaktadır.

2.4.2 Haar Özellikleri Tabanlı Tanımlayıcı

2.4.2.1 Teori

Haar özellikleri, nesne tespitinde kullanılan sayısal görüntü özellikleridir. Adını Haar dalgacık dönüşümüne olan benzerliğinden almaktadır. Haar özellikleri ilk olarak gerçek zamanlı yüz tespit sistemi için kullanılmıştır [19].


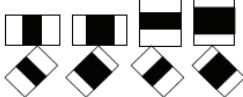
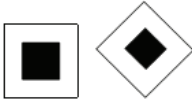
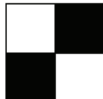
Haar özellikleri kullanarak nesne tespiti yaklaşımı Haar dalgacık dönüşümünden uyarlanarak Viola ve Jones tarafından geliştirilmiştir [28]. Temel olarak herhangi bir nesnenin alt parçacıklarının farklı renk yoğunluğuna ve dağılımına sahip olduğu bilgisi kullanılarak nesne farklı alt parçalara ayrılır. Her bir alt parçaya ait özellikler farklı özellik setleri ile ifade edilerek nesnenin tümü tanımlanır. Örneğin, insan yüzünde gözler yanak bölgesinden daha koyudur. Göz ve yanak bölgesinin her birine birbirine komşu iki dikdörtgen koyarak, dikdörtgenlerin altındaki bölgelerin piksellerin yoğunlukları farkı yüz tespitinde kullanılır. Yüz üzerinde buna benzer başka özellikler de kullanılarak insan yüzünü diğer nesnelere ayıracak özellik kümesi inşa edilir. Elde edilen özellik kümesi farklı eğitim setleri kullanılarak bir sınıflandırıcı marifeti ile nihai nesne tanımlayıcıyı elde etmek üzere eğitilir. Son olarak eğitilen nesne tanımlayıcı aynı tip nesnelere, örneğin insan yüzü ya da yaya, tespit etmek için kullanılır.

2.4.2.2 Algoritma

2.4.2.2.1 Dikdörtgen Temelli Haar Özellikleri

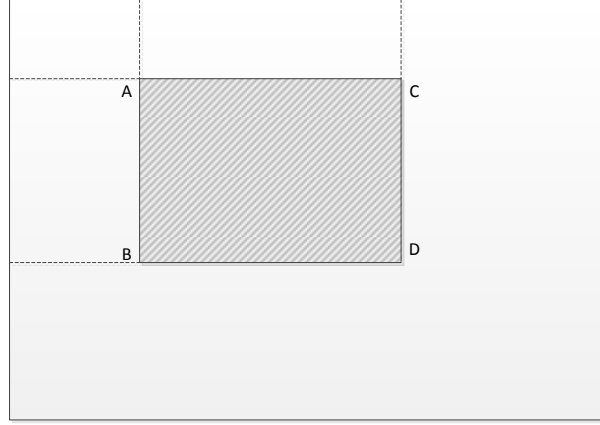
Dikdörtgen temelli en basit Haar özelliği iki farklı dikdörtgenin altındaki piksel yoğunlukları toplamının farkı olarak ifade edilebilir. Bu özelliğe 2-dikdörtgen özelliği denir. 2-dikdörtgen özelliği yanında 3 ya da 4-dikdörtgen özelliği de kullanılır. Tablo 2-5'te farklı özellik çekirdekleri verilmektedir. Örneğin Tablo 2-5(a)'da verilen 2-dikdörtgen özelliği karanlık ve aydınlık bölge arasındaki sınırın belirlenmesinde ve tespit edilmesinde kullanılır.

Tablo 2-5 Dikdörtgen Alana Sahip Haar Özellik Çekirdekleri [29]

 <p>(a) Kenar özelliği</p>	 <p>(b) Çizgi özelliği</p>
 <p>(c) Merkez-kenar özelliği</p>	 <p>(d) Çapraz çizgi özelliği</p>

2.4.2.2.2 Haar Özelliklerinin Hızlı Hesaplanması (Tümleşik Görüntü)

Toplam alan tablosu, Haar özelliklerini hesaplarken hesaplamayı hızlandırmak için geliştirilmiştir. Toplam alan tablosu, tümleşik görüntü olarak da adlandırılır. Tümleşik görüntü, iki boyutlu bir arama tablosudur ve orijinal görüntü boyutunda bir matris olarak ifade edilir [30]. Tümleşik görüntü matrisindeki her bir element sol üstte yer alan piksellerin yoğunluk değerlerinin toplamıdır. Bu matris sayesinde, görüntüdeki herhangi bir dikdörtgen alanda yer alan piksellerin yoğunluk değeri elde edilebilir.



Şekil 2-4 Tümleşik Görüntü Üzerindeki Taralı Alan

Şekil 2-4'te verilen büyük dikdörtgenin içindeki taralı alana sahip dikdörtgenin alanı "T" tümleşik görüntüye ait matrisi belirtmek üzere aşağıdaki denklemde verilmiştir.

$$\text{Taralı Alan} = T(D) + T(A) - T(B) - T(C) \quad (2-6)$$

Farklı Haar özellikleri hesaplanırken dörtten fazla tümleşik matris indeksine ihtiyaç duyulabilir. Örneğin Tablo 2-5(a)'da yer alan 2-dikdörtgen özelliğinin hesaplanması için 4 adet indeks değeri gerekirken, Tablo 2-5(b)'de yer alan 3-dikdörtgen özelliğinin hesaplanması için ise 6 adet indeks değerine ihtiyaç duyulur.

2.4.2.2.3 Kuvvetlendirilmiş Basamaklı Sınıflandırıcı

Basamaklı öğrenme, farklı sınıflandırıcıların artarda bağlanması ile oluşan bir sınıflandırıcı türüdür. Sıralı sınıflandırıcılardan önce olanın çıkışı, takip eden sınıflandırıcı için giriş bilgisi olarak kullanılır. Sıralı sınıflandırıcılardan herhangi

birinden geçemeyen bir özellik seti elenir. Sıralı sınıflandırıcıda bir özellik setinin pozitif olarak nitelendirilebilmesi için o özellik setinin tüm basamaklardaki kıstaslardan geçmesi gerekmektedir. Basamaklı sınıflandırıcı Denklem 2-7’de verilen formül ile ifade edilir.

$$h(x) = \text{sign} \left(\sum_{j=1}^M \alpha_j h_j(x) \right); h_j(x) = \begin{cases} -s_j & \text{eğer } f_j < \theta_j \\ s_j & \text{aksi halde} \end{cases} \quad (2-7)$$

(2-7’te verilen θ_j ifadesi her bir özellik (f_j) için eşik değerini belirtmekte iken α_j ifadesi her bir özellik sınıflandırıcı için kullanılan katsayıları/ağırlığı ifade eder.

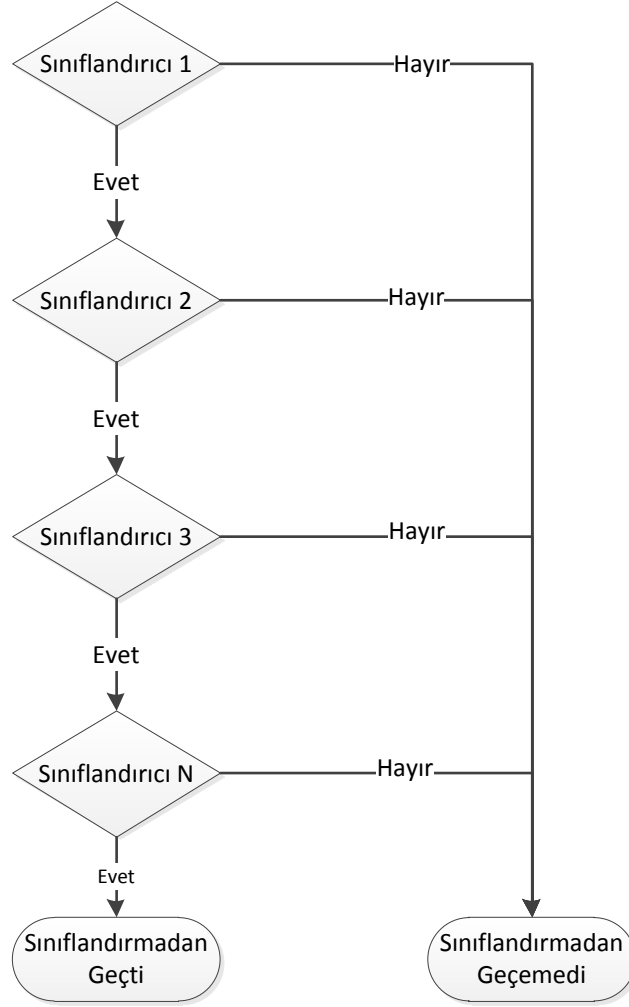
N adet pozitif ve negatif eğitim için seçilen görüntü olmak üzere, basamaklı sınıflandırıcı algoritmasının akışı şu şekildedir [31].

- 1) Her görüntüyü pozitif ve negatif görüntü olarak etiketle.
- 2) Ağırlık katsayısını $\alpha_j = 1/N$ olacak şekilde ayarla.
- 3) Her bir eğitim safhasında var olan özellik f_j $j=1 \dots M$ için,
 - a) Ağırlık katsayılarını toplam 1 olacak şekilde tekrar normalleştir. (Ağırlık katsayılarını kendi toplamlarına böl.)
 - b) Özelliği görüntü seti içerisindeki tüm görüntülere uygula. En iyi θ_j ve s_j değerlerini sınıflandırma hatasını en aza indirecek şekilde bul.
 - c) h_j için atanan α_j ağırlık değerini hata oranına ters orantılı olacak şekilde ata. Yani hata oranı fazla çıkan özellik için daha düşük katsayı atayarak hata oranı düşük olan özelliği ön plana çıkar.
 - d) Sınıflandırma hatalarına göre takip eden safha için ağırlık katsayısını güncelle.
 - e) Son olarak hesaplanan katsayıları Denklem 2-7’de verilen sınıflandırıcıya gir.

Basamaklı sınıflandırıcıdaki “kuvvetlendirilmiş” sıfatı ise özellik kümesi içindeki en iyi ve en anlamlı özellik alt kümesinin seçilmesi anlamında kullanılır.

İlk basamaklı sınıflandırıcı Viola ve Jones tarafından yüz tespiti için kullanılmıştır [32]. Bu tip bir sınıflandırıcının tasarlanmasının gerekçesi düşük işlemci gücüne

sahip sistemlerde, örneğin gömülü işlemciye ve video kameraya sahip bir sistemde yüz tespiti problemini etkili bir hız ve doğrulukta çözmektir.

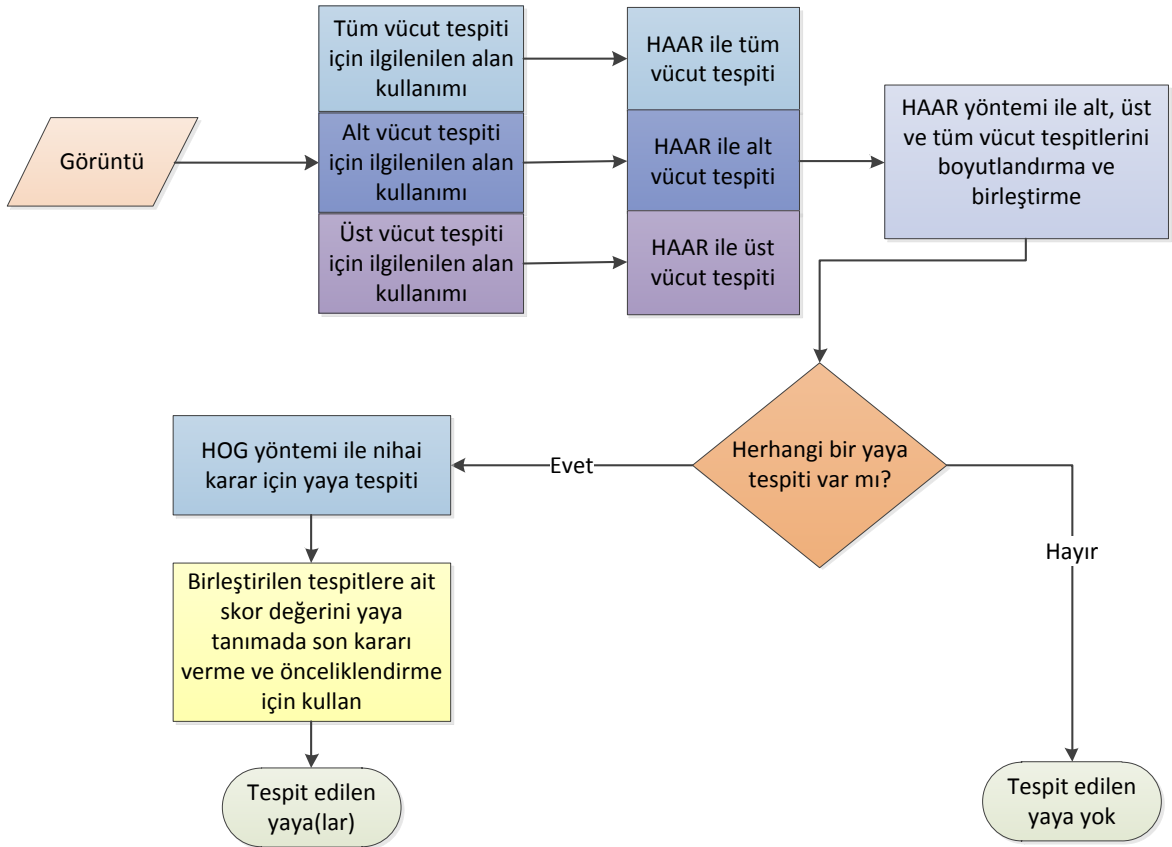


Şekil 2-5 Kuvvetlendirilmiş Basamaklı Sınıflandırıcı Sistem Akış Şeması

Şekil 2-5'te klasik bir basamaklı sınıflandırıcı yapısı verilmektedir. Her basamakta özellik kümesindeki bir özellik teste tabii tutulur. Özellik seti bir testi geçerse bir takip eden aşamaya geçer. Testi geçemezse takip eden aşamalara gelmeden test altındaki görüntü bölgesi için negatif puan verilerek özellik seti sınıflandırmada elenmiş olur.

3. GERÇEK ZAMANLI YAYA TESPİT ALGORİTMASI

Çalışmada Bölüm 2.3'te yer alan gerçek zamanlı yaya tespit kistaslarını göz önünde tutarak var olan yaya tespit yöntemlerini de kullanarak yeni bir yaklaşım ortaya konulmaktadır. Bu yaklaşımda ön planda tutulmaktadır: tespit hızı ve tespit doğruluğu. Tespit hızını artırmak için işleme hızı yüksek bir yöntem ön işlemede kullanılmıştır. Ön işleme ile görüntüdeki yaya olması muhtemel alanların belirli bir hatalı pozitif kestirim ile tespit edilmesi amaçlanmaktadır. Ön işlemeden sonra ise ön işleme ile elde edilen tespit sonuçları Bölüm 3.5'te belirtildiği üzere birleştirilmektedir. Tespit doğruluğunu ve tespit güvenilirliğini artırmak üzere tespit oranı Haar yöntemine göre daha yüksek olan HOG yöntemi ile son tespit gerçekleştirilmektedir. Sistemin genel akışı Şekil 3-1'de verilmektedir.



Şekil 3-1 Geliştirilen Yaya Tespit Algoritmasının Genel Akış Şeması

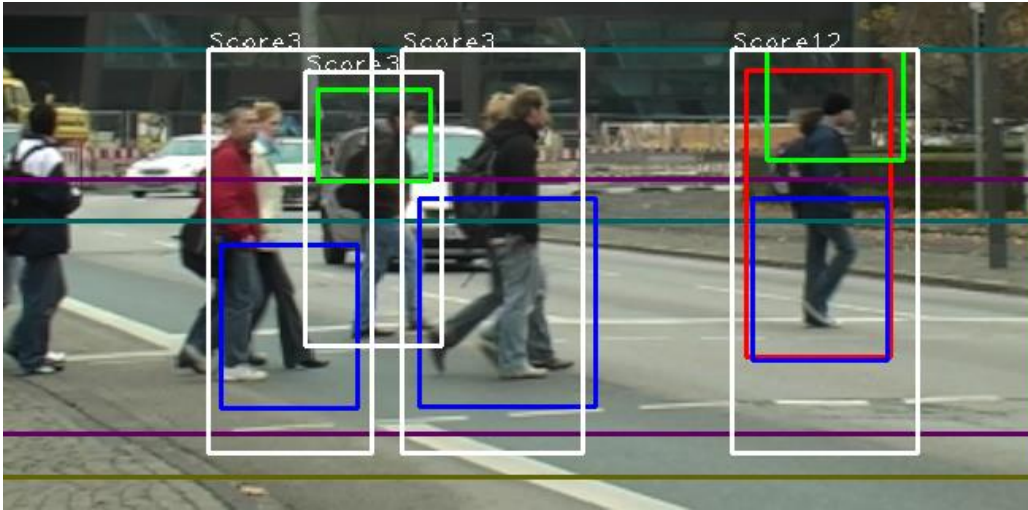
Şekil 3-1'de verilen algoritmanın genel akışının ayrıntıları takip eden alt bölümlerde verilmektedir. Akış içerisinde verilmeyen sınıflandırıcı eğitimi, ilgilenilen

alan seçimi gibi alt bölümler algoritmanın genel akışı içerisinde değil, algoritma haricinde var olan hesaplama ve destek bölümleridir. Yani algoritmada kullanılan parametre ve sınıflandırıcı yapıları algoritma ana sistem akışı içerisinde yer almayıp algoritma çalıştırılmadan hazır edilen hesaplama ve ön bilgi kısımlarıdır.

3.1 Haar Yöntemi ile Parçasal ve Tüm Vücut Tespiti

Yaya tespit sistemlerinin birçoğu tüm vücudu tanıyacak tanımlayıcılar ve sınıflandırıcılar geliştirmiş iken gerçek hayatta yayalar farklı nesnelere arkasında veya farklı arka planların önünde yer alabildiklerinden dolayı tüm vücut tanımlayıcılar iyi sonuçlar vermeyebilir. Bu sebeple tüm vücut tespiti ile birlikte tüm vücudun tanınmasının mümkün olmadığı durumlarda yaya tespitini mümkün kılacak şekilde parçasal yani alt ve üst vücut kısımlarını ayrı ayrı tanıyacak tanımlayıcılar geliştirme ihtiyacı ortaya çıkmaktadır. Bu kapsamda vücut üst bölgesini yani gövdeyi ve başı tanımlayan bir tanımlayıcı ile birlikte vücut alt bölgesini yani bacakları tanımlayan bir tanımlayıcı geliştirilmiştir. Bu tanımlayıcılar Haar özellik çekirdekleri kullanılarak belirlenmiştir.

Parçasal tespitinin en büyük avantajı gerçek ortam şartlarında ortaya çıkmaktadır. Yayanın alt ve üst kısmı arka plandan ya da bir engelden dolayı tespit edilemezse, sadece alt veya üst vücut kısmını tanıyarak yaya tespiti gerçekleştirmek mümkündür.



Şekil 3-2 Alt, Üst ve Tüm Vücut Kısımlarının Ayrı Tespiti (Görüntü kaynağı: [33])

Şekil 3-2'de verilen görüntüde soldan sağa 4 farklı tespit verilmektedir. Tespitler beyaz dikdörtgen içine alınmış olup yeşil, mavi ve kırmızı dikdörtgenler ise sırayla

üst, alt ve tüm vücut tespitlerini içermektedir. Verilen görüntüde örneğin soldaki ilk tespitite sadece alt vücut (bacak) tespiti varken tespit tüm vücudu kapsayacak şekilde genişletilmiştir. Bu sayede tam vücut tespiti yokken yaya tespiti mümkün hale gelmiştir. Sağdaki son tespitite ise alt, üst ve tüm vücut tespitleri birlikte gerçekleşmiş olup beyaz dikdörtgenle belirtilen son tespit kararı daha yüksek bir olasılıkla verilmiştir. Birleştirme ve son karar verme işleminin detayları Bölüm 3.5'te detaylı olarak verilecektir.

Bölüm 3.1.1 ve Bölüm 3.1.2'de parçasal vücut tespiti kapsamındaki alt ve üst vücut tespiti için izlenen yöntemi detaylı olarak verilmektedir. Bölüm 3.1.3'te ise Haar yöntemi ile gerçekleştirilen tüm vücut tespitine yer verilmektedir.

3.1.1 Alt Vücut Tespiti

Alt vücut tespiti yayanın üst kısmının arka plandan, ışıktan ve yoğunluktan dolayı tespit edilemediği durumlarda kullanılmaktadır. Birçok çalışmada vücudun alt kısmı olan bacak kısmı yaya tespitinde ayrı bir tespit unsuru olarak kullanılmaz. Çünkü bacaklar tespit edilmek için çok fazla özellik seti çıkarılmasına izin vermez. Literatürde yapılan çalışmalarda [29] [34] [35] genelde üst vücut ve tüm vücut bilgisi birlikte ya da ayrı ayrı kullanarak yaya tespitine ağırlık verilmektedir. Alt vücut tespiti ile yaya tespit yaklaşımı olan çalışmalarda ise hareket vektörü üzerinden hesaplama yapılır. Bacağa ait hareket vektörlerini kullanarak yürüyen yayaları tespit eden bir çalışmada video üzerindeki peş peşe gelen görüntüler üzerinden yaya tespiti yapılmaktadır [36]. Tespit yapılırken yürüyüşe ait yola ait özellikler HOG yöntemi ile çıkarılmakta ve sınıflandırılmaktadır. Tek bir görüntü üzerindeki alt vücut bilgisinin (bacaklar) kullanımı ve tespiti literatürde ilk olma özelliği taşımaktadır. Alt, üst ve tüm vücut bilgisini tespit için ayrı ayrı kullandıktan sonra tespit sonuçlarının birleştirilmesi işlemi alt vücut bilgisini kullanmaktan kaynaklı hatalı tespitlerin önüne geçmeyi amaçlamaktadır. Tespit sonuçlarının birleştirilmesi yönteminin detayları Bölüm 3.5'te verilmektedir.

Çalışmamızda Bölüm 3.4'te detayları verilen ilgilenilen alan seçimi yaklaşımı ile alt vücudun olduğu bölge sisteme önceden tanımlanmıştır. Böylece, alt vücut bölgesinin aranacağı alanın daraltılarak hatalı tespitlerin önüne geçilmesi amaçlanmıştır. Aynı zamanda, alt vücut tespiti için kullanılan alan görüntü üzerinde daha az bir alanı kapsadığı için tespit hızında da artış elde edilmesi planlanmıştır.

3.1.2 Üst Vücut Tespiti

Üst vücut yayayı tanımlayan en önemli parçadır. Kafanın, omuzların ve konumların konumları tespit için mükemmele yakın özellik setleri çıkarılmasına imkân verir. Bu yüzden literatürde yapılan çalışmalarda genelde üst vücut ve/veya tüm vücut bilgisini kullanarak yaya tespitine ağırlık verilmektedir.

Üst vücut bölgesini parçalara ayırarak; örneğin kolları, başı ve gövdeyi ayrı ayrı tespit edip bu parçaların birbirlerine göre konumlarını kullanarak, birleştirilmesi ve tespit edilmesi yaklaşımını uygulayanlar olduğu gibi [37] çalışmamızda olduğu gibi üst vücut bölgesi için tek bir tanımlayıcı kullanan [29] [34] [35] çalışmalar da vardır. Bizim yaklaşımımız üst vücut için Haar özellik tabanlı tek bir tanımlayıcı kullanmak olmuştur. Çünkü kolları, başı ve gövdeyi ayrı ayrı tespit edip birleştirme işlemi hem tespit sürecini karmaşıklarıştırmakta, hem de beklenen hız ve performans kıstaslarına ulaşmada önemli bir fayda sağlamamaktadır.

Çalışmamızda, Bölüm 3.4'te detayları verilen ilgilenilen alan seçimi yaklaşımı ile üst vücudun olduğu bölgeyi sisteme önceden tanımlanmıştır. Böylece, üst vücut bölgesinin aranacağı alanı daraltarak hatalı tespitlerin önüne geçilmesi amaçlanmıştır. Aynı zamanda üst vücut için kullanılan alan görüntü üzerinde daha az bir alanı kapsadığı için tespit hızında da artış elde edilmesi planlanmıştır.

3.1.3 Tüm Vücut Tespiti

Tüm vücut tespiti birçok durumda mümkün olmamaktadır. Alt veya üst vücudun değişik arka plan, engel veya gizlenme durumu tüm vücut tespitini zaman zaman imkânsız hale getirmektedir. Ancak tüm vücut tespiti yayanın varlığına ait kesine yakın bir bilgi verir. Bu yüzden literatürde yapılan çalışmalarda [29] [34] [35] genelde tüm vücut bilgisini kullanarak yaya tespitine ağırlık verilmektedir.

Çalışmamızda, tüm vücut tespitini güç kılan faktörleri de hesaba katarak alt ve üst vücut tespitini de paralelde yürüterek Bölüm 3.5'te anlatılan yaklaşım ile 3 farklı tespit sonucu birleştirilmiştir. Bu sayede hatalı pozitif tespitlerin miktarı azaltılmıştır. 3 farklı tespit her biri diğerinin tespit edemediği muhtemel yayayı tespit ederek toplam yaya tespit oranının artmasına destek sağlamaktadır. Örneğin, arka planın karmaşıklığından dolayı tüm vücudu tespit edilememiş olan bir yayanın sadece alt vücut bölgesi tespiti ile yayanın tespiti sağlanmaktadır.

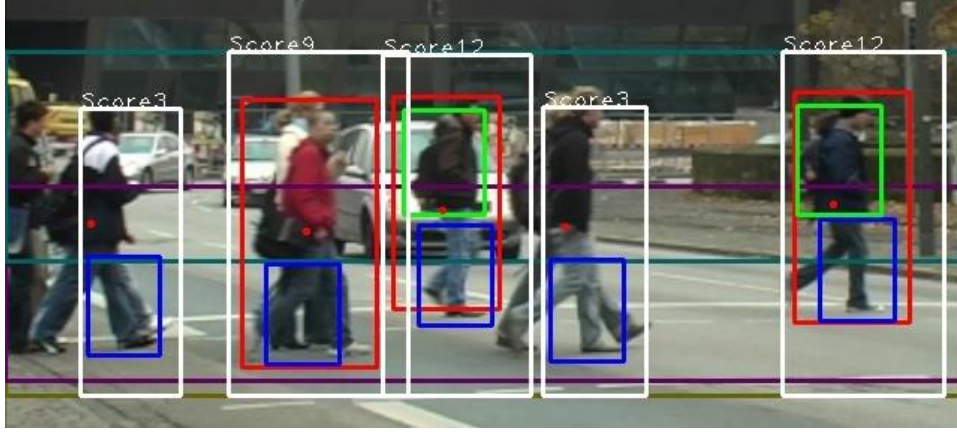
Böylece parçasal bilgi kullanarak da yayanın varlığı hakkında uyarı verilmesi sağlanmaktadır

3.2 HOG Yöntemi ile Tüm Vücut Tespiti

HOG yöntemi ile vücut tespiti literatürde geniş kabul görmüş ve başarısı kanıtlanmış bir yöntemdir. Bu yönetime ait teori Bölüm 2.4.1'de anlatılmaktadır. O yüzden bu bölümde sadece HOG yönteminin kendi uygulamamızdaki detayları anlatılacaktır.

Tez çalışmasında HOG yöntemine ait OPENCV kütüphanesine ait varsayılan sınıflandırıcı (Varsayılan Yaya Tespit Edicisi) yeterli tespit (%80 tespit) oranını karşıladığı için yeni bir sınıflandırıcı tasarımına ihtiyaç duyulmamıştır. Tez çalışmasında nihai olarak elde edilen yaya tespit oranı HOG yönteminden elde edilen skor olmasa da %90 civarında olmaktadır. HOG yöntemi nihai güvenlik seviyesi olarak toplam yaya tespit oranını etkilememektedir. Şekil 3-1'de verilen sistem akış diyagramından da görüleceği üzere HOG yöntemi ilk 3 Haar tespit edicinden herhangi bir yaya gelmezse çalışmamaktadır. Bu durumda sistemin yaya tespit oranı ilk 3 yönetime bağlıdır. Sonuç olarak çalışmamızda HOG yöntemi ile yaya tespiti için OPENCV yazılım kütüphanesine ait varsayılan yaya tanımlayıcısı kullanılmaktadır.

Haar yöntemi ile elde edilen 3 farklı tip yaya tespiti Bölüm 3.5'te detayları verilen farklı tespitlerin birleştirilmesi işleminden geçtikten sonra HOG yöntemi çalıştırılır. Bu işlemin nasıl gerçekleştiğini anlamak için aşağıda verilen şekli incelemek yeterli olacaktır. Bu şekilde 5 farklı yayaya ait parçasal bilgiler ayrı ayrı elde edilip birleştirilmekte ve nihai tespit için hazır hale getirilmektedir.



Şekil 3-3 HOG Yönteminde Kullanılan Birleştirilmiş Tespitler (Görüntü kaynağı: [33])

Şekil 3-3'te verilen görüntüde soldan sağa 5 farklı tespit verilmektedir. HOG yöntemi onayına girecek tespitler beyaz dikdörtgen içine alınmış olup yeşil, mavi ve kırmızı dikdörtgenler ise sırayla üst, alt ve tüm vücut tespitlerini içermektedir. HOG yöntemine alt veya üst vücut bölgesi tespit edilen yayalar girdi olarak verildiği gibi tüm, alt ve üst vücut bölgesinin üçü birden tespit edilen en sağdaki yaya tespiti de girdi olarak sağlanmaktadır. Farklı tespitler arasındaki fark, nihai tespitten sonra her bir tespit için elde edilen skorlar olacaktır. Tespitlere ait skorların hesaplanması ile ilgili yöntem Bölüm 3.6'da detaylı olarak verilmektedir.

Sonuç olarak, HOG yöntemi hem güvenilirlik hem de yaygın kabul görmesi sebebi ile nihai tespit onaylayıcı olarak kullanılmıştır. Ancak çalışmamızda bununla yetinilmeyip güvenilirlik ve öncelik atama mantığı kurmak için Bölüm 3.6'da detayları verilen skor kıstası geliştirilmiştir.

3.3 Haar Yöntemi için Sınıflandırıcı Seçimi

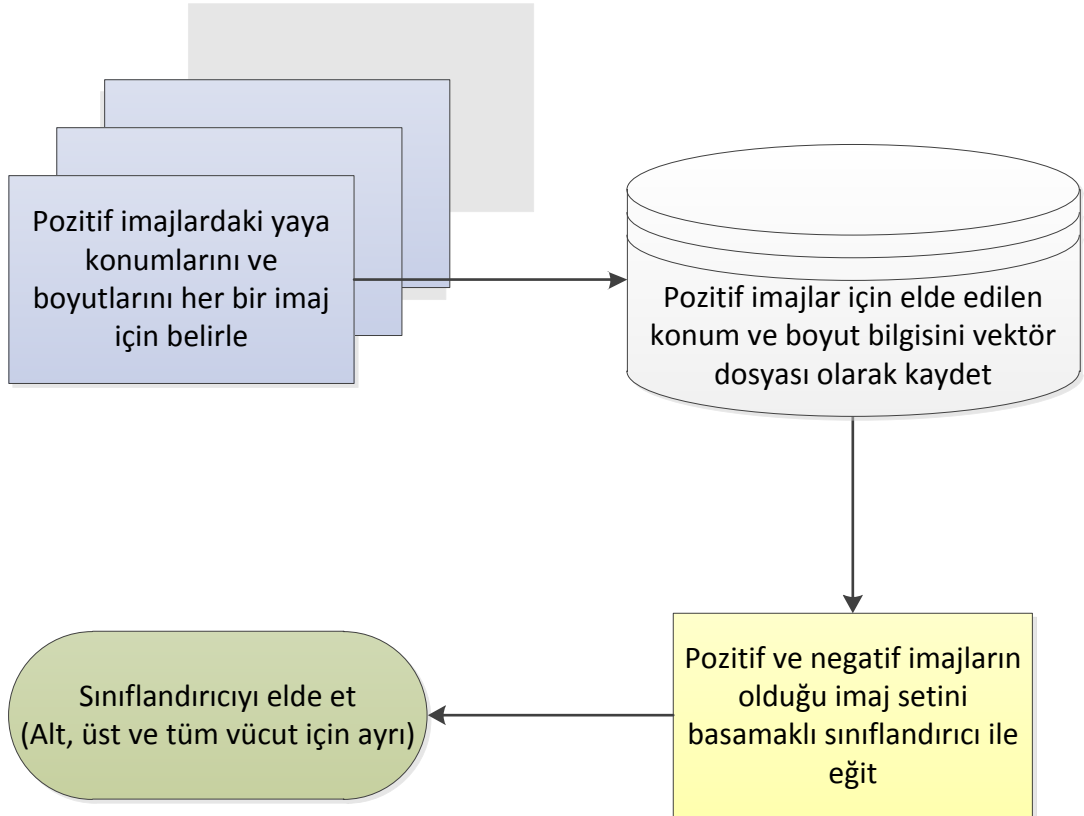
Tez çalışmasında 3 farklı vücut bölgesi için 3 farklı sınıflandırıcı Haar yöntemi için eğitilmiştir. HOG yöntemi için ise hazır tespit edici kullanıldığından dolayı sınıflandırıcı ile eğitimi yapılmamıştır. Bu bölümde Haar yöntemine ait 3 farklı yaya tespitine ait tespit edicinin nasıl eğitildiği anlatılmaktadır.

Haar yöntemine ait sınıflandırıcı eğitiminde OPENCV kütüphanesine ait farklı sınıflandırıcı eğitimi uygulamaları kullanılmıştır. Kullanılan uygulamaların isimleri sınıflandırıcı eğitiminin safhalarında verilmektedir.

Sınıflandırıcı eğitimi yaya tespitindeki en kritik safhalardan biridir. Çünkü tespit oranı ve tespit hızı yüksek bir sınıflandırıcı Bölüm 2.3.2’de belirtildiği üzere yaya tespit probleminde toplam performansı etkileyen en önemli unsurlardan birisidir. Çalışmamızda parçasal vücut tespitine ait alt, üst ve tüm vücut bölümleri için üç ayrı sınıflandırıcı eğitilmiştir. Bunun için her bir vücut bölümü için ayrı pozitif görüntü setleri kullanılmıştır. Örneğin, alt vücut sınıflandırıcısı için pozitif görüntülerdeki alt vücut bölümlerinin konum ve boyut bilgileri kullanılmıştır Diğer vücut bölümleri için de aynı yaklaşım kullanılmıştır.

Sınıflandırıcı eğitimi temel olarak üç aşamadan oluşur:

- Pozitif görüntülerin içerisinde tespit edilmek istenen nesneyi (örneğin yaya, insan yüzü vs.) her bir pozitif görüntü için işaretlemek,
- Pozitif görüntülerden elde edilen konum ve boyut bilgisini vektör dosyası olarak kaydetmek,
- Negatif ve pozitif görüntü setini birlikte kullanarak sınıflandırıcıyı eğitmektir.



Şekil 3-4 Sınıflandırıcı Eğitimi Safhaları

Şekil 3-4'te sınıflandırıcı eğitim safhaları verilmektedir. Bu safhalar alt, üst ve tüm vücut için ayrı ayrı tekrar edilerek her biri için ayrı sınıflandırıcılar elde edilir.

Pozitif ve negatif görüntü seti için INRIA'nın açık kullanıma sunduğu veri seti kullanılmıştır. Veri seti, farklı çözünürlüklere sahip farklı ışık ve renk çeşitliliğine sahip negatif ve pozitif görüntüleri içeren kullanışlı bir veri setidir [38]. Veri setindeki negatif görüntüler herhangi bir işleme tabii tutulmazken pozitif görüntülerdeki yaya pozisyonları her bir görüntü için tek tek belirlenmelidir.

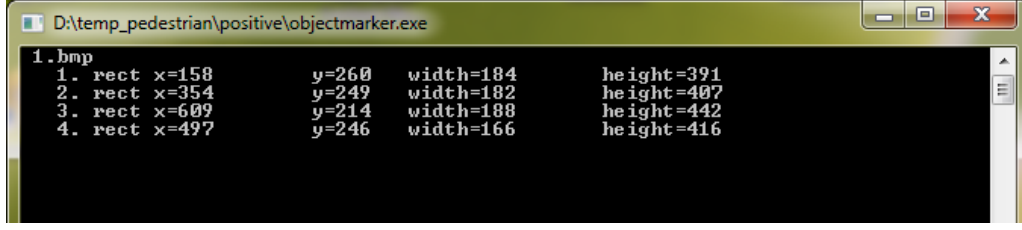
Çalışmamızda sınıflandırma için kullandığımız araçlar OPENCV kütüphanesine ait olup açık kaynak kodlu olarak indirilebilir [39] ve üzerine çalışılabilir araçlardır. Çalışmamızda sınıflandırma için kullandığımız araçlar (haartraining) pozitif görüntülerde yer alan yayaları x, y koordinat düzleminde ve en, boy şeklinde bir liste halinde işlediği için şöyle bir formatlamaya gidilmiştir:

Dosya yolu/dosya adı #yaya sayısı x0 y0 en0 boy0 x1 y1 en1 boy1....



Şekil 3-5 Sınıflandırıcı için Kullanılan Örnek Bir Yaya Görüntüsü (Görüntü kaynağı: [38])

Örneğin yukarıda Şekil 3-5'te verilen görüntüdeki yayaların boyutları ve konumları "objectmarker" programı kullanılarak elde edilmiştir. Seçilmek istenen yaya fare imleci yardımı seçilmekte ve yayaya ait konum ve boyut bilgisi bir sonraki aşama olan eğitim safhası için dosyaya kaydedilmektedir.



```
D:\temp_pedestrian\positive\objectmarker.exe
1. bmp
1. rect x=158      y=260      width=184      height=391
2. rect x=354      y=249      width=182      height=407
3. rect x=609      y=214      width=188      height=442
4. rect x=497      y=246      width=166      height=416
```

Şekil 3-6 Örnek Görüntüdeki Yaya Boyutları ve Konumları

Şekil 3-6'da pozitif görüntülerdeki yaya konumlarının belirlenmesinde kullanılan programın kullanıcı ara yüzü verilmektedir. Şekil 3-5'te verilen görüntüden seçilen dört adet yayanın konum ve boyut bilgisi verilmektedir. Bu bilgi sınıflandırıcı eğitiminde kullanılacak olan pozitif görüntüler üzerinden hesaplanacak özellik setlerinin çıkarılacağı alanları belirlemek için kullanılır. Yani, özellik setleri sadece görüntüde yayanın bulunduğu işaretlenmiş alanlar için çıkarılır.

Çalışmada kullanılan basamaklı sınıflandırıcıda 15 adet basamak kullanılmıştır. Basamak sayısı arttıkça tespit hatası düşerken tespit hızı düşmektedir [40]. Bu yüzden tespit oranını mümkün olduğunca yüksekte tutup hızdan da kaybetmeden mümkün olan en iyi basamak sayısı denemeler sonucunda 15 olarak belirlenmiştir.

Pozitif görüntü listesinin elde edilmesinden sonra yine OPENCV araçlarından olan `createsamples` uygulaması kullanılarak görüntü listesi vektör biçimine dönüştürülür. Son olarak hazırlanan negatif görüntü seti, pozitif görüntülere ait vektör dosyası ve eğitime ve sınıflandırıcıya ait diğer parametreler de ayarlanarak işlem başlatılır.

Haar tüm, alt ve üst vücut tespiti için kullanılan parametre seti tablosu aşağıda verilmektedir.

Tablo 3-1 Sınıflandırıcı Eğitiminde Kullanılan Parametre Setleri

	Tüm Vücut Sınıflandırıcısı	Alt Vücut Sınıflandırıcısı	Üst Vücut Sınıflandırıcısı
Pozitif Görüntü Sayısı	365	278	315
Negatif Görüntü Sayısı	1218	1218	1218

Sınıflandırıcı Basamak Sayısı	15	15	15
Tanımlayıcı Boyutları	En: 14; Boy: 28	En: 14; Boy: 28	En: 14; Boy: 28

3.4 İlgilenilen Alan seçimi

Gerçek zamanlı yaya tespit kıstasında önemli unsurlardan birisinin ön bilgi kullanmak olduğundan Bölüm 2.3.3'te bahsedilmişti. Bu kapsamda, parçasal vücut tespitinde birbiri ile doğrudan ilintili olmayan vücut parçalarını bir diğerinin bulunma ihtimali olan alanda aramak gereksiz ve anlamsız bir uğraş olmaktadır. Örneğin, Şekil 3-7'de verilen görüntüde alt vücut bölgesinin olduğu bölgede üst vücut tespiti yapmak olanak dışıdır. Bu yüzden üst vücut tespiti için kullanılan tanımlayıcının alt vücudun bulunma ihtimali yüksek olan alanlarda çalıştırılması performans kaybına neden olduğu gibi yanlış tespitlere de neden olabilir.

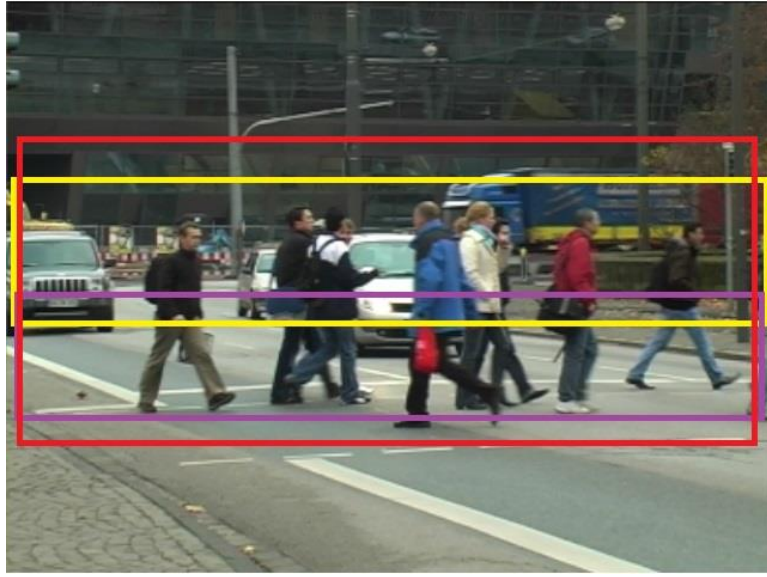
Literatürdeki birçok çalışmada ilgilenilen alan seçimi yaklaşımının eksik olduğu gözlemlenmiştir. Literatürde yapılan bazı çalışmalarda ise üst ve tüm vücut için ilgilenilen alan seçimi olsa da seçilen alan üst ve tüm vücut için aynı olmaktadır. Bu bağlamda, çalışmamızdaki parçasal vücut tespiti yaklaşımı ve alt, üst ve tüm vücut bölgeleri için ayrı ayrı ilgilenilen alan seçimi de yaya tespit problemi için yeni bir yaklaşımdır.

Görüntü üzerinde alt, üst ve tüm vücut parçaları tespit edilirken tespit oranı ve hız artışı sağlamak için "ilgilenilen alan" belirlenmiştir. Bu sayede örneğin alt vücut bölgesi tespiti yapılırken daha dar bir alanda ve diğer vücut bölgelerinin bulunmadığı bir alanda tespit yapıldığı için hız ve tespit oranında artış sağlanmaktadır.

Çalışmada iki tip ilgilenilen alan seçimi yaklaşımı denenmiştir. Birinci yöntemde test için kullandığımız görüntülerdeki yayaların dikey konumları bilindiği için sabit ve yatay olarak birbirinden ayrılan bölgeler seçilmektedir. İkinci ve otomatik yöntemde ise görüntü üzerinde alt, üst ve tüm yaya tespitleri algoritma koşturuldukça elde edilen verilerden yola çıkarak istatistiksel olarak farklı vücut bölümlerinin nerede olduğuna dair bilgi elde edilmektedir. İki yaklaşıma ait detaylar sonraki iki alt bölümde verilmektedir.

3.4.1 Ön Bilgi Kullanılarak Yapılan Sabit İlgilenilen Alan Seçimi

İlgilenilen alan seçiminin temeli kullanılan video yakalama cihazının konumlandırılmasına göre yayaların görüntünün hangi bölgesinde ve hangi boyutlarda olunacağı bilinmesine dayanmaktadır. Bu yaklaşımda ayrıca video kameraya sadece belirli uzaklıktan daha yakın yayaların tespit edilebilir olduğu varsayımı vardır. Çalışmada kullandığımız veri setinde [33] yaya geçidinden geçen yayalar kullanılmaktadır. Bu senaryoda alt, üst ve tüm vücuda ait ilgilenilen alanlar önceden bilinmekte ve net olarak birbirinden ayrıştırılmaktadır. Aşağıda verilen görüntüde örnek sabit alan seçimi verilmektedir.

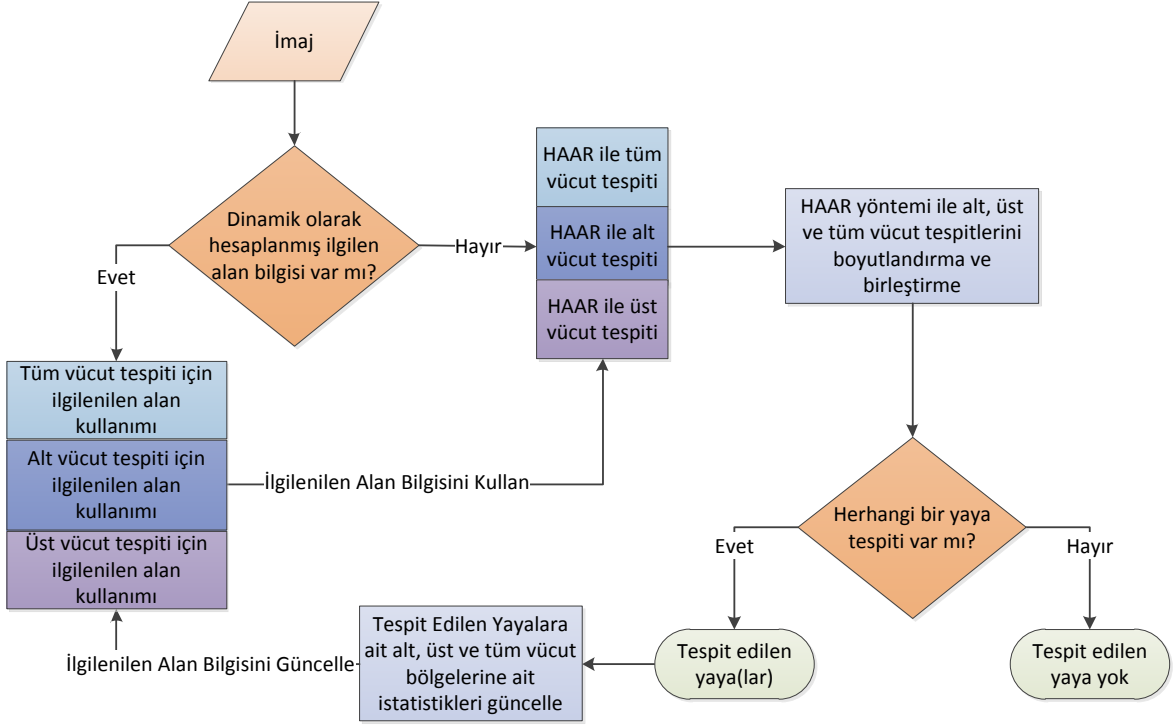


Şekil 3-7 İlgilenilen Alanın Ön Bilgi ile Sabit Belirlenmesi (Görüntü kaynağı: [33])

Şekil 3-7'de sarı, mor ve kırmızı dikdörtgenlerin ilgilenilen alanları belirtmektedir. Sarı dikdörtgenin içindeki alan üst vücut için ilgilenilen alanı belirtirken, mor dikdörtgenin içindeki alan alt vücut için ilgilenilen alanı belirtir. Kırmızı dikdörtgenin içindeki alan ise tüm vücut için ilgilenilen alanı belirtir. Yani, üst vücut tespitine ait tanımlayıcı sadece sarı alanda üst vücut tespit etmeye çalışırken diğer alanlarda herhangi bir işlem gerçekleştirmez. Bu yaklaşım tespit süresini yarıya yakın kısaltırken hatalı pozitif tespitlerin yapılmasının da önüne geçer.

3.4.2 Ön Bilgi Kullanılmadan Yapılan Dinamik İlgilenilen Alan Seçimi

Dinamik olarak ilgilenilen alan (İA) seçimi, ön bilgi olmadan yaya tespit sistemi çalıştıkça elde edilen verileri istatistiksel olarak toplayıp sistemin gelecekte kullanmasını amaçlar. Bu yaklaşıma ait akış şeması aşağıda verilmiştir.

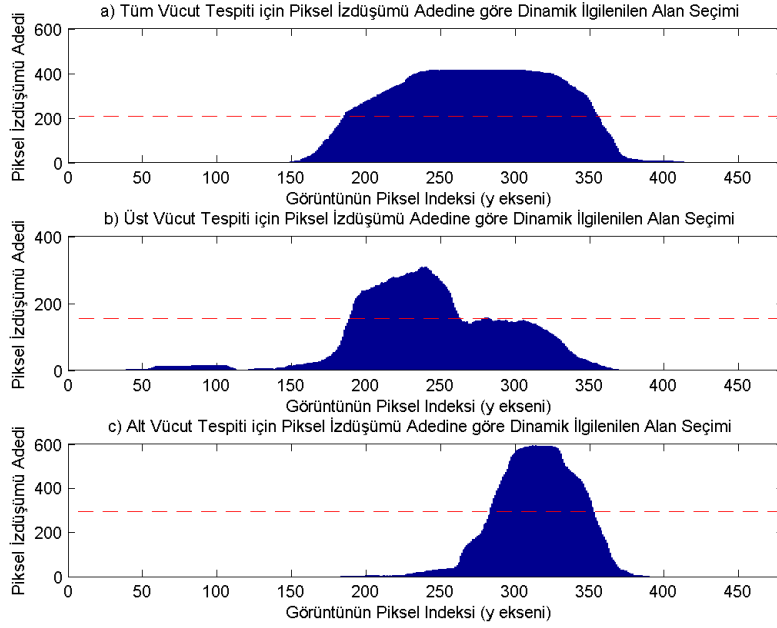


Şekil 3-8 Dinamik İlgilenilen Alan Seçimi Yaklaşımı

Şekil 3-8’de verilen dinamik ilgilenilen alan seçimi algoritma akışında sistem önce herhangi bir ön bilgi olmadan alt, üst ve tüm vücut tespitlerini gerçekleştirir. Her tespitten sonra ilgili tespitlere ait ilgilenilen alan bilgileri güncellenir. Elde edilen ilgilenilen alan bilgisi sonraki yaya tespit işlemlerinde kullanılır. Bu yaklaşımın başarılı sonuçlar vermesi için dinamik olarak oluşan bu bilginin ilk kullanıma sunulmasını 100 tespitten sonra başlatılmalıdır. Yani, 100 adet alt vücut tespitine ait görüntüdeki konum bilgilerinin istatistiki ortalaması alındıktan sonra bu bilgi sonraki yaya tespitleri için uygun hale gelmektedir.

İlgilenilen alan bilgisinin güvenilirliğini artırmak için belirli sayıda verinin toplanması gerektiğinden dolayı istatistiki bir yaklaşım geliştirilmiştir. Geliştirilen yaklaşım ile 100 tüm, alt ve üst vücut tespitinden sonra bir histogram elde edilir. Histogram şu şekilde elde edilmektedir:

- I. Y eksenindeki piksellerin adedi kadar histogram bacağı oluştur.
- II. Her bir görüntüde tüm, alt ve üst vücut tespiti yap.
- III. Yapılan her bir tespit için tespitlerin y ekseninde işgal ettiği çizgi kadar histogramın o piksellere denk gelen bacaklarındaki adedi 1 artır.



Şekil 3-9 Tüm, Alt ve Üst Vücut Tespitleri Histogramı ile Dinamik Alan Seçimi

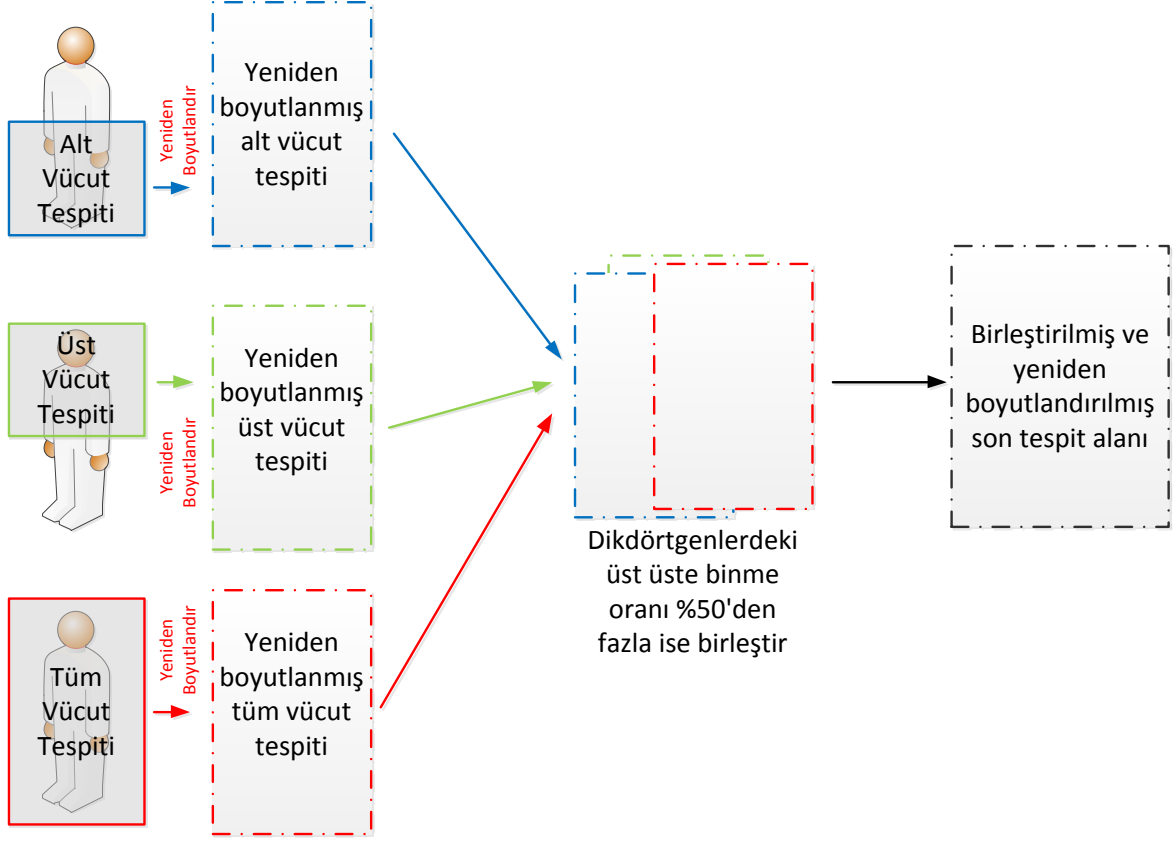
Şekil 3-9(a, b, c)'de tüm, üst ve alt vücut için yapılan dinamik alan seçiminde kırmızı kesikli eşik değerini belirtmektedir. Eşik değeri tepe noktasının yarısı olarak belirlenmiştir. Eşik değerinin üzerini geçen histogram değerleri tüm vücut tespiti için ilgilenilen alanın sınırlarının y ekseninde 190 ile 350 arasında olduğunu belirler. Eşik değeri yaklaşımı alt ve üst vücut ilgilenilen alan seçimi için de kullanılmıştır. Buna göre üst ve alt vücut dinamik ilgilenilen alan seçimi için oluşturulan histogramda sırayla y eksenli sınırları 190-260 ve 290-360 olmaktadır. Dinamik ilgilenilen alan seçimi yaklaşımı ile alt, üst ve tüm vücut için kullanılacak ilgilenilen alan seçimi ön bilgi kullanılmadan elde edilmiş olmaktadır. Bu yaklaşım sayesinde ortam değişse bile tespit hızı ve oranındaki performansın devamlılığı hedeflenmiştir. İlgilenilen alan bilgisi sistemde istenildiği zaman tekrar hesaplatılarak sisteme farklı ortamlarda farklı ilgilenilen alan tanımlaması yaptırılabilir. Bu sayede farklı ortamlarda tespit hızı ve tespit oranı performansı korunabilir.

3.5 Farklı Tespitlerin Birleştirilmesi

Haar yöntemi kullanılarak gerçekleştirilen yaya tespitinde alt, üst ve tüm vücut tespitleri farklı boyutlarda ve farklı pozisyonlarda yer alır. Çünkü aranan parçasal ve tüm bilgi görüntünün farklı yerlerindedir. Bizim ilgilendiğimiz bilgi ise özel olarak

tüm tespitlerin birleşimi olan tek bir yayanın görüntü üzerindeki boyutları ve konum bilgisidir. Bu sebeple, aynı yaya için elde edilen farklı boyut ve konumlarda elde edilen alt, üst ve tüm vücut yaya tespitlerinin birleştirilmesi gereklidir.

Aynı yayaya ait farklı tespitlerin birleştirilmesi için uygulanan yöntem aşağıdaki şekilde verilmektedir.



Şekil 3-10 Aynı Yayaya Ait Farklı Tip Tespitlerin Birleştirilmesi

Şekil 3-10'da sol tarafta aynı yayaya ait muhtemel 3 farklı tespit verilmektedir. Sırayla aynı yayaya ait alt, üst ve tüm vücut tespitleri önce tekrar boyutlandırılıp alt ve üst vücut tespitleri ortalama yaya boyutlarına çıkarılır. Boyutlandırma sırasında elde edilen alt ve üst yaya tespitine ait boyut sabit bir çarpan ile yeniden boyutlandırılır. Seçilen çarpan en düşük ve en yüksek yaya boyutları göz önüne alınarak seçilmektedir. Sonra, varsa 3 tespite ait dikdörtgen alanlarının yatay eksenindeki üst üste binme oranı %50'den fazla ise tespitler birleştirilir. Birleştirme işlemi aynı yayaya ait muhtemel elde edilmiş tespitleri birleştirerek son tespit

aşaması olan HOG yöntemi için çok daha rafine ve anlamlı tespitin ulaşmasını sağlar.

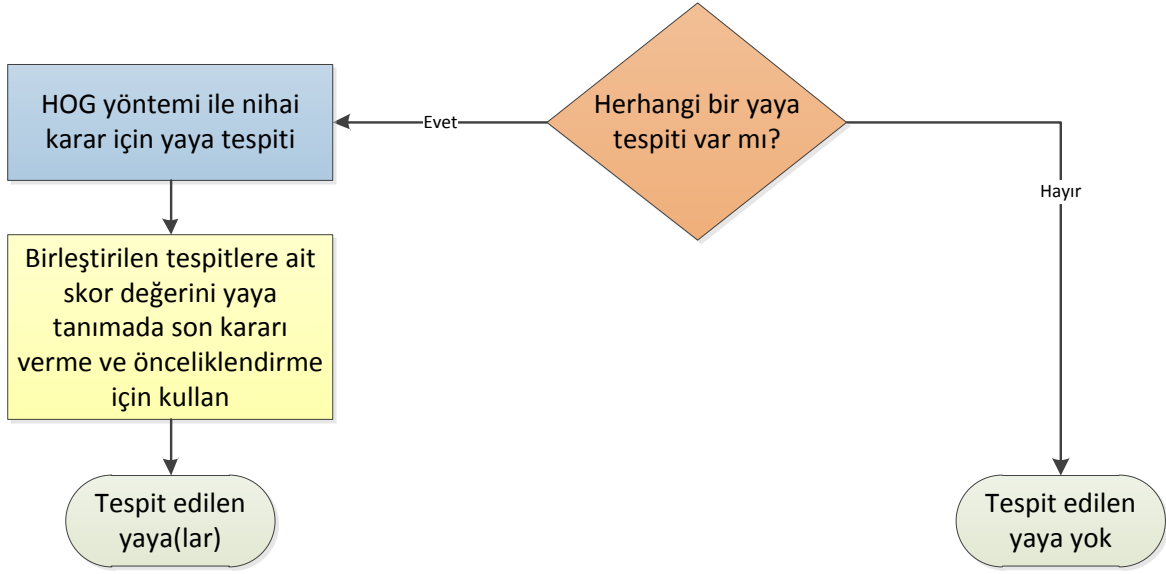
3.6 Haar Ön İşleme ile HOG Yönteminde Elde Edilen İyileşme

Tez çalışmasında nihai tespit için kullanılan HOG tespit edicisinin performansını artırmak için Haar ön işleme kullanılmaktadır.

3.7 Skor Kıstası ile Birleştirilen Tespitleri Önceliklendirme

Yaya tespitinde hatalı tespitin önüne geçmek ve daha iyi ve güvenilir tespit sonuçları elde etmek için tespit sonuçlarına belirli bir güvenilirlik skoru ile atanabilir. Bu yaklaşımı esas alarak, 4 farklı tip (Haar-tüm, Haar-alt, Haar-üst ve HOG-tüm vücut) tespit birleştirilirken tespit tiplerine farklı skorlar verilerek sureti ile tespitlere öncelik atanması planlanmıştır.

Skor atamanın yaya tespitinin hangi aşamasında olduğunu göz önüne getirmek için Şekil 3-1'de verilen ana sistem akış diyagramında tespit edilen yaya bilgisinin elde edilmesi alt kısmının detayları alınarak Şekil 3-11'da tekrar verilmektedir. Bu şemaya göre, Haar yöntemi kullanılarak gerçekleştirilen 3 farklı tip sınıflandırıcıdan hiçbir tespit çıkmazsa sistem HOG yöntemine girmeden yaya olmadığı bilgisini verir. Haar yöntemi kullanılarak gerçekleştirilen 3 farklı tip sınıflandırıcıdan en az bir tane tespit çıkarsa birleştirilen tespitlere ait dikdörtgen alanlar HOG yöntemi ile nihai yaya varlığı testine tabi tutulur. Bu aşamadan sonra ise elde edilen tüm tespitler bu bölümde anlatılan skor kıstasına tabi tutularak sistem akışı sonlandırılır. Bu yaklaşımın detayları sistem akış şemasından alınarak aşağıdaki şemada tekrar gösterilmiştir.



Şekil 3-11 Tespit Edilen Yayaların Nihai Karar Aşaması Akış Şeması

Skor kıstası, birleştirilen tespitleri öncelik sırasına koymak ve tespitlerin güvenilirlik seviyesini elde etmek için gereklidir. Haar yöntemine ait alt ve üst vücut tespitleri %15 alırken, Haar yöntemine ait tüm vücut tespiti skor olarak %30 almaktadır. HOG yöntemine ait tüm vücut tespiti ise skor olarak %40 almaktadır. Farklı tip tespitlerin birleşimleri ise aşağıdaki Tablo 3-2'ye göre skor atanmaktadır. Tablo 3-2'de yaya tespiti var ise 1 ile yok ise 0 ile ifade edilen ikili sayı sistem kullanılmaktadır. Bu şekilde $2^4=16$ adet olasılık vardır.

Tablo 3-2 Birleştirilen Tespitler için Kullanılan Skor Tablosu

HOG tüm (% 40)	Haar tüm (% 30)	Haar üst (% 15)	Haar alt (% 15)	Toplam Skor (%)
0	0	0	0	0
0	0	0	1	15
0	0	1	0	15
0	0	1	1	30
0	1	0	0	30
0	1	0	1	45

0	1	1	0	45
0	1	1	1	60
1	0	0	0	0*
1	0	0	1	60
1	0	1	0	60
1	0	1	1	70
1	1	0	0	70
1	1	0	1	85
1	1	1	0	85
1	1	1	1	100

*: Haar yöntemine ait 3 farklı tip (alt, üst ve tüm) tespitin de negatif çıkması durumunda HOG yaya tespiti mümkün olmadığından skor atanmamaktadır.

Örneğin, yalnız üst vücut yaya tespiti var ise skor %15 olurken, yalnız tüm vücut yaya tespiti var ise skor kıstasından %30 elde edilmektedir. Bu şekilde bir skor yöntemi seçilmesinin sebebi, tüm vücut tespiti için kullanılan tanımlayıcının hata yapma olasılığının düşük olması sebebiyledir. Yani tüm vücut tespiti yapılmışsa bu sonuç üst veya alt vücut tespitinden daha öncelikli ve güvenilir olmalıdır. Bu yüzden tüm vücut tespitine daha yüksek skor verilmektedir.

Tablo 3-2'ye göre, Haar yöntemine ait alt ve üst vücut yaya tespiti var ise skor %30 olurken; Haar yöntemine ait tüm, alt ve üst vücut yaya tespiti var ise skor %60 olmaktadır. Bu 3 tespitinin üzerine HOG tespiti de pozitif çıkarsa toplam skor %100 olur. Yani, 4 farklı tip tespit birlikte gerçekleşir ise elde edilen nihai tespit en yüksek güvenilirliğe sahiptir.

Skor kıstası için hazırlanan tablo farklı ihtiyaçlar ve ortamlar için değiştirilerek tespitler arasındaki öncelik sırası uygulamaya göre ayarlanabilir. Örneğin, çoğunlukla üst vücut tespitinin kolay ve erişilebilir olduğu bir uygulamada üst vücut tespiti sonucuna daha yüksek bir skor atabilir. Haar yöntemi ile elde edilen yaya

tespitlerini birleřtirdikten sonra skor kıstası kullanılarak sistem řu görevleri gerekleřtirir:

- Gerek zamanlı sűrűcű uyarı sisteminde tespit edilen birden ok yayanın ncelik sırasına konması ve en yűksek gűvenilirlik skora sahip yayanın bulunduėu konumun sűrűcűye bildirilmesi saėlanabilir. rneėin, video kamerada hem saėda hem solda iki tane yaya varsa ve saėdaki yaya daha yűksek gűvenilirlik skoruna sahip ise sűrűcűye “dikkat saėda yaya var” uyarısı verilerek tespitler arasında ncelik atanması saėlanmış olur.
- Nihai kararı veren HOG yntemi ile birleřtirilmiř tespit űzerinde yaya varlıėı onaylanamaz ise dahi skor kıstası kullanılarak, elde edilen skor belirli bir sayının űzerinde ise sűrűcű uyarılabilir. Bunun yapılabilmesi iin sűrűcűye farklı alarm seviyeleri semesi iin bir ara yűz sunulur: dűřűk, orta ve yűksek alarm seviyeleri vs.

4. ÇALIŞMANIN FARKLI PLATFORMLAR ÜZERİNDE GERÇEKLENMESİ

Sistemin farklı donanım ve yazılım ortamlarındaki performansına bakmak ve aynı zamanda sistemin farklı donanım ve yazılım platformlarında çalışabildiğini göstermek için 3 farklı yazılım-donanım ortamında testler gerçekleştirilmiştir. Testlerin detayları ve test ortamı sonraki 3 bölümde verilmektedir. Testlerde, geliştirilen yöntem olan HAAR(FUL) yönteminin literatürde sıkça kullanılan yalnız Haar ve yalnız HOG yöntemi ile tespit oranı ve tespit hızı bağlamında karşılaştırması yapılmaktadır.

4.1 Testlerde Kullanılan Parametreler

HOG ve Haar yöntemlerini kullanarak yapılan testlerde kullanılan parametre setleri Tablo 4-1’de verilmektedir. Tabloda verilen değerler nihai testlerden önce yapılan geliştirme faaliyetleri sonucunda belirlenen değerlerdir. Bu değerler, kullanılan veri setinde yaya tespit hızını ve tespit oranını tez çalışması en iyi yapan değerler olarak belirlenmiştir.

Tablo 4-1 Testlerde Kullanılan Parametre Setleri

	Haar Tüm Vücut Tespit Edici Parametreleri	Haar Alt Vücut Tespit Edici Parametreleri	Haar Üst Vücut Tespit Edici Parametreleri	HOG Tüm Vücut Tespit Edici Parametreleri
En düşük yayaya ait boyut	En: 55; Boy: 120	En: 30; Boy: 35	En: 30; Boy: 65	En: 55; Boy: 120
En yüksek yayaya ait boyut	En: 120; Boy: 230	En: 60; Boy: 80	En: 50; Boy: 100	En: 120; Boy: 230
Tanımlayıcı Boyutlandırma Katsayısı	1,05	1,05	1,05	1,05

Tabloda HOG ve Haar tüm vücut tespit edicisinde aynı parametreler kullanılmaktadır. Alt ve üst Haar tespit edicisinde ise alt ve üst vücut bölgelerinin boyutları tüm vücuttan farklı olduğu için farklı en düşük ve en yüksek vücut boyutu belirlenmiştir.

Tanımlayıcı boyutlandırma katsayısı testlerde kullanılan ve geliştirme sürecinde en iyi hale getirmek için üzerinde zaman harcanan bir parametredir. Bu parametre ile düşük bir boyuta sahip bir tanımlayıcı boyutları (Örneğin En: 14; Boy: 28) her seferinde tekrar boyutlandırılarak tanımlayıcının sisteme girilen en düşük ve en yüksek yaya boyutlarına göre yaya tespit etmesine olanak sağlar. Parametrenin büyük olması tespit hızını artırıp tespit oranını düşürürken parametrenin küçük olması tespit hızını düşürüp tespit oranını artırmaktadır. Bu yüzden bu parametrenin doğru seçimi çok önemlidir. Bu sebeple tez çalışmasında tanımlayıcı boyutlandırma katsayısı parametresi geliştirme sürecinde üzerinde durulan bir parametre olmuştur.

4.2 Testlerde Kullanılan Veri Seti ve Yaklaşım

Testler, 3 farklı yöntem için gerçekleştirilmiştir. Bu yöntemler sırayla:

- Sadece HOG yöntemi ile tüm vücut tespiti (HOG)
- Sadece Haar yöntemi ile tüm vücut tespiti (HAAR)
- Haar alt, üst ve tüm vücut tespiti yöntemi ile birlikte HOG yöntemi kullanılarak yapılan tespit (HAAR (FUL))

Test için Max-Planck Enstitüsü tarafından açık kullanıma sunulan 640x480 çözünürlüğüne sahip 100 adet yaya içeren fotoğraf kullanılmıştır [33]. Görüntü üzerinde tespit edilmeye çalışılan yayaların boyutları {genişlik, yükseklik} olarak {70, 160} ile {140, 230} piksel arasındadır. 100 adet yaya içeren fotoğrafta toplam 629 adet yaya bulunmaktadır.

Testlerde kullanılan veri seti için yaya konumlarını belirten bir dosya bulunmaktadır. Dosyanın her bir satırında her bir görüntü için yaya konumları verilmektedir. Bu dosyada bulunan yaya konumları ile tespit edilen yayaların konumları kıyaslanarak toplam yaya tespit sayısı, yaya olmayan tespit sayısı, toplam tespit edilemeyen yaya sayısı hesaplanmaktadır.

Testlerde elde edilen yaya tespit hızına ait sonuçlar grafiğe aktarılırken 10'luk yumuşatıcı filtre kullanılmaktadır. Test sonuç grafiklerinde ve tablolarında verilen

yaya tespit oranları ve tespit hızları 100 görüntü üzerinden elde edilen ortalama değerlerdir.

4.3 CPU Tabanlı Sistemdeki Test Sonuçları

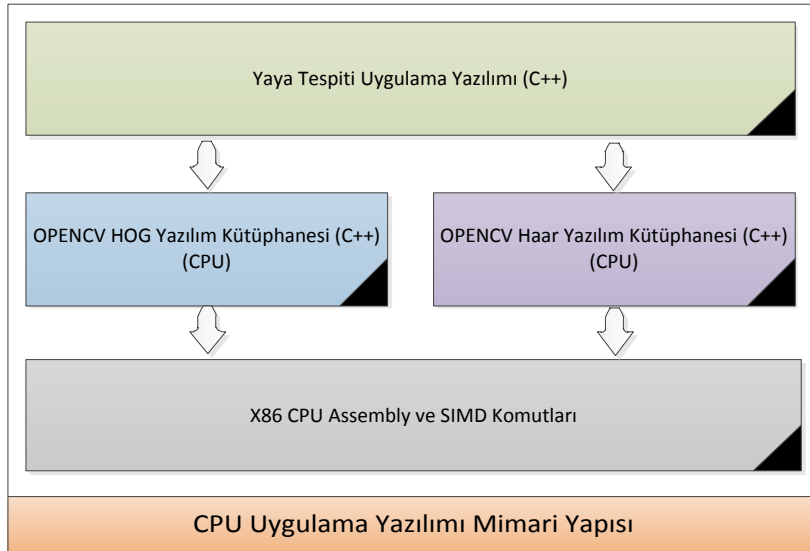
4.3.1.1 Donanım ve Yazılım Yapısı

Bu test için Intel X86 tabanlı bir bilgisayar kullanılmaktadır. Kullanılan bilgisayarın özellikleri aşağıdaki tablodaki verilmektedir.

Tablo 4-2 CPU Tabanlı Sistemdeki Donanım ve Yazılım Yapısı

CPU	Intel Core i7-4500U CPU @1,8 GHz Dual Core
Uçucu Bellek	DDR 3, 8 GB
İşletim Sistemi	Windows 7 64 bit

CPU için geliştirilen yaya tespit yazılımı uygulamasının mimarisi aşağıdaki şekilde verilmektedir. Yazılım OPENCV kütüphanesinin üzerinde paralel işlemeye olanak verecek şekilde geliştirilmiş ve derlenmiştir. Sistemde, gerçek zamanlı sistem ve algoritma geliştirmeye uygun C++ yazılım dili ve kütüphaneleri kullanılmıştır.

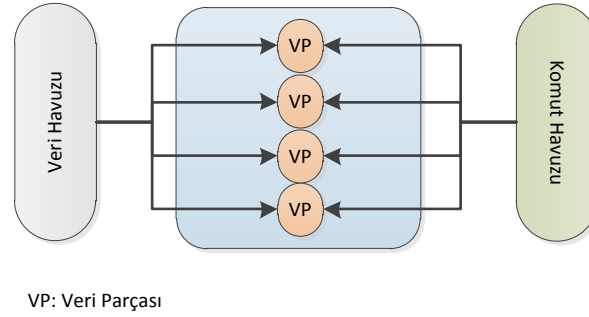


Şekil 4-1 CPU Tabanlı Sistemdeki Yaya Tespit Yazılımı Mimari Yapısı

Şekil 4-1’de verilen mimari yapıya göre hiyerarşik olarak en tepede yaya tespiti uygulama yazılımı yer almaktadır. Yaya tespiti uygulama yazılımı, OPENCV HOG

ve Haar yöntemlerini çalıştıran yazılım kütüphanelerini kullanmaktadır. OPENCV yazılım kütüphanesi kullanılırken sınıflandırıcı olarak Haar yöntemi için eğitilen 3 farklı vücut yapısı için kullanılan sınıflandırıcılar kullanılmaktadır. Geliştirilen algoritmaya ait kaynak kodu ve OPENCV yazılım kütüphanesine ait kaynak kodu x86 tabanlı işlemciye göre derlenerek kullanılmaktadır.

Şekil 4-1'de kullanılan SIMD kısaltması tek komut-çoklu veri anlamına gelmektedir [41]. SIMD mimari yapısı aşağıdaki şekilde verilmektedir.



Şekil 4-2 SIMD Mimari Yapısı [41]

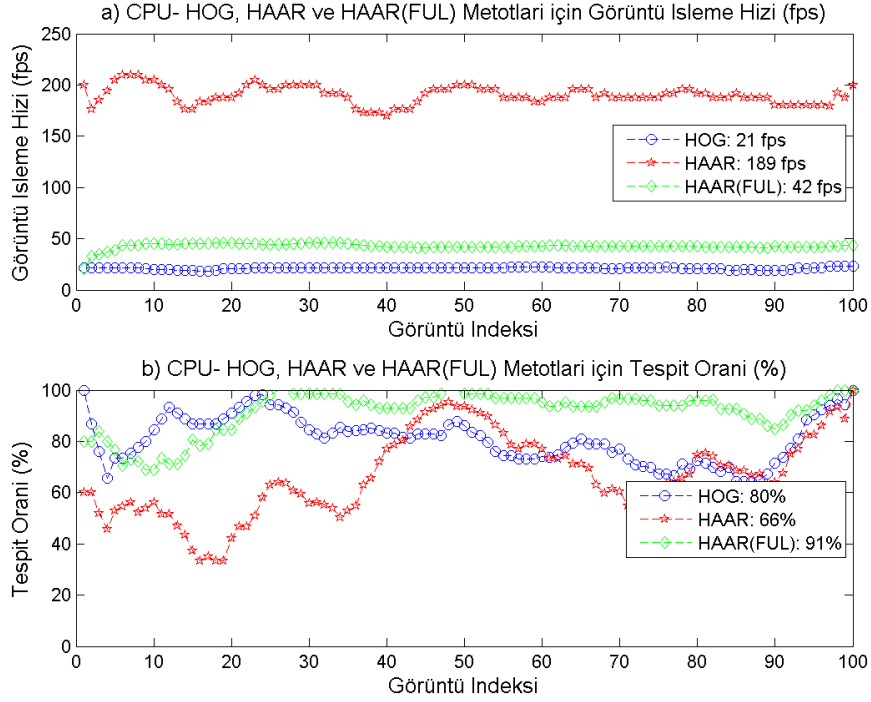
Şekil 4-2'de verilen komut seti aynı anda birden çok toplama ve çarpma işleminin yapılmasına imkân vererek OPENCV yazılım kütüphanesinin daha hızlı çalışmasını sağlar. Bu komut seti, tek bir komutla aynı anda 4 adet çarpma işlemi başlatma kararı alarak paralel işlemeye elverişli algoritmalara hız avantajı kazandırır.

4.3.1.2 Test Sonuçları

CPU tabanlı sistem için geliştirilen yöntem ve klasik yöntem için ayrı ayrı testler gerçekleştirilmiştir. Klasik yöntemde ilgilenilen alan seçimi yaklaşımı yoktur. Klasik yöntemin ve geliştirilen yöntemin test sonuçları sonraki iki bölümde verilmektedir.

4.3.1.2.1 Geliştirilen Yöntemin Test Sonuçları

Testler sonucunda elde edilen sonuçların grafiksel gösterimi Şekil 4-3'te gösterilmektedir. Bu şekillerden birincisi görüntü indeksine (1'den 100'e kadar) göre çerçeve işleme hızını (fps); ikincisi ise görüntü indeksine göre tespit oranını yüzde olarak (%) belirtmektedir. Bu tez kapsamında geliştirilen yöntem HAAR(FUL) adı altında grafiklerde yer almaktadır. Tüm vücut tespiti hedefli Haar yöntemi HAAR olarak ve yalnız HOG yöntemi ise HOG adı altında grafiklerde yer almaktadır.



Şekil 4-3 CPU Tabanlı İşlemci için 3 Yöntemin Karşılaştırılması

Şekil 4-3(a)'da verilen görüntü indeksine göre görüntü işleme hızı grafiğine (üstteki grafik) göre HAAR yöntemi HOG yöntemine göre yaklaşık 9 kata yakın hız artışı sağlamaktadır. Hız artışı olumlu olsa da Şekil 4-3'te verilen görüntü indeksine göre tespit oranı grafiği incelendiğinde tespit edilen yaya oranında HAAR yönteminde HOG yöntemine göre düşüş olduğu (%80'den %66'ya) gözlemlenmektedir. Geliştirilen yöntem olan HAAR (FUL) yöntemi ile HOG yöntemine göre tespit hızında %100 artış (21 fps'den 42 fps'ye) sağlanmıştır. Aynı zamanda, Şekil 4-3(b)'de verilen grafikte görüleceği üzere HAAR (FUL) yöntemi ile HOG yönteminden iyi (%11 fazla), HAAR yönteminden ise çok daha iyi (%25 fazla) tespit oranı elde edildiği gözlemlenmiştir. Bu sayede HOG yöntemine göre tespit oranında artış sağlanırken aynı zamanda tespit hızında da artış sağlanmaktadır.

Şekil 4-3'te verilen grafikteki verilerin tablosal gösterimi ve ek veriler Tablo 4-3'te verilmektedir.

Tablo 4-3 CPU Tabanlı İşlemci için 3 Yöntemin Yaya Tespit Rakamları

Yöntem	HOG	HAAR	HAAR(FUL)
Toplam yaya sayısı	629		
Tespit edilen yaya sayısı	505	431	581
Tespit edilemeyen yaya sayısı	124	198	47
Yaya olmayan tespit sayısı	3	2	6
Tespit hızı (fps)	34	194	58
Tespit hızı standart sapması (fps)	0,4	14,6	2
Tespit oranı (%)	80	66	91

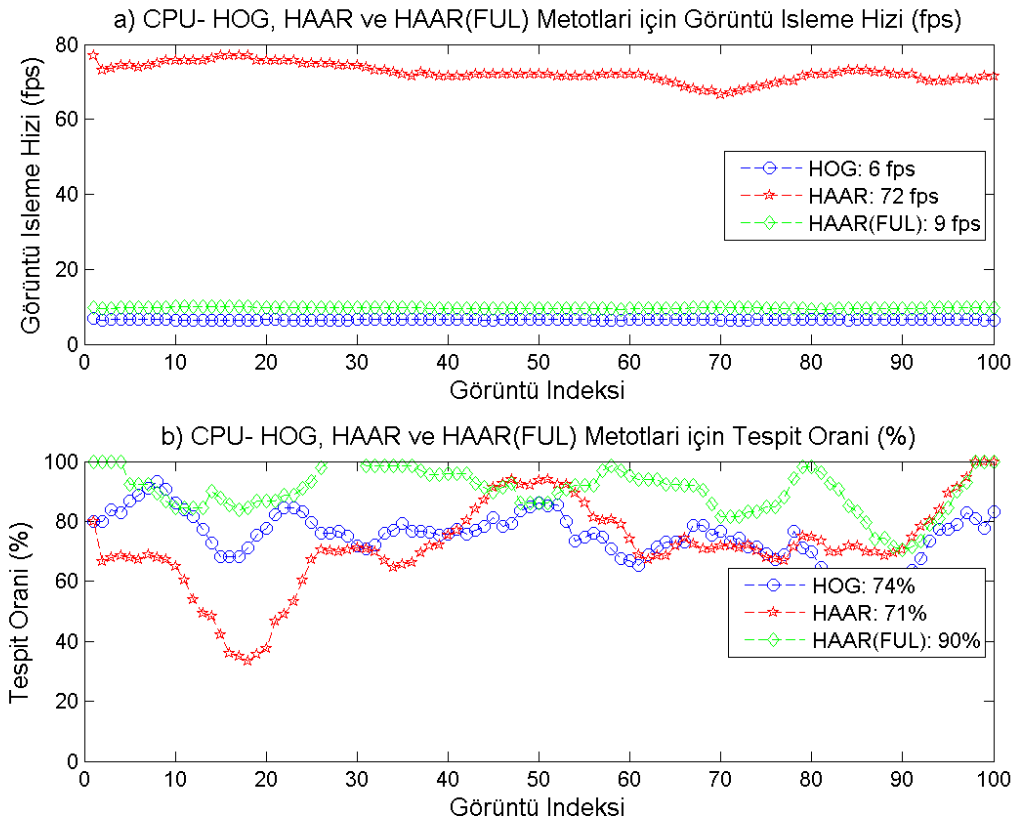
Tablo 4-3 incelendiğinde geliştirilen yöntem olan HAAR(FUL) için yaya tespit sayısının diğer yöntemlere göre arttığı gözlemlenmektedir. Geliştirilen yöntem ile 100 farklı görüntüde yer alan toplam 629 yayanın 581'i geliştirilen yöntem ile tespit edilebilmektedir. Tespit oranı bu durumda %91 olmaktadır.

100 farklı görüntü üzerinden yapılan tespit hızı ölçümünde standart sapma HAAR yöntemi için 9 fps olarak çıkmaktadır. Sapmanın fazla olması bu yöntemin gerçek zamanlı görüntü işleme için uygun olmadığı anlamına gelmektedir. Geliştirilen yöntemde ise tespit hızındaki standart sapma ise 2 fps olarak bulunmuştur. Standart sapmanın düşük olması yöntemin gerçek zamanlı görüntü işleme için uygun olduğunu ortaya koymaktadır. Çünkü gerçek zamanlı görüntü işlemede önemli hususlardan biri görüntü işleme hızının sabit olmasıdır. Değişken görüntü işleme hızına sahip bir yaya tespit yöntemi ile gerçek zamanlı görüntü işleme yapmak mümkün değildir [11].

Sonuç olarak, geliştirilen yöntem tespit hızında ve tespit oranında HOG yöntemine göre sırayla %100'e yakın hız artışı ve %11 tespit oranı artışı sağlamaktadır. Geliştirilen yöntem, gerçek zamanlı ve tespit kritik bir sistemden beklenildiği gibi %90 üzeri tespit oranına ulaşmaktadır [18].

4.3.1.2.2 Klasik Yöntemin Test Sonuçları

Görüntü üzerinde bir ön işleme yapılmasının (ilgilenilen alan seçimi) önemini vurgulamak ve geliştirilen yöntemin tespit hızı ve tespit oranı HAAR ve HOG yöntemlerine göre ortaya koymak maksadı ile geliştirilen yöntem ile klasik HOG ve HAAR yöntemleri ile karşılaştırılmıştır. Bunun için algoritma ilgilenilen alan seçimi olmadan koşturulmuştur. Bu tez kapsamında geliştirilen yöntem HAAR(FUL) adı altında grafiklerde yer almaktadır. Tüm vücut tespiti hedefli Haar yöntemi HAAR olarak ve HOG yöntemi ise HOG adı altında grafiklerde yer almaktadır.



Şekil 4-4 CPU Tabanlı İşlemci için Klasik HOG ve Haar Yöntemi Sonuçları

Şekil 4-4'te verilen klasik HOG ve Haar yöntemlerine ait sonuçları Şekil 4-3'te verilen geliştirilen yöntem ile karşılaştırdığımızda tespit hızında ciddi bir düşüş gözlemlenmektedir. Geliştirilen yöntemle ait tespit hızı ortalama 42 fps iken hiçbir ilgilenilen alan seçimi olmadan kullanılan HAAR (FUL) yönteminde tespit hızı ortalama 9 fps'ye düşmektedir. Tespit oranında değişiklik kayda değer bir değişiklik (%91'den %90'a) gözlenmemektedir. Ancak, geliştirilen yöntemin klasik

yönteme göre tespit hızındaki artış (9 fps'den 42 fps'ye) gerçek zamanlı görüntü işleme için gerekli performansı sağlamaktadır.

Geliştirilen yöntemde, yaya alt, üst ve tüm vücut tespiti için ayrı ayrı ilgilenilen alan seçimi yapılmaktadır. Sonuç olarak ön bilgi olarak ilgilenilen alan kullanılmasının tespit hızına olumlu etkisi klasik yöntemle ait tespit hızının geliştirilen yöntemle göre çok daha düşük olmasından anlaşılmaktadır.

4.4 GPU ve CPU Tabanlı Sistemdeki Test Sonuçları

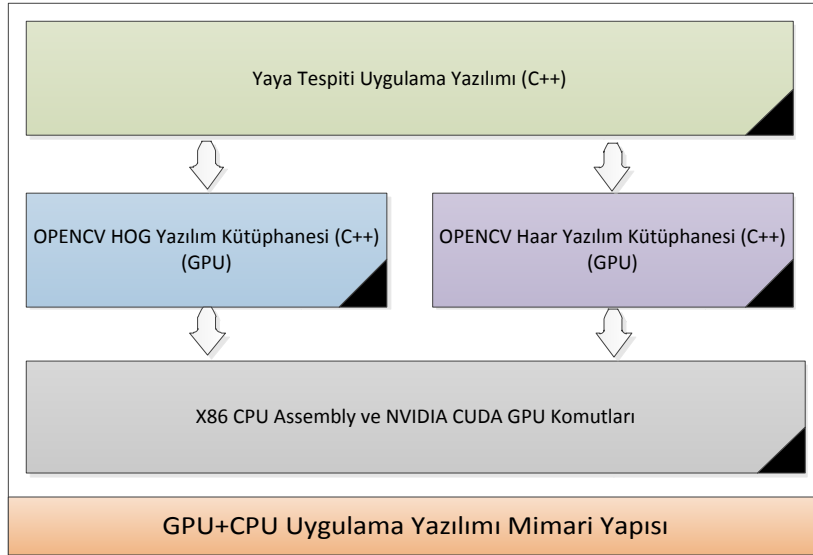
4.4.1.1 Donanım ve Yazılım Yapısı

Bu test adımımda NVIDIA CUDA tabanlı GPU'ya sahip dizüstü bilgisayarda GPU ve CPU birlikte kullanılarak yaya tespiti yapılmaktadır. Bu testin önceki CPU testinden farkı, yaya tespiti için kullanılan HOG ve Haar yöntemlerinin tamamen GPU üzerinde çalıştırılmasıdır. Algoritmanın GPU üzerinde koşan tespit kısmından sonra tespit birleştirme ve birleştirilen nihai tespite skor atama kısımları ise CPU üzerinde çalışmaktadır. Kullanılan donanımın özellikleri aşağıdaki tablodaki verilmektedir.

Tablo 4-4 GPU ve CPU Tabanlı Sistemdeki Donanım ve Yazılım Yapısı

CPU	Intel Core i7-4500U CPU @1,8 GHz Dual Core
Uçucu Bellek	DDR 3, 8 GB
GPU	NVIDIA GEFORCE GT 740 M 64 bit @1,6-1,8 GHz
İşletim Sistemi	Windows 8,1 64 bit

GPU ve CPU kullanılarak geliştirilen yaya tespit yazılımı uygulamasının mimari yapısı aşağıdaki şekilde verilmektedir. Yazılım OPENCV kütüphanesinin üzerinde GPU üzerinde koşan yaya tespit algoritmaları ile diğer algoritmaların birlikte çalışması sağlanarak geliştirilmiştir. Algoritma için, gerçek zamanlı sistem geliştirmeye uygun C++ yazılım dili kullanılmaktadır.

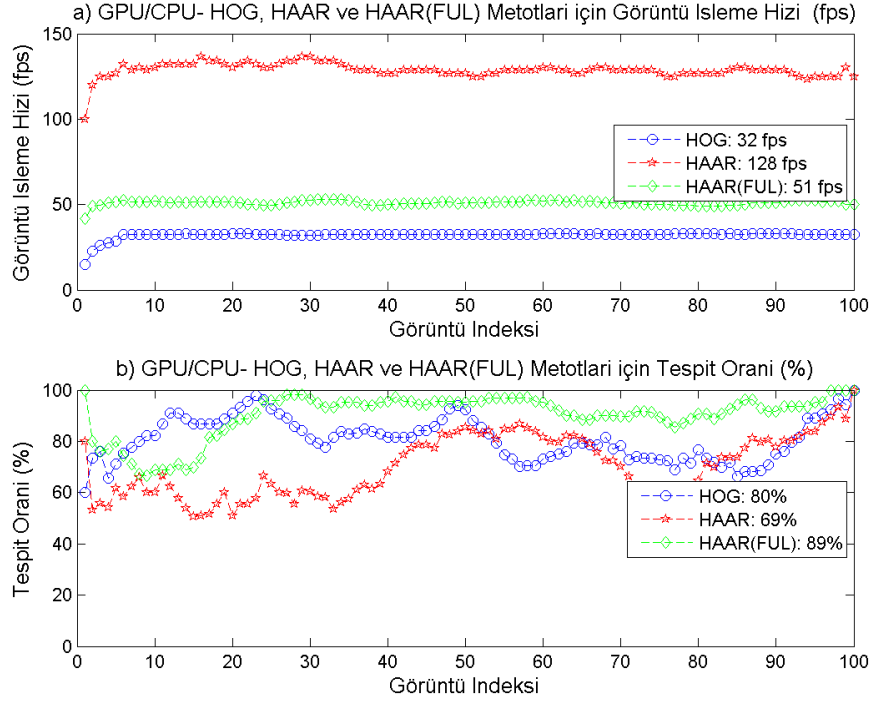


Şekil 4-5 GPU ve CPU Yaya Tespit Yazılımı Mimari Yapısı

Şekil 4-5'te verilen mimari yapıya göre hiyerarşik olarak en tepede yaya tespiti uygulama yazılımı yer almaktadır. Yaya tespiti uygulama yazılımı, OPENCV HOG ve Haar yöntemlerini çalıştıran yazılım kütüphaneleri kullanmaktadır. OPENCV yazılım kütüphanesi kullanılırken sınıflandırıcı olarak Haar yöntemi için eğitilen 3 farklı vücut yapısı için kullanılan sınıflandırıcılar kullanılmaktadır. OPENCV yazılım kütüphanesinde GPU üzerinde sınıflandırıcı algoritmasını koşturmak için gerekli yazılım kütüphaneleri yer alır. Geliştirilen algoritmaya ait kaynak kodu x86 tabanlı işlemciye için ve NVIDIA CUDA tabanlı GPU için derlenerek kullanılmıştır.

4.4.1.2 Test Sonuçları

Testte Bölüm 4.2'de verilen veri seti ve test yöntemi uygulanmıştır. Bu testte önceki testten farklı olarak Bölüm 4.4.1.1'de verilen donanım ve yazılım alt yapısı kullanılmıştır. Testler sonucunda elde edilen sonuçlara ait grafik Şekil 4-6'da gösterilmektedir. Bu şekillerden birincisi görüntü indeksine (1'den 100'e kadar) göre çerçeve işleme hızını (fps); ikincisi ise görüntü indeksine göre tespit oranını (%) belirtmektedir.



Şekil 4-6 GPU ve CPU Tabanlı İşlemci için 3 Yöntemin Karşılaştırılması

Şekil 4-6(a)'da verilen grafiğe göre tez kapsamında geliştirilen yöntem olan HAAR(FUL) HOG yöntemine göre tespit hızında %60 artış (32 fps'den 51 fps'ye çıkan hız) sağlanmıştır. Aynı zamanda, Şekil 4-6(b)'de verilen grafikte görüleceği üzere HAAR (FUL) yöntemi ile HOG yönteminden iyi (%9 fazla), HAAR yönteminden ise çok daha iyi (%20 fazla), tespit oranı elde edildiği gözlemlenmiştir.

Şekil 4-6'de verilen grafikteki verilerin tablosal gösterimi ve ek veriler Tablo 4-5'te verilmektedir.

Tablo 4-5 GPU/ CPU Tabanlı İşlemciler için 3 Yöntemin Yaya Tespit Rakamları

Yöntem	HOG	HAAR	HAAR(FUL)
Toplam yaya sayısı	629		
Tespit edilen yaya sayısı	507	451	573
Tespit edilemeyen yaya sayısı	122	178	58
Yaya olmayan tespit sayısı	4	3	6
Tespit hızı (fps)	32	128	51
Tespit hızı standart sapması (fps)	0,3	3,2	1,1
Tespit oranı (%)	80	69	89

Tablo 4-5 incelendiğinde geliştirilen yöntem olan HAAR(FUL) için yaya tespit sayısının diğer yöntemlere göre arttığı gözlemlenmektedir. Geliştirilen yöntem ile 100 farklı görüntüde yer alan toplam 629 yayanın 573'si geliştirilen yöntem ile tespit edilebilmektedir. Tespit oranı bu durumda %89 olmaktadır.

Tespit oranının CPU tabanlı sistemden %2 farklı olması GPU hesaplamasında kullanılan çekirdek yazılım kütüphanesinde GPU'ya özel değişiklikler ve farklılıklar barındırması sebebiyledir. GPU'da koşan yazılım ile CPU'da koşan yazılım bire bir aynı değildir. GPU belleğine gönderilen görüntü üzerinde yapılan işlemler donanıma ait çekirdek kütüphaneleri kullandığından ve kütüphaneler donanıma göre özelleştigiğinden dolayı sonuçlarda sapma olduğu düşünülmektedir. Yine de birbirinden oldukça farklı ortamlarda elde edilen yaya tespit oranı farklılığı ihmal edilebilir düzeydedir. Sonuç olarak, donanımsal ve mimari farklılıklar koşturulacak algoritma için yapılan özelleştirmelere etki ettiğinden dolayı tespit oranındaki küçük farklılık normal karşılanmıştır.

100 farklı görüntü üzerinden yapılan tespit hızı ölçümünde standart sapma HAAR yöntemi için 3,2 fps olarak çıkmaktadır. Sapmanın CPU tabanlı sisteme göre (9 fps) daha az olması donanım tabanlı görüntü işleminin gerçek zamanlı görüntü

işlemeye daha uygun olduğu anlamına gelmektedir. Geliştirilen yöntemde tespit hızındaki standart sapma ise 1,1 fps olarak bulunmuştur ve gerçek zamanlı görüntü işleme için daha uygundur.

GPU ve CPU kullanılarak yaptığımız test sonucunda CPU kullanılarak yapılan teste göre HOG yönteminde %52 civarında (21 fps'den 32 fps'ye) bir hız artışı gözlemlenirken, geliştirilen yöntem olan HAAR(FUL) yönteminde ise %21 civarında (42 fps'den 51 fps'ye) bir hız artışı tespit edilmiştir. Yani geliştirilen yöntemin CPU'dan CPU/GPU mimarisine geçişteki hız artışı HOG yöntemindeki hız artışından daha azdır. Bunun sebebi, Haar yönteminde kullanılan sınıflandırıcının basamaklı yapısıdır. Basamaklı yapıdan dolayı GPU üzerindeki paralel bloklardan yeterince fayda elde edilememekte ve bu yüzden ciddi bir hız artışı gözlenmemektedir.

4.5 Mobil İşlemci Tabanlı Sistemdeki Test Sonuçları

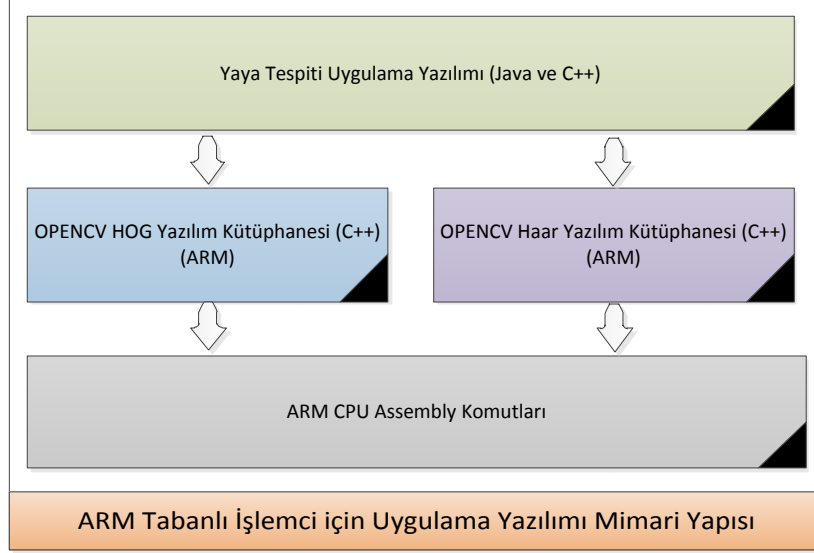
4.5.1.1 Donanım ve Yazılım Yapısı

Tez çalışması kapsamında geliştirilen yöntemin mobil işlemci üzerinde de çalıştırılmıştır. Geliştirilen yaya tespit sisteminin mobil bir platforma yüklenmesinin ve denenmesinin amacı sistem daha çok kullanıcının erişimine sunulmak maksadı ile yaygınlaştırıldığında ortaya çıkacak çalışma ve performans sonuçlarını görmektir. Bu sebeple ARM tabanlı işlemciye ve ANDROID işletim sahibine sahip mobil telefon yaya tespiti için kullanılmıştır. Algoritmanın tümü ARM tabanlı platform üzerinde koşturulmuştur. Kullanılan mobil platformun donanım ve yazılım özellikleri aşağıdaki tablodaki verilmektedir.

Tablo 4-6 Mobil İşlemci Donanım ve Yazılım Yapısı

CPU	QualComm MSM8930AB Snapdragon 400 @1,7 Ghz Dual-Core
Geçici Bellek	1,5 GB
İşletim Sistemi	ANDROID OS v4,2,2 (Jelly Bean)

ARM tabanlı işlemci kullanılarak geliştirilen yaya tespit yazılımı uygulamasının mimari yapısı aşağıdaki şekilde verilmektedir. Yazılım OPENCV kütüphanesinin üzerinde geliştirilmiştir. Yazılımın kullanıcı ara yüzü Java programlama dilinde, algoritmaya ait yazılım fonksiyonları ise gerçek zamanlı sistem geliştirmeye uygun C++ yazılım dili kullanılarak geliştirilmiştir.



Şekil 4-7 ARM İşlemcili Yaya Tespit Yazılımı Mimari Yapısı

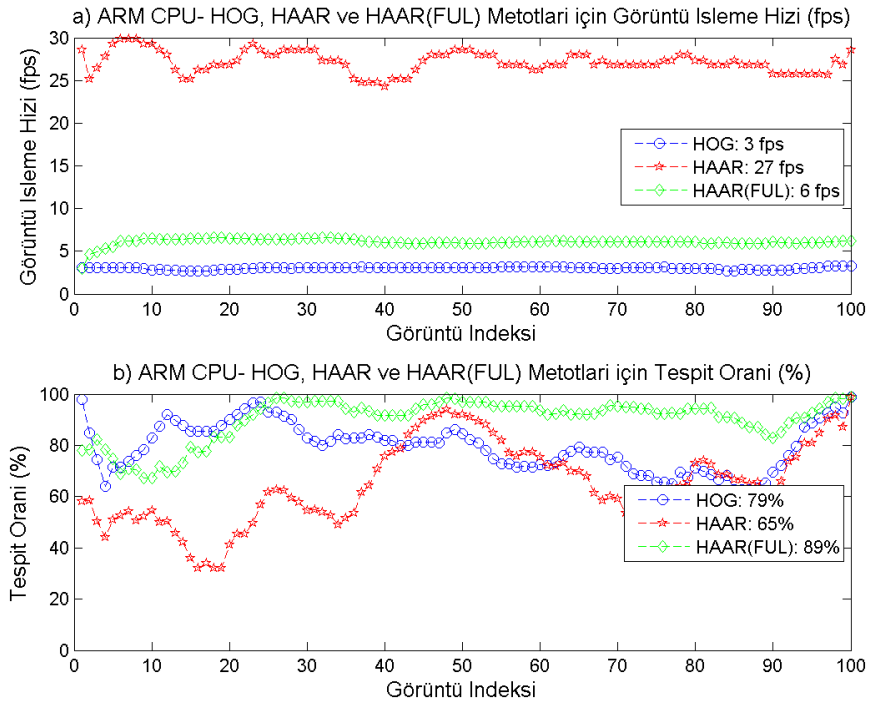
Şekil 4-7’de verilen mimari yapıda verilen yaya tespit uygulama yazılımının ara yüz parçaları JAVA dilinde; algoritmaya ait parçaları ise C++ dilinde geliştirilmiştir. C++ dilinde yazılan fonksiyonları JAVA ortamından çağırmak için JNI yapısı kullanılmıştır. Böylece ARM mimarisine uygun olarak derlenen uygulamada hem alt seviye hem üst seviye iki dil olan JAVA ve C++ birlikte kullanılmıştır. Literatürde yapılan çalışmalarda da benzer bir yaklaşımın yer aldığını görülmektedir [42]. Yani, yüksek performans gerektiren algoritma parçaları C++’de geliştirilirken; ara yüz kısımlarını JAVA dilinde geliştirmek performans için zorunlu bir tercih olmaktadır.

4.5.1.2 Test Sonuçları

Testte Bölüm 4.2’de verilen veri seti ve test yöntemi uygulanmıştır. Bu testte Bölüm 4.3.1.2’deki testten farklı olarak Bölüm 4.5.1.1’de verilen donanım ve yazılım alt yapısı kullanılmıştır. Testler sonucunda elde edilen sonuçlara ait grafik Şekil 4-8’de gösterilmektedir. Bu şekillerden birincisi görüntü indeksine (1’den

100'e kadar) göre çerçeve işleme hızını (fps); ikincisi ise görüntü indeksine göre tespit oranını (%) belirtmektedir.

Şekil 4-8(a)'ya göre geliştirilen yöntem olan HAAR (FUL) yöntemi ile HOG yöntemine göre tespit hızında %100 artış (3 fps'den 6 fps'ye) sağlanmıştır. Aynı zamanda, Şekil 4-8(b)'de verilen grafikte görüleceği üzere HAAR (FUL) yöntemi ile HOG yönteminden iyi (%10 fazla), HAAR yönteminden ise çok daha iyi (%23 fazla) tespit oranı elde edildiği gözlemlenmiştir. Bu sayede HOG yöntemine göre tespit oranında artış sağlanırken aynı zamanda tespit hızında da artış sağlanmaktadır.



Şekil 4-8 ARM Tabanlı İşlemci için 3 Yöntemin Karşılaştırılması

Şekil 4-8'de verilen grafikteki verilerin tablosal gösterimi ve ek veriler Tablo 4-7'da verilmektedir. Tablo 4-7 incelendiğinde geliştirilen yöntem olan HAAR(FUL) için yaya tespit sayısının diğer yöntemlere göre arttığı gözlemlenmektedir. Geliştirilen yöntem ile 100 farklı görüntüde yer alan toplam 629 yayanın 578'i geliştirilen yöntem ile tespit edilebilmektedir. Tespit oranı bu durumda %92 olmaktadır. 100 farklı görüntü üzerinden yapılan tespit hızı ölçümünde standart sapma HAAR

yöntemi için 3,3 fps olarak çıkmaktadır. Geliştirilen yöntemde tespit hızındaki standart sapma ise 0,3 fps olarak bulunmuştur ve gerçek zamanlı görüntü işleme için uygundur.

Tespit oranının CPU tabanlı sistemden %2 farklı olması ARM tabanlı işlemcide kullanılan çekirdek yazılım kütüphanesinde ARM işlemciye ve ANDROID işletim sistemine özel değişiklikler barındırması sebebiyledir. OPENCV kütüphanesinin hem farklı işletim sistemi hem de farklı işlemci mimarisinde çalıştırılmasında gözlenen bu farklılık donanım ve yazılım alt yapısının tamamen farklı olduğu iki sistem için kabul edilebilir düzeydedir.

Tablo 4-7 ARM Tabanlı Mobil İşlemci için 3 Yöntemin Yaya Tespit Rakamları

	HOG	HAAR	HAAR(FUL)
Toplam yaya sayısı	629		
Tespit edilen yaya sayısı	497	409	559
Tespit edilemeyen yaya sayısı	132	220	90
Yaya olmayan tespit sayısı	4	5	9
Tespit hızı (fps)	3	29	6
Tespit hızı standart sapması (fps)	0,2	2,3	0,6
Tespit oranı (%)	79	65	89

Sonuç olarak geliştirilen yöntem ile Mobil işlemci alt yapısı kullanılarak yapılan test sonucunda tespit hızında, yaygın olarak kullanılan HOG yöntemine göre, %100'e yakın bir artış sağlanmaktadır. Tespit oranında ise gerçek zamanlı ve tespit kritik bir sistemden beklenildiği gibi %89 civarında bir tespit oranına ulaşmaktadır.

Sonuç olarak mobil işlemci alt yapısı kullanılarak yaptığımız test sonucunda, elde edilen sistemin gerçek zamanlı olması için gereken tepkime süresinin (500 milisaniye ~ 2fps) oldukça altında (160 milisaniye civarında ~ 6 fps) iyi bir süre ile beklentileri karşılamaktadır. Geliştirilen yöntemde tespit hızındaki standart sapma

ise 0,6 fps olarak bulunmuştur ve yöntem gerçek zamanlı görüntü işleme için daha uygundur.

4.6 Test Sonuçlarının Değerlendirilmesi

Elde edilen test sonuçlarına göre geliştirilen yöntem ayrı ayrı HOG ve Haar yöntemlerin göre daha başarılıdır. Geliştirilen yöntemin HOG ve Haar yöntemlerine göre tespit hızında ve tespit oranında daha iyi olmasının bazı nedenleri vardır. Bu nedenler:

- HOG yöntemi sadece tüm vücut tespiti yaparken geliştirilen yöntem alt ve üst vücut bölgelerini de tespit ederek parçasal bilgi ile tespit yapabilir.
- HOG yöntemi Haar yöntemine göre daha yavaş bir yöntemdir. Elde edilen test sonuçlarında tüm vücut Haar yöntemi ile tüm vücut HOG yöntemi kıyaslandığında oldukça yavaş kalmaktadır (Bkz. 4.3.1.2.1). Geliştirilen yöntem ise ön işlemede hızlı olan Haar yöntemini kullanarak hızda ciddi avantaj sağlamaktadır. Geliştirilen yöntemde HOG yöntemi tüm görüntü üzerinden değil sadece nihai tespitler üzerinden işlem yaparak hesaplama zamanından ciddi kazançlar sağlanmaktadır (Bkz. 4.3.1.2.1).
- Haar yöntemi ile tüm vücut tespitinde elde edilen tespit oranının düşüklüğü problemi geliştirilen yöntemin alt ve üst vücut bölgelerini de tespit etmesi ve 3 farklı tespiti birleştirmesi sayesinde çözülmektedir. Bu sayede Haar yöntemi tüm vücut tespiti yanında üst ve alt vücut tespiti için de kullanılarak tespit hızındaki dezavantaj giderilmektedir.
- Geliştirilen sistemde Haar yöntemine ait 3 farklı tespitin çalışma sırası önemli değildir. 3 farklı tespit birbirinden bağımsız ve ayrı sırada çalışarak nihai tespit için bir alt küme oluşturur. Bu sayede istenirse 3 tespit edici paralel olarak çalışarak toplam tespit süresini daha da kısaltabilir.
- Haar tespit edicilerin sonuçlarından oluşan alt küme Bölüm 3.5'te anlatılan farklı tespitlerin birleştirilmesi işleminden geçerek nihai HOG tespit edicisine gider. Nihai tespitler HOG yönteminden de skor alırsa tespitin güvenilirlik katsayısı artar. Ancak HOG yöntemi olumlu yanıt vermese dahi Haar yöntemine ait tespitlerden elde edilen skor kullanıcıya uyarı vermek için yeterli olmaktadır.

3 farklı donanım ve yazılım yapısına sahip 3 farklı sistemde yaptığımız testler sonucunda GPU ve CPU birlikte kullanılarak gerçekleştirilen Bölüm 4.4'te verilen

yazılım ve donanım yapısı geliştirilen yöntem olan HAAR(FUL) için en yüksek tespit hızına ulaşılmaktadır. Mobil işlemciadaki uygulama ise elde edilen tespit hızı CPU veya CPU/GPU sistemlerinden yavaş olmakla birlikte, mobil platformdaki sistem, yaya tespit sisteminin gerçek zamanlı olması için gereken tepkime süresinin altında kalarak (< 500 milisaniye) ile beklentileri karşılamaktadır [11]. 3 farklı platform için de denenen sistem gerçek zamanlı uygulama için elverişlidir. Yaya tespiti için sistemden beklenen tepkime süresinin (500 milisaniye) çok daha altında bir sürede yayayı tespit edip uyarı vermek mümkün olmaktadır.

Bölüm 4'te verilen algoritma test sonuçlarına göre, geliştirilen algoritmadaki hesaplama işlemleri paralel hale getirildiği için ve algoritma donanıma ait hızlandırma fonksiyonlardan faydalandıkça tespit hızında artış sağlanmaktadır. Bu yaklaşım da literatürde donanım destekli yazılım olarak geçmektedir. Gün geçtikçe işlemcilerde özel hesaplama parçaları (modül) ve çekirdekleri geliştirilmekte ve bu çekirdekler yüksek performanslı ve paralel işleme ihtiyacı olan yazılımların kullanımına sunulmaktadır.

5. SONUÇLAR VE ÖNERİLER

Tez çalışmasında gerçek zamanlı uygulamalar için yaya tespit sistemi geliştirilmiştir. Bu kapsamda yaya tespit hızı ve tespit oranı gerçek zamanlı uygulama için uygun bir yöntem önerilmektedir. Geliştirilen yöntem 3 farklı tespiti birleştirerek ve nihai olarak HOG tespiti ile tespit edilen yayayı onaylayan çok katmanlı bir yapıya sahiptir. Çok katmanlı tespit sayesinde her bir tespit için ayrı skor belirlenerek nihai tespite tespit adedince ve tespitin ağırlığına göre (Bkz. 3.6) skor atanması sağlanmaktadır.

Geliştirilen yöntemin çok katmanlı tespit ve skor mekanizması ile tespit edilen hem sağda hem solda iki tane yaya varsa tespitlere öncelik atayabilir. Böylece, örneğin sağdaki yaya daha yüksek güvenilirlik skoruna sahip ise sürücüye “dikkat sağda yaya var” uyarısı verilerek tespitler arasında öncelik atanması sağlanmış olur. Öncelik atamanın yanında ortamdaki hatalı tespitleri engellemek için sürücüye farklı alarm seviyeleri seçmesi için bir ara yüz de sunulabilir: düşük, orta ve yüksek alarm seviyeleri vs. Böylece sürücü karmaşık ortamlarda alarm seviyesini yüksekte tutarak hatalı alarmları önlemiş olur.

Tez çalışmasında CPU, CPU/GPU ve mobil işlemci platformlarında aynı algoritma test edilerek farklı çözümler ele alınmıştır. Donanıma ait özel fonksiyonlar (GPU üzerinde paralel hesaplama yapmak) kullanıldığında tespit hızında kayda değer bir artış elde edilmektedir.

Gelecekte bu alanda yapılacak çalışmalarda FPGA tabanlı çözümlerin kullanılacağını öngörmekteyiz. Çünkü FPGA tabanlı çözüm ile probleme yönelik özel işlemci yapıları tasarlanıp geleneksel işlemcilerin hız kısıtları önemli oranda aşılabilmektedir. Literatürde yer alan bir çalışmada bazı geometrik şekillerin tanımlanmasında HOG yönteminin FPGA tabanlı çözümü gerçekleştirilmiş ve gelecek vaat eden performans ve doğruluklar elde edilmiştir [43].

Taşınabilirlik, uygulanabilirlik ve yaygınlaştırma bağlamında Bölüm 4.5'te verilen mobil platform diğer sistemlerden öndedir. Mobil platformlar yıldan yıla artan işlemci kapasitesi, işlemci çekirdek sayısı ve bellek miktarı ile bu tür uygulamalar için doğal çözüm olma yolundadır. Mobil platformların neredeyse her kesimden kullanıcı ve araç sürücüsü tarafından erişilebilir olması da yaya tespit sisteminin yaygınlaştırılması bağlamında çok büyük bir artıdır.

Mobil işlemcideki uygulamadaki tepkime süresinin iyi olması (~150 ms) sisteme yaya tespitine ek olarak bisikletli insanları ve trafik işaretlerini de tespit için de yeterli işlemci performansı ve süresi (en kötü tepki süresi 500 ms) tanımaktadır. Yani, sisteme ek yetenek kazandırmak için bisikletli insan, engelliler, sokak hayvanları (köpek vs.) ve trafik işaretlerini bulan tespit ediciler de eklenebilir. Bu sayede sistem trafikte sürücüye farkındalık oluşturmak adına ek yeteneklerle trafikte güvenliği bozabilecek diğer engellere de cevap verebilecek bir seviyeye getirilebilir.

KAYNAKLAR

- [1] D. Gavrila, "Sensor Based Pedestrian Protection," *Intelligent Systems*, vol. 16, no. 6, 77-81, **2001**.
- [2] "Mercedes-Benz Tech Center: PRE-SAFE," Mercedes-Benz, http://techcenter.mercedes-benz.com/en/pre_safe_brake_2013/detail.html (Ağustos **2015**).
- [3] D. Gavrila, J. Giebel, S. Munder, "Vision-based Pedestrian Detection: The Protector System," in *Intelligent Vehicle Symposium*, **2004**.
- [4] S. Heueli H. Rohling, "Pedestrian Recognition in Automotive Radar Sensors," in *International Radar Symposium*, **2013**.
- [5] T. Gandhi, M. M. Trivedi, "Pedestrian collision avoidance systems: a survey of computer vision based recent studies," in *Intelligent Transportation System Conference*, **2006**.
- [6] L. Xia, K. Fujimura, "Pedestrian detection using stereo night vision," *Vehicular Technology*, vol. 53, no. 6, 1657-1665, **2004**.
- [7] H. Nanda, L. Davis, "Probabilistic template based pedestrian detection in infrared videos," in *Intelligent Vehicle Symposium*, **2002**.
- [8] M. Enzweiler, D. M. Gavrila, "Monocular Pedestrian Detection: Survey and Experiments," *Pattern Analysis and Machine Intelligence*, vol. 31, no. 12, 2179-2195, **2009**.
- [9] Z. Shanshan, C. Bauckhage, D. Klein, A. Cremers, "Moving Pedestrian Detection Based on Motion Segmentation," in *Robot Vision*, **2013**.
- [10] Z. Liang, C. Thorpe, "Stereo- and neural network-based pedestrian detection," *Intelligent Transportation System*, vol. 1, no. 3, 148-154, **2000**.

- [11] D. Gavrilu, "Pedestrian detection from a moving vehicle," in *Lecture Notes in Computer Science*, vol. 1843, Berlin, Springer, 37-49, **2000**.
- [12] N. Dalal, T. Bill, S. Cordelia, "Human detection using oriented histograms of flow and appearance," in *Lecture Notes on Computer Vision*, Berlin, Springer, 428-441, **2006**.
- [13] T. Tsuji, H. Hattori, M. Watanebe, N. Nagaoka, "Development of Night-Vision System," *Intelligent Transportation System*, vol. 3, no. 3, 203-209, **2002**.
- [14] C. Wakim, S. Capperon, J. Oksman, "A Markovian Model of Pedestrian Behavior," in *Systems, Man, Cybernetics*, **2004**.
- [15] Y. Abramson, B. Steux, "Hardware-friendly pedestrian detection and impact prediction," in *Intelligent Vehicle Symposium*, **2004**.
- [16] H. Shimizu, T. Poggio, "Direction estimation of pedestrian from multiple still images," in *Intelligent Vehicle Symposium*, **2004**.
- [17] D. Makris, T. Ellis, "Spatial and Probabilistic Modelling of Pedestrian Behaviour," in *British Machine Vision Conference*, **2005**.
- [18] R. Benenson, M. Mathias, R. Timofte, L. V. Gool, "Pedestrian detection at 100 frames per second," in *Computer Vision and Pattern Recognition*, **2012**.
- [19] P. Viola, M. Jones, "Robust real-time face detection," in *Internation Journal of Computer Vision*, **2004**.
- [20] C. Keller, D. Liorca, D. Gavrilu, "Dense stereo-based roi generation for pedestrian detection," in *Pattern Recoginiton*, Berlin, Springer, 81-90, **2009**.
- [21] P. Dollar, C. Wojek, B. Schiele, P. Berona, "Pedestrian detection: A benchmark," in *Computer Vision and Pattern Recognition*, **2009**.
- [22] R. Benenson, R. Timofte, L. V. Gool, "Stixels estimation without depth map computation," in *Computer Vision Workshops*, **2010**.

- [23] A. Lehmann, P. Gehler, L. V. Gool, "Branch&rank: Non-linear object detection," in *British Machine Vision Conference*, **2011**.
- [24] N. Dalal, B. Triggs, "Histograms of Oriented Gradients for Human Detection," in *Computer Vision and Pattern Recognition*, **2005**.
- [25] D. G. Lowe, "Distinctive Image Features from Scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, 91-110, **2004**.
- [26] B. E. Boser, I. M. Guyon, V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the fifth annual workshop on Computational learning theory*, **1992**.
- [27] C. Corinna, V. Vladimir, "Support-Vector Networks," in *Machine Learning*, Boston, Kluwer Academic Publishers, 273-297, **1995**.
- [28] P. Viola, M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features," in *Computer Vision and Pattern Recognition*, **2001**.
- [29] W. Xing, Y. Zhao, R. Cheng, J. Xu, S. Lv, X. Wan, "Fast Pedestrian Detection Based on Haar Pre-Detection," *International Journal of Computer and Communication Engineering*, vol. 1, no. 3, 207-209, **2012**.
- [30] F. C. Crow, "Summed-area Tables for Texture Mapping," in *Computer Graphics and Interactive Techniques*, New York, **1984**.
- [31] R. Szeliski, "Face and Object Recognition," in *Computer Vision: Algorithms and Applications*, Springer, 663, **2010**.
- [32] P. Viola, M. Jones, "Robust Real-Time Object Detection," in *International Journal on Computer Vision*, **2001**.
- [33] M.-P. Institut, "Computer Vision and Multimodal Computing," Max-Planck Institut, <http://datasets.d2.mpi-inf.mpg.de/andriluka08cvpr/tud-crossing-sequence.tgz>. (Ağustos **2015**).

- [34] Y. Xin, S. Xiaosen, S. Li, "A Combined Pedestrian Detection Method Based on Haar-like Features and HOG Features," in *3rd International Workshop on Intelligent Systems and Applications (ISA)* , **2011**.
- [35] Y. Wei, Q. Tian, T. Guo, "An Improved Pedestrian Detection Algorithm Integrating Haar-Like Features and HOG Descriptors," *Advances in Mechanical Engineering*, vol. 2013, **2013**.
- [36] M. Kılıçarslan, J. Zheng, "Detecting Walking Pedestrians from Leg Motion in Driving Video," in *Intelligent Transportation Systems*, Qingdao, China, **2014**.
- [37] S. Zhang, C. Bauckhage, A. B. Cremers, "Informed Haar-like Features Improve Pedestrian Detection," in *Computer Vision and Pattern Recognition*, **2014**.
- [38] INRIA, "INRIA Pedestrian Data Set,"
<ftp://ftp.inrialpes.fr/pub/lear/douze/data/INRIAPerson.tar> (Ağustos **2015**).
- [39] OpenCV, "Downloads openCV," <http://opencv.org/>. (Ağustos **2015**).
- [40] N. Vasconcelos, M. J. Saberian, "Boosting Classifier Cascades," in *Advances in Neural Information Processing Systems 23*, **2010**.
- [41] "SIMD," Wikipedia, <https://en.wikipedia.org/wiki/SIMD> (Ağustos **2015**).
- [42] R. Parakkal, "Is Intel within ARM's reach? Pedestrian Detection shows the way," www.edn.com/electronics-blogs/systems-interface/4419918/Is-Intel-within-ARM-s-reach--Pedestrian-Detection-shows-the-way (Ağustos **2015**).
- [43] F. Karakaya, H. Altun, M.A. Çavuşlu, "Implementation of HOG algorithm for Real Time Object Recognition Applications on FPGA based Embedded System," in *IEEE Signal Processing and Communications Applications Conference*, **2009**.

EKLER

EK-1

Terimler Sözlüğü

Algılayıcı Tabanlı Birleştirme	Sensor based fusion
Boyutlandırma Katsayısı	Scale Coefficient
Çekirdek hilesi	Kernel trick
Çok boyutlu düzlem	Hyper plane
Gözetimli öğrenme	Supervised learning
Kuvvetlendirilmiş Basamaklı Sınıflandırıcı	Boosted cascade classifier
Tanımlayıcı	Descriptor
Tümleşik görüntü	Integral image
Varsayılan Yaya Tespit Edicisi	Default People Detector
Yön vektörü	Gradient

ÖZGEÇMİŞ

Kimlik Bilgileri

Adı Soyadı : Muhammed Enis MIHÇIOĞLU

Doğum Yeri : Hamburg/ ALMANYA

Medeni Hali : Evli

E-posta : enesmihci@gmail.com

Adresi : Osmanlı Mah. Paşa Sok. 12/5 Sincan/ANKARA

Eğitim

Lisans : Bilkent Üniversitesi Elektrik Elektronik Mühendisliği Bölümü

Yüksek Lisans :

Doktora :

Yabancı Dil Düzeyi

İngilizce-iyi, Almanca-başlangıç

İş Deneyimi

2006- TÜBİTAK İLTAREN / Uzman Araştırmacı

Deneyim Alanları

Elektronik Destek Sistemlerinde işaret işleme, kullanıcı ara yüzleri, gerçek zamanlı işletim sisteminde yazılım geliştirme, gömülü sistemler vs.

Tezden Üretilmiş Projeler ve Bütçesi

Tezden Üretilmiş Yayınlar

Tezden Üretilmiş Tebliğ ve/veya Poster Sunumu ile Katıldığı Toplantılar