

GERÇEK ZAMANLI UYKULU SÜRÜŞ ALGILAMA SİSTEMİ

Bilgisayar Müh. Sedat GOLGİYAZ

**Yüksek Lisans Tezi
Bilgisayar Mühendisliği Anabilim Dalı
Danışman: Prof. Dr. Erhan AKIN
TEMmuz-2013**

T.C
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

GERÇEK ZAMANLI UYKULU SÜRÜŞ ALGILAMA SİSTEMİ

YÜKSEK LİSANS TEZİ

Bilgisayar Müh. Sedat GOLGİYAZ

(102129102)

Tezin Enstitüye Verildiği Tarih : 09 Temmuz 2013

Tezin Savunulduğu Tarih : 31 Temmuz 2013

Tez Danışmanı : Prof. Dr. Erhan AKIN (F.Ü)

Diğer Jüri Üyeleri : Yrd. Doç. Dr. Haluk EREN (F.Ü)

Yrd. Doç. Dr. Mehmet KARAKÖSE (F.Ü)

TEMMUZ-2013

ÖNSÖZ

Son senelerde bilgisayar teknolojisinin ucuzlaması ve gömülü yazılım sektöründe büyük çaplı gürbüz uygulamaların geliştirilmesiyle birlikte, maddi ve can kayıplı trafik kazalarının azaltılmasına yönelik birçok çalışma yapılmıştır ve yapılmaktadır. Uykulu Sürüş Algılama Sistemi 2011'in son yarısında lüks taşıtlarda kullanılmaya başlanmasına rağmen bu yönde yapılan akademik ve ticari çalışmaların gerekli olgunluğa eriştiği söylenemez. Trafik kazalarının sosyal ve ekonomik boyutları düşünüldüğünde Uykulu Sürüş Algılama Sistemi ileride de araştırmacıların ilgisini çekmeye devam edecek bir araştırma konusudur. Bu tez çalışmasında mevcut Uykulu Sürüş Algılama Sistemleri incelenmiş icra zamanı ve içerdiği yeniliklerle gerçek zamanlı olarak koşabilecek bir çalışma sunulmuştur.

Çalışmalarım konusunda bilgi ve desteğini esirgemeyen değerli danışman hocam sayın Prof. Dr. Erhan AKIN'a, lisans ve yüksek lisans eğitimim boyunca bana destek olan tüm hocalarım ve aileme teşekkür ederim.

Sedat GOLGİYZ

Elazığ, 2013

İÇİNDEKİLER

	<u>Sayfa No</u>
ÖNSÖZ	II
İÇİNDEKİLER	III
ÖZET	VI
SUMMARY	VII
ŞEKİLLER LİSTESİ	VIII
TABLolar LİSTESİ	X
SEMBOLLER LİSTESİ	XI
KISALTMALAR	XII
1. GİRİŞ	1
1.1. Uyku ve Trafik Kazaları	2
1.2. Ülkemizde Uykulu Sürüş Bilançosu	3
1.3. Uykulu Sürüş Kazalarında Risk Altında Olan Sürücüler	3
1.3.1. Uykusu Bozulmuş veya Yorgun Sürücüler	3
1.3.2. Genç Sürücüler	4
1.3.3. Vardiyalı Çalışan Sürücüler	4
1.3.4. Ticari Araç Sürücüleri	4
1.3.5. Tanı Konulmamış Uyku Bozuklukları Olan Sürücüler	4
1.4. Uykulu Sürüşten Kaynaklanan Kazaların Belirtileri	5
1.5. Uyuklama ve Yorgunluk Halinin Belirtileri	5
1.6. Mevcut Uykulu Sürüş Algılama Sistemi Çalışmaları	6
1.6.1. Fiziksel Vücut Özelliklerinin İncelenmesi	6
1.6.2. Aracın Tepkilerinin Algılanması	7
1.6.3. Sürücünün Verdiği Uyarının Algılanması	7
1.7. Kapsam	8
2. MATERYAL ve METOT	10
2.1. Biyometrik Sistemler ve Akıllı Araç Uygulamaları	10
2.2. Bilgisayarlı Görme	11
2.2.1. Görüntü Yakalama	11
2.2.2. Önışleme	12
2.2.3. Bölütleme	12
2.2.5. Sınıflandırma	12
2.3. Opencv Kütüphanesi	13
2.3.1. Core Modülü	13

2.3.2.	İmgproc Modülü.....	14
2.3.3.	Video Modülü.....	14
2.3.4.	Calib3d Modülü.....	14
2.3.6.	Objdetect Modülü.....	14
2.3.7.	Highgui Modülü	15
2.3.8.	Gpu Modülü.....	15
2.4.	Uykulu Sürüş Algılama Sistemi	15
2.4.1.	Yüz Algılama.....	17
2.4.1.1.	Yüz Tespit Yöntemleri	18
2.4.1.1.1.	Bilgi Tabanlı Yöntemler.....	19
2.4.1.1.2.	Özellik Tabanlı Yöntemler	21
2.4.1.1.3.	Şablon Eşleme Tabanlı Yöntemler.....	21
2.4.1.1.4.	Görünüm Tabanlı Yöntemler	21
2.4.1.2.	Opencv ve Yüz Tespiti	24
2.4.1.2.1.	Viola-Jones Yüz Tespit Yöntemi	25
2.4.1.2.1.1.	Haar Benzeri Özellikler ve Genişletilmiş Haar Benzeri Özellikler.....	25
2.4.1.2.1.2.	İntegral İmge	28
2.4.1.2.1.3.	Adaboost Öğrenme	29
2.4.1.2.1.4.	Elemeli Kaskad Yapı.....	32
2.4.1.2.2.	LBP Yöntemi.....	34
2.4.2.	Yüz Tanıma	39
2.4.2.1.	Yüz Tanıma Yöntemleri	40
2.4.2.1.1.	Resimden Yüz Tanıma Yöntemleri	41
2.4.2.1.1.1.	Özellik Tabanlı Yöntemler	41
2.4.2.1.1.2.	Bütünsel Yöntemler.....	44
2.4.2.1.1.2.1.	İstatistiksel Yöntemler.....	44
2.4.2.1.1.2.2.	AI Yöntemleri.....	45
2.4.2.1.1.3.	Çoklu Sınıflandırıcı Sistemler	47
2.4.2.1.2.	Video Dizilerinden Yüz Tanıma	48
2.4.2.1.3.	Diğer Algılayıcılarda Yüz Tanıma	48
2.4.2.2.	Opencv ve Yüz Tanıma	49
2.4.2.2.1.	Opencv Yüz Tanıma Modelleri	49
2.4.2.2.2.	LDA (Fisher) Yüz Tanıma Algoritması ve Modeli	50
2.4.3.	Göz Tespiti	53

2.4.3.1. Göz Tespit Yöntemleri	54
2.4.3.1.1. Şekil Tabanlı Yöntemler.....	54
2.4.3.1.1.1.1. Basit Eliptik Şekil Modelleri	54
2.4.3.1.1.1.2. Karmaşık Şekil Modelleri.....	55
2.4.3.1.2. Özellik Tabanlı Yöntemler	56
2.4.3.1.3. Görünüm Tabanlı Yöntemler	57
2.4.3.1.4. Melez Yöntemler	57
2.4.3.1.5. Diğer Yöntemler	58
2.4.3.2. Opencv ve Göz Tespiti	58
2.4.4. Uykulu Sürüş Algılama Sistemleri	58
2.4.4.1. Uykulu Sürüş Algılama Sisteminde Çıkış Üretilmesi	59
3. DDDS UYGULAMASI	62
3.1. Yüz Tespit Uygulaması	62
3.1.1. Yüz İçin Ön İşlemler	65
3.1.1.1. Yeniden Boyutlandırma.....	65
3.1.1.2. Yüzün Sağ ve Sol Kısmı İçin Histogram Eşitleme	66
3.1.1.3. Yüz Filtreleme	66
3.1.1.4. Yüzün Yüz Bileşenlerinden Ayrıştırılması	67
3.2. Yüz Tanıma Uygulaması	68
3.3. Göz Algılama Uygulaması	70
3.3.1. Göz İçin Ön İşlemler	72
3.3.1.1. Yeniden Boyutlandırma.....	73
3.3.1.2. Göz Bölgesi İçin Histogram Eşitleme	73
3.3.1.3. Göz Filtreleme	73
3.3.1.4. Gözün Ayrıştırılması	74
3.4. Geliştirilen Göz Durumu Tespit Uygulaması.....	74
4. SONUÇLAR ve BULGULAR	78
KAYNAKLAR.....	86
ÖZGEÇMİŞ.....	96

ÖZET

Uykulu sürüş her ülkede olduğu gibi ülkemizde de büyük maddi kayıplı ve ölümlü trafik kazalarına sebep olmaktadır. Bu sistemler sosyal ve ticari değeri yönüyle üzerinde araştırma yapılmaya değer bir konudur.

Bu çalışmada bir Uykulu Sürüş Algılama Sistemi sunulmuştur. Bu sistem video tabanlı test verileri ile yürütülmektedir. Modelin uygulaması OpenCV kütüphanesinin 2.4.5 sürümü kullanılarak yapılmıştır. Visual C++ programlama dili, Visual Studio 2010 ortamında kullanılmıştır. Test için ZJU (Zhejiang University) veri tabanı kullanılmıştır. Bu veri tabanı standart bir veri tabanı olup daha önce Uykulu Sürüş Algılama çalışmalarında kullanılmıştır. Mevcut Uykulu Sürüş Algılama Sistemi temel olarak 4 modülden oluşmaktadır. Birinci modül de video yakalayıcıdan gelen görüntüden yüz tespit edilir. İkinci modülde konumlandırılmış yüz görüntüsünden kişi tanınır. Üçüncü modül de konumlandırılmış yüz görüntüsünden gözler bulunur. Son modül ise göz görüntülerinden göz kırpma oranını hesaplar ve göz kırpma oranı ile sürücünün yola bakıp bakmamasına göre sürücünün ikaz edilip edilmeyeceğine karar verir.

Bu tez çalışmasında sınıflandırıcı olarak yüz ve göz tespiti için kaskad sınıflandırıcı kullanılmıştır. Yüz tanıma için LDA (Linear Discriminant Analysis) yöntemi kullanılmıştır. Göz durumu tespiti için son senelerde işlem basitliği ve sağlamlığıyla ilgi çekip gerçek zamanlı uygulamalarda tercih edilen LBP (Local Binary Pattern) yöntemi kullanılmıştır.

Geliştirilen sistemin test edildiği veri tabanında ışık kontrolünün olmaması, arka planın dinamik olması ve aynı deneklerin gözlüksüz ve iki çeşit gözlük takmasıyla elde edilen veriler olmasından dolayı elde edilen sonuçların kısmen de olsa gerçek verilere yakın olduğu söylenilebilir.

Anahtar Kelimeler: Uykulu Sürüş Algılama Sistemi, Yüz Bulma, Yüz Tanıma, Göz Bulma, Göz Durumu Tespit.

SUMMARY

DROWSY DRIVING DETECTION SYSTEM

Drowsy driving in Turkey as in every country causes of great financial lossy and fatal traffic accidents. These systems which have aspects of the social and commercial value worth doing research on the issue.

In this study, a Drowsy Driving Detection System is presented. The system is implemented with video-based test data. Application of the model is carried out using version 2.4.5 of OpenCV library. Visual C + + programming language was used in the Visual Studio 2010 environment. To test the system ZJU (Zhejiang University) database were used. This is a VIIort heVII database and that have been used on Drowsy Driving Detection System previously. Current Drowsy Driving Detection System mainly consists of 4 modules. The first module is detected face image from the video capture device. The second module is recognized person from positioned face images. The third module is located the eyes from positioned face images. The last module calculates the ratio of the eye blink from images of eyes and it decides whether to alert the driver by blink rate and regretting the driver looked at the road.

In this study, the cascade classifier was used VIIort he detection of the face and eyes. For face recognition LDA (Linear Discriminant Analysis) method was used. VIIort he detection of eye state the method of LBP (Local Binary Pattern) is used which in recent years, pull interest with simplicity of operation and durability preferred for real-time applications.

Due to, the proposed system's testing database which the absence of light control and to the dynamic nature of the background and the same subjects without glasses and wore two kinds of glasses, the results of the the system obtained can be said to be at least partly close to actual data.

Keywords: Drowsy Driving Detection, Face Detection, Face Recognition, Eye Finder, Eye State Detection.

ŞEKİLLER LİSTESİ

Sayfa No

Şekil 2. 1. Bilgisayarlı görü sistemi.....	11
Şekil 2.2. Örnek bir DDDS diyagramı [20]......	16
Şekil 2.3. Bilgisayar sistemleri için yazılım katmanları [30].	17
Şekil 2.4. Örnek bir resim için düşük çözünürlüklü kopyaların elde edilmesi [33].	20
Şekil 2.5. Örnek bir resimlerin yatay/düşey iz düşümünün alınması [33].	20
Şekil 2.6. Yapay sinir ağları ile yüz algılama şeması [54].	23
Şekil 2.7. Viola-Jones çalışması için özellikler	26
Şekil 2.8. Yüz tanıma için örnek Haar benzeri özellik elde edilmesi [34].	26
Şekil 2.9. Genişletilmiş Haar benzeri özellikler için kenar özellikleri (a), çizgi özellikleri (b) ve merkezi çevreleyen özellikler (c) [57].	27
Şekil 2.10. İntegral imge ile alt dörtgen alanı hesaplama.....	29
Şekil 2.11. Yükseltme için şematik bir örnek [31].	30
Şekil 2.12. Viola-Jones yöntemi için elemeli kaskad yapı diyagramı	33
Şekil 2.13. Temel LBP operatörünün uygulanaşı	34
Şekil 2.14. ELBP operatörü için örnekler [36].	35
Şekil 2.15. Benzersiz, rotasyona karşı değişmez, $LBP_{8,R}^i$ için LBP örüntüleri [36].	36
Şekil 2.16. ILBP operatörünün uygulanaşı.....	37
Şekil 2.17. ELBP operatörü örneği [78].	38
Şekil 2.18. Yüzün LBP yöntemi ile ifade edilmesi [79].	39
Şekil 2.19. Yüz tanıma işlemi için geometrik özellikleri (beyaz) [84].	42
Şekil 2.20. EBGM yönteminde yüz tanıma için ızgaralar [86].	43
Şekil 2.21. Temel PCA (a) ve 1D iz düşürülmüş PCA (b) [81].	45
Şekil 2.22. Temel bir YSA modeli örneği	46
Şekil 2. 23. Yüz tanıma işlemi için çoklu sınıflandırıcı örneği [88].	47
Şekil 2.24. Göz modeli için parametreler [48].	55
Şekil 3.1. Tasarlanan DDDS modeli blok diyagramı.....	62
Şekil 3. 2. Tasarlanan yüz tespit uygulaması blok diyagramı	63
Şekil 3.3. DDDS Uygulaması girişleri için örnek renkli (a) ve gri kareler (b)	64
Şekil 3.4. DDDS için örnek tespit edilen yüz (a) ve histogramı eşitlenmiş yüz (b).	66
Şekil 3.5. Filtrelenmiş yüz.....	67
Şekil 3.6. Maskelenmiş yüz.....	68

Şekil 3.7. Tasarlanan yüz tanıma uygulaması blok diyagramı	68
Şekil 3.8. Tasarlanan yüz tanıma uygulamasında elde edilen eğitim verisi örneği.....	70
Şekil 3.9. Tasarlanan göz tespit uygulamasının blok diyagramı	71
Şekil 3.10. Göz algılama girişi	71
Şekil 3.11. Tespit edilen göz, sağ (a) ve sol (b) örneği	72
Şekil 3.12. HE yapılmış sağ (a) ve sol (b) göz örneği	73
Şekil 3.13. Filtrelenmiş göz, sağ (a) ve sol (b) örneği.....	74
Şekil 3.14. Maskelenmiş sağ (a) ve sol (b) göz örnekleri	74
Şekil 3.15. Tasarlanan göz kırpma ve PERCLOS hesaplama uygulaması blok diyagramı ..	75
Şekil 3.16. Göz durumu tespiti için açık (a) ve kapalı (b) göz eğitim verisi örnekleri	76

TABLÖLAR LİSTESİ

	<u>Sayfa No</u>
Tablo 4.1. Yüz tespit modülünün minFeatureSize parametresi için sistem çıktıları.....	79
Tablo 4.2. minNeighbors = 3 için searchScaleFactor bağlı göz başarımları ile saniyede işlenen kare sayısı	80
Tablo 4.3. minNeighbors = 4 için searchScaleFactor bağlı göz başarımları ile saniyede işlenen kare sayısı	80
Tablo 4.4. Yüz tespit modülünün, 4 video türü için uygulama sonuçları.....	81
Tablo 4.5. Yüz tanıma modülünün 4 video türü için uygulama sonuçları	82
Tablo 4.6. Göz tespiti modülünün 4 video türü için uygulama sonuçları	82
Tablo 4.7. ZJU veri tabanı için tespit edilen göz kırpmaları.....	83

SEMBOLLER LİSTESİ

A	: Matris
A^{-1}	: Ters Matris
AA	: Matris çarpımı
Av	: Matris vektör çarpımı
$e_{i,j}$	: Sınıflandırma Hatası
$f_j(x)$	: Dörtgen özellik
$h(x)$	: Adaboost sınıflandırıcı
$h_j(x)$	: Zayıf sınıflandırıcı
i	: Skaler
LBP^{ri}	: Rotasyona Bağımsız Local Binary Pattern
LBP^{SE}	: İstatistiksel Efektif Local Binary Pattern
LBP^{u2} , ULBP	: Tek şekil Local Binary Pattern
$LBP_{P,R}$	: Local Binary Pattern (R yarı çap, P komşular)
p_j	: Eşitsizlik yönü
$R_{n \times n}$	: $n \times n$ uzayında gerçekteki uzay
ROR	: Bit Düzeyinde sağa kaydırma işlemi
S_B	: Dağılım matrisi
S_w	: Dağılım matrisi
θ	: Açı
θ_j	: Eşik değeri
X	: Çarpma işlemi
v	: Vektör
$w_{i,j}$	: Ağırlıklar
W	: İz düşüm
W_{opt}	: Optimum iz düşüm
W_{pca}	: PCA iz düşüm
W_{fld}	: FLDA iz düşüm
y_i	: Test verisi etiketi
μ	: Ortalama
μ_i	: Sınıf hatası
δ	: Hata Oranı
(x_c, y_c)	: Orijinal resimdeki piksel değeri

KISALTMALAR

OpenCV	: Open Source Computer Vision
Matlab	: Matrix Laboratory
TÜİK	: Türkiye İstatik Kurumu
BSD	: Berkeley Software Distribution
DDDS	: Uykulu Sürüş Algılama Sistemi
NHTSA	: ABD Ulusal Otoyol Trafik Güvenliği Yönetimi
DSÖ	: Dünya Sağlık Örgütü
EEG	: Electroencephalography
CCD	: Charge Coupled Device
PERCLOS	: Percentage of eye closure
FPGA	: Field Programmable Gate Array
AI	: Artificial Intelligence
GPU	: Graphics Processing Unit
IPP	: Intel's Integrated Performance Primitives
1D	: Tek Boyutlu
2D	: İki Boyutlu
3D	: Üç Boyutlu
FPS	: Frame Per Second
Chi	: Chi-Kare Uzaklık Ölçütü
DCT	: Ayrık Kosinüs Transformasyonu
XML	: Genişletilebilir İşaretleme Dili
YAML	: Yet Another Multicolumn Layout
HE	: Histogram Eşitleme
HI	: Histogram Kesişimi Uzaklık Ölçütü
LBP	: Local Binary Pattern
LBPH	: Local Binary Pattern Histogram
LDA	: Linear Discriminant Analysis
MBLBP	: Multi-scale Block Local Binary Pattern
MLBP	: Multi-scale Local Binary Pattern
MLBPH	: Multi-scale Local Binary Pattern Histogram
PCA	: Principal Component Analysis
ICA	: Independent Component Analysis
EBGM	: Elastic Bunch Graph Matching
AAM	: Active Appearance Model
SVM	: Support Vector Machine

1. GİRİŞ

Her ülkede olduğu gibi ülkemizde de maddi ve can kayıplı trafik kazalarına sebep olan uykululuğu tespit sistemleri; hem sosyal yönüyle hem de ticari değeri yönüyle üzerinde araştırma yapılmaya değer bir konudur. Otomotiv firmalarının öncülüğünde akıllı araç uygulamalarına (parmak izini tanıyan, kendini park eden vb.) ilgi her geçen gün artmaktadır. Mercedes-Benz, Volvo, Saab, Nissan ve Hyundai gibi büyük otomotiv firmaları DDDS (Drowsy Driving Detecftion System) olarak adlandırılan Uykulu Sürüş Algılama Sistemi ile sürücü yorgunluk seviyesini tespit eden yeni bir teknoloji geliştirilmektedirler [1]. Sürüş güvenliğini artırıcı akıllı araç uygulamalarının taşıtlarda kullanımında, son senelerde dikkate değer bir artış olmuştur.

NHTSA (ABD Ulusal Otoyol Trafik Güvenliği Yönetimi) araştırmasına göre Amerika 1.550'si ölümcül olan 40.000 yaralamalı, 56.000 trafik kazasının sürücü uykululuğu kaynaklandığı belirtilmiştir [2]. Ayrıca Amerika'da yetişkin sürücülerin %60 son bir yıl içinde uykulu iken araç sürdüğü ve %37'sinin araç sürerken uyuduğu bildirilmiştir [3].

Otomatik gerçek zamanlı DDDS'nin araçlarda kullanılmasıyla büyük yıkımlara sebep olan trafik kazaları azaltılabilir. Uykulu Sürüş Algılama Sistemi kaçınılmaz olarak ilerde araçların çoğunda kullanılmaya başlanacaktır.

DDDS gerçekleştirmenin en büyük zorluğu çok değişken bir ortam üzerinde çalışılması zorunluluğudur. Tasarlanacak DDDS; İnsan alışkanlıklarından kaynaklanan davranış çeşitliliği, insan yüzündeki takı-aksesuar ve makyaj çeşitliliği, yol durumundan kaynaklanan titreşimler, araba içinde sürücünün hareketli olması, araba içinin dinamik olması, mevsimlere göre çevreden oluşan değişiklikler, gece-gündüz ortam farkları, diğer araçlardan sebep olduğu ani ışık değişimleri gibi gürültülerden etkilenecektir. Bu sistem tüm değişken durumlarda çalışabilecek, ucuz donanıma sahip ve gerçek zamanda koşabilecek gürbüz bir yazılımla gerçekleştirilinceye kadar ticari firmaların ve araştırmacıların dikkatini çekmeye devam edecektir.

Gerçek zamanlı DDDS'yi gerçekleştirmenin zorluklarına rağmen birçok uygulamaya göre bazı üstünlükleri de vardır. Bu avantajları şöyle sıralayabiliriz;

- Kamera ile sürücü arasında en az ve en çok mesafenin kestirilebilmesi.
- Sürücü-kamera pozisyonunun sınırlana bilmesi.

- Sistemin çalışma şartlarında sürücüyü tanımanın sisteme ek bir yük getirmemesi.
- Bir kişinin sürekli (uzun bir süre) aynı arabayı kullanacağı varsayıldığında sürücü alışkanlıklarının sisteme kolayca dâhil edilebilmesi.

Gerçek çevre koşulları ve ucuz donanıma sahip sistemlere yönelik çalışmalar artmadığı sürece DDDS'nin yaygınlaşması mümkün olmayacaktır. Bu yüzden tasarlanacak sistemlerin, Matlab (matrix laboratory) gibi programlar yerine basit donanıma sahip gömülü sistemler üzerinde koşabilecek programlar olması tercih edilmelidir. Bu tez çalışmasında BSD (Berkeley Software Distribution) lisanslı OpenCV (Open Source Computer Vision) açık kaynak kodlu kütüphanesi kullanılmıştır. OpenCV gerçek zamanlı bilgisayarlı görme için en yaygın kütüphanedir. Ayrıca platform bağımsız olması yanında C ve C++ programlama dili temellidir. Bu yüzden gerçekleştirilecek uygulamalar basit donanıma sahip gömülü bilgisayarlarda bile icra edilmesi mümkün olabilecektir.

1.1. Uyku ve Trafik Kazaları

Trafik, yol, araç ve insan üçlüsünden oluşmakta, bunlardan herhangi birinde oluşan bozukluk trafik kazalarına neden olmaktadır [4]. NHTSA'a göre ölümcül, tek araçlı ve yoldan çıkarak oluşan kazaların en önemli sebebi sürücü performansını etkileyen uykululuk hali, alkollü olma ve hızlı sürüş sebep olmaktadır [5]. Buna ek olarak ölümlü kazaların %3,6'sı uykulu ve yorgun sürücülerle ilişkilidir [6]. Ülkemizde trafik kazalarının ana sebebi sanılanın aksine alkollü araç kullanımı değil yorgun sürücüler teşkil etmektedir. Avrupa da sürücü kaynaklı kazalarda; Uykulu sürücülerden kaynaklanan kazaların oranı %1-16 iken ülkemizde bu oran %70.02 [7] civarındadır. TÜİK (Türkiye İstatistik Kurumu)'in verilerine göre ölümlü, yaralı ve maddi hasarlı kazalarda sürücü, yolcu, yol, taşıt ve yaya kusur oranında, sürücü kusuru %94,5'dir [8].

Ülkemizde 2009-2011 yılları arasında taşıt cinslerine göre trafik kazasına karışan taşıtların kayıtlı taşıt içindeki oranı şöyledir; Otobüs %2,5-2,7, minibüs %1,5-1,6, kamyonet ve otomobil %1,2-1,3, kamyon %1-1,2, motosiklet %0,9-1,1 ve diğer taşıtlar ise %0,5-0,8 arasındaki bir oranla kazaya karışmıştır [8]. Oranlardan da anlaşıldığı gibi yolcu taşıma araçları öncelikli risk grubuna girmektedir. Kamyonlarda kaza oranı düşük olmasına rağmen ağır taşıtlar olması sebebiyle karışıklı kazalarda büyük tahribata sebep

olmaktadır. Dolayısıyla DDDS'nin tüm taşıt türleri için kullanılması trafik kazalarını azaltmak için elzem olmaktadır.

1.2. Ülkemizde Uykulu Sürüş Bilançosu

TÜİK verilerine ülkemizde 2002-2011 arasında kayıtlara geçen trafik kazaları istatistiği ve kaza bilançosuna göre: ülkemizde her yıl ortalama 794.647 trafik kazası olmaktadır [8]. Bu kazalardan 96.416'sı ölümlü-yaralı kazaları teşkil etmektedir. Bu kazalarda ortalama her yıl 4.305 kişi ölmekte 171.870 kişi ise yaralanmaktadır. Bu sayılar olay yerinde ölenleri içermekte kazadan sonra hastaneye kaldırılıp orada yaşamını yitirenleri içermemektedir. Oysa DSÖ (Dünya Sağlık Örgütü) kazadan bir ay sonraya kadar ölümleri trafik kazası ölümü olarak kabul etmektedir. Bu nedenle trafik kazası kurbanı olan kişilerin daha fazla olduğu söylenebilir.

1.3. Uykulu Sürüş Kazalarında Risk Altında Olan Sürücüler

Uykululuğu tetikleyen fiziksel, çevresel ve genetik faktörler olmakla birlikte genel olarak uykusuzluk sebebi ile gerçekleşen kazalar açısından en çok risk altında olan sürücüler, şu başlıklar altında sınıflandırılmıştır [4].

1.3.1. Uykusu Bozulmuş veya Yorgun Sürücüler

Mola vermeksizin uzun süre araç kullananlar; Gece, öğleden sonra ve normalde uyuduğu saatlerde araç kullananlar; Uyku yapan ilaçlar ya da alkol alanlar; Tek başına araç kullananlar; Uzun, kırsal, sıkıcı yollarda araç kullananlar; Sık yolculuk edenler bu kategoriye girer [4].

İngiltere'de yapılan araştırmada uyku ilişkili kazaların saat 02.00, 06.00 ve 16.00 dolaylarında 3 pik yaptığı saptanmıştır [4]. TÜİK verilerine göre 2011 yılında ülkemizde trafik kazalarının en fazla saat 15.00-19.00 arası görülmüştür. Saat 08.00-09.00 arasında bir pik olmasına rağmen ülkemizde kazaların büyük çoğunluğu 15.00-19.00 saat aralığında meydana gelmiştir [8].

1.3.2. Genç Sürücüler

Uyku ilişkili kazaların geç kalma eğiliminde olan, az uyuyan ve gece araba kullanan gençlerde yaygın olduğu saptanmıştır. Kuzey Carolina'da bu tür kazaların %55'inin 25 yaş ya da daha genç kişilerce yapıldığı, sürücülerin %78'nin erkek olduğu görülmüştür. İngiltere ve Norveç'te yapılan çalışmalar da benzer sonuçlar elde edilmiştir [4].

1.3.3. Vardiyalı Çalışan Sürücüler

Bütün ülkelerde ister bazı meslek gruplarının (hemşirelik v.b.) doğal yapısından olsun isterse üretimi artırmak veya tam kapasite çalışmak için vardiyalı çalışma benimsenmiştir. Vardiyalı çalışanların %20-30'unun yorgunluk ilişkili araba kazası geçirdiği saptanmıştır. Özellikle gece vardiyasından eve dönüş tehlikeli olmaktadır [4].

1.3.4. Ticari Araç Sürücüleri

Özellikle kamyon sürücüleri yorgunluğa bağlı kazalara eğilimlidirler. Kamyon sürücülerinde uyku apnesi olarak adlandırılan uyku ve solunum bozukluğu yüksek oranda görülebilir. Tüm ağır kamyon kazalarının en az %30-40'ında sürücünün yorgunluğunun, katkıda bulunan etmenlerden biri olduğu ileri sürülmektedir [4].

1.3.5. Tanı Konulmamış Uyku Bozuklukları Olan Sürücüler

Uyku bozukluklarının kazaları arttıran bir risk etmeni olduğu bildirilmiştir. Kronik insomnia (uykusuzluk), uyku apnesi ve narkolepsi gibi aşırı gündüz uyuklamasına neden olan bozukluklar görülebilmektedir. Uyku bozukluğu olan pek çok kişi tanısız ve tedavisiz kalmaktadır. Örneğin; uyku apnesi orta yaşlı erkeklerin %4'ünde, aynı yaş grubundaki kadınların %2'sinde bulunmaktadır. Bu bozukluk kaza riskini 3-7 kat arttırmaktadır [4].

1.4. Uykulu Sürüşten Kaynaklanan Kazaların Belirtileri

Bir kazanın kesin sebebinin bilmek mümkün olmamakla birlikte uykulu sürüşten kaynaklanan kazaların genel belirtileri şu şekilde bildirilmiştir [4];

- Sürücünün alkol metre ve kan alkol düzeyinin yasal limitin altında olması.
- Aracın yoldan çıkmamış ya da başka bir aracın arkasına çarpmamış olması.
- Kaza olay yerinde fren izlerine olmaması.
- Araçta mekanik bir sorun olmaması.
- İyi hava koşulları ve görüşün açık olması.
- Hız ve öndeki araca çok yakın kullanmak gibi nedenlerin elenmesi.
- Olay yerine gelen polis memurlarının temel neden olarak uyuklamaktan kuşkulanması.
- Kazadan hemen birkaç saniye önce sürücünün çıkış noktasını ya da çarptığı aracı açıkça görmesi.

1.5. Uyuklama ve Yorgunluk Halinin Belirtileri

Büyük kayıplara neden trafik kazalarına engel olmak için sürücülerin kendi kendileri için bir otokontrol mekanizmaları benimsemeleri ve uyukluk/yorgunluk halinde araç sürmeyi bırakmaları trafik kazalarını azaltmak için çok önemlidir. Genel olarak uyukluk/yorgunluk halinde sürücülerde şu durumlar gözlemlenir [4]:

- Sürücü araba kullandığı son birkaç kilometreyi anımsamaz.
- Sürücü yolda sağa-sola sapıyorsa veya yol kenarında bulunan, sürücüyü hız ve yol sınırı açısından uyaran bariyerlere çarpar.
- Sürücü dalıyor ya da dikkatini toplayamaz.
- Sürücü gözlerini açık tutmakta zorlanır.
- Sürücü önündeki arabaya çok yakın kullanıyor ya da trafik işaretlerini kaçıır.
- Sürücü başını tutmakta zorlanır.
- Aracı ani duruş ve kalkışlarla sarsa-sarsa kullanır.

Bu belirtilerin biri veya birkaçının olması durumunda sürücü araç kullanırken uyuma tehlikesiyle karşılaşabilir.

Sürüş güvenliğini arttırmak için; Şerit takibi, araç takibi, kavşaklarda ilerleme, gidiş yolunu planlama sürücü yardımcı sistemlerinin kullanımı artırılmalıdır [9]. Bu sistemlerin getirdiği pek çok avantaj vardır. Deneysel olarak sürücü yardımcı sistemlerinin varlığını durumunda sürücülerin çok daha rahat hareket ettiklerini gösterilmiştir [10].

1.6. Mevcut Uykulu Sürüş Algılama Sistemi Çalışmaları

Mevcut DDDS çalışmaları genel olarak sürücünden (bilgisayarlı görü teknikleri v.b.), araçtan (gaz pedalı, direksiyondan v.b) veya fiziksel vücut özellikleri (kalp atımı, nabız atımı v.b) değerlendirilerek oluşturulan sistemler olarak 3'e ayrılabilir [1,11-13]. Ayrıca bu sistemler sürücüyü rahatsız edip gerçek hayatta kullanılması zor olan veya sürücüyü rahatsız etmeyen yöntemler olmak üzere ikiye de ayrılabilir [14]. Buna ek olarak tek parametreyle karar alabilen veya birden çok parametreyi modele dâhil eden çalışmalara rastlamak mümkündür. Ayrıca çoğu bilgisayar görmesi probleminde karşılaştığımız gerçek zamanda koşabilirlik problemini aşmak için mevcut model ve çalışmaların daha hızlı olarak icra edilebilmesi için modelde kestirim (örnek uzayını kısıtlama, ön kabuller v.b.) yapan ve FPGA (Field Programmable Gateway Array) gibi hızlı icra edilebilen donanım üzerinde geliştirilen çalışmalara rastlamak mümkündür.

1.6.1. Fiziksel Vücut Özelliklerinin İncelenmesi

Bu yöntemi kullanan çalışmaların büyük bir kısmı vücut sıcaklığı, nabız sayımı, kalp atım sayısı veya beyin elektriksel sinyal ölçüsü gibi [15,16] sürücü vücudunun verdiği fizyolojik sinyallere sistem gerçeklenmek istenir. Sürücü beyin elektriksel sinyalleri EEG (Electroencephalography) yöntemiyle ölçülür [12]. Yapılan çalışmalar uyanıklık düzeyinin azalmasıyla EEG sinyalindeki uyku ile ilişkili olan alfa ve beta bantlarında artışlar gözlemlenmiştir [17]. Bu yöntem üst düzey bir otomotiv firmasının ilgisini çekmiştir ve bu firma bu yönde ar-ge çalışmaları yapmasına rağmen EEG başlığı giymeyi gerektirdiği için sürücüyü rahatsız etmekte ve mevcut yardımcı aparatlarla pratikte uygulanması zordur.

1.6.2. Aracın Tepkilerinin Algılanması

Bu tekniklerde aracın üzerindeki gaz pedalı ve direksiyon gibi araç donanımına yerleştirilen duyargalardan gelen [18] sinyallerden sürüş tipi ve araç durumunu analiz eder. Bu teknik sürücüyü rahatsız etmemesi, çevre şartlarından çok az etkilenmesine rağmen gerçekte uygulanmasının zorlukları vardır. Yapılan bir çalışmada gaz pedalının kullanımında [19] sürücünden sürücüye farklılıklar olduğu belirtilmiştir. Bu yüzden bu teknikler bünyesinde birçok olumlu yönler bulundurmasına rağmen uygulamada sürücüye uyarlanması gerekebilir.

1.6.3. Sürücünün Verdiği Uyarının Algılanması

Bu tezinde kapsamına giren sürücünün verdiği uyarıları algılayan teknikler genel olarak uykululuk esnasında meydana gelen ve yüzle ilgili hareket ve görünüm değişikliklerini tanıyabilen bilgisayarlı görü sistemlerine odaklanılmıştır [1,11-13,20-23]. Bilgisayarlı görü teknikleri uykulu sürüş belirlemede sürücüyü rahatsız etmeyen bir tekniktir. Bu tip sistemlerde sürücü yorgunluğu seviyesi tespiti için arabaya yerleştiren bir CCD (Charge Coupled Device) kameradan sürücü görüntüsü alınır. Alınan sürücü görüntüsünden bir veya birkaç parametre değerlendirilerek sürücü yorgunluk seviyesi için çıktı üretilir. Tek parametrelili ve çok parametrelili sistemlerde sürücü gözünden gelen görüntü bilgisi kullanılmasına rağmen en çok kullanılan parametre PERCLOS (Percentage of eye closure)'dur [14]. PERCLOS, göz kapanma oranı ve belli bir sürede gözün %80 kapalı olduğu zaman oranını ifade eden bir parametredir.

Bu tekniklerde genellikle bilgisayarlı görü teknikleri kullanılmasına rağmen farklı tekniklerde kullanılabilir. Örneğin [24]'de kızıl ötesi algılayıcıdan gelen baş hareket bilgileri değerlendirilerek uykululuk hali tespit edilmeye çalışılmıştır. [24]'deki yöntem probleme farklı bir bakış açısı getirmesine rağmen 2 boyutta çalışıldığı için sistemin gerçekleşme doğruluğu doğal olarak çok düşüktür.

Bilgisayarlı görü tekniklerinin en büyük sorunu gerçek test verileri elde etmenin imkânsıza yakın olmasıdır. Literatürde iki tip veri tabanı kullanan çalışmalara rastlamak mümkündür. Birincisinde göz kapalılığı veya göz kırpması veri tabanları kullanılır. İkincisinde deneğe araba kullanma otomasyonu ile araba kullanılarak elde edilen veriler

kullanılır. Bu tip çalışmaların güvenilirliği tartışmalıdır. Çünkü gerçek hayatta risk faktörü insan tepkilerini ve davranışlarını etkileyen önemli bir faktördür. Fakat bu otomasyondaki denekler böyle bir risk faktörüyle karşı karşıya değildir.

Uykulu sürüş belirlemede en güvenilir parametrenin PERCLOS olduğu [25] belirtilmiştir. [1]'de OpenCV kullanılarak yüz bulunmuş akabinde gözler bulunmuş ve göz kırpma sayısı uykululuk için referans alınmıştır. [2]'de insan ten rengi algılamaya en müsait dönüşüm olan YCbCr renk uzayı dönüşümü kullanılarak sistem gerçekleştirilmiştir. [3]'de mevcut çalışmalara ek olarak YCbCr renk dönüşümünden faydalanılarak göz bulunmuş, göz takibi için kalman filtresi kullanılmıştır. [4-6]'da göz kapanması ve baş hareketlerinin analizinden farklı olarak uykululuk halinde yüzle ilgili ifadeler ve hangi yüz ifadelerinin daha belirgin olduğunun tespiti üzerine çalışılmıştır. [5]'de uykuya dalmadan önceki anlarda sürücülerin daha çok değil daha az esnediğini ortaya koyulmuştur. [9]'da iris tanıma ile sürücüyü yetkilendiren, göz kapalılığı ve kızıl ötesi sensorlerle vücut sıcaklığı ölçülerek uykululuğun tespit edildiği melez bir çalışma sunulmuştur. [10]'da [9]'a benzer olarak bilgisayarlı görüyle gözden gelen görünüm bilgisi ve parmaklardan algılanan nabız atımı sayısı ile karar alabilen melez bir yapı önerilmiştir. [26]'de YSA (Yapay Sinir Ağları) ile sistemin doğruluğu ile katman sayısı ve nöron sayısı arasındaki ilişki incelenmiştir. [27]'de iki kızıl ötesi filtre kullanılarak gözden yansıyan ışık farkı ile göz tespitini yapan bir yapı önerilmiştir. [14]'de FPGA tabanlı, retinadan yansıyan ışığın yoğunluğunun farklı olacağı esasına dayanarak gözün bulunması ve göz kırpma oranına göre sonuç üreten bir yöntem sunulmuştur.

1.7. Kapsam

Bu tez de maddi ve can kayıplı trafik kazalarının düşürülmesine yardımcı olabilecek DDS'ler incelenerek gerçek zamanlı koşabilecek bir model sunulmuştur. Gerçek zamanlı olarak bilgisayarda koşacak bu çalışma içerdiği yenilik ve sonuçlarıyla bu yönde yapılacak çalışmalara öncü olabilecek bir çalışma olmuştur. Sistem temel olarak kamera ile alınan görüntüyü işleyerek sürücü durumu değerlendirilerek karar verecektir. Sistem dört adet modülden oluşmaktadır. Birinci modül, alınan görüntüden yüzün tespit edilmesi modülüdür. İkinci modül, tespit edilen yüzden kişinin tanınmasıdır. Üçüncü modül, tespit edilen yüzde gözlerin tespit edilmesi modülüdür. Son modül ise göz kırpma

oranı ve sürücünün yolu bakıp bakmamasına göre sonuç üreten DDDS karar modülüdür. Mevcut modelin dört modülü yöntem ve algoritmalarıyla ikinci bölümde sunulmuştur. Üçüncü bölümde geliştirilen DDDS uygulması sunulmuştur. Son bölümde dört modül için elde edilen test sonuçları verilmiş olup bu test sonuçları üzerinde değerlendirme yapılmıştır.

2. MATERYAL ve METOT

2.1. Biyometrik Sistemler ve Akıllı Araç Uygulamaları

Biyometrik sistemler, insanların fiziksel vücut özelliklerini giriş verisi olarak alıp kişinin kimliği hakkında çıkış verisi üreten bilgisayar kontrollü, otomatik sistemleri ifade etmek için kullanılır. Çok eskiden beri suçlu tespiti ve yetkilendirme gibi alanlarda kullanılmalarına rağmen son yıllarda kişisel bilgisayar, otomotiv, personel giriş-çıkış kontrolü gibi değişik sektörlerde kullanımında artış olmuştur. Genel olarak kişi doğrulama ve yetkilendirme sistemlerinde, kullanıcıları tanımlamak yerine kullanıcının sunduğu kimlik kartı veya giriş kartı gibi bilgi içericiye onay verilmektedir. Uygulamada bu güvenlik açığına sebep olmakla birlikte kullanıcının kendisini tanımlayan veya yetkilendiren bir tanıtıcı taşıma zorunluluğu kullanıcıya külfet verebilir. Bu yüzden biyometrik sistemler, güvenlik ve konfor açısından daha cazip olmaktadır. Yüz tanıma, parmak izi tanıma, el damar ağı tanıma, retina tanıma ve iris tanıma yaygın olarak kullanılan biyometrik sistemlere örnek olarak verilebilir.

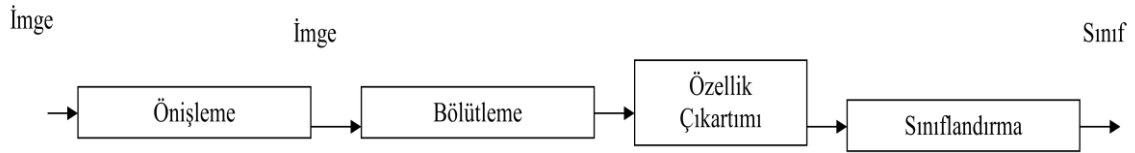
Eskiden beri insan özelliklerini biyometrik sistemlerde kullanan araştırmacılar bilgisayar teknolojisindeki gelişmelere paralel olarak son yıllarda insan fiziksel özellikleri, biyolojik vücut verileri ve insan davranışlarını giriş verisi olarak ele alan ilgi çekici çalışmalar yapmıştır. DDDS bu tip uygulamalara örnek olarak verilebilir [1,11-13,16,17,20-23,25,28,29].

Akıllı araç uygulamaları kaba olarak 2'ye ayrılabilir. Birinci tip sistemler: sürücünün parmak izini veya yüzünü tanıyıp yetkilendiren sistemlerdir. İkinci tip sistemler: DDDS, yakınlık algılama teknolojisi, yaya koruma ve çarpışmadan kaçınma gibi sürücü, taşıt ve çevreden veri alarak gerçekleşen sistemlerdir. Bazen problemin doğasından bazen de sistemi gerçeklemek için alternatif yöntemler olmasından dolayı uygulamada çok değişik teknoloji veya modelle bu tip sistemler gerçekleşir. Bu tezinde kapsamına giren DDDS ve birçok çalışmada bilgisayarlı görü teknikleriyle sistem gerçekleştirilmesine odaklanılır.

2.2. Bilgisayarlı Görme

Bilgisayarlı görme: algılanan görüntüye dayalı olarak gerçek fiziksel nesne ve sahneler hakkında kullanışlı kararlar oluşturmaktır [30]. İnsanların kolayca gerçekleştirdiği görme eyleminin makineler tarafından gerçekleştirilmesi çok zordur. Bilgisayarlar resmin değişik açılardan elde edilmiş görüntülerini kullanarak 3 boyutlu olarak yeniden oluşturmak mümkünken bilgisayarlar, mevcut bir resimdeki obje ve hayvanları tanımada iki yaşındaki bir çocuk kadar başarılı değillerdir [31].

Makinelere sahneleri anlama yeteneği kazandırma, insan görmesini anlama ve yardımcı olma ve görsel işlemleri otomatığe bağlamak için bilgisayar görmesi disiplininde birçok çalışma yapılmaktadır. Bilgisayarların işlem kabiliyetlerinin artmasına paralel olarak bilgisayar görmesiyle gerçekleştirilen uygulamalar artmaktadır. Örneğin tıbbi imge analizi, güvenlik (biyometri, gözetim, iz sürme, hedef tanıma), uzaktan algılama, robotik, endüstriyel denetim-kalite kontrol, doküman analizi, yardımcı yaşam ve insan-bilgisayar etkileşimi gibi değişik alanlarda yapılan çalışmalar bulunmaktadır. Tipik bir bilgisayarlı görü sistemi Şekil 2.1’ de verilmiştir.



Şekil 2. 1. Bilgisayarlı görü sistemi

2.2.1. Görüntü Yakalama

Nesne üzerinden yansıyan ışığın kameradaki optik-elektrik mekanizma ile elektriksel sinyallere dönüştürülmesiyle görüntüler sayısallaştırılarak bilgisayar ortamına alınır. Bilgisayarlı görmede resim veya resim dizileri üzerinde işlem yapılır. Çoğu sistemde tek resimden sistem için çıkış üretilir. Bazı uygulamalarda ise videodaki belli bir zaman aralığı veya kameradaki anlık veriyle belli bir zamana kadar olan geçmiş veri birlikte değerlendirilerek sistem için çıkış üretilir.

2.2.2. Önışleme

Görüntüde istenmeyen her türlü bilgiye gürültü denir. Sistemin doğru sonuç vermesi için gürültülerin bu aşamada giderilmesi hedeflenir. Buna ek olarak giriş verisi üzerinde bazı düzenlemeler yapılabilir. Örneğin bu aşamada görüntü histogram yayarak belirginleştirilir veya görüntü üzerindeki yapısal bozuklukları yok edilmeye çalışılır. Kısaca mevcut veri, beklenen veriye yaklaştırılır.

2.2.3. Bölütleme

Görüntü bölütleme, imgede benzer görsel karakteristikleri paylaşan piksellerin aynı gruba alınma işlemedir. Benzer olarak görüntüde elde edilmek istenilen veri olarak da tanımlayabiliriz. Örneğin plaka tanıma sistemi için plaka bölgesindeki her karakteri oluşturan piksellerin aynı gruba alınması işlemidir. Yani karakterlerin birbirinden ve plaka arka planından ayrıştırılması işlemidir.

2.2.4. Özellik Çıkartma

Kısaca sınıflandırma için ayırt edici özelliklerin belirlenmesi aşamasıdır. Sistemin doğruluğunu direk etkilediği için bu aşamadaki yöntem ve algoritmaların doğru seçimi sistem doğruluğu ve performansı açısından önemlidir. Günümüzde çoğu uygulamada ham veriler yerine veriden elde edilmiş özelliklerle sonuca gidilir.

2.2.5. Sınıflandırma

Sistem modelinin çıktısı sınıflandırma aşamasındadır. Literatürde doğrusal ve doğrusal olmayan birçok sınıflandırıcı önerilmiştir. Uygun sınıfın seçilmesi sistem performansı açısından çok önemlidir.

Bu tezin uygulaması OpenCV kütüphanesiyle gerçekleştirildiği için bundan sonra kısaca OpenCV'ye giriş yapılacaktır.

2.3. OpenCV Kütüphanesi

OpenCV, açık kaynak kodlu bir bilgisayarlı görü kütüphanesidir. Bu kütüphane C ve C++ programlama dillerinde yazılmıştır. OpenCV kütüphanesi Linux, Windows ve Mac OS X ortamlarında icra edilebilir. Ayrıca bu kütüphane Python, Ruby, Matlab ve diğer dilleri için aktif geliştirme ara yüzüne sahiptir.

OpenCV daha çok gerçek zamanlı uygulamalar düşünülerek sayısal hesaplama verimliliği arttırmak için geliştirilmiştir. OpenCV optimize edilmiş C dilinde yazılmış olup çok çekirdekli işlemcileri destekler. Bu kütüphane Intel mimarisine sahip donanımda, birçok algoritmanın parçaları için alt seviyede optimizasyonu sağlayan IPP (Intel's Integrated Performance Primitives)'nin sistemde yüklü olması durumunda otomatik olarak kullanır.

OpenCV geliştiricilerin bilgisayarlı görü sistemlerini hızlı ve kolay bir şekilde tasarlamaları için geliştirilmiştir. OpenCV bilgisayarlı görüyle ilgili 500'e yakın fonksiyon içerir. Bilgisayarlı görme ve makine öğrenmenin bir biriyle olan sıkı ilişkisinden dolayı OpenCV, genel amaçlı bir MLL (Machine Learning Library) kütüphanesini içerir. Bu alt kütüphanede istatistiksel örüntü tanıma ve kümelendirme (clustering) odaklanılmıştır. MLL daha çok görme işlemi için tasarlanmasına rağmen herhangi bir makine öğrenmesi probleminde de kullanılabilir [32].

OpenCV modüller bir yapıya sahiptir. Yani paylaşılan veya statik kütüphaneleri içeren parçalardan oluşur. Güncel OpenCV 2.4 versiyonu için OpenCV modülleri kısaca özetlenecektir.

2.3.1. Core Modülü

Diğer modüllerin kullandığı temel veri yapıları, çok boyutlu diziler ve temel fonksiyonların tanımlandığı kompakt bir modüldür. OpenCV resim işleme ve bilgisayarlı görme için kolay kullanımlı veri yapıları içerir ve bu veri yapıları bu modülde tanımlanmıştır. Kural olmamakla birlikte temel olarak her OpenCV uygulamasına bu modül eklenilir. Çünkü OpenCV'nin gücü birazda bilgisayarlı görme için sunduğu basit veri yapılarıyla da ilişkilidir.

2.3.2. İmgproc Modülü

Görüntü işleme modülüdür. Bu modülde doğrusal ve doğrusal olmayan görüntü filtreleri, geometrik görüntü dönüşümleri (yeniden boyutlandırma, affine ve perspektif eğilme), renk uzayı dönüşümü, histogramlar v.b. gibi işlemler için fonksiyonların bulunduğu modüldür.

2.3.3. Video Modülü

Hareket tahmini, arka plan çıkarma ve nesne izleme algoritmalarını içeren bir video analiz modülüdür. Görüntü işleme ve bilgisayar görmesi alanlarında iyi bilinen problemlerde kullanılan bir modüldür.

2.3.4. Calib3d Modülü

Temel çoklu görünüm geometri algoritmaları, tek ve stereo kamera kalibrasyonu, nesne poz tahmini, stereo benzeşme algoritmaları ve 3D yeniden kurma unsurları içerir. Bilgisayarlı görmeye yeni yeni araştırmacıların ilgisini çeken 3D yöntemler sunar.

2.3.5. Features2d Modülü

Giriş verisi için ayırt edici özellik algılayıcıları, veri tanımlayıcıları ve tanımlayıcı eşleştiricileri içeren bir modüldür. Örneğin giriş verisindeki köşelerin algılanarak bunun sisteme giriş olarak verilmesi için kullanılabilir.

2.3.6. Objdetect Modülü

Nesneleri ve önceden tanımlanmış sınıfların örneklerini (yüzler, gözler, kupalar, insanlar, araba v.b. gibi) algılamak için kullanılan modüldür. Bu modül kullanılarak yaygın bir çok algoritma için OpenCV ile gerçekleştirilmiş uygulamalar mevcuttur.

2.3.7. Highgui Modülü

Kolay kullanım ara yüzüyle video çekimi, görüntü ve video kodekleri sağlamasında kullanılır. Ayrıca OpenCV platform bağımsız olduğu için kullanıcı ara yüzü tasarımı için de bu modül kullanılır. Bu şekilde programlama dili ve platform sorunlarına takılmadan ara yüz tasarlanabilir.

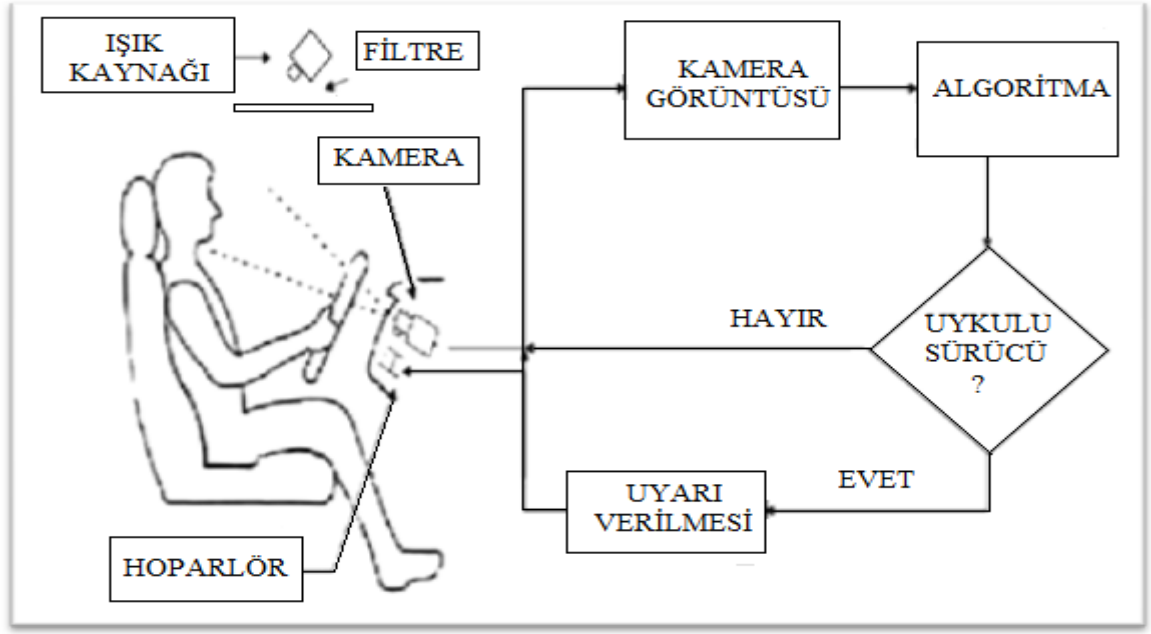
2.3.8. Gpu Modülü

Farklı OpenCV modüllerindeki GPU (Graphics Processing Unit) algoritmalarını hızlandıran bir modüldür. Temel olarak tüm modellerle iç içe veya onların etkinliğini arttırmayı hedefleyen bir modüldür.

OpenCV kütüphanesinde bu modüller haricinde, Flann ve Google test sarmalayıcılar ve Python bağlamaları gibi bazı yardımcı modüller de bulunmaktadır.

2.4. Uyku Sürüş Algılama Sistemi

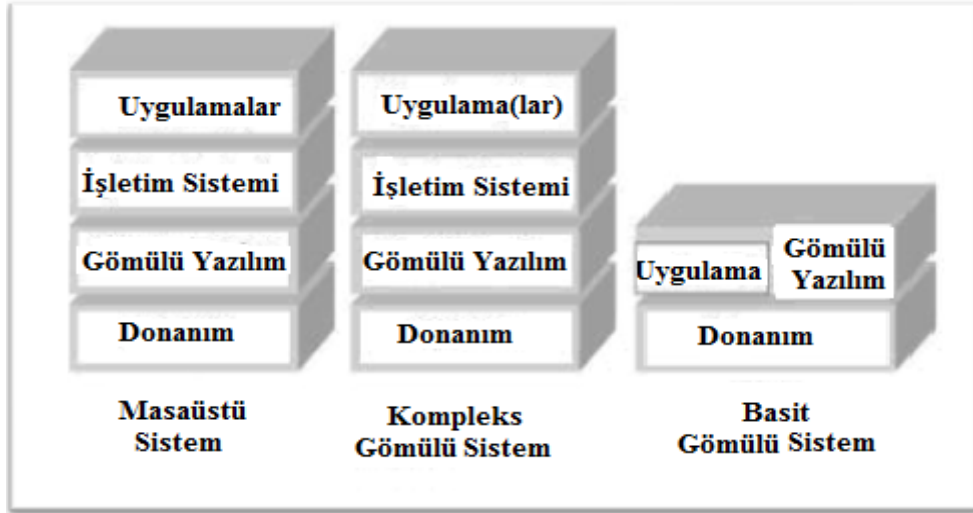
Giriş bölümde belirtildiği gibi DDDS akıllı araç sistemlerinin potansiyel uygulamalarından biridir. Bu sistemle sürücü yüzünün değerlendirilmesi sonucunda, sürücü tanınarak, sürücü yüz ifadesinden uyku olup olmadığı hakkında sonuç üretilir. Sürücü yardımcı sistemlerinden biri olan DDDS, sürücü kaynaklı trafik kazaları sebeplerinden biri olan yorgunluk/uykululuk hali için sürücü ikaz edilerek trafik kazalarını önlemeyi/azaltmayı hedefler. Sosyal ve ekonomik boyutu düşünüldüğünde geçmişte araştırmacıların dikkatini çektiği gibi halen araştırmacıların üzerinde yoğun çalıştığı ve gelecekte de çalışmaya devam edecekleri potansiyele sahip bir konudur. Bu sistem yazılım ve donanım olmak üzere iki bileşenden oluşacaktır. Şekil 2.2’de bu sistemin blok diyagramı verilmiştir. Bu sistemin hedeflenen kitleye ulaşabilmesi için ucuz maliyetli olması gerekir. Trafikteki mevcut DDDS’siz araçları düşündüğümüzde portatif olması gerekir. Bunlar gibi sebeplerden dolayı bu sistemin gömülü sistemler hedeflenerek tasarlanması daha doğru olacaktır. Temel olarak C ve C++ dilleriyle gerçekleştirilmiş sistemler gerçek zamanlı uygulamalar için daha uygun olmaktadır.



Şekil 2.2. Örnek bir DDDS diyagramı [20].

Günümüzde karmaşık gömülü bilgisayarlar masa üstü bilgisayarlar arasında fazla bir fark yoktur. Çoğu karmaşık gömülü bilgisayar sistemleri üzerinde bir işletim sistemi mevcuttur. Şekil 2.3’de masa üstü bilgisayar ile karmaşık ve basit gömülü bilgisayarlar için yazılım katmanları verilmiştir.

Şekil 2.3’de de görüldüğü gibi bir yazılım gerçekleştirirken basit gömülü sistemler için gereken özel ayarlamalar ve kısıtlamalar karmaşık gömülü bilgisayarlar için yoktur. Çünkü donanım üzerinde donanım ve yazılım arasındaki ilişkiyi düzenleyen işletim sistemi mevcuttur. OpenCV kütüphanesi C ve C++ tabanlı olduğu için bu kütüphane ile icra edilecek DDSS hem masa üstü hem de gömülü bilgisayarlarda yazılımsal olarak pek bir fark olamadan çok rahat koşabilecektir. Tüm bu sebeplerden anlaşılacağı gibi DDDS’nin ana problemi sistemin uygun bir modelle gerçekleştirildiği yazılımdır. Bu tez çalışmasında sunulan DDDS modeli yüz algılama, yüz tanıma, göz bulma ve göz durumu tespit olmak üzere 4 ana modülden oluşmaktadır.



Şekil 2.3. Bilgisayar sistemleri için yazılım katmanları [30].

2.4.1. Yüz Algılama

Giriş imgesinde yüz tanıma işleminin icra edilebilmesi için ilk önce yüzün tespit edilmesi gerekir. Yüz tespit aşaması, giriş imgesi olarak alınan imgede yüz olup-olmadığı ve varsa yüzün yerinin (konumunun) belirlenmesi aşamasıdır.

İnsan yüzleri, insanların kimliklerini ve duygularını temsil etmelerinden dolayı yüz tespit aşaması; Yüz tanıma, yüz ifadesi tespiti, insan-bilgisayar etkileşimi, cinsiyet/yaş belirleme, yüz doğrulama/yetkilendirme ve yüz modelleme gibi popüler uygulamaların ilk aşamasıdır. Dolayısıyla bu aşamanın doğru ve etkin bir şekilde icra edilmesi tasarlanacak sistemlerin doğruluk ve etkinliğini doğrudan etkileyecektir. Verilen imgede yüz olup olmadığı tespitini zorlaştıran sebepler şöyle özetlenebilir:

- Yüzü tanımlayacak genel bilgiler olmasına rağmen yaş, cinsiyet, çekik gözlülük, yüz şekli (yassı, oval v.b), ten rengi gibi insanoğlunun sahip olabileceği değişik özelliklerinden dolayı doğal olarak yüzler çok çeşitlidir.
- Kamera-yüz pozisyonuna (cepheden, yandan, yukarıdan v.b.) bağlı olarak resimde yüzler farklı şekillerde görünebilir veya pozisyona bağlı olarak göz kulak gibi yüz bileşenleri tamamen veya kısmen görünmeyebilir.
- Sakal, bıyık gibi yapısal bileşenler, gözlük ve şapka gibi aksesuarlar ve makyajın var olup olmaması yüz için şekli, renk ve boyut olarak geniş bir çeşitliliğe sebep olur.

- İnsan duygularının deęişkenlięi ve bu duyguların yüz görünümünde çeşitlilięe sebep olan farklı yüz ifadesi (gülme, kızma, şaşırma gibi) olarak ortaya çıkabilmesi.
- Görüntüde yüzün bir kısmının dięer yüz, eller gibi nesneler tarafından kapana bilmesi.
- Elde edilen resmin, kameranın optik eksenile yaptığı açıdan dolayı (cihazın eğik tutulması) resmin yönü deęişebilir.
- Görüntü elde edilirken ki ıřık (spektrum, kaynak daęılımı ve ıřık şiddeti) ve kamera özellikleri (sensor tepkisi, lensler) elde edilen resim üzerinde gürültüye sebep olabilir
- Gönümüzde kullanılan teknoloji ve modellemelerin; İşlem kapasitesi ve modelleme kabiliyetleri bakımından insanın sahip olduęu görme yetisine göre çok yetersiz olması.

2.4.1.1.Yüz Tespit Yöntemleri

Yüz tespiti için günümüze kadar birçok teknik geliştirilmiştir. Bazı uygulamalarda ham veri yerine dönüşüm yapılmış imge üstünde işlem yapılarak sonuca gidilmeye çalışılır. Bazı yöntemlerde ise imge yerine imgenin özellikleri sistemin giriş verisi olmaktadır. Sistem tasarlanırken tek bir teknik yerine bazı uygulamalarda karma tekniklerin önerildięi melez modellerle de uygulama gerçeklenmek istenir. Yüz tespiti çalışmaları, 2002 yılında Yang v.d. tarafından özetlenmiştir ve bu yöntemler 4 temel grupta ayrılmıştır [33]:

- Bilgi tabanlı yüz tespiti yöntemleri.
- Deęişmeyen özellik tabanlı yüz tespiti yöntemleri.
- Şablon eşleme tabanlı yüz tespiti yöntemleri.
- Görünüm tabanlı yüz tespiti yöntemleri.

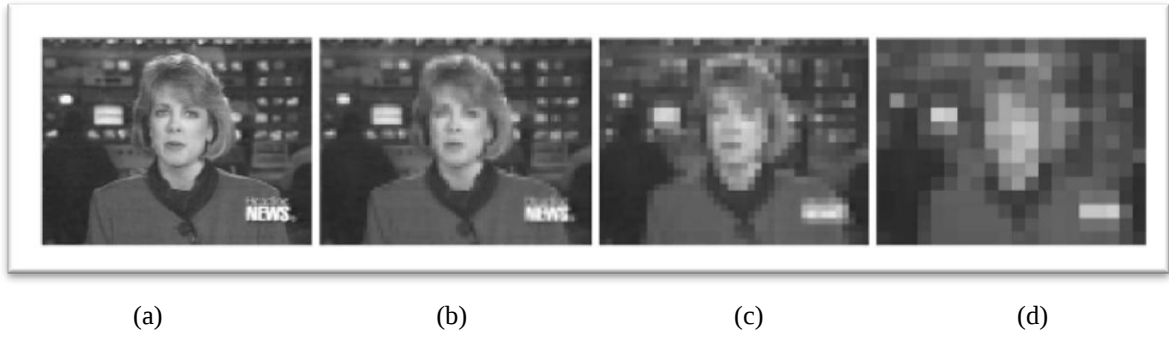
2000 yılından sonra iki yöntem etkinlięiyle öne çıkmıştır. Bunlar Viola-Jones'un 2001 yılında sunduęu sistem [34] ve Ojala v.d. tarafından 1994 yılında doku analizi için sunulan [35] LBP (Local Binary Pattern) yöntemidir. 2002 yılında Ojala v.d. tarafından sunulan [36] çalışmasıyla görüntü işleme ve bilgisayarlı görme alanında LBP yöntemine olan ilgiyi uyandırmıştır ve geniş bir alanda kullanılmasına vesile olmuştur. Bu tez çalışmasındaki DDDS uygulamasında yüz tespit Viola-Jones yöntemi gerçekleştirildięi için

ilerleyen kısımlarda ayrıntılı olarak anlatılacaktır. Bu kısımda yüz bulma yöntemleri hakkında kısa bilgi verilecektir. Daha ayrıntılı bilgi için literatür özetinin verilip yöntemlerin karşılaştırıldığı; 2000 yılına kadar olan çalışmaları içeren makale [33] ve 2010 yılına kadar ki çalışmaları içeren teknik rapora [37] bakılabilir.

2.4.1.1.1. Bilgi Tabanlı Yöntemler

Bilgi tabanlı yüz bulma yöntemleri insan yüzü için oluşturan kuralların giriş imgesi için değerlendirilmesiyle ilgilenilen bölgenin yüz olup olmadığına karar veren yöntemlerdir. Yüze ait kaş, burun ve kulak gibi yapısal bileşenler arasındaki özelliklerden yüzü ifade edecek kurallar oluşturulur. Örneğin tipik bir yüzde iki adet göz bir ağız ve bunlar arasında kalan bir burun bulunur. Uygulamada yüzü oluşturan bu bileşenlerin birbirlerine göre uzaklıkları ve pozisyonları kullanılarak yüzün ifade edilmesi mümkündür. Tanıma aşamasında ilkin bu yapısal parçalar bulunur daha sonra ise bu bileşenlerle göre oluşturulmuş kurallara göre sistemden sonuç üretilir. Bu yöntemde yüzü ifade eden kuralların kesin ve detaylı olması yüzü bulmayı zorlaştırırken genel kurallar tanımlanması durumunda ise yüz olmayan aday bölgeler yüz olarak etiketlenebilir. Bu yöntemde diğer bir problem de kamera-yüz pozisyonundan kaynaklanan resimdeki yüzü oluşturan bileşenler arasındaki göreceli bilgilerin değişebilmesidir [33].

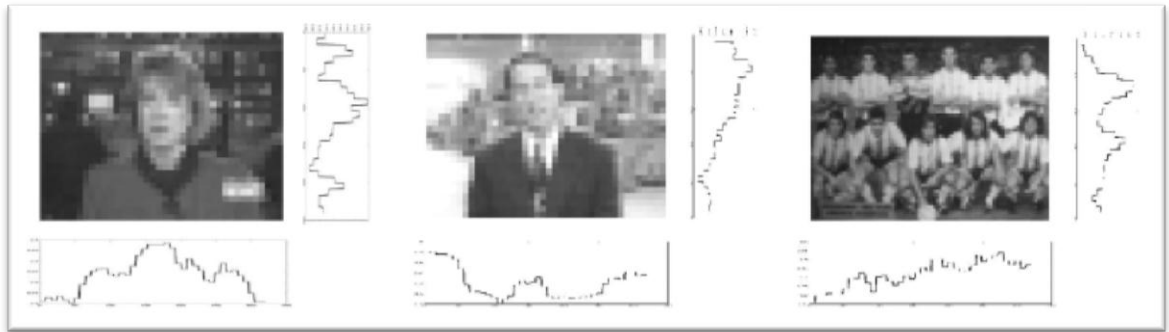
Kafa görünümüne göre tüm olası durumları kural tabanına almak pratikte mümkün olmamaktadır. Bu nedenle bu yöntem genelde ön cepheden çekilmiş ve karmaşık bir arka plana sahip olmayan resimler için daha başarılı sonuçlar vermektedir [33]. Belirtilen problemlerin bir kısmını aşmak için hiyerarşik kural tabanlı bir model [38] geliştirilmiştir. Bu yöntemde resmin birbirinden farklı düşük çözünürlüklü kopyaları elde edilmekte ve bunların her birine farklı kurallar uygulanmaktadır. Şekil 2.4’de düşük çözünürlüklü kopya elde etmek için $n \times n$ ’lik piksel komşuluğunun ((a) $n=1$, (b) $n=4$, (c) $n=8$ ve (d) $n=16$) ortalaması alınarak merkez pikselin değerini hesaplamaya dayanan yöntem kullanılmıştır.



Şekil 2.4. Örnek bir resim için düşük çözünürlüklü kopyaların elde edilmesi [33].

Bu yöntemde en düşük çözünürlüklü resimlere daha genel kurallar uygulanmakta ve bu kurallardan geçen aday bölgelerin aynı bölgelerin daha yüksek çözünürlüklü hali için daha ayrıntılı kurallar uygulanmaktadır. Bu elemeli yapıyla hesaplama maliyeti düşürülerek sistem performansı artırılmaya çalışılır.

Kontropoulos ve Pitas [38,39]'deki çalışmalara benzer olarak yüz konumu tespiti için kural tabanlı bir yöntem [40] önermişlerdir. Bu yöntemde ilk önce yüz konumunun, iz düşüm metoduyla tespit edilebileceğini gösteren çalışmaya [39] benzer olarak resmin yatay ve dikey iz düşümlerinin kullanılarak yüzün konumu belirlenir.



Şekil 2.5. Örnek bir resimlerin yatay/dikey iz düşümünün alınması [33].

Yatay ve dikey iz düşümü grafiğindeki tepe noktaları kullanılarak bir giriş resmi üzerinde yer alan tek bir yüz resminin bulunması mümkündür. Ancak Şekil 2.5'te görüldüğü gibi bu yaklaşım karmaşık arka planlarda ve birden fazla yüzün bulunduğu resimlerde birden çok yüzün veya yüzle benzer piksel yoğunluğuna sahip arka planın aynı iz düşüm konumuna gelmesi durumlarında bu yöntemin başarılı sonuçlar üretmesi zordur.

2.4.1.1.2. Özellik Tabanlı Yöntemler

Özellik tabanlı yöntemlerde kafa yönelimi ya da ışıklandırma gibi koşullar değişse bile kenar, renk tipi, renk yoğunluğu, desen ve şekil gibi insan yüzüne ait değişmez özellikler kullanılarak yüz belirlenmeye çalışılır. Bu yöntemlerde önce test resimlerindeki aday yüzler/özellikler bulunur ardından bunların gerçekten insan yüzü olup olmadığını tasarlanan bir modelle karar verilmeye çalışılır. Bu yöntemde algoritmaların başarısının aydınlatma, gürültü ve tespit edilecek yüzün bir kısmının başka bir nesne tarafından kapatılması gibi durumlarda veya karmaşık arka plana sahip imgelerde yüz bulma zorlaşır. Literatürde değişik yöntemleri kullanan birçok çalışmaya [20,41-43] rastlamak mümkündür.

2.4.1.1.3. Şablon Eşleme Tabanlı Yöntemler

Şablon eşleştirme yöntemlerinde genelde ön cepheden standart bir yüz modeli, elle ya da bir fonksiyon kullanılarak tanımlanmaktadır [33]. Verilen test resmi ile model arasındaki ilişki; Yüz hattı, gözler, ağız ve burun için ayrı ayrı hesaplanır. Benzerliğin belli oranın üstünde olması durumunda giriş imgesinde yüzün tespit edildiği varsayılır. Gerçekleştirmenin basit olması avantaj sağlamakla birlikte boyut, kafa yönelimi ve şekil değişimlerinden fazla etkilenmesi yöntemin önemli bir dezavantajıdır. Bu yöntem gerçek yüze yakın olarak yüz tanımlana bilmesine rağmen hızlı yüz bulmak için pek uygun değildir [31]. Literatürde yüz tanıma işlemi için şablon eşlemeye dayalı değişik çalışmalar mevcuttur. Örneğin çizgi çıkarmaya [44], adaptif modelleme yaklaşımıyla [45], istatistiksel özelliklerin algılanmasıyla [46], esnek görüntü modelleriyle [47], deforme edile bilen şablonlarla [48] ve şablonlar kullanarak [49] nesne/yüz tanıyan birçok çalışmaya rastlamak mümkündür.

2.4.1.1.4. Görünüm Tabanlı Yöntemler

Görünüm tabanlı yüz bulma yöntemleri çoğunlukla istatistiksel analiz ve makine öğrenmesi konuları kapsamında geliştirilmiş iyi bilinen tekniklerle görüntüde yüz olan ve

olmayan bölgelerin ayırt edici özelliklerini bulmaya dayanırlar. Tipik bir görünüm tabanlı yüz bulma süreci aşağıdaki temel adımları içermektedir [50]:

- Giriş resimlerinin ön işleme aşamasından (yeniden boyutlandırma, histogram eşitleme) geçirilmesi.
- Model için öğrenme ve test resimlerinin standart hale getirilmesi.
- Sınıflandırma modelinin pozitif ve negatif giriş verileriyle eğitilmesi.
- Yüzleri bulmak için bir karar verme stratejisinin uygulanması.

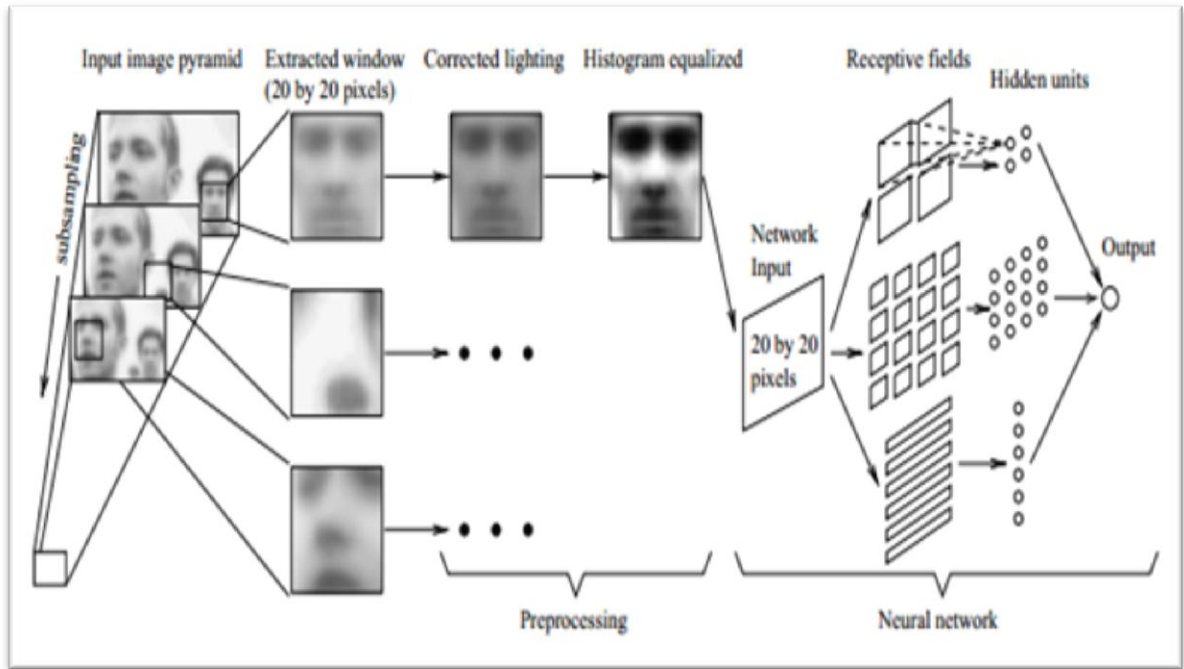
Görünüm tabanlı yöntemler kendi içlerinde eigenfaces (özyüzler), dağılım tabanlı yöntemler, sinir ağları, destek vektörü yöntemi, Bayes sınıflandırıcı ve saklı Markov modeli olarak gruplara ayrılabilir [33].

Öz vektörler kullanarak yüz tanıma uygulaması ilk olarak Kohonen [51] tarafından gerçekleştirilmiştir. Geliştirdiği basit sinir ağı, yüz görüntülerine ait oto korelasyon matrisinin öz vektörlerinden bir yüz tanımı çıkarmaktadır. Bu öz vektörler daha sonra öz yüzler olarak adlandırılmıştır [33]. Turk ve Pentland [52] ise PCA (Principal Component Analysis) yöntemi kullanarak yüz görüntülerinden oluşan bir eğitim setinden öz yüzleri oluşturmuştur. Oluşturulan öz yüzler, resim uzayının yüz uzayı olarak adlandırılan bir alt uzaya izdüşümüdür. Bu yöntemle yüz içeren görüntülerden elde edilen izdüşümler ile yüz içermeyen görüntülerden elde edilen izdüşümleri birbirinden farklı olmaktadır. Verilen giriş imgesinden yüz tespit yapmak için görüntünün alt pencereleri (aday yüzler) ile yüz uzayı arasındaki mesafeden yüz haritası çıkarılır. Bu haritanın en küçük değeri alt pencere (örtüşmenin en fazla olduğu aday yüz bölgesi) için bulunan yüz olarak etiketlenir. Bu yöntemde her ihtimalde bir sonuç üretilir. Bu yüzden yanlış algılamaya mahal vermemek için karar aşamasında sistem tahmini için belli bir eşik değerinde benzerlik şartı getirilmesi gerekir. Aksi takdirde hiç yüz içermeyen görüntüde bile pozitif sonuç üretilecektir. PCA yöntemi yüz tanıma yöntemleri içinde en çok atıf yapılan yöntemdir [53]. Bu tezin uygulamasında, yüz konumlandırma işlemi için kaskad sınıflandırıcı kullanılmış fakat yüz tanıma modülünde LDA yöntemi kullanılmıştır. LDA yönteminde de öz yüzler kullanılmaktadır.

Sung ve Poggio [50] yüz sezimi için görünüm tabanlı bir yöntem geliştirmişlerdir. Bu yöntemle yayılmış resim örüntülerinde bir sınıfın nesnesini, nesne sınıfının pozitif ve negatif örnekleriyle eğitilerek öğrenilebileceğini göstermişlerdir. Bu yöntem yüz ve yüz olmayan örüntüler için dağılım tabanlı model ve çok katmanlı perspetron sınıflandırıcı olmak üzere iki bileşenden oluşur. Çalışmalarında test aşamasında yüz olmadığı halde yüz

olarak algılanan giriş imgelerini toplayarak sisteme yüz olmayan örüntü olarak tekrar tanıtmışlardır. Bootstrap olarak adlandırılan bu yöntem daha sonraki çalışmalarda [34] kullanılmıştır.

Literatürde yüz tespiti ile ilgili birçok yapay sinir ağı mimarisi vardır. Yapay sinir ağları yüz tespiti için kullanılarak yüz örüntüsündeki görünüm çeşitliliği tanıyacak şekilde sistem eğitilebilir. Yüz tespiti için yapay sinir ağlarını kullanarak gerçekleştirilen birçok çalışma vardır. Rowley v.d. [54,55] çalışmaları en dikkat çekici olanlardandır. Rowley v.d. Şekil 2.6'da önerdiği yapı daha sonraki bir çok çalışmaya ilham kaynağı olmuştur. Bu sistem günümüzün en popüler yüz algılama yöntemi olan Viola-Jones'un önerdiği kaskad yüz algılama sistemiyle benzerlikler içerir.



Şekil 2.6. Yapay sinir ağları ile yüz algılama şeması [54].

Destek vektör makineleri yapay sinir ağları gibi öğrenme hatasını en aza indirmek yerine genel hatayı en aza indirmeye çalışırlar. Yani farklı sınıfların birbirinden ayırmak için sınıflar arasındaki ayırtıcı düzlemin sınıflara olan uzaklığını en büyük yapmaya çalışırlar. Destek vektör makineleriyle giriş resmi için dalgacık dönüşümü yapılarak yüzü algılayan çalışmalarda mevcuttur.

Bayes sınıflandırıcı kullanılarak yapılmış önemli çalışmalardan biri Schneiderman ve Kanade tarafından [56] gerçekleştirilmiştir [33]. Bayes sınıflandırıcı yüzün genel

şeklinden ziyade yerel görüntülerin birleşe bilirliliği ve yüz örüntüsünün pozisyonu için çoklu çözünürlükte gerçekleşmesine imkân verir.

Saklı Markov modeliyle ilgili birçok çalışma yapılmasına rağmen günümüzde artık tercih edilmeyen bir yöntemdir. Saklı Markov modeliyle ilgili ayrıntılı literatür özeti için [33] nolu kaynağa başvurulabilir.

Birçok yüz algılama yöntemi halen kullanılmasına rağmen görünüm tabanlı yöntemlerden olan yükseltme (boosting), Viola-Jones [34] tarafından gerçekleştirilen yöntem, son yılların en popüler yöntemidir [31]. Bu yöntem etkinliğiyle gerçek zamanlı uygulamalarda tercih sebebi olmaktadır. OpenCV’de yüz, ağız ve göz gibi nesneler için bu yöntemle eğitilmiş hazır modeller mevcuttur. Literatürde DDDS için Viola-Jones yöntemini kullanarak yüzü tespit eden çalışmalar [1,29] mevcuttur. Bu yöntem, dijital kameralar ve resim işleme uygulamaları gibi birçok alanda yüz tespiti için kullanılmaktadır. Viola-Jones yöntemi, Rowley v.d. [54] önerdiği sistemden 15 kat daha hızlı yüz tespit ettiği bildirilmiştir [34].

2002 yılında Ojala v.d. tarafından sunulan [36] çalışmayla görüntü işleme ve bilgisayarlı görme alanında LBP yöntemine olan ilgiyi uyandırmıştır. LBP yöntemi etkinliğiyle son yıllarda ilgi odağı olup gerçek zamanlı hızın önemli olduğu bazı çalışmalarda Viola-Jones yönteminde Haar özellikler yerine tercih edilmiştir. Viola-Jones yöntemi ve LBP yöntemi sonraki kısımda ayrıntılı açıklanacaktır.

2.4.1.2. OpenCV ve Yüz Tespiti

OpenCV’de nesne/yüz algılamak için yaygın olarak Viola-Jones tarafından tasarlanan kaskad sınıflandırıcı modeli kullanılmaktadır. Bu modellerde giriş verisi olarak Viola-Jones tarafından sunulan Haar benzeri özellikler ve LBP kodeleri kullanılmaktadır. Leinhart ve Maydt çalışmalarında Viola-Jones yönteminin kullandığı Haar benzeri özellikler için genişletilmiş Haar benzeri özellikler önermişlerdir [57]. OpenCV’de Viola-Jones’un sunduğu modelde Haar benzeri özellikler yerine bu genişletilmiş Haar benzeri özellikler kullanılmaktadır.

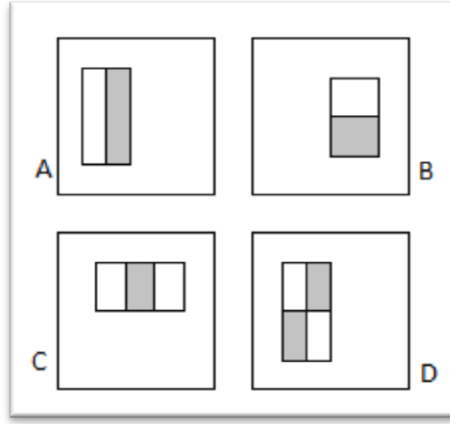
2.4.1.2.1. Viola-Jones Yüz Tespit Yöntemi

Viola-Jones yüz tespit sistemi, gerçek-zamanlı nesne tespiti için kendinden önceki en başarılı çalışmalardan olan Rowley v.d. önerdiği sistemden 15 kat daha hızlı yüzün tespit edildiği bir sistemdir. Birçok nesnenin tespiti için kullanılabilecek olan bu sistem yüz tespit problemi için yaygın olarak kullanılmaktadır. Gerçek zamanda çalışabilecek bir Viola-Jones yüz tespit sistemi dört temel bileşenden oluşur. Bunlar Haar benzeri özellikler olarak adlandırılan basit dörtgen özellikleri, basit dörtgen özelliklerin hızlı bir şekilde hesaplanmasına olanak veren integral imge, basit sınıflandırıcıların bir araya getirilerek güçlü bir sınıflandırıcı elde edildiği AdaBoost sınıflandırıcı ve hesaplamaları asıl ilgilenilen bölgeye yönlendiren elemeli kaskad yapısıdır.

2.4.1.2.1.1. Haar Benzeri Özellikler ve Genişletilmiş Haar Benzeri Özellikler

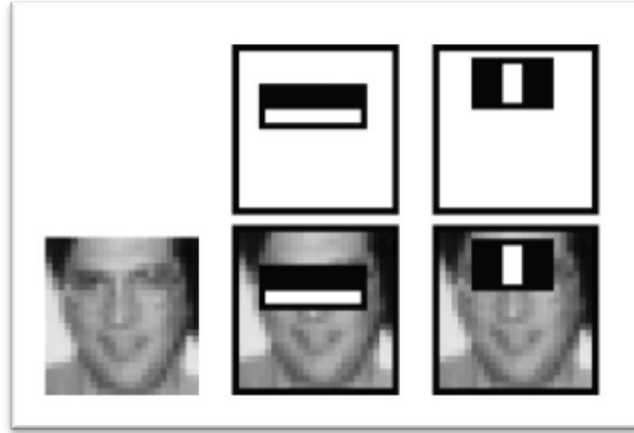
Viola-Jones nesne tespit modeli ham piksel verileri yerine sistem girişi olarak ismini Haar dalgacıklarından alan Haar benzeri özellikler kullanırlar. Ham veri yerine özellikleri kullanmak birçok uygulamada özellikle gerçek zamanlı uygulamalarda hız için tercih edilir. Haar fonksiyonlarını andıran basit özellikler ilk kez Papageorgiou v.d. [58] tarafından kullanılmıştır.

Viola-Jones nesne tespit çalışmalarında Şekil 2.7’de görüldüğü gibi 3 çeşit dörtgen özellik kullanmışlardır. Bu özelliklerin ilki, iki dikdörtgen bölgenin (Şekil 2.7 (a) ve (b)) piksel değerlerinin farkının değeridir. Bölgeler birbiriyle aynı uzunluk ve şekle sahip olup yatayda veya dikeyde birbirine bitişiktir. İkinci özellik üç dörtgen özelliğidir. Bu özellik için dıştaki dörtgenler (Şekil 2.7 (c)) toplanıp merkezdeki bölgenin piksel değerleri toplamından çıkarılarak elde edilir. Üçüncü dörtgen özelliği dört dörtgen özelliğidir. Bu özellik için çapraz dörtgenlerin (Şekil 2.7 (d)) piksel değerleri toplanıp birbirinden çıkarılarak elde edilir.



Şekil 2.7. Viola-Jones çalışması için özellikler

Ham giriş verisinden sonuç üretmek yerine giriş verisinin dönüşüm işlemlerinden geçirilerek özellik veya özellikler elde edilmesi bilgisayarlı görme alanında sık kullanılan bir yöntemdir. Viola-Jones yöntemindeki Haar benzeri özelliklerin ayırt edici özelliklerinin bilgisayarlı görme alanında kullanılabileceğini birçok çalışmada gösterilmiştir.

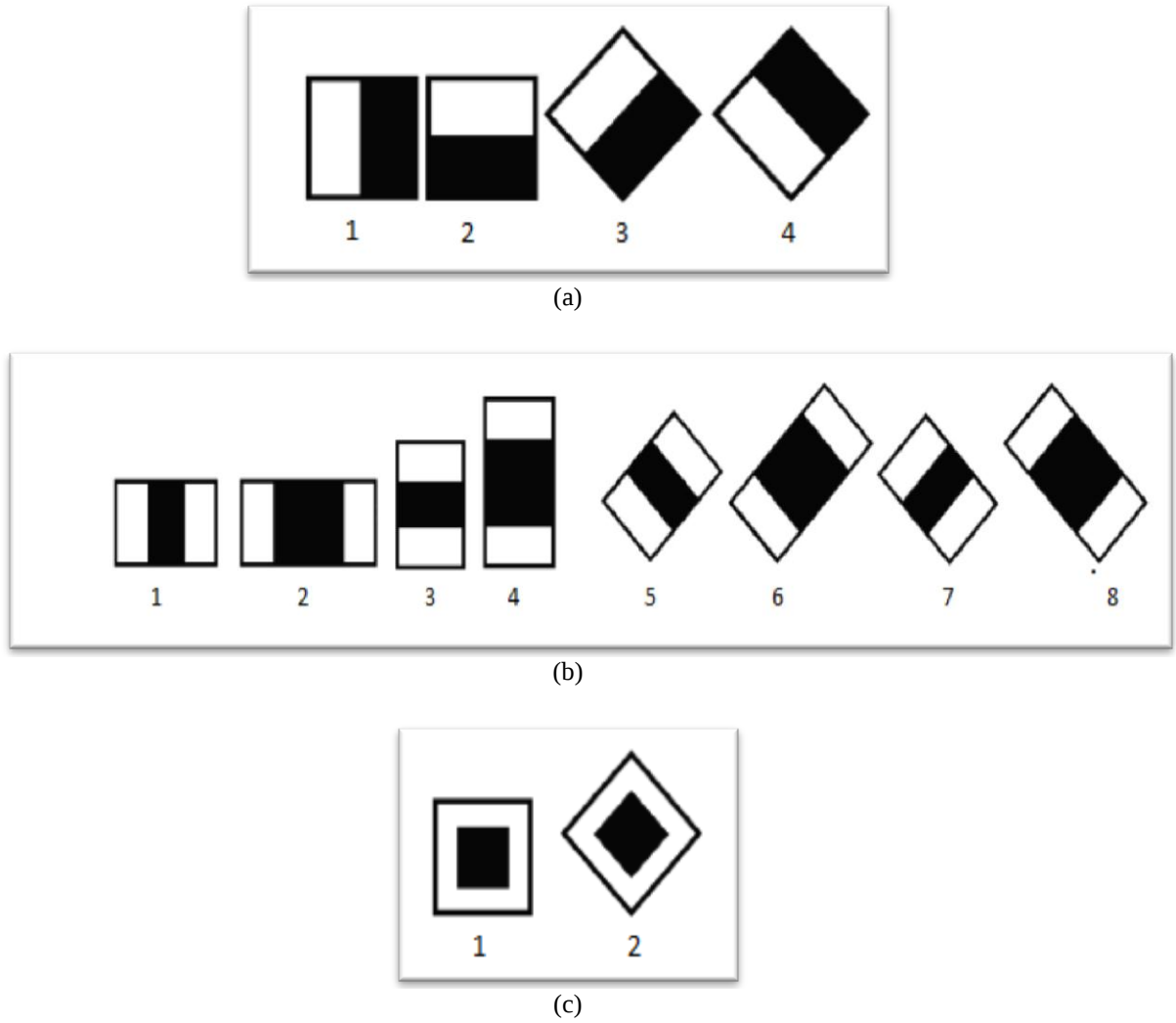


Şekil 2.8. Yüz tanıma için örnek Haar benzeri özellik elde edilmesi [34].

Viola-Jones çalışmalarında Şekil 2.8’de görüldüğü bu dörtgen pencerelerini resim üzerinde gezdirerek sistemim giriş verisi olan özellikleri elde etmişlerdir. Şekil 2.8’de Bu özelliklerden ilkinde gözler ve yanağın üst kısımları arasındaki yoğunluk farkı elde edilmiştir. Bu özellikten gözlerin yanağın üst kısmından sıklıkla daha karanlık olduğu belirlenmiştir. İkinci özellikten gözler ve burnun üst kısmı için yoğunluk farkı elde edilmiştir.

Geniřletilmiř Haar benzeri zellikler Leinhart ve Maydt [57] tarafından nerilmiřtir. İlgili alıřmalarında Haar benzeri zellikleri 45 derece dndrerek elde ettikleri yeni zelliklerle yz algılama sistemi iin hatalı algılamayı %10 azalttıklarını belirtmiřlerdir. Bu alıřmada ayrıca ykseltme algoritması iin yanlış algılamayı %12,5 yakın azaltan yeni bir yntem nermiřlerdir.

Leinhart ve Maydt geniřletilmiř Haar benzeri zelliklerinde drt tane kenar, sekiz tane izgi ve iki tane merkezi vreleyen zellik olmak zere 14 zellik bulunmaktadır. Viola-Jones nerdiėi drt drtgen zelliėi (řekil 2.7 (d)) bu alıřmada kullanılmamıřtır. řekil 2.9 (a)’da kenar zelliėi, (b) ‘de izgi zelliėi, (c)’de merkezi vreleyen drtgen zelliėi verilmiřtir.



řekil 2.9. Geniřletilmiř Haar benzeri zellikler iin kenar zellikleri (a), izgi zellikleri (b) ve merkezi vreleyen zellikler (c) [57].

OpenCV’de genişletilmiş Haar benzeri özellikler nesne tanıma için kullanılmaktadır. Haar özelliklerde olduğu gibi ilgili özellikler resim üzerinde gezdirilerek elde edilen siyah piksellere toplamının ortalamasından beyaz piksellerin toplamının ortalamasından çıkarılmasıyla elde edilir. Yüz tanıma için önceden eğitilmiş modeller hazır bulunabildiğinden dolayı birçok uygulamada Haar benzeri özellikler kullanılmaktadır. OpenCV kütüphanesi değişik nesnelerin Viola-Jones yöntemiyle kolayca tanınması için ara yüz sağlar. Hatta popüler bazı problem için hazır eğitilmiş sistem verilerine kolayca ulaşılabilir. Bu çalışmalar açık kaynak kodlu olduğu için akademik amaçlı kullanılmasında herhangi bir sakınca yoktur.

2.4.1.2.1.2. İntegral İmge

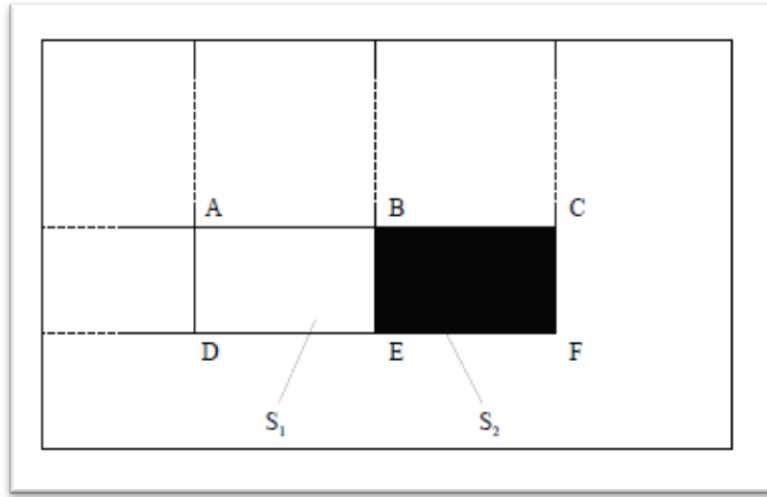
İntegral imge resmin bir başka ifade ediliş şeklidir. Şöyle ki resimdeki her piksel değeri kendisinin üstündeki ve solundaki tüm piksel değerlerinin toplamının değerini içerir. İntegral imge, Viola-Jones yönteminin giriş verisi olan dörtgen özelliklerinin hızlı bir şekilde icra edilmesine imkân verir.

Viola-Jones yönetiminde Haar benzeri özelliklerin hızlı bir şekilde hesaplaması için integral imge kullanılmıştır. İntegral imge resmin sol üst köşesinden başlanarak sağa doğru resim dolaşarak basit toplama işlemleriyle gerçekleştirir. İntegral imge denklem 2.1’de gösterildiği şekilde oluşturulur.

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y'), \quad (2.1)$$

Denklem 2.1’ de $ii(x, y)$ integral imge ve $i(x', y')$ değeri ise orijinal imgedir. İntegral imge bir kez icra edildikten sonra herhangi bir yerdeki ve herhangi bir büyüklükteki dörtgenin alanı çok kolay hesaplanabilir. Örneğin Şekil 2.10’daki dörtgen özellik $S_1 - S_2$ değeri olup, taralı alanı (S_2) basitçe hesaplamak için aşağıdaki denklem 2.2 kullanılır.

$$\sum_{(x,y) \in B_{CEf}} i(x, y) = ii(F) + ii(B) - ii(C) - ii(E) \quad (2.2)$$



Şekil 2.10. İntegral imge ile alt dörtgen alanı hesaplama

Denklem 2.2’de de görüldüğü gibi herhangi bir alt dörtgenin alanını hesaplamak için konumdan ve büyüklükten bağımsız olarak bir adet toplama ve iki adet çıkarma işlemi yeterli olmaktadır. İntegral imge bu özelliği ile gerçek zamanlı uygulamalarda işlem verimliliği için çok yararlı bir dönüşümdür.

2.4.1.2.1.3. AdaBoost Öğrenme

Viola-Jones yönteminde anlamlı Haar benzeri özellikleri ve Haar benzeri özellik için eşik değerleri belirlemek için AdaBoost (adaptif yükseltme) olarak adlandırılan makine öğrenmesi algoritmasını kullanır.

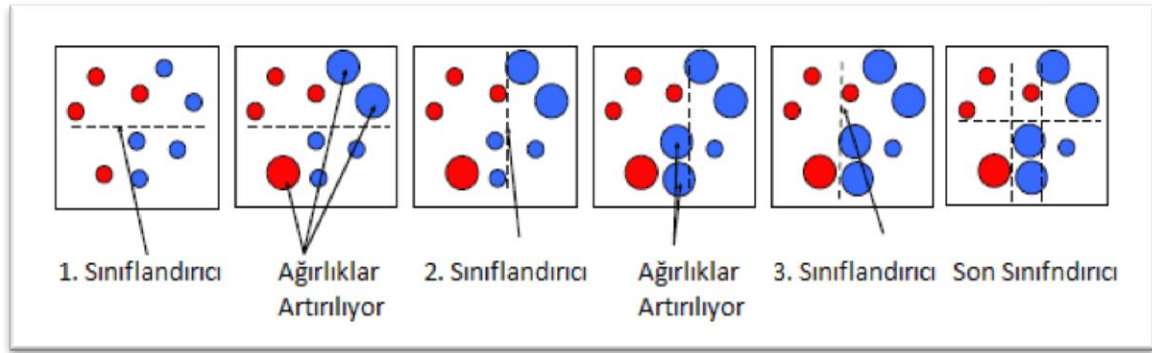
Yükseltme zayıf sınıflandırıcıların birleştirilerek güçlü bir sınıflandırıcı elde edilmesidir. Zayıf sınıflandırıcıdan kasıt işaret fonksiyonu gibi yalnız iki sonuç veya ona yakın sonuç üreten sınıflandırıcıdır. Pratikte bir uygulama için zayıf sınıflandırıcı pek bir anlam ifade etmezken sistem çıkışını çok az bile etkileyen bir sürü zayıf sınıflandırıcının doğrusal birleşiminden güçlü sınıflandırıcılar elde etmek mümkündür. Adaboost yönteminde sistem çıkışı için anlamlı zayıf sınıflandırıcılar seçilip onlara bir ağırlık atanır. Bu derecelendirilmiş zayıf sınıflandırıcıların birleşiminden güçlü yani karmaşık sistemler için bile doğru sonuç üretecek sınıflandırıcı elde edilir.

Viola-Jones yüz tespit çalışmasında dörtgen özellikler 180.000 kez çağırılmaktadır. Bu değer piksel sayısından fazladır. Dörtgen özellikler hızlı bir şekilde hesaplanmasına

rağmen tüm özelliklerin hesaplanması maliyetli olmaktadır. Viola-Jones çalışmalarında bu özelliklerin sadece bir kısmıyla iyi bir sınıflandırıcı oluşturulabileceğini göstermişlerdir. Bu anlamlı özelliklerin bulunması en önemli problemdir. Bu amaçla dörtgen özellikleri pozitif ve negatif olarak ayıran zayıf bir öğrenici tasarlamışlardır. Zayıf öğretici her özellik için en iyi eşik sınıflandırma fonksiyonunu karar verir. Birçok uygulamada zayıf öğrenici denklem 2.3’de verildiği gibi bir eşik fonksiyonudur.

$$h_j(x) = \begin{cases} a_j, & p_j f_j(x) < p_j \theta_j \\ b_j, & \text{Diğer durumlarda} \end{cases} \quad (2.3)$$

Geleneksel birçok uygulamada a_j ve b_j , ± 1 olarak alınır. $a_j = -s_j$ ve $b_j = +s_j$ olarak alındığında geriye f_j , θ_j ve p_j ’nin seçilmesi kalır. Viola-Jones çalışmalarında $(a_j, b_j) \in [0, 1]$ olarak alınmıştır.



Şekil 2.11. Yükseltme için şematik bir örnek [31].

Uygulamada tek sınıflandırıcı düşük hatayla icra edilir. Yükseltme zayıf öğrenicilerin toplamından bir sınıflandırıcı $h(x)$ oluşturulmasıdır. Yükseltme işlemi denklem 2.4’de gösterilmiştir. Şekil 2.11.’de yükseltme için şematik bir örnek verilmiştir.

$$h(x) = \text{sign} \left[\sum_{j=0}^{m-1} \alpha_j h_j(x) \right] \quad (2.4)$$

Her aşamadan sonra arttırarak zayıf öğrenicilerin seçilmesi ve öğrenme verilerinin yeniden ağırlıklandırılması yükseltme yönteminin başarısıdır. AdaBoost yöntemi bunu başarmak için her örneği bir fonksiyonmuş gibi her aşamada doğru sınıflandırılıp

sınıflandırılmamasına göre yeniden ağırlıklandırır ve her aşamadaki sınıflandırma hatası ortalamasıyla denklem 2.4’de verilen zayıf sınıflandırıcılar için α_j ‘nin en son ağırlığa karar verir. Uygulamada sistemin çıktısı çok hızlı iken, eğitim aşaması çok yüksek olmaktadır. Bazı uygulamalarda sistemi eğitmek haftalar almaktadır. Çünkü teorik olarak sistemin giriş verisi olan büyük ölçekli özelliklerin (dörtgen farklarının), sistemin her aşamasında test edilmesi gerekmektedir. AdaBoost algoritması için işlem adımları şu şekilde verilebilir:

- Eğitim aşaması için pozitif ve negatif örneklerin alınması. Eğitim verileri $\{(x_i, y_i)\}$ ile ifade edilir. $y_i = 1$ pozitif örnekleri, $y_i = -1$ ise negatif örnekleri gösterir.
- $w_{i,1} \leftarrow \frac{1}{N}$ için tüm ağırlıkların atanması. Burada N eğitim örneklerinin sayısıdır. Viola-Jones pozitif ve negatif örnekler olmak üzere ayrı N_1 ve N_2 kullanmışlardır.
- Her eğitim aşaması için $j = 1 \dots M$:
 - (a) Ağırlıklar toplamı 1 olacak şekilde yeniden normalize et (ağırlıkları, ağırlıklar toplamına böl).
 - (b) Ağırlıklandırılmış sınıflandırma hatasını en aza indiren sınıflandırıcıyı arayarak en iyi sınıflandırıcıyı $h_j(x; f_j, \theta_j, s_j)$ ’yi seç.

$$e_j = \sum_{i=0}^{N-1} w_{i,j} e_{i,j}, \quad (2.5)$$

$$e_{i,j} = 1 - \delta(y_i, h_j(x; f_j, \theta_j, s_j)) \quad (2.6)$$

Herhangi bir f_j fonksiyonu için optimum (θ_j, s_j) değeri ağırlıklı ortalama hesaplamının varyantı kullanılarak doğrusal olarak hesaplanır.

- (c) Değiştirilmiş hata oranı (β_j) ve sınıflandırıcı ağırlığını (α_j) hesapla,

$$\beta_j = \frac{e_j}{1 - e_j} \text{ ve } (\alpha_j = -\log \beta_j) \quad (2.7)$$

- (d) Sınıflandırma hatası ($e_{i,j}$)’ye göre ağırlıkları güncelle

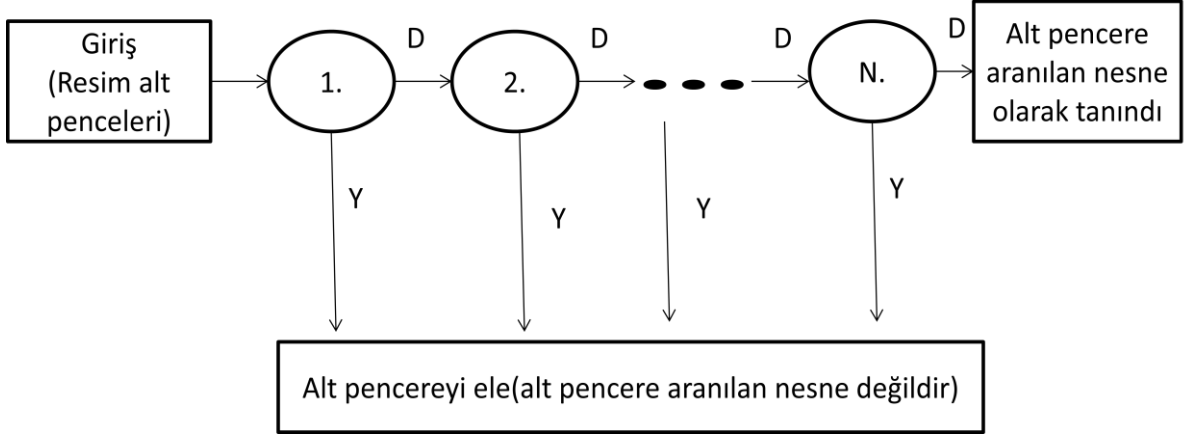
$$w_{i,1} \leftarrow w_{i,1} \beta_j^{1-e_{i,j}} \quad (2.8)$$

- Denklem 2.4.'e göre son sınıflandırıcıyı belirle.

İşlem adımları sayılan Adaboost algoritması genel Adaboost algoritması olup literatürde birbirine göre üstünlükleri olan değişik Adaboost algoritmaları bulunmaktadır [31].

2.4.1.2.1.4. Elemeli Kaskad Yapı

Elemeli kaskad yapı aşaması, Viola-Jones yönteminde sistemin efektifliğinin arttırmayı hedefleyen aşamadır. Bu aşamada sistem girişi olan ilgili resim kısımlarında arka planı hızlı bir şekilde eleyip hesaplamaları nesneye benzeyen bölgelere yani pozitif bölgelere yoğunlaştırmak hedeflenir. Negatif alt pencerelerin büyük bir kısmı karmaşık sınıflandırıcılara gelmeden kaskad yapının ilk aşamalarında (basit sınıflandırıcılarında) elenir. Bu işlem yani en baştan eleme işlemi verimliliği ve sistem doğruluğu olumlu yönde etkiler. Zayıf sınıflandırıcılar birleştirilerek AdaBoost sınıflandırıcılar, Adaboost sınıflandırıcılar birleştirilerek elemeli kaskad yapı oluşturulur. Kaskad yapının girişinden sonuç düğümüne doğru her düğümde zayıf sınıflandırıcıların sayısı genellikle artırılır bu şekilde sistemin sağlam kararlar alması hedeflenir. Şekil 2.12'de görüldüğü gibi girdi olarak verilen alt pencereler sırası ile ilk düğümden son düğüme doğru geçmektedir. Her düğümde alt pencerenin bir sonraki düğüme iletilip ileilmeyeceğine dair bir karar alınır. Bir sonraki düğüme bir önceki düğümün nesne olarak tanıdığı alt pencereler iletilir. Bunun bir sonucu olarak bir sonraki düğüm kendinden önceki sınıflandırıcıdan daha karmaşık bir görevi icra eder. Kaskad yapıda düğümlerde dereceli bir zorluk söz konusudur. Şöyle ki; ilk düğümlerde fazla alt pencere doğru kabul edilip sonraki aşamalarda zor bir sınıflandırılmadan geçirilir. Bu şekilde tüm düğümlerden geçen alt pencereler sistem tarafından tanınan nesne olurlar.



Şekil 2.12. Viola-Jones yöntemi için elemeli kaskad yapı diyagramı

Uygulamada bir nesnenin mesela yüzün eğitim aşamasında en geniş şekilde sisteme tanıtıldığı gibi yüz olmayan tüm nesnelerinde negatif örnek olarak sisteme tanıtılması gerekir. Bu uygulamada mümkün olmamaktadır. Viola-Jones bunu aşmak için Sung ve Poggio'nin çalışmalarında [50] önerdiği bootstrap olarak adlandırılan bir işlem uygulamışlardır. Bootstrap işleminde her düğüm için bir eşik değeri, mümkün olan en fazla negatif örneği pozitif örnek olarak algılayıp sonraki düğüme aktaracak şekilde elle belirlenir [37]. Bu şekilde her düğüm kendisinden önceki düğüm yokmuş gibi ayrı ayrı olarak eğitilir. Eğitim aşamasında bu şekilde yüz olmayan nesnelerin her katmanda algılanması hedeflenir.

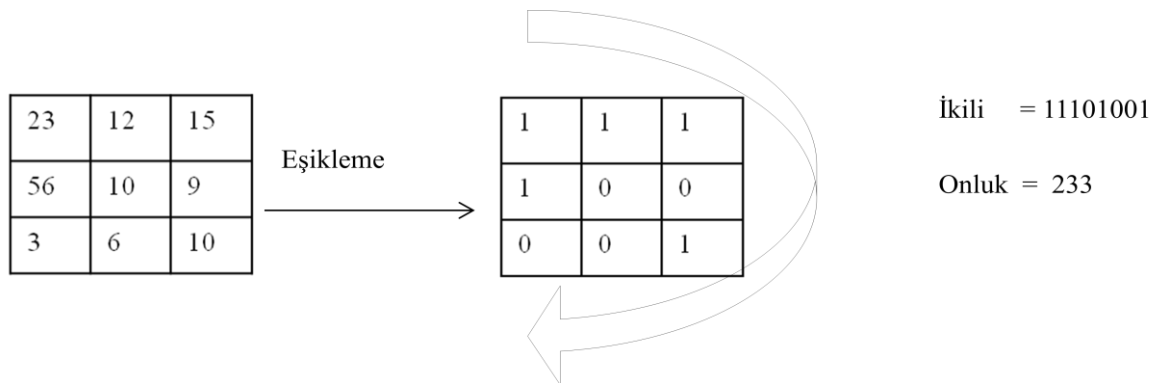
Viola-Jones yönteminde temel olarak çözülmesi gereken iki şey vardır. Birincisi uzun eğitim aşaması, diğeri kaskad parametrelerinin seçimidir. Lienhart v.d. [59] zayıf sınıflandırıcı olarak basit karar ağacı yerine CART (Classification And Regression decision Trees) kullanmışlardır. Çalışmalarında benzer süreler için daha iyi sonuç aldıklarını bildirmişlerdir. CART, ikili kırılım yaklaşımına sahiptir. Sınıflandırma ve regresyon yapabilir. Sayısal ve kategorik girdiler alabilir. Eksik verilerle sonuç üretebilirler. Ağaç budama yetenekleri desteklenmiştir. CART ikili kırılım yaklaşımıyla daha yüksek kestirim kapasitesine sahiptir ve yüksek kestirim kapasitesinin arandığı alanlarda daha çok tercih edilir.

OpenCV'de Viola-Jones yönteminde eğitim aşaması haftalarca vakit alabilen birçok uygulamalar için bazı nesneler (örneğin yüz ve gözler) önceden eğitilerek elde edilmiş hazır modeller mevcuttur. Bu veriler OpenCV kütüphanesinde "data" klasörünün içindedir.

2.4.1.2.2. LBP Yöntemi

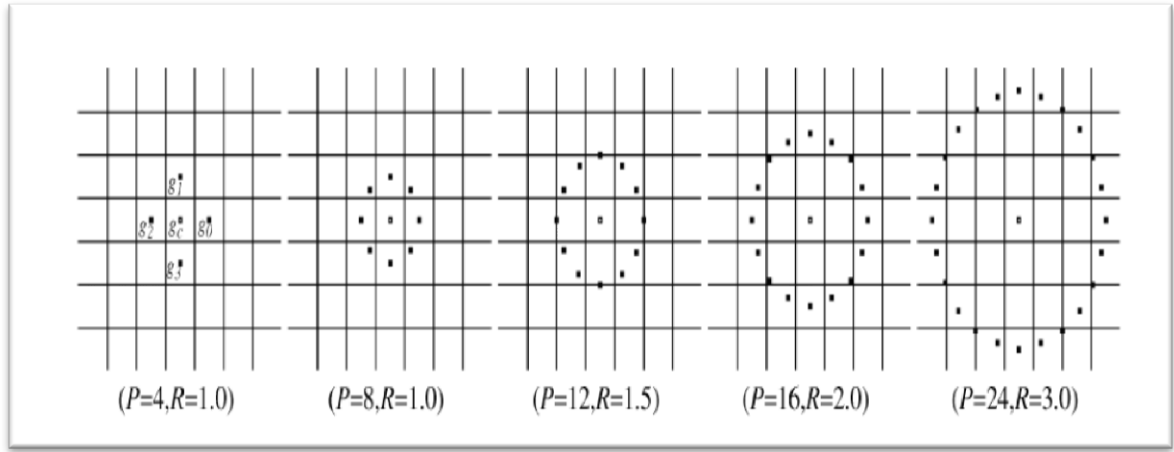
LBP resmin yerel yapılarını efektif olarak temsil eden parametresiz bir tanımlayıcıdır. Bu yöntem aslında doku analizi için sunulmuştur [60]. 2002 yılında Ojala v.d. tarafından sunulan [36] çalışmayla bilgisayarlı görme alanında LBP yöntemine olan ilgiyi uyandırmıştır ve geniş bir alanda kullanılmasına vesile olmuştur. LBP yöntemi etkinliğiyle son yıllarda ilgi odağı olmuştur. Temelde doku analizi için tasarlanmış olmasına rağmen günümüzde yüz analizi [61,62], resim ve video alma [63,64], çevre modelleme [65,66], görsel denetim [67,68], hareket analizi [69,70], biyomedikal-havadan görüntü analizi [71,72] ve uzaktan algılama [73] gibi değişik alanlarda kullanılır. Parametresiz bir yöntem olup, rotasyona ve ışıklandırma problemlerine karşı kararlı sonuçlar üretmesiyle öne çıkmıştır. LBP yönteminde gri seviyede işlemler yapılır. Giriş resmi renkli ise gri düzeye çevrilir daha sonra ise yerel komşuluk değerlendirilerek ilgili piksel değeri için LBP değeri belirlenir.

LBP yönteminin en önemli özelliği tek çeşit ışıklandırma değişikliklerini tolere etmesi ve gerçekleşmesinin basit olmasıdır. Orijinal LBP operatörü LBPs ve ya LBP kodekleri olarak adlandırılan ondalık sayıları, resmin piksel değerleri olarak atar. Şekil 2.13'te LBP operatörünün nasıl uygulandığı gösterilmiştir. Her piksel kendisinin 3x3 komşuluğundaki piksellerle ayrı ayrı olarak çıkarttırılır. Eğer sonuç negatif ise ikili değer olarak sıfır, eğer sonuç sıfır veya pozitif ise ikili değer olarak bir atanır. Daha sonra ise sol üst köşeden başlanıp saat yönünde gidilerek merkez pikselin komşuluğundaki 8 adet ikili veriden 8 bitlik ikili bir sayı elde edilir. Türetilen bu ikili sayıya LBPs veya LBP kodekleri denir. Bu ikili sayının onluk tabandaki karşılığı merkez pikselin değeri olarak atanır.



Şekil 2.13. Temel LBP operatörünün uygulandığı

Temel LBP operatörü 3x3 komşulukta çalıştırıldığından büyük ölçekli yapılarda belirgin özellikleri yakalayamaz. Değişik ölçekteki doku özellikleri için LBP değişik komşuluklarda uygulanacak şekilde Ojala v.d. tarafından sunulan çalışmayla [36] geliştirilmiştir. Yerel komşuluk, bir daire üzerinde örneklenmiş noktalar olarak tanımlanır. Merkezdeki piksel LBPs değeriyle etiketlenecek pikseldir. Örneklenmiş noktalar ise piksele denk gelmeyen noktalar için bilinear (çift doğrusal) interpolasyon yöntemi ile ara değer üretilir. Şekil 2.14’de genişletilmiş LBP (ELBP) operatörü örnekleri verilmiştir. (P,R) ifadesinde P, komşulukta örneklenen noktaları, R ise dairenin yarıçapını göstermektedir.



Şekil 2.14. ELBP operatörü için örnekler [36].

(x_c, y_c) olarak verilmiş bir pikselin LBP ondalık değeri denklem 2.11.’de gösterilmiştir. Bu denklemde i_c ve i_p gri düzeydeki piksel değerlerini ifade etmektedir. i_c merkez piksel değerini, i_p ise merkezi çevreleyen piksel değerlerini ifade etmektedir.

$$LBP_{P,R}(x_c, y_c) = \sum_{p=0}^{P-1} s(i_p - i_c) 2^p \quad (2.11)$$

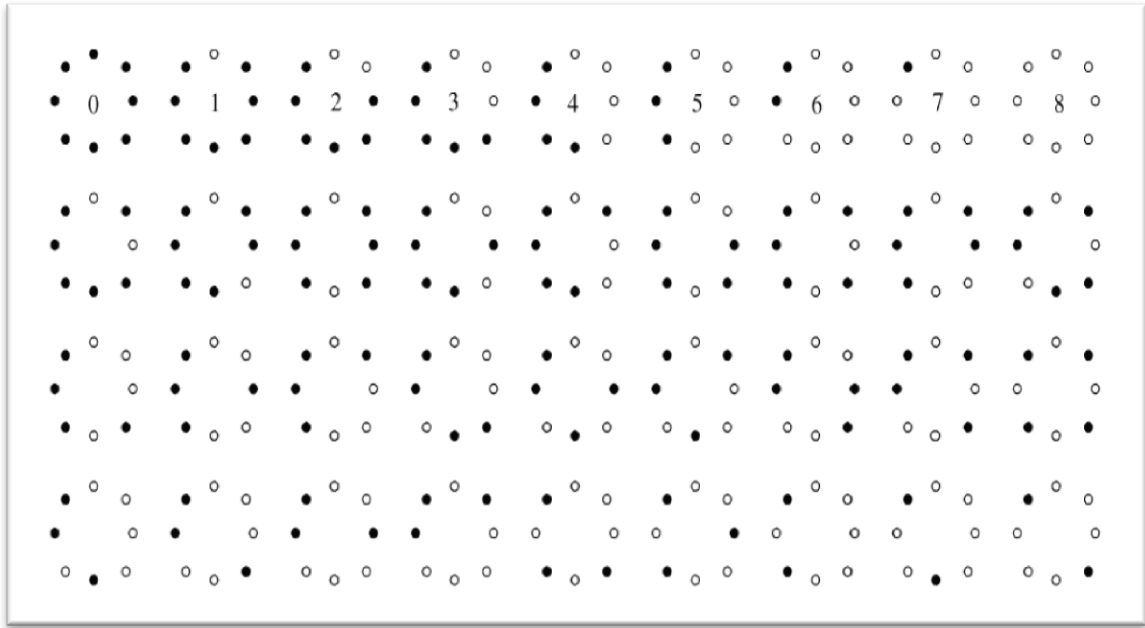
Bu denklemde $s(x)$ aşağıdaki denklem 2.12’de görüldüğü gibi bir işaret fonksiyonudur.

$$s(x) = \begin{cases} 1, & \text{eğer } x \geq 0 \\ 0, & \text{eğer } x < 0 \end{cases} \quad (2.12)$$

LBP operatörünün tanımlamasının doğal bir sonucu olarak yerel komşuluktaki piksel yoğunluklarını koruyan monoton gri düzey dönüşümlerinde değişmezdir. Herhangi bir bölgedeki LBP etiketlerinin histogramı kullanılarak doku tanımlanabilir. $LBP_{P,R}$ operatörü, P piksel komşuluğundaki piksellerden 2^P kadar değişik değer üretir. Üretilen bu örüntülerden bir kısmı birbirinin simetrisi olmaktadır. Mevcut resim döndürüldüğünde ilgili örüntü için değişik değerler üretilir. Örüntüleri rotasyona karşı değişmez yapmak için üretilen her örüntünün ikili değeri için denklem 2.13'deki formülle rotasyona karşı değişmez örüntü elde edilir. Tanıma ve karar verme genellikle bu örüntülerin histogramına bakılarak karar verilir.

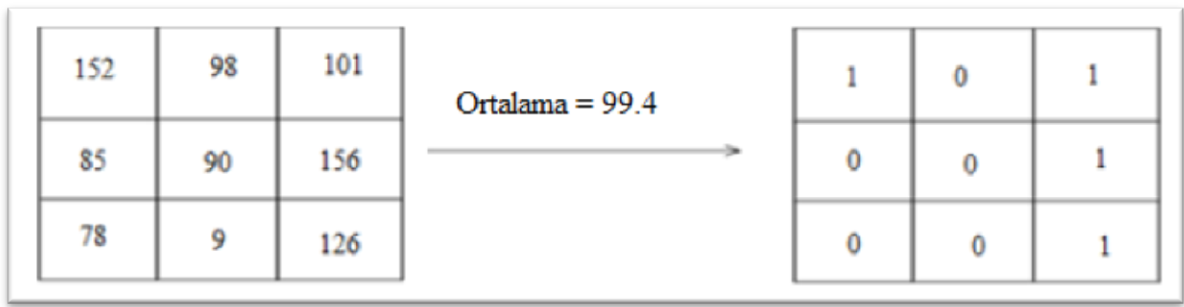
$$LBP_{P,R}^{ri} = \min\{ROR(LBP_{P,R}, i)\}, i = 0, 1, \dots, P - 1\} \quad (2.13)$$

Denklem 2.13'de $ROR(x, i)$ bit düzeyinde dairesel sağa kaydırma işlemini ifade etmektedir. $LBP_{P,R}^{ri}$ ifadesi rotasyona karşı değişmez benzersiz LBP örüntülerini ifade etmektedir. Şekil 2.15.'de 3x3 komşuluk için rotasyona karşı bağımsız LBP örüntüleri verilmiştir. Bu şekilde siyah ve beyaz noktalar bit düzeyinde 0 ve 1'e karşılık gelmektedir.



Şekil 2.15. Benzersiz, rotasyona karşı değişmez, $LBP_{8,R}^{ri}$ için LBP örüntüleri [36].

Klasik LBP operatörü uygulanırken merkez pikselin değeri hiç hesaba katılmaz. Jin v.d. tarafından geliştirilmiş LBP (ILBP) yöntemi sunulmuştur [74]. Bu yöntemde merkez piksel dâhil tüm pikseller ilgili komşuluktaki yoğunluk ortalamasından çıkartılır. Bu şekilde 3x3 komşuluk için 9 bit elde edilir. Klasik LBP operatöründe olduğu gibi sol üst köşeden başlanarak ikili sayı oluşturulur. Merkez pikselin ikili değeri ise en anlamlı bit'e atanır. Bu şekilde $(2^9 - 1)$ adet örüntü elde edilir. Şekil 2.16'da ILBP'ye örnek verilmiştir.

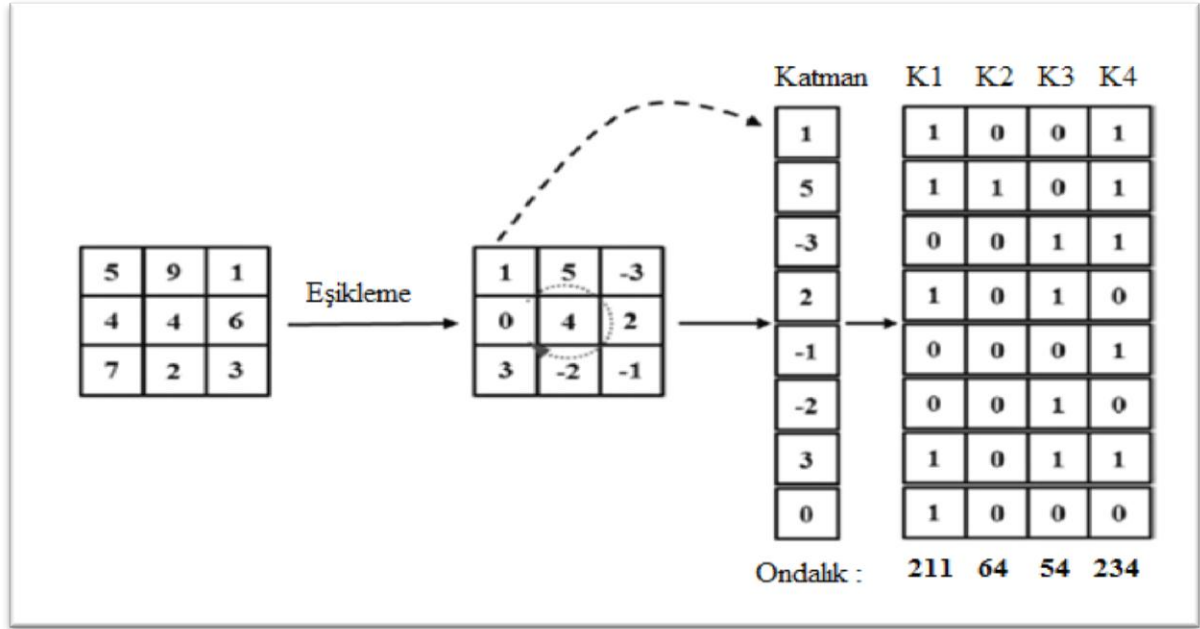


Şekil 2.16. ILBP operatörünün uygulanışı

Başka çalışmalarda [75,76] LBP değeri hesaplanırken ki çıkarma işleminde sayının sayısal büyüklüğü ve çıkarma işleminin sonucunun pozitif veya negatif olduğuna bakılarak klasik LBP yöntemi genişletilmiştir. Genişletilmiş LBP (ELBP) yönteminde örüntünün ayırt edici özelliği yükseltilmiştir. Bu yöntemde temel olarak 4 katman vardır. Birinci katmanda klasik LBP yönteminde olduğu gibi sayının işaretine göre, diğer katmanlarda sayının ikili değerine göre atama yapılır. İkinci, üçüncü ve dördüncü katman için sırayla anlamlı bittten başlanarak bitlerin ikili değerine göre katmanlar için ikili değerler belirlenir. Daha sonra her katmanda alt alta gelen ikili değerlerden onluk sayı bulunur. Şekil 2.17'de ELBP yöntemi için şematik bir gösterim bulunmaktadır.

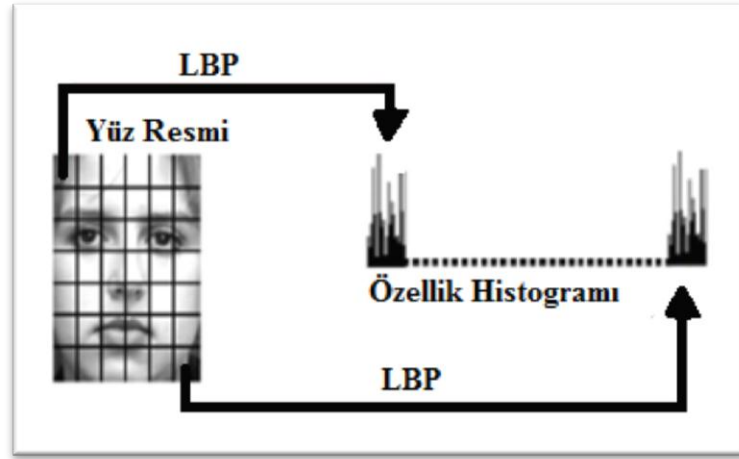
LBP yöntemi ile sadece mikro yapıları yakalamak yerine LBP yöntemi ile makro yapıları yakalamayı mümkün kılan MB-LBP(Multi-scale Block Local Binary Pattern) yöntemi Li v.d. tarafından sunulmuştur [77]. Bu yöntemde resmin tamamı dörtgen bloklara ayrılır. Her blok için ortalama değer hesaplanarak klasik LBP operatöründe olduğu gibi

çıkarma işlemi yapılarak örüntü elde edilir. Hesaplama etkinliğini arttırmak için Viola-Jones yönteminde olduğu gibi integral imge kullanılabilir.



Şekil 2.17. ELBP operatörü örneği [78].

Literatürde LBP yöntemlerinde öğrenme aşmasında, giriş resmini m yerel bölgeye ayırıp bu yerel bölgelerin LBP histogramını yan yana getirmekle tek histogram elde edilmesi sonucu işlemler bu histogram üzerinde yapılır. Bu şekilde oluşturulmuş uzamsal geliştirilmiş özellik histogramı hem yerel doku bilgisini hem de resmin genel şekli hakkında bilgi verir. Klasik LBP kodekleri resmin mikro örüntüleri hakkında konum bilgisi hakkında bilgi içermezken yüzün kısımlarından oluşturulmuş Şekil 2.18.'deki gibi LBPH ile ifade edilişinde konum bilgisi yanında doku bilgisi de içerdiğinden yüzün bu şekilde ifade edilerek bulunması mümkündür. OpenCV'de LBPH yöntemi ile gerçekleştirilmiş yüz modeli mevcut olup değişik amaçlar için kullanılmaya müsaittir. Öğrenme aşamasında temel LBP veya ILBP gibi değişik yöntemlerden elde edilmiş özellikler LDA gibi sınıflandırıcılar kullanılarak veya histogram kesişmesi yapılarak yüz bulunabilir.



Şekil 2.18. Yüzün LBP yöntemi ile ifade edilmesi [79].

2.4.2. Yüz Tanıma

Yüz tanıma uzun bir zamandan beri bilimsel çalışmalarda ve ticari firmalar tarafından oldukça geniş ilgi görmüştür. Günümüzde bilgisayarların işlem kapasitelerinin artması ve mikroçip teknolojisindeki ucuzlamaya paralel olarak birçok uygulamada kullanılmasına rağmen gerçek zamanlı her türlü duruma karşı kararlı uygulamaların geliştirilmesi problemi olarak resim analizi ve bilgisayar görmesi alanlarında üzerinde çalışılmaya devam edilmektedir.

Yüz tanıma temel olarak kişi doğrulama ve kişi tanıma olmak üzere iki görev için kullanılmaktadır. Kişi doğrulamada, mevcut kişinin beklenen kişi olup olmasına karar verilir. Bu tip tanımda bire-bir tanıma işlemi vardır. Yüz tanıma işleminde ise bire-çok bir karşılaştırma vardır. Mevcut, kimliği bilinmeyen bir kişi resmi, veri tabanındaki kişilerle karşılaştırılarak verilen resimdeki kişinin kim olduğu kararı verilir. Yüz tanıma işlemi temel olarak şu uygulamalarda kullanılır;

- Banka, elektronik ticaret, pasaport ve ulusal kimlik gibi genel kimlik doğrulama uygulamaları
- Kayıp çocuk ve polis veri tabanı gibi adli sistem uygulamaları
- Akıllı kart uygulamaları
- Bina girişi, ATM makineleri ve sınır kapıları gibi güvenlik uygulamaları
- Adli ve güvenlik amaçlı mobese uygulamaları

- İnsan bilgisayar etkileşimi uygulamaları
- Video indeksleme (video da yüz etiketleme) uygulamaları

Yüz tanıma işlemi, nesne tanıma problemlerinin zorlu bir kısmıdır. Bu problemin zorluğu yüzlerin önden, yandan ve arkadan gibi değişik formlarda olabilmesi ve bu görüş farklılığından dolayı yüzün tamamen veya kısmen farklı olarak görünmesine sebep olamasıdır. Birçok uygulamada bu problemin önüne geçmek için sadece kişinin karşıdan alınmış görüntüsü üstünde tanıma işlemi gerçekleştirir. Çoğu uygulamanın doğasından dolayı sadece karşıdan yüz alma işlemi yeterli olabilmektedir. Örneğin bu tezin konusu olan DDDS uygulamasında sürücü yola baktığından dolayı sadece karşıdan alınacak görüntü yeterli olur.

Yüz algılama probleminde de belirtildiği gibi kişinin saç, yüz ifadesi ve takı aksesuar gibi etmenlerin olup-olmaması gibi sebeplerden dolayı aynı kişi çok farklı şekillerde görünebilir. Cinsiyet, yaş ve kişinin kökenine göre yüzlerin yassı, oval v.b. gibi değişik olabilmesi gibi durumlar yüz uzayını hem genişletir hem de mevcut yüzleri birbirine yaklaştırır. Bundan dolayı yüzün ayırt edici özelliklerini belirlemek zorlaşır. Tüm bunlara ek olarak giriş imgesi elde edilirken ki aydınlatma, çözünürlük, gürültü v.b. resim elde etme parametreleri giriş imgesinde çeşitliliğe sebep olur. Dolayısıyla tüm durumları dikkate alarak kararlı sonuçlar üreten uygulamalar geliştirmek zorlaşır.

2.4.2.1. Yüz Tanıma Yöntemleri

Yüz tanıma işlemi için literatürde farklı gruplandırmalar mevcuttur. Yüz tanıma teknikleri veri elde etme yöntemlerine göre kaba olarak 3 gruba ayrılabilir [80]. Bunlar giriş resminden yüz tanıma, video dizilerinden yüz tanıma ve kızıl ötesi kameralarla 3D boyutlu veri sağlayan algılayıcılardan gelen veri değerlendirilerek yüz tanıma olmak üzere 3'e ayrılır. Uygun yöntemin seçimi problemin doğasına göre değişmektedir. Örneğin adli resim veri tabanında suçlu tespiti için resim üzerinde işlem yapılarak gerçekleştirilen uykulu sürücü belirleme işleminde videodan gelen veri işlenerek sonuç üretilir. Benzer olarak görüş açısından bağımsız insan algılamasına benzer doğrulukta yüz tespiti için yüzün 3D boyutlu olarak alınıp ona göre tanıma işlemi gerçekleştirilmesi hedeflenir. Bu

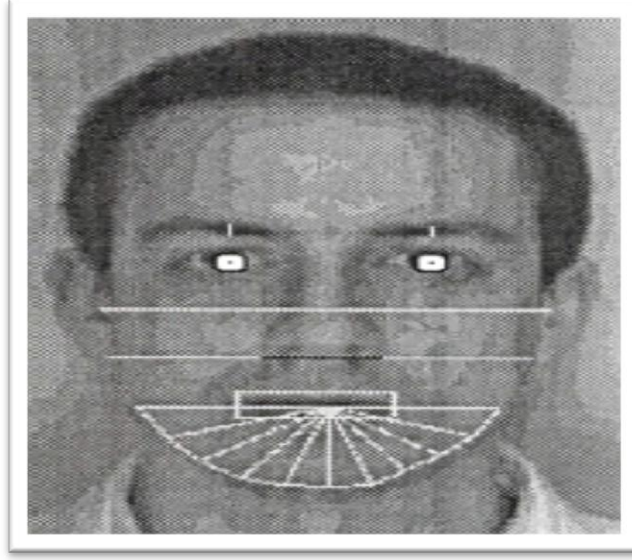
kısımda yüz tanıma yöntemleri kısaca özetlenecektir. Ayrıntılı bilgi için [78, 80-82] nolu referans kaynaklara başvurulabilir.

2.4.2.1.1. Resimden Yüz Tanıma Yöntemleri

Resimden yüz tanıma yöntemleri iki ana gruba ayrılabilir. Bunlar özellik tabanlı ve bütünsel (holistik) yöntemlerdir [80].

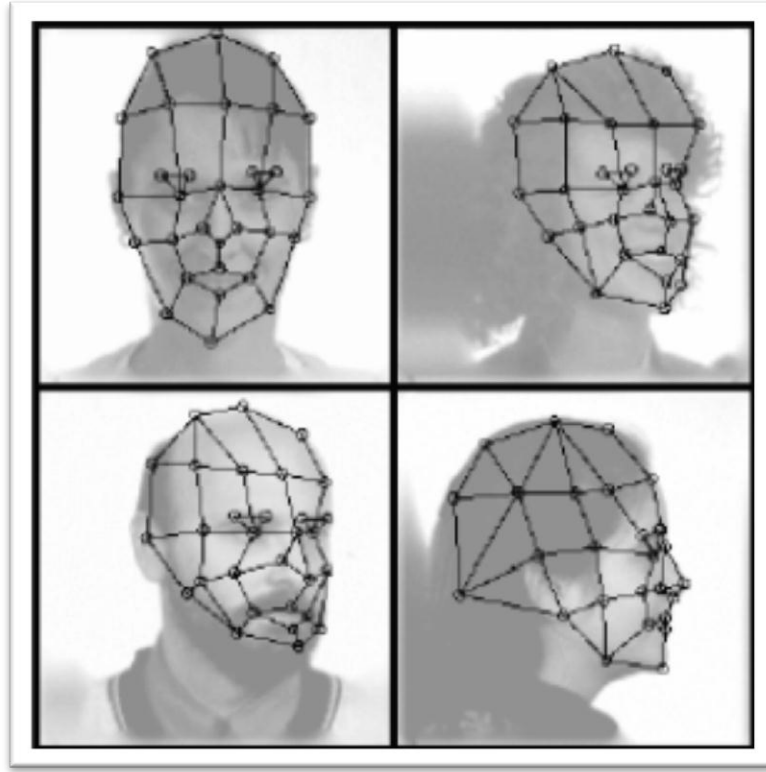
2.4.2.1.1.1. Özellik Tabanlı Yöntemler

Özellik tabanlı yöntemlerde tanıma işleminin icra edilmesi ve ayırt edici yüz özelliklerin belirlenmesi için ilkin göz, burun, kaş, çene gibi yüz bileşenlerinin bulunması gerekir. Bu özellikler ve diğer kullanılabilecek noktalar arasındaki geometriksel ilişkiyle yüz ifade edilir. Bu şekilde problem uzayı indirgenir. Yani ham piksel verileri yerine yüzü ifade eden özellikler arasındaki ilişkiden yüz tanınmaya çalışılır. Bu yöntemlerde standart istatistiksel örüntü tanıma teknikleriyle özelliklerden elde edilen ölçülerle yüz tanınmaya çalışılır. İlk yüz tanıma çalışmalarının çoğu bu yöntem temel alınarak gerçekleştirilmeye çalışılmıştır. Bu yöntemde sistemin doğruluğunu ve kararlılığını arttırmak için zaman içinde birçok teknik sunulmuştur. Yeniden oluşturulabilir şablonlar, Hough dönüşümü, morfolojik işlemler ve Reisfeld simetri operatörü gibi değişik teknikler kullanılarak yüz tanınmaya çalışılmıştır. Bu yöntemlerde yüzü ifade etmek için genel özellikler belirlenmesine rağmen yüzdeki yapıların birebir ifade edildiği söylenemez. Genel olarak otomatik özellik çıkarma yöntemlerini kullanan algoritmaların yüksek işlem gereksinimlerine rağmen istenilen iyi doğruluk oranıyla tanıma işlemini gerçekleştirdikleri söylenemez [83]. Şekil 2.19'da yüz tanıma işlemi için örnek geometrik özellikler verilmiştir.



Şekil 2.19. Yüz tanıma işlemi için geometrik özellikleri (beyaz) [84].

Bu yöntemde iyi bilinen bir diğer yöntem ise EBGM (Elastic Bunch Graph Matching) yöntemidir. Bu yöntemde bir dikdörtgenler ızgarası insan yüzüyle örtüştürülerek yüz üzerindeki kritik noktalar belirlenmesi hedeflenir. İki boyutlu Gabor dalgacıklarının katlama setinin dikdörtgenler ızgarası üzerindeki her nokta için hesaplanmasıyla, kritik noktaları ifade eden özellik vektörü elde edilir. Bu şekilde herhangi bir yüz için ilgili noktalardan elde edilmiş bir graf elde edilir. Daha sonra graf eşleme yöntemlerinden biri kullanılarak elde edilen graf şablonlarının karşılaştırılması yapılır. Bu yöntemlerde genellikle yüksek doğrulukla yüz tanınır. Ancak büyük boyutlarda yüz resimlerine ihtiyaç duyarlar. Bu yüzden bu yöntemlerde büyük resimlerle çalışmasına ihtiyaç duyarlar. Şekil 2.20’de EBGM örneği verilmiştir. İlk çalışmalarda yüz üzerindeki noktaların elle belirlenmesi problemi daha sonra yeniden oluşturulabilir şablonlar kullanılarak otomatik yapıldığı çalışma sunulmuştur [85].



Şekil 2.20. EBGM yönteminde yüz tanıma için ızgaralar [86].

Bu yöntemde diğer ciddi bir çalışma türü de yüzü yandan tanımaya yönelik çalışmalardır. İlk çalışmalarda gerek doğruluk oranı olsun gerekse sistemi gerçekleştirirken ki kullanılan veri tabanının ne kadar genel olduğu soru işaretidir [80]. Daha sonraki [87] nolu çalışmada daha genel bir veri tabanı için %90 doğrulukla yüz tanındığı bildirilmiştir.

Bu yöntemlerin en büyük avantajı yüzü ifade eden özelliklerle, gerçek yüzün ifade edilebilmesidir. Yüz resimlerinde tanımaya mani olabilecek boyut, poz, aydınlanma ve yüz ifadesi gibi özelliklerin modelleme üzerindeki olumsuz etkileri giderilebilir. Yüzü kısa bilgilerle ifade edip yüksek hızda tanımayı mümkün kılan yöntemlerdir.

Bu yöntemlerin dezavantajı olarak özelliklerin belirlenmesi zordur. Yüze ait özelliklerin otomatik olarak elde edilmesi kolay değildir. Hangi özelliğin önemli olduğuna karar verilmesi gerekir. Bu yöntemlerde giriş verisi olarak yüksek çözünürlüklü resimlere ihtiyaç duyulur.

2.4.2.1.1.2. Bütünsel Yöntemler

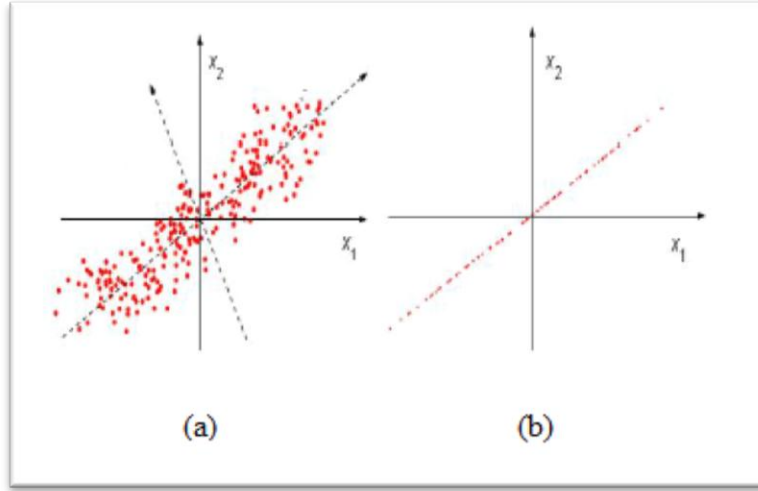
Bütünsel yöntemlerde verilen resimdeki yüz için yerel özellikler yerine giriş resminin tamamından yüz ifade edilerek yüz tanınmaya çalışılır. Bu yöntemleri kendi araların istatistiksel ve AI (Yapay Zekâ) yöntemleri olmak üzere ikiye ayırmak mümkündür.

2.4.2.1.1.2.1. İstatistiksel Yöntemler

İstatistiksel yöntemlerin en genel halinde resim 2D boyutlu yoğunluk dizisi olarak ifade edilir ve tanıma işlemi mevcut yüzün veri tabanındaki diğer tüm yüzlerle karşılaştırması sonucu elde edilen ilişkinin bir sonucu olur. Bu yöntemlerin başarımı için ışıklandırma, ölçeklendirme ve poz gibi beklenen uygun şartların olmasına bağlıdır. Bu yöntemlerin en iyi bilinenleri aşağıda sıralanacaktır.

PCA yöntemi resimdeki küçük bölgeleri en iyi şekilde ifade edecek vektörlerin hesaplanmasına dayanır. Bu yöntemin ana amacı verilen resimdeki yüzleri en iyi ifade edecek vektörlerin bulunması esasına dayanır. Şekil 2.21’de Temel PCA işlemi gösterilmiştir. PCA yönteminde mevcut resim, yüz uzayı olarak adlandırılan bir alt uzaya iz düşürülür. Mevcut yüzün yüz uzayındaki izdüşümü ile uzayda konumu bilinen diğer bağımsız yüzlerle karşılaştırılarak yüz bulunmaya çalışılır. PCA yönteminde varyantın maksimum olduğu bir alt uzay oluşturulmaya çalışılır.

LDA yönteminde verilerin sınıflandırılması için gerekli olan ayırt edici öznitelikleri seçip, ayırt edici olmayan öznitelikleri elenmesi hedeflenir. LDA aynı zamanda FLDA (Fisher Doğrusal Diskriminant Analizi) yöntemi olarak anılır. FLDA sınıflar arası farklılığı artırırken, sınıf içi saçılımı küçültmeyi hedefleyen doğrusal bir öznitelik çıkarma yöntemidir. Böylelikle görüntüleri analiz ederken onların içerikleri değil özniteliklerine göre analiz edebilen bir yöntemdir. İstatistiksel yöntemlerin ayrıntılı literatür özeti için [80-82] nolu kaynaklara başvurulabilir.



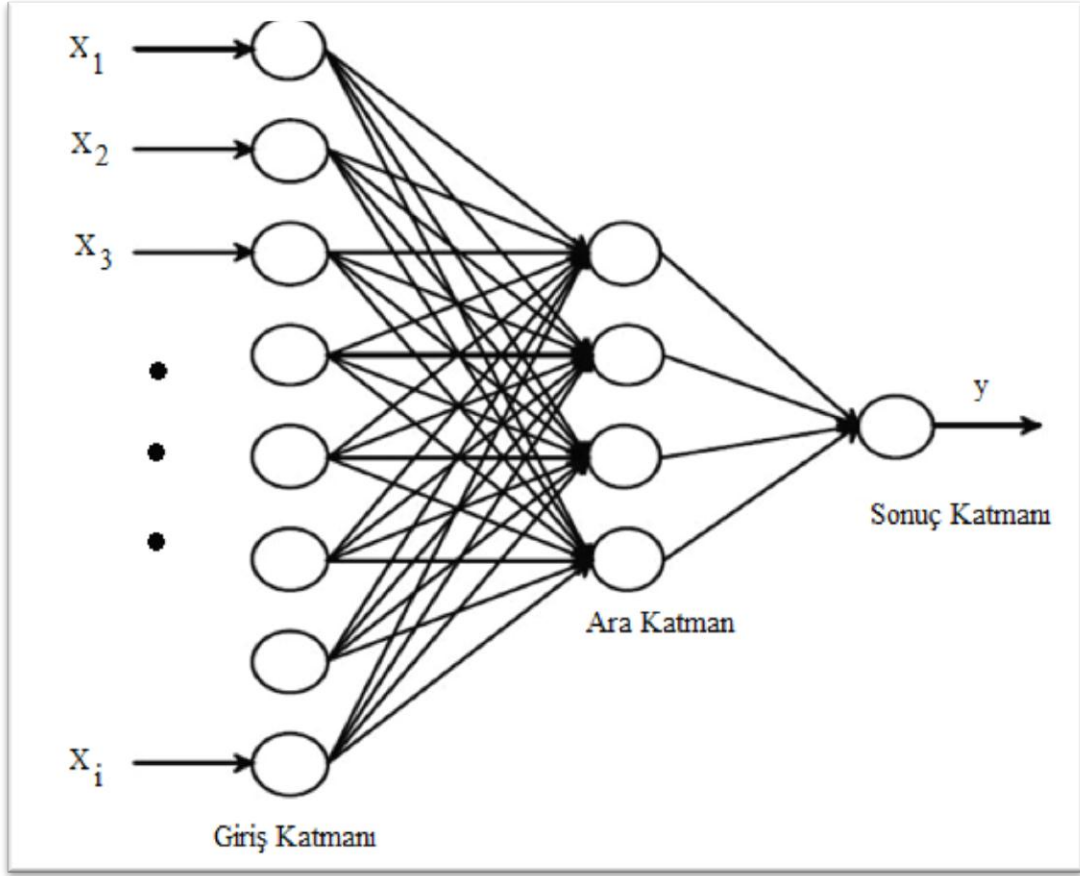
Şekil 2.21. Temel PCA (a) ve 1D iz düşürülmüş PCA (b) [81].

ICA (Independent Component Analysis) yönteminde mevcut yüzdekilerle ilişkili olmayan (istatistiksel olarak bağımsız), sadece en anlamlı özellik çıkarılmaya çalışılır. Performansı arttırmak için ek bileşenler analizi gerekebilir.

Bu yöntemlerde yüz tanıma problemi çok iyi bilinen ve başarıımı yüksek olan yüz uzayı analizi problemine dönüşür. Bu yöntemlerde düşük çözünürlüklü resimlerle çalışılmasına olanak vermesi bu yöntemlerin üstün olduğu yön olarak verilebilir. Bu yöntemlerin dezavantajı çok sayıda test yüz resmi gerekmesi, poz aydınlatma ve yüz ifadesi gibi özellikleri modele katmanın güç olması sayılabilir. Bu tez çalışmasının uygulamasında LDA yöntemiyle yüz tanındığı için LDA yöntemi ayrıntılı olarak anlatılacaktır.

2.4.2.1.1.2.2. AI Yöntemleri

AI yöntemlerinde YSA ve makine öğrenmesi teknikleriyle yüzün tanınması hedeflenir. Bu yöntemler çok popüler olup uygulamada birçok farklı teknikle gerçekleştirilen model sunulmuştur. YSA’ da ilkin temel bileşenler(özellikler) çıkarılır daha sonra ise bu temel bileşenler eğitilmiş çok katmanlı bir yapay sinir ağına verilerek sonuç elde edilir. Temel bir yapay sinir ağı şekil 2.22’de verilmiştir.



Şekil 2.22. Temel bir YSA modeli örneği

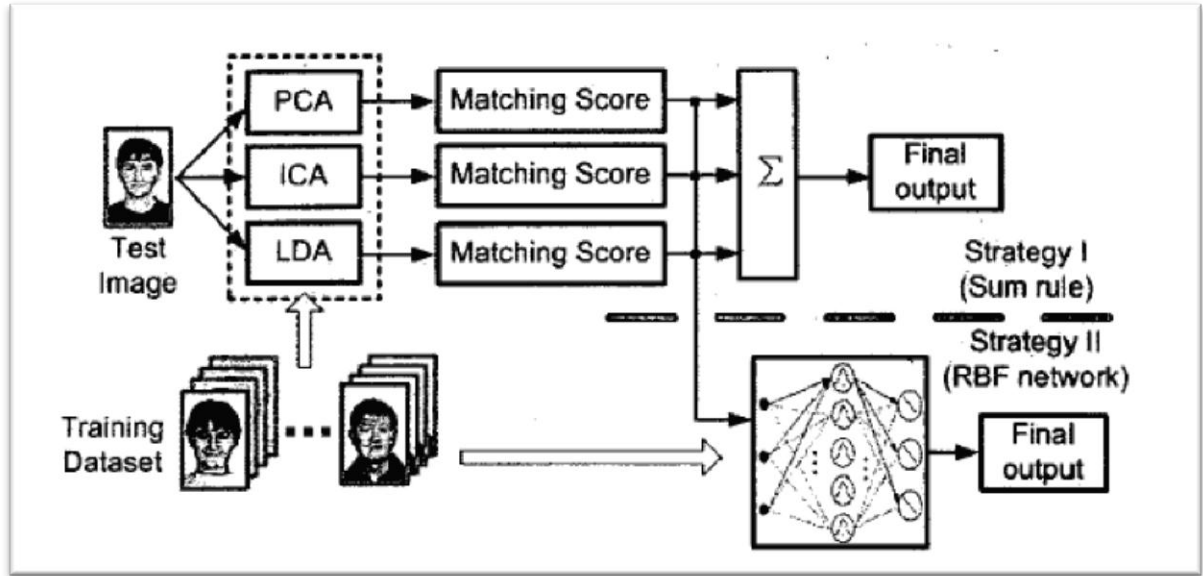
Literatürde standart çok katmanlı, hiyerarşik ve ileri beslemeli gibi değişik YSA modelleriyle gerçekleştirilmiş yüz tanıma uygulamaları mevcuttur. Özellik çıkarmak için temel bileşenler ve öz yüzler gibi değişik yöntemlerle kullanılabilir veya yüzdeki bileşenler parçalı olarak YSA' ya giriş verisi olabilir. Klasik YSA problemlerindeki dezavantaj ve avantajlar yüz tanıma problemi içinde geçerlidir.

Makine öğrenmesi algoritmalarında daha önce yüz bulunması kısmında değinildiği gibi yükseltme algoritması (Adaboost), Saklı Markov Modeli ve SVM (Destek Vektör Makineleri) gibi iyi bilinen algoritmalar kullanılarak sistem gerçekleştirilmek istenir. Sisteme giriş verisi olarak yükseltme işleminde LBP yöntemi ile elde edilmiş özellik histogramından Chi-kare ile en anlamlı özellik verilerin belirlenmesi ve Adaboost ile ayırt edici özellikler belirlenerek yüz tanıma çalışılır. Bu yöntemlerde yüksek doğruluk oranıyla sistem

gerçeklenebilir. Özellikle Adaboost algoritmasının hızlı bir şekilde icra edilmesinden dolayı gerçek zamanlı uygulamalarda tercih sebebi olur.

2.4.2.1.1.3. Çoklu Sınıflandırıcı Sistemler

Bu yöntemlerde PCA, LDA, ICA ve YSA gibi tek sınıflandırıcı kullanmak yerine bu sınıflandırıcılardan bir kaçını birlikte kullanarak sistemin doğruluğu artırılarak sistem gerçekleştirilmek istenir. Şekil 2.23'te çoklu sınıflandırıcıya örnek verilmiştir.



Şekil 2. 23. Yüz tanıma işlemi için çoklu sınıflandırıcı örneği [88].

Şekil 2.23'de görüldüğü gibi çoklu sınıflandırıcı kullanılarak yüz tanımanın herhangi bir sınıflandırıcı dâhil edilerek gerçekleştirilebileceği gösterilmiştir. Çoklu sınıflandırıcı kullanmanın sistemin doğruluğunu artırdığı bildirilmiştir [88].

2.4.2.1.2. Video Dizilerinden Yüz Tanıma

Son yıllarda yapılan yüz tanıma çalışmalarının birçoğu bu alanda yapılmaktadır. Gerçek zamanlı olacak şekilde güvenlik amaçlı yüz tanıma sistemlerinde, bir video yakalayıcıdan gelen resim dizeleri değerlendirilerek yüz tanıma işlemi gerçekleştirilir.

Video tabanlı yüz tanıma sistemleri genel olarak yüz tespiti, yüz izleme ve yüzü tanıma olmak üzere 3 ana bileşenden oluşur. Bu yöntemlerde kişi hareketli olduğundan ve genelde dinamik bir ortamda çalışıldığından iyi bir video karesi seçilerek yüz tanıma işlemi seçilen kare üzerinde icra edilir. Saniyede yakalanan kare sayısının fazla olduğu durumlar göz önüne alınarak bazen sadece kare atlanarak bazen de yüz izlenerek aynı yüz olduğu kanaati getirilerek her kare için tanıma işlemi icra edilmez. Tanınmak istenilen kişinin hareketli olduğu varsayıldığında belli bir açıdaki poz , +90/-90 gibi, hesaba katılarak kişi tanınmaya çalışılır. Yüksek doğrulukta tanıma işlemi gerçekleştiren sistemler gerçekleştirilmiştir [80]. Bu yöntemlerde ne kadar gerçek hayat koşulları dikkate alınarak yüz tanınmaya çalışılsa da gerçek hayatın dinamikliği yani çevresel etmenler oldukça yüksek gürültülere sebep olabilmektedir. Çevresel etmenler ve görüntü elde edilirken ki sebeplerden oluşan gürültüler, poz ve diğer istenmeyen etmenlerin varlığından dolayı bu yöntemlerin gerçek zamanlı uygulamalarda kullanılması zorluluğundan dolayı ilerde de üzerinde de çalışılmaya devam edecek potansiyele sahip yöntemlerdir.

2.4.2.1.3. Diğer Algılayıcılarda Yüz Tanıma

Günümüzde birçok çalışmada 2D boyutlu resimlerden yüz tanımaya odaklanmıştır. Son yıllarda 3D veri veya erim veri ve kızıl ötesi resimlemeyle elde edilmiş veriden tanıma işlemi gerçekleştirilmeye çalışılmıştır.

3D model tabanlı sistemlerde alın şekli, çene çizgisi ve yanaklar gibi şekil ve eğrilik özellikleri kullanılarak yüz tanınmaya çalışılır. Bu şekilde tanımada 2D tanıma probleminde görülen ışıklandırma, rotasyon ve arka plan gürültüleri sistemi etkilemeyecek hale gelir.

Isıya duyarlı kızıl ötesi görüntülemeyle ışıklandırmadan bağımsız yüz tanıma işlemi gerçekleştirilebilir. Bu şekilde tanıma ışıktan dolayı klasik tanıma problemlerinde görülen birçok probleme otomatik çözüm getirmesine rağmen kızıl ötesi kameraların yüksek

maliyetinden dolayı normal uygulamalarda tercih edilmez. Yüksek maliyet bu tür görüntülemenin tek sorunu değildir. Bu tür görüntülemeye çözünürlük düşük olup gözlük gibi aksesuarlar yüzün bir bileşeni olarak görünebilir. Ayrıca kişi vücut sıcaklığı, rüzgâr ve ortam sıcaklığı elde edilen veride gürültüye sebep olabilir.

2.4.2.2. OpenCV ve Yüz Tanıma

Bu kısımda OpenCV yüz tanıma modelleri hakkında genel bilgi verilecek olup DDDS uygulamasında kullanılan LDA (Fisher) yüz tanıma yöntemi hakkında ayrıntılı bilgi verilecektir.

2.4.2.2.1. OpenCV Yüz Tanıma Modelleri

OpenCV kütüphanesindeki tüm tanıma modelleri, yüz tanıma algoritmalarına tek şekilde erişimi sağlayan soyut FaceRecognizer sınıfından türetilir. Aşağıda FaceRecognizer sınıfı verilmiştir [89]:

```
class FaceRecognizer : public Algorithm {
public:
    // Sanal yıkıcı
    virtual ~FaceRecognizer() {}
    // Model eğitim fonksiyonu
    virtual void train(InputArray src, InputArray labels) = 0;
    // Model güncelleme fonksiyonu
    virtual void update(InputArrayOfArrays src, InputArray labels);
    // Model'den bir tahmin alır
    virtual int predict(InputArray src) const = 0;
    // verilen bir etiket ve resim için doğruluk tahmin eder
    virtual void predict(InputArray src, int &label, double &confidence) const = 0;
    // Modeli kaydeder
    virtual void save(const string& filename) const;
    // Modeli yükler
```

```

virtual void load(const string& filename);
// cv:: FileStorage ile verileni kaydet
virtual void save(FileStorage& fs) const = 0;
// cv:: FileStorage ile verileni yükle
virtual void load(const FileStorage& fs) = 0;
};

```

Tüm algoritmalar, modelleri XML ve YAML dosyaları şeklinde, modeli yükleme ve kaydetmek için kendi prosedürlerini içerirler. Bunların tekrardan yazılmasına gerek yoktur.

Eğitim aşamasında yüz veri tabanındaki tanınacak kişiler için bir eğitim resmi veya resimleri seçilerek kişinin kimliğine göre etiketlenir. Etiketlenen bu veriler modele verildiğinde model ilgili kişileri tanıyacak duruma gelir.

Tahmin aşamasında ise herhangi bir kişiye ait veri sisteme verilerek sistemin daha önce öğrendiği verilere göre sonuç üretmesi beklenir. Geri dönen sonuç sistemin tanıdığı kişi etiketidir.

Yükleme ve kayıt işlemleri modellerin tekrar tekrar eğitimine gerek duymadan yüklenmesine olanak verdiği için uygulamada büyük avantaj sağlar. İlgili modeller XML veya YAML olarak kaydedilebilir.

Modeller varsayılan değerleriyle oluşturula bileceği gibi ilgili parametreler değiştirile bilmektedir. Örneğin aşağıdaki kod parçacığı 10 adet öz yüzü ve 10.0 eşik değeriyle PCA yöntemiyle bir yüz tanıma modeli oluşturur.

```

int num_components = 10;
double threshold = 10.0;
Ptr<FaceRecognizer> model = createEigenFaceRecognizer(num_components, threshold);

```

Model parametreleri atanabilirdiği gibi ilgili parametreler okunabilir. Örneğin “std::string name = model->name();” kod parçacığı modelin adını verir.

2.4.2.2.2. LDA (Fisher) Yüz Tanıma Algoritması ve Modeli

LDA yöntemi sınıf-spesifik boyut azaltmayı sağlayan bir yöntem olup ünlü istatistikçi R.A. Fisher tarafından sunulmuştur [90]. Bu yöntem [91] nolu çalışma ile yüz tanıma problemine uyarlanmıştır. LDA yönteminde verilerin sınıflandırılması için gerekli

olan ayırt edici öznitelikleri seçip, ayırt edici olmayan öznitelikleri elenmesi hedeflenir. LDA aynı zamanda FLDA (Fisher Doğrusal Diskriminant Analizi) yöntemi olarak da anılır. FLDA sınıflar arası farklılığı arttırırken, sınıf içi saçılımı küçültmeyi hedefleyen doğrusal bir öznitelik çıkarma yöntemidir. LDA yöntemi, öz yüzlerin temeli olan PCA yöntemine göre ışıktandırma problemlerine karşı daha kararlıdır [92]. Bu yöntemin optimum parametreleri için değişik çalışmalar yapılmıştır. Bu yöntem temel olarak şu şekilde ifade edilebilir;

- X , c - nesnelerinden alınmış örnekler olsun

$$X = \{X_1, X_2, \dots, X_c\}$$

$$X_i = \{x_1, x_2, \dots, x_n\}$$

- Genel ortalamayı (μ) hesapla

$$\mu = \frac{1}{N} \sum_{i=1}^N x_i \quad (2.14)$$

- μ_i sınıf ortalamasını hesapla $i \in \{1, \dots, c\}$

$$\mu_i = \frac{1}{|X_i|} \sum_{i=1}^N x_i \quad (2.15)$$

- Dağılım matrisleri S_B ve S_W hesaplanır

$$S_B = \sum_{i=1}^c N_i (\mu_i - \mu) (\mu_i - \mu)^T \quad (2.16)$$

$$S_W = \sum_{i=1}^c \sum_{x_i \in X_i} (x_i - \mu_i) (x_i - \mu_i)^T \quad (2.17)$$

- Klasik Fisher algoritması nesne ayrılabilirliğini maksimize eden W iz düşümünü arar

$$W_{opt} = \operatorname{argmax}_W \frac{|W^T S_B W|}{|W^T S_W W|} \quad (2.18)$$

Bu optimizasyon problemi [91] nolu çalışmada genel öz değerler problemi çözülerek şu şekilde ifade edilmiştir.

$$S_B v_i = \lambda_i S_W v_i \quad (2.19)$$

$$S_W^{-1} S_B v_i = \lambda_i v_i \quad (2.20)$$

Bu optimizasyon yönteminde çözülmesi gereken bir problem daha var. N örnekli ve c nesneli problemde S_W 'nin derecesi genellikle (N-c)'dir. Örüntü tanıma problemlerinde genellikle örneklem sayısı N genellikle giriş datası boyutundan (piksel sayısından) küçüktür. Bu yüzden dağılım matrisi S_W tekil olur [93]. Dolayısıyla bu matrisin tersi alınmaz. Bu problem [91] nolu çalışmada giriş verisine PCA uygulanarak örneklerin (N-c) boyutuna iz düşürülmesiyle çözülmüştür. Daha sonra LDA iz düşürülmüş veriye uygulanır. Çünkü S_W artık tekil olmaktan çıkmıştır. Optimizasyon probleminin son hali denklem 2.21 ve denklem 2.22'de verilmiştir.

$$W_{pca} = \operatorname{argmax}_W |W^T S_B W| \quad (2.21)$$

$$W_{fld} = \operatorname{argmax}_W \frac{|W^T W_{pca}^T S_B W_{pca} W|}{|W^T W_{pca}^T S_W W_{pca} W|} \quad (2.22)$$

Bir örneği (c-1) boyutuna atan transformasyon matrisi W şu denklem 2.23'deki gibi hesaplanır:

$$W = W_{fld}^T W_{pca}^T \quad (2.23)$$

Bu modelde resimler gri düzeyde olmalıdır. Renkli resimlerin gri düzeye çevrilmesi gerekir. Fisher yönteminde eğitim aşamasında kullanılan verilerin boyutuyla sisteme giriş olacak verilerin boyutunun aynı olması gerekir aksi takdirde yöntem işlemez bu yüzden OpenCV fonksiyonlarından `resize()` fonksiyonu kullanılarak resmin boyutunun standartlaştırılması lazımdır. Bu model güncellemeyi desteklemez. Model şu şekilde tanımlanır;

Ptr<FaceRecognizer> createFisherFaceRecognizer(int num_components=0, double threshold=DBL_MAX).

Burada “num_components” tanınacak kişilerin sayısının belirlene bildiği bir değişkendir. Eğer boş bırakılırsa veya sıfıra olarak atanırsa varsayılan olarak eğitildiği veri adedinin bir eksiği olarak atanır. “threshold” ise mevcut tanınacak veri için ölçülen değerin karşılaştırılacağı referans değerdir. Modelin sonucu bu eşik değerinden büyükse varsayılan olarak kişinin tanınmadığı kabul edilir ve geriye (-1) döndürür.

Modelin iç parametreleri şu şekildedir;

- num_components : Yukarıda açıklandı.
- threshold : Yukarıda açıklandı.
- eigenvalues : LDA analiz için azalan sırada öz yüzler
- eigenvectors : LDA analizi için özyüz değerine göre sıralı özyüz vektörleri
- mean : Eğitim setinden hesaplanan ortalama değer
- projections : Eğitim setinin iz düşümü
- labels : İz düşüme göre etiket (kişi kimliği)

2.4.3. Göz Tespiti

Göz tespiti, son 30 yıldır birçok uygulamada kullanılmasına rağmen göz çeşitliliği, ölçekleme, konum ve ışıklandırma gibi problemlerin varlığından dolayı hala üzerinde çalışılan bir konu olmaya devam etmektedir. Göz tespiti popüler birçok uygulamanın bir aşaması veya yardımcı bir modülü olarak bilgisayar görmesi alanında ilgi görmüştür. Göz tespiti çalışmaları; DDDS sisteminde [1] karar sürecinde, yüz tanıma probleminde [94] yardımcı modül ve bazen sadece kararlı göz tespiti [95] için göz tespiti konusunda çalışılmıştır.

DDDS sisteminde karar sürecinde PERCLOS kullanılacak olup bu parametre uykululuğu tespit için en çok kullanılan parametredir [14]. Dolayısıyla sistemin çıkış üretebilmesi için ilk önce gözlerin bulunması gerekir. DDDS modülündeki göz algılama işlemi için [1]’de YSA kullanılarak göz tespit edilmiştir. [20,21]’de insan teni algılanarak gözler bulunmuştur. [27]’de iki kızıl ötesi filtre kullanılarak gözden yansıyan ışık farkı ile göz tespitini yapan bir yapı önerilmiştir. [14]’da FPGA tabanlı, retinadan yansıyan ışığın yoğunluğunun farklı olacağı esasına dayanarak göz bulmuştur.

2.4.3.1. Göz Tespit Yöntemleri

Literatürde nesne tanıma problemleri genel olarak birbiriyle benzer yöntemlerle tanındığından dolayı yüz algılama kısmın da anlatılan yöntemler genel olarak göz tespiti içinde kullanılmaktadır. Farklı sınıflandırmalar olabileceği gibi genel olarak göz tespiti yöntemleri; Şekil tabanlı, özellik tabanlı, görünün tabanlı, melez yöntemler ve diğer yöntemler olarak beşe ayırabilir [96].

2.4.3.1.1. Şekil Tabanlı Yöntemler

Açık bir göz onun şekliyle kolayca ifade edilebilir. Açık bir göz, iris ve göz bebeği hatları ile gözün çevresindeki (göz kapağı) şekilleri yardımıyla ifade edilebilir. Şekil tabanlı yöntemlerin sınıflandırılması mevcut modelin basit eliptik ya da daha karmaşık bir halde olmasına bağlıdır. Şekil modelleri genellikle bir geometrik göz modeli ve bir benzerlik ölçüsü olmak üzere iki bileşenden oluşturulmaktadır.

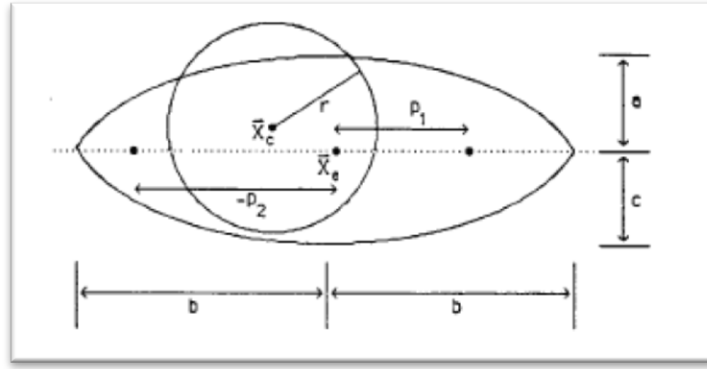
Geometrik modelin parametreleri izin verilen şablon deformasyonları tanımlar ve katı olmayan şablon deformasyonları için parametreler ile benzerlik dönüşümleri için parametreler içerir. Deforme şekil modelleri genellikle genel bir deforme şablona dayanırlar. Bu şablon yüz modelinin bir enerji eniyileme yoluyla şeklin deforme edilmesiyle göz konumlandırılır. Bu yöntemlerin önemli bir özelliği şekil, ölçeklendirme ve rotasyon problemlerini kendi doğal özelliklerinden dolayı elemine edebilmesidir.

2.4.3.1.1.1. Basit Eliptik Şekil Modelleri

Birçok göz izleme uygulaması sadece iris veya göz bebeğinin algılanıp izlenmesi yeterli olmaktadır. Görüş açısına bağlı olarak iris ve göz bebeği genellikle eliptiktir ve 5 parametre ile iris ve göz bebeği modellenir. Basit eliptik modeller; oylama tabanlı yöntemler [97,98] ve model uyarlama yöntemler [99,100] olmak üzere kendi arasında ikiye ayrılır. Oylama yöntemleri bir oylama algoritmasına göre özellikleri ağırlıklandırılır daha sonra ise bu ağırlıklardan sonuç üretirler. Model uyarlama yöntemleri ise özellikleri mevcut modele uyarlayarak sonuç üretirler.

2.4.3.1.1.2. Karmaşık Şekil Modelleri

Karmaşık Şekil Modelleri gözün şeklinin daha ayrıntılı bir şekilde modellendiği yöntemlerdir. Bu yöntemlerden önemli bir çalışma, deforme edilebilir şablonlarla Yuille v.d. tarafından sunulmuştur [48]. Gözün deforme edilebilir şablonunda, göz kapağı iki adet parabol ile iris ise bir adet daire ile temsil edilmiştir. Yuille sunduğu model bir güncelleme kuralına göre vadiler, kenarları, resim zirveleri ve iç kuvvetlerin enerji fonksiyonlarından bağımsız olarak resme uygulanmıştır. Şekil 2.24’de, [48]’de sunulan model verilmiştir. Ivin ve Porril [101] video görüntülerinde irisin 3D hareketini izleyen bir yöntem sunmuşlardır. Mevcut çalışmaları yüksek kalitede çözünürlük gerektiren bir video yakalayıcı gerekmektedir.



Şekil 2.24. Göz modeli için parametreler [48].

Deforme edilebilir şablon tabanlı yöntemler mantıklı ve genellikle doğru sonuç üreten yöntemler olarak bilinir [96]. Bu yöntemlerin dezavantajları şu şekilde sıralanabilir:

- Hesaplamaları zordur
- Yüksek kontrastlı görüntüler gerektirir
- Genellikle yüksek başarımlar için gözen yakına getirilmesi gerekir
- Büyük baş hareketlerinde yöntem başarısız olacağı için başarımlarını sağlamak için başka yöntemler gerekir
- Yüz poz değişiklikleri ve göz tıkanıklığı sorunlarında yöntem başarımları iyi değildir

2.4.3.1.2. Özellik Tabanlı Yöntemler

Özellik tabanlı yöntemler gözü ifade eden ayırt edici özellikleri belirlemeye dayanır. Limbus (gözde yer alan kornea ve skleranın birleşme yeri), gözbebeği (parlak / koyu gözbebeği resimleri) ve kornea yansımaları, gözü konumlandırmak için en çok kullanılan özelliklerdir. Genel yöntemlerle karşılaştırıldığında özellik tabanlı yöntemler yüz ve gözü tanımada yerel tanımlayıcı özellikleri tanımlamayı hedefler ve ışık ve görüş açısına değişimlerine göre daha az hassastır.

Waiti ve Vincent [102] nolu çalışmalarında kısım tabanlı bir yöntem sunmuşlardır. Bu yöntemde, göz köşeleri veya göz kapakları bir kısmı ifade edip bu kısımlara mikro yapılar denir. İlgili yöntemlerinde çok katmanlı YSA kullanarak gözü yüzde konumlandırmak için yüz özelliklerini kullanmışlardır.

Herpels v.d. [103] çalışmalarında çizgi, köşe, rotasyon, uzunluk ve ölçekleme gibi yerel özellikleri bulan bir yöntem sunup göz algılama için bir göz şekli modeli sunmuşlardır.

Kawato ve Ohya [104] göz özellikleri algılamak yerine iki göz arasındaki bölgeyi algılamışlardır. Onların sunduğu yöntemde göre bu alanın iki yanındaki karanlık bölgeler göz ve göz kapaklarından oluşan bölgedir.

Sirohey v.d. [105,106] çalışmalarında doğrusal ve doğrusal olmayan filtrelerle gözü algılamışlardır. Gözün kenarları dört adet Gabor dalgacıkları ile tespit edilmiştir. Doğrusal olmayan bir filtre sol ve sağ göz köşeleri için aday bölgeleri belirlemede kullanılmıştır. Bu köşeler daha sonra göz bölgesi tespitinde kullanılmıştır. Geriye dönük bir işlemle bu bölgelerden anlamlı olanlar çıkarılmıştır. Bir oylama yöntemiyle gözün iris bölgesi konumlandırılmıştır.

Yang v.d. [107] çalışmalarında yenilemeli bir eşikleme metodu sunmuşlardır. Bu metodu uygulayarak insan tenini de kullanarak buldukları iki karanlık bölgeyi göz bebeğini olarak etiketlemişlerdir. Bu metodun başarımı büyük ölçüde insan teni algılama modelinin başarımına bağlıdır.

Özellik tabanlı yöntemlerle genellikle ışık değişimlerinde bile sağlam sonuçlar elde edilmiş çalışmalar sunulmuştur. Görüş açısındaki değişimleri büyük görünüm farklılıklarına sebep olsa bile bazı özellikler değişmez. Göz bebeğini algılamaya yönelik teknikler kapalı mekânlarda karanlıkta bile iyi sonuç verirler. Fakat dışarıda iyi sonuç

vermez çünkü dışarıda göz bebeği küçülür ve göz bebeğinin yoğunlukları ışık değişimine göre farklılık gösterir.

2.4.3.1.3. Görünüm Tabanlı Yöntemler

Gözün şekli önemli bir tanımlayıcı olduğu gibi aynı şekilde gözün görünümü de önemli bir tanımlayıcıdır. Bu yöntemler aynı zamanda görüntü şablonu veya holistik yöntemler olarak da bilinir [96]. Görünüm tabanlı yöntemler gözü algılamak ve izlemek için gözün resimdeki görünümü kullanırlar. Görünüm tabanlı yöntemler hem uzamsal hem de dönüştürülmüş uzayda yürütülebilirler.

Holistik yöntemler yüz algılama ve yüz tanıma yöntemlerinde anlatıldığı gibi istatistiksel teknikleri kullanırlar. Genel olarak yüz algılamak için anlatılan avantaj ve dezavantajlar göz içinde geçerli olmaktadır. Bu tezinde göz tespiti modülü bu tip yöntemler kategorisine giren, OpenCV kütüphanesinin sağladığı, kaskad sınıflandırıcılar kullanılarak yapılmıştır. [108] nolu çalışma, bu tip yöntemleri kullanarak tasarlanmış göz tespiti uygulamasına örnek olarak verilebilir.

2.4.3.1.4. Melez Yöntemler

Melez yöntemler değişik göz algılama tekniklerinin üstün olduğu yönleri birleştirerek daha sağlam bir göz algılama tekniği oluşturmayı hedefleyen yöntemlerdir. Witzner v.d. bu yöntemleri kendi aralarında ikiye ayırmıştır [96]. Bunlar şekil ve yoğunluk tabanlı yöntemler ile şekil ve renk tabanlı yöntemlerdir.

Şekil ve görünüm tabanlı yöntemler şekil ve görünümün birleşiminden kararlı sonuçlar üretmeyi hedefleyen yöntemlerdir. Bu yöntemlerde genellikle modeller parçalara ayrılmıştır. Xie v.d. [109] parçalı bir model sunmuşlardır. Bu model mevcut bir şekil modeli ve çeşitli alt bileşenlerden oluşan bir modeldir.

Şekil ve renk tabanlı modeller de ise gözün renk bilgisinin çevresinden çok farklı olmasına rağmen çok düşük doğrulukta gerçekleşen sistemlerin doğruluğunu arttırmak için gözü konumlandırmada yardımcı bir bileşen kullanan yöntemleri ifade eder. Wintzer v.d. renk modeli kullanarak [110] kaba-ölçekli izleme ve AAM (Active Appearance Model) kullanmış ve gözün konumunu kesinleştirmek gri bir AAM kullanmışlardır.

2.4.3.1.5. Diğer Yöntemler

Simetri operatörleri, zamansal bilgiyi kullanan yöntemler ve IR (kızılötesi) ışıklandırma ile elde edilmiş görüntülerden göz tespiti bu gruba dâhil edilebilir.

Kızılötesi ışıklandırma ile elde edilen göz görüntüleri için genellikle 780-880 nm dalga boylarında ışık kaynağı kullanılır. Bu dalga boylarında piyasada ticari ürünler olup bu dalga boyu insan gözü tarafında görülmez ve göz bebeğine zarar vermez. [14] nolu çalışmada kullanılan iki IR filtreden geçirilmiş iki kamera görüntüsünün birleştirilerek tasarlanmış sistem buna örnek olarak verilebilir. Bu tip yöntemler basit olması ve kesin sonuca yakın sonuç üretmeleriyle tercih sebebi olmaktadır. Fakat donanım gereksinimi uygulanabilirlik açısından soru işaretidir.

Simetri özelliği yüz ve göz algılama için kullanılan bir değerlendirme yöntemidir. Reisfeld ve Yeshurun genel simetri dönüşümü tanımlamışlardır [111]. Reisfeld ve Yeshurun çalışmalarında yerel radyan simetri ve yüksek kontrastlı bölgeleri belirlemişlerdir. Loy ve Zelinsky [112] hızlı radyan simetri dönüşümünü sunmuşlardır. Gofman ve Kiryati [113] genel bir eniyileme yöntemi önermişlerdir. Simetri operatörünün göz tespitinde kullanılması için özellikler için bir eşik değerinin seçilmesi gerekmektedir.

2.4.3.2. OpenCV ve Göz Tespiti

Yüz tespiti kısmında anlatıldığı gibi OpenCV’de nesne/göz algılamak için yaygın olarak kaskad sınıflandırıcılar kullanılmaktadır. Mevcut DDDS uygulamasında da Haar özellikleri giriş olarak alan ve daha önceden eğitilmiş kaskad sınıflandırıcı modeli kullanılmıştır.

2.4.4. Uyku Sürüş Algılama Sistemleri

Görünüm tabanlı, DDDS sistemlerinde sürücü görüntüsünden alınan bir veya birkaç parametre değerlendirilerek sürücü, yorgunluk seviyesi için çıktı üretilir. Tek parametrelili ve çok parametrelili sistemlerle sistem gerçekleştirilebileceği gibi en çok kullanılan parametrenin PERCLOS olduğu bildirilmiştir[14].

DDDS uygulamalarında giriş kısmında da değinildiği gibi genellikle bilgisayarlı görü teknikleri kullanılmasına rağmen farklı tekniklerde kullanılabilir. Örneğin [24]'de kızıl ötesi algılayıcıdan gelen baş hareket değerlendirilerek çıkış üretilmiştir. [24]'deki yöntem probleme farklı bir bakış açısı getirmesine rağmen 2 boyutta çalışıldığı için sistemin başarımı doğal olarak çok düşüktür.

Uykulu sürücü belirlemede en güvenilir parametrenin PERCLOS olduğu [25] belirtilmiştir. [1]'de OpenCV kullanılarak yüz bulunmuş ve göz kırpma sayısı uykululuk için referans alınmıştır. [20]'de insan YCbCr renk uzayı dönüşümü kullanılarak sistem gerçekleştirilmiştir. [21]'de mevcut çalışmalara ek olarak YCbCr renk dönüşümünden faydalanılarak göz bulunmuş, göz takibi için kalman filtresi kullanılmıştır. [11-13]'da göz kapanması ve baş hareketlerinin analizinden farklı olarak uykululuk halinde yüzle ilgili ifadeler ve hangi yüz ifadelerinin daha belirgin olduğunun tespiti üzerine çalışılmıştır. [22]'de iris tanıma ile sürücüyü yetkilendiren, göz kapalılığı ve kızıl ötesi algılayıcılarla vücut sıcaklığı ölçülerek uykululuğun tespit edildiği melez bir çalışma sunulmuştur. [23]'da [22]'a benzer olarak bilgisayarlı görüyle gözden gelen görünüm bilgisi ve parmağın algılanan nabızla karar alabilen melez bir yapı önerilmiştir. [27]'de iki kızıl ötesi filtre kullanılarak gözden yansıyan ışık farkı ile göz tespitini yapan bir yapı önerilmiştir. [14]'da FPGA tabanlı, retinadan yansıyan ışığın yoğunluğunun farklı olacağı esasına dayanarak gözün bulunması ve göz kırpma oranına göre sonuç üreten bir yöntem sunulmuştur.

2.4.4.1. Uykulu Sürüş Algılama Sisteminde Çıkış Üretilmesi

Gerçek zamanlı koşacak DDDS uygulamasında hız çok önemli olmaktadır. Bildirilen çalışma sonuçları değerlendirilerek DDDS sisteminde PERCLOS'un kullanılmasına karar verilmiştir. Efektif işlem kabiliyetlerinden dolayı yüz ve göz tespiti için kaskad sınıflandırıcı kullanılmıştır. Aşağıdaki sebeplerden dolayı göz durumu tespit uygulaması için özellik çıkarıcı olarak LBP, sınıflandırıcı olarak LBPH kullanılmıştır:

- LBP resmin yerel yapılarını efektif olarak temsil eden parametresiz bir tanımlayıcıdır. Bu yöntem aslında doku analizi için sunulmasına geniş bir alanda kullanılmaktadır.

- LBP yöntemi etkinliğiyle son yıllarda ilgi odağı olmuştur. Temelde doku analizi için tasarlanmış olmasına rağmen günümüzde yüz analizi, resim ve video alma, çevre modelleme, görsel denetim, hareket analizi, biyomedikal-havadan görüntü analizi ve uzaktan algılama gibi değişik alanlarda kullanılır. Buda yöntemin başarımı için iyi bir referans sunar.
- Parametresiz bir yöntem olup, rotasyona ve ışıklandırma problemlerine karşı kararlı sonuçlar üretmesiyle öne çıkmıştır.
- LBP yönteminde gri seviyede işlemler yapılır. Giriş resmi renkli ise gri düzeye çevrilir daha sonra ise yerel komşuluk değerlendirilerek ilgili piksel değeri için LBP değeri belirlenir. Gri seviyede çalışmak işlem karmaşıklığını azaltır.
- LBP yöntemi tek çeşit ışıklandırma değişikliklerini tolere eder ve gerçekleşmesi basittir.
- LBP yöntemi PCA ve LDA gibi eğitim aşamasında bir nesneye ait birden çok farklı görünüme sahip resim örnekleri gerektirmez. Dolayısıyla modeli oluşturmak daha kolaydır.
- Modeli oluşturmak için tek eğitim verisi yeterlidir dolayısıyla gerçekleşmesi basittir.

OpenCV kütüphanesindeki LBP modeli, yüz tanıma algoritmalarına tek şekilde erişimi sağlayan soyut FaceRecognizer sınıfından türetilir. Aşağıda FaceRecognizer sınıfı verilmiştir :

```
class FaceRecognizer : public Algorithm {
public:
    // Sanal yıkıcı
    virtual ~FaceRecognizer() {}
    // Model eğitim fonksiyonu
    virtual void train(InputArray src, InputArray labels) = 0;
    // Model güncelleme fonksiyonu
    virtual void update(InputArrayOfArrays src, InputArray labels);
    // Model'den bir tahmin alır
    virtual int predict(InputArray src) const = 0;
    // verilen bir etiket ve resim için doğruluk tahmin eder
    virtual void predict(InputArray src, int &label, double &confidence) const = 0;
```

```

// Modeli kaydeder
virtual void save(const string& filename) const;
// Modeli yükler
virtual void load(const string& filename);
// cv:: FileStorage ile verileni kaydet
virtual void save(FileStorage& fs) const = 0;
// cv:: FileStorage ile verileni yükle
virtual void load(const FileStorage& fs) = 0;
};

```

Model şu şekilde oluşturulur:

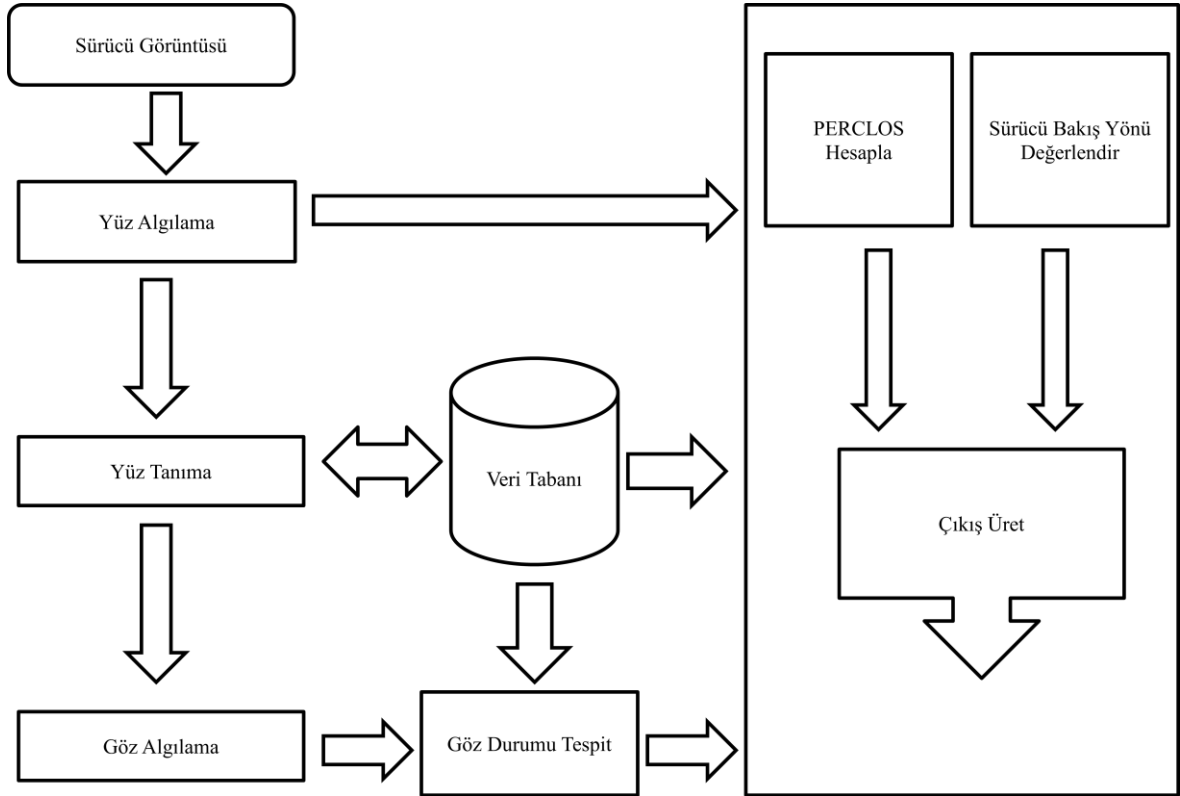
```

Ptr<FaceRecognizer> createLBPHFaceRecognizer(int radius=1, int neighbors=8, int
grid_x=4, int grid_y=4, double threshold=DBL_MAX).

```

3. DDDS UYGULAMASI

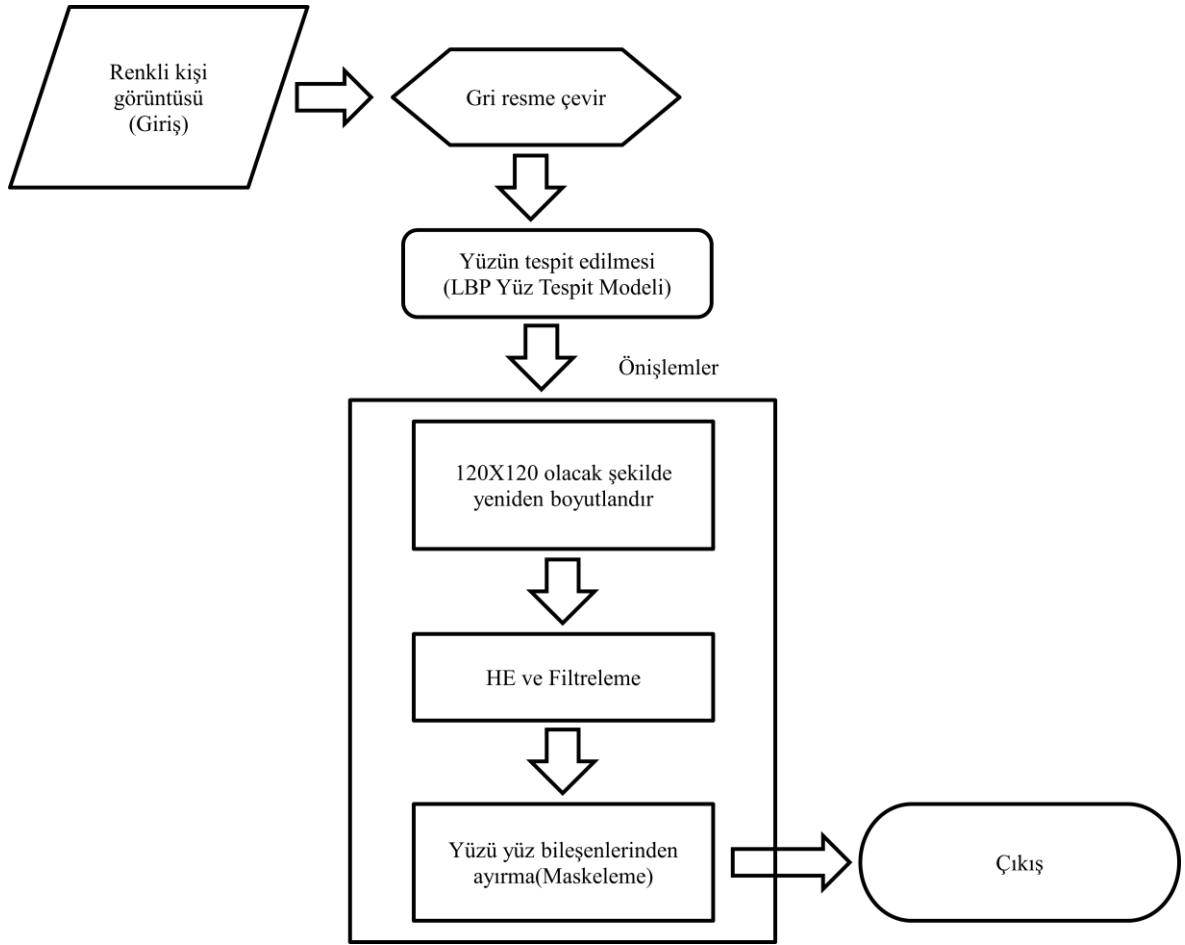
Bu tez kapsamında gerçekleştirilen DDDS modelinin blok diyagramı Şekil 3.1’de verilmiştir. Temel olarak yüz algımla, yüz tanıma, göz algılama ve göz durumu tespit olmak üzere 4 modülden oluşan modelin her modülü ayrı ayrı verilecektir.



Şekil 3.1. Tasarlanan DDDS modeli blok diyagramı

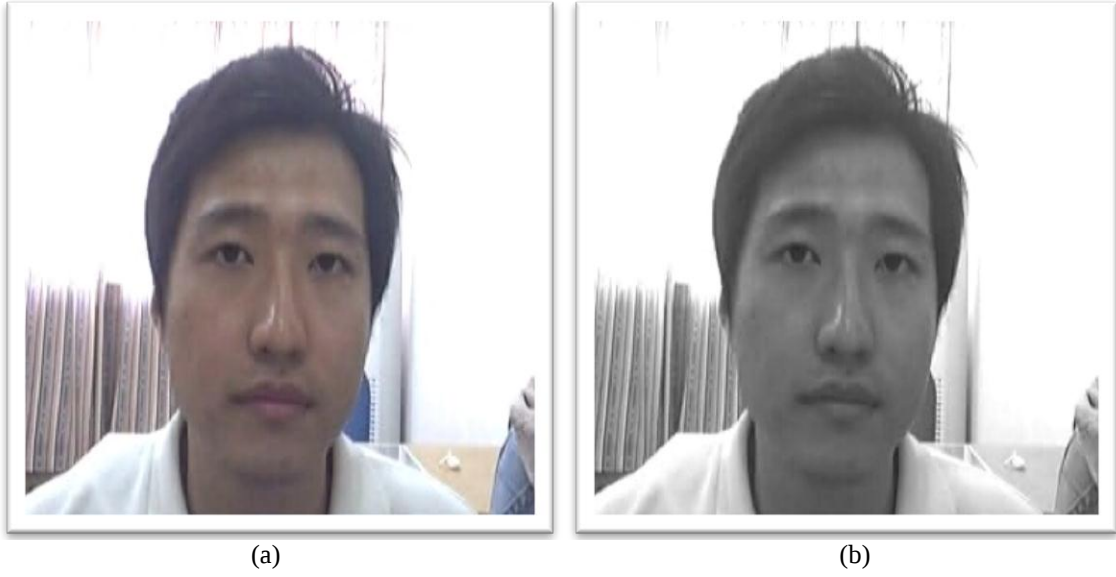
3.1. Yüz Tespit Uygulaması

Yüz tespit modülünün sonucu yüz tanıma modülüne ve göz konumu tespit modüllerine giriş verisi olmaktadır. Dolayısıyla bu modülün doğru ve etkin bir şekilde icra edilmesi sistem doğruluğunu direkt etkileyecektir. DDDS uygulamasındaki test verileri için ZJU [94] veri tabanı kullanılmıştır. Bu veri tabanında 320x240 boyutlarında kaydedilmiş renkli resimler bulunmaktadır. Yüz tespit uygulaması için OpenCV kütüphanesi içinde bulunan kaskad sınıflandırıcı kullanılmıştır. Gerçeklenen yüz tespit uygulamasının blok diyagramı Şekil 3.2’de verilmiştir.



Şekil 3. 2. Tasarlanan yüz tespit uygulaması blok diyagramı

Sistemi gerçekleştirmek için ilkin video dosyasından ardışık olarak kareler sisteme giriş verisi olarak alınmaktadır. Renkli resimler daha sonra gri düzeye çevrilerek işlemler gri düzeyde devam edilir. ZJU veri tabanında 20 adet farklı video dosyası bulunmaktadır. Bundan sonra gösterilecek uygulama örnek ve sonuçlarında “000006M_FNNavi” video dosyasının ilk karesi kullanılacaktır. Şekil 3.3’de örnek video dosyası ilk karesi için renkli hali ve gri düzeye dönüştürülmüş hali mevcuttur. Gri düzeye dönüşüm işlemi için “OpenCV cvtColor(InputArray src, OutputArray dst, CV_BGR2GRAY)” fonksiyonu kullanılmıştır.



Şekil 3.3. DDDS Uygulaması girişleri için örnek renkli (a) ve gri kareler (b)

Yüz konumlandırma işlemi için OpenCV kütüphanesinde tanımlı kaskad sınıflandırıcı(CascadeClassifier) kullanılmıştır. Kaskad sınıflandırıcı genel olarak video görüntülerinde nesne tanıma işlemi için kullanılmaktadır. Hem Viola-Jones yöntemindeki Haar özellikleri hem de LBP için kullanılabilir. Kaskad sınıflandırıcı “load” fonksiyonu ile yüklenilir. Mevcut DDDS uygulamasında karşıdan alınmış görüntüler için algılamayı gerçekleştiren “lbpcascade_frontalface.xml” sınıflandırıcısı kullanılmıştır. Tanıma işlemi detectMultiScale fonksiyonu tarafından icra edilir. İlgili fonksiyon şu şekildedir; void CascadeClassifier::detectMultiScale(const Mat& image, vector<Rect>& objects, double scaleFactor=1.1, int minNeighbors=3, int flags=0, Size minSize=Size(), Size maxSize=Size())

Bu fonksiyon için parametreler şu şekildedir:

- image – Tespit edilecek bir nesne için nesne görüntüsünün piksel bilgisini içeren CV_8U tipinde matristir.
- objects – Tespit edilen her nesne için konum ve boyut bilgisini içeren dikdörtgenler vektörü.
- ScaleFactor – Arama işlemi için her görüntü boyutu azaltılmasında azaltma işleminin ne kadar olacağını belirtir.
- minNeighbors – Aday dikdörtgenler için kullanılmak üzere doğru tanıma işlemi için en az kaç tane komşusu olması gerektiğini belirtir.

- flags - cvHaarDetectObjects fonksiyonunda olduğu gibi. Bu yeni bir kaskadlar için kullanılmaz. Geriye dönük uyumluluk için muhafaza edilmiştir.
- minSize – Algılanacak nesnenin minimum boyutunu belirtir. Daha küçük nesneler göz ardı edilir.
- maxSize - Algılanacak nesnenin maksimum nesne boyutudur. Daha büyük nesneler göz ardı edilir.

Mevcut DDDS uygulamasında yüz algılama işlemi ilgili parametreler şöyle uygulanmıştır; minSize = (100,100), maxSize = (240,240) searchScaleFactor =1.1f ve minNeighbors = 3 olarak alınmıştır. Geri kalan parametreler varsayılan değerindedir.

DDDS uygulamasında kamera sürücünün tam karşısında olacağı için sürücü yüzü doğal olarak tespit edilecek en büyük yüz olacaktır. DDDS uygulamasında en büyük yüzün sürücü yüzü olduğu varsayılarak yüz tanıma uygulamasına ve göz tespit uygulamasına bu tespit edilen en büyük yüz giriş verisi olarak verilmiştir.

3.1.1. Yüz İçin Ön İşlemler

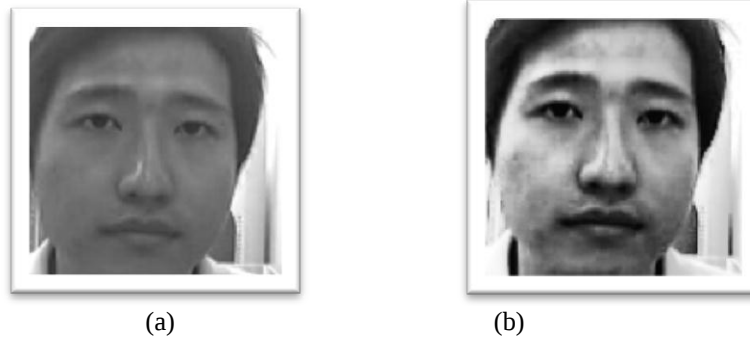
Konumlandırılan yüz, yüz tanıma aşamasına geçirilmeden önce bir dizi ön işleme tabi tutulmuştur. Bu ön işlemler sonucunda sistemin değişken şartlarda kararlı sonuçlar vermesi için mevcut veri beklenen veriye yaklaştırılır. Bu şekilde sistemin kararlı sonuçlar alması hedeflenmiştir.

3.1.1.1. Yeniden Boyutlandırma

Yüz tanıma modeli sabit boyutlu verilerle eğitildiğinden dolayı mevcut sistemin giriş verisinde elde edilen yüzlerinde modelin eğitim verisinde kullanılan resimlerle aynı boyutta olması gerekir. Tespit edilen yüzlerin boyutu 120x120 olacak şekilde yeniden boyutlandırılmıştır. Bu işlem için; void resize(InputArray src, OutputArray dst, Size dsize, double fx=0, double fy=0, int interpolation=INTER_LINEAR)' fonksiyonu kullanılmıştır.

3.1.1.2. Yüzün Sağ ve Sol Kısmı İçin Histogram Eşitleme

Mevcut bilgisayarlı görme problemlerinin çoğunda özellikle DDDS sisteminde ışık kaynağının yönü görüntü üzerinde gürültülere sebep olmaktadır. Işığın geliş yönüne göre yüz üstünde gölgeler oluşmaktadır. Bu problemi çözmek için [92] nolu kaynakta belirtildiği gibi tespit edilen yüz bölgesinin sağ ve sol kısmına ayrı ayrı histogram eşitleme işlemi uygulanmıştır. Yüzün iki tarafına ayrı ayrı bu işlem uygulandığında resmin ortasında keskin bir kenar oluşur bu istenmeyen durumu çözmek için sol ve sağ histogramlar eşitlendikten sonra resmin tamamının (yüzün) histogramı bir daha eşitlenir. En son aşamadan bu 3 histogram harmanlanarak tek histogram elde edilir. Bu işlem sonucunda yüzün sol ve sağ kısmı için ışıklandırma ve kontrast iyileştirilmiş olur. Şekil 3.4’de DDDS sistemi test verisi için histogram eşitlemeden önce ve sonra örnek bir resim verilmiştir.



Şekil 3.4. DDDS için örnek tespit edilen yüz (a) ve histogramı eşitlenmiş yüz (b)

3.1.1.3. Yüz Filtreleme

Tasarlanan sistemin kararlı sonuçlar üretmesi için eğitim verisi ve sitsem giriş verisi OpenCV fonksiyonlarından bilateralFilter filtre kullanılarak ilgili resim yumuşatılarak görüntüdeki gürültüler giderilmiştir. OpenCV bilateralFilter fonksiyonu şu şekildedir; void bilateralFilter (InputArray src, OutputArray dst, int d, double sigmaColor, double sigmaSpace, int borderType=BORDER_DEFAULT).

DDDS uygulaması için filtre için kullanılan fonksiyon parametreleri şöyledir; $d=0$, $\sigma_{\text{Color}} = 20.0$, $\sigma_{\text{Space}} = 2.0$ olarak belirlenmiştir. Şekil 3.5'te şekil 3.5.'te verilmiş resim filtrelenmiş hali görülmektedir.



Şekil 3.5. Filtrelenmiş yüz

3.1.1.4. Yüzün Yüz Bileşenlerinden Ayrıştırılması

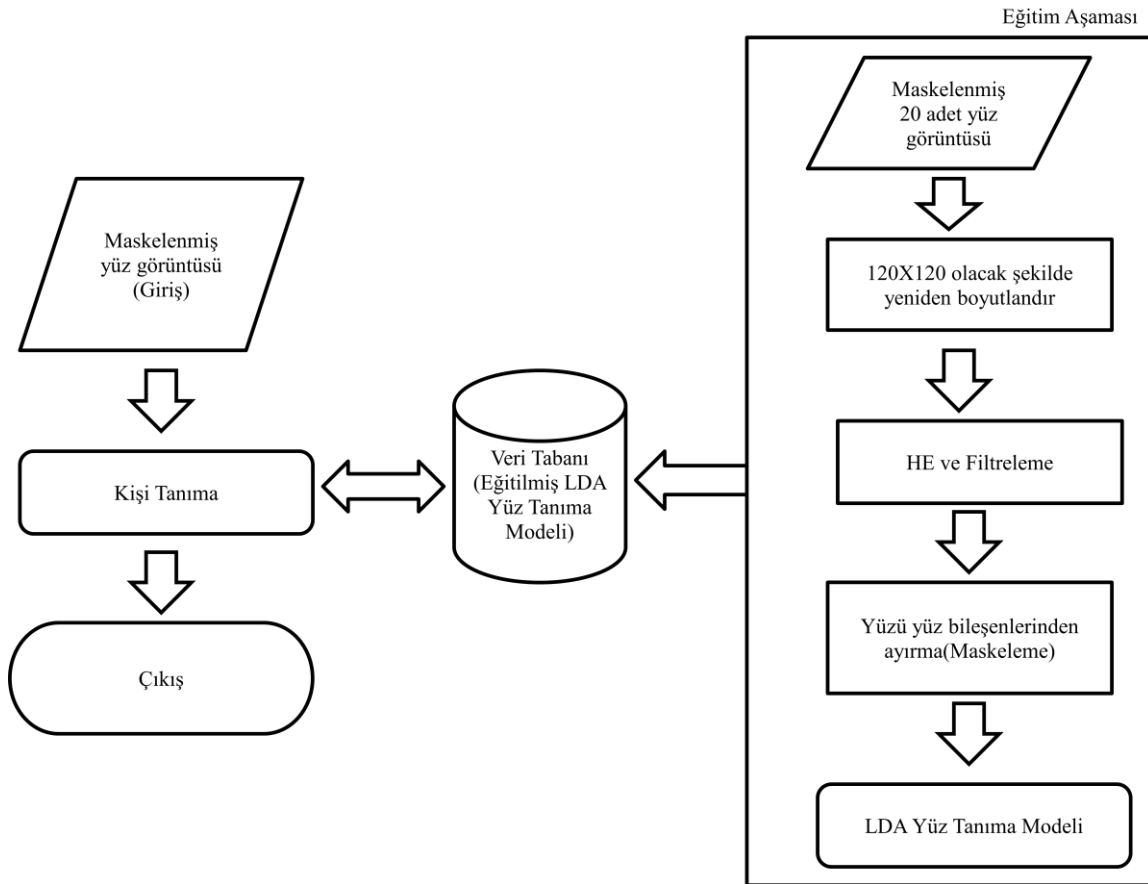
İnsan yüzü, insanlar için her ne kadar ayırt edici olursa olsun aynı kişiye ait yüzde bile birçok değişik görünüm olabilmektedir. Bu görünümün çevresel olabilmekle birlikte kişinin kendisinden kaynaklanan durumlarda belirebilir. Kişi için en değişken olabilecek saç tipi ile sakalın varlığı ve yokluğuna göre yüzün tanınması güçleşir. Bu problemi çözmek için tespit edilen yüz bölgesi sakal, saç ve arka planı elemine edecek bir maske ile maskelenmiştir. Bu işlem için OpenCV kütüphanesiyle ilkin bir maske tanımlanmıştır. Bu maske tüm verileri sıfır olarak atanmış bir veri tipidir. Maskenin bu değeri 0-255 arasında herhangi bir sayı olabilir. Fakat bu uygulamada hız önemli olduğu için işlemlerin daha hızlı icra edilmesi için maske içeriği sıfır olarak atanmıştır. Maskeyi yüze uygulamak için bir elips tanımlanır ve yüz bölgesi ile maske katlanır. Elips'in üst kısımlarına maske verisi olan sıfır atanırken iç kısmı ise yüz olarak etiketlenmiş bölgenin değerlerini içerir. Elipsin merkezi, programda bir sabitle yüzün üst kısmına çekilerek değişkenliği yüksek olan çene bölgesi yüz bölgesinden elemine edilmiştir. Şekil 3.6'da maskelenmiş yüz örneği verilmiştir. Bu şekilde elde edilen veri eğitim aşamasında yüz tanıma sisteminin eğitim verisi olurken icra aşamasında ise yüz tanıma probleminin giriş verisi olmaktadır.



Şekil 3.6. Maskelenmiş yüz

3.2. Yüz Tanıma Uygulaması

Yüz tanıma uygulaması için Fisher yüz tanıma modeli kullanılmıştır. Modelin başlangıç parametreleri şu şekilde seçilmiştir: $\text{minFeatureSize} = \text{Size}(100,100)$, $\text{maxFeatureSize} = \text{Size}(240,240)$, $\text{searchScaleFactor} = 1.1f$ ve $\text{minNeighbors} = 3$ olarak seçilmiştir. Şekil 3.7’de tasarlanan yüz tanıma uygulamasının blok diyagramı verilmiştir.



Şekil 3.7. Tasarlanan yüz tanıma uygulaması blok diyagramı

Gerçeklenen yüz tanıma modeli ZJU veri tabanı için eğitilmiş olup aynı veri tabanında test edilmiştir.

Daha önce de değinildiği gibi LDA modelinde ilgili eğitim aşamasındaki test verileri ile sistemin girişindeki resimlerin aynı boyutlu olması gerekir. Test verileri oluşturulurken gerçekleştirilen yazılımla tüm resimler 120x120 olacak şekilde yeniden ölçeklendi. Aynı şekilde sistem girişi yüz tespit modülünün girişi olduğu için ilgili ön işlemlerden geçip maskelenmiş yüz görüntüsü eğitim aşaması ve karar aşamasının giriş verisi olmaktadır. Yüz tanıma modeli için giriş verisi, elde edilen veriler içinden manuel seçildi. Her bireysel kişi için 20 adet test verisiyle sistemin kişiyi tanıması hedeflendi. Sistemin çıkışı, sürücü kimliği hakkındaki tahmindir. Şekil 3.8’de ZJU veri tabanındaki altıncı kişi için oluşturulan test verileri görülmektedir.

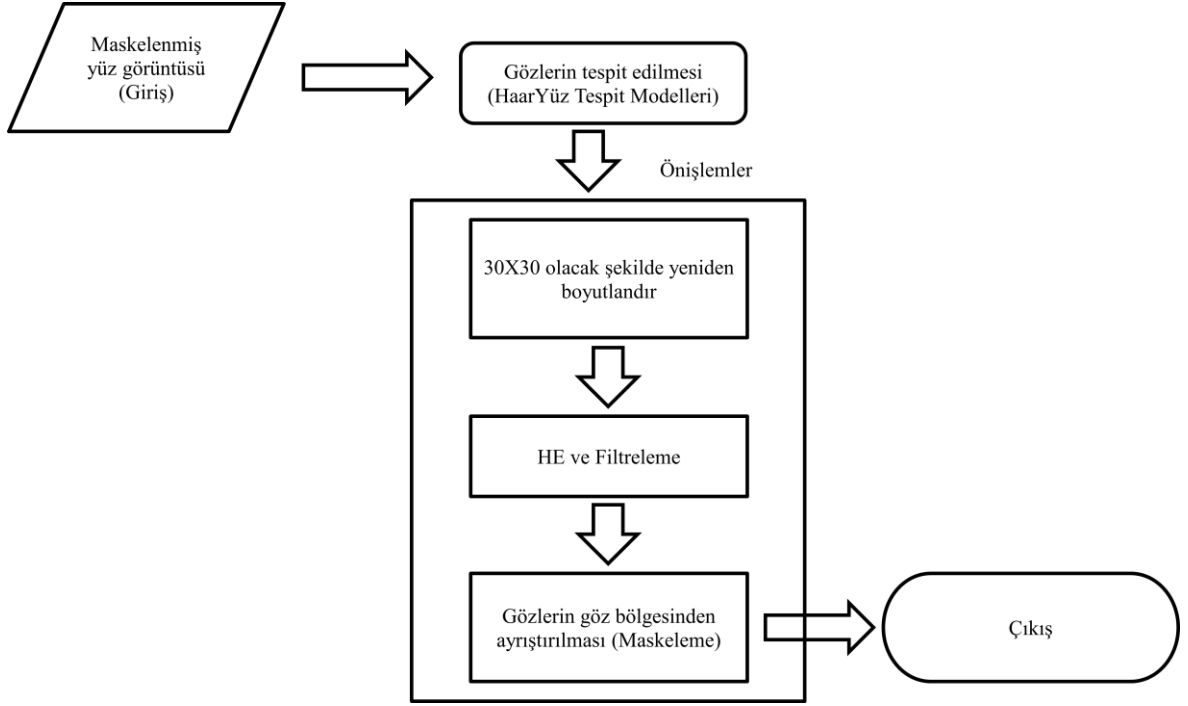
DDDS modelinin bu aşamasında sürücünün tanınması hedeflenir, sürücü tanındıktan sonra sürücü hakkındaki bilgiler kullanılarak problem uzayı küçültülebilir. Örneğin kişinin göz açıklığı öğrenilerek daha sağlam sonuçlar elde edilebilir veya kişinin göz bölgesi için tek sistem eğitilerek örnek uzayı küçültülebilir. Şimdiye kadar ki çalışmalarda hiç araştırılmamış olan PERCLOS’un kişiden kişiye değişip değişmediği sürücünün kimliği belirlendikten sonra araştırılabilir. Yani kişi tanındıktan sonra ilgili kişi için ortalama saniyedeki göz kırpma sayısı bulunabilir. Bu şekilde daha sağlam karar alabilen sistemler tasarlanabilir. Çünkü herkes için tek aşik değeriyle sonuç üreten sistemler yerine kişiye özel sonuç üreten sistemlere göre daha az başarılı olur.



Şekil 3.8. Tasarlanan yüz tanıma uygulamasında elde edilen eğitim verisi örneği

3.3. Göz Algılama Uygulaması

Yüz tespit modülünün sonucu yüz tanıma modülünde giriş verisi olduğu gibi göz algılama modülünün de giriş verisi olmaktadır. Tasarlanan göz tespit uygulamasının blok diyagramı Şekil 3.9’da verilmiştir.



Şekil 3.9. Tasarlanan göz tespit uygulamasının blok diyagramı

Göz tespit uygulaması için OpenCV kütüphanesi içinde bulunan kaskad sınıflandırıcı kullanılmıştır. Sistemin girişi yüz tanıma modülünün girişine benzer olarak, yüz tespit modülünün çıkışı olan maskelenmiş, saç-sakal ile arka plan elemine edilmiş, yüz resmi olacaktır. Şekil 3.10'da göz algılama uygulaması girişi olan maskelenmiş yüz örneği verilmiştir.



Şekil 3.10. Göz algılama girişi

Göz konumlandırma işlemi için OpenCV kütüphanesinde tanımlı kaskad sınıflandırıcı kullanılmıştır. Kaskad sınıflandırıcı daha önce değinildiği gibi genel olarak nesne görüntülerinden nesne tanıma işlemi için kullanılmaktadır. Bu tezin uygulamasında

geliştirilen DDDS projesinde, konumlandırılmış yüzden göz tespiti için iki adet kaskad sınıflandırıcı kullanılmıştır. Yani sağ ve sol göz ayrı ayrı algılanmıştır. Bu işlem için OpenCV kütüphanesi içinde “data” klasöründe bulunan önceden tanımlanmış kaskad göz tespit modelleri kullanılmıştır. Sol göz tespiti için “Haarcascade_lefteye_2splits.xml” ve sağ göz tespiti için “Haarcascade_righteye_2splits.xml” modelleri kullanılmıştır. Tanıma işlemi detectMultiScale fonksiyonu tarafından icra edilir. İlgili fonksiyon parametreleri şöyle uygulanmıştır; minFeatureSize=(30,30), maxFeatureSize =(39,39), searchScaleFactor =1.03f ve minNeighbors = 3 olarak alınmıştır. Geri kalan parametreler varsayılan değerindedir.

Mevcut göz tanıma işleminin sonucu olarak tek yüz için bile farklı ölçeklerde arama yapıldığı için birden fazla göz bulundu sonucu çıkabilmektedir. Tasarlanan DDDS uygulamasında gözün konumlandırılmasında bir hata olmaması için en büyük aday göz bölgesi, aranılan göz olarak alınıp DDDS sonuç kısmına giriş verisi olarak verilmiştir. Bundan sonra gösterilecek uygulama örnek ve sonuçlarında “000006M_FNNavi” video dosyasının ilk karesi kullanılacaktır. Şekil 3.11’de örnek kare için yüz bölgesinde bulunan sol ve sağ gözler gösterilmiştir.



Şekil 3.11. Tespit edilen göz, sağ (a) ve sol (b) örneği

3.3.1. Göz İçin Önışlemler

Aday göz bölgeleri içinde konumlandırılan göz, DDDS sonuç aşamasına geçirilmeden önce biz dizi önışleme tabi tutulmuştur. Bu önışlemler sonucunda sistemin değişken şartlarda kararlı sonuçlar vermesi için mevcut veri beklenen veriye yaklaştırılmıştır. DDDS modelinde sağ ve sol göz ayrı ayrı bulunduğu için ilgili işlemler sağ ve sol göz için bir birinden bağımsız olarak yapılmıştır.

3.3.1.1. Yeniden Boyutlandırma

Göz algılama modeli sabit boyutlu verilerle eğitildiğinden dolayı mevcut sistemin giriş verisinden bulunan aday göz bölgeleri ile modelin eğitim verisinde kullanılan resimlerle aynı boyutta olması gerekir. Tespit edilen sağ ve sol gözlerin ayrı ayrı olarak boyutu 30x30 olacak şekilde yeniden boyutlandırılmıştır. Bu işlem yüz algılama modülündeki gibi yapılmıştır.

3.3.1.2. Göz Bölgesi İçin Histogram Eşitleme

Histogram eşitleme işlemi yüz algılama modülündeki gibi gözün her iki yarısı için ayrı ayrı yapılmamış olup OpenCV kütüphanesinin sağladığı standart Histogram eşitleme fonksiyonu kullanılarak yapılmıştır. Şekil 3.12’de göz çifti için histogramı eşitlenmiş resim örnekleri verilmiştir.



Şekil 3.12. HE yapılmış sağ (a) ve sol (b) göz örneği

3.3.1.3. Göz Filtreleme

Yüz algılama modülündeki bilateralFilter fonksiyonu ile filtreleme bu modül için kullanılmamıştır. Onun yerine histogram kuantalama işlemi yapılmıştır. LBP kodekleri tek piksel değerindeki tek değişimde bile değişik sonuç vereceğinden dolayı algılanacak açık ve kapalı göz deseninin daha genel ifade edilebilmesi için aday göz bölgelerinin tüm piksel değerleri 32 ile bölünüp elde edilen tam sayı tekrar 32 ile çarpılarak histogram kuantalama işlemi yapılmıştır. Normalde piksel değerlerini 32 ile çarpmaya gerek yoktur fakat ilgili gri düzey değerleri çok düşük olacağından dolayı insan gözü aradaki farkı algılamayabilir. Yani sonuçların görünebilmesi için öteleme işlemi yapılmıştır. Bu şekilde sistemin başarımının artırılması hedeflenmiştir. Şekil 3.13’te bu şekilde filtrelenmiş göz çifti örnekleri verilmiştir.



Şekil 3.13. Filtrelenmiş göz, sağ (a) ve sol (b) örneği

3.3.1.4. Gözün Ayırıştırılması

Göz bölgesindeki kaş ve göz bölgesi komşuluğundaki gözle alakası olmayan bölgelerin karar sürecini etkilememesi için göz bölgesi maskelenmiştir. Bu işlem için OpenCV kütüphanesiyle ilkin bir maske tanımlanmıştır. Bu maskenin tüm verileri yüz tespitinde anlatıldığı gibi hesaplama verimliği açısından sıfır olarak atanmıştır. Maskeyi yüze uygulamak için bir çember tanımlanıp yüz bölgesi ile maske katlanmıştır. Çemberin üst kısımları maske verisi olan sıfır olarak atanırken iç kısmı ise göz olarak etiketlenmiş bölgenin değerlerini içerir. Çemberin merkezi, kaş bölgesini elemek için geliştirilen uygulamada aday göz bölgesi yüksekliğiyle orantılı bir değişkenle aday göz bölgesinin alt kısmına ötelenmiştir. Şekil 3.14’de maskelenmiş göz örneği verilmiştir. Bu şekilde elde edilen veri eğitim aşamasında göz kırpma sisteminin eğitim verisi olurken icra aşamasında ise göz kırpma sisteminin giriş verisi olmaktadır.

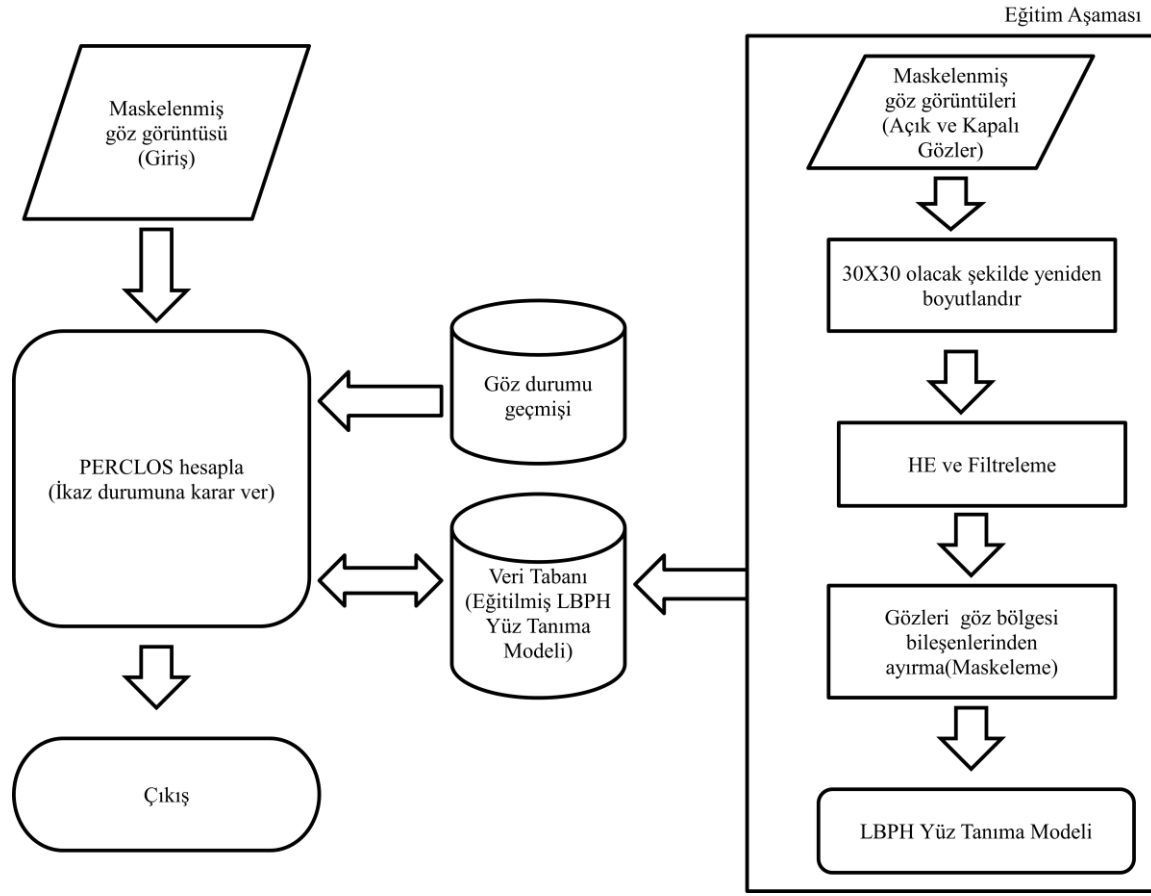


Şekil 3.14. Maskelenmiş sağ (a) ve sol (b) göz örnekleri

3.4. Geliştirilen Göz Durumu Tespit Uygulaması

DDDS uygulamasında karar sürecinde PERCLOS kullanıldığı için göz kırpmasının tespit edilmesi gerekir. Bu işlem için LBPH yüz tespit modeli göz kırpmasını algılayacak

şekilde eğitilmiştir. Geliştirilen göz kırpmasının tespit edilmesi uygulması blok diyagramı Şekil 3.15'te verilmiştir.



Şekil 3.15. Tasarlanan göz kırpması ve PERCLOS hesaplama uygulması blok diyagramı

Göz durumu tespit uygulaması için özellik çıkarıcı olarak LBP, sınıflandırıcı olarak LBPH kullanılmıştır. Model şu şekilde oluşturulur:

```
Ptr<FaceRecognizer> createLBPHFaceRecognizer(int radius=1, int neighbors=8, int grid_x=4, int grid_y=4, double threshold=DBL_MAX).
```

Bu modelde eğitim ve karar süreçlerinde algoritmik temelleri yüz tespiti kısmında anlatılan ELBP örüntüleri kullanılır. Bu model Fisher modelinden farklı olarak güncellemeye izin verir. Yani icra zamanında modelin referans (eğitim) verileri değiştirilebilir.

Modelin parametreleri şu şekildedir:

- radius: ELBP örüntüleri oluşturmak için kullanılacak yarıçap.

- neighbors: ELBP örüntüleri oluşturmak için kullanılacak örnek sayısı. Yüksek örnek sayısı yüksek işlem maliyetine sebep olur.
- grid_x: Düşey yöndeki hücre sayısıdır. 8 tavsiye edilen değerdir. Bu tez çalışmasında 4 alınmıştır. Çünkü göz bölgesi küçüktür. Daha çok hücre yani ince ızgaralar ortaya çıkan özellik vektörlerinin boyutunu artırır.
- grid_y: Yatay yöndeki hücre sayısıdır. 8 tavsiye edilen değerdir. Bu tez çalışmasında 4 alınmıştır. Çünkü göz bölgesi küçüktür. Daha çok hücre yani ince ızgaralar ortaya çıkan özellik vektörlerinin boyutunu artırır.
- threshold: Eşikleme tahmine uygulanır. Eğer ölçülen en küçük uzaklık eşik değerinden büyükse model tahmin olarak -1 döndürür. Yani giriş tanınmamış olur.

Eğitim aşamasında açık ve kapalı gözler için eğitim resmi veya resimleri seçilerek gözün kapalı veya açık olduğu etiketlenir. Etiketlenen bu veriler modele verildiğinde model ilgili gözlerle açık ve kapalı durumu için eğitilmiş olur. Şekil 3.16’da açık ve kapalı göz için eğitim seti örnekleri verilmiştir. Tahmin aşamasında ise herhangi bir göze ait veri sisteme verilerek sistemin daha önce öğrendiği verilere göre sonuç üretmesi beklenir. Geri dönen sonuç gözün açık veya kapalı olduğu karardır.



Şekil 3.16. Göz durumu tespiti modeli için açık (a) ve kapalı (b) göz eğitim verisi örnekleri

Bu aşamada göz algılama kısmında anlatıldığı gibi önışlemlerden geçirilmiş göz bölgeleri bu aşamanın giriş verisi olmaktadır. Sistemin çıkışı, gözün kapalı veya açık olduğu bilgisi hakkındaki tahmindir. DDDS modelinin bu aşamasında gözün kapalı veya açık olduğu kararı verilir.

Göz durumu belirlendikten sonra ilgili PERCLOS hesaplanarak ve sürücünün yola bakıp bakmadığına göre sistemin çıkışı belirlenir. Göz kırpması için bir önceki göz görüntüsünün açık, mevcut göz durumun ise kapalı olduğu durumda kırpma işlemi gerçekleşmiş olur. Video görüntüleri saniyede 30 kare yakaladığından dolayı bir kırpma işlemine ait birden çok göz kapalı durumu sonucu çıkabilir. Yanlış kırpma oranı hesaplamanın önüne geçmek için tasarlanan sistemde açık durumdan kapalı duruma geçiş göz kırpma olarak alınmıştır. Sürücünün son 3 dakikası değerlendirilerek PERCLOS

hesaplanır. Eđer belli bir eřik deęerinde gz kırpma iřlemi gereklenirse kiřinin uykulu olduęu sonucu ıkarılır. Yola bakıp bakmama durumu iin referans, yzn ve iki gzden birisinin bulunmasıdır. Bu belirlene bilen bir durumdur. nk ilgili yz belirleme alıřmasında sistem giriř grntleri iin yz tespiti ve gz tespiti yapar. Gzn ve yzn olmaması demek src bakıř aısının yola doęru olmaması anlamına gelir. Bu durumlarda srcnn yola bakmadıęı varsayılarak belli bir sre gemesi durumunda hala durum geerliyse srcy ikaz edecek bir durumun oluřtuęu varsayılır. Gereklenen sistem bu řekilde DDDS iin ikaz durumuna karar verir.

4. SONUÇLAR ve BULGULAR

Gerçeklenen DDDS uygulaması ZJU [94] veritabanında test edilmiştir. Uygulama sonuçları ve değerlendirme bu veri tabanı sonuçları üzerinden yapılmıştır.

ZJU veritabanı; 20 kişinin dört farklı durumda elde edilmiş toplam 80 adet renkli video dosyasını içermektedir. Birinci tip video kayıt yönteminde denekler gözlüksüz olup görüntüler karşıdan alınmıştır. İkincisinde denekler ince çerçeveli gözlük takıp görüntü önden alınmıştır. Üçüncü tip görüntü kaydında görüntüler tekrar önden alınmış olup denekler siyah çerçeveli bir gözlük takmışlardır. Son tip video kliplerinde denekler gözlüksüz olup görüntü aşağıdan yukarı doğru alınmıştır. Her denek tüm kliplerinde olağan hızda göz kırpma eylemi icra etmiştir. Tüm video klipler 320x240 çözünürlükte kayıt edilmiştir. Videolardaki göz kırpma sayıları bir ile altı arasında değişmektedir. İlgili veritabanında toplamda 255 göz kırpması vardır. Tüm videolar kapalı ortamda ışık kontrolünün olmadığı ve arka planın dinamik olduğu bir ortamda kayıt edilmiştir.

Mevcut uygulama; OpenCV kütüphanesi ve .Net 2010 platformunda Visual C++ dili kullanılarak gerçekleştirilmiştir. Uygulama Intel i5 işlemcili 4 GB paylaşımlı belleği olan Toshiba M450 2.4 GHz işlemci hızlı bir netbook ile test edilmiştir.

Gerçeklenen yazılımda öncelik ilgili nesnelerin yani gözün ve yüzün bulunmasına verilmiştir. Performans düşünülmeyle beraber en iyi oranlarda yüz ve göz bulunması için OpenCV kütüphanesinde bulunan en iyi modeller kullanılmış olup performansı azaltıcı etkileri ikinci plana atılmıştır. Ayrıca elde edilen görüntülerde gürültünün miktarına bakılmaksızın sağlam sonuçların elde edilmesi için her görüntü test ve eğitim aşamasında filtrelerden geçirilmiştir.

DDDS uygulamasıyla ilgili olarak temel olarak iki tip sonuç verilecektir. Mevcut uygulama gerçek zamanlı bir uygulama düşünülerek gerçekleştirildiği için performans ve sistem doğruluğu arasındaki ilişkiyi iyi belirlemek gerekir. Birinci tip uygulama sonucunda, bu sistemin modüllerinden yüz algılama ve göz algılama modülleri için farklı parametreler için test edilmiştir. Uygulama için test edilen bu sonuçta yüz algılama, göz algılama, yüz tanıma ve ortalama saniyede işlenen kare sayısı verilecektir. İkinci tip sonuçlarda dört modülden oluşan sistemin, her modülle ilgili sonuçları ayrı ayrı verilecektir. Ayrıca dört tip kayıt olduğundan dolayı her kayıt türü için ayrı ayrı sonuçlar verilecektir.

Yüz algılama işlemi için daha önce anlatıldığı gibi OpenCV kütüphanesindeki detectMultiScale() fonksiyonu kullanılmıştır. Bu fonksiyonun minFeatureSize parametresinin optimum değeri bu tez kapsamında araştırılmıştır. Geri kalan parametreler daha önce değinildiği gibi sabit tutulmuştur. Tablo 4.1’de minFeatureSize’in 80, 90 ve 100 seçildiği durumlar için yüz algılama ve yüz tanıma modülleri ile saniyede işlenen kare sayısı verilmiştir. Saniyede işlenen kare sayısı verisi elde edilirken bu test modülü verileri için göz algılama modülünde searchScaleFactor =1.1 olarak alınmıştır.

Tablo 4.1. Yüz tespit modülünün minFeatureSize parametresi için sistem çıktıları

	minFeatureSize = (80,80)	minFeatureSize = (90,90)	minFeatureSize = (100,100)
Tespit Edilen Yüz	%92.97	%92.97	%92.95
Tanınan Yüz	%100	%100	%100
Yanlış Tanınan Yüz	%0	%0	%0
İşlenen Kare Sayısı (fps)	11.17	11.02	21.67

Tablo 4.1’de görüldüğü gibi minFeatureSize parametresinin değişimi tespit edilen yüz ve tanınan yüz oranlarında fazla bir fark yoktur. Fakat saniyede işlenen kare sayısına bakıldığında bu parametreyi 100 almak performansı çok olumlu yönde etkilemiştir. Bu yüzden ilgili uygulamada bu parametre 100 olarak seçilmiştir. Bundan sonra verilecek sonuçlarda yüz algılama modülünün ilgili parametreleri bu tezde ilgili konu başlığında anlatılan değerlerdedir.

Göz algılama modülü için ilgili konu başlığında anlatıldığı gibi OpenCV fonksiyonlarından detectMultiScale() fonksiyonu kullanılmıştır. Optimum parametre seçimi için minNeighbors ve searchScaleFactor parametreleri göz tespiti için ayrıntılı olarak incelenmiştir. Çünkü göz durumu tespiti için yani DDDS’nin çıktı üretebilmesi için mutlaka gözlerin bulunması gerekir. Gerçeklenen sistemde optimum parametre seçimi yapılırken en fazla iyileşme bu modülde gerçekleşmiştir. Bu modülün doğru bir şekilde icrası sistem performansını doğrudan etkileyecektir. Çünkü göz bulunmadan göz durumu tespit edilemez yani PERCLOS hesaplanamaz. Göz tespiti için iki adet tablo verilecektir (Tablo 4.2 ve Tablo 4.3). Tablo 4.2’de minNeighbors=3 alınarak değişik searchScaleFactor

değerleri için göz başarımları oranları ile saniyede işlenen kare sayısı değerleri karşılaştırılmalı olarak verilmiştir.

Tablo 4.2. minNeighbors = 3 için searchScaleFactor bağlı göz başarımları oranları ile saniyede işlenen kare sayısı

	searchScale Factor = 1.01	searchScale Factor = 1.02	searchScale Factor = 1.03	searchScale Factor = 1.05	searchScale Factor = 1.1
Enaz Bir Göz Bulunan	%99.98	%99.81	%99.53	%97.15	%92.19
İki Gözde Bulunan	%70.80	%68.11	%64.56	%59.28	%47.39
Tespit Edilmeyen	%0.02	%0.19	%0.48	%2.84	%7.81
İşlenen Kare Sayısı (fps)	8.07	11.22	15.26	16.09	22.37

Tablo 4.3’de minNeighbors=4 alınarak değişik searchScaleFactor değerleri için göz başarımları oranları ile saniyede işlenen kare sayısı karşılaştırılmalı olarak verilmiştir. Tablo 4.2 ve Tablo 4.3. beraber değerlendirildiğinde göz tespiti için ilgili parametreler seçilmiş olur.

Tablo 4.3. minNeighbors = 4 için searchScaleFactor bağlı göz başarımları oranları ile saniyede işlenen kare sayısı

	searchScale Factor = 1.01	searchScale Factor = 1.02	searchScale Factor = 1.03	searchScale Factor = 1.05	searchScale Factor = 1.1
Enaz Bir Göz Bulunan	%99.97	%99.63	%99.08	%95.77	%88.81
İki Gözde Bulunan	%70.12	%66.37	%62.22	%56.36	%43.57
Tespit Edilmeyen	%0.03	%0.37	%0.92	%4.23	%11.19
Saniyede İşlenen Kare Sayısı	7.66	11.65	16.41	19.34	20.03

Tablo 4.2 ve Tablo 4.3.’den de anlaşılacağı gibi searchScaleFactor büyük tutulduğunda saniyede işlenen kare sayısı artmaktadır. Benzer olarak searchScaleFactor küçük tutulduğunda her defasında küçük adımlarla göz arama işlemi yapılacağından işlem karmaşıklığı artmakta yani saniyede işlenen kare sayısı azalmaktadır. searchScaleFactor ve

saniyede işlenen kare sayısındaki gibi bir ters ilişki bulunan göz sayısı ile searchScaleFactor içinde geçerlidir. Optimum parametre seçimi yapılırken göz başarımları ve saniyede işlenen kare sayısı dikkate alınmıştır. DDDS uygulamasında öncelik gözün bulunmasına verildiği için minNeighbors = 3 olarak alınmıştır. searchScaleFactor 1.03'den 1.05'e değişiminde başarımlar; Tablo 4.2'de %2.32 azalırken saniyede işlenen kare sayısı fazla değişmemiştir. Bu yüzden DDDS uygulamasında searchScaleFactor 1.03 olarak alınmıştır.

Gerçekleştirilen DDDS uygulamasında ikinci tip sonuçlar bu kısımda verilecektir. DDDS uygulamasının test edildiği ZJU veritabanında 4 şekilde elde edilmiş veri olduğundan dolayı 4 veri tipi için yüz algılama, yüz tanıma ve göz algılama modüllerinin her veri türü için ayrı ayrı sonuçları verilecektir. Yüz tespit modülünün, 4 video tipi için uygulama sonuçları Tablo 4.4'de verilmiştir.

Tablo 4.4. Yüz tespit modülünün, 4 video türü için uygulama sonuçları

	1. Tip Video	2. Tip Video	3. Tip Video	4. Tip Video
Tespit Edilen Yüz	%96.74	%99.78	%96.78	%78.37
Bulunmayan Yüz	%3.26	%0.22	%3.21	%21.63
Yanlış Etiketlenen Yüz	0%	0%	%0	%0

DDDS modelinde kamera sürücünün karşısında kabul edildiğinden dolayı ZJU veritabanındaki dördüncü tip kayıt türünde bazı deneklerin video çekimleri DDDS uygulaması için uyarı gerektiren bir durumdur. Çünkü bazı videolarda denekler video çekiminde karşıya bakmıyor. Yüz bulma uygulamasının sonuçlarının kısmen de olsa düşük olmasının sebebi dördüncü tip video kayıt klipleridir.

Gerçekleştirilen DDDS uygulamasında yüz tanıma modülünün 4 video tipi için uygulama sonuçları Tablo 4.5'te verilmiştir. İlgili benzer çalışmalarla karşılaştırıldığında yüz tanıma modülü sisteme dâhil edilmesine rağmen hızın düşmemesi yani iyi bir gerçekleşme zamanında sistemin icra edilmesi gerçekleştirilen modelin başarımıdır. Kullanılan veri tabanında çok çeşitli durumlar olmasına rağmen elde edilen yüksek doğrulukta başarımları doğru yöntemlerin kullanıldığının göstergesidir.

Tablo 4.5. Yüz tanıma modülünün 4 video türü için uygulama sonuçları

	1. Tip Video	2. Tip Video	3. Tip Video	4. Tip Video
Tanınan Yüz	%100	%100	%100	%100
Bulunmayan Yüz	%0	%0	%0	%0
Yanlış Etiketlenen Yüz	%0	%0	%0	% 0

DDDS uygulamasında yüz tanıma, sisteme ek bir yük getirmemektedir. Gerçeklenen sistem için iki durum söz konusudur. Birincisi başlangıç ve uyarı sonrası durum, ikincisi ise normal gözlem durumudur. Kişinin ilk görüntüsü alındığında kişi tanındıktan sonra bir ikaz durumu oluşmayıncaya kadar yüz bir daha tanınmaz çünkü bu durumda arabayı süren kişi yerinden kalkmamış demektir. Kişi tanındıktan sonra karar alma sürecinde başlangıç değerleri yerine kişiye özel parametreler sisteme dâhil edilebilir. Örneğin kişi göz kırpma sayısı herkeste benzer olmasına rağmen kişi tanındıktan sonra bu parametre ilgili kişi için daha kesin olarak belirlenebilir.

Gerçekleştirilen DDDS uygulamasında göz tespiti modülü için elde edilen sonuçların 4 video tipi için uygulama başarımı Tablo 4.6’da verilmiştir.

Tablo 4.6. Göz tespiti modülünün 4 video türü için uygulama sonuçları

	1. Tip Video	2. Tip Video	3. Tip Video	4. Tip Video
Enaz Bir Göz Bulunan	%99.70	%100	%98.94	%99.43
İki Gözde Bulunan	%53.53	%77.29	%59.69	%68.10
Tespit Edilmeyen	%0.30	%0	%1.05	%0.57

Tablo 4.6’da görüldüğü gibi tespit edilen yüzde göz tespit işlemi çok yüksek oranlarda gerçekleşmektedir. Her üç video türünde de elde edilen sonuçlar tatmin edicidir. Yani tasarlanan sistemin handikapı göz tespit modülü değildir. Bu modülün başarımı en yüksek başarıma sahip modüldür.

Tasarlanan sistemin son modülünde göz kırpma oranları Tablo 4.7’de verilmiştir. Oranların kısmende olsa düşük olmasının sebebi tespit edilmeyen gözlerin olmasıdır. Ayrıca daha önce değinildiği gibi 4. Tip videolar aşağıdan yukarıya görüntü alındığı için ilgili videolarda göz kırpma tespiti başarımı doğal olarak düşüktür. Toplam 255 göz kırpmasından yüz ve göz tespit edilenlerin videolardaki toplam 227 adet göz kırpması tespit edilmiştir.

Tablo 4.7. ZJU veri tabanı için tespit edilen göz kırpma sayıları

Video No	Göz Kırpma Sayısı	Tespit Edilen Göz Kırpma Sayısı	Video No	Göz Kırpma Sayısı	Tespit Edilen Göz Kırpma Sayısı	Video No	Göz Kırpma Sayısı	Tespit Edilen Göz Kırpma Sayısı
1	3	1	31	6	6	61	2	2
2	3	4	32	5	5	62	3	2
3	4	0	33	4	4	63	2	2
4	3	3	34	2	2	64	2	2
5	4	3	35	5	5	65	4	4
6	5	5	36	4	4	66	3	3
7	4	1	37	3	3	67	3	3
8	4	2	38	3	3	68	3	3
9	4	3	39	2	2	69	4	4
10	3	3	40	4	4	70	2	2
11	3	3	41	2	2	71	3	2
12	5	0	42	2	2	72	5	5
13	3	3	43	2	2	73	3	3
14	2	2	44	1	1	74	3	3
15	1	0	45	4	4	75	4	3
16	2	2	46	3	3	76	2	2
17	3	3	47	3	3	77	3	3
18	2	2	48	4	4	78	1	1
19	3	3	49	2	2	79	3	3
20	3	3	50	3	3	80	4	4
21	6	5	51	2	2			
22	5	5	52	2	2			
23	6	6	53	4	4			
24	6	6	54	2	2			
25	5	5	55	3	2			
26	5	6	56	3	2			
27	5	5	57	2	2			
28	5	0	58	2	2			
29	6	3	59	2	1			
30	7	7	60	1	1			

Mevcut uygulama saniyede 16 adet kare işleyen uygulama daha önceki çalışmalarda bildirilen saniyede 6 kare işleme zorunluluğu sağlamıştır. Mevcut uygulamada yüz ve göz tespiti için en iyi modeller kullanıldığından dolayı paralelinde en çok işlem gerektiren modeller olmasından dolayı performansı olumsuz yönde etkilemişlerdir. Deneklerin iki çeşit gözlük takması ve gözlüklerden yansıyan ışık sistem performansını olumsuz yönde etkilemiştir. Fakat elde edilen sonuçların doğruluğu hakkında olumlu bilgi vermesi açısından uygulama sonuçları gözlükler elimine edilerek verilmemiştir. Bu sistemde görüntü elde edininim şekli çok çeşitlidir. Tüm modüllerin sonuçlarında ZJU veri tabanındaki hiçbir örnek çıkarılmamıştır. ZJU veri tabanında ışık kontrolünün olmaması, arka planın dinamik olması ve bağımsız deneklerle farklı şartlarda kayıt yapılması veritabanını gerçek verilerle kısmen de olsa örtüşür hale getirmiştir. Dolayısıyla tasarlanan DDDS uygulamasının sonuçlarının gerçek zamanlı koşacak bir sistemin doğruluğuna yakın olduğu söylenebilir.

Bu tez çalışmasında literatürde sadece gözlüklü veya sadece elle konumlandırılmış gözden uykululuk tespit çalışmalarına göre daha genel ve gerçek zamanlı çalışmaya daha uygun bir model tasarlanarak uygulaması yapılmıştır. Kişi yetkilendirmek için retina tanıyan uygulamaya göre daha ucuz donanımlı bir model sunulmuştur. FPGA ile gerçekleştirilen DDDS sistemine göre daha ucuz donanımlı ve başarımı daha yüksek bir model sunulmuştur. Kaskad sınıflandırıcı ile yüz ve göz tespiti yapan uygulamalarda yanlışlıkla etiketlenen giriş imgeleri için yanlış etiketlemeyi tamamen ortadan kaldıran bir yöntem sunulmuştur. Birçok giriş imgesinde ağzın yüz olarak etiketlendiği görülmüştür. Bu yanlış etiketleme maksimum göz büyüklüğü yüz büyüklüğü ile ilişkilendirilerek ortadan kaldırılmıştır. LBPH yüz tanıma modeli araştırmalarımız kadarıyla ilk kez bu tez kapsamında göz durumu tespit problemine uyarlanmıştır.

Mevcut tasarlanan DDDS uygulamasının tüm kodları OpenCV tabanlı ve C ile C++ dilleri standardında olduğu için platform bağımsız olarak koşabilecektir. Geriye dönük optimizasyon çalışması yapıp, sistemin parametrelerini daha iyi olacak şekilde sistem tekrardan gerçekleştirilmeye müsaittir.

Gelecek çalışmalarda mevcut kaskad sınıflandırıcılar için kamera ile yüz arasındaki en az ve en çok mesafe kestirilerek sistem gerçekleştirirse yani model eğitimlerinde büyüklük kestirimi yapılırsa mevcut uygulamanın gerçekleştirme zamanını olumlu yönde etkileyecektir. Çünkü Viola-Jones yönteminde eğitim aşaması çok uzun sürdüğü için

genelde hazır modeller yaygın olarak kullanılmaktadır. Fakat hazır modeller problemi en geniş şekliyle tanımladığı için icra süreleri çok uzun sürmektedir.

Sonuç olarak gerçekleştirilen sistem gerçek zamanlı icra edilecek modeller için güzel bir örnektir. Tasarlanan model yüz tanımayı gerçekleştirmenin sisteme yük getirmeyeceği varsayımıyla, incelemelerimiz kadarıyla ilk ve tek çalışmadır. Problem uzayını sıralı olarak küçültme, benzer çalışmalarda sunulsa da bu çalışmada bir adım ileriye gidilerek göz durumu tespit işlemi için problemi bütünden ayırt edici özellikler bulmak yerine tek kişi için mevcut durumu değerlendirmeye müsait olması nedeniyle aday bölge uzayının düşürülebilmesini destekleyen bir yapı sunulmuştur.

Bu tez çalışması, gerçekleştirilen DDDS uygulamasındaki dört modül hakkında yapılan literatür çalışması ve sunulan modelle bundan sonra bu yönde yapılacak çalışmalara öncü olabilecek potansiyele sahip bir çalışma olmaya adaydır.

KAYNAKLAR

- [1] **Danisman, T., Bilasco, I.M., Djeraba, C., and Ihaddadene, N.,** 2010. "Drowsy driver detection system using eye blink patterns", *Machine and Web Intelligence (ICMWI), 2010 International Conference on* , pp. 230-233, 3-5 Oct. 2010.
- [2] NCSDR/NHTSA Expert Panel on Driver Fatigue and Sleepiness, "Drowsy driving and automobile crashes," National Highway Traffic Safety Administration (NHTSA) report.
- [3] **Rosekind, M. R.,** 2005. "Underestimating the societal costs of impaired alertness: safety, health and productivity risks," *Sleep Medicine*, vol. 6, pp. S21-S25, 2005.
- [4] **Hancı, İ.H., ve Eşiyok, B.** "Uyku ve Trafik Kazaları", <http://www.ttb.org.tr/STED/sted0301/1.html>, 6 Mayıs 2012.
- [5] **Liu, C. and Subramanian, R.,** 2009. "Factors Related to Fatal Single-Vehicle Run-Off-Road Crashes," U.S. Department of Transportation, American National Highway Traffic Safety Administration, DOT HS 811 232, Washington, D.C., November, 2009.
- [6] **Knipling, R.R. and Wang, J.S.,** 1994. "Crashes and Fatalities Related to Driver Drowsiness, Fatigue," National Highway Traffic Safety Administration, Washington, D.C., Information Management Consultants Inc., McLean, VA., 1994.
- [7] http://www.traport.org.tr/tk_genelbilgiler.php?yID=55, 6 Mayıs 2012.
- [8] **TÜİK,** 2002-2011 yılı Trafik Kaza İstatistikleri Kara Yolu, Türkiye İstatistik Kurumu Matbaası, Ankara, 2012.
- [9] **Yenikaya, G.,** 2006. "Sürücü Davranışlarının Modellenmesine Bilişsel Mimari Yaklaşımı", *Doktora Tezi*, Uludağ Üniversitesi, Fen Bilimleri Enstitüsü.
- [10] **Takada, Y. and Shimoyama, O.,** 2001. Evaluation of driving-assistance systems based on drivers' workload. International Driving Symposium on Human Factors in Driving Assessment, Training and vehicle design, Aspen, Colorado, August 14-17.
- [11] **Vural, E., Cetin, M., Ercil, A., Littlewort, G., Bartlett, M. and Movellan, J.,** 2007. "Drowsy Driver Detection Through Facial Movement Analysis", IEEE International Workshop, HCI 2007 Rio de Janeiro, Brazil 2007, pp. 6-18 October 20, 2007.
- [12] **Vural, E., Cetin, M., Ercil, A., Littlewort, G., Bartlett, M. ve Movellan, J.,** 2008. "Bilgisayarlı Görü Yöntemleriyle Sürücüde Uykululuğun Sezimi", *Signal*

Processing, Communication and Applications Conference, 2008. SIU 2008. IEEE 16th , pp.1-4, 20-22 April 2008.

- [13] **Vural, E., Cetin, M., Ercil, A., Littlewort, G., Bartlett, M. and Movellan, J.,** 2010. "Discrimination of Moderate and Acute Drowsiness Based on Spontaneous Facial Expressions", *Pattern Recognition (ICPR), 2010 20th International Conference on* , pp. 3874-3877, 23-26 Aug. 2010.
- [14] **Vitabile, S., De Paola, A. and Sorbello, F.,** 2011. "A real-time non-intrusive FPGA-based drowsiness detection system", *Journal of Ambient Intelligence and Humanized Computing*, vol. 2, no. 4, pp 251-262, Dec. 2011.
- [15] **Cobb, W.,** 1983 Recommendations for the practice of clinical neurophysiology. Elsevier (1983).
- [16] **Nikhil, R.P., Chien-Yao, C. Ko, L.W., Chao, C.F., Jung, T.P., Liang S.F. and Lin, C.T.,** 2008. "EEG-based Subject-and Session-independent Drowsiness Detection: *An Unsupervised Approach*," *EURASIP Journal on Advances in Signal Processing*, ISSN: 1110-8657, January 2008.
- [17] **Chung, H. K.,** 2005. "Electroencephalographic study of drowsiness in simulated driving with sleep deprivation". *International Journal of Industrial Ergonomics*. vol. 35, isseu 4, April 2005, pp. 307-320.
- [18] **Takei, Y. and Furukawa, Y.,** 2005. "Estimate of driver's fatiguetthrough steering motion". *Man and Cybernetics 2005 IEEE International Conference*, vol. 2, pp.1765-1770.
- [19] **Igarashi, K., Takeda, K., Itakura, F. amd Abut, H.,** 2005. "DSP for In-Vehicle and Mobile Systems". Springer (2005).
- [20] **Hayırlı, S.,** 2005. "Uykulu Sürücü Belirleme Sistemi", *Yüksek Mühendislik Tezi*, Hacettepe Üniversitesi, Fen Bilimleri Enstitüsü.
- [21] **Seng, K.P., Chew, W.J. and Ang L.M.,** 2012. "Automatic Vision-Based Drowsy Driver Detection System ", http://www.ecti-thailand.org/assets/papers/365_pub_24.pdf, 15 Mart 2012.
- [22] **Garg, R., Gupta, V. and Agrawal, V.,** 2009. "A Drowsy Driver Detection and security system", *Ultra Modern Telecommunications & Workshops, 2009. ICUMT '09. International Conference on*, pp.1-8, 12-14 Oct. 2009
- [23] **Deshmukh, S.V., Radake, D.P. and Hande, K.N.,** 2011. "Driver fatigue Detection Using Sensor Network", *International Journal of Engineering Science and Technology (IJEST)*. pp 89-92, Feb. 2011.
- [24] **Lee, D., Oh, S., Heo, S. and Hahn, M.,** 2008. " Drowsy Driving Detection Based on the Driver's Head Movement using Infrared Sensors", *Universal Communication, 2008. ISUC '08. Second International Symposium on* , pp.231-236, 15-16 Dec. 2008.

- [25] **Dinges, D.**, 1998. "PERCLOS: "A valid psychophysiological measure of alertness as assessed by psychomotor vigilance indianapolis", Federal Highway Administration, Office of Motor Carriers, Technical report, MCRT-98-006, 1998.
- [26] **Vats, M., and Garg, A.**, 2012. "Security System for Drowsy Driver by using Artificial Neural Network and Varying Layer Method", *International Journal of Electronics & Communication Technology (IJECT)* vol. 3, issue 1, pp. 257-261, Jan. - March 2012.
- [27] **Race, R., Byrne, V.E., Bierman, D.M., Legrand, J.-M., Gricourt, D., Davis, B.K., Staszewski, J.J. and Carnahan, B.**, 1998. "A drowsy driver detection system for heavy vehicles," *Digital Avionics Systems Conference, 1998. Proceedings., 17th DASC. The AIAA/IEEE/SAE* , vol. 2, pp. I36/1 - I36/8 vol. 2, 31 Oct-7 Nov. 1998.
- [28] **Vaisnavi, N.M. and Santhosh, R.R.**, 2010. " Performance evaluation of statistical approach for drowsiness detection for driver's with and without spectacles", *Computing Communication and Networking Technologies (ICCCNT), 2010 International Conference on* , pp.1-5, 29-31 July 2010.
- [29] **Xu, C., Zheng, Y. and Wang, Z. F.**, 2008. "Efficient eye states detection in realtime for drowsy driving monitoring system", *IEEE Conf. on ICIA*, vol. 1-4, pp. 170-174.
- [30] BiometriCS, 2006. "Gömülü Sistemler Eğitim Notları", Ankara, 2006.
- [31] **Szeliski , R.**, 2010. "Computer Vision: Algorithms and Applications", Springer.
- [32] **Bradski, G. and Kaehler, A.**, 2008. "Learning OpenCV", O'Reilly, Tokyo
- [33] **Yang, M. H., Kriegman, D.J., Ahuja, N.**, 2002. "Detecting faces in images: a survey", *Pattern Analysis and Machine Intelligence, IEEE Transactions on* , vol.24, no.1, pp.34-58, Jan 2002.
- [34] **P. Viola and M. Jones**, 2001. "Rapid object detection using a boosted cascade of simple features", *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on* , vol.1, pp. I-511- I-518 vol.1, 2001.
- [35] **Ojala, T., Pietikainen, M. and Harwood, D.**, 1994. "Performance evaluation of texture measures with classification based on Kullback discrimination of distributions," *Pattern Recognition, 1994. Vol. 1 - Conference A: Computer Vision & Image Processing., Proceedings of the 12th IAPR International Conference on* , vol.1, no., pp.582-585 vol.1, 9-13 Oct 1994.

- [36] **Ojala, T., Pietikainen, M. and Maenpaa, T.,** 2002. "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002.
- [37] **Zhang, C. and Zhang, Z.,** "A Survey of Recent Advances in Face Detection" June 2010 Technical Report MSR-TR-2010-66, Microsoft Research, Microsoft Corporation. http://research.microsoft.com/en-us/um/people/chazhang/publications/tr2010-66_chazhang.pdf, 20.02.2013.
- [38] **Yang, G. and Huang, T. S.,** 1994. "Human Face Detection in Complex Background," *Pattern Recognition*, vol. 27, no. 1, pp. 53-63, 1994.
- [39] **Kanade, T.,** 1973. "Picture Processing by Computer Complex and Recognition of Human Faces," PhD thesis, Kyoto Univ., 1973.
- [40] **Kotropoulos, C. and Pitas, I.,** 1997. "Rule-Based Face Detection in Frontal Views," *Proc. Int'l Conf. Acoustics, Speech and Signal Processing*, vol. 4, pp. 2537-2540, 1997.
- [41] **Chetverikov, D. and Lerch, A.,** 1993 "Multiresolution Face Detection", *Theoretical Foundations of Computer Vision*, vol. 69, pp. 131-140, 1993.
- [42] **Augusteijn, M.F. and Skujca, T.L.,** 1993. "Identification of Human Faces through Texture-Based Feature Recognition and Neural Network Technology," *Proc. IEEE Conf. Neural Networks*, pp. 392-398, 1993.
- [43] **Chai, D. and Ngan, K.N.,** 1998. "Locating Facial Region of a Head-and- Shoulders Color Image," *Proc. Third Int'l Conf. Automatic Face and Gesture Recognition*, pp. 124-129, 1998.
- [44] **Sakai, T., Nagao, M. and Fujibayashi, S.,** 1969. "Line Extraction and Pattern Detection in a Photograph", *Pattern Recognition*, vol. 1, pp. 233-248.
- [45] **Govindaraju, V., Srihari, S.N. and Sher, D.B.,** 1990, "A Computational Model for Face Location," *Proc. Third IEEE Int'l Conf. Computer Vision*, pp. 718-721, 1990.
- [46] **Cootes T.F. and Taylor, C.J.,** 1996 "Locating Faces Using Statistical Feature Detectors," *Proc. Second Int'l Conf. Automatic Face and Gesture Recognition*, pp. 204-209, 1996.
- [47] **Lanitis, A., Taylor, C.J. and Cootes, T.F.,** 1995. "An Automatic Face Identification System Using Flexible Appearance Models", *Image and Vision Computing*, vol. 13, no. 5, pp. 393-401.
- [48] **Yuille, A., Hallinan, P. and Cohen, D.,** 1992. "Feature Extraction from Faces Using Deformable Templates", *Int'l J. Computer Vision*, vol. 8, no. 2, pp. 99-111.

- [49] **Kwon Y.H. and Lobo, N. V.**, 1994. "Face Detection Using Templates," *Proc. Int'l Conf. Pattern Recognition*, pp. 764-767, 1994.
- [50] **Sung, K. ve Poggio, T.**, 1998. "Example-based Learning for View-based Human Face Detection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (20):39-51.
- [51] **Kohonen, T.**, 1989. "Self-Organization and Associative Memory", Springer.
- [52] **Turk, M. and Pentland, A.**, 1991. "Eigenfaces for Recognition", *J. Cognitive Neuroscience*, vol. 3, no. 1, pp. 71-86.
- [53] <http://www.face-rec.org/algorithms/>, 6 Şubat 2013.
- [54] **Rowley, H., Baluja, S. and Kanade, T.**, 1996. "Neural Network-Based Face Detection", *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 203-208.
- [55] **Rowley, H., Baluja, S., and Kanade, T.**, 1998. "Neural Network-Based Face Detection", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, no. 1, pp. 23-38.
- [56] **Schneiderman H. and Kanade, T.**, 2000. "A Statistical Method for 3D Object Detection Applied to Faces and Cars", *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 1, pp. 746-751.
- [57] **Lienhart, R. and Maydt, J.**, "An extended set of Haar-like features for rapid object detection," *Image Processing. 2002. Proceedings. 2002 International Conference on* , vol.1, pp. I-900- I-903 vol.1, 2002.
- [58] **Papageorgiou, C., Oren, M. and Poggio., T.**, 1998. "A general framework for object detection", *In International Conference on Computer Vision*, 1998.
- [59] **Lienhart, R., Kuranov, A. and Pisarevsky. V.**, 2003. "Empirical analysis of detection cascades of boosted classifiers for rapid object detection." *In Proceedings of the 25th DAGM-Symposium*, pages 297–304, Magdeburg, Germany, 2003.
- [60] **Ojala, T., Pietikainen, M. and Harwood, D.**, 1996. "A comparative study of texture measures with classification based on featured distribution," *Pattern Recog.*, vol. 29, no. 1, pp. 51–59, 1996.
- [61] **Ahonen, T., Hadid, A. and Pietikainen, M.**, 2004. "Face recognition with local binary patterns," in *Proc. Euro. Conf. Comput. Vis.*, 2004, pp. 469–481.
- [62] **Hadid, A., Pietikainen, M., and Ahonen, T.**, 2004. "A discriminative featurespace for detecting and recognizing faces," in *Proc. Int. Conf. Comput. Vis. Pattern Recog.*, 2004, pp. 797–804.

- [63] **Huijsmans, D. P. and Sebe, N.**, 2003. "Content-based indexing performance: A class size normalized precision, recall, generality evaluation," in Proc. Int. Conf. Image Process., 2003, pp. 733–736.
- [64] **Grangier, D. and Bengio, S.**, 2008 "A discriminative kernel-based approach to rank images from text queries," IEEE Trans. Pattern Anal. Mach. Intell., vol. 30, no. 8, pp. 1371–1384, Aug. 2008.
- [65] **Ali, W., Georgsson, F. and Hellström, T.**, 2008. "Visual tree detection for autonomous navigation in forest environment," in Proc. IEEE Intell. Veh. Symp., Jun. 2008, pp. 560–565.
- [66] **Nanni, L. and Lumini, A.**, 2008. "Ensemble of multiple pedestrian representations," IEEE Trans. Intell. Transp. Syst., vol. 9, no. 2, pp. 365–369, Jun. 2008.
- [67] **Maenpaa, T., Viertola, J. and Pietikainen, M.**, 2003. "Optimising colour and texture features for real-time visual inspection," Pattern Anal. Appl., vol. 6, no. 3, pp. 169–175, 2003.
- [68] **Turtinen, M., Pietikainen, M., and Silven, O.**, 2006. "Visual characterization of paper using Isomap and local binary patterns," IEICE Trans. Inform. Syst., vol. E89D, no. 7, pp. 2076–2083, 2006.
- [69] **Heikkila, M. and Pietikainen, M.**, 2006. "A texture-based method for modeling the background and detecting moving objects," IEEE Trans. Pattern Anal. Mach. Intell., vol. 28, no. 4, pp. 657–662, Apr. 2006.
- [70] **Kellokumpu, V., Zhao, G. and Pietikainen, M.**, "Human activity recognition using a dynamic texture based method," presented at the Brit. Mach. Vis. Conf., Leeds, U.K., 2008.
- [71] **Oliver, A., Lladó, X., Freixenet, J. and Martí, J.**, 2007. "False positive reduction in mammographic mass detection using local binary patterns," in Proc. Med. Image Comput. Comput.-Assisted Intervention Conf., 2007, pp. 286–293.
- [72] **Kluckner, S., Pacher, G., Grabner, H., Bischof, H. and Bauer, J.**, 2007. "A 3D teacher for car detection in aerial images," in Proc. IEEE Int. Conf. Comput. Vis., 2007, pp. 1–8.
- [73] **Lucieer, A., Stein, A. and Fisher, P.**, 2005. "Multivariate texture-based segmentation of remotely sensed imagery for extraction of objects and their uncertainty," Int. J. Remote Sens., vol. 26, no. 14, pp. 2917–2936, 2005.
- [74] **Jin, H., Liu, Q., Lu, H. and Tong, X.**, 2004. "Face detection using improved LBP under Bayesian framework," in Proc. Int. Conf. Image Graph., 2004, pp. 306–309.

- [75] **Huang, D., and Wang, Y.**, 2007. "A robust method for near infrared face recognition based on extended local binary pattern," in Proc. Int.Symp. Vis. Comput., 2007, pp. 437–446.
- [76] **Huang, Y., Wang, Y. and Tan, T.**, 2006. "Combining statistics of geometrical and correlative features for 3D face recognition," in Proc. Brit. Mach. Vis. Conf., 2006, pp. III: 879–888.
- [77] **Zhang, L., Chu, R., Xiang, S. and Li, S. Z.**, 2007. "Face detection based on Multi-Block LBP representation," in Proc. Int. Conf. Biometrics, 2007, pp. 11–18.
- [78] **Huang, D., Shan, C., Ardabilian, M., Wang, Y. and Chen, L.**, 2011 "Local Binary Patterns and Its Application to Facial Image Analysis: A Survey," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on* , vol.41, no.6, pp.765-781, Nov. 2011.
- [79] **Shan, C., Gong, S. and McOwan, P. W.**, 2009. "Facial expression recognition based on local binary patterns: A comprehensive study," *Image Vis. Comput.*, vol. 27, no. 6, pp. 803–816, May 2009.
- [80] **Jafri, R. and Arabnia, H. R.**, 2009. "A Survey of Face Recognition Techniques", *Journal of Information Processing Systems*, Vol.5, No.2, June 2009.
- [81] **Xianwei, L. and Haiyang, Z.**, 2013. "A Survey of Face Recognition Methods", *Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering (ICCSEE 2013)*, Published by Atlantis Press, Paris, France.
- [82] **Zhao, W., Chellaappa, R. and Rosenfeld, A.**, 2003. "Face Recognition: A Literature Survey", *ACM Computing Surveys*, Vol. 35, No. 4, December 2003, pp. 399–458.
- [83] **Lawrence, S., Giles, C. L., Tsoi, A. C. and Back, A. D.**, 1997. "Face Recognition: A Convolutional Neural Network Approach," *IEEE Transactions on Neural Networks*, Special Issue on Neural Networks and Pattern Recognition, pp.1-24, 1997.
- [84] **Brunelli R. and Poggio, T.**, 1993. "Face recognition: features versus templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.15, pp.1042- 1052, 1993.
- [85] **Campadelli, P., and Lanzarotti, R.**, 2005. "A Face Recognition System Based on Local Feature Characterization," in *Advanced Studies in Biometrics*, Vol.3161, *Lecture Notes in Computer Science*, M. Tistarelli, J. Bigun, and E. Grosso, Eds. Berlin: Springer, 2005, pp.147- 152.
- [86] **Wiskott, L., Fellous, J. M., Krüger, N. and von der Malsburg, C.**, 1999. "Face Recognition by Elastic Bunch Graph Matching," in *Intelligent Biometric*

Techniques in Fingerprint and Face Recognition, L. C. Jain, U. Halici, I. Hayashi, S. B. Lee, and Jae-Ho, Eds.: CRC Press, 1999, pp.355-396.

- [87] **Liposcak Z. and Loncaric, S.**, 1999. "A scale-space approach to face recognition from profiles," in *Proceedings of the 8th International Conference on Computer Analysis of Images and Patterns, Vol. 1689, Lecture Notes In Computer Science*. London, UK: Springer- Verlag, 1999, pp.243-250.
- [88] **Lu, X., Wang, Y. and Jain, A. K.**, 2003. "Combining Classifiers for Face Recognition," in *Proc. IEEE International Conference on Multimedia & Expo (ICME 2003)*. Baltimore, MD, 2003, pp.13-16.
- [89] The OpenCV Reference Manual, Release 2.4.5, 2013. <http://docs.OpenCV.org/OpenCV2refman.pdf>, 12 Haziran 2013.
- [90] **Fisher, R. A., 1936.** The use of multiple measurements in taxonomic problems. *Annals Eugen.* 7 (1936), 179–188.
- [91] **Belhumeur, P. N., Hespanha, J., and Kriegman, D.**, 1997. "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19, 7 (1997), 711–720.
- [92] **Baggio, D.L., Emami, S., Escrivá D.M., Ievgen, K., Mahmood, N., Saragih, J. and Shilkrot. R.**, 2012. "Mastering OpenCV with Practical Computer Vision Projects", Packt Publishing, Birmingham, pp. 260-310.
- [93] **Raudys S. and Jain. A.K.**, 1991, "Small sample size effects in statistical pattern recognition: Recommendations for practitioners." - *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13, 3 (1991), 252-264.
- [94] **Pan, G., Sun, L., Wu Z. and Lao, S.**, 2007. "Eyeblink-based Anti-spoofing in Face Recognition from a Generic Webcamera," The 11th IEEE International Conference on Computer Vision (ICCV'07), Rio de Janeiro, Brazil, October, 2007.
- [95] **Charn, L.H., Rasid, L.N. and Suandi. S.A.**, 2010. "A Study on the Effectiveness of Different Patch Size and Shape for Eyes and Mouth Detection", *International Journal on Computer Science and Engineering (IJCSE)* vol. 02, no. 03, 2010.
- [96] **Hansen, D.W., Qiang Ji**, 2010. "In the Eye of the Beholder: A Survey of Models for Eyes and Gaze," *Pattern Analysis and Machine Intelligence, IEEE Transactions on* , vol.32, no.3, pp.478,500, March 2010.
- [97] **Kim K. N. and Ramakrishna, R. S.**, 1999. "Vision-Based Eye-Gaze Tracking for Human Computer Interface," *Proc. IEEE Int'l Conf. Systems, Man, and Cybernetics*, 1999.

- [98] **Young, D. Tunley, H. and Samuels, R.**, 1995 “Specialised Hough Transform and Active Contour Methods for Real-Time Eye Tracking,” Technical Report 386, School of Cognitive and Computing Sciences, Univ. of Sussex, 1995.
- [99] **Daugman, J.**, 2003. “The Importance of Being Random: Statistical Principles of Iris Recognition,” *Pattern Recognition*, vol. 36, no. 2, pp. 279-291, 2003.
- [100] **Li, D. Winfield, D. and Parkhurst, D. J.**, 2005 “Starburst: A Hybrid Algorithm for Video-Based Eye Tracking Combining Feature- Based and Model-Based Approaches,” *Proc. Vision for Human- Computer Interaction Workshop, IEEE Computer Vision and Pattern Recognition Conf.*, 2005.
- [101] **Ivins J. P. and Porrill, J.**, 1998. “A Deformable Model of the Human Iris for Measuring Small 3-Dimensional Eye Movements,” *Machine Vision and Applications*, vol. 11, no. 1, pp. 42-51, 1998.
- [102] **Waite J. and Vincent, J. M.**, 1992. “A Probabilistic Framework for Neural Network Facial Feature Location,” *British Telecom Technology J.*, vol. 10, no. 3, pp. 20-29, 1992.
- [103] **Herpers, R, Michaelis, M., Lichtenauer, K. and Sommer, G.**, 1996. “Edge and Keypoint Detection in Facial Regions,” *Proc. Int’l Conf. Automatic Face and Gesture-Recognition*, pp. 212-217, 1996.
- [104] **Kawato S. and Ohya, J.**, 2000. “Real-Time Detection of Nodding and Head-Shaking by Directly Detecting and Tracking the Between- Eyes,” *Proc. IEEE Fourth Int’l Conf. Automatic Face and Gesture Recognition*, pp. 40-45, 2000.
- [105] **Sirohey, S. Rosenfeld, A. and Duric, Z.**, 2002. “A Method of Detecting and Tracking Irises and Eyelids in Video,” *Pattern Recognition*, vol. 35, no. 6, pp. 1389-1401, June 2002.
- [106] **Sirohey S.A. and Rosenfeld, A.**, 2001. “Eye Detection in a Face Image Using Linear and Nonlinear Filters,” *Pattern Recognition*, vol. 34, pp. 1367-1391, 2001.
- [107] **Yang, J., Stiefelhagen, R, Meier, U. and Waibel, A.**, 1998 “Real-Time Face and Facial Feature Tracking and Applications,” *Proc. Conf. Auditory-Visual Speech Processing*, pp. 79-84, 1998.
- [108] **Fasel, I., R., Fortenberry, B. and Movellan, J. R.**, 2005. “A Generative Framework for Real Time Object Detection and Classification,” *Computer Vision and Image Understanding*, vol. 98, no. 1, pp. 182- 210, Apr. 2005.
- [109] **Xie, X., Sudhakar, R. and Zhuang, H.**, 1994. “On Improving Eye Feature-Extraction Using Deformable Templates,” *Pattern Recognition*, vol. 27, no. 6, pp. 791-799, June 1994.

- [110] **Comaniciu, D., Ramesh, V. and Meer, P.,** 2003. "Kernel-Based Object Tracking," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 564-577, May 2003.
- [111] **Reisfeld D. and Yeshurun, Y.,** 1992. "Robust Detection of Facial Features by Generalized Symmetry," *Proc. Int'l Conf. Pattern Recognition*, vol. I, pp. 117-120, 1992.
- [112] **Loy G. and Zelinsky, A.,** 2003. "Fast Radial Symmetry for Detecting Points of Interest" *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 25, no. 8, pp. 959-973, Aug. 2003.
- [113] **Gofman Y. and Kiryati, N.,** 1996. "Detecting Symmetry in Gray Level Images: The Global Optimization Approach," *Proc. Int'l Conf. Pattern Recognition*, 1996.

ÖZGEÇMİŞ

Sedat GOLGİYAZ
Fırat Üniversitesi
Mühendislik Fakültesi
Bilgisayar Mühendisliği Bölümü
23119, Elazığ
sedatg@bingol.edu.tr

- 1981** Bingöl’de doğdu.
- 2000-2005** Karadeniz Teknik Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümünden mezun oldu.
- 2011-** Bingöl Üniversitesi Bilgisayar Mühendisliği Bölümünde Araştırma Görevlisi olarak çalışmaktadır.