

**İNSAN YÜZÜ İMGELERİNDE DAİRESEL HOUGH DÖNÜŞÜMÜ  
KULLANILARAK GÖZ DURUMU TESPİTİ**

**Ömer Faruk SÖYLEMEZ**

**Yüksek Lisans Tezi**

**Bilgisayar Mühendisliği Anabilim Dalı**

**Danışman: Yrd. Doç. Dr. Burhan ERGEN**

**EKİM-2012**

T.C.  
FIRAT ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ

İNSAN YÜZÜ İMGELERİNDE DAİRESEL HOUGH DÖNÜŞÜMÜ  
KULLANILARAK GÖZ DURUMU TESPİTİ

YÜKSEK LİSANS TEZİ

Ömer Faruk SÖYLEMEZ

(101129113)

Tezin Enstitüye Verildiği Tarih : 3 Ekim 2012

Tezin Savunulduğu Tarih : 20 Eylül 2012

Tez Danışmanı: Yrd. Doç. Dr. Burhan ERGEN

Diğer Jüri Üyeleri : Doç. Dr. Arif GÜLTEN

Yrd. Doç. Dr. Ahmet ÇINAR

EKİM - 2012

## ÖNSÖZ

Çalışmalarım boyunca değerli yardım ve katkılarıyla beni yönlendiren saygıdeğer hocam Yrd. Doç. Dr. Burhan ERGEN'e, tez çalışmam boyunca benden hiçbir zaman desteğini esirgemeyen, sabır ve anlayışıyla beni destekleyen, en çekilmez olduğum anlarda bile yanımda olan biricik eşim Nesrin HARK SÖYLEMEZ'e sonsuz teşekkürlerimi sunarım.

## İÇİNDEKİLER

ÖNSÖZ .....	II
İÇİNDEKİLER.....	III
ÖZET .....	V
ABSTRACT .....	VI
ŞEKİLLER LİSTESİ .....	VII
TABLolar LİSTESİ.....	VIII
DENKLEMLER LİSTESİ .....	IX
1. GİRİŞ .....	1
2. YÜZ TESPİTİ.....	2
2.1. Yüz Tespiti Problemleri.....	2
2.2. Yüz Tespiti Çalışmaları .....	3
2.2.1. Bilgi Tabanlı Yöntemler .....	4
2.2.2. Değişmeyen Özellik Yaklaşımları.....	5
2.2.3. Şablon Eşleştirme Yöntemleri.....	6
2.2.4. Görünüm Tabanlı Yöntemler .....	7
3. GÖZ TESPİTİ.....	9
3.1. Şablon Tabanlı Yaklaşımlar.....	9
3.2. Özellik Tabanlı Yaklaşımlar .....	10
3.3. Görünüm Tabanlı Yaklaşımlar.....	10
4. GELİŞTİRİLEN YÖNTEM .....	11
4.1. Yüz Konumu Tespiti .....	11
4.1.1. Viola-Jones Nesne Tespit Sistemi ile Yüz Tanıma .....	12
4.1.1.1. İntegral İmge .....	12
4.1.1.2. AdaBoost Öğrenme.....	14
4.1.1.3. Elemeli Kaskad Yapı .....	17
4.2. Göz Konumu Tespiti .....	18
4.3. Göz Durumu Tespiti .....	19
4.3.1. Kontrast Yükseltme .....	20
4.3.2. Eşikleme.....	22
4.3.3. Doldurma ve Küçük Şekilleri Eleme .....	23
4.3.4. Kenar Çıkarma .....	23
4.3.5. Dairesel Şekil Tespiti.....	24

5. UYGULAMA SONUÇLARI .....	27
5.1. Yüz Konumu Tespiti .....	27
5.2. Göz Konumu Tespiti .....	28
5.3. Göz Durumu Tespiti .....	30
6. SONUÇLAR.....	37
REFERANSLAR.....	38
ÖZGEÇMİŞ.....	42

## ÖZET

Göz durumu, günümüzde bir çok uygulamaya giriş verisi olarak kullanılmaktadır. Bu uygulamalara örnek olarak yüz ifadelerinin tanınması, insan-bilgisayar etkileşimi ve sürücü yorgunluk tespit sistemi verilebilir. Bu tarz uygulamaların akademik ve ticari alanlarda kullanılmasıyla göz durumu tespiti son birkaç yılda büyük bir ilgiyle karşılaşmıştır.

Bu tez çalışmasında, irisin göz içerisinde oluşturduğu dairesel yapının, dairesel Hough dönüşümü vasıtasıyla tespit edilmesine dayalı bir göz durumu tespit sistemi gerçekleştirilmiştir. Başlangıçta verilen karmaşık bir görüntüden Viola- Jones nesne tespit sistemi yardımıyla yüz imgeleri ve göz çifti imgeleri elde edilmiştir. Ardından bu göz çifti imgeleri kullanılarak asıl ilgi alanımız olan gözler elde edilmiştir. İrisin oluşturduğu dairesel yapının göz akından ayrıştırılabilmesi için histogram eşitleme ve eşikleme gibi görüntü iyileştirme yöntemleri uygulanmıştır. Bu aşamaların ardından elde edilen siyah beyaz imge üzerinde ilgilendiğimiz bölümü öne çıkaracak işlemler yapılmıştır. Devamında Canny kenar dedektörü yardımıyla kenarlar çıkarılmış ve çıkarılan kenarlar dairesel Hough dönüşümü kullanılarak gerçekleştirilen daire tespit işlemi için kullanılmıştır.

**Anahtar Kelimeler:** Göz Durumu Tespiti, Dairesel Hough Dönüşümü

## **ABSTRACT**

### **EYE STATE DETECTION IN HUMAN FACES BY USING CIRCULAR HOUGH TRANSFORM**

Nowadays eye states are used as inputs to many applications. Facial expression recognition, human-computer interaction and driver fatigue detection systems could be given as an examples to this applications. By employing these kinds of applications in academical and commercial fields, eye state detection has drawn great attention in the past decade.

In this thesis study, an eye state detection system has been developed which is based on detecting the circular structure that iris forms inside the eye by means of circular Hough transform. Initially, face and eye pair images are obtained by employing Viola-Jones object detection framework. Later, eye images are cropped from eye pair images that are acquired from the previous step. For extracting the iris from the sclera, image enhancement techniques like histogram equalization and thresholding were used. Sequel to this, some process has been made for putting forward the region of interest on the black and white image. With the help of canny edge detector, edges were obtained in the black and white images and these edges were used for circle detection with the help of circular Hough transform.

**Key Words:** Eye State Detection, Circular Hough Transform

## ŞEKİLLER LİSTESİ

Şekil 4.1.	Geliştirilen yöntemin blok diyagramı .....	11
Şekil 4.2.	ABCD dikdörtgen alanı .....	13
Şekil 4.3.	Kenar özellikleri .....	13
Şekil 4.4.	Çizgi özellikleri .....	13
Şekil 4.5.	Ortayı çevreleyen özellikler .....	13
Şekil 4.6.	Örnek AdaBoost uygulaması .....	16
Şekil 4.7.	Nihai sınıflandırıcı .....	16
Şekil 4.8.	Elemeli Kaskad yapı .....	18
Şekil 4.9.	Viola-Jones nesne tespit sistemi ile elde edilen göz imgeleri .....	19
Şekil 4.10.	Viola-Jones nesne tespit sistemi ile elde edilen göz çifti imgeleri .....	19
Şekil 4.11.	Göz çifti imgelerinden kırılarak çıkarılan göz imgeleri .....	19
Şekil 4.12.	Göz çifti tespit edilemeyen yüz imgeleri .....	19
Şekil 4.13.	Dairesel Hough dönüşümü için kullanılacak parametre uzayı .....	25
Şekil 4.14.	3 nokta ve sabit bir yarıçap ile yapılan dairesele Hough dönüşümü .....	26
Şekil 5.1.	Yüz tespit edilen resimlere ve yüz imgelerine örnekler .....	28
Şekil 5.2.	Yüz tespit edilemeyen resimlere örnekler .....	28
Şekil 5.3.	Viola-Jones nesne tespit sistemi ile elde edilen göz imgeleri .....	29
Şekil 5.4.	Viola-Jones nesne tespit sistemi ile elde edilen göz çifti imgeleri .....	29
Şekil 5.5.	Göz çifti imgelerinden kırılarak çıkarılan göz imgeleri .....	30
Şekil 5.6.	Göz çifti tespit edilemeyen yüz imgeleri .....	30
Şekil 5.7.	Göz imgesi histogram eşitleme .....	31
Şekil 5.8.	Eşikleme .....	32
Şekil 5.9.	İkili imgelerin doldurma ve filtrelemeden sonraki durumları .....	32
Şekil 5.10.	Kenar çıkarma .....	33
Şekil 5.11.	Açık olarak tespit edilen açık göz örnekleri .....	34
Şekil 5.12.	Kapalı olarak tespit edilen açık göz örnekleri .....	34
Şekil 5.13.	Kapalı olarak tespit edilen kapalı göz örnekleri .....	34
Şekil 5.14.	Açık olarak tespit edilen kapalı göz örnekleri .....	35



## TABLÖLÄR LİSTESİ

Tablo 5.1.	Yüz konumu tespiti başarı oranları .....	28
Tablo 5.2.	Göz çifti konumu tespiti başarı oranları .....	29
Tablo 5.3.	Göz durumu tespiti başarı oranları .....	36

## DENKLEMLER LİSTESİ

Denklem 4.1. İntegral imge gösterimi.....	12
Denklem 4.2. İntegral imge kullanılarak alan hesaplama.....	12
Denklem 4.3. $i$ gri seviyesindeki bir pikselin görüntü içerisinde bulunma olasılığı .....	21
Denklem 4.4. Kümülatif dağılım fonksiyonu (KDF) ve $p_x$ .....	21
Denklem 4.5. K katsayısının hesaplanması.....	21
Denklem 4.6. Lineer $y = T(x)$ dönüşümü .....	21
Denklem 4.7. $xy$ düzlemi üzerindeki doğru denklemi.....	21
Denklem 4.8. Göz durumu tespiti başarı oranları .....	24
Denklem 4.9. Parametre uzayında doğru denklemi .....	25
Denklem 4.10.Daire denklemi .....	25
Denklem 4.11.Dairenin parametrik gösterimi .....	25

## 1. GİRİŞ

Göz durumu, günümüzde bir çok uygulamaya giriş verisi olarak kullanılmaktadır. Yüz ifadesi tanıma sistemleri, sürücü yorgunluk tespit sistemleri ve insan-bilgisayar etkileşimi bu uygulamalara verilebilecek birkaç örnektir. Yüz ifadesi tanıma sistemlerinde, insanların gülmek, ağlamak, şaşırmaq gibi davranışları sergiledikleri esnada göz açıklıklarının tespit edilmesi ve derecelendirilmesiyle buldukları ruh hallerine ilişkin çıkarımlar yapılmaktadır. Sürücü yorgunluk tespit sistemlerinde göz kırpma sayısı, göz kırılması esnasında geçen süre ve göz kırpma sıklığı gibi bilgilerin yardımıyla sürücünün uykulu olup olmadığı tespit edilmekte ve gerektiği zaman sürücüyü uyarmak için ikazlar gönderilmektedir. İnsan bilgisayar etkileşiminde ise bilgisayarlar, gözün bakışlarını yönlendirdiği yer ve yaptığı hareketleri analiz ederek kullanıcıların isteklerini anlamakta ve bu istekleri yerlerine getirmektedirler. Bu tarz uygulamaların akademik ve ticari alanlarda kullanılmasıyla göz durumu tespiti son birkaç yılda büyük bir ilgiyle karşılaşmıştır.

Göz durumu tespiti için yapılan çalışmalarda iris, göz çevresi, göz kenarı ve göz kapaklarından yararlanılmaktadır. Wang ve Yang [1], chain code tracing ile siyah-beyaz göz resminden kenarları çıkardıktan sonra iris dairesini tespit etmek için Hough dönüşümünü kullanmışlardır. Tian, Kanade ve Cohn [2], göz durumu tespiti için Gabor dalgacıkları ve yapay sinir ağlarından faydalanmıştır. Seneratne v.d. [3], karar destek makineleri ve naive bayes sınıflandırıcılarını kullanarak göz bloklarının yoğunluk değerlerini karşılaştırmıştır. Xu, Zeng ve Wang [4] Local Binary Pattern (LBP) olarak adlandırılan histogram özelliğini ve Adaboost tabanlı kaskad sınıflandırıcıyı kullanarak göz durumunu tespit etmeye çalışmıştır.

Bu çalışmanın devamı şu şekildedir. Bölüm 2’de yüz tespiti probleminin tanımı ve yüz tespiti üzerine yapılan çalışmalardan bahsedilmektedir. Bölüm 3’de göz tespiti problemi ve göz tespiti üzerine yapılan çalışmalardan bahsedilmektedir. Bölüm 4’de yüz ve göz tespiti için kullanılan yöntem ile beraber göz tespit için geliştirilen yöntem hakkında bilgi verilmektedir. Bölüm 5’de geliştirilen yöntem uygulanmaktadır. Bölüm 6’da uygulama sonuçları hakkında bilgi verilmektedir.

## 2. YÜZ TESPİTİ

Bilgisayarlar; işlem güçlerinin artması ve algılama, analiz etme, yorumlama yeteneklerinin gelişmesiyle gitgide daha akıllı makinelere dönüşmektedirler. Birçok araştırma projesi ve ticari ürün, bilgisayarların kameralar yardımıyla insanlara bakarak veya mikrofonlar vasıtasıyla onları dinleyerek insanlarla etkileşime girebileceklerini göstermişlerdir [5]. Yüz tespiti bu tarz bir insan bilgisayar etkileşiminin temel taşlarından birisidir.

Yüz tespitinin amacı, verilen bir resimde yüz olup olmadığının kontrol edilmesi ve eğer var ise her bir yüzün koordinatlarının geri döndürülmesidir [6]. İnsanlar için çok basit bir işlem olan yüz tespiti bilgisayarlar için bir çok problemin çözülmesi anlamına gelmektedir.

### 2.1. Yüz Tespiti Problemleri

Yüz tespiti ile yakından ilişkili birçok problem vardır. Bu problemlerin bir çoğunun çözümü için öncelikle yüzün tespit edilmesi gerekmektedir. Bu problemler şu şekilde sıralanabilir:

- **Yüz yerelleştirme (Face localization):** Verilen resimde sadece bir yüz bulunduğu varsayılarak bu yüzün yeri bulunmaya çalışılır
- **Yüz özelliklerinin tespiti (Detection of facial features):** Verilen resimde sadece bir yüz bulunduğu varsayılarak bu resimde göz, burun, burun deliği, kaş, ağız, dudak, kulak v.b. özelliklerin yerleri tespit edilmeye çalışılır.
- **Yüz tanıma (Face recognition):** Girilen bir yüz resmi veritabanıyla karşılaştırılarak olası eşleşmeler gösterilir.
- **Yüz doğrulama (Face verification):** Girilen bir yüz resminin iddia edilen kişi olup olmadığının tespit edilmesidir.
- **Yüz takibi (Face tracking):** Sürekli bir biçimde alınan görüntü üzerinde eş zamanlı yüz konumu tespiti yapılmasıdır.
- **Yüz ifadesi tanıma (Facial expression recognition):** Verilen yüz resmi üzerinde hangi yüz ifadesinin mevcut olduğunun (mutlu, üzgün, iğrenmiş v.b.) tespit edilmesidir.

Her örüntü tanıma probleminde olduğu gibi yüz tespiti probleminin de zorlukları vardır. Bu zorluklar şu parametrelerle açıklanabilir:

- **Poz:** Görüntülerdeki yüzler ön cepheden, yandan ya da çeşitli ara açılardan görülebilmektedir. Kameranın yüzü gördüğü açıya göre burun ya da gözler gibi bazı yapısal özelliklerin görünümü değişmektedir. Bu özellikler bazı açılar için kısmen görülebilmekte, bazıları için ise hiç görünmemektedir.

- **Yapısal bileşenlerin ve aksesuarların varlığı veya yokluğu:** Sakal ve bıyık gibi yüze dahil olabilecek yapısal bileşenler ile gözlük gibi aksesuarlar yüz tespitini zorlaştıran etkenlerdir. Ayrıca boyutlarına bağlı olarak bu işlemi daha da zorlaştırabilirler.

- **Yüz ifadeleri:** Yüz ifadesine göre yüz nesnesinin görünümü değişkenlik gösterebilmektedir.

- **Kapanma:** Yüzlerin üstü başka objeler tarafından kapanabilir. Örnek olarak bir çok yüzün bulunduğu bir resimde bir yüzün başka yüzler tarafından kapanması verilebilir.

- **Resim yönü:** Alınan resmin kameranın optik eksenine göre yaptığı açıdan dolayı resmin yönü değişmiş olabilir.

- **Resimin elde edilme durumları:** Işık (spektrum, kaynak dağılımı, ışık şiddeti, gölge) ve kamera özellikleri (sensör tepkisi, lensler, odak) elde edilen resim üzerinde değişikliklere neden olur.

## 2.2. Yüz Tespiti Çalışmaları

Yüz tespiti alanında son on yıl içerisinde büyük gelişmeler olmuştur. Özellikle Viola-Jones [7] tarafından geliştirilen sistem, yüz tespitinin dijital kameralar ve fotoğraf düzenleme yazılımları gibi gerçek zamanlı uygulamalar içerisinde pratik olarak kullanılmasına olanak sağlamıştır. Yüz tespiti alanında yapılan çalışmalar Yang v.d. [6] tarafından dört kategoride toplanmıştır:

- Bilgi Tabanlı Yöntemler
- Değişmeyen Özellik Yaklaşımları
- Şablon Eşleştirme Yöntemleri
- Görünüm Tabanlı Metotlar

### 2.2.1. Bilgi Tabanlı Yöntemler

Bilgi tabanlı yöntemler (Knowledge based methods), tipik bir insan yüzünü oluşturan yapısal parçaların belli kurallar yardımıyla çözümlenmesi ile çalışmaktadır. Belirlenen kurallar yüzü oluşturan parçaların birbirleriyle olan ilişkilerini tanımlamaktadır. Örneğin, bir insan yüzünde genellikle birbirlerine göre simetrik duran iki göz, bir ağız ve bir burun bulunmaktadır. Bu yapısal parçalar arasındaki ilişkiler birbirlerine göre uzaklıkları ve pozisyonları kullanılarak tanımlanabilir. Yüz nesnesinin tespitinde öncelikle verilen resimde bu yapısal parçalar aranmakta daha sonra kurallara uyan aday bölgeler belirlenmektedir. Hatalı tespit yapılmasını engelleyebilmek için genellikle belirlenen aday bölgeler tekrar incelenip bunlar üzerinde doğrulama işlemi yapılmaktadır.

Bu yöntemin kullanımında beliren en önemli problem kuralların uygun şekilde tanımlanmasının zorluğudur. Kurallar fazla detaylı ve katı olursa bazı yüzler tüm kuralları geçememekte ve tespit edilememektedir. Buna karşın kuralların çok genel tanımlanması durumunda birçok hatalı tespit yapılabilmektedir. Ayrıca diğer bir problem de resimdeki yüzlerin değişik kafa yönelimlerine sahip olabilmesidir. Değişik kafa yönelimlerinde yapısal parçalar arasındaki ilişkiler değişmekte ve olası her durumu tanımlayabilmek mümkün olamamaktadır. Bu nedenle bu yöntem genelde ön cepheden çekilmiş ve karmaşık bir arka plana sahip olmayan resimler için daha başarılı sonuçlar vermektedir.

Belirtilen problemlerin bir kısmını aşmak için Yang ve Huang [8] hiyerarşik bir model geliştirmiştir. Test edilecek resmin birbirinden farklı düşük çözünürlüklü kopyaları elde edilmekte ve bunların her birine farklı kurallar uygulanmaktadır. En düşük çözünürlüklü resimlere daha genel kurallar uygulanmakta ve bu kurallardan geçen aday bölgelerin daha yüksek çözünürlüklü versiyonlarına daha detaylı kurallar uygulanmaktadır. Kullanılan hiyerarşik yaklaşım sayesinde gerekli hesaplama miktarından tasarruf sağlanmaktadır. Yöntem çok başarılı tespit oranları vermese de hiyerarşik yaklaşımın daha sonraki yüz tespiti çalışmalarında da yer bulması açısından önemlidir.

### 2.2.2. Değişmeyen Özellik Yaklaşımları

Değişmeyen özellik yaklaşımlarında (Feature invariant methods), kafa yönelimi ya da ışıklandırma gibi koşullar değişse bile bazı yapısal özellikler ile yüz tespit edilebilmektedir. Kaşlar, gözler, burun, ağız, saç hattı gibi yapısal özelliklerin yerleri genellikle kenar çıkarıcılar (edge detector) yardımı ile belirlenmektedir. Bu yapısal parçalar arasındaki ilişkileri tanımlayan bir istatistiksel model hazırlanmakta ve hazırlanan model kullanılarak yüz tespiti yapılmaktadır. Ayrıca deri rengi, yüz şekli veya yüz ve saç dokusu gibi özellikler de yüz tespiti için kullanılabilir. Bu global özellikler resim içerisinde aday bölgeler belirlemek için kullanılmakta daha sonra kaş, göz, ağız gibi yerel detaylı özellikler ile bu bölgelerin kontrolü yapılmaktadır.

Bu yöntemdeki en önemli problem algoritmaların başarısının aydınlatma, gürültü ve tespit edilecek yüzün bir kısmının başka bir şey tarafından kapatılması gibi durumlardan aşırı etkilenmesidir. Bu durumlarda oluşan gölgeler, gerçekte olmayan ayrıtların belirmesine neden olabilmekte veya ayrıtların sınırlan belirsizleşebilmekte ve dolayısıyla yöntem kullanışsız hale gelebilmektedir.

Sirohey [9] karmaşık bir arkaplandan yüzü ayırtıran bir lokalizasyon yöntemi geliştirmiştir. Canny kenar dedektörü ile elde edilen kenarlar gruplanarak sadece yüze ait olanlar belirlenmektedir. Baş ile arkaplan arasındaki sınıra uygun bir elips oturtulmaktadır.

Leung, Burl ve Perona [10] yerel öznitelik tespit edicileri kullanarak karmaşık bir arkaplanda yüz lokalizasyonu yapan istatistiksel bir yöntem geliştirmiştir. Tipik bir yüzü tanımlamak için iki göz-iki burun deliği ve burun-dudak birleşim bölgesi (junction) olmak üzere beş adet öznitelik kullanılmaktadır. Bir grup resim setinden herhangi iki öznitelik çiftinin birbirlerine göre uzaklığı hesaplanarak bir gauss dağılımı ile modellenmektedir. Görüntü setinin çok yönlü (multiorientation) ve çok boyutlu (multiscale) türevsel gauss filtrelerine gösterdikleri ortalama tepkilerden bir yüz şablonu oluşturulmaktadır. Test resmi üzerinde bu filtrelere yüksek tepki gösteren noktalar aday öznitelik noktalarını vermektedir. En yüksek tepkiyi veren iki aday öznitelik noktasında hazırlanmış olan istatistiksel model ile diğer noktalar kestirilmektedir.

### 2.2.3. Şablon Eşleştirme Yöntemleri

Şablon eşleştirme yöntemlerinde (Template matching methods), genelde ön cepheden standart bir yüz modeli, elle ya da bir fonksiyon kullanılarak tanımlanmaktadır. Verilen test resmi ile model arasındaki ilinti yüz hattı, gözler, ağız ve burun için ayrı ayrı hesaplanmaktadır. İlintinin yüksek olması durumunda yüz tespit edilmiş olmaktadır. Gerçekleştirmenin basit olması avantaj sağlamakla birlikte boyut, kafa yönelimi ve şekil değişimlerinden fazla etkilenmesi yöntemin önemli bir dezavantajıdır. Dezavantajları çözümlenmek amacıyla çok çözünürlüklü (multiresolution) şablonlar, çok boyutlu (multiscale) şablonlar, alt şablonlar (subtemplates) ve deforme edilebilen şablonlar (deformable templates) tanımlanmıştır. Buna rağmen yöntem yüz tespitinde çok iyi sonuçlar verememektedir.

Sakai [11] ön cepheden görülen yüzlerin tespiti için bir yöntem geliştirmiştir. Gözler, ağız, burun, ve yüz kenarının çizgi parçaları (line segments) ile tanımladığı alt şablonlar hazırlamıştır. İki aşamalı yönteminde, aday bölgeler işlenmiş resimle kenar şablonunun ilintisine göre belirlenmekte daha sonra diğer alt şablonların eşleştirilmesiyle detaylı bir analiz yapılarak yüz tespiti tamamlanmaktadır.

Cravv [12,13] lokalizasyon problemini çözümlenmek için şekilsel şablon kullanarak ön cepheden yüz modeli geliştirmiştir. Yöntem test resimlerinden Sobel filtreleri ile kenarlar çıkarmakta ve yüz konturunu oturtuktan sonra gözler, kaşlar gibi diğer özniteliklerin lokalizasyonunu sağlayabilmek için arka arkaya uygulanmaktadır.

Samal ve Iyengar [14] temel bileşen analizi (PCA, principle component analysis) yöntemiyle bir dizi yüz silueti elde etmektedir. Elde edilen siluetler genelleştirilmiş Hough dönüşümü ile birlikte kullanılarak yüz lokalizasyonu yapmak için test resimlerine uygulanmaktadır.

Yuille [15] ve Lanitis [16] ise geliştirdikleri yöntemde deforme edilebilen şablonlar kullanmaktadır.



#### 2.2.4. Görünüm Tabanlı Yöntemler

Görünüm tabanlı yöntemlerde (Appearance based methods), şablon eşleştirme yönteminden farklı olarak şablon elle hazırlanmamakta, bir grup örnek setinden elde edilmektedir. Nesne içeren ve nesne içermeyen görüntüler ile ilgili karakteristikler istatistiksel dağılımlar ya da ayırtaç (discriminant) fonksiyonları ile ifade edilmektedir. Hesaplama miktarından tasarruf sağlamak ve tespit oranlarını iyileştirmek için genelde boyut azaltımı uygulanmaktadır. Görünüm tabanlı yöntemler kendi içlerinde eigenfaces (öz yüzler), dağılım tabanlı yöntemler (distribution-based methods), sinir ağları (neural networks), destek vektörü yöntemi (SVM, support vector machines), Bayes sınıflandırıcı (naive Bayes classifier) ve saklı Markov modeli (HMM, hidden Markov model) olarak gruplara ayrılabilir.

Öz vektörler (eigenvectors) kullanarak yüz tanıma uygulaması ilk olarak Kohonen [17] tarafından gerçekleştirilmiştir. Geliştirdiği basit sinir ağı, yüz görüntülerine ait otokorelasyon matrisinin öz vektörlerinden bir yüz tanımı çıkarmaktadır. Bu öz vektörler daha sonra öz yüzler olarak adlandırılmıştır. Turk ve Pentland [18] ise PCA yöntemi kullanarak yüz görüntülerinden oluşan bir eğitim setinden öz yüzler oluşturulmuştur. Oluşturulan öz yüzler resim uzayının yüz uzayı (face space) olarak adlandırılan bir alt uzaya izdüşümüdür. Yüz içeren görüntülerden elde edilen izdüşüm belli noktalarda, yüz içermeyen görüntülerden elde edilen izdüşüm çok değişken noktalarda odaklanmaktadır. Bir görüntüde yüz tespiti yapmak için görüntünün her noktasında görüntü uzayı ile yüz uzayı arasındaki mesafe hesaplanmakta ve böylece yüz haritası (face map) çıkarılmaktadır. Bu haritanın yerel minimumu yüzün bulunduğu konumu belirlemektedir.

Sung ve Poggio [19,20] yüz sezimi için dağılım tabanlı bir sistem geliştirmiştir. Geliştirilen sistem, dağılım tabanlı modelleme ve çok katmanlı perseptron sınıflandırıcı (MLP, multilayer perceptron classifier) olmak üzere iki aşamadan oluşmaktadır. Dağılımlar yüz içeren ve içermeyen resimlerden oluşan bir görüntü setinden elde edilmektedir. Yüz içermeyen resimlerin toplanması için bootstrap yöntemi kullanılmıştır. Yüz içermeyen küçük bir resim grubu ile eğitilen MLP sınıflandırıcı rasgele bir grup resimde test edilmektedir. Bu resimlerde hatalı olarak sınıflandırılmış yüz içermeyen görüntüler eğitim setine eklenmekte böylece daha başarılı bir sınıflandırıcı tasarımı

mümkün olmaktadır. Bootstrap adı verilen bu yöntem daha sonraki çalışmalarda da kullanılmıştır.

Sinir ağları nesne tespit uygulamalarında yaygın olarak kullanılmaktadır. Kompleks koşullu olasılık dağılımlarına uygun olarak sistem eğitimi yapılmasına imkan vermesi yöntemin önemli bir avantajıdır. Ancak tatmin edici sonuçlar elde edilebilmesi için katman sayısı (number of layers), kullanılan düğüm sayısı (number of nodes) ve öğrenme oranı parametrelerinin uygun şekilde ayarlanmış olması gerekmektedir. Yapılmış çalışmalardan en dikkat çekici olanları Rowley [21,22] tarafından gerçekleştirilmiştir.

Bayes sınıflandırıcı kullanılarak yapılmış önemli çalışmalardan biri Schneiderman ve Kanade [23] tarafından gerçekleştirilmiştir. Görsel özellikler ile ilgili istatistikler histogramların çarpımı ile gösterilmektedir. Her bir histogram bir grup dalgacık katsayısı ile bunların nesne üzerindeki birleşik istatistiksel dağılımlarını vermektedir. Farklı yönelimlere sahip nesnelerin tespiti için her bir yönelime özel olarak birer sınıflandırıcı tasarlanmaktadır. Bu sınıflandırıcılardan gelen sonuçlar Adaboost algoritması kullanılarak birleştirilmekte ve düşük yanlış tespit oranına sahip bir sınıflandırıcı elde edilmektedir. Böylece ön cepheden, yandan ya da farklı ara açılardan görülen yüzlerin tespiti, yüksek doğruluk tespit oranı ile yapılabilmektedir. Yazılım aynı zamanda araç tespiti için de uygulanmıştır.

Görünüm tabanlı bir diğer önemli çalışma da Viola-Jones [7] tarafından gerçekleştirilmiştir. Birçok nesne tespit yönteminde tespitçi, imgenin her noktasında ve olası tüm ölçeklerde uygulandığından çok fazla hesaplama ihtiyacı duyulmaktadır. Taranan sahnedeki konumların çoğu yüz imgesi içermediğinden her yerde yüz bulma olasılığı çok düşüktür. Dolayısıyla yüz içermeyen imge bloklarının çoğu, basit sınıflandırıcılar kullanılarak elenebilir. Bu çalışmada böyle bir eleme stratejisi ile art arda eleyicilerden oluşan çok katlı sınıflandırıcı geliştirilmiştir. Haar dalgacık özniteliklerine dayalı bu basit sınıflandırdılar birleşerek düşük yanlış ret oranlı ve yüksek doğru tespit oranlı sonuçlar üretmektedir. Kullanılacak olan öznitelikler Adaboost algoritması ile seçilmekte ve gereken toplam öznitelik sayısı hedeflenen yanlış kabul oranına bağlı olarak değişmektedir.

### 3. GÖZ TESPİTİ

Göz tespiti problem tıpkı yüz tespiti problemi gibi bir örüntü tanıma problemidir. Yüz tespiti probleminde bütün imge üzerinde yüz aranırken, göz tespiti probleminde ise sadece belirlenen yüz imgesi üzerinde arama yapılmaktadır. Göz tespiti alanındaki yaklaşımları kaba bir şekilde üç sınıfa ayrılabiliriz:

- Şablon tabanlı yaklaşımlar
- Özellik tabanlı yaklaşımlar
- Görünüm tabanlı yaklaşımlar

#### 3.1. Şablon Tabanlı Yaklaşımlar

Şablon tabanlı yaklaşımlarda göz şeklinin bilgisinden faydalanılarak göz tespiti yapılır. Öncelikle göz şeklinin bilgisi yardımıyla bir şablon oluşturulur. Ardından bu şablon görüntü penceresine eşleştirilerek göz konumu tespit edilir. Yuille v.d. [24], iris bölgesini ve göz kapaklarını kullanarak bir şablon tasarlamışlardır. Şablonun göz bölgesi ile eşleşmesi için 11 farklı parametre yardımıyla irisin merkezinin yönü, boyutu ve pozisyonu değiştirilmiştir. Kenarlar, vadiler ve tepeler yardımıyla enerji fonksiyonları belirlenmiştir ve bu fonksiyonların aldıkları değerler şablonun göz ile örtüşüp örtüşmediği tespit etmek amacıyla kullanılmıştır.

Bu yaklaşım her ne kadar doğru sonuçlar verse de örtüşmenin yavaş olması, uzun çalışma süresi ve farklı kafa pozisyonlarına uyum sağlayamaması gibi nedelerle tercih edilmemektedir. Bu sorunları çözmek için Lam ve Yan [25], göz kenarlarını tespit eden bir şablon sunmuşlardır. İlk şablon göz kenarları bilgisi kullanılarak oluşturulmuştur. Optimizasyon işlemi için ise hızlı bir algoritma olan greedy algoritması yardımıyla aktif kenar modelleme kullanılmıştır. Li, Qi ve Yang [26] ise göz tespiti için dört farklı bölgenin (göz dışı, göz içi, göz kapağı ve göz kuyruğu) şablonunu kullanmıştır. Bu yöntemde eşleştirme, orijinal şablon ile resimden çıkarılan şablonun her bir bölgesinin örtüşme yüzdesinin ölçülmesiyle yapılır.

### 3.2. Özellik Tabanlı Yaklaşımlar

Özellik tabanlı yaklaşımlar yüz üzerinde göz tespiti için siyah olan göz bebeği, beyaz olan göz akı, irisin yuvarlak şekli ve göz kapaklarının elipsellikleri gibi ayrıştırıcı olan özellikleri kullanırlar. Sirohey ve Rosenfel [27] göz tespiti için Gabor dalgacıklarını kullanmışlardır. İlk aşamada gabor dalgacıkları tabanlı lineer filtreler, göz köşelerinin tespiti için kullanılmıştır. İkinci aşamada ise lineer olmayan Gauss filtresi yardımıyla iris tespit edilmiştir. Kawaguchi, Hidaka ve Rizon [28] ise Hough dönüşümünü kullanarak göz tespiti yapmışlardır. Peng v.d. [29] ise göz simetrik özellikleri kullanarak göz çiftlerinin yerini tespit etmiştir.

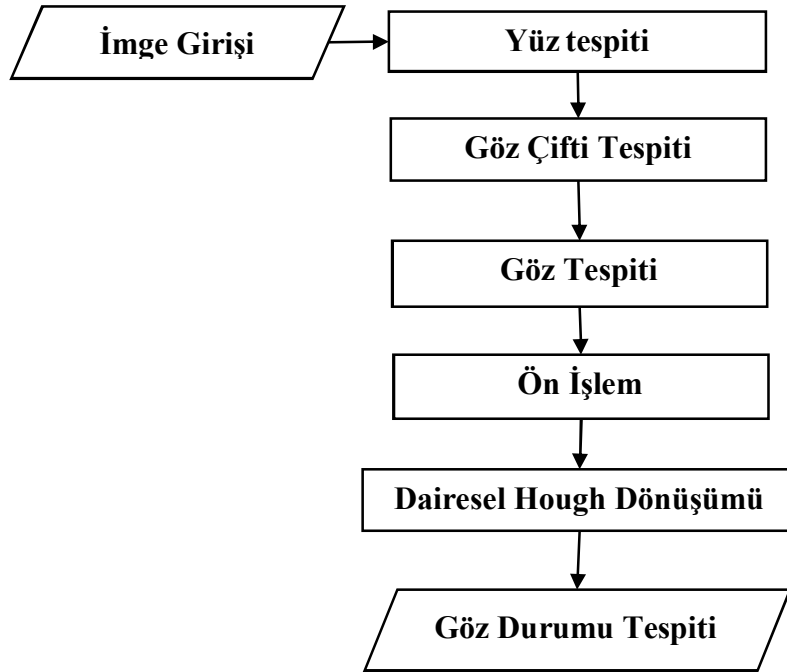
### 3.3. Görünüm Tabanlı Yaklaşımlar

Görünüm tabanlı yaklaşımlar, çok sayıda denekten farklı durumlarda (kafa ve göz pozisyonları, ışıklandırma) alınan bir çok göz örneğini, yapay sinir ağları veya karar destek makineleri yardımıyla sınıflandırmaya çalışırlar. Balya ve Roska [30] hücresele yapay sinir ağıyla insan yüzünü tespit etmeye çalışmışlardır. Yüz tespit edilip göz konumu da kabaca tespit edildikten sonra gri seviye özelliklerini kullanan başka bir hücresele yapay sinir ağıyla gözler tespit edilmiştir. Huang ve Wechsler [31] sıkıştırılmış göz görüntülerini tespit etmek için optimal dalgacık paketlerini kullanmışlardır. Ayrıca göz ve göz olmayan alanların tespiti için radial basis fonksiyonlarını kullanmışlardır. Peng v.d. [32] ayırt edici özellikler yardımıyla göz özelliklerini nitelendirmişlerdir. Ayırt edici özellikleri olasılıksal bir sınıflandırıcıya sunarak bu özelliklerin dağılımı tabanlı göz veya göz olmayan desenleri tespit edilmiştir. Ardından Adaboost yardımıyla bir çok sınıflandırıcı seçilerek en son sınıflandırıcı yapılandırılmıştır. Zhu ve Ji [33] kızıl ötesi ışıklandırmada göz bebeği özellikleri ve karar destek makinaları yardımıyla göz tespiti yapmıştır.

Yukarıda anlatılan yöntemlerin hepsi göz bebeğini göz alanı içerisinde düşük yoğunluklu bir bölge olarak tanımlamaktadırlar. Zayıf ışıklandırma ve renkli iris gibi şartlar altında bu yöntemlerden sadece birini kullanarak isabetli sonuçlara ulaşamamaktadır.

## 4. GELİŞTİRİLEN YÖNTEM

Bu bölümde göz durumu tespiti için geliştirilen yöntem anlatılacaktır. Çalışmamızda BioID [34] veritabanı kullanılmıştır. BioID veritabanı 23 farklı kişiden alından 1521 gri seviye resimden oluşmaktadır. BioID veritabanı farklı ışıklandırmalara, artalanlara ve yüz boyutlarına yer vererek gerçek dünya koşullarını sağlamaktadır. Geliştirme ortamı olarak Matlab kullanılmıştır. Çalışmamızda geliştirdiğimiz yöntemin blok diyagramı Şekil 4.1'deki gibidir.



Şekil 4.1. Geliştirilen yöntemin blok diyagramı

### 4.1. Yüz Konumu Tespiti

Yüz konumu tespiti için çalışmamızda Viola-Jones [7] tarafından tasarlanan nesne tespit sistemi kullanılmıştır. Viola-Jones nesne tespit sistemi, gerçek-zamanlı nesne tespiti için en yüksek tespit oranları veren ilk nesne tespit sistemidir. Birçok nesnenin tespiti için kullanılabilir olan bu sistem yüz tespiti problemi için yaygın olarak kullanılmaktadır.

#### 4.1.1. Viola-Jones Nesne Tespit Sistemi ile Yüz Tanıma

Gerçek zamanda çalışabilecek bir yüz dedektörü olan Viola-Jones nesne tespit sistemi üç temel bileşenden oluşur. Bunlar integral imge, AdaBoost ile eğitilen bir sınıflandırıcı ve elemeli kaskad yapısıdır.

##### 4.1.1.1. İntegral İmge

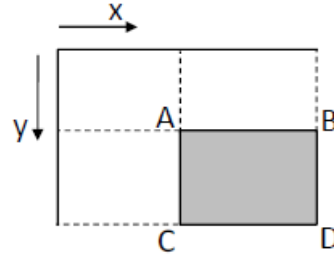
Toplanmış alan tablosu olarak da bilinen integral imge, bir ızgaranın alt kümeleri olan dikdörtgen alanların değerlerini, hızlı ve etkili bir biçimde hesaplamaya yarar. İntegral imge ilk olarak mipmaplerde kullanılmak üzere Crow [35] tarafından ortaya çıkarılmıştır. Viola-Jones, Haar benzeri özellikleri seri bir şekilde hesaplamak için integral imgeyi kullanmışlardır. İntegral imge denklem 4.1'de gösterildiği şekilde oluşturulur.

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y') \quad (4.1)$$

Burada  $ii(x, y)$ ,  $(x, y)$  konumundaki integral imgedir ve  $i(x', y')$  ise orijinal imgedir. İntegral imgeyi kullanarak herhangi bir dikdörtgen alanın toplamını hesaplamak son derece basittir. Şekil 4.2'de gösterilen ABCD dikdörtgen alanının piksel değerlerinin toplamı 4 dizi referansı ile denklem 4.2'de gösterildiği şekilde hesaplanabilir.

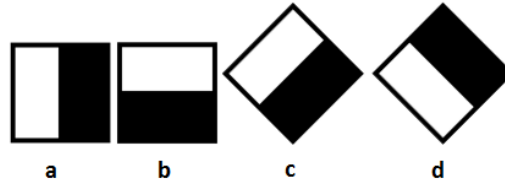
$$\sum_{(x,y) \in ABCD} i(x, y) = ii(D) + ii(A) - ii(B) - ii(C) \quad (4.2)$$

İntegral imge basit Haar benzeri özellikleri hesaplamak için kullanılabilir. Haar benzeri özellikler, nesne tanıma alanında kullanılan dijital imge özellikleridir ve isimlerini Haar dalgacıklarından almışlardır. Papageorgiou, Oren ve Poggio [36] tarafından ortaya atılan imgelerde özellik tespiti için piksel değerlerinin kullanımına alternatif olarak Haar dalgacıklarını kullanan özellik kümesi kullanılması fikri, Viola-Jones tarafından benimsenmiş ve uygulamaya konulmuştur. Haar benzeri özellikler birbirlerine bağlı siyah ve beyaz karelerden oluşurlar.

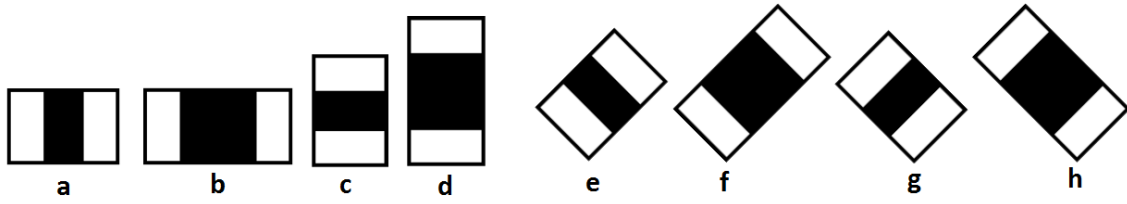


Şekil 4.2. ABCD dikdörtgen alanı

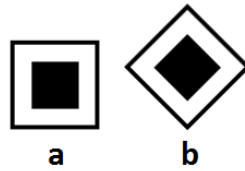
Bir Haar benzeri özelliğin değeri, siyah ve beyaz alanlardaki piksellerin gri seviye değerleri toplamalarının farkıdır. Şekil 4.3’de gösterilen kare özelliklerin ölçümünde kareler bir kenarlarını paylaştıkları için 6 dizi referans gereklidir. Buna benzer şekilde Şekil 4.4 ve Şekil 4.5’deki çizgi özellikleri ve ortayı çevreleyen özellikler için ise 8 dizi referans gereklidir.



Şekil 4.3. Kenar özellikleri



Şekil 4.4. Çizgi özellikleri



Şekil 4.5. Ortayı çevreleyen özellikler

#### 4.1.1.2. AdaBoost Öğrenme

Boosting (Yükseltme), her biri orta doğrulukta bir çok ‘zayıf’ hipotezin birleştirilerek yüksek doğrulukta bir hipotez elde edilmesi anlamına gelir [37,38].

AdaBoost (Adaptive Boosting) algoritması genel olarak daha pratik boosting algoritmalarına doğru atılan ilk adım olarak bilinir [39].

Algoritma,  $(x_1, y_1), \dots, (x_m, y_m)$  şeklinde gösterilen ve  $m$  adet örnek içeren bir eğitim setini girdi olarak almaktadır. Eğitim setinde  $x_i$ 'ler örnekleri  $y_i$ 'ler bu örneklere ait sınıf etiketlerini göstermektedir. Bütün örneklerin oluşturduğu set  $X$  ile, örneklere ait sınıfların oluşturduğu set  $Y$  ile gösterilmektedir. Eğitim setindeki her bir örnek iki sınıftan birinde yer almakta ve sınıf etiketi -1 veya +1 olmaktadır. Algoritma eğitim setindeki her bir örneğe bir katsayı vermektedir.

Bu katsayılar seti bir dağılım ile gösterilmektedir. En başta bütün örneklere birbirine eşit katsayılar atanmaktadır. Her bir turda bu katsayılar güncellenmektedir. Eğitim setindeki  $x_i$  örneğinin  $t$  turundaki katsayısı  $D_t(i)$  olarak gösterilmektedir. Algoritma verilen  $T$  adet zayıf  $h_t$  sınıflandırıcısını sırayla çağırır. Çağrılan  $h_t$  sınıflandırıcısı  $D_t$  dağılımı kullanarak eğitilir. Eğitim setindeki tüm örnekler  $h_t$  sınıflandırıcısı ile tekrar sınıflandırılır. Sınıflandırma sonuçları kullanılarak  $h_t$  sınıflandırıcısının hata oranı  $\varepsilon_t$  ve  $\alpha_t$  katsayısı ile hesaplanır. Daha sonra tüm  $D_t(i)$  katsayıları güncellenerek  $D_{t+1}$  dağılımı hesaplanır ve bir sonraki sınıflandırıcıya geçilir.  $D_{t+1}$ 'in bir dağılım olması için güncellenen  $D_t$  katsayıları  $Z_t$  değişkeni ile normalize edilmektedir.

Tüm sınıflandırıcılar tamamlandığında nihai sınıflandırıcı  $H(x)$  belirlenmiş olmaktadır. Her bir zayıf sınıflandırıcının sonucu farklı bir  $\alpha_t$  katsayısı ile çarpılarak toplanmakta ve nihai sınıflandırıcı  $H(x)$  hesaplanan signum fonksiyonundan geçirilmesi ile elde edilmektedir.  $\alpha_t$  katsayısı ait olduğu  $h_t$  sınıflandırıcısının önemini göstermektedir. Bu nedenle nihai sınıflandırıcının sonucu belirlenirken her bir  $h_t$  sınıflandırıcısının sonucu kendisine ait  $\alpha_t$  katsayısı ile ağırlandırılmaktadır.

Daha düşük hata oranına sahip zayıf sınıflandırıcıların  $\alpha_t$  katsayısı yüksek olmakta ve böylece nihai sınıflandırıcıdaki ağırlığı fazla olmaktadır. Eğitim setindeki her örneğin sahip olduğu katsayı  $D_t(i)$  ile gösterilmekte ve o örneğin ağırlığını ifade etmektedir. Her bir turda çağrılan  $h_t$  sınıflandırıcısı  $D_t$  dağılımı ile eğitilmekte ve bütün eğitim seti  $h_t$  ile

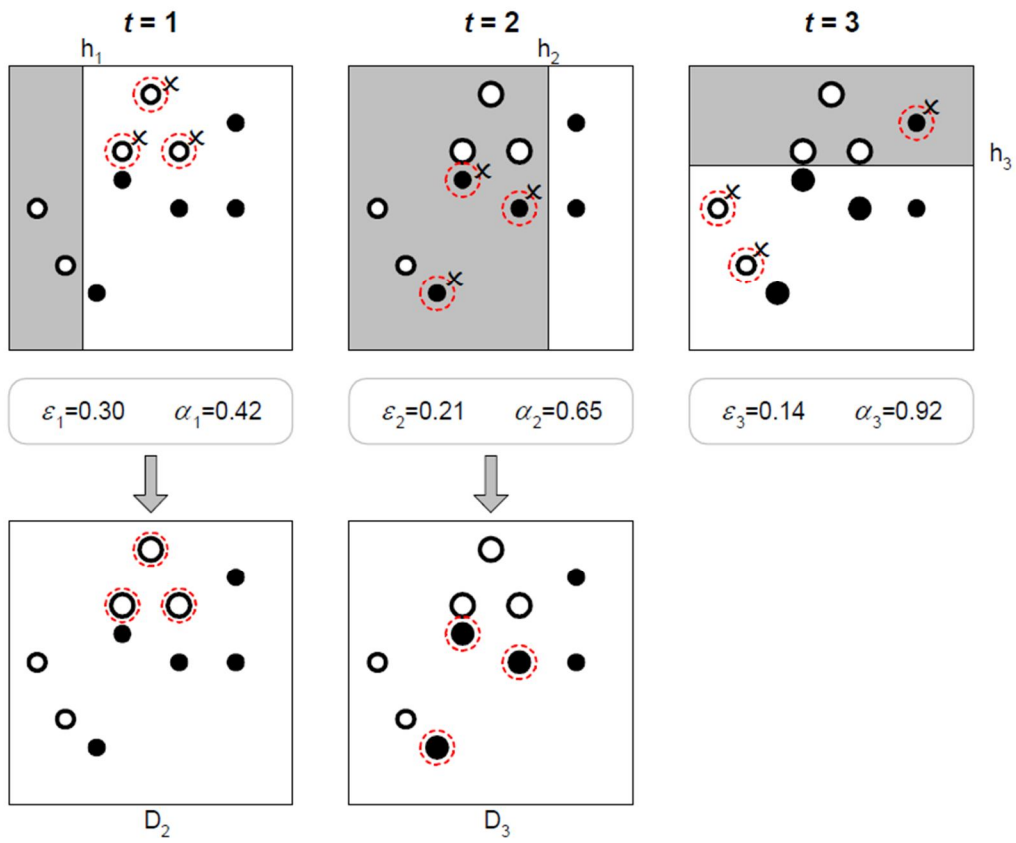


sınıflandırıldıktan sonra  $h_{t+1}$  sınıflandırıcısının eğitiminde kullanılacak olan  $D_{t+1}$  dağılımı,  $D_t$  dağılımının güncellenmesiyle elde edilmektedir. Katsayılar güncellenirken  $h_t$  sınıflandırıcısı tarafından sınıflandırılmış zor örneklerin katsayıları arttırılmaktadır. Böylece zor örneklerin ağırlığı arttırılmakta ve sınıflandırıcının bu örnekler üzerinde yoğunlaşması sağlanmaktadır.

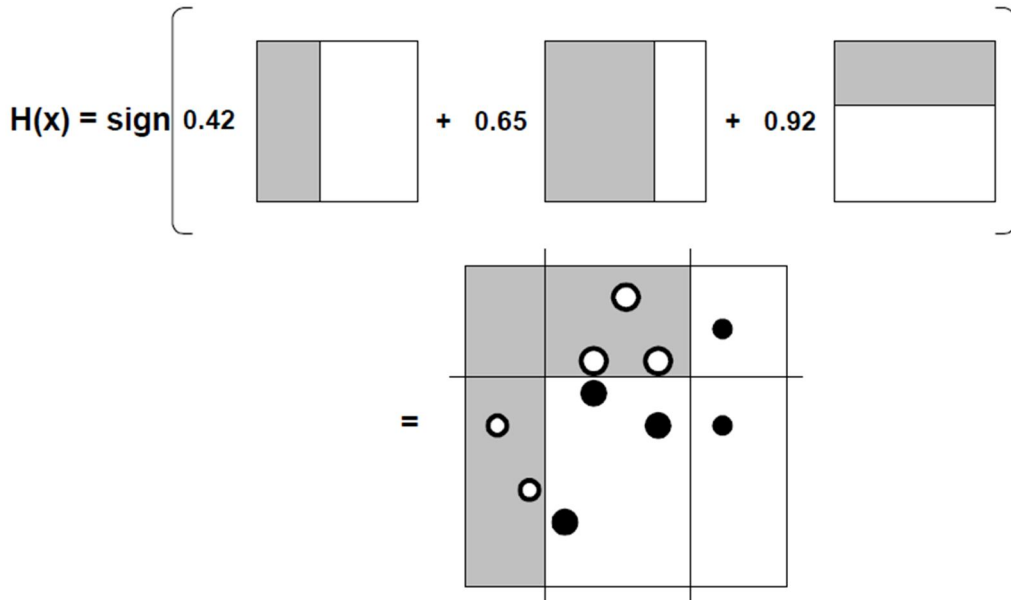
Adaboost algoritması birçok defa çalıştırılabilir. İlkinden sonraki uygulamalarda  $D_1$  dağılımı olarak bir önceki uygulamada en son güncellenen dağılım kullanılmaktadır. Adaboost algoritmasının uygulanma sayısında teorik olarak bir sınır bulunmamaktadır ve her uygulandığında hata oranının üstel olarak azalacağı Freund ve Shaphire tarafından ispatlanmıştır.

Şekil 4.6'da örnek bir Adaboost uygulaması verilmiştir.  $h_1$ ,  $h_2$  ve  $h_3$  zayıf sınıflandırıcıları kullanılarak güçlü  $H$  sınıflandırıcısı elde etmek amaçlanmaktadır. Eğitim seti içi boş ve içi dolu yuvarlaklar olmak üzere iki farklı sınıftan toplam on adet örnek içermektedir. Örneklerin katsayıları yuvarlakların boyutu ile gösterilmektedir. Katsayılar arttıkça yuvarlakların boyutu büyümektedir.

Başlangıçta bütün örneklere eşit katsayı verilmiştir.  $h_1$  sınıflandırıcısı ile eğitim seti sınıflandırıldığında üç adet içi boş yuvarlağı yanlış sınıflandırdığı görülür.  $h_1$  sınıflandırıcısına ait  $\epsilon_1$  ve  $\alpha_1$  hesaplanır, örneklerin katsayıları güncellenerek  $D_2$  dağılımı hesaplanır ve  $h_2$  sınıflandırıcısına geçilir.  $h_2$  sınıflandırıcısı ile eğitim seti sınıflandırıldığında üç adet içi dolu yuvarlağı yanlış sınıflandırdığı görülür.  $h_2$  sınıflandırıcısına ait  $\epsilon_2$  ve  $\alpha_2$  hesaplanır, örneklerin katsayıları güncellenerek  $D_3$  dağılımı hesaplanır ve  $h_3$  sınıflandırıcısına geçilir.  $h_3$  sınıflandırıcısı ile eğitim seti sınıflandırıldığında iki adet içi boş yuvarlağı ve bir adet içi dolu yuvarlağı yanlış sınıflandırdığı görülür.  $h_3$  sınıflandırıcısına ait  $\epsilon_3$  ve  $\alpha_3$  hesaplanır ve işlem tamamlanır. Oluşturulan  $H$  sınıflandırıcısı Şekil 4.7'de görülmektedir.



Şekil 4.6. Örnek AdaBoost uygulaması



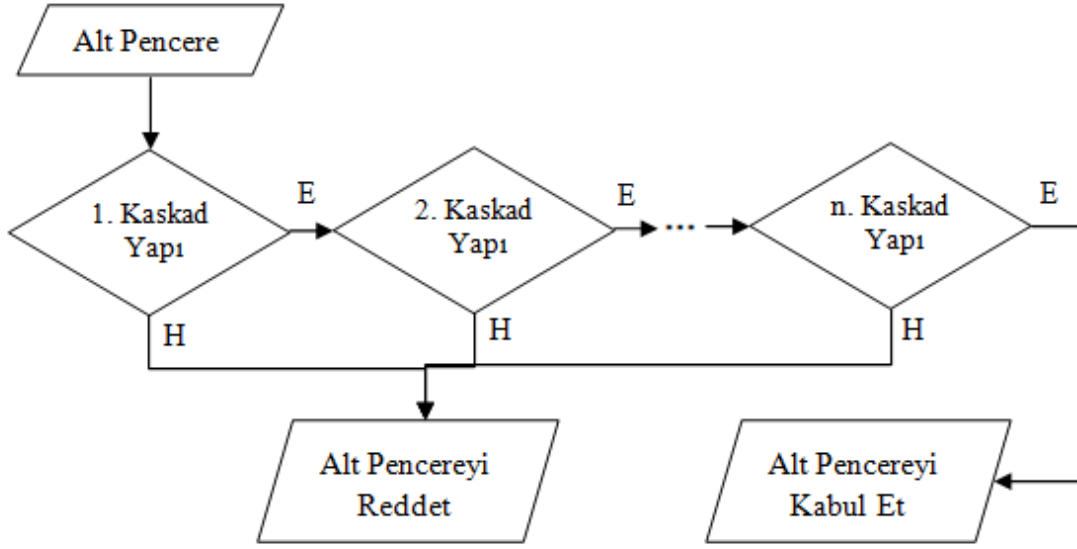
Şekil 4.7. Nihai sınıflandırıcı

#### 4.1.1.3. Elemeli Kaskad Yapı

Elemeli kaskad yapı Viola-Jones nesne dedektörünün en önemli özelliklerinden birisidir. Bu yapıdaki ana amaç daha küçük ve daha verimli olan sınıflandırıcıların, imge üzerindeki negatif alt-pencerelerin bir çoğunu reddetmek ve pozitif alt-pencerelerin hemen hemen hepsini kabul etmek üzere kullanılmasıdır. Buna bağlı olarak negatif alt-pencerelerin büyük bir kısmı kaskad yapının ilk aşamalarında elenecektir ve tespit işlemi çok daha verimli olacaktır.

Bir veya birden fazla sınıflandırıcının birleştirilerek oluşturulduğu farklı yapılar birbirleriyle birleştirilerek elemeli kaskad yapı oluşturulur. Şekil 4.8’de görüleceği gibi girdi olarak verilen alt pencereler sırası ile düğümlerden geçmektedir. Her düğümde alt-pencerenin bir sonraki aşamaya iletilip iletmeyeceğine dair ikili bir karar verilmektedir. Alt-pencere kaskad boyunca ilerledikçe her bir düğümdeki zayıf sınıflandırıcıların sayısı genellikle artmaktadır. Örneğin Viola-Jones [7] çalışmasında ilk 5 düğüm sırasıyla 1, 10, 25, 25, 50 zayıf sınıflandırıcıdan oluşmaktadır. Bu şekilde ilk düğümde gereğinden fazla doğru kabul edilip bir sonraki aşamaya iletilen alt-pencere bir sonraki aşamada daha zor bir sınıflandırılmadan geçirilmektedir ve böylece kaskad yapının sonunda doğru kabul edilen pencerelere ulaşılmaktadır.

Kaskad yapının öğrenme işlemi üzerinde de etkisi vardır. Yüz tespiti, nadir bulunan nesnenin tespit edilmesi gereken bir problemdir. Bu nedenle yüksek performanslı bir dedektörün inşası için milyonlarca negatif örneğe gerek vardır. Yüksek sayıdaki negatif örnek gereksiniminin üstesinden gelmek için Viola-Jones [7] bootstrap işlemini kullanmışlardır. Bootstrap işleminde öncelikle her düğüm için bir eşik değeri elle belirlenmektedir. Kısmi sınıflandırıcılar negatif örnek kümesini tarayarak buldukları daha az reddedilen negatif örnekleri, eğitilmeleri için bir sonraki düğüme göndermektedirler. Sonrasında ise her bir düğüm bu yeni gelen örneklerle daha önceden eğitilmemiş gibi yeniden eğitilmektedir.



Şekil 4.8. Elemeli kaskad yapısı

#### 4.2. Göz Konumu Tespiti

Yüz için tasarlanan Haar kaskadları ile yüz imgesinin bulunmasının ardından, yüz imgesine göz ve göz çifti için tasarlanan Haar kaskadları uygulanarak gözler ve göz çiftleri elde edilmiştir.

Şekil 4.9’da elde edilen göz imgeleri gösterilmektedir. Bu imgelerde gözlere ek olarak kaşların da imgeye dahil olduğu görülmektedir. Kaşlar ile gözlerin aynı imge içerisinde bulunması yöntemimizin performansını düşürebilmektedir.

Şekil 4.10’da göz çifti imgeleri gösterilmiştir. Göz çifti imgeleri göz imgelerinden daha az kaş içermektedir. Bu yüzden göz imgesi olarak göz çifti imgesinden kırılarak çıkarılacak göz imgelerinin kullanılmasının daha uygun olduğu tespit edilmiştir. Kırpma işleminde kullanılacak koordinatları elde etmek için istatistiki bir yaklaşım kullanılmıştır. İstatistiksel olarak bir göz çifti imgesinde gözlerin kapladıkları uzunluk sağ ve sol taraftan ortalama %30 ile %37 aralığındadır. %5 hata payı da ekleyerek olabilecek maksimum uzunluk %42 olmaktadır. %42 den sonrası göz ile alakasız olan burunu temsil etmektedir. Bu değerler referans alınarak yapılan göz çifti imgesini kırpma işleminden sonra sağ ve sol göz imgeleri elde edilmiştir. Şekil 4.11’de bu yöntem ile elde edilen sağ ve sol göz imgeleri gösterilmiştir. Çalışmamızda sağ ve sol göz imgesi olarak göz çifti imgesinden

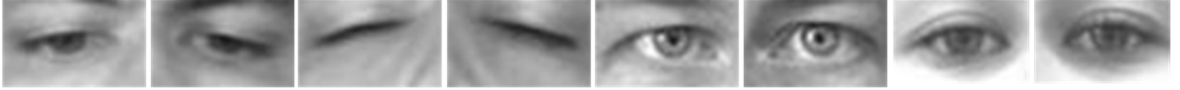
kırpılan bu imgeler kullanılmıştır. Şekil 4.12’de göz çifti elde edilemeyen yüz imgelerine örnekler gösterilmiştir.



Şekil 4.9. Viola-Jones nesne tespit sistemi ile elde edilen göz imgeleri.



Şekil 4.10. Viola-Jones nesne tespit sistemi ile elde edilen göz çifti imgeleri



Şekil 4.11. Göz çifti imgelerinden kırılarak çıkarılan göz imgeleri



Şekil 4.12. Göz çifti tespit edilemeyen yüz imgeleri

### 4.3. Göz Durumu Tespiti

Çalışmamızda göz durumu tespiti için iris ve göz bebeğinin göz içerisinde oluşturduğu dairesel yapılardan faydalanılmıştır.

Bir gözün görme işlevini yerine getirebilmesi için öncelikle siyah bir nokta olan göz bebeğinin ışık alması gerekmektedir. Göz bebeğinden içeri giren ışınlar gözün en arkasında yer alan ve sinir liflerinden oluşan retina tabakası üzerine odaklanır. Retinada

oluşan cismin görüntüsü optik sinir vasıtasıyla beyindeki görme merkezine taşınır ve görme işlemi gerçekleşir.

İris ve göz bebeği aynı merkeze sahip iç içe iki dairedir. Bu ilişkinin gereği olarak irisin yarısının açık olması göz bebeğinin de yarısının açık olduğu anlamına gelmektedir. İrisin ve göz bebeğinin en az yarısının gözükmemesi halinde görme işlevi kısmen de olsa gerçekleşebilir. Bu bilgiler ışığında çalışmamızda açık göz kriteri olarak iris dairesinin varlığı kullanılmıştır. Göz bebeği ve irisin gözükmeyeceği durumlarda ise gözün kapalı olduğu varsayılmıştır.

Çalışmamız şu şekilde devam etmektedir. Öncelikle alınan göz imgesinin kontrastı yükseltilmiştir. Böylece göz akına nazaran daha düşük yoğunluk değerine sahip olan iris ve göz bebeği, göz akından ayrıştırılmaktadır. Ardından bu imgeye gri seviye eşikleme uygulanmaktadır. Sonrasında elde edilen siyah-beyaz imgede küçük pikseller yok edilmektedir. Bu yeni imgenin içerisindeki kapalı alanların içleri doldurulmaktadır. Bu aşamadan sonra canny kenar dedektörü yardımıyla kenarlar çıkarılmaktadır. Son olarak dairesel Hough dönüşümü ile imge içerisinde dairesel şekiller aranmaktadır. Dairesel şekillerin var olması gözün açık olduğu anlamına gelmektedir. Dairesel şekillerin bulunmadığı zamanlar ise göz kapalı olarak sınıflandırılmaktadır.

#### **4.3.1. Kontrast Yükseltme**

Zıtlık anlamına gelen kontrast, bir görüntüdeki piksellerin yoğunluk değerlerinin dağılımı ile ilgili bir parametredir. Kontrastın az olması, görüntü içerisindeki piksellerin arasındaki yoğunluk farklarının az olduğu anlamına gelmektedir. Kontrastın az olduğu görüntülerde, piksel yoğunluk değişimlerinin az olması sebebiyle geçişleri ve kenarları tespit etmek zorlaşacaktır.

Kontrastı yükseltmek için histogramlar kullanılmaktadır. Histogramlar bir görüntüdeki ışık yoğunluğuna ait değerlerin grafiksel gösterimine denir. Bu grafik üzerinde, ışık yoğunluk değerleri yatay eksen ve bu yoğunluk değerlerinin görüntü içinde hangi sıklıkta olduğunu belirten değerlerde dikey eksen yer alır. Histogramlar, görüntüdeki piksellerin koordinat bilgisini içermez. Ancak görüntüde yer alan yoğunluk bilgisinin dağılımı, görüntünün karanlık mı aydınlık mı olduğu hakkında bilgi verir. Ayrıca bir görüntünün belli dağılıma sahip ışık yoğunluk değerleri üzerinde işlemler yapılarak, görüntüde istenen bazı özellikler daha belirgin hale getirilebilir. Histogram üzerinde

yoğunluk değerlerinin belirli bir yerde kümelenmiş olması görüntünün kontrastının düşük olduğu anlamına gelmektedir. Yoğunluk değerlerini bütün histograma yayarak yüksek kontrastlı bir resim elde edilebilmektedir. Bu durumda yoğunluk geçişlerini ve dolayısıyla kenarları tespit etmek kolaylaşacaktır.

Çalışmamızda kontrast yükseltme işlemi için histogram eşitleme kullanılmıştır. Elimizde gri seviye bir  $\{x\}$  görüntüsü olsun.  $n_i$ ,  $i$  gri seviyesindeki piksellerin bulunma sayısı olsun.  $i$  gri seviyesindeki bir pikselin görüntü içerisinde bulunma olasılığı denklem 4.3'de verilmiştir.

$$p_x(i) = p(x = i) = \frac{n_i}{n}, \quad 0 \leq i \leq L \quad (4.3)$$

Burada  $L$  görüntü içerisindeki gri seviye sayısını,  $n$  görüntüdeki toplam piksel sayısını ve  $p_x(i)$  ise  $i$  piksel değerinin  $[0,1]$  olasılıksal aralığında resimde bulunma olasılığını göstermektedir. Kümülatif dağılım fonksiyonu (KDF) ve ayrıca görüntünün toplanmış ve normalleşmiş histogramı olan  $p_x$ , denklem 4.4'de gösterildiği şekilde hesaplanır.

$$kdf_y(i) = \sum_{j=0}^i p_x(j) \quad (4.4)$$

Yeni bir  $\{y\}$  görüntüsü oluşturmak için bu görüntünün KDF'sinin değer aralığında lineer olacağı  $y = T(x)$  şeklinde bir dönüşüm oluşturmalıyız.  $K$  bir sabit olmak üzere bu dönüşüm denklem 4.5 ve 4.6'da gösterildiği şekilde oluşturulmaktadır.

$$kdf_y(i) = iK \quad (4.5)$$

$$y = T(x) = kdf_x(x) \quad (4.6)$$

Burada  $T$  değeri seviyeleri  $[0,1]$  aralığındadır. Bu değerleri orijinal aralıklarına döndürmek için denklem 4.7'de gösterilen basit bir dönüşümün sonuca uygulanması gerekmektedir.

$$y' = y \cdot (\max\{x\} - \min\{x\}) + \min\{x\} \quad (4.7)$$

Burada  $y$  gri seviye değeri,  $\max\{x\}$  resimdeki en yüksek yoğunluk değerini,  $\min\{x\}$  ise resimdeki en düşük yoğunluk değerini göstermektedir.

#### 4.3.2. Eşikleme

Eşikleme, görüntüleri segmentlere ayırmak için kullanılan en basit yöntemlerden birisidir. Eşikleme yardımıyla gri seviye bir görüntüden siyah ve beyaz renklerden oluşan ikili görüntü oluşturulabilir.

Eşikleme işleminde verilen eşik değerinden daha büyük değere sahip pikseller “nesne”, eşik değerine eşit veya daha küçük değere sahip pikseller ise “arkaplan” olarak sınıflandırılır. Nesne olarak sınıflandırılan piksellere “1”, arkaplan olarak sınıflandırılan piksellere “0” değeri verilerek siyah-beyaz olan ikili görüntü elde edilir.

Eşikleme işleminin en önemli parametresi eşik değerinin seçilmesidir. Eşik değerinin seçilmesi için iki yöntem mevcuttur. Bunlar, kullanıcının eşik değerini elle girdiği manuel eşikleme ve bir eşikleme algoritmasının eşik değerini otomatik olarak seçtiği otomatik eşiklemedir.

Sezgin ve Sankur [40], eşikleme metotlarını, algoritmaların kullandığı bilgilere bağlı olarak altıya sınıfa ayırmışlardır:

- Yumuşatılmış histogramın tepelerinin, vadilerinin ve eğrilerinin analiz edildiği histogram şekline dayalı yöntemler.
- Gri seviye piksellerin, nesne ve arkaplan olmak üzere kümelendirildiği kümeleme tabanlı yöntemler.
- Gri seviye ve siyah beyaz resimlerin arkaplan ve önplan bölgelerinin entropilerini kullanan entropi tabanlı yöntemler.
- Gri seviye ve siyah beyaz resimlerde bulanık şekil benzerliği, kenar tespiti gibi algoritmaları kullanan nesne özelliği tabanlı yöntemler.
- Pikseller arasındaki yüksek dereceli olasılıksal dağılımı ve/veya ilişkiyi kullanan uzaysal yöntemler.
- Görüntünün yerel karakterine göre eşik değeri belirleyen yerel yöntemler.



### 4.3.3. Doldurma ve Küçük Şekilleri Eleme

Elde edilen siyah beyaz resimler üzerinde siyah alanlar artalanı beyaz alanlar ise ilgilendiğimiz şekil olan irisi ve göz bebeğini içermektedir. Bazı görüntülerde beyaz alanların iç kısımlarında da siyah alanlar bulunabilmektedir. Buna, göz bebeğinden yansıyan ışıklar sebep olmaktadır. Şekillerin içlerindeki siyah alanların giderilmesi için ‘imfill’ komutu ‘holes’ parametresiyle kullanılmıştır.

İlgilendiğimiz büyük beyaz alanın dışında küçük beyaz alanlarda bulunmaktadır. Bu alanların oluşmasına kirpikler, göz etrafındaki ışık almayan çukurlar gibi düşük yoğunluğa sahip pikseller sebep olmaktadır. Bu küçük beyaz alanlar ileriki aşamalarda kullanılacak olan dairesel Hough dönüşümünde yanlış sonuçlara neden olmaktadır. Bu alanları yok etmek için görüntü üzerinde “bwareaopen” komutu kullanılmıştır.

### 4.3.4. Kenar Çıkarma

Kenar çıkarma, özellik çıkarma işleminde önemli bir alt problemdir. Amacı, görüntü içerisinde komşuları ile farklı yoğunluklara sahip piksellerin tespit edilmesidir. Bu işlem genellikle iki aşamadan oluşur. Birinci aşamada, güncel pikselin yoğunluğu, komşu pikseller ile karşılaştırılarak yoğunluk değişimleri tespit edilir. Geleneksel kenar çıkarıcılar olarak bilinen Sobel operatörü, Prewitt operatörü ve Robert operatörü bu aşamaya ihtiyaç duyarlar. İkinci aşamada ise, elde edilen değişimlerin önceden belirlenmiş bir eşik değeri ile karşılaştırılarak önemli bir değişim olup olmadığı tespit edilir ve bu tespite göre güncel pikselin kenar olup olmadığı hakkında karar verilir.

Kenar çıkarma algoritmalarında iki önemli özelliğin olması gerekmektedir. Bunlardan birincisi, kenar çıkarma işleminin sonucunun, kenar yönlerinden bağımsız olmasıdır. Yani çıkarılan kenar bilgisi, kenarı çıkarılan nesne yön değiştirdiğinde de değişime uğramamalıdır. İkincisi ise, çıkarılan kenar çizgisi bir piksel (nokta) kalınlığında olmasıdır. Böylece elde edilen kenar bilgisi hafızada daha az yer kaplayacaktır.

Geleneksel kenar çıkarma algoritmaları, çoğu zaman bu kriterlere uygun sonuçlar çıkartamazlar. Bu algoritmalarda sık sık yanlış kenar yeri tespit etme, köşeleri tam çıkarmama ve bulanık kenar çıkarma gibi problemler ile karşılaşılabilir. Geleneksel kenar

çıkarma algoritmaları içerisinde en iyilerden biri olarak bilinen Canny kenar çıkarma algoritması üç temel prensibe dayanır:

- İyi kenar çıkarma: Yüksek oranda kenar olma olasılığına sahip noktaları “kenar”, düşük oranda kenar olma olasılığına sahip noktaları ise “kenar değil” olarak sınıflandırmak.
- İyi kenar yeri belirleme: Geçerli kenar yerine yakın olan noktaları kenar noktası olarak işaretlemek.
- Tek nokta kalınlığında kenar çıkarma: Tek boyutlu bir işaretle kenar çizgisini tek nokta kalınlığında göstermek.

Canny kenar çıkartma algoritması bu kriterlere doyurucu bir yanıt verebilmek için kenar çıkartma işlemine başlamadan önce görüntüyü Gaussian fonksiyonunun ilk türevine karşılık gelen bir filtreden geçirir. Görüntü içerisindeki gürültü elendikten sonra filtelenmiş görüntüye Canny operatörü uygulanır. Genellikle Canny operatörü uygulandıktan sonra elde edilen kenar görüntüsü istenmeyen kenar bilgileri içermektedir. Bu bilgiler nesnelerin yanlış algılanmasına yol açar. İstenmeyen bu bilgileri elemek için non-maxima baskısı uygulanır.

#### 4.3.5. Dairesel Şekil Tespiti

Hough dönüşümü  $x,y$  düzlemindeki bir noktanın parametre uzayına dönüşümü olarak tanımlanabilir. Parametre uzayı ilgilenilen nesnenin şekline göre tanımlanır.

$x,y$  düzlemi üzerinde  $x_1,y_1$  noktasından geçen bir doğru, denklem 4.8’de gösterildiği şekilde ifade edilir.

$$y_1 = ax_1 + b \quad (4.8)$$

Kartezyen koordinat sisteminde bir doğrunun denklemini ifade eden bu gösterimde  $a$  ile  $b$  doğrunun parametrelerini temsil eder. Yalnız doğrular için kullanılan Hough dönüşümünde bu denklem kullanılmaz. Çünkü  $x$  eksenine dik olan çizgiler sonsuz  $a$  değerine sahip olmaktadır. Bu da  $a,b$  parametre uzayının sonsuz boyutta olmasını gerektirir. Bunun yerine bir doğruyu temsil etmek için  $\theta$  açısı ve  $\rho$  uzunluğuyla gösterilen doğrunun normali kullanılır. Denklem 4.9’da parametre uzayındaki doğru denklemi verilmiştir.

$$\rho = x\cos(\theta) + y\sin(\theta) \quad (4.9)$$

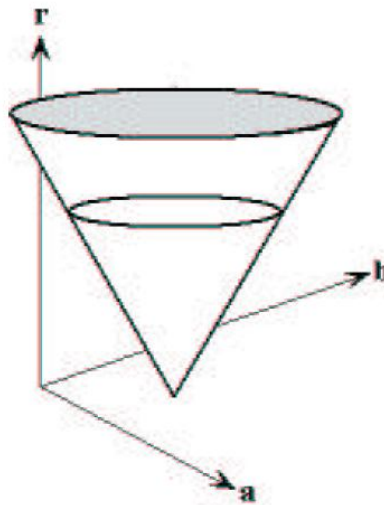
Daireler, doğrulara oranla parametre uzayında daha basit bir şekilde ifade edilebilir. Çünkü dairenin parametreleri doğrudan parametre uzayına transfer edilebilmektedir. Bir daire denklem 4.10'da gösterildiği şekilde ifade edilir.

$$r^2 = (x - a)^2 + (y - b)^2 \quad (4.10)$$

Denklemden de görüldüğü üzere dairenin  $a, b$  ve  $r$  olmak üzere üç parametresi vardır. Burada  $a$  ve  $b$ ,  $x, y$  düzleminde dairenin yönünü  $r$  ise çapını belirtmektedir. Dairenin parametrik gösterimi denklem 4.11'deki gibidir.

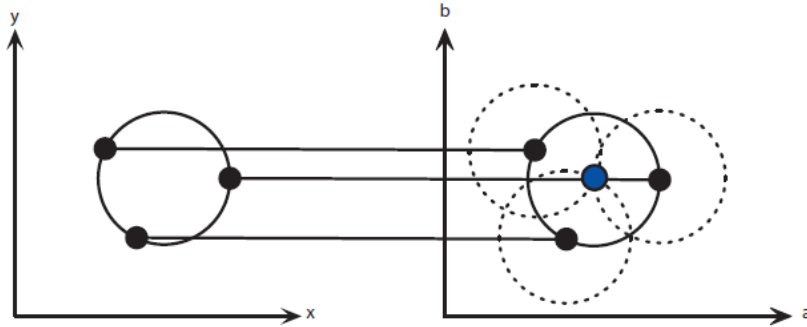
$$\begin{aligned} x &= a + r\cos(\theta) \\ y &= b + r\sin(\theta) \end{aligned} \quad (4.11)$$

Bu nedenle dairenin parametrik uzayı  $\mathbb{R}^3$ 'e ait iken doğrunun parametrik uzayı  $\mathbb{R}^2$ 'ye aittir. Şekli ifade eden parametreler arttıkça parametre uzayı  $\mathbb{R}$  de artacaktır ve böylece Hough dönüşümü daha karmaşık bir hale gelecektir. Şekil 4.13'de dairesel Hough dönüşümü için kullanılacak parametre uzayı gösterilmiştir.  $a, b$  ve  $r$  bu parametre uzayının parametreleridir. Çizilebilecek daireler  $ab$  düzlemi üzerinde herhangi bir noktada merkeze sahiptirler. Dairenin yarıçapı olan  $r$  bilindiğinde ise bu daire parametre uzayı üzerinde çizilebilmektedir.



Şekil 4.13. Hough dönüşümü için kullanılacak parametre uzayı

Hough dönüşümü yardımıyla görüntüler üzerine daireleri bulmak için öncelikle şekil üzerinde kenarlar çıkarılır. Her bir kenar noktasında merkezi bu kenar noktası olmak üzere belirlediğimiz yarı çap büyüklüğünde bir daire çizilir. Bu çizim işlemi x eksenini a, y eksenini b ve yarı çap z olmak üzere parametre uzayında yapılır. Bu işlemden sonra parametre uzayı ile aynı büyüklükte bir matris olan akümülatör, çizilen dairenin koordinatları kullanılarak artırılır. Bütün kenar noktalarına bu işlemler uygulandıktan sonra dairelerin tespiti için akümülatör kullanılır. Akümülatörün içerisindeki sayılar o koordinattan geçen dairelerin sayısını göstermektedir. Buradaki en yüksek sayıyı içeren koordinat dairenin merkezi olarak seçilmektedir. Şekil 4.14’de üç nokta ve sabit bir yarıçap yardımıyla yapılan Hough dönüşümü gösterilmektedir. İlk şekilde bir dairenin kenarlarında yer alıp almadığı tespit edilecek olan üç nokta vardır. Gösterim açısından bir dairenin üzerinde olduğu resmedilmiştir. Bir sonraki örnekte merkezi bu noktalar olmak üzere aynı yarıçapta daireler çizilmiştir. Dairelerin en çok kesiştiği yer olan mavi nokta üzerinde toplam 3 daire kesişmiştir ve Hough dönüşümünün sonucu olarak bu nokta, üç noktanın da üzerinde bulunduğu dairenin merkezi olarak belirlenmiştir.



Şekil 4.14. 3 nokta ve sabit bir yarıçap yardımıyla yapılan Hough dönüşümü

## 5. UYGULAMA SONUÇLARI

Çalışmamızda göz durumu tespiti için şu sıra takip edilmiştir. Öncelikle verilen görüntüye Viola-Jones yüz dedektörü uygulanarak yüz imgesi tespit edilmiştir. Ardından bu yüz imgesine Viola-Jones göz çifti dedektörü uygulanarak göz çiftleri elde edilmiştir. Göz çifti imgesi üzerinden sağ göz ve sol göz kırpılmıştır. Elde edilen sağ ve sol göz bir takım önışlemeden geçirilmiştir. Son olarak dairesel Hough dönüşümü yardımıyla yardımıyla göz durumu tespiti yapılmıştır.

### 5.1. Yüz Konumu Tespiti

Çalışmamızda BioID [34] veritabanı kullanılmıştır. BioID veritabanı 23 farklı kişiden alından 1521 gri seviye resimden oluşmaktadır. BioID veritabanı farklı ışıklandırmalara, artalanlara ve yüz boyutlarına yer vererek gerçek dünya koşullarını sağlamaktadır.

Viola-Jones nesne tespit sistemi ile yapılan yüz konumu tespit işleminde kaskad nesne dedektörü [7] kullanılmıştır. Dedektörün sınıflandırma modeli olarak 'FrontalFaceCART' seçilmiştir. Göz çukuru yüksekliğinin yüz yüksekliğine oranının %16 ile %18 arasında değiştiği istatistiksel yöntemler ile tespit edilmiştir. Göz tespitini yapabilmek için ihtiyacımız olan en düşük çözünürlük ise 40\*20 pikseldir. Çünkü Dairesel Hough dönüşümü algoritması yarıçapı 6 dan küçük daireleri tespit ederken hatalar verebilmektedir. Bu bilgilerin ışığında bulunacak en küçük yüzün çözünürlüğü 100\*100 piksel olarak ayarlanmıştır.

Viola-Jones yüz dedektörü, gözlük kullanılan ve göz tespiti yapılamayacak kadar karanlık olan resimler elenerek seçilen 1010 gri seviye resim üzerinde çalıştırılmıştır. 964 doğru yüz konumu tespiti eden dedektör 46 resimde yüz tespit edememiştir. Ayrıca dedektör hiçbir resimde yanlış yüz tespit etmemiştir. Tablo 5.1'de yüz konumu tespiti başarı oranları verilmiştir. Şekil 5.1'da yüz tespit edilen resimlere, şekil 5.2'de yüz tespit edilemeyen resimlere örnekler gösterilmiştir.

Tablo 5.1. Yüz konumu tespiti başarı oranları

Toplam	Doğru Tespit	Yanlış Tespit	Tespit Edilemeyen
1010	964	-	46
%100	%95,5	-	%4,5



Şekil 5.1. Yüz tespit edilen resimlere ve yüz imgelerine örnekler



Şekil 5.2. Yüz tespit edilemeyen resimlere örnekler

## 5.2. Göz Konumu Tespiti

Göz konumu tespit işleminde Viola-Jones nesne tespit sistemi [7] kullanılmıştır. Dedektörün sınıflandırma modeli olarak sağ göz 'RightEyeCART', sol göz 'LeftEyeCART' ve göz çifti 'EyePairBig' olarak seçilmiştir. Bulunacak en küçük göz boyutu 25\*15 piksel, göz çifti boyutu ise 70\*15 piksel olarak belirlenmiştir.

Şekil 5.3’de elde edilen göz imgeleri gösterilmektedir. Bu imgelerde gözlere ek olarak kaşların da imgeye dahil olduğu görülmektedir. Kaşlar ile gözlerin aynı imge içerisinde bulunması yöntemimizin performansını düşürebilmektedir.

Şekil 5.4’de göz çifti imgeleri gösterilmiştir. Göz çifti imgeleri göz imgelerinden daha az kaş içermektedir. Bu yüzden göz imgesi olarak göz çifti imgesinden kırılarak çıkarılacak göz imgelerinin kullanılmasının daha uygun olduğu tespit edilmiştir. Kırpma işleminde kullanılacak koordinatları elde etmek için istatistiki bir yaklaşım kullanılmıştır. İstatistiksel olarak bir göz çifti imgesinde gözlerin kapladıkları uzunluk sağ ve sol taraftan ortalama %30 ile %37 aralığındadır. %5 hata payı da ekleyerek olabilecek maksimum uzunluk %42 olmaktadır. %42 den sonrası göz ile alakasız olan burunu temsil etmektedir. Bu değerler referans alınarak göz çifti imgesi kırıldıktan sonra sağ ve sol göz imgeleri elde edilmiştir. Şekil 5.5’de bu yöntem ile elde edilen sağ ve sol göz imgeleri gösterilmiştir. Çalışmamızda sağ ve sol göz imgesi olarak göz çifti imgesinden kırılan bu imgeler kullanılmıştır. Şekil 5.6’da göz çifti elde edilemeyen yüz imgelerine örnekler gösterilmiştir.

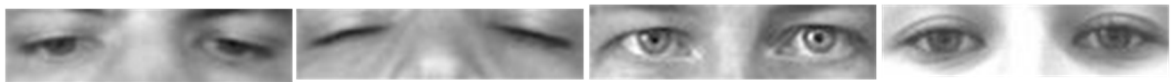
Viola-Jones göz çifti dedektörü bir önceki aşamada doğru olarak tespit edilen 964 adet yüz resmine uygulanmıştır. 946 adet göz çifti doğru tespit edilmiş olup hiçbir resimde yanlış göz çifti tespit edilmemiştir. 18 adet resimde ise göz çifti tespit edilememiştir. Tablo 5.2’de göz çifti konumu tespiti başarı oranları verilmiştir.

Tablo 5.2. Göz çifti konumu tespiti başarı oranları

Toplam	Doğru Tespit	Yanlış Tespit	Tespit Edilemeyen
964	946	-	18
%100	%98	-	%2



Şekil 5.3. Viola-Jones nesne tespit sistemi ile elde edilen göz imgeleri.



Şekil 5.4. Viola-Jones nesne tespit sistemi ile elde edilen göz çifti imgeleri



Şekil 5.5. Göz çifti imgelerinden kırılarak çıkarılan göz imgeleri



Şekil 5.6. Göz çifti tespit edilemeyen yüz imgeleri

### 5.3. Göz Durumu Tespiti

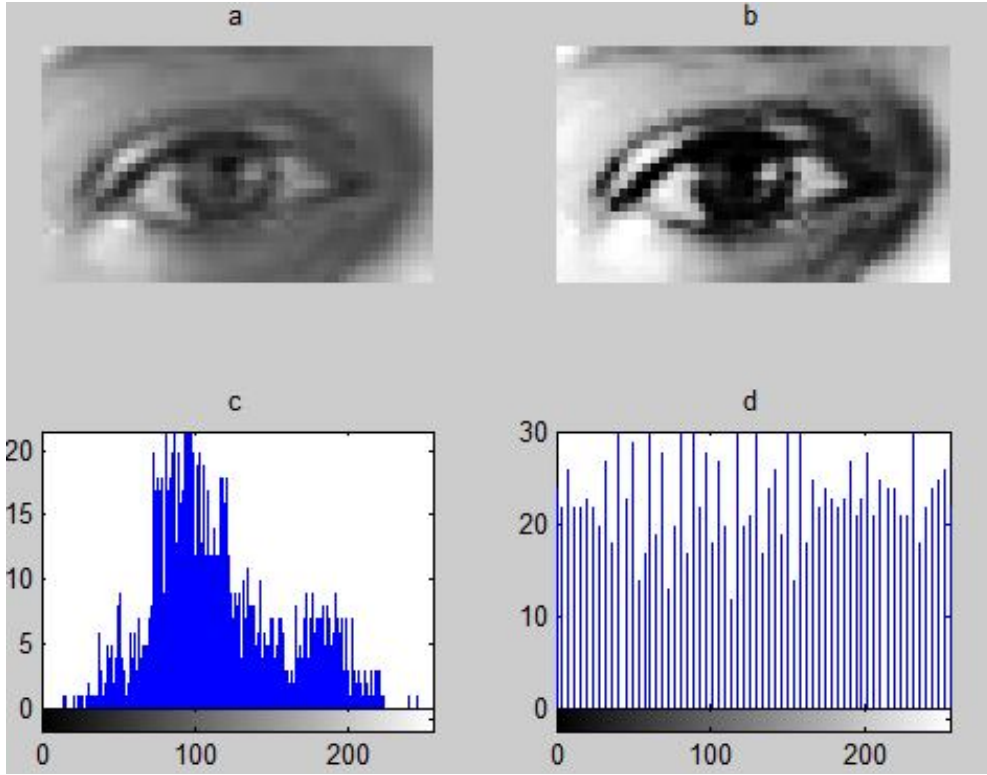
Göz durumu tespiti için öncelikle alınan göz imgesinin kontrastı yükseltilmiştir. Ardından bu imgeye gri seviye eşikleme uygulanmaktadır. Sonrasında elde edilen siyah-beyaz imgede küçük pikseller yok edilmektedir. Bu yeni imgenin içerisindeki kapalı alanların içleri doldurulmaktadır. Bu aşamadan sonra canny kenar dedektörü yardımıyla kenarlar çıkarılmaktadır. Son olarak dairesel Hough dönüşümü ile imge içerisinde dairesel şekiller aranmaktadır. Aşağıda her bir aşamanın uygulaması ve elde edilen sonuçlar verilmiştir.

#### 5.3.1. Kontrast Yükseltme

Kontrast yükseltme işlemi, düşük yoğunluktaki iris ve göz bebeğinin, göz akı ve gözün geri kalan bölümlerinden ayrıştırılması için kullanılmıştır. Kontrast yükseltme işlemi için histogram eşitleme kullanılmıştır.

Şekil 5.7’de örnek göz imgesi üzerindeki işlemler gösterilmiştir. Görüldüğü üzere kümelenmiş olan histogram bandı genele yayılmıştır.



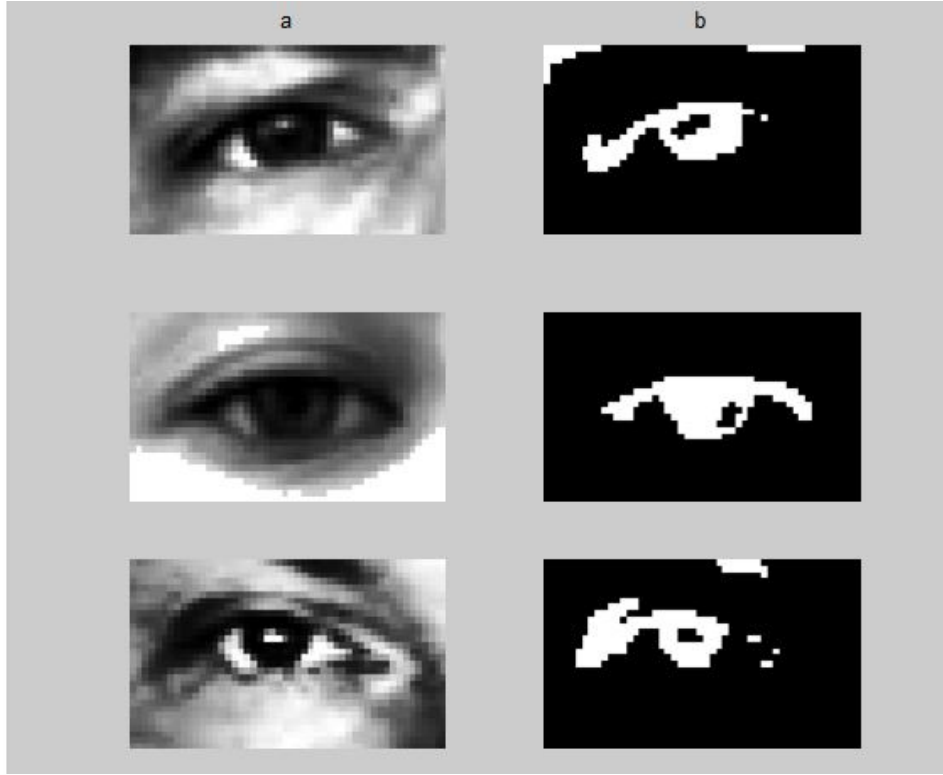


Şekil 5.7. Göz imgesi histogram eşitleme a. Göz İmgesi b. Histogramı eşitlenmiş göz imgesi c. Göz imgesi histogramı d. Histogramı eşitlenmiş göz imgesi histogramı

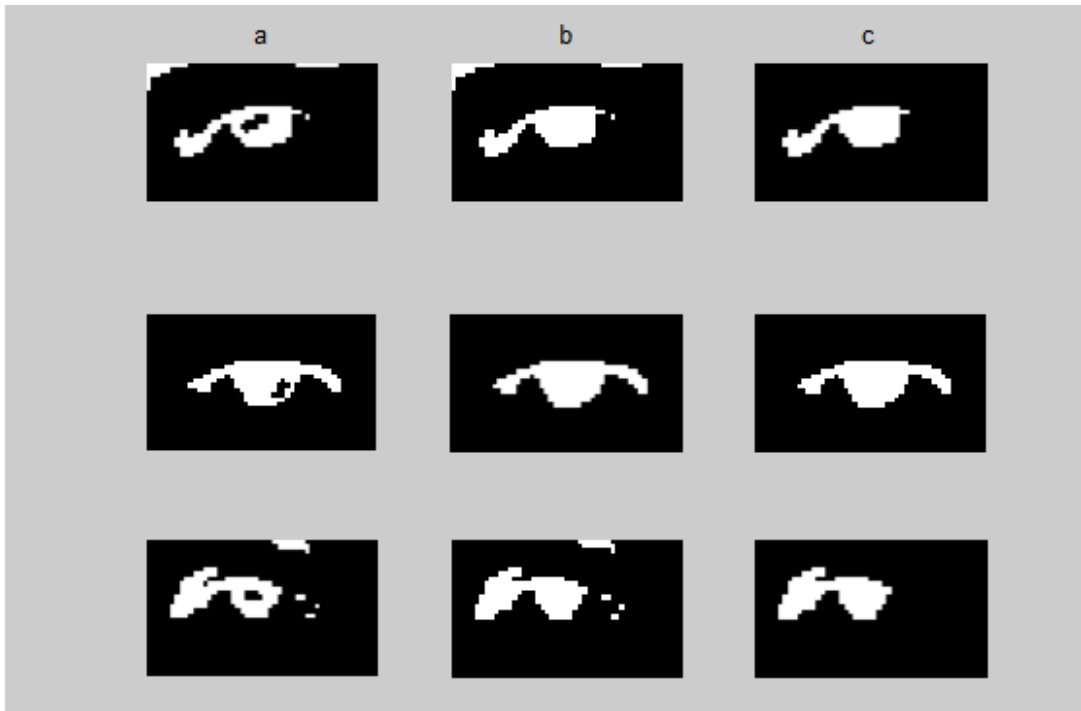
### 5.3.2. Doldurma ve Küçük Şekilleri Eleme

Elde edilen siyah beyaz (SB) resimler üzerinde siyah alanlar artalanı beyaz alanlar ise ilgilendiğimiz şekil olan irisi ve göz bebeğini içermektedir. Bazı görüntülerde beyaz alanların iç kısımlarında da siyah alanlar bulunabilmektedir. Buna, göz bebeğinden yansıyan ışıklar sebep olmaktadır. Şekillerin içlerindeki siyah alanlar tespit işleminin daha sağlıklı gerçekleşmesi için doldurulmuştur.

İlgilendiğimiz büyük beyaz alanın dışında küçük beyaz alanlarda bulunmaktadır. Bu alanların oluşmasına kirpikler, göz etrafındaki ışık almayan çukurlar gibi düşük yoğunluğa sahip pikseller sebep olmaktadır. Bu küçük beyaz alanlar ileriki aşamalarda kullanılacak olan dairesel Hough dönüşümünde yanlış sonuçlara neden olmaktadır. Yazılım ortamında bu küçük alanlar yok edilmiştir. Böylece bir sonraki aşamada işlem yükümüz hafıleyecek ve doğru sonuç verme olasılığımız artacaktır. Şekil 5.9'da ikili imgelerinin doldurma ve filtrelemeden sonraki durumları gösterilmektedir.



Şekil 5.8. Eşikleme a. Histogramı eşitlenmiş göz imgeleri b. Eşikleme sonucu



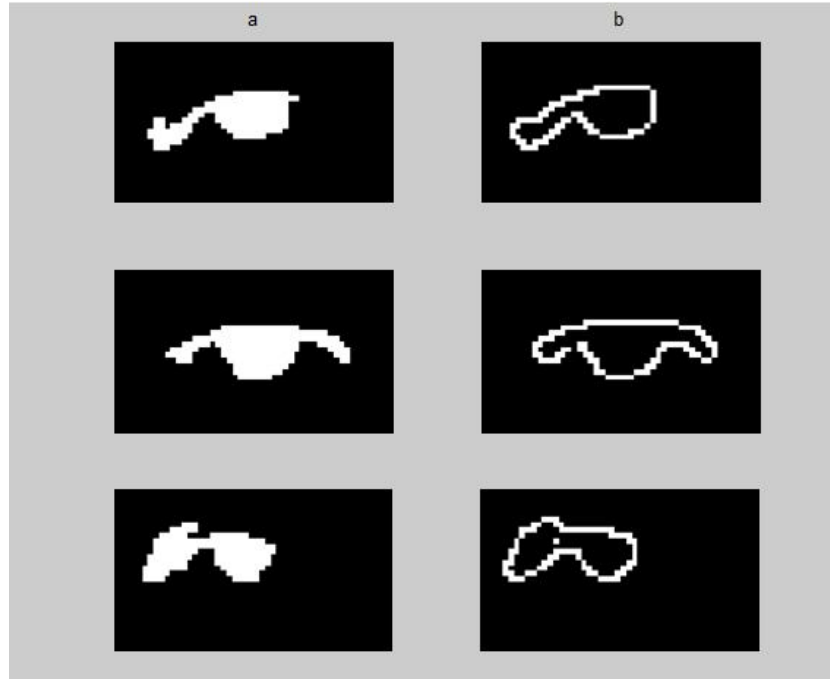
Şekil 5.9. İkili imgelerin doldurma ve filtrelemeden sonraki durumları a. Siyah Beyaz görüntü b. Doldurulmuş Görüntü c. Seçilmiş Görüntü

### 5.3.4. Kenar Çıkarma

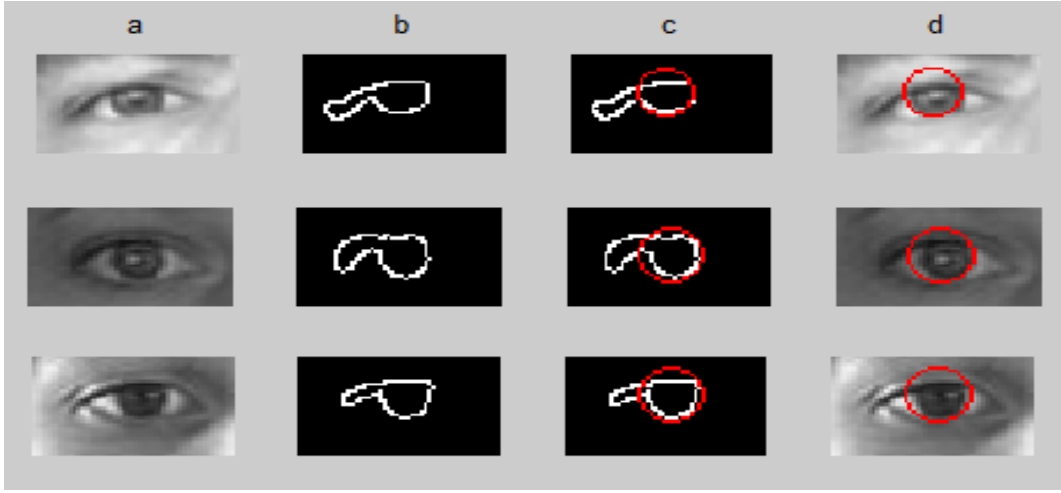
Çalışmamızda kenar çıkarma işlemi için ‘canny’ kenar dedektörü kullanılmıştır. Şekil 5.10’da bir önceki aşamadan gelen göz imgeleri ve kenar dedektörü yardımıyla çıkarılan kenarlar gösterilmektedir.

### 5.3.3. Dairesel Şekil Tespiti

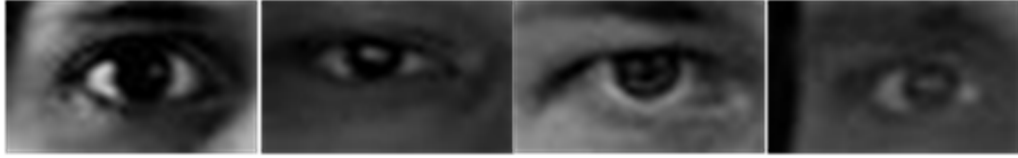
Çalışmamızda dairesel olan göz bebeği tespiti için Hough dönüşümünü kullanılmıştır. Hough dönüşümü için 6 ile 14 arasında yarıçap değeri alan daireler ve .85 hassaslık parametresiyle kullanılmıştır. Hough dönüşümü sonucu elde edilen daireler kırmızı ile çizilmiştir. Şekil 5.11’de kenarları çıkarılmış ikili göz imgeleri ve bu imgelere uygulanan dairesel Hough dönüşümü sonucu açık olarak tespit edilen açık göz örnekleri gösterilmektedir. Şekil 5.12’de kapalı olarak tespit edilen açık göz örnekleri gösterilmektedir. Şekil 5.13’de kapalı olarak tespit edilen kapalı göz örnekleri gösterilmektedir. Şekil 5.14’de açık olarak tespit edilen kapalı göz örnekleri gösterilmektedir.



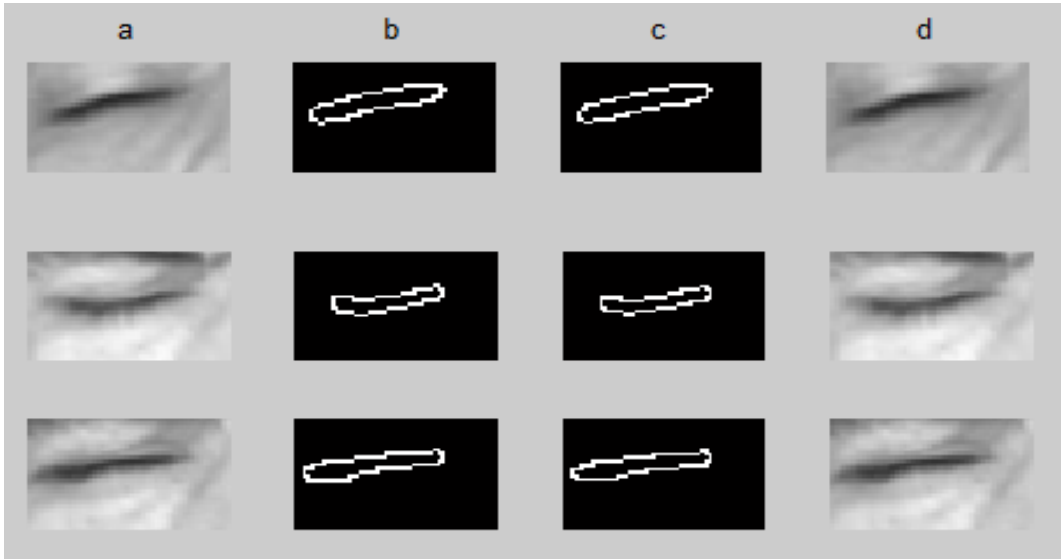
Şekil 5.10. Kenar Çıkarma a. Siyah Beyaz İmge b. Çıkarılmış Kenarlar



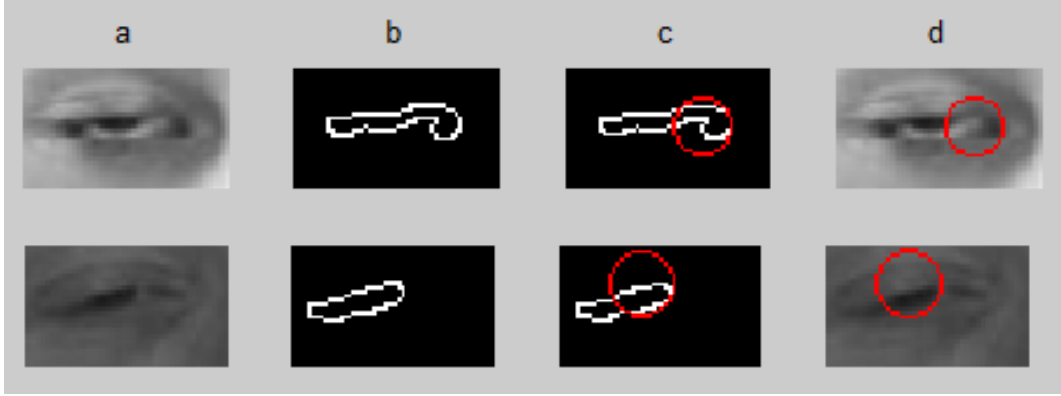
Şekil 5.11. Açık olarak tespit edilen açık göz örnekleri. a. Göz imgesi b.Kenar Haritası c. Hough dönüşümü d. Göz bebeği



Şekil 5.12. Kapalı olarak tespit edilen açık göz örnekleri



Şekil 5.13. Kapalı olarak tespit edilen kapalı göz örnekleri. a. Göz imgesi b.Kenar Haritası c. Hough dönüşümü d. Göz bebeği



Şekil 5.14. Açık olarak tespit edilen kapalı göz örnekleri. a. Göz imgesi b.Kenar Haritası c. Hough dönüşümü d. Göz bebeği

#### 5.4. Değerlendirme

Uygulamalar, Intel'in Core2Duo T5600 çift çekirdek ve 1.83 GHz saat hızına sahip işlemcisi, 533 MHz 4 GB bellek ile Matlab yazılım ortamında gerçekleştirilmiştir. Yazılımımız yüz konumu tespiti için 0.14 sn, göz çifti tespiti için 0.06 sn, göz çıkarımı için 0.0002 sn, önışleme işlemleri için 0.014 sn ve dairesel Hough dönüşümü için 0.016 sn zaman harcamaktadır. Uygulamamız 23 milisaniyede 1 kare işlemektedir. Bu da ortalama 4 fps (kare/saniye) hızına denk gelmektedir.

Dairesel Hough dönüşümü algoritması yarıçapı 6 dan küçük daireleri tespit ederken hatalar verebilmektedir. Bu sebepten dolayı 40\*25 piksel boyutundan daha düşük boyuta sahip resimlerde göz içerisindeki dairesel alanlar başarılı bir şekilde tespit edilememektedir. Bunu dikkate alarak göz çiftlerinden gözler elde edildikten sonra göz durumu tespiti için en az 40\*25 piksel büyüklüğünde gözler kullanılmıştır.

Göz durumu tespiti için toplam 946 adet doğru tespit edilen resim içerisinde yukarıdaki şartı sağlayan 523 adet göz resmi kullanılmıştır. Bu resimler içerisinde 501 adet açık 22 adet kapalı göz resmi bulunmaktadır. Açık durumda olan göz resimleri, %95.2 oranında açık, %4.8 oranında kapalı olarak tespit edilmiştir. Kapalı durumda olan göz resimleri ise %77.2 oranında kapalı, %22.8 oranında açık olarak sınıflandırılmıştır. Tablo 5.3'te göz durumu tespitine ait başarı oranları verilmiştir.

Tablo 5.3. Göz durumu tespiti başarı oranları

<b>Gerçek\Tespit Edilen</b>	<b>Açık</b>	<b>Kapalı</b>	<b>Toplam</b>
<b>Açık</b>	477(%95.2)	24(%4.8)	501(%100)
<b>Kapalı</b>	5(%22.8)	17(%77.2)	22(%100)

Açık gözlerin kapalı olarak tespit edilmesindeki en önemli etkenin iris rengi olduğu tespit edilmiştir. Eşikleme işlemi esnasında, göz bebeği ve irisin göz akına nazaran daha düşük yoğunluklu olduğu gerçeğine dayanılarak bölütleme yapılmaktadır. Renkli olmayan gözlerde bu varsayım ile bölütleme işe yararken, renkli gözlerin yoğunluklarının fazla olmasından dolayı bölütleme işlemi başarıyla gerçekleşmemektedir. Renkli gözlerde dairesel Hough dönüşümü ile göz durumu tespit işlemini başarılı bir şekilde uygulamak için renkli göz imgelerine ihtiyaç vardır. Renkli göz imgelerinde, HSI ve HSV renk uzaylarındaki renk doygunluğu parametresinin yardımıyla göz kenarları iris ve göz bebeği kenarları çıkarılabilir ve bu kenarlara dairesel Hough dönüşümü uygulanarak göz durumu tespiti yapılabilir. Açık gözlerin kapalı olarak tespit edilmesindeki diğer etkenler ise ışıklandırmanın yetersizliği, kısmen kapalı gözler ve gözün sağ veya sol kenarı ile birleşmiş olan iris olarak örneklendirilebilir.

Kapalı gözlerin açık olarak tespit edilmesindeki en önemli etkenin kirpikler olduğu tespit edilmiştir. Gözler, kapalı oldukları zaman düz bir çizgi teşkil etmektedirler. Kısa kirpikler bu düz çizginin içerisinde sonlanırken, uzun kirpikler bu çizginin alt kısmından dışarıya uzarlar ve bir eğrilik oluştururlar. Dairesel Hough dönüşümü esnasında bu eğrilik dairenin görünen bir kenarı olarak algılanır ve yanlış sonuca sebep olur.

## 6. SONUÇLAR

Bilgisayarlı görme tekniklerinin gelişmesiyle yüz tespiti ve göz durumu tespiti bir çok uygulama açısından önem kazanmıştır. Bu çalışmamızda da göz durumu tespiti için bir uygulama gerçekleştirilmiştir.

Öncelikle verilen görüntülerde yüz imgesi tespiti için Viola-Jones yüz dedektörü kullanılmıştır. Viola-Jones yüz dedektörü ile verilen 1010 resimin 964'ünde yüz tespit edilerek %95,5 oranında başarı sağlanmıştır. Ardından bu yüz imgelerine Viola-Jones göz çifti dedektörü uygulanarak göz çiftleri elde edilmiştir. Bir önceki aşamadan elde edilen 964 yüz görüntüsünden, 946 göz çifti tespit edilerek %98 oranında başarı sağlanmıştır. Devamında göz çifti imgesi kırılarak sağ ve sol gözler elde edilmiştir.

Göz durumu tespiti için bir önceki aşamadan alınan göz imgesinin kontrastı yükseltilmiştir. Ardından bu imgeye gri seviye eşikleme uygulanmıştır. Sonrasında elde edilen siyah-beyaz imgede kullanacağımız özellik alanı dışında kalan artık alanlar yok edilmiştir. Bu yeni imgenin içerisindeki kapalı alanların içleri doldurulmuştur. Bu aşamadan sonra canny kenar dedektörü yardımıyla kenarlar çıkarılmıştır. Son olarak dairesel Hough dönüşümü ile imge içerisinde dairesel şekiller aranmıştır. Çözünürlüğü düşük ve göz tespit edilemeyecek kadar karanlık olan göz imgeleri çıkarıldığında elde kalan 523 imgede göz imgesinde, açık gözlerin %95.2'i kapalı gözlerin ise %77.2'si doğru olarak tespit edilmiştir.

Yapılan bu çalışmada gri seviye imgeler kullanılmıştır. Renkli imgeler kullanılarak yapılacak çalışmalar daha iyi sonuçlar üretebilir. Ayrıca kızıl ötesi görüntüleme kullanılarak görünür ışık olmayan ortamlarda yüz ve göz tespiti yapılabilir. Bu yöntemler ile elde edilecek görüntülerde göz durumu tespiti daha başarılı bir şekilde gerçekleştirilebilir.

Göz durumu bilgisi, günümüzde birçok uygulamaya giriş verisi olarak kullanılmaktadır. Daha hızlı ve doğru sonuç veren göz durumu tespit algoritmalarının artmasıyla beraber, göz durumu daha geniş alanlarda kullanıma yer bulacaktır.

## REFERANSLAR

- [1] **Q. Wang and J. Y. Yang**, 2006. "Eye location and eye state detection in facial images with unconstrained background", *Journal of Information and Computing Science*, Vol. 1, No. 5, pp. 284-289.
- [2] **Y. L. Tian, T. Kanade and J. F. Cohn**, 2000. "Eye-state action unit detection by gabor wavelets", *International Conference on Multimodal Interfaces 2000*, LNCS 1948, Vol. 1948, pp. 143-150.
- [3] **R. Senaratne, D. Hardy, B. Vanderaa, S. Halgamuge and D. Liu et al.**, 2007. "Driver Fatigue Detection by Fusing Multiple Cues", *Lecture Notes In Computer Science* Vol. 4492, pp. 801-809.
- [4] **C. Xu, Y. Zheng and Z. F. Wang**, 2008. "Efficient eye states detection in realtime for drowsy driving monitoring system", *IEEE Conf. on ICIA*, vol. 1-4, pp. 170-174.
- [5] **C. Zhang and Z. Zhang**, June 2010. "A Survey of Recent Advances in Face Detection", *Technical Report MSR-TR-2010-66*, Microsoft Research.
- [6] **M.-H. Yang, D. J. Kriegman, and N. Ahuja**, 2002. "Detecting faces in images: A survey", *IEEE Trans. on PAMI*, 24(1):34–58.
- [7] **P. Viola and M. Jones**, 2001. "Rapid object detection using a boosted cascade of simple features", In *Proc. of CVPR*..
- [8] **G. Yang and T. S. Huang**, 1994. "Human Face Detection in Complex Background", *Pattern Recognition*, vol. 27, no. 1, pp. 53-63.
- [9] **S.A. Sirohey**, 1993. "Human Face Segmentation and Identification", *Technical Report CS-TR-3176*, Univ. of Maryland.
- [10] **T.K. Leung, M.C. Burl, and P. Perona**, 1995. "Finding Faces in Cluttered Scenes Using Random Labeled Graph Matching", *Proc. Fifth IEEE International Conference Computer Vision*, pp. 637-644.
- [11] **T. Sakai, M. Nagao, and S. Fujibayashi**, 1969. "Line Extraction and Pattern Detection in a Photograph", *Pattern Recognition*, vol. 1, pp. 233-248.
- [12] **I. Craw, H. Ellis, and J. Lishman**, 1987. "Automatic Extraction of Face Features", *Pattern Recognition Letters*, vol. 5, pp. 183-187.



- [13] **I. Craw, D. Tock, and A. Bennett**, 1992 “Finding Face Features”, Proc. Second European Conf. Computer Vision, pp. 92-96,.
- [14] **A. Samal and P.A. Iyengar**, 1995. “Human Face Detection Using Silhouettes”, International Journal of Pattern Recognition and Artificial Intelligence, vol. 9, no. 6, pp. 845-867.
- [15] **A. Yuille, P. Hallinan, and D. Cohen**, 1992. “Feature Extraction from Faces Using Deformable Templates”, Int’l J. Computer Vision, vol. 8, no. 2, pp. 99-111.
- [16] **A. Lanitis, C.J. Taylor, and T.F. Cootes**, 1995 .“An Automatic Face Identification System Using Flexible Appearance Models”, Image and Vision Computing, vol. 13, no. 5, pp. 393-401.
- [17] **T. Kohonen**, 1989. “Self-Organization and Associative Memory”, Springer.
- [18] **M. Turk and A. Pentland**, 1991. “Eigenfaces for Recognition”, J. Cognitive Neuroscience, vol. 3, no. 1, pp. 71-86.
- [19] **K. Sung and T. Poggio**, 1994. “Example-Based Learning for View- Based Human Face Detection”, Technical Report AI Memo 1521, Massachusetts Inst. Of Technology AI Lab.
- [20] **K. Sung and T. Poggio**, 1998. “Example-Based Learning for View-Based Human Face Detection”, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 20, no. 1, pp. 39-51.
- [21] **H. Rowley, S. Baluja, and T. Kanade**, 1996. “Neural Network-Based Face Detection”, Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 203-208.
- [22] **H. Rowley, S. Baluja, and T. Kanade**, 1998. “Neural Network-Based Face Detection”, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 20, no. 1, pp. 23-38.
- [23] **H. Schneiderman and T. Kanade**, 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.
- [24] **A.L. Yuille, P.W. Hallinan, D.S. Cohen**, 1992. “Feature extraction from faces using deformable templates”, International Journal of Computer Vision vol. 8, no. 2, 104–109.

- [25] **K.M. Lam, H. Yan**, 1996 . “An improved method for locating and extracting the eye in human face images”, in: Proceedings of the IEEE International Conference on Pattern Recognition (ICPR’96), IEEE: Vienna, Austria, pp. 411–415.
- [26] **Y. Li, X. Qi, Y. Wang**, 2001. “Eye detection using fuzzy template matching and feature-parameter-based judgement”, Pattern Recognition Letters vol. 22, no. 10, 1111–1124.
- [27] **S. Sirohey, A. Rosenfeld**, 2001. “Eye detection in a face image using linear and nonlinear filters”, Pattern Recognition Letters (2001) 1367–1391.
- [28] **T. Kawaguchi, D. Hidaka, M. Rizon**, 2000. “Detection of eyes from human faces by Hough transform and separability filter”, in: Proceedings of International Conference on Image Processing, pp. 49–52.
- [29] **K. Peng, et al.**, 2005. “A robust algorithm for eye detection on gray intensity face without spectacles”, Journal of Computer Science & Technology vol. 5, no. 3, 127–132.
- [30] **D. Balya, T. Roska**, 1999. “Face and eye detection by CNN (Cellular Neuron Network) algorithms”, Journal of VLSI Signal Processing, 497–511.
- [31] **J. Huang, H. Wechsler**, 1999. Eye detection using optimal wavelet packets and radial basis functions (rbfs), International Journal of. Pattern Recognition and Artificial Intelligence, vol. 13, no. 7, 1009–1025.
- [32] **W. Peng, et al.**, 2005. “Automatic eye detection and its validation”, in: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05), pp. 164–171.
- [33] **Z. Zhu, Q. Ji**, 2005. “Robust real-time eye detection and tracking under variable lighting conditions and various face orientations”, Computer Vision and Image Understanding-Special Issue on Eye Detection and Tracking, vol.98, no.01, 124–154.
- [34] <http://www.bioid.com/downloads/software/bioid-face-database.html>, 27.02.2012
- [35] **F. Crow**, 1984. “Summed-area tables for texture mapping”, In Proc. of SIGGRAPH, volume 18, pages 207–212,.
- [36] **Papageorgiou, Oren and Poggio**, 1998. "A general framework for object detection", International Conference on Computer Vision.

- [37] **R. Meir and G. Ratsch**, 2003. “An introduction to boosting and leveraging”, S. Mendelson and A. J. Smola Ed., *Advanced Lectures on Machine Learning*, Springer-Verlag Berlin Heidelberg, pages 118–183.
- [38] **J. Friedman, T. Hastie, and R. Tibshirani**, 1998, “Additive logistic regression: a statistical view of boosting”, Technical report, Dept. of Statistics, Stanford University.
- [39] **Y. Freund and R. E. Schapire**, 1997. “A decision-theoretic generalization of on-line learning and an application to boosting”, *Journal of Computer and System Sciences*, vol. 55, no. 1, p. 119–139.
- [40] **M. Sezgin and B. Sankur**, 2004. “Survey over image thresholding techniques and quantitative performance evaluation”, *Journal of Electronic Imaging* vol. 13, no. 1, 146–165.

## **ÖZGEÇMİŞ**

Ömer Faruk SÖYLEMEZ, 1987 yılında Hasköy/MUŞ'da doğdu. Orta öğrenimini Yamanlar Anadolu Lisesi'nde bitirdi. 2004 – 2009 yılları arasında Eskişehir Osmangazi Üniversitesi Bilgisayar Mühendisliği Bölümünü okudu. 2010 yılında Fırat Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Donanım Anabilim dalında Yüksek lisans öğrenimine başladı. Aynı zamanda Dicle Üniversitesinde Araştırma Görevlisi olarak çalışmaktadır.