T.C.
DÜZCE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

# YÜZ BULMA VE TANIMA SİSTEMLERİ KULLANARAK KİMLİK TESPİTİNİN YAPILMASI

YÜKSEK LİSANS TEZİ

RECEP HOLAT

TEMMUZ 2014

DÜZCE

# KABUL VE ONAY BELGESİ

Recep HOLAT tarafından hazırlanan Yüz Bulma ve Yüz Tanıma Sistemi Kullanarak Kimlik Tespiti Yapma isimli lisansüstü tez çalışması, Düzce Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun ..... tarih ve ..... sayılı kararı ile oluşturulan jüri tarafından Elektrik ve Elektronik Mühendisliği Anabilim Dalı'nda Yüksek Lisans Tezi olarak kabul edilmiştir.

Üye
(Tez Danışmanı)
Yrd. Doç. Dr. Selman KULAÇ
Düzce Üniversitesi

Üye
Doç. Dr. Nedim TUTKUN
Düzce Üniversitesi

Üye
Doç. Dr. Resul KARA
Düzce Üniversitesi

Üye
Doç. Dr. Ali ÖZTÜRK
Düzce Üniversitesi

Üye
Yrd. Doç. Dr. Mehmet UÇAR
Düzce Üniversitesi

Tezin Savunulduğu Tarih : 14.07.2014

## ONAY

Bu tez ile Düzce Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu .....'ın Elektrik ve Elektronik Mühendisliği Anabilim Dalı'nda Yüksek Lisans derecesini almasını onamıştır.

Prof. Dr. Haldun MÜDERRİSOĞLU
Fen Bilimleri Enstitüsü Müdürü

# BEYAN

Bu tez çalışmasının kendi çalışmam olduğunu, tezin planlanmasından yazımına kadar bütün aşamalarda etik dışı davranışımın olmadığını, bu tezdeki bütün bilgileri akademik ve etik kurallar içinde elde ettiğimi, bu tez çalışmasıyla elde edilmeyen bütün bilgi ve yorumlara kaynak gösterdiğimi ve bu kaynakları da kaynaklar listesine aldığımı, yine bu tezin çalışılması ve yazımı sırasında patent ve telif haklarını ihlal edici bir davranışımın olmadığını beyan ederim.

14 Temmuz 2014 (Tarih)

(İmza)

Recep HOLAT

*Sevgili Aileme*

# TEŞEKKÜR

Yüksek lisans öğrenimim ve bu tezin hazırlanmasında süresince gösterdiği her türlü destek ve yardımdan dolayı çok değerli hocam Yrd. Doç. Dr. Selman KULAÇ'a en içten dileklerimle teşekkür ederim.

Bu çalışma boyunca yardımlarını ve desteklerini esirgemeyen sevgili aileme ve çalışma arkadaşlarıma sonsuz teşekkürlerimi sunarım.

**14 Temmuz 2014** **Recep HOLAT**

**İÇİNDEKİLER** **Sayfa**

**TEŞEKKÜR SAYFASI ……..……..……..I**

**İÇİNDEKİLER …….……II**

**ŞEKİL LİSTESİ …….....IV**

**ÇİZELGE LİSTESİ …….....VI**

**SİMGELER VE KISALTMALAR LİSTESİ ……...….VII**

**ÖZET …….....….....1**

**ABSTRACT …….…….....2**

**EXTENDED ABSTRACT ……...…….…….……..…..3**

**1. GİRİŞ ……...6**

**1.1. BİYOMETRİK SİSTEMLER……6**

**1.2. YÜZ TANIMA SİSTEMLERİNE GENEL BAKIŞ…….8**

**2. MATERYAL VE YÖNTEM …….... 10**

**2.1. YÜZ TANIMA SİSTEMİ ……..……..….10**

**2.2. YÜZ TANIMA UYGULAMA ALANLARI …….…..12**

**2.3. YÜZ TANIMADA KULLANILAN TEMEL YAKLAŞIMLAR …….....13**

**2.3.1. Özellik Tabanlı Yöntemler.……13**

**2.3.2. Şablon Tabanlı Yöntemler …….…….…….....13**

**2.3.3. Görünüş Tabanlı Yöntemler ……14**

**2.4. YÜZ TANIMADAKİ ZORLUKLAR ……..14**

**2.4.1. Aydınlatma Varyasyonu ……...15**

**2.4.2. İfade Çeşitlilikleri ……..15**

**2.4.3. Tıkanıklıklar/Kapanmalar ……...16**

**2.4.4. Poz Varyasyonu ……...……...16**

**2.5. YÜZ TANIMA TEKNİKLERİ.....17**

**2.5.1. Özyüz Yaklaşımı ile Yüz Tanıma .....17**

*2.5.1.1. Özyüz Algoritması (Temel Bilesenler Analizi) .....18*

**2.5.2. Fisher Yaklaşımı ile Yüz Tanıma .....24**

**2.5.3. Yerel İkili Örüntü Yöntemi İle Yüz Tanıma .....28**

**2.6. AYDINLATMA ETKİSİNİN AZALTILMASI .....30**
**2.6.1. Aydınlatma Normalleştirme Metodları .....30**
*2.6.1.1. Histogram Eşitleme .....30*
*2.6.1.2. Yerel Histogram Eşitleme .....33*
*2.6.1.3. Kontrast Sınırlı Adaptif Histogram Eşitleme .....33*
**2.6.2. Gürültü Azaltma (Filtreleme) .....34**
*2.6.2.1. Gauss Filtresi .....35*
*2.6.2.2. Medyan Filtresi .....36*
*2.6.2.3. Laplace Filtresi .....38*
*2.6.2.4. Yüksek Artış Filtresi .....39*
**2.7. C# DİLİNDE ÖZYÜZ YÖNTEMİ İLE YÜZ TANIMA SİSTEMİ GELİŞTİRİLMESİ .....41**
**2.7.1. Yüz Tanıma Sistemi Mantıksal Tasarımı .....41**
**2.7.2. Yüz Bulma Modülü .....43**
*2.7.2.1. Adaboost Algoritması İle Yüz Bulma .....43*
*2.7.2.1. Yüz Bulma Modülü Temel Bilgiler ve Çalışma Şekli .....46*
**2.7.3. Yüz Tanıma Modülü .....50**
*2.7.3.1. Yüz Tanıma Modülü Temel Bilgiler ve Çalışma Şekli .....50*
*2.7.3.2. Öklid Mesafesi ile Karar Verme .....55*
*2.7.3.3. Yüz Tanıma Sistemi Önerilen Yöntem .....56*
**2.7.4. Programda Kullanılan Önemli Fonksiyon Ve Yöntemler .....56**
*2.7.4.1. Yüz Bölgesinin Tespiti .....56*
*2.7.4.2. Yüz Tanıma .....58*
*2.7.4.3. Gerekli EmguCV Kütüphaneleri .....58*
**3. BULGULAR VE TARTIŞMA.....60**
**3.1. Yale Veritabanı Deneyleri .....60**
**3.2. ORL Veritabanı Deneyleri .....63**
**4. SONUÇLAR VE ÖNERİLER .....66**
**5. KAYNAKLAR .....68**
**ÖZGEÇMİŞ .....71**

# ŞEKİL LİSTESİ


| | | Sayfa No |
|---|---|---|
| Şekil 2.1. | Yüz tanıma akış şeması | 11 |
| Şekil 2.2. | Şablon eşlemeye dayalı yüz tanıma | 13 |
| Şekil 2.3. | Işıklandırma faktörü | 15 |
| Şekil 2.4. | Özyüzler (Eigenfaces) | 18 |
| Şekil 2.5. | TBA sınıflandırma doğrultusu | 19 |
| Şekil 2.6. | NxN görüntü örneği ve dönüştürülmüş durumu | 19 |
| Şekil 2.7. | Ortalama görüntünün hesaplanması | 20 |
| Şekil 2.8. | Elde edilen en büyük özdeğerlere sahip özyüzler | 22 |
| Şekil 2.9. | Özyüz algoritması ile sınıflandırma işlevsel blok şeması | 23 |
| Şekil 2.10. | DAA için sınıfları ayıran yön | 24 |
| Şekil 2.11. | Fisher yüz algoritması ile sınıflandırma işlevsel blok seması | 27 |
| Şekil 2.12. | LBP değerinin hesaplanmasının grafiksel ifadesi | 28 |
| Şekil 2.13. | (a) (8,1), (b) (16,2), (c) (8,2) göre oluşturulmuş LBP | 29 |
| Şekil 2.14. | Orijinal imge (Solda) ve LBP operatörü ile oluşturulmuş imge (Sağda) | 29 |
| Şekil 2.15. | (a) 3×3 (b) 5×5 (c) 7×7 Bölgeye ayrılmış imge | 30 |
| Şekil 2.16. | Histogramların eklenmesi | 30 |
| Şekil 2.17. | Görüntünün histogramları a. Histogram eşitlemeden önceki hali (sol sütun), b. Histogram eşitleme sonrası (sağ sütun) | 31 |
| Şekil 2.18. | Histogram dönüşüm fonksiyonu | 32 |
| Şekil 2.19. | 512x512'lik bir görüntünün 64 eşit kare bölgeleye ayrılmış yapısı | 33 |
| Şekil 2.20. | (a) Orjinal, (b) eşikleme +CLAHE uygulanmış (bi-lineer enterpolasyondan önce), (c) (b) görüntüsüne bi-lineer enterpolasyon uygulandıktan sonra elde edilen sonuç görüntüsü | 34 |
| Şekil 2.21. | (a) Orijinal görüntü (b) Gauss filtresi uygulanmış görüntü | 36 |

| | | |
|---|---|---|
| Şekil 2.22. | (a) Orijinal Görüntü, (b) 3x3 Medyan Filtre Soncu, (c) 5x5 Medyan Filtre Sonucu, (d) 7x7 Medyan Filtre Sonucu | 37 |
| Şekil 2.23. | (a) Orijinal Görüntü, (b) Laplace Filtresi Uygulanmış Görüntü | 39 |
| Şekil 2.24. | Yüksek geçirgenli filtre uygulaması | 40 |
| Şekil 2.25. | Sıfır toplamlı kenar filtresi | 41 |
| Şekil 2.26. | Yüz tanıma sistemi mantıksal yapısı | 42 |
| Şekil 2.27. | "Opencv-Haar-Classifier" katmanlı yapısı | 44 |
| Şekil 2.28. | Adaboost algortiması ile tespit edilen yüzler | 45 |
| Şekil 2.29. | Yüz bulma modülü; görüntü elde etme özelliğinin kameradan al yapılması | 46 |
| Şekil 2.30. | Yüz bulma modülü akış diyagramı | 47 |
| Şekil 2.31. | Görüntü elde etme özelliğinin dosyadan yükle yapılması | 48 |
| Şekil 2.32. | Yüz tanıma modülü | 50 |
| Şekil 2.33. | Eğitici kaydının 4 farklı şekilde yapılması | 51 |
| Şekil 2.34. | Veritabanında eşleşen görüntü | 52 |
| Şekil 2.35. | Tümü seçildiğinde yüz tanıma oranı grafiği | 53 |
| Şekil 2.36. | Tümü seçildiğinde yüz tanıma ortalama hesaplama süresi grafiği | 53 |
| Şekil 2.37. | Yüz tanıma oranları grafiği | 54 |
| Şekil 2.38. | Yüz tanıma ortalama hesaplama süresi grafiği | 55 |
| Şekil 3.1. | Yale veritabanından örnek görüntüler | 60 |
| Şekil 3.2. | ORL veritabanından rastgele seçilen 3 bireyin görüntüleri | 63 |

# ÇİZELGE LİSTESİ


| | | **Sayfa No** |
|---|---|---|
| Çizelge 1.1. | Biyometrik tanıma sistemlerinin çeşitlerinin taradıkları özellikler | 7 |
| Çizelge 1.2. | Biyometrik teknolojilerin karşılaştırılması | 7 |
| Çizelge 3.1. | Deneklere ait normal görüntülerin yer aldığı eğitim veritabanında yapılan test sonuçları(%'lik tanıma oranları) | 61 |
| Çizelge 3.2. | Deneklere ait normal ve merkezden aydınlanmış görüntülerin yer aldığı eğitim veritabanında yapılan test sonuçları(%'lik tanıma oranları) | 62 |
| Çizelge 3.3. | Deney 1 ve deney 2 sonuçların karşılaştırılması | 63 |
| Çizelge 3.4. | ORL veritabanı deney sonuçları | 64 |
| Çizelge 3.5. | ORL veritabanı deney sonuçları | 65 |
| Çizelge 3.6. | Deney 1 ve deney 2 sonuçların karşılaştırılması | 65 |

# SİMGELER VE KISALTMALAR

| | |
|---|---|
| C# | C Sharp |
| LDA | Doğrusal Ayraç Analizi |
| HE | Histogram Eşitleme |
| ORL | Olivetti-Oracle Research Laboratory |
| PCA | Temel Bileşen Analizi |
| LBP | Yerel İkili Örüntü (Local Binary Pattern) |

# ÖZET

## YÜZ BULMA VE TANIMA SİSTEMLERİ KULLANARAK KİMLİK TESPİTİNİN YAPILMASI

Recep HOLAT
Düzce Üniversitesi
Fen Bilimleri Enstitüsü, Elektrik Elektronik Mühendisliği Anabilim Dalı
Yüksek Lisans Tezi
Danışman: Yrd. Doç. Dr. Selman KULAÇ
Haziran 2014, 82 sayfa

Son yıllarda yüz tanıma alanında önemli başarılar elde edilmiştir. Yüz tanıma, bankacılıkta kimlik onaylamada, kontrollü alanlara girişte, başta havaalanlarında olmak üzere güvenliğin üst düzey olduğu yerlerde, makineleri kontrol etmede ve kişilerin takibinde kullanılan özel bir örüntü tanımadır. Bu tez çalışmasında bir yüz tanıma sistemi tasarlanmış, PCA (Temel Bileşen Analizi), LDA (Doğrusal Ayraç Analizi) ve LBP (Yerel İkili Örüntü) yüz tanıma yöntemleri kullanılarak Yale ve ORL yüz veritabanları üzerinde test edilmiştir. Yüzün tespit edilmesinde Adaboost algoritması kullanılmıştır. Yale veritabanı, sağdan aydınlanmış, merkezden aydınlanmış, soldan aydınlanmış, gözlüksüz, gözlüklü, normal, göz kırpmış, uykulu, şaşkın, mutlu, üzgün yüz görüntüleri içermektedir. Yüz tanıma ön işleme adımlarında HE (Histogram Eşitleme), HE+Medyan Filtresi, HE+Gaussian Filtresi, HE+Laplace Filtresi kullanılmıştır. Görüntünün poz ve aydınlatma durumuna göre sistemin en uygun yöntemi seçmesi sağlanmıştır. Bu şekilde yüz tanıma oranında %6' ya kadar olan başarım artışları elde edilmiştir.

Uygulama Microsoft Visual Studio 2010 C#.Net programı kullanılarak geliştirilmiştir. Görüntü işleme algoritmaları için EMGU CV kütüphanesi, veritabanı işlemleri için SQL Server 2008 Express kullanılmıştır.

**Anahtar sözcükler:** Görüntü Tanıma, Yüz Bulma, Yüz Tanıma

# ABSTRACT

## ID IDENTIFICATION BY USING FACE DETECTION AND RECOGNITION SYSTEMS

Recep HOLAT
Duzce University
Graduate School of Natural and Applied Sciences, Departmant of Electrical Electronics Engineering
Master of Science Thesis
Supervisor: Assist. Yrd. Doç. Dr. Selman KULAÇ
July 2014, 82 pages

In recent years, significant achievements have been achieved in the field of face recognition. Face recognition are special pattern recognition which are used in banking for identity approving and the entrance of controlled areas, the places where the security control impending to airports, to control machines,to follow-up of persons. In this study, A face recognition system is designed, implemented and tested on the Yale and ORL face databases have been performed by using PCA (Principal Component Analysis), LDA (Linear Discriminant Analysis), LBP (Local Binary Patterns) face recognition methods. Yale database consist of right-light, center-light, left-right, no glasses, glasses, normal, wink, sleepy, surprised, happy, sad images. In the face recognition pre-processing steps, HE (Histogram Equalization), HE+ Median Filter, HE+Gaussian filter, He+Laplace Filter was used. It is provided to select the appropriate method from the system depending on state of image. Thus the face recognition performance increases of up to 6% was gained.

Application was developed by using Microsoft Visual Studio 2010 C #. EMGU CV library for image processing algorithms and SQL Server 2008 Express for database processing were used

**Keywords:** Face Detection, Face Recognition, Image Recognition

# EXTENDED ABSTRACT

## ID IDENTIFICATION BY USING FACE DETECTİON AND RECOGNİTİON SYSTEMS

Recep HOLAT
Duzce University
Graduate School of Natural and Applied Sciences, Departmant of Electrical Electronics Engineering
Master of Science Thesis
Supervisor: Assist. Yrd. Doç. Dr. Selman KULAÇ
July 2014, 82 pages

### 1. INTRODUCTION:

Face recognition has been interested for many researchers. In particular, biometric authentication is a popular research topic. Face recognition systems automatically identfy or verify a person from images or videos. In recent years, significant achievements have been achieved in the field of face recognition. Today, face recognition are special pattern recognition which are used in banking for identity approving and the entrance of controlled areas, the places where the security control impending to airports, to control machines,to follow-up of persons.

Face recognition is a visual pattern recognition problem. A face recognition system generally consist of four main parts: detection, alignment, feature extraction and matching. The success of face recognition under the controlled environment has not been achieved in uncontrolled environments. The reasons for this; variations in lighting, pose and facial expression changes differences.

Numerous image processing and facial recognition algorithms have been developed within the scope. In this thesis, PCA (Principal Component Analysis), LDA (Linear Discriminant Analysis), LBP (Local Binary Patterns) face recognition methods are used for face recognition.

## 2. MATERIAL AND METHODS:

Face recognition systems automatically identfy or verify a person from images or videos. The application which is developed in thesis consists of five main sections. These are;

- ✓ Data acquisition for both training and testing from video or input image
- ✓ Face detection which is the process of determining the presence of a face and locating it within many objects and patterns in the input image or video frame
- ✓ Reducing the impact of light and noise on the image and image enhancement
- ✓ Feature extraction
- ✓ Face recognition

Face detection is the first step of face recognition system. Output of the detection can be location of face region and location of face region. The AdaBoost algorithm was used for face detection

The detected face image dimensions should be equal with the images contained in the database. Therefore, the image is resized to the size equal to the faces contained in the database. In this study, face images were resized 50 x 50 pixels.

Lighting and poses changes that cause problems for face recognition system are among the topics. To reduce this effect, it should be preprocessed. Preprocessing operations are histogram equalizing of grayscale face image. Histogram Equalization, Local Histogram Equalization, Contrast Limited Adaptive Histogram Equalization were used in this thesis.

PCA (Principal Component Analysis), LDA (Linear Discriminant Analysis), LBP (Local Binary Patterns) face recognition methods are used for feature extraction and face recognition.

The system of designed for face recognition was used Euclidean distance for recognition as calculating the minimum Euclidean distance between the test image and others.

## 3. RESULTS AND DISCUSSIONS:

Tests were performed on the Yale face and ORL face databases by using application which developed in C # (C Sharp) programming language. Yale database consist of right-light, center-light, left-right, no glasses, glasses, normal, wink, sleepy, surprised, happy, sad images of 15 distinct subjects. ORL database consist of ten different images of each of 40 distinct subjects.

Two tests were performed in yale database. In a first test, subjects belonging to the set of trainers just normal images added and all the other images were made with the recognition process. According to test results, the most successful method is LBP+HE (77 %). The recognition rate of the proposed method is 81%. Proposed method gave nearby 4 % better results. Proposed method is provided to select the appropriate method from the system depending on state of image.

In a second test, subjects belonging to the set of trainers just normal and center-light images added and all the other images were made with the recognition process. According to test results, the most successful method is LBP+HE+Medyan (81 %). The recognition rate of the proposed method is 91 %. Proposed method gave nearby 6 % better results.

## 4. CONCLUSION AND OUTLOOK:

Face recognition system was designed and tested. Test results show that system has acceptable performance. The applications developed include two modules. One of these modules is create the training database and other is real-time recognition capable module.

Facial expressions, poses and light variation adversely affects face recognition rate. To decrease this negative effect were used median, Gauss, Laplace filters. The recognition success rate and performance can be improve by doing the following

- ✓ Face recognition algorithms should be implemented by the researcher.
- ✓ To detect automatically pose and lighting condition.

# 1. GİRİŞ

## 1.1. BİYOMETRİK SİSTEMLER

Biyometri, kişinin fiziksel ya da davranışsal özelliklerini analiz ederek kimlik doğrulaması yapan bir bilimdir. Bu özellikler; kişiye özgü olan ve aynı zamanda kişiden kişiye kolayca aktarılamayacak kadar güvenilir, hayatın başlangıcından sonuna kadar değişmeyen kalıcı bir yapıdadır. Bu nedenle değiştirilmesi daha zordur. Bu özelliğinden dolayı çeşitli alanlarda kullanılmaktadır. Bilgisayar teknolojisindeki gelişmeler ve yüksek güvenlik gerektiren uygulamalara gereksinimden dolayı biyometri üzerinde yapılan çalışmaların sayısı artmaktadır [1].

Biyometri, fiziksel giriş çıkış kontrol noktalarında kimlik tespitinin yapılmasında ve takibinde, benzer güvenlik fonksiyonlarının uygulanmasında, bilgisayar ve bilgisayar ağlarında haberleşmenin sağlanmasında ve kaynakların paylaşımında, ağ güvenliğinin sağlanmasında kullanılmaktadır. Ayrıca, personel devam kontrol sistemlerinde, ticari işlemler ve benzer birçok alanda, banka güvenliğinde elektronik fon transferi veya ATM güvenliği gibi işlemlerde, çek ve kredi kartı işlemlerinde, gümrük ve göç işlemlerinde, ulusal kişi tanıma sistemlerinde daha güvenli ve hızlı işlem için, seçmen ve sürücü kayıtlarında, internet işlemlerinde verimli bir şekilde uygulandığı ve insan hayatında geniş bir alanda kullanılan ve çözümler sunan bir teknoloji olduğu görülmektedir [2].

Günümüzdeki biyometrik tanıma yöntemleri şunlardır [3]:

Fizyolojik özelliklere dayalı olan biyometrik tanıma sistemleri:

1. Parmak İzi Tanıma
2. El Geometrisi Tanıma
3. DNA Tanıma
4. Retina Tanıma
5. Yüz Tanıma
6. Ses Tanıma
7. Yüz Termogramı
8. İris Tanıma
9. Damar Tanıma

Davranışsal özelliklere dayalı olan biyometrik tanıma sistemleri:

1. İmza Atımı
2. Yürüyüş
3. Konuşma

Biyometrik tanıma sistemlerinin çeşitlerinin taradıkları özellikler Çizelge 1.1'de, biyometrik teknolojilerin karşılaştırılması Çizelge 1.2'de verilmiştir [3].

**Çizelge 1.1.** Biyometrik tanıma sistemlerinin çeşitlerinin taradıkları özellikler [3].

| **Biyometrik Karakteristik** | **Özelliklerin Açıklaması** |
|---|---|
| Parmak İzi | Parmak satırları, gözenek yapısı |
| İmza Tanıma | Basınç ve hız ile yazma farkları |
| Yüz Geometrisi | Göz, burun vs arası uzaklıklar |
| İris Tanıma | İris deseni |
| Retina | Retina yapısına(desenine) göre |
| El Geometrisi | Parmak ve avuç içi ölçülerine göre |
| Parmak Geometri | Parmak ölçme |
| El Damar Yapısı | Elin arkası, parmak veya avuç içi damar yapısı |
| Kulak Formu | Kulağın belirgin boyutları |
| Ses | Ton ya da ses rengi |
| DNA | Kalıtsal taşıyıcı olan DNA |

**Çizelge 1.2.** Biyometrik teknolojilerin karşılaştırılması [3].

| **Biyometrik Karakteristlikler** | **Evrensellik** | **Eşsizlik** | **Süreklilik** | **Elde Edilebilirlik** | **Performans** | **Kabul Edilebilirlik** | **Yaygınlık** |
|---|---|---|---|---|---|---|---|
| DNA | Yüksek | Yüksek | Yüksek | Düşük | Yüksek | Düşük | Düşük |
| Kulak | Orta | Orta | Yüksek | Orta | Orta | Yüksek | Orta |
| Yüz | Yüksek | Düşük | Orta | Yüksek | Düşük | Yüksek | Yüksek |
| Yüz Termogramı | Yüksek | Yüksek | Düşük | Yüksek | Orta | Yüksek | Düşük |
| Parmak İzi | Orta | Yüksek | Yüksek | Orta | Yüksek | Orta | Orta |
| El Geometrisi | Orta | Orta | Orta | Yüksek | Orta | Orta | Orta |

**Çizelge 1.2 (devam).** Biyometrik teknolojilerin karşılaştırılması [3].

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| İris | Yüksek | Yüksek | Yüksek | Orta | Yüksek | Düşük | Düşük |
| Retina | Yüksek | Yüksek | Orta | Düşük | Yüksek | Düşük | Düşük |
| Ses | Orta | Düşük | Düşük | Orta | Düşük | Yüksek | Yüksek |
| Damar | Yüksek | Orta | Orta | Orta | Yüksek | Orta | Düşük |

## 1.2. YÜZ TANIMA SİSTEMLERİNE GENEL BAKIŞ

Yüz tanıma, günümüzde oldukça ilgi gören, etkin bir araştırma alanıdır. Bu ilgi ve alakanın en önemli nedeni; insanın sahip olduğu mükemmel yüz tanıma kabiliyetini çözme ve bu yeteneği makineye de kazandırma tutkusudur. Bilimsel araştırma ilgilerinin yanı sıra yüz tanıma; gözetim, güvenlik, haberleşme ve insan-bilgisayar etkileşimleri gibi pek çok ticari ve adli uygulama alanlarına sahiptir ve bu uygulama alanları her geçen gün derinleşmekte ve artmaktadır [4].

Biyometrik kimlik doğrulama, üzerinde daha çok durulan bir alandır. Kontrol altında tutulan ortamlarda elde edilen başarıma kontrolsüz ortamlarda ulaşılamamıştır. Bunun nedenleri; aydınlatmadaki farklılıklar, poz değişimleri ve yüz ifadesi farklılıklar sayılabilir.

Yapılan araştırmalar yüzün etkin bir biyometrik belirleyici olduğunu göstermektedir [5]. Yüz tanıma son zamanlarda çok fazla ilgi gören bir alan olmuştur [6].

Yüz tanıma teknolojileriyle kimlik doğrulama ve bilgisayar insan etkileşimi, kullanıcı dostu, kullanımı kolay ve daha güvenli olabilir. Retinal tarama ve parmak izi analizi gibi diğer pek çok biometrik kimlik tanıma yöntemleri, katılımcıların dayanışması üzerine kuruluyken, yüz resimlerine dayanan kimlik tanıma sistemi katılımcıların dayanışmasını ve bilgisinin gerektirmeden de çoğu zaman etkin bir şekilde çalışır.

Yüz tanıma, insanlar için sıradan bir olaydır ve insan algı sisteminin önemli bir yeteneğidir. Yapılan deneyler ve bulgular gösteriyor ki, 1 ila 3 günlük arasındaki bebekler yüzler arasında ayrım yapabilmektedirler. İnsanlar için normal olan bu yeteneği, makinelere kazandırmak ne kadar zor olabilir ki? Burun, göz, ağız gibi organların iç özellikleri veya kafa şekli, saç çizgisi gibi dış özellikler başarılı bir yüz

tanıma için kullanılabilir mi? Bir insan, resmi nasıl analiz eder ve onu beyninde nasıl kodlar? David Hubel ve Torsten Wiesel çalışmalarında gösterdi ki, insan beyni çizgiler, kenarlar, açılar veya hareket gibi bir sahnenin belirgin yerel özelliklerine yanıt veren özelleşmiş sinir hücrelerine sahiptir. Yüz tanıma sistemini, insan algı sisteminden ilham alınarak makinelere otomatik yüz tanıma yeteneği kazandırma olarak tanımlayabiliriz.

Makine veya bilgisayarlar için otomatik yüz tanıma problemi, resimler üzerinden veya video görüntülerinden bir veya daha fazla kişiyi yüz veritabanını kullanarak tanıma ve doğrulama problemi olarak tanımlanabilir. Bunun için ilk önce yüzü ayırt eden ve karakterize eden öznitelikler çıkartılmalı ve ondan sonra veritabanında eşleştirilmelidir [4].

# 2. MATERYAL VE YÖNTEM

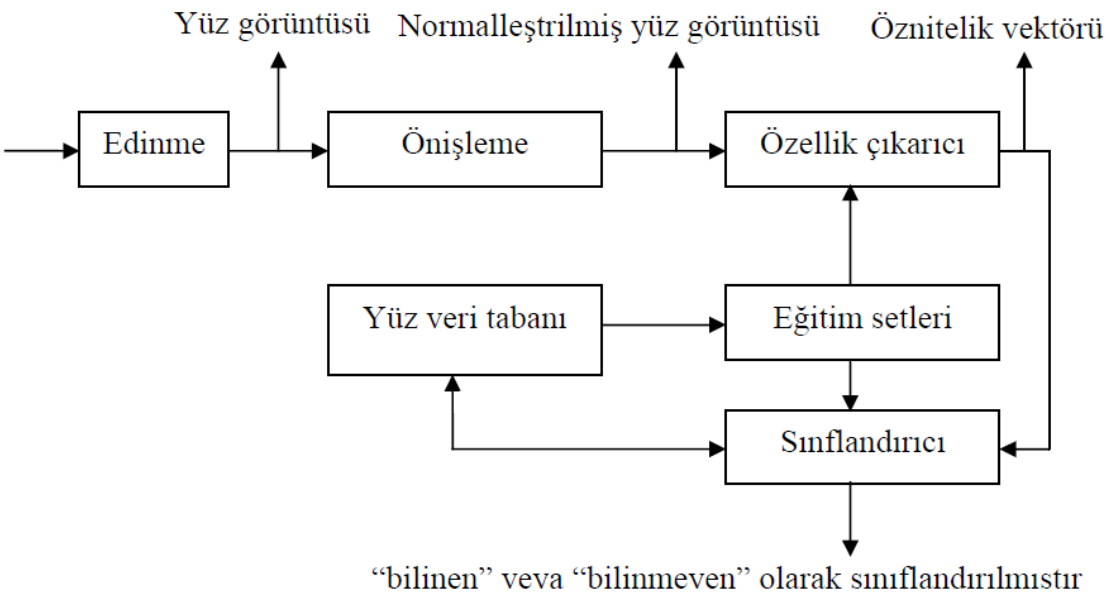
Genel olarak yüz tanıma, sorgulanan bir yüzün veritabanındaki kayıtlı bilinen yüzlerle karşılaştırılması sonucu, sorgulanan yüzün kimliğini doğrulamak veya bu yüzün kimliğini tespit etmek olarak tanımlanabilir. Kimlik doğrulama işleminde bire bir karşılaştırma yapılarak "aynı kişi mi?" sorusuna cevap aranır. Kimlik belirleme işleminde ise çoklu karşılaştırma yapılarak "bu kişi kim?" sorusu cevaplanmaya çalışılır [7].

## 2.1. YÜZ TANIMA SİSTEMİ

Yüz tanıma sistemleri sayısal bir görüntüden kişinin otomatik olarak belirlenmesini sağlayan bilgisayar destekli uygulamalardır. Yüz tanıma problemi, herhangi bir video ya da imgeden bilinmeyen kişi/kişilere ait öznitelikleri eğitim setindeki etiketli özniteliklerle karşılaştırarak bu kişi/kişilerin belirlenmesi problemi olarak tanımlanmaktadır [8].

Geometrik şekillere bağlı metotlar üzerindeki bazı çalışmalar yapılmasına rağmen ilk otomatikleştirilmiş yüz tanıma sistemini geliştiren insan Kanade' dır [9]. Geometrik şekiller metodu insanların bir takım aynı özelliklerine bağlı olarak ortaya çıkar. Bunlar; iki göz, bir burun, bir ağız v.s.'dir. Bu bileşenler arasındaki ilgili boyut ve mesafe kişiden kişiye farklılaştığından dolayı bu durum sınıflandırma amaçlı kullanılabilir.

Yüz tanıma genel anlamda, statik bir görüntüden ya da bir video görüntüsünden bir veya daha fazla kişiyi, veri tabanında yer alan yüz görüntülerini kullanarak kimliklendirme ya da doğrulamadır. Bir yüz tanıma sisteminin uygulama akış şeması, Şekil 2.1' deki gibi beş adımdan oluşmaktadır [10].




**Şekil 2.1.** Yüz tanıma akış şeması [10].

Bu adımlar;

- ✓ Edinme modülü. Bu, yüz tanıma sürecinin giriş noktasıdır. Söz konusu yüz görüntüsünün elde edildiği modüldür. Başka bir deyişle, bu modülde, kullanıcıdan yüz tanıma sistemine bir yüz görüntüsü sunması istenir. Bu görüntü statik bir görüntü veya video üzerinde yakalanan bir sahne/kare olabilir.
- ✓ Yüzün Tespit Edilmesi. Bir görüntünün tamamı yüze ait olmadığından, ilk olarak yüzün tespit edilmesi gerekmektedir;
- ✓ Ön İşleme Modülü. Yüzün tespit edilmesini etkileyen faktörlerin çıkarılması işleminin yapıldığı adımdır.
- ✓ Özellik Çıkarımı. Yüzün ayırt edici özelliklerinin çıkarılmasını ve tanıma, doğrulama yapılmasını sağlayan adımdır.
- ✓ Sınıflandırıcı Modülü. Yüz veritabanında bulunan görüntülerle yeni görüntünün karşılaştırıldığı sınıflandırıcı adımı ve görüntünün bilinen veya bilinmeyen olarak sınıflandırıldığı adımdır. Bu adımlar temel adımlardır. Ortam ve poz durumlarına göre başka adımlar eklenebilir.

Yüz tespit adımı yüz tanıma işleminin başlangıç noktasıdır. Yüz görüntüsü manyetik bir diskte bulunan bir görüntü olabilir; bir görüntü yakalayıcı tarafından alınabilir veya bir tarayıcı yardımıyla kâğıttan taranabilir. Görüntüdeki ışıklandırma yüzün tanınmasını zorlaştıracak ve hatta tanınamamasına neden olacak şekilde olabilir. Ya da arka plan o

kadar karışık olabilir ki arka plan içerisinde yüzün tespit edilmesi işlemi yapılamayabilir. Buna benzer nedenlerden dolayı sisteme önişleme adımının eklenmesi gerekmektedir.

Sınıflandırma modülünde örüntü sınıflandırıcının da yardımıyla, yüze ait özellikler ile yüz veritabanındaki resimlerin özellikleri karşılaştırılır. Karşılaştırma sonunda resim "tanınıyor" ya da "tanınmıyor" olarak sınıflandırılır. Eğer istenirse tanınmayan resimde yüz veritabanına eklenebilir. Böylece bir daha ki tanıma işleminde bu resim tanınanlar sınıfında yer almış olur. Özellik çıkarma ve sınıflandırma modülleri, yeni bir yüzün tanınması sürecinde eldeki tanınan sınıf elemanlarının özelliklerini kullanırlar [10].

## 2.2. YÜZ TANIMA UYGULAMA ALANLARI

Yüz tanıma sisteminin günlük yasamda pek çok uygulama alanı mevcuttur. Bunlardan bazıları; geçiş kontrol, personel devam kontrol, bankacılık ve finans, güvenlik ve emniyet, bilgi veri güvenliği, ulaşım ve pasaport ve sağlık sektöründeki uygulamaları olarak sıralanabilir. Bu alanlar aşağıdaki gibi özetlenebilir [11].

- ✓ Güvenlik (Binalara giriş ve çıkış, havaalanı/liman, sınır güvenlik noktaları, ATM makineleri, bilgisayar/ağ güvenliği, multimedya iş istasyonları üzerinde e-posta doğrulama).
- ✓ Gözetim (Suçluların aranması).
- ✓ Genel kimlik doğrulama (Banka işlemleri, elektronik ticaret, şifre, araç lisansı)
- ✓ Adalet sistemi (Adli tıp, olay sonrası inceleme).
- ✓ İmge veritabanı araştırması (Kayıp arama).
- ✓ "Akıllı Kart" uygulamaları (İmgeleri bir veritabanında korumak yerine, akıllı kart içinde bar kod olarak saklanabilir, kimlik doğrulama kayıtlı görüntü ile gerçek zamanlı görüntünün eşleşmesiyle sağlanır).
- ✓ İnsan bilgisayar etkileşimi (İçerik algılama, yaşlı ve çocuk koruma merkezlerinde davranışları izlemek, müşteri tanıma ve onların isteklerini cevaplama).
- ✓ Video indeksleme (Video içinde yüzleri etiketleme).

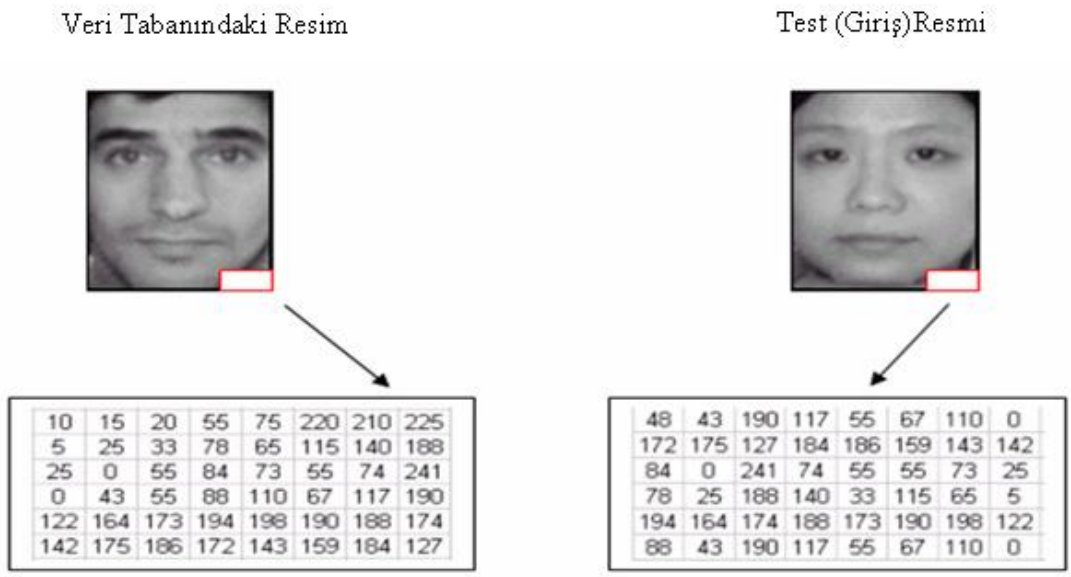
## 2.3. YÜZ TANIMADA KULLANILAN TEMEL YAKLAŞIMLAR

### 2.3.1. Özellik Tabanlı Yöntemler

Özellik tabanlı yaklaşımda, göz, burun ve ağız genişliği, kaş kalınlığı, yüz genişliği, yanaklar ve konumları geometrik özellik olarak yüzü temsil eder. Özellik yüz yaklaşımı şablon tabanlı yöntemlere göre daha az bellek ihtiyacı duyar, daha hızlı ve kısa sürede tanıma yapar. Özellikle, yüz ölçek normalleşmesi ve tahmini poz tabanlı 3-D kafa modeli için kullanışlıdır. Bununla birlikte, özelliklerin kusursuz çıkarımı; uygulanmasının zor olduğunu göstermiştir.

### 2.3.2. Şablon Tabanlı Yöntemler

En basit şablon eşleme, genellikle orijinal yüz görüntüsünün kenar haritası olan tek bir şablon kullanarak yüzün temsil edilmesidir. En karmaşık şekilde şablon eşleştirme, farklı bakış açılarından tanıma yapmak için her yüz için birden çok şablonun kullanılmasıdır. Bir diğer önemli şablon eşleme çeşidi; tek bir noktadan göz, burun, ağıza karşılık gelen yüze ait daha küçük özellik şablon kümelerini kullanmaktır [12]. Şekil 2.2'de bir örneği verilmiştir.




**Şekil 2.2.** Şablon eşlemeye dayalı yüz tanıma [12].

### 2.3.3. Görünüş Tabanlı Yöntemler

Bu yaklaşım, yüz görüntülerini düşük boyutlarda lineer altuzayı içine yansıtmayı hedeflemektedir. Böyle bir altuzayın ilk sürümü, Temel Bileşen Analizi (PCA) ile eğitim görüntü setlerinden inşa edilen özyüz alanıdır. Daha sonra özyüzler kavramı, yüze ait özelliklerin tespit edilmesinde, özağız, özgöz gibi özvektörlere genişletilmiştir. Son zamanlarda fisher yüz uzayı ve aydınlatma altuzayı değişen aydınlatma koşullarında tanıma için önerilmiştir [12].

Görünüş tabanlı yöntemlerde resimler direkt piksellerin aldığı yoğunluk değerleri ile yada bu piksellere uygulanan Gabor filtre çıktıları veya Yerel İkili Örüntü [13] gibi lokal imge tanımlama teknikleri ile gösterilirler. Bu imge tanımlamaları ile gösterilen yüz resimleri genellikle çok yüksek boyutludur ve resimler vektörlere çevrilerek bu yüksek boyutlu uzaydaki noktalar olarak kabul edilir. Birçok görünüş tabanlı yaklaşımlarda yüz resim vektörlerinin vektör uzayındaki dağılımını analiz etmek için istatistiksel teknikler kullanılır.

Son yıllarda ise yüz verilerinin yüksek boyutlu uzayda doğrusal olmayan imge manifoldları üzerinde olduğu kabul edilmiş ve yüz tanıma için doğrusal olmayan ayırt etme yöntemleri kullanılmaya başlanmıştır. Bu amaçla ISOMAP [14], Locally Linear Embedding [15], Laplacianfaces [16] gibi yöntemlerle birlikte kernel hilesini kullanan Kernel Temel Bileşen Analizi [17], Kernel Doğrusal Ayırıcı Analizi [18] ve Kernel Ayırt Edici Ortak Vektör [19] gibi yöntemler denenmiş ve başarılı sonuçlar elde edilmiştir. Kernel hilesini kullanan yöntemlerdeki ana prensip, yüz verilerini verilerin doğrusal olarak ayrıştırılabileceği çok boyutlu bir uzaya çıkarmak ve doğrusal ayırt etme tekniklerini bu uzayda uygulamaktır. Tüm işlemler kernel hilesi sayesinde etkili bir şekilde giriş uzayında yapılmakta ve çok boyutlu uzayda çalışılmamaktadır.

## 2.4. YÜZ TANIMADAKİ ZORLUKLAR

Görüntü işleme uygulamalarında başarımın yüksek olmaması resimdeki poz farklılıkları, ölçek ve aydınlatma gibi faktörlerden kaynaklanmaktadır. Tanıma işlemi, yüz resimlerindeki yüz ifadesi; eğilme, değişik saç şekilleri, makyaj ve sakalın veya bıyığın var olması veya yok olması gibi faktörlerden etkilenir. Bazı durumlarda aynı kişinin iki resmi arasındaki fark, farklı kişilerin resimleri arasındaki farktan daha

fazladır. Yaşlanma da tanımayı etkileyen diğer bir faktördür. Bir kişinin belli aralıklarla çekilmiş iki resmi arasında büyük farklar olabilir. Tanımayı zorlaştıran bir diğer faktör de resimdeki gürültülerdir.

### 2.4.1. Aydınlatma Varyasyonu

Bir görüntünün ışıklandırma durumları, iç-dış ortam farklılıkları, zaman farklılıkları gibi koşulların değişmesi ile değişebilir. Yüzün yapısından dolayı, doğrudan ışık veren bir kaynak ile yüz üzerinde gölgeler meydana gelir ve bu gölgeler ile yüz öğeleri çok vurgulanabilmekte ya da ayırt edilemez olabilmektedir.

Işıklandırma koşullarının yorumlanması oldukça güçtür. Işıklandırma farklılığı 2 şekilde oluşabilir. Biri, yüzden yansıyan ışık miktarı ile, diğeri gamma düzeltmesi, kontrast ayarı gibi dahili kamera kontrolleri üzerindeki tutarsızlıktan dolayı oluşur. Her ikisinin de yüz görünümü üzerinde büyük etkileri oluşmaktadır. Ancak ikinci faktör insan gözü ile algılanamamakta ve bilgisayarlı görme yöntemleri ile değerlendirilmektedir [20].

**Şekil 2.3.** Işıklandırma faktörü [20].

CMU PIE veritabanında 2 adet yüz görüntüsü Şekil 2.3'de gösterilmiştir. Bu görüntülerden ilki normal oda koşullarında, diğeri ise flash kullanılarak görüntülenmiştir [20].

### 2.4.2. İfade Çeşitlilikleri

Yüz değişken ve hareket halinde bir öğe olduğundan, ifade çeşitlilikleri yüz tanıma uygulamalarını birçok açıdan etkiler. İnsan yüzündeki ifadelerin yüz tanımaya olan

etkileri hala tam anlaşılmamaktadır [6], bu nedenle yüz tanıma çalışmalarının çoğunda normal ifadeler ve tebessüm ifadeleri dışında yüz ifadeleri tercih edilmemektedir.

İfade varyasyonları yüz görünümünü önemli derecede etkiler. Aşırı ifadeler haricinde, örneğin çığlık gibi, geliştirilen algoritmalar yüz ifadelerini tanımada oldukça güvenilirdir.

Donato [21] oniki yüz ifadesini sınıflandırmak için çeşitli yöntemleri araştırmıştır. Yaptığı çalışmalar, Gabor tabanlı filtreler ve Bağımsız Bileşen Analizi (ICA) yöntemlerinin diğer yöntemler arasında en iyi performansa sahip olduğunu göstermiştir.

Tian ve arkadaşları [22] yüz ifadelerin analizini yapan Otomatik Yüz Analiz sistemi geliştirmişlerdir. Yüze ait ifadeler, hem kalıcı özellikler kaşlar, gözler, ağız gibi hem de ön-yüz görünümündeki geçici yüz özellikleridir. Yaptıkları çalışmalarda yüzün üst tarafına ait ifadeler de tanıma oranını % 96,4 alt tarafına ait tanıma oranını % 96,7 olarak raporlamışlardır.

### 2.4.3. Tıkanıklıklar/Kapanmalar

Yüz görüntüsü elde edilirken şapka, gözlük gibi aksesuarlara sahip olması ya da diğer yüz objeleri ile üst üste gelmesi durumunda ortaya çıkarır. Kapanma etkisi yüz görüntüsü üzerinden veri kaybı anlamına gelmekte olup, tanımayı olumsuz yönde etkiler.

Yerel yaklaşımlar yüzü farklı bölümlere ayırarak eşleme yapar. Bu yöntem tıkanan görüntülerdeki sorunu çözmek için en popüler yoldur. Martinez ve arkadaşları [23] bu metodu kullandılar. Bu yöntemle Gauss dağılımı kullanarak, yüzün her yüz görüntüsünü **k** farklı yerel bölgeye ayırdılar. Gauss dağılımı yerelleştirme hatası sorunlarını ortadan kaldırmak içindir. Bu yöntem sadece yüzün tıkalı olan bölgelerinde başarılıdır.

### 2.4.4. Poz Varyasyonu

Yüz 3 boyutlu bir nesne olduğundan kamera açısı, elde edilen yüz görüntüsünü etkiler. Bu nedenle kamera açısı ve poz değişiklikleri yüz tanıma uygulamalarında başarı oranını etkiler. Aynı şekilde bir açıdan alınan görüntü ile farklı bir açıdan alınan görüntü mevcut eğitim kümesi ile kıyaslanırken açı değerleri aynı olmadığından başarı oranını düşürür. Yapılan bir çalışma, 32 derece' den az şekilde dönmüş yüz açısı bulunan

görüntülerde, 32 dereceden fazla dönmüş yüz açısı bulunanlara oranla, yüz tanıma başarısının arttığını göstermektedir [24].

## 2.5. YÜZ TANIMA TEKNİKLERİ

### 2.5.1. Özyüz Yaklaşımı ile Yüz Tanıma

Özyüzler (Eigenfaces) yaklaşımı ve PCA ilk olarak 1987 yılında Sirovich ve Kirby tarafından kullanılarak yüzü etkin bir şekilde göstermek için kullanılmıştır. Temel bilesen analizi olarak da bilinen Karhunen- Louve dönüşümüne dayanmaktadır. Bu, bilgi teorisinde veriyi kodlama ve kodunu çözmede iyi bilinen bir tekniktir. Bu kişiler yüz resimleri gruplarından başlayarak bu resimlerin temel bileşenlerini hesaplamışlardır. Daha sonra da öz vektörün sadece küçük parçalarının ağırlıklı birleşimini kullanılarak yüz resmini yeniden oluşturmuşlardır [25].

Turk ve Pentland, özyüzler yöntemini ilk kez tam otomatik bir siteme dönüştürerek özyüzlerin hesaplanması ve bir yüz resminin özyüzler kullanılarak sınıflandırılması hakkında detaylı bir çalışma sunmuşlardır [26].

Bilgi teorisi dilinde, bir yüz görüntüsünün ilgili bilgisi çıkarılır, mümkün olduğunca verimli bir şekilde kodlanır ve sonra benzer şekilde kodlanan bir model, veri tabanı ile karşılaştırılır [27]. Bir yüz görüntüsü içerisinde yer alan basit bir yaklaşım, yüz görüntüleri grubundaki değişikliği, herhangi bir özellik yargısından bağımsız bir şekilde yakalamak ve bu bilgiyi ayrı ayrı yüz görüntülerini kodlamak ve karşılaştırmak üzere kullanmaktır [27].

Matematiksel anlamda, yüzlerin dağıtımının ana bileşenleri veya yüz görüntüleri kümesinin kovaryans matrisinin öz vektörleri, çok yüksek boyutlu bir uzayda görüntüyü nokta (veya vektör) olarak ele alacak şekilde aranır.

Öz vektörler oluşturulur ve bunların her biri yüz görüntüleri arasındaki değişiklik miktarını gösterir. Bu öz vektörler, yüz görüntüleri arasında varyasyonu karakterize eden bir özellikler seti olarak tanımlanır. Her görüntü konumu aşağı yukarı her öz vektöre katkıda bulunarak, bu özvektörlerin "özyüz" adı verilen bir tür hayalet yüz şeklinde görüntülenmesini sağlar. Şekil 2.4 özyüzleri göstermektedir.

Şekil 2.4. Özyüzler (Eigenfaces) [28].

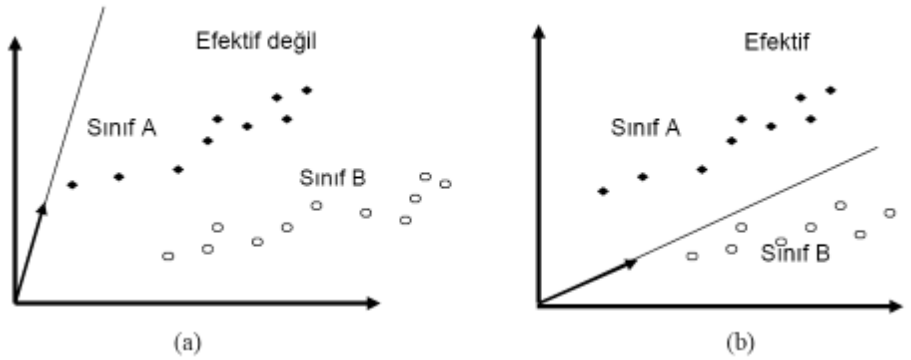
Özyüz yaklaşımı, yüksek boyutlu verilerin PCA kullanılarak daha düşük boyutta simgelenebilmesi için, iki boyutlu matrislere uyarlanmış özel bir uygulamadır. PCA'da, veri tabanındaki sinyallerin kovaryans matrisinin özdeğer ve özvektörlerinin bulunması, yüksek özdeğerlere karşılık gelen özvektörlerin yeni veritabanında saklanması esastır. Bu sayede düşük özdeğere karşılık gelen özvektörler ihmal edilerek veritabanında boyut azaltımı sağlanmaktadır [28].

*2.5.1.1. Özyüz Algoritması (Temel Bilesenler Analizi)*

PCA boyut azaltmak ve görüntüler arasındaki temel farklılıklar çıkarmak için kullanılan bir yöntemdir [26]. PCA karşılıklı ilişkisi olan verinin N-boyutlu uzay yerine daha düşük boyutlu uzayda önemli bilgi kaybı olmadan boyut azaltmanın en uygun yoludur [26].

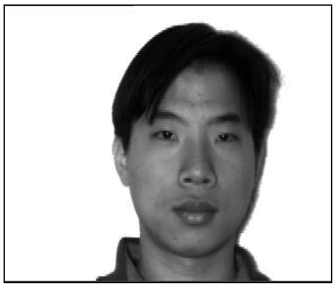
PCA yöntemiyle yüz resimlerinden elde edilen kovaryans matrisinin öz vektörleri elde edilerek yüz görüntülerinin yüz uzayı olarak adlandırılan alt uzayı elde edilir. Bu yöntemle elde edilen vektörlerin görünüm olarak yüze benzemesi ve yüz resimlerini işaret etmesi sebebiyle özyüzler olarak isimlendirilmiştir [29].

PCA, tanımanın doğru yapılmasını sağlayan doğrultuların bulunmasını sağlar. Şekil 2.5.b'de [27] bu sınıflandırma en iyi şekilde ifade edilmektedir.




**Şekil 2.5.** TBA sınıflandırma doğrultusu [27].

Özyüz yaklaşımında, her biri N x N boyutunda olan görüntüler $N^{2}$'lik vektörler haline getirilmektedir.



$$= \begin{bmatrix} I1_{11} \ldots\ldots\ldots\ldots I1_{1N} \\ I1_{21} \ldots\ldots\ldots\ldots I1_{2N} \\ . \quad\quad\quad\quad\quad\quad . \\ . \quad\quad\quad\quad\quad\quad . \\ . \quad\quad\quad\quad\quad\quad . \\ . \quad\quad\quad\quad\quad\quad . \\ . \quad\quad\quad\quad\quad\quad . \\ I1_{N1} \ldots\ldots\ldots\ldots I1_{NN} \end{bmatrix} \qquad I1= \begin{bmatrix} I1_{11} \\ . \\ I1_{1N} \\ I1_{21} \\ . \\ I1_{2N} \\ . \\ I1_{N1} \\ . \\ I1_{NN} \end{bmatrix}$$

**Şekil 2.6.** NxN görüntü örneği ve dönüştürülmüş durumu [28].

Şekil 2.6' da I yüz veritabanındaki görüntüleri N ise piksel sayısını göstermektedir.

Daha sonra veritabanındaki tüm görüntülerden ortalama görüntü vektörü bulunur (Eş. 2.1).

$$\Psi = \frac{1}{M} \sum_{n=1}^{M} \Gamma_n \tag{2.1}$$

Burada Γ: görüntü vektörü, Ψ: ortalama görüntü vektörü, M: toplam görüntü sayısını gösterir.

$$\frac{1}{M}=\begin{bmatrix} I1_1+\ldots\ldots\ldots\ IM_1 \\ I1_{21}+\ldots\ldots\ldots+I1_{2N} \\ . \quad . \\ . \quad . \\ . \quad . \\ . \quad . \\ . \quad . \\ I1_{N1}+\ldots\ldots\ldots+I1_{NN} \end{bmatrix}$$

**Şekil 2.7.** Ortalama görüntünün hesaplanması [28].

Şekil 2.7'de I yüz veritabanındaki görüntüleri, N piksel sayısını ve M toplam görüntü sayısını göstermektedir. Kolon vektörü haline getirilen görüntülere ait pikseller toplanıp toplam görüntü sayısına bölünür, elde edilen kolon matris NxN'lik matrise dönüştürülür [28].

Ortalama görüntü vektörünün, her bir görüntü vektöründen çıkarılması ile sıfır ortalamaya sahip bir veri tabanı matrisi (A) elde edilmiş olur (Eş. 2.2, Eş. 2.3).

$$\Phi_i=\Gamma_i-\Psi \tag{2.2}$$

Es. 5.2'de Φ: ortalama görüntüden farkı alınmış görüntü vektörü, i=1, 2, ..., M ise görüntü sayısını göstermektedir.

$$A = [\Phi\ \Phi\ \Phi\ \ldots\ \Phi\ ] \tag{2.3}$$

Eş. 5.3'te A: $N^2$ x M boyutundaki görüntü veritabanı matrisi'ni ifade eder.

Oluşturulan görüntü veri tabanından özyüzlerin elde edilebilmesi için A matrisine kovaryans işlemi uygulanır. (Eş. 2.4).

$$C = \frac{1}{M}\sum_{n=1}^{M}\Phi_n\Phi_n^T = AA^T \tag{2.4}$$

C: kovaryans matrisini ifade etmektedir.

Eş. 2.4 ile elde edilen kovaryans matrisinin boyutu, görüntü vektörünün uzunluğu ile ilişkilidir. $N^2$'lik görüntü vektörlerinden elde edilecek olan kovaryans matrisi $N^2$ x $N^2$ boyutundadır. Resim vektör boyutunun, toplam görüntü sayısından fazla olduğu durumlarda ($N^2$>>M), bu büyüklükteki bir matrisin elde edilmesi, özdeğer ve özvektörlerinin oluşturulması oldukça zordur. $N^2$ x $N^2$'lik kovaryans matrisinden elde

edilecek olan $N^2$ adet özvektörden sadece M-1 tanesi anlamlıdır. Anlamlı olan özvektör sayısı veritabanındaki görüntü sayısı ile ilişkilidir. Oluşturulan diğer özvektörler, sıfır değerine sahip özdeğerlere karşılık gelmektedir. Bu nedenle, $N^2$ x $N^2$ boyutundaki bir matrisin kullanılması yerine M x M boyutundaki matrisin elde edilmesi ve işlenmesi daha doğru olacaktır.

Kovaryans matrisinin daha düşük boyutta işlem görmesi için Eş. 2.4 yerine, Eş. 2.5 kullanılabilir.

$$C = \frac{1}{M} \sum_{n=1}^{M} \Phi_n^T \Phi_n = A^T A \tag{2.5}$$

Eş. 2.5 ile elde edilen kovaryans matrisin özdeğer ve özvektörlerinden, Eş. 2.4'ün kovaryans matrisinin özdeğer ve özvektörlerine basit bir matris uzayı değişimi ile ulaşılabilmektedir. Eş. 2.5 ile elde edilen kovaryans matrisinin özvektörleri $V_i$ olarak gösterilerek,

$$A^T A V_i = \mu_i V_i \tag{2.6}$$

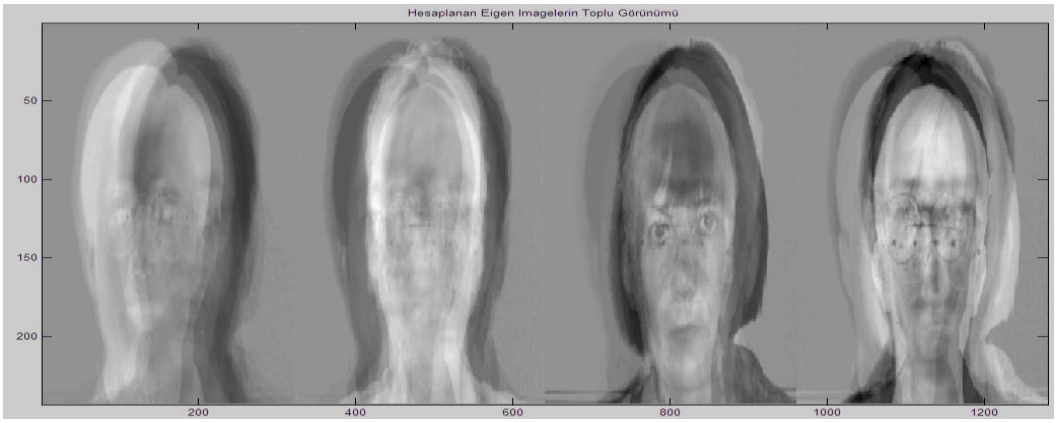
eşitliğine ulaşılmaktadır. Burada vi: özvektörler, μi: özdeğerler'i ifade etmektedir.

Eşitliğin her iki tarafı da A ile çarpılarak,

$$A A^T A V_i = \mu_i A V_i \tag{2.7}$$

$AA^T$ matrisinin özvektörleri $AV_i$ olarak bulunur.

Eş. 2.5 ile elde edilen özvektörler A matrisi ile çarpılarak istenen özvektörlere ulaşılmıştır. Bu sayede işlem yükü azaltılmıştır. Ortaya çıkan M tane özvektörden M' kadar olanı, uzay enerjisini en iyi görüntüleyecek şekilde, özyüz değerlerine göre saklanmaktadır. Şekil 2.8'de elde edilen, en büyük özdeğere sahip ilk dört özyüz örnek olarak verilmektedir [28].




**Şekil 2.8.** Elde edilen en büyük özdeğerlere sahip özyüzler [28].

Oluşturulan özvektörler, özyüz; özyüzlerin oluşturduğu matris ise görüntü uzayı olarak tanımlanmaktadır. Tanımanın gerçekleşmesi için, her bir görüntünün, görüntü uzayı üzerindeki izdüşümünün bulunması ve ağırlık vektörünün oluşturulması gerekmektedir (Eş. 2.8).

$$\omega_{k=} V_k^T \text{ A} \tag{2.8}$$

Eş. 2.8'de ω: ağırlık vektörü (1 x M), k = 1, 2, ..., M' göstermektedir.

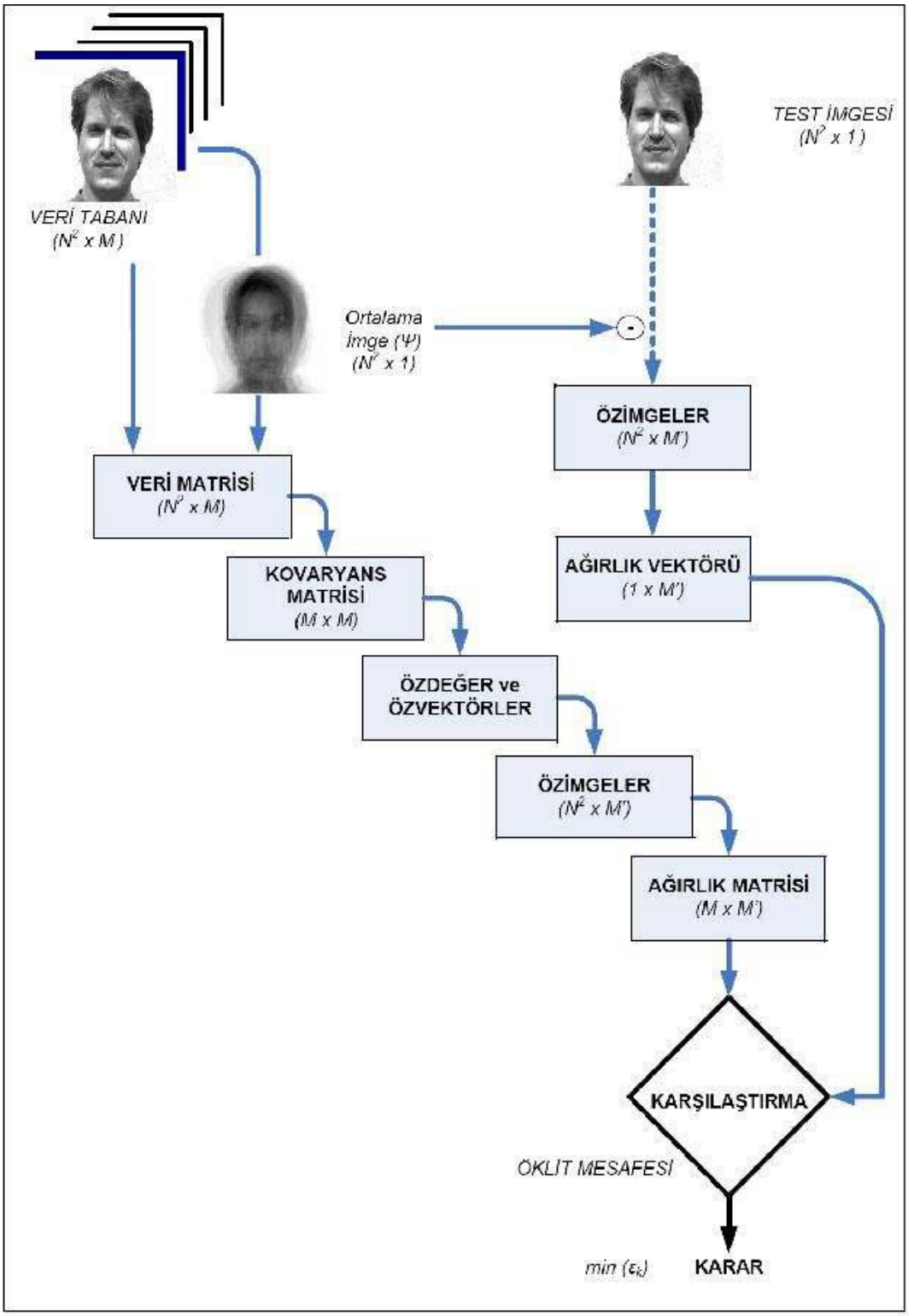
Her bir görüntüye ait ağırlık vektörleri, veri tabanında Ω matrisinde saklanmaktadır (Eş. 2.9). Ω matrisi, ağırlık matrisidir.

$$\Omega=[\omega_1^T\, \omega_2^T \ \ldots \ \omega_M^T] \tag{2.9}$$

Eş. 2.9'da Ω: ağırlık matrisi (M x M')' ni göstermektedir.

Tanıma işlemi, görüntü uzayına düşürülen her bir görüntünün, ağırlık matrisinden ne kadar uzak olduğu ölçütüne göre gerçekleştirilmektedir. Yakınlık ölçütü, öklit mesafesi (Euclidian distance) ile ifade edilmektedir (Eş. 2.10).

$$\varepsilon_k^2 = ||(\Omega - \Omega_k)||^2 \tag{2.10}$$



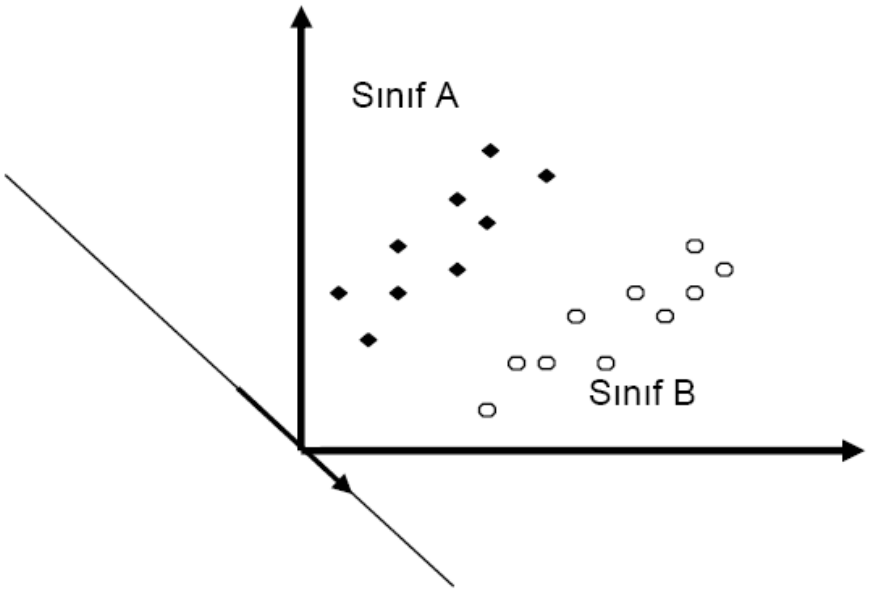

**Şekil 2.9.** Özyüz algoritması ile sınıflandırma işlevsel blok şeması [28].

Ağırlık matrisine göre öklit mesafesi elde edilen test görüntüsünün, en küçük öklit değerine karşılık gelen görüntü olduğu kabul edilerek tanıma gerçekleşmektedir. Özyüz yaklaşımına ilişkin akış şeması Şekil 2.9'da verilmektedir [28].

### 2.5.2. Fisher Yüz Yaklaşımı İle Yüz Tanıma

Fisher yüz algoritması 1997 yılında P. Belhumeur tarafından Doğrusal Ayırtaç Analizi (LDA) kullanılarak geliştirilmiştir. Fisher yüz algoritması (Fisherface approach), sınıflandırma yapan bir yöntemdir. Parmak izi, yüz gibi iki boyutlu görüntüleri tanımlamada kullanılmak üzere, LDA'nın iki boyutlu verilere uyarlanması ile elde edilmiştir. Aynı kişiye ait farklı görüntülerin veri tabanına tanıtılarak sınıf içi dağılımın en aza, sınıflar arası dağılımın ise en fazlaya ulaşması hedeflenmiştir. Bu sayede, Fisher yaklaşımında tanımlama başarım oranında artış gözlenmiştir.

LDA veriler arasındaki ayrımı sağlayan yönün bulunmasını sağlar (Şekil 2.10).




**Şekil 2.10.** LDA için sınıfları ayıran yön [28].

Fisher yaklaşımında, her biri N x N boyutunda olan görüntüler $N^2$'lik vektörler haline getirilerek veritabanındaki tüm görüntülerden ortalama görüntü vektörü elde edilir (Eş. 2.11).

$$\Psi = \frac{1}{M_t} \sum_{k=1}^{M_t} \Gamma_k \tag{2.11}$$

Eş. 2.11'de Γ: görüntü vektörünü, Ψ: ortalama görüntü vektörünü, $M_t$: toplam görüntü sayısını ifade eder.

Veri tabanında, **c** farklı sınıfa ait toplam $M_t$ adet görüntü vektörü kaydedilmektedir. Ortalama görüntü vektörü hem tüm veri tabanı içerisinde hem de her bir sınıf için ayrı

ayrı hesaplanmaktadır. Sınıf içi ortalama görüntü vektörü, $\Psi_{Ci}$ olarak gösterilmektedir.

$$\Phi_i=\Gamma_i-\Psi_{Ci} \tag{2.12}$$

Eş. 2.12'de Φ: ortalama görüntüden farkı alınmış sınıf içi görüntü vektörünü gösterir. i = 1, 2, ..., c değerlerini alır.

Oluşturulan vektörlerden sınıf içi ve sınıflar arası saçılma matrisleri oluşturulmaktadır. Sınıf içi saçılma matrisi aşağıda gösterilmiştir;

$$S_W = \sum_{i=1}^{c} P(C_i)\Sigma_i \tag{2.13}$$

Eş. 2.13'da $S_w$: sınıf içi saçılma matrisi, $P(C_i)$: önsel sınıf olasılığı'dır. Burada,

$$P(C_i) = \frac{1}{C} \tag{2.14}$$

ve

$$\Sigma_i = E\left[\Phi_i \Phi_i^T\right] = E[(\Gamma_i - \Psi_{ci})(\Gamma_i - \Psi_{ci})^T] \tag{2.15}$$

formülleri ile gösterilir.

Sınıflar arası saçılma matrisi Eş. 2.16' da gösterilmiştir.

$$S_b = \sum_{i=1}^{c} P(C_i)(\Psi_{ci} - \Psi)(\Psi_{ci} - \Psi)^T \tag{2.16}$$

Eş. 2.13 ve Eş. 2.16 'da saçılma matrisleri Eş. 2.17'de kullanılarak en iyi W matrisi bulunması hedeflenmektedir. Bu sayede, sınıflar arası saçılma en fazlaya, sınıf içi saçılma en aza düşürülmektedir.

$$W = \underset{T}{\operatorname{argmax}}\left(J(T)\right) \Rightarrow \max\left(J(T)\right) = \frac{|T^T S_b T|}{|T^T S_w T|} I_T = W \tag{2.17}$$

W matrisinin çözümü genelleştirilmiş özdeğer çözümü ile yapılmaktadır (Eş. 2.18).

$$S_b W = S_w W \lambda_w \tag{2.18}$$

Tanımlama için kullanılacak olan W matrisi, Sb ve Sw matrisleri kullanılarak oluşturulan özvektör matrisidir. Özvektör matrisi, anlamlı özdeğerlere göre sıralandığında en fazla c-1 adet özyüzden meydana gelmektedir. Bu değer, veri tabanında yer alan birbirinden farklı sınıf sayısı ile ilişkilidir. Özyüz yaklaşımına benzer

şekilde, her bir görüntünün özvektörler ya da görüntü uzayına yansıtılması ile o görüntüye ait ağırlık vektörü bulunmuştur (Eş. 2.19).

$$g(\Phi_i) = W^T \Phi_i \quad (2.19)$$

Eş. 2.19'de g: ağırlık vektörünü göstermektedir.

Her bir görüntüye ait ağırlık vektörü, tanımlama için kullanılmak üzere veri tabanında saklanmaktadır. Test amaçlı olarak kullanılacak olan görüntünün, ortalama görüntüden farkı alınarak ve görüntü uzayına yansıtılarak ağırlık vektörü bulunmaktadır (Eş. 2.20, Eş. 2.21).

$$\Phi = \Gamma_T - \Psi \quad (2.20)$$

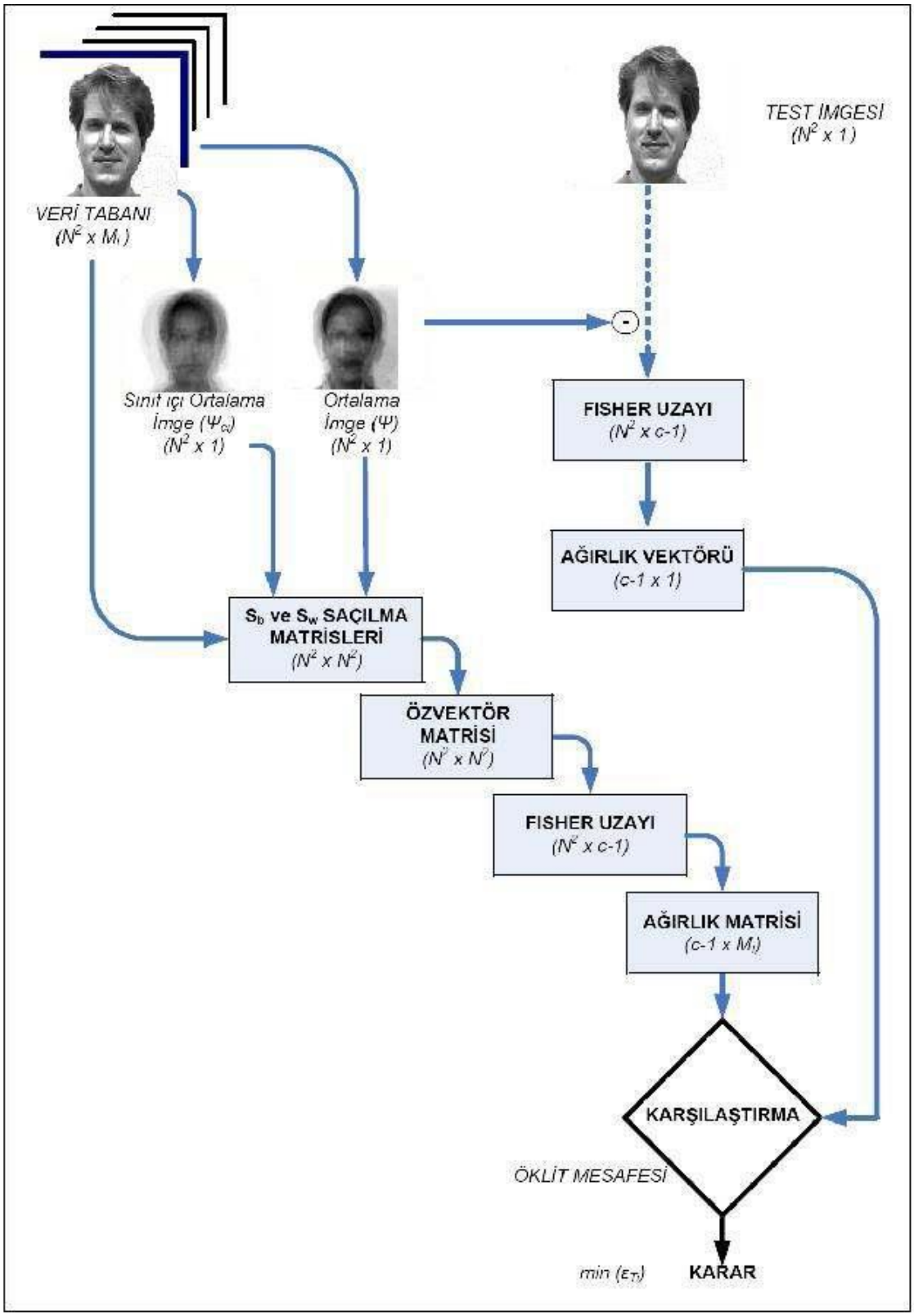
Eş. 2.20'de $\Gamma_T$: test görüntüsünü, $\Psi$: ortalama görüntü vektörünü göstermektedir.

$$g(\Phi_T) = W^T \Phi_T \quad (2.21)$$

Tanımlama, görüntü uzayına düşürülen görüntünün, ağırlık matrisinden ne kadar uzak olduğu ölçütüne göre yapılmaktadır. Yakınlık ölçütü öklit mesafesi ile tanımlanmaktadır (Eş. 2.22).

$$\varepsilon_{Ti}^2 = \|(g(\Phi_T) - g(\Phi_i))\|^2 \quad (2.22)$$

Ağırlık matrisine göre öklit mesafesi elde edilen test görüntüsünün tanımlanması, en küçük öklit uzaklığına karşılık gelen görüntü olduğu kabul edilerek yapılmaktadır. Fisher yaklaşımına ilişkin akış şeması Şekil 2.11'de gösterilmektedir [28].



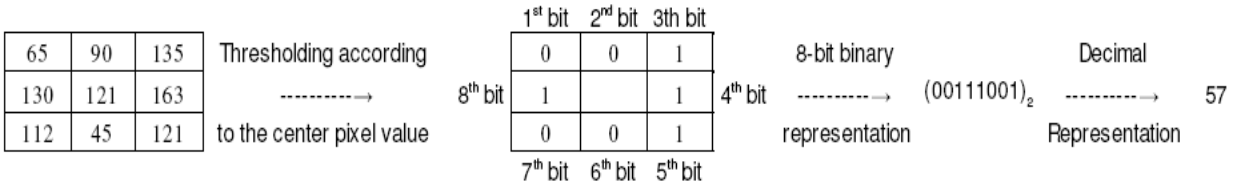

**Şekil 2.11.** Fisher yüz algoritması ile sınıflandırma işlevsel blok seması [28].

### 2.5.3. Yerel İkili Örüntü Yöntemi İle Yüz Tanıma

Bilgisayarda görüntü doku analizi, görüntü işlemede önemli bir yere sahiptir. Yapılan çalışmalar neticesinde teorik ve hesaplama açısından basit; fakat doku analizinde çok etkili olan Yerel İkili Örüntü (LBP) yöntemi geliştirilmiştir.

Yukarıda anlatıldığı üzere LBP yaklaşımı özellikle doku analizlerinde çok başarılıdır. Bir diğer önemli özelliği ise; gerçek yaşamda yapılan uygulamalarda ışık şiddetinde oluşabilecek olan değişmelere karşı etkili olmasıdır. Ayrıca hesaplama açısından da kolaylık sağlar [30].

LBP operatörü bir imgedeki her pikseli, kendisini çevreleyen 3x3'lük komşuluk bölgesinde bulunan komşu piksellerini merkez piksel değerine göre eşikleyerek etiketler. Eğer komsu pikselin değeri merkez piksel değerinden büyükse veya eşitse komşu piksel 1, küçükse 0 değerini alır. Bu şekilde bir komşuluk bölgesi için 8-bitlik bir LBP kodu yapılmış olur. Bu kodun onluk sistemdeki değeri ise merkez pikseli çevreleyen 3x3'lük bölgedeki yerel yapıyı gösterir (Şekil 2.12).




**Şekil 2.12.** LBP değerinin hesaplanmasının grafiksel ifadesi [31].

Şekil 2.12'de eşikleme işleminden sonra sol üst kösedeki ikili sayı ilk basamak kabul edilerek saat yönü sırasıyla 8-bitlik LBP kodu oluşturulmaktadır. Daha sonra bu kodun onlu sistemdeki değeri hesaplanarak merkez pikselimizin LBP değeri bulunmaktadır.

LBP'nün matematiksel ifadesi ;

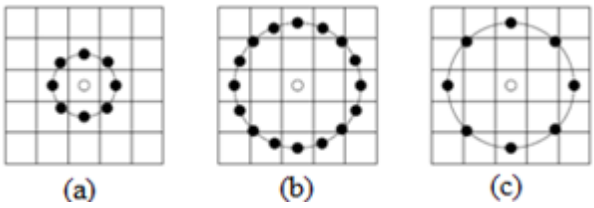
$$LBP(X) = \sum_{i=1}^{8} s(G(X_i) - G(x))2^{i-1} \quad (2.23)$$

$$S(t) = \begin{cases} 1, t \geq 0 \\ 0, t < 0 \end{cases} \quad (2.24)$$

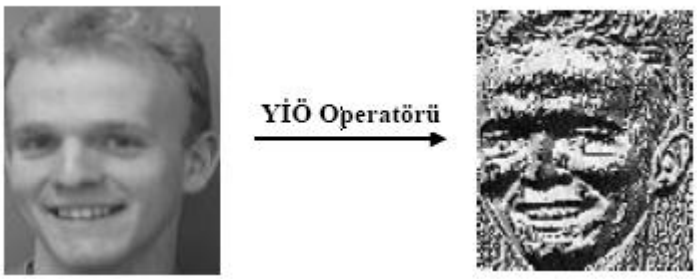
burada **x** merkez pikselin konumu, $x_i$ i indisli komşu pikselin konumunu ve G(.) piksel yoğunluk değerini ifade etmektedir.

LBP'nün farklı gösterimleri de vardır. Örneğin; 2 piksel uzaklık komşuluklu 8-bit ikili gösterim, 2 piksel uzaklık komşuluklu 16-bit ikili gösterim ve 3 piksel uzaklık komşuluklu 24-bit ikili gösterimi gibi. 2 piksel uzaklık komşuluklu 8-bit ikili gösteriminde 2 piksel uzaklıktaki gri seviye değerleri tam olarak bir piksel konumunun merkezine düşmezse yoğunluk değeri ara değerleme kullanılarak tahmin edilir [31].

Oluşturulan bütün LBP kodları kullanılması mümkün değildir. Bunun yerine (0) dan (1)'e değişim sayısı iki veya daha az olanlar tercih edilir. Bu şablonlar tekbiçim (uniform) olarak isimlendirilir. Örneğin 11110011 veya 000111111 tekbiçim şablonlardır. Fakat 01100110 veya 10100110 tekbiçim olmayan şablonlardır. Tekbiçim Yerel İkili Örüntü $Y\dot{I}\ddot{O}^{u}_{P,R}$ ifadesi ile belirtilir. Burada $P$ nokta sayısı, $R$ yarıçapı ifade etmektedir. Şekil 2.13'de farklı *(P,R)* çiftleri için oluşturulmuş LBP şekilleri gösterilmektedir. Şekil 2.14'de de LBP operatörünün yüz imgesine uygulanmasıyla oluşturulan imge gösterilmektedir.

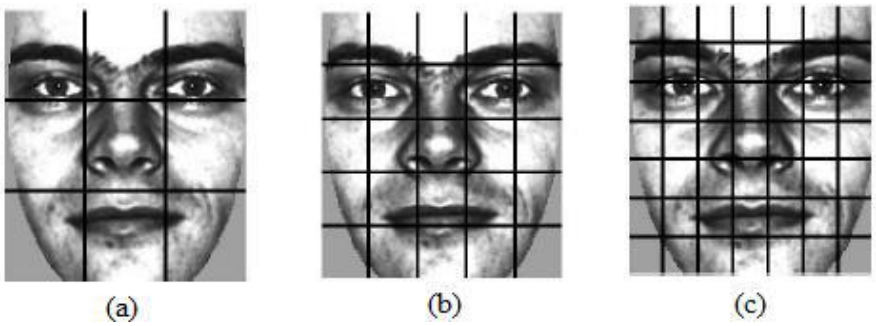



**Şekil 2.13.** (a) (8,1), (b) (16,2), (c) (8,2) göre oluşturulmuş LBP [30].

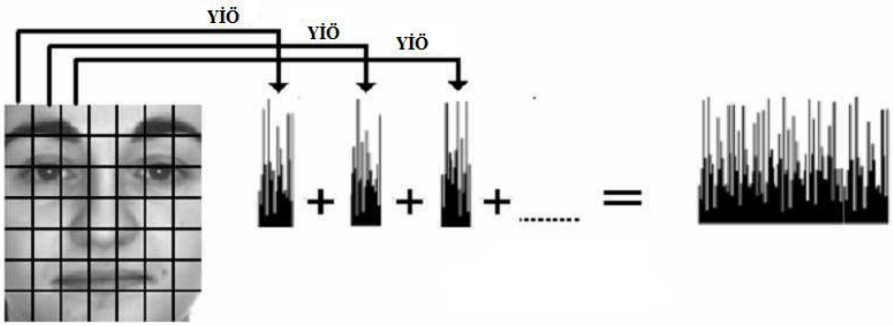



**Şekil 2.14.** Orijinal imge (Solda) ve LBP operatörü ile oluşturulmuş imge (Sağda) [30].

İmgeler belirli sayıda bölgeye ayrıldıktan sonra her bölge için $Y\dot{I}\ddot{O}^{u}_{P,R}$ kodları ile histogramlar elde edilir (Şekil 2.15) sonrasında bu histogramlar birbirine eklenir (Şekil 2.16). Oluşturulan bu yeni histogram özellik vektörü olarak kullanılır.




**Şekil 2.15.** (a) 3×3 (b) 5×5 (c) 7×7 Bölgeye ayrılmış imge [30].




**Şekil 2.16.** Histogramların eklenmesi [30].

## 2.6. AYDINLATMA ETKİSİNİN AZALTILMASI

### 2.6.1. Aydınlatma Normalleştirme Metodları

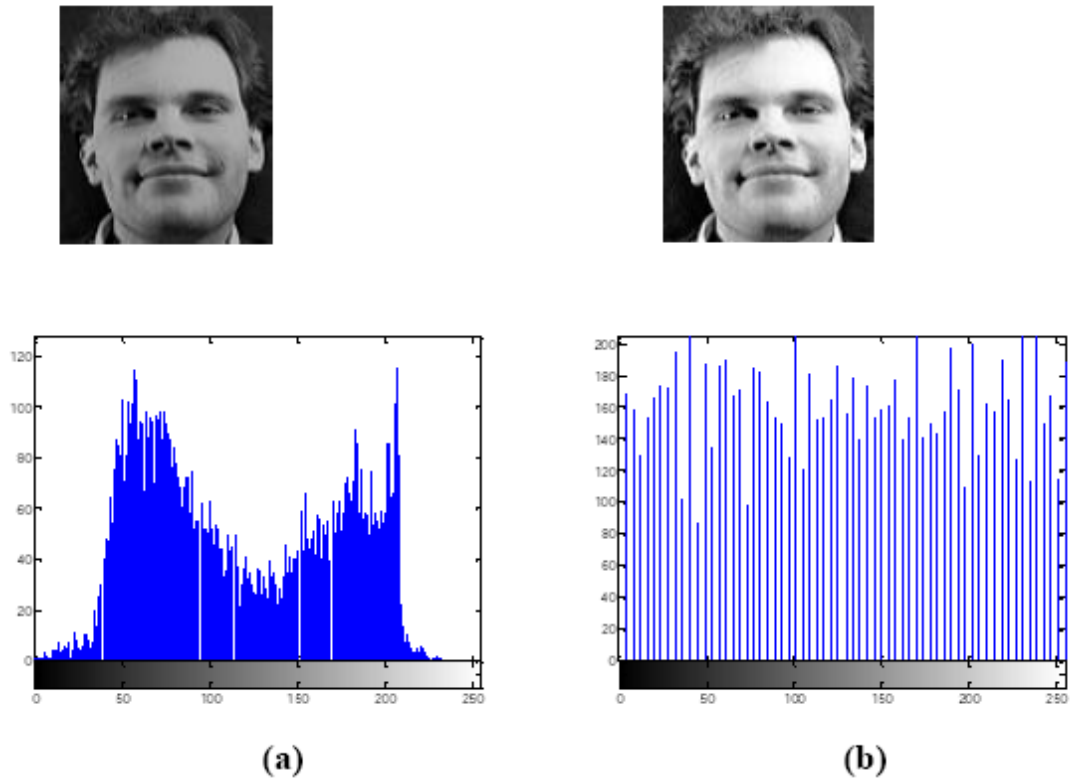
Aydınlatma değişimleri yüz tanıma sistemi için sorun teşkil eden konular arasındadır. Shan vd., üç farklı yöntem kullanarak yüz tanımada farklı ışık koşulları altında çekilmiş görüntülerin aydınlatma normalleştirmesini yapmışlardır [32]. Histogram eşitleme ve tanımlama, Gamma yoğunluk doğrulaması ve Tekrar Bölüm Aydınlatması (Quotient Illumination Relighting, QIR) yöntemlerini kullanmışlardır.

Histogram Eşitleme, yerel Histogram Eşitleme ve uyarlanır Histogram Eşitleme yöntemleri aydınlatma etkisini azaltmak için kullanılan yöntemlerdir. Bu yöntemler aydınlatma normalleştirme yöntemleri olarak da bilinmektedir.

#### *2.6.1.1. Histogram Eşitleme*

Histogram, sayısal bir görüntüde her renk değerinden kaç tane olduğunu gösteren bir grafiktir. Bu grafik görüntünün parlaklık durumu ya da tonları hakkında bilgi verir.

Diğer bir ifadeyle histogram, görüntü üzerindeki piksel değerlerinin grafiksel gösterimidir. Histogram, görüntü histogramı veya gri-seviye histogramı olarak da isimlendirilmektedir. Görüntü histogramı, görüntünün her bir noktasındaki pikselin seviyesini ve bu seviyedeki piksellerin sayısının ne olduğunu gösterir. Bu sayede histogram üzerinden görüntü ile ilgili olarak gri seviye dağılımı, görüntünün parlaklık ve kontrast durumu ile kontrast iyileştirme analizi gibi çeşitli bilgilerin çıkartılması sağlanır (Şekil 2.17) [33].




**Şekil 2.17.** Görüntünün histogramları **a.** Histogram eşitlemeden önceki hali (sol sütun), **b.** Histogram eşitleme sonrası (sağ sütun) [33].

Histogram eşitleme, renk değerleri düzgün dağılımlı olmayan görüntüler için elverişi bir görüntü eşitleme yöntemidir. Görüntünün tamamında kullanılabileceği gibi sadece belli bir bölgesine de uygulanabilir. Tüm görüntüye uygulanırsa global histogram eşitleme, görüntü belli bölgesine uygulandığında ise yerel histogram eşitleme adını alır. Histogram eşitleme tekniği genelde bir görüntünün kontrastının iyileştirmesinin etkisini araştırmak ve görmek için yapılır. Kontrast iyileştirmenin mantığı, işlenen görüntüdeki gri ton değerlerinin dinamik aralığını artırmaktır. Düşük kontrastlı görüntülere zayıf aydınlatma koşulları neden olmuş olabilir.

Matematiksel olarak, bir sayısal görüntü histogramı Eşitlik 2.25'teki gibi tanımlanmaktadır.

$$P(r_k) = \frac{n_k}{n_p} \tag{2.25}$$

Burada;

$r_k$ : k'ıncı gri seviyesini ifade eder.

$n_k$ : Gri seviyeye sahip toplam piksel adedini gösterir.

$n_p$ : Görüntü üzerindeki toplam piksel adedini gösterir.

Dönüştürülen ve orijinal histogramlar olasılık yoğunluk fonksiyonları ile ifade edilebilirler. Aşağıdaki histogram işleme tekniğinde, T (r) dönüşüm fonksiyonunda tanımlı olan olasılık yoğunluk fonksiyonu kullanılarak resmin görüntüsünde iyileştirme sağlanmaktadır (Eş. 2.26).

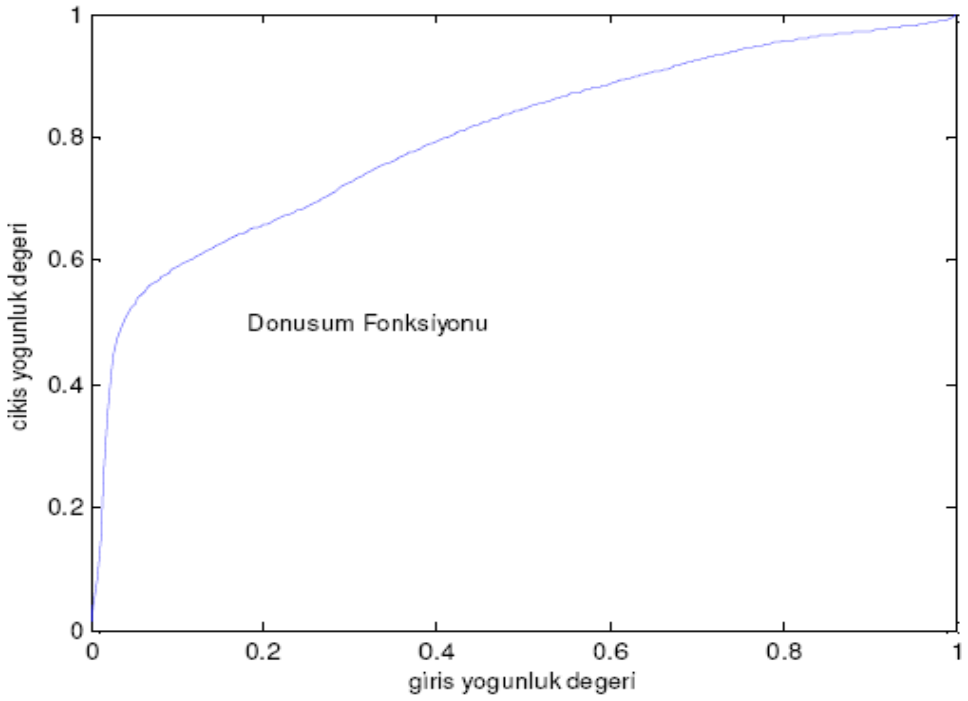
$$s = T(r) = 255\ x \sum_{w=0}^{r} h_i(w)dw = 255 \text{ x } CDF(r)\ r \in [0,255] \tag{2.26}$$

T(r), dönüşüm fonksiyonu (Şekil 2.18).

CDF(r), kümülatif dağılım fonksiyonu.

$h_i$ , olasılık fonksiyonu.

*i*, resimdeki işlenen pikselin gri ton değerleri [33].




**Şekil 2.18.** Histogram dönüşüm fonksiyonu [33].

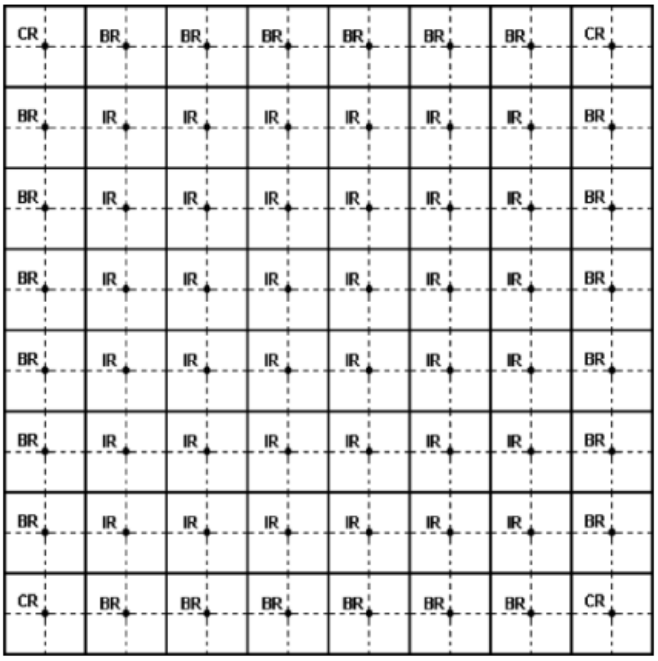
*2.6.1.2. Yerel Histogram Eşitleme*

Histogram eşitleme yerel bir bölgeye uygulandığında yerel histogram eşitleme adını almaktadır [33].

*2.6.1.3. Kontrast Sınırlı Adaptif Histogram Eşitleme*

Histogram eşitleme ile görüntünün yoğunluk dağılımı normalize edilerek düzgün bir yoğunluk dağılımına sahip görüntü elde edilir ve böylece görüntü kalitesi iyileştirilebilir. Fakat histogram eşitlemede tüm görüntünün yoğunluk dağılımı işleme konduğundan bu bazı görüntülerde bozucu etkiye neden olabilmektedir. Örneğin; ortalama yoğunluk orta seviyeye getirildiğinde solmuş etkiye sebep olabilir ve dar bir bölgede kalabalık bir yoğunluk dağılımına sahip görüntülerde ise bir çok gürültü pikselinin oluşmasına neden olabilir.

Adaptif histogram eşitleme değiştirilmiş bir histogram eşitleme işlemidir ve yerel veri üzerinde iyileştirme yapar. Buradaki amaç, görüntüyü ızgara şeklinde dikdörtgen bölgelere ayırarak her bir bölgeye standart histogram eşitleme işlemi uygulamaktır (Şekil 2.19).

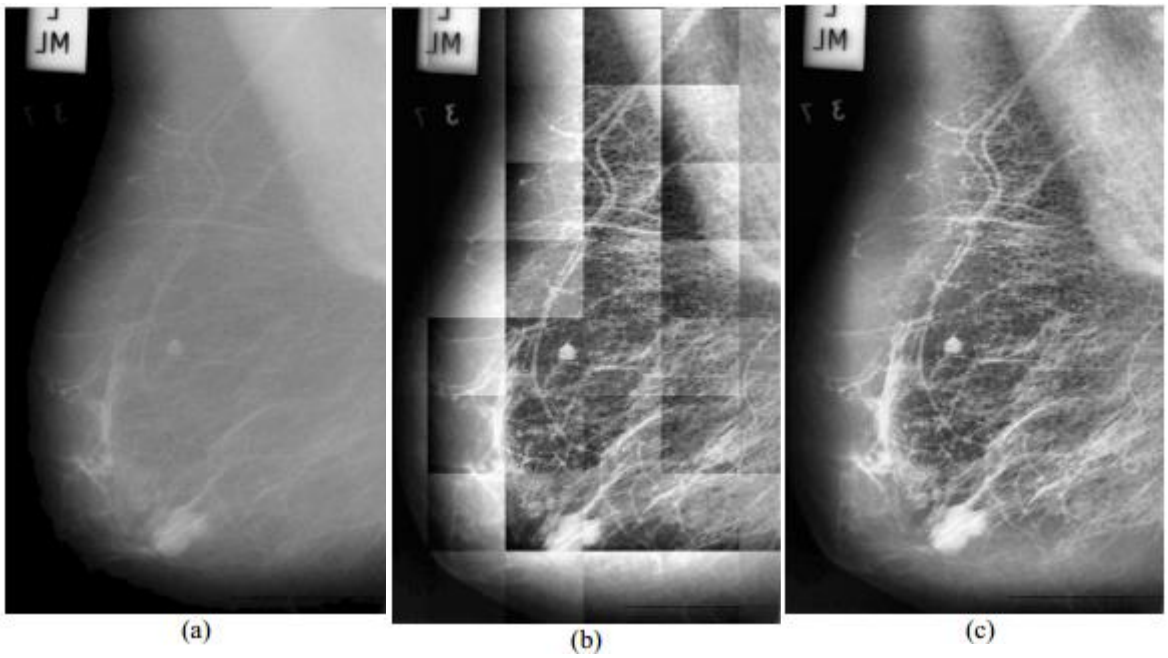



**Şekil 2.19.** 512x512'lik bir görüntünün 64 eşit kare bölgeleye ayrılmış yapısı [34].

Optimum bölge boyutları ve sayısı görüntüye göre farklılaşmaktadır. Görüntü alt bölgelere ayrılarak bu bölgelere histogram eşitleme işlemi uygulandıktan sonra alt bölgeler bi-lineer enterpolasyon yöntemi ile birleştirilerek iyileştirilmiş bütün bir

görüntü elde edilir. Ancak adaptif histogram eşitleme yapılan görüntülerde gürültüler oluşmaktadır. Bunu önüne geçmek için homojen bölgelerde kontrast iyileştirmenin sınırlandırılması gerekmektedir, bu nedenle de kontrast sınırlı adaptif histogram eşitleme yöntemi geliştirilmiştir [34].

Bu yaklaşımda ilk olarak her bir bölgenin histogramı ayrı ayrı hesaplanır. Sonra istenen kontrast genişlik sınırına bağlı olarak histogramların kırpılması için kırpma sınır değeri bulunur. Daha sonra her histogram bu kırpma sınır değerini aşmayacak şekilde yeniden dağıtılır. Son olarak, gri seviye haritalama için elde edilen kontrast sınırlı histogramların kümülatif dağılım fonksiyonları hesaplanır. Bu şekilde her bölge için gri seviye haritalama yapılarak sonuç görüntüsü elde edilir (Şekil 2.20)




**Şekil 2.20.** (a) Orjinal, (b) eşikleme +CLAHE uygulanmış (bi-lineer enterpolasyondan önce), (c) (b) görüntüsüne bi-lineer enterpolasyon uygulandıktan sonra elde edilen sonuç görüntüsü [34].

### 2.6.2. Gürültü Azaltma (Filtreleme)

Filtreler görüntü zenginleştirme amacı ile de uygulanan, belirli ayrıntıların temizlenmesi ya da daha belirgin hale getirilmesi vb. gibi işleri gerçekleştiren operatörlerdir. Farklı durumlar için farklı filtreleme operatörleri vardır. Bunlara;

- Kenar keskinleştirme
- Kenar yakalama
- Görüntü yumuşatma

gibi daha bir çok amaçla kullanılan filtreler örnek verilebilir. Görüntüyü meydana getiren pikseller konumları ve gri değerleri ile tanımlanabilmektedir. Daha doğrusu her görüntü matris formuna sahiptir. Bu matrisin üzerinde şablon matris gezdirilerek her piksel değeri yeniden hesaplanır. Filtreler sayesinde girdi resminden yeni resim değişik efektler verilerek elde edilir [35].

Doğrusal ve doğrusal olmayan filtreleme yöntemleri vardır. Doğrusal filtreler konvolüsyon veya Fourier dönüşümleri kullanılarak uygulanır. Gauss, Laplace ve Yüksek artış filtreler gibi çeşitleri bulunur. Doğrusal olmayan filtreler ise, doğrusal filtrelerin yetersiz kaldığı durumlarda etkin olarak kullanılan ve Medyan gibi çeşitleri bulunan filtrelerdir. Alçak geçiren filtrelerin ortalama alarak gürültünün etkisini azaltması nedeniyle, gürültüyü azaltmak için genelde alçak geçiren filtreler kullanılmaktadır. Yüksek geçiren filtreler ise kenar netliğini koruduğu için kenar belirleme ve kenar keskinleştirme amacıyla kullanılmaktadır [33].

*2.6.2.1. Gauss Filtresi*

Gürültü giderici bir filtre; yüksek frekanslı gürültüleri temizleyebilmek için, frekans düzleminde mümkün olduğu kadar dar olmalı, kenarları belirleyebilmek için ise uzay düzleminde mümkün olduğu kadar dar olmalıdır. Her iki özelliği de sahip en iyi filtre Gauss filtresidir. Bundan dolayı gürültü gidermede Gauss filtresi çok tercih edilen ve en iyi doğrusal filtrelerden birisidir. Genel olarak Gauss maskesi alçak geçiren bir filtredir ancak, kenar bulma amacıyla da kullanılmaktadır. Bu filtrede merkez pikselin ağırlıklı komşularınkinden daha fazla olduğu için Düzgün Dağıtılmış Filtredeki olumsuzluklar bu filtrede daha az bulunmaktadır. Eşitlik 2.26 Gauss filtresinin fonksiyonu göstermektedir [33].

$$G(x,y) = \left(\frac{1}{2\pi\sigma^2}\right) + exp\left(\frac{x^2}{2\sigma^2}\right).exp\left(-\frac{y^2}{2\sigma^2}\right) \quad (2.26)$$

G(x,y) : Gauss Fonksiyonu

x : Satır sayısını gösteren eksen

y : Sütun sayısını gösteren eksen

$\sigma$: Standart sapma

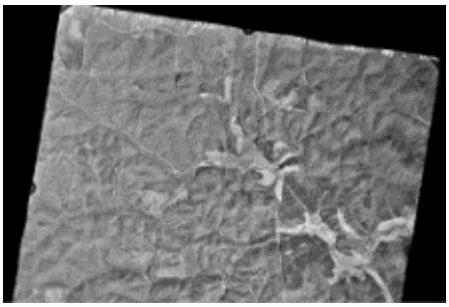
Örnek Gauss filtreleri aşağıda gösterilmektedir [33].

$$h_{Gauss} = \frac{1}{6}\begin{bmatrix} 0 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

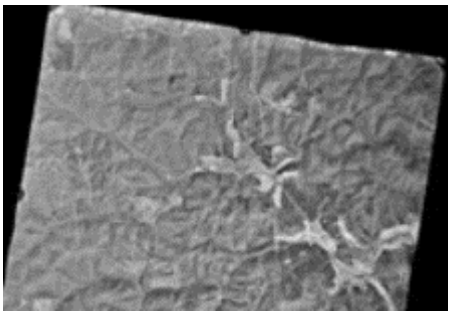
$$h_{Gauss} = \frac{1}{8}\begin{bmatrix} 0 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$h_{Gauss} = \frac{1}{12}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 4 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$h_{Gauss} = \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$



(a)



(b)

**Şekil 2.21.** (a) Orijinal görüntü (b) Gauss filtresi uygulanmış görüntü.

Şekil 2.21'de gauss filtresi uygulanmış görüntü örneği verilmiştir.

*2.6.2.2. Medyan Filtresi*

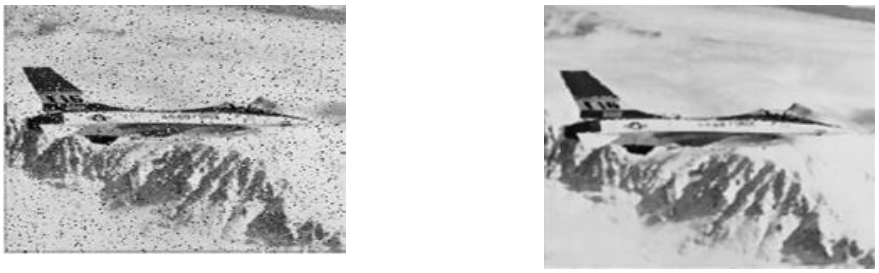
Doğrusal filtreler Gauss tipindeki gürültüyü etkili olarak giderirken ikili gürültüyü yok etmede yetersiz kalmaktadır. Bundan ötürü bu tür gürültüleri gidermek maksadıyla Medyan filtre gibi doğrusal (lineer) olmayan filtreler tercih edilmektedir. Medyan filtre bir pikselin değerini komşusu olan piksellerin ortancasının değeri ile değiştiren filtre

çeşididir. Bu filtrede, konvolüsyonda olduğu gibi bir pencere görüntü üzerinde hareket ettirilir ve penceredeki parlaklık değerlerinin medyanı alınıp bulunan değer bir pikselin parlaklık değeri olarak belirlenir. Başka bir deyişle bir pikselin değerini, o piksele komşu alanlardaki gri seviyelerin medyanı ile yer değiştirir. (pikselin orijinal değeri medyanın hesaplanmasına katılmaktadır).

Medyan filtrelerin kullanımı oldukça yaygındır, bunun sebebi çeşitli gürültüler için mükemmel gürültü azaltma kabiliyeti göstermeleri ve benzer boyuttaki lineer düzeltme filtrelerine göre çok daha az bulanıklığa neden olmalarıdır. Literatürde Medyan filtre histogram tabanlı bir filtre olarak da bilinmektedir. Bu filtre kenarları fazla etkilemeden gürültü temizleyebilme, görüntü içerisindeki ani değişimleri temizleyebilme gibi özelliklere sahiptir ve özellikle tuz biber gürültüsü için çok efektiftir. Doğrusal olmayan bir filtre olduğu için Fourier dönüşümünü kullanmaz.

Medyan filtresi darbe gürültüsünü gidermek için kullanılan lineer olmayan yöntemlerin başında gelmektedir. Tek boyutlu veride giriş verisi üzerinden bir pencere geçirilir. Bu pencerenin medyan değeri bulunarak bulunan medyan değeri pencerenin orta değeri ile değiştirilerek işlem tamamlanır. Bu şekilde verinin herhangi bir yerinde darbe (impulse) niteliğinde bir gürültü varsa bu gürültü kaldırılmış olur.

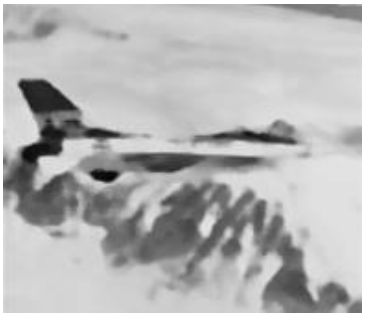
Görüntü işlemede ise medyan filtresinin 2 boyutlu olması gerekir. Medyan filtresinin en pratik yolu kare (3x3, 5x5, 7x7) şeklinde bir pencere alarak görüntünün sol üst köşesinden başlayarak görüntünün tamamını taramak ve gürültüleri gidermektir [33]. Şekil 2.22'de gürültüsüne maruz kalmış bir görüntünün medyan filtresi kullanılarak temizlenmiş hali görülmektedir.



(a) (b)

**Şekil 2.22.** (a) Orijinal Görüntü, (b) 3x3 Medyan Filtre Soncu, (c) 5x5 Medyan Filtre Sonucu, (d) 7x7 Medyan Filtre Sonucu.

(c)



(d)

**Şekil 2.22 (devam).** (a) Orijinal Görüntü, (b) 3x3 Medyan Filtre Soncu, (c) 5x5 Medyan Filtre Sonucu, (d) 7x7 Medyan Filtre Sonucu.

*2.6.2.3. Laplace Filtresi*

Kenarların belirgin bir şekilde gösterilmesinde başarılı olan bir filtredir. Bir görüntüye uygulanan bu tekniklerin aynı kameradan alınan diğer görüntülerde de başarılı olduğu gözlenmiştir. Bir görüntünün iyileştirmesi için Laplace filtresini kullanırsak; girdi görüntünün f(x,y), Laplace filtresi $\nabla^2 f\,(x,y)$ aşağıdaki şekilde tanımlanır [33].

$$\nabla^2 f\,(x,y) = \frac{\partial^2 f\,(x,y)}{\partial x^2} + \frac{\partial^2 f\,(x,y)}{\partial y^2} \tag{2.27}$$

Genel olarak ikinci türev yaklaşımı kullanılır.

$$\frac{\partial^2 f}{\partial x^2} = f\,(x+1,y) + f\,(x-1,y) - 2f\,(x,y)$$

ve

$$\frac{\partial^2 f}{\partial y^2} = f\,(x,y+1) + f\,(x,y-1) - 2f\,(x,y)$$

sonuç olarak

$$\nabla^2 f = [f\,(x+1,y) + f\,(x-1,y) + f\,(x,y+1) + f\,(x,y-1)] - 4f\,(x,y) \tag{2.28}$$

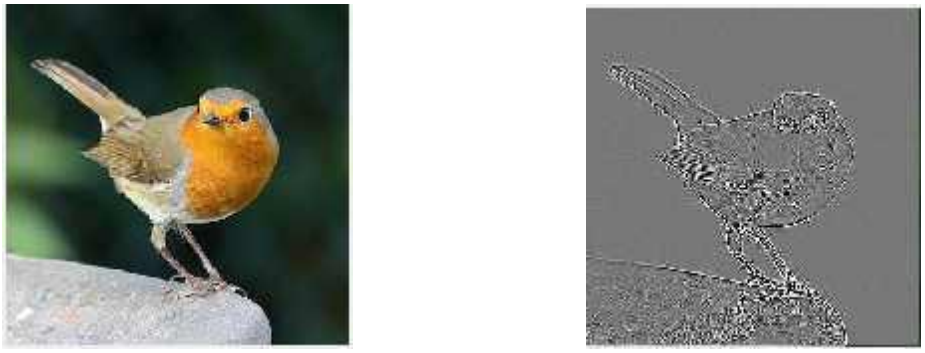
elde edilir.

Laplace filtresi, ikinci dereceden türev özelliğindedir ve bütün yönlerdeki kenarları tespit eder. Eğer maskedeki bütün katsayıların toplamı sıfır ise sadece kenarları bulur, merkezdeki katsayı diğer katsayıların toplamından büyükse hem kenarları bulur hem de orijinal görüntüyü ekler [33].

Laplace operatörü her yönde keskinleştirme yapmaya yarar. Aşağıdaki matrisleri kullanır [35].

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad , \quad \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Şekil 2.23'de Laplace Filtresi uygulanmış görüntü bulunmaktadır [35].



(a) (b)

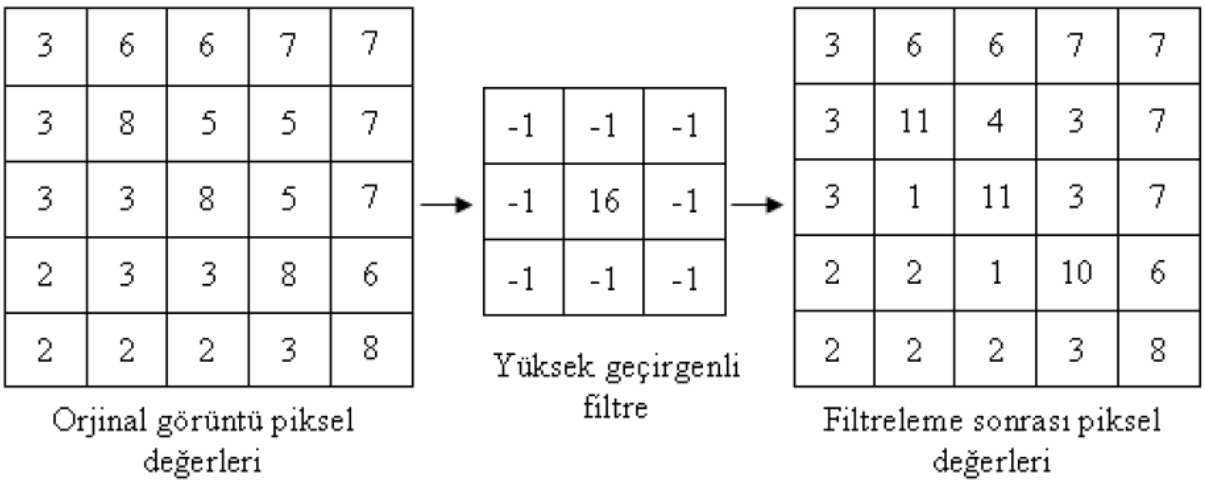
**Şekil 2.23.** (a) Orijinal Görüntü, (b) Laplace Filtresi Uygulanmış Görüntü [35].

*2.6.2.4. Yüksek Artış Filtresi*

Yüksek frekansları vurgulayan ve düşük frekansları bastıran filtrelere yüksek geçirgenli filtreler (high-pass filters) denir. Yüksek geçirgenli filtreler açık ve koyu tonlu detaylar arasındaki farkı daha da belirginleştirir. Benzer olarak orta ve alçak geçirgenli filtreler de vardır. Alçak geçirgenli filtreler (low-pass filters) açık ve koyu detaylar arasındaki ton farkını azaltırlar. Alçak ve yüksek geçirgenli filtreleri uygulamanın en basit yolu uzaysal komşuluk ortalaması ile yapılır. Örneğin bir alçak geçirgenli filtre, orijinal görüntünün her pikselinin çevresindeki piksellerin ortalaması ve bu ortalamanın işlenmiş görüntüde piksel gri renk tonu düzeyi olarak kullanılması ile uygulanmaktadır. Basit bir yüksek geçirgenli filtre ise, orijinal görüntüden alçak geçirgenli filtre ile

filtrelenmiş bir görüntünün çıkarılması ile yada merkezdeki piksel için pozitif, etrafını çevreleyen pikseller için negatif ağırlıklara sahip bir nokta yayılım fonksiyonu kullanılarak döndürülmesi ile oluşturulabilir.

Yüksek geçirgenli bir filtreleme örneği Şekil 2.24'de verilmiştir. 3x3 boyutlu filtre kutusunun ortasına karşılık gelen piksellerin yeni piksel değerleri, kutudaki sayıların eşlenik piksel değerleri ile çarpılıp toplanması ve filtredeki sayıların toplamına bölünmesi ile elde edilir. Yeni görüntüde orijinal görüntüdeki yüksek değerli pikseller daha yüksek düşük değerlikli pikseller daha düşük olarak hesaplanmıştır [36].




**Şekil 2.24.** Yüksek geçirgenli filtre uygulaması [36].

Filtre orijinal görüntünün ortasına yerleştirildiğinde filtreleme sonrası bu pikselin yeni değeri şu şekilde hesaplanır.

[(-1*8)+(-1*5)+(-1*5)+(-1*3)+(16*8)+(-1*5)+(-1*3)+(-1*3)+(-1*8)]/[(-1)+(-1)+(-1)+(-1)+16+(-1)+(-1)+(-1)+(-1)] = (128-40)/(16-8) = int(88/8) = 11

Görüntüdeki sınırların belirginleştirilmesi için sıfır toplamlı doğrusal filtrelerde kullanılabilmektedir. Bu filtrelerde katsayılar toplamı sıfırdır. Filtre zenginleştirilecek detaya (sınır) bağlı olarak düşey yada yatay doğrultuda geçirilir. Filtrenin hangi doğrultuda geçirileceği hangi doğrultulardaki detayların zenginleştirileceğine bağlıdır [36]. Düşey doğrultu için tasarlanmış 3x3 boyutlu filtre örneği Şekil 2.25'de verilmiştir.

| | | |
|---|---|---|
| 1 | 1 | 1 |
| 1 | -2 | 1 |
| -1 | -1 | -1 |

**Şekil 2.25.** Sıfır toplamlı kenar filtresi [36].

## 2.7. C# DİLİNDE ÖZYÜZ YÖNTEMİ İLE YÜZ TANIMA SİSTEMİ GELİŞTİRİLMESİ

Bu bölümde tez çalışmasında geliştirilen yazılımın tasarımı ve bunun uygulamasına yer verilmiştir. Geliştirilen uygulama 4 ana bölümden oluşmaktadır;

- Yüzün tespit edilmesi; yüz bulma
- Görüntüdeki aydınlatma etkisinin azaltılması ve görüntünün iyileştirilmesi, gürültü azaltma, önişleme
- Poz ve ışıklandırma durumlarına göre en uygun yöntemin belirlenmesi
- Elde edilen en uygun yönteme göre kişi tanıma

Uygulama Microsoft Visual Studio 2010 C#.Net programı kullanılarak geliştirilmiştir. Görüntü işleme algoritmaları için EMGU CV kütüphanesi, veritabanı işlemleri için SQL Server 2008 Express kullanılmıştır. Sistem tasarımında aşağıdaki 2 özelliğe dikkat edilmiştir.

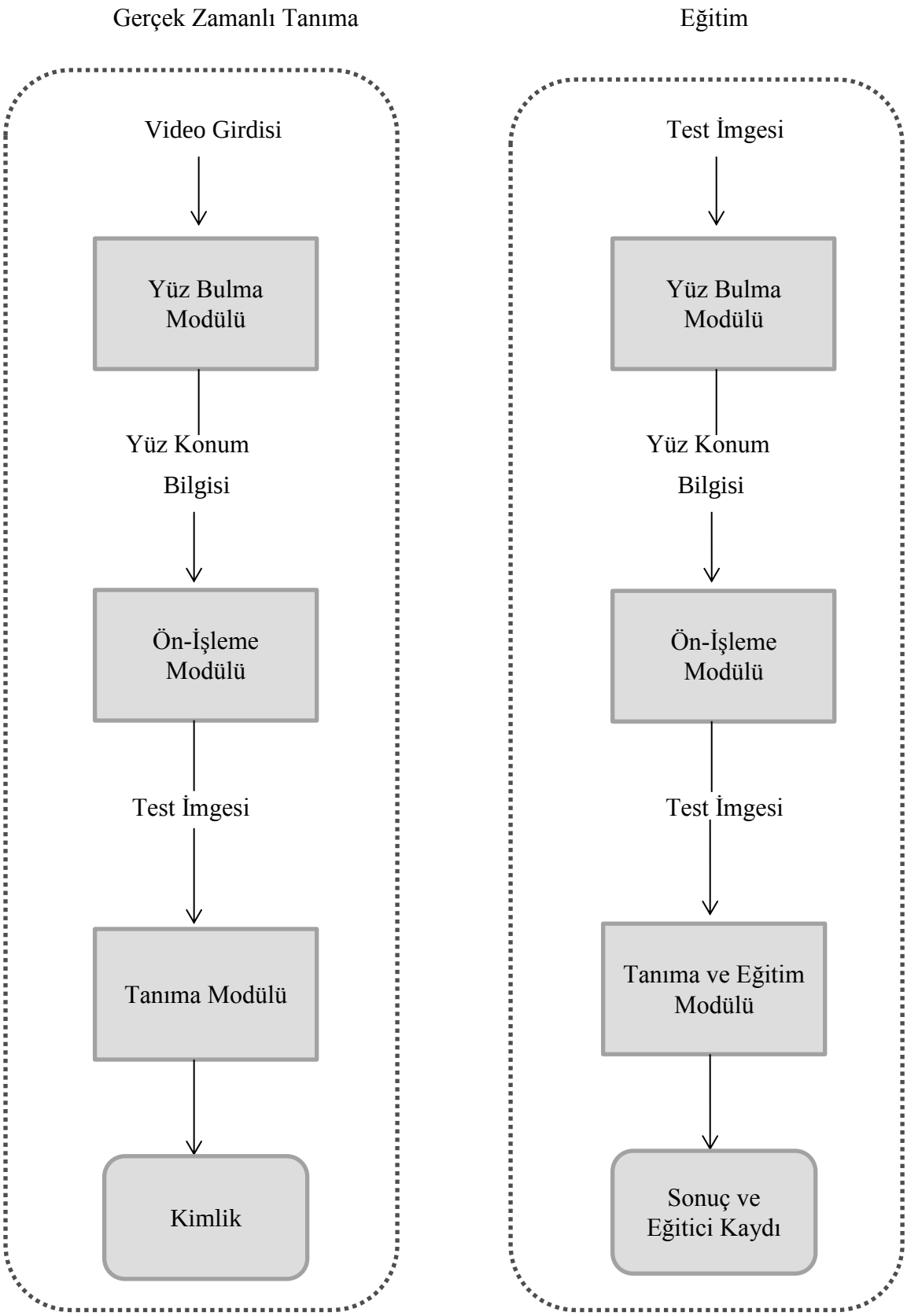
- **Geliştirilebilirlik:** Yüz tanıma sistemi geliştirilmeye açık bir yapıya sahiptir. Gerek duyulan ihtiyaçlara göre sistem modülleri geliştirilebilir veya değiştirilebilir.

- **Modülerlik:** Yüz tanıma sistemi birbirinden bağımsız birçok modülün birleşimiyle meydana gelmiştir. Böylelikle sistemde karmaşıklığın önüne geçilmiştir.

### 2.7.1. Yüz Tanıma Sistemi Mantıksal Tasarımı

Sistem hem video kameradan yakaladığı anlık görüntü için hem de bilgisayar üzerinde kayıtlı olan bir görüntü için yüz tespiti ve yüz tanıması yapmaktadır. Sistemde kullanılacak kamera sistemi kullanım amacına göre farklılık gösterebilir. Güvenlik kamerası, web cam veya el kamerası sistemde kullanılabilir. Ancak kameradan,

görüntülerin net olarak alınabilmesini sağlamak için yüksek çözünürlüklü kameralar tercih edilmelidir. Şekil 2.26'da yüz tanıma sisteminin mantıksal yapısı gösterilmektedir.




**Şekil 2.26.** Yüz tanıma sistemi mantıksal yapısı.

### 2.7.2. Yüz Bulma Modülü

Bir görüntünün tamamı yüze ait olmadığından, ilk olarak yüzün tespit edilmesi gerekmektedir. Bu çalışmada Haar dalgacık özniteliklerine dayalı çok katlı sınıflandırıcı kullanılmıştır. Öznitelikler Adaboost algoritması ile seçilmekte ve gereken toplam öznitelik sayısı hedeflenen yanlış kabul oranına bağlı olarak değişmektedir. Adaboost algoritması ilk olarak Freund ve Schapire tarafından önerilmiştir [37].

*2.7.2.1. Adaboost Algoritması İle Yüz Bulma*

Adaboost algortiması;

- Eğitim örnekleri ($x_1$ ,$y_1$ ),...,( $x_n$,$y_n$ ) olarak düzenlenir. Burada $y_i$ ,negatif örnekler için 0, pozitif örnekler için 1 olarak kabul edilir.
- m ve n sırasıyla pozitif ve negatif örnek sayısını göstermek üzere ağırlıklar $w_{1,i} = \frac{1}{2m}$ , $\frac{1}{2l}$ olacak şekilde her $y_i \in \{0,1\}$ olarak başlangıç değeridir.
- T iterasyon sayısı olmak üzere, her t=1,..,T için:
  - Ağırlıklar normalize edilir.

$$w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}} \tag{2.29}$$

  - Her bir j özniteliği için , sadece bu özniteliği kullanan bir $h_j$ sınıflayıcısı eğitilir. Hata $w_t$ ağırlığına göre ölçülür.

$$\varepsilon_j = \sum_i w_{i|h_j(x_i)-y_i|} \pi r^2 \tag{2.30}$$

  - En az $\varepsilon_j$ hatasına sahip $h_t$ sınıflayıcısı seçilir.
  - Ağırlıklar güncellenir.

$$w_{t+1,i} = w_{t,i}\beta_t^{1-e_i} \tag{2.31}$$

$x_i$ doğru olarak sınıflandırıldı ise $e_i$=0, aksi halde $e_i$=1 olur. $\beta_t$ ise,

$$\beta_t = \frac{\varepsilon_t}{1-\varepsilon_t} \tag{2.32}$$

olarak bulunur.

- Sonunda oluşan sınıflayıcı:

$$h(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) > \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & diğer \end{cases} \quad (2.33)$$

$$\alpha_t = \log \frac{1}{\beta_t} \quad (2.34)$$

olarak alınır [38].

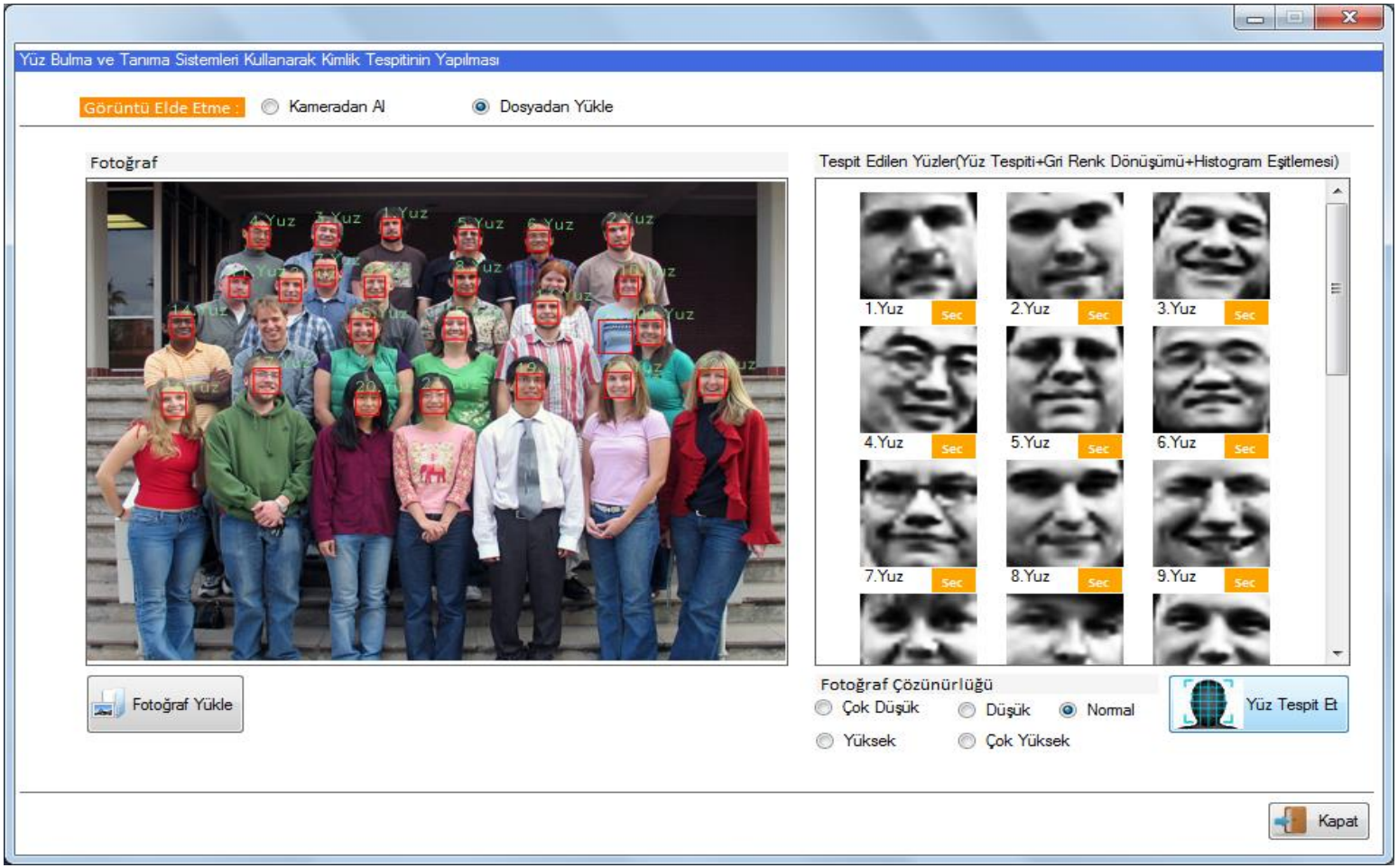
Bu uygulamada Adaboost algoritması "opencv-haar-classifier" sınıflandırıcısını kullanmaktadır. Yüz bulma işlemi bu sınıflandırıcıya göre yapılmaktadır. Şekil 2.27' de bu sınıflandırıcının xml yapısından bir kesit gösterilmektedir.

```
<?xml version="1.0"?>
...
<opencv_storage>
  <haarcascade_frontalface_default type_id="opencv-haar-classifier">
    <size>24 24</size>
    <stages>
      <_>
        <!-- stage 0 -->
        <trees>
          <_>
            <!-- tree 0 -->
            <_>
              <!-- root node -->
              <feature>
                <rects>
                  <_>6 4 12 9 -1.</_>
                  <_>6 7 12 3 3.</_>
                </rects>
                <tilted>0</tilted>
              </feature>
              <threshold>-0.0315119996666908</threshold>
              <left_val>2.0875380039215088</left_val>
              <right_val>-2.2172100543975830</right_val>
            </_>

          </_>
          <_>
            <!-- tree 1 -->
            <_>
              <!-- root node -->
              <feature>
```

**Şekil 2.27.** "Opencv-Haar-Classifier" katmanlı yapısı

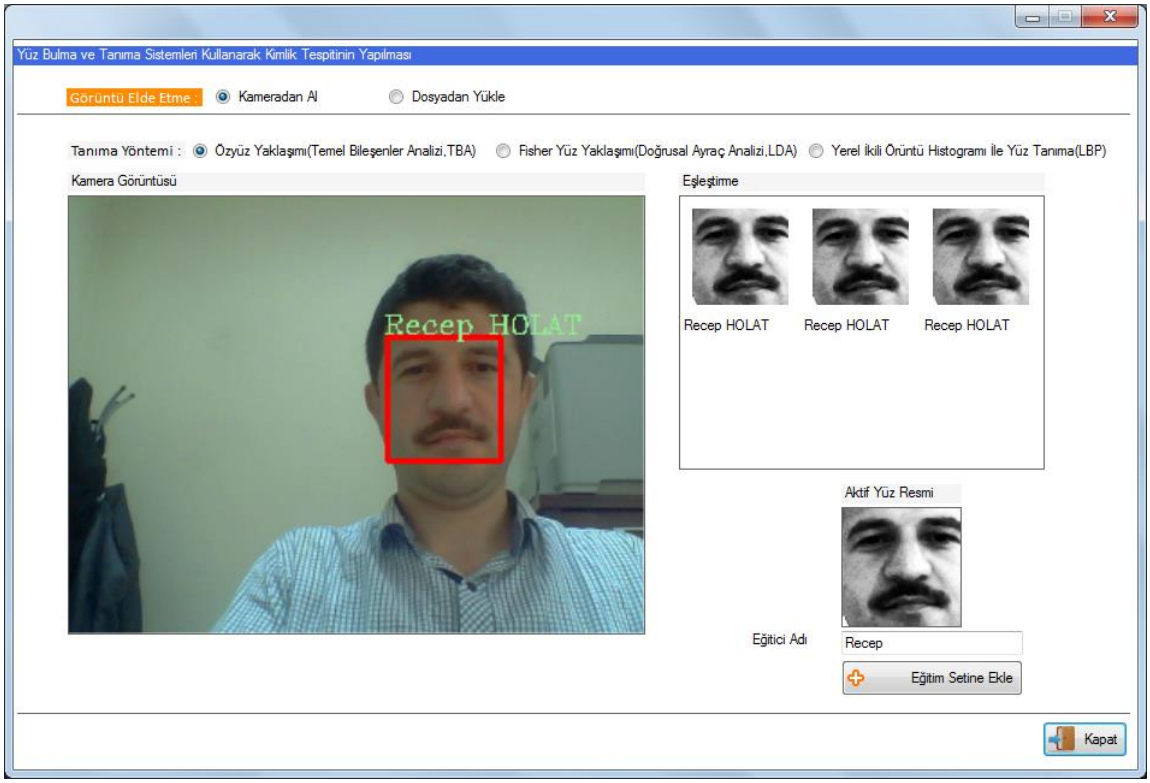
Şekil 2.28’ de Adaboost algoritması kullanılarak tespit edilen, bulunan yüzler görülmektedir.




**Şekil 2.28.** Adaboost algortiması ile tespit edilen yüzler.

*2.7.2.1. Yüz Bulma Modülü Temel Bilgiler ve Çalışma Şekli*

Yüz bulma modülü, hem kameradan elde edilen anlık görüntü üzerinde hem de kayıtlı bir fotoğraf üzerinde yüz bulma işlemini yapmaktadır. Bu işlem, şekil 1'de de görüldüğü gibi Görüntü Elde Etme özelliğinin Kameradan Al veya Dosyadan Yükle seçeneklerinden birinin seçilmesi ile gerçekleşmektedir.




**Şekil 2.29.** Yüz bulma modülü; görüntü elde etme özelliğinin "kameradan al" yapılması.

Görüntü Elde Etme özelliği Kameradan Al seçildiğinde Şekil 2.29'da gösterilen ekran karşımıza çıkmaktadır. Bu ekran üzerinde, Özyüz Yaklaşımı, Fisher Yüz Yaklaşımı, Yerel İkili Örüntü yüz tanıma yöntemleri bulunmaktadır. Bu modül hem yüz bulma hem de yüz tanıma işlemini aynı anda yapmaktadır. Yüz bulma işlemi Adaboost algoritması ile yüz tanıma ise bahsedilen yöntemlerden birinin seçilmesi ile yapılmaktadır.

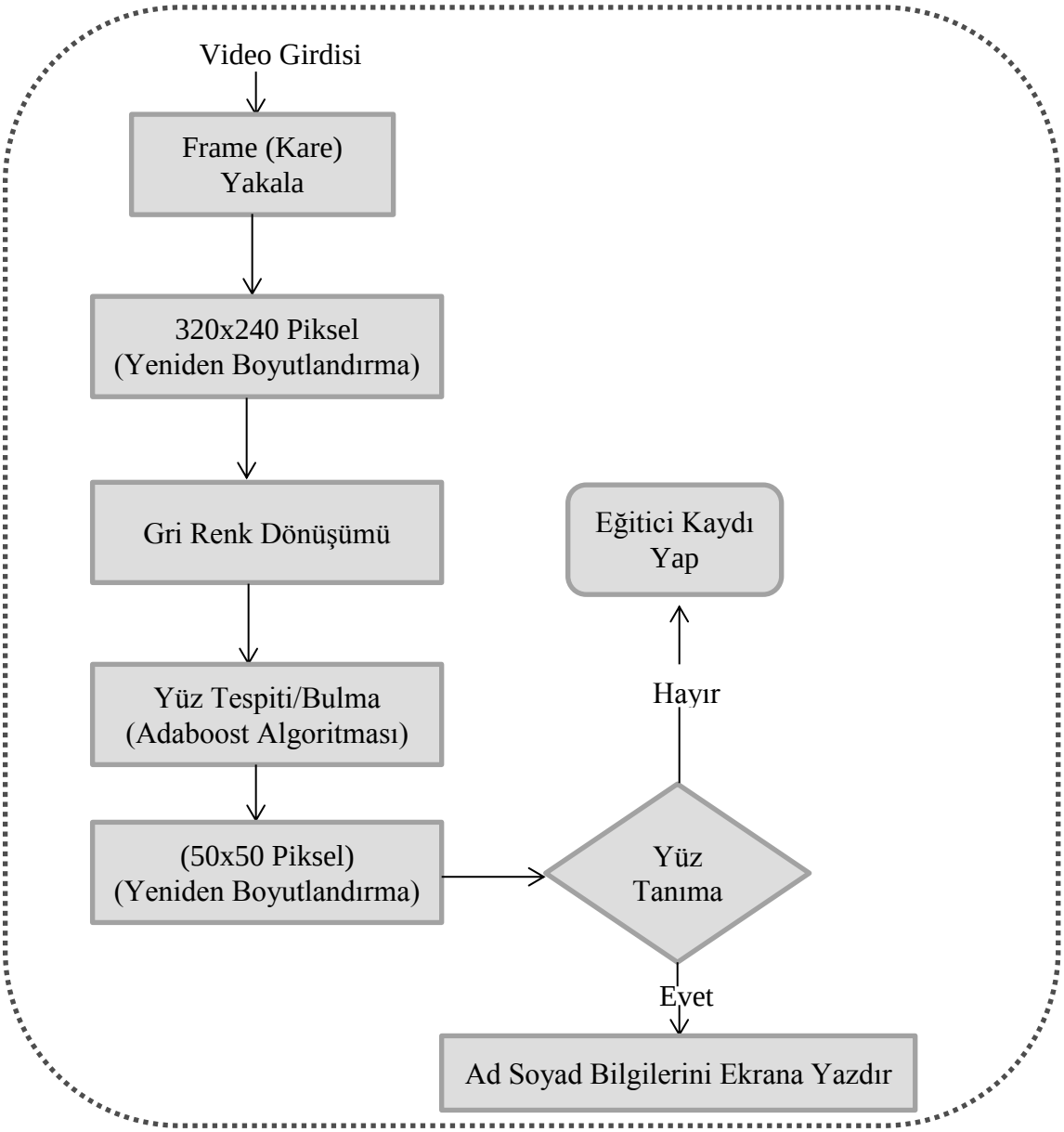
Anlık kamera görüntüsünden elde edilmiş olan görüntü üzerinde yüzler bulunup daha sonrada bulunan bu yüzler eğitici veritabanındaki diğer yüz verileri karşılaştırılarak Şekil 2.29'da sağ üst kısımda bulunan Eşleştirme alanında görüntülenmektedir. Karşılaştırma sonucunda tanınan kişilerin ad, soyad bilgileri anlık video görüntüsü üzerinde yazılmaktadır. Tespit edilen en son yüz imgesi Eğitici Adı alanında

görüntülenmektedir. Bunun amacı eğer görüntünün eğitici veritabanındaki karşılaştırma sonucunda tanınmaması halinde sisteme yeni eğitici verilerini kaydetmektir.

Yukarıda anlatılanları özetlersek, Görüntü Elde Etme Kameradan Al seçildiğinde;

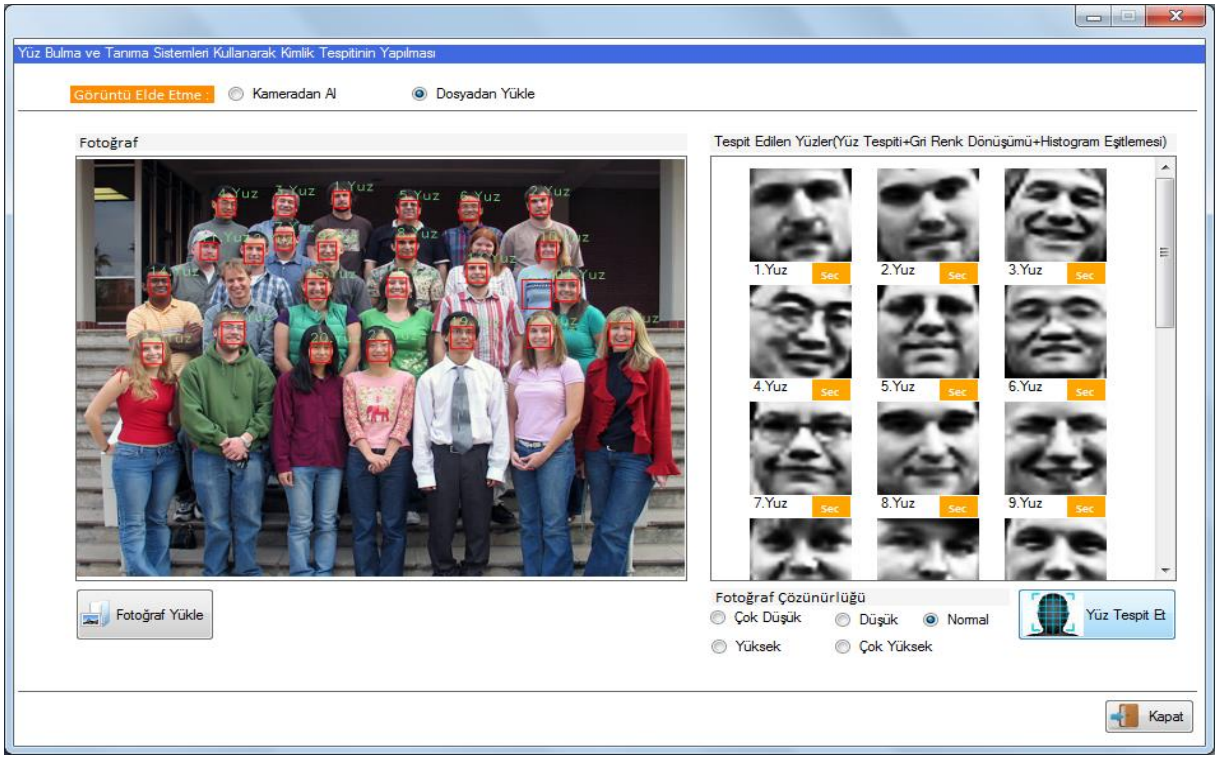
- Yüz Bulma
- Yüz Tanıma
- İstenirse Yeni Eğitici Kaydı

işlemleri yapılmaktadır. Bu işlemler Şekil 2.30'da gösterilmiştir.




**Şekil 2.30.** Yüz bulma modülü akış diyagramı.

Görüntü Elde Etme özelliği Dosyadan Yükle seçildiğinde Şekil 2.31'deki ekran görüntülenmektedir.




**Şekil 2.31.** Görüntü elde etme özelliğinin "dosyadan yükle" yapılması.

Bu ekranda sadece yüz bulma işlemi yapılmaktadır. Yüz bulma işlemine başlamak için ilk olarak Fotoğraf Yükle düğmesi kullanılarak ilgili fotoğraf yüklenir. Daha sonra Fotoğraf Çözünürlüğü ayarlanarak Yüz Tespit Et düğmesine tıklanır. Fotoğraf Çözünürlüğü fotoğrafın boyutu ile ilgili ayarları içermektedir.

Fotoğraf Çözünürlüğü sayesinde program sırasıyla fotoğrafı aşağıdaki ölçülerde yeniden boyutlandırarak yüz bulma işlemini gerçekleştirir.

Çok Düşük:

1. 320 x 240 Piksel
2. 50 x 45 Piksel
3. Orijinal Piksel X  x Orijinal Piksel Y
4. (Orijinal Piksel X / 2)  x (Orijinal Piksel Y / 2)
5. (Orijinal Piksel X / 3)  x (Orijinal Piksel Y / 3)
6. (Orijinal Piksel X / 4)  x (Orijinal Piksel Y / 4)

Düşük:

1. 640 x 480 Piksel
2. (Orijinal Piksel X / 2)  x (Orijinal Piksel Y / 2)

Normal:

1. 1280 x 960 Piksel
2. Orijinal Piksel X x Orijinal Piksel Y

Yüksek:

1. 2560 x 1920 Piksel
2. Orijinal Piksel X x Orijinal Piksel Y

Çok Yüksek:

1. 5120 x 3840
2. Orijinal Piksel X x Orijinal Piksel Y

Çözünürlük yüz bulmada önemli bir yere sahiptir. Şekil 2.31'deki fotoğraf kullanıldığında bahsedilen çözünürlük değerlerine göre tespit edilen yüzlerin sayıları;

- Çok Düşük: Hiç yüz tespit edilmemiştir.
- Düşük: 1
- Normal: 24
- Yüksek: 25
- Çok Yüksek: 28 olarak bulunmuştur.

Yüzler tespit edildikten sonra, bulunan yüzler ekranın sağ tarafına sırasıyla 1.Yüz,2.Yüz şeklinde gösterilmektedir. Bulunan her yüz ekranda gösterilmeden önce aşağıdaki işlemlerden geçirilir;

- Gri renk dönüşümü
- 50 x 50 piksel yeniden boyutlandırma
- Histogram eşitlemesi

Tespit edilen yüz imgesi boyutları, veritabanında bulunan imgelerle eşit olmalıdır. Bu nedenle yüzler, veritabanında bulunan imgelerle eşit boyutlara gelecek şekilde yeniden boyutlandırılır. Bu çalışmada yüz imgeleri, gri renk dönüşümü yapılarak, 50 x 50 piksel olarak boyutlandırılmıştır. Aydınlatma değişimleri yüz tanıma sistemi için sorun teşkil

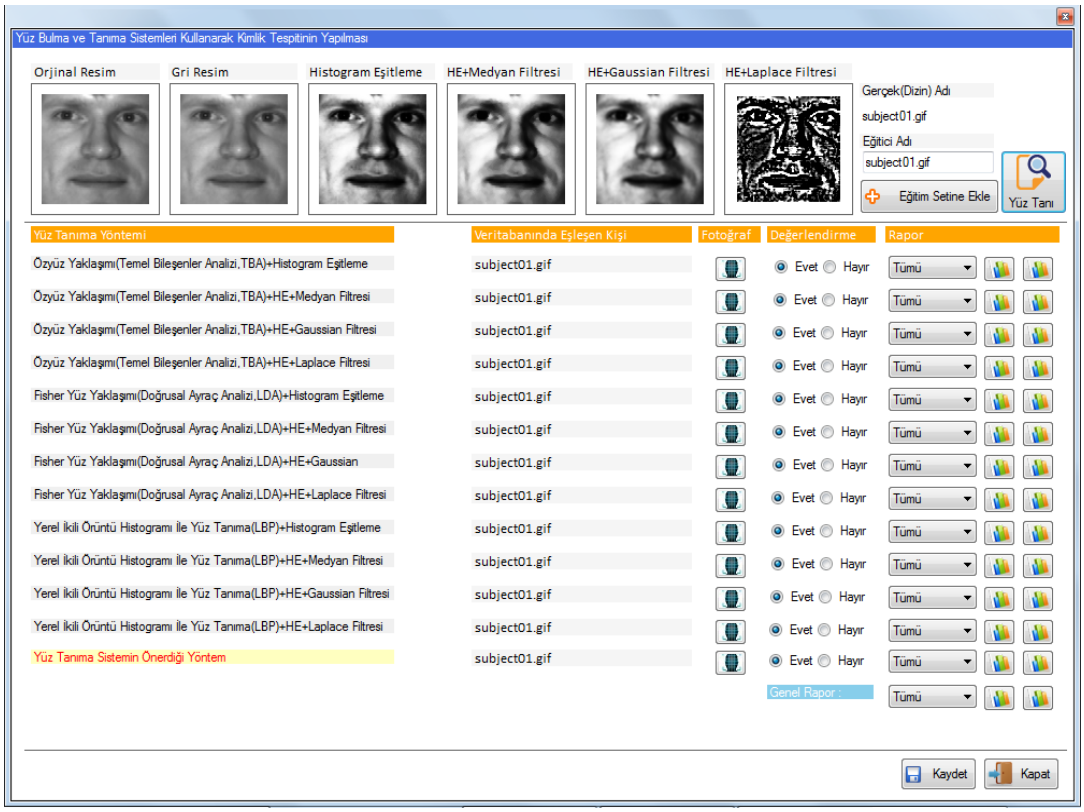
eden konular arasındadır. Bunun etkisini en aza indirmek için gri renk dönüşümü ve histogram eşitlemesi yapılmıştır.

Histogram, görüntüdeki her bir renk değerinin ağırlık dağılımını gösterir. Histogram eşitleme görüntünün kontrastının iyileştirmek için yapılır. Renk değerleri düzgün dağılımlı olmayan görüntüler için uygun bir görüntü eşitleme metodudur.

### 2.7.3. Yüz Tanıma Modülü

#### *2.7.3.1. Yüz Tanıma Modülü Temel Bilgiler ve Çalışma Şekli*

Yüz bulma işleminden sonra yüz tanıma işlemine geçilir. Bunu yapmak için Şekil 2.31'da tespit edilen yüzler alanındaki her bir yüzün altında yer alan Seç düğmesine tıklanır. Şekil 2.32'deki ekrana ulaşılır.




**Şekil 2.32.** Yüz tanıma modülü.

Yüz tanıma uygulamasının bu modülü;

- Eğitici Kaydı
- Yüz Tanıma
- İstatiksel Verilerin Grafiksel Gösterimi adımlarını içermektedir.

Yeni bir eğitici kaydetmek için, Eğitim Setine Ekle düğmesi kullanılır. Bu işlem sonucunda eğitici verisi, Şekil 2.33'te gösterildiği gibi, gri renk dönüşümü yapıldıktan sonra aşağıdaki işlemlerden geçirilerek sistemde 4 farklı şekilde saklanmaktadır.

Bunlar;

- Histogram Eşitleme
- Histogram Eşitleme ve Medyan Filtresi
- Histogram Eşitleme ve Gaussian Filtresi
- Histogram Eşitleme ve Laplace Filtresi

**Şekil 2.33.** Eğitici kaydının 4 farklı şekilde yapılması.

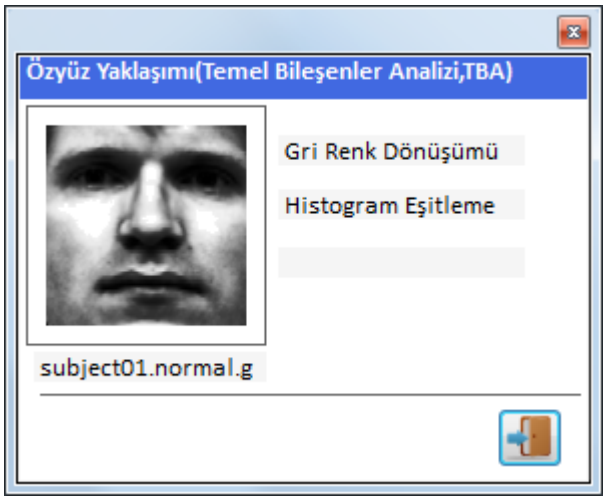
Bu çalışmada yüz tanıma yöntemleri olarak;

- Özyüz Yaklaşımı ile Yüz Tanıma
- Fisher Yüz Yaklaşımı ile Yüz Tanıma
- Yerel İkili Örüntü Histogramı ile Yüz Tanıma

yöntemleri kullanılmıştır. Her bir yöntem yukarıda geçen 4 farklı eğitici kaydı için ayrı ayrı uygulanmakta ve sonuçları sistemde kayıt altına alınmaktadır.

Yüz tanıma işlemini başlatmak için Yüz Tanı düğmesi tıklanır. Uygulama her bir yöntem için yüz tanıma yaparak sonuçları Veritabanında Eşleşen Kişi başlığı altına yazdırmaktadır. Eşleşmeyen durumlar için bu alan boş bırakılmaktadır. Çıkan sonuçlar, Değerlendirme başlığı altındaki Evet veya Hayır seçenekleri kullanılarak değerlendirilir. Eşleşen fotoğraf doğru ise evet seçeneği, eşleşmiyorsa veya yanlış eşleştiği durumlarda da hayır seçeneği işaretlenerek kaydedilir.

Fotoğraf başlığı altında bulunan düğmeye tıklandığında ise veritabanında eşleşen fotoğraf görüntülenmektedir (Şekil.2.34).



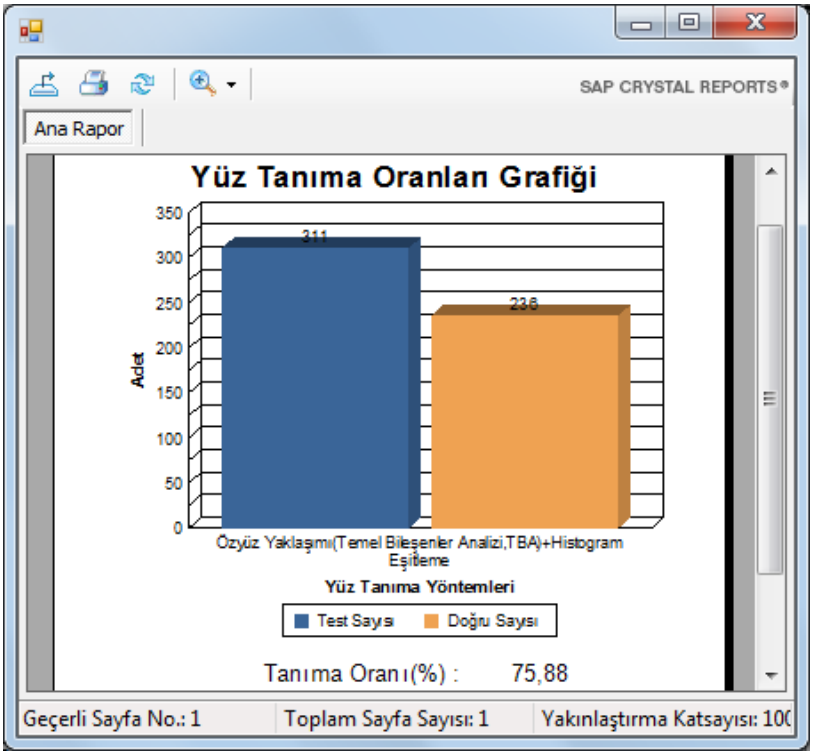

**Şekil 2.34.** Veritabanında eşleşen görüntü.

Bu çalışmada Yale ve ORL veritabanları kullanılmıştır. Yale veritabanı 15 deneğe ait toplam 165 görüntü içermektedir. Her deneğe ait ifade ve aydınlatma farklılıkları içeren 11 görüntü bulunmaktadır. Bu sistemde, kullanılan yöntem ve filtrelerin tanıma başarı oranları ayrı ayrı hesaplanmaktadır. Ayrıca görüntünün aydınlatma ve poz durumları dikkate alındığında her yöntemin bahsedilen durumlar için tanıma başarı oranları grafiksel olarak gösterilmektedir. Bu grafiklere ulaşmak için Rapor başlığı altındaki düğmeler kullanılmalıdır. Bu başlık altındaki açılır liste aşağıdaki değerleri içermektedir;

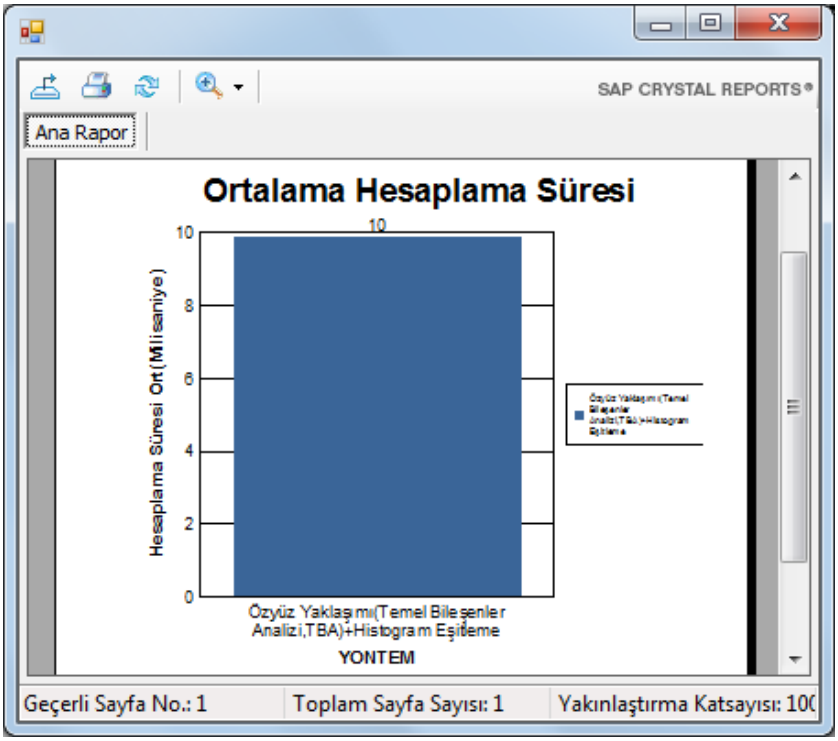
- ✓ Tümü
- ✓ Merkezden Aydınlanmış
- ✓ Gözlüklü
- ✓ Mutlu
- ✓ Soldan Aydınlanmış
- ✓ Gözlüksüz
- ✓ Sağdan Aydınlanmış
- ✓ Üzgün
- ✓ Uykulu
- ✓ Şaşkın
- ✓ Göz Kırpmış

Raporlama başlığı altındaki birinci düğme kullanıldığında; açılır listedeki Tümü seçeneği seçilmiş ise her poz ve aydınlatma durumları için ilgili yüz tanıma yöntemin

genel tanıma başarı oran grafiği (Şekil 2.35), diğer seçenekler içinde tanıma başarı oranı grafikleri yüzde olarak gösterilmektedir.



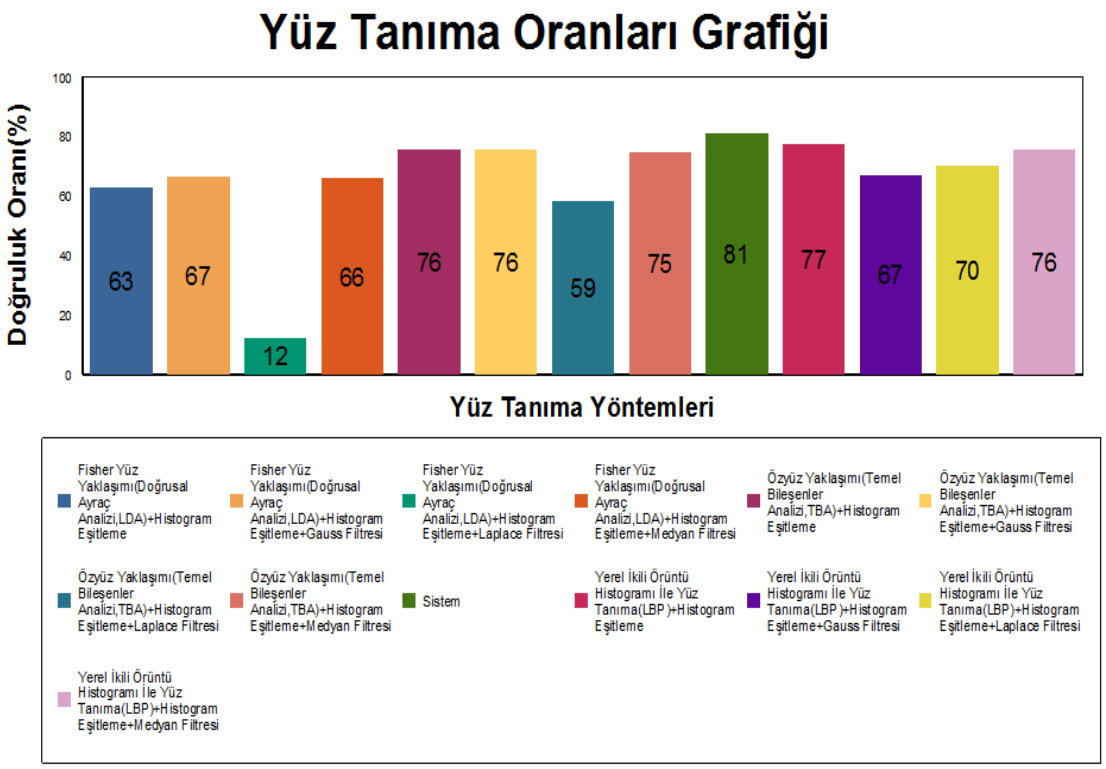

**Şekil 2.35.** Tümü seçildiğinde yüz tanıma oranı grafiği.




**Şekil 2.36.** Tümü seçildiğinde yüz tanıma ortalama hesaplama süresi grafiği.

Raporlama başlığının altındaki ikinci düğme ile yüz tanıma ortalama hesaplama süresi grafiğine ulaşılır (Şekil 2.36).

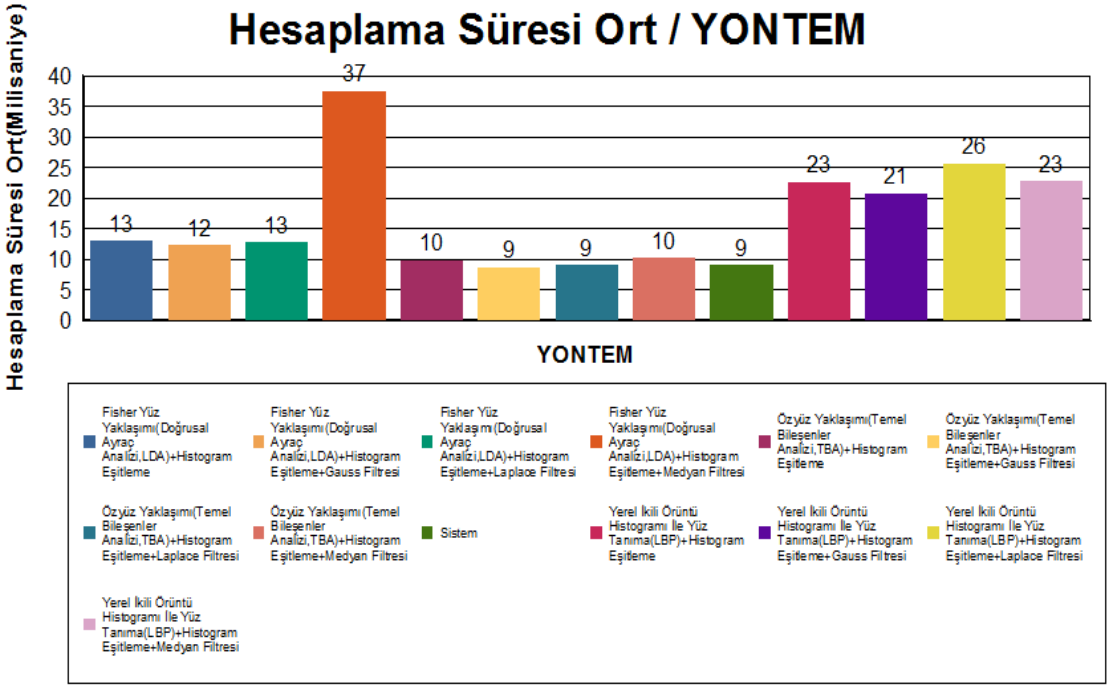
Şekil 2.32'de Genel Rapor, her poz ve aydınlatma durumları için kullanılan yüz tanıma yöntemlerin tanıma başarı oranları grafiğini (Şekil 2.37) ve ortalama hesaplama süreleri grafiğini (Şekil 2.38) göstermektedir.




**Şekil 2.37.** Yüz tanıma oranları grafiği.

Şekil 2.37 yüz tanıma yöntemlerin, genel yüz tanıma başarı oranlarını göstermektedir. Bu grafik, açılır listeden biri seçilerek alındığında ise geçerli poz ve aydınlatma durumu için yüz tanıma başarı oranları grafiğini göstermektedir.

Şekil 2.38 yüz tanıma yöntemlerin, genel ortalama hesaplama sürlerini göstermektedir. Bu grafik, açılır listeden biri seçilerek alındığında ise geçerli poz ve aydınlatma durumu için yüz tanıma ortalama hesaplama süreleri grafiğini göstermektedir.




**Şekil 2.38.** Yüz tanıma ortalama hesaplama süresi grafiği.

*2.7.3.2. Öklid Mesafesi ile Karar Verme*

Sınıflandırma, bir grup nesnenin benzer özelliklerinden yola çıkarak gruplandırılması anlamına gelmektedir. Bilimsel anlamda bir karar verme işlemidir. Sınıflandırma yapılabilmesi için ilk olarak görüntülerden öznitelik çıkartılması gerekmektedir. Bu çalışmada özniteliklerin çıkarılmasında; Temel Bileşen Analizi, Doğrusal Ayraç Analizi ve Yerel İkili Örüntü kullanılmıştır. Literatürde çok sayıda sınıflandırma algoritması vardır. Bu çalışmada öklid uzaklığı kullanılmıştır. Öklid uzaklığı, karşılaştırılmak istenen verinin örnek kümeler arasında en yakın olanını bulmak için kullanılır.

Öklid uzaklığı (2.35) aşağıdaki formülle hesaplanır;

$$\sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2} \qquad (2.35)$$

Öklid uzaklığı, Pisagor Teoremi temel alarak bir noktanın diğer noktaya olan doğrusal uzaklığını hesaplama yöntemidir. Sınıflandırmada karşılaştırmak için oluşturulan veri kümelerinin sayısına göre, bilgisayar tüm veri setlerini tek tek tarayarak yeni verinin ortalamasıyla karşılaştır ve nesneyi en yakın kümeye atar [39].

*2.7.3.3. Yüz Tanıma Sistemi Önerilen Yöntem*

Önerilen yöntem, poz ve aydınlatma durumlarına göre en başarılı yöntem hangisi ise yüz tanıma işlemini o yönteme göre yapan yöntemdir. Buna karar verirken içerdeki test verilerinden yararlanmaktadır. Hangi yöntem daha başarılı ise onun üzerinden işlemlerini yapan yöntemdir.

### 2.7.4. Programda Kullanılan Önemli Fonksiyon ve Yöntemler

*2.7.4.1. Yüz Bölgesinin Tespiti*

1. public CascadeClassifier Face = new CascadeClassifier(Application.StartupPath + "/Cascades/haarcascade_frontalface_default.xml")

   Yüz tespiti yapan metoddur. Bu metot yüz bulma işlemini haarcascade sınıflandırıcısını kullanarak gerçekleştirir.

2. MCvFont font = new MCvFont(FONT.CV_FONT_HERSHEY_COMPLEX, 0.5, 0.5)

   Yüz bulma ve tanıma video kameradan yapılıyorsa, bulunan kişi bilgileri hemen yüz bölgesinin altına yazılmaktadır. Yukarıdaki ifade kullanılan yazıya ait tanımlamayı gösterir.

3. Capture grabber

   Video kameradan kare (frame) kare görüntüleri yakalayan ve bunları kendinde saklayan Emgu CV sınıfından oluşturulmuş bir değişkendir.

4. Rectangle[] facesDetected = Face.DetectMultiScale(gray_frame, 1.2, 10, new Size(50, 50), Size.Empty)
   Verilen görüntüde dikdörtgen bölgeleri bulur ve bir dizi olarak bu bölgeleri saklar. Görüntüdeki yüz bölgesinin koordinatlarını hesaplar.
5. currentFrame.Convert<Gray, Byte>();
   Video kameradan yakalanan anlık görüntünün rengini gri olarak değiştiren fonksiyondur.
6. currentFrame.Copy(facesDetected[i]).Convert<Gray, byte>().Resize(100, 100, Emgu.CV.CvEnum.INTER.CV_INTER_CUBIC)

   Görüntüdeki yüz bölgesini tespit ettikten sonra bu bölgeyi kopyalayan fonksiyondur.

7. currentFrame.Draw(facesDetected[i], new Bgr(Color.Red), 2)
   Görüntü üzerinde yüz bölgesini çerçeve olarak çizen fonksiyondur.
8. void ADD_Face_Found(Image<Gray, Byte> img_found, string name_person, int match_value)

```
{
   PictureBox PI = new PictureBox();
   PI.Location = new Point(faces_panel_X+10, faces_panel_Y+10);
   PI.Height = 80;
   PI.Width = 80;
   PI.SizeMode = PictureBoxSizeMode.StretchImage;
   PI.Image = img_found.ToBitmap();
   Label LB = new Label();
   LB.Text = name_person + " " + match_value.ToString();
   LB.Location = new Point(faces_panel_X, faces_panel_Y + 100);
   //LB.Width = 80;
   LB.Height = 15;

   this.Faces_Found_Panel.Controls.Add(PI);
   this.Faces_Found_Panel.Controls.Add(LB);
   faces_count++;
   if (faces_count == 3)
   {
      //faces_panel_X = 0;
      faces_panel_X = 0;
      faces_panel_Y += 100;
      faces_count = 0;
   }
   else faces_panel_X += 100;

   if (Faces_Found_Panel.Controls.Count > 12)
   {
      Clear_Faces_Found();
   }
}
```

Bulunan yüzlerin ekranın sağ tarafında buluna  Tespit Edilen Yüzler(Yüz Tespiti+Gri Renk Dönüşümü+Histogram Eşitlemesi) kısmında gösterilmesini sağlayan kullanıcı tanımlı fonksiyondur.

*2.7.4.2. Yüz Tanıma*

1. EigenFaceRecognizer recognizerEigen =new EigenFaceRecognizer(80, double.PositiveInfinity);
   Özyüz yaklaşımı ile yüz tanımada kullanılan değişkeni ifade eder.
2. FisherFaceRecognizer recognizerFisher = new FisherFaceRecognizer(0, 3500);
   Fisher yaklaşımı ile yüz tanımada kullanılan değişkeni ifade eder.
3. LBPHFaceRecognizer recognizerLBPH = new LBPHFaceRecognizer(1, 8, 8, 8, 100);
   Yerel ikili örüntü yaklaşımı ile yüz tanımada kullanılan değişkeni ifade eder.

*2.7.4.3. Gerekli EmguCV Kütüphaneleri*

1. Emgu.CV.dll
   OpenCV resim işleme fonksiyonlarını içerir. Bunlara;
   - ✓ Capture
   - ✓ EigenFaceRecognizer
   - ✓ EigenObjectRecognizer
   - ✓ EigenObjectRecognizer.RecognitionResult
   - ✓ FisherFaceRecognizer
   - ✓ HaarCascade
   - ✓ Retina

   örnek verilebilir.
2. Emgu.CV.UI.dll
   Resim görüntülemek için kullanıcı arayüzü sağlar. Bu arayüzlerden bazıları şunlardır;
   - ✓ HistogramBox
   - ✓ HistogramViewer
   - ✓ ImageBox
   - ✓ ImageViewer
   - ✓ MatrixViewer

3. Emgu.Util.dll

   Emgu.CV projeleri tarafından kullanılan araçlar kolleksiyonudur. Bunlara örnek olarak;

   - ✓ CvException
   - ✓ CvToolbox
   - ✓ DataLogger
   - ✓ TbbTaskScheduler
   - ✓ VectorOfDataMatrixCode

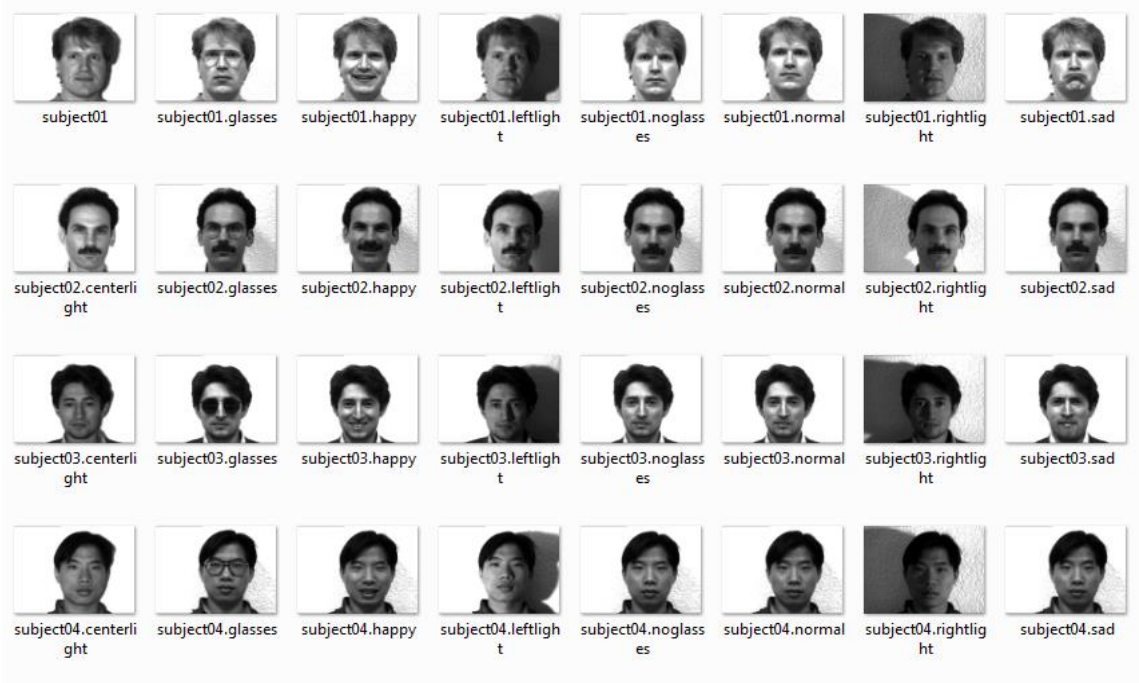
   verilebilir.

# 3. BULGULAR VE TARTIŞMA

Bu bölümde deneylerin sonuçları sunulacaktır. Deneyler C# (C Sharp) programlama dili ile yazılan yazılım kullanılarak yapılmıştır. Deney sonuçları SQL Server 2008 veritabanına kaydedilmekte ve Crystal Reports ile raporlanmaktadır.

## 3.1. Yale Veritabanı Deneyleri

Yale veritabanı 15 deneğe ait toplam 165 görüntü içermektedir. Her deneğe ait ifade ve aydınlatma farklılıkları içeren 11 görüntü bulunmaktadır. Bunlar, sağdan aydınlanmış, merkezden aydınlanmış, soldan aydınlanmış, gözlüksüz, gözlüklü, normal, göz kırpmış, uykulu, şaşkın, mutlu, üzgün görüntülerdir (Şekil 3.1).




**Şekil 3.1.** Yale veritabanından örnek görüntüler

Yale veritabanında 2 ayrı deney yapılmıştır. Birinci deneyimizde eğitici setine deneklere ait sadece normal görüntüleri eklenmiş ve diğer tüm görüntüler ile tanıma işlemi yapılmıştır. Sonuçlar aşağıdaki Çizelge 3.1'de gösterilmiştir.

**Çizelge 3.1.** Deneklere ait normal görüntülerin yer aldığı eğitim veritabanında yapılan test sonuçları(%'lik tanıma oranları).

| Yöntemler | Filtreler | Yale Veritabanı | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Tümü | Merkezden Aydınlanmış | Sağdan Aydınlanmış | Soldan Aydınlanmış | Gözlüklü | Gözlüksüz | Normal | Uykulu | Şaşkın | Mutlu | Üzgün | Göz Kırpmış |
| TBA+HE | | 76 | 74 | 39 | 13 | 93 | 87 | 100 | 87 | 90 | 87 | 87 | 87 |
| | Medyan | 75 | 56 | 39 | 13 | 93 | 87 | 100 | 87 | 97 | 87 | 87 | 87 |
| | Gauss | 76 | 74 | 39 | 13 | 93 | 87 | 100 | 87 | 90 | 87 | 87 | 87 |
| | Laplace | 59 | 37 | 21 | 20 | 77 | 87 | 100 | 77 | 57 | 60 | 63 | 60 |
| LDA+HE | | 63 | 41 | 11 | 7 | 80 | 87 | 100 | 87 | 63 | 60 | 80 | 87 |
| | Medyan | 66 | 41 | 11 | 7 | 80 | 87 | 100 | 87 | 77 | 73 | 87 | 87 |
| | Gauss | 67 | 44 | 11 | 7 | 90 | 87 | 100 | 87 | 70 | 73 | 87 | 87 |
| | Laplace | 12 | 0 | 0 | 0 | 13 | 53 | 100 | 0 | 0 | 0 | 7 | 0 |
| LBP+HE | | 77 | 70 | 18 | 47 | 93 | 93 | 100 | 100 | 87 | 63 | 100 | 87 |
| | Medyan | 76 | 52 | 11 | 33 | 93 | 93 | 100 | 100 | 93 | 73 | 100 | 87 |
| | Gauss | 67 | 30 | 18 | 20 | 93 | 87 | 100 | 100 | 67 | 60 | 87 | 87 |
| | Laplace | 70 | 59 | 36 | 27 | 63 | 93 | 100 | 93 | 67 | 73 | 93 | 77 |
| Önerilen Yöntem | | 81 | 64 | 29 | 53 | 87 | 93 | 100 | 100 | 93 | 87 | 100 | 80 |

Çizelge 3.1 incelendiğinde en yüksek tanımayı gerçekleştiren yöntemler % 77 ile LBP+HE yöntemleridir. Sistemin önerdiği yönteme göre tanıma oranı ise % 81 olup en iyi tanıma oranından % 4 daha iyi sonuç vermiştir.

Merkezden, sağdan ve soldan aydınlanmış görüntüler için sonuçlar değerlendirildiğinde TBA+HE ve LBP+HE yöntemlerin daha başarılı olduğu söylenebilir.

Eğitici kaydı olan normal görüntülerin sonuçları incelendiğinde tüm yöntemlerin % 100 başarıya sahip olduğu görülmektedir.

Yapılan ikinci deneyimizde ise eğitim setine deneklere ait normal ve merkezden aydınlanmış görüntüler eklenerek tanıma işlemi yapılmıştır. Tanıma işlemi birinci deneyde olduğu gibi 165 görüntü kullanılarak yapılmıştır. Sonuçlar Çizelge 3.2' de gösterilmiştir.

**Çizelge 3.2.** Deneklere ait normal ve merkezden aydınlanmış görüntülerin yer aldığı eğitim veritabanında yapılan test sonuçları(%'lik tanıma oranları).

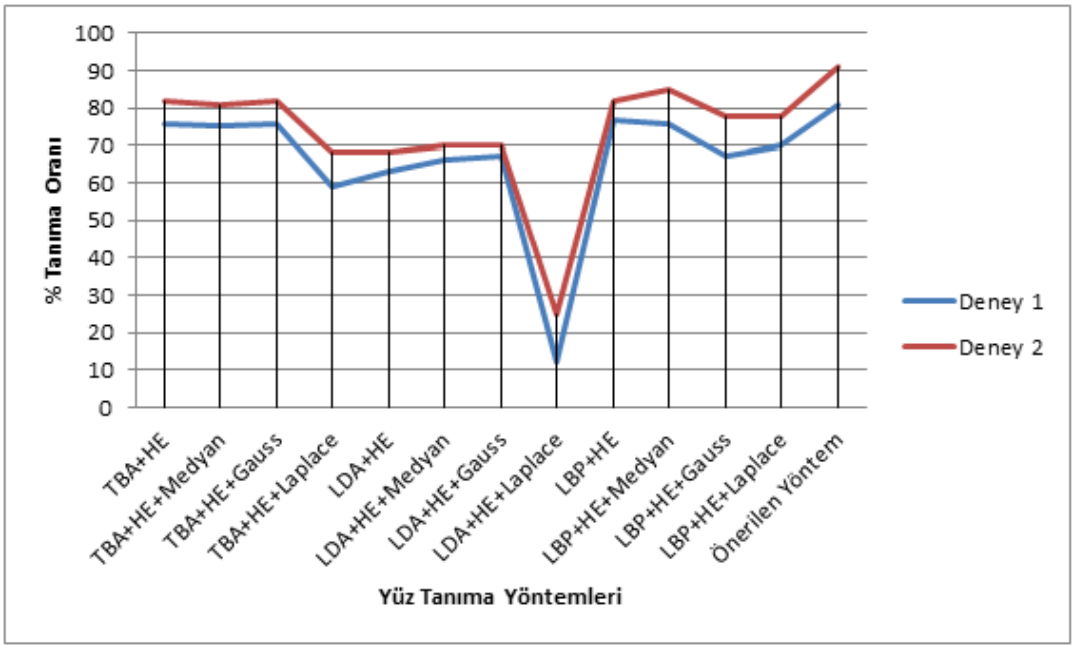
| Yöntemler | Filtreler | Yale Veritabanı | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Tümü | Merkezden Aydınlanmış | Sağdan Aydınlanmış | Soldan Aydınlanmış | Gözlüklü | Gözlüksüz | Normal | Üykulu | Şaşkın | Mutlu | Üzgün | Göz Kırpmış |
| TBA+HE | | 82 | 100 | 43 | 40 | 93 | 87 | 100 | 87 | 87 | 93 | 87 | 88 |
| | Medyan | 81 | 100 | 43 | 33 | 93 | 87 | 100 | 87 | 87 | 87 | 87 | 88 |
| | Gauss | 82 | 100 | 36 | 40 | 93 | 87 | 100 | 87 | 100 | 87 | 87 | 88 |
| | Laplace | 68 | 100 | 36 | 27 | 87 | 87 | 100 | 67 | 60 | 60 | 53 | 75 |
| LDA+HE | | 68 | 100 | 7 | 13 | 80 | 87 | 100 | 87 | 53 | 60 | 80 | 81 |
| | Medyan | 70 | 100 | 7 | 13 | 80 | 87 | 100 | 87 | 60 | 60 | 80 | 88 |
| | Gauss | 70 | 100 | 7 | 7 | 80 | 87 | 100 | 87 | 67 | 60 | 80 | 88 |
| | Laplace | 25 | 93 | 7 | 0 | 7 | 53 | 100 | 0 | 0 | 0 | 7 | 6 |
| LBP+HE | | 82 | 100 | 21 | 60 | 93 | 93 | 100 | 100 | 87 | 60 | 100 | 88 |
| | Medyan | 85 | 100 | 21 | 67 | 93 | 93 | 100 | 100 | 93 | 73 | 100 | 88 |
| | Gauss | 78 | 100 | 21 | 53 | 93 | 87 | 100 | 100 | 67 | 53 | 87 | 88 |
| | Laplace | 78 | 100 | 57 | 40 | 60 | 93 | 100 | 93 | 67 | 73 | 93 | 81 |
| Önerilen Yöntem | | 91 | 100 | 50 | 73 | 93 | 93 | 100 | 100 | 100 | 93 | 100 | 87 |

Çizelge 3.2 incelendiğinde en iyi tanıma yöntemi % 85 doğruluk oranına sahip LBP+HE+Medyan yöntemidir. Sistemin doğruluk oranı ise % 91 olup, % 6 daha iyi sonuç vermiştir.

Eğitici kaydı yapılan normal ve merkezden aydınlanmış görüntülerin sonuçları incelendiğinde LDA+HE+Laplace dışında tüm yöntemlerin yüz tanıma başarı oranlarının % 100 olduğu görülmektedir.

Her iki çizelgedeki değerler karşılaştırıldığında eğitici sayısı ile tanıma oranının doğru orantılı olduğu gözlenmiştir. Çizelge 3.3'te her iki çizelgedeki genel başarı tanıma oranları grafiksel olarak gösterilmiştir.

**Çizelge 3.3.** Deney 1 ve deney 2 sonuçlarının karşılaştırılması.




## 3.2. ORL Veritabanı Deneyleri

ORL veritabanı, 40 farklı kişiden alınan 10 farklı görüntü ile toplamda 400 görüntüden oluşan bir veritabanıdır. Veritabanı, 40 kişiden farklı zaman dilimlerinde farklı aydınlanma koşulları, farklı yüz ifadeleri, farklı yüz detaylarına bağlı olarak alınan görüntülerden oluşmaktadır. ORL veritabanından kullanılan bazı görüntüler Şekil 3.2'de gösterilmiştir.

**Şekil 3.2.** ORL veritabanından rastgele seçilen 3 bireyin görüntüleri.

ORL veritabanı deneylerinin birincisinde eğitim veri kümesine her kişiye ait 2 fotoğraf eklenmiş ve 40 test kümesinin her bir bileşene karşılık gelen toplamda 400 test verisinin tanıma işlemi yapılarak, doğru tanıma oranlarının ortalaması hesaplanmıştır. Deney sonuçları Çizelge 3.4'te gösterilmiştir.

**Çizelge 3.4.** ORL veritabanı deney sonuçları.

| **Yüz Tanıma Yöntemi** | **Filtre** | **Test Kümesi Doğru Tanıma Oranı (%)** |
|---|---|---|
| TBA+HE | | 73 |
| | Medyan | 74 |
| | Gauss | 75 |
| | Laplace | 33 |
| LDA+HE | | 44 |
| | Medyan | 49 |
| | Gauss | 51 |
| | Laplace | 5 |
| LBP+HE | | 75 |
| | Medyan | 77 |
| | Gauss | 73 |
| | Laplace | 69 |

Çizelge 3.4 incelendiğinde, yöntemlerin doğru tanıma verimliliği en yüksek olanı LBP+HE+Medyan 'dır. ORL veritabanı için fisher yaklaşımı ile yüz tanıma (LDA+HE) daha az başarılı olduğu söylenebilir.

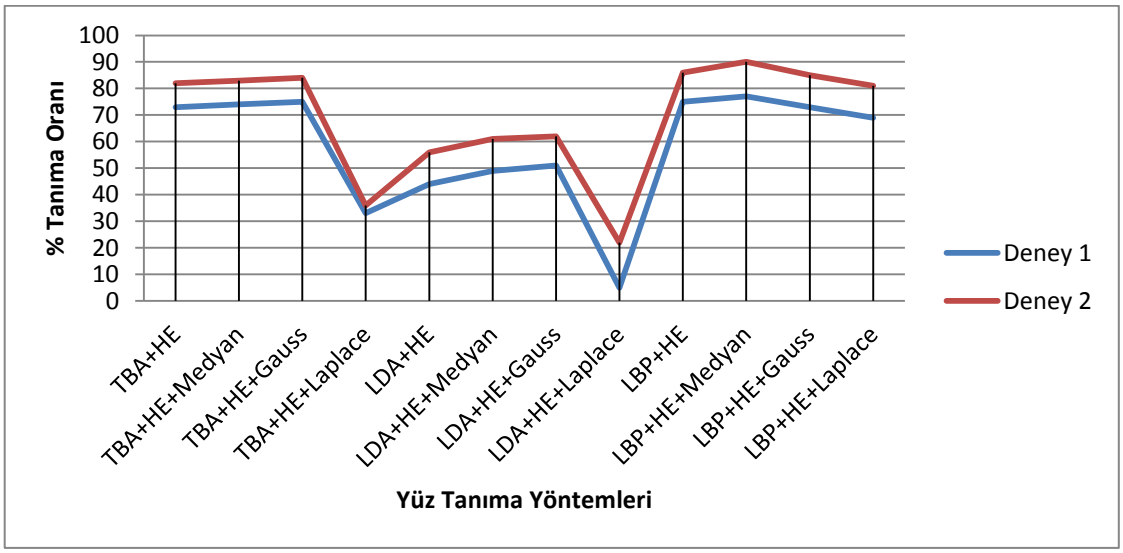
ORL veritabanı deneylerinin ikincisinde eğitim veri kümesine her kişiye ait 3 fotoğraf eklenmiş ve 40 test kümesinin her bir bileşene karşılık gelen toplamda 400 test verisinin tanıma işlemi yapılarak, doğru tanıma oranlarının ortalaması hesaplanmıştır. Deney sonuçları Çizelge 3.5'te gösterilmiştir.

**Çizelge 3.5.** ORL veritabanı deney sonuçları.

| Yüz Tanıma Yöntemi | Filtre | Test Kümesi Doğru Tanıma Oranı (%) |
|---|---|---|
| TBA+HE | | 82 |
| | Medyan | 83 |
| | Gauss | 84 |
| | Laplace | 36 |
| LDA+HE | | 56 |
| | Medyan | 61 |
| | Gauss | 62 |
| | Laplace | 22 |
| LBP+HE | | 86 |
| | Medyan | 90 |
| | Gauss | 85 |
| | Laplace | 81 |

Çizelge 3.5 incelendiğinde, bu deneyimizde de yöntemlerin en başarılı olanı LBP+HE+Medyan olarak karşımıza çıkmaktadır. Aynı zamanda tüm yöntemlerin tanıma oranlarında, test verisinin artışı ile birlikte bir artışın olduğu görülmektedir. Çizelge 3.4 ve çizelge 3.5'in sonuçları Çizelge 3.6'da karşılaştırılmıştır.

**Çizelge 3.6.** Deney 1 ve deney 2 sonuçların karşılaştırılması.



# 4. SONUÇLAR VE ÖNERİLER

Bu çalışmada günlük yaşamda kullanılabilecek bir yüz tanıma sistemi tasarımı hedeflenmiştir. Bu kapsamda geliştirilen uygulama 2 modül içermektedir. Bu modüllerden biri eğitim setleri oluşturulması ve yüz tanıma testlerinin yapılması amacıyla kullanılan modüldür. Diğeri ise video kamera üzerinde gerçek zamanlı yüz tanıması yapan modüldür.

Gerçek zamanlı tanıma sisteminde ise video kamera ile yakalanan görüntülerden yüz bölgesinin bulunması, bulunan bu bölgelerin kesilip çıkartılması ve tanıma işleminin gerçekleştirilmesi sağlanmıştır. Sistem, her bir yöntemin ayrı ayrı çalışabileceği şekilde tasarlanmıştır. Bu da kullanıcıya geniş bir seçim olanağı sunmakta, farklı yöntemlerin performanslarını izleme ve değerlendirme imkanı vermektedir. Böylelikle önerilen yöntem ortaya çıkmıştır. Yale veritabanı ile yapılan deneylerde önerilen yöntem ile sistemin tanıma başarı oranında % 6'ya kadar olan artışlar gözlenmiştir.

Gerçek zamanlı testlerde ise tanıma performansını; yüz bulma, ön işleme ve tanıma için geçen süreler doğrudan etkilemektedir. Bu çalışmada her bir yöntemin tanıma süreleri kayıt altına alınmaktadır.

Yüz ifadeleri, poz ve aydınlık gibi parametrelerde meydana gelen değişimler yüz tanıma oranını olumsuz yönde etkilemektedir. Bunlar hemen hemen bütün yüz tanıma sistemlerinin başlıca sorunlarıdır. Buna karşılık daha geniş kapsamlı yüz tanıma sistemlerinde, filtreler kullanılarak ışığın ayarlanması gibi önlemler ile bu sorunların etkileri azaltılmaktadır. Bu çalışmada da medyan, gauss, laplace filtreleri kullanılmıştır.

Yapılan deneylerde aynı kişiden alınan birden fazla resim olması durumunda performansın arttığı ve veritabanındaki farklı kişi sayısı azaltıldığı takdirde de performansın arttığı gözlemlenmiştir.

Tanıma işlemi için geçen süre, veritabanında bulunan görüntü boyutlarına ve toplam görüntü sayısına bağlı olarak değişmektedir. Çünkü, görüntülerin boyutları büyüdükçe veya toplam görüntü sayısı arttıkça, oluşan matrisler de büyümektedir. Buda tanıma hızını doğrudan etkilemektedir.

Uygulamanın geliştirilmesinde hazır EMGU CV hazır kütüphaneleri kullanılmıştır. Aynı zamanda çalışmada, tanıma başarım oranını artırmaya yönelik kodlama teknikleri uygulanmıştır. Hazır kütüphane kullanımı beraberinde bazı kısıtlamaları da getirmiştir. Buna rağmen % 91 'e ulaşan başarımlar elde edilmiştir. Literatürdeki yöntemlere göre başarılı olduğu görülmektedir.

Konunun araştırılması ve öğrenilmesi açısından faydalı bir çalışma olmuştur.

Bu tez kapsamında yapılan çalışmalar sonunda edindiğimiz bilgiler doğrultusunda, tanıma başarı oranını ve performansını artırmaya yönelik şunlar yapılabilir;

- ✓ Yüz tanıma algoritmalarını anlayarak, araştırmacı tarafından yazılması, kodlanması yüz tanıma sistemine uyarlanması
- ✓ Yüz tanıma sisteminin gerçek zamanlı çalışma hızını arttırma amaçlı optimizasyonların yapılması
- ✓ Gerçek zamanlı bir video görüntüsü üzerinde veya fotoğraf üzerinde bulunan yüzlere ait poz ve ışıklandırma durumlarını otomatik tespit ederek optimizasyonların yapılması

# 5. KAYNAKLAR


[1] Jain A. K., Ross A., Pankanti S., Biometrics: a tool for information security, *IEEE Transactions on Information Forensics and Security*, vol.1, no.2, **(2006)** 125-143.

[2] Tüzün S., Yüz tanıma sistemlerinde kullanılan öznitelik vektörlerinin optimizasyonu, *Yüksek Lisans Tezi*, İstanbul Üniversitesi, **(2010)**.

[3] Şan S., Parmak damar teknolojisi, *Yüksek Lisans Tezi*, Fırat Üniversitesi, **(2013)**.

[4] Yamuç A., Ölçek ve pozdan bağımsız yüz tanıma, *Yüksek Lisans Tezi*, Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü, **(2013)**.

[5] Lu X., Image analysis for face recognition, Dept. of Computer Science & Engineering, Michigan State Univesity, *Personal Notes*, **(2001).**

[6] Zhao W.Y, Chellappa R, *Image based face recognition: issues and methods*, Image Recognition and Classification, Editörler: Ed. B. Javidi, Mercel Dekker, **(2002)** 375-402.

[7] Smeets, D., Claes, P., Vandermeulen, D., Clement, J. G., Objective 3D face recognition : evolution, approaches and challenges, *Forensic Science International*, **(2010)** 125-132.

[8] Zhao W., Chellappa R., Phillips J., Rosenfeld A., 2003, Face Recognition in still and video images: A literature survey, ACM Computing Surveys, **(2003)** 399-458.

[9] Kanade T., Picture processing system by computer complex and recognition of human faces, *Dept. of Information Science*, Kyoto University, **(1973)**.

[10] Gençay S., Yüz tanıma işlemi ile personel takibi yapan bir yazılımın geliştirilmesi ve uygulanması, *Yüksek Lisans Tezi*, Muğla Üniversitesi, **(2012)**.

[11] Jafri R., Arabnia H. R., A survey of face recognition techniques, *Journal of Information Processing Systems*, **(2009)** 41-68.

[12] Sayar B., Face detection and recognition methods, *Yüksek Lisans Tezi*, Yeditepe Üniversitesi, **(2006)**.

[13] Ahonen T., Hadid A., Pietikäinen M., Face description with local binary patterns: application to face recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **(2006)**.

[14] Tenenbaum J. B., Silva V. de and Langford, J. C., A global geometric framework for nonlinear dimensionality reduction, *Science*, **(2000)** 2319–2323.

[15] Roweis, S. T. and Saul, L. K., Nonlinear dimensionality reduction by locally linear embedding, *Science*, (2000) 2323–2326.

[16] He, X., Yan, S., Hu, Y., Niyogi, P. and Zhang, H. J., Face recognition using aplacianfaces, *IEEE Trans. on Pattern Analysis and Machine Intelligence,* **(2005)** 328-340.

[17] Scholkopf, B., Smola, A. and Muller, K., Nonlinaer component analysis as a kernel eigenvalue problem, *Neural Computation*, **(1998)** 1299–1319.

[18] Mika S., Ratsch G., Weston J., Scholkopf B., Muller K. R., Fisher discriminant analysis with kernels, *Neural Networks for Signal Processing*, **(1999)** 41-48.

[19] Cevikalp, H., Neamtu, M. and Wilkes, M., Discriminative common vector method with kernels, *IEEE Trans. on Neural Networks*, 17 (5) **(2006)** 1150-1565.

[20] Gross R., Shi J., Cohn J., Quo vadis Face Recognition, Carnegie Melon University, **(2001)**.

[21] Donato G., Bartlett M. S., Hager J. C., Ekman P., Sejnowski T. J.,.Classifying Facial Actions, *IEEE Trans. PAMI*, **(1999)** 974-989.

[22] Tian Y., Kanade T., Cohn J. F., Recognition Action Units for Facial Expression Analysis, *IEEE Trans. PAMI*, **(2001)** 97-115.

[23] Martinez A., Recognizing Imprecisely Localized, Partially Occluded and Expression Variant Faces from a Single Sample per Class, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **(2002)** 748-763.

[24] Gross R., Shi J., Cohn J., Quo vadis Face Recognition, Carnegie Melon University, *Technical Report CMU-RI-TR-01-17*, **(2001)**.

[25] Kıymacı K., Yüz tanıma sistemi algoritmaların geliştirilmesi, *Yüksek Lisans Tezi*, Kocaeli Üniversitesi , **(2010)**.

[26] Turk M., Pentland A., Eigenfaces for recognition, *Journal of Cognitive Neuroscience*, **(1991)** 71–86.

[27] Anonim , http://isl.ira.uka.de/mmseminar/download/2004/ A_RothFaceRecognition.ppt.", (Erişim Tarihi: 4 Mart **2014**).

[28] Erdoğan A.Y., Yüz tanımada özyüz ve fisher algoritmaların incelenmesi, *Yüksek Lisans Tezi*, Ankara Üniversitesi, **(2010)**.

[29] Öz K., Yüz tanıma sisteminin paralel programlama ile optimizasyonu, *Yüksek Lisans Tezi*, Karabük Üniversitesi, **(2012)**.

[30] Gündüz H., Altuzay temelli yaklaşımlar kullanarak gerçek zamanlı yüz tanıma, *Yüksek Lisans Tezi*, Eskişehir Osmangazi Üniversitesi, **(2010)**.

[31] Tekeli E., Çetin M., Erçil A., Yerel ikili örüntü (LBP) ve önsel şekil bilgisi tabanlı bir desen bölütleme metodu, *IEEE SİU'07 Kurultayı* ,Sabancı Üniversitesi

[32] Shan Sh., Gao W., Cao B., Zhao D., Illumination normalization for robust face recognition against varying lighting conditions, *IEEE International Workshop on Analysis and Modeling of Faces and Gestures*, **(2003)** 157-164.

[33] S.Khurshed E., Yüz tanımada aydınlanmanın etkisinin uyarlanır histogram eşitleme ile azaltılması, *Yüksek Lisans Tezi*, Gazi Üniversitesi, **(2009)**.

[34] Kurt B., Nabiyev V. V., Dijital mamografi görüntülerinin kontrast sınırlı adaptif histogram eşitleme ile iyileştirilmesi, *TURKMIA'10 Proceedings VII. Ulusal Tıp Bilişimi Kongresi Bildirileri*, (**2010**).

[35] Taşçı M., FPGA kontrollü robotik göz, *Yüksek Lisans Tezi*, Balıkesir Üniversitesi, **(2011)**.

[36] Altuntaş C., Çorumluoğlu Ö, Filtreleme yöntemi ile digital görüntü zenginleştirme ve örnek bir yazılım, *Selçuk Üniversitesi Teknik Bilimler Meslek Yüksekokulu Teknik-Online Dergi*, 10 (1) (**2011**) 99-107

[37] Freund Y. and Schapire, R.E., A Decision-Theoretic generalization of on-line learning and an application to boosting, *Journal of Computer and System Sciences*, **(1997)** 119- 139

[38] Viola P., Jones M. J., Rapid object detection using a boosted cascade of simple features, *Proceedings of the 2001 IEEE Computer Society Conference* ,**(2001)** 511-518.

[39] Kuncan M., Ertunç H.M., Küçükyıldız G., Hızarcı B., Ocak H., Öztürk S., Görüntü işleme tabanlı zeytin ayıklama makinesi, *Otomatik Kontrol Ulusal Toplantısı*, Malatya, **(2013)** 459-464.

# ÖZGEÇMİŞ

***Kişisel Bilgiler***


Soyadı, adı : HOLAT, Recep

Uyruğu : T.C.

Doğum tarihi ve yeri : 18.05.1981 Düzce

Telefon : 05356811283

E-posta : recepholat@duzce.edu.tr


***Eğitim***

| **Derece** | **Eğitim Birimi** | **Mezuniyet tarihi** |
|---|---|---|
| Yüksek Lisans | Düzce Üniversitesi | |
| Lisans | Sakarya Üniversitesi | 2004 |
| Lise | Düzce Lisesi | 1999 |

***İş Deneyimi***

| **Yıl** | **Yer** | **Görev** |
|---|---|---|
| 2011- | Düzce Valiliği | Mühendis |
| 2011-2011 | Düzce Üniversitesi | Uzman |
| 2006-2011 | Standard Profil A.Ş. | Yazılım Uzmanı |

***Yabancı Dil***

İngilizce (KPDS : 62)

***Yayınlar***


**1.** Holat R., Kulaç S., Yüz Tanıma Sistemleri Kullanarak Kimlik Tanıma, *Signal Processing and Communications Applications Conference (SIU)*, (2014).