

**GÖMÜLÜ SİSTEMLERDE YÜZ TANIMA  
ALGORİTMALARININ KARŞILAŞTIRILMASI**

YÜKSEK LİSANS TEZİ

Mehmet Fatih ÖZDEMİR

DANIŞMAN

Doç. Dr. Uçman ERGÜN

İNTERNET VE BİLİŞİM TEKNOLOJİLERİ YÖNETİMİ

ANABİLİM DALI

Haziran 2018

Bu tez çalışması 16.FEN.BİL.30 numaralı proje ile Afyon Kocatepe Üniversitesi  
Bilimsel Araştırma Projeleri Koordinasyon Birimi tarafından desteklenmiştir.

**AFYON KOCATEPE ÜNİVERSİTESİ**  
**FEN BİLİMLERİ ENSTİTÜSÜ**

**YÜKSEK LİSANS TEZİ**

**GÖMÜLÜ SİSTEMLERDE YÜZ TANIMA**  
**ALGORİTMALARININ KARŞILAŞTIRILMASI**

**Mehmet Fatih ÖZDEMİR**

**Danışman**

**Doç. Dr. Uçman ERGÜN**

**İNTERNET ve BİLİŞİM TEKNOLOJİLERİ YÖNETİMİ**  
**ANABİLİM DALI**

**Haziran 2018**

## TEZ ONAY SAYFASI

Mehmet Fatih ÖZDEMİR tarafından hazırlanan "Gömülü Sistemlerde Yüz Tanıma Algoritmalarının Karşılaştırılması" adlı tez çalışması lisansüstü eğitim ve öğretim yönetmeliğinin ilgili maddeleri uyarınca 27/06/2018 tarihinde aşağıdaki jüri tarafından **oy birliği** ile Afyon Kocatepe Üniversitesi Fen Bilimleri Enstitüsü **İnternet ve Bilişim Teknolojileri Yönetimi Anabilim Dalı'nda YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

**Danışman** : Doç. Dr. Uçman ERGÜN

İmza

**Başkan** : Prof. Dr. Sabri KOÇER  
Necmettin Erbakan Üniversitesi Mühendislik Fakültesi



**Üye** : Doç. Dr. Uçman ERGÜN  
Afyon Kocatepe Üniversitesi Mühendislik Fakültesi



**Üye** : Dr. Öğr. Üyesi Gür Emre GÜRAKSIN  
Afyon Kocatepe Üniversitesi Mühendislik Fakültesi



Afyon Kocatepe Üniversitesi  
Fen Bilimleri Enstitüsü Yönetim Kurulu'nun  
...../...../..... tarih ve  
..... sayılı kararıyla onaylanmıştır.

.....  
Prof. Dr. İbrahim EROL  
Enstitü Müdürü

**BİLİMSEL ETİK BİLDİRİM SAYFASI**  
**Afyon Kocatepe Üniversitesi**

**Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmasında;**

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

**beyan ederim.**

**27/06/2018**

**Mehmet Fatih ÖZDEMİR**

## ÖZET

Yüksek Lisans Tezi

### GÖMÜLÜ SİSTEMLERDE YÜZ TANIMA ALGORİTMALARININ KARŞILAŞTIRILMASI

Mehmet Fatih ÖZDEMİR

Afyon Kocatepe Üniversitesi

Fen Bilimleri Enstitüsü

İnternet ve Bilişim Teknolojileri Yönetimi Anabilim Dalı

**Danışman:** Doç. Dr. Uçman ERGÜN

Günümüzde biyometri; insan hayatının nerdeyse vaz geçilmez bir parçası haline gelmeye başladı. Parmak izinden sese gözden yüz tanımaya kadar insanları birbirinden ayırtıran her konu biyometrik sistemlerin sahasını oluşturmaktadır. Bu sistemler güvenlik, haberleşme, pazarlama, savunma ve eğitim gibi birçok alanda kullanılmaktadır. Endüstri 4.0 ile birlikte gelen nesnelerin interneti kavramı gömülü sistemleri internete açmasıyla ilgili cihazlara erişimde güvenlik ve kişileri tanımlama ihtiyacını da beraberinde doğurmuştur. Akıllı evler ve giyilebilir teknolojiler güvenliği ön plana gelmesinin hızını artırmıştır. Artık akıllı evlere yerleştirilen gömülü cihazlar üzerine monte edilmiş kamera kişiyi tanıyarak tercihlerine göre cihaz hizmet verebilir. Örneğin televizyonun karşısına geçince kişisel tercihlere göre istenilen kanalları açabilir. Buna benzer sebeplerle sahibini tanıyan akıllı evlerin sayısı gün geçtikçe artmaktadır. Bu tezde OpenCV kütüphanesinin sunmuş olduğu yüz tanıma algoritmalarını gömülü ortam içerisine çalışmasını inceleyerek karşılaştırılmış ve hangi algoritmanın verimli olduğu araştırılmıştır.

**2018, x + 81 sayfa**

**Anahtar Kelimeler:** Yüz tanıma, Gömülü sistemler, openCV, Raspberry Pi

## **ABSTRACT**

M.Sc. Thesis

### **COMPARISON OF FACE RECOGNITION ALGORITHMS**

#### **IN THE EMBEDDED SYSTEMS**

Mehmet Fatih ÖZDEMİR

Afyon Kocatepe University

Graduate School of Natural and Applied Sciences

Department of Internet and Information Technologies Management

**Supervisor:** Assoc. Prof. Uçman ERGÜN

Today, biometry; it has become an indispensable part of human life. Every subject separating people from fingerprints to recognizing face to face is the field of biometric systems. These systems are used in many areas such as security, communication, marketing, defense and education. The concept of Internet of objects that came with Industry 4.0 brought about the need to define security and access to the devices related to the internet to open the embedded systems. Smart homes and wearable technologies have increased the speed of safety at the forefront. The camera is now installed on the embedded devices placed in smart homes, and the camera can be recognized by the camera. For example, when facing the television, it can open the channels according to personal preferences. Due to similar reasons, the number of smart homes that recognize the owner is increasing day by day. In this thesis, the face recognition algorithms presented by OpenCV library were examined by examining the operation of the embedded environment, and which algorithm was investigated.

**2018, x + 81 pages**

**Keywords :** Face recognition, Embedded systems, openCV, Raspberry Pi

## TEŐEKKÖR

Yüksek lisans eğitimin boyunca beni desteklemenin, bilgi ve deneyimleriyle çalışmanın her aşamasında beni yönlendiren, sonuçların değerlendirilmesi ve yazımı aşamasında yapmış olduđu büyük katkılarıyla tez çalışmasının ortaya çıkmasını sağlayan değerli danışmanım Sayın Doç. Dr. Uçman ERGÜN'e içten teşekkürlerimi sunarım.

Ömrümün son 14 yılına anlam kazanmasına sebep olan, yüksek lisans yapmam konusunda beni teşvik eden, tüm zorlu süreçlerimde yanımda olan ve en kadim dostum, hayat arkadaşım, eşim Nurgül ÖZDEMİR'e bu çalışmayı adayarak sonsuz teşekkür ediyorum.

Ayrıca bu tez çalışmasını 16.FEN.BİL.30 numaralı proje ile destekleyen Afyon Kocatepe Üniversitesi Bilimsel Araştırmalar Koordinasyon Birimine de teşekkür ederim.

Mehmet Fatih ÖZDEMİR  
AFYONKARAHİSAR, 2018

# İÇİNDEKİLER DİZİNİ

	<b>Sayfa</b>
ÖZET .....	i
ABSTRACT .....	ii
TEŞEKKÜR .....	iii
İÇİNDEKİLER DİZİNİ.....	iv
KISALTMALAR DİZİNİ .....	vi
ŞEKİLLER DİZİNİ .....	vii
ÇİZELGELER DİZİNİ.....	ix
RESİMLER DİZİNİ .....	x
1. GİRİŞ.....	1
2. LİTERATÜR BİLGİLERİ .....	4
2.1 Biyometri Nedir?.....	4
2.2 Biyometrik Sistemler.....	4
2.3 Biyometrik Sistemlerin Tarihi.....	8
2.4 Biyometrik Teknolojiler .....	9
2.4.1 İris Tanıma .....	9
2.4.2 Parmak İzi Tanıma .....	10
2.4.3 El Geometrisi Tanıma .....	11
2.4.4 Ses Tanıma.....	12
2.4.5 Retina Tanıma.....	13
2.4.6 Yüz Termogramı.....	13
2.4.7 DNA.....	14
2.5 Yüz Tanıma .....	15
2.5.1 Yüz Tanımanın Tarihi.....	17
2.5.2 Yüz Tanımanın Uygulama Alanları.....	19
2.6 Gömülü Sistemler.....	22
2.6.1 Gömülü Sistemlerin Tarihçesi .....	24
2.6.2 Gömülü Sistem Bileşenleri .....	25
3. MATERYAL ve METOT .....	27



3.1 Giriş.....	27
3.2 Sistemin Tanıtımı .....	27
3.3 Geliştirme Ortamı.....	27
3.3.1 Raspberry Pi.....	28
3.3.2 Raspberry Pi'nin Donanım Özellikleri.....	28
3.3.3 Raspberry Pi İşletim Sistemi.....	30
3.3.4 Python .....	32
3.3.5 Python IDE .....	33
3.4 OpenCV.....	35
3.5 OpenCV Algoritmaları.....	39
3.5.1 Eigenfaces (Özyüz).....	39
3.5.2 Fisherfaces .....	44
3.5.3 LBPH .....	48
3.6 Yazılımı Oluşturan Mimari ve Hazırlık .....	51
3.6.1 Kameradan Görüntü Alma.....	51
3.6.2 Kameradan Yüz Bulma.....	51
3.6.3 Kameradan Yüz Tanıma .....	53
3.6.4 OpenCv Haar Cascade Sınıflandırıcısı Eğitimi .....	55
3.6.5 Yüz Veri Setleri ve Eğitimi .....	58
4. BULGULAR .....	65
4.1 Yüz veri Setleri Eğitim Performansı .....	65
4.2 OpenCV Algoritmalarının Denekler ile Karşılaştırılması.....	65
4.2.1 Araştırma Bölgesi ve Denekler.....	65
4.2.2 EigenFace ile Test Bulguları.....	66
4.2.3 FisherFace ile Test Bulguları .....	68
4.2.4 LBPH ile Test Bulguları .....	70
4.2.5 Elde Edilen Bulguların Karşılaştırılması .....	72
5. TARTIŞMA ve SONUÇ .....	74
6. KAYNAKLAR.....	76
6.1 İnternet Kaynakları.....	79
ÖZGEÇMİŞ.....	81

## KISALTMALAR DİZİNİ

### Kısaltmalar

---

ARM	Acorn RISC Machine
CPU	Merkezi İşlem Birimi
CSI	Camera Serial Interface (Kamera seri arabirimi)
CV	Computer Vision
DAA	Doğrusal Ayırtaç Analizi
DSI	Display Serial Interface (Görüntü seri arabirimi)
GNU	Genel Kamu Lisansı
GPU	Grafik İşlem Birimi
GUI	Graphic User Interface (Grafik kullanıcı arayüzü)
HDMI	High Definition Multimedia Interface (Yüksek çözünürlüklü çokluortam arayüzü)
IDE	Integrated Development Environment (Tümleşik Geliştirme Ortamı)
INCITS	International Committee for Information Technology Standarts - Uluslararası Bilgi Teknolojileri Standartları Komitesi
IoT	Internet of Things (Nesnelerin interneti)
LBP	Yerel İkili Örüntü
LBPH	Yerel İkili Örüntü Histogramları
LCD	Liquid Crystal Display (Sıvı kristal ekran)
MB	Mega Byte
MHz	Mega Hertz
MLL	Machine Learning Library
RAM	Read Access Memory
TBA	Temel Bileşen Analizi
USB	Universal Serial Bus
UWP	Universal Windows Platform
Wi-Fi	Kablosuz Bağlılık
YSA	Yapay Sinir Ağı

---

## ŞEKİLLER DİZİNİ

	<b>Sayfa</b>
Şekil 2.1 Biyometrik sistem modeli. ....	6
Şekil 2.2 Örnek iris tanıma (İnt.Kyn.4).....	10
Şekil 2.3 Örnek parmak izi deseni (Chaudhary et al. 2014).....	11
Şekil 2.4 Örnek el tanıma deseni (Stergiopoulou and Papamarkos 2009) .....	12
Şekil 2.5 Ses tanımda karşılaştırma süreci Örneği (İnt.Kyn.5). ....	12
Şekil 2.6 Örnek retina taraması (İnt.Kyn.6). ....	13
Şekil 2.7 Yüz termogramı örneği (İnt.Kyn.7). ....	13
Şekil 2.8 Soldan sağa, A, B ve Z DNA sarmalları (İnt.Kyn.8). ....	15
Şekil 2.9 Yüz tanıma teknolojisine örnek yüz hattı (İnt.Kyn.9).....	16
Şekil 3.1 OpenCV bileşen yapısı (Kaehler and Bradski 2008). ....	35
Şekil 3.2 OpenCV Haar Cascade sınıfının çalışmasındaki mantıksal şekiller. ....	37
Şekil 3.3 Yüz üzerinde burnun belirlenmesi şekli (İnt.Kyn.12).....	38
Şekil 3.4 Haar Cascade Sınıflandırıcısı göz bölgesi belirleme (İnt.Kyn.12). ....	38
Şekil 3.5 TBA sınıflandırma 2D gösterim örneği (İnt.Kyn.15). ....	42
Şekil 3.6 NxN görüntü örneği ve vektörel dönüşümü (Turk and Pentland 1991).....	43
Şekil 3.7 DAA için sınıfları ayıran yön (İnt.Kyn.16).....	45
Şekil 3.8 Bir doku tanımlamak ve bir yerel ikili desen (LBP) hesaplamak için kullanılan üçlü komşu örneği (İnt.Kyn.23).....	49
Şekil 3.9 LBP operatörünün komşu veri tipleri (İnt.Kyn.14).....	50
Şekil 3.10 Örnek Haar Cascade XML dosyası içeriği. ....	58
Şekil 3.11 Raspberry Pi ile veri seti kaydı. ....	61
Şekil 3.12 OpenCV eğitim sonrası oluşan yml uzantılı dosya içeriği. ....	62
Şekil 4.1 Eigen algoritmasına göre yüz tanıma sonuçları. ....	67
Şekil 4.2 Eigenface algoritması denek sonuçları grafiği. ....	68
Şekil 4.3 Fisher algoritmasına göre yüz tanıma sonuçları. ....	69
Şekil 4.4 Fisherface algoritması denek sonuçları grafiği. ....	70

<b>Şekil 4.5</b> LBPH algoritmasına göre yüz tanıma sonuçları. ....	71
<b>Şekil 4.6</b> LBPH algoritması denek sonuçları grafiği.....	72
<b>Şekil 4.7</b> Tüm algoritmaların karşılaştırma grafiği. ....	72

## ÇİZELGELER DİZİNİ

	<b>Sayfa</b>
<b>Çizelge 2.1</b> Biyometrik Teknolojilerin Karşılaştırılması. ....	7
<b>Çizelge 4.1</b> Veri setindeki resim sayısına göre eğitim. ....	65
<b>Çizelge 4.2</b> Eigenface ile denek sonuçları tablosu. ....	67
<b>Çizelge 4.3</b> Fisherface ile denek sonuçları tablosu. ....	69
<b>Çizelge 4.4</b> LBPH ile denek sonuçları tablosu. ....	71

## RESİMLER DİZİNİ

	<b>Sayfa</b>
<b>Resim 2.1</b> Gömülü sistem bileşenleri (İnt.Kyn.3).....	25
<b>Resim 3.1</b> Raspberry Pi 3 model B (İnt.Kyn.10). ....	29
<b>Resim 3.2</b> Raspbian masaüstü görünümü. ....	30
<b>Resim 3.3</b> IEEE 2017 yılı Dünyanın en popüler yazılım dilleri sıralaması (Cass 2017). .....	33
<b>Resim 3.4</b> Raspbian içerisinde standart kurulu gelen Python 3.5.3 IDE'si ekran görüntüsü. ....	34
<b>Resim 3.5</b> Özyüz algoritması hayaletimsi yüz örnekleme (İnt.Kyn.13).....	40
<b>Resim 3.6</b> Özyüz hayaletimsi görüntüden normal resme dönüşüm görseli (İnt.Kyn.14). .....	41
<b>Resim 3.7</b> FisherFaces algoritması ile oluşturulmuş örneklem (İnt.Kyn.14). ....	46
<b>Resim 3.8</b> LBPH yöntemi ile elde edilmiş örnek görsel (İnt.Kyn.14). ....	49
<b>Resim 3.9</b> Haar Cascade sınıflandırıcısı için örnek pozitif resimler (İnt.Kyn.19).....	56
<b>Resim 4.1</b> Gönüllü deneklerin tek bir karede görünümü. ....	66

## 1. GİRİŞ

Son on yıl içinde, görüntü işleme ve analiz alanındaki gelişmelerle birlikte teknolojinin hızla gelişmesi biyometri konusunun popülerliğini artırmıştır. Hızla gelişen teknolojiyle birlikte güvenlik ihtiyacının da artması biyometrik sistemlerden birisi olan yüz tanıma sistemlerinin gelişmesine de katkıda bulunmuştur. Gerçek zamanlı görüntü işleme, makine öğrenme, yapay zeka gibi konular bu sürecin iyileşmesinde ve hızla ilerlemesinde etkin rol oynamaktadır.

Yüz bulma ve tanıma tekniklerinin ortam şartlarının uygun olmasıyla verimliliğinin arttığı gözükmemektedir. Gerçek zamanlı yüz bulma için geliştirilen teknoloji ve yazılımlar günümüzde bu konunun birçok saha da kullanılmasına olanak sağlamaktadır.

Mobil teknolojilerin hızla gelişmesi ve endüstri 4.0 ile gelen yeni teknolojiler gömülü sistemlerde güvenlik kaygılarının artmasına neden olmuştur. Nesnelerin interneti adı verilen bu dünyada veri alışverişi ve güvenliği git gide zaruri ihtiyaç haline gelmiştir. Gömülü sistemlerde güvenlik, klasik yollardan biri olan bir klavye veya numarator üzerinden el ile giriş gibi olgular artık pratikliğini yitirmesinin yanı sıra kopyalanmasının önüne geçilememiştir. Bu yüzden insanoğlu güvelliği eşsiz olan ve kolay kullanılabilir ihtiyacından dolayı biyometrik sistemlere başvurmayı tercih etmiştir.

Teknolojinin giderek ilerlemesi gömülü sistemlerinde işlemci ve bellek kapasitesini de artırarak gelişmesini sağlamıştır. Görüntü işleme, derin öğrenme konuların da teknoloji ile birlikte gelişmesi biyometrik alanda kullanılan algoritmalarında kalitesinin artmasına olanak sağlamıştır. Teknolojik cihazlar hızlanmanın yanında git gide küçülmüşlerdir.

Gömülü sistemlerin en yaygın olanlarından mobil telefonlara kadar giren bu teknolojik yenilikler biyometri konusunda gelişmelerin hızlanmasına veya var olan algoritmaların daha verimli hale gelmesini sağlamıştır. Öyle ki parmak izi okuyucularının daha hassas yapıda çalışabilir bir şekilde mobil telefonlarda nerdeyse tümünde artık bir standart olarak gelmiş olması bu sürecin bir parçasıdır.

Günümüzde görsel olguların internet ortamı içerisinde erişebilir olması, güçlü işlemcili yeni nesil mikro işlemciler, GPU adı verilen grafik işlemcilerin gücünün artması yüz bulma ve tanıma sistemlerinin uygulanabilirliğini arttırmıştır. Yeni nesil kamera ve gece görüş tekniklerinin oldukça küçülmesi bu işlemler için mobilize ortamlar oluşturmuştur.

Son yıllarda gelişen güvenlik ürünleri de bu gelişmelere kayıtsız kalmayarak yüz tanımlamalı eve giriş veya sizi tanıyan odalara kadar yaygınlaşmaya başlamıştır. Bu süreçte şüphesiz nesnelerin internetinin gömülü ortamlara etkisinin ve teknolojik gelişmelerdeki ilerlemenin yanı sıra teknolojinin ucuzlaması da bu süreci hızlandırmıştır.

Nesnelerin internetini ile gömülü ortamlarda güvenlik gereksinimi ortaya çıkması yazılım dünyasını da gelişmelere kayıtsız kalmamasını dağlamıştır. Çeşitli firmaların yanı sıra üniversitelerin bu konuda çeşitli çalışmaları gün geçtikçe gelişmiş mobil ortamlar dahil gömülü ortamlara kadar hemen hemen tüm akıllı cihazlara yönelik yazılımlar geliştirmişlerdir. Bu yazılımlar gerek internet üzerinde servis tabanlı gerekse gömülü ortamlarda kullanılmak üzere kütüphaneler şeklinde ortaya çıkmıştır.

Bu tez çalışmasında gömülü ortamlarından biri olan Raspberry Pi 3 teknolojisinden yararlanılarak OpenCV kütüphaneleri içerisinde bulunan Eigenface, Fisherface ve LPBH algoritmalarının performansı karşılaştırılacaktır.

Araştırmada OpenCV kütüphanesinin son ve en güncel algoritması olan LPBH algoritmasının çeşitli ışıktaki ortamlarda performansı karşılaştırılmış ve karşılaştırma sonuçları incelenecektir.

Ayriyeten gömülü ortamlarda OpenCV kütüphanesinin yüz tanıma algoritmaları için gerekli veri setlerinin hazırlanması ve performansı konusuna değinilecektir. Veri setlerinin hazırlanması ve özellikle LPBH algoritmasının gereksimi olan eğitim modelinin nasıl yapılandırıldığı konusunda çalışma yapılacaktır. Gerek veri setlerinin oluşturulması gerekse LPBH algoritmasının ihtiyaç duyduğu eğitim modelinin oluşturulma sürecinde Raspberry Pi 3 ortamını teknolojik olarak performansı konusuna



bakılacaktır.

Karşılaştırmanın Raspberry Pi 3 IoT cihazı için tasarlanmış Linux GNU lisanslı Raspbian işletim sistemi ile yapılması düşünüldü. Linux tabanlı mimarilerde verimli çalışan ve Raspbian içerisinde editörü ile birlikte kurulu gelen Python dili kullanılarak yapılması planlandı.

## 2. LİTERATÜR BİLGİLERİ

### 2.1 Biyometri Nedir?

Biyometri: İnsanı otomatik tanıma veya tanımlama amacıyla gözündeki, yüzündeki, parmak izindeki, sesindeki insanın kendisine has çeşitli kişisel özelliklerinin bir takım uygulama ve desenleri sayısal verilere dönüştürülmesine denmektedir (İnt.Kyn.1). Özetle, biyometri kişinin ölçülebilir biyolojik izlerini ifade etmektedir. Biyometri, bireyleri etiketlemek ve tanımlamak için kullanılan ayırt edici, ölçülebilir niteliklerdir. Genellikle davranışsal özelliklerine karşı fizyolojik olarak kategorize edilir. Fizyolojik özellikler vücudun şekli ile ilgilidir. Parmak izi, avuç içi damarları, yüz, DNA, avuç içi, el geometrisi, iris, retina gibi örnekler fizyolojik özellikleri oluşturmaktadır. Bazı araştırmacılar, diğer taraftan biyometriyi tanımlamak için davranış stratejisini tanımlamışlardır. Davranışsal özellikler, ritim, yürüyüş ve sesin yazılması dahil; ancak bunlarla sınırlı olmamak kaydıyla, bir kişinin davranış modeliyle ilgilidir (Jain and Ross 2008).

Tarihsel olarak ilk Antik Babil’de parmak iziyle kil tabletler üzerinde ticari işlemler için kullanılmıştır. Yakın geleceğe kadar el ve parmak izlerinden, anlaşma, onaylama gibi ticari veya itibari işler için faydalanılmıştır. Fakat 1891 yılında Juan Vucetich’in Arjantin’de suçluların parmak izlerini karşılaştırmak için ilk defa katalog oluşturmaya başlaması bilinmektedir (İnt.Kyn.2).

### 2.2 Biyometrik Sistemler

Biyometrik sistemlerin basit halleri ile binlerce yıl önceden beri kullanıldığı bilinmektedir. Yakın zamanda ise araştırmacıların insanların fiziksel özellikleri ve karakteristiklerinin suça eğilimleri ile bir ilgisinin olup olmadığını araştırmaları biyometri alanına ilgiyi arttırmıştır.

Günümüzde biyometrik incelemelerin boyutu, çeşitliliği ve kullanım alanları artmıştır. Bu sayede pek çok yeni biyometrik kimlik doğrulama sistemi yerini almıştır. Doğrulama sistemleri, veri tabanında daha önceden kaydedilen bir şablon ile

karşılaştırma yapılarak kişinin doğru kişi olup olmadığının doğrulanmasını hedeflemektedir. Bu sistemler erişim kontrolü, güvenlik takip gibi amaçlara hizmet etmek üzere farklı uygulama alanlarına sahiptir. Bu uygulamalar da şüphesiz güvenlik önemli bir konuya sahiptir. Tanıma sistemlerinde ise veri tabanında yer alan tüm şablonlar ile örnek veri karşılaştırılarak, biyometrik özellikten kimlik tespiti yapılmaya çalışılmaktadır (Prabhakar *et al.* 2003).

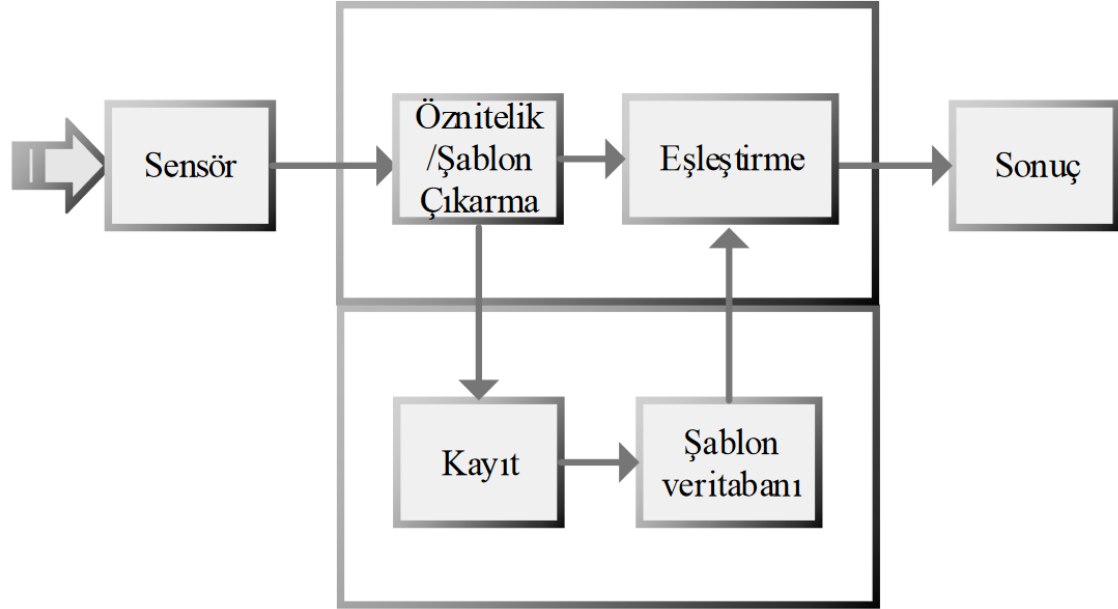
Biyometrik sistemlerin uygulama alanları günümüzde oldukça çeşitlidir. Özellikle havaalanları giriş ve çıkış işlemleri, kredi kartı uygulamaları, kriminal amaçlı teşhis ve tespit uygulamaları, sigorta şirketleri, ağ ve veri güvenliği, sosyal güvenlik, vergi süreçleri gibi kamu hizmetleri, e-ticaret, elektronik imza uygulamaları, internet bankacılığı, ATM'ler, çağrı merkezleri, personel takibi, hasta takibi bu gibi sosyal sistemlerde kullanılmalarının yanında artık, bilgisayarlar, PDA olarak adlandırılan el bilgisayarları, cep telefonları ve ev kilit sistemlerinde de kullanılmaktadır.

Örneğin parmak izi, iris veya yüz tanıma sistemini barındıran bir bilgisayar, kimliğini doğrulayamayan kullanıcıların bilgileri açmasına ve işlem yapmasına izin vermemektedir. Yüz, el geometrisi ve iris tanınması ise son yıllarda ilginin artmasıyla kullanılmaya başlanması üzerine, özelliklerin işlenmesini kolaylaştıran önemli bir etkidir.

Biyometri uygulayıcılarının genel amacı: Kişilerin kimliklerini doğrulayabilmeleri için, akıllarında tutmaları gereken herhangi bir bilgi ya da yanlarında taşımak, kaybetmemek ya da unutmamak zorunda oldukları kart, anahtar gibi araçların yerine; kopyalanması ya da taklit edilmesi imkânsız olan özelliklerini kullanmalarını sağlamaktır. Biyometrik sistemlerde, kimlik belirleme işlemi, kişilerin fiziksel ya da davranışsal özelliğine dayanarak gerçekleştirildiği için başkasına devredilmesi, unutulması ya da kaybedilmesi durumu söz konusu değildir.

Basit bir biyometrik sistem temel olarak dört adımdan oluşacak şekilde incelenebilmektedir. Bu adımlar biyometrik verilerin bir algılayıcı aracılığıyla sisteme alınması, alınan veriden öznitelik vektörlerinin elde edilmesi, elde edilen öznitelik

vektörlerinin daha önce yapılan bir kayıt işlemi ile veri tabanlarına kaydedilen şablonlar ile eşleştirmesi ve yapılan eşleştirme sonucunda alınan skora göre sisteme erişim kararının oluşturulmasıdır (Delac and Grgic 2004).



Şekil 2.1 Biyometrik sistem modeli.

Biyometrik sistemler diğer yöntemlere göre çok daha az riske sahiptir. Ancak sistemlerin oluşturulabilmesi için bazı standart ölçüler kullanılmalıdır.

Biyometrik ölçüler olarak adlandırılan bu ölçülerin şifrelerde kullanımı için INCITS (International Committee for Information Technology Standards-Uluslararası Bilgi Teknolojileri Standartları Komitesi) tarafından oluşturulmuş uluslararası bir standart mevcuttur (Şamlı ve Yüksel 2009).

Tüm biyometrik sistemler aşağıda açıklanmış olan beş özelliğe sahip olmalıdır (Chellappa *et al.* 1995).

1. Evrensellik: Tüm bireyler biyometrik özelliklere sahip olmalıdır.
2. Eşsiz olma: Biyometrik karakteristiğın her insanda farklı bir şekilde yer alması.
3. Süreklilik: Karakteristiğın zamanla değışmemesi.
4. Elde edilebilirlik: Biyometrik özelliklerin bazı pratik cihazlarla ölçülebilir

olması.

5. Kabul edilebilirlik: Bireylerin biyometriğinin ölçüm ve toplanmasında itirazları olmamalı.

Biyometride yukarıda adı geçen özelliklerin dışında bir takım bilim insanları kulak, yüz, yüz ısı dağılımı, el ısı dağılımı, el geometrisi, DNA, parmak izi, iris, retina, imza, ses gibi biyometrik karakteristiklerini genel anlamda aşağıdaki tabloda karşılaştırmışlardır (Jain *et al.* 2007).

**Çizelge 2.1** Biyometrik Teknolojilerin Karşılaştırılması.

Biyometrik Karakteristik	Evrensellik	Eşsizlik	Süreklilik	Elde Edilebilirlik	Performans	Kabul Edilebilirlik	Yaygınlık
<b>DNA</b>	Y	Y	Y	D	Y	D	D
<b>Kulak</b>	O	O	Y	O	O	Y	O
<b>Yüz</b>	Y	D	O	Y	D	Y	Y
<b>Yüz Termogramı</b>	Y	Y	D	Y	O	Y	D
<b>Parmak İzi</b>	O	Y	Y	O	Y	O	O
<b>El Geometrisi</b>	O	O	O	Y	O	O	O
<b>İris</b>	Y	Y	Y	O	Y	D	D
<b>Retina</b>	Y	Y	O	D	Y	D	D
<b>İmza</b>	D	D	D	Y	D	Y	Y
<b>Ses</b>	O	D	D	O	D	Y	Y

**Y: Yüksek    O: Orta    D: Düşük**

Çizelge 2.1’de görüldüğü gibi biyometrik karakteristikler evrensellik, eşsizlik, süreklilik, elde edilebilirlik, performans, kabul edilebilirlik ve yaygınlık olmak üzere sahip olmaları gereken özellikler açısından karşılaştırılmıştır. Karşılaştırma sonuçlarında elde edilebilirlik ve kabul edilebilirlik önemli bir yere sahiptir. Elde

edilebilirlik, her yerden ölçümleyebilen cihazlar ile pratik elde edilebilirlik artık günümüzde ana etken olmuştur. Kabul edilebilirlik ise elde edilebilirlik ile elde edilen sonuçların herkesçe itiraz edilmeden onaylanmasıdır. Bu iki kritere yüksek cevap veren yüz ve imza olmak üzere iki biyometrik karakteristik bulunmaktadır. İmza, doğrudan elde edilebilen olmayıp, el hareketlerinin koordinasyonu ile eylemsel hareket sonrası elde edilebilen davranışsal biyometrik sistemlere girmektedir. Yüz ise fizyolojik özelliğe dayalı olduğundan genel anlamda kabul edilmektedir.

### **2.3 Biyometrik Sistemlerin Tarihi**

Günümüz dünyasında yaygın kullanımı olan biyometri teknolojisinin mantalite olarak geçmişe dayanan bir mazisi vardır. Biyometrik sistemlerin basit halleri ile binlerce yıl önceden beri kullanıldığı bilinmektedir. Şöyle ki eski Mısır'da bu teknoloji günlük hayatta yaygın olarak kullanılmıştır. Keza antik mısır kaynaklarında insanların kendilerine has özellikleri olan göz, saç, kaş vb. özellikler insanları nitelendirmede kullanıldığı belirlenmiştir. Birçok yöntemin tarımsal süreçler başta olmak üzere hukuki süreçlerde de kullanıldığı yine incelenen kaynaklarda görülmüştür. Yine geçen yüzyılda kriminalistlerin; fiziksel özelliklerinin insanların kriminal eğilimler noktasında etkisi olup olmadığı konusundaki araştırmaları dikkat çekmiştir. Bu çalışmalar neticesinde birçok alet, edevat ve ölçüm ortaya çıkmıştır. Elde edilen ölçümlerin doğruluğu tartışılmakla birlikte özellikle parmak izi kolluk kuvvetleri tarafından kullanılmaya başlanmıştır ve giderek kabul görmüştür. Fakat bu durum her ne kadar tartışmalı bir konu olsa da kolluk kuvvetlerince yaygın kullanımı olmuş ve halen günümüz dünyasında da kullanılmaktadır. Diğer yandan biyometrik özelliklerin bilişim sistemleri tarafından otomatize edilmiş olarak tanıma meselesi ticari ve askeri alanda yıllarca ciddi araştırma konusu haline gelmiştir. Bu alanda birçok proje yapılmıştır. Örneğin el geometrisi okuyucusu bunlardan bir tanesidir. Fazla başarılı olmamakla birlikte bu cihaz daha başka çalışmaların yapılması konusunda araştırmacıları cesaretlendirmiş ve yeni çalışmalara yönlendirmiştir. Sonuç olarak bu çalışmalar meyvesini vermiş, değişik firmalar bu alanda değişik ürünler pazara sürmüş ve bu sektörün mihenk taşı haline gelmiştir. Diğer yandan parmak izi alanında da çalışmalar yapılmış ve ciddi anlamda ilerleme kaydedilmiştir.

Biyometrik sistemlerin uygulama alanları günümüzde oldukça çeşitlidir. Özellikle havaalanları giriş ve çıkış işlemleri, kredi kartı uygulamaları, kriminal amaçlı teşhis ve tespit uygulamaları, sigorta şirketleri, ağ ve veri güvenliği, sosyal güvenlik, vergi süreçleri gibi kamu hizmetleri, e-ticaret, elektronik imza uygulamaları, internet bankacılığı, ATM'ler, çağrı merkezleri, personel takibi, hasta takibi bu gibi sosyal sistemlerde kullanılmalarının yanında artık dizüstü ve tablet bilgisayarlar da cep telefonları ve ev kilit sistemlerinde de kullanılmaktadırlar (Yozgat 2004).

Örneğin parmak izi, iris veya yüz tanıma sistemini barındıran bir bilgisayar, kimliğini doğrulayamayan kullanıcıların bilgileri açmasına ve işlem yapmasına izin vermemektedir. Yüz, el geometrisi ve iris tanınması ise son yıllarda ilginin artmasıyla kullanılmaya başlanması üzerine, özelliklerin işlenmesini kolaylaştıran önemli bir etkidir.

İçinde bulunduğumuz yüzyıl başta olmak üzere özellikle doksanlardan itibaren yüz tanıma ve retina tanıma başta olmak üzere temassız tanıma teknolojileri ağırlık kazanmıştır. Son çeyrek yüzyıllık dönemde birçok üretici bu alanda uzmanlığını artırmış ve daha iyi bir pazar payına sahip olmak için yeni teknolojiler ve metodolojiler üzerinde çalışmalar yapılmıştır.

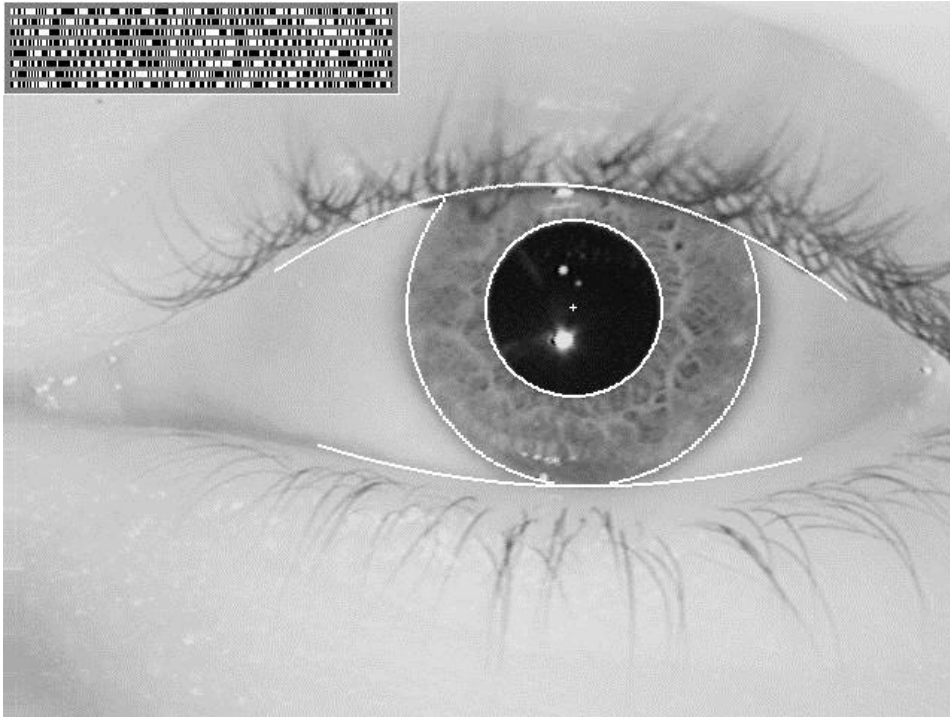
## **2.4 Biyometrik Teknolojiler**

Biyometrik sistemlerde özellikle kimlik doğrulamada birçok teknoloji kullanılmakla birlikte birkaç teknoloji aşağıda yer almaktadır. Günümüzdeki birçok biyometrik tanıma sistemi olmakla birlikte bunlardan başlıcaları iris, parmak izi, retina, DNA, damar, yüz, el geometrisi, ses, yüz termogramıdır.

### **2.4.1 İris Tanıma**

İris tanıma, mesafe ve ortam koşullarının önemli olduğu temassız bir tanıma teknolojisidir. İris karakteristikleri kısa bir mesafeden tanınabilmektedir. İris bölümü

gözün renkli kısmında yer almakta olup insandan insana değişkenlik göstermektedir. Öyle ki aynı insanda bile iki gözün iris desenleri farklılık arz etmektedir. Yine diğer biyometrik teknolojilerde olduğu gibi bu teknolojide de iris deseni sayısal hale getirilip veri tabanında saklanmakta ve karşılaştırma yapılmaktadır. İris tanıma bu alandaki teknolojiler arasındaki en güvenli olanıdır. Genel olarak parmak izi tanımaya benzetilen bu sistemin, parmak izine göre en önemli avantajı, parmak izi kullanılan biyometrik sistemlerde 60 veya 70 karşılaştırma noktası bulunurken, iris taramada karşılaştırma için yaklaşık 200 referans noktası kullanılmasıdır.



Şekil 2.2 Örnek iris tanıma (İnt.Kyn.4).

#### 2.4.2 Parmak İzi Tanıma

Bu teknolojide birçok yöntem bulunmaktadır. Çünkü geçmişten gelen yaygın kullanımı söz konusudur. Bilinen yaklaşımlardan en bilineni kolluk kuvvetlerinin kullandığı yöntemlerdir. İlk kullanılmaya başlandığı yıllardan bu yana gerek yazılım gerekse donanım alanında parmak izi sistemlerinde önemli ilerlemeler kaydedilmiştir. Deri tabakası, parmak uçlarında, vücudun diğer noktalarından farklı olarak pürüzlü, çukur ve tümseklik arz eden bir şekil alır (İnt.Kyn.5). Parmak uçlarındaki üst deri tabakasının



kendine has bir desen yapısı vardır. Parmak ucu ile düz bir zemine basıldığında, parmak ucunda yer alan desen yapısı yüzeyde kendisi ile aynı desende bir iz bırakır. Bu iz, parmak izi olarak adlandırılmaktadır. Bu nedenledir ki diğer yöntemlere göre daha çok cihaz ve çeşidi söz konusudur. Fakat parmak izine dayanan tanımda daha etkili yöntemler söz konusudur.

Mesela daha hassas ölçüm sağlayan termal görüntüleme bunlardan biridir. Keza bu yöntemde parmak ısısından faydalanılmaktadır. Parmak bir sensörden geçirilir ve parmak ısısının izleri okunarak kayıt altına alıp kıyaslama yapılır. Başka yöntem de parmak izi elektrik akımından yararlanarak yapılan yöntemdir. Bu yöntem birçok alanda kullanılmaktadır.

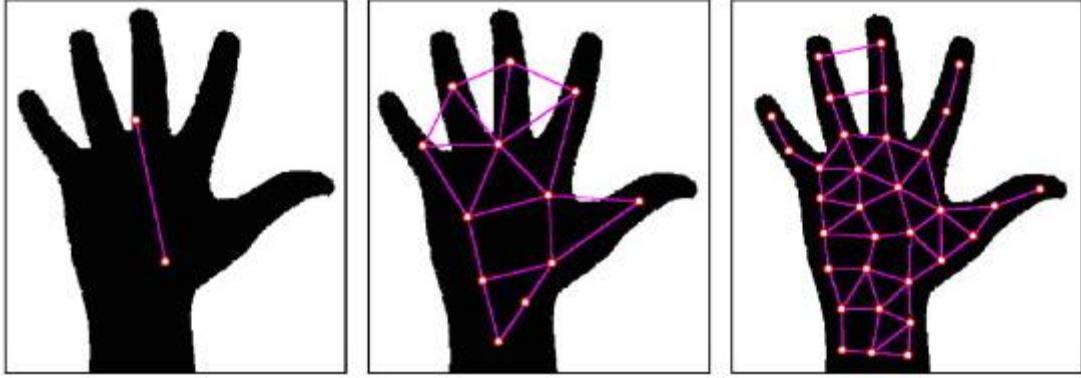


Şekil 2.3 Örnek parmak izi deseni (Chaudhary *et al.* 2014).

### 2.4.3 El Geometrisi Tanıma

Bu yöntem de her insana özgü olan el uzuv oranlarından yararlanarak yapılan tanımadır. Kişinin elinin veya kullanılan sisteme göre iki parmağının geometrik yapısı analiz edilir. Tabi bu tanıma işlemi elin üç boyutlu bir görüntüsü elde edilerek gerçekleştirilir. Tanıma işleminde ise el bir tarayıcıdan geçirilerek üç boyutlu bir profili elde edilerek veri tabanındaki diğer kayıtlar ile karşılaştırmak suretiyle gerçekleştirilmektedir. Bu

teknolojinin kullanımı ve uygulanması diğer teknolojilere göre daha rahattır ve diğer yöntemlere nazaran entegrasyonu daha kolay olması sebebiyle kullanım alanı yaygınlaşmıştır.



Şekil 2.4 Örnek el tanıma deseni (Stergiopoulou and Papamarkos 2009)

#### 2.4.4 Ses Tanıma

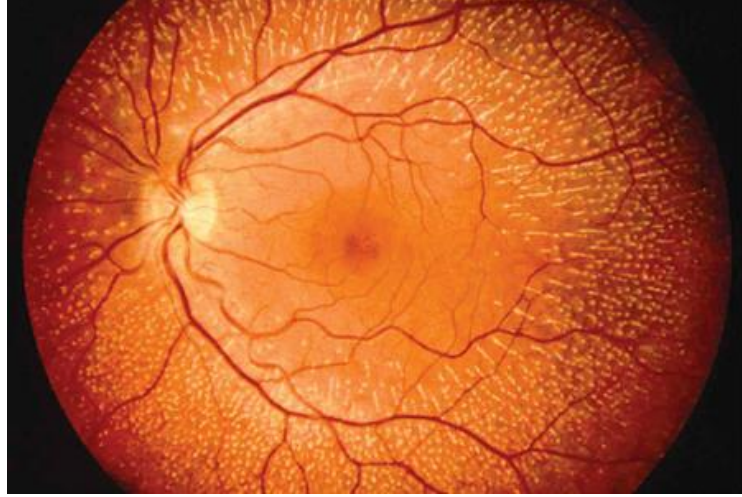
Hayatımızda sesin ne kadar kullanıldığı konusu düşünüldüğünde etkili bir tanıma yöntemi olarak düşünülse de seslerin kolay taklit edilebilme özelliği nedeniyle güvenliği zayıftır. Fakat insan sesi birçok yönden spesifik özellikler içermesi nedeniyle ayırt edici bir yönü bulunmaktadır. Bu alanda birçok yöntem olmakla birlikte ses frekansları küçük birimler olarak sınıflandırılıp karakterize edilerek veri tabanına kaydedilir.



Şekil 2.5 Ses tanımda karşılaştırama süreci Örneği (İnt.Kyn.5).

#### 2.4.5 Retina Tanıma

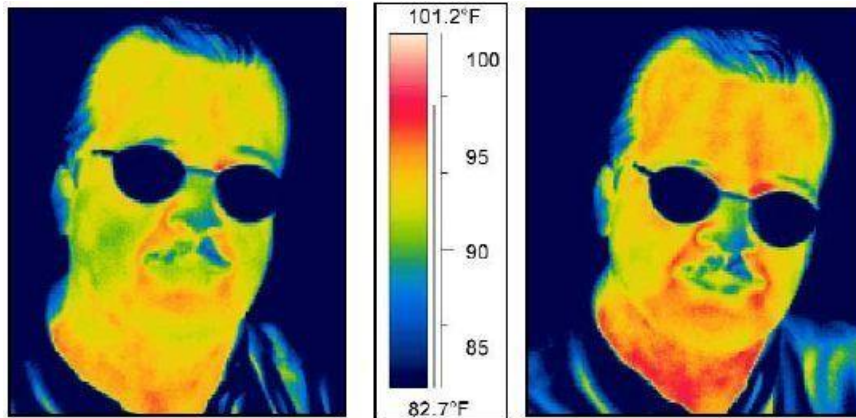
Gözün kendine has optik yapısı itibariyle kullanımı zordur. Bu yapının taranması mantığı; belirli bir noktaya bakma mecburiyeti nedeniyle, güvenlik seviyesi iyi olmasına rağmen kullanım zorluğu olmasındır.



Şekil 2.6 Örnek retina taraması (İnt.Kyn.6).

#### 2.4.6 Yüz Termogramı

Çok yaygın olmasa da kullanılan teknolojilerden biridir. Yüzün ısı haritasının analizi yapılarak kimlik tespiti ve doğrulama işlemlerinde kullanılan biyometrik yöntemlerden biridir. Yüzün ısı haritasının değişiklik göstermesi sık rastlanılan sıkıntılarındandır.

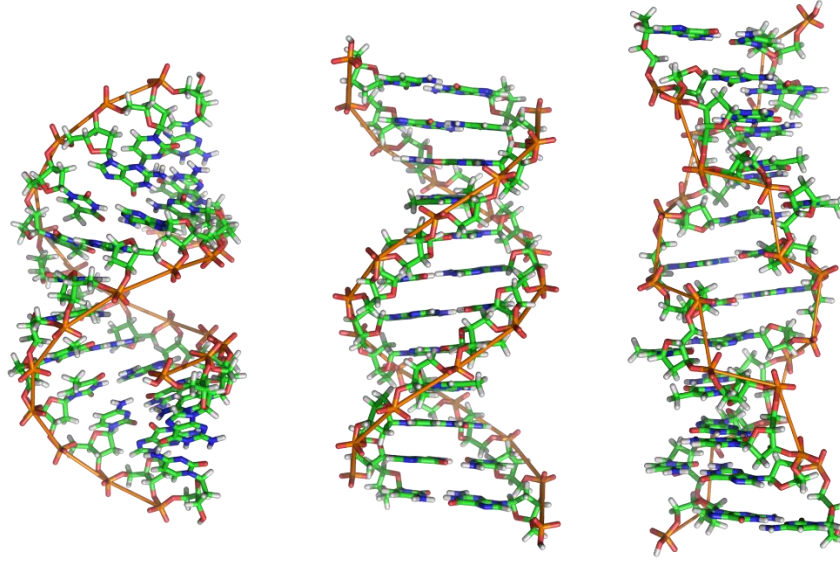


Şekil 2.7 Yüz termogramı örneği (İnt.Kyn.7).

#### 2.4.7 DNA

DNA (Deoksiribonükleik Asit), yapısının kişiye özgü bir karakteristik olduğu ilk olarak 1985 yılında keşfedilmiştir. DNA tanıma günümüzde en güvenilir kimlik doğrulama yaklaşımlarından biridir ve en çok babalık testleri ve adli işlemlerde kullanılmaktadır. DNA tanımada saç, kan ve diğer biyolojik materyaller incelenmektedir. Yöntemde hücre nükleuslarındaki kromozomlarda saklanan DNA molekülleri kullanılmaktadır. Her bir DNA molekülü 4 kimyasal birim içerir. Bunlar: Adenin (A), Guanin (G), Sitozin (C) ve Timin (T) olarak sıralanabilir. Bu yapılar şeker molekülleri ve fosfat grupları ile birlikte pozisyon ve diziliş açısından genetik bilginin kodlandığı çatıyı oluştururlar.

DNA'nın çeşitli biçimleri (konformasyonları) mevcuttur. Ancak, canlılarda sadece A-DNA, B-DNA, ve Z-DNA gözlemlenmiştir. Bu üç biçimden "B" biçimi, hücrelerde bulunan şartlar altında en sık görülenidir. DNA dizilerinin depolanması ve aranması için özellikle dizi arama algoritmaları, makine öğrenimi ve veritabanı teorisi konularında yöntemlerin geliştirilmesi sayesinde bilgisayar bilimlerinde önemli ilerlemeler kat edilmiştir. Dizi arama ve eşlendirme algoritmaları harflerden oluşan uzun diziler içinde daha kısa harf dizilerinin bulunmasıyla ilgilidir, bunlar belli nükleotit dizilerinin bulunması için geliştirilmiştir (İnt.Kyn.8).



**Şekil 2.8** Soldan sağa, A, B ve Z DNA sarmalları (İnt.Kyn.8).

## 2.5 Yüz Tanıma

Yüz tanıma yıllardan beri üzerinde çalışılan ve daha iyi sonuçlar elde etmek için çaba sarf edilen bir tanıma sistemidir. İnsan tanıma o insana ait eşsiz özelliklerin kullanılarak otomatik olarak tanıma işlemidir. Yüz tanıma hakkında birçok yaklaşım olmakla birlikte, yüz bölgesinin bir kısmı ya da tamamı üzerinde değişik teknikler uygulayarak yapılabilmektedir. Bu teknikler değişmekle birlikte, esas olan insana özgü yüz hat ve değerlerini tespit edip tanıma işlemini gerçekleştirme şeklinde yapılmaktadır.

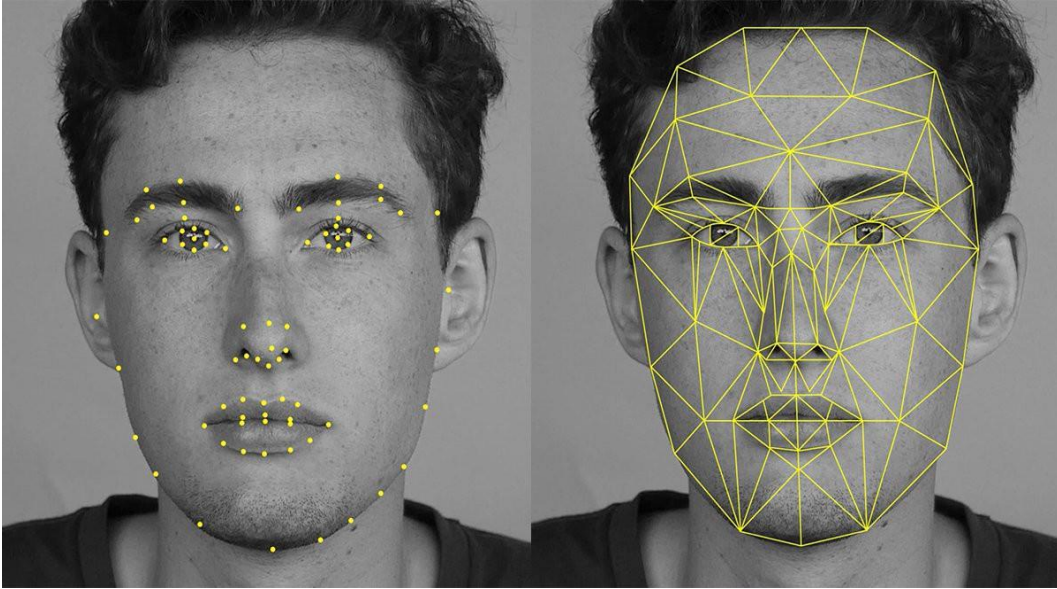
Yüz tanıma sistemleri yüz izi (faceprints) denilen insan yüzündeki 80 belirli noktanın analizi sonucunda ortaya çıkan sayısal kodlar ile çalışır. Bu belirli noktalar:

- Elmacık kemikleri,
- Göz çukurlarının derinliği,
- Burun uzunluk ve genişliği gibi kişiye özel özelliklerin tümünü içerir.

Bu sayısal kodlar daha önceden veri tabanına işlenmiş sayısal kodlar ile karşılaştırılarak yüz tanıma sistemi gerçekleştirilir.

Yüz tanıma sistemini ortam koşulları etkileyebilir. Ortamdaki ışığın az olması, profilde bir kayma yüz tanıma sisteminin doğru çalışmasını engelleyebilir. Gelişen teknoloji ile birlikte yüz tanıma sistemleri üç boyutlu modelleme ile daha doğru çalışması sağlanmıştır. Uygun koşullar altında %100'e yakın doğru sonuç vermektedir.

Akıllı telefonların gelişmesiyle birlikte yüz tanıma sistemleri telefonlarda da kullanılabilir hale getirdi. Telefonun kilidini açmak için önceden verisi alınmış resmin bilgileri ile kilidi açmak istediğimiz zaman çektiğimiz resmin verileri karşılaştırılır ve eşleşir ise kilit açılır.



Şekil 2.9 Yüz tanıma teknolojisine örnek yüz hattı (İnt.Kyn.9).

Yüz tanıma, ileri teknoloji silahlarının yönetimi için, özellikle ABD'de sıkça kullanılmaktadır. Bunun dışında, caddelere yerleştirilen güvenlik kameraları ile caddelerin izlenmesi ve aranmakta olan bir suçlunun bu şekilde yakalanması gibi uygulamalarda da kendilerine yer edinmişlerdir. Özellikle son yıllarda uygulama alanlarının artması nedeniyle yüzlerin otomatik olarak tanınması popüler bir konu haline gelmiştir. Günümüzde yaygın kullanımı olmamakla birlikte ilerde gelişen teknoloji paralelinde daha yaygın kullanım alanı olacağını düşünmekteyim.

### 2.5.1 Yüz Tanımının Tarihi

Yüz tanıma sistemi bilim dünyası için aslında yeni bir kavramdır. 1964 ve 1965 yılları arasında, bilim adamları insan yüzlerini tanımak için bilgisayar ortamında çalıştılar (Bledsoe 1966a, 1966b, Bledsoe and Chan 1965). Bu işten gurur duyuyordu, ancak finansman çok fazla tanıtım yapılmasına izin verilmeyen isimsiz bir istihbarat dairesi tarafından sağlandığından, çalışmaların çoğu yayınlanmadı. Amaç, görüntülerin ve fotoğrafların geniş veri tabanı içerisinde görüntü kayıtlarından birinin seçilen fotoğrafla eşleşeceği şekilde küçük bir kayıt kümesini seçmektir. Metodun başarısı, cevap listesinin veri tabanındaki kayıt sayısına oranı olarak ölçülebildi.

Yüz tanımının zorlaşmasındaki problem başın açısı ve eğimi, ışık şiddeti ve açısı, yüz ifadesi, yaşlanma vb. durumlardaki büyük değişikliklerdir. Dijital ortamda yüz tanıma girişimleri çok az değişim durumlarında veya hiç değişiklik olmadığında izin vermemiştir. Yine de, bazı araştırmacılar tarafından sıklıkla kullanılan işlenmemiş optik verilere ait korelasyon yöntemi (veya desen eşlemesi), değişkenliğin büyük olduğu durumlarda başarısızlığa uğrayacaktır. Özellikle, iki farklı kafa rotasyonuna sahip aynı kişinin iki resmi arasında korelasyon çok düşüktür (Bledsoe 1966b).

Bu proje insan-makine olarak adlandırıldı, çünkü insan fotoğraflarının bir dizi özelliklerinin koordinatlarını çıkartılarak daha sonra bilgisayar tarafından tanınmak için kullanıldı. Bir grafik tableti (GRAFACON veya RAND TABLET) kullanarak, operatör öğrencilerin merkezi, gözlerin iç köşesi, gözlerin dış köşesi, çukur-zirve noktası vb. gibi özelliklerin koordinatlarını çıkarırdı. Bu koordinatlardan, ağız genişliği ve göz genişliği, göz bebeği gibi 20 farklı mesafenin listesini oluşturdu. Bu listedeki operatörler saatte yaklaşık 40 resim işleyebildi. Veri tabanını oluştururken, fotoğraftaki kişinin adı hesaplanmış mesafeler listesiyle ilişkilendirilerek bilgisayarda saklanmıştır. Tanıma aşamasında, uzaklıklar, her fotoğraf için karşılık gelen mesafe ile karşılaştırılmış ve fotoğraf ile veri tabanı kaydı arasında bulgu elde edilmiştir. En yakın kayıtlar iade edilir.

Herhangi iki resmin kafa rotasyonuna, eğilmeye ve ölçeğe (kameranın uzaklığına) denk

gelmesi olası olmadığından, her bir mesafe kümesi yüzü bir ön yönde gösterecek şekilde normalleştirilir. Bu normalizasyonu gerçekleştirmek için, program ilk olarak eğimi, yalınlığı ve dönüşü belirlemeye çalışır. Daha sonra, bu açıları kullanarak bilgisayar, bu dönüşümlerin hesaplanan mesafeler üzerindeki etkisini ortadan kaldırır. Bu açıları hesaplamak için, bilgisayarın kafanın üç boyutlu geometrisini bilmesi gerekir. Gerçek kafalar mevcut olmadığı için, Bledsoe (1964) yedi kafa üzerindeki ölçümlerden elde edilen standart bir kafa kullanmıştır.

1966'da PRI'den ayrıldıktan sonra, bu çalışmaya Stanford Araştırma Enstitüsünde devam etti. 2000'den fazla fotoğrafın bulunduğu bir veri tabanında yapılan deneylerde, bilgisayar aynı tanıma görevleri ile sunulduğunda tüm sonuçlarda insanlardan daha iyi performans göstermiştir (Bledsoe 1968).

1988'de yüzün güvenilir eşsiz bir kimliğini oluşturulması için yüzün 100 ayrı noktasından veri alınması gerektiği savunuldu ve daha kararlı bir sistem ortaya çıktı. Teknolojinin gelişmesiyle, bu teknik daha güçlü yazılımlar ile daha sağlıklı kararlar alarak desteklenmektedir.

Yaklaşık 1997 yılında Bochum sistemi, Amerika Birleşik Devletleri Ordusu Araştırma Laboratuvarı tarafından finanse edilerek Christoph von der Malsburg ve Almanya'daki Bochum Üniversitesi mezunu öğrenciler tarafından geliştirildi. Yazılım, ZN-Face olarak satıldı ve Deutsche Bank ve havalimanı işletmecileri ve diğer yoğun yerler gibi müşteriler tarafından kullanıldı. Yazılım, kusursuz yüz görüşlerinden daha az tanımlamak için yeterince güçlüydü. Ayrıca, bıyık, sakal, saç stilleri ve gözlükler, hatta güneş gözlüğü gibi durumlarda sıklıkla karşılaşılan engelleri görebiliyordu (ScienceDaily 1997).

2006 yılında, en son yüz tanıma algoritmalarının performansı Face Recognition Grand Challenge'da performansı değerlendirildi. Testlerde yüksek çözünürlüklü yüz görüntüleri, 3 boyutlu yüz taramaları ve iris görüntüleri kullanılmıştır. Sonuçlar, yeni algoritmaların, 2002'nin yüz tanıma algoritmalarından 10 kat ve 1995'e göre 100 kat daha doğru olduğunu göstermiştir. Algoritmaların bazıları, yüzleri tanımada insan



katılımcılardan daha iyi performans gösterebilmiş ve tek yumurta ikizlerini benzersiz olarak tanımlayabilmiştir (Williams 2007).

## **2.5.2 Yüz Tanımının Uygulama Alanları**

### **2.5.2.1 Sosyal Medyalar**

Sosyal medya platformları, farklı uygulamalardan gelen sert rekabetin yanı sıra daha geniş bir kullanıcı tabanını çekmek için işlevselliklerini çeşitlendirmek amacıyla yüz tanıma yeteneklerini benimsemiştir.

2013 yılında kurulan Looksery, Kickstarter'daki yüz değiştirme uygulaması için çalışmalara başladı. Başarılı bir süreçten sonra, Ekim 2014'te Looksery piyasaya çıktı. Uygulama, kullanıcıların görünümünü değiştiren yüzler için özel bir filtreyle başkalarıyla görüntülü sohbet etmeyi sağlıyor. FaceTune ve Perfect365 gibi görüntü büyütme uygulamaları olsa da bunlar statik görüntülerle sınırlıyken, Looksery artırılmış gerçekliğin canlı videolara izin verdi. 2015'in sonlarında, SnapChat daha sonra dönüm noktası lens fonksiyonu olacak olan Looksery'yi satın aldı (Business Insider Singapore 2015).

SnapChat'in yüz tanıma teknolojisini kullanan animasyonlu objektifleri, kullanıcıların görünüşlerini değiştirmek için filtreler eklemelerine izin vererek selfie yeniden oluşturdu ve yeniden tanımladı. Filtrelerin seçimi her gün değişiyor; bazı örnekler, kullanıcıların kendilerini yaşlanmış, başlarının üstüne sanal çiçek taçlarını yerleştiren bir görüntü gibi görünmesini sağlayan örnekler sunmaktadır.

DeepFace, Facebook'ta bir araştırma grubunun oluşturduğu derin bir öğrenme yeteneğine sahip yüz tanıma sistemidir. Dijital görüntülerde insan yüzlerini tanımlar. 120 milyondan fazla bağlantı ağırlığına sahip dokuz katmanlı bir sinir ağı kullanıyor ve Facebook kullanıcıları tarafından yüklenen dört milyon görüntü üzerinde eğitilmektedir (Simonite 2014). Sistemin, FBI'nın Yeni Nesil Tanımlama sistemi için %85'ine kıyasla %97 doğru olduğu söyleniyor (Brandom 2014).

### **2.5.2.2 Mobil Teknolojiler**

Apple, parmak izi tabanlı bir sistem olan Touch ID'ye biyometrik kimlik doğrulama ardılı olarak amiral gemisi iPhone X'de Face ID'yi tanıttı. Face ID, iki bölümden oluşan bir yüz tanıma sensörüne sahiptir, kullanıcının yüzüne 30.000'den fazla kızılötesi nokta yansıtan bir "Romeo" modülü ve deseni okuyan bir "Juliet" modülü geliştirmiştir. Telefon sahibinin yüzü ile bir eşleşmeyi teyit etmek için, model cihazın merkezi işlem biriminde (CPU) yerel bir "Güvenli Muhafaza" ya gönderilir. Yüz desenine Apple tarafından erişilemiyor. Sistem, yetkisiz erişimi engellemek amacıyla gözleri kapalı olarak çalışması engellenmiştir (Brandom 2017).

### **2.5.2.3 Güvenlik ve Denetim**

Avustralya Sınır Gücü ve Yeni Zelanda Gümrük Hizmetleri, e-pasaport mikroçipindeki verilerle yolcunun yüzünü karşılaştıran yüz tanıma özelliğini kullanan SmartGate adlı otomatik bir sınır işleme sistemi kurmuştur. Ön inceleme Kiosk programının bir parçası olarak yeni bir yüz tanıma programı olarak kullanıldı. Bu program ilk olarak 2017 başlarında Ottawa Uluslararası Havaalanı'nda kullanıldı ve 2018 yılında diğer havaalanlarına uygulanması beklenmektedir. Panama'daki Tocumen Uluslararası Havaalanı, havalimanından geçen aranan kişileri belirlemek için yüzlerce canlı yüz tanıma kameralarını kullanan bir havaalanı çapında gözetim sistemi işletiyor. Birleşik Krallık'taki polis güçleri, 2015'ten bu yana halka açık etkinliklerde canlı yüz tanıma teknolojisinde denemeler gerçekleştiriyorlar. Ancak, Big Brother Watch adlı bağımsız organizasyon tarafından yayınlanan yeni bir rapor ve araştırmayla bu sistemlerin %98'e kadar kusurlu olduğunu tespit etti (Watch 2018).

### **2.5.2.4 Ulusal Güvenlik**

ABD Dışişleri Bakanlığı, dünya genelindeki en büyük yüz tanıma sistemlerinden birini kullanarak, genellikle sürücü ehliyetli fotoğraflarından çekilen fotoğraflarla birlikte, 117 milyon Amerikalı yetişkinden oluşan bir veri tabanı ile çalışıyor. Hala tamamlanmaktan uzak olsa da fotoğrafta kim olduğu hakkında ipuçları vermek için belirli şehirlerde kullanıma sunulmaktadır. FBI, fotoğrafları, pozitif tanımlama için değil, araştırmacı bir araç olarak kullanıyor (Ritchie, 2016).

Son yıllarda Maryland, insanların yüzlerini sürücü lisans fotoğraflarıyla karşılaştırarak yüz tanıma özelliğini kullanmıştır. Sistem, polisin gözetiminde Freddie Grey'in ölümünden sonra ateşli protestocuları tutuklamak için Baltimore'da kullanıldığında tartışmalara yol açmıştır. Diğer birçok devlet benzer bir sistemi kullanıyor veya geliştiriyor, ancak bazı eyaletlerin kullanımını yasaklayan yasaları vardır.

FBI ayrıca, yüz tanıma ve hem ceza hem de sivil veri tabanlarından çekilebilen parmak izleri ve iris taramaları gibi daha geleneksel biyometrikleri içerecek “Yeni Nesil Güvenli Giriş” (NGI) programını başlatmıştır (FBI 2016).

2017'nin sonlarına doğru Çin, Sincan'da yüz tanıma teknolojisini uygulamaya koydu. Bölgeyi ziyaret eden muhabirler, her yüz metrede birkaç şehirde gözetleme kameralarının yanı sıra benzin istasyonları, alışveriş merkezleri ve cami girişleri gibi alanlarda yüz tanıma kontrol noktaları olduğunu kaydettiler.

#### **2.5.2.5 Diğer Kullanımlar**

Güvenlik sistemleri için kullanılmaya ek olarak, yetkililer yüz tanıma sistemleri için başka uygulamalar da bulmuşlardır. Ocak 2001'de Super Bowl XXXV'de, Tampa Bay, Florida'daki polis, etkinliğe katılacak potansiyel suçluları ve teröristleri araştırmak için Viisage yüz tanıma yazılımını kullandı. Küçük sabıka kaydı olan 19 kişi potansiyel olarak tespit edildi.

2000 Meksika başkanlık seçimlerinde, Meksika hükümeti seçmen sahtekarlığını önlemek için yüz tanıma yazılımı kullanmıştır. Bazı kişiler, birden fazla oylama girişimi için birkaç farklı isimle oylamaya kayıt olmuşlardır. Yeni yüz görüntülerini seçmen veri tabanında bulunanlarla karşılaştırarak yetkililer çift kayıtlarını azaltabildiler. ABD'de sahte kimlik kartları ve sürücü ehliyetlerinin alınmasını önlemek için benzer teknolojiler kullanılmaktadır (Facial recognition at DMV 2007).

Diğer taraftan yüz tanıma, çeşitli bilgi işlem platformları ve cihazları için bir biyometrik kimlik doğrulama biçimi olarak geliştirilmiştir (Bonsor and Johnson 2008). Dünyada en yaygın işletim sistemlerinden biri olan Android 4.0 ve sonrasında, bir akıllı telefonun

ön kamerasını kullanarak, cihazların kilidini açma aracı olarak yüz tanıma özelliğini eklenmiştir (Heater 2017). Microsoft, Xbox 360 video oyun konsoluna, Kinect aksesuarı, ve Windows 10 ile "Windows Hello" platformu (kızılötesi aydınlatmalı kamera gerektirir) aracılığıyla yüz tanıma girişini tanıtmıştır. Apple'ın iPhone X akıllı telefonu, kızılötesi aydınlatma sistemi kullanan "Face ID" platformu ile ürün hattına yüz tanıma özelliğini tanıtmıştır.

Yüz tanıma sistemleri, fotoğraf konularını tanımlamak için fotoğraf yönetimi yazılımı tarafından da kullanılmıştır. Bu, fotoğrafların kişi tarafından aranması gibi özelliklerin sunulması ve fotoğrafta varlığı tespit edildiğinde fotoğrafların belirli bir kişiyle paylaşılmasını önermektedir (Perez 2017).

Günümüzde yüz tanıma, web sitelerinde, telefon uygulamaları ve ödeme yöntemleri genellikle ek güvenlik olarak kullanılmaktadır.

## **2.6 Gömülü Sistemler**

Gömülü bir sistem, genellikle gerçek zamanlı hesaplama kısıtlamaları ile daha büyük bir mekanik veya elektrik sistemi içinde özel bir işlevi olan bir bilgisayar sistemidir (Heath 2003). Genellikle donanım ve mekanik parçalar dahil olmak üzere komple bir cihazın bir parçası olarak gömülü şekilde çalışmaktadır. Gömülü sistemler, günümüzde yaygın olarak kullanılan birçok cihazı kontrol etmektedir. Tüm mikroişlemcilerin %98'i gömülü sistemlerin bileşenleri olarak üretilmektedir (Barr and Massa 2006).

Tipik gömülü bilgisayarların genel özelliklerini muadillerle karşılaştırıldığında düşük güç tüketimi, küçük boyut, stabil çalışabilme ve birim başına maliyetinin düşük oluşu avantajlarıdır. Kısıtlı işlem kaynaklarına sahip olmaları, programlamak ve etkileşimde bulunmak için daha çok efora ihtiyaç duyulmaktadır. Bu da farklı maliyetleri oluşturur. Bununla birlikte, donanımın üstünden iletişime geçerek bağlantıları kurar veya mevcut sensörlerden ve gömülü birimlerden oluşan bir ağın varlığından yararlanarak, kaynakları en iyi şekilde yönetebilir ve zenginleştirilmiş fonksiyonlar sağlayabilir (Alippi 2014).

Klasik gömülü sistemler genellikle mikro denetleyicilere sahiptirler, günümüzde sıradan bilgisayarlarda daha karmaşık sistemlerde yaygın olarak kullanılan mikroişlemciler, gelişen ARM mimarisi ile modern gömülü sistemlerde de kullanılmaya başlamıştır. Her iki durumda da, kullanılan işlemci(ler), genel amaçları olan belirli hesaplama veya ilgili uygulama için özel olarak tasarlanmış yapılara sahip olabilirler.

Gömülü sistem belirli görevlere ayrılmış olduğundan, ürünün boyutunu küçültmek, maliyetini azaltmak ve performansı artırmak için ürüne özel üretilir veya boyutlandırılırlar. Bu çerçevede gömülü sistemleri aşağıdaki gibi örneklendirebiliriz (İnt.Kyn.21);

- Banka ATM'leri
- Uçuş kontrol donanım/yazılımı ve uçak ve füzelerdeki diğer tümleşik sistemlerden oluşan havacılık elektroniği parçaları
- Yönlendirici (router), timeserver ve güvenlik duvarı (firewall) gibi bilgisayar ağ ekipmanları
- Bilgisayar yazıcıları
- Fotokopi makineleri
- Disket sürücüler (floppy disket sürücüler ve sabit disk sürücüler)
- Termostat, klima, sprinkler ve güvenlik izleme sistemleri gibi ev otomasyonu ürünleri
- Hesap makineleri
- Mikro dalga fırınlar, çamaşır makinesi, televizyon setleri ve DVD oynatıcı/kaydedici gibi ev elektroniği ürünleri
- Tıbbi ekipmanlar
- Çok işlevli kol saatleri
- İnternet radyo alıcıları, TV set top box, ve dijital uydu alıcılar gibi çoklu ortam uygulamaları
- Çok işlevli yazıcılar
- PDA'lar gibi küçük avuç içi bilgisayarlar
- PDA ve Java destekli gelişmiş cep telefonları
- Endüstriyel otomasyon ve izleme için PLC'ler

- Video oyun konsolları ve avuç içi oyun konsolları
- Taşınabilir bilgisayarlar

Nispeten düşük maliyetli bir mikro denetleyici, çok sayıda ayrı bileşenle aynı rolü yerine getirmek üzere programlanabilir. Bu bağlamda gömülü bir sistem genellikle geleneksel bir çözümden daha karmaşık olmasına rağmen, karmaşıklığın çoğu mikro denetleyicinin kendisinde yer alır. Çok az ek bileşen gerekli olabilir ve tasarım çabasının çoğu yazılımda yer alır. Yazılım prototip ve testi, gömülü bir işlemci kullanmadan yeni bir devrenin tasarımı ve inşasına kıyasla daha hızlı olabilir.

### **2.6.1 Gömülü Sistemlerin Tarihçesi**

İlk tanınabilir modern gömülü sistemlerden biri 1965 MIT Enstrümantasyon Laboratuvarında Charles Stark Draper tarafından geliştirilen “Apollo Rehber Bilgisayarı” olmuştur. Aya yapılan yolculuklarda iki tane kullanılırdı ve komuta modülü ve LEM’in eylemsiz rehber sistemlerini çalıştırıyordu. Projenin başlangıcında, Apollo kılavuz bilgisayarı, boyut ve ağırlığı azaltmak için yeni geliştirilen monolitik tümleşik devreleri kullanması nedeniyle Apollo projesinde en riskli madde olarak kabul edildi.

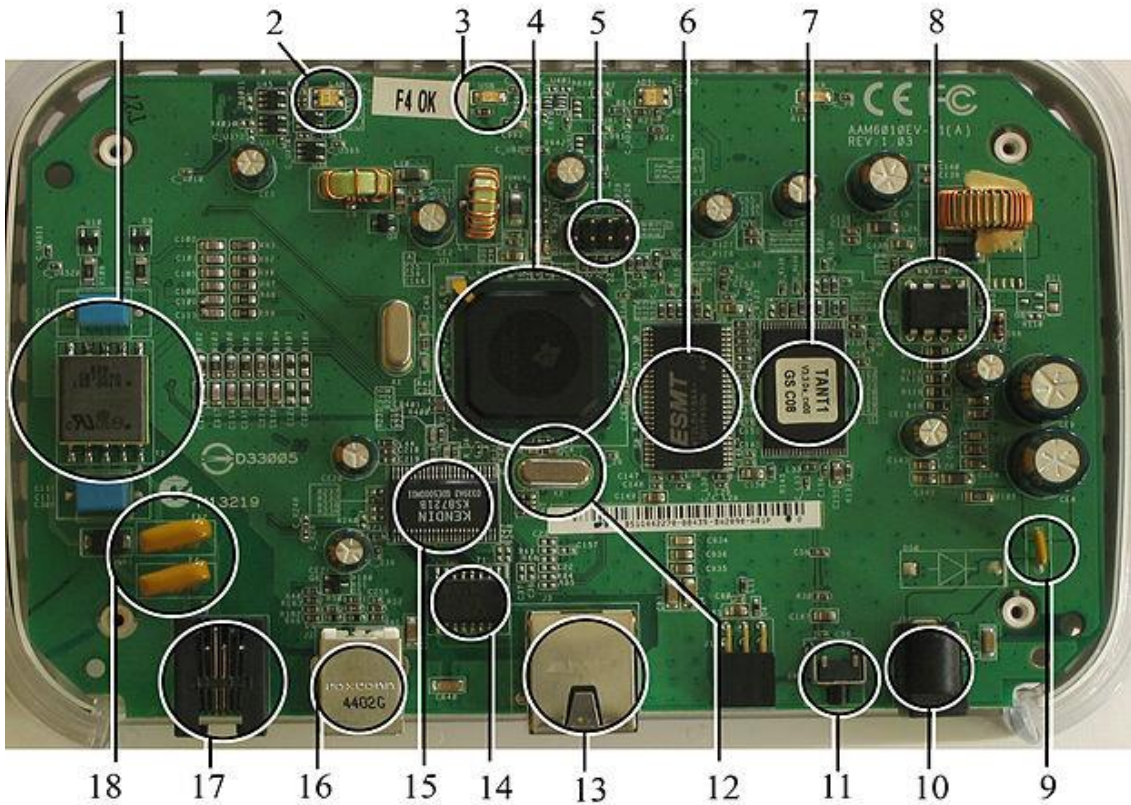
Kitlesel olarak üretilen bir gömülü sistem, 1961'de yayımlanan “Minuteman Füzesi” için “Autonetics D-17 Rehber Bilgisayarı” oldu. Minuteman II, 1966'da üretime geçtiğinde, D-17, ilk defa yüksek hacimli tümleşik devrelerin kullanıldığı yeni bir bilgisayara yerini bıraktı.

1960'larda gömülü sistemlerde, maliyet fiyatları düşmesinin yanı sıra işlem gücü ve işlevsellikte önemli bir yükseliş oldu. Örneğin, ilk özel bir mikro işlemci olan Intel 4004, hesap makineleri ve diğer küçük sistemler için kullanıldı. Fakat yine de harici bellek ve destek çiplerine ihtiyaç duyuldu. 1978'de National Engineering Manufacturers Association, tek kartlı bilgisayarlar, sayısal ve olay tabanlı kontrolörler gibi hemen hemen tüm bilgisayar tabanlı kontrol cihazları dahil olmak üzere programlanabilir mikro denetleyiciler için bir "standart" yayınladı.

Mikroişlemci ve mikro kontrolörlerin maliyeti düştükçe, potansiyometre ve değişken kondansatörler gibi pahalı analog düğme bileşenlerin yerine, tüketici ürünlerinde bile bir mikroişlemci tarafından okunan yukarı / aşağı düğmeleri yerleştirmek mümkün hale gelmiştir. 1980'lerin başlarına gelindiğinde, mikro denetleyici oluşturan işlemci ile bellek, giriş ve çıkış sistemi bileşenleri aynı çipe entegre edilmiştir.

## 2.6.2 Gömülü Sistem Bileşenleri

Elektronik cihazlar genellikle mikroişlemci ya da mikro kontrolör kullanırlar. Bazı büyük ya da eski sistemler ise genel maksatlı büyük bilgisayarlar ya da minibilgisayarlar kullanırlar. Modern gömülü sistemler günümüzde Resim 1'deki gibi bir tasarıma sahiptirler. Bu tasarımlar işlevselliğine göre farklılıklar gösterebilirler de genelde birbirlerine yakındırlar.



**Resim 2.1** Gömülü sistem bileşenleri (İnt.Kyn.3).

1. Telefon ayırma elektroniği (ADSL için)
2. Çok renkli LED (ağ durumunu gösterir)

3. Tek renkli LED (USB durumunu gösteren)
  4. Ana işlemci (CPU / Mikro kontrolör),
  5. JTAG (Ortak Test Eylem Grubu) test ve programlama portu
  6. RAM
  7. Flash bellek
  8. Güç kaynağı regülatörü
  9. Ana güç kaynağı sigorta
  10. Güç bağlantısı
  11. Yeniden başlatma tuşu
  12. Kuvars kristali
  13. Ethernet bağlantı noktası
  14. Ethernet Güç Trafo
  15. Ethernet PHY alıcı-verici
  16. USB girişi
  17. Telefon (RJ11) portu
  18. Telefon konektörü sigortaları
- bileşenlerinden oluşmaktadır.



### **3. MATERYAL ve METOT**

#### **3.1 Giriş**

Bu bölümde tez çalışması içerisinde kullanılan teknolojiler, yardımcı kütüphaneler ve geliştirme ortamları ile ilgili bilgilendirmeler verilecektir. Kısaca özetlemek gerekirse uygulama; Raspberry Pi 3 mini bilgisayarı ile Raspbian işletim sistemi üzerinde Python dili kullanılarak OpenCV kütüphanesinde geliştirilen yüz tanıma algoritmalarının karşılaştırılması yapılacaktır. OpenCV görüntü işleme odaklı bir kütüphane olması dolayısı ile yüz tanıma ve bulma konularında oldukça yaygın kullanılmaktadır.

Geliştirme Platformu :	Raspberry Pi 3
Yazılım Dili :	Python 3
Kütüphane :	OpenCV
Kullanılacak IDE :	Pyhton 3.5.3 IDE

#### **3.2 Sistemin Tanıtımı**

IoT ortamında geliştirilen Yüz tanıma sistemi yazılımı, Endüstri 4.0 konusu ile oluşan nesnelerin interneti ortamlarında biyometrik sistemlerden biri olan yüz tanımanın gerek güvenlik gerekse kullanıcıların tespiti ve tanımlanması için kullanılmasının uygun ve yeterli olması konusunu araştırmaktır. Bu amaç doğrultusunda mini bilgisayar diye adlandırılan Raspberry Pi teknolojisi üzerinde geliştirme amaçlanmaktadır.

#### **3.3 Geliştirme Ortamı**

Endüstri 4.0 ile birlikte geliştiriciler için oldukça küçük farklı CPU ve GPU seçeneklerine sahip teknolojileri barındıran mini hatta micro denilecek seviyede bilgisayarlar üretilmiştir. Projemiz için oldukça yaygın kullanılan ve üzerinde görüntü işlemeye olanak sağlayan GPU barındıran serinin sonuncusu olan Raspberry Pi 3 teknolojisi kullanılması tercih edilmiştir. Raspberry PI 3 gömülü ortamı için ek modül bileşenleri geliştirilmiştir. Araştırma kapsamında en önemli modül olarak bu cihaz için

geliştirilmiş CSI bağlantı noktası kullanan Raspberry PI kamera modülü 1.3 versiyonu kullanılmıştır. Ek olarak görüntüyü almak ve her ortamda rahatlıkla kullanabilmek için Raspberry PI için üretilmiş ve Raspbian tarafından tanınabilen 5 inc'lik dokunmatik HDMI girişli LCD ekran monte edilmiştir. Tamamen Raspbian işletim sistemi tarafından tanınabilen bu ekran kolayca takılıp çıkartılabilmektedir.

### **3.3.1 Raspberry Pi**

Raspberry Pi, Birleşik Krallık'ta Raspberry Pi Vakfı tarafından okullarda bilgisayar bilimini öğretmek amacıyla geliştirilmiş kredi kartı büyüklüğünde tek kartlı tümleşik bir bilgisayardır (İnt.Kyn.24). Kuruluşun kendi adını üzerinde taşıyan cihaz, küçük ve ucuz bir bilgisayar alternatifi olması için üretilmiştir.

Raspberry Pi'nin bu bilgisayarı üretmesinin ardında yer alan temel fikir okullarda çocuklara bilimsel çalışmaların yaptırılabilmesine ucuz ve kolay olanaklar sağlamaktır. Her ne kadar ürünün üretilmesinin ardındaki temel fikir eğitim olsa da Raspberry Pi'nin çoğunlukla kişisel kullanıcılara ve minyatür bilgisayar meraklılarına da hitap ettiği görülmüştür. Az enerjiye ihtiyaç duyması ve küçük boyutlarından dolayı gömülü ortamlarda kullanılmaya başlanmıştır.

### **3.3.2 Raspberry Pi'nin Donanım Özellikleri**

Bu bilgisayar ilk olarak 700 Mhz ARM işlemci ve 256 MB yada 512 MB RAM ile birlikte kullanıcılarına ulaşmıştır. Farklı modeller altında sunulan mini bilgisayarlar USB 2.0 port desteğine sahip ve MicroSD yardımıyla depolama kapasitesi artırma imkanı bulunmaktadır. Cihazın üzerinde görüntü ve ses aktarımı için kullanılacak bir de HDMI çıkışı yer almaktadır. Ethernet girişi ve USB yardımıyla kazandırılan Wi-Fi desteğiyle de Raspberry Pi'lerin internet ihtiyacı karşılanabilmektedir.



**Resim 3.1** Raspberry Pi 3 model B (İnt.Kyn.10).

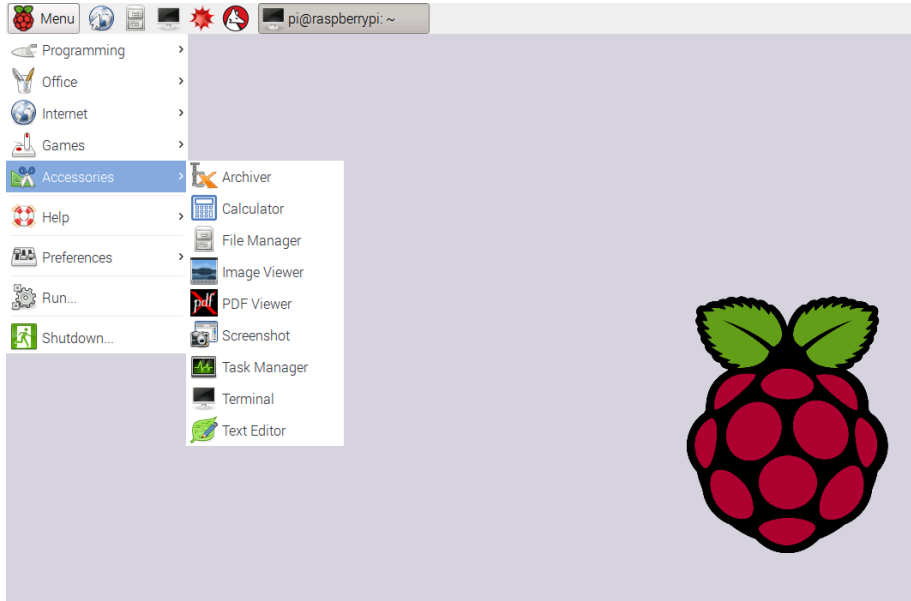
Proje kapsamında kullandığımız Raspberry Pi 3 Teknik Özellikleri:

- Broadcom BCM2837 SoC
- 1.2 GHz 4 çekirdekli 64-bit ARM Cortex-A53 işlemci
- 2 çekirdekli Videocore IV® Multimedia işlemcisi
- 1 GB LPDDR2 bellek
- Dahili 802.11b/g/n destekli WiFi
- Bluetooth 4.1, low-energy destekli
- 10/100 Mbit destekli Ethernet portu
- HDMI portu (HDMI 1.4 destekli)
- Kompozit video ve ses çıkışı için 3.5mm TRRS (4 uçlu) konektör
- 4 adet USB2.0 portu
- 40 adet GPIO pini, önceki Raspberry Pi modelleri ile uyumlu
- WiFi/Bluetooth için dahili çip anten
- CSI (kamera) ve DSI (ekran) konektörleri
- Mikro SD kart yuvası
- Boyutlar: 85 x 56 x 17 mm

### 3.3.3 Raspberry Pi İşletim Sistemi

Cihazın kullandığı işlemcisi mimarisi nedeniyle güncel Windows sürümlerinden sadece Windows 10 IoT Core ile kullanılabilir. Fakat Microsoft'un çıkarmış olduğu bu işletim sistemi oldukça kısıtlı bir ekran ara yüzüne sahiptir. Buda geliştiricilerin bu ortamı kullanmasında oldukça güçlükler çıkarmaktadır. Windows 10 IoT Core içerisinde bir kod yüklemek için Visual Studio IDE ortamına ihtiyaç duyulmaktadır. Cihaz üzerinde derleme veya doğrudan koda müdahale etmek nerdeyse imkansızdır.

Diğer bir yandan Raspberry Pi üzerinde Linux varyantlı işletim sistemleri kullanımı oldukça yaygındır. Raspberry Pi ile şu an için sorunsuz şekilde kullanılabilen başlıca işletim sistemleri Archlinux, Fedora, Puppy Linux, Slackware, FreeBSD, NetBSD, openSUSE, ve RISC OS'dir. Ayrıca Raspberry Pi için geliştirilmiş Raspbian işletim sistemi de bulunmaktadır.



**Resim 3.2** Raspbian masaüstü görünümü.

Raspbian, Raspberry Pi için optimize edilmiş Debian'a dayalı tamamen ücretsiz bir işletim sistemidir. Bu işletim sistemi, Raspberry Pi çalıştıran temel programlar ve yardımcı programlar kümesidir. Ayrıca, Raspbian bir işletim sisteminden daha fazlası vardır. Raspbian işletim sistemleri önceden derlenmiş 35.000'den fazla yazılım geliştirme

paketi ve kütüphaneleriyle yüklü gelmektedir. Raspberry Pi cihazlarına kolay kurulum olacak şekilde paketlenmiştir (İnt.Kyn.25).

Raspbian işletim sistemi günümüz işletim sistemi özelliklerini taşır hale gelmiştir. Standart Linux işletim sistemlerindeki başlat menüsüne benzer menü ile gelen işletim sistemine, duvar kağıdı da uygulanabilmektedir.

İşletim sisteminin son versiyonunda ekranın köşesinde iki yeni uyarı simgesi gelmiştir. Voltaj düşük olduğunda bir yıldırım simgesi ve Raspberry Pi ısıyorsa termometre göstergesi bu simgelerdir.

Pencere yapısı Windows 3.1'den biraz daha işlevsel olarak tasarlanmış şekilde karşımıza çıkmaktadır. Son birkaç yıldır masaüstünde Roboto yazı tipi kullanılıyordu. Ancak Infinality font oluşturma paketi dahil edilerek PIXEL'in daha iyi görünmesi sağlandı. Bu yazı tiplerinin ekranda nasıl oluşturulacağını optimize etmek için bazı özelleştirmeler gerçekleştirilmiştir.

WiFi ve Bluetooth menülerinde ilgili birimleri kapatmak için seçenekler mevcuttur. Bu amaçla dahili Wifi ve Bluetooth açıp, kapatılabilir butonları vardır. İşletim sistemi ile birlikte Sense HAT üzerinde sıcaklık, basınç ve nem sensörleri bulunur. Standartta kendi donanımında bulunmayan bu sensörleri ek modül olarak kullanabilmektedir. Ayrıca bu sensörlerden alınan bilgiler ile farklı uygulamalar geliştirilebilir. İşletim sisteminin son versiyonlarında web tarayıcısı olarak Chromium kullanılmaktadır. Chromium Chrome'un Raspberry Pi sürümüdür. Chromium siteleri ve video akışını hızlandırmak için Pi'nin donanım özelliklerini de kullanabilme kabiliyeti ile donatılmıştır. Chromium Raspberry Pi 1 ve Pi ZERO için biraz yavaş olabilir ancak Pi 2 ve Pi 3 üzerinde sorunsuz çalışmaktadır. Chromium önceden yüklenmiş iki eklenti ile gelmektedir. Ayrıca Block ve Youtube'da daha hızlı videolar için h264ify desteklenmektedir.

### 3.3.4 Python

Python, Guido Van Rossum adlı Hollanda'lı bir programcı tarafından yazılmış bir programlama dilidir. Geliştirilmesine 1990 yılında başlayan Python; C ve C++ gibi programlama dillerine kıyaslırsak şöyle sonuçlar elde edebiliriz.

- Daha kolay öğrenilir.
- Program geliştirme sürecini kısaltır yani hızlı yazılır.
- Diğer programlama dillerine aksine ayrı bir derleyici ihtiyacı duymaz.
- Hem daha okunaklı, hem daha temiz kodsall söz dizimine sahiptir.

Python'un bu ve buna benzer özelliklerinden dolayı, dünya çapında ün sahibi büyük kuruluşlar (Google, Yahoo! ve Dropbox gibi) yani Linux tabanlı işletim sistemleri üzerinden hizmet veren birçok yazılım, bu dil ile geliştirme yapmışlardır.

Bu arada Python dilinin, her ne kadar Python Programlama dili ile ilgili çoğu görsel malzemenin üzerine yılan resmi olarak görölse de, Python kelimesi aslında çoğu kişinin zannetiği aksine Piton anlamına gelmiyor. Python programlama dili ismini, Guido Van Rossum'un çok sevdiği Monty Python adlı altı kişilik bir İngiliz komedi grubun Monty Python's Flying Circus adlı gösterisinden gelmektedir.

Programlama Dili	Ortam	Değerlendirme Sonucu
1. Python	🌐 🖥️	100.0
2. C	📱 🖥️ 🖨️	99.7
3. Java	🌐 📱 🖥️	99.4
4. C++	📱 🖥️ 🖨️	97.2
5. C#	🌐 📱 🖥️	88.6
6. R	🖥️	88.1
7. JavaScript	🌐 📱	85.5
8. PHP	🌐	81.4
9. Go	🌐 🖥️	76.1
10. Swift	📱 🖥️	75.3

**Resim 3.3** IEEE 2017 yılı Dünyanın en popüler yazılım dilleri sıralaması (Cass 2017).

Yüz atmış ülkede dört yüz bin'den fazla üyesi bulunan IEEE'nin (Elektronik Mühendisleri Enstitüsü) hazırladığı 2017 yılı dünyanın en popüler yazılım dilleri sıralamasında Python Dünya genelinde birinci sırada yer almaktadır. Günümüzde yaygın olarak kullanıldığı için yazılım geliştirirken herhangi bir hususta doküman bulmak çok kolaydır.

### 3.3.5 Python IDE

IDE tanımı "integrated development environment" kelimelerinin kısaltılmasıyla oluşmuştur. Türkçe anlamı ise "tümleşik geliştirme ortamı" demektir. IDE bilgisayar programcılarının hızlı ve rahat bir şekilde program geliştirebilmesini amaçlayan, geliştirme sürecini organize edebilen birçok araç ile birlikte geliştirme sürecinin verimli kullanılmasına katkıda bulunan araçların tamamını içerisinde barındıran bir yazılım türüdür.

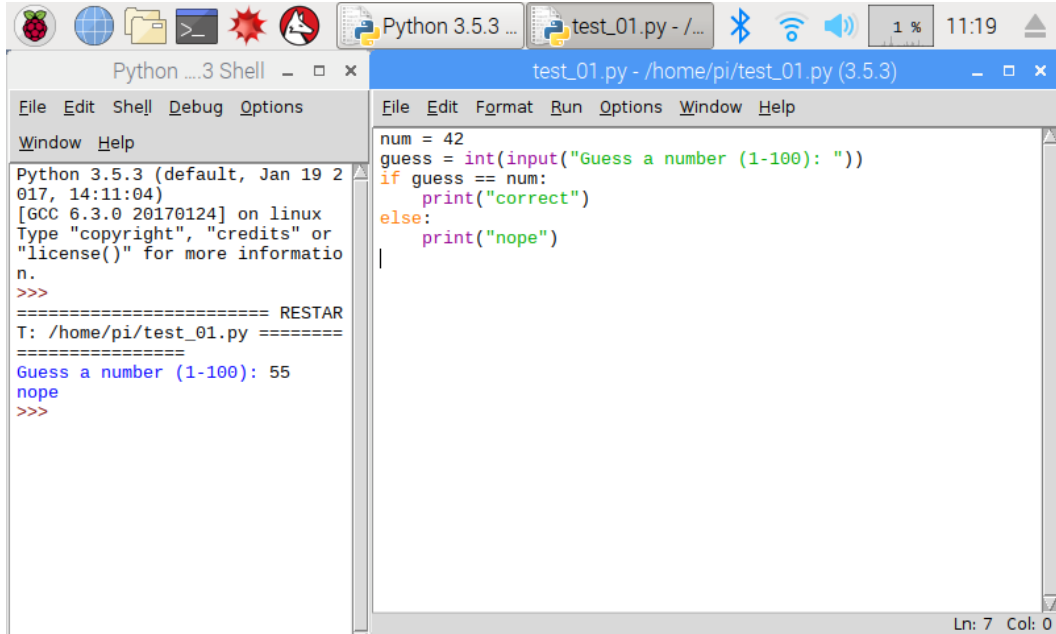
Tümleşik geliştirme ortamlarında olması gerekli en temel özellikler aşağıdaki gibidir:

- Programlama diline göre sözdizimi renklendirmesi yapabilen kod yazım editörü.
- Kod dosyalarının hiyerarşik olarak görülebilmesi amacıyla hazırlanmış gerçek

zamanlı bir dizelge.

- Tümüleşik bir derleyici, yorumlayıcı ve hata ayıklayıcı.
- Yazılımın derlenmesi, bağlanması, çalışmaya tümüyle hazır hale gelmesi ve daha birçok ek işi otomatik olarak yapabilmek amacıyla küçük inşa araçları.

Python IDE’de söz dizimi vurgulama özelliği sunar, ancak hiçbir satır numarası içermez. Mevcut satırı ve sütunu görmek için metin giriş penceresinin sağ alt kenarında bulunan statü alanı üzerinde bilgilendirme olarak gözükmektedir. Editörün sol kenarında satır numaralarını göstermek tümleşik geliştirme ortamını çok daha hantal yapmaktadır. Başka bir dezavantajı ise, bu tümleşik geliştirme ortamı sadece Python dili için çalışabilmekte ve ancak bir adımlık hata ayıklayıcı sunmaktadır. Masaüstü veya dizüstü bilgisayarlardaki Windows veya Linux tabanlı işletim istemi içerisinde çalışan birçok IDE birden çok hata adımlarını sonuç ekranlarına yansıtılabilmektedir. Raspbian tabanlı Python IDE’si, gömülü ortam içerisinde hızlı ve kolay çalışması amacıyla bu özelliklerden muaf tutulmuştur.



```
Python 3.5.3 Shell - x
File Edit Shell Debug Options
Window Help
Python 3.5.3 (default, Jan 19 2017, 14:11:04)
[GCC 6.3.0 20170124] on linux
Type "copyright", "credits" or
"license()" for more informatio
n.
>>>
===== RESTART
T: /home/pi/test_01.py =====
Guess a number (1-100): 55
nope
>>>
```

```
test_01.py - /home/pi/test_01.py (3.5.3)
File Edit Format Run Options Window Help
num = 42
guess = int(input("Guess a number (1-100): "))
if guess == num:
    print("correct")
else:
    print("nope")
|
```

Ln: 7 Col: 0

**Resim 3.4** Raspbian içerisinde standart kurulu gelen Python 3.5.3 IDE’si ekran görüntüsü.

Python IDE hata ayıklayıcısı, hata durumların da çok detay vermeden sadece yüzeysel hata bildirimini yapmaktadır. Bu durum yeni başlayanlar için dezavantaj oluşturmaktadır. Çıkan hataları daha iyi yorumlayabilmek için hata ayıklayıcısını başlatarak kod

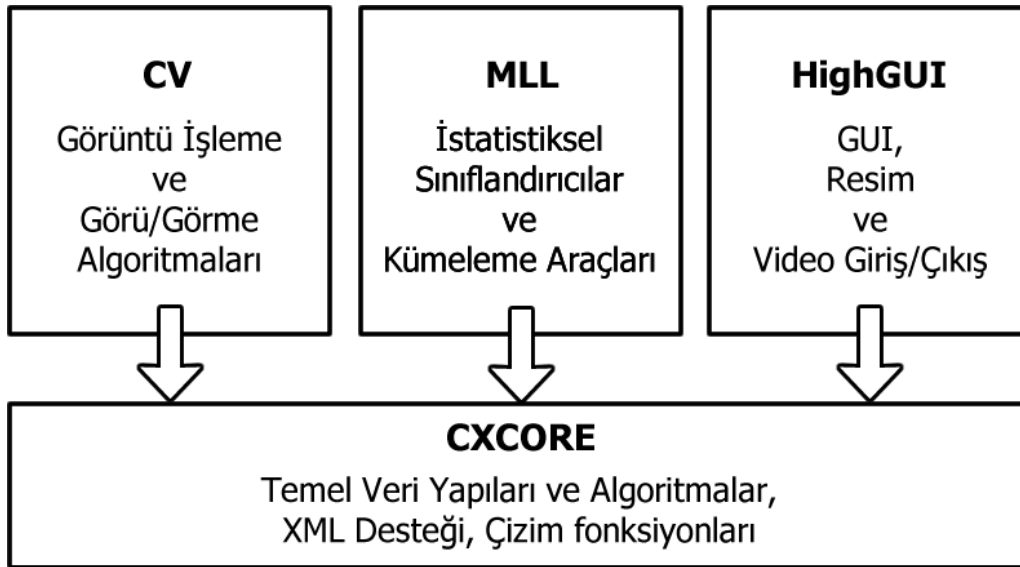


içerisinde “Kesme Noktası Belirleme”yi seçerek kodun çalışması sırasında hata yakalama noktaları ayarlanabilir.

Bu çalışmada Raspberry PI 3 ortamına entegre ettiğimiz beş inçlik dokunmatik ekran ile kodu takip etmek oldukça zor olduğu gözlemlendi. HDMI çıkışı ile daha büyük ekranlar ile çalışmak IDE kullanımını için kolaylık olacaktır.

### 3.4 OpenCV

OpenCV (Open Source Computer Vision) açık kaynak kodlu görüntü işleme kütüphanesidir. 1999 yılında Intel tarafından geliştirilmeye başlanmış daha sonra Itseez, Willow, Nvidia, AMD, Google gibi şirket ve toplulukların desteği ile gelişim süreci devam etmektedir.



Şekil 3.1 OpenCV bileşen yapısı (Kaehler and Bradski 2008).

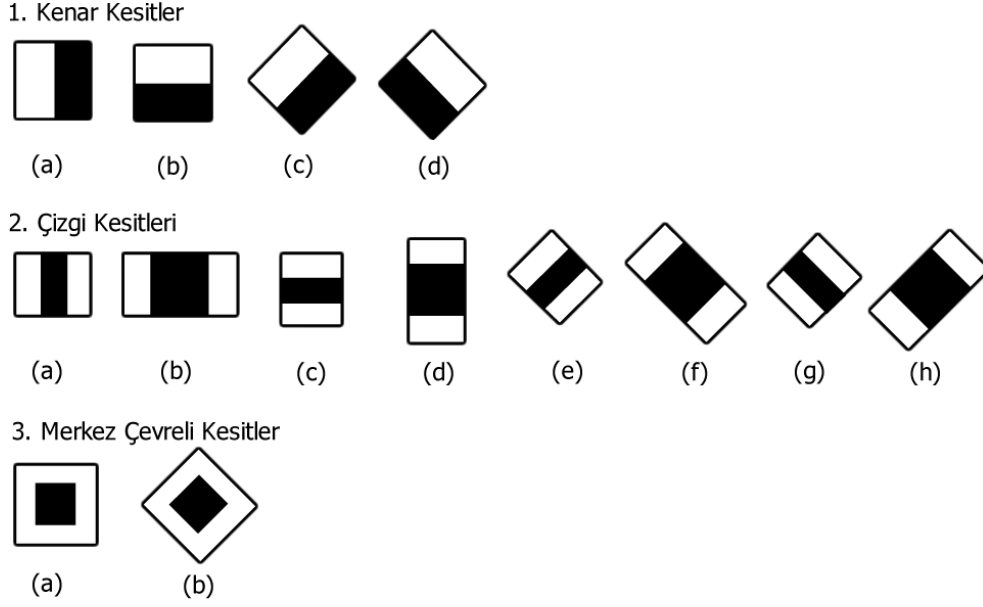
Computer Vision (Bilgisayarla Görme) kelimesinin baş harfleri kullanılarak isimlendirilen CV bileşeni, temel resim işleme fonksiyonları ve Bilgisayarla Görme için kullanılan yüksek seviyeli algoritmaları bünyesinde barındıran beş temel kütüphaneden biridir. Machine Learning Library kelimesinin baş harfleri kullanılarak isimlendirilen MLL bileşeni, adından da anlaşılacağı üzere makina öğrenmesi dalı için gerekli

istatistiksel verilere ulaşmak, mevcut verileri sınıflandırmak için kullanılan fonksiyonları/araçları içeren diğer bir kütüphanedir. HighGUI bileşeni, slider, form gibi OpenCV kütüphanesi içerisinde tanımlanmış pek çok nesneyi yaratabilmemizi sağlayan bir grafik arabirimi olmakla beraber, resim ve videoları kaydetmek, yüklemek, hafızadan silmek için gerekli giriş/çıkış (I/O) fonksiyonlarını da içerir.

İlk sürüm olan OpenCV alfa 2000 yılında piyasaya çıkmıştır. İlk etapta C programlama dili ile geliştirilmeye başlanmış ve daha sonra birçok algoritması C++ dili ile geliştirilmiştir. Açık kaynak kodlu bir kütüphanedir ve BSD lisansı ile altında geliştirilmektedir. BSD lisansına sahip olması bu kütüphaneyi istediğiniz projede ücretsiz olarak kullanabileceğiniz anlamına gelmektedir. OpenCV platform bağımsız bir kütüphanedir, bu sayede Windows, Linux, FreeBSD, Android, Mac OS ve iOS platformlarında çalışabilmektedir.

OpenCV son yıllarda ARM işlemcili cihazlarda çalışma isteği bu kütüphanenin daha küçük ve hafif versiyonu olan OpenCV 2 kütüphanesinin oluşmasına sebep olmuştur.

OpenCV kütüphanesinin görüntü işleme için çok kullanışlı ve çok gerekli olan video veya resimlerdeki nesnelere bulmak ve onları takip etme konusunda kuşkusuz en yaygın kullanılan kütüphane olarak karşımıza çıkmaktadır. OpenCv kütüphanesi içinde bulabileceğiniz haar cascade sınıflandırıcısı; Paul Viola ve Micheal Jones tarafından, nesnelere bulmada gerekli kasnakları oluşturarak video ve resim içerisinden görüntü işleme sorunlarının çözümü için oluşturulmuştur. Aynı zamanda Viola and Jones object detection framework (Viola ve Jones nesne bulma yapısı) olarak da bilinir (Viola and Jones 2001). En temel manada belirli bir algoritmaya göre bulunması istenen nesnelere önce bilgisayara tanıtılır ve daha sonra ona benzer şekillerin bulunduğu resimler veya video kareleri taranarak o nesne bulunmaya çalışılır (İnt.Kyn.11).



**Şekil 3.2** OpenCV Haar Cascade sınıfının çalışmasındaki mantıksal şekiller.

Öncelikle sınıflandırıcının eğitimi için içerisinde aranılan nesnenin bulunduğu pozitif resimlere ve içerisinde o nesnenin bulunmadığı negatif resimlere ihtiyaç vardır. Sınıflandırıcı eğitimde, pozitif resimlerdeki nesnelere aşağıdaki gibi belirli ölçülerde ayarlanmış çerçeveleri tarayarak, çerçeve içerisinde bulunan siyah bölgedeki piksel değerleri toplamı ile beyaz bölgedeki piksel değerleri toplamından, karanlık ve aydınlık değerleri kontrol edilerek, belirli hedef değerleri oluşturulur.

*Yüz Hattı (Feature)* denilen bu çerçevelere zayıf sınıflandırıcılar denilmektedir. Çünkü tek başına doğru bir sınıflandırıcı olamazlar. Bir nesnede bu zayıf sınıflandırıcılardan birçoğu olacaktır ve bu zayıf sınıflandırıcıların toplandığı noktada büyük doğruluk oranıyla aranılan nesne var demektir. Sınıflandırıcı, en temel mantığıyla bu şekilde çalışmaktadır.

Çerçeveler aşağıdaki gibi örnek pozitif resimler üzerinde taranır.



**Şekil 3.3** Yüz üzerinde burnun belirlenmesi şekli (İnt.Kyn.12).

Yukarıdaki çerçeveye için yanakların parlaklık oranının burun bölgesindeki parlaklık oranından daha düşük olması ile burun kısmı belirlenir.



**Şekil 3.4** Haar Cascade Sınıflandırıcısı göz bölgesi belirleme (İnt.Kyn.12).

Aynı zamanda göz bölgesinin beyaz ile gösterilen alt bölgeden daha karanlık olması da bu özelliklerden bir tanesidir. Haar Cascade sınıflandırıcısında, buna benzer birçok özellikler içinde nesnenin bulunduğu resimler üzerinden geçirilerek değerler oluşturulur. Örneğin yüz taramasında ağız, burun, alın, saç gibi bölgelerde birçok karanlık veya aydınlık özellikleri oluşturulacaktır. Bunların her birinde hedef değerler oluşturmakta ve bu işlem sırasında da çerçeve büyüklüğü değişerek diğer aşamalarda tekrar etmektedir.

Gerçek zamanlı görüntü işlemede hız çok önemlidir. Haar Cascade sınıflandırıcısında öncelikle resimlerin integralleri alınır. Böylelikle piksel değerlerinin tek tek toplamları hesaplanmak yerine, integrale hesaplanarak bilgisayara büyük bir işlem gücü kazandırılacaktır.

Ayrıca nesnelerin bulunması aşamasında da her bir büyüklükteki çerçeve tarafından tekrar tekrar taranmak yerine, sadece önceki aşamalarla eşleşme olan kısımlar taranarak bir işlem yükü de ortadan kalkmaktadır. Bu açılardan ve yaptığı iş açısından oldukça hızlı oldukları söylene de uygulama aşamasında gerçek zamanlı çalışıldığında alınan görüntüyü belirli oranda yavaşlattığı görülmektedir. Bu hız sınıflandırıcının eğitilme şekli, örnek sayısı gibi durumlara göre değişmektedir. Performans açısından önemli bir etken oluşturmaktadır.

### 3.5 OpenCV Algoritmaları

OpenCV içerisinde yüz tanıma ve karşılaştırma yapabilmek için FaceRecognizer sınıfı ile gelmektedir. FaceRecognizer sınıfı türetilen tüm sınıflar için aşağıdaki özellikleri sunar (İnt.Kyn.21);

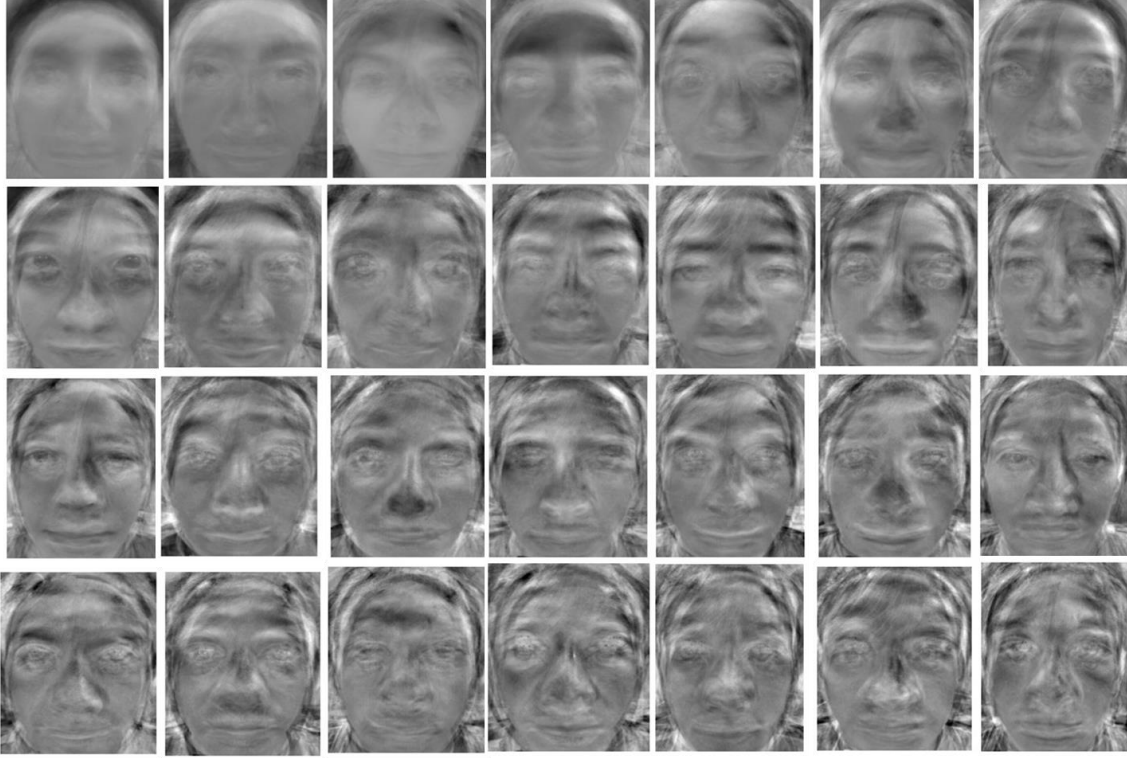
- “Sanal kurucu” (virtual constructor) ile her bir algoritma türevi program başlangıcında kaydedilir ve kayıtlı algoritmaların listesinden birini create() fonksiyonu ile oluşturulur.
- Algoritma parametrelerini OpenCV HighGUI modülünden video yakalama veya resim yakalama işlevselliği kullanılabilmektedir.
- XML veya YAML dosyalarından / dosyalarına parametre okuma ve yazma işlevi gerçekleştirilebilir. Her Algoritma türevi için tüm parametreleri saklanabilir ve daha sonra okunabilmektedir. Her seferinde yeniden uygulamaya gerek yoktur.

OpenCV FaceRecognizer sınıfı içerisinde Eigenfaces, Fisherfaces ve LBPH olmak üzere üç adet algoritma mevcuttur. Bu algoritmaların ortak amacı ve yapısı hafızadaki görüntülerle o anki tanımlanan yüz arasındaki korelasyona bakarak, tahmin işlevini gerçekleştirmektedir.

#### 3.5.1 Eigenfaces (Özyüz)

Eigenfaces algoritması, yüzün ayırt edici burun, göz, ağız, çene gibi bileşenlerinden elde edilen "hayaletsi" yüzlerdir (İnt.Kyn.22). Hafızadaki tüm resimlerin ortalamasını

olarak, ortaya çıkan görüntünün ortalamadan farkını alır. Böylece her görüntü aslında bir diferansiyel görüntü gibi değerlendirilir (Turk and Pentland 1991).



**Resim 3.5** Özyüz algoritması hayaletimsi yüz örnekleme (İnt.Kyn.13).

Özyüzlerin ardındaki motivasyon, önceki çalışmanın, sınıflandırma için hangi özelliklerin önemli olduğu sorusunu göz ardı etmesidir. Özyüzler, yüzlerin görüntülerinin ana bileşen analizini kullanarak bunu yanıtlamayı amaçlamasıdır. Bu analiz, sadece yüz tanıma için kritik olan özellikleri bırakarak, eğitim setinin boyutsallığını azaltır.

Yüzleri, galeri görüntülerini, sistem tarafından görülenleri tanımak için, her bir özyüzün bu görüntüye katkısını açıklayan ağırlıklar topluluğu olarak kaydedilir. Sınıflandırma için sisteme yeni bir yüz sunulduğunda, kendi hesaplanan ağırlıkları, görüntünün özyüzler koleksiyonuna yansıtılmasıyla bulunur. Bu ağırlıklar daha sonra en yakın eşleşmeyi bulmak için galeri setindeki tüm ağırlıklara göre sınıflandırılır. En yakın komşu yöntemi, iki vektör arasındaki Euclidean mesafesini bulmak için basit bir yaklaşımdır, burada minimum en yakın konu olarak sınıflandırılabilir (Turk and

Pentland 1991).

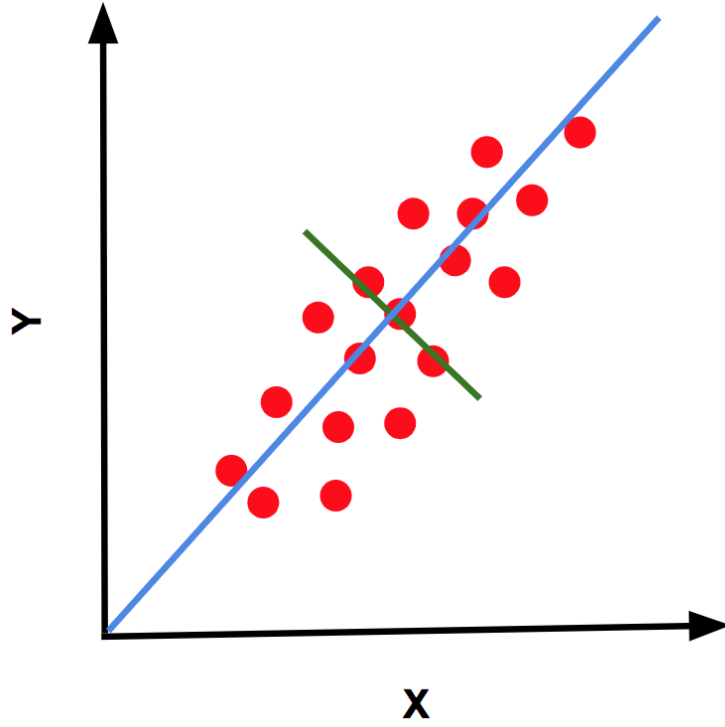


**Resim 3.6** Özyüz hayaletimsi görüntüden normal resme dönüşüm görseli (İnt.Kyn.14).

Sezgisel olarak, özyüz yöntemi ile tanıma süreci, sorgulama resimlerini, hesapladığımız özyüzler tarafından yayılan yüz boşluğuna yansıtma ve bu yüz alanında bir yüz sınıfına en yakın eşleşmeyi bulmaktır. Bu işlem için Temel Bileşen Analizi (TBA) yöntemi kullanılmaktadır.

TBA boyut azaltmak ve görüntüler arasındaki temel farklılıklar çıkarmak için kullanılan bir yöntemdir. TBA karşılıklı ilişkisi olan verinin N-boyutlu uzay yerine daha düşük boyutlu uzayda önemli bilgi kaybı olmadan boyut azaltmanın en uygun yoludur. Azaltılmış alt uzayı kullanarak, veri tabanı üzerindeki hesaplamalar daha verimli hale gelir. Üretilen betimlemeler, genel olarak öz uzaylar olarak adlandırılır ve yüz tanıma uygulamalarında ise öz yüzler olarak isimlendirilir (Turk and Pentland 1991). Öz vektörlerse, özellik vektörlerinin oluşturulması için kullanılır. Bu özellik vektörleri daha sonra tanıma işleminde kullanılarak gelen örnek görüntüsünün en çok benzediği eğitim seti elemanını bulur. TBA, tanımanın doğru yapılmasını sağlayan doğrultuların bulunmasını sağlar. Bu sınıflandırmayı

en iyi ařađıdaki řekil ifade etmektedir.



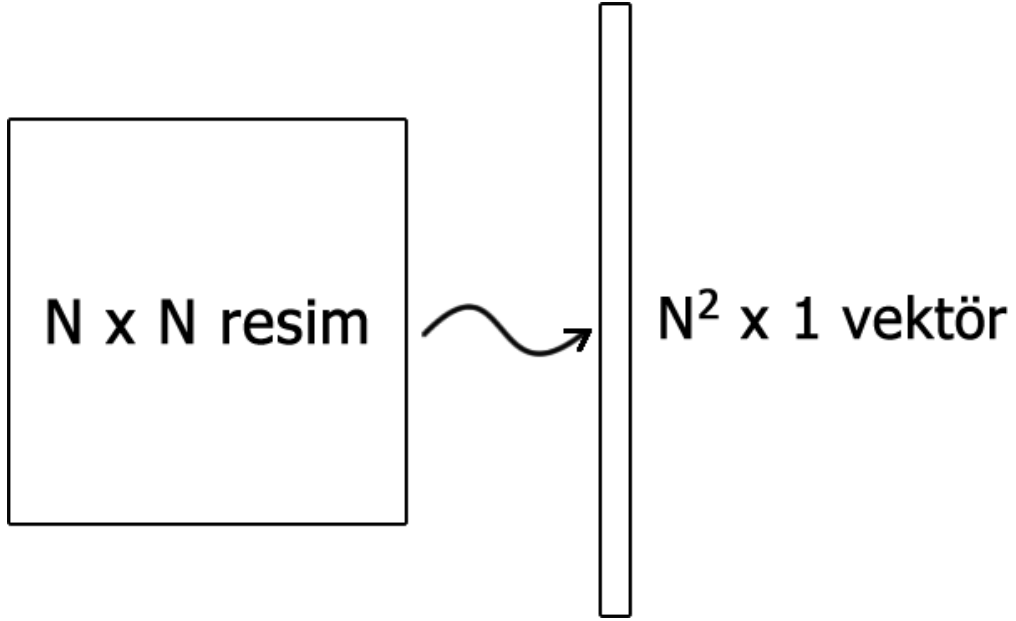
**řekil 3.5** TBA sınıflandırma 2D gösterim örneđi (İnt.Kyn.15).

TBA ayrıca sınıflar arası dađılımı maksimum yapmaktadır. řekil 3.5’de TBA örneđinin 2D gösterimi yapılmıřtır.

Dođrudan TBA'nın görüntülerin kovaryans matrisinde gerekleřtirilmesi ođu zaman hesaplanabilir deđildir. Kçük resimler kullanılıyorsa, örneđin  $100 \times 100$  piksel boyutlarında her görüntü 10.000 boyutlu bir alanda bir noktadır ve kovaryans matrisi  $S$   $10,000 \times 10,000 = 108$  öđeden oluřan bir matristir. Bununla birlikte, kovaryans matrisinin sıralaması, eđitim örneđlerinin sayısı ile sınırlıdır. Eđer  $N$  eđitim örneđleri varsa, sıfır olmayan öz deđerlere sahip en ok  $N-1$  özyüz vektörleri olacaktır. Eđitim örneđlerinin sayısı, görüntülerin boyutundan daha kçükse, ana bileřenler daha kolay hesaplanabilir (İnt.Kyn.22).

Özyüz yaklařımında, her biri  $N \times N$  boyutunda olan görüntüler,  $N^2$ 'lik vektörler haline getirilmektedir.





Şekil 3.6 NxN görüntü örneği ve vektörel dönüşümü (Turk and Pentland 1991).

Elde edilen görüntü veri tabanından öz yüzlerin elde edilebilmesi için A matrisinden kovaryans işlemi gerçekleştirilmektedir. Dönüşüm sürecinin hesaplanması 7 adımda gerçekleşmektedir:

1. Eğitim için girilen veri setleri üzerinden  $I_1, I_2, \dots, I_M$  görüntülerini elde edilir.
2. Her bir  $N \times N$  matrisi ile elde edilen görüntü  $I_i$  bir  $\Gamma_i$  vektörü temsil edilir.
3. Yüz vektörü ortalaması  $\Psi$  hesaplanır:

$$\Psi = \frac{1}{M} \sum_{i=1}^M \Gamma_i \quad (3.1)$$

4. Ortalama yüz çıkartılır:

$$\Phi_i = \Gamma_i - \Psi \quad (3.2)$$

5.  $C$  Kovaryans matrisi hesaplanır:

$$C = \frac{1}{M} \sum_{n=1}^M \Phi_n \Phi_n^T = A A^T \quad (3.3)$$

$C$  Kovaryans matrisi  $N^2 \times N^2$  matrisini oluşturan yüz ortalamalarından oluşan  $A$  matrisi dizisinden oluşmaktadır.

$$A = [\Phi_1 \Phi_2 \dots \Phi_M] \quad (3.4)$$

6. Özyüz vektörlerinin hesaplanması:

Matris  $AA^T$  çok büyük olduğundan pratik bir sonuç vermemektedir. Bu yüzden Özyüz vektörleri  $v_i$  hesaplaması:

$$A^T A v_i = m_i v_i \quad (3.5)$$

şeklinde gerçekleşir.

7. En büyük özyüz değeri olan  $K$  özyüz vektörlerden elde etme ile sonuçlanır.

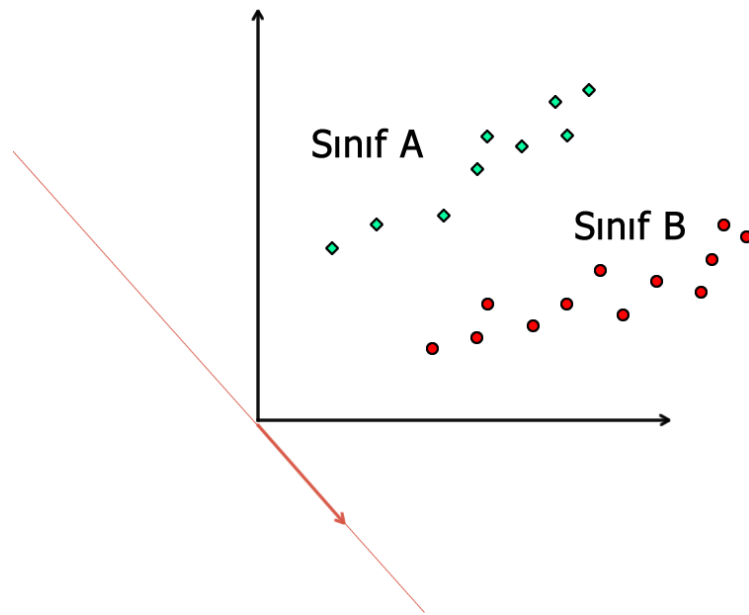
Denklemden  $C$  kovaryans matrisini ifade etmektedir. Elde edilen kovaryans matrisinin boyutu, görüntü vektörünün uzunluğu ile ilişkilidir.  $N^2$ 'lik görüntü vektörlerinden elde edilecek olan kovaryans matrisi  $N^2 \times N^2$  boyutundadır. Resim vektör boyutunun, toplam görüntü sayısından fazla olduğu koşulda ( $N^2 \gg M$ ), bu büyüklükteki bir matrisin elde edilmesi, öz değer ve öz vektörlerinin oluşturulması oldukça güçtür.  $N^2 \times N^2$ 'lik kovaryans matrisinde elde edilecek olan  $N^2$  adet öz vektörden sadece  $M-1$  tanesi anlamlı olacaktır. Anlamlı olan öz vektör sayısı, veri tabanındaki görüntü sayısı ile ilintilidir. Oluşturulan diğer öz vektörler, sıfır değerine sahip öz değerlere karşılık gelmektedir. Bu nedenle,  $N^2 \times N^2$  boyutundaki bir matrisin çözülmesi yerine  $M \times M$  boyutundaki matrisin elde edilmesi ve işlenmesi daha anlamlı olacaktır (Turk and Pentland 1991).

### 3.5.2 Fisherfaces

Eigenfaces'e nazaran daha da gelişmiş bir yüz tanıma algoritmasıdır. Fisher yüz algoritması 1997 yılında P. Belhumeur tarafından Doğrusal Ayırtaç Analizi (DAA) temel alınarak geliştirilmiştir.

Bir kişiye ait birden fazla fotoğrafı bir sınıfın üyeleri olarak kabul edip sınıflar arası tanıma işlemi yapmaya dayanır. Aynı kişiye ait birden fazla, farklı görüntünün veri tabanına tanıtılarak sınıf içi dağılımın en aza, sınıflar arası dağılımın ise en fazlaya ulaşabilmesi için geliştirilmiştir. Bu sayede, Fisher yaklaşımında tanımlama başarımlarında artış gözlenmektedir (Martinez 2011) .

Şekil 3.7’de de görüldüğü gibi DAA esas olarak veriler arasındaki ayrımı sağlayan yönün bulunmasını sağlar.



Şekil 3.7 DAA için sınıfları ayıran yön (İnt.Kyn.16).

Fisherfaces yöntemi, sınıfa özgü bir dönüşüm matrisini öğrenir, dolayısıyla Eigenfaces yöntemi olarak açık bir şekilde aydınlatma yakalamaz. Discriminant Analysis, bunun yerine, insanlar arasında ayrım yapmak için yüz özelliklerini bulur. Tabii ki, Fisherfaces'lerin performansı, giriş verilerine de bağlı olarak değişmektedir. Pratik olarak şöyle sonuçlanmaktadır, eğer sadece iyi aydınlatılmış resimler için Fisherfaces'i eğitim sırasında kullanılırsa ve kötü aydınlatılmış sahnelerdeki yüzleri tanıtmaya çalışılırsa, o zaman metodun yanlış bileşenleri bulması olasıdır. Çünkü tanımlanan hatlar, kötü ışıklı görüntülerden baskın olamayabilir. Bu yöntemin biraz daha mantıklı olmasının nedeni önceki algoritmaya göre daha net çizgilere ulaşmaya çalışmasıdır (Belhumeur *et al.* 1997).



**Resim 3.7** FisherFaces algoritması ile oluşturulmuş örneklem (İnt.Kyn.14).

Fisherfaces, yansıtılan görüntünün Eigenfaces'in yaptığı gibi yeniden tanımlanmasını sağlar. Ancak, yalnızca özneleri birbirinden ayırmak için hatları belirlendiğinden, orijinal görüntünün yeniden yapılandırılması gerekmektedir.

Fisher yaklaşımında tam olarak, her biri  $N \times N$  boyutunda olan görüntüler  $N^2$ 'lik vektörler haline getirilmekte ve veri tabanındaki tüm görüntülerden ortalama görüntü vektörü bulunmaktadır. Hesaplama aşağıdaki adım süreci dahilinde hesaplanır;

1.  $X$ ,  $c$  sınıflarından alınan örneklerle rastgele bir vektör olsun:

$$x = \{x_1, x_2, \dots, x_c\} \quad (3.6)$$

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

2. Dağılım matrisleri  $S_B$  ve  $S_W$  şu şekilde hesaplanır:

$$S_W = \sum_{i=1}^c \sum_{j=1}^{n_j} (x_{ij} - \mu_j)(x_{ij} - \mu_j)^T \quad (3.7)$$

$$S_B = \sum_{j=1}^c (\mu_j - \mu)(\mu_j - \mu)^T$$

3.  $\mu$  tüm sınıfların ortalamasını temsil eder:

$$\mu = \frac{1}{N} \sum_{i=1}^N x_i \quad (3.8)$$

4. Sınıf  $i \in \{1, \dots, c\}$  için  $\mu_i$  ortalamaları:

$$\mu_i = \frac{1}{|X_i|} \sum_{x_j \in X_i} x_j \quad (3.9)$$

5. Şimdi Fisher yönteminin klasik algoritması, sınıf ayırma ölçütünü en üst düzeye çıkaran bir projeksiyon  $W$  arar:

$$W_{opt} = \arg \max_W \frac{|W^T S_B W|}{|W^T S_W W|} \quad (3.10)$$

6. Ortaya çıkan optimizasyon probleminin çözümü, Genel Özyüz Problemini çözmek için verilir:

$$\begin{aligned} S_B v_i &= \lambda_i S_W v_i \\ S_W^{-1} S_B v_i &= \lambda_i v_i \end{aligned} \quad (3.11)$$

Çözülmesi gereken bir sorun:  $S_W$  sıralaması  $N$  örnekleri ve  $c$  sınıfları ile en fazla  $(N-c)$  ile oluşmaktadır. Örüntü tanıma problemlerinde  $N$  numunesi sayısı hemen hemen her zaman giriş verisinin boyutundan (piksel sayısı) daha fazladır. Dolayısıyla dağılma matrisi  $S_W$  tekil olur (Raudys and Jain 1991). Daha sonraları bu veriler üzerinde bir Temel Bileşen Analizi gerçekleştirilerek ve numunelerin  $(N-c)$  boyutlu uzaya yansıtılmasıyla çözüldü. Daha sonra indirgenmiş veriler üzerinde bir Doğrusal Diskriminant Analizi yapıldı, çünkü  $S_W$  artık tekil değildir (Belhumeur et al. 1997).

7. Oluşan optimizasyon problemi şu şekilde yeniden yazılabilir:

$$W_{pca} = \arg \max_W |W^T S_T W| \quad (3.12)$$

$$W_{fld} = \arg \max_W \frac{|W^T W_{pca}^T S_B W_{pca} W|}{|W^T W_{pca}^T S_W W_{pca} W|}$$

8. Dönüşüm matrisi  $W$ , (c-1) boyutlu alana bir örnek yansıtması daha sonra şu şekilde verilir:

$$W = W_{fld}^T W_{pca}^T \quad (3.13)$$

Bu hesaplanma sürecinde  $S_B$ : sınıflar arası saçılma matrisidir.  $S_W$  ve  $S_B$  matrislerinden elde edilen saçılma matrisleri 3.9 denklemlerindeki formülde kullanılarak 3.13 denklemindeki gibi en iyi  $W$  matrisi bulunması amaçlanmaktadır. Bu sayede, sınıflar arası saçılma en fazlaya, sınıf içi saçılma ise en aza indirgenmektedir (Martinez and Kak 2001).

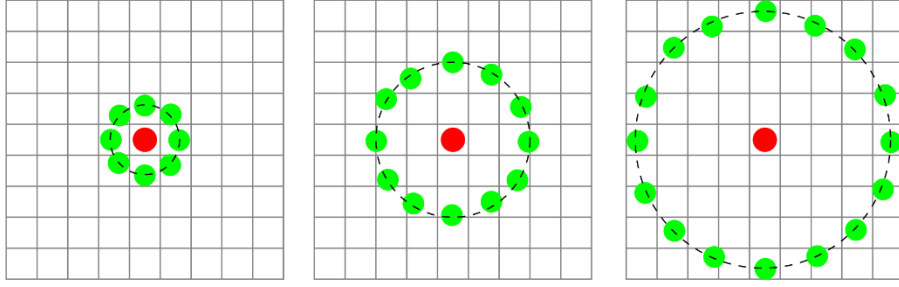
### 3.5.3 LBPH

Türkçesi Yerel İkili Örüntü olarak adlandırılan LBP, bilgisayar görüşünde sınıflandırmak için kullanılan bir görsel tanımlayıcı türüdür ve 1990 yılında önerilen Doku Spektrum Modelinin adı verilen modelin özel durumudur (He and Wang 1990). Tam olarak LBP ilk kez 1994'te tanımlanmıştır (Ojala *et al.* 1996). O zamandan beri nesne sınıflandırması için güçlü bir özellik olduğu anlaşılmıştır. Ayrıca, LBP'nin "Yönlendirilmiş Gradyanların Histogramı" (HOG) tanımlayıcısı ile birleştirildiği de belirlenmiştir. Bazı veri kümelerinde algılama performansını önemli ölçüde geliştirmiştir (Wang *et al.* 2009).

LBP ile uzamsal olarak geliştirilmiş özellik vektörü daha sonra yerel histogramların birleştirilmesiyle elde edilir. Bu histogramlara LBPH denir. LBPH, yerel ikili desen histogramları manasına gelen Local Binary Patterns Histograms kelimelerinin baş harflerinden oluşmuştur.

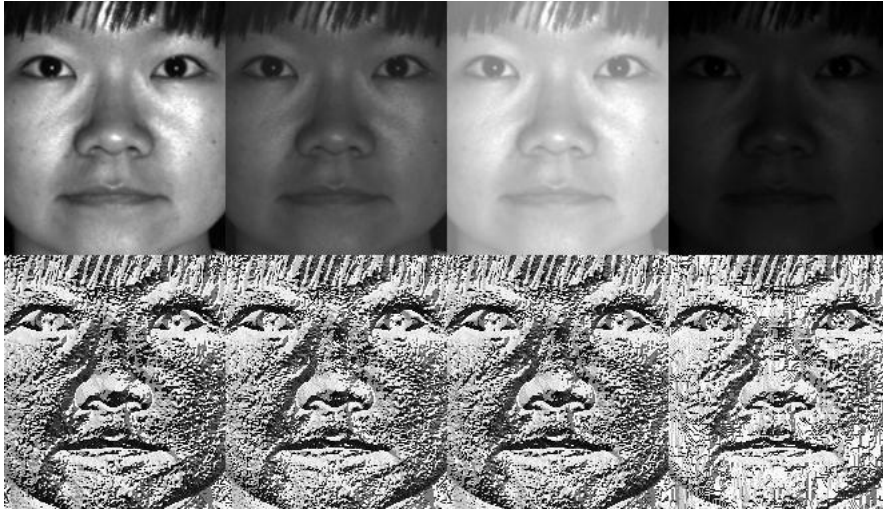
Bir noktanın etrafındaki piksellerin yoğunluk değerlerine göre oluşan ikili örüntülerden

yaratılır ve bu örüntüler üzerinde işlemler gerçekleştirerek nesneyi aramaktadır.



**Şekil 3.8** Bir doku tanımlamak ve bir yerel ikili desen (LBP) hesaplamak için kullanılan üçlü komşu örneği (İnt.Kyn.23).

Tanım olarak LBP, monotonik gri ölçekli dönüşümlerle sağlamaktadır. Yapay olarak değiştirilmiş bir resmin LBP görüntüsüne bakarak bunu kolayca doğruluya bilmektedir. Resim 3.8’de bir LBP görüntüsünün nasıl görüldüğünü örneklendirilmiştir.



**Resim 3.8** LBPH yöntemi ile elde edilmiş örnek görsel (İnt.Kyn.14).

LBPH, bir noktanın etrafındaki piksellerin yoğunluk değerlerine göre oluşan ikili örüntülerden monotonik gri ölçekli dönüşümler ile hesaplama yapmaktadır. Hesaplama aşağıdaki adım süreci dahilinde hesaplanır;

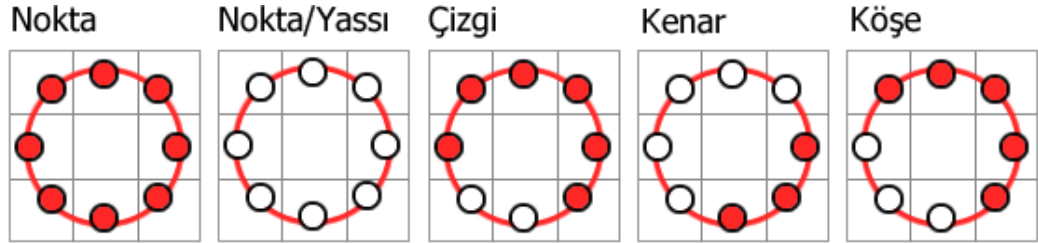
1. LBP operatörü formülü genel olarak şu şekildedir:

$$LBP(x_c, y_c) = \sum_{p=0}^{P-1} 2^p s(i_p - i_c) \quad (3.14)$$

2.  $(x_c, y_c)$  merkezi pikseli,  $i_p$  ve  $i_c$  komşu piksellerinin yoğunluğunun başlangıç noktasını oluşturmaktadır. Buna göre  $s$ , aşağıdaki gibi tanımlanan işaret işlevidir:

$$s(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{else } x < 0 \end{cases} \quad (3.15)$$

Bu durum görüntülerde çok ince taneli detayları yakalamayı sağlar. LBP operatörün yayınlanmasından kısa bir süre sonra, sabit bir komşunun örtüğü oluşturduğu sabit ölçekte farklı detayları kodlayamadığı gözlemlendi. Böylece LBP operatörü bir değişken komşu veri tipi kullanacak şekilde genişletildi. Buradaki fikir, Şekil 3.9'da komşuları yakalayabilmenizi sağlayan değişken bir yarıçapa sahip bir çevredeki anormal sayıda komşunun takip edilmesidir.



Şekil 3.9 LBP operatörünün komşu veri tipleri (İnt.Kyn.14).

3.  $(x_c, y_c)$  noktası  $(x_p, y_p)$  komşu noktalarını verebilmesi için  $p \in P$  hesaplanması:

$$x_p = x_c + R \cos\left(\frac{2\pi p}{P}\right) \quad (3.16)$$

$$y_p = y_c + R \sin\left(\frac{2\pi p}{P}\right)$$

3.3'teki denklemlerde kullanılan  $R$ , dairenin yarıçapıdır ve  $P$ , örnek noktaların sayısıdır.

4. Son aşamada bir dizi akıllı enterpolasyon şemasına sahip, çift yönlü



enterpolasyon yapılır:

$$f(x,y) \approx [1 - x \ x] \begin{bmatrix} f(0,0) & f(0,1) \\ f(1,0) & f(1,1) \end{bmatrix} \begin{bmatrix} 1 - y \\ y \end{bmatrix} \quad (3.17)$$

### 3.6 Yazılımı Oluşturan Mimari ve Hazırlık

Yazılımı oluşturan kod yapısı ana hatları ile sıralı bir şekilde verilmiştir.

#### 3.6.1 Kameradan Görüntü Alma

Öncelikle Raspbery Camera Modülü üzerinden görüntü yakalamak gerekmektedir. Görüntüyü almak için aşağıdaki komutlar yeterli olacaktır.

```
camera = PiCamera()
camera.resolution = (640, 480)
camera.framerate = 32
rawCapture = PiRGBArray(camera, size=(640, 480))
```

Alınan bu resimler karşılaştırmaya yönelik üç algoritma için çalıştırılacaktır. Her algoritma için aynı denek için beş ayrı anlık görüntü bir saniye ara ile alınarak karşılaştırma tablosuna eklenecektir.

#### 3.6.2 Kameradan Yüz Bulma

OpenCV yapısal olarak video ve resimlerden doğrudan yüz bulabildiği gibi kamera üzerinden gerçek zamanlı yüz bulabilmektir. Bu işlem kamera üzerinden gelen görüntüyü frame başına yakalanan resimler üzerinde tarama yapılarak gerçekleştirilmektedir. Bu işlemi yaptığımız kodlama örneği aşağıda açıklanarak verilmiştir.

```
# Video görüntüsü yakalanmaya başlanır
vid_cam = cv2.VideoCapture(0)
```

```

# Haarcascade ile eğitilmiş kasnak ile yüz görüntüden tanımlama
yaptırılır
face_detector =
cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

# Döngü Başlatılarak her yakalanan görüntü incelenir.
while(True):

    # Hazırda duran görüntü kameradan belleğe alınır.
    _, image_frame = vid_cam.read()

    # resim Gri versiyona çevrilmesi için tanımlama yapılır.
    gray = cv2.cvtColor(image_frame, cv2.COLOR_BGR2GRAY)

    # Eğitilmiş sınıfa göre farklı boyutlardaki yüzler gri şekilde
    listeye çekilir.
    faces = face_detector.detectMultiScale(gray, 1.3, 5, minSize=(25,
25))

    # Bulunan yüzleri alabilmek için tekrar döngü başlatılır
    for (x,y,w,h) in faces:

        # belirlenen yüz resmi aktif görüntü içerisindeki lokasyonda
        gösterilmek için çerçeve oluşturulur.
        cv2.rectangle(image_frame, (x,y), (x+w,y+h), (255,0,0), 2)

        # diğer resim için sayaç artırılarak döngü devam ettirilir.
        count += 1

        # Bulunan yüz video üzerinde gösterilir.
        cv2.imshow('frame', image_frame)
# video durdurulur
vid_cam.release()

# programdan çıkılır.
cv2.destroyAllWindows()

```

Aslında Python ve OpenCV kullanarak, bir yüz algılamak için gereken kod sadece birkaç satırdır. Bu durumun en büyük sebebi OpenCV ile kullanılan haarcascade sınıfı

içerisinde tanımlı yüz tanımlama modellerinin hazır bir şekilde olmasıdır.

Yukarıda bulunan kodun en önemli kısmı aşağıdaki satırıdır.

```
faces = face_detector.detectMultiScale(  
    gray,  
    scaleFactor = 1.3,  
    minNeighbors = 5,  
    minSize = (25, 25)  
)
```

Bu kod OpenCV kütüphanesi içerisinde bulunan görüntü işleme fonksiyonları ile etkileşime giren haarcascade sınıfı içerisindeki yüz tanıma modellerine bakarak resim veya kameradan gelen görüntüyü saliseler içerisinde inceleyerek yüz ifadelerini bir liste olarak getirmektedir. “detectMultiscale” fonksiyonu işlemini yapabilmesi için bir takım parametrelere ihtiyaç duymaktadır. Sırası ile incelenecek olunursa (İnt.Kyn.17);

1. Gray (gri), adından da anlaşıldığı gibi sonucu gri tonlu görüntü olarak getirmesi içindir.
2. scaleFactor, görüntü boyutunun her bir görüntü ölçeğinde ne kadar azaldığını veya arttığını belirleyen parametredir. Ölçek piramidi oluşturmak için kullanılır.
3. minNeighbors, her aday dikdörtgenin kaç tane komşusunun onu sahip olması zorunda olduğunu belirten parametredir.

Bu parametre tespit edilen yüzlerin kalitesini etkileyecektir. Daha yüksek değer daha az algılama ile sonuçlanır, ancak daha yüksek kalite elde etmek için 3 ~ 6 değer aralığı kullanılması uygun olacaktır.

4. minSize, kamerada veya tanımlı resimde olması istenen minimum değere sahip bulunan yüz boyutudur. Bundan daha küçük nesnelere sayılır.

### 3.6.3 Kameradan Yüz Tanıma

OpenCV algoritmaları karşılaştırmada 3 ayrı algoritmayı aşağıdaki kod üzerinde çalıştırmak mümkündür.

```
# Local Binary Patterns Histograms ile Yüz tanıma sınıf tanımlama  
recognizer = cv2.face.createLBPHFaceRecognizer()
```

```

# Tanıma için Eğitilmiş modu yükle
recognizer.load('trainer/trainer.xml')

# Yüzü bulabilmek için önceden oluşturulmuş Haar Cascade sınıfı
cascadePath = "haarcascade_frontalface_default.xml"

# Önceden oluşturulmuş modelden sınıflandırıcı oluştur
faceCascade = cv2.CascadeClassifier(cascadePath);

# tanımlanmış Yüz üzerine yazılacak isim için fontunu tanımlama
font = cv2.FONT_HERSHEY_SIMPLEX

# Tekrar video çağırma
cam = cv2.VideoCapture(0)

# Döngü
while True:
# Kameradaki o anki görüntü resim olarak alınır.
ret, im =cam.read()

# Karşılaştırmada ışıktan etkilenmemek için görüntüyü griye çevrilir.
gray = cv2.cvtColor(im,cv2.COLOR_BGR2GRAY)

# Tüm Yüzler tespit edilir
faces = faceCascade.detectMultiScale(gray, 1.2,5)

# Bulunan tüm yüzler için döngü başlatılır.
for(x,y,w,h) in faces:

# yüzün etrafını dikdörtgen içine alma
cv2.rectangle(im, (x-20,y-20), (x+w+20,y+h+20), (0,255,0), 4)

# Yüzü tanımayla çalışma ile kayıtlı resimler arasından id tespit etme
Id = recognizer.predict(gray[y:y+h,x:x+w])

# kayıtlı id değerine göre isim verme
if(Id == 1):
Id = "fatih"
elif(Id == 2):

```

```

Id = "behzat"
else:
    Id = "Unknow"

# Ekranda çerçeveye alıp ismi yazma
cv2.rectangle(im, (x-22,y-90), (x+w+22, y-22), (0,255,0), -1)
cv2.putText(im, str(Id), (x,y-40), font, 2, (255,255,255), 3)

# yazılı veriyi ekranda gösterme
cv2.imshow('im',im)

```

Örneği incelediğimizde bu yüz tanımlama `createLBPHFaceRecognizer` algoritması ile yapıldığını görülecektir. Diğer algoritmalar için;

```
recognizer = cv2.face.createLBPHFaceRecognizer()
```

yazan komutta kullanılan `createLBPHFaceRecognizer()` yerine `createEigenFaceRecognizer` ve `createFisherFaceRecognizer` sınıflarını kullanılması gerekmektedir. Böylelikle diğer algoritmaları da kullanarak değerlendirme yapılması sağlanabilir.

Yüz karşılaştırmada OpenCV kütüphanesi içerisinde asıl işlevi aslında tek bir satır yapmaktadır.

```

Id = recognizer.predict(
    gray[y:y+h,x:x+w]
)

```

`Predict` komutu `detectMultiScale` komutu ile bulunan gri resimleri veri seti içerisindeki resimlerle karşılaştırarak benzerlik oranı en yüksek olanların ilgili id sonucunu dönmektedir. Konut id sonucu ile birlikte benzerlik oranını da vermektedir.

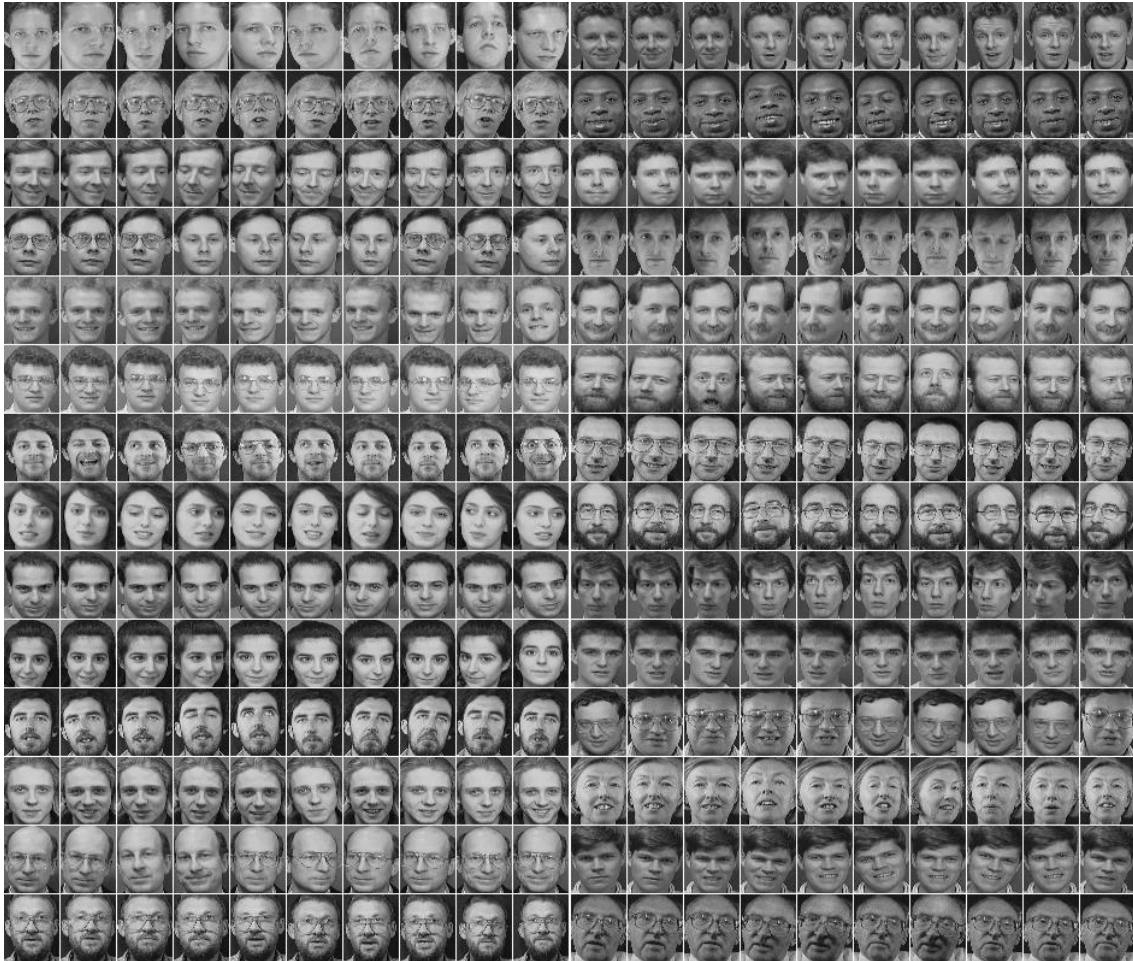
### 3.6.4 OpenCv Haar Cascade Sınıflandırıcısı Eğitimi

OpenCv Haar Cascade yöntemi kullanarak; yüz, araç, el gibi istediğiniz türden farklı nesnelere sınıflandırıcıya tanıtıp, farklı ortamlardaki resimlerde taratarak

buldurmaktadır. Bu sınıflandırıcının eğitimi için pozitif ve negatif resimlere ihtiyacımız bulunmaktadır. Sınıflandırıcınız ne kadar çok örnek ile eğitirse, size de o kadar iyi sonuç elde etmenizi sağlayacaktır.

Pozitif resimler, içerisinde bulmak istediğiniz nesnelere bulduğu örneklerdir. Bu örnekler farklı parlaklık ve farklı açılarla olursa size daha efektif bir sınıflandırıcı oluşturacaktır. İyi bir sınıflandırıcı için binlerce örnek ile eğitilmesi gerekmektedir.

Daha sonra bulunması istenilen, objenin tanımlı olmadığı fotoğraflar toplanması gereklidir. Bu fotoğraflar herhangi bir resim, manzara ve benzeri olabilir. Önemli olan hedefin üzerinde obje olmamasıdır. Bu işlem için gerekli olan ve kullanılan resimlere "negatif resimler" denmektedir.



**Resim 3.9** Haar Cascade sınıflandırıcısı için örnek pozitif resimler (İnt.Kyn.19).

Sınıflandırıcının efektif olarak nesneyi bulabilmesi için, binlerce pozitif resme ihtiyaç duyduğunu belirtilmişti. 2000 ile 4000 civarındaki pozitif resim sayısının iyi bir sonuç vermesi için, en az seviyede yeterli olduğu yapılan birçok denemede gözlenmiştir. Negatif resimlerin de 800 ile 1000 gibi bir adette olması iyi olabilecektir. Çünkü sınıflandırıcı pozitif resimleri negatif resimler üzerinde işlemektedir. Ancak bu kadar fazla negatif resim zorunlu değildir. Daha azıyla da sınıflandırıcı oluşturulabilir. Sınıflandırıcıya önceden hazır olması için pozitif ve negatif klasörler oluşturulur. Örneklerle doldurduktan ve txt dosyaları içerisinde ayrı ayrı liste şeklinde dosya yolları ve isimleri kaydettikten sonra, pozitif resimleri Haar Cascade sınıflandırıcısında kullanabilmek için gri skalaya resimlere çevrilmesi ve nesnenin bulunduğu kısmı kırarak bir vektör dosyası oluşturulması gerekmektedir. Bunun için OpenCV içerisinde bulunan `opencv_createsamples.exe` dosyası bu iş için hazırlanmıştır. Bu exe dosyası parametre olarak pozitif resimlerin isim ve adres yolunun bulunduğu txt dosyasını, pozitif resim sayısını, oluşturulacak olan vektör dosyasının ismini ve örneklerin boyutlandırılacağı oran değerlerini almaktadır.

Parametre değerlerini `opencv_createsamples.exe` dosyasına gönderebilmek için işletim istemlerinde bulunan komut satırı uygulamaları veya ekranından çalışılır ya da `.bat` uzantılı bir toplu işlem dosyası oluşturulur ve komut satırı penceresine yazılacak komutlar kaydedilerek dosya çalıştırılır.

Son aşamada sınıflandırıcının eğitim aşamasına gelinmektedir. Bu eğitim OpenCV kütüphanesinde bulunan iki exe dosyası ile yapılabilmektedir. Bu dosyalar `opencv_haartraining.exe` ve `opencv_traincascade.exe` dosyalarıdır. `opencv_haartraining.exe`, sadece Haar yöntemini kullanan eski algoritmaları kullanmaktadır. `opencv_traincascade.exe` ise en güncel sınıflandırıcı eğitim dosyası olup hem Haar yöntemi hem de Local Binary yönteminde eğitim yapabilmektedir.

```

<opencv_storage>
<cascade_type_id="opencv-cascade-classifier"><stageType>BOOST</stageType>
  <featureType>HAAR</featureType>
  <height>24</height>
  <width>24</width>
  <stageParams>
    <maxWeakCount>211</maxWeakCount></stageParams>
  <featureParams>
    <maxCatCount>0</maxCatCount></featureParams>
  <stageNum>25</stageNum>
  <stages>
    <_>
      <maxWeakCount>9</maxWeakCount>
      <stageThreshold>-5.0425500869750977e+00</stageThreshold>
      <weakClassifiers>
        <_>
          <internalNodes>
            0 -1 0 -3.1511999666690826e-02</internalNodes>
          <leafValues>
            2.0875380039215088e+00 -2.2172100543975830e+00</leafValues></_>
        <_>
          <internalNodes>
            0 -1 1 1.2396000325679779e-02</internalNodes>
          <leafValues>
            -1.8633940219879150e+00 1.3272049427032471e+00</leafValues></_>
      <_>
    </_>
  </stages>
</opencv_storage>

```

Şekil 3.10 Örnek Haar Cascade XML dosyası içeriği.

Pozitif resimleri, negatif resimler üzerine işleyerek belirlenen minhitrate ve maxfalsealarmrate değerlerine ulaştırmaya çalışır. Her bir aşamada pozitif resimler farklı çerçevelerle taranarak değerleri oluşturulur. Belirlenen aşama değerine ulaşıldığında ise nesneyi bulmak için kullanılabilir .XML dosyası oluşacaktır.

### 3.6.5 Yüz Veri Setleri ve Eğitimi

Bu araştırmada OpenCV içerisinde bulunan Eigenface, Fisherface ve LBPH algoritmalarının karşılaştırmaları yapabilmek için Raspberry PI 3 gömülü ortamı içerisinde öncelikle veri setlerini oluşturmak gerekmektedir. Bu karşılaştırma için öncelikle yüzün kaydedilmesi gerekmektedir. Bu kayıt ve eğitim OpenCV standardı ile gelen haarcascade\_frontalface\_default.xml yüz tanımlama sınıflandırıcısı gelen görüntüdeki yüzleri tespit ederek ve tanımlanarak gri resimler halinde alınması ile gerçekleşir. Amaç taranacak yüzün doğru kişiye ait olup olmadığının



sınıflandırılmasıdır.

Eigenface, Fisherface algoritmaları için eğitim yapmadan doğrudan veri seti kullanarak tanımlama yapmak yeterli olmaktadır. Eğitim Eigenface, Fisherface algoritmalarına göre biraz daha kompleks olan LBPH algoritması için gerekmektedir. Bu da LBPH algoritmasının daha kesin sonuçlar çıkartmak için tasarlanmış bir algoritma olmasından kaynaklanmaktadır.

Veri setlerinin oluşturulmasında “3.6.2 Kameradan Yüz Bulma” başlığı altında bulunan yüz bulma kodu ile bulunan yüzleri oluşturduğumuz bir klasöre kaydetmekteyiz. Bu işlemde veri setinin kalitesi için aynı kişiye ait yüzün değişik açılardan alınmış resimlerinin oluşturulması önem teşkil etmektedir. Çünkü kamera üzerinden gerçek zamanlı yüz bulma işleminde farklı açılarda bile yüzün tanımlanmasını ve benzerlik oranının üst seviyede çıkması önemlidir.

### **3.6.5.1 Gömülü Ortamda Veri Seti Hazırlama ve Performans**

Gömülü ortamlar performans açısından oldukça düşük kapasiteye sahip mini bilgisayarlardır. Bu mini bilgisayarların ekran kartı işlemcisi ile merkezi işlemcisi soğutucuya ihtiyaç duymayan basit işlemler için tasarlanmış olduklarından büyük kapasiteli işlemleri yapmak için tasarlanmamışlardır. Bu araştırma için kullanmış olduğumuz Raspberry marka PI 3 modeli diğer gömülü ortamlar için hazırlanmış kartlara göre en güncel olan cihazlardandır. Raspberry tümleşik ana kartı ile ekran kartını, RAM'ı, merkezi işlemcisini, kablosuz ağ, USB bağlantıları gibi kısacası bir bilgisayarda olması beklenen giriş ve çıkış ünitelerini kendi üzerine barındırmaktadır. Cihaz ARM mimarisi ile oluşturulmuş dört çekirdekli 1.2 GHz. işlem gücüne, 1 GB 900 MHz. RAM ve 400MHz ekran kartı işlemcisiyle piyasada çalışan yeni nesil tabletlere eş değer bir güce sahiptir.

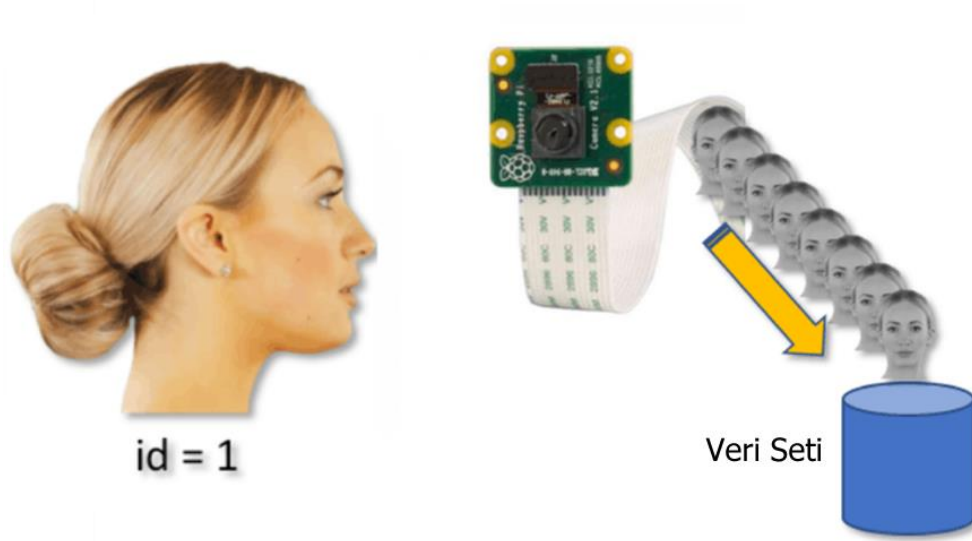
Veri setini hazırlamak için cihazın yanında onun için üretilmiş ve ideal işletim sistemi olan Raspian işletim sistemi kullanılmıştır. Bu işletim sistemi tamamen ARM işlemcisine göre tasarlanmış bir yapıda çalışan özel bir Linux işletim sistemi platformudur.

Karşılaştırmamızın kaynak kodlarını oluşturan görüntü işleme için oluşturulmuş OpenCV kütüphanesi uzunca yıllardır geliştirilmekte olan bir kütüphanedir. OpenCV kütüphanesi C++ programlama dili ile geliştirilmiştir. C++ dili yapısı itibariyle tüm işlemci mimarileriyle çalışabilmektedir.

Araştırma da Raspberry PI 3 cihazı üzerine Rasbian işletim sistemini kurduktan sonra OpenCV kütüphanesini doğrudan kullanılamamaktadır. OpenCV kütüphanesinin hali hazırda derlenmiş olan versiyonları 32 bit ve 64 bit Intel tabanlı bilgisayarlar içindir. Arm işlemcili bilgisayarların işletim sistemi içerisinde kullanabilmesi ve Python dili içerisinde referans kütüphane olarak çalışabilmesi açısından derlenmesi gerekmektedir.

OpenCV kurulumu için öncelikle <http://opencv.org> adresinden kaynak kodlarını indirilmelidir. İndirilen kaynak kodları birkaç adımla kurulmaktadır. Bu süreç yaklaşık 4-6 saat arasında sürmektedir. Bunun en büyük sebebi ise kütüphanenin işletim sistemi üzerine kurulu ve Python programlama dili araçlarının içerisine eklenmesidir.

Tüm kurulum bittikten sonra Python 3 ortamı içerisinde veri modelimizi tanımlayan kodu çalıştırarak kameradan görüntü alınması ve alınan resmin kaydı yapılır. Kayıt işlemlerini ne kadar çok resim ile yapacak olunur ise o kadar sonuç verimli çıkmaktadır. Örneğin aynı yüz için 20 resim ile yapılan kayıt ile 200 resim ile yapılan kayıt arasında yüz tanımda verim farkı olacaktır.



Şekil 3.11 Raspberry Pi ile veri seti kaydı.

Şekil 3.11'den de anlaşılacağı gibi veri setlerini her yüz için ayrı ayrı oluşturmak gereklidir.

### 3.6.5.1 Gömülü Ortamda Veri Setlerinin Eğitimi

Yüz tanıma işlemini gerçekleştirmek için, önceden kaydedilmiş veri kümesi kullanarak bir yüz tanıyıcının eğitilmesi gerekir. Yüz tanıma sistemi için tanımlı veri kümesini oluşturduktan sonra OpenCV kütüphanesi içerisinde bulunan “train” sınıfı ile eğitim gerçekleştirilmektedir. Eğitim için kullanılan bir yüze ait veri setinde bulunan resim sayısının çok olması eğitimi daha iyi sonuç vermesini sağlayacaktır. OpenCV kütüphanesinin eğitim sınıfı çalıştığında program içerisinde belirtilen klasör içerisine belirtilen bir isim ile yml uzantı bir dosya oluşturur.

Oluşan eğitim dosyasının içeriğine bakıldığında veri setlerini oluşturan resimlerden elde edilen yerel ikili örüntü histogramların bilgisinin yer aldığı görülmektedir. Ayrıyeten komsu nokta sayısının yanı sıra, taranın resmi 8 x 8 matrisler halinde bölüğü gözükmemektedir.

```

%YAML:1.0

radius: 1
neighbors: 8
grid_x: 8
grid_y: 8
histograms:
  - !!opencv-matrix
    rows: 1
    cols: 16384
    dt: f
    data: [ 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
      0., 2.77777780e-02, 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
      2.77777780e-02, 0., 0., 1.38888896e-01, 0., 8.33333358e-02,
      2.77777780e-02, 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
      0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
      1.38888896e-01, 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
      0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
      0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
      0., 0., 0., 0., 0., 0., 2.77777780e-02, 0., 0., 0., 0., 0.,
      0., 0., 0., 0., 0., 0., 5.55555560e-02, 0., 2.77777780e-02,
      2.77777780e-02, 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
      0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
      0., 0., 0., 2.77777780e-02, 0., 0., 0., 0., 0., 0., 0., 0.,
      0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
      0., 0., 0., 0., 0., 0., 0., 0., 2.77777780e-02, 0.,
      2.77777780e-02, 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
      0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
    ]

```

**Şekil 3.12** OpenCV eğitim sonrası oluşan yml uzantılı dosya içeriği.

Eğitim datasını oluştun kod aşağıdaki gibidir;

```

import cv2, os
import numpy as np

# Python Image Library (PIL) ekleme
from PIL import Image

# Yüz karşılaştırma için Local Binary Patterns Histograms sınıfı
eklenir.
recognizer = cv2.face.createLBPHFaceRecognizer()

# Yüzü bulabilmek için önceden oluşturulmuş Haar Cascade sınıfı
detector =
cv2.CascadeClassifier("haarcascade_frontalface_default.xml");

# Resim ve Etiket datası için fonksiyon oluşturulur

```

```

def getImagesAndLabels(path):

    # Tüm dosyalar hafızaya alınır
    imagePath = [os.path.join(path,f) for f in os.listdir(path)]

    # Boş faceSamples listesi oluşturulur
    faceSamples=[]

    # Boş ids listesi oluşturulur
    ids = []

    # Tüm dosyalar için döngü çalıştırılır
    for imagePath in imagePath:

        # Gri tona çevrilerek resim dosyadan okunur
        PIL_img = Image.open(imagePath).convert('L')

        # resim image data tipine çevrilir
        img_numpy = np.array(PIL_img,'uint8')

        # Split fonksiyonu ile dosya isminden id bulunur
        id = int(os.path.split(imagePath)[-1].split(".")[1])
        print(id)

        # Image datası içerisindeki yüzler tespit edilir
        faces = detector.detectMultiScale(img_numpy)

        # Tespit edilen yüzler için döngü çalıştırılır.
        for (x,y,w,h) in faces:

            # faceSamples listesine yüz eklenir.
            faceSamples.append(img_numpy[y:y+h,x:x+w])

            # yüz ile eşlenen id ids listesine eklenir
            ids.append(id)

    # bulunan yüz listesi ve id listesi fonksiyon için dönüş değeri
    olarak verilir
    return faceSamples,ids

```

```
# Dosya yolu verilerek tanımlanan fonksiyon ile yüz ve id listesi
çağrılır
faces,ids = getImagesAndLabels('dataset')

# OpenCV karşılaştırma için oluşturulmuş eğitim metodu çalıştırılır
recognizer.train(faces, np.array(ids))

# karşılaştırma için hazırlanmış eğitim histogramları yml uzatılı
dosyaya kaydedilir.
recognizer.save('trainer/trainer.yml')
```

## 4. BULGULAR

Bu araştırma ile Raspberry Pi 3 üzerinde OpenCV kütüphanesi kullanılarak program geliştirilmiştir. Geliştirilen yazılımla aynı ortam ışığında yüz tanıma ve bulma algoritmaları karşılaştırılmaya çalışılmış ve sonuçları bu kısımda açıklanmıştır.

### 4.1 Yüz veri Setleri Eğitim Performansı

İlk bakışta gözle görülür bir şekilde gömülü ortam ile masa üstü bilgisayar arasında OpenCV kütüphanesinin derlenmesinde ortaya çıkmaktadır. Masa üstü bilgisayarlarda dakikalar içerisinde yapılan derleme gömülü ortamda saatlerce beklemeye sebep olmaktadır. İşlemci gücü ayrımı bu noktada belirgin bir şekilde gözükmemektedir.

Araştırmamızda veri setinin oluşturulmasında örnek resimlerin sayısını 20 adetten başlayarak 1000 adete kadar alınmasında performans ölçümlemesine bakılmıştır. Veri setleri oluşturma işlemi OpenCV için tüm algoritmalarda aynı yöntem ile yapılmaktadır.

**Çizelge 4.1** Veri setindeki resim sayısına göre eğitim.

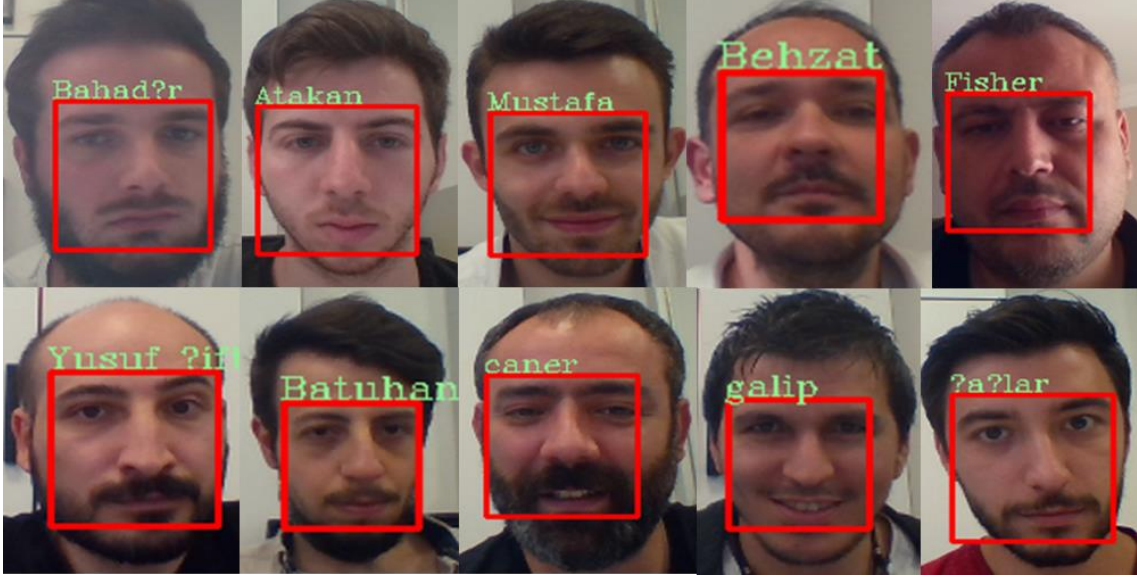
Resim Adeti	Elde Etme Zamanı(dd:ss.sn)
20	00:07.49
50	00:19.51
100	00:34.10
250	01:20.46
500	08:46.32
1000	12:50.35

### 4.2 OpenCV Algoritmalarının Denekler ile Karşılaştırılması

#### 4.2.1 Araştırma Bölgesi ve Denekler

Araştırma bölgesi olarak gönüllü on bireyle çalışma yapılmıştır. Geliştirilen yazılım

gönüllü denekler üzerinde Eigenface, Fisherface ve LBPH algoritmaları kullanılarak yapılmıştır.



**Resim 4.1** Gönüllü deneklerin tek bir karede görünümü.

Çalışmamız sırasında kullanılan Haar Cascade sınıfı OpenCV ile birlikte gelen yüz bulma ve tanıma için geliştirilmiş sınıftır. Bu sınıf dünya genelinde standartları yansıtan bir değer olarak kabul edilir. Daha önceden anlattığımız OpenCV Haar Cascade sınıflandırıcı eğitimi ile görüntü işleme çalışması için oluşturulabilir. Bu oluşturma için elinizde oldukça fazla, belli açılardan çekilmiş simalara ihtiyaç bulunmaktadır. Resimler ne kadar fazla olur ise sınıfın eğitilmesi ile birlikte, yanlışma oranının da daha düşük olması büyük önem kazanmaktadır.

Bulgular için tüm denekler bir ortam içinde, aynı ışık açısı, aynı konum ve aynı açı ile çalışmaya tabi tutulmuştur.

#### **4.2.2 EigenFace ile Test Bulguları**

Şekil 4.1’de deneklerin bir saniye ara ile alınmış yüzleri bulunmaktadır. Alınan yüzlerin benzerlik puantajı resimlerin altlarında gözükmektedir. Bu sonuçları karşılaştırma işlevini aşağıdaki tabloda detaylı incelersek, aynı kişi için o anki yüz ifadesine göre



alınan beş farklı resimde çıkan sonuçların birbirini tutmadığı hatta çok farklı rakamlarda verdiği görülmektedir. Bu duruma ana etken, ışığın anlık açısı ve kameranın netliğinin o anlık durumudur. Gözükten kişinin yüzüne farklı bir açıdan baktığından dolayı, farklı sonuçlar vermektedir.

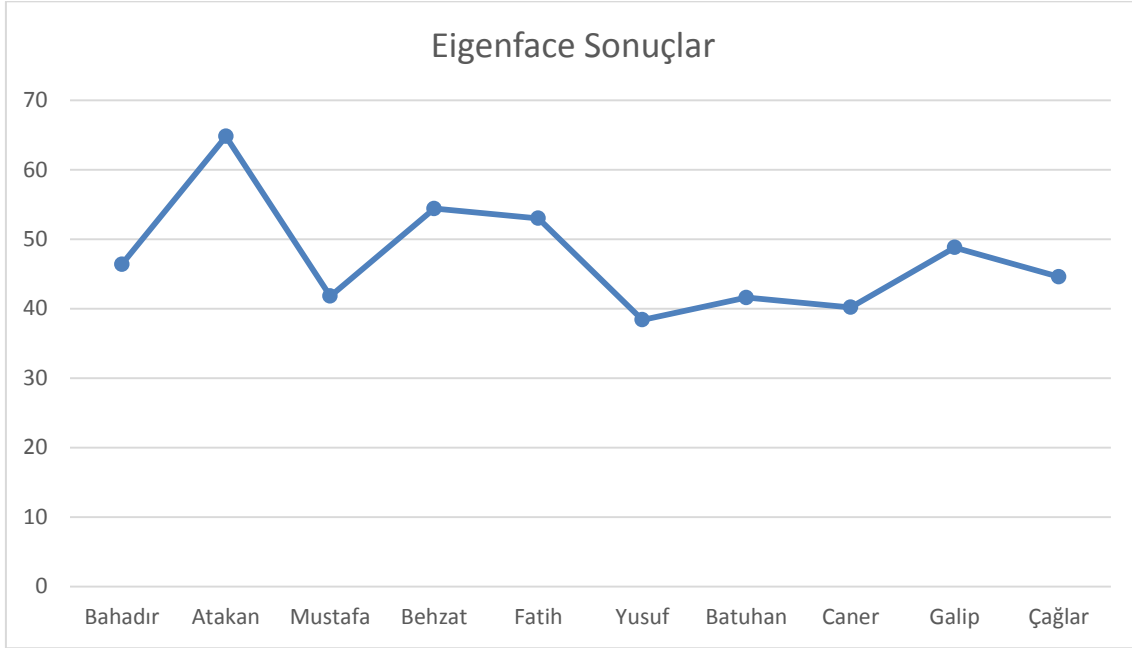


Şekil 4.1 Eigen algoritmasına göre yüz tanıma sonuçları.

Çizelge 4.2 Eigenface ile denek sonuçları tablosu.

Denek	Eigenface ile Sonuçlar					Ortalama
	Birinci	İkinci	Üçüncü	Dördüncü	Beşinci	
Bahadır	48	46	44	47	47	46,4
Atakan	67	65	63	63	66	64,8
Mustafa	63	46	37	32	31	41,8
Behzat	57	54	54	53	54	54,4
Fatih	47	47	63	54	54	53
Yusuf	36	37	36	41	42	38,4
Batuhan	42	42	43	39	42	41,6
Caner	37	44	41	41	38	40,2
Galip	47	52	55	47	43	48,8
Çağlar	46	45	43	46	43	44,6

Bu sonuçları grafiğe yansıtırsak;



Şekil 4.2 Eigenface algoritması denek sonuçları grafiği.

#### 4.2.3 FisherFace ile Test Bulguları

Fisherface algoritması ile de aynen Eigenface algoritması ile yaptığımız çalışma şeklinde Şekil 4.3'deki deneklerin bir saniye ara ile alınmış yüzleri bulunmaktadır. Burada da alınan yüzlerin benzerlik puantajı altlarında gözükmetedir. Sonuçlar, karşılaştırmalı şekilde aşağıdaki tabloda detaylı incelenmiştir. Fisherface metodunda da aynı kişi için o anki yüz ifadesine göre alınan beş farklı resimde ortaya çıkan sonuçların, birbirini tutmadığı görülmektedir. Bu duruma ana etken, ışığın anlık açısı ve kameranın netliğinin o anlık durumudur.

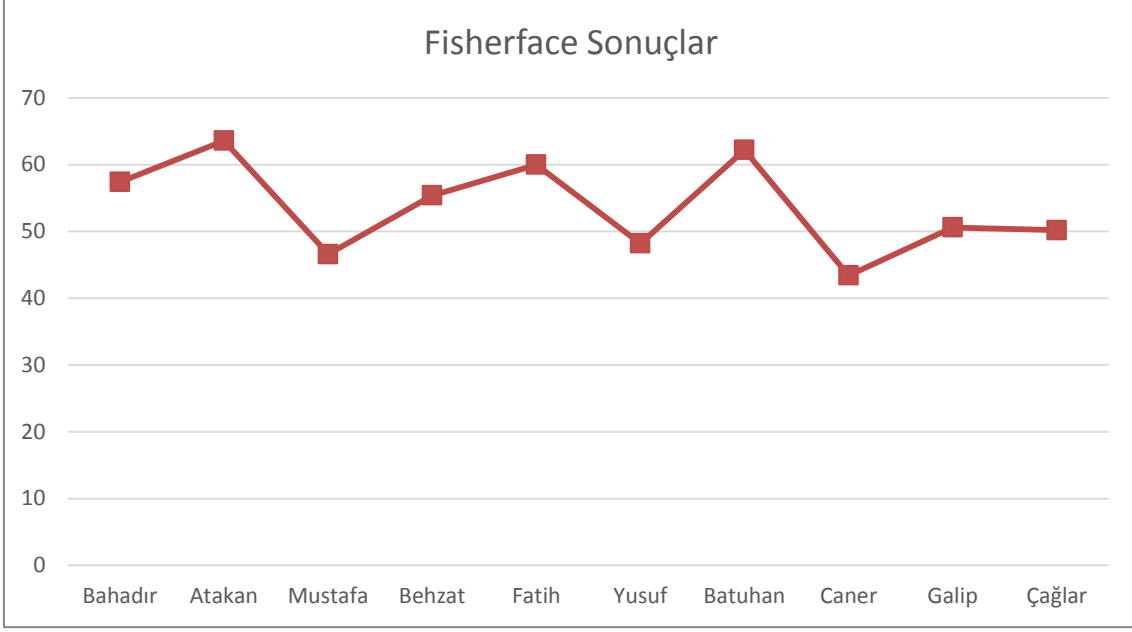


Şekil 4.3 Fisher algoritmasına göre yüz tanıma sonuçları.

Çizelge 4.3 Fisherface ile denek sonuçları tablosu.

Denek	Fisherface ile Sonuçlar					Ortalama
	Birinci	İkinci	Üçüncü	Dördüncü	Beşinci	
Bahadır	48	57	63	55	64	57,4
Atakan	51	70	69	66	62	63,6
Mustafa	38	45	50	48	52	46,6
Behzat	61	58	48	56	54	55,4
Fatih	64	55	67	54	60	60
Yusuf	33	63	48	48	49	48,2
Batuhan	58	85	58	58	52	62,2
Caner	45	38	47	44	43	43,4
Galip	58	55	52	43	45	50,6
Çağlar	47	40	45	53	66	50,2

Bu sonuçları grafiğe yansıtırsak;



Şekil 4.4 Fisherface algoritması denek sonuçları grafięi.

#### 4.2.4 LBPH ile Test Bulguları

LBHP algoritmasında da aynen Eigenface algoritması ile yaptığımız alıřma řeklinde, Şekil 4.5'deki deneklerin bir saniye ara ile alınmış yüzleri bulunmaktadır. Burada da alınan yüzlerin benzerlik puantajı resimlerin altlarında gözükmemektedir. Sonuçlar karşılaştırmalı řekilde ařaęıdaki tabloda detaylı incelenmiştir. Fisherface metodunda da yine aynı kiři için o anki yüz ifadesine göre alınan beř farklı resimde ıkan sonuçların birbirini tutmadığı görülmektedir. Bu duruma ana etken ışığın anlık açısı ve kameranın netlięinin o anlık durumudur.

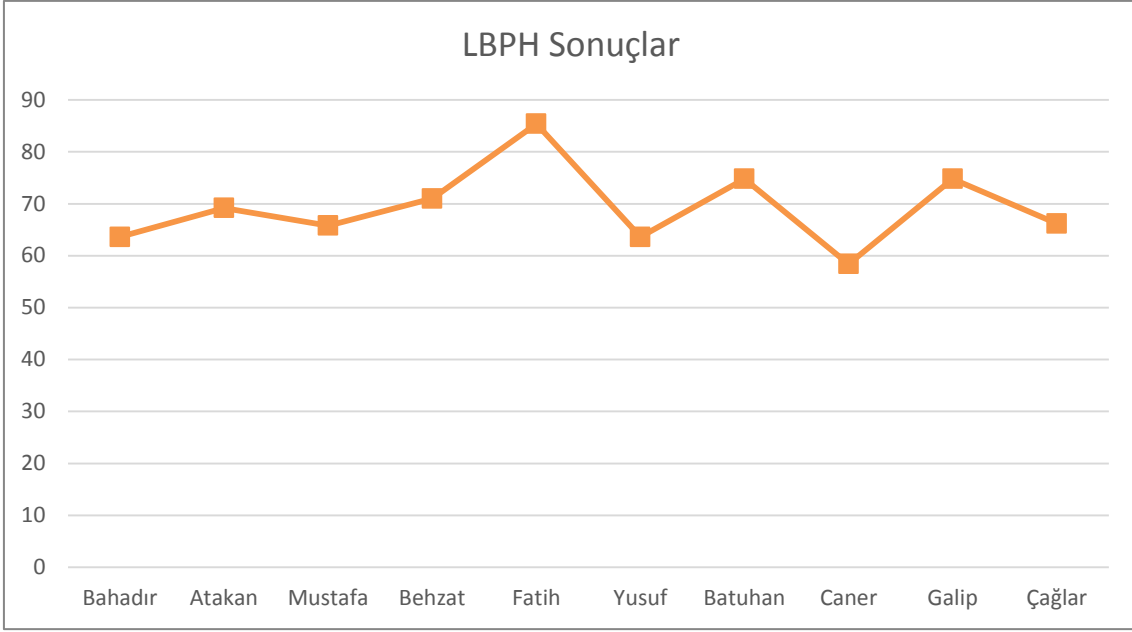


Şekil 4.5 LBPH algoritmasına göre yüz tanıma sonuçları.

Çizelge 4.4 LBPH ile denek sonuçları tablosu.

Denek	LBPH ile Sonuçlar					Ortalama
	Birinci	İkinci	Üçüncü	Dördüncü	Beşinci	
Bahadır	66	62	65	61	64	63,6
Atakan	64	78	56	74	74	69,2
Mustafa	44	72	72	74	67	65,8
Behzat	60	71	97	75	52	71
Fatih	88	82	84	87	86	85,4
Yusuf	81	60	59	58	60	63,6
Batuhan	75	88	67	75	69	74,8
Caner	58	50	56	61	67	58,4
Galip	78	68	72	63	93	74,8
Çağlar	57	64	78	67	65	66,2

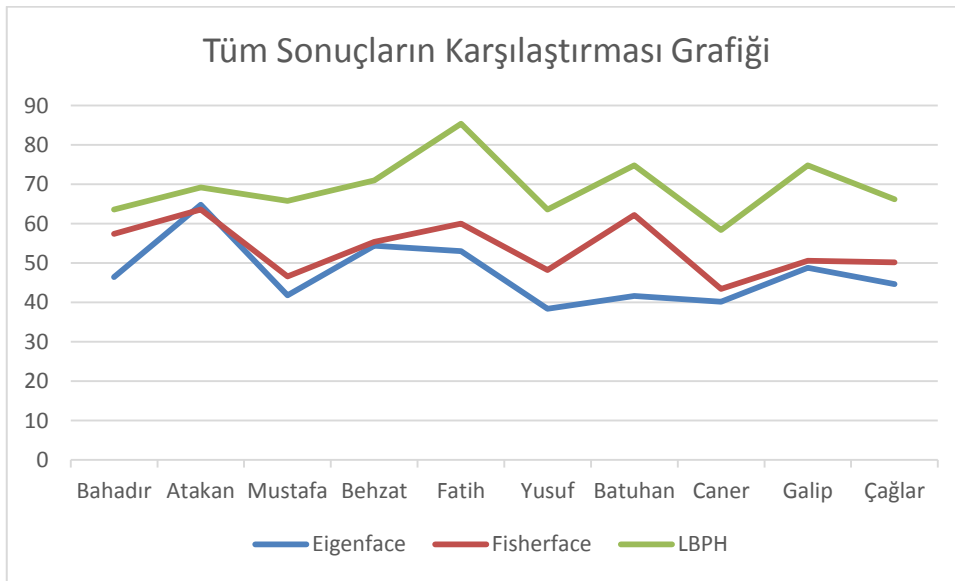
Bu sonuçları grafiğe yansıtılırsa;



Şekil 4.6 LBPH algoritması denek sonuçları grafiği.

#### 4.2.5 Elde Edilen Bulguların Karşılaştırılması

Denekler üzerinde elde ettiğimiz sonuçların ortalamaları, farklı algoritmaların performanslarını ortaya koyacaktır. Yapılan karşılaştırma, öncelikle her bir algoritmayı bir diğeri ile karşılaştırarak sonucun grafiksel olarak gözle görülür bir şekilde belirtilmesi amaçlanmaktadır.



Şekil 4.7 Tüm algoritmaların karşılaştırma grafiği.

Eigenface ve LBPH algoritmalarını karşılaştırdığımızda, önceki Fisherface algoritmasına göre, gözle görünür ölçüde farklılık oluşturduğu ayırt edilebilmektedir. Bu sonuç LBPH algoritmasının benzetme oranının, daha da başarılı olduğunu göstermektedir.

## 5. TARTIŞMA ve SONUÇ

Bu çalışmada OpenCV kütüphaneleri içerisinde bulunan yüz bulma ve tanıma algoritmaları olan Eigenface, Fisherface ve LBPH algoritmaları ile gömülü sistemlerde verimlilik karşılaştırılması yapılmıştır. Bu amaçla OpenCV içerisinde standart gelen birçok geliştirici ve saha çalışması için örnek yüz tanıma sınıflandırıcısı olan Haar Cascaded Sınıflandırıcısı ile çalışılmıştır.

Yapılan çalışmada sınıflandırıcıların ne kadar önemli olduğunun yanı sıra ortamdaki ışık miktarına bağlı sonuçların değişim gösterdiği gözlenmiştir. Sonuç olarak literatüre en son giren bir algoritma olan LBPH algoritmasının diğer algoritmalara göre daha başarılı olduğu tespit edilmiştir. Lakin henüz daha %100 oranında kesin sonuçların alınmadığı ve sadece tek kameranın sağladığı 2D görüntüler ile güvenilebilirliğinin tartışılabilir seviyede olduğu görünmektedir.

Teknolojinin hızla gelişmesi biyometri alanındaki yüz tanıma teknolojilerinde gelişmesi sağlayarak, kapı geçiş sistemlerinden bankacılık uygulamalarına kadar gömülü sistem içerisine giren ek kamera ve sensörler ile yüz tanıma son birkaç yıldır yaygın olarak kullanılmaktadır. 3D derinlik algılamaya yardımcı sensörlerle birlikte yüz tanıma sistemlerinin kullanılması güvenlik alanında güvenilir çözümler sağlamaktadır. Bu süreçte birçok mobil teknoloji üreten firmalar kameranın yanına ikinci kamera veya derinlik ölçümü yapabilen sensörler yerleştirerek 3D algılama ve tanıma yazılım ve algoritmaları ile 2D'ye göre oldukça yüksek verim elde etmektedirler.

Genel anlamda nesnelerin dünyası (IoT) adını alan gömülü sistemlerin birbirleri ile iletişim kurmasını hedefleyen Endüstri 4.0 ve yenilikleri ile güvenlik oldukça önem kazanmıştır. Önemli bir pozisyonda çalışan bir kişinin akıllı telefonunun ele geçirilmesi birbirine bağlı teknolojiler olması nedeniyle birçok bilginin ele geçirilmesine, kötü amaçlar için sistemlerin amaçları dışında kullandırılmasından mali zarara kadar pek çok problemler meydana gelmektedir. Yüz tanıma özelliğinin gömülü sistemlere uygulanmasının maliyet açısından oldukça düşük olması nedeniyle bu tür cihazlar içerisinde üst seviye güvenlik için kullanılmasını arttırmıştır. Bu sebeple iş istasyonu



olarak satılan yeni birçok taşınabilir bilgisayarlarda yüz tanıma yaygın olarak karşımıza çıkmaya başlamıştır. Teknolojik gelişmeler yüz bulma ve tanıma işlemlerinin verimliliğinin gerçek zamanlı olarak düşük işlemcilerde bile iyi sonuçlar verdiğini gözlemlemiş olmamız kullanılabilirlik açısından oldukça tatmin edici gözükmektedir.

Yapay zekaya sahip bilgisayarlar, görüntü işleme tekniklerini ve yüz tanıma ile birlikte sahiplerini ortam içerisinde tanımladıkları anda iletişim kurmaları karşımıza çıkacağı aşikardır. Teknolojinin akıl almaz bir hızla ilerlediği günümüzde artık insanlık yapay zekanın gelişmesi ile ortaya çıkacak yeni dünya düzenini merakla ama biraz da ürkerek beklemektedir.

## 6. KAYNAKLAR

- Barr, M. and Massa, A.J. (2006). Programming embedded systems: with C and GNU development tools, 2nd Edition, Kindle Edition, O'Reilly Media, USA.
- Belhumeur, P.N., Hespanha, J. and Kriegman, D. (1997). Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **19**: 711–720.
- Bledsoe, W. (1966). Man-Machine Facial Recognition: Report on a Large-Scale Experiment, Technical Report PRI 22. Panoramic Research, Inc., Palo Alto, California, USA.
- Bledsoe, W. (1968). Semiautomatic Facial Recognition, Technical Report SRI Project 6693. Stanford Research Institute, Menlo Park, California, USA.
- Buolamwini, J. and Gebru, T. (2018). Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification. *Proceedings of Machine Learning Research*, **81**: 1-15.
- C.Alippi. (2014). Intelligence for Embedded Systems. Springer, USA:
- Chaudhary, U., Bhardwaj, S. and Sabharwal, H. (2014). Fingerprint Recognition using orientation features. *International Journal of Advanced Research in Computer Science and Software Engineering*, **4**: 1403-1413.
- Chellappa, R., Wilson, C. and Sirohey, S. (1995). Human and machine recognition of faces: a survey. *Proceedings of the IEEE*, **83**: 705 - 741.
- Delac, K. and Grgic, M. (2004). A survey of biometric recognition methods. 46th International Symposium on Electronics in Marine, Croatia, 18 June 2004, 184-193.
- Ergen, B. ve Çalışkan, A. (2011). Biyometrik Sistemler ve El Tabanlı Biyometrik Tanıma Karakteristikleri. 6th International Advanced Technologies Symposium. Elazığ: Fırat Üniversitesi, 16-18 Mayıs 2011, 455-460.
- He, D. and Wang, L. (1990). Texture Unit, Texture Spectrum, And Texture Analysis, *Geoscience and Remote Sensing. IEEE Transactions on Geoscience and Remote Sensing*, **28**: 509 - 512.
- Heath, S. (2003). Embedded systems design. 2. Editions, Oxford: Newnes, UK.
- Jain, A. K. and Ross, A. (2008). Introduction to Biometrics. In. Jain, A., Flynn,

- P., Ross, A.A. (Eds.), Handbook of Biometrics, Springer, USA, 1-22.
- Kaehler, A. and Bradski, G. (2008). Learning OpenCV: Computer Vision with the OpenCV Library. O'Reilly Media, Inc., Sebastopol, California, USA.
- Martinez, A.M. and Kak, A.C. (2001). PCA versus LDA. *IEEE Transactions On Pattern Analysis And Machine Intelligence*, **23**: 228-233.
- Mounica, C. and Venugopal, P. (2016). Face Detection and Recognition Using LBPH. *International Journal of Engineering Research and Science & Technology*, **5**: 10-16.
- Ojala, T., Pietikäinen, M. and Harwood, D. (1996). A Comparative Study of Texture Measures with Classification Based on Feature Distributions. *Pattern Recognition*, **29**: 51-59.
- Ötkün, A. and Karlık, B. (2013). YSA ve Pencere Ortalamaları Kullanılarak Yüz Tanıma Sistemi. Otomatik Kontrol Ulusal Toplantısı, Malatya, 26-28 Eylül 2013, 996-1000.
- Prabhakar, S., Pankanti, S. and Jain, A.K. (2003). Biometric recognition: Security and privacy concerns. *IEEE Security & Privacy*, **99**: 33-42.
- Rani, K. and Kambo, S. (2017). Face Recognition Technique Using ICA and LBPH. *International Research Journal of Engineering and Technology*, **4**: 104-108.
- Raudys, S. and Jain, A. (1991). Small sample size effects in statistical pattern recognition: Recommendations for practitioners. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **13**: 252-264.
- Stergiopoulou, E. and Papamarkos, N. (2009). Hand gesture recognition using a neural network shape fitting technique. *Engineering Applications of Artificial Intelligence*, **22**: 1141-1158.
- Şamlı, R. and Yüksel, M. (2009). Biyometrik Güvenlik Sistemleri. Akademik Bilişim'09 - XI. Akademik Bilişim Konferansı Bildirileri , Şanlıurfa: Harran Üniversitesi, 11-13 Şubat 2009, 683-689.
- Turk, M.A. and Pentland, A.P. (1991). Face recognition using eigenfaces. Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Hawaii, 3-6 June 1991, 586-591.
- Turk, M. and Pentland, A. (1991). Eigenfaces for Face Detection/Recognition. *Journal of Cognitive Neuroscience*, **3**: 71-86.

- Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, **1**: 511-518.
- Wang, X., Han, T.X. and Yan, S. (2009). An HOG-LBP Human Detector with Partial Occlusion Handling. ICCV 2009 : International Conference on Computer Vision, Kyoto, Japan, 29 Sep. - 2 Oct 2009, 32 - 39.
- Yozgat, M. (2004). Bilgisayarda Parmak İzi Tanıma. Yüksek Lisans Tezi, Gazi Üniversitesi Bilişim Enstitüsü, Ankara.

## 6.1 İnternet Kaynakları

1. <https://tr.wikipedia.org/wiki/Biyometri>, 28.01.2018
2. <http://onin.com/fp/fphistory.html>, 10.04.2018
3. [https://commons.wikimedia.org/wiki/File:ADSL\\_modem\\_router\\_internals\\_labeled.jpg](https://commons.wikimedia.org/wiki/File:ADSL_modem_router_internals_labeled.jpg), 10.04.2018
4. [https://www.cl.cam.ac.uk/~jgd1000/iris\\_recognition.html](https://www.cl.cam.ac.uk/~jgd1000/iris_recognition.html), 01.05.2018
5. <https://searchcrm.techtargget.com/definition/voice-recognition>, 02.05.2018
6. <https://www.blinkidentity.com/blink-blog/2018/3/12/identity-primer-retina-vs-iris>, 02.05.2018
7. <http://biometrics.mainguet.org/types/face.htm>, 15.05.2018
8. <https://tr.wikipedia.org/wiki/DNA>, 18.05.2018
9. <https://hackernoon.com/building-a-facial-recognition-pipeline-with-deep-learning-in-tensorflow-66e7645015b8>, 22.05.2018
10. [https://tr.wikipedia.org/wiki/Raspberry\\_Pi](https://tr.wikipedia.org/wiki/Raspberry_Pi), 02.06.2018
11. [https://docs.opencv.org/3.4.3/d7/d8b/tutorial\\_py\\_face\\_detection.html](https://docs.opencv.org/3.4.3/d7/d8b/tutorial_py_face_detection.html), 04.06.2018
12. [https://en.wikipedia.org/wiki/Viola%E2%80%93Jones\\_object\\_detection\\_framework#Feature\\_types\\_and\\_evaluation](https://en.wikipedia.org/wiki/Viola%E2%80%93Jones_object_detection_framework#Feature_types_and_evaluation), 05.06.2018
13. [http://people.ece.cornell.edu/land/courses/ece4760/FinalProjects/s2011/bjh78\\_caj65/bjh78\\_caj65/](http://people.ece.cornell.edu/land/courses/ece4760/FinalProjects/s2011/bjh78_caj65/bjh78_caj65/), 26.05.2018
14. [https://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec\\_tutorial.html](https://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec_tutorial.html), 25.05.2018
15. <https://www.learnopencv.com/principal-component-analysis/>, 26.05.2018
16. <https://www.slideshare.net/marina761/face-recognitionpptx-state-of-the-art-in-face-recognition>, 30.05.2018
17. [https://docs.opencv.org/2.4/modules/objdetect/doc/cascade\\_classification.html#cascadeclassifier-detectmultiscale](https://docs.opencv.org/2.4/modules/objdetect/doc/cascade_classification.html#cascadeclassifier-detectmultiscale), 02.06.2018
18. <http://onin.com/fp/fphistory.html>, 05.06.2018
19. <https://cs.nyu.edu/~roweis/data.html>, 02.04.2018
20. [https://tr.wikipedia.org/wiki/G%C3%B6m%C3%BCl%C3%BC\\_sistem](https://tr.wikipedia.org/wiki/G%C3%B6m%C3%BCl%C3%BC_sistem), 08.04.2018

21. [https://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec\\_api.html#FaceRecognizer%20:%20public%20Algorithm](https://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec_api.html#FaceRecognizer%20:%20public%20Algorithm), 03.05.2018
22. <https://en.wikipedia.org/wiki/Eigenface>, 02.06.2018
23. [https://en.wikipedia.org/wiki/Local\\_binary\\_patterns](https://en.wikipedia.org/wiki/Local_binary_patterns), 08.06.2018
24. [https://tr.wikipedia.org/wiki/Raspberry\\_Pi](https://tr.wikipedia.org/wiki/Raspberry_Pi), 05.06.2018
25. <https://tr.wikipedia.org/wiki/Raspbian>, 01.06.2018
26. <https://www.sciencedaily.com/releases/1997/11/971112070100.htm>, 25.05.2018
27. <https://www.businessinsider.sg/snapchat-buys-looksery-2015-9/>, 22.05.2018
28. <https://www.theverge.com/2014/7/7/5878069/why-facebook-is-beating-the-fbi-at-facial-recognition>, 07.05.2018
29. <https://www.theverge.com/2017/9/12/16298156/apple-iphone-x-face-id-security-privacy-police-unlock>, 12.05.2018
30. <https://bigbrotherwatch.org.uk/wp-content/uploads/2018/05/Face-Off-final-digital-1.pdf>, 25.05.2018
31. <https://phys.org/news/2016-12-facial-recognition-technology-real-world.html>, 23.05.2018
32. [https://en.wikipedia.org/wiki/Facial\\_recognition\\_system](https://en.wikipedia.org/wiki/Facial_recognition_system), 21.05.2018
33. [https://web.archive.org/web/20070205015138/http://www.oregon.gov/ODOT/DMV/news/cards\\_facialrec.shtml](https://web.archive.org/web/20070205015138/http://www.oregon.gov/ODOT/DMV/news/cards_facialrec.shtml), 18.05.2018
34. <https://techcrunch.com/2017/05/17/google-photos-upgraded-with-new-sharing-features-photo-books-and-google-lens/>, 23.05.2018
35. <https://spectrum.ieee.org/computing/software/the-2017-top-programming-languages>, 18.05.2018
36. <http://www.scholarpedia.org/article/Fisherfaces>, 22.05.2018
37. <https://electronics.howstuffworks.com/gadgets/high-tech-gadgets/facial-recognition.htm>, 02.06.2018

## ÖZGEÇMİŞ

Adı Soyadı : Mehmet Fatih Özdemir  
Doğum Yeri ve Tarihi : Ankara / 1979  
Yabancı Dili : İngilizce  
İletişim (Telefon/e-posta) : 0(533)5153133 / mfozdemir@gmail.com

### Eğitim Durumu (Kurum ve Yıl)

Lise : Başkent Lisesi, Fen Bilimleri, (1994-1997)  
Önlisans : Osmangazi Üniversitesi, Bilgisayar Programcılığı,  
(1997-2000)  
Lisans : Anadolu Üniversitesi, İktisat Bölümü, (2003-2009)  
Yüksek Lisans : Afyon Kocatepe Üniversitesi Fen Bilimleri  
Enstitüsü İnternet ve Bilişim Teknolojileri  
Anabilim Dalı (2015-2018)

### Çalıştığı Kurum/Kurumlar ve Yıl :

Ofisim.com A.Ş	2017-2018	Çözüm Yöneticisi
Serbest Çalışan	2013-2017	Bilişim Danışmanı
Karaca Mühendislik - irp.com.tr	2012-2013	Reklam Mühendisi Proje ortağı ve Proje Yöneticisi
Todaie – e-Devlet Merkezi	2011-2012	Proje Bilişim Uzmanı
Başkent Üniversitesi	2010-2012	Kısmi Zamanlı Usta Öğretici
Mybilet Bilişim Teknolojileri	2007-2010	Proje Yöneticisi – Kıdemli Yazılımcı
Avrasya Stratejik Araştırmalar Merkezi (ASAM)	1999-2006	Bilgi İşlem Müdürü