**T.C.**

**Fırat** **Üniversitesi**

**Mühendislik Fakültesi**

**Elektrik-Elektronik Mühendisliği Bölümü**

**BİTİRME PROJESİ**

**GÖRÜNTÜ İŞLEME İLE İRİS TANIMA**

**Bitirme Projesi Yazarı/Yazarları**

**16220014-Burak Elhaman**

**15220055-Yunus Yılmaz**

**Danışman**

**Dr. Öğr. Üyesi Duygu KAYA**

**HAZİRAN 2021**

**T.C.**

**Fırat** **Üniversitesi**

**Mühendislik Fakültesi**

**Elektrik-Elektronik Mühendisliği Bölümü**

**BİTİRME PROJESİ**

|  |  |
| --- | --- |
| Başlığı: | Görüntü İşleme İle İris Tanıma |
| Yazar/Yazarlar: | Burak ELHAMAN / Yunus YILMAZ |
| Sunum Tarihi: | .2021 |

**BİTİRME PROJESİ ONAYI**

Fırat Üniversitesi Mühendislik Fakültesi Elektrik-Elektronik Mühendisliği Bölümü Bitirme Projesi yazım kurallarına göre hazırlanan bu Bitirme Projesi aşağıda imzası bulunan danışman/jüri tarafından değerlendirilmiş ve akademik dinleyicilere açık yapılan sunum sonucunda kabul edilmiştir.

*İmza*

Danışman:

Jüri :

Jüri :

Beyan

Fırat Üniversitesi Mühendislik Fakültesi Elektrik-Elektronik Mühendisliği Bölümü Bitirme Projesi yazım kurallarına uygun olarak hazırladığımız “Görüntü İşleme İle İris Tanıma” Başlıklı Bitirme Projesinin içindeki bütün bilgilerin doğru olduğunu, bilgilerin üretilmesi ve sunulmasında bilimsel etik kurallarına uygun davrandığımı, kullandığım bütün kaynakları atıf yaparak belirttiğimi, maddi ve manevi desteği olan tüm kurum/kuruluş ve kişileri belirttiğimi, burada sunduğum veri ve bilgileri unvan almak amacıyla daha önce hiçbir şekilde kullanmadığımı beyan ederim.

Tarih:27/06/2021

Yazar/Yazarlar:

Burak Elhaman

Yunus Yılmaz

ÖNSÖZ

İris tanıma sistemini diğer biyometrik sistemlerden ayıran en önemli özellik kendi kendine oluşturulmuş benzersiz modeli harici olarak görülebilir, yetişkin yaşamı boyunca sabit kalır ve kopyalanamaz. Bu anahtar faktörler iris'i tanımlamak için biyometrik olarak uygun hale getirir. Bu özellik sayesinde geçiş kontrol sistemleri arasındaki en güvenli çözümlerden biri olan iris tanıma sistemi, hızlı tarama, düşük uygulama maliyeti ve hijyenik olmasıyla öne çıkar.

Bitirme projemizin planlanmasında, araştırılmasında, yürütülmesinde ve oluşumunda ilgi ve desteğini esirgemeyen, engin bilgi ve tecrübelerinden yararlandığımız sayın hocamız Dr. Öğr. Üyesi Duygu KAYA’ya sonsuz teşekkürlerimi sunarız.

Teşekkürlerin az kalacağı diğer üniversite hocalarımın da bana 4 yıllık üniversite hayatım boyunca kazandırdıkları her şey için ve beni gelecekte söz sahibi yapacak bilgilerle donattıkları için hepsine teker teker teşekkürlerimi sunuyorum. Bizi bu günlere sevgi ve saygı kelimelerinin anlamlarını bilecek şekilde yetiştirerek getiren ve bizden hiçbir zaman desteğini esirgemeyen bu hayattaki en büyük şansımız olan ailemize sonsuz teşekkürler.

Yazar/Yazarlar:

Burak Elhaman

Yunus Yılmaz

Elazığ, 2021

İçindekiler

Sayfa

[ÖNSÖZ iv](#_Toc75630770)

[İçindekiler v](#_Toc75630771)

[Özet vii](#_Toc75630772)

[Abstract viii](#_Toc75630773)

[ŞEKİLLER LİSTESİ ix](#_Toc75630774)

[Tablolar Listesi x](#_Toc75630775)

[Ekler Listesi xi](#_Toc75630776)

[1. GİRİŞ 1](#_Toc75630777)

[2. Biyometrik Tanıma Sistemleri 2](#_Toc75630778)

[**2.1.** **Biyometrik Sistemler** 2](#_Toc75630779)

[**2.2.** **Biyometrik Sınıflandırma** 3](#_Toc75630780)

[**2.2.1. Parmak İzi Tanıma** 3](#_Toc75630781)

[**2.2.3. Ses Tanıma** 4](#_Toc75630782)

[**2.2.4. Retina** 4](#_Toc75630783)

[**2.2.5. El Geometrisi** 4](#_Toc75630784)

[**2.2.6. İmza Dinamikleri** 5](#_Toc75630785)

[**2.3.** **İris Tanıma** 5](#_Toc75630786)

[**2.3.1. İris Tanıma'nın Özellikleri** 5](#_Toc75630787)

[**2.4. Özet** 8](#_Toc75630788)

[3. Görüntü İşleme 9](#_Toc75630789)

[**3.1. Görüntü İşleme** 9](#_Toc75630790)

[**3.1.2. Kameralar ve Gözler** 9](#_Toc75630791)

[**3.2. Filtre Uygulamaları** 10](#_Toc75630792)

[**3.2.1. Histogram Eşitleme** 12](#_Toc75630793)

[**3.3. Özet** 13](#_Toc75630794)

[4. Projenin İlerleme Aşamaları 14](#_Toc75630795)

[**4.1. Uygun Çalışma Ortamının Belirlenmesi** 14](#_Toc75630796)

[**4.1.1. Raspberry Pi İşletim Sistemi Yüklenmesi** 15](#_Toc75630797)

[**4.1.2. Raspberry Pi Güncelleme ve Yükseltme** 16](#_Toc75630798)

[**4.1.3. Raspberry Pi OpenCV Kütüphanesi Yükleme** 17](#_Toc75630799)

[**4.2. PyCharm Python IDE Kullanılması ve Gerekli Yazılımların Yüklenmesi** 17](#_Toc75630800)

[5. İris Tanıma Aşamaları 20](#_Toc75630801)

[**5.1. Görüntü Alma** 21](#_Toc75630802)

[**5.2. İris Algılama** 22](#_Toc75630803)

[**5.3. İris Normalleştirme** 24](#_Toc75630804)

[**5.4. Özellik Çıkarma** 25](#_Toc75630805)

[**5.5. Veri Tabanı Oluşturma** 25](#_Toc75630806)

[**5.6. Karşılaştırma** 26](#_Toc75630807)

[6. Sonuçlar 27](#_Toc75630808)

[Kaynaklar 28](#_Toc75630809)

[Ekler 29](#_Toc75630810)

[Özgeçmiş 33](#_Toc75630811)

[Özgeçmiş 33](#_Toc75630812)

Özet

Dummy

Görüntü İşleme ile İris Tanıma

Bitirme Projesi

Fırat Üniversitesi

Mühendislik Fakültesi

Elektrik-Elektronik Mühendisliği Bölümü

Haziran, 2021, Sayfa: xi + 33

Günlük hayatta yapılan her türlü işlemlerde zaman, hız ve güvenilirlik önem taşır. Büyük bir şirketin girişinde, çalışanların her giriş ve çıkışlarında kart okutmaları ya da benzeri yöntemler kullanmaları kaçınılmaz olmaktadır. Fakat kullanılan bu yöntemlerin güvenlik zincirinin kırılmaması gerekmektedir. Bunun için de her geçen gün daha yeni ve performansı yüksek metodlar ortaya atılmaktadır. İris, gözün iyi korunmuş bir parçasıdır. Kendi kendine oluşturulmuş benzersiz modeli harici olarak görülebilir ve yetişkin yaşamı boyunca sabit kalır. Bu anahtar faktörler İris'i tanımlamak için biyometrik olarak uygun hale getirir.

Bu proje, İris tanıma sisteminin oluşturulmasını amaçlamaktadır. Bu sistem kimlik doğrulayabilen tanıma ve tanımlama sistemidir. Hedefin doğruluğunu kanıtlamak ve veri hazırlanması için göz görüntüsü üzerine görüntü işleme teknikleri uygulanır. Görüntü işleme teknikleri, göz görüntüsünün -standart ışıklandırma ve odaklanmada- alınmasından veri elde etmek için gerekli olan çok net bir iris görüntüsü almaya yönelik adımları göstermektedir. Bu görüntü, tüm gözü (iris, gözbebeği ve kirpikler) içermektedir. Bu sebeple, irisin yerinin belirlenmesi çok önemlidir. İris yerinin belirlenmesi, göz bebeğinin siyahlık yoğunluğunun izlenmesiyle yapılmaktadır.

**Anahtar Kelimeler:** Görüntü İşleme, İris Tanıma, Filtreleme, Biyometrik Veri

Abstract

dummy

Iris Recognition With Image Processing

FIRAT UNIVERSITY

faculty of engineering

Department of Electrical-Electronics Engineering

June,2021, Pages: xi + 33

Time, speed and reliability are important in all kinds of transactions in daily life. At the entrance of a company, it is inevitable for employees to read cards or use similar methods at each entrance and exit. However, the security chain of these methods should not be broken. For this, newer and higher performance methods are being introduced every day. The iris is a well-preserved part of the eye. Its unique self-built pattern is externally visible and remains constant throughout its adult life. These key factors make the Iris biometrically suitable for identifying.

This project aimed to create the iris recognition system. This system is an authentication and identification system that can authenticate. Image processing techniques are applied on the eye image to prove the accuracy of the target and to prepare data. Image processing techniques show the steps for obtaining a very clear iris image, which is necessary to extract data from the eye image -in standard lighting and focusing-. This image includes the entire eye (iris, pupil, and eyelashes). Therefore, it is very important to determine the location of the iris. The iris location is determined by monitoring the black density of the pupil.

**Keywords:** Image processing, Iris Recognition, Filtering, Biometric Data

ŞEKİLLER LİSTESİ

Sayfa

**Şekil 2.1.** Sıklıkla Kullanılan Biyometrik Karakteristik Örnekleri…….…………………………….….. 3

**Şekil 2.2.** Parmak İzi Örneği…………………………………...………………………………….…….. 3

**Şekil 2.3.** Retina Örneği…………………………………………..…………………….……………….. 4

**Şekil 2.4.** El Resmi Örneği………………………………………..………………………………….….. 5

**Şekil 2.5.** İmza Resmi Örneği……………………………………..…………………………….……….. 5

**Şekil 2.6.** İris Örneği………….…………………………………..………………………….………….. 6

**Şekil 2.7.** Gözün Bölümleri…...…………………………………..………………………….………….. 6

**Şekil 3.1.** Sobel Filtre Maskeleri...………………………………..……………………….…………….. 10

**Şekil 3.2.** Sobel Filtre Kenar Algılama Örneği……………….……………………………….………… 10

**Şekil 3.3.** Gauss Bulanıklaştırma Örneği……………………..……………………………….……..….. 11

**Şekil 3.4.** Histogram Eşitleme Filtresi…………………………..……...………….……………….…... 12

**Şekil 4.1.** Raspberry Pi Model 2B Örneği……………………...…..……………………………………. 14

**Şekil 4.2.** Raspberry Pi İmager Ekranı……………………………..……………………………………. 15

**Şekil 4.3.** Raspberry Pi İşletim Sistemleri…………………………..……………………………………. 15

**Şekil 4.4.** Raspberry Pi Terminal Ekranı.…………………………..……………………………………. 16

**Şekil 4.5.** Raspberry Pi OpenCV Sürümleri……………………………..………………………………. 17

**Şekil 4.6.** OpenCV Sürümleri……….……………………………..…………………….………………. 18

**Şekil 4.7.** Kütüphanelerin Kontrolü….……………………………..…………….………………………. 19

**Şekil 5.1.** İlerleyiş……..…………….……………………………..…………….………………………. 20

**Şekil 5.2.** Aşamalar.………………….……………………………..…………….………………………. 20

**Şekil 5.3.** Inca IC-3262……..……….……………………………..…………….………………………. 21

**Şekil 5.4.** Alınan Gözün Görüntüsü…..…..………………………..…………….………………………. 21

**Şekil 5.5.**  Görüntünün Gray Filtresi Uygulanmış Hali…….………..…………….………………………. 22

**Şekil 5.6.**  Medyan Filtresi Uygulanmış Hali.………………….…..…………….………………………. 22

**Şekil 5.7.** Görüntünün Küçültülmüş Hali…..………..……………..…………….………………………. 23

**Şekil 5.8.** Threshold Uygulanmış Hal………..……………………..…………….………………………. 23

**Şekil 5.9.**  Boyanmış Hal…………………………………………....…………….………………………. 24

**Şekil 5.10.** Normalleştirme Tekniği………….…………..…………..…………….………………………. 24

**Şekil 5.11.** İrisin Normalleştirilmiş Hali…...………………………..…………….………………………. 24

**Şekil 5.12.** İrisin Özellik Çıkartılmış Hali.....………………………..…………….………………………. 25

**Şekil 5.13.** Database İçindeki Değerler……..………………………..…………….………………………. 25

**Şekil 5.14.** Karşılaştırma Sonucu…………………...………………..…………….………………………. 26

Tablolar Listesi

Sayfa

**Tablo 2.1.** Temel Biyometrik Karakteristiklerin Karşılaştırılması…….………….……………………….... 7

Ekler Listesi

Sayfa

Ek- 1: Database Oluşturma………………………..…….………….………………………....... 29

Ek- 2: Ana Kod…………………….……………...…….………….………………………....... 30

# **1. GİRİŞ**

Son yüzyıldan beri, insanların tanımlanması için çeşitli biyometrik teknikler kullanıldı. Bu teknikler; İris tanıma, Yüz tanıma, Parmak izi tanıma, Ses tanıma vb. İris Tanıma, insanın iki irisi arasındaki eşleşmeyi doğrulamanın otomatik yöntemini ifade eder. İris, bireyleri tanımlamak ve kimliklerini doğrulamak için kullanılan birçok biyometrik formdan biridir.

Özellik çıkarma, iris tanıma sisteminde çok önemli bir adımdır. Sistemin tanıma oranı, iris görüntüsünden çıkarılan anlamlı verilere bağlıdır. Bu nedenle, görüntülerden önemli özellikler çıkarılmalıdır. Elde edilen görüntüler aynı kişiye ait olsalar bile, aynı irisin görüntüleri arasında asla tam bir eşleşme olamaz.

Genel amaç, bir gözün irisi için taranan rastgele modellerin matematiksel bir analizi ile bireyin kimliğinin yüksek güvenirlikte tanınmasıdır. Kişilerin güvenilir bir şekilde otomatik olarak tanınması uzun süredir çekici bir hedeftir. Hedefteki adımlar şu şekildedir; Görüntü alınması, alınan görüntüye filtrelerin uygulanması, filtre sonrası iris yerinin berlirlenmesi, belirlenen irisin şeklinin düzeltilmesi, özelliklerinin çıkartılması, özelliklerin veritabanına kaydedilmesi, gelen görüntü ile veritabanındaki görüntünün karşılaştırılması.

# **2. Biyometrik Tanıma Sistemleri**

Biyometri, kişileri bireysel ve farklı fizyolojik veya davranışsal özelliklere dayalı olarak tanımak ve tanımlamak için kullanılan bir metodolojidir. İyi bir biyometrinin iki temel özelliği vardır: Kararlılık ve ayırt edicilik. Kararlı bir biyometri zamanla değişmez. Saç uzunluğu iyi bir biyometrik tanımlayıcı değildir. Ayırt edici biyometrik bireye özgüdür.

Kişisel tanımlama için geleneksel yöntemler, bir kişinin neye sahip olduğuna (fiziksel bir anahtar, kimlik kartı vb.) veya bir kişinin ne bildiğine (gizli bir şifre vb.) dayanır. Bu yöntemlerin bazı sorunları vardır. Anahtarlar kaybolabilir, kimlik kartları taklit edilebilir ve şifreler unutulabilir.

Son yıllarda biyometrik kişisel kimlik, endüstriyel ve akademik bakış açısıyla ilgi çekici bir alan olarak gelişiyor. Dolayısıyla biyometrik yöntemler, kişisel tanımlama için insan özelliklerini kullanan güvenlik teknolojileridir. İris tanıma sistemleri, benzersiz tanımlayıcılar olarak iris dokularını kullanır.

## **2.1. Biyometrik Sistemler**

Biyometrik teknolojiler ölçtüklerine göre iki ana kategoriye ayrılabilir:

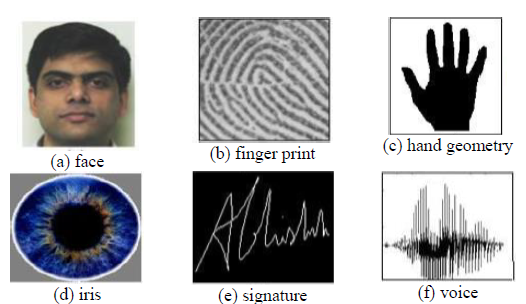
* Bir kişinin fizyolojik özelliklerine (parmak izi veya el geometrisi gibi) dayalı cihazlar.
* Bir kişinin davranış özelliklerine dayalı sistemler (imza dinamikleri gibi).

Günümüzde biyometrik tanıma, fizyolojik veya davranışsal özelliklere dayalı olarak yaşayan bir kişinin kimliğini doğrulamanın yaygın ve güvenilir bir yoludur.

Fizyolojik özellikler, parmak izi, iris deseni, yüz özelliği, el silüeti vb. gibi nispeten kararlı fiziksel özelliklerdir. Bu tür bir ölçüm temelde önemli bir zorlama olmaksızın değiştirilmez. Davranışsal bir özellik, bireyin psikolojik yapısının imza, konuşma kalıbı veya klavyede nasıl yazı yazıldığının bir yansımasıdır. Fiziksel bir özellikteki kişisel çeşitlilik derecesi, davranışsal bir özellikten daha küçüktür.

Örneğin, bir imza hem kontrol edilebilir eylemlerden hem de psikolojik faktörlerden etkilenir, sözlü ifade, konuşma da mevcut duygusal durumlardan etkilenirken parmak izi bu gibi durumlardan bağımsızdır.

Genel olarak fizyolojik biyometri sistemleri, Şekil 2.1'de gösterildiği gibi davranışsal biyometri sistemlerinden daha doğru ve daha yüksek maliyetlidir.



**Şekil 2.1.** Sıklıkla kullanılan biyometrik karekteristik örnekleri

## **2.2. Biyometrik Sınıflandırma**

### **2.2.1. Parmak İzi Tanıma**

Parmak izi tanıma teknolojisi, muhtemelen en yaygın kullanılan ve en iyi bilinen biyometrik teknolojisidir. Bir parmak izi örneği Şekil 2.2’de verilmiştir. Parmak izi tanıma, parmak uçlarındaki belirgin çıkıntıların özelliklerinin çıkarılmasına dayanır. İki tür parmak izi vardır: düz veya yuvarlağımsı. Düz baskılar, parmak yüzeyinin yalnızca orta alanının bir izlenimidir, yuvarlatılmış baskılar ise parmağın kenarlarındaki çıkıntıları ve ayrıca uç ile ilk eklem arasındaki orta kısmı yakalar. Parmak izi görüntüleri taranır, geliştirilir ve ardından şablonlara dönüştürülür. Bu şablonlar, optik, silikon veya ultrason tarayıcılar kullanılarak gelecekteki karşılaştırmalar için bir veri tabanına kaydedilir. Ultrason en doğru olanı gibi görünmektedir, ancak nadiren kullanılır; optik tarayıcılar en yaygın olarak kullanılır.



**Şekil 2.2.** Parmak izi örneği

### **2.2.3. Ses Tanıma**

İnsanları tanımaya yönelik olan ses tanıma teknolojisi, öğrenilmiş konuşma alışkanlıklarından kaynaklanan ses farklılıklarına dayanır. Bir kişi kaydolduğunda, kişi belirli yazılı bilgileri bir mikrofona veya telefona birden çok kez söylerken, sistem kişinin konuşmasının örneklerini yakalar. Bu bilgi "parola" olarak bilinir. (Önceden tanımlanmış bir cümle gerektirmeden insanların seslerini ayırt edebilen bazı biyometrik sistemler de mevcuttur). Parola daha sonra dijital bir biçime dönüştürülür ve konuşmacı için bir şablon oluşturmak üzere ayırt edici özellikler (örneğin, perde, kadans, ton) ayıklanır. Ses tanıma şablonları, tüm biyometrik şablonlar arasında en fazla veri alanı ihtiyacı duyulan şablondur.

Ancak ses tanıma, mevcut fiziksel ve duygusal durumdan olumsuz etkilenebilen davranış özelliklerine dayalı biyometrik bir tekniktir. Ses tanımanın doğruluğu, giriş sinyalindeki arka plan ve sistem gürültüsünden de etkilenebilir.

### **2.2.4. Retina**

Retina taraması, gözün retinasındaki kan damarı düzenine dayanır. Retina tarama teknolojisi, gözün bir kısmını da kullanan iris tarama teknolojisinden daha eskidir. İlk retina tarama sistemleri 1985 yılında EyeDentify tarafından piyasaya sürüldü. Bir lazer ışığı gözün korneasından yönlendirilmelidir. Ayrıca retina tarayıcının çalışması kolay değildir. Şekil. 2.3’te verilmiştir.



**Şekil 2.3.** Retina örneği

### **2.2.5. El Geometrisi**

El geometrisi, hemen hemen her insanın elinin farklı olması ve belli bir yaştan sonra kişinin elinin şeklinin değişmemesi esasına dayanmaktadır. El geometrisi sistemleri, parmakların uzunluğu ve genişliği gibi elin belirli ölçümlerinin tahminlerini üretir. El ölçmek için çeşitli yöntemler kullanılır. Optik yöntem günümüzde çok daha yaygındır. Optik el geometrisi tarayıcıları, elin görüntüsünü yakalar ve elin özelliklerini hesaplayan görüntü kenarı algılama algoritmasını kullanır. Bu durumda elin sadece İki Boyutlu (2D) özellikleri kullanılabilir. Şekil. 2.4’te verilmiştir.



**Şekil 2.4.** El resmi örneği

### **2.2.6. İmza Dinamikleri**

İmza dinamiği tanıma, imzanın kendisinin daha sonra doğrudan karşılaştırılmasından ziyade, imza oluşturma dinamiklerine dayanır. Dinamikler basınç, yön, hızlanma ve vuruşların uzunluğu, vuruş sayısı ve süreleri olarak ölçülür. İmza doğrulamanın öncüleri ilk olarak 1970'lerde güvenilir bir istatistiksel yöntem geliştirdi. Bu, kalemin kaç kez kaldırıldığı, toplam yazma süresi ve dönüm noktalarının zamanlaması gibi on veya daha fazla yazma özelliğinin çıkarılmasını içeriyordu. Bir imza örneği Şekil 2.5’de verilmiştir.

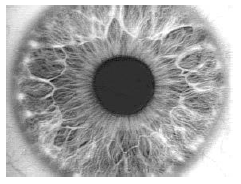


**Şekil 2.5.** İmza resmi örneği

## **2.3. İris Tanıma**

## **2.3.1. İris Tanıma'nın Özellikleri**

İris, bize gözümüzün rengini veren güzel pigmentli renkli dairesel bir kastır (irisin merkezi açıklığı göz bebeğidir). Şekil 2.6.’da gösterilen iris (dairesel kas), gözbebeğinin boyutunu kontrol eder, böylece aydınlatma koşullarına bağlı olarak daha fazla veya daha az ışığın göze girmesine izin verilir.

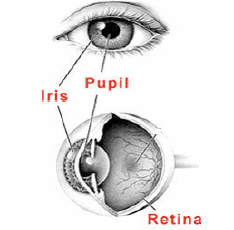


**Şekil 2.6**. İris örneği

Çizgiler, noktalar, halkalar, yarıklar, çukurlar, kriptalar, çiller, çizgiler, kasılma olukları, koronalar ve serpantin damar sistemi bir iriste bulunabilecek şeylerdir.

Bilim adamları, her insanın irisine özgü 250 özellik belirlemişler - parmak izleri için bu sayı yaklaşık 40 özelliktir – ve yaşlanma veya yaralanma yoluyla değiştirilebilen bir ses, yüz, parmak izi veya el modelinin aksine, bir kişinin irisi yaşamı boyunca sabit kalır.

Bu nedenle iris tanımlama, sesleri, yüzleri ve parmak izlerini tanımlayan diğer yüksek teknikli tanımlama sistemlerinden daha doğrudur.



**Şekil 2.7.** Gözün bölümleri

İris, gebeliğin üçüncü ayında oluşmaya başlar ve pigment birikimi doğum sonrası ilk yıllara kadar devam edebilir. Sekizinci ayda bebeğin iris pigmentini oluşturan yapılar büyük ölçüde tamamlanır. Ek olarak, iris çevreden iyi korunur.

İnsan vücudunun bireysel olarak en ayırt edici özelliğidir. İki iris, ikizler arasında bile birbirine benzemez. Hatta bir bireyin sol ve sağ irisleri de aynı değildir. İki irisin tam olarak aynı olma olasılığı 1072 'de 1 olarak tahmin edilmektedir; İris tanıma, istatistiksel olarak DNA testinden daha doğrudur.

Tüm bu nedenlerden dolayı İris tanıma, görüntüleme bir metreden daha kısa mesafelerde yapılabildiğinde ve bir yanlış eşleşmeye maruz kalmadan çok büyük veri tabanlarında arama yapılması gerektiğinde, kişilerin güvenilir bir şekilde görsel olarak tanınmasına olanak sağlanır.

Iris tanımada görüntü kalitesinin her zaman yüksek olması beklenilmez ancak farklı kişiler arasındaki model farklılığının çok büyük olması gibi büyük matematiksel avantajlar sayesinde bu problemler aza indirgenmektedir. İris rengi esas olarak ön tabakasındaki ve stromadaki melanin pigmentinin yoğunluğu ile belirlenir. Pigment yokluğundan kaynaklanan mavi iris rengine uzun dalga boylu ışık nüfuz ederken daha kısa dalga boyları stroma tarafından geri yansıtılır.

**Tablo 2.1:** Temel Biyometrik Karekteristiklerin Karşılaştırılması

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| KARAKTERİSTİK | Parmak İzleri | El Geometrisi | Retina | İris | Yüz | İmza | Ses |
| Kullanım kolaylığı | Yüksek | Yüksek | Düşük | Orta | Orta | Yüksek | Yüksek |
| Hata sıklığı | Kuruluk, kir, yaş | El yaralanması, yaş | Gözlük | Zayıf aydınlatma | Aydınlatma, yaş, gözlük, | İmzaları değiştirme | Gürültü, soğuk algınlığı, hava |
| Doğruluk | Yüksek | Yüksek | Çok Yüksek | Çok Yüksek | Yüksek | Yüksek | Yüksek |
| Kullanıcı kabulü | Orta | Orta | Orta | Orta | Orta | Orta | Yüksek |
| Güvenlik seviyesi  Uzun vadeli kararlı | Yüksek  Yüksek | Orta  Orta | Yüksek  Yüksek | Çok Yüksek  Yüksek | Orta  Orta | Orta  Orta | Yüksek  Orta |

## **2.4. Özet**

Her insanın parmak izi, göz, yüz, el, ses ve imza gibi hayati fizyolojik özelliği, kişisel tanımlamada önemli gelişmeler sağlamıştır. Bu biyometrideki özellik çıkarma, parolaları kullanarak kimliğin (kişilerin) geleneksel şekilde doğrulanmasıyla ilgili sorunların ve zayıflıkların birçoğunun azaltılmasında önemli bir başarı sağlamıştır. Bu tekniklerle elde edilen yüksek güvenlik derecesine rağmen, doğruluk tanıma sistemleri henüz %100'e ulaşmadı. Bu nedenle, özellik çıkarma ve görüntü tanıma için yeni tekniklerin tasarımı önemli bilgisayar bilimi haline geldi.

# **3. Görüntü İşleme**

## **3.1. Görüntü İşleme**

Görüntü işleme, gerekli özelliklere sahip yeni görüntü elde etmek için dijital olarak bir piksel toplamı olarak depolanan görüntüler üzerinde gerçekleştirilen çeşitli işlemleri ifade eder. Görüntü işlemede, görüntüyü sanatsal olarak dönüştürmek, bir görüntüdeki nesneleri bulmak ve tanımak ve görüntünün kalitesini artırmak veya düşürmek dahil olmak üzere birçok konu vardır.

Görüntülerin bilgileri uzamsal alanda kodlanmıştır. Başka bir deyişle, görüntülerdeki özellikler sinüzoidlerle değil kenarlarla temsil edilir. Bu, piksellerin aralığının, sayısının, örnekleme teoreminin resmi kısıtlamalarından ziyade, görülmesi gereken özelliklerin ne kadar küçük olduğuna göre belirlendiği anlamına gelir.

### **3.1.2. Kameralar ve Gözler**

Gözün yapısı ve işleyişi elektronik kameraya çok benzer ve bunları birlikte tartışmak doğaldır. Her ikisi de iki ana bileşene, bir lens düzeneğine ve bir görüntüleme sensörüne dayanmaktadır.

Lens düzeneği, bir nesneden yayılan ışığın bir kısmını yakalar ve onu görüntüleme sensörüne odaklar. Görüntüleme sensörü daha sonra ışık modelini elektronik veya sinirsel bir video sinyaline dönüştürür.

Her ikisi de bir ucunda bir lens ve diğer ucunda bir görüntü sensörü bulunan hafif sıkı muhafazalardır. Göz şeffaf bir sıvı ile doldurulurken kamera hava ile doldurulur. Her lens sisteminin iki ayarlanabilir parametresi vardır: odak ve iris çapı. Lens uygun şekilde odaklanmazsa, nesne üzerindeki her nokta görüntüleme sensöründe dairesel bir bölgeye yansıtılarak görüntünün bulanık olmasına neden olur. Kamerada odaklama, merceğin görüntüleme sensörüne doğru veya ondan uzağa fiziksel olarak hareket ettirilmesiyle sağlanır. Karşılaştırıldığında, göz iki merceği, göz küresinin ön tarafında kornea adı verilen bir çıkıntı ve gözün içinde ayarlanabilir bir mercek bulunur. Kornea ışığın kırılmasının çoğunu yapar, ancak şekil ve konum olarak sabittir. Odaklanmanın ayarlanması, siliyer kasların hareketiyle deforme olabilen esnek bir yapı olan iç mercek tarafından gerçekleştirilir. Bu kaslar kasıldıkça lens düzleşerek nesneyi keskin bir odağa getirir. Her iki sistemde de iris, merceğin ne kadarının ışığa maruz kaldığını ve dolayısıyla görüntüleme sensörüne yansıtılan görüntünün parlaklığını kontrol etmek için kullanılır. Gözün irisi, gözbebeğini (ışık açıklığı) büyütmek için kasılabilen opak kas dokusundan oluşur. Bir kameradaki iris, aynı işlevi gören mekanik bir düzenektir.

## **3.2. Filtre Uygulamaları**

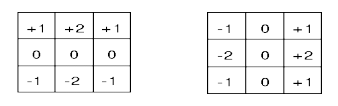
Filtreleme, bir görüntünün belirli özelliklerini vurgulamak veya diğer özelliklerini kaldırmak için kullanılabilen bir görüntüyü değiştirme veya iyileştirme tekniğidir. Filtreleme, çıkış görüntüsündeki herhangi bir pikselin o piksele göre değerinin, ilgili giriş pikselinin komşuluğundaki piksellerin değerlerine bazı algoritmalar uygulanarak belirlendiği bir komşuluk işlemidir; burada pikselin komşuluğu, konumları tarafından tanımlanan bir piksel kümesidir.

Aşağıda bazı filtrelere genel bir bakış bulunmaktadır:

1-) Sobel Filtresi:

Sobel filtre maskeleri, görüntü kenarı algılaması için kullanılır; hem yatay hem de dikey yönlerde kenarları aralar ve ardından bu bilgi parçasını tek bir metrikte birleştirirler. Maskeler aşağıda Şekil.3.2'de gösterilmektedir.

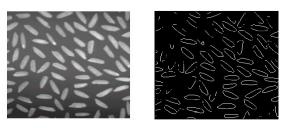
Sıra Maskesi Kolon Maskesi



**Şekil 3.1**. Sobel filtre maskeleri

Bu maskelerin her biri görüntüyle birleştirilmiştir. Şekil 3.2'de aşağıdakilere ilişkin bir örnek verilmektedir:

Sobel filtre kenar algılama:



a-) Orijinal fotoğraf b-) Sobel filtresinden sonraki görüntü

**Şekil 3.2**. Sobel filtre kenarı algılama örneği

2-)Gauss bulanıklık filtresi:

Görüntüler üzerinde yumuşatma etkisine sahiptir. Aslında bir alçak geçiren filtredir. Dairesel olarak simetrik olmanın yanı sıra, çeşitli yönlerdeki kenarlar ve çizgiler de benzer şekilde ele alınır. Gauss bulanıklaştırma filtrelerinin bazı çok avantajlı özellikleri vardır:

* Yatay ve dikey vektörlerin çarpımı olarak ayrılabilirler.
* Büyük çekirdekler, küçük çekirdeklerin sıralı uygulamasına ayrıştırılabilir. Bu nedenle, Gauss bulanıklık filtresinin filtre çekirdeği, bir dikey vektör ve bir yatay vektörün ürününde ayrıştırılabilir. Birbirleriyle çarpıldığında, gauss bulanıklık filtreleri üreteceklerdir.

Gauss bulanıklık filtresinden etkilenmiş bir görüntü örneği Şekil 3.3’te verilmiştir.

a-) Orijinal fotoğraf b-) Gauss filtresinden sonraki görüntü

**Şekil 3.3**. Gauss bulanıklaştırma filtresi örneği

3-) Ortalama filtresi:

a-) Orijinal fotoğraf b-) Ortalama filtresinden sonraki görüntü

### **3.2.1. Histogram Eşitleme**

Histogram eşitleme, Histogram modelleme tekniklerine bir örnektir, bir görüntünün dinamik aralığını ve kontrastını, o görüntüyü yoğunluk histogramı istenen bir şekle sahip olacak şekilde değiştirerek karmaşık bir yöntem sağlar. Şekil 3.4’te örneğini görebilirsiniz.



a-) Orijinal fotoğraf b-) Histogram eşitleme görüntüsü

**Şekil 3.4**. Histogram eşitleme filtresi

## **3.3. Özet**

Bu bölümde, görüntülerin bazı temel bileşenlerini ve bunların görüntüler üzerindeki bazı uygulamalarını tanımladık. Görüntüleri daha kesin ve net hale getirmek için bazı filtreleme teknikleri uygulanmaktadır. Filtre uygulamaları bu bölümde anlatılmaktadır.

# **4. Projenin İlerleme Aşamaları**

## **4.1. Uygun Çalışma Ortamının Belirlenmesi**

İris tanıma projesi için kullanılabilecek birçok çalışma ortamı tespit ettik. Bunlar; Matlab, Raspberry Pi, Python IDE vb. Bu tespitlerimizi kullanım kolaylığına, deneyimlerimize, kaynak çokluğuna göre belirledik. Elimizde halihazırda bulunması, kullanımının ve taşınmasının kolay olması açısından Raspberry Pi geliştirme kartını seçtik. Bizde bulunan kartın modeli Raspberry Pi Model-2B’dir. Şekil 4.1.’de görülmekte ve özellikleri aşağıda bulunmaktadır[1].

****

**Şekil 4.1.** Raspberry Pi Model 2B örneği

**Raspberry Pi 2 Teknik Özellikleri**

4 çekirdek ARMV7 Cortex 900 MHZ işlemciye sahip

1 GB LPDDR2 SD RAM

10/100 Ethernet RJ45 jack

4 x USB2.0 port

Video/Ses Çıkışı: HDMI ve 4-kutup 3.5mm konektör

microSD kart yuvası

40 Pin GPIO

Güç girişi: 5V 600mA mikro-usb adaptör ve GPIO üzerinden

Windows 10, Raspian, Debian, Fedora, Arch, RISC OS ve daha birçok OS desteği

Kart boyutları 8.6cm x 5.6cm x 2.0cm

## **4.1.1. Raspberry Pi İşletim Sistemi Yüklenmesi**

Raspberry Pi geliştirme kartına işletim sistemini yüklemek için ihtiyacımız olan şeyler sadece bir bilgisayar ve en az 8 GB hafızaya sahip bir SD kart. Alışılmışın dışına çıkan Raspberry Pi işletim sistemini hafıza kartında saklamaktadır.

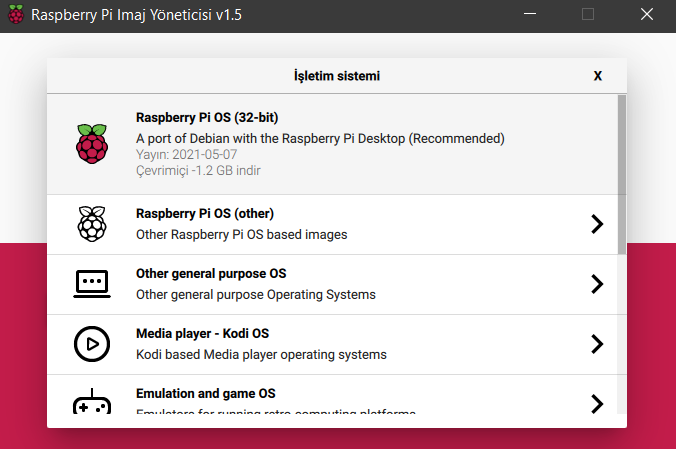
Raspberry Pi resmi sitesinden indirilebilecek bir program vasıtasıyla, kolay adımlarla işlem tamamlanabilmektedir. Yapılması gereken adımlar şöyle ilerlemektedir;

* Raspberry Pi sitesi üzerinden imager indirilir[2].
* Hafıza kartı bilgisayara takılır.
* Program açılır ve Şekil 4.2.’deki gibi görünür



**Şekil 4.2.** Raspberry Pi imager ekranı

* İşletim sistemi seçin kısmından yüklenmek istenilen işletim sistemi seçilir.
* İçerisine Linux tabanlı Debian işletim sistemini yükleyeceğiz.

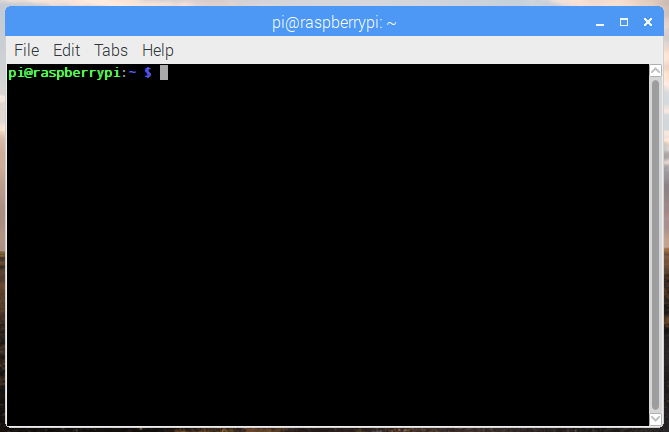


**Şekil 4.3.** Raspberry Pi İşletim Sistemleri

* Yüklemek istenilen SD kart da seçildikten sonra yükleme işlemi başlar ve bir süre içerisinde tamamlanarak kullanıma hazır hale gelir.

## **4.1.2. Raspberry Pi Güncelleme ve Yükseltme**

Hafıza kartı Raspberry Pi’a takılıp, güç ve internet bağlantıları gerçekleştirildikten sonra kurulum ayarı ekranı çıkar ve kişisel olarak ayarlanır. Bunlardan sonra yazılımsal olarak güncel olup olunmadığı kontrol edilir. Kartımız Linux tabanlı işletim sistemine sahip olduğundan indirme, güncelleme, yükseltme, kaldırma işlemlerini terminal üzerinden birkaç kod ile gerçekleştirebiliriz. Şekil 4.4.’te terminali görebilirsiniz.



**Şekil 4.4.** Raspberry Pi Terminal Ekranı

Raspberry Pi içerisindeki bütün yazılımları update komutu ile güncelleyebiliriz. Sudo ise yönetici olarak işlemleri yaptığımızı belirtir.

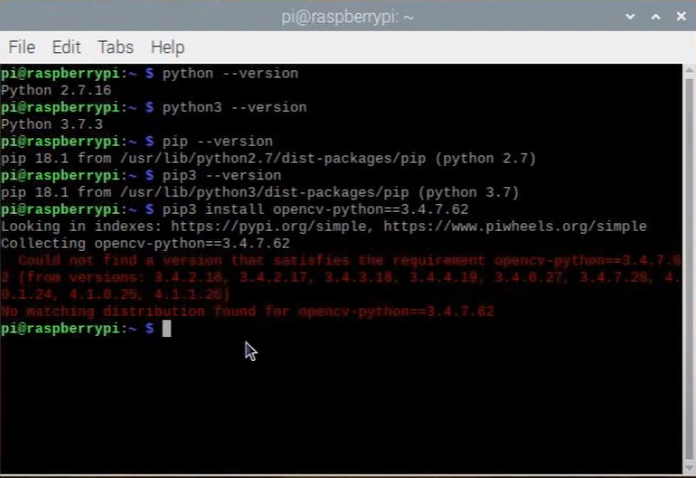


İndirilen güncel yazılım ve programların yüklenmesi gerekmektedir. Yine aynı metodoloji ile upgrade komutu kullanılarak indirilen bütün yazılımlar yüklenir.



## **4.1.3. Raspberry Pi OpenCV Kütüphanesi Yükleme**

Görüntü işleme tekniklerinin olmazsa olmazı tabiki de OpenCV kütüphanesi’dir. Bu kütüphane sayesinde ekstra birçok şeye ihtiyaç duymadan gerekli işlemler yürütülebilir. OpenCV kütüphanesini yükleyebilmek için bulunan python versiyonlarının son sürüm olması önem arz ediyor. Bu kontrollerden sonra rastgele sürüm numarası ile OpenCV yüklenmeye çalışılır. Hemen altta Şekil 4.5.’te görüldüğü gibi mevcut indirilebilecek OpenCV sürümleri belirir. Bunlardan biri seçilerek yüklenir.



**Şekil 4.5.** Raspberry Pi OpenCV sürümleri

Raspberry Pi 5V/2A değerlere sahip bir adaptör ile beslenmektedir. Çalışırken kendisine, kameraya, fareye, klavyeye enerji sağlayıp işlemleri yürütmesi gerektiğini varsayarak, Model 2B’nin özelliklerinin yeterli gelmeyeceğini düşünerek ve gerekli bazı kütüphanelerin yüklenmesinin zorluk çıkardığını görerek bilgisayar üzerinden işlerimize devam etme kararı aldık.

## **4.2. PyCharm Python IDE Kullanılması ve Gerekli Yazılımların Yüklenmesi**

Bilgisayarda Python ile kodlamaya başlamadan önce uygun Python ve OpenCV sürümlerinin indirilmesi gerekmektedir. Sürüm numaraları birbirleriyle uyumlu çalışması açısından önem arz etmektedir[3]. Python’ın kendi resmi sitesi üzerinden Python 3.6.8 sürümü indirip kurmak yeterli olacaktır. Ancak OpenCV kütüphanesi için her şey bu kadar kolay değil. Şekil 4.6.’da görüldüğü gibi birçok faklı OpenCV sürümü bulunmaktadır. Bu sürümleri Kaliforniya Üniversitesinin sitesinden bulabilmekteyiz[4].



**Şekil 4.6.** OpenCV sürümleri

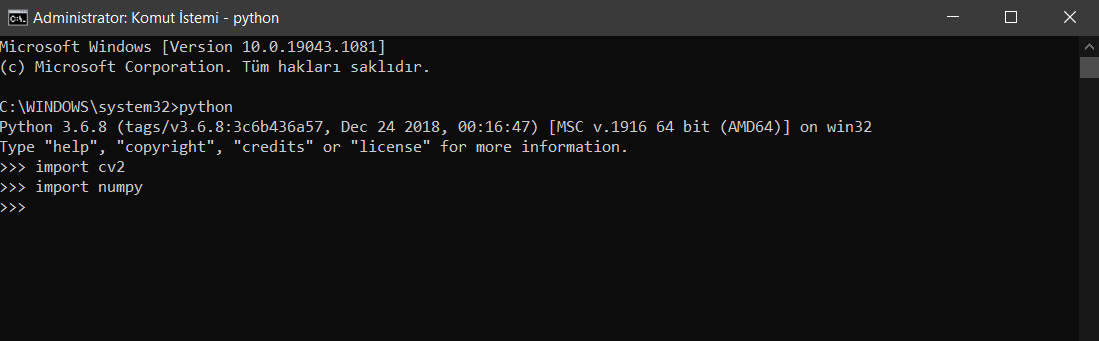
Şekil 4.6.’dan görüldüğü üzere birçok Python sürümü için OpenCV sürümü bulunmaktadır. Burada cp39 kavramı Python 3.9 sürümü ile uyumlu olduğunu belirtmektedir. cp34 ise Python 3.4 sürümü ile uyumlu olduğu anlamına geliyor. Dikkat edilirse bazı sürümlerde contrib yazmaktadır. Bunlar da içlerinde birçok kütüphaneyi barındırmaktadır. Python 3.6.8 sürümü kullandığımız için win32 için 4.4.0 sürümlü cp36 ile uyumlu olanı indirip yükleyeceğiz.

Sonrasında yapmamız gereken komut istemini yönetici olarak çalıştırmak. İndirdiğimiz OpenCV sürümü herhangi bir klasörde olabilir. Onu bulup bulunduğu klasöre geçiş yapabilmek için yolunu kopyalamamız gerekmekte. Örneğin şöyle bir yol: C:\Users\pinolo\Downloads.

Komut istemi içerisine “cd C:\Users\pinolo\Downloads” yazarak gitmek istediğimiz klasörün içerisine gidilebilir. Ardından (python pip -m install “indirilen dosya ismi”) ile kurulum yapılabilir. Yazılması gereken: (python pip –m install opencv\_python 4.4.0-cp36-cp36m-win32.whl)

Tüm bunların yanında matematiksel işlemlerimizi yapmak için numpy kütüphanesini de yükleyelim. Bunun için yapmamız gereken sadece “python pip -m install numpy” yazmak.

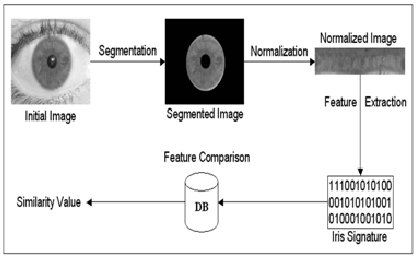
Kurulan kütüphanelerin çalışıp çalışmadığını komut istemi üzerinden kontrol edebiliriz. Python yazarak python içerisine giriyoruz ve import cv2 ve import numpy komutlarını giriyoruz. Herhangi bir hata almadığımız için yüklemelerin başarılı olduğu anlaşılabilmektedir.



**Şekil 4.7.** Kütüphanelerin kontrolü

# **5. İris Tanıma Aşamaları**

İris, gözün ön kısmında bulunan ve fibroz (lifli) dokudan oluşan renkli tabakadır. İriste 250’den fazla görsel karakteristik bulunmaktadır. Bunlar daireler, benekler, çizgiler gibi belirleyici şekillerdir. İris, bebek embriyo olarak anne karnındayken oluşur ve insanın ölümüne kadar değişmez. İris tarama biyometrik taramalar içerisinde en basit olanlarından biridir. Bu özelliklerden yaralanarak iris tabanlı bir uygulama gerçekleştirilmiştir. Şekil 5.1’de ilerleyiş görülebilmektedir. İlk olarak görüntü kaydedilir. Bunun için kameradan belli aralıklarla kare alınır. Tuşa basıldığı esnada alınan kare işleme aşamaları için kaydedilir. Kaydedilen görüntü analizin daha da kolaylaşması amacıyla RGB renginden Gray rengine dönüştürülür. Graye dönüştürülmüş görüntüye blurlama uygulanır. Blurlama yapıldıktan sonra görsel içerisindeki daireler bulunur. Bu daireler en başta bizim verdiğimiz parametrelere göre bulunur. Bulunan dairenin yarıçapına göre gözün iris kısmının ayırt edilmesi kolaylaşır.



**Şekil 5.1.** İlerleyiş

**Şekil 5.2.** Aşamalar

## **5.1. Görüntü Alma**

Görüntü alma aşamasında kamera kalitesi, ortamın aydınlatılması gibi etkenler büyük öneme sahiptir. Kamera olarak elimizde bulunan 6 Megapiksel görüntü kalitesine ve 10x zoom özelliğine sahip olan Inca IC-3562 harici kamera kullanıldı. Performansı gayet iyi olan bu kameranın kendi aydınlatmasına ve manuel olarak ayarlanan zoom’a sahip olması büyük avantaj sağladı. Aydınlatmasının kullanılması sayesinde alınan her görüntünün birbirine benzer görüntü özelliklerine sahip olması ve iris kısmının çok net görülmesi sağlanmıştır. Bu özelliği karşılaştırma işlemlerini daha kararlı hale getirmektedir. Şekil 5.3.’te kullanılan kamera görülmektedir.



**Şekil 5.3.** Inca IC-3262

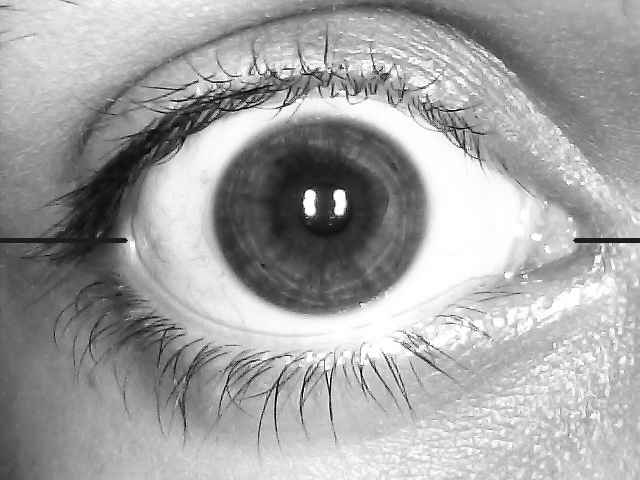
Şekil 5.4.’te alınan göz görüntüsüne dikkat edilirse sağ ve sol tarafta birer adet mavi çizgi bulunmaktadır. Görüntü kaydetme aşamasında göz canlı olarak görülebilmektedir. Bu çizgilerin uçları Şekil 5.4.’teki gibi gözün her iki yanına değmesi sağlanmalıdır. İrisin kağıt modeli çıkartılırken her zaman aynı noktadan başlaması için bu konu önem arz etmektedir.



**Şekil 5.4.** Alınan Göz Görüntüsü

## **5.2. İris Algılama**

Alınan görüntü ilk olarak RGB renk skalasından Gray renk skalasına dönüştürülmektedir. Bu dönüşüm kullanılan diğer yöntemlerin daha etkili olmasını sağlamaktadır. Şekil 5.5.’te görüntünün Gray filtresi uygulanmış hali görülmektedir.



**Şekil 5.5.** Görüntünün Gray Filtresi Uygulanmış Hali

Görüntü üzerinde bulunan gürültülerin yok edilmesi ve görüntünün daha saf hale getirilmesi ve göz çevresinin daha rahat tespit edilmesi için Medyan Filtresi uygulanır. Şekil 5.6.’da Medyan filtresi uygulanmış hali bulunmaktadır.



**Şekil 5.6.** Medyan Filtresi Uygulanmış Hali

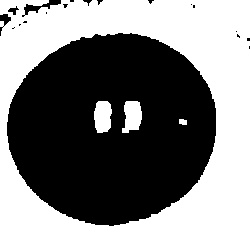
Medyan fitresi sayesinde görüntünün her yeri daha bulanık olur ve bulunması gereken iris çevresi daha rahat bulunur. Aksi halde görüntünün birçok yerinde istenilmeyen dairelerin bulunması büyük olasılıktır. Görüntü içerisindeki dairelerin bulunmasında Hough Circles fonksiyonu kullanıldı. Bir çemberin kartezyen koordinat sisteminde tespiti aşağıdaki formülle bulunur.

Burada a ve b noktaları çemberimizin merkez noktasının koordinatları, r ise çemberimizin yarıçapıdır. Hough Circles fonksiyonu resim üzerinde hayali daireler çizerek görüntüdeki gerçek çemberler üzerinde denetleme işlemi gerçekleştirir. Hayali dairelerimizi ise OpenCV'nin içinde bulunan HoughCircles fonksiyonu ve parametreleri ile ayarlıyoruz. İris tespit edildikten sonra görüntü iris içinde kalacak şekilde küçültülür ve Şekil 5.7.’deki halini alır.



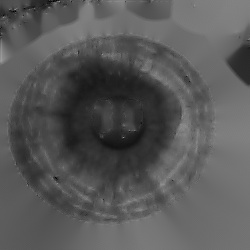
**Şekil 5.7.** Görüntünün Küçültülmüş Hali

Yapılan yukarıdaki işlem ile bulunan dairenin yarıçapı da kaydedilmiş oldu. Bu yarıçap kullanılmadan önce küçültülmüş iris görüntüsüne eşik değeri uygulanarak siyah ve beyaz hale dönüştürülür ve renk ayırımları daha net hale getirilir. Şekil 5.8.’de görülebilmektedir.



**Şekil 5.8.** Threshold Uygulanmış Hali

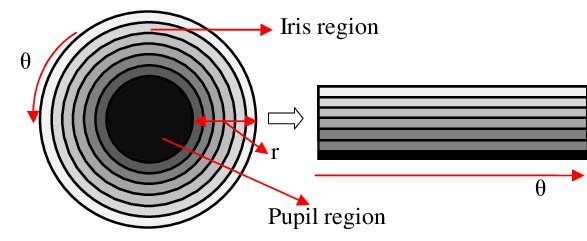
Siyah kısım dışında kalan kısımlar(İris içerisinde görünen 2 adet beyaz genişlik dahil) işimize yaramayacağı için küçültülmüş görüntü üzerinden soluklaştırılır. Ve Şekil 5.9.’daki halini alır.



**Şekil 5.9.** Boyanmış Hali

## **5.3. İris Normalleştirme**

İnsanların iris boyutları birbirinden ayrıdır ve farklı özelikler gösterilebilir. Aynı kişilerin iris resimleri aydınlanmadan dolayı farklı gözükebilir. Bu farklılıklar görüntünün normalize edilmesi ile ortadan kaldırılır. Bu yönteme Daugman Rubber Sheet modeli denilmektedir. Yöntemin işleyişi Şekil 5.10.’da görülmektedir. Kutupsal koordinat sisteminden Kartezyen koordinat sistemine geçiş yapılmaktadır.



**Şekil 5.10.** Normalleştirme tekniği

Normalleştirme işlemi sırasında 1 adet 60x360 piksel boyutlarında boş görüntü oluşturulur. Kutupsal halden alınan her bir değer bu görüntüye aktarılır ve Şekil 5.11.’deki halini alır. İçerisinde 21600 adet piksel bulundurmaktadır.

C:\Users\Burak\AppData\Local\Microsoft\Windows\INetCache\Content.Word\daugman.jpg

**Şekil 5.11.** İrisin Normalleştirilmiş Hali

## **5.4. Özellik Çıkarma**

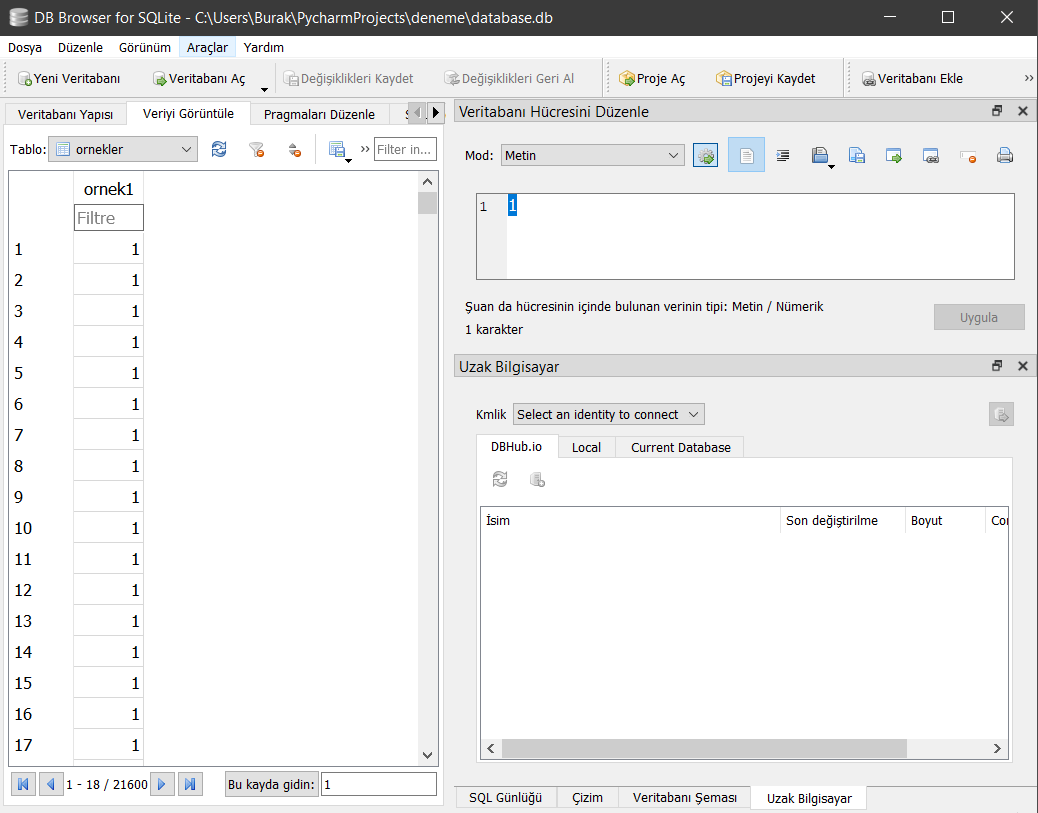
Özellik çıkarım aşamasında görüntü piksellerinin değerleri baz alınmaktadır. Gray renk skalasında olduğumuz için RGB renk değerlerinin hepsi aynı değer çıkmaktadır. Daha kararlı bir özellik çıkarma yapmak için bütün piksel değerleri toplanır ve toplam piksel sayısı ile orantılı bir sayı ile bölünerek ortalama bir değer elde edilmektedir. Bu değerin altında veya üstünde kalan pikseller siyah veya beyaz olmaktadır. Eğer beyazsa 255 değerini, siyahsa 0 değerini almaktadır. Bu değerler 0 ve 1 olarak düzenlenir. Şekil 5.11.’den de anlaşılacağı üzere irisin ayırt edici özelliği beyaz kısımlar olmaktadır. Bu nedenle beyaz kısımlar siyaha, siyah kısımlar beyaza dönüştürüldü. Şekil 5.12.’de son hali görülmektedir.

C:\Users\Burak\AppData\Local\Microsoft\Windows\INetCache\Content.Word\the.jpg

**Şekil 5.12.** İrisin Özellik Çıkarılmış Hali

## **5.5. Veri Tabanı Oluşturma**

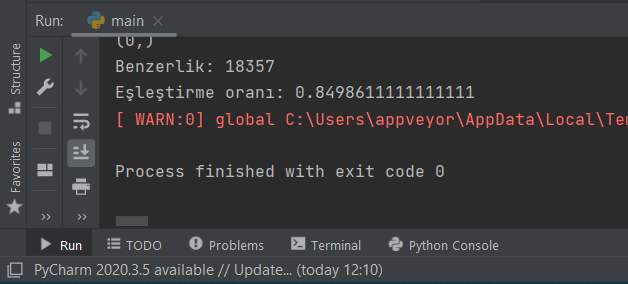
Veri tabanı oluşturmak için Python’ın sqlite 3 kütüphanesi kullanıldı. Özelliği çıkartılmış görüntü Ek-1’de bulunan kod vasıtasıyla özellik çıkarma aşamasında belirtilen yöntemler kullanılarak kısaca ortalaması alınarak database e kaydedildi ve database.db dosyası oluşturuldu. Bu dosya Ek-2’de bulunan ana kodun bulunduğu klasöre gönderildi ve karşılaştırma için ana kodun çalıştırılması yeterli olacaktır. Database içerisindeki değerleri DB Browser for SQlite programı sayesinde görebilmekteyiz. Şekil 5.13.’de database içerisindeki değerleri görebilmekteyiz.



**Şekil 5.13.** Database İçindeki Değerler

## **5.6. Karşılaştırma**

Ana kod başlatıldığı anda kamera görüntüsü ekrana yansımaktadır. Sağ ve solda bulunan çizgiler yerlerinde yerleştirildikten sonra karşılaştırmanın başlaması için 0’a basmak yeterli olacaktır. Bu esnadan sonra yeni gelen görüntüye yukarıdaki bütün işlemler uygulanmaktadır. Karşılaştırma yöntemi olarak Hamming Distance yöntemi kullanılmaktadır. Bu yöntemle database içerisine kaydedilen ve yeni alınan görüntü arasındaki benzerlik oranı bulunabilmektedir. Aynı gözler arasındaki bir karşılaştırma sonucunu Şekil 5.14.’te görebilmekteyiz. Yüzdesel olarak benzerlik oranımız %84.98 çıkmaktadır. Toplam piksel sayımız 21600 ve eşleşen piksel sayımız ise 18357 çıkmaktadır.



**Şekil 5.14.** Karşılaştırma Sonucu

# **6.** **Sonuçlar**

Projede insan gözünün hiçbir zaman değişmeyen iris bölgesinin belirlenmesi ve kişinin tanınması için bir sistem geliştirilmiştir. Sistem donanımı bilgisayar, kamera ve kameranın aldığı görüntülerden oluşmaktadır. Yazılım kısmı ise ikiye ayrılmaktadır. İlk kısımda irisin yeri birçok filtrenin yardımı ile tespit edilir. Uygun dikdörtgen model çıkartılır ve özellik çıkarmaya hazır hale getirilir. İkinci kısım ise database oluşturma kısmıdır. Özellik çıkarmaya uygun olan görüntü üzerinden ortalama değerler baz alınarak görüntünün piksel değerleri ortaya çıkar ve özellikleri çıkartılmış olur.

Bu özelliklerin karşılaştırılması esnasında kullanılan Hammig Distance yönteminin bazı olumsuz yönleri bulunmuştur. Kamera ile kullanıcı arasındaki mesafenin, kameranın veya kullanıcı gözünün açısının farklı olması (dikdörtgen model çıkartılırken sorun yaratıyor) gibi etkenler uyuşma oranını büyük derecede etkilemektedir. Bu olumsuz yönleri ortadan kaldırmak için şunlar yapıldı; kameranın zoom ayarının sabit kalması ile gözün her zaman aynı mesafede netleşmesi ve bu sayede veri tabanındaki görüntü ile mesafe farkının az olması, ekranın sağ ve soluna eklenen iki adet çizgi ile kamera ve göz arasındaki açının her zaman veri tabanındaki görüntüdeki açı ile çok yakın olması.

Eşleştirme oranı sabit bir mekanik tasarım yapılarak da arttırılabilir. Bu mekanik tasarımlara göz hastanelerinde muayene olurken kullanılan tasarımların örnek olarak kullanılması mümkündür.

Kaynaklar

[1] https://www.muhendisbeyinler.net/raspberry-pi-2-model-b/

[2] https://www.raspberrypi.org/software/

[3] https://pysource.com/2019/03/15/how-to-install-python-3-and-opencv-4-on-windows/

[4] https://www.lfd.uci.edu/~gohlke/pythonlibs/#opencv

Ekler

Ek- 1: Database Oluşturma

import cv2  
import sqlite3  
  
baglanti = sqlite3.connect("database.db")#database oluşturur, yoksa kurar, varsa bağlanır  
imlec = baglanti.cursor()  
  
imlec.execute("CREATE TABLE IF NOT EXISTS ornekler (ornek1 INT)")  
  
img = cv2.imread('daugman.jpg')  
  
u = 0  
for i in range(60):  
 for j in range(360):  
 r, g, b = img[i, j]  
 u += r  
 ortalama = int(u / (21600 / 1.3))  
ret, threshold = cv2.threshold(img, ortalama, 255, cv2.THRESH\_BINARY\_INV)  
cv2.imwrite("thresh.jpg", threshold)  
print("ortalama", ortalama)  
  
x = [0]\*21600  
l = 0  
for a in range(60):  
 for b in range(360):  
 r, g, b = threshold[a, b]  
  
 if r == 255:  
 x[l] = 1  
 imlec.execute("INSERT INTO ornekler VALUES (1)")  
 else:  
 x[l] = 0  
 imlec.execute("INSERT INTO ornekler VALUES (0)")  
  
 l = l + 1  
l = 0  
  
imlec.execute("""SELECT \* from ornekler""")  
data = imlec.fetchall()  
k = 0  
y = [0]\*21600  
for i in range(21600):  
 print("data: ", data[i])  
 if data[i] == (0,): y[i] = 0  
 else: y[i] = 1  
 print(y[i])  
 k += 1  
k = 0  
  
baglanti.commit()  
baglanti.close()  
  
cv2.waitKey(0)  
cv2.destroyAllWindows()

Ek- 2: Ana Kod

import cv2  
import numpy as np  
import keyboard  
import sqlite3  
  
cap = cv2.VideoCapture(2)  
  
baglanti = sqlite3.connect("database.db")#database oluşturur, yoksa kurar, varsa bağlanır  
imlec = baglanti.cursor()  
  
imlec.execute("CREATE TABLE IF NOT EXISTS ornekler (ornek1 INT)")  
  
def eslestirme():  
 gelen = cv2.imread('ornek.jpg')  
 image\_roi, rounds = processing(gelen, 30)  
 image\_roi = recflection\_remove(image\_roi)  
 image\_nor = daugman\_normalizaiton(image\_roi, 60, 360, rounds, 55)  
 cv2.imwrite("daugman.jpg", image\_nor)  
 özellik = özellikcikar(image\_nor)  
 print(özellik)  
 deger = karşılaştır(özellik)  
 return deger  
  
def processing(image\_path, r):  
 image\_path = cv2.resize(image\_path, (640, 480), interpolation=cv2.INTER\_LINEAR)  
 image\_path = cv2.cvtColor(image\_path, cv2.COLOR\_BGR2GRAY) #graya dönüştürür  
 gray = cv2.medianBlur(image\_path, 11) #Blur yapar ve gürültüleri siler  
 ret, \_ = cv2.threshold(gray, 0, 255, cv2.THRESH\_BINARY + cv2.THRESH\_OTSU) #Thresh binnary eşik değeri ile siyah ve beyazı ayırır  
 cv2.imwrite('gray.jpg', gray)  
 cv2.imwrite('processing.jpg', image\_path)   
 circles = cv2.HoughCircles(gray, cv2.HOUGH\_GRADIENT, 1, 50, param1=ret, param2=30, minRadius=20,  
 maxRadius=100)  
 circles = circles[0, :, :]  
 circles = np.int16(np.around(circles))   
 for i in circles[:]:   
 image\_path = image\_path[i[1] - i[2] - r:i[1] + i[2] + r, i[0] - i[2] - r:i[0] + i[2] + r]  
 radus = i[2]/(2.5)  
 print("radus:", radus)  
 cv2.imwrite('circle.jpg', image\_path)  
 return image\_path, radus  
  
def recflection\_remove(img):  
 ret, mask = cv2.threshold(img, 150, 255, cv2.THRESH\_BINARY)  
 kernel = np.ones((5, 5), np.uint8)  
 dilation = cv2.dilate(mask, kernel, iterations=1

dst = cv2.inpaint(img, dilation, 4, cv2.INPAINT\_TELEA)   
 cv2.imwrite('dilation.jpg', dilation)  
 cv2.imwrite('dst.jpg', dst)  
 return dst  
  
def daugman\_normalizaiton(image, height, width, r\_in, r\_out):  
 thetas = np.arange(0, 2 \* np.pi, 2 \* np.pi / width) # Theta değerleri (0, 2\*pi, 2\*pi/640) 0 ile 360 arasında 360/640 derece aralıklarla  
 r\_out = r\_in + r\_out  
  
 flat = np.zeros((height, width, 3), np.uint8)#boş bir fotoğraf yarat  
 circle\_x = int(image.shape[0] / 2)  
 circle\_y = int(image.shape[1] / 2)  
  
 for i in range(width):  
 for j in range(height):  
 theta = thetas[i] # theta koordinat değerleri  
 r\_pro = j / height # yarıçap koordinat(normalized) değerleri  
  
 Xi = circle\_x + r\_in \* np.cos(theta)  
 Yi = circle\_y + r\_in \* np.sin(theta)  
 Xo = circle\_x + r\_out \* np.cos(theta)  
 Yo = circle\_y + r\_out \* np.sin(theta)  
  
 Xc = (1 - r\_pro) \* Xi + r\_pro \* Xo  
 Yc = (1 - r\_pro) \* Yi + r\_pro \* Yo  
  
 color = image[int(Xc)][int(Yc)] # pixellerin renkleri  
  
 flat[j][i] = color  
 return flat   
  
def özellikcikar(image):  
 u = 0  
 for i in range(60):  
 for j in range(360):  
 r, g, b= image[i, j]  
 u += r  
 ortalama = int(u/(21600/1.3))  
 ret, threshold = cv2.threshold(image, ortalama, 255, cv2.THRESH\_BINARY\_INV)  
  
 print("ortalama:", ortalama)  
 x = [0] \* 21600  
 l = 0  
 for c in range(60):  
 for d in range(360):  
 r, g, b = threshold[c, d]  
 if r == 255:  
 x[l] = 1  
 else:  
 x[l] = 0  
 l = l + 1  
 l = 0  
 cv2.imwrite('the.jpg', threshold)  
 return x  
  
def karşılaştır(x):  
 y = [0] \* 21600  
 sayac = 0  
 imlec.execute("SELECT \* FROM ornekler")  
 data = imlec.fetchall()  
 for i in range(21600):  
 print(data[i])  
 print(x[i])  
 if x[i] == 0:  
 y[i]=(0,)  
 else: y[i] = (1,)  
 print(y[i])  
 if data[i] == y[i] :  
 sayac += 1  
 print("sayac:", sayac)  
 sonuc = sayac/21600  
 return sonuc  
  
a = 1  
while True:  
 \_, frame = cap.read()  
 frame2=frame  
 frame2 = cv2.line(frame2, (0, 240), (125, 240), (255, 0, 0), 3)  
 frame2 = cv2.line(frame2, (575, 240), (640, 240), (255, 0, 0), 3)  
 cv2.imshow("resim", frame2)  
  
 if a == 1:  
 print("Lütfen eşleştirmeyi yapmak için 0'a basın!\n"  
 "Kapatmak için herhangi bir tuşa basın!")  
 if keyboard.is\_pressed('0'):  
 cv2.imwrite('ornek.jpg', frame)  
 oran = eslestirme()  
 print("Eşleştirme oranı:", float(oran))  
 a = 1  
 break  
 else:  
 a = 0  
 if cv2.waitKey(1) & 0xFF == ord('s'):  
 break  
  
cap.release()  
cv2.destroyAllWindows()

Özgeçmiş

**Yunus YILMAZ**

**KİŞİSEL BİLGİLER**

**Doğum Yeri :** Aksaray

**Doğum Yılı :** 03.01.1997

**Uyruğu :** Türk

**E-posta :** yunusyilmazeem@gmail.com

**Yabancı Diller :** İngilizce (Başlangıç: A1)

**ARAŞTIRMA DENEYİMİ**

* C, C++, MATLAP, Python, Verilog
* Multisim, Proteus, Autocad

Özgeçmiş

**Burak ELHAMAN**

**KİŞİSEL BİLGİLER**

**Doğum Yeri :** Diyarbakır

**Doğum Yılı :** 16.08.1997

**Uyruğu :** Türk

**E-posta :** burak.elhaman2@gmail.com

**Yabancı Diller :** İngilizce (Orta: B1)

**ARAŞTIRMA DENEYİMİ**

* C, C++, Assembly, Verilog, Vhdl, Python, MATLAP
* Multisim, Proteus(Ares, Isis), Autocad