

**T. C.
BOZOK ÜNİVERSİTESİ
MÜHENDİSLİK MİMARLIK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**



PYTHON İLE YÜZ ALGILAMA MODELİ GELİŞTİRİLMESİ

**Berk BAYRAKTARGİL
16008117059**

BMH481 BİTİRME PROJESİ

YOZGAT 2022

BOZOK ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ



PYTHON İLE YÜZ ALGILAMA MODELİ
GELİŞTİRİLMESİ

Berk BAYRAKTARGİL

16008117058

Bitirme / Uygulama Projesi Danışmanı:

Dr. Öğr. Üyesi Ahmet Sertol KÖKSAL

YOZGAT 2022

T.C.
BOZOK ÜNİVERSİTESİ
MÜHENDİSLİK MİMARLIK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

PYTHON İLE YÜZ ALGILAMA MODELİ GELİŞTİRİLMESİ

Berk BAYRAKTARGİL
I. ÖĞRETİM

Danışman: Dr. Öğr. Üyesi Ahmet Sertol KÖKSAL

Bölüm Başkanı: Doç. Dr. Mehmet BAKIR

YOZGAT 2022

TEŞEKKÜRLER

Eğitim hayatımın en başından beri desteğini esirgemeyen, konu içerisinde beni daima yönlendiren ve her koşulda yanımda olan saygıdeğer hocam, bitirme tez çalışma danışmanım sayın Dr. Öğr. Üyesi Ahmet Sertol KÖKSAL'a, hayatımın her anında yanımda olduklarını hissettiğim aileme ve son olarak Lisans eğitimim süresince beni hiç yalnız bırakmayan sevgili arkadaşlarım ve saygıdeğer hocalarıma teşekkür ederim.

Temmuz 2022

ÖZET

Bu çalışma, günümüzde kullanılan yüz tanıma sistemlerinin çalışma prensiplerini, arka planda kullanılan teknolojik alt yapılarını araştırma ve örnek bir uygulama geliştirmeyi amaçlanmaktadır.

Derslerde alınan yoklama işlemi, öğrencilerin derslerine katılma oranını ve performansını artırıcı bir faktördür. Öğrenciler arasında dağıtılan yoklama kağıdıyla veya öğretmenin isimleri sesli bir şekilde okuyarak alınan yoklamalar hem zaman alıcı hem de hileli olabilmektedir. Bir iş yerinin giriş çıkışlarını kontrol etmede kullanılması kolay ve hızlı bir şekilde işyelişi dolayısıyla çok önemlidir. Birçok alanda yüz tanıma sistemlerinin faydasını görmek mümkündür.

Çalışma, Yozgat Bozok Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği bölümü bitirme ödevi olarak hazırlanmıştır.

Proje kapsamında yüz tanımanın topluma sunabileceği, suçları önlemek ve güvenliği artırmaktan gereksiz insan etkileşimini ve emeğini azaltmaya kadar birçok faydası vardır. Bazı durumlarda tıbbi çabaları desteklemeye bile yardımcı olabilir, parmak izi gibi diğer güvenlik önlemleri türlerinden daha az insan kaynağı gerektirir. Hırsızlığa karşı önemli ölçüde korur. Güvenlik önlemi alma konusunda ciddi yardımı dokunur. Temas büyük ölçüde azalır. Eksileriyle ve artılarıyla yararlı bir teknolojidir.

Anahtar sözcükler: Görüntü Yakalama, Yüz Tanıma, Görüntü Tanıma, Yüz Algılama, Yüz bulma, özyüz, yerel ikili desen, yüz tanıma doğruluğu

ABSTRACT

This study aims to research the working principles of the face recognition systems used today, the technological infrastructures used in the background and to develop an example application.

Attendance in classes is a factor that increases students' participation rate and performance. Attendance, which is taken with the attendance sheet distributed among the students or by reading the teacher's names aloud, can be both time-consuming and fraudulent. It is very important to use it to control the entrances and exits of a workplace easily and quickly because of its business. It is possible to see the benefits of face recognition systems in many areas.

The study has been prepared as a final assignment of the Department of Computer Engineering, Faculty of Engineering, Yozgat Bozok University.

Within the scope of the project, there are many benefits that facial recognition can offer to society, from preventing crime and increasing security to reducing unnecessary human interaction and effort. In some cases it can even help support medical efforts, requiring less human resources than other types of security measures such as fingerprinting. Significantly protects against theft. It helps a lot in taking security measures. Contact is greatly reduced. It is a useful technology with its pros and cons.

Keywords: Image Capture, Face Recognition, Image Recognition, Face Detection, Face detection, eigenface, local binary pattern, face recognition accuracy

TEŞEKKÜRLER.....	iv
ÖZET	v
ABSTRACT.....	vi
ŞEKİL DİZİNİ	ix
ÇİZELGE DİZİNİ	x
KISALTMALAR	xi
1. GİRİŞ	1
1.1 Literasyon Çalışması.....	1
1.2 Yüz Tanıma Sistemleri	2
2. KULLANILAN TEKNOLOJİLER	3
2.1 Görüntü İşleme.....	3
2.2 Python	4
2.3 OpenCV Kütüphanesi.....	5
2.4 Python Yüz Tanıma	5
2.5 Tensorflow	6
3. YÜZ TANIMA NEREDE KULLANILIR.....	7
3.1 Şahsi Telefonun Kilidini Açma	7
3.2 Emniyet Kurumlarında Suçlu Tespiti.....	7
3.3 Sınır Denetimi.....	8
3.4 İnsan Kaçakçılığını Engelleme.....	8
3.5 Hırsızlık Gibi Suçlarının Azaltılması	8
3.6 Ödeme İşlemlerinin İyileştirilmesi.....	8
3.7 Bankacılık İşlemleri	9
3.8 Pazarlama	9
3.9 Sağlık Alanları.....	9
3.10 Eğitim Alanları	9
3.11 Trafik.....	9

3.12 Kumar	10
4. TASARIM SÜRECİ	10
4.1 Kütüphane Seçimi	10
4.2 Kodlama	12
5. UYGULAMADA KULLANILAN ÖNEMLİ FONKSİYONLAR.....	12
5.1 Yüz Bölgesinin Tespiti	12
5.2 Yüz Tanıma.....	13
6. SONUÇLAR VE ÖNERİLER.....	14
7. KAYNAKLAR	19
8. EKLER	21
9. ÖZGEÇMİŞ	45

ŞEKİL DİZİNİ

		<u>Sayfa No</u>
Şekil 1	Yüz Tanıma	11
Şekil 2	Görüntü İşleme Örnek Görsel	14
Şekil 3	Dil İşlevinin Gösterimi	15
Şekil 4	Yüz Tanıma Akış Şeması	16
Şekil 5	Haar Pikselleri Görseli	21
Şekil 6	Haar İle Seçilen Göz İçin Adımlar	23
Şekil 7	AdaBoost Algoritması Çalışma Mantığı Gösterimi	23
Şekil 8	Uygulamanın Anlık Yüz Tespiti Sağlamasına Ait Görsel	24
Şekil 9	Yüz Tespiti Sağlayan Güvenlik Cihazlarına Örnek Görsel	28

ÇİZELGE DİZİNİ

		<u>Sayfa No</u>
Çizelge 1	Uygulamanın Anlık Yüz Tespiti Hızının Ölçüldüğü Çizelge	27
Çizelge 2	FaceID ve Uygulamanın Tespit Süreleri	28
Çizelge 3	FaceID ve Uygulamanın Tespit Oranı	28

KISALTMALAR

PYT	Python
OpenCV	Open Source Computer Vision Library
BSD	Berkeley Software Distribution
HOG	Histogram Of Oriented Gradient
SVM	Support Vector Machine
MS	Milisaniye
S	Saniye

1. GİRİŞ

Gelişmekte olan teknoloji ile birlikte ihtiyaç ve imkanlara göre yüz tanıyan sistemler oldukça gelişmiştir. Asıl amacı güvenlik amaçlı kullanılan sistemler kullanım ihtiyacına göre farklılıklar gösterir, kullanılan yöntemlere göre ihtiyaç duyulabilecek donanım ve yazılım oldukça maliyeti artırabilir. Günümüzde yaygın olarak kullanılan veya bilinen tanıma sistemleri; parmak izi, yüz, el gibi çeşitli özelliklere göre tespit yapan sistemlerdir. Bu yazıda yüz tanıma sistemi ele alınacak ve kodlanacaktır.



Şekil 1 - Yüz Tanıma [22]

1.1 Literasyon Çalışması

Literatür taramasında benzer çalışma ve yaklaşımlar ile ilgili yeterli düzeyde döküman bulunmaktadır. Literatür taraması yaparken öncelikli olarak diğer çalışmaların hangi altyapıyla ne kadar sürede yüz tespiti sağladığı bilgiler esas alınmıştır. makale, konferans ve raporlar incelenmiş, yol haritaları belirlenmiştir.

Bu projede kullanılması amaçlanan python ile yüz tanıma için gerekli olan yüzün tespit yapılmasıdır. Elbette minimum hata ve minimum süre amaçlanmıştır.

Her insanın yüz yapısı doğal olarak farklıdır, simetrisi, yapısı, herhangi bir kaza sonucu yüzde oluşan yaralar, kısaca yaradılış temelinde insanın doğası farklılık

göstermektedir. Çene, burun T bölgesi ve alın olmak üzere bu yazılım üç farklı bölgeyi baz almaktadır. Yaş ilerlediği için yüz hatları farklılık gösterir buna rağmen farklılık göstermeyecek olması da genel olarak kesin sonuç vermede ideal bir yer tutmaktadır. Bilgisayar ortamında var olan güvenlik protokollerinden faydalanarak yüz yapısı algılanıp tespit edilecektir. Nesne tanıma ve sınıflandırma gibi işlemlerde de kullanılabilecektir.[2]

Python altyapılı yüz tanıma sistemleri farklı alanlarda kullanılabilir. Genel amacı güvenlik olmasına rağmen bazen hiç güvenli olmayabilir. Python ile geliştirilen yüz tanıma uygulamaları benzersiz şekilde işler. İlk olarak yüz tanıma algoritmalarının sabit olmayan hareketli görüntüler için Python kullanılır. Güvenlik amaçlı kullanılabildiği için polis, kolluk kuvvetleri tarafından tercih edilebilir. Genelde mobese sistemlerinde bulunması sayesinde suçluların daha kısa sürede yakalayabilirler.

Güvenlik alanında popüler olan Python yüz tanıma algoritmaları sürekli gelişime açıktır. Python ile yüz tanıma elbette kolay değildir, bu alan ciddi bir hizmet verecek yöndedir. Grafik işlemci üniteleri ve teknolojinin gelişmesinden dolayı daha hızlı çalışır hale gelmiştir. Verileri daha aktif şekilde işleyerek, özellikleri algılar. Verileri karşılaştırmada veya belirlemede etkili olmasının sebebi grafik işlemci üniteleri sayesinde. Bu yüzden en uygun bir yapıda destek vermeye de devam eder.[1]

1.2 Yüz Tanıma Sistemleri

Öncelikle yüz tanıma nedir nasıl çalışır mantığını her olayda olduğu gibi kavrayabilmek için temel düzeyde bilmek gerekir. Yüz tanıma, bir insanın yüzünü baz alarak kimliğini tanımlamak veya doğrulamaktır. Yüz tanıma sistemleri, insanları fotoğraflarda, videolarda veya anlık olarak tanımlamak tespit etmek için kullanılabilir.

Yüz tanıma, biyometrik güvenliğin bir kategorisidir. Bu teknoloji, çoğunlukla güvenlik alanında kullanılır. Peki yüz tanıma nasıl çalışır? Elbette birçok insan, telefonunun kilidini açmak için kullanılan örneğin Apple firmasının iPhone için geliştirdiği FaceID sayesinde yüz tanıma gerçekleştirir. Yüz tanıma bir kişinin kimliğini belirlemek için büyük bir fotoğraf veritabanı kullanmaz, sadece tek kişiyi

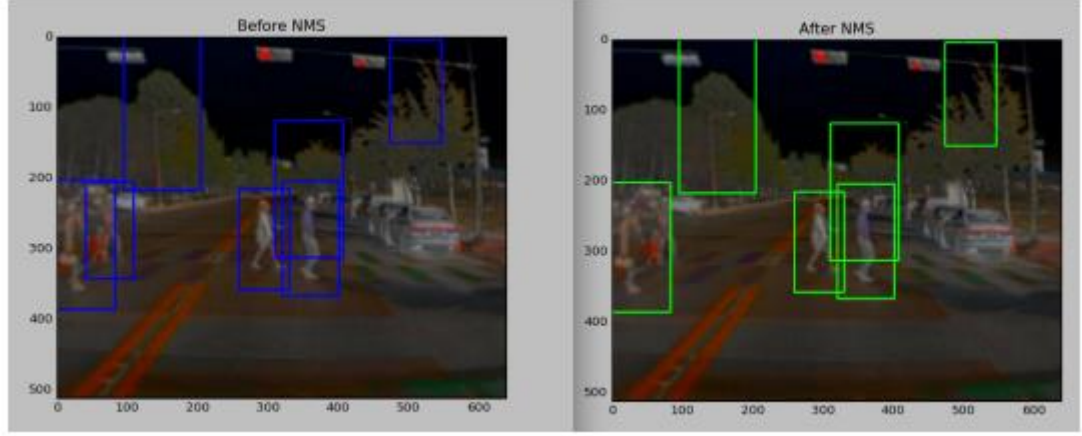
cihazın tek sahibi olarak belirler ve başka birinin erişimini engeller. Mantığı anlık olarak yüzü tarar kayıtlı veri tabanındaki verilerle eşler ve telefonun kilidini açar. Bunu milisaniye sürelerinde gerçekleştirir. Kısacası yüzü kayıt eder ve telefonu açmak istediğinizde telefonun kamera ve çeşitli sensörlerle telefon karşısındaki kişinin telefon sahibi olup olmadığını doğrular. Yani en basit şekilde anlatmak gerekirse Yüzü tespiti > tanılama > verileri elde etme > verileri veri tabanındaki veriler ile karşılaştırma > eşleşme.

2. KULLANILAN TEKNOLOJİLER

Genel olarak üst kısımda yüz tanıma sistemleri kısmında verilen bilgiler ile yüz tanıma, tespit edilen yüzün veri tabanında kayıtlı olan yüz veya yüzler ile eşleşmesi sonucu, sorgulanan yüzün kimliğini doğrulamak veya bu yüzün kimliğini tespit etmektir. Kısacası birebir karşılaştırma yapılır. [7].

2.1 Görüntü İşleme

Görüntü işleme ölçülmüş veya kaydedilmiş olan elektronik görüntü verilerini, elektronik ortamda amaca uygun şekilde değiştirmeye yönelik yapılan bilgisayar çalışması Görüntü işleme, verilerin, yakalanıp ölçme ve değerlendirme işleminden sonra, bir diğer aygıtta okunabilir bir biçime dönüştürülmesi ya da bir elektronik ortamdan başka bir elektronik ortama taşınmasına yönelik bir çalışma olan "Sinyal işlemeden" farklı bir işlemdir. Günümüzde işletmelerin çeşitli yönleriyle kullandıkları görüntü işleme sistemleri hızla gelişen teknolojiler arasında yerini almaktadır. Görüntü İşleme, mühendislik ve bilgisayar bilimleri disiplinlerinde de temel araştırma alanını oluşturur. Görüntü işleme temel olarak aşağıdaki üç adımı içerir. [14]



Şekil 2 – Görüntü İşleme Örnek Görsel [23]

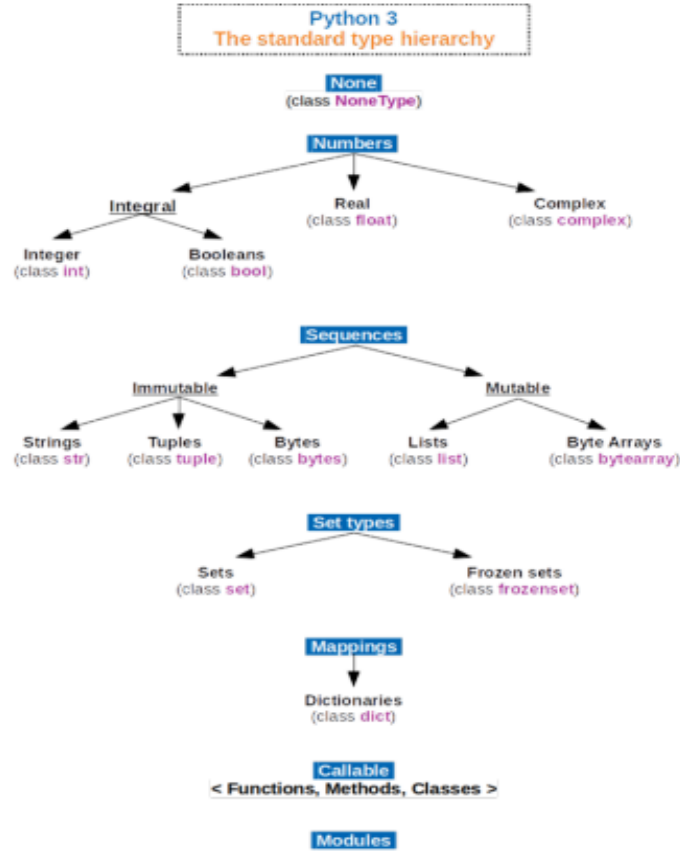
1. Görüntünün optik tarayıcı ile veya dijital fotoğraflarla alınması.
2. Görüntü iyileştirme, verileri sıkıştırma ve elbette uydu fotoğrafları gibi insan gözünü algılayamayan durumlarda lekeleme kalıplarını içeren görüntüyü analiz etme.
3. Çıktı, sonuçların görüntü analizine dayalı olarak değiştirilmiş, kullanıma hazır hale getirmektir.

Görselleştirme görünmesi zor nesneleri gözlemlene görüntü keskinleştirme ve restorasyon gürültülü görüntüleri iyileştirme görüntü alımı ilgi çekici ve yüksek çözünürlüklü görüntü arama desen tanıma bir görüntüdeki çeşitli nesneleri tanımlama. Görüntü tanıma bir görüntüdeki nesneleri ayırt etmektir.[9]

2.2 Python

Python esas olarak nesneye dayalı programlamayı, fonksiyonel programlamayı destekleyen bir programlama dilidir. 1990'lı yıllarda ilk sürümü geliştirilmeye başlanan Python kolaylığı, desteklediği geniş kütüphane ağı ve dinamikliğiyle şuan oldukça popülerleşmiştir.

Sıkça tercih edilen Python, 2000'li yıllardan itibaren yazılımsal gelişmeler ile aynı doğrultuda bir çok kişi tarafından kullanılmıştır. Python yorumlanabilirlik özelliğiyle ve elbette veri yapıları kullandığından kolay öğrenilir, akılda kalır. Yazılımsal açıdan açık kaynak kod yapısıyla elde edilen özgürlük, yazılım akımının yaygınlık kazanması, geliştiricilerin internet üzerinden ve geliştirilen araçlar aracılığıyla beraberlik sağanalarak çalışmaları beraberinde getirmiştir. [13]



Şekil 3 – Dil İşlevinin Gösterimi [5]

2.3 OpenCV Kütüphanesi

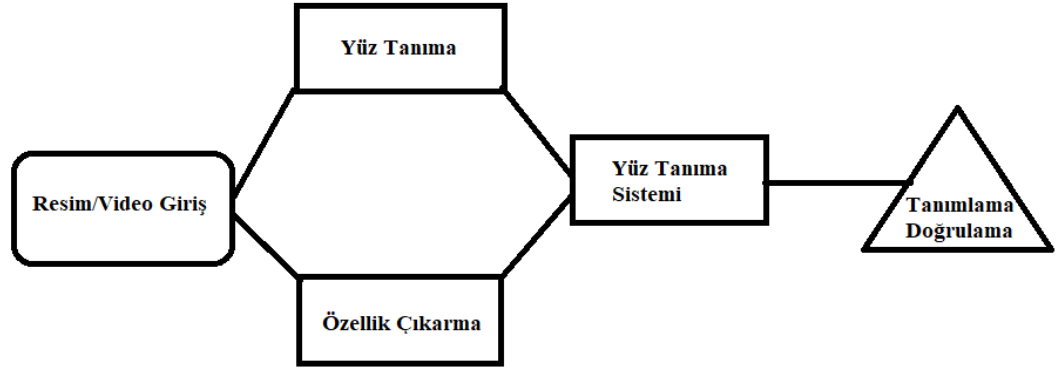
OpenCV, açık kaynak kodlu yapı bulunduran görüntü işleme kütüphanesidir. 1999 yılında İntel tarafından geliştirilmeye başlansada daha sonra Nvidia, AMD, Google gibi şirketlerin desteği ile gelişimi devam etmiştir. İlk sürümü olan OpenCV alfa 2000 yılında çıkmıştır. Açık kaynak kodlu bir kütüphanedir. Elbette BSD lisansına sahip olması kütüphaneyi istediğiniz projede ücretsiz olarak kullanabileceğiniz anlamına gelmektedir. OpenCV bağımsız bir kütüphane olduğu için Windows, Linux, Mac OS ve iOS gibi platformlarında çalışabilmektedir. [13]

2.4 Python Yüz Tanıma

Face recognition yani yüz tanıma sistemi, bir kişiyi dijital bir görüntüden, bir video karesinden tanımlayabilen veya doğrulayabilen bir teknolojidir.

Yüz tanıma sistemlerinin çalıştığı birçok yöntem vardır, ancak genel olarak, verilen görüntüden seçilen yüz özelliklerini bir veritabanındaki yüzlerle karşılaştırarak

alışırlar.



Şekil 4 - Yüz Tanıma Akış Şeması

Ayrıca, yüzlerin farklılığını önemsemenden analiz ederek kişiyi benzersiz bir şekilde tanımlayabilen Biyometrik Yapay Zekaya dayanan bir uygulama gibide düşünülebilir. Bilgisayar uygulaması olduğu için, mobil platformlarda, robotik alanlarda ve diğer teknoloji platformlarında da kullanım görmüştür.

Genel olarak güvenlik sistemlerinde erişim kontrolü için kullanılır ve diğer tanıma sistemleri gibi diğer uygulamalar ile karşılaştırılabilir. Popüler olduktan sonra, pazarlama aracı olarak kullanıldı. Gelişmiş insan-bilgisayar etkileşimi, gözetim, görüntülerin otomatik verilmesi ve veritabanı bulunmaktadır.[20]

2.5 Tensorflow

Tensorflow, Google tarafından yıllar süren bir çalışma sonucunda oluşturulmuş, yapay zekanın derin öğrenme çalışmaları yapan geliştiriciler için üretilmiş, 2015 yılından beri kullanıcıların kullanımına açılmış açık kaynaklı bir kütüphanedir. İlk olarak sadece Python kullanılarak geliştirilmiş olan bu framework, günümüzde Python'un yanında JavaScript, C, C++ gibi akıllara gelebilecek en popüler yazılım dilleri tarafından desteklenmiş durumdadır. Yazılım dillerinin bu desteği TensorFlow'u herhangi bir yazılım diline hakim olan yazılım geliştiricilerin rahat bir şekilde kullanmasına olanak sağlamaktadır. TensorFlow'un fazlaca biliniyor ve kullanılıyor olmasının en temel sebebi de budur.[18,19]

3. YÜZ TANIMA NEREDE KULLANILIR

3.1 Şahsi Telefonun Kilidini Açma

En gelişmiş akıllı cihazlar da dahil olmak üzere birçok telefon, tabletin kilidini açmak için yüz tanıma teknolojisini kullanır. Bu teknoloji, insanlara kişisel verilerini koruyabilmek adına güçlü bir yöntem sunar ve akıllı cihazlarımızın çalınması durumunda özel kişisel verilere erişilebilmesine engel olur. Yüz tanıma teknolojisini kullanan en ünlü firmalar, rastgele bir yüzün kişisel akıllı cihazlarımızın kilidini açma olasılığının milyonda bir ihtimal olduğunu garanti etmektedir. [17]

3.2 Emniyet Kurumlarında Suçlu Tespiti

Yüz tanıma, emniyet güçleri tarafından sık sık kullanılır. Bazı raporlara göre teknoloji, Amerika'daki emniyet güçleri daha fazla kullanılmaktadır. Bu, diğer ülkeler için de geçerlidir. Polisler, sabıka fotoğraflarını toplar ve veritabanlarıyla karşılaştırır. Suçlunun fotoğrafı çekildiğinde, polisin daha sonra kullanması için veritabanlarına eklenir. [16]

Mobil yüz tanıma sayesinde polisler, akıllı telefon veya diğer taşınabilir teknolojik cihazları kullanarak insanların fotoğrafını çekebilir ve bu fotoğrafı kimlik tespiti için yüz tanıma veritabanıyla anında karşılaştırabilir.



Şekil 3 - Sınır Denetiminde Kullanılan Örnek Yüz Tanıma Sistemi [9]

3.3 Sınır Denetimi

Yüz tanıma teknolojisi, neredeyse hayatımızın her alanında görmeye alıştığımız bir yenilik oldu. Bu teknolojinin havaalanlarında sağladığı kolaylık ise göz ardı edilemez. Gün geçtikçe daha fazla sayıda insan biyometrik pasaport kullanıyor ve havaalanında işlemlerini daha da hızlandırmak adına, fazlaca uzun olan pasaport kontrolü sırasını atlayıp, otomatik e-pasaport denetiminden geçiyor. Yüz tanıma, bekleme sürelerini minimuma indirmenin yanı sıra havaalanlarının güvenliğinin artmasına da imkan sağlıyor. ABD İç Güvenlik Bakanlığı, yüz tanıma teknolojisinin gelecek yıldan itibaren yolcuların %97'sinde kullanılacağını ön görüyor. Bu teknoloji havaalanları ve sınır geçişlerinin yanı sıra, olimpiyatlar gibi yüksek oranda katılım sağlanan etkinliklerde yüksek güvenilirlik sağlamak için de kullanılıyor.[17]

3.4 İnsan Kaçakçılığını Engelleme

Yüz tanıma teknolojisi, bulunamayan insanları ve insan kaçakçılığı mağdurlarını bulmak için de kullanılmaktadır. Bu kişiler veritabanına eklendiğinde, kamuya açık alanlarda (havaalanı, mağaza vs.) yüz tanıma tarafından tespit edildiği anda emniye güçlerine bildirilebilir. [11]

3.5 Hırsızlık Gibi Suçlarının Azaltılması

Yüz tanıma; dolandırıcılık geçmişine sahip kişiler mağazalara girdiğinde onları tanımak için de kullanılabilir. Hırsızlık potansiyeli oluşturan kişiler mağazaya geldiğinde suçlu veritabanlarıyla eşleşiyorsa; kayıp önleme ya da güvenlik personeli yetkilileri haberdar edebilir.

3.6 Ödeme İşlemlerinin İyileştirilmesi

Yüz tanıma teknolojisi, müşterilerin deneyimlerinin iyileştirilmesi için yüksek imkanlar sunar. Örneğin mağazalardaki danışma ekranları müşterileri tanıyarak satın alma geçmişlerinin göz önünde bulundurarak yeni ürün önerilerinde bulunabilir ya da müşterilere ürünün yerlerini gösterebilir. "Yüzle ödeme" teknolojisi ise, müşterilerin oldukça yavaş ödeme yöntemlerini kullanıldığı mağazalardaki uzun kasa kuyruklarına maruz kalmamasına imkan tanıyabilir.[16]

3.7 Bankacılık İşlemleri

Biyometrik çevrimiçi bankacılık, yüz tanımanın önemli faydalarından biridir. Müşteriler, tek seferlik parolalarla vakit kaybetmek yerine akıllı cihazlarına bakarak işlemleri onaylayabilir. Yüz tanıma kullanıldığı için art niyetli kişilerin ele geçirebileceği bir parola olmaz. Kötü kişilerin galeri arşivinizi erişip ele geçirdiği takdirde, kaynağının kendisi mi yoksa sahte bir temsil mi olduğunu tespit etmek için kullanılan canlılık algılama tekniği, fotoğrafları sizi taklit etme amacıyla kullanılmasını (teoride) engeller. Yüz tanıma teknolojisi, banka kartlarını ve imzaları çok geride bırakabilir.[19]

3.8 Pazarlama

Yüz tanıma, pazarlamacılar tüketici deneyimlerini geliştirmek için kullanılmıştır. Medya şirketleri, kullanıcıların film ve TV bölümlerindeki olaylara, karakterlere tepkisini ölçmek için yüz tanımayı kullanır. [7]

3.9 Sağlık Alanları

Hastanelerde hastalar için yüz tanımayı kullanır. Sağlıkçılar; hasta kayıtlarına erişim, kayıtları kolaylaştırmak, hatta genetik hastalıkların teşhisinde yardımcı olması için yüz tanıma teknolojisini test etmektedir. AiCure, hastaların ilaçlarını reçeteye uygun şekilde aldığından emin olmak için yüz tanımayı kullanan bir uygulama geliştirdi. Biyometrik teknolojinin ucuzladıkça sağlık sektöründe daha fazla uygulamaya uyarlanması bekleniyor.[14]

3.10 Eğitim Alanları

Öğrencilerin dersleri asmasına engel olmak ve emin olmak için Çin'de eğitim kurumlarında yüz tanıma bulunur. Kullanılan tabletlerle öğrencilerin yüzleri taranır ve veritabanıyla eşleşme sonucu kimlik doğrulama yapılır. Daha geniş çaplı olarak yüz tanıma teknolojisi, işverenlerin çalışan devamsızlığını takip edebilmesi için çalışanların iş yerine giriş ve çıkışında kullanılabilir.

3.11 Trafik

Otomobil şirketleri, yüz tanıma teknolojisine oldukça ilgi duyar çünkü araba anahtarlarının yerini alacağı konuşuluyor.[13]

3.12 Kumar

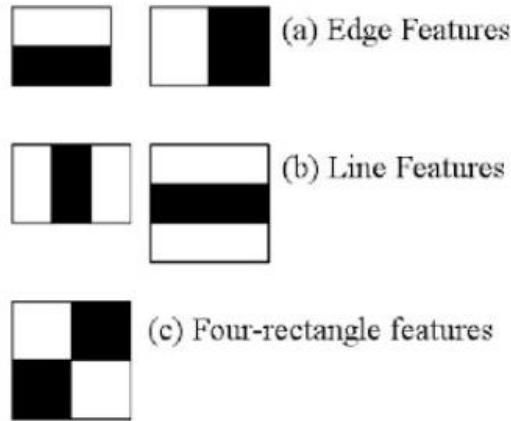
Kumarhaneler gibi büyük, kalabalık alanlarda kumar alanlarına giren ve etrafta gezinen müşterileri takip etmek insan personel için zor bir iştir. Bu sektörde bile faydası vardır.

4. TASARIM SÜRECİ

4.1 Kütüphane Seçimi

OpenCV HAAR Kütüphanesi İlk olarak 2001 yılında yayınlanan, Boosted Cascade of Simple Features kullanılarak Rapid Object Detection'da tanıtılan Haar basamakları, biranda OpenCV'nin en popüler nesne algılama algoritması haline gelmiştir.

Elbette, birçok algoritma Haar basamaklarından (HOG + Linear SVM... birkaç isim) daha etkili veya daha doğru sonuçlar elde edebilsede, Haar bugün hala teknolojiyle içiçi alakalı ve kullanışlıdır. Haar faydalarından biri, çok hızlı olmasıdır ve hızlarına yetişmek cidden zordur.[6]



Şekil 5 – Haar Pikselleri Görseli [3]

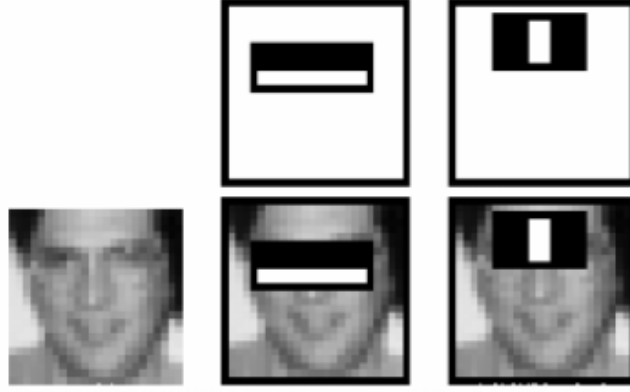
Tabiki her iyi şeyin bir dezavantajı bulunur, yanlış algılamalara eğilimli olmaları, çıkarım için kullanıldıklarında parametre ayarlaması gerektirirler ve genel olarak

modern algoritmalar kadar doğru olamazlar.

Neden Haar:

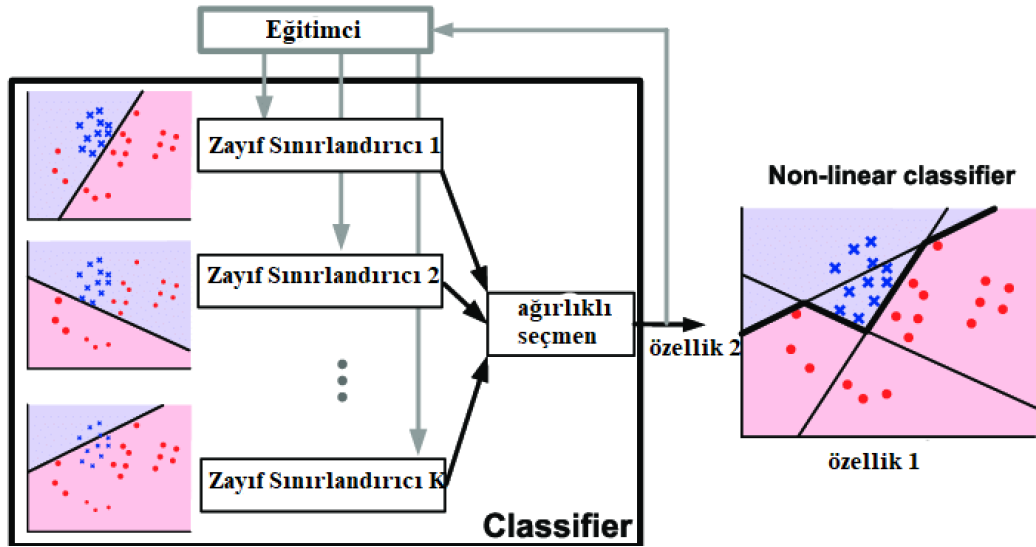
1. Görüntü işleme literatürünün önemli bir parçasıdır.
2. Hala OpenCV ile kullanılıyor.
3. Kaynak kısıtlı cihazlarda, maliyeti düşürür.

Aşağıda bulunan resim üst sıra iki iyi özelliği gösterir. Seçilen ilk özellik, göz bölgesinin genellikle burun bölgesinden daha koyu olması bakıyor gibi bir sonuç çıkıyor olsada seçilen ikinci özellik, gözlerin burun bölgesinden daha koyu olması özelliğine dayanır. Ancak yanaklara veya başka herhangi bir yere uygulanan aynı pencerelerin önemi yoktur. **Adaboost** tarafından elde edilir.[7]



Şekil 6 – Haar İle Seçilen Göz İçin Adımlar [1]

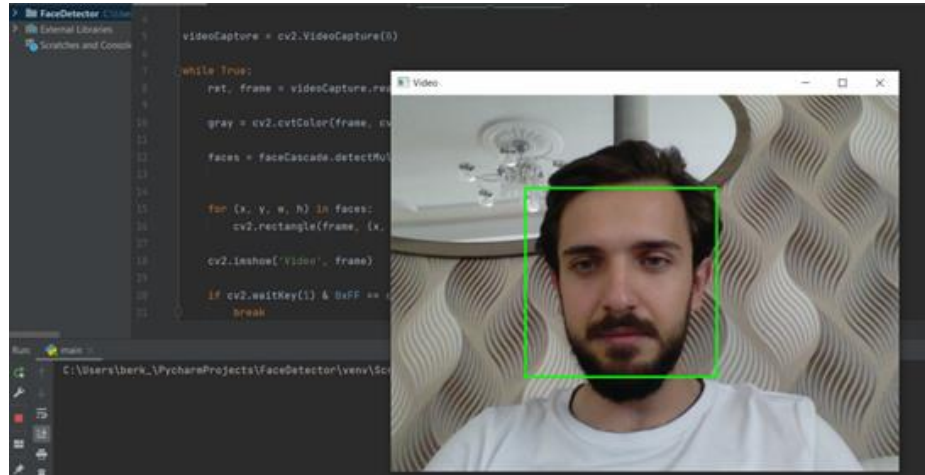
AdaBoost, ikili sınıflandırıcıların verimliliğini artırmak için oluşturulmuş bir meta öğrenme olarak da bilinir. AdaBoost, zayıf sınıflandırıcıların hatalarından ders çıkarmak ve onları güçlü olanlara dönüştürmek için yinelemeli bir yaklaşım kullanır.



Şekil 7 – AdaBoost Algoritması Çalışma Mantığı Gösterimi [9]

4.2 Kodlama

Python ile yazıldı. Kullanımı oldukça sık olan Haar-cascade sınıflandırma algoritması kullanıldı ve biçimlendirildi. Gereksinim duyulan zayıf sınıflandırıcıların güçlendirilmesi için AdaBoost algoritması eklendi. Kodlama sistemin hızlı çalışması ve fazla zorlanma yaşanmaması için minimum düzeyde tutuldu.



Şekil 8 – Uygulamanın Anlık Yüz Tespiti Sağlamasına Ait Görsel

5. UYGULAMADA KULLANILAN ÖNEMLİ FONKSİYONLAR

5.1 Yüz Bölgesinin Tespiti

```
faceCascade = cv2.CascadeClassifier("bayraktargils_library.xml")
```

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

biçimlendirilmiş onlarca hata çözülmüştür sistem bu sayede çok hızlı çalışmaktadır.

5.2 Yüz Tanıma

import cv2 *#dosyayı adresleriz.*

faceCascade = cv2.CascadeClassifier("haarcascade_frontalface_default.xml") *#anlık olarak video kaydı yapılır.*

videoCapture = cv2.VideoCapture(0) *#video karelerden framelerden oluşur, yeni frame oluştuğu sürece gerçekleşecek koda ihtiyaç var while kullanılır.*

while True:

ret, frame = videoCapture.read() *#bu kod şu şekilde çalışır, görüntüyü griye çevirip öyle yüzü algılıyor ama görünen görüntü gri olmayacaktır.*

gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY) *#yüzleri tanımlamak gerekiyor.*

faces = faceCascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5, minSize=(30, 30),

flags=cv2.CASCADE_SCALE_IMAGE) *#yüzleri nasıl görürüz? etrafına dikdörtgen oluşturmalyız.*

for (x, y, w, h) in faces:

cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)

cv2.imshow('Video', frame)

#peki bu uygulamadan nasıl çıkılmalı. while döngümüz var. bir tuş atamak gerek.

if cv2.waitKey(1) & 0xFF == ord('q'):

break *# 'ord' q nun harf kodunu belirler ascii koduna çevirir.*

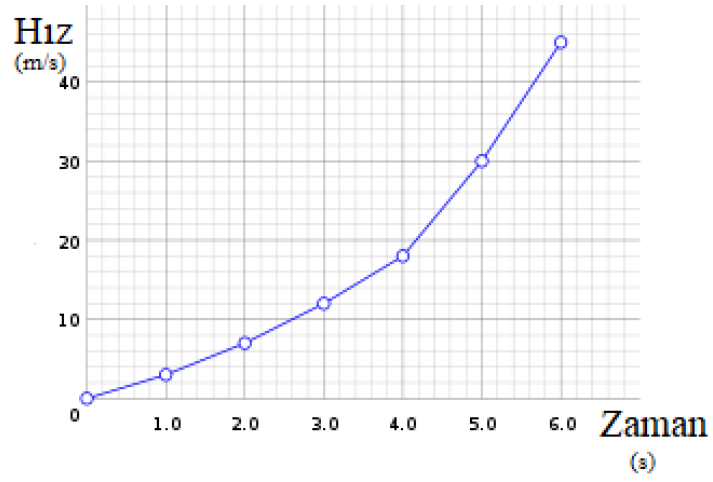
videoCapture.release()

cv2.destroyAllWindows()

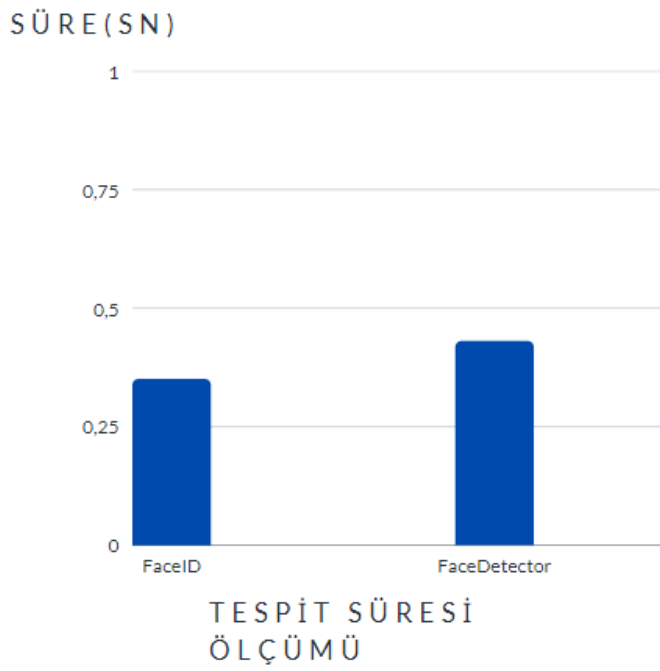
6. SONUÇLAR VE ÖNERİLER

Yüz tanıma teknolojisi elbette telefon kilidinin açılmasından ibaret değildir bir çok alanda kullanıldığı zaten biliniyor. Uluslararası düzeyde düşünürsek, teröristlerin ülkeler arası giriş çıkış ülke içi dolaşımları veya farklı suçluların tespitinde kullanılabilir. Yani telefonun kilidini açmak için çıkılan yolda devasa boyutta faydaları olduğu ortaya çıkıyor. Elbette bunun için hız ve tespit oranının yüksek olması anında müdahale için şarttır. Bu sayede daha düşük suç oranları elde edilebilir. Kurumsal firmalarda, kişisel bilgisayarlara erişmek için parola girmek yerine yüz tanıma teknolojisini kullanabilir. Herkesin başına gelebilen polis çevirmeleri, elbette gerekli olsada sebepsiz yere çevirme hatta aramaya ilişkin sıkıntılar doğurabiliyor. Yüz tanıma teknolojisi bu süreci iyileştirebilir. Müşteriler ödeme gerçekleştirmeye gerek duymadan kredi kartı veya nakit çıkarmak yerine alışveriş merkezlerinde yüzleriyle ödeme yapabilir. Yüzün tanınması bir saniye hatta daha kısada sürer, bu da bir çok konuda, örneğin tanımayı kullanan firmalar için avantajlar sunar. Siber saldırıların önüne geçebilecek kadar güvenlik sağlar. İnsanlar, yüz tanıma teknolojisi kullanımının kitlesel izlemenin veri kayıtlarının ve bu verilere göre sunulan reklam veya kişilik analizlerinin, oluşturulan profillerinin ve bu bilgilerinin dağıtımının farkındalar ve bireysel özgürlüğü kısıtlayabileceğinden endişelenmektedirler. Sıran insanları izlerken hükümetler kimin suçlu olduğunu bilemeden aslında suçlu kapasitesinde olan insanları da takibe aldığı için suçların önüne geçmede oldukça ileri gitmişlerdir. Çin en büyük örneğidir. Unutulmaması gereken önemli bir husus olan yüz tanıma verileri hatasız değildir asla %100 güven söz konusu olamaz elbette bu nedenle kişilerin işlemedikleri suçlar nedeniyle suçlanmasına hatta yargılanmasına yol açabilir. Etik ve gizlilik, en çok tartışmaya neden olan konudur. Yüz tanıma yazılımı, devasa verileri "öğrenmesi" gereken makine öğrenimi verilerini depolar. Bu tür büyük veri kümeleri için üst düzey veri depolama alanları gerekir. Bu maliyeti karşılamaya malesef her kurumun gücü yetmemektedir.

Uygulamada tespit sırasında kayıplar yaşansada, yüz tespitinin hızı oldukça yukarıda tutuldu. Anlık olarak yüz tespiti %82 oranda ölçüldü. Anlık kamera karşısında yüz tespiti gerçekleşmesi yanı sıra tespit sırasında ki hızı 0.43 saniye olarak ölçüldü. Faydalarıyla zararlarıyla ortada bulunan bu teknolojiye hıza odaklanıldı. FaceID ile karşılaştırmalar yapıldı.

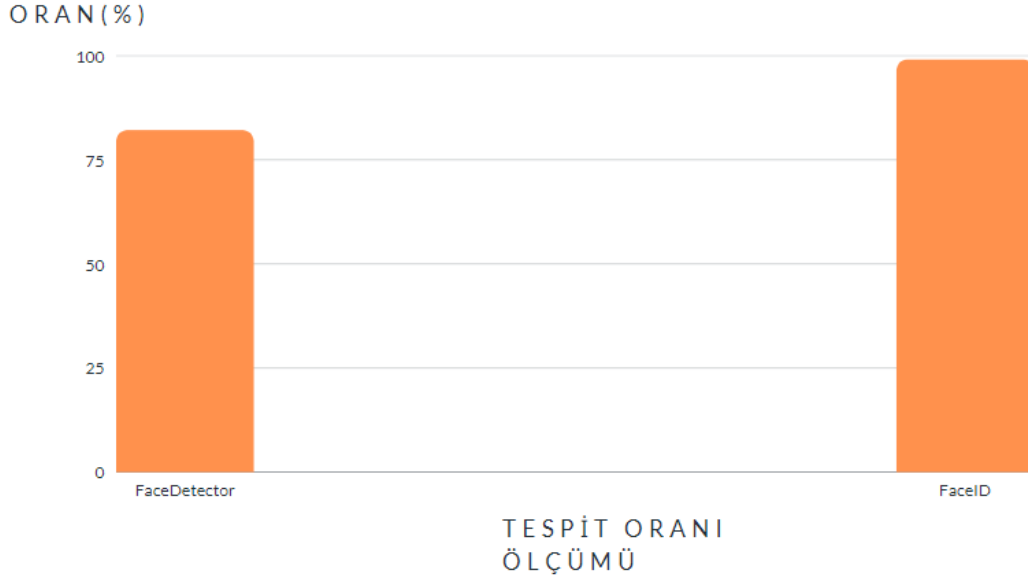


Çizelge 1 – Uygulamanın Anlık Yüz Tespiti Hızının Ölçüldüğü Çizelge
İlk aşamada ciddi zaman kaybı ve tespitte bozukluklar yaşansada daha sonra alınan onca hata
çözülmüş ve ideal oranlar elde edilmiştir.



Çizelge 2 – FaceID ve Uygulamanın Tespit Süreleri

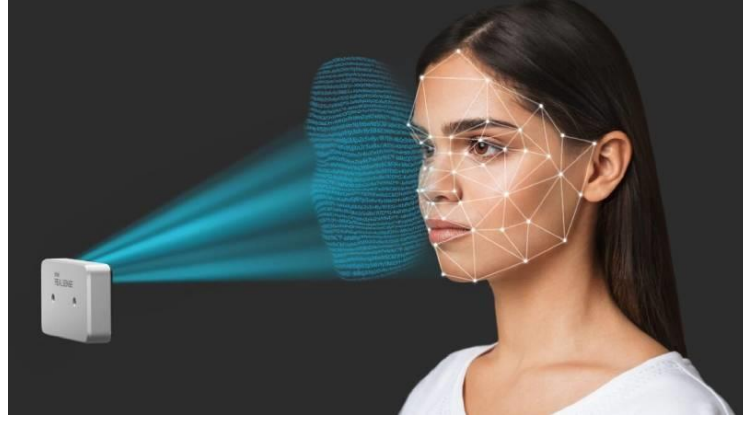
En iyi tespit süreleri ele alındığında FaceID 37ms sürede tespit yapmıştır. Uygulamamız ise 43 ms sürede tespit sağlamıştır. Bu sonuç oldukça tatmin edicidir 6ms farkla uygulama geride kalmıştır.



Çizelge 3 – FaceID ve Uygulamanın Tespit Oranı

Malesef tespit oranında ciddi fark gözlemlenmiştir. Uygulamada hıza odaklanıldığı için hız faktörü önde tutulduğu için buna bağlı olarak tespit hızı karşı yenik düşmüştür.FaceID %99 oran ile yüzleri tespit ettiği halde uygulamamız %82 oranla oldukça geride kalmıştır.

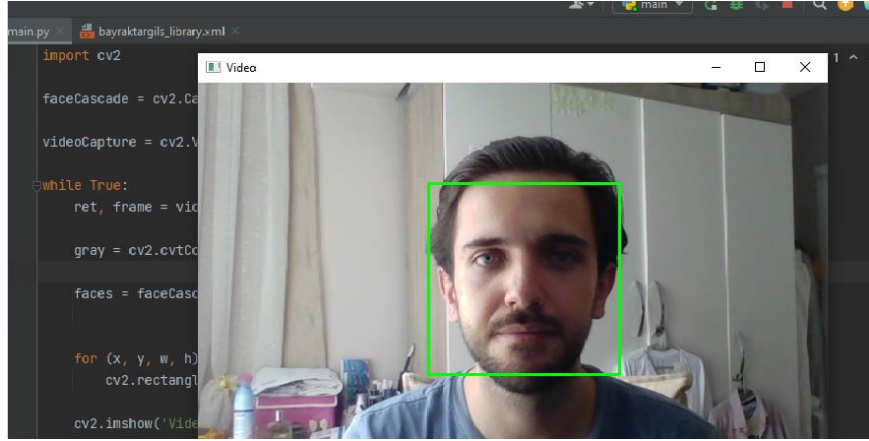
Uygulama geliştirilmeye açıktır. Kullanılabileceği alanlar oldukça fazladır. Tespit oranları ortalama esas alınmıştır uygulamanın en iyi sonucu %95 oranla tespit etmektedir. Ancak uygun koşullar sağlanmadığında (düşük ışık, karanlık ortam, yüzde bulunan yara bazlı yanlış tespitler vb.) oran oldukça düşmektedir. Bu dezavantaja nazaran tespit hızının tatmini bu dezavantajı absorbe etmektedir.



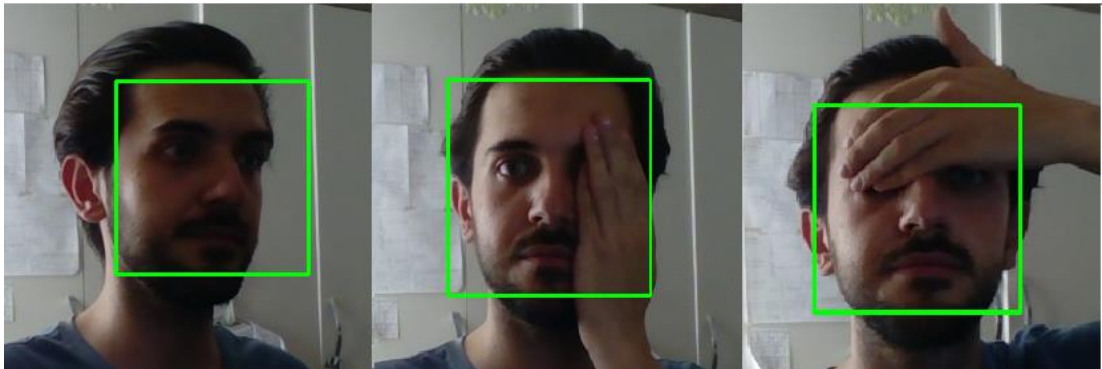
Şekil 9 – Yüz Tespiti Sağlayan Güvenlik Cihazlarına Örnek Görsel [19]

Uygulamaya ait test görüntüleri:

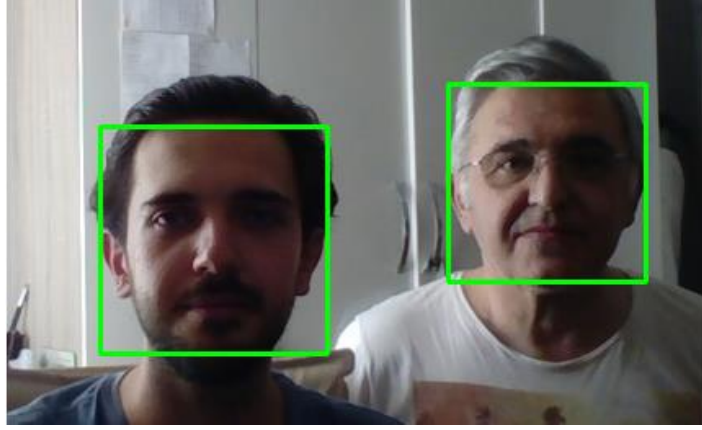
Düşük ışık, arka plan uygunsuzluğu, yüz tanımlamaya engel unsurlar olmasına rağmen tespit sağlanmıştır.



Şekil 10 – Uygulama Çıktısı Örnek



Şekil 11 – Düşük Işıktaki Uygulama Çıktısı Örnek



Şekil 12 – Düşük Işıktaki Farklı Yüz Tespiti Uygulama Çıktısı Örnek

7. KAYNAKLAR

- [1] YILMAZ, Okan, Hakan AYDIN, and Ali Çetinkaya. "Faster R-CNN Evrişimsel Sinir Ağı Üzerinde Geliştirilen Modelin Derin Öğrenme Yöntemleri ile Doğruluk Tahmini ve Analizi: Nesne Tespiti Uygulaması." *Avrupa Bilim ve Teknoloji Dergisi* 20 (2020): 783-795.
- [2] Oral, A. Zeynep. "Türk işaret dili çevirisi." (2016).
- [3] Tıǧlı, Celalettin. "Görme özürlüler için GPS-uzaysal veritabanı-kamera destekli doğrultu belirleme sistemi." (2007).
- [4] ALGUR, Özlem, Vedat TÜMEN, and Özal YILDIRIM. "Dış Ortam Görüntülerindeki İnsan Hareketlerinin Hibrit Derin Öğrenme Yöntemleri Kullanarak Sınıflandırılması." *Fırat Üniversitesi Mühendislik Bilimleri Dergisi* 30.3 (2018): 121-129.
- [5] Kemalolu, YUSUF "Engellilik, Kulak Burun Boğaz (KBB) hekimliğı ve işaret dili." *Bozok Tıp Dergisi* 1.1 (2014): 38-53.
- [6] Azhar Murzaeva, Hasibe Panik, Zeki Abay ve Esra Yorulmaz. "Yazı ve sesi işaret diline çeviren bir mobil uygulama(Seslik)". *Ondokuz Mayıs Üniversitesi*
- [7] Hasan YAKUT. "İşaret Dili Harflerinin Görüntü İşleme Yöntemleriyle Tanınması İçin Bir Uygulama". *FIRAT ÜNİVERSİTESİ*, (2013).
- [8] Polat Timur, Rukiye. Derin öğrenme yöntemi ile köpek davranışlarının analizi ve sınıflandırılması. MS thesis. Bilecik ,Seyh Edebalı Üniversitesi, Fen Bilimleri Enstitüsü, 2019.
- [9] Savaş, Osman Oğuzhan, and Tülay Yildirim. "The Effect of Convolutional Neural ~ Network Parameters on Sign Language Recognition." *2018 Innovations in Intelligent Systems and Applications Conference (ASYU)*. IEEE, 2018.
- [10] Kın, Zeren Berna. Türk işaret dili alfabesinin derin öğrenme yöntemi ile sınıflandırılması. MS thesis. Başkent Üniversitesi Fen Bilimleri Enstitüsü, 2019.
- [11] Jupyter Notebook, <https://farukeryilmaz.com/jupyter-notebook-nedirnasil-kullanilir-veri-bilimi/ZiyaretTarihi:19Haziran2022>
- [12] Behzat Bükün ve Ali Ekinci. "Türk İşaret Dili Tanıma". *Yıldız Teknik Üniversitesi*, 2005
- [13] Halil, Abang Irfan Abang Abdul, and ROSDIYANA SAMAD. Hand Gesture Recognition System Using Image Processing. Diss. KUKTEM, (2007).

- [14] Görüntü İşleme, <https://tr.wikipedia.org/wiki> Ziyaret Tarihi: 20 Ocak 2021
- [15] Python, <https://www.python.com/> Ziyaret Tarihi: 20 Haziran 2022
- [16] OPENCV, <https://opencv.org/> Ziyaret Tarihi: 10 Haziran 2022
- [17] Çelik, Özer, and Alper ODABAS. "Sign2Text: Konvolüsyonel Sinir Ağları Kullanarak Türk İşaret Dili Tanıma." Avrupa Bilim ve Teknoloji Dergisi 19 (2020): 923-934.
- [18] TensorFlow Nesne Algılama, <https://farukeryilmaz.com/jupyter-notebook-nedir-nasil-kullanilir-veri-bilimi/> Ziyaret Tarihi: 21 Haziran 2022
- [19] TensorFlow, <https://medium.com/@mbektas/tensorflow-ortam> Ziyaret Tarihi: 22 Temmuz 2022
- [20] CuDNN, <http://yapayzekalabs.blogspot.com/2018/07/ubuntu-1604-icin-cudnn-kurulum.html> Ziyaret Tarihi: 18 Temmuz 2022
- [21] Albumentations, https://albumentations.ai/docs/getting_started/bounding_boxes_augmentation/ Ziyaret Tarihi: 11 Temmuz 2022
- [22] Tensorflow, https://www.tensorflow.org/api_docs/python/tf/data/Dataset#batchZ
- [23] Tensorflow, https://www.tensorflow.org/api_docs/python/tf/data/Dataset#batchZ

8. EKLER

Taniyici.py

```
import cv2
import numpy as np
import os

recognizer = cv2.face.LBPHFaceRecognizer_create()
recognizer.read('trainer/trainer.yml')
faceCascade =
cv2.CascadeClassifier("Cascade/haarcascade_frontalface_default.xml
")
font = cv2.FONT_HERSHEY_SIMPLEX
id = 0
names = ['None', 'samed', 'rabia']
cam = cv2.VideoCapture(0)
cam.set(3, 640)
cam.set(4, 480)
minW = 0.1*cam.get(3)
minH = 0.1*cam.get(4)
while True:
    ret, img =cam.read()
    img = cv2.flip(img, 1)
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    faces =
    faceCascade.detectMultiScale(gray,scaleFactor=1.2,minNeighbors=5,
    minSize=(int(minW), int(minH)),)
    for (x, y, w, h) in faces:
        """"color = img[y:y + h, x:x + w]""""
        cv2.rectangle(img, (x, y), (x + w, y + h), (0, 255, 0), 2)
        id, confidence = recognizer.predict(gray[y:y + h, x:x + w])
        if (confidence < 100):
            id = names[id]
            confidence = " {0}%".format(round(100 - confidence))
```



```

else:
id = "bilinmiyor"
confidence = " {0}%".format(round(100 - confidence))
cv2.putText(img, str(id), (x + 5, y - 5), font, 1, (255, 255, 255), 2)
cv2.putText(img, str(confidence), (x + 5, y + h - 5), font, 1, (255, 255,
0), 1)
"""img_item = "my_image.jpg"
cv2.imwrite(img_item,color)"""
cv2.imshow('camera', img)
k = cv2.waitKey(10) & 0xff
if k == 27:
break
cam.release()
cv2.destroyAllWindows()

```

ToplananVeriler.py

```

import cv2
import zilches
cam = cv2.VideoCapture( 0)
=cv2.CascadeClassifier("Cascade/haarcascade_frontalface_default.xml")
face_id = input(' n bir isim giriniz == >')
print(" n( Bilgi) Yüz yakalama başlatılıyor. Kamerası bak ve bekle.")
count = 0
while True
ret, img = cam.read()
img = cv2.flip( img, 1)
gray = cv2.cvtColor( img,cv2.COLOR_BGR2GRAY)
faces = face_detector. detectMultiScale( argentine,1.3, 5)
for( x, y, w, h) in faces
cv2.rectangle( img,( x, y),( x w, y h),( 255, 0, 0), 2)
count = 1
(" data stoner." str(face_id)'. ' str( count)". jpg", argentine( yy h, xx w))
cv2.imshow(' image', img)

```

```

k = cv2.waitKey( 100) & 0xff
if k == 27
break
elif count>= 30
break
()
()

```

OgrenilenVeriler.py

```

import cv2
import numpy as np
from PIL import Image
import os
path = 'data'
recognizer = cv2.face.LBPHFaceRecognizer_create()
detector =
cv2.CascadeClassifier("Cascade/haarcascade_frontalface_default.xml
")
def getImagesAndLabels(path):
imagePaths = [os.path.join(path,f) for f in os.listdir(path)]
faceSamples = []
ids = []
for imagePath in imagePaths:
PIL_img = Image.open(imagePath).convert('L')
img_numpy = np.array(PIL_img,'uint8')
id = int(os.path.split(imagePath)[-1].split(".")[1])
faces = detector.detectMultiScale(img_numpy)
for (x,y,w,h) in faces:
faceSamples.append(img_numpy[y:y+h,x:x+w])
ids.append(id)
return faceSamples,ids
print("\n Yüzler taranıyor.Birkaç saniye sürecek bekle...")
faces, ids = getImagesAndLabels(path)
recognizer.train(faces,np.array(ids))

```

```
recognizer.write('trainer/trainer.yml')
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Main.py

```
import cv2
faceCascade = cv2.CascadeClassifier("bayraktargils_library.xml")
videoCapture = cv2.VideoCapture( 0)
while True
ret, frame = videoCapture.read()
gray = cv2.cvtColor( frame,cv2.COLOR_BGR2GRAY)
faces = faceCascade.detectMultiScale( argentine, scaleFactor = 1.1,
minNeighbors = 5, minSize = ( 30, 30),
flags = cv2.CASCADE_SCALE_IMAGE)
for( x, y, w, h) in faces
cv2.rectangle( frame,( x, y),( x w, y h),( 0, 255, 0), 2)
(' Video', frame)
ifcv2.waitKey( 1) & 0xFF == ord(' q')
break
()
()
```

bayraktargils_library.xml

```
<?xml version="1.0"?>
<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></_>
      <_>
        <internalNodes>
          0 -1 2 2.1927999332547188e-02</internalNodes>
        <leafValues>
          -1.5105249881744385e+00
1.0625729560852051e+00</leafValues></_>
      <_>
        <internalNodes>
          0 -1 3 5.7529998011887074e-03</internalNodes>
        <leafValues>
          -8.7463897466659546e-01
1.1760339736938477e+00</leafValues></_>
      <_>
        <internalNodes>
          0 -1 4 1.5014000236988068e-02</internalNodes>
        <leafValues>

```

```

-7.7945697307586670e-01
1.2608419656753540e+00</leafValues></_>
<_>
<internalNodes>
0 -1 5 9.9371001124382019e-02</internalNodes>
<leafValues>
5.5751299858093262e-01 -
1.8743000030517578e+00</leafValues></_>
<_>
<internalNodes>
0 -1 6 2.7340000960975885e-03</internalNodes>
<leafValues>
-1.6911929845809937e+00 4.4009700417518616e-
01</leafValues></_>
<_>
<internalNodes>
0 -1 7 -1.8859000876545906e-02</internalNodes>
<leafValues>
-1.4769539833068848e+00 4.4350099563598633e-
01</leafValues></_>
<_>
<internalNodes>
0 -1 8 5.9739998541772366e-03</internalNodes>
<leafValues>
-8.5909199714660645e-01 8.5255599021911621e-
01</leafValues></_></weakClassifiers></_>
<_>
<maxWeakCount>16</maxWeakCount>
<stageThreshold>-4.9842400550842285e+00</stageThreshold>
<weakClassifiers>
<_>
<internalNodes>
0 -1 9 -2.1110000088810921e-02</internalNodes>
<leafValues>

```

1.2435649633407593e+00 -
 1.5713009834289551e+00</leafValues></_>
 <_>
 <internalNodes>
 0 -1 10 2.0355999469757080e-02</internalNodes>
 <leafValues>
 -1.6204780340194702e+00
 1.1817760467529297e+00</leafValues></_>
 <_>
 <internalNodes>
 0 -1 11 2.1308999508619308e-02</internalNodes>
 <leafValues>
 -1.9415930509567261e+00 7.0069098472595215e-
 01</leafValues></_>
 <_>
 <internalNodes>
 0 -1 12 9.1660000383853912e-02</internalNodes>
 <leafValues>
 -5.5670100450515747e-01
 1.7284419536590576e+00</leafValues></_>
 <_>
 <internalNodes>
 0 -1 13 3.6288000643253326e-02</internalNodes>
 <leafValues>
 2.6763799786567688e-01 -
 2.1831810474395752e+00</leafValues></_>
 <_>
 <internalNodes>
 0 -1 14 -1.9109999760985374e-02</internalNodes>
 <leafValues>
 -2.6730210781097412e+00 4.5670801401138306e-
 01</leafValues></_>
 <_>
 <internalNodes>

```

    0 -1 15 8.2539999857544899e-03</internalNodes>
    <leafValues>
    -1.0852910280227661e+00 5.3564202785491943e-
01</leafValues></_>
  <_>
    <internalNodes>
    0 -1 16 1.8355000764131546e-02</internalNodes>
    <leafValues>
    -3.5200199484825134e-01 9.3339198827743530e-
01</leafValues></_>
  <_>
    <internalNodes>
    0 -1 17 -7.0569999516010284e-03</internalNodes>
    <leafValues>
    9.2782098054885864e-01 -6.6349899768829346e-
01</leafValues></_>
  <_>
    <internalNodes>
    0 -1 18 -9.8770000040531158e-03</internalNodes>
    <leafValues>
    1.1577470302581787e+00 -2.9774799942970276e-
01</leafValues></_>
  <_>
    <internalNodes>
    0 -1 19 1.5814000740647316e-02</internalNodes>
    <leafValues>
    -4.1960600018501282e-01
1.3576040267944336e+00</leafValues></_>
  <_>
    <internalNodes>
    0 -1 48 -7.8795999288558960e-02</internalNodes>
    <leafValues>
    -1.7491459846496582e+00 2.8475299477577209e-
01</leafValues></_>

```

```

<_>
  <internalNodes>
    0 -1 49 2.1110000088810921e-03</internalNodes>
  <leafValues>
    -9.3908101320266724e-01 2.3205199837684631e-
01</leafValues></_>
<_>
  <internalNodes>
    0 -1 50 2.7091000229120255e-02</internalNodes>
  <leafValues>
    -5.2664000540971756e-02
1.0756820440292358e+00</leafValues></_>
<_>
  <internalNodes>
    0 -1 51 -4.4964998960494995e-02</internalNodes>
  <leafValues>
    -1.8294479846954346e+00 9.9561996757984161e-
02</leafValues></_></weakClassifiers></_>
<_>
  <maxWeakCount>32</maxWeakCount>
  <stageThreshold>-4.4531588554382324e+00</stageThreshold>
  <weakClassifiers>
    <_>
      <internalNodes>
        0 -1 52 -6.5701000392436981e-02</internalNodes>
      <leafValues>
        1.1558510065078735e+00 -
1.0716359615325928e+00</leafValues></_>
    <_>
      <internalNodes>
        0 -1 53 1.5839999541640282e-02</internalNodes>
      <leafValues>
        -1.5634720325469971e+00 7.6877099275588989e-
01</leafValues></_>

```



```

<_>
  <internalNodes>
    0 -1 54 1.4570899307727814e-01</internalNodes>
  <leafValues>
    -5.7450097799301147e-01
1.3808720111846924e+00</leafValues></_>
<_>
  <internalNodes>
    0 -1 55 6.1389999464154243e-03</internalNodes>
  <leafValues>
    -1.4570560455322266e+00 5.1610302925109863e-
01</leafValues></_>
<_>
  <internalNodes>
    0 -1 56 6.7179999314248562e-03</internalNodes>
  <leafValues>
    -8.3533602952957153e-01 5.8522200584411621e-
01</leafValues></_>
<_>
  <internalNodes>
    0 -1 57 1.8518000841140747e-02</internalNodes>
  <leafValues>
    -3.1312099099159241e-01
1.1696679592132568e+00</leafValues></_>
<_>
  <internalNodes>
    0 -1 58 1.9958000630140305e-02</internalNodes>
  <leafValues>
    -4.3442600965499878e-01 9.5446902513504028e-
01</leafValues></_>
<_>
  <internalNodes>
    0 -1 59 -2.7755001187324524e-01</internalNodes>
  <leafValues>

```

```

1.4906179904937744e+00 -1.3815900683403015e-
01</leafValues></_>
<_>
<internalNodes>
0 -1 60 9.1859996318817139e-03</internalNodes>
<leafValues>
-9.6361500024795532e-01 2.7665498852729797e-
01</leafValues></_>
<_>
<internalNodes>
0 -1 61 -3.7737999111413956e-02</internalNodes>
<leafValues>
-2.4464108943939209e+00 2.3619599640369415e-
01</leafValues></_>
<_>
<internalNodes>
0 -1 62 1.8463000655174255e-02</internalNodes>
<leafValues>
1.7539200186729431e-01 -
1.3423130512237549e+00</leafValues></_>
<_>
<internalNodes>
0 -1 63 -1.1114999651908875e-02</internalNodes>
<leafValues>
4.8710799217224121e-01 -8.9851897954940796e-
01</leafValues></_>
<_>
<_>
<internalNodes>
0 -1 2904 3.7589999847114086e-03</internalNodes>
<leafValues>
-5.8301001787185669e-02 3.4483298659324646e-
01</leafValues></_>
<_>

```

```

    <internalNodes>
      0 -1 2905 1.6263000667095184e-02</internalNodes>
    <leafValues>
      -1.5581500530242920e-01 8.6432701349258423e-
01</leafValues></_>
  <_>
    <internalNodes>
      0 -1 2906 -4.0176000446081161e-02</internalNodes>
    <leafValues>
      -6.1028599739074707e-01 1.1796399950981140e-
01</leafValues></_>
  <_>
    <internalNodes>
      0 -1 2907 2.7080999687314034e-02</internalNodes>
    <leafValues>
      -4.9601998180150986e-02 -8.9990001916885376e-
01</leafValues></_>
  <_>
    <internalNodes>
      0 -1 2908 5.2420001477003098e-02</internalNodes>
    <leafValues>
      1.1297199875116348e-01 -
1.0833640098571777e+00</leafValues></_>
  <_>
    <internalNodes>
      0 -1 2909 -1.9160000607371330e-02</internalNodes>
    <leafValues>
      -7.9880100488662720e-01 -3.4079000353813171e-
02</leafValues></_>
  <_>
    <internalNodes>
      0 -1 2910 -3.7730000913143158e-03</internalNodes>
    <leafValues>
      -1.9124099612236023e-01 2.1535199880599976e-

```

```

01</leafValues></_>
  <_>
    <internalNodes>
      0 -1 2911 7.5762003660202026e-02</internalNodes>
    <leafValues>
      -1.3421699404716492e-01
1.6807060241699219e+00</leafValues></_>
  <_>
    <internalNodes>
      0 -1 2912 -2.2173000499606133e-02</internalNodes>
    <leafValues>
      4.8600998520851135e-01 3.6160000599920750e-
03</leafValues></_></weakClassifiers></_></stages>
<features>
  <_>
    <rects>
      <_>
        6 4 12 9 -1.</_>
      <_>
        6 7 12 3 3.</_></rects></_>
    <_>
      <rects>
        <_>
          6 4 12 7 -1.</_>
        <_>
          10 4 4 7 3.</_></rects></_>
    <_>
      <rects>
        <_>
          3 9 18 9 -1.</_>
        <_>
          3 12 18 3 3.</_></rects></_>
    <_>
      <rects>

```

```

<_>
  8 18 9 6 -1.</_>
<_>
  8 20 9 2 3.</_></rects></_>
<_>
  <rects>
    <_>
      3 5 4 19 -1.</_>
    <_>
      5 5 2 19 2.</_></rects></_>
<_>
  <rects>
    <_>
      6 5 12 16 -1.</_>
    <_>
      6 13 12 8 2.</_></rects></_>
<_>
  <rects>
    <_>
      5 8 12 6 -1.</_>
    <_>
      5 11 12 3 2.</_></rects></_>
<_>
  <rects>
    <_>
      11 14 4 10 -1.</_>
    <_>
      11 19 4 5 2.</_></rects></_>
<_>
  <rects>
    <_>
      4 0 7 6 -1.</_>
    <_>
      4 3 7 3 2.</_></rects></_>

```

```

<_>
<rects>
  <_>
    6 6 12 6 -1.</_>
  <_>
    6 8 12 2 3.</_></rects></_>
<_>
<rects>
  <_>
    6 4 12 7 -1.</_>
  <_>
    10 4 4 7 3.</_></rects></_>
<_>
<rects>
  <_>
    1 8 19 12 -1.</_>
  <_>
    1 12 19 4 3.</_></rects></_>
<_>
<rects>
  <_>
    0 2 24 3 -1.</_>
  <_>
    8 2 8 3 3.</_></rects></_>
<_>
<rects>
  <_>
    9 9 6 15 -1.</_>
  <_>
    9 14 6 5 3.</_></rects></_>
<_>
<rects>
  <_>
    5 6 14 10 -1.</_>

```

```

<_>
  5 11 14 5 2.</_></rects></_>
<_>
  <rects>
    <_>
      5 0 14 9 -1.</_>
    <_>
      5 3 14 3 3.</_></rects></_>
<_>
  <rects>
    <_>
      13 11 9 6 -1.</_>
    <_>
      16 11 3 6 3.</_></rects></_>
<_>
  <rects>
    <_>
      7 5 6 10 -1.</_>
    <_>
      9 5 2 10 3.</_></rects></_>
<_>
  <rects>
    <_>
      10 8 6 10 -1.</_>
    <_>
      12 8 2 10 3.</_></rects></_>
<_>
  <rects>
    <_>
      2 5 4 9 -1.</_>
    <_>
      4 5 2 9 2.</_></rects></_>
<_>
  <rects>

```

```

<_>
  18 0 6 11 -1.</_>
<_>
  20 0 2 11 3.</_></rects></_>
<_>
<rects>
  <_>
    0 6 24 13 -1.</_>
  <_>
    8 6 8 13 3.</_></rects></_>
<_>
<rects>
  <_>
    9 6 6 9 -1.</_>
  <_>
    11 6 2 9 3.</_></rects></_>
<_>
<rects>
  <_>
    7 18 10 6 -1.</_>
  <_>
    7 20 10 2 3.</_></rects></_>
<_>
<rects>
  <_>
    5 7 14 12 -1.</_>
  <_>
    5 13 14 6 2.</_></rects></_>
<_>
<rects>
  <_>
    0 3 24 3 -1.</_>
  <_>
    8 3 8 3 3.</_></rects></_>

```



```

<_>
<rects>
  <_>
    5 8 15 6 -1.</_>
  <_>
    5 11 15 3 2.</_></rects></_>
<_>
<rects>
  <_>
    9 6 5 14 -1.</_>
  <_>
    9 13 5 7 2.</_></rects></_>
<_>
<rects>
  <_>
    9 5 6 10
<_>
<rects>
  <_>
    13 13 10 8 -1.</_>
  <_>
    18 13 5 4 2.</_>
  <_>
    13 17 5 4 2.</_></rects></_>
<_>
<rects>
  <_>
    1 13 10 8 -1.</_>
  <_>
    1 13 5 4 2.</_>
  <_>
    6 17 5 4 2.</_></rects></_>
<_>
<rects>

```

```

<_>
  15 8 4 15 -1.</_>
<_>
  15 13 4 5 3.</_></rects></_>
<_>
<rects>
  <_>
    5 8 4 15 -1.</_>
  <_>
    5 13 4 5 3.</_></rects></_>
<_>
<rects>
  <_>
    6 11 16 12 -1.</_>
  <_>
    6 15 16 4 3.</_></rects></_>
<_>
<rects>
  <_>
    2 11 16 12 -1.</_>
  <_>
    2 15 16 4 3.</_></rects></_>
<_>
<rects>
  <_>
    14 12 7 9 -1.</_>
  <_>
    14 15 7 3 3.</_></rects></_>
<_>
<rects>
  <_>
    10 1 3 21 -1.</_>
  <_>
    10 8 3 7 3.</_></rects></_>

```

```

<_>
<rects>
  <_>
    13 11 9 4 -1.</_>
  <_>
    13 13 9 2 2.</_></rects></_>
<_>
<rects>
  <_>
    3 10 17 9 -1.</_>
  <_>
    3 13 17 3 3.</_></rects></_>
<_>
<rects>
  <_>
    13 8 8 15 -1.</_>
  <_>
    13 13 8 5 3.</_></rects></_>
<_>
<rects>
  <_>
    3 8 8 15 -1.</_>
  <_>
    3 13 8 5 3.</_></rects></_>
<_>
<rects>
  <_>
    11 14 10 8 -1.</_>
  <_>
    16 14 5 4 2.</_>
  <_>
    11 18 5 4 2.</_></rects></_>
<_>
<rects>

```

```

<_>
  0 18 22 6 -1.</_>
<_>
  0 18 11 3 2.</_>
<_>
  11 21 11 3 2.</_></rects></_>
<_>
<rects>
  <_>
    0 16 24 4 -1.</_>
  <_>
    0 16 12 4 2.</_></rects></_>
<_>
<rects>
  <_>
    6 20 12 3 -1.</_>
  <_>
    12 20 6 3 2.</_></rects></_>
<_>
<rects>
  <_>
    18 12 6 12 -1.</_>
  <_>
    21 12 3 6 2.</_>
  <_>
    18 18 3 6 2.</_></rects></_>
<_>
<rects>
  <_>
    0 12 6 12 -1.</_>
  <_>
    0 12 3 6 2.</_>
  <_>
    3 18 3 6 2.</_></rects></_>

```

```

<_>
<rects>
  <_>
    15 17 9 6 -1.</_>
  <_>
    15 19 9 2 3.</_></rects></_>
<_>
<rects>
  <_>
    1 6 22 10 -1.</_>
  <_>
    1 6 11 5 2.</_>
  <_>
    12 11 11 5 2.</_></rects></_>
<_>
<rects>
  <_>
    15 17 9 6 -1.</_>
  <_>
    15 19 9 2 3.</_></rects></_>
<_>
<rects>
  <_>
    0 18 18 2 -1.</_>
  <_>
    0 19 18 1 2.</_></rects></_>
<_>
<rects>
  <_>
    3 15 19 3 -1.</_>
  <_>
    3 16 19 1 3.</_></rects></_>
<_>
<rects>

```

```

<_>
0 13 18 3 -1.</_>
<_>
0 14 18 1 3.</_></rects></_>
<_>
<rects>
<_>
15 17 9 6 -1.</_>
<_>
15 19 9 2 3.</_></rects></_>
<_>
<rects>
<_>
0 17 9 6 -1.</_>
<_>
0 19 9 2 3.</_></rects></_>
<_>
<rects>
<_>
12 17 9 6 -1.</_>
<_>
12 19 9 2 3.</_></rects></_>
<_>
<rects>
<_>
3 17 9 6 -1.</_>
<_>
3 19 9 2 3.</_></rects></_>
<_>
<rects>
<_>
16 2 3 20 -1.</_>
<_>
17 2 1 20 3.</_></rects></_>

```

```

<_>
  <rects>
    <_>
      0 13 24 8 -1.</_>
    <_>
      0 17 24 4 2.</_></rects></_>
  <_>
    <rects>
      <_>
        9 1 6 22 -1.</_>
      <_>
        12 1 3 11 2.</_>
      <_>
        9 12 3 11 2.</_></rects></_></features></cascade>
</opencv_storage>

```

9. ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : BAYRAKTARGİL, Berk

Uyruğu : T.C.

Doğum tarihi : 05.06.1998

Doğum yeri : Ankara

Telefon : 05343977517

E-posta : bbayraktarg@gmail.com

Eğitim

Derece	Eğitim	Mezuniyet tarihi
Lisans	Yozgat Bozok Üniversitesi	2022
Lise	Çorlu Mimar Sinan Anadolu Lisesi	2016

İş Deneyimi

Tarih	Yer/Konum	Görevi
2019 Ağustos-Eylül	Albayrak Holding (İstanbul)	Stajyer
2021 Eylül-Ekim	Piri Medya (İstanbul)	Stajyer
2022 Ağustos-Eylül	Baykar Teknoloji (Çorlu)	Stajyer

Yabancı Dil

İngilizce Orta, Almanca Basit

(YOKDİL : 67)

(YDS : 59)