



Sakarya Üniversitesi
Bilgisayar ve Bilişim Bilimleri Fakültesi
Bilgisayar Mühendisliği Bölümü

Bitirme Projesi
İnsan Yüzü Tanıma

Proje Danışmanı
Dr.Öğr.Üyesi Serap ÇAKAR

Proje Grubu
Tahsin NALCI - G171210082

Mayıs 2021

Önsöz

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 içinde her geçen gün daha yeni ve performansı yüksek methodlar ortaya atılmaktadır. Otomatik insan yüzü tanıma bunlardan biridir. İnsan yüzündeki özelliklerin değişmez ve kolay ayırt edilebilir olmasından ötürü gelecek vadeden bir yöntem olarak görülmektedir.

Daha önceden yapılan çalışmalar ışığında, günümüz ihtiyaçları da göz önüne alınarak otomatik insan yüzü tanıma, Viola-Jones Algoritması kullanılarak yapılmış ve sonuç olarakta oldukça yüksek performans elde edilmiştir.

Preface

Time, speed and reliability are important in all kinds of operations in daily life. At the entrance of a large company, it is inevitable that employees will use card readers or similar methods at every entrance and exit. But the security chain of these methods used should not be broken. In this, more and more new and high-performance methods are being introduced every day. Automatic human face recognition is one of them. It is seen as a promising method due to the fact that the features on the human face are invariant and easily distinguishable.

In light of previous studies, automatic human face recognition, taking into account today's needs, was performed using the Viola-Jones algorithm, and as a result, quite high performance was achieved.

İÇİNDEKİLER

Özet.....	4
Abstract.....	5
1. Giriş.....	6
2. İnsan Yüzü Tanımada Daha Önceden Yapılmış Çalışmalar	8
2.1. İnsanların Kullandıkları Yüz Karakteristikleri.....	8
2.2. Template Matching	8
2.3. Özellik Tabanlı Denkleştirme	8
2.4. İstatistiksel Yaklaşım	9
2.5. Neural Ağ Yaklaşımı.....	9
2.6. Özellik Vektör Yaklaşımı.....	9
3. İnsan Yüzü Tanıma Sistemi	10
3.1. Viola-Jones Algoritması.....	10
3.1.1. Haar Benzeri Özellikler.....	10
3.1.2. İntegral Görüntüler	13
3.1.3. AdaBoost.....	15
3.1.4. Sınıflandırıcı Basamakları Oluşturma.....	19
3.1.5. Viola-Jones Sınıflandırıcı Kullanımı	21
3.2. OpenCV.....	23
3.2.1. OpenCV Bileşenleri	24
3.3. Alternatif Görüntü İşleme Kütüphaneleri	24
4. İnsan Yüzü Tanıma Programı	26
5. Sonuçlar ve Öneriler	32
6. Ekler	33
6.1. Görüntü İşleme Temel Kavramlar.....	33
6.1.1. Pikseller ve Pencere	33
6.1.2. Histogram.....	33
6.2. Temel Görüntü İşleme Operatörleri	34
6.2.1. Monadic: Tek nokta dönüşümleri	34
6.2.2. Dyadic: Çift nokta dönüşümleri	34
6.2.3. Konvolüsyon :Şablon dönüşümleri	35
6.3. Görüntü İyileştirme	35
6.3.1. Alçak geçiren filtre.....	35
6.3.2. Yüksek geçiren filtre	35
6.3.3. Kenar belirleme operatörleri	36
6.3.4. Medyan Filtre	38
6.4. Görüntü Tanıma Teknikleri ve Örnekleri.....	38
7. Kaynaklar	43

Özet

İnsanlar birbirlerini yüzlerine bakarak kolayca tanırlar. Bunun en güzel kanıtı bebeklerin ebeveynlerinin yüzlerini kolayca öğrenip tanımalarıdır. Bilgisayarın bir kimsenin yüzüne bakarak tanımayı öğrenmesini düşünmek oldukça mantıksaldır. Aynı zamanda bilgisayar ancak bir bilgisayar gibi öğrenebilir, bir insan gibi değil.

Bilgisayarların yüksek tekrarlı görevleri yürütebildikleri bilindiğinden beri onlara insan yüzlerini tanıyabilmeleri için birçok çaba harcanmıştır. Otomatik insan yüzü tanıma normal olarak iki probleme sahiptir : İnsan yüzlerine göre tanıma ve doğrulama. Tanıma görevinde yeni bir insan yüzü geldiğinde, veri tabanında saklanan ve kimlik bilgileri bilinen insan yüzleriyle karşılaştırılır. Her bir karşılaştırma benzerlik değeri üretir ve bunlar karşılaştırılan insan yüz çiftleri için benzerlik dereceleri oluşturur. Sonuçta, çakışan aday listesi büyükten küçüğe oluşturulur. Doğrulama görevinde ise kişinin yüz resmi ve iddia ettiği kimliği girilir ve bilgisayarda ile iddia edilen kimliğe ait yüz bilgileri veri tabanından çekilir ve kişinin yüz resmi ile karşılaştırılır sonuç olarak, bilgisayar iddia edilen kimliği kabul eder veya reddeder.

Bu projede yapılan çalışmalar insan yüzünün real-time olarak tanınması hedef alınmıştır. Tanıma yöntemi olarak “Viola-Jones Algoritması” kullanılmıştır.

Abstract

People easily recognise one another by looking at each other's face. There is an evidence that infants quickly learn to identify the faces of their parents. Hence, it is logical for us to imagine that a computer easily can be taught how to recognise an individual by looking at his/her face. After all, a computer learns like a computer and not yet like a human.

Since computers are known for being capable of performing highly repetitive tasks, many efforts are still being made to "teach" them to recognise human faces. The automated human face-recognition tasks normally are formulated into two categories of problems: identification and verification based on human faces. In an identification task, a human face is presented and compared with each of the human faces stored in a data base, whose identities are known. Each comparison produces a similarity score, which indicates the degree of similarity between the pair of human faces compared. As a result, a matching candidate list can be produced in descending order of the similarity scores. In a verification task, a human face is presented together with the claimed identity for comparisons. The computer retrieves the stored human face (or faces) of the claimed identity and compares that with the presented face. A similarity score is again produced for each comparison. This similarity score is then used to determine whether the individual is of the identity he/she claims.

The aim of the work performed in the project is to develop a real time face recognition system and the method used in the project is Viola-Jones algorithm.

1.Giriş

Sayısal Görüntü İşleme tıp, basın, endüstri, uzay ve bilimde uygulanan, her geçen gün gelişen ve yeni araştırmalar yapılan bir alandır. Bilgisayar teknolojilerindeki hızlı gelişim araştırma konularının uygulanabilirliğini artırmıştır. İnsan yüzü tanıma, sayısal görüntü işlemeye dayanmaktadır ve temel son zamanlarda çok fazla ilgi görmektedir.

İnsan yüzü tanıma başlıca iki alanda kullanılmaktadır. Birincisi ticari amaçlı uygulamalardır. Büyük çaplı şirketlerde kapı girişi kontrollerinde, banka kartları, kredi kartları ve ATM kartlarında kimlik tespitinde kullanım alanı bulabilmektedir. İkincisi ise kanun içerikli konulardır. Suçlu tespiti, resimli kimlik kartları, ehliyet ve pasaporttan kimlik tespiti, videodaki hareketli görüntülerden şüphelilerin tespit edilmesinde kullanılabilmektedir. Anlaşılacağı gibi kullanım hedefi tekdir : Güvenlik.

90'lı yıllardan sonra insan yüzü tanıma teknolojisinde önemli bir gelişme gözlenmiştir. Araştırmalar çoğunlukla vesikalık fotoğraf gibi sabit resimlerden tanıma üzerine yoğunlaşmıştır. Bu fotoğraflar profil görüntüleri olabileceği gibi, tam karşıdan alınan görüntüler de olabilmektedir. Ayrıca gülme, ağzı açma gibi yüzün değişik biçimlerinde tanıma için çalışmalar yapılmıştır. Fakat bütün çalışmalarda temel olarak, alınan görüntünün içinde yüzün belirlenmesi ve bölümlere ayrılmasından başlayarak göz, burun, ağız gibi kısımların yerinin tespiti yapılmıştır. Bunun yanı sıra son zamanlarda istatistiksel ve nöron ağları kullanarak yüzün tanınması üzerine araştırmalar yapılmaktadır.

Bu çalışmada hedefimiz video kameradan alınan görüntüleri kullanarak kimlik tespiti yapabilmektir. Alınan görüntüler ile ilgili bilgiler bir veri tabanında tutulacaktır. Yeni bir görüntü alındığında veri tabanı ile karşılaştırılacak eğer bulunursa kimliği tespit edilecek, bulunamazsa sistem tanımayacaktır. Alınan görüntünün farklı açılardan olabileceği göz önüne alınarak insan yüzünün bilgileri çeşitli yöntemler uygulanarak bulunacaktır.

Bir insanı tarif edebilmek için birçok özellik sayılabilir. Boyu, kilosu, saç rengi bunlardan bazılarıdır. Fakat yapılan çalışmalara göre kimlik belirlemede bu sayılanlar yeterli olmamaktadır. Saç rengi kolayca değiştirilebilir. Ayrıca aynı boy ve kiloda birçok insan vardır. Oysa tanımda insan yüzünü kullanma hem avantajlı hem de daha belirgindir. Çünkü insan yüzü gözler, burun, ağız, kaşlar gibi değişmeyen kısımlara sahiptir ve bu nedenle tanımda hız ve kolaylık sağlanmaktadır.

İnsan yüzü tanımda kullanılan teknikler iki grupta toplanabilir : statik (kontrollü ve düzenli fotoğraflardan tanıma) ve dinamik (kontROLSÜZ video görüntülerinden tanıma) uyumu. Bu gruplar arasında görüntü kalitesine ve arka planın karmaşıklığına bağlı olarak önemli farklılıklar vardır. Eğer görüntü kalitesi iyi değilse yapılabilecekler : parlaklık ayarı, uygun filtreleme yöntemini kullanma ya da uygun eşik değerini uygulamaktır. Eğer arka plan ile ilgili sorunlar varsa uygun bölümlendirme ve sınıflama teknikleri kullanılarak iyi sonuçlar alınabilir.

İnsan yüzü tanıma teknikleri kısaca aşağıda özetlenmiştir :

Uygulama	Avantajları	Dezavantajları
Kredi Kartları, Ehliyet, pasaport ve kimlik kartları	Kontrollü görüntü Kontrollü bölümlene Kaliteli görüntü	Hazır veri tabanı Büyük veri tabanına sahip olmaması
Suçlu tespiti	Karışık görüntü kalitesi Birden fazla görüntü	Nadiren arama
Banka/Şirket güvenliği	Kolaylık, hız Bölgesel aramalar	KontROLSÜZ bölümlene Düşük görüntü miktarı
Kalabalık gözetimi	Kolaylık, hız Küçük dosya boyutu Video görüntü uygunluğu	KontROLSÜZ bölümlene Düşük görüntü miktarı Gerçek-zamanlı

2. İnsan Yüzü Tanımada Daha Önceden Yapılmış Çalışmalar

İnsan yüzü tanıma konusunda değişik yaklaşımlarla, değişik çalışmalar yapılmıştır. Bu yaklaşımların her birinin önemli avantajları olmuştur. Fakat şu ana kadar mükemmel bir metot geliştirilememiştir.

2.1. İnsanların Kullandıkları Yüz Karakteristikleri

İnsanlar birbirlerini genellikle yüzlerinden tanırlar. Bazen bir insanı arkasından, tanıdığımız birisine benzetebiliriz. Fakat sadece yüzüne bakarak tahminimizin doğru olup olmadığını anlayabiliriz. Ressamlarda genellikle insan yüzü portreleri yaparlar. 1973 yılında bir Bilimsel Amerikan yayınında bu konuda ilginç bir yazı yayınlanmıştır.

Bir ressama gösterilen önden çekilmiş bir fotoğraf ışığında, ressam kişinin yüz özelliklerini bir kağıda tarif etmiştir. Bu tarifler başka bir ressama gösterilerek onun fotoğraftaki kişinin resmini yapması istenmiştir. Elde edilen sonuç, bazı özelliklerin aynen, bazılarının ise belirsiz olarak tarif edilmesidir.

2.2. Template Matching

Template matching, bilgisayar yardımı ile yüz tanıma işlemine bir yaklaşım olarak, tüm resmin grilik seviyesi kalıplarını kullanmaya dayanan çok basit bir tanıma metodudur. Resim template matching yönteminin en basit versiyonunda koyuluk değerlerini içeren bir matris olarak gösterilir ve bütün yüzü belirten tek bir kalıpla karşılaştırılır. Elbette template matching yöntemini uygulayabileceğimiz daha akıllıca yollar vardır. Mesela, grilik seviyelerini içeren bir matris farklı bakış yönlerinden tanıma yapmak için kullanılabilir. Diğer önemli bir metot ise ,tek bir bakış yönü için çoklu kalıplar kullanmaktır.

2.3. Özellik Tabanlı Denkleştirme

Yüz tanıma işlemi, yüz kısımları detaylar halinde çıkarılarak kaba bir şekilde yapılabilir. Ayırım için yüz kısımlarının kapsam-geometrik konfigürasyonu yeterlidir. Kapsamlı konfigürasyon yüzün ana kısımlarının boyutunu ve yerini belirten bir nümerik data vektörü tarafından belirtilebilir. Elde edilmiş yüz kısımlarının, yüzün dönmesinden, ölçüsünden ve yerinden bağımsız olabilmesi için, uygun bir şekilde normalize edilmiş olması lazımdır. Yüz tanıma zor olmasına rağmen, sunduğu ilginç avantajları yüz kısımlarının iyileştirilmesinde kullanıyoruz. İlk önemli avantajı iki taraflı simetridir. Diğer kolaylıklar için ise, her yüzün benzer bir şekilde iki göz ,bir burun ve bir ağızdan oluşması gerçeğinden çıkartılabilir. Bu durum sınıflandırma işlemini zorlaştırmasına rağmen yüz kısımlarının çıkartılması işlemini kolaylaştırır. Yüz kısımlarını çıkartılması için en kullanışlı yöntem integral izdüşümüdür. Elimizdeki resim $T(X,Y)$ matrisi olarak ifade edilir

$T(X,Y)$ nin integral izdüşümleri :

$T(X,Y)$ nin $(x_1,y_1)*(x_2,y_2)$ dikdörtgeni içerisindeki düşey integral izdüşümü : $V(X)$, y nin y_1 den y_2 ye kadar olan değerleri için $T(X,Y)$ lerin toplamına eşittir. $T(X,Y)$ nin yatay integral izdüşümü:

$V(Y)$, x in x_1 den x_2 ye kadar olan değerleri için $T(X,Y)$ lerin toplamına eşittir.

Dikdörtgen pencere yüz kısımlarının üzerinde uygun bir şekilde hareket ettirilerek integral izdüşümleri bulunur. Hesaplanan integral izdüşümleri yüz kısımlarının yerini saptamada kullanılır.

Laplacian operatörlerinin kullanılması kenar yönleri hakkında bize bilgi vermez.

Yatay farklar (gradient) yüzün ve burnun sağ ve sol sınırlarını çıkarmak için ,düşey farklar ise gözleri, burun tabanını ve ağız bulmak için kullanılır.

Yüz kısımlarını çıkarmak için yapacağımız ilk iş, template matching yöntemini kullanarak gözlerin yerini tam olarak saptırız. Yüzün kalan kısımlarının nerede olduğu, aralarındaki uzaklık oranı hakkında bir ön tahmine sahip olduğumuz için diğer kısımların (burunun, ağızın) yerini saptamak çok kolay olur.

2.4. İstatistiksel Yaklaşım

Bu yöntemde eigenfaces kullanılır. Veri tabanındaki her bir görüntünün gösterilmesinde kullanılan ağırlık vektörleri görüntünün eigenfaces bileşenlerinin tasarımıyla elde edilir. Test edilen görüntünün tanınması veri tabanındaki referans ağırlıklarının karşılaştırılmasıyla gerçekleştirilir. Bulunan en yakın ağırlık test görüntüsüyle adlandırılır. Yine de bu sistem, aydınlatma, büyüklük, deneme ve test şartları düşünülerekten küçük bir sapma değerine sahiptir.

2.5. Neural Ağ Yaklaşımı

Genel olarak ağ, gerekli haritayı girişten çıkışa doğru öğrenir. Öğrenilen harita karşılıklı bağlantı ilişkisi ağırlıklarını değiştirebilmeyi içerir ve bu hata düzeltmede kullanılan bir yöntemdir. Her bir giriş sonucunda elde edilen çıkış, istenen çıkışla karşılaştırılır ve doğru çıkışı elde edebilmek için kullanılan ağırlıklar değiştirilebilir.

Neural ağlar sonuç fonksiyonuna göre düğüm karakteristiğindedir. Nöronların yenilenmesi kuralı tamamen dinamik yapıdadır.

En basit yapı eşik nöronlarındadır ki sonuç fonksiyonu sıfırdan bireddir. Burada nöron toplam girişlerini uygular ve bunu bazı iç eşik seviyeleriyle karşılaştırır. Eğer seviye sağlanırsa onu çevirir, sağlanmazsa kaldırılır.

Nöronlar birbirine bağlanarak bir ağ yapısı oluşturulur. En büyük değişme miktarı deneme ve öğrenme kurallarındadır. Bu kurallar giriş ağırlıklarıyla oluşturulur ve daha sonra gerektiğinde değiştirilir.

2.6. Özellik Vektör Yaklaşımı

Bu yöntemde öncelikle video kameradan görüntü alınır. Görüntüde yüzü ve yüzdeki diğer kısımları belirginleştirmek için başın, gözlerin, ağızın ve burunun etrafı dikdörtgenlerle çizilir. Bunun için görüntünün yoğunluğundan ve yüzdeki kısımların dağılımlarındaki simetriden yararlanılır. Temel olarak üç dikdörtgen vardır : Birincisi başı arka plandan ayırmak için. İkincisi gözleri ve üçüncüsü de ağız, burunu ayırmak içindir. Gözün yerini bulabilmek için aşağıda sayılan metotlar uygulanır.

1. Eşik Uygulaması : Düşük eşik değeri uygulanarak irisin de içinde bulunduğu karanlık bölge ayrıştırılır.
2. Şablon Uyumu : Yuvarlak bir model her bir göz penceresinde karşılaştırılırken merkez ve çap etrafında çakışan noktalar sayılır. Bulunan maksimum değer sonucu verir.
3. Yuvarlak Arama : İris merkezlerini bulabilmek için rasgele noktalar alınır ve yuvarlak kenarların bu noktaları çevreleyip çevrelemediği kontrol edilir. Eğer yuvarlak içindeki ortalama yoğunluk dışarısına göre küçükse kabul edilir. Yoksa reddedilir.

3.İnsan Yüzü Tanıma Sistemi

3.1.Viola-Jones Algoritması

Bu algoritma, 2001 yılında yöntemi öneren iki bilgisayarlı görme araştırmacısının adını almıştır: Paul Viola ve Michael Jones.

Gerçek zamanlı olarak rekabetçi nesne algılama oranları sağlayabilen genel bir nesne algılama çerçevesi geliştirdiler . Çeşitli tespit problemlerini çözmek için kullanılabilir, ancak asıl motivasyon yüz tanımadan gelir.

Viola-Jones algoritmasının 4 ana adımı vardır ve her biri hakkında aşağıdaki bölümlerde daha fazla bilgi edineceksiniz:

1. Haar benzeri özelliklerin

seçilmesi 2.Ayrılmaz bir görüntü

oluşturmak 3.AdaBoost eğitimini

çalıştırma

4.Sınıflandırıcı basamakları oluşturma

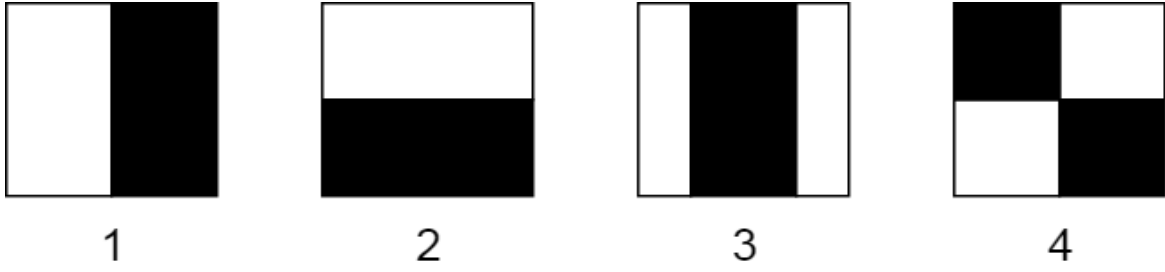
Bir görüntü verildiğinde, algoritma birçok küçük alt bölgeye bakar ve her alt bölgedeki belirli özellikleri arayarak bir yüz bulmaya çalışır. Birçok farklı konumu ve ölçeği kontrol etmesi gerekir, çünkü bir görüntü çeşitli boyutlarda birçok yüz içerebilir. Viola ve Jones, yüzleri algılamak için Haar benzeri özellikler kullandı.

3.1.1. Haar Benzeri Özellikler

Tüm insan yüzleri bazı benzerlikleri paylaşır. Bir kişinin yüzünü gösteren bir fotoğrafa bakarsanız, örneğin göz bölgesinin burun kemerinden daha koyu olduğunu görürsünüz. Yanaklar da göz bölgesine göre daha parlaktır. Bir görüntünün insan yüzü içerip içermediğini anlamamıza yardımcı olması için bu özellikleri kullanabiliriz.

Hangi bölgenin daha açık veya daha koyu olduğunu bulmanın basit bir yolu, her iki bölgenin piksel değerlerini toplamak ve bunları karşılaştırmaktır. Daha koyu bölgedeki piksel değerlerinin toplamı, daha açık bölgedeki piksellerin toplamından daha küçük olacaktır. Bu, Haar benzeri özellikler kullanılarak gerçekleştirilebilir .

Haar benzeri bir özellik, bir görüntünün dikdörtgen bir parçası alınarak ve bu dikdörtgeni birden çok parçaya bölerek temsil edilir. Genellikle siyah beyaz bitişik dikdörtgenler olarak görselleştirilir:



Temel Haar benzeri dikdörtgen özellikler

Bu görselde, Haar benzeri özelliklerin 4 temel türünü görebilirsiniz:

İki dikdörtgene sahip yatay özellik

İki dikdörtgene sahip dikey özellik

Üç dikdörtgene sahip dikey özellik

Dört dikdörtgene sahip çapraz özellik

İlk iki örnek, kenarları tespit etmek için kullanışlıdır. Üçüncüsü çizgileri algılar ve dördüncüsü, köşegen özellikleri bulmak için iyidir.

Özelliğin değeri tek bir sayı olarak hesaplanır: siyah alandaki piksel değerlerinin toplamı eksi beyaz alandaki piksel değerlerinin toplamı. Duvar gibi tek tip alanlar için bu sayı sıfıra yakın olur ve size anlamlı bir bilgi vermez.

Yararlı olması için, Haar benzeri bir özelliğin size büyük bir sayı vermesi gerekir, bu da siyah ve beyaz dikdörtgenlerdeki alanların çok farklı olduğu anlamına gelir. İnsan yüzlerini algılamak için çok iyi performans gösteren bilinen özellikler vardır:



Göz bölgesine uygulanan haar benzeri özellik. (Resim: [Wikipedia](#))

Bu örnekte göz bölgesi aşağıdaki bölgeden daha koyu. Bu özelliği, bir görüntünün hangi alanlarının belirli bir özellik için güçlü bir yanıt (çok sayıda) verdiğini bulmak için kullanabilirsiniz:



Burun köprüsüne uygulanan haar benzeri özellik. (Resim: [Wikipedia](#))

Bu örnek burun köprüsüne uygulandığında güçlü bir tepki verir. Bir görüntü bölgesinin insan yüzü içerip içermediğini anlamak için bu özelliklerin çoğunu birleştirebilirsiniz.

Belirtildiği gibi, Viola-Jones algoritması bir görüntünün birçok alt bölgesinde bu özelliklerin çoğunu hesaplar. Bu hızla hesaplama açısından pahalı hale gelir: bir bilgisayarın sınırlı kaynaklarını kullanmak çok zaman alır.

Bu sorunu çözmek için Viola ve Jones, integral görüntüleri kullandı.

3.1.2. İntegral Görüntüler

Bir integral görüntü (aynı zamanda bir toplam alan tablosu olarak da bilinir), hem bir veri yapısının hem de bu veri yapısını elde etmek için kullanılan bir algoritmanın adıdır. Bir görüntüdeki veya bir görüntünün dikdörtgen kısmındaki piksel değerlerinin toplamını hesaplamanın hızlı ve verimli bir yolu olarak kullanılır.

Ayrılmaz bir görüntüde, her noktanın değeri, hedef piksel dahil olmak üzere yukarıdaki ve soldaki tüm piksellerin toplamıdır:

1	3	7	5
12	4	8	2
0	14	16	9
5	11	6	10

1	4	11	16
13	20	35	42
13	34	65	81
18	50	87	113

Piksel değerlerinden ayrılmaz bir görüntü hesaplama

İntegral görüntü, orijinal görüntü üzerinden tek geçişte hesaplanabilir. Bu, bir dikdörtgenin içindeki piksel yoğunluklarının, dikdörtgen boyutundan bağımsız olarak dört sayı içeren yalnızca üç işlem halinde toplanmasını azaltır:

1	4	11	16
13	20	35	42
13	34	65	81
18	50	87	113

ABCD = ?

Turuncu dikdörtgendeki piksellerin toplamını hesaplayın.

ABCD dikdörtgenindeki piksellerin toplamı, $D - B - C + A$ formülü kullanılarak A , B , C ve D noktalarının değerlerinden türetilir . Bu formülü görsel olarak anlamak daha kolaydır:

1	4	11	16
13	20	35	42
13	34	65	81
18	50	87	113

D = 113

1	4	11	16
13	20	35	42
13	34	65	81
18	50	87	113

C = 50

1	4	11	16
13	20	35	42
13	34	65	81
18	50	87	113

B = 42

1	4	11	16
13	20	35	42
13	34	65	81
18	50	87	113

A = 20

$$113 - 50 - 42 + 20 = 41$$

Piksellerin toplamını adım adım hesaplama

Her iki çıkarılarak fark edeceksiniz B ve C araçları ile tanımlanan alan o A geri yeniden eklemeniz gerekir, böylece iki kez alınmış olur.

Artık iki dikdörtgenin piksel değerlerinin toplamı arasındaki farkı hesaplamanın basit bir yolu var. Bu, Haar benzeri özellikler için mükemmeldir!

3.1.3. AdaBoost

Güçlendirme şu soruya dayanır: "Bir grup zayıf öğrenci , tek bir güçlü öğrenci yaratabilir mi?" Zayıf bir öğrenci (veya zayıf sınıflandırıcı), rastgele tahmin etmekten yalnızca biraz daha iyi olan bir sınıflandırıcı olarak tanımlanır.

Yüz algılamada bu, zayıf bir öğrencinin bir görüntünün bir alt bölgesini bir yüz veya yüzüz olarak rastgele tahminden yalnızca biraz daha iyi sınıflandırabileceği anlamına gelir. Güçlü bir öğrenci, yüz olmayanlardan yüz seçmede büyük ölçüde daha iyidir.

Güçlendirmenin gücü, birçok (binlerce) zayıf sınıflandırıcının tek bir güçlü sınıflandırıcıda birleştirilmesinden gelir. Viola-Jones algoritmasında, Haar benzeri her özellik zayıf bir öğrenciyi temsil eder. AdaBoost, son sınıflandırıcıya giren bir özelliğin türüne ve boyutuna karar vermek için ona sağladığınız tüm sınıflandırıcıların performansını kontrol eder.

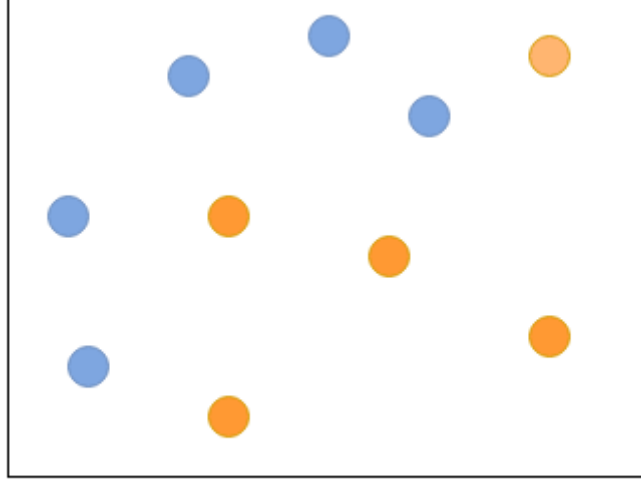
Bir sınıflandırıcının performansını hesaplamak için, onu eğitim için kullanılan tüm görüntülerin tüm alt bölgelerinde değerlendirirsiniz. Bazı alt bölgeler sınıflandırıcıda güçlü bir yanıt üretecektir. Bunlar pozitif olarak sınıflandırılır, yani sınıflandırıcı bir insan yüzü içerdiğini düşünür.

Sınıflandırıcıların görüşüne göre, güçlü bir yanıt üretmeyen alt bölgeler insan yüzü içermez. Negatif olarak sınıflandırılacaklar.

İyi performans gösteren sınıflandırıcılara daha fazla önem veya ağırlık verilir . Nihai sonuç, en iyi performans gösteren zayıf sınıflandırıcıları içeren, güçlendirilmiş sınıflandırıcı olarak da adlandırılan güçlü bir sınıflandırıcıdır.

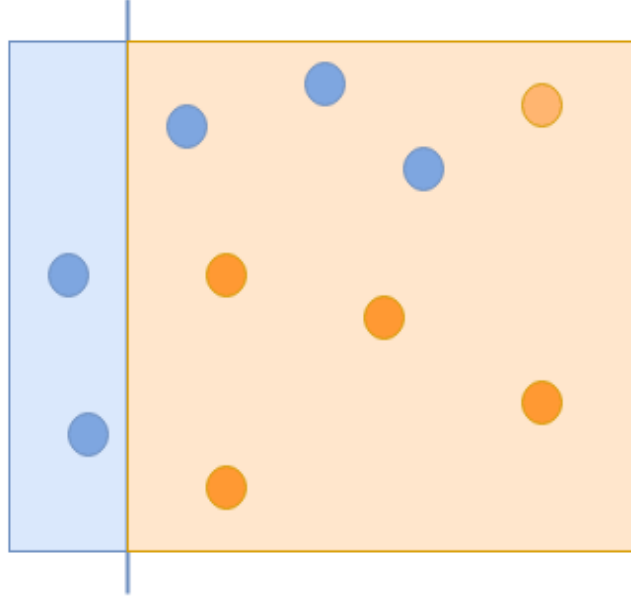
Algoritma uyarlanabilir olarak adlandırılır çünkü eğitim ilerledikçe yanlış sınıflandırılan görüntülere daha fazla vurgu yapar. Bu zor örneklerde daha iyi performans gösteren zayıf sınıflandırıcılar diğerlerinden daha güçlü ağırlıklandırılır.

Bir örneğe bakalım:



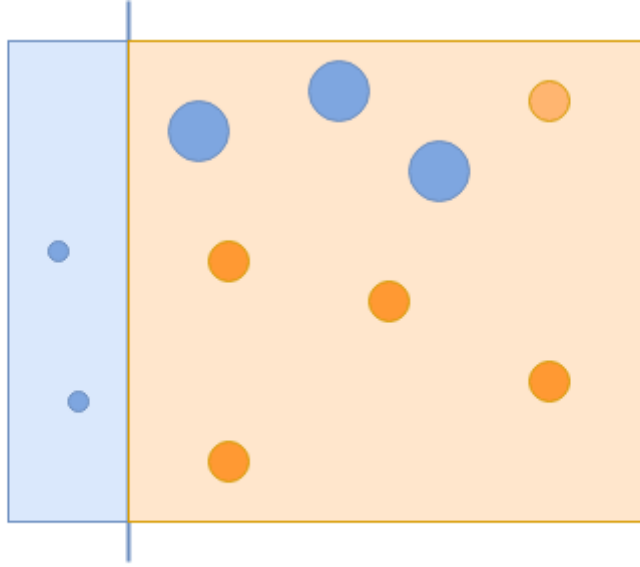
Mavi ve turuncu daireler, farklı kategorilere ait örneklerdir.

Aşağıdaki görüntüde bir dizi zayıf sınıflandırıcı kullanarak mavi ve turuncu daireleri sınıflandırmanız gerektiğini düşünün:



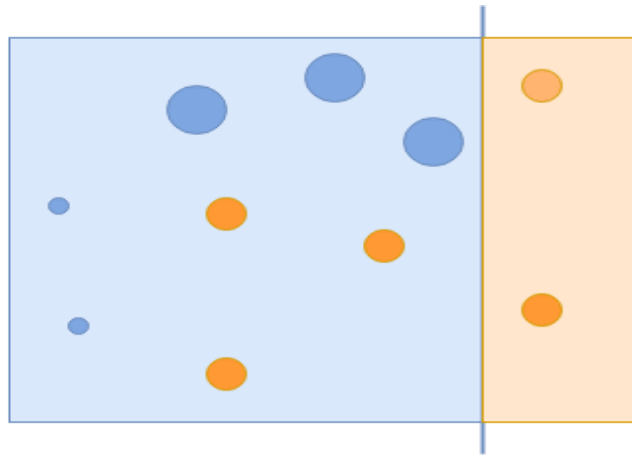
İlk zayıf sınıflandırıcı, mavi dairelerin bazılarını doğru bir şekilde sınıflandırır.

Kullandığınız ilk sınıflandırıcı, mavi dairelerin bazılarını yakalar, ancak diğerlerini gözden geçirir. Bir sonraki yinelemede, kaçırılan örneklerle daha fazla önem veriyorsunuz:



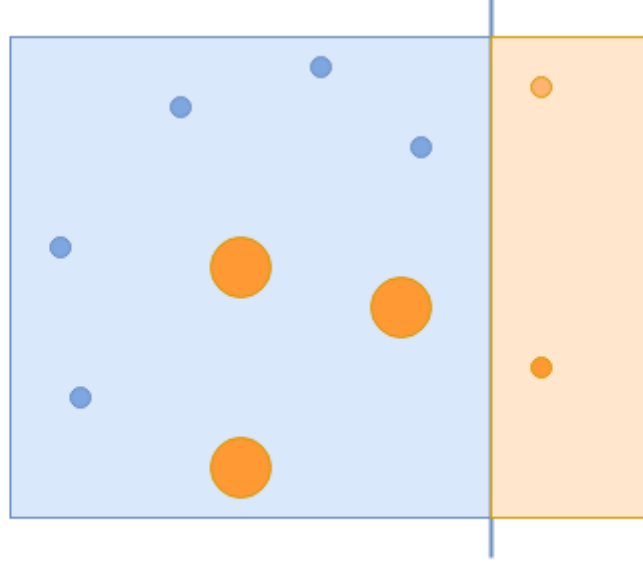
Kaçırılan mavi numunelere boyutlarına göre daha fazla önem verilmektedir.

Bu örnekleri doğru bir şekilde sınıflandırmayı başaran ikinci sınıflandırıcı, daha yüksek bir ağırlık alacaktır. Unutmayın, zayıf bir sınıflandırıcı daha iyi performans gösterirse, daha yüksek bir ağırlık elde eder ve dolayısıyla nihai, güçlü sınıflandırıcılara dahil edilme şansı artar:



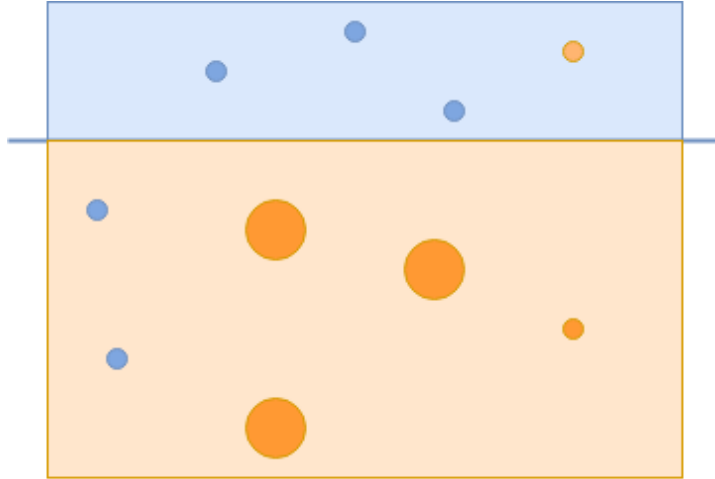
İkinci sınıflandırıcı, daha büyük mavi daireleri yakalar.

Şimdi tüm mavi daireleri yakalamayı başardınız, ancak bazı turuncu daireleri yanlış bir şekilde yakaladınız. Bu yanlış sınıflandırılmış turuncu dairelere bir sonraki yinelemede daha fazla önem verilmektedir:



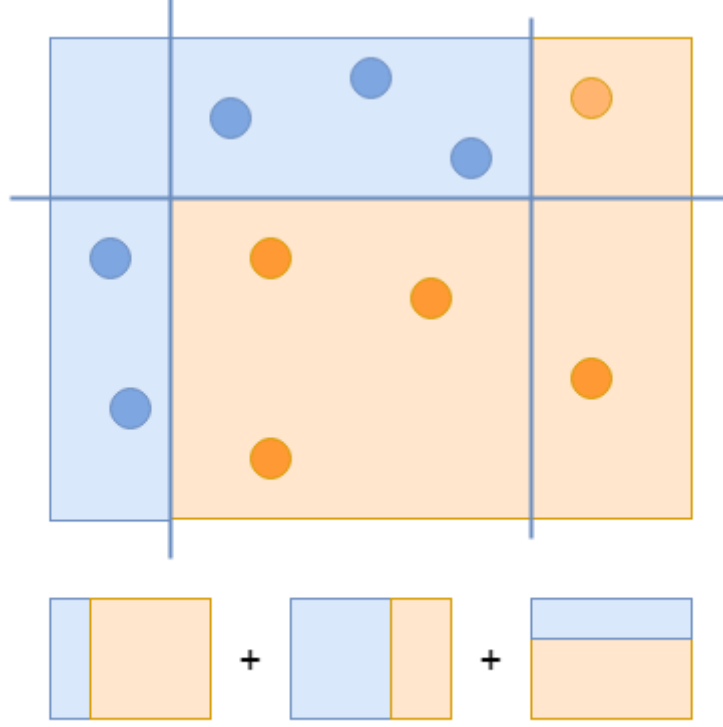
Yanlış sınıflandırılan turuncu dairelere daha fazla önem verilir ve diğerleri azaltılır.

Son sınıflandırıcı, bu turuncu daireleri doğru bir şekilde yakalamayı başarır:



Üçüncü sınıflandırıcı, kalan turuncu daireleri yakalar.

Güçlü bir sınıflandırıcı oluşturmak için, tüm örnekleri doğru bir şekilde sınıflandırmak için üç sınıflandırıcıyı da birleştirirsiniz:



Son, güçlü sınıflandırıcı, üç zayıf sınıflandırıcının tümünü birleştirir.

Viola ve Jones, bu sürecin bir varyasyonunu kullanarak, görüntülerde yüz bulmada uzmanlaşmış yüz binlerce sınıflandırıcıyı değerlendirdi. Ancak tüm bu sınıflandırıcıları her görüntüdeki her bölgede çalıştırmak hesaplama açısından pahalı olurdu, bu yüzden sınıflandırıcı kaskad adı verilen bir şey yarattılar .

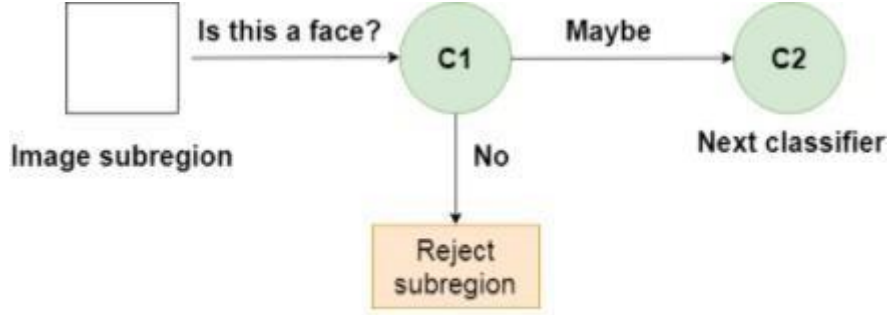
3.1.4. Sınıflandırıcı Basamakları Oluşturma

Bir çağlayanın tanımı, birbiri ardına gelen bir dizi şeledir. Benzer bir kavram, karmaşık bir problemi basit birimlerle çözmek için bilgisayar biliminde kullanılır. Buradaki sorun, her görüntü için hesaplama sayısını azaltmaktır.

Viola ve Jones, bunu çözmek için güçlü sınıflandırıcılarını (binlerce zayıf sınıflandırıcıdan oluşan) her zayıf sınıflandırıcının bir aşamayı temsil ettiği bir kademeye dönüştürdüler. Basamaklamanın işi, yüz olmayan kişileri hızla atmak ve değerli zaman ve hesaplamaları boşa harcamamaktır.

Bir görüntü alt bölgesi kademeye girdiğinde, ilk aşama tarafından değerlendirilir. Bu aşama alt bölgeyi olumlu olarak değerlendirirse, yani bir yüz olduğunu düşünürse, sahnenin çıktısı belki olur .

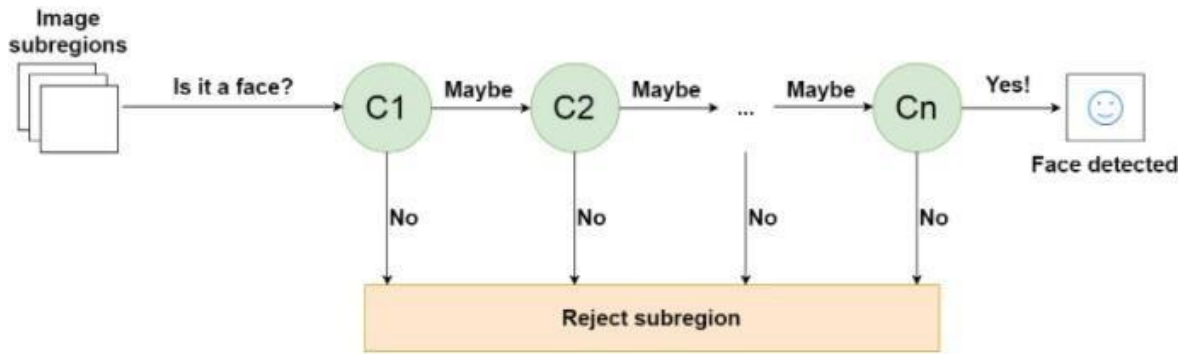
Bir alt bölge belki alırsa , kaskadın bir sonraki aşamasına gönderilir. Bu bir olumlu bir değerlendirme verirse, o zaman bu başka bir şeydir belki , ve görüntü üçüncü aşamaya gönderilir:



Bir kademede zayıf bir sınıflandırıcı

Bu işlem, görüntü kaskadın tüm aşamalarından geçene kadar tekrarlanır. Tüm sınıflandırıcılar görüntüyü onaylarsa, nihayet insan yüzü olarak sınıflandırılır ve kullanıcıya bir algılama olarak sunulur.

Bununla birlikte, ilk aşama olumsuz bir değerlendirme verirse, görüntü insan yüzü içermediği için hemen atılır. İlk aşamayı geçip ikinci aşamayı geçemezse, o da atılır. Temel olarak, görüntü, sınıflandırıcının herhangi bir aşamasında atılabilir:



Yüz algılama için bir dizi _n_ sınıflandırıcı

Bu, yüz olmayanların çok hızlı bir şekilde atılması için tasarlanmıştır, bu da çok fazla zaman ve hesaplama kaynağı tasarrufu sağlar. Her sınıflandırıcı bir insan yüzünün bir özelliğini temsil ettiğinden, pozitif bir tespit temelde "Evet, bu alt bölge bir insan yüzünün tüm özelliklerini içerir" der. Ancak bir özellik eksik olduğu anda tüm alt bölgeyi reddeder.

Bunu etkili bir şekilde başarmak için, en iyi performans gösteren sınıflandırıcılarınızı kademenin başlarına yerleştirmeniz önemlidir. Viola-Jones algoritmasında, gözler ve burun köprüsü sınıflandırıcılar en iyi performans gösteren zayıf sınıflandırıcılara örnektir.

Artık algoritmanın nasıl çalıştığını anladığınıza göre, onu Python ile yüzleri tespit etmek için kullanmanın zamanı geldi.

3.1.5. Viola-Jones Sınıflandırıcı Kullanımı

Bir Viola-Jones sınıflandırıcısını sıfırdan eğitmek uzun zaman alabilir. Neyse ki, önceden eğitilmiş bir Viola-Jones sınıflandırıcısı, OpenCV ile kutudan çıkar! Algoritmayı iş başında görmek için bunu kullanacaksınız.

Önce, insan yüzlerinin varlığını tespit etmek için taramak istediğiniz bir görüntü bulun ve indirin. İşte bir örnek:



Değişken hedef görüntü için tüm algılamalarını içermektedir. Algılamaları görselleştirmek için, tüm algılamaları yinelemeniz ve algılanan yüzlerin üzerine dikdörtgenler çizmeniz gerekir.

OpenCV `rectangle()`, görüntülerin üzerine dikdörtgenler çizer ve sol üst ve sağ alt köşenin piksel koordinatlarını bilmesi gerekir. Koordinatlar, görüntüdeki piksel sırasını ve sütununu gösterir.

Neyse ki, tespitler piksel koordinatları olarak kaydedilir. Her algılama, sol üst köşe koordinatları ve algılanan yüzü çevreleyen dikdörtgenin genişliği ve yüksekliği ile tanımlanır.

Satıra genişlik ve sütuna yükseklik eklemek size resmin sağ alt köşesini verecektir:

`rectangle()` aşağıdaki argümanları kabul eder:

- Orijinal görüntü
- Algılamanın sol üst noktasının koordinatları
- Algılamanın sağ alt noktasının koordinatları
- Dikdörtgenin rengi (kırmızı, yeşil ve mavi (0- 255) miktarını tanımlayan bir demet)
- Dikdörtgen çizgilerin kalınlığı

İşte sonuç:

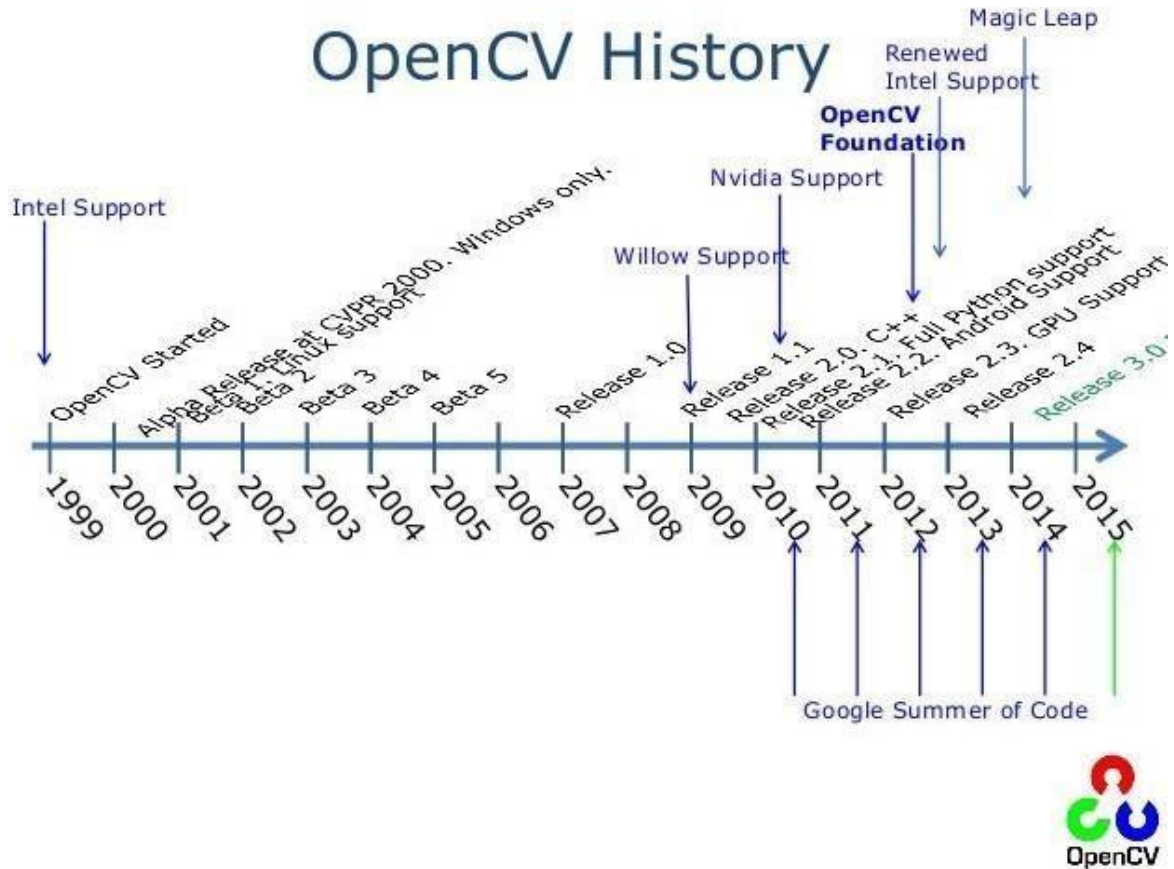


3.2. OpenCV

OpenCV (Open Source Computer Vision) açık kaynak kodlu görüntü işleme kütüphanesidir. 1999 yılında Intel tarafından geliştirilmeye başlanmış daha sonra Itseez, Willow, Nvidia, AMD, Google gibi şirket ve toplulukların desteği ile gelişim süreci devam etmektedir. İlk sürüm olan OpenCV alfa 2000 yılında piyasaya çıkmıştır. İlk etapta C programlama dili ile geliştirilmeye başlanmış ve daha sonra birçok algoritması C++ dili ile geliştirilmiştir. Open source yani açık kaynak kodlu bir kütüphanedir ve BSD lisansı ile altında geliştirilmektedir. BSD lisansına sahip olması bu kütüphaneyi istediğiniz projede ücretsiz olarak kullanabileceğiniz anlamına gelmektedir.

OpenCV platform bağımsız bir kütüphanedir, bu sayede Windows, Linux, FreeBSD, Android, Mac OS ve iOS platformlarında çalışabilmektedir. C++, C, Python, Java, Matlab, EmguCV kütüphanesi aracılığıyla da Visual Basic.Net, C# ve Visual C++ dilleri ile topluluklar tarafından geliştirilen farklı wrapperlar aracılığıyla Perl ve Ruby programlama dilleri ile kolaylıkla OpenCV uygulamaları geliştirilebilir.

OpenCV kütüphanesi içerisinde görüntü işlemeye (image processing) ve makine öğrenmesine (machine learning) yönelik 2500'den fazla algoritma bulunmaktadır. Bu algoritmalar ile yüz tanıma, nesneleri ayırt etme, insan hareketlerini tespit edebilme, nesne sınıflandırma, plaka tanıma, üç boyutlu görüntü üzerinde işlem yapabilme, görüntü karşılaştırma, optik karakter tanımlama OCR (Optical Character Recognition) gibi işlemler rahatlıkla yapılabilmektedir.



3.2.1. OpenCV Bileşenleri

OpenCV kütüphanesini daha iyi anlamak için mimarisinden ve OpenCV'yi oluşturan bileşenlerden bahsetmek istiyorum.

- **Core:** OpenCV'nin temel fonksiyonları ve matris, point, size gibi veri yapılarını bulundurur. Ayrıca görüntü üzerine çizim yapabilmek için kullanılabilecek metotları ve XML işlemleri için gerekli bileşenleri barındırır.
- **HighGui:** Resim görüntüleme, pencereleri yönetme ve grafiksel kullanıcı arabirimleri için gerekli olabilecek metotları barındırır. 3.0 öncesi sürümlerde dosya sistemi üzerinden resim dosyası okuma ve yazma işlemlerini yerine getiren metotları barındırmaktaydı.
- **Imgproc:** Filtreleme operatörleri, kenar bulma, nesne belirleme, renk uzayı yönetimi, renk yönetimi ve eşikleme gibi neredeyse tüm fonksiyonları içine alan bir pakettir. 3 ve sonra sürümlerde bazı fonksiyonlar değişmiş olsada 2 ve 3 sürümünde de bir çok fonksiyon aynıdır.
- **Imgcodecs:** Dosya sistemi üzerinden resim ve video okuma/yazma işlemlerini yerine getiren metotları barındırmaktadır.
- **Videoio:** Kameralara ve video cihazlarına erişmek ve görüntü almak ve görüntü yazmak için gerekli metotları barındırır. OpenCV 3 sürümü öncesinde bu paketteki birçok metot video paketi içerisindeydi.

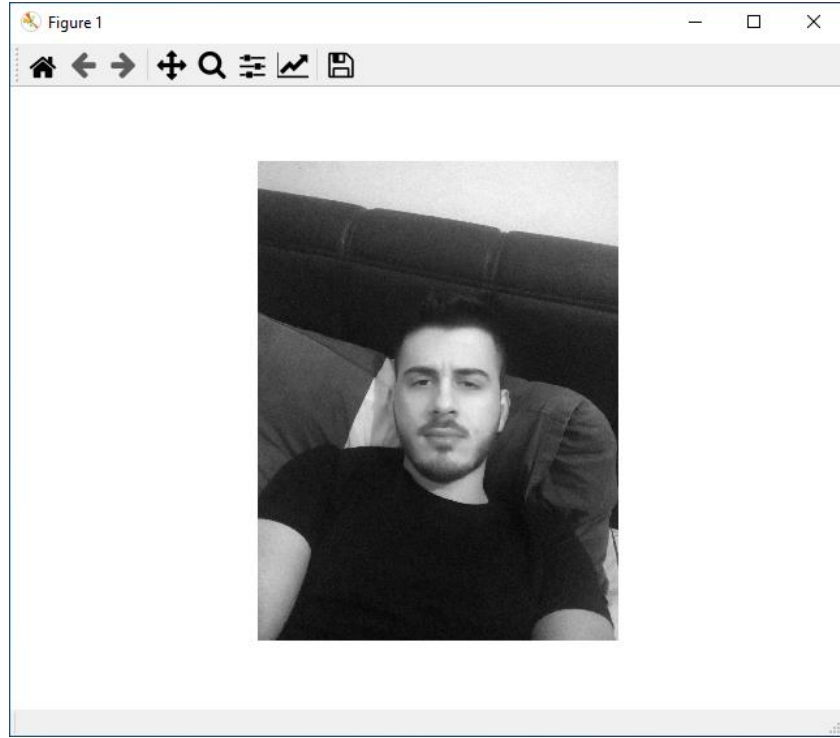
3.3. Alternatif Görüntü İşleme Kütüphaneleri

Görüntü işleme projelerinizde kullanacağınız kütüphaneyi amacınıza uygun olarak seçmeniz önemlidir. Bu seçimi yaparken ne yapmak istediğinize doğru karar vermelisiniz, örneğin sadece kameradan (usb, ip vs.) görüntü almak için projenize OpenCV entegre etmenize gerek olmayabilir. Bu gibi durumlar için ve OpenCV'nin neden iyi olduğunu anlayabilmek amacıyla alternatif olarak görüntü işleme kütüphanelerine de bakalım.

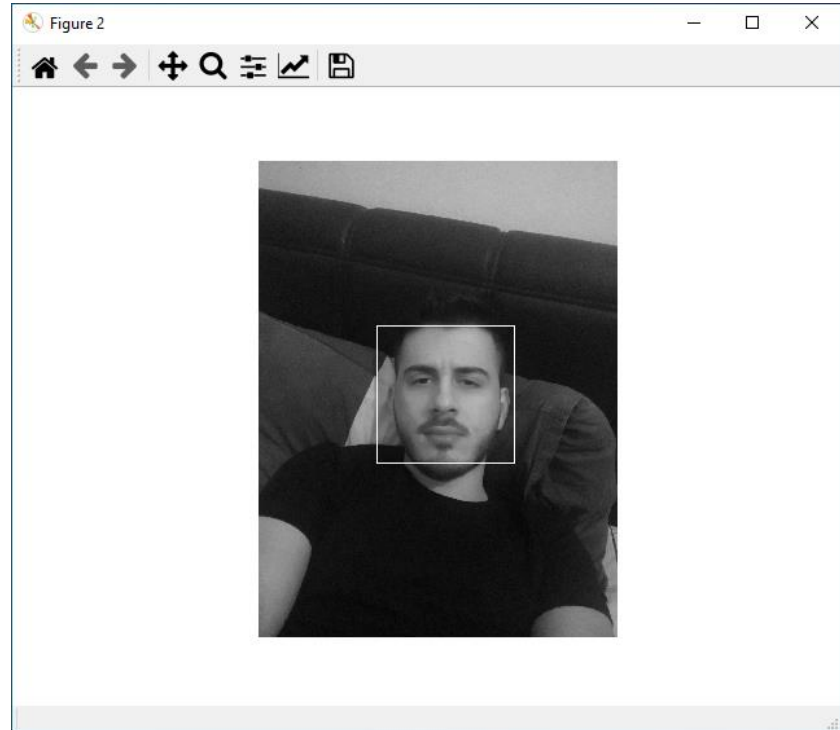
- **MATLAB:** Matlab için bir görüntü işleme kütüphanesi olarak bahsetmek doğru değildir fakat içerisinde görüntü işlemeye yönelik temel algoritmaları barındırmaktadır. Dördüncü nesil ve çok amaçlı bir programlama dilidir. Akademik araştırmalarınızda, performansın önemli olmadığı durumlarda temel görüntü işlemleri için tercih edebilirsiniz. Matlab kullanarak OpenCV Kütüphanesi ile etkileşimli olarak da uygulamalarda geliştirmek mümkündür.

- **Halcon:** Endüstriyel projeler için tercih edilen, kendi içerisinde geliştirme ortamının yanı sıra çeşitli programlama dilleri (C, C++, VS C++, C#, VB.NET) için kütüphanesi bulunan, yapay görme (machine vision) odaklı ticari bir yazılımdır. İçerisinde birçok hazır fonksiyon bulundurur bu sayede hızlı uygulamalar geliştirilebilir. OpenCV açık kaynak kodlu, ücretsiz bir kütüphanedir ve computer vision odaklıdır. Bu yönleri ile Halcon'dan ayrılmaktadır.
- **OpenFrameworks:** Açık kaynak olarak geliştirilen bu kütüphane C++ programlama dili için geliştirilen bu proje OS X, Linux, Embedded Linux (ARM), iOS, Android platformlarında çalışabilmektedir. OpenCV kütüphanesinin bir çok algoritmasını kullanır ve temel çıkış amacı kolay ve hızlı uygulama geliştirmektir. Örneğin OpenCV ile 2t sürede gerçekleştirdiğiniz bir işi 1t sürede gerçekleştirebilirsiniz, bunun temel sebebi ise bir çok fonksiyonu aracılığıyla standart hale getirilmiş olan işleri tek satır ile yapabilmesidir (Nesne tespiti, takibi renk belirleme, karşılaştırma vb.).
- **CIMG:** Açık kaynak kodlu bir görüntü işleme kütüphanesidir. Windows, Linux ve OS X platformu üzerinde çalışmaktadır. Sadece C++ dili için desteği bulunmaktadır fakat yazılmış wrapperlar ile Java ve Python ile de uygulama geliştirilebilmektedir. Birçok algoritmayı barındırmaktadır fakat OpenCV kadar performanslı ve geniş bir algoritma altyapısına sahip değildir.
- **Fiji:** Java platformu için geliştirilmiş açık kaynak kodlu GPL lisansına sahip bir görüntü işleme kütüphanesidir. Windows, Linux ve MAC OSX Intel 32-bit veya 64-bit üzerinde çalışır. Bilimsel görüntü analizi için geliştirilmiştir. Genetik, hücre biyolojisi, nöro-bilim gibi alanlar için özelleştirilmiş algoritmalara sahiptir.
- Endrov, ImageJ, Lead tools, Pink, Image Magick, Boost ise görüntü işleme kütüphanelerinden bazılarıdır.

4.İnsan Yüzü Tanıma Programı

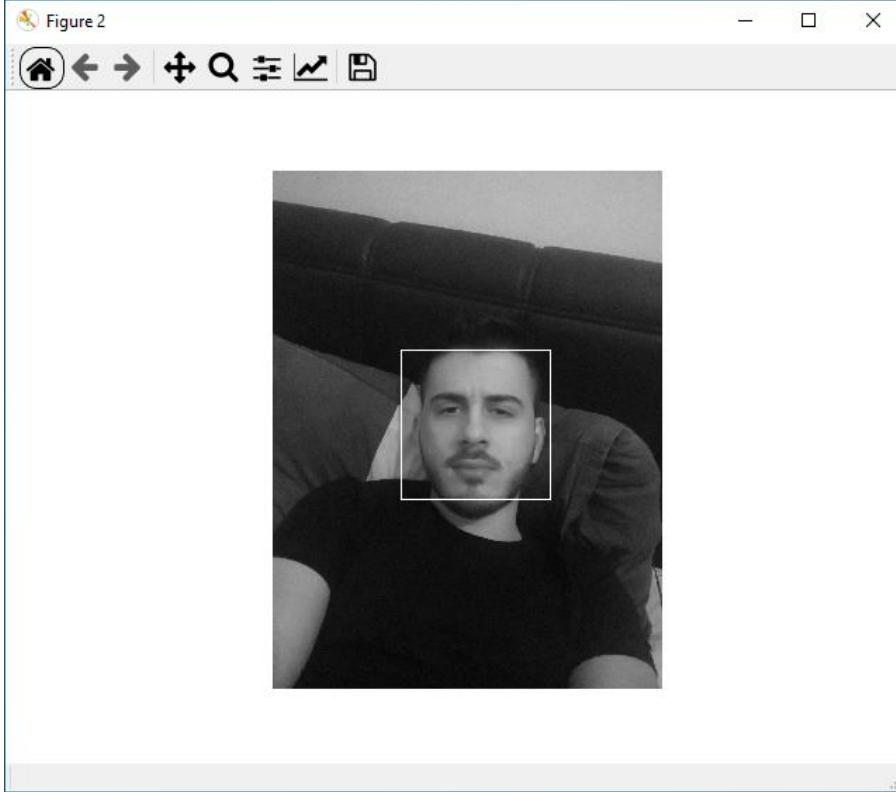


Ana Ekran

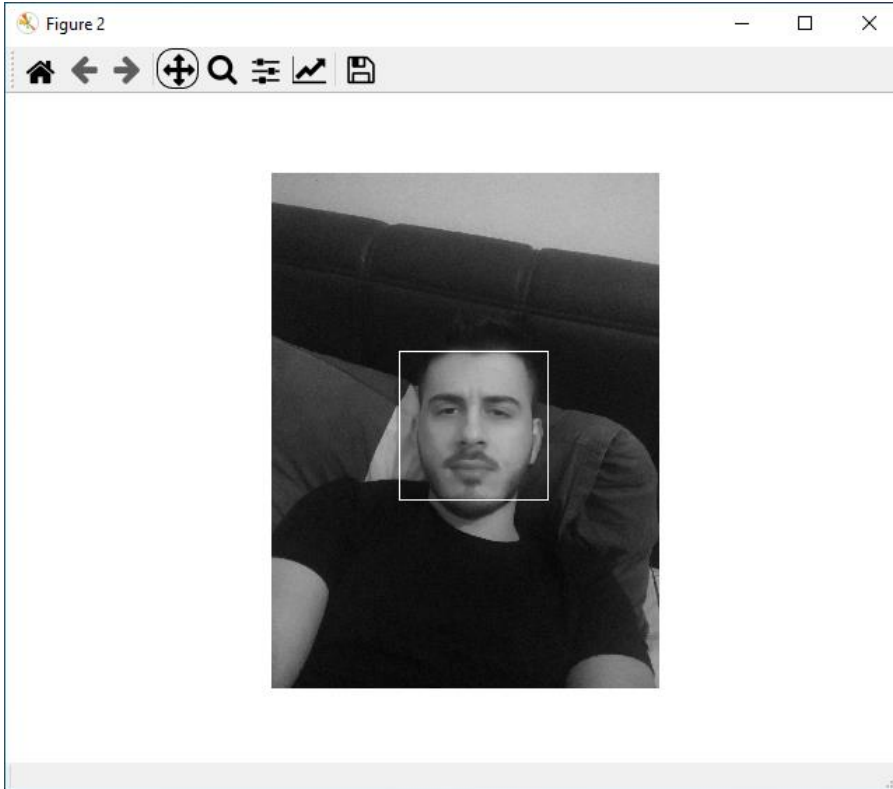


Yüz Tanınmış Ekran

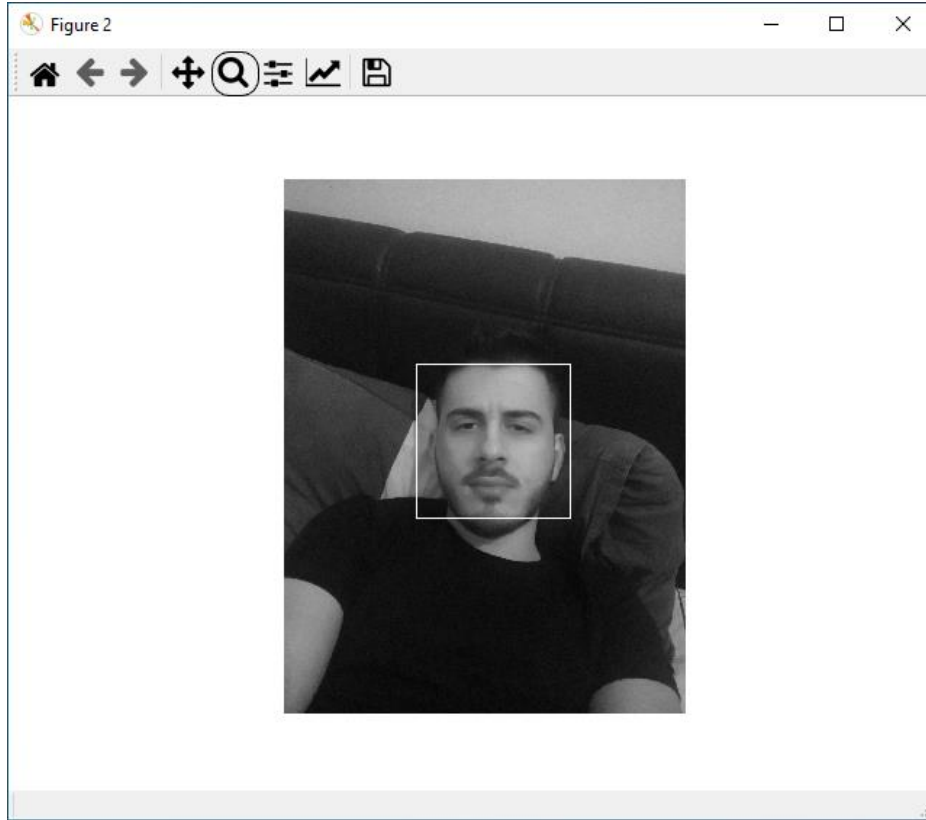
Programın Kullanımı



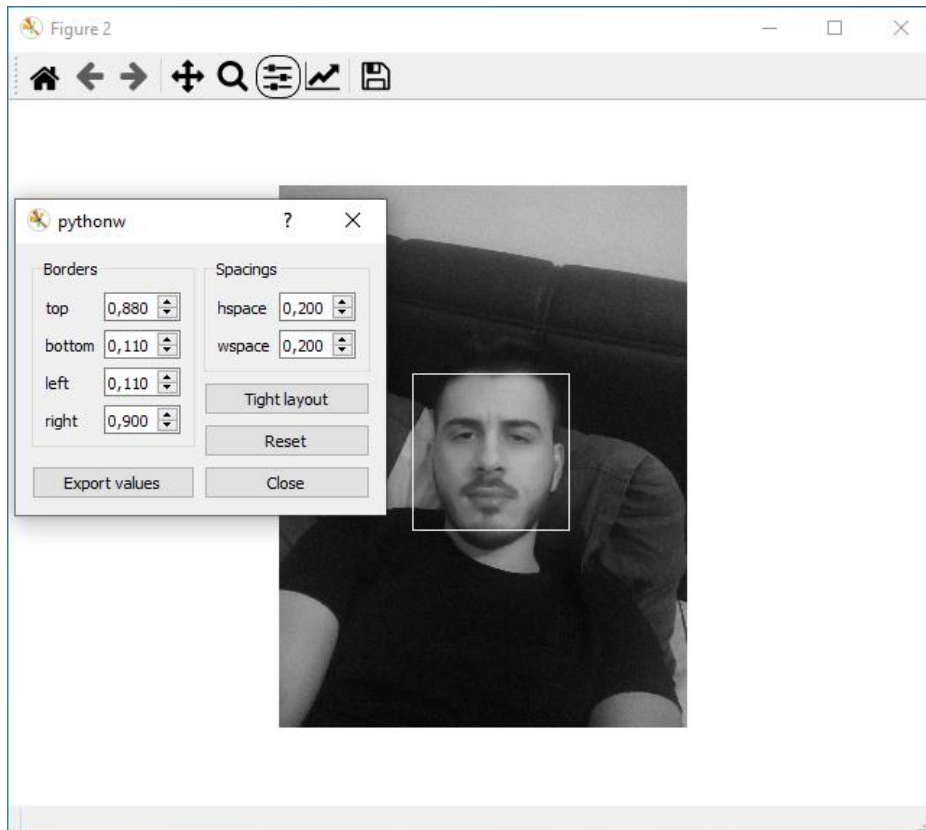
Orijinal görüntüye sıfırlar.



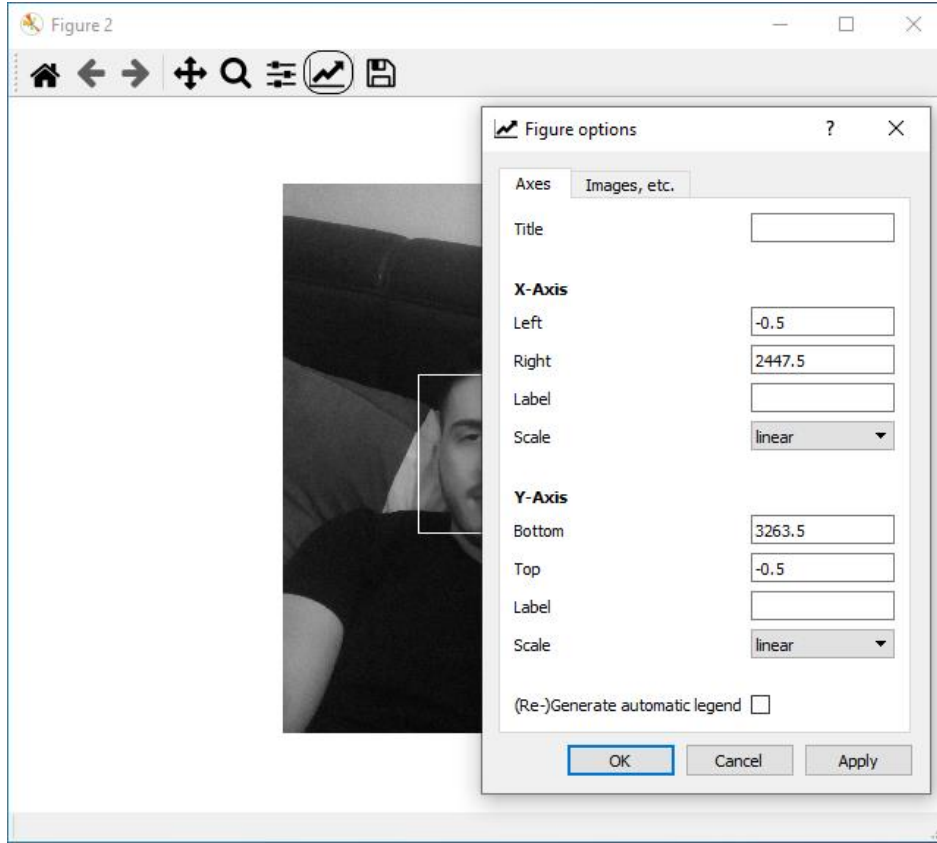
Fare'nin sol tuşuyla eksenleri kaydırır,sağ tuşuyla yakınlaştırır.



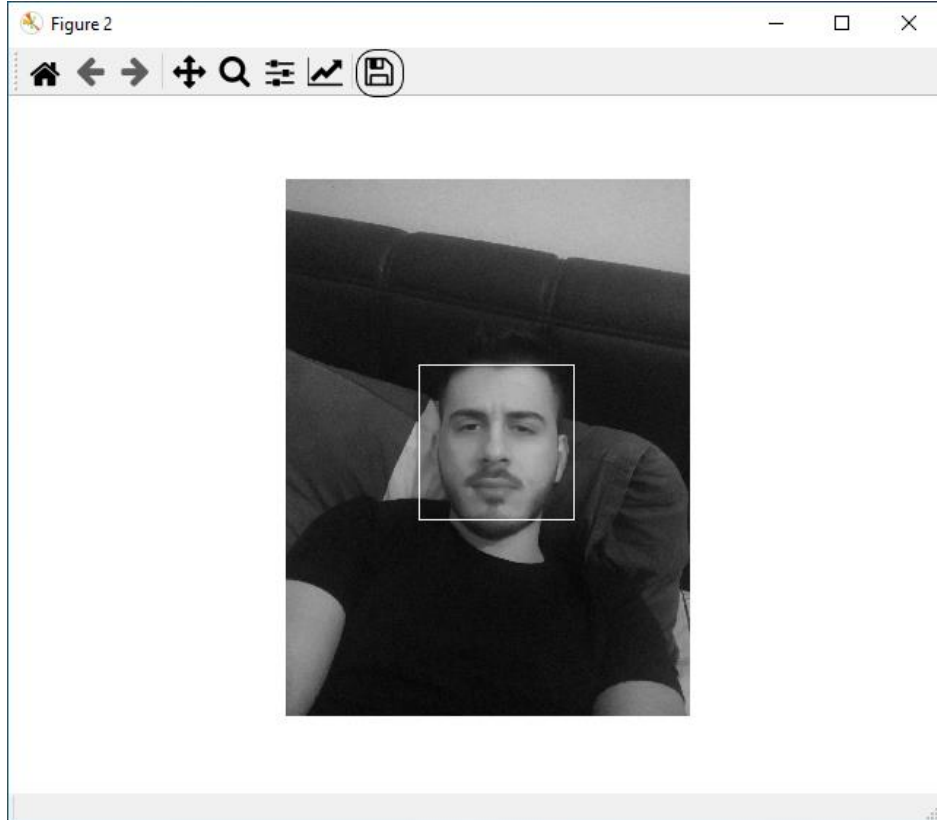
Farenin sol tuşuyla seçilen dikdörtgen alanı yakınlaştırır.



Alt noktaları yapılandırır.

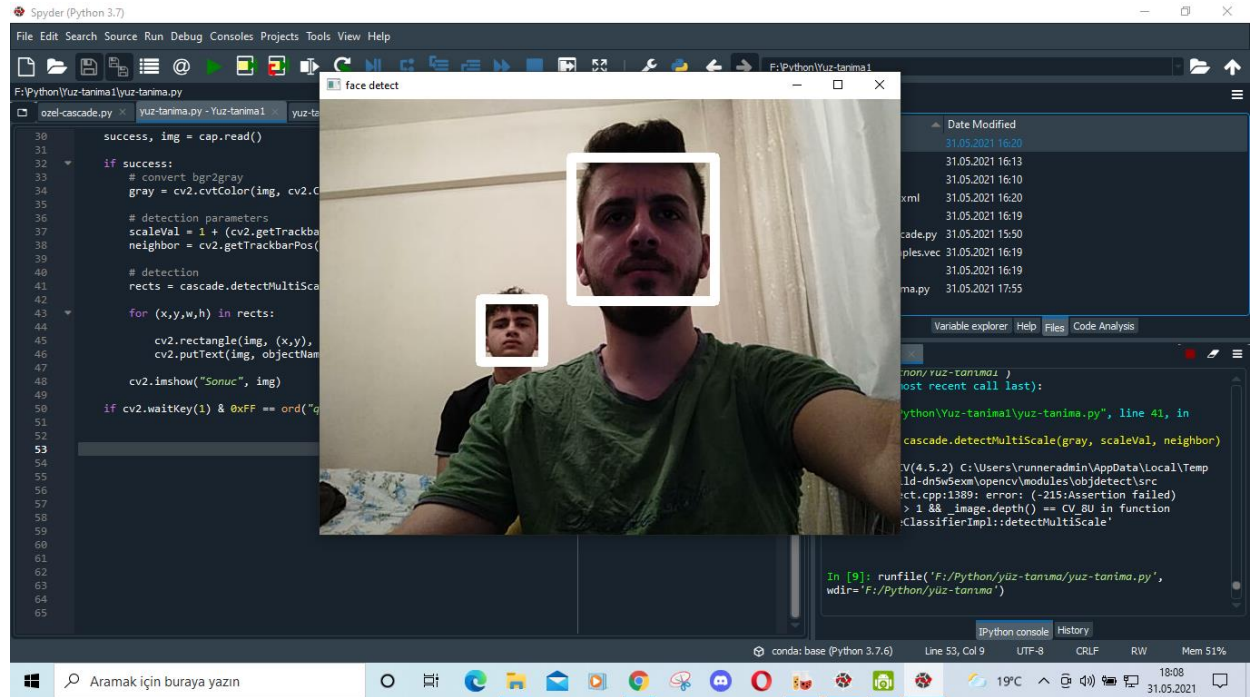


Eksen, eğri ve görüntü parametrelerini düzenler.



Görüntüyü kaydeder.

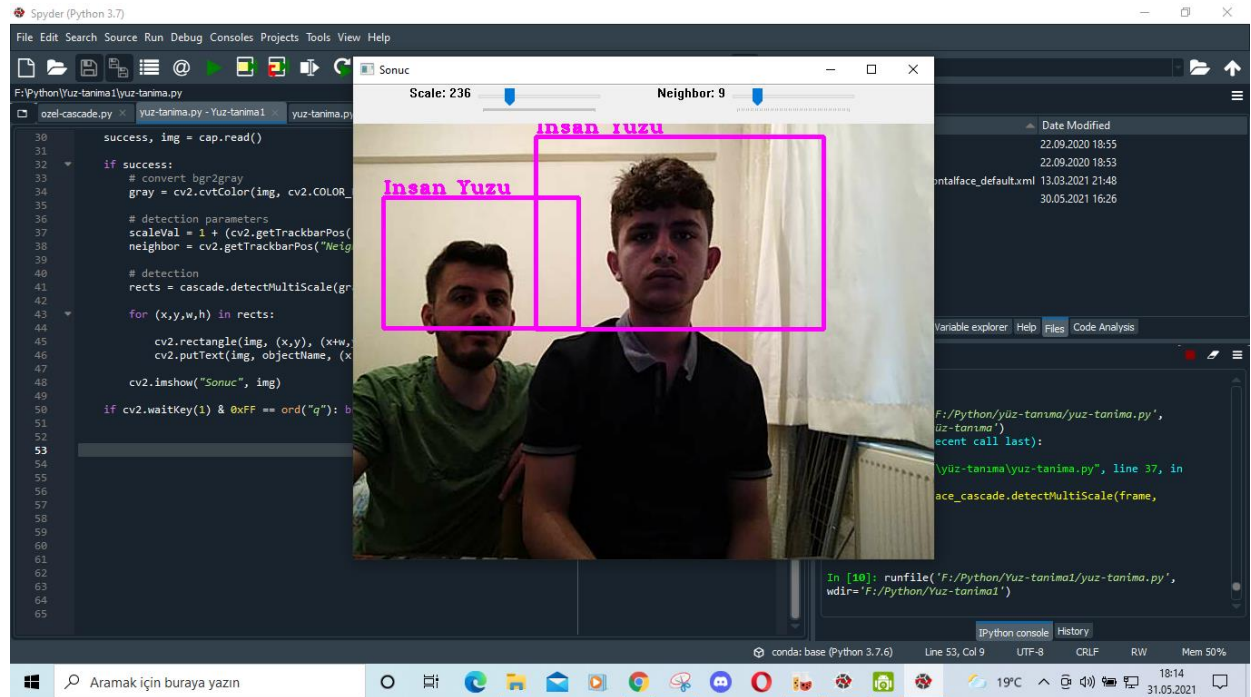
Hazır Cascade Kullanılarak Yapılan Yüz Tanıma



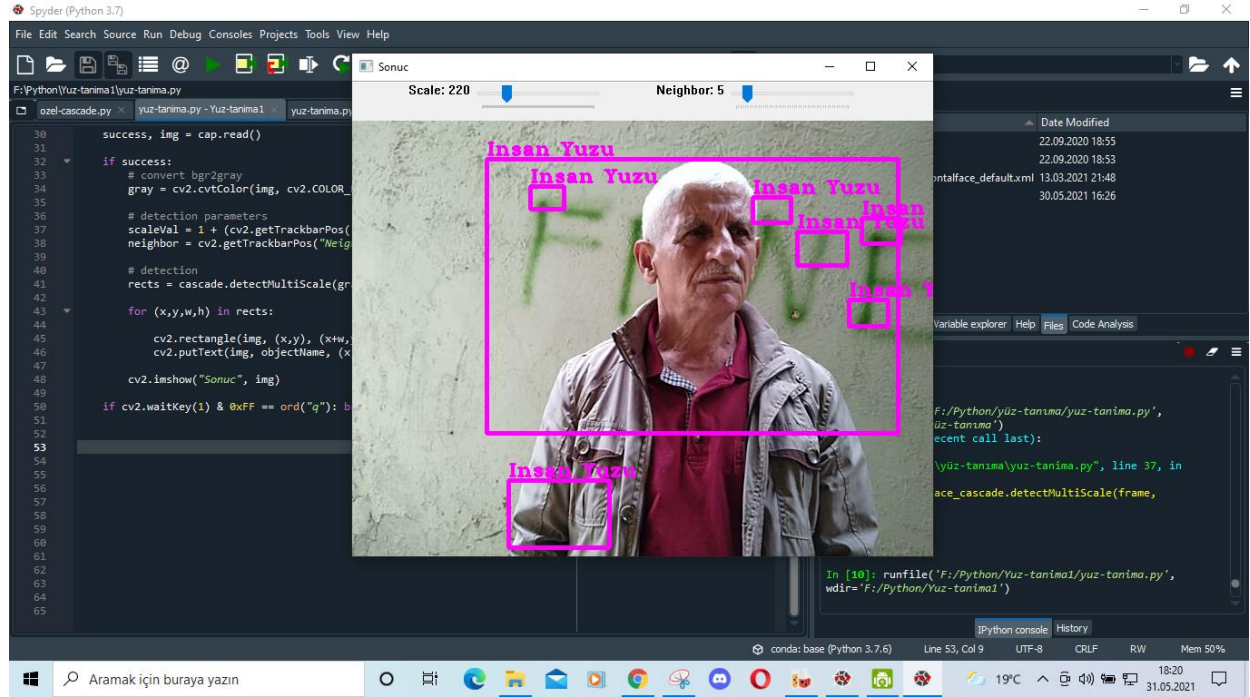
Hazır cascade ile gayet net bir şekilde yüzleri tanıyabiliyor. Bu görüntüde canlı video ile programı test etmiş bulunmaktayım.

Kendi Oluşturduğum Cascade İle Yapılan Yüz Tanıma

Topladığım pozitif ve negatif görüntülerle bir cascade oluşturdum. Bu sefer kendi oluşturduğum cascade'yi kullanarak programı yazdım. Aşağıda canlı video ile edilen test sonuçları bulunmaktadır.



Bazen tam bulurken bazen de tam bulamayabiliyor.



Burada görüldüğü gibi insan yüzünü bulsa da çerçeve istenenden büyük. Arka plandaki insan yüzü olmayan bir çok şeyi insan yüzüymüş gibi tanıyor. Bunun nedeni de pozitif ve negatif görüntü sayısının yeterli olmaması. Pozitif ve negatif görüntü sayılarını arttırdıkça programda gelişecek ve daha net sonuçlar elde edilebilecektir.

5.Sonuçlar ve Öneriler

Proje dahilinde geliştirilen İnsan Yüzü Tanıma Programı yazılımı kullanılarak bir çok deneme ve inceleme yapıldı. Sonuçta Viola-Jones algoritması ile yapılmaya karar verildi. Yine de bazı durumlarda insan yüzlerini tam algılayamıyabiliyordu. Bunu başlıca nedeni insanların kameraya tam dik bakma zorunluluğu ve gözlük vs. kullanılması durumunda tanıma için kullanılacak özelliklerin elde edilmesidir.

Sonuçları iyileştirebilmek için arka plan görüntüsü kullanılarak aydınlatma farklılıklarının ortadan kaldırılması kullanılabilir ancak CCD kamera kullanıldığında bu yöntem pek etkili olmuyor. Parlaklık için alternatif bir çift çözüm pratikte getirilebilir. Birincisi eğer güvenlik sisteminde kullanılırsa tanıma yapılan ortamın ışık seviyesi ve dağılımının sabit tutulmasıdır. Diğeri ise farklı aydınlatmalar altında kişilerin referans yüzlerinin kullanılmasıdır.

Diğer bir iyileştirme getirilebilecek nokta da yüzün yerinin bulunmasıdır. Eğer saçların yüzün yerinin bulunmasında sorun oluşturmayacak bir yöntem kullanılırsa yüz resimleri aynı boyutlarda olacağından daha yüksek başarımlar elde edilebilir. Bir ileri adım ise tanımda yüz ile birlikte saçlarında kullanılması yerine sadece kaş, göz, dudak, çeneyi içeren bir görüntü üzerinde çalışılmasıdır. Böylece kişinin saç şeklinin değiştirmesi sorun oluşturmaz.

Yapılan çalışmalar insan yüzü tanıma üzerine olsa da temel alınan PCA yöntemi, doku tanıma (ör: kilim deseni) veya üretim aşamasında kalite kontrolün görüntü ile yapılabilecek alanlarda da kullanılabilir.

6.EKLER

6.1. Görüntü İşleme Temel Kavramlar

6.1.1. Pikseller ve Pencere

Sayısal görüntüler $N \times M$ 'lik matrisler olarak ifade edilir ve bu matrisin her bir gözüne piksel adı verilir. N yükseklik, M de genişliktir. Gri tonlu resimlerde her piksel o noktanın aydınlanma değerini belirler.

	1	2	3	4	5	6
1	0	0	0	0	0	0	0	0
2	0	0	0	7	7	7	0	0
3	0	0	7	7	7	7	0	0
4	0	7	7	7	7	7	0	0
..	0	7	7	7	7	7	0	0
..	0	0	0	0	0	0	0	0

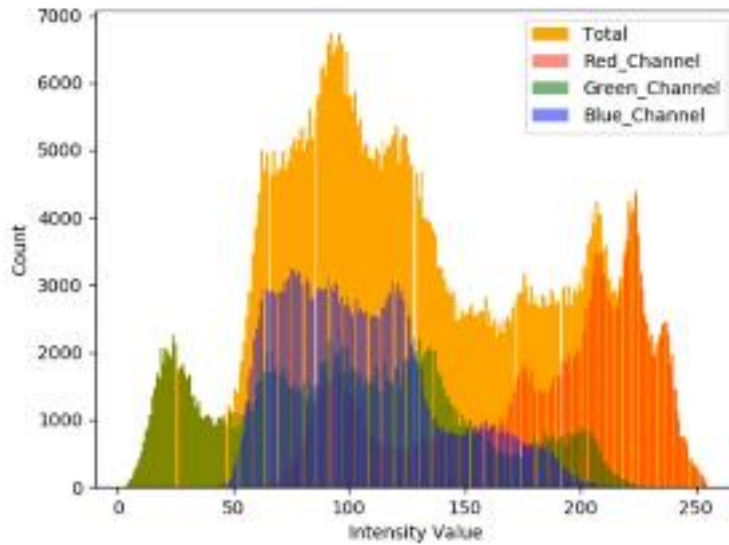
Bir resmin dikdörtgensel bir alt alanına pencere adı verilir. Yukarıda gri renkli olan alan bir pencereye örnektir.

6.1.2. Histogram

Histogram bir resimde kullanılan her parlaklık seviyesinin kullanım frekansının grafiksel olarak ifadesidir.

Histogram her grilik seviyesinden kullanılan frekans hakkında bilgi sağlar ancak piksellerin konumları hakkında herhangi bilgi taşımaz.

Bir şeklin döndürülmesinden o şeklin histogramı etkilenmez. Eğer değişiyorsa görüntünün çözünürlüğünün yetersiz olduğunu gösterir. Bir görüntünün tek bir histogramı vardır ancak bir hitogram tek bir görüntüye ait olmayabilir. Histogram resimler üzerinde çeşitli işlemler yapmaya imkan sağlar.



6.2. Temel Görüntü İşleme Operatörleri

6.2.1. Monadic: Tek nokta dönüşümleri

Monadic operatörler ile yapılan dönüşümlerden elde edilen sonuç görüntün pikselleri orijinal görüntünün aynı yerinde bulunan piksellerden birebir olarak elde edilir.

Identity

Sonuç görüntü başlangıç görüntünün tamamen aynı bir kopyasıdır.

Inverse

Çıktı olarak girdi olan görüntünün negatifini verir. Bu yapılırken her pikselin değeri maksimum grilik seviyesinden çıkarılarak yeni piksel hesaplanır.

Threshold

Belli bir değerden yüksek olan gri seviyelerini 1 diğerlerini 0 yapar.

Inverted threshold

Belli bir değerden yüksek olan gri seviyelerini 0 diğerlerini 1 yapar.

Binary Threshold

Belli değerler arasındaki gri seviyelerini 1 diğerlerini 0 yapar.

Inverted Binary Threshold

Belli değerler arasındaki gri seviyelerini 0 diğerlerini 1 yapar.

Gray Scale Threshold

Belli değerler arasındaki grilik seviyelerini korur diğerleri 0 yapar.

Inverted Gray Scale Threshold

Belli değerler arasındaki grilik seviyelerinin inversini alır diğerleri 0 yapar.

Stretch

Belli iki grilik seviyesini tüm grilik seviyeleri arsında yayar.

Gray Level Reduction

Grilik seviyelerin belli değerlere sahip olanları ortak değer atak böylece grilik seviyesini azaltmış olur.

6.2.2. Dyadic: Çift nokta dönüşümleri

Dyadic çift nokta operatörleri iki resmin aynı pozisyonlarında bulunan piksellerinin birebir işlenip üçüncü bir resim oluşturmalarıdır.

Resim Toplama

Resim toplama resimde bulunan gürültülerin azaltılmasını sağlamak için kullanılabilir.

Resim Çıkarma

Resim çıkarma bir zaman aralığında iki görüntü arasındaki değişikliğin belirlenmesi amacıyla birebir tanımda kullanılabilir. Resim çıkarma yolu ile görüntüde meydana gelen hareketler takip edilebilir.

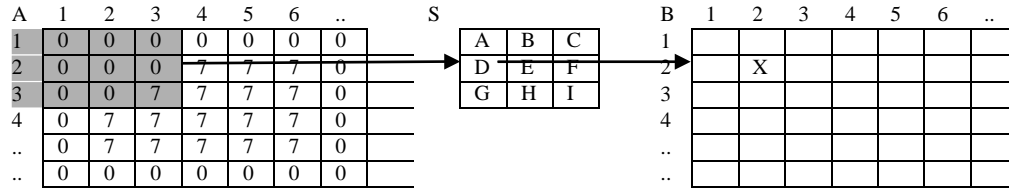
Resim Çarpma

Senörlerin lineer olmayan bir görüş alanını düzeltmek amacıyla kullanılır.

6.2.3. Konvolüsyon :Şablon dönüşümleri

Şablon filtrelerde elde edilecek yeni görüntü matrisi her pikselin komşuları ile bir fonksiyonu olarak elde edilebilir. Kullanılacak şablonun boyu tek sayı seçilir.

Örnek olarak 3x3 bir pencereye uygulanan bir filtreyi ele alırsak bu şablon görüntü matrisi aşağıdaki şekilde hesaplanır.



$$3 \times 3 \Rightarrow N=3$$

$$B_{i,j} = \sum_{k=-1..1} \sum_{l=-1..1} A_{i+k,j+l} * S_{k,l}$$

Şablon filtreler kullanıldığında şeklin etrafında N/2 kadar kullanılmayan alan oluşur, eğer şeklin son boyutunu belli bir değerde olması isteniyorsa alınacak resmin dışında sabit değerler bulunduğu varsayılabilir.

6.3. Görüntü İyileştirme

6.3.1. Alçak geçiren filtre

Alçak geçiren filtreler görüntülerde bulunan ani ton değişikliklerini eler ve yumuşak geçişleri etkilemez. Burada ani değişiklikler gösteren alanlar gürültü ise bu gürültünün azaltılması için kullanılabilir. Yukarıdaki şekilde Alçak geçiren filtrenin çalışması gösterilmiştir. Yüksek frekanslı bileşenleri eler ve sadece düşük frekanslı bileşenler resmi oluşturur. Alçak geçiren filtreler uygulandığında resimlerde bulanıklık oluşur ve bunun derecesi kullanılacak şablona bağlı olarak değişir.

S

A	B	C
D	E	F
G	H	I

Alçak geçiren filtreler için iki temel kural vardır.

- Şablonun tüm katsayıları pozitif olmalı
- Tüm katsayıların toplamı bir olmalı.

6.3.2. Yüksek geçiren filtre

Yüksek geçiren filtreler alçak geçiren filtrelerin ters karakteristiklerine sahiptir. Filtre yüksek frekanslı bileşenleri değiştirmeyecek ancak alçak frekanslı olanları eleyecektir.

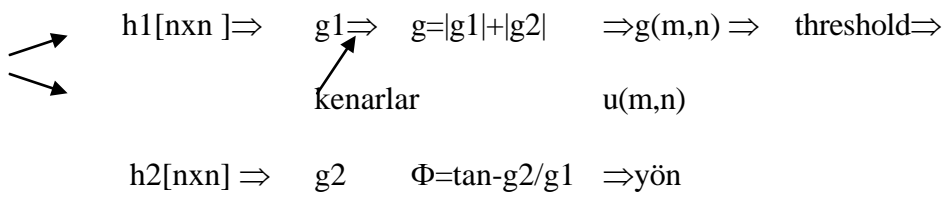
Yüksek geçiren filtreler için iki temel kural vardır.

- Katsayıları pozitif veya negatif olabilir.
- Tüm katsayıların toplamı sıfır olmalı.

Yüksek geçiren filtreler resimlerde bulunan ani geçişleri vurgulaması nedeniyle resmin özelliklerinin çıkarılmasında kullanılır. En çok kenar ve çizgi belirlemede kullanılır.

6.3.3. Kenar belirleme operatörleri

Gradient Operatorleri



Robert's
H1

S	1	2
1	0	1

2	-1	0
---	----	---

H1

S	1	2
---	---	---

Prewitt

H1

S	1	2	3
1	-1	0	1
2	-1	0	1
3	-1	0	1

1	1	0
2	0	-1

H2

S	1	2	3
1	-1	-1	-1
2	0	0	0

Sobel

H1

S	1	2	3
1	-1	0	1
2	-2	0	2
3	-1	0	1

H2

S	1	2	3
1	-1	-2	-1
2	0	0	0

Isotropic

H1

S	1	2	3
1	-1	0	1
2	$-\sqrt{2}$	0	$\sqrt{2}$
3	-1	0	1

H2

S	1	2	3
1	-1	0	1
2	$-\sqrt{2}$	0	$\sqrt{2}$

Laplacian Gradient

S	1	2	3
1	0	-1	0
2	-1	4	-1
3	0	-1	0

S	1	2	3
1	-1	-1	-1
2	-1	8	-1
3	-1	-1	-1

S	1	2	3
1	1	-2	1
2	-2	4	-2
3	1	-2	1

Pusula Operatörleri

$U(m,n) \Rightarrow h_k[n \times n] \Rightarrow g_k(m,n) \Rightarrow g(m,n) = \max_{k=1..8} \{|g_k(m,n)|\} \Rightarrow$
threshold \Rightarrow kenarlar

Prewitt-a

S	1	2	3
1	1	1	1
2	1	-2	1
3	-1	-1	-1

S	1	2	3
1	1	1	1
2	-1	-2	1
3	-1	-1	1

S	1	2	3
1	-1	1	1
2	-1	-2	1
3	-1	1	1

S	1	2	3
1	-1	-1	1
2	-1	-2	1
3	1	1	1

S	1	2	3
1	-1	-1	-1
2	1	-2	1
3	1	1	1

S	1	2	3
1	1	-1	-1
2	1	-2	-1
3	1	1	1

S	1	2	3
1	1	1	-1
2	1	-2	-1
3	1	1	-1

S	1	2	3
1	1	1	1
2	1	-2	-1
3	1	-1	-1

Prewitt-b

S	1	2	3
1	1	1	1
2	0	0	0
3	-1	-1	-1

S	1	2	3
1	0	1	1
2	-1	0	1
3	-1	-1	0

S	1	2	3
1	-1	0	1
2	-1	0	1
3	-1	0	1

S	1	2	3
1	-1	-1	0
2	-1	0	1
3	0	1	1

Sobel

S	1	2	3
1	1	2	1
2	0	0	0
3	-1	-2	-1

S	1	2	3
1	0	1	2
2	-1	0	1
3	-2	-1	0

S	1	2	3
1	-1	0	1
2	-2	0	2
3	-1	0	1

S	1	2	3
1	-2	-1	0
2	-1	0	1
3	0	1	2

Kirsh

S	1	2	3
1	-3	-3	-3
2	-3	-2	-3
3	5	5	5

S	1	2	3
1	-3	-3	-3
2	5	-2	-3
3	5	5	-3

S	1	2	3
1	5	-3	-3
2	5	-2	-3
3	5	-3	-3

S	1	2	3
1	5	5	-3
2	5	-2	-3
3	-3	-3	-3

S	1	2	3
1	5	5	5
2	-3	-2	-3
3	-3	-3	-3

S	1	2	3
1	-3	5	5
2	-3	-2	5
3	-3	-3	-3

S	1	2	3
1	-3	-3	5
2	-3	-2	5
3	-3	-3	5

S	1	2	3
1	-3	-3	-3
2	-3	-2	5
3	-3	5	5

6.3.4. Medyan Filtre

Medyan filtre genelde resimlerde bulunan gürültünün temizlenmesi için kullanılır. Özellikle CCD kameralar ile alınan görüntülerin gürültüleri bu filtre ile temizlenir.

Mantık aynı olmasına rağmen medyan filtre ya dizi halinde ya da pencere halinde uygulanır. Her iki yöntemde de var olan değerlerin medyanı orta piksele yerleştirilir.

A	1	2	3	4	5	6	..
1	3	5	1	2	0	0	0
2	5	0	2	7	7	7	0
3	1	2	7	7	7	7	0
4	2	7	7	7	7	7	0
..	0	7	7	7	7	7	0
..	0	0	0	0	0	0	0

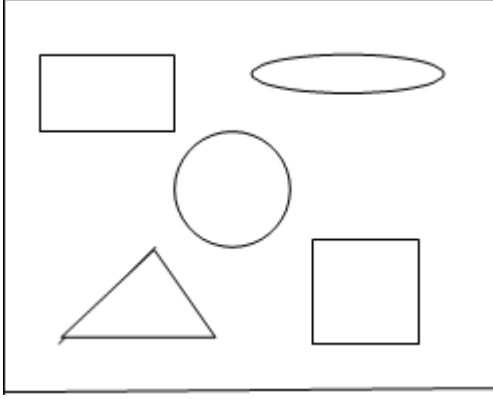
0,1,1, 2, 2,3,5,5,7 \Rightarrow 2

6.4. Görüntü Tanıma Teknikleri ve Örnekleri

Görüntü analizinde en büyük problem, görüntüde oluşan değişiklikleri ya da görüntüde nesnenin olup olmadığını fark edebilmektir. Bu tür problemlere örnek: uydu görüntülerinden yapılan hava tahminleri, tıbbi görüntülerde hastalığın teşhisi ve radar görüntülerinde hedefin belirlenmesidir.

Şablon Uyumu:

Nesne tanıma yöntemlerinden biridir. Bulunmak istenen nesne daha önceden belirlenen şablon ile karşılaştırılır. Eğer tamamen aynılarsa aranan nesne bulunmuştur ve şablon nesnesi ile aynı isimle adlandırılır.



Burada amacımız nesne kümesi içinde üçgenin var olup olmadığının ve eğer varsa hangi nesnenin üçgen olduğunun bulunmasıdır. Bu yüzden şablon, nesne kümesindeki nesnelerle tek tek karşılaştırılır.

Eğer alınan görüntü net değilse, parazitler varsa şablona uygun olan nesnenin bulunması kesin değildir.

$-M \leq m \leq M$ ve $-N \leq n \leq N$ olmak üzere, şablon ve nesne arasında $D(m,n)$ farkı ölçülür. Belirlenen $L_d(m,n)$ seviyesinden küçük olan bir ölçüm yapıldığında sonuç bulunmuş olur.

Üçgen Şablon

Genel olarak fark ölçümü: 2

$$D(m,n) = \sum_{j=-M}^M \sum_{k=-N}^N [F(j,k) - T(j-m,k-n)]$$

j k

Burada $F(j,k)$ araştırılan görüntü alanını, $T(j,k)$ ise şablonu ifade etmektedir. Eşitlik genişletilirse:

$$D(m,n) = D_1(m,n) - 2D_2(m,n) + D_3(m,n)$$

2

$$D_1(m,n) = \sum_{j=-M}^M \sum_{k=-N}^N [F(j,k)]$$

j k

$$D_2(m,n) = \sum_{j=-M}^M \sum_{k=-N}^N [F(j,k) * T(j-m,k-n)]$$

j k

2

$$D_3(m,n) = \sum_{j=-M}^M \sum_{k=-N}^N [T(j-m,k-n)]$$

j k

Burada $D_3(m,n)$ toplam şablon enerjisini ifade etmektedir ve bu değer (m,n) koordinatından bağımsızdır.

Kullanılan karşılıklı ilişki değeri maksimum alınırsa, nesne ile şablonu karşılaştırma da elde edilen fark küçük olmaktadır.

Şablonun karşılaştırılacağı görüntü alanındaki nesneler rotasyon yaparak değişikliğe uğramışsa alınan sonuç iyi olmamaktadır.

Direkt Arama Metodları:

Nesne, araştırılan bölgeye göre küçük ise kullanılan bir yöntemdir.

İki - Boyutlu Logaritmik Arama:

Bu yöntemle $n \times n$ 'lik alanda, araştırma sayısı yaklaşık olarak $\log n$ değerine indirilmektedir.

$M \times N$

$$D(i,j) = 1/MN \prod_{m=1}^M \prod_{n=1}^N f(v(m,n) - u(m+i,n+j)), \quad -p \leq i, j \leq p$$

$m=1 \quad n=1$

Burada $f(x)$, x 'in pozitif ve artan fonksiyonudur. $u(m,n)$, $m \times n$ 'lik şablon ve $v(m,n)$ de gözlenen görüntüdür.

Sıralı Arama:

Aramayı hızlandırmanın bir yolu da artan hatayı hesaplamaktır.

$p \quad q$

$$ep, q(i,j) = \prod_{m=1}^p \prod_{n=1}^q |v(m,n) - u(m+i,n+j)|, \quad p \leq M, q \leq N$$

$m=1 \quad n=1$

$ep, q(i,j)$ değerleri belirlenen eşik değerinden küçük olduğu sürece arama devam eder.

Diğer bir yöntem de alternating conjugate directions olarak bilinmektedir. İ doğruğultusunda minimum değer bulunana kadar arama yapılır, minimum değer bulunduğunda j doğruğultusunda arama yapılmaya başlanmaktadır. Bulunan minimum değer, yer değışmeden kaldığında sonuca ulaşılmış demektir.

Hiyerarşik Arama:

Eğer gözlenen görüntü çok büyükse ilk olarak küçültölmüş şablon kullanılmaktadır. Bu durumda birden fazla çakışma oluşursa, arama alanını küçöltmek için büyötlölmüş şablon uygulanmaktadır.

Görüntü Bölömleme:

Görüntünün aynı özellikleri taşıyan küçük parçalara bölömlenmesidir. Renksiz görüntülerde parlaklık değeri, renkli görüntülerde renk değeri bölömleme için en belirgin özelliktir. Ayrıca görüntünün kenarları ve yapısı da belirleyici özelliklerdir. Bölömlleyici görüntüyü sadece alt parçalara böler, bölömleri tanımaya veya aralarındaki ilişkiyi çıkarmaya çalışmaz. Görüntü bölömlemesi için belirgin bir teori yoktur. Bu yüzden tek bir standart method kullanılmamaktadır. Fakat popüler olmuş bazı methodlar vardır. Bunlardan Haralick ve Shapino, iyi bir görüntü bölömlemesi için şunları belirlemiştir: Görüntü bölömlemenin bölgeleri tek ve homojen olmalıdır, mesela gri tona göre. Bölge basit olmalıdır ve küçük delikler bulunmamalıdır. Karakteristiklerine göre komşu bölgeler farklı değeriye sahip olmalıdır. Her bir bölömlün sınırı basit, kesiksiz ve belirgin olmalıdır.

Genlik Bölömlleme Methodları:

Tek Seviyeli Parlaklık Eşığı:

Birçok görüntüde, arka plan farklı olmak üzere, arka plan ile nesnenin parlaklığı tek çeşittir. Mesela el yazması ya da bilgisayarla yazılmış textlerde. Bu tür görüntülerde, nesneyi arka plandan ayırabilmek için kullanılan ayırt edici özellik parlaklık değeriştir. Eğer siyah bir arka plan üzerinde beyaz bir nesne varsa, orta-gri eşik değeri uygulanarak, nesne arka plandan ayırt edilebilmektedir. Fakat bazı problemler ortaya çıkabilir: Gözlenen görüntü net olmayabilir. Hem nesne hem de arka plan gri renk seviyelerine sahip olabilmektedir.

Eşik değeriini belirlemek çok önemlidir. Bu değeri çok yüksek tutulursa, nesne belirgin hale gelir, ama koyuluklardan bazı özellikleri kaybedilebilmektedir. Eğer çok az tutulursa, bu seferde nesne belirsizleşir ve bazı kısımları görölemmez olur.

Parlaklık eşik değeriini belirlemek için önerilen bir yöntem şöyledir: Artan gri ölçü değeri sayılır ve önceki dağılım tahminiyle karşılaştırılır. Mesela bilinir ki bilgisayarla yazılan sayfada siyah karakterler sayfanın %25'ini kaplamaktadır. Bu yüzden düşük parlaklık ile piksellerin dörtte birine siyahın hükmetmesi sağlanmalıdır.

Diğer bir metot ta: Histogramda iki yükseklik arasındaki min değeri eşik değeri olarak almaktır. Fakat minimum değere karar vermek zordur. Bu nedenle Laplacian mask kullanılarak, histogram parlaklık eşik değerini seçebilecek uygunluğa getirilebilmektedir.

Çok Seviyeli Parlaklık Eşiği:

Recursive bir yöntemdir. İlk olarak görüntüye parlak bölgeleri karanlık bölgelerden ayıran eşik değeri uygulanmaktadır. Bu histogramda parlaklık mode'ları arasındaki minimum değerdir. Her parçalanmış kısımda histogram gözlenir. Eğer histogram tek parça değilse, parçalara tekrar eşik uygulanmaktadır. Histogram parçası tek kalana kadar bu yöntem sürdürülmektedir.

Çok Seviyeli Renkli Eleman Eşiği:

Parçalama, renklere (RGB), tristimulus değerlerine (YIQ) ya da keskinlik, koyuluk ve renk tonlarına göre yapılmaktadır. Eşik değerine karar verildikten sonra görüntü alt parçalara bölünmektedir. İşlemler histogram sonucu tek parça olana kadar ya da parçalama istenen ayırma ulaşana kadar sürdürülmektedir.

Genlik İzdüşümü:

Görüntünün ortalama genlik izdüşümü ile, satır ve sütunu boyunca görüntü parçalara ayrılabilir. Belirlenen satır ve sütun izdüşümleri şöyledir:

$$H(k)=1/N \sum_{m=1}^N F(j,k)$$
$$V(j)=1/N \sum_{k=1}^N F(j,k)$$

Eleman Sınıflaması:

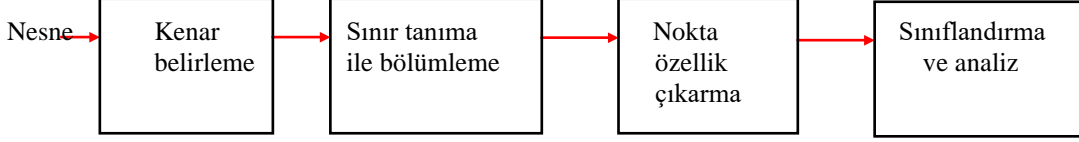
Bu metot la, piksellerin komşularıyla birleştikleri yerler araştırılır ve biri birine bağlı kümeler sınıflandırılır. Kullanılan yöntemlerden biri Piksel Sınıflamasıdır. Bu yöntemde görüntü soldan sağa, yukardan aşağı taranır. X pikseli nesneye dahilse 1, boşluğa dahilse 0 diye sınıflandırılmaktadır. Bu işlem X'in A,B,C ve D komşularıyla olan bağlantılarını araştırarak yapılmaktadır. Eğer X=1 ise, X nesneye dahildir. Eğer iki ya da daha çok sınırlı nesneler varsa bunlar eşit kabul edilir ve merge yapılırlar. 0'dan 1'e değişim görüldüğünde yeni nesne sınıflandırması yapılmaktadır. Bir piksel sınıflandırıldığında, o nesnenin özellikleri update edilmektedir.

C	A	D
B	X	

Sınır Tabanlı Yaklaşım:

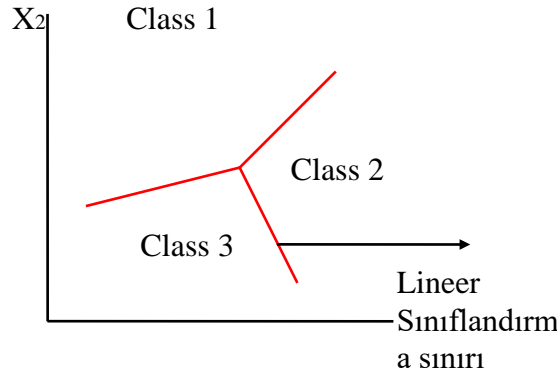
Kenar belirlemenin genel özellikleri kullanılarak bir görüntü bölümlenebilir.

Farklı özellikler için farklı kenar belirlemeler kullanılır. Görüntü parazitliyse veya iki bölge arasında çok küçük bölge eleman farklılıkları varsa, bulunan sınır kesiklidir. Bu kesiklikler, kenar bağlama teknikleriyle düzeltilir.



Kümeleyen Bölümleme Metotları:

Görüntünün herhangi bir (j,k) koordinatına göre $X=[X_1, X_2, \dots, X_N]^T$ vektörü oluşturulur. Bu vektör belirli özelliklere göre bulunan değerlerden oluşur. Komşuluk ölçüsü, nokta renk parçaları gibi. Bu oluşturulan vektöre göre bölümleme yapılır.



Bölge Bölümleme Metodu:

Bölge Büyümesi : Aynı genişliğe sahip komşu pikseller bir araya gruplanır ve bölgeler oluşturulur. Bölgeler oluşturulurken bağlantı sayıları da göz önüne alınır.

Ayrırma ve Birleştirme : Orijinal görüntü bölümü tek özellikten oluşmuyorsa, bu sahip olduğu özelliklere göre parçalara ayrılır. Ayrılan parçalardan komşu olanlar benzer özelliklere sahipse birleştirilirler.

Şablon Uyumu : Bölümlenecek görüntü şablon listesi ile karşılaştırılır. Bulunan nesneler bölümlenir ve geri kalan görüntü diğer tekniklerle analiz edilir.

Dokuma Bölümlemesi : Görüntüdeki nesneler arka plan ile yapılandırıldığında kullanılan bir yöntemdir. Eğer dokuma filtrelenmezse, yüksek kenar yoğunluğuna sahip olduğundan sınır tabanlı yaklaşım yöntemi kullanılamaz. Kümeleme ve bölge tabanlı yaklaşım yöntemleri dokuma bölgelerini bölümlemek için kullanılır. Önceden varolan dokumalar hakkında bilgi sahibi olmak çözümlerde kolaylık sağlar.

Sınıflandırma Teknikleri:

Sınıflama ve bölümleme birbirleriyle yakından ilgili yöntemlerdir. Biri uygulanınca diğeri onu takip eder. Görüntüde piksellerin sınıflandırılması diğeri bir eleman isimlendirmesidir ki bu durum görüntüde değişik nesnelerin bölümlendirilmesiyle ortaya çıkar.

a.Denetleyerek Öğrenme (Sınıflama) : Öğretme ile öğrenir. Hata yapılmaz. Ama eğitim zamanı uzundur. İki çeşittir :

Serbest Dağılım : Önceki tahmini dağılım fonksiyonlarının bilinmesine gerek yoktur. Nedenlere bağlıdır. Tek bölgeyi sıra ile N özellikteki parçalara ayırır. Her bir sınıf karara varana kadar test edilmez. Sınıf sayısı fazla olduğunda bu bir avantaj sağlar.

İstatistiksel Sınıflama : Tahmini dağılım modeline bağlıdır. Bu yöntem ile, özellik vektörünün yanlış sınıfa konma riski ve ortalama kaybı minimize edilir.

b.Denetlemeden Öğrenme ve Kümeleme : Özellik boşluğunda kümeleri ya da doğal grupları tanımaya çalışır. Kendi Kendine öğrenir. Başboş bırakılırsa paraziti bile tanımaya uğraşır. Tecrübelerden yararlanıp, elindeki örneklerle bakarak, kendi kendine karar verir.

7.Kaynaklar

- <https://www.udemy.com/course/derin-ogrenme-ile-goruntu-isleme-python-opencv-ve-keras/>
- https://docs.opencv.org/master/d6/d00/tutorial_py_root.html
- <https://arxiv.org/abs/1512.03385>
- <https://www.learnopencv.com/histogram-of-oriented-gradients/>
- <https://autonom.medium.com/bilgisayar%C4%B1-g%C3%B6r%C3%BCm-computer-vision-nedir-f319fec18700>
- *Face Detection System On AdaBoost Algorithm Using Haar Classifiers M.Gopi Krishna, A. Srinivasulu 2012*
- *Face Recognition with GNU Octave/MATLAB Philipp Wagner 2012*
- *Haar-like Features and Integral Image Representation David Gerónimo 2009*
- <https://medium.com/patron-ai/viola-jones-algoritmas%C4%B1-ile-y%C3%BCz-tespiti-t%C3%BCrk%C3%A7e-38ea73c910e3>
- <https://realpython.com/traditional-face-detection-python/>
- <https://mesutpiskin.com/blog/opencv-nedir.html>