



**GÖRÜNTÜ İŞLEME VE BİLGİSAYARLI GÖRÜ TEKNOLOJİLERİYLE
EŞGAL TESPİT VE TANIMA SİSTEMİ ALGORİTMALARI
GELİŞTİRİLMESİ**

Sezayi YEŞİLBAĞ

**YÜKSEK LİSANS TEZİ
ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ ANA BİLİM DALI**

**GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

HAZİRAN 2022

ETİK BEYAN

Gazi Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmada;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
- Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
- Tez çalışmada yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
- Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- Bu tezde sunduğum çalışmanın özgün olduğunu,

bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

Sezayi YEŞİLBAĞ

21/06/2022

GÖRÜNTÜ İŞLEME VE BİLGİSAYARLI GÖRÜ TEKNOLOJİLERİYLE EŞGAL TESPİT VE TANIMA SİSTEMİ ALGORİTMALARI GELİŞTİRİLMESİ

(Yüksek Lisans Tezi)

Sezayi YEŞİLBAĞ

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Haziran 2022

ÖZET

Yüz tanıma ve özellikle maskeli yüz tanıma, günümüzde güvenlik açığını önlemek için sıkça kullanılmaktadır. Okullarda, hastanelerde, alışveriş merkezlerinde, askeri alanlarda ya da herhangi kontrollü alanlarda personel takibi ve suçlu insanların tespitinde son derece kullanışlı bir sistemdir. Görüntü işleme ve bilgisayarlı görü teknolojilerinin gelişmesi ile nesne ve yüz tanıma yapmak için çeşitli yöntemler türetilmiştir. Bu çalışmada, daha başarılı sonuç veren, gri görüntü tabanlı çalıştığı için daha hızlı bir metot olan LBPH (Local Binary Patterns Histogram) metodu kullanılmıştır. LBPH metodu ile önce maskesiz yüz tanıma yapılmıştır ve maskeli yüzler içinde kullanılabilir olduğu yapılan testler sonucu görülmüştür. Çalışmada, maskesiz yüz tanıma, maskeli yüz tanıma, yüz işaretlerini kullanabilme, yapay maskeli yüz oluşturma ve maskeli yüz tespit algoritmaları oluşturulmuştur. Tüm bu parça algoritmalar üzerinde yüzün hangi açıya kadar tespit edilip tanındığı incelenmiştir. Ortam parlaklığından etkilenip etkilenmediği, yüz tespit parametrelerinin tanımadaki etkisi incelenmiştir. Bu incelemelerden sonra veri kümesinin değiştirilmesine karar verilmiştir. Yüzü daha geniş açılardan tespit edebilen Mediapipe veri kümesi ile ilerlenmiştir. Bu noktada eğitim veri kümesinin sayısı ve çeşitliliğin yüz tespitine ve tanımaya etkisi gibi çeşitli analizler yapılmıştır. Çalışmanın sonunda oluşturulan bu alt algoritmalar birleştirilmiştir. Yapılan analizler dikkate alınarak mevcut veri kümesine göre en optimal yüz tanıma sistemi oluşturulmuştur. Sistem canlı bir görüntüyü kameradan alır ve maskeli olup olmadığına karar verir; maskeli ise tüm yüzün üstten 0,4'lük kısmını alır ve daha önce oluşturulmuş LBP histogramları ile karşılaştırır. Bulunan yüzün ismi ya da numarası ekrana yansıtılır. Maskesiz ise tüm adımlar yarım yüz yerine tam yüz ile yapılır. Bu şekilde çalışan birleştirilmiş son sistemin maskesiz yüzleri tanıma başarıları %99,33 ve maskeli yüzleri tanıma başarıları %95,78'dir.

Bilim Kodu : 90521

Anahtar Kelimeler : Görüntü işleme, yüz tespiti, yapay maske, yüz tanıma, maskeli yüz tanıma, OPENCV, LBPH (local binary patterns histogram)

Sayfa Adedi : 67

Danışman : Prof. Dr. Hasan Şakir BİLGE

DEVELOPMENT OF DETECTION AND RECOGNITION SYSTEM ALGORITHMS WITH IMAGE PROCESSING AND COMPUTERIZED VISION TECHNOLOGIES

(M. Sc. Thesis)

Sezayi YEŞİLBAĞ

GAZİ UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

June 2022

ABSTRACT

Face recognition, and especially masked face recognition, is frequently used today to prevent security vulnerability. It is an extremely useful system for personnel tracking or detecting criminals in schools, hospitals, shopping malls, military areas or any controlled areas. With the development of image processing and computer vision technologies, various methods have been derived to make object and face recognition. In this study, the LBPH (Local Binary Patterns Histogram) method, which gives more successful results and which is faster because it works based on gray images, was used. Unmasked face recognition was first performed with the LBPH method and it was seen as a result of the tests that it can be used in masked faces. In the study, unmasked face recognition, masked face recognition, using face signs, artificial masked face creation and masked face detection algorithms were created. On all these algorithms, the angle at which the face is detected and recognized has been examined. Whether it is affected by the ambient brightness or not and the effect of face detection parameters on recognition has been examined. After these examinations, it was decided to change the dataset. Progressed with the Mediapipe dataset, which can detect the face from wider angles. At this point, various analyzes such as the effect of the number and diversity of the training dataset on face detection and recognition were made. At the end of the study, these sub-algorithms were combined. Considering the analyzes made, the most optimal face recognition system was created according to the existing dataset. The system takes a live image from the camera and decides whether it is masked or not, if it is masked, it takes 40% of the entire face from the top and compares it with the previously created LBP histograms. The name or number of the found face is reflected on the screen. Without a mask, all steps are done with a full face instead of a half face. The combined final system working in this way has a 99.33% success in recognizing unmasked faces and a 95.78% success in recognizing masked faces.

Science Code : 90521

Key Words : Image processing, face detection, artificial mask, face recognition, masked face recognition, OPENCV, local binary patterns histogram

Page Number : 67

Supervisor : Prof. Dr. Hasan Şakir BİLGE

TEŞEKKÜR

Bu tez çalışmasının ortaya çıkmasında yardımcı olan, desteğini esirgemeyen, motivasyonumu sürekli artıran, tezim ile ilgili teknik konularda ve prosedürlerde yardımcı olan ve doğru yönlendirmelerde bulunan değerli danışman hocam Prof. Dr. Hasan Şakir BİLGE' ye en içten şekilde teşekkür ederim. Üç dört yıldır aynı iş yerinde birlikte çalıştığım hatta ortak dersler dahi aldığım tüm yüksek lisans eğitim hayatım boyunca yardımlaştığımız birbirimizi motive ettiğimiz arkadaşlarım Selçuk ÇOBAN, Hasan TEMİR, Muhammed Yasin ALTINTAŞ, Ulaş TURNA ve Hasan ÖZKARA arkadaşlarıma ayrı ayrı teşekkür ediyorum. Eğitime ve kişisel gelişimimize önem veren, bizlere katkı sağlayan, yüksek lisans çalışmamda her türlü imkânı sağlayan, izin veren, teşvik eden ve çalışmaktan her zaman mutluluk duyduğum ASELSAN' a çok teşekkür ederim. Bu çalışmayı ortaya çıkarmamda bana her zaman destek olan ve her zaman yanımda bulunan çok değerli aileme de teşekkürlerimi sunarım.

İÇİNDEKİLER

	Sayfa
ÖZET	iv
ABSTRACT.....	v
TEŞEKKÜR.....	vi
İÇİNDEKİLER	vii
ÇİZELGELERİN LİSTESİ.....	ix
ŞEKİLLERİN LİSTESİ.....	x
RESİMLERİN LİSTESİ	xi
SİMGELER VE KISALTMALAR.....	xii
1. GİRİŞ.....	1
2. GÖRÜNTÜ İŞLEME VE YÜZ TANIMA.....	7
2.1. Algoritmanın Belirlenmesi.....	9
2.2. LBPH (Local Binary Patterns Histogram) Algoritması	10
2.2.1. LBPH algoritmasının çalışması	10
2.2.2. LBPH uygulama alanları.....	14
2.3. Haar Cascades	15
2.3.1 Haar basamaklı sınıflandırıcı yapımı	15
2.4. Renkli Görüntünün Gri Görüntüye Dönüştürülmesi.....	19
2.4.1. Ortalama yöntem	19
2.4.2. Ağırlıklı yöntem	20
2.5. Kullanılan Program ve Kütüphaneler.....	20
3. YAPILAN ÇALIŞMALAR.....	23
3.1. Maskesiz Yüz Tanıma	23
3.1.1. Yüz tespiti.....	25

	Sayfa
3.1.2. Özellik çıkarma	25
3.1.3. Veri kümesi oluşturma	26
3.1.4. Veri kümesi eğitimi	27
3.1.5. Yüz tanıma.....	28
3.2. Yüz İşaretlerinin Çıkarılması	29
3.3. Maskeli-Maskesiz Ayrımının Yapılması	31
3.4. Yapay Maskeli Yüz Oluşturma	33
3.5. Maskeli Yüz Tanıma	34
3.6. Maskeli ve Maskesiz Tanımanın Birleştirilmesi.....	36
4. BULGULAR VE ANALİZLER.....	39
4.1. Yatay Eksende Ön Yüz Açısına Göre Yüz Tespiti Başarısı.....	39
4.2. Eğitim Veri Kümesinin Tanımadaki Etkisi	41
4.2.1. Tanınmayan ya da yanlış tanınan yüzlerin analizi	42
4.3. Yüz Tespit Parametrelerine Bağlı Tespit ve Tanıma Başarısı	44
4.4. Eğitim Veri Kümesinin Maskeli-Maskesiz Ayrımındaki Başarıya Etkisi.....	47
4.5. Mevcut Maskesiz Yüz Tanıma Sisteminin Maskeli Yüze Tepkisi	49
4.6. Yüz Tanıma Başarı Oranları ve Karşılaştırılması	50
5. SONUÇ VE ÖNERİLER	57
KAYNAKLAR	60
ÖZGEÇMİŞ	65

ÇİZELGELERİN LİSTESİ

Çizelge	Sayfa
Çizelge 2.1. Histogram hesaplama adımları	11
Çizelge 4.1. Eğitimde kullanılan maskeli-maskesiz yüz sayısına göre kodun performansı	47
Çizelge 4.2. Maskesiz yüzü doğru bilme olasılığı	48
Çizelge 4.3. Maskeli yüzü doğru bilme olasılığı	49
Çizelge 4.4. Mevcut maskesiz yüz tanıma sisteminin maskeli yüz tanıma başarısı	50
Çizelge 4.5. Maskesiz yüzlerin tanınma başarısı	50
Çizelge 4.6. Maskeli yüzlerin tanınma başarısı	50
Çizelge 4.7. Yöntem 1, yüz tespit hata matrisi (OPENCV ön yüz)	51
Çizelge 4.8. Yöntem 2, yüz tespit hata matrisi (Mediapipe)	52
Çizelge 4.9. Yöntem 2, maskeli yüz tespit hata matrisi (Mediapipe)	53
Çizelge 4.10. Eğitim seti maskeli tam yüz olduğunda tanıma başarısı.....	54
Çizelge 4.11. Tez kapsamında yapılan yüz tanıma sırasında elde edilen doğruluk ve çeşitli sistemlerin maskeli yüzleri tanıyıp tanımadığına dayalı olarak mevcut sistemlerle karşılaştırma tablosu	55

ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 2.1. Local binary patterns histogram hesaplama gösterimi	11
Şekil 2.2. Birleştirilmiş histogram gösterimi	13
Şekil 2.3. Haar özelliklerinin türleri	16
Şekil 2.4. İntegral bir görüntünün nasıl çalıştığını gösteren çizim	17
Şekil 2.5. Boosting algoritmasının temsili	18
Şekil 2.6. Kademeli sınıflandırıcıların akış şeması.....	18
Şekil 3.1. Maskesiz tanıma akış şeması.....	24
Şekil 3.2. Veri kümesi oluşturma ve eğitim süreci	27
Şekil 3.3. Yüz tanıma.....	29
Şekil 3.4. Mediapipe yüz noktalarının yerleri.....	30
Şekil 3.5. Maskeli yüz tespit akış şeması.....	32
Şekil 3.6. Maskeli yüz tanıma adımları	35
Şekil 3.7. Maskeli/maskesiz yüzlerin tanınması için oluşturulan algoritmanın akış şeması	37
Şekil 4.1. Farklı açılardaki yüzün tespitindeki başarı oranı	40
Şekil 4.2. Algoritmanın eğitim veri kümesine göre performansı ve tanıma başarısı.....	41
Şekil 4.3. Yüz tespiti fonksiyonu ve parametreleri.....	44
Şekil 4.4. Ölçekleme parametresine bağlı tanıma başarısı ve test süresi.....	45
Şekil 4.5. minNeighbors parametresine bağlı tanıma oranı ve test süresi	46

RESİMLERİN LİSTESİ

Resim	Sayfa
Resim 1.1. Farklı maske türleri kullanmış insan yüzleri	2
Resim 2.1. (a) İlgili görüntü, (b): Gri görüntü, (c): Renkli görüntü	7
Resim 2.2. Farklı parlaklıktaki görüntülerin LBP görüntüleri	12
Resim 2.3. Haar sınıflandırıcı eğitiminin genel bir temsili	15
Resim 3.1. Yüz işaretlerinin insan yüzündeki gösterimi	31
Resim 3.2. Maskeli pozitif görüntülerin bazıları	32
Resim 3.3. Maskesiz negatif görüntülerin bazıları	33
Resim 3.4. Yapay maske eklenmiş insan yüzleri (a) Cerrahi maske (b) N97 maske.....	33
Resim 3.5. Maskeli yüze ait tanımda kullanılacak yüz parçası	35
Resim 3.6. Oluşturulan son algoritmaya ait maskeli yüze ait tanıma çıktısı	38
Resim 3.7. Oluşturulan son algoritmaya ait maskesiz yüze ait tanıma çıktısı	38
Resim 4.1. Tek kişiye ait farklı açılardaki yüz görüntüleri	39
Resim 4.2. Tanınmayan yüzler için tespit edilen test görüntüleri 12 adet	42
Resim 4.3. Tanınmayan yüzler için tespit edilen test görüntüleri 9 adet	43

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Simgeler	Açıklamalar
sn	Saniye
%	Yüzde
Kısaltmalar	Açıklamalar
FN	False Negative
FP	False Positive
ID	Kimlik
LBPH	Yerel ikili modeller histogramı (Local Binary Patterns Histogram)
MYT	Maskeli Yüz Tanıma
OPENCV	Açık Kaynaklı bilgisayarla görme (Open Source Computer Vision)
PCA	Temel bileşenler Analizi (Principal Components Analysis)
PYTHON	Programlama dili
RGB	Kırmızı yeşil mavi (RedGreenBlue)
TN	True Negative
TP	True Positive

1. GİRİŞ

Yüz tespiti ve tanıma, bilgisayarlı görü teknolojilerinin uygulandığı ve sürekli iyileşmenin olduğu günlük yaşantımızda dahi sıkça karşılaşmaya başladığımız bir sistemdir. Yüz tespiti sistemi görüntüdeki insan yüzlerinin varlığını belirten bir sistem iken yüz tanıma sistemi aynı zamanda o yüzün kime ait olduğunu belirten bir sistemdir. Yüz tanıma günlük hayatımızda çoğu yerde karşımıza çıkmaktadır önemi göz ardı edilemez. Günümüzde çoğu sektör güvenliklerini sağlamak ve kendilerine yardımcı olması için yüz tanımayı kullanması neredeyse zorunluluk haline gelmiştir. Emniyet ve askeri alanlarındaki suçu işleyen kişinin tespiti, terör suçluların tespiti, pasaport doğrulamaları, telefonlarda güvenli giriş yöntemi, bankalarda yine güvenli doğrulama yöntemi, okullarda yoklamanın otomatik yapılması ve bunlar gibi birçok kontrollü kontrolsüz ortamlarda yüz tanıma sistemleri yaygın bir şekilde kullanılmaktadır. Bu sistemlerde insan yüzünün biyometrik özelliklerinden yararlanır. Biyometrik özellik kullanılarak parmak izi avuç içi tanıma göz tanıma gibi farklı tanıma yöntemleri de mevcuttur ancak en popüler olanı yüz tanımadır.

Biyometrik özellikler uzun yıllardır tanıma sistemlerinde kullanılmaktadır. Özellikle kontrollü ortamlarda ve uygulamalarda avuç içi okuma, göz tanıma, imza analizi, konuşma tanıma, yüz tanıma gibi sistemler hayatımıza girmiş durumdadır. Görüntü işlemede tanıma sistemlerini etkileyen çok fazla sayıda etmen vardır ve dolayısı ile her sistemin başarısı her ortamda aynı olmayabilir. Işıklandırma, kameraya olan uzaklık, yüzün yatay ve dikey derecesi, makine öğrenmesi yapılan eğitim veri kümesindeki görüntü ile o anlık görüntü arasındaki saç, sakal, jest, mimiklerde bulunan farklılıklar gibi birçok dikkat edilmesi gereken ve göz ardı edilmemesi gereken problemler bulunmaktadır. Bundan dolayıdır ki sadece yüz tanıma sistemi kullanılması ziyade yüksek başarıya sahip yüz tanıma sisteminin kullanılması daha doğru bir yaklaşımdır.

Gerçek zamanlı, canlı, akan bir görüntü ve videoda tanıma sistemlerindeki başarı genellikle daha düşüktür. Çünkü bu durumda tanımda kullanacağımız görüntülerin kalitesinin düşük olması, net olmaması, bulanık olması, yarım olması, uzak olması vb. bir sürü etmen tanımayı olumsuz etkilemektedir. Bu yüzden yüz tanımda görüntü ve yüz üzerinde kırpma, yeniden boyutlandırma, keskinleştirme, gürültü giderme, normalleştirme, iyileştirme gibi bazı ön işleme teknikleri kullanılmaktadır [1].

2019 yılında dünyada baş gösteren COVID-19 virüsünün yayılmasını azaltmak için kullanılan maskeler, yüz tanıma sistemleri ile uğraşan araştırmacıları ve bilim insanlarını normal yüz tanıma algoritmalarını tekrardan gözden geçirmeye itti. Bu şekilde son yıllarda maskeli yüz tanıma daha önemli ve popüler bir konu olmaya başladı. Normal yüz tanıma sistemlerinin çoğunda tanınanın güvenli olabilmesi için maskenin çıkarılması gerekmektedir. Bir insanın maskeli bir şekilde bile tanınması maskeyi çıkarınca oluşacak temas riskini azaltacaktır. Maskeli bir insan yüzünde maskenin altında kalan ağız, burun, çene gibi tanımda önemli bir yer tutan yüz özellikleri tanımda kullanılamayacağı için bu anlamda başarının düşmesi kaçınılmaz. Ancak bu eksiklikten minimum seviyede etkilenecek sistemler hiç şüphe yok ki tanımda kullanılabilir.

Tıbbi maskelerin yanı sıra esas olarak bazı teröristler ve suçlular kılık değiştirmek için yüzlerini maskeyle kapatırlar. Bunun yanında güneş gözlüğü, şapka, vb. de bir çeşit maske görevi görür. Farklı çeşitlerde yüz maskeleri, insan yüzünün tanınmasındaki temel özellikleri etkilemektedir ve azalmaktadır. Maskeli yüzdeki daha az sayıda yüz özelliği, diğer normal yüz tanıma tekniklerine göre zorluklara neden olur [2]. Sonuç olarak, tanıma doğruluk oranı azalmaktadır. Bu sebeple maskeli yüzler, yüz tanımda aşılması gereken bir konu olmaya başlamıştır. Resim 1.1 farklı türde maskelenmiş insan yüzlerinden örneklerdir.



Resim 1.1. Farklı maske türleri kullanmış insan yüzleri [3]

Yüz tanıma sistemlerine genel hatları ile bakacak olursak hepsinin ortak yaptığı bazı aşamalar vardır. Bunlardan ilki görüntüde bir insan yüzünün olup olmadığına karar vermektir. Eğer insana ait bir yüz bulundu ise eğitim veri kümesindeki yüz görüntüleri ile karşılaştırır ve ilgili yüzü tespit etmeye çalışır. Bu karşılaştırma yüze öznitelikler üzerinden gerçekleşir. Karşılaştırma çeşitli yöntem ve algoritmalarla yapılabilir. Bu çalışmada, yüz tanıma için LBPH (Local Binary Patterns Histogram) metodu kullanılmıştır. LBPH daha sonra başlık şeklinde anlatılacaktır. Ayrıca daha sonra sıkça söz edilecek olan bazı kelimelerin tanımları aşağıdaki gibidir:

Eğitim veri kümesi: Yüz tanıma sistemlerin tanınmanın gerçekleşebilmesi için girdi olarak aldığı yüz görüntüsünü karşılaştıracığı ve önceden sistem tarafından bilinen görüntüleri içermektedir.

Test veri kümesi: Yüz tanıma sistemi devreye alınmadan sistemin performansını test edebilmek için eğitim veri kümesinde mutlaka bulunmaması gereken görüntüleri içermektedir.

OPENCV (Açık Kaynaklı Bilgisayarla Görme): Intel tarafından 1999'da başlatılan popüler bir bilgisayarla görme kütüphanesidir. Platformlar arası kütüphane, odağını gerçek zamanlı görüntü işlemeye ayarlar ve en son bilgisayarla görme algoritmalarının patentsiz uygulamalarını içerir [4].

PYTHON: Yorumlanmış, etkileşimli, nesne yönelimli bir programlama dilidir. Modüller, istisnalar, dinamik yazım, çok yüksek seviyeli dinamik veri türleri ve sınıfları içerir. Fonksiyonel programlama gibi nesne yönelimli programlamanın ötesinde çoklu programlama paradigmalarını destekler [5].

Çalışmada kullanılan tüm yöntem bulgu ve sonuçlar daha sonra ayrıntılı bir şekilde anlatılacağından özet şeklinde bakılacak olursa çalışmanın amacı kontrollü ortamlarda eşkâl tespit ve tanınmanın gerçekleştirilerek güvenlik açığının önüne geçmektir. Ayrıca bir diğer amacı ise COVID-19 ve benzeri maske takmamızı gerektirecek durumda tanınmanın maskeli bir şekilde gerçekleşmesi ile birlikte maskenin yüzden çıkarılarak oluşabilecek virüs temas ihtimalini ortadan kaldırmaktır.

Yüz tanıma kapsamında bu çalışmada görüntü işleme ve bilgisayarlı görü teknolojisi çerçevesinde yüz tanıma sistemlerinde kullanılan OPENCV kütüphane ve fonksiyonların tanımayla nasıl etki ettiğinin analizi edilerek en başarılı bir şekilde insan yüzünün tanınması için testler yapılmıştır. Çalışma her ne kadar maskeli ya da maskesiz olmaksızın başarılı bir şekilde tanıma yapıp son aşamaya gelse de bu aşamaya gelene kadar yapılan tüm analiz ve bulgulara da yer verilmiştir.

Yüz tespitinin gerçekleşmesi için yapılması gereken adımlar, maskesiz normal yüz tespiti aşamaları eğitim veri kümesinin sayısı ve çeşitliliğinin tanımadaki etkisi, yüz işaretlerinin tanımda kullanılabilirliği, maskeli yüz tespitin ve tanınmasının aşamaları gibi konular üzerinde çalışılmış olup bunları yapabilen kodlar PYTHON programlama dili kullanılarak oluşturulmuştur. Gerekli testler ve analizler yapılmıştır. Araştırma kullanılan ilk yüz tespit yönteminde yüzün hangi dereceye kadar tespit edilebildiğini ve bu yüzden sadece ön yüzlerin tanınmasını kapsamaktadır. Ancak daha sonra bu açının artırılması için eğitim veri kümesinde kullanılan yüzlerin açısının da artırılması gerektiğini ve kullanılan yöntemin ve fonksiyonların bunu desteklemesi gerektiği sonucuna varılmıştır.

Yüz tanımda tanıma için kullanılan yöntem ve yaklaşımlar çok fazladır. Bu bir anlamda birbirine alternatif olarak kullanılabilecek sistemler oluşturmuştur. Bu yöntem ve yaklaşımlara daha önceki çalışmalar ve literatür üzerinden bakarsak farklılıkları görebiliriz. Bu anlamda literatüre bakacak olursak, dünyadaki teknolojik liderlerin son gelişmeleri yüz tanımayla olan ilgi ve ilerlemeyi göstermektedir. Apple, Özel Etkinliğinde “Face ID” teknolojisini sundu. Facebook ve fotoğraflarda insanları etiketlemek için sosyal ağlar tarafından tanıma teknolojisinin aktif olarak tanıtılması bu yönde liderdir. Facebook tarafından geliştirilen bu algoritma %93 tanıma başarısı göstermektedir [6].

İnsan yüzüne ait geometrik yapısı kullanılarak yapılan yüz tanıma, bu alanda en sezgisel yöntemlerden biridir. İlk otomatik yüz tanıma sistemlerinden biri [7]'te tarif edilmiştir: bir özellik vektörü (noktalar arasındaki mesafe, aralarındaki açı, ...) oluşturmak için işaret noktaları (gözlerin, kulakların, burnun konumu, ...) kullanıldı. Tanıma, bir sondanın öznitelik vektörleri ile referans görüntü arasındaki Öklid uzaklığı hesaplanarak gerçekleştirilmiştir. Böyle bir yöntem, doğası gereği aydınlatmadaki değişikliklere karşı dayanıklıdır, ancak büyük bir dezavantajı vardır: İşaret noktalarının doğru şekilde kaydedilmesi, en gelişmiş algoritmalarla bile karmaşıktır. Geometrik yüz tanıma

konusundaki en son çalışmalardan bazıları [8]'de yapılmıştır. [8]'de 22 boyutlu bir özellik vektörü kullanıldı ve büyük veri kümeleri üzerinde yapılan deneyler, geometrik özelliklerin tek başına yüz tanıma için yeterli bilgi taşımayabileceğini gösterdi.

[9]'te yazarlar çalışmalarında K-Nearest Neighbor (KNN) algoritmasını kullanmışlardır. Burada yüz bölgesini tespit etmek için Haar Cascade sınıflandırıcısı ve kişiyi bilinen veya bilinmeyen olarak sınıflandırmak için KNN algoritması kullanılmaktadır. Bu çalışmanın ana dezavantajı, veri boyutundaki artışla sistemin daha yavaş çalışmasıdır. [10]'te yazarlar, LBPH tanıyıcı kullanan panjurlar için yüz ve el hareketi tanıma sistemi geliştirdiler. Burada, yüz algılama için Haar Cascade sınıflandırıcı ve yüz tanıma için LBPH tanıyıcı uygulanmaktadır. Elde edilen doğruluk, el hareketi tanıma için %95,2 ve yüz tanıma için %92'dir. [11]'da görme engelliler için sesli yardımcı yüz tanıma sistemi geliştirilmiştir. Yazarlar bu çalışmada kişinin tespiti ve tanınması için OpenCV-Python kütüphanesini kullanmışlardır. [12] çalışmasında ise maskeli yüz tespiti ve tanınması için Maskeli Yüz Algılama Veri Kümesi, Gerçek Dünya Maskeli Yüz Tanıma Veri Kümesi ve Simüle Edilmiş Maskeli Yüz Tanıma Veri Kümesi dahil olmak üzere üç tür maskeli yüz veri kümesi önermektedir. [3], maskeli yüz tanımda PCA algoritmasını başarılı bir istatistiksel teknik olduğu için ve yaygın olarak kullanıldığı için seçmişlerdir. Maskeli yüz görüntü tanıma doğruluğu ortalama %72, maskelenmemiş yüz ise ortalama %95'tir. Bu nedenle PCA, maskelenmemiş yüzler yerine maskeli yüz görüntüleri için düşük tanıma oranı verdiği anlaşılmıştır.

Maskeli yüz tanıma yönelik yapılan çalışmalara ve yöntemlere biraz daha detaylı bakılacak olursa çok farklı metotlar ve başarı oranları görülmektedir. [13]'deki çalışmada "Bag of features" paradigması CNN algoritmasına uygulanmış ve TL, Covariance, CNN+SVM tekniklerine göre daha yüksek doğruluğa sahip olduğu belirtilmiştir. Aynı karşılaştırmayı RMFRD (%91,3) ve SMFRD (%88,9) veri kümesi için denenmiş ve önermiş oldukları CNN+BoF tekniğinin daha başarılı olduğunu belirtmiştir. Çalışmada ayrıca bu tezde de olduğu gibi maskeli yüz tanıma, maskesiz kısımların karşılaştırılması ile yapılmıştır.

Bir kısım çalışmada tıpkı bu tez gibi yüz özellikleri olarak tanımda yüzün maskesiz kısımlarını alan çalışmalar olduğu gibi bu durumda sadece göz tanıma yapan maskeli kısma sanal yüz oturtmak ve çeşitli şekillerde kullanmak gibi değişik yöntemler kullanarak tanıma

sistemi yapılan çalışmalar da mevcuttur. [14]'deki çalışmada Haar Cascades, LBPH ve OPENCV kullanılmış ve maskeli yüz tanıma üzerinde çalışılmıştır. Maskeli yüzleri -15 ile +15 dereceye kadar tanımıştır. Bu çalışmada da bu tezin ilk kısmında maskesiz yüzler için kullanılan veri kümesi kullanılmıştır. Bu da gösteriyor ki daha geniş açılarda veri kümesi çok önem kazanmaktadır. Bu yüzden [14]'de veri kümesine sadece göz ve çeşitli açılardan insan yüzü konularak tanıma gerçekleştirilmiştir. Ayrıca görüntü içinde birden fazla kişi olunca tanıma oranına bakılmış ve maskesiz yüzlerde %90 maskeli yüzlerde ise %80 oranında başarı sağlanmıştır.

[15]'deki çalışmada ise yine maskeli yüz tanıma farklı veri kümelerinin farklı metotlar ile birleştirilmesi ile başarı oranı paylaşılmıştır. Bu bilgilere göre sentezlenmiş maskelenmiş CASIA NIR-VIS 2.0 veri kümesinde, sentezlenmiş maskelenmiş Oulu-CASIA NIR-VIS veri kümesi ve yine sentezlenmiş maskelenmiş BUAA-VisNir veri kümelerinden genel olarak daha başarılı olunmuştur. Sırası ile maksimum başarılar: %98,60, %91,3 ve %98,60'dır. Yine bu çalışmada maskeli yüz tanıma farklı metotlar ile de denenmiş ve genel olarak en yüksek başarı HSST (Triplet) metodunda çıktığı belirtilmiştir. Görüldüğü üzere başarıyı etkileyen en önemli iki parametre veri kümesi ve kullanılan yöntemdir. Bu çalışmanın benzer çeşitli bilimsel çalışma ile karşılaştırılmış hali Çizelge 4.11'de verilmiştir.

Kalan bölümlerde önce çalışma içerisinde kullanılan metot ve tanımlamalar detaylandırılacak sonrasında çalışma kapsamında yapılan analiz karşılaştırma ve iyileştirilmeler aktarılacak, tartışma bölümüne yer verilecek ve son olarak sonuç ve öneriler kısmında tüm edinilen deneyimler özetlenerek aktarılacaktır.

2. GÖRÜNTÜ İŞLEME VE YÜZ TANIMA

Görüntü, bir özneye (genellikle fiziksel bir nesne) benzeyen bir fotoğraf veya başka iki boyutlu resim gibi görsel algıyı betimleyen ve böylece onun bir tasvirini sağlayan bir çıktıdır.

Görüntü Türleri: Görüntü türlerini 3 başlıkta ele alabiliriz.

İkili görüntü: Yalnızca 0'lar ve 1'ler içeren, sırasıyla siyah ve beyaz olarak yorumlanan mantıksal dizi [16].

Gri görüntü: Piksel değerleri yoğunluk değerlerini belirten uint8, uint16, int16, single (tek) veya double (ikili) sınıfı dizisi. Tek veya çift diziler için değerler [0, 1] aralığındadır. uint8 için değerler [0,255] aralığındadır. uint16 için değerler [0, 65535] aralığındadır. int16 için değerler [-32768, 32767] aralığındadır.

Renkli görüntü: Literatürde RGB olarak da bahsedilebilir. Gerçek bir renkli görüntü, her pikselin, piksel skalerinin kırmızı, mavi ve yeşil bileşenleri için birer tane olmak üzere üç değerle belirtildiği bir görüntüdür. Piksel değerleri yoğunluk değerlerini belirten uint8, uint16, single veya double sınıfının M by-n-by-3 dizisi. Tek veya çift diziler için değerler [0, 1] aralığındadır. uint8 için değerler [0, 255] aralığındadır. uint16 için değerler [0, 65535] aralığındadır [17-18].



Resim 2.1. (a): İkili görüntü

(b): Gri görüntü

(c): Renkli görüntü [18]

Görüntü işleme, daha gelişmiş bir görüntü elde etmek veya bir görüntüden bazı yararlı bilgiler çıkarmak için görüntü üzerinde bazı işlemleri gerçekleştirme yöntemidir. Girişin bir görüntü veya görüntü bilgisi olduğu ve çıkışın görüntü veya o görüntüyle ilişkili özellikler/özellikler olabileceği bir çeşit sinyal işleme türüdür. Günümüzde görüntü işleme sistemleri hızla gelişen teknolojiler arasında yer almaktadır. Mühendislik ve bilgisayar bilimlerinde de temel araştırma alanı olarak kabul gören konulardandır. Görüntü işleme temelde bazı adımlar içerir [19]:

Görüntü elde etme araçları aracılığıyla görüntünün içe aktarılması;

Görüntüyü analiz etme ve oynama yapma;

Görüntü analizine dayalı olarak sonucun değiştirilebileceği görüntü veya rapor çıktısı [19].

Görüntü işlemede analog ve dijital görüntü işleme olmak üzere iki tür yöntem bulunmaktadır. Çıktı ve fotoğraf gibi basılı kopyalar için analog görüntü işleme kullanılabilir. Bu görsel teknikleri kullanırken çeşitli yorumlama temelleri kullanır. Sayısal görüntü işleme teknikleri, sayısal görüntülerin bilgisayar kullanılarak işlenmesine yardımcı olur. Dijital tekniği kullanırken her tür verinin geçmesi gereken üç genel aşama, ön işleme, geliştirme ve görüntüleme, bilgi çıkarmadır [19].

Yüz tanıma sistemi, dijital bir görüntüden veya bir video karesinden bir insan yüzünü bir yüz veri tabanı ile eşleştirebilen, genellikle kimlik doğrulama hizmetleri aracılığıyla kullanıcıların kimliğini doğrulamak için kullanılan, belirli bir görüntüdeki yüz özelliklerini tam olarak belirleyip ölçerek çalışan bir teknolojidir.

1960'larda benzer sistemlerde geliştirme, bir bilgisayar uygulaması biçimi olarak başladı. Yüz tanıma sistemleri, başlangıcından bu yana, akıllı telefonlarda ve robotik gibi diğer teknoloji biçimlerinde son zamanlarda daha geniş kullanımlar gördü. Bilgisayarlı yüz tanıma, bir insanın fizyolojik özelliklerinin ölçülmesini içerdiğinden, yüz tanıma sistemleri biyometrik olarak sınıflandırılır [20].

2.1. Algoritmanın Belirlenmesi

Görüntü işlemede ve ilgileneceğimiz olan yüz tanıma sistemlerinde çeşitli algoritmalar ve yöntemler mevcuttur. Bunları en yaygın ve çok kullanılan metotlarına şu anda mevcut algoritmalar şunlardır:

- Eigenfaces (Özyüzler, EigenFaceRecognizer) ya da Principal Component Analysis (PCA)
- Fisherfaces (FisherFaceRecognizer)
- Local Binary Patterns Histograms (Yerel İkili Modeller Histogramı, LBPHFaceRecognizer) [21]

Eigenfaces ve Fisherfaces, yüz tanıma için biraz bütünsel bir yaklaşım benimsemektedir. Yüz verileri, yüksek boyutlu bir görüntü uzayında bir vektör olarak ifade edilir. Yüksek boyutluluk işlem sırasında genel olarak istenmeyen birşeydir, bu nedenle yüze ait bilgilerin korunduğu daha düşük boyutlu bir alt uzay tanımlanması daha iyi bir yöntemdir. Eigenfaces yaklaşımı toplam saçılımı en üst düzeye çıkarır, bu da varyans harici bir kaynak tarafından üretilirse sorunlara yol açabilir, çünkü tüm sınıflar üzerinde maksimum varyansa sahip bileşenler sınıflandırma için yararlı değildir. Gerçek hayat görüntüleri mükemmel değildir. Resimlerde veya bir kişinin 10 farklı resminde mükemmel ışık ayarlarını garanti edilemeyebilir. İnsana ait yalnızca bir yüz görüntüsü varsa alt uzay için kovaryans tahminleri yüksek derecede yanlış olabilmektedir, tanıma da aynı şekilde. Bu nedenle, iyi tanıma oranları elde etmek için her kişi için en az 8(+1) görüntüye ihtiyaç bulunacak ve Fisherfaces yöntemi burada yardımcı olmamaktadır. Bu nedenle çoğu araştırma, insan yüzüne ait öz nitelikleri çıkarmaya odaklandı. Buradaki yaklaşım, görüntünün tamamına bakmak yerine, ilgili nesnenin yalnızca tanımada kullanılacak özelliklerini tanımlamaktır. Bu şekilde çıkarılan özellikler dolaylı olarak düşük boyutlu olacaktır. Görüntülerde ölçekleme, öteleme veya döndürme gibi etkenlere karşı en azından biraz sağlam olmalıdır. Local Binary Patterns Histograms metodolojisinin kökleri 2 boyutlu analize dayanır. Yerel İkili Modellerin genel özeti, bir görüntüdeki genel yapıyı ve her pikseli komşularıyla karşılaştırarak daha öz bir hale getirmektir. Merkez pikselin değerini komşu piksel ile karşılaştırır ve eğer komşu pikselin değeri merkez pikselden büyükse '1' değilse '0' atanır. Bu şekilde histograma dönüştürülen görüntü bu etmenlerden daha az etkilenir. Bu sebeplerden dolayı çalışmada Local Binary Patterns Histogram metodu kullanılmıştır [21].

2.2. LBPH (Local Binary Patterns Histogram) Algoritması

LBPH ilk olarak 1994'te tanımlanmıştır ve o zamandan beri nesne sınıflandırması için güçlü bir yöntem olarak kullanılmaktadır. LBP'yi histogramlarla birleştirerek yüz görüntülerini basit bir veri vektörü ile temsil edilebilir. LBP görsel bir tanımlayıcı olduğundan, aşağıdaki adım adım açıklamada görülebileceği gibi yüz tanıma görevleri için de kullanılabilir [22].

2.2.1. LBPH algoritmasının çalışması

LBPH algoritması aşağıdaki dört parametre ile işlemlerini yapar:

- Yarıçap: Dairesel yerel ikili desenin merkez pikselden çevresine olan mesafesine eşittir ve genellikle 1 değerini alır [23].
- Komşular: Dairesel bir yerel ikili model içindeki veri noktalarının sayısı. Genellikle değeri 8'dir [23].
- Izgara X: Yatay düzlemdeki hücre sayısı verir ve genellikle 8 değerindedir [23].
- Izgara Y: Dikey düzlemdeki hücre sayısını verir ve genellikle 8 değerindedir [23].

Yukarıda belirtilen parametreler göz önüne alındığında, LBPH aşağıdaki gibi çalışır; Bir kamera ile görüntüler veya kaydedilen görüntüler girdi olarak alınır. Ardından görüntüdeki kişiye eşsiz bir isim ya da numara verilir. Ardından yüz görüntülerin veri tabanına eklenmesiyle veri kümesi oluşturulur. Eğitim veri kümesi için her kişiden çok sayıda ve farklı görüntü alınması tavsiye edilir. Veri kümesinin bir kısmı algoritmanın eğitimi için kullanılırken kalanı ise algoritmayı test için kullanılır [23].

Dairesel bir komşuluk konsepti (seçilen bir alan etrafında tamsayı olmayan piksel noktaları alan) kullanılır. Yüz görüntüsündeki LBP kodlarının görünümünün sayısı bir histogram oluşturmak üzere bir araya getirilir. Sınıflandırma daha sonra karşılaştırılan histogramların temel benzerliklerinin hesaplanması yoluyla gerçekleştirilir [23].

Bu histogram, bir bireyin tanımını üç farklı düzeyde içerir: piksel düzeyinde, etiketler bölgesel bir yapı oluşturmak için birleştirilir daha sonra bölgesel histogramlar kombinasyon halinde kişinin genel bir tanımını histograma dökmüş olur. Yüz tanıma için LBPH algoritması bazı adımları takip eder. Bu adımlar aşağıda belirtildiği şekilde iki aşamada gerçekleştirilebilir [23]:

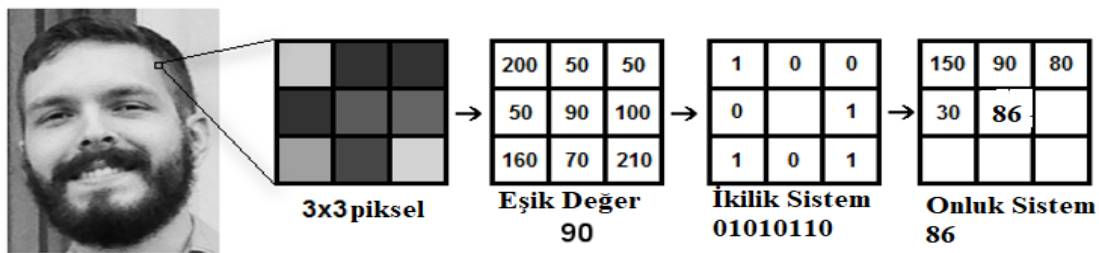
Algoritmanın eğitimi

İlk adım algoritmanın öğrenilmesidir. Tanıma sürecine dahil edilecek kişilerin görüntülerinden oluşan bir eğitim veri kümesinden yararlanır. Algoritmanın bu bilgiyi görüntüyü tanımlamak ve çıktıyı dışa aktarmak için kullanabilmesi için her görüntüye sayı veya bir kişinin adı olarak benzersiz bir numara ya da isim verilir. Aynı kişinin resimleri her zaman aynı kimliğin altına yerleştirilir [23].

LBP işleminin uygulanması:

Hesaplama adımlarının ilk adımıdır. Burada, iki parametre dikkate alınarak kayan pencere konsepti aracılığıyla orijinal görüntüyü daha iyi temsil etmek için komşu ve yarıçap bilgilerini içeren bir ara görüntü oluşturulmuştur. 8 komşu değer ile eşik değeri karşılaştırılarak ikilik formda yeni değerler oluşturulur.

Eşik değerinden büyük her komşu değer için değer 1, eşik değerinden küçük her komşu değer için 0 olarak ayarlanır. Bu, eşiği hariç tutan bir ikili sayı matrisi oluşturur. Matrisin merkezi bir değeri, ikili sayının orijinal görüntünün piksellerine karşılık gelen ondalık bir değere dönüştürülmesiyle oluşturulur. Bu orijinal görüntünün özelliklerinin daha iyi temsili için yapılmaktadır [23,24].



Şekil 2.1. Local binary patterns histogram hesaplama gösterimi [24]

Çizelge 2.1. Histogram hesaplama adımları

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

1	0	0
0	?	1
1	0	1

$$? = 2^1 + 2^2 + 2^4 + 2^6 = 2 + 4 + 16 + 64 = 86$$

LBPH Yönteminin Algoritmik Tanımı

LBP operatörünün daha matematiksel bir açıklaması şu şekilde verilebilir:

$$\text{LBP}(x_c, y_c) = \sum_{p=0}^{p-1} 2^p s(ip - ic) \quad (2.1)$$

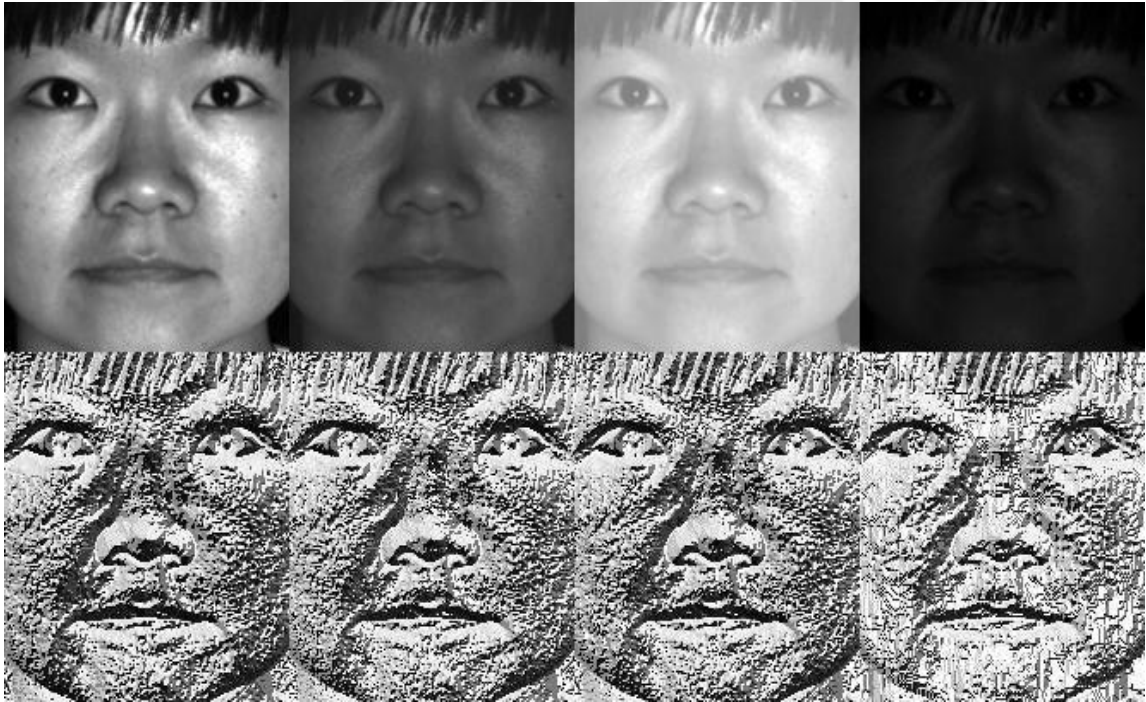
“(xc, yc)” yoğunluğu “ic” olan merkezi piksel olarak;

ip: komşu pikselin yoğunluğu

s, şu şekilde tanımlanan işaret fonksiyonu:

$$s = \begin{cases} 1 & \text{eğer } x \geq 0 \\ 0 & \text{değilse} \end{cases} \quad (2.2)$$

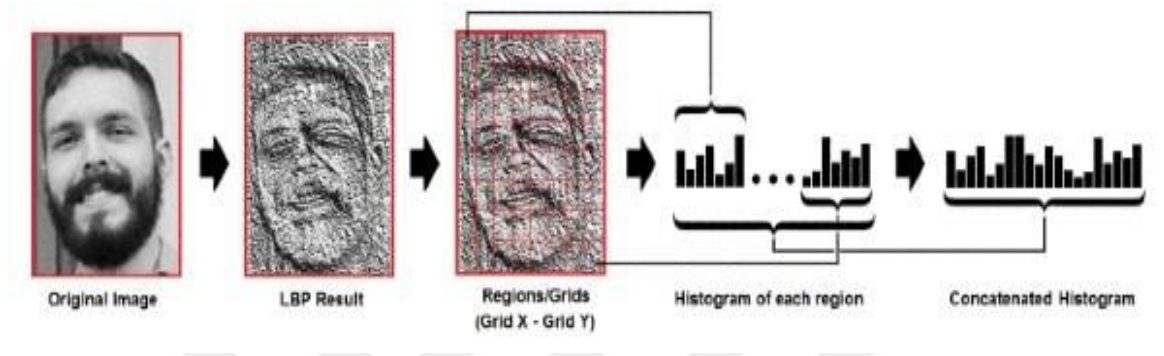
Tanım olarak LBP operatörü, monotonik gri tonlama dönüşümlerine karşı dayanıklıdır ve ortam aydınlığından çok az etkilenir. Bir görüntünün LBP görüntülerine bakarak çok kolay bir şekilde çok benzer olduğu anlaşılır.



Resim 2.2. Farklı parlaklıktaki görüntülerin LBP görüntüleri [9]

Resim 2.2’ye bakıldığında LBP metodunun ortamın ve görüntünün parlaklığı ile büyük oranda etkilenmemektedir. Bu sebepten çalışmada bu metot kullanılmıştır.

Histogramları çıkarmak için elde edilen Resim 2.2'deki görüntü, X ve Y Grid parametrelerinin yardımıyla çoklu ızgaralara bölünür. Bu görüntü gri tonlamalıdır ve ızgaraların her birinin histogramları piksel oluşumların yoğunluğunu temsil etmektedir. Her histogram daha sonra orijinal görüntünün özelliklerini temsil eden yeni ve nihai histogramı oluşturmak için birleştirilir. Şekil 2.2. yukarıdaki kesinliği göstermektedir [23].



Şekil 2.2. Birleştirilmiş histogram gösterimi [23]

Yukarıdaki resme dayanarak, her bölgenin histogramını aşağıdaki gibi çıkarabiliriz:

Görüntü gri tonlamalı olduğundan dolayı, her histogram (her ızgaradan), her piksel yoğunluğunun oluşumlarını temsil edebilmek için yalnızca 256 konum (0~255) içerecektir. Ardından, yeni ve daha büyük bir histogram oluşturmak için her bir histogramın birleştirilmesi gerekmektedir. İlgili ızgara 8x8 olduğunu varsayılırsa, son histogramda $8 \times 8 \times 256 = 16384$ konum bulunacak anlamına geliyor. Son histogram, görüntünün orijinal görüntüsünün özelliklerini temsil etmiş olur [24].

Doğru yüz tanıma:

Eğitim veri kümesindeki her bir yüz görüntüsü için bir histogram oluşturulduktan sonra giriş görüntüsünde eğer yüz var ise o yüze en yakın histogram eşleşmesine sahip görüntünün çıktısını almak için iki histogram karşılaştırılır. Bu çıktı, görüntünün numarası veya adını çıktı olarak verir. Bu algoritma aynı zamanda hesaplanan mesafe olan bir güven ölçümü de verir. Algoritmanın görüntüyü tanımadaki doğruluğu, güven ve eşik değeri ile belirlenir. Doğru bir tanıma, tanımlanan eşikten daha düşük bir güven değeri ile gösterilir. Her yeni test görüntüsü için bu işlem yüze tekrarlanır [23].

Sonraki adımda, algoritma eğitilmiş durumdadır ve oluşturulan her bir histogram, eğitim veri kümesindeki kayıtlı görüntüleri temsil eder. Algoritma boyunca girdi görüntüsünü değiştikçe yeni görüntü için aynı adımlar tekrarlanır [24].

Histogramları karşılaştırmak için (iki histogram arasındaki mesafeyi hesaplamak) çeşitli yaklaşımlar kullanılabilir, örneğin: öklid mesafesi, ki-kare, mutlak değer, vb. Bu çalışmada, Öklid mesafesini temel alarak kullanıldı.

$$D=\sqrt{\sum_{i=1}^n (histogram1i - histogram2i)^2} \quad (2.3)$$

Tanınan görüntü ya da algoritma çıktısı, en yakın histograma sahip görüntünün kimliğidir. Algoritma ayrıca bir "güven" (confidence) ölçümü yapar. Çeşitli sistemlerde kullanılabilecek hesaplanan mesafeyi de çıktı olarak verir. İlgili fonksiyon için düşük güven daha iyidir anlamına gelmektedir ve histogramlar arasındaki mesafenin daha yakın olduğu anlamına gelir [24].

Bu aşamada, algoritmanın görüntüyü doğru bir şekilde tanıyıp tanımadığını tahmin etmek için bir eşik ve "güven" değeri kullanılmalıdır. Güven, tanımlanan eşik değerinden düşükse, algoritmanın başarıyla tanıdığını anlamına gelmektedir. LBPH, en kolay ve başarılı yüz tanıma algoritmalarından biridir. Görüntülerdeki belirleyici özellikleri temsil eder ve iyi sonuçlar elde etmek mümkündür. Monotonik gri tonlama dönüşümlerine karşı dayanıklıdır. Algoritma OpenCV kütüphanesi (Open Source Computer Vision Library) tarafından sağlanır [24].

2.2.2. LBPH uygulama alanları

LBPH algoritması ile yüz tanımda olduğu gibi aşağıdaki alanlarda da kullanılabilir:

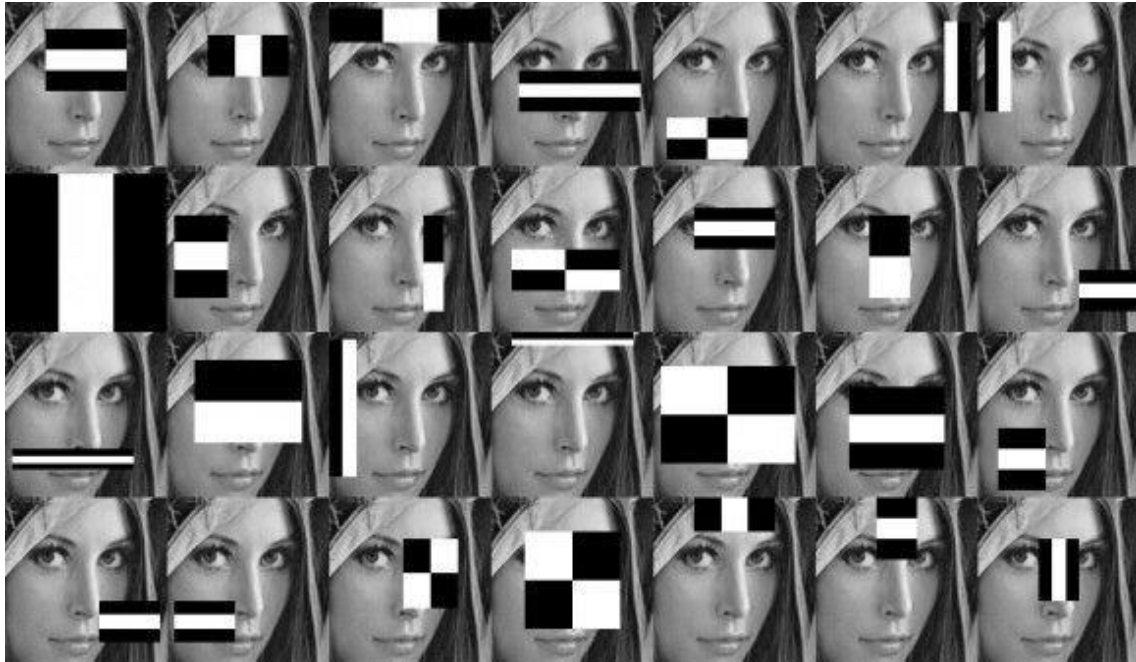
Doku analizi: araştırmalarda ve tıbbi görüntüleme gibi uygulamalarda günümüzde hala uygulanabilmektedir. Bu algoritma, analizde önemli bir ilerlemeye yol açan görüntülerin doku segmentasyonu nedeniyle görüntü işlemeyi daha kolay hale getirmiştir [23].

Biyometri: Avuç içi tanıma, parmak izinden tanıma, gözden tanıma, yürüyüş analizi ile tanıma gibi biyometride ve bir yaş tahmini çalışmalarında kullanılabilir [23].

Bilgisayarla görme: hareket analizi gibi bilgisayarla görmede kullanılır. LPH, bilgisayar sistemlerinin görüntüler üzerindeki bilgileri anlamasını ve bu bilgileri anlamlandırmasını sağlar [23].

2.3. Haar Cascades

Çalışmada ön yüz tanıma kısmında yüz görüntülerini fotoğraftan, videodan ya da kameradan tespit etmek için haar cascades yöntemi ile çıkarılmış yüz özelliklerinden faydalanılmıştır.



Resim 2.3. Haar sınıflandırıcı eğitiminin genel bir temsili [25]

Haar sınıflandırıcılar, ilk gerçek zamanlı yüz tespitte kullanılan sınıflandırıcılar. Haar sınıflandırıcı veya Haar kademeli sınıflandırıcı, bir görüntü ve videodaki nesneleri tanımlayan bir makine öğrenimi ile nesne algılama yöntemidir [25].

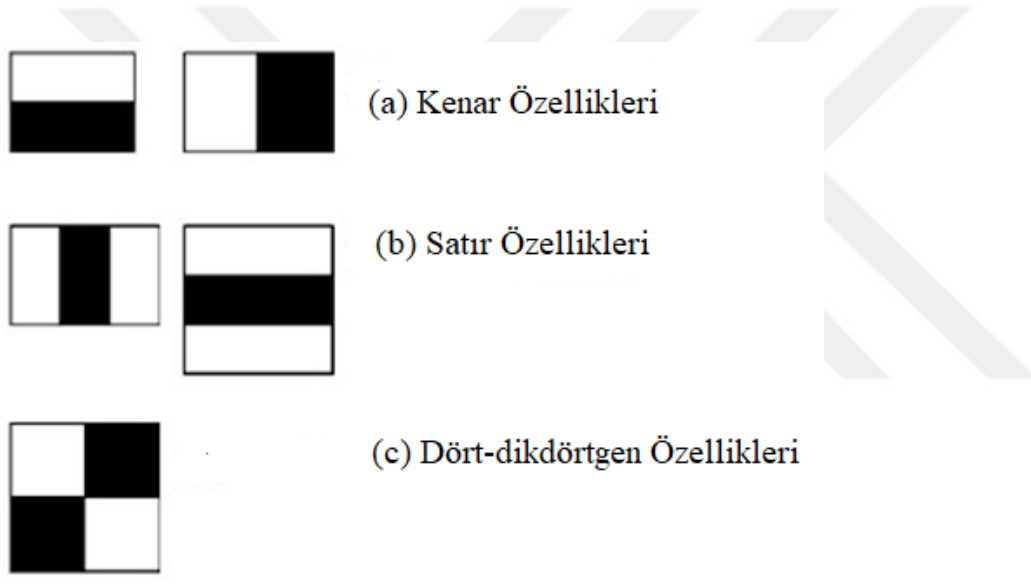
2.3.1. Haar basamaklı sınıflandırıcı yapımı

Bu algoritmanın, diğer makine öğrenimi modellerine benzer şekilde, sınıflandırıcıyı eğitmek için insan yüzlerinin bulunduğu çok sayıda pozitif ve yüz olmayanların negatif görüntülere ihtiyaç vardır. Algoritma dört aşamada açıklanabilir: Haar özelliklerini

hesaplama, integral görüntüler oluşturma, adaboost'u kullanma ve basamaklı sınıflandırıcıları uygulama [25].

Haar Özelliklerini Hesaplama

İlk adım, görüntüdeki Haar özelliklerini toplamaktır. Bir Haar özelliği, ilgili görüntü üzerinde bir algılama penceresinde belirli bir konumda bitişik dikdörtgen bölgeler üzerinde gerçekleştirilen hesaplamaların çıktısıdır. Hesaplama, her bölgedeki piksel yoğunluklarının toplanmasını içerir ve daha sonra toplamlar arasındaki farkların hesaplanmasını içerir. Aşağıda bu Haar özelliklerinin bazı örnekleri verilmiştir [25].

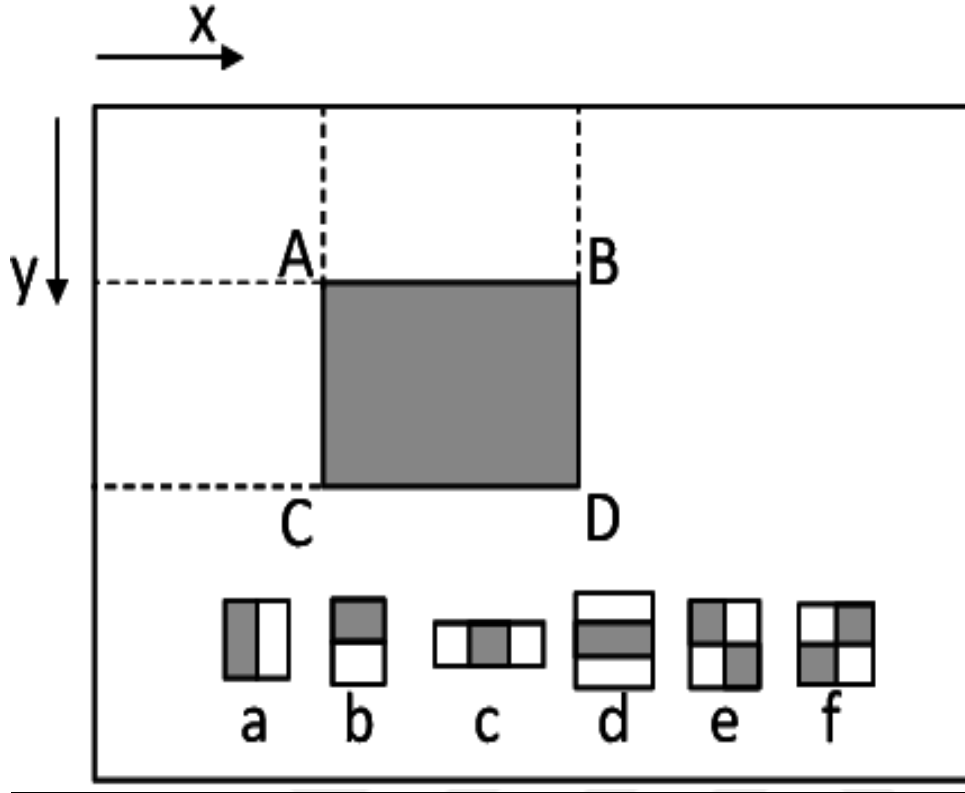


Şekil 2.3. Haar özelliklerinin türleri [25]

Bu özelliklerin büyük bir görüntü ve görüntü kümesi için belirlenmesi zor ve uzun süreli olabilmektedir. İntegral görüntülerin kullanılmasıyla işlem sayısı azaltıldığından, integral görüntülerin devreye girdiği yer tam olarak burasıdır [25].

İntegral Görüntüler Oluşturma

Tümleşik görüntüler, bu Haar özelliklerinin hesaplanmasını esasen hızlandırır. Pratiklik ve performans için her pikselde hesaplama yapmak yerine, alt dikdörtgenler oluşturur ve bu alt dikdörtgenlerin her biri için dizi referansları oluşturulur. Bunlar daha sonra bu Haar özelliklerini belirlemek ya da hesaplamak için kullanılır [25].

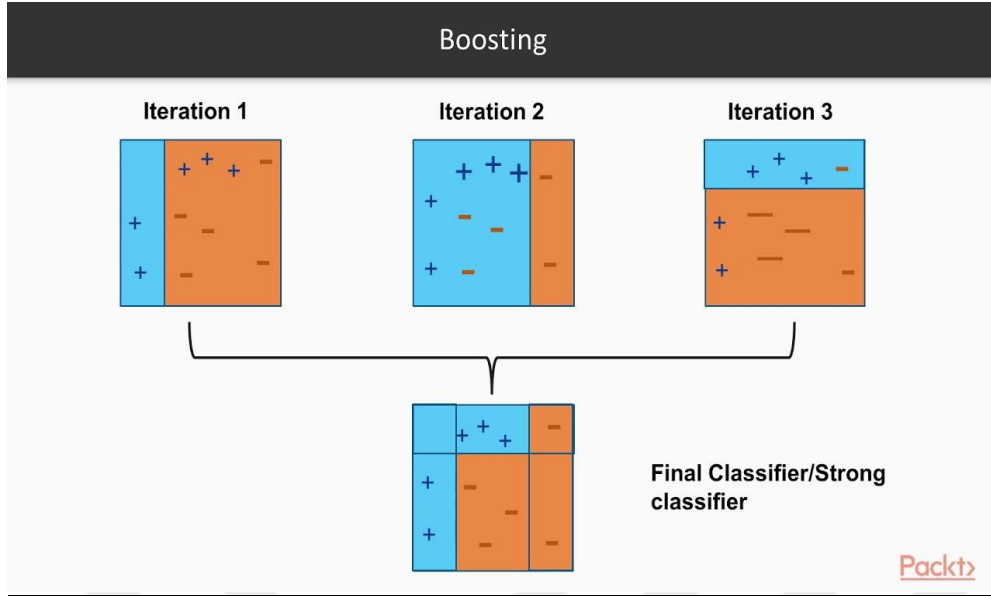


Şekil 2.4. İntegral bir görüntünün nasıl çalıştığını gösteren çizim [25]

Nesne algılama yapılırken neredeyse tüm Haar özelliklerinin alakasız olabileceği gözden kaçırılmamalıdır. Burada önemli olan şey ilgili nesnenin özellikleridir. Ancak, yüz binlerce Haar özelliğinden bir nesneyi temsil eden en iyi özelliklerin belirlendiği yer adaboost eğitimidir [25].

Adaboost eğitimi

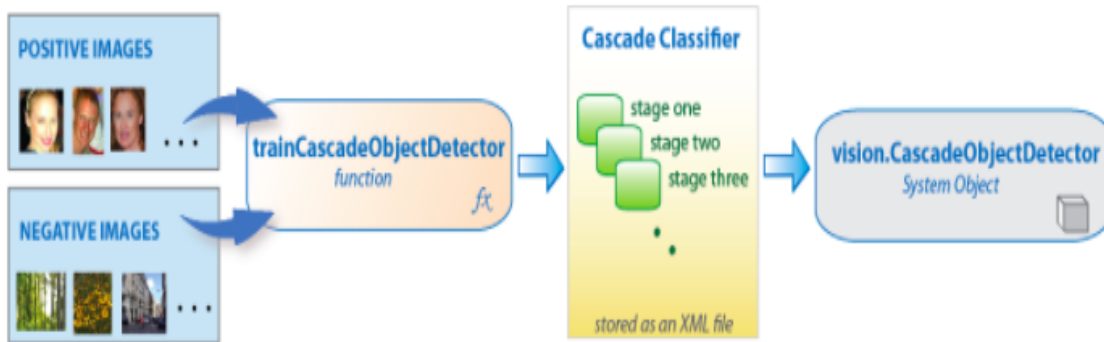
Adaboost, bu aşamada görev olarak en iyi özellikleri seçer ve sınıflandırıcıları bunları kullanmaları için eğitir. Algoritmanın nesneleri algılamak için "zayıf sınıflandırıcılar" kombinasyonunu kullanır ve kullanabileceği bir "güçlü sınıflandırıcı" oluşturur. Zayıf sınıflandırıcılar, bir pencereyi giriş görüntüsü üzerinde hareket ettirerek ve görüntünün her alt bölümü için Haar özelliklerini çıkarır. Bu çıkarılmış fark, nesne olmayanları nesnelerden ayıran öğrenilmiş bir eşik değeri ile karşılaştırılır. Bunlar "zayıf sınıflandırıcılar" olduğundan, güçlü bir sınıflandırıcı oluşturmak için çok sayıda Haar özelliği gereklidir [25].



Şekil 2.5. Boosting algoritmasının temsili [25]

Son adım, bu zayıf öğrenicileri basamaklı sınıflandırıcılar kullanarak güçlü bir öğrenici olarak birleştirir.

Basamaklı Sınıflandırıcıları Uygulama



Şekil 2.6. Kademeli sınıflandırıcıların akış şeması [25]

Kademeli sınıflandırıcı adımı, her bir aşamanın zayıf öğrenenlerin bir koleksiyonu olduğu bir dizi adımlardan oluşur. Zayıf öğrenenler, tüm zayıf öğrenicilerin ortalama tahminini kullanarak son derece doğru bir sınıflandırıcıya izin veren bir yapıda güçlendirme kullanılarak eğitilir [25].

Bu tahmin eşliğinde, sınıflandırıcı ya bir nesnenin bulunduğunu (pozitif) durumu belirtmeye ya da bir sonraki bölgeye (negatif) geçmeye karar verir ve gerçekleştirir. Tüm Aşamalar, pencerelerin çoğu belirleyici aslında hiçbir şey içermediğinden, negatif örnekleri olabildiğince hızlı reddedecek şekilde tasarlanmıştır. Düşük bir yanlış negatif oranını en üst düzeye çıkarmak oldukça önemlidir. Bunun nedeni bir nesneyi nesne olmayan olarak sınıflandırmak nesne algılama algoritmasını olumsuz bir şekilde bozar. Haar basamakları yöntemi, nesne tespiti için kullanılan birçok algortmadan sadece biridir. Haar basamakları hakkında dikkat edilmesi gereken en önemli nokta, yanlış negatif oranı düşürmenin çok önemli olmasıdır [25].

Çalışmada tanınmanın gerçekleşebilmesi için gri görüntülere ihtiyaç bulunmaktadır. Bu yüzden bu algoritmaların çalışabilmesi için görüntülerin mutlaka griye dönüştürülmesi gerekmektedir.

2.4. Renkli Görüntünün Gri Görüntüye Dönüştürülmesi

Oluşturulan algortmada gri görüntüler kullanılmıştır. Bunun için renkli görüntüler griye dönüştürülmüştür.

Gri tonlamanın bazı avantajları bulunmaktadır:

- Daha düşük boyuttadırlar. Örneğin, RGB (RedGreenBlue) görüntülerde üç renk kanalı vardır ve üç boyutludur, gri tonlamalı görüntüler tek boyutludur.
- Model karmaşıklığını azaltır.
- Diğer algoritmaların çalışması için önemlidir. Birçok algoritma yalnızca gri tonlamalı görüntülerde çalışacak şekilde özelleştirilmiştir.

Bir RGB görüntüsünü gri tonlamalı bir görüntüye dönüştürmek için ortalama yöntem ve ağırlıklı yöntem gibi yaygın olarak kullanılan birkaç yöntem vardır.

2.4.1. Ortalama yöntem

Ortalama yöntemi, gri tonlama değeri olarak R, G ve B'nin ortalama değerini alır.

$$\text{Gri tonlamalı} = (R + G + B) / 3 \quad (2.4)$$

Teorik olarak formül %100 doğrudur. Ancak kod yazarken uint8 taşma hatasıyla karşılaşılma ihtimali vardır. R, G ve B toplamı 255'ten büyüktür. İstisnadan kaçınmak için

sırasıyla R, G ve B hesaplanmalıdır. Ortalama yöntem genel olarak basittir ancak beklendiği kadar başarılı değildir. Çünkü, insan gözü RGB'ye farklı tepki gösterir. Gözler yeşil ışığa en duyarlı, kırmızı ışığa daha az duyarlı ve mavi ışığa en az duyarlı şekildedir. Bu sebepten dolayı, bu üç rengin dağılımında farklı ağırlıkları olması gerekmektedir. Bunun için ağırlıklı yöntem kullanılmalıdır [26].

2.4.2. Ağırlıklı yöntem

Parlaklık yöntemi olarak da adlandırılır ve ağırlıklı yöntem, dalga boylarına göre kırmızı, yeşil ve maviyi insan gözü için dengeler. Bu dönüşüm Luminosity yöntem olarak da bilinir. [26]. Geliştirilmiş formül aşağıdaki gibidir:

$$\text{Gri tonlamalı} = 0.299R + 0.587G + 0.114B \quad (2.5)$$

OPENCV renkli görüntüyü griye dönüştürürken ağırlıklı yöntemi kullanmaktadır [27]. Bu çalışmada ağırlıklı yöntem kullanılmıştır.

2.5. Kullanılan Program ve Kütüphaneler

PYTHON

Esas olarak nesne tabanlı programlamayı, belli bir oranda da fonksiyonel programlamayı destekleyen genel amaçlı bir programlama dilidir. 1991 yılında ilk sürümü geliştirilen Python görece kolaylığı, geniş standart kütüphanesi ve dinamik yapısı nedeniyle görüntü işleme dahil çoğu alanda kullanılmaktadır [28]. (PYTHON 3.10.0 sürümü kullanılmıştır.)

OPENCV

OpenCV (Açık Kaynaklı Bilgisayarla Görme), Intel tarafından 1999'da başlatılan çok popüler ve aktif olarak kullanılan bir bilgisayarla görme kütüphanesidir. Platformlar arası bu kütüphane, odağını gerçek zamanlı görüntü işlemeye çevirir. Ayrıca en son bilgisayarla görme algoritmalarının birçok patentsiz uygulamalarını içerir. Günümüzde C, C++, Python ve Android gibi uygulamalarda bir programlama ara yüzü ile birlikte gelmektedir. BSD lisansı altında yayınlandığı için hem akademik projelerde hem de ticari ürünlerde

kullanılmasında bir sakınca yoktur. OpenCV' de FaceRecognizer ile yüz tanımayı nasıl gerçekleştireceğinizi gösterir ve arkasındaki algoritmalara bir giriş sağlar [21]. (OPENCV 4.5.4.58 sürümü kullanılmıştır.)

Mediapipe kütüphanesi

Mediapipe, yüz algılama görevini de kapsayan, Google tarafından desteklenen bir makine öğrenimi çözümüdür. Canlı ve akış verilerine odaklanır. Bu yüzden sağlamlığının yanında yüksek hız ile gelir. Ayrıca platformlar arası bir desteği var ve python istemcisi ile kullanılabilir [29]. Yüz algılama, çoklu el izleme, nesne algılama ve izleme ve çok daha fazlası gibi son teknoloji Makine Öğrenimi Modelleri yapmak için kullanılabilir. MediaPipe temel olarak, herhangi bir platformda çalışan sistemler için modellerin uygulanmasını ele almak için bir arabulucu görevi görür ve geliştiricinin sistemden ziyade modellerle denemeye daha fazla odaklanmasına yardımcı olur [30]. (Mediapipe 0.8.9.1 sürümü kullanılmıştır.)

NumPy

Dünyadaki sayısal hesaplama için en popüler kütüphanelerden biridir. Görüntü işleme, finans, biyoinformatik ve daha fazlasını içeren çeşitli disiplinlerde kullanılır [31]. (NumPy 1.21.2 sürümü kullanılmıştır.)



3. YAPILAN ÇALIŞMALAR

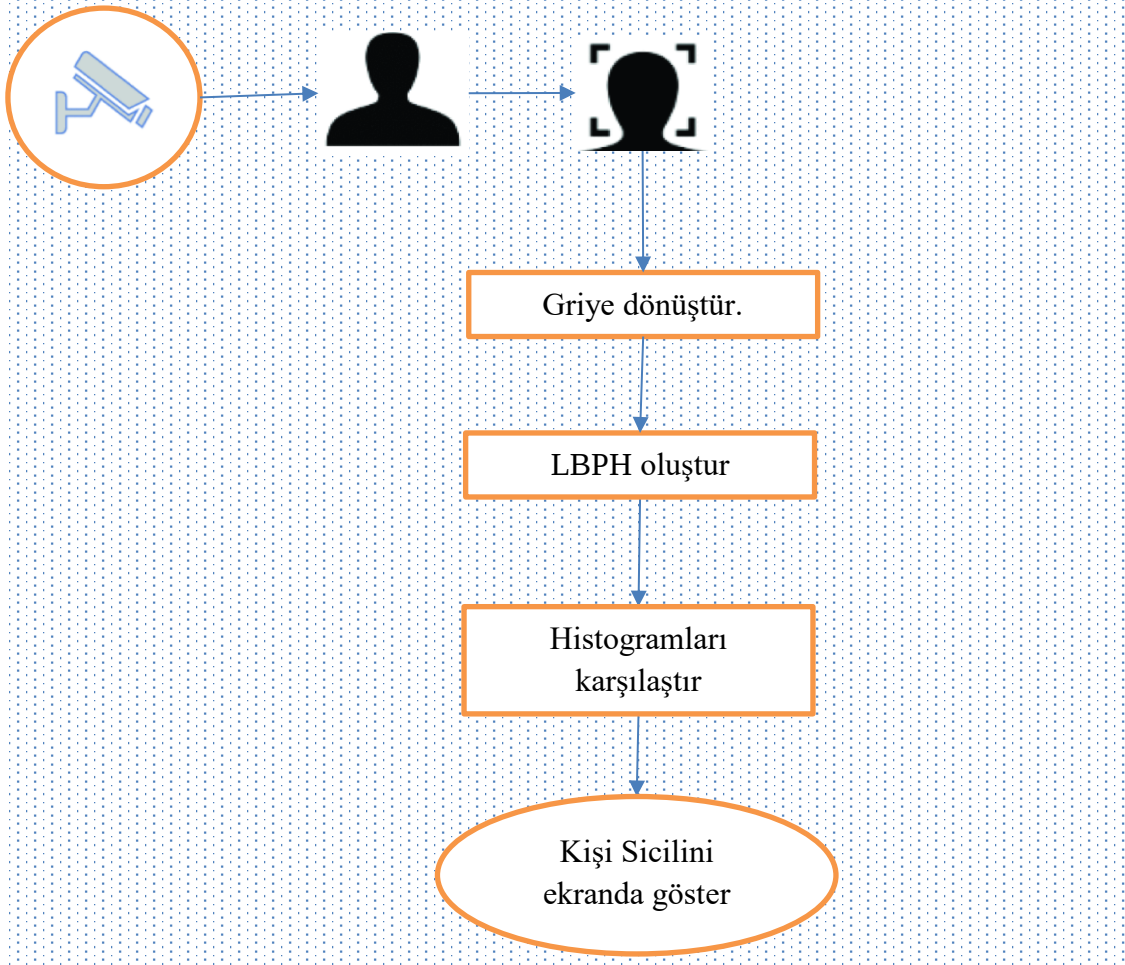
Basitten başlayıp daha kapsamlı maskeli yüz tanıma algoritmalarına doğru kademeli bir şekilde yüz tespiti ve tanıma algoritmaları üzerinde çalışılmıştır. Öncelikle maskesiz yüz tanıma üzerinde çalışılmış ve 90 kişiye ait farklı açılarda görüntülerin bir kısmı eğitim bir kısmı da test veri kümesi olarak kullanılmıştır. Daha sonra veri kümesi farklı yüzler ile zenginleştirilmiştir. Ayrıca veri kümesine sürekli ekleme yapacak ve tanıma işlemi için eğitimi yapabilecek kodlar oluşturulmuştur. Burada kullanılan eğitim veri kümesinin tanımaya ve performansa olan etkisi de irdelenmiştir ve bulgular başlığında detaylıca anlatılmıştır.

Maskesiz normal bir yüzün tanınması çok demode ve sıradanlaştığı için çalışmanın geri kalanında maskeli yüz tanımaya yönelik çalışmalar yapılmıştır. Bunu yapabilmek için çeşitli metot ve yöntemler üzerinde incelemeler yapılmıştır denenmiştir. Bunlardan bir tanesi yüz işaretlerinin tanıma da kullanılıp kullanılamayacağı ya da bir başkası maskeli olan yüzler için sadece iris tanıma yapılması gibi. Her biri üzerinde detaylıca düşünülmüş ve en mantıklı olanın maskesiz kalan kısımların tamamını tanımada kullanmak olduğu anlaşılmıştır. Çünkü yüze ait ne kadar öznitelik ve özellik var ise tanıma o oranda başarılı ve güvenilir olacaktır. O yüzden akış genel olarak alınan yüzü maskeli mi maskesiz mi diye ayırmakta maskeli ise maskesiz kısımları eğitim veri kümesi ile karşılaştırmaktadır. Maskeli değil ise normal maskesiz tanıma yapılmıştır. Bu süreçte yapılan tüm çalışmalar aşağıdaki alt başlıklarda verilmiştir.

3.1. Maskesiz Yüz Tanıma

Maskesiz yüz tanımada ilk yöntem olarak haar cascades yöntemi ile oluşturulmuş ön yüz verileri kullanılmıştır. Bu eğitim kümesinin ön yüzleri hangi açıya kadar tespit edebildiği test edilmiştir ve sonuçlar bulgular kısmına eklenmiştir. Maskesiz yüz tanımada yüz tespitinin ve tanıma başarısının artması için eğitim veri kümesinin çeşitliliği ve sayısı oldukça önemlidir. Diğer önemli bir nokta eğitim veri kümesindeki görüntülerinin kalitesi ve dolayısı ile öznitelikleridir. Bu noktada LBPH algoritması kullanılmıştır ve başarısı oldukça yüksektir. En azından ortam parlaklığından daha az etkilenmektedir.

Maskesiz tanıma işleminin gerçekleşmesi için bazı aşamaların yapılmış olması gerekmektedir. Bunlar: görüntüdeki yüzün tespiti, özelliklerinin çıkarılması, veri kümesi oluşturulması ve eğitimi ve sonrasında tanıma işlemidir. Her biri aşağıdaki alt başlıklarda ayrı ayrı anlatılmıştır. Maskesiz bir yüzü referans alacak olursak çalışmada yapılan yüz tanıma kabaca Şekil 3.1'deki gibidir.



Şekil 3.1. Maskesiz tanıma akış şeması

Maskesiz yüz tanıma akışı kısaca şöyle açıklanabilir: Öncelikle video, kamera ya da fotoğraf biçimindeki görüntü girdi olarak alınır. Görüntü içerisinde yüz olup olmadığı tespit edilir ve yüz var ise bu kısım alınıp griye dönüştürülür. Çünkü LBPH algoritması gri görüntüler üzerinde tasarlanmış bir algoritmadır. Gri görüntü LBP ile histograma dönüştürülür ve daha önce eğitim veri kümesine ait histogramlar ile karşılaştırılır ve belirtilen benzerlik oranının üzerindeki en yakın histogramın numarasını, sicilini, ismini ya da nasıl kaydedildi ise onu çıktı olarak döner.

3.1.1. Yüz tespiti

Yüz tespiti tanıma için olmazsa olmaz ve en önemli adımlardan biridir. Öyle ki yüz tespiti yapılmadan tanımadan söz edilemez. Bunun için tespit başarısı neredeyse %100 olmalıdır. Bu çalışmanın maskesiz yüz tespit kısmında yüz tespiti için haar cascades yöntemi ile oluşturulmuş ön yüzler kullanılmıştır ve algoritma bu yönde eğitilmiştir. Yüz tespiti hem eğitim hem test hem de çalışan bir sistemde olmalıdır. Çünkü ilgilendiğimiz kısım insan yüzü olduğu için tespit edildikten sonra sonraki işlemler devam etmelidir.

Çalışmada iki farklı yüz tespit veri kümesi kullanılmıştır. Bunlardan ilki daha az gelişmiş kapsamlı olanı diğeri ise daha gelişmiştir. Öyle ki ilk dosya maskeli yüzleri tespit noktasında %35 gibi bir başarıya sahip ve -30 ile +30 derece açıdaki yüzleri güvenli bir şekilde tespit edebilmektedir. İkinci ise çalışmadaki test veri kümesinde %100 tespit yapabilmektedir. Yüz algılama modelinin tahminini başarılı olarak değerlendirmek için gereken 0 ile 1 arasındaki minimum algılama güvenidir. Varsayılan değeri 0,5'tir (yani %50), bu, tahmin güveni 0,5'ten düşük olan tüm algılamaların varsayılan olarak yok sayıldığı anlamına gelir [32].

Yüz tespiti kısaca şu şekilde gerçekleşir: Öncelikle içerisinde yüz olan ve olmayan görüntüler ikiye ayrılır. Yüzün bulunduğu görüntüler pozitif ve yüzün bulunmadığı görüntüler ise negatif olacak şekilde makine öğrenmesi ile oluşturulacak dosya, yani yüz özniteliklerinin bulunduğu veri kümesini oluşturur. Bu veri kümesi sayesinde yeni verilen bir görüntü de yüze ait özelliklerin bulunup bulunmadığı tespit edilir. Yüz tespiti için kullanılacak olan veri kümesi ya da dosya kullanılacak amaca yönelik bilgiler yani görüntüler içermektedir. Örneğin sadece belli açılardaki yüzün tespiti yapılmak istenirse sadece o açılarda görüntüler, bıyıklı insan görüntüleri tespit edilecekse sadece bıyıklı insan görüntüleri ile eğitimin yapılması gerekmektedir. Eğer ne olursa olsun yüz tespiti yapılacak ise de ne kadar farklı yüz olursa tespit o kadar başarılı olur.

3.1.2. Özellik çıkarma

Bir görüntüde önce yüz algılama yapılır. Daha sonra yüze ait kaş, göz, burun, çene, ağız gibi özellikleri çıkarılır, buna özellik çıkarma denir. Özellik çıkarımı günümüzde birçok yöntem ile yapılabilir. Özellik çıkarma, bir yüzü tanımanın en önemli adımlarından belki

de birincisidir. Bu çalışmada, öznitelikler LBPH algoritması kullanılarak çıkarılmıştır. LBP daha önce de açıklandığı gibi yüzün değişen bölgelerini keskinleştirmiş olur ve dolayısı ile yüz özellikleri çıkarılmış olur. Burada bilmeliyiz ki yüze ait her bir organ (kaş, göz, burun, dudak vb.) bulunduğu bölgede belirli olduğu için bu başlangıç, bitiş ve detayları histogramlarda korunmuş olur.

3.1.3. Veri kümesi oluşturma

Veri kümesi karışıklık olmaması adına iki anlamda anlaşılmalıdır. Birincisi yüzün tespiti için gerekli olan veri kümesi bunlar yüzün bulunduğu pozitif ve yüzün bulunmadığı negatif görüntülerdir. İkincisinde ise tanıma için kullanılacak olan veri kümesidir. Bu veri kümesinin ise bir kısmı eğitim için bir kısmı da test için kullanılmıştır. Oluşturulan algoritma ve kodun başarısını ve performansını ölçmek için mutlaka eğitim için kullanılan eğitim veri kümesi ile bu sistemde tanıma için kullanılacak test veri kümesinin farklı görüntüler olması gerekmektedir. Aksi halde aynı olmaları halinde her iki görüntünün de histogramları bire bir aynı olacağı için tanımanın olması gerçek başarıyı göstermeyecektir.

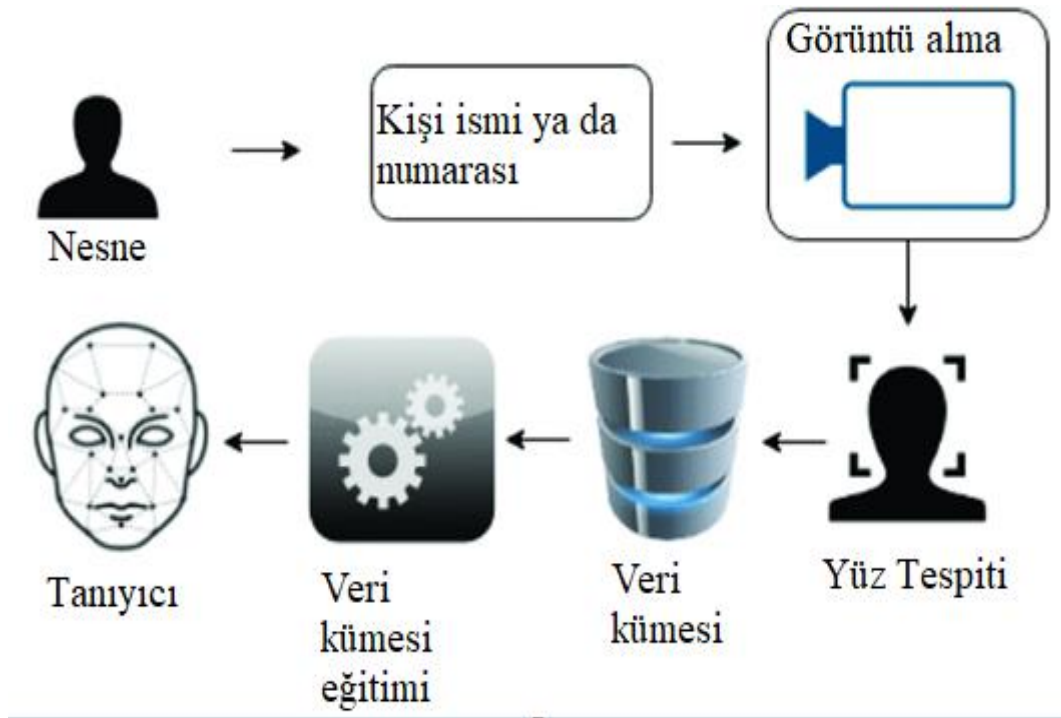
Yüz tanıma için kullanılacak veri kümesinde ilk etapta canlı video akışından yüzü algılamak için Haar Cascade sınıflandırıcı kullanılır. Ayrıca, Haar kademeli sınıflandırıcılar, canlı bir sistemde yüz algılama için daha az hesaplama gerektirir ve daha hızlı çalışır. Çalışmada saniye ve görüntü sayısı ayarlanabilmektedir ve video akışından istenilen saniye aralıklarla istenilen sayıda eğitim veri kümesi oluşturulabilir. Veri toplama sırasında daha iyi bir veri kümesi oluşturmak için farklı baş pozisyonu önerilir. Her görüntü dosyasının etiketi, kişinin adı, kimliği gibi bilgilerle kaydedilebilir. Ancak bunlar her kişi için özel olmalıdır.

Veri kümesi oluşturulurken eğitim veri kümesi ne kadar farklı ve çeşitli olur ise tanıma başarısı da artacaktır. Ancak kişiye ait ne kadar görüntüsünün olması gerektiği sistemden sisteme değişiklik gösterebilir. Çünkü eğitim kümesini çok fazla artırmak her zaman mantıklı olmayabilir. Çünkü tanıma algoritmasının her çalışması belirli bir iş ve süre demektir. Bu anlamda mutlaka oluşturulan yüz tanıma sisteminin performansı göz ardı edilmemelidir.

3.1.4. Veri Kümesi eğitimi

Veri kümesi eğitimi tanıma işlemi öncesi oluşturulması gereken ve yüz özniteliklerinin bulunduğu dosyanın oluşturulma aşamasıdır. Canlı bir görüntü tanınması için bu dosyada belirtilen sınırlarda bir yüz histogramı ile eşleşmelidir. Bu adımda hem ön yüzü hem de çeşitli açılardan tanıyabilen LBPH kullanılır. Etkili olduğu kadar basit bir doku operatörüdür. Hedef görüntünün pikselleri, her pikselin komşuluğu eşiklenerek etiketlenir ve daha sonra sonuç ikilik sayı olarak alınır. Yerel İkili Modelin (LBP) yönlendirilmiş gradyanların histogramları modeliyle entegre edilmesi, birkaç veri kümesinde algılama performansını önemli ölçüde artırır [33].

İlk olarak, LBPH yüz özelliklerini vurgulayarak gerçek görüntüyü daha uygun bir şekilde yorumlayan bir ara görüntü oluşturur. Burada, eğitilmiş bir tanıyıcı oluşturmak için OpenCV yüz tespit işlevi kullanılır. Bu fonksiyonda önce veri kümesi dizinine giden yol verilir, ardından görüntü dosya adı bölünerek bazı etiketler ve kimlikler oluşturulur. Tanıyıcı modeli oluşturduktan sonra ileride kullanılmak üzere yüz verilerini içeren bir dosya oluşturulur. Şekil 3.2.'de veri kümesi oluşturma ve bu veri kümesini eğitim süreci gösterilmektedir [34].



Şekil 3.2. Veri kümesi oluşturma ve eğitim süreci [34]

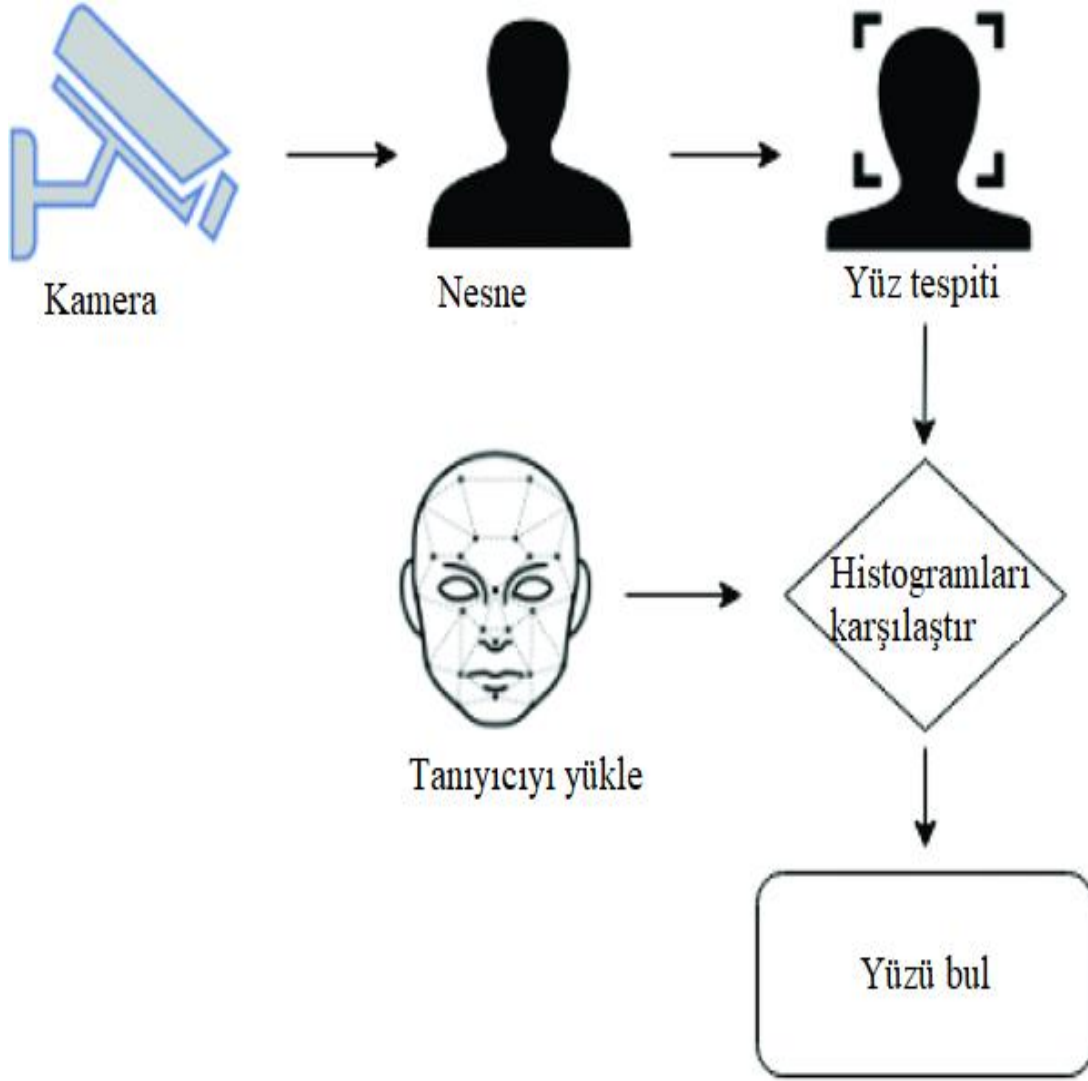
Burada Tanıyıcı eğitim veri kümesinin yüz özelliklerinin bulunduğu bir dosyadır. Bu aşamadan sonra tanıma için gerekli altyapı oluşturulmuş oldu ve yüz tanıma adımına geçildi.

3.1.5. Yüz tanıma

Yüz tanıma adımı, yüz tanıma sisteminin son noktasıdır. Yüz tanıma, yüz algılandıktan sonra yüzün özelliklerini çıkararak bir yüzü tanıma yöntemidir. Yüz tanıma, bir kişinin gözetimi, güvenliği ve kimlik tespiti gibi çok farklı ve çeşitli uygulamalarda kullanılabilir [35]. Tanıma işleminde test edilecek ya da canlı alınan görüntü içindeki kişiyi tespit etmeye çalışırken her bir histogram ile ayrı ayrı karşılaştırılır. En çok benzeyen görüntünün ID numarası çıktı olarak verilir ve kişiyi bulmuş olunur. Bu aşamaya gelmeden önce tanınacak görüntü tıpkı eğitim veri kümesi gibi tüm adımdan geçer. Son adımda sadece histogramlar karşılaştırılır.

Şekil 3.3, yüz tanımanın adımlarını göstermektedir. Bu adımda sistem, uygun bir konuma yerleştirilmiş bir video kameradan ya da bilgisayar kamerasında insanların görüntüsünü alabilir. Sistem daha sonra Haar Cascade sınıflandırıcıyı kullanarak kameranın devam eden video akışından insanların yüzlerini algılayabilir. Algılanan yüzler daha önce eğitilmiş hazır bir veri kümesiyle karşılaştırılır. Eğitim veri kümesindeki bulunan yüzler için, görüntüyü tanımlamak üzere bir özel histogram oluşturulur. Yüz tanıma işlemi için bir görüntü girdi olarak verildiğinde, bu görüntüyü temsil etmek için de bir histogram oluşturmak için aynı süreç devam eder. Girdi insan yüzünü histogramını tanıyıcıdaki diğer histogramlarla karşılaştırmak için farklı metotlar uygulanabilir [34].

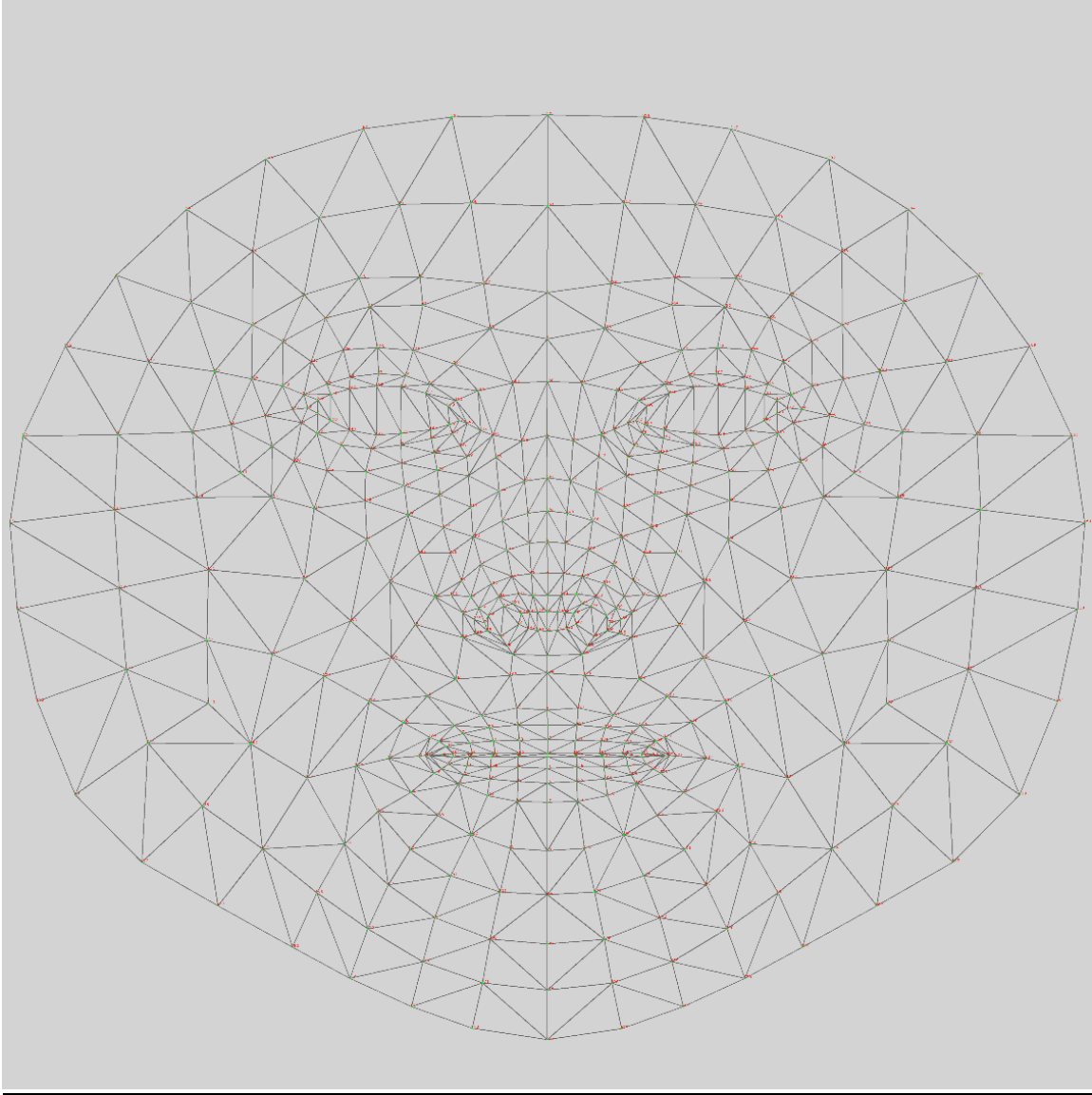
Burada histogram1i, histogram2i sırasıyla veri kümesindeki görüntünün ve tanıma için verilen girdi insan yüzünün histogramlarıdır. Algoritma, en düşük mesafeye sahip görüntünün etiketinden ismi ile ya da numarası ile sonuçlanır. Şimdi, tanıma sonucunun doğru olup olmadığını öğrenmek için bir "eşik" belirlenir ve hesaplanan mesafe önceden tanımlanmış eşik ile karşılaştırılır. Mesafe eşikten daha düşük olduğu durumda tanıma sonucu doğrudur denilebilir [34].



Şekil 3.3. Yüz tanıma [34]

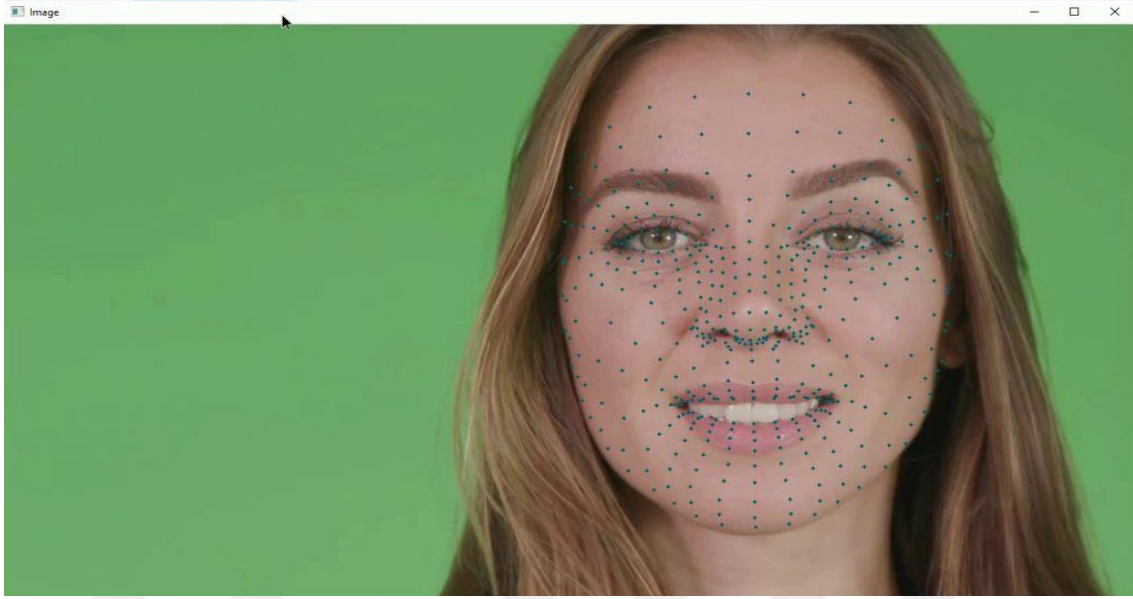
3.2. Yüz İşaretlerinin Çıkarılması

Yüz işaretleri (face landmarks) günümüzde çoğu uygulamada ve platformda yüzü filtreleme, fotomontaj, yüze bir şeyler ekleme vb. gibi yüze yapılabilecek aklımıza gelen çoğu işlemler için kullanılmaktadır. Mediapipe'e ait bu yapı ön yüze ait 468 tane yüz işareti vermektedir [36]. Görüldüğü üzere yüze ait özniteliklerin özelliklerin yoğun olduğu bölümlerde yoğunlaşmış noktalar mevcuttur.



Şekil 3.4. Mediapipe yüz noktalarının yerleri [36]

Şekil 3.4.'de her nokta aynı yeri temsil etmektedir. Örneğin 70 numaralı nokta her yüzde sağ kaş sonuna denk gelir. Bu yüzden sabit noktaların konumları değişmeyeceği için ön yüzün tanınmasında kullanılabilir durmaktadır. Fakat yapılan incelemeler ve araştırmalar sonucunda tek başına yeterli olmayacağı sonucuna varılmıştır. 2 nokta arasındaki mesafe görüntünün uzaklığına ve yakınlığına göre değişeceğinden direk mesafe yerine ölçeklenmiş bir değer olarak tanıma da fayda sağlayabilir. Ayrıca ölçeklenmiş bu değer farklı kişilerin de aynı olabilir. Bu yüzden tek başına tanımda yetersiz kalacaktır ancak yardımcı görev alabilecektir. Bu sebeplerden dolayı maskeli yüz tanıma da kullanılmamıştır ve maskeli yüz tanıma için farklı bir yöntem kullanılmıştır.



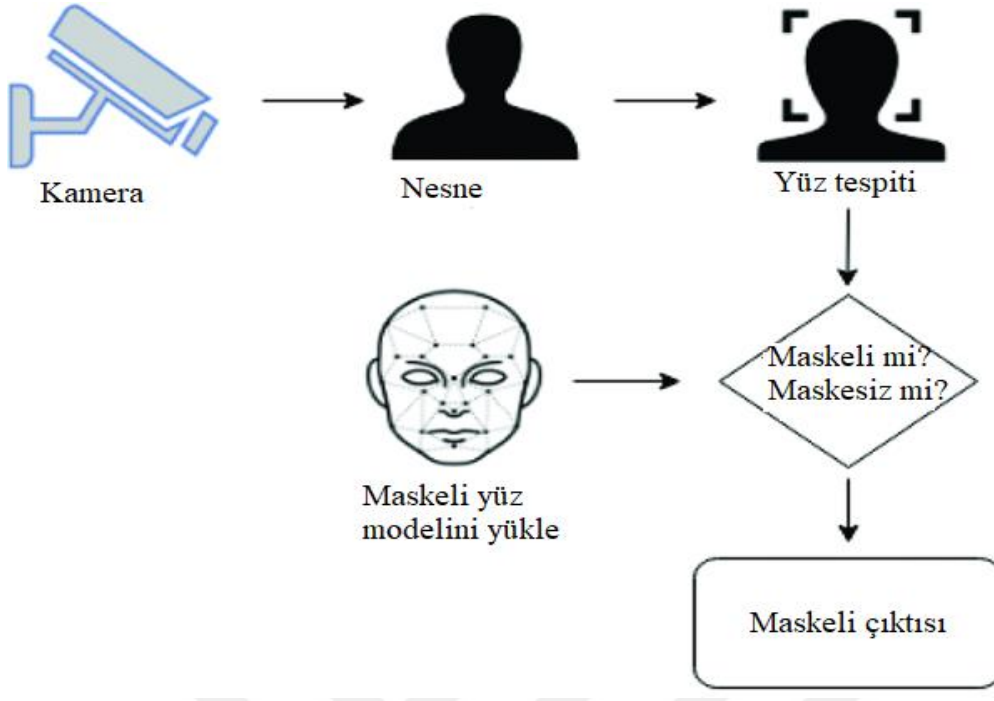
Resim 3.1. Yüz işaretlerinin insan yüzündeki gösterimi [37]

3.3. Maskeli-Maskesiz Ayrımının Yapılması

Maskesiz tanıma da yapılan testler sonucunda LBP yönteminin doğruluğundan ve yapılan incelemeler sonucunda tanıma başarısı artırılmıştır %99lara getirilmiştir. Fakat bu hali ile maskeli yüz tanımda başarısı %34lere düşmüştür. Bunun sebebi incelendiğinde ilk etapta maskeli yüzlerin tespiti yapılamadığı görüldü ve tespit edilen yüzlerin ise bir kısmını yanlış tanıdı. Bu yüzden kullanılan veri kümesi değiştirildi. Yeni yöntemde maskeli ya da maskesiz olmaksızın tüm yüzlerin tespiti yapılabildi.

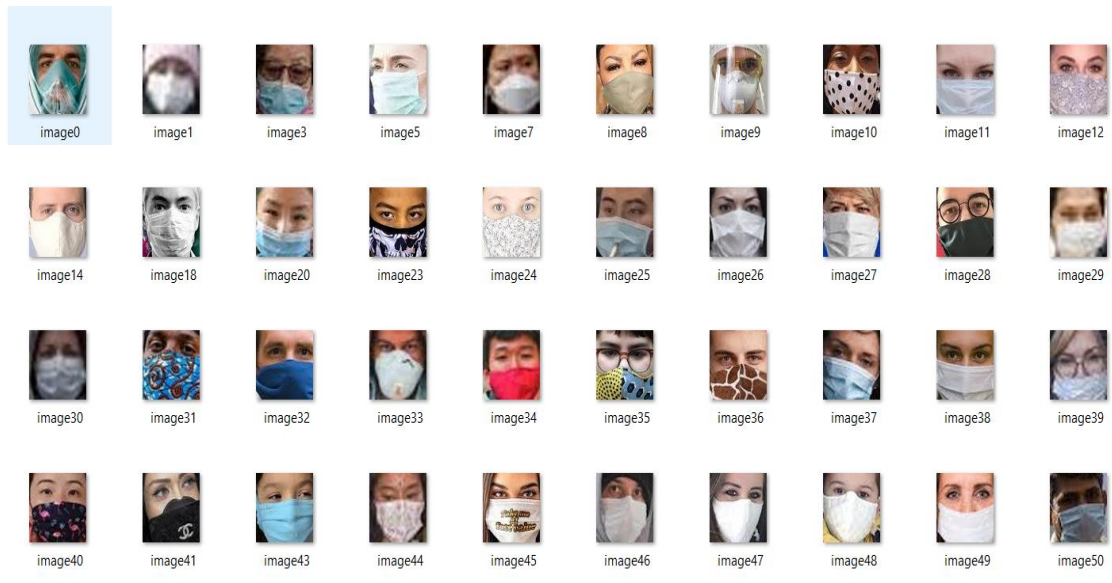
Maskeli yüz kavramını yazılımda kullanabilmek için maskeli ve maskesiz ayrımının yapılması gerekmektedir. Bunun için tıpkı en başta yapılan normal yüzü bulan model gibi benzer şekilde maskeli yüz modeli oluşturulması gerekmektedir. Bu her ikisini ayırt etmek için gereklidir. Bu modeli oluşturabilmek için mutlaka maskeli ve maskesiz insan yüzü görüntülerine ihtiyaç vardır.

Maskeli yüzleri tespit etmek istediğimiz için maskeli yüzleri pozitif ve maskesiz yüzleri negatif görüntü olarak düşünüp eğitimi yapabiliriz. Bu eğitimden sonra oluşturulan model verilen bir insan yüzünün maskeli mi yoksa maskesiz mi olduğunu tespit etmektedir. Bu tespitin doğru ve başarılı olması için veri kümesinin artırılması ve yanlış ayırt edilen görüntüleri doğru kümede tekrar eğitilmesi gibi adımlar uygulanabilir.

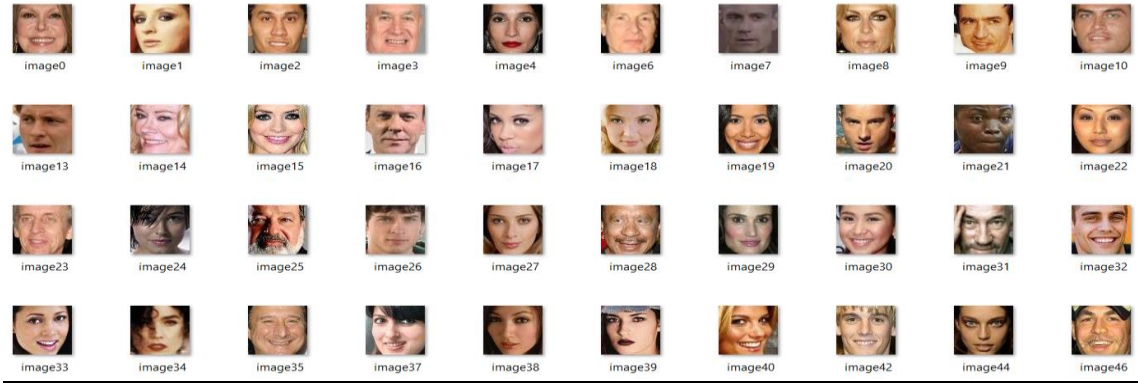


Şekil 3.5. Maskeli yüz tespit akış şeması

Şekil 3.5’de görüldüğü üzere maskesiz yüzleri zaten tanınabildiği için koda bu aşamada maskeli yüzleri de tespit edebilme yeteneği eklendi. İlgili kod parçası daha önce belirtilen yüz tespit yöntemi ile başarılı sonuç vermediğinden Mediapipe kütüphanesi kullanılarak bu sorun çözülmüştür. Dolayısı ile mevcut durumda tespit edilen yüzün maskeli mi maskesiz mi olduğu ayırt edilebilmektedir.



Resim 3.2. Maskeli pozitif görüntülerin bazıları [41]



Resim 3.3. Maskesiz negatif görüntülerin bazıları [41]

3.4. Yapay Maskeli Yüz Oluşturma

Maskeli yüz modelini oluşturmak için ihtiyaç duyulan maskeli yüzlerin bulunmaması durumunda ve maskeli tanınmanın yapıldığı durumda oluşturulan sistemi test etmek için maskeli yüz bulunmama durumunda ne yapılacaktı? Böyle bir sorunu ortadan kaldırmak için yüze yapay bir maske takma fikri kulağa hiçte kötü gelmemektedir. Eklenen yapay maske gerçeği gibi olursa elbette ki kullanılabilir.

Yapay maske ile yapılacak eğitim ve test işlemlerinde gerçeği temsil etmek zorunda olduğundan birebir aynı şartlar düşünülmelidir. Bunun için yüze farklı çeşitte maskeler konumlandırmak, bu anlamda sistemin doğruluğunu eğitim yönünden ve test etmek için oldukça önemlidir. Resim 3.4 bu çalışma neticesinde eklenmiş maskelerdir. Orijinal yüzler [38] numaralı kaynakta mevcuttur.



(a)

(b)

Resim 3.4. Yapay maske eklenmiş insan yüzleri (a) Cerrahi maske (b) N97 maske [38]

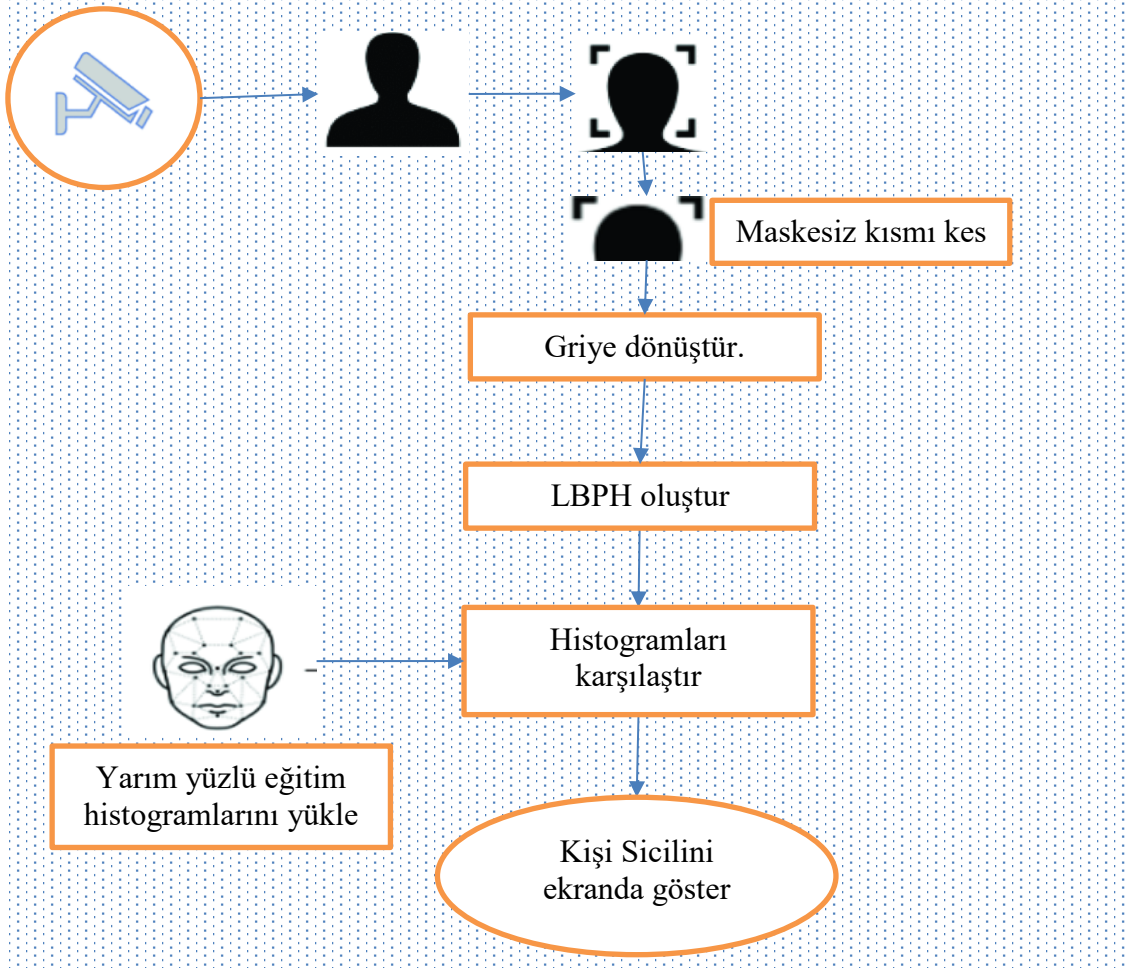
3.5. Maskeli Yüz Tanıma

Maskeli yüz tanıma işlemi için literatürde farklı metotlarda uygulamalar mevcuttur. Bu çalışmada tıpkı maskesiz tanıma olduğu gibi LBPH metodu kullanılmıştır. Bunun sebebi yapılan araştırmalar ve uğraşlar sonucunda bu metodun daha hızlı ve başarılı olduğudur. Çalışmada maskeli yüz tanıma başarı oranı %95 seviyesinde çıkmaktadır. LBPH metoduna karar verilmeden önce yüz işaretleri kullanımı, göz tanıma gibi çeşitli yöntemler düşünülse de kaybedilen her bir yüz özelliği tanımayı olumsuz yönde etkileyeceğinden mantıklı bulunmamıştır.

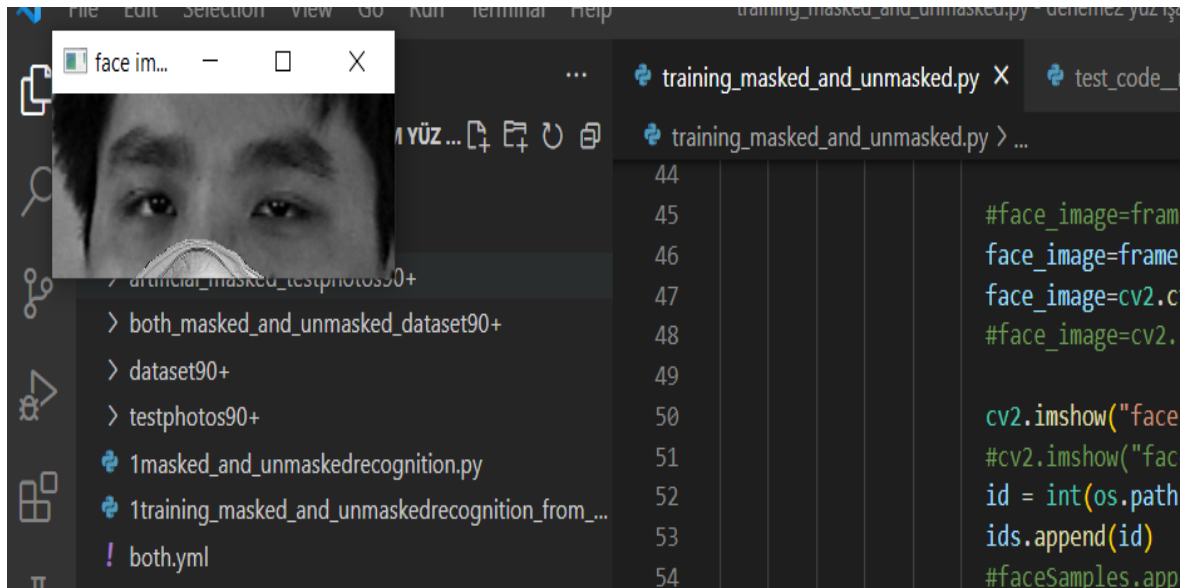
Bunun yerine sadece maskeli kısmın çıkarıldığı ve kalan yüzün tamamının tanıma kullanılacağı bir algoritma daha mantıklıdır. Bu şekilde daha kapsamlı tanıma olmaktadır. Çünkü belirtmiş olduğum kısımda zaten burnun üst kısmından alnın bitimine kadar kapsamakta ve gözleri tanıma işlemi için içerisinde kullanmaktadır. Bu aşamada geriye yüzün ilgili kısmı kalmaktadır. Bunu da yüz tespiti yapıldıktan sonra tüm yüzün üstten başlayarak 4/10'luk kısmını alınca istediğimiz ve tanıma için kullanacağımız yarım yüz görüntülerini elde etmemize yarayacaktır. Bu nokta kabaca maskenin bittiği noktaya denk gelmektedir.

Maskeli yüz tanıma işlemi için ayrıca tıpkı maskesiz yüz tanıma olduğu gibi veri kümesinden eğitim ve yöntemlere kadar tüm adımlar uygulanmalıdır. Buradaki tek fark karşılaştırma için gerekli olan kısımlar yüzün ilgili burun üzeri maskesiz kısmı alan parçasıdır. Daha iyi bir pratik olması açısından kaydedilen yüzler eğer tüm yüz eğitimi ile birlikte aynı zamanda yarım yüz eğitimi de yapılırsa iki farklı kod çalıştırılmak zorunda kalınmaz.

Bu çalışmada tek bir kod içerisinde yapılmaktadır. Yüz tespitinden sonra kişinin maskeli olduğu bilindiği için üst parça alınarak eğitim tamamlanır tanıma da ise test görüntüsü ya da canlı görüntüdeki yüz ile karşılaştırılır. Sonuç olarak oluşturulan maskeli yüz tanıma başarısı maskesiz gibi %99larda olmasa da %95ler gibi iyi bir başarıya sahiptir. Resim 3.5'de bulunan yarım yüze ait orijinal tam yüz [38]'de mevcuttur.



Şekil 3.6. Maskeli yüz tanıma adımları



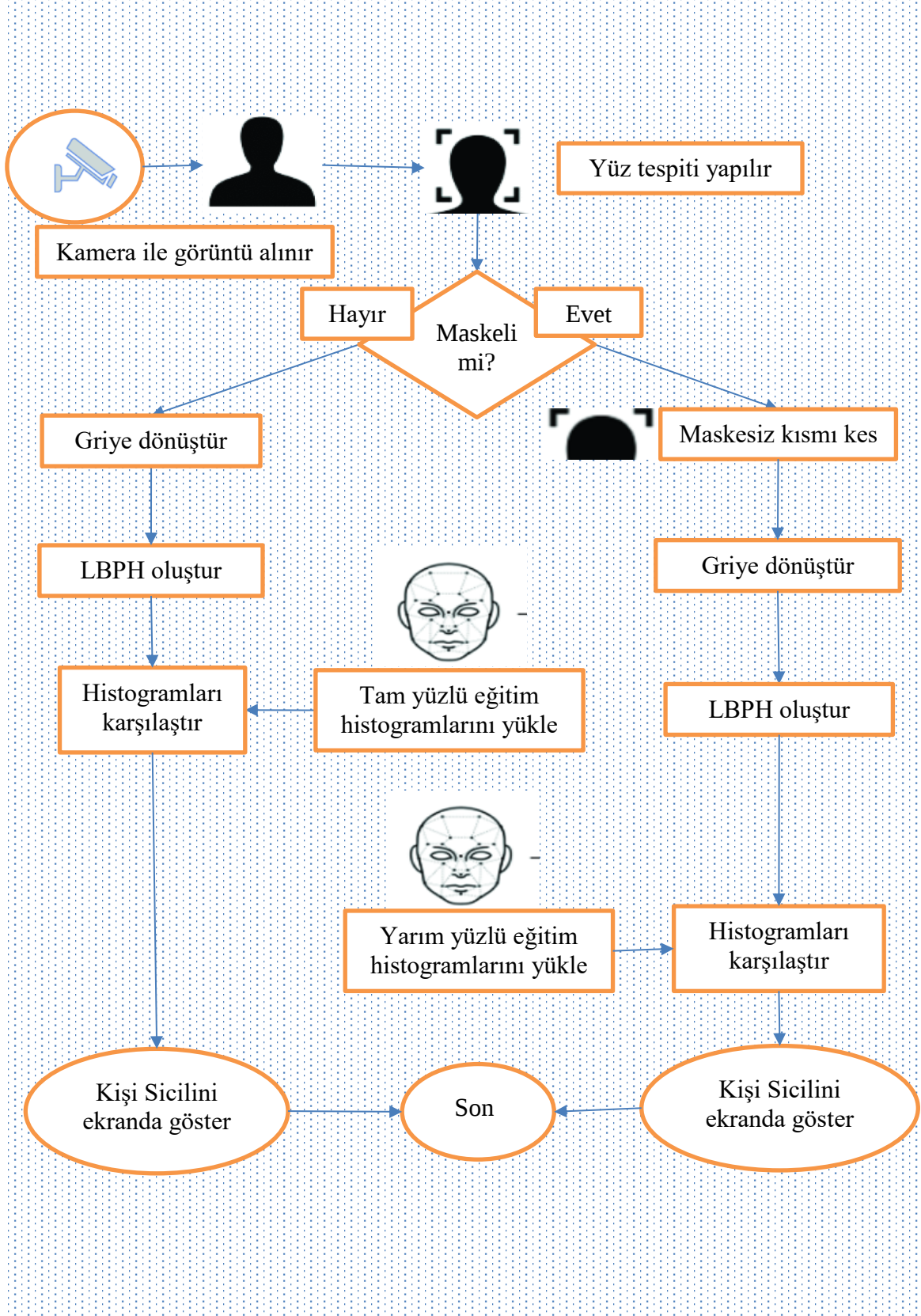
Resim 3.5. Maskeli yüze ait tanımda kullanılacak yüz parçası

3.6. Maskeli ve Maskesiz Tanımanın Birleştirilmesi

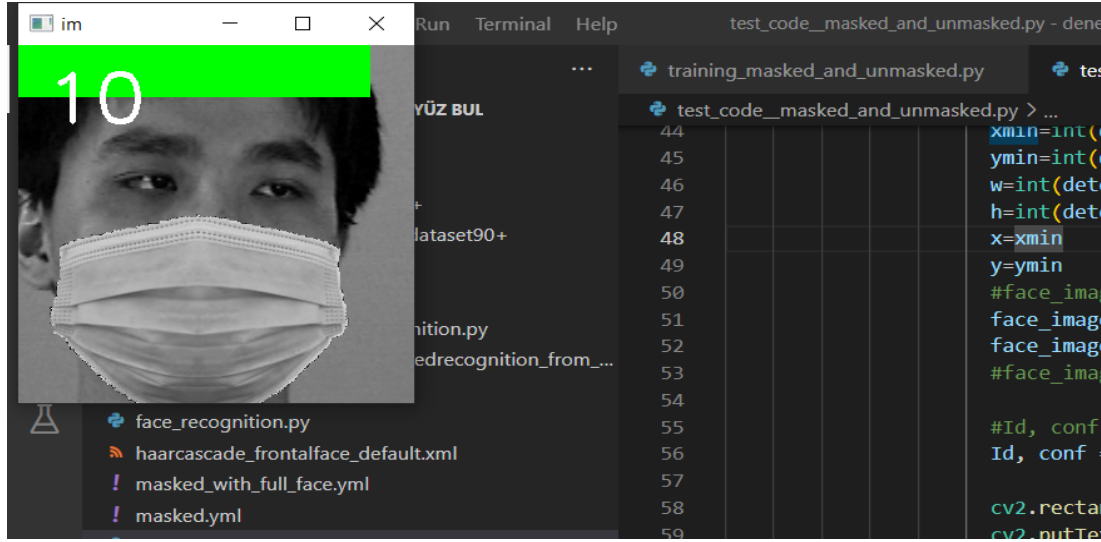
Çalışmanın son kısmına gelindiğinde ayrı ayrı maskesiz yüz tanıma ve maskeli yüz tanıma yapılabilmektedir. Ayrıca maskeli yüz tespit kısmı da hazırdır. Bulunmaması durumunda veya sistemi test etme durumunda kullanılacak olan yapay maskeli yüzlerde mevcuttur. Bu aşamada yapılması gereken tüm bu parçaları birleştirmektir. Genel akış şemasına bakılacak olursa Şekil 3.7. bu adımları genel olarak içermektedir.

Şekil 3.7’de belirtilen akış şeması eşliğinde son sistemden bahsetmek gerekirse, öncelikle görüntü kamera yardımı ile alınır daha sonra yüz tespiti yapılır. Tespiti yapılan yüzün maskeli olup olmadığına bakılır. Maskeli ise yüzün maskesiz kısmı kesilir, yüzün üst kısmının histogramı ile daha önce oluşturulmuş yine eğitim veri kümesine ait yüzün üst kısmının histogramlarının bulunduğu eğitim dosyası ile karşılaştırılıp yüzü tespit etmeye çalışır. Eğer maskesiz ise normal tüm yüz histogramları ile karşılaştırma yapar tanınan yüzün isim ya da numarasını verir. Her iki durumda da yüz tespiti, griye döndürme ve LBPH oluşturma ortak olarak yapılır.

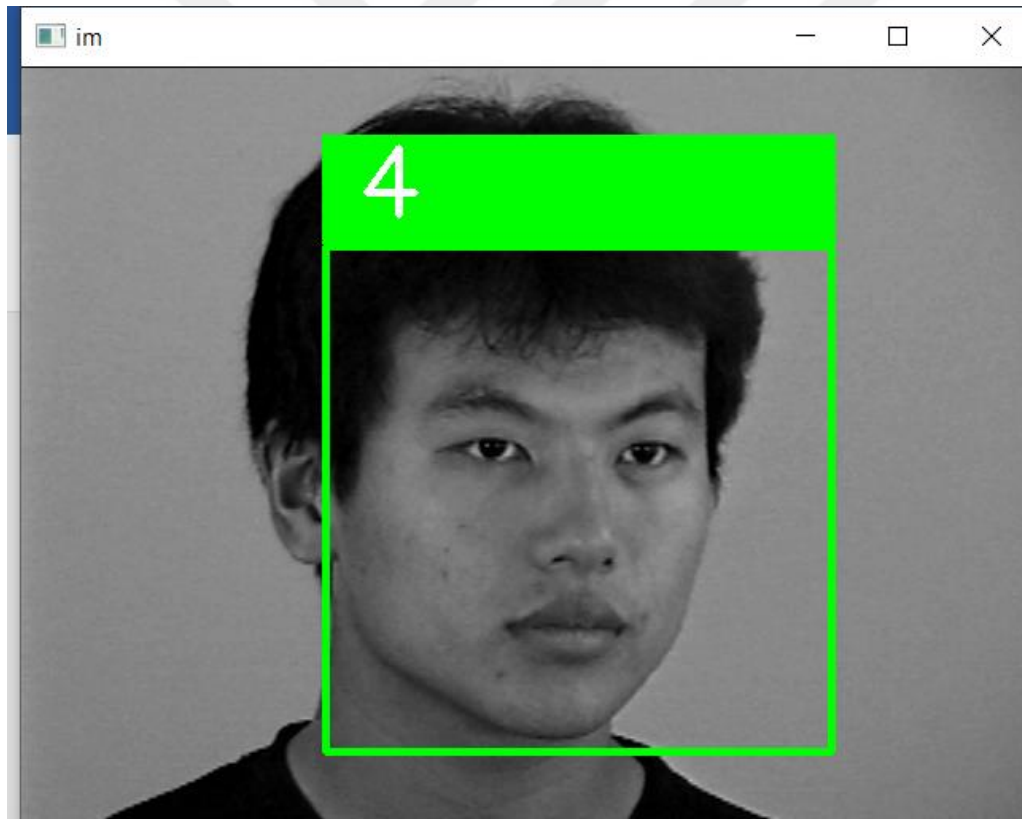
Tanıma işlemi maskesiz tam yüz ve maskeli için oluşturulmuş yarım yüzlerinde bulunduğu tek bir tanıma veri kümesi ve dosya da yapılabilirdi. Ancak alınan görüntü dosya üzerindeki histogramı en yakın görüntüyü bulana kadar tüm veri kümesinin tarayacağından zaman kaybı olacak ve doğal olarak kodun performansı düşecektir. Maskeli yüzün maskesiz yüz histogramları ile ve maskesiz yüzlerin maskeli yüz histogramları ile karşılaştırılarak oluşacak zaman kaybını önlemek için algoritmanın başına maskeli yüz tespit etme algoritması eklemiştir.



Şekil 3.7. Maskeli/maskesiz yüzlerin tanınması için oluşturulan algoritmanın akış şeması



Resim 3.6. Oluşturulan son algoritmaya ait maskeli yüze ait tanıma çıktısı



Resim 3.7. Oluşturulan son algoritmaya ait maskesiz yüze ait tanıma çıktısı

Resim 3.4, 3.5, 3.6, 3.7 bu çalışma neticesinde oynanmış görüntülerdir. Orijinal yüzler [38] numaralı kaynakta mevcuttur.

4. BULGULAR VE ANALİZLER

Maskeli ve maskesiz yüzlerin tanınması çalışması yapılırken bazı deneyler ve analizler yapılarak kullanılan yöntem, algoritma ve oluşturulan sistem üzerinde çeşitli sonuçlara ve bulgulara ulaşılmıştır. Bu bölümde bu bulgulara yer verilmiştir. Bazen karşılaştırmalı bazen de çıkan sonuçlara göre çeşitli yorumlar yapılarak çıktılar irdelenmiştir.

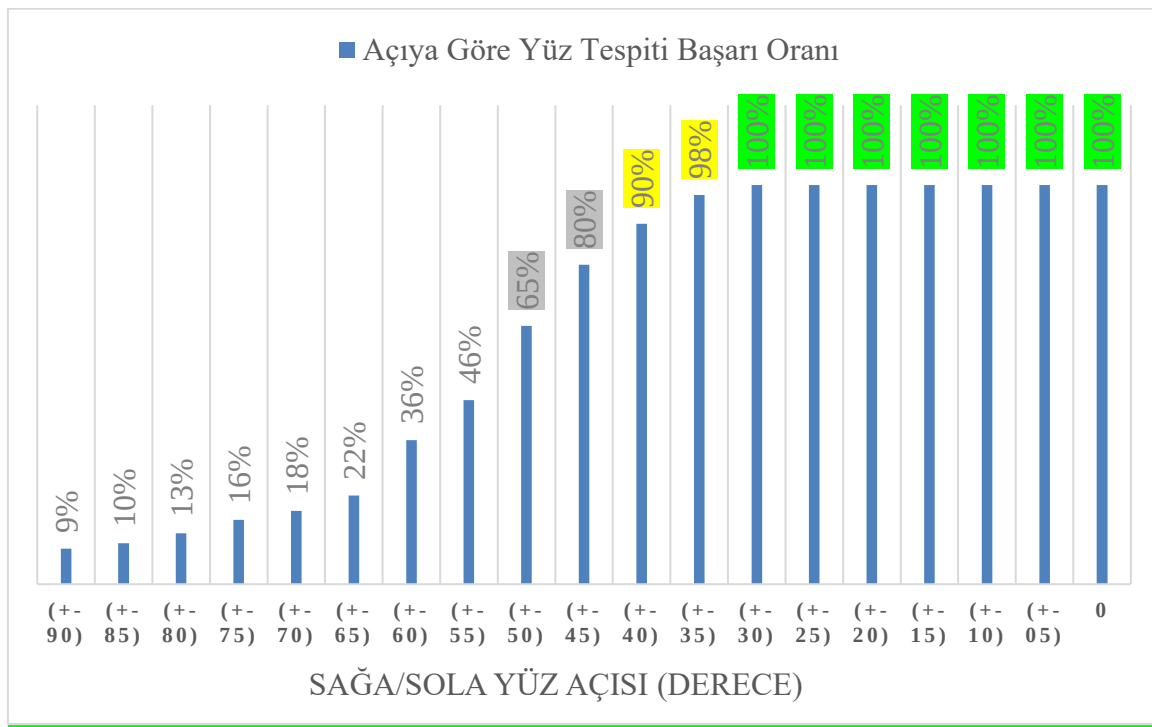
4.1. Yatay Eksende Ön Yüz Açısına Göre Yüz Tespiti Başarısı

Yüz tespiti tanıma öncesi mutlaka yapılması gereken kritik ve önemli bir adımdır. Aksi halde tanımadan söz edilemez. Çalışmanın ilk maskesiz yüz tespit kısmında Haar cascades sınıflandırıcıları ile oluşturulmuş [39]'de belirtilen ön yüz verileri kullanılmıştır ve OPENCV'ye ait olan bu veri kümesi eğitilerek girdi verilen bir yüzün kaç dereceye kadar tespit edilebildiği araştırılmıştır. Bu kapsamda -90 ile +90 derece arasındaki 5 derecelik farklarla Resim 4.1'deki gibi test görüntüleri kullanılmıştır.



Resim 4.1. Tek kişiye ait farklı açılardaki yüz görüntüleri [38]

Yüz tespit algoritmasında koşturulan test görüntülerine göre tespit başarı oranı Şekil 4.1’de verilmiştir. Bu analiz, tespit parametreleri olan ölçekleme faktörü (scaleFactor) 1.25 ve her bir aday dikdörtgenin kaç komşusunu tutması gerektiğini belirten (minNeighbors) parametresinin 2’ ye eşit olduğu duruma aittir. +-35 ve +-40 derecelerdeki görüntüler başarı seviyesi iyi sayılabilecek seviyede olduğu için tanıma algoritmasında eğitim veri kümesinde kullanılmıştır. Test görüntüleri ise tespiti kesin olması için -30 ve +30 derece arasında seçilmiştir. Her kişiye ait her açıdan 4 tane fotoğraf vardır. Örneğin 12. Kişinin 70lik açıda sağa ve sola 2şer fotoğrafı vardır. Her açıda $90 \times 4 = 360$ fotoğraf bulunmaktadır.

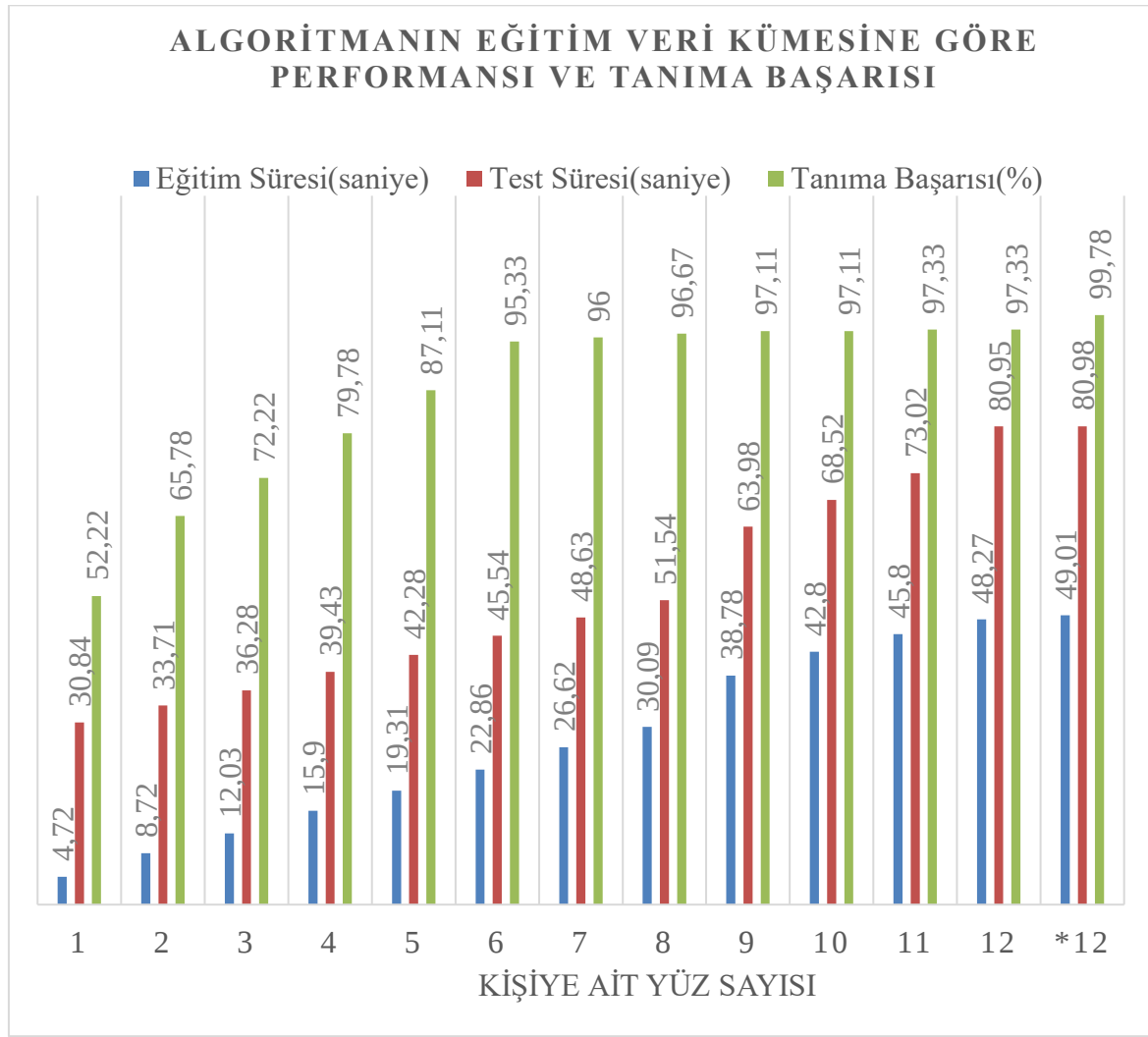


Şekil 4.1. Farklı açılardaki yüzün tespitindeki başarı oranı

Yüzün açı analizi maskesiz yüzlerde yapılmıştır ve eğitim için 1080 görüntü kullanılmıştır. Bu haliyle ilgili analiz sadece -30 ve +30 derece arasındaki yüzlerin tespiti yapabildiği için ve maskeli yüzleri tespit edemediği için yüz tespit veri kümesi değiştirilmiş olup Mediapipe’ye ait veri kümesi kullanılmıştır. Test kümesine 50 kişiye ait [40]’de belirtilen veri kümesinin de eklenmesi ile yapılan aynı test her açıda %100 başarı ile tespit edilmiştir. Bu analizden anlaşılmaktadır ki tanıma öncesi mutlaka tanınanın gerçekleşeceği ön yüz açısı belirlenmelidir ve ona göre veri kümesi kullanılmalıdır. Bu açı bazen kasti olarak belirli limitlerde olabilir. Böyle durumlarda veri kümesi bu açılardaki insan yüzlerinden oluşmalıdır. Şekil 4.1 sadece maskesiz yüzler üzerinde alınmış sonuçtur.

4.2. Eğitim Veri Kümesinin Tanımadaki Etkisi

Kişiye ait referans fotoğraf (eğitim veri kümesinde kullanılan) sayısı arttıkça o kişiyi tanıma başarısı da artmaktadır. Fakat artan sayı programın hızını düşürmektedir. Bu çalışmada tanınmanın çok etkilenmeyeceği ve kodun çok yavaşlamayacağı optimal görüntü sayısı tercih edilmiştir.



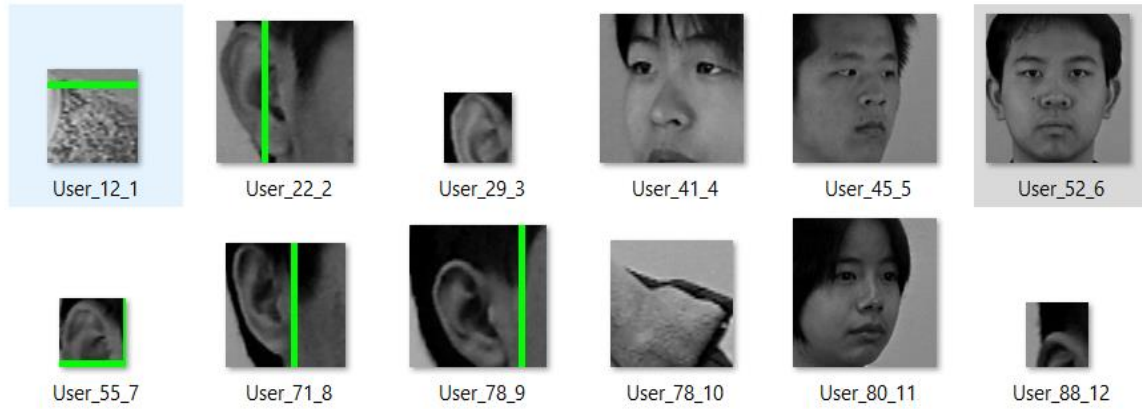
Şekil 4.2. Algoritmanın eğitim veri kümesine göre performansı ve tanıma başarısı

Analiz 90 kişiye ait 1080 eğitim veri kümesi görüntüsü ve 450 test veri kümesi görüntüsüne göre yapılmıştır. Ölçekleme faktörü (scaleFactor) 1,25 ve her bir aday dikdörtgenin kaç komşusunu tutması gerektiğini belirten (minNeighbors) parametresinin 2'ye eşit olduğu duruma aittir. Bu parametrelerde başarı %97,33 olarak bulunmuştur.

Başarısız olunan görüntüler üzerinde yapılan incelemede, başarısızlığın sebebi yüzün tanıma aşamasına gelmeden önceki tespit aşamasında olduğu sonucuna varılmıştır. Bu yüzden yüz tespitindeki parametreler bazında yapılan analiz sonucu en iyi başarı, ölçekleme faktörü, $scaleFactor=1,2$ ve her bir aday dikdörtgenin kaç komşusunu tutması gerektiğini belirten $minNeighbors=11$ olduğu durumda ortaya çıkmıştır ve %99,78 olarak ölçülmüştür. Şekil 4.2. sadece maskesiz yüzler üzerinde alınmış sonuçtur.

4.2.1. Tanınmayan ya da yanlış tanınan yüzlerin analizi

İlk etapta 450 adet test fotoğrafında 12 adet tanınmayan görüntü bulunmaktadır (başarı %97,33 olduğu durumda). Öncelikle tanınmayan yüzlerin niçin tanınmadığını analiz etmek için öncelikle yüz tespiti algoritmasının sonucuna bakıldı. Ve sonuç aşağıdaki gibidir. 12 adet tanınmayan fotoğrafın yüz tespiti çıktısı aşağıdaki gibidir.



Resim 4.2. Tanınmayan yüzler için tespit edilen test görüntüleri 12 adet

Hatalı görsellere baktığımızda 12 adet 8 adetinde yüz bulamadığı için, 1 tanesinde yüzü yarım bulduğu için, diğer 3 yüzü ise tespit etmesine rağmen tanıyamadığı görülmüştür. Çalışmanın bu kısmında kullanılan fonksiyonunda iyileştirme yapılmak için parametreler üzerinde çalışılmıştır ve bazı değişikliğin olduğu görülmüştür. Örneğin 12 adet hatalı görüntü ölçekleme faktörü, $scaleFactor=1,1$ iken olan sonuçlardır ve başarı %97,33'tür. Onu 1,25 yapınca tanınamayan yüz sayıları 9'a düşmüştür ve başarı %98'e yükselmiştir. Asıl enteresan konu bu 9 görüntü bir önceki tanınamayanlardan farklıdır. Yanlış tespit ve yanlış tanıma mevcuttur. Aslında bu parametreler yüz tespitini ilgilendirdiği için öncelikle sağlıklı parametrelerin bulunması gerekmektedir.



Resim 4.3. Tanınmayan yüzler için tespit edilen test görüntüleri 9 adet

Herhangi bir parametrede tanınmayan ya da yanlış tanınan görüntünün doğru ID ile eğitim veri kümesinde kullanılması hata oranını sıfırlayacaktır. Fakat bu yöntem mevcut sorunu çöze de canlı bir sistemi çözmeyecektir. Mevcut sorunu çözmek için bunu yapmak yerine parametre analizi yapılmıştır ve 4.3. başlıkta verilmiştir. Yüz tespitindeki parametreler bazında yapılan analiz sonucu en iyi başarı, ölçekleme faktörü, scaleFactor=1,2 ve her bir aday dikdörtgenin kaç komşusunu tutması gerektiğini belirten minNeighbors=11 olduğu durumda ortaya çıkmıştır ve %99,78 olarak ölçülmüştür.

Yüz tespiti tanımayı doğrudan etkilediği için tanımanın artması çok normaldir. Ancak burada şöyle bir soru çıkmaktadır: Eğitim veri kümesine her yeni eklenen kişi için parametre analizi mi yapılacak? Bu sorunun cevabı eğer yüzler veri kümesi olarak farklı bir algoritmadan çıkmışsa evet gerekecektir. Fakat yine bu yüz tespit algoritmasından geçtikten sonra eğitim için kullanılacaksa buna gerek kalmayacaktır.

En nihayetinde bu çalışma -30 ve +30 derece arasındaki test görüntülerini kullanmış olduğumuz ve OPENCV'ye ait Haar cascades sınıflandırıcıları ile ön yüz verileri kullanıldığı için yüz tespitinde Şekil 4.1'deki gibi sorunlar yaşanmaktadır. Çalışmanın sonunda yüz tespit veri kümesi değiştirildiği için tespitten kaynaklı hata en azından test edilen görüntüler için sıfıra inmiş oldu.

4.3. Yüz Tespit Parametrelerine Bağlı Tespit ve Tanıma Başarısı

Yüz tespit fonksiyonu, yüzü tespit ederken bazı parametreler almaktadır. Bu parametrelere göre yüz tespiti ve tanıma başarısı değişmektedir. Bu yüzden bu parametreleri bu çalışma için optimum değeri bulunmaya çalışılmıştır.

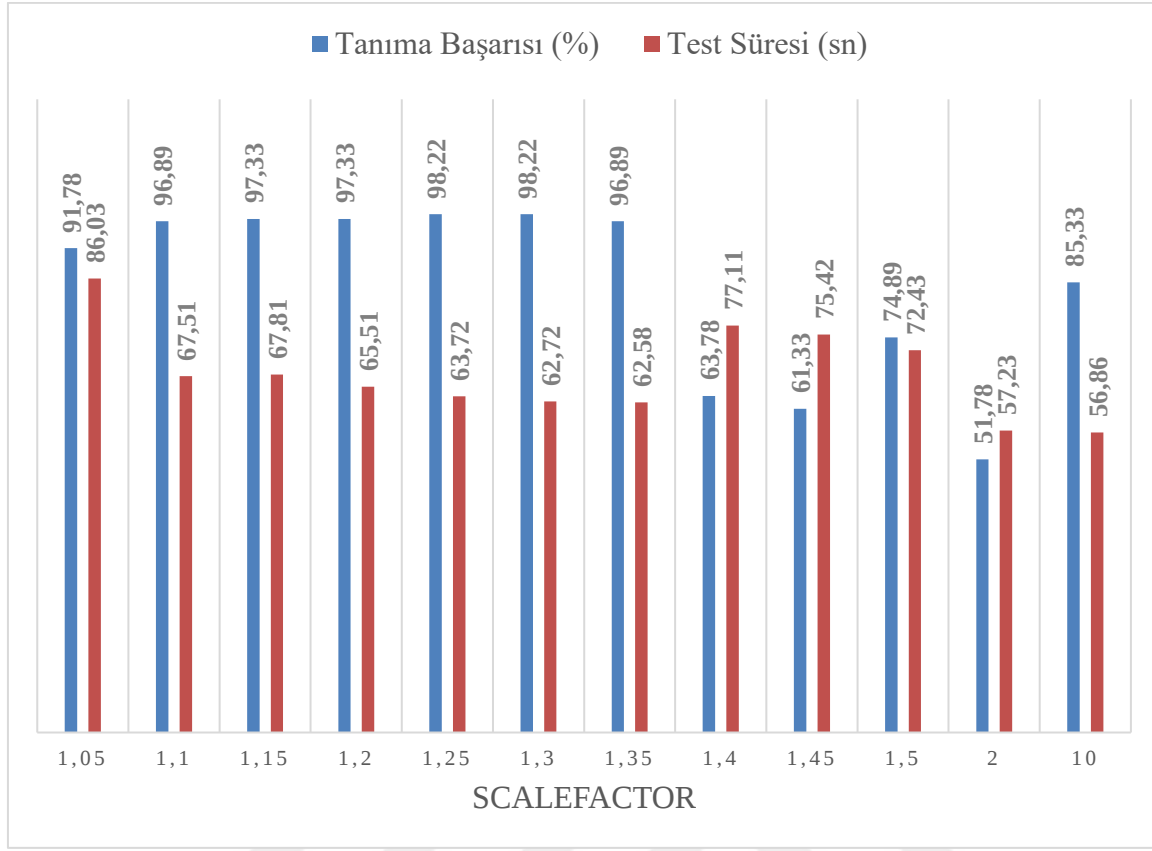
```
◆ detectMultiScale() [1/3]

void cv::CascadeClassifier::detectMultiScale ( InputArray      image,
                                              std::vector< Rect > & objects,
                                              double          scaleFactor = 1.1 ,
                                              int             minNeighbors = 3 ,
                                              int             flags = 0 ,
                                              Size            minSize = Size() ,
                                              Size            maxSize = Size()
                                              )
```

Şekil 4.3. Yüz tespit fonksiyonu ve parametreleri

- 1-görüntü (image): Nesnelerin algılandığı bir görüntüyü içeren CV_8U türündeki matris.
- 2-nesneler (objects): Her dikdörtgenin algılanan nesneyi içerdiği dikdörtgenlerin vektörü, dikdörtgenler kısmen orijinal görüntünün dışında olabilir.
- 3-görüntü ölçekleme faktörü (scaleFactor): Her görüntü ölçeğinde görüntü boyutunun ne kadar küçültüleceğini belirten parametre.
- 4-minNeighbors: Her aday dikdörtgenin onu tutmak için kaç komşusu olması gerektiğini belirten parametre.
- 5-flags: cvHaarDetectObjects işlevinde olduğu gibi eski bir basamak için aynı anlama sahip parametre.
- 6-minSize: Görüntüye ait algılanması gereken minimum nesne boyutunu ifade eder.
- 7-maxSize: Görüntüye ait algılanması gereken maksimum nesne boyutunu ifade eder.

Çalışmada minimum ve maksimum boyuta ait kısıt kullanılmadı. detectMultiScale fonksiyonu tespit edilen yüzlerin (x,y,w,h) konum bilgisini tutmaktadır. Tanımayı iyileştirmek adına scaleFactor ve minNeighbors parametrelerine bağlı tanıma başarısı analiz edilmiştir.

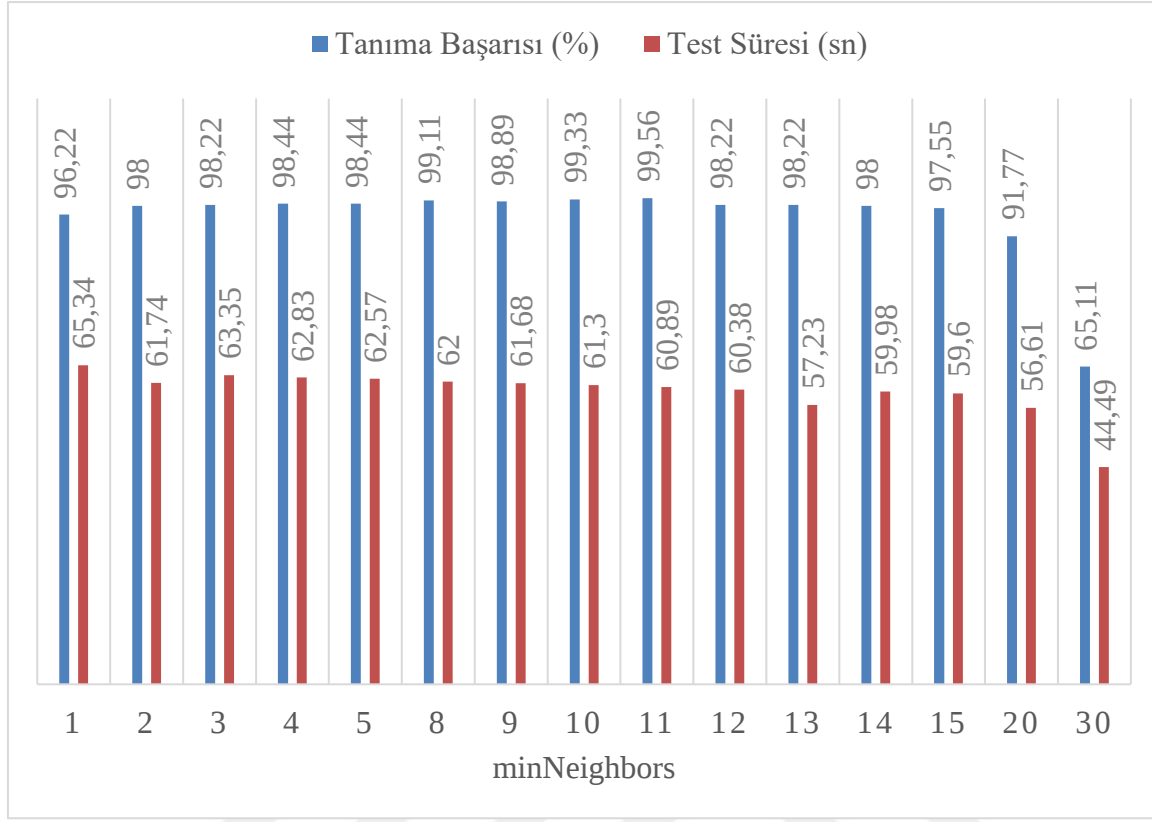


Şekil 4.4. Ölçekleme parametresine bağlı tanıma başarısı ve test süresi

ScaleFactor: Bu parametre aldığı görüntüyü ölçeklendirir ve “1” rakamından daha büyük bir değere sahip olmalıdır. Aksi halde kod hata verecektir. Ayrıca tanıma ya da daha doğrusu yüzü tespit için kullanılan bir parametre olduğu için bunun önemi daha çok negatif ve pozitif yüz bilgilerinin bulunduğu eğitim veri kümesi dosyası ile alakalıdır. Çünkü ilgili fonksiyon yüzü bulurken belirtmiş olduğumuz bu dosyayı referans almaktadır. Bu dosyada tanıtılan yüzlerin uzaklığı boyutu bulacağı yüzü de kapsamalıdır. Aksi halde fotoğrafta gerçekten yüz olsa bile uzaklık yakınlıktan dolayı bulamayabilir. Ya da eğer scaleFactor ölçeklendirmesi sonrası karşılaştırma yaptığımız yüz bilgilerinin kapsamı dışında kalırsa yüzü bulamaz. Biraz daha açmak gerekirse, çok büyük seviyede ölçeklendirme yaparsak yüz özelliklerini kaybederiz ve doğal olarak yüzü tespit etmesi zorlaşır ve imkansızlaşabilir. Bu çalışmada da test veri kümesine göre en iyi tanıma oranına ulaşacak ölçekleme faktörü için farklı değerler denenmiş ve Şekil 4.4’de belirtilmiştir.

MinNeighbors: Şekil 4.4’deki sonuçlar minNeighbors:2 olduğu durumda alınmış sonuçlardır. Şimdi de scaleFactor parametresinin yüz tespitinde en iyi olduğu değerde (1,25) bir diğer parametremiz olan minNeighbors’ı inceleyelim. Bu parametre tam sayı

olmalıdır.



Şekil 4.5. minNeighbors parametresine bağlı tanıma oranı ve test süresi

Tüm parametre analizi yapıldıktan sonra, tanıma oranı ve gerekli ise çalışma süresi göz önüne alınarak maksimum ya da optimum değerlerin seçilmesi gerekmektedir. Bu çalışmada optimum değerler olarak scaleFactor:1,25 ve minNeighbors:11 seçilmiştir. Aslında maksimum tanıma oranını ortaya çıkaracak daha da başarılı bir tanıma istenmesi durumunda bu parametreleri mantıklı limitler çerçevesinde kombinasyonlu denemeler yapılabilir. Örnek olarak minNeighbors:11 iken bu defa tekrar scaleFactor ile oynamak. Çeşitli örnekler yapıldı ve 11 iken scaleFactor'ü 1,3'e getirildi ve başarı %98'e indi. Fakat yine 11 iken scaleFactor'ü 1,2'ye getirildi, tanıma oranı muazzam arttı ve %99,78 olarak kaydedildi. Bu şekilde deneme ile belki %100 bile bulunabilir. Fakat daha büyük veri kümesinde bunu yapmak baya süre alacak bir iş olduğu için otomatik edilmesi bu anlamda fayda sağlayacaktır.

Bir üst bölümde belirtmiş olduğum hatalı yüz tespiti, hiç bulamamayı ya da yanlış tanımadan kaynaklı hataların büyük bir kısmı parametreler ile oynanarak gidermiş olduk. Canlı görüntü girişinde bu hatalara ek olarak bozuk görüntü ya da ön yüz açısının çok

sapmasından kaynaklı da olabilir. Ayrıca tanımada hatalı olan görseller veri kümesine eklenip tekrardan eğitilince bu sorun ortadan kalkmaktadır. Sonuç olarak yüksek başarı ölçekleme faktörü, scaleFactor=1,2 ve her bir aday dikdörtgenin kaç komşusunu tutması gerektiğini belirten minNeighbors=11 olduğu durumda ortaya çıkmıştır ve %99,78 olarak ölçülmüştür.

Şekil 4.4 ve 4.5'teki analizler ilk veri kümesine ait sonuçlardır. Her yeni eklenen yüze ve ver kümesine göre parametre analizi yapmak her zaman mantıklı olmayabilir. Bunun yerine daha geniş veri kümesi kullanılırsa tespitten kaynaklı hatalar neredeyse sıfıra indirilebilir. Buradaki tanınmayan yüzler yine yüz tespitinden kaynaklı başarısızlıktan dolayıdır.

4.4. Eğitim Veri Kümesinin Maskeli-Maskesiz Ayrımındaki Başarıya Etkisi

Bu analizde Resim 3.2 ve 3.3'de bulunan yüz sayılarına göre maskeli ve maskesiz yüzlerin doğru tespit edilip edilemediği incelenmiştir. Bu kapsamda maskeli ya da maskesiz yüz görüntülerini ayırt edebilmek için oluşturulan xml formatındaki eğitim dosyasının başarısı içindeki pozitif ve negatif görüntü sayısına belli oranda bağlıdır. Bununla birlikte içerisindeki görüntülerin çeşitliliğine de büyük oranda bağlıdır. Aşağıdaki inceleme için 632 tane maskesiz test yüzü ve 250 adet maskeli test yüzü kullanılmıştır.

Çizelge 4.1. Eğitimde kullanılan maskeli-maskesiz yüz sayısına göre kodun performansı

Adım	A	B	C (sn)	D (%)	E (%)	F (%)	G (sn)	H (%)	I (%)	J (%)	K (sn)
1	10	10	1,46	94,8	3,2	2	11,72	50,63	46,36	3	36,73
2	20	20	1,83	97,2	0,8	2	12,52	70,41	28,78	0,8	34,47
3	30	30	2,4	96	2	2	13,33	77,69	21,52	0,8	33,57
4	50	50	3,41	96,4	1,6	2	15,19	85,44	13,92	0,63	38,86
5	100	100	6,63	97,6	0,8	1,6	18,24	83,07	16,3	0,63	45,93
6	100	200	8,54	97,6	0,8	1,6	21,76	84,81	14,55	0,63	54,24
7	200	200	10,81	98	0,4	1,6	20,95	78,32	21,04	0,63	56,67
8	300	300	16,29	98,4	0,4	1,2	27,44	78,32	21,04	0,63	69,77
9	506	449	29,33	98,63	0	0,8	37,01	99,37	0	0,63	87,31

- A: Eğitimde kullanılan maskeli yüz sayısı,
 B: Eğitimde kullanılan maskesiz yüz sayısı,
 C: Modelin eğitilme süresi,
 D: Maskeli test kümesi için doğru ayırt etme başarısı,
 E: Maskeli test kümesi için yanlış ayırt etme oranı,
 F: Maskeli test kümesi için yüz bulamama oranı,
 G: Maskeli test kümesi için test süresi,
 H: Maskesiz test kümesi için doğru ayırt etme başarısı,
 I: Maskesiz test kümesi için yanlış ayırt etme oranı,
 J: Maskesiz test kümesi için yüz bulamama oranı,
 K: Maskesiz test kümesi için test süresi.

Son adımda (9. adım) başarının yüksek olmasının sebebi yanlış ayırt edilen görüntülerin de eğitim kümesine dahil edilmesidir. Bu şekilde yanlış ayırt etme ihtimalini neredeyse sıfıra indirilmektedir. Görüldüğü üzere eğitim veri kümesinin belli bir sınırdan sonra sayıdan çok tanımadığı fotoğrafı öğretmek düzeltilmektedir. En azından bu yöntem için. Burada şüphesiz test için kullandığımız görüntüler ile eğitim için kullandığımız görüntülerin kapsamı ve ortamları (küçük yüz, büyük yüz, renkli gri vb.) önem arz etmektedir.

9. adım sonrasında test veri kümesi ile yapılan test sonuçları Çizelge 4.2. ve 4.3'deki gibidir.

Kod, yapılan test sonucunda 632 adet maskesiz test görüntüsünün 628 tanesini maskesiz 0 tanesini maskeli belirtmiştir. Bu bilemediği 4 adet yüzü tespit edemediği anlamına gelmektedir. Bu noktadaki başarı oranı %99,37'dir (maskesiz yüzü doğru bilme olasılığı).

Çizelge 4.2. Maskesiz yüzü doğru bilme olasılığı

Toplam yüz sayısı	Maskeli yüz sayısı	Maskesiz yüz sayısı
632	0	628

Aynı testi maskeli yüz verisinin bulunduğu görüntülerle denediğimiz zaman: 3003 maskeli veri kümesi ile test gerçekleşmiştir ve 2962 tanesi doğru 2 tanesi yanlış ayırt edilmiştir. Başarı oranı %98,63 (maskeli yüzü doğru tahmin etme olasılığı)

Çizelge 4.3. Maskeli yüzü doğru bilme olasılığı

Toplam yüz sayısı	Maskeli yüz sayısı	Maskesiz yüz sayısı
3003	2962	2

Ayrıca yukarıdaki bulgular, min_detection_confidence'nin 0,5'e eşit olduğu durum için geçerlidir. Bu fonksiyon normal yüz tespitten farklıdır. 0,5 demek yüze benzerlik oranı en az %50 olması gerek demek. Bizde maskeli yüz test ettiğimiz için maskesize nazaran maskeli yüz tespitindeki başarının düşük olmasının sebeplerinden bir tanesi de: maskenin, yüzün benzerlik oranını %50'den daha fazlasını kapsaması durumudur.

4.5. Mevcut Maskesiz Yüz Tanıma Sisteminin Maskeli Yüze Tepkisi

Maskeli tanıma işlemini yapmadan önce yapılan incelemeler ve analiz sonucu mevcut maskesiz yüz tanıma sisteminin aynı işlemi maskesiz yüzler için yaptığında başarının %34 gibi baya düşük olduğu görülmüştür. Bunun öncelikli sebebi ilk veri kümesinde maskeli yüzlerin tespitindeki başarısızlıktı. Fakat bu noktada yüz tespit yöntemi değiştirilmesine rağmen tanıma oranı tüm yüz referans alındığında yine düşük çıkmıştır (%14,04). Bunun sebebi araştırıldığında ise göze çarpan ilk şey: Eğitim veri kümesinin maskesiz yüzlerden oluşmuş LBP Histogramlarına sahipti fakat canlı görüntü ve test görüntüsü maskeli yüzlerden oluşmaktadır. Bu sebepten dolayı maskeli kısım tüm yüzlerde aynı olduğu için yanlış tanımakta ya da tanınamamaktadır. Bu sorun maskeli yüzlerin tanınması için yarım yüzlerin histogramlarını oluşturup maskeli maskesiz ayrımının yapıldıktan sonra maskeli yüzler için bu yarım yüz histogramlarının kullanılması ile çözüldü ve başarı %95lere yükseldi.

Çizelge 4.4. Mevcut maskesiz yüz tanıma sisteminin maskeli yüz tanıma başarısı

Toplam test edilen yüz sayısı	Tanınan yüz sayısı	Tanıma oranı
2230	313	%14,04

4.6. Yüz Tanıma Başarı Oranları ve Karşılaştırılması

Başarı oranlarını vermeden önce her bir çalışmanın ayrı ayrı başarı oranları verilecektir ve son kodun başarısı verilecektir. Teorikte maskesiz yüz tanıma, maskeli yüz tanıma ve maskeli-maskesiz yüz tespitin birleştirilmesi ile oluşan hata: 2 farklı durumun hatası ile maske tespit parçasının hataları toplamına eşit olması gerekmektedir. Ancak çok çok düşük oranlarda da olsa sistem bazı görüntüleri birkaç taneyi bazen doğru bazen de yanlış tanıyabilmektedir. Bu yüzden sadece son kodun nihai başarı oranları verilmiştir. Bir önceki başarılar, üst başlıklardaki bulgularda verilmiştir.

Çizelge 4.5. Maskesiz yüzlerin tanıma başarısı

Toplam test edilen yüz sayısı	Tanınan yüz sayısı	Tanıma oranı
450	447	%99,34

Çizelge 4.5’de görüldüğü üzere daha önce sadece maskesiz yüz tanıma kodunun başarısı %99,78 iken maskeli-maskesiz yüz tespit hata oranından dolayı bir miktar düşmüştür.

Çizelge 4.6. Maskeli yüzlerin tanıma başarısı

Toplam test edilen yüz sayısı	Tanınan yüz sayısı	Tanıma oranı
2680	2567	%95,78

Çizelge 4.6’de belirtilen maskeli yüzlerin başarı oranı için iki farklı yöntem denenmiştir sonuçlar aynı çıkmıştır. Bunları birincisi tam yüz eğitim görüntülerin üst parçasının LBP histogramları ile karşılaştırmak oldu. İkincisi ise Yapay maskeli veri kümesinin üst parçasının LBP histogramları ile karşılaştırmak oldu. Başarı oranları literatür çalışmalarına kıyasla iyi sayılabilecek seviyededir. Anacak başarı mevcut veri kümesinin sonucu olduğu için farklı veri kümesinde artma ya da azalma olabilmektedir.

OPENCV’nin “Haar Cascades” yöntemi ile üretilen ve ilk başta maskesiz yüz tanımda kullanılan veri kümesinin tespitteki hata analizi Çizelge 4.7.’da verilmiştir. Bu çizelgede veri kümesinin yatay eksenindeki açısal kısıttan dolayı -35 ile +35 derece arasındaki görüntüler kullanılmıştır. Çizelge 4.8. ve 4.9.’da ise maskeli ve maskesiz yüzleri tanıyan son kodun yüz tespitindeki hata analizini yapacak olursak ele alınması gereken en önemli nokta öncelikle görüntüde yüzü bulup bulmaması daha sonra bulunan yüzün maskeli ve maskesiz olma durumunu doğru tespit etmesidir.

TP: True Positive (Pozitif sınıfı doğru tahmin)

TN: True Negative (Negatif sınıfı doğru tahmin)

FP: False Positive (Pozitif sınıfı yanlış tahmin)

FN: False Negative (Negatif sınıfı yanlış tahmin)

Çizelge 4.7. Yöntem 1, yüz tespit hata matrisi (OPENCV ön yüz)

		Tahmin		Toplam
		Yüz yok	Yüz var	
Yüz Tespiti	Yüz bulunmayan görüntüler	TN=998	FP=2	1000
	Yüz bulunan görüntüler	FN=8	TP=2602	2610
	Toplam	1006	2604	

Bu metodun maskeli yüz tespit başarısı %34 olduğu için bu hata analizi maskesiz yüz içeren görüntülere göre yapılmıştır. Aşağıdaki hesaplamalarda da görüldüğü gibi ilk kodunda -35 ve +35 derece arası tespit doğruluğu yüksek olmasına karşın daya yan duruşlarda bu başarı düştüğü için veri kümesi ve fonksiyon değiştirilmiştir.

$$\text{Doğruluk: } (TP + TN) / \text{TOPLAM} = (998+2602) / (1006+2604) = 0,9972$$

$$\text{Hata Oranı: } (FN + FP) / \text{TOPLAM} = (8+2) / 3610 = 0,0028$$

$$\text{Doğru Pozitif Oranı / Hassasiyet: } TP / \text{Gerçek Varlar} = 2602 / 2610 = 0,9969$$

$$\text{Yanlış Pozitif Oranı: } FP / \text{Gerçek Yoklar} = 2 / 1000 = 0,0020$$

$$\text{Seçicilik / Özgüllük: } TN / \text{Gerçek Yoklar} = 998 / 1000 = 0,9980$$

$$\text{Precision: } TP / (FP + TP) = 2602 / 2604 = 0,9992$$

$$\text{Prevalence: } (FN + TP) / \text{Toplam} = (8+2602) / 3610 = 0,7230$$

Hesaplamalara referans ile şöyle bir hesaplama yapabiliriz. Bu aşamada tanıma başarısı parametre ayarları ile birlikte %99,78 bulunmuştur. Ancak hatalı kısım, yani %0,22 içerisinde yüz olupta tespit kısmının kaçırdığı yüzleri de barındırmaktadır. Dolayısı ile yüz bulunan görüntülerdeki ve tespit edilen görüntülerdeki asıl tanıma oranı için aynı veri kümesinde tespit edilemeyen yüzlerin çıkarılması gerekmektedir. Bu şekilde tespit etkilemediği asıl tanıma başarısı ortaya çıkar.

Çizelge 4.8. Yöntem 2, yüz tespit hata matrisi (Mediapipe)

		Tahmin		Toplam
		Yüz yok	Yüz var	
Yüz Tespiti	Yüz bulunmayan görüntüler	TN1= 999	FP1= 1	1000
	Yüz bulunan görüntüler	FN1= 41	TP1= 3594	3635
	Toplam	1040	3595	

Çizelge 4.9. Yöntem 2, maskeli yüz tespit hata matrisi (Mediapipe)

		Tahmin		Toplam
		Yüzde maske var	Yüzde maske yok	
Maskeli yüz tespit	Maskeli yüzler	TN2= 2962	FP2=2	2964
	Maskesiz yüzler	FN2=4	TP2=628	632
	Toplam	2966	630	

Görüntüdeki yüz tespiti (Görüntüde yüzün bulması pozitif olarak referans alınmıştır)

Doğruluk: $(TP1 + TN1) / TOPLAM = 4593 / 4635 = 0,9909$

Hata Oranı: $(FN1 + FP1) / TOPLAM = 42 / 4635 = 0,0091$

Doğru Pozitif Oranı / Hassasiyet: $TP1 / Gerçek Varlar = 3594 / 3635 = 0,9887$

Yanlış Pozitif Oranı: $FP1 / Gerçek Yoklar = 1 / 1000 = 0,0010$

Seçicilik / Özgüllük: $TN1 / Gerçek Yoklar = 999 / 1000 = 0,9990$

Precision: $TP1 / (FP1 + TP1) = 3594 / 3595 = 0,9997$

Prevalence: $(FN1 + TP1) / Toplam = 3635 / 4635 = 0,7843$

Yüz tespitinden sonraki maskeli-maskesiz yüz tespiti (Yüzde maske olmaması pozitif olarak referans alınmıştır)

Doğruluk: $(TP2 + TN2) / TOPLAM = 3590 / 3596 = 0,9983$

Hata Oranı: $(FN2 + FP2) / TOPLAM = 6 / 3596 = 0,0017$

Doğru Pozitif Oranı / Hassasiyet: $TP2 / Gerçek Varlar = 628 / 632 = 0,9937$

Yanlış Pozitif Oranı: $FP2 / Gerçek Yoklar = 2 / 2964 = 0,0007$

Seçicilik / Özgüllük: $TN2 / Gerçek Yoklar = 2962 / 2964 = 0,9993$

Precision: $TP2 / (FP2 + TP2) = 628 / 630 = 0,9968$

Prevalence: $(FN2 + TP2) / Toplam = 632 / 3596 = 0,1758$

Yine Çizelge 4.8. ve 4.9 için de benzer şekilde önce her adımın hata oranı düşünülerek her parçayı kendisi için hata analizi yapıldı. Maskeli yüz tespiti görüntüde herhangi bir yüz tespitinden sonra geldiği için kendi başarısını ve hata analizini görmek adına yüz tespit yapılan durum referans alınmıştır. Dolayısı ile başlık 4.4'de verilen başarıdan daha yüksektir.

Son bir ara analiz olarak maskeli yüz tanımada yarım yüzleri değil de maskeli bir şekilde yüzün tamamını tanıma algoritmasına sokulduğunda başarı oranı ne olur sorusudur. Bu sonucu vermeden eğitim veri kümesinin yapısının bilinmesinde fayda vardır. Öncelikle eğer eğitim veri kümesi maskesiz yüzlerin eğitimi yapılmış ve test olarak da maskeli bir yüz verilirse başarı son kod için %14,04'dür. Fakat eğitim veri kümesinin tam yüzü yerine maskesiz kısımları karşılaştırdınca başarı %95,78'dir. Eğitim veri kümesi de maskeli bir şekilde eğitilip yine test olarak maskeli tam yüz test için verildiğinde ise tanıma başarısı yine düşüktür ve Çizelge 4.10'de gözüktüğü gibi %11,83'dür. Ancak maskeli yüz tanıma için maskenin kapattığı bölgeleri karşılaştırmak gereksiz performans ve zaman kaybı oluşturmasının yanında tanıma için referans alacağı paydada belirleyici olmayan histogram kısımları değerlerden de anlaşılacağı üzere başarıyı olumsuz etkilemektedir.

Çizelge 4.10. Eğitim seti maskeli tam yüz olduğunda tanıma başarısı

Toplam test edilen yüz sayısı	Tanınan yüz sayısı	Tanıma oranı
186	22	%11,83

Bu tez kapsamındaki başarı oranı ve metodun yine benzer konuda yapılmış çeşitli bilimsel çalışmalarla karşılaştırması Çizelge 4.11'de verilmiştir. Tablo her veri kümesindeki görüntü sayısı ile birlikte kullanılan tekniğe ve veri kümesine dayalı bir karşılaştırmayı da gösterir. Tüm yöntemler maskesiz yüz tanımayı yüksek doğrulukla gerçekleştirir. Ancak, mevcut yöntemlerin çoğu maskelenmiş yüzleri başarılı bir şekilde tanıyamaz. Tez kapsamında önerilen sistem, maskelenmiş yüzleri yüksek doğrulukla başarıyla tanır [51].

Çizelge 4.11. Tez kapsamında yapılan yüz tanıma sırasında elde edilen doğruluk ve çeşitli sistemlerin maskeli yüzleri tanıyıp tanımadığına dayalı olarak mevcut sistemlerle karşılaştırma tablosu [51]

Metot/ Çalışma	Doğruluk (%)	*MYT	Veri Kümesi	Görüntü Sayısı	Teknik
[52]	95	Hayır	Özel veri kümesi	11600	Local Autocorrelations Multiscale Integration
[53]	90	Hayır	Özel veri kümesi	1200	SVM 3D morphable model
[54]	88,9	Hayır	IPES – 1280 Face DB	-	CNN with multiple distance face
[55]	99	Hayır	Özel veri kümesi	-	LBPH
[56]	91.3	Evet	RMFRD	5000 masked 9000 non-masked	VGG-16 pre-trained model
[57]	-	Hayır	ORL, UMIST	400, 575	DF-LDA
[58]	81,49	Hayır	FRGC	12776	Fourier-based LDA
[51]	88,92	Evet	Özel veri kümesi	2000	FaceMaskNet-21
[51]	82,22	Evet	RMFRD	5000	FaceMaskNet-21
[59]	99,99	Evet	LFW	13233	MTArcFace
[3]	72	Evet	-	500	PCA algorithm
TEZ	95,78	Evet	[35], [37]	6660, 1000	LBPH

*MYT: Maskeli Yüz Tanıma

Doğruluk, değeri maskeli yüz tanıma yapılmayan çalışmalarda maskesiz başarıyı göstermekte iken maskeli yüz tanıma olan sistemlerde maskeli tanıma başarısını temsil etmektedir. Tabloya bakıldığında maskeli yüz tanımanın normal maskesiz yüz tanıma oranla daha düşük olduğu görülmektedir. Bu durum maskenin bazı yüz özelliklerini kapattığı için normaldir. Ancak yine de maskesiz tanıma yakın çalışmalarla da karşılaşılmaktadır. Genel olarak çalışmanın başarılarını göz önüne alırsak veri kümesi ve yöntem tanımda en belirleyici iki husustur. Diğer bir husus ise yine yönteme ve veri kümesine ek olarak işlem sırasında görüntüye yapılan onarma ve iyileştirme adımlarıdır. [59]'da ifade edilen başarı oranı çok yüksek olarak göze çarpmaktadır. Burada çalışmada veri kümesinin çok büyük olduğunu görüyoruz. Bu da başarıyı arttırmıştır. Öte yandan [3]'de belirtilen başarının düşük olmasının bir kısmı ise veri kümesi azlığına bağlanabilir.



5. SONUÇ VE ÖNERİLER

Çalışma neticesinin en önemli sonuçlarından bir tanesi maskeli yüz tanımanın maskesiz normal yüz tanımaya göre daha az başarıya sahip olmasıdır. Literatüre bakıldığında aksi bir örnek ile karşılaşılmamıştır. Bu başarının daha düşük olması elbette ki tesadüf değildir. Hangi yöntem ya da hangi algoritma olursa olsun yüz tanıma işlemi yüzün detayları ve özellikleri ile yapılmaktadır. Kaşlar, gözler, burun, çene, dudak gibi yüzde bulunan belirleyici birçok organ ve yüzün genel olarak orantısı büyüklüğü tanımda muazzam kolaylık sağlamaktadır. Durum böyle iken maskeli yüzü düşündüğümüz taktirde, yüze ait çene, ağız, dudak yapısı ve çeşitli yüz büyüklüğü ve oranları kaybolmaktadır. Dolayısı ile tanıma kalan bölge ile yapılmak zorundadır. Tüm bu kaybolan kısım doğaldır ki tanıma için karşılaştırma yapacağı alanı azaltmakta ve tanıma sistemlerini karar vermekte zora düşürmektedir. Öyle ki pandemi sırasında takılan maskelerden dolayı sosyal hayatta bile insanların maskeli olarak birbirlerini tanıyamaması ya da güçlük çekmesi buna örnek verilebilir.

Maske sağlık dışında aynı zamanda güvenlik sistemlerinden kaçmak için art niyetli insanların yüzlerini saklamak için kullandığı bir kalkandır. Dolayısı ile maskeli yüz tanımanın önemi yüksektir. Ayrıca tanıma sistemlerin hem maskeli hem de maskesiz yüzleri tanıyacak sistemleri birleştirerek tek bir sistem kullanması hem güvenlik hem de maliyet anlamında daha etkili olacaktır. Bu birleştirilmiş genel bir yüz tanıma sistemini tüm kontrollü bölgelerde kullanılması tavsiye edilmektedir. Bazı okullarda, hastanelerde, iş yerlerinde, askeriyede ve polis merkezlerinde oldukça fayda sağlayabilecek bu sistem personel takibi ve güvenlik açığı gibi amaçlarla kullanılmaktadır ve kullanılması kritik bir konu olabilmektedir. Aynı zamanda salgın zamanlarında maskeli yüz tanıma kullanılarak, tanıma için maskenin çıkarılmasından kaynaklı temas riskini sıfıra indirecektir.

Çalışma neticesinde edinilen teknik sonuçlara bakılacak olursa özellikle yüz tespitinde -30 derece ile +30 derece arasında tespiti yapılan yüzler kullanılan veri kümesi değiştirilerek her açıda tam tespit sağlanmıştır. İlk kullanılan veri kümesi OPENCV'ye ait haar cascades sınıflandırıcı ile çıkarılmış yüzler iken tam tespit başarısı sağlanan veri kümesi ve fonksiyon Mediapipe kütüphanesinde bulunmaktadır. Bu anlamda Mediapipe benzer çalışma yapacak araştırmacılara önerilir.

Maskeli yüz tespiti yaparken olabildiğince maskesiz kalan kısmın tamamından faydalanmak başarıyı arttıracaktır. Maskeli tanıma yapılabilmesi için öncesinde bu ayrım yapılması gerekmektedir. Bu çalışmada bu ayrım eğitilen maskeli ve maskesiz yüz veri kümeleri ile sağlanmıştır. Burada bu tespit için kullanılan veri kümesinde özellikle maskeli yüzlerde bulunması gereken türdeki maskeli yüze ait örneklerin olması tespitini kolaylaştıracaktır. Ne kadar çeşitli ve farklı yüzler olursa veri kümesinin kalitesi o kadar artmış olur. Yapılan diğer çalışmada bu veri kümesinin sayısına göre kodun maskeli maskesiz tespit başarısı analiz edilmiştir. Bu analize göre eğitim veri kümesindeki görüntü sayısının ve çeşitliliğinin artması, tespit başarısını belli bir yere kadar artırmaktadır. Daha fazla iyileştirme yapmak için yanlış ayırt ettiği maskeli/maskesiz yüzleri doğru kümede tekrar eğitim yapıldı ve başarı maksimuma çıkarılmış oldu. Bu yöntem aslında çeşitli nesne tespit sistemlerinde kullanılmaktadır.

Tanıma ve tespit algoritmalarında önemli olan bir diğer konu da elbette ki sistemin kullanıcıya verdiği tepki, hızı ve performansdır. Veri kümesini artırmak her ne kadar tanımaya olumlu etki etse ve başarısını artırsa da alınan test görüntüsünün karşılaştırma yapacağı fotoğraf sayısı artacağı için bu sistemi yavaşlatacaktır. Bu yüzden sistemin çok yavaşlatmayan ve tanıma başarısının çok düşük olmadığı optimum bir seçim yapılması önerilmektedir. Yine çalışma neticesinde yapılan parametre analizleri göz önüne alınacak olursa yüz tanıma sisteminin çalışacağı ortamda, sistem devreye alınmadan testlerin yapılması ve ortam şartlarının (kameraya uzaklık, parlaklık, açı vb.) da düşünülmesi gerekmektedir.

Çalışma kapsamında sahip olunan görüntülerin ve kapsamının düzgün alınan görüntüler referans alındığı için test edilen ya da canlı görüntülerdeki bozukluklar üzerinde çalışılmamıştır. Ancak bu görüntülerde çalışmak isteyen araştırmacılar için iyileştirme ve onarma teknikleri önerilmektedir. Bu anlamda gerçek bir sistemde çevresel ve bileşenlerden kaynaklı olarak görüntüler gürültülü gelebilmektedir. Bu gürültüleri gidermenin lineer ve lineer olmayan yöntemleri bulunmaktadır. Lineer yöntemlerin uygulanabilirliği ve tasarımı kolay olmasına karşın gürültüyü gidermek adına asıl görüntü yapısını bozabilir. O yüzden gürültülü görüntülerde bu alanda sıkça kullanılan Medyan Filtre önerilmektedir. Görüntü bulanıklığında, parlaklık (aydınlık karanlık) seviyesinin beklenmedik durumlarda olması vb. gibi sorunları düzeltmede oldukça başarılıdır. Görüntü iyileştirmenin çıktısı öznelidir bu yüzden farklı metotlar da denenebilir. Bir diğer işlem ise

görüntü onarmadır ve bu objektiftir. Görüntü onarma için yine oldukça başarılı ve kabul görmüş ters, kör ters katlama ve Wiener filtreleri önerilmektedir. Bunlar uygulanarak görüntü netliği artmaktadır.

Çalışmanın neticesinde görülmüştür ki farklı veri kümelerinde ve farklı yöntemlerde çok farklı başarı oranları oluşmaktadır. Ancak genel bir sonuç oluşmaktadır. Bunlar eğitim veri kümesinin çeşitliliği, görüntülerdeki onarma ve iyileştirme işlemleri, kullanılan yöntem hepsi aynı anda düşünülmelidir. Bu ve benzer çalışmalarla maskeli yüzlerin de yüksek oranda tanınma gerçekleştiği görülmüştür. Çalışmada ayrıca [42-50] arası numaralı kaynaklardan da yararlanılmıştır.





KAYNAKLAR

1. Dharavath, K., Talukdar, F. A. and Laskar, R. H. (2014). Improving Face Recognition Rate with Image Preprocessing. *Indian Journal of Science and Technology*, 7(8), 1170-1175.
2. Wei, B., Jiangjian, X., Chuanhong Z., Minmin, Y. and Chengbin, P. (2017, November). *A Cascade Framework for Masked Face Detection*. International Conference on CIS and RAM, Ningbo, China, 458-462.
3. Ejaz, M. S., Islam, M. R., Sifatullah, M., and Sarker, A. (2019, May). *Implementation of principal component analysis on masked and non-masked face recognition*. 1st international conference on advances in science, engineering and robotics technology (ICASERT), Dhaka, Bangladesh, 1-5.
4. İnternet: OpenCV, (2022, 22 Mayıs). Face Recognition with OpenCV, URL: https://docs.opencv.org/3.4/da/d60/tutorial_face_main.html, Son Erişim Tarihi: 23 Mayıs 2022.
5. İnternet: Python, (2022, 22 Mayıs), General Python FAQ, URL: <https://docs.python.org/3/faq/general.html>, Son Erişim Tarihi: 23 Mayıs 2022.
6. Boyko, N., Basystiuk, O. and Shakhovska, N. (2018). *Performance Evaluation and Comparison of Software for Face Recognition, Based on Dlib and Opencv Library*. Second International Conference on Data Stream Mining and Processing (DSMP), Lviv, Ukraine, 478-482.
7. Kanade, T. (1974). Picture processing system by computer complex and recognition of human faces. *image and video processing*, 16(3), 605-612.
8. Brunelli, R., and Poggio, T. (1992, May). *Face recognition through geometrical features*. In European Conference on Computer Vision. Springer, Berlin, Heidelberg, 792-800.
9. Sarwar, M. G., Dey, A., & Das, A. (2021, April). *Developing a LBPH-based face recognition system for visually impaired people*. 1st International Conference on Artificial Intelligence and Data Analytics (CAIDA), Riyadh, Saudi Arabia, 286-289.
10. Sharma, S., and Jain, S. (2019, March). *A static hand gesture and face recognition system for blind people*. 6th International Conference on Signal Processing and Integrated Networks (SPIN), Noida, India, 534-539.
11. Sridhar Chakravarthy, G., Anupam, K., Harish Varma, P. N. S., Teja, G. H., and Rodda, S. (2019, June). *Face recognition with voice assistance for the visually challenged*. In International Conference on Intelligent Computing and Communication, Springer, Singapore, 701-709.
12. Wang, Z., Wang, G., Huang, B., Xiong, Z., Hong, Q., Wu, H., ... and Liang, J. (2020). Masked face recognition dataset and application. *Image and video processing*, 16(3), 65-68.

13. Hariri, W. (2022). Efficient masked face recognition method during the covid-19 pandemic. *Signal, image and video processing*, 16(3), 605-612.
14. Arya, Z. A., and Tiwari, V. (2021). An Implementation of Real-time Automatic Masked and Unmasked Face Recognition Using LBPH Algorithm. *ADBU Journal of Engineering Technology*, 10(1), 56-60.
15. Du, H., Shi, H., Liu, Y., Zeng, D., and Mei, T. (2021). Towards NIR-VIS masked face recognition. *IEEE Signal Processing Letters*, 28, 768-772.
16. İnternet: WIKIPEDIA, Image, (2022, 8 Mayıs), URL: <https://en.wikipedia.org/wiki/Image>, Son Erişim Tarihi: 23 Mayıs 2022.
17. Kumar, T., and Verma, K. (2010). A Theory Based on Conversion of BGR image to Gray image. *International Journal of Computer Applications*, 7(2), 7-10.
18. Du, H., Kumar T., Verma K., (2010), A Theory Based on Conversion of RGB image to Gray image; *ResearchGate, International Journal of Computer Applications* 7(3), 1140-1493.
19. İnternet: University of Tartu, Digital Image Processing, (2014, Ocak) URL: <https://sisu.ut.ee/imageprocessing/book/1> , Son Erişim Tarihi: 23 Mayıs 2022.
20. İnternet: Wikipedia, Facial recognition system, (2022, 19 Mayıs), URL: https://en.wikipedia.org/wiki/Facial_recognition_system , Son Erişim Tarihi: 23 Mayıs 2022.
21. İnternet: OpenCV, Face Recognition with OpenCV, URL: https://docs.opencv.org/3.4/da/d60/tutorial_face_main.html, Son Erişim Tarihi: 23 Mayıs 2022.
22. İnternet: IEEE, Xplore Digital Library, URL: https://ieeexplore.ieee.org/abstract/document/8698572?casa_token=EaEoFAdCu38AAA:KswR4NJv84FzF_wOOg7URYFFJqiYJScL45oqo_V-gUU75al3GP__3ORRB2c9GTRdRrE8KsAFqU, Son Erişim Tarihi: 23 Mayıs 2022.
23. İnternet: Mare R., (2021, 16 Temmuz), Understanding Facial Recognition Using Local Binary Pattern Histogram (LBPH) Algorithm; *Section*, URL: <https://www.section.io/engineering-education/understanding-facial-recognition-using-local-binary-pattern-histogram-algorithm/>, Son Erişim Tarihi: 23 Mayıs 2022.
24. İnternet: Salton do Prado K., (2017, 11 Kasım), Face Recognition: Understanding LBPH Algorithm; *Towards Data Science*, URL: <https://towardsdatascience.com/face-recognition-how-lbph-works-90ec258c3d6b>, Erişim Tarihi: 23 Mayıs 2022.
25. İnternet: Mittal A. (2020, 21 Aralık), Haar Cascades, Explained; *Analytics Vidhya*, URL: <https://medium.com/analytics-vidhya/haar-cascades-explained-38210e57970d>, Erişim Tarihi: 23 Mayıs 2022.
26. İnternet: Dynamsoft, (24 Mayıs 2019), Image Processing 101 Chapter 1.3: Color Space Conversion, URL: <https://www.dynamsoft.com/blog/insights/image-processing/image-processing-101-color-space-conversion/>, Erişim Tarihi: 23 Mayıs 2022.

27. İnternet: OpenCV, (31 Aralık 2019), Miscellaneous Image Transformations, URL: https://docs.opencv.org/2.4/modules/imgproc/doc/miscellaneous_transformations.html#void%20cvtColor%28InputArray%20src,%20OutputArray%20dst,%20int%20code,%20int%20dstCn%29, Son Erişim Tarihi: 23 Mayıs 2022.
28. İnternet: İşnet, (8 Ekim 2019), Python Programlama Dili Nedir ve Ne İşe Yarar?; URL: <https://www.isnet.net.tr/BlogIcerik/Phyton-Programlama-Nedir-isnetblog>, Son Erişim Tarihi: 23 Mayıs 2022.
29. İnternet: Serengil S. İ., (14 Ocak 2022), Deep Face Detection with Mediapipe, URL: <https://sefiks.com/2022/01/14/deep-face-detection-with-mediapipe/>, Son Erişim Tarihi: 23 Mayıs 2022.
30. İnternet: GeeksforGeeks, (3 Kasım 2021), Python – Facial and hand recognition using MediaPipe Holistic, URL: <https://www.geeksforgeeks.org/python-facial-and-hand-recognition-using-mediapipe-holistic/>, Son Erişim Tarihi: 23 Mayıs 2022.
31. İnternet: musicinformationretrieval.com, NumPy and SciPy, URL: https://musicinformationretrieval.com/numpy_basics.html, Son Erişim Tarihi: 23 Mayıs 2022.
32. İnternet: Anwer T. ve Naeem R. (29 Ekim 2021), Facial Landmark Detection with Mediapipe & Creating Animated Snapchat filters ; *Bleed AI*, URL: <https://bleedai.com/facial-landmark-detection-with-mediapipe-creating-animated-snapchat-filters/>, Son Erişim Tarihi: 23 Mayıs 2022.
33. İnternet: Dynamsoft, (24 Mayıs 2019), Image Processing 101 Chapter 1.3: Color Space Conversion, URL: <https://www.dynamsoft.com/blog/insights/image-processing/image-processing-101-color-space-conversion/>, Son Erişim Tarihi: 23 Mayıs 2022.
34. Sarwar, M. G., Dey, A., and Das, A. (2021, April). *Developing a LBPH-based face recognition system for visually impaired people*. 1st International Conference on Artificial Intelligence and Data Analytics (CAIDA), Riyadh, Saudi Arabia, 286-289.
35. Naik, A. U., and Guinde, N. (2020, August). *LBPH Algorithm for Frontal and Side Profile Face Recognition on GPU*. Third International Conference on Smart Systems and Inventive Technology (ICSSIT), Tirunelveli, India, 776-779.
36. İnternet: Bryan C., (31 Mayıs 2021), MediaPipe in TouchDesigner 5; Magic & Love Interactive, URL: <http://www.magicandlove.com/blog/tag/face-mesh/>, Son Erişim Tarihi: 23 Mayıs 2022.
37. İnternet: Canu S., Facial Landmarks Detection | with Opencv, Mediapipe and Python; *Pysource*, URL: <https://pysource.com/2021/05/14/facial-landmarks-detection-with-opencv-mediapipe-and-python/>, Son Erişim Tarihi: 23 Mayıs 2022.
38. İnternet: Robotics Lab, Databases for Face Detection and Pose Estimation, URL: http://robotics.csie.ncku.edu.tw/Databases/FaceDetect_PoseEstimate.htm#Our_Database_, Son Erişim Tarihi: 23 Mayıs 2022.

39. İnternet: OpenCV, [opencv/data/haarcascades/haarcascade_frontalface_default.xml](https://github.com/opencv/opencv/blob/master/data/haarcascades/haarcascade_frontalface_default.xml), *Github*, URL: https://github.com/opencv/opencv/blob/master/data/haarcascades/haarcascade_frontalface_default.xml, Son Erişim Tarihi: 23 Mayıs 2022.
40. İnternet: Astawa N. G. A., (22 Nisan 2020), KomNET: Face Image Dataset from Various Media; *Mendeley Data*, URL: <https://data.mendeley.com/datasets/hsv83m5zbb/2>, Son Erişim Tarihi: 23 Mayıs 2022.
41. İnternet: Solano G., (29 Ocak 2021), GabySol/OmesTutorials2021; *Github*, URL: <https://github.com/GabySol/OmesTutorials2021>, Son Erişim Tarihi: 23 Mayıs 2022.
42. İnternet: Örnek A.H. (29 Mayıs 2021), Yeni Başlayanlar İçin OPENCV ve PYTHON ile Yüz Tespiti Yapalım; *Youtube*, URL: <https://www.youtube.com/watch?v=f2wAuvJ4594&list=LL&index=34>, Son Erişim Tarihi: 23 Mayıs 2022.
43. İnternet: MakeCad, (4 Nisan 2021), Görüntü İşleme Tabanı ile Yüz Tanıma [OpenCV - Face Recognition], *Youtube*, URL: https://www.youtube.com/watch?v=FnWLpl0_FCY&list=LL&index=27&t=838s, Son Erişim Tarihi: 23 Mayıs 2022.
44. İnternet: OMES, Malla Facial (MediaPipe Face Mesh) | Python - MediaPipe – OpenCV, *Youtube*, URL: <https://www.youtube.com/watch?v=TCUipOXuCBQ&list=LL&index=1&t=524s>, Son Erişim Tarihi: 23 Mayıs 2022.
45. İnternet: Mediapipe, (2020) MediaPipe Face Mesh, URL: https://google.github.io/mediapipe/solutions/face_mesh, Son Erişim Tarihi: 23 Mayıs 2022.
46. İnternet: Mediapipe Team, (6 Mayıs 2022), google/mediapipe; *Github*, URL: <https://github.com/google/mediapipe>, Son Erişim Tarihi: 23 Mayıs 2022.
47. İnternet: Mediapipe Team, (6 Mayıs 2022), mediapipe/mediapipe/python/solutions/; *Github*, URL: <https://github.com/google/mediapipe/tree/master/mediapipe/python/solutions>, Son Erişim Tarihi: 23 Mayıs 2022.
48. İnternet: Machine Learning Hub, Face Filters like #Instagram using OpenCV Python | With Code |, *Youtube*, URL: https://www.youtube.com/watch?v=o0L85hCOo_4&list=LL&index=7, Son Erişim Tarihi: 23 Mayıs 2022.
49. İnternet: Bhavsar K., (14 Nisan 2020), Spidy20/Insta_filters_with_python, *Github*, URL: https://github.com/Spidy20/Insta_filters_with_python, Son Erişim Tarihi: 23 Mayıs 2022.

50. İnternet: OMES, Detecta si una persona lleva mascarilla | OpenCV con Python, Youtube, URL: https://www.youtube.com/watch?v=_SCD62HmJwA&list=LL&index=6&t=3s, Son Erişim Tarihi: 23 Mayıs 2022.
51. Golwalkar, R., and Mehendale, N. (2022). Masked-face recognition using deep metric learning and FaceMaskNet-21. *Applied Intelligence*, 3(2), 1-12.
52. Goudail F, Lange E, Iwamoto T, Kyuma K, Otsu N. (1996). Face recognition system using local autocorrelations and multiscale integration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(10), 1024–1028.
53. Huang J., Heisele B., and Blanz V. (2003). *Component-based face recognition with 3D morphable models*. International Conference on Audio-and Video-Based Biometric Person Authentication (Springer), Washington, DC, USA, 27–34.
54. Moon, H. M., Seo, C. H., and Pan, S. B. (2017). A face recognition system based on convolution neural network using multiple distance face. *Soft Computing*, 21(17), 4995-5002.
55. Bah SM, Ming F. (2020). An improved face recognition algorithm and its application in attendance management system. *Array*, 5(2), 100014.
56. Hariri, W. (2022). Efficient masked face recognition method during the covid-19 pandemic. *Signal, image and video processing*, 16(3), 605-612.
57. Lu, J., Plataniotis, K. N., and Venetsanopoulos, A. N. (2003). Face recognition using LDA-based algorithms. *IEEE Transactions on Neural networks*, 14(1), 195-200.
58. Hwang W, Wang H, Kim H, Kee SC, Kim J (2010). Face recognition system using multiple face model of hybrid Fourier feature under uncontrolled illumination variation. *IEEE transactions on image processing*, 20(4), 1152–1165.
59. Montero, D., Nieto, M., Leskovsky, P., and Aginako, N. (2021). Boosting masked face recognition with multi-task arcface. *Soft Computing*, 21(17), 4995-5002.





GAZİ GELECEKTİR..