T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜZ TANIMA İÇİN LBP VE ORB TABANLI YENİ BİR YÖNTEM GELİŞTİRİLMESİ

WISAM HUSAM QASIM AL-OBAIDI

YÜKSEK LİSANS TEZİ
ELEKTRONİK VE HABERLEŞME MÜHENDİSLİĞİ BÖLÜMÜ
ELEKTRONİK PROGRAMI

DANIŞMAN
YRD. DOÇ.DR. LALE ÖZYILMAZ

İSTANBUL, 2017

T.C.

YILDIZ TEKNİK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

YÜZ TANIMA İÇİN LBP VE ORB TABANLI YENİ BİR YÖNTEM GELİŞTİRİLMESİ

Wisam Husam Qasım AL-OBAIDI tarafından hazırlanan tez çalışması 13.02.2017 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Elektronik ve Haberleşme Mühendisliği Anabilim Dalı'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Tez Danışmanı

Yrd. Doç. Dr. Lale ÖZYILMAZ

Yıldız Teknik Üniversitesi

Jüri Üyeleri

Yrd. Doç. Dr. Lale ÖZYILMAZ

Yıldız Teknik Üniversitesi _____

Yrd. Doç. Dr. Serkan KURT

Yıldız Teknik Üniversitesi _____

Yrd. Doç. Dr. Necdet Tuğrul ARTUĞ

İstanbul Arel Üniversitesi _____

ÖNSÖZ

Bir takım akıllı eylemler yapabilen makinelere olan ihtiyacın giderek artması sonucunda robotik görüşü, son yirmi yıldır araştırma alanında çok önemli bir konu haline gelmiştir. Amaçlardan biri akıllı insan işçiler olmadan tekrarlanan işlemleri gerçekleştirebilmek, diğeri ise elektrik ve kimyasal madde tehlikeleri ile nükleer tehlikelerinin bulunduğu riskli alanlardaki işleri yapmaktır.

Araştırmacılar, mühendisler ve bilim adamları tarafından çok sayıda yayın, teori ve patent geliştirilmiş olmasına rağmen, robot görüşü alanında çok sayıda araştırma konusu halen karmaşıklığını korumakta ve daha fazla gayret gerektirmektedir.

Karmaşık çevre koşulları, kalabalık ortamlar, çok sayıda insan hareketi, aydınlatma düzeyinde farklılıklar ve daha birçok konu araştırma sorunlarının karmaşıklığını artırmaktadır.

İnsan algılama, yüz algılama ve tanıma en karmaşık araştırma başlıklarından biridir. İnsan yüzünün algılanması ve tanınması akrabaların, kardeşlerin, aile bireylerinin ve ikizlerin yüzlerindeki benzerlik ile ırklar ve cilt renkleri arasındaki farklılıklardan etkilenir ve yaşlanmaya, saç şekillerini değiştirmeye, gözlük takmaya, sakal bırakmaya, şapka takmaya, vb. bağlı olarak etkilenebilir.

Yüz algılama ve tanıma algoritmaları, kabul edilebilir bir hata oranıyla tümleşik sistemler kullanılarak uygulanabilir.

Şubat 2017

Wisam Husam Qasım Al-Obaidi

İÇİNDEKİLER

Sayfa

SİMGE LİSTESİ.....vii

KISALTMA LİSTESİ.....viii

ŞEKİL LİSTESİ.....ix

ÇIZELGE LİSTESİ.....xi

ÖZET.....xii

BÖLÜM 1

GİRİŞ..... 1

1.1 Literatür Özeti..... 1
1.1.1 Rijit Şablonlu Yüz Algılama Algoritmaları..... 2
1.1.2 Sinir Ağlarının Kullanımıyla Rijit Şablonlu Yüz Algılama..... 3
1.1.3 Rijit Yüz Algılama Için Diğer Öğrenme Düzenleri..... 4
1.2 Tezin Amacı..... 5
1.3 Hipotez..... 5

BÖLÜM 2

GENLİK DAĞLIMI OLASILIĞI LBP (PADL) ALGORİTMASININ FARKLILIK SEVİYELERİ.....6

2.1 LBP Algoritması..... 6
2.1.1 LBP Konsepti..... 6
2.2 LBP Genişletme ve Düzenleme..... 8
2.3 K-ortalama Kümeleme..... 9
2.3.1 K-ortalama Kümeleme Ayarları..... 9
2.3.2 Teorik Hesaplama..... 9
2.3.3 K-ortalama Kümeleme Prosedürü..... 10
2.4 Bayes Teoremi..... 11

BÖLÜM 3

YENİ YÜZ ALGILAMA YÖNTEMİ....13

3.1 Prosedür 15
3.2 Yüz Ayrıştırması Için K-ortalama Kümeleme 15
3.3 LBP Öznitelik Vektörü Farklılıklarının Hesaplanması 17
3.4 LBP Öznitelik Vektörü 18
3.5 Görüntü Sınıflandırması 19
3.6 Deney Sonuçları 24
3.6.1 K-ortalama Uygulaması 24
3.6.2 Kutuları Kaydırma 25
3.7 Sonuç 29
3.8 Sistem Zayıflığı 30

BÖLÜM 4

YÜZ TANIMA....31

4.1 Yüz Tanıma 31
4.1.1 Eigenfaces ve Fisherfaces 33
4.1.2 Lokal İkilik Değer Düzenleri Histogramı 34
4.2 Önerilen Yüz Tanıma (PADL Temelinde) 34
4.3 Yüz Tanıma (ORB (Yönelimli FAST ve Döndürülmüş BRIEF) Temelinde) 36
4.3.1 (Fast) Köşe Algılama 36
4.3.2 BRIEF (İkili Sağlam Bağımsız Temel Öznitelikler) 38
4.3.3 ORB (Yönelimli BRIEF) 40
4.4 ORB metodu 40
4.4.1 R-BRIEF: (Döndürmeye Duyarlı Brief) 41
4.4.2 Yönlendirilmiş BRIEF 41

BÖLÜM 5

SONUÇLAR....44

5.1 ORB için Yüz Tanıma Sonuçları....44
5.2 Sonuç....49
KAYNAKLAR....51
ÖZGEÇMİŞ....53

SİMGE LİSTESİ

d (xi, x j)	İki Nokta Arasındaki Mesafe
m_k	Kth Kümesinin Ortalama Vektörü
C(*i*)	Ith Gözlemi Için Küme Numarası
W(C)	Küme Dağılımı Içinde
B(C)	Küme Dağılımı Arasında
p(x)	Bir Düzleştirilmiş Yama p Içerisindeki Piksel Yoğunluğu
f_{nd} (P)	BRIEF Tanımlayıcı
nd (x, y)	Konum Çiftleri Benzersiz Olarak Bir Ikilik Değer Testleri Setini Tanımlar
$M_{p\,q}$	Bir Yamanın Momentleri
C	Ağırlık Merkezleri
R_Φ	Döndürme Matrisi
gn	Yönlendirilmiş BRIEF Işleci
Φ	Yamanın Yönelimi

KISALTMA LİSTESİ

BRIEF	İkili Sağlam Bağımsız Temel Öznitelikler
CNN	Kıvrımlı Sinir Ağı
FAST	Hızlandırılmış Segment Testinden Nesneler
HOG	Yönelimli Eğimlerin Histogramı
LBP	Lokal İkilik Değer Düzenleri
ML	En çok Olabilirlik
MLP	Çok Katmanlı Algılayıcı
OPENCV	Açık Kaynak Kodlu Bilgisayarla Görme Kütüphanesi
ORB	Yönelimli FAST ve Döndürülmüş BRIEF
PADL	LBP Farkı Genlik Dağılımı Seviyeleri Olasılığı
SIFT	Ölçekten Bağımsız Öznitelik Dönüşümü
SURF	Hızlandırılmış Gürbüz Öznitelikler
SVM	Destek Vektör Makineleri

ŞEKİL LİSTESİ

Sayfa

Şekil 2.1 Orta piksel ile piksel komşuları arasındaki ilişki 7
Şekil 2.2 LBP hesaplama prosedürü (örnek, fark, eşik değer) 7
Şekil 2.3 16x16 bloklar ile görüntü temsili ve bunun histogramı ve son öznitelik vektörü 8
Şekil 2.4 Başlangıçtaki ağırlık merkezleri, son tekrarlama sonunda ağırlık merkezleri 11
Şekil 3.1 Önerilen prosedürün akış şeması 14
Şekil 3.2 Matlab kullanılarak K-ortalama kümeleme uyarınca ayrıştırılan görüntüler (soldan sağa orijinal görüntü, ayrıştırma 1, ayrıştırma 2, ayrıştırma 3) 16
Şekil 3.3 Orijinal yüz, LBP algoritması uygulandıktan sonraki yüz 17
Şekil 3.4 LBP uygulamasında sınır piksellerinin etkisi 17
Şekil 3.5 Test edilen görüntü 169*177, test edilen görüntü için öznitelik vektörü 18
Şekil 3.6 Yüzler ve sahneler görüntüsü için LBP 19
Şekil 3.7 Yüz5 ve yüz7 arasındaki fark 20
Şekil 3.8 Yüz-yüz ve yüz-sahne farkları 23
Şekil 3.9 Test edilen örnek yüz 24
Şekil 3.10 3 Bölgeye ayrılmış görüntüde renkler ayrıştırılarak renk aralıklarına göre ayrıştırılmış görüntüler 24
Şekil 3.11 Renk bölgelerinin (5) bölgede olacağı seçildiğinde görüntü için beş ayrıştırma 25
Şekil 3.12 Test edilen bir görüntünün engeli için LBP'ye benzerliğini bulmak üzere algılama prosesi LBP öznitelik vektörü kullanılarak yapılan örnek yüz görüntüsü 26
Şekil 3.13 Değiştirilmiş (Aberdeen küçük veri seti) veri setinden algılanan yüzlerin örneği 28
Şekil 3.14 Önerilen algoritma kullanılarak tam karşıya bakmayan yüzlerin algılanması örneği 29
Şekil 3.15 Yüzleri bulma doğruluğu - dahil edilen sıfırların sayısı 30
Şekil 3.16 Örnek görüntü, küme 1, küme 2, küme 3, sonuçta oluşan görüntü 30
Şekil 4.1 Eğitim seti 33
Şekil 4.2 Şekil 4.1'de gösterilen görüntünün Eigen vektörleri 33
Şekil 4.3 Şekil 4.1 veri setinin LBP açıklaması 34

Şekil 4.4 İlgi noktasını algılamak için FAST köşe algılama 37
Şekil 4.5 FAST köşe algılama 38
Şekil 4.6 BRIEF örnekleme düzeni 39
Şekil 4.7 ORB öznitelik eşleştirme 43
Şekil 5.1 Ölçeğin noktaların doğruluğu üzerindeki etkisi 45
Şekil 5.2 Ölçek değişimi - hata yüzdesi 46
Şekil 5.3 Döndürmenin öznitelik noktaları üzerindeki etkileri 46
Şekil 5.4 Döndürmenin hata yüzdesi üzerindeki etkileri 47
Şekil 5.5 Aydınlatmanın öznitelik noktaları üzerindeki etkileri 48
Şekil 5.6 Aydınlatma - hata yüzdesi 48

ÇİZELGE LİSTESİ

Sayfa

Çizelge 2.1 Örnek bilgisi 12
Çizelge 3.1 LBP'nin farklılıklarının dağılım seviyelerinde bulunan noktalar 20
Çizelge 3.2 Yüzler - Yüzlerin değerleri ile Yüzler - Sahnelerin değerleri arasındaki farklılıklar 21
Çizelge 2.3 Görüntü karşılaştırması için sıfırların sayısının sıfır olmayanlara piksel oranı 21
Çizelge 3.1 Yüzün kendisi ile diğer yüzler ve başka sahneler arasındaki farklar 35
Çizelge 5.1 Öznitelik noktalarının sayısı ve hesaplanması için gereken süre 44
Çizelge 5.2 Ölçek değişimi - hata yüzdesi 45
Çizelge 5.3 En güçlü 10 nokta için ORB sonuçları (döndürme etkileri) örnekleri (adrian1 - yüz1) 46
Çizelge 5.4 En güçlü 10 nokta için ORB sonuçları (aydınlatma etkileri) 47
Çizelge 5.5 En güçlü 10 nokta eşik etkisi için ORB sonuçları örnekleri (adrian1 - diğerleri) 49

ÖZET

YÜZ TANIMA İÇİN LBP VE ORB TABANLI YENİ BİR YÖNTEM GELİŞTİRİLMESİ

Wisam Husam Qasım AL-OBAIDI

Elektronik ve Haberleşme Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

Tez Danışmanı: Yrd. Doç. Dr. Lale ÖZYILMAZ

Yüz algılama ve tanıma, robot görüşü alanında en zorlu araştırma konularından biridir. Çevre ve aydınlatma koşullarının değişmesiyle ilişkili birçok sorunu çözmek için çok sayıda algoritma önerilmektedir. Bu araştırmada ise otonom biçimde yüz algılama ve tanıma için yeni bir algoritma önerilmektedir. Önerilen yöntem, iyi bilinen LBP (Local binary patterns) algoritmasını ve yüz segmentasyonu için K-ortalama kümeleme yöntemini ve çıkış verisini sınıflandırmak için maksimum benzerliği kullanmaktadır.

Bu yöntem, özellik vektörü genliklerinin üçü pozitif ve üçü negatif olmak üzere altı seviyede dağılımına bağlı olarak yüzlerin algılanması ve tanınması işlemi olarak özetlenebilir. Algılama ve tanıma, bu dağılım değerlerinin sınıflandırılması ve bunun bir yüz olup olmadığına ve yüzün kime ait olduğuna karar verilmesi ile yapılabilir.

Bu işlem başlatılmadan önce, yüzün arka planından ayrılması için bir ön işlem yapılmalıdır ve bu işlem K-ortalama kümeleme yöntemi kullanılarak yapılır.

Yüz algılama ve tanıma konusunda başarısının, kabul edilebilir bir algılama oranına sahip olduğunu test etmek üzere belirli bir veri setinde algoritma için çok sayıda test yapılmıştır.

Anahtar Kelimeler: Yüz algılama, K-ortalama, ORB tabanlı , LBP tabanlı, yüz tanıma

ABSTRACT

FACE RECOGNITION BASED ON NEW MODIFIED METHOD OF LBP AND ORB ALGORITHMS

Wisam Husam Qasım ALOBAIDI

Department of Electronics and Communications Engineering

MSc. Thesis

Adviser: Asst. Prof. Dr. Lale ÖZYILMAZ

The face detection and recognition is one of the most challenging research topics in the field of robotic vision, there is a large number of algorithms proposed in order to solve many problems related to changing of environment and lighting conditions, in our proposed research we introduced a new algorithm for face detection and recognition autonomously, the proposed method uses the well-known LBP (Local binary patterns) algorithm and K-mean clustering for face segmentation and maximum likelihood to classify the output data.

the method can be summarized as a process of detection and recognition faces depending on the distribution of feature vector amplitudes on six levels, three for positive and three for negative amplitudes, the detection and recognition can be made by classifying these distribution values and make decision if it is a face or not and to whom that face belongs to.

Before this process is started a preprocessing step should be made to separate the face from its background and this is made by using the k-mean clustering.

Many of tests have been made for the algorithm to a certain dataset to test its ability to detect and recognize face in acceptable rate of detection.

Keywords: Face detection, K-mean clustering, ORB algorithm, LBP algorithm, face recognition

BÖLÜM 1

GİRİŞ

1.1 Literatür Özeti

Sadece büyüleyici yapısı nedeniyle değil, aynı zamanda ilgi çekici bir uygulama alanı olduğu ve ihtiyaç duyulduğu için yüz algılama, bilgisayarla görüntüleme ve biyometrik mühendisliği alanlarında üzerinde çok çalışılan bir başlık olmuştur. Geçtiğimiz 15 yıl içinde çok sayıda algoritma sunulmuştur. Bu algoritmalar iyi bilinen Viola -Jones algoritmasından [1] sonra sunulmuş olup bu yüz algılama algoritmaları; esas yöntemlerin güçlendirilmesiyle öğrenilen rijit şablonlar ya da derin sinir sistemi veya yüzü parçaları ile açıklayan biçimi değiştirilebilir modeller olmak üzere başlıca iki düzende kategorize edilebilir. Bu tezde başarılı birkaç yöntem üzerinde durulacaktır. Bu tanıtımda yüz algılama için kullanılan ana veri tabanından da bahsedilecektir.

Yüz algılamanın önemi, örneğin yüz algılama ve doğrulama, yüz izleme, cinsiyet belirleme, yaş tanıma, yüz ifadesi tanıma ve başka birçok uygulama gibi yüzün özelliklerine bağlı çok sayıda uygulama için kilometre taşı olmasından ileri gelir.

En önemli yüz algılama algoritması, yüz algılama teknolojisinin gerçek yaşamda uygulanabilmesine imkan tanıyan Viola ve Jones algoritması olup halen çok sayıda gerçek dünya sisteminde de kullanılmaktadır. Bu alanda çok sayıda başarı elde edilmiştir. Modern yüz algılama algoritmalarında aşağıdakilerden de faydalanılmıştır:

- Scale Invariant Feature Transform (SIFT) özellikleri [3], Local Binary Patterns (LBP'ler) [2] ve bunların varyasyonları gibi özellik çıkartım metodolojilerinin ve Speeded Up Robust Features (SURF) [4], Histograms of oriented Gradients (HoGs)

[5] gibi benzeştirme parçaları gibi özelliklerin sunulmasıyla, bu özellik çıkartım yöntemleri yüzlerin tanımlanması ve algılanması için kullanılmıştır.

- Bu yeni veri setleri LFW, PASCAL, FDDB, vb. gibi araştırmacılar tarafından sunulmuştur.
- Çok sayıda sınıflandırma ve Support Vector Machines (SVM) [6], Boosting ve (derin) sinir sistemi yapısı gibi kapsamlı yöntemler oluşturulmuştur.
- Herkes tarafından ulaşılabilecek yüksek kaliteli kaynaklar bulunmaktadır, bunlar arasında OPENCV ve piyasaya sürülmesi ve aynı zamanda yakın zamanda yaşanan gelişmeler ile sinir sistemi yapısında yapılan araştırmalar bulunur.

Yüz algılama algoritmaları, aşağıdaki gibi dört grupta sınıflandırılabilir:

- Özelliği değişmez yöntemler.
- Bilgi tabanlı yöntemler.
- Şablon eşleme yöntemleri.
- Görünüm tabanlı yöntemler.

Özelliği değişmez yöntem, farklı duruşlarda ve aydınlatma koşullarında yüz algılayabilmeyi amaçlamaktadır.

Bilgi tabanlı yöntem, yüzlerin algılanması için insan kuralları ve insan bilgisi esaslıdır.

Şablon eşleme yöntemleri, bir görüntüde yüzün konumunu algılamak için önceden kurulumu yapılmış yüz şablonunu görüntü ile karşılaştırır.

Görünüm tabanlı yöntemler, yüz algılamak için yüz modellerini eğitim amaçlı kullanılan yüz setlerinden öğrenir.

1.1.1 Rijit Şablonlu Yüz Algılama Algoritmaları

Viola-Jones Yüz Detektörü: bu iyi bilinen algoritma, gerçek zamanlı yüz algılaması yapabilmek için üç önemli karakteristik özelliğe sahiptir ve bunlar aşağıda verilmiştir:

- Tümleşik görüntü.
- AdaBoost kullanarak sınıflandırıcı öğrenme.

- Dikkat ile ilişkili kademeli yapı.

Tümleşik görüntü: Bir ızgaranın dörtgen bir alt setinde değerlerin toplanması için hesaplamada kullanılan bir yöntemdir. Viola-Jones yüz detektörü, Haar-benzeri özelliklerin hızlıca hesaplanması için tümleşik görüntüyü uygulamıştır.

Boosting: Her biri orta düzey doğruluk seviyesine sahip çok sayıda zayıf hipotezi bir araya getirerek güçlü ve doğru bir hipotez bulma yöntemidir.

AdaBoost algoritmasının genelleştirilmiş bir yöntemi RealBoost adıyla sunulmuştur.

Dikkat ile ilişkili kademeli yapı: Dikkat ile ilişkili kademeli yapı Viola-Jones detektörün ana fikridir. Kilit nokta daha küçük ve bu nedenle daha verimli güçlendirilmiş sınıflandırıcıların oluşturulabilmesidir; bunlar negatif alt pencerelerin büyük bir çoğunluğunu reddederken neredeyse tüm pozitif örnekleri tutmaktadır.

Detektörün erken kademelerinde alt pencerelerin büyük bir çoğunluğu reddedilecektir; böylece algılama işlemi son derece verimli olacaktır.

1.1.2 Sinir Sistemi Kullanılarak Rijit Şablonlu Yüz Algılama

Sinir sistemi, yüz algılamada kullanılan yeni ve popüler bir yaklaşımdır. Sinir sistemi kullanılarak yüz algılama için yapılan ilk çalışmalar [8] tarafından [10]'da yapılmıştır. [8], sinir sistemi kullanarak yüz algılama için yöntemlerden birini üretmiştir. Bu yöntemde kısıtlı üretken model (CGM - Constrained Generative Model) adı verilen bir model kullanılmıştır. CGM yüzler için üretken bir model oluşturmak üzere eğitilmiş üç katmandan oluşan çoklu bir katman algılamasıdır (MLP - Multi Layer Perception). Çoklu görünüm yüz algılama, çok sayıda CGM'nin hatalarının yeniden yapılandırılmasıyla ölçülür. Koşullandırılmış karışım ve bir MLP geçit sistemi ile birleştirilir.

Başka bir yaklaşım [11]'de sunulmuştur; yazarlar evrimli bir sinir sistemi yapısı temelinde bir yüz algılama düzeni önermiştir. Alışılagelmiş özellik tabanlı yaklaşımlarla karşılaştırıldığında, evrimli sinir sistemi, çıkarım için özellikler hakkında varsayımlar yapmadan veya analiz etmek için yüzün alanlarını araştırmadan, eğitim örneklerinden otomatik olarak özellik çıkarımlarını türetmektedir.

İşin gerçeği, sinir sistemi kullanılarak yüz algılama için çok fazla sayıda yaklaşım bulunmaktadır.

1.1.3 Rijit Yüz Algılama Için Diğer Öğrenme Düzenleri

Viola ile Jones tarafından yapılan çalışmanın haricinde, iyi ve rekabetçi sonuçlar sağlayan bir dizi şablon kullanılarak yüz algılama sorununu çözümlemek için birkaç farklı araştırma girişimi de yapılmıştır.

Anti-faces olarak adlandırılan yöntemlerin önerildiği başka bir yaklaşım [13]'te sunulmuştur. Anti-faces, resimlerdeki yüzler dahil olmak üzere, nesnelerin algılanması için çok sayıda şablon kullanılan bir düzendir. Temel fikir, [1]'de açıklanan kademeli yönteme çok benzemektedir; yüzleri algılamak için bir dizi sıralı sınıflandırıcı kullanır ve yüz olmayanları hızlı bir şekilde reddeder. [13]'te bulunan her sınıflandırıcı (detektör), kısıtlandırılmış optimizasyon aracılığıyla elde edilen bir şablon görüntüdür. Örnek görüntülere sahip şablonun iç ürünleri en aza indirilir ve sonraki şablonlar öncekilerden bağımsızdır. Bu yaklaşımda, negatif görüntü Boltzmann dağılımı ile modellenmiştir ve düzgün olduğu varsayılır. Böylece şablon oluşturma sırasında bir şey gerekmez. Başka bir yaklaşım, [14]'te önden yüz algılamasının önerildiği bir Bayes ayrıştırma özellikleri yönteminde sunulmuştur. Yüz sınıfı, çok değişkenli normal dağılım olarak modellenmiştir. Yüz sınıfına en yakın olan yüz olmayan görüntülerin bir alt seti, yüz sınıfı modeli tabanında seçilmiştir ve yine çok değişkenli normal bir dağılım ile modellenmiştir. Yüz olup olmadığı yönünde son karar bir Bayes sınıflandırıcı tarafından alınmıştır. Yüz sınıfına en yakın yüz olmayan görüntüler modellendiği için, yüz olmayan görüntülerin büyük bir çoğunluğu sınıflandırma sırasında yok sayılmıştır. Bu yaklaşım, sınıflandırma sınırını tanımlamak için eğitim örneklerinin (destek vektörleri) sadece bir alt setinin kullanıldığı destek vektörü makinelerine (SMV'ler - Support Vector Machines [15]) ilham kaynağı olmuştur.

1.2 Tezin Amacı

Bu araştırmada ise otonom biçimde yüz algılama ve tanıma için yeni bir algoritma önerilmektedir. Önerilen yöntem, iyi bilinen LBP (Local binary patterns) algoritmasını ve yüz segmentasyonu için K-ortalama kümeleme yöntemini ve çıkış verisini sınıflandırmak için maksimum benzerliği kullanmaktadır.

1.3 Hipotez

Bu algoritmanın ana fikri, LBP histogramı ile arasındaki benzerlik temelinde bir yüz algılamaktır. Karşılaştırma esasen özellik vektörü temelinde değil, noktaların histogram genliği dağılımı sayıları temelinde olacak ve bu prosedür daha sonra detaylı olarak açıklanacaktır.

BÖLÜM 2

GENLİK DAĞLIMI OLASILIĞI

LBP (PADL) ALGORİTMASININ FARKLILIK SEVİYELERİ

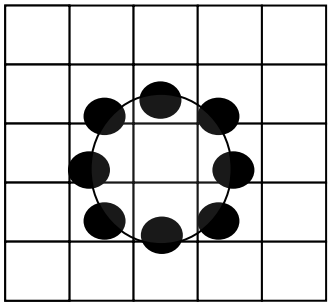
2.1 LBP Algoritması

Lokal ikili değer düzenleri sınıflandırmada kullanılan bir açıklama yöntemi türüdür ve robot görüşü alanında çok sayıda uygulaması bulunmaktadır. İlk olarak [2]'te açıklanmıştır ve başlıca uygulama alanı doku sınıflandırılması alanındadır. Bazı yayınlarda, bazı veri setleri [16] için performans algılamasını iyileştiren HOG ile birlikte kullanılmaktadır. Ayrıca özellikle arka planı çıkarma [17] alanında LBP için önemli iyileştirmeler uygulanmıştır. LBP'nin en önemli avantajı basitliği ve çok verimli doku işlemcisi olmasıdır.

2.1.1 LBP Konsepti

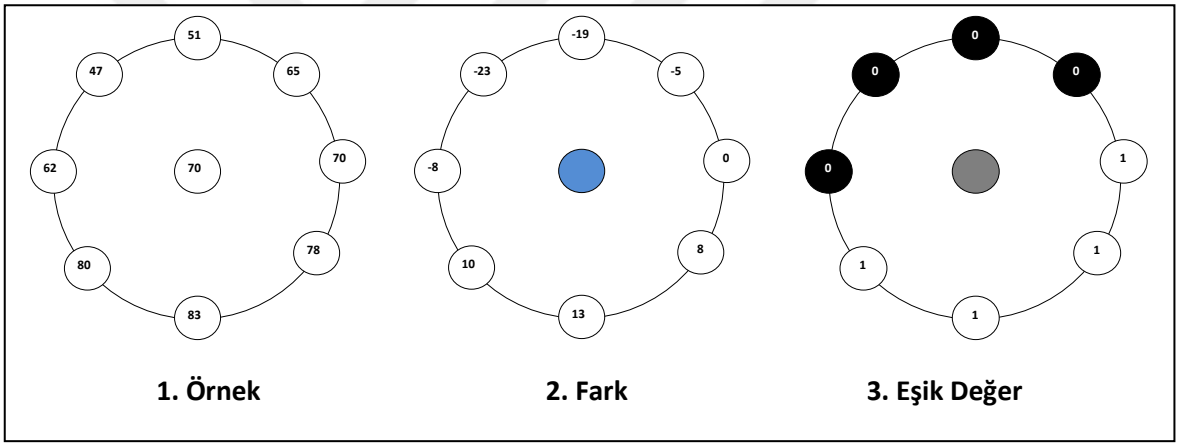
LBP, görüntü bilgisinin eşsiz bir ikili değer düzeni oluşturacak şekilde geri döndürüldüğü bir işlemdir; LBP özellik vektörü prosedürü aşağıdaki adımlarda görülebilir:

1. Görüntü 16x16 bloklara bölünür ve prosedür her bir bloğa ayrı ayrı uygulanır.
2. Her piksel, aşağıda bulunan Şekil 2.1'de gösterildiği gibi, sol üst köşeden başlayarak saat yönünde veya saat yönünün tersine 8 komşusu ile karşılaştırılır.



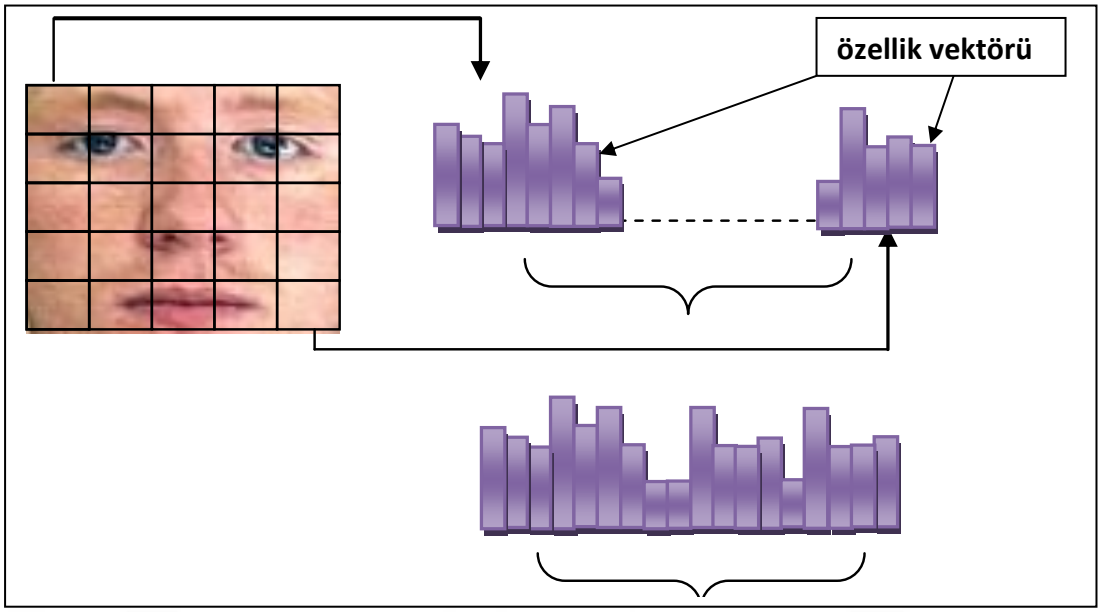
Şekil 2.1 Orta piksel ile piksel komşuları arasındaki ilişki

3. Orta noktalar ile çevresindeki pikseller arasındaki farklılıklar için kıyaslama yapılır. Komşu pikselin değeri orta pikselin değerinden daha düşükse "0", aksi takdirde "1" yazılır. Sonuç 8-bit ikili değer bir sayı olacaktır. Daha sonra bu sayının, aşağıda bulunan Şekil 2.2'de gösterildiği gibi bir ondalık sayıya dönüştürülmesi gerekir.



Şekil 2.2 LBP hesaplama prosedürü (örnek, fark, eşik değer)

4. Her bir 16x16 blok için histogram hesaplaması ve sonuç 256 özellikli bir vektör olacaktır.
5. Normalleştirilmiş histogram (opsiyonel).
6. Tüm blokların peşi peşine birleştirilmesi durumunda, görüntünün tamamı, aşağıda bulunan Şekil 2.3'te gösterildiği gibi, eşsiz bir ikili değer açıklaması sağlayacaktır.



Şekil 2.3 16x16 bloklar ile görüntü temsili ve bunun histogramı ve son özellik vektörü

7. Bilgileri sınıflandırmak için, bir makine öğrenme algoritması (destek vektör makinesi, adaboost, vb.) kullanılabilir.

2.2 LBP Genişletme Ve Düzenleme

Doku tanımlayıcı [16] gibi bazı uygulamaların performansını artırmak için LBP'ye çok sayıda düzenleme yapılmıştır. Bu düzenlemelerin büyük bir kısmı komşu boyutlarının değiştirilmesi ile yapılır. Başka bir genişletme, özellik vektörünün uzunluğunu düşüren ve basit dönüşlü değişmez tanımlayıcı sağlayan, yeknesak düzenler olarak adlandırılan başka bir faktör eklenerek yapılır.

Tek tip düzen 0'dan 1'e veya 1'den 0'a geçişi en fazla 2'ye eşit olacak şekilde tanımlanır ve bit çapında geçiş olarak adlandırılır. Örneğin bir pikselin düzeni 00000000 ise bit çapında geçiş 2'den düşük olduğu için tek tip olarak adlandırılır ya da pikselin komşu düzeni 01111110 ise bit çapında geçiş iki kez olduğu için yine tek tip olarak adlandırılır, rakam 01010111 ise bit çapında geçiş 5 kez olduğu için tek tip olarak adlandırılmaz ve bu şekilde devam eder.

2.3 K-ortalama Kümeleme Yöntemi

Kümeleme için kullanılan en popüler algoritmalardan biridir. Kümeleme bilgileri gruplara ayırma işlemi olarak tanımlanır. Her bir grup bilgisi birbirine daha çok benzer ve başka grup bilgisine daha az benzer. K-ortalama kümeleme yöntemi ilk olarak [21] tarafından sunulmuştur. K-ortalama kümeleme yöntemi, gözetimsiz kümeleme algoritması olarak değerlendirilir. (K) küme sayısını temsil eder. (K) değerini otomatik olarak tahmin etmek için kullanılabilecek algoritmaların toplamı olarak kullanıcı tarafından girilen bir değerdir. Sadece sayısal veriler için çalışır ve uygulanması biraz daha kolaydır.

2.3.1 K-ortalama Kümeleme Ayarları

- (x1, x2……..xn) kümelenecek veri noktaları veya vektörlerdir.
- Her bir gözlemin sadece bir kümeye atanması gereklidir.
- Ci, gözlemin C kümesi anlamına gelmektedir.
- Öklit mesafesi kullanılarak benzemezlik ölçülür.
- K-ortalaması küme noktası dağılımı içinde en aza indirilir.

2.3.2 Teorik Hesaplama

$$T = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} d(x_i, x_j) \tag{1.1}$$

Burada d (xi, xj) iki nokta arasındaki mesafeyi belirtmektedir

T aşağıdaki şekilde yeniden yazılabilir:

$$T = \frac{1}{2}\sum_{k=1}^{K}\sum_{C(i)=k}\Big(\sum_{C(j)=k} d(x_i, x_j) + \sum_{C(j)\neq k} d(x_i, x_j)\Big)$$
$$= W(C) + B(C) \tag{2.2}$$

Burada

$$W(C) = \frac{1}{2}\sum_{k=1}^{K}\sum_{C(i)=k}\sum_{C(j)=k} d(x_i, x_j) \tag{2.2}$$

Burada W(C) kümeleme dağılımı içindedir

$$B(C) = \frac{1}{2}\sum_{k=1}^{K}\sum_{C(i)=k}\sum_{C(j)\neq k} d(x_i, x_j) \tag{2.4}$$

Burada B(C) kümeleme dağılımı arasındadır

Eğer (d) kare Öklit mesafesiyse, bu durumda:

$$W(C) = \sum_{k=1}^{K} N_k \sum_{C(i)=k} \|x_i - m_k\|^2 \tag{2.5}$$

Ve

$$B(C) = \sum_{k=1}^{K} N_k \|m_k - m\|^2 \tag{2.6}$$

Burada (m) büyük ortalama anlamına gelmektedir

W(*C*)'nin küçültülmesi *B*(*C*)'nin büyütülmesi ile eşdeğerdir

2.3.3 K-ortalama Kümeleme Prosedürü

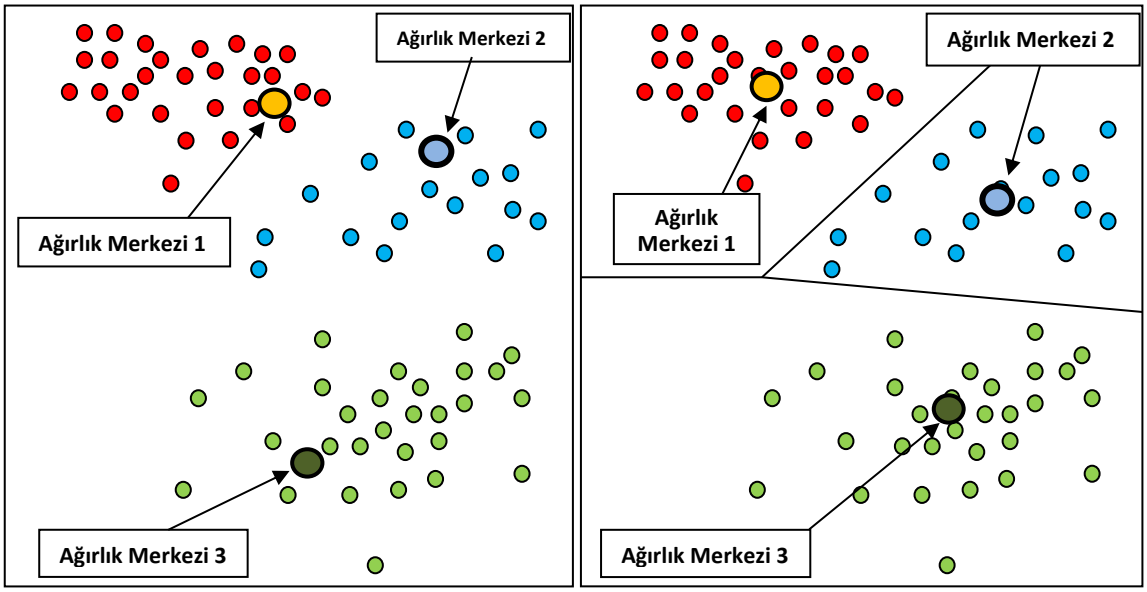
- Belirli bir kümeleme için veri noktalarının (*C*) atanmasıyla kümeleme ortalaması hesaplanır:

$$m_k = \frac{\sum_{i:C(i)=k} x_i}{N_k},\ k = 1, \ldots, K. \tag{2.7}$$

- Mevcut kümeleme ortalaması için, her bir gözlem aşağıdaki ilişkiye göre atandığında:

$$C(i) = \arg\min_{1\leq k\leq K} \|x_i - m_k\|^2,\ i = 1, \ldots, N \tag{2.8}$$

- Yukarıdaki iki adım yakınsama oluşuncaya kadar tekrarlanır



(a) (b)

Şekil 2.4 (a) Başlangıçtaki ağırlık merkezleri, (b) Son tekrarlama sonunda ağırlık merkezleri

2.4 Bayes Teoremi

$$P(h|D) = P(D|h) * P(h) / P(D) \quad (2.9)$$

- **P(h)=** hipotezin öncelikli olasılığı h
- **P(D)=** eğitim verisinin öncelikli olasılığı D
- **P(h|D)=** verilen h olasılığı D
- **P(D|h)=** verilen h için D olasılığı

Eğitim verileri için en uygun hipotezi bulunur.

Son gelen hipotezi maksimum değeri h_{MAP}:

$$h_{MAP} = \arg\max_{h \in H} P(h|D)$$

$$= \arg\max_{h \in H} P(D|h) * P(h) / P(D) \quad (2.10)$$

$$= \arg\max_{h \in H} P(D|h) * P(h)$$

$P(h_i) = P(h_j)$ varsayımıyla daha da sadeleştirilir ve maksimum benzeşme (ML) hipotezi seçilir

h ML= arg maks hiєH P(D|hi) (2.11)

Örnek:

Çizelge 2.1 örnek bilgisi

NO	Yaş	giriş	öğrenci	Kredi Notu	Bilgisayar satın alır
1	35	Ortalama	Evet	Çok İyi	Evet
2	30	Yüksek	Hayır	Orta	Hayır
3	40	Düşük	Evet	İyi	Hayır
4	35	Ortalama	Hayır	Çok İyi	Evet
5	45	Düşük	Hayır	Çok İyi	Evet
6	35	Yüksek	Hayır	Mükemmel	Evet
7	35	Ortalama	Hayır	İyi	Hayır
8	25	Düşük	Hayır	İyi	Hayır
9	28	Yüksek	Hayır	Orta	Hayır
10	35	Ortalama	Evet	Orta	Evet

P (hi) = P (hj) olduğu varsayımı ile hesaplanan olasılıklar aynı miktara ulaşır

Biraz daha sadeleştirme ile:

hML = arg maks P(D/hi) (hi değeri H değerine aittir)

P (bilgisayar satın alır = evet) = 5/10 = 0.5

P (bilgisayar satın alır = hayır) = 5/10 = 0.5

P (müşteri 35 yaşındadır ve 50,000 ABD Doları kazanmaktadır) = 4/10 = 0.4

P (müşteri 35 yaşındadır ve 50,000 ABD Doları kazanmaktadır / bilgisayar satın alır = evet) = 3/5 =0.6

P (müşteri 35 yaşındadır ve 50,000 ABD Doları kazanmaktadır / bilgisayar satın alır = hayır) = 1/5 = 0.2

Müşteri bir bilgisayar satın alır P(h1/D) = P(h1) * P (D/ h1) / P(D) = 0.5 * 0.6 / 0.4

Müşteri bir bilgisayar satın almaz P(h2/D) = P(h2) * P (D/ h2) / P(D) = 0.5 * 0.2 / 0.4

Kesin Sonuç = arg maks {P(h1/D) , P(h2/D)} = maks(0.6, 0.2)

=> Müşteri bir bilgisayar satın almaz

BÖLÜM 3

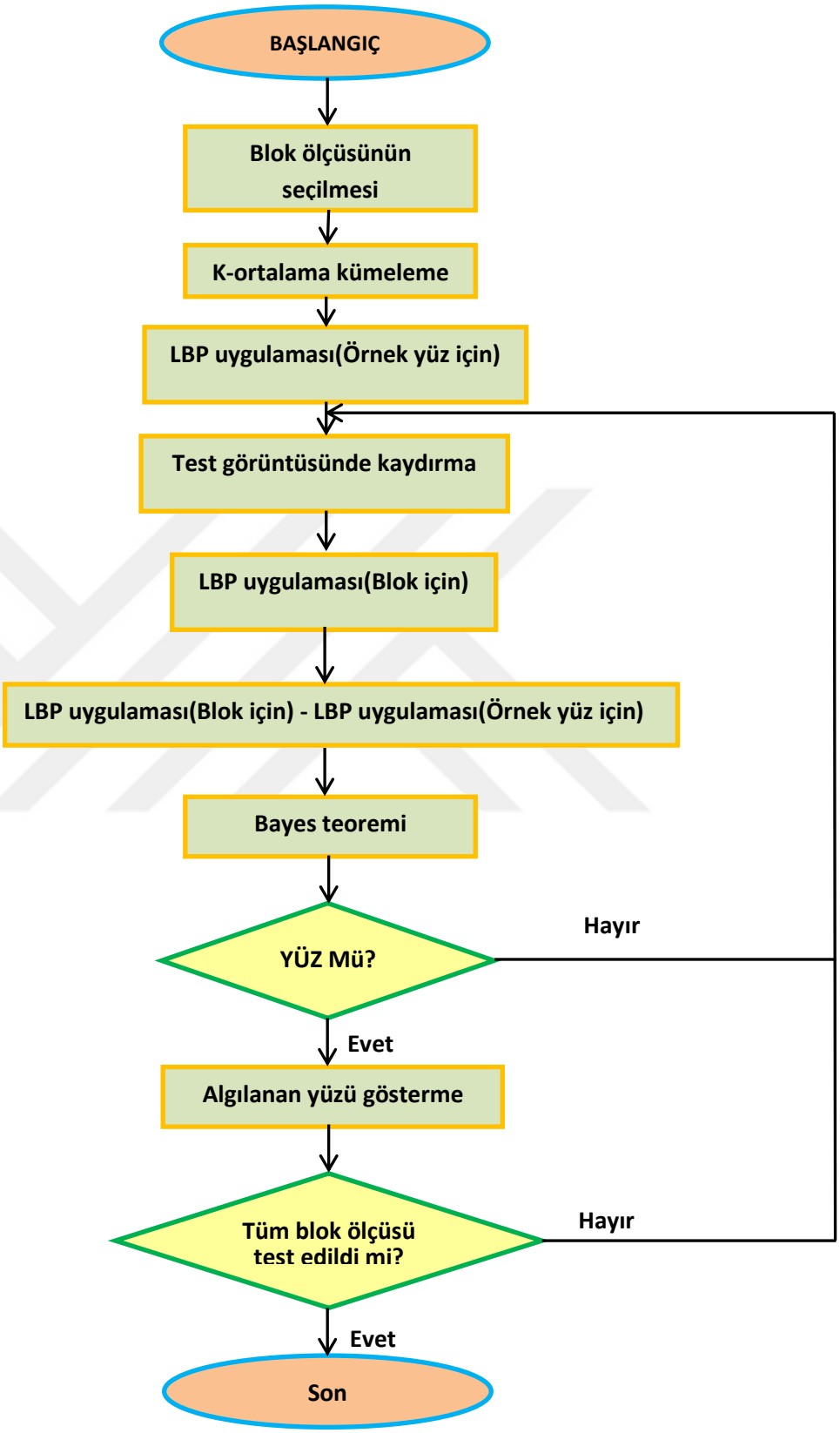
YENİ YÜZ ALGILAMA YÖNTEMİ

Bu yeni yöntem üç kombinasyonu kullanmaktan oluşur

- K-ortalama kümeleme yöntemi.
- LBP algoritması.
- Bayes teoremi.

Yüzü arka plandan ayırma işlemi k-ortalama kümeleme yöntemi ile başlar. Özellik vektörünün bulunması için LBP algoritması uygulanır. Ardından görüntü bloğunun LBP'si ile eğitim örneği görüntüsünün LBP'si arasındaki fark bulunur. Bayes teoremi test edilen parçanın bir yüz görüntüsü olup olmadığına karar vermek için uygulanır.

Yeni algoritma için akış şeması aşağıda bulunan Şekil 3.1'de gösterildiği gibidir:



Şekil 3.1 Önerilen prosedürün akış şeması

3.1 Prosedür

Bu algoritmanın ana fikri, LBP histogramı ile arasındaki benzerlik temelinde bir yüz algılamaktır. Karşılaştırma esasen özellik vektörü temelinde değil, noktaların histogram genliği dağılımı sayıları temelinde olacak ve bu prosedür daha sonra detaylı olarak açıklanacaktır.

Özellik vektörünü oluşturan histogramda, genliklerin alanı üçü pozitif genlikli ve üçü negatif genlikli olmak üzere altı seviyeye ayrılacaktır. Bu noktaların dağılımından sınıflandırma parçasının yüz ile yüz olmayan alanı ayrıştırması mümkün olacaktır.

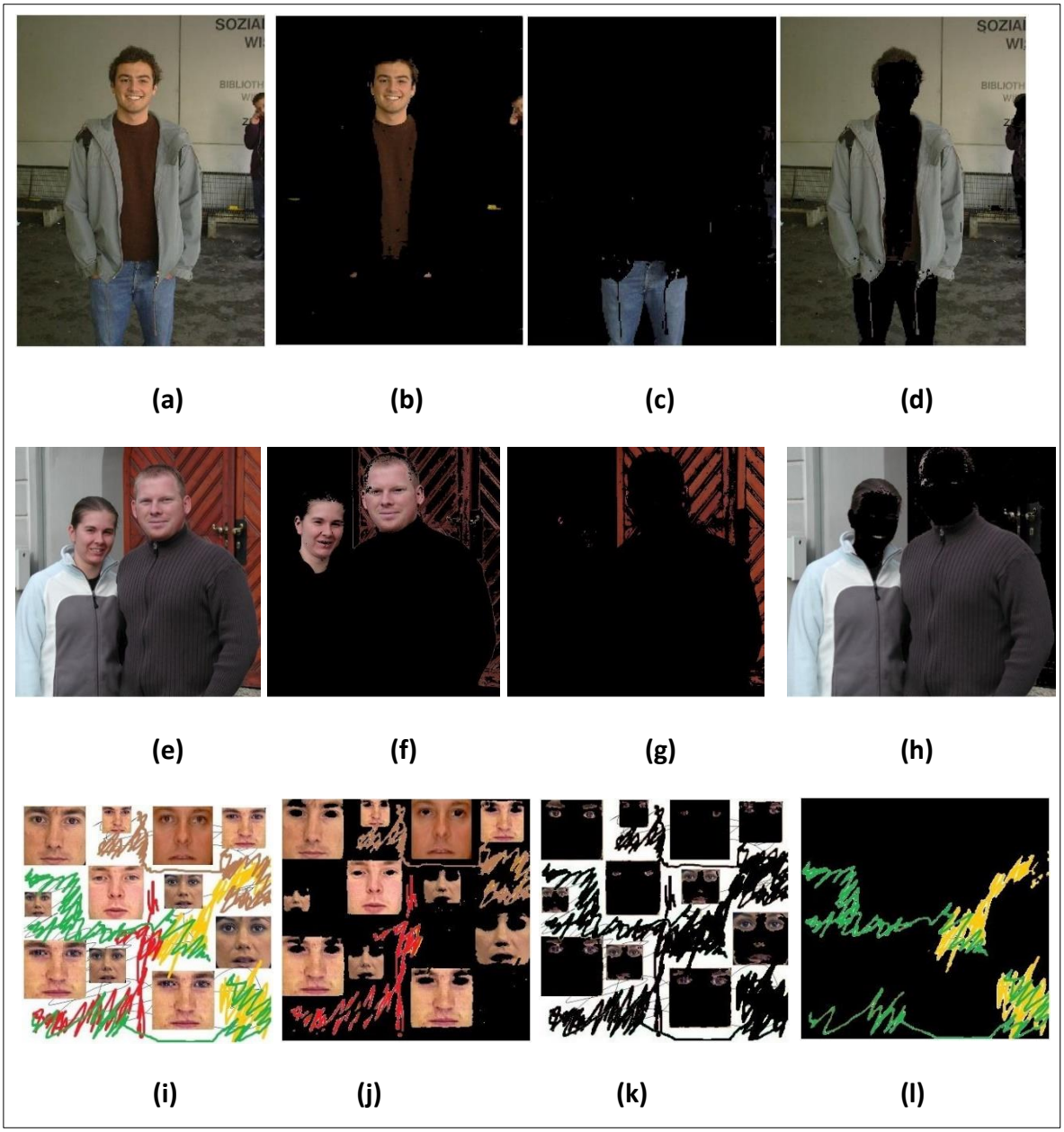
Bu algoritmada en önemli nokta, iki yüz arasında fark olduğunun varsayılmasıdır. Ancak sonunda yüz ile yüz olmayan görüntü arasındaki ilişkide benzeşmeden daha fazlasının olması gereklidir.

Bu yöntem, test altındaki görüntüye k-ortalama kümeleme uygulanması ile açıklanabilir. K-ortalama kümeleme uygulanmasının amacı, görüntüde bulunan diğer elemanlardan yüzleri ayırmak için test edilen görüntüde farklı renklerin ayrıştırılmasıdır. Diğer elemanların karşılaştırılması için hiç zaman kaybetmeden, sadece yüzün bulunduğu alanda karşılaştırma yapılacağı için karşılaştırma sürecinin işlem zamanını düşecektir. Test edilen görüntü bloklara ayrılacak, her bir blok için LBP özellik vektörü hesaplanacak ve örnek görüntünün LBP özellik vektöründen çıkarılacaktır. Aradaki fark, Bayes teoremi maksimum benzeşme (ML) yöntemi uygulanarak değerlendirilecek ve bunun bir yüz elemanı olup olmadığına yönelik farklılık belirlenecektir. Bir yüz olarak değerlendirilmişse blok ile aynı ölçüde bir sınır çizgisi gösterilecektir. Bu süreç, tüm test görüntüsü bloklarının bir yüz içerip içermediğini belirlemek üzere devam edecektir.

3.2 Yüz Ayrıştırması Için K-ortalama Kümeleme Yöntemi

Bu çalışmada sunulan algoritmadaki birinci prosedür, yüzü görüntüden ayrıştırmak amacıyla, yüz algılaması için harcanan zamanın azaltılması için k-ortalama kümeleme kodu kullanılarak görüntü üzerinde ayrıştırma yapmaktır. Bu durumda aşağıdaki sonuçlar elde edilir:

Yüz içeren bir görüntü örnek olarak ele alınmış ve bu görüntüler aşağıda bulunan Şekil (3.2)'de gösterildiği gibi ayrıştırılmıştır.



Şekil 3.2 K-Ortalamalı Kümeleme Yöntemi kullanılarak görüntülerin ayrıştırılması (soldan sağa -orijinal görüntü, ayrıştırma 1, ayrıştırma 2, ayrıştırma 3)

Önceki şekilde yüzlerin uygun bir biçimde görünütüden ayrıştırıldığı görülebildiği için arama alanı düşürülmüş, böylelikle görüntüde bulunan yüzün yaklaşık alanını bulma yönünde iyi bir adım atılmıştır.

Ayrıştırma süreci, görüntüde bulunan yüzler için aday alanın bulunmasında çok önemlidir.

3.3 LBP Özellik Vektörü Farklılıklarının Hesaplanması

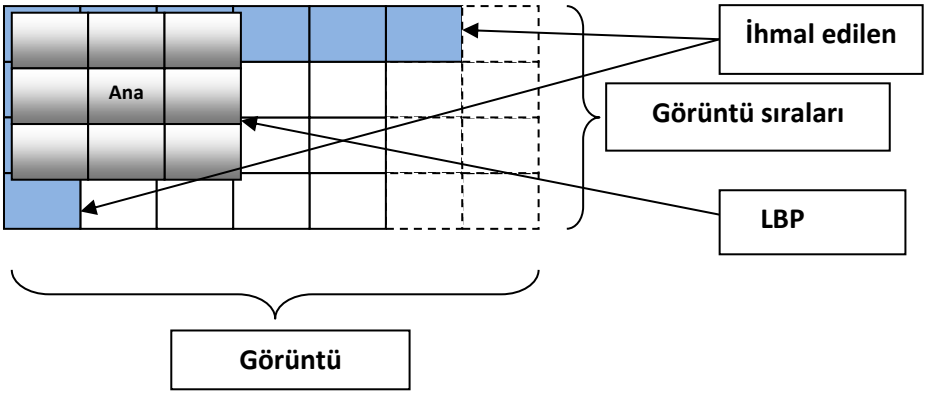
Bu tezde daha önce belirtildiği gibi, yüz algılama, yüz örneğinin LBP'si ile görüntüden alınan bloğun LBP'si arasındaki farklılıklar ele alınarak ve ardından iki özellik vektörü arasındaki benzeşme hesaplanarak ve aşağıda bulunan Şekil 3.3'te görüldüğü gibi yüz alanı ile yüz olmayan alan ayrıştırılarak yapılır.



(a) (b)

Şekil 3.3 (a) Orijinal yüz (b) LBP algoritması uygulandıktan sonraki yüz

Algoritmada değişiklikler yapabilmek ve kendi amacımıza uygun biçimde kullanabilmek için LBP algoritmasının uygulanması tarafımızca yapılmıştır. Her bir (16x16) için (3x3) uygulanması nedeniyle LBP algoritmasının uygulanması sırasında bir sorunla karşılaşılmıştır. Bu sorun görüntünün dış çerçevesinde sınır çizgileri etkisi olarak özetlenebilir. Maske aracılığıyla bu sınırlara ulaşılamamaktadır, bu yüzden, Şekil 3.4'te gösterildiği gibi, LBP maskesi birinci ve sonuncu satırlar ile birinci ve sonuncu sütunları yok sayacaktır.



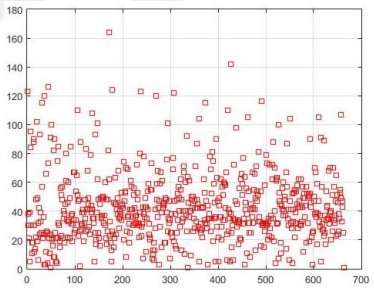
Şekil 3.4 LBP uygulamasında sınır piksellerinin etkisi

Birinci ve sonuncu satırlar ile sütunlarda görüntü üzerine LBP maskesi uygulanamaz. LBP maskesi (3x3) maskenin orta pikseline uygulanacaktır. Birinci ve sonuncu satırların ve sütunların piksellerinde, bu bölgede bulunan pikseller yan piksellere sahip değildir. Bu yüzden (16x16) maske için uygulama işlemi ikinci satırdan başlayarak yapılacak ve sonuncu sıradan önce tamamlanacaktır. Bu işlem LBP uygulaması üzerinde olumsuz bir etki yaratacaktır. Bu sorunun üstesinden gelmek amacıyla, LBP maskesinin görüntü piksellerinin tamamını kapsaması için birinci ve sonuncu satırlar ve sütunların çoğaltılması gereklidir.

3.4 LBP Özellik Vektörü

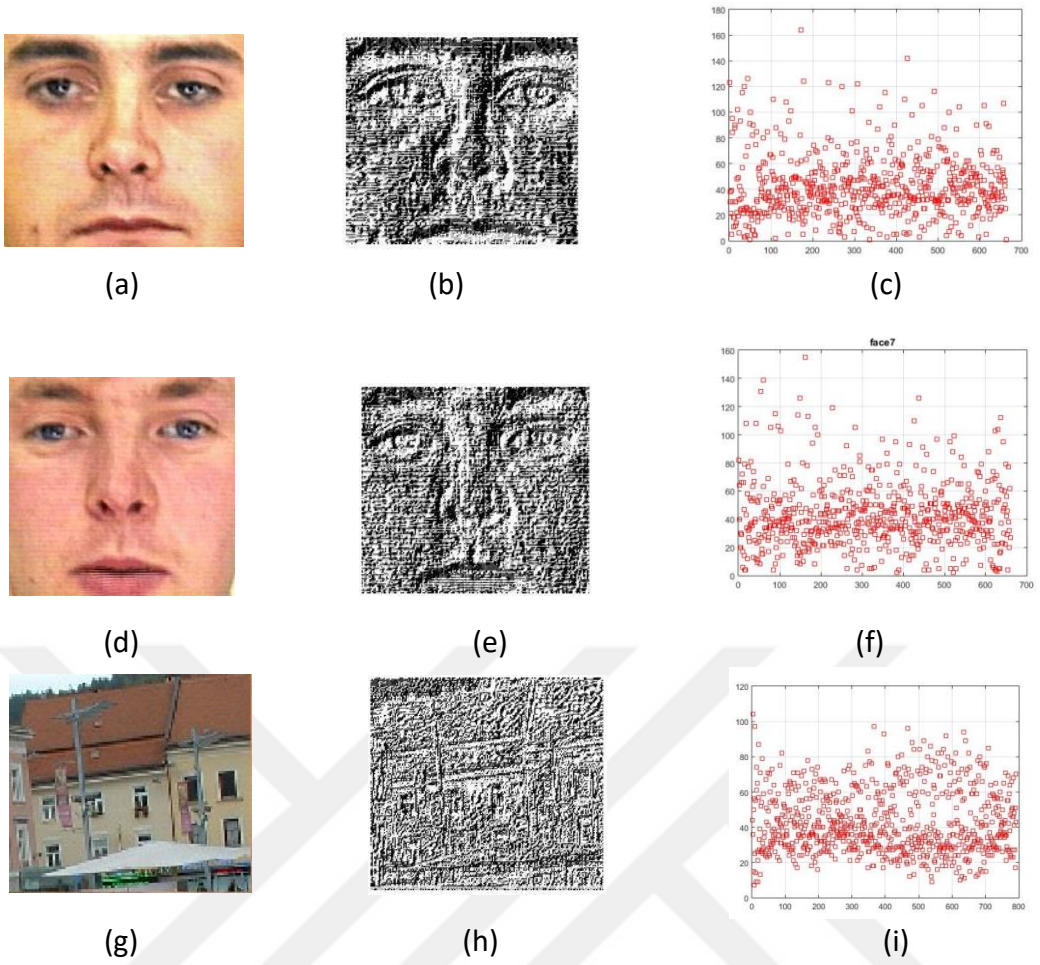
Bir görüntüde LBP uygulamasının kesin sonucu her (16x16) maske için histogramı içeren bir vektör tarafından temsil edilebilir. Küçük vektörlerin seri halinde birleştirilmesiyle, aşağıda bulunan Şekil 3.5'te gösterildiği gibi, test edilen görüntünün son kesin açıklaması oluşturulacaktır:





Şekil 3.5 (a) Test edilen görüntü 169*177 (b) Test edilen görüntü için özellik vektörü

Yüzler ile yüz olmayan nesneler arasında yapılan karşılaştırma, özellik vektörleri arasındaki farklılıklar ele alınarak yapılır. Sınıfların sonuçlandırılmasıyla görüntüde bulunan yüzler ile yüz olmayan nesneler, aşağıda bulunan Şekil 3.6'da gösterildiği gibi ayrıştırılacaktır.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

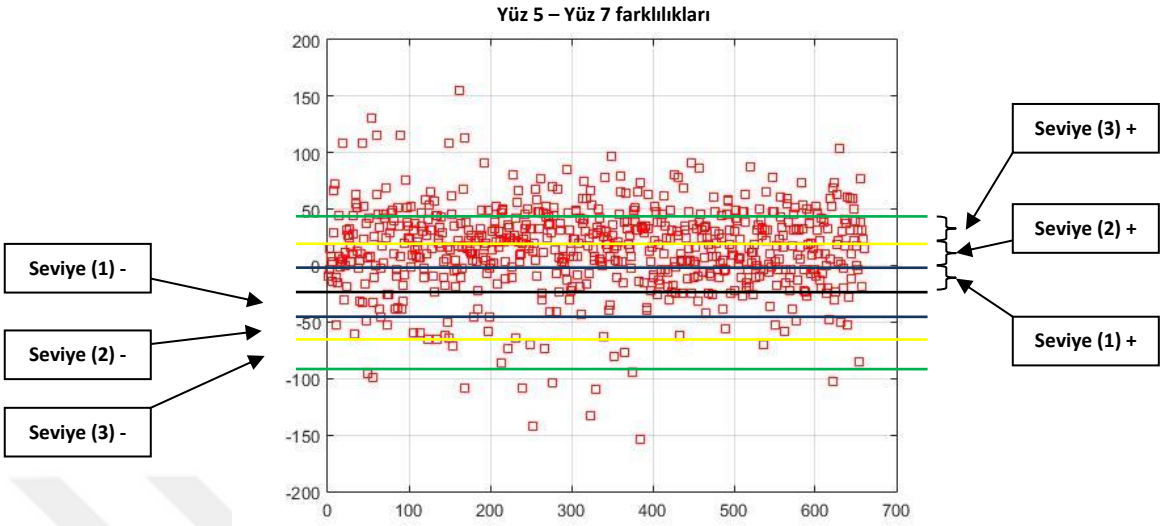
Şekil 3.6 (a) Yüz 5 (b) Yüz 5 LBP görüntüsü (c) Yüz 5 LBP özellik vektörü, (d) Yüz 7 (e) Yüz 7 LBP görüntüsü (f) Yüz 7 LBP özellik vektörü, (g) Görüntü 2 (h) Görüntü 2 LBP görüntüsü (i) Görüntü 2 LBP özellik vektörü

Önerilen algoritmanın ana fikri, özellik vektörlerini sınıflandırarak, bu özellik vektörü tarafından tanımlanan nesnenin bir yüzü veya yüz olmayan bir nesneyi içerdiğini tahmin etmektir. Bizim araştırmamızla yapılan katkı, sınıflandırmanın, bu araştırmanın sonraki konularında açıklandığı gibi, belirli seviyedeki değerler için özellik vektörünün genlik dağılımına göre yapılmasıdır.

3.5 Görüntü Sınıflandırması

Görüntü sınıflandırma işlemi, (**LBP(PADL) farklılıklarının genlik dağılım seviyelerinin olasılığı**) diye adlandırılan bir yöntem önerilerek yapılmaktadır ve bu yöntem, özellik vektörü histogramının genlik değerlerini 6 seviyeye (3'ü pozitif genlikler ve 3'ü negatif genlikler için) bölme süreci olarak açıklanabilir, böylece genlik değerlerinin dağılımlarının sayısından, tanımlanan nesnenin bir yüzü veya yüz olmayan bir nesneyi

içerdiğini tahmin etmek için sınıflandırma işlemi yapılacaktır. Şekil 3.7 dağılım seviyelerini ve bu seviyelerde nokta oluşumlarını göstermektedir.



Şekil 3.7 Yüz 5 ile Yüz 7 farklılıkları

Deney sonuçları ile noktaların dağılım seviyeleri, aşağıda gösterilen Çizelge 3.1'de gösterildiği gibi yapılmıştır:

Çizelge 3.1 LBP farklılıklar dağılım seviyeler

numara	seviye	Alt sınır	Üst sınır
1	Seviye (1) pozitif	0	20
2	Seviye (2) pozitif	20	40
3	Seviye (3) pozitif	40	60
4	Seviye (1) negatif	-20	0
5	Seviye (2) negatif	-40	-20
6	Seviye (3) negatif	-60	-40

Bu algoritmanın kullanılmasının amacı, sınıflandırma işlemini basitleştirmek ve daha sonra açıklanacağı gibi büyük miktarlarda veri aktarımına gerek kalmamasını sağlamaktır.

Bu araştırma için yüzleri içeren basit bir veri seti hazırlanmıştır. Bu veri setinde 10 adet yüz ve 10 adet yüz olmayan nesne ya da görüntü bulunmaktadır. Yüzey veri setleri yüzün bütününden değil, kaşlardan başlayan ve dudakların altında sonlanan bir alandan alınmıştır, bunun nedeni:

1. Algılama için hesaplama süresini düşürmektir.
2. Bu alanın dışında kalan bilgiler, sınıflandırma süreci için faydalı bilgiler içermez.

Bu yüzlerde yapılan deneylerin sonuçları, iki veri seti için Çizelge 3.2'de gösterilmiştir.

Çizelge 3.2'den görülebileceği gibi, önerilen yöntemde yüzlerin verileri yüz olmayan görüntü verilerinden ayrıca sınıflandırılabilir ve maksimum benzerlik (ML) yöntemi kullanılarak kolaylıkla sınıflandırılabilir. Maksimum benzerlik (ML), bu sınıflandırma sürecinin aşağıdaki gerekçelerle kullanılması gerçeğinden ortaya çıkmıştır:

1. Kodunun uygulamaya konulması kolaydır.
2. Hesaplaması daha az karmaşıklık içerir ve daha az zaman harcanır.
3. Eğitim süreci için çok fazla veri gerektirmez.

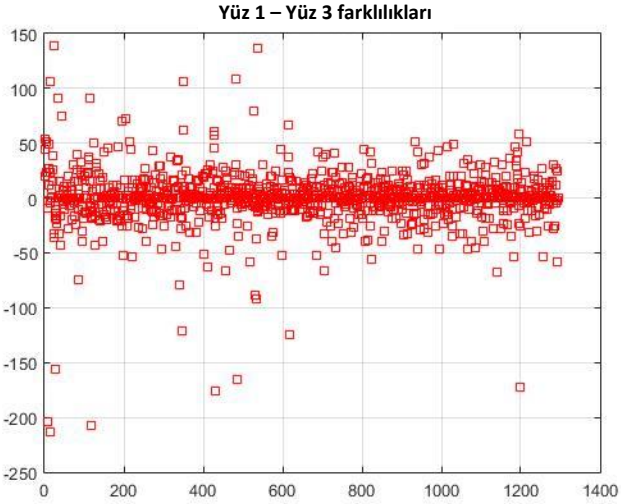
Çizelge 3.2 Yüzler ve yüz değerleri ile yüzler ve görüntü değerleri arasındaki farklılıklar

No.	LBP farkı	Seviye (1) Poz.	Seviye (2) Poz.	Seviye (3) Poz.	Sıfırların sayısı	Seviye (1) Neg.	Seviye (2) Neg.	Seviye (3) Neg.
1.	Yüz 1 - Yüz 5	294	94	15	427	271	93	31
2.	Yüz 2 - Yüz 5	335	79	27	421	287	71	20
3.	Yüz 3- Yüz 5	338	94	21	408	282	76	19
4.	Yüz 4 - Yüz 5	311	113	29	417	247	89	27
5.	Yüz 10 - Yüz 5	317	93	26	411	251	85	25
6.	Yüz 6 - Yüz 5	314	81	22	441	312	66	17
7.	Yüz 7 - Yüz 5	291	94	22	435	297	84	19
8.	Yüz 8 - Yüz 5	320	78	36	425	269	97	21
9.	Yüz 9 - Yüz 5	314	100	16	426	251	102	27

Çizelge 3.2 Yüzler ve yüz değerleri ile yüzler ve görüntü değerleri arasındaki farklılıklar (devam)

10.	Yüz 5 – Yüz 5	0	0	0	1296	0	0	0
11.	Yüz 1 - Görüntü 1	162	299	163	377	56	32	17
12.	Yüz 1 - Görüntü 2	250	137	44	395	215	98	46
13.	Yüz 1 - Görüntü 3	131	299	173	320	59	78	35
14.	Yüz 1 - Görüntü 4	266	132	46	399	197	123	48
15.	Yüz 1 - Görüntü 5	239	156	55	384	198	90	58
16.	Yüz 1 - Görüntü 6	143	258	152	219	137	114	61
17.	Yüz 1 - Görüntü 7	230	158	51	395	193	103	58
18.	Yüz 1 - Görüntü 8	128	298	172	286	107	38	37
19.	Yüz 1 - Görüntü 9	245	178	39	413	139	78	59

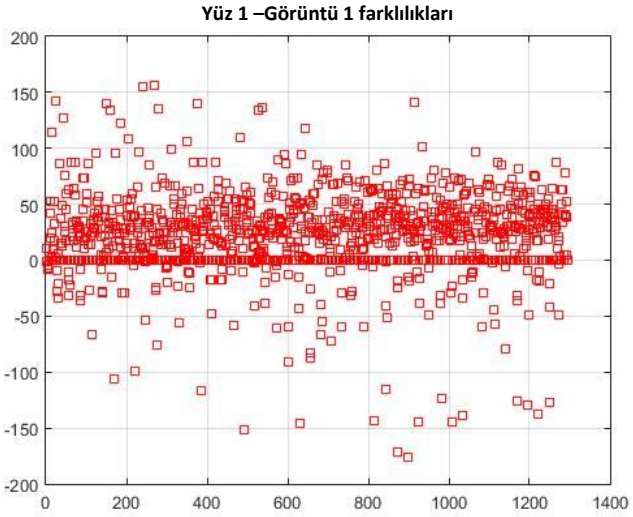
Çizelge 3.2'de verilen değerler her bir seviye için nokta oluşumlarını temsil etmektedir. Bu konu yüzler ile yüzler ve yüzler ile yüz olmayan nesneler için LBP özellik vektörü arasında farklılıkları temsil eden şekillerle, aşağıda bulunan Şekil 3.8'de gösterildiği gibi sonuçlandırılmıştır:





Yüz 1 Yüz 3

(a) Yüz 1 - Yüz 3





Yüz 1 Görüntü 1

(b) Yüz 1 - Görüntü 1

Şekil 3.8 Şekiller yüz ve yüz ile yüz ve görüntü arasındaki farklılıkları temsil eder

Örnek görüntü ölçülerinin (193x205) olduğunu, LBP'nin RGB renk skalası düzeninden gri skala düzenine aktarıldığının belirtilmesi önemlidir.

Yüz ve yüz karşılaştırmasında, iki yüz arasında bazı benzerlikler olduğu için, LBP özelliği farklılıklarının tepe noktalarının ana dağılımının sıfır seviyesine yakın olduğu kesinlikle açıktır. Ancak yüz ve görüntü karşılaştırmasında, yüz ve yüz karşılaştırma görüntülerinden farklı olarak, noktaların büyük bir kısmı sıfır seviyesinden uzak olmaktadır. Bu konudan başlayarak, Çizelge 3.2'de gösterilen değerler ile nesnelerin sınıflandırılması kolaydır.

3.6 Deney Sonuçları

(**PADL**) algoritmasını uygulamak için rastlantısal görüntüler kullanılmış ve aşağıda bulunan Şekil 3.9'te gösterildiği gibi örnek bir yüzü ele alınmıştır

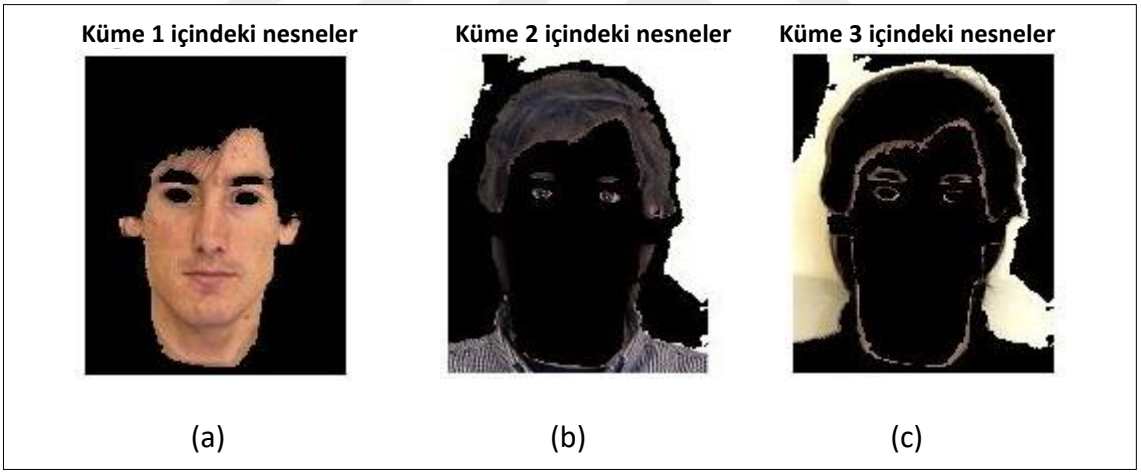


Şekil 3.9 Test edilen örnek yüz

Örnek görüntüye aşağıdaki adımlar uygulanmıştır:

3.6.1 K-ortalama Kümeleme

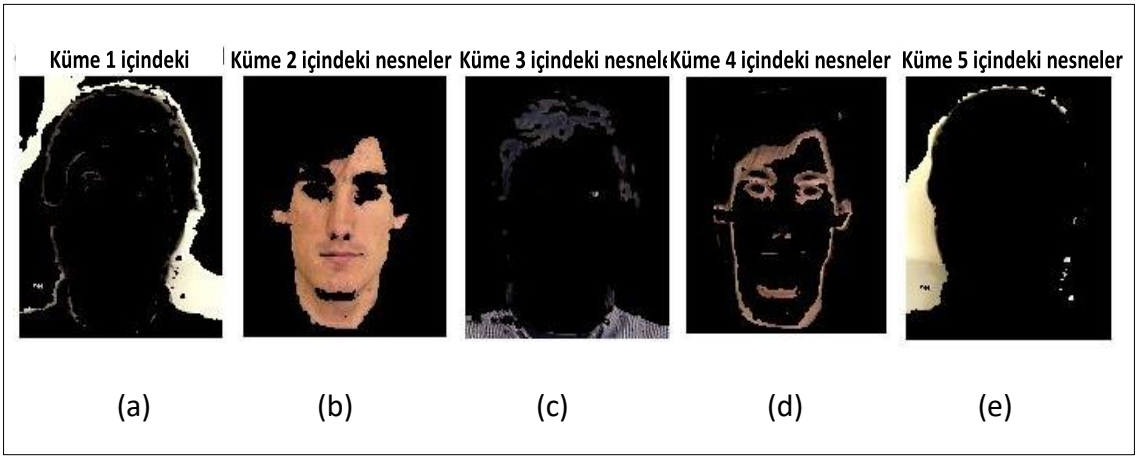
Birinci adım örnek görüntüde K-ortalama kümeleme uygulamaktır. Bunun sonucu aşağıda bulunan Şekil 3.10'da gösterildiği gibidir:



Şekil 3.10 3 Bölgeye ayrılmış görüntüde renkler ayrıştırılarak renk aralıklarına göre ayrıştırılmış görüntüler

K-ortalama kümeleme için renk aralığında düşük değerler seçilmesi gerektiğinin belirtilmesi oldukça önemlidir. Yüz elemanlarını ayrıştırmak için (3) veya (4) seçilmesini örnek olarak ele alalım. Bu durumun gerçekleşmesi durumunda yüz elemanlarının bölgelere ayrıştırılması ve tüm yüzeyin ayrıştırma süreci ile yüz algılama sürecinin başarısız olacağı kesindir.

Ayrıştırma için (3) veya (4) renk aralıklarından fazlasının kullanılması sorunu, aşağıda bulunan Şekil 3.11'de görülebilir:



Şekil 3.11 Renk bölgelerinin (5) bölgede olacağı seçildiğinde görüntü için beş ayrıştırma

Şekil 3.11 (b)'den renk ayrıştırma fazlalaştıkça yüz elemanlarının kaybolacağı sonucuna kolaylıkla ulaşılabilir. Yüz algılama sürecinde hata payı fazlalaşacak ve algılama sürecinde hataların fazlalaşmasıyla renk ayrıştırma bölgelerinin fazlalaşma verimi görülebilecektir.

3.6.2 Kutuları Kaydırma

Sonraki adım yüzün örnek görüntüsünü almak ve bunun için LBP hesaplamak olacaktır. Bundan sonra test edilen görüntü (mxn) adet bloklara bölünerek kaydırma süreci başlatılacaktır. Burada n ve m (örnek yüz görüntüsü) ile aynı ölçülere sahiptir. Daha sonra kaydırma işlemi aşağıdaki gibi yapılır:

- Test edilen görüntüden (mxn) ölçülerinde bir parça alınır. Burada m ve n değerleri yüz örneği görüntüsünün ölçüsünden alınmalıdır.
- Örnek yüz görüntülerinin her ikisinin LBP'si hesaplanır ve aşağıda bulunan Denklem 3.1'de belirtildiği gibi aralarındaki fark bulunur:

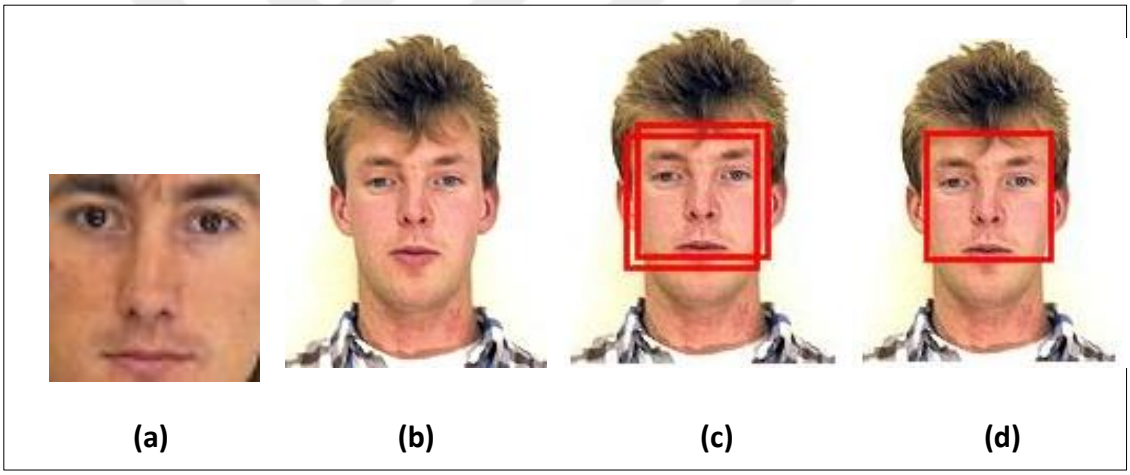
$$LBP_{diff} = LBP_{f.s.m} - LBP_{görüntü\ bloku} \quad (3.1)$$

- LBP farklılıklarından, elemanlarına göre değil, Çizelge 3.2'de gösterildiği gibi, altı seviyede genlik dağılımına göre özellik vektörü kolaylıkla sınıflandırılabilir.

- Eğer sınıflandırma görüntü bloğu içindeki bilgilerin bir yüz olduğunu doğrularsa, blok görüntünün çevresine kırmızı renkli bir kutu çizilecektir. Süreç diğer bloklar için teker teker sonuna kadar devam edecektir.

- Önemli bilgileri (yüzleri) içeren görüntü çerçevesini tanımlanamadığı için süreç ayrıştırılmış renk bölgelerinin tamamı için yapılacaktır. Her ne kadar bu sürecin hesaplanması zahmetli gibi görünse de bundan kaçınmak için, kodun içine eğer bölge sıfırlardan oluşan eşik değerler içeriyorsa testin atlanmasını belirten basit bir koşul eklenebilir.

Bir örnek olarak, aşağıda bulunan Şekil 3.12(a)'de gösterildiği gibi bir örnek görüntü alalım. Ardından test edilen bir görüntü üzerinde, aşağıda bulunan Şekil 3.12(b)'de gösterildiği gibi bir prosedür uygulayalım. Sonuç Şekil 3.12(c)'de gösterildiği gibi olacaktır.



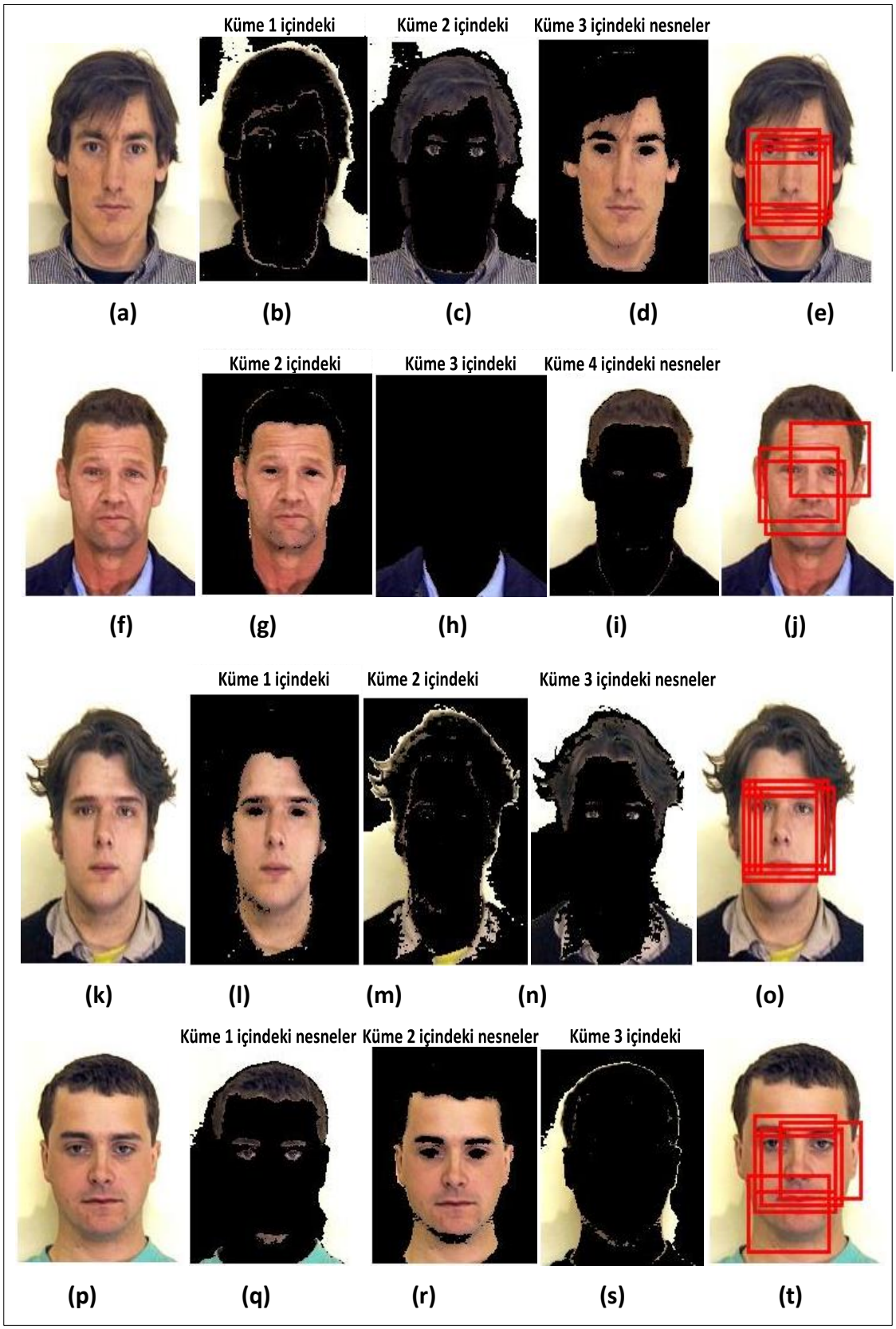
Şekil 3.12 (a) Test edilen bir görüntü bloku için LBP'ye benzerliğini bulmak üzere kendi LBP özellik vektörünü kullanarak algılama süreci uygulanan örnek yüz görüntüsü, (b) test edilen görüntü, (c) sıfır sayıları karşılaştırması olmadan algılanan yüz, (d) sıfır sayıları karşılaştırması ile algılanan yüz.

Burada algılanan yüzü çevreleyen iki kutu görülmektedir. Bunun nedeni kayar kutuların biraz hareket etmesidir. Yüzün içinde kalan alanda birkaç diğer konumda da benzerlik koşulu doğru olmaya devam edecektir.

Bu sorunun üstesinden gelmek için kutuların sol üst köşesindeki noktalar ve ardından ortalama değeri alınmıştır. Böylece bir kutunun sol üst köşe noktası olan bu yeni değer, Şekil 3.12(d)'de gösterildiği gibi sonuç olarak elde edilen yüzün konumunu temsil etmektedir.

Önerilen algoritma ile yüz algılaması sonuçlarını test etmek için farklı bir veri seti (Aberdeen'deki Craw'dan 687 renkli yüz bilgisi içeren 2D bir yüz veri seti) kullanıldı. 1 ile 18 arasındaki görüntüler 90 bireyin görüntüleridir. Aydınlatmada bazı varyasyonlar bulunmaktadır. 8 adedinde değiştirilen bakış açıları bulunmaktadır. Çözünürlük 336x480 ile 624x544 arasındadır. Ancak bu veri setinde içerilen yüzlerin ölçüleri, işlem yapma ve algılama süresinin düşürülmesi için azaltılmıştır.

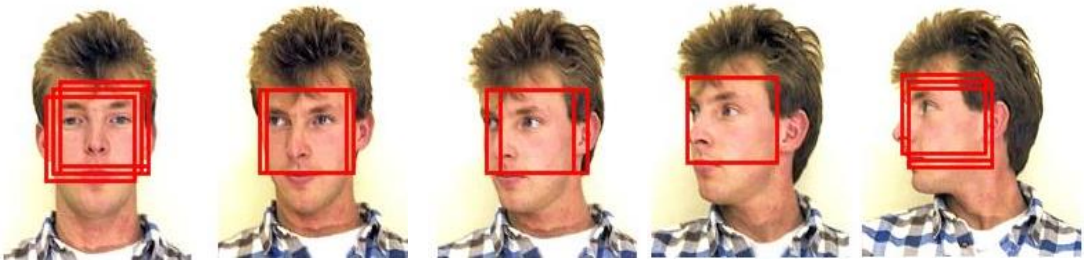
Algoritma örnek yüzlere uygulandıktan sonra elde edilen sonuçlardan bazıları aşağıda gösterilmiştir.



Şekil 3.13 Değiştirilmiş (aberdeen) veri setinden algılanan yüzlerin örneği

3.7 Sonuç

Önerilen algoritmanın sağladığı en önemli katkı tam karşıya bakmayan yüzlerin ve yandan görünen yüzlerin algılanabilme becerisidir. Aşağıda bulunan Şekil 3.14'te gösterildiği gibi, test edilen görüntünün herhangi bir yerinde bulunan yüz dokusu algılanmaktadır.



(0) derece **(22) derece** **(45) derece** **(67) derece** **(90) derece**

Şekil 3.14 Önerilen algoritma kullanılarak tam karşıya bakmayan yüzlerin algılanması örneği

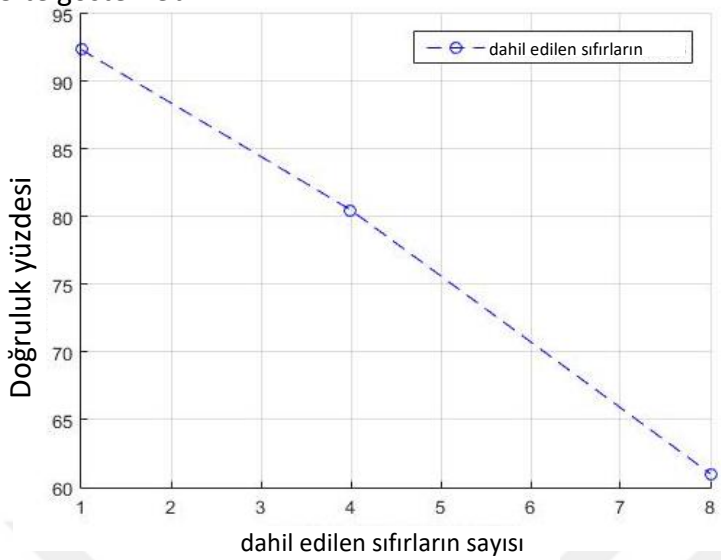
- Çizelgede sıfırların sayısı, karşılaştırmada, çevreleyen boş piksellerin karşılaştırma kutusuna dahil edildiği ilgili bölgeyi seçmek anlamına gelir.

Çizelge 3.3 Görüntü karşılaştırması için sıfırların sayısının sıfır olmayanlara piksel oranı

No	Kutudaki sıfırların sayısı/kutudaki sıfır olmayanların sayısı	Algılanan kişi sayısı	Algılanmayan kişi sayısı	Doğruluk yüzdesi
1.	1/1	118	9	%92,37
2.	1/4	118	23	%80,508
3.	1/8	118	46	%61,016
4.	1/12	-	-	-

Her bir bloğun karşılaştırılması sürecinde, her bir bloktaki sıfır piksellerin ve sıfır olmayan piksellerin sayısını hesaplayarak karşılaştırma süresini azaltacak basit bir prosedür hazırlanmıştır. Bu kapsamda bir eşik değer seçilmiştir. Oranın eşik değerini aşması halinde karşılaştırma atlanacak, aksi takdirde işlenecektir.

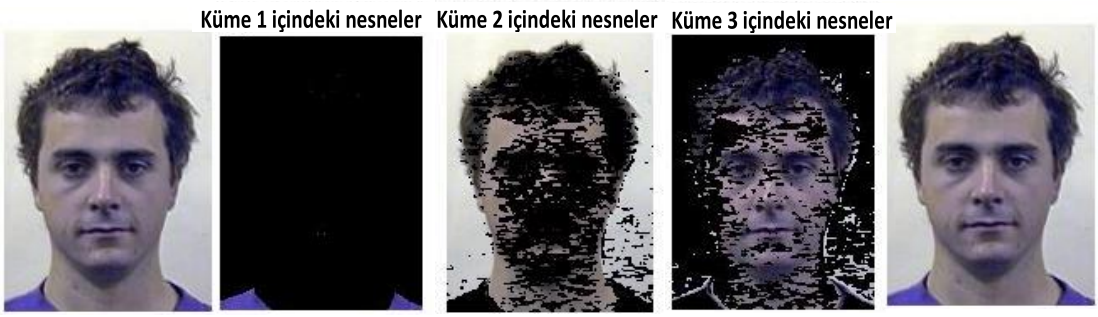
Bulgu yüzlerinin doğruluğu ile dahil edilen sıfırların sayısı arasındaki ilişki aşağıdaki şekilde 3.15'te gösterilebilir:



Şekil 3.15 Yüzleri bulma doğruluğu - dahil edilen sıfırların sayısı

3.8 Sistem Zayıflığı

Değiştirilmiş (aberdeen) veri setinde bulunan yüzlerin testinde Şekil 3.16'da gösterildiği gibi sistemin başarısız olduğu bazı durumlar da oluşmuştur.



Şekil 3.16 (a) Örnek görüntü (b) küme 1 (c) küme 2 (d) küme 3 (e) sonuç görüntü

K-ortalama kümeleme yönteminin yüz bölgesini diğer bölgelerden ayırmakta başarısız olması nedeniyle ön işlem yapma adımı da başarısız olmuştur. Bunun sonucunda yüz algılaması başarılı olarak yapılamamıştır. Bunun gerekçesi, yüz renginin arka plan renk aralığına yakın olmasıdır. Bu nedenle, Şekil 3.16(c) ve (d)'da görülebileceği gibi, yüz bölgesinin tamamı ayrıştırılamamaktadır.

BÖLÜM 4

YÜZ TANIMA

4.1 Yüz Tanıma

Yüz tanıma bilgisayarla görüntülemede en önemli ve zorlu araştırma başlıklarından biridir. Görüntüler veya video çerçeveleri temelinde insanların kimliklerini tanımayı amaçlar.

Yüz tanıma, aşağıdakiler nedeniyle zorlayıcı bir yapısı olan insan yüzünü araştıran ilgi çekici bir araştırma başlığıdır:

- Başın yatay ve dikey hareketlerinde çok fazla değişkenlik.
- Aydınlatma yoğunluğu ve açısı.
- Yüz ifadesi.
- Yaşlanma.
- Kapanma, vb.

Yüz tanıma sistemi çok önemli uygulama alanlarında kullanılmaktadır: biyometrik sistemler, insan bilgisayar karşılıklı etkileşimi, sanal gerçeklik, veri tabanı kurtarma, multimedya, sürücü belgeleri, otomatik kimlik doğrulama, sınır kontrolleri, yasal zorunluluklar, video gözetimleri, araştırma, kişisel güvenlik, sürücü izleme sistemi, bilgisayarla eğlence, bilgi güvenliği, işletim sistemi, medikal kayıtlar, çevrim içi bankacılık, kişisel kimlik kartları, pasaportlar.

Bütünsel yaklaşım gibi, görünüm tabanlı yöntemi, özellik tabanlı yaklaşımı ve hibrit yaklaşımı içeren pek çok türde yaklaşım bulunmaktadır. Bütünsel yaklaşımda, yüz alanının bütünü yüz algılama sistemi içine girdi olarak verilir. Bütünsel yöntemlerin en iyi örneklerinden biri olan Eigen faces, Şekil 4.1'de gösterildiği gibi, yüz tanıma için kullanılan en yaygın yöntemdir. Bu yöntemle önce gözler, burun ve ağız gibi lokal özellikler ile bunların konumları çıkarılır ve lokal istatistikleri (geometrik ve/veya görünüm) yapısal bir sınıflandırıcıya uygulanır.

Özellik çıkarma yöntemleri için özellik "restorasyonu" büyük bir zorluktur. Bu durum sistem büyük farklılıklar nedeniyle görünmez olan özellikleri bulmayı denediği zaman ortaya çıkar. Örneğin önden bir yüz görüntüsü ile yandan bir yüz görüntüsü karşılaştırılmaya çalışıldığı zaman kullanılır. Çeşitli özellik çıkarma yöntemleri bulunmaktadır. Bunlardan birincisi kenarlar, çizgiler ve eğriler temelindeki jenerik yöntemlerdir. İkincisi ise özellik-şablon tabanlı yöntemler ve üçüncüsü ise özelliklerin geometrik kısıtlamalarını dikkate alan yapısal eşleştirme yöntemleridir. Hibrit yaklaşım bütünsel ve özellik tabanlı yaklaşımların her ikisini bir arada kullanır. Hibrit yaklaşım, Huang tarafından geliştirilmiş bir sistemde kullanılmıştır.

Opencv'de yüz algılama için başlıca üç yöntem bulunmaktadır:

- Eigenfaces.
- Fisherfaces.
- Lokal İkili Değer Düzenleri Histogramları (LBPH).

Bu üç yöntemde yeni yüzler, bilinen yüzlerden oluşan bazı eğitim yüzü görüntüleri ile karşılaştırılır. Bu üç yöntemin tamamı diğerlerinden farklı bir eğitim seti kullanır.

Eigenfaces ve Fisherfaces yüzün en önemli özelliğini açıklamak için matematiksel bir tanımlama kullanırken, Lokal İkili Değer Düzenleri Histogramları (LBPH) eğitim setinde bulunan her bir yüzü ayrı ayrı analiz eder.

Örnek bir eğitim seti, aşağıda bulunan Şekil 4.1'de gösterilmiştir.



Şekil 4.1 Eğitim seti

4.1.1 Eigenfaces ve Fisherfaces

Doğrusal cebirde, her bir vektör alanının dik açılı bir temeli olduğu bilinir. Bu temelin elemanları bir araya getirilerek, her vektör bu vektör alanında birleştirilebilir. Bunun tersi de doğrudur, vektör alanında bulunan her bir vektör temel elemanlarına ayrılabilir.



Şekil 4.2 Şekil 4.1'de gösterilen görüntünün Eigen vektörleri

Sisteme bilinmeyen yeni bir yüz verildiği zaman, sistem bulunan temele göre bunu ayırabilir ve ardından hangi Eigen vektörünün (vektörlerinin) yüzlerin büyük bir kısmını "açıklayabildiğini" görebilir ve böylece bu yüzün kime ait olduğunu belirler.

4.1.2 Lokal İkili Değer Düzenleri Histogramı

LBPH ile eigenfaces yöntemi arasındaki en önemli fark, LBPH'nin her bir yüzü sadece veri setinden analiz etmesi, ancak eigenfaces'in veri setini bir bütün olarak analiz etmesidir.

Bir başka fark, her bir görüntünün veri setinde lokal olarak uygulanması nedeniyle LBPH'nin daha kolay olmasıdır. Şekil 4.3'te Şekil 4.1 veri setinin LBP analiz çıktıları gösterilmiştir.



Şekil 4.3 Şekil 4.1 veri setinin LBP analiz çıktıları

4.2 Önerilen Yüz Tanıma (PADL Temelinde)

Önerilen algoritmanın en önemli katkısı, yüzlerin aynı zamanda algılanması ve tanınmasıdır. Yüzler arasında bazı benzerlikler olduğunu ve yüzler ile görüntüde bulunan diğer nesneler arasında büyük ölçüde benzeşme olmadığını kanıtladığımız gerçeğine bağlı olarak ve farklı görüntülerde aynı yüzün birebir aynı şekilde görünmesi sebebiyle, tanıma prosedürü adım adım algılama prosedürü ile aynıdır. Aralarındaki tek fark sınıflandırma kararındadır. Ancak bu konu ileride yapılacak bir araştırmada incelenecektir.

Aynı yüzün farklı görüntüleri ile görüntüde bulunan diğer yüzler ve diğer nesneler arasındaki değerleri anlamak için aşağıda bulunan Çizelge 4.1 incelenebilir.

Çizelge 4.1 Yüzün kendisi ile diğer yüzler ve başka görüntüler arasındaki farklar

No.	LBP farkı	Seviye (1) Poz.	Seviye (2) Poz.	Seviye (3) Poz.	Sıfırları n sayısı	Seviye (1) Neg.	Seviye (2) Neg.	Seviye (3) Neg.
1.	Yüz 1a - Yüz 1b	415	13	1	460	362	26	2
2.	Yüz 1b - Yüz 1c	408	19	0	458	374	24	1
3.	Yüz 1c - Yüz 1a	385	38	6	435	348	44	7
4.	Yüz 1a - Yüz 3	335	79	27	421	287	71	20
5.	Yüz 1a - Yüz 4	338	94	21	408	282	76	19
6.	Yüz 1a - Yüz 5	311	113	29	417	247	89	27
7.	Yüz 1a - Görüntü 1	162	299	163	377	56	32	17
8.	Yüz 1a - Görüntü 2	250	137	44	395	215	98	46
9.	Yüz 1a - Görüntü 3	131	299	173	320	59	78	35

Çizelge 4.1'den yüzün kendisi ile diğer yüzler arasında bir fark olmasının tanıma adımını temsil ettiği sonucuna varılabilir. Bu şekilde yüz ile görüntüler arasındaki fark, algılama adımını temsil etmektedir ve bu algoritmanın yüzleri algılama ve tanıma süreçlerini aynı sürede yapabildiği anlamına gelmektedir.

Ayrıca, Çizelge 4.1'de ilk üç satırın yüzün kendisi için 0-20 aralığında bir maksimum genlik değerine karşılık geldiği ve bu değerlerin ilk sütundaki diğer 6 değerden büyük olduğu görülebildiği için kolaylık sağlamak açısından yüzler ve görüntüler kod içerisinde basit bir şekilde (eğer koşulu ile) sınıflandırılabilir. Bu şekilde, sınıflandırmaya yönelik basit bir prosedürle yüz, aynı zamanda kolayca algılanıp tanınabilir.

4.3 Yüz Tanıma (ORB (Yönelimli FAST ve Döndürülmüş BRIEF) temelinde)

Yüz tanıma, bilgisayar görüşü alanında son derece zorlu bir konudur. Kişisel bilgisayar görüş sistemindeki en büyük problem, gerçek zamanlı uygulamalarda pahalı tanıma algoritmaları veya öznitelik dönüştürme algoritmalarıyla başa çıkamamasıdır. SIFT veya SURF şeklindeki bu algoritmalar iyi bir performansa sahiptir fakat özerk veya çevrimiçi tanımayı çok yavaş ve verimsiz bir şekilde gerçekleştirirler. ORB (Yönelimli FAST ve Döndürülmüş BRIEF) adlı algoritma bu probleme yönelik iyi bir çözümdür ve çevrimiçi nesne tanımada en uygun performansa sahip olduğunu kanıtlamaktadır.

ORB'nin en önemli avantajları şu özelliğe sahip olmasıdır:

1. ORB, BRIEF [19] (İkili Sağlam Bağımsız Temel Öznitelikler) ve fast tanımlayıcı metodolojilerini iyileştirir.
2. SIFT ve SURF tanımlayıcılar üzerinde bulunan lisans sınırlamasını içermez.
3. Döndürmeden ve ölçekten bağımsızdır.
4. GPU kullanımı olmadan düşük güçlü cihazları çalıştırabilmek için gerçek zamanlı uygulamalarda kullanılabilir.

4.3.1 (FAST) Köşe Algılama

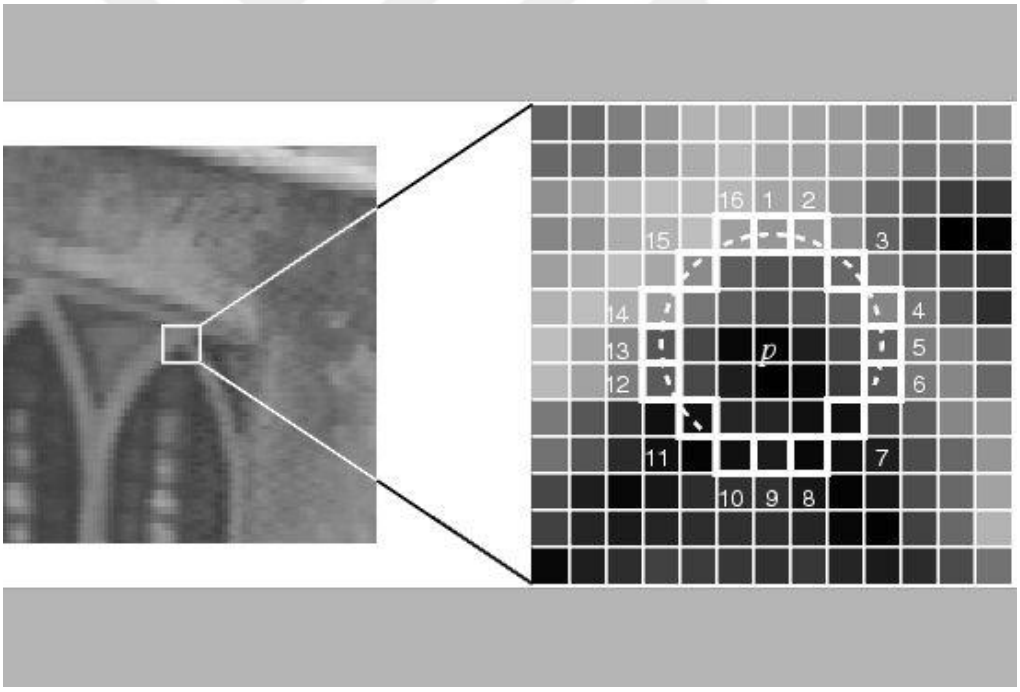
Köşe algılamaya yönelik FAST [20] özniteliği, öznitelik dönüştürme algoritmalarının çoğu için anahtar sözcüktür. İlk defa Rosten ve Drummond tarafından tanıtılmıştır. Ana amacı görüntüdeki belirli noktaları bulmaktır. Bu noktalar köşelerdir, fakat bu köşeler görüntüdeki pek çok köşe arasında en güçlü olanlardır. En güçlü köşe noktaları, ölçek değişiminden etkilenmeyecekleri anlamına gelir.

Yönelim ve aydınlatmadan bağımsız olduğu ve uygulama kolaylığına sahip olduğu için fast algoritmasını kullanarak köşeleri bulma metodu son derece verimlidir. Sadece pikseller seviyesinde çalışır. Bu metod aşağıdaki adımlarla özetlenebilir:

1. Herhangi bir görüntüde (p) adlı bir piksel test edilen noktadır.
2. Eşik değerinin (T) örneğin %30 olarak ayarlanması (Kullanım alanı daha sonra açıklanacaktır.)

3. Şekil 4.4'te gösterildiği üzere test edilen pikseli (p) çevreleyen daireler şeklinde (16) pikseller seçilirken pikselin (p) diğer piksellerle arasındaki mesafe (3) piksel olmalıdır (bu düzenleme Bresenham dairesi olarak adlandırılır).
4. Çevreleyen pikseller (p-T) veya (p+T) değerinin üzerinde veya altında (N) komşu piksel değerine sahipse, noktanın (p) bir ilgi noktası olduğu söylenebilir.
5. Bir köşenin varlığını daha hızlı bir şekilde algılayabilmek için, noktanın kontrol mü edileceği yoksa göz ardı mı edileceği kararını hemen vermek için bir başlangıç testi yapılmalı ve bu test çevreleyen dairedeki (1), (5), (9), (13) noktaları kontrol edilerek gerçekleştirilmelidir.

En az 3 nokta (p-T) veya (p+T) altında veya üzerinde değilse, nokta bir köşe değildir ve ayrıntılı testi yapmadan önce göz ardı edilecektir. Bu prosedür aşağıda şekil 4.4'te gösterilmektedir:



Şekil 4.4 İlgi noktasını algılamak için FAST köşe algılama

Komşu piksel sayısının (N) seçilmesi bir ölçüde kafa karıştırıcıdır, çünkü (12)'den az olduğu takdirde bu durum çok sayıda pikselin seçileceği anlamına gelir. Bazı çalışmalarda (N=9) değerinin köşe algılama için optimum[20] değer olduğu söylenmesine rağmen bu algoritmanın doğruluğunu etkileyebilmektedir.

Matlab'de fast köşe algılamayı göstermek için bir kod yazılmış ve kameraman görüntüsünün üzerine uygulanmış olup aşağıdaki şekil görüntü üzerindeki sonucu göstermektedir.



Şekil 4.5 FAST köşe algılama

Şekil 4.5'ten görüntüde tüm köşelerin değil sadece algoritmanın koşullarını karşılayan köşelerin seçildiğini görebiliriz.

FAST köşe algılamanın önemi uygulama kolaylığından gelir, dolayısıyla çevrimiçi uygulamalar için performansı çok yüksek olacaktır.

4.3.2 BRIEF (İkili Sağlam Bağımsız Temel Öznitelikler) [19]

Öznitelik dönüştürme metodları ana ilgi noktalarını tarif etmek için son derece önemli metodlar olup en önemli metodlar SURF ve SIFT'tir. Bunlar önemli ilgi noktalarının konumunu belirleme konusunda neredeyse mükemmeldir, fakat SIFT'in tanımlayıcılar için 128-dim vektör kullanması şeklinde önemli bir probleme sahiptir. Yüzen nokta sayıları kullandığı bilinmesinden ötürü temel olarak 512 bayt kaplayacaktır. Benzer şekilde SURF de minimum 256 bayt kaplamaktadır (64-dim için). Binlerce öznitelik için çok büyük bir vektör yaratmak çok fazla hafıza kaplar ve çok fazla zaman alacağı için bu durum tümleşik sistemler için verimli değildir.

BRIEF metodu, göreceli olarak az sayıda ikili yoğunluk karşılaştırmaları alınarak görüntü yamalarının verimli bir şekilde sınıflandırılabileceğini göstermektedir. Bu testlerin sonuçları rastgele sınıflandırma ağaçlarını veya bir Naive Bayes sınıflandırıcıyı çoklu bakış açılarından görülen yamaları tanımak üzere eğitmek için kullanılmıştır. BRIEF metodunda hem sınıflandırıcı hem de ağaçlar kullanılmaktadır ve basit olarak test yanıtlarından, görüntü yamasını düzleştirdikten sonra hesaplayacağı bir vektör yaratmaktadır.

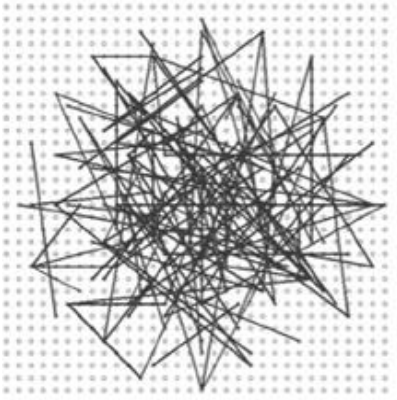
Daha spesifik olarak, test S × S boyutlarına sahip yama p üzerinde τ değerini şu şekilde tanımlamaktadır:

$$\tau(P; x, y) := \begin{cases} 1 & P(x) < P(y) \text{ ise} \\ 0 & \text{Aksi halde} \end{cases} \qquad (4.1)$$

Burada **p**(**x**), **x** = (*u, v*)'te düzleştirilmiş yama **p**'deki piksel yoğunluğudur. Bir *nd* (**x***,* **y**)-seti seçildiğinde konum çiftleri benzersiz olarak bir ikilik değer testleri setini tanımlar. BRIEF tanımlayıcının *nd*- boyutsal bit dizgisi olarak düşünülmüştür.

$$f_{nd}(P) := \sum_{1 \le i \le nd} 2^{i-1} \tau(P; x, y) \qquad (4.2)$$

Araştırmada *nd* = 128, 256 ve 512 olarak düşünülmüş olup bunun hız, depolama verimliliği ve tanıma oranı bakış açısından iyi sonuçlar vereceği görülmektedir. Bazen BRIEF tanımlayıcılar, *k* = *nd*/8 değerinin tanımlayıcıyı depolamak için gereken *bayt* sayısını temsil ettiği BRIEF-*k* şeklinde de gösterilebilir. Aşağıda yer alan şekil 4.6 BRIEF örnekleme düzenini göstermektedir.



Şekil 4.6 BRIEF örnekleme düzeni

4.3.3 ORB (Yönelimli BRIEF)

Yönelimli brief (ORB), (BREIF) tabanında bir tanımlayıcıdır. İyi bilinenleri (SIFT ve SURF) karşılaştırma konusunda son derece hızlıdır. Döndürmeden bağımsızdır ve gürültüye karşı dirençlidir. Deneyler ORB'nin (SIFT ve SURF)'ten daha hızlı iki büyüklük kertesine sahip olduğunu göstermektedir. ORB'nin temel avantajı, GPU kullanma ihtiyacı olmadan tümleşik sistemler ve cep telefonları üzerindeki gerçek zamanlı uygulama için kullanılabilmesidir

4.4 ORB Metodu

ORB'nin ana fikri, FAST köşe algılamayı bir yönelim bileşeniyle birlikte sağlamaktır. Öncelikle, 9 sayısının komşu pikseli (N=9) ifade ettiği FAST-9 şeklindeki belirli bir FAST türü ele alındığında, deney sonuçları bu değerin çok iyi performans sonuçları verdiğini göstermiştir.

Kilit noktaları seçme konusundaki temel husus, köşelerin (Nval) değeri elde edilmek istendiğinde, FAST eşik değerini (Nval) değerini aşan değerler elde edene dek düşürmemiz gerektiğidir. Ardından, alınan köşeleri sıralamak için Harris köşesi[22] ölçeği kullanılmalıdır ve sonrasında da bir eşit köşeler sayısı (Nval) seçilmelidir.

FAST çok ölçekli öznitelikler sağlamaz. Bu nedenle, ölçek piramidi için köşe algılama sağlamak amacıyla çok ölçekli alana yönelik olarak ORB için FAST gerçekleştirilmelidir. ORB, yönelim açısını hesaplamak için (yoğunluk ağırlık merkezi) kavramını kullanmaktadır. Yoğunluk ağırlık merkezi, bir köşeye ait yoğunluğun merkezinden kaymış olduğunu varsaymaktadır. Dolayısıyla bu vektör yönelimi hesaplamak için kullanılabilir. Rosin bir yamanın momentlerini şu şekilde tanımlamaktadır:

$$M_{p\,q}=\sum_{x,\,y} x^{p}\, y^{p}\, I(x, y) \qquad (4.3)$$

Ve bu momentleri kullanarak aşağıdaki şekilde ağırlık merkezleri elde edilebilir.

$$C= ((m10/m00), (m01/m00)) \qquad (4.4)$$

Yönelim açısını bulmak için köşelerin merkezinden (O) ağırlık merkezine (C) olan vektör bulunmalıdır. Yamanın OC yönelimi şu şekildedir:

ϕ = atan2 (m01, m10) (4.5)

4.4.1 R-BRIEF (Döndürmeye Duyarlı Brief)

ORB algoritmasında, – iddia ettikleri gibi – en iyi performansa sahip bir test dağılımları türü (yamanın merkezi etrafındaki Gaussian dağılımı), n = 256 vektör uzunluğu seçilerek kullanılmaktadır. Yazarlar, testi gerçekleştirmeden önce görüntüler için düzleştirme yaptıklarını söylemektedir. Düzleştirme işlemini, her bir test noktasının 31*31 piksellik bir yamaya ait 5*5 ölçüsündeki alt pencereden oluştuğu bir integral görüntüyü kullanarak yapmışlardır.

4.4.2 Yönlendirilmiş BRIEF

Bir yönlendirilmiş BRIEF anahtar noktaların yönelimine göre hesaplanabilir. Bir test konumu (x_i , y_i) öznitelik setinden (n), (2*n) matrisi şu şekilde tanımlanır.

$$S=\begin{pmatrix} x_1,\dots,x_n \\ y_1,\dots,y_n \end{pmatrix} \quad (4.6)$$

Yama yönelimi ϕ ve ilgili döndürme matrisi R_ϕ kullanıldığında, S'nin S_ϕ şeklinde yönlendirilmiş bir versiyonunu oluşturabiliriz:

$$S_\varphi = R_\varphi * S \quad (4.7)$$

Yönlendirilmiş BRIEF işleci şu şekilde olur:

$$\mathrm{gn}(\mathrm{p}, \varphi) := \mathrm{fn}(\mathrm{p}) | (\mathrm{xi}, \mathrm{yi}) \in S_\varphi \quad (4.8)$$

Bunlar açıyı 2π/30 (12 derecelik) artışlarla ayrıklaştırır ve önceden hesaplanmış BRIEF düzenlerinden oluşan bir başvuru çizelgesi oluşturur.

Örnekleme çiftlerinin sahip olması gereken iki özellik vardır. Bunlardan biri korelasyon olmamasıdır – örnekleme çiftleri korelasyonsuz olacak ve böylece her yeni çift tanımlayıcının taşıdığı bilgi miktarını maksimuma çıkarmak için tanımlayıcıya yeni

bilgiler getirecektir. Diğeri ise çiftlerin yüksek değişintisidir – yüksek değişinti bir özniteliği daha fazla ayırt edici hale getirir.

ORB yazarları, bu özelliklere sahip olduklarından emin olmak amacıyla örnekleme çiftlerini öğrenmeye yönelik bir proses ileri sürmektedir. Bir hesaplama, göz önünde bulundurulması gereken yaklaşık 205.000 olası test (örnekleme çifti) olduğunu göstermektedir. Bu test miktarından sadece 256 test seçilecektir.

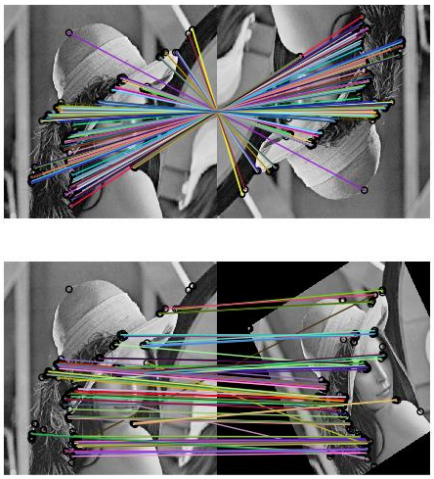
Öğrenme prosesi aşağıdaki şekildedir. Öncelikle, PASCAL 2006 veri setinden [6] çizilmiş yaklaşık 300.000 kilit noktadan oluşan bir eğitim seti oluşturulur. Sonrasında, aşağıdaki algoritma uygulanır:

1. Herhangi bir testi tüm eğitim yamalarına karşı uygulayın.
2. Testleri bir vektör T oluşturacak şekilde 0,5 ortalamasından mesafelerine göre sıralayın.
3. Greedy araması:
    1. Birinci testi sonuç vektörü R'ye getirip T'den çıkarın.
    2. T'den bir sonraki testi alıp R içerisindeki tüm testlere karşı kıyaslayın. Mutlak korelasyonu bir eşikten daha büyükse atın, değilse R'ye ekleyin.
    3. R içerisinde 256 test olana kadar önceki adımı tekrarlayın. Sayı 256'dan azsa, eşiği yükseltip tekrar deneyin.

Bu algoritma sona erdiğinde, 256 adet yüksek değişintili, göreceli olarak korelasyonsuz testten oluşan bir set elde edilir.

ORB'nin BRIEF'e benzeyen fakat fazladan döndürme bağımsızlığı ve öğrenilmiş örnekleme çiftleri avantajlarına sahip ikili değer tanımlayıcı olduğu söylenebilir. Geometrik olmayan dönüşümde (görüntü yakalamaya bağlı olan ve bulanıklık, JPEG sıkıştırma, poz ve aydınlatma gibi bakış açısına dayanmayanlar), BRIEF aslında ORB'den daha üstündür. İlgili dönüşümde, BRIEF bu gibi değişimleri ele alacak şekilde tasarlanmadığı için büyük döndürme veya ölçek değişikliği durumunda yetersiz performans gösterir. Bakış açısı değişiminden kaynaklanan perspektif dönüşümlerinde,

BRIEF şaşırtıcı bir şekilde ORB'den biraz daha üstündür. Aşağıda yer alan şekil 4.7 ORB öznitelik noktalarının bir örneğini göstermektedir:



Şekil 4.7 ORB öznitelik eşleştirme

BÖLÜM 5

SONUÇLAR

5.1 ORB Için Yüz Tanıma Sonuçları

ORB uygulamasını test etmek için C++ ve visual studio kullanılmıştır.

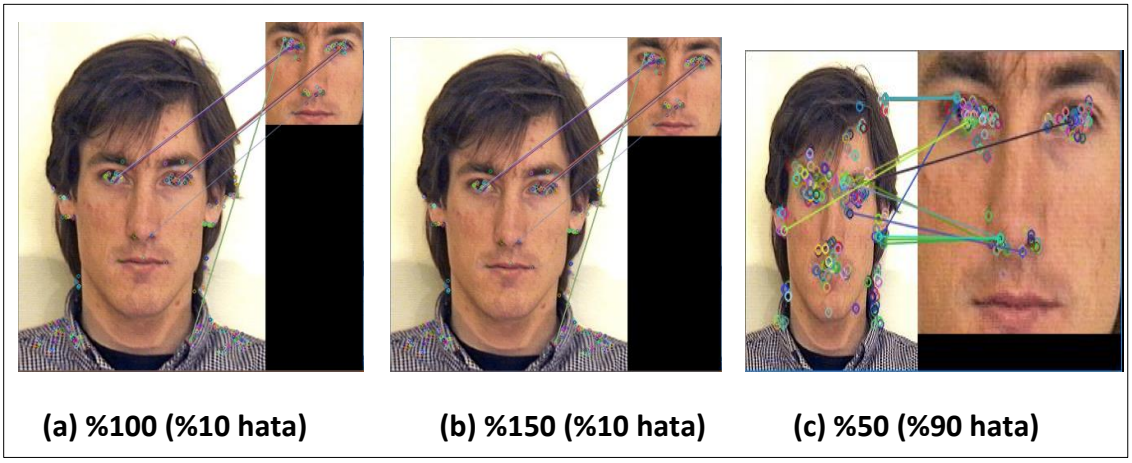
Çizelge 5.1 öznitelik noktalarının sayısı ve hesaplanması için gereken süre

Yöntem	Öznitelik sayısı	Zaman (ms)
ORB	157	8,34
LBP [21]	3.776	22,9
SIFT[2]	8.192	63,7
DFD [22]	50.176	1.511,2
CBFD%%	32.000	227,3

En güçlü 10 nokta için ORB sonuçları (ölçekleme etkileri) (adrian2 - yüz1) örnekleri Çizelge 5.2'de verilmiştir. Bu Çizelge ölçeklemenin tanıma prosesinin doğruluğu üzerindeki etkisini temsil etmektedir.

Çizelge 5.2 ölçek değişimi - hata yüzdesi

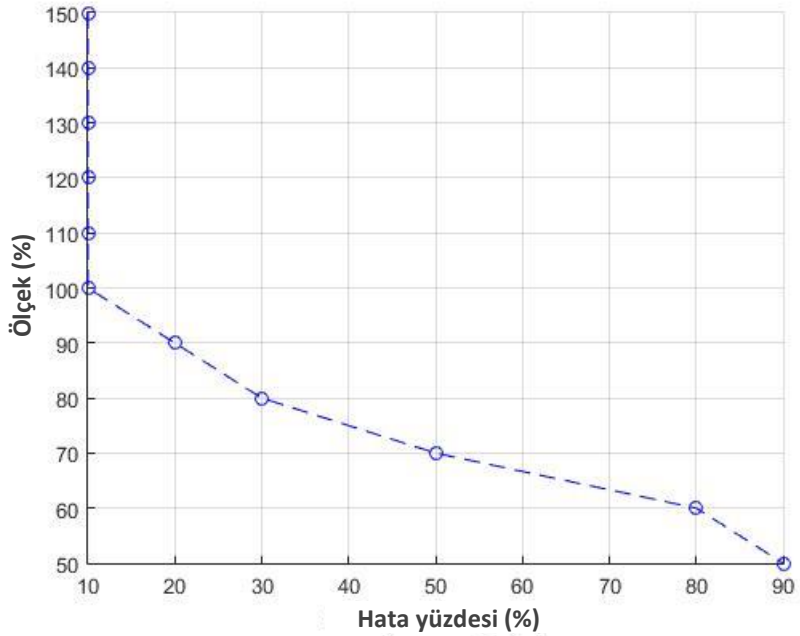
No	Ölçek(%) (adrian2)	Doğru noktalar	Yanlış noktalar	Hata yüzdesi (%)
1.	150	9	1	10
2.	140	9	1	10
3.	130	9	1	10
4.	120	9	1	10
5.	110	9	1	10
6.	100	9	1	10
7.	90	8	2	20
8.	80	7	3	30
9.	70	5	5	50
10.	60	2	8	80
11.	50	1	9	90



(a) %100 (%10 hata) **(b) %150 (%10 hata)** **(c) %50 (%90 hata)**

Şekil 5.1 Ölçeğin noktaların doğruluğu üzerindeki etkisi

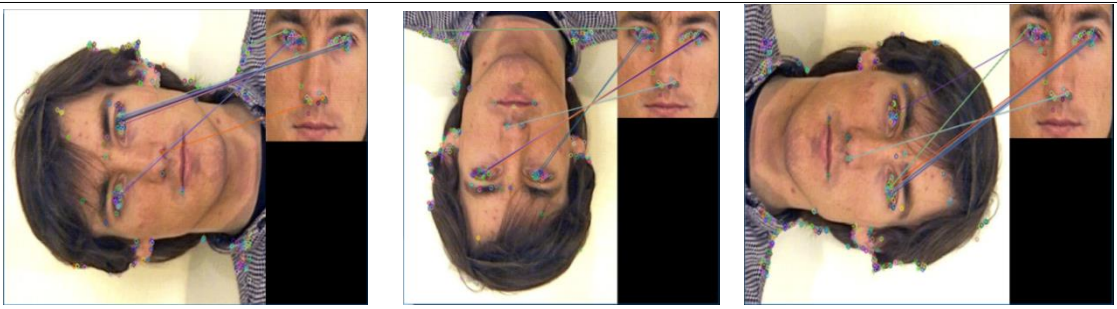
Şekil 5.1 tanıma doğruluğunu ve yanlış noktaların sayısını temsil etmektedir. Ölçek %100'den daha küçük olursa doğruluğun azaldığını ifade edilebilir.



Şekil 5.2 Ölçek değişimi - hata yüzdesi

Çizelge 5.3 en güçlü 10 nokta için ORB sonuçları (döndürme etkileri) örnekleri (adrian1 - yüz1)

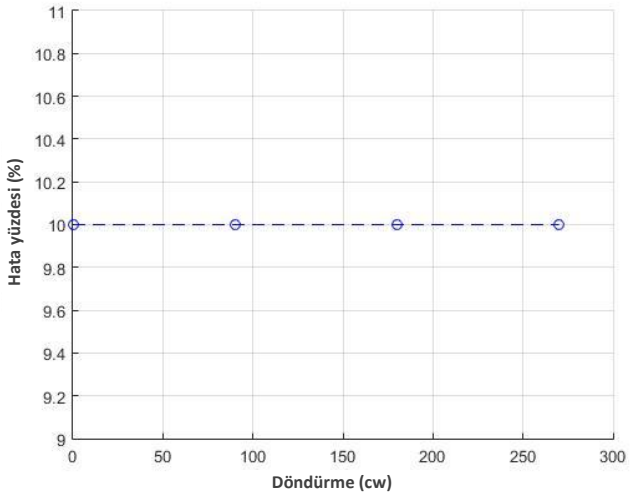
No	Açı(%) (adrian1)	Doğru noktalar	Yanlış noktalar	Hata yüzdesi (%)
1.	0	9	1	10
2.	90 (saat yönünde)	9	1	10
3.	180 (saat yönünde)	9	1	10
4.	270 (saat yönünde)	9	1	10



90 derece c.w **180 derece c.w** **270 derece c.w**

Şekil 5.3 Döndürmenin öznitelik noktaları üzerindeki etkileri

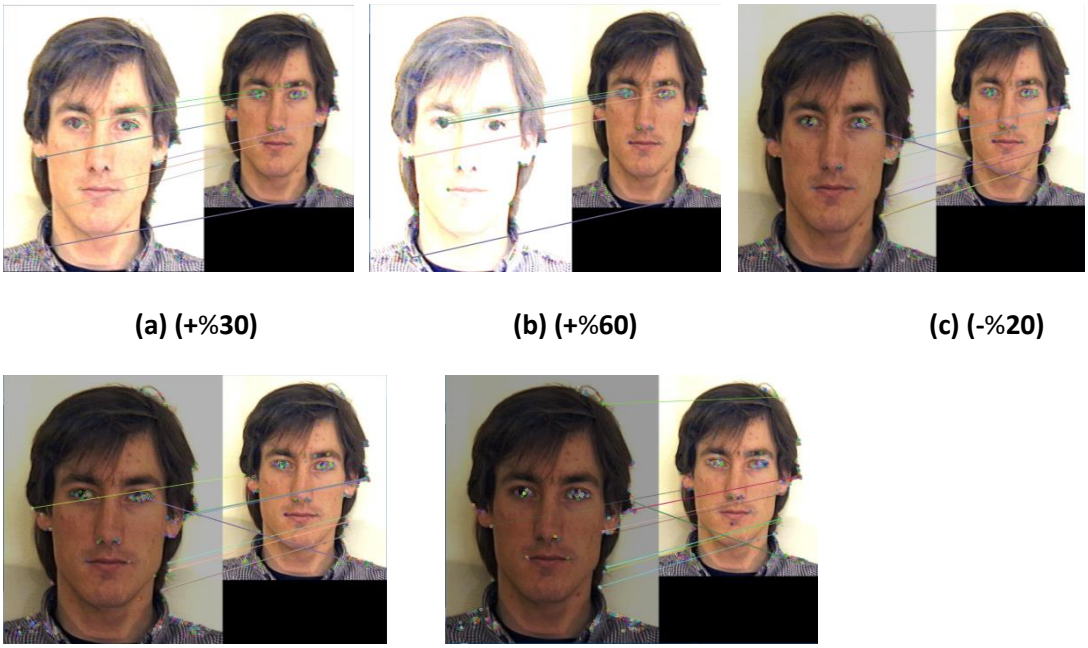
Yukarıdaki Çizelge ve şekil döndürmenin tanıma doğruluğu üzerindeki etkisini temsil etmektedir. Döndürmenin tanıma prosedürünü etkilemediği sonucuna varılabilir. Bu da ORB'nin ölçekten bağımsız olduğu anlamına gelir.



Şekil 5.4 Döndürmenin hata yüzdesi üzerindeki etkileri

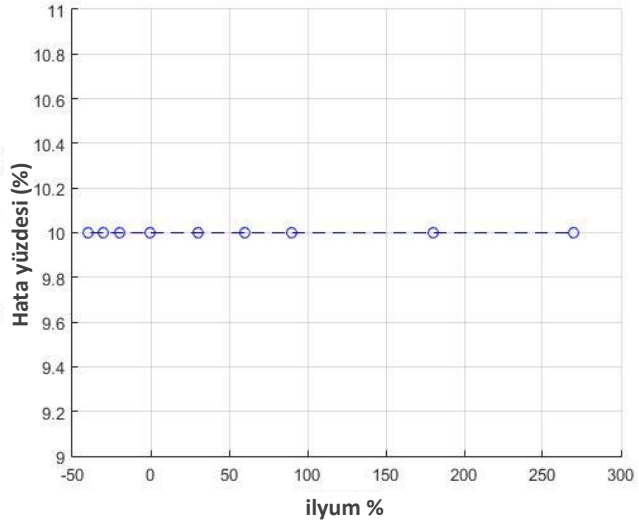
Çizelge 5.4 en güçlü 10 nokta için ORB sonuçları (aydınlatma etkileri)

No	Aydınlatma(%) (adrian1)	Doğru noktalar	Yanlış noktalar	Hata yüzdesi (%)
1.	60	9	1	10
2.	30	9	1	10
3.	0	9	1	10
4.	-20	9	1	10
5.	-30	9	1	10
6.	-40	9	1	10



Şekil 5.5 Aydınlatmanın öznitelik noktaları üzerindeki etkileri

Yukarıda yer alan Çizelge 5.4 ve Şekil 5.5 aydınlatmanın tanıma doğruluğu üzerindeki etkisini temsil etmektedir. Aydınlatmanın tanıma doğruluğunu etkilemediği ve bunun da ORB'nin aydınlatma farklarından bağımsız anlamına geldiği sonucuna varılabilir.



Şekil 5.6 Aydınlatma - hata yüzdesi

Çizelge 5.5 En güçlü 10 nokta eşik etkisi için ORB sonuçları örnekleri (adrian1 - diğerleri)

No	adrian1 - adrian1 mesafe	adrian1 - adrian2 mesafe	adrian1 - adrian3 mesafe	adrian1 - adrian4 mesafe	adrian1 - alec1 mesafe	adrian1 - alison1 mesafe	adrian1 - alison2 mesafe	adrian1 - alister1 mesafe	adrian1 - alister2 mesafe	adrian1 - alister3 mesafe
1.	0	8	12	17	32	24	21	34	34	33
2.	0	9	13	28	33	26	23	35	34	35
3.	0	10	17	29	36	27	24	36	34	35
4.	0	11	17	29	37	28	29	36	34	35
5.	0	12	18	30	37	29	33	36	35	35
6.	0	13	18	31	37	29	33	39	36	35
7.	0	13	18	31	38	29	34	39	37	36
8.	0	13	18	32	38	30	34	39	37	37
9.	0	14	19	32	39	31	35	39	38	37
10.	0	14	19	32	39	32	35	39	38	38
11.	0	14	20	32	39	32	35	39	39	38

Yukarıda yer alan Çizelge 5.5 aynı yüz ve diğer yüzler için ORB mesafeleri değerlerini temsil etmektedir. Yüzler arasındaki farklar ilk mesafeden görülebilir.

5.2 Sonuç

Bu tezde yüz algılamasına yönelik yeni bir yaklaşım oluşturulmuştur. PADL adı verilen bu yaklaşımın temel avantajı LBP'ye ait özellik vektörünü uzun değerden çok kısa bir değere düşürmektir. Bu ise algılama amacıyla bir görüntü sunumunun kısa bir sayıya dönüştürüldüğü anlamına gelir.

Bu değişiklik, sınıflandırma sürecini çok daha kolay ve daha az zaman alıcı hale getirmiştir, diğer katkı ise çok basit bir şekilde yüzleri, yuvarlak yüzleri veya köşeli yüzleri bulmak için LBP dokusu kullanılarak yapılır.

Değişiklik süreci, daha önce açıklandığı şekilde histogram genlik dağıtım seviyeleri tekniği bu tezde sunulan yeni bir teknik kullanılarak gerçekleştirilir.

Yüz tanımasında, birçok durumda yüzü tanımak için yüzlerdeki özellik noktalarını seçmek için ORB algoritması kullanılmıştır, tanıma sürecinde ölçekleme, döndürme ve aydınlatmanın etkinliği araştırılmış ve gerçek zamanlı simülasyona uygulanmıştır.

KAYNAKLAR

[1] P. Viola, M. Jones., (2001). “Rapid object detection using a boosted cascade of simple features”, Proc. of CVPR, 511-518.

[2] T. Ojala, M. Pietikäinen, ve D. Harwood., (1994). “Performance evaluation of texture measures with classification based on Kullback discrimination of distributions”, 12th IAPR International Conference on Pattern Recognition (ICPR 1994), 1: 582 - 585.

[3] D. G. Lowe., (2004). “Distinctive image features from scale-invariant keypoints“, International Journal of Computer Vision, 60(2): 91–110.

[4] H. Bay , T. Tuytelaars, ve L. Van Gool., (2006). “Surf : Speeded up robust features” , European Conference on Computer Vision,1, 2.

[5] N. Dalal ve B. Triggs., (2005). “Histograms of Oriented Gradients for Human Detection”, IEEE Conf. Computer Vision and PatternRecognition (CVPR'05) ,01: 886-893.

[6] ST. Andrews, I. Tsochantaridis, ve T. Hofmann., (2003). “Support Vector Machinesfor Multiple-Instance Learning”, Advances in Neural Information Processing Systems, 15:561-568.

[7] Yoav Freund., (1999). “ An adaptive version of the boost by majority algorithm”, Proceedings of the Twelfth Annual Conference on Computational Learning Theory,102-113.

[8] H. A. Rowley, S. Baluja, T. Kanade., (1996). “Neural network-based face detection”, IEEE Transactions on pattern analysis and machine intelligence, 20:23-38.

[9] D. Roth, M.-H. Yang, N. Ahuja., (2000). “A SNoW-based face detector”, Proc. of NIPS,24.

[10] R. Feraud, O. J. Bernier, J.-E. Viallet, M. Collobert., (2001). “A fast and accurate face detector based on neural networks”, IEEE Trans.on PAMI,23 (1): 42-53.

[11] C. Garcia, M. Delakis., (2004). “Convolutional face finder: A neural architecture for fast and robust face detection”, IEEE Trans. on PAMI,26:1408–1423.

[12] M. Osadchy, Y. L. Cun, M. L. Miller., (2007). " Synergistic face detection and pose estimation with energy-based models", The Journal of Machine Learning Research 8, 1197–1215.

[13] D. Keren, M. Osadchy, C. Gotsman., (2001). "Antifaces: A novel fast method for image detection", IEEE Trans. on PAMI,23:747–761.

[14] C. Liu., (2003). "A bayesian discriminating features method for face detection", IEEE Trans. on PAMI,25:725–740.

[15] N. Cristianini, J. Shawe-Taylor., (2000). "An Introduction to Support Vector Machines and other Kernel-Based Learning Methods", Cambridge University Press.

[16] Xiaoyu Wang, Tony X. Han, Shuicheng Yan., (2009). " An HOG-LBP Detector with Partial Occlsion Handling", IEEE 12th International Conf. on Computer Vision.

[17] C. Silva, T. Bouwmans, C. Frelicot., (2015). "An extended Center-Symmetric Local Binary Pattern for Background Modeling and Subtraction in Videos", VISAPP Berlin, Germany.

[18] R. Ethan, R. Vincent, K. Kurt, B. Gary., (2011). "ORB: an efficient alternative to SIFT or SURF", in Proc.IEEE International Conference on Computer Vision (ICCV),2–5.

[19] M. Calonder, V. Lepetit, C. Strecha, ve P. Fua. ÖZET., (2010). "Binary Robust Independent Elementary Features", The European Conference on Computer Vision (ECCV), 1, 2, 3, 5.

[20] E. Rosten, T. Drummond., (2006). "Machine learning for high-speed corner detection", European conference on computer vision, 430-443.

[21] MacQueen, J.B., (1967). "Some Methods for classification and Analysis of Multivariate Observations. Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability", University of California Press.281–297.

[22] Harris ve M. Stephens., (1988). "A combined corner and edge detector", The 4th Alvey Vision Conference, 147–151.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı - Soyadı	: Wisam ALOBAIDI
Doğum tarihi ve yeri	:Musul/IRAK, 21.2.1975
Yabancı Diller	: İngilizce/Türkçe
E-posta	: wisam_en@yahoo.com

ÖĞRENİM DURUMU

Derece	**Bölüm**	**Üniversite**	**Mezuniyet yılı**
Lisans	Elektronik ve Haberleşme Mühendisliği Bölümü	Musul Üniversitesi	2004
Lise	Fen Bilimleri	Alresala Lisesi	1994

İŞ DENEYİMİ

Yıl	**Kurum/Kuruluş**	**Kayıt**
2005-2013	IRAK/Elektrik Bakanlığı	Trafo Merkezi Mühendisi

YAYINLAR

Konferans Belgeleri

1. Alobaidi, W.,(2016), “Face Recognition based on ORB Feature Extraction” Istanbul Turkey October 13-14, ISBN No. 978-969-670-955-5. International Conference on “Engineering and Technology, Computer, Basic and Applied Sciences”- ECBA-2016.