

**KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

BİLGİSAYAR İLE PSİKOLOJİK DURUM DEĞERLENDİRİLMESİ

YÜKSEK LİSANS TEZİ

Bilgisayar Mühendisi Burçin KURT

**TEMMUZ 2007
TRABZON**

**KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

BİLGİSAYAR İLE PSİKOLOJİK DURUM DEĞERLENDİRİLMESİ

Bilgisayar Mühendisi Burçin KURT

**Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsünde
"Bilgisayar Yüksek Mühendisi"
Unvanı Verilmesi İçin Kabul Edilen Tezdir.**

**Tezin Enstitüye Verildiği Tarih : 18.06.2007
Tezin Savunma Tarihi : 05.07.2007**

**Tez Danışmanı : Prof. Dr. Vasif V. NABIYEV
Jüri Üyesi : Yrd. Doç. Dr. Hüseyin PEHLİVAN
Jüri Üyesi : Yrd. Doç. Dr. Ali GANGAL**

Enstitü Müdürü :Prof. Dr. Emin Zeki BAŞKENT

Trabzon 2007

ÖNSÖZ

Bilgisayar ile psikolojik durum değerlendirilmesi, yüz ifadelerinin tanınması üzerine dayanmaktadır. Bu çalışmada, yüz ifadelerinin tanınması ve değerlendirilmesi araştırılmıştır.

Çalışmam süresince bilgi, görüş ve önerileriyle bana yol gösteren, her zaman desteğini gördüğüm çok değerli danışman hocam sayın Prof. Dr. Vasıf V. NABİYEV' e teşekkürü bir borç bilir şükranlarımı sunarım.

Tez çalışmam süresince sabır, destek ve sevgileriyle hep yanımda olan aileme teşekkür eder saygılarımı sunarım.

Çalışmamda, yüz ifadelerine ait veritabanının oluşturulmasında katkıları olan herkese teşekkür ederim.

Programın doğruluğunun test edilmesi amacıyla insanlar tarafından yapılan psikolojik durum değerlendirme deneysel sonuçlarının oluşturulmasında katkıları olan Bilgisayar Mühendisliği bölümü 2.sınıf öğrencilerine ve herkese teşekkür ederim.

Burçin KURT
Trabzon, 2007

İÇİNDEKİLER

Sayfa No

ÖNSÖZ.....	II
İÇİNDEKİLER.....	III
ÖZET.....	V
ABSTRACT.....	VI
ŞEKİLLER DİZİNİ.....	VII
TABLolar DİZİNİ.....	IX
1. GENEL BİLGİLER.....	1
1.1 Giriş.....	1
1.2. Yüze Ait Özellikler.....	2
1.3. Yüz Tespiti Metotları (Face Detection Methods).....	4
1.3.1. Özellik tabanlı metotlar (Feature-based methods).....	5
1.3.2. Görüntü tabanlı metotlar (Image-based methods).....	6
1.3.2.1. Ten Bulma Yöntemi (Skin Detection Method).....	7
1.3.3. Yapay Sinir Ağları (YSA).....	9
1.3.3.1. Yapay Sinir Ağlarının Özellikleri.....	10
1.3.3.2. YSA'nın Uygulama Alanları.....	10
1.3.3.3. Ağın Yapısı ve Eğitimi.....	11
1.3.3.4. Çerçeve Birleştirme (Box Merge).....	16
1.4. Psikolojik Durum Değerlendirilmesi.....	16
1.4.1. Özellik Çıkarma.....	19
1.4.1.1. Optik Akış Hesaplaması (Optical Flow Computation).....	20
1.4.1.2. Gabor Filtreleri (Gabor Filters).....	21
1.4.1.3. Yerel İkili Örnekler (Local Binary Patterns – LBP).....	21
1.4.1.3.1. LBP Operatörleri.....	22
1.4.1.3.2. Temel LBP.....	22
1.4.1.3.3. LBP Alt Kümeleri.....	23
1.4.1.3.3.1. Tek Yönde Arama (Beam Search).....	23
1.4.1.3.3.2. Düzenli Örnekler (Uniform Patterns).....	24

1.4.2.	Örnek Tanıma.....	26
1.4.2.1.	K- Ortalama Algoritması ile Kümeleme (K-Means Algorithm).....	28
1.4.2.2.	Minimum Uzaklık (Minimum Distance).....	29
1.4.2.3.	Destek Vektör Makineleri (Support Vector Machines – SVMs).....	29
1.4.2.4.	Uyarlanabilir Yükseltme (Adaptive Boost- AdaBoost).....	31
1.4.2.5.	Model Eşleme (Template Matching).....	32
1.4.2.5.1.	Chi Karesi İstatistik (Chi Square Statistics).....	32
1.4.2.5.2.	Ağırlıklandırılmış Chi Karesi İstatistiği (Weighted Chi Square Statistics)	32
1.4.2.6.	Yapay Sinir Ağları.....	33
2.	YAPILAN ÇALIŞMALAR, BULGULAR VE İRDELEME.....	34
2.1	Sistem Gereksinimleri.....	34
2.2.	Sistemin Yapısı ve Aşamaları.....	35
2.2.1.	Yüz Tespiti.....	35
2.2.1.1.	Ten Bulma.....	36
2.2.1.2.	YSA.....	41
2.2.1.3.	Ölçekleme.....	44
2.2.1.4.	Yüz Bulgularını Tek Bulguya İndirgeme.....	46
2.2.1.5.	Yüz Tespitine İlişkin Sonuçlar	48
2.2.2.	Yüz İfadesi Tanıma	51
2.2.1.	LBP ile Özellik Çıkarma	52
2.2.2.	Sınıflayıcılar ile Örnek Tanıma.....	57
2.2.2.1.	Minimum Uzaklık ile Sınıflama.....	59
2.2.2.2.	Değiştirilmiş Minimum Uzaklık ile Sınıflama.....	59
2.2.2.3.	Ağırlıklandırılmış Chi Karesi İstatistiği ile Sınıflama	60
2.2.2.4.	Yapay Sinir Ağları ile Sınıflama	62
3.	SONUÇLAR VE ÖNERİLER.....	66
4.	KAYNAKLAR.....	75
5.	EKLER.....	79
	ÖZGEÇMİŞ.....	81

ÖZET

Psikolojik durum değerlendirilmesi, yüz ifadelerinin tanınmasına dayalı bir çalışmadır. Yüz ifadelerini tanıma becerisi bakımından insanlar arasında bile bireysel ayrılıklar vardır. Bu nedenle yüz ifadelerini tanıma becerisi ölçülebilecek bir değişkendir. Çalışmada, psikolojik durumun bilgisayarla değerlendirilmesi gerçekleştirilmiştir. Bu amaçla, öncelikle deri analizi ve Yapay Sinir Ağları kullanılarak yüz tespiti gerçekleştirilmiştir. İkinci aşamada, yüz örneği üzerinde özellikler çıkarılarak sınıflandırma sonucunda tanıma işlemi gerçekleştirilmiştir. Yüz resimleri üzerinde özellik çıkarma işlemi için Yerel İkili Örnekler(LBP-Local Binary Patterns) yaklaşımı kullanılmış, bu şekilde yüze ait çok küçük önemli örnekler betimlenmiştir. Daha sonra çıkarılan özellikler üzerinde sınıflama gerçekleştirilerek tanıma işlemi tamamlanmıştır. Özelliklerin sınıflanmasında model eşleme tabanlı sınıflayıcılar ve yapay sinir ağları kullanılmıştır.

Çalışmada *Gülen, Doğal, Üzgün, Şaşkın ve Kızgın* olmak üzere 5 farklı yüz ifadesi incelenmiş, ayrıca gülme olgusu tam gülen ve gülümseyen olmak üzere iki alt sınıfa parçalanarak tanınmaya çalışılmıştır.

Sistemin başarısının ölçülmesi amacıyla insanlar tarafından yapılan psikolojik değerlendirme veritabanı oluşturulmuştur ve bu deneysel sonuçlardan, sistemin insana benzer sonuçlar verdiği görülmüştür.

Anahtar Kelimeler: Ten Bulma, Yapay Sinir Ağları, Yüz Tespiti, Özellik Çıkarma, Örnek Tanıma, Yüz İfadesi Tanıma

ABSTRACT

PSYCHOLOGICAL CASE EVALUATION

Psychological case evaluation is a study depending on facial expression recognition. On account of a ability of facial expression recognition , there are individual differences even in between people. Therefore, the ability of facial expression recognition is a measurable variable. In this study, psychological case evaluation with computer has been verified. With this aim, first of all , face detection has been implemented by using skin detection and artificial neural networks. On the second stage face sample, features had been extracted and then as a result of features classification, recognition process has been implemented. For feature extraction on detected faces, Local Binary Patterns(LBP) approach has been used and in this way, major micro patterns belonging to face have been described. After that, classification has been implemented on the extracted features and thus, recognition process has been completed. For classifying features, template matching based classifiers and artificial neural networks have been used.

In the study, five different facial expressions have been observed which are *Smile*, *Neutral*, *Sad*, *Surprised* and *Angry* and also the smile fact has been tried to be recognized by separating into two sub classes which are laughing and smiling.

To confirm system truth, a pscychological case evaluation database has been formed which was evaluated by people and from these experimental results, similarity between systems results and people has been seen.

Key Words: Skin Detection, Artificial Neural Networks, Face Detection, Feature Extraction, Pattern Recognition, Facial Expression Recognition

ŞEKİLLER DİZİNİ

Sayfa No

Şekil 1.	Kadın ve erkeklerde temel yüz oranları	3
Şekil 2.	Mona Lisa' ya ait yüz ve beden oranları	3
Şekil 3.	Matlab ile üç boyutlu yüz gösterimi.....	4
Şekil 4.	Yüz tespit metotları	5
Şekil 5.	(a)Üç boyutlu YCbCr renk uzayı, (b) Üç boyutlu HSV renk uzayı.....	7
Şekil 6.	YCbCr renk uzayındaki ten piksellerinin dağılımı	9
Şekil 7.	Çok katmanlı ağ yapısı.....	12
Şekil 8.	Dudak bölgeler	20
Şekil 9.	Hareket istatistiklerinin gösterimi	20
Şekil 10.	Gabor filtreleri.....	21
Şekil 11.	LBP operatörleri	22
Şekil 12.	3x3 komşuluklu LBP kodu hesaplaması	22
Şekil 13.	Kümeleme	27
Şekil 14.	Sınıflayıcının gelişimi	27
Şekil 15.	Minimum uzaklık sınıflaması	29
Şekil 16.	Ana düşünceye ait örnek	30
Şekil 17.	Chi karesi ile doğru ve yanlış tanıma oranları	32
Şekil 18.	Sistemin genel yapısı.....	35
Şekil 19.	RGB-HSV ve RGB-YCbCr dönüşüm sonuçları ve histogramları.....	36
Şekil 20.	Resme ait 3 boyutlu HSV ve YCbCr renk uzayları görüntüsü	36
Şekil 21.	Ten bulma işlem adımları.....	37
Şekil 22.	Ten bulma arayüzü	39
Şekil 23.	Farklı faktör değerlerine ait ten bulma sonuçları	39
Şekil 24.	Ten bulunmuş resimler.....	40
Şekil 25.	Eğitim aşaması akış şeması	41
Şekil 26.	YSA eğitim çıktısı.....	43
Şekil 27.	Yüz veritabanına ait bazı örnekler	44
Şekil 28.	Yüz olmayan veritabanına ait bazı örnekler.....	44
Şekil 29.	Ölçekleme sonuçları.....	46

Şekil 30.	Çerçeveleme sonuçları	47
Şekil 31.	Yöntemlerin karşılaştırılması	48
Şekil 32.	Yüz tespiti arayüzü.....	50
Şekil 33.	Yüz Tespiti Sonuçları.....	50
Şekil 34.	LBP'nin genel yapısı.....	52
Şekil 35.	Yüz resmine ait H_i histogramı	54
Şekil 36.	(a) Yüz , (b) Yüz olmayan resimlere ait H_i histogramları	55
Şekil 37.	Bölünmüş resim ve yerel bölgelere ait LBP histogramları	56
Şekil 38.	(a) Gülen , (b) Üzgün, (c) Şaşırmış, (d) Kızgın, (e) Doğal , modeli	57
Şekil 39.	(a) Tam gülen, (b) Gülümseyen, modeli	58
Şekil 40.	Yüz ifadelerine ait veritabanı	58
Şekil 41.	Minimum uzaklık ile yüz ifadesi tanıma.....	59
Şekil 42.	Değiştirilmiş minimum uzaklık ile yüz ifadesi tanıma	60
Şekil 43.	Ağırlıklandırılmış chi karesi modeli	60
Şekil 44.	Chi karesi istatistiği ile yüz ifadesi tanıma	62
Şekil 45.	YSA ile sınıflama diyagramı.....	63
Şekil 46.	Yüz ifadesi eğitim arayüzü.....	64
Şekil 47.	YSA ile yüz ifadesi sorgulama arayüzü	64
Şekil 48.	Yüz ifadesi tanıma arayüzü	65
Şekil 49.	Kişi psikolojik değerlendirme veri girişi arayüzü.....	66
Şekil 50.	Sorgulama arayüzü.....	67
Şekil 51.	Resme ait veritabanının görüntülenmesi.....	67
Şekil 52.	Program ve insanlara ait değerlendirme sonuçları	68
Şekil 53.	(a) Ten bulma + YSA ile yüz tespiti, (b) YSA ile yüz tespiti	70
Şekil 54.	Yüz tespitine ilişkin eğitilen örnek sayısına göre başarı yüzdesi.....	71
Şekil 55.	Yüz tespiti için koşma süreleri	72
Şekil 56.	Yüz ifadesi tanıma için koşma süreleri	72
Şekil 57.	Kızma ifadesine ait sonuçlar	74
Ek Şekil 1.	Psikolojik durum değerlendirme deneysel sonuçlarına ait veritabanı.....	79

TABLULAR DİZİNİ

Sayfa No

Tablo1. Geometrik tabanlı TAN ve kullanılan LBP tabanlı model eşleme metotlarının karşılaştırılması.....	69
Tablo 2. Yüz ifadesi tanıma metotlarının karşılaştırılması.....	69
Tablo 3. Kullanılan metot ile diğer güçlü metotların karşılaştırılması.....	70
Tablo 4. Yüz ifadelerine ait tanıma başarıları.....	74

1. GENEL BİLGİLER

1.1. Giriş

İnsanlar arası iletişim, kişilerin birbirlerine bilinçli yada bilinçsiz olarak iletmek istedikleri duygu ve düşüncelerini aktardıkları bir süreçtir. İletişim sırasında aktarılan mesaj bireyin psikososyal yaşantılarından oluşur. Yüzdeki ifade, el ve vücut hareketleri vücudun duruşu ve göz teması sözsüz iletişimde önemli yer tutar. Depresyon geçiren bir hastanın yüzünün kederli olacağı düşünülür. Oysa yüz bölüm bölüm incelendiğinde depresyon durumundaki kişinin yüz ifadesinin öfke ve korkuya işaret ettiği (frenlenmiş , eyleme dönüştürülemeyen öfke) ispatlanmıştır[1]. İnsanlar mimikleri sayesinde o anki duygularını kolaylıkla ifade edebilirler. Bu durumun nedeni, yüz kaslarının iki yüz elli bin değişik şekle sokulabilmesidir. Kare kare incelenen filmler ve kasetler; acıma, endişe, kıskançlık, kin, korku, öfke gibi elem duygularını belirten mimiklerin birbirini izleyen, birbiri ile bütünleşen ağlama mimikleri olduğunu ortaya koymuştur. Neşe, sevinç, umut gibi haz doğrultusundaki duyguları anlatan mimiklerin de gülümseme mimiklerinin birbiri ardına dizilmesinden oluştuğu anlaşılmıştır. Kısaca, ağlama ve gülme mimikleri temel olup, öteki duygu durumlarını belirten mimikler bu temel mimiklerden oluşur[2].

Altı temel yüz ifadesi vardır. Bu ifadeler, şaşırma, korku, tikslenme, öfke, mutluluk ve üzüntüdür. Bu duygular çeşitli bileşimler halinde de görülebilir. Örneğin, şaşırma ve korku, öfke ve tiksinti, mutluluk ve hüznün aynı anda yaşanabilir ve yüze yansiyabilir[3].

Tüm bunlar göz önüne alındığında ve ayrıca yüz ifadelerinin tanınmasının insanlar tarafından bile bazen kolay olmadığı düşünüldüğünde bilgisayar ile bu işlemin gerçekleşmesinin ne derecede zor olduğu anlaşılmaktadır.

1950 yıllarının sonlarında dama, satranç ve diğer Yapay Zeka problemleri bilgisayarlarda programlandıktan sonra çok yakında “düşünen makinelerin” yapılacağı tahmin ediliyordu. Örneğin, *H.Simon* ve *A. Newell*, 1958 yılında “*on yıl içerisinde bir çok psikolojik teorisinin bilgisayar programları biçiminde ifade edileceğini*” iddia etmekteydi. Fakat 50 yıla yakın geçen süre sonunda bu tahminlerin abartılı olduğu anlaşılmıştır[4]. Son 10 yılda psikolojik değerlendirmelerde bilgisayarlı uygulamalara yoğun bir şekilde yer verilmektedir.

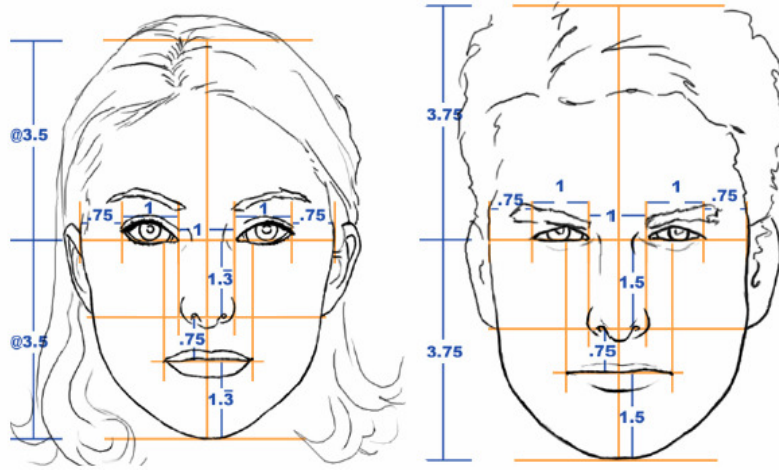
Çalışmada psikolojik durumlardan olan çeşitli yüz ifadelerinin tanınması üzerinde durulmaktadır. Bunun için, çalışma iki aşamadan oluşur. İlk aşamada yüz tespit işlemi gerçekleştirilir. Burada ten işleme ile iyileştirilmiş yapay sinir ağları yaklaşımı kullanılmıştır. İkinci aşamada ise yüz ifadesinin tanınması gerçekleşir. Yüz ifadesinin tanınması iki aşamada gerçekleşmektedir. Bunlar, özellik çıkarma ve sınıflama işlemleri olarak tanımlanır. Özellik çıkarmada iki ana yaklaşım vardır; geometrik özelliğe dayalı metotlar ve görünüşe dayalı metotlar[5]. Gabor dalgacıkları görünüş özelliklerinin, geometrik özelliklerden daha etkili olduğu kanıtlanmıştır[6], ve gerçekte daha iyi çalıştığı görülmüştür[7]. Yaygın olarak Gabor dalgacıkları betimlemesi tercih edilmesine rağmen, yüz resimlerini Gabor filtreleri ile evriştirmek hesaplama açısından oldukça masraflıdır. Bu konularda birçok yaklaşım geliştirilmiştir. Shinohara ve Otsu, özellik vektörünü oluşturmak için Yüksek Düzenli Yerel Oto Korelasyon(HLAC-Higher-Order Local Auto-Correlation) özellikleri ile Fisher ağırlık haritasını (Fisher weight map) birleştirmişler [8]. Lyons ve arkadaşları 2D Gabor dalgacıkları (Gabor wavelets) betimlemesi ve lineer fark sınıflayıcısı (LDC - linear discriminant classifier) şeması kullanmıştır [9]. Zhang vb. yüz betimlemede geometrik pozisyon ile Gabor dalgacıklarının birleşimini, yüz ifadesi tanımada ise çok katmanlı perseptron kullanmaktadır [6], [10]. Çalışmada, diğer bir görünüşe dayalı özellik çıkarma metodu olan Yerel İkili Örnekler (LBP-Local Binary Patterns) kullanılmıştır. LBP özellikleri hızlı bir şekilde çıkarılabilir ve ayrıca yüze ait çok küçük örneklerin betimlenmesinde oldukça etkilidir. Bununla birlikte, diğer yöntemler ile karşılaştırıldığında LBP hesaplaması daha az karmaşık olduğundan daha hızlı hesaplanmaktadır ve çok boyutlu uzayda daha az yer kaplamaktadır. Çıkarılan özelliklerin sınıflanmasında model eşleme tabanlı (template matching) metotlar ve yapay sinir ağları kullanılmıştır.

1.2. Yüze Ait Özellikler

Bir bireyin ölçülebilen fizyolojik ya da davranışsal özelliklerine biyometrik (biometric) özellikler denilmektedir[11]. Biyometrik özelliklerden olan yüz ve ses, tanıma sistemlerinde kolaylıkla uygulanabilmesine rağmen, çevresel faktörlerden de güçlü bir şekilde etkilenmektedir.

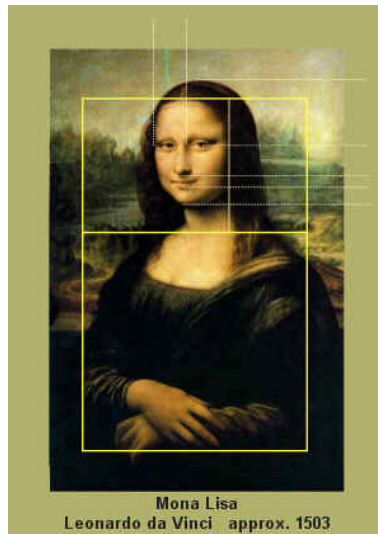
Beden oranlarında olduğu gibi insan yüzünde de genel olarak doğru kabul edilebilecek belirli oranlar vardır. İnsan yüzünde yer alan bazı altın oranlar şöyledir:

Yüzün boyu / Yüzün genişliği,
 Dudak-kaşların birleşim yeri arası / Burun boyu,
 Yüzün boyu / Çene ucu-kaşların birleşim yeri arası,
 Ağız boyu / Burun genişliği,
 Burun genişliği / Burun delikleri arası
 Göz bebekleri arası / Kaşlar arası



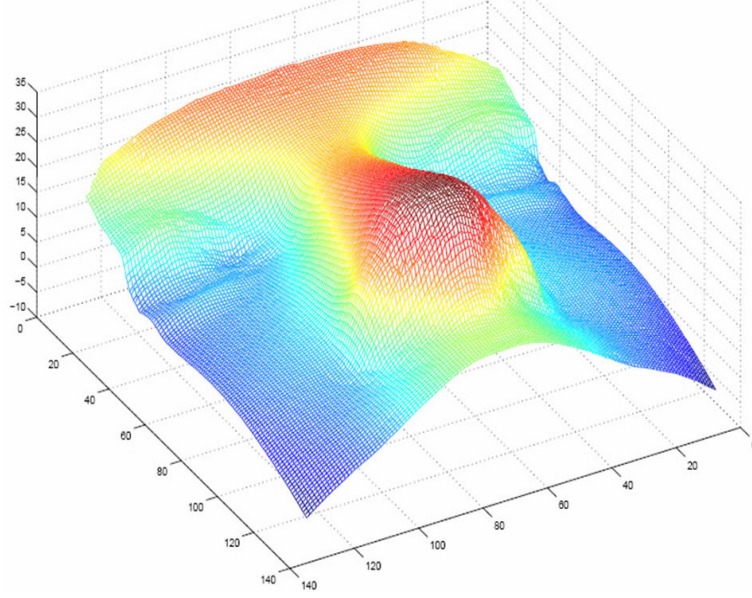
Şekil 1. Kadın ve erkeklerde temel yüz oranları

Verilen oranlar her bireyde farklılık gösterse de genelleştirilebilecek bir ölçüde geçerlidir. Morfolojik bilgiler kullanılarak oluşturulan *yapay yüzler* bu ölçülere uymaktadır. Sanatta da yine yüze ait oranlar kullanılmaktadır.



Şekil 2. Mona Lisa'ya ait yüz ve beden oranları

Tanıma işlemlerinde yüze ait biyometrik özellikler sıkça kullanılmaktadır. Yüzü üç boyutlu uzayda incelersek, tüm dağları ve çukurları daha iyi gözlemleyebiliriz.



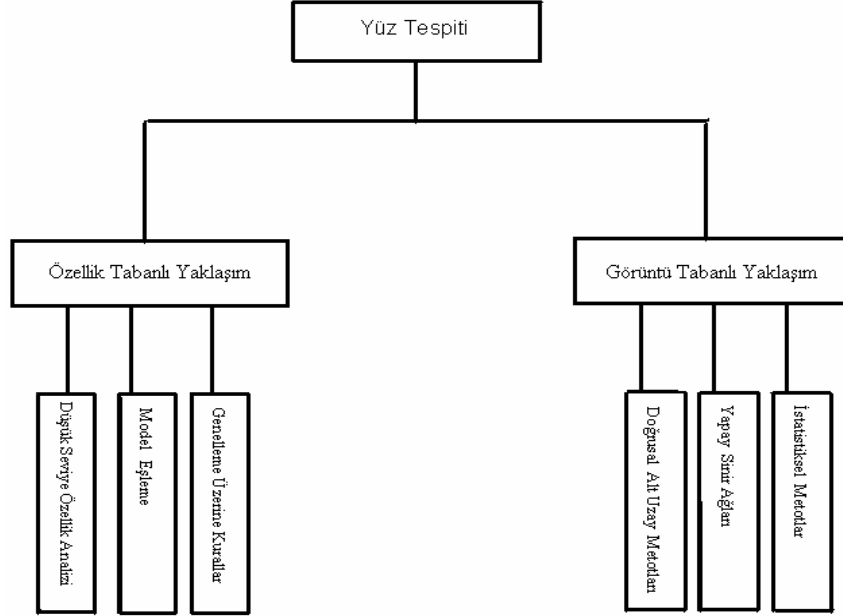
Şekil 3. Matlab ile üç boyutlu yüz gösterimi

Şekil 3'de de görüldüğü gibi yüz bölgesinde en çok öne çıkan kısım burun kısmıdır, daha sonra alın kısmı gelir. En çukur bölgeler ise göz bölgeleridir. Psikolojik durum değerlendirilmesinde yüz ifadesine ait en az öneme sahip bilgiyi burun bölgesi tutmaktadır.

1.3. Yüz Tespiti Metotları (Face Detection Methods)

Yüz bulma, yapay görünümün zorlu problemlerinden biridir. Yüz bulma probleminin amacı, bir tek veya ardışık görüntü verildiğinde, yüz içeren bölgeleri 3 boyutlu transformasyona ve ışıklandırma koşullarına bağımlı olmadan belirlemektir. Bununla birlikte yüz tespiti, görüntünün yüz içeren kısmı üzerinde sayısal kaynakları ayarlama becerisinden dolayı bir çok yüz tanıma sisteminin temel kısmıdır[12]. Yüz tespiti işlemi gelişi güzel seçilmiş her resimde, ten rengi, poz, ifade, pozisyon, yön , gözlük, yüzün bir kısmını kaplayan saçlar ve şapka gibi değişkenlerden dolayı oldukça karmaşık bir işlemdir. Bununla birlikte, aydınlatma koşulları, resim çözünürlüğü gibi etkiler de düşünüldüğünde daha da karmaşık hale gelir.

Yüz tespiti *özellik tabanlı* (feature-based method) ve *görüntü tabanlı* (image-based method) metotlar ile gerçekleştirilebilir.



Şekil 4. Yüz tespit metotları

1.3.1. Özellik Tabanlı Metotlar (Feature-based methods)

Bu alan, *düşük seviyeli analiz* (low-level analysis) olarak sınıflandırılan teknikleri içerirler. Bu metotlar, gri seviye yada renk gibi piksel özelliklerini kullanan görsel özelliklerin segmentasyonu ile ilgilidir. Düşük seviyeli metotların ortaya çıkardığı bu özellikler anlaşılması güç olsa da gerçekleştirilmesi oldukça basit ve hızlıdır. Tekniklerin diğer bir alanına ise *özellik analizi* (feature analysis) denir, burada yüz tespiti yüz geometrisi bilgilerini kullanan yüze ait özellikler üzerine dayanmaktadır. Özellik analizinde, özellik belirsizlikleri azaltılır, yüze ilişkin özelliklere ve yüz yerlerine karar verilir. Son grup ise *aktif biçim modellerini* (active shape models) içerir. Bunlar, göz ve dudak izleme gibi kompleks ve esnek özellik çıkarma amaçlı geliştirilmişlerdir. Düşük seviyeli analiz, *kenarlar* (edges) ve *renk segmentasyonu* (colour segmentation) olmak üzere ikiye ayrılır. Kenar çıkarma yaklaşımı, yüz tespiti uygulamalarındaki en ilkel özelliktir. Bu çalışma ilk [13] tarafından yapılmıştır. Çalışmanın temeli, yüze ilişkin özellikleri belirlemek amacıyla resimlerdeki yüzleri basit doğrular ile çizmeye dayanır. Bu konuda daha fazla çalışma [14] tarafından gerçekleştirilmiştir.

Burada insan kafasının konturu izlenir. Bu tespit programının temelini oluşturur, bir sonraki adım ise özellik analizidir. Burada, elde edilen şeklin insan yüz şekli olup olmadığına karar verilir.

Kenar çıkarma yaklaşımı, kompleks bir arka plana sahip olmayan ve önden görünümü net yüzlerde gerçekleştirilebilir. Renk segmentasyonu, ten bulma temeline dayanır. Özellik analizinde, özellik araması kullanılır. Burada, resimde göze çarpan özellikler saptanır. Daha sonra yüz geometrisi standart ölçüleri kullanılarak daha az göze çarpan özellikler araştırılır. Genellikle, gözler kullanılır. Diğer özellikler başın tepesini ve ana yüz eksenini kapsar. Bu metod da düzgün arka plan ve önden yüzler ile sınırlandırılmıştır. Bununla birlikte, yüze ait saç, küpe yada gözlük var ise başarısız olur. Aktif biçim modelleri, *çizgiler* (snakes) ve *noktasal dağıtılmış modeller* (point distributed models) olmak üzere ikiye ayrılabilir. Çizgiler, ilk olarak [15] tarafından tanıtılmıştır. Bunlar genellikle başın sınırını belirlemek için kullanılır. Çizgiler, minimum kıvrım sağlamaya çalışır ve bu da tamamen konveks olmayan yüz biçimleri için problem olabilir ve yanlış sonuçlar döndürmesine neden olabilir. Noktasal dağıtılmış modellerde, verilen resimde şeklin istatistiksel bilgisini alır ve bu şeklin baş olup olmadığına karar vermek için daha önceden tanımlanmış eğitim kümesi ile karşılaştırılır[16]. Bu yaklaşımın, resimlerdeki birden fazla yüzü tespit edebilmesi için daha fazla geliştirilmesine gerekmektedir.

1.3.2. Görüntü Tabanlı Metotlar (Image-based methods)

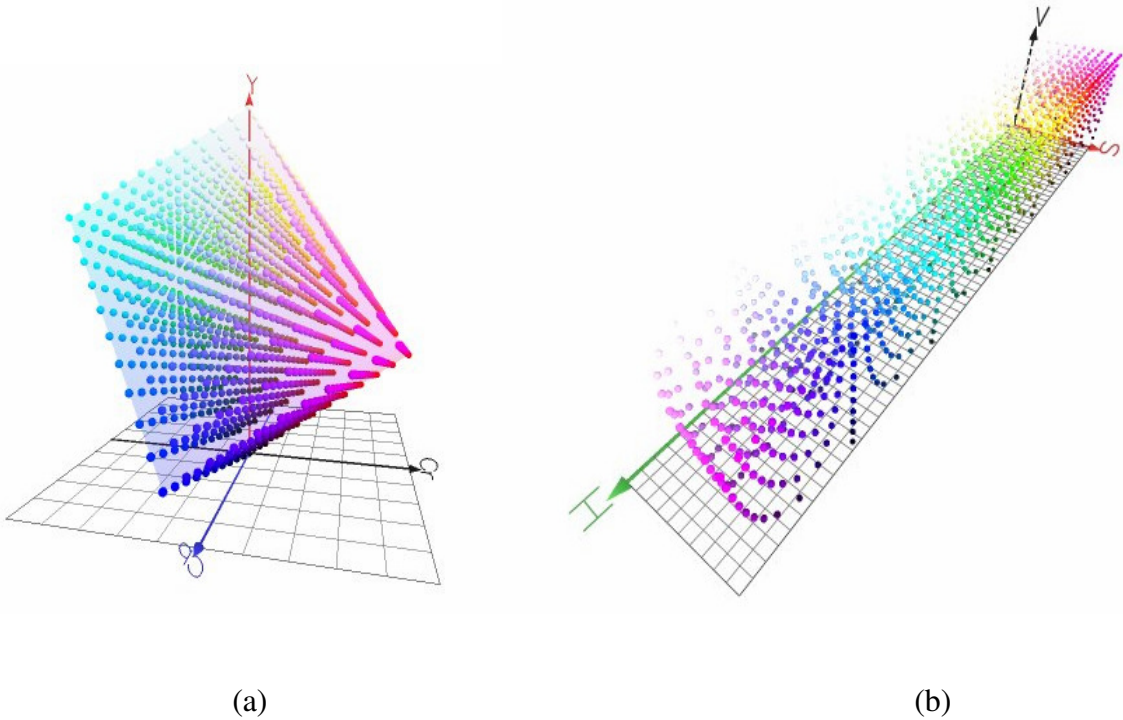
Görüntü tabanlı yaklaşım, resimlerin çevresel koşullarının tahmin edilemezliği ve önceden belirlenemeyen yüz görünüşleri ile uğraşır. Bu yaklaşım da genellikle, düz bir arka planda ideal koşullarda tek yüz tespiti ile sınırlandırılır. Bu nedenle, kompleks bir arka planda birden fazla yüz tespiti için yüz tespitinin *örnek tanıma* (pattern recognition) alanı geliştirilmiştir. Burada, yüz tanıma işlemi diğer yüz örnekleri ile karşılaştırılarak gerçekleştirilir. Bu yaklaşım, yüz ve yüz olmayan örnekleri sınıflamaya dayalıdır. Bu nedenle, yüz ve yüz olmayan örnek kümeleri oluşturulup, eğitilir. *Öz yüzler* (eigenfaces), *sinir ağları* (neural networks) ve *destek vektör makineleri* (support vector machines), görüntü tabanlı yaklaşımı kullanan kompleks tekniklerden bazılarıdır.

Çalışmada, yüz tespiti için başarılı bir teknik olan yapay sinir ağları (YSA) kullanılmıştır ve buna ten bulma yöntemi de eklenerek iyileştirme sağlanmıştır.

1.3.2.1. Ten Bulma Yöntemi (Skin Detection Method)

Renkli resimlerde, ten bulma işlemi yüz bulma için oldukça popüler ve yararlı bir tekniktir [17]. Çünkü RGB bileşenleri aydınlatma koşullarına bağımlıdır ve bu nedenle de yüz bulma, aydınlatma koşulları değiştiğinde başarısız olabilir. Bununla birlikte, ten bulma yöntemleri çoğu zaman tek başlarına yetersiz kalmaktadır. Bunun nedeni, yüz olmayan ama ten rengine çok yakın bölgelerin de bu yöntemle yüz olarak çıkarılabilemesidir. Bu nedenle, çalışmada ten bulma yöntemi yapay sinir ağları ile birleştirilerek güçlendirilmiştir. Yapay sinir ağlarında yanlış sonuç verebilecek bölgeler ten bulma ile elenirken, ten bulma yöntemi ile yanlış sonuç verebilecek bölgeler de yapay sinir ağları ile elenmiştir.

Ten bulma için RGB, normalize RGB, HSV, YIQ ve YCbCr gibi farklı renk uzayları kullanılabilir. Ten bulma işlemi için hangi renk uzayının en iyi olduğu hala bir tartışma konusudur. Shin[18] ve arkadaşları, ten ve ten olmayan renklerin arasındaki ayrılabilirliğin seçilen renk modeline dayandığını göstermişlerdir. 18 farklı renk uzayında dört tane ayrılabilir ölçü kullanarak, RGB ve YCbCr renk uzaylarının en iyi performans verdiği sonucuna varmışlardır. Bununla beraber, bu sonuçlar daha sonra Vezhnevets[19] ve arkadaşları tarafından sorgulanmıştır.



Şekil 5. (a) Üç boyutlu YCbCr renk uzayı, (b) Üç boyutlu HSV renk uzayı

Şekil 5’de YCbCr ve HSV uzaylarındaki renklerin dağılımları görülmektedir. Çalışmada, HSV ve YCbCr renk uzaylarını karşılaştırmak amacıyla resimler her iki renk uzayına da çevrilerek gözlemlenmiştir. Bunun için RGB bileşenleri aşağıdaki formül ile YCbCr bileşenlerine çevrilebilir.

$$\begin{aligned}
 \mathbf{Y} &= 0,299\mathbf{R} + 0,587\mathbf{G} + 0,114\mathbf{B} \\
 \mathbf{Cb} &= -0,169\mathbf{R} - 0,332\mathbf{G} + 0,500\mathbf{B} \\
 \mathbf{Cr} &= 0,500\mathbf{R} - 0,419\mathbf{G} - 0,081\mathbf{B}
 \end{aligned} \tag{1}$$

Ve RGB-HSV dönüşüm algoritması da aşağıdaki şekildedir.

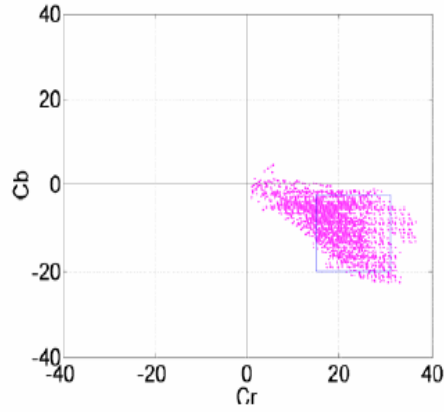
RGB-HSV dönüşüm algoritması

```

R=R/255 , G=G/255, B=B/255      // değerler 0-1 aralığına normalize edilir
min=MIN(R,G,B)                  // R, G, B değerlerinin minimumu
max=MAX(R,G,B)                  // R, G, B değerlerinin maksimumu
delta=max-min                    // maksimum ve minimum değerlerin farkı
v=max;
Eğer ( max != 0 ),               s = delta / max
Aksi takdirde ,                 s=0
Eğer (delta!=0),
Eğer ( r == max ),              h =60* ( g – b ) / delta
Aksi takdirde eğer ( g == max ), h =60* 2 + ( b – r ) / delta
Aksi takdirde,                 h = 60*4 + ( r – g ) / delta
Eğer (delta ==0)               h=0
Eğer ( h < 0 ),                 h+=360
s*=255, v*=255, h/=360

```

Çalışmada, hesaplama karmaşıklığının daha az olması ve daha küçük bir uzayda yer alması nedeniyle ten bulma işlemi için YCbCr renk uzayı seçilmiştir. YCbCr renk uzayında, Y bileşeni parlaklık bilgisini tutarken, Cb ve Cr bileşenleri ise renklilik bilgilerini tutar. Bu yüzden parlaklık bilgisi kolaylıkla çıkarılabilir. Ten bulma işleminde, Cb ve Cr bileşenleri kullanılarak her piksel ten yada ten olmayan olarak sınıflandırılır.



Şekil 6. YCbCr renk uzayındaki ten piksellerinin dağılımı

Sınıflama işleminde, Cb ve Cr bileşenlerinin maksimum ve minimum değerleri hesaplanarak bu değerler arasında kalan pikseller ten olarak işaretlenir. Cb ve Cr bileşenlerinin minimum ve maksimum değerlerine, bu bileşenlerin ortalama değerine (mean) ve standart sapmalarına göre karar verilir. Cb ve Cr bileşenlerinin minimum ve maksimum değerleri aşağıdaki gibi hesaplanır.

$$\begin{aligned}
 \mathbf{Cb}_{\min} &= \text{Mean_Cb} - \text{Std_Cb} * f \\
 \mathbf{Cb}_{\max} &= \text{Mean_Cb} + \text{Std_Cb} * f \\
 \mathbf{Cr}_{\min} &= \text{Mean_Cr} - \text{Std_Cr} * f \\
 \mathbf{Cr}_{\max} &= \text{Mean_Cr} + \text{Std_Cr} * f
 \end{aligned} \tag{2}$$

Çalışmada, Mean_Cb= -11.1051, Std_Cb= 4.3568, Mean_Cr= 22.9265, Std_Cr= 3.9479 olarak alınmıştır. Faktör (f), değeri ise değiştirilerek farklı sonuçlar gözlenebilir. Bu kısım daha sonra anlatılacaktır.

1.3.3. Yapay Sinir Ağları (YSA)

Yapay sinir ağları (YSA) birbirine bağlı çok sayıda işlemci birimin oluşturduğu bir sistemdir. İnsan beyni deneyimlerinden faydalanarak öğrenir [20]. YSA, biyolojik sinir ağları model alınarak geliştirilmiştir. Genel anlamda YSA, beynin bir işlevi yerine getirme yöntemini modellemek için tasarlanan bir sistem olarak da tanımlanabilir.

Yapay sinir ağları günümüzde bir çok probleme çözüm üretebilecek yeteneğe sahiptir. Değişik şekillerde tanımlanmaktadır. Tanımların ortak birkaç noktası vardır. Bunların en

başında yapay sinir ağlarının birbirine hiyerarşik olarak bağlı ve paralel olarak çalışabilen yapay hücrelerden oluşmaları gelmektedir. Proses elemanları da denilen bu hücrelerin birbirlerine bağlandıkları ve her bağlantının bir değerinin olduğu kabul edilmektedir. Bilginin öğrenme yolu ile elde edildiği ve proses elemanlarının bağlantı değerlerinde saklandığı dolayısıyla dağıtık bir hafızanın söz konusu olduğu da ortak noktaları oluşturmaktadır. Proses elemanlarının birbirleri ile bağlanmaları sonucu oluşan ağa *yapay sinir ağı* denmektedir. Bu ağın oluşturulması biyolojik sinir sistemi hakkındaki bulgulara dayanmaktadır.

1.3.3.1. Yapay Sinir Ağlarının Özellikleri

YSA' nın temel işlem elemanı olan hücre *doğrusal değildir*. Dolayısıyla hücrelerin birleşmesinden meydana gelen YSA da doğrusal değildir ve bu özellik bütün ağa yayılmış durumdadır. Bu özelliği ile YSA, doğrusal olmayan karmaşık problemlerin çözümünde en önemli araç olmuştur. YSA, ilgilendiği problemdeki değişikliklere göre ağırlıklarını ayarlar. Yani, belirli bir problemi çözmek amacıyla eğitilen YSA, problemdeki değişimlere göre tekrar eğitilebilir, değişimler devamlı ise gerçek zamanda da eğitime devam edilebilir. Bu özelliği ile YSA, uyarlamalı örnek tanıma, sinyal işleme, sistem tanılama ve denetim gibi alanlarda etkin olarak kullanılır. YSA, paralel yapısı nedeniyle büyük ölçekli entegre devre teknolojisi ile gerçekleştirilebilir. Bu özellik, YSA' nın hızlı bilgi işleme yeteneğini artırır ve gerçek zamanlı uygulamalarda tercih edilir.

1.3.3.2. YSA'nın Uygulama Alanları

Son yıllarda YSA' ları, özellikle günümüze kadar çözümü güç ve karmaşık olan yada ekonomik olmayan çok farklı alanlardaki problemlerin çözümüne uygulanmış ve genellikle başarılı sonuçlar alınabilmiştir. Silahların otomasyonu ve hedef izleme, nesnelere/görüntüleri ayırma ve tanıma, yeni algılayıcı tasarımı ve gürültü önleme alanlarında uygulanmıştır. Uçaklarda otomatik pilot sistemi otomasyonu, ulaşım araçlarında otomatik yol bulma/gösterme, robot sistemlerin kontrolü, doğrusal olmayan sistem modelleme ve kontrolü, elektrikli sürücü sistemlerin kontrolü için kullanılmaktadır. Bununla birlikte, ses tanıma, yazılan karakteri tanıma, resim işleme ve yüz tanıma sistemlerinde de yaygın bir biçimde kullanılmaktadır.

Tanıma sistemlerinde genellikle çok katmanlı ağlar (ÇKA) kullanılmaktadırlar. Bunun nedeni, ÇKA'nın doğrusal olmayan problemleri başarıyla çözmesidir. Fakat çözebileceği problemlerin sınırları çok iyi bilinmemektedir.

1.3.3.3. Ağın Yapısı ve Eğitimi

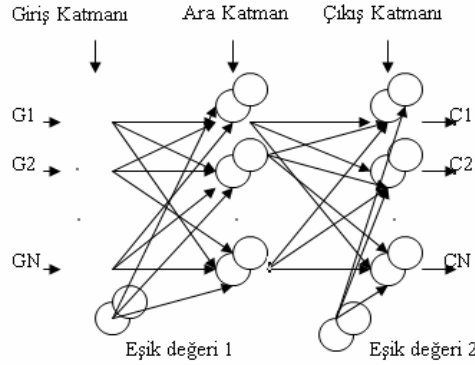
Çalışmada, yüz tespiti için çok katmanlı ağ kullanılmıştır. Çok Katmanlı ağlar, birbirine bağlanan üç katmandan oluşur. Bunlar, dışarıdan bilgiyi kabul eden bir girdi katmanı, bilgiyi işleyen ve özellikler çıkaran bir ara katman ve sorgulanan problem üzerinde ağın kararı hakkında ağ dışı kaynakları bilgilendiren bir çıktı katmanıdır. Bilgi, bu katmanlar arasındaki bağlantılarda yayılır. Ağ, öğrenme kuralı vasıtasıyla kendi zekasını geliştirir ve bu zeka, ağın ağırlık değerleri olarak bağlantılarda saklanır. Çok katmanlı ağlar öğretmenli öğrenme stratejisine göre çalışırlar. Yani bu ağlara eğitim sırasında hem girdiler hem de o girdilere karşı üretilmesi gereken çıktılar gösterilir. Bir girdi/ çıktı örneği ağa verilir. Ağ, girdiyi süzgeçten geçirir, çıktıyı üretir ve üretilen çıktıyı doğru çıktı ile karşılaştırır. Bağlantılardaki ağırlıklar, daha iyi çıktıyı üretmek için yeniden ayarlanır ve bu işlem, kabul edilebilir bir hata seviyesine erişinceye dek devam eder.

Girdi Katmanı : Dış dünyadan gelen girdileri (G_1, G_2, \dots, G_N) olarak ara katmana gönderir. Bu katmanda bilgi işleme olmaz. Gelen her bilgi geldiği gibi bir sonraki katmana gider. Birden fazla girdi gelebilir. Her proses elemanının sadece bir tane girdisi ve bir tane çıktısı vardır. Bu çıktı bir sonraki katmanda bulunan bütün proses elemanlarına gönderilir. Yani, girdi katmanındaki her proses elemanı bir sonraki katmanda bulunan proses elemanlarının hepsine bağlıdır.

Ara Katman : Ara katmanlar girdi katmanından gelen bilgileri işleyerek bir sonraki katmana gönderir. Bir ÇKA ağından birden fazla ara katman ve her katmanda birden fazla proses elemanı olabilir. Ara katmandaki her proses elemanı bir sonraki katmandaki bütün proses elemanlarına bağlıdır.

Çıktı Katmanı : Ara katmandan gelen bilgileri işleyerek ağa girdi katmanından verilen girdilere karşılık ağın ürettiği çıktıları ($\zeta_1, \zeta_2, \dots, \zeta_N$) belirleyerek dış dünyaya gönderir. Bir çıktı katmanında birden fazla proses elemanı olabilir. Her proses elemanı bir önceki katmanda

bulunan bütün proses elemanlarına bağlıdır. Her proses elemanının sadece bir tane çıkışı vardır.



Şekil 7. Çok katmanlı ağ yapısı

Yüz tespitinde, yüz ve yüz olmayan resimler için tek çıkış kullanılmaktadır. Yüz örnekleri için 1, yüz olmayan örnekler için ise 0 çıkışı yüklenmektedir.

Çok katmanlı ağlar, en küçük kareler yöntemine dayanan genelleşmiş delta öğrenme kuralını kullanırlar[21]. Genelleşmiş delta öğrenme kuralı iki kısımdan oluşur:

- 1) *İleriye doğru hesaplama* : Ağın çıkışı hesaplanır.
- 2) *Geriye doğru hesaplama* : Ağırlıklar değiştirilir.

İleriye doğru hesaplama : Giriş katmanındaki k. sinirin çıkışı aşağıdaki şekilde hesaplanır ve ara katmana iletilir.

$$Ç_k^i = G_k \quad (3)$$

Burada G_k , k. girişi ve $Ç_k^i$ ise k.girişin, ara katmandaki i. sinire giden çıkışı göstermektedir.

Ara katmandaki her proses elemanı girdi katmanındaki bütün proses elemanlarından gelen bilgileri bağlantı ağırlıklarının (A_1, A_2, \dots) etkisi ile alır. Önce ara katmandaki proses elemanlarına gelen net girdi (NET_j^a) aşağıdaki formül kullanılarak hesaplanır.

$$NET_j^a = \sum A_{kj} Ç_k^i \quad (4)$$

Burada A_{kj} k. girdi katmanı elemanını j. ara katman elemanına bağlayan bağlantının ağırlık değerini göstermektedir. j. ara katman elemanının çıktısı ise bu net girdinin aktivasyon fonksiyonundan geçirilmesi ile hesaplanır. Aktivasyon fonksiyonu olarak gerçek sinir hücrelerine benzerliğinden dolayı sigmoid fonksiyonu kullanılmıştır. Sigmoid fonksiyonu aşağıdaki şekilde tanımlanır.

$$y_j = \frac{1}{1 + e^{-x_j}} \quad (5)$$

Sigmoid fonksiyonunun kullanılması halinde çıktı,

$$Ç_j^a = \frac{1}{1 + e^{-(NET_j^a + \beta_j^a)}} \quad (6)$$

şeklinde olacaktır. Burada β_j , ara katmanda bulunan j. elemana bağlanan eşik değer elemanının ağırlığını göstermektedir. Bu eşik değeri ünitesinin çıktısı sabit olup 1'e eşittir. Ağırlık değeri ise *sigmoid* fonksiyonunun oryantasyonunu belirlemek üzere konulmuştur. Eğitim esnasından ağ bu değeri kendisi belirlemektedir.

Ara katmanın bütün proses elemanları ve çıktı katmanının proses elemanlarının çıktıları aynı şekilde kendilerine gelen NET girdinin hesaplanması ve *sigmoid* fonksiyonundan geçirilmesi sonucu belirlenirler. Çıktı katmanından çıkan değerler, yani çıktıları, ($Ç_1, Ç_2, \dots$) bulununca ağın ileri hesaplama işlemi tamamlanmış olur.

Geriye doğru hesaplama : Ağa sunulan girdi için ağın ürettiği çıktı ağın beklenen çıktıları (B_1, B_2, \dots) ile karşılaştırılır. Bunların arasındaki fark hata olarak kabul edilir. Amaç bu hatanın düşürülmesidir. O nedenle geriye hesaplamada bu hata ağın ağırlık değerlerine dağıtılarak bir sonraki iterasyonda hatanın azaltılması sağlanır. Çıktı katmanındaki m. Proses elemanı için oluşan hata (E_m),

$$E_m = B_m - Ç_m \quad (7)$$

olacaktır. Bu bir proses elemanı için oluşan hatadır. Çıktı katmanı için oluşan toplam hatayı (TH) bulmak için bütün hataların toplanması gerekir. Bazı hata değerleri negatif olacağından toplamın sıfır olmasını önlemek amacı ile ağırlıkların kareleri hesaplanarak sonucun kare kökü alınır. ÇKA ağıının eğitilmesindeki amaç bu hatayı en aza indirmektir. TH şu formül ile bulunur.

$$TH = 0.5 * \sum_m E_m^2 \quad (8)$$

Toplam hatayı en aza indirmek için bu hatanın kendisine neden olan proses elemanlarına dağıtılması gerekmektedir. Bu ise proses elemanlarının ağırlıklarını değiştirmek demektir. Ağı ağırlıklarını değiştirmek için iki durum söz konusudur.

- Ara katman ile çıktı katmanı arasındaki ağırlıkların değiştirilmesi
- Ara katmanlar arası veya ara katman girdi katmanı arasındaki ağırlıkların değiştirilmesi

Ara katman ile çıktı katmanı arasındaki ağırlıkların değiştirilmesi : Ara katmanındaki j. proses elemanını çıktı katmanındaki m. Proses elemanına bağlayan bağlantının ağırlığındaki değişim miktarına ΔA^a denirse ; herhangi bir t zamanında (t. iterasyonda) ağırlığındaki değişim miktarı aşağıdaki gibi hesaplanır.

$$\Delta A^a_{jm}(t) = \lambda \delta_m \zeta_j^a + \alpha \Delta A^a_{jm}(t-1) \quad (9)$$

Burada λ öğrenme katsayısını, α momentum katsayısını göstermektedir. Öğrenme katsayısı ağırlıkların değişim miktarını, momentum katsayısı ise ÇKA ağıının öğrenme esnasında yerel bir optimum noktaya takılıp kalmaması için ağırlık değişim değerinin belirli bir oranda bir sonraki değişime eklenmesini sağlar.

$$\delta_m = \zeta_m (1 - \zeta_m) \cdot E_m \quad (10)$$

Burada δ_m , m. çıktı ünitesinin hatasını göstermektedir. Değişim miktarı hesaplandıktan sonra ağırlıkların t. iterasyondaki yeni değerleri aşağıdaki gibidir.

$$A^a_{jm}(t) = A^a_{jm}(t-1) + \Delta A^a_{jm}(t) \quad (11)$$

Benzer şekilde eşik değer ünitesinin de ağırlıklarını değiştirmek gerekmektedir. Onun için öncelikle değişim miktarını hesaplamak gerekir. Eğer çıktı katmanında bulunan proses elemanlarının eşik değer ağırlıkları β^c ile gösterilirse; bu ünitenin çıktısının sabit ve 1 olması nedeni ile değişim miktarı,

$$\Delta\beta_m^c(t) = \lambda\delta_m + \alpha \Delta\beta_m^c(t-1) \quad (12)$$

olacaktır. Eşik değer t . iterasyondaki ağırlığının yeni değeri ise,

$$\beta_m^c(t) = \beta_m^c(t-1) + \Delta\beta_m^c(t) \quad (13)$$

şeklinde hesaplanacaktır.

Ara katmanlar arası veya ara katman girdi katmanı arasındaki ağırlıkların değiştirilmesi : Dikkatle incelenirse, ara katman ile çıktı katman arasındaki ağırlıkların değişiminde her ağırlık için sadece çıktı katmanındaki bir proses elemanının hatası dikkate alınmıştır. Bu hataların oluşmasında girdi katmanı ve ara katman arasındaki ağırlıkların payı vardır. Çünkü, en son ara katmana gelen bütün bilgiler girdi katmanı veya önceki ara katmandan gelmektedir. O nedenle girdi katmanı ile ara katman arasındaki ağırlıkların değiştirilmesinde çıktı katmanındaki proses elemanların hepsinin hatasından payını alması gerekir. Bu ağırlıklardaki değişim ΔA^i ile gösterilirse değişim miktarı aşağıdaki gibi olacaktır.

$$\Delta A_{kj}^i(t) = \lambda\delta_j^a C_k^i + \alpha \Delta A_{kj}^i(t-1) \quad (14)$$

Buradaki hata terimi δ^a ise şöyle hesaplanacaktır.

$$\delta_j^a = C_j^a (1 - C_j^a) \cdot \sum_{\text{III}} \delta_m A_{jm}^a \quad (15)$$

Hata değeri hesaplandıktan sonra (14) eşitliği kullanılarak değişim miktarları bulunur. Ağırlıkların yeni değerleri ise,

$$A_{kj}^i(t) = A_{kj}^i(t-1) + \Delta A_{kj}^i(t) \quad (16)$$

şeklinde olacaktır. Benzer şekilde, eşik değer ünitesinin yeni ağırlıkları da yukarıdaki gibi hesaplanır. Ara katman eşik değer ağırlıkları β^a ile gösterilir.

$$\Delta \beta_j^a(t) = \lambda \delta_j^a + \alpha \Delta \beta_j^a(t-1) \quad (17)$$

Ağırlıkların yeni değerleri ise t. iterasyonda şöyle hesaplanacaktır.

$$\beta_j^a(t) = \beta_j^a(t-1) + \Delta \beta_j^a(t) \quad (18)$$

Böylece ağırlıklarının hepsi değiştirilmiş olacaktır. Bir iterasyon hem ileri hem de geriye hesaplamaları yapılarak tamamlanmış olacaktır. İkinci bir örnek verilerek sonraki iterasyona başlanır ve aynı işlemler öğrenme tamamlanıncaya kadar yinelenir.

1.3.3.4. Çerçeve Birleştirme (Box Merge)

Resim sorgulama sonucunda bir yüze ait birden fazla bulgu elde edilebilmektedir, bunun tek bir bulguya indirgenmesi için *Çerçeve Birleştirme* (Box- Merge) algoritması kullanılmıştır[17]. Bulgular çerçeve ile gösterildiğinden bir yüz için birden fazla bulgu yani çerçeve var ise tek çerçeveye indirgenir. Burada, örtüşen çerçeveler bulunur ve bu çerçevelerin sol üst köşe noktalarının ortalaması alınarak bu ortalama en yakın çerçeve seçilir ve sonuç tek bulguya indirgenmiş olur.

1.4. Psikolojik Durum Değerlendirilmesi

İnsan, ilişkileri içinde sürekli yeniden tanımlanan bir varlıktır. Diğer insanlarla hiç ilişkisi olmayan bir insan düşünülemez. İnsanlarla kurduğumuz ilişkiler bireysel ihtiyaçlarımız ve toplumsal rolümüz gereği çok değişik düzey ve boyuttadır. İnsanlar arası ilişki değişik semboller aracılığıyla kurulur. Bu semboller kimi zaman sözlü-yazılı, kimi zamanda sözsüz sembollerdir. Kendi gerçeklerimizi karşımızdakine çeşitli sembollerle aktarıyoruz. İnsan bilerek yada bilmeyerek, çoğunlukla farkında olmaksızın günlük beden dilini son derece etkili kullanır. Ancak bedenini kelimeleri kontrol ettiği gibi kontrol edemez. Bedenimiz olaylara veya durumlara karşı çok daha kendiliğinden tepkiler verir. Gerçek duygu ve düşüncelerimizi kelimelerin ardına gizlememiz mümkündür ama, beden dilimizi gizlememiz çoğu zaman mümkün değildir. İyi bir dinleyici, iletişim kurduğu kişinin yalnız söylediklerini değil, yüzü, eli, kolu ve bedeniyle yaptıklarını da duyar. çünkü, yüz ifadeleri, el ve kol hareketleri, beden duruş tarzı, sesin tonu gibi sözsüz mesajlar kullanarak da iletişim kurulur. Yüz İfadeleri : Zihnimizden, gönlümüzden silinmeyen, zengin anlamlarıyla içimizde

derinleşen pek çok yüz vardır. İnsanlar genellikle duygu ve yüz ifadelerinin birbiri ile doğrudan ilişkili olduğuna inanırlar. Ancak, bilim adamları duygular ile bu duyguların yüzle ifadeleri arasındaki ilişkinin böylesine açık olmadığını düşünmüşler ve çeşitli araştırmalara yönelmişlerdir. Bu araştırmaların sonucunda kişiye ve sosyal yapıya ait bilinç ve bu bilincin oluşturduğu düşüncenin, insan yüz ifadesini etkilediği ortaya çıkmıştır[22]. İnsanın düşüncesi; duygu ve yüz ifadeleri arasındaki doğrudan bağlantıyı engeller, etkiler ve yönlendirir. Bununla birlikte, yüz ifadeleri evrensel bir dil gibidir. Bazı yüz hareketleri bütün kültürlerde aynı anlamda kullanılmaktadır: Şaşırıldığında kaşları kaldırmak veya sinirlendiğinde kaşları çatmak gibi. Dolayısıyla neşe, sinirlilik veya bıkkınlığın Avustralyalı bir yerlide bulunduğu yüz ifadesiyle Norveç'teki bir balıkçının arasında pek bir fark yoktur.

Bazı kültürlerde insanlar bıkkınlık ve sinirlilik halinde de gülümseyebilmekle beraber alın, kaşlar ve gözler daha güvenilirdir ve kültürel farklılıklardan en az etkilenir. Ekman ve Freisen'in (1969) yaptığı bir çalışmada farklı kültürlerden insanlara beyaz ırka ait insan yüzlerindeki çeşitli duyguların gösterilip deneklerden bu yüzleri duygulanımla eşleştirmeleri istendiğinde, kültürler arasında şaşkınlık ve korkuyu tanımada farklılıklar olduğu, diğer duyguları tanımada bir fark bulunmadığı gösterilmiştir. Aynı yöntemle İngiliz, İtalyan ve Japon deneklerle yapılan bir diğer çalışmada, her 3 grubun İtalyan ve İngiliz duygulanımını tanıyabildiği, fakat Japon yüzlerini tanımada güçlük çektiği gözlenmiştir (Ekman ve Keltner 1997).

Tıbbi açıdan incelendiğinde, yüze ait mikro ifadeler bulunmaktadır[23]. Yüz mimiklerinin çoğu yaklaşık 1 saniye sürerler. Ancak bunlardan daha kısa süren mikro ifadeler de vardır. Bunlara dikkat edilerek maskelenmiş ifadeler hakkında değerli bilgiler edinilebilir. Mikro ifadeler saniyenin 1/5-1/25'i arasında süren kısa ifadelerdir. Bu süre yaklaşık olarak bir göz kırpma süresi kadardır. Bununla birlikte, İnsanın en önemli ifade organının gözleri olduğunu söyleyebiliriz. Kaşların 40, göz kapaklarının ise 23 ayrı ifade şekillerinin olduğu bilinmektedir. Bu ifadeler birbirleriyle kombine edildiğinde ve buna ifadenin süre ve şiddeti eklendiğinde ortaya çok büyük bir ifade seçeneği çıkmaktadır. Gözler, kaşlar ve alına göre korkuyu ifade etmede daha etkili iken sinirlilik ve bıkkınlığı ifade etmede o kadar etkili değildirler. Hatta tek başına alt göz kapağı bile çok şey ifade edebilir. İnsanlar hoşlarına giden bir şey gördüklerinde gözbebekleri büyürken hoşlarına gitmeyen bir şey gördüklerinde küçülür. Bu istemsiz sinyal hastanın duygularını anlamak için yeterli olabilir. Belki de bu nedenle mücevher alıp satan birçok tüccar değerli bir taşla karşılaştıklarında gözbebeklerinin görülmemesi için koyu renkli gözlük takarlar.

Yapılan ilginç bir araştırmada, Kör ve sağır olarak doğan 6 aylık bebeklerde yapılan bir çalışmada, bu bebeklerin sağlıklı bebeklerle aynı yüz ifadeleriyle duygularını aktardıkları gösterilmiştir (Aboitiz ve Garcia 1997) [24]. Bununla birlikte Şizofreni hastaları, yüz ifadelerini, davranış veya ses tonlarını değiştirerek duygusal tepkilerini dışa vuramazlar. Mutluluk yada üzüntü verici olaylara duyarsız kalır veya uygun biçimde davranış göstermezler. Ayrıca, deneyimli doktorlar, yüz ifadesine göre insanın sağlık durumu, onun bağışıklık sistemi potansiyeli ve nihayet, karakter yapısı ve psikolojisi hakkında bilgi edinebiliyorlar. Rusya'nın ünlü iç hastalıkları uzmanı G.A. Zaharin, hastanın dış görünüşüne bakarak neredeyse yüzde yüz teşhis koyabiliyordu[25]. Bir zamanlar N.I. Pirogov, "Hasta insanın Yüzü" haritasını çıkarmıştı. O, her bir hastalığın insan yüzünde belli bir iz bıraktığını iddia ediyordu. Fakat, insanın yüz ifadesine göre teşhis koyma yöntemi Uzakdoğu ülkelerinde (özellikle Çin ve Kore) yaygın şekilde kullanılmıştır. Tibet tıp yöntemlerini bilen hiçbir deneyimli doktor hastanın yüzünü dikkatle incelemeyen teşhis koymaz.

Tüm bunlar, yüz ifadelerinin yaşamımızdaki önemini gösterir. Çalışmada, psikolojik durum değerlendirilmesi için yüz ifadeleri tanınması üzerinde durulmuştur. Kişi eylemlerinin izlenmesinde güvenlik sistemlerinde, hastanelerde hasta izleme otomatik uyarı sistemlerinde de uygulamaları yapılabilir.

Altı temel yüz ifadesi ve onların sayısız bileşimleri üç büyük yüz bölgesinde açığa vurulur. Bu ifadeler; şaşırma, korku, tikslenme, öfke, mutluluk ve üzüntüdür. Bu duygular çeşitli bileşimler halinde de görülebilir. Örneğin, şaşırma ve korku, öfke ve tiksinti yada mutluluk ve hüznün aynı anda yaşanabilir ve yüze yansır. Üç büyük yüz bölgesi kaş, gözler ve alt yüzdür. Yüz ifadelerinin doğru yorumlanabilmesi için bu bölgelerin özelliklerinin bilinmesi ve gözlenmesi gerekir. Yapılan bir araştırmada neşe ve mutluluk ifadelerinin en iyi ağız ve gözle, kızgınlık ifadesinin kaş ve dudak biçimiyle, hayret ve sürpriz ifadesinin ise en iyi gözle belirtildiği ortaya çıkmıştır[3][26].

Yüz ifadesi tanınmasında, insanlar tarafından bile bazen kolay olmadığı düşünüldüğünde, bilgisayar ile bu işlemin gerçekleşmesinin ne derecede zor olduğu anlaşılmaktadır. Ayrıca, bu ifadelerin kişiler arasında farklılığını da düşünürsek problemin karmaşıklığı daha iyi anlaşılır. Bu nedenle, yüzdeki en ufak mimikler ve ayrıntılar oldukça önemlidir.

Son 10 yılda yüz tanımada bilgisayarlı uygulamalara yoğun bir şekilde yer verilmektedir. Bu konularda birçok yaklaşım geliştirilmiştir. Shinohara ve Otsu, özellik vektörünü oluşturmak için Yüksek Düzenli Yerel Oto Korelasyon(HLAC-Higher-Order Local

Auto-Correlation) özellikleri ile Fisher ağırlık haritasını (Fisher weight map) birleştirmişler [8]. Lyons ve arkadaşları 2D Gabor dalgacıkları (Gabor wavelets) betimlemesi ve lineer fark sınıflayıcısı (LDC - linear discriminant classifier) şeması kullanmıştır [9]. Zhang vb. yüz betimlemede geometrik pozisyon ile Gabor dalgacıklarının birleşimini, yüz ifadesi tanımada ise çok katmanlı perseptron kullanmaktadır [6], [10].

Yüz ifadelerinin tanınması iki ana kısımdan oluşur:

- 1) *Özellik Çıkarma*
- 2) *Örnek Tanıma*

1.4.1. Özellik Çıkarma

Özellik çıkarma işlemi için genel olarak iki yaklaşım geliştirilmiştir; geometrik özellik tabanlı metotlar ve görünüm tabanlı metotlar[5]. Gabor dalgacıkları görünüm özelliklerinin, geometrik özelliklerden daha etkili olduğunu kanıtlamıştır[6].

- *Geometrik Özellik Tabanlı metotlar* : Birbirleri ile ilişkili yüze ait belirli noktaların geometrik pozisyonları kullanılarak özellik vektörü oluşturulur. Burada anahtar noktalar, pozisyonları yüzün güvenilen noktaları olarak belirlenen noktalardır. Tipik olarak bu noktalar, göz, kaş ve ağız bölgelerinde yer alır. Bu noktaların hareketleri ölçülerek yüz ifadelerine karar verilir. Bu yaklaşımı kullanan çeşitli yöntemler geliştirilmiştir. Optik akış (Optic Flow), Elastik graf eşleme (Elastic Graph Matching) ve Aktif görünüm modelleri (Active Appearance Models) [27], bunlardan bazılarıdır.

- *Görünüm Tabanlı Metotlar* : Bu yaklaşımda yüz ifadeleri, yüz piksellerinin renkleri baz alınarak sınıflandırılır. Görünüm tabanlı metotlar oldukça yaygındır, bunlardan bazıları : Temel Bileşen Analizi (Principle Component Analysis – PCA), Gabor filtreleri, Yerel İkili Örnekler (Local Binary Patterns – LBP).

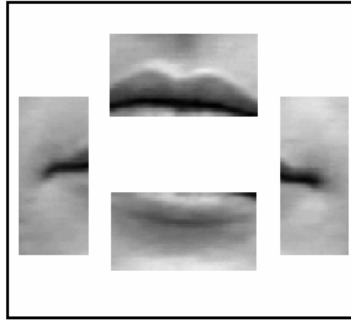
Özellik vektörünün uzay boyutunun azaltılması amacıyla araştırmalar, Temel Bileşen Analizi (Principle Component Analysis – PCA) yada Özgörüntü (Eigenimage) tabanlı tekniklere[28,29], Gabor filtre analizine[28], Yerel İkili Örneklerle[30] ve birleştirilmiş metotlara yönelmiştir. Temel Bileşen Analizi, eğitim setindeki yüz resimlerine ait öz yüzlerin

yada dikey temel vektörün azaltılmış kümesini hesaplar. Temel Bileşen Analizi, orijinal görüntü uzayından, boyutu azaltılmış dikey öz uzaya optimum doğrusal dönüşüm sağlar[31].

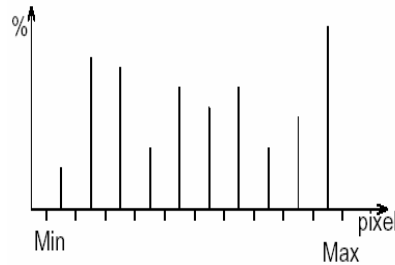
Çalışmada, görünüme dayalı özellik çıkarma metodu olan Yerel İkili Örnekler (LBP-Local Binary Patterns) kullanılmıştır. LBP özellikleri hızlı bir şekilde çıkarılabilir ve ayrıca yüze ait çok küçük örneklerin betimlenmesinde oldukça etkilidir. Bununla birlikte, Gabor dalgacıklarına göre daha az boyutlu bir uzayda tanımlanabilir.

1.4.1.1. Optik Akış Hesaplaması (Optical Flow Computation)

Optik akış algoritması P.Anandan tarafından ileri sürülmüştür[32]. Optik akış hesaplaması tespit edilmiş kaşlar üzerine uygulanmaktadır. Dudaklar ise 4 bölgeye bölünür ve her bölge üzerine optik akış algoritması uygulanır. Yüze ait özelliklerin ortalama optik akışını hesaplamak için istatistiksel bir yaklaşım kullanılmaktadır. İstatistikler, her pencere için optik akış alanından elde edilir. Özelliğin bütün hareketi olarak en sıradan hareket seçilir. Hareket bileşenleri x ve y, birkaç alana bölünür. En yüksek yüzdeli alan seçilir ve bu alan içerisindeki ortalama hareket ortalama akış olarak alınır[33].



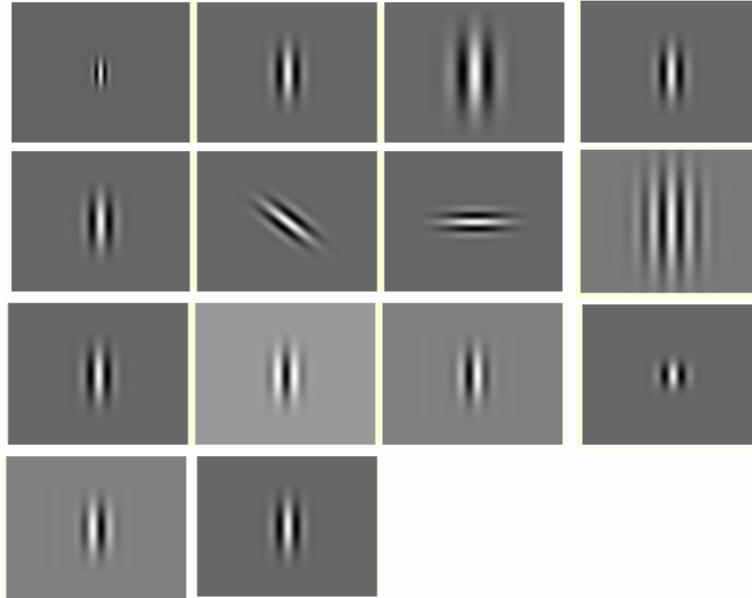
Şekil 8. Dudak bölgeler



Şekil 9. Hareket istatistiklerinin gösterimi

1.4.1.2. Gabor Filtreleri (Gabor Filters)

Gabor filtreleri, Gabor dalgacıkları (Gabor wavelets) olarak da bilinir ve yüz ifadesi tanımada sıkça kullanılır. İki boyutlu Gabor fonksiyonu, Gaussian tarafından modüle edilen W frekanslı bir sinüzoid tanımlar. Gabor fonksiyonları bir bütün oluşturur fakat ortogonal değildir, temel kümedir ve verilen $f(x,y)$ fonksiyonu basit fonksiyonlara ayrıştırılabilir[34]. Bu ayrıştırma, çeşitli ölçek ve yönlerdeki özellik tespitini mümkün kılmaktadır. Görüntünün Gabor filtreleri kullanılarak katlanması ile göz, ağız ve burun gibi çarpıcı yerel özelliklerin üretilmesi sağlanır ve bunlar görüntü tanımada kullanılmaktadır.



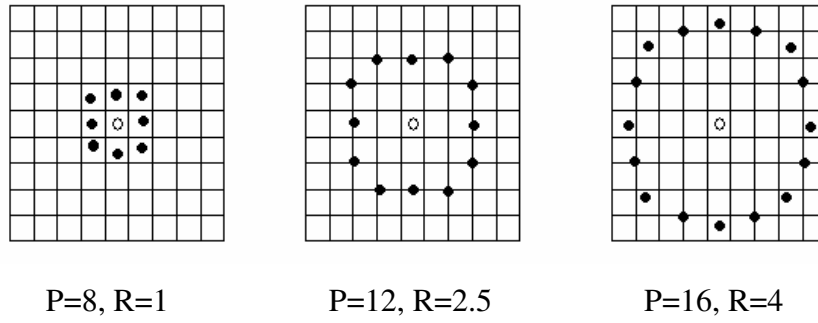
Şekil 10. Gabor filtreleri

1.4.1.3. Yerel İkili Örnekler (Local Binary Patterns – LBP)

Yerel İkili Örnekler operatörü, ilk olarak Ojala ve arkadaşları tarafından ortaya çıkarılmıştır ve doku tanımlayıcısı olarak önerilmiştir[35]. Daha sonra yüzleri betimlemek için de etkili bir yol olduğu görülmüştür[36, 37]. LBP, farklı yönlerdeki tek bir operatör ile kenar, eğri/doğru, köşe, eğri/doğru sonları ve nokta gibi 256 tane basit özellik bulucuyu kodlar. En önemli nokta ise her piksel için sadece en iyi eşleme çıkışını veren özellik bulucu kullanılmaktadır [38]. LBP'nin gücünün ana kısmı, örneklerin küçük alt kümeler şeklinde kodlanmış olmasıdır. Ve bu alt kümelerden en iyi örnek alt kümesi seçilerek kullanılır.

1.4.1.3.1. LBP Operatörleri

LBP operatörü, gri seviye ölçüsü olarak tanımlanır. Görüntüdeki her piksel için yerel komşuları merkez piksele göre eşiklendirilerek bir ikili kod üretilir. Bu gri seviye operatörünün, değişmeyen örneklerin sabit bir rotasyon kümesi içerdiği görülmüştür[31]. Örneklerin dairesel gösteriminde sınırlı sayıda geçişler vardır. Komşuluklarına göre farklı LBP operatörleri tanımlanabilir.

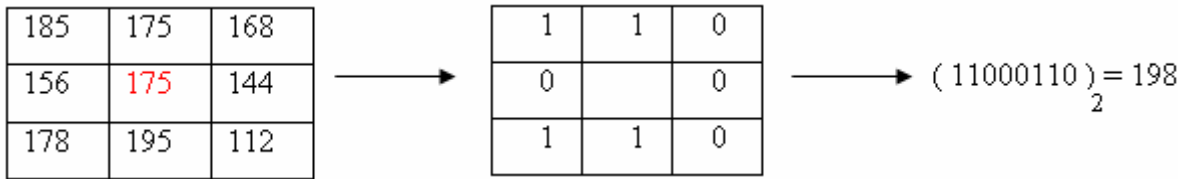


Şekil 11. LBP operatörleri

Genel olarak $LBP_{P,R}$ üç farklı dairesel simetrik komşuluklar kümesi ile tanımlanabilir, P komşu sayısını, R ise örnekleme yarıçapını gösterir.

1.4.1.3.2. Temel LBP

Temel LBP operatöründe 3x3 komşuluklar alınır ve komşuluklar merkez pikselin değerine göre eşiklendirilir. Eşiklendirilen piksel değerleri, ilişkilendirilen ağırlıklar ile çarpılır. Ve son olarak bu komşuluk için sekiz piksel değeri toplanarak tek bir LBP kodu elde edilir. LBP, tek düze gri seviye dönüşümlerinde piksel değerlerinin sırası aynı kaldığı sürece LBP operatörünün çıkışı değişmez, sabit kalır[39]. 3x3 komşuluğu için LBP kodu aşağıdaki şekilde hesaplanır.



Şekil 12. 3x3 komşuluklu LBP kodu hesaplaması

Genel olarak LBP kodunun hesaplaması aşağıdaki formül ile ifade edilir.

$$L = \sum_{i=0}^P s(g_i - g_c) * 2^i, g_c = \text{merkez pikselin gri seviye deęeri} \quad (19)$$

Bir özel dönüşüm fonksiyonu olan $s(g_i - g_c)$ ise (20) bağıntısına göre verilmektedir.

$$s(g_i - g_c) = \begin{cases} 0, & \text{eger } g_i < g_c \\ 1, & \text{aksi takdirde} \end{cases} \quad (20)$$

1.4.1.3.3. LBP Alt Kümeleri

3x3 komşuluğu LBP_{8,1} operatörü şeklinde ifade edilir. Daha önce de belirtildiği gibi burada 8 komşu sayısını, 1 ise 1 uzaklığındaki komşulara bakıldığını gösterir. Buna göre LBP_{8,1} operatörü için bakıldığında $2^8 = 256$ tane farklı LBP ikili kodu olmaktadır. Bütün LBP örneklerinin doku özelliğini iyi tanımlamadığı bilinmektedir. LBP, merkez pikselin komşularına karşı düzenliliğini aynen alır, fakat komşularının düzenliliğini aynen almaz [40]. Önemli örneklerin araştırılmasında iki yaklaşım ortaya çıkmıştır ; analitik ve sezgisel yaklaşımlar[38].

1.4.1.3.3.1. Tek Yönde Arama (Beam Search)

Örnekler için optimum alt kümeyi bulmak zaman alan bir işlem olabilir. Eğer bir D örnek kümesi yada toplam 256 örnekten daha az örnek seçilir ise F farklı küme ile sonlandırılır, F aşağıda tanımlanmıştır.

$$F = \sum_{i=1}^D \frac{256!}{(256-i)! \cdot i!} \quad (21)$$

Kolaylıkla görülüyor ki, orta D değerleri için bile örneklerin optimum alt kümesinin bulunmasının pratik olarak gerçekleşmesi mümkün değildir. Bu nedenle, optimum alt küme yerine iyi bir alt küme kabul edilir. Bu işlem için ışın arama metodu geliştirilmiştir. Burada, örnek kümelerinin boyutu iteratif olarak D'ye yükseltilir ve her zaman en iyi B örnek

kümeleri üretilmeye çalışılır. Bu yaklaşıma ilişkin algoritma aşağıdaki şekilde ifade edilmektedir.

Tek yönde arama algoritması

- S kümesi oluşturulur ve böylece $|S| = 1$ and $|s_j| = 0$ olur
- $d = 1$ to D
- L kümesi oluşturulur ve $|L| = 0$ olur
- $b = 1$ to B
- $a-s_b$ 'den s_b 'ye kadar sırayla her örnek için ilave edilerek bütün mümkün örnek s_i kümelerinin yerine tek bir S' kümesi oluşturulur
- Her $s_i' \in S$ için verilen optimizasyon kümesine karşılık olarak sınıflama hataları elde edilir
- $L = L + S'$
- Eğer $d=1$ ise, b döngüsünden çık
- b döngüsünün sonu
- L 'yi elde edilen sınıflama hatalarına göre sırala (aynı l_i değerleri için bir tane al)
- $S = \{ l_i \mid i = 1, \dots, B \}$
- d döngüsünün sonu

Burada, S örnek kümelerini, $|S|$; S kümesindeki alt kümelerin sayısını, s_i , $i= 1,2,\dots |S|$, alt kümeleri ve a ise tüm örneklerin kümesini ifade eder. Her iterasyonda yapılan optimizasyon kümesi sınıflandırma işlemi boyutu azaltılmış LBP histogramı kullanılarak gerçekleşir. Belirlenmiş sayıda iterasyondan sonra, S kümesi B tane belirgin örnek kümesi içerir ve buradan optimum örnekler seçilir.

1.4.1.3.3.2. Düzenli Örnekler (Uniform Patterns)

Genel olarak, tek yönde arama yaklaşımına alternatif olarak düzenli yerel ikili örneklerden oluşan özel bir kümenin kullanılması önerilir. Belirli bir örnekler kümesi kullanılmasının avantajı, arama prosedürü için harcanan zamandan tasarruf edilir. Bunun için bir düzenlilik ölçüsü U tanımlanmıştır. İkili dizi dairesel olduğunda 0-1 ya da 1-0 en fazla iki bit geçişi varsa LBP örneği düzenlidir denir. Buna göre, 0 (00000000) ve 255 (11111111) örneklerinin düzenlilik değeri $U=0$ iken, 1, 2, 4, 8, 16, 32, 64 ve 128 (00000001,

00000010,...) örneklerinin düzenlilik değeri $U=2$ 'dir. Benzer şekilde, 00000011, 00000111, 00001111, 00011111, 00111111, 01111111 örneklerinin düzenlilik değeri $U=2$ ve diğer örneklerin düzenlilik değerleri en az $U=4$ 'tür. Düzenlilik ölçüsü belirlenirken, eğim yada yön gibi geometrik dönüşümler üzerinde istenmeyen değişikliklere daha az maruz kalan örnekler dikkate alınmıştır[38]. Bunun sonucunda, düzenlilik ölçüsü en fazla 2 olarak belirlenmiştir. Buna göre, düzenlilik aşağıdaki formüller ile ifade edilebilir.

$$U_c = \sum_{i=0}^P |s(g_i - g_c) - s(g_{i-1} - g_c)| \quad (22)$$

$$LBP_{c,j} = \begin{cases} \sum_{i=0}^{P-1} s(g_i - g_c), & \text{eger } U_i \leq j, \quad j=2 \\ P+1, & \text{aksi takdirde} \end{cases} \quad (23)$$

Burada, U merkez piksel c 'nin düzenlilik ölçüsünü ifade etmektedir.

$LBP_{8,1}$ operatörü için bakıldığında toplam $2^8 = 256$ tane farklı LBP örnekleri bulunmaktadır. Düzenlilik formülleri kullanıldığında 58 tane düzenli örnek elde edilir ve $LBP^{u2}_{8,1}$ ile ifade edilir. 58 düzenli örneğin dışında kalan örnekler tek bir örnek altında etiketlenir ve böylece 256 yerine $58+1=59$ tane özellik vektörü elde edilir. Etiketlenmiş görüntünün $f(x,y)$, histogramı aşağıdaki şekilde tanımlanır.

$$H_i = \sum I(f(x,y) = i), \quad i=0, \dots, n-1 \quad (24)$$

Burada n , LBP operatör tarafından üretilen farklı etiketlerin sayısını gösterir, $I(A)$ ise aşağıdaki şekilde tanımlı bir özel dönüşüm fonksiyonudur.

$$I(A) = \begin{cases} 1, & \text{eğer } A \text{ doğru ise} \\ 0, & \text{eğer } A \text{ yanlış ise} \end{cases} \quad (25)$$

Bu histogram, kenarlar, noktalar ve köşeler gibi yerel bir çok küçük örneklerin tüm görüntüde dağılımı hakkında bilgi içerir.

Yüz resimleri, LBP histogramları ile etkili bir biçimde tanımlanabilen çok küçük örneklerin birleşimi olarak görülebilir. Tanıma işlemi için yerel bölgeler üzerindeki LBP histogramları daha güvenilir bir tanımlama sağlamaktadır. Bu nedenle, LBP özelliklerinin elde edilmesi için yüz resimleri eşit olarak küçük bölgelere bölünür. Bununla birlikte yüz resim boyunun ve pozisyonunun sabit olması LBP için bir dezavantajdır, fakat iyi sonuçlar elde edilmesinde oldukça etkilidir [41]. Daha sonra belirtileceği gibi yüz ifadesi tanıma uygulamalarında işlemin zorluğu nedeniyle bu tür kısıtlamalar bulunmaktadır.

Bölümleme sonucu N bölge elde edildiyse, her bölgeye ait 59 özellik olacağından özellik vektörünün boyutu $59 \times N$ olmaktadır. Buna göre, yerel LBP histogramları aşağıdaki formül kullanılarak ifade edilir.

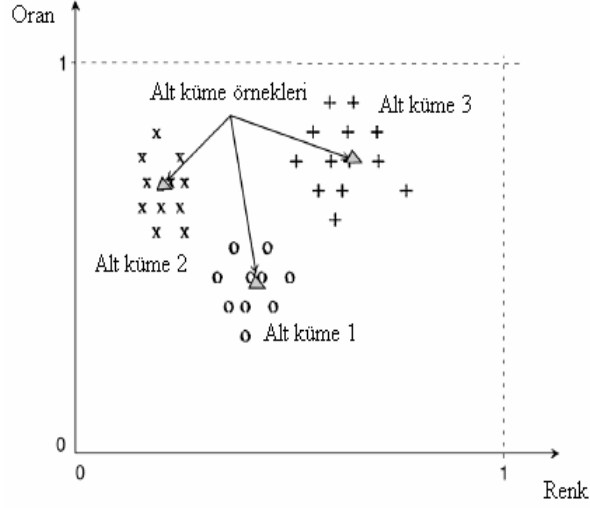
$$H_{i,j} = \sum_{x,y} I(f(x,y) = i), \quad (26)$$

Burada $I(f(x,y) \in R_j)$, $i=0, \dots, n-1$; $j=0, \dots, m-1$ olmaktadır. m bölge sayısını, n ise LBP operatörü tarafından üretilen 59 etiket sayısını ve R_j bölgeleri göstermektedir. LBP operatörünün bir avantajı da histogramların çok hızlı bir şekilde hesaplanabilmesidir.

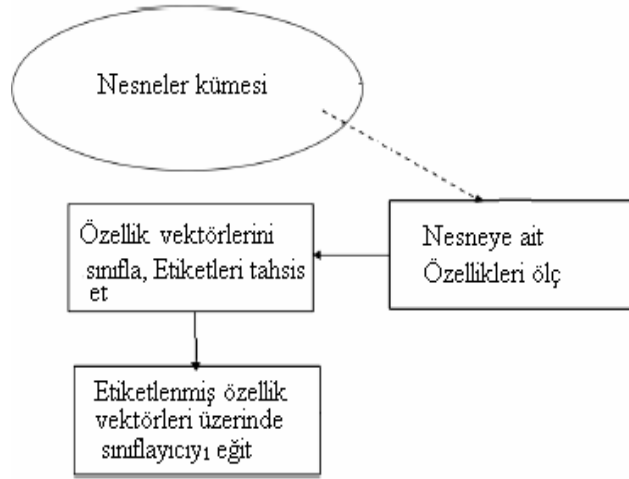
1.4.2. Örnek Tanıma

Giriş verilerinin daha önceden veritabanında yer alan bilgilere dayanarak girilen verilerinin özelliklerinin ayrışabilir sınıflara göre tanımlanmasına örnek tanıma (pattern recognition) denir. Sınıflama, örnek uzayının özelliklerine göre sınıflara ayrılması işlemidir. Örnek sınıfının özellikleri diğer örneklere ait özelliklerden farklı olarak sadece bu sınıfa ait olan karakteristiklerdir. Bu özellikler ile örnek sınıfları birbirinden ayrılırlar ve bazı farklılıklar gösterirler. Örnekler vektör şeklinde gösterilirler ve her bir X_i bir örnek sınıfını temsil eder. Sınıflama işleminde, bu örnekler incelenir ve benzerlik ve farklılıklarına göre sınıf adı verilen alt kümelere parçalanır. Bu tip işleme, öz organizasyon (self-organization), denetlenmemiş öğrenme (unsupervised learning) yada örnekleri kümelere kümeleme (clustering) denir. Öte yandan, diğer bir işlem ile örnek vektörü sınıflara kümelendir ve her nesneye bir sınıf etiketi ve özellik sınıfını belirten indeks değeri tahsis edilir. Ve aynı popülasyondan bilinmeyen nesnelere de sınıf etiketi tahsis etmek için bir sınıflayıcı

eğitilebilir. Bu eğitim işlemine denetlenmiş öğrenme (supervised learning) yada sınıflayıcının eğitimi denir[42]. Eğitilmiş sınıflayıcı örnek tanıma işlemini gerçekleştirir.



Şekil 13. Kümeleme



Şekil 14. Sınıflayıcının gelişimi

Sınıflama işlemi için özellik seçimi oldukça önemlidir [43]. Eğer iki özellik birbirleri ile güçlü ilişkilendirilmiş ise aynı özellik vektörleri olarak ayırt edilirler ve fazlalık olarak nitelendirilirler. Fazlalık, bir örnek üzerinde özelliklerin bağıntılarının gerçekleşmesi ile tespit edilir. En az kümeyi elde etmek için ilişkilendirilmiş özellik vektör çiftlerinin her birini elemine edebiliriz. Her çift sınıfı ayıran en az bir özellik olmalıdır.

1.4.2.1. K- Ortalama Algoritması ile Kümeleme (K-Means Algorithm)

Kümeleme, özellik vektörlerinden oluşan örnek kümesini sınıfları temsil eden alt kümelere parçalayan bir denetlenmemiş öğrenme işlemidir. K- ortalama algoritması sınıflama algoritmalarının en basitidir. 1965’de Forgy[44] ve 1967’de MacQueen[45] tarafından geliştirilmiştir.

Forgy’nin k-means algoritması

Adım 1: Q etiketlenmemiş özellik vektörleri, K ise sınıf sayısı olsun

Adım 2: Q özellik vektörlerinden ilk sınıfı seç

Adım 3 : K sınıflarını oluşturmak için her Q özellik vektörünü en yakın prototipe tahsis et, $c[q]=k$, $x^{(q)}$ nun k sınıfına ait olduğunu belirtir, $s[k]$ küme boyutunun sayısını belirtir ve her sınıfa bir vektör tahsis edildiğinde arttırılır.

Adım 4 : Yeni K merkezlerini bulmak için her sınıfın özellik vektörlerinin ortalaması alınır. k sınıfının özellik vektörlerinin ortalaması alınırken aşağıdaki yol izlenir. $a[n][k]=0.0$ ile başlanır, $a[n][k]$ k sınıfının ortalama vektörünün n bileşeninin değerini ve $x[n][q]$ ise $x^{(q)}$ nun n. Bileşenini göstermektedir.

```

for k = 1 to K do           // Her k sınıfı için:
for n = 1 to N do  $a[n][k] = 0.0$ ; // Başlangıç ortalamalarını setle
for q = 1 to Q do         // Verilen k sınıfı için, bul
if ( $c[q] = k$ )           sonra // k kümesindeki bütün vektörler
for n = 1 to N do         // ve her n bileşeni için,
 $a[n][k] = a[n][k] + x[n][q]$ ; // bu bileşeni topla
if ( $s[k] > 1$ )           sonra
 $a[n][k] = a[n][k]/s[k]$ ; // k kümesinin ortalaması

```

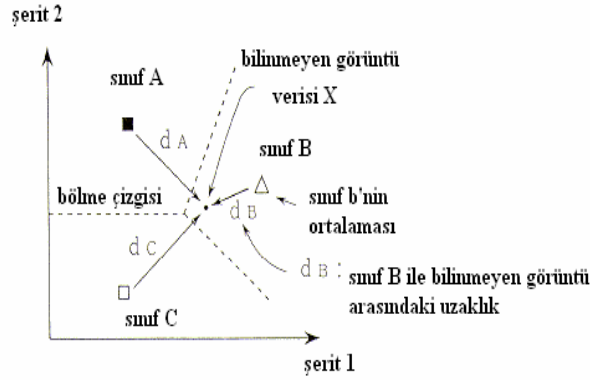
Adım 5 : Eğer sınıf değişmemiş ise çık, aksi takdirde adım 3’e geri dön.

Burada, Q etiketlenmemiş özellik vektörleri kümesini $\{ x^{(1)}, x^{(2)}, \dots, x^{(Q)} \}$, $x^{(q)} = \{ x^{(q)}_1, x^{(q)}_2, \dots, x^{(q)}_N \}$ N boyutlu özellik vektörünü, K ise kullanıcı tarafından belirlenen sınıf sayısını göstermektedir.

Sınıflama işleminden sonra artık her sınıfa ait özellik vektörü oluşturulmuş ve birbirlerinden ayrılmışlardır. Bundan sonra, verilen girişin hangi sınıfa ait olduğu tespit edilerek tanıma işlemi gerçekleşir.

1.4.2.2. Minimum Uzaklık (Minimum Distance)

Minimum uzaklık sınıflayıcısı, bilinmeyen görüntü verilerini, çoklu özellik uzayında görüntü ve sınıf mesafelerini küçültmek için kullanılmaktadır. Bu mesafe bir benzerlik indeksi olarak tanımlanır. Böylece minimum uzaklık maksimum benzerlikle hemen hemen aynıdır.



Şekil 15. Minimum uzaklık sınıflaması

Şekil 15'de minimum uzaklık sınıflayıcısına ait örnek bir şema gösterilmiştir. Burada, X verisi B sınıfına en yakın uzaklıktadır ve bu durumda x verisi B sınıfına atılır sonucuna varılır.

$$D^2 = \sum_{ij} (S_{ij} - M_{ij})^2 \quad (27)$$

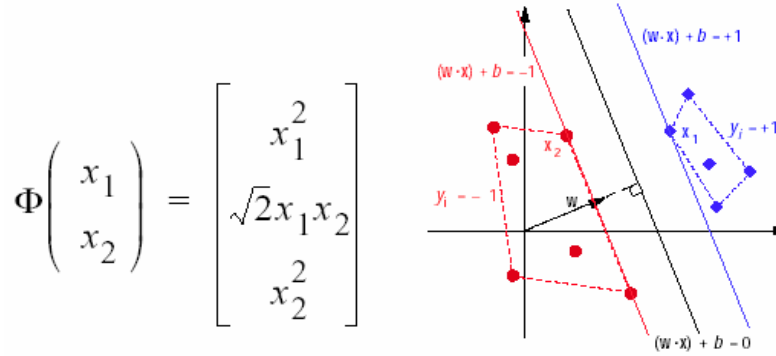
Burada S örneği, M ise modeli yani sınıfları göstermektedir.

1.4.2.3. Destek Vektör Makineleri (Support Vector Machines – SVMs)

Destek vektör makineleri (Cristianini & Shawe-Taylor, 2000), yüksek performans garantilemek için istatistiksel öğrenme teorisinin sonuçlarına dayandırılan çok yüksek düzeyde hiper düzlem sınıflama metodudur. Doğrusal olarak ayrılamayan giriş verisini yüksek boyutlu özellik uzayına etkili bir biçimde haritalamak için çekirdek fonksiyonları

kullanılır ve daha sonra bu uzayda doğrusal metotlar uygulanabilir. SVM, makul sayıda eğitim verisi olduğu zaman iyi bir sınıflama doğruluğu sergiler[46]. SVM, benzerlik ve doku tanınmasında, DNA analiz uygulamalarında başarılı bir şekilde kullanılmıştır.

- Ana düşünce : Yüksek boyutlu uzayda sınırlar doğrusal olabilir.
-



Şekil 16. Ana düşünceye ait örnek

- Destek vektörleri, kenarlardaki eğitim noktalarına dayanır.
- Verilen izdüşüm Φ için tek optimum çözüm
- Çözüm sadece izdüşürülmüş uzaydaki merkeze en yakın sonuçları içerir.

$$k(x,y) = \phi(x) \cdot \phi(y) = (x \cdot y)^2 \quad (28)$$

Burada, $k(x,y)$ çekirdek fonksiyonunu göstermektedir.

Yüz ifadesi tanımada alt bölgelere bölünmüş resim için SVM sınıflama formülü aşağıdaki şekilde ifade edilmektedir.

$$F(x) = \text{sgn} \left(\sum_{i=1}^l \alpha_i y_i K(x_i, x) + b \right) \quad (29)$$

Burada, T, etiketlenmiş örneklerden oluşan eğitim kümesini $T = \{ (x_i, y_i), i=1, \dots, l \}$, $x_i \in \mathbb{R}^n$ olmak üzere giriş verisi, \mathbb{R}^n yüze ait bölgeleri, $y_i \in \{ 1, -1 \}$ ve $K(x_i, x)$ çekirdek fonksiyonunu göstermektedir. Ayrıca, α_i ikili optimizasyon probleminin Lagrange çarpanlarını ve b, optimum hiper düzleme ait parametreyi göstermektedir. Sgn(x) fonksiyonu ise 0 yada 1 değerini üreten özel bir dönüşüm fonksiyonudur. Sonuç olarak SVM 1 yada 0 değerini üretir.

1.4.2.4. Uyarlanabilir Yükseltme (Adaptive Boost- AdaBoost)

Yükseltme (Boosting) metodu ile zayıf sınıflama fonksiyonları birleştirilerek güçlü bir sınıflayıcı oluşturulur. Uyarlanabilir yükseltme, bir dizi sınıflayıcıyı yükseltmek için kullanılan uyarlanabilir bir algoritmadır. Burada, ağırlıklar bir önceki öğrenmedeki hatalara göre dinamik olarak güncellenmektedir[47]. Uyarlanabilir yükseltme bir çeşit büyük çevreli sınıflayıcılardandır. Her zayıf sınıflayıcı iki sınıfı ayırmak için bir eşik değeri kullanır ve bu eşik değeri, tek boyutlu uzayda iki sınıf arasındaki orta nokta hesaplanarak elde edilir. T basamaklı arttırmadan sonra, T tane zayıf sınıflayıcı ile T tane özellik seçilir. Tieu ve Viola'nın[48] uyarlanabilir artırma algoritması aşağıdaki gibidir.

Uyarlanabilir Yükseltme Algoritması

Giriş : 1) n tane eğitim örneği, $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ ve $y_i = 1$ yada 0
2) T , iterasyon sayısı

Başlangıç ağırlıkları : $y_i = 1$ yada 0 için sırasıyla $w_{1,i} = (1/2l)$ yada $(1/2m)$, $n = m+l$

For $t=1 \dots T$

1) Her özellik j için w_t ile h_j hipotezini eğit ve hata $\epsilon_j = \Pr_i^{w_t} [h_j(x_i) \neq y_i]$

2) $h_t(\cdot) = h_k(\cdot)$ seç öyle ki her $j \neq k$, $\epsilon_k < \epsilon_j$. $\epsilon_t = \epsilon_k$

3) Ağırlıkları güncelle $w_{t+1,i} = w_{t,i} \beta_t^{e^i}$, $e^i = 1$ yada 0 sırasıyla x_i doğru yada yanlış sınıflandı, $\beta_t = (\epsilon_t) / (1 - \epsilon_t)$ ve $\alpha_t = \log(1/\beta_t)$

4) Ağırlıkları normalize et, $w_{t+1,i} = w_{t,i} / (\sum_{j=1}^n w_{t+1,j})$

Çıkış : sonuncu hipotez

$$h_f(x) = \begin{cases} 1 & \text{eğer } \sum_{t=1}^T \alpha_t \cdot h_t(x) \geq \frac{1}{2} \cdot \sum_{t=1}^T \alpha_t \\ 0 & \text{aksi takdirde} \end{cases} \quad (30)$$

1.4.2.5. Model Eşleme (Template Matching)

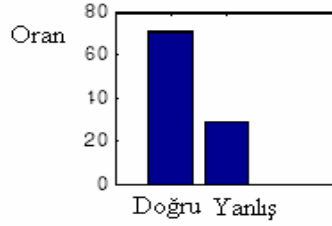
Eğitim kümesindeki aynı etiketli ifadelerin ortalaması alınarak modellere ait özellik vektörleri oluşturulur. Chi karesi istatistik ile model ve test örneği arasındaki farklılık ölçüsü belirlenir.

1.4.2.5.1. Chi Karesi İstatistik (Chi Square Statistics)

$$X^2(s, m^c) = \sum_j \frac{(s_j - m_j^c)^2}{s_j + m_j^c} \quad (31)$$

$$X^2(s, m^{c^d}) = \min_{1 \leq c \leq n} X^2(s, m^c) \quad (32)$$

Burada, $c=1, \dots, n$ yüz ifadelerini, m^c c.ifadeye ait modeli, s test örneğine ait özellik vektörünü, c^d test örneği ile eşleşen ifadeyi göstermektedir.



Şekil 17. Chi karesi ile doğru ve yanlış tanıma oranları

Yanlış tanıma sonuçları incelendiğinde yanlış çıkan sonuçların çoğunun ikinci minimum uzaklığın doğru sonucu verdiği gözlenmiştir[49].

1.4.2.5.2. Ağırlıklandırılmış Chi Karesi İstatistiği (Weighted Chi Square Statistics)

Model tabanlı yaklaşımların tanıma yeteneklerinin sınırlanmasının sebebi her ifade sınıfı için sadece tek bir model olması ve bu ifadelerin etkili bir biçimde temsili için oldukça önemli bir sınırlayıcıdır. Bu nedenle ağırlıklandırılmış Chi karesi istatistiği geliştirilmiştir.

$$X_w^2(s, m^c) = \sum_j w_j \frac{(s_j - m_j^c)^2}{s_j + m_j^c} \quad (33)$$

Yüze ait ifadelerin, çoğunlukla göz ve ağız bölgeleri gibi belli bölgelerde bulunduğunu ve bu bölgelerin yüz ifadesi tanımada daha yararlı bilgiler içerdiğini gözlemlenmiştir [50]. Bu yüzden, her yüz bölgesi için içerdiği bilginin önemine göre bir ağırlık belirlenebilir. Ağırlıklar renkleme ile betimlenir.

1.4.2.6. Yapay Sinir Ağları

Yapay sinir ağları günümüzde geliştirilmiş en güncel ve en mükemmel örüntü tanıyıcı ve sınıflandırıcılardan sayılabilirler ; bu ağları bu kadar güncel yapan da, eksik bilgiler ile çalışabilme ve normal olmayan verileri işleyebilme yetenekleridir. Özellikle çok sayıda veriyi işleme gerektiren işlerde çok avantajlı sonuçlar üretilmektedirler.

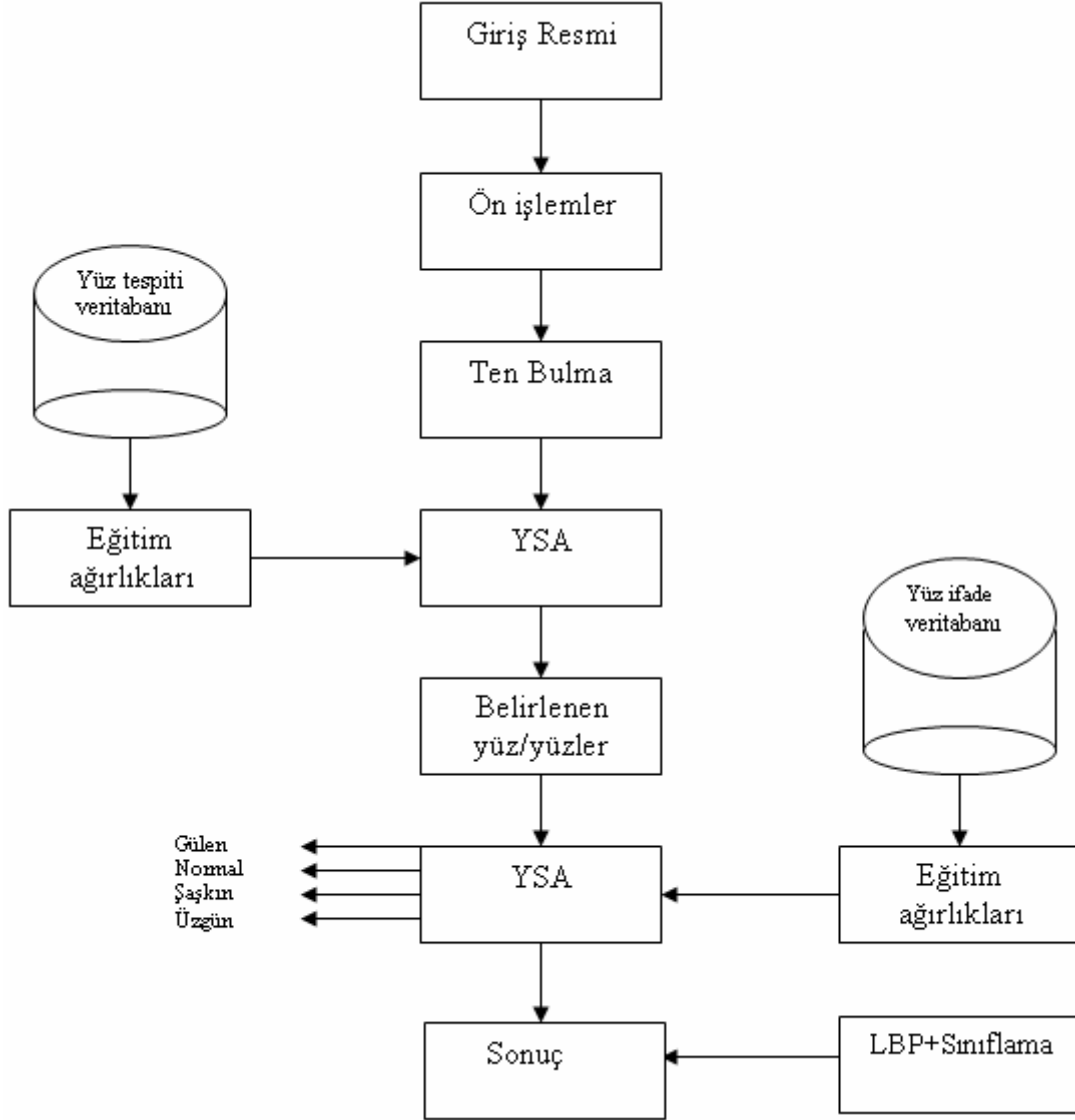
Genelde, verilen bir girdi setine karşılık çıktı değerleri verilerek belirtilen öğrenme kuralına göre ağırlık değerleri otomatik olarak değiştirilmektedir. Eğitim verisinin tamamlanmasından sonra eğitilmiş olan ağ, ağırlık değerlerinin son durumuna göre, verilen herhangi bir veri setinin sonucunu tahminleyebilmektedir. (Bu bölüm daha önce ayrıntılı olarak anlatılmıştır.)

2. YAPILAN ÇALIŞMALAR, BULGULAR VE İRDELEME

2.1. Sistem Gereksinimleri

Psikolojik durum değerlendirilmesi C++ ile geliştirilmiştir. Uygulama C++ Builder 5.0 ile gerçekleştirilmiştir. Sistem geliştirilirken YSA ile eğitim için JCreator Pro kullanılmıştır.

2.2. Sistemin Yapısı ve Aşamaları



Şekil 18. Sistemin genel yapısı

Sistem iki ana aşamadan oluşur. Birinci aşamada ten bulma ile iyileştirilmiş yapay sinir ağları yardımıyla yüz tespiti gerçekleştirilir. İkinci aşamada ise yüz resimlerine ait özellik vektörleri çıkarılıp sınıflama işlemi ile yüz ifadeleri tanıma işlemi gerçekleştirilir.

2.2.1. Yüz Tespiti

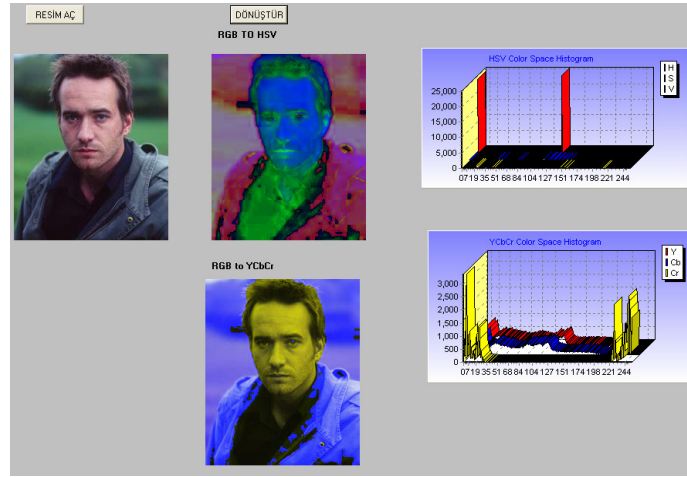
Yüz tespiti işlemi üç kısımdan oluşur.

1) Ten bulma

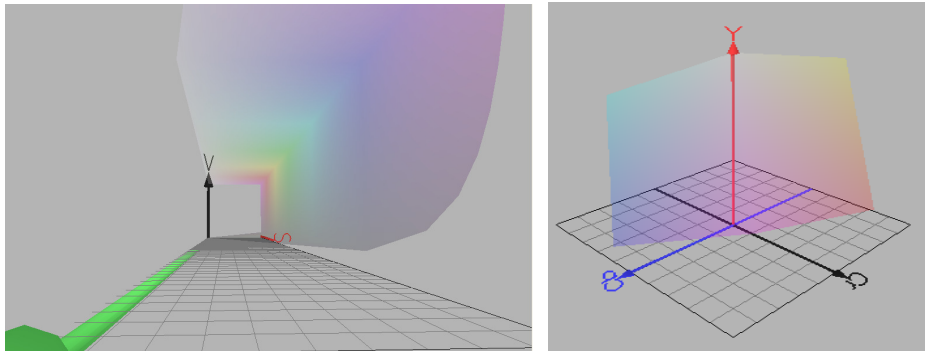
- 2) YSA
- 3) Ölçekleme
- 4) Yüze ait bulguların tek bulguya indirgenmesi

2.2.1.1. Ten Bulma

HSV ve YCbCr renk uzaylarını karşılaştırmak amacıyla çalışmada resimler her iki renk uzayına da çevrilerek gözlemlenmiştir. Dönüşüm formülleri *Genel Bilgiler* kısmında verilmiştir.



Şekil 19. RGB-HSV ve RGB-YCbCr dönüşüm sonuçları ve histogramları



Şekil 20. Resme ait 3 boyutlu HSV ve YCbCr renk uzayları görüntüsü

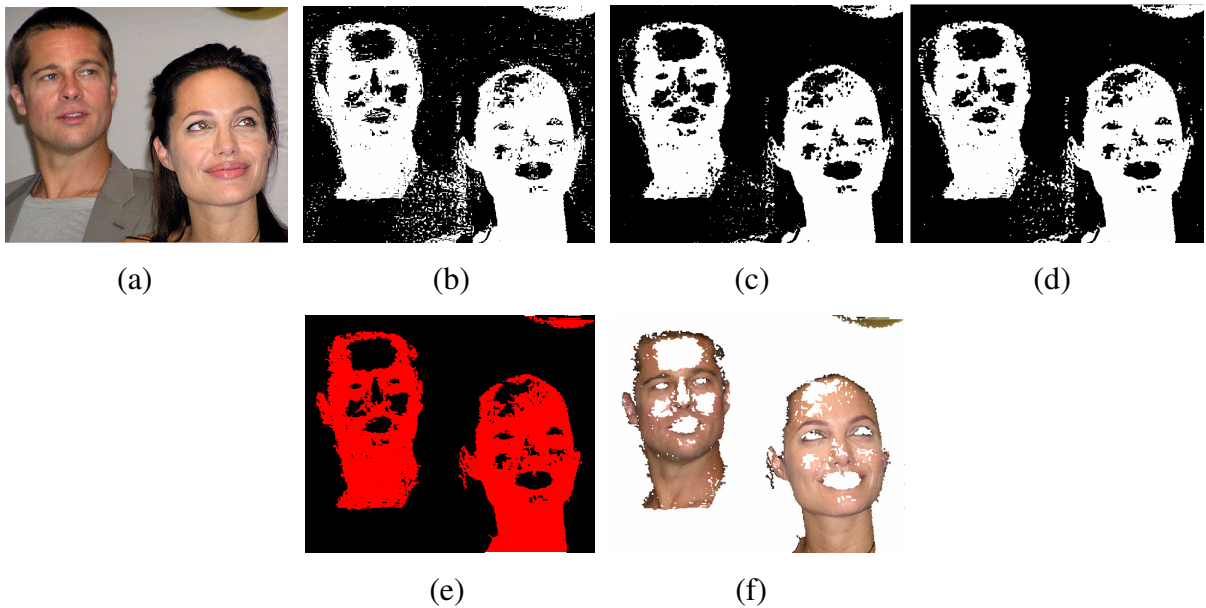
Şekil 20' den görüldüğü gibi YCbCr renk uzayı üç boyutlu uzayda HSV renk uzayına göre daha az yer işgal eder. Ayrıca, RGB-YCbCr dönüşümünün, RGB-HSV dönüşümüne

göre hesaplama karmaşıklığı daha azdır. Ten bulma işleminde iki uzayda oldukça başarılıdır. Bu nedenle, ten bulma işlemi için YCbCr renk uzayı kullanılmıştır.

RGB-YCbCr dönüşümünden sonra (2) formül kullanılarak ten pikselleri işaretlenir. Ten piksellerinin işaretlemesine ait C++ kodu aşağıda verilmiştir.

```
for(int i=0;i<Bitmap5->Height;i++)
for(int j=0;j<Bitmap5->Width;j++){
// y, cb, cr degerlerini ve bu bileşenlere ait min. ve max. değerleri hesapla
if((cb>min_Cb)&&(cb<max_Cb)) {
Bitmap5->Canvas->Pixels[j][i]=clWhite; }
if((cr>min_Cr)&&(cr<max_Cr)) {
Bitmap5->Canvas->Pixels[j][i]=clWhite; }
else Bitmap5->Canvas->Pixels[j][i]=0;
} //ten bulundu
```

Ten pikselleri işaretlendikten sonra küçük gürültülerin elemine edilmesi için erozyon işlemi ve erozyon işleminden sonra da zayıflatılan bağların güçlendirilmesi için genişleme (dilation) işlemi uygulanmıştır. Daha sonra, etiketleme (labelling) yöntemi ile ten olarak işaretlenen tüm kapalı alanlar etiketlenir. Özellikle kompleks bir arka plana yani ten rengi ile karışabilecek renklerde bir arka plana sahip olan resimlerde bu işlemlerin önemi daha iyi görülür.



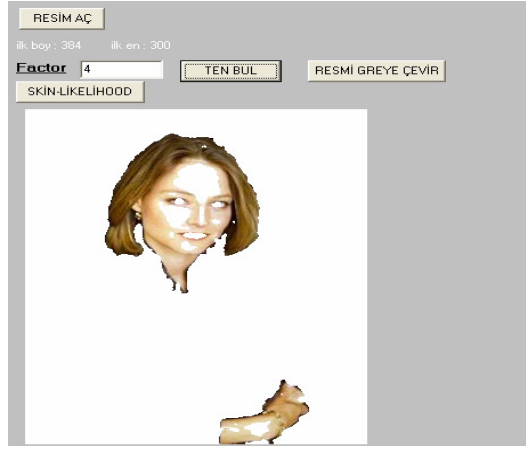
Şekil 21. Ten bulma işlem adımları

Şekil 21’ de (a) Orijinal resim, (b), Ten pikselleri işaretlenmiş resim(beyaz ile), (c) Erozyon uygulanmış resim, (d) Genişleme uygulanmış resim, (e) Etiketlenmiş resim, (f) Ten bulunmuş resim (Faktör(f)=4) ‘dir.

Etiketleme işlemi yüz olamayacak kadar küçük bölgelerin elenmesi için kullanılmıştır. Burada, 300’den küçük alanlar elenir. Çalışmada minimum 25x25 boyutlarındaki yüzler bulunması amaçlandığından $25 \times 25 = 625$ ve yüzün yarım görüldüğünü düşünürsek $625/2 = 312$ elde edilir ve bu sebeple tam olarak görünmeyen yüzleri de kaçırmamak amacıyla eşik değeri 300 olarak alınmıştır. Bu şekilde, çok büyük boyutlarda olmayan resimlerde el bölgelerinin elemesi de gerçekleşmiştir. Burada, yüz bölgeleri üzerindeki farklı gölgelemeler ve aynı resimde farklı ten rengi tonları sebebiyle bazı ten kopuklukları oluşmuştur. Ten renginin ırklara, güneş yanmalarına, ışıklandırma ve fotoğraf çözünürlüğü koşullarına göre değişkenliği göz önüne alındığında işlemin karmaşıklığı daha iyi anlaşılır. Etiketleme işlemine ait C++ kodu aşağıdaki gibidir.

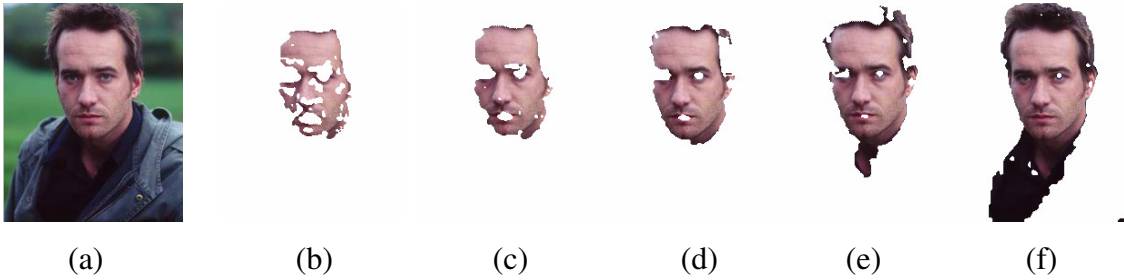
```
//cisimler dizisinin elemanlarını sıfırla
for( sat=0; sat<Bitmap17->Height; sat++){
for(sut=0; sut<Bitmap17->Width; sut++)
{ cis[sat][sut]=0;}}
say=0;
do{
//ilk görüntü pikselini cisim olarak seç
...
//komşu pikselleri aynı cisme ata
for( sat=0; sat<Bitmap17->Height; sat++){
for(sut=0; sut<Bitmap17->Width; sut++) {
if(resim3[sat][sut]==255) {
for(int sat1=sat-1; sat1<sat+2; sat1++){
for(int sut1=sut-1; sut1<sut+2; sut1++){
if((cis[sat][sut]!=0)&&(resim3[sat1][sut1]==255))
{cis[sat1][sut1]=cis[sat][sut]; // atanmış etiketin komşularını da etiketle
} }}}}
int x=0;
//cismi çıkar
```


Yukarıdaki C++ kodunda, *cis* dizisi ten bölge piksellerinin etiketlerini tutar, *say* değişkeni ise etiketi belirtir.



Şekil 22. Ten bulma arayüzü

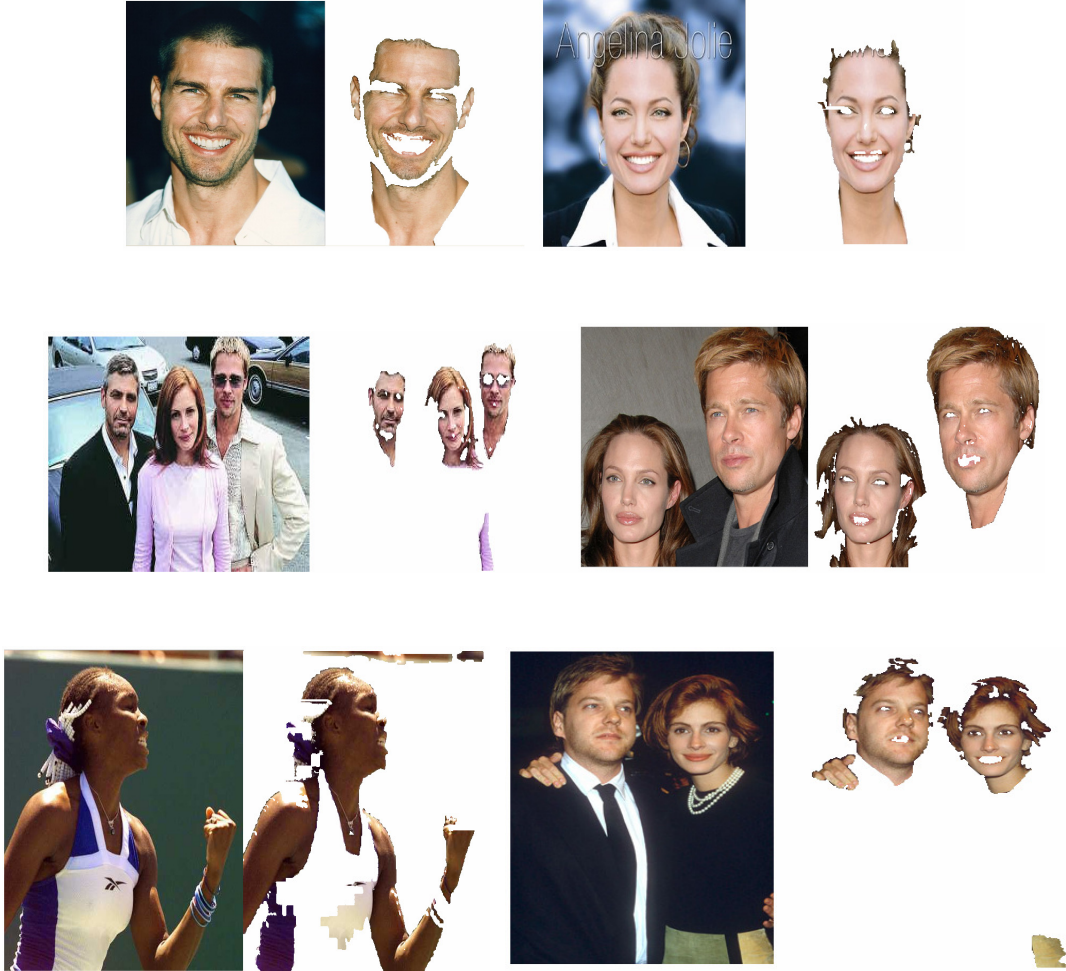
Formül (2) ile ifade edilen eşitlikte ten piksellerinin bulunması için faktör(f) değeri oldukça etkilidir. Bunun sonuçları aşağıdaki şekilde gözlenmiştir.



Şekil 23. Farklı faktör değerlerine ait ten bulma sonuçları

Şekil 23'de (a) Orijinal resim, (b) $f=1$ için ten bulunmuş resim, (c) $f=2$ için ten bulunmuş resim, (d) $f=3$ için ten bulunmuş resim, (e) $f=4$ için ten bulunmuş resim, (f) $f=5$ için ten bulunmuş resim'dir. Şekilde görüldüğü gibi faktör değeri küçük ise ten bulmada fazla miktarda kopukluklar gözlenmektedir fakat bu kopuklukları çok azaltmak için faktör değeri çok artırıldığında da bazı ten olmayan bölgelerin de ten olarak bulunduğu görülmektedir. Bununla birlikte, bir resimde en iyi sonucu $f=x$ değeri için verirken başka bir resimde $f=y$ değeri için verebilir, bunun sebebi ten renginin arka plan, gölgeleme ve genetik koşullardan da etkilenerek resimlere göre değişiklik göstermesidir.

Önemli olan yüz teninin dış hatlarının belirlenmesidir. Çünkü ileride anlatılacağı üzere, yüz araması ten bölgelerinin sınırları içerisinde gerçekleşecektir bu durumda sınırlar elde edildikten sonra ara kopuklukların önemi yoktur. Çalışmada, tüm bu sonuçlar değerlendirilerek optimum faktör değeri 4 olarak seçilmiştir.



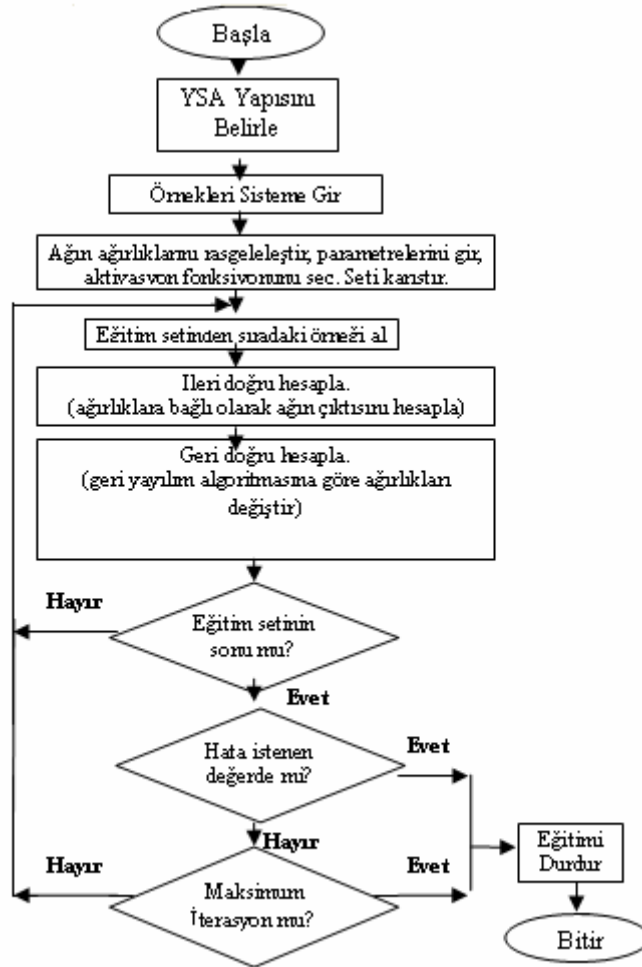
Şekil 24. Ten bulunmuş resimler

Şekil 24'den de görüldüğü gibi çalışmada koyu tenlerin de bulunması gerçekleşmiştir. Yüz tespitinde yüz kaçırılmaması amacıyla hiçbir ten bölgesinin kaçırılmaması gerekmektedir. Bunun sonucunda, bazen ten olmayan bazı küçük bölgeler de ten olarak işaretlenebilmektedir. Bunlar da bir sonraki aşama olan yüz tespitinde yapay sinir ağları ile elenecektir. Sonuç olarak, ten bulma işlemi ile sadece ten bölgeleri taranarak zamandan kazanılır ve yanlış sonuç verebilecek alanların elenmesi gerçekleştirilir.

2.2.1.2. YSA

Genel Bilgiler kısmında ayrıntılı olarak anlatılmıştır. Çalışmada, tanıma sistemlerindeki başarısından dolayı çok katmanlı ağlar kullanılmıştır. Resimlerdeki küçük yüzleri tespit etmek ve aynı zamanda yüz özelliklerini de betimleyebilmek açısından eğitim için 25x25 boyutunda resimler kullanılmıştır. Bunun sonucunda ağın giriş sayısı 625’dir, yani giriş katmanında 625 tane sinir kullanılmıştır. Ara katman için 10 sinir kullanılmıştır, yüz ve yüz olmayan örneklerin eğitimi için 10 sinir yeterli olmuştur. Yüz ve yüz olmayan resimler için tek çıkış kullanılmıştır. Yüz örnekleri için 1, yüz olmayan örnekler için ise 0 çıkışı yüklenmiştir.

Çalışmada, momentum değeri 0.2, öğrenme katsayısı ise 0.1 olarak alınmıştır. Eğitim aşaması aşağıdaki şekilde de ifade edilebilir.



Şekil 25. Eğitim aşaması akış şeması

Yapay sinir ağlarının eğitimine ait java kodu aşağıda verilmiştir.

```

for(int o=0; o<orsay; o++) {

    int j=0;

    While(j<arasay){

        aradeger=0.0;

        for(int g=0;g<girissay;g++){

            aradeger+=Giris[g][j]*Ornek[o][g];

        } aradeger+=esik1[j];

        Acikis[j]=sigmoid(aradeger);

        j++; }

        aradeger=0.0;

        for(int c=0;c<cikissay;c++) ara=0.0;

        for(int a=0;a<arasay;a++) {

            aradeger+=Agir[a][c]*Acikis[a];

        }aradeger+=esik2[c];

        Cikis[c][o]=sigmoid(aradeger); }

        error=false;

        for(int cs=0;cs<cikissay;cs++)

            error=(error||((Beklecik[cs][o]-Cikis[cs][o])!=0));

            if(error){ ... }

            for(int cs=0;cs<cikissay;cs++){

                sampleerror+=(Beklecik[cs][o]-Cikis[cs][o])*(Beklecik[cs][o]-Cikis[cs][o]); }

                sumerror=sumerror+sampleerror/2;

```

```
}/o for ornek sayisi kadar donen
```

```
System.out.println("toplamhata"+sumerror);
```

Yukarıdaki Java kodunda, *orsay*, *arasay*, *girissay*, *cikissay* sırasıyla örnek sayısını, ara katman sayısını, giriş sayısını ve çıkış sayısını belirtmektedir. *Giris*, *Ornek*, *esik1*, *Acikis*, *Agir*, *esik2*, *Cikis*, *Beklecik* dizileri sırasıyla girdi katmanı ile ara katman arasındaki ağırlık değerlerini, örneğe ait giriş değerlerini, ara katman eşik değerleri, ara katman çıkış değerleri, ara katman ile çıktı katmanı arasındaki ağırlıklar, çıktı katmanı eşik değerleri, örneğe ait çıkış değerleri ve beklenen çıkış değerlerini göstermektedir. *Sampleerror* ve *sumerror* değişkenleri ise sırasıyla örnek hatası ve toplam hatayı belirtir.

Eğitimde her örnek için maksimum iterasyon sayısı 950 olarak alınmıştır. Programda giriş, ara katman ve çıkış sayıları dinamiktir, kullanıcı tarafından belirlenir.

```
C:\PROGRA-1\XINOXS-1\JCREAT-1\GE2001.exe
giris sinir sayisi 625
ara sinir sayisi 10
cikis sayisi 1
agirliklar dosyadan okunacaksa 1, rasgele atanacaksa 2 girin : 1
hesaplanadan sonra sorgulama yapilacaksa 1, yapilmayacaksa 2 girin : 2
ornek sayisi 882
toplamhata0.4985985054288005
toplamhata0.4985773808608297
toplamhata0.49855700554294585
toplamhata0.4985371701770583
toplamhata0.49851771128453704
toplamhata0.4984984984256848
toplamhata0.4984794256468338
toplamhata0.4984604055431904
toplamhata0.498441365027182
toplamhata0.49842224224207077
toplamhata0.4984029842626382
toplamhata0.4983835453455559
toplamhata0.4983638855665271
toplamhata0.49834396972843015
toplamhata0.49832376645524146
toplamhata0.49830324740679766
toplamhata0.4982823865633168
toplamhata0.49826115953847677
toplamhata0.4982395428874171
toplamhata0.4982175133824135
toplamhata0.4981950472350152
toplamhata0.4981721192497048
toplamhata0.49814870190099836
toplamhata0.4981247643334434
toplamhata0.4981002712920656
toplamhata0.4980751819989403
toplamhata0.49804944899902653
toplamhata0.49802301700412116
toplamhata0.4979958217666647
toplamhata0.49796778901405897
toplamhata0.49793883346827017
toplamhata0.4979088579645261
toplamhata0.49787775266720363
toplamhata0.4978453943616612
toplamhata0.4978116457795224
toplamhata0.49777635489390704
toplamhata0.49773935410233844
toplamhata0.4977004592000477
toplamhata0.4976594680358904
toplamhata0.4976161587365006
toplamhata0.4975702873801969
toplamhata0.4975315840870850
```

Şekil 26. YSA eğitim çıktısı

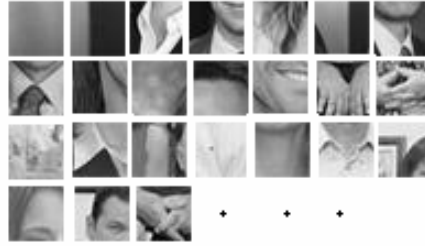
Yapay sinir ağlarının başarısında veritabanı örneklerinin çok büyük etkisi vardır. İyi bir veritabanı oluşturulur ise iyi sonuçlar alınır. Çalışmada, 300 yüz ve 615 yüz olmayan

toplam 915 örnek kullanılmıştır. Örnekler 25x25 boyutlarındadır. Küçük yüzleri kaçırmamak ve aynı zamanda yüz özelliklerini de korumak amacıyla 25x25 boyutları seçilmiştir.



Şekil 27. Yüz veritabanına ait bazı örnekler

Şekil 27’de görüldüğü gibi veritabanında $\pm 30^\circ$ dönmüş yüz örnekleri de bulunmaktadır.



Şekil 28. Yüz olmayan veritabanına ait bazı örnekler

2.2.1.3. Ölçekleme

Yüz tespitinde ölçekleme işlemi oldukça önemlidir. Ölçeklemeye bağlı olarak yüz tespiti sonuçları değişiklik gösterebilir. Bununla birlikte, bir resimde farklı büyüklükte yüzler olabilir, bunların yakalanması için de gereklidir.

Küçültme işleminde, küçültme oranı kadar pikselin ortalaması alınarak tek bir piksel ile ifade edilir. Gri seviye için kullanılan C++ küçültme kodu aşağıda verilmiştir.

```
for(int sat=0; sat < resboy; sat=sat+aen){
    kboy=0;
    for(int sut=0; sut < resen; sut=sut+aboy){
        sum=0;
        for(int i=0;i<aen&&(sat+i)<resboy;i++)
        {for(int j=0;j<aboy&&(sut+j)<resen;j++) {
            sum+=resim[sat+i][sut+j]; }}
    }
```

```

sum=sum/(aen*aboy);
kucuk[ken][kboy]=sum;
kboy++; } ken++; }

```

Yukarıdaki C++ kodunda, *aen*, *aboy* sırasıyla en ve boy küçültme oranlarını, *ken*, *kboy* ve *kucuk* değişkenleri ise sırasıyla küçültülen resmin en, boy ve piksel değerlerini göstermektedir.

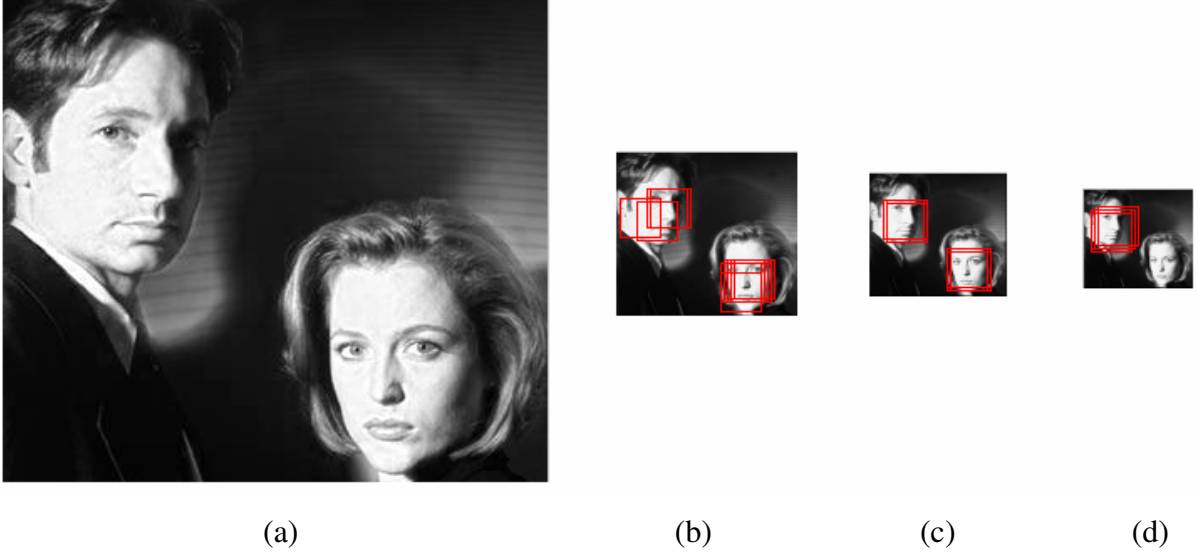
Büyütme işlemindeki mantık da küçültme işlemine benzer. Burada amaç piksel sayısını arttırmak. Bu nedenle, 2 katı büyütmek için tek bir pikselin 2 piksel ile ifade edilmesi gerekir. Bunun için, pikselin kendisi alınır, ikinci olarak da komşusu ile ortalaması alınarak 2 piksel elde edilir ve resim 2 katı büyütülmüş olur. RGB resmi için büyütme işlemine ait C++ kodu aşağıdaki gibidir.

```

for(int sat=0; sat < Resim->Height; sat++){
lineb=(byte *)Resim->ScanLine[sat];
boyb=0;
for(int sut=0; sut < Resim->Width; sut++){
sum1b=0;sum2b=0;sum3b=0;
res1b[sat][sut]=lineb[3*sut];
res2b[sat][sut]=lineb[3*sut+1];
res3b[sat][sut]=lineb[3*sut+2];
sum1b+=(res1b[sat][sut]+res1b[sat][sut+1])/2;
sum2b+=(res2b[sat][sut]+res2b[sat][sut+1])/2;
sum3b+=(res3b[sat][sut]+res3b[sat][sut+1])/2;
buyuk1b[enb][boyb]=res1b[sat][sut];
buyuk2b[enb][boyb]=res2b[sat][sut];
buyuk3b[enb][boyb]=res3b[sat][sut];
buyuk1b[enb][boyb+1]=sum1b;
buyuk2b[enb][boyb+1]=sum2b;
buyuk3b[enb][boyb+1]=sum3b;
... } enb=enb+2; }

```

Yukarıdaki kodda *enb* ve *boyb* değişkenleri büyütülen resmin yeni en ve boy değerlerini göstermektedir.



Şekil 29. Ölçekleme sonuçları

Şekil 29’da (a) Ten bölge sınırları bulunmuş resim, (b) 3x3 oranında ölçeklenmiş resim, (c) 4x4 oranında ölçeklenmiş resim, (d) 5x5 oranında ölçeklenmiş resim’dir. Şekilde görüldüğü gibi resim 3x3 ve 4x4 oranında küçültüldüğünde resimdeki iki yüz de bulunmaktadır fakat 5x5 oranında küçültüldüğünde sadece bir yüz bulunmaktadır. Çünkü resim 5x5 oranında ölçeklendiğinde diğer yüz 25x25 değerinden küçük olmaktadır ve bu nedenle YSA ile bulunamamaktadır.

2.2.1.4. Yüz Bulgularını Tek Bulguya İndirgeme

Yüz tespitine bir yüze ait birden fazla bulgunun birleştirilmesi ve tek bir bulgu elde edilmesidir. Bunun için, *Çerçeve Birleştirme* algoritması kullanılmıştır. Burada, örtüşen çerçeveler bulunur ve bu çerçevelerin sol üst köşe noktalarının ortalaması alınarak bu ortalamaya en yakın çerçeve seçilir ve sonuç tek bulguya indirgenmiş olur. Çerçeve birleştirme algoritmasına ait C++ kodu aşağıda verilmiştir.

```
void boxmerge() {
....
if((box[a][2]<(2*box[b][2]))&&(box[b][2]<(2*box[a][2]))){
sum=box[a][2]+box[b][2];
```



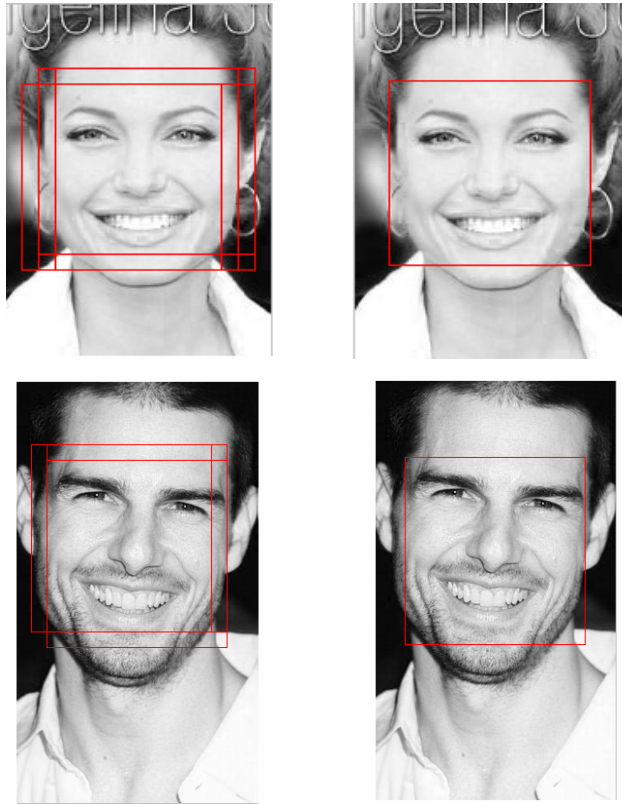
```

dist=sqrt((box[a][1]-box[b][1])*(box[a][1]-box[b][1])+(box[a][0]-
box[b][0])*(box[a][0]-box[b][0]));
if((dist<box[a][2])||(dist<box[b][2])){
... }}}}
...
// çerçevelerin sol koselelerini topla
// ortalamalarını al
// ortalamaya en yakın çerçeveyi seç
.....

```

Yukarıdaki kodda, $box[][0]$, $box[][1]$ ve $box[][2]$ değişkenleri sırasıyla, çerçeveye ait merkez pikselin yatay değerini, merkez pikselin dikey değerini ve çerçevenin en değerinin yarısını göstermektedir.

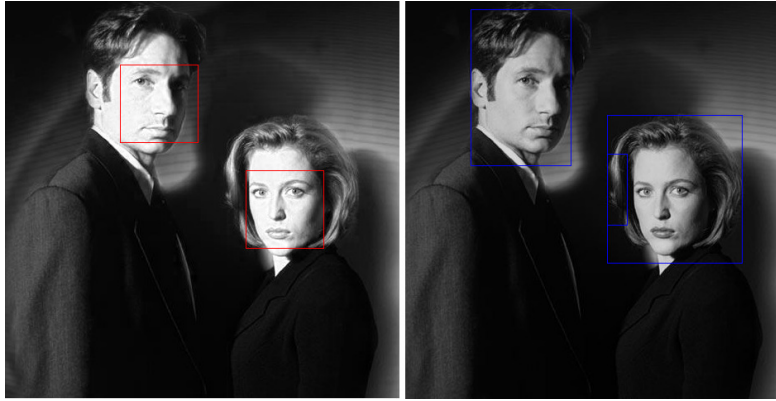
Çerçeve birleştirmeye ilişkin sonuçlar aşağıdaki şekilde gösterilmiştir.



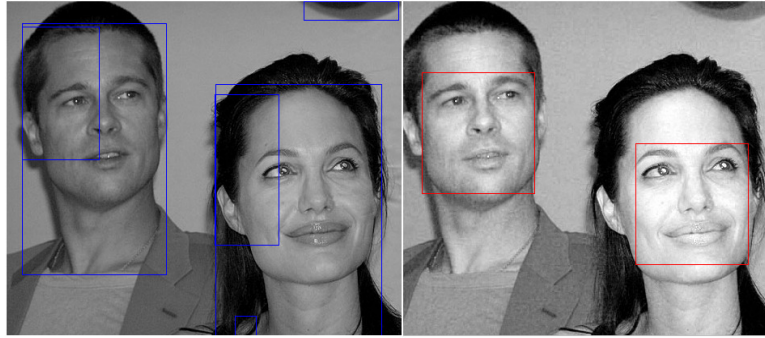
Şekil 30. Çerçeveleme sonuçları

2.2.1.5. Yüz Tespitine İlişkin Sonuçlar

Yapay sinir ağları kullanılmadan, ten bulunmuş resim üzerinde ten bölgeleri etiketlenerek minimum kapsayan çerçeveler ile yüz olarak işaretlenerek yüz tespiti işlemi gerçekleştirilir. Fakat bu yaklaşım ten bölgelerinin tamamen bulunduğu ve el gibi yüze ait olmayan küçük ten bölgelerinin elendiği ve birbirine nerdeyse yapışık gibi çok yakın yüzlerin bulunmadığı resimlerde iyi sonuç üretir. Çalışmada, yapay sinir ağları yaklaşımı ile karşılaştırmak amacıyla bu yaklaşımda denenmiştir. Ve elde edilen sonuçlar aşağıdaki gibidir.



(a)



(b)



Şekil 31. Yöntemlerin karşılaştırılması

Şekil 31’de (a),(b),(c) (Sol) Sadece ten bulma yöntemi ile bulunan yüzler, (Sağ) Ten bulma ile iyileştirilmiş YSA ile bulunan yüzlerdir. Şekilden de görüldüğü üzere yüz tespiti için sadece ten bulma yetersiz kalmaktadır. Bu nedenle, yüz tespiti için ten bulma ile iyileştirilmiş yapay sinir ağları kullanılmıştır.

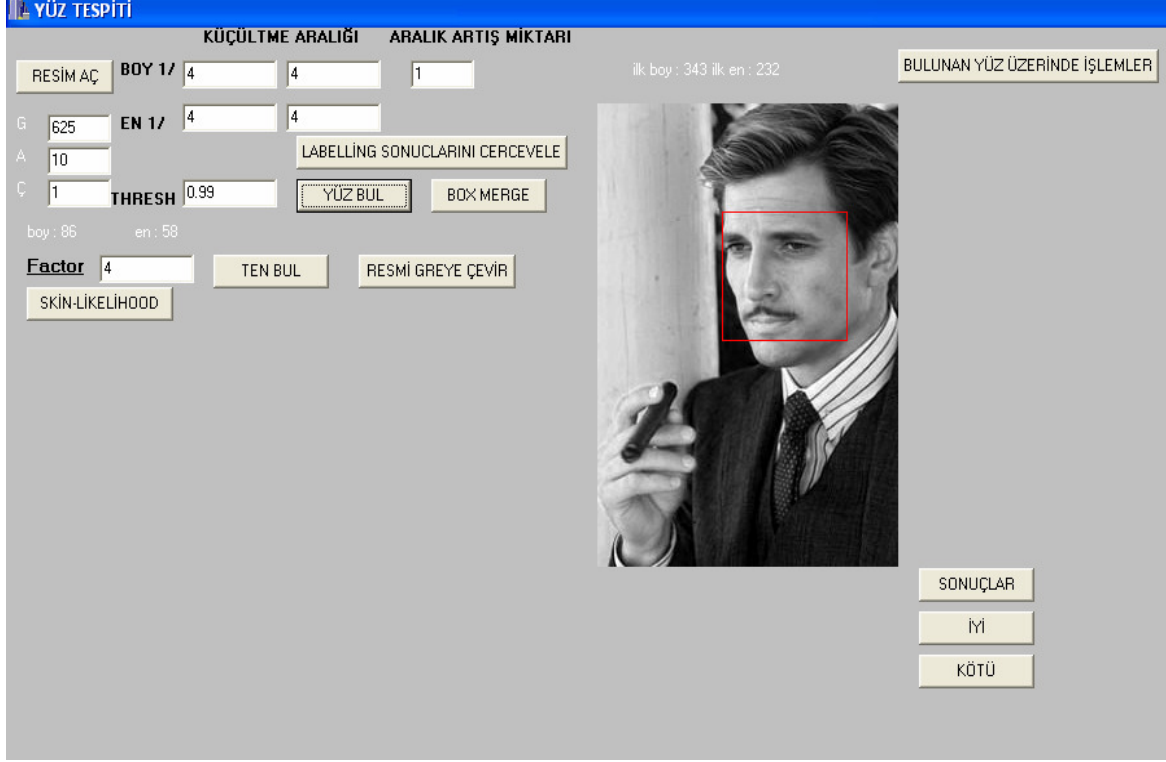
Sonuç olarak, yüz tespiti için önce ten bulma işlemi gerçekleştirilmiştir. Daha sonra resim ten bölgelerini kapsayacak minimum dörtgene daraltılır. Böylece tarama alanı küçültülmüş olur. Daha sonra 25x25 pencereler ile resim taranır, bu pencere en az 300 (yarım yüzler için) ten pikseli içeriyor ise yapay sinir ağlarına verilir. Böylece, ten olmayan bölgeler elemine edilmiş olur ve yanlış çıkabilecek sonuçlar elenmiş olur. Şekil 17’de görüldüğü üzere $\pm 30^\circ$ kadar dönmüş yüzler tespiti edilebilmektedir. Bu kısma ait C++ kodu aşağıda verilmiştir.

```

for(int sat=0;sat<boy-25+1;sat=sat+2) {
    for(int sut=0;sut<en-25+1;sut=sut+2)
    { aa=0; ten=false; tensay=0;
    for(int i=sat;i<sat+25 ;i++)
    for(int j=sut;j<sut+25 ;j++){
    if(Image7->Canvas->Pixels[j][i]!=clWhite)
    tensay++;
    sor[aa]=anares[i][j];
    aa++; }
    if(tensay>=300)
    ten=true;
    if(ten)
    sorgula(sor);
    else
    C[0]=0;
    if(C[0]>thresh){
    ShowMessage("yüz bulundu");
    yuzucervele(sut,sat);
    ...

```

Yukarıdaki kodda, *sor* dizisi ana resim üzerinde gezdirilen 25x25'lik çerçevenin içerdiği piksel değerlerini tutar. *Sorgula* fonksiyonu 25x25'lik girişleri eğitilen ağa vererek çıkış değerini döndürür ve *tensay* değişkeni ise ten piksellerinin sayısını belirtir.



Şekil 32. Yüz tespiti arayüzü



Şekil 33. Yüz Tespiti Sonuçları

Şekil 33'ün devamı



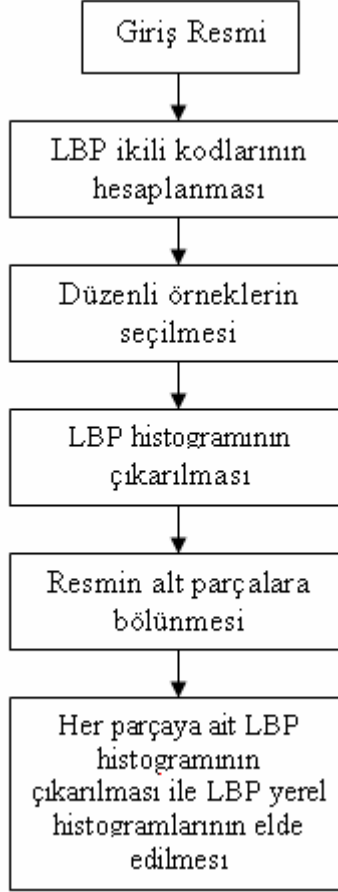
2.2.2. Yüz İfadesi Tanıma

Çalışmada *Gülen, Doğal, Şaşkın, Kızgın ve Üzgün* olmak üzere 5 farklı yüz ifadesi incelenmiş, ayrıca gülme olgusu tam gülen ve gülümseyen olmak üzere iki alt sınıfa parçalanarak tanınmaya çalışılmıştır. Çalışmada, 100x100 boyutlu resimler kullanılmıştır. Yüz ifadesi tanımda küçük boyutlu resimler kullanılırsa ayrıntılar kaçırılabilir ve bu durumda istenmeyen sonuçlar çıkabilir. Bu nedenle, küçük olmamakla beraber yüzü kapsayan resimler kullanılmalıdır.

Yüz ifadelerinin tanınması iki ana kısımdan oluşur:

- 1) *Özellik Çıkarma*
- 2) *Örnek Tanıma*

2.2.1. LBP ile Özellik Çıkarma



Şekil 34. LBP'nin genel yapısı

LBP ile özellik çıkarma işleminde, öncelikle resim üzerinde $n \times n$ komşuluklu pencere gezdirilerek her merkez pikseli için LBP ikili kodu hesaplanır. Yüz ifadesi ve doku tanımda etkili olması sebebiyle çalışmada 3×3 komşuluğu LBP $_{8,1}$ operatörü kullanılmıştır. 3×3 komşuluklar için LBP ikili kodunun hesaplanmasına ait C++ kodu aşağıdaki gibidir.

```

for(int i=1; i<Bitmap2->Height-1; i++)
for(int j=1; j<Bitmap2->Width-1; j++){
b=resim[i][j];
toplam=0;
if(resim[i-1][j-1]>=b) {
g[7]=1;

```

```

toplam+=pow(2,7); }
else {
g[7]=0;
toplam+=0; }
.....
g[8]=g[0];
...

```

Yukarıdaki kodda *resim[][]* dizisi resmin gri seviye değerlerini, *g[]* dizisi ise 3x3 komşuluktaki, komşulara ait ikili değerleri göstermektedir.

Buna göre LBP_{8,1} operatörü için bakıldığında $2^8 = 256$ tane farklı LBP ikili kodu elde edilmiştir. Fakat bütün LBP örnekleri yüz ifadesini iyi tanımlamadığından düzenli örnekler seçilerek kullanılır. Düzenli örneklerin bulunmasına ait C++ kodu aşağıda verilmiştir.

```

uc=0;
for(int h=1; h<9; h++)
uc+=abs(g[h]-g[h-1]);
int sum=0;
if(uc<=2){
for(int h=0; h<8; h++)
sum+=g[h];}
else
sum=9;
lbpu[i][j]=sum; }

```

Yukarıdaki kodda, *uc* örneklerin düzenlilik değerini gösterir ve *lbpu[][]* ise örneklerin düzenlilik değerlerini tutar. Bu şekilde 58 tane düzenli örnek elde edilir. 58 düzenli örneğin dışında kalan örnekler tek bir örnek altında etiketlenir ve böylece 256 yerine 58+1=59 tane özellik vektörü elde edilir. Tüm örnekler etiketlendikten sonra artık tüm resme ait LBP histogramı çıkarılabilir. Gri seviye resme ait formül (24) ile tanımlanmış LBP H_i histogramının hesaplamasına ait C++ kodu aşağıda verilmiştir.

```

for(int i=1; i<Bitmap2->Height-1; i++)

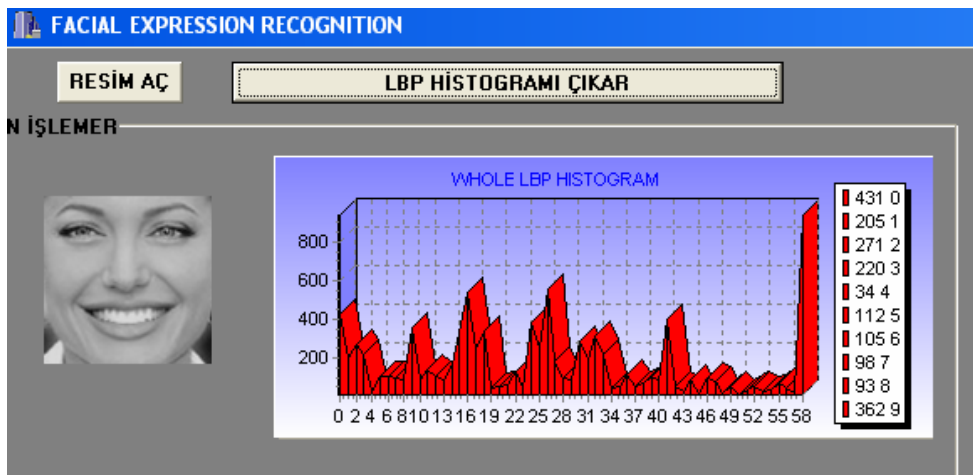
```

```

for(int j=1; j<Bitmap2->Width-1; j++){
  if(lbpu[i][j]!=9){
    for(int v=0; v<59; v++){
      if(labels[v]!=-1){
        if(labels[v]==lbpu[i][j])
          v=59;}
      else {
        labels[v]=lbpu[i][j];
        v=59;} } }
    if(lbpu[i][j]!=9){
      for(int v=0; v<59; v++){
        if(labelcode[v]!=-1){
          if(labelcode[v]==lbpu[i][j])
            v=59;}
        else {
          labelcode[v]=lbpu[i][j];
          v=59; } } } }

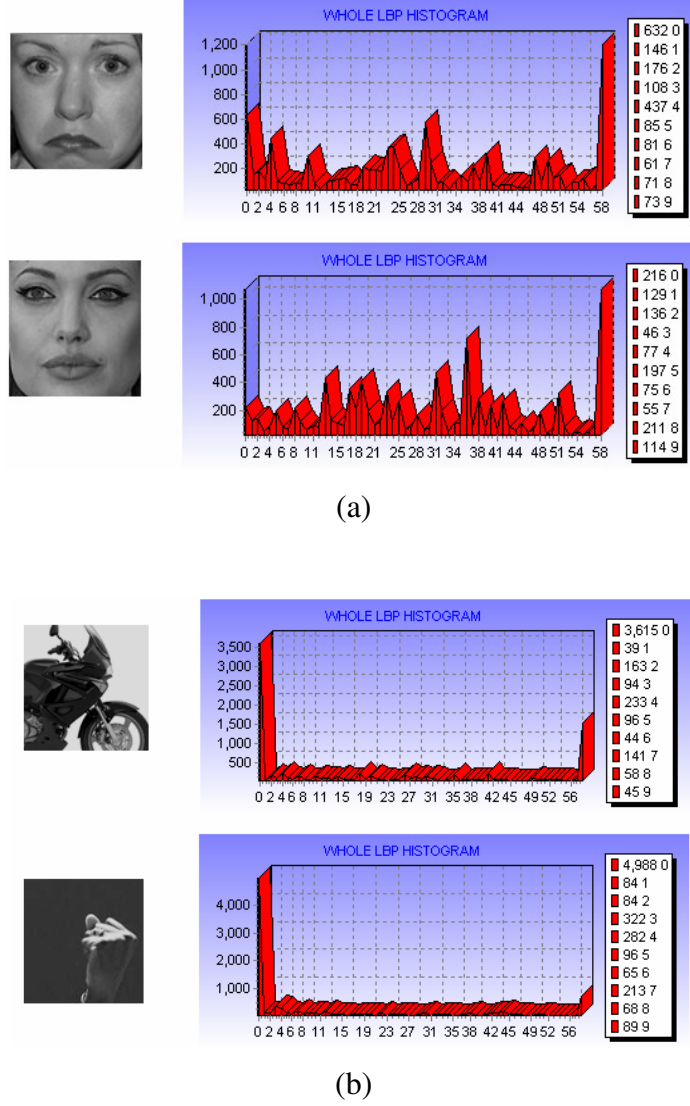
```

Yukarıdaki kodda, *labels[]* dizisi etiketlere ait düzenlilik değerlerini ve *labelcode[]* dizisi ise düzenli örneklerin LBP ikili kod değerlerini tutar.



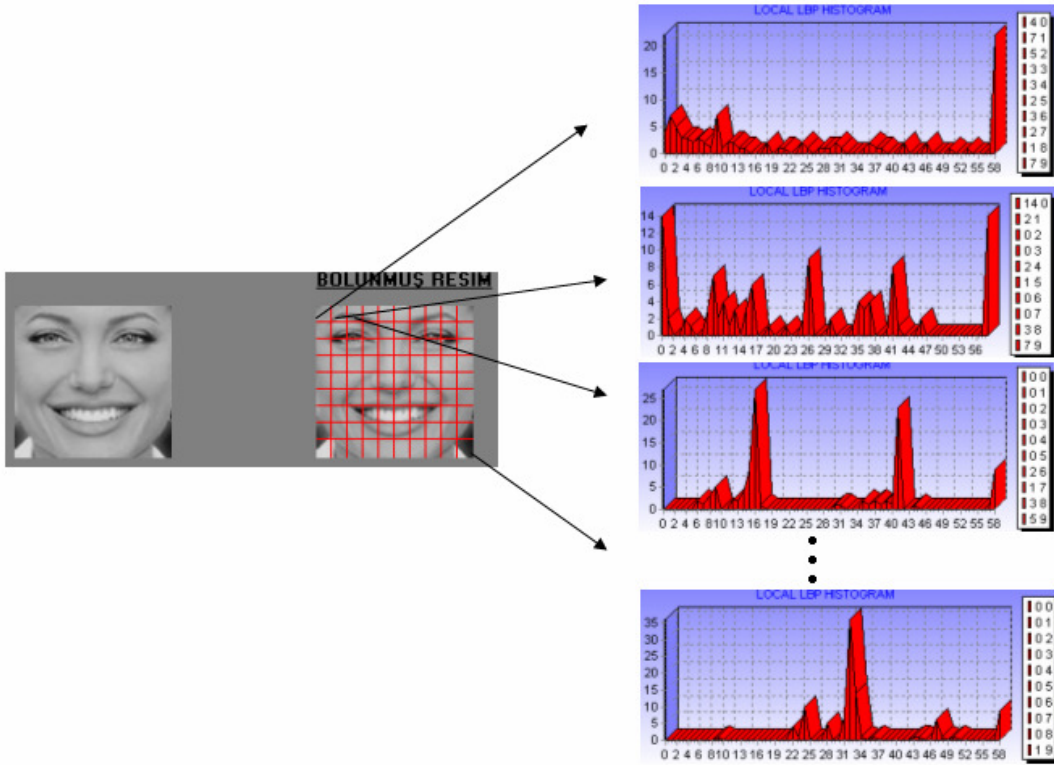
Şekil 35. Yüz resmine ait H_i histogramı

Bu çalışmada görüntülerin H_i histogramları incelendiğinde yüz resimlerinin genelde $H[58]$ 'de, bazı yüz olmayan resimlerin ise $H[0]$, $H[1]$, ... $H[4]$ 'de en yüksek değerlere sahip oldukları gözlemlenmiştir.



Şekil 36. (a) Yüz , (b) Yüz olmayan resimlere ait H_i histogramları

Yerel bölgeler üzerindeki LBP histogramlarının çıkarılması için resim bölgelere bölünmelidir. Bu şekilde, küçük mimiklere ait ayrıntılar da yakalanabilir. Bu nedenle, çalışmada kullanılan 100x100 resimler 10x11 boyutlu alt bölgelere bölünmüştür. Bölümleme işlemindeki boyut aynı zamanda özellik vektörünün boyutunu belirler ve incelemek istediğiniz ayrıntılara da bağlıdır.



Şekil 37. Bölünmüş resim ve yerel bölgelere ait LBP histogramları

Şekil 37’de bölünmüş resimdeki yerel bölgelere ait LBP histogramları gösterilmiştir ve bu bölgelere ait histogramların birleştirilmesi ile yerel LBP histogramları elde edilir. Bu şekilde, 90 bölge elde edildiğinden 59x90 tane özellik elde edilmiştir. Buna göre, yerel LBP histogramları (26) formülü kullanılarak elde edilmiştir. LBP operatörünün bir avantajı da histogramların çok hızlı bir şekilde hesaplanabilmesidir. Yerel histogramların çıkarılmasına ilişkin C++ kodu aşağıda verilmiştir.

```

for(int v=0; v<59; v++)
for(int d=0; d<300; d++)
histo2[v][d]=0;
for(int i=1; i<Bitmap2->Height-1; i++)
for(int j=1; j<Bitmap2->Width-1; j++){
for(int d=0; d<label; d++){
if(bolge[i][j]==d){
if(lbpu[i][j]!=9){
for(int v=0; v<58; v++){

```

```

if(lbp[i][j]==labelcode[v]) {
  histo2[v][d]++;
  v=58;
} } }
else
  histo2[58][d]++;
d=label; } } }

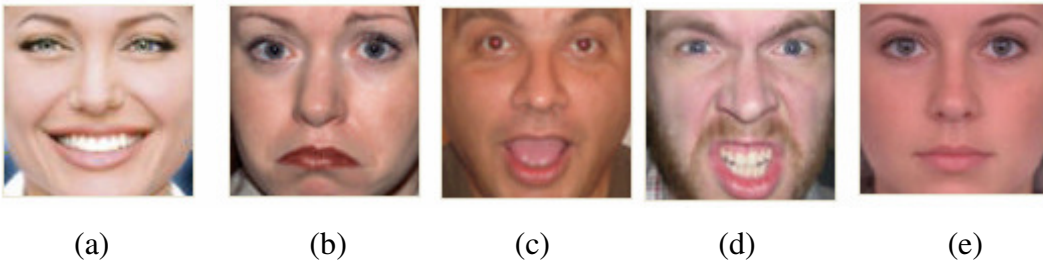
```

Yukarıdaki kodda, *histo2[][]* ve *bolge[][]* dizileri sırasıyla yerel LBP histogramlarını ve bölünmüş resmin alt bölgelerini göstermektedir.

LBP'nin işlem karmaşıklığına bakarsak, $O((M-2)X(N-2))$ ile ifade edilebilir, burada M, resmin en uzunluğunu; N ise boyunu göstermektedir. 2 çıkarılmasının sebebi ise 3x3 çerçeveler ile taranarak merkez piksellerin değerlendirilmesidir.

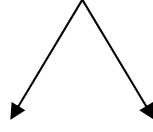
2.2.2. Sınıflayıcılar ile Örnek Tanıma

Yüz ifadelerinin tanınmasında, ilk aşama olan özellik çıkarma işleminin ikinci aşaması örnek tanıma işlemidir. Çalışmada, *Gülme,Doğal, Şaşırmış, Kızgın ve Üzgün* olmak üzere 5 ana ifade incelenmiş ve bununla birlikte, gülme olgusu tam gülen ve gülümseyen olmak üzere iki alt sınıfa parçalanarak tanınmaya çalışılmıştır. Burada öncelikle incelenen yüz ifadelerine ait modellerin belirlenmesi gerçekleşir. Modeller belirlenirken, her ifade için % 100 o ifadeye ait yüzlerin seçilmesi önemlidir. Çünkü sorgulama işleminde modeller ile karşılaştırma yapılacağından, modellerin doğru seçilmesi sistemin daha sağlam sonuçlar üretmesini sağlamaktadır.



Şekil 38. (a) Gülen , (b) Üzgün, (c) Şaşırmış, (d) Kızgın, (e) Doğal , modeli

GÜLME OLGUSU



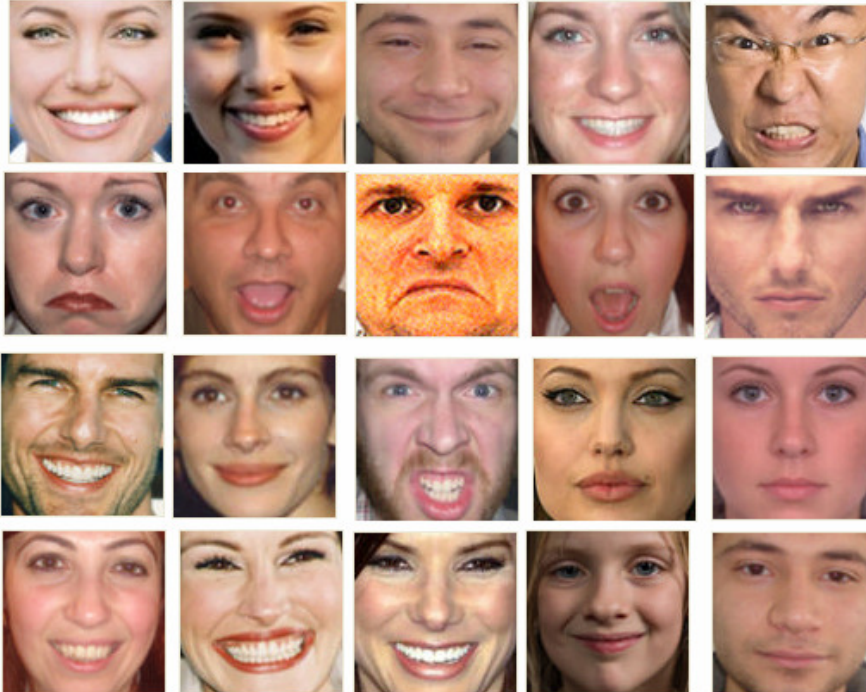
(a)



(b)

Şekil 39. (a) Tam gülen, (b) Gülümseyen, modeli

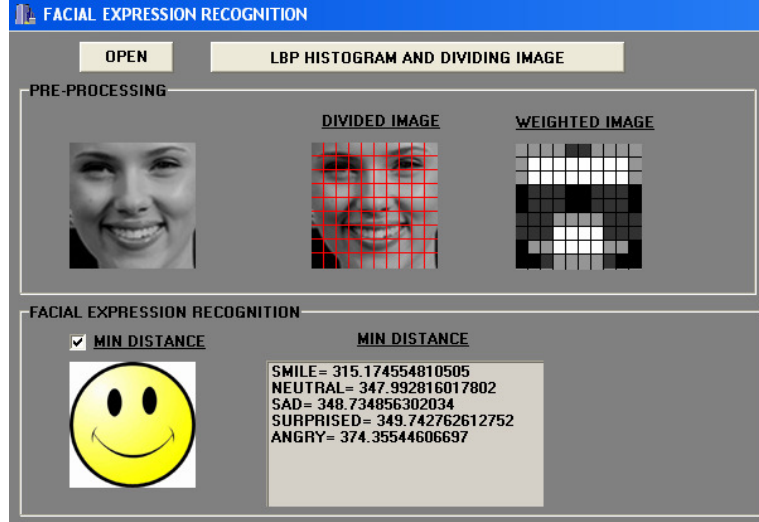
Her yüz ifadesine ait modele uygun eğitim kümeleri oluşturulur ve özellikler çıkarılır. Her ifadeye ait özellik vektörünün oluşturulmasında, o ifadeye ait eğitim kümesindeki yüz resimlerinin yerel LBP histogramlarının ortalaması alınır. Bu şekilde elde edile ortalama yerel LBP histogramı bize o ifadeye ait özellik vektörünü verir.



Şekil 40. Yüz ifadelerine ait veritabanı

2.2.2.1. Minimum Uzaklık ile Sınıflama

Giriş resminin modellere olan uzaklığı bulunur, bu uzaklıklardan minimum değere sahip olan bize o resme ait yüz ifadesini verir.

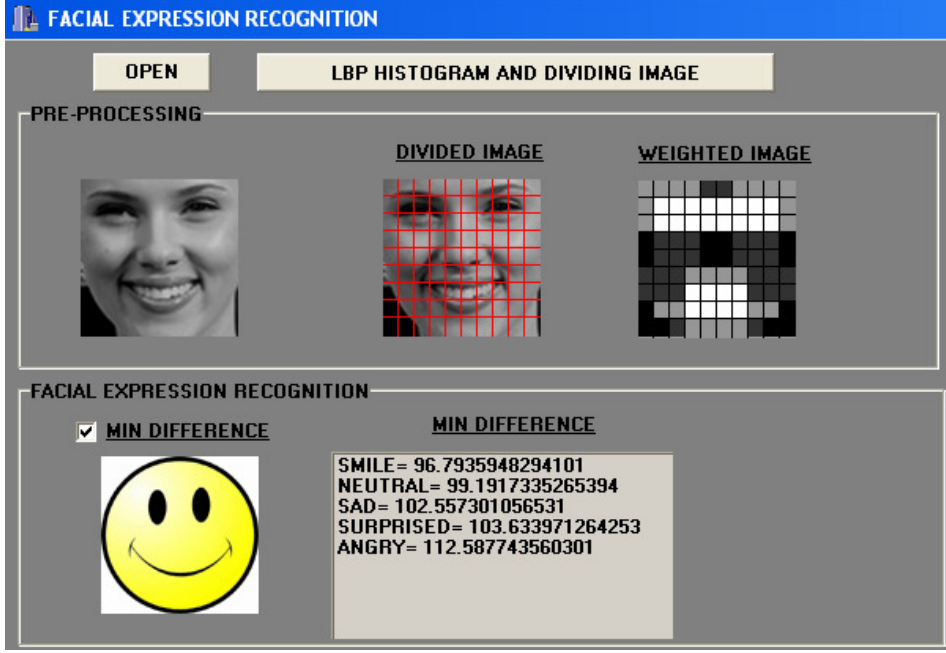


Şekil 41. Minimum uzaklık ile yüz ifadesi tanıma

2.2.2.2. Değiştirilmiş Minimum Uzaklık ile Sınıflama

Burada, minimum uzaklıktaki farkların kareleri yerine mutlak değerleri alınmıştır. Giriş ve model örnekleri arasındaki fark ne kadar az ise o kadar çok yakındırlar mantığı ile farkların mutlak değerleri alınıp toplandığında ve daha sonra da bu toplama kare kökü alma işlemi uygulandığında iyi sonuçlar elde edilmiştir.

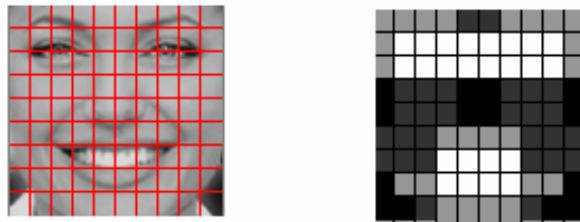
$$D^2 = \sum_{ij} |S_{ij} - M_{ij}| \quad (34)$$



Şekil 42. Değiştirilmiş minimum uzaklık ile yüz ifadesi tanıma

2.2.2.3. Ağırlıklandırılmış Chi Karesi İstatistiği ile Sınıflama

Yüze ait ifadelerin, çoğunlukla göz ve ağız bölgeleri gibi belli bölgelerde bulunduğunu ve bu bölgelerin yüz ifadesi tanımda daha yararlı bilgiler içerdiği belirtilmiştir. Bu yüzden, her yüz bölgesi için içerdiği bilginin önemine göre bir ağırlık belirlenebilir. Ağırlıklar renkleme ile betimlenir. Siyah kareler ağırlık olarak 4.0, koyu gri kareler 2.0, açık gri kareler 1.0 ve beyaz kareler 0.0 göstermektedir.



Şekil 43. Ağırlıklandırılmış chi karesi modeli

Chi karesi istatistiğine göre sınıflama işlemine ait C++ kodu aşağıdaki gibidir.

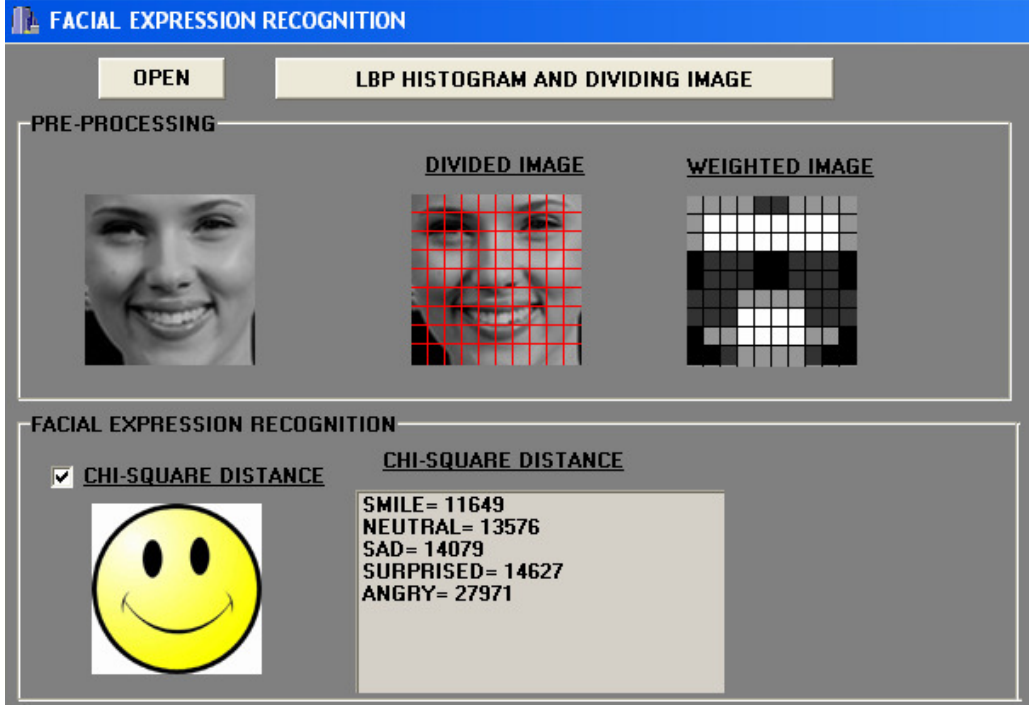
```
for(int j=0; j<label; j++) {
```

```

for(int v=0; v<59; v++) {
if((histo2[v][j]+gulenhisto[v][j])==0)
summ+=0;
else
summ+=(w[j]*(histo2[v][j]-gulenhisto[v][j])*(histo2[v][j]-
gulenhisto[v][j]))/(histo2[v][j]+gulenhisto[v][j]); } }
x2[0]=summ;
for(int j=0; j<label; j++) {
for(int v=0; v<59; v++) {
if((histo2[v][j]+kizginhisto[v][j])==0)
summ+=0;
else
summ+=(w[j]*(histo2[v][j]-kizginhisto[v][j])*(histo2[v][j]-
kizginhisto[v][j]))/(histo2[v][j]+kizginhisto[v][j]); } }
x2[1]=summ;
summ=0;
//sonucları sırala
...

```

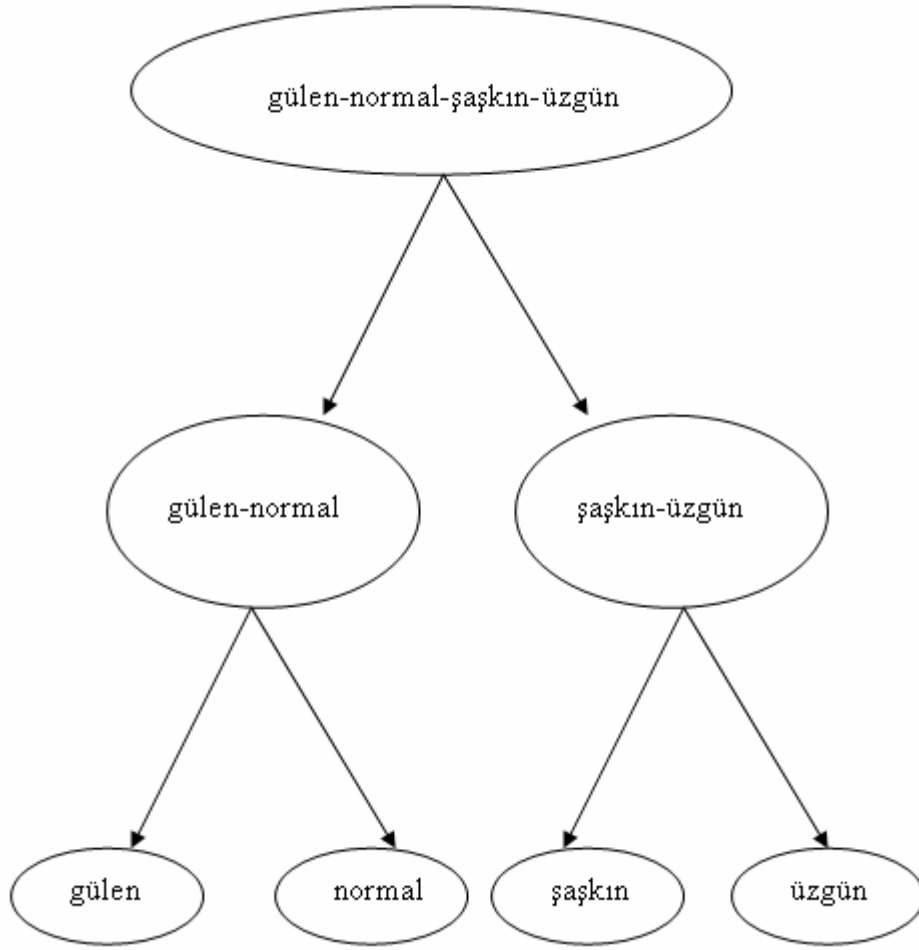
Yukarıdaki kodda, *gulenhisto*, *kizginhisto* dizileri sırasıyla gülen ve kızgın modellerine ait LBP özelliklerini tutmaktadır ve *w* dizisi ise yüz bölgelerine ait ağırlık değerlerini göstermektedir.



Şekil 44. Chi karesi istatistiği ile yüz ifadesi tanıma

2.2.2.4. Yapay Sinir Ağları ile Sınıflama

Çalışmada, yapay sinir ağları ile sınıflama işleminde 25x25 resimler kullanılmıştır. Ara katman sayısı olarak 10, 20, 30 değerleri denenmiş fakat eğitim işlemi gerçekleştirilememiştir. Ara katman sayısı 40 alınarak eğitim işlemi gerçekleşmiştir. Momentum ve öğrenme katsayısı değerleri sırasıyla 0.5 ve 0.1 olarak alınmıştır. Burada da yüz tespiti işleminde kullanılan çok katmanlı ağ ile sınıflama işlemi gerçekleşmiştir. Sadece ara katman sayısı ve çıkış sayıları farklıdır. Burada yüz ifadeleri ikili gruplanarak sınıflama yapılmıştır. Çıkış sayısı 2 olarak alınmıştır.



Şekil 45. YSA ile sınıflama diyagramı

Yüz ifadeleri gruplanarak sınıflama yapıldığından YSA eğitim işleminin yükü azaltılmıştır ve sınıflama işlemi daha hızlı bir biçimde gerçekleşmiştir.

```

C:\PROGRA-1\XINOXS-1\JCREAT-1\GE2001.exe
giris sinir sayisi 625
ara sinir sayisi 40
cikis sayisi 2
agirliklar dosyadan okunacaksa 1, rasgele atanacaksa 2 girin : 1
hesaplamadan sonra sorgulama yapilacaksa 1, yapilmayacaksa 2 girin : 2
ornek sayisi 166
toplanhata0.9926539366781224
toplanhata0.9926332120037641
toplanhata0.9926124667782451
toplanhata0.9925916474136266
toplanhata0.9925707524763867
toplanhata0.9925497806156024
toplanhata0.992528730547199
toplanhata0.9925076010841616
toplanhata0.9924863910546802
toplanhata0.9924650993713734
toplanhata0.9924437249882871
toplanhata0.992422266903549
toplanhata0.9924007241549685
toplanhata0.9923790958160612
toplanhata0.9923573809924505
toplanhata0.992335788186086
toplanhata0.9923136884549029
toplanhata0.9922917090849179
toplanhata0.9922696399130259
toplanhata0.9922474801621789
toplanhata0.9922252290719045
toplanhata0.9922028858964861
toplanhata0.992180449903304
toplanhata0.992157920371331
toplanhata0.9921352965897601
toplanhata0.992112577856757
toplanhata0.9920897634783254
toplanhata0.992066852767271
toplanhata0.9920438450422584
toplanhata0.9920207396269528
toplanhata0.9919975358492387
toplanhata0.9919742330405026
toplanhata0.9919508305349856
toplanhata0.9919273276691885
toplanhata0.9919037237813312
toplanhata0.991880018210858
toplanhata0.9918562102979893
toplanhata0.9918322993833087
toplanhata0.9918082848073895
toplanhata0.9917841659104527
toplanhata0.9917599420320543

```

Şekil 46. Yüz ifadesi eğitim arayüzü

ARTIFICIAL NEURAL NETWORK

NEURAL NETWORK RESULTS

GÜLEN ÇIKIŞ DEĞERİ : 0.998203915657071
 NORMAL ÇIKIŞ DEĞERİ : 0.00179609434292919
 SONUÇ : GÜLÜYÖR

TEST PROCESSING

REDUCTION

TEST1 FOR SMILE-NEUTRAL GROUP OR SURPRISED-SAD GROUP

IF SMILE-NEUTRAL GROUP ELSE IF SURPRISED-SAD GROUP

TEST2 SMILE OR NEUTRAL TEST2 SURPRISED OR SAD

SHOW THE RESULT

TRAINING PROCESSING

1) ADD SMILE TRAINING SET

ADD SMILE-NEUTRAL TRAINING SET

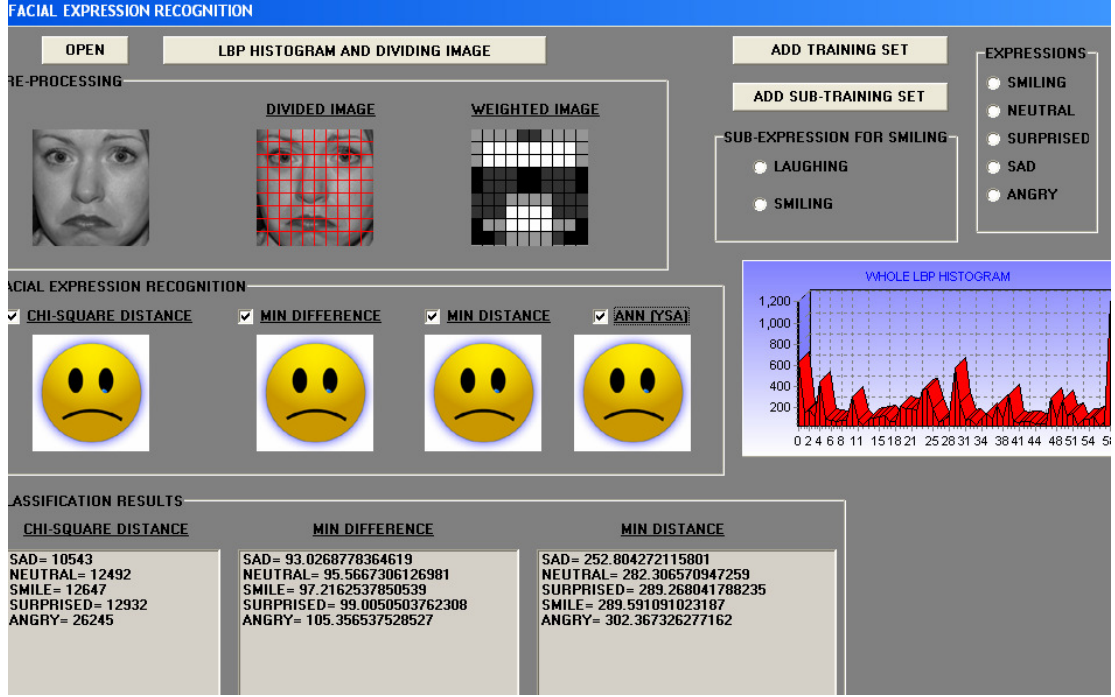
2) ADD NEUTRAL TRAINING SET

1) ADD SURPRISED TRAINING SET

ADD SURPRISED-SAD TRAINING SET

2) ADD SAD TRAINING SET

Şekil 47. YSA ile yüz ifadesi sorgulama arayüzü



Şekil 48. Yüz ifadesi tanıma arayüzü

Tüm sınıflandırma yöntemleri birleştirilerek yüz ifadesi tanıma arayüzü oluşturulmuştur ve yukarıdaki şekilde gösterilmiştir.

3. SONUÇLAR VE ÖNERİLER

Bir yüze ait birden fazla ifade olabilir. Bu durumlar da sınıflama algoritmaları farklı sonuçlar verebilir. Bu sonuçların birleştirilerek yüzdeli olarak ifade edilmesi ile sistem daha kararlı ve sağlam duruma getirilebilir. Bu nedenle, sınıflama algoritmalarının birleştirilmesi için sezgisel bir yaklaşım izlenmiştir. Bu yaklaşımda, her sınıflama sonucu 0-1 değerleri arasına normalize edilir. Yüz ifadeleri %100 üzerinden değerlendirileceğinden, sınıflama algoritmalarına doğruluk değerlerine göre puanlar dağıtılır. Bu puan değerleri normalize edilmiş sonuçlar ile çarpılarak sağlam sonuçlar elde edilmiştir. Buna göre sezgisel değerlendirme formülü aşağıdaki şekilde ifade edilebilir.

$$\sum_{i=0}^n K_i * P_i \quad (35)$$

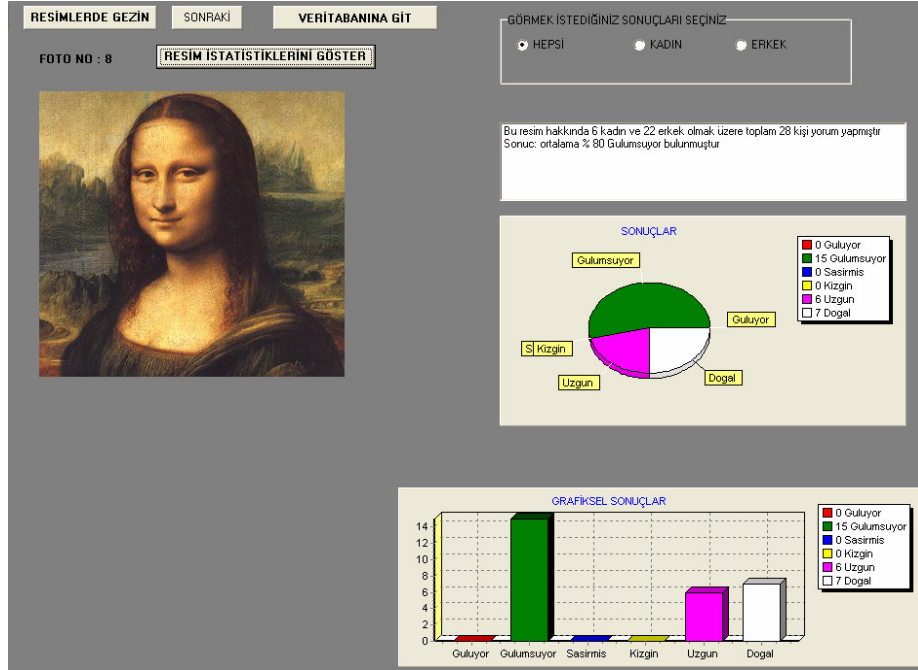
Burada, n sınıflandırma algoritmalarının sayısını, K_i sınıflandırma algoritmasına ait normalize edilmiş sonucu ve P_i ise sınıflandırmaya ait yüzde ağırlıkları göstermektedir. Çalışmada 4 tane sınıflandırma yöntemi kullanılmıştır. Bu yöntemlere ait puanlamalar sırasıyla; Chi karesi İstatistiği %30, minimum uzaklık %20, değiştirilmiş minimum uzaklık %20 ve YSA %30 şeklindedir. Bu şekilde yüze ait daha sağlam sonuçlar elde edilmiştir.

Sistemin doğruluğunun ölçülmesi amacıyla insanlardan da psikolojik değerlendirme sonuçları alınmıştır. Bunun için bir veritabanı programı hazırlanmıştır ve kişiler resimlere ait psikolojik değerlendirme sonuçlarını girmişlerdir.

DURUM	% (YÜZDESİ)
<input type="checkbox"/> GÜLÜYÖR	<input type="text"/>
<input type="checkbox"/> GÜLÜMSÜYÖR	<input type="text"/>
<input type="checkbox"/> ŞAŞIRMİŞ	<input type="text"/>
<input type="checkbox"/> KIZGIN	<input type="text"/>
<input type="checkbox"/> ÜZGÜN	<input type="text"/>
<input type="checkbox"/> DOĞAL	<input type="text"/>

Şekil 49. Kişi psikolojik değerlendirme veri girişi arayüzü

Şekil 49’da gösterilen arayüz kullanılarak kişilerden veriler alınmıştır.



Şekil 50. Sorgulama arayüzü

Sorgulama sayfası kullanılarak istenilen resme ait yapılan değerlendirme sonuçları grafiksel olarak Şekil 50’deki gibi görülebilir.

Bu resim hakkında 6 kadın ve 22 erkek olmak üzere toplam 28 kişi yorum yapmıştır
Sonuç: ortalama % 80 Gulumssuyor bulunmuştur

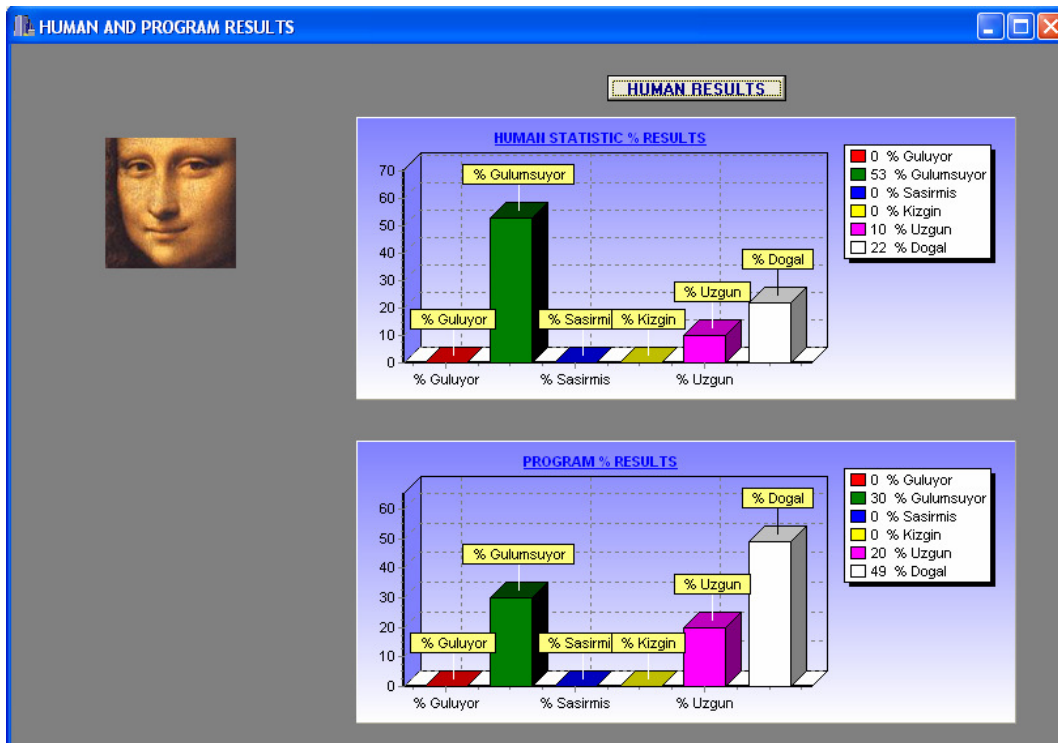
Cinsiyet	Fotono	Gulen	Gulumssuyor	Saskin	Kizgin	Uzgün	Dogal	Sonuc
Kadın	8	0	40	0	0	60	0	Uzgün
Erkek	8	0	90	0	0	0	0	Gulumssuyor
Kadın	8	0	10	0	0	30	40	Dogal
Kadın	8	0	20	0	0	40	30	Uzgün
Erkek	8	0	90	0	0	0	0	Gulumssuyor
Kadın	8	0	18	0	0	0	25	Uzgün
Erkek	8	0	90	0	0	0	0	Gulumssuyor
Erkek	8	0	50	0	0	0	0	Gulumssuyor
Erkek	8	0	90	0	0	0	0	Gulumssuyor
Erkek	8	0	0	0	0	0	100	Dogal
Erkek	8	0	90	0	0	0	0	Gulumssuyor
Erkek	8	0	40	0	0	50	0	Uzgün
Erkek	8	0	30	0	0	50	0	Uzgün
Erkek	8	0	90	0	0	0	0	Gulumssuyor
Erkek	8	0	60	0	0	0	40	Gulumssuyor
Erkek	8	0	90	0	0	0	0	Gulumssuyor
Erkek	8	0	5	0	0	10	80	Dogal
Erkek	8	0	90	0	0	0	0	Gulumssuyor
Erkek	8	0	15	0	0	0	30	Dogal
Erkek	8	0	10	0	0	30	20	Uzgün
Erkek	8	0	90	0	0	0	0	Gulumssuyor
Erkek	8	0	0	0	0	0	85	Dogal

Şekil 51. Resme ait veritabanının görüntülenmesi

Şekil 51’de görüldüğü gibi istenilen zamanda resme ait veritabanı bilgilerine erişilebilmektedir.

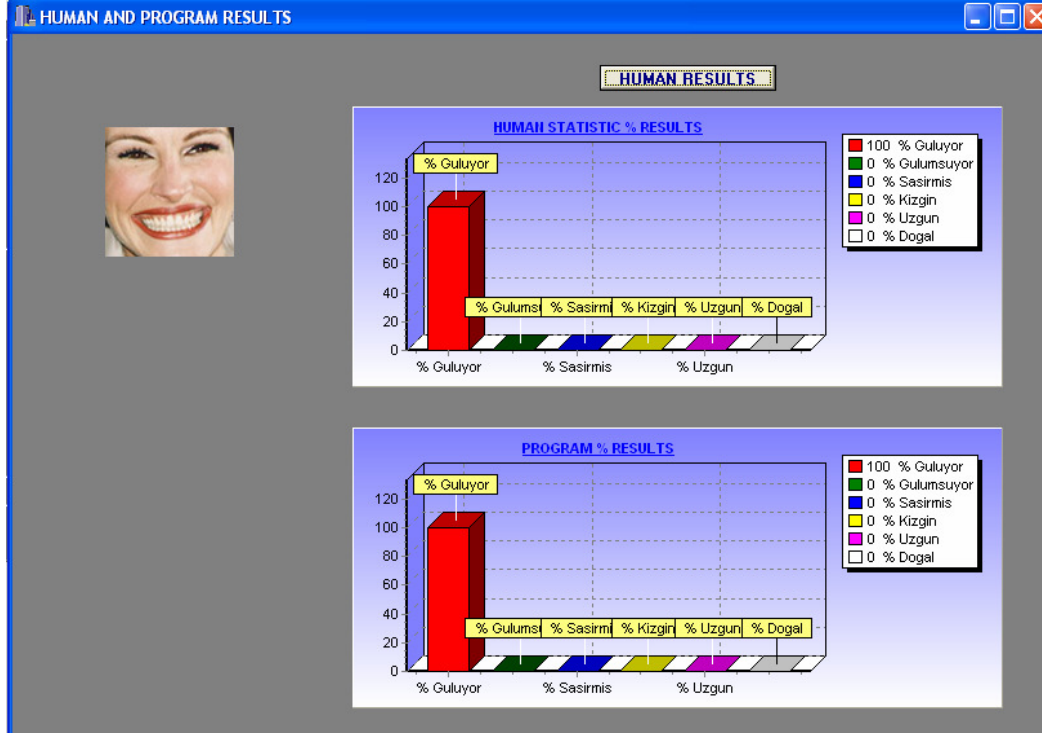
Veritabanında 45 resme ait toplam 800 kayıt bulunmaktadır. Ortalama olarak her kişinin her resme ait değerlendirme yaptığı varsayılırsa bir resimle ilgili yaklaşık 18 kişinin yorum yaptığı sonucuna varılabilir. Veriler incelendiğinde bazı resimlere ait kişiler arasında bile farklılıklar görüldüğü gözlemlenmiştir. Yüz ifadesi gibi psikolojik durumun değerlendirilmesinin, insanlar tarafından bile bazen kolay olmadığı gözlemlendiğinde, bilgisayar ile bu işlemin gerçekleşmesinin ne derecede zor olduğu anlaşılmaktadır.

Yüz ifadesinin tanınmasında elde edilen deneysel sonuçlar ile program sonuçları karşılaştırıldığında sistemin insana benzer sonuçlar verdiği görülmüştür.



Şekil 52. Programa ve insanlara ait değerlendirme sonuçları

Şekil 52'nin devamı



Yüz ifadelerinin tanınmasında kullanılan yöntemler karşılaştırıldığında aşağıdaki tablolar elde edilmiştir.

Tablo1.Geometrik tabanlı TAN ve kullanılan LBP tabanlı model eşleme metotlarının karşılaştırılması

Metotlar (Özellik çıkarma + Sınıflama)	Tanım Sonuçları
LBP + Model eşleme	% 79.1
Geometrik özellik + Ağaç artırılmış saf Bayes (Tree-Augmented-Naive-Bayes, TAN) [51]	% 73.2

Tablo 2. Yüz ifadesi tanıma metotlarının karşılaştırılması

Metotlar	Kısıtlamalar	Tanım Sonuçları
HLAC + Fisher ağırlık haritaları [8]	Yüzü el ile kesmek	% 69.4
Wavelet + PCA + LDA [9]	34 el ile seçilmiş noktalar	% 75
LBP + Ağırlıklandırılmış chi karesi istatistiği	Yüzü modele uygun kesmek	% 77

Tablo 3. Kullanılan metot ile diğer güçlü metotların karşılaştırılması

Metotlar	Tanıma Sonuçları		
	Lineer	Polinom	RBF
LBP + SVM [50]	% 87.2	% 88.4	% 87.6
Gabor + SVM [52]	% 84.8	Lineer ve RBF'den daha kötü	% 86.9
LBP + ağırlıklandırılmış chi karesi, min. uzaklık, değiştirilmiş min. uzaklık, YSA (önerilen metot)	% 89		

Tablo sonuçlarından da görüldüğü gibi LBP yüz özelliklerini tanımlamada oldukça etkili bir yöntemdir ve Gabor ile kıyaslandığında çok boyutlu uzayda daha az yer işgal eder ve dolayısıyla işlem karmaşıklığı daha azdır. Sınıflama metotları incelendiğinde SVM oldukça etkilidir, daha sonra model eşleme tabanlı metotlar gelir. Çalışmada 4 sınıflama metodunun birleştirilmesi ile güçlü bir yüz ifadesi tanıma sistemi geliştirilmiştir.

Yüz tespitinde ten bulma işleminin önemi büyüktür. Yanlış sonuçların elenmesini sağlayarak sistemi güçlendirmiştir.



(a)

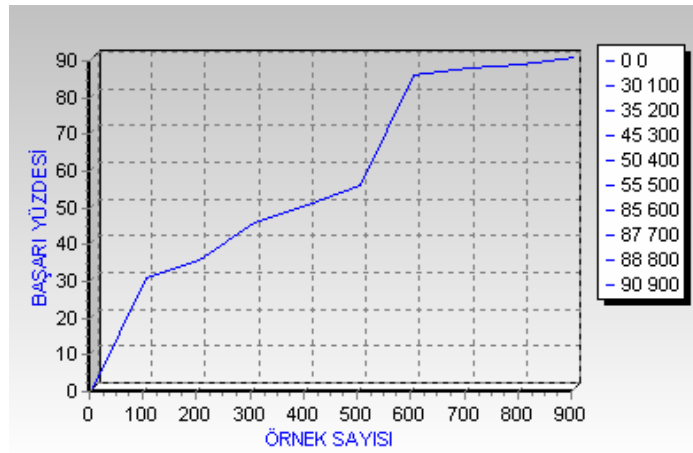
Şekil 53. (a) Ten bulma + YSA ile yüz tespiti, (b) YSA ile yüz tespiti

Şekil 53'ün devamı



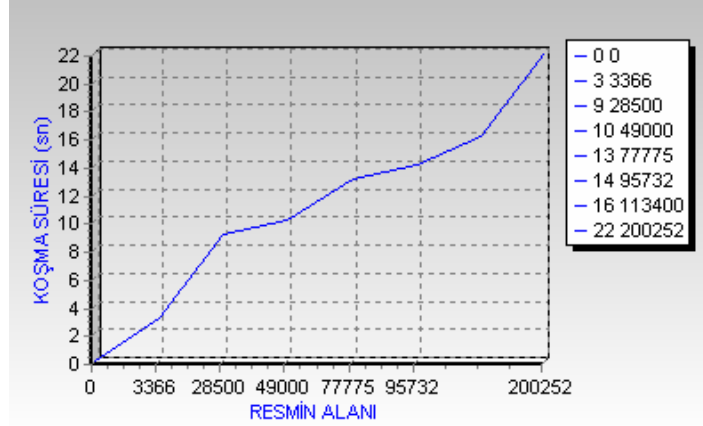
(b)

Yapay Sinir Ağları'ndan iyi sonuç alınabilmesi için eğitim çok önemlidir. Özellikle örneklerin iyi seçilmesi, ağın da iyi öğrenmesini sağlar. Bununla birlikte, eğitimde kullanılan örnek sayısı da etkilidir. Ne kadar çok öğretirseniz o kadar iyi öğrenebilir fakat belli bir örnek sayısından sonra öğrenme sonuçları pek değişmeyebilir hatta belli bir değerden sonra ağ artık iyi ayırt edemez ve kötü sonuçlar vermeye başlayabilir. Çalışmada, yüz tespiti için eğitilen örnek sayısı ve başarıya ilişkin grafik aşağıdaki şekilde verilmiştir.

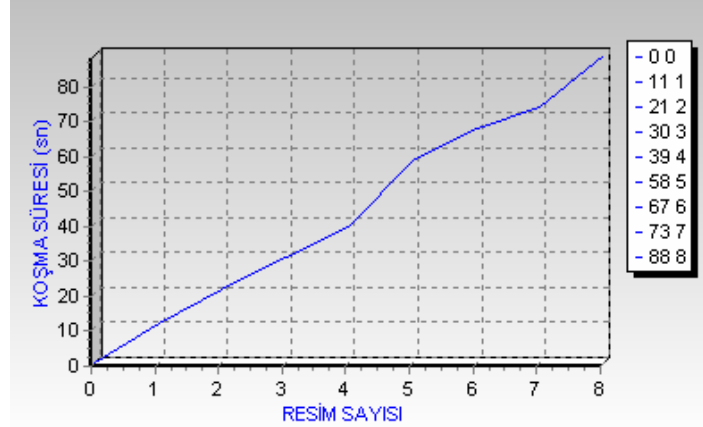


Şekil 54. Yüz tespitine ilişkin eğitilen örnek sayısına göre başarı yüzdesi

Farklı boyutlardaki tek yüz içeren resimler için yüz tespiti koşma süreleri tutulmuştur ve aşağıdaki grafik elde edilmiştir.



Şekil 55. Yüz tespiti için koşma süreleri

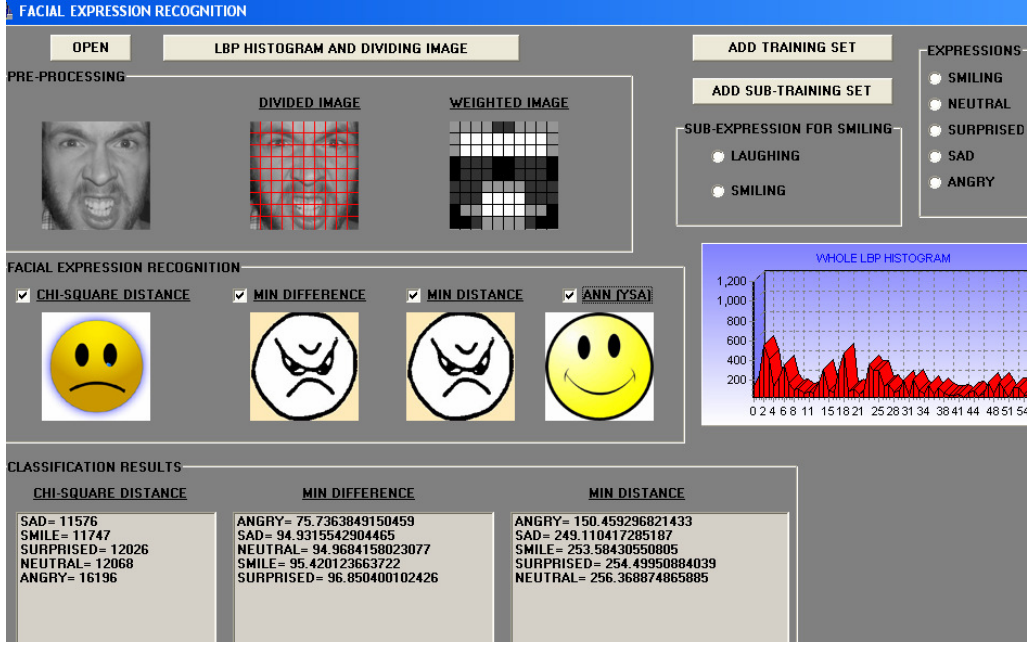


Şekil 56. Yüz ifadesi tanıma için koşma süreleri

Şekil 58'de yüz resimlerine ait yüz ifadelerinin tanınması için geçen süreler tutulmuştur. Resim sayısı ile koşma süreleri arasındaki ilişki grafikte gösterilmiştir. Buna göre, 4 tane resim için koşma süreleri hemen hemen doğrusal bir artışa sahip iken 4-7 resim arasında bir değişim göstermektedir.

Psikolojik durum deęerlendirilmesine dayalı yüz ifadelerinin tanınmasında en küçük ayrıntıların bile büyük önemi olmaktadır. Çünkü bazı ifadeler bizim bile göremediğimiz çok küçük mimikler, çizgiler ile tutulabilmektedir. Bununla birlikte resmin zumu da oldukça etkilidir. Bu insanlar için de geçerlidir, bir kişiye yakından veya uzaktan bakmak arasında fark vardır. Yüz ifadelerinin tanınmasında da bu böyledir. Yüz resminin zumu ile bölgelere düşen pikseller deęişeceğinden sonuçlar da deęişebilmektedir. Ayrıca yüzün kesilme şekli de yine bölgelere düşen pikselleri deęiştireceğinden oldukça etkilidir.

Yüz ifadelerinin bilgisayar ile tanınması oldukça zor bir problemdir. Çünkü ifadeler birbirine çok yakındır ve bu nedenle ayrılması güçtür. Bununla birlikte, ifadelerin kişiler arasındaki farklılığı da göz önüne alındığında karmaşıklığı daha iyi anlaşılmaktadır. Çalışmada kişiye bağımsız psikolojik durum deęerlendirilmesi gerçekleştirilmiştir. *Gülen, Üzgün, Normal, Şaşkın ve Kızgın* olmak üzere 5 ana ifadenin tanınması sağlanmıştır. Ayrıca gülme olgusu tam gülen ve gülümseyen olmak üzere iki alt duruma ayrılarak tanınması gerçekleştirilmiştir. Kızma olgusunun tanınması diğer ifadelere göre daha karmaşık ve zordur. Bu nedenle sınıflayıcılar oldukça önemlidir. Çalışmada, kızma olgusu ağırlıklandırılmış Chi karesi istatistięi ve YSA ile tanınmazken minimum uzaklık ve deęiştirilmiş minimum uzaklık sınıflayıcıları ile tanınması sağlanmıştır. Bu nedenle, bu sınıflayıcılar kızma çıkışı verdiğinde Chi karesi ve YSA sınıflayıcılarının sonuçları dikkate alınmamaktadır.



Şekil 57. Kızma ifadesine ait sonuçlar

Yüz ifadelerinin tanınmasında veritabanı da oldukça önemlidir, özellikle ifadelere ait iyi modellerin seçilmesi gerekmektedir. Veritabanının oluşturulması da zaman alan kısımlardandır. Çalışmada bir çok kişiden farklı ifadelere ait resimler alınmıştır. Yapılan çalışmalarda, elde edilen deneysel sonuçlardan, sistemin insana benzer sonuçlar verdiği görülmüştür.

Tablo 4. Yüz ifadelerine ait tanıma başarıları

Yüz İfadeleri	Tanıma sonuçları
Gülen	%93 (tam gülen %91, gülümseyen %89)
Normal	%92
Üzgün	%93
Şaşkın	%90
Kızgın	%86

4. KAYNAKLAR

1. Schober, O., Beden Dili Davranış Anahtarı, İstanbul. Arion Yayınevi. 1999.
2. Özcan, K., İnsanı Anlamak, İstanbul, Altın Kitaplar Yayınevi. 1997.
3. Robertson, A.K., Etkili Dinleme İstanbul, Hayat Yayınları. 1999.
4. NabiyeV, V.V., “Gülme olgusunun Bilgisayarda Tanınması”, Bilgisayar Mühendisliği Bölümü, Yapay Zeka Uygulamaları Grubu, Karadeniz Teknik Üniversitesi. 2006.
5. Ying-li, T., Takeo, K. ve Jeffrey, F.C., Facial Expression Analysis, Handbook of Face Recognition, Springer, Ekim 2003.
6. Zhang, Z., Lyons, M., Schuster, M. ve Akamatsu, S., Comparison Between Geometry-based and Gabor-Wavelet based Facial Expression Recognition Using Multi-layer Perceptron, IEEE FG, Nisan 1998.
7. Ying-li, T., Evaluation of face resolution for expression analysis, Workshop on Face Processing in Video, 2004.
8. Shinohara, Y. ve Otsu, N., Facial Expression Recognition Using Fisher Weight Maps, IEEE Conf. on Automatic Face and Gesture Recognition, (2004) 499-504.
9. Lyons, M., Budynek, J. ve Akamatsu, S., Automatic Classification of Single Facial Images, IEEE Trans. Pattern Analysis and Machine Intelligence, 21 (1999) 1357-1362.
10. Zhang, Z., Feature-based Facial Expression Recognition: Sensitivity Analysis and Experiment with a Multi-layer Perceptron, Pattern Recognition and Artificial Intelligence 13 (1999) 893-911.
11. NabiyeV, V.V., Yapay Zeka : Problemler, Yöntemler, Algoritmalar, 764 say., Seçkin Yayıncılık, 2. baskı, 2005, Mayıs, Ankara.
12. Phil, B., Face Detection using Neural Networks, Meng Electronic Engineering School of Electronics and Physical Sciences, 2004.
13. Sakai, T., M., Computer analysis and classification of photographs of human faces , Nagao and T. Kanada, 1972.
14. Yale face database, <http://cvc.yale.edu/projects/yalefaces/yalefaces.html>, 17 Ocak 2007.
15. Kass, M., Witkin, A. ve Terzopoulos, D., Snakes: active contour models. In Proc. of 1st Int Conf. on Computer Vision, Londra, 1987.

16. Cootes, T.F. ve Taylor, C.J., Active shape models—‘smart snakes’. In Proc. Of British Machine Vision Conference, 1992, 266-275.
17. Kim, I., Shim, J.H. ve Yang, J., Face Detection, Stanford University, http://www.stanford.edu/class/ee368/Project_03/Project/reports/ee368group02.pdf, 12 Kasım 2006.
18. Shin, M.C., Chang, K.I. ve Tsap, L.V., Does Colorspace Transformation Make Any Difference on Skin Detection? IEEE Workshop on Applications of Computer Vision, (2002) 275–279.
19. Vezhnevets, V., Sazonov, V. ve Andreeva, A., A Survey on Pixel-Based Skin Color Detection Techniques. In Proc. Graphicon, 2003, 85–92.
20. Ho, S.Y., Huang, H.L., An analytic solution for the pose determination of human faces from monocular image, Pattern Recognition Letters, Elsevier, 1998.
21. Prof.Dr.Ercan, Ö., Yapay Sinir Ağları, Papatya Yayıncılık, 2003.
22. Rehberlik Hizmetleri Portalı, <http://www.rehberliknet.net>, 2003.
23. Aile Hekimi, İletişim Temeleri, <http://www.ailehekimirehberi.com>, 2006.
24. Bora, B., Abdülkadir, Ç., Somatizasyonun kültürel boyutları, Ankara Üniversitesi Tıp Fakültesi, Psikiyatri AD, 2007.
25. Yüze göre teşhis, <http://www.fizyognomi.com>.
26. Cüceloğlu D., Yeniden İnsan İnsana. İstanbul. Remzi Kitabevi. 2001.
27. Cootes, T.F., Edwards, G.J. ve Taylor, C.J., Active appearance models, Lecture Notes in Computer Science, 1407 (1998) 484-498.
28. Donato, G., Barlett, M.S., Hager, J.C., Ekman, P., and Sejnowski, T.J., Classifying Facial Actions, IEEE Trans. Pattern Analysis and Machine Intelligence, 21,10 (1999) 974-989.
29. Turk, M., Pentland, A., Eigenfaces for recognition, Journal of Cognitive Neuroscience, no.3 (1991) 71-86.
30. Feng, X., Hadid, A. ve Pietikainen, M., A Coarse-to-Fine Classification Scheme for Facial Expression Recognition, ICIAR Proceedings, Lecture Notes in Computer Science 3213, Springer, 2 (2004) 668-675.
31. Stan, Z.L., ChunShui, Z., Meng, A. ve Zhen, L., Learning to Fuse 3D+2D Based Face Recognition at Both Feature and Decision Levels , Center for Biometrics and Security Research & National Laboratory of Pattern Recognition Institute of Automation, Chinese Academy of Sciences, 2005 .

32. Anandan, P., A Computational Framework and an Algorithm for the Measurement of Visual Motion, International Journal of Computer Vision, 2 (1989) 283-310.
33. Teo, W.K., Liyanage, C.S. ve Prahlad, V., Facial expression detection and recognition system, Journal of The Institution of Engineers, Singapore, 2004.
34. Luana, B.B. ve Herman, M.G., Photogenic Expression Recognition using Gabor Filters and Support Vector Machine, Universidade Federal de Campina Grande, Departamento de Sistemas e Computação, 2005.
35. Ojala, T., Pietikainen, M. ve Harwood, D., A comparative study of texture measures with classification based on feature distributions, *Pattern Recognition* 29 (1996) 51–59.
36. Ahonen, T., Hadid, A. ve Pietikainen, M., Face recognition with local binary patterns. Proceedings of the European Conference on Computer Vision, Prague, Czech (2004) 469–481.
37. Hadid, A., Pietikainen, M. ve Ahonen, T., A discriminative feature space for detecting and recognizing faces, Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2 (2004) 797–804.
38. Mäenpää, T., Ojala, T., Pietikäinen, M. ve Soriano, M., Robust Texture Classification by Subsets of Local Binary Patterns, Machine Vision and Media Processing Unit Infotech Oulu and Department of Electrical Engineering P.O.Box 4500, FIN-90014 University of Oulu, Finland, 2000.
39. Ojala, T., Pietikäinen, M. ve Mäenpää, T., Gray Scale and Rotation Invariant Texture Classification with Local Binary Patterns, *Proc. ECCV 2000*, in press.
40. Ojala, T., Pietikainen, M. ve Menp, T., Multi-resolution grayscale and rotation invariant texture classification with local binary patterns, IEEE PAMI, 24,7 (2002).
41. Caifeng, S., Shaogang, G. ve Peter, W.M., Conditional Mutual Information Based Boosting for Facial Expression Recognition, Queen Mary, University of London, Londra, 2005.
42. Carl, G.L., *Pattern Recognition* , University of Nevada ,Reno, Nevada, 2003.
43. Looney, C., *Pattern Recognition Using Neural Networks*, Oxford University Press, New York, 1997.
44. Forgy, E., Cluster analysis of multivariate data: efficiency versus interpretability of classifications, *Biometrics*, 21 (1965) 768–776.
45. MacQueen, J. B., Some methods for classification and analysis of multivariate observations, *Proc. 5th, Berkeley Symp. Probability Statistics*, University of California Press, Berkeley, (1967) 281–297.
46. Philipp, M. ve Rana, E.K., Facial Expression Recognition Using Support Vector

Machines, Computer Laboratory University of Cambridge, Cambridge, 2003.

47. Freund, Y. ve Schapire, R.E., A decision-theoretic generalization of online learning and an application to boosting. *J. Comp. & Sys. Sci.*, 55,1 (1997) 19-139.
48. Tieu, K. ve Viola, P., Boosting image retrieval, in Proc. Computer Vision and Pattern Recognition Conf., 1 (2000) 228–235.
49. Feng, X., Facial Expression Recognition Based on Local Binary Patterns and Coarse-to-Fine Classification, College of Electronics and Information, Northwestern Polytechnic University, Xi' an, 710072, Çin, 2004.
50. Shan, C., Shaogang, G. ve Peter, W.M., Robust Facial Expression Recognition Using Local Binary Patterns, Queen Mary, University of London, Londra, 2005.
51. Cohen, I., Sebe, N., Garg, A., Chen, L. ve Huang, T., Facial expression recognition from video sequences: Temporal and static modeling, *CVIU*, 91 (2003) 160-187.
52. Bartlett, M.S., Littlewort, G., Fasel, I. ve Movellan, R., Real time face detection and facial expression recognition: Development and application to human computer interaction, in *CVPR Workshop on CVPR for HCI*, 2003.

5. EKLER

Database Desktop - [Table : durum.DB]									
File Edit View Table Record Tools Window Help									
durum	Cinsiyet	Fotono	Gulen	Gulumseyen	Saskin	Kizgin	Uzgun	Dogal	Sonuc
1	Erkek	45.00	0.00	30.00	0.00	0.00	0.00	50.00	Dogal
2	Erkek	45.00	0.00	20.00	0.00	0.00	0.00	60.00	Dogal
3	Kadin	45.00	0.00	25.00	0.00	0.00	0.00	60.00	Dogal
4	Kadin	45.00	0.00	40.00	0.00	0.00	5.00	60.00	Dogal
5	Erkek	45.00	0.00	30.00	0.00	0.00	10.00	60.00	Dogal
6	Erkek	45.00	0.00	50.00	0.00	0.00	0.00	50.00	Gulumsuyor
7	Erkek	45.00	0.00	35.00	0.00	0.00	0.00	60.00	Dogal
8	Kadin	45.00	0.00	45.00	0.00	0.00	0.00	40.00	Gulumsuyor
9	Kadin	45.00	0.00	35.00	0.00	0.00	5.00	50.00	Dogal
10	Kadin	45.00	0.00	35.00	0.00	0.00	5.00	40.00	Dogal
11	Erkek	45.00	0.00	25.00	0.00	0.00	10.00	30.00	Dogal
12	Erkek	45.00	0.00	40.00	0.00	0.00	0.00	50.00	Dogal
13	Kadin	45.00	0.00	30.00	0.00	0.00	0.00	50.00	Dogal
14	Erkek	45.00	0.00	40.00	0.00	0.00	15.00	30.00	Gulumsuyor
15	Kadin	45.00	0.00	30.00	0.00	0.00	10.00	40.00	Dogal
16	Kadin	45.00	0.00	0.00	0.00	0.00	0.00	50.00	Dogal
17	Kadin	45.00	0.00	0.00	0.00	0.00	30.00	20.00	Uzgun
18	Kadin	45.00	0.00	30.00	0.00	0.00	0.00	0.00	Gulumsuyor
19	Kadin	44.00	0.00	0.00	0.00	0.00	65.00	0.00	Uzgun
20	Erkek	44.00	0.00	0.00	0.00	0.00	70.00	0.00	Uzgun
21	Erkek	44.00	0.00	0.00	0.00	0.00	90.00	0.00	Uzgun
22	Kadin	44.00	0.00	0.00	0.00	0.00	90.00	10.00	Uzgun
23	Erkek	44.00	0.00	0.00	0.00	0.00	75.00	10.00	Uzgun
24	Kadin	44.00	0.00	0.00	0.00	0.00	85.00	5.00	Uzgun
25	Kadin	44.00	0.00	0.00	0.00	0.00	75.00	0.00	Uzgun
26	Erkek	44.00	0.00	0.00	0.00	0.00	90.00	0.00	Uzgun
27	Erkek	44.00	0.00	0.00	0.00	5.00	80.00	0.00	Uzgun
28	Erkek	43.00	100.00	0.00	0.00	0.00	0.00	0.00	Guluyor
29	Erkek	43.00	100.00	0.00	0.00	0.00	0.00	0.00	Guluyor
30	Kadin	43.00	98.00	0.00	0.00	0.00	0.00	0.00	Guluyor
31	Kadin	43.00	98.00	0.00	0.00	0.00	0.00	0.00	Guluyor
32	Kadin	43.00	100.00	0.00	0.00	0.00	0.00	0.00	Guluyor
33	Kadin	43.00	100.00	0.00	0.00	0.00	0.00	0.00	Guluyor
34	Kadin	43.00	100.00	0.00	0.00	0.00	0.00	0.00	Guluyor

durum	Cinsiyet	Fotono	Gulen	Gulumseyen	Saskin	Kizgin	Uzgun	Dogal	Sonuc
179	Kadin	4.00	0.00	98.00	0.00	0.00	0.00	0.00	Gulumsuyor
180	Kadin	2.00	0.00	0.00	0.00	0.00	90.00	0.00	Uzgun
181	Kadin	1.00	0.00	0.00	80.00	0.00	0.00	0.00	Sasirmis
182	Kadin	1.00	0.00	0.00	80.00	0.00	0.00	0.00	Sasirmis
183	Erkek	9.00	0.00	0.00	0.00	0.00	15.00	85.00	Dogal
184	Erkek	18.00	0.00	0.00	0.00	0.00	0.00	90.00	Dogal
185	Erkek	17.00	0.00	75.00	0.00	0.00	0.00	25.00	Gulumsuyor
186	Erkek	17.00	0.00	100.00	0.00	0.00	0.00	0.00	Gulumsuyor
187	Erkek	16.00	0.00	20.00	0.00	0.00	0.00	80.00	Dogal
188	Erkek	15.00	0.00	100.00	0.00	0.00	0.00	0.00	Gulumsuyor
189	Erkek	14.00	0.00	0.00	0.00	0.00	0.00	100.00	Dogal
190	Erkek	13.00	100.00	0.00	0.00	0.00	0.00	0.00	Guluyor
191	Erkek	12.00	0.00	0.00	0.00	30.00	0.00	70.00	Dogal
192	Kadin	12.00	0.00	0.00	0.00	90.00	0.00	10.00	Kizgin
193	Erkek	11.00	0.00	0.00	0.00	5.00	11.00	80.00	Dogal
194	Kadin	10.00	0.00	90.00	0.00	0.00	0.00	0.00	Gulumsuyor
195	Kadin	9.00	0.00	80.00	0.00	0.00	0.00	20.00	Gulumsuyor
196	Erkek	8.00	0.00	90.00	0.00	0.00	0.00	0.00	Gulumsuyor
197	Kadin	7.00	100.00	0.00	0.00	0.00	0.00	0.00	Guluyor
198	Kadin	6.00	0.00	0.00	0.00	80.00	0.00	10.00	Kizgin
199	Kadin	6.00	0.00	0.00	0.00	10.00	0.00	0.00	Kizgin
200	Kadin	5.00	0.00	0.00	0.00	0.00	0.00	95.00	Dogal
201	Kadin	4.00	0.00	100.00	0.00	0.00	0.00	0.00	Gulumsuyor
202	Kadin	3.00	0.00	0.00	0.00	50.00	0.00	25.00	Kizgin
203	Kadin	2.00	0.00	0.00	0.00	0.00	60.00	0.00	Uzgun
204	Kadin	1.00	0.00	0.00	90.00	0.00	0.00	10.00	Sasirmis
205	Erkek	3.00	0.00	0.00	0.00	98.00	0.00	0.00	Kizgin
206	Erkek	2.00	0.00	0.00	0.00	0.00	70.00	0.00	Uzgun
207	Kadin	1.00	0.00	0.00	90.00	0.00	0.00	0.00	Sasirmis
208	Erkek	1.00	0.00	0.00	0.00	0.00	0.00	0.00	Sasirmis
209	Erkek	21.00	100.00	0.00	0.00	0.00	0.00	0.00	Guluyor
210	Erkek	20.00	0.00	0.00	0.00	0.00	0.00	46.00	Dogal
211	Erkek	19.00	100.00	0.00	0.00	0.00	0.00	0.00	Guluyor

Ek Şekil 1. Psikolojik durum değerlendirme deneysel sonuçlarına ait veritabanı

Ek Şekil 1'in devamı

durum	Cinsiyet	Fotono	Gulen	Gulumseyen	Saskin	Kizgin	Uzgun	Dogal	Sonuc
379	Erkek	12.00	0.00	0.00	0.00	65.00	0.00	0.00	Kizgin
380	Erkek	13.00	95.00	0.00	0.00	0.00	0.00	0.00	Guluyor
381	Erkek	14.00	0.00	0.00	0.00	0.00	0.00	100.00	Dogal
382	Erkek	15.00	0.00	75.00	0.00	0.00	0.00	90.00	Dogal
383	Erkek	18.00	0.00	0.00	0.00	0.00	0.00	75.00	Dogal
384	Erkek	16.00	0.00	34.00	0.00	0.00	0.00	35.00	Dogal
385	Erkek	11.00	0.00	0.00	0.00	0.00	5.00	50.00	Dogal
386	Erkek	8.00	0.00	30.00	0.00	0.00	50.00	0.00	Uzgun
387	Kadin	17.00	0.00	40.00	0.00	0.00	0.00	50.00	Dogal
388	Kadin	17.00	0.00	40.00	0.00	0.00	0.00	50.00	Dogal
389	Kadin	17.00	0.00	40.00	0.00	0.00	0.00	50.00	Dogal
390	Kadin	7.00	90.00	0.00	0.00	0.00	0.00	0.00	Guluyor
391	Kadin	7.00	90.00	0.00	0.00	0.00	0.00	0.00	Guluyor
392	Kadin	2.00	0.00	0.00	0.00	0.00	90.00	0.00	Uzgun
393	Kadin	1.00	0.00	0.00	80.00	0.00	0.00	0.00	Sasirmis
394	Kadin	1.00	0.00	0.00	80.00	0.00	0.00	0.00	Sasirmis
395	Erkek	9.00	0.00	0.00	0.00	0.00	15.00	85.00	Dogal
396	Erkek	18.00	0.00	0.00	0.00	0.00	0.00	90.00	Dogal
397	Erkek	17.00	0.00	75.00	0.00	0.00	0.00	25.00	Gulumsuyor
398	Erkek	17.00	0.00	55.00	0.00	0.00	0.00	0.00	Gulumsuyor
399	Erkek	16.00	0.00	20.00	0.00	0.00	0.00	80.00	Dogal
400	Erkek	15.00	0.00	100.00	0.00	0.00	0.00	0.00	Gulumsuyor
401	Erkek	14.00	0.00	0.00	0.00	0.00	0.00	100.00	Dogal
402	Erkek	13.00	100.00	0.00	0.00	0.00	0.00	0.00	Guluyor
403	Erkek	12.00	0.00	0.00	0.00	30.00	0.00	90.00	Dogal
404	Kadin	12.00	0.00	0.00	0.00	90.00	0.00	90.00	Dogal
405	Erkek	11.00	0.00	0.00	0.00	5.00	0.00	80.00	Dogal
406	Kadin	10.00	0.00	90.00	0.00	0.00	0.00	0.00	Gulumsuyor
407	Kadin	9.00	0.00	80.00	0.00	0.00	0.00	20.00	Gulumsuyor
408	Erkek	8.00	0.00	90.00	0.00	0.00	0.00	0.00	Gulumsuyor
409	Kadin	7.00	100.00	0.00	0.00	0.00	0.00	0.00	Guluyor
410	Kadin	6.00	0.00	0.00	0.00	80.00	0.00	10.00	Kizgin
411	Kadin	6.00	0.00	0.00	0.00	10.00	0.00	0.00	Kizgin
412	Kadin	5.00	0.00	0.00	0.00	0.00	0.00	95.00	Dogal

durum	Cinsiyet	Fotono	Gulen	Gulumseyen	Saskin	Kizgin	Uzgun	Dogal	Sonuc
767	Kadin	5.00	0.00	0.00	0.00	0.00	0.00	90.00	Dogal
768	Kadin	4.00	0.00	100.00	0.00	0.00	0.00	0.00	Gulumsuyor
769	Kadin	3.00	0.00	0.00	0.00	50.00	0.00	0.00	Kizgin
770	Kadin	2.00	0.00	25.00	0.00	0.00	70.00	0.00	Uzgun
771	Kadin	1.00	5.00	0.00	20.00	0.00	0.00	0.00	Sasirmis
772	Kadin	2.00	0.00	0.00	0.00	0.00	90.00	0.00	Uzgun
773	Kadin	1.00	0.00	0.00	80.00	0.00	0.00	0.00	Sasirmis
774	Kadin	1.00	0.00	0.00	80.00	0.00	0.00	0.00	Sasirmis
775	Erkek	9.00	0.00	0.00	0.00	0.00	15.00	85.00	Dogal
776	Erkek	18.00	0.00	0.00	0.00	0.00	0.00	90.00	Dogal
777	Erkek	17.00	0.00	35.00	0.00	0.00	0.00	25.00	Gulumsuyor
778	Erkek	17.00	0.00	40.00	0.00	0.00	0.00	0.00	Gulumsuyor
779	Erkek	16.00	0.00	20.00	0.00	0.00	0.00	80.00	Dogal
780	Erkek	15.00	0.00	100.00	0.00	0.00	0.00	0.00	Gulumsuyor
781	Erkek	14.00	0.00	0.00	0.00	0.00	0.00	100.00	Dogal
782	Erkek	13.00	100.00	0.00	0.00	0.00	0.00	0.00	Guluyor
783	Erkek	12.00	0.00	0.00	0.00	30.00	0.00	90.00	Dogal
784	Kadin	12.00	0.00	0.00	0.00	60.00	0.00	10.00	Kizgin
785	Erkek	11.00	0.00	0.00	0.00	0.00	0.00	80.00	Dogal
786	Kadin	10.00	0.00	90.00	0.00	0.00	0.00	0.00	Gulumsuyor
787	Kadin	9.00	0.00	80.00	0.00	0.00	0.00	20.00	Gulumsuyor
788	Erkek	8.00	0.00	90.00	0.00	0.00	0.00	0.00	Gulumsuyor
789	Kadin	7.00	100.00	0.00	0.00	0.00	0.00	0.00	Guluyor
790	Kadin	6.00	0.00	0.00	0.00	80.00	0.00	10.00	Kizgin
791	Kadin	6.00	0.00	0.00	0.00	40.00	0.00	0.00	Kizgin
792	Kadin	5.00	0.00	0.00	0.00	0.00	0.00	95.00	Dogal
793	Kadin	4.00	0.00	100.00	0.00	0.00	0.00	0.00	Gulumsuyor
794	Kadin	3.00	0.00	0.00	0.00	50.00	0.00	25.00	Kizgin
795	Kadin	2.00	0.00	0.00	0.00	0.00	60.00	0.00	Uzgun
796	Kadin	1.00	0.00	0.00	90.00	0.00	0.00	10.00	Sasirmis
797	Erkek	3.00	0.00	0.00	0.00	98.00	0.00	0.00	Kizgin
798	Erkek	2.00	0.00	0.00	0.00	0.00	70.00	0.00	Uzgun
799	Kadin	1.00	0.00	0.00	90.00	0.00	0.00	0.00	Sasirmis
800	Erkek	1.00	0.00	0.00	80.00	0.00	0.00	0.00	Sasirmis

ÖZGEÇMİŞ

1981 yılında Trabzon'da doğmuştur. İlköğrenimin 1.sınıfını Trabzon 24 Şubat ilkokulunda, kalan kısmını ise Trabzon 100.Yıl İlköğretim okulunda (1987 – 1992), orta ve lise öğrenimini Trabzon Anadolu Lisesi'nde (1992 – 1999) tamamladı. 2000 yılında Karadeniz Teknik Üniversitesi Mühendislik Mimarlık Fakültesi Bilgisayar Mühendisliği Bölümünü kazandı. 2000 yılı güz döneminde hazırlık eğitimini tamamlayarak 2001 yılında lisans eğitimine başladı ve 2005 yılında bu bölümden mezun oldu. Aynı yıl Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı'nda Yüksek Lisans Programı'na başladı.