

**YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**ŞABLON EŞLEME VE ÇOK KATMANLI ALGILAYICI
KULLANILARAK YÜZ TANIMA SİSTEMİNİN
GERÇEKLENMESİ**

Elektronik Mühendisi Serhan TUNA

**FBE Elektronik ve Haberleşme Mühendisliği Anabilim Dalı Elektronik Programında
Hazırlanan**

YÜKSEK LİSANS TEZİ

Tez Danışmanı : Yrd. Doç. Dr. Lale Özyılmaz (Yıldız Teknik Üniversitesi)

İSTANBUL, 2008

İÇİNDEKİLER

	Sayfa
SİMGE LİSTESİ	iv
KISALTMA LİSTESİ	v
ŞEKİL LİSTESİ	vi
ÇİZELGE LİSTESİ	viii
ÖNSÖZ	ix
ÖZET	x
ABSTRACT	xi
1 GİRİŞ	1
2 YÜZ TANIMA SİSTEMLERİNDE KULLANILAN YÖNTEMLER	4
2.1 Geometrik Özellik Tabanlı Yöntemler	4
2.2 Şablon Eşleme Tabanlı Yöntemler	6
2.3 Yapay Sinir Ağları	7
2.3.1 Yapay Sinir Ağlarının Genel Özellikleri	8
2.3.2 Temel Yapay Sinir Ağları	9
2.3.2.1 Tek Katmanlı Algılayıcılar (TKA)	9
2.3.2.2 Çok Katmanlı Algılayıcı (ÇKA)	10
3 KULLANILAN VERİTABANI VE GERÇEKLENEN UYGULAMALAR	12
4 BAŞARI ORANI SONUÇLARI	15
4.1 Bütün Yüz Kullanılarak Şablon Eşleme	16
4.1.1 Hafıza Veritabanı İmajlarının, Boş İfadeli Test İmajları ile Bütün Yüz Kullanılarak Test Edilmesi	16
4.1.2 Hafıza Veritabanı İmajlarının, Rastgele İfadeli Test İmajları ile Bütün Yüz Kullanılarak Test Edilmesi	17
4.2 Yüzdeki Belirli Bölgelerin Kullanılması ile Başarı Oranı Artırımının Gerçekleştirilmesi	21
4.2.1 Boş İfadeli Test İmajları ve Hafıza İmajlarının Belirli Bölgelerinin Çıkarılması ile Test Edilmesi	21
4.2.2 Rastgele İfadeli Test İmajları ve Hafıza İmajlarının Belirli Bölgelerinin Çıkarılması ile Test Edilmesi	22
4.3 Gözler Arasındaki Mesafenin Kullanılması ile Başarı Oranının İncelenmesi	23
4.4 Yapay Sinir Ağlarının Sistemimizde Kullanılışı ve Başarı Oranı Sonuçları	26
4.4.1 Örneklerin Toplanması	26
4.4.2 Ağın Topolojik Yapısı	30
5 BOYUT KÜÇÜLTME İŞLEMİ İLE İŞLEM ZAMANININ AZALTILMASI	33

5.1	Simulasyon Sonuçları	35
5.1.1	Gülme İfadeli İmajların Performans Sonuçları.....	35
5.1.2	Boş İfadeli İmajların Performans Sonuçları	37
5.1.3	Kızma İfadeli İmajların Performans Sonuçları.....	39
5.1.4	Şaşıрма İfadeli İmajların Performans Sonuçları	41
6	SONUÇ VE GELECEK ÇALIŞMA	44
KAYNAKLAR.....		46
EKLER		49
Ek 1 Matlabta Yazılan Kodlar		50
ÖZGEÇMİŞ.....		57

SİMGE LİSTESİ

p	Uzaklık formülü parametresi
d	Uzaklık Simgesi
Σ	Toplam İşareti
E	Hata değeri
y	Yapay sinir hücresinin çıktı değeri
Δw	Yapay sinir hücresine bağlayan bağlantının ağırlığındaki değişim miktarı
η	Öğrenme katsayısı
α	Momentum katsayısı
δ	Yapay sinir hücresine ait hata faktörü

KISALTMA LİSTESİ

CNNL	Computation Neural Network Laboratory
TDNN	Zaman gecikmeli sinir ağı
NN	Neural Network
YSA	Yapay Sinir Ağı
CB	City Block
ÇKA	Çok Katmanlı Algılayıcı
TKA	Tek Katmanlı Algılayıcı
LVQ	Vektör Kuantalamalı Öğrenme
RTFA	Radyal Tabanlı Fonksiyon Ağı
TH	Toplam Hata

ŞEKİL LİSTESİ

Şekil 2.1 Bir yüzde belirlenen 22 adet geometrik özellik (R. Brunelli vd., 1992)	5
Şekil 2.2 Geometrik parametrelerinin yüz üzerinde gösterilmesi (E. Dede, 2000)	6
Şekil 2.3 Öğrenme ve test etme	8
Şekil 2.4 En basit TKA modeli.....	9
Şekil 2.5 ÇKA modeli	10
Şekil 3.1 CNNL veritabanında bulunan 1200 kişinin etnik köken dağılımı	12
Şekil 4.1 İmajların Matematiksel ifadeye dönüştürülmesi	15
Şekil 4.2 Boş ifade test imajları ile elde edilen sonuçlar.....	17
Şekil 4.3 Rastgele ifade test imajları ile elde edilen sonuçlar	18
Şekil 4.4 Ağız ve burun bölgelerine sıfır piksel değeri atanması.....	21
Şekil 4.5 Boş ifade test imajları ile elde edilen sonuçlar.....	22
Şekil 4.6 Rastgele ifade test imajları ile elde edilen sonuçlar	23
Şekil 4.7 Sağ göz.....	24
Şekil 4.8 Sol göz.....	24
Şekil 4.9 Gözler arasındaki mesafenin hesaplanması.....	24
Şekil 4.10 Gözler arasındaki mesafenin kullanılması ile elde edilen sonuçlar (Boş ifadeli test imajları.....	25
Şekil 4.11 Gözler arasındaki mesafenin kullanılması ile elde edilen sonuçlar (Rastgele ifadeli test imajları	26
Şekil 4.12 Canny metodunun yüz imajlarında uygulanması.....	28
Şekil 4.13 Sobel metodunun yüz imajlarında uygulanması	28
Şekil 4.14 Roberts metodunun yüz imajlarında uygulanması	28
Şekil 4.15 Prewitt metodunun yüz imajlarında uygulanması	28
Şekil 4.16 Yapay sinir ağı girişi için kullanılan maskeler.....	29
Şekil 4.17 Kullanılan ağın topolojik yapısı	31
Şekil 4.18 Yüz tanıma ağının oluşumuna ait eğitim performans grafiği.....	32
Şekil 4.19 YSA başarı oranı sonuçları	32
Şekil 5.1 İnterpolasyon prosesi.....	34
Şekil 5.2 Boş ifadeli test imajları ile gülme ifadelerinin karşılaştırılmasının süre başarı oranı grafiği	36
Şekil 5.3 Rastgele ifadeli test imajları ile gülme ifadelerinin karşılaştırılmasının süre başarı oranı grafiği.....	37
Şekil 5.4 Boş ifadeli test imajları ile boş ifadeli hafıza imajlarının karşılaştırılmasının süre	

başarı oranı grafiği	38
Şekil 5.5 Rastgele ifadeli test imajları ile boş ifadeli hafıza imajlarının karşılaştırılmasının süre başarı oranı grafiği.....	39
Şekil 5.6 Boş ifadeli test imajları ile kıзма ifadelerinin karşılaştırılmasının süre başarı oranı grafiği	40
Şekil 5.7 Rastgele ifadeli test imajları ile kıзма ifadeli imajların karşılaştırılmasının süre başarı oranı grafiği.....	41
Şekil 5.8 Boş ifadeli test imajları ile şaşırma ifadeli imajların süre başarı oranı grafiği	42
Şekil 5.9 Rastgele ifadeli test imajları ile şaşırma ifadeli imajların süre başarı oranı grafiği ..	43

ÇİZELGE LİSTESİ

Çizelge 1.1 Biyometrik teknolojilerin karşılaştırılması	1
Çizelge 4.1 Örnek bir eşleme sonucu çizelgesi	20
Çizelge 4.2 Her bir kişiye ait 2 gözbebeği arasındaki mesafe	24
Çizelge 4.3 YSA giriş matrisi	29
Çizelge 4.4 YSA Çıkış (Hedef) Matrisi	30
Çizelge 5.1 İmaj boyutlarının piksel boyutlarına karşılık gelen değerleri	35

ÖNSÖZ

Bu tez çalışmasında yüz tanıma sistemi ele alınmış ve sistemin başarı oranının artırılması için çeşitli metotlar denenerek elde edilen başarı oranları incelenmiştir.

Çalışma bir yüksek lisans tez konusu olmakla birlikte, birkaç yıl içinde gerçek kameralı bir sistemle denenmesi amaçlanmaktadır.

Çalışmamın her aşamasında bana her türlü desteği gösteren, bilgi ve birikimini benimle paylaşan danışman hocam Yrd. Doç. Dr. Lale Özyılmaz'a ve Wayne State Üniversitesi laboratuvarlarında büyük bir uğraşı ve sabırla hazırlanan CNL veritabanında emeği geçen Sayın Mehmet Artıklar ve Sayın Metin Artıklar hocalarıma sonsuz teşekkürlerimi sunuyorum. Çalışmalarım sırasında manevi desteğini hiç esirgemeyen eşime de en içten sevgilerimi sunuyorum.

ÖZET

Bu tezde farklı yöntemlerin kullanılması ile yüz tanıma sistemleri gerçekleştirilmiştir. Çalışmada Wayne State üniversitesinde hazırlanan CNNL veritabanı kullanılmıştır. Farklı etnik kökenlerdeki 1266 kişiye ait, değişik zamanlarda çekilmiş 10'ar adet ön yüz imajı içeren CNNL veritabanı, test seti ve eğitim seti olmak üzere iki kısımdan oluşmaktadır. Test setinde her bir kişinin 2 farklı yüz ifadesi bulunmaktadır. Eğitim seti ise her bir kişiye ait 8 farklı yüz ifadesi içermektedir.

Çalışmada ilk olarak şablon eşleme metodu kullanılmıştır. Şablon eşleme metodu ile yüz tanıma işlemi, Minokowski sınıf metriklerinden birisi olan Şehir Blok Uzaklık formülü kullanılarak gerçekleştirilmiştir.

İkinci bölümde, ilk bölümde elde edilen başarı oranının artırılması için 2 farklı yöntem kullanılmıştır. İlk olarak yüzün belirli bölgelerinin çıkarılması ile elde edilen görüntüler kullanılarak tekrar şablon eşleme yöntemi uygulanmıştır. İkinci olarak, kişilerin gözbebekleri arasındaki mesafenin kullanılması ile yüz tanıma yapılmıştır.

Üçüncü bölümde, yüz tanıma sisteminin performansı Çok Katmanlı Algılayıcı kullanılarak değerlendirilmiştir.

Son bölümde ise, imaj boyutları küçültülerek ve şablon eşleme yöntemi kullanılarak sistemin performansı (başarı oranı ve sistem hızı) incelenmiştir.

ABSTRACT

In this thesis, face recognition systems has been carried out using different methods. In order to test the methods, CNL database has been used. CNL database includes 10 different expressions of each person that belongs to 1266 people. CNL database has been composed of 2 sections which are memory set and test set. Test set includes 2 different expressions of each person. Memory set includes 8 different expressions of each person.

Firstly, template matching method has been used. Face recognition system with template matching method has been carried out using City Block Distance, one of the Minokowski class metrics.

Secondly, in order to boost the value of success rate which has been obtained in the previous section, 2 different methods has been used. In the first method; second template matching method has been applied by extracting some parts of the image which decreases the success rate. In the second method; pupils has been automatically detected and the distance between 2 pupils has been measured. Success rate has been boosted with this distance.

Thirdly, success rate has been examined using Multi Layer Neural Networks.

In the last section, image dimensions has been reduced and the performance of the system (success rate and process time) has been examined.

1. GİRİŞ

Son zamanlarda çıkan savaş, terör, cinayet ve hırsızlık gibi olayların artması güvenlik sistemlerinin önemini çok açık bir şekilde ortaya çıkarmaktadır. Kişinin kendisine has olan tekil fizyolojik ve/veya davranışsal bir takım karakteristik özellikleri ile onu tanımak ve kimliklendirmek için kullanılan metotlar bütününe biyometri denir. Başka bir anlatımla, insan biyolojisinden insanları tanıma ve ayırt etmeyi ifade eden bir metottur. Parmak izi bir örnek olarak verilebilir. Bilindiği gibi her kişinin parmak izleri farklı olduğundan kişilerin tanınması kolayca gerçekleştirilebilir. Fakat parmak izi insanların aktif katılımıyla bir başka deyişle kişilerin bilerek bir sisteme parmağını dokundurması ile alınıp kişilerin tanınması gerçekleştirilebilir. Fakat yüz tanıma tekniği ile kişilerin tanınması pasif olarak yani kişilerin doğrudan ve bilinçli bir şekilde yardımcı olmasına gerek kalmadan gerçekleştirilebilir (A.Samal vd., 1992).

Kişi tanımada, kişilerin fiziksel özellikleri kullanarak otomatik olarak tanınması amaçlanmıştır. Biyometri yöntemleri; parmak izi eşleştirme, iris tanıma, retina taraması, ses ve konuşma tanıma, yüz tanıma, el yazısı ve imza tanıma, el geometrisi tanımada kullanılır (Welch P.D., 1967). Kişi tanıma; o kişiye ait yüz, parmak izi, el geometrisi, iris, retina, DNA ve ses ana başlıkları altında sayılan biyometri özellikler kullanılarak yapılabilir. Kişi tanımada kullanılan bu teknolojilerin karşılaştırılması Çizelge 1.1 'de verilmiştir (Anıl vd., 1997).

Çizelge 1.1 Biyometrik teknolojilerin karşılaştırılması

Biyometrik	Evrensellik	Eşsizlik	Kalıcılık	Toplanabilirlik	Performans	Kabul Edilebilirlik	Atlatılabilirlik
Yüz	Yüksek	Düşük	Orta	Yüksek	Düşük	Yüksek	Düşük
Parmak İzi	Orta	Yüksek	Yüksek	Orta	Yüksek	Orta	Yüksek
El Şekli	Orta	Orta	Orta	Yüksek	Orta	Orta	Orta
El Damarları	Orta	Orta	Orta	Orta	Orta	Orta	Yüksek
İris	Yüksek	Yüksek	Yüksek	Orta	Yüksek	Düşük	Yüksek
Retinal Tarama	Yüksek	Yüksek	Orta	Düşük	Yüksek	Düşük	Yüksek
İmza	Düşük	Düşük	Düşük	Yüksek	Düşük	Yüksek	Düşük
Ses İzi	Orta	Düşük	Düşük	Orta	Düşük	Yüksek	Düşük
Yüz Termogramı	Yüksek	Yüksek	Düşük	Yüksek	Orta	Yüksek	Yüksek

Kişi tanımada günümüzde en yaygın yöntem olarak yüz tanıma kullanılmaktadır. Yüz tanıma her dönemde güncelliğini korumuş ve üzerinde fazla sayıda çalışma yapılan bir konu olmuştur

(Anil K. Jain, 2000). Yüz tanımanın uygulama alanları çok geniştir. Bazı uygulama alanları şu şekilde listelenebilir: kredi kartları, sürücü ehliyetleri, pasaportlar, kişisel tanıma kartları ve banka güvenliği.

Yüz tanıma sistemlerinde genel olarak iki yaklaşım kullanılmaktadır. Bunlar, yüz tabanlı tanıma sistemi ve özellik tabanlı tanıma sistemleridir. Yüz tabanlı tanıma sistemlerinde bütün imaj, bazı standart teknikleri veya bir nöron ağını kullanarak iki boyutlu yoğunluk fonksiyonu olarak analiz edilir (W.J. Welsh, 1991)

Son 10 yılda yüzün tanınması için birçok yöntem denenmiş ve gerçekleştirilmiştir. Bu yöntemler arasında görünüşe dayalı yöntemler, yüz imgelerini iki boyutlu nesneler olarak işlemektedirler. Bu yöntemlerde $w \times h$ boyutlu imgeler, wh -boyutlu uzaydaki noktalar olarak ifade edilirler. Örnek uzayı olarak isimlendirilen bu uzay genellikle çok büyük boyutludur. Bununla birlikte insan yüzlerinin benzer yapıya sahip olması nedeniyle, yüz imgeleri daha küçük boyutlu bir uzayda fazla bilgi kaybı olmadan ifade edilebilirler. Bu amaçla Turk (1991), Belhumeur (1997), Null Space ve Direct -LDA yöntemleri geliştirilmiştir. Bunun yanında yüz ifadesi değişimleri için, örtük görüntüler ve aydınlanma değişimi durumlarında tanıma başarısını artırmak amacıyla Gabor dalgacık dönüşümü ile otomatik olarak bulunan öznelikler kullanılmıştır. Gabor dalgacık dönüşümü ile öznelikler temel olarak kenar köşe gibi değişimin yüksek olduğu bölgelere yerleştirildiğinden aydınlanma değişimlerinin yüz tanıma üzerindeki etkisi azaltılmıştır. Öznelik vektörlerinin lokal olarak karşılaştırılması ifade değişimleri ve örtük görüntülerinin başarıyla tanınmasına imkan sağlamıştır. Ancak rotasyon ve yönelim etkileri çözümsüz kalmıştır (Kepenççi, 2002).

Yüz tanımda rotasyon ve yönelimin değiştiği görüntülerde başarıyı artırmak amacıyla başlıca iki yöntem kullanılmıştır. Bunlardan ilki birden fazla, farklı rotasyon ve yönelimlerdeki yüz görüntülerinin sistemin eğitiminde kullanılmasıdır. Böylelikle test edilen görüntü daha önceden sisteme tanıtılmış örneklerden rotasyonu en yakın olanına en fazla benzerlik gösterecektir. İkinci yöntem ise 3 boyutlu yüz modellerinin kullanılmasıdır. Bu yöntemde 3 boyutlu yüz modellerinden farklı rotasyon ve aydınlanmalarda 2 boyutlu sentez yüz görüntüleri oluşturulmaktadır (Huang vd., 2006).

Yüz algılama için düşünülen modellerde yukarıda bahsedilen etkenler düşünülerek geliştirilmiş algoritmalar mevcuttur. Govindaraju (1990) yüzün ön cepheden görünüşünün kenar detay bilgisine dayalı bir algoritma geliştirmiş ve başarı oranını %75 olarak vermiştir. Sanger ve Miyake (1997) dudak rengine dayalı ikili çapraz medyan süzgeç kullanarak dudak piksellerini bulmaya çalışmışlar ve %92 oranında bir başarı elde etmişlerdir.

Yapay sinir ağıları, yüzün doğru olarak tanınabilmesine yönelik algoritmalarda etkili bir şekilde kullanılmışlardır. Rowley (1996) sinir ağına dayalı önden görünüşlü dikey yüz bulan bir sistem tasarlamış, Yacoub (1999) da çok-katmanlı bir algılayıcı ve Fourier dönüşümüne dayalı bir sistem geliştirmiştir. Soulie, Vinnet ve Lamy (1993) bir zaman gecikmeli sinir ağı (TDNN) yöntemi kullanmışlardır. Han (2000) sinir ağlarına dayalı morfoloji tabanlı bir yöntem ile görüntü üzerinde göze benzer parçalar arayarak yüz bulmaya çalışmıştır. Bu yöntem için başarı oranı %94 olarak verilmiştir. Görüntü üzerinde bazı sınırlamalar olması durumunda bu yöntemin başarı oranı çok düşmektedir (Han, 2000).

Bizim gerçekleştirdiğimiz yüz tanıma sisteminde ise iki kavram önemlidir. Bu iki kavram, Yüz Veritabanı ve Yüz Tanıma Algoritmasıdır.

Yüz veritabanı yüz imajlarının birikiminden oluşmaktadır. Bu veritabanında her bir kişinin farklı yüz ifadeleri bulunmaktadır, bütün yüz imajlarının özellik yönünden birbirlerinden farkı yoktur. Aynı ebatlara sahip ve kapalı ortamlarda ön cepheden çekilerek elde edilmiştir.

Yüz tanıma algoritması ise giriş imajı ile veritabanında bulunan imajlar arasında benzerlik bulmaya yarayan bir yöntemdir. Sisteme girilen imaj, algoritma sayesinde veritabanındaki imajlarla karşılaştırılmış, resmin veritabanına ait olup olmadığına karar verilmiştir.

Yüz tanıma sistemlerinde birçok problem ile karşılaşmaktadır. Örneğin yüzde bulunan gözlük gibi aparatlar veya kişinin sakallı ve/veya bıyıklı olması, saçının uzunluğu veya kısalığı, resmin üzerinde bulunan ışık seviyelerinin fazlalığı veya azlığı başarı oranını önemli derecede etkilemektedir. Bu tezde mümkün olduğu kadar bu dezavantajları minimize ederek yüz tanıma işlemi gerçekleştirilmiştir.

Tezin ikinci bölümünde yüz tanıma sisteminde sıklıkla kullanılan yöntemlerden bahsedilmiştir.

Üçüncü bölümde, kullanılan veritabanı ve gerçekleştirilen sistem detaylı bir şekilde ele alınmıştır.

Dördüncü bölümde, çalışmada kullanılan yöntemler anlatılmıştır. Ayrıca başarı oranı sonuçları hem sayısal hem de grafiksel olarak gösterilmiştir.

Beşinci bölümde, algoritmanın uygulanmaya başlanmasından sonuca ulaşana kadar geçen sürecin performansı incelenmiş ve boyut küçültme işlemi yapılarak bu zaman performansı artırılmıştır. Son olarak altıncı bölümde ise, sonuç ve gelecek çalışmalardan bahsedilmiştir.

2. YÜZ TANIMA SİSTEMLERİNDE KULLANILAN YÖNTEMLER

İnsanlar birbirlerini ayırt etmede en kolay yüzlerinden faydalanırlar. Bu sebeple bu yöntem neredeyse insanlık tarihiyle yaşıt bir biyometrik ayırt etme yöntemidir. Fakat bilimsel çalışmada bir ayırt edici özellik olarak bilgisayar ortamında kullanılması oldukça yeni sayılabilecek bir konu olmuştur. Örneğin, bir suçluyu tanımlamada polis tarafından kullanılan bir yöntemdir. Polis, suçluyu anlatan kişiye suçlu hakkında sorular sorup yüz şeklini belirlemeye çalışır. Bu konuda polis ressamlarının daha önce çizmiş oldukları standart yüz parçaları vardır. Verilen tarife göre bu yüz parçaları bir araya getirilerek suçlunun yüz şekline yakın bir yüz şekli bulunmaya çalışılır.

Bilgisayarın hayatın içerisindeki hızlı kullanımı, yüz tanıma yöntemleri konusunda da önemli gelişmelere olanak sağlamıştır. Yüz tanıma sistemleri son yıllarda güvenlik sebebiyle daha da önemli araştırma konularından birisi olmuştur. Yüzün doğru bir şekilde tanınması için birçok yöntem uygulanmış ve başarı oranı artırılmaya çalışılmıştır.

Yüz tanıma sistemlerinde kullanılan ve en popüler olarak bilinen yöntemleri üç ana sınıf altında toplayabiliriz. Bunlar, geometrik özellik tabanlı yöntemler, şablon eşleme yöntemleri ve sinir ağı tabanlı yöntemlerdir (öğrenme tabanlı yöntemlerdir).

2.1 Geometrik Özellik Tabanlı Yöntemler

İnsanı görsel olarak incelediğimizde, yüz en önemli kısımlar arasında gelmektedir. Çocukluk devrinden bu yana her geçen yıl önemli değişiklikler göstermektedir. Aynı anda bile aynı kişinin her bir ifadesinde yüz şekilleri değişir ve bu değişim her insanda birbirinden farklıdır.

Herhangi bir yüz, en önemli kısımları tam olarak net olmasa da (göz, burun ve ağız gibi) tanınabilir. Bunu gerçekleştirmek için net olmayan bu kısımların özellik çıkarma yöntemi ile geometrik kısımlarının belirlenmesi yeterlidir (Turk, 1991; Swets, 1996). Goldstein (1971) bir bilgisayar programıyla bu işlemi gerçekleştirmiş ve memnun edici bir performans ile karşılaşmıştır.

Geometrik tanıma, yüz üzerindeki belirleyici özelliğe sahip alanların boyutlarının birbirleri ile olan uzaklıkları ve geometrisi veya bunların oluşturduğu vektörlerin karşılaştırılması ile yapılan tanımlama yöntemidir. Bugüne kadar kullanılan ilk yüz tanımlama metodudur. Goldstein (1964) tarafından kullanılan yöntemde, yüz üzerinde belirleyici özellikteki noktalar manuel olarak belirlenmiş, geometrisi hesaplanmış ve böylece tanımlama yapılmıştır. Kaya ve

Kaya (1972) yüzdeki belirli noktaların arasındaki Euclidean (Öklit) uzaklığını hesaplamış ve bu noktaların belirlenmesinde bazı önemli faktörler belirlemişlerdir.

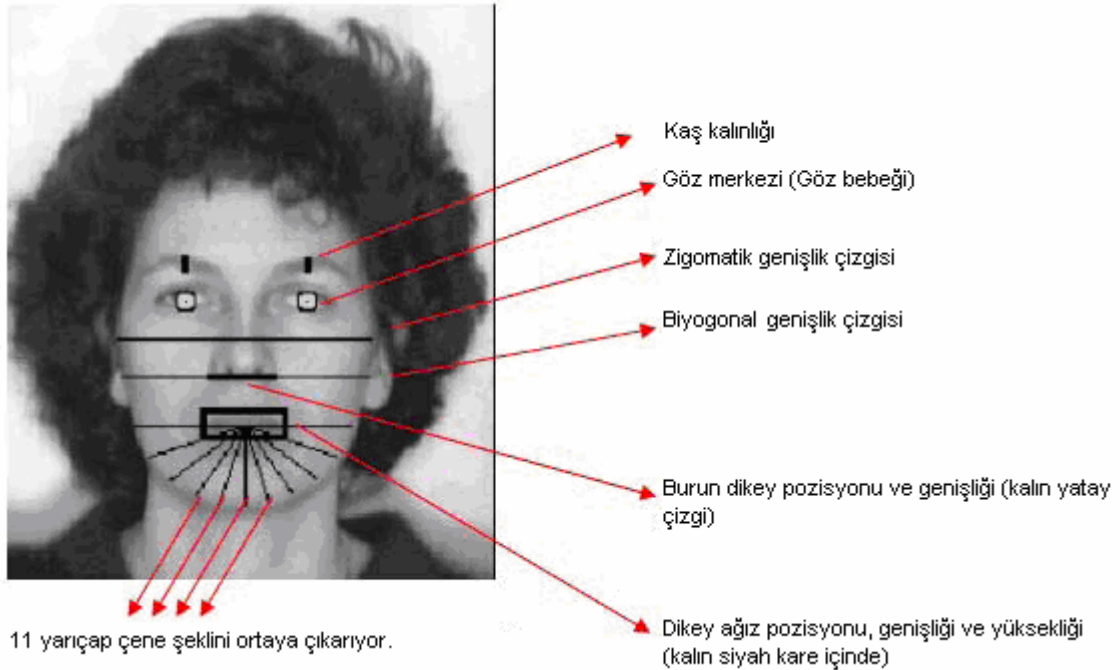
1. Tahmini kolay olmalı
2. Işığa duyarlılık mümkün olduğunca az olmalı
3. Yüzde değişiklik (mimik) olan noktalardan mümkün olduğunca bağımsız olmalı
4. Ayırt edici özelliğe sahip bilgi mümkün olduğunca fazla olmalı gibi.

Sonrasında, bu fikir Kanade (1973) tarafından geliştirilmiş ve bilgisayar tabanlı çalışan yüz tanıma sistemi yapılmıştır.

Yüz üzerinde geometrik tanımayı gerçekleştirmek için seçilen noktaları örneklendirecek olursak; yüz üzerindeki tanımlayıcı özelliğe sahip olan göz, kaş, ağız, burun ve çene örnek olarak sayılabilir.

Aşağıdaki imajda, geometrik tanımda kullanılan belirleyici özelliğe sahip noktaları gösterebiliriz (R. Brunelli vd., 1992).

R.Brunelli ve T.Poggio (1992) çalışmalarında Şekil 2.1’de görüldüğü üzere 22 geometrik noktayı yüz için belirleyici kabul etmişlerdir.

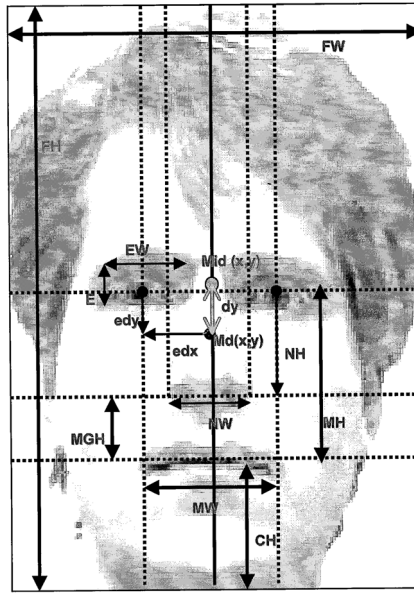


Şekil 2.1 Bir yüzde belirlenen 22 adet geometrik özellik (R. Brunelli vd., 1992).

22 adet geometrik özelliğin 11 tanesi, ağız merkezinde ve çenenin dış çevre noktalarına olan uzaklıklarda, 4 tanesi ağız kısmını çevreleyen dikdörtgenin kenar uzunluklarında, 1 tanesi burnun alt kısmında genişliğini veren çizgide, 2 tanesi nefes alma hareketinin olduğu çizgi ve

elmacık kemiklerinin üzerinden yüz boyunca geçen çizgilerde, 2 tanesi göz merkezlerinde ve son 2 tanesi de göz merkezlerine karşılık gelen noktalarındaki kaşların yüksekliğinde bulunmaktadır. Elde edilen 22 adet değer, En Yakın Komşu (*Nearest Neighbour-NN*) algoritması kullanılarak karşılaştırılır ve en küçük değere sahip olan veri tabanı resmi ile eşleştirilir (R. Brunelli 1992).

Aşağıdaki imajda yüz geometrisi belirlemenin bir örneği verilmiştir. Bu şekil sayesinde göz, burun ve ağız kısımları ve diğer önemli kısımları ortaya çıkarılarak yüz tanıma sistemi gerçekleştirilmeye çalışılmıştır



Şekil 2.2 Geometrik parametrelerinin yüz üzerinde gösterilmesi (E. Dede, 2000)

2.2 Şablon Eşleme Tabanlı Yöntemler

Şablon eşleme yöntemi geometrik tanımadan farklı olarak daha basit bir tekniğe sahiptir. Şablon eşleme, iki farklı imaj arasındaki piksel farkının, uygulanan bir algoritma ile belirlenerek karşılaştırılmasıdır. Şablon eşleme yöntemi iki veri arasındaki benzerlikleri tanımlayan desen tanıma için kullanışlı olan bir yöntemdir. Bu iki veri; sinyal, vektör, dizi ya da 2 boyutlu imaj desenleri olabilir. Seçilen imajlar gri renk düzlemindedir. Eşleme algoritması, eğitim ve test veri tabakaları arasındaki benzerlikleri ve farkları hesaplar. Şablon eşlemesi tüm resme veya resmin muhtelif kısımlarına uygulanmalıdır. Genel olarak şablonlar arasındaki en çok benzerliklerin veya en çok farkların bulunabileceği kısımlar seçilir. Şablon desenleri, eğitim verilerinin alınan referans desenleri ya da referans kümeleri olarak kabul edilir. Bir test resmi, bu kalıp desenleri kullanılarak tanımlanmaya çalışılır. Sonuç olarak, eğitim ve test desenleri arasındaki benzerliklerinin ya da farklılıklarının hesaplanması

gereklidir. Genel olarak, bu benzerliklerin veya farklılıkların hesaplanması test desenlerinin veri desenleri ile karşılaştırılması ile yapılır. Desenler değişik boyutta ve oryantasyon düzeylerinde olabilirler. Bundan dolayı bunların aynı boyuta ve oryantasyon düzeyine çekilmesi gerekir.

Şablon eşleme yönteminin en önemli avantajı sistemimize uygun olması ve diğer sistemlere göre daha basit olmasıdır.

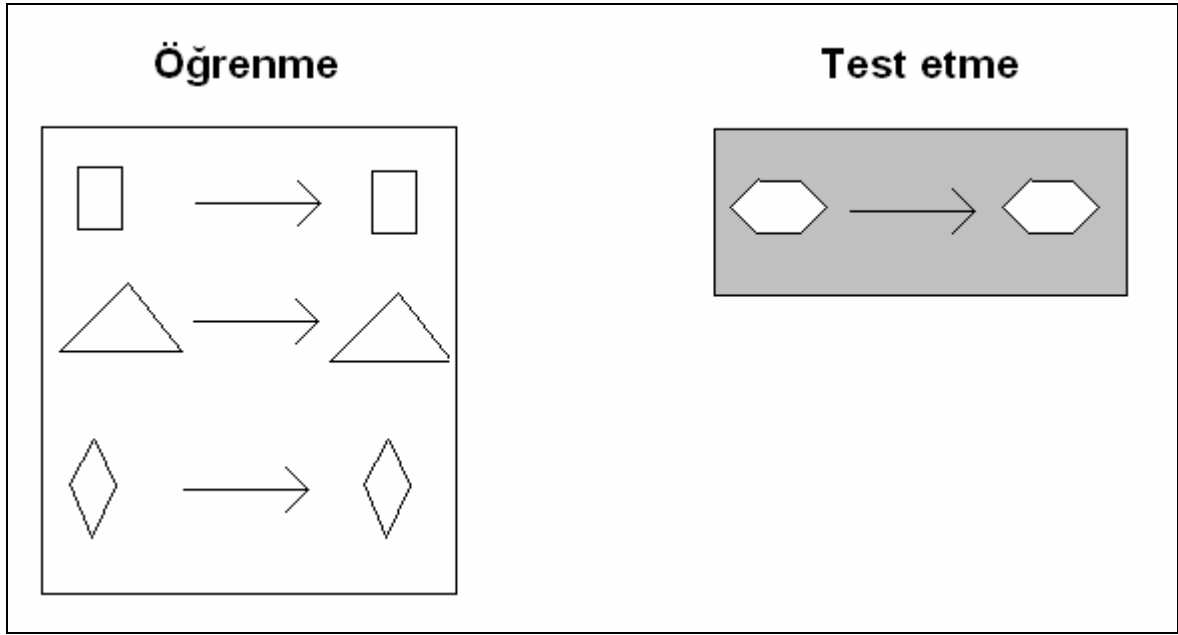
2.3 Yapay Sinir Ağları

Bu bölüm de ilk olarak yapay sinir ağları ile ilgili bilgiler verildikten sonra yapay sinir ağlarının yüz tanıma sistemlerinde genel olarak nasıl kullanıldığı anlatılmıştır.

Yapay sinir ağları, insan beyninin özelliklerinden olan öğrenme yolu ile yeni bilgiler türetebilme, yeni bilgiler oluşturabilme gibi yetenekleri herhangi bir yardım almadan otomatik olarak gerçekleştirmek amacı ile geliştirilen bilgisayar sistemleridir. Günümüzde kullandığımız geleneksel programlama yöntemleri ile bu yetenekleri gerçekleştirmek mümkün değildir. Buradan da anlaşılacağı üzere yapay sinir ağlarının, programlanması mümkün olmayan durumlar için geliştirilmiş adaptif (örnekleri kullanarak ilgili olay hakkında genelleme yapabilme) bilgi işleme ile ilgilenen bir bilgisayar bilim dalı olduğu söylenebilir.

Yapay Sinir Ağları, gerçek beyin fonksiyonlarının ürünü olan örnekleri kullanarak olayları öğrenebilen, çevreden gelen olaylara karşı nasıl tepkiler üreteceğini belirleyebilen bilgisayar sistemleridir. Yapay sinir ağları, insan beyninin fonksiyonel özelliklerine benzer şekilde, öğrenme, sınıflandırma vb. gibi konularda başarılı bir şekilde uygulanmaktadırlar. Örneklerden elde ettikleri bilgiler ile kendi işlemlerini oluşturup, benzer konularda yakın kararlar verirler. Yapay sinir ağları, birbirine hiyerarşik olarak bağlı ve paralel olarak çalışabilen yapay hücrelerden oluşmaktadır. Hücrelerin her bir bağlantısının bir değeri bulunmaktadır.

Teknik olarak, bir yapay sinir ağının en temel görevi, kendisine gösterilen bir girdi setine karşılık gelebilecek bir çıktı seti belirlemektir. Bunu yapabilmesi için ağ, şekilde gösterildiği gibi ilgili olayın örnekleri ile eğitilerek (öğrenme) genelleme yapabilecek yeteneğe kavuşturulur. Bu genelleme ile benzer olaylara karşılık gelen çıktı setleri belirlenir.



Şekil 2.3 Öğrenme ve test etme

2.3.1 Yapay Sinir Ağlarının Genel Özellikleri

- Yapay Sinir ağları makine öğrenmesi gerçekleştirirler yani temel işlevi bilgisayarların öğrenmesini sağlamaktır. Olayları öğrenerek benzer olaylar karşısında benzer kararlar vermeye çalışırlar.
- Yapay sinir ağlarında bilgi ağı bağlantılarının değerleri ile ölçülmekte ve bağlantılarda saklanmaktadır. Diğer programlarda olduğu gibi veriler veritabanında değildir. Bilgiler ağı üzerinde saklı olup ortaya çıkartılması ve yorumlanması zordur.
- Örnekleri kullanarak öğrenirler.
- Ağ kendisine gösterilen örneklerden genellemeler yaparak görmediği örnekler hakkında bilgiler üretebilir.
- Yapay sinir ağlarının en önemli özelliklerinden birisi sınıflandırma yapmasıdır. Verilen örneklerin kümelendirilmesi ve belirli sınıflara ayrıştırılarak daha sonra gelen bir örneğin hangi sınıfa gireceğine karar vermesi hedeflenmektedir.
- Bazı durumlarda ağa eksik bilgileri içeren bir şekil verilir. Ağı bu eksik bilgileri bulması istenir. Örneğin yırtık bir resmin kime ait olduğunu belirlemesi ve tam resmi vermesi gibi bir sorumluluk ağdan istenebilmektedir. Bu tür konularda yapay sinir

ağlarının çok etkin çözümler getirdiği bilinmektedir.

- Yapay sinir ağlarının örnekler ile kendisine gösterilen yeni durumlara adapte olması ve sürekli yeni olayları öğrenebilmesi mümkündür.
- Yapay sinir ağları eğitildikten sonra eksik bilgiler ile de çalışabilir ve gelen yeni örneklerde eksik bilgi olmasına rağmen sonuç üretebilirler.
- Hata toleransına sahiptirler. Ağın bazı hücrelerinin bozulması ve çalışamaz duruma düşmesi halinde ağ çalışmaya devam eder.
- Yapay sinir ağları sadece nümerik bilgiler ile çalışırlar. Sembolik ifadeler ile gösterilen bilgiler nümerik gösterime çevrilmeleri gerekmektedir.

2.3.2 Temel Yapay Sinir Ağları

Bu bölümde tek katmanlı algılayıcılar, perseptron modellerinin yapısından kısaca bahsedilmiştir. Bu modeller daha sonra geliştirilen modellere temel oluşturdıklarından önemlidir.

2.3.2.1 Tek Katmanlı Algılayıcılar (TKA)

Tek katmanlı yapay sinir ağları sadece girdi ve çıktı katmanlarından oluşur. Her ağın bir veya daha fazla girdisi ve çıktısı vardır. Çıktı üniteleri bütün girdi ünitelerine bağlanmaktadır. Her bağlantının bir ağırlığı vardır. Şekilde TKA modeline bir örnek verilmiştir. Ağın çıktısının sıfır olmamasını sağlayan birde eşik değeri vardır. Eşik değeri girdisi daima 1'dir.



Şekil 2.4 En basit TKA modeli

Ağın çıktısı ağırlıklandırılmış girdi değerlerinin eşik değeri ile toplanması sonucu bulunur. Bu

girdi değeri bir aktivasyon fonksiyonundan geçirilerek ağın çıktısı hesaplanır. Tek katmanlı algılayıcılara, basit algılayıcı modeli (*Persptron*), Adaline ve Madaline üniteleri örnek olarak verilebilir. Tek katmanlı algılayıcılar sadece doğrusal olan problemlerin çözümünde kullanılabilir. Doğrusal olmayan problemlerin çözümü için Çok Katmanlı Algılayıcılar (ÇKA) geliştirilmiştir.

2.3.2.2 Çok Katmanlı Algılayıcı (ÇKA)

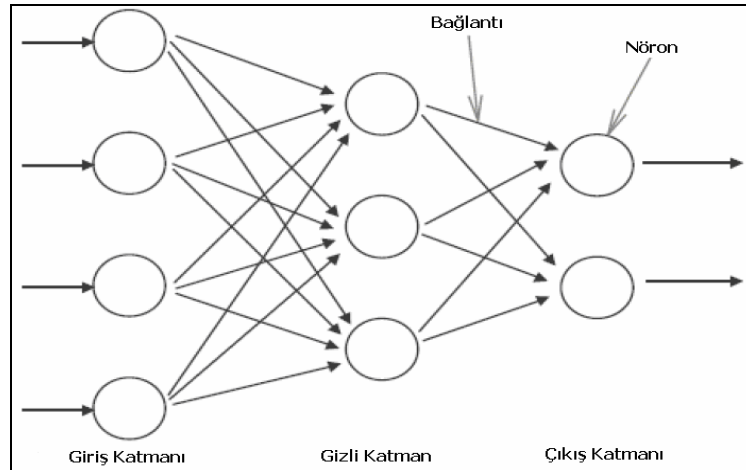
Rumelhart tarafından 1986 yılında geliştirilmiş olan bu modele hata yayma modeli veya geriye yayım modeli (backpropagation network) de denmektedir. Yüz tanıma sistemimizde de kullanmış olduğumuz bu model, özellikle sınıflandırma, tanıma ve genelleme gerektiren problemlerin çözümünde çok etkin rol oynamaktadır. Bu model *Delta Öğrenme Kuralı* denilen bir öğrenme yöntemini kullanmaktadır. Bu modelin temel amacı ağın beklenen çıktısı ile ürettiği çıktı arasındaki hatayı en aza indirmektir. ÇKA modeli ileriye doğru bağlantılı ve 3 katmandan oluşan bir ağıdır. Bu katmanlar sırasıyla;

a) Giriş Katmanı: Bu katmandaki yapay sinir hücreleri dış dünyadan bilgileri alarak ara katmanlara transfer ederler. Bazı ağlarda girdi katmanında herhangi bir bilgi işleme olmaz.

b) Ara Katman (Gizli Katman): Giriş katmanından gelen bilgiler işlenerek çıkış katmanına gönderilirler. Bu bilgilerin işlenmesi ara katmanlarda gerçekleştirilir. Bir ağ içinde birden fazla ara katman olabilir.

c) Çıkış Katmanı: Bu katmandaki işlem elemanları ara katmandan gelen bilgileri işleyerek ağın girdi katmanından sunulan girdi seti için üretmesi gereken çıktıyı üretirler. Üretilen çıktı dış dünyaya gönderilir.

Şekil 2.5’de örnek bir ÇKA modeli gösterilmiştir.



Şekil 2.5 Örnek bir ÇKA modeli

Tezde Çok Katmanlı Algılayıcılardan birisi olan, ileri beslemeli geri yayılım algoritması kullanılmıştır.

Geri yayılım algoritması (Back-propagation) birçok uygulamalarda kullanılmış en yaygın öğrenme algoritmasıdır. Anlaşılması kolay ve tercih edilen öğretim algoritmasıdır. Bu algoritma; hataları geriye doğru çıkıştan girişe azaltmaya çalışmasından dolayı geri yayılım ismini almıştır. Geri yayımlı öğrenme kuralı ağ çıkışındaki mevcut hata düzeyine göre her bir tabakadaki ağırlıkları yeniden hesaplamak için kullanılmaktadır. Bir geri yayımlı ağ modelinde giriş, gizli ve çıkış olmak üzere 3 katman bulunmakla birlikte (Şekil 2.5), problemin özelliklerine göre gizli katman sayısını artırabilmek mümkündür.

Geri yayılım ağında; bir katmandan bir başka katmana, aradaki katmanı atlayarak geçebilmek mümkün değildir. Bir giriş verisinin ağı ilk katmanında yer alan düğümlere uygulandığında en üst katman olan çıkış katmanına erişinceye kadar, bu veri üzerinde çeşitli işlemler gerçekleştirilir. Bu işlemlerin sonucunda elde edilen fiili çıktı, olması gereken çıktı ile karşılaştırılır. Fiili ve olması gereken değerler arasındaki fark, her çıktı düğümü için bir hata sinyali olarak hesaplanır. Hesaplanan hata sinyalleri, her çıktı düğümüne karşı gelen ara katmandaki düğümlere aktarılır. Böylece ara katmandaki düğümlerin her biri toplam hatanın sadece hesaplanan bir kısmını içerir. Bu süreç, her katmandaki düğümler toplam hatanın belirli bir kısmını içerecek şekilde giriş katmanına kadar tekrarlanır. Elde edilen hata sinyalleri temel alınarak, bağlantı ağırlıkları her düğümde yeniden düzenlenir. Bu düzenleme tüm verilerin kodlanabileceği bir duruma ağı yakınsamasını sağlar.

Yukarıda anlatıldığı gibi; ileri besleme safhasında, giriş tabakasındaki nöronlar veri değerlerini doğrudan gizli katmana iletirler. Gizli katmandaki her bir nöron kendi giriş değerlerini ağırlandırarak toplam değer hesap ederler ve bunları bir taşıma fonksiyonu ile işleyerek bir ileriki tabakaya veya doğrudan çıkış katmanına iletirler. Katmanlar arasındaki ağırlıklar başlangıçta rastgele küçük rakamlardan seçilir.

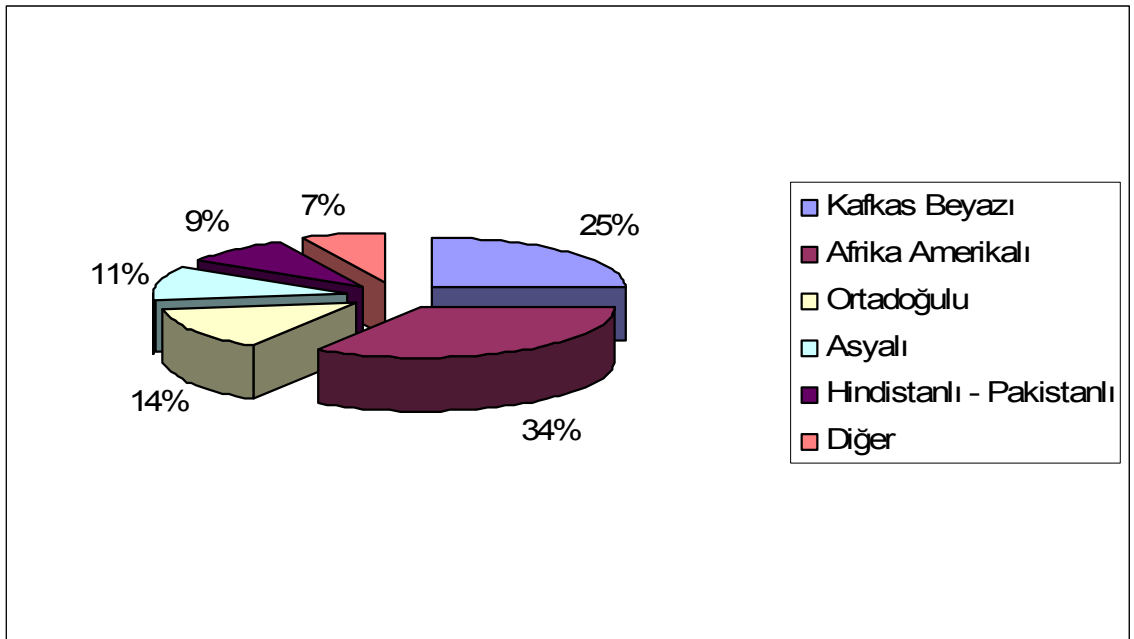
Çıkış katmanındaki, her bir nöron ağırlıklandırılmış değeri hesaplandıktan sonra, bu değer yine taşıma fonksiyonu ile karşılaştırılarak mevcut hata minimize edilmeye çalışılır. Hata değeri belli bir mertebeye ininceye kadar iterasyon işlemine devam edilir ve böylece ağı eğitim aşaması tamamlanmış olur. Katmanlar arası bağlantılardaki ağırlık değerleri eğitimi tamamlamış olan ağıdan alınarak deneme safhasında kullanılmak üzere saklanır.

3. KULLANILAN VERİ TABANI VE GERÇEKLENEN UYGULAMALAR

Bu bölümde ilk olarak tezimde kullanmış olduğum önden çekilmiş yüz imajlarını içeren veritabanının özelliklerinden, daha sonra bu veritabanını kullanarak gerçekleştirdiğim uygulamalardan bahsedilmiştir.

Giriş bölümünde bahsettiğimiz gibi, sistemimizin test edilebilmesi için kontrollü ortamda ve ön cepheden çekilmiş yüz imajlarının bulunduğu bir veritabanı kullanılmıştır. Bu imajların test edilebilmesi için her bir imaj Wayne State Üniversitesi laboratuvarlarında belirli düzenlemelerden geçirilmiştir. Bu sayede yüz imajlarının üzerine düşen ışık seviyelerinin aynı olması ve bütün imajların aynı boyuta getirilmesi sağlanmıştır (Artıklar, 2002).

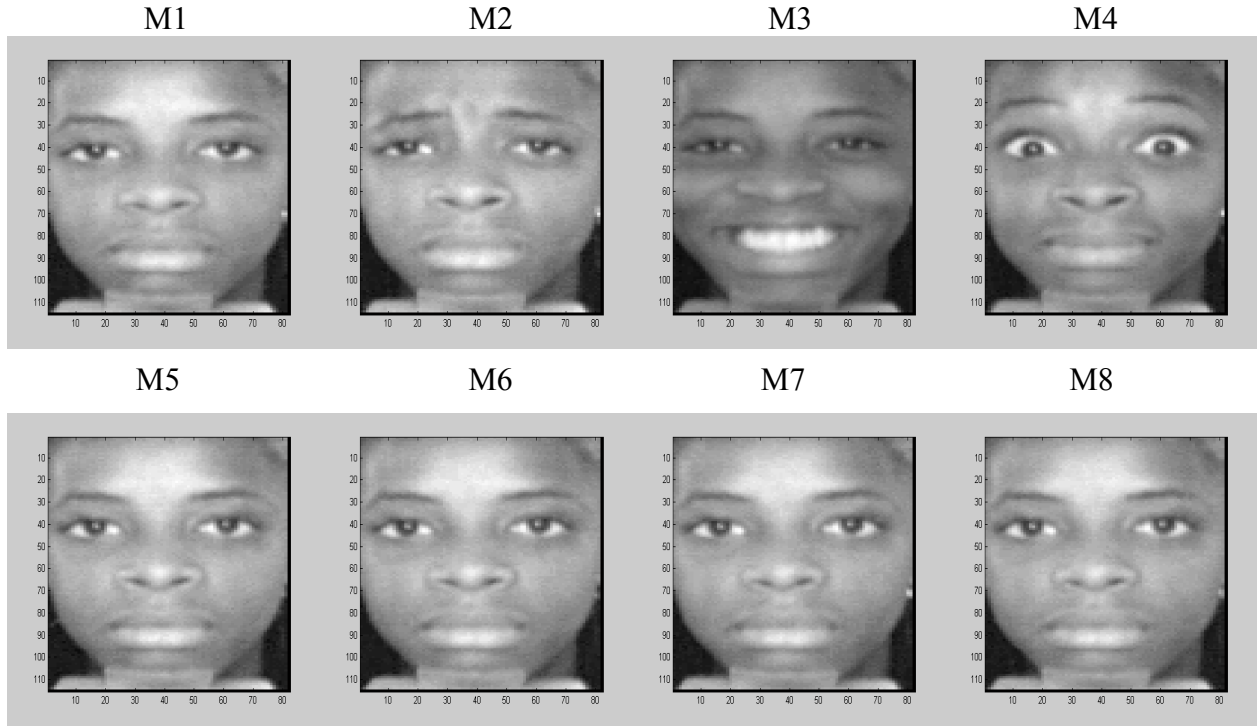
Bu tezde yüz veritabanı olarak, CNNL veritabanı kullanılmıştır. CNNL veritabanı farklı etnik kökenlerdeki 1266 kişinin farklı yüz ifadelerinden oluşmaktadır. CNNL veritabanında bulunan 1266 kişinin 766 adedi erkeklerden, 500 adedi kadınlardan oluşmaktadır. Bu veritabanı Şekil 3.1’de görüldüğü üzere, %25 Kafkas Beyazı, %34’ü Afrika /Amerikalı, %14’ü Ortadoğulu, %11’i Asyalı, %9’u Hindistan-Pakistanlı, %7 diğer farklı etnik kökenlerden oluşmaktadır (Yılmaz vd., 2005).



Şekil 3.1 CNNL veritabanında bulunan 1200 kişinin etnik köken dağılımı

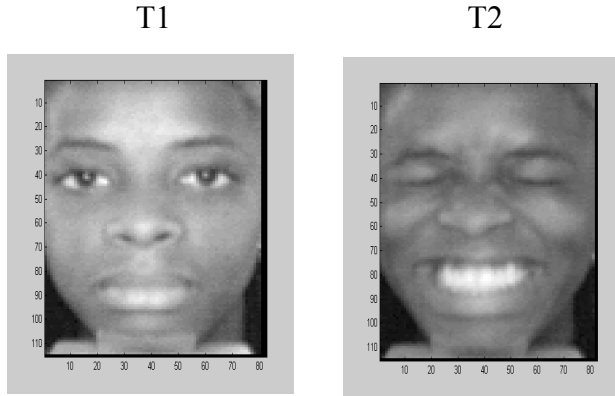
CNNL veritabanında bulunan her bir kişinin 8 adet farklı yüz ifadelerinden oluşan hafıza imajları ve 2 adet boş ve rastgele yüz ifadesinden oluşan test resmi bulunmaktadır (Artıklar,2002). Burada test imajlarının kullanılmasının amacı, kullanılan algoritmanın nasıl bir başarı oranı göstereceğini izlemektir.

Eğitme (Hafıza) Veritabanı Örneği;



M1: Boş ifade, M2: Kızma ifadesi, M3: Gülme ifadesi, M4: Şaşırma ifadesi, M5: Boş ifade1, M6: Boş ifade2, M7: Boş ifade3, M8: Boş ifade4

Test Veritabanı Örneği



T1: Boş ifade, T2: Rastgele yüz ifadesi

Eğitme ve test veritabanlarında kullanılan imajlar 82x115 boyutundadır. İmajlar algoritma ile transpozese alınarak 115x82 durumuna getirilir. Kullanılan imajlar orijinal imajların gri formatına çevrilmesi ile elde edilmiştir.

Bu tez çalışmasında test veritabanında bulunan imajlar ile performans test edilmiştir. Bu test imajları kontrollü ortamdan çekilen 115x82 boyutuna getirilmiş rastgele ve boş yüz ifadelerinden oluşmaktadır. Örnek olarak test veritabanlarında bulunan rastgele veya boş

ifadeler ile hafıza veritabanındaki farklı yüz ifadeleri karşılaştırılmıştır. Bu sayede farklı yüz ifadeleri denenerek hangi yüz ifadesinin daha iyi performans sergilediği belirlenmiştir.

Çalışmalarımızın en önemli kısımlarından biri, performansları artırmak adına farklı yöntemlerin kullanılmasıdır. Bu yöntemlerde, performansı negatif yönde etkileyecek olan parametreler belirlenerek minimize edilmesi amaçlanmıştır. Performansı saptamak için bir eşik değeri belirlenmiştir. Eşik değerini belirleme işlemi, performans ortalamaları temel alınarak hesaplanmıştır. Daha açık bir şekilde ifade edecek olursak, hafıza veritabanında bulunan bir imaj ile test veritabanında bulunan bir imajın aynı olduğunu farz edelim; burada fark sonucumuz sıfır olarak kabul edilir. İmajların her bir pikselinin değeri aynı olduğu için sonuç sıfır olur; yani 1. karşılaştırmada 1 doğru eşleşme gerçekleştirilmiş; %100 başarı oranı elde etmiş oluruz. Tabii ki veritabanında bulunan imajlar birinci bölümde de anlatıldığı üzere farklı imajlardan oluşmaktadır. Her bir test imajı ile hafıza imajlarının farkı alınır ve en küçük sonuç ile eşleştirilir. Bu tezde çıkan sonuçlara göre, kişi yüzlerinin birbirine benzeyip benzemediğini orantılı şekilde görebilmekteyiz. Çıkan sonuç büyük olursa, kişilerin birbirine hiçbir şekilde benzemediği diğer taraftan, bu sonuç ne kadar küçük olursa da test imajı ile hafıza imajının birbirine benzeme oranının o kadar arttığı gözlemlenmiştir.

Ayrıca, başarı oranını artırmak adına yüzdeki bölgelerden kesimler alarak hangi bölümlerin başarı oranını düşürdüğü incelenmiş daha sonra bu bölümler çıkarılmıştır ve bunun sonucunda başarı oranının yeniden önemli ölçüde arttığı gözlemlenmiştir. Diğer taraftan, iki göz bebeği arasındaki mesafenin kullanılması ile başarı oranı artırılmaya çalışılmıştır.

Tezimizin bir sonraki aşamasında ise yapay sinir ağları ile yüz tanıma işlemi gerçekleştirilmiştir. Bu işlemi gerçekleştirmek için sırasıyla; örneklerin toplanması, ağın topolojik yapısının belirlenmesi, öğrenme parametrelerinin belirlenmesi ve öğrenme setinden örneklerin seçilmesi işlemleri gerçekleştirilmiştir.

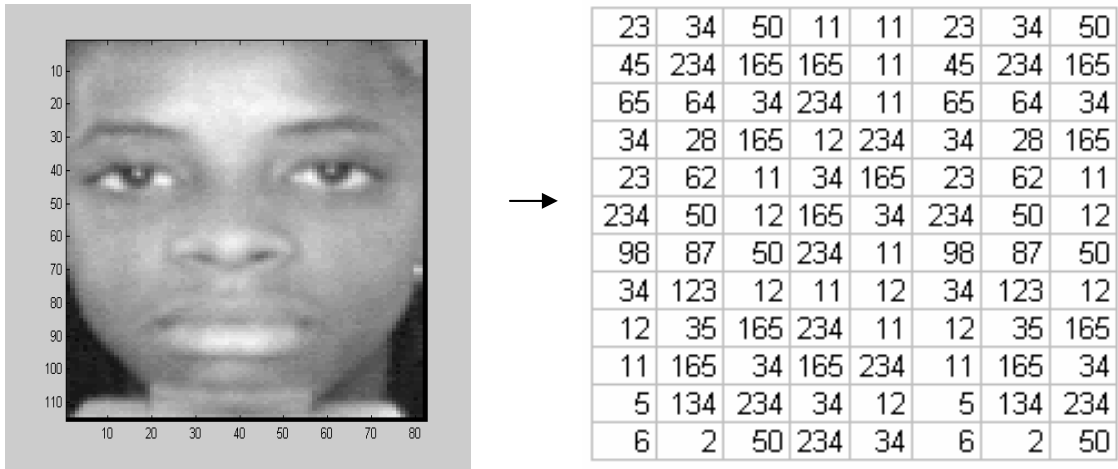
4. BAŞARI ORANI SONUÇLARI

Bu tezde ilk olarak şablon eşleme yöntemi ile yüz tanıma işlemi gerçekleştirilmiştir. Şablon Eşleme metodunda Minokowski metriklerinden birisi olan Şehir Blok Uzaklığı formülü kullanılmıştır (Xiaoyan Mu vd., 2003). Simülasyon sonuçları bu formül kullanılarak elde edilmiştir.

$$d(s, t) = \left(\sum_{i=1}^N |s_i - t_i|^p \right)^{\frac{1}{p}} \quad (4.1)$$

p=1, Şehir blok uzaklığı (City Block Distance, **CBD**)

Şablon eşleme yapmadan önce test imajları ve veritabanı imajları matematiksel ifadeye Şekil 4.1'deki gibi çevrilmiştir. Sonuç olarak her bir imajın sayısal bir sonucu bulunmaktadır. İmajların boyutu 82x115 olduğundan imajda toplam 9430 adet piksel bulunmaktadır. İmajların transpozese alınarak 115x82 şekline getirilir. İmajları gri tona çevirmemizin amacı ise her pikselin renk tonları 0 ve 255 değerleri arasında olması içindir.



Şekil 4.1 İmajların matematiksel ifadeye dönüştürülmesi

Test imajının 115x82'lik matrisi ile hafıza veritabanındaki imaj matrislerinin farkı alınır. Çıkan sonuçların mutlak değeri alınır. Bulunan 115x82'lik matris sonuçlarının sıra ve kolonları toplanarak tek bir sonuç elde edilir (4.1). Elde edilen sonuçlardan en küçük olanı, test imajına en yakın imaj olarak seçilir. Elde edilen başarı oranları en küçük sonuca göre belirlenmiştir.

4.1 Bütün Yüz Kullanılarak Şablon Eşleme

Bu bölümde CNL veritabanındaki ön yüz imajları değiştirilmeden üzerinde işlem yapılmadan orijinal haliyle kullanılmıştır. İlk olarak hafıza veritabanı imajları ile boş ifadeli test imajları karşılaştırılmıştır. Eşlemeler 1000 imaj üzerinden yapılmıştır. Her bir test imajının 1000 adet veritabanı imajı ile karşılaştırılması sonucunda başarı oranları elde edilmiştir.

Bölüm 4.1.1’de test imajı olarak boş ifadeli test imajları kullanılmıştır. Bölüm 4.1.2’de ise test imajı olarak rastgele ifadeli test imajları kullanılmıştır.

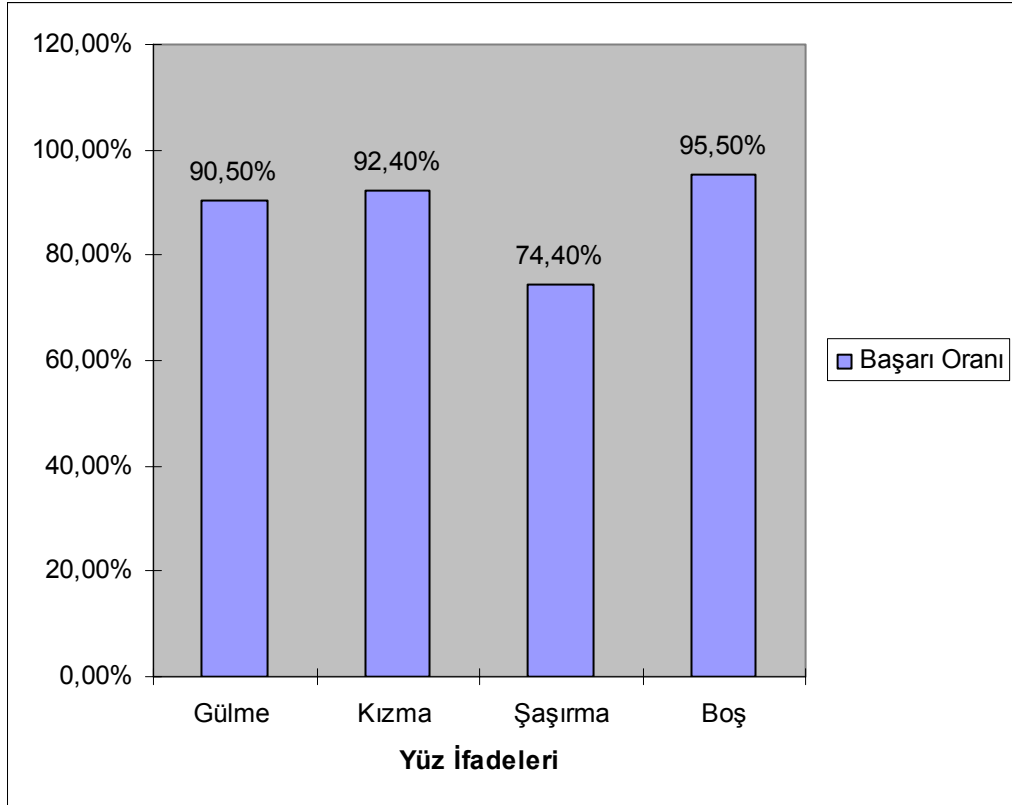
4.1.1 Hafıza Veritabanı İmajlarının, Boş İfadeli Test İmajları ile Bütün Yüz Kullanılarak Test Edilmesi

Veritabanlarındaki ilk 1000 imaj karşılaştırılmış ve aşağıdaki sonuçlar elde edilmiştir. Hafıza veritabanındaki 1000 adet gülme ifadeli imaj ile 1000 adet boş ifadeli test imajı karşılaştırılmıştır. 1000 imajdan 905 adedi doğru, 95 adedi ise hatalı eşleştirilmiştir.

Hafıza veritabanındaki 1000 adet kızma ifadeli imaj ile 1000 adet boş ifadeli test imajı karşılaştırılmıştır. 1000 imajdan 924 adedi doğru, 76 adedi ise hatalı eşleştirilmiştir.

Hafıza veritabanındaki 1000 adet şaşırma ifadeli imaj ile 1000 adet boş ifadeli test imajı karşılaştırılmıştır. 1000 imajdan 744 adedi doğru, 256 adedi ise hatalı eşleştirilmiştir.

Veritabanındaki 1000 adet boş hafıza ifadeli imaj ile 1000 adet boş ifadeli test imajı karşılaştırılmıştır. 1000 imajdan 955 adedi doğru, 45 adedi ise hatalı eşleştirilmiştir. Sonuçlar Şekil 4.2’de verilmiştir.



Şekil 4.2 Boş ifade test imajları ile elde edilen sonuçlar

Test imajı olarak boş ifade test imajları kullanıldığında çıkan sonuçlar yukarıda verilmiştir. Test imajı olarak rastgele ifade test imajı kullanılıp işlemler tekrarlandığında aşağıdaki sonuçlar gözlemlenmiştir.

4.1.2 Hafıza Veritabanı İmajlarının, Rastgele İfadeli Test İmajları ile Bütün Yüz Kullanılarak Test Edilmesi

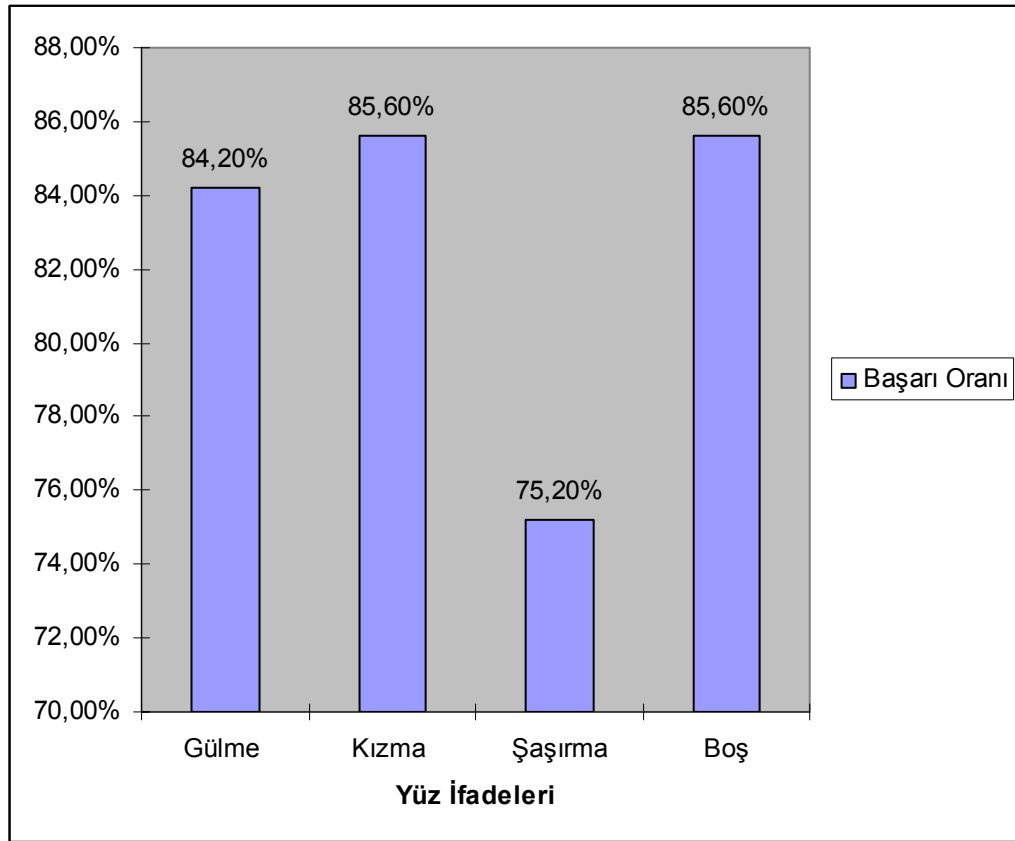
Veritabanlarındaki ilk 1000 imaj karşılaştırılmış ve aşağıdaki sonuçlar elde edilmiştir. Hafıza veritabanındaki 1000 adet gülme ifadeli imaj ile 1000 adet rastgele ifadeli test imajı karşılaştırılmıştır. 1000 imajdan 842 adedi doğru, 158 adedi ise hatalı eşleştirilmiştir.

Hafıza veritabanındaki 1000 adet kızma ifadeli imaj ile 1000 adet rastgele ifadeli test imajı karşılaştırılmıştır. 1000 imajdan 856 adedi doğru, 144 adedi ise hatalı eşleştirilmiştir.

Hafıza veritabanındaki 1000 adet şaşırma ifadeli imaj ile 1000 adet rastgele ifadeli test imajı karşılaştırılmıştır. 1000 imajdan 752 adedi doğru, 248 adedi ise hatalı eşleştirilmiştir.

Veritabanındaki 1000 adet boş ifadeli hafıza imajı ile 1000 adet rastgele ifadeli test imajı karşılaştırılmıştır. 1000 imajdan 856 adedi doğru, 144 adedi ise hatalı eşleştirilmiştir.

Sonuçlar Şekil 4.3’de verilmiştir.



Şekil 4.3 Rastgele ifade test imajları ile elde edilen sonuçlar

İnsan yüz ifadesi çok fazla değiştiği takdirde tanımak daha da zorlaşmaktadır. Şaşırma ifadesinde yüz şekilleri biraz daha belirginleştiğinden tanıma performansı oranının diğer ifadelere oranla daha düşük seviyede olduğu gözlemlenmiştir.

Sıradaki bölümlerde başarı oranını artırmak için bu bölümde elde edilen sonuçlar kullanılmıştır. Sonuçların analizi sırasında, doğru eşlemenin birinci sırada sonuçlanmadığı durumlarda, doğru eşleme çoğu zaman 2. sıra ile 10. sıra arasında sonuçlanmıştır. Çizelge 4.1 bu sonuca bir örnek olarak verilmiştir. Bu çizelgede görüldüğü üzere test veritabanındaki 91-97 numaralı kişiler örnek olarak incelenmiştir. “Eğitim veritabanı & Sonuçlar” kısmında mavi ile yazılan değerler test kişileri ile en iyi eşleşen eğitim setindeki kişileri, kırmızı ile yazılan değerler ise eşleşme sonucunda ortaya çıkan uzaklık sonuçlarını göstermektedir. Çizelge 4.1’de algoritma sonucu bölmesinde yazılan “kesin sonuç” ve “ŞÜPHELİ” ifadeleri; en iyi eşlemedeki uzaklık sonucu ile 2. eşlemede elde edilen uzaklık sonucu arasındaki farka göre belirlenmiştir. Bu iki değer arasındaki fark eşik değerinin üzerinde ise “kesin sonuç”, altında ise “ŞÜPHELİ” ifadeleri atanmıştır. Bu eşik değeri deneme yanılma yoluyla, 5000 olarak belirlenmiştir. “ŞÜPHELİ” olarak eşleşen “test edilen kişi” (95 numaralı kişi) eğitim

veritabanındaki eşleşen en yakın ilk 10 kişi ile tekrar belirli bir algorithmadan geçirilmiştir. Tezde bu algoritma için 2 yöntem kullanılarak başarı oranının artırılması hedeflenmiştir. Bu işlem, Bölüm 4.2 “Yüzdeki Başarı Oranını Düşüren Kısımların Çıkarılması” ve Bölüm 4.3 “Gözler Arasındaki Mesafenin Kullanılması” kullanılarak gerçekleştirilmiştir. Bu yöntemler uygulanarak elde edilen kişinin veritabanı numarası “ihtimal doğru eşleşme” kısmına yazılarak başarı oranı tekrar hesaplanmıştır.

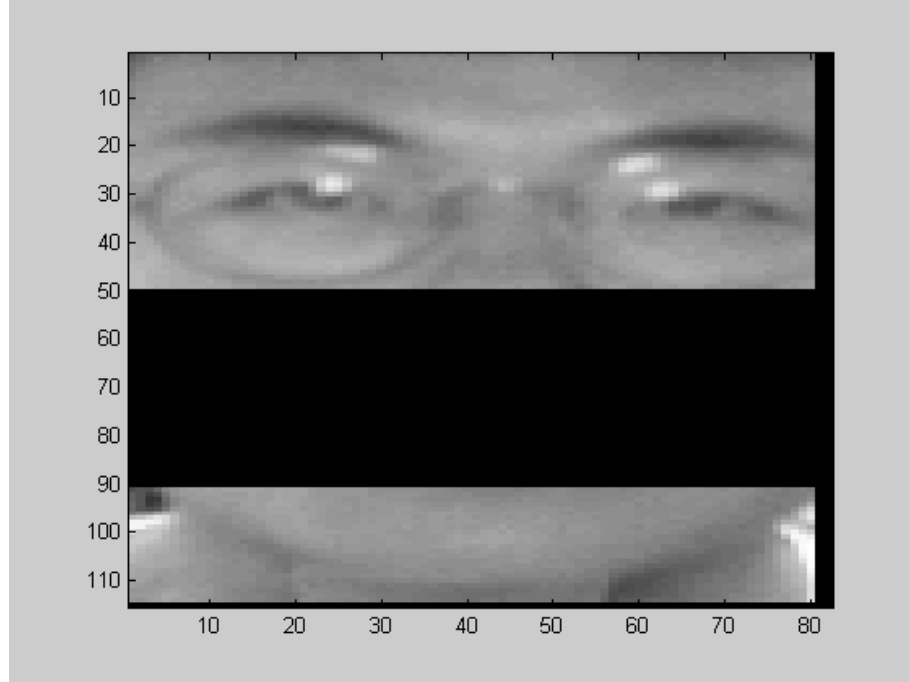
Çizelge 4.1 Örnek bir eşleme sonucu çizelgesi

Test Edilen Kişiler (Test Veritabanı)	EĞİTİM VERİTABANINDAKİ EN İYİ İLK ON EŞLEŞME & UZAKLIK SONUÇLARI										Bölüm 5.2 ve Bölüm 5.3 için gereken parametreler	İhtimal doğru eşleşme
	1.	2.	3.	4.	5.	5.	7.	8.	9.	10.	Algoritma Sonucu	
91. Kişi	91.00	66.00	78.00	65.00	90.00	87.00	65.00	49.00	57.00	61.00		En iyi Eşleşme ile 2. Eşleşme arasındaki fark 3422
	137395.00	158931.00	162711.00	166609.00	173463.00	175320.00	182396.00	182675.00	184253.00	207161.00	kesin sonuç	
92. Kişi	92.00	95.00	60.00	62.00	67.00	55.00	68.00	89.00	56.00	66.00		
	76527.00	190474.00	196316.00	199052.00	198491.00	204076.00	205966.00	214333.00	216920.00	221960.00	kesin sonuç	
93. Kişi	93.00	68.00	67.00	97.00	95.00	56.00	64.00	14.00	89.00	55.00		
	87707.00	155342.00	165467.00	189811.00	206220.00	206772.00	216680.00	223908.00	232907.00	237172.00	kesin sonuç	
94. Kişi	94.00	64.00	92.00	68.00	67.00	95.00	93.00	56.00	97.00	62.00		
	134571.00	227266.00	228829.00	234820.00	237731.00	239202.00	248871.00	254510.00	256365.00	266088.00	kesin sonuç	
95. Kişi	68.00	95.00	67.00	56.00	97.00	55.00	93.00	89.00	62.00	92.00		
	138477.00	142899.00	156334.00	163363.00	166126.00	190707.00	194064.00	196304.00	201073.00	208898.00	(SÜPHELI)	
96. Kişi	96.00	60.00	69.00	63.00	70.00	53.00	66.00	59.00	79.00	71.00		95
	85039.00	157656.00	166244.00	186340.00	170253.00	178161.00	181071.00	188946.00	191401.00	193683.00	kesin sonuç	
97. Kişi	97.00	66.00	56.00	67.00	95.00	62.00	14.00	89.00	93.00	55.00		
	98850.00	167269.00	176483.00	203759.00	209156.00	210663.00	213039.00	216196.00	217632.00	225997.00	kesin sonuç	

4.2 Yüzdeki Belirli Bölgelerin Kullanılması ile Başarı Oranı Artırımının Gerçekleştirilmesi

Yüz tanımda başarı oranını belirleyen en önemli kısımlar ağız ve burun kısımlarıdır. Bu bölgeler, yüzün farklı bölgelerindeki kısımlar çıkarılarak test edilmesi ile elde edilen başarı oranlarına göre belirlenmiştir. Yaklaşık 10 bölge çıkarılarak işlem yapılmış ve en iyi başarı oranı, burun ve ağız bölgelerinin çıkarılması ile gerçekleştirilmiştir.

Yeni bir algoritma yazılarak en küçük 10 sonuç bu algoritma ile tekrar kontrol edilmiştir. Bu algoritma yazılırken hem test hem de hafıza imajlarında yüzün belirli bölgeleri alınarak, tekrar şablon eşleme işlemi gerçekleştirilmiştir. Burun ve ağız kısımlarının her bir pikseline Şekil 4.4’de görüldüğü gibi sıfır değeri atanarak simülasyonlar gerçekleştirilmiştir.



Şekil 4.4 Ağız ve burun bölgelerine sıfır piksel değeri atanması.

4.2.1 Boş İfadeli Test İmajları ve Hafıza İmajlarının Belirli Bölgelerinin Çıkarılması ile Test Edilmesi

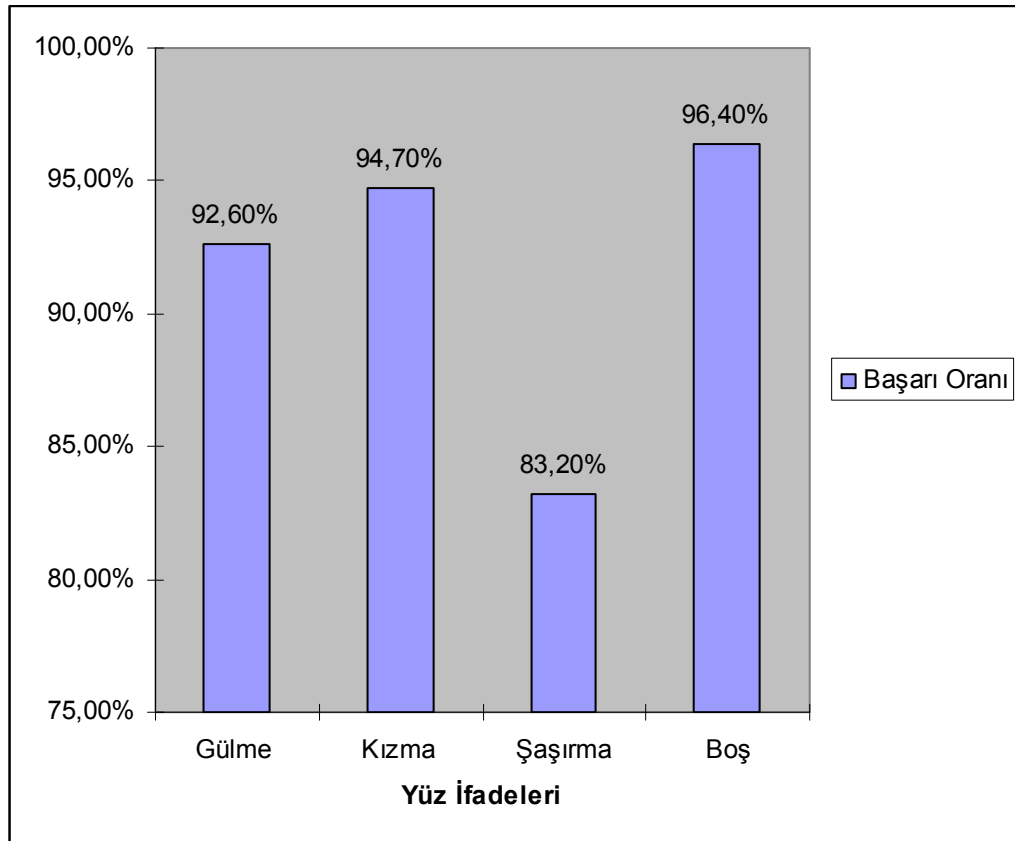
Veritabanlarındaki ilk 1000 imaj yüzün ağız ve burun bölgeleri dikkate alınmadan karşılaştırılmış ve aşağıdaki sonuçlar elde edilmiştir. Hafıza veritabanındaki 1000 adet gülme ifadeli imaj ile 1000 adet boş ifadeli test imajı karşılaştırılmıştır. 1000 imajdan 926 adedi doğru, 74 adedi ise hatalı eşleştirilmiştir.

Hafıza veritabanındaki 1000 adet kızma ifadeli imaj ile 1000 adet boş ifadeli test imajı karşılaştırılmıştır. 1000 imajdan 947 adedi doğru, 53 adedi ise hatalı eşleştirilmiştir.

Hafıza veritabanındaki 1000 adet şaşırma ifadeli imaj ile 1000 adet boş ifadeli test imajı karşılaştırılmıştır. 1000 imajdan 832 adedi doğru, 168 adedi ise hatalı eşleştirilmiştir.

Veritabanındaki 1000 adet boş ifadeli hafıza imajı ile 1000 adet boş ifadeli test imajı karşılaştırılmıştır. 1000 imajdan 964 adedi doğru, 36 adedi ise hatalı eşleştirilmiştir.

Sonuçlar Şekil 4.5'te verilmiştir.



Şekil 4.5 Boş ifade test imajları ile elde edilen başarı oranları

4.2.2 Rastgele İfadeli Test İmajları ve Hafıza İmajlarının Belirli Bölgelerinin Çıkarılması ile Test Edilmesi

Veritabanlarındaki ilk 1000 imaj yüzün ağız ve burun bölgeleri dikkate alınmadan karşılaştırılmış ve aşağıdaki sonuçlar elde edilmiştir. Hafıza veritabanındaki 1000 adet gülme ifadeli imaj ile 1000 adet rastgele ifadeli test imajı karşılaştırılmıştır. 1000 imajdan 854 adedi doğru, 146 adedi ise hatalı eşleştirilmiştir.

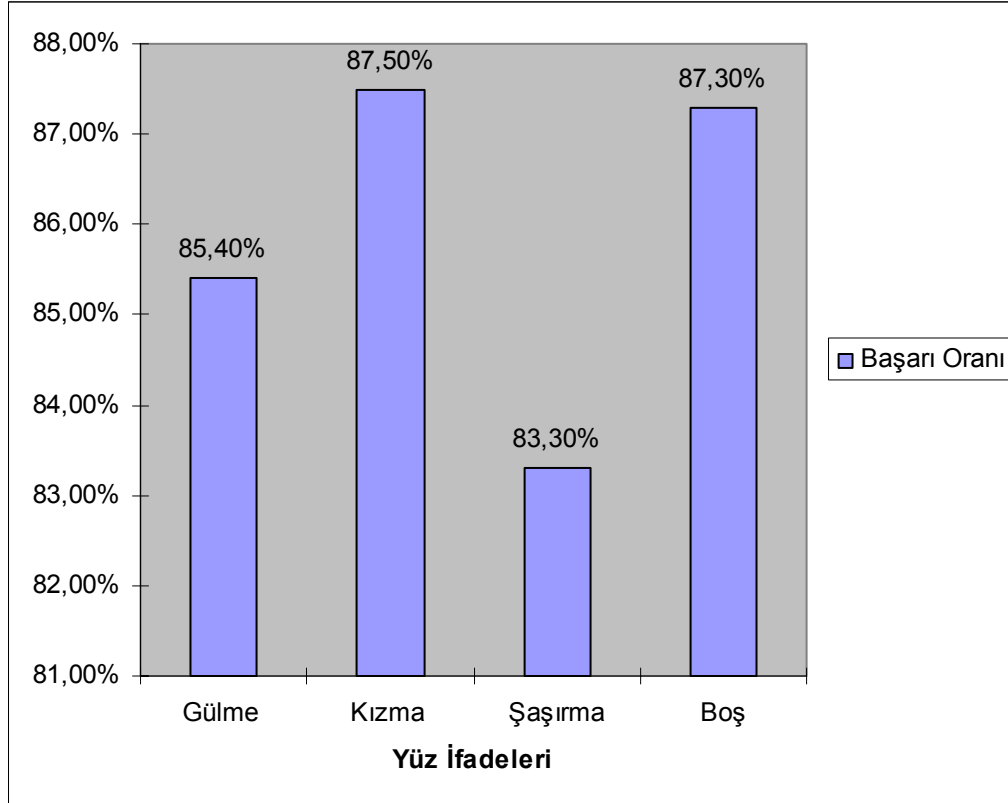
Hafıza veritabanındaki 1000 adet kızma ifadeli imaj ile 1000 adet rastgele ifadeli test imajı

karşılaştırılmıştır. 1000 imajdan 875 adedi doğru, 125 adedi ise hatalı eşleştirilmiştir.

Hafıza veritabanındaki 1000 adet şaşırma ifadeli imaj ile 1000 adet rastgele ifadeli test imajı karşılaştırılmıştır. 1000 imajdan 833 adedi doğru, 167 adedi ise hatalı eşleştirilmiştir.

Veritabanındaki 1000 adet boş hafıza ifadeleri ile 1000 adet rastgele ifadeli test imajı karşılaştırılmıştır. 1000 imajdan 873 adedi doğru, 127 adedi ise hatalı eşleştirilmiştir.

Sonuçlar Şekil 4.6’da verilmiştir.



Şekil 4.6 Rastgele ifadeli test imajları ile elde edilen başarı oranları

4.3 Gözler Arasındaki Mesafenin Kullanılması ile Başarı Oranının İncelenmesi

Bu yöntem yüzün geometrik yapısı ile ilgili olduğundan, hem şablon ve hem de gometrik özellik kullanılarak yüz tanıma işlemi gerçekleştirilmiştir. Çizelge 4.1’de anlatıldığı üzere hatalı eşleştiği kabul edilen test resmi, bu yöntem kullanılarak yeniden incelenmiştir. Bölüm 4.2’de belirli bölgelerin çıkarılması ile sistemin başarı oranının artırılması sağlanmıştır. Bu bölümde ise, hatalı eşleştiği kabul edilen test imajı ve ilk eşleşme sonucunda elde edilen ilk 10 eğitim veritabanı imajlarının gözbebekleri arasındaki mesafeler belirlenip başarı oranı tekrar incelenmiştir.

İlk olarak imajlardaki gözbebeklerinin bulunma işlemi gerçekleştirilmiştir. Piksel arasındaki

farkın en çok bulunduğu kısım işaretlenerek göz bebeği olduğuna karar verilmiştir. Çünkü yüzdeki piksel farkının en çok bulunduğu kısım gözdür. Göz bebeği en siyah ve göz bebeğinin etrafında bulunan bölüm ise en beyaz kısımlardır. Hata oranının minimize edilmesi için sadece gözlerin bulunduğu kısımlar üzerinde işlem yapılmıştır (Şekil 4.7 ve 4.8).

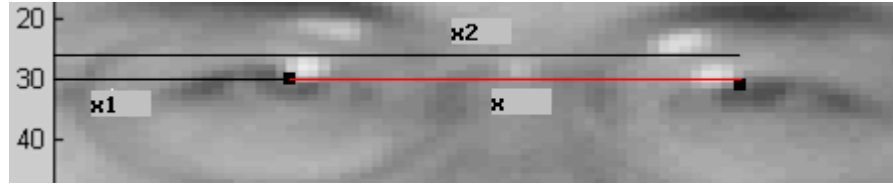


Şekil 4.7 Sağ göz



Şekil 4.8 Sol göz

Her iki şekilde, yan yana olan iki piksel arasındaki fark maksimum ise bu kısımlar işaretlenmiştir. İki gözbebeği bulunup işaretlendikten sonra bu noktaların $x=0$ eksenine uzaklıkları olan x_1 ve x_2 değerleri bulunmuştur (Şekil 4.9).



Şekil 4.9 Gözler arasındaki mesafenin hesaplanması

İki gözbebeği arasındaki uzaklık x olarak adlandırılırsa, x ; x_2 ile x_1 'in farkı olarak hesaplanmıştır.

İki gözbebeği arasındaki uzaklık hesaplandıktan sonra yüz tanıma işlemine geçilmiştir. Daha öncede anlatıldığı üzere; Bölüm 4.1 de elde edilen sonuçlarda şüpheli olarak eşleşen kişiler, burada x değerleri kullanılarak tekrardan eşleştirilmiştir.

Örnek vererek anlatacak olursak; Çizelge 4.1'de hatalı eşleşen 95. test resmindeki kişinin x değeri, daha sonra test imajına en çok benzeyen 10 eğitim veritabanı imajlarının ayrı ayrı x değerleri hesaplanmıştır. Çizelge 4.2'de bütün değerler verilmiştir.

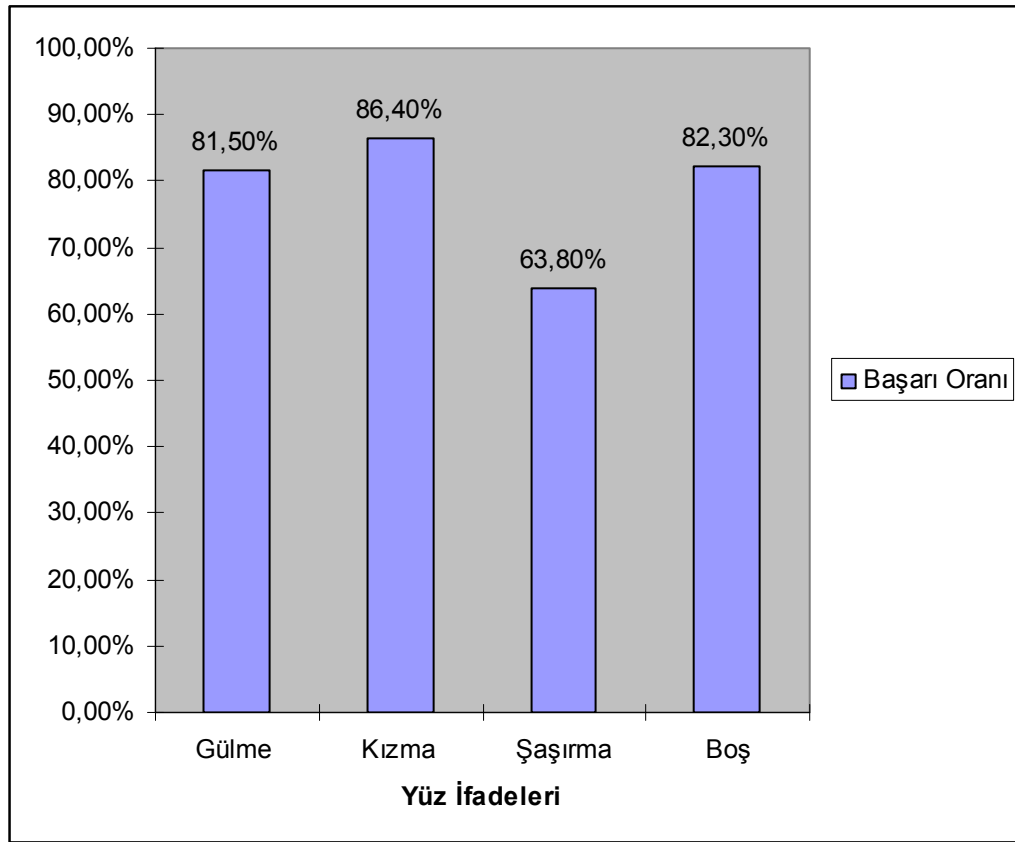
Çizelge 4.2 Her bir kişiye ait 2 gözbebeği arasındaki mesafe

Test & x değeri	Test imajına en çok benzeyen 10 eğitim veritabanı imajı & x değerleri									
95. Kişi	68. Kişi	95. Kişi	67. Kişi	56. Kişi	97. Kişi	55. Kişi	93. Kişi	89. Kişi	62. Kişi	92. Kişi
117.342	110.451	116.987	112.543	111.543	119.932	113.542	116.241	115.321	114.417	110.368
	6.891	0.355	4.799	5.799	2.590	3.800	1.101	2.021	2.925	6.974

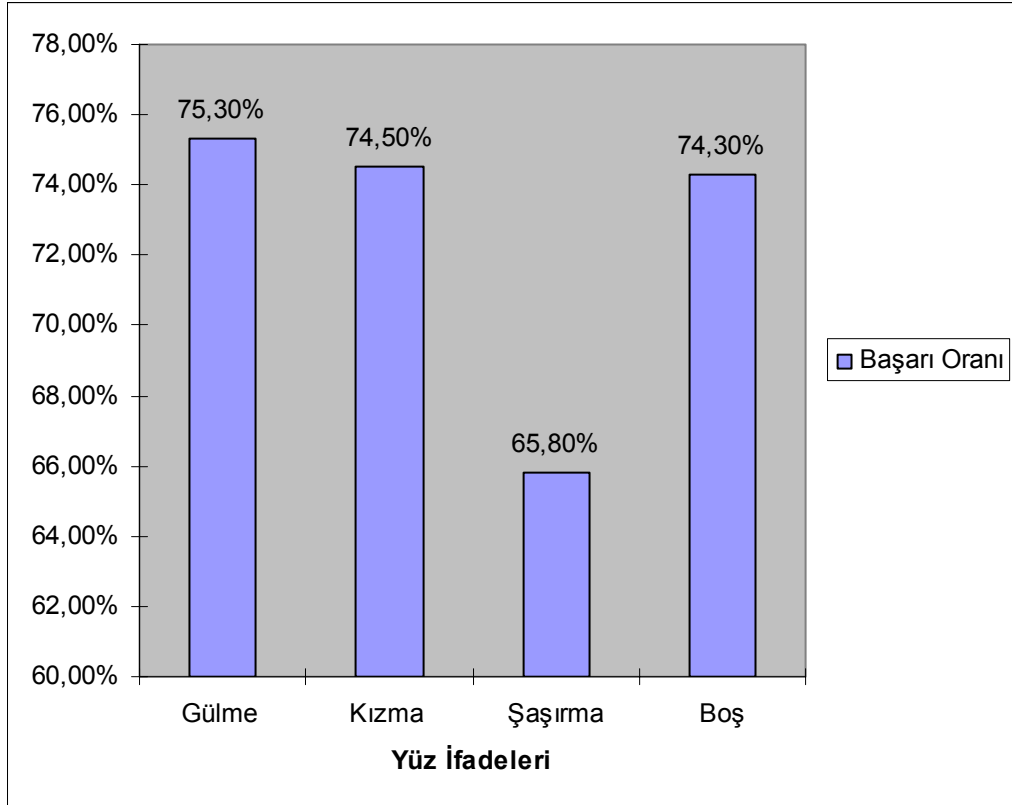
Test veritabanında bulunan 95. kişinin x değerinden eğitim veritabanındaki x değerleri

çıkarılıp mutlak değerleri alınır. Elde edilen en küçük fark test kişi ile eşleştirilir. Çizelge 4.2'den de görüldüğü üzere en iyi sonuç 95. eğitim veritabanı resmi kullanılarak elde edilen sonuçtur. İlk eşlemede 95. test resmi hatalı şekilde 68. veritabanı resmi ile eşleşmesine rağmen, gözler arasındaki mesafenin kullanılması ile tekrar eşleme yapıldığında doğru eşleştirilmiştir.

1000 adet eğitim veritabanındaki farklı ifadelerin, 1000 adet boş ifadeli test imajları ve 1000 adet rastgele ifadeli imajları ile ayrı ayrı karşılaştırılmaları sonucu elde edilen başarı oranı şekilleri Şekil 4.10 ve Şekil 4.11'de verilmiştir.



Şekil 4.10 Gözler arasındaki mesafenin kullanılması ile elde edilen başarı oranları (Boş ifadeli test imajları)



Şekil 4.11 Gözler arasındaki mesafenin kullanılması ile elde edilen başarı oranları (Rasgele ifadeli test imajları)

Yukarıdaki şekillerden de anlaşılacağı üzere bu metotla başarı oranı artışı gerçekleşmemiştir. Aksine performanslar büyük ölçüde düşmüştür.

4.4 Yapay Sinir Ağlarının Sistemimizde Kullanılışı ve Başarı Oranı Sonuçları

CNNL veritabanındaki kişilerin tanınması için çok katmalı yapay sinir ağı kullanılarak bir yüz tanıma sisteminin tasarımı gerçekleştirilmiştir. Yapay sinir ağlarının sistemimiz içinde kullanılması için aşağıdaki maddeler sırasıyla gerçekleştirilmiştir.

4.4.1 Örneklerin Toplanması

Bu adımda ağın eğitilmesi için eğitim seti, test edilmesi içinde test seti oluşturulmuştur. Sistemin uygulaması Matlab 7.0.4 üzerinden olmuştur. Yapay sinir ağıımız, CNNL eğitim veritabanındaki 800 adet farklı yüz imajı ile eğitilmiştir. Daha sonra ağ, eğitim ve test setlerindeki imajlar ile test edilmiştir. Ağın eğitilmesi sırasında test seti ağa hiç gösterilmez. Eğitim setindeki örnekler tek tek gösterilerek ağın olayı öğrenmesi sağlanmıştır. Ağ olayı öğrendikten sonra test setindeki örnekler gösterilerek ağın performansı ölçülmüştür. Hiç görmediği örnekler karşısındaki başarısı, ağın iyi öğrenip öğrenmediğini ortaya koymaktadır.

İlk olarak özellik çıkarma işlemi uygulanmıştır. Bunun için kenar çıkarma “*edge detection*” kullanılmıştır. Kenar çıkarma, nesne tanımda çok önemli rol oynar. İnsan görme sisteminde, nesnelerin geometrik özelliklerini tanıma süreci, nesnelerin dış çizgilerine göz gezdirilmesi ile başlar. Bu yaklaşım yapay görme sistemlerine uyarlandığında, şayet nesnenin sınırları başarılı bir şekilde izlenebilirse nesne tanımda çok daha iyi sonuçlar alınabileceği açıktır. Buna göre, nesne tanıma işleminde kenar çıkarma işlemi önemli bir rol oynar. Çoğu görüntüler somut nesneleri içermez ve bu görüntüleri anlamak onların yapısal özelliklerine bağlıdır. Yapısal özelliklerin çıkarımı ise kenar çıkarma ile ilgilidir (Bağcıoğulları vd., 1994).

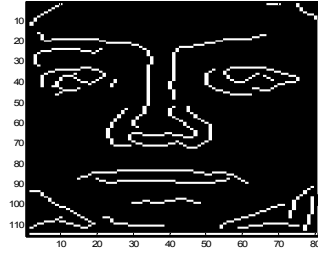
Kenar çıkarma, özellik çıkarma işleminde önemli bir alt problemdir. Amacı, görüntü içerisinde komşuları ile farklı yoğunluklara sahip piksellerin tespit edilmesidir. Bu işlem genellikle iki aşamadan oluşur. Birinci aşamada, güncel pikselin yoğunluğu, komşu pikseller ile karşılaştırılarak yoğunluk değişimleri tespit edilir. Geleneksel kenar çıkarıcılar olarak bilinen Sobel operatörü, Prewitt operatörü ve Robert operatörü (Abdou vd., 1979; Costa vd., 2001) bu aşamaya ihtiyaç duyarlar. İkinci aşamada ise, elde edilen değişimlerin önceden belirlenmiş bir eşik değeri ile karşılaştırılarak önemli bir değişim olup olmadığı tespit edilir ve bu tespite göre güncel pikselin kenar olup olmadığı hakkında karar verilir.

Geleneksel kenar çıkarma algoritmaları (Sobel, Prewitt ve Robert), istenilen bu özelliklere çoğu zaman doyurucu bir sonuç vermezler. Bu algoritmalarda sık sık yanlış kenar yeri, köşeleri tam çıkaramama ve bulanık kenar çıkarma gibi problemler ile karşılaşılır (Pinho vd., 1994).

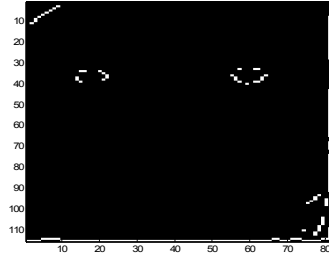
Geleneksel kenar çıkarma algoritmaları içerisinde en iyilerden biri olarak bilinen Canny kenar çıkarma algoritmasına göre:

- İyi kenar çıkarma: Yüksek oranda kenar olma olasılığına sahip noktaları “kenar”, düşük oranda kenar olma olasılığına sahip noktaları ise “kenar değil” olarak sınıflandırmak.
- İyi kenar yeri belirleme: Geçerli kenar yerine yakın olan noktaları kenar noktası olarak işaretlemek.
- Tek nokta kalınlığında kenar çıkarma: Tek boyutlu bir işaretle kenar çizgisini tek nokta kalınlığında göstermek.

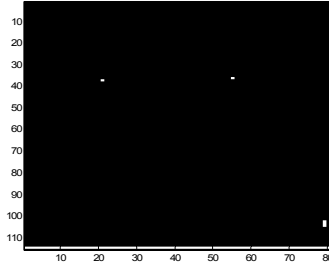
Canny kenar çıkartma algoritması bu kriterlere doyurucu bir yanıt verebilmek için kenar çıkartma işlemine başlamadan önce görüntüyü Gaussian fonksiyonunun ilk türevine karşılık gelen bir filtreden geçirir. Görüntü içerisindeki gürültü elendikten sonra filtrelenmiş görüntüye Canny operatörü uygulanır (Talü, F vd.).



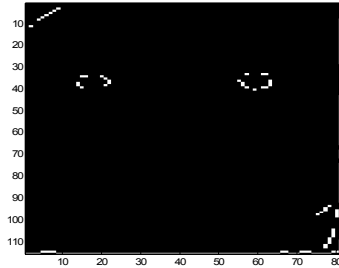
Şekil 4.12 Canny metodunun yüz imajlarında uygulanması



Şekil 4.13 Sobel metodunun yüz imajlarında uygulanması



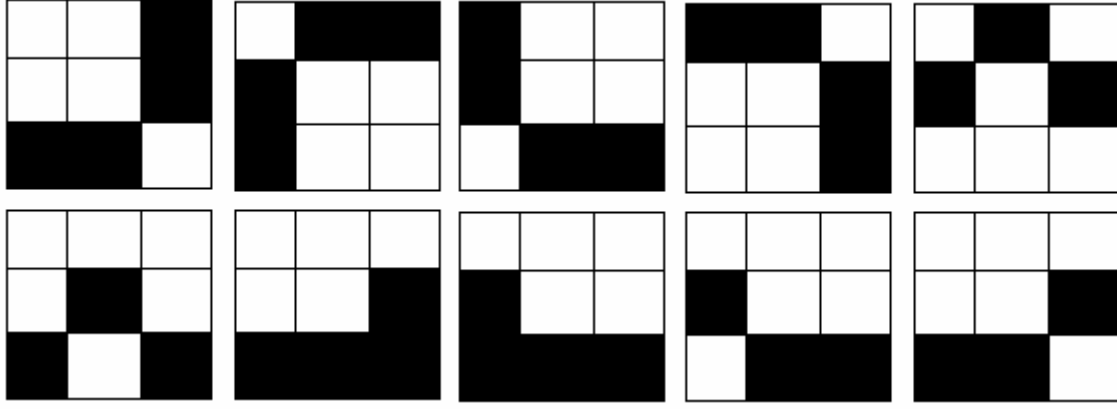
Şekil 4.14 Roberts metodunun yüz imajlarında uygulanması



Şekil 4.15 Prewitt metodunun yüz imajlarında uygulanması

Yukarıdaki şekillerden de görüldüğü üzere en fazla örnek Canny metodu kullanılarak elde edilmiştir.

Birbirinden farklı maskeler belirlenerek her bir maskenin, Canny kenar belirleme metodu uygulanmış imajlarda birebir eşleşme sayısı yapay sinir ağıımıza girdi olarak verilmiştir (Çizelge 4.3). Kullanılan maskeler Şekil 4.16'da verilmiştir.



Şekil 4.16 Yapay sinir ağı girişi için kullanılan maskeler

Çizelge 4.3’de görüldüğü üzere sütunlar her bir resimde maskelerin birebir eşleşme sayısını göstermektedir. Örnek olarak Çizelge 4.3’de 1 numaralı kişinin 8 farklı ifadesi 1-8 sırası arasındadır. 1 numaralı maske (M1), eğitim veritabanındaki 1 numaralı kişinin boş ifadeli imajında 7 adet, gülme ifadeli imajında 1 adet, kızma ifadeli imajında 2 adet, şaşırma ifadesinde 4 adet birebir eşleştiğini göstermektedir. Eğitim veritabanındaki 2 numaralı kişinin 8 farklı ifadesi 9-16 numaraları arası olup her bir imajda 10 maskenin birebir eşleşme sayılarını göstermektedir. 100 kişinin arasında test yapılmıştır. Her bir kişinin 8 farklı imajı olduğundan dolayı 800 sıra bulunmaktadır. Çıkış çizelgesinden de görüldüğü üzere ikilik tabanda hedefler girilmiştir. İlk 8 sıraya 0000001 değeri atanmıştır, yani CNNL veritabanındaki 1 numaralı kişiyi göstermektedir. Son 8 sıraya ise 1100100 değeri atanmıştır, yani CNNL veritabanındaki 100 numaralı kişiyi göstermektedir.

Çizelge 4.3 YSA giriş matrisi

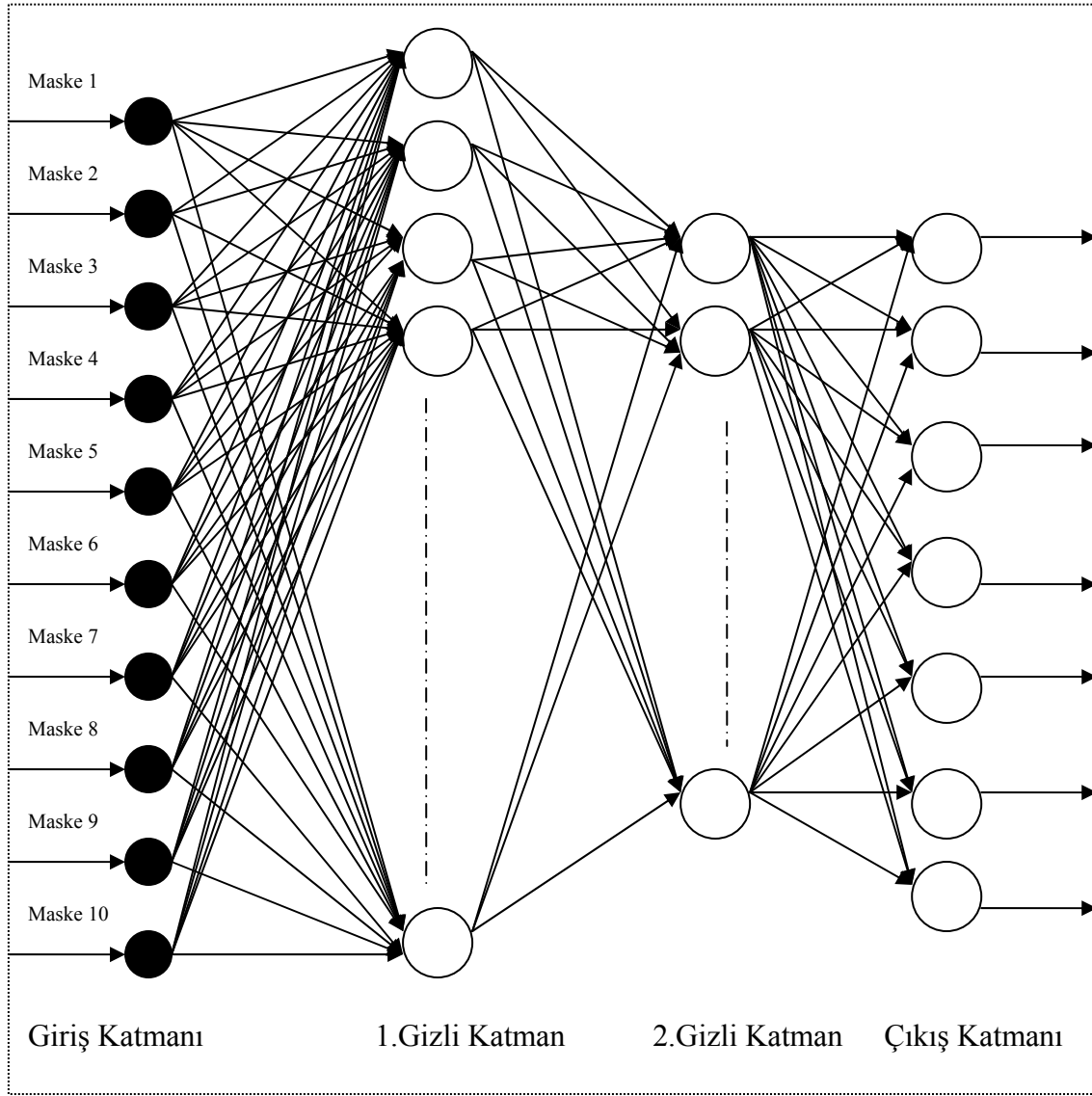
	M1	M2			M10
1	7	5			2
2	1	3			4
3	2	1			0
4	4	8			1
....					
797	4	4			7
798	5	4			3
799	5	7			4
800	5	7			4

Çizelge 4.4 YSA Çıkış (Hedef) Matrisi

	1	2	3	4	5	6	7
1	0	0	0	0	0	0	1
2	0	0	0	0	0	0	1
3	0	0	0	0	0	0	1
4	0	0	0	0	0	0	1
5	0	0	0	0	0	0	1
6	0	0	0	0	0	0	1
7	0	0	0	0	0	0	1
8	0	0	0	0	0	0	1
9	0	0	0	0	0	1	0
10	0	0	0	0	0	1	0
11	0	0	0	0	0	1	0
.....							
.....							
797	1	1	0	0	1	0	0
798	1	1	0	0	1	0	0
799	1	1	0	0	1	0	0
800	1	1	0	0	1	0	0

4.4.2 Ağın Topolojik Yapısı

Kullanmış olduğumuz çok katmanlı yapay sinir ağı 7 çıkıştan oluşmaktadır. Giriş katmanı ve gizli katman proses elemanı sayıları çıkışta elde edilen etkinliğe bağlı olarak belirlenmiştir. Giriş katmanındaki proses elemanı sayısı 40, gizli katmandaki proses elemanı sayısı 20 olarak belirlenmiştir. Kullanılan ağın topolojik yapısı Şekil 4.14'te verilmiştir.



Şekil 4.17 Kullanılan ağın topolojik yapısı

Giriş ve çıkış matrisleri elde edildikten sonra aşağıdaki kodla yapay sinir ağı oluşturulmuştur.

```
net=newff(minmax(giris), [40 20 7], {'tansig' 'tansig' 'purelin'}, 'traingd');
```

Ağın eğitilmesi ise aşağıdaki kodlarla gerçekleştirilmiştir.

```
net.trainparam.epochs=5000;
```

```
net.trainparam.goal=0.01;
```

```
net.trainparam.lr=0.2;
```

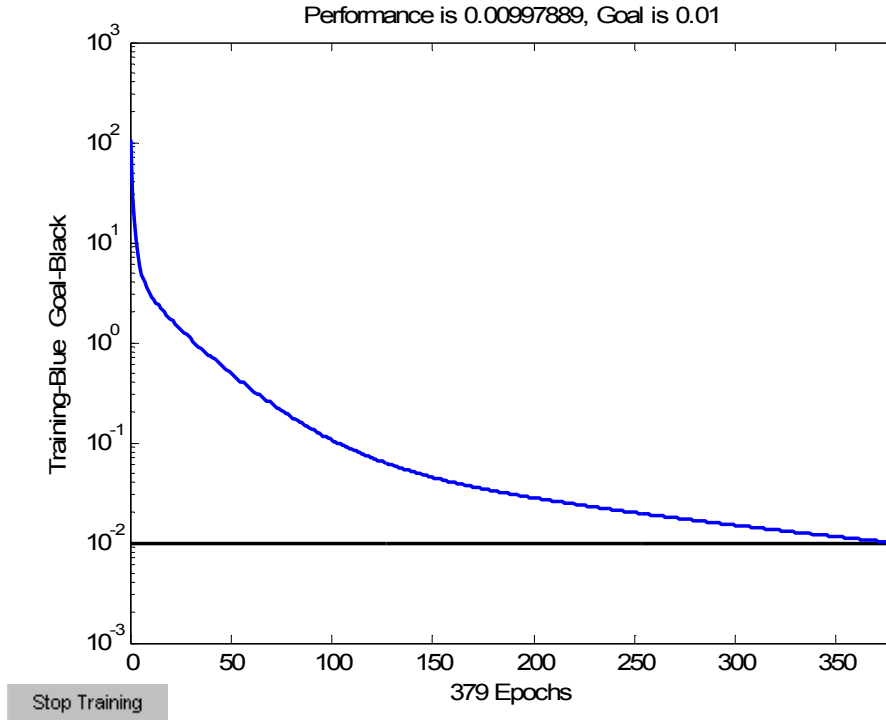
```
net.trainparam.max_fail=5;
```

```
net.trainparam.max_grad=1e-010;
```

```
net=train(net,giris,hedef);
```

Yüz tanıma ağının eğitiminde; giriş katmanı, 2 gizli katman ve çıkış katmanından oluşan bir YSA yapısı kullanılmıştır. Gizli katman sayısı ve bu katmandaki proses elemanı sayısı değişik sınamalarda hatanın istenen değere düşmesini sağlayacak şekilde seçilmiştir. Transfer fonksiyonu olarak tanjant sigmoid fonksiyonu, öğrenme algoritması olarak da geriye yayılım

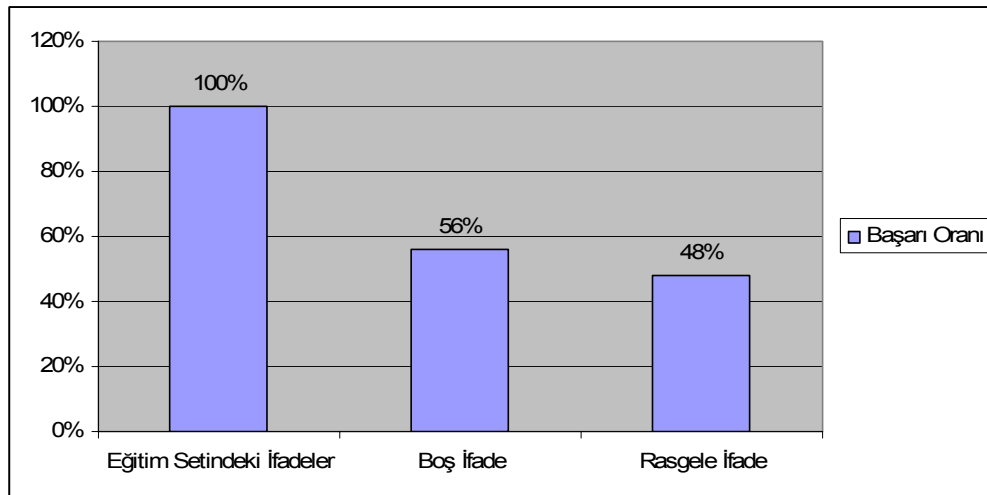
algoritmalarından biri olan gradient bazlı algoritma kullanılmıştır. İterasyon sayısı 5000 olup hedeflenen hata oranı 0.01 olarak ayarlanmıştır.



Şekil 4.18 Yüz tanıma ağına ait eğitim performans grafiği

Sistemimizi test etmek için kullanılacak olan yüz imajları aşağıdaki kod ile test edilmiştir.
sonuc=sim(net,test);

Sonuçlar Excel'e otomatik olarak atılmıştır. Eğitilmiş olan yüz imajları test edildiğinde %100 başarı sağlanmıştır. Fakat test resimlerinden boş ifadeli imajlar ile test edildiğinde %56, rastgele yüz ifadesi ile test edildiğinde ise %48 oranında bir başarı oranı ortaya çıkmıştır (Şekil 4.19).



Şekil 4.19 YSA başarı oranı sonuçları

5. BOYUT KÜÇÜLTME İŞLEMİ İLE İŞLEM ZAMANININ AZALTILMASI

Bu bölümde, veritabanında kullanılan imajların boyutları belirli oranlarda küçültülerek karşılaştırmalar yapılmış ve boyut küçültme işleminin yüz tanıma sisteminin performansına etkisi araştırılmıştır. Bu bölümdeki sonuçlar, şablon eşleme metodu kullanılarak elde edilen sonuçlardır. Burada boyut küçültmenin performansı doğrusal bir şekilde etkileyip etkilemediği de incelenmiştir. Asıl amacımız, sistem başarı oranını düşürmeyip süreci mümkün olduğu kadar azaltmaktır.

İmaj boyutlarını küçültme işlemi interpolasyon yöntemi kullanılarak yapılmıştır. İnterpolasyon imaj pikselleri değiştiğinde imaj pikselleri arasındaki lokasyonun değerini tahmin eden bir proses işlemidir. Örneğin, bir imaj boyutu küçültüldüğünde orijinalinden daha az piksel içerir. İnterpolasyon işlemi, yeni küçültülen imajdaki piksellerin değerini hesaplayarak, kaybı en aza kadar indirmeye yaramaktadır. Daha basit bir dille anlatmak istersek, interpolasyon ara değer anlamına gelmektedir. En basit haliyle; 1,?,3,4,5 dizisinde yerine ne gelir sorusunun cevabını bulma yöntemi ara değeri yani interpolasyon yöntemidir. 1,2,3,4,? dizisinde, ? işareti yerine gelen sayıyı bulma işlemi de ekstrapolasyon işlemidir. İnterpolasyon kuramı aşağıdaki gibidir;

f fonksiyonu (a,b) aralığında sürekli ve $y_1 < y_2$ iken,

$f(a)=y_1$, $f(b)=y_2$ olsun,

$a < b$ ise

$x_1 < f(c) < x_2$ {veya $x_1 > f(c) > x_3$ } koşulunu gerçekleyen en az 1 adet c vardır ki $a < c < b$ koşuluna uyar.

Gözde daha iyi canlanması için yukarıda verilen ara değer örneğini ele alalım (1,?,2,3,4,5). Ve bunu x-y düzleminde çizelim. (noktalar: (1,1), (2,?), (3,3), (4,4)..)

Geometrik benzerlikten faydalanalım: $f(2) = f(1) + (f(3) - f(1)) / [(3)-(1)] = 1+(3-1)/2 = 2$.

Çok çeşitli interpolasyon yöntemleri mevcuttur, bunlardan en çok kullanılanlar bikübik, bilineer ve en yakın komşu (Nearest Neighbour) yöntemleridir. Bu yöntemler genel olarak kalite kaybolmadan görüntüyü büyötmek için kullanılır fakat bu yöntemler tezde imajın kalitesi kaybolmadan küçölmek için kullanılmıştır.

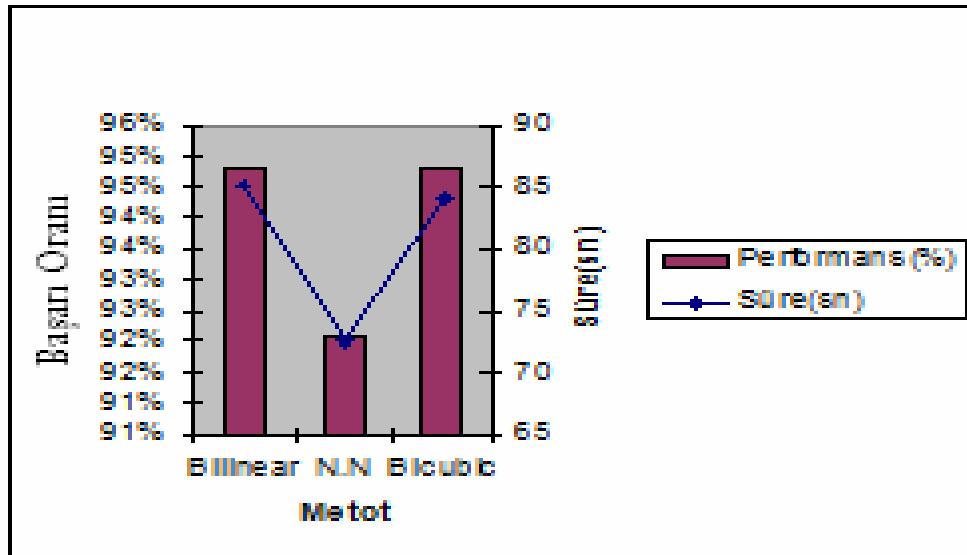
En yakın komşu, görüntüdeki pikselleri azaltıp çoğaltan hızlı ancak daha az kesin yöntemdir. Bu yöntem, keskin kenarları korumak ve daha küçük dosya oluşturmak için kenar

yumuşatması olmayan kenarları içeren çizimlerde kullanılmak içindir. Ancak bu yöntem, görüntüyü deforme ettiğinizde veya ölçeklendirdiğinizde ya da seçim üzerinde çok sayıda değişiklik gerçekleştirdiğinizde görünür olan pürüzlü etkiler üretebilir. En yakın komşuluk yöntemi aynı zamanda sıfırinci dereceden interpolasyon yöntemi olarak da adlandırılır. Bu yöntem ile yeni piksel gri değerleri giriş görüntüsündeki en yakında yer alan değerlerin kopyalanması ile elde edilir, dolayısı ile giriş görüntüsü ile aynı değeri alır. Bundan dolayı görüntünün radyometrik karakteri değişmez. Diğer taraftan sonuç görüntüde blok yapı bozulur.

Bilineer, çevreleyen piksellerin renk değerlerinin ortalamasını alarak piksel ekleyen yöntemdir. Orta kalitede sonuçlar üretir. Bilineer interpolasyon sonuçları görüntüyü yumuşatır. Görüntüdeki keskin sınırlar sonuç görüntüde yumuşar. Bu obje yakalama tekniğinin görüntüde uygulanmasını güçleştirir.

Bikübik, çevreleyen piksellerin değerlerinin incelemesini temel alan daha kesin bir yöntemdir. Bikübik, daha karmaşık hesaplamalar kullanarak En Yakın Komşu veya Bilineer yöntemine göre daha yumuşak ton geçişleri üretir. Bu yöntem en elverişli yöntemdir (adobe.com.tr; goruntuisleme.org).

Tezde bu üç interpolasyon metodu karşılaştırılmıştır. 1200 imaj arasında yapılan araştırma ile bikübik yönteminin diğer interpolasyon metotlarına göre biraz daha iyi performans gösterdiği Şekil 5.1’de görülmektedir.



Şekil 5.1 İnterpolasyon prosesi

Boyut küçültme işleminde amacımız, boyutu başarı oranı etkilenmeyecek şekilde küçülttebildiğimiz kadar küçültmek böylelikle hızı maksimuma çıkarmaktır.

Çizelge 5.1’de görüldüğü üzere veritabanındaki imajlarda gerçekleştirilecek olan küçültme oranları ve bu küçültme oranlarına karşılık gelen piksel boyutları verilmiştir.

Çizelge 5.1 İmaj boyutlarının piksel boyutlarına karşılık gelen değerleri

İmajların yüzdesel boyutu (%)	Piksel Boyutu
100	82 X 115
50	41 X 58
25	21 X 29
20	16 X 23
10	8 X 12

Yukarıdaki çizelgeden de görüldüğü üzere imajların boyutları %100 den, % 50, %25 ve %20’ye daha sonra %10’a düşürülmüştür.

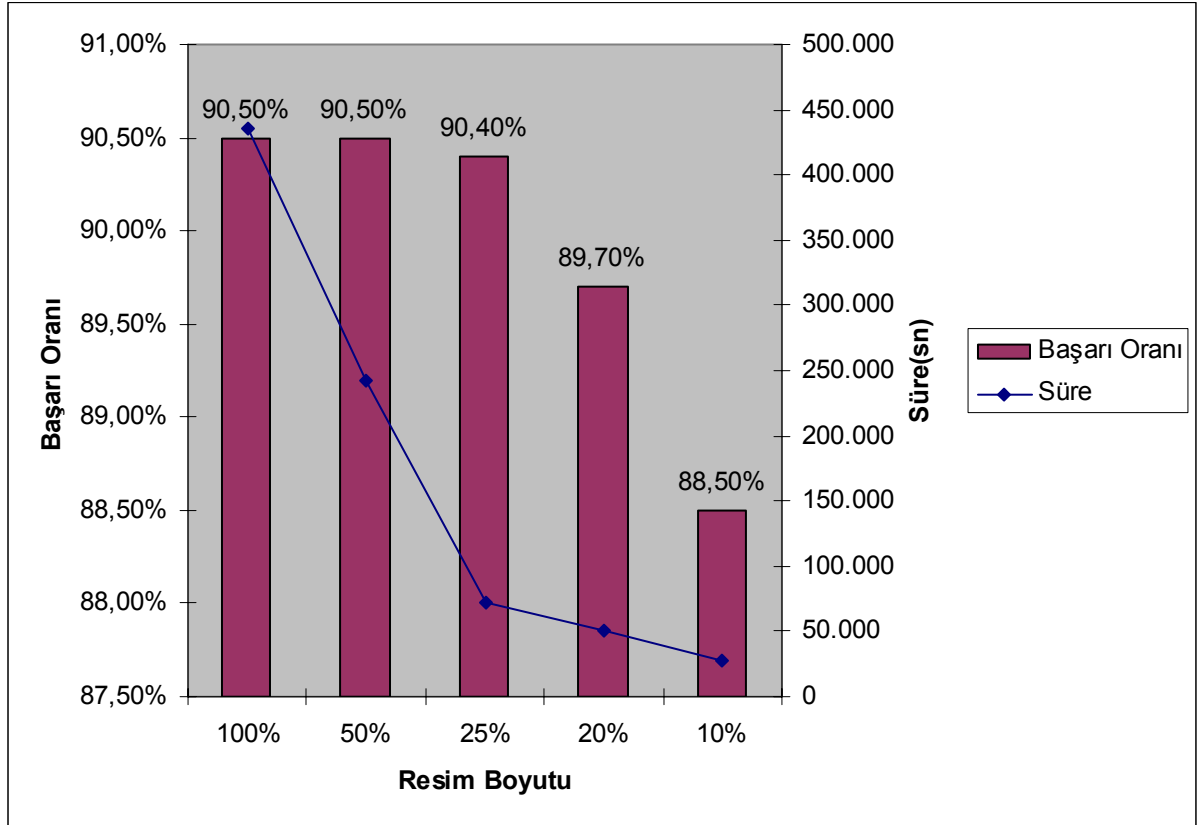
Tezde yapılan bütün işlemler Pentium IV 1700 GHz lik bilgisayarla gerçekleştirilmiştir. Bilgisayar sistemi 512 MB RAM’e sahiptir. Simulasyonlar Matlab ile yapılmıştır.

5.1 Simulasyon Sonuçları

Bu bölüm 4 kısımdan oluşmaktadır. Test veritabanındaki boş ve rastgele ifadeli imajlar kullanılarak algoritma denenmiştir. İlk kısımda gülme ifadeleri ikinci kısımda boş ifadeler, üçüncü kısımda kızma ifadeleri ve dördüncü kısımda ise şaşırma ifadelerinin boyutları Çizelge 5.1’de gösterildiği gibi sırasıyla küçültülerek performanslar incelenmiştir.

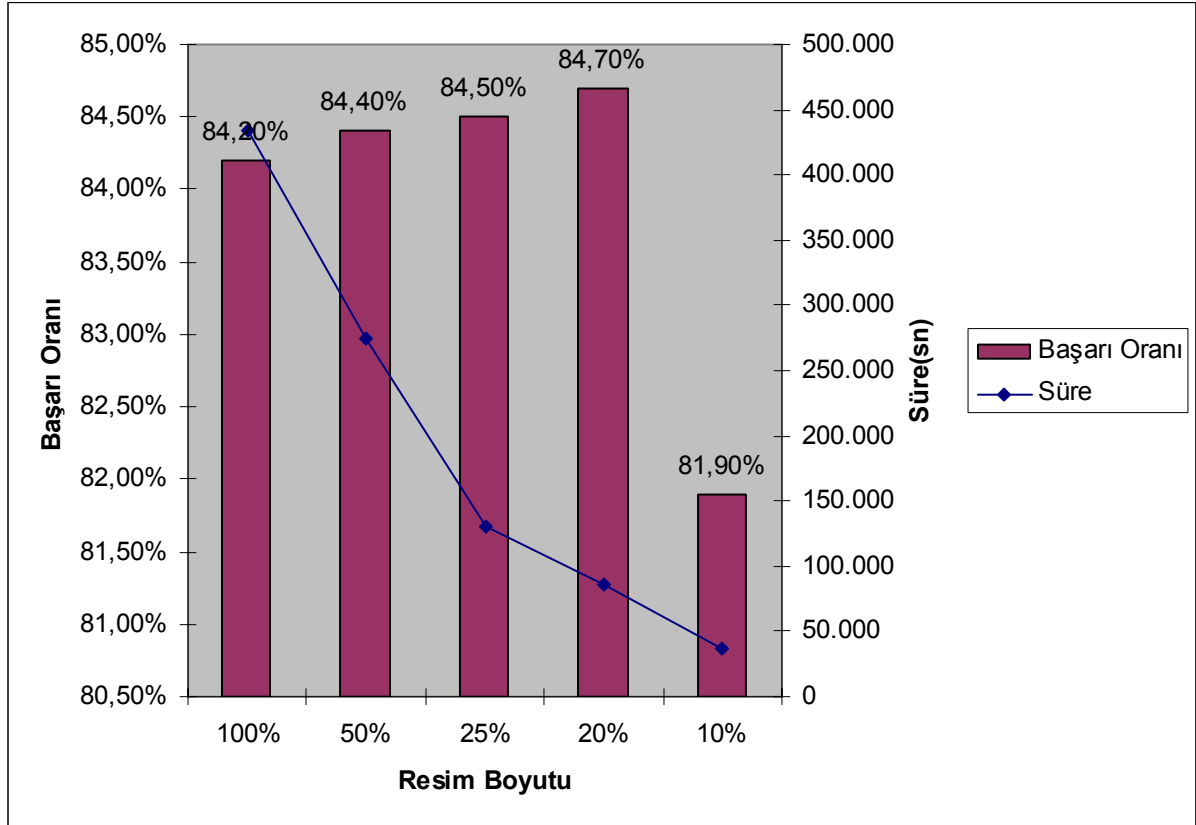
5.1.1 Gülme İfadeli İmajların Performans Sonuçları

Bu bölümde test imajları, hafıza veritabanındaki gülme ifadeleri ile karşılaştırılmıştır. Boş ve rastgele ifadeli test imajları ile gülme ifadeli imajlar farklı boyutlarda karşılaştırılmıştır. Başarı oranının süreye karşı grafiği Şekil 5.2 ve Şekil 5.3’te verilmiştir.



Şekil 5.2 Boş ifadeli test imajları ile gülme ifadelerinin karşılaştırılmasının süre başarı oranı grafiği

Şekil 5.2’de görüldüğü üzere en iyi performans %50 boyutlu imajlarda çıkmıştır. %25 boyutlu imajlarda ise başarı oranı çok küçük bir oranda düşmüştür buna karşılık süre altıda bir oranda azalmıştır. Boyutlar daha çok küçüldükçe başarı oranının da düştüğü gözlemlenmektedir.

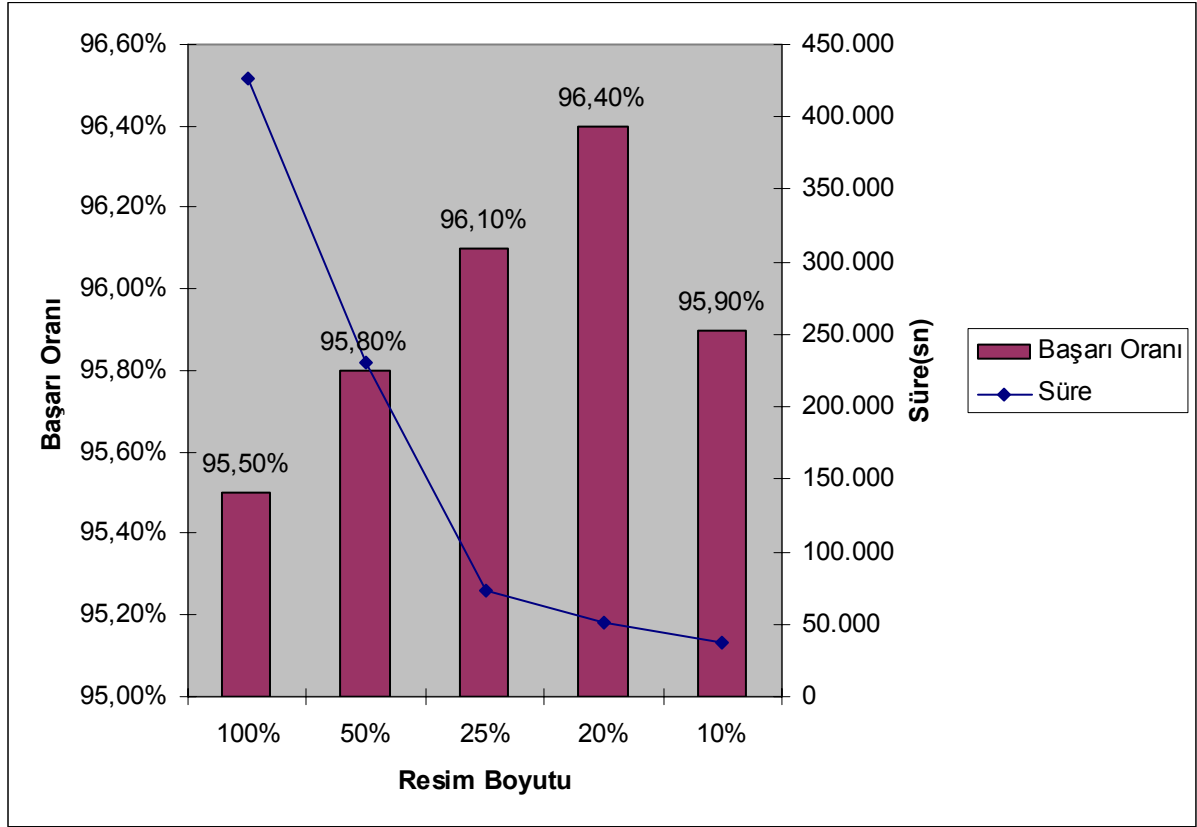


Şekil 5.3 Rastgele ifadeli test imajları ile gülme ifadelerinin karşılaştırılmasının süre başarı oranı grafiği

Şekil 5.3’de görüldüğü üzere en iyi performans %20 boyutlu imajlarda çıkmıştır. Başarı oranı boyut küçüldükçe artmış ve süre beşte bir oranda azalmıştır.

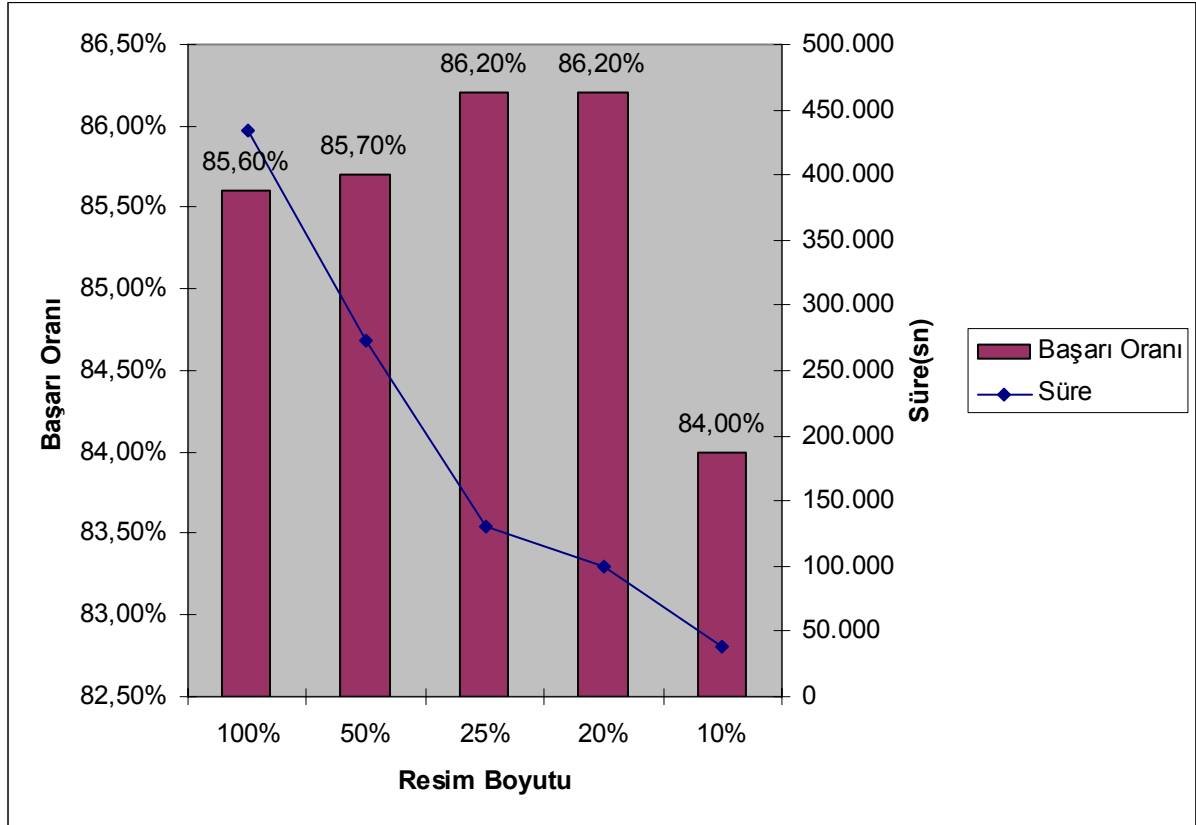
5.1.2 Boş İfadeli İmajların Performans Sonuçları

Bu bölümde test imajları, hafıza veritabanındaki boş ifadeler ile karşılaştırılmıştır. Boş ve rastgele ifadeli test imajları ile eğitim veritabanındaki boş ifadeli imajlar farklı boyutlarda karşılaştırılmıştır. Başarı oranının süreye karşı grafiği Şekil 5.4 ve Şekil 5.5’te verilmiştir.



Şekil 5.4 Boş ifadeli test imajları ile boş ifadeli hafıza imajlarının karşılaştırılmasının süre başarı oranı grafiği

Şekil 5.4'te görüldüğü üzere en iyi performans %20 boyutlu imajlarda çıkmıştır. Başarı oranı boyut küçüldükçe artmış ve süre dokuzda bir oranda azalmıştır.

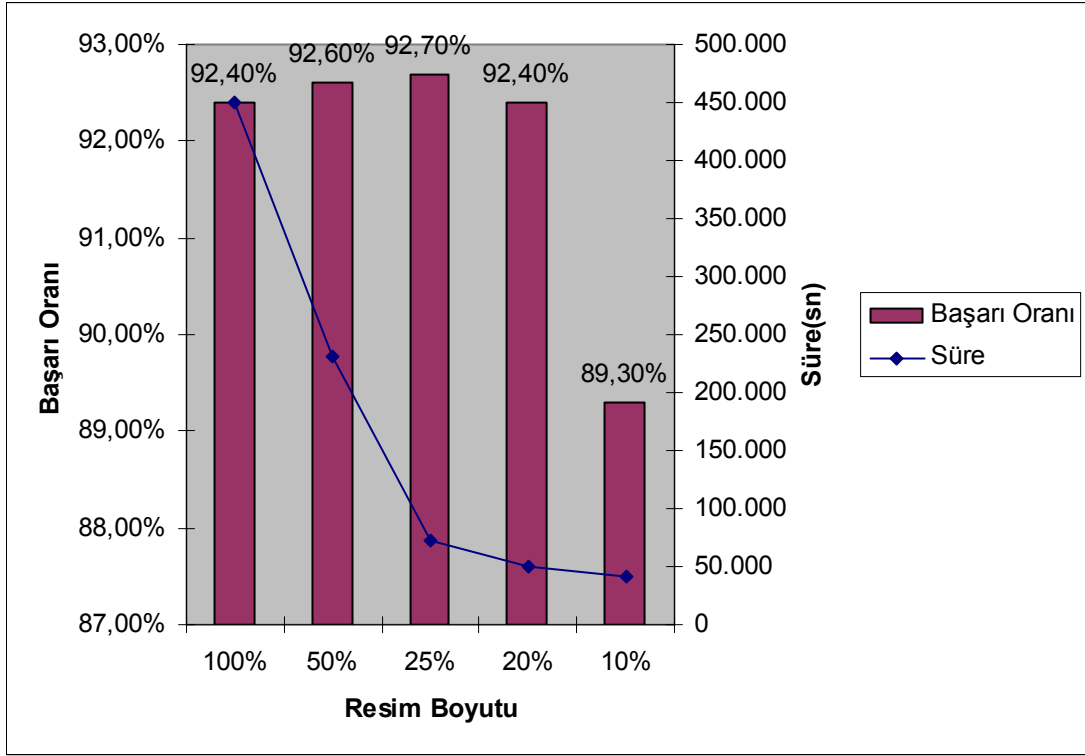


Şekil 5.5 Rastgele ifadeli test imajları ile boş ifadeli hafıza imajlarının karşılaştırılmasının süre başarı oranı grafiği

Şekil 5.5’de görüldüğü üzere en iyi performans %20 boyutlu imajlarda çıkmıştır. Başarı oranı boyut küçüldükçe artmış ve süre dörtte bir oranda azalmıştır.

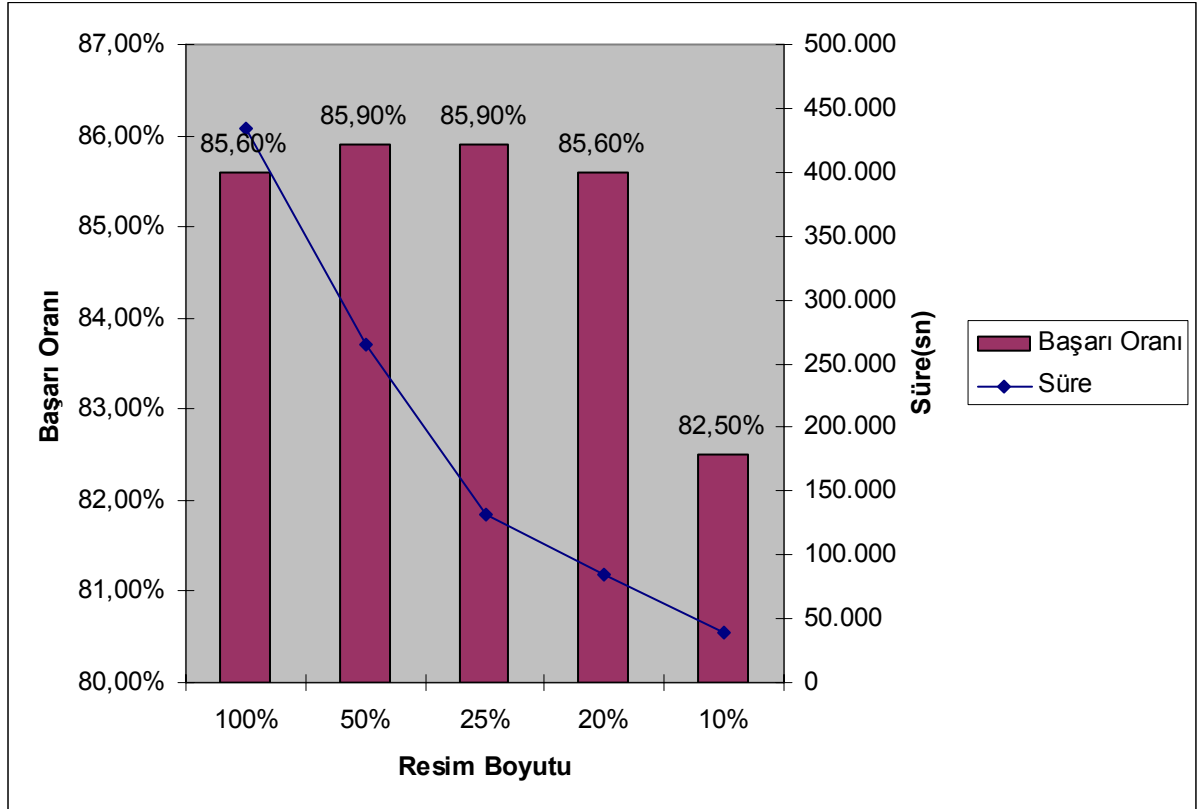
5.1.3 Kızma İfadeli İmajların Performans Sonuçları

Bu bölümde test imajları, hafıza veritabanındaki kızma ifadeleri ile karşılaştırılmıştır. Boş ve rastgele ifadeli test imajları ile kızma ifadeli imajlar farklı boyutlarda karşılaştırılmıştır. Başarı oranının süreye karşı grafiği Şekil 5.6 ve Şekil 5.7’de verilmiştir..



Şekil 5.6 Boş ifadeli test imajları ile kızma ifadelerinin karşılaştırılmasının süre başarı oranı grafiği

Şekil 5.6'da görüldüğü üzere en iyi performans %25 boyutlu imajlarda çıkmıştır. Başarı oranı %20 boyuta kadar küçüldükçe artmış ve süre dokuzda bir oranda azalmıştır.

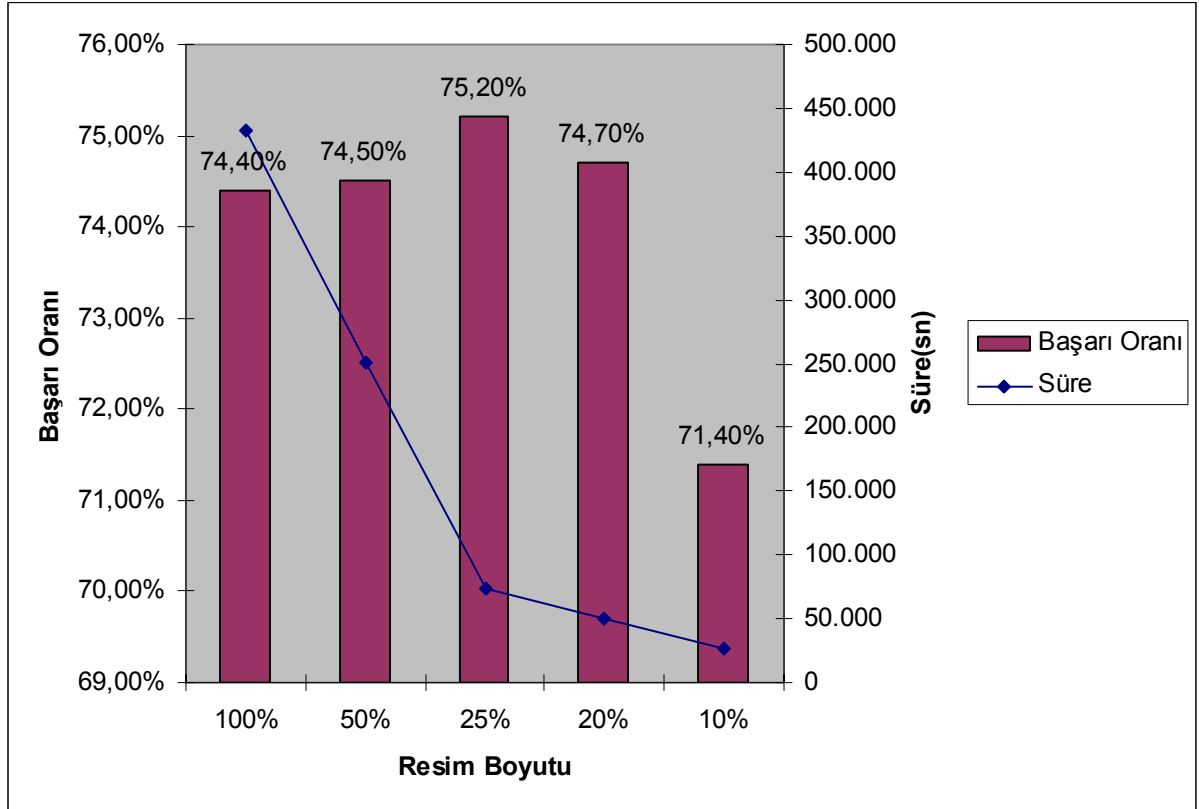


Şekil 5.7 Rastgele ifadeli test imajları ile kızma ifadeli imajların karşılaştırılmasının süre başarı oranı grafiği

Şekil 5.7’de görüldüğü üzere en iyi performans %25 boyutlu imajlarda çıkmıştır. Başarı oranı %20 boyuta kadar küçüldükçe artmış ve süre beşte bir oranda azalmıştır.

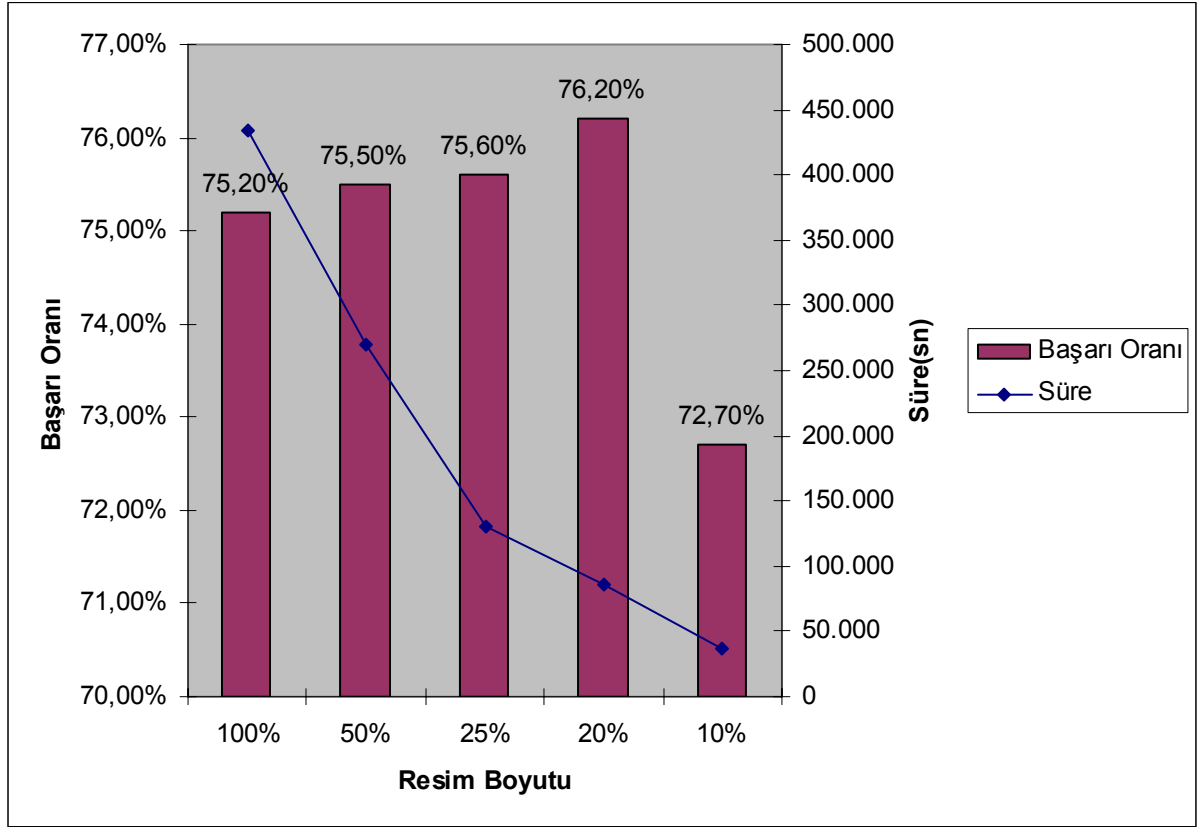
5.1.4 Şaşırma İfadeli İmajların Performans Sonuçları

Bu bölümde test imajları, hafıza veritabanındaki şaşırma ifadeleri ile karşılaştırılmıştır. Boş ve rastgele ifadeli test imajları ile gülme ifadeli imajlar farklı boyutlarda karşılaştırılmıştır. Başarı oranının süreye karşı grafiği Şekil 5.8 ve Şekil 5.9’da verilmiştir.



Şekil 5.8 Boş ifadeli test imajları ile şaşırma ifadeli imajların karşılaştırılmasının süre başarı oranı grafiği

Şekil 5.8’de görüldüğü üzere en iyi performans %25 boyutlu imajlarda çıkmıştır. Başarı oranı %25 boyuta kadar küçüldükçe artmış ve süre dokuzda bir oranda azalmıştır.



Şekil 5.9 Rastgele ifadeli test imajları ile şaşırma ifadeli imajların karşılaştırılmasının süre başarı oranı grafiği

Şekil 5.9’de görüldüğü üzere en iyi performans %20 boyutlu imajlarda çıkmıştır. Başarı oranı boyut küçüldükçe artmış ve süre beşte bir oranda azalmıştır.

6. SONUÇ VE GELECEK ÇALIŞMA

Bu tezde çeşitli yöntemler uygulanarak bir yüz tanıma sistemi tasarlanmıştır. Tasarlanan sistemde performansın (başarı oranı ve işlem süresi) artırılması için belirli işlemler gerçekleştirilmiştir. Bu işlemler sırasıyla; şablon eşleme kullanılarak yüz tanıma sisteminde başarı oranının belirlenmesi, iki farklı yöntem (gözler arasındaki mesafenin kullanılması ve performansı düşüren kısımların çıkarılması) ile elde edilen başarı oranının artırılmaya çalışılması, Çok Katmanlı Algılayıcı kullanarak farklı bir yüz tanıma sisteminin tasarlanması ve son olarak imaj boyutlarının küçültülmesi ve şablon eşleme metodunun kullanılması ile işlem süresinin kısaltılmasıdır.

Çalışmada, ilk olarak şablon eşleme metodu kullanılmıştır. Şablon eşleme yöntemi ile yüz tanıma işlemi, Minokowski sınıf metriklerinden birisi olan Şehir Blok Uzaklık formülü kullanılması ile gerçekleştirilmiştir. Bu bölümdeki sonuçlar CNL veritabanındaki ilk 1000 kişi kullanılarak elde edilmiştir. Test ifadelerinden boş ifadeli imajlarla test edilmesi sonucu, 4 ifadenin (Gülme, kızma, şaşırma ve boş) başarı oranı ortalaması % 88.200 olmuştur. Test ifadelerinden rastgele ifadeli imajlarla test edilmesi sonucu, 4 ifadenin başarı oranı ortalaması % 82.650 olmuştur.

İkinci kısımda, ilk bölümde elde edilen başarı oranının artırılması için 2 farklı yöntem kullanılmıştır. Birinci yöntem; veritabanı imajlarının üzerinde oynama yaparak, yüzün belirli bölgelerinin çıkarılması ile tekrar şablon eşleme gerçekleştirilmiştir. Test ifadelerinden boş ifadeli imajlarla test edilmesi sonucu, 4 ifadenin başarı oranı ortalaması % 91.725 olmuştur yani %3.825 oranında başarı oranı artışı gerçekleştirilmiştir. Test ifadelerinden rastgele ifadeli imajlarla test edilmesi sonucu, 4 ifadenin başarı oranı ortalaması % 85.875 olmuştur yani % 3.225 oranında başarı oranı artışı gerçekleşmiştir. İkinci yöntem olarak; imajlardaki kişilerin gözbebekleri otomatik olarak bulunup iki gözbebeği arasındaki mesafenin kullanılması ile başarı oranı artırılmaya çalışılmıştır. Fakat bu bölümde başarı oranı artışı gerçekleşmemiş, başarısız olunmuştur.

Üçüncü kısımda Çok Katmanlı Algılayıcı kullanarak sistemin başarı oranı değerlendirilmiştir. Sistem CNL veritabanındaki ilk 100 kişi kullanılarak test edilmiştir. Eğitim setinde her bir kişiye ait 8 farklı ifade olduğundan dolayı 800 adet veri girişi ile ağ eğitilmiştir. Eğitilen veriler test seti olarak kullanıldığında %100 başarı oranı sağlanmıştır. Test setindeki boş ifadeli imajlar ve rastgele ifadeli imajlar test olarak kullanıldığında ise sırasıyla %56 ve %48 oranında başarı oranları elde edilmiştir.

Son olarak, veritabanında kullanılan imajların boyutları belirli oranlarda küçültülerek karşılaştırmalar yapılmış ve boyut küçültme işleminin yüz tanıma sisteminin performansına etkisi araştırılmıştır. Bu kısımdaki simülasyonlar veritabanındaki ilk 1000 kişi kullanılarak gerçekleştirilmiştir. Bu bölümde işlemler şablon eşleme yöntemi kullanılarak gerçekleştirilmiştir. Ayrıca boyut küçültmenin performansı lineer bir şekilde etkileyip etkilemediği de incelenmiştir. Bu bölümdeki asıl amacımız, sistemin başarı oranını düşürmeksizin işlem zamanını mümkün olduğu kadar azaltmak olmuştur. Sonuç olarak yapılan çalışmada başarı oranı artırılırken işlem zamanları azaltılmıştır. En iyi başarı oranı %25 ve %20 boyutlu imajlarda görülmüş ve bu boyutlarda işlem süresi 60 saniyeye kadar düşürülmüştür.

KAYNAKLAR

- Samal, A. ve Iyengar, P. A., (1992), "Automatic recognition and analysis of human faces and face expressions", *Pattern Recognition*, 25, 65-77.
- Welch P. D., (1967), "The use of fast Fourier Transform for the Estimation of Power Spectra: A method Based on Time Averaging Over Short, Modified Periodograms", *IEEE Trans. Audio Electroacoust. Cilt No AU-15*, Sayfa 70-73.
- Anil, K. J., Lin, H., Pankanti, S., Bolle, R., (1997), "Authentication System Using Fingerprints", *Proceedings Of The IEEE*, 9 Sept. 1997, Vol. (85)
- Welsch, W. J. ve Shackleton, M. A. (1991), "Classification of facial features for recognition", *Proc. IEEE Conf. Comput Vis. Pattern Recognition CVIP-91*, 573-579.
- Turk, M. ve Petland, A.P., (1991), "Eigenfaces for recognition,", *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71-86.
- Belhumeur, P. N., Hespanha, J.P. and Kriegman, D.J., (1997), "Eigenfaces vs. Fisherfaces: recognition using class specific linear projection", *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711-720.
- Kepenkeçi, B. and Tek, F.B., Akar, G.B., (2002), "Occluded Face Recognition Based on Gabor Wavelets", *IEEE Int. Conf. on Image Processing (ICIP)*, September, Rochester, New York.
- Huang, J., Huang, X., Metaxes, D., ve Banerjee, D., (2006), "3d Tumor Shape Reconstruction from 2d Bioluminescence Images", *3rd IEEE Int. Symp.* Vol, Issue, 6-9 April, 606-609.
- Govindaraju, V., (1990), "A Computational Model For Face Location," *Proc. of the Int'l Conf. on computer Vision*, Dec. 1990, pp. 718-721, Osaka, Japan.
- Sanger, D., Miyake Y., Haneishi, H., Tsumura, N., (1997), *Journal of Imaging Science and Technology*.
- Soulié, F. F., Viennet, E., Bertrand Lamy, B., (1993), "Multi-Modular Neural Network Architectures: Applications in Optical Character and Human Face Recognition", *Journal IJPRAI* Vol. 7, No. 4, 721-755.
- Han, C., Liao, H-Y-M., Yu, G. J., ve Chen, L-H., (2000), "Fast face detection via morphology-based pre-processing", *Pattern Recognition*, Vol. 33, Issue 10, October 2000, 1701-1712.
- Chen, L-F., Liao, H-Y-M., Ko, M-T, Lin, J-C and Yu, G-J., (2000) "A new LDA-based face recognition system which can solve the small sample size problem", *Pattern Recognition*, vol. 33, pp. 1713-1726.
- Yacoub, S., Fasel, B., Luttin, J., (1999), "Fast face detection using MLP and FFT", *Proceedings of Second International Conference on Audio and Video-based Biometric Person Authentication, AVBPA*.
- Rowley, A. H., Baluja, S., ve Kanade, T., (1996) "Neural network - based face detection", *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, San Francisco, CA, pp. 203-207.
- Swets, D.L., and Weng, J., (1996), "Using discriminant eigenfeatures for image retrieval," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, August, vol. 18, no. 8, 831-836.

- Goldstein, A.J., Harmon L. D., ve A. B., (1971), "Lesk. Identification of human faces", In Proc. IEEE, Vol. 59, sayfa 748.
- Kaya, Y., ve Kobayashi, (1972), "A basic study on human face recognition", In S. Watanabe, editor, Frontiers of Pattern Recognition, sayfa 265.
- Brunelli, R., Poggio, T., (1992), "Face Recognition Through Geometrical Features", In Proceedings of ECCV '92, Margherita Ligure, pages 792-800.
- Dede, E., (2000) "Face recognition using geometric futures and template matching", Orta Doğu Teknik Üniversitesi, Kasım.
- James, W. (1890), "Psychology (Briefer Course), New York: Holt, Chapter XVI, association" 253-279.
- Hebb, D.O., (1949), "The organization of behaviour", New York: Wiley, Introduction and Chapter 4, The first stage of perception: growth of the assembly", pp. Xi-xix, 60-78.
- McCullosh, W.S. ve Pitts, W.A. (1943), "A logical calculus of the ideas immanent in nervous activity", Bulletin of Mathematics and Biophysics, pp. 115-133.
- Farely, B. G. ve Clark, W.A. (1954), "Simulation of self-organizing systems by digital computers", IEE Transactions of Professional Group of Information Theory, PGIT-4, pp. 76-84.
- Rosenblatt, F.(1958) "The perceptron: A probabilistic model for information storage and organization in the brain", Psychoanalytic Review, 65, , pp. 386-408
- Widrow, B. ve Hoff, M. E. (1960), "Adaptive switching circuits ", WEST-CON Convention, Record Part IV, 1960, pp. 96-104.
- Nilson, N. J., (1965), "Learning Machines", McGraw-Hill.
- Rummelhart, D. E., Hinton, D. E. ve Williams, R. J. (1986), "Learning representation by backpropagating errors" Nature 323(9), pp. 533-536.
- Parker, D. B. (1985), "Learning-logic", M.I.T. Cen. Computational Res. Economics Management Sci., Cambridge, MA, TR-47.
- Werbos, P. J. (1974), "Beyond regression: new tools for prediction and analysis in the behavioural sciences", PhD. dissertation, Committee on Appl. Math., Harvard Univ., Cambridge, M. A.
- Zeidler, J. R. (1990), "Performance analysis of LMS adaption filters", Proceedings of IEEE, Vol. 78, No. 12, pp. 1781-1806.
- Widrow, B., and Lehr, M.A. (1990), "30 years of adaptif neural Networks: Perceptron, Madaline and Backpropogation", proceedings of the IEEE, Vol 78, No9.
- Zurada, M. J. (1992), "Introduction to Artificial Neural Systems" West Publishing Company, Inc. New York.
- Sağıroğlu, S., Beşdok, E., Erler, M., Mühendislikte Yapay Zeka Uygulamaları-1 Ufuk Yayıncılık, Kayseri.
- Haykin, S. (1994), "Neural Networks, A Comprehensive Foundation", Macmillan College Publishing Co. Inc.

Artıklar, M. (2002), "Capacity Analysis of Voting Networks with Application to Human Face Recognition", Wayne State University College of Engineering Ph. D. Thesis 445.

Yılmaz, S. ve Artıklar, M. (2005), "On Recognizing Human Faces from Frontal Views", KSU Fen ve Mühendislik dergisi 8(1)-2005, 41-44.

Krauss P., Shure I., Little J.N., (1996), "MATLAB Signal Processing Toolbox User's Guide", The Mathworks Inc. 1996.

Xiaoyan M., Hassoun, M. H., (2003) "Combining Gabor Features: Summing vs. Voting in Human Face Recognition", Systems, Man and Cybernetics, IEEE International Conference, pp 737-743.

Talu F., Tatar Y., (2004) "Model Tabanlı Yapay Sinir Ağı Kullanılarak Geliştirilmiş Bir Kenar Çıkarma Metodu", Fırat Üni. Bil. Müh. Bölümü.

http://help.adobe.com/tr_TR/Photoshop/10.0/help.html?content=WSfd1234e1c4b69f30ea53e41001031ab64-793c.html

http://www.goruntuisleme.org/index.php?option=com_content&task=view&id=69&Itemid=2

Abdou, I. E., ve Pratt, W. K., (1979), "Quantitative design and Evaluation of enhancement/thresholding high. Edge detectors", Proc. IEEE, vol. 67, no. 5, pp. 753-763.

Costa, L. F., and Cesar, R. M., (2001), "Shape Analysis and Classification", pp. 240-244.

Pinho, J., and Almeida, L.M., (1994), "Some Results On Edge Enhancement With Neural Networks", Proc. of the 1st IEEE Int. Conf. on Image Processing, ICIP'94, Austin, TX, vol. III, pp. 893-897.

EKLER

Ek 1 Matlabta Yazılan Kodlar

Ek 1**MATLABTA YAZILAN KODLAR**

Bu bölümde Matlabta yazdığım kodların bir bölümünü sunacağım. İşlemi hızı artırılması adına yazılan Matlab kodu aşağıdaki gibidir.

```
function Mdosyaları=Mreduce(ima_j_no, l_resmi, hafiza_tipi,
icerik,reduce,metot);

sira=82;

column=115;

rprow=round((sira*reduce)/100);

rpcolumn=round((column*reduce)/100);

Mdosyaları=zeros(ima_j_no,rprow*rpcolumn);

for i=1:ima_j_no
    icerik=icerik(:,i);

    mfilename=sprintf('C:\\images\\memory\\image%d%s.raw', icerik,
hafiza_tipi);

    fpmem=fopen(mfilename,'r');

    if (fpmem<0)
        error('Files or a file could not read on this path')
        exit(0);
    end

    [memimg,countmem]=fread(fpmem,[1 inf]);

    if reduce ~= 100;
        memimg=resizeres(memimg,reduce,metot);
    else reduce == 100;
        memimg=memimg;
    end

    Mdosyaları(i, :)=memimg;

    fclose(fpmem);

end
```

```
function kucultme1=resizeres(kucultme2,reduce,metot)
```

```
row=82;

column=115;

rprow=round((row*reduce)/100);
```

```

rpcolumn=round((column*reduce)/100);
kucultme2=reshape(kucultme2,row,column);
kucultme2l=resizem(kucultme2,[rprow rpcolumn],metot);
kucultme1=reshape(kucultme2,1,rprow*rpcolumn);

```

```

function T_dosyalari=RTReduceSize(imaj_n, imaj_l, test_tipi,
icerik,reduce,metot);

T_dosyalari=zeros(imaj_n,imaj_l);
for i=1:imaj_n
    mnumarasi=icerik(:,i);

    mfilename=sprintf('C:\\images\\test\\image%d%s.raw', mnumarasi,
test_tipi);

    fpmem=fopen(mfilename,'r');

    if (fpmem<0)
        error('Files or a file could not read on this path')
        exit(0);
    end

    [testing,countmem]=fread(fpmem,[1 inf]);

    if reduce ~= 100;
        testing=resizeres(testing,reduceP,metot);
    else reduce == 100;
        testing=testing;
    end

    T_dosyalari(i, :)=testing;

    fclose(fpmem);
end

```

```

function [icerik,kazanan]=CBoneM(n_or_r,HMP,memory_type,test_type,kac_kere)
n_or_r=n;
kac_kere=1;
lengthpic=9430;
if n_or_r=='n'
    icerik=1:HMP;
else n_or_r=='r'
    icerik1=su(HMP);

```

```

    icerik=sort(icerik1);
end

matmemory=rmreader(HMP,lengthpic,memory_type,icerik);
kazanan=zeros(HMP*2,10);

for perftestpic=1:HMP
    tname=sprintf('C:\\images\\test\\image%d%s.raw', icerik(:,perftestpic),
test_type);
    fpmem=fopen(tname,'r');

    if (fpmem<0)
        error('Files or a file could not read on this path')
        exit(0);
    end

    testing=fread(fpmem,[1 inf]);

    fclose(fpmem);
    perfdistance=0;
    for imempic=1:HMP
        perfabs=abs(testing-matmemory(imempic,:));
        perfabs=sqrt(perfabs);
        perfint=sum(sum(perfabs));
        perfdistance(:, imempic)=perfint;
    end
    for i=1:10
        [mini indice]=min(perfdistance);
        win1(1,i)=indice;
        win2(1,i)=mini;
        perfdistance(:,indice)=999999999;
    end
    kazanan((perftestpic*2)-1,:)=win1(1,:);
    kazanan((perftestpic*2),:)=win2(1,:);
end
end

```

```

for t=1:2*HMP
    for k=1:1
        fark1= kazanan(t,k);
        fark2= kazanan(t,k+1);
        fark(t)=abs(fark2-fark1);

    end
end

filename1XLS=sprintf('EDoneMperformtest2-%s-%s-%d-
%d.xls',memory_type,test_type,HMP,kac_kere);
fid=fopen(filename1XLS,'w');
fprintf(fid,'DATABASE icerik NUMBER AND City Block DISTANCES \n\n');
if n_or_r =='n'
    fprintf(fid,'database icerik number selected normaly')
    for i=1:HMP*2

        fprintf(fid,'\n');
        fprintf(fid,'%3.2f\t',kazanan(i,:));
        if fark (i)>100
            if fark(i)<9000
                fprintf(fid,'%3.2f\t(ŞÜPHELİ)\t',fark(i));
                for p=1:5
                    KS(p)=999999;
                end
                for p=1:5
                    L=kazanan(i-1,p);
                    KS(p)=compare(L,i/2,memory_type,test_type);
                %Q=gozbul1(i/2);
                %W=gozbul2(L);
                KS(p)=abs(Q-W);
                %KS
                if min(KS)==KS(p)
                    TS=L;
                end
            end
        end
    end
end

```

```

        fprintf(fid,'%d',TS);
    else
        fprintf(fid,'kesin sonuç');
    end
end
end
if n_or_r =='r'

    fprintf(fid,'database icerik number selected randomly')
    for i=1:HMP*2
        fprintf(fid,'\n');
        fprintf(fid,'%3.2f\t',kazanan(i,:));
        if fark(i)>100
            if fark(i)<9000
                fprintf(fid,'%3.2f (ŞÜPHELİ)',fark(i));
            else
                fprintf(fid,'kesin sonuç');
            end
        end
    end
end
fclose(fid);
%Hafızadan imaj okutturmak için yazılan kodlar;

function RMfiles=rmreader(npicture, lpicture, mtype, index);
if mtype == 'b'
    RMfiles=zeros(npicture,lpicture);
    for i=1:npicture
        mnumber=index(:,i);
        mfilename=sprintf('C:\\images\\memory\\image%d%c.raw', mnumber, 'b');
        fpmem=fopen(mfilename,'r');

        if (fpmem<0)
            error('Files or a file could not read on this path')
            exit(0);
        end
    end
end

```

```

end

[memimg, countmem]=fread(fpmem,[1 inf]);

%RMfiles=memimg;
%RMfiles=RMfiles';
RMfiles(i, :)=memimg;
%imge=reshape(memimg,82,115);
%imge=imge';
%figure, imagesc(imge)
%colormap(gray)
fclose(fpmem);
end

end

if mtype == 's'
    RMfiles=zeros(npicture,lpicture);
    for i=1:npicture
        mnumber=index(:,i);
        mfilename=sprintf('C:\\images\\memory\\image%d%c.raw', mnumber, 's');
        fpmem=fopen(mfilename,'r');

        if (fpmem<0)
            error('Files or a file could not read on this path')
            exit(0);
        end

        [memimg, countmem]=fread(fpmem,[1 inf]);
        RMfiles(i, :)=memimg;
        %imge=reshape(memimg,82,115);
        %imge=imge';
        %figure, imagesc(imge)
        %colormap(gray)

        fclose(fpmem);

    end

end

if mtype == 'u'

```

```

RMfiles=zeros(npicture,lpicture);
for i=1:npicture
mnumber=index(:,i);
    mfilename=sprintf('C:\\images\\memory\\image%d%c.raw', mnumber, 'u');
                                fpmem=fopen(mfilename,'r');

    if (fpmem<0)
        error('Files or a file could not read on this path')
        exit(0);
    end
    [memimg,countmem]=fread(fpmem,[1 inf]);

    RMfiles(i, :)=memimg;
    fclose(fpmem);
end
end
if mtype == 'a'
    RMfiles=zeros(npicture,lpicture);
    for i=1:npicture
        mnumber=index(:,i);
        % read files from the directory and load in mfilename pointer.
        mfilename=sprintf('C:\\images\\memory\\image%d%c.raw', mnumber, 'a');
        fpmem=fopen(mfilename,'r');

        if (fpmem<0)
            error('Files or a file could not read on this path')
            exit(0);
        end

        [memimg,countmem]=fread(fpmem,[1 inf]);
        RMfiles(i, :)=memimg;
        fclose(fpmem);
    end
end
end

```

ÖZGEÇMİŞ

Doğum tarihi	20.04.1983	
Doğum yeri	Adana	
Lise	1997-2000	Abdulkadir Paksoy Kız Lisesi
Lisans	2000-2005	Fatih Üniversitesi Mühendislik Fak. Elektronik Mühendisliği Bölümü

Çalıştığı kurum(lar)

2005-2006 Dal Group
2006-Devam ediyor Pro-Tek