

YAPAY SİNİR AĞLARI İLE YÜZ TANIMA

Dokuz Eylül Üniversitesi Fen Bilimleri Enstitüsü

Yüksek Lisans Tezi

Mekatronik Mühendisliği Bölümü, Mekatronik Mühendisliği Anabilim Dalı

Ozan TAŞOVA

Haziran, 2011

İZMİR

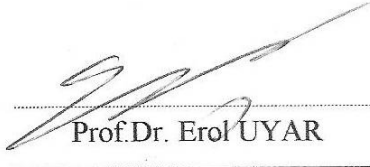
YÜKSEK LİSANS TEZİ SINAV SONUÇ FORMU

OZAN TAŞOVA tarafından DOÇ. DR. ZEKİ KIRAL yönetiminde hazırlanan “YAPAY SİNİR AĞLARI İLE YÜZ TANIMA” başlıklı tez tarafımızdan okunmuş, kapsamı ve niteliği açısından bir Yüksek Lisans tezi olarak kabul edilmiştir.



Doç.Dr. Zeki KIRAL

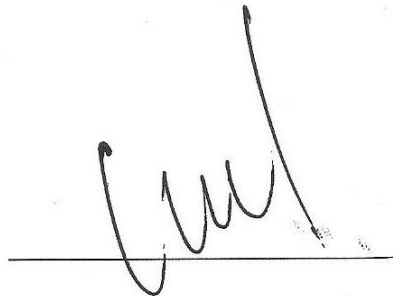
Yönetici



Jüri Üyesi



Jüri Üyesi



Prof.Dr. Mustafa SABUNCU

Müdür

Fen Bilimleri Enstitüsü

TEŞEKKÜRLER

Bu tez çalışmasının planlanmasında, araştırılmasında benden ilgi, destek ve güvenini hiçbir zaman esirgemeyen, engin bilgi ve tecrübeleri ile çalışmamı bilimsel temeller ışığında şekillendiren başta danışman hocam Doç. Dr. Zeki KIRAL olmak üzere; gerekli ekipmanları sağlayan, tecrübelerini bizle paylaşan ve fikirleri ile destek olan Prof. Dr. Erol UYAR' a sonsuz teşekkürlerimi sunarım.

Tüm eğitim ve tez çalışmalarım boyunca sabır, anlayış ve sevgileri ile her zaman yanımda olan, desteklerini hiçbir zaman esirgemeyen ve bana güvenen sevgili aileme tüm kalbimle teşekkür ederim.

Ozan TAŞOVA

YAPAY SİNİR AĞLARI İLE YÜZ TANIMA

ÖZ

Yüz tanıma, günümüzde önemini hızla arttırmakta olan bir örüntü tanıma konusudur. Yüz tanıma, insan beyninin çok zorluk çekmeden gerçekleştirdiği bir görevidir. İnsan yüzü oldukça ayırt edici özelliklere sahiptir ve insan beyni yüze ait görsel bilgileri, biometrik tanımlayıcılar olarak kullanır.

İnsan beyni karmaşık bir yapıya sahiptir ve bu nedenle analiz edilmesi zordur. Fakat beynin en temel işlevi olan öğrenme için benzetim yapılabilir. Yüz tanıma konusunda beynin temel işlevlerini benzetimle gerçekleştiren bilgisayar algoritmaları (Yapay sinir ağları) geliştirilmiştir.

Yüz tanıma problemi, girdi olarak verilen bir yüz görüntüsü ile, bilinen kişilerin yüz görüntülerini içeren bir veritabanı kullanarak, bir bireyi tanımlamak veya doğrulamak olarak tanımlanabilir. Teknolojideki gelişmeler ve yüz tanımayı kullanan uygulamalara duyulan ihtiyacın artması nedeniyle, bu alanda yapılan araştırmalar son yıllarda hızla artmıştır.

Bu tezde yapay sinir ağları kullanılarak yüz tanıma geliştirilmiştir. Yüz tanıma işleminde orijinal resmin eğim (gradient) görüntüsü kullanılmıştır. Elde edilen grafikten belirli sayıda öz nitelik değerleri üretilmiştir. Elde edilen öz nitelik değerleri yapay sinir ağında kullanılarak yüz tanıma işlemi gerçekleştirilmiştir. Yapılan denemeler geliştirilen yöntemin yüz tanıma için başarılı olduğunu göstermiştir.

Anahtar sözcükler: Yüz tanıma, örüntü tanıma ve yapay sinir ağları.

FACE RECOGNITION WITH ARTIFICIAL NEURAL NETWORK

ABSTRACT

Face recognition is a subject of pattern recognition which has rapidly increasing importance. Face recognition is a task performed by the human brain without much difficulty. The human face has a distinctive features and the human brain uses the visual information of a face as a biometric identifier.

The human brain has a complex structure and therefore difficult to analyze. But for the learning, which is the basic function of the brain, simulation can be done. Computer algorithms (Artificial Neural Networks) have been developed for face recognition that perform the basic functions of the brain by simulation.

Face recognition problem can be defined as to identify or verify a person using a database including the collection of known people with a input face pattern. With the advances in technology and increasing need for face recognition applications, researches in this area have increased rapidly in recent years.

In this thesis, a face recognition using artificial neural networks has been developed. The gradient pattern of the original image is used in the face recognition process. Some features are derived form the gradient pattern. The face recognition task is performed by using these derived features in the artificial neural network. The tests show that the developed method is succesfull for face recognition.

Keywords: Face recognition, pattern recognition and artificial neural Networks.

İÇİNDEKİLER

	Sayfa
YÜKSEK LİSANS TEZİ SINAV SONUÇ FORMU.....	ii
TEŞEKKÜRLER.....	iii
ÖZ.....	iv
ABSTRACT.....	v
BÖLÜM BİR- GİRİŞ.....	1
BÖLÜM İKİ-BİOMETRİ	3
2.1 Biometri Nedir	3
2.2 Biometrik Yöntemler.....	4
2.2.1 İris Tanımlama.....	4
2.2.2 El Geometrisi Tanıma	5
2.2.3 Ses Tanıma	6
2.2.4 Parmak İzi Tanıma	7
BÖLÜM ÜÇ-BİOMETRİK VERİLERİN GİRİLMESİ VE İŞLENMESİ	9
BÖLÜM DÖRT-YAPAY SİNİR AĞLARI.....	11
4.1 YSA'nın Tanımı ve Tarihçesi	11
4.2 YSA'nın Uygulama Alanları	14
4.2.1 Örüntü Tanıma (Pattern Recognition).....	14
4.2.2 Arıza Analizi ve Tespiti	15
4.2.3 Tıp Alanında	15
4.2.4 Otomasyon ve Kontrol	15
4.2.5 Optimizasyon İşlemlerinde.....	15

4.3 YSA'ların Genel Özellikleri.....	16
4.3.1 Doğrusal Olmama	16
4.3.2 Öğrenebilirlik.....	16
4.3.3 Genelleme.....	16
4.3.4 Uyarlanabilirlik.....	17
4.3.5 Hata Toleransı.....	17
4.3.6 Donanım ve Hız	17
4.4 Sinir Hücreleri ve Sinir Ağları	18
4.4.1 Biyolojik Sinir Hücresi.....	18
4.4.2 Yapay Sinir Ağı Hücresi	19
4.5 Aktivasyon Fonksiyonu	21
4.5.1 Doğrusal Aktivasyon Fonksiyon (Lineer Fonksiyon).....	21
4.5.2 Step Aktivasyon Fonksiyonu.....	22
4.5.3 Sigmoid Aktivasyon Fonksiyonu.....	22
4.5.4 Hiperbolik Tanjant Aktivasyon Fonksiyonu	24
4.6 Yapay Sinir Ağlarının Yapısı.....	25
4.6.1 Giriş Katmanı.....	25
4.6.2 Ara Katman (Gizli Katman)	26
4.6.3 Çıkış Katmanı	26
4.7 Yapay Sinir Ağlarının Sınıflandırılması.....	26
4.7.1 İleri Beslemeli Yapay Sinir Ağları (Feed Forward)	27
4.7.2 Geri Beslemeli Yapay Sinir Ağları	28
4.7.3 Geri Dönürlü Yapay Sinir Ağları (Recurrent).....	29
4.8 Yapay Sinir Ağlarının Eğitilmesi	30
4.9 Yapay Sinir Ağlarında Öğrenme Kuralları	31
4.9.1 Hebb Kuralı	31
4.9.2 Delta Kuralı	31
4.9.3 Kohonen Kuralı.....	32
4.9.4 Hopfield Kuralı	32

4.10 Yapay Sinir Ağlarında Öğrenme Algoritmaları	33
4.10.1 Danışmanlı Öğrenme.....	33
4.10.2 Danışmansız Öğrenme	33
4.10.3 Takviyeli Öğrenme.....	33
BÖLÜM BEŞ-YÜZ TANIMA SİSTEMİ.....	34
5.1 Kenar Bulma (Edge Detection)	35
5.2 Eğim Operatörü (Gradyent).....	36
BÖLÜM ALTI-YÜZ TANIMA UYGULAMASI	37
6.1 Renk Sistemleri Ve Renk Uzayları	37
6.1.1 RGB Renk Uzayı	37
6.1.2 Lab Renk Uzayı	38
6.2 Ten Rengine Bağlı Yüz Tespiti.....	39
BÖLÜM YEDİ- SONUÇLAR VE DEĞERLENDİRME	61
KAYNAKLAR.....	62

BÖLÜM BİR

GİRİŞ

Tezin amacına ulaşmak ve bu konuda önceden yapılmış çalışmaların incelenmesi için literatür taraması yapılmıştır.

Yapılan araştırmalar sonucu, biometrik sistemler ve yüz tanıma ile ilgili bazı çalışmalara ulaşılmıştır.

(a) Tüzün ve Akan (bt) ortak çalışmasında, yüzlerin oluşturduğu uzayı dikleştirilip, dik bir alt uzay elde edilmesi yöntemi üzerinde çalışmışlardır. Bu çalışmada verilen bir yüzün, dik kümedeki elemanların doğrusal birleşimi ile elde edilen kestirimine olan uzaklığını en küçülterek uygun bir tanıma gerçekleştirilmektedir.

(b) Arı ve Akarun (2009) ortak çalışmasında, işaret dilinde yaygın kullanılan yüz ifadelerinin tanınması için, aktif şekil modeli tabanlı bir yüz öznitelik izleyicisi ve saklı markov modeli tabanlı bir ifade sınıflandırıcı içeren bir sistem geliştirmişlerdir. Yüz izleyici, değişik pozlardaki yüzleri hızlı ve etkin bir şekilde tanıyabilmek için çok-çözünürlüklü ve çok-pozlu olarak çalışmaktadır. İzleyici ve sınıflandırıcı başarımının ölçümü için ayrıntılı testler yapılmış ve sistemin yüksek başarımla güvenilir olarak çalıştığı görülmüştür.

(c) Gökberk, İrfanoğlu, Doğu, Akarun ve Alpaydın (2003) ortak çalışmasında, yüz tanımada yerel öznitelik tabanlı yüz tanıma sistemlerinde, tanıma başarımının yükseltilmesi için, resimdeki belirgin noktaların bulunması gerektiğini söylemişlerdir. Yüzdeki en iyi öznitelik yerlerini ve bu özniteliklerin birleşimini bulmak için bir dizi öznitelik seçim algoritması kullanmışlardır.

(d) Arı, Alsaran ve Akarun (2011) ortak çalışmasında, duygu tanıma problemi irdelenmekte, video üstünde gerçek-zamanlı olarak çalışabilen ve şaşkınlık, kızgınlık, mutluluk, üzülmeye, korku, iğrenme ve yalın ifadeyi tanıyabilen yeni bir yöntem önerilmektedir. Önerilen yöntemde çok-çözünürlüklü aktif şekil modeline dayalı takipçi ile nirengi noktaları takip edilmekte ve ardından nokta konumlarından ve yüzün belli bölgelerindeki değişimlerden çıkarılan üst seviye öznitelik vektörlerinin duygusal ifadelere ait öznitelik vektörlerine olan uzaklıkları kullanılarak sınıflandırma yapılmaktadır.

(e) Çelik ve Oral (2003) ortak çalışmasında, bir plaka tanıma sistemi üzerinde durulmuştur. Çalışmada, bu işlem için geriye yayılım algoritması kullanan yapay sinir ağı (YSA) kullanılmıştır. Tanıma işlemi iki aşamada gerçekleşmektedir; plaka bölgesi içerisinde karakter yer tespiti ve bu karakterlerin tanınması. Ardından belirlenen karakter bölgesi, plakalarda kullanılan 22 harf ve 10 rakamı tanımaya eğitilmiş YSA'ya beslenmiş ve plaka tanıma gerçekleştirilmiştir.

(f) Baltacı (2011) yaptığı çalışmada, Yapay sinir ağları ve parmak izi analizi ile kimlik tayini çalışmasında parmak izlerinden çıkardığı öznitelik değerlerini yapay sinir ağına besleyerek tanıma gerçekleştirmiştir. Bu çalışma sonucunda kullanılan algoritmaların doğru sonuçlar verdiği tespit edilmiştir.

Yukarıda bahsedilen çalışmaların başarı yüzdesi değerleri Tablo 1.1'de gösterildiği gibidir.

Tablo 1.1 Başarı yüzdesi aralığı

Çalışma	Başarı Yüzdesi Aralığı	Çalışma	Başarı Yüzdesi Aralığı
a	% 95 - % 97	d	% 61,9 - % 76,19
b	% 53,57 - % 84,82	e	% 80,7 - % 86,6
c	% 82,13 - % 87,9	f	% 90 - % 93

BÖLÜM İKİ

BIOMETRİ

2.1 Biometrik Nedir

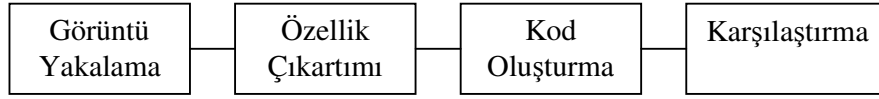
Biometrik kelimesi “bio”(yaşam) ve “metron”(ölçüm) kelimelerinden türemiş olup biyolojik veriyi ölçme ve istatistiksel olarak analiz etme bilimidir. Biometrik sistemler, bireylerin fiziksel ve davranışsal özelliklerini tanıyarak kimlik saptamak için geliştirilmiş bilgisayar kontrollü sistemler olarak da tanımlanabilir.

Biometrik sistemlerin tanımlamada kullanılmasının temel nedeni insanın biyolojisinin benzersiz olmasıdır. Her insana özgü olan bu biyolojik özelliklerin bir benzerinin olması mümkün değilken bir başkası tarafından kullanılması ve taklit edilmeleri de çok zordur. Bu yüzden biometrik tanıma sistemleri güvenlik gereksinimi duyulan alanlarda en etkili çözümdür.

Biometrik sistemler temelde, kişinin sadece kendisinin sahip olduğu, değiştiremediği ve diğerlerinden ayırt edici olan, fiziksel veya davranışsal özelliklerinin tanınması prensibi ile çalışmaktadır.

Biometrik tanıma sistemlerini kısaca özetlersek, Biometrik sistemler tanıma ve doğrulamadan oluşan iki aşamalı sistemlerdir. Tanıma kısmında kayıtlar toplanır ve bu kayıtlar sadece o kişiye özgü tek ve benzersiz bir koda çevrilerek veritabanında saklanır. Doğrulama kısmında ise toplanmış olan bu kayıtlar ile ilgili kişi anında karşılaştırılır ve sonuca varılır.

Şekil 2.1’de biometrik tanıma sistemlerinin blok diyagramı gösterilmektedir.



Şekil 2.1 Biometrik tanıma sistemlerinin blok diyagramı

Biometrik tanımlama sistemlerinde en yaygın kullanılan yöntemler aşağıda sıralanmıştır.

- Yüz Tanıma
- İris Tanıma
- Parmak İzi Tanıma
- El Geometrisi Tanıma
- Ses Tanıma

2.2 Biometrik Yöntemler

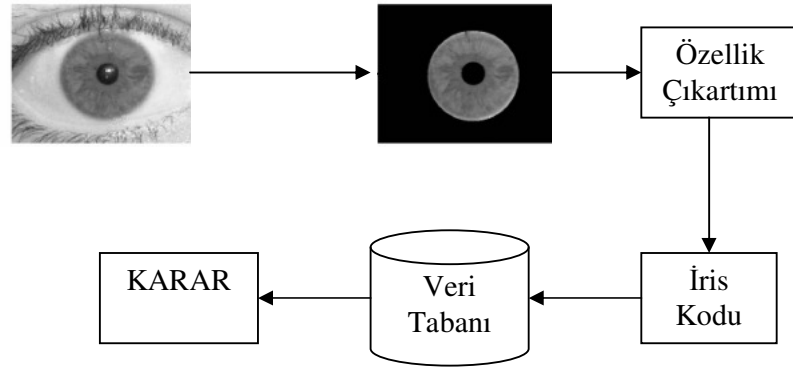
2.2.1 İris Tanımlama

Yaklaşık 30 senedir kullanılan iris tanıma sistemlerinin çıkış noktası; kişinin sahip olduğu iris şeklinin ömrü boyunca değişmemesi, diğer biometrik sistemlere göre gözüün daha az deforme olması ve dış etkenlerden daha az zarar göreceği bir yapıya sahip olmasıdır.

Bazı özelliklerinden dolayı iris biyometrisinin, güvenilirlik açısından ayrı bir konuma getirmektedir. Bunlar; her iris eşsizdir. Herhangi iki kişide aynı iris kodunun olma olasılığı $1/10^{78}$ 'dir. Sistemin hatalı kabul olma olasılığı $1/10^{42}$ 'dir. İkizlerde bile farklı olan son derece karmaşık bir desen yapısı bulunmaktadır. Desenler doğumdan sonraki 6 ay ve 2 yaş arasında sabitleşir ve hayat boyu değişmeden kalır (Sarı, bt). Uykusuzluk, gözyaşı ve hastalıklar iris yapısını etkilemekle beraber diğer yöntemlerdeki kadar bariz bir etkilenme söz konusu değildir.

Genel olarak parmak izi tanımaya benzetilen bu sistemin, parmak izine göre en önemli avantajı, parmak izi kullanılan biyometrik sistemlerde 60 veya 70 karşılaştırma noktası bulunurken, iris taramada karşılaştırma için yaklaşık 200 referans noktası kullanılmasıdır (Şamlı, 2010).

Üç aşamadan oluşan basit bir iris tanıma sistemi Şekil 2.2’de gösterilmektedir. İlk aşamada orijinal resimden iris bölgesinin bölütlenmesi, ikinci kısımda bölütlenen bölgeden özellik çıkartımı algoritmasının uygulanması, en son olarak da elde edilen özellik vektörü nünün elimizdeki veritabanı ile karşılaştırılması ve karar verilmesi yapılır.



Şekil 2.2 Iris tanıma sistemi

2.2.2 El Geometrisi Tanıma

El geometrisi aynı zamanda el taraması (Damar Tanıma) olarak da bilinir. Bu sistemde el üç boyutlu olarak taranarak elin ve parmakların geometrik yapısı analiz edilir. Tarama sırasında parmakların uzunluğu, birleşme noktaları arasındaki uzaklıklar, parmaklardaki oynak yerlerinin geometrisi gibi noktalara dikkat edilir. Bazı sistemlerde yalnızca üç parmak (baş, orta ve işaret parmağı) taranır. Doksan dan fazla ölçüm yapılır. El geometrisi ile çalışan biyometrik sistemler; parmak izi, yüz ve iris tanıma sistemleri kadar hassas ve güvenilir değildir (Mesleki Eğitim ve Öğretim Sisteminin Güçlendirilmesi Projesi [MEGEP], 2007).

Kullanıcı sayısının fazla olmadığı, sisteme çok fazla erişimin olmadığı ortamlarda ve fazla kullanım disiplini gerektirmeyen ortamlarda tercih edilebilir (Saday ve Akhan, bt).

El geometrisi diğer biyometrik yöntemler kadar olmasa da; doğruluk oranı yüksek bir yöntemdir. Ancak bazı dezavantajları mevcuttur. Bunlar; sistemin büyük ve ağır okuma cihazlarına gereksinim duyması, maliyet açısından diğerlerine göre daha pahalı olması, elle temas gerektirdiği için bazı hastalıkların (SARS) kişiden kişiye bulaşmasına neden olabileceğidir. Bunların dışında elde bulunan yüzük gibi aksesuarlar, yara bandı gibi maddeler sebebiyle ya da yaralanma ve parmakların kaybedilmesi, gut veya kireçlenme gibi bir takım hastalıklar nedeniyle elin tanınması zorlaşır. Çocuklarda el ve ayakların çok hızlı büyüdüğü hastalıklara sahip olan kişilerde ise bu sistem kullanılamamaktadır (Şamlı ve Yüksel, 2009).

Diğer yöntemlerle karşılaştırıldığında aralarındaki hız farkı nedeniyle çok büyük dezavantaja sahiptir. Kişileri sistem kullanımına hazırlamak yaklaşık 15 saniye almaktadır. Diğer sistemlerde örneğin yüz taramasında 1sn retina ve iris 4-5 sn arasında yenileme (refresh) süresi almaktadır (Özpınar, bt).

2.2.3 Ses Tanıma

İnsanların ses karakteristiği bir bütün olarak; ses sistemi, ağız, burun çukurları ve diğer insan konuşma mekanizmalarından meydana gelir. Bu karakteristiğe, biyometrik yöntem olarak ses izi ismi verilir. Her insanda tektir ve bu özellikler diğer insanlardan farklıdır (Bilgin, 2008). Bu sistem metne dayalı ve metne dayalı olmayan tanıma olarak ikiye ayrılır. Metne dayalı ses izi tanımada her kez önceden belirlenmiş bir metni okuyarak kaydeder. Metne dayalı olmayan ses izi tanımada ise önceden belirlenmiş bir metin yerine kişinin herhangi bir ortamdaki sesi kaydedilerek yapılır. Daha sonraki tanımlama işlemlerinde bu sözcük veya kelime grupları kullanılır.

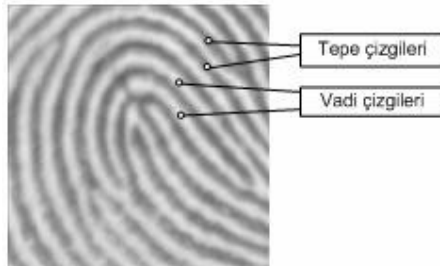
Ses karakteristiği belirlenirken sesin ses spektrogramına bakılır. Spektrogram basitçe dikey ekseninde insan sesinin frekansı, yatay ekseninde ise zaman bulunan bir grafikdir.

Bu sistemde onaylama ve tanımlama işlemleri için, el geometrisi okuyucusu gibi aletlere gerek yoktur. Tanımlama için cep telefonundan konuşma bile yeterli olabilmektedir (Egm Güvenlik, 2010). Ancak kaydedilen ses bilgisinin çok fazla yer kaplaması, sesin hastalık veya psikolojik durumlardan dolayı değişmesi, arka plandaki gürültüler ve hem kayıt sırasında hem de sisteme giriş sırasında okunan metinde yanlış sözcük kullanılması gibi dezavantajlar sahiptir. Bu yüzden diğer yöntemlere göre daha güvenilirsizdir.

Ses izi tanımanın en büyük sıkıntısı, kapasitesi yüksek olan sistemlerde doğruluk oranının düşmesidir. Özellikle, çok fazla tanımlı insan olması durumunda ses izlerinin benzer olması nedeniyle yanlış eşleştirmeler söz konusu olabilmektedir.

2.2.4 Parmak İzi Tanıma

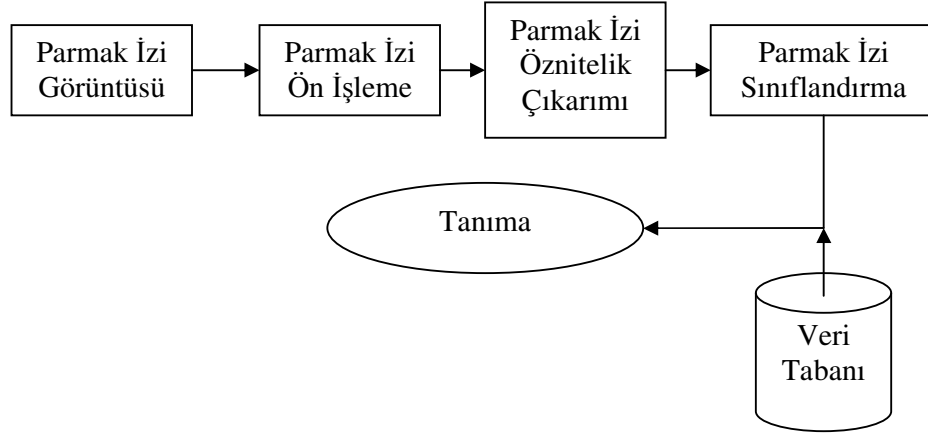
Parmak izi Şekil 2.3’de gösterildiği gibi parmak ucu derisindeki göz ile görülebilen çıkıntılardan (tepe) meydana gelen hatlar ve vadilerin oluşturduğu desenler topluluğudur. Parmak izinin yapısı, insanın doğumundan önce cenin periyodunda ortaya çıkan ve yaşlanmayla değişmeyen ayırt edici özneliktir (Moenssens, 1971).



Şekil 2.3 Örnek parmak izi

Yüz yıllardır parmak izi kişisel tanıma yöntemi olarak kullanılan en yaygın yöntemdir. Özellikle biometrik yöntem denildiği zaman akla ilk olarak gelen yöntemlerden biri parmak izidir. İlk kullanılmaya başlandığı yıllardan bu yana gerek yazılım gerekse donanım alanında parmak izi sistemlerinde önemli ilerlemeler kaydedilmiştir (Sönmez, Özbek ve Özbek, 2007).

Bir otomatik parmak izi tanıma sisteminde (OPTS) parmak izinde bulunan öznitelik noktalarının ve bunlara ait parametrelerin karşılaştırılması esasına dayanır (Sağıroğlu ve Özkaya, 2006). Şekil 2.4’de parmak izi tanıma sisteminin akış diyagramı gösterilmiştir.



Şekil 2.4 Parmak izi tanıma sisteminin akış diyagramı

Bir insana ait her parmak izi birbirinden farklı olduğu gibi her insanın parmak izi de birbirinden farklıdır. Tek yumurta ikizlerinin DNA yapıları bile aynı olmasına karşın parmak izi görüntüleri belirli farklılıklar içerir.

Bu sistemlerin en önemli dezavantajı, parmak izinin taklit edilmesi durumunda sistemin yanılabilesidir. Diğer bir dezavantajı bazı kişilerin pek çok sebepten ötürü (organ eksikliği, yanma, deri hastalıkları...) parmak izlerinin bulunmamasıdır.

BÖLÜM ÜÇ

BİOMETRİK VERİLERİN GİRİLMESİ VE İŞLENMESİ

Biometrik verilerin girilmesi ve işlenmesine örnek olarak bir resmi ele alalım. Resim ilk defa alındığında üzerinde karşılaştırma, arama, eşleme ve veri tabanına kaydetme gibi işlemleri yapmadan önce belirli işlemlere tabii tutulması gerekmektedir. Bu işlemlere ön işleme adı verilir. Resim üzerinde işlem yapabilmek için fiziksel ortamdaki resmin manyetik ortama aktarılarak sayısallaştırılması gerekir.

Resim üzerindeki her renk; Kırmızı (Red), Yeşil (Green) ve Mavi (Blue) ana renklerinin belirli oranlarda birleştirilmesi ile elde edilmektedir.

Bir resmi sayısallaştırmak demek her renge karşılık gelen sayıyı bulmak demektir. Eğer bir resim içerisinde kırmızı 15702248 renk kodu ile gösterilmişse, tüm resim içerisinde kırmızı aynı renk kodu ile gösterilir. Kırmızı renkteki en ufak bir değişiklik bu sayıyı etkiler. Sayısallaştırma işlemi; ekran piksellerine karşılık gelen renk kodunun herhangi bir program aracılığı ile alınarak bir dosyaya V(i,j) formatında yazdırılması demektir.

Ön işleme yapılmamış sayısallaşmış durumundaki bir resimde dikkat edilirse iç içe girmiş bir sürü sayı olduğu görülür. Bu haliyle resim üzerinde işlem yapmak pek mümkün olmamaktadır. Resmi üzerinde işlem yapılabilir duruma getirmemiz gerekmektedir. Öncelikle bu resim üzerinde bulunan değişik renk kodlarını ortadan kaldırarak resmi anlaşılabilir kodlara dönüştürmek gerekmektedir. Bu işleme resmi sadece siyah ve beyaz renklerden oluşan bir görüntü haline getirmek de diyebiliriz.

Resim üzerinde bulunan renkleri siyah - beyaza dönüştürmek için değişik yöntemler kullanılabilir. Bu yöntemlerden birisi 1993 yılında Mehtre'nin yapmış olduğu; V(i,j) dizisi içerisinde bulunan her bir elemanın sabit bir sayıya göre karşılaştırılması ve karşılaştırma işlemi sonucunda sabit sayıdan büyük olanlara 0, sabit sayıdan küçük olanlara 1 değerinin verilmesi yöntemidir.

Bu işlem yapıldığında $V(i,j)$ dizisiyle ifade edilen örnek bir parmak izi Şekil 3.1'deki gibi görülecektir.



Şekil 3.1 Örnek parmak izi

Parmak izi üzerinde ayrıntılı arama yapabilmek için siyah-beyaza dönüştürülmüş parmak izi üzerinde inceltme işlemi yapılmalıdır.

BÖLÜM DÖRT

YAPAY SİNİR AĞLARI

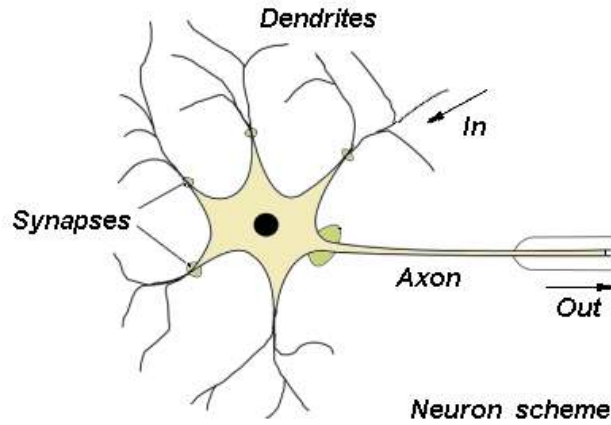
4.1 YSA'nın Tanımı ve Tarihçesi

YSA, beynin bir işlevini yerine getirme yöntemini modellemek için tasarlanan bir sistem olarak tanımlanabilir. YSA, yapay sinir hücrelerinin birbirleri ile çeşitli şekillerde bağlanmasından oluşur ve genellikle katmanlar şeklinde düzenlenir. Bu tanıma yakın bir tanımda yapay sinir ağı yazınında çok tanınan Teuvo KOHONEN'e ait bir tanımdır :” Yapay sinir ağları paralel olarak bağlantılı ve çok sayıdaki basit elemanın, gerçek dünyanın nesneleriyle biyolojik sinir sisteminin benzeri yolla etkileşim kuran hiyerarşik bir organizasyonudur.”

Beynin bilgi işleme yöntemine uygun olarak YSA, bir öğrenme sürecinden sonra bilgiyi toplama, hücreler arasındaki bağlantı ağırlıkları ile bu bilgiyi saklama ve genelleme yeteneğine sahip paralel dağılmış bir işlemcidir.

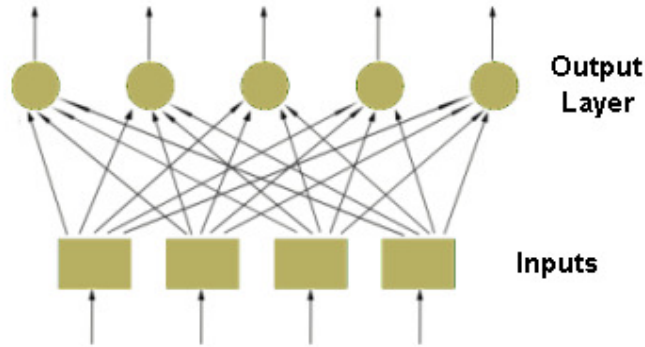
Yapay sinir ağlarının dayandığı ilk hesaplama modelinin temelleri 1940'ların başında araştırmalarına başlayan W.S. McCulloch ve W.A. Pitts'in, 1943 yılında yayınladıkları bir makaleyle atılmıştır. Mc Culloch ve Pitts, bir biyolojik nöronun temel fonksiyonlarının basit bir eşik cihazı olarak modellenebileceğini göstermişlerdir.

1949'da Donald Hebb ise “The Organization of Behaviour” adlı kitabında hücresel seviyede beyinin öğrenme mekanizmasından bahsetmiştir (Ekşi, 2010). Bu öğrenme kuralına göre; bir nöronun dentrit yoluyla gelen ve bir akson yoluyla alınan giriş onun bir darbe üretmesine sebep olur. Sonraki aksonal girişlerin darbe üretmesi olasılığı artar. Böylelikle yapılan davranışın sonucu ortaya çıkar. Şekil 4.1 de biyolojik nöron/sinir hücresinin şematik yapısı gösterilmiştir.



Şekil 4.1 Biyolojik nöron/sinir hücresinin şematik yapısı

Hızlı hesaplama yönelik ilk YSA çalışmaları 1950’li yıllarda başlamıştır. 1950’lerde Frank Rosenblatt tarafından basit nöron modellerine dayalı bir hesaplama modeli önerilmiş (Rosenblatt, 1959) ve ardından perceptron diye bilinen tek katmanlı ilk YSA modeli ortaya çıkmıştır (Rosenblatt, 1962). Şekil 4.2 gösterildiği gibidir.



Şekil 4.2 Tek katmanlı YSA modeli

Daha sonra 1954 yılında B.G. Farley ve W.A. Clark tarafından bir ağ içerisinde uyarılara tepki veren, uyarılara adapte olabilen model oluşturulmuştur.

1960'lı yıllarda Widrow ve Hoff, bu basit nöron modellerini kullanarak öğrenebilen ilk adaptif sistemler üzerinde çalışmış ve delta kuralı diye bilinen; gerçek çıkış ile istenen çıkış arasındaki farka eşit bir hata terimi kullanarak ağ ağırlıklarının değiştirildiği bir öğrenme kuralını ortaya koymuşlardır (Ekşi, 2010).

1982'de J.J. Hopfield tarafından yayınlanan "Neural Networks and Physical Systems" adlı çalışma ile çağdaş YSA devri başlamıştır (Hopfield, 1982). Bu çalışmada Hopfield, nöronların karşılıklı etkileşimlerine dayanan bir nöral hesaplama modeli önermiştir. Bu model, bir enerji fonksiyonunu, alabileceği en az değere indiren 1. mertebe lineer olmayan diferansiyel denklemlerden oluşmuştur. Hopfield; ağ seviyesinde, tek tek nöron seviyesinde var olmayan hesaplama kapasitesinin bulunduğunu öne sürmüştür. Bu tür YSA ya, "Hopfield Ağı" denilmektedir. (Ekşi, 2010).

1986'da Grossberg, ART yani Uyarlanabilir Rezonans Teorisi (Adaptive Resonance Theory) adında bir YSA yapısını geliştirmiştir. O sıralarda Kohonen'de "kendi kendini düzenleyen nitelik haritası"nı (self-organizing maps) geliştirmiştir (Kohonen, 1984). 1986'da Rumelhart ve arkadaşları paralel dağılımlı işleme adlı kitaplarında, ileri beslemeli (feed-forward) ağlarda yeni öğrenme modeli olan hatanın geriye yayılma algoritmasını (backpropagation algorithm) geliştirmişlerdir (Rumelhart, McClelland, ve PDP Research Group, 1986).

Bugün endüstride birçok YSA uygulamasında bu öğrenme yöntemi ile bunun değişik varyasyonları kullanılmaktadır. Geriye yayılma algoritması, kullanımı çok yaygın olan ve öğrenilmesi kolay bir ağıdır. İç içe girmiş konveks yapıdaki kümelerin sınıflandırılması rahatlıkla yapılmaktadır.

4.2 YSA'nın Uygulama Alanları

Yapay sinir ağları günümüz teknolojisinde çok farklı alanlarda karşımıza yeni yeni çıkmaktadır. Özellikle son 20-25 yılda hızlı bir şekilde gelişen yapay sinir ağları birçok araştırma ve çalışmadan sonra her geçen gün değişik uygulamalarda karşımıza çıkmaktadır.

YSA' ları çok farklı alanlara uygulanabildiğinden bütün uygulama alanlarını burada sıralamak zor olmakla birlikte genel bir sınıflandırma ile YSA' nın uygulama alanları aşağıdaki gibi toplanabilir.

4.2.1 Örüntü Tanıma (*Pattern Recognition*)

- *Hareket Tespiti, Hedef Tespiti, Yüz Tanıma*

Özellikle güvenliğin önemli olduğu yerlerde, askeri uygulamalarda akan görüntülerin içerisinde belirli nesnelerin ya da kişilerin tespitinde, silahların otomasyonu ve hedef izleme de kullanılır.

- *Ses tanıma*

Sesli yanıt sistemlerinde, asansörlerde, otomatik kapılarda ve hatta cep telefonların da bile kullanılmaktadır (Bolat, 2005).

- *Karakter, imza, parmak izi tanıma*

El yazısı tanıma, OCR (Optical Character Recognition) yazılımlarında, parmak izinden ve yüzünden şahıs tanıma (Rowley, Baluja ve Kanade, 1998) imza analizlerinde kullanılır.

- *İris tanıma*
- *Damar tanıma*
- *Veri iletimi, Robotik sistemlerde, Kalite kontrolünde*

Robotların öğrenmesinde, el kol koordinasyonunda kullanılır. Üretim sistemlerinin programlanmasında, ürün analizinde, yönetim analizinde... v.s. kullanım alanları mevcuttur.

4.2.2 Arıza Analizi ve Tespiti

Bir sistemde, cihazın ya da elemanın düzenli (doğru) çalışma şeklini öğrenen bir YSA yardımıyla bu sistemlerde meydana gelebilecek arızaların tanımlanma olanağı mevcuttur. Bu amaçla YSA; elektrik makinelerinin, uçakların ya da bileşenlerinin, entegre devrelerin v.s. arıza analizinde kullanılır.

4.2.3 Tıp Alanında

EEG ve EKG gibi tıbbi sinyallerin analizi, kanserli hücrelerin analizi, protez tasarımı v.s gibi konularda uygulanmaktadır.

4.2.4 Otomasyon ve Kontrol

Uçaklarda otomatik pilot sistemi otomasyonu, robot sistemlerin kontrolü, elektrikli sürücü sistemlerin kontrolü v.s. gibi alanlarda uygulanmaktadır.

4.2.5 Optimizasyon işlemlerinde

Birçok ticari ve bilimsel konularda incelenen olayların verilen kısıtlar altında hedefin maksimize ya da minimize edilmesi optimizasyon olarak bilinir. Optimizasyon için önceki çalışmalarda klasik birçok yöntem geliştirilmiş olmasına karşılık bunun YSA modellemesi ile yapılması en azından sınırlayıcı matematiksel kabullerin bulunmaması açısından yararlıdır (Şen, 2004).

4.3 YSA'ların Genel Özellikleri

YSA' nın hesaplama ve bilgi işleme gücünü, paralel dağılmış yapısından, öğrenbilme ve genelleme yeteneğinden aldığı söylenebilir. Genelleme; eğitim ya da öğrenme sürecinde karşılaşılmayan girişler için de YSA' nın uygun tepkileri üretmesi olarak tanımlanır. Bu üstün özellikleri, YSA' nın karmaşık problemleri çözebilme yeteneğini gösterir. Günümüzde birçok bilim alanında YSA, aşağıdaki özellikleri nedeniyle etkin olmuş ve uygulama yeri bulmuştur.

4.3.1 Doğrusal Olmama

YSA'larının en önemli özelliklerinden birisidir. YSA' nın temel proses elemanı olan hücre doğrusal değildir. Dolayısıyla hücrelerin birleşmesinden meydana gelen YSA da doğrusal değildir ve bu özellik bütün ağa yayılmış durumdadır. Bu özelliği ile doğrusal olmayan karmaşık problemlerin çözümünde en önemli araç olmuştur.

4.3.2 Öğrenebilirlik

YSA' nın arzu edilen davranışı gösterebilmesi için amaca uygun olarak ayarlanması gerekir. Bu, hücreler arasında doğru bağlantıların yapılması ve bağlantıların uygun ağırlıklara sahip olması gerektiğini ifade eder. YSA' nın karmaşık yapısı nedeniyle bağlantılar ve ağırlıklar önceden ayarlı olarak verilemez ya da tasarlanamaz. Bu nedenle YSA, istenen davranışı gösterecek şekilde ilgilendiği problemten aldığı eğitim örneklerini kullanarak problemi öğrenmelidir.

4.3.3 Genelleme

YSA, ilgilendiği problemi öğrendikten sonra eğitim sırasında karşılaşmadığı test örnekleri için de arzu edilen tepkiyi üretebilir. Örneğin, karakter tanıma amacıyla eğitilmiş bir YSA, bozuk karakter girişlerinde de doğru karakterleri verebilir ya da bir sistemin eğitilmiş YSA modeli, eğitim sürecinde verilmeyen giriş sinyalleri için de sistemle aynı davranışı gösterebilir.

4.3.4 Uyarlanabilirlik

YSA, ilgilendiği problemdeki değişikliklere göre işlem elemanlarının 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 ve denetim gibi alanlarda etkin olarak kullanılır.

4.3.5 Hata Toleransı

YSA, çok sayıda paralel dağılmış bir yapıya sahiptir ve ağına sahip olduğu bilgi, ağdaki bütün bağlantıların üzerine dağılmış durumdadır. Bu nedenle, eğitilmiş bir YSA'nın bazı bağlantılarının hatta bazı hücrelerinin etkisiz hale gelmesi, ağına doğru bilgi üretmesini etkilemez. Bu nedenle, geleneksel yöntemlere göre hatayı tolere etme yetenekleri son derece yüksektir.

4.3.6 Donanım ve Hız

Yapay sinir ağları paralel yapısı nedeniyle hızlı bir şekilde çalışıp problem çözme yeteneğine sahiptir.

4.4 Sinir Hücreleri ve Sinir Ağları

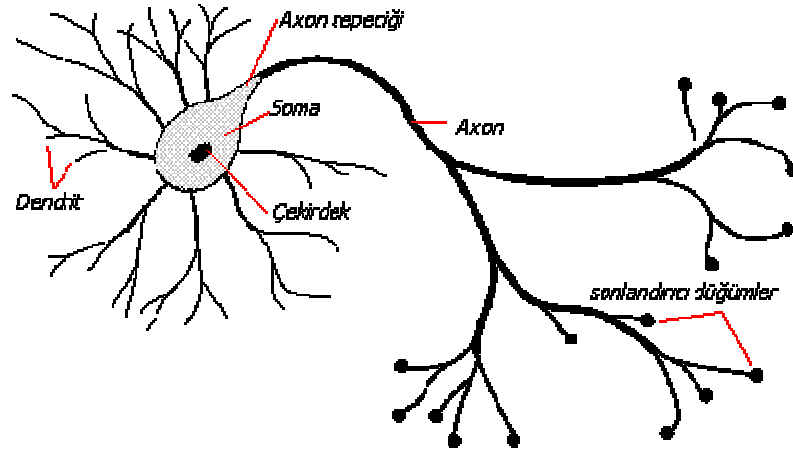
Yapay Sinir Ağları (YSA) isminden de anlaşılabilceği gibi beyni oluşturan sinir hücrelerini (nöron) matematiksel olarak taklit ederek akıllı bir sistem oluşturmaya çalışan bir yapay zekâ yöntemidir. YSA başka bir deyişle biyolojik sinir ağlarını taklit eden bilgisayar programlarıdır (Tolon ve Tosunoğlu, bt).

İnsan beyninin çalışma prensibini taklit ederek çalışan bu sistemler, her ne kadar bilgisayar teknolojisi işlem hızı nano saniyeler mertebesine inmiş olsa da, bırakalım insan beynini, ilkel bir canlı beyninin fonksiyonları dahi baz alındığında, böyle bir organizmanın yanında çok ilkel kalmaktadır. Nano saniyeler bazındaki işlem hızları ile YSA'lar, mili saniyeler mertebesindeki işlem hızları ile işlem yapan insan beyninin işlevselliğinin henüz çok uzağındadır.

Burada kısa bir hatırlatma yapmak gerekirse; insan beyinde yaklaşık 10^{11} sinir hücresinin var olduğundan bahsedilmekle birlikte, bu sayının bilgisayar ortamında modellenmesi şu an için mümkün görünmemektedir. Fakat karar hızı açısından insan beyni ile henüz yarışmasalar bile, YSA'lar yapısallıkları ve hassas eşleştirmelerin başarı ile gerçekleştirebilmeleri ile gün geçtikçe daha fazla uygulama alanı bulmaktadır.

4.4.1 Biyolojik Sinir Hücresi

Biyolojik sinir ağının temel inşa bloğu olan basit bir sinir hücresi nöron olarak adlandırılır. Şekil 4.3'de şematik diyagramı gösterilen tipik sinir hücresi üç ana bölümden oluşur. Soma olarak adlandırılan hücre gövdesi, akson ve dentritler. Dentritler üzerinden girişler alınır, soma tarafından girişler işlenir. Nörondaki sinyalleri taşıyan uzun bir sinirsel bağlantı halindeki akson ise, işlenen girişleri çıkışa aktarır. Akson dendrit bağlantısı ise sinaps olarak adlandırılır. Sinaps nöronlar arasında elektrokimyasal bağlantıyı sağlamaktadır.



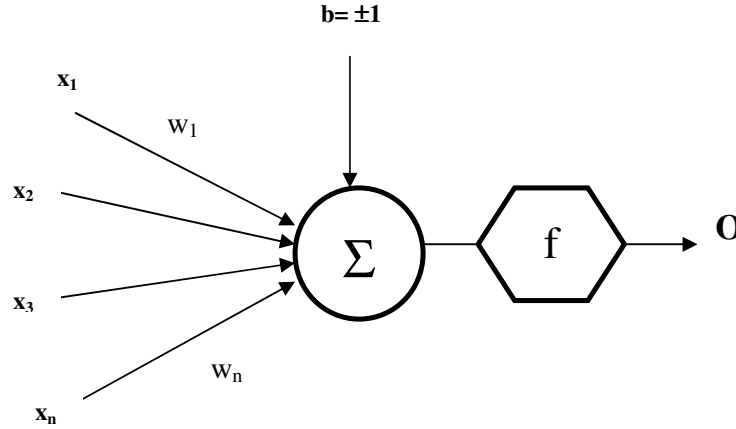
Şekil 4.3 Tipik sinir hücresi

Bir insanın beyin korteksinde yaklaşık 10 milyar nöron ve yaklaşık 60 trilyon sinaps veya bağlantının bulunduğu tahmin edilmektedir. Özellikle beyin enerjik verimliliği, her saniyede her bir işlem için yaklaşık 10^{-16} joule'dur, bu değer bugünün en iyi bilgisayarlarında yaklaşık 10^{-6} joule'dur. Beyin son derece kompleks, doğrusal olmayan ve paralel bir bilgisayardır. Beynin nöronları organize etme ve kesin hesaplamaları gerçekleştirmesi (örüntü tanıma, algılama gibi) bugünün en hızlı sayısal bilgisayarlarından daha hızlıdır (Subaş, 2010).

4.4.2 Yapay Sinir Ağı Hücresi

Temel bir yapay sinir ağı hücresi biyolojik sinir hücresine göre çok daha basit bir yapıya sahiptir. En basit yapay sinir hücresi Şekil'4.4 de görüleceği üzere dış ortamdan ya da diğer nöronlardan alınan veriler yani girişler, ağırlıklar, birleştirme fonksiyonu, aktivasyon fonksiyonu ve çıkış olmak üzere 5 ana bileşenden oluşmaktadır. Dış ortamdan alınan veriler ağırlıklar aracılığıyla nörona bağlanır ve bu ağırlıklar ilgili girişin etkisini belirler. Toplam fonksiyonu ise net girişi hesaplar. Net giriş; dış ortamdan alınan verilerle ve bu verilerin ağırlıkların çarpımının bir sonucudur. Aktivasyon fonksiyonu işlem süresince net çıkışı hesaplar ve bu işlem aynı zamanda nöron çıkışını verir.

Genelde aktivasyon fonksiyonu doğrusal olmayan (nonlinear) bir fonksiyondur. Şekil 4.4’de görülen b bir sabittir, bias veya aktivasyon fonksiyonunun eşik değeri olarak adlandırılır.



Şekil 4.4 Temel yapay sinir ağı hücresi.

Girdiler ($x_1, x_2, \dots, x_n$), diğer hücrelerden ya da dış ortamlardan hücreye giren bilgilerdir. Bunlar ağı öğrenmesi istenen örnekler tarafından belirlenir. Ağırlıklar ($w_1, w_2, \dots, w_n$), girdi kümesi veya kendinden önceki bir tabakadaki başka bir işlem elemanının bu işlem elemanı üzerindeki etkisini ifade eden değerlerdir. Her bir girdi, o girdiyi işlem elemanına bağlayan ağırlık değeriyle çarpılarak, toplam fonksiyonu aracılığıyla birleştirilir. Toplam fonksiyonu Denklem 4.1’de verildiği şekilde hesaplanır.

$$net = \sum_{i=1}^n W_i X_i + b \quad (4.1)$$

Toplam fonksiyonundan elde edilen değer doğrusal ya da doğrusal olmayan türevlenebilir bir transfer fonksiyonundan geçirilerek işlem elemanının çıktısı olarak Denklem 4.2’deki gibi hesaplanır.

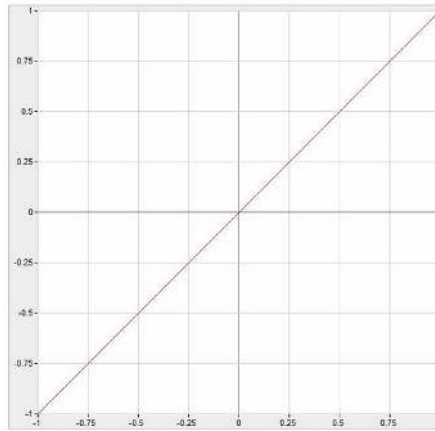
$$Y = f(net) = f\left(\sum_{i=1}^n w_i x_i + b\right) \quad (4.2)$$

4.5 Aktivasyon Fonksiyonu

Matematiksel olarak modellenmiş bir yapay sinir hücresinin birleştirme fonksiyonundan elde edilen net girdiyi bir işlemde geçirerek hücre çıktısını belirleyen ve genellikle doğrusal olmayan bir fonksiyondur. Sıkıştırma, transfer, işlemci veya eşik fonksiyonu olarak da isimlendirilebilir. Herhangi bir ağdaki tüm hücrelerin aktivasyon fonksiyonu birbirinden farklı olabilir. En çok kullanılan aktivasyon fonksiyonları aşağıda gösterilmiştir.

4.5.1 Doğrusal Aktivasyon Fonksiyonu (Lineer Fonksiyon)

Doğrusal bir problemi çözmek amacıyla kullanılan bu fonksiyon, hücrenin net girdisini doğrudan hücre çıkışı olarak verir. Doğrusal aktivasyon fonksiyonu matematiksel olarak $Y = A * v$ şeklinde tanımlanabilir. “A” sabit bir katsayıdır. Şekil4.5’de doğrusal aktivasyon fonksiyonunun grafiği gösterilmektedir.



Şekil 4.5 Doğrusal aktivasyon fonksiyonu

4.5.2 Step Aktivasyon Fonksiyonu

Step aktivasyon fonksiyonu; eğer net değeri sıfırdan küçükse sıfır, sıfırdan daha büyük bir değer ise net çıkışında +1 değerini verir. Fonksiyon doğası gereği zaman serisi analizine pek uygun olmayıp, özellikle doğrusal olmayan özellik gösteren serilerin analizinde yetersiz kalacaktır. Step fonksiyonu daha çok resim tanıma, görüntü işleme ve sınıflandırma problemlerinin çözümünde kullanılmaya uygundur. Step aktivasyon fonksiyonunu Denklem 4.3'deki gibi tarif edersek;

$$\hat{y} = f(v_i) = \begin{cases} 0 & v_i \leq 0 \\ 1 & v_i > 0 \end{cases} \quad (4.3)$$

Değer kümesi de Denklem 4.4'de gösterilmiştir.

$$\hat{y} \in (0,1) \quad (4.4)$$

Denklem 4.3 ve 4.4 yerlerine yazdıktan sonra Denklem 4.5'deki genel denklem elde edilir.

$$\hat{y} = f(v_i) = \begin{cases} 0 & \sum_{i=1}^n w_i x_i - Q \leq 0 \\ 1 & \sum_{i=1}^n w_i x_i - Q > 0 \end{cases} \quad (4.5)$$

4.5.3 Sigmoid Aktivasyon Fonksiyonu

Yapay sinir ağlarında kullanılan bu fonksiyonun türevi alınabilir, sürekli ve doğrusal olmayan bir fonksiyon olması nedeniyle doğrusal olmayan problemlerin çözümünde kullanılan en yaygın aktivasyon fonksiyonudur. Denklem 4.6'da sigmoid tipli aktivasyon fonksiyonu matematiksel olarak gösterilmiştir.

$$\hat{y} = f(v_i) = \frac{1}{1 + e^{-\beta v_i}} \quad (4.6)$$

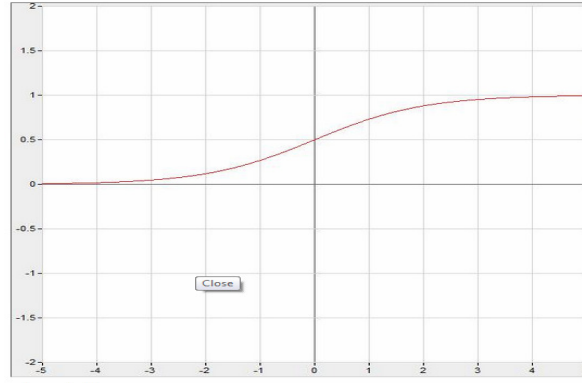
Denklem 4.6’ da verilen sigmoid tipli aktivasyon fonksiyonunun alabileceği değerlerin aralığı (değer kümesi) Denklem 4.7’de gösterilmiştir.

$$0 < \hat{y} < 1 \quad \hat{y} \in R \quad (4.7)$$

Denklem 4.6’ da verilen sigmoid aktivasyon fonksiyonunda, i v değişkeninin değeri yerine yazıldığında Denklem 4.8’ de gösterilen ifadeye ulaşılmaktadır.

$$\hat{y} = f(v_i) = \frac{1}{1 + e^{-\beta \left[\sum_{i=1}^n w_i x_i - Q \right]}} = \frac{1}{1 + e^{-\beta v_i}} \quad (4.8)$$

Şekil 4.6’da Sigmoid aktivasyon fonksiyonunun grafiği gösterilmektedir.



Şekil 4.6 Sigmoid aktivasyon fonksiyonu

Sigmoid aktivasyon fonksiyonu gibi lojistik dağılım gösteren fonksiyonlar, özellikle olasılık modellerinde (logit, probit, vs.) mükemmel bir seçim olarak kullanılabilir.

4.5.4 Hiperbolik Tanjant Aktivasyon Fonksiyonu

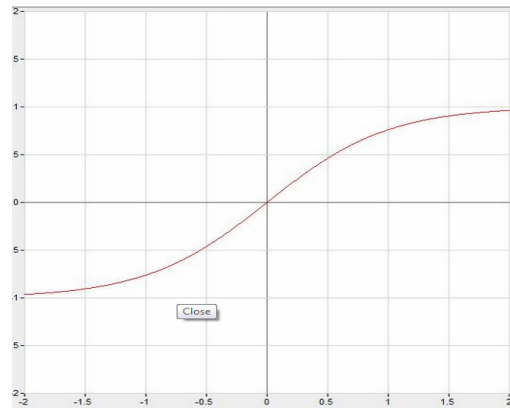
YSA' larda yoğun olarak kullanılan bir diğer aktivasyon fonksiyonu olmasıyla birlikte bu fonksiyon, sigmoid fonksiyonuna benzer bir fonksiyondur. Ayrıca bu fonksiyonun, sigmoid tipli fonksiyona göre avantajı, negatif değişkenler içeren zaman serilerini de modelleyebilmesidir. Kısaca özetlersek sigmoid fonksiyonunda çıkış değerleri 0 ile 1 arasında değişirken, hiperbolik tanjant fonksiyonunun çıkış değerleri -1 ile 1 arasında değişmektedir. Denklem 4.9' da hiperbolik tanjant tipli aktivasyon fonksiyonu verilmiştir.

$$\hat{y} = f(v_i) = \frac{e^{\beta v_i} - e^{-\beta v_i}}{e^{\beta v_i} + e^{-\beta v_i}} \quad (4.9)$$

Denklem 4.9' da verilen, hiperbolik tanjant tipli aktivasyon fonksiyonunda, i v değeri matematiksel olarak yerine yazıldığında denklem 4.10' a ulaşılmaktadır.

$$\hat{y} = f(v_i) = \frac{e^{\beta \left[\sum_{i=1}^n w_i x_i - Q \right]} - e^{-\beta \left[\sum_{i=1}^n w_i x_i - Q \right]}}{e^{\beta \left[\sum_{i=1}^n w_i x_i - Q \right]} + e^{-\beta \left[\sum_{i=1}^n w_i x_i - Q \right]}} = \tanh(\beta v_i) \quad (4.10)$$

Şekil 4.7'de hiperbolik tanjant tipli aktivasyon fonksiyonun grafiği gösterilmektedir.

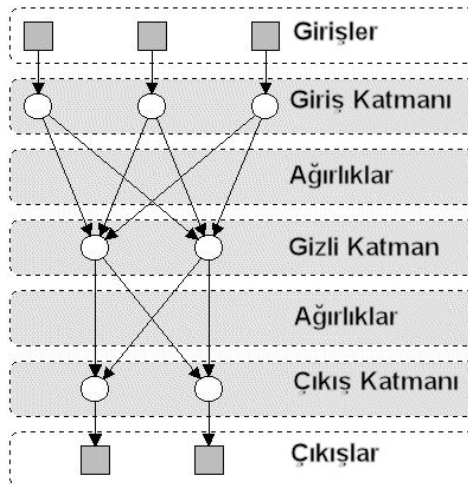


Şekil 4.7 Hiperbolik tanjant aktivasyon fonksiyonu

4.6 Yapay Sinir Ağlarının Yapısı

Yapay sinir ağları, aynen milyarlarca biyolojik sinir hücresinin birleşerek beyni oluşturduğu gibi, birden fazla yapay sinir hücresinin birleşiminden oluşur. Sinir hücreleri genellikle birkaç katman halinde dizilerek bir yapay sinir ağını meydana getirirler. Başka bir anlatımla, genellikle bir yapay sinir ağı birden fazla katmandan ve birden fazla yapay sinir hücresinden meydana gelir.

İlk katman genellikle giriş katmanıdır. Çıkış katmanı ise son katmandır. Aradaki diğer katmanlar ise gizli katman ya da ara katman olarak adlandırılırlar. Bir ağda birden fazla gizli katman olabilir. Şekil 4.8’de örnek bir yapay sinir ağı gösterilmiştir.



Şekil 4.8 Genel bir yapay sinir ağı

4.6.1 Giriş Katmanı

Bu katmandaki işlem elemanları, dışardan bilgileri alarak bir sonraki katman olan ara katmanlara transfer ederler. Bazı ağlarda girdi katmanında herhangi bir bilgi işleme olmaz.

4.6.2 Ara Katman (Gizli Katman)

Giriş katmanından gelen bilgiler işlenerek çıkış katmanına gönderilirler. Ara katman sayısı ağdan ağa değişebilir. Bazı yapay sinir ağlarında ara katman bulunmadığı gibi bazı yapay sinir ağlarında ise birden fazla ara katman bulunmaktadır. Ara katmanlardaki nöron sayıları giriş ve çıkış sayısından bağımsızdır. Birden fazla ara katman olan ağlarda ara katmanların kendi aralarındaki nöron sayıları da farklı olabilir. Ara katmanların ve bu katmanlardaki nöronların sayısının artması hesaplama karmaşıklığını ve süresini arttırmasına rağmen yapay sinir ağının daha karmaşık problemlerin çözümünde de kullanılabilmesini sağlar.

4.6.3 Çıkış Katmanı

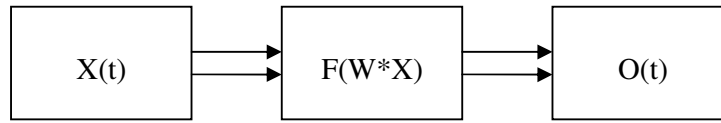
Çıkış katmandaki işlem elemanları ara katmandan gelen bilgileri işleyerek ağın girdi katmanından sunulan girdi seti için üretilmesi gereken çıktıyı üreten katmandır ve üretilen çıktı dışarıya gönderilir. Geri beslemeli ağlarda bu katmanda üretilen çıktı kullanılarak ağın yeni ağırlık değerleri hesaplanır.

4.7 Yapay Sinir Ağlarının Sınıflandırılması

Yapay sinir ağları işleyiş olarak benzer olmalarına rağmen herhangi bir tasarım ve işleyiş standardı bulunmamaktadır. Genel olarak YSA' ları, birbirleri ile bağlantılı sinir hücrelerinden oluşurlar. Her bir sinir hücresi arasındaki bağlantıların yapısı ağın yapısını belirler. Hedeflenen değere ulaşmak için bağlantıların nasıl değiştirileceği öğrenme algoritması tarafından belirlenir. Kullanılan öğrenme algoritmasına göre, hatayı sıfıra indirecek şekilde, ağın ağırlıkları değiştirilir. YSA' lar yapılarına göre sınıflandırılırlar.

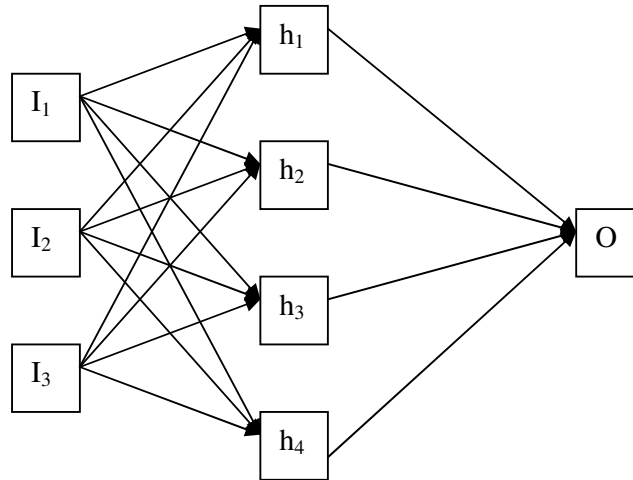
4.7.1 İleri Beslemeli Yapay Sinir Ağları (Feed Forward)

İleri beslemeli yapay sinir ağları tek katmanlı ve çok katmanlı olmak üzere iki gruba ayrılır; tek katmanlı ileri beslemeli yapay sinir ağı en basit ve temel ağ yapısıdır. Bir giriş katmanı ve bir çıkış katmanı vardır. Bu tip bir ağda bilgi girişten çıkışa doğru ilerler yani ağ ileri beslemelidir. Tek katmanlı olarak isimlendirilmesinin sebebi, giriş katmanının veri üzerinde hiçbir işlem yapmadan veriyi çıkış katmanına Şekil 4.9'daki gibi iletmesidir.



Şekil 4.9 Tek katmanlı ileri beslemeli yapay sinir ağı

Çok katmanlı ileri beslemeli yapay sinir ağında ise gizli katmanların bir den fazla mevcuttur ve Şekil 4.10'da gösterildiği gibidir. Bu sebeple giriş katmanından gelen verilere bir takım işlemler uygulanabilmektedir.



Şekil 4.10 Çok katmanlı ileri beslemeli yapay sinir ağı

İleri beslemeli yapay sinir ağlarında gecikme yoktur. Bunun sebebi işlemin girişlerden çıkışlara doğru ilerlemesidir. Çıkış değerleri, öğreticiden alınan istenen çıkış değeriyle karşılaştırılarak bir hata sinyali elde edilerek ağ ağırlıkları güncellenir.

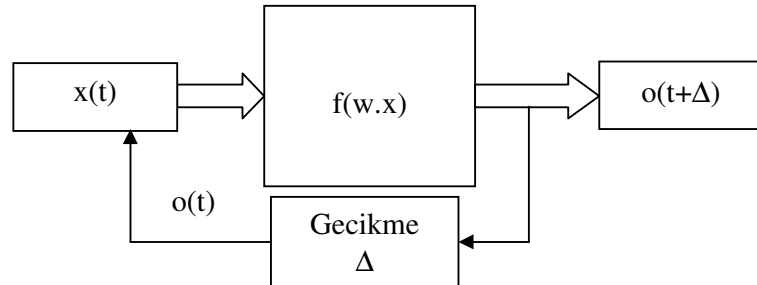
4.7.2 Geri Beslemeli Yapay Sinir Ağları

Bir geri beslemeli yapay sinir ağı, çıkış ve ara katlardaki çıkışların, giriş birimlerine veya önceki ara katmanlara geri beslendiği bir ağ yapısıdır. İleri beslemeli yapay sinir ağlarının aksine; bu ağda tıpkı kontrol uygulamalarında olduğu gibi gecikmeler söz konusudur.

Bu çeşit yapay sinir ağlarının dinamik hafızaları vardır ve her hangi bir andaki çıkış hem o andaki hem de önceki girişleri yansıtır. Bundan dolayı, özellikle önceden tahmin uygulamaları için uygundurlar. Geri beslemeli nöral ağ, çıkışları girişlerine bağlanan ileri beslemeli bir ağdan elde edilir. Ağın t anındaki çıkışı $o(t)$ ise, $t + \Delta$ anındaki çıkışı ise $o(t+\Delta)$ 'dır. Buradaki Δ sabiti sembolik anlamda gecikme süresidir. Geri beslemeli yapay sinir ağları Denklem 4.11'deki gibi ifade edilebilir.

$$o(t + \Delta) = f[W.o(t)] \quad (4.11)$$

Denklem 4.11; Şekil 4.11'de ifade edilmiştir. Dikkat edilmesi gereken nokta başlangıç anında $x(t)$ 'ye ihtiyaç duyulmasıdır. Başlangıç anında $o(0) = x(0)$ 'dir.



Şekil 4.11 Geri beslemeli yapay sinir ağı

4.7.3 Geri DönüŖlü Yapay Sinir Ağları (Recurrent)

Geri dönüŖlü yapay sinir ağlarının en büyük özelliđi hem ileri beslemeli hem de geri beslemeli çevriminin her ikisini de kendi içinde bulundurmasıdır. Geri dönüŖlü yapay sinir ağlarından en basiti ve kullanımı en kolay olanı Elman ađıdır. Geri dönüŖlü yapay sinir ađına örnek olarak Hopfield, Counterpropagation, Cognitron, Kendini Ayarlayan Haritalı Ağlar (SOM - Self Organizing Maps), Boltzman Makinesi örnek verilebilir.

Geri dönüŖlü YSA'lar özellikle birinci dereceden doğrusal sistemleri modellemekte oldukça başarılıdırlar. Zamana bađlı olayları izlemede, daha önce elde edilen sonuçları deđerlendirmedeki başarılı çıktıları ile özellikle ses ve karakter tanıma problemlerinde etkin olarak kullanılmaktadır.

4.8 Yapay Sinir Ağlarının Eğitilmesi

Dünya üzerindeki her canlı varlık doğumdan sonraki gelişme sürecinde çevresinden duyu organlarıyla algıladığı davranışları yorumlar ve bu bilgileri diğer davranışlarında kullanır.

Artık meydana gelen herhangi bir olay karşısında nasıl tepki göstereceğini çoğu zaman bilmektedir. Ama hiç karşılaşmadığı bir olay karşısında yine tecrübesiz kalabilir. Yapay sinir ağlarının öğrenme sürecinde de, tıpkı dış ortamdan gözle veya vücudun diğer organlarıyla uyarıların alınması gibi dış ortamdan girişler alınır, bu girişlerin beyin merkezine iletilerek burada değerlendirilip tepki verilmesi gibi yapay sinir ağında da aktivasyon fonksiyonundan geçirilerek bir tepki çıkışı üretilir. Bu çıkış yine tecrübeyle verilen çıkışla karşılaştırılarak hata bulunur. Çeşitli öğrenme algoritmalarıyla hata azaltılıp gerçek çıkışa yaklaşılmaya çalışılır.

Bu çalışma süresince yenilenen, yapay sinir ağının ağırlıklarıdır. Ağırlıklar her bir çevrimde yenilenerek amaca ulaşmaya çalışılır. Amaca ulaşmanın veya yaklaşmanın ölçüsü de yine dışarıdan verilen bir değerdir. Eğer yapay sinir ağı verilen giriş-çıkış çiftleriyle amaca ulaşmış ise ağırlık değerleri saklanır. Ağırlıkların sürekli yenilenip istenilen sonuca ulaşılan kadar geçen zamana öğrenme adı verilir. Yapay sinir ağı öğrendikten sonra daha önce verilmeyen girişler verilip, sinir ağı çıkışıyla gerçek çıkışı yaklaşımı incelenir.

Eğer yeni verilen örneklerle de doğru yaklaşıyorsa sinir ağı işi öğrenmiş demektir. Sinir ağına verilen örnek sayısı optimum değerden fazla ise sinir ağı işi öğrenmemiş ezberlemiştir. Genelde eldeki örneklerin yüzde sekseni ağa verilip ağ eğitilir, daha sonra geri kalan yüzde yirmilik kısım verilip ağın davranışı incelenir diğer bir deyişle ağ böylece test edilir.

4.9 Yapay Sinir Ağlarında Öğrenme Kuralları

Literatürde kullanılan çok sayıda öğrenme algoritması mevcuttur. Bu algoritmaların çoğunluğu matematik tabanlı olup ağırlıkların güncelleştirilmesi için kullanılırlar. Yapay sinir ağının mimarisi, karşılaşılan sorunun niteliğine göre farklılık gösteren bu öğrenme algoritmalarının yüzden fazla çeşidi bulunmaktadır. Bu algoritmaların birçoğu aşağıda sırlanan kurallardan esinlenerek geliştirilmiştir:

- Hebb Kuralı
- Delta Kuralı
- Kohonen Kuralı
- Hopfield Kuralı

4.9.1 Hebb Kuralı

Bu kuralın basit bir mantığı vardır; bir sinir hücresi diğer bir hücreden bilgi alırsa ve eğer her ikisi de matematiksel olarak aynı işareti taşıyorsa yani aktif ise bu iki hücre arasındaki bağlantı kuvvetlendirilmelidir. Tersi durumda ise zayıflatılmalıdır.

Ya da bu kuralı basitçe “Birlikte ateşlenenleri birlikte bağla” şeklinde özetleyebiliriz. Buna göre iki sinir hücresinin birlikte ateşlenmeleri ile ayrı ayrı ateşlenmeleri arasında sistemin eğitimi açısından fark olmaktadır. Basitçe aynı anda ateşlenen sinir hücreleri ayrı ayrı ateşlenselerdi sisteme etkileri daha düşük olur.

4.9.2 Delta Kuralı

Delta kuralı ilk olarak Widrow ve Hoff tarafından geliştirilmiş olmasıyla birlikte daha çok mühendislik kökenli bir algoritmadır ve Hebb kuralının gelişmiş şeklidir. Bu kural, nöronun gerçek çıkışı ile istenilen çıkış değerleri arasındaki farkı azaltan, giriş bağlantılarını güçlendiren ve sürekli olarak değiştiren bir düşünceye dayanmaktadır.

Bu kuralda en küçük ortalamalı kareler (least mean square) hesaplaması kullanılır. Sistemin çalışmasını basitçe beklenen değer ile gerçekleşen değer arasındaki farkın birbirine yaklaştırılması olarak düşünebiliriz.

4.9.3 Kohonen Kuralı

Kohonen öğrenme kuralı, Kohonen tabakasındaki süreç elemanlarının birbirleri ile yarışmaları ilkesine dayanır. Yani bu kuralda sinir hücreleri ağırlıklarını değiştirmek için birbirleri ile yarışır. En büyük çıktıyı üreten hücre kazanan hücre olur ve bağlantı ağırlıklarını değiştirir. Kazanan hücre yakınındaki hücrelere göre daha kuvvetli hale gelmektedir. Kohonen kuralına göre ağ kendi kendini danışmana ihtiyaç duymadan eğitebilmektedir.

4.9.4 Hopfield Kuralı

Bir farklılık dışında bu kural Hebb kuralına oldukça çok benzemektedir. Hopfield ağı tek katmanlı ve geri dönüşümlü bir ağıdır. İşlem elemanlarının tamamı hem girdi hem de çıktı elemanlarıdır. Ağın bağlantı değerleri bir enerji fonksiyonu olarak saklanmaktadır. Eğer beklenen çıktuların ve girdilerin her ikisi de aktif/pasif ise öğrenme katsayısı kadar ağırlık değerlerini de kuvvetlendirir/zayıflatır. Ağırlıkların kuvvetlendirilmesi ya da zayıflatılması öğrenme katsayısı yardımıyla gerçekleştirilir. Bu katsayı genellikle 0 ile 1 arasında kullanıcı tarafından belirlenen sabit bir pozitif değerdir.

4.10 Yapay Sinir Ağlarında Öğrenme Algoritmaları

Yapay sinir ağlarının verilen girdilere göre çıktı üretebilmesinin yolu ağın öğrenebilmesidir. Bu öğrenme işleminin de birden fazla yöntemi vardır. Yapay sinir ağları öğrenme algoritmalarına göre danışmanlı, danışmansız ve takviyeli öğrenme olarak üçe ayrılır.

4.10.1 Danışmanlı Öğrenme

Danışmanlı öğrenmede, yapay sinir ağı kullanılmadan önce eğitilmesi gerekir. Eğitim işlemi için giriş bilgilerinin yanında çıkış bilgileri de verilmelidir. Çoğu uygulama için ağa gerçek örnek kümesi verilme zorunluluğu vardır. Bu örnek kümesi ile ağ eğitilip istenen istatistiksel doğruluk elde edildiğinde eğitim işlemi tamamlanmış olur. Ağ kullanılmaya başladığında eğitim sonucunda elde edilen ağırlık değerleri çoğunlukla sabit kalır, bir daha değiştirilmez.

4.10.2 Danışmansız Öğrenme

Danışmansız öğrenmede ağa öğrenme sırasında sadece örnek girdiler verilmektedir. Herhangi bir beklenen çıktı bilgisi verilmez. Girişte verilen bilgilere göre ağ her bir örneği kendi arasında sınıflandıracak şekilde kendi kurallarını oluşturur. Ağ bağlantı ağırlıklarını aynı özellikte olan dokuları ayırabilecek şekilde düzenleyerek öğrenme işlemini tamamlar.

4.10.3 Takviyeli Öğrenme

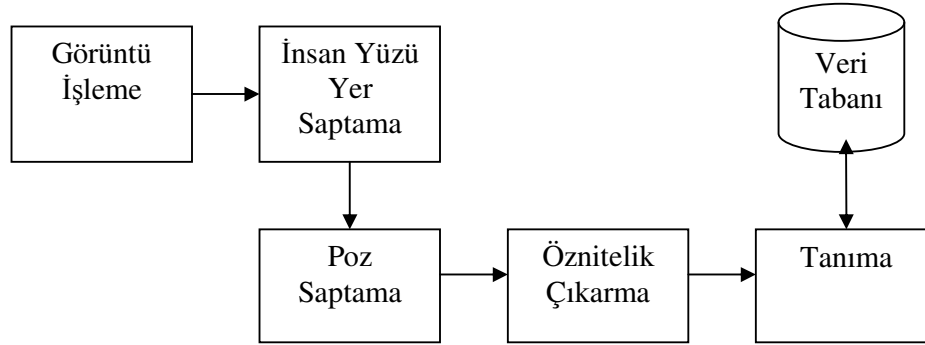
Bu öğrenme kuralı danışmanlı öğrenme kuralının özel bir şeklidir. Her girdi seti için olması (üretilmesi) gereken çıktı setini sisteme göstermek yerine, sistemin kendisine gösterilen girdilere karşılık çıktısını üretmesini bekler ve üretilen çıktının doğru veya yanlış olduğunu gösteren bir sinyal üretir. Bu sinyal dikkate alınarak, eğitim süreci devam ettirilir. (Elmas, 2003)

BÖLÜM BEŞ

YÜZ TANIMA SİSTEMİ

Son yıllarda en çok ilgi çeken örüntü tanıma konuları arasında birinci olan yüz tanımadır ve halen önemini korumaktadır. Bu sistemin; güvenlik sistemleri, savunma sistemlerinde, kredi kartı doğrulama, eğlence sektörü gibi birçok uygulama alanı mevcuttur. Günümüzde belli koşullar altında oldukça yüksek başarı gösteren sistemler mevcuttur. Ama yüzün poz açısı, aydınlanma değişimi gibi etkenler halen doğru tanımayı zorlaştırmakta ve bu etkenler altında en iyi çalışacak yüz tanıma sistemlerinin geliştirilmesi üzerine çalışmalar devam etmektedir.

Yüz tanıma esas olarak üç bölümden oluşmaktadır: Birincisi yüz bölgesinin belirlenmesi ve çıkarılması, ikincisi bulunan yüz bölgesinden öznitelik vektörlerinin çıkartılması ve sonuncu olarak da en zor kısım olan yüz tanıma işleminin gerçekleştirilmesidir. Şekil 5.1’de yüz tanıma sisteminin blok diyagramı gösterilmiştir.



Şekil 5.1 Yüz tanıma sisteminin blok diyagramı

İlk olarak yüz bölgesinin belirlenmesinde ten rengi, yüzün geometrik yapısı gibi özelliklere bakılmaktadır. Yüzün özniteliklerinin bulunması için temel bileşenler analizi, bağımsız bileşen analizi, eğim (gradient) alma yöntemi, gabor dalgacık dönüşümü, ayrık dalgacık dönüşümü, ayrık kosinüs dönüşümü temelli yöntemler önerilir. Sınıflandırma aşamasında ise destek vektör makinaları, yapay sinir ağları gibi teknikler kullanılır.

Yüz tanımda kullanılan yöntemlerde, yüz üzerinde yer alan ağız, göz, burun gibi organların konumları ile bunlar arasındaki ilişkilerden öznitelikler çıkarılarak tanıma gerçekleştirilir.

Yüz tanıma sistemlerinde kullanılan en önemli algoritma sobel kenar algıma algoritmasıdır.

5.1 Kenar Bulma Algoritması (Edge Detection)

Bilgisayarla görü ve görüntü işlemede temel kavramlardan biri olan kenar bulma, görüntü üzerindeki keskin geçişlerin olduğu noktaların tespitinde kullanılan algoritmalarla öznitelik bulma ve öznitelik oluşturma için kullanılır (Eski, 2008). Görüntünün karakteristiğine ait önemli bilgileri tutan kenarlar siyah beyaz görüntüler için tek bilgi kaynağıdır.

Kenarlar görüntünün eğim değerleri ile bulunur. Eğim görüntüdeki her piksele karşılık düşen yoğunluk ve yön bilgisini içerir. Matematiksel olarak, görüntünün yatay ve düşey yönündeki türevlerinden oluşan 2 boyutlu bir vektördür; bu vektör o noktadaki en büyük yoğunluk (intensity) değişim yönünü gösterir, vektörün büyüklüğü ise o yöndeki değişimin oranını verir.

Kullandığı maskenin hesaplama işlemi basit olması nedeniyle kenar bulma işlemi için sobel operatörü kullanılmıştır.

Sobel operatöründe yatay ve düşey yöndeki türevleri hesaplamak için 3x3'lük maskeler kullanır. I giriş görüntüsü olmak üzere, Sobel operatörlerinin cevabı bu giriş görüntüsünün 3x3'lük (S_x ve S_y ; x ve y yönündeki filtreleme sonuçlarıdır.) aşağıda verilen Denklem 5.1 de maskelerle konvolüsyonudur (Eski 2008).

$$S_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} * I \quad S_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * I \quad (5.1)$$

Elde edilen vektörünün genliği ve açısı Denklem 5.2 ile hesaplanır.

$$|s| = \left(\sqrt{s_x^2 + s_y^2} \right) \quad s_a = \left(\arctan \left(\frac{s_y}{s_x} \right) \right) \quad (5.2)$$

Sobel kenar algılama algoritması görüntü işleme algoritmaları arasında en çok kullanılan yöntemdir. Verilen herhangi bir resimdeki kenarları elde etmeye yarar. Böylelikle resimlerin içindeki nesneler algılanıp gerekli işlemler yapılabilir. Şekil 5.2’de sobel filtresi uygulanmış resim gösterilmiştir.



Şekil 5.2: Sobel filtresi uygulanmış örnek resim

5.2 Eğim Operatörü (Gradyent)

Eğim vektör, (x,y) noktasındaki, f görüntüsünün maksimum değeri yönünde bir vektördür ve birinci derece türev almada kullanılır. Bir görüntünün eğiminin hesaplanması her pikselin $\partial f / \partial x$ ve $\partial f / \partial y$ kısmi türevlerinin hesaplanmasıyla elde edilir. Denklem 5.3 de gösterildiği gibidir.

$$\nabla f = [G_x \ G_y] = \left[\frac{\partial f}{\partial x} \ \frac{\partial f}{\partial y} \right] \quad \nabla f = \left[(G_x^2) + (G_y^2) \right]^{1/2} \quad (5.3)$$

BÖLÜM ALTI

YÜZ TANIMA UYGULAMASI

Bu çalışmada insan yüzünün ön profilden çekilmiş fotoğraflarıyla değişik yöntemler kullanarak yüzden her insana ait öznitelikler çıkarılmıştır. Elde edilen öznitelikleri yapay sinir ağında kullanarak tanıma işlemi gerçekleştirilmiştir. Yapılan işlemler aşağıdaki sıralamaya göre takip edilmiştir;

İlk baştan değişik yollardan elde edilen resim dijital ortama aktarılmıştır. Aktarılan resimde yüz bölgesinin tespiti için ten rengine bağlı çıkartım, yüz özniteliklerin belirlenmesinde resmin eğimi (gradienti) alınmıştır ve en son olarak da tanıma ve sınıflandırmada yapay sinir ağı kullanılmıştır.

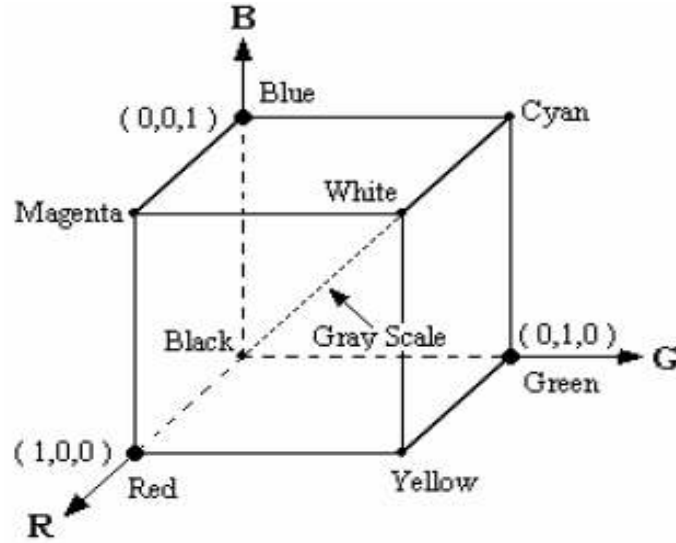
Bu işlemleri yapmadan önce bazı renk sistemlerinin ve renk uzaylarının bilinmesi gerekir.

6.1 Renk Sistemleri Ve Renk Uzayları

Renk uzayları renkleri tanımlamak için kullanılan matematiksel modellerdir. Renk uzayları, bütün renkleri temsil edecek şekilde oluşturulur ve 3D olarak tasarlanır. Her renk uzayının kendine özgü biçimde renk oluşturma için bazı standartları vardır. Renk uzayları oluşturulurken bir başka renk uzayına doğrusal ya da doğrusal olmayan yöntemlerle dönüşüm yapılabilir.

6.1.1 RGB Renk Uzayı

RGB (kırmızı, yeşil, Mavi: KYM) modeli kartezyen koordinat sisteminin her bir eksenine üç ana renge karşılık gelecek şekilde tasarlanmıştır ve Şekil 6.1'deki gibi gösterilir.



Şekil 6.1 RGB modelinin kartezyen koordinat sisteminde gösterimi

Herhangi bir rengi koordinat sisteminde göstermek için, bu üç bileşen kullanılır. RGB modeli ile ifade edilen bir resmin her bir beneğine karşılık gelen renk K,Y,M nin belli katsayılar ile birleşiminden oluşur. Bu durumda K, Y, M için ayrı matrisler oluşturulabilir.

6.1.2 Lab Renk Uzayı

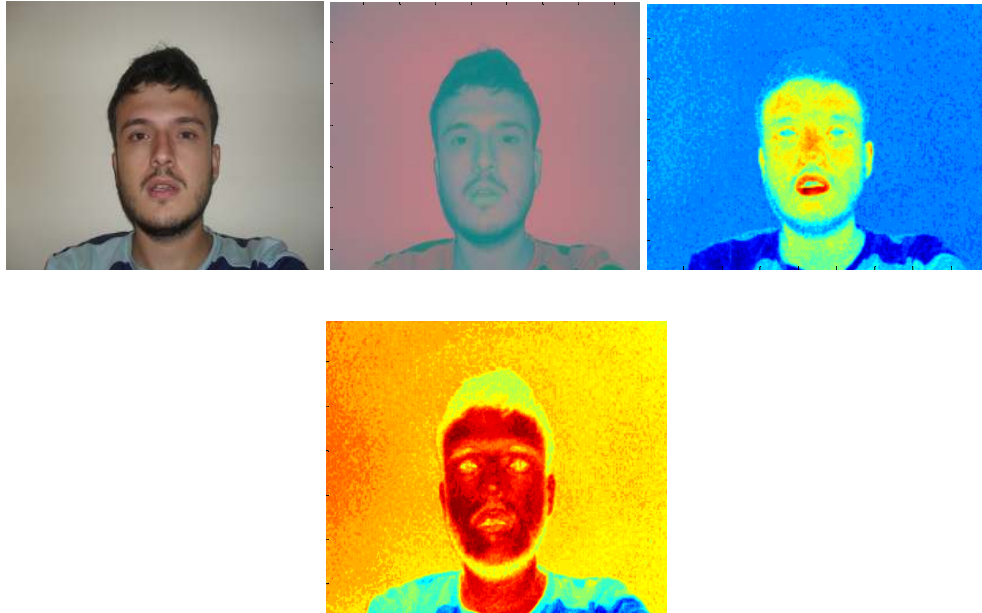
Lab renkleri, insan gözünün görebildiği renklerden de fazlasını kapsayan bir renk uzayıdır. Lab bileşenleri şöyledir:

- L^* : Rengin parlaklığını belirtir, 0 değeri siyahı verirken, 100 değeri dağılmış beyaz rengi verir
- a^* : Kırmızı/magenta ile yeşil arasındaki renk dengesini belirtir, eksi değerler yeşili gösterirken artı değerler magentayı gösterir.
- b^* : Sarı-mavi arasındaki renk dengesini belirtir, eksi değerler maviyi gösterirken artı değerler sarıyı gösterir.

6.2 Ten Rengine Bağlı Yüz Tespiti

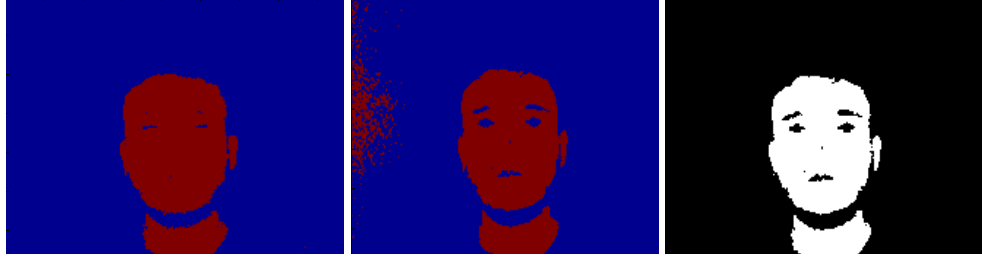
Renk bilgisine bağlı yüz tespiti, yüz tanımda sıkça başvurulan yöntemlerden birisidir. Bu yöntem diğer yöntemlere göre hem daha hızlı hem de daha basittir. Sisteme girilen resimler ilk başta boyutları küçültülür daha sonra resimlerin renk uzaylarının değişimine göre hesaplanarak çıkarılır.

İlk baştan dijital ortama aktarılan resmin renk uzayını RGB den Lab uzayına dönüştürülür. Dönüştürülen resmin Lab renk uzayında ikinci ve üçüncü bileşenine ayrılır. Ayrılan bileşenler Şekil 6.2’de gösterilmiştir.



Şekil 6.2 Sırasıyla RGB resim, Lab resim, Lab resimin ikinci bileşeni, Lab resimin üçüncü bileşeni

İkinci ve üçüncü seviyesi bulunan resimler Lab uzayında siyah-beyaz resme dönüştürülür. Şekil 6.3’de Lab uzayındaki ikinci ve üçüncü bileşenlerin siyah beyaz resme dönüştürüldüğü gösterilmiştir. Dönüştürme işlemi tamamlandıktan sonra gerekli matematiksel işlemler (iki resim çarpılır) ve dönüşümler (Lab uzayından RGB uzayına) resim üzerinde yapılır ve resim üzerindeki alan hesabından yüz kısmı tespit edilir. Resimdeki yüz bölgesi tespit edildikten sonra resim siyah- beyaz dan yeniden RGB renk uzayına dönüştürülür.



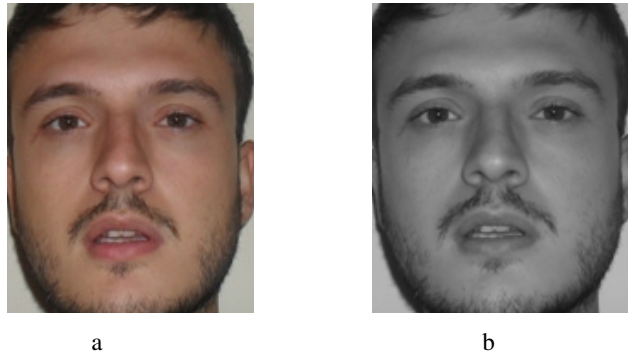
Şekil 6.3 Sırasıyla Lab uzayının 2. ve 3. bileşenlerinin siyah-beyaz resimleri ve çarpılmış hali (RGB)

Resim içinde yüz belirlendikten sonra yüzün dışında kalan fazlalıklar Şekil6.4'deki gibi kesilerek atılır. Bu işlem sistemin gereksiz kısımlarını işleyerek zaman harcamasını engeller hem de sistemin daha hızlı çalışmasını sağlar.



Şekil 6.4 Sırasıyla orijinal resim, kesildikten sonraki resim

RGB modelindeki resim üzerinde işlem yapıp ayırt edici özellikleri seçmek her zaman gri seviyedekine göre daha zordur. Bu yüzden kesilip alınan bölgenin rengini Şekil 6.5'deki gibi gri seviyeye getirilerek işlemlere devam edilir.

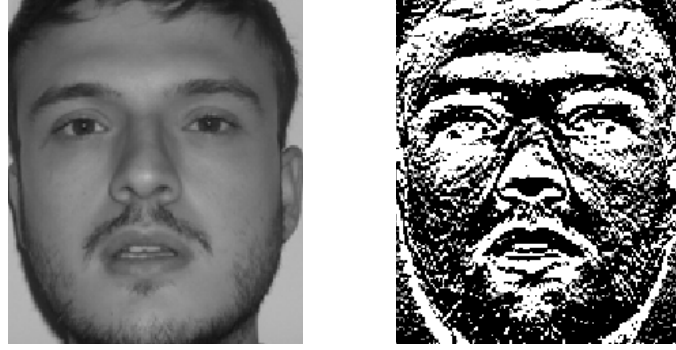


a

b

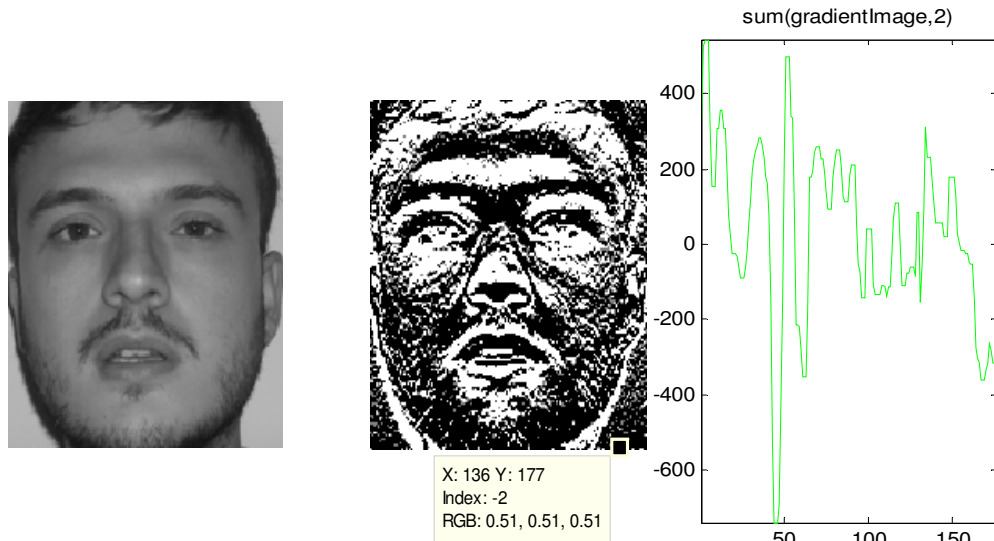
Şekil 6.5 a)Orijinal resim RGB b)Resmin gri renk seviyesine Dönüştürülmüş hali

İnsan yüzündeki göz, burun ve ağız gibi önemli bölgelerin bulunması için geliştirilmiş yöntemlerden farklı olarak bu çalışmada eğim alma yaklaşımı kullanılmıştır. Burada kesilen ve gri seviyeye dönüştürülen resmin Matlab de eğimi alınır. Şekil 6.6'da gri seviyeli resim ve eğimi alınmış resim gösterilmiştir.



Şekil 6.6 Sırasıyla orijinal gri seviyeli resim, eğimi alınmış resim

Daha sonra eğimi alınan resmin yatay izdüşümü alınır. Bu işlem eğimi alınan resmin yataylamasına olan tüm indeks değerlerinin matlab komutları kullanılarak toplanmasıdır. Bu işlemleri yaparken resim üzerindeki siyah noktaların negatif değer aldığı ve beyaz noktaların ise pozitif değerler aldığı Şekil 6.7'deki resimler incelendiğinde tespit edilmiştir.



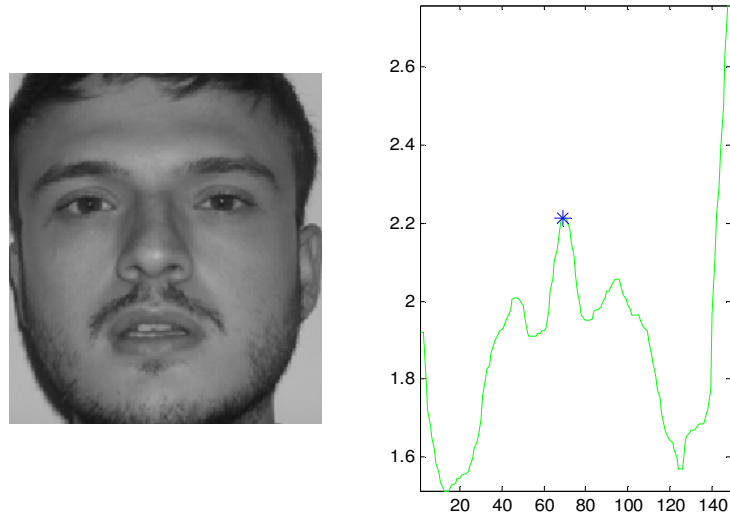
Şekil 6.7 Sırasıyla; orijinal resim (gri seviyedeki), eğimi alınmış resim, eğimi alınmış resmin yatay izdüşümü

Şekil 6.7’de ki grafik incelendiğinde grafiğin en düşük değerinden en yüksek değerine çıktığı aralık eğimi alınan insan yüzünün kaşının üst kısmı ve alt kısmının denk geldiği Şekil 6.8’de tespit edilmiştir.



Şekil 6.8 Yatay izdüşümünün en düşük ve en yüksek değerler aralığı

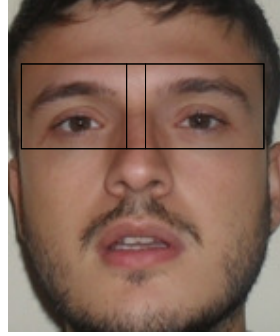
Resmin yatay izdüşümü alındıktan sonra dikey izdüşümü alınır. Bu işlemin yatay izdüşümünden tek farkı resimdeki yatay indeksleri değil dikey indekslerin toplanmasıdır. Şekil 6.9’deki grafikte dikey izdüşümü alınan resim görülmektedir.



Şekil 6.9 Sırasıyla; orijinal resim (gri seviyedeki), resmin dikey izdüşümü

Elde edilen grafikte mavi nokta ile gösterilen yerin resmin orta noktasına denk geldiği görülmektedir.

Yatay ve düşey izdüşümleri alınan resimlerden bulunan üç özel değeri (kaşın üst ve alt yeri, yüzün orta noktası) kullanarak insan yüzü standartları çerçevesinde göz mesafesini kutu içine alacak şekilde belirli bir miktar çıkartıldıktan / toplandıktan sonra Şekil 6.10'daki gibi dikdörtgen çizilir.



Şekil 6.10 Göz Bölgesi

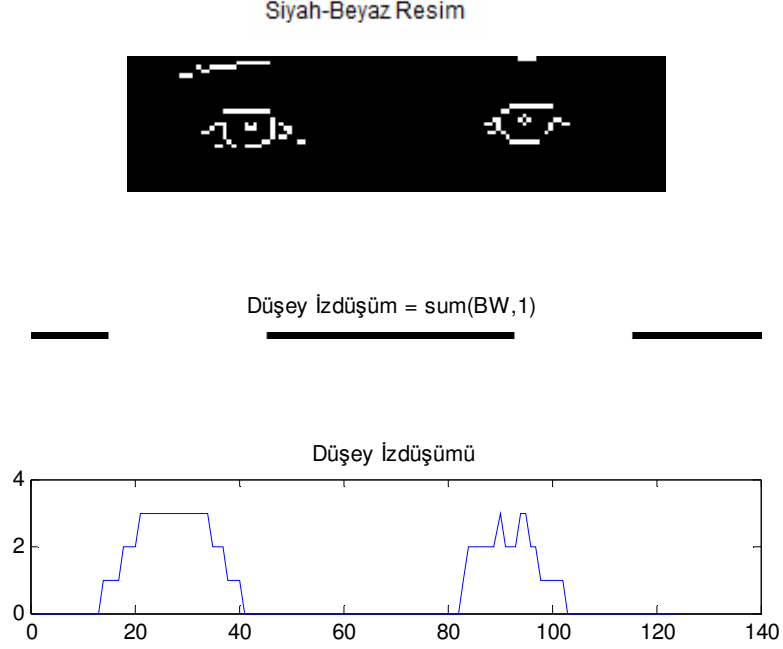
Elde ettiğimiz kutunun dışındaki diğer fazlalıklar Matlab da çıkartıldıktan sonra resme Sobel kenar algılama algoritması işlemi uygulanmıştır. Bu işlemin sonucunda Şekil 6.11 elde edilmiştir.



Şekil 6.11 Sobel filtresi (göz bölgesi)

Elde edilen resimden gözün başlangıç ve bitiş noktalarını bulmak için matlab de resim bir takım işlemlere tabi tutulmuştur. Bu işlem sayesinde resmin tüm değerlerini bir doğru üzerine dizmiş oluruz.

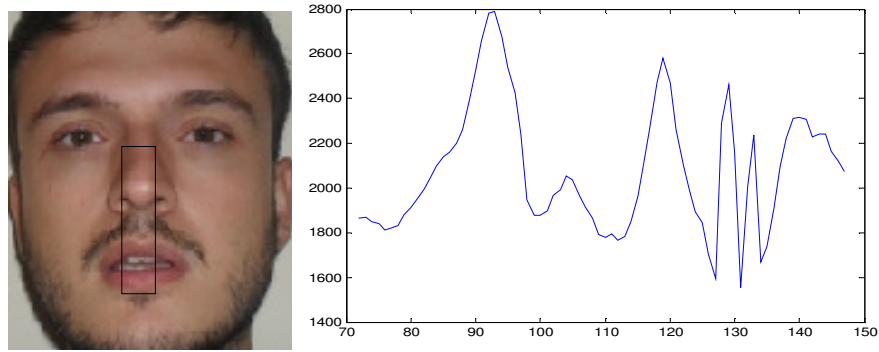
Bu işlem sonucunda Şekil 6.12'deki sayı doğrusuna baktığımızda belli aralıklarla sıfır sayılarının bir arada toplandığını görürüz.



Şekil 6.12 Sobel filtresi uygulanmış göz bölgesinin düşey izdüşümü

Bu toplu halde bulunan sıfırların ilkinin sonu sağ gözün dış başlangıcına denk geldiği ve ikinci sıfır topluluğunun başlangıcı sağ gözün bitiş kısmına denk geldiği tespit edilmiştir. Aynı şekilde ikinci sıfır kümesinin bitişi ve üçüncü sıfır kümesinin başlangıcı da sol gözün başlangıcı ve bitişiye denk gelmektedir.

Daha önceden bulunan resmin orta noktasından (insan yüzü standartlarına uygun) belirli miktarlarda aşağı, sağa, sola gidilerek; burun ucu ve dudak kesimini içine alan bir dikdörtgen çizilmiştir. Elde ettiğimiz dikdörtgenin yatay izdüşümü alınır. Alınan iz düşümü Şekil 6.13'de gösterildiği gibidir.



Şekil 6.13 Burun ve dudak bölgesinin tespiti

Şekil 6.13’deki grafik incelendiğinde en yüksek değere karşılık gelen nokta burun ucuna, en düşük değere karşılık gelen noktanın dudak kesimi olduğu tespit edilmiştir.

Tüm işlemleri gerçekleştirdikten sonra her birine nokta işareti koyarız. Şekil6.14’de değişik kişilerin gözlerinin başlangıcı, bitiş noktası, burun ucu ve dudak kesim noktaları (öz nitelikler) değişik örneklerle gösterilmiştir.



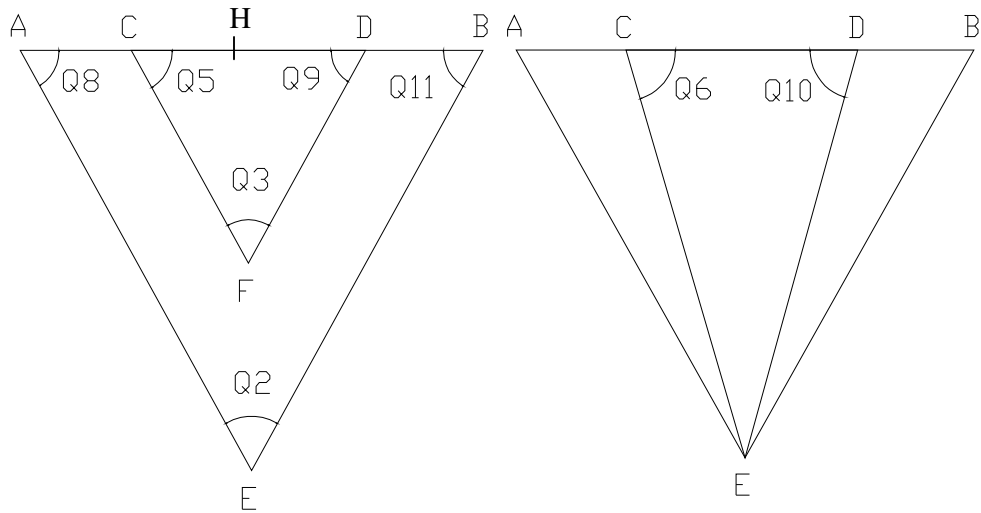
Şekil 6.14 Öz nitelikleri bulunmuş örnekler

Sistemimiz doğru öz nitelikler bulmasına rağmen bazı resimlerde değişik sorunlar yüzünden doğru sonuç vermemektedir. Bu sorunlar arasında poz açısı, çekim hataları, ameliyat izleri, gözlük, dış etmenler, insan yüzü biyolojisi... vb diye sıralanabilir. Şekil 6.15’de örnek birkaç resim gösterilmiştir.

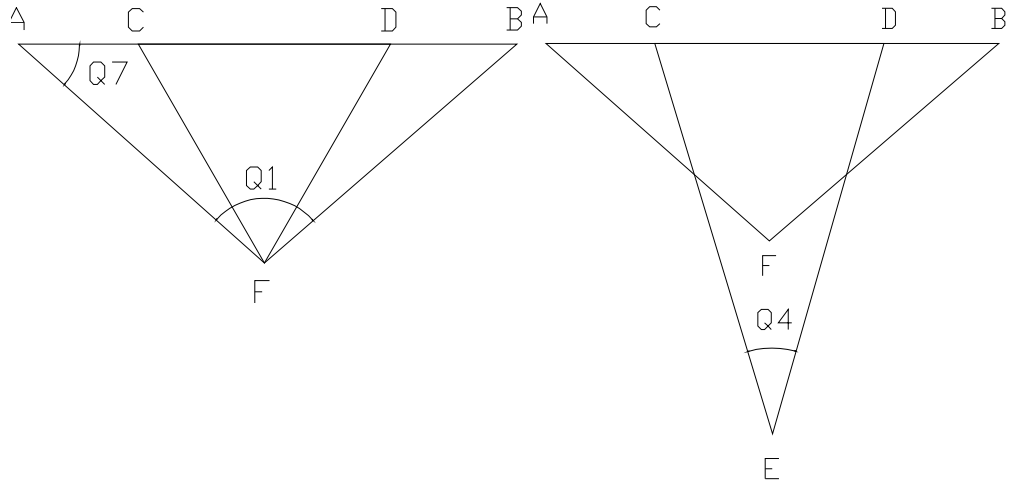


Şekil 6.15 Hatalı öz nitelik

Bu işlemler tamamlandıktan sonra elimizde her bir yüze ait 6 özel nokta bulunmaktadır. Bu noktaları birleştirerek 4 farklı üçgen çizilmiştir. Şekil 6.16 ve Şekil 6.17’de bu üçgenler gösterilmiştir. Bu üçgenlerin açı ve kenar uzunlukları hesaplanmıştır.



Şekil 6.16 Elde edilen özel üçgenler-1-



Şekil 6.17 Elde edilen özel üçgenler-2-

Bu dört farklı üçgenden Matlab de mesafe ve açı bulma fonksiyonları yazılarak her resimden belirli sayıda açı ve mesafe değerleri elde edilmiştir. Bu elde ettiğimiz değerler her insana özgü olan değerlerdir. Bu açı ve mesafelerin yerleri aşağıdaki gibidir.

$Q1 = \text{Burun ucunun}$ sağ ve sol gözlerin dış noktaları ile birleşmesinden oluşan açı değeri.

$Q2 = \text{Dudak kesiminin}$ sağ ve sol gözlerin dış noktaları ile birleşmesinden oluşan açı değeri.

$Q3 = \text{Burun ucunun}$ sağ ve sol gözlerin iç noktaları ile birleşmesinden oluşan açı değeri.

$Q4 = \text{Dudak kesiminin}$ sağ ve sol gözlerin iç noktaları ile birleşmesinden oluşan açı değeri.

Q5 = Sağ gözüün iç kısmının burun ucu ile sol gözüün iç kısmıyla yaptığı açısı.

Q6 = Sağ gözüün iç kısmının dudak kesimi ile sol göz iç kısmıyla yaptığı açısı.

Q7 = Sağ gözüün dış kısmının burun ucu ile sol göz dış kısmıyla yaptığı açısı.

Q8 = Sağ gözüün dış kısmının dudak kesimi ile sol göz dış kısmıyla yaptığı açısı.

Q9 = Sol gözüün iç kısmının burun ucu ile sağ göz iç kısmıyla yaptığı açısı.

Q10 = Sol gözüün iç kısmının dudak kesimi ile sağ göz iç kısmıyla yaptığı açısı.

Q11 = Sol gözüün dış kısmının dudak kesimi ile sağ göz dış kısmıyla yaptığı açısı.

Yukarıdaki değerler elde edildikten sonra açısı değerleri Q2 açısına göre (Q2 yerine diğer açılarda olabilir.) normalize edilmiştir. Açılardan toplamda 11 tane öz nitelik değeri elde edilmiştir. Resimdeki açılar bulduktan sonra yüzde bulduğumuz noktaların arasındaki mesafede aşağıdaki gibidir.

D1 = Burun ucunun dudak kesimine olan mesafesi.(EF arasındaki mesafe)

D2 = Sağ ve sol iç gözlerin arasındaki mesafe. (CD arasındaki mesafe)

D3 = Sağ ve sol dış gözlerin arasındaki mesafe.(AB arasındaki mesafe)

D4 = Dudak kesiminin yüzün merkezine olan mesafesi. (EH arasındaki mesafe)

D5 = Burun ucunun yüzün merkezine olan mesafesi. (FH arasındaki mesafe)

D6 = Dudak kesiminin sağ dış göze olan mesafesi. (EA arasındaki mesafe)

D7 = Burun ucunun sağ iç göze olan mesafesi. (FA arasındaki mesafe)

D8 = Dudak kesiminin sol dış göze olan mesafesi. (EB arasındaki mesafe)

D9 = Burun ucunun sol iç göze olan mesafesi. (FD arasındaki mesafe)

D10= Sağ iç gözüün dudak ucuna olan mesafesi. (CE arasındaki mesafe)

D11=Sol iç gözüün dudak ucuna olan mesafesi. (DE arasındaki mesafe)

Resimdeki yüzlerin kameraya uzaklıklarının ne kadar aynı olmasını istesek de küçükte olsa oynamalar mevcuttur. Bu yüzden bulduğumuz kenar uzunlukları aynı kişide farklılık göstermektedir. Bu farklılığı en aza indirmek için kenarların oranlarını kullanarak 9 tane öz nitelik elde edilmiştir. Bu oranlar aşağıdaki gibidir.

$$L1= D3/D2 \longrightarrow (AB/CD)$$

$$L2= D4/D5 \longrightarrow (EH/FH)$$

$$L3= D6/D7 \longrightarrow (EA/FA)$$

$$L4= D8/D9 \longrightarrow (EB/FD)$$

$$L5= D2/D1 \longrightarrow (CD/EF)$$

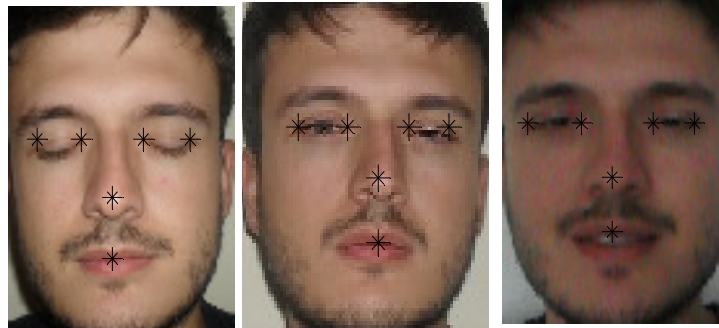
$$L6= D10/D11 \longrightarrow (CE/DE)$$

$$L7= D2/D5 \longrightarrow (CD/FH)$$

$$L8= D3/D4 \longrightarrow (AB/EH)$$

$$L9= D5/D1 \longrightarrow (FH/EF)$$

Yukarıdaki işlemleri aynı kişinin farklı resimleri üstünde denediğimizde Şekil6.18 ve Tablo6.1'deki veriler elde edilmiştir.



Şekil 6.18 Aynı kişinin farklı resimlerdeki öz nitelik noktaları

Tablo6.1 Aynı kişini farklı resimlerindeki öz nitelik değerleri; a=uzunluk oranı b= açı oranı

No	L1	L2	L3	L4	L5	L6	L7	L8	L9
1	2,4364	2,0217	2,1582	2,1186	1,1702	0,99127	1,1957	1,4409	0,97872
2	2,5471	2,1052	2,3643	2,2867	1,0625	0,99102	1,2593	1,5254	0,84375
3	2,3857	2,0392	2,2787	2,0334	1,3208	0,96581	1,3725	1,5058	0,96226

a

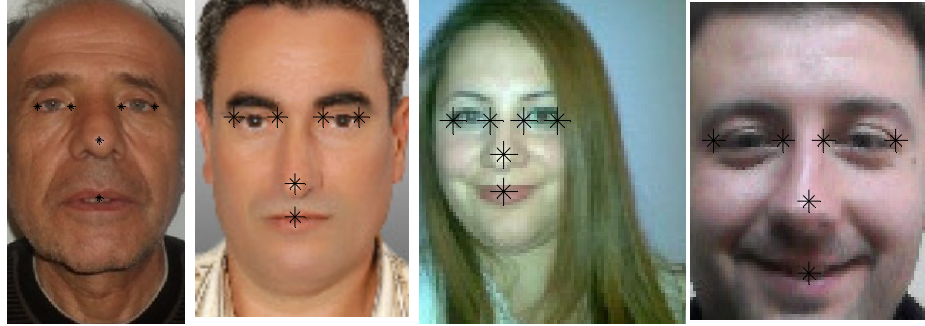
No	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11
1	1,5523	1	0,8625	0,4604	0,8460	1,039	0,4875	0,7637	0,80758	1,016	0,7524
2	1,5813	1	0,8617	0,4304	0,7948	1,002	0,4146	0,7053	0,75415	0,978	0,7053
3	15114	1	0,8826	0,4785	0,7788	0,96	0,4033	0,6588	0,66052	0,883	0,6631

b

Şekil 6.18'deki aynı kişinin farklı şekillerde çekilen resimlerinden elde edilen öz nitelikleri Tablo 6.1'deki gibi olduğu tespit edilmiştir. Elde edilen veriler incelendiğinde her üç resmin aynı yerdeki öz nitelik değerlerinin yaklaşık olarak aynı çıktığı görülmüştür. Bu bizim sistemimizin, değişik pozlarda resimlerin çekilse bile doğru öz nitelikler çıkardığını göstermiştir.

Çıkan öz nitelikler arasında çok az da olsa bir farklılık gözlenmiştir. Bu farklılığın nedenleri arasında resim çekilirken yerle yüz arasındaki açının tam 90 derece olmamasından, resim çekim hatasından, poz değişkenliği, dış etmenler... vb sebeplerden meydana geldiği gözlemlenmiştir.

Şekil 6.19’da farklı kişilerin öz niteliklerinin başarılı şekilde bulunduğu gözlemlenmiştir ve Tablo 6.2’de bu öz niteliklerin sayısal değerleri gösterilmiştir.



Şekil 6.19 4 farklı kişinin öz nitelik noktaları

Tablo 6.2 Farklı kişilerin öz nitelik değerleri; a=uzunluk oranı b= açı oranı

No	L1	L2	L3	L4	L5	L6	L7	L8	L9
1	2,3889	2,68	2,5209	2,6464	0,8571	1,0151	1,44	1,2836	0,59524
2	2,6522	1,4923	1,6842	1,6356	1,4375	0,9816	0,707	1,2577	2,0313
3	3,1364	2	2,3248	2,1694	0,9565	0,9711	0,956	1,5	1
4	3,875	2,28	2,2811	2,6247	0,6250	0,6250	1,015	0,8	1,3596

a

No	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11
1	1,8297	1	1,0907	0,4598	0,7856	1,22	0,4459	0,8864	0,9778	1,2727	0,9571
2	1,3425	1	0,6039	0,4141	1,1458	1,22	0,7351	0,9057	1,0485	1,1573	0,8925
3	1,5771	1	0,6852	0,3635	0,9609	1,08	0,4731	0,7373	0,7960	0,9914	0,7049
4	1,6712	1	0,6324	0,2903	0,9272	1,13	0,4725	0,8080	1,0715	1,206	0,8231

b

Bu öz niteliklerin değerleri incelendiğinde birkaç değer haricinde resimlerin kendi aralarında bir ilişki ya da yakın bir değer elde edilmemiştir. Bu da resimler arasında ayırt edici bir etmen olarak başarıya ulaşılmıştır.

Toplamda her bir resim için 20 öz nitelik değeri elde edilmiştir. Bu bulunan öz nitelikler her insanda farklılık gösterir. Yapacağımız tanıma işleminde ne kadar fazla örnek değerimiz olursa sistemin performansı o kadar iyi olur.

Bu sistemde her bir kişi için 30 tane farklı resim kullanılmıştır. Bu aynı kişiden aldığımız 30 farklı resmin her birinden 20 farklı öz nitelik değeri elde edilmiştir.

Elimizde var olan her kişiden toplam 30 tane resmin 22 tanesini yapay sinir ağının eğitim verisi olarak kullanılmıştır. Yapay sinir ağının test verisi olarak 30 resimden 5 tanesini ve geriye kalan 3 resimde tanıma işlemini sırasında matlab GUI da veri tabanından çıkacak resim olarak kullanılmıştır.

Tüm bulduğumuz bu sayısal verileri toplu halde bir araya getirdikten sonra matlab de yapay sinir ağı uygulamasına geçilmiştir. Yapay sinir ağı mimarisinde sistemimize en uygun sonucu vermesi için uygun sayıda nöron, aktivasyon sayısı ve cinsi, eğitim algoritması ve ağı eğitimi için temel parametreler seçilmiştir. Seçilen parametreleri yerlerine yerleştirdiğimizde Denklem 6.1 elde edilmiştir.

```
Net = newff (minmax (trainingData) , [16 1] , { 'tansig' , 'purelin' },
'trainlm');

net.trainParam.show=25;
net.trainParam.epochs=10000;
net.trainParam.goal=0.04;
```

(6.1)

Denklem 5.1 e bakıldığında net adlı bir ağı oluşturulmuş olduğu görülmektedir. Bu ağı iki katmandan meydana gelmektedir. İlk katmanında 16 giriş elemanı ve 1 çıkış elemanı mevcuttur. Her bir katmanda kullanılan aktivasyon fonksiyonları ise ilk katmanda tansig ikinci katmanda ise purelin aktivasyon fonksiyonu işlem görmektedir. En son kısımda eğitim algoritması olarak trainlm algoritması kullanılmıştır.

Tansig: Bu aktivasyon fonksiyonun dinamik değişim aralığı $[-1 \ 1]$ aralığıdır ve fonksiyon nöron toplam girişe bağımlı olarak bu aralıkta lineer olmayan bir değişim gösterir. Bu fonksiyon literatürde hiperbolik-tangent fonksiyonu olarak da isimlendirilmektedir.

Purelin: Bu aktivasyon fonksiyonunda nöron girişlerinin değişimine göre nöron çıkışı lineer olarak değişmektedir. Dinamik değişim aralığı $[-1 \ 1]$ aralığıdır.

Yapay sinir ağıının performansını etkileyen faktörlerden birisi de eğitim algoritmasıdır. Yapılan denemeler sonucunda eğitim algoritması olarak Levenberg-Marquart (trainlm) algoritmasının kullanıldığı zaman ağ yapısının iyi sonuçlar verdiğinden ve diğerlerine göre daha hızlı bir eğitim algoritması olduğundan trainlm seçilmiştir.

Yapay sinir ağıında show parametresi kaç tekrarda (iterasyonda) bir eğitim durumunun matlab ekranına aktarılacağını belirtmekte, epoch parametresi tekrar (iterasyon) sayısını, goal parametresi hedeflenen hata değerini ifade etmektedir. Bu parametrelerden epoch veya goal değerlerinden birisi sağlandığında eğitim işlemi durdurulacaktır.

Yapay sinir ağıında ağıın eğitime başlaması için train komutu kullanılır. Bileşenlerini tamamladığımız yapay sinir ağıının eğitime başlaması için Denklem 6.2'deki kod yazılarak eğitime başlatılmıştır.

$$[net, tr] = \text{train}(net, \text{trainingData}, \text{trainingLabels}); \quad (6.2)$$

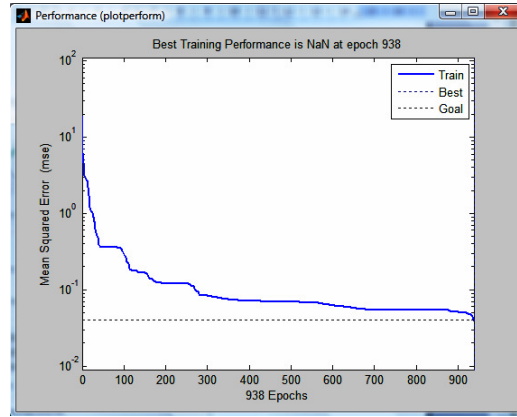
Hedeflenen hata oranına erişilmiş olup ağıın eğitimi durdurulmuştur. Ağıın eğitilip eğitilmediğini test etmek için giriş verisi ile ağıı tekrardan karşılaştırılıp sistemimizin yani tanıma işlemimizin performansı ölçülmüştür. İki grup halinde oluşturulan veri tabanının ilki 10 farklı kişiden ikincisi 15 farklı kişiden oluşturulmuştur.

10 kişilik veri tabanının verilerini yapay sinir ağıına uygun girişlerle eğittikten sonra sistemimizin performansı ve kullanılan yapay sinir ağıı elamanları Tablo 6.3'de gösterilmiştir.

Tablo 6.3 10 kişilik yapay sinir ağıı sonuçları

Nöron Sayısı	Aktivasyon Cinsi	Eğitim Algoritması	Goal	Performans
[8 1]	'tansig', 'purelin'	'trainlm'	0,01	86
[8 1]	'tansig', 'purelin'	'trainlm'	0,02	90
[8 1]	'tansig', 'purelin'	'trainlm'	0,03	94
[8 1]	'tansig', 'purelin'	'trainlm'	0,04	98
[8 1]	'tansig', 'purelin'	'trainlm'	0,05	88

Elde ettiğimiz yapay sinir ağı performans grafiği Şekil 6.20’de gösterilmiştir. Sistemin goal değeri 938. tekrarda (iterasyonda) sağladığı tespit edilmiştir. 938. tekrarda (iterasyonda) sistemin performansının yüzde 98 olduğu görülmüştür.



Şekil 6.20 10 Kişilik performans eğrisi

Sistemin performansının en yüksek olduğu değerde kullanılan eğitim ve test setlerindeki performans yüzdeleri Tablo 6.4 de gösterildiği gibidir.

Tablo 6.4 Yapay Sinir Ağı Başarı Durumu

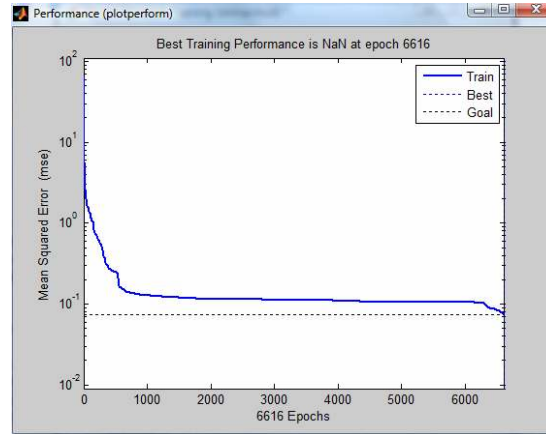
Nöron Sayısı	Tekrar	Goal	Eğitim Seti			Test Seti		
			Veri Sayısı	Doğru	Yüzde	Veri Sayısı	Doğru	Yüzde
[8 1]	938	0,04	220	216	98,18	50	49	98

15 kişilik veri tabanı için uygulanan yapay sinir ağı elemanları Tablo 6.5'te gösterilmiştir.

Tablo 6.5 15 kişilik yapay sinir ağı sonuçları

Nöron Sayısı	Aktivasyon Cinsi	Eğitim Algoritması	Goal	Performans
[16 1]	'tansig', 'purelin'	'trainlm'	0,073	79

Uygulanan yapay sinir ağının goal=0,073 değerinde performansının yüzde 79 olduğu görülmüştür. Bu değeri sistem 6616. tekrarda (iterasyonda) ulaştığı tespit edilmiştir. Sistemin goal değerine ulaştığı andaki performans grafiği Şekil 6.21'da gösterilmiştir.



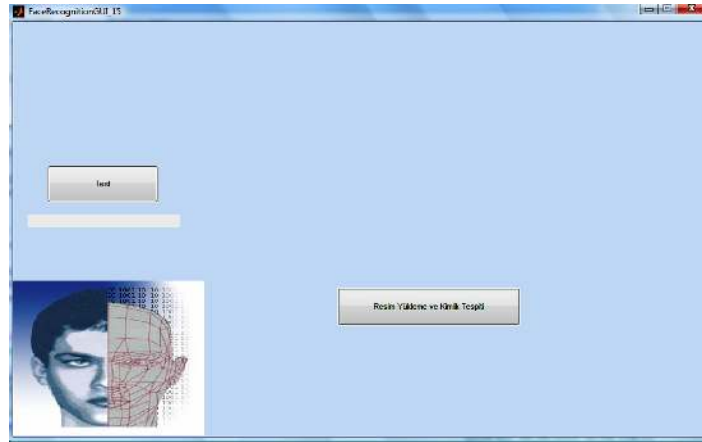
Şekil 6.21 15 Kişilik performans eğrisi

Sistemin performansının en yüksek olduğu değerde kullanılan eğitim ve test setlerindeki performans yüzdeleri Tablo 6.6 da gösterildiği gibidir.

Tablo 6.6 Yapay sinir ağı başarı durumu

Nöron Sayısı	Tekrar	Goal	Eğitim Seti			Test Seti		
			Veri Sayısı	Doğru	Yüzde	Veri Sayısı	Doğru	Yüzde
[16 1]	6616	0,073	330	208	93.3	75	59	78,9

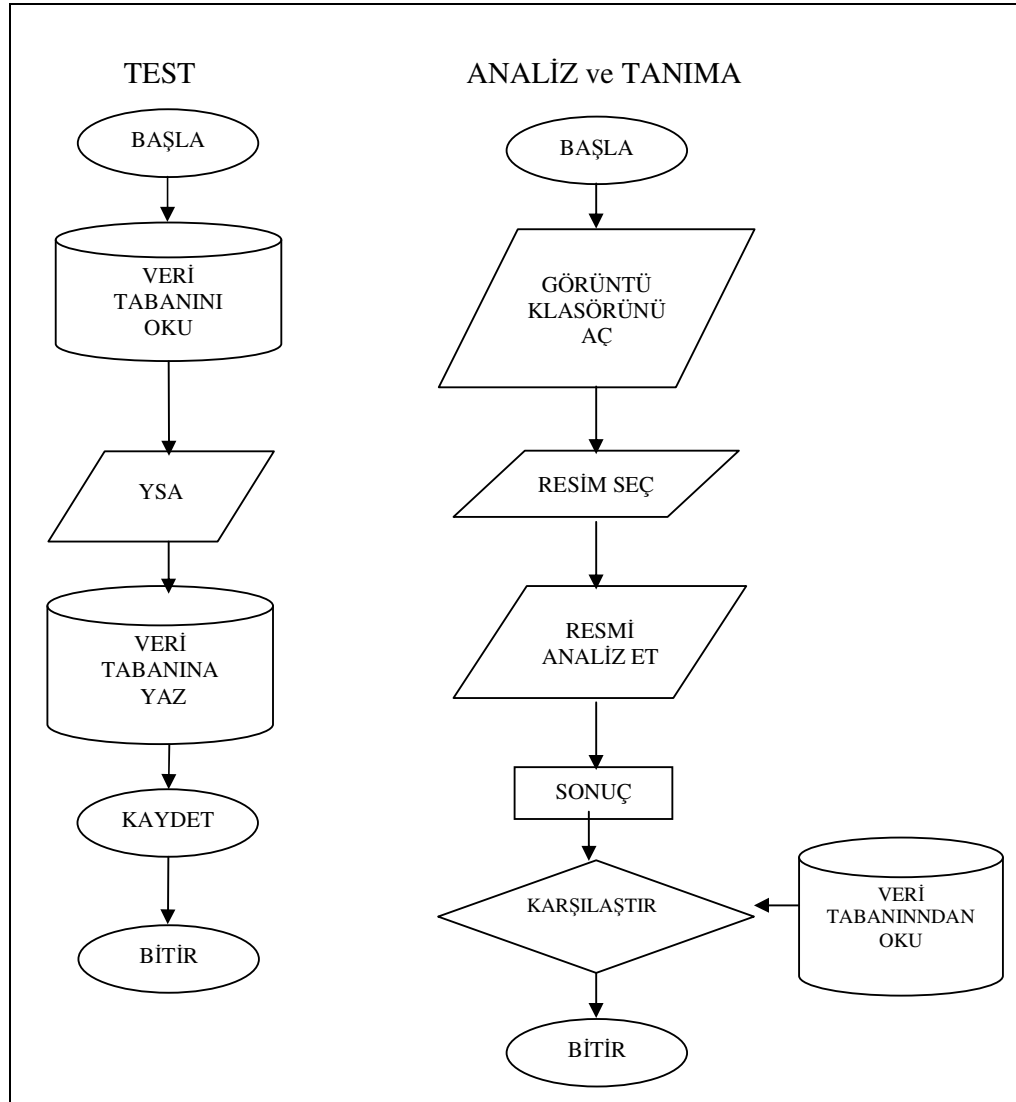
Yapay sinir ağı uygulamasında kullanılan nöronların sayısını arttırarak tekrar denediğimizde sistemimizin çok fazla yavaşladığı görülmektedir. Elde edilen bu bilgileri rahat kullanmak için matlab de kullanıcı arayüzü hazırlanmıştır. Bu arayüz Şekil 6.22'deki gibidir.



Şekil 6.22 Kullanıcı arayüzü

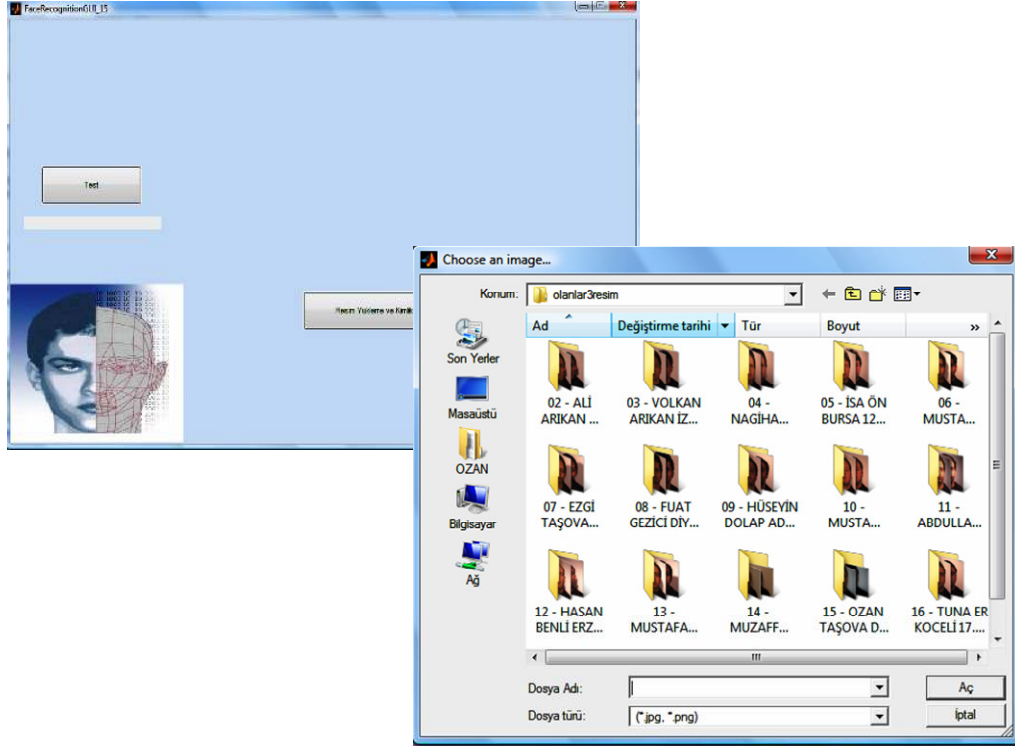
Tasarlanan arayüzde iki tane işlevsel buton bulunmaktadır; bunlardan birincisi sistemin performansını gösteren buton ikincisi sisteme resim yüklemek ve tanıma işlemini başlatan buton dur.

Geliştirilen uygulamaya hazırlanan kullanıcı arayüzün akış diyagramı Şekil6.23’de verilmiştir.



Şekil 6.23 Akış diyagramı

İlk olarak sistemin performansı ölçüldükten sonra sistemin yükleme butonuna basılır ve Şekil 6.24'deki açılan resim dosyasından tanıma yapılacak resim seçilir.



Şekil 6.24 Resim seçme arayüzü

Tanıma yapılacak kişi resim seçme arayüzünden seçildikten sonra sistem otomatik olarak seçilen resime en uygun olan kişiyi veri tabanından tarayarak seçilen resmin yanına getirir. Tanınan kişinin kişi bilgileri çıkan resmin üst kısmında yer alır. Doğru tanımanın yapıldığı bir örnek Şekil 6.25'de gösterilmiştir



Şekil 6.25 Doğru tanıma yapılan örnek

Şekil 6.25’de doğru olarak tanıma yapılan aynı kişinin iki farklı resminin öznitelikleri aşağıdaki Tablo 6.7’de gösterilmiştir.

Tablo 6.7 Öznitelik değerleri; a=uzunluk oranı b= açısı oranı

Resim	L1	L2	L3	L4	L5	L6	L7	L8	L9
1	2,2	2,153	2,149	2,187	1,16	1,005	1,346	1,375	0,866
2	2,109	2,132	2,176	2,047	1,2	0,981	1,377	1,376	0,898

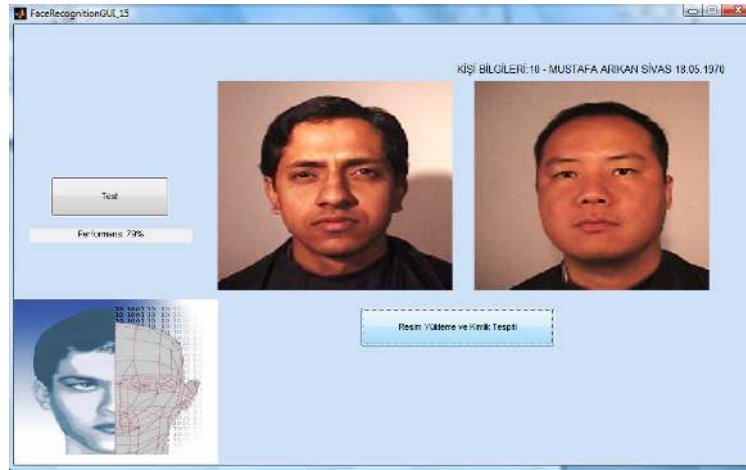
a

Resim	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11
1	1,621	1	0,983	0,502	0,801	1,045	0,493	0,804	0,823	1,06	0,81
2	1,607	1	0,999	0,522	0,841	1,066	0,501	0,8	0,767	1,01	0,84

b

Tablo 6.7 incelendiğinde iki resmin öznitelik değerlerinin bir birine çok yakın olduğu gözlemlenmiştir. Bunun sonucu olarak tanıma işlemi başarılı şekilde sonuçlanmıştır.

Her ne kadar sistemin hatasız bir şekilde çalışmasını istesek de bazı sebeplerden dolayı sistemimizin doğruluk performansı yüzde yüz olmadığından yapılan tanıma ve sınıflandırma aşamasında bazı hatalar veya eşleştirmeler olmaktadır. Şekil 6.26’de buna örnek gösterilmiştir.



Şekil 6.26 Örnek yanlış eşleştirme

Şekil 6.26’deki yanlış eşleştirmeyi irdelememiz gerekir. Bu irdeleme sonucunda iki resmin öznitelik değerleri aşağıdaki Tablo 6.8’da gösterildiği gibidir.

Tablo 6.8 Öznitelik değerleri; a=uzunluk oranı b= açı oranı

Resim	L1	L2	L3	L4	L5	L6	L7	L8	L9
1	2,738	2,142	2,375	2,247	1,05	0,7727	1,219	1,5333	0,875
2	2,449	2,125	2,229	2,204	1,08	0,9783	1,225	1,4918	0,888

a

Resim	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11
1	1,565	1	0,820	0,416	0,885	1,031	0,491	0,789	0,76	0,99	0,68
2	1,597	1	0,892	0,457	0,886	1,083	0,519	0,819	0,782	1,021	0,74

b

Tablo 6.8 incelendiğinde iki resmin öznitelik değerleri bir birine çok yakın olduğu gözlemlenmiştir. Eşleştirmenin yanlış oluşmasının sebepleri arasında bu değerlerin çok yakın olmasından dolayı kaynaklandığı düşünülmektedir.

BÖLÜM YEDİ

SONUÇLAR VE DEĞERLENDİRME

Bu tez çalışmasında yapay sinir ağları ile yüz tanıma üstünde durulmuştur. İnternet kaynaklı ve gerçek kişilerden elde edilen resimlerin içindeki insan yüzü ten tanıma yöntemi ile başarılı şekilde bulunmuştur. Daha sonra bulunan yüz resminin eğimi alınıp incelenmiştir. Grafiklere göre insan yüzünde çok sayıda belirgin noktaların var olduğu tespit edilmiştir. Bu tespit edilen noktalardan yapay sinir ağını beslemek için öznitelik değerleri çıkarılmıştır. Çıkarılan öznitelik değerlerinin her insanda farklılık gösterdiği tespit edilmiştir.

Bu tez için iki farklı veri tabanı hazırlanmıştır. Bu iki farklı veri tabanındaki öznitelik değerleri yapay sinir ağına uygun girdiler ve ağırlıklarla eğitim işlemi tamamlanmıştır. Bu işlem sonucunda 10 kişilik veri tabanının performans oranı en yüksek yüzde 98, 15 kişilik veri tabanının performansı en yüksek yüzde 79 olarak elde edilmiştir. Bu değerler girişteki girdi eleman sayısı artırılarak denendiğinde sistemin doğruluk performansının düştüğü hem de sistemin yavaşladığı gözlemlenmiştir.

Literatürde yer alan benzer çalışmaların başarı sonuçları %53,57 - % 97 aralığında olduğu Tablo 1.1 (Giriş bölümünde) tespit edilmiştir. Bizim elde ettiğimiz performans değerleri bu aralık içinde olduğundan doğru tanıma yüzdelerimizin yeterli olduğu sonucuna varılmıştır.

İleride yapılacak çalışmalarda, gerçek kişilerden daha fazla sayıda resim, daha iyi standartlar içinde resimlerin oluşturulması, yapay sinir ağına daha uygun ağırlık değeri bulunması, değişik eğitim algoritmaları ve aktivasyon fonksiyonlarının kullanılması ve sistem içinde kullanılan yöntemlerin yerine daha başka yöntemler kullanıldığı zaman sistemin performansı ve başarısında artan sonuçlar üreteceği düşünülmektedir.

KAYNAKLAR

- Arı, İ. & Akarun, L. (2009). *Yüz Özniteliklerinin Takibi ve İşaret Dili için İfade Tanıma*. İstanbul: Boğaziçi Üniversitesi Bilgisayar Mühendisliği Bölümü.
- Arı,İ., Alsaran, O. & Akarun, L. (2011). *Görü tabanlı Gerçek zamanlı Duygu Tanıma*. İstanbul: Boğaziçi Üniversitesi Bilgisayar Mühendisliği Bölümü.
- Baltacı, Ö (2011). *Yapay sinir ağları ve parmak izi analizi yöntemi ile kimlik tayini*. İzmir: Dokuz Eylül Üniversitesi Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi.
- Bilgin, M. (2008). *Biyometrik Seçim Sistemi Tasarımı Ve Gerçekleştirilmesi*. Konya: Selçuk Üniversitesi Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi.
- Bolat, B. (2005). *Recognition of the REED Instrument Sounds by Using Statistical Neural Networks*. Journal of Engineering and Natural Sciences İstanbul: Yıldız Teknik Ünı.
- Çelik, M. & Oral, U. (2003). *Motorlu Araç Plaka Görüntülerinden Karakter Ayırıştırma ve Tanıma*. IJCI Proceedings of International Conference on Signal Processing, ISSN 2003, 1304-2386.
- Egm Güvenlik (2010). *Ses Karakteristiği*. 25 Eylül 2010, <http://www.egmguvenlik.net/ansiklopedi-ses-karakteristigi.html>
- Ekşi, Ö. (2010). *Yapay Sinir Ağları*. 24 Ocak 2010, http://www.omereksi.com/?p=30#_ftnref5
- Elmas, Ç. (2003). *Yapay Sinir Ağları*. Ankara: Seçkin Yayıncılık.

- Eski, S. (2008). *Görüntü İşleme Yöntemleri ile Araç Marka ve Türlerinin Tanınması*. İstanbul: İstanbul Teknik Üniversitesi Elektrik-Elektronik Fakültesi, Lisans Tezi.
- Gökberk, B., İrfanoğlu, M.O., Doğu, H., Akarun, L. & Alpaydın, E. (2003). *Yüz Tanıma İçin Eniyi Gabor Parametrelerinin Seçimi*. İstanbul: Boğaziçi Üniv. Bilgisayar Mühendisliği
- Hopfield, J.J. (1982). *Neural networks and physical systems with emergent collective computational abilities*. Proc.Nat.Acad.Sci 79:2554-2558.
- Kohonen, T. (1984). *Self-Organization and Associative Memory*, Berlin: Springer – Verlag,
- Mesleki Eğitim ve Öğretim Sisteminin Güçlendirilmesi Projesi (2007). *Elektrik Elektronik Teknolojisi Geçiş Kontrol Sistemlerinin Bağlantıları*.
- Moenssens, A. (1971). *Fingerprint Techniques*. London: Chilton Book Company, 321
- Özpınar, A. (bt). *Rfid Ders Notları*. İstanbul Ticaret Üniversitesi Bilgisayar Mühendisliği Bölümü . 1 (1), 23-24.
- Rosenblatt, F. (1959). *Principles of Neuradynamics*, New York: Spartan Books.
- Rosenblatt, F. (1962). *Principles of Neuradynamics: Perceptrons and The Theory of Brain Mechanisms*. Washington D.C.: Spartan Boks.
- Rowley, H.A, Baluja, S. & Kanade T. (1998). *Neural Network - Based Face Detection*. PAMI, IEEE.

- Rumelhart, D.E., McClelland, J.L. & PDP Research Group (1986). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Cambridge, MA.: v. 1:Foundations, MIT Pres.
- Saday, T. & Akhan, N. (bt). *Bilgisayar Destekli Kimlik Tespit Sistemlerinde Biometrik Yöntemlerin Değerlendirilmesi*. Konya: Selçuk Üniversitesi, Mühendislik Mimarlık Fakültesi, Bilgisayar Mühendisliği Bölümü.
- Sağiroğlu, Ş. & Özkaya, N. (2006). Otomatik Parmakizi Tanıma Sistemlerinde Kullanılan Önişlemler İçin Yeni Yaklaşımlar. Ankara: *Gazi Üniv. Müh.Mim.Fak. Der. Cilt 21, No 1, 11- 19, 2006*.
- Sarı, M.A. (bt). *Biometrik Tanımlama Sistemleri*. bt, <http://www.gulnarcity.com/index.asp?sayfa=yazioku&id=49>
- Sönmez, E.B., Özbek, N.Ö. & Özbek, Ö. (2007). *Avuç İzi ve Parmak İzine Dayalı Bir Biyometrik Tanıma Sistemi*. İstanbul: İstanbul Bilgi Üniversitesi Bilgisayar Bilimleri.
- Subaş, H. (2010). *Yapay Sinir Ağı İle Atıksu Arıtma Performansının Modellenmesi*. Adana: Çukurova Üniversitesi Fen Bilimleri Enstitüsü Yüksek Lisans Tezi.
- Şamlı, R. (2010). *Biyometrik Güvenlik Sistemleri*. İstanbul: *İ.Ü. Bilgisayar Mühendisliği Bilişim dergi sayı 21*.
- Şamlı, R. & Yüksel, M.E.(2009). *Biyometrik Güvenlik Sistemleri. Akademik Bilişim'09 - XI. Akademik Bilişim Konferansı Bildirileri Harran Üniversitesi, Şanlıurfa*.
- Şen, Z. (2004). “*Yapay Sinir Ağları ilkeleri*”, İstanbul: Su Vakfı Sayfa: 18.

Tolon, M. & Tosunoğlu, N.G. (bt). *Tüketici Tatmini Verilerinin Analizi: Yapay Sinir Ağları ve Regresyon Analizi Karşılaştırması*. Ankara: Gazi Üniversitesi.

Tüzün, S. & Akan, A. (bt). *Yüz Uzayının Dikleştirilmesine Dayanan Yeni Bir Yüz Tanıma Yöntemi*. İstanbul: İstanbul Üniversitesi Mühendislik Fakültesi Elektrik-Elektronik Mühendisliği Bölümü.