

50956

T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YAPAY SİNİR AĞLARI İLE NESNE TANIMA

İbrahim TÜRKOĞLU

YÜKSEK LİSANS TEZİ

ELEKTRİK-ELEKTRONİK ANABİLİM DALI

Bu Tez, Tarihinde, Aşağıda Belirtilen Jüri Tarafından Oybirliği /
Oyçokluğu İle Başarılı / Başarısız Olarak Değerlendirilmiştir.

Danışman

Yrd.Doç.Dr. Ahmet ARSLAN

ÖZET

Yüksek Lisans Tezi

YAPAY SİNİR AĞLARI İLE NESNE TANIMA

İbrahim TÜRKOĞLU

Fırat Üniversitesi

Fen Bilimleri Enstitüsü

Elektrik-Elektronik Mühendisliği Anabilim Dalı

1996, Sayfa: 93

Son zamanlarda, yapay sinir ağları ile nesne tanıma üzerine oldukça yoğun çalışmalar yapılmaktadır. Bunun sebebi; karmaşık ve belirsiz veriler ile nesne tanıma problemlerinin çözümünde, yapay sinir ağlarının öğrenerek karar verme prensibinin başarılı olmasıdır. Yapay sinir ağları kendilerine gösterilen bir nesneyi daha önce öğrendikleri ile mukayese ederek aradaki benzerlikleri bulup, nesnelere belirli sınıflara ayırma özelliğine sahiptir. Temelde nesne tanıma işlemi, nesnelere sınıflandırılmasından farklı bir işlem değildir.

Bu tezde; temel nesne tanıma yaklaşımları incelenmiş, eğitici ve eğitici yapay sinir ağı modelleri kullanılarak bozuk (eksik veya gürültülü) nesnelere tanınması gerçekleştirilmiş ve ayrıca dönmeden ve konum değişiminden bağımsız bir nesne tanıma sistemi geliştirilmiştir. Geliştirilen sistem, nesnenin öznitelik vektörünün çeşitli ön işlemlerden sonra çıkarılması ve bu vektöre göre nesneyi sınıflandırarak tanımlama, aşamaları üzerine kurulmuştur. Nesnelere öznitelik vektörü hem yerel ve hem de genel yöntemlerle bulunmuş ve yapay sinir ağı modelleri ile sınıflandırılarak başarılı sonuçlar elde edilmiştir.

ANAHTAR KELİMELER : Nesne Tanıma, Yapay Sinir Ağları, Örüntü Sınıflandırma, Özellik Çıkarma.

SUMMARY

Masters Thesis

OBJECT RECOGNITION USING ARTIFICIAL NEURAL NETWORKS

İbrahim TÜRKOĞLU

Firat University

Graduate School of Natural and Applied Sciences

Department of Electrical and Electronics Engineering

1996, Page: 93

In recently, the studies on the object recognition by using artificial neural networks are very increased. In the solution of object recognition problems under complicated and fuzzy data, the success of the decision principle of the artificial neural networks by learning has been effective on these studies. Artificial neural networks has the peculiarity that compares given objects with the objects which it has learned before and finds the similarities between them to classify the objects into certain groups. The object recognition is not a different process from the objects classification in basic.

In this thesis; basic object recognition approaches are examined and the recognition of the distorted (incomplete or noisy) objects is implemented by using supervised and unsupervised artificial neural networks models. In addition, an object recognition system, regardless rotation and position variation is developed. The developed system is set up on the derivation of object's feature vector after some preprocesses and the steps of object recognition by classifying and according to the obtained vector. The object's feature vector extraction by using both local and global methods are classified with the artificial neural networks and successful results are obtained.

KEY WORDS : Object Recognition, Artificial Neural Networks, Pattern Classification, Feature Extraction.

TEŞEKKÜR

Bu tez çalışmam boyunca, ilgi ve yardımlarını esirgemeyen danışmanım Sayın Yrd.Doç.Dr. Ahmet ARSLAN'a, çalışmam esnasında değerli fikir ve yardımlarından yararlandığım Yrd.Doç.Dr. Bekir KARLIK'a, Öğr.Gör. Muammer GÖKBULUT'a, ve tezin düzenlenmesindeki yardımlarından ötürü Arş.Gör. Zafer AYDOĞMUŞ'a teşekkürlerimi ve şükranlarımı sunarım.

İÇİNDEKİLER

	<u>Sayfa</u>
ÖZET	II
SUMMARY	III
TEŞEKKÜR	IV
İÇİNDEKİLER	V
ŞEKİLLER LİSTESİ	VIII
TABLolar LİSTESİ	X
SİMGELER LİSTESİ	XI
1. GİRİŞ	1
2. NESNE TANIMA	7
2.1. Robot Görmesi	7
2.2. Bilgisayar Görmesi	9
2.3. Nesne Tanıma	9
2.4. Boyutsal Nesne Tanıma Sistemleri	10
2.4.1. Veri tabanları, özellik çıkarımları ve sınıflandırma	11
2.4.1.1. Veri tabanları	11
2.4.1.2. Özellikler	12
2.4.1.3. Sınıflandırma	13
2.4.2. İki boyutlu nesne tanıma	13
2.4.2.1. Genel özellik çıkarım yöntemi	15
2.4.2.2. Yapısal özellik çıkarım yöntemi	15
2.4.2.3. İlişkisel grafik yöntemi	16
2.4.3. 21/2 boyutlu nesne tanıma	16
2.4.4. Üç boyutlu nesne tanıma	17
2.4.4.1. Tam nesne tanıma	18
2.4.4.2. Çok görünümlü nesne tanıma	18
2.5. Nesne Tanıma Yaklaşımları	19
2.5.1. İstatistiksel yaklaşım	20
2.5.2. Yapısal yaklaşım	20
2.5.3. Yapay sinir ağları yaklaşımı	21

2.6. Nesne Tanımda Karşılaşılan Problemler	22
3. İKİ BOYUTLU NESNE TANIMA SİSTEMİ	24
3.1. Öznitelik Vektörünün Elde Edilmesi	24
3.1.1. Eşikleme	24
3.1.2. Kenar çıkarma	25
3.1.3. Öznitelik vektörleri	26
3.1.3.1. Otoregresif model	27
3.1.3.2. Poligonal yaklaşım	28
3.1.3.3. Zincir kodları	29
3.1.3.4. Yapısal nesne tanıma modelleri	31
3.2. Sınıflandırıcılar	32
3.2.1. Klasik sınıflandırıcılar	33
3.2.1.1. En yakın komşu yöntemi	33
3.2.2. Bulanık (fuzzy) sınıflandırıcılar	35
3.2.3. Yapay sinir ağları sınıflandırıcıları	36
3.2.3.1. Yapay sinir ağının tanımı ve modeli	36
3.2.3.2. Biyolojik sinir ağları ve yapay sinir ağları arasındaki ilişki	37
3.2.3.3. Yapay sinir ağlarının yapıları	40
3.2.3.4. Yapay sinir ağlarında öğrenme ve öğrenme algoritmaları	42
3.2.3.5. Yapay sinir ağlarının temel özellikleri	46
4. YAPAY SİNİR AĞLARI İLE İKİ BOYUTLU NESNE TANIMA İÇİN GELİŞTİRİLEN SİSTEM VE UYGULAMALARI	49
4.1. Öznitelik Vektörünün Çıkarılması	49
4.1.1. Kenar çıkarma	50
4.1.2. Kenar izleme	54
4.1.3. Öznitelik vektörleri	59
4.1.3.1. İstatistiksel yaklaşımla öznitelik vektörünün elde edilmesi	59

4.1.3.2. Geometriksel yaklaşımla öznitelik vektörünün elde edilmesi	60
4.2. Gerçekleştirilen Yapay Sinir Ağları ile Sınıflandırma	63
4.2.1. Kohonen'in kendi kendini düzenleyen öznitelik haritaları sınıflandırıcısı	64
4.2.2. Geriye yayılım ağı sınıflandırıcısı	74
5. SONUÇ VE DEĞERLENDİRME	87
6. KAYNAKLAR	90



ŞEKİLLER LİSTESİ

	<u>Sayfa</u>
Şekil-1.1. Bir nesne tanıma sisteminin blok diyagramı	4
Şekil-1.2. Öğrenme işleminide gerçekleştiren nesne tanıma sisteminin blok diyagramı	5
Şekil-2.1. Robot görmesi ile üretim hattında çalışan robot kolu	8
Şekil-2.2. Veri tabanlı bir nesne tanıma sistemi	11
Şekil-2.3. Görme kontrollü bir taşıyıcı bant	14
Şekil-2.4. Genel nesne tanıma yaklaşımı sistemi	19
Şekil-2.5. İstatistiksel nesne tanıma yaklaşımı sistemi	20
Şekil-2.6. Yapısal nesne tanıma yaklaşımı sistemi	21
Şekil-3.1. Eşikleme işlemi	25
Şekil-3.2. Görülen nesnenin merkeze göre kenar uzunluklarının değişimi	27
Şekil-3.3. Poligonal yaklaşım ile nesnenin kritik noktalarının bulunması	28
Şekil-3.4.a. Dörtlü zincir kod	29
Şekil-3.4.b. Sekizli zincir kod	29
Şekil-3.5. Verilen sayısal görüntünün sınırlarının dörtlü zincir kod gösterimi	30
Şekil-3.6. Verilen nesnenin sınırlarının sekizli zincir kod ile kodlanması	30
Şekil-3.7. T alfabetik karakter örüntüsünün yapısal modeli	32
Şekil-3.8. Veri tabanındaki sınıflar ve en yakın komşu yöntemine göre k vektörünün sınıflandırılması	34
Şekil-3.9. Biyolojik nöronun şematik yapısı	38
Şekil-3.10. Yapay sinir hücresinin modeli	39
Şekil-3.11. Eşik fonksiyonları	40
Şekil-3.12.a. İleri doğru tek yönlü yapay sinir ağı modeli	41
Şekil-3.12.b. Çift yönlü tamamen bağlantılı yapay sinir ağı modeli	41
Şekil-3.13. Yapay sinir ağı sınıflandırıcıları	42
Şekil-4.1.a. n_5 nesne noktasının komşuları	50
Şekil-4.1.b. Geliştirilen kenar çıkarma şablonu	50
Şekil-4.2. Kenar çıkarma işlemi uygulanacak nesnelere	52
Şekil-4.3. Kenar çıkarma işleminden sonra nesne görüntüleri	53
Şekil-4.4. Matris veri tabanında kenar izleme algoritması için yönler	55

Şekil-4.5. Herhangi bir kenar noktasının komşularını izleme sıralaması	56
Şekil-4.6. Herhangi bir (x,y) noktasının komşularının konumları	56
Şekil-4.7. Kenar izleme işlemi başarılı olmuş bazı nesnelere	58
Şekil-4.8. A alfabetik karakter örüntüsünün veri tabanı üzerinde gösterimi	59
Şekil-4.9. Dikdörtgen bir nesnenin veri tabanı üzerindeki görünümü	60
Şekil-4.10. Kare nesnesinin matris veri tabanı üzerinde gösterimi	61
Şekil-4.11. Kare nesnesinin merkez-kenar değişimi (a,b)	62
Şekil-4.12. A alfabetik karakter örüntüsünün farklı görünüşleri (a,b) ve merkez-kenar grafiği (c)	63
Şekil-4.13. Kohonen'in kendi kendini düzenleme haritasının ağ birimleri	65
Şekil-4.14. Kohonen sistem modeli	65
Şekil-4.15. Meksika şapkası fonksiyonu	66
Şekil-4.16. Bir işlem elemanının yapısal komşuluğu	67
Şekil-4.17. Kohonen yapay sinir ağını eğitmek ve nesne tanımak için geliştirilen programın akış şeması	69
Şekil-4.18. Alfabetik karakter örüntülerinin Kohonen yapay sinir ağı tarafından istatistiksel olarak tanınma yüzdeleri	71
Şekil-4.19.a. Alfabetik karakter örüntülerinin Kohonen yapay sinir ağı tarafından geometriksel olarak tanınma yüzdeleri	72
Şekil-4.19.b. Geometrik nesnelere Kohonen yapay sinir ağı tarafından geometriksel olarak tanınma yüzdeleri	73
Şekil-4.20. Çok katmanlı yapay sinir ağı	74
Şekil-4.21. Hata geri yayılım algoritmasını kullanan yapay sinir ağının blok şeması ...	76
Şekil-4.22. Hata geriye yayılma algoritmasını kullanan yapay sinir ağının eğitime akış şeması	78
Şekil-4.23. Geriye yayılım ağının eğitime esnasında sistem hatasının değişimi	80
Şekil-4.24. Yapay sinir ağında nesne tanıma işleminin akış şeması	80
Şekil-4.25. Alfabetik karakter örüntülerinin geriye yayılım ağı tarafından istatistiksel olarak tanınma yüzdeleri	81

Şekil-4.26. Geriye yayınım ağının eğitme esnasında sistem hatasının değişimi	82
Şekil-4.27. Alfabetik karakter örüntülerinin geriye yayınım ağı tarafından geometriksel olarak tanınma yüzdeleri	83
Şekil-4.28. Geriye yayınım ağının eğitme esnasında sistem hatasının değişimi	84
Şekil-4.29. Geometriksel nesnelerin geriye yayınım ağı tarafından geometriksel olarak tanınma yüzdeleri	85

TABLULAR LİSTESİ

Sayfa

Tablo-3.1. T harf örüntüsünün açı / uzunluk yöntemine göre yapısal özellikleri	32
--	----

SİMGELER LİSTESİ

- d_i : İstenen nöron çıkışı
 ε : Öğrenme katsayısı
 α : Momentum katsayısı
Net : Giriş nesne öznelik vektörü ile ağırlık vektörünün çarpımının toplamı
 N_c : Komşuluk kümesi
 X : Giriş nesne öznelik vektörü
 W : Ağırlık vektörü,
 $f(x,y)$: Nesnenin herhangi bir noktası
 $f(\text{net})$: Hesaplanan nöron çıkışı
 $\text{uzk}()$: İki küme merkezi arasındaki öklit uzaklığı
 $E(t)$: Geriye yayılım aşında hata fonksiyonu
 (x_a, y_a) : Nesnenin ağırlık merkez koordinatı
- 2-B : İki Boyut
3-B : Üç Boyut
BDT : Bilgisayar Destekli Tasarım
YSA : Yapay Sinir Ağı

BÖLÜM 1

1.GİRİŞ

Teknolojideki başdöndürücü gelişmeler, son yıllarda bilim adamlarının insana ve insan davranışlarına olan ilgisini artırmakta, aynı zamanda da düşüncelerini uygulamaya dökme imkanı tanımaktadır. Sinir ağları, insan ve yüksek sınıftan diğer canlılarda bulunan yaşamsal mekanizmanın temelidir. Düşünme gibi sadece insan türünde bulunan yeteneklerin dışında sinir ağları, canlının fiziksel özelliklerini, kimyasal dengesini, savunma mekanizmasını v.s.'de oluşturur. Bununla beraber böylesine yoğun ve karmaşık kontrol işlemleri yürüten sinirlerin iç yapısına baktığımızda, günümüzdeki yarı iletken elektronik devre elemanlarından çok daha yavaş bir nöron hücresi görürüz. İşte bilim adamları son on yıl içerisinde, bu basit ve nispeten yavaş hüreciklerin bir araya geldiklerinde nasıl olup da dünyanın en hızlı bilgisayarından daha hızlı işlem yapabildiklerini araştırmaktadırlar. Araştırmalar insanları nöronların basit modellerini yapıp, bu modeller ile oluşturdukları devrelere insansı bazı işlemleri yaptırma çalışmalarına götürmüştür. Nöronun modellenmesi, ele alınan örneklere bağlı olarak çok farklı şekillerde olabilmektedir. Örneğin, öğrenme yeteneği olan bir nöron yanında, öğrenme yeteneği olmayan fakat çok hızlı cevap verebilen bir nöron modeli yapmak da mümkündür. Yapay sinir ağları çalışmalarının yoğunlaştığı iki büyük alandan bir tanesi öğrenme, diğeri ise tanımadır. Öğrenmede devrenin bilgi saklama kapasitesi en önemli rolü oynarken, tanımda giriş verilerindeki bozulmalara karşı dayanıklılık, ortamdan bağımsızlık ve hız daha fazla önem kazanmaktadır.

Yapay Sinir Ağları (YSA) insan beyninin çalışma sisteminin yapay olarak benzetimi çabalarının bir sonucu olarak ortaya çıkmıştır. En genel anlamda bir yapay sinir ağı, insan beynindeki bir çok nöronun ya da basit işlemcilerin birbirlerine değişik etki seviyeleriyle bağlanması ile oluşan karmaşık bir sistem olarak düşünülebilir. Önceleri tıp bilminde, insan beynindeki nöronların matematiksel modelleme çabaları ile başlayan çalışmalar şu sıralarda hızlanmış ve çok farklı bilim dallarında araştırma konusu haline gelmiştir. Genel olarak yapay sinir ağlarının pratik kullanımı, çok farklı yapıda ve biçimlerde bulunabilen

iletişim verilerini hızlı bir şekilde tanımlama ve algılama düzenidir. Aslında mühendislik uygulamalarında yapay sinir ağlarının geniş çaplı kullanımının en önemli nedeni, klasik tekniklerle çözümü zor problemler için etkin bir alternatif oluşturmasıdır.

Tarihsel gelişimi içinde hızlı hesaplama yönelik ilk yapay sinir ağı çalışmaları 1950’li yıllarda başlamış ve basit nöron modellerine dayalı bir hesaplama tekniği Rosenblatt tarafından önerilmiştir. 1960’lı yıllarda Widrow ve Hoff, bu basit nöron modellerini kullanarak ilk öğrenebilen adaptif sistemler üzerinde çalışmışlardır. Ancak 1969’da Minsky ve Papert yayınladıkları “Perceptron” (algılayıcı) adlı bir kitapta, yapay sinir ağları yardımıyla öğrenmede ve hesaplamada aşılması zor engel olduğunu iddia etmeleri yapay sinir ağları konusundaki çalışmaları yavaşlatmıştır. Günümüz çalışmaları, 1982’de Hopfield tarafından yayınlanan “Neural Networks and Physical Systems” adlı çalışma ile başlar. Bu çalışmada Hopfield nöronların karşılıklı etkileşimlerine dayanan nöral hesaplama modeli önermiştir. Model, bir hata fonksiyonunu alabileceği minimum değerine indiren, birinci mertebe doğrusal olmayan diferansiyel denklemlerden oluşmuştur. Hopfield, ağ seviyesinde, nöron seviyesinde var olmayan hesaplama kapasitesinin bulunduğunu öne sürmektedir. Buna, Hopfield ağları denir. Hopfield ’ın yapay sinir ağı modeli, geri beslemeli sistemi ve bunun kullanılabilirliğini göstermesi sebebiyle yapay sinir ağları konusundaki çalışmaları hızlandırmıştır. 1976’da Grosberg ’in ortaya koyduğu ART (Adaptive Resonans Theory) modeli çok gelişmiş bir yapay sinir ağıdır. Henüz çok fazla sayıda probleme uygulanmamıştır. Aynı yıllar da Kohonen de, kendi kendini düzenleyen öznitelik haritalarını (self-organization mapping) geliştiriyordu. Rumelhart ve arkadaşlarının 1986’da “Parallel Distributed Processing” gurubu geri beslemeli modellerde yeni öğrenme yöntemi olan, hatanın geriye yayılması algoritmasını (Back-Propagation Algorithm) geliştirerek bu konuda daha önce iddia edilen aksaklıkların aşılabileceğini göstermiştir. Bugün birçok yapay sinir ağı, bunun değişik varyasyonlarını kullanmaktadır. Örnek olarak, ADALINE (Adaptive Linear Element) ve MADALINE (Multi ADALINE) yapay sinir ağı çalışmaları verilebilir. Günümüzde yapay sinir ağlarının teorik çalışmaları tamamlanmış olup uygulamaya yönelik çalışmaları devam etmektedir (Zurada,1992).

Yapay sinir ağlarının, mühendislik başta olmak üzere birçok bilimsel alanda, karmaşık ve belirsiz verilere sahip problemlere çözüm ürettikleri ispatlanmıştır. Bu

teknoloji, bilgisayar dünyasına, insan beyninin ve sinir sisteminin davranışlarını taklit etme esası üzerine kurulmuş, yeni bir bilgi işleme yaklaşımı getirmiştir. Algoritmasız, tamamıyla paralel, adaptif, öğrenebilen ve paralel dağıtılmış bir hafızaya sahip olma, bu sistemlerin temel özellikleridir. Bu teknolojinin, insana benzer robotların oluşturulmasına temel teşkil edecek çalışmaların başlamasına ve ilk adımlarının atılmasına sebep olacağı umulmaktadır.

Bir robotun çevre ile iletişimini sağlayan en önemli duyum elemanlarından biri görmedir. Robotun çevresindeki nesnelere tanınması, bu nesnelere hakkındaki belirgin ilişkiyi kavraması ve duruma uygun cevap vermesi, pek çok robotik çalışmaların temelini teşkil etmektedir. Robotun görmesi, özellikle endüstriyel uygulamalarda, tehlikeli ortamların incelenmesinde ve tıp araştırmalarında etkilidir. Ancak başarılı robot uygulamalarına en büyük engellerden biri, görsel duyu eksikliğidir. Üretimde görsel duyu, nesnelere (malzeme, eşya ... v.s.) tanınması, denetlenmesi, montajı ve kullanılması için gereklidir. Montaj ve malzemelerin kullanımı işlemlerinde, parçaların tanınması, pozisyon ve yönelimlerin (oryantasyonların) tam olarak bilinmesi gereklidir. Bir robot, şayet parça doğru pozisyonda ve oryantasyonda ise görsel duyu kullanmadan parçaya dokunabilir. Fakat parçalar rastgele yerleşmiş ise, görme gereklidir. Görme sayesinde, robotun nesneyi kavrayışı daha doğru olur ve çalışma hızı artar. Böylece insan tarafından doğrulukla yerine getirilemeyen monoton işler ile çok pahalı muayene işleri, otomatik hale getirilebilir. Bu amaçla nesne tanıma çalışmaları bu monoton işleri daha kaliteli ve daha düşük maliyetli hale getirmeye yöneliktir (Nof,1985).

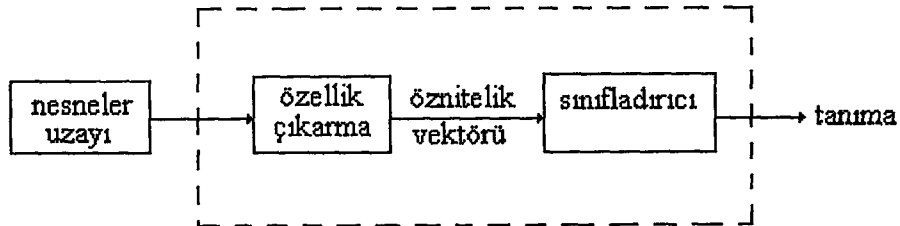
Son yıllarda büyük gelişme gösteren nesne (şekil, görüntü,...v.s.) tanıma veya nesnelere sayısal hali olan örüntüleri tanıma (pattern recognition) ve öğrenme (learning) tekniği birçok dalda önemli uygulama alanları bulmuştur. Bu dalların başlıcaları, harf tanıma, robotik görme, elektrokardiyografi (EKG) ve benzeri işaretlerin analizleri, istatistiksel haberleşme sistemi v.b. problemlerdir.

Nesne tanıma; aralarında ortak özellik bulunan ve aralarında bir ilişki kurulabilen şekilleri veya nesnelere bazı tesbit edilmiş özellikler veya karakterler vasıtası ile tanımlama veya sınıflandırma olayıdır. Ortak özelliklere sahip nesne ve cisimlerin oluşturduğu topluluğa nesne sınıfları ismi verilir. Örneğin Türk alfabesindeki harflerin tanınması probleminde, her harf bir nesne sınıfı oluşturacağından, toplam 29 nesne sınıfına gerek

vardır. Diğer taraftan Çin alfabesindeki harflerle Türk harflerinin ayrılması düşünülüyorsa, bu iş için iki nesne sınıfı yeterli olacaktır. Nesne tanımanın en önemli iki amacı;

- Bilinmeyen nesne sınıflarına belirli bir şekil vermek,
- Bilinen bir sınıfa ait olan nesneyi teşhis etmektir.

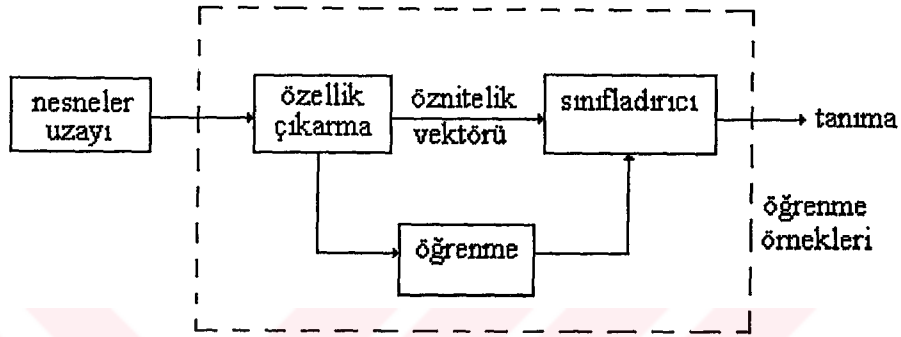
Nesne tanıma sistemleri genellikle iki kısımdan oluşur; birinci kısım, nesnelerin tanınması için gerekli özelliklerin veya ölçülecek büyüklüklerin seçilmesi işlemini gerçekleştirir. Bu işleme, özellik çıkarma (feature extraction) ve bu işlemi gerçekleştiren sisteme, özellik çıkarıcı (feature extractor) adı verilmektedir. Şekil veya nesneleri belirlemede kullanılan her bir özelliğin veya ölçülecek büyüklüğün, ölçüm sonucunu veren gerçel sayıya özellik adı verilmekte ve bileşenleri bu özelliklerden oluşmuş vektöre, özelliklerin alındığı nesneye ilişkin öznitelik vektörü denilmektedir. Pratikte uygun özellik seçimi, bir nesne tanıma probleminin en güç ve en önemli bölümünü oluşturur. Eğer seçilen özellikler nesnelerin bütün özelliklerini tam olarak taşımıyorsa, tanıma ve sınıflandırma işlemlerinde yapılacak hatalar büyük olacaktır. Nesne tanımanın ikinci kısmı ise, elde edilen öznitelik vektörlerinden faydalanarak nesnelerin sınıflandırılmasıdır. Bir nesne tanıma sisteminin blok diyagramı şekil-1.1'de verilmiştir.



Şekil-1.1. Bir nesne tanıma sisteminin blok diyagramı.

Nesne sınıflandırma işlemi, öznitelik vektörlerinden oluşmuş uzayın her bölümü bir nesne sınıfına karşı düşecek şekilde, birbirlerinden ayrık bölümlere ayrılması ve her bölüme ilişkin bir doğrusal ayırım fonksiyonu tanımlanması temeline dayanır. Nesne tanıma modellerinde, nesne sınıflarını ayıran ayırım fonksiyonlarının matematiksel

biçimlerinin bilinmemesi durumunda, önce bir öğrenme işleminin gerçekleştirilmesi gerekir. Öğrenme işlemi, bu ayırım fonksiyonlarının bilinmeyen gerçek değerlerinin, nesne sınıflarının herbirinden alınan örnek nesnelere yardımıyla belirlenmesi ya da öğrenilmesini sağlar. Örnek nesnelere, literatürde öğrenme örnekleri (learning samples) veya eğitim nesnelere (training samples) ismi verilmektedir (Açıkgöz,1993). Öğrenme işlemini içine alan nesne tanıma sisteminin blok diyagramı şekil-1.2’de verilmiştir.



Şekil-1.2. Öğrenme işlemini de gerçekleştiren nesne tanıma sisteminin blok diyagramı.

Nesne tanımada yeni bir yaklaşım olan yapay sinir ağları, olayları öğrenerek karar verme prensibi üzerine kurulduğundan, yapay sinir ağları ile nesne tanıma klasik nesne tanıma modellerine göre daha başarılı sonuçlar verir. Bu başarı, özellikle bu ağ yapılarının programlama yerine verilmiş bir öğrenme kümesi üzerinden öğrenebilmesi sayesinde. Ama bilinmelidir ki, yapay sinir ağının herhangi bir uygulamada kolayca çok başarılı olabilmesi söz konusu değildir; herhangi bir yöntem gibi değişik durumlarda verilmesi gereken kararlar ve bunların getirdiği üstünlük ve kısıtlamalar vardır (Özmetler,1989).

Bu tezde, yapay sinir ağ modelleri ve nesne tanıma yaklaşımları incelenerek, yapay sinir ağ modellerinden yararlanıp, bir nesne tanıma sisteminin bilgisayarda benzetimi yapılacak ve bilgisayar ekranında çizilen bazı nesnelere, değişik yapay sinir ağ modellerine öğretme ve bu yapay sinir ağlarınca öğrenilen nesnelere değişik biçimlerinin tanınarak sınıflandırma işlemleri gerçekleştirilecektir. Bu amaçla yapılan çalışma aşağıdaki bölümlerden meydana gelmektedir.

İkinci bölümde, nesne tanıma ile ilgili olarak, iki ve üç boyutlu nesne tanıma kavramları ve nesne tanıma yaklaşımları incelenmektedir.

Üçüncü bölümde, iki boyutlu nesne tanıma sisteminin temel öğeleri incelenerek her bir sistem öğesi ayrıntılı olarak ele alınmıştır.

Dördüncü bölümde, iki boyutlu nesne tanıma için gerçekleştirilmiş olan sistem sunulmuştur. Bu sistemin iki ana öğesi olan öznitelik vektörünün elde edilmesi ile geriye yayılım öğrenme algoritması ve Kohonen 'in kendi kendini düzenleyen öznitelik haritasını kullanan yapay sinir ağı sınıflandırıcıları tanıtılmış ve bu sınıflandırıcılar ile çeşitli nesne tanıma uygulamaları gerçekleştirilmiştir. Ayrıca yine bu bölümde geliştirilen yeni bir kenar çıkarma şablonunun çeşitli nesnelere uygulaması yapılmıştır. Bunun için gerekli olan yazılımlar, C programlama dilinde geliştirilmiştir.

Son bölümde ise, elde edilen sonuçlar irdelenerek, ileride yapılabilecek benzeri çalışmalar ve uygulama alanları için öneriler sunulmuştur.

BÖLÜM 2

2. NESNE TANIMA

Görme duyusu, insanın sahip olduğu en önemli algılama sistemidir. Öyle ki, insan gözünün çok kısa bir süre için açılıp kapanması esnasında, görme organının çevreyle ilgili olarak insana kazandıracığı bilgi oldukça fazladır. Görme işlemi, kişinin dikkatini çeken görüntülerin dışında, çevre hakkında birçok önemli görüntüyü de yansıtır. Bu ise, yapay görme sistemlerinde, algılama cihazlarının ne kadar karmaşık olacağını gösterir.

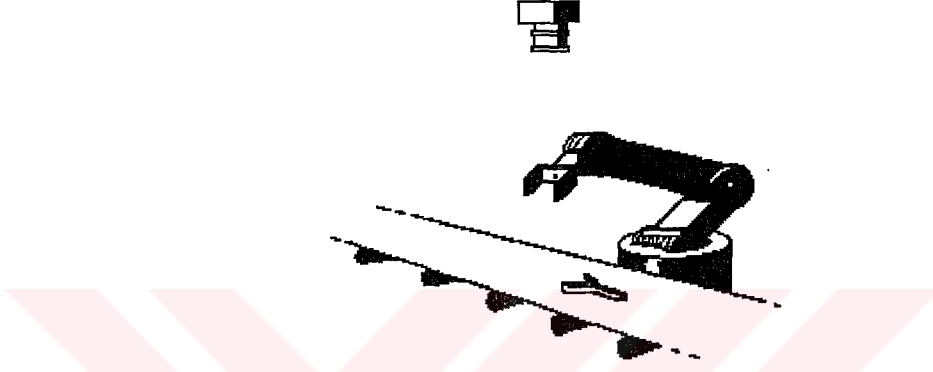
2.1. Robot Görmesi

Robot görmesi, üç boyutlu dünyadan elde edilen görüntülerin verdiği bilgileri yorumlamak ve karakterize etmek şeklinde tanımlanabilir. Robot görmesi, araştırmalarının amacı, fiziksel bir ortamda çalışan robotlar için önceden tahmin edilemeyen olaylar karşısında görsel duyu teknolojisini geliştirmektir. Görme işlemi, robotun icra ettiği pek çok iş için, çevre hakkında bilgi aldığı en önemli kaynaktır. Çevresindeki nesnelere tanıması, bu nesnelere arasında belirgin ilişkileri öğrenmesi ve duruma uygun cevap vermesi, çoğu robot çalışmalarının temelinde yatmaktadır (Pastacı ve Karlık, 1991).

Robot görme sistemleri tasarımında en önemli faktörler maliyet, gerçek zamanda çalışma, esneklik ve güvenilirliktir. Endüstride kullanılan robot görme sistemleri uygun bir maliyette ve karşılaştırılabilir işler için insan gücünden daha ucuz olmalıdırlar. Gerçek zaman çalışmalarının insan çalışanlar ile yarışması gerekir, bu ise bir dakika veya daha az zamanda bir görüntüyü işleyebilen bir görüntü işleme sistemi gerektirir (Nof,1985).

Son yıllarda elektronik teknolojisindeki başdöndürücü gelişmeye paralel olarak robot görmesi alanında da önemli gelişmeler kaydedilmiştir. Özellikle kontrollü çalışma alanları içinde, belirli bir amaca yönelik uygulamalarda robot görmesi başarı ile uygulanabilmektedir. Şekil-2.1'de bir robot kolunun görsel algılama sistemi yardımıyla bir üretim hattı üzerinden parçaları alışı gösterilmektedir. Bu sistem, günümüzdeki robot

görmesi uygulamalarına tipik bir örnektir. Henüz genel amaçlı görme sistemlerinin gerçekleşmesi uzak bir hedef olarak kalmakla birlikte, böyle sınırlı uygulamalardan cesaret verici sonuçlar alınmaktadır. Özel amaçlı robot görmelerinin, genel amaçlı robot görme sistemlerine dönüştürülmesindeki temel güçlük, işlenecek verilerin büyük boyutlara varmasından ve bu verilerden görüntüye ait sembolik tanımların elde edilmesi için bir hayli zamana ihtiyaç duyulmasından kaynaklanmaktadır (Tezer, 1993).



Şekil-2.1. Robot görmesi ile üretim hattında çalışan robot kolu.

Şekil-2.1 'de, üretim hattı üzerindeki cisimler, tavandaki bir kameranın önünden geçmektedir. Bu uygulamada sembolik tanımın; nesnelere tanıması, konumlarına ve doğrultularına ait sınırlı sayıda bilgi vermesi, yeterli olacaktır. Nesnelere diğer özelliklerine, renk vb. ait görsel bilgilere bu uygulamada yer yoktur ve sembolik tanımın bunları içermesi gerekmemektedir.

Robot görmesi aşağıdaki araştırma alanlarına yakından bağlıdır;

- Görüntü işleme
- Nesne tanıma
- Sahne analizi

Görüntü işleme, algılanan görüntüden yeni bir görüntü elde edilmesi ile ilgilidir. Nesne tanıma, görüntülerden cisimlerin tanınması amacıyla bilgiler çıkarılmasına yönelik bir çalışma alanıdır. Sahne analizi ise, görüntüden alınan bilgilerin yapılacak iş için gerekli biçime dönüştürülmesi ile ilgilidir.

2.2. Bilgisayar Görmesi

Bilgisayar görme sistemleri karmaşıklık gösterse de, hepsi için geçerli olan birkaç teknik vardır. Kameradan gelen bir görüntünün kullanışlı bir şekilde işleme tabi tutulmasından önce, bu görüntünün bilgisayar tarafından anlaşılabilir şekle dönüştürülmesi gerekmektedir. Bu dönüşüm işlemine görüntünün sayısallaştırılması denir. Bu işlemde; görüntünün her bir piksel değeri bir sayı olarak hafızada depolanacak olan karelere bölünür. Her piksel noktasında görüntünün parlaklığını ve koyuluğunu temsil eden bir tamsayı bulunur. Bütün piksel değerleri için bu işlem gerçekleştirildiğinde, görüntü tamsayılardan oluşmuş bir matris şekline dönüşür. Resim bilgisi bu biçime getirildiği zaman, yazılım tarafından işlenmeye hazırdır.

Bilgisayar görmesi bazı ölçütlere göre sağlanır. Fakat bu ölçütler nelere bağlıdır, nelerle sınırlıdır tam olarak açıklanamaz. Bu sınırları daha iyi inceleyebilmek için, araştırmacılar bilgisayar görmesini aşağıdaki bölümlere ayırmışlardır (Nof, 1985).

- Resim işleme
- Görüntü işleme
- Nesne tanıma
- Durum analizi
- Optik işleme
- Video işleme
- Görüntü yorumlama

Gerek robot görmesinde ve gereksede bilgisayar görmesinde olsun nesne tanıma önemli bir yer tutmaktadır. Çalışmanın bundan sonraki kısmında nesne tanıma, ayrıntıları ile incelenecektir.

2.3. Nesne Tanıma

Nesne tanıma, bir girdiyi çeşitli özelliklerini göz önüne alarak kendisine en yakın özelliklere sahip olan sınıfa dahil etmektir. Yani tanıma işlemi aslında, nesnelerin sınıflandırılmasından farklı bir işlem değildir.

Son yıllarda nesne tanıma problemine olan ilgi hızlı bir şekilde artmıştır. Nesne tanıma uygulamaları, yörünge bulma, tıbbi teşhis, biomedikal sinyallerin analizi, uzaklık algılama, insan yüzü ve parmak izi tanıma, ses tanıma ve anlama, makina parçalarını tanıma ve benzeri uygulamaları içerir. Nesne tanıma, görüntü işleme ile yakından ilgilidir. Görüntü işleme yalnızca kodlama, filitreleme ve onarmayı (restorasyon) içermez, aynı zamanda görüntünün analizini ve tanınmasını da içerir. Diğer yandan nesne tanıma alanı yalnızca özellik çıkarma, sınıflandırma değil ön işlemeyi ve örnek tanımayı da içerir.

Nesne tanıma, fiziksel ve zihinsel işlemlerden alınan ölçümlerin tanımlandırılması ve ölçümlendirilmesi ile ilgilidir. Örnekleri etkili bir şekilde tanımak için gürültü ve ölçümlerdeki lüzümsüz kısımları çıkarmak gereklidir. Sayısal veya sayısal olmayan karakteristik ölçümler ve bu ölçümler arasındaki ilişkiler örnek simgelendirilmesi ile çıkarılır. Özel bir amaç için örneklerin sınıflandırılması veya tanımlanması simgelemeye dayanır.

Endüstriyel bir ortamda tanınması istenen nesnelere, önceden bilinen nesnelere. Nesne tanıma, bir veri tabanında saklanmış nesne modelleriyle, nesne görüntüsünün eşleştirilmesi işleminden oluşur. Veri tabanı, tanınacak her nesneye ilişkin tüm tanımlamaları içerir ve tanıma işleminden önce hazırlanır. Nesne türüne ve uygulamaya bağlı olarak değişik tanımlama türleri kullanılmaktadır.

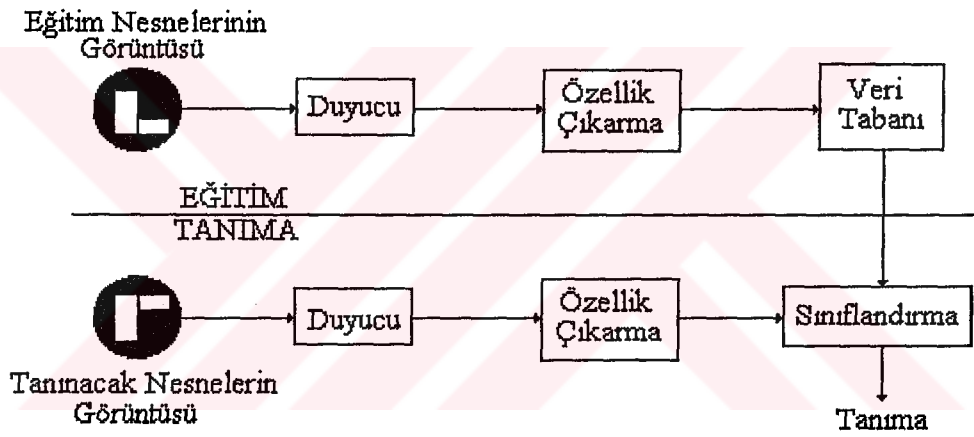
2.4. Boyutsal Nesne Tanıma Sistemleri

Nesne tanıma sistemleri, nesnelere tanımlama boyutlarına bağlı olarak iki boyutlu (2-B), üç boyutlu (3-B) ve 2½-B olmak üzere üç değişik sınıfa ayrılabilir. Her gruptaki sistemler için farklı özellik çıkarma (feature extraction), veri tabanı (modelleme) ve sınıflandırıcı (eşleme) teknikleri kullanılmaktadır. 2-B nesne tanıma izleyici merkezli bir koordinat sistemi kullanılır. Tanımlamalar, ikili ve gri-ton görüntülerden elde edilir. 3-B tanımlamalar, nesne-merkezli bir koordinat sistemi kullanır ve izleme noktasından bağımsızdır. Herhangi bir izleme noktasına ait bilgiler, bu tanımlamadan hesaplanarak bulunabilir. 2½-B tanımlamalar, 2-B ve 3-B tanımlama özelliklerinin ikisini de kullanır. Tanımlama, izleyici merkezlidir. Her görünümdeki

konum ve derinlik gibi yerel yüzey özellikleri kullanılır. Dolayısı ile her açıdan görünümünün veri tabanında bulunması gerekir (Chin ve Dyer, 1986).

2.4.1. Veri tabanları, özellik çıkarımları ve sınıflandırma

Tipik bir veri tabanlı tanıma sistemi, şekil-2.2 'de görülmektedir. Sistem, eğitim ve tanıma olmak üzere iki evreden oluşur. Sistemin en önemli elemanları; özellik çıkarma (nitelme), veri (model) tabanı oluşturma ve sınıflandırma (eşleme) bileşenleridir. Eğitim ve tanıma evrelerindeki duyucu (sensör) ve özellik çıkarma elemanları değişik olabilir. Alt bölümlerde her elemanın işlevi, daha ayrıntılı olarak tartışılacaktır.



Şekil-2.2. Veri tabanlı bir nesne tanıma sistemi.

2.4.1.1. Veri tabanları

Veri tabanları genellikle geometrik özellikler kullanılarak kurulur. Her gösterim türü için değişik özellikler kullanılmaktadır.

2-B veri tabanları, nesnenin olası 2-B görünümüleri kullanılarak kurulur. Pek çok uygulamada sınırlı sayıda görünüm yeterlidir. Özellik olarak nesnenin sınırları veya kenarları kullanılır. Özellik çıkarma işlemi, 2-B görünümlemler kullanılarak bulunmaya çalışılan 3-B veri tabanına (modellemeye) göre daha kolaydır. 2-B veritabanının olumsuz yanı bir boyutun kaybolmasıdır. Veritabanının yeterliliği, nesnenin karmaşıklığına ve olası

görüş açılarının sayısına bağlıdır. Bazı durumlarda 2-B görüntüler değişik 3-B nesnelere olarak yorumlanabilir.

2 1/2 -B modellerde yüzey bilgisi kullanılır. Dolayısıyla 2-B veritabanlarına göre daha fazla bilgi taşır. Tanımlamalar izleyici merkezlidir. Bu nedenle her görüş açısı için farklı bir tanımlama gerektirir. Bu gösterim türü için en önemli sorun, yüzey bilgilerinin elde edilmesidir.

3-B gösterimi izleyiciden bağımsız yüzeye bağımlı bir gösterimdir. En genel ve kapsamlı bilgiyi içerir. Bu veri tabanları, BDT (Bilgisayar Destekli Tasarım) sistemleri kullanılarak doğrudan elde edilebilirler. Ancak 2-B görüntülerden elde edilen özellikler, 3-B veri tabanlarıyla doğrudan karşılaştırılmaz. Dolayısıyla 2-B ve 3-B sınıflandırma (eşleştirme, correspondence) sorununun çözülmesi gerekir.

Sonraki bölümlerde üç tür nesne tanıma ile ilgili daha ayrıntılı bilgiler verilecektir.

2.4.1.2. Özellikler

Nesne özellikleri, kullanılan veritabanı ve uygulama alanına göre farklılık gösterir. Temel özellikler; kenar, köşe, doğru ve eğri çizgiler, delik ve sınır eğriselliği (curvature) dir. Nesne tanımlamaları bu özelliklerin birinden veya birkaçının birleştirilmesinden elde edilir. Gri-ton görüntü bilgileri, diğerlerine göre üst düzey bilgi taşıdıklarından, gürültüye karşı daha az duyarlıdır.

Endüstriyel uygulamalarda çoğu zaman nesne sınırları ve bu sınırlardan türetilmiş ölçümler, özellik olarak kullanılır. Bu özellikler genel, yerel ve ilişkisel (relational) olmak üzere üç gruba ayrılabilir. Genel özelliklere örnek olarak çevre, ağırlık merkezi, sınır noktalarının ağırlık merkezine uzaklığı, eğrisellik, alan, eylemsizlik momenti gibi özellikler verilebilir. Doğru parçaları, sabit eğriselliği olan cember parçaları yerel özellikler için örnek verilebilir. İlişkisel özelliklere örnek olarak, nesnenin alt parçalarının birbirlerine göre uzaklıkları, açıları gibi parametreler verilebilir.

Bazı uygulamalarda, kontrollü aydınlatma ve kontrastlı bir arka plan kullanılarak ikili görüntüler elde edilebilir. Daha sonra, ikili görüntülerden 2-B özellikler elde edilir. Nesnenin görüntü sınırları içindeki kenar bilgileri kullanılmadığından, özellikle 3-B nesnelere için bu yöntemler yetersiz kalmaktadır.

İkili görüntüler yerine, gri-ton görüntüler kullanılarak daha karmaşık tanıma işlemleri gerçekleştirilebilir. Gri-ton görüntülerden özellik çıkarma, daha zor ve daha zaman alıcıdır. Sınıflandırma, her ne kadar 2-B özelliklere dayansa bile daha güvenilir sonuçlar verir.

2.4.1.3. Sınıflandırma

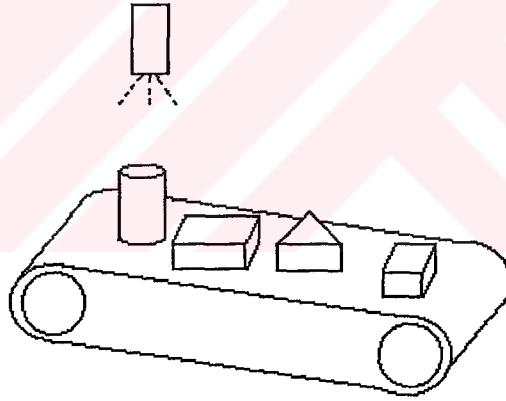
Sınıflandırma (eşleme) işleminin amacı, tanınması istenen nesnenin görüntüsünden elde edilen bir dizi özelliği, yaklaşık olarak veri tabanındaki model bir nesnenin bazı özellikleri ile eşleştirmektir. Sınıflandırma işlemi kullanılan nesne gösterim türüne göre değişir. 2-B genel özellikler için, çoğunlukla istatistiksel nesne tanıma yöntemleri kullanılır (Chin ve Dyer,1986). Yerel özellikler ile gösterilen nesne modeller için sintaktik sınıflandırma yöntemleri, ilişkisel özellikler ile gösterilen nesne modelleri için grafiksel (çizge) sınıflandırma yöntemleri kullanılır. 2 1/2 -B nesne modellerinde ise, yüzeyler karşılaştırılarak sınıflandırma yapılır. Bu karşılaştırma, veritabanındaki nesne model yüzeyleri ile görüntü yüzeyleri arasında en iyi çakışan bölgelerin aranması veya yüzeylerden elde edilen bazı özelliklerin karşılaştırılması biçiminde yapılabilir. 3-B nesne modellerini içeren bir veritabanı kullanıldığı durumlarda sınıflandırma algoritması, 3-B nesne modelinin 2-B izdüşümlerinden, özellikleri, görüntüden bulunan özelliklere en uygun olanını bulmaya çalışır. Eğer tanınacak nesneden üç boyutlu özellik bilgileri ("shape from shading, range, stereo imaging" gibi yöntemlerle) elde edilmişse, sınıflandırma işlemi doğrudan üç boyutlu uzayda yapılır.

2.4.2. İki boyutlu nesne tanıma

İki boyutlu görme sisteminde işlenecek bütün görüntüler sanki düzmüş gibi incelenmektedir. Bundan dolayı çok iyi kontrol edilen ve sınırlı bir alan gerekir. İki boyutu belirlemek gerektiğinden yüksek zıtlıkta siyah-beyaz (gri-ton) görüntü kullanılır. Şekil-2.3'de görme kontrollü bir taşıyıcı bant (konveyör) sistemi görülmektedir.

İki boyutlu sistemler; yönelimin ve yerleştirmenin tanımada önemli olduğu otomasyon montaj hattı gibi uygulama alanlarına sahiptir. İki boyutlu sistemlerde en

önemli gereksinim üç boyuttan iki boyuta indirgeme yapılırken, tanımlayıcı özelliklerin kaybolmamasıdır. Nesnelerin tanınmasında iki boyutlu görme çok sınırlıdır. Örneğin kare tabanlı bir pramit yukarıdan bakan bir kamera tarafından bir kare olarak tanınabilir ve yanlış yere yerleştirilebilir. Bu gibi durumlarda ya alternatif bir kamera pozisyonundan faydalanılır ya da ikinci bir iki boyutlu görme sistemi kullanılır. İki boyutlu görme sistemlerinin nesnelere tanımakta en önemli sorunu, nesnelerin üst üste gelme durumunda ortaya çıkar. Üçüncü boyut bilgisi olmadığından, birbirini tutmayan hatalı bilgilerin neden kaynaklandığı bilgisayara verilir. Fakat bu bilgiler ile hata tam olarak çözümlenmeyebilir. Mesela, yanlışlıkla bir karenin üstüne üçgen konulmuş olsun. Bilgisayar bunu anlamayacaktır. Bu tür problemler ancak üç boyutlu görme sistemlerinde çözülebilir. Üç boyutlu tanıma olayının karışıklığından dolayı, gerekli durumlar hariç, robotlarda ve genel olarak diğer görme sistemlerinde iki boyutlu görme kullanılmaktadır. Çoğu problem yüksek zıtlık metodu kullanılarak çözülebilir.



Şekil-2.3. Görme kontrollü bir taşıyıcı bant.

İki boyutlu nesnelere, değişik açılardan alınan iki boyutlu görüntüleri ile modellenilebilir. Bu görüntüler, doğrudan BDT gösteriminden hesaplanarak da elde edilebilir. Nesne tanıma işlemi ise, iki boyutlu görüntülerden elde edilen nesne modellerinden oluşan bir veritabanı kullanılarak sınıflandırma şeklinde gerçekleştirilir. Kullanılan veri tabanına ve sınıflandırma sistemine göre 2-B nesne tanıma yöntemleri üç

grupta incelenebilir. Bunlar genel özellik çıkarımı, yapısal özellik çıkarımı ve ilişkisel grafik (çizge) yöntemleridir.

2.4.2.1. Genel özellik çıkarım yöntemi

Bu yöntem çevre uzunluğu, alan, ağırlık merkezi, eylemsizlik momenti ve fourier tanımlayıcıları gibi genel özellikleri kullanır. Bu özelliklerin elde edilmesi oldukça kolaydır. Özelliklerin sıralanması ile bir öznitelik vektörü (veya özellik vektörü) oluşturulur. Tanıma işlemi için, bilinmeyen nesnenin öznitelik vektörü ile veritabanındaki model nesnelerin öznitelik vektörleri istatistiksel örüntü tanıma yöntemleri kullanılarak karşılaştırılır. Bu karşılaştırma sonucunda nesne, veri tabanında en çok benzediği nesne modeline göre sınıflandırılır.

Öznitelik vektörlerinin boyutlarının ve sayılarının sınırlı olması ve bir görüntüden kolayca elde edilebilmeleri nedeni ile bu yöntem oldukça hızlıdır. Ayrıca bazı özellikler dönme ve konumdan bağımsızdır. Ancak bu yöntemin sınırlamaları şunlardır :

- Her olası görüş açısı için ayrı bir model kullanılmalıdır.
- Nesneler birbirlerine değmemeli veya birbirini örtmemeli ve önemli kusurları olmamalıdır.
- Bir görüntüdeki tüm nesnelere ayırmak için tek bir eşik değerini kullanması gerekmektedir.

“SRI Vision Module” (Gleason ve Agin, 1979) genel özellik çıkarımını kullanan tanıma sistemi için bir örnek olarak verilebilir.

2.4.2.2. Yapısal özellik çıkarım yöntemi

Bu yöntemde doğru çizgi, eğri parçası, köşe gibi yerel özellikler sıralı bir liste biçiminde birleştirilerek yapısal bir veritabanı oluşturulur. Sıralama o şekilde yapılır ki ardarda birleştirildiğinde nesnenin sınırları çiziliyormuş gibi olur. Böyle veritabanı oluşturma türünde, nesnelerin birbirini örtmesinden dolayı bazı özellikler eksik olsa bile

tanıma olasılığı bulunmaktadır. Sınıflandırma hipotez sınama yolu ile gerçekleştirilir. Özellikler, nesne görüntüsünden faydalanarak veri tabanı kurulurken kullanılan yöntem ile bulunur. Bulunan özellikler veri tabanındaki nesne modellerinin özellikleri ile eşlenerek hipotezleri sağlayıp sağlamadığına bakılır.

Yapısal özellik çıkarım yöntemi, genel özellik çıkarım yöntemine göre daha üstündür. Ancak, daha karmaşık eğitim ve tanıma işlemleri gerektirir. Gri-ton görüntülerden elde edilen özellikler ikili görüntülerden elde edilen özelliklerden daha güvenilir ve sayıca çok daha fazladır. Bundan dolayı, veri tabanı oluşturma ve sınıflandırma işleminin, bütün durumların sınanmasını gerektirmeyecek biçimde tasarlanması gerekir.

Yapısal özellik çıkarım yöntemi için örnek olarak Stockman vd. (1982) tarafından geliştirilen yöntem verilebilir. Bu yöntemde veritabanı, sınır parçalarını birleştiren bir vektör ve parçanın deliklerini birleştiren doğruları tanımlayan vektörlerden oluşur.

2.4.2.3. İlişkisel grafik yöntemi

Bu yöntemde kenar, köşe gibi yerel özellikler arasındaki geometrik ilişkiler önem taşır. İlişkisel yapılar bir grafik ile gösterilebilir. Grafiğin her bir düğümü bir yerel özelliğe karşı gelir. Her düğüme, ilgili yerel özelliklere ilişkin bir dizi özellik verilir. Tanıma işlemi bir grafik sınıflandırma işlemidir.

Bolles ve Chain (1982) tarafından geliştirilen bir yöntem ilişkisel grafik yöntemi için örnek olarak verilebilir. Bu yöntemde delikler ve köşeler yerel özellikler olarak kullanılmıştır. Her özellik için nesnenin ağırlık merkezine olan uzaklığı ve konumu kaydedilir. Eğitim sürecinde her nesneyi diğerinden farklı yapan özellikler öbeklenir. Tanıma sürecinde her nesneyi belirleyen öbekleri oluşturan özellikler aranır ve böylece sınıflandırılmaya çalışılır.

2.4.3. 21/2 boyutlu nesne tanıma

Bu gösterim izleyici merkezli bir koordinat sisteminde, bir nesne tanıma türüdür. İki boyutlu gri-ton bilgiye ek olarak her pikseli tanımlayan bazı yüzey özellikleri, örneğin

her pikselin derinliđi, açısız konumu, rengi, hızı kaydedilir. Birden fazla görünüm gerekiyorsa her biri ayrı ayrı modellenerek veritabanı oluşturulur.

Yüzey bilgilerinin elde edilmesi gri-ton görüntülerin elde edilmesinden çok daha zordur, ultrasonik yöntemler ve özel aydınlatma teknikleri gerektirir.

Oshima ve Shirai (1983) tarafından geliştirilen bir yöntemde yüzey derinlik bilgileri ilişkisel grafik gösterimi içinde kullanılarak bir modelleme yapılmıştır. Sınıflandırma, gözlenen görüntüden elde edilen ilişkisel grafiđin, nesnenin her görünümü için çıkarılmış ilişkisel grafikleri ile karşılaştırılarak yapılır.

2.4.4. Üç boyutlu nesne tanıma

Üç boyutlu sistemlerin esas amacı genelde, başka bir nesne tarafından herhangi bir şekilde görünmesi engellenen nesnelerin problemlerini ele almaktır. İkinci bir amaç ise görüntüden topografiksel bilgiler çıkarmaktır. Mesela, bir ülkenin yer haritalarının çıkarılmasında kullanılabilir. Üç boyutlu görmenin gerektiđi bir çok uygulamada yalnızca bir kameralı görme sistemi yeterli değildir. Bunu bir gözümüzü kapatarak nesnelere bakmayı denemek şeklinde açıklamak çok kolaydır ama bunu bilgisayarda sağlamak oldukça zordur. Tek gözle üç boyutlu görmeye devam ederiz. Çünkü cisimlerin sınır çizgileri hakkındaki bilgiden çok daha fazla bilgiye (renk, gölge, uzaklık, parlaklık) sahibiz.

Üç boyutlu görmede iki boyutlu görmede olmayan bazı problemlerin çözülmesi gerekir. Bunlardan en önemlisi, bilgisayarın işlediđi görüntünün çok fazla ayrıntı içermesidir. Yüksek zıtlıktaki gri-ton görüntüde, her piksel ya siyah yada beyaz olduğundan bir bit olarak yüklenmiştir. Ancak, üç boyutlu bilgiyi elde etmek için her pikselin bađı parlaklıđı hakkında bilgi gerekir. Aynı zamanda, yüksek zıtlıktaki bir görüntüye göre bu formdaki görüntüyü analiz etmek çok daha fazla zaman alır.

2-B ve 2 1/2 -D nesne tanıma yöntemleri izleyici merkezlidir ve dolayısı ile herbir görünüm ayrı ayrı modellenmek zorundadır. Nesne karmaşık oldukça, model sayısı artar. Bu sorun izleyiciden bađımsız, oylumsal bir gösterim ile çözülebilir. Her nesne için bir tek öznelik vektörü kullanılır ve bu model nesnenin tüm görünümünü içerir. 3-B nesne tanıma iki grupta toplanabilir :

- Tam nesne tanıma (exact representation)
- Çok görünümlü nesne tanıma (multiview representation)

2.4.4.1. Tam nesne tanıma

Tam nesne tanıma türleri, nesneyi tüm olarak yüzeylerin tanımlaması ile modelleyerek veri tabanı oluştururlar. Yüzey veri tabanları nesnenin kenarlarını ve her yüzünün geometrik tanımlamalarını içerir. Yüzey veri tabanları, nesneyi genelleştirilmiş silindirler, küpler, kareler ve dikdörtgen blokların birleştirilmesi ile tanımlarlar. Her ne kadar bu nesne tanıma türleri nesneyi tam olarak modellese de gerçek zaman uygulamalarında bu veritabanlarını kullanmak çok zordur. Bunun nedeni, 2-B görüntüde bulunan özelliklerin, 3-B veri tabanına eşlenebilmesi için 2-B den 3-B ye ve 3-B den 2-B ye izdüşüm hesaplamalarının yapılmasına gerek duyulmasındandır.

Bolles vd. (1984) tarafından geliştirilen 3DPO sistemi (three dimensional part orientation systems) tam nesne tanımayı kullanan sistemlere örnek olarak gösterilebilir.

2.4.4.2. Çok görünümlü nesne tanıma

Bu nesne tanıma türü, nesnenin 2-B (veya 2 1/2-B) nesne tanıma modellerini birleştirerek bir tek model oluşturur. Çok görünümlü gösterim için iki yaklaşım kullanılmaktadır. Birinci yaklaşımda, nesnenin topolojik olarak farklı kararlı görünümleri kullanılarak veritabanı oluşturulmaktadır. Bu görünümler izleme noktasındaki küçük değişmelerin nesnenin görünen özelliklerinin oluşturduğu topolojik yapısında (kenarlar, köşeler gibi) herhangi bir değişiklik yapmayacak şekilde seçilmektedir.

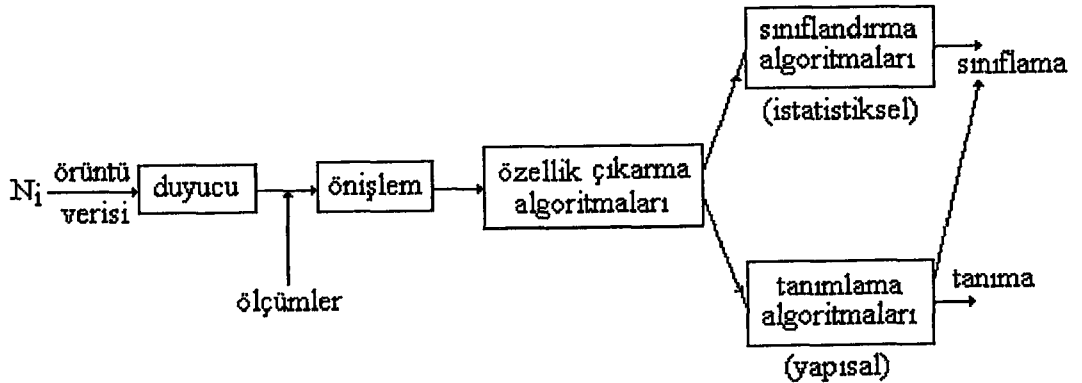
İkinci yaklaşım, ayrık gözlem küresi gösterimidir (discrete view sphere representation). Nesne, bu yaklaşımda onu çevreleyen bir kürenin değişik noktalarından izlenir. 2-B izleyici merkezli görünümleri birleştirilerek tek bir model elde edilir.

Lamdan ve Wolfson (1988) tarafından önerilen geometrik dağıtma (geometric hashing) yöntemi, çok görünümlü nesne tanımayı kullanmaktadır. Eğitim evresinde gözlem küresi üzerindeki noktalardan alınan 2-B görüntülerden ilgili noktalar (örneğin

köşeler) bulunur. Daha sonra her görüntü için, görüntü içindeki iki nokta kullanılarak bir koordinat sistemi oluşturulur ve diğer noktaların bu koordinat sistemindeki koordinatları hesaplanır. Bu koordinatlar bir tablo için indeks olarak kullanılır ve tabloda bu indekse karşı gelen öğeye nesne adı, görünüm numarası ve koordinat için seçilen baz (iki nokta) kaydedilir. Daha sonra bu işlem diğer nokta çiftleri için tekrarlanır. Son olarak her görünüm için bu işlemler tekrarlanır. Tanıma evresinde, görüntüde bulunan ilgi noktalarından ikisi seçilerek bir koordinat sistemi oluşturulur. Eğitim evresindeki gibi diğer noktaların koordinatları hesaplanır. Bu koordinatlar daha önce oluşturulan tablodan nesne adı, görünüm numarası ve baz çiftini okumak için kullanılır. Bir oylama tablosu oluşturularak, veri tabanındaki model nesne görünümlerinin hangisinin daha çok oy aldığına bakılır. En çok oy alan nesne modelinin hangi sınıfa ait olduğu belirtilerek nesne tanınır.

2.5. Nesne Tanıma Yaklaşımları

Nesne tanımada başlıca iki ana yaklaşım bulunmaktadır: istatistiksel ve yapısal yaklaşım. Son zamanlarda bunlara, yapay sinir ağı kullanan yaklaşımlar da eklenmiştir. Nesne tanıma yaklaşımlarının dönmeden , ötelemeden ve ölçeklemeden bağımsız olması en önemli olgudur. Tipik bir nesne tanıma sisteminin yapısı şekil-2.4 'de gösterilmektedir.

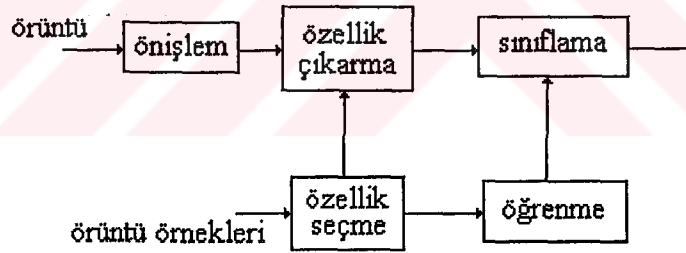


Şekil-2.4. Genel nesne tanıma yaklaşımı sistemi.

Bu yapıda, duyucular aracılığı ile alınan örüntü, çeşitli önışlemlerden (süzgeçleme, pekiştirme vb.) geçirildikten sonra öznitelik vektörü çıkarımı yapılır. Öznitelik vektörü, her bir nesne sınıfını en iyi belirleyen özellikler olarak tanımlanabilir. Daha sonra çeşitli sınıflandırma yaklaşımları kullanılarak giriş örüntüsünün hangi sınıfa ait olduğuna karar verilir.

2.5.1. İstatistiksel yaklaşım

İstatistiksel yaklaşımda, sınıflama algoritmaları için istatistiksel temel öngörülür. Aynı sınıfa ait nesnelere, istatistiksel olarak tanımlanan benzer özelliklere sahiptirler. Bu yaklaşımda, özellik olarak nitelendirilen karakteristik ölçümler nesne örneklerinden çıkarılır. Her nesne bir vektör ile tanımlanır. Genelde sınıflandırıcıyı oluşturan karar ve sınıflandırma yöntemleri üzerinde önemle durulur (Duda ve Hart, 1989). Sınıflandırıcı tasarımı, ölçümler ve olasılıklar gibi işlenebilir nesne bilgilerini birleştirmeyi esas alır. Şekil-2.5 'de istatistiksel nesne tanıma sistemi görülmektedir.



Şekil-2.5. İstatistiksel nesne tanıma yaklaşımı sistemi.

2.5.2. Yapısal yaklaşım

Yapısal (geometrik) yaklaşımda, verilen bir nesne formal yapıdan temel karakteristik tanımlanmaya indirgenir. Çoğu zaman, nesnelere çıkarılan bilgi yalnızca özellikler setinin sayısal değerlerinde değildir. Özelliklerin birbirine bağlanması veya aralarındaki ilişki, yapısal tanımlamayı ve sınıflandırmayı kolaylaştıran önemli yapısal bilgiye sahiptir. Yapısal yaklaşımda, yapısal bilgi nesnenin özellikleri arasındaki ilişki ve

bağlantılardan çıkarılır. Mesela, nesnenin köşe sayısı, kenar açıları vb. gibi. Genel olarak yapısal yaklaşımda daha basit alt nesnelere karışık nesnelere hiyerarşik tanımlamalarını formüle eder. Yapısal yaklaşımda her nesne, bileşenlerinin bir kompozisyonu olarak ele alınır (Banks, 1990). Şekil-2.6'da bir yapısal nesne tanıma sistemi görülmektedir.



Şekil-2.6. Yapısal nesne tanıma yaklaşımı sistemi.

Yapısal nesne tanıma yaklaşımında çeşitli birimler arasındaki ilişki çok büyük önem taşır ve gerçek tanımda kullanılan bazı formal notasyonlar tarafından belirtilir. Mesela, ekrandaki bir masayı tanıma, “köşelerinden eşit uzunlukta bacaklar tarafından desteklenen yatay bir dikdörtgen yüzey” gibi yapısal tanımlamayı temel alarak gerçekleştirilebilir. Bu yöntemde, çeşitli parçalar arasında ilişkilerin olması gerekir. Yapısal yaklaşım ve istatistiksel yaklaşım birleştirilerek daha güçlü bir nesne tanımlama gerçekleştirilebilir.

2.5.3. Yapay sinir ağları yaklaşımı

Yapay sinir ağları modelleri, nesne tanıma alanında uzun zamandır incelenmektedir. Yapay sinir ağları yaklaşımına göre tanıma modelleri, nesnelere öznitelik vektörlerini giriş olarak alan ve çıkış ayaklarının birinde cevap üreten, pek çok doğrusal olmayan hesaplama elemanlarının birleşimidir. Yapay sinir ağlarında her bir çıkış ayağı gözlenen nesnelere farklı bir sınıfa tahsis edilir. Yapay sinir ağları yaklaşımı temel olarak veri ve kural tabanlı sistemler ile yapılan sınıflandırma ve kategori temsili olan bir yapısal yaklaşımdır. Yapay sinir ağlarını temel alan pek çok sınıflandırma algoritması geliştirilmiştir. Bunlardan bazıları Hopfield Net, Perceptron, Hamming Net, Carpenter-

Grossberg sınıflandırıcısı, Back-Propagation algoritması ve Kohonen 'in kendi kendini düzenleyen öznitelik haritalarıdır (Nelson ve Illingworth,1991).

İstatistiksel, yapısal ve yapay sinir ağları ile nesne tanıma arasında kesin bir ayırım yoktur. Bu nesne tanıma yaklaşımları arasındaki sınırlar bulanıklık arzeder. Bu yaklaşımlar, genel ve ortak özellikleri ve amaçları paylaşırlar. Verilen belirli bir nesne tanıma probleminin çözümünde istatistiksel yaklaşıma göre nesnenin yapısı anlamsız olabilir. Yapı ancak uygun özellik seçimiyle yansıtılabilir. İstatistiksel nesne tanımada; yapısal bilginin ifade edilmesinde görülen zorluk, yapısal nesne tanımada kendini yapısal kuralların öğrenilmesinde gösterir. Buna karşın yapay sinir ağı yaklaşımı, istatistiksel ve yapısal yaklaşımlardan türetilmiştir. Açık bir şekilde nesne hakkındaki yapısal bilgi değerli olduğunda yapısal nesne tanıma yaklaşımını seçmek daha doğrudur. Yapısal bilgi değersiz ve maksada uygun değilse istatistiksel yöntemi seçmek daha doğrudur. Yapay sinir ağları, istatistiksel ve yapısal yaklaşıma alternatif teknikler sağlayan ve nesne tanımada yeni ortaya çıkmış bir kavram olarak düşünülebilir (Schalkoff, 1990).

2.6. Nesne Tanımada Karşılaşılan Problemler

Nesne tanıma sistemlerinde karşılaşılan birkaç ana problem vardır. Bunlardan biri, tanınması istenen nesnelere ölçülen giriş bilgilerinin nitelendirilmesidir. Her ölçülen nicelik nesne veya örüntünün karakteristiğini tanımlar. Nesne sınıflarını ayırmak için bu ölçümler yeterli olmayabilir. İkinci problem ise giriş bilgisinden öznitelik vektörünün çıkarımı ve bu vektörün boyutunun azaltılmasıdır. Sonuç olarak giriş nesnesinin daha özet hali elde edilmiş olur. Nesne tanıma sisteminin üçüncü problemi ise sınıflandırma işlemi için gereken minimum karar işlemlerinin saptanmasıdır. Sınıflandırma işlemi giriş nesnesinin sınıfını saptar. Nesnelere tanımak için yapılan eğitme işlemi ile, öznitelik vektörleri uzayında nesne sınıflarını ayıran karar sınırlarının belirlenmesi ile tanıma problemine çözüm getirilebilir.

Nesnelerin tanınmasında, özellikle nesnenin tanıma sistemine alınmasında en çok karşılaşılan problemler; sapma, gürültü, zayıf aydınlatma, yansıma ve nesnelerin bir kısmının herhangi bir nedenden dolayı görünmemesidir. Bu problemler nesne tanımada belirsizlik, yanlış tanıma gibi olayların ortaya çıkmasına neden olmaktadır. Bağımsız

nesne tanıma için birçok yaklaşım getirilmiştir. Fakat çoğunda gürültü ve belirsizlik ana engel olarak görülmektedir. Bir teknik, hiçbir belirsizliği olmayan bir nesne için doğru bir tanıma sağlarken, bu nesnenin herhangi bir kısmı örtülü olduğunda tanıma yapamayabilir veya yanlış tanıyabilir.

Nesne tanıma problemine çözüm, her nesne sınıfının özelliklerini çıkarmak, herbir giriş nesne vektörüne birkaç basit işlem uygulamaktır. Bu tür işlemler ile farklı sınıflara ait olan farklı nesnelere birbirinden ayrılır. Ancak giriş nesne vektörüne uygulanması gereken işlemleri saptamak için kesin bir kriter olmamakla birlikte, tanıma problemi özellik uzayında nesne sınıflarını ayıran karar sınırlarının üretilmesi olarak gözlenir.

Ekrandaki şekil bilgisini çıkarma; sayısal kamera çıkışı, ekranda matris data oluşturmakta kullanılan her piksel hakkında yeterli bilgiyi sağlar. Nesne ile arka plan arasındaki renk zıtlığı yeterli olduğu sürece cisme ait alanlar kolaylıkla oluşturularak kullanılabilir (Ioannis, 1993). Renk zıtlığından başka bir diğer problem ise, ekranda birden fazla nesne olduğunda ve bu nesnelere bazılarının üst üste geldiğinde oluşur. İki ve üç boyutlu sistemler kullanarak görüntüyü yorumlama problemleri çözülebilir. Fakat görüntüyü oluşturan nesnelere ve özelliklerini tanımlama problemlerini çözmek zordur. Bir görme sistemi oluşturulurken karşılaşılan en önemli problem nesnelere üst üste binmesidir. Bunu çözmek için gölgelemedeki farklılıklar yeterli ipuçları verir. Ana problem bir nesnenin belirli bir kısmını, nesneyi tanıyabilmek için saptayabilmektir. Mesela, bir üçgenin tanımlamasını üç kenarı, üç açısı var diye yaparsak ve görme sisteminde üst üste binmeden dolayı üçgenin bir kenarı gözükmeyebilir ise bilgisayar bu nesnenin üçgen olduğunu çıkartamaz. Bu sorunu çözmeye kontrollü halüsinasyon diye isimlendirilen insan gözünün uyguladığı yöntem çok yakın bir yöntem kullanılmaktadır. Sınırlı bilgiler yardımıyla gördüğünün bir üçgen olduğunu öngörerek kabul eder ve bunu kanıtlamaya çalışır. Örneğin, iki çizgi görüntünün dışında kesişiyorsa, bunu programlama ile kanıtlama çok dikkat isteyen bir işdir. Bu tür problemlerde en iyi çözüm yapay sinir ağları yaklaşımını kullanmaktır. Diğer bir güçlük ise nesnelere buldukları sınıflara göre tanımlamaktır. Özel bir nesneyi tanımak, belirli bir sınıftaki nesnelere tanımlamaktan daha zordur (Ikeuchi ve Hong, 1991).

BÖLÜM 3

3. İKİ BOYUTLU NESNE TANIMA SİSTEMİ

Nesne tanıma konusundaki tasarımlar için birçok teorik metodlar ve deneysel yazılım, donanım sağlanmıştır. Nesne tanıma işlemi, gözlenen nesnelerin daha önce tanımlanmış bir veya birden fazla kategoriye göre sınıflandırılmasıdır. Bundan dolayıdır ki nesne tanıma, bilgisayarın bir girdiyi çeşitli özelliklerini göz önüne alarak kendisine en yakın özelliklere sahip olan sınıfa dahil etmesidir. Buna göre, genel anlamda bir nesne tanıma sistemi iki aşamadan oluşur; nesnenin öznitelik vektörünün elde edilmesi ve nesnenin sınıflandırılması. İlk adımda, tanınması istenen nesnenin öznitelik vektörü çeşitli algoritmalar kullanılarak bulunur. Daha sonra bu nesneyi tanımlamak için elde edilen bu öznitelik vektörü (ki nesnenin özetidir) bir sınıflandırıcıya verilir. Sınıflandırıcı bu öznitelik vektörünü kendisine en yakın özellikler arzeden bir sınıfa dahil ederek, nesneyi tanımlar (Türkoğlu ve Arslan, 1996). Alt başlıklarda, genel bir nesne tanıma sisteminin bölümleri olan öznitelik vektörünün elde edilmesi ve sınıflandırıcılar sırasıyla ayrıntılı olarak anlatılmaktadır.

3.1. Öznitelik Vektörünün Elde Edilmesi

Öznitelik vektörünün elde edilmesinin amacı, sınıflandırıcı girişi için uygun bir veri yapısı oluşturmaktır. Öznitelik vektörü bir anlamda temsil ettiği nesnenin özetidir. Duyucular vasıtasıyla, bilgisayara sayısal olarak alınmış olan nesneye ait görüntüye, daha basit bir şekle getirmek için sırasıyla, görüntü eşikleme, kenar çıkarma ve nesnenin özelliklerini tesbit etme yöntemleri uygulanarak, elde edilen bu özelliklerden öznitelik vektörü kurulur. Sonraki bölümlerde bu aşamalar detaylı olarak verilmektedir.

3.1.1. Eşikleme

Sayısal bir görüntünün eşikleme işlemine tutulmasındaki amaç, nesnenin (görüntünün) özelliklerini belirlemede kolaylık sağlamaktır. Eşikleme işlemiyle, görüntü

iki renkle ifade edilebilir biçime getirilir. Görüntüyü eşikleme işlemine tabi tutmadan önce bir eşik değeri saptanır. Eşik değerinden daha yüksek gri seviye değerine sahip olan piksellere "1" değeri, daha küçük değerlere sahip olan piksellere ise "0" değer ataması yapılarak görüntü daha basit bir biçime (siyah-beyaz) getirilmiş olur. Şekil-3.1 'de bir görüntünün, eşikleme işleminden önceki ve sonraki durumu gösterilmektedir. Görüntü analizinde kullanılan eşikleme metodları hakkında ayrıntılı bilgi (Ioannis,1993) de bulunabilir.



Şekil-3.1. Eşikleme işlemi.

3.1.2. Kenar çıkarma

Kenar çıkarmadaki amaç, görüntünün içerdiği bilgiyi değerlendirip, gereksiz ve tanıma işlemlerinde zaman kaybettiren bilgiyi eleyerek yeterli düzeye indirgemektir. Kenar çıkarma, görüntü işleminin en temel konularından biridir. Görüntü analizinde ve nesnelerin sınıflandırılmasında çok değerli bilgiler taşıdığından, ilgi çeken bir araştırma alanı olmuştur.

Kenar çıkarma, robotik nesne tanımada çok önemli rol oynar. Bunu şu şekilde açıklayabiliriz. İnsan görme sistemi, nesnelere tanıma süreci içinde nesnenin dış çizgilerini izleyerek bir göz gezdirilir. Bu yaklaşımı yapay görme sistemlerine uyarladığımızda, şayet nesnenin sınırları başarılı bir şekilde izlenirse nesne tanımada çok daha iyi sonuçlar alınabilir. Böylece tanıma olayında kenar çıkarma önemli bir rol oynar. Çoğu görüntüler somut nesnelere içermez ve bu görüntüleri anlamak onların yapısal özelliklerine bağlıdır. Yapısal özelliklerin çıkarımı ise kenar çıkarma ile ilgilidir (Bağcıoğulları, 1994).

Bir kenar çıkarma yönteminin başarısını değerlendirmede kenar noktasının tanımı çok önemli yer tutar. Görüntüdeki bir nesnenin kenarı, nesnenin yüzey yoğunluğundaki değişimi ile ilgilidir. Kenar, farklı aydınlık değerlerindeki iki homojen alan arasındaki sınır olarak tanımlanabilir. Bu tanım, kenarın görüntüdeki aydınlık seviyesinin yerel değişimi

şeklinde de düşünülmesini sağlar. Kenar çıkarıcılarının etkinliği, homojen alan noktalarından gerçek kenar noktalarını ayırt etme yeteneğine bağlıdır.

Robotlarda nesne tanınması amacıyla geliştirilen kenar çıkarma algoritmalarının, iki boyutlu görme işlemini hızlı bir şekilde gerçekleştirmeye çalışırken, yüksek bir başarı yüzdesine ulaşmaları beklenir. Nesne tanımada sağlanabilecek olan en üst düzey başarı; boyuttan, parlaklıktan, yer ve duruş açısındaki değişimlerden etkilenmemedir. Bu başarıya ulaşmak için, görüntü işlenirken bilgi kaybı olmamalıdır. Fakat tüm görüntünün işlenmesi çok fazla zaman almaktadır. Bu sebeple, tanınması istenen nesnelerin kenarları çıkarılarak, nesne tanıma işleminin hızlı ve yüksek performanslı olması sağlanır (Uzunalioglu,1992).

Nesne tanınması işleminde, en önemli bilginin nesnenin kenarları olduğunu belirtmiştik. Nesnenin kenarlarını çıkarma algoritmalarında arzulanan sonuç hiçbir bilgi kaybına olanak tanımadan, hafızada en az yer tutan, tek piksel genişliğindeki en ince kenar izinin bulunmasıdır. Böylece nesne tanıma işlemi daha hızlı yapılabilir ve aynı zamanda nesne tanımak için oluşturulacak veri tabanları daha kolay oluşturulabilir. Bölüm 4 'de geliştirilen böyle bir kenar çıkarma algoritması verilmiştir. Ayrıca görüntü analizinde kullanılan diğer kenar çıkarma algoritmaları vardır. Bunlardan bazıları; Prewitt kenar çıkarma, Sobel kenar çıkarma (Ioannis, 1993) dir. Fakat bu kenar çıkarma yöntemlerinin hepsinde, nesnenin kenarları tek piksel kalınlığında değildir. Dolayısıyla tanıma işlemi, nesnenin kenarı daha fazla nokta ile tanımlandığından uzun zaman alabilmektedir.

3.1.3. Öznitelik vektörleri

Endüstriyel nesne tanıma çalışmalarında tanınması istenen nesne, sınırları ile tanımlanır. Çoğu nesne tanıma çalışmalarında, nesnenin sınırları üzerinde yapılan bazı nitelendirmeler sonucunda öznitelik vektörleri elde edilir. Öznitelik vektörünün oluşturulmasında sadece nesnenin sınırlarının kullanılması, nitelendirilecek bilginin azalmasından dolayı oldukça önemlidir. Bunun için nesnenin sınırlarını nitelemede pek çok teknik geliştirilmiştir. Çoğu tanıma tekniği, ya genel öznitelikleri ya da yerel öznitelikleri kullanır. Başlıca genel öznitelik yöntemleri; fourier tanımlayıcısı (Ioannis, 1993), otoregresif model (Kuhl, 1982), görüntüyü iki boyutlu simgeleme (Kollias ve Triakis, 1991), zerrike momentleri (Ioannis, 1993) ve yapısal örüntü tanıma (Karlık ve

Pastacı, 1993) uygulamalarıdır. Yerel öznitelik yöntemlerine ise, zincir kodları (Ioannis, 1993) ve poligonal yaklaşım (Ioannis, 1993) yöntemlerini örnek olarak verebiliriz. Nesneye ait yerel özellikleri işleyen yöntemlerde kritik noktalar veya delik ve köşe bilgileri türünden özellikler kullanılmaktadır. Şimdi bu öznitelik elde etme metotlarından bazılarını inceleyelim.

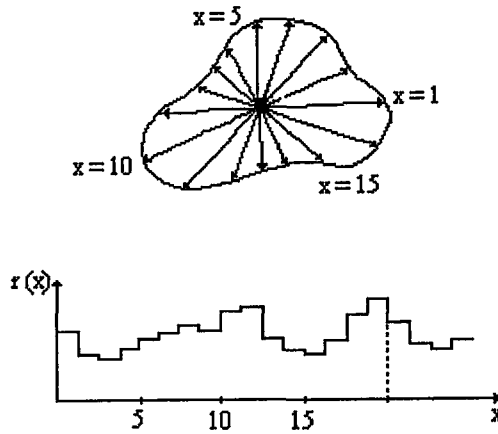
3.1.3.1. Otoregresif model

Bu modelin esası, nesnenin sınırının bir boyutlu simgelenmesinde merkezsel uzaklığın kullanımına dayanır. Algoritmanın ilk aşamasında merkez noktası hesaplanır. Daha sonraki aşamada nesnenin kenarını oluşturan her bir noktanın merkeze göre öklit uzaklığı bulunur. Bu işlemler için aşağıdaki formüllerden faydalanılır.

$$X_g = \frac{\sum_x \sum_y f(x, y) \cdot x}{\sum_x \sum_y f(x, y)} \quad Y_g = \frac{\sum_x \sum_y f(x, y) \cdot y}{\sum_x \sum_y f(x, y)}$$

$$d(i) = \sqrt{(x_i - X_g)^2 + (y_i - Y_g)^2}$$

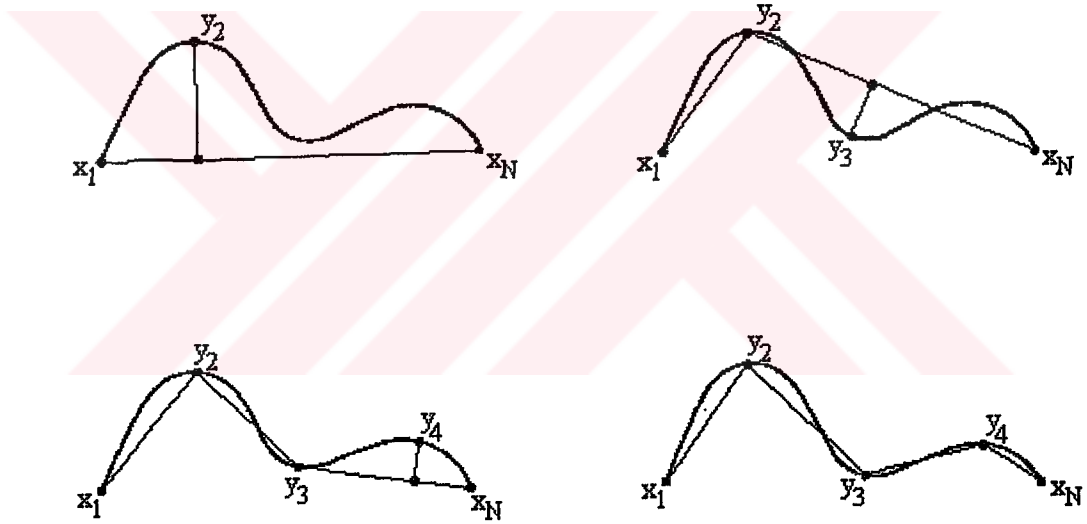
Eğer $f(x, y)$ nesneye ait ise 1 değerini, değilse 0 değerini alır. Böylece açığa bağlı olan değişimler bulunmuş olur. Şekil-3.2' de değişim görülmektedir. Elde edilen değerler vektör formunda yazılarak öznitelik vektörü elde edilir.



Şekil-3.2. Görülen nesnenin merkezine göre kenar uzunluklarının değişimi.

3.1.3.2. Poligonal yaklaşım

Poligonal yaklaşımın temelinde, nesneye ait kritik noktaların hesaplanması yatmaktadır. Kritik noktaların tesbitini sağlayan algoritmanın tanımı kısaca şu şekilde verilmektedir: Nesnenin en sol üst sınır pikseli ile en sağ alt sınır pikseli başlangıç kritik noktaları olarak alınır. Bu iki nokta arasında düz bir çizgi çizilir. Daha sonra bu düz çizgiye dik doğrular saptanır. Bu doğruların kenar piksellerini kestiği noktalar belirlenir. Düz çizgi ile bu noktalar arasındaki dik doğru parçalarının uzunluğu hesaplanır. Belirli bir eşikleme değerinin üstündeki noktalar kritik noktalar olarak tespit edilir. Bu noktaların tek boyutlu simgelenmesi ile elde edilen özelliklerden öznitelik vektörü oluşturulur. Şekil-3.3 'de örnek olarak alınan nesne kenarı üzerinde kritik noktaların bulunuşu gösterilmektedir.



Şekil-3.3. Poligonal yaklaşım ile nesnenin kritik noktalarının bulunması.

Yukarıdaki şekilde, verilen nesne kenarına poligonal yaklaşım uygulanarak nesnenin kritik noktaları tespit edilmiştir. Bu yaklaşım şu şekilde uygulanmaktadır;

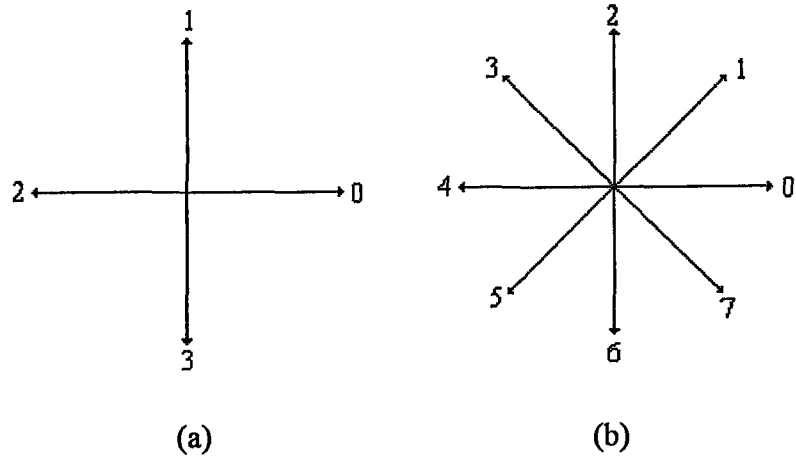
- Nesne üzerinde iki nokta seçilir.
- Bu iki nokta arasında bir doğru çizilerek, nesne kenarının üzerindeki noktaların bu doğruya olan izdüşüm uzaklıkları hesaplanır.

- Bu izdüşüm uzaklıklarından maksimum olanı seçilir. Bulunan bu maksimum uzaklık verilen eşik uzaklık değerinin üstünde ise nesnenin bu noktası kritik nokta (nesne köşesi) olarak alınır.
- Başlangıç noktasından başlanarak nesne köşeleri sırayla ardışık olarak birbirine doğru ile bağlanır.
- Nesne kenarlarının bu yeni çizilen doğrular üzerindeki izdüşümlerine olan uzaklıkları tekrar hesaplanır ve 3, 4, 5. adımlar, verilen eşik değerinden küçük veya eşit olana kadar tekrarlanır.

Bu işlemler sonucunda nesne sınır çizgisine yakın küçük doğru parçalarından oluşan yeni bir nesne kenarı elde edilir. Bu küçük doğrular kritik noktaların birleştirilmesi ile elde edilir. Kritik noktalardan oluşan nesneye ait özellikler bir vektör halinde yazılarak nesnenin öznitelik vektörü elde edilir.

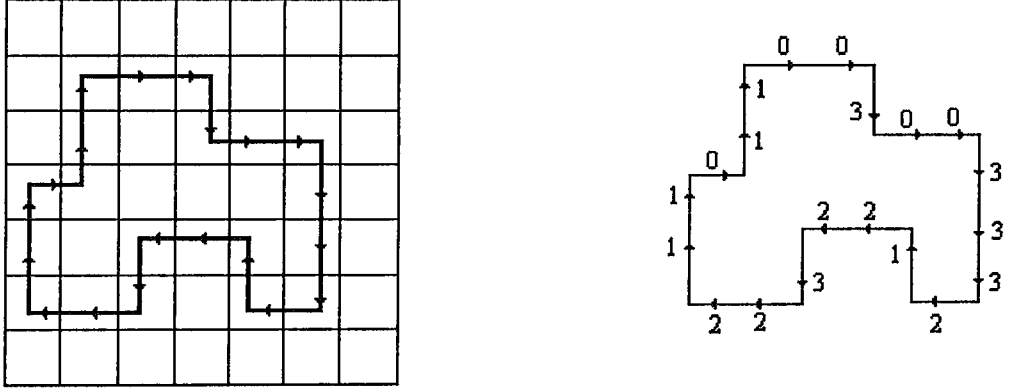
3.1.3.3. Zincir kodları

Sayısal görüntü, kenar çıkarma işleminden geçirildikten sonra, nesnenin sadece dış çizgileri kalır. Bundan sonraki adımda nesnenin kenar bilgilerinin kodlanması gerekir. Kodlama işlemi için zincir kodları kullanılabilir. İki tip zincir kodlama şekli mevcuttur; dörtlü ve sekizli zincir kod (Tezer, 1993).



Şekil-3.4. a) Dörtlü, b) Sekizli zincir kod.

Şekil-3.5 'de basit bir nesne ve bu nesneye ait dörtlü zincir kod gösterimi verilmiştir.

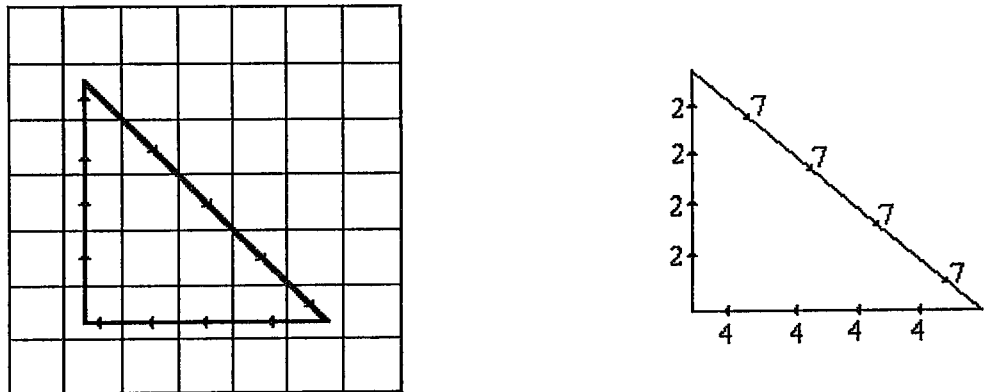


Şekil-3.5. Verilen sayısal görüntünün sınırlarının dörtlü zincir kod gösterimi.

Buna göre verilen nesnenin öznelik vektörü şu şekilde oluşur.

$$V = [0\ 0\ 3\ 0\ 0\ 3\ 3\ 3\ 2\ 1\ 2\ 2\ 3\ 2\ 2\ 1\ 1\ 0\ 1\ 1]$$

Nesnenin kenar bilgilerini daha hassas bir şekilde kodlamak için sekizli zincir kod kullanılması daha iyi sonuç verir. Şekil-3.6'da buna ilişkin bir örnek verilmiştir.



Şekil-3.6. Verilen nesnenin sınırlarının sekizli zincir kod ile kodlanması

Bu nesnenin öznitelik vektörü ise şu şekilde oluşur;

$$V = [7\ 7\ 7\ 7\ 4\ 4\ 4\ 4\ 2\ 2\ 2\ 2]$$

Sekizli zincir kodunun dörtlü zincir koduna göre en önemli üstünlüğü yatay ve dikey kenar çizgilerinin yanı sıra çapraz kenar çizgilerini de kodlama imkanı tanınmasıdır.

Zincir kodları ile nesnenin kenarlarının hangi yönde ne kadar devam ettiği tespit edildikten sonra elde edilen bu öznitelik vektörüne, çeşitli işlemler uygulanarak daha küçük boyutta ve tüm nesnelere için daha genel bir vektör oluşturulabilir. Mesela; öznitelik vektörü şu hale getirilebilir.

$$V = [V_0, V_1, V_2, \dots, V_7]$$

Burada;

V_0 : Toplam köşe sayısı,

V_1 : En küçük kenarın en büyük kenara oranı,

V_2 : 0 ile 45 derece arasında kaç açının kaç tane olduğu,

V_3 : 46 ile 90 derece arasındaki açının kaç tane olduğu,

V_4 : 91 ile 135 derece arasındaki açının kaç tane olduğu,

V_5 : -0 ile -45 derece arasındaki açının kaç tane olduğu,

V_6 : -46 ile -90 derece arasındaki açının kaç tane olduğu,

V_7 : -91 ile -135 derece arasındaki açının kaç tane olduğunu gösterir.

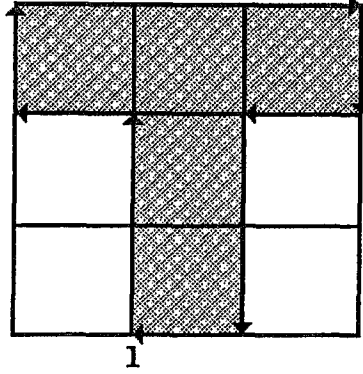
Elde edilen bu öznitelik vektörü, sonraki bölümlerde anlatılacak sınıflandırıcıların girişine verilerek nesne sınıflandırılması yapılabilir.

3.1.3.4. Yapısal nesne tanıma modelleri

Pek çok yapısal (geometrik) nesne tanıma modeli vardır. Bunlardan biri olan açı/uzunluk yönteminin yapısı, şekil-3.7 'de görülen, T harfinin örüntüsü örnek alınarak açıklanmaya çalışılmıştır.

Bu yöntemde T harfi yönlü bir geometrik şekil olarak düşünülmüştür. Harf birim uzunluk ve açı ile tanımlanmıştır. '1' numara ile gösterilen kısım referans açı olarak alınmış diğerleri de ona bağlı açılarla belirtilmiştir. T harfini tanıyan açı ve uzunluk değerleri,

Tablo-3.1 'de gösterilmiştir. Bu değerler kullanılarak verilen nesne için öznitelik vektörü kurulabilir.



Şekil-3.7. T alfabetik karakter örüntüsünün yapısal modeli.

Tablo-3.1. T harf örüntüsünün açı/uzunluk yöntemine göre yapısal özellikleri.

T alfabetik örüntüsü İçin		
No	Açı	Uzunluk
1	0	1
2	90	2
3	0	1
4	90	1
5	180	3
6	270	1
7	0	1
8	270	2

3.2. Sınıflandırıcılar

Sınıflandırıcılar, herhangi bir giriş vektörünü çeşitli özelliklerini gözönüne alarak, daha önceden oluşturulmuş bir veri yapısında belirtilen özelliklerle kıyaslayarak kendisine en yakın özelliklere sahip olan sınıfı bulma işlemini gerçekleştirirler. Bu amaçla nesne tanımada kullanılan sınıflandırıcılar, girilen herhangi bir nesnenin öznitelik vektörünü işleyerek, kendisine en yakın özelliklere sahip sınıfı bularak nesneyi tanımlarlar. Başlıca nesne sınıflandırıcı türleri;

- Klasik yöntemleri kullanan sınıflandırıcılar,
- Bulanık (fuzzy) sınıflandırıcılar,
- Yapay sinir ağları sınıflandırıcılarıdır.

Nesneler çok sayıda değişkene ve belirsizliğe sahip olduklarından, nesne sınıflandırma işlemlerinde klasik seri programlama teknikleri ile genelde yeterli başarıya ulaşılamamıştır. Bu nedenle, nesne sınıflandırma işleminde kullanılan ilk iki sınıflandırıcı tipi, ardışıl çalıştıklarından, istenilen (hız, tanıma doğruluğu,...vs.) şartlarda nesne tanımada başarı sağlayamamaktadırlar. Buna karşılık yapay sinir ağı sınıflandırıcıları, birbirine bağlı basit hesaplama elemanlarının paralel çalışmalarına ve insan sinir sisteminin işleyişini taklide dayanan, geniş amaçlı olarak nesne tanımada yeni yeni kullanılmaya başlanan umut verici bir modeldir. Model, basit hesaplama elemanları kullanmasına rağmen, çok sayıda elemanın paralel çalışmaları sonucunda, istenilen şartları çok yüksek oranlarda sağlayabilmektedir. Alt bölümlerde ilk iki tip nesne sınıflandırıcısından kısaca bahsedilmiş olup, konumuz açısından asıl önemli olan yapay sinir ağı sınıflandırıcıları ayrıntılı olarak verilmiştir.

3.2.1. Klasik sınıflandırıcılar

Klasik sınıflandırıcıların nesne tanıma problemine getirdiği çözümü en genel anlamda şu şekilde ifade edebiliriz; m tane farklı sınıf olup herbiri (n_1, n_2, \dots, n_m) şeklinde elemanlardan oluşan bir veri tabanı olsun. Her bir sınıfın her elemanı ise p tane özelliğe sahip (x_1, x_2, \dots, x_p) bir vektördür. Tanınması istenen nesnenin öznitelik vektörü, aşağıda belirtilecek olan “en yakın komşu” yöntemini kullanan klasik sınıflandırıcıya göre, veri tabanında en çok benzediği sınıfa dahil edilerek tanımlanır (Ünlüakın ve Gürgen, 1993).

3.2.1.1. En yakın komşu yöntemi

Bu yöntem, nesne tanımının en klasik metotlarından birisi olup, tanımlanması istenen nesnenin vektörünü, veritabanındaki en yakın komşusunun sınıfına dahil ederek

tanımlar. Yöntem, örnek vektörün istatistiksel dağılımından bağımsız olup, yalnızca en yakın komşunun sınıfına göre bir sınıflandırma yaparak tanıma işlemini gerçekleştirir.

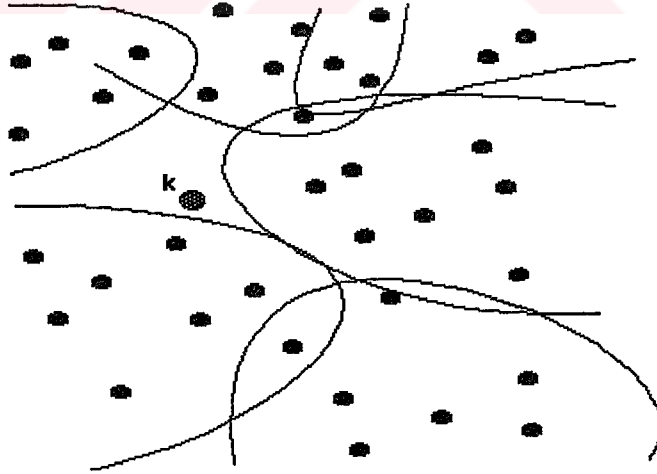
Bu yöntemde, örnek nesnenin vektörü alınarak, veri tabanındaki her bir vektöre olan uzaklığı ölçülür. En çok kullanılan mesafe ölçüsü öklit uzaklığı olsa da, başka herhangi bir ölçü de kullanılabilir. Tanımlanacak olan örnek nesnenin vektörü, veri tabanındaki kendisine en çok benzeyen, nesnenin sınıfından sayılır (şekil-3.8). Böylece örnek nesnenin vektörü en çok hangi sınıfın içinde yer alıyorsa, örnek nesne bu sınıf türünden bir nesne tipi olarak tanımlanır. Yöntem, daha matematiksel olarak aşağıdaki gibi ifade edilebilir :

Herhangi bir k vektörünün sınıflandırılacağını varsayalım. Elimizdeki veri tabanında toplam vektör sayısı n olsun. Veri tabanındaki her vektörün sınıfı bilinmektedir ve sınıfla vektör arasındaki ilişki;

$$C(x_i) = j \quad (j \in [1, m]) \quad \text{olarak verilir.}$$

Yine $uzk()$ adında, iki vektör arasındaki uzaklığı veren bir fonksiyon olduğunu varsayalım. O halde en yakın komşu yöntemi şu şekilde ifade edilebilir;

$$C(k) = C(\min(uzk(k, x_i))) \quad i \in [1, n] \quad \forall x_i \text{ için}$$



Şekil-3.8. Veri tabanındaki sınıflar ve en yakın komşu yöntemine göre k vektörünün sınıflandırılması.

3.2.2. Bulanık (fuzzy) sınıflandırıcılar

Bir bulanık sınıflandırıcının temelinde, "bir örüntünün w_i sınıfına üyelik derecesi β dir" tanımlaması yatmaktadır. Genel olarak bir bulanık sınıflandırıcı, bir örüntünün maksimum üyelik derecesine sahip olduğu sınıfın bulunmasını sağlar.

Bulanık sınıflandırıcıların, diğer bir önemli avantajı ise lineer olmayan olaylardaki başarılarıdır. Bir nesne hakkındaki bilgilerin belirsizlikler içerdiğini belirtmiştik; bu belirsizlikler üyelik fonksiyonları ile ifade edilebilirler. Sınıflandırma, bu fonksiyonların incelenmesinden elde edilen üyelik dereceleri ile gerçekleştirilir.

En çok bilinen bulanık sınıflandırıcıları ise, Fuzzy isodata ve Fuzzy c-mean algoritmasıdır (Bağcıoğulları,1995). Fuzzy c-mean algoritması, bir bulanık objektif fonksiyonunu optimize etmeyi temel alan ve data noktalarının kümelendirilmesi için en bilinen ve en popüler algoritmadır. Bu algoritma temel olduğundan aşağıda kısaca verilmiştir.

Objektif fonksiyonu :

$$J_m(u, v) = \sum_{k=1}^n \sum_{i=1}^c (u_{ik})^m (d_{ik})^2$$

Burada, u_{ik} k örneğinin i. sınıfına üye olma değeri, d_{ik} k örnek değerinin i sınıfının kümeleme merkezine mesafesi, U bütün u_{ik} 'ların toplamı, V bütün v_{ik} lerin toplamı, v i sınıfının kümeleme merkezi, n toplam örnek sayısı, c toplam sınıf sayısı ve m uygulama ile belirlenen bir faktördür ($m > 1$).

$$u_{ik} = \frac{1}{\sum_{j=1}^c \left(\frac{d_{ik}}{d_{jk}} \right)^{\frac{2}{m-1}}} \quad \text{ve} \quad d_{ik} \|x_k - y_i\| = 0$$

$1 \leq i \leq c$ olduğu takdirde J yerel olarak minimum olur. Burada uygulanan tanıma kararında, 0 durumunda giriş nesnesinin öznitelik vektörü ile veri tabanındaki modelin öznitelik vektörleri arasındaki minimum mesafe temel alınmıştır.

3.2.3. Yapay sinir ağı sınıflandırıcıları

Yapay sinir ağı sınıflandırıcıları, tanınması istenen nesnenin öznitelik vektörünü giriş olarak alan ve çıkış ünitelerinin birinde bu nesnenin sınıfını belirleyen bir cevap üreten, pek çok doğrusal olmayan hesaplama elemanlarının paralel işleyişinden meydana gelmiş tümleşik bir yapıdır. Yapay sinir ağlarında her çıkış ünitesi gözlenen olayın farklı bir sınıfını belirler (Zurada, 1992). Yapay sinir ağlarının paralel yapıları, bilgisayarları geleneksel yöntemlerden çok daha farklı kullanarak özellikle seri bilgisayarlarda bilinen yöntemlerle yapılması mümkün olmayan ve çok zor olan bir takım işlevleri (ses ve örüntü tanıma gibi) rahatlıkla yapmaları, yapay sinir ağlarını üstün kılmıştır.

3.2.3.1. Yapay sinir ağının tanımı ve modeli

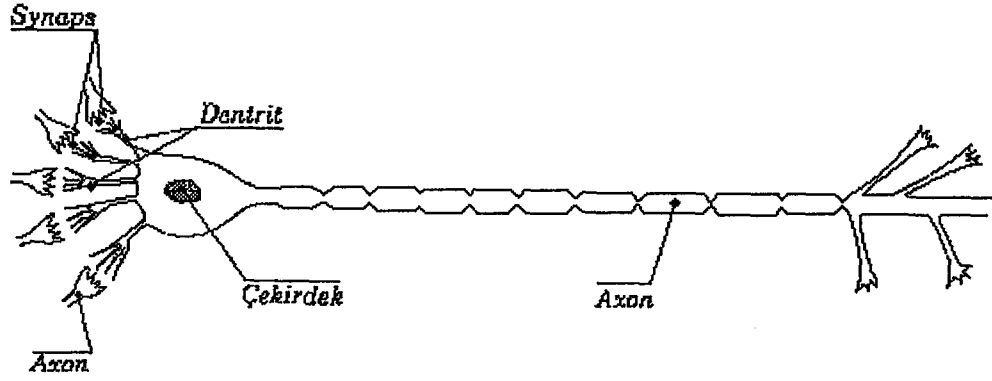
Yapay sinir ağları genel bir deyişle, insan beyninin çalışma prensibini taklit etme esası üzerine kurulmuş bilgi işleme yöntemleridir. Bu ağlar birbirine paralel olarak bağlanmış işlem elemanlarından (yapay sinir ağı hücresi, nöron, ünite, birim, düğüm) ve onların hiyerarşik bir organizasyonundan oluşurlar. Yapay sinir ağının temel düşüncesiyle insan beyninin fonksiyonları arasında benzerlikler vardır. Yapay sinir ağları, her ne kadar temel yapı itibarıyla bir kısım özellikleri insan beyninin fiziki özelliklerinden esinlenerek ortaya atılmış ise de, kesinlikle şu andaki halleri ile insan beyninin ne tam ne de yaklaşık bir modeli olarak değerlendirilemezler (Özmetel, 1996).

İnsan beyninin ne olduğu ve nasıl çalıştığı henüz kesinlik derecesinde keşfedilmiş sayılmaz. Günümüzde her ne kadar karmaşık matematiksel hesaplamaları ve hafıza işlemlerini eldeki mevcut bilgisayarlarla hızlı ve doğru yapmak mümkün ise de, aynı bilgisayarlarla beynin birçok basit fonksiyonunu (görmek, duymak koklamak, ...vs.) yerine getirmek ya mümkün olmamakta veya oldukça zor olmaktadır. Aynı şekilde biyolojik beyin, tecrübe ile öğrenme ve bilgiyi kendi kendine yorumlama, hatta eksik bilgilerden sonuçlar çıkartma kabiliyetine sahiptir. Bu, daha çok biyolojik sistemlerin, hücreler üzerinde dağıtılmış bilgiyi paralel olarak işleme özelliklerinden kaynaklanır. Hücreler birbirine bağlı ve paralel çalıştıklarından bazılarının işlevini yitirmesi halinde,

diğerleri çalıştığı için sinir sistemi, fonksiyonunu tamamen yitirmez. İşte yapay sinir ağları, bu özellikleri bünyesinde toplayacak şekilde geliştirilmektedir. Yapay sinir ağlarını daha iyi anlamak için, önce biyolojik sinir ağlarına bakmak faydalı olacaktır.

3.2.3.2. Biyolojik sinir ağları ve yapay sinir ağları arasındaki ilişki

Biyolojik sinir ağının en temel elemanı olan sinir hücresi, sinir sistemi içerisindeki fonksiyon ve görevlerine göre değişik şekil ve büyüklükte olabilir (Şekil-3.9). Bütün hücrelerin ortak bazı özellikleri vardır. Nöronun bir ucunda “dendrit” adı verilen ve hücreye, diğer hücrelerden veya dış dünyadan gelen bilgileri toplayan bağlantı elemanı, diğer ucunda ise tek bir life benzer “axon” adı verilen ve hücreden diğer hücrelere ve dış dünyaya bilgi taşıyan bağlantı elemanı vardır. Axon diğer hücrelerle birleşme esnasında dağınık dallara ayrılmaktadır. Bu iki uçtaki bağlantı noktalarının, elektrofizyolojik olarak hücrelerdeki bilgileri işlemede önemli yeri vardır. Hücrelerin birbiri ile elektrik sinyalleri vasıtasıyla irtibat kurduğu belirlenmiştir. Sinyaller bir hücrenin axon ’undan, diğerinin dendrit ’ine gönderilir. Bir axon birden fazla dendrit ile bağlantı kurabilir. Bu bağlantıların yapıldığı yere “synaps” denir. Hücreler, elektrik sinyalini hücre duvarlarındaki voltajı değiştirerek üretirler. Bu ise, hücrenin içinde ve dışında dağılmış iyonlar vasıtası ile olur. Bu iyonlar sodyum, kalsiyum, potasyum ve klorin gibi iyonlardır. Bir hücre, diğer hücreye elektrik enerjisini bu kimyasal iyonlar vasıtasıyla transfer eder. Bazı iyonlar elektrik ve magnetik kutuplaşmaya sebep olurken, bazıları kutuplaşmadan kurtulup hücre zarını açarak iyonların hücreye geçmesine olanak sağlar. Zaten sinyallerin bir hücreden diğerine akmasını sağlayan da bu kutuplaşmanın azalması olayıdır. Sinyaller, hücrenin etkinliğini belirler. Bir hücrenin etkinliği, hücreye gelen synaps sayısı, synaps ’lardaki iyonların konsantrasyonu ve bir de synaps’ın sahip olduğu güç olmak üzere üç faktöre bağlıdır. Bir hücre sahip olduğu dürtü miktarınca diğer hücreleri etkiler. Bazı hücreler diğerlerinin dürtülerini pozitif yönde, bazı hücreler de negatif yönde etkiler. İnsan sinir ağı sistemi, bu şekilde çalışan milyonlarca hücrenin bir araya gelmesinden oluşur (Karlık, 1995).

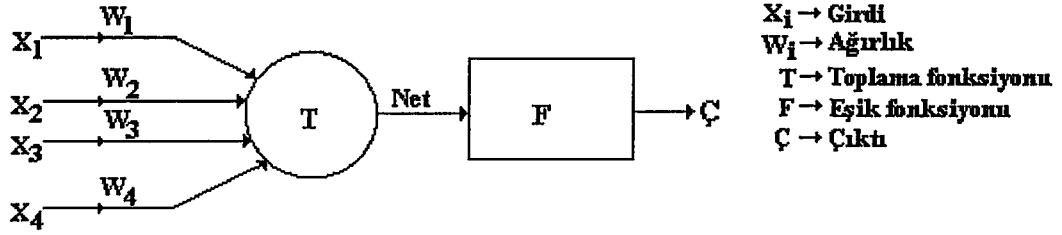


Şekil-3.9. Biyolojik nöronun şematik yapısı.

Biyolojik beynin en önemli özelliklerinden birisi de, öğrenme olayıdır. İnsanlar ve hayvanlar, sürekli olarak içerisinde buldukları çevre ile ilişki neticesinde bir öğrenme işlemi içerisindeyler. Öğrenilen her yeni bilgi, beynin fonksiyonlarını hemen etkileyerek, yapılan davranışlara yansır. Yapay sinir ağlarının gerçekleştirilmesinde bu özellik esas teşkil eder.

Yapay sinir ağları, biyolojik sinir ağlarından esinlenerek modellendirilmiş olup, onlardan çok daha basit bir yapıya sahiptir. Geliştirilen birçok yapay sinir ağı biyolojik sinir ağlarının bilinen birkaç özelliğini (öğrenme kabiliyeti gibi) temsil etmek üzere geliştirilmiştir. Bir takım özellikler ise nörofizyolojik yaklaşımlar yerine mühendislik yaklaşımı ile geliştirilmektedir. Bir yapay sinir ağının yapısını belirleyen bazı faktörler vardır. Yapay sinir hücreleri veya mühendislik tabiriyle işlem elemanları, sinir ağının yapısal modelleri, ağın sahip olduğu kural ve stratejisi bunların başında gelir. Bunlar hakkında detaylı bilgiler ve ilgili formüller (Zurada, 1992) bulunmakla beraber burada kısa bilgiler verilmesi yararlı olacaktır.

Yapay sinir ağları, biyolojik sinir ağları gibi yapay sinir hücrelerinin veya işlem elemanlarının bir araya gelmesinden oluşur. Herbir işlem elemanı şekil-3.10 'da görüldüğü gibi beş parçadan oluşur. Bunlar; girdiler, ağırlıklar, toplama fonksiyonu, eşik (dürtü veya çıktı) fonksiyonu ve çıktıdır.



Şekil-3.10. Yapay sinir hücresinin modeli.

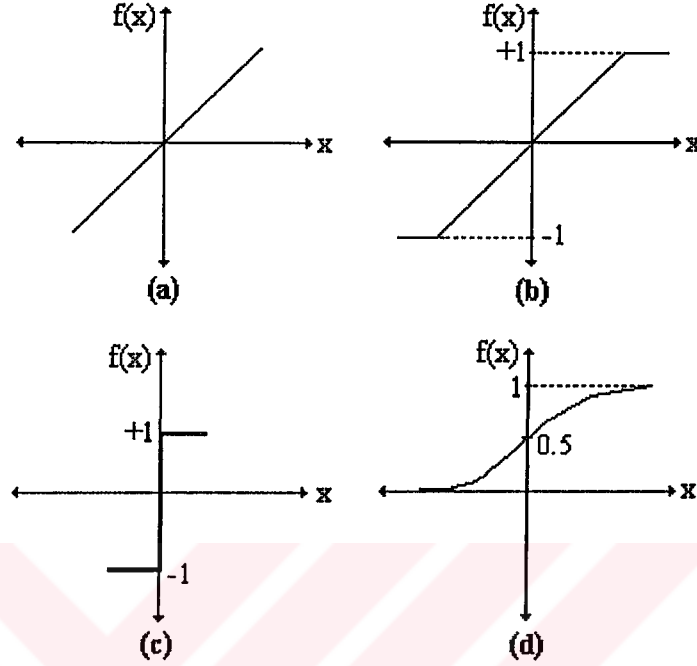
Tıpkı biyolojik sinir hücresi gibi, işlem elemanına birden fazla girdi gelmekte ve sadece bir çıktı gitmektedir. Girdiler, dentrit 'lere benzer şekilde diğer yapay hücrelerden bağlantılar vasıtasıyla işlem elemanına bilgi gelmesini sağlarlar. Bazı durumlarda bir işlem elemanı kendisine de bilgi geri gönderebilir (feedback). Bahsedilen bu bilgiler elemanlar arasında bulunan bağlantı hatları üzerinde depolanır. Her bağlantının bir ağırlığı vardır. Bu ağırlık bir işlem elemanının diğeri üzerindeki etkisini gösterir. Ağırlık büyüdükçe etki de büyür. Ağırlığın sıfır olması hiç bir etkinin olmaması, negatif olması ise etkinin ters yönde olması demektir. Bu ağırlıklar sabit olabildikleri gibi değişken de olabilirler. Toplama fonksiyonu, bir işlem elemanına gelen net girdiyi hesaplayan bir fonksiyondur. Net girdi genellikle gelen bilgilerin ilgili bağlantıların ağırlıkları ile çarpılıp toplanması ile belirlenir. Bu nedenle adı, toplama fonksiyonu olarak verilmiştir. Eşik fonksiyonu da, toplama fonksiyonu tarafından belirlenen net girdiyi alarak, işlem elemanının çıktısını belirleyen fonksiyondur. Genel olarak türevi alınabilen bir fonksiyon olması tercih edilir (şekil-3.11). Toplama ve çıktı fonksiyonları, ilgili probleme bağlı olarak farklı şekiller alabilirler. İşlem elemanının çıktı ünitesi ise çıktı fonksiyonunun ürettiği dürtüyü diğer işlem elemanlarına veya dış dünyaya aktarma işlevini yapar. İşlem elemanları ağın topolojik yapısına bağlı olarak tamamen birbirinden bağımsız ve paralel olarak çalışabilirler. Şekil-3.10'da verilmiş olan yapay sinir hücresinin matematiksel ifadesi şu şekildedir;

$$\text{Net} = \sum_{i=1}^4 w_i \cdot x_i = w_1 \cdot x_1 + w_2 \cdot x_2 + w_3 \cdot x_3 + w_4 \cdot x_4$$

$$\text{Ç} = F(\text{Net})$$

Eşik fonksiyonları, işlem elemanlarının sınırsız sayıdaki girişini önceden belirlenmiş sınırdaki çıkış olarak düzenler. En çok kullanılan dört tane eşik (aktivasyon) fonksiyonu

vardır. Bunlar, lineer, rampa, basamak ve sigmoid fonksiyonudur. Şekil-3.11 'de bu fonksiyonlar gösterilmiştir.



Şekil-3.11. Eşik fonksiyonları.

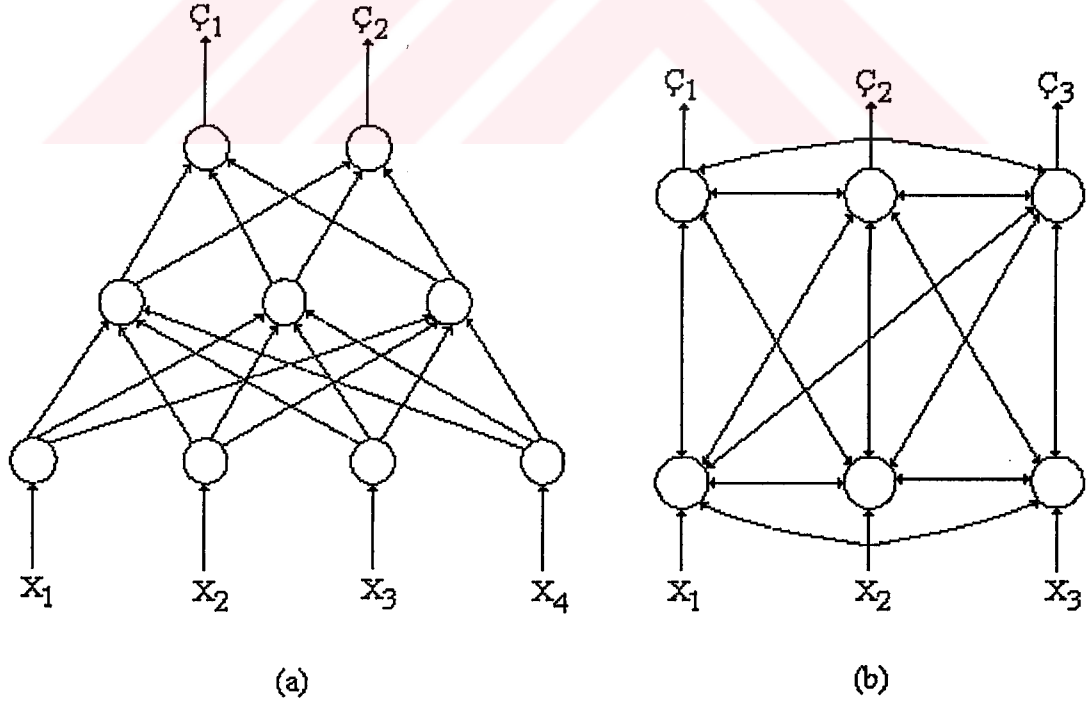
3.2.3.3. Yapay sinir ağlarının yapıları

Yapay sinir ağlarının görevlerini gerçekleştirmede, sahip oldukları fiziksel yapının da önemi vardır. Bugün 50 'ye yakın farklı yapılanma, diğer bir deyişle farklı model, görülmekte ve bu sayı her geçen gün artmaktadır (Özmetler, 1989). Farklı yapılaşma, işlem elemanlarının birbirleri ile olan bağlantılarından ve uygulanan öğrenme kuralından kaynaklanmaktadır. İşlem elemanları ya tamamen birbirleri ile bağlantılı veya yerel olarak guruplar halinde bağlantılı olabildikleri gibi değişik şekilde de birbirleri ile bağlanabilmektedirler. Bilgi akışı bu bağlantılar üzerinden tek yönlü olduğu gibi, çift yönlü de olabilir. Bir gurup işlem elemanı bir araya gelerek bir katman oluştururlar. Genel itibariyle yapay sinir ağlarında 3 tür katman bulunur. Sinir ağının dış dünya ile bağlantısını kuran girdi katmanı, gelen bilgileri işleme kabiliyetine sahip ara katmanlar ve sinir ağının kararlarını dış dünyaya aktaran bir çıktı katmanı (şekil-3.12). Girdi katmanında çoğu

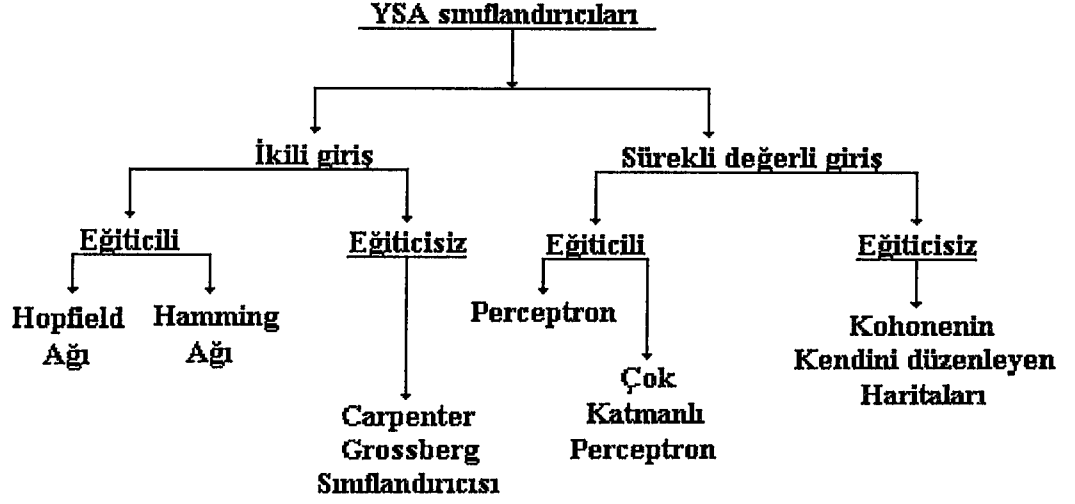
zaman bilgi işleme söz konusu olmaz. Bu katmandaki işlem elemanları aldıkları bilgiyi herhangi bir değişikliğe uğratmadan ara katmandaki işlem elemanlarına aktarırlar. Burada sözü edilen bilgi, sinir ağının işlem elemanları arasındaki bağlantı hatları üzerinde bulunan ağırlıklarla ifade edilir. Dolayısı ile bilgi, bütün ağa dağıtılmış demektir. Bu bilgiler, yapay sinir ağının belleğidir veya diğer bir deyişle yapay sinir ağları eğitildikten sonra kullanılacağı sistem için bir veri tabanıdır.

İşlem elemanları, onların birbirleri ile ilişkileri ve katmanlar arası ilişkiler değişik yapısal modellerin oluşmasına neden olmaktadır. Bu modelleri burada tek tek izah etmek oldukça fazla yer ve zaman gerektireceğinden, sadece örnek iki yapay sinir ağı modeline değinilecektir. Şekil-3.12.a 'da tek yönlü bağlantıların olduğu, bir katmandaki bilgilerin sadece bir üst katmana geçtiği ve yine bir katmandaki işlem elemanlarının bir üst katmana tamamen bağlandığı bir ağ yapısı gösterilmiştir. Bilinen meşhur geriye yayılım ağı modeli bu tür yapıya örnek olarak verilebilir. Şekil-3.12.b 'de ise çift yönlü ilişkilerin olduğu bir yapı sergilenmektedir. Hopfield modeli bu tür ağlara örnek olarak gösterilebilir.

Yapay sinir ağı sınıflandırıcıları giriş veri tiplerine göre, ikili giriş ve sürekli değerli giriş olmak üzere şekil-3.13 'deki gibi iki şekilde sınıflandırılırlar.



Şekil-3.12. a) İleri doğru tek yönlü, b) Çift yönlü tamamen bağlantılı yapay sinir ağı modeli.



Şekil-3.13. Yapay sinir ağı sınıflandırıcıları.

3.2.3.4. Yapay sinir ağlarında öğrenme ve öğrenme algoritmaları

Daha önce belirtildiği gibi bir yapay sinir ağının sahip olduğu bilgi, işlem elemanları arasındaki bağlantı hatları üzerinde saklanır ve ağırlıklar vasıtası ile gösterilir. Ağ, olaylar hakkında girdi ve çıktılar arasındaki ilişkiyi, elde bulunan mevcut örneklerden genellemeler yaparak öğrenir ve bu genelleme ile yeni oluşan ve ortaya çıkan daha önce hiç görülmemiş olaylar hakkında karar verir. Yani ağa, bir örnek olay gösterildiğinde, girdi katmanından alınıp ara katmanlarda işlenerek, ağın o olay hakkında ürettiği sonuç, çıktı katmanına sunulur. Bu bilgiyi işleme, ağın sahip olduğu tecrübeye göre, bilginin ara katmanlarda çağrıştırılması ile gerçekleştirilir. Bu çağrıştırma olayı modelden modele değişmektedir. Mesela, ara katmanlardaki işlem elemanları sahip oldukları bağlantılar ile kendi kararlarını üretir ve çıktı katmanındaki işlem elemanlarına gönderirler. Çıktı katmanındaki işlem elemanları da, yine ilgili ağırlıkları kullanarak ağın en son kararını oluştururlar. Bu ağırlıklar, tıpkı ilgili olayın belirli özelliklerini hafızada saklayan elemanlar gibi düşünülebilirler. Bilgi işleme ise (örüntü tanıma gibi), bir olay gösterildiğinde hafızadan ilgili özellikleri çağırarak ve bunlar ile ilgili girdileri birlikte analiz ederek karar vermek şeklinde yorumlanabilir.

Yapay sinir ağının öğrenme işlemini gerçekleştirebilmesi için, sahip olduğu bütün ağırlıkların, ilgili problemde öğrenilmesi istenen özellikleri genelleştirecek şekilde doğru

değerlere sahip olması gerekir. Bu doğruluk ne kadar artarsa ağın sınıflandırma kabiliyeti dolayısı ile öğrenme işlemi o kadar iyi olur. Doğru ağırlık değerleri, bir öğrenme kuralına göre tespit edilirler. Çoğunlukla bağlantılara başlangıç değerleri olarak rastgele ağırlıklar atanır ve bu ağırlıklar eldeki örnekler incelendikçe bir kurala göre değiştirilerek doğru ağırlık değerleri bulunmaya çalışılır. Kısaca belirtmek gerekirse öğrenme kuralları, bir işlem elemanının örnekleri gördükçe kazandığı tecrübeye göre, ilgili bağlantı ağırlıklarını nasıl değiştireceğini belirleyen algoritmalarıdır. Bu olayı bir örnek üzerinde açıklamak gerekirse, kendisine gösterilen bir nesnenin daire mi, yoksa üçgen mi olduğunu tanıyan bir ağ oluşturulmak istenirse, girdi katmanında sayısallaştırılmış bir nesneyi gösterecek sayıda, çıktı katmanında ise bir daire, bir de üçgeni göstermek üzere iki tane işlem elemanı olacaktır. Daire nesnesi gösterildiğinde 1. işlem elemanının aktif olması, üçgen nesnesi gösterildiğinde 2. işlem elemanının aktif olması istenerek ağ eğitilecektir. Dolayısıyla daire nesnesi gösterildiğinde 1. eleman aktif olacak şekilde ağırlıklar değişecek, üçgen nesnesi gösterildiğinde ise 2. işlem elemanı aktif olacak şekilde, ağırlıklar tekrar ayarlanacaktır. 10 daire ve 10 üçgen nesnesi olduğu var sayılırsa bunların tamamını doğru şekilde tanıyuncaya kadar bu ağırlık değiştirme işlemi tekrar tekrar devam edecektir. Bu ağırlıklar nesnelere ayıracak şekle geldiği zaman, artık ağ kendisine daha önce hiç gösterilmemiş (veya öğretilmemiş) bir nesnenin sınıfını (daire veya üçgen) belirleyebilecektir. Bu öğrenme olayı, genelde değişik yöntemlerden birisi, yani eğitici öğrenme olayıdır. Çünkü ağa, nesnenin daire mi, üçgen mi olduğu söylenmektedir. Yani ilgili işlem elemanının aktif olması istenmek suretiyle bu öğretim gerçekleştirilmektedir. Öğrenme yöntemleri temelde üç grupta toplanır (Zurada, 1992).

Eğitici Öğrenme : Bu yöntemde, dışardan bir eğiticinin sinir ağına müdahalesi söz konusudur. Eğitici, sinir ağının ilgili girdi için üretmesi gereken sonucu sinir ağı sistemine verir. Yani yapay sinir ağına girdi/çıkı ikilisinden oluşan örnekler sunulur. Bu ikili, ağın öğrenmesi gereken özellikleri temsil eder. Ağ girdi kısmını alır ve o anki bağlantı ağırlıklarının temsil ettiği bilgi ile bir çıktı oluşturur. Bu çıktı, olması gereken çıktı ile mukayese edilir ve aradaki hata tekrar ağa aktarılarak ağırlıklar bu hatayı azaltacak şekilde değiştirilirler.

Takviyeli Öğrenme : Bu yöntemde de yine eğiticiye ihtiyaç vardır. Yukarıdaki yöntemden farkı ise, bu durumda eğiticinin, ağın üretmesi gereken sonuc yerine, onun ürettiği sonucun sadece doğru veya yanlış olduğunu söylemesidir. Bu ise ağa bir takviye sinyalinin gönderilmesi ile gerçekleştirilir. Bu yöntem, ilgili örnek için beklenen çıktının oluşturulmadığı durumlarda çok faydalıdır.

Eğitici-siz Öğrenme : Bu durumda hiç bir eğiticiye ihtiyaç yoktur. Bu nedenle çoğu zaman buna kendi kendine organize olma (self-organized learning) da denilmektedir. Ağ, kendine gösterilen örnekleri alır ve belli bir kritere göre sınıflandırır. Bu kriter önceden bilinmeyebilir. Ağ, kendi öğrenme kriterlerini kendisi oluşturmaktadır.

Öğrenme işleminin, başlangıçtaki rastgele seçilmiş ağırlıkların, belirli bir kurala göre değiştirilmesine dayandığı belirtilmişti. Bu öğrenme kurallarının önemli olanlarından bazıları aşağıda yer almaktadır. Bu algoritmalarda, $w_i = [w_{i1} w_{i2} \dots w_{im}]$ vektörü ağdaki i . hücreye gelen n tane girişin ağırlık katsayılarını, $x = [x_1 x_2 \dots x_n]$ vektörü ağa gelen n tane girişi, y_i ve d_i 'de sırasıyla i . hücrenin hesaplanan ve istenen çıkışını temsil etmektedirler.

Hebb Öğrenme Algoritması :

Hebb öğrenme algoritmasında temel fikir, hücrenin çıkış üretmesine sebep olan bağlantıların ağırlık katsayılarını büyütürük etkilerini arttırmaktır. Buna göre değişim aşağıdaki gibi tanımlanmıştır.

$$w_i(t+1) = w_i(t) + \epsilon \cdot f(w_i(t) \cdot x(t)) \cdot x(t)$$

veya

$$w_i(t+1) = w_i(t) + \epsilon \cdot y_i(t) \cdot x(t)$$

Burada ϵ 'a öğrenme katsayısı veya öğrenme oranı denilmektedir. Dışardan istenilen bir çıkış uygulanmadığından, eğitici-siz bir öğrenme algoritmasıdır.

Algılayıcı (perceptron) Öğrenme Algoritması :

Rosenblatt tarafından önerilen algılayıcı öğrenme algoritmasında w değişimi, nöron çıkışı ve istenilen çıkışın farkı ile orantılıdır. Dolayısıyla eğitici bir öğrenme algoritmasıdır.

$$w_i(t+1) = w_i(t) + \varepsilon \cdot [d_i(t) - \text{sgn}(w_i(t) \cdot x(t))] \cdot x(t)$$

Ağın ürettiği çıkış istenen çıkıştan farklı ise, hata azaltılacak yönde hareket edilmektedir. Hata sıfır olduğunda w değişimi durmaktadır.

Delta Kuralı (Eğimdüşme) Öğrenme Algoritması :

Eğimdüşme (gradient descent) öğrenme algoritması veya “delta kuralı” öğrenme algoritması, sadece türetilbilir fonksiyonlara sahip olan ağlara uygulanabilmektedir. Eğiticili bir öğrenme algoritmasıdır. w değişimi, hatanın eğiminin tersi yöndedir. Böylece, hata fonksiyonunun minimum olduğu yere doğru hareket edilmektedir.

$$w_i(t+1) = w_i(t) - \varepsilon \cdot \frac{\partial E(t)}{\partial w(t)}$$

$E(t)$, hata fonksiyonunu temsil etmektedir ve aşağıdaki gibi tanımlanmıştır.

$$E(t) = \frac{1}{2} (d_i(t) - y_i(t))^2$$

$$E(t) = \frac{1}{2} [d_i(t) - f(w_i(t) \cdot x(t))]^2$$

$$\frac{\partial E(t)}{\partial w(t)} = -[d_i(t) - f(w_i(t) \cdot x(t))] \cdot \frac{\partial f(w_i(t) \cdot x(t))}{\partial (w_i(t) \cdot x(t))} \cdot x(t)$$

Bilinen hata geriye yayılması algoritmasının esasında, bu öğrenme kuralına dayanmaktadır.

Widrow-Hoff Öğrenme Algoritması :

Widrow-Hoff öğrenme algoritması, eğiticili bir öğrenme algoritmasıdır. w değişimi ağın çıkışından bağımsızdır.

$$w_i(t+1) = w_i(t) + \varepsilon \cdot (d_i(t) - w_i(t) \cdot x(t)) \cdot x(t)$$

İlintili Öğrenme Algoritması :

Hebb öğrenme algoritmasının eğiticili uyarlamasıdır. $f(w_i(t).x_i(t))$ yerine $d_i(t)$ istenen çıkış alındığında aşağıdaki denklem elde edilir.

$$w_i(t+1) = w_i(t) + \varepsilon.d_i(t).x(t)$$

Yarışmacı Öğrenme Algoritması :

Bu algoritmada esas, giriş işaretine en yakın olan bağlantı ağırlık katsayılarını bulmaktır. Bu, w 'lara ilişkin nöron kazanan nöron olarak adlandırılmaktadır. Sadece kazanan nöron bir çıkış üretmekte ve bu nörona ilişkin w 'lar değişime uğramaktadır. Eğiticiyiz öğrenmenin tipik bir örneği olan bu algoritma sonuçta ağa gelen girişleri sınıflandırmaktadır.

$$w(t+1) = w(t) + \varepsilon.(x(t) - w_m(t))$$

Burada w_m kazanan hücrenin bağlantı ağırlık katsayısı olup aşağıdaki gibi tanımlanmıştır.

$$w_m \cdot x = \max(w_i \cdot x) \quad \text{ve} \quad i = 1, 2, \dots, p \quad \text{dir.}$$

Outstar Öğrenme Algoritması :

Eğiticili öğrenme algoritma türü olan bu öğrenme algoritmasında amaç, w 'ları istenen çıkışa benzetmektir.

$$w(t+1) = w(t) + \varepsilon.(d_i(t) - w(t))$$

3.2.3.5. Yapay sinir ağlarının temel özellikleri

Yapay sinir ağlarını hem popüler yapan, hem de geleneksel bilgi işleme metotlarından ayıran özellikleri vardır. Yapay sinir ağlarının bir takım özellikleri, her ne kadar ilgili problemin yapısına ve kullanılan sinir ağı modeline bağlı olsada, genel özellikleri aşağıdaki gibi verilebilir (Karlık, 1995).

Öğrenme : Daha önce söz edildiği gibi, yapay sinir ağları, olaylar hakkındaki ilişkiyi belirli bir algoritmaya dayanarak çözmek yerine o ilişkiyi gösteren örnekleri incelemek suretiyle çözümler üretmeyi sağlarlar. Burada, ilgili olay ile alakalı sinir ağına örneklerden başka hiç bir ön bilginin verilmemiş olması dikkat çekicidir. Ağ, kendisine gösterilen örnekleri tekrar tekrar inceleyerek aradaki ilişkiyi kavramaya çalışır. Her yeni örnek, ağın sahip olduğu bilgiye bir yenisini ekler ve bu işlem tekrar ettikçe ilgili problem hakkında bazı genellemeler yapılır.

Genelleme : Alışagelmiş bilgisayarların bir karakteristiği, istenen çıkışı üretmek için, tam olarak doğru girişlere ihtiyaç göstermeleridir. Öte yandan yapay sinir ağları, girişlerinde değişimler olsa bile doğru çıkışı üretebilirler. Mesela, basılı harfleri tanımak için eğitilmiş bir sistem, gürültüden dolayı giriş karakterlerinin bozulması halinde bile doğru olarak çalışmaktadır. Yani sistem, daha önce o tipten hiç bir şey görmemiş olmasına rağmen, insanlar gibi tamam olmayan veya kısmen hatalı girişlerle bile, harfleri tanıyabilmektedir. Çalışmalar, insanların çoğunun bir metindeki harflerin yarısından fazlası silinmiş olsa bile, o metni okuyabildiklerini göstermiştir.

Yapay sinir ağları, kendilerine gösterilen bir şekli daha önce öğrendikleri ile mukayese ederek ve aradaki benzerlikleri ortaya koyarak, belirli sınıflara ayırma özelliklerine sahiptirler.

Çıkarım Yapma : Yapay sinir ağları tam doğru olmayan bir eğitime kümesinden, tam doğruyu çıkarabilirler. Ses tanımak için eğitilmiş bir yapay sinir ağına, gürültü tarafından bozulmuş ses verilebilir. Buna rağmen eğitimden sonra, sistem girişi bozuk ses olmasına rağmen, çıkışta ses mükemmel bir şekilde oluşturulabilir. Yani sistem, eğitim kümesinin özünü çıkarıp saklamıştır. Böylece, eksik veya gürültülü girişlere karşın uygun şekilde cevap verebilmektedir.

Hata Toleransı : Verilerde eğer bir eksiklik söz konusu olursa, geleneksel yöntemler çalışmazlar. Daha önce belirtildiği üzere, iyi eğitilmiş ve genelleme kapasitesi yüksek bir sinir ağı, kendisine takdim edilen veriler eksik olsada karar verme işlemine devam eder. Aynı şekilde, yapay sinir ağı üzerinde birtakım problemler ve bozukluklar olabilir. Geleneksel sistemlerin tersine yapay sinir ağları, bu durumda da çalışmalarına devam

ederler. Verilerdeki eksiklik veya yapay sinir ağındaki yapısal bozukluk arttıkça yapay sinir ağıının performansı yavaş yavaş azalmaya başlar. Fakat sistem fonksiyonunu tamamen durdurmaz ve mutlaka bir sonuç üretilir. Bu özellikler yapay sinir ağıının yapısından kaynaklanmaktadır. Çünkü ağıın sahip olduğı bilgi, ağı üzerindeki hücrelerin birbiri ile olan bağlantıları üzerine dağıtılmıştır. Zaten böyle bir durumda tek bir bağlantı ve onun üzerindeki bilgi, başlı başına hiçbir zaman bir mana ifade etmez. Ancak, bir grup halinde ve tam olarak bağlantıların birlikte düşünülmesi sonucu anlamlı bilgiler üretilir. Bundan dolayı birkaç bağlantının etkisiz hale gelmesi, sonucu ya etkilemez veya performansı yavaş yavaş düşürür. Yapay sinir ağıı sahip olduğı diğere bağlantılar nedeniyle işlevine devam eder. Geleneksel sistemlerin ardışıl çalışmalarından dolayı, sistemdeki en küçük bir hatanın veya bozulmanın ulaşabileceğı boyutlar düşünülürse bu özelliğın ne kadar önemli olduğı ortaya çıkar.

Hız : Bilgi işleme hızı bilgisayar teknolojisinde önemli bir yer teşkil eder. Sistemlerin her geçen gün biraz daha karmaşık olduğı, dolayısı ile daha fazla hacimde veriyi daha verimli bir şekilde işleme gerekliliğı, yeni yazılım/donanım sistemlerinin zorunluluğunu ortaya çıkarmıştır. Diğere yandan insan beyni oldukça fazla bilgiyi gerçek zamanlı (real-time) olarak, oldukça hızlı bir şekilde işleyebilmektedir. Yapay sinir ağlarının da, yine birbirlerine bağılı ve paralel işlem elemanlarından oluştuğundan böyle hızlı işleyebilmeleri bu ağlara özellikle endüstriyel hayatta çok önemli olan gerçek zamanlı çalışma kabiliyeti kazandırır.

BÖLÜM 4

4. YAPAY SİNİR AĞLARI İLE İKİ BOYUTLU NESNE TANIMA İÇİN GELİŞTİRİLEN SİSTEM VE UYGULAMASI

Nesne tanıma çalışmalarının çoğunda karşılaşılan en önemli problemler; nesne bozulması, gürültü ve örüntünün bir kısmının herhangi bir nedenden dolayı gözükmemesi veya fazlalıklar oluşmasıdır. Bu problemler nedeni ile nesne tanıma sistemlerinde bazı belirsizlikler ve yanlış tanımlar görülmektedir. Bundan dolayıdır ki nesne tanımada, klasik seri programlama teknikleri ile genelde yeterli başarı sağlanamamaktadır. Yapay sinir ağları ise, insan sinir ağı sistemine benzer bir paralel çalışma yapısı sergilediğinden, nesne tanıma çalışmaları için umut vericidir.

Bu çalışmada, yapay sinir ağlarının üstünlüklerinden yararlanarak genel bir nesne tanıma sistemi oluşturulmuştur. Daha önceki bölümlerde belirttiğimiz gibi nesne tanıma işlemi iki aşamadan oluşur. İlk olarak nesnenin öznitelik vektörü çıkarılır ve daha sonra bu öznitelik vektörü bir sınıflandırıcıya (yapay sinir ağına) verilerek nesnenin tanımlanması sağlanır. Aşağıda, sırasıyla öznitelik vektörünün elde edilmesi için geliştirilen algoritmalar ve yapay sinir ağı sınıflandırıcıları, uygulamaları ile birlikte verilmiştir.

4.1. Öznitelik Vektörünün Çıkarılması

Öznitelik vektörü, nesnenin çeşitli özelliklerini ihtiva ederek yapay sinir ağı sınıflandırıcısının girişi için uygun bir veri yapısı sağlamak için kullanılır. Bir nesnenin öznitelik vektörünün elde edilmesi için, nesneye ait sayısal görüntüye, ilk önce kenar çıkarma ve kenar çizgisini izleme işlemleri uygulanır. Alt bölümlerde sayısal haldeki görüntüye uygulanmak üzere bu tez çalışması kapsamında geliştirilen, yeni bir kenar çıkarma ve kenar çizgisini izleme algoritması verilmiştir. Bu algoritmalar nesne tanıma çalışmasında çok büyük önemi olan öznitelik vektörlerinin çıkarılmasında, önemli oranda kolaylık sağlamaktadırlar.

4.1.1. Kenar çıkarma

Nesne tanıma çalışmalarının en önemli konularından bir tanesi, nesnenin kenar çizgisinin tek piksel kalınlığında elde edilebilmesidir. Böylece nesneyi işlemek için daha az bilgi kullanıldığından, nesne tanımda oldukça önemli olan hız problemine zamandan tasarruf sağlanarak çözüm getirilebilir. Yıllardır, nesne kenarı çıkarmak için Sobel, Prewitt, Laplacian kenar sezme yöntemleri kullanılmaktadır. Ama bunların hepsinde, kenar çıkarmada arzulanan nesne kenarının tek bir piksel kalınlığındaki çizgi ile ifadesi mümkün olamamıştır. Bu çalışmada geliştirilen kenar çıkarma algoritması arzu edilen bu sonucu sağlamaktadır.

Geliştirilen kenar çıkarma algoritmasında, nesnenin sınırlarını temsil eden çizgiler tek hat halinde elde edilebilmektedir. Sayısal ve iki renk (zemin siyah ve nesne beyaz gibi) halindeki nesne görüntüsüne aşağıda belirtilen kenar çıkarma algoritması uygulanarak nesne sınırları istenen biçimde elde edilir.

Bu algoritma; nesnenin her noktasını diğer komşu noktalar ile birlikte yorumlayarak, bu noktanın nesnenin kenarına ait olup olmadığını tespit eder. Şekil-4.1.a 'da nesnenin her hangi bir noktası (n_5) ve komşuları ($n_1, n_2, n_3, n_4, n_6, n_7, n_8, n_9$) gösterilmiştir.

n_1	n_2	n_3
n_4	n_5	n_6
n_7	n_8	n_9

(a)

0	1	0
1	n_5	1
0	1	0

(b)

Şekil-4.1. a) n_5 nesne noktasının komşuları, b) Geliştirilen kenar çıkarma şablonu.

Sayısal görüntüde, şayet nokta nesneye ait ise değeri "1", değil ise "0" olarak alınmak üzere şekil-4.1.b 'deki şablon, her bir nesne noktası ve komşularına uygulanarak, o noktanın nesnenin kenarına ait olup olmadığı belirlenir. Bu kenar çıkarma şablonunu klasik var/yok mantığı ile aşağıdaki gibi ifade edebiliriz.

$$Ç = n_2 * n_4 * n_6 * n_8$$

Yukarıdaki ifadenin sonucu “0” çıkarsa, n_5 noktasının nesnenin kenarına ait olduğu anlaşılır ve dolayısı ile n_5 noktasının değeri “1” olarak kalır. Şayet ifadenin sonucu “1” çıkarsa n_5 noktası nesne kenarına ait olmayıp, bu noktanın değeri “0” yapılır. Aşağıda, kenar çıkarma algoritmasının daha iyi irdelenmesi için iki örnek verilmiştir.

örnek-1 : Nesnenin incelenen kenar noktasının kenara ait olması durumunu ele alalım. Şöyle ki;

1 1 0 Burada, $n_2 = 1$, $n_4 = 1$, $n_6 = 0$, $n_8 = 0$ olup
 1 1 0 $\text{Ç} = 1 * 1 * 0 * 0 = 0$
 1 0 0

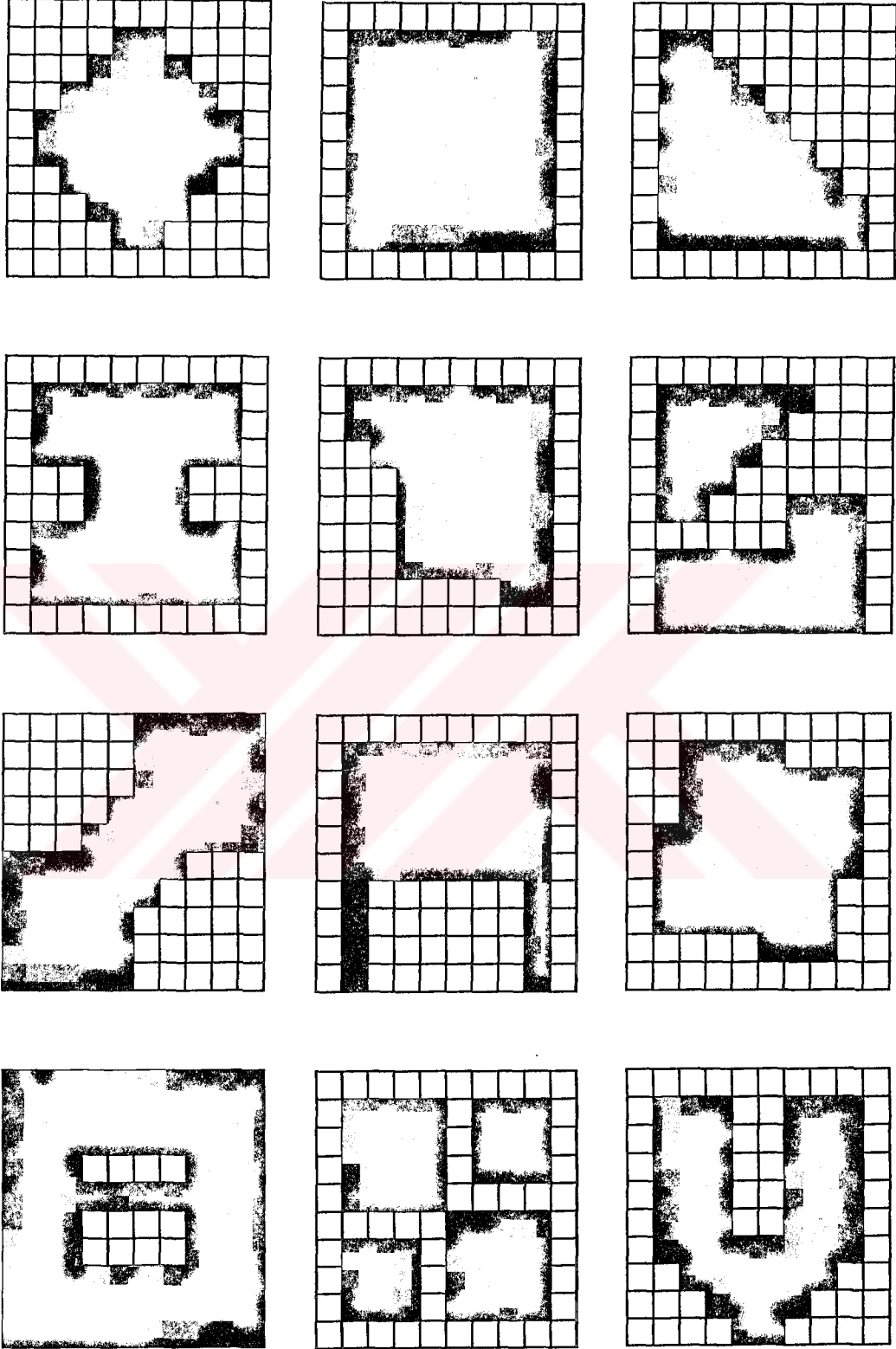
Sonuç “0” olduğu için n_5 noktasının kenara ait olduğu ve değerinin değiştirilmeyeceği anlaşılır.

örnek-2 : Nesnenin incelenen kenar noktasının kenara ait olmama durumunu inceleyelim. Bu durumda;

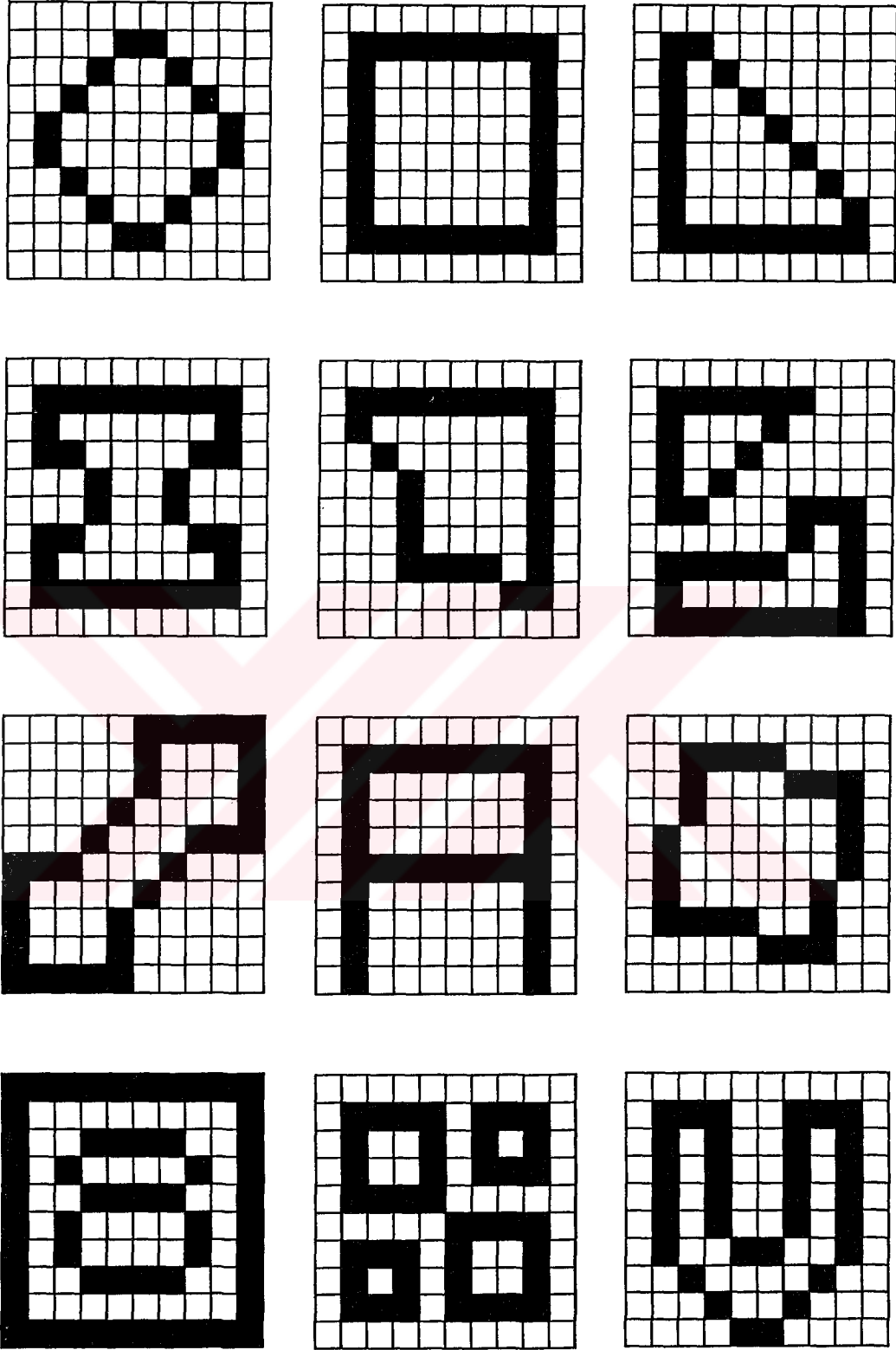
1 1 0 Burada, $n_2 = 1$, $n_4 = 1$, $n_6 = 1$, $n_8 = 1$ olup
 1 1 1 $\text{Ç} = 1 * 1 * 1 * 1 = 1$
 1 1 0

Sonuç “1” olduğu için n_5 noktasının kenara ait olmadığı, değerinin “0” olarak değiştirileceği anlaşılır.

Bu algoritmaya, giriş bilgisi olarak, şekil-4.2 'de görülen nesnelere girilmiş ve çıkış bilgisi olarakta şekil-4.3 'deki kenarları belirlenmiş nesnelere elde edilmiştir.



Şekil-4.2. Kenar çıkarma işlemi uygulanacak nesnelere.



Şekil-4.3. Kenar çıkarma işleminden sonra nesne görüntüleri.

Araştırmacılara göre kenar çıkarma algoritmalarında iki önemli özelliğin olması gerekmektedir (Marefat, 1990). Bunlar;

- Kenar çıkarma işleminin sonucu, kenar yönlerinden bağımsız olmalıdır. Yani çıkarılan kenar bilgisi, kenarı çıkarılan nesne yön değiştirdiğinde de değişime uğramamalıdır.
- Çıkarılan kenar çizgisi bir piksel kalınlığında olmalıdır.

Geliştirilen kenar çıkarma işleminde, nesnenin sayısal görüntü girişi temel alınmış ve algoritma bu temel üzerine kurulmuştur. Sisteme alınan sayısal görüntüye kenar çıkarma algoritmamızın sayısal filtre olarak ilave edilmesi ile kenar çıkarma problemine gerçek zamanda çözüm getirilebilir. Ayrıca geliştirilen algoritma, yukarıda bahsedilen kenar çıkarma algoritmalarının istenen iki temel özelliğini taşımakta olup, nesne tanıma işlemine özellikle zaman açısından büyük kolaylık getireceği aşikardır.

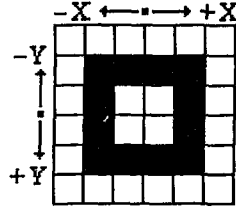
4.1.2. Kenar izleme

Nesne tanımda diğer bir önemli konu, nesnenin kenarlarını izlemektir. Böylece nesnenin kenarlarının yön değişiminden faydalanarak tanıma gerçekleştirilebilir. Şöyle ki; nesnenin herhangi bir kenar çizgisi başlangıç kabul edilerek nesne izlenir. Bu izleme sonucunda yaptığı yön değişiklikleri sayısı, veri tabanındaki model nesnelerin yön değişiklikleri ile karşılaştırılarak hangi tip nesne olduğu saptanabilir. Örneğin izlenen şekil bir üçgen olsun. Herhangi bir kenar noktası başlangıç noktası olarak alınır. Bu noktadan itibaren kenar izlendiğinde üç defa yön değişimi olur. Bu üç yön değişimi şeklin bir üçgen olduğunu belirlemede yardımcı olur. Kenar tarama işlemi, başlangıç olarak alınan noktaya ulaşıncaya kadar sürer.

Bu yöntem ile herbir nesne için elde edeceğimiz öznitelik vektörü sadece yön değişim sayısını içereceğinden, 1 elemana sahip olur. Dolayısıyla böyle bir tek elemana sahip öznitelik vektörünü yapay sinir ağına vererek sınıflandırma yapmak gereksizdir. Bu durumda nesne tanımak için klasik yöntemleri sınıflandırıcı olarak kullanmak daha doğru olacaktır.

Nesne kenarı izlemenin sağladığı en önemli bilgi, nesne sınırlarıdır. Bu sınır bilgileri, kullanılarak istatistiksel veya geometriksel öznitelik vektörleri oluşturulur.

Nesne kenarını doğru parçacıkları halinde izlemek için, Hough ve Rotasyonel dönüşümleri (Uzunalioglu, 1992) gibi çeşitli yöntemler olmakla beraber, aşağıda yapay sinir ağları ile nesne tanıma çalışması kapsamında özel olarak geliştirilen ve nesnenin kenarlarını noktasal biçimde izleyebilen algoritma verilmiştir. Şekil-4.4 'de kenarı izlenerek öznelik vektörü elde edilmek istenen nesnenin, matris şeklindeki bir veri tabanı üzerinde görünümü ve kenar izleme algoritması için yönler verilmiştir.



Şekil-4.4. Matris veri tabanında kenar izleme algoritması için yönler.

1. adım : Matris veri tabanı ilk satırdan itibaren $+X$ yönünde taranarak nesnenin ilk kenar noktası bulunur.

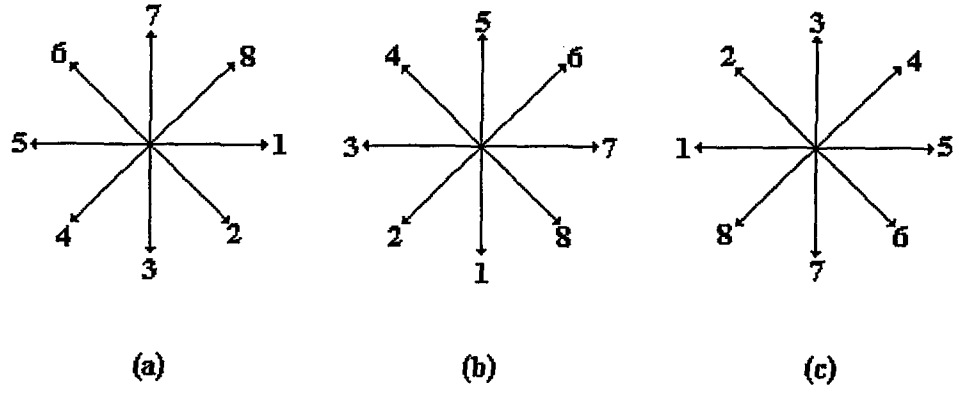
2. adım : Bulunan kenar noktasına komşu noktalar şekil-4.5.a 'da belirtildiği gibi takip edilerek taranır. Bu sıralı tarama esnasında ilk bulunan nokta, nesnenin devam eden kenar noktası olarak alınır.

3. adım : 2.adımda bulunan nesne kenar noktasının koordinatları bir önceki nesne kenar noktasına göre;

- $(+X, +Y)$ yönünde bir nokta ise tarama sıralaması şekil-4.5.a 'daki gibi alınır.
- $(-X, +Y)$ yönünde ise şekil-4.5.b 'deki tarama sıralaması esas alınır.
- $(+X, -Y)$ yönünde ise şekil-4.5.c 'deki verilen tarama sırası takip edilir.
- $(-X, -Y)$ yönünde ise şekil-4.5.b veya c 'deki tarama sırası kullanılır.

4.adım : Şayet hiçbir yönde devam eden kenar noktası yoksa nesnenin kenarında kopukluk vardır.

5.adım : Kenar çizgisi izleme işlemi, yukarıda belirtilen adımlar esas alınıp devam ettirilerek tekrar başlangıç noktasına gelinmiş ise başarı ile tamamlanmış olur.



Şekil-4.5. Herhangi bir kenar noktasının komşularını izleme sıralaması.

Şekil-4.5'de verilen kenar izleme yönlerinden hangisinin kullanılacağı, şekil-4.6. verilen komşuluk konumlarına göre belirlenir.

$(x-1,y-1)$	$(x,y-1)$	$(x+1,y-1)$
$(x-1,y)$	(x,y)	$(x+1,y)$
$(x-1,y+1)$	$(x,y+1)$	$(x+1,y+1)$

Şekil-4.6. Herhangi bir (x,y) noktasının komşularının konumları.

Nesnenin hangi yönde ilerleyeceğini tesbit etmek için, kullanılacak matematiksel ifadeler;

$$x = x + f(\cos \varphi)$$

$$y = y + f(\sin \varphi)$$

$$f(z) = \begin{cases} z < -0.5 & -1 \\ z > 0.5 & 1 \\ \text{diğer} & 0 \end{cases}$$

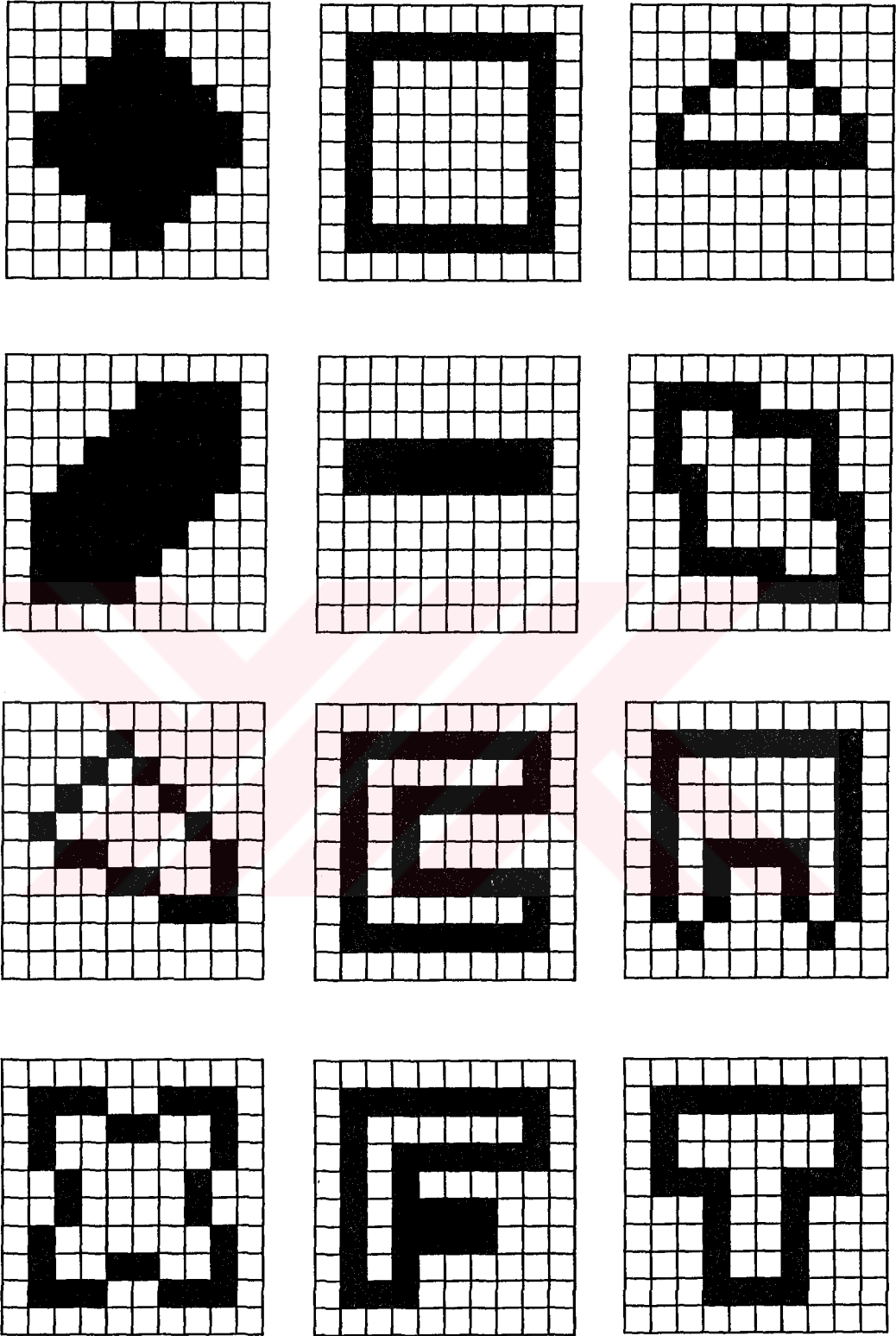
olarak tanımlanmıştır. φ açısının değişimi ise 45 derecelik adımlar halinde en fazla $\Delta\varphi = 360$ derece olabilmektedir. Yani;

- Şekil-4.5.a'daki sıralama takip edilecekse,
 $\varphi = 0; \varphi \leq 360; \varphi = \varphi + 45$.
- Şekil-4.5.b'deki sıralama takip edilecekse,
 $\varphi = 90; \varphi \leq 450; \varphi = \varphi + 45$
- Şekil-4.5.c'deki sıralama takip edilecekse,
 $\varphi = 180; \varphi \leq 540; \varphi = \varphi + 45$

şeklinde döngüler kullanılacaktır.

Geliştirilen bu kenar izleme algoritmasıyla nesnenin kenarlarından faydalanılarak oluşturulacak öznitelik vektörleri daha kısa sürede gerçekleştirilebilir. Çünkü, matris veri tabanının tüm noktalarını, nesne kenarı var mı, yok mu diye tarama yerine bu yöntemle doğrudan nesnenin kenarlarının olduğu noktalar taranmaktadır. Ayrıca bu kenar izleme yönteminde, nesnenin kenarlarını çıkarma işlemine gerek yoktur. Çünkü nesnenin sadece dış çizgileri izlenmektedir. Yine bu yöntem, nesnenin dönmesinden ötelenmesinden bağımsızdır. Bu algoritmanın dezavantajı ise, nesnenin içinde başka bir şekil olduğunda içteki şeklin kenarları, izlenememektedir. Ayrıca nesnenin kenarlarında herhangi bir kopukluk varsa kenar izleme işlemi yapılamamaktadır. Bir diğer kısıtlayıcı unsur ise, bir nesnenin başka bir nesne tarafından bir kısmının örtülmüş olması durumudur. Bu takdirde kenar izleme işleminde hata yapılabilir ve izleme işlemi yapılırken takip edilen nesneye ait değilse, diğer bir nesneye ait olan nokta bulunup, bu nesne üzerinde kenar izleme işlemine devam edilebilir. Bu ise anlamsız bir nesnenin bulunmasına neden olur.

Şekil-4.7 'de geliştirilen kenar izleme algoritmasının, yukarıda belirtilen kısıtlamalar dikkate alınarak başarı ile uygulandığı bazı nesnelere görülmektedir.



Şekil-4.7. Kenar izleme işlemi başarılı olmuş bazı nesnelere.

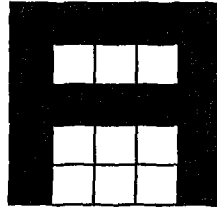
4.1.3. Öznitelik vektörleri

Yapay sinir ağları ile nesne tanımda kullanılan öznitelik vektörleri genel olarak, yerel özellikleri kullanan istatistiksel ve genel özellikleri kullanan geometriksel yöntemlerle çıkarılırlar. Aşağıda her iki yöntemle öznitelik vektörünün elde edilmesi verilmiştir.

4.1.3.1. İstatistiksel yaklaşımla öznitelik vektörünün elde edilmesi

İstatistiksel yaklaşım, nesne tanıma algoritmaları için temel teşkil etmektedir. Aynı sınıfa ait nesnelere, istatistiksel olarak tanımlanan benzer özelliklere sahiptir. İstatistiksel yaklaşımda, özniteliksel olarak nitelendirilen karakteristik ölçümler nesne örneklerinin yerel özelliklerinden çıkarılırlar. Her nesne bir vektör ile tanımlanır (Türkoğlu ve Arslan, 1995). Bu yöntem ile öznitelik vektörünün çıkarımına ilişkin aşağıda iki örnek sunulmuştur.

örnek-1 : Şekil-4.8 'de görülen 5x5 matris veri tabanı üzerinde bulunan bir "A" alfabetik örüntüsünün öznitelik vektörünün (V) çıkarımı;



Şekil-4.8. A alfabetik karakter örüntüsünün veri tabanı üzerinde gösterimi.

$$V_i = \begin{cases} 1 & \text{Kare dolu} \\ 0 & \text{Kare bos} \end{cases}$$

Fonksiyonunun tanımlanması dikkate alınarak "A" alfabetik örüntüsünün öznitelik vektörü,

$V = [1,1,1,1,1,0,0,0,1,1,1,1,1,1,0,0,0,1,1,0,0,0,1]$ şeklinde yazılabilir.

örnek-2 : Şekil-4.9'da gösterilen 5x5 matris veri tabanı üzerinde bulunan dikdörtgensel bir nesnenin öznitelik vektörünün çıkarımı;



Şekil-4.9. Dikdörtgen bir nesnenin veri tabanı üzerindeki görünümü.

Verilen nesnenin içi dolu (şekil-4.9.a) olduğu halde öznitelik vektörü;

$V = [0,1,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0]$ dir.

Bu şekildeki bir öznitelik vektörü verilen nesnenin tüm noktasal bilgilerini üzerinde taşır. Nesnenin içini geliştirdiğimiz kenar çıkarma algoritmasıyla boşalma işlemine tabi tuttukten sonra (şekil-4.9.b) bu nesnenin öznitelik vektörü ise;

$V = [0,1,1,1,0,0,1,0,1,0,0,1,0,0,1,0,0,1,0,0,1,1,0]$ dir.

Böyle bir öznitelik vektörü, verilen nesnenin sadece kenar bilgilerinden oluşmaktadır.

4.1.3.2. Geometriksele yaklaşım ile öznitelik vektörünün elde edilmesi

Geometriksele yolla öznitelik vektörünün çıkarımı için birçok yöntem mevcuttur. Bunların neler olduğu, daha önceki bölümlerde belirtilmişti. Bu tezde geliştirilen kenar çıkarma algoritmasından da faydalanarak, geometriksele yaklaşım ile nesnenin genel özellikleri kullanılarak öznitelik vektörü, aşağıda verilen adımlar sonucunda çıkarılabilir.

1.adım : Verilen nesne görüntüsü, kenar çıkarma işleminden geçirilerek nesnenin kenarlarının tek nokta kalınlığında olması sağlanır.

2.adım : Sadece kenarları kalan nesnenin, ağırlık merkezi (x_a, y_a) aşağıdaki gibi hesaplanır.

$$x_a = \frac{\sum_x \sum_y f(x, y) \cdot x}{\sum_x \sum_y f(x, y)} \quad y_a = \frac{\sum_x \sum_y f(x, y) \cdot y}{\sum_x \sum_y f(x, y)}$$

$$f(x, y) = \begin{cases} 1, & \text{Nesne kenarına ait ise} \\ 0, & \text{Nesne kenarına ait değilse} \end{cases}$$

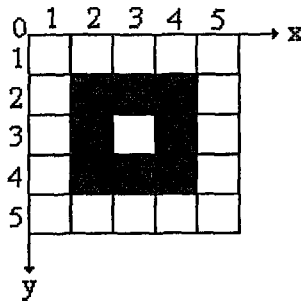
3.adım : Nesnenin her bir kenar noktasının (x_a, y_a) noktasına olan öklit uzaklığı, alta verilen matematiksel ifade kullanılarak bulunur.

$$\text{uzk}(i) = \sqrt{(x_i - x_a)^2 + (y_i - y_a)^2}$$

4.adım : Bulunan uzaklıklar büyükten küçüğe doğru (veya tersi) sıralanarak, verilen nesneye ait öznitelik vektörü çıkarılmış olur.

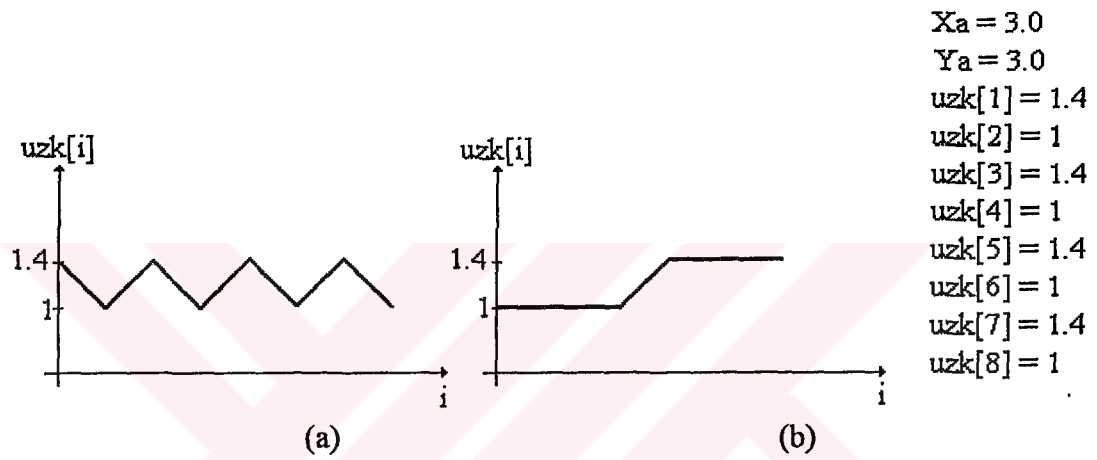
Bu yöntemi de, iki örnek üzerinde gösterelim:

örnek-1 : Şekil-4.10 'da matris veri tabanı üzerinde verilen, karesel bir nesne için, öznitelik vektörünün çıkarımı.



Şekil-4.10. Kare nesnesinin matris veri tabanı üzerinde gösterimi.

Şekil-4.11.a 'da verilen kare nesnesinin kenarlarının, nesnenin ağırlık merkezine göre değişimi görülmektedir. Bu değişimin sıralaması değiştirilmeden öznitelik vektörü oluşturulursa, elde edilen bu öznitelik vektörü nesnenin dönmesinden bağımsız olmaz. Bu şekilde öznitelik vektörü elde etmek yerine, şekil-4.11.a 'daki nesne kenar değişimi grafiğinden maksimum veya minimum noktalar sayılarak nesnenin kaç köşesinin olduğunu içeren bir öznitelik vektörü elde edilebilir. Böyle bir öznitelik vektörü için, yapay sinir ağı sınıflandırıcısı kullanmak gereksizdir.



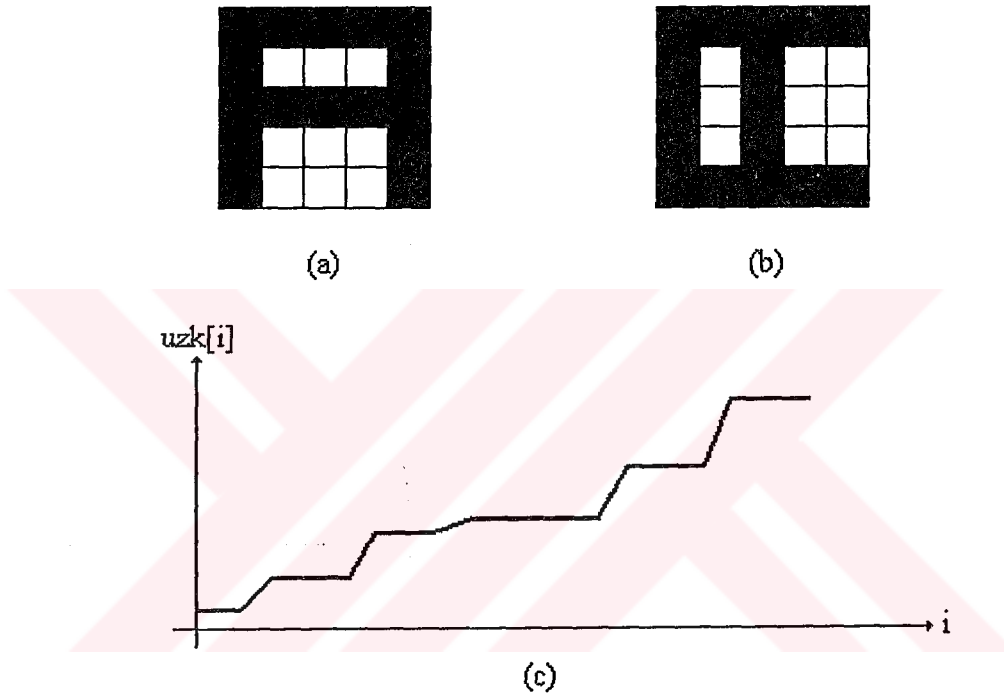
Şekil-4.11. Kare nesnesinin merkez-kenar değişimi (a,b).

Şekil-4.11.b 'de ise verilen kare nesnesinin kenarlarının ağırlık merkezine göre değişimi, hesaplanan uzunlukların küçükten büyüğe doğru sıralanmış sonucu çizilmiştir. Yani, nesnenin merkez-kenar uzaklıkları kendi aralarında bir sıralama işleminden geçirildikten sonra nesnenin öznitelik vektörü, oluşturulmuştur. Bu şekildeki bir öznitelik vektörü nesnenin dönmesinden, ötelenmesinden bağımsızdır ve yapay sinir ağı sınıflandırıcısı kullanılmasına elverişlidir.

örnek-2 : A alfabetik karakter örüntüsünün değişik görünümleri için merkez-kenar değişimi.

Şekil-4.12.a ve b' de "A" alfabetik örüntüsünün değişik görünümleri verilmiştir. Her iki görünüm için elde ettiğimiz merkez-kenar uzunluklarını sıraladıktan sonra

şekil-4.12.c 'deki gibi kenarların merkeze göre değişimi elde edilmiştir. Bu da göstermektedir ki, her iki görünümün de öznitelik vektörü aynıdır. Dolayısıyla sınıflandırıcı bu iki görünümün de tek bir nesne olduğunu belirlemelidir. Bu özellik robotik görmede ve nesne tanıma alanında oldukça önemlidir. Mesela; akan bir bant üzerinde görme temelli olarak çalışan bir robot kolu, gelen aynı tip nesnelerin konumları birbirine göre farklı olsa bile, hepsini başarıyla tanıyarak istenen işlevi yerine getirebilir.



Şekil-4.12. A alfabetik karakter örüntüsünün farklı görünümleri (a,b) ve merkez-kenar grafiği (c).

4.2. Geçekleştirilen Yapay Sinir Ağları ile Sınıflandırma

Yapay sinir ağları nesne tanıma alanında, özellikle bilgisayar teknolojisinin ve yapay sinir ağ modellerinin geliştirilmesi ile kullanılmaya başlanmıştır. Yapay sinir ağları sınıflandırıcıları, nesnelerin tanıma modellerini içeren öznitelik vektörünü giriş olarak alan ve çıkış ünitelerinin birinde cevap üreten, çok sayıda doğrusal olmayan işlem elemanlarının birleşimidir. Herbir düğüm noktasının içerdiği bilgi, ağırlık vektörleri ile

gösterilir. İstenen çıkış için uygun ağırlık vektörlerinin bulunması, öğrenme olarak tanımlanır. Yapay sinir ağı sınıflandırıcılarının her çıkış ünitesi, gözlenen olayın farklı bir sınıfını temsil eder. Yapay sinir ağlarını esas alan pekçok sınıflandırma algoritması geliştirilmiştir. Bunlardan bazıları; Hopfield Net, Hamming Net, Perceptron, Carpenter-Grossberg sınıflandırıcısı, Hata Geri-Yayımla Algoritması ve Kohonen 'in kendi kendini düzenleyen öznelik haritalarıdır.

Bu çalışmada, eğitici bir yapay sinir ağı olan Kohonen'in kendi kendini düzenleyen öznelik haritaları ile hatanın geriye yayılma algoritmasını kullanan eğitici bir yapay sinir ağı (geriye yayılım ağı) kullanılmıştır. Her iki yapay sinir ağ modeli, geliştirilen öznelik vektörü elde etme yöntemleri ile birlikte kullanılarak, nesne tanımadaki başarıları karşılaştırılmıştır. Aşağıda her iki yapay sinir ağ modeli ayrı ayrı incelenip, daha sonra uygulamaları verilecektir.

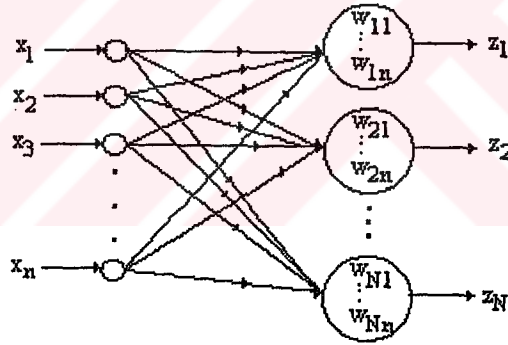
4.2.1. Kohonen 'in kendi kendini düzenleyen öznelik haritaları sınıflandırıcısı

Kendi kendini düzenleyen öznelik haritaları, eğitici olmayan durumda giriş sinyallerinin analizi ve kodlanması için otomatik bir yöntemden meydana gelen yapay sinir ağlarıdır. Kendini düzenleme algoritması, beyinde bulunan haritalara benzer. Beyindeki duyu yollarının önemli bir düzenleme prensibi, nöronların yerleşmesinin sıralı olması ve algılanan dış uyarımın bazı fiziksel özelliklerini yansıtmalarıdır. Mesela, duyma yolunun her seviyesinde, sinir hücreleri ve fiberler, belli frekanslara en yüksek cevabı veren nöron dikkate alınarak anatomik yapıda düzenlenmiştir. Duyma yolundaki bu düzenleme, beyin duyma kabuğuna kadar gitmektedir. Beyinde alt seviye düzenlemesinin büyük bir kısmı genetik olarak önceden belirlenmiş olmasına rağmen, daha yüksek seviyedeki düzenlemenin bir kısmının kendi kendini düzenlemeyi sağlayan algoritmalar tarafından, öğrenme esnasında meydana geldiği düşünülmektedir. Kohonen, beyine benzer, kendi kendini düzenleme öznelik haritası adını verdiği bir ağ modeli sunmuştur.

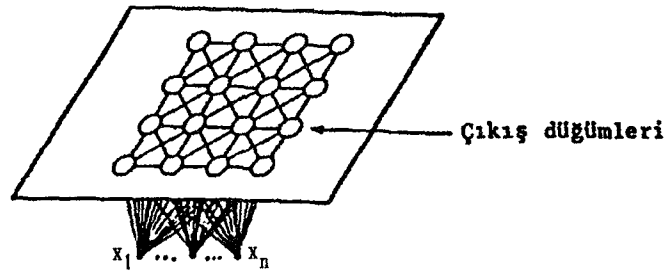
Kohonen yapay sinir ağının, giriş verisinin bir haritasını gösterdiği söylenebilir. Bu yapay sinir ağında yapısal olarak benzer girişler, yapısal olarak birbirine yakın çıkışlar üretir. Kohonen algoritması, yarışmacı öğrenme (competitive learning) algoritması olarak da adlandırılır. Yarışmacı öğrenme, giriş örüntü setini sınıflandırmak için kullanılabilen

belirgin ve genel özellikleri keşfetmek için bir yol sağlar. Yarışmacı kategoriye bağlı öğrenme kurallarının hepsinde ortak özellik, bir yarışma stratejisinin işlem elemanlarının tamamına veya bir kısmına uygulanmasıdır. Bu yarışma sonucu, kazanan işlem elemanının ağırlıkları değiştirilir (Türkoğlu ve Arslan, 1996).

Basit işlem elemanlarından oluşan Kohonen yapay sinir ağı, şekil-4.13 'de gösterildiği gibidir. Burada N çıkış işlem elemanı (z_1, z_2, \dots, z_N) sayısı, $x_i = (x_1, x_2, \dots, x_n)$ n boyutlu giriş örüntü vektörü, $w_i = (w_{z1}, w_{z2}, \dots, w_{zn})$ her çıkış işlem elemanının ağırlık vektörü olup, w_{ij} ise x_j girişi ile i . işlem elemanı arasındaki ağırlıktır. Ağırlık vektörünün başlangıç değerleri keyfi seçilebildiği gibi, ilk giriş örüntü vektörü ağırlıklar olarak alınabilir. Temelde Kohonen yapay sinir ağı, her çıkış birimine (bu birimlere sınıf da diyebiliriz) giriş vektörünün bir elemanının bağlanması ile meydana gelir. Böylece farklı çıkış birimleri farklı girişlere duyarlı hale gelebilir. Kohonen yapay sinir ağının yapısal sistem modeli şekil-4.14 'de gösterilmiştir. Şekilde de görüleceği üzere Kohonen yapay sinir ağı sadece iki katmana, giriş ve çıkış katmanına sahiptir.



Şekil-4.13. Kohonen 'in kendi kendini düzenleme haritasının ağ birimleri.

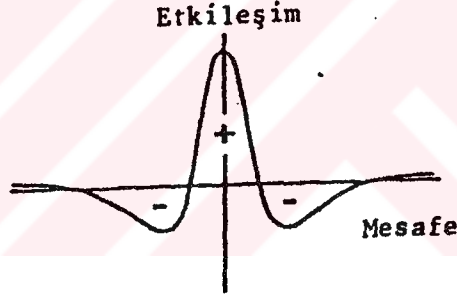


Şekil-4.14. Kohonen sistem modeli.

Giriş sinyalleri doğal nesnelere gözlenmesinden türetildiği için, giriş örüntü vektörü ayrık zamanda $x = x(t)$ olarak formülize edilir. Buna karşılık $w_i(t)$ ise farklı işlem elemanlarının zamana bağlı değişken ağırlık vektörleridir.

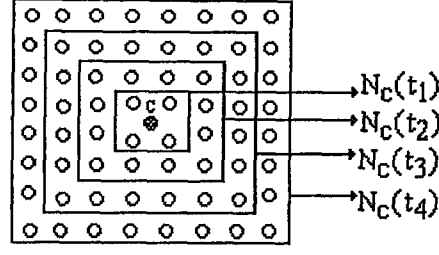
Bir biyolojik sinir sisteminin dış uyarıya cevabı, tek bir sinir hücresi üzerinde yoğunlaşmış olmasına rağmen, çok sayıda geri besleme bağlantısından oluşan oldukça karmaşık etkileşimler ile belirlenir. Yani komşu sinir hücreleri arasında yerel bir etkileşim söz konusudur.

Benzer şekilde Kohonen yapay sinir ağına, gelen giriş örüntü vektörüne cevabı, çıkış katmanındaki herhangi bir işlem elemanı üzerinde yoğunlaşmış olmasına rağmen, fazla sayıda geri besleme bağlantısı sonucu oluşan etkileşimler ile sistemin çıkışı belirlenir. Komşu işlem elemanları arasında yerel bir etkileşim vardır. En yakın komşular birbirini pozitif yönde etkilerken, birbirinden uzak komşuluktaki işlem elemanları birbirini negatif olarak etkilerler. Bu tip etkileşime "meksika şapkası fonksiyonu" denir (şekil-4.15).



Şekil-4.15. Meksika şapkası fonksiyonu.

Kohonen yapay sinir ağı, iki boyutlu bir yapıya sahiptir. İki boyutlu ağ içindeki etkileşimlerin yapısı, hangi işlem elemanlarının komşu olduklarını belirler. Ağırlık vektörünün başlangıçtaki keyfi dağılımı, giriş örüntü vektörlerinin ağ tarafından öğrenilmeye başlanması ile kendi kendine ayarlanır. Giriş sinyaline karşı istenen en iyi cevabı veren işlem elemanı etrafında, bir $N_c(t)$ komşuluk kümesi oluşur. Bu komşuluk kümesi iki boyutlu düzlem üzerinde, dikdörtgen vb. yapılarda oluşabilir (şekil-4.16).



Şekil-4.16. Bir işlem elamanının yapısal komşuluğu ($t_1 < t_2 < t_3$).

Herhangi bir çıkış işlem elamanının, verilen bir dairesel yarıçap ile bütün komşulukları belirlenebilir.

Kohonen 'in kendi kendini düzenleyen haritaları tarafından incelenen nesnelere, x vektörü ile simgelenmektedir. Bu x vektörü, giriş nesnesinin özneliklerini içeren vektör olup, iki boyutlu düzlemsel çıkış uzayında haritalandırılır. Sonuçta benzer nesnelere iki boyutlu düzlemsel çıkış uzayında birbirine yakın yerlere yerleşirler. Böylece yapay sinir ağının öğrenme işlemi, eğiticiye bir şekilde sağlanmış olur. Öğrenme gerçekleştirildikten sonra benzer giriş nesnelere, çıkış düzleminde birbirine yakın haritalandığı gözlenebilir. Buna karşılık birbirine benzemeyen giriş nesnelere birbirinden uzağa haritalanırlar.

Aşağıda Kohonen'in kendi kendini düzenleyen öznelik haritasını sağlayan algoritmanın işlem adımları verilmiştir :

1. adım : N tane giriş düğümlerinden M tane çıkış düğümlerine olan ağırlıklar küçük rastgele değerlere ayarlanır ve komşuluğun ilk yarıçapı (eşik) değeri belirlenir.
2. adım : Yeni giriş uygulanır.
3. adım : Girişin, tüm çıkış düğümlerine (sınıflara) olan mesafesi hesaplanır.

$$d_j = \sum_{i=1}^N (x_i(t) - w_{ij}(t))^2$$

d_j : Giriş ile j . çıkış düğümü arasındaki mesafe,

$x_i(t)$: x giriş vektörünün i . elemanının t anındaki değeri,

$w_{ij}(t)$: i. girişten j. çıkış düğümüne olan ağırlığın t anındaki değerini göstermektedir.

4. adım : Minimum mesafeye sahip çıkış düğümü belirlenir. En küçük d_j mesafesine sahip j. çıkış düğümü seçilir.

5. adım : j. çıkış düğümüne ve komşularına ait ağırlıklar yeniden hesaplanır.
Yeni ağırlıklar;

- yarışmayı kazanan çıkış düğümleri için,

$$w_{ij}(t+1) = w_{ij}(t) + \alpha(t) \cdot (x_i(t) - w_{ij}(t)) \quad \forall i \in Nc(t)$$

- yarışmayı kaybeden çıkış düğümleri için,

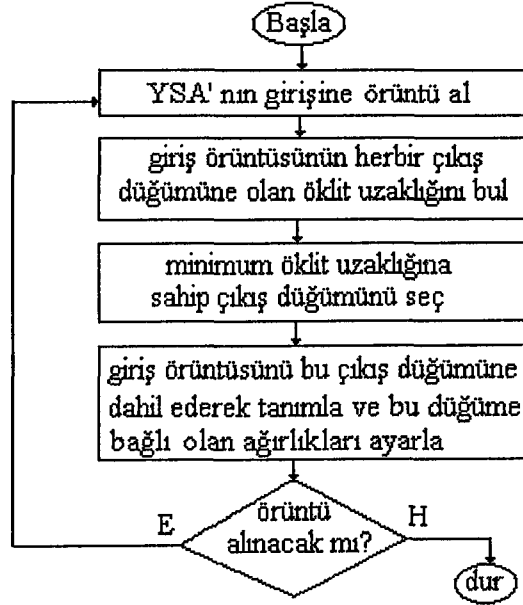
$$w_{ij}(t+1) = w_{ij}(t) \quad \forall i \notin Nc(t) \quad \text{olur.}$$

$\alpha(t)$: zamanla azalan bir kazanç terimi. ($0 < \alpha(t) < 1$)

$Nc(t)$: zamanla değişen komşuluk kümesi.

6. adım : 2. adıma gidilerek işlemler tekrarlanır.

Nesne tanıma amaçlı olarak yukarıda verilen algoritmanın, Borland C programlama dili kullanılarak bilgisayar benzetimi yapılmıştır (Türkoğlu, 1996). Nesnelere, bilgisayar ekranında oluşturulan bir matris veri tabanı üzerinde fare kullanılarak simüle edilebilmektedir. Öyle ki geliştirilen bilgisayar programı, kameradan alınmış sayısal bir görüntünün bu matris veri tabanı üzerine yerleştirilerek işlenmesine olanak sağlayabilecek niteliktedir. Program, nesne çeşitleri açısından genel olup, hangi tür nesne bu veritabanı üzerinde oluşturulursa oluşturulsun onunla çalışmaya elverişlidir. Kohonen yapay sinir ağını eğitmek ve nesne tanımak için geliştirilen bilgisayar programı şekil-4.17 'deki akış şemasını takip etmektedir.



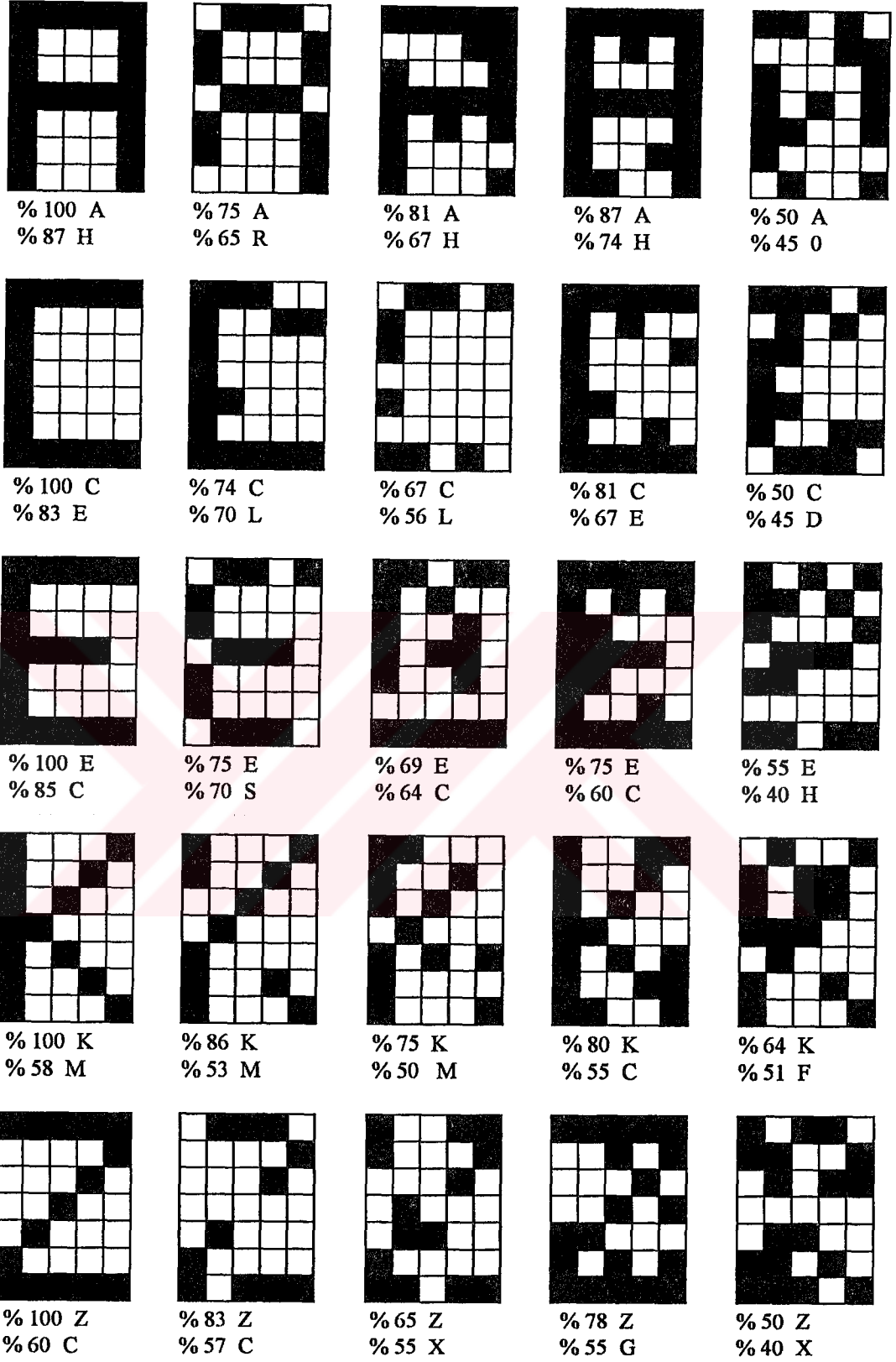
Şekil-4.17. Kohonen yapay sinir ağını eğitmek ve nesne tanımak için geliştirilen programın akış şeması.

Burada örnek olması amacıyla, nesne olarak alfabetik karakter örüntüleri ve geometriksel şekiller alınmıştır. Matris veri tabanı üzerinde bulunan nesnelerin öznelik vektörleri bölüm 4.1.3 'de söz edilen her iki şekilde de ayrı ayrı elde edilerek Kohonen yapay sinir ağına verilmiştir. Ağa, ilk önce nesnelerin bozuk olmayan halleri verilerek bunları öğrenmesi %100 oranında sağlanmıştır. Daha sonraki aşamada öğrenilen nesnelerin, bozuk biçimleri verilerek bunları tanıma oranları yüzde olarak hesaplanmıştır. Şekil-4.18 ve şekil-4.19 'da görüleceği üzere, ilk sütunda nesnelerin bozuk olmayan halleri, sonraki sütunlarda ise her bir nesnenin bozuk biçimi mevcuttur. Her nesnenin altında ise, o nesnenin yapay sinir ağına öğretilen nesneler içinde en çok benzedikleri yüzde olarak verilmiştir.

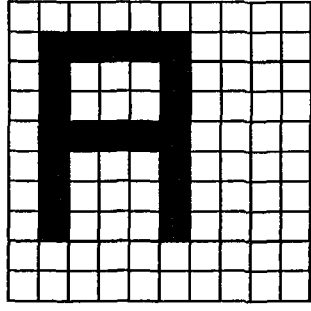
Şekil-4.18' de alfabetik karakter örüntülerinin öznelik vektörü istatistiksel yöntemle (bkz. bölüm 4.1.3.1.) elde edilerek yapay sinir ağı eğitilmiş ve örüntü tanıma gerçekleştirilmiştir. Komşuluk yarıçapı ise yapılan denemeler sonunda ancak 2 olarak seçildiğinde, her bir alfabetik karakter yapay sinir ağına çıkışında bir sınıf oluşturabilecek şekilde birbirinden ayrılabilmiştir.

Şekil-4.19.a 'daki karekter örüntülerinin ve şekil-4.19.b 'de görülen geometriksel nesne şekillerinin öznelik vektörleri, geliştirilen geometriksel yöntemle (bkz. bölüm 4.1.3.2.) elde edilerek, yapay sinir ağı eğitilmiş ve nesne tanıma gerçekleştirilmiştir. Komşuluk yarıçap değeri ise yine yapılan denemeler sonucunda şekil-4.19.a 'daki nesneler için 0.01 ve şekil-4.19.b 'dekiler için ise 0.5 olarak seçildiğinde, yapay sinir ağına öğretilen herbir nesne gurubunun sınıfları belirlenebilmiştir.

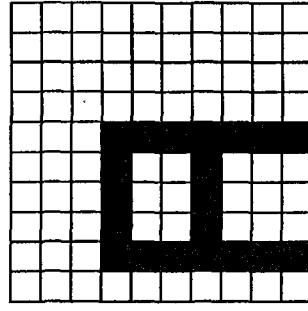




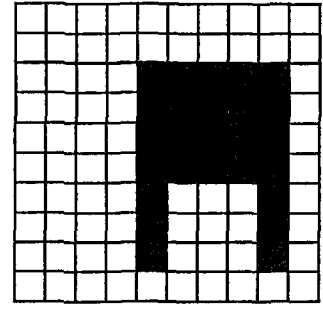
Şekil-4.18. Alfabetik karakter örüntülerinin Kohonen yapay sinir ağı tarafından istatistiksel olarak tanınma yüzdeleri.



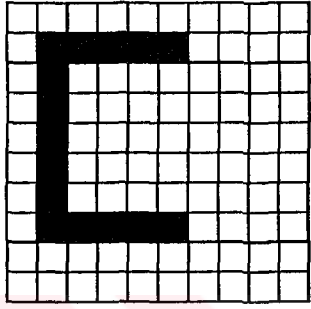
% 100 A
% 95 B



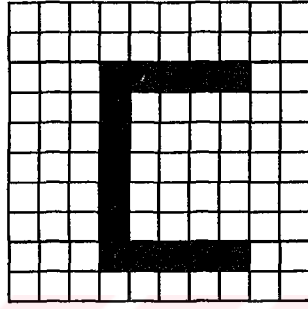
% 100 A
% 95 B



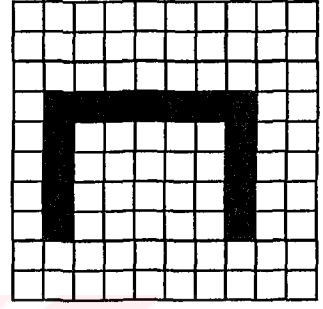
% 100 A
% 95 B



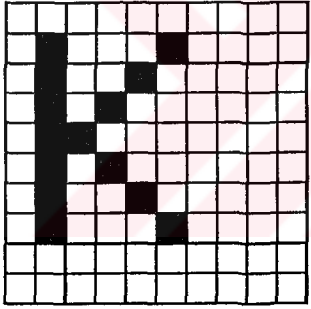
% 100 C
% 98 Z



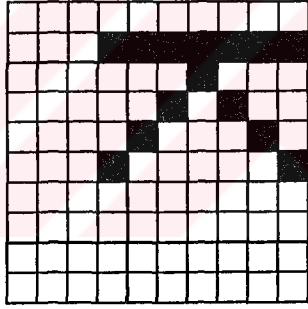
% 100 C
% 98 Z



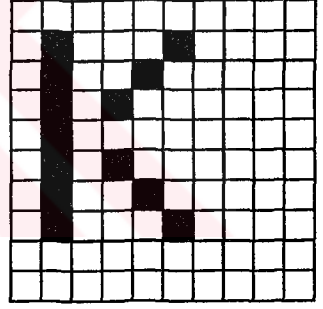
% 100 C
% 98 Z



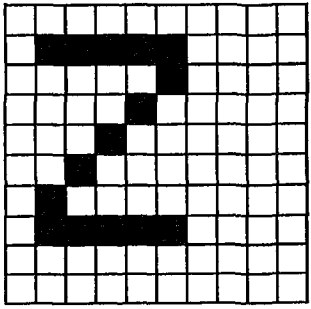
% 100 K
% 82 S



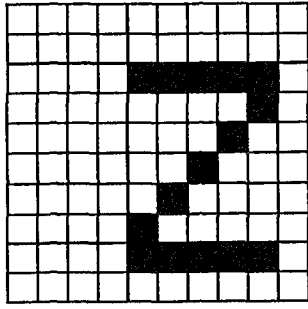
% 100 K
% 82 S



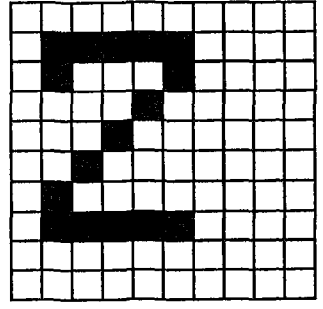
% 79 K
% 99 F



% 100 Z
% 98 C

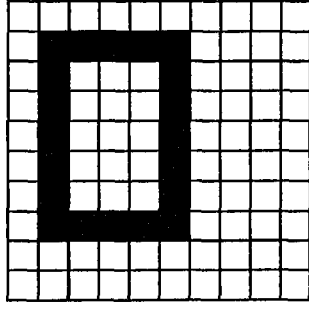


% 100 Z
% 98 Z

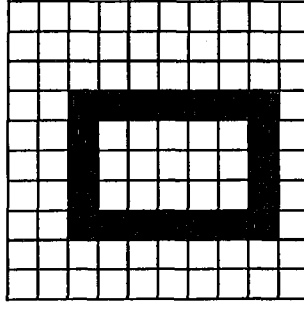


% 74 Z
% 93 D

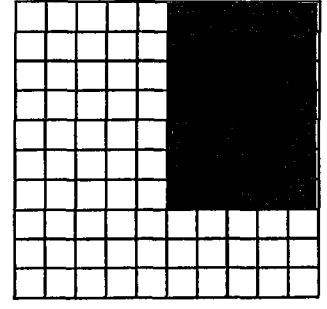
Şekil-4.19.a. Alfabetik karakter örüntülerinin Kohonen yapay sinir ağı tarafından geometriksel olarak tanıma yüzdeleri.



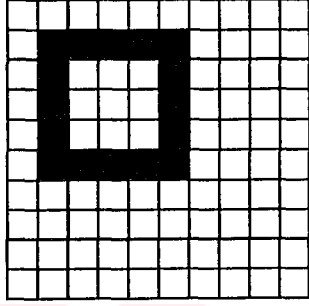
% 100 dikdörtgen
% 44 kare



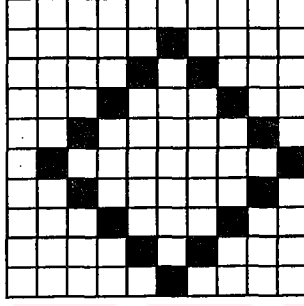
% 100 dikdörtgen
% 44 kare



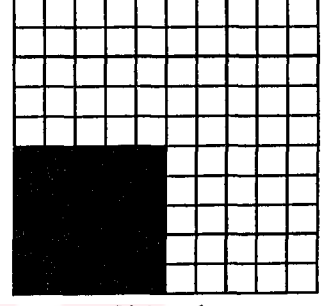
% 100 dikdörtgen
% 44 kare



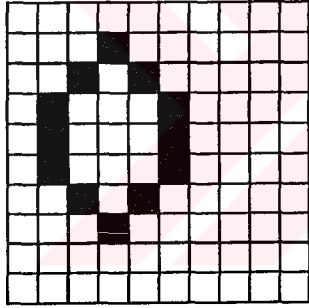
% 100 kare
% 40 altıgen



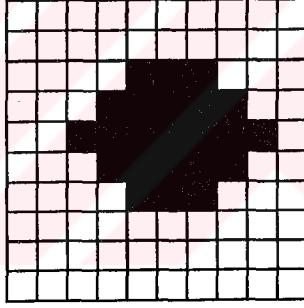
% 82 kare
% 28 dikdörtgen



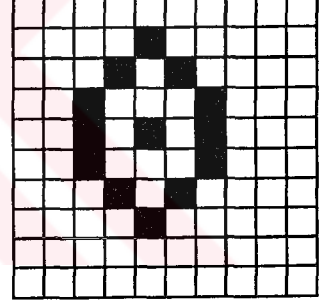
% 100 kare
% 40 altıgen



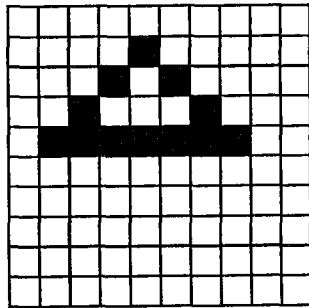
% 100 altıgen
% 97 üçgen



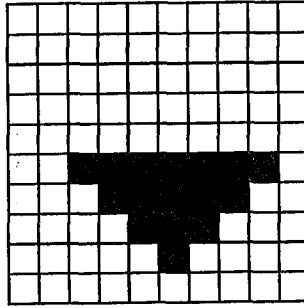
% 100 altıgen
% 97 üçgen



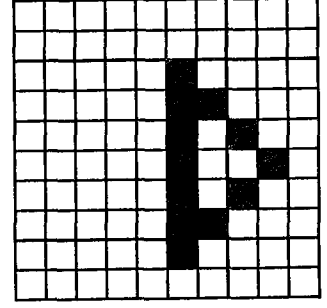
% 85 altıgen
% 86 üçgen



% 100 üçgen
% 97 altıgen



% 98 üçgen
% 93 altıgen



% 98 üçgen
% 93 altıgen

Şekil-4.19.b. Geometrik nesnelerin Kohonen yapay sinir ağı tarafından geometriksel olarak tanınma yüzdeleri.

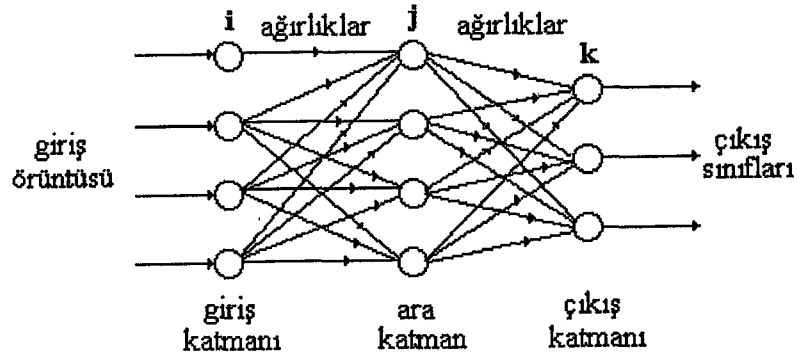
Yukarıdaki benzetim sonuçları, eğiticiyiz bir yapay sinir ağı olan Kohonen 'in kendi kendini düzenleyen öznitelik haritalarının sınıflandırıcı olarak kullanıldığı ve nesnelerin öznitelik vektörlerinin, istatistiksel ve geometriksel olmak üzere iki farklı şekilde çıkarıldığı, nesne tanıma sisteminden elde edilmiştir.

Şekil-4.18 'de herbir alfabetik karakter örüntüsünün bozuk olmayan şekli, yapay sinir ağına öğretilip, sonra herhangi bir nedenden dolayı bu örüntülerde meydana gelecek olan bozulma durumunda eğiticiyiz yapay sinir ağının bu örüntüleri tanıma başarısı ölçülmüştür. Alfabetik karakter örüntüleri, eğiticiyiz yapay sinir ağına çok kısa bir sürede %100 olarak öğretilmiş ve bu öğretilen örüntülerde %40 'lara varan bozulma olduğu halde, eğiticiyiz yapay sinir ağı yinede doğru tanıma yapabilmıştır.

Şekil-4.19 'da nesne olarak alfabetik karakterler ve geometriksel şekiller alınmıştır ve her bir nesnenin öznitelik vektörü, geliştirilen geometriksel yöntemle bulunmuştur. Dolayısıyla nesnelerin genel özellikleri kullanılmıştır. Bu yöntem, nesnelere dönmeden ve konum değiştirmeden bağımsız olarak kesin bir şekilde tanıyabilmekte, buna karşılık nesneleredeki küçük bir yerel bozulma, nesnenin yanlış tanınmasına neden olabilmektedir.

4.2.2. Geriye yayılım ağı sınıflandırıcısı

Geriye yayılım ağı, çok katmanlı ileri yön bağlamalı bir ağıdır. Giriş ile çıkış katmanları arasında, birden fazla katmanın kullanıldığı yapay sinir ağı sistemidir. Gizli katman olarak isimlendirilen bu ara katmanlarda, aracısız giriş alamayan ve aracısız çıkış veremeyen işlem elemanları vardır. Şekil-4.20 'da çok katmanlı algılayıcı (Multi Layer Perceptron, MLP) yapay sinir ağının genel yapısı verilmiştir.



Şekil-4.20. Çok katmanlı yapay sinir ağı.

Çok katmanlı ağlarda veriler, giriş katmanı tarafından kabul edilirler. Ağ içindeki işlemler sonucunda çıkış katmanında oluşan cevap ile giriş verisine karşı istenen cevap karşılaştırılır. Hesaplanan cevap ile istenen cevap arasında herhangi bir farklılık varsa, katmanlar arasında bulunan ağırlıklar bu farkı azaltacak şekilde yeniden düzenlenir. Yapay sinir ağının girişindeki veriler, ağırlıklar uygun noktaya ulaşana kadar değişmez. Hesaplanan çıkış cevapları, istenen cevaplarla karşılaştırılarak, sonuçta hata işareti elde edilir. Hata işareti, ara katmanlardan çıkış katmanına olan ağırlıkları değiştirmekte de kullanılır. Fakat bunu yaparken, giriş katmanından ara katmana gelen ağırlığın değiştirilip değiştirilemediğini düşünmek gerekir. Ara katmanlarda ne tür bir çıkış istendiği bilinmeyeceği için, ara katmanların çıkışında hata işaretinin verilmesi kolay değildir. Bunun yerine her bir işlem elemanın, çıkış işlem elemanının hatalarına olan etkisi bilinmelidir. Bu hatalı birim için, gizli birime bağlı olan çıkış birimlerinin hata işaretlerinin ağırlıkları toplamı alınarak ayarlama yapılır. Çok sayıda ara katmana sahip sistemlerde her katmanın hata işaretleri, bir önceki katmanın düzeltilmiş hata işaretlerinden çıkartılarak işlem tekrarlanır. Sonuç olarak ağırlık düzeltme işlemi çıkış seviyesine bağlı ağırlıklardan başlar ve işlem ters yönde giriş seviyesine varana kadar devam eder. Sistem, sonuçta yine hatalar yapar, fakat bu hatalardan birşeyler öğrenip istenen cevabı bulana kadar işlem tekrarlanır. Bu yöntem hatanın geriye yayılma algoritması denir. Aşağıda bu algoritma ayrıntılı olarak verilmiştir.

Hatanın geriye yayılma algoritması, karesi alınmış hata fonksiyonunu minimize eden bir algoritma olup, genelleştirilmiş delta kuralını eğitime işlemi için kullanır. Sistemin ana yapısı, şekil-4.21 'de verilmiştir.

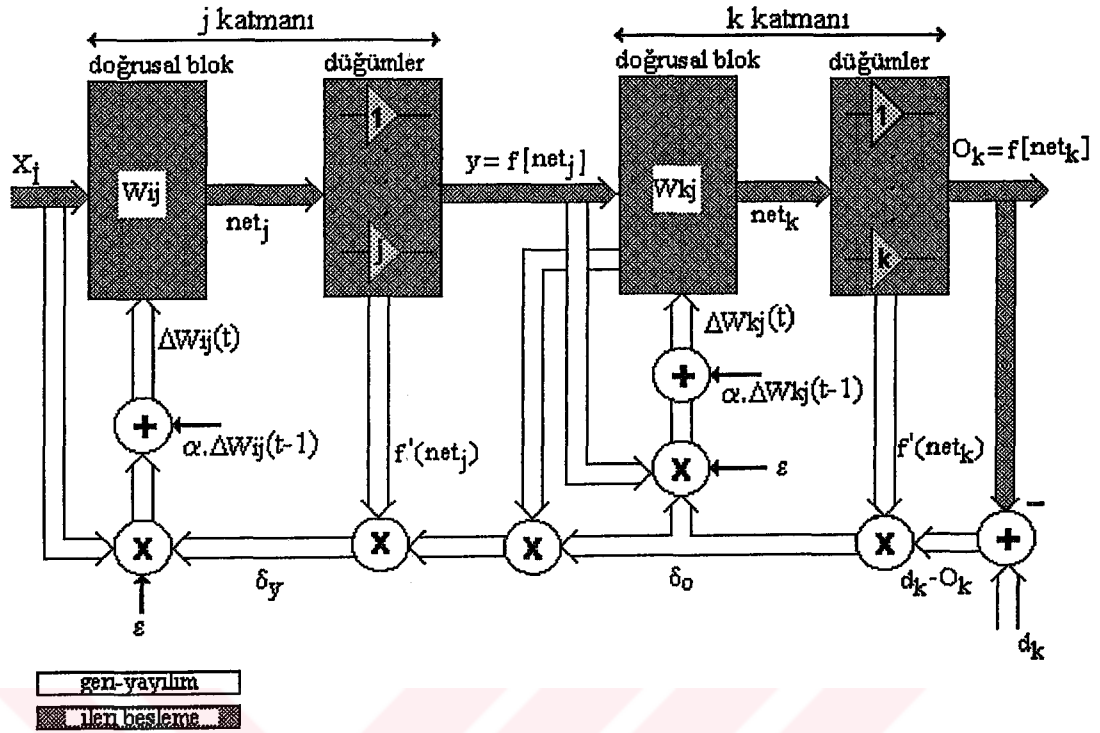
Burada; $x_i = (x_1, x_2, \dots, x_n)$ i. giriş örüntü vektörü, $w_{ij} = (w_{11}, w_{12}, \dots, w_{1n})$ i. giriş ile j. ara katman işlem elemanı arasındaki ağırlık vektörü, benzer şekilde w_{kj} ara katman ile çıkış katmanı arasındaki ağırlık vektörüdür. Herbir nöronun çıkışı,

$$f(\text{net}_j) = f(x_1 \cdot w_{11} + x_2 \cdot w_{12} + \dots + w_{1n} \cdot x_n)$$

şeklinde hesaplanır. $f(\cdot)$ eşik fonksiyonu olup, hata geriye yayılma algoritmasında genellikle

$$f(\text{net}) = \frac{1}{1 + e^{-\text{net}}}$$

sigmoid fonksiyonu kullanılır.



Şekil-4.21. Hatanın geriye yayılma algoritmasını kullanan yapay sinir ağının blok şeması.

Hatanın geriye yayılma algoritması ile yapay sinir ağını eğitme için işlem adımları;

1. adım : Tüm ağırlıkların ilk değerlerini belirle.

Tüm ağırlıkları küçük rastgele değerlere ayarla.

2. adım : Girişi ve istenen çıkışı uygula.

Giriş $x_i = (x_1, x_2, \dots, x_n)$ vektörünü uygula ve buna karşılık istenen çıkış

$d_i = (d_1, d_2, \dots, d_m)$ vektörünü belirle.

3. adım : Gerçek çıkışları hesapla.

En son katmandaki birimlerin çıkışlarını belirleyebilmek için, her katmandaki birimlerin çıkışlarını sırasıyla belirle ve sakla.

4. adım : Ağırlıkları ayarla.

Çıkış katmanındaki işlem elemanlarından ilk katmandaki işlem elemanlarına kadar, geriye doğru çalışan bir yinelemeli algoritma kullanılacaktır. Ağırlıkları aşağıdaki denkleme göre ayarla;

$$w_{ij}(t+1) = w_{ij}(t) + \varepsilon \cdot h_j \cdot y_i$$

Burada $w_{ij}(t)$, t. adımda j. işlem elemanına ya bir i. ara katman işlem elemanından ya da sabit bir girişten gelen ağırlıktır. Buna göre y_i i. katmanın çıkış veya sabit giriş vektörünü gösterir. h_j j. katmandaki hata işaretidir. Eğer j bir çıkış katmanı ise,

$$h_j = (d_j - y_j) \cdot y_j \cdot (1 - y_j) \quad \text{olur. } j \text{ bir ara katman ise,}$$

$$h_j = y_i \cdot (1 - y_i) \cdot \sum_k h_k \cdot w_{jk} \quad \text{ifadesi kullanılır. } k, j. \text{ ara katmandan sonra gelen}$$

katmandaki her işlem elemanı üzerinde çalışır. Aşağıdaki öğrenme denklemi kullanılırsa yakınsama daha hızlı olabilir.

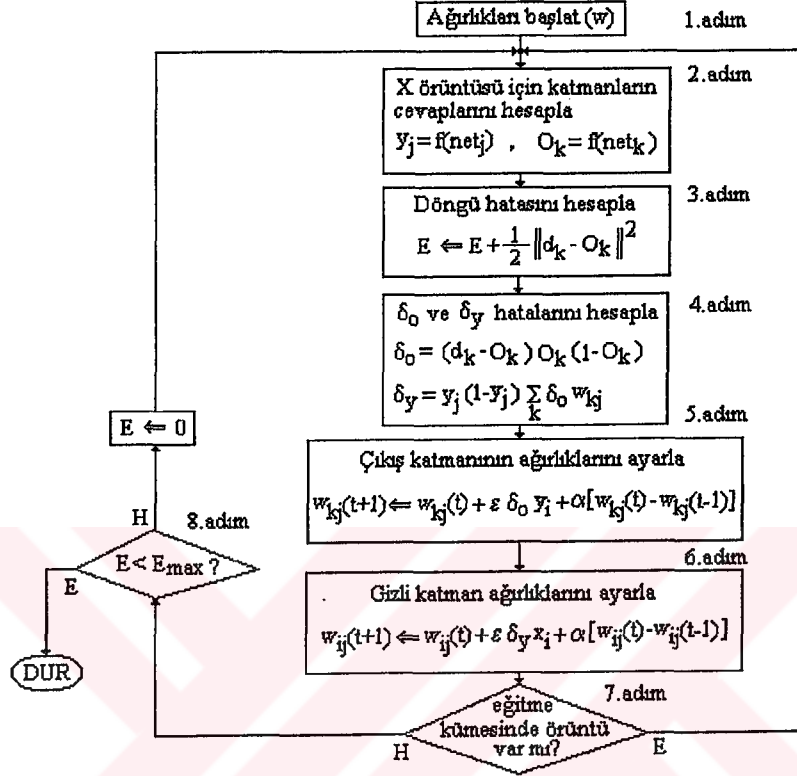
$$w_{ij}(t+1) = w_{ij}(t) + \varepsilon \cdot h_j \cdot y_i + \alpha \cdot (w_{ij}(t) - w_{ij}(t-1))$$

Burada, $0 < \alpha < 1$ ve $0.01 < \varepsilon < 10$

5. adım : 2. adıma giderek tekrarla.

Yapay sinir ağlarının öğrenme yeteneğini iyileştirmek için ε ve α katsayıları kullanılır. Bunlar, sırasıyla öğrenme ve momentum katsayıları olup yapay sinir ağının eğitilmesinde önemli rol oynarlar. Ağırlıkların değişiminin büyük değerlerde olması, sistemin davranışının bozulmasına neden olur ve sistem osilasyon yaparak kararlılığa ulaşamaz. Bunu önlemek için öğrenme katsayısı küçük tutulur ($0.01 < \varepsilon < 10$). Öğrenme katsayısının çok küçük bir değerde seçilmesi ise öğrenme işleminin yavaşlamasına yol açar. Momentum katsayısı ($0 < \alpha < 1$) bu noktadan hareketle ortaya atılmıştır. Bu katsayı, ağırlıktaki değişimi önceki değişimin değerine bağlayan bir katsayıdır. Böylece, daha düşük ε değeri ile daha hızlı öğrenme işleminin gerçekleştirilmesine olanak sağlanır.

Yukarıda belirtilen işlem adımlarını şekil-4.21 'deki sistemi dikkate alarak, akış şeması halinde verelim (şekil-4.22).



Şekil-4.22. Hatanın geriye yayılma algoritmasını kullanan yapay sinir ağının eğitime akış şeması

Şekil-4.22 'deki akış şeması Borland C programlama dilinde gerçekleştirilerek (Türkoğlu, 1996), bilgisayar ekranında oluşturulan matris veri tabanından, çeşitli nesnel alınmış ve yapay sinir ağı eğitilmiştir. Bir sonraki aşamada, eğitilen yapay sinir ağına, yine matris veri tabanında oluşturulan nesnel verilerek yapay sinir ağının bunları tanıma başarıları tesbit edilmiştir. Yapılan bu deneysel çalışmalar aşağıda sunulmuştur.

Uygulama 1: İstatistiksel nesne tanıma yöntemi kullanılarak 5x7 'lik matris veri tabanı üzerinden alınan, bozuk olmayan 26 temel alfabetik karakter örüntüsü ile, geriye yayılım ağı eğitilerek bu alfabetik karakterlerin değişik biçimlerinin tanınması.

Her bir örüntünün öznitelik vektörü, bölüm 4.1.3.1 'de açıklanan istatistiksel yöntemle çıkarılmıştır. Kullanılan yapay sinir ağının yapısı; her bir örüntü 5x7=35 noktadan oluştuğu için giriş katmanında 35 nöron, çıkış katmanında temel alfabetik karakter sayısı kadar (26) nöron seçilmiş ve ara katmandaki nöron sayısını belirlemede belli bir kriter olmamakla birlikte denemeler sonucunda giriş ile çıkış katmanlarındaki nöron sayılarından büyük olanına eşit veya ondan büyük seçildiğinde başarılı sonuçlar alınabildiği tespit edilmiştir. Buna göre ara katmandaki nöron sayısı, 35 olarak seçilmiştir. Böylece yapay sinir ağı giriş, ara ve çıkış olmak üzere 3 katmandan oluşmuş, sırasıyla 35-35-26 nöron sayılarını içerecek şekilde yapılandırılmıştır. Yapay sinir ağını eğitmek için gerekli diğer bilgiler ise şu şekildedir;

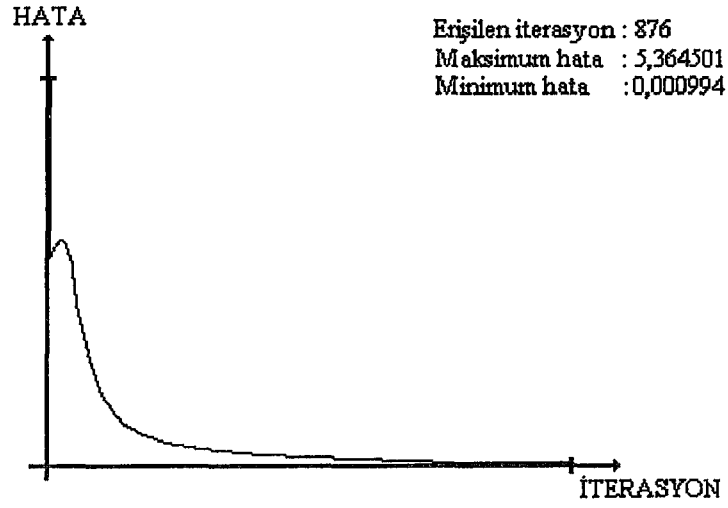
Öğrenme katsayısı (ϵ) : 0.9

Momentum katsayısı (α) : 0.7

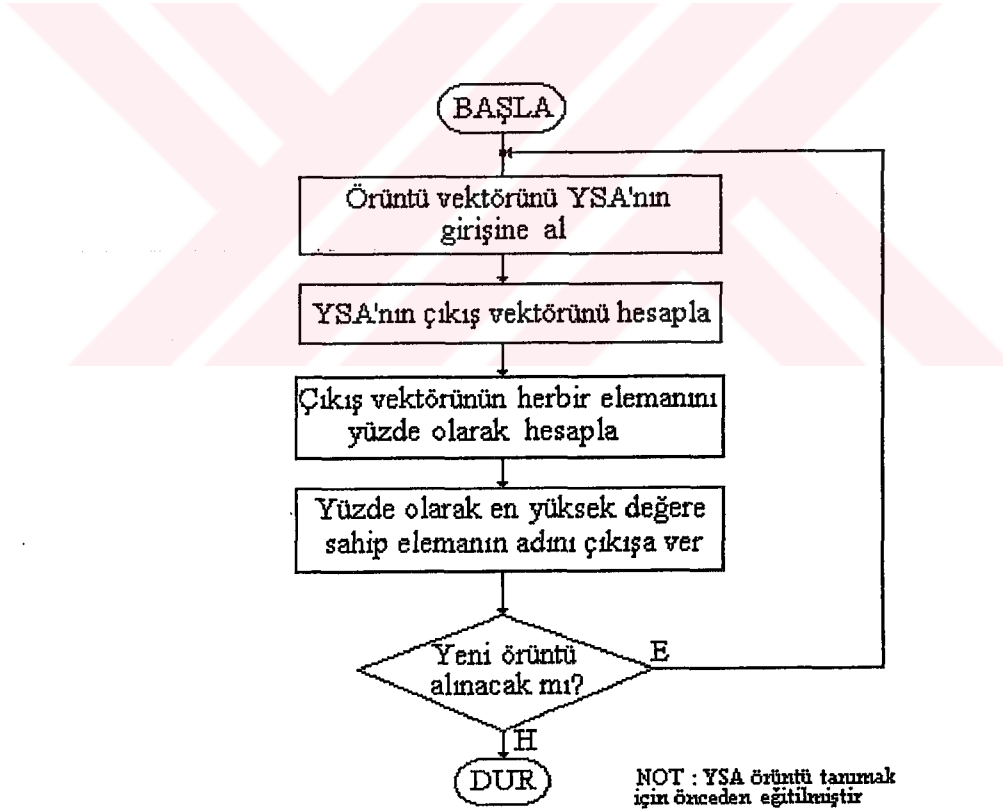
Tüm örüntüler için izin verilen maksimum sistem hatası : 0.001

Tek bir örüntü için izin verilen maksimum sistem hatası : 0.0001

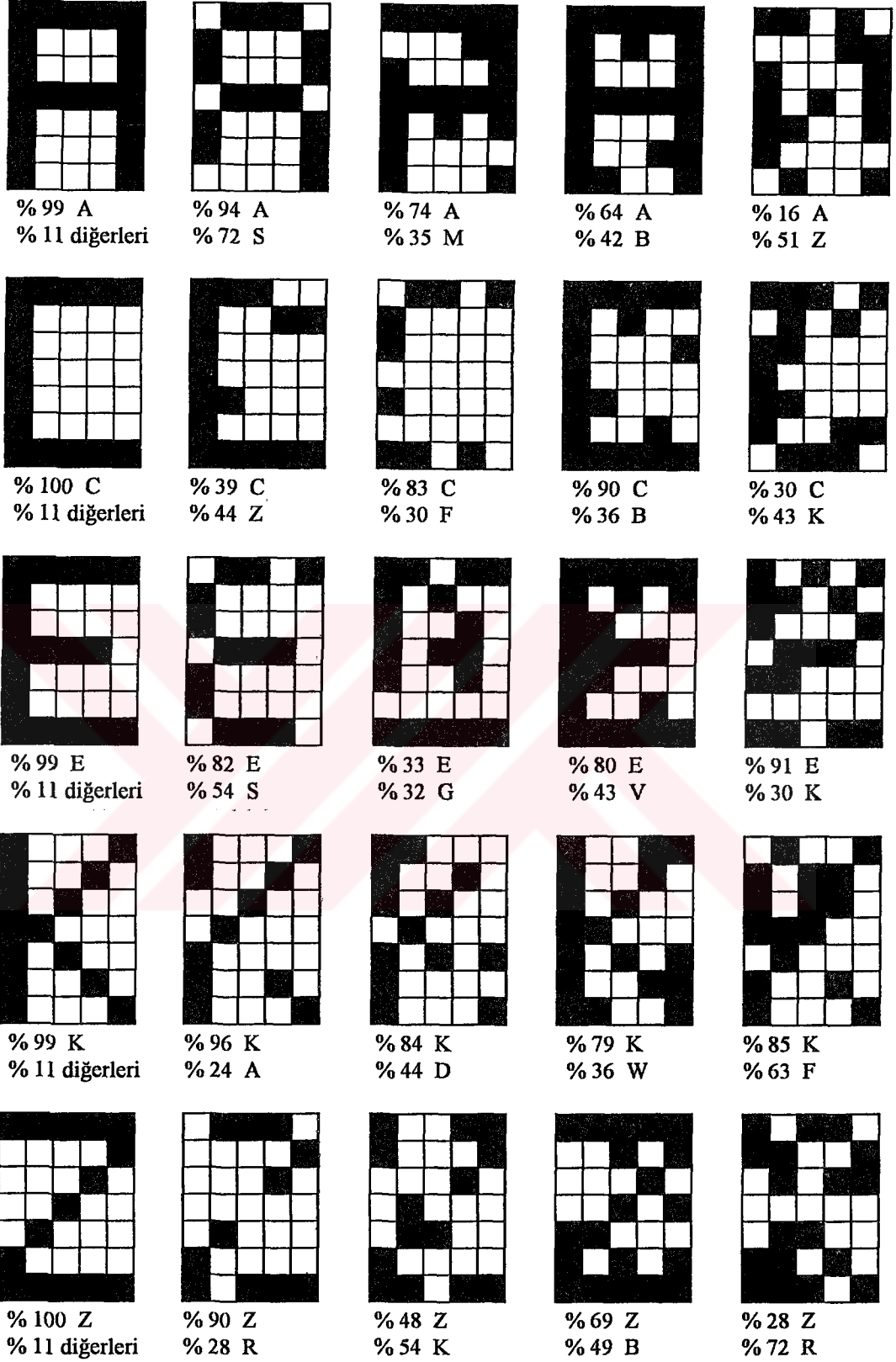
Bu şartlar altında 26 giriş örüntüsü için eğitilen sistemin hatasının, her bir iterasyon (eğitme döngüsü) için değişimi şekil-4.23 'de verilmiştir. Yapay sinir ağının eğitilmesinde %99 oranında bir başarı elde edilmiştir. Eğitim işlemi tamamlandıktan sonra şekil-4.24 'de akış şeması verilen bilgisayar programı çalıştırılarak eğitilmiş yapay sinir ağının tanıma başarısı ölçülmüştür. Bu ölçüm sonuçları şekil-4.25 'de verilmiştir. Şekil-4.25 'de ilk sütün yapay sinir ağına gerek eğitime ve gerekse tanıma için verilen örüntülerin bozuk olmayan hali olup, diğerleri ise bozuk olanlardır. Her örüntünün altında verilen değerler, o örüntünün en çok benzediği örüntü olup yapay sinir ağının çıkış vektörüne bakılarak, yüzde olarak hesaplanmıştır.



Şekil-4.23. Geriye yayılım ağının, eğitime esnasında sistem hatasının değişimi.



Şekil-4.24. Yapay sinir ağında nesne tanıma işleminin akış şeması.



Şekil-4.25. Alfabetik karakter örüntülerinin geriye yayılım ağı tarafından istatistiksel olarak tanıma yüzdeleri.

Uygulama 2 : Geometrik nesne tanıma yöntemi kullanılarak 10x10 'luk matris veri tabanı üzerinden alınan temel 26 alfabetik karakter örüntüsü ile geriye yayılım ağının eğitilmesi ve bu karakterlerin değişik biçimlerinin geriye yayılım ağı tarafından tanınması.

Bu uygulamada alfabetik karakter örüntülerinin öznitelik vektörleri, geliştirilen geometrik metot (bkz. bölüm 4.1.3.2.) kullanılarak elde edilmiştir. Öznitelik vektörleri geometrik olarak elde edildiğinde, vektörün boyu matrisel giriş veritabanındaki parça (grid) sayısı kadar olmak yerine, daha az elemanlı olacaktır. Dolayısıyla yapay sinir ağının giriş katmanındaki nöron sayısı, istatistiksel yöntemdekine göre daha az sayıdadır.

- Yapay sinir ağının yapısı;

Katman sayısı : 3

Katmanlardaki nöron sayısı : 33-60-26 (Giriş-Ara-Çıkış)

- Eğitime şartları;

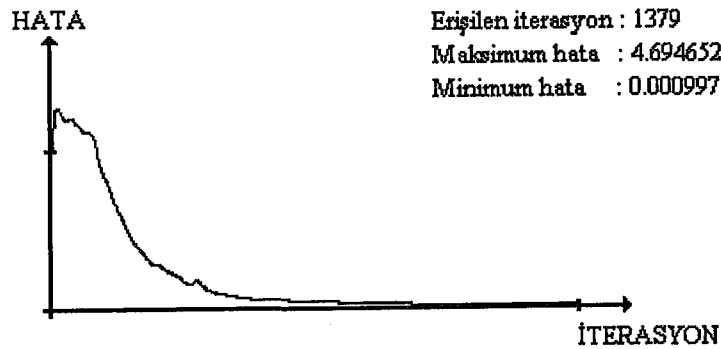
Öğrenme katsayısı (ϵ) : 0.5

Momentum katsayısı (α) : 0.3

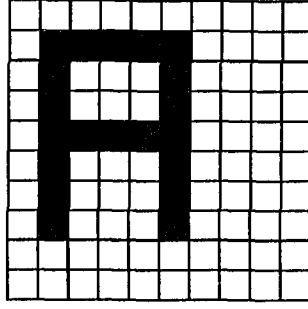
Tüm örüntüler için izin verilen maksimum sistem hatası : 0.001

Tek bir örüntü için izin verilen maksimum sistem hatası : 0.0001

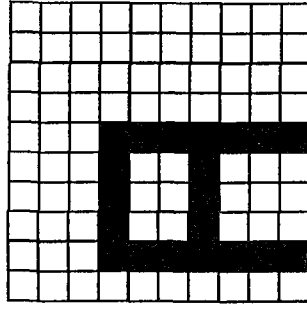
Şekil-4.26 'da yapay sinir ağının alfabetik karakter örüntülerinin, geometrik yöntemle elde edilen öznitelik vektörleri ile eğitilmesinde, sistem hatasının eğitime döngüsüne göre değişimi görülmektedir. Şekil-4.27 'de ise, ilk sütun yapay sinir ağına gerek eğitime ve gerekse tanıma için verilen örüntülerin bozuk olmayan hali olup, diğerleri ise bozuk olanlardır. Her örüntünün altında, o örüntünün en çok benzediği örüntü, yapay sinir ağının çıkış vektörüne bakılarak, yüzde olarak hesaplanmıştır.



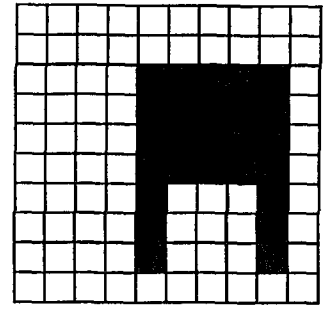
Şekil-4.26. Geriye yayılım ağının eğitime esnasında sistem hatasının değişimi.



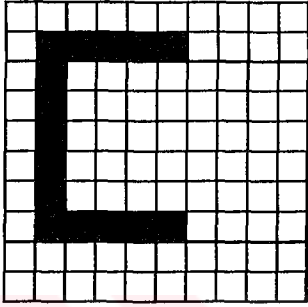
% 99 A
% 11 diğerleri



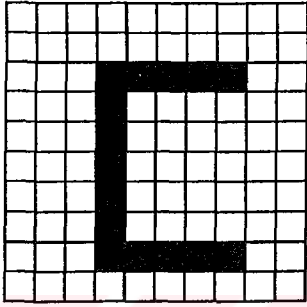
% 99 A
% 11 diğerleri



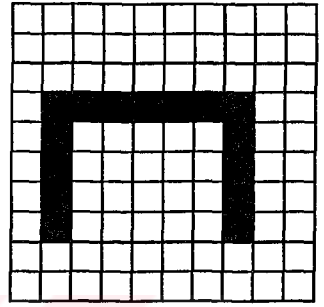
% 99 A
% 11 diğerleri



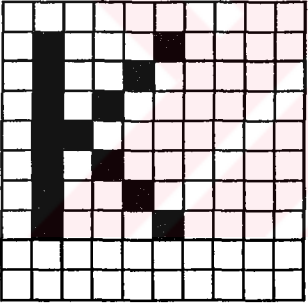
% 99 C
% 12 Z



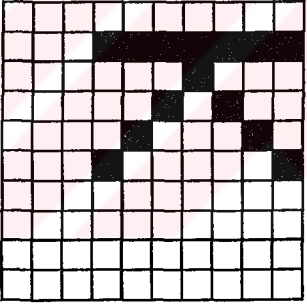
% 99 C
% 12 Z



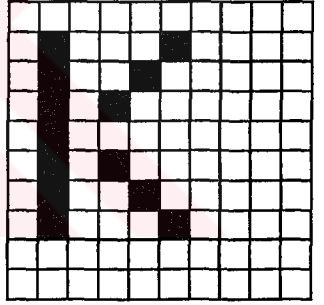
% 99 C
% 12 Z



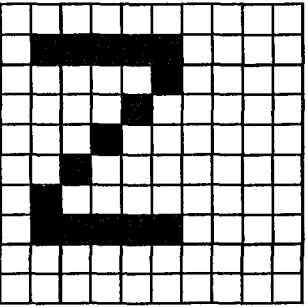
% 99 K
% 11 diğerleri



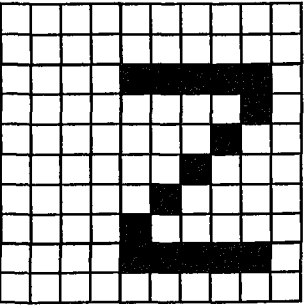
% 99 K
% 11 diğerleri



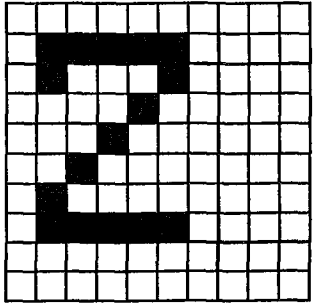
% 9 K
% 82 F



% 99 Z
% 11 diğerleri



% 99 Z
% 11 diğerleri



% 5 Z
% 91 D

Şekil-4.27. Alfabetik karakter örüntülerinin geriye yayılım ağı tarafından geometriksel olarak tanınma yüzdeleri.

Uygulama 3 : Geometrik nesne tanıma yöntemi kullanılarak 10x10 'luk matrisel veri giriş tabanı üzerinden alınan geometrik nesne şekilleri ile geriye yayılım ağının eğitilmesi ve bu nesnelerin değişik biçimlerinin geriye yayılım ağı tarafından tanınması.

Bu uygulamada da geometrik nesnelerin öznitelik vektörleri, geliştirilen geometrik yöntem (bkz. bölüm 4.1.3.2.) kullanılarak elde edilmiştir.

- Yapay sinir ağının yapısı;

Katman sayısı : 3

Katmanlardaki nöron sayısı : 33-50-26 (Giriş-Ara-Çıkış)

- Eğitime şartları;

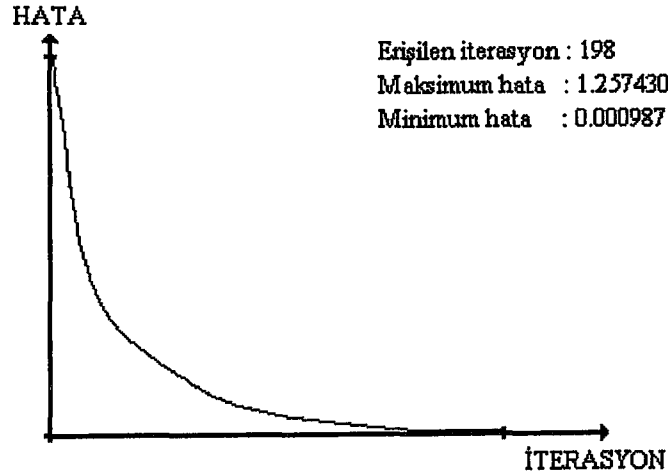
Öğrenme katsayısı (ϵ) : 0.4

Momentum katsayısı (α) : 0.1

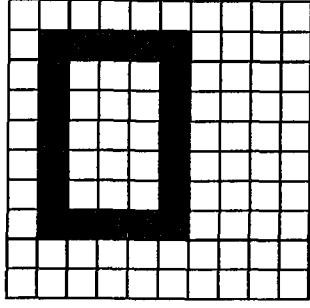
Tüm örüntüler için izin verilen maksimum sistem hatası : 0.001

Tek bir örüntü için izin verilen maksimum sistem hatası : 0.0001

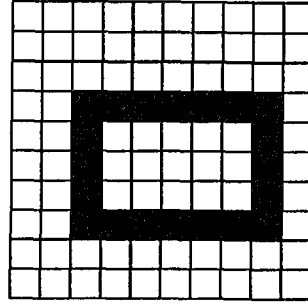
Şekil-4.28 'de yapay sinir ağının, geometrik nesnelerin geometrik yöntemle elde edilen öznitelik vektörleri ile eğitilmesinde sistem hatasının, eğitime döngüsüne göre değişimi görülmektedir. Uygulama sonuçları ise şekil-4.29 'de verilmiştir. Bu şekilde ilk sütun yapay sinir ağına gerek eğitime ve gerekse tanıma için verilen geometrik nesnelerin bozuk olmayan hali olup, diğer sütunlar ise nesnelerin bozuk durumlarıdır. Her nesnenin altında, o nesnenin en çok benzediği nesne, yapay sinir ağının çıkış vektörüne bakılarak, yüzde olarak belirtilmiştir.



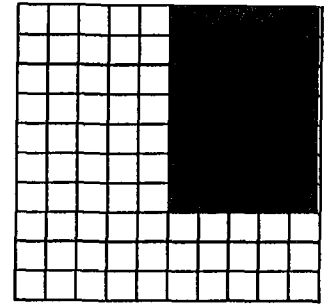
Şekil-4.28. Geriye yayılım ağının eğitime esnasında sistem hatasının değişimi



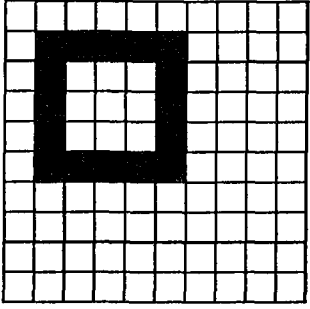
% 99 dikdörtgen
% 11 diğerleri



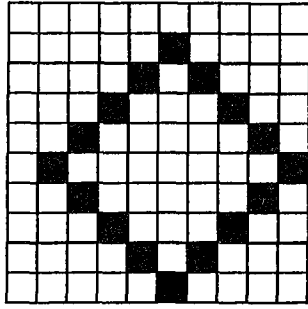
% 99 dikdörtgen
% 11 diğerleri



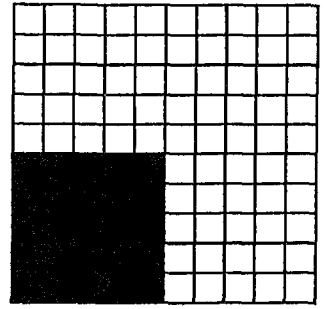
% 99 dikdörtgen
% 11 diğerleri



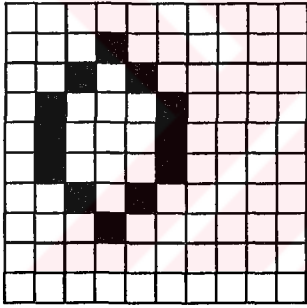
% 99 kare
% 11 diğerleri



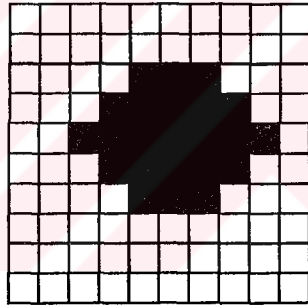
% 100 kare
% 8 diğerleri



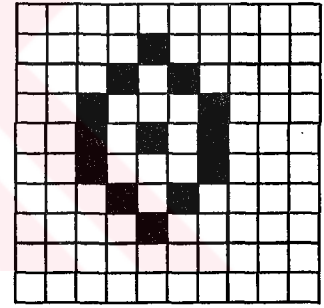
% 99 kare
% 11 diğerleri



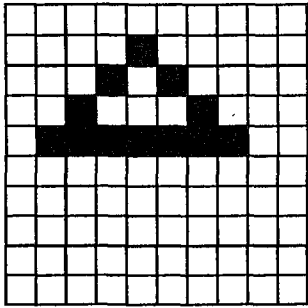
% 97 altıgen
% 13 üçgen



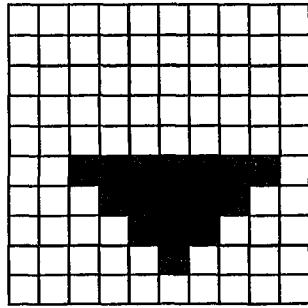
% 97 altıgen
% 13 üçgen



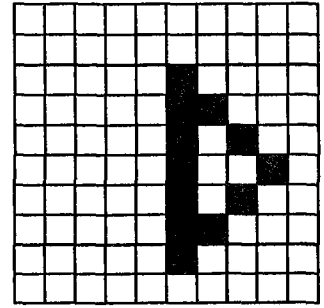
% 27 altıgen
% 53 üçgen



% 98 üçgen
% 13 altıgen



% 100 üçgen
% 12 dikdörtgen



% 100 üçgen
% 12 dikdörtgen

Şekil-4.29. Geometrisel nesnelerin geriye yayılım ağı tarafından geometrisel olarak tanınma yüzdeleri.

Yukarıdaki benzetim sonuçları, eğitici bir yapay sinir ağı olan, hatanın geriye yayılma algoritmasını kullanan çok katmanlı geriye yayılım ağının sınıflandırıcı olarak kullanıldığı ve nesnenin öznelik vektörlerinin iki farklı şekilde (istatistiksel ve geometriksel) çıkarıldığı nesne tanıma sisteminden elde edilmiştir.

Şekil-4.25 'de herbir alfabetik karakter örüntüsünün bozuk olmayan şekli yapay sinir ağına öğretilip, sonra herhangi bir nedenden dolayı bu örüntülerde meydana gelecek olan bozulma durumunda, eğitici yapay sinir ağının bu örüntüleri tanıma başarısı ölçülmüştür. Alfabetik karakter örüntüleri, eğitici yapay sinir ağına 15 dakika (486DX-33Mhz bilgisayarda) gibi bir sürede belirtilen hata sınırlamaları çerçevesinde ancak %99 ulaşan bir oranla öğretilmiş ve bu öğretilen örüntülerde %40 'lara varan bozulma durumunda eğitici yapay sinir ağı, eğitici yapay sinir ağına sağlanan başarıya ulaşamamıştır. Eğitici yapay sinir ağının eğitilme esnasındaki sistem hatasının, eğitme döngüsüne bağlı olarak değişimi şekil-4.23 'de verilmiştir. Bu değişim, denemeler sonucunda bulunan en uygun parametreler (ϵ ve ∞) sonucunda elde edilmiştir. Bu grafikten görüldüğü gibi yapay sinir ağının öğrenme işlemi fazla bir osilasyon yapmadan gerçekleşmiştir. Eğer uygun parametreler seçilmezse sistem osilasyon yapar, dolayısıyla öğrenme işlemi kararsız bir seyir halinde uzun süre alabilir veya yakınsamayabilir.

Şekil-4.27 ve şekil-4.29 'de ise, nesne olarak alfabetik karakterler ve geometriksel şekiller alınmıştır. Herbir nesnenin öznelik vektörü, geliştirilen geometriksel yöntemle bulunmuştur. Dolayısıyla, nesnelerin genel özellikleri kullanılmıştır. Bu yöntem, nesnelere dönmeden ve konum değiştirmeden bağımsız olarak kesin bir şekilde tanıyabilmekte, buna karşılık nesneleredeki küçük bir yerel bozulma, nesnenin yanlış tanınmasına neden olabilmektedir. Eğitici yapay sinir ağı ile her bir nesne sınıfı, daha belirgin olarak ayrılmıştır. Nitekim herhangi bir nesne için elde edilen tanıma yüzdeleri arasındaki farkın yüksek olmasından bu durum görülebilmektedir. Her iki nesne grubunun sistem hatasının değişimi, yine denemeler sonucunda bulunan en uygun parametrelerin kullanılması ile elde edilmiştir.

BÖLÜM 5

5. SONUÇ VE DEĞERLENDİRME

Bu tez çalışmasında, nesne tanıma konusunda yapılan araştırmalar ve yöntemler incelenmiştir. İncelemeler sonucunda genelde nesne tanıma sistemlerinin, öznelilik çıkarıcı ve sınıflandırıcı olmak üzere iki ana bölümden meydana geldiği görülmüştür. Tezin, ilerleyen bölümlerinde farklı yapay sinir ağı modelleri kullanan iki boyutlu bir nesne tanıma sistemi geliştirilmiştir.

Bölüm 2 'de yapılan literatür taramasıyla, nesne tanıma esaslı robot görmesi ve bilgisayar görmesi kavramları, boyutsal nesne tanıma kavramları ve nesne tanıma yaklaşımları ayrıntılı olarak açıklanmıştır. Bölüm 3 'de ise yapay sinir ağları ile gerçekleştirilen nesne tanıma sisteminin alt yapısını oluşturmaya yönelik çalışma yapılmıştır. Bölüm 4 'de daha önce esasları belirlenen iki boyutlu nesne tanıma sisteminin geliştirilen bazı temel elemanları ile birlikte uygulamalarına yer verilmiştir.

Yapılan uygulamalar neticesinde, istatistiksel nesne tanıma Kohonen yapay sinir ağı, geriye yayılım yapay sinir ağına göre daha yüksek başarı oranları sağlamıştır. Geliştirilen istatistiksel nesne tanıma modeli, nesnenin yön değişiminden ve konum değişiminden bağımsız olmamakla birlikte, nesnenin yapay sinir ağına öğretilen biçiminin %40 oranında bozulmasına rağmen, nesneyi doğru bir şekilde tanıyabilmiştir. Geliştirilen geometriksel nesne tanıma modeli ise, nesnenin konum değiştirmesinden ve dönmesinden bağımsız olarak, hem Kohonen ve hem de geriye yayılım ağı tarafından başarılı bir şekilde tanıyabilmiştir. Fakat geriye yayılım ağına, Kohonen yapay sinir ağına göre nesne sınıfları, daha belirgin olarak ayrılmıştır. Bu ise, geriye yayılım ağının geometriksel nesne tanıma Kohonen yapay sinir ağına göre daha üstün olduğunu göstermektedir. Yapılan denemeler sonucunda görülmüştür ki, geometriksel nesne tanıma modelinde, nesnelerin yapay sinir ağına öğretilen şeklinin küçük bir değişimi dahi, nesnenin her iki yapay sinir ağınca da yanlış tanınmasına neden olabilmektedir. Bu uygulamalar dikkate alınarak diyebiliriz ki; eğitici bir ağ olan Kohonen yapay sinir ağı istatistiksel nesne

tanımada, eğitici bir ağ olan geriye yayılım aği ise, geometriksel nesne tanımada daha başarılıdır.

Ayrıca, tanıma işlemi yapılacak olan nesnelere öznitelik vektörlerinin çıkarılmasında kullanılan ve bu tez kapsamında geliştirilen kenar çıkarma şablonu, yapılan denemeler sonucunda kenar çıkarmada istenen şartları oldukça başarılı bir şekilde sağlamıştır. Yine aynı amaçla geliştirilen kenar izleme algoritması ise nesnelere karmaşıklık durumlarına göre, kısmen olumlu sonuçlar vermiştir.

Nesne tanımada sınıflandırıcı olarak yapay sinir ağlarının dışında bulanık mantık sınıflandırıcılarında kullanılmasına yönelik bazı çalışmalar vardır. Yapay sinir ağları ile bulanık mantığın beraber kullanılması (benzer paralel özelliklerinden dolayı) nesne tanıma probleminin çözümünde çok uygun bir bütünlük oluşturabilir. Yapay sinir ağları, bulanık mantık için gerekli olan kuralların ve üyelik fonksiyonlarının belirlenmesinde yardımcı olur. Bulanık mantık ve yapay sinir aği ikilisinin bütünlüğü, nesne tanıma olayına yeni bir bakış açısı getirebilir.

Yapay sinir ağları ile nesne tanıma çalışmasının başlıca uygulama alanlarına örnek olarak, makina parçalarını seçme, sınıflandırma, hatalı parçaların ayrılması, kaplama, paketleme, montaj gibi nesne tanıma tabanlı uygulamaları sayabiliriz. Kısaca, geliştirilen bu nesne tanıma sistemi görmenin esas olduğu pek çok sahaya, özellikle robot uygulamalarında gerekli düzenlemeler yapılarak ve sistemin gerektirdiği donanım ilave edilerek rahatlıkla uygulanabilir. Bunların dışında nesne tanıma işlemi, aşağıda verilen alanlara da uygulanabilir.

El yazılarının tanınmasında, yapay sinir ağları ile başarılı çalışmalar yapılmaktadır. Örnek el yazıları kullanılarak, ağ eğitilir. Bu çalışmalar, ülkemiz için çok önemlidir. Çünkü bu yolla kütüphanelerde bulunan sayısız el yazması kitapların (özellikle Osmanlıca yazılmış olanların) latin alfabesine çevrilmesi mümkün olabilir.

Askeri uygulamalarda ise, yapay sinir ağlarının özelliklerinden değişik alanlarda faydalanmak mümkündür. Özellikle askerlerin, pilotların eğitilmesinde kullanılan ve bilgisayarlar yardımıyla oluşturulan düşman kuvvetlerinin modellenmesi, en çok potansiyel sahip uygulama alanı gibi görünmektedir. Bilgisayarın, oluşturduğu bir uçağın hangi şartlarda hangi manevraları yapacağını öğrenmesi gibi bir durumda, örnek

manevralar kullanılarak deęişik durumlar için deęişik manevralar öğretilir. Daha sonra eğitilmiş uçak benzer durumlarda benzer manevraları yapabilir.

Bu örnekleri çoğaltmak mümkündür. Görünen odur ki, önümüzdeki yıllarda daha fazla sayıda bilim adamı yapay sinir aęları üzerinde araştırma yapacak ve yeni öğrenme kuralları, yeni aę yapıları ortaya çıkacaktır. Özellikle biyologlar, mühendisler nörofizyologlar, bilgisayarıcılar, fizikçiler vb. bir çok bilim adamı, bu teknolojiye katkıda bulunacak ve teknoloji geliştikçe, insanoęlu için bir sır olan insan beyninin çalışma prensipleri hakkında, bir takım ipuçları elde edilecektir. Belki de yakın gelecekte, insanlığın hizmetine sunulacak görme esası üzerine dayalı gelişmiş robotlar üretilebilecektir.

Sonuç olarak, geleceęi çok daha verimli olacak olan yapay sinir aęlarıyla görme incelendikçe, insanın biyolojik sinir aęları ve görme sistemi karşısındaki hayranlığı, her geçen gün artacaktır.

KAYNAKLAR

- AÇIKGÖZ, G., (1993). **Çok Boyutlu Gauss Süreçlerine Dayanan Parametrik Şekil Tanıma Yöntemleri Ve Bu Yöntemlerle Kalp Aritmilerinin Belirlenmesi.** Yüksek Lisans Tezi, İ.T.Ü. Fen Bilimleri Enstitüsü, İstanbul.
- BAĞCIOĞULLARI, F., ve BAĞCIOĞULLARI, C.H., (1994). Nesne tanıma sistemi için geliştirilmiş özel bir kenar çıkarma algoritması. **2. Signal İşleme ve Uygulamaları Sempozyumu Bildiriler Kitabı.** s.47-52. Muğla.
- BAĞCIOĞULLARI, F., ve BAĞCIOĞULLARI, C.H., (1995). **Nesne Tanımada Yeni Bir Yaklaşım : Fuzzy Kümeler.** Tübitak marmara araştırma merkezi raporu. Kocaeli.
- BANKS, S., (1990). **Signal Processing, Image Processing and Pattern Recognition.** Sheffield University. USA.
- BOLLES, R.C., BRADY, M., ve PAUL, R., (1984). 3DPO : A three - dimensional part orientation System. **Robotics Research : The 1st International Symposium.** pp.413.
- CHIN, R.T., ve DYER, C.R., (1986). Model-Based recognition in robot vision. **Computing Surveys.** Vol.18.
- DUDA, R.O., ve HART, P.E., (1989). **Pattern Classification and Scene Analysis.** Stanford Research Institute. California. USA.
- FUKUMI, M., OMATU, S., TAKEDA, F., ve KOSAKA, T., (1992). Rotation - invariant neural pattern recognition system with application to coin recognition. **IEEE Trans. on Neural Networks.** vol.3, no.2, pp.272-278.
- GLEASON, G.J., ve AGIN, G.J.,(1979). Modular system for sensor controlled manipulation and inspection. **Proc. of The 9th International Symposium on Industrial Robots.**

- IOANNIS, P., (1993). **Digital Image Processing Algorithms**. Prentice hall international series in signal processing. New York.
- IKEUCHI, K., ve HONG, K.S., (1991). Determining linear shape change : Toward automatic generation of object recognition programs. **CVGIP : Image Understanding**. Vol.53, pp.154.
- KARLIK, B., (1995). **Yapay Sinir Ağları**. Yüksek lisans ders notları.
- KARLIK, B.,ve PASTACI, H., (1993). Yapay sinir ağlarında T / C tanıma problemi. **Elektrik Mühendisliği 5. Ulusal Kongresi Bildiriler Kitabı**. s.847-852. Trabzon.
- KOLLIAS, S.,ve TRIAKIS, A., (1991). **An efficient approach to invariant recognition of image using higher-order neural networks**. Elsevier Science Publishers. pp.87.
- KUHL, F.P., (1981). Elliptic fourier features of a closed contour. **Computer Graphics Image Process**. vol.18, s.236-258.
- LAMDAN, Y., ve WOLFON, H.J., (1988). Geometric hashing : A general and efficient model-based recognition scheme. **2nd International Conference on Computer Vision**. pp.238. Florida.
- MAREFAT, K., (1990). Geometric reasoning for recognition of 3-D object feature. **IEEE Transactions on Pattern Analysis and Machine Intelligence**. vol.12, s.950-964.
- NELSON, N., ve ILLINGWORTH, W.T., (1991). **A Practical Guide to Neural Nets**. Addison-Wesley Publishing.
- NOF, Y.S., (1985). **Handbook of Industrial Robotics**. Jhon Wiley & Sons, New York.

- OSHIMA, M., ve SHIRAI, Y., (1983). Object recognition using there - dimensional information. **IEEE Trans. Pattern Anal. Mach. Intell.** vol.5, no.4, pp.353.
- ÖZMETEL E., (1996). Bilgisayarda öğrenme ve yapay nöral ağları. **Otomasyon.** mart, sayı.45, s.134-140.
- ÖZMETERLER, E., (1989). **Yapay Nöral Ağları.** Yüksek Lisans Tezi, İ.T.Ü. Fen Bilimleri Enstitüsü, İstanbul.
- PASTACI, H., ve KARLIK B., (1991). Robotlarda görme ve görüntü algılama. **Elektrik Mühendisliği IV. Ulusal Kongresi Bildiriler Kitabı.** s. 615-617. İzmir.
- SCHALKOFF, R., (1990). **Pattern Recognition : Statistical, Structural and Neural Approaches.** John Wiley & Sons.
- STOCKMAN, G.C., (1982). Matching images to models for registration and object detection via clustering. **IEEE Trans. Pattern Anal. Mach. Intell.** vol.1, no.3, pp.57.
- TEZER, A., (1993). **Robot Görmesi.** Tübitak marmara araştırma merkezi raporu. Kocaeli.
- TÜRKOĞLU, İ., (1996). **C Programlama Dilinde Geliştirilmiş Yapay Sinir Ağları ile Nesne Tanıma Yazılım Paketi.**
- TÜRKOĞLU, İ., ve ARSLAN, A., (1996). Yarışmacı öğrenme kuralı ile bozuk örüntüleri tanıma. **IX. Mühendislik Sempozyumu, Elektronik-Haberleşme Mühendisliği Bildiriler Kitabı.** s.165-170. Isparta.
- TÜRKOĞLU, İ. ve ARSLAN, A., (1996). Yapay sinir ağları ile bozuk örüntü tanıma. **F.Ü. Fen Bilimleri Enstitü Dergisi.** Elazığ.
- UZUNALIOĞLU, H., (1992). **Model-Based Recognition of Polyhedral Objects.** Yüksek Lisans Tezi. ODTU Fen Bilimleri Enstitüsü. Ankara.

ÜNLÜAKIN, A.U., ve GÜRGEN, F.S., (1993). Yapay nöral ağları ile klasik sınıflandırma teknikleri arasında bir karşılaştırma. **Elektrik Mühendisliği 5. Ulusal Kongresi Bildiriler Kitabı**. s.275-279. Trabzon.

ZURADA, M.J., (1992). **Introduction to Artificial Neural Systems**. West Publishing Company, Inc. New York.

