

YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

139845

YAPAY SİNİR AĞLARIYLA 3-D BİLGİSAYAR
GÖRMESİ

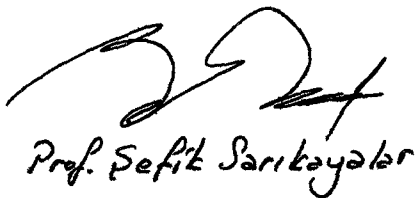
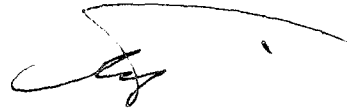
-139845-

Elektronik-Haberleşme. Müh. Muharrem MERCİMEK

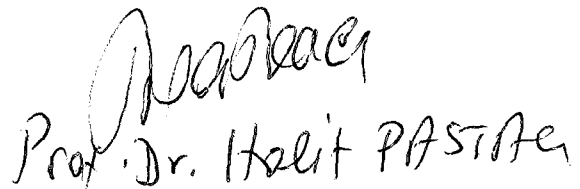
F.B.E. Elektronik ve Haberleşme Anabilim Dalı Elektronik Programında
Hazırlanan

YÜKSEK LİSANS TEZİ

Tez Danışmanı: Doç. Dr. Tülay YILDIRIM



Prof. Şefik Sarıkoç



Prof. Dr. Halit PAŞTA

İSTANBUL, 2003

YILDIZ TEKNİK ÜNİVERSİTESİ
BİLGİ TEKNOLOJİLERİ ENSTİTÜSÜ

İÇİNDEKİLER

	Sayfa
SİMGE LİSTESİ	v
KISALTMA LİSTESİ	vi
ŞEKİL LİSTESİ	vii
ÇİZELGE LİSTESİ	viii
ÖNSÖZ	ix
ÖZET	x
ABSTRACT	xi
1. GİRİŞ	1
1.1 Şablon eşleme	3
1.2 İstatistiksel Yaklaşım	4
1.3 Söz Dizilimsel veya Yapısal Yaklaşım	4
1.4 Sinir Ağları	5
1.5 Bilgisayar Görmesi Problemine Getirilmiş Genel Çözümler	5
2 GÖRÜNTÜ İŞLEME	8
2.1 Dijital Görüntü	8
2.2 Görüntü İyileştirme	9
2.2.1 Noktasal İşlemler	9
2.2.1.1 Kontrast Yayma	10
2.2.1.2 Kesme ve Eşikleme	10
2.2.1.3 Sayısal Negatif	11
2.2.1.4 Aralık Sıkıştırma	11
2.2.1.5 Histogram Dengeleme	11
2.2.2 Uzaysal İşlemler	12
2.2.2.1 Komşuluk Ortalaması	12
2.2.2.2 Medyan Filtreleme	12
2.2.2.3 Alçak, Yüksek, Band Geçiren Filtrelemeler	13
2.2.3 Dönüşüm İşlemleri	13
2.3 Görüntü Bölütleme	14
2.3.1 Sınır Temelli Yaklaşımlar	14
2.3.1.1 Kenar Belirleme	14
2.3.1.1.1 Gradyan Temelli Metotlar	14
2.3.1.1.2 Laplasyen Temelli Metotlar	16
2.3.2 Bölge Temelli Metodlar	17
2.3.2.1 Piksel Toplanması	17
2.4 Görüntü Analizi ve Tanımlanması	17
2.4.1 Uzaysal Tanımlayıcılar	18
2.4.1.1 Genliğe Dayalı Özellik Çıkarımı	18

2.4.2	Dönüşüm Tanımlayıcıları	18
2.4.3	Sınır Tanımlayıcıları	18
2.4.3.1	Zincir Kodları	18
2.4.4	Bölge Tanımlayıcıları	19
2.4.4.1	Run-Length Kodları	19
2.4.5	Moment Tanımlayıcıları	20
3	MOMENT ENVARYANTLARI	21
3.1	Moment Tanımlayıcıları Kullanarak Özellik Çıkarımı	22
3.2	Düzenli Moment Envaryantları (RMI)	24
3.2.1	Moment Envaryantları Dönüşümleri	24
3.2.1.1	Öteleme	24
3.2.1.2	Ölçekleme	25
3.2.1.3	Döndürme	26
3.2.1.4	Yansıma	26
3.3	Ortogonal Moment Envaryantları	27
3.3.1	Zernike Moment Envaryantları	27
4	İSTATİSTİKSEL SINIFLAYICILAR	30
4.1	Parametrik Bayes Sınıflayıcıları	32
4.1.1	Eşit Varyanslı Bağımsız Özellikler	33
4.1.2	Eşit Olmayan Varyanslı Bağımsız Özellikler	34
4.1.3	Keyfi Kovaryanslar	35
4.2	Non-Parametrik Kernel Tabanlı Sınıflandırma Algoritmaları	36
4.2.1	K-En Yakın Komşu (K-Nearest Neighbor (k-NN))	37
4.2.2	Parzen Pencereleeri (Parzen Windows)	37
4.2.3	h terimi seçimi	38
5	YAPAY SİNİR AĞLARI	39
5.1	Temel Nöron Modeli	41
5.2	İleri Besleme İşlemi ve Sınıflandırma	42
5.3	Geriye Yayılım (Backpropagation) Algoritması	44
5.3.1	Ağın Eğitilmesi	45
5.4	Diğer Bazı Ağ Mimarileri Ve Öğrenme Algoritmaları	49
5.4.1	Radyal Fonksiyon Temelli Ağlar	49
5.4.2	Genelleştirilmiş Regresyon Ağları	50
5.4.3	Kazanma Algoritmali Öğrenme Ağı	51
6	YAPAY SİNİR AĞLARIYLA 3-D BİLGİSAYAR GÖRMESİ UYGULAMASI	53
6.1	Görüntülerin Elde Edilmesi	53
6.2	Üç Boyutlu Cisimlerin İki Boyutlu Görüntülerle İfade Edilmesi	54
6.3	Görüntülerin İşlenmesi	55
6.4	İşlenmiş Görüntülerden Özellik Vektörü Çıkarımı	56
6.5	Eğitim ve Test Verisi Dosyaları Oluşturulması	57
6.6	Sınıflandırma	58

7	SONUÇLAR	61
	KAYNAKLAR.....	63
	EKLER	65
EK 1	Moment Fonksiyonları Kullanarak Özellik Vektörleri Oluşturma Programı.....	66
EK 2	Kullanılan Cisimlere Ait Farklı Açılardan Alınmış Kontur Görüntüleri.....	68
EK 3	Sınıflandırma İşleminde Kullanılan Ağ Konfigürasyonları Tablosu.....	70
	ÖZGEÇMİŞ.....	73



SİMGE LİSTESİ

b	Bias değeri
C_j	Sınıflar
d	İstenen hedef çıkış vektörü
∇	Gradyan operatörü
∇^2	Laplasyen operatörü
$\partial J/\partial w$	Hatanın gradient descent'i
$f(.)$	Dijital görüntü ışık yoğunluk fonksiyonu
$\Phi(.)$	Aktivasyon fonksiyonu
g_j	Diskriminant fonksiyonu
h_{BP}	Band geçiren filtre
h_{LP}	Alçak geçiren filtre
$J(.)$	Hata fonsiyonu
$M_1.M_7$	Moment fonksiyonları
μ_j	Örnekler için ortalama vektörü
η	Öğrenme oranı
$P(C_j)$	Öncül sınıf olasılığı
$P(C_j/x)$	Soncul olasılık
$p(x)$	x örneğinin herhangi bir sınıfa ait olma olasılığı
$p(x/C_j)$	Sınıfa bağlı yoğunluk veya olasılık
$\mathfrak{R}$	Özellik uzayı
σ_f	Yerel varyans
Σ_j	Örnek sınıfları için kovaryans matrisi
u	Aktivasyon değeri
w_i	Bağlantı ağırlıkları
x	Pikselin x koordinatı
x_i	Yapay sinir ağı giriş vektörü
y	Pikselin y koordinatı
y_i	Yapay sinir ağı çıkış vektörü

KISALTMA LİSTESİ

ANN	Artificial Neural Networks
AR	Auto Regressive
BP	Backpropagation
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
EKG	Elektrokardiogram
GAL	Gaining Algorithm Learning
GRNN	Generalized Regression Neural Networks
IR	Infra Red
k-NN	k-Nearest Neighbor
LMS	Least Mean Square
LVQ	Learning Vector Quantization
MLP	Multilayer Perceptron
RBF	Radial Basis Functions
RGB	Red Green Blue
RMI	Regular Moment Invariants
YSA	Yapay Sinir Ağları



ŞEKİL LİSTESİ

Şekil 2.1	Gri renk seviyeleri ve binary renk seviyeleriyle kuantalanmış görüntüler.....	9
Şekil 2.2	Kontrast yayma işlemi	10
Şekil 2.3	Kesme ve eşikleme	10
Şekil 2.4	Histogram dengeleme	12
Şekil 2.5	a)Alçak geçiren filtreleme b)Yüksek geçiren filtreleme c) Band geçiren filtreleme	13
Şekil 2.6	Tipik bir 1-boyutlu kenar için (a) $f'(x)$, (b) $f''(x)$, (c) $f'''(x)$	15
Şekil 2.7	Gradyan temelli kenar bulma algoritması	16
Şekil 2.8	Laplasyen temelli kenar bulma algoritması.....	16
Şekil 2.9	Piksel toplanması.....	17
Şekil 2.10	Sınır tanımlayıcılarından zincir kodlama	19
Şekil 2.11	Run-length kodlama	20
Şekil 3.1	Örnek tanıma işleminin temel elemanları.....	22
Şekil 3.2	Moment envaryantlarının sınıflandırılması	23
Şekil 4.1	İstatistiksel sınıflayıcı modeli	30
Şekil 4.2	Eşit varyanslı sıfır kovaryanslı ve lineer diskriminanta sahip sınıflayıcı	34
Şekil 4.3	Farklı varyanslı sıfır kovaryanslı ve lineer diskriminanta sahip sınıflayıcı	35
Şekil 4.4	1 en yakın komşu metoduyla bulunan keyfi discriminanta sahip sınıflayıcı.....	37
Şekil 5.1	Temel nöron modeli.....	41
Şekil 5.2	En yaygın aktivasyon fonksiyonları	42
Şekil 5.3	Çok katmanlı yapay sinir ağı yapısı	43
Şekil 5.4	Tek katmanlı ağlar ile çok katmanlı ağların karşılaştırılması.....	44
Şekil 5.5	Hatanın geriye yayılımı	46
Şekil 5.6	Radyal Fonksiyon Temelli Ağlar	49
Şekil 5.7	Genelleştirilmiş Regresyon Ağ Yapısı	51
Şekil 5.8	GAL ağının yapısı	51
Şekil 6.1	Kullanılan 3-D cisimler	54
Şekil 6.2	Uygulamada kullanılan görüntüleme sistemi	54
Şekil 6.3	Görüntü işleme adımları sırasında elde edilen görüntüler.....	55

ÇİZELGE LİSTESİ

Çizelge 7.1 Ağ yapılarıyla elde edilen sınıflandırma sonuçları.....	61
---	----



ÖNSÖZ

Yapay Sinir Ağlarıyla 3-D Bilgisayar Görmesi isimli tez çalışmamın hazırlanmasında katkılarını esirgemeyerek yardımcı olan tez danışmanı hocam sayın Doç. Dr. Tülay Yıldırım'a, verdikleri değerli fikirlerle tezime yön veren sayın Dr. Abdullah Bal'a ve hocam sayın Prof. Dr. Halit Pastacı' ya, maddi manevi beni her zaman teşvik eden aileme, desteklerinden dolayı araştırma görevlisi arkadaşlarıma, teşekkürü bir borç bilirim.



ÖZET

Bu çalışmada üç boyutlu fiziksel dünyaya ait katı cisimlerin tanınması temel problem olarak ele alınmıştır. Üç boyutlu hacimsel cisimleri sınıflandırma işlemi için iki boyutlu görüntülerin sınıflandırılması işleminin sentezinden faydalanılmıştır. İyi tanımlanmış sınırları yeterince belirlenmiş tanıma problemi (örneklerde küçük sınıf içi değişimleri ve büyük sınıflar arası değişimler) az yer kaplayan örnek sunumlarına ve basit karar verme mekanizmalarının yeterliliğine götürecektir. Bu nedenle, en etkin şekilde fiziksel 3-D cisimlerin özellik vektörlerinin çıkarılması ve sınıflandırılması tez çalışmasının amacı olarak sunulmuştur.

Tabiatta analog işaretler veya sürekli uzay işaretleri grubuna dahil olan görüntü işaretlerinin sayısal ortama taşınması ilk adımdır. Sınıf büyüklüklerini ifade eden üç ayrı cisim için toplam 216 görüntü elde edilmiştir. İkinci adım olarak görüntüyü verimli kullanmak adına çeşitli iyileştirme ve belirginleştirme algoritmaları kullanılarak görüntülere dair asıl ilgilendiğimiz kısımlar elde edilmiştir. Eşiklendirme, kontur belirleme vb. işlemler yapılarak görüntülere ait ikili (binary) aydınlık seviye değerlerine kuantalanmış dış hatların görüntüleri elde edilmiştir. Bu görüntüler kullanılarak görüntü veya şekilde boyut pozisyon ve ölçek değişimleri ve kombinasyonları oluşmasına karşılık değişmezlik özelliği gösteren katsayılar olan Moment fonksiyonları katsayıları elde edilmiştir. Bu katsayılardan elde edilen özellik vektörleriyle tanımlı örnekler kullanılarak tezin asıl amacı olan 3-D bilgisayar görmesi çeşitli yapay sinir ağı yapılarıyla sağlanmıştır.

Anahtar Kelimeler : Yapay sinir ağları, 3-D bilgisayar görmesi, moment envaryantları.

ABSTRACT

The main focus of this thesis is to establish a system that can recognize the completely visible 3-D solid objects of the real life. The synthesis of analyzing two-dimensional images that are taken from different angle of views of the objects is the main process that leads us to achieve our objective. The selection of “Good” features those satisfying two requirements (small intraclass invariance, large interclass separation) is a crucial step in the process since the next stage sees only these features and acts upon them. A flexible recognition system that can compute the best features for a high classification is investigated in this study.

Acquisition and converting the analog image signals of real life is the first step. Total 216 images were taken from three different objects are in order to create input space of our problem. The second is reprocessing which involves removing noise, enhancing the picture, and segmentation for eliminating the unnecessary parts. After operations of thresholding and creating contour images binary coded boundary images are obtained. Only the pixels on the boundary of the object were used to calculate the moment invariants. For object recognition regardless of its orientation, size and position, feature vectors were computed with the assistance of nonlinear moment invariant functions. After an efficient feature extraction, the main focus of this thesis, recognition performance of artificial classifiers in conjunction with moment-based feature sets, is introduced.

Keywords: Artificial Neural Networks, 3-D computer vision, moment invariants.

1. GİRİŞ

Örneklerin otomatik olarak tanınması, tanımlanması, sınıflandırılması ve kümelendirilmesi biyoloji, psikoloji, tıp, pazarlama, bilgisayar görmesi, yapay zeka, gibi çok çeşitli mühendislik ve bilimsel disiplinler açısından çok önemli problemlerdir. Watanabe, örnek'i şu şekilde tanımlar, 'örnek kaostan tersine bağımsız olarak mevcut olan, isimlendirilebilen şeydir'. Mesela örnek; bir parmak izi görüntüsü, el yazısı, düzensiz bir kelime, bir insan yüzü veya ses işareti olabilir. (Jain vd., 2000)

Sunulmuş bir örnek için tanıma/sınıflandırılma işlemi eğitici veya eğitici olmayan sınıflandırma işlevlerinden birini içerebilir. Eğitici sınıflandırmada (örn, diskriminant analizi) örnekler daha önce tanımlanmış bir sınıfın üyeleri olarak belirtilmiştir. Karar verme, önceden sınıfı bilinen örneklerle bakılarak daha sonraki örneklerin hangi gruba ait olduğunu çıkış olarak bildirmektir. Eğitici olmayan sınıflandırmada ise (örn, kümeleme) örnekler o ana kadar bilinmeyen sınıfın üyeleri olarak ayrılır.

Buradaki tanıma problemi, sınıflandırma veya kategorize etme işi olarak ortaya çıkmaktadır ki sınıflar daha önceden sistemi oluşturan kişi tarafından belirlenmiştir (eğitici öğrenme) veya örnekler arasındaki benzerlikleri öğrenme yoluyla oluşturulur (eğitici olmayan öğrenme). Örnek tanıma işleminin kullanım alanları günümüzde ortaya çıkan sadece uğraştırıcı değil aynı zamanda yoğun hesap yükü gerektiren uygulamalarla yenilenmiştir. Bu uygulamalar veri işleme (bir örneğin tanınması), doküman sınıflandırılması (text dokümanların taranması), finansal tahmin, multimedya verilerinin organizasyonu ve eldesi, ve biyometrik (yüz ve parmak izi gibi ayırt edici fiziksel özelliklere dayalı kişisel tanıma) gibi konuları içermektedir. Picard'ın geliştirerek duyuşal hesaplama ismini verdiği örnek tanıma uygulamasında bilgisayara, insan hislerine akıllı karşılık verme ve mantıklı karar vermesine yardımcı olacak mekanizmaları geliştirmesi için, hisleri tanıma ve ifade edebilme yeteneği kazandırmıştır. Bu uygulamaların genel karakteristiği örnekleri sayısal olarak tanımlamaya yarayan çok boyutlu verilerin direkt kullanılmasının sistem uzmanları tarafından benimsenmemesi ve veri işleme algoritmalarıyla optimize etme ve sıkıştırmaya gidilmiş olmasıdır.

Devasa boyutlu veri kümelerinin daha hızlı işlenmesini kolaylaştıran hızla gelişen ve ulaşılması kolay bilgisayar gücü, aynı zamanda veri analizi ve sınıflandırılması için gerekli olan çok parametrelili ve ayrık metodların da kullanılabilmesine imkan tanımıştır. İlave olarak

çok büyük veri tabanlarının elde edilebilirliği ve katı performans gerekliliklerine (hız, doğruluk ve fiyat) bağlı olarak otomatik örnek tanıma sistemlerine talepte hızla artmaktadır. Günümüz uygulamalarının çoğunda tek başına hiç bir yaklaşımın optimal olmayacağı, ve birleşik metotların ve yaklaşımların kullanılması gereği açık olarak görülmüştür. Bunun sonucunda da bir kaç algılama modulünün ve sınıflayıcının bütünleştirilerek kullanımı örnek tanımda yaygınlaştırılmıştır.

Örnek tanıma yüksek seviyeli görüntü analiz sistemlerinin temelini teşkil eden bir işlemdir. Bilgisayar görmesi sisteminin nihai hedefi, verilen görüntünün analiz edilmesi ve görüntünün bileşenlerinin tanınmasıdır. Benzer sistemler robotik, askeri, keşif, uzaktan algılama, belge işleme ve endüstriyel otomasyon gibi bir çok alanda kullanımdadır. Bu çalışmada da olduğu gibi genel bilgisayar görmesi ile ilgili sistemlerin çoğu dört bloktan oluşan genel bir ortak yapıyı taşımak zorundadırlar.

- 1) Görüntü elde edimi, sayısal hesaplamalar için elde edilen görüntülerin sayı dizilerine dönüştürülmesidir.
- 2) Elde edilen görüntünün ön işlemlere tabi tutulması, ilgilenilen örneği arka plandan soyutlamak, gürültüyü azaltmak, örneği normalize etmek gibi çeşitli işlemler gerçekleştirerek örneği tanımaya katkı sağlayacak yoğun bilgiler elde etmektir.
- 3) Özellik çıkarımı, işlenen görüntülerden gereksiz parçaları atıp görüntüyü ifade edecek sayısal bilgiler elde edilmesidir. Seçilen özellik vektörü kümesinin sınıf ayrıştırma aşamasında orjinal veriyi en iyi şekilde temsil edebilmesi gereklidir. En iyi özellik vektörlerinin çıkarımı sınıflandırmayı etkileyecek bir aşamadır. Özellik vektörlerinin şu nitelikleri taşıması gereklidir;

- a) Aynı sınıftan örnekler için küçük sayısal değer farklılıkları,

- b) Farklı sınıftan örnekler için büyük sayısal değer farklılıkları.

Ek olarak görüntülerin boyutlarına, yapılan döndürme ve öteleme etkilerine duyarlı olmayan esnek bir tanıma sistemi oluşturmak lazımdır. Bu gereksinimler özellik vektörleri için ölçekten, döndürmeden, ötelemeden bağımsızlık olarak ifade edilir.

- 4) Bilgisayar görmesi için oluşturulan yapının son bloğu sınıflandırmadır. Bu görüntü tanıma sisteminin son aşamasıdır ki etiketi bilinmeyen bir görüntü veya nesne için elde edilmiş özellik vektörlerini sunarak daha önce sunulmuş etiketi belli örneklerin özellik vektörleriyle eğitilmiş ağdan sınıflama sonucu almaktır.

Problem tabanının yapısı algılayıcıların, ön işleme tekniklerinin, sunuş yapısının ve karar verme modelinin sistem tasarımcısı tarafından kullanımını zorunlu kılar. Genel olarak kabul edilen bir gerçek vardır ki, iyi tanımlanmış, sınırları yeterince belirlenmiş tanıma problemi (örneklerde küçük sınıf içi değişimleri ve büyük sınıflar arası değişimler) az yer kaplayan örnek sunumlarına ve basit karar verme mekanizmalarının yeterliliğine götürecektir. Daha önceden elde edilmiş veri kümesini (eğitme kümesi) kullanarak öğrenme, birçok örnek tanıma sistemi için ihtiyaç duyulan ve önemli olan bir özelliktir.

Örnek tanıma için en yaygın kullanılan dört yaklaşım aşağıda verilmiştir.

- 1) Şablon eşleme,
- 2) İstatistiksel sınıflandırma,
- 3) Söz dizilimsel veya yapısal eşleme,
- 4) Sinir ağları.

1.1 Şablon eşleme

Örnek tanımaya dair en basit ve en eski yaklaşımlar şablon eşleme yöntemine dayanır. Şablon eşleme basit bir şekilde tanımlanmak istenirse şu soru sorulmalıdır: Görüntü daha önce tanımlanan bir nesne içeriyor mu? Eşleme iki yapı arasında benzerlikleri (noktalar, eğriler, veya şekiller) belirlemeye yarayan genel operasyonlardan oluşan örnek tanıma sistemidir. Şablon eşleme yönteminde bir şablon (genellikle 2-D formunda) veya tanınacak örneğin bir prototipi mevcuttur. Tanımlanacak örnek, şablonun mümkün olabilecek tüm duruş pozisyonları (döndürme, öteleme) ve ölçek değişimleri göz önünde bulundurularak eşleştirilmeye çalışılır. Dijital şekil tanıma probleminde bu karşılaştırma önceden tespit edilmiş şablonun şekil üzerinde sistematik olarak dolaştırılmasıyla olur. Şablondaki her pikselin değeri sayfadaki, kendisine karşılık düşen pikselin değerine eşit olduğu zaman görüntüdeki cisim tespit edilmiş olur. Tarama sırasında sayfa biterse o zaman cisim görüntüde yok demektir. Genelde cisim bir görüntünün içinde arandığı için şablon sayfadan küçük olur. Daha gerçekçi bir şablon eşleme için benzerlik ölçütü belirlemeye gidilmesi daha mantıklıdır, zira hatasız bir eşlemeye ulaşmak mümkün değildir. Benzerlik ölçütü olarak genellikle korelasyon bağıntısı kullanılır. Şablon eşleme yöntemi için büyük bir hesaplama yükü oluşmaktadır. Günümüzde daha hızlı işlemcilerin olması bu yaklaşımı daha olası hale getirebilmektedir. Yukarıda bahsedildiği gibi değişmez, katı bir yapıya sahip bu sistem halen bazı uygulama alanlarında etkin olarak kullanılmakla beraber birçok da dezavantajı beraberinde getirmiştir. Örneğin görüntüleme süreci, bakış açısı değişimi ve sınıflar içi

örneklerdeki büyük farklılıklara bağlı olarak tanıma problemine çözüm getirmede başarısız olmaktadır.

1.2 İstatistiksel Yaklaşım

İstatistiksel yaklaşımda, tüm örnekler d adet özellik veya ölçüm değerlerinden oluşan ve d boyutlu uzaydaki noktalar şeklinde gözlemlenecek şekilde ifade edilir. Nihai hedef, farklı kategorilere ait özellik vektörlerinden oluşan noktaların d -boyutlu uzayda mümkün olduğu kadar belirli bir bölgede toplanmış olması ve sınıfların ayık olarak dağılmasını sağlayan özellik vektörlerinin iyi seçilmesidir. Örneklerin sunum uzayının (özellik vektörleri) etkin olarak belirlenmiş olması, farklı sınıflardan örneklerin iyi bir şekilde ayrılabilir olmasını sağlar. Her sınıftan elde edilmiş eğitim örnekleri kümesini kullanarak özellik uzayında farklı sınıftan örneklerin ayrılabilmesi için karar verme sınırları oluşturma istatistiksel yaklaşımın amacıdır. İstatistiksel karar teorilerine dayanan yaklaşımda, karar sınırları her sınıfa ait örneklerin olasılık dağılımları kullanılarak belirlenir.

Ayrıca sınıflandırmaya diskriminant analiz tabanlı yaklaşım da getirilebilir: bu durumda ilk olarak karar verme sınırlarının (örn, lineer veya quadratik) parametrik şekilleri kesinleştirilir, daha sonra giriş örneklerinin sınıflandırılmasıyla kesinleştirilen şekilde en iyi karar sınırları bulunur. Bu karar sınırları örneğin karesel hata kriteri kullanılarak oluşturulabilir. İstatistiksel yaklaşım Bölüm 4'te ayrıntılı olarak işlenecektir.

1.3 Söz Dizilimsel veya Yapısal Yaklaşım

Birçok tanıma problemi karışık örnekler içerir. Bir örneğin basit alt örneklerden oluştuğunu ve bu alt örneklerinde daha basit alt örneklerden oluştuğunu düşünmek daha uygun olabilir. Tanınması istenen bu en basit temel alt örneklere primitive adı verilir. Verilen karışık örnek bu primitivelerin birbiriyle olan ilişkileri şeklinde sunulur. Söz dizilimsel veya yapısal örnek tanımada örneklerin yapısıyla bir dilin söz dizimi arasında şekilsel bir benzerlik kurulur. Örnekler bir dile ait cümleler, primitiveler de dilin alfabesi olarak kabul edilir. Ve cümleler bir dilbilgisine göre kurulur. Böylece büyük karışık örnekler topluluğu az sayıda primitiveler ve dilbilgisi kurallarıyla ifade edilir. Her örnek için dilbilgisi mevcut eğitim örnekleri kullanılarak çıkarılır.

Yapısal örnek tanıma, sınıflamaya ek olarak, örneklerin primitiveler tarafından nasıl oluşturulduğuna dair bir tanımlama sağlar. Bu paradigma, EKG dalga şekilleri, dokusal

görüntüler gibi belirli yapıları olan kurallar kümesi şeklinde elde edilebilen örnekler için yaygın olarak kullanılmıştır. Gürültülü örneklerin söz konusu olması durumunda ise sınıflama ve eğitim örneklerinden dilbilgisi çıkarımı zorlaşmaktadır. Söz dizilimsel yaklaşım ve istatistiksel yaklaşımının birleştirilmesiyle daha nitelikli dilbilgisi oluşturma fikri de ortaya atılmıştır.

1.4 Sinir Ağları

Çok fazla sayıda bağlantılara sahip çok sayıda basit işlem birimleriyle yoğun bir şekilde paralel hesaplama yapabilen yapılar olarak tanımlanırlar. Sinir ağı modelleri belli organizasyon ilkelerini (öğrenme, genelleştirme, adapte olabilme, hata toleransı, hesaplama) gerçekleştirme amacıyla kurulmuştur. Belli girişlere cevap üretmek üzere meydana getirilmiş hücre düğümleri ve yönlü ağırlıklı bağlantılardan oluşur. Sinir ağlarının ana karakteristiği kompleks lineer olmayan giriş-çıkış bağıntılarını öğrenebilmesi, yapısını gelen örnek dizilerine göre değiştirmesi ve adapte olabilmesidir.

Sinir ağları, özellik çıkarımı (saklı katmanları kullanarak) ve sınıflandırma için kullanılır. Ek olarak, yapay sinir ağları mimarisi üzerinde kurulan özellik çıkarımı ve sınıflandırma algoritmaları , donanımsal olarak da tümleştirilebilir. Yapısı tamamen farklı olarak gözüксе de yapay sinir ağ modelleri dolaylı olarak istatistiksel örnek tanıma modellerine eşit veya denktir.

Bir çok çalışmada yapay sinir ağlarıyla istatistiksel örnek tanıma arasındaki ilişkiyi incelenmiştir. Anderson'a göre, "Yapay sinir ağları amatörlerin istatistiksel örnek tanıma modelleridir. Birçok yapay sinir ağı istatistiksel kazanımlarını kullanıcıdan saklar" (Jain vd., 2000). Bu benzerliklerin yanında yapay sinir ağları sınıflandırma ve örnek çıkarımı için birleştirilmiş yaklaşımlar kullanma ve nonlineer çözümleri bulmak için esnek prosedürlere sahip olma gibi avantajlar sunar. Yapay sinir ağları Bölüm 5'te ayrıntılı olarak işlenecektir.

1.5 Bilgisayar Görmesi Problemine Getirilmiş Genel Çözümler

Robotik uygulamalarda, robotun çalışma ortamında bulunan endüstriyel parçaların yerini belirleyip ne olduklarını anlaması ya da bir mobil robotun hareket yönünü belirleyebilmesi için yol olan bölgeleri yol olmayan bölgelerden ayırması ve hareket planını buna göre yapması istenebilir. Ne tür bir uygulama olursa olsun oluşturulacak tanıma sistemi benzer yapılar taşır.

Cisim tanımda en önemli adımlardan biri Hu'nun (1962) moment envaryantları üzerine çalışmalarıyla atılmıştır. Geometrik moment envaryantları ve bunlardan türetilen ortogonal moment envaryantları kullanılarak bir çok çalışma oluşturulmuştur.

(Zion vd., 1999)'da tatlı su balıkçılığı endüstrisinde kullanılmak üzere balık sınıflandırma sistemi oluşturulmuştur. Şeffaf bir gölgeden geçen balıklara ait görüntüler kamerayla alınmış, bu görüntülerden Hu moment fonksiyonları elde edilerek oluşturulan özellik vektörleriyle diskriminant analizi yapılarak sınıflandırma gerçekleştirilmiştir.

(Shutler vd., 2001)'da ortogonal moment envaryantlarına hız bileşeni de eklenerek cisimlere ait zamanla değişen yapısal bilgilerin tanımlanabilmesi sağlanmıştır. Farklı insanlara ait yürüyüş görüntüleri kullanılarak özellik vektörleri oluşturulmuş ve k-NN metoduyla sınıflandırma işlemi gerçekleştirilmiştir.

Moment fonksiyonları kullanılarak görüntülerden özellik çıkarımı işlemi, cisim tanımaya ek olarak pek çok karakter tanıma uygulamalarında da etkin sonuçlar vermiştir. (Khotanzad vd., 1990)'da el yazısı İngilizce yazı karakterleri kümesi kullanılarak Hu moment fonksiyonları, Zernike moment fonksiyonlarıyla elde edilen özellik vektörleri çeşitli istatistiksel sınıflandırma yapıları ve yapay sinir ağları kullanılarak sınıflandırılmıştır.

(Wechsler ve Zimmerman, 1998)'da cisimlere ait görüntüleri kutupsal koordinatlara haritalama işlemi, dağılmış çağrışimli belleklerle (distributed associative memory) birleştirilerek cisimleri dönme ve ölçekleme değişimlerinden bağımsız olarak tanıyan bir sistem geliştirilmiştir.

(Ayache ve Faugeras, 1986)'da robot kolu görmesi uygulaması işlenmiştir. Cisimlere ait lokal özellikler kullanılarak diskriminant analizi hipotezi geliştirilmiş, bu hipotezler kullanılarak başka cisimler arasına karışmış hedef cisimlerin tanınması sağlanmıştır.

Bu çalışmada ise temel problem üç boyutlu fiziksel dünyada bulunan katı cisimlerin tanınmasıdır. Elde edilen görüntülerden faydalanarak sınıflayıcılar aracılığıyla hangi cismin görüntü içerisinde olduğunu tespit edebilecek bir bilgisayar görmesi sistemi geliştirilmiştir. Görüntüler ideal olmayan fiziksel dünyaya ait cisimlere aittir. Cismin pozisyonunun sabit olmadığı, algılayıcının optik eksenine göre farklı mesafe ve açılarda durumları olabileceği göz

önüne alınarak ölçek, dönme ve ötelemeye bağımsız bir görme sistemi oluşturulmaktadır. Görüntü işleme, özellik çıkarımı ve sınıflandırma programları için Matlab paket programlarından yararlanılmıştır.



2. GÖRÜNTÜ İŞLEME

2.1 Dijital Görüntü

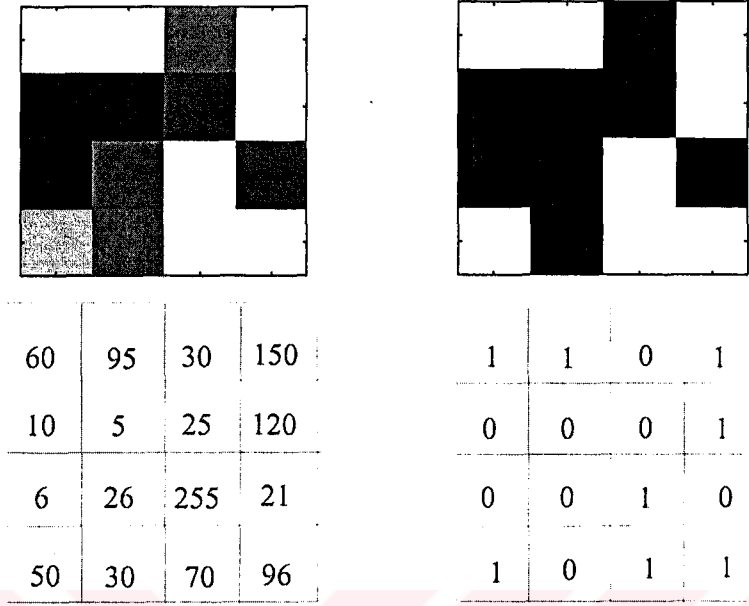
Tabiattaki işaretleri üç ana grup altında toplamak mümkündür. İlk grupta analog veya diğer bir deyişle sürekli uzay işaretleri bulunmaktadır. Bu işaretler uzayda ve genlikte süreklidirler. Görüntü ve ses işaretleri gibi pratikteki işaretlerin çoğu bu gruba girmektedir. İkinci gruba giren işaretler ayrık uzay işaretleridir. Bu işaretler uzayda ayrık genlikte olan sürekli işaretlerdir. Ayrık uzay işaretlerini elde etmek için sıkça kullanılan bir yol analog işaretleri örneklemektir. Üçüncü gruba giren işaretler sayısal veya ayrık işaretlerdir. Sayısal işaretleri elde etmenin bir yolu ayrık uzay işaretlerinin kuantalanmasıdır. Sayısal işaretler noktalardan oluşan dizilerdir. Sayısal işaretleri temel alan işaret işleme kavramlarının gelişiminde genlikler üzerine yapılan işlemler bulunmaktadır. Bu işlemlerin karmaşası yüzünden çoğu işaret işleme kavramı ayrık uzay işaretlerini temel alarak geliştirilmiştir (Ölmez, 2002).

Normalde analog özellik gösteren görüntü işaretleri sayısal ortama taşınırken ayrık işaretlere dönüştürülmektedir. Bu çalışmada ilgilendiğimiz ayrık görüntü işaretleri veya dijital görüntü işaretleri ise normal görüntülerin hem uzaysal koordinatlarda hem de aydınlık düzeyi olarak ayrıştırılmış bir $f(x,y)$ görüntüsüdür. $f(x,y)$ görüntünün x ve y uzaysal koordinatlarındaki aydınlık seviyesi veya gri seviyesidir. Geleneksel olarak koordinatların görüntünün sol üst köşesinden başlayıp x için sağa y için sola doğru arttığı varsayılmaktadır. Bu şekilde tanımlanan koordinatları ve gri seviyesi belli noktaların her birine ise piksel adı verilir.

Zorunluluk olmamakla beraber dijital görüntüleri genellikle 512x512, 256x256 pikselden oluşan matrisler şeklinde 2'nin katları boyutlu oluşturmak görüntü işleme algoritmaları için kolaylık sağlamaktadır.

Analog görüntülerdeki noktaların aydınlık seviyelerinin dijital görüntülere aktarılması konusunda ise değişik dijital görüntü çeşidi oluşturma normları getirilmiştir. Bu çalışmada ise ilk olarak oluşturulan dijital görüntülerde pikseller 0-255 (siyah-beyaz) arasındaki gri seviyesi değerlerini almaktadır. Daha sonra ise sadece 0 siyah ve 1 beyaz aydınlık seviyesi değerlerine sahip piksellerden oluşan ikili (binary) görüntüler oluşturulmaktadır.

Sonuç olarak görüntüyü sıra ve sütunları bir görüntü noktasını temsil eden matris olarak düşünebiliriz (Şekil 2.1).



Şekil 2.1 Gri renk seviyeleri ve binary renk seviyeleriyle kuantalanmış görüntüler

2.2 Görüntü İyileştirme

Görüntülerin izleyicilere görünümünü iyileştirmek veya diğer görüntü işleme sistemlerinin performanslarını arttırmak için işlenmesidir. Metotları ve amaçları uygulamaya göre değişir. Makine ile nesne tanıma gibi diğer bazı uygulamalarda ise, makine performansını arttırmak için görüntü önceden işlenebilir. Kullanılan kriterler genellikle duruma özgü olup bu kriterlerin kullanışlı genel ifadelerle çevrilmeleri zordur. Dolayısıyla, iyileştirme algoritmaları çoğunlukla basit ve niteleyici olurlar. Bununla birlikte, verilen herhangi bir uygulamada bir görüntü sınıfı için iyi performans gösteren bir algoritma, diğer görüntü sınıfları için iyi sonuç vermeyebilir. Görüntü işleme algoritmalarındaki ortak net yarar görüntüye dair faydalı bilgilerin anlamsız bilgiler arasından elde edilmesidir.

2.2.1 Noktasal İşlemler

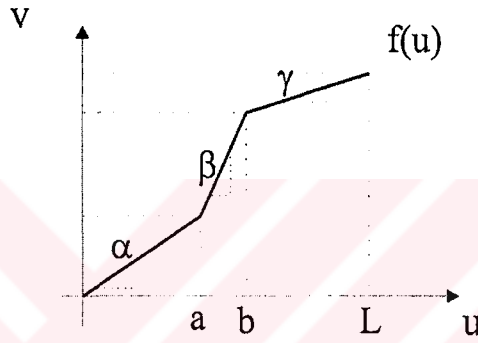
Görüntüde u bir pikselin gri seviyesi ve $u \in [0, L]$ olmak üzere, u değerini $v = f(u)$ şeklinde v gri seviyesine haritalayan $f(\cdot)$ fonksiyonları noktasal işlemleri ifade eder (Jain, 1998).

2.2.1.1 Kontrast Yayma

Düşük kontrastlı görüntülerin oluşması, genellikle zayıf veya düzgün olmayan aydınlatma koşulları, ya da görüntü algılayıcısının lineer olmaması veya dinamik aralığın küçük olması gibi durumlarla ilgilidir.

$$v = \begin{cases} \alpha.u, & 0 \leq u < a \\ \beta.(u - a) + v_a, & a \leq u < b \\ \gamma.(u - b) + v_b, & b \leq u < L \end{cases} \quad (2.1)$$

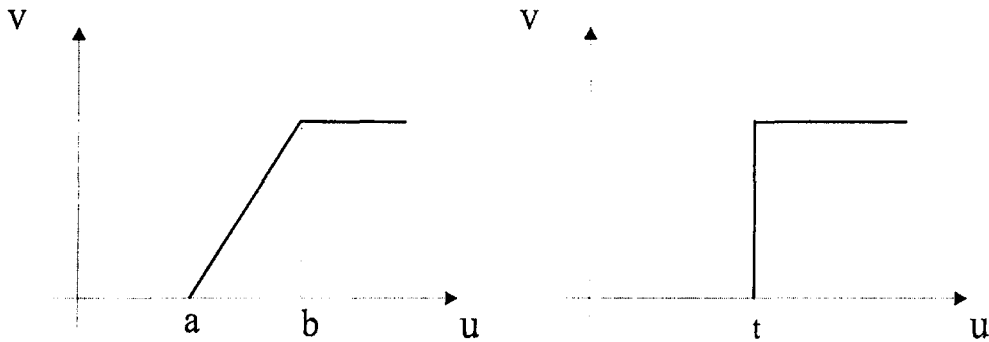
şeklinde belirli değerlerdeki aydınlık seviyelerinin yeterli hale getirilmesidir (Şekil 2.2).



Şekil 2.2 Kontrast yayma işlemi

2.2.1.2 Kesme ve Eşikleme

$\alpha = \gamma = 0$ olacak şekilde, özel bir kontrast yayma işlemine *kesme* denir. Bu işlem, giriş işaretinin $[a, b]$ aralığında olduğu bilindiğinde gürültü giderme amacı ile kullanılır (Şekil 2.3). Eşikleme, ise özel bir kesme işlemidir. Bu işlemin sonunda görüntüde belirlenen t değerinin altındaki aydınlık seviyeleri sıfıra çekilmiş olur (Şekil 2.3).



Şekil 2.3 Kesme ve eşikleme

2.2.1.3 Sayısal Negatif

Negatif bir görüntü, gri seviyeleri aşağıdaki işlem uyarınca ölçekleyerek elde edilir. Özellikle medikal görüntüler için bu işlem yaygın olarak kullanılır

$$v = L - u \quad (2.2)$$

2.2.1.4 Aralık Sıkıştırma

Bazen görüntü verisinin dinamik aralığı çok geniş olabilir. Böyle durumlarda dinamik aralık, aşağıdaki logaritmik dönüşüm uyarınca sıkıştırılabilir;

$$v = c \log_{10}(1 + |u|) \quad (2.3)$$

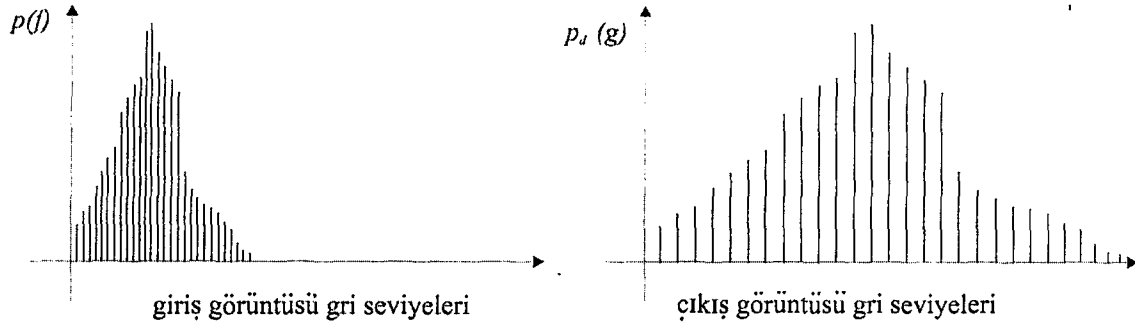
Burada c , ölçekleme sabitidir. Bu dönüşüm, büyük genlikli pikseller ile karşılaştırıldığında küçük genlikli pikselleri iyileştirir.

2.2.1.5 Histogram Dengeleme

Gri Skala Modifikasyonu, görüntünün dinamik aralığını veya kontrastını modifiye etmek için kullanılan etkili bir metottur. Bir $f(n_1, n_2)$ giriş görüntüsüne ait gri skala veya şiddet seviyesi, özel bir dönüşüme göre modifiye edilir. f giriş yoğunluğunu, g çıkış yoğunluğuna dönüştüren $g = T[f]$ dönüşümü genellikle bir tablo veya nokta diyagramı ile verilir.

Tipik uygulamalarda iyi bir dönüşüm, giriş görüntüsünün histogramını belirleyip bunun karakteri üzerinde çalışılarak tanımlanabilir.

Giriş görüntüsüne ait histogram $p(f)$, çıkış görüntüsüne ait istenilen histogram, $p_d(g)$ ile gösterilir (Şekil 2.4). $p_d(g)$, dinamik aralığın ortası civarlarında bir maksimuma sahiptir. Şiddet arttıkça veya azaldıkça yavaşça düşer. Amacımız verilen bir görüntü için sonuç çıkış görüntüsünün $p_d(g)$ 'ye benzer bir histogramının olmasını sağlayacak transfer fonksiyonunu belirlemektir. Bu işlem, temel olasılık teorisinin bir problemi olarak ifade edilebilir. Özel olarak $p(f)$ ve $p_d(g)$ histogramları, sırası ile, f ve g rasgele değişkenlerinin ölçeklendirilmiş olasılık yoğunluk fonksiyonları olarak gösterilebilir. Yapılacak işlem, $p(f)$ 'nin $p_d(g)$ 'ye eşit veya çok yakın olmasını sağlayacak şekilde $g = T[f]$ dönüşümünün bulunmasıdır. $T[f]$ burada, f 'in monoton azalmayan bir fonksiyonu olmalıdır.



Şekil 2.4 Histogram dengeleme

2.2.2 Uzaysal İşlemler

Bu tür işlemler genellikle giriş piksellerinin yerel komşuluğundaki pikseller üzerinde yapılmaktadır. Genellikle de görüntünün uzaysal maske olarak ifade edilen filtrelerle konvolusyonu cevabı çıkış olarak alınır.

2.2.2.1 Komşuluk Ortalaması

Daha önceden tanımlanan komşuluk içinde her pikselin gri seviye değeri o pikselin komşuluğundaki piksellerin gri seviye değerlerinin ortalaması olur. w komşuluğunun merkezindeki $y(m, n)$ pikseli için çıkış değeri;

$$v(m, n) = \frac{1}{N_w} \sum_k \sum_l y(m - k, n - l) \quad (k \& l \in w) \quad (2.4)$$

olarak ifade edilir.

2.2.2.2 Medyan Filtreleme

Giriş pikseli merkez seçilerek yapılan medyan filtreleme şu şekilde ifade edilebilir;

$$v(m, n) = \text{median}[y(m - k, n - l), (k \& l) \in w]. \quad (2.5)$$

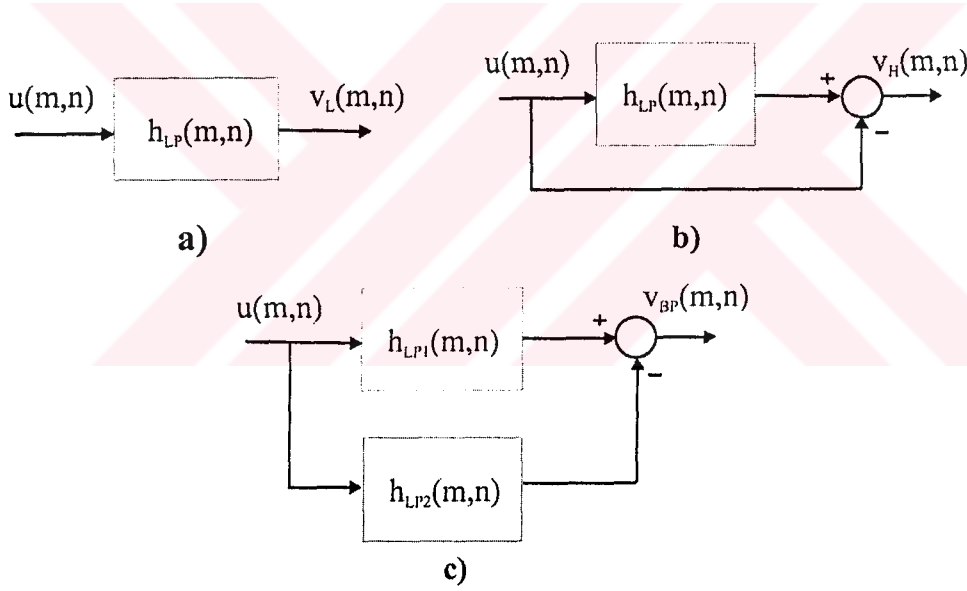
Algoritmada belirlenen komşuluk içerisindeki pikseller artan veya azalan sıraya göre dizilir ve ortadaki gri seviyesi değeri giriş pikseline atanır. Bu işlemi yapabilmek için w komşuluğu boyutları 3, 5 gibi tek sayılardan seçilir.

2.2.2.3 Alçak, Yüksek, Band Geçiren Filtrelemeler

Filtrelemeler çeşitli bağıntılar neticesinde bulunmuş uzaysal filtreler ve görüntünün konvolusyonu yoluyla yapılmaktadır.

Alçak geçiren filtre interpolasyon ve pürüzsüzleştirme işlemlerinde kullanılır. Görüntünün keskin hatları bozulur. Yüksek geçiren filtreleme bir işarettaki yüksek frekanslı bileşenleri geçirirken, alçak frekanslı bileşenleri bastırır. Kenarlar ve ince detaylar yüksek frekans bileşenlerinin görüntüdeki ana kaynakları olduğundan dolayı, yüksek geçiren filtreleme genellikle yerel kontrastı yükseltir ve görüntüyü keskinleştirir. Bazı endüstriyel uygulamalarda da görüntülerin iletilmesi sırasında band sınırlılığı söz konusu ise görüntünün yüksek geçiren filtre çıkışını iletmek daha uygun olur (Jain, 1998).

Band geçiren filtreleme ise görüntüdeki kenarların iyileştirilmesinde yarar sağlamaktadır.



Şekil 2.5 a)Alçak geçiren filtreleme b)Yüksek geçiren filtreleme
c) Band geçiren filtreleme

2.2.3 Dönüşüm İşlemleri

Görüntü dönüşümü, görüntüye ait verilerden çeşitli ortogonal seri açılımları yapılarak oluşturulan katsayılarla görüntüyü farklı bir domende ifade etmek olarak açıklanabilir. Görüntü dönüşümleri kullanılarak görüntü iyileştirme tekniklerinde hafızasız işlemler gerçekleştirilir. Domende çeşitli işlemler yapıldıktan sonra dönüşümün tersi uygulanarak çıkış görüntüleri elde edilir. Sözelimi bir görüntüye DCT (ayrık kosinüs dönüşümü), DFT (ayrık

fourier dönüşümü), Haadamard dönüşümleri vb., uygulandıktan sonra elde edilen katsayılar matrisi üzerinde genelleştirilmiş lineer filtreleme, kök filtreleme, homomorfik filtreleme işlemleri yapılabilir. Daha sonra ters dönüşüm işlemleriyle çıkış elde edilebilir.

2.3 Görüntü Bölütleme

Bir görme sisteminin en önemli parçası görüntüden ilgilenilen yapıyı çıkarmaktır. Analiz ve tanıma süreçleri için cisimlerin sınırlarının tespit edilmesi gerekir. Temel olarak görüntü bölütlemeye yönelik iki yaklaşım vardır. İlgili yapının dış hatlarına ilişkin doğrudan bilgi üreten yaklaşım, sınır temelli yaklaşım olarak; yapının bulunduğu bölgeye ilişkin bilgi üreten yaklaşım ise, bölge temelli yaklaşım olarak adlandırılır. İlk yaklaşımda bölütleme çıkışı yapının sınırlarıdır (Ölmez, 2002).

2.3.1 Sınır Temelli Yaklaşımlar

Yukarıda da bahsedildiği gibi, sınır temelli yaklaşımlarda bir görüntü içindeki ilgilendiğimiz yapının dış hatlarını bulmaya yönelik işlemler yapılır. Bu yaklaşımların en önemlilerinden birisi kenar belirlemedir.

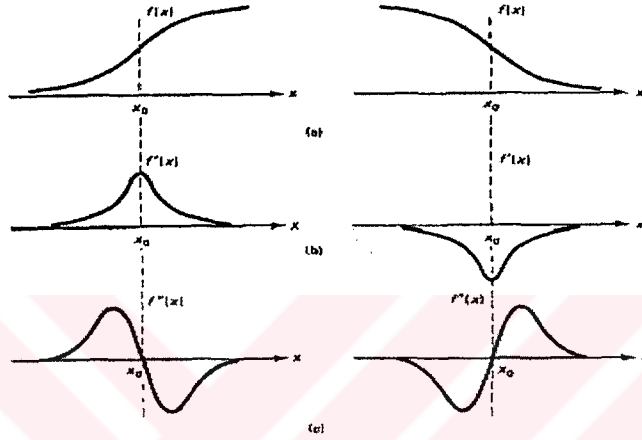
2.3.1.1 Kenar Belirleme

Bir görüntüde *kenar*, görüntünün yüzey yansıması, parlaklık, görünen yüzeylerin izleyiciden uzaklığı gibi fiziksel görünüşünde anlamlı bir değişimin meydana geldiği sınır veya dış hattır. Fiziksel görünüşteki değişimler; görüntü şiddetindeki değişimler, rengindeki değişimler gibi farklı şekillerde kendini belli eder. Burada sadece görüntü şiddetindeki değişimler üzerinde durulacaktır. Kenar bulma işlemi birçok uygulamada faydalıdır. Örneğin, nesne tanıma gibi tipik bir görüntü anlama uygulamasında önemli bir adım, görüntüyü içindeki değişik nesnelere bağlı olarak bölmelere ayırmaktır. Kenar bulma görüntü bölütlemesinde genellikle ilk adımdır. Sadece kenarlardan oluşan bir görüntünün yüksek seviyede anlaşılır olduğu bilinmektedir. Görüntüde fiziksel bir değişimin anlamlılığı uygulamaya göre değişir. Nesne içindeki ek detayları ifade eden dış hatlar kenar olarak düşünülmeyebilir. Dolayısıyla bir kenar tanımlanamazsa uygulamanın alanı dışındadır. Bununla birlikte, geniş bir uygulama alanında faydalı olan kenar belirleyici algoritmalar geliştirilmiştir.

2.3.1.1.1 Gradyan Temelli Metotlar

Şekil 2.6'da gösterildiği gibi tipik bir 1-boyutlu kenarı temsil eden analog $f(x)$ fonksiyonunu düşünelim. Tipik problemlerde, şekildeki x_0 değerini bir kenar noktası olarak düşünmek

mantıklıdır. x_0 'ı belirlemenin bir yolu birinci türev $f'(x)$ veya ikinci türev $f''(x)$ 'i hesaplamaktır. Şekil.2.6.b ve Şekil.2.6.c $f'(x)$ ve $f''(x)$ fonksiyonlarını göstermektedir. Bu şekillerden x_0 'ın değeri $f'(x)$ 'in yerel ekstremumuna (maksimum veya minimum) bakarak veya $f''(x)$ 'in işaretini değiştirdiği sıfır geçiş noktasını bularak belirlenebilir. Bu metotta $f''(x)$ 'in özelliklerine dikkat edilmemektedir. Olası kenar noktası x_0 'ı belirlemenin yanında $f'(x)$, kenarın yönünü ve kuvvetini tahmin etmede de kullanılabilir.

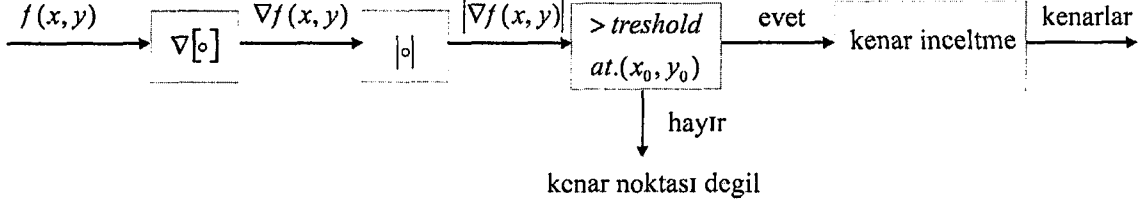


Şekil 2.6 Tipik bir 1-boyutlu kenar için (a) $f(x)$ (b) $f'(x)$ (c) $f''(x)$

$f'(x)$ 'in 2-boyutlu $f'(x, y)$, fonksiyonuna genellenmesi aşağıdaki gibi verilen $\nabla f(x, y)$ gradyanı ile olur;

$$\nabla f(x, y) = \frac{\partial f(x, y)}{\partial x} \hat{i}_x + \frac{\partial f(x, y)}{\partial y} \hat{i}_y \quad (2.6)$$

Yukarıdaki gözlemlere dayandırılarak kenar bulmaya yönelik bir yaklaşım, Şekil 2.7'de gösterilen sistemi kullanmaktır. Eğer $|\nabla f(x, y)|$ tahmini kenar noktasında belirlenen yönlerin herhangi birisi boyunca yerel maksimum ise, tahmini kenar noktası olarak alınır. $|\nabla f(x, y)|$ belli bir eşikten büyük olacak şekilde, kenar olarak bulunursa kenarlar çizgi değil de şerit halinde belirir. Kenar aday noktalarının oluşturduğu şeritten kenar çizgisini ayırt etme işlemine kenar inceltme adı verilir.



Şekil 2.7 Gradyan temelli kenar bulma algoritması

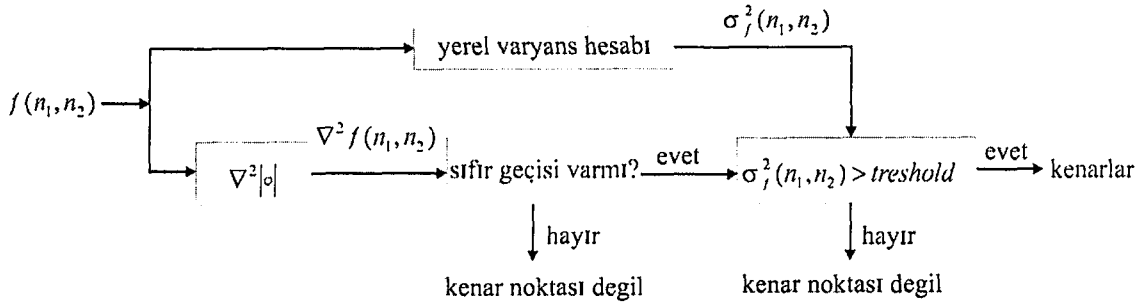
$|\nabla f(x, y)|$ yerine $|\partial f(x, y)/\partial x|$ kullanılırsa yatay yöndeki $|\partial f(x, y)/\partial y|$ kullanılırsa dikey yöndeki kenarları bulabilme imkanı oluşur.

2.3.1.1.2 Laplasyen Temelli Metotlar

Kenar belirleme amacı ile 2-boyutlu $f(x, y)$ 'nin Laplasyeni $\nabla^2 f(x, y)$ şu şekilde verilir;

$$\nabla^2 f(x, y) = \nabla(\nabla f(x, y)) = \frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2} \quad (2.7)$$

Kenarları $\nabla^2 f(x, y)$ veya $\nabla^2 f(n_1, n_2)$ 'in sıfır geçiş noktalarına bakarak belirlediğimizden, $f(n_1, n_2)$ 'deki en küçük bir hareketlilik yanlış kenar hatları oluşturacaktır. Bu yanlış hatların birçoğunu gidermenin bir yolu, kenar noktasında yerel varyansın tatmin edici seviyede büyük olup olmadığına bakmaktır.



Şekil 2.8 Laplasyen temelli kenar bulma algoritması

Şekil 2.8'deki sistem, bir gradyan temelli metot olarak da değerlendirilebilir. Yerel varyans $\sigma_f^2(n_1, n_2)$ gradyan genliği ile yakın ilgilidir. $\sigma_f^2(n_1, n_2)$ 'yi bir eşik ile karşılaştırmak, gradyan genliğini bir eşik ile karşılaştırmak gibidir. $\nabla^2 f(n_1, n_2)$ 'nin kenarda sıfırdan geçiyor olma zorunluluğu, kenar inceltme şeklinde yorumlanabilir. Bu yorum ile, Şekil 2.8'deki

sistemi, önce $\sigma_f^2(n_1, n_2)$ 'yi hesaplayıp, $\nabla^2 f(n_1, n_2)$ 'nin sıfır geçiş noktalarını, sadece $\sigma_f^2(n_1, n_2)$ 'nin belli bir eşikten büyük olduğu noktalarda bularak da gerçekleştirebiliriz.

2.3.2 Bölge Temelli Metodlar

R bütün görüntüyü temsil eden pikseller kümesi olsun. Bu durumda görüntü parçalamayı R 'yi n alt kümeye bölme işlemi bölge temelli bölütlemedir. Bunun için aşağıdaki koşullar sağlanmalıdır;

- Görüntüdeki her piksel sınıflandırılmalıdır,
- Her bölgedeki pikseller arasında bağlantı olmalıdır,
- Bir piksel birden fazla bölgeye ait olmamalıdır,
- Her piksel belirli bir özelliğinden dolayı belirli bir bölgeye dahil edilmiştir,
- Her bölge diğerinden farklıdır.

2.3.2.1 Piksel Toplanması

Pikselleri bir araya getirerek daha büyük bölgeler oluşturmaya dayanan bir sistemdir. Pikseller aydınlık düzeyi, renk veya daha başka özelliklerine bakılarak birleştirilir. Şekil 2.9'da piksel toplanması işlemi gösterilmiştir. Orijinal görüntüde piksel toplanması işlemi sonrasında ışık şiddeti değeri 3'ten küçük olan piksellerle oluşturulan a değerli pikseller içeren bölge ile 3'ten büyük olan piksellerle oluşturulan b değerli pikseller içeren bölgeler verilmektedir (Üstün, 1999).

1	0	0	0	a	a	a	a
0	1	2	1	a	a	a	a
2	1	3	7	a	a	b	b
1	2	9	6	a	a	b	b

Şekil 2.9 Piksel toplanması

2.4 Görüntü Analizi ve Tanımlanması

İnsan örnek sınıflandırması yaparken fiziksel ve topolojik özelliklerin ikisini birden kullanır. Çünkü insan gözü en kolay bu özellikleri seçer. Mesela elma ve muz sınıflandırması yaparken karar verme mekanizmamız renk, şekil, sertlik, ve doku gibi özellikleri kullanır.

Bununla birlikte makineler sınıflandırma aracı olarak kullanıldığında tanımlayıcıların sadece bunlarla sınırlı kalması mantıklı bir çözüm olmaz. Örnek tanıma sistemlerindeki makinelerin matematiksel ve istatistiksel özellikleri elde edebilmesi ve kullanabilmesi en uygun olanıdır. Aşağıda tanımlayıcılar genel hatlarıyla verilecektir.

2.4.1 Uzaysal Tanımlayıcılar

2.4.1.1 Genliğe Dayalı Özellik Çıkarımı

Bir cisme ait en basit belki de en önemli özellikler cisme ait yansıtma, geçirgenlik, multispektral cevap gibi bazı fiziksel özelliklerin genlik değerleridir. Bazı basit noktasal işlemlerle görüntüye ait genlik özellikleri çıkarılabilir. X-ışınları ile görüntülemeye vücuda ait doku kütlelerinin ışığı emme farklılıklarından yararlanılarak oluşturulan gri renk seviyelerinde piksellerden oluşan görüntü teşhis aşamasında uzman kişilere yardımcı olur. Atmosfere ait kızılötesi (IR) görüntülerde ise ısı farklılıklarının görüntülenmesi sağlanır ve bulutların ayrıştırılması işlemi gerçekleştirilebilir (Jain, 1998).

2.4.2 Dönüşüm Tanımlayıcıları

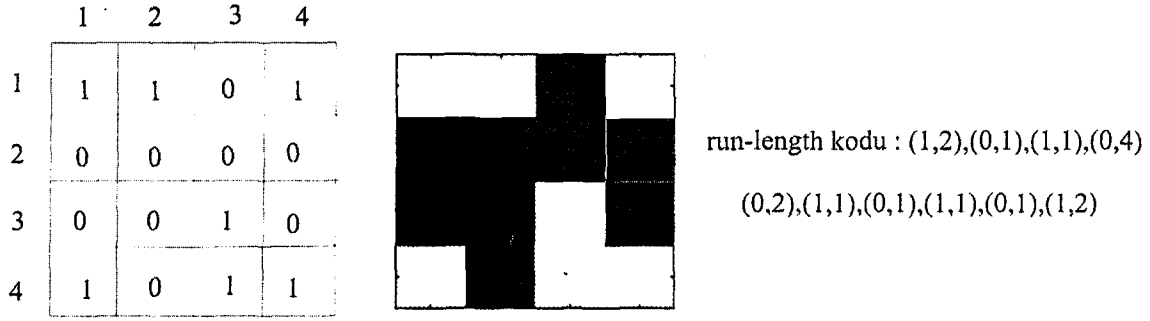
Görüntü dönüşümleri veriye ait frekans domen bilgilerini elde etmeye yarar. Dönüşüm özellikleri seçilen dönüşüm uzayında bölgesel filtreleme yoluyla yapılır. Bölgesel filtre aynı zamanda özellik maskesi olarak da bilinir. Yaygın olarak yüksek frekans özellikleri kenar ve sınır belirlemede kullanılır. Fourier, Haadamard, Haar potansiyel olarak özellik çıkarımında etkin dönüşümlerdir.

2.4.3 Sınır Tanımlayıcıları

Nesne sınırlarının uygun bir şekilde tanımlanması şekil analizi ve sentezinde önemli bir noktadır. Şekil analizi genel bir tema içindeki nesnelerin belirlenmesi ve tanınmasında kullanılır. Video oyunları, çizgi filmler gibi görüntü simülasyonlarında şekil sentezinden yararlanılmaktadır.

2.4.3.1 Zincir Kodları

Zincir kodlamada uygun sınır pikselleri arasına yönlü vektörler yerleştirme metodu uygulanır. Genel kullanımda 8 yönde oluşturulmuş yön vektörleri 3 bit ile kodlanarak zincir kodları oluşturulur. Bir başlangıç noktası belirlenir. Bu başlangıç noktasından itibaren uygun sınır piksellerine uygun yön vektörleriyle ulaşılarak başlangıç noktasına dönülür. Başlangıç



Şekil 2.11 Run-length kodlama

2.4.5 Moment Tanımlayıcıları

Moment tanımlayıcıları veya moment envaryantları birçok örnek tanıma uygulamasında kullanıla gelmiştir. Görüntü veya şekil özellik envaryantları, görüntü veya şekilde boyut, pozisyon, rotasyon değişiklikleri ve kombinasyonları oluşmasına karşılık değişmezlik özelliği gösteren görüntüye özel katsayılardır.

Tezin ana teması olan Moment tanımlayıcıları Bölüm 3'te ayrıntılı olarak anlatılacaktır.

3. MOMENT ENVARYANTLARI

Sayırsız rnek tanıma uygulamalarında en can alıcı işlem görüntü verisinden önemli özelliklerin çıkarımıdır ki görüntünün mantıklı bir şekilde temsil edilebilmesi, yorumlanabilmesi ve anlaşılması ancak bu şekilde mümkün olabilir. Zarfların tasnif edilmesi, etiket okuma işlemleri, iç organların boyutlarının ölçülmesi ve kromozom analizi gibi birçok uygulamada görüntü analizi işlemlerinin makineler tarafından otomatik olarak yapıldığını görme imkanını bulabiliriz.

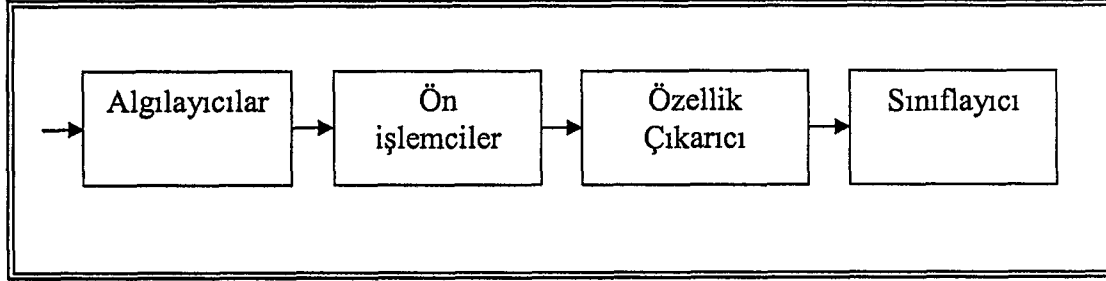
Bir cismin uzaysal özellikleri denince cismin gri renk seviyesi ve bunların tümleşik olasılık dağılımları anlaşılabilir. Bir cismin tanımaya yönelik belki de en yararlı özellikleri ışığı yansıtmaya yada geçirme kabiliyeti, veya multispektral yanıt verebilmesi gibi fiziksel özelliklerinin genliğidir. Örneğin tıbbi X-ışınları vasıtasıyla oluşturulmuş görüntülerde gri renk seviyesi genlikleri, yapıların ışığı emme karakteristikleri neticesinde şekillenir ve kemiklerin dokulardan veya sağlıklı dokuların hastalıklı dokulardan ayırt edilebilmesini sağlar.

Dönüşüm özellikleri görüntünün başka bir domene dönüştürülmesiyle elde edilir. En yaygın tekniklerden biri Fourier tanımlayıcılarıdır ki Fourier serileri açılımındaki katsayılardan elde edilir. Ayrık kosinüs dönüşümü, ayrık sinüs dönüşümü gibi başka frekans domeni tabanlı görüntü tanımlama yaklaşımları da vardır. Frekans domeni tabanlı dönüşümler daha çok görüntü sıkıştırma algoritmalarında kullanılırlar.

Bir görüntüyü daha az sayıda tanımlayıcı veya özellikle ifade edebilmek örnek tanıma ve veri sıkıştırma gibi uygulamalarda çok önemlidir. Özellik çıkarıcılar yapısal ve istatistiksel olmak üzere iki kategoriye ayrılırlar. Kontur belirleme ve zincir kodları oluşturma tekniklerine dayanan yapısal özellik çıkarımı örnek tanımada önemli bir kullanım alanına sahiptir. İstatistiksel yaklaşımlardan auto regressive (AR) modeller, fourier analizi, ve moment envaryantları etkin özellik çıkarım tekniklerindendir.

Bir örnek tanıma sisteminin basit modeli Şekil 3.1’de gösterilmiştir. Bu modelin ilk basamağında örneklerin orijinal domeninden sistemin anlayabileceği kabul edilebilir bir domene dönüştürme işleminin yapılmasını sağlayan algılayıcılar yer almaktadır. Algılayıcının çıkışı genellikle karmaşık sayısal bilgiler yükü topluluğudur. Ön işlemciler, algılayıcılardan

sonra gelir ve özellik çıkarımı için verileri düzenlemede kullanılırlar. Sayısal filtreleme, eşiklendirme, ve kontur belirleme ön işlemcilerin yapabileceği işlemlerdendir. Özellik çıkarıcı ise işlenmiş örneklerden sayısal değerlerle ifade edilen bir dizi özelliği içeren vektör oluşturma işleminde kullanılır. En son safha ise etiketi bilinmeyen özellik vektörleriyle ifade edilmiş örnekleri, daha önceden kaydedilmiş n sınıftan örnekleri içeren örnekleri referans olarak sınıflandırmaktır.



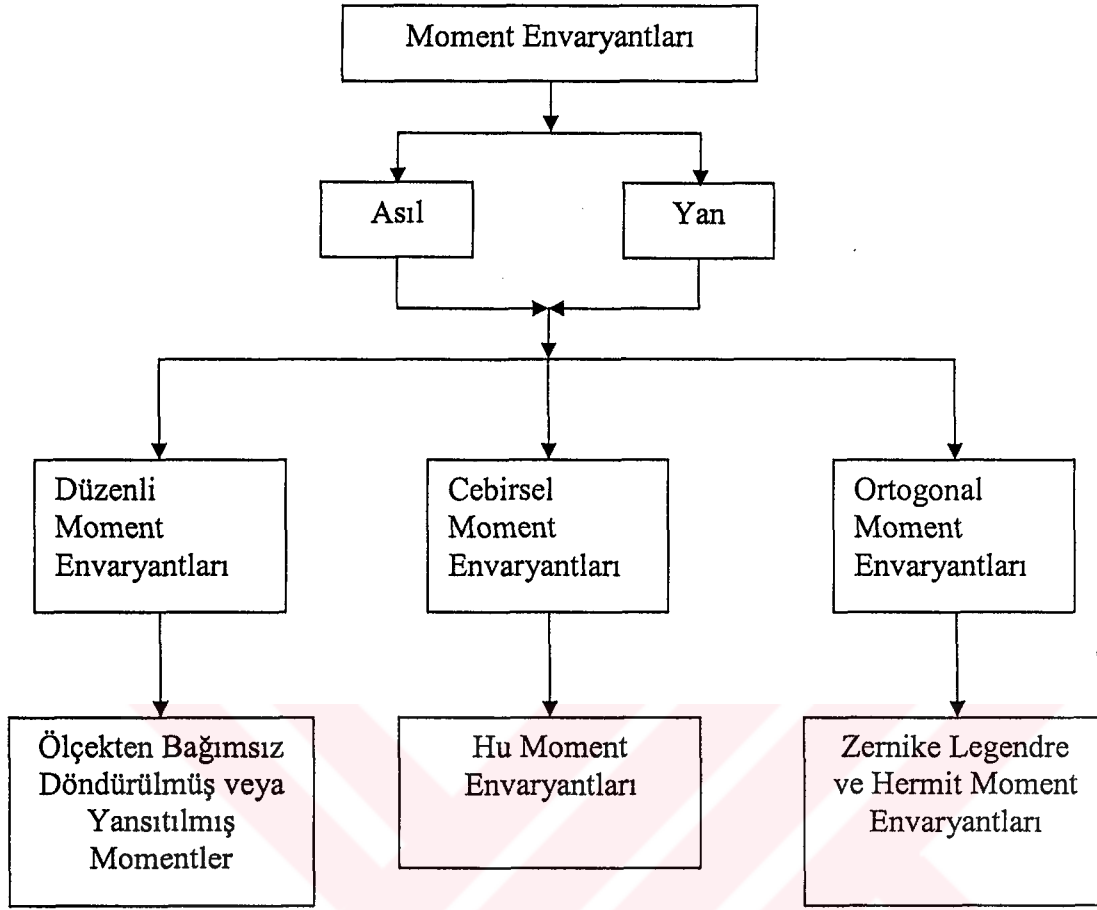
Şekil 3.1 Örnek tanıma işleminin temel elemanları

3.1 Moment Tanımlayıcıları Kullanarak Özellik Çıkarımı

Moment tanımlayıcıları veya moment envaryantları birçok örnek tanıma uygulamasında yaygın olarak kullanılmıştır. Hu'nun (1962) cebirsel envaryantlar teorisini kullanarak envaryantlar kümesi türetmesinden sonra momentlerin şekil tanımada kullanılabilirliği fikri yaygınlaşmıştır. Görüntü veya şekil özellik envaryantları, görüntü veya şekilde aşağıdaki değişmeler ve kombinasyonları oluşmasına karşılık değişmezlik özelliği gösteren görüntüye özel katsayılardır,

- 1) Boyut değişimleri,
- 2) Pozisyon değişimleri,
- 3) Rotasyon değişimleri,
- 4) Yansıma.

Moment envaryantları özellikle 2-d görüntülerin özellik çıkarımında çok etkili bir yöntemdir. Moment envaryantları, yan ve asıl moment envaryantları olmak üzere iki alt kategoriye ayrılabilir. Yan moment envaryantları boyut, öteleme ve döndürmeden bağımsızken, asıl moment envaryantları bu değişmelere ek olarak yansımadan da bağımsızdır. Şekil 3.2 bu moment envaryantlarının sınıflandırılmasını göstermektedir.



Şekil 3.2 Moment envaryantlarının sınıflandırılması

Hu (1962) görüntülerin moment temsiliyle temel dönüşümlerden, öteleme, ölçek, döndürme değişimlerinin nasıl elde edileceğini göstermiştir. Döndürmeden bağımsız moment katsayılarının (düzenli momentler) nonlinear mutlak kombinasyonlarından oluşan moment envaryantları görüşünü sunmuştur. Lineer dönüşümler altında değişmezlik özelliğini koruyan cebirsel tanımlardan oluşan cebirsel envaryantlar teorisi temelli gözlemler yapmıştır. Bu gözlemler sonucunda döndürmeden bağımsızlığı sağlamanın en zor işlem olduğunu tespit etmiş ve döndürmeden bağımsız moment envaryantlarını hesaplamak için iki farklı metot önermiştir. İlk metot 'temel eksenler metodu' olarak bilinen 2. dereceden momentleri kullanarak hedef nesneyi kuşatan elipsin birincil ve ikincil eksenlerini hesaplamayı, ikinci metot ise moment büyüklüklerinde normalizasyon işlemi temelli dönmeden bağımsızlığı sağlamaktadır. Tanımlanan 3. dereceye kadar olan geometrik momentlerden oluşan moment envaryantları farklı şekil ve görüntü yapılarını tanımda kullanılmaktadır.

3.2 Düzenli Moment Envaryantları (RMI)

Momentler görüntülere uygulandıklarında eksenlere bağlı olarak görüntü içeriğini yansıtır. Görüntü hakkında yerel bilgi elde etmek üzere bulunmuşlardır. (Khotanzad ve Lu, 1990)

$f(x, y)$ fonksiyonu için momentler,

$$m_{pq} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} (x)^p \cdot (y)^q f(x, y) \cdot dx dy \quad (3.1)$$

2 boyutlu, $M \times M$ boyutlarında $(x, y = 0, \dots, M-1)$, gri renk seviye fonksiyonu $f(x, y)$ olan bir görüntü fonksiyonuna uygulanırsa,

$$m_{pq} = \sum_{x=0}^{M-1} \sum_{y=0}^{M-1} (x)^p \cdot (y)^q f(x, y) \quad p, q = 0, 1, 2, 3, \dots \quad (3.2)$$

3.2.1 Moment Envaryantları Dönüşümleri

3.2.1.1 Öteleme

$f(x, y)$ fonksiyonunu (a, b) kadar ötelendiğinde momentler;

$$m'_{pq} = \sum_x \sum_y (x+a)^p \cdot (y+b)^q f(x, y) \quad (3.3)$$

şeklinde olacaktır. $f(x, y)$ fonksiyonunun orijinal momentleri cinsinden bu eşitliği şu şekilde ifade edebiliriz;

$$m'_{pq} = \sum_{r=0}^p \sum_{s=0}^q \binom{p}{r} \binom{q}{s} (a)^{p-r} (b)^{q-s} m_{rs} \quad (3.4)$$

Eğer ötelemeye karşı normalizasyon olarak ifade edebileceğimiz $a = -\bar{x}$ ve $b = -\bar{y}$ işlemi yapılırsa ötelemeye bağımlılık ortadan kalkacaktır. Şeklin $\bar{x}$ ve $\bar{y}$ merkez koordinatları momentler cinsinden şu şekilde hesaplanmıştır;

$$\bar{x} = \frac{m_{10}}{m_{00}} \text{ ve } \bar{y} = \frac{m_{01}}{m_{00}} \quad (3.5)$$

Bu durumda merkezi momentler olarak ifade edilen momentler;

$$m'_{pq} = \mu_{pq} = \sum_{r=0}^p \sum_{s=0}^q \binom{p}{r} \binom{q}{s} (-\bar{x})^p \cdot (-\bar{y})^q m_{rs} \quad (3.6)$$

veya,

$$m'_{pq} = \mu_{pq} = \sum_x \sum_y (x - \bar{x})^p \cdot (y - \bar{y})^q f(x, y) \quad (3.7)$$

olacaktır.

3.2.1.2 Ölçekleme

β gibi bir katsayıya bağlı olarak ölçek değişimi yapılmış $f(x, y)$ fonksiyonuna ait merkezi momentler;

$$m''_{pq} = \sum_x \sum_y (x)^p \cdot (y)^q f\left(\frac{x}{\beta}, \frac{y}{\beta}\right) \quad (3.8)$$

olacaktır. Orijinal merkezi momentler cinsinden

$$\mu'_{pq} = \beta^{p+q+2} \mu_{pq} \quad (3.9)$$

olarak ifade edilebilir. $\mu'_{00} = 1$ eşitliği kullanılarak, ölçek normalizasyonu yapılırsa

$$\beta = \frac{1}{\sqrt{\mu_{00}}}, \quad \mu'_{pq} = \frac{1}{\sqrt{\mu_{00}}}^{p+q+2} \mu_{pq} \quad (3.10)$$

bulunur. Buradan ölçekten bağımsız merkezi momentler şu şekilde ifade edilir;

$$\eta_{pq} = \frac{\mu_{pq}}{\mu'_{00}}, \quad \gamma = \frac{p+q}{2} + 1. \quad (3.11)$$

3.2.1.3 Döndürme

$f(x, y)$ fonksiyonu orijin etrafında θ derecelik açıyla dönmesiyle momentler ;

$$m_{pq}^m = \sum_x \sum_y (x \cos \theta + y \sin \theta)^p \cdot (y \cos \theta - x \sin \theta)^q \cdot f(x, y) \quad (3.12)$$

ifadesine eşit olacaktır. Orijinal $f(x, y)$ fonksiyonunun momentleri cinsinden;

$$m_{pq}^m = \sum_{r=0}^p \sum_{s=0}^q \binom{p}{r} \binom{q}{s} (-1)^{q-s} \cdot (\cos \theta)^{p-r+s} (\sin \theta)^{q+r-s} m_{p+q-r-s, r+s} \quad (3.13)$$

olarak verilmiştir.

Burada döndürmeden bağımsızlığın sağlanması için ‘temel eksenler’ metodu kullanılmıştır. Temel eksenler momentleri, merkezi momentlerin μ_{11} sıfır olana kadar eksenlerin döndürülmesiyle elde edilmiştir. Bu işlem sonrasında orijinal eksenlerle yeni eksenler arasında θ açısı oluşmaktadır. Temel eksenler ;

$$\tan 2\theta = \frac{2\mu_{11}}{\mu_{20} - \mu_{02}} \quad (3.14)$$

ve

$$\theta = \frac{1}{2} \tan^{-1} \frac{2\mu_{11}}{\mu_{20} - \mu_{02}} \quad (3.15)$$

bağıntılarıyla bulunur. Olası değişimler sonrasındaki düzenli moment envaryantları RMI şu şekilde yazılabilir;

$$(RMI)_{pq} = \sum_{r=0}^p \sum_{s=0}^q \binom{p}{r} \binom{q}{s} (-1)^{q-s} \cdot (\cos \theta)^{p-r+s} (\sin \theta)^{q+r-s} \eta_{p+q-r-s, r+s} \cdot \quad (3.16)$$

3.2.1.4 Yansıma

Y eksenini boyunca bir yansıma, y ordinatının üssü olan q katsayısına bağlı olarak eski momentlerden bazılarının negatif işarete sahip olmasına neden olur.

$$m_{pq}^{(m)} = \sum_x \sum_y (x)^p \cdot (y)^q f(x, -y) = (-1)^q \cdot m_{pq} \quad (3.17)$$

benzer sonuç X eksenini boyunca yapılan yansıma için de geçerlidir.

Hu (1962) bu gözlemlerden ve hesaplamalardan sonra 3. dereceye kadar döndürme ve ötelemeden bağımsız olan merkezi momentlerden oluşan moment fonksiyonları kümesi türetmiştir. Bunlar aşağıda verilmiştir;

$$\begin{aligned} M_1 &= (\mu_{20} + \mu_{02}) \\ M_2 &= (\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2 \\ M_3 &= (\mu_{30} - 3\mu_{12})^2 + (3\mu_{21} - \mu_{03})^2 \\ M_4 &= (\mu_{30} + \mu_{12})^2 + (\mu_{21} + \mu_{03})^2 \\ M_5 &= (\mu_{30} - 3\mu_{12})(\mu_{30} + \mu_{12})[(\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2] + \\ &\quad (3\mu_{21} - \mu_{03})(\mu_{21} + \mu_{03})[3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2] \\ M_6 &= (\mu_{20} - \mu_{02})[(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2] + 4\mu_{11}(\mu_{30} + \mu_{12})(\mu_{21} + \mu_{03}) \\ M_7 &= (3\mu_{21} - \mu_{03})(\mu_{30} + \mu_{12})[(\mu_{30} + \mu_{12})^2 - 3(\mu_{21} - \mu_{03})^2] - \\ &\quad = (\mu_{30} + 3\mu_{12})(\mu_{21} + \mu_{03})[3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2] \end{aligned} \quad (3.18)$$

Hu'nun moment envaryantları artan moment derecesiyle büyüyen karmaşıklığı nedeniyle hesaplama zorluğuna sahiptir. Bir diğer sorun ise ortogonal olmayan fonksiyonlar kümesinden oluşmaları ve cismin şekli hakkında eksik bilgi taşımalarıdır.

3.3 Ortogonal Moment Envaryantları

3.3.1 Zernike Moment Envaryantları

Düzenli moment fonksiyonları $f(x, y)$ fonksiyonunun $x^p \cdot y^q$ çarpanı üzerine izdüşümü şeklinde ifade edilebilir. Ama $x^p \cdot y^q$ taban kümesi ortogonal değildir. Ek olarak ortogonal temelli fonksiyonlar gerçekleştirilecek bilgi fazlalığının indirgenmesinin optimal hale getirilmesinden yoksundur.

Zernike birim dairenin ($x^2 + y^2 \leq 1$) iç bölgesi üzerinde tamamen ortogonal küme oluşturan karmaşık polinomlar kümesini sundu (Khotanzad ve Lu, 1990; Liao ve Pawlak, 1998) ;

$$V_{nm}(x, y) = V_{nm}(\rho, \theta) = R_{nm}(\rho) \exp(jm\theta) , \quad n = 0, 1, 2, \dots, \infty \quad (3.19)$$

$n - |m|$ çift ve $|m| \leq n$ olmak zorundadır.

$\rho, (x, y)$ pikselinin orijinden uzaklığı, θ ise ρ ve x eksenleri arasında saat yönünün tersinde kalan açı olmak üzere, R_{nm} radyal polinomu;

$$R_{nm} = \sum_{s=0}^{n-|m|/2} \frac{(-1)^s [(n-s)!] \rho^{n-2s}}{s! \left(\frac{n+|m|}{2} - s\right)! \left(\frac{n-|m|}{2} - s\right)!} \quad (3.20)$$

şeklindedir.

$$\int \int_{x^2+y^2} [V_{nm}^*(x, y)] V_{pq}(x, y) . dx dy = \frac{\pi}{n+1} \delta_{np} \delta_{mq} \quad (3.21)$$

$$\delta_{ab} = \begin{cases} 1 & a = b \\ 0 & a \neq b \end{cases} \quad (3.22)$$

yukarıdaki eşitliklerle $V_{nm}(x, y)$ polinomlarının ortogonal olduğu ispatlanmaktadır.

Zernike momentlerini görüntü fonksiyonunun bu ortogonal fonksiyonlar üzerine izdüşümü olarak ifade edebiliriz. n . dereceden m kez tekrarlamalı sürekli $f(x, y)$ görüntü fonksiyonları için Zernike momentleri;

$$A_{nm} = \frac{n+1}{\pi} \int \int_{x^2+y^2 \leq 1} f(x, y) V_{nm}^*(\rho, \theta) dx dy \quad (3.23)$$

olarak ifade edilir. Fiziksel dünyadaki sayısal bir görüntü için ise integralin yerini ayrık toplamalar alacaktır;

$$A_{nm} = \frac{n+1}{\pi} \sum_x \sum_y f(x, y) V_{nm}^*(\rho, \theta), (x^2 + y^2 \leq 1). \quad (3.24)$$

Verilen sayısal görüntü için zernike momentleri hesaplanırken, görüntünü merkezi orjin olarak alınır ve piksel koordinatları birim daireye göre haritalanır. Birim daire dışındaki pikseller için hesaplama yapılmaz. Ayrıca $A_{n,-m} = A_{nm}^*$ 'dir.

Zernike momentleriyle tanımlanan özellikler bu momentlerin dönme ile ilgili özellikleri kullanılarak türetilir. ϕ 'lık bir açı kadar dönüldüğünü varsayalım. Yeni ve eski görüntülerin zernike momentleri arasında;

$$A'_{nm} = A_{nm} \cdot \exp(-jm\phi) \quad (3.25)$$

bağıntısından anlaşılacağı gibi, zernike momentlerinin basit rotasyonel dönüşüm özellikleri vardır. Zernike momentlerinin $|A_{nm}|$ genlik değerleri dönme ile değişmez.

Ölçekten ve ötelemeden bağımsızlığı sağlam için ise bu değişkenlere göre normalizasyon yapıp daha sonra zernike momentlerinin hesaplanması yapılmalıdır. Bu normalizasyonlar düzenli momentler kullanılarak yapılır.

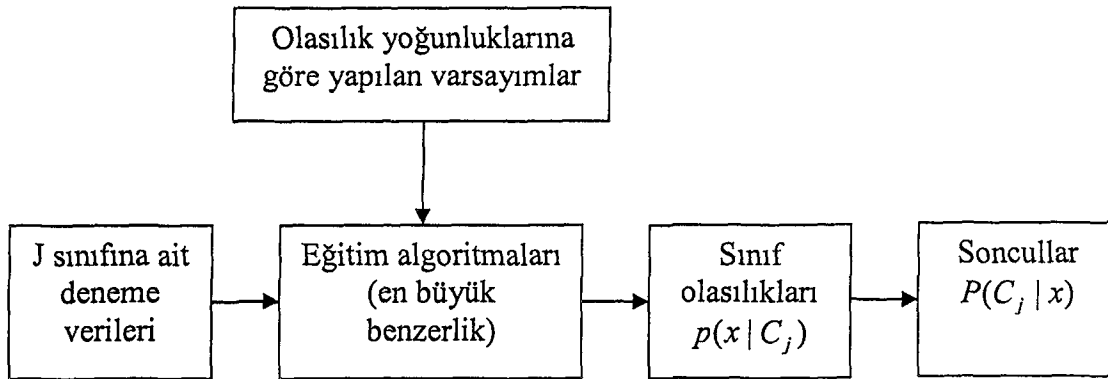
Ötelemeden bağımsızlık $f(x, y)$ fonksiyonunu $f(x + \bar{x}, y + \bar{y})$ haline getirmektir. Böylece m_{01} ve m_{10} momentleri sıfıra eşitlenir. Ölçekten bağımsızlık ise $f(x, y)$ fonksiyonunun $f(\frac{x}{a}, \frac{y}{a})$ fonksiyonuna dönüştürülmesidir ve m'_{00} momentini daha önceden belirlenmiş bir β (yeni görüntüdeki piksellerin sayısı) değerine eşitlemektir. $m'_{00} = a^2 m_{00} = \beta$ ile a değeri hesaplanarak yeni fonksiyon oluşturulur.

4. İSTATİSTİKSEL SINIFLAYICILAR

Ticari örnek tanıma sistemlerinde, istatistiksel örnek tanıma yaygın bir şekilde kullanılmaktadır. bir çok kullanıla gelmiştir. Her bir örnek d boyutlu uzayda bir nokta oluşturan d adet özellik veya nitelikten oluşmaktadır. İstatistiksel karar vermede oluşturulan ve kullanılan genel düşünce, örnekler içeren sınıflar arasında karar sınırları oluşturmaktır. Örnek tanıma ise çok boyutlu $x \in \mathcal{R}^d$ giriş büyüklüğünün C_j , $j = 1, \dots, m$ sınıflar kümesinden birisine dahil etmektir. Bazen de bir veya birden fazla sınıfta olması muhtemel olmayan örnekler için reddetme işlevi eklenebilir. Kısacası sınıflama $\mathcal{R}^d$ düzleminden $\{C_1, C_2, \dots, C_M, C_{REJ}\}$ düzlemine haritalamadır (Alpaydın ve Gürgen, 1998).

İstatistiksel karar verme teorisinde hamlelerin maliyeti olduğu ve minimum maliyete sebep verecek kararların alınması gerektiği varsayımı yapılır. İstenen minimum maliyet için doğru kararların maliyetinin olmadığı, yanlış kararların eşit maliyette olduğu ve ret kararının olmadığı düşünülerek oluşturulan genel ilişkilendirme Bayes karar verme kuralının belirttiği gibi verilen bir girişin en yüksek soncul olasılıkla bir sınıfa dahil edilmesi şu şekilde yapılmalıdır;

$$c = \arg \max_j P(C_j | x) \quad (4.1)$$



Şekil 4.1 İstatistiksel sınıflayıcı modeli

Soncul olasılıklar hiçbir zaman tam olarak bilinemez , tahmin edilirler. $\chi = \{x_i, y_i\}$ örneği verilsin , $i = 1, \dots, n$ ve $y = \{C_1, C_2, \dots, C_M\}$. İstatistiksel örnek tanıma teorisinde (Şekil 4.1), Bayes kuralı;

$P(C_j)$: öncül sınıf olasılığı, C_j 'nin oluşma olasılığı,

$p(x)$: örneğin herhangi bir sınıfa ait olma olasılığı,

$p(x | C_j)$: sınıfa bağlı yoğunluk veya olasılık, ile

$P(C_j | x)$: örnekleme veri setinden tahmin yoluyla belirlendiği için soncul olasılığı olarak ifade edilen büyüklük cinsinden;

$$P(C_j | x) = \frac{p(x | C_j)P(C_j)}{p(x)} = \frac{p(x | C_j)P(C_j)}{\sum_k p(x | C_k)P(C_k)}. \quad (4.2)$$

olarak ifade edilir.

Öncül sınıf olasılığı $P(C_j)$ 'nin tahmini, bir sınıfa ait örneklerin sayısının tüm örneklerin sayısına oranı vasıtasıyla hesaplanır. n_j j sınıfına ait örneklerin sayısı olmak üzere;

$$\sum_j n_j = n \quad (4.3)$$

ise

$$\hat{P}(C_j) = n_j / n \quad (4.4)$$

olarak verilir.

Buradaki en önemli problem, sınıfa bağlı olasılık yoğunlukları $p(x | C_j)$ değerinin tahminidir. Üç temel yaklaşım vardır;

1) Parametrik Metotlar: Sınıfa bağlı yoğunlukların kesin bir parametrik formda olduğu varsayımı yapılır. Bu metotlar, genellikle veriler üzerinde uygun uzaklık metrikleri seçilerek yapılan varsayımlara bağlı olarak oluşturulan uzaklık temelli metotlara indirgenir.

2) Non-parametrik Metotlar: Varsayımların yapılamadığı durumlarda yoğunlukların direkt veriler göz önüne alınarak tahmin edilmesi gerekmektedir. Bu metotlar, Kernel temelli tahmin metotları olarak da bilinirler.

3) Yarı Parametrik Metotlar: Yoğunluklar parametreleri tahmin edilen model birleşimleri şeklinde ifade edilir. Bu yaklaşım, Kohonen'in LVQ'su ve radyal fonksiyon tabanlı ağların (RBF) kümeleme temelli stratejilerine benzer bir karakter çizer (Alpaydın ve Gürgen, 1998).

(4.1) eşitliğiyle verilen karar verme kuralı, giriş uzayını karar bölgeleri olarak ifade edilen her biri m sınıftan birini temsil eden kapalı bölgelere ayırır. Bu bölgelerin sınırları, karar sınırları veya bir sınıfın örneklerini diğer örneklerden ayıran diskriminantlardır. Örnek sınıflama, $g_j(x)$ olarak ifade edilen uygun diskriminant fonksiyonları belirleme olarak düşünülebilir. x özellik vektörü aşağıdaki şartı sağlıyorsa c sınıfına aittir.

$$g_c(x) = \max_j g_j(x) \quad (4.5)$$

veya eğer $i \neq j$ olmak üzere tüm j sınıfları için

$$g_i(x) > g_j(x) \quad (4.6)$$

ise $x, \mathcal{R}^i$ karar bölgesinin içindedir ve C_i sınıfına dahildir. Sonuç olasılık veya bileşenleri bir tür diskriminant fonksiyondur.

Aşağıdaki bağıntıların hepsi eşdeğerdir;

$$g_j(x) = P(C_j | x) \quad (4.7)$$

$$g'_j(x) = p(x | C_j)P(C_j) \quad (4.8)$$

$$g''_j(x) = \ln p(x | C_j) + \ln P(C_j) \quad (4.9)$$

4.1 Parametrik Bayes Sınıflayıcıları

Parametrik sınıflandırma algoritmalarında, sınıflar arasındaki karar sınırlarını (decision boundary) oluşturmak için her sınıfın olasılık dağılımı için varsayım yapılır. Parametrik algoritmalar iki adımdan oluşur, ilki model parametrelerinin tahmin edilmesidir. İkinci adımda diskriminant fonksiyonlarının hesaplanmasıdır. Örnekleri sınıflandırmak için bu diskriminantlardan faydalanılır (Duda vd., 2000).

Bayes sınıflayıcısı tarafından tanımlanan karar bölgelerinin şekli $p(x|C_j)$ için varsayılan yapıya bağlıdır. İdeal sınıf prototipi sınıf verilerinin ortalaması μ_j ile verilir. Varsayılan gürültü karakteristiği ise kovaryans matrisi Σ_j şeklinde ifade edilir. Eğer $p(x|C_j) \sim N(\mu_j, \Sigma_j)$ ise yani x örneğinin C_j sınıfına bağlı yoğunluğu $p(x|C_j)$ çok değişkenli normal fonksiyon özelliği gösteriyorsa;

$$\hat{p}(x|C_j) = \frac{1}{(2\pi)^{d/2} |\Sigma_j|^{1/2}} \exp\left[-(1/2)(x - \mu_j)^T \Sigma_j^{-1} (x - \mu_j)\right] \quad (4.10)$$

ile verilir. İlişki matrisi Σ_j daima simetrik ve pozitif yarı düzlemde dir. (4.9) denklemi uyarınca,

$$g_j(x) = -(1/2) \ln |\Sigma_j| - (1/2)(x - \mu_j)^T \Sigma_j^{-1} (x - \mu_j) - (d/2) \ln 2\pi + \ln P(C_j) \quad (4.11)$$

bulunur. x d boyutlu olduğunda, Ortalama vektörü μ_j d adet, ilişki matrisi Σ_j $d(d+1)/2$ bağımsız parametreden oluşur. Bağımsız parametrelerin fazlalığı sistemi daha kompleks hale getirdiği gibi güvenilir tahmin için daha fazla örneğe ihtiyaç duyuluyor. Bu yüzden varsayımlar yapılırken parametrelerin küçük tutulması amacıyla düzenlemeler yapılır.

4.1.1 Eşit Varyanslı Bağımsız Özellikler

Özelliklerin istatistiksel olarak bağımsız ve tüm boyutlar boyunca aynı varyansa sahip olduğu durumda en basit diskriminant fonksiyonu oluşur. En küçük sınıflandırma yöntemi yaklaşımı bu koşullarda uygulanabilir. Bu durumda kovaryans matrisi diagonal ve birim matris I 'nın σ^2 katı kadardır. Geometrik olarak bu durum örneklerin C_j sınıfını temsil eden μ_j ortalama vektörünü merkez alan eşit boyutta hiperküresel kümeler içerisine girmesi anlamına gelir (Şekil 4.2). Σ_j kovaryans matrisinin determinanı $|\Sigma_j|$ ve tersi olan Σ_j^{-1} değerlerini bulmak kolaylaşır.

$$|\Sigma_j| = \sigma^2, \quad \Sigma_j^{-1} = \left(\frac{1}{\sigma^2}\right) I \quad (4.12)$$

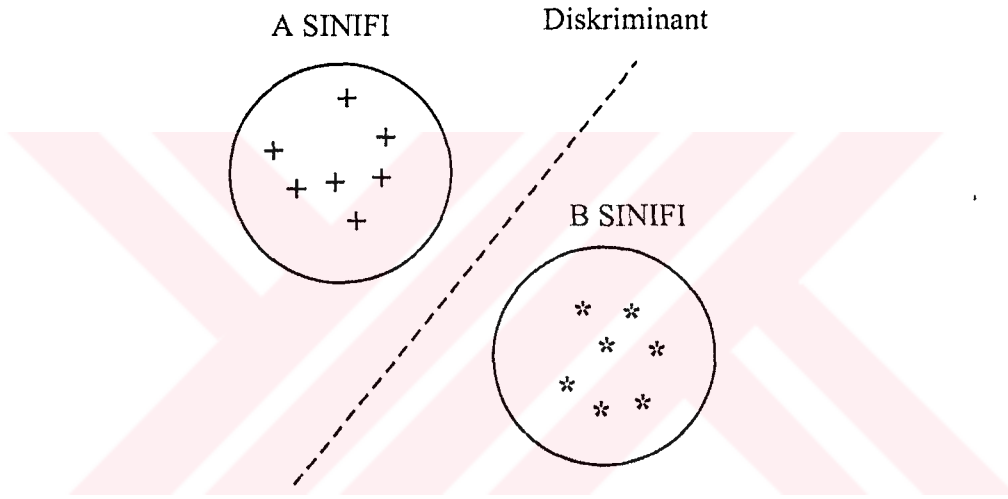
Σ_j ve $(d/2) \ln 2\pi$ katsayıları j 'den bağımsız hale geldikleri için ihmal edilebilirler. Böylece (4.11) bağıntısı şu hale gelir;

$$g_j(x) = -\frac{\|x - \mu_j\|^2}{2\sigma^2} + \ln P(C_j) \quad (4.13)$$

$\|\cdot\|$ öklid normunu ifade eder.

$$\|x - \mu_j\|^2 = (x - \mu_j)'(x - \mu_j). \quad (4.14)$$

Eğer öncül olasılıklar eşit değilse karesel uzaklık $\|x - \mu_j\|^2$, σ^2 varyansı ile normalize edilecek ve $P(C_j)$ offseti eklenecektir. Şekil 4.2'de görüldüğü gibi diskriminantlar lineer formda oluşur.



Şekil 4.2 Eşit varyanslı sıfır kovaryanslı ve lineer diskriminantta sahip sınıflayıcı

Eğer öncül olasılıklar tüm C sınıfları için aynıysa $\ln P(C_j)$ terimi sabit değer haline geldiği için ihmal edilebilir. Bu durumda optimum karar kuralı şu şekilde belirlenir; x özellik vektörünü sınıflandırmak için tüm C sınıflarının ortalama vektörleriyle x arasında öklid uzaklığı hesaplanır. En yakın olarak tespit edilen ortalama vektörünün sınıfı x 'e atanır. m tane sınıf için diskriminant hesaplanmasında ortalama vektörleri için $m.d$ adet ve σ^2 için de 1 tane parametreye ihtiyaç duyulmaktadır.

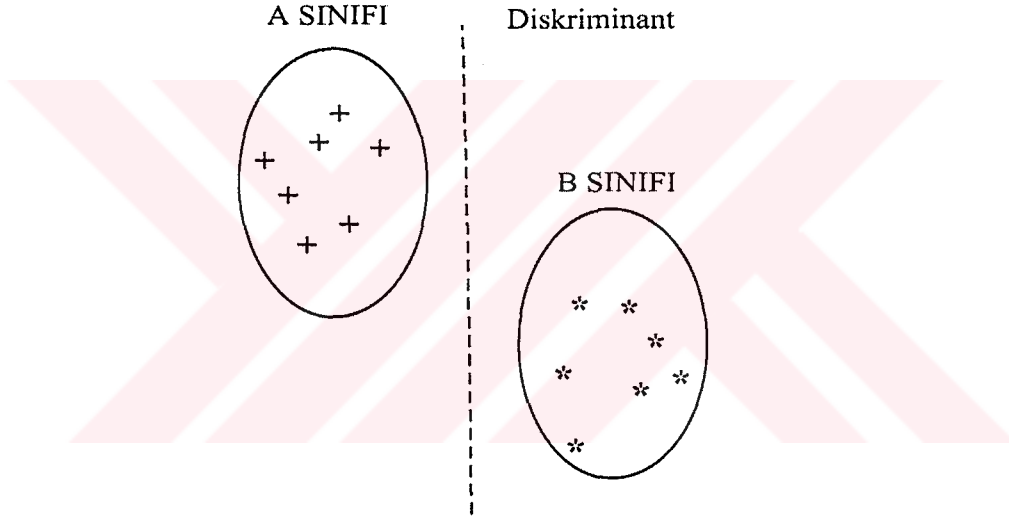
4.1.2 Eşit Olmayan Varyanslı Bağımsız Özellikler

Bir başka durum ise tüm sınıfların kovaryans matrislerinin özdeş fakat keyfi değerlerde olmasıdır. Geometrik olarak bu örneklerin C_j sınıfını temsil eden μ_j ortalama vektörünü

merkez alan eşit boyutta şekildeki hiperelipsoidal kümelerin (cluster) içerisine düşmesi anlamına gelir (şekil 4.3). $\sum_j$ ve $(d/2) \cdot \ln 2\pi$ terimleri j 'den bağımsız hale geldikleri için ihmal edilebilirler. Böylece (4.11) bağıntısı şu hale gelir;

$$g_j(x) = -(1/2)(x - \mu_j)^T \Sigma_j^{-1} (x - \mu_j) + \ln P(C_j). \quad (4.14)$$

Eğer öncül olasılıklar tüm C sınıfları için aynıysa $\ln P(C_j)$ terimi ihmal edilebilir. Bu durumda optimum karar kuralı şu şekilde belirlenir; x özellik vektörünü sınıflandırmak için tüm C sınıflarının ortalama vektörleriyle x arasında Mahalanobis uzaklığı hesaplanır. En yakın olarak tespit edilen ortalama vektörünün sınıfı x 'e atanır. m tane sınıf için diskriminant hesaplanmasında $m \cdot d + d$ adet parametreye ihtiyaç duyulmaktadır.



Şekil 4.3 Farklı varyanslı sınıfların lineer diskriminantla ayrılması

4.1.3 Keyfi Kovaryanslar

Genelde kovaryans matrisi tüm sınıflar için farklıdır. (4.11) eşitliğinde sadece $(d/2) \cdot \ln 2\pi$ terimi ihmal edilebilir ve aşağıdaki eşitlik oluşur;

$$g_j(x) = -(1/2) \cdot \ln |\Sigma_j| - (1/2)(x - \mu_j)^T \Sigma_j^{-1} (x - \mu_j) + \ln P(C_j) \quad (4.15)$$

Karar yüzeyleri hiperdüzlemler, hiperdüzlem çiftleri, hiperhiperelipsoidler, hiperparaboloidler, ve çeşitli hiperhiperboloid türleri şeklinde tanımlanmış hiperkuadratik formlardadır.

Tahmin edilecek toplam parametre sayısı $m.(d + d(d + 1)/2)$ 'dir ve bu sadece yeterince küçük d büyüklükleri ve/veya çok fazla sayıda örnek olması durumunda mümkündür. Fischer'in lineer diskriminant'ı, Friedman'ın regüle edilmiş diskriminant analizi ve temel bileşen analizi gibi çalışmalarda hesaplama yükünün azaltılması sağlanmıştır.

4.2 Non-Parametrik Kernel Tabanlı Sınıflandırma Algoritmaları

Araştırmalar örnekleme verilerinin olasılık dağılım diyagramlarının keyfi dağılımlar gösterdiğini ortaya koymuştur. Bu nedenle örnekleme verilerinin olasılık dağılımlarının gerçekçi bir biçimde tanımlanabilmesi için farklı yaklaşımlar geliştirilmiştir. Bu amaçla farklı dağılım parametreleri ve istatistik testler (ortalama değer ve standart sapmanın yanında ek parametreler, normalite testi, uyuşum testi vb.) kullanılabilir. Ek parametreler kullanılarak tanımlanan dağılımın normal dağılım varsayımından farklı olup olmadığı istatistiksel olarak test edilebilir.

Parametrik sınıflandırma yöntemlerinin yetersiz kaldığı bir diğer nokta da örnekleme veri seti büyüklükleri sınıflar arasında belirgin bir biçimde değiştiğinde, soncul olasılıkların $(P(C_j | x))$ doğru biçimde tahmin edilememesidir.

Parametrik kestirimde parametreleri örnekleme verisinden çıkarılan olasılık dağılım fonksiyonu bilgisini kullanarak varsayımlar yapılır. Non-parametrik algoritmalarda ise olasılık dağılım fonksiyonu direkt olarak tahmin edilir. Kernel kestirimi şu şekilde verilmektedir;

$$\hat{p}(x | C_j) = \frac{1}{n_j h^d} \sum_{i=1}^{n_j} K\left(\frac{x - x_i}{h}\right). \quad (4.16)$$

h pencere genişliği veya yumuşatma parametresidir. K 'nın şekline bağlı olarak değişik türlerde kestirim algoritmaları oluşturulabilir.

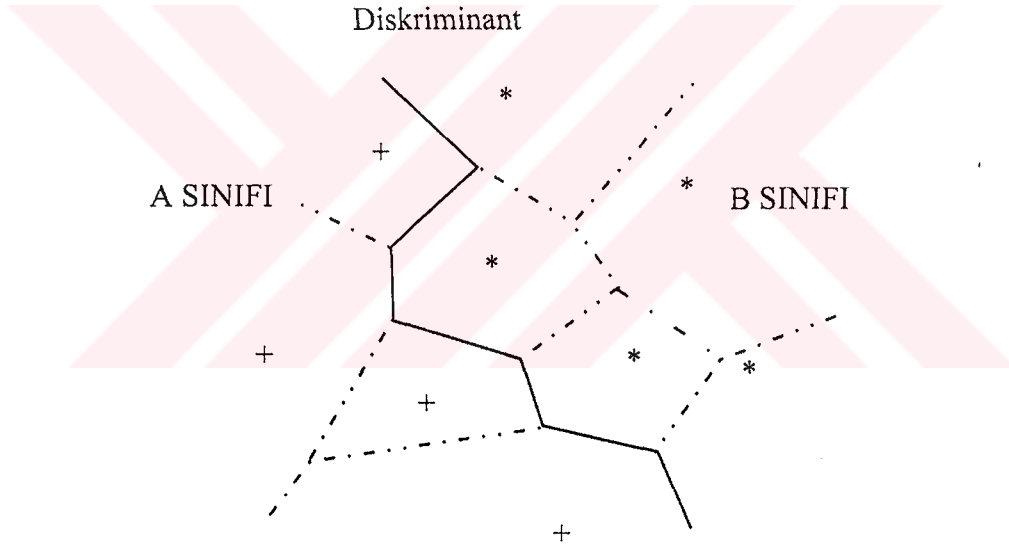
Kernel kestirimlerinin bir dezavantajı tüm örneklerin depolanması gereğidir. Aşırı bilgi taşınması önlemenin bir yolu seçici yöntemlerle örneklerin bir kısmının atılması olabilir. Bir diğer metot ise verilerin kümelenmesi ve örneklerin kendileri yerine onları temsil eden referans vektörlerinin saklanmasıdır.

4.2.1 K-En Yakın Komşu (K-Nearest Neighbor (k-NN))

x örneğine k . en yakın örneği $x^{[k]}$ ile gösterelim ve $V^k(x)$ da, d boyutlu $r^k = \|x - x^{[k]}\|$ çaplı kürenin hacmi olsun. O zaman c_d , d boyutlu birim kürenin hacmi olmak üzere $V^k(x) = r^k \cdot c_d$ 'dir. Örneğin, $c_1 = 2$, $c_2 = \pi$, $c_3 = 4\pi/3$ vb. x örneği civarındaki yoğunluğu tahmin için x örneği içeren tek elemanlı, iki elemanlı, n elemanlı olarak süren $\mathfrak{R}_1, \mathfrak{R}_2, \mathfrak{R}_3, \dots$ bölgeler dizisini ele alalım. $V^k(x)$ veya $V_n(x)$ hacimleri bu bölgelere denk düşer. Bu bölgelerden birisi içine düşen örnek sayısı k_n olmak üzere $p(x)$ için n . kestirim ;

$$p_n(x) = \frac{k_n/k}{V_n} \quad (4.17)$$

şeklinde olur.



Şekil 4.4 1 en yakın komşu metoduyla bulunan keyfi diskriminanta sahip sınıflayıcı

4.2.2 Parzen Pencereleeri (Parzen Windows)

Parzen pencereleri yaklaşımı olasılık dağılımlarını $\mathfrak{R}_n$ bölgelerini d boyutlu hiperküpler şeklinde varsayarak tahmin etmeye çalışmaktır. h_n hiperküpün kenar uzunluğu olmak üzere, bölgelerin hacimleri

$$V_n(x) = h_n^d \quad (4.18)$$

Hiperküpün içine düşen örnek sayısı k_n terimini analitik olarak hesaplayabilmek için şu pencere fonksiyonunu kullanalım;

$$\varphi(u) = \begin{cases} 1 & |u_j| \leq 0.5 \\ 0 & \text{diğer} \end{cases} \quad (4.19)$$

O zaman $\varphi(u)$ merkezi orijin olan birim hiperküpü ifade etsin. Eğer x_i örneği x merkezli hiperküp hacminin içine girerse $\varphi((x - x_i)/h_n)$ şeklinde belirlenen fonksiyon çıkışı 1 aksi halde sıfır olacaktır. Bu durumda bu hiperküpün içindeki örnek sayısı ;

$$k_n = \sum_{i=1}^n \varphi\left(\frac{x - x_i}{h_n}\right) \quad (4.20)$$

eşitliğiyle bulunur ve bu değer (4.17) eşitliğinde yerine konursa;

$$p_n(x) = \frac{1}{n} \sum_{i=1}^n \frac{1}{V_n} \varphi\left(\frac{x - x_i}{h_n}\right) \quad (4.21)$$

kestirim fonksiyonu ortaya çıkar.

Kernel kestirimleri çeşitli yumrusal (bumps) yapıların bütünleştirilmesi şeklindedir. K bu yumrusal yapıların şeklini belirlerken h bu şekillerin genişliklerinin belirler. Tüm boyutlarda genişliğin h olduğu küresel yumru yapılarında Euclidean uzaklıkları kullanılır. Eğer bu eşit varyanslar varsayımı geçerli değilse farklı varyanslar ve kovaryansların tahmin edilmesinde Mahalanobis uzaklığı kullanılır. (Duda vd., 2000).

4.2.3 h terimi seçimi

Kernel tabanlı kestirimlerde dağılım veya örtü parametresi h değerinin seçimi kritiktir. Eğer çok büyükse uzaktaki komşular bile x civarındaki yoğunluğun çok yumuşak geçişli bir fonksiyona sahip olmasına neden olacaktır. Eğer çok küçükse $\hat{p}$ örnek noktalarını merkez alan n tane keskin darbenin süperpozisyonudur.

Parzen pencereleri kullanırsak h çok küçükse ve az sayıda örnek varsa kernel bölgelerine örnek düşmemesine ve sıfır kestirime neden olur bunun aksine k -NN kernel bölgelerine k tane örnek düşmesini garantiler. h değerinin küçük veya büyük olması kestirim başarısını düşürür.

5. YAPAY SİNİR AĞLARI

Tanım olarak yapay sinir ağı (Artificial Neural Networks-ANN); karmaşık transfer problemlerin çözümü için basit işlem birimlerinin (nöronlar) ağırlıklı bağlantılarından oluşmuş ağ yapısıdır. Birimler arasındaki bağlantıların topolojisi ve bağlantıların ağırlıkları ağın gerçekleştireceği işlemi belirler. Yapay sinir ağı algoritmalarını çeşitli bilimsel alanlardaki uygulamalarında hızlı bir artış görülmektedir. Örnek tanıma alanında yapay sinir ağı kullanılarak oluşturulan sınıflayıcıların bir çok avantajı vardır;

- 1- Parametrik olmayan yapısı,
- 2- Standart sınıflandırma yöntemlerine alternatif teşkil eden, verilen bir hata kriteri doğrultusunda yapısını optimize ederek sonraki deneme girişlerine cevap verebilecek şekilde öğrenme yeteneği,
- 3- Sınıfların olasılık modelleriyle ilgili hiçbir varsayım yapmaması,
- 4- Özellik uzayında yüksek dereceden ve nonlinear karar sınırları oluşturma yeteneği,
- 5- Giriş verilerini değişken olarak kullanan nonlinear fonksiyonlar üretebilmeleri,
- 6- Yüksek derecede paralel ve düzenli mimarileriyle donanımsal yapılara tümleştirilebilme uyumlulukları (Alpaydın ve Gürgeç, 1998).

Sınıflandırma işleminde nonlinear fonksiyonların seçimi akıllıca yapılırsa keyfi karar verme bölgeleri en küçük hataya netice verecek şekilde oluşturulabilir. Fakat uygun nonlinear fonksiyonların seçimi temel zorluktur. Alternatif olarak sınıflandırma problemine dair ön bilgiye sahip olabiliriz ve bu da bize nonlinearlik seçiminde yardımcı olabilir. Bununla birlikte bu bilgilerin eksikliğinde bir yönteme dayanarak ve otomatik olarak nonlinearliği bulabilen metotlar henüz gelişmiş değildir. O halde araştırılması gereken nonlinearliği öğrenmek ve farklı veriler için ön bilgi olarak kullanmaktır. Bu, çok katmanlı yapay sinir ağlarının veya daha özel olarak çok katmanlı algılayıcıların kullandığı yaklaşımdır. Nonlinear haritalamayı yönlendiren parametreler lineer diskriminantları yönlendiren parametrelerle aynı anda öğrenme işlemini gerçekleştirirler (Duda vd.,2000).

Deneme verisinden oldukça basit fakat etkili oluşturulmuş algoritmalarla nonlinearliğin yapısını öğrenebilir. Bu modeller çok etkilidir, teorik özelliklere sahiptir ve pratikte sayısız uygulamaya tatbik edilebilir.

Çok katmanlı ağırları eğitmede kullanılan en yaygın metotlardan biri hatanın gradient descent'ini bulmaya dayanan geriye yayılım (Backpropagation) algoritması olarak bilinen LMS (Least Mean Square) algoritmasının genişletilmiş halidir. Geriye yayılım algoritması yapay sinir ağlarının en temel öğrenme algoritması olduğu için bu çalışmada derinlemesine incelenecektir. Bu modellerin dizaynının basitliği nedeniyle, kullanıcılar çok değişik modelleri hızlı bir şekilde test edebilirler. Kavramsal ve algoritmik basitliğiyle geriye yayılım algoritması, gerçek dünyadaki bir çok probleme cevap verdiği gibi örnek tanıma sahasında da en büyük destekleyici metottur.

Geriye yayılım algoritmasının temel mantığı basit olmasına karşın bir takım parametrelerle ağı performansı ve hızını artırmak mümkündür. Ağların ve fonksiyonlarının analizi yapılarak elde edilen bilgiler ışığında giriş değerlerinin ölçeklendirilmesi, başlangıç ağırlıklarının belirlenmesi, istenen hedef çıkışlara sayısal değerler atama gibi seçimler uygun bir şekilde yapılabilir.

Ağ mimarisi ve topolojisi yapay sinir ağlarıyla yapılan sınıflandırmada önemli rol oynar ve optimal topoloji eldeki problemin yapısına birebir bağlıdır. Burada yapay sinir ağlarını bir başka üstün özelliği ortaya çıkmaktadır; genellikle problem domenine ait düzenlenmemiş, işlenmemiş bilgi ağ mimarisine saklı katman sayısı, birim sayısı ve geri besleme bağlantıları gibi seçimlerle dahil edilebilir. Ağ topolojisini kurarak ve geriye yayılım parametrelerin değer tahmini yaparak model oluşturmadaki pratik kolaylık kullanıcının çok kolay bir şekilde alternatif modeller denemesini sağlar.

Yapay sinir ağları teknikleri kullanımında en derin problem ağı karmaşıklığını ayarlama olarak ifade edilebilecek regularizasyon problemidir. Girişlerin ve çıkışların sayısı özellik uzayı ve sınıf sayısından yola çıkılarak belirlenebilirken ağırlıklı bağlantıların ve diğer parametrelerin sayısı direkt olarak bulunamaz. Eğer çok fazla bağımsız parametre kullanılıyorsa genelleştirme çok zayıf olacaktır. Tersine eğer çok az parametre varsa eğitim verisi yeterince öğrenilemeyecektir.

Herhangi bir sinir ağı uygulaması şu genel aşamalardan oluşur;

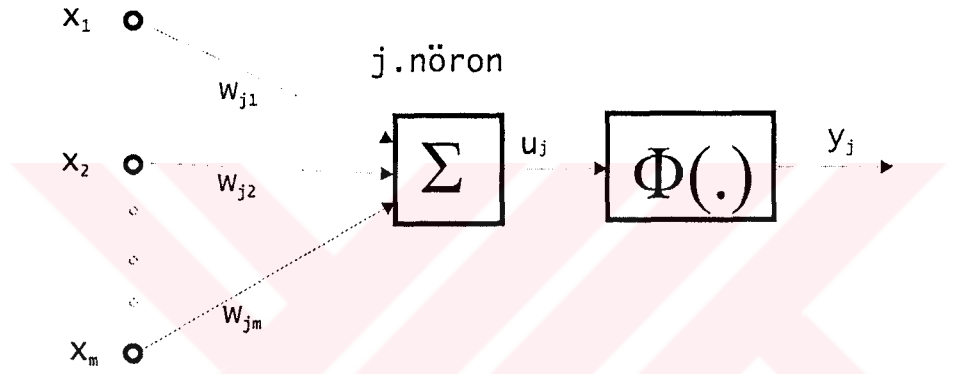
- Verilerin hazırlanması,
- Ağ mimarisinin tasarlanması,
- Probleme yönelik parametrelerin tahmini (ağın çalışması),

- Ağ performansının değerlendirilmesi.

Yapay sinir ağlarının kullanıcıyı veri ve problem domeni bilgileriyle uğraşmaktan tamamen muaf tuttuğunu düşünmenin yanlışlığını belirtmek gerekir. Ağlar, sınıflayıcılar oluşturmak için çok güçlü ve hızlı yapılardır ve kullanıcı da analizler ve tekrarlanan çok geniş dağılımlı problem türleri üzerine yapılan deneylerle uzmanlık kazanır, sezgilerini kuvvetlendirir.

5.1 Temel Nöron Modeli

Yapay sinir ağlarında ‘nöron’ olarak adlandırılan temel işlem birimi biyolojik nöronlarla aynı işlevi yapabilecek bağlantılara sahiptir. Şekil 5.1 en basit nöron modelini şematize etmektedir.



Şekil 5.1 Temel nöron modeli

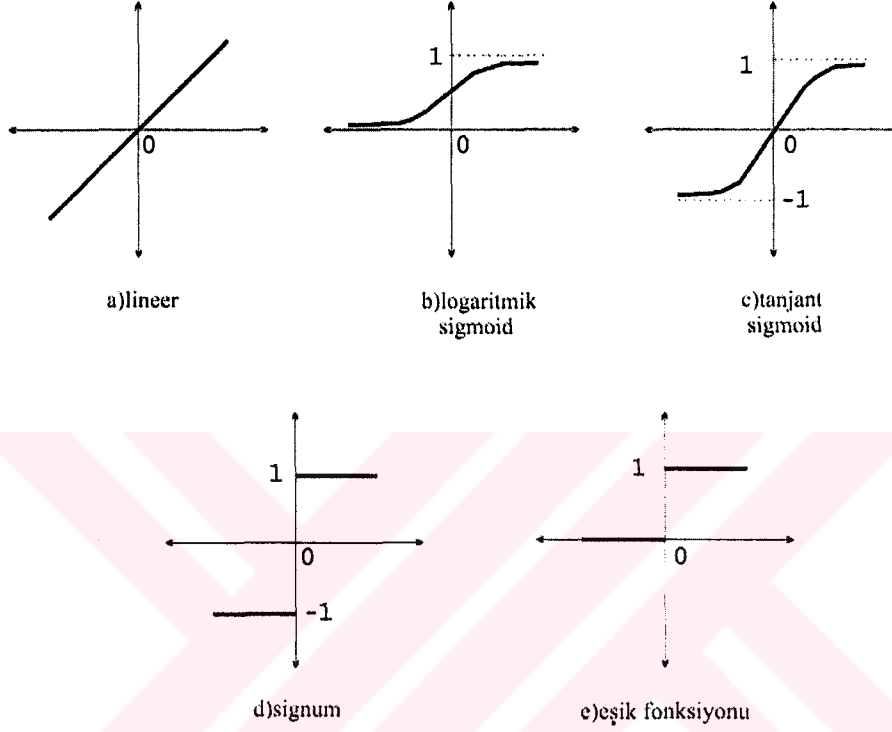
x vektörü nöronun girişleridir. Bütün girişler eş zamanlı olarak ağa sunulur ve belirli bir ağırlıklı etki oranında çıkışı belirlerler. Hücre bağlantı ağırlıkları hücrenin w_j vektörüyle giriş x vektörünün çarpımının hesaplar.

$$u_j = \sum_{i=1}^m x_i w_{ji} + b_j \equiv w_j^T \cdot x + b_j \quad (5.1)$$

Hesaplanan bu değer u_j hücre aktivasyonudur. Biyolojik nöron ile bağlantı kurulurak ağırlıkların işlem birimlerine bağlantı noktalarına sinaps, ağırlıkların kendilerine de sinaptik ağırlık adı verilebilir. Ağırlıklı toplam sunucunda oluşan aktivasyon değeri $\Phi(.)$ ile gösterilen fonksiyona giriş değeri olarak girer ve j . nöronun veya işlem biriminin çıkışı belirlenir;

$$y_j = \Phi(u_j). \quad (5.2)$$

$\Phi(.)$ fonksiyonunu, aktivasyon fonksiyonu veya yine biyolojik nörondaki giriş çıkış nonlineerliğiyle bağlantı kurularak işlem biriminin nonlineerliği olarak da ifade edilebilir. İşlem birimi çıktısı, biyolojik nöronlarda aksonlar boyunca iletilen cevap sinyalinin temsil eder. Yapay sinir ağlarında aktivasyon fonksiyonları seçimi girişlerin ve sınıfların sayısal değerleriyle birebir ilişkili olmalıdır. Değişik formlarda aktivasyon formları seçilebilir.



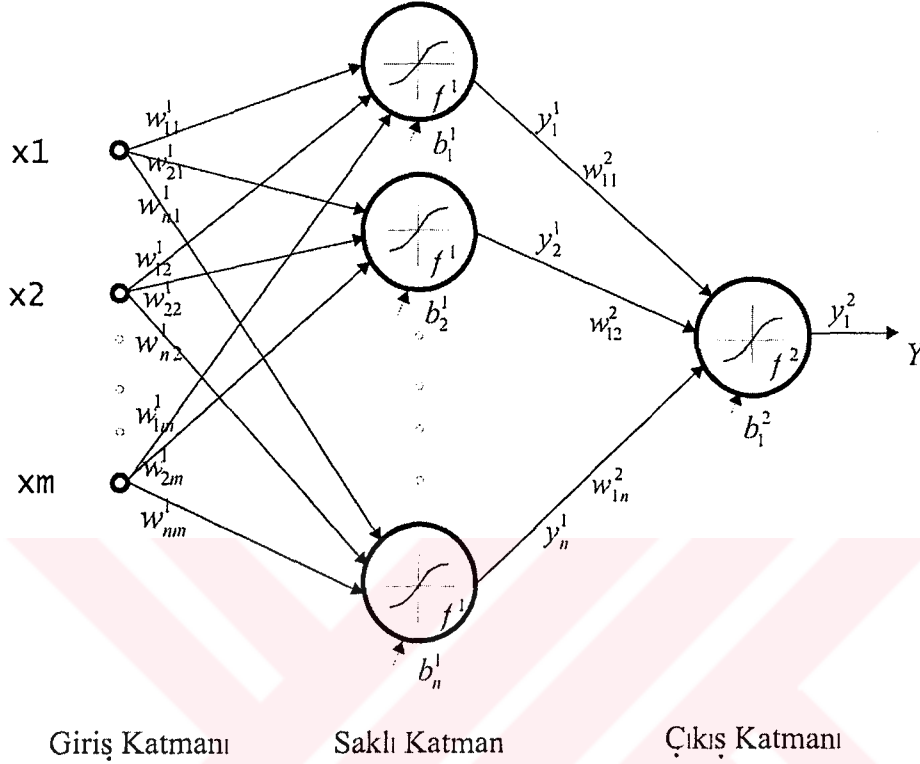
Şekil 5.2 En yaygın aktivasyon fonksiyonları

5.2 İleri Besleme İşlemi ve Sınıflandırma

Günümüzde de en çok kullanılan sinir ağı sınıflandırma yöntemi çok katmanlı algılayıcı sinir ağıdır (Multilayer Perceptron Network, MLP). Şekil 5.1 bir giriş katmanı, bir saklı katman ve bir çıkış katmanı ve bu katmanlar arasında ayarlanabilir ağırlıklardan oluşan basit 3 katmanlı bir ağ yapısını temsil ediyor. Ayrıca her birimde girişlere ek olarak bir de basit bias değeri toplanıyor. İşlem birimleri olan yapay hücrelerin karakteristiği biyolojik hücrelerin özellikleri üzerine çalışmalar yapılarak oluşturulmuştur.

Örnek tanıma probleminde her bir özellik vektörünün terimleri $x_1, x_2, \dots, x_m$ girişleriyle ve sınıfların ayrılmasında yararlanılacak diskriminant fonksiyonları çıkış katmanına gelen $y_1^1, y_2^1, \dots, y_N^1$ girişleriyle ve spektral sınıfları temsil eden çıkış Y ile temsil edilmektedir. Üst

indisler katman numarasını ifade etmektedir. Gizli tabakalar giriş verilerinde saklı olan işlenmemiş bilgileri hesaplayarak bağlantı ağırlıklarını olarak yansıtır. Bu ağırlıklarla da özellik uzayında karar sınırları belirlenir.

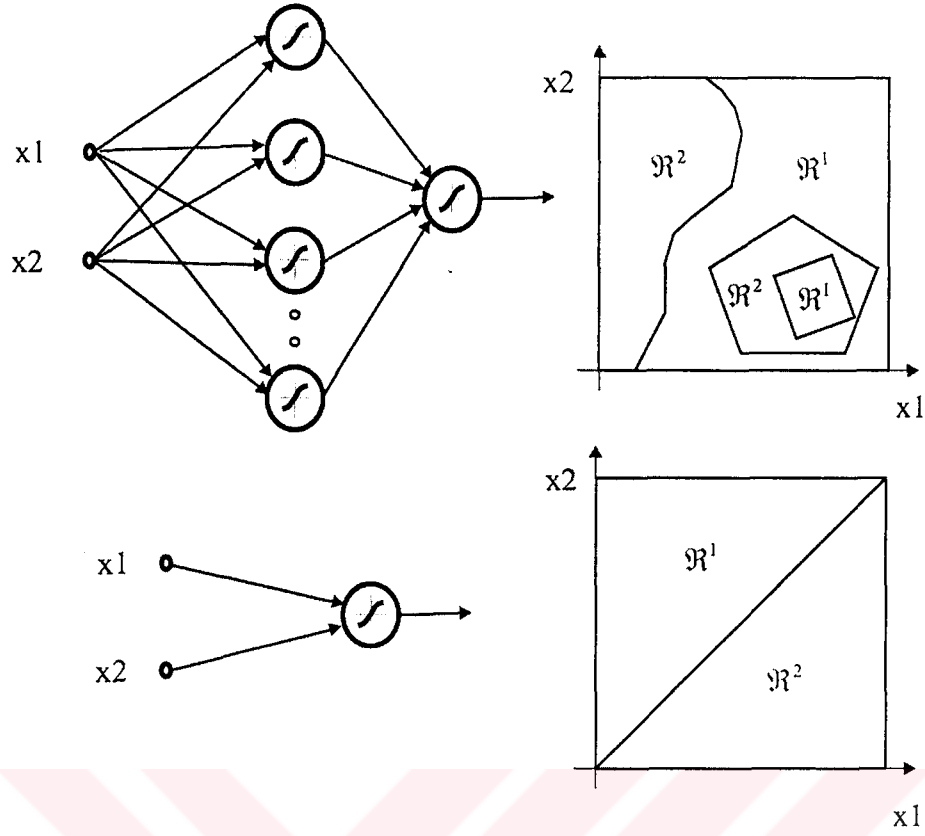


Şekil 5.3 Çok katmanlı yapay sinir ağı yapısı

Yukarıdaki modelde belirtilen yapay sinir ağı, saklı katmandan yoksun diğer bütün ağlara göre daha fazla etkin hesaplama yeteneğine ve güce sahiptir. Bu tür bir ağla tek katmanlı ağların yapamadığı XOR problemini çözebilmek mümkündür. Yukarıda çıkış katmanında tek nöronu ve yalnız bir saklı katmanı olan modelde W matris formunda y, x ve b ise vektörel formda olmak üzere genelleştirilmiş ileri besleme hesaplaması şu şekildedir;

$$y = f^2(W^2 f^1(W^1 x + b^1) + b^2) \quad (5.3)$$

Saklı katmanlar ve çıkış katmanındaki aktivasyon fonksiyonları genellikle birbirinden farklı seçilir. Matematiksel analiz açısından her katmandaki tüm birimlerin aktivasyon fonksiyonlarını aynı seçmek en uygun olanıdır.



Şekil 5.4 Tek katmanlı ağlar ile çok katmanlı ağların karşılaştırılması

Şekil 5.3'te tek katmanlı ağlar ve oluşturabileceği lineer karar sınırları ile çok katmanlı ağlar ve uygun sayıda verilen saklı katman nöronları sayesinde keyfi karar sınırları gösterilmiştir.

5.3 Geriye Yayılım (Backpropagation) Algoritması

Çok katmanlı ağlarla manual olarak giriş ve çıkış arasında kurulan çok sayıda diskriminant fonksiyonlarının bileşimi ile keyfi dağılım gösteren örnek uzayının sınıflandırılmasının yapılabileceği yukarıda ifade edildi. Fakat asıl önemli nokta, ağın başlangıç ağırlıkları veya diğer ağ parametrelerini eğitim verisi ve istenen çıkışlara bağlı olarak ayarlayabilmesi neticesinde kendi kendine bu bağıntıyı kurabilmesidir. (Duda vd., 2000)

Geriye yayılım algoritması en basit ve en genel çok katmanlı ağların eğitici öğrenme metodudur, LMS algoritmasının genişletilmiş halidir. Daha hızlı veya diğer bazı istenen özelliklere sahip başka metotlar olabilir fakat ağın öğrenmesi olarak ifade edebileceğimiz işlevi daha iyi bir şekilde gerçekleştiren metot sayısı çok azdır. LMS algoritması iki katmanlı sistemler için kullanılır, istenen çıkış değerleriyle ağ çıkışları arasındaki farkın karesiyle orantılı olan hata değeri her bir çıkış birimi için değerlendirilebilir. Fakat benzer şekilde çok

katmanlı ağlarda sadece ağ çıkışı veya dolayısıyla hesaplanan hatayla, saklı katman ile çıkış katmanı arasındaki ağırlıkların arasındaki ilişkiyi değerlendirebilmek mümkündür.

Fakat giriş vektörlerinin nonlinear dönüşümünden sorumlu girişlerden saklı katmana veya birden fazla saklı katman varsa saklı katmandan saklı katmana olan ağırlıkların öğrenmesi nasıl gerçekleşecek? Saklı katmanlar için giriş ve istenen çıkış arasındaki hata bağıntısı kurulabilseydi -ki kurulamaz, saklı katman çıkışlarını istenen çıkışlara yaklaştırmak gibi bir ihtimal söz konusu değildir- girişlerden saklı katmana veya birden fazla saklı katman varsa saklı katmandan saklı katmana olan ağırlıklar da ayarlanabilirdi. Geriye yayılım algoritması her saklı katman birimi için etkili bir hata hesaplama metodu sunar ve girişten saklı katmana veya saklı katmandan saklı katmana olan ağırlıkların eğitilebilmesini sağlar.

5.3.1 Ağıın Eğitilmesi

Ağın eğitilmesi giriş katmanına eğitim örneklerinin sunulması, ağırlıklı hesaplamalar yapılarak çıkış katmanında nihai sonucun belirlenmesinden sonra başlar. Çıkış katmanında ağ çıkışları istenen hedef değerlerle karşılaştırılır aradaki fark hata işaretidir. Hata fonksiyonu hesaplanır ve ağ çıkışı istenen değerlere yakınsayınca kadar bu hata fonksiyonu minimize edilmeye çalışılır. Bunun için ağ ağırlıkları hata ölçütünü küçültmek için ayarlanır.

Bir örnek verisi sunulduğunda eğitime hatası ağın çıkış değeriyle istenen hedef değer arasındaki farkın karesi olarak verilir. y_k çıkış katmanındaki k . nöron çıkışı, d_k istenen hedef değeri, c saklı katmandaki nöron sayısını ifade etmek üzere E^p ağın p örneğiyle eğitime hatası;

$$J(w) = E^p = \frac{1}{2} \sum_{k=1}^c (d_k - y_k)^2 = \frac{1}{2} \|d - z\|^2 \quad (5.4)$$

bağıntısıyla bulunur. Geriye yayılım öğrenme kuralı ağırlığın bir fonksiyonu olan hata fonksiyonunun gradient descent'ini bulmaya dayanır. Başlangıç ağırlıkları rasgele değerlerle başlatılır ve hata fonksiyonunu azaltacak yönde ;

$$\Delta w = -\eta (\partial J / \partial w) \quad (5.5)$$

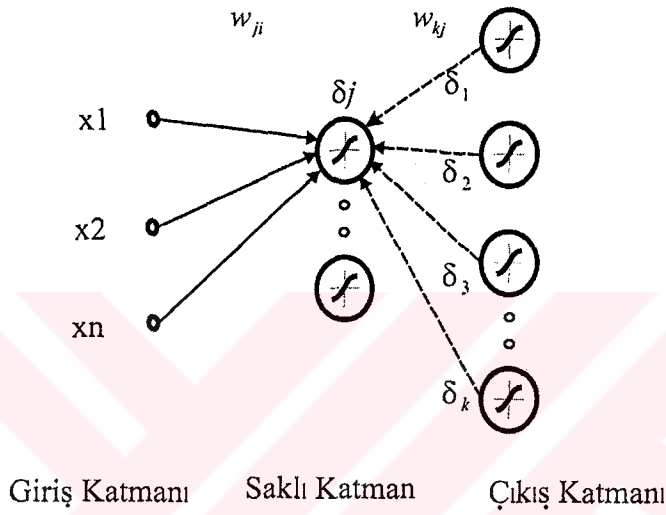
veya her bir ağırlık ele alınacak olursa ;

$$\Delta w_{ji} = -\eta(\partial J / \partial w_{ji}) \quad (5.6)$$

olarak verilir. η öğrenme oranı, ağırlık değişiminin duyarlılığını belirler. Ağırlık güncellemesi;

$$w(m+1) = w(m) + \Delta w(m) \quad (5.7)$$

eşitliğiyle sağlanır.



Şekil 5.5 Hatanın geriye yayılımı

Üç katmanlı bir ağ ele alalım. w_{ji} vektörü giriş ile saklı katman arasındaki bağlantıları, w_{kj} vektörü saklı katman ile çıkış katmanı arasındaki bağlantıları, u_k çıkış katmanındaki k. birimin aktivasyon değerini ifade etsin. Zincir kuralı uygulanırsa;

$$\frac{\partial J}{\partial w_{kj}} = \frac{\partial J}{\partial u_k} \cdot \frac{\partial u_k}{\partial w_{kj}} = -\delta_k \frac{\partial u_k}{\partial w_{kj}} \quad (5.8)$$

sonucuna ulaşırız.

$$\delta_k = -\frac{\partial J}{\partial u_k}, \quad (5.9)$$

olmak üzere aktivasyon fonksiyonunun türevi alınabilen bir fonksiyon olduğunu varsayarsak;

$$\delta_k = -\frac{\partial J}{\partial u_k} = \frac{\partial J}{\partial y_k} \frac{\partial y_k}{\partial u_k} = (d_k - y_k) \cdot f'(u_k) \quad (5.10)$$

$\frac{\partial u_k}{\partial w_{kj}} = y_j$ eşitliği de göz önüne alınırsa saklı katmandan çıkış katmanına ağırlık güncelleme

veya öğrenme kuralı;

$$\Delta w_{kj} = \eta \delta_k y_j = \eta (d_k - y_k) f'(u_k) y_j. \quad (5.11)$$

olarak belirlenir. Giriş saklı katman arasındaki ağırlıkların, w_{ji} güncellenmesi saklı katman çıkış arasındaki ağırlıkların, w_{kj} güncellenmesinden daha karmaşık bir işlemdir. Çıkıştaki toplam hataya w_{ji} ağırlıklarının etkisini aşağıdaki bağıntı ile bulabiliriz;

$$\frac{\partial J}{\partial w_{ji}} = \frac{\partial J}{\partial y_j} \frac{\partial y_j}{\partial u_j} \frac{\partial u_j}{\partial w_{ji}} \quad (5.12)$$

Sağ taraftaki ilk terim için;

$$\begin{aligned} \frac{\partial J}{\partial y_j} &= \frac{\partial}{\partial y_j} \left[\frac{1}{2} \sum_{k=1}^c (d_k - y_k)^2 \right] \\ &= -\sum_{k=1}^c (d_k - y_k) \frac{\partial y_k}{\partial y_j} \\ &= -\sum_{k=1}^c (d_k - y_k) \frac{\partial y_k}{\partial u_k} \frac{\partial u_k}{\partial y_j} \\ &= -\sum_{k=1}^c (d_k - y_k) f'(u_k) w_{kj} \end{aligned} \quad (5.13)$$

eşitliği kullanılarak;

$$\delta_j = f'(u_j) \sum_{k=1}^c w_{kj} \delta_k \quad (5.14)$$

elde edilir. Buradan ağırlık güncellemeleri;

$$\Delta w_{ji} = \eta x_i \delta_j = \eta \left[\sum_{k=1}^c w_{kj} \delta_k \right] f'(u_k) x_i \quad (5.15)$$

bağıntısı ile yapılır. (5.11) ve (5.15) bağıntıları geriye yayılım algoritması veya hatanın geriye yayılması olarak bilinen eğitime protokolüdür. Eğitim sırasında giriş katmanından saklı katmana olan ağırlıkların öğrenmesini sağlamak için, çıkış katmanından saklı katmana doğru hatanın yayılımı gerçekleştirilir. Temel olarak geriye yayılım, sürekli fonksiyonlarla zincir kuralı oluşturmak suretiyle ağırlıklara göre türevin hesaplanması ile gradient descent işleminin katmansal olarak yapılmasıdır.

Tüm gradient descent temelli protokollerde olduğu gibi geriye yayılım algoritmasının gerçek davranışı başlangıç noktasına bağlıdır. (5.15) bağıntısından da anlaşılacağı gibi w_{kj} ağırlıklarının sıfır değerinden başlatılması durumunda geriye yayılan hata da sıfır olacağı için girişten saklı katmana olan ağırlıklar hiçbir zaman değişmeyecektir. Bu yüzden eğitim süreci rasgele ağırlık değerleriyle başlatılır.

Yapılan analiz basit üç katmanlı ağırlar göz önünde bulundurularak yapılmıştır. İleri beslemeli olması koşuluyla;

- Üçten fazla katmanının olması,
- Farklı katmanlarda farklı nonlineerliğin olması,
- Veya her işlem biriminin kendine özel nonlineerliği olması,
- Her işlem biriminin farklı öğrenme oranına sahip olması,

durumlarında (5.11) ve (5.15) bağıntıları genelleştirilebilir. Eğitim algoritmalarının katmanlar arası veya sonraki katmandan önceki katmana olan ağırlıklar içeren ağlarda uygulanması büyük bir problemdir.

Geriye yayılım algoritmasının etkinliği;

- Aktivasyon fonksiyonları
- Giriş vektörünün ölçeği,
- Hedeflenen çıkış büyüklüklerinin sayısal değerleri,
- Saklı katman sayısı,
- Ağırlıklara ilk değerler atanması,
- Öğrenme oranları,
- Hatanın minimize edilmesinde kararlılığı sağlayacak yönde etki eden momentum katsayısı ,
- Hata kriteri fonksiyonu,
- Aşırı öğrenmeyi önleme için eğitimin durdurulma noktası,

parametreleri uygun duruma getirilerek artırılabilir (Duda vd.,2000).

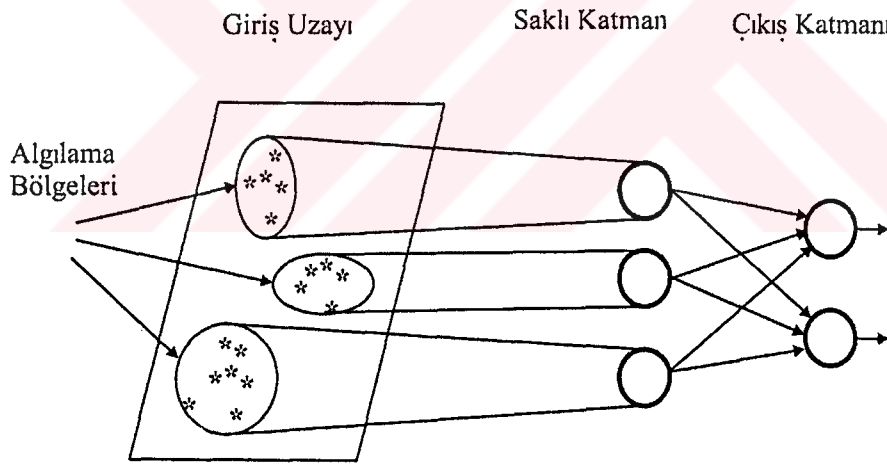
5.4 Diğer Bazı Ağ Mimarileri Ve Öğrenme Algoritmaları

Bu bölümde tez çalışmasında da kullanılan bazı temel yapay sinir ağı yapıları anlatılmaktadır. Diğer ağ ve öğrenme kuralları Hines'de (1997) bulunabilir.

5.4.1 Radyal Fonksiyon Temelli Ağlar

Radyal fonksiyon temelli ağlar (Radial Basis Function Network, RBF), MLP ağlarına benzer şekilde fonksiyonel haritalama işlemini yapar. Fakat yapısı ve birimlerinin işleyişi farklıdır.

RBF ağları giriş, saklı katman ve çıkış katmanı olmak üzere üç katmandan oluşur. Saklı katman ve çıkış katmanlarındaki nöronlar farklıdır. Saklı katmanda bulunan radyal temelli fonksiyonlar kullanan nöronlar lokal haritalama işlemini gerçekleştirir. Bu fonksiyonlar özellik uzayında algılama bölgelerinin merkezine yerleştirilmiştir. Her özellik uzayı algılama bölgesi, bir radyal temelli fonksiyon içeren saklı katman nöronunu harekete geçirir.



Şekil 5.6 Radyal Fonksiyon Temelli Ağlar

x giriş vektörü, μ_j algılama bölgeleri olarak ifade edilen bir bölgenin merkezi, σ_j algılama bölgesinin genişliği olmak üzere;

$$g_j(x) = \exp\left[-\frac{(x - \mu_j)^2}{\sigma_j^2}\right] \quad (5.16)$$

eşitliği $g_j(x)$ j . saklı katman nöronun çıkışını verir.

Çıkış katmanı standart lineer nöronlardan oluşmuştur ve saklı katman nöronlarının lineer dönüşümünü sağlar. Eğer bir giriş vektörünün algılama alanlarından birinin merkezine yakın olması durumunda saklı katman nöronu uyarılacaktır. Eğer bir giriş vektörünün iki adet algılama alanının merkezinin arasında ve ikisinin bölge genişliği σ , içinde bulunursa bu bölgelere denk düşen saklı katman nöronlarının ikisi de kısmen uyarılacaktır. Eğer bir giriş vektörünün, algılama alanlarının uzağında bulunması durumunda hiçbir saklı katman nöronu uyarılmayacak ve RBF çıkışı, çıkış katmanındaki nöronların bias değerlerine eşit olacaktır. RBF’de lokal haritalama söz konusudur yani yalnızca algılama bölgelerinin yakınlarında bulunan giriş değerleri saklı katmanların uyarılmasını sağlayabilir, MLP’de ise tersine global haritalama söz konusudur yani tüm girişler çıkış değeri oluşturur.

Etkin RBF ağları oluşturmada radyal temelli fonksiyonlar ve bunların genişlik parametrelerinin seçimi önemlidir. Genişlik öyle seçilmelidir ki tüm algılama bölgeleri kapsanmalı fakat bir fonksiyonla tüm giriş uzayını kapsanmamalıdır. Bu da tek bir giriş vektörü için tüm saklı katman nöronlarının uyarılmaması anlamına gelir.

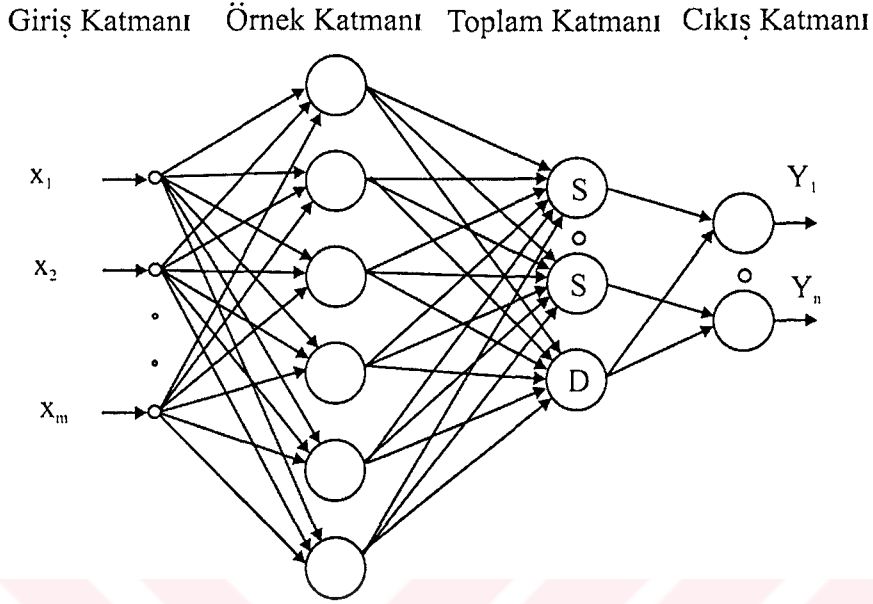
5.4.2 Genelleştirilmiş Regresyon Ağları

Genelleştirilmiş Regresyon Ağları (Generalized Regression Neural Networks, GRNN) özellikle sistem modelleme ve kestirimi gibi fonksiyon yaklaşırma uygulamalarına en uygun ileri beslemeli sinir ağlarıdır.

GRNN (Generalized Regression Neural Network) dört katmandan oluşur. İlk katman giriş katmanıdır ve ikinci katmana tam olarak bağlanmıştır. İkinci katmanda her bir giriş örneğini temsil eden bir nöron vardır. Bu katman RBF ağlarındaki ilk katmanın işlevine benzer bir işlev gerçekleştirir; girişlerle depolanmış örnekler arasındaki uzaklık ölçütünü hesaplayarak çıkış üretirler. Üçüncü katman toplama katmanı olarak isimlendirilir, S-toplama ve D-toplama nöronlarından oluşur. S-toplama nöronları ikinci katmanın çıkışlarının ağırlıklı toplamını gerçekleştirir. D-toplama nöronları ise bu çıkış değerlerini ağırlıksız olarak toplar. Dördüncü katmandaki çıkış nöronlarının her birine bağlı sadece bir S-toplama nöronu ve hepsine ortak olarak bağlantılı bir adet D-toplama nöronu vardır.

2. katmanda σ genişlik katsayısına sahip gauss fonksiyonları kullanılır. 3.katmanda ise toplama katmanında ağırlıklı toplama olarak ifade edilebilecek S-toplamaları ve ağırlıksız

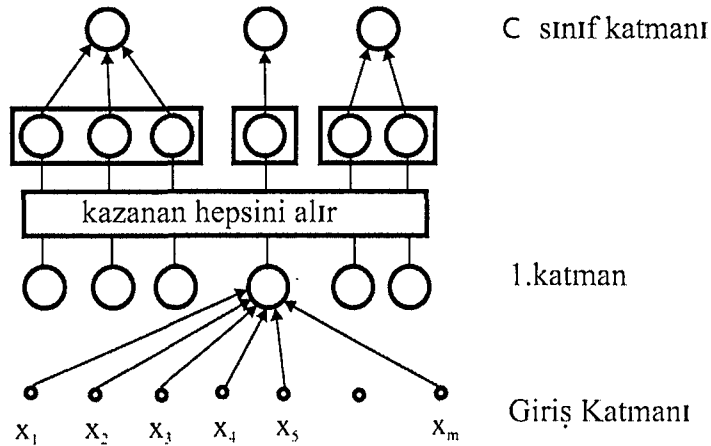
toplama olarak ifade edilebilecek D-toplamı gerçekleştirilir. Çıkış katmanında ise her bir S-toplamı D-toplamına bölünerek çıkışlar elde edilir.



Şekil 5.7 Genelleştirilmiş Regresyon Ağ Yapısı

5.4.3 Kazanma Algoritmali Öğrenme Ağı

Kazanma Algoritmali Öğrenme Ağı (Gaining Algorithm Learning, GAL), öz nitelik uzayındaki sınıf sınırlarını en yakın mesafe ölçütüne göre belirler. Giriş vektörü ile ağdaki tüm düğümlere olan mesafeler hesaplanır. Giriş vektörünün sınıfı, ağdaki düğümlere en yakın mesafede olan düğümün sınıfı olarak belirlenir. (Ölmez, 2002). Ağın düğüm sayısı, eğitim sırasında ihtiyaca göre otomatik olarak bulunur. Şekil 5.8’de GAL ağını yapısı gösterilmiştir



Şekil 5.8: GAL ağının yapısı

İlk katman, düğüm ağırlıkları ile giriş vektörü arasında minimum mesafeyi bulmada kullanılırken; ikinci katman, ağdaki düğümlerin ait oldukları sınıfı tanımlamak için kullanılır. İkinci katmandaki ağırlıklar başlangıçta 0 değerini alır, eğitim sırasında bu bağlantılar 1'lenir. İkinci katman, aynı sınıftan çıkışları lojik olarak OR'lamak için kullanılır.

GAL ağının en önemli özelliği, düğüm sayısının eğitim sırasında problemin yapısına bağlı olarak kendiliğinden belirlenebilmesidir. GAL ağının yapısı, başlangıçta ağa verilen giriş vektörlerinin sırasına çok bağlıdır. Ağ içinde daha önceden anlamlı olan, ancak ağa yeni düğümler ilavesi ile anlamını yitiren ve bir daha kullanılmayan düğümler oluşmaktadır. Bu düğümler unutma algoritması tarafından ağdan çıkarılır. Unutma algoritmasının amacı, çıkarıldığı zaman ağın performansını değiştirmeyen düğümleri bulup, bu düğümleri ağdan atmaktır.



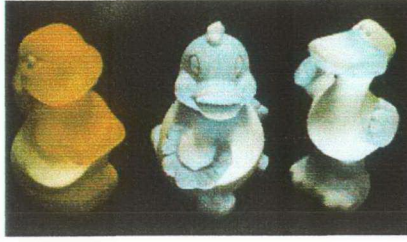
6. YAPAY SİNİR AĞLARIYLA 3-D BİLGİSAYAR GÖRMESİ UYGULAMASI

Bu tezdeki görüntü sınıflandırması uygulaması için genel bilgisayar görmesi sistemi akış diyagramında verilen temel adımların atılması gerektiği Bölüm 1’de belirtilmiştir. Tabiatta analog işaretler veya sürekli uzay işaretleri grubuna dahil olan görüntü işaretlerinin sayısal ortama taşınması ilk adımdır. İkinci adım olarak görüntüyü verimli kullanmak adına çeşitli iyileştirme ve belirginleştirme algoritmaları kullanılarak görüntülere dair asıl ilgilenilen kısımlar elde edilmiştir. Uygun bir şekilde her görüntü için özellik çıkarımı yapılmış ve daha sonra oluşturulan özellik vektörleriyle tanımlı örnekler kullanılarak tezin asıl amacı olan 3-D bilgisayar görmesi yapay sinir ağları ile sağlanmıştır.

6.1 Görüntülerin Elde Edilmesi

Normalde analog özellik gösteren görüntü işaretleri sayısal ortama taşınırken ayrı işaretlere dönüştürülmektedir. Işığı yansıtma yada geçirme kabiliyeti veya multispektral yanıt verebilmesi gibi fiziksel özelliklerinden yararlanarak cisimleri görüntülemek mümkündür. Bu çalışmada ilgilenilen ayrı görüntü işaretleri veya dijital görüntü işaretleri, cisimlerin ışığı yansıtma özelliği kullanılarak hem uzaysal koordinatlarda hem de aydınlık düzeyi olarak ayrıştırılmış bir $f(x,y)$ görüntüsüdür. Görüntüyü işleyebilmek için öncelikli olarak bu dijital görüntü işaretini elde etmek gereklidir ki bu da ancak çeşitli algılayıcılar (transducers) aracılığıyla yapılabilir. Algılayıcı olarak bu çalışmada 2 mega piksel çözünürlüğe sahip dijital kamera kullanılmıştır. Algılama işlemi tam olarak sürekli tondaki görüntüler kullanılarak piksel olarak belirtilen dijital görüntü elemanlarının kuantalanmış aydınlık seviyelerinin belirlenmesidir. Bu piksel değerleri kameranın içerisinde görüntüyü oluşturmak üzere belleğe yazılmaktadır. Kamera USB yardımıyla bilgisayara bağlanabilmekte ve elde edilen görüntüleri aktarabilmektedir. Eğer bir bellek bölgesi görüntü verisi içeriyorsa frame buffer olarak adlandırılır. Bilgisayar buradaki pikselleri işlem den geçirdikten sonra dijital/analog dönüştürücüler vasıtasıyla monitöre gerekli sinyalleri aktarabilmektedir.

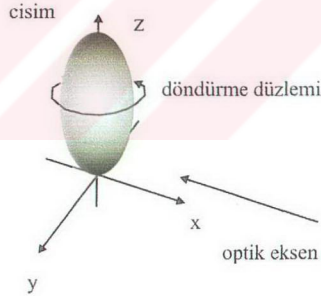
Uygulamada sınıflandırmak üzere kullanılan üç adet üç boyutlu cisim Şekil 6.1’de gösterilmiştir. RGB (red-green-blue) normunda olan bu görüntünün cisimlerin kameradan alınan işlenmemiş ilk görüntüleri olduğunu ve bu çalışmada cisimlere ait renklerden daha çok, alınan bu ve benzeri RGB görüntülerin gri renk seviyelerinde kuantalanmış görüntülere çevrilmektedir. Tüm algoritmalar ve özellik çıkarımı gri seviyeli görüntüler üzerine kurulmuştur.



Şekil 6.1 Kullanılan 3-D cisimler

6.2 Üç Boyutlu Cisimlerden İki Boyutlu Görüntülerle İfade Edilmesi

Üç boyutlu cisimler bilgisayara aktarılırken veya bilgisayar oyunları gibi bilgisayarda üç boyutlu hacimsel cisim simülasyonları yapılırken iki boyutlu görüntülerin sentezinden faydalanılır. Bu çalışmadaki üç boyutlu cisim tanınması uygulamasında da benzer bir yaklaşımla cisimlere ait görüntülerin tanınması sağlanarak amaca ulaşımaya çalışılmaktadır. Cisme bağlı bir xyz sistemi göz önüne alınsın. x yatayda kameranın optik eksenine paralel, y yatayda optik eksene dik, z 'de dikey yukarıya doğru olsun (Şekil 6.2).



Şekil 6.2 Uygulamada kullanılan görüntüleme sistemi

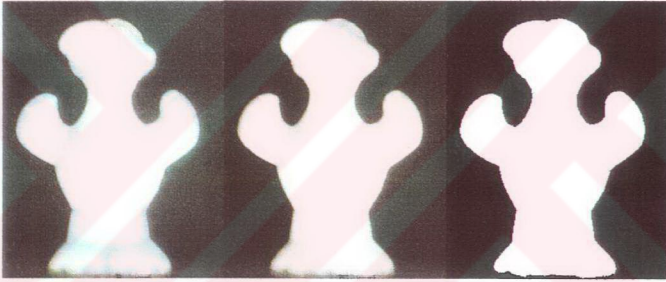
Normalde cismin döndürülebileceği üç farklı açı vardır. Fakat bu çalışmada cisimlerin xy düzleminde z eksenine etrafında dönmesi sağlanmakta ve görüntü olarak da yz düzlemine izdüşümleri kaydedilmektedir. z eksenine etrafında 0° 'den 360° 'ye 5° aralıklarla rotasyonları neticesinde her cisim için 72 adet, toplamda 216 görüntü elde edilmektedir. Optik eksen uzaklığı ve cismi döndürmekte kullanılan xy düzlemi üzerine yerleştirilen platformun merkezi

sabit tutulmaya çalışılmakla beraber bu parametrelerin sabitlenememesinden dolayı öteleme sorunu da ortaya çıkmaktadır. Aydınlatma koşulları da yeterli olmamakla beraber tüm görüntüler aynı anda alındığı için tüm görüntülerde benzer dağılımda arka fon gürültüsü oluşmakta ve görüntü yok edilmesine dair aynı işlemler yürütülmektedir.

6.3 Görüntülerin İşlenmesi

Uygulamada kullanılan görüntü işleme adımları ve bu adımlar sonucunda elde edilen görüntüler (Şekil 6.3)'te verilmiştir.

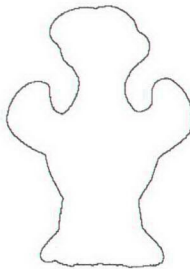
- a. Cisimlerin alınan görüntülerindeki tüm pikselleri kullanmaya gerek olmadığından görüntülerin 300x340 boyutlarında cismi içeren kısmı alınarak görüntü algoritmalarına geçilmektedir.



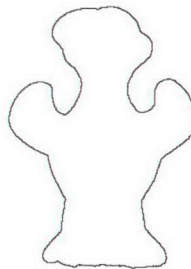
a)

b)

c)



e)



f)

Şekil 6.3 Görüntü işleme adımları sırasında elde edilen görüntüler

- b. Uygulamada cisimlerin özellik vektörleri oluşturulurken cisimlerin RGB renk tonlarıyla renklerle ilgilenmeye ihtiyaç olmadığı için kamerayla alınan RGB normundaki görüntüler gri renk seviyelerinde kuantalanmış görüntülere çevirmektedir.
- c. İkinci olarak 0-255 arasındaki gri seviyelerinden oluşan görüntülerde arka fondaki gürültü olarak da ifade edilebilecek gereksiz piksellerin bir sonraki aşama olan kontur bulma işlemini etkilememesi için eşiklendirme işlemi yapılmaktadır. Bu işlemde deneysel olarak en iyi sonucun 90 gri seviyesinde alındığı görülmüştür. Gri aydınlık seviyesi 90'dan büyük olan pikseller 255 gri seviyesi değerine, 90'dan küçük olan pikseller ise 0 gri seviyesi değerine çekilmiştir. Eşiklendirme sonucunda sadece 0 siyah ve 255 beyaz tonlarında pikseller oluşmaktadır.
- d. Özellik çıkarımında kullanılan Moment fonksiyonları cismin konturu yani dış hatları göz önüne alınarak hesaplandığından eşiklendirilmiş görüntü bir kontur bulma algoritması ile işlenir. Kontur bulma işlemi görüntüyü soldan sağa ve aşağıdan yukarıya tararken cisim ve arka plan arasındaki geçişleri yakalamaya çalışmaya dayanır.
- e. Son adım ise 0-255 gri seviyelerinde kodlanmış görüntülerden ikili olarak (binary) 0 siyah ve 1 beyaz seviyelerinde kodlanmış görüntüler elde edilmesidir.

Üç cisme ait toplam 216 görüntü üzerine bu adımların tamamı uygulanmıştır. Değişik cisimler için farklı açılardan çekilmiş görüntülere ilişkin kontur görüntüleri Ek 2'de verilmiştir.

6.4 İşlenmiş Görüntülerden Özellik Vektörü Çıkarımı

Özellik çıkarma, işlenmiş örneklerden sayısal değerlerle ifade edilen bir dizi özelliği içeren vektör oluşturma işlemi olarak daha önce ifade edilmişti. Ne tür bir ayırma kullanılırsa kullanılsın bu aşama sınıflandırma başarısını doğrudan etkileyecek bir aşamadır. Etkin bir özellik çıkarımı işlemi sonrasında özellik vektörü uzayında tanımlanan örnekler için aynı sınıfta olanlar arasındaki uzaklık en az, farklı sınıftan olanlar arasındaki uzaklık en fazla olmalıdır. Görüntü tanımlayıcıları olarak da ifade edebileceğimiz özellik çıkarcılardan en uygunun hangisinin olduğuna problem domaini göz önüne alınarak karar verilmelidir.

Bu uygulama için görüntünün bir dizi moment envaryantı fonksiyonu ile temsil edilmesi fikri kullanılmıştır. Hu'nun (1962) görüntü veya şekil özellik envaryantları, görüntü veya şekilde boyut, pozisyon ve ölçek değişimleri ve kombinasyonları oluşmasına karşılık değişmezlik özelliği gösteren görüntüye özel katsayılardır. Momentler hesaplanırken normalde içindeki piksel değerlerini de kullanmak olasıdır fakat bu uygulamada cismin hatlarını oluşturan pikseller kullanılmıştır.

Uygulamada kullanılan bu katsayılar Bölüm 3'te (3.18) eşitliğiyle verilmiştir. Fakat daha önce de belirtildiği gibi bu fonksiyonlarda ölçekleme etkisi bir başka ifadeyle kameranın cisimden uzaklığının değişimi etkisi göz önüne alınmamıştır. Bu etki Üstün'de (1999) ifade edildiği gibi dülemsel bir şeklin salınım yarıçapı r 'nin, optik eksen uzaklığı d 'ye uzaklığı ile çarpımının sabit olması kullanılarak ölçeklemeden bağımsızlık bu katsayılar eklenmektedir;

$$\begin{aligned}
 M_1' &= (\mu_{20} + \mu_{02})^{1/2} d = r \cdot d \\
 M_2' &= M_2 / r^4 \\
 M_3' &= M_3 / r^6 \\
 M_4' &= M_4 / r^6 \\
 M_5' &= M_5 / r^{12} \\
 M_6' &= M_6 / r^8 \\
 M_7' &= M_7 / r^{12}
 \end{aligned} \tag{6.1}$$

İlk moment fonksiyonu için d uzaklığının bilinmesi gereklidir fakat bu çalışmada bu uzaklığın bilindiği varsayılmaktadır. Moment fonksiyonlarını bulmada kullanılan program Ek 1'de verilmektedir.

6.5'Eğitim ve Test Verisi Dosyaları Oluşturulması

Sunulmuş bir örnek için tanıma/sınıflandırılma işlemi eğitici veya eğitici sınıflandırma işlevlerinden birini içerebilir. Eğitici sınıflandırmada (örn, diskriminant analizi) örnekler daha önce tanımlanmış bir sınıfın üyeleri olarak belirtilmiştir. Karar verme, önceden sınıfı bilinen örneklerle bakılarak daha sonraki örneklerin hangi gruba ait olduğunu çıkış olarak bildirmektir. Eğitici sınıflandırmada ise (örn, kümeleme) örnekler o ana kadar bilinmeyen sınıfın üyeleri olarak ayrılır.

Bu çalışmada sınıfları bilinen n boyutlu örnek uzayındaki noktalar şeklindeki eğitim örnekleriyle eğitilen ağ, farklı sınıftan örneklerin ayrılabilmesi için karar verme sınırları oluşturmaktadır. Daha sonra bu karar sınırlarının güvenilirliğini denemek amacıyla test örnekleri ağı sunulmakta ve ağın bu örnekleri sınıflandırması istenmektedir.

Daha öncede belirtildiği gibi bu çalışmada cisimlerin (şekil 6.2) xy düzleminde z eksenini etrafında dönmesi sağlanmakta ve görüntü olarak da xz düzleminde izdüşümleri kaydedilmektedir. z eksenini etrafında 0° 'den 360° 'ye 5° aralıklarla rotasyonları neticesinde her cisim için 72 adet toplamda 216 görüntü elde edilmektedir. Her görüntüdeki basit yapının konturu yani dış hatları göz önüne alınarak hesaplanan 7 adet Moment fonksiyonuyla her görüntünün sadece 7 adet sayıyla temsili sağlanmaktadır. Toplam 216 özellik vektöründen oluşan özellik matrisinden eğitim ve test kümesi oluşturma yoluna gidilmektedir. Her cisim için $n = 0,1,2,\dots,35$ olmak üzere cismin başlangıç açısına göre z eksenini etrafında $(10n)^\circ$ ile döndürülmesiyle oluşan görüntülerden elde edilen özellik vektörleri eğitim kümesine, $(10n + 5)^\circ$ ile döndürülmesiyle oluşan görüntülerden elde edilen özellik vektörleri ise test kümesine atılmıştır. Bu aşamada sınıflandırıcının başarısı eğitime kümesi adıyla oluşturduğumuz özellik vektörleriyle eğitilen ağın, test kümesi adıyla oluşturduğumuz özellik vektörleriyle test edilmesi ile ölçülür.

6.6 Sınıflandırma

Bu çalışmada, sınıflandırıcı olarak yapay sinir ağları kullanılmaktadır. Eğitici öğrenme işlemi sonucu eğitim ve test setindeki örnekleri en iyi şekilde sınıflandıran ağ yapısı aranan ağ yapısıdır. Bu amaçla probleme çeşitli yapay sinir ağı yapıları uygulanarak elde edilen etkinliğin derecesi araştırılmaktadır.

3-D bilgisayar görmesi sisteminde cisimlere ait farklı açılardan çekilmiş 2-D görüntüler özellik uzayında temsil edilirken 7 adet katsayı kullanıldığı için ağ modelleri oluşturulurken 7 adet giriş hücresi kullanılmıştır. Üç adet cisim olduğuna göre $\mathbb{R}^7$ özellik uzayına indirgenen örnekler $\{C_1, C_2, C_3\}$ kümelerinden birine haritalanacaktır. Sınıfları temsil etmek üzere birinci cisim için '1', ikinci cisim için '2', üçüncü cisim için '3', sayısal değerleri kullanılmaktadır. $c(i)$ eğitilen ağı sunulan girişler için elde edilen çıkışlar ve c_1, c_2, c_3 sırasıyla birinci, ikinci, üçüncü sınıflara ait doğru sınıflandırılan örnek sayısı olmak üzere;

```

c1=0; c2=0; c3=0;
if c(i)>=0.5 & c(i)<1.5
c1=c1+1;
if c(i)>=1.5 & c(i)<2.5
c2=c2+1;
if c(i)>=2.5 & c(i)<3.5
c3=c3+1;

```

şeklinde karar mekanizması oluşturulmuş ve kullanılan modeller için doğru sınıflandırmaların yüzdesi belirlenmiştir.

Problemin yapısı ve örneklerin uzayda dağılımı göz önünde tutularak sadece lineer karar verme sınırları oluşturabilen ağ yapıları (perseptron vb.) üzerine çalışmaya gerek duyulmamaktadır. Örnek tanıma uygulamalarında evrensel olarak kullanılan çok katmanlı algılayıcı (MLP) ağ modeli ve geriye yayılım (Backpropagation) eğitim algoritması üzerinde yoğunlaşmaktadır. MLP ağ yapıları oluşturmak için Matlab programı içindeki toolbox'lardaki komutlar kullanılmıştır. Bu amaçla 7 giriş hücresi, 12 saklı katman hücresi ve 1 çıkış hücresinden oluşan MLP yapısı kurulmuştur. Saklı katman hücre sayısı belirlenirken en optimum çözümün 12 hücre kullanılarak elde edildiği gözlenmiştir. Ağın eğitim örnekleriyle eğitilmesi sırasında istenen çıkış değerleri olan '1', '2', '3' değerlerini verebilmesi için ağ çıkışındaki hücrede aktivasyon fonksiyonu olarak lineer fonksiyon kullanılmıştır. Saklı katmandaki hücreler için ise tanjant sigmoid ve logaritmik sigmoid fonksiyonları kullanılmıştır. Öğrenme kuralları olarak geriye yayılım algoritmasını farklı şekillerde oluşturan trainlm, trainoss, trainbfg, trainrp öğrenme kuralları kullanılmıştır. Öğrenme oranları değiştirilerek ağ iteratif olarak eğitilmiştir ve sınıflandırma işlemi gerçekleştirilmiştir. Ağın iterasyon sayısı ilk önce 3000 olarak belirlenmiş fakat ağın deneme verilerine aşırı hassas olmasını ifade eden aşırı uydurma problemi oluşmuştur. Bunun üzerine ağı aşırı uydurma probleminden uzaklaştırmak, genelleştirme özelliğini kazandırabilmek için ağın iterasyon sayısı 1000 gibi normal bir değerde tutulmuştur. Hata fonksiyonu olarak ortalama karesel hata kriteri uygulanmış hatanın 10^{-6} değerine yakınsaması hedeflenmiştir. Oluşturulan konfigürasyonlar ve sınıflandırma etkinlikleri Ek 3'te verilmektedir.

İkinci olarak RBF ağları kullanılmıştır. RBF yapılarını oluşturmak üzere Matlab programı içindeki toolbox'lardaki komutlar kullanılmıştır. RBF ağları örnek uzayında radyal

fonksiyonları kullanarak hiperküreler şeklinde karar sınırları oluşturarak sınıflandırma işlemi yapmaktadır. Hiperkürelerin genişlikleri radyal fonksiyondaki genişlik (spread) değerine bağlıdır. Bu çalışmada da kurulan RBF ağında sınıflandırmayı birinci derecede belirleyen genişlik değerleri değiştirilerek en etkili sınıflandırmaya ulaşılmaya çalışılmıştır. RBF ağı yapısı oluşturmada kullanılan newrb ve newrbe komutlarıyla farklı genişlik değerlerine sahip konfigürasyonlar ve sınıflandırma etkinlikleri Ek 3'te verilmiştir.

Üçüncü olarak GRNN ağları kullanılmıştır. GRNN ağlarında ikinci katmanda her bir giriş örneğini temsil eden bir nöron olması gerektiğinden ilk katmanda giriş vektörü boyutumuz olan 108 adet nöronlu bir yapı oluşturulmaktadır. Bu katman RBF ağlarındaki ilk katmanın işlevine benzer bir işlev gerçekleştirir; girişlerle depolanmış örnekler arasındaki uzaklık ölçütünü hesaplayarak çıkış üretirler ve RBF ağlarında olduğu gibi genişlik değerlerini kullanırlar. GRNN yapılarını oluşturmak üzere Matlab programı içindeki toolbox'lardaki komutları farklı genişlik değerler için kullanarak en iyi şekilde sınıflandıran ağ yapısı araştırılmıştır. Oluşturulan konfigürasyonlar ve sınıflandırma etkinlikleri Ek 3'te verilmektedir.

Son olarak algoritma tabanlı oluşturulan GAL ağları kullanılmıştır. Bölüm 5'te de ifade edildiği gibi GAL ağları yapısını giriş verilerine göre belirler. 2. katmanda ağın kaç düğümlle oluşturulacağı, sınıfların örnek uzayında dağılımlarıyla ilişkilidir. Uygulamada kullanılan giriş vektörü ve test vektörü tek olduğu için oluşturulabilecek tek bir konfigürasyon vardır.

7. SONUÇLAR

3-D cisimlerin sınıflandırılmasında; parametrik olmayan yapısı, olasılık dağılımı ile ilgili varsayımda bulunmaması, farklı veri türlerini kullanabilme yeteneği ve sınıflandırmaya yardımcı olabilecek ek bilgileri bütünleştirebilme özellikleri nedeniyle yapay sinir ağı teknolojisi yeni ve etkin bir araç olmuştur.

Bu çalışmada üç boyutlu katı cisimlere ait giriş uzayının boyutlarının optimum seviyeye indirgenmesi işleminden yararlanılarak cisimlerin tanınması temel problem olarak ele alınmıştır. Uygun ağ yapılarının belirlenmesi için yapılan deneylerle ağ büyüklüğünün (saklı katman nöronları, katman sayısı) son derece önemli olduğu vurgulanmış; ağıın kapasitesi, öğrenme zamanı ve genelleştirme yeteneğinin buna bağlı olduğu belirtilmiştir. Büyük boyutlu ağların çalıştırılması daha çok zaman almakta ve daha fazla örnekleme verisine ihtiyaç duymaktadır. Küçük boyutlu ağlar ise yerel minimuma yakınsayarak iyi bir genelleştirme yapamamaktadır. Uygulamada sınıf büyüklüklerini ifade eden üç ayrı cisim için toplam 216 görüntü elde edilmiş uygun vektör çıkarımı işleminden elde edilen özellik vektörleriyle eğitici sınıflandırma yapıları oluşturulmuştur. Çeşitli ağ yapılarında, ağ parametreleri değiştirilerek en iyi sınıflandırma araştırılmıştır. Bu yapılarla ilişkin tüm sonuçlar Ek 3'de verilmektedir.

Çizelge 7.1 Ağ yapılarıyla elde edilen sınıflandırma sonuçları

Ağ Mimarisi	Sınıflandırma Başarısı
MLP	%100
RBF	%95.37
GRNN	%92.6
GAL	%89.8

Uygulama sonuçlarında yapay sinir ağları ile yüksek sınıflandırma oranlarına ulaşılabileceği gösterilmiştir (Çizelge 7.1). Görüntü işleme teknolojisinin diğer dallarında da, oluşturulan özellik çıkarımı metodu ve yapay sinir ağı sınıflayıcılarının kullanım alanı bulabileceğini öngörmek zor değildir. Problemin tanımlanması sırasında yapılan ön kabullerin çalışma etkinliğini doğrudan etkilediğini belirtmek gerekmektedir. Cisimlerin tek bir eksen etrafında değil de üç eksen etrafında döndürülmesi durumunda nasıl sonuçlar alınacağı bilinmemektedir. Seçilen cisimlerin keskin hatlarının olmaması da elde edilen özellik

vektörlerinin sınıf içi dağılımlarının birbirine yakın olmasını sağlamış bu da sınıflandırmanın yüksek doğrulukla yapılabilmesini sağlamıştır.

Sonuç olarak yapay sinir ağları, 3-D cisim tanımasında kuramsal ve pratik açılımlar getiren, gelişmiş, duyarlı ve etkin bir teknoloji olarak değerlendirilebilir.



KAYNAKLAR

Alpaydın, E., Gürgen, F., (1998), "Comparison of Statistical and Neural Classifiers and Their Applications to Optical Character Recognition and Speech Classification", 61-88, Image Processing and Pattern Recognition, Londes C. T.(Derl.), Academic Press, California.

Ayache, N., Faugeras, O. D., (1986), "HYPER: A New Approach for the Recognition and Positioning of Two Dimensional Objects", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI 8, No.1, January 1986.

Duda, R. O., Hart, P. E., Stork D. G., (2000), Pattern Classification-2nd edition, John Wiley & Sons, New York.

Hines, J. W., (1997). Matlab Supplement to Fuzzy and Neural Approaches in Engineering, John Wiley & Sons, Inc., New York.

Hu, M., (1962), "Visual Pattern Recognition by Moment Invariants", IRE Trans. Information Theory, Vol. IT-8, pp.179-187, February 1962.

Jain, A. K., (1989), Fundamentals of Image Processing, Prentice Hall, New Jersey.

Jain, A. K., Duin, R. P. W., Mao J., (2000), "Statistical Pattern Recognition: A Review", IEEE Transactions on Pattern Analysis and Machine Learning, Vol.22, No.1, January 2000.

Khotanzad, A., Lu, J.-H., (1990), "Classification of Invariant Image Representations Using a Neural Network", IEEE Transactions on Acoustics, Speech and Signal Processing, Vol.38, No.6, June 1990.

Laaksonen, J., Oja, E., (1996), "Classification with Learning k-Nearest Neighbors", Proceedings of ICCN'96, June 1996, Washington DC., 1480-1483.

Li, Q., Tufts, D. W., (1997), "Principal Feature Classification", IEEE Transactions on Neural Networks, Vol.8, No.1, January 1997.

Liao, S. X., Pawlak, M., (1998), "On the Accuracy of Zernike Moments for Image Analysis", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.20, No.12, December 1998.

Ölmez, T., (2002), Medikal Görüntüleme Dersi Notları, İ.T.Ü.

Shutler, J. D., Nixon, M. S., Harris, C. J., (2001), "Zernike Velocity Moments for Description and Recognition of Moving Objects", BMVC 2001, 705-714, 2001

Sid-Ahmed, M. A., (1995), Image Processing, McGraw-Hill, Singapore.

Üstün, A., (1999), Cisim Tanıma Problemine Yapay Sinir Ağlarının Uygulanması, Yüksek Lisans Tezi, İ.T.Ü Fen Bilimleri Enstitüsü.

Wechsler, H., Zimmerman, G. L., (1998), "2-D Invariant Object Recognition Using Distributed Associative Memory", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.10, No.6, November 1998.

Zion, B., Shklyar A., Karplus, I., (1999), "Sorting Fish by Computer Vision", Computers and Electronics in Agriculture, Vol.8, 93-104, 1999.



EKLER

EK 1: Moment Fonksiyonları Kullanarak Özellik Vektörleri Oluşturma Programı

EK 2: Kullanılan Cisimlere Ait Farklı Açılardan Alınmış Kontur Görüntüleri

EK 3: Sınıflandırma İşleminde Kullanılan Ağ Konfigürasyonları Tablosu



EK 1: Moment Fonksiyonları Kullanarak Özellik Vektörleri Oluşturma Programı:

```

clear

image=imread('IM00726.bmp','bmp');
image=double(image);
image=255-255*image;

figure;
imshow(image);

mpq00=moment(0,0,0,0,image);
mpq10=moment(1,0,0,0,image);
mpq01=moment(0,1,0,0,image);

xort=mpq10/mpq00;
yort=mpq01/mpq00;

Mu00=moment(0,0,xort,yort,image);
Mu01=moment(0,1,xort,yort,image);
Mu02=moment(0,2,xort,yort,image);
Mu03=moment(0,3,xort,yort,image);

for i=10:13;
    for j=10:13;
        Mu(i,j)=moment(i-10,j-10,xort,yort,image);
    end
end

for i=10:13;
    for j=10:13;
        Nu(i,j)=Mu(i,j)/power( Mu(10,10) , (((i+j-20)/2)+1) );
    end
end

fi(1)=Nu(12,10)+Nu(10,12);

fi(2)=power(Nu(12,10)-Nu(10,12),2)+4*power(Nu(11,11),2);

fi(2)=fi(2)/power(fi(1),2);

fi(3)=power(Nu(13,10)-3*Nu(11,12),2) +power(3*Nu(12,11)-
Nu(10,13),2);

fi(3)=fi(3)/power(fi(1),3);

fi(4)=power(Nu(13,10)+ Nu(11,12),2)+
power(Nu(12,11)+Nu(10,13),2);

fi(4)=fi(4)/power(fi(1),3);

fi(5)=(Nu(13,10)-3*Nu(11,12))*(Nu(13,10)+Nu(11,12)) *
(power(Nu(13,10)+Nu(11,12) ,2)-3*power(Nu(12,11)+Nu(10,13),2))+
(3*Nu(12,11)-Nu(10,13))*(Nu(12,11)+Nu(10,13))*
(3*power(Nu(13,10)+Nu(11,12),2)-power(Nu(12,11)+Nu(10,13),2));

```

```

fi(5)=fi(5)/power(fi(1),6);

fi(6)=(Nu(12,10)-Nu(10,12))*(power(Nu(13,10)+Nu(11,12),2)-
power(Nu(12,11)+Nu(10,13),2))+
4*Nu(11,11)*(Nu(13,10)+Nu(1,12))*(Nu(12,11)+Nu(10,13));

fi(6)=fi(6)/power(fi(1),4);

fi(7)=(3*Nu(12,11)-Nu(10,13))*
(Nu(13,10)+Nu(11,12))*(power(Nu(13,10)+Nu(11,12),2)-
3*power(Nu(12,11)+Nu(10,13),2))+(3*Nu(11,12)-
Nu(13,10))*(Nu(12,11)+Nu(10,13))*(3*power(Nu(13,10)+Nu(11,12),2)-
power(Nu(12,11)+Nu(10,13),2));

fi(7)=fi(7)/power(fi(1),6);

fi=[fi(1) fi(2) fi(3) fi(4) fi(5) fi(6) fi(7)];

```

```

zip=fopen('IM000726.txt','w');
fprintf(zip,'%5f %5f %5f %5f %5f %5f %5f \r\n',fi);
fclose(zip);

```

```

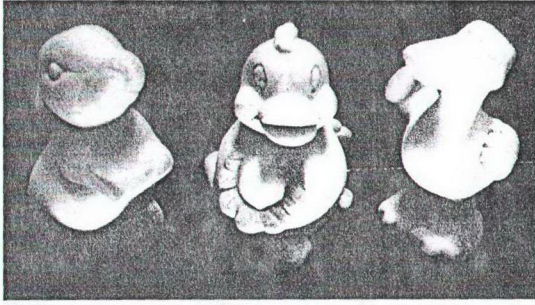
% _____

function[mom]=moment(p,q,ym,xm,img)

mom=0;
for i=1:400
    for j=1:300
        if img(i,j) ==0
            ;
        else
            mom=mom+power((i-xm),p)*power((j-ym),q)*img(i,j);
        end
    end
end
end

```

EK 2: Kullanılan Cisimlere Ait Farklı Açılardan Alınmış Kontur Görüntüleri

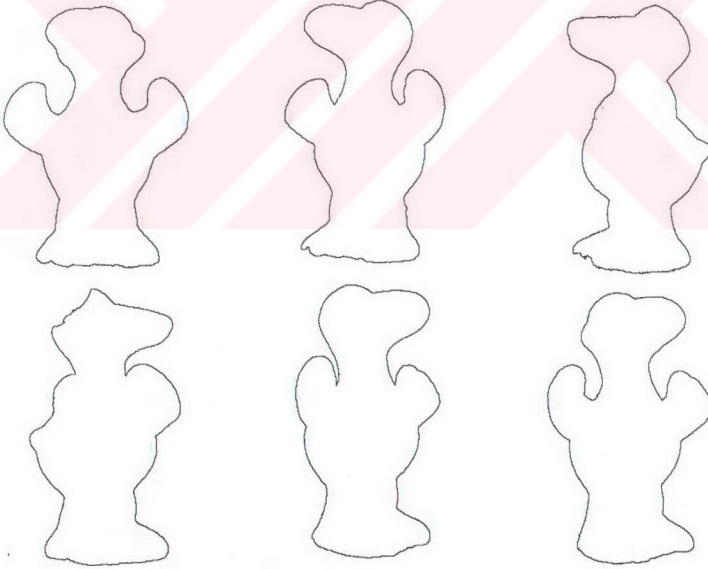


3.cisim

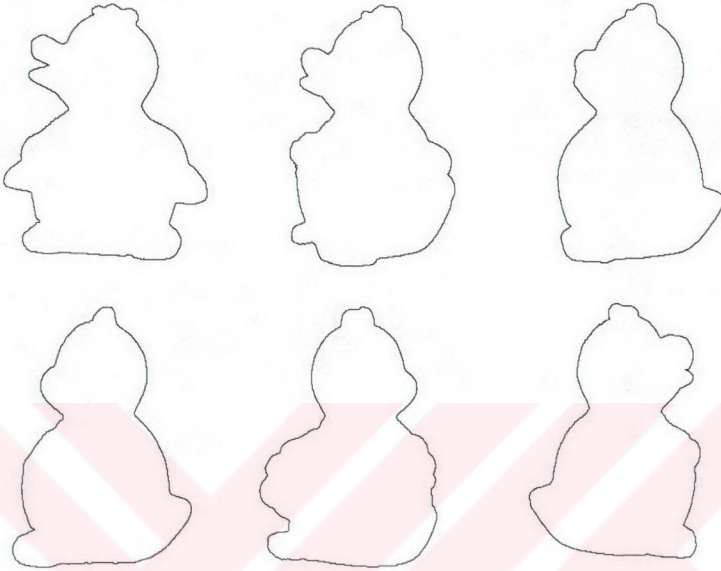
2.cisim

1.cisim

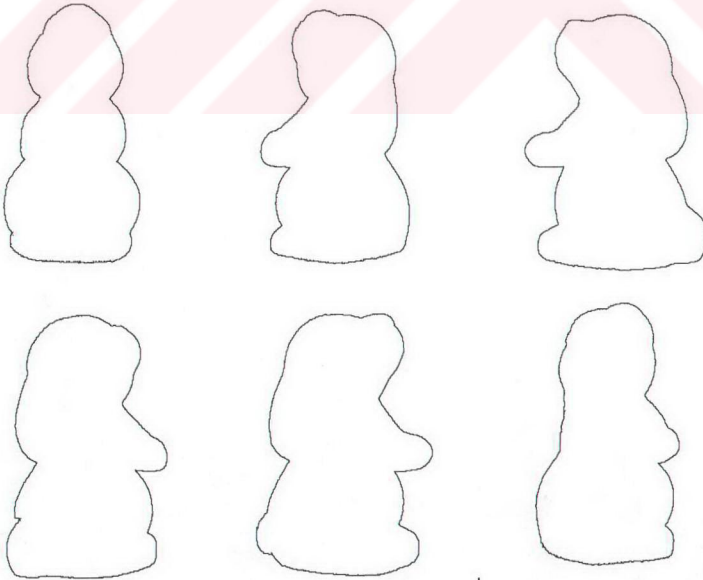
1.Cisim İçin Bazı Kontur Görüntüleri:



2.Cisim İçin Bazı Kontur Görüntüleri:



2.Cisim İçin Bazı Kontur Görüntüleri:



EK 3: Sınıflandırma İşleminde Kullanılan Ağ Konfigürasyonları Tablosu

- Test kümeleri kullanarak ağ yapılarının sınıflandırma doğrulukları gösterilmiştir.

MLP (trainoss)	Öğrenme Oranı	1. Sınıf	2.Sınıf	3.Sınıf
	0.1	36	34	34
	0.2	36	35	35
	0.3	36	35	34
	0.4	36	33	34
	0.5	36	35	35
	0.6	36	35	34
	0.7	36	36	36
	0.8	36	35	35
	0.9	35	35	34
	1	34	36	34

MLP (trainlm)	Öğrenme Oranı	1. Sınıf	2.Sınıf	3.Sınıf
	0.1	30	32	32
	0.2	29	30	34
	0.3	25	33	30
	0.4	18	31	32
	0.5	22	31	30
	0.6	25	30	32
	0.7	22	29	29
	0.8	26	29	31
	0.9	26	33	29
	1	27	31	32

MLP (trainrp)	Öğrenme Oranı	1. Sınıf	2.Sınıf	3.Sınıf
	0.1	36	36	36
	0.2	34	35	34
	0.3	36	35	35
	0.4	36	35	34
	0.5	36	34	34
	0.6	36	35	35
	0.7	36	34	35
	0.8	36	36	35
	0.9	36	36	35
	1	36	36	35

MLP (trainbfg)	Öğrenme Oranı	1. Sınıf	2.Sınıf	3.Sınıf
	0.1	36	34	35
	0.2	35	33	35
	0.3	31	32	33
	0.4	36	35	36
	0.5	32	28	31
	0.6	36	35	34
	0.7	35	35	36
	0.8	35	32	34
	0.9	33	35	36
	1	32	30	31

RBF (newrb)	Genişlik	1. Sınıf	2.Sınıf	3.Sınıf
	0.1	23	30	31
	0.2	27	30	32
	0.3	28	31	34
	0.4	24	31	32
	0.5	28	34	33
	0.6	30	35	32
	0.7	32	35	32
	0.8	33	34	33
	0.9	34	35	34
	1	33	35	33

RBF (newrbe)	Genişlik	1. Sınıf	2.Sınıf	3.Sınıf
	0.1	21	28	26
	0.2	17	22	28
	0.3	21	25	34
	0.4	22	25	35
	0.5	17	22	34
	0.6	15	21	32
	0.7	14	22	32
	0.8	14	22	31
	0.9	10	22	31
	1	14	20	33

GRNN	Genişlik	1. Sınıf	2.Sınıf	3.Sınıf
	0.01	36	34	30
	0.02	36	34	30
	0.03	33	33	28
	0.04	31	34	26
	0.05	30	34	25
	0.06	30	34	23
	0.07	27	34	23
	0.08	26	34	22
	0.09	25	34	21
	0.1	23	34	20

GAL	1. Sınıf	2.Sınıf	3.Sınıf
	34	33	30

ÖZGEÇMİŞ

Doğum tarihi	30.09.1979	
Doğum yeri	Kahramanmaraş	
Lise	1994-1997	Kayseri Fen Lisesi
Lisans	1997-2001	Yıldız Teknik Üniversitesi Elektrik-Elektronik Fak. Elektronik ve Haberleşme Mühendisliği Bölümü
Yüksek Lisans	2001-2003	Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Elektronik ve Hab. Mühendisliği Anabilim Dalı, Elektronik Programı

Çalıştığı kurum(lar)

2001-Devam ediyor YTÜ Elektrik Müh., Araştırma Görevlisi