

**KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

**ÖRÜNTÜ TANIMA AMAÇLI DOĞRUSAL OLMAYAN YÖNTEMLERİN
GELİŞTİRİLMESİ VE UYGULANMASI**

YÜKSEK LİSANS TEZİ

Bilgisayar Müh. Murat AYKUT

**AĞUSTOS 2007
TRABZON**

**KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

**ÖRÜNTÜ TANIMA AMAÇLI DOĞRUSAL OLMAYAN
YÖNTEMLERİN GELİŞTİRİLMESİ VE UYGULANMASI**

Bilgisayar Mühendisi Murat AYKUT

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

Tezin Enstitüye Verildiği Tarih : 09.07.2007

Tezin Savunma Tarihi : 02.08.2007

Tez Danışmanı : Yrd. Doç. Dr. Murat EKİNCİ

Jüri Üyesi : Yrd. Doç. Dr. Cemal KÖSE

Jüri Üyesi : Doç. Dr. Temel KAYIKÇIOĞLU

Enstitü Müdürü : Prof. Dr. Emin Zeki BAŞKENT

Trabzon 2007

ÖNSÖZ

Nesnelerin bilgisayarla otomatik olarak tanınması, teknolojinin gelişmesiyle birlikte araştırmacıların ilgi odağı olmuştur. Günümüzde tıbbî uygulamalardan biometriklerin tanınmasına, saldırı tespitinden optik karakter tanımaya kadar birçok alanda tanıma yöntemleri yaygın olarak kullanılmaktadır. Bu yüzden araştırmacılar bu alanlarda daha iyi performans sağlayacak yöntemler geliştirmek için çalışmaktadır.

Çalışmamda, aynı yöntemin birçok örüntü tanıma probleminde değiştirilmeye gerek kalmadan uygulanabildiği görünüşe dayalı yaklaşımlar ele alınmış ve literatürdeki çalışmalar titizlikle incelenmiştir. Tanıma başarısının artırılması için güncel yaklaşımlar üzerinde durulmuş, doğrusal olmayan yöntemlerin bu yaklaşıma sağlayabileceği yararların araştırılması ana hedefimiz olmuştur. Araştırmalarımızda ayrıca, öznitelik ayrıştırma tekniklerinin etkileri de incelenmiştir. Sonuç olarak, ortaya konulan yaklaşımlarla oldukça yararlı bir çalışma yapılmıştır.

Bu çalışmamda danışmanlığımı üstlenen değerli hocam Yrd. Doç. Dr. Murat EKİNCİ'ye ilgi, alaka ve yardımlarından dolayı teşekkürü bir borç bilirim. Ayrıca mesai arkadaşım Eyüp GEDİKLİ'ye ve aileme de destekleri için teşekkür ederim.

Murat AYKUT
Trabzon 2007

İÇİNDEKİLER

	<u>Sayfa No</u>
ÖNSÖZ.....	II
İÇİNDEKİLER.....	III
ÖZET.....	VI
SUMMARY.....	VII
ŞEKİLLER DİZİNİ.....	VIII
TABLolar DİZİNİ.....	X
SEMBOLLER DİZİNİ.....	XI
1. GENEL BİLGİLER.....	1
1.1. Giriş.....	1
1.2. Öznitelik Ayırıştırma.....	2
1.2.1. Fourier Dönüşümü.....	2
1.2.2. Ayırık Kosinüs Dönüşümü.....	3
1.2.3. Dalgacık Dönüşümü.....	4
1.2.3.1. Görüntü İşleme Açısından Dalgacık Dönüşümü.....	4
1.2.3.2. Dalgacık Analizinin Etkinliği.....	6
1.2.3.3. Dalgacıklar ve Dalgacık Dönüşümü.....	7
1.2.3.4. Filtreler ve Filtre Bankaları.....	8
1.2.3.4.1. Örnek Seyreltme (Altörnekleme)	9
1.2.3.4.2. Aradeğerleme (Üstörnekleme)	11
1.2.3.5. DWT'un Sayısal Gerçekleştirilmesi.....	12
1.2.4. Gabor Filtresi.....	18
1.2.4.1. Gabor Dalgacığının Yapıtaşı: Karmaşık Sinüzoid.....	18
1.2.4.2. Gabor Yapıtaşı: Gaussian Fonksiyonu.....	19
1.2.4.3. Karmaşık Gabor Fonksiyonunun Geliştirilmesi.....	20
1.2.4.4. Karmaşık Gabor Fonksiyonu.....	22
1.2.4.5. Gabor Dalgacıklarının Üretimi.....	24
1.2.4.6. Gabor Filtre Sözlüğü Tasarımı.....	27
1.2.4.7. Gabor Dalgacıklarının Biyolojik Önemi.....	28

1.3.	Öznitelik Çıkartımı.....	29
1.3.1.	Temel Bileşen Analizi.....	29
1.3.1.1.	Giriş.....	29
1.3.1.2.	PCA'nın Matematiksel Arka Planı.....	30
1.3.1.3.	Temel Bileşen Analizinin Uygulanması.....	35
1.3.1.4.	Öz Uzay İzdüşümü İçin Enstantane Metodu.....	41
1.3.2.	Çekirdek Öznitelik Uzayı.....	42
1.3.2.1.	Öznitelik Uzayında Öğrenme.....	43
1.3.2.2.	Öznitelik Uzayına Dolaylı Eşleme.....	45
1.3.2.3.	Çekirdeklerin Oluşturulması.....	49
1.3.2.3.1.	Mercer Teoremi.....	50
1.3.2.4.	Çekirdek Fonksiyonlarına Örnekler.....	51
1.3.2.5.	Çekirdekleri Normalleştirme.....	52
1.3.3.	Çekirdek PCA.....	53
1.3.3.1.	KPCA'nın Matematiksel Olarak İncelenmesi.....	54
1.3.3.2.	KPCA Algoritmasının İşlem Adımları.....	56
1.3.3.3.	KPCA'nın Özellikleri.....	57
1.3.3.3.1.	Hesaplama Karmaşıklığı.....	58
1.3.3.3.2.	Yorumlanabilirlik.....	59
1.3.3.3.3.	Boyut Azaltımı, Öznitelik Çıkartımı ve Yeniden Üretim.....	60
1.3.3.4.	KPCA Deneyleri.....	60
1.4.	Sınıflandırma.....	62
1.4.1.	Euclidean Doğrusal Ayırtaçları.....	62
1.4.2.	En Yakın Komşuluk Kuralı.....	65
1.4.3.	Destek Vektör Makineleri.....	68
1.4.3.1.	Doğrusal Olarak Ayrılabilen Veriler.....	69
1.4.3.2.	Karush-Kuhn-Tucker Koşulları.....	72
1.4.3.3.	SVM ile Sınıflandırma.....	73
1.4.3.4.	Doğrusal Olarak Ayrılamayan Veriler.....	74
1.4.3.5.	Çok Sınıflı Destek Vektör Makineleri.....	78
1.4.3.6.	Doğrusal Olmayan Destek Vektör Makineleri.....	79
1.4.3.7.	SVM'nin Avantajları.....	81
1.4.3.8.	SVM'nin Dezavantajları.....	81

2.	YAPILAN ÇALIŞMALAR, BULGULAR VE İRDELEME.....	83
2.1.	Giriş.....	83
2.2.	Avuç İzine Dayalı İnsan Tanıma.....	84
2.2.1.	Önişlemler.....	85
2.2.2.	Öznitelik Ayrıştırma.....	87
2.2.3.	Öznitelik Çıkartma.....	89
2.2.4.	Sınıflandırma.....	89
2.2.5.	Deneysel Sonuçlar ve Karşılaştırma.....	90
3.	SONUÇLAR.....	103
4.	ÖNERİLER.....	105
5.	KAYNAKLAR.....	107
ÖZGEÇMİŞ		

ÖZET

Son zamanlarda görünüŖe dayalı tanıma yaklaşımı, örüntü tanıma alanında önemli bir araştırma konusu olmuştur. Modele dayalı yaklaşımların problemlere özgü olması ve çıkartılan özniteliklerin uygun seçilmesindeki zorluk, görünüŖe dayalı yaklaşımları daha etkin ve popüler yapmıştır. Yüz, avuç içi, iris gibi biometriklerin tanınması, nesne türlerinin belirlenmesi, tıbbî tanıma uygulamaları, tam veya kısmî olarak görünüŖe dayalı yaklaşımlarla yapılmaktadır.

Bu çalışmada, nesneleri daha iyi karakterize edebilmek amacıyla yüksek istatistiğı sağlayan doğrusal olmayan yaklaşımların kullanılması önerilmektedir. İlk aşama olarak örüntüler öznitelik ayrıştırma ile tayfsal domene aktarılmış, bu domende başarıyı etkilemeyen veya gürültü içeren bilgiler elenmiştir. Daha sonra doğrusal yöntemleri işlem karmaşıklığına neden olmadan doğrusal olmayan biçime dönüştüren çekirdek kavramı üzerinde durularak, öznitelik ayrıştırma için doğrusal ve doğrusal olmayan yaklaşımlar karşılaştırılmıştır. Örüntü tanımada son aşama olan sınıflandırmada ise temel yöntemlerin yanında güncel bir yaklaşım olan SVM ele alınmıştır. Önerdiğimiz yöntem avuç izi veritabanlarında uygulanmıştır. Sonuçlara bakıldığında yapılan çalışmanın oldukça başarılı olduğu görölmektedir.

Anahtar Kelimeler: Örüntü Tanıma, Çekirdek Yöntemler, Temel Bileşen Analizi, Destek Vektör Makineleri, Avuç İzi Tanıma.

SUMMARY

An Application and Development of Nonlinear Methods for Pattern Recognition

Recently, appearance based recognition approach became an important research topic in pattern recognition area. Model based approaches are special to the problem and proper selection of extracted features are very difficult. Therefore, appearance based approaches become more effective and popular. Applications for recognition of biometrics, e.g. face, palm, iris; generic object recognition, medical recognition are being developed in order to partially or fully performed by appearance based approaches.

In this work, for better characterization of objects, usage of nonlinear approaches which provide higher order statistics is proposed. As a first stage, patterns are transformed to the spectral domain with feature decomposition, and redundant or noisy data are removed from this domain. Then, kernel concept which converts linear approaches to the nonlinear approaches without causing computational complexity is followed up with interest. Linear and nonlinear approaches for extracting features are compared. In the classification which is final stage of pattern recognition, in addition to the basic approaches, one of the current approaches SVM is utilized. Our proposed method is applied to the palmprint databases and experimental results show that this method is quite successful.

Key Words: Pattern Recognition, Kernel Methods, Principal Component Analysis, Support Vector Machines, Palmprint Recognition.

ŞEKİLLER DİZİNİ

	<u>Sayfa No</u>
Şekil 1.1. Ayırık dönüşümlerin zaman-frekans grafikleri (a) STFT (b) Dalgacık dönüşümü için.....	6
Şekil 1.2. Örnek bir dalgacık fonksiyonu.....	6
Şekil 1.3. İki ile altörnekleyen bir adımlık filtre bankası.....	10
Şekil 1.4. İşaretin düşük çözünürlük bileşenlerine ayrıştırılması.....	10
Şekil 1.5. Dalgacık domenindeki işaretin yeniden üretilmesi	11
Şekil 1.6. Dalgacıklara örnekler.....	12
Şekil 1.7. Çoklu çözünürlük piramit ayrıştırmasının işlem adımları.....	13
Şekil 1.8. Dalgacık ayrıştırmasının görüntüye uygulanmasının bir adımı.....	16
Şekil 1.9. Frekans domenindeki 3 dalgacığın ve ölçekleme fonksiyonunun tayfsal dağılımı.....	17
Şekil 1.10. Görüntü üzerindeki dalgacık ayrıştırmasının üç seviyesi (a) Orijinal görüntü (b) Birinci seviye (c) İkinci seviye (d) Üçüncü seviye	17
Şekil 1.11. Karmaşık sinüzoidin zaman domenindeki gösterimi.....	18
Şekil 1.12. Tek ve iki boyutlu Gaussian fonksiyonu.....	19
Şekil 1.13. Frekans ve uzay sınırlamasında iki uçdeğer olan sinüs ve delta fonksiyonları.....	21
Şekil 1.14. Gabor temel fonksiyonunun öteleme ve modülasyonla oluşturulması.....	22
Şekil 1.15. Karmaşık Gabor fonksiyonunun iki görünümü.....	23
Şekil 1.16. Gabor dalgacıklarına örnekler.....	26
Şekil 1.17. Gabor filtre sözlüğündeki filtre tepkelerinin yarı tepe genliğini gösteren çizgiler.....	28
Şekil 1.18. Örnek PCA verilerinin 2 boyutlu uzaydaki gösterimi.....	36
Şekil 1.19. Ortalamanın çıkartıldığı normalleştirilmiş veriler ve kovaryans matrisinin özvektörleri.....	36
Şekil 1.20. Verilerin, iki özvektör kullanılarak PCA ile elde edilen öz uzaydaki görüntüleri.....	39
Şekil 1.21. Öznitelik eşlemesi sınıflandırma işlemini kolaylaştırabilir.....	44
Şekil 1.22. Öznitelik uzayına eşlenmiş ikili sınıflandırma problemi örneği.....	46
Şekil 1.23. KPCA tarafından inşa edilmiş öznitelik çıkartıcı.....	56

Şekil 1.24. KPCA'nın temel amacı.....	57
Şekil 1.25. İki boyutlu basit bir KPCA örneği.....	61
Şekil 1.26. Gaussian çekirdeğin kullanıldığı 2 boyutlu bir örneğin ilk 16 doğrusal olmayan temel bileşenine karşı düşen görüntüler.....	62
Şekil 1.27. w_1 sınıfı için karar bölgesi doğrusal ayırtaçların diğer sınıflara bağlı olduğunu göstermektedir.....	64
Şekil 1.28. x giriş vektörü için en büyük karar fonksiyonu sınıflandırıcı.....	64
Şekil 1.29. x öznitelik vektörünün maksimum ilişki yaklaşımı ile sınıflandırılması.....	65
Şekil 1.30. Düzlemde bir nokta kümesi ile tanımlanmış Voronoi mozaığının bir kısmı	66
Şekil 1.31. Minkowski ölçütünde değişik k değerleri için orijinden 1.0 uzaklığındaki noktaları kapsayan bölgeler.....	67
Şekil 1.32. İki adet doğrusal ayrılabilen veri kümesi. (a) Rastgele seçilmiş ayırma düzlemi (b) En büyük marjini verecek ayırma düzlemi..	68
Şekil 1.33. Kuralsal hiper düzlemlerle marjinin grafiksel gösterimi	71
Şekil 1.34. Ayrılabilen veriler için doğrusal ayırma hiper düzlemi.....	75
Şekil 1.35. Doğrusal olarak ayrılabilir veriler (solda) ve ayrılabilen veriler (sağda)...	77
Şekil 1.36. Yumuşak marjin SVM kullanılarak çözülen ikili sınıflandırma problemi örneği.....	80
Şekil 2.1. Görünüşe dayalı tanıma sisteminin temel işlem adımları.....	83
Şekil 2.2. Avuç izi yakalama sisteminin tasarım ilkesi.....	85
Şekil 2.3. 1. Bölge seçme yönteminin temel aşamaları (a) Orijinal görüntü, (b) İkili görüntü, (c) Görüntünün sol tarafından olan uzaklıkla oluşturulan işaret, (d) Avucun orta bölgesi olarak üretilen avuç izi görüntüsünün çıkartılması	86
Şekil 2.4. Değişik dalgacık alt bandlarındaki avuç izi görüntüleri (a) Birinci seviye dalgacık ayrıştırması (b) İkinci seviye dalgacık ayrıştırması (c) Üçüncü seviye dalgacık ayrıştırması.....	88
Şekil 2.5. Gabor dalgacıklarının avuç izine uygulanmasına bir örnek (a) Uygulanan Gabor dalgacıklarının reel bileşenleri (b) Örnek avuç izi görüntüsü (c) Filtrelenmiş avuç izi görüntülerinin genlik değerleri.....	88
Şekil 2.6. İlk avuç izi veritabanına göre ilk 2 ve sonuncu özavuç görüntüleri.....	89
Şekil 2.7. PolyU-I veritabanında KPCA+ SVM ve WED'nin karşılaştırılması.....	96
Şekil 2.8. PolyU-II veritabanında KPCA+ SVM ve WED'nin karşılaştırılması.....	96
Şekil 2.9. PolyU-I veritabanında PCA+ SVM ve WED'nin karşılaştırılması.....	97
Şekil 2.10. Ayrı oturumlardaki görüntüler arasındaki farklılıklar.....	98
Şekil 2.11. Aynı oturumdaki görüntüler arasındaki farklılıklar.....	99

TABLÖLAR DİZİNİ

	<u>Sayfa No</u>
Tablo 1.1. Bazı dalgacık filtrelerinin katsayıları.....	12
Tablo 1.2. PCA için örnek veri kümesi ve ayarlanması (a) orijinal küme (b) ayarlanmış küme	36
Tablo 1.3. Örnek için dönüştürülmüş veriler.....	39
Tablo 1.4. Örnek için sadece en anlamlı özvektörün kullanılmasıyla dönüştürülmüş veriler.....	39
Tablo 2.1. PolyU-I veritabanında 1. bölge seçimi ile elde edilmiş ilk oturumdaki görüntüler arasında LED sınıflandırmasının karşılaştırmalı sonuçları.....	90
Tablo 2.2. PolyU-I veritabanında 1. bölge seçimi ile elde edilmiş her iki oturum görüntüleri arasında LED sınıflandırmasının karşılaştırmalı sonuçları.....	91
Tablo 2.3. PolyU-I veritabanında 1. bölge seçimi ile elde edilmiş her iki oturum görüntüleri arasında LED ve WED sınıflandırmasının karşılaştırmalı sonuçları.....	92
Tablo 2.4. PolyU-I veritabanında 2. bölge seçimi ile elde edilmiş her iki oturum görüntüleri arasında WED sınıflandırmasının karşılaştırmalı sonuçları.....	93
Tablo 2.5. PolyU-I veritabanında 1. bölge seçimi ile elde edilmiş görüntüler için oturumlar arası WED sınıflandırmasının karşılaştırmalı sonuçları.....	93
Tablo 2.6. PolyU-I veritabanında 2. bölge seçimi ile elde edilmiş görüntüler için oturumlar arası WED sınıflandırmasının karşılaştırmalı sonuçları.....	94
Tablo 2.7. PolyU-II veritabanında ilk oturumdaki görüntüler arasında LED ve WED sınıflandırmasının karşılaştırmalı sonuçları.....	94
Tablo 2.8. PolyU-II veritabanında oturumlar arası WED sınıflandırmasının karşılaştırmalı sonuçları.....	95
Tablo 2.9. PolyU-I veritabanında Gabor için farklı oturum testleri.....	98
Tablo 2.10. PolyU-II veritabanında Gabor için farklı oturum testleri.....	98
Tablo 2.11. PCA'da eğitim örneği sayısının önemi.....	99
Tablo 2.12. Uygulanılan yöntemlerin gerçekleştirilme süreleri (saniye olarak).....	100
Tablo 2.13. Karışık oturum testlerinin diğer yöntemlerle kıyaslanması.....	101
Tablo 2.14. Aynı oturum testlerinin diğer yöntemlerle kıyaslanması.....	102

SEMBOLLER DİZİNİ

A.G.	Alçak Geçiren
CWT	Sürekli Dalgacık Dönüşümü (Continuous Wavelet Transform)
DCT	Ayrık Dalgacık Dönüşümü (Discrete Cosine Transform)
DFT	Ayrık Fourier Dönüşümü (Discrete Fourier Transform)
DWT	Ayrık Dalgacık Dönüşümü (Discrete Wavelet Transform)
ERM	Deneyisel Risk Minimizasyonu (Empirical Risk Minimization)
FFT	Hızlı Fourier Dönüşümü (Fast Fourier Transform)
FLD	Fisher Doğrusal Ayırtacı (Fisher Linear Discriminant)
ICA	Bağımsız Bileşen Analizi (Independent Component Analysis)
IDWT	Ters Ayrık Dalgacık Dönüşümü (Inverse Discrete Wavelet Transform)
KFDA	Çekirdek Fisher Ayırtaç Analizi (Kernel Fisher Discriminant Analysis)
KPCA	Çekirdek Temel Bileşen Analizi (Kernel Principal Component Analysis)
LED	Doğrusal Euclidean Ayırtacı (Linear Euclidean Distance)
LPP	Bölge Korumalı İzdüşüm (Locality Preserving Projection)
PCA	Temel Bileşen Analizi (Principal Component Analysis)
SRM	Yapısal Risk Minimizasyonu (Structured Risk Minimization)
STFT	Kısa Zamanlı Fourier Dönüşümü (Short Time Fourier Transform)
SVM	Destek Vektör Makinesi (Support Vector Machine)
UDP	Danışmansız Ayırtaç İzdüşümü (Unsupervised Discriminant Projection)
WED	Ağırlıklandırılmış Euclidean Uzaklığı (Weighted Euclidean Distance)
Y.G.	Yüksek Geçiren
K_{ij}	Gram matris
$k(x_i, x_j)$	Çekirdek fonksiyonu
2-D	2 Boyutlu

1. GENEL BİLGİLER

1.1. Giriş

Örüntü analizi, verilerdeki örüntünün otomatik algılanması ile ilgilenir ve bir çok yapay zeka ve bilgisayar bilimleri probleminde merkezi rol oynar. Örüntü ile belirtilen, herhangi bir ilişki, düzenlilik, verilerde içerilen yapılar ya da görüntünün tamamı gibi ham veriler olabilir. Mevcut verilerdeki önemli örüntüler algılanarak sistem, aynı kaynaktan gelen yeni veriler hakkında tahminde bulunabilir. Bu işleme örüntülerin tanınması adını veriyoruz.

Örüntülerin tanınmasında iki genel yaklaşım vardır. Bunlardan ilki, ham veriden yapısal öznitelikler (şekil, konum, renk bilgisi gibi) çıkartarak sınıflandırma yöntemlerine vermektir. Bu yöntemde çıkartılan özniteliklerin uygun seçilmesi çok büyük önem arz eder. Ayrıca çıkartılacak olan öznitelikler de problemde problemde değişebilir. Diğer yöntem olan görünüme dayalı yöntemlerde [41] görüntünün tamamı ham veri olarak alınır ve işlenir. Bu yöntemlerde görüntünün tamamı ele alındığından çıkartılacak öznitelikler otomatik olarak öznitelik çıkartma yöntemleri ile belirlenmekte ve problemde problemde değişmemektedir. Çıkartılan öznitelikler bizim gözle fark edemeyeceğimiz bilgileri de içerdiğinden sonuçları bakımından tercih edilmektedirler. Bu sebeplerden ötürü son zamanlarda araştırmacıların yoğun ilgisini çekmiştir.

Bu çalışmada öncelikle, global frekans bilgilerinin belirlenmesinde eskiden beri kullanılan Fourier, kosinüs dönüşümleri ile yerleştirilmiş frekans bilgisini içeren ortogonal dalgacıklar ve Gabor dalgacıkları yöntemleri incelenmiştir. Bu dönüşümler sonunda verilerimiz sınıflandırma için daha anlamlı bilgilere sahip olacaktır. Bu öznitelik ayırtırmadan sonra, görünüme dayalı tanımanın ana iskeletini oluşturan PCA yöntemi ele alınmıştır. PCA doğrusal bir dönüşüm olup verilerdeki doğrusal olmayan karakteristiği yansıtmamaktadır. Bu amaçla, doğrusal yöntemleri doğrusal olmayan yöntemlere kolaylıkla ve büyük işlem karmaşıklığına neden olmadan dönüştüren çekirdek kavramı ele alınmıştır. Çekirdek kavramının PCA'ya uygulanması olan KPCA da anlatılarak, doğrusal olmayan öznitelikler konusu işlenmiştir. Son olarak sınıflandırma için geleneksel Euclidean doğrusal ayırtıcı ve en yakın komşuluk yöntemleri ile beraber popüler bir yöntem olan Destek Vektör Makineleri incelenmiştir.

1.2. Öznitelik Ayırıştırma

1.2.1. Fourier Dönüşümü

İki boyutlu işaretlerin biçimlendirilmesinin ve analizinin anlaşılması, bir çok ortogonal dönüşüm mevcut olduğu için mümkündür. Fourier dönüşümü, sadece görüntünün doğasını ve biçimini anlamak için değil, aynı zamanda görüntüyü işlemek için de yaygın olarak kullanılan en önemli araçlardan biridir. Fourier dönüşümü kullanılarak görüntü, her biri kesin bir frekansa sahip değişik yönlerdeki uzaysal sinüzoidlerin kümesi olarak analiz edilir.

Fonksiyon ya da işaret, ayrık örnekler dizisi kullanılarak ayrık biçimde gösterildiğinde ayrık işarete karşı düşen Fourier dönüşümü, Ayrık Fourier Dönüşümü (DFT) olur [1].

N boyutlu $f(x)$ fonksiyonunun tek boyutlu ayrık Fourier dönüşümü (1.1)'de gösterilmektedir.

$$F(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) \exp \left[-j \frac{2\pi ux}{N} \right] \quad (1.1)$$

Tek boyutlu ters DFT ise (1.2)'de gösterilmektedir.

$$f(x) = \sum_{u=0}^{N-1} F(u) \exp \left[j \frac{2\pi ux}{N} \right] \quad (1.2)$$

$M \times N$ büyüklüğünde iki boyutlu $f(x,y)$ işaretinin iki boyutlu ayrık Fourier dönüşümü eşitlik (1.3)'de ifade edilmektedir.

$$F(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \exp \left[-j2\pi \left(\frac{ux}{M} + \frac{vy}{N} \right) \right] \quad (1.3)$$

İki boyutlu ters DFT ise (1.4) ile tanımlanmaktadır.

$$f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) \exp \left[j2\pi \left(\frac{ux}{M} + \frac{vy}{N} \right) \right] \quad (1.4)$$

Reel bir fonksiyonun Fourier dönüşümü karmaşık bir fonksiyondur:

$$F(u, v) = R(u, v) + jI(u, v) \quad (1.5)$$

Burada $R(u, v)$ ve $I(u, v)$, sırasıyla $F(u, v)$ fonksiyonunun reel ve sanal kısımlarıdır.

Genlik fonksiyonu, görüntünün frekans tayfı olarak adlandırılır ve (1.6) eşitliği ile, faz tayfı ise (1.7) ile hesaplanabilir.

$$|F(u, v)| = \sqrt{R^2(u, v) + I^2(u, v)} \quad (1.6)$$

$$\phi(u, v) = \tan^{-1} \left[\frac{I(u, v)}{R(u, v)} \right] \quad (1.7)$$

Ayrık Fourier dönüşümü hesaplama açısından çok pahalı bir yöntemdir. N elemanlı bir küme için N^2 karmaşık çarpma işlemi gerekmektedir. Bu amaçla bir çok araştırma yapılmış ve hızlı Fourier dönüşümü (kısaca FFT) olarak bilinen yöntemler ortaya atılmıştır. Böylece Fourier dönüşümü gerçek zamanlı uygulamalarda rahat bir şekilde kullanılmaya başlanmıştır.

1.2.2. Ayrık Kosinüs Dönüşümü

Ayrık kosinüs dönüşümü (DCT) Ahmed vd. [2] tarafından ortaya atılmış olup bir çok görüntü sıkıştırma algoritması için temel teşkil etmektedir. DCT'nin DFT'ye göre avantajlarından biri karmaşık sayılar üzerinde çalışılmaya gerek olmayışıdır. İleri yönlü DCT eşitliği (1.8) ifadesinde verilmektedir.

$$H(u, v) = \frac{2}{MN} C(u)C(v) \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} h(x, y) \cos \left[\frac{(2x+1)u\pi}{2M} \right] \cos \left[\frac{(2y+1)v\pi}{2N} \right] \quad (1.8)$$

Ters yönlü DCT ise eşitlik (1.9) ile verilmektedir.

$$h(x, y) = \frac{2}{\sqrt{MN}} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} C(u)C(v)H(u, v) \cos\left[\frac{(2x+1)u\pi}{2M}\right] \cos\left[\frac{(2y+1)v\pi}{2N}\right]$$

$$C(\gamma) = \begin{cases} \frac{1}{\sqrt{2}}, & \gamma = 0 \text{ için} \\ 1, & \gamma > 0 \text{ için} \end{cases} \quad (1.9)$$

Forier serileri gibi görüntüler DCT ile temel fonksiyonlar kümesine ayrıştırılabilir. Bu demektir ki görüntü, temel fonksiyonların uygun toplamı ile oluşturulabilir [3].

1.2.3. Dalgacık Dönüşümü

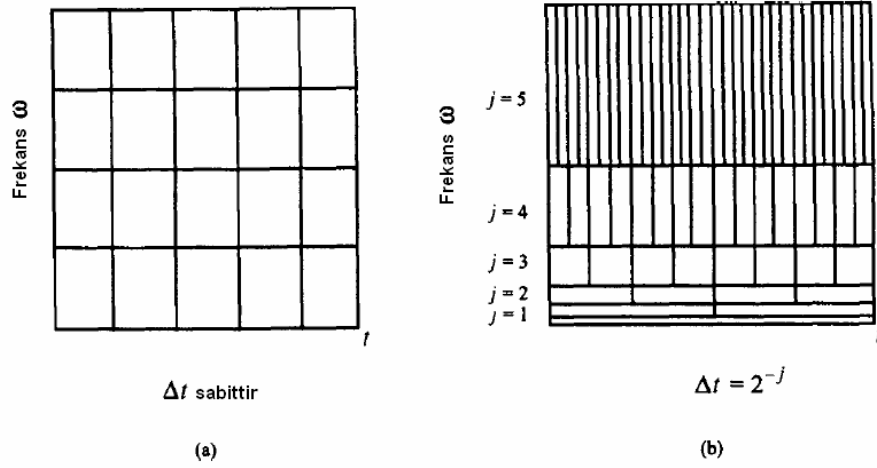
1.2.3.1. Görüntü İşleme Açısından Dalgacık Dönüşümü

Görüntüyü işlemek için bir algoritma seçmeden önce görüntüdeki ne tip bilgiyle (yerel mi, global mi) ilgilendiğimizi belirlememiz gerekir. Uzaysal domende gerçekleştirilen önışlemler görüntüdeki piksellerin komşuluk bilgilerinin kullanılmasıyla yerel bilgi çıkartımı için uygun yöntemlerdir. İlgilendiğimiz global bilgi ise (geometri ve parlaklıkla ilgili global görüntü özellikleri gibi) frekans domeni analizi algoritmaları uygun tercih olur. Bu algoritmalarda ayırık görüntü verisi matris biçiminde gösterilir ve görüntü fonksiyonunun dönüşüm matrisi ile çarpılmasıyla elde edilen ayrılabilir doğrusal dönüşüm görüntünün tümünün gösterimi için temel fonksiyonlar kümesi üretmede kullanılır. Fourier dönüşümü, Walsh dönüşümü, Hadamard dönüşümü, vs. bu tip dönüşüm için birer örnektir. Görüntü dönüşümü yaklaşımları ele alındığında, altta yatan görüntülerin dönüştürülen temel fonksiyonlarla ilişkili olabilecek olan bazı karakteristiklere sahip olduğu varsayılır ve bütün görüntü tek bir varlık olarak düşünülür, parçalara ayıramaz.

Görüntü dönüşümleri (iki boyutlu Fourier dönüşümü gibi) aslında görüntü analizi için çok güçlü ve etkindir. 2 boyutlu görüntü işaretini uzaysal domenden, görüntü işaretinin bir çok karakteristiğinin ortaya çıktığı bir domene dönüştürür. Bununla beraber dönüşüm çekirdeği (Fourier dönüşümünde $\exp[-j2\pi(ux+vy)]$ gibi) global bir fonksiyon olduğu için Fourier dönüşümündeki ikili tümleştirme işlemi görüntünün bütünü bilinmeden elde edilemez. Bu, uzaysal eksen boyunca fonksiyonun herhangi bir noktasında meydana gelebilecek küçük bir değişimin frekans eksenindeki bütün noktalara tesir edeceği anlamına gelir. Eğer $f(x,y)$ görüntü fonksiyonunu $\exp[-j2\pi(ux+vy)]$ için ayarlama fonksiyonu olarak düşünürsek xy eksenlerindeki herhangi bir noktada meydana gelecek bir

değişim bütün u,v eksenlerine yayılır. Diğer bir deyişle Fourier tayfı görüntü işareti hakkında hiç bir yer bilgisi sağlamamaktadır. Fourier tayfindan olayın nerede gerçekleştiği konusunda bir bilgi alamayız. Bu yüzden Fourier dönüşümü hareketsiz veya tek karelik görüntüler için uygundur, ama hareketli veya geçici karakteristikler için uygun değildir. Bu eksikliğin üstesinden gelmek için görüntüde konum ve frekans domeni analizlerini gerçekleştirebilen bir yaklaşıma ihtiyacımız vardır. Böyle bir yaklaşımla görüntünün yerel frekans içeriğini çıkartabiliriz. Bu sebepten, kısa-zamanlı Fourier dönüşümü (short-time Fourier transform=STFT) geliştirilmiştir.

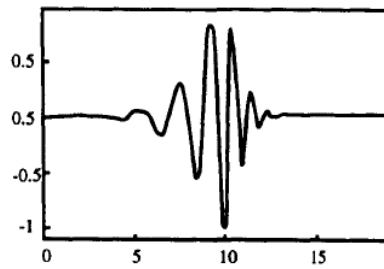
STFT bir boyutlu işaret için zaman-frekans, iki boyutlu görüntü için konum-frekans analizidir. STFT kısaca şu şekilde yorumlanabilir: Düzlemsel görüntüdeki istenilen konumun komşuluğundaki iki boyutlu işaretin yerel frekans içeriğini bilmek istediğimizde verilen görüntüden istenilen kısım bir pencere fonksiyonu ile kaldırılır ve bu kaldırılan kısmın Fourier dönüşümü yapılır. STFT'nin pencereleme doğasından dolayı bu dönüşüm pencerelenmiş Fourier dönüşümü olarak da bilinir. Bu dönüşümün avantajı, xy uzaysal domeninde nerede ve sinyalin hangi frekans içeriğinde gibi bilgilerin sağlanmasıdır. Bununla birlikte bu dönüşümün bazı kusurları da vardır. Bir kere pencere seçildiği zaman pencerenin boyu bütün frekanslar için sabit kalır ve konum-frekans çözünürlüğü işlem boyunca sabitlenir. Bu yüzden değişken boyutlu alanlara sahip bir pencerelemenin mümkünlüğü sorusu ortaya çıkmıştır. Böyle bir teknikte, büyük alan kapsayan (frekansı düşük) bir pencere düşük frekans bilgisini elde etmek için kullanılabilir, ve küçük alana sahip bir pencere yüksek frekans bileşenlerini elde etmek için kullanılabilir. Bu amaç, dalgacık fonksiyonunun $\psi(x,y)$ ve işaretin frekanslar yerine değişik çözünürlük ölçeklerine ayrıştırıldığı dalgacık dönüşümünün geliştirilmesine sebep olmuştur. Çoklu çözünürlük, frekansları ω 'dan $\omega+\Delta\omega$ 'ya tekdüze bandlar yerine ω 'dan 2ω 'ya oktav bandlara böler. Şekil 1.1. (a) ve (b) kısa zamanlı Fourier dönüşümü için zaman-frekans çizimini ve dalgacık analizi için zaman-ölçek çizimini gösterir. Şekilden kısa zamanlı aralıkların yüksek frekanslar için doğal olduğu görülebilir.



Şekil 1.1. Ayrık dönüşümlerin zaman-frekans grafikleri (a) STFT (b) Dalgacık dönüşümü için

1.2.3.2. Dalgacık Analizinin Etkinliği

Dalgacık, Şekil 1.2.'de gösterildiği üzere sinüzoid gibi salınımlı dalga karakteristiğine sahiptir. Bununla beraber, ortalama süresi 0 olan etkin sınırlı sürenin dalga şekline sahip olması ile sinüzoidden ayrılır. Bu özellikten dolayı dalgacık açılımı daha doğru yerel tanımlama ve işaret karakteristiklerinin ayrılmasına imkan tanır. Aksine, bir Fourier katsayısı $-\infty$ 'dan $+\infty$ 'a kadar bütün zamanlar için süren bir bileşen gösterir. Bu sebepten ötürü delta fonksiyonu gibi geçici bir olay sonsuz sayıda sinüs fonksiyonuna ihtiyaç duyacaktır. Bununla beraber, dalgacığı kullandığımızda böyle bir durumla karşılaşmayız. Çünkü, dalgacık katsayısının kendisi, yerel (sınırlı) bir bileşeni temsil eder. Dalgacık, görüntüdeki değişken boyutlardaki, değişken frekanslardaki ve değişken bölgelerdeki yerel öznitelikleri çıkartmak için iyi bir matematiksel araçtır.



Şekil 1.2. Örnek bir dalgacık fonksiyonu

1.2.3.3. Dalgacıklar ve Dalgacık Dönüşümü

Fourier dönüşümü ile dalgacık dönüşümü arasında esaslı bir fark olduğunu hatırlatmak gerekir. Fourier dönüşümünde kullanılan ortogonal temelli fonksiyonlar $\sin(k\omega_0 t)$ ve $\cos(k\omega_0 t)$ 'dir. Eksi sonsuz ile artı sonsuz arasında uzanırlar ve domeninin tamamında 0'dan farklıdırlar. Bu yüzden Fourier dönüşümü yoğun desteğe (compact support) sahip değildir. Kısa zamanlı Fourier dönüşümünün sağlamamasına rağmen dalgacık dönüşümü zaman-frekans bilgisini sağlar. Bu, STFT'de pencere büyüklüğünün, seçilmesinden itibaren bütün frekanslar için sabit kalmasından ötürüdür.

Bununla beraber, dalgacık dönüşümü yoğun desteğe sahiptir. Çünkü kullanılan ana fonksiyon (1.10)'da gösterildiği şekildedir.

$$\psi_{j,k}(t) = 2^{j/2} \psi(2^j t - k) \quad (1.10)$$

Bu ifade (1.11) ve (1.12) koşullarını sağlar.

$$\int_{-\infty}^{\infty} \frac{|\psi(\omega)|^2}{\omega} d\omega < \infty \quad (1.11)$$

$$\int_{-\infty}^{\infty} \psi(t) dt = 0 \quad (1.12)$$

Bu koşullardan ölçekleme parametresinin azaltılmasıyla zaman ve frekansta $\psi_{j,k}$ 'nin desteğinin azaldığı açıkça görülmektedir.

Bir $f(t) \in L^2$ fonksiyonunun sürekli dalgacık dönüşümü, bir $\psi_{j,k}$ çözümleyiş dalgacığına göre şu şekilde tanımlanır:

$$W_{\psi} f(t) = \langle f, \psi_{j,k} \rangle = \int_{-\infty}^{\infty} f(t) \psi_{j,k}(t) dt \quad (1.13)$$

$$\psi_{j,k}(t) = 2^{j/2} \psi(2^j t - k) \quad (1.14)$$

(1.13) ve (1.14) eşitliklerindeki j ve k reel sayılardır. j değişkeni ölçeği yansıtırken k , t eksenindeki öteleme pozisyonunu belirtir. Dalgacık dönüşüm katsayıları her bir temel

fonksiyonla, dönüştürülmüş fonksiyonun iç çarpımı olarak hesaplanır. Ters dönüşüm (1.15) ve (1.16) eşitlikleriyle verilmektedir.

$$f(t) = \sum_j \sum_k c_{j,k} \psi_{j,k}(t) \quad (1.15)$$

$$c_{j,k} = \langle f(t), \psi_{j,k}(t) \rangle = \int_{-\infty}^{\infty} f(t) \psi_{j,k}(t) dt \quad (1.16)$$

(1.16) eşitliğinde hesaplanan $c_{j,k}$ açılım katsayıları kümesi $f(t)$ 'nin ayrık dalgacık dönüşümü (Discrete Wavelet Transform=DWT) olarak adlandırılır. (1.15) eşitliği ise ters dalgacık dönüşümü (Inverse Discrete Wavelet Transform=IDWT) olarak bilinir.

Bir $f(x,y)$ fonksiyonu için dalgacık fonksiyonları ve iki boyutlu sürekli dalgacık dönüşümü sırası ile (1.17) ve (1.18)'de gösterilmektedir.

$$\psi_{j,kx,ky}(x,y) = 2^j \psi(2^j x - k_x, 2^j y - k_y) \quad (1.17)$$

$$W_\psi f(x,y) = \langle f(x,y) \bullet \psi_{j,kx,ky} \rangle = \int \int_{-\infty}^{\infty} f(x,y) \psi_{j,kx,ky}(x,y) dx dy \quad (1.18)$$

İki boyutlu CWT formüllerindeki k_x ve k_y ifadeleri sırası ile x ve y koordinat eksenlerindeki ötelemeleri göstermektedir. Ters 2 boyutlu CWT, (1.19) ve (1.20) eşitlikleriyle belirtilmektedir.

$$f(x,y) = \sum_j \sum_{kx} \sum_{ky} c_{j,kx,ky} \psi_{j,kx,ky}(x,y) \quad (1.19)$$

$$c_{j,kx,ky} = \langle f(x,y), \psi_{j,kx,ky}(x,y) \rangle = \int \int_{-\infty}^{\infty} f(x,y) \psi_{j,kx,ky}(x,y) dx dy \quad (1.20)$$

1.2.3.4. Filtreler ve Filtre Bankaları

Filtre, doğrusal zamanla değişmeyen $h(n)$ operatörüdür. Katlama işlemini gerçekleştirir. Eğer $x(n)$ ve $y(n)$ vektörleri sırasıyla giriş ve çıkış vektörlerini gösterirse (1.21)'de verilen ilişki kurulabilir.

$$y(n) = \sum_k h(k)x(n-k) \quad \text{veya} \quad y = h * x \quad (1.21)$$

Eğer $x(n-k)$ vektörü $n=k$ 'da birim tepke olursa çıkış vektörü (1.22)'deki gibi gösterilmektedir.

$$y(n) = h(n), \quad n = 0, 1, 2, \dots \quad (1.22)$$

Bir işaret alçak frekans ve yüksek frekans içeriklerinden oluşur. Alçak frekans içeriği çoğunlukla işaretin en önemli kısmını teşkil eder. Yüksek frekans içeriği ayrıntı bilgisini bildirir. Dalgacık analizinde yaygın olarak kullanılan iki teknik terim vardır: yaklaştırma (A) ve ayrıntı (D). Yaklaştırma, uzatılmış dalgacıkla eşleşebilecek olan, işaretin yüksek ölçek faktörü ve düşük frekans bileşenini gösterir. Ayrıntı da sıkıştırılmış dalgacıkla ifade edilebilecek olan, işaretin düşük ölçek faktörü ve yüksek frekans bileşenini gösterir. İşaretin bu iki bileşeni bir filtre bankası kullanılarak ayırt edilebilir. Bir filtre bankası giriş işaretini frekans bandlarına ayırt etmek için kullanılan filtreler kümesidir. Burada, banka için çoğunlukla iki filtre (yüksek geçiren ve alçak geçiren) kullanılacaktır. İşaretin düşük frekans bileşeni alçak geçiren filtre ile yüksek frekans bilgileri de yüksek geçiren filtre ile çıkışta elde edilir.

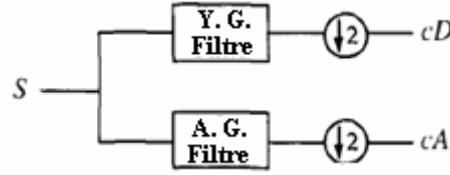
1.2.3.4.1. Örnek Seyreltme (Altörnekleme)

İşaretlerin yüksek frekans ve düşük frekans bileşenleri yüksek ve alçak geçiren filtreleri içeren filtre bankası ile ayırt edilebilir. Filtre bankasının çıkışında örnek sayısının toplamı iki kat olacaktır. Örnek sayısını önceki ile aynı tutmak için örnek seyreltmeye ya da diğer adıyla altörneklemeye ihtiyaç duyulacaktır. Bu işlemi kolayca gerçekleştirmek için sadece çift örnekleri almak yeterli olacaktır. Bu durumda $x(n)$ girişi ve $y(n)$ çıkışı arasındaki ilişki (1.23)'te gösterilmektedir.

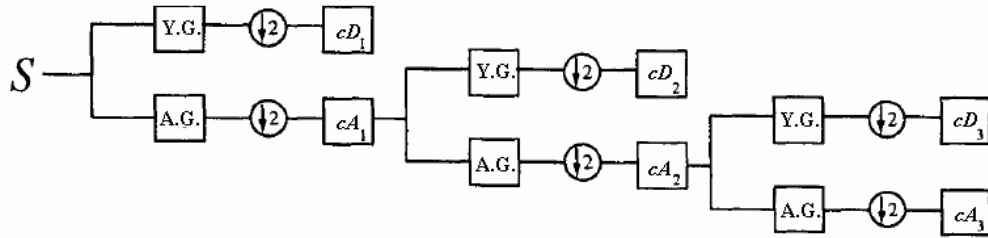
$$\begin{aligned} y(n) &= x(2n), \quad n \in \mathbb{Z} \\ \text{veya} \\ y(n) &= \sum_k x(n) \delta(n - 2k), \quad k \in \mathbb{Z} \end{aligned} \quad (1.23)$$

(1.23) ifadesindeki $\delta(n - 2k)$, birim tepkeler dizisidir.

Şekil 1.3., 2 ile altörneklemeyi içeren filtre bankasının bir adımını göstermektedir. Burada S , orijinal $x(n)$ işaretini gösterir. cA yaklaştırması orijinal işaretin yarı çözünürlükteki düşük frekans bileşenini, cD ayrıtması da yüksek frekans bileşenini gösterir. Bu ayrıştırma işlemi ardışıl yaklaşıklama ile tekrarlanabilir. Algoritma, dalgacık katsayı kümesini oluşturan cD_1 , cD_2 , cD_3 , vs. çıkışlarını vermektedir. Böylece S işareti Şekil 1.4.'te de gösterildiği gibi bir çok alçak çözünürlük bileşenine ayrılmış olur. Şekilde cD_2 , cD_1 'in yarı çözünürlüğünde; cD_3 de cD_2 'nin yarı çözünürlüğündedir. Bunlar dalgacık katsayıları olup işaretin ayrıtma bilgisini vermektedirler. Ayrıştırma işlemi her bir ayrıtma bilgisi tek bir örnek veya görüntüde tek bir piksel içerene kadar devam edecektir. Bununla beraber, uygun seviye sayısı her bir problem için farklı da seçilebilir.



Şekil 1.3. İki ile altörnekleyen bir adımlık filtre bankası



Şekil 1.4. İşaretin düşük çözünürlük bileşenlerine ayrıştırılması

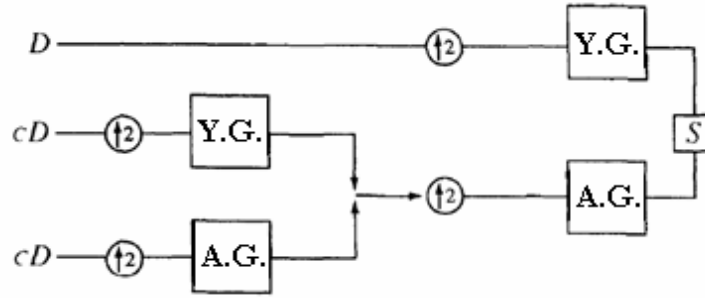
Dalgacık çözümleme ve birleştirmede 3 bakış açısı mevcuttur: (a) İşareti bir takım dalgacık katsayısı olarak göstermek için parçalamak – bu, ayrık dalgacık dönüşümü (DWT) olarak adlandırdığımızdır; (b) dalgacık katsayılarını değiştirmek – dalgacık dönüşümü domeninde işlem yapmak. Gürültü kaldırmak amacı ile gürültüye benzeyen bu istenmeyen bileşenleri arayabilir ve kaldırabiliriz. Benzer şekilde veri sıkıştırma için önemsiz olan bu dönüşüm katsayılarını göz ardı edebiliriz. (c) İşareti değiştirdikten sonra katsayılarla yeniden inşa etmek. Bu işlem ters ayrık dalgacık dönüşümü (IDWT) olarak adlandırılır.

1.2.3.4.2. Aradeğerleme (Üstörnekleme)

Dalgacık katsayılarından orijinal işareti üretmek için üstörnekleme (veya aradeğerleme) ve ardından katlama yapmak gerekir. Kısaca üstörnekleme, işaretin bileşenlerini uzatmak için örnekler arasına sıfırlar konarak gerçekleştirilen bir işlemdir. Matematiksel olarak, üstörneklemeden sonra elde edilen sonuçlar (1.24)'te gösterilmektedir.

$$x'(n) = \begin{cases} y(n/2), & n = -\infty, \dots, -4, -2, 0, 2, 4, 6, \dots, \infty \text{ (çift } n \text{) için} \\ 0, & n = -\infty, \dots, -5, -3, -1, 1, 3, 5, \dots, \infty \text{ (tek } n \text{) için} \end{cases} \quad (1.24)$$

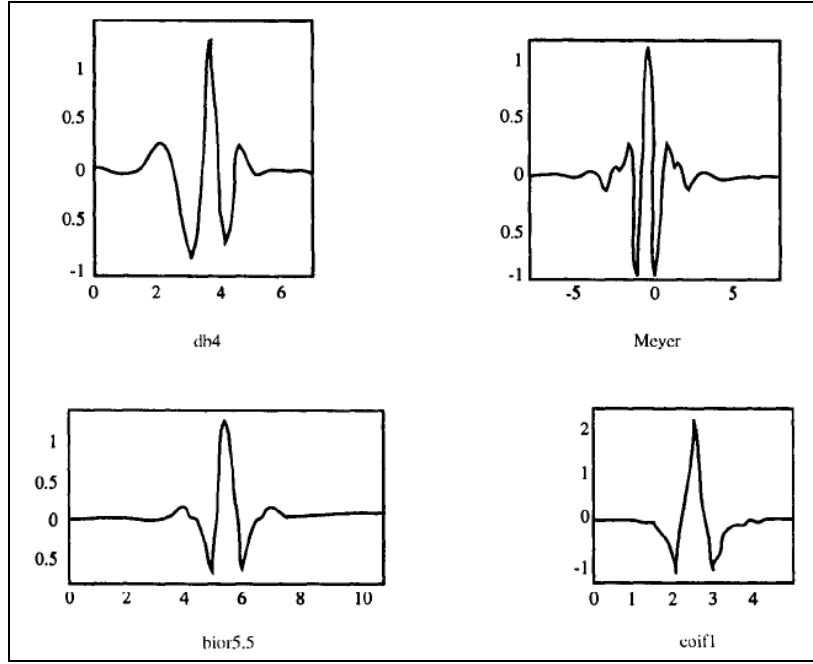
(1.24) ifadesinde $x'(n)$ ve $y(n)$ sırasıyla, işlem kısmının çıkış ve girişleridir. Şekil 1.5.'te işaretin yeniden eldesi süreci gösterilmektedir.



Şekil 1.5. Dalgacık domenindeki işaretin yeniden üretilmesi

Şekil 1.3. ve Şekil 1.4. katlama işlemini ve ardından gelen işaret üzerindeki altörneklemeyi göstermekte, Şekil 1.5. de üstörneklemeyi gerçekleştiren ters işlem dizisini göstermektedir. Bu iki işlem kümesi ayrık dalgacık dönüşümü ve ters ayrık dalgacık dönüşümü algoritmalarındaki en önemli yapı taşlarıdır.

Dalgacık katsayıları üzerinde değişiklik yapılmadığında, dalgacık dönüşümünün tersinirlik gereksiniminden dolayı değişik ölçeklerdeki bileşenlerden orijinal fonksiyon yeniden elde edilecektir. Şekil 1.6.'da değişik dalgacık örnekleri ve Tablo 1.1.'de bazı dalgacık filtre bankalarına örnekler verilmiştir [4].



Şekil 1.6. Dalgacıklara örnekler

Tablo 1.1. Bazı dalgacık filtrelerinin katsayıları

k	<i>Haar</i>	<i>Daub 4</i>	<i>Daub 6</i>	<i>Daub 8</i>	<i>Coiflet 6</i>	<i>Symmlet 8</i>
0	1	0.48296291	0.33670552	0.23037781	0.03858077	0.03222310
1	1	0.83651630	0.80689150	0.71484657	-0.12696912	-0.01260396
2		0.22414386	0.45987750	0.63088076	-0.07716155	-0.09921954
3		-0.12940952	-0.13501102	-0.02798376	0.60749164	0.29785779
4			-0.08544127	-0.18703481	0.74568755	0.80373875
5			0.03522629	0.03084138	0.22658426	0.49761866
6				0.03288301		-0.02963552
7				-0.01059740		-0.07576571

1.2.3.5. DWT'un Sayısal Gerçekleştirilmesi

Ayrık dalgacık dönüşümünün tanımının hatırlanması için (1.13) ve (1.14) eşitlikleri tekrar gözden geçirilmelidir.

Dalgacık dönüşümü işlemi aslında bir grup katsayının hesaplanmasıdır. Bu katsayıların her biri $f(t)$ giriş fonksiyonunun bir çeşit ψ temel fonksiyonu ile iç çarpımıdır. Bu yüzden (1.13) ifadesi (1.25)'te gösterilen biçime getirilebilir.

$$c(\text{ölçek, pozisyon}) = \int_{-\infty}^{\infty} f(t)\psi(\text{ölçek, pozisyon})dt \quad (1.25)$$

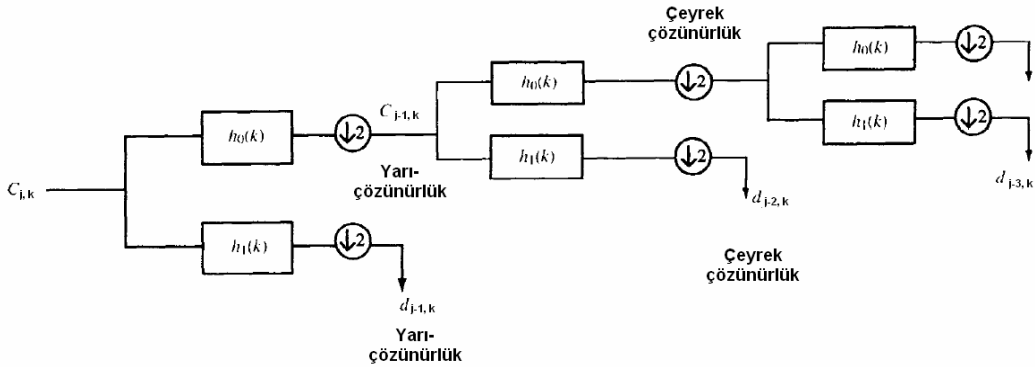
(1.25) eşitliğinde c dalgacık katsayısı, $f(t)$ işaret ve ψ (ölçek, pozisyon) ψ temel fonksiyonunun j ölçek ve k uzaklığındaki biçimidir. Bu, c (ölçek, pozisyon) katsayısının, $f(t)$ giriş fonksiyonu ile temel fonksiyonun bu özel hali arasındaki benzerliğin derecesini verdiği anlamına gelir. 2^j ölçekli k ötelemeye sahip $c_{j,k}$ açılım katsayılarının kümesi $f(t)$ fonksiyonunu temsil için temel fonksiyon üzerinde genlik ağırlıklandırılmış çarpan olarak kullanılabilir.

$$f(t) = \sum_j \sum_k c_{j,k} \psi_{j,k}(t) \quad (1.26)$$

Temel fonksiyonlar iyi seçildiğinden ve biri diğerine dik olduğundan herhangi iki temel fonksiyon arasındaki iç çarpım 0 olacaktır. Bu, iki fonksiyonun birbirinden tamamen farklı olduğunu belirtir. Bir işaret, temel fonksiyonların hiç birine benzemeyen bileşenlerden meydana getirilir. Bu bileşenler kendilerini benzer temel fonksiyonlar için büyük katsayılarla, benzemeyenler için de küçük katsayılarla gösterecektir. Bu yüzden, bir kaçı hariç katsayıların çoğu küçük olacaktır. Böylece işaret az sayıda dönüşüm katsayısı ile sıkı bir şekilde temsil edilmiş olacaktır. Bu, ayrık dalgacık dönüşümünün kalbi sayılabilecek öneme sahiptir.

Dalgacık dönüşümü domeninde istenmeyen bileşenler (gürültü gibi) temel fonksiyonlardan bir veya bir kaçına benzediğinde onları araştırmak kolay olacaktır. Gürültü kaldırma işlemi sadece karşı düşen dönüşüm katsayılarını kaldırarak ya da sıfırlayarak ve (1.26) eşitliği ile işareti yeniden oluşturarak gerçekleştirilir.

Şekil 1.7., dalgacık katsayılarını oluşturmak için çoklu çözünürlük piramit ayrıştırmasını göstermektedir. Bu şekildeki $c_{j,k}$, v_j uzayında olduğu varsayılan orijinal işaretteki işaret bileşenini göstermektedir.



Şekil 1.7. Çoklu çözünürlük piramit ayrıştırmasının işlem adımları

$$c_{j,k} = \langle f, \phi_{j,k} \rangle = \int f(t) \phi(2^j t - k) dt \quad (1.27)$$

Böylece $f(t)$ fonksiyonu şu şekilde ifade edilebilir:

$$f(t) = \sum_k c_{j-1,k} \phi_{j-1,k}(t) + \sum_k d_{j-1,k} \psi_{j-1,k}(t) \quad (1.28)$$

Burada $c_{j-1,k}$ yarı çözünürlük, alçak frekans işaret bileşeninin katsayılarıdır. $d_{j-1,k}$ da yarı çözünürlük, yüksek frekans işaret bileşeninin katsayılarıdır. $d_{j-2,k}$ çeyrek çözünürlük yüksek frekans bileşenini ve $d_{j-3,k}$ da 1/8 çözünürlük yüksek frekans bileşenini vs. gösterir. Eğer $N=2^m$ ise m adımdan sonra N örneğe sahip bir işaret tek bir veri noktası olacaktır. Tek boyutlu işaretin dalgacık dönüşümünde geliştirilen kavramlar, iki boyutlu işaretlerde uygulanmak üzere değiştirilebilirler.

İki boyutlu sürekli dalgacık dönüşümü ifadesini (1.18) ele alalım. Burada k_x, k_y sırasıyla x ve y eksenlerindeki ötelemeleri göstermektedir. $\psi_{j,k_x,k_y}(x, y)$, $k_x, k_y = 0, 1, 2, \dots$ temel dalgacığın değişik biçimlerini göstermektedir. Bu filtrelerin her biri iki boyutlu vuruş tepkisi ve sadece görüntünün kısmi bir bölgesinde değişik boylardaki nesnelere cevap olacaktır. Diğer bir deyişle, eğer görüntü en azından bir $\psi_j(u, v)$ 'nin 0'dan farklı olduğu belirli bir aralığa band sınırlı ise $f(x, y)$ o filtrenin çıkışından yeniden elde edilebilir.

İki boyutlu işareti (veya bir görüntüyü) göstermek için iki boyutlu dalgacıklar ve iki boyutlu ölçekleme fonksiyonları kullanılır. İki boyutlu ölçekleme fonksiyonu $f(x, y)$ görüntü fonksiyonu için 2^j ölçeğinde ortogonal bir fonksiyondur (1.17). İki boyutlu ölçekleme fonksiyonunun ayrılabilir olduğu durumlar için (1.29) eşitliği yazılabilir.

$$\phi_{k_x,k_y} = \phi(2^j x - k_x) \phi(2^j y - k_y) \quad (1.29)$$

Eğer $\psi_{k_x,k_y}(x, y)$, onun eş dalgacığı ise üç değişik 2 boyutlu dalgacık daha oluşturabiliriz:

$$\psi_{k_x,k_y}^1(x, y) = \phi(2^j x - k_x) \psi(2^j y - k_y) \quad (1.30)$$

$$\psi_{k_x,k_y}^2(x, y) = \psi(2^j x - k_x) \phi(2^j y - k_y) \quad (1.31)$$

$$\psi_{k_x, k_y}^3(x, y) = \psi(2^j x - k_x) \psi(2^j y - k_y) \quad (1.32)$$

ψ sembollerindeki üst simgeler sadece indislerdir. $\psi_{k_x, k_y}^l(x, y)$, $l=1,2,3$ katsayılarının her biri (1.33) koşulunu sağlayan dalgacıklardır:

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \psi_{k_x, k_y}(x, y) dx dy = 0, \quad l=1,2,3 \text{ için} \quad (1.33)$$

Mesela $f(x, y)$ görüntüsü ($j=0$, ölçek $2^j=2^0=1$) ile başlanılsın. Görüntü, aşama aşama işlenecektir. Dönüşümün her bir aşamasında, $f(x, y)$ 'nin dalgacık tabanlı görüntülerden biri ile iç çarpımı ve x ve y 'de 2 'ye altörneklenmesi sonucu üretilmesiyle elde edilen 4 tane çeyrek boy lu görüntüye ayrıştırılır. İlk ayrıştırma aşaması için (1.34) eşitliği verilebilir.

$$f_{-1}^0(x, y) = c_{-1, k_x, k_y}^{LL} = \langle f(x, y) \phi(2^{-1} x - k_x) \phi(2^{-1} y - k_y) \rangle \quad (1.34)$$

(1.34) eşitliği $f_{-1}^0(x, y)$ 'nin 2^{-1} kaba çözünürlüğünde yaklaştırma katsayılarını veren ilk alt görüntüdür. Diğer üç alt görüntü sırasıyla şunlardır:

$$f_{-1}^1(x, y) = d_{-1, k_x, k_y}^{LH} = \langle f(x, y) \phi(2^{-1} x - k_x) \psi(2^{-1} y - k_y) \rangle \quad (1.35)$$

$$f_{-1}^2(x, y) = d_{-1, k_x, k_y}^{HL} = \langle f(x, y) \psi(2^{-1} x - k_x) \phi(2^{-1} y - k_y) \rangle \quad (1.36)$$

$$f_{-1}^3(x, y) = d_{-1, k_x, k_y}^{HH} = \langle f(x, y) \psi(2^{-1} x - k_x) \psi(2^{-1} y - k_y) \rangle \quad (1.37)$$

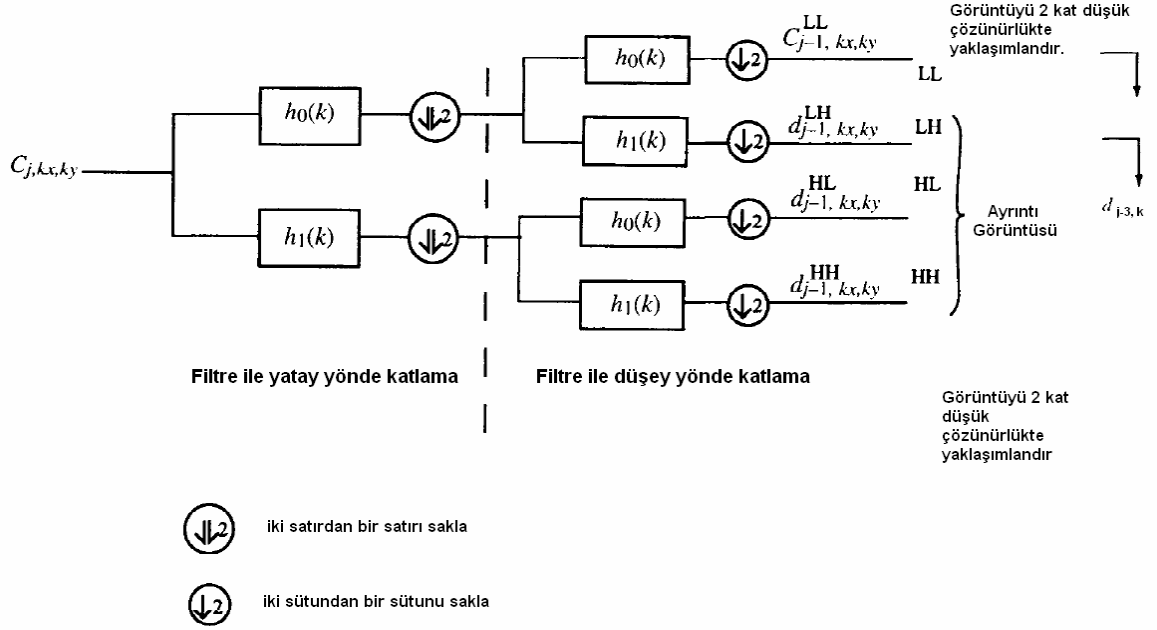
Burada d_{-1, k_x, k_y}^{LH} , d_{-1, k_x, k_y}^{HL} ve d_{-1, k_x, k_y}^{HH} , 2^{-1} kaba çözünürlükteki ayrıntı katsayılarıdır.

İki boyutlu görüntü fonksiyonu $f(x, y)$, (1.38)'de gösterildiği üzere fonksiyonların birleşimi şeklinde ifade edilebilir.

$$\begin{aligned} f(x, y) = & \sum_{k_x} \sum_{k_y} c_{-1, k_x, k_y}^{LL} \phi_{-1, k_x, k_y} + \sum_{k_x} \sum_{k_y} d_{-1, k_x, k_y}^{LH} \psi_{-1, k_x, k_y}^{LH} \\ & + \sum_{k_x} \sum_{k_y} d_{-1, k_x, k_y}^{HL} \psi_{-1, k_x, k_y}^{HL} + \sum_{k_x} \sum_{k_y} d_{-1, k_x, k_y}^{HH} \psi_{-1, k_x, k_y}^{HH} \end{aligned} \quad (1.38)$$

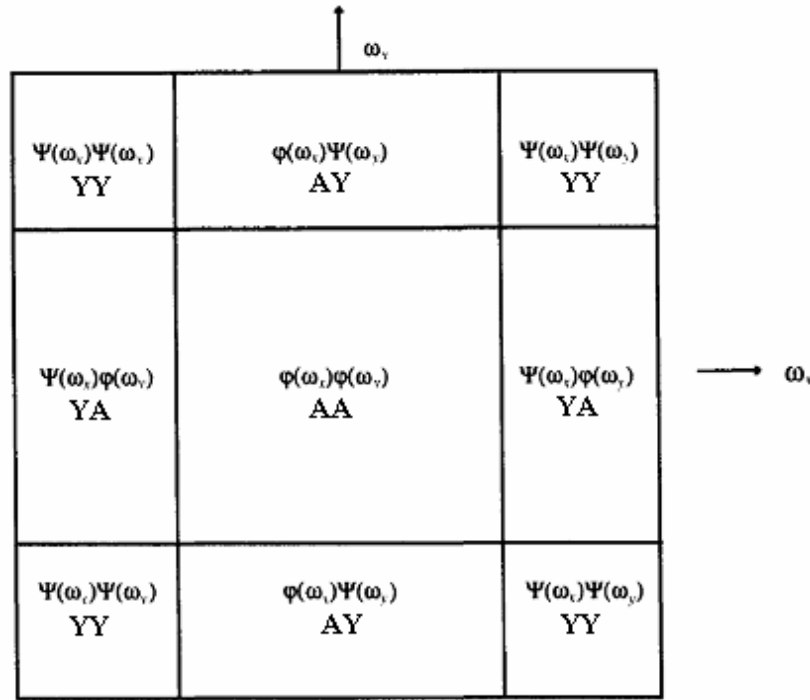
$$c_{-1, k_x, k_y}^{LL} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \phi_{-1, k_x, k_y}(x, y) dx dy \quad (1.39)$$

Şekil 1.8., görüntünün ayrıştırmasının bir adımını göstermektedir. $c_{-1,kx,ky}^{LL}$, $d_{-1,kx,ky}^{LH}$, $d_{-1,kx,ky}^{HL}$ ve $d_{-1,kx,ky}^{HH}$ ayrılabilir işaret filtreleri ile iki koordinat eksenini boyunca hesaplanabilir. 2 boyutlu DWT, x ve y eksenleri boyunca tek boyutlu DWT olarak görülebilir.

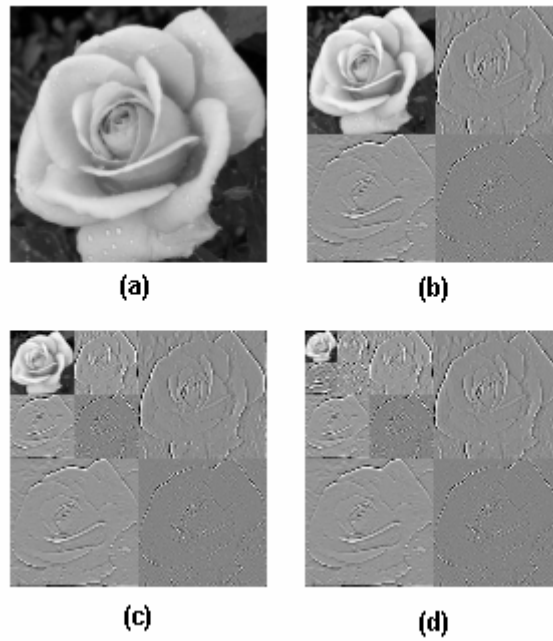


Şekil 1.8. Dalgacık ayrıştırmasının görüntüye uygulanmasının bir adımı

Şekil 1.9., ölçekleme fonksiyonunun ve frekans domenindeki dalgacıkların her birinin tayfsal dağılımını göstermektedir. İki boyutlu tayfsal düzlemin değişik bölgelerini kaplarlar. AA, AY, YA ve YY olarak etiketlenen tayfsal bantlar iki boyutlu yaklaştırma fonksiyonunun tayfına ve $\psi_{kx,ky}^1(x,y)$, $\psi_{kx,ky}^2(x,y)$ ve $\psi_{kx,ky}^3(x,y)$ dalgacıklarına karşı düşer. A ve Y sembolleri filtrelerin alçak geçiren (A) veya yüksek geçiren (Y) olduğunu belirler. AA olarak belirlenen bölgede x ve y yönlerinin her ikisinde de alçak geçiren filtre uygulanmıştır. Bu, ölçekleme fonksiyonunun tayfsal dağılımıdır. AY olarak etiketlenmiş bölge, x yönünde alçak geçiren, y yönünde yüksek geçiren tayfsal içeriği gösterir. Bu, $\psi_{kx,ky}^1(x,y)$ dalgacığının tayfsal dağılımıdır. Benzer şekilde YA ve YY de düşünülebilir. Yaklaştırma alt görüntüsü AA, ayrıştırılmaya devam edilebilir ve düşük çözünürlük yaklaştırma görüntüsü ile ayrıntı alt görüntüleri elde edilir [5]. Şekil 1.10., üç seviyeli dalgacık ayrıştırmasını göstermektedir.



Şekil 1.9. Frekans domenindeki 3 dalgacığın ve ölçekleme fonksiyonunun tayfsal dağılımı



Şekil 1.10. Görüntü üzerindeki dalgacık ayrıştırmasının üç seviyesi
 (a) Orijinal görüntü (b) Birinci seviye (c) İkinci seviye
 (d) Üçüncü seviye

1.2.4. Gabor Filtresi

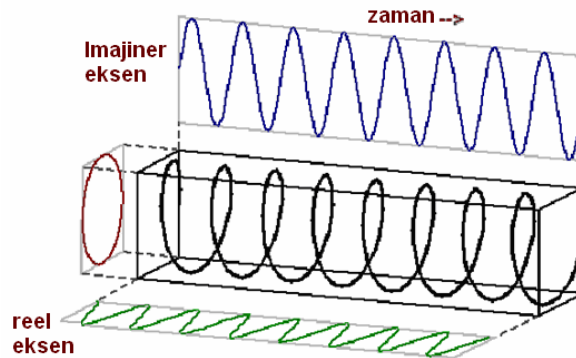
Gabor filtresi, doku analizi, bölütleme, sınıflandırma gibi bir çok görüntü işleme uygulamasında yaygın olarak kullanılan dalgacık filtrelerine bir örnektir. Bu tür uygulamaların hepsinde görüntünün uzaysal frekans bileşenlerinin yerleştirilmiş biçimde analiz edilmesi gerekir. Yerleştirilmiş frekans analizi için genişliği karmaşık sinüzoidin frekansı ile ayarlanan Gaussian zarfa sahip olunması istenir. Gabor dalgacıkları uzayda daha iyi yerleşmeyi sağlayan aynı türden bir fonksiyon sınıfı biçimlendirir [1].

Gabor filtresini daha iyi anlamak için temel yapıtaşları olan karmaşık sinüzoid ile Gaussian fonksiyonlarını öncelikle ele alalım.

1.2.4.1. Gabor Dalgacığının Yapıtaşı: Karmaşık Sinüzoid

Sinüzoid, sinüs fonksiyonu eğrisidir. Bazen, sinüs fonksiyonu dairenin etrafında dolanan noktanın yüksekliği olarak düşünülebileceğinden dairesel fonksiyon olarak da adlandırılır.

Karmaşık sinüzoid, 2 boyutlu sinüzoiddir. İlk boyut reel eksenidir ve kosinüs dalgası içerir. İkinci boyut ilk boyuda diktir, imajiner kısımdır ve sinüs dalgası içerir. Şekil 1.11. karmaşık sinüzoidi zaman domeninde göstermektedir. Karmaşık sinüzoidin reel ve imajiner kısımlarının vektörel birleşimi sarmal dairesel eğri olarak gözükmemektedir.



Şekil 1.11. Karmaşık sinüzoidin zaman domenindeki gösterimi

Karmaşık sinüzoid, matematiksel olarak (1.40)'da ifade edilmektedir.

$$s(x, y) = \exp(j(2\pi(u_0x + v_0y) + P)) \quad (1.40)$$

(1.40) eşitliğinde u_0 , v_0 kartezyen koordinatlardaki uzaysal frekanslar, P ise fazdır.

Karmaşık sinüzoidin reel ve imajiner kısımlarının matematiksel ifadesi (1.41) ve (1.42) eşitliklerinde verilmektedir.

$$\text{Re}(s(x, y)) = \cos(2\pi(u_0x + v_0y) + P) \quad (1.41)$$

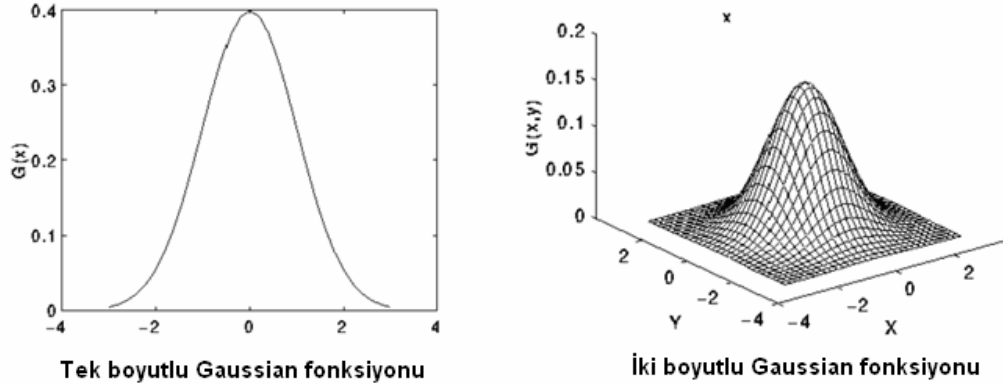
$$\text{Im}(s(x, y)) = \sin(2\pi(u_0x + v_0y) + P) \quad (1.42)$$

Kutupsal koordinatlarda karmaşık sinüzoid (1.43) eşitliği ile ifade edilebilir:

$$s(x, y) = \exp(j(2\pi F_0(x \cos \omega_0 + y \sin \omega_0) + P)) \quad (1.43)$$

1.2.4.2. Gabor Yapıtaşı: Gaussian Fonksiyonu

Gaussian fonksiyonu M ortalaması etrafında σ sapması ile rastgele dağılmış verilerin matematiksel ifadesidir. Şekil 1.12., orijin etrafında birim sapmaya sahip bir ve iki boyutlu Gaussian fonksiyonunu göstermektedir.



Şekil 1.12. Tek ve iki boyutlu Gaussian fonksiyonu

Tek boyutlu Gaussian fonksiyonu (1.44) eşitliği ile verilebilir.

$$\exp\left(-\frac{(x - x_0)^2}{2\sigma^2}\right) \quad (1.44)$$

Eğrinin altındaki alan (1.45) ile verildiğinden normalleştirilmiş tek-boyutlu Gaussian fonksiyonunu (1.46) eşitliği ile belirtebiliriz.

$$\int_{-\infty}^{\infty} \exp\left(-\frac{(x-x_0)^2}{2\sigma^2}\right) dx \quad (1.45)$$

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-x_0)^2}{2\sigma^2}\right) \quad (1.46)$$

Tek boyutlu Gaussian fonksiyonundaki argümanlar kullanılarak iki boyutlu Gaussian fonksiyonu kartezyen koordinatlarda yazılabilir (1.47).

$$f(x, y) = K \exp\left(-\pi\left(R_x(x-x_0)^2 + R_y(y-y_0)^2\right)\right) \quad (1.47)$$

İki boyutlu Gaussian fonksiyonu ifadesindeki (1.47) x ve y uzaydaki yatay ve düşey konumu, K Gaussian'ın tepe genliğini, x_0 ve y_0 yatay ve düşey eksenindeki orta noktayı, R_x ve R_y yatay ve düşey ölçekleme parametresini belirtmektedir.

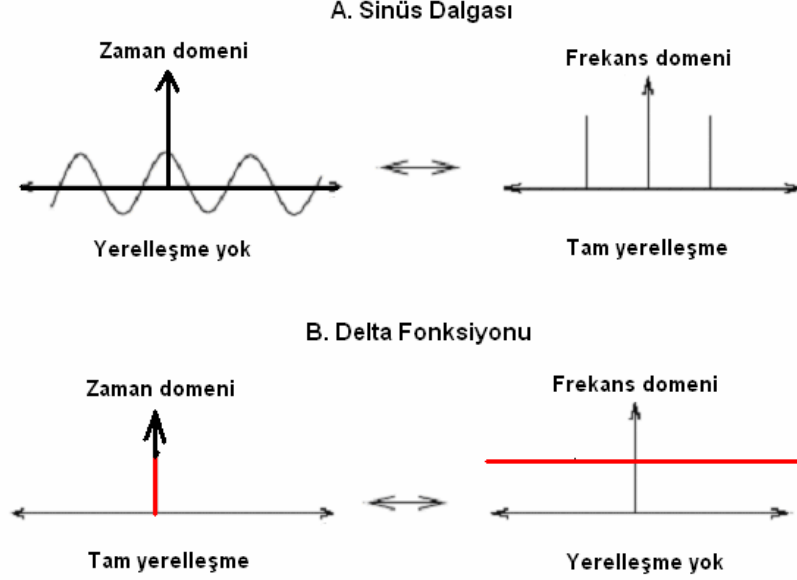
1.2.4.3. Karmaşık Gabor Fonksiyonunun Geliştirilmesi

1946'dan önce Fourier sistemi işaret işlemede güncel bir yaklaşımdı. Fourier sisteminin temeli rastgele işaretlerin Fourier serileri adı verilen trigonometrik fonksiyonlarla gösterimidir. Fourier analizi, matematik, mühendislik ve fizik gibi değişik alanlardaki uygulamalarda güçlü bir araçtır. Bununla beraber sınırlamaları vardır. Sadece mevcut frekansların global bilgisini ölçer. İşaretin hangi bitlerinin frekans tayfındaki bitleri ortaya çıkardığı hakkında bilgi vermez.

1946'da Dennis Gabor diğer bilim adamları gibi zaman/uzay ve frekans domenlerinin her ikisinde de yerelleştirmeyi sağlama problemi üzerinde çalışmıştır. Heisenberg'in belirsizlik ilkesini de içeren kuantum mekaniğindeki ilerlemeler ve Nyquist ve Hartley'in bir kanal üzerindeki bilginin iletimi için sınırlama sonuçları Gabor'u harekete geçirmiştir.

Yerelleştirmenin iki uç değerini yani zaman uzayı domeninde yerelleşmemiş fakat frekans domeninde son derece yerelleşmiş sinüs dalgası ile zaman domeninde tamamen

yerelleşmiş fakat frekans domeninde yerelleşmemiş delta fonksiyonunu incelemiştir (Şekil 1.13.).



Şekil 1.13. Frekans ve uzay sınırlamasında iki uçdeğer olan sinüs ve delta fonksiyonları

İşaretleri, zaman ve frekans domeninde tek bir bloğu veya ana işarete öteleme ve modülasyonla elde edilen temel fonksiyonları kullanarak yerelleştirmeyi ortaya atmıştır. Önerisi ile sonlu T süresine ve F band genişliğine sahip herhangi bir $f(t)$ işareti sonlu sayıdaki logon adı verilen temel bilgi hücrelerine bölünebilir oldu. Her bir logon Δt süresine ve ΔF band genişliğine sahiptir ve her bir domende değişik zaman ve frekanslarda yerleştirilmiştir.

Eğer $M = T/\Delta t$ ve $N = F/\Delta f$ alırsak MN temel bilgi hücresi oluşur ve işaret MN logon bilgiyi temsil eder. Bu, işaret tarafından gösterilebilen en fazla bilgidir ve bu MN karmaşık katsayıları işareti yeniden üretebilmek için yeterlidir [6].

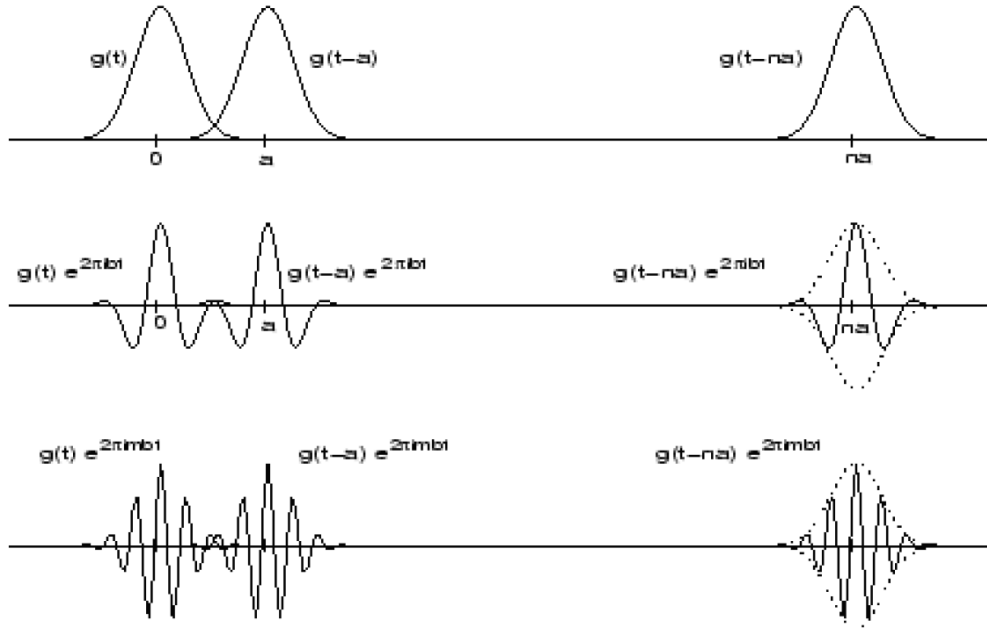
Sunduğu temel işaretler Gaussian fonksiyonu ve onun zamanda kaydırma parametresi (a) kullanarak ötelenmesi ve frekansta kaydırma parametresi (b) kullanarak modülasyonudur. Gabor, günümüzde Gabor fonksiyonu olarak adlandırılan bu temel işaretlerin zaman frekans düzleminde en küçük alanı işgal ettiğini ve böylece zaman ve frekans domenlerinde benzer bilgi miktarının en aza indirildiğini ispatlamıştır. Böylece Gabor temel fonksiyonu en küçük kuantum bilgisini gösterir.

Matematiksel olarak ifadesi (1.48)'de verilmektedir.

$$f(t) = \sum_{n,m \in \mathbb{Z}} c_{m,n} g_{m,n}(t) \quad (1.48)$$

$$g_{m,n}(t) = g(t - na) e^{2\pi i m b t}$$

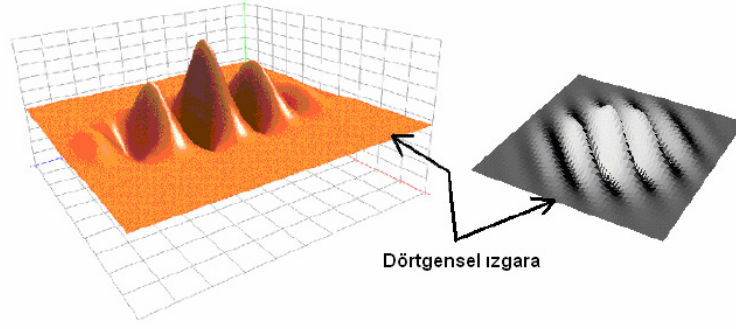
Gabor fonksiyonunun Gaussian'dan öteleme ve modülasyonla elde edilmesi Şekil 1.14.'te gösterilmektedir.



Şekil 1.14. Gabor temel fonksiyonunun öteleme ve modülasyonla oluşturulması

1.2.4.4. Karmaşık Gabor Fonksiyonu

Gabor fonksiyonu 2-D Gaussian zarf ile modüle edilmiş bir karmaşık sinüzoiddir. Şekil 1.15.'te gösterilen 2-D karmaşık Gabor fonksiyonunda da aynı mantık kullanılmaktadır.



Şekil 1.15. Karmaşık Gabor fonksiyonunun iki görünümü

Parametreler için genel notasyon kullanılarak 2-D Gabor fonksiyonu uzaysal domende (1.49) eşitliği ile ifade edilir.

$$g(x, y) = K \exp\left(-\pi\left(a^2(x - x_0)^2 + b^2(y - y_0)^2\right)\right) \exp\left(j(2\pi(u_0x + v_0y) + P)\right) \quad (1.49)$$

(1.49) eşitliğinin ilk kısmı Gaussian zarfını gösterirken ikinci kısım karmaşık sinüzoidi gösterir. İfade (1.50) ile yeniden yazılabilir.

$$g(x, y) = K \exp\left(-\pi\left(a^2(x - x_0)^2 + b^2(y - y_0)^2\right) + j(2\pi(u_0x + v_0y) + P)\right) \quad (1.50)$$

(1.50) ifadesinde değişkenler ayrılırsa (1.51) elde edilir.

$$g(x, y) = K \exp\left(-\pi\left(a^2(x - x_0)^2 + b^2(y - y_0)^2\right)\right) \left(\cos(2\pi(u_0x + v_0y) + P) + j \sin(2\pi(u_0x + v_0y) + P)\right) \quad (1.51)$$

Bu yüzden Gabor'un reel ve sanal kısmı (1.52) ve (1.53)'teki gibi oluşur.

$$\text{Re}(g(x, y)) = K \exp\left(-\pi\left(a^2(x - x_0)^2 + b^2(y - y_0)^2\right)\right) \cos(2\pi(u_0x + v_0y) + P) \quad (1.52)$$

$$\text{Im}(g(x, y)) = K \exp\left(-\pi\left(a^2(x - x_0)^2 + b^2(y - y_0)^2\right)\right) \sin(2\pi(u_0x + v_0y) + P) \quad (1.53)$$

Kutupsal koordinatlarda Gabor eşitliği (1.54) şeklini alır.

$$g(x, y) = K \exp\left(-\pi\left(a^2(x - x_0)^2 + b^2(y - y_0)^2\right)\right) \exp\left(j\left(2\pi F_0(x \cos \omega_0 + y \sin \omega_0) + P\right)\right) \quad (1.54)$$

Kutupsal koordinatlardaki Gabor ifadesi ortak paranteze alındığında (1.55) elde edilir.

$$g(x, y) = K \exp\left(-\pi\left(a^2(x - x_0)^2 + b^2(y - y_0)^2\right) + j\left(2\pi F_0(x \cos \omega_0 + y \sin \omega_0) + P\right)\right) \quad (1.55)$$

Ayrıca değişkenleri kutupsal koordinatlarda reel ve sanal olarak ayırırsak (1.56) ve (1.57) eşitlikleri ortaya çıkar.

$$\text{Re}(g(x, y)) = K \exp\left(-\pi\left(a^2(x - x_0)^2 + b^2(y - y_0)^2\right)\right) \cos\left(2\pi F_0(x \cos \omega_0 + y \sin \omega_0) + P\right) \quad (1.56)$$

$$\text{Im}(g(x, y)) = K \exp\left(-\pi\left(a^2(x - x_0)^2 + b^2(y - y_0)^2\right)\right) \sin\left(2\pi F_0(x \cos \omega_0 + y \sin \omega_0) + P\right) \quad (1.57)$$

Karmaşık sinüzoid, karmaşık Gabor fonksiyonunun karmaşık karakteristiğini oluşturur. Karmaşık Gabor fonksiyonu da karmaşık sinüzoidde olduğu gibi reel ve sanal kısımlara ayırt edilebilir. Birbilerine diktirler ve bu yüzden kareleştirme çifti adı verilir. Karmaşık Gabor fonksiyonunun reel kısmı ayrıca çift simetrik olduğundan çift Gabor fonksiyonu olarak da adlandırılır. Benzer şekilde sanal kısmı da tek Gabor fonksiyonu olarak adlandırılır.

1.2.4.5. Gabor Dalgacıklarının Üretimi

Dalgacıklar ana dalgacığın genişleme ve ötelemeleri ile üretilen temel fonksiyonlar ailesidir. Temel fonksiyonlar herhangi bir dalgacık ailesinin fonksiyonel temel inşa bloğudur.

Bu kavram Gabor ailesine genişletilmeden önce şimdiye kadar incelediğimiz Gabor fonksiyonunun, karmaşık sinüzoidle belirlenen sabit F_0 frekans genliğine ve sabit ω_0 oryantasyonuna sahip bir fonksiyon olduğuna dikkat edin. Ayrıca yarım tepe büyüklüğünde frekans ve oryantasyon band genişliği ($\Delta F_{1/2}$ ve $\Delta \omega_{1/2}$) ile de karakterize edilebilir.

Uygunluk açısından şimdiye kadar tartıştığımız karmaşık Gabor fonksiyonunu ana Gabor veya temel dalgacık olarak ele alalım. Bu temel dalgacığın frekans domeninde ölçeklenmesi ve uzaysal domende döndürülmesi ile Gabor dalgacıkları elde edilir.

Uzaysal domende döndürme, iki boyutlu uzaysal domene döndürülmüş koordinat sistemi kullanılarak koordinat dönüşümü uygulanmasıyla gerçekleştirilir.

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \quad (1.58)$$

Oryantasyon oranı ya da oryantasyon band genişliği $0 \leq \theta \leq \pi$ değerleri arasındadır. Pratik amaçlar için döndürme band genişliği N aralığa ayrılmıştır. Dolayısıyla her bir band genişliği (1.59) ile belirtilmiştir.

$$\theta_k = \frac{k\pi}{N}, \quad k = 1, 2, \dots, N \quad (1.59)$$

Gabor önermesi ile uyum için her bir döndürme veya oryantasyon için değerlerin birleşik frekans domen kümesi vardır. Bu, dalgacığı ölçekleyerek gerçekleştirilir. Fakat dalgacığı ölçeklemek merkez frekansı olan ayarlanmış frekansı değiştirir ve o da cevap olarak band genişliğini değiştirir. Örneğin ölçekleme çarpanı a 0.5 olduğunda her bir ardışık dalgacık bir öncekinin iki katına ayarlanır ve band genişliğinin iki katına sahip olur. Bu sebepten dolayı pratik düşünüldüğünde frekans domenindeki ölçekleme parametresi üsteldir. Bu yüzden ki, logaritmik frekans ölçeklemede her bir dalgacığın transfer fonksiyonu özdeştir. Böylece, Gabor dalgacığın tasarımında ortaya konulan logaritmik ölçekleme çarpanı her bir dalgacığın enerjisinin, ana fonksiyonun ölçeklemesinden bağımsız olduğunu garanti eder.

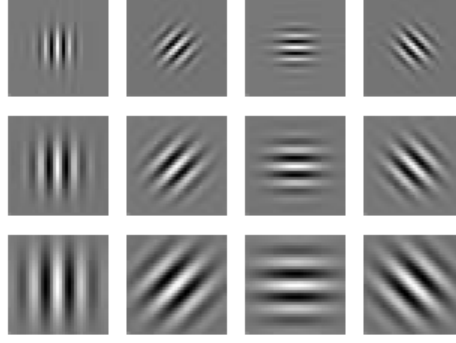
Ölçekleme, modüle eden Gaussian'ın standart sapması ve karmaşık sinüzoidin frekansı ile hazır hale getirilir.

Genişleme (ölçekleme) ve oryantasyon parametreleri tanımlarına sahip olduğumuzda Gabor dalgacık ailesi (1.60) ile ifade edilmektedir.

$$\left(g \left(a^j (x - x_0, y - y_0), \theta_k \right) \right), \quad a \in \mathbb{R}, j = \{0, -1, -2, \dots\} \quad (1.60)$$

m frekans ölçeği ve n oryantasyon ile ölçeklenmiş ana Gabor g için aynı türden filtre sözlüğü, üretme fonksiyonu ile elde edilebilir (1.61).

$$g_{mn}(x, y) = a^{-m} g(x', y'), \quad a > 1 \quad (1.61)$$



Şekil 1.16. Gabor dalgacıklarına örnekler

Fonksiyonel analiz domeninde üretilen Gabor dalgacıkları ortogonal olan temel küme oluşturmazlar. Dolayısıyla verilen Gabor dalgacık ailesi için Gabor dalgacığının görüntü üzerine sade izdüşümü ile ağırlığı hesaplamak mümkün olmamaktadır. İşareti açmaya ihtiyaç duyan ağırlıklar veya katsayıların belirlenmesi zor olduğundan, filtrelenmiş görüntülerde fazlalık bilgilerin olduğu görülür. Bu, işaret analizinde Gabor dalgacığını kullanmanın dezavantajıdır. Bununla birlikte Gabor dalgacıkları, görüntünün (veya işaretin) tam gösterimini biçimlendirebilir [7].

Gabor dalgacıkları, onların şu cazip özelliklerinden ötürü tercih edilirler.

1. Gabor fonksiyonları teorik olarak minimum uzay-frekans band genişliği sonucunu elde eder [8]. Bu, verilen bir band genişliği için uzaysal çözünürlüğün en yüksek değere çıkartılması anlamına gelir.
2. Bir dar-band Gabor fonksiyonu analitik bir fonksiyona yaklaşır. Analitik fonksiyonla katlanmış işaretler de analiktir ve uzaysal domende genlik ve faz karakteristiklerinin ayrı ayrı çözülmesine izin verirler.
3. Bir Gabor fonksiyonunun frekans domenindeki genlik tepkisi hiçbir yan taraf içermez.
4. Gabor fonksiyonlarının insanların görme sistemleri ile bir çok özelliği paylaştığı görülmektedir [9].

1.2.4.6. Gabor Filtre Sözlüğü Tasarımı

İki boyutlu bir Gabor fonksiyonu $g(x,y)$ ve onun Fourier dönüşümünü $G(u,v)$ tekrar ele alalım.

$$g(x, y) = \left(\frac{1}{2\pi\sigma_x\sigma_y} \right) \exp \left[-\frac{1}{2} \left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2} \right) + 2\pi j W x \right] \quad (1.62)$$

$$G(u, v) = \exp \left\{ -\frac{1}{2} \left[\frac{(u - W)^2}{\sigma_u^2} + \frac{v^2}{\sigma_v^2} \right] \right\}, \quad \sigma_u = 1/2\pi\sigma_x \text{ ve } \sigma_v = 1/2\pi\sigma_y \quad (1.63)$$

Gabor fonksiyonları tam fakat ortogonal olmayan temel kümeyi oluşturur. Bu temelleri kullanarak bir işareti açmak bize yerleşmiş frekans tanımlamasını sağlar. Gabor dalgacıkları diye adlandırılacak olan aynı türden fonksiyonlar sınıfı şimdi ele alınmıştır. $g(x,y)$ ana Gabor dalgacı olsun. Bu aynı türden filtre sözlüğü $g(x,y)$ nin (1.64) üretim fonksiyonu aracılığıyla uygun genişlemeleri ve dönmeleri sonucu elde edilebilir.

$$\begin{aligned} g_{mn}(x, y) &= a^{-m} g(x', y'), \quad a > 1, \\ x' &= a^{-m} (x \cos \theta + y \sin \theta), \quad y' = a^{-m} (-x \sin \theta + y \cos \theta), \quad \theta = n\pi / K \end{aligned} \quad (1.64)$$

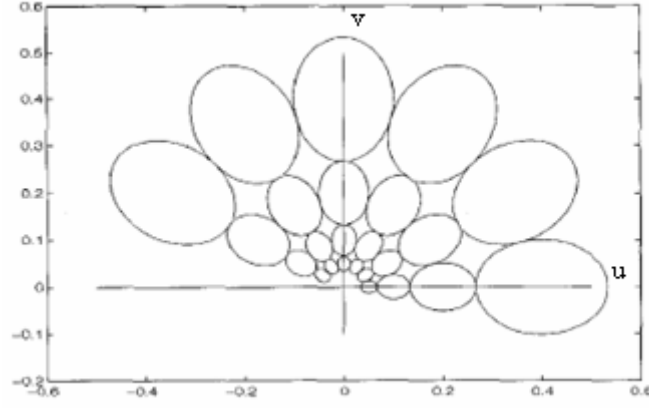
(1.64) ifadesinde K toplam oryantasyon sayısıdır. a^{-m} ölçekleme çarpanı enerjinin m den bağımsız olduğu anlamına gelir.

Gabor dalgacıklarının ortogonal olmayışı filtrelenmiş görüntülerde gereksiz bilgilerin bulunduğunu gösterir. Bu gereksiz bilgileri azaltmak için şu strateji geliştirilmiştir. U_l ve U_h ilgilenilen alt ve üst merkez frekanslarını ifade etsin. K , çoklu çözünürlük ayırtırmadaki oryantasyon sayısını ve S de ölçekleme sayısını versin. Tasarım stratejisi, yarım-tepe genliğinin frekans tayfındaki filtre tepkisinin Şekil 1.17.'de gösterildiği gibi birbirine değmesini sağlamanın garanti edilmesidir. Bu, σ_u ve σ_v (ve böylece σ_x ve σ_y) filtre parametrelerinin hesaplanması için (1.65)'te verilen formülleri meydana çıkarır.

$$a = (U_h/U_l)^{\frac{1}{s-1}}, \quad \sigma_u = \frac{(a-1)U_h}{(a+1)\sqrt{2\ln 2}},$$

$$\sigma_v = \tan\left(\frac{\pi}{2k}\right) \left[U_h - 2 \ln\left(\frac{2\sigma_u^2}{U_h}\right) \right] \left[2 \ln 2 - \frac{(2 \ln 2)^2 \sigma_u^2}{U_h^2} \right]^{-\frac{1}{2}} \quad (1.65)$$

(1.65)'te $W = U_h$ ve $M=0,1,\dots,S-1$ dir. Filtre tepkisinin mutlak parlaklık değerlerine duyarlılığını ortadan kaldırmak için 2D Gabor filtrelerinin reel (çift) bileşenleri onları sıfır ortalamalı yapmak amacıyla bir sabit eklenerek saptırılır (bu ayrıca $G(0,0)$ ın 0 yapılmasıyla da sağlanabilir).



Şekil 1.17. Gabor filtre sözlüğündeki filtre tepkelerinin yarı tepe genliğini gösteren çizgiler

$I(x,y)$ görüntüsü verildiğinde onun Gabor dalgacık dönüşümü (1.66) ile tanımlanabilir [10].

$$W_{mn}(x,y) = \int I(x_1,y_1) g_{mn} * (x-x_1, y-y_1) dx dy \quad (1.66)$$

1.2.4.7. Gabor Dalgacıklarının Biyolojik Önemi

1980'de Marcelja [9] görme bölgesinin alıcı bölge profili ve görme sisteminin basit kortikal hücrelerinin uzaysal frekans ayarlama karakteristikleri üzerinde çalışmış ve görüntünün görme bölgesindeki temsilinin uzay ve uzaysal frekans değişkenlerinin her

ikisini de içerdği sonucuna varmıştır. Görme bölgesinin alıcı bölgeleri ve uzaysal frekans ayarlama eğrileri Gabor temel işaretlerinin fonksiyonel yapısına yakından benzemektedir.

Marcelja'nın iddiası Daugman [11] ve Jones [12] tarafından desteklenmiştir. Daugman, görme bölgesindeki basit hücrelerin alıcı bölgelerinin uzaysal toplama özelliklerini modellemek için iki boyutlu Gabor fonksiyonlarını sunmuştur. Bu biyolojik buluşların harekete geçirmesi ile Daugman, Moshe Porat ve Yehoshua Y. Zeevi [13] Gabor fonksiyonlarını görüntü analizi ve görüntü sıkıştırma gibi görüntü işleme uygulamalarında kullanmayı önermişlerdir. O günden bu yana Gabor fonksiyonlarının doku analizi ve tıbbi işaret işleme gibi değişik bir çok uygulamada yararlı olduğu gösterilmiştir [7].

1.3. Öznitelik Çıkartımı

1.3.1. Temel Bileşen Analizi (PCA)

1.3.1.1. Giriş

PCA, bilgisayarla görme ve örüntü tanımda yaygın şekilde kullanılan klasik öznitelik çıkartma ve veri temsili tekniğidir [14]. Sirovich ve Kirby ilk defa PCA'yı insan yüzlerinin etkin temsili için kullanmışlardır [15]. Onlar, yüz resimlerinin yaklaşık olarak yeniden oluşturulabileceğini ispatlamışlardır. Özresimler yüz görüntülerini temsil etmede oldukça iyi olduğundan onlar üzerinden izdüşümlerin insan yüzlerini ayırt etmede sınıflandırma öznitelikleri olarak kullanılması düşünülmüştür. Bu amaçla, Turk ve Pentland 1991'de iyi bilinen özyüzler yöntemini ortaya atmışlardır [16]. Bu yöntemde özyüzler, yüz kovaryans matrisinin baskın özdeğerleriyle ilişkili özvektörlere karşı düşmektedir. Özyüzler orjinal uzayı güçlü bir şekilde indirgeyen bir öznitelik uzayı tanımlar ve yüz tanıma bu indirgenmiş uzayda gerçekleştirilir. Daha sonraları, PCA geniş şekilde incelenmiş ve yüz tanımda en başarılı yöntemlerden biri olmuştur. Penev ve Sirovich gösterim için özyüzlerin kullanılması halinde yüz uzayının boyutluluğu problemini tartışmışlardır [17]. Zhao ve Yang, PCA tabanlı görme sistemlerinde keyfi aydınlanma efektlerinin etkilerini açıklamaya çalışmışlardır [18]. Bu amaçla özel aydınlanma koşulundaki kovaryans matrisinin formülünün kapalı biçimi analitik olarak, keyfi aydınlanma koşulundaki eşitlik de aydınlanma eşitliği ile oluşturulmuştur. Bununla beraber Wiskott, Fellous, Krüger ve von der Malsburg şuna dikkati çekmiştir ki, eğitim verisinde açıkça belirtilmedikçe PCA en basit bir değişimden bağımsızlığı bile desteklememektedir [19]. Bu yüzden PCA'nın

zayıflığının üstesinden gelmek için elastik grup graf eşleme diye bilinen tekniği sunmuşlardır.

1.3.1.2. PCA'nın Matematiksel Arka Planı

Bu bölüm PCA işlemini anlamak için gerekli bazı matematiksel temelleri vermeye çalışacaktır.

$d \times d$ boyutlarında M matrisi verilsin. Önemli bir eşitlik sınıfı (1.67)'de belirtilmiştir.

$$Mx = \lambda x \quad (1.67)$$

(1.67) eşitliğinde λ sayısı için eşitlik (1.68)'de verildiği şekilde yazılabilir.

$$(M - \lambda I)x = 0 \quad (1.68)$$

(1.67) ve (1.68) eşitliklerinde I birim matris, 0 da sıfır vektörüdür. Çözüm vektörü $x=e_i$ ve karşı düşen $\lambda=\lambda_i$ sayısı özvektör ve ilişkili özdeğeri olarak adlandırılır. Eğer M reel ve simetrik ise, muhtemelen ayrı olmayan d adet çözüm vektörü $\{e_1, e_2, \dots, e_d\}$ ve bunlarla ilişkili $\{\lambda_1, \lambda_2, \dots, \lambda_d\}$ özdeğerleri vardır. M ile çarpıldığında özvektörler sadece genlikte değişim geçirir (yönde değil).

$$Me_j = \lambda_j e_j \quad (1.69)$$

Eğer M köşegen matrisse özvektörler koordinat eksenine paraleldir.

Özdeğerleri bulmanın bir yolu karakteristik eşitliği her bir d adet kök için (λ_j) çözmektir:

$$|M - \lambda I| = \lambda^d + a_1 \lambda^{d-1} + \dots + a_{d-1} \lambda + a_d = 0 \quad (1.70)$$

Sonra, her bir kök için ilişkili özvektörü (e_j) bulmak amacıyla bir grup lineer eşitlik çözülür.

Sonunda matrisin izinin özdeğerlerin toplamı ve determinantının da özdeğerlerin çarpımı olduğu görülebilir.

$$\text{iz}[M] = \sum_{i=1}^d \lambda_i \quad \text{ve} \quad |M| = \prod_{i=1}^d \lambda_i \quad (1.71)$$

Eğer matris köşegen matrisse, özdeğerleri köşegendeki 0 olmayan değerler ve özvektörleri de koordinat eksenlerine paralel birim vektörlerdir.

Bir vektörün beklenen değeri, orjinal bileşenlerin beklenen değerlerinden oluşan vektör olarak tanımlanır. Böylece, eğer $f(x)$ n-boyutlu ve d-boyutlu rastgele x vektörünün (1.72)'de verildiği gibi vektörel fonksiyonu ise f 'nin beklenen değeri (1.73)'teki gibi tanımlanır.

$$f(x) = \begin{bmatrix} f_1(x) \\ f_2(x) \\ \vdots \\ f_n(x) \end{bmatrix} \quad (1.72)$$

$$E[f] = \begin{bmatrix} E[f_1(x)] \\ E[f_2(x)] \\ \vdots \\ E[f_n(x)] \end{bmatrix} = \sum_x f(x)P(x) \quad (1.73)$$

Özellikle, d-boyutlu ortalama vektörü μ (1.74)'te verildiği üzere tanımlanır.

$$\mu = E[x] = \begin{bmatrix} E[x_1] \\ E[x_2] \\ \vdots \\ E[x_d] \end{bmatrix} = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_d \end{bmatrix} = \sum_x xP(x) \quad (1.74)$$

Benzer şekilde Σ kovaryans matrisi, ij. elemanı σ_{ij} , x_i ve x_j nin kovaryansı olan kare matristir.

$$\sigma_{ij} = \sigma_{ji} = E[(x_i - \mu_i)(x_j - \mu_j)] \quad i, j = 1 \dots d \quad (1.75)$$

$(x-\mu)(x-\mu)^T$ vektör çarpımını, kovaryans matrisini (1.76) eşitliği şeklinde yazmak için kullanabiliriz:

$$\Sigma = E[(x - \mu)(x - \mu)^T] \quad (1.76)$$

Böylece, Σ hem simetrik hem de köşegen elemanları, her bir farklı x elemanının varyansı olan (negatif olmayan) bir matris olur. Köşegen dışındaki elemanları ise pozitif veya negatif olabilen kovaryans değerleridir. Eğer varyanslar istatistiksel olarak bağımsız iseler kovaryansları 0'dır ve kovaryans matrisi köşegenseldir. Cauchy-Schwarz eşitsizliğine göre eğer w herhangi bir d -boyutlu vektörse, $w^T x$ in varyansı negatif olamaz. Bu, $w^T \Sigma x$ karesel biçiminin negatif olamamasına sebep olur. Bu şekildeki matrislere yarı kesin pozitif ismi verilir; böylece, Σ kovaryans matrisi de yarı kesin pozitif olmak zorunda olur. Bu da Σ kovaryans matrisinin özdeğerlerinin hiç birinin negatif olmaması gerektiğini gösterir.

PCA'yı matematiksel olarak analiz etme işine, n adet d -boyutlu örnek kümesindeki $(x_1, \dots, x_n)$ vektörlerin hepsinin tek bir x_0 vektörü ile temsili probleminin ele alınmasıyla başlanılabilir. Daha açık olarak, x_0 ile çeşitli x_k lar arasındaki uzaklıkların karelerinin toplamı mümkün olduğunca küçük olan bir x_0 vektörü bulmak istediğimizi varsayalım. Hata ölçütü fonksiyonu $J_0(x_0)$ 'ı (1.77)'deki eşitlikle tanımlayarak J_0 'ı en küçük değerine ulaştıran x_0 'ı ararız.

$$J_0(x_0) = \sum_{k=1}^n \|x_0 - x_k\|^2 \quad (1.77)$$

Bu problemin çözümünün $x_0 = m$ ile verildiği aşıkardır (burada m ortalama vektördür).

$$m = \frac{1}{n} \sum_{k=1}^n x_k \quad (1.78)$$

Bu, kolaylıkla (1.79)'da gösterildiği şekilde doğrulanabilir.

$$\begin{aligned}
J_0(x_0) &= \sum_{k=1}^n \|x_0 - m\|^2 - 2 \sum_{k=1}^n (x_0 - m)^T (x_k - m) + \sum_{k=1}^n \|x_k - m\|^2 \\
&= \sum_{k=1}^n \|x_0 - m\|^2 - 2(x_0 - m)^T \sum_{k=1}^n (x_k - m) + \sum_{k=1}^n \|x_k - m\|^2 \\
&= \sum_{k=1}^n \|x_0 - m\|^2 + \underbrace{\sum_{k=1}^n \|x_k - m\|^2}_{x_0 \text{ 'dan bağımsız}}
\end{aligned} \tag{1.79}$$

İkinci toplam, x_0 'dan bağımsız olduğu için bu ifadenin $x_0=m$ seçimi ile minimize edilebilirliği açıkça gözükmektedir.

Örneklerin ortalama vektörleri veri kümesinin sıfır boyutlu gösterimidir. Bu basittir, ama verideki değişkenliği ortaya çıkaramamaktadır. Veriyi, örneğin ortalaması etrafında koşan satır üzerine izdüşürerek daha ilginç ve tek boyutlu gösterim elde edebiliriz. e , satırın yönündeki birim vektör olsun. Bu durumda satırın eşitliği $x = m + ae$ şeklinde yazılabilir. Burada a sayısı, herhangi bir x noktası ile m ortalaması arasındaki uzaklığa karşı düşmektedir. Eğer x_k 'yı $m+ae$ ile gösterirsek hata ölçütü fonksiyonunu minimize ederek a_k nın en uygun katsayı grubunu bulabiliriz.

$$\begin{aligned}
J(a_1, \dots, a_n, e) &= \sum_{k=1}^n \|(m + a_k e) - x_k\|^2 \\
&= \sum_{k=1}^n \|a_k e - (x_k - m)\|^2 \\
&= \sum_{k=1}^n a_k \|e\|^2 - 2 \sum_{k=1}^n a_k e^T (x_k - m) + \sum_{k=1}^n \|x_k - m\|^2
\end{aligned} \tag{1.80}$$

$\|e\|=1$ kabul ederek, kısmen a_k 'ya göre ayırt ederek ve türevi 0'a ayarlayarak (1.81) sonucunu elde ederiz.

$$a_k = e^T (x_k - m) \tag{1.81}$$

Geometrik olarak bu sonuç sadece, x_k vektörünü örnek ortalaması civarında seyreden e 'nin yönündeki satıra izdüşürerek en küçük kareler yöntemini elde edebileceğimizi söyler. Bu bize daha ilginç bir problemi getirir: satır için en iyi e yönünü bulma. Bu problemin çözümü sözde dağılım matrisini (S_t) içerir.

$$S_t = \sum_{k=1}^n (x_k - m)(x_k - m)^T \tag{1.82}$$

Dağılım matrisi şöyle gözükmelidir: Sadece $n-1$ kere örnek kovaryans matrisi. (1.81)'de bulduğumuz a_k 'yı (1.80)'de yerine koyduğumuzda (1.83) elde edilir.

$$\begin{aligned}
 J_1(e) &= \sum_{k=1}^n a_k^2 - 2 \sum_{k=1}^n a_k^2 + \sum_{k=1}^n \|x_k - m\|^2 \\
 &= - \sum_{k=1}^n [e^T (x_k - m)]^2 + \sum_{k=1}^n \|x_k - m\|^2 \\
 &= - \sum_{k=1}^n e^T (x_k - m) (x_k - m)^T e + \sum_{k=1}^n \|x_k - m\|^2 \\
 &= -e^T S_t e + \sum_{k=1}^n \|x_k - m\|^2
 \end{aligned} \tag{1.83}$$

Daha açıkcası, J_1 'i minimize eden e vektörü ayrıca $e^T S_t e$ yi de maksimize eder. $e^T S_t e$ öznesini $\|e\|=1$ kısıtlaması altında maksimize etmek için Lagrange çarpıcılar metodu kullanıldı. λ belirsiz çarpıcı olsun. Bu durumda (1.84) eşitliği yazılabilir.

$$u = e^T S_t e - \lambda (e^T e - 1) \tag{1.84}$$

(1.84) eşitliğinin e 'ye göre türevi alınırsa (1.85) bulunur.

$$\frac{\partial u}{\partial e} = 2S_t e - 2\lambda e \tag{1.85}$$

Bu gradyant vektörünü 0'a eşitlersek e 'nin, dağılım matrisinin özvektörü olması gerektiğini görürüz.

$$S_t e = \lambda e \tag{1.86}$$

Özellikle, $e^T S_t e = \lambda e^T e = \lambda$ olduğundan $e^T S_t e$ yi maksimize edebilmek için dağılım matrisinin en büyük özdeğerine karşı düşen özvektörü seçmemiz gerekir. Diğer bir deyişle, en iyi tek boyutlu izdüşümü bulmak için dağılım matrisinin en büyük özdeğerine sahip özvektörünün yönünde ve örnek ortalaması civarında bir satıra verimizi izdüşürmemiz gerekecektir.

Bu sonuç kolayca tek boyutlu izdüşümden d ' boyutlu izdüşüme genişletilebilir. Eşitlik (1.79)'da (1.87) yerine yazılırsa (1.88) ölçüt fonksiyonu eşitliği elde edilir.

$$x = m + \sum_{i=1}^{d'} a_i e_i \quad (1.87)$$

$$J_{d'} = \sum_{k=1}^n \left\| \left(m + \sum_{i=1}^{d'} a_{k_i} e_i \right) - x_k \right\|^2 \quad (1.88)$$

(1.88) eşitliğindeki ölçüt fonksiyonu değeri $e_1, e_2, \dots, e_{d'}$ vektörleri, en büyük özdeğerlere sahip özvektörlere eşit olduğunda minimize edilmiş olmaktadır. Çünkü dağılım matrisi reel ve simetriktir, bu özvektörler ortogonaldır. Bunlar, herhangi bir x öznitelik vektörünün gösterimi için temel vektörlerin doğal kümesidir. (1.87)'deki a_i katsayıları x 'in bu temeldeki bileşenleridir ve temel bileşenler olarak adlandırılırlar. Geometrik olarak, eğer d boyutlu, hiperelipsoidle bulutların $x_1, \dots, x_n$ veri noktaları resmedilirse, dağılım matrisinin özvektörleri bulutların dağılımının en fazla olduğu yönlerde dikkati çeken öznitelik uzayının boyutları olacaktır.

1.3.1.3. Temel Bileşen Analizinin Uygulanması

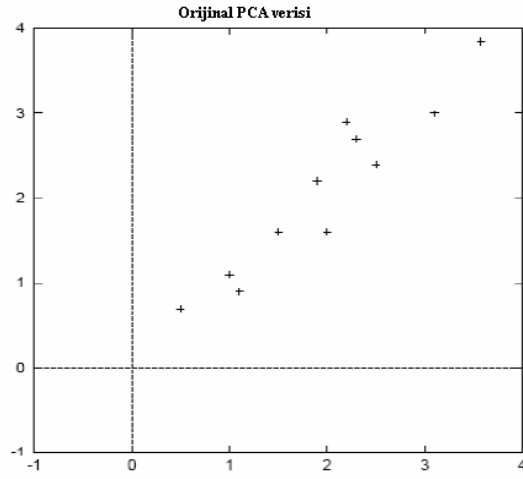
Önceden de belirtildiği üzere PCA, verideki örüntüleri tanımlamada ve verilerin benzerliklerinin ve farklılıklarının altının çizilmesinde bir yoldur. Veride bu örüntüleri bulduğunuzda, veriyi sıkıştırmak mümkün olur (boyut azaltımı gerçekleyerek – çok fazla bilgi kaybı olmadan). Bu teknik ilerde göreceğimiz gibi veri sıkıştırmada da kullanılmaktadır. Bu bölüm, bir grup veri üzerinde PCA'nın gerçekleştirilme adımlarını gösterecektir. Tekniğin tam olarak niçin çalıştığını belirtmeyeceğiz, ama her bir noktada nelerin olduğunu açıklamaya çalışacağız.

Yöntemin işlem adımları şöyledir:

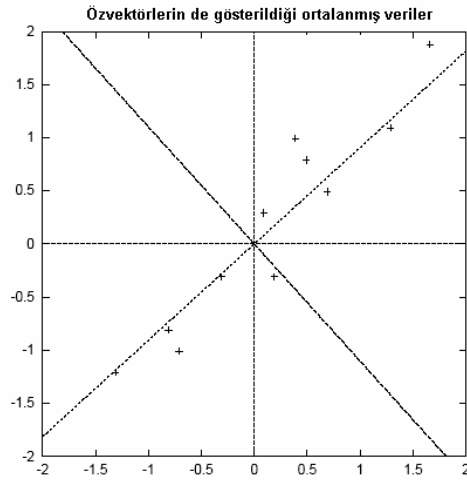
- Adım 1: Verileri al. Bu basit örnekte, daha önceden var olan bir grup veriyi kullanacağız. Veriler sadece iki boyuta sahip, bu sayede PCA'nın her bir adımda ne yaptığını göstermek için verinin çizimi gerçekleştirilebilecektir.
- Adım 2: Ortalamayı çıkart. Bütün x değerlerinden $\bar{x}$ (bütün veri noktalarının x değerlerinin ortalaması), bütün y değerlerinden de $\bar{y}$ değerini çıkart. Bu, ortalaması 0 olan bir küme oluşturmamızı sağlar.
- Adım 3: Kovaryans matrisini hesapla. Data matrisi, kovaryans matrisini oluşturmak için kendi transpozesiyle çarpılır.

Tablo 1.2. PCA için örnek veri kümesi ve ayarlanması (a) orijinal küme (b) ayarlanmış küme

(a)		(b)	
x	y	x	y
2.5	2.4	0.69	0.49
0.5	0.7	-1.31	-1.21
2.2	2.9	0.39	0.99
1.9	2.2	0.09	0.29
3.1	3.0	1.29	1.09
2.3	2.7	0.49	0.79
2.0	1.6	0.19	-0.31
1.0	1.1	-0.81	-0.81
1.5	1.6	-0.31	-0.31
1.1	0.9	-0.71	-1.01



Şekil 1.18. Örnek PCA verilerinin 2 boyutlu uzaydaki gösterimi



Şekil 1.19. Ortalamanın çıkartıldığı normalleştirilmiş veriler ve kovaryans matrisinin özvektörleri

Örneğe göre sonuç aşağıdaki gibidir.

$$\text{cov} = \begin{pmatrix} 0.616555556 & 0.615444444 \\ 0.615444444 & 0.716555556 \end{pmatrix}$$

- Adım 4: Kovaryans matrisin özdeğerlerini ve özvektörlerini hesapla. Özdeğerler ve onlara karşı düşen özvektörleri, kovaryans matrisinden hesaplanır $\Omega V = \Lambda V$. Burada, V , Λ özdeğeriyle ilişkilendirilmiş özvektörlerin kümesidir. Özdeğerler ve özvektörler örnek için şunlardır:

$$\begin{aligned} \text{özdeğerler} &= \begin{pmatrix} 0.0490833989 \\ 1.28402771 \end{pmatrix} \\ \text{özvektörler} &= \begin{pmatrix} -0.735178656 & -0.677873399 \\ 0.677873399 & -0.735178656 \end{pmatrix} \end{aligned}$$

Bu kovaryans matrisin köşegenin dışındaki elemanları pozitif olduğundan x ve y değişkenlerinin ikisinin de beraber artması beklenir.

Şunu da gözden kaçırmamak gerekir: Buradaki özvektörlerin hepsi birim özvektörlerdir (uzunlukları 1 birim). Bu, PCA için çok önemlidir. Bir çok matematik paketi özvektör hesabı istendiğinde birim özvektörleri vermektedir. Şekil 1.19.'da çizilmiş veri oldukça güçlü bir örüntüye sahiptir. Kovaryans matristen beklendiği gibi, aslında iki değişken de beraber artar. Verinin başında, ayrıca özvektörler de çizilmiştir. Özvektörler şekilde çapraz noktalı çizgiler olarak gözükmemektedirler. Özvektör bölümünde ifade edildiği gibi, her biri birbirine diktir, fakat daha önemlisi verideki örüntüler hakkında bize bilgi vermektedirler. Özvektörlerin noktaların ortasından nasıl geçtiğini görün (sanki en iyi doğru uydurma gibi). İlk özvektör bize bu iki veri kümesinin o çizgi boyunca nasıl ilişkilendirildiğini gösterir. İkinci özvektör de verideki diğer (daha az önemli olan) örüntüyü verir.

Böylece, bu kovaryans matrisin özvektörlerini bulma işlemi ile veriyi karakterize eden doğrular bulunmuş oluyor. Geri kalan adımlar veriyi dönüştürmeyi içerir. Özvektör ve özdeğerlerin hesaplanmasında yaygın olarak kullanılan teknikler SVD, QR ve QL ayrıştırmasıdır. Bu yöntemler hakkında ayrıntılı bilgi sayısal çözümleme kaynaklarında mevcuttur [20].

- Adım 5: Bileşenleri seç ve öznitelik vektörü oluştur. Bu bölüm, veri sıkıştırma ve indirgenmiş boyutların, işlemin içine girdiği bölümdür. Önceki bölümden özdeğerlere ve özvektörlere bakarsanız, özdeğerlerin oldukça farklı olduğunu göreceksiniz. Aslında, en büyük değere sahip özvektör veri kümesinin temel bileşenidir. Örneğimizde, en büyük değerlikli özdeğere sahip özvektör veri boyutları arasındaki en önemli ilişkiyi işaret etmektedir.

Genelde, kovaryans bulunduğunda, sonraki adım onları özdeğerlerine göre büyükten küçüğe dizmektir. Bu, bileşenleri önemlerine göre vermeyi sağlar. Şimdi, eğer isterseniz düşük öncelikli bileşenleri gözardı edebilirsiniz. Böylece bazı bilgilerin kaybolmasına sebep olabilirsiniz, ama özdeğerler küçükse çok fazla bilgi kaybetmemiş olursunuz. Bazı bileşenleri atlarsanız, sonuç veri kümesi orjinalinden daha küçük olacaktır. Daha açıkçası, orjinalde verinizde n boyut varsa, ve n adet özdeğer ve özvektör hesaplamış ve p özvektör seçmişseniz, sonuç veri kümeniz sadece p boyuta sahip olacaktır. Şimdi, vektörlerin matrisi olarak bilinen bir öznitelik vektörü oluşturmalsınız. Bu, saklamak istediğiniz özvektörleri alarak ve bu vektörleri sütunlara koyarak kurulur.

$$\text{ÖznitelikVektörü} = (\text{öz}v_1 \quad \text{öz}v_2 \quad \text{öz}v_3 \quad \cdots \quad \text{öz}v_n)$$

Örnek veri grubumuz verildiğinde ve dolayısıyla iki özvektörümüz olduğunda iki seçeneğimiz vardır: Bu iki vektörün her ikisi ile bir öznitelik vektörü oluşturmak:

$$\begin{pmatrix} -0.677873399 & -0.735178656 \\ -0.735178656 & 0.677873399 \end{pmatrix}$$

ya da daha az anlama sahip olanı kaldırmak:

$$\begin{pmatrix} -0.677873399 \\ -0.735178656 \end{pmatrix}$$

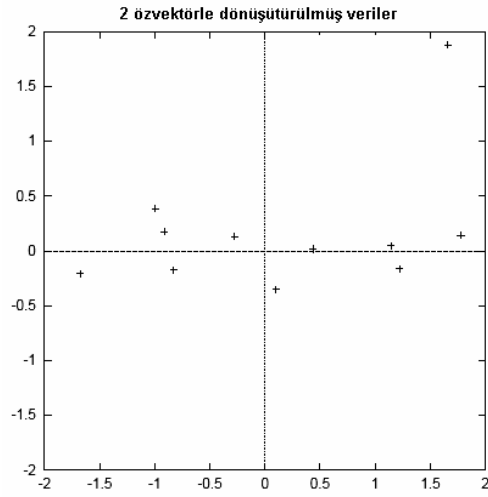
- Adım 6: Yeni veri kümesi al. Bu, PCA'daki son adımdır ve en basitidir. Bileşenleri seçtikten ve öznitelik vektörünü oluşturduktan sonra, vektörün transpozisini alıyor ve transpoze olmuş orjinal veri kümesinin kalanıyla çarpıyoruz.

$$\text{Sonuç Verisi} = \text{SatırÖznitelikVektörü} \times \text{DüzenlenmişSatırVerisi}$$

SatırÖznitelikVektörü sütunların satıra dönüştürüldüğü özvektör matrisinin transpozudur (en anlamlı özvektör en yukarıda). DüzenlenmişSatırVerisi, ortalamaya göre düzenlenmiş transpoze veridir (yani her bir satırda ayrı boyut, her bir sütunda da veri öğeleri olan). Sonuç Verisi, sütunlarda veri öğeleri, satırlarda boyutlar olan sonuç veri kümesidir.

Tablo 1.3. Örnek için dönüştürülmüş veriler

x	y
-0.827970186	-0.175115307
1.77758033	0.142857227
-0.992197494	0.384374989
-0.274210416	0.130417207
-1.67580142	-0.209498461
-0.912949103	0.175282444
0.0991094375	-0.349824698
1.14457216	0.0464172582
0.438046137	0.0177646297
1.22382056	-0.162675287



Şekil 1.20. Verilerin, iki özvektör kullanılarak PCA ile elde edilen öz uzaydaki görüntüleri

Tablo 1.4. Örnek için sadece en anlamlı özvektörün kullanılmasıyla dönüştürülmüş veriler

x
-0.827970186
1.77758033
-0.992197494
-0.274210416
-1.67580142
-0.912949103
0.0991094375
1.14457216
0.438046137
1.22382056

Bu bize sadece seçtiğimiz vektörlere göre dizilmiş orjinal veriyi verecektir. Orjinal veri kümemiz 2 eksene (x ve y) sahipti, bu yüzden verimiz de onlara göre idi. Veriyi istediğiniz iki eksenle ifade etmek mümkündür. Eğer bu eksenler dikey iseler, ifade en etkin biçimdedir. Bu, özvektörlerin her zaman birbirlerine göre dik olmalarının neden gerektiğidir. Verimizi x ve y eksenlerine göre olmaktan iki özvektörümüze göre olmaya çevirdik. Yeni veri kümemiz indirgenmiş boyuta sahip olduğunda, yeni verimiz sadece saklamayı düşündüğümüz vektörlere göre dir.

Bunu verimizde göstermek için, son dönüşümü mümkün her bir öznitelik vektörüne uyguladık. Sonucu tablo biçiminde göstermek için transpozisini aldık. Ayrıca bileşenlerle nasıl bağlantı kurduklarını göstermek için son noktaları çizdik.

Eğer dönüşüm için her iki özvektörü de tutarsak veriyi Şekil 1.20.'de çizildiği gibi elde ederiz. Bu çizim aslında özvektörlerin eksenler olacak şekilde döndürüldüğü orjinal veridir. Böylece bu bölümlemede hiç bir bilgi kaybı olmamış olur.

Yapılabilecek diğer bir dönüşüm en büyük özdeğerlere karşı düşen özvektörleri almaktır. Bu amaçla özdeğerler büyükten küçüğe doğru sıralanır ve genlikleri toplamı bütün özdeğerlerin genlikleri toplamının %95'ini (bu değer çoğu uygulama için uygun bir değerdir) bulan en az sayıda özdeğere karşı düşen özvektörler seçilir. Bundan çıkan sonuç Tablo 1.4.'te gösterilmiştir. Beklendiği gibi sadece bir boyutu vardır. Eğer bu veriyi her iki özvektörü kullananla karşılaştırsak bu veri kümesinin diğerinin ilk sütunu olduğu gözükecektir. Bu yüzden, eğer bu veriyi çizerseniz tek boyutlu olacaktır ve çizgi üzerindeki noktalar olacaktır. Etkin bir şekilde diğer özvektöre karşı düşen ekseni atmış bulunuyoruz.

Temelde, verimizi aralarındaki örüntüye (örüntüler, veri arasındaki ilişkiyi en iyi şekilde tanımlayan çizgilerdir) göre ifade edilecek şekilde dönüştürmüş oluyoruz. Bu faydalıdır, çünkü veri noktalarımız her bir satırdaki katkıların kombinasyonu olarak sınıflandırılmıştır. Başlangıçta x ve y eksenlerine sahiptik. Bu bir derece iyidir, ama her bir noktanın x ve y değerleri o noktanın diğer verilerle olan ilişkisini tam olarak belirtememektedir. Şimdi, veri noktalarının değerleri tam olarak verinin eğilimini vermektedir. Her iki özvektörün kullanıldığı dönüşüm durumunda genel eksenler yerine özvektörlere göre veri değiştirilmektedir. Fakat, tek özvektör parçalaması küçük özvektörün katkısını kaldırmakta ve sadece diğerine bağlı bir veri vermektedir.

PCA danışmansız öğrenme yöntemidir ve çıkış bilgisini kullanmaz. Bununla beraber sınıflandırmada çok sayıda sınıf vardır. Karhunen-Loeve açılımı sınıf bilgisinin

kullanımına izin verir. Bütün örneklerin kovaryans matrisini kullanmak yerine ortalamalarını kovaryans matrisi olarak ele alır ve onun özvektörlerini kullanır [21].

1.3.1.4. Öz Uzak İzdüşümü İçin Enstantane Metodu

Yukarıda anlatılan yöntem, çok geniş kovaryans matrisleri oluşturabilir. Örneğin 64x64 büyüklüğündeki görüntüler, 4096xp büyüklüğündeki bir veri matrisi ve 4096x4096 büyüklüğünde bir kovaryans matrisi oluşturulmasına neden olur. Kovaryans matrisini ve kovaryansın özvektörleri/özdeğerlerini hesaplamak bir problem oluşturur. NxM matris için, matrisin sahip olabileceği 0 olmayan özvektörlerin maksimum sayısı $\min(N-1, M-1)$ olduğu bilinir. Eğitim görüntülerinin sayısı (p) genellikle piksel sayısından (N) küçük olduğu için bulunabilecek en büyük özvektörler/özdeğerler P-1'dir.

Lineer cebirde yaygın olarak kullanılan bir teorem $\overline{XX}^T$ ve $\overline{X}^T \overline{X}$ in özdeğerlerinin aynı olduğunu ifade eder. Üstelik, $\overline{XX}^T$ nin özvektörleri, $\overline{X}$ matrisiyle çarpılan ve normalize edilen $\overline{X}^T \overline{X}$ in özvektörleri ile aynıdır. Bu teoremi kullanarak, NxN yerine PxP lik bir öz uzayı oluşturmak için enstantane metodu kullanılabilir. Bunun için aşağıdaki adımlar izlenir:

1 ve 2 (veriyi ortalama ve veri matrisinin oluşturulması) orjinal metodla aynıdır.

3. Kovaryans matrisi oluşturma: Bir kovaryans matrisi oluşturmak için veri matrisinin transpozesi data matrisiyle çarpılır.

$$\Omega' = \overline{X}^T \overline{X} \quad (1.89)$$

4. O' nun özdeğerleri ve özvektörlerini hesaplama: Özdeğerleri ve karşı düşen özvektörleri Ω' dan hesaplanır.

5. Veri matrisi özvektörlerle çarpılır. Özvektörleri normlarına bölünür.

6. Özdeğerler sıralanır.

Yöntemi [22] önceki örnek üzerinde inceleyebiliriz.

İlk iki adım aynı olduğu için sonuçlar da aynı olacaktır. Kovaryans matrisini yeniden oluşturursak:

$$\text{cov}' = \begin{pmatrix} 0.716 & -1.496 & 0.754 & 0.204 & 1.424 & 0.725 & -0.020 & -0.955 & -0.365 & -0.984 \\ -1.496 & 3.180 & -1.708 & -0.468 & -3.008 & -1.597 & 0.126 & 2.041 & 0.781 & 2.152 \\ 0.754 & -1.708 & 1.132 & 0.322 & 1.582 & 0.973 & -0.232 & -1.117 & -0.427 & -1.276 \\ 0.204 & -0.468 & 0.322 & 0.092 & 0.432 & 0.273 & -0.072 & -0.307 & -0.117 & -0.356 \\ 1.424 & -3.008 & 1.582 & 0.432 & 2.852 & 1.493 & -0.092 & -1.927 & -0.737 & -2.016 \\ 0.725 & -1.597 & 0.973 & 0.273 & 1.493 & 0.864 & -0.151 & -1.036 & -0.396 & -1.145 \\ -0.020 & 0.126 & -0.232 & -0.072 & -0.092 & -0.151 & 0.132 & 0.097 & 0.037 & 0.178 \\ -0.955 & 2.041 & -1.117 & -0.307 & -1.927 & -1.036 & 0.097 & 1.312 & 0.502 & 1.393 \\ -0.365 & 0.781 & -0.427 & -0.117 & -0.737 & -0.396 & 0.037 & 0.502 & 0.192 & 0.533 \\ -0.984 & 2.152 & -1.276 & -0.356 & -2.016 & -1.145 & 0.178 & 1.393 & 0.533 & 1.524 \end{pmatrix}$$

Bu matrisin 0'dan büyük özdeğerleri ve özvektörleri de şu şekilde oluşmaktadır:

$$\begin{aligned} \text{özdeğerler} &= \begin{pmatrix} 11.55624 \\ 0.44175 \end{pmatrix} \\ \text{özvektörler} &= \begin{pmatrix} -0.24356 & 0.26347 \\ 0.52290 & -0.21493 \\ -0.29187 & -0.57831 \\ -0.08066 & -0.19622 \\ -0.49296 & 0.31520 \\ -0.26855 & -0.26372 \\ 0.02915 & 0.52633 \\ 0.33669 & -0.06983 \\ 0.12885 & -0.02672 \\ 0.36000 & 0.24475 \end{pmatrix} \end{aligned}$$

Burada elde ettiğimiz özvektörleri veri matrisimizle çarptığımız zaman diğer yöntemle elde ettiğimiz özvektörlere ulaşırız. Dolayısıyla sonraki adımlar da aynen devam eder.

1.3.2. Çekirdek Öznitelik Uzayı

Doğrusal öğrenme makinelerinin sınırlı hesaplama gücü 1960'larda Minsky ve Papert tarafından vurgulanmıştır. Genelde, karmaşık sahici uygulamalar doğrusal fonksiyonlardan daha anlamlı hipotez uzaylarına ihtiyaç duyarlar. Bu problemi görmenin diğer bir yolu şudur: Hedef kavram verilen niteliklerin basit doğrusal kombinasyonu ile ifade edilemez, fakat genelde veriler yararlanılmış daha fazla soyut özniteliklere ihtiyaç duyarlar. Eşiklenmiş doğrusal fonksiyonların çoklu katmanları bu probleme çözüm olarak sunulmuştur. Bu yaklaşım sistemi eğitmek için çoklu-katmanlı sinir ağları ve geri yayımlı gibi öğrenme algoritmalarının geliştirilmesine ihtiyaç duyar.

Çekirdek gösterimler, doğrusal öğrenme makinelerinin hesaplama yeteneğini arttırmak için verileri yüksek boyutlu öznitelik uzayına eşleyerek alternatif bir çözüm

sunarlar. Doğrusal makinelerin ikili gösterimi bu adımı dolaylı olarak gerçekleştirmemizi mümkün kılar. Eğitim örnekleri hiçbir zaman tek başına bırakılmış görünmez, her zaman örnek çiftleri arasında iç çarpım biçiminde görünür. Makineleri ikili gösterimiyle kullanmanın avantajı bu gösterimde ayarlanabilir parametre sayısının kullanılan özellik sayısına bağımlı olmaması gerçeğinden ortaya çıkmaktadır. İç çarpımı uygun bir şekilde çekirdek fonksiyonu ile değiştirerek, ayarlanabilir parametre sayısını arttırmadan yüksek boyutlu öznitelik uzayına doğrusal olmayan bir eşleme dolaylı olarak gerçekleştirilebilir. Burada çekirdek, iki girişe karşı düşen öznitelik vektörlerinin iç çarpımını gerçekleştirmektedir.

1.3.2.1. Öznitelik Uzayında Öğrenme

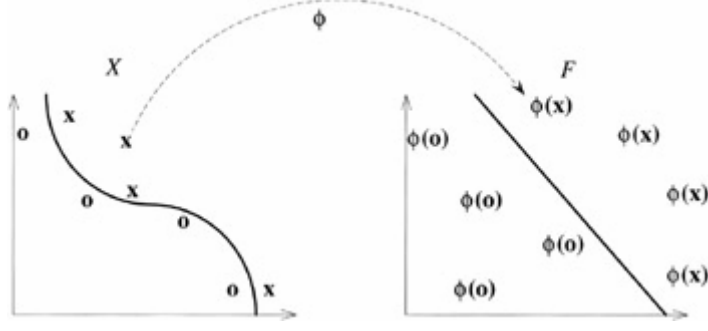
Öğrenilecek hedef fonksiyonun karmaşıklığı temsil edilme yoluna bağlıdır ve öğrenme işinin zorluğu da aynı sebepten dolayı değişiklik gösterebilir. İdealde belirli bir öğrenme problemini eşleyen gösterim seçilmelidir. Bu yüzden, otomatik öğrenmede yaygın kullanılan bir önışlem stratejisi veri gösteriminin değiştirilmesidir:

$$x = (x_1, \dots, x_n) \mapsto \phi(x) = (\phi_1(x), \dots, \phi_N(x)) \quad (1.90)$$

Bu adım X giriş uzayını yeni bir uzaya ($F = \{\phi(x) \mid x \in X\}$) eşlemeye eşdeğerdir.

Sadece verileri diğer bir uzaya eşleme uzun zamandır otomatik öğrenmede bilinen ve birtakım tekniğin ortaya çıkmasına sebep olan verilerin en iyi temsilini seçmek işini oldukça kolaylaştırır. En uygun gösterimi seçme işi öznitelik seçimi olarak bilinir. X uzayı giriş uzayını gösterirken, $F = \{\phi(x) \mid x \in X\}$ uzayı öznitelik uzayı olarak adlandırılır.

Şekil 1.21., iki boyutlu giriş uzayından iki boyutlu öznitelik uzayına bir öznitelik eşleme örneği göstermektedir. Bu örnekte veriler doğrusal bir fonksiyonla giriş uzayında ayrılmazlarken öznitelik uzayında ayrılabilir hale gelmişlerdir.



Şekil 1.21. Öznitelik eşlemesi sınıflandırma işlemini kolaylaştırabilir

Öznitelik seçimi için değişik yaklaşımlar mevcuttur. Orijinal niteliklerde içerilen temel bilgiyi taşıyan en küçük öznitelik kümesini tanımlamak için araştırma yapılır. Bu boyut azaltımı olarak bilinir.

$$x = (x_1, \dots, x_n) \mapsto \phi(x) = (\phi_1(x), \dots, \phi_d(x)), d < n \quad (1.91)$$

Bu, bazen boyutluluk belası olarak adlandırılan bir olgu olan öznitelik sayısı arttığında hem hesaplama hem de genelleştirme performansının düşmesi açısından çok yararlı olabilecektir. Yüksek boyutlu öznitelik uzayındaki zorluklar, öznitelik kümesi geniş oldukça öğrenilecek fonksiyon sayısı da azalmış olacağından yersizdir.

Diğer bir değişik öznitelik seçme işlemi çıkış değerini etkilemeyen ilgisiz özniteliklerin belirlenmesi ve elenmesidir.

Temel bileşen analizini kullanma, verilerin öznitelik uzayına eşlenmesini sağlar. Bu öznitelik uzayında yeni öznitelikler orijinal niteliklerin doğrusal fonksiyonudurlar ve verilerin her bir yönde sergilediği varyans miktarına göre sıralanırlar. Boyut azaltımı bazen, sadece verilerin düşük varyansa sahip olduğu yönler karşı düşen özniteliklerin (her ne kadar bu özniteliklerin hedef sınıflandırmayı gerçeklemede gerekli olmadığının garanti edilememesine rağmen) kaldırılması ile gerçekleştirilir.

Şimdi, ek öznitelik boyutlarının da yararlı olabileceği iki boyutlu giriş uzayı durumunu ele alalım. Problem hakkındaki deney öncesi bilgimizin, ilgili bilgilerin 2. dereceden tek terimliler biçiminde kodlandığının tahmin edilmesi olduğunu varsayalım. Bu yüzden problemi bu tür bilgilerin açık olarak verildiği ve öğrenme makinesinin kullanımı için hazır olduğu öznitelik uzayında göstermek istiyoruz. Olası bir eşleme şu şekildedir:

$$(x_1, x_2) \mapsto \phi(x_1, x_2) = (x_1^2, x_2^2, x_1 x_2) \quad (1.92)$$

Aynı şekilde, $\binom{n+d-1}{d}$ boyutunda öznitelik uzayına sebep olacak d dereceli özniteliklerin kullanılması istenebilir. Bu durumda makul sayıda nitelik ve öznitelik derecesi için hesaplama uygunsuz hale gelmektedir. Bu tür öznitelik uzayını kullanmak için öznitelik uzayına dolaylı eşlemeyi içeren özel bir tekniğe ihtiyacımız olacaktır.

Hesaba dayalı problemler sadece kullandığımız öznitelik uzayının boyutu ile bağlantılı değildir. Diğer bir zorluk kaynağı, hipotezlerin standart fonksiyon sınıfları için gösterim boyutuna duyarlı olabilen öğrenme makinelerinin genelleştirilmesidir.

Önceki örnekten de açık olduğu gibi, öznitelik seçimi öğrenme sürecinin bir parçasıdır ve mümkün olduğunca otomatikleştirilmiş olmalıdır. Diğer yandan bu, altta yatan hedef fonksiyonu hakkındaki deney öncesi tahminlerimizi yansıtan biraz keyfi bir adımdır. Öğrenmenin teorik modelleri bu adımı hesaba katmalıdır: Genelleştirme bir şekilde kontrol altına alınamazsa çok büyük öznitelik kümesi oluşturmak aşırı yükleme problemlerine yol açabilir.

1.3.2.2. Öznitelik Uzayına Dolaylı Eşleme

Doğrusal olmayan ilişkileri doğrusal bir makine ile öğrenmek için bir grup doğrusal olmayan öznitelik seçmemiz ve yeni gösterimde verileri yeniden yazmamız gerekmektedir. Bu, verilerin doğrusal makine tarafından kullanılabilen öznitelik uzayına sabit doğrusal olmayan eşlenmesine eşdeğerdir. Bundan dolayı ele alacağımız hipotezler kümesi şu biçimdeki fonksiyonlar olacaktır:

$$f(x) = \sum_{i=1}^N w_i \phi_i(x) + b \quad (1.93)$$

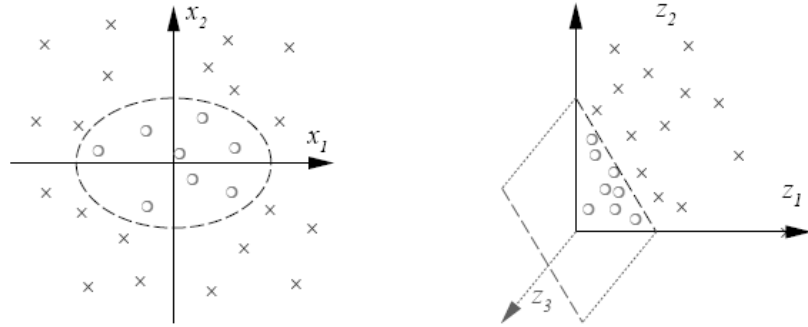
Burada $\phi: X \rightarrow F$, giriş uzayından öznitelik uzayına doğrusal olmayan eşlemedir. Bu, doğrusal olmayan makineleri iki adımda inşa edecek olmamız demektir: önce sabit doğrusal olmayan eşleme, verileri bir F öznitelik uzayına dönüştürür ve ardından bir doğrusal makine onları öznitelik uzayında sınıflandırmak için kullanır.

Doğrusal öğrenme makinelerinin önemli bir özelliği ikili gösterimle ifade edilebiliyor olmalarıdır. Bu demektir ki, hipotezler karar kuralının sadece test ve eğitim noktaları arasındaki iç çarpımını kullanarak hesaplanabileceğinden eğitim noktalarının doğrusal kombinasyonu olarak ifade edilebilirler.

$$f(x) = \sum_{i=1}^{\ell} \alpha_i y_i \langle \phi(x_i) \cdot \phi(x) \rangle + b \quad (1.94)$$

Eğer $\langle \phi(x_i) \cdot \phi(x) \rangle$ iç çarpımını öznitelik uzayında orijinal giriş noktalarının bir fonksiyonu olarak doğrudan hesaplayabiliyorsak doğrusal olmayan öğrenme makinesini inşa etmek için gerekli iki adımı birleştirmemiz mümkün olacaktır. Bu şekilde doğrudan hesaplama yöntemine çekirdek fonksiyon diyoruz [23].

Şekil 1.22.'de gösterilen örnekte ayrılamayan problem, giriş verilerinin ϕ_2 ile önişlemden geçirilmesi sonucu ayırma hiper düzleminin oluşturulmasına dönüştürülmüştür. Bunun hesaplama bakış açısından (hiper düzlemi hesaplamak için etkin bir algoritmanın olması) ve istatistiksel bakış açısından (hiper düzlemin test noktaları için iyi genelleştirileceğinin garanti edilmesi) bakıldığında yararları görülecektir.



Şekil 1.22. Öznitelik uzayına eşlenmiş ikili sınıflandırma problemi örneği

Şekil 1.22.'de giriş uzayında (sol panel) uygun karar sınırının elips olduğunu varsayıyoruz. Öğrenme süreci, bu sınırın her iki sınıfa ait eğitim noktalarını içeren gözlemsel verilere dayanılarak tahmin edilmesidir. Doğrusal olmayan $\phi_2(x) = (z_1, z_2, z_3) = (|x|_1^2, |x|_2^2, \sqrt{2}|x|_1|x|_2)$ eşlemesi ile öznitelik uzayına (sağ panel) eşlendiğinde elips bir hiper düzlem halini alır. Bu, elipslerin (z_1, z_2, z_3) girişlerine sahip

lineer eşitlikler olarak yazılabileceğinden dolayıdır. Bu yüzden, öznitelik uzayında problem, eşlenen veri noktalarından hiper düzlemin tahminine indirgenmiş oluyor. Polinomsal çekirdeklerle üç boyutlu uzaydaki iç çarpım ϕ_2 'yi hesaplamadan bulunabilir.

Çoğu gerçekçi durumda, örneğin x , elemanları piksel değerleri olan bir görüntüyü temsil ederse, polinomsal çekirdekler $\langle x, x' \rangle^d$ bizim herhangi d piksel değer çarpımı ile uzanan uzayda çalışmamıza imkan tanır. Çalışmamızı eşlenmiş örüntünün herhangi bir açık kullanımını gerektirmeden iç çarpımlarla gerçekleştirmemizi sağlar. Çekirdekleri kullanmakla, zaman ve bellek karmaşıklığında kombinezon patlamasına neden olmadan yüksek dereceden istatistiği hesaba katabiliriz [24].

Çekirdek, bütün $x, z \in X$ için (1.95) koşulunu sağlayan bir fonksiyondur.

$$K(x, z) = \langle \phi(x) \cdot \phi(z) \rangle \quad (1.95)$$

Burada ϕ , X 'den F öznitelik uzayına eşlemedir.

Çekirdek ismi, çekirdekler ve onların öznitelik uzayları arasındaki ilişki teorilerinin bir çoğunun temelini oluşturan integral operatörü teorisinden çıkarılmıştır. İkili gösterimin önemli bir sonucu, öznitelik uzayının hesaplamının etkilenmesine yol açmayışıdır. Öznitelik vektörlerinin açıkça gösterilmesine gerek olmadığı gibi, çekirdek fonksiyonu ile iç çarpımı hesaplamak için ihtiyaç duyulan işlem sayısı kesin olarak öznitelik sayısı ile orantılı değildir. Çekirdeklerin kullanılması verilerin öznitelik uzayına dolaylı olarak eşlenmesine ve doğrusal makinenin böyle bir uzayda eğitilmesine imkan tanır. Eğitim örnekleri hakkında kullanılan tek bilgi onların öznitelik uzayındaki Gram matrisidir. Bu matris ayrıca çekirdek matris olarak da adlandırılır ve bu bağlamda onu göstermek için K sembolünü kullanacağız. Bu yaklaşımın anahtar kısmı, etkin bir şekilde hesaplanabilen çekirdek fonksiyonunu bulmaktır. Böyle bir fonksiyona sahip olduğumuzda karar kuralı en fazla çekirdeğin ℓ hesaplaması ile belirlenebilir:

$$f(x) = \sum_{i=1}^{\ell} \alpha_i y_i K(x_i, x) + b \quad (1.96)$$

Çekirdek kullanmanın ilginç olgularından biri, öznitelik uzayında öğrenmeyi gerçekleştirebilmek için altta yatan öznitelik eşlemeyi bilmemize gerek olmayışıdır.

Çekirdek kavramı doğrudan doğruya sezgisel bir kavram değildir. Öncelikle, çekirdek fikri giriş uzayındaki standart iç çarpımı genelleştirir. Bu iç çarpımın, öznitelik eşlemenin özdeş yapılmasıyla çekirdeğe bir örnek teşkil ettiği açıktır.

$$K(x, z) = \langle x \cdot z \rangle \quad (1.97)$$

Ayrıca öznitelik eşlemeyi herhangi bir x (A matrisi için Ax) sabit doğrusal dönüşümü olarak ele alabiliriz. Bu durumda çekirdek fonksiyonu (1.98) şeklinde verilir.

$$K(x, z) = \langle Ax \cdot Az \rangle = x' A' A z = x' B z \quad (1.98)$$

Burada $B = A' A$ yarı kesin pozitif kare matristir. Amacımız doğrusalsızlığı öznitelik uzay eşlemesi ile ortaya koymaktır. Bu yüzden aşağıdaki ilişkiyi dikkate alarak elde edilen doğrusal olmayan eşlemeye basit, fakat açıklayıcı bir örnek üzerinde duracağız.

$$\begin{aligned} \langle x \cdot z \rangle^2 &= \left(\sum_{i=1}^n x_i z_i \right)^2 = \left(\sum_{i=1}^n x_i z_i \right) \left(\sum_{j=1}^n x_j z_j \right) \\ &= \sum_{i=1}^n \sum_{j=1}^n x_i x_j z_i z_j = \sum_{(i,j)=(1,1)}^{(n,n)} (x_i x_j) (z_i z_j) \end{aligned} \quad (1.99)$$

Buradan çekirdek işleminin öznitelik vektörleri arasındaki iç çarpıma eşdeğer olduğu söylenebilir.

$$\langle x \cdot z \rangle^2 = \langle \phi(x) \cdot \phi(z) \rangle, \quad \phi(x) = \left(x_i x_j \right)_{(i,j)=(1,1)}^{(n,n)} \quad (1.100)$$

Bu durumda özniteliklerin hepsi 2. dereceden tek terimlilerdir. Fakat şunu da not etmek gerekir ki $i \neq j$ olduğu durumlarda $x_i x_j$ özniteliği iki defa gerçekleştiğinden ağırlığı x_i^2 özniteliğinin ağırlığının iki katı olmaktadır. Daha genel öznitelik uzayı çekirdeğin (1.101)'deki gibi hesaba katılmasıyla elde edilir.

$$\begin{aligned}
(\langle x \cdot z \rangle + c)^2 &= \left(\sum_{i=1}^n x_i z_i + c \right) \left(\sum_{j=1}^n x_j z_j + c \right) \\
&= \sum_{i=1}^n \sum_{j=1}^n x_i x_j z_i z_j + 2c \sum_{i=1}^n x_i z_i + c^2 \\
&= \sum_{(i,j)=(1,1)}^{(n,n)} (x_i x_j) (z_i z_j) + \sum_{i=1}^n (\sqrt{2c} x_i) (\sqrt{2c} z_i) + c^2
\end{aligned} \tag{1.101}$$

Burada $\binom{n+1}{2} + n + 1 = \binom{n+2}{2}$ özniteliğin hepsi 2'ye kadar olan derecelere sahip tek terimlidir. Fakat 1 ve 2 dereceliler arasındaki göreceli ağırlıklandırmalar ile öznitelikler 0. derecenin kuvvetini belirleyen sabit öznitelik c parametresi aracılığıyla kontrol edilir. Benzer türetmeler çekirdek fonksiyonlar için de yapılabilir.

$$K(x, z) = \langle x \cdot z \rangle^d \quad \text{ve} \quad K(x, z) = (\langle x \cdot z \rangle + c)^d, \quad d \geq 2 \text{ için} \tag{1.102}$$

İlk çekirdek için $\binom{n+d-1}{d}$ ayrı özniteliğin hepsi, ağırlıkları üstelin yapısına bağlı olarak değişecek olmasına rağmen d derecelidirler. İkinci çekirdek için bütün tek terimlilerin d ve d 'ye kadar olan derecelere sahip olduğu $\binom{n+d}{d}$ ayrı öznitelik vardır. Bu öznitelik uzaylarında bir hiper düzleme karşı düşen giriş uzayındaki karar sınırı d dereceli polinomsal eğrilerdir. Bu yüzden çekirdekler çoğunlukla polinomsal çekirdekler olarak adlandırılır.

1.3.2.3. Çekirdeklerin Oluşturulması

Çekirdek fonksiyonunun kullanımı hesaba dayalı cazibeli bir kısa yoldur. Eğer bu yaklaşım kullanılmak istenirse önce karmaşılaştırılmış öznitelik uzayının oluşturulması, sonra bu uzayda iç çarpımın ne olacağı çözülür ve son olarak da o değeri orijinal girişler cinsinden hesaplamak için direkt bir yöntemin bulunması gerekir. Pratikte anlatılan bu yaklaşım, doğrudan bir çekirdek fonksiyonu tanımlamaktır. Bu yüzden dolaylı olarak öznitelik uzayı tanımlanır. Bu yolla, öznitelik uzayından sadece iç çarpımların hesaplanmasında değil aynı zamanda öğrenme makinelerinin tasarımında da kaçınılmaz

olunur. Giriş uzayı için çekirdek fonksiyonu tanımlamanın çoğunlukla karmaşılaştırılmış öznitelik uzayı oluşturmaktan daha doğal olduğunu iddia ediyoruz. Bu yolu izlemeden önce ilk olarak $K(x,z)$ fonksiyonunun bir öznitelik uzayı için çekirdek olduğundan emin olmak amacıyla hangi özelliklerinin gerekli olduğunu belirlememiz gerekiyor. Açık bir biçimde, fonksiyon simetrik olmalı,

$$K(x, z) = \langle \phi(x) \cdot \phi(z) \rangle = \langle \phi(z) \cdot \phi(x) \rangle = K(z, x) \quad (1.103)$$

ve Cauchy-Schwarz eşitsizliğinin sonucu olan aşağıdaki eşitsizliğin sağlanması gerekmektedir.

$$\begin{aligned} K(x, z)^2 &= \langle \phi(x) \cdot \phi(z) \rangle^2 \leq \|\phi(x)\|^2 \|\phi(z)\|^2 \\ &= \langle \phi(x) \cdot \phi(x) \rangle \langle \phi(z) \cdot \phi(z) \rangle = K(x, x)K(z, z) \end{aligned} \quad (1.104)$$

Bununla birlikte bu koşullar öznitelik uzayının var olduğunun garanti edilmesi için yeterli değildir.

1.3.2.3.1. Mercer Teoremi

Bu alt bölümde $K(x,z)$ fonksiyonunun bir çekirdek olup olmadığını belirleyen Mercer teoremini sunacağız.

Teorem: $X, \mathbb{R}^n$ 'nin yoğun altkümesi, K da (1.105) ifadesi pozitif olacak şekilde seçilen sürekli simetrik fonksiyon olsun.

$$(T_K f)(\bullet) = \int_X K(\bullet, x)f(x)dx, \quad T_K : L_2(X) \rightarrow L_2(X) \quad (1.105)$$

Diğer bir ifadeyle, (1.106)'da verilen eşitsizlik sağlansın.

$$\int_{X \times X} K(x, z)f(x)f(z)dxdz \geq 0, \quad \text{bütün } f \in L_2(X) \text{ için} \quad (1.106)$$

Bu durumda $K(x,z)$ 'yi, $X \times X$ üzerinde düzgün yakınsayan seriler olan, $\|\phi_j\|_{L_2} = 1$ olacak şekilde normalize edilmiş ve pozitif ilişkilendirilmiş özdeğerleri $\lambda_j \geq 0$ olan T_K 'nin öz fonksiyonları $\phi_j \in L_2(X)$ 'i kullanarak genişletebiliriz (1.107).

$$K(x, z) = \sum_{j=1}^{\infty} \lambda_j \phi_j(x) \phi_j(z) \quad (1.107)$$

Çekirdek yöntemlerin karakteristikleri hakkında daha fazla bilgi için çekirdek yöntemlerle ilgili kaynaklara müracaat edilmelidir [23, 24, 25].

1.3.2.4. Çekirdek Fonksiyonlarına Örnekler

Aşağıdaki örnekler için $X \subset \mathbb{R}^N$ olarak ele alalım.

En yaygın kullanılan çekirdeklerden biri olan polinomsal çekirdeklerin homojen ve homojen olmayan biçimleri (1.108)'de gösterilmektedir.

$$\begin{aligned} k(x, x') &= \langle x, x' \rangle^d, \\ k(x, x') &= (\langle x, x' \rangle + c)^d, \quad d \in \mathbb{N}, c \geq 0 \end{aligned} \quad (1.108)$$

Bu çekirdekler, anlaşıldığı üzere d parametresi 1 olarak verildiğinde doğrusal çekirdeğe dönüşürler ve algoritmaların doğrusal biçimlerinin çalışmasına neden olurlar.

Diğer yaygın olarak kullanılan çekirdek radyal taban fonksiyonu çekirdeğidir. Bu çekirdekler (1.109)'da gösterilmektedir.

$$k(x, x') = f(d(x, x')) \quad (1.109)$$

(1.109)'da d , X üzerinde bir ölçü ve f de $\mathbb{R}_0^+$ üzerinde bir fonksiyondur. Örnek olarak Boser, Guyon ve Vapnik'in [26, 27, 28] ileri sürdüğü Gaussian'ı verebiliriz.

$$k(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right), \quad \sigma > 0 \quad (1.110)$$

Sigmoid çekirdekler ise (1.111)'de verildiği şekilde tanımlanmaktadır.

$$k(x, x') = \tanh(\kappa \langle x, x' \rangle + v), \quad \kappa > 0 \text{ ve } v < 0 \quad (1.111)$$

Sigmoid çekirdeğin aslında kesin pozitif çekirdek olmadığı görülebilir. Bununla birlikte pratikte başarılı biçimde uygulanmaktadır [24].

1.3.2.5. Çekirdekleri Normalleştirme

Eğer giriş değişkenleri çok fazla ise, çekirdeğin değeri eğitim işleminin düzgün çalışmamasına neden olabilecek derecede çok küçülür veya büyür. Bu problemin üstesinden gelebilmek için çekirdeklerin normalleştirilmesi önerilir.

[0,1] aralığındaki giriş aralığı için d dereceli polinomsal çekirdek için en büyük değer $(m+1)^d$ 'dir. Burada m giriş değişkenlerinin sayısıdır. Böylece büyük m değerleri için (100'den büyük) (1.112)'deki normalleştirilmiş polinomu elde ediyoruz.

$$H(x, x') = \frac{(x^T x' + 1)^d}{(m+1)^d} \quad (1.112)$$

RBF çekirdekte [0,1] giriş aralığı için en büyük $\|x - x'\|^2$ değeri m 'dir. Bu yüzden (1.113)'te verilen normalleştirilmiş RBF çekirdeği kullanabiliriz.

$$H(x, x') = \exp\left(-\frac{\gamma}{m} \|x - x'\|^2\right) \quad (1.113)$$

Bu çekirdeğin kullanımı, değişik sayıdaki giriş değişkenlerine sahip problemler için uygun γ değerinin seçimi için avantajlıdır [29].

1.3.3. Çekirdek PCA

Daha önce anlatıldığı gibi PCA klasik doğrusal öznitelik çıkartma tekniğidir. Son yıllarda doğrusal olmayan öznitelik çıkartma yöntemleri (çekirdek temel bileşen analizi (KPCA) gibi) ilgi odağı olmuştur.

KPCA doğrusal olmayan öznitelik çıkartma için, destek vektör makinelerinde uygulanan yöntemlerle yakından alakalı bir tekniktir. Çeşitli uygulamalar için gürültüyü kaldırma ve regresyon problemlerinde ön adım olma gibi yararları bulunmuştur. KPCA, Romdhani, Gong ve Psarrou [30] tarafından doğrusal olmayan istatistiksel yüz şeklinin modellenmesinin inşasında uygulanmıştır. Fakat onların şekil değişikliğini sınırlama yaklaşımı genel olarak geçerli değildir [31].

Giriş verisinin karmaşık, doğrusal olmayan ilişkilerini etkin bir şekilde gösterebilmek için çekirdek hüner açıklanmıştır ve son zamanlardaki çekirdek tabanlı doğrusal olmayan analiz yöntemleri çok ilgi uyandırmıştır. Çok yönlü olduğundan çekirdek yöntemler veri analizi araçları arasında günümüzde çok revaçtadır. Böyle algoritmalarda anahtar nesne çekirdek matrisidir (veri örneği üzerinde inşa edilmiş Gram matris) ve tayfı, algoritmanın performansına bağlıdır. Çekirdek matrislerin özdeğerlerinin davranışını, sağlamlığını ve ilgili çekirdek integral işleminin özdeğerleri ile nasıl ilişkide olduğunu öğrenmek, çekirdek tabanlı algoritmaların istatistiksel özelliklerini kavramak için kesinlikle önemlidir.

Çekirdek hüner önce giriş verisini doğrusal olmayan eşleme ile gizli öznitelik uzayı F' 'ye eşler ve ardından veriler F' 'de incelenir. KPCA, Schölkopf ve arkadaşları [32] tarafından geliştirilmiştir. Schölkopf, çekirdek hüneri PCA ile birleştirmeyi öne sürmüş ve öznitelik temsili için KPCA'yı geliştirmiştir. Önce, giriş verisi gizli öznitelik uzayı F' 'ye çekirdek hüner ile izdüşürülür ve ardından giriş verilerinin doğrusal olmayan temel bileşenlerini çıkartmak için doğrusal PCA, F uzayında gerçekleştirilir. Bu ayrıca, doğrusal olmayan altuzay analiz yöntemi olarak da adlandırılabilir. Yang'ın çalışmasında [33] KPCA'nın PCA'yı yüz tanımda performans açısından geçtiği rapor edilmiştir. Daha iyi sonuçlar Kim, Jung ve Kim [34] tarafından KPCA ile SVM sınıflandırıcının birleştirilmesi ile elde edilmiştir. Bununla beraber PCA gibi KPCA da giriş verilerinin toplam varyansını en aza indirmek için tasarlanmıştır ve ayırıklaştırma amacı için en uygun yöntem olduğu tam olarak söylenemez.

1.3.3.1. KPCA'nın Matematiksel Olarak İncelenmesi

n adet d -boyutlu örnek kümesi $(x_1, \dots, x_n)$ eğitim örnekleri olarak verilsin. PCA, bu örneklerin S_t varyansını en büyük değere çıkaran izdüşüm yönünü bulmaya çalışır. Bu da kovaryans matristen özdeğerleri bulmaya denktir.

$$S_t e = \lambda e \quad (\lambda \geq 0 \text{ özdeğerleri ve } e \in \mathbb{R}^d \text{ özvektörleri için}) \quad (1.114)$$

KPCA'da her bir x vektörü giriş uzayından $(\mathbb{R}^d)$ çok boyutlu F öznitelik uzayına doğrusal olmayan eşleme fonksiyonu ile $(\Phi: \mathbb{R}^d \rightarrow F)$ izdüşürülür. Öznitelik uzayının boyutunun keyfi olarak büyük olabileceğine dikkat edin. F 'de karşı düşen özdeğer problemi (1.115)'teki şekli alır.

$$S_t^\phi e^\phi = \lambda e^\phi, \quad S_t^\phi = \frac{1}{N} \sum_{i=1}^N \phi(x_i) \phi^T(x_i) \quad (1.115)$$

(1.115)'te S_t^ϕ kovaryans matrisidir. $\lambda \neq 0$ koşulunu sağlayan bütün e^ϕ çözümleri $\phi(x_1), \dots, \phi(x_n)$ 'lerin kapsadığı alanda uzanırlar. Diğer bir deyişle e^ϕ , öznitelik vektörlerinin doğrusal kombinasyonu şeklinde yazılabilir (1.116).

$$e^\phi = \sum_{i=1}^n \alpha_i \phi(x_i) \quad (1.116)$$

(1.115)'te (1.116)'yı yerine yazarsak (1.117) elde edilir.

$$\sum_{i=1}^N \sum_{j=1}^N \alpha_j \phi(x_i) k(x_i, x_j) = N \lambda \sum_{j=1}^N \alpha_j \phi(x_j) \quad (1.117)$$

(1.117)'de $k(x_i, x_j)$, (1.118)'da belirtilen şekilde tanımlanan iç çarpım çekirdeğidir.

$$k(x_i, x_j) = \phi^T(x_i) \phi(x_j) \quad (1.118)$$

(1.117)'de her iki tarafı, her iki tarafta da sadece $k(x_i, x_j)$ 'li terimlerin kalması için $\phi^T(x_k)$ ile çarparsak (1.119) sonucunu elde ederiz:

$$\sum_{i=1}^N \sum_{j=1}^N \alpha_j k(x_k, x_i) k(x_i, x_j) = N\lambda \sum_{j=1}^N \alpha_j k(x_k, x_j) \quad (1.119)$$

Şimdi bazı matris gösterimlerini ele alalım.

K , ij . elemanı $k(x_i, x_j)$ olan $n \times n$ çekirdek matris, $\alpha = [\alpha_1, \alpha_2, \dots, \alpha_N]^T$ olsun.

(1.119) eşitliği matris biçiminde tekrar yazılabilir.

$$K^2 \alpha = N\lambda K \alpha \quad (1.120)$$

(1.120)'de, $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_N$, K 'nın özdeğerlerine delalet eder ($\lambda_j = N\lambda_j$ gibi).

Buradan (1.121) eşitliğini elde ederiz:

$$K \alpha = \lambda \alpha \quad (1.121)$$

α , e^ϕ birim vektör olacak şekilde normleştirilmiştir. Bu aşağıdaki (1.122) eşitliği ile sağlanır.

$$\alpha_k^T \alpha_k = \frac{1}{\lambda_k} \quad (1.122)$$

Yukarıdaki çıkarımlarda bütün izdüşürölmüş örneklerin F 'de ortalandığı varsayımı yapılmaktadır.

Geleneksel PCA'nın KPCA'nın birinci dereceden polinomsal çekirdeğe sahip özel bir durumu olduğuna dikkat edin. Yani KPCA, değişik doğrusal olmayan izdüşürümler için değişik çekirdeklerin kullanıldığı geleneksel PCA'nın genelleştirilmiş şeklidir.

F 'deki vektörler, e^ϕ özvektörleri tarafından yayılarak oluşturulan daha düşük boyutlu uzaya izdüşürölür. x , F 'deki izdüşümü $\phi(x)$ olan test örneği olsun. $\phi(x)$ 'in e^ϕ özvektörlerine izdüşümü ϕ 'ye karşı düşen doğrusal olmayan temel bileşendir.

$$e^\phi \cdot \phi(x) = \sum_{i=1}^n \alpha_i (\phi(x_i) \cdot \phi(x)) = \sum_{i=1}^n \alpha_i k(x_i, x) \quad (1.123)$$

Diğer bir deyişle, ilk q ($1 \leq q \leq n$) doğrusal olmayan temel bileşen, çekirdek fonksiyonu kullanarak (açıkça külfetli bir işlem olan örnekleri çok boyutlu F uzayına eşleme işlemini yapmayarak) çıkartılabilir.

1.3.3.2. KPCA Algoritmasının İşlem Adımları

Matematiksel olarak ele aldığımız yöntemde yapılacak işlem adımlarını şu şekilde özetleyebiliriz:

1. Eğitim için K Gram matrisini oluştur (İstenilen çekirdek fonksiyonu kullanılabilir).

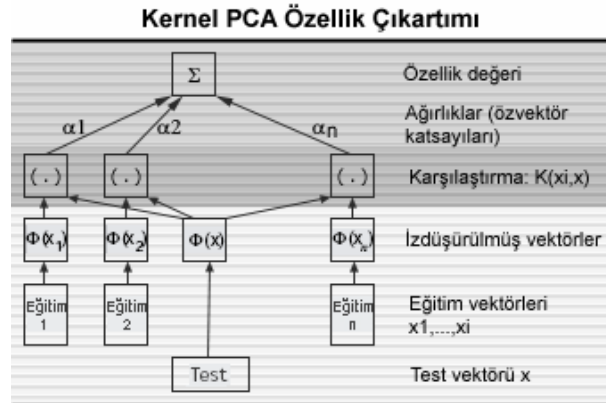
$$K_{ij} = k(x_i, x_j) = \phi(x_i) \cdot \phi(x_j) \quad (1.124)$$

2. K matrisinin özdeğer ve özvektörlerini bul.

3. $\alpha_k^T \alpha_k = \frac{1}{\lambda_k}$ olacak şekilde özvektörleri normalize et.

4. Her bir örnek için temel bileşen çıkar. ($\tilde{q}_k^T \phi(x) = \sum_{j=1}^N \alpha_{kj} k(x_j, x)$) (Şekil 1.23.)

5. Bu çıkartılan bileşenleri sınıflandırmada eğitim örnekleri olarak kullanmak üzere ayır ve test için benzer şekilde temel bileşenler çıkart.



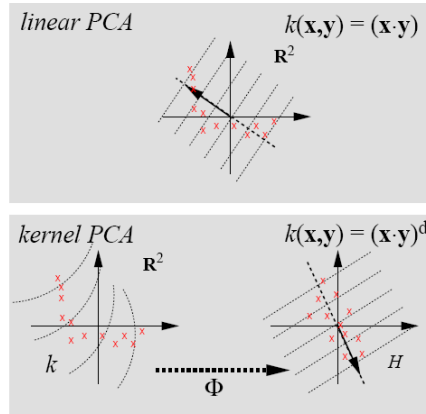
Şekil 1.23. KPCA tarafından inşa edilmiş öznelik çıkartıcı

Eğer Mercer'in koşulunu sağlayan bir çekirdek kullanıyorsak, bu prosedürün tam olarak çok boyutlu özellik uzayında standart PCA'ya karşı düştüğü (bu uzayda zor hesaplamaları gerçekleştirmeye gerek kalmadan) görülmektedir.

1.3.3.3. KPCA'nın Özellikleri

Mercer kernel'ları için, aslında F uzayında standart PCA'yı uyguladığımızı biliyoruz. Bu sebepten, PCA'nın bütün matematiksel ve istatistiksel özellikleri KPCA için de ufak bir farklılıkla (ifadelerde $\mathbb{R}^N$ yerine F uzayında bir grup $\phi(x_i)$, $i = 1, \dots, n$ noktası kullanılıyor) geçerlidir. Böylece, F 'de PCA'nın şu özelliklere sahip ortogonal dönüşüm olduğunu beyan ediyoruz (özvektörlerin, özdeğerlere göre azalan sırada sıralandığını varsayıyoruz):

- İlk q ($q \in [1, n]$) temel bileşen (bunlar özvektörlere izdüşümdür) diğer herhangi q ortogonal yönden daha fazla varyans içermektedir.
- Gözlemleri ilk q temel bileşen ile temsil etmede, ortalama karesel yaklaşım hatası en küçük değerdedir.
- Temel bileşenler ilişkisizdir.
- İlk q temel bileşen girişlere göre en fazla kalıtsal bilgiye sahiptir (bu Gaussian yaklaşımında geçerlidir ve böylece seçilen özel çekirdeğe ve veriye bağlıdır).



Şekil 1.24. KPCA'nın temel amacı

Şekil 1.24. göstermiştir ki, sayısı tam bilinmeyen çok boyutlu F uzayında (sağ altta) doğrusal PCA gerçekleştirmekteyiz. F doğrusal olmayarak (ϕ aracılığı ile) giriş uzayı ile ilişkili olduğundan temel özvektöre izdüşüm doğruları giriş uzayında doğrusal olmayana

dönüşecektir. Giriş uzayında özvektörün ön görüntüsünü çizemeyeceğimize dikkat edin. KPCA’da çok önemli bir gerçek, F' ’ye izdüşürmeyi gerçekleştirmemize gerek olmadığıdır. Bütün gerekli hesaplamalar k çekirdek fonksiyonu ile giriş uzayında gerçekleştirilmektedir. PCA’nın F ’deki bu özelliklerini giriş uzayındaki verilere göre değiştirirsek, çekirdeğin belirli seçimleri için incelenmesi gerekir.

Daha önce giriş koordinatlarındaki bütün mümkün d . dereceden tek terimlileri kapsayan boyutlara sahip öznitelik izdüşümüne karşı düşen polinomsal çekirdeklerin nasıl kullanıldığı açıklanmıştı. Farklı boyutlar her bir d pikseline ait sıralı çarpımın sayısının karekökü ile ölçeklenmiştir. Bu ölçekleme çarpanı kesinlikle, bütün ortogonal dönüşümler (dönme ve aynalama işlemleri gibi) grubunda KPCA’nın bağımsızlığını sağlar. Bu istenilen bir özelliktir (çıkartılan özniteliklerin giriş verimizin temsili için hangi ortonormal koordinat sistemini kullandığımıza bağlı olmaması özelliği).

1.3.3.3.1. Hesaplama Karmaşıklığı

256 boyutlu giriş uzayında 5. dereceden polinomsal çekirdek 10^{10} boyutlu öznitelik uzayı oluşturur. İki sebepten ötürü çekirdek PCA bu büyük boyutlarla uğraşabilir. İlk olarak, bütün F uzayındaki özvektörlere değil sadece F' ’deki x_k gözlemlerimizin görüntüleri tarafından yayılan alt uzaya bakmamıza gerek vardır. İkinci olarak, çekirdek fonksiyonları kullandığımızdan F' ’deki vektörler arasında açıkça iç çarpımı hesaplamamıza gerek yoktur. Çekirdek PCA böylece $\ell \times \ell$ iç çarpım ile ℓ gözlem üzerinde doğrusal PCA ile hesapsal olarak karşılaştırılabilir. Eğer k ’nın hesaplaması polinomsal çekirdeklerde olduğu gibi kolay ise, hesaplama karmaşıklığı iç çarpımın yerine çekirdek fonksiyonları hesapladığımız için neredeyse hiç değişmemiş olacaktır. Bundan başka, büyük ℓ değerlerinde gözlem kullanmamız gerektiğinde güç yöntemi gibi sadece en büyük özdeğerleri hesaplama algoritması ile çalışmak isteyebiliriz. Ayrıca, $M < \ell$ örnek altkütmesi ile hesaplanan K matrisinin kestirimini ele alabiliriz. Durum, temel bileşen çıkartımı için değişik olabilir. Çekirdek PCA’da çekirdek fonksiyonlarını, çıkartılan her bir temel bileşen için M kere hesaplamamız gerekirken, doğrusal PCA’da her bir bileşen için bir iç çarpım hesaplamamız gerekecektir. Tabi ki, F' ’nin boyutu 10^{10} ise bu, F' ’de doğrusal temel bileşen çıkartımından oldukça hızlı olacaktır. Bununla birlikte, temel bileşenlerin sınıflandırmada önışlem olarak çıkartmak gibi bazı durumlarda olayı daha hızlandırmak isteriz. Bu,

indirgenmiş küme ile gerçekleştirilebilir. Sunulan yöntemde her bir özvektörü aşağıdaki şekilde yaklaşımlandırırız:

$$v = \sum_{i=1}^{\ell} \alpha_i \phi(x_i) \quad (1.125)$$

Diğer bir vektör $m < \ell$ istenilen hızlanmaya göre deney öncesi seçilmiştir.

$$\tilde{v} = \sum_{j=1}^m \beta_j \phi(z_j), \quad z_j \in \mathbb{R}^N, \quad j = 1, \dots, m \quad (1.126)$$

Bu, karesel farkın minimize edilmesiyle yapılır.

$$\rho = \|V - \tilde{V}\|^2 \quad (1.127)$$

Bu, muhtemelen çok fazla boyuta sahip F uzayıyla açıkça uğraşmadan gerçekleştirilir. Çünkü, ρ 'nın β_j 'ye ve z_j 'ye göre gradyanti çekirdek fonksiyon açısından kolayca ifade edilebilir.

$$\rho = \|V\|^2 + \sum_{i,j=1}^m \beta_i \beta_j k(z_i, z_j) - 2 \sum_{i=1}^{\ell} \sum_{j=1}^m \alpha_i \beta_j k(x_i, z_j) \quad (1.128)$$

Böylece ρ , standart gradyant yöntemi ile minimize edilebilir. El yazısı harfi tanıma işinde bu teknik, doğrulukta hiçbir kayıp vermeden 50 kat hızlanmayı sağlamıştır. Son olarak, çekirdek temel bileşen analizi çıkartımının doğrusal karşılığına göre hesaplama açısından daha masraflı olmasına rağmen ek getirilerinin olacağını söyleyebiliriz.

1.3.3.3.2. Yorumlanabilirlik

PCA'da bazen, temel bileşen çıkartımı ile izdüşürülecek olan altuzaya yayılan belirli eksenlerin seçilebilmesi istenir. Bu yolla, örneğin yorumlanmaya daha açık değişkenleri seçmek mümkün olabilir. Doğrusal olmayan durumda, ek bir problem vardır: F 'nin bazı

elemanlarının giriş uzayında öngörüntüleri yoktur. Bunu anlayabilmek için, eğitim örneklerinin doğrusal yayılımının öznitelik uzayına eşlemesinin en fazla M (örnek sayısı) boyut içerebildiğini düşünün. Eğer örnek sayısı giriş uzayının boyutunu geçerse her bir vektörün öngörüntüsünün olması olasılık dışı olacaktır. Yorumlayabilmek için, ϕ dönüşümü altındaki görüntüleri F altuzayında yayılan giriş uzayındaki yönleri bulmamız gerekecektir.

1.3.3.3. Boyut Azaltımı, Öznitelik Çıkartımı ve Yeniden Üretim

Doğrusal PCA'dan farklı olarak sunulan yöntem giriş boyutunu geçebilecek temel bileşen çıkartımını mümkün kılmaktadır. Gözlemlerin sayısının (M), giriş boyutunu (N) geçtiğini varsayın. Doğrusal PCA, $M \times M$ iç çarpım matrisine dayandığında bile en fazla N adet 0 olmayan özdeğer bulur. Buna karşın, çekirdek PCA M adet 0 olmayan özdeğer bulabilir (bu, çekirdek PCA'nın $N \times N$ kovaryans matrisiyle direkt hesaplanamayacağının göstergesidir).

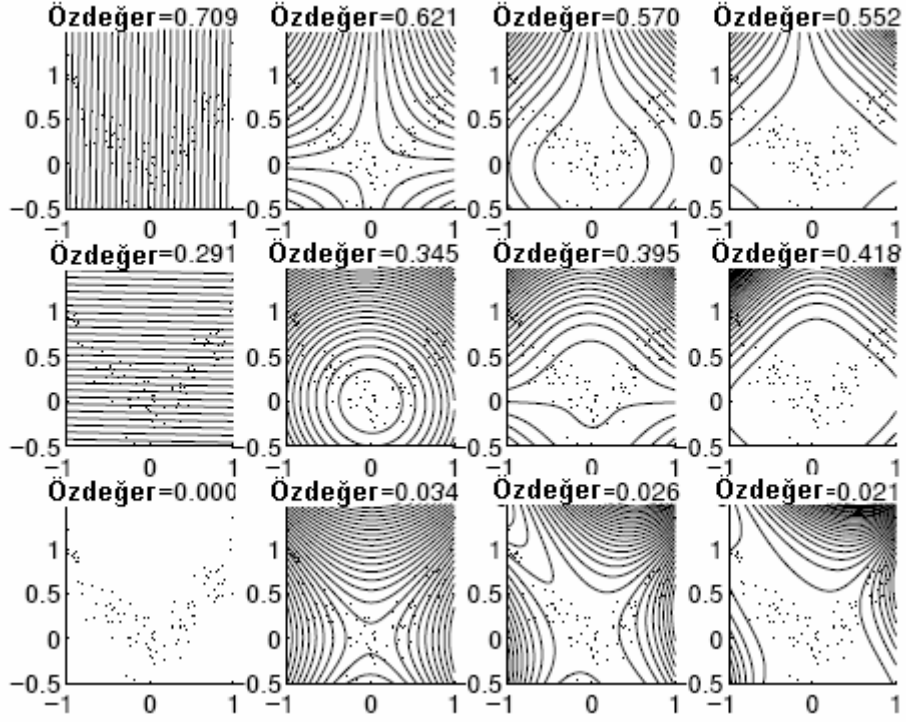
Temel dönüşüm olmasıyla PCA, orijinal görüntünün yeniden eldesine izin verir. Genelde yeniden üretimler iyi sonuç verir. Çekirdek PCA'da bu biraz daha zordur: F'deki görüntünün görüntüsünü onun doğrusal olmayan bileşenleriyle yeniden üretebiliriz. Bununla beraber, eğer sadece yaklaşık yeniden oluşturmaya sahipsek giriş uzayındaki öngörüntünün yeniden üretme ile tam olarak bulunabileceği garanti edilemez. Bu durumda yaklaştırma yöntemine başvurmamız gerekir. Alternatif olarak uygun regresyon yöntemini de kullanabiliriz [35].

1.3.3.4. KPCA Deneyleri

Bu bölümde, özellik çıkartımı için çekirdek PCA'nın kullanıldığı basit bir örnek incelenecektir.

H'deki PCA'nın giriş uzayında nasıl bir davranış sergilediğini kavramak için polinomsal çekirdekleri kullanarak yapay 2 boyutlu veri kümesi tanımlıyoruz.

$$k(x, x') = \langle x, x' \rangle^d, \quad d = 1, \dots, 4 \quad (1.129)$$

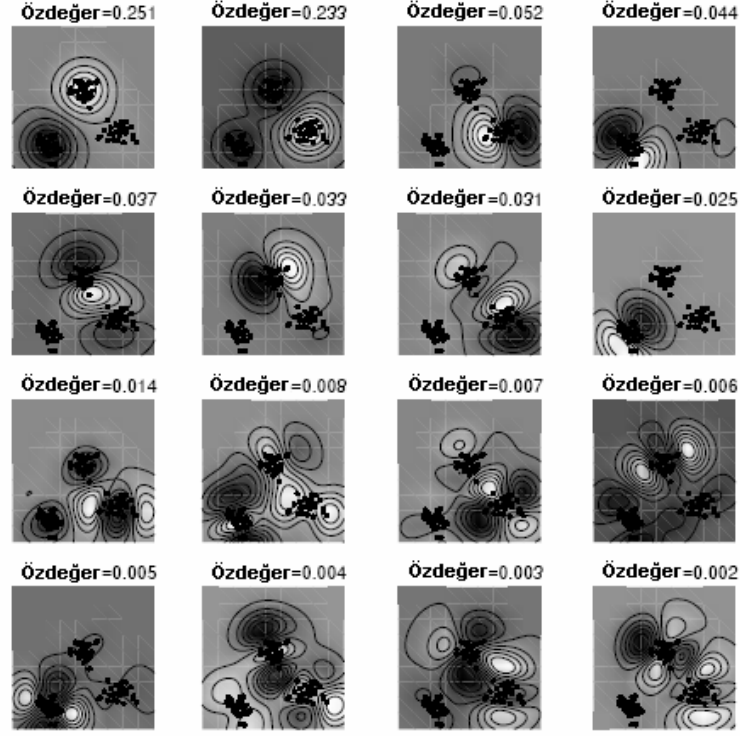


Şekil 1.25. İki boyutlu basit bir KPCA örneği

Şekil 1.25.'te veriler şu şekilde üretilmiştir: x değerleri $[-1,1]$ aralığında düzgün dağılmış, y değerleri de $y_1 = x_1^2 + v$ (v , 0.2 standart sapmaya sahip gürültüyü ifade ediyor) formülü ile hesaplanmıştır. Soldan sağa çekirdekteki polinomsal dereceler 1'den 4'e değişmektedir. Üstten alta ilk 3 özvektör özdeğerleri büyükten küçüğe olacak şekilde sıralanmaktadır. Şekiller sabit temel bileşen çizgilerini göstermektedir. Doğrusal durumda bunlar özvektörlere diktirler.

Doğrusal PCA (Şekil 1.25.'te) giriş uzayının boyut sayısı 2 olduğundan sadece 2 tane sıfırdan farklı özdeğer ile sonuçlanır. Aksine, doğrusal olmayan PCA daha fazla bileşen çıkmasına izin verir. Şekilde, doğrusal olmayan PCA'nın (verinin yapısını doğrusal PCA'dan daha iyi yansıtarak) sabit öznelik değerlerinin sınır çizgilerini ürettiğine dikkat edin. Her durumda ilk temel bileşen verinin temelinde yatan parabol boyunca tek düze olarak değişir. Doğrusal olmayan durumlarda ikinci ve üçüncü bileşenler de davranışı gösterebilir. Polinomsal derece 2 olduğunda şekilden de görüldüğü gibi küçük özdeğerle gelen üçüncü bileşen gürültüden kaynaklanan değişimleri içermektedir. Bu bileşeni kaldırmak gürültüyü azaltmaya karşılık gelmektedir.

Diğer bir örnekte de (Şekil 1.26.) radyal taban fonksiyonu çekirdeğinin kullanıldığı örüntü uzayı gösterilmektedir.



Şekil 1.26. Gaussian çekirdeğin kullanıldığı 2 boyutlu bir örneğin ilk 16 doğrusal olmayan temel bileşenine karşı düşen görüntüler

KPCA, diğer yaklaşımlarla karşılaştırıldığında, hiç bir doğrusal olmayan optimizasyon içermemesi en büyük avantajıdır. Ayrıca, çıkartmak istediğimiz bileşen sayısını önceden belirtmemize de gerek yoktur. Sinirsel yaklaşımlarla karşılaştırıldığında eğer çok fazla sayıda gözlemler işlenecekse bu durum büyük K matrisine yol açacağı için KPCA dezavantajlı olabilir. Temel eğrilerle kıyaslandığında KPCA'nın giriş uzayında yorumlanmasının daha zor olduğu görülür; bununla beraber, en azından polinomsal çekirdekler için yüksek dereceden öznitelikleri kolayca yorumlanabilmektedir [24].

1.4. Sınıflandırma

1.4.1. Euclidean Doğrusal Ayırtaçları

Euclidean uzaklığı kullanan iki boyutlu iki sınıflı en küçük uzaklık sınıflandırıcı, d boyutlu x öznitelik uzayı ve m_k prototipleri ile temsil edilen istenilen sayıda w_k sınıfı için

apaçık bir genelleştirmeye sahiptir. x öz nitelik vektörü ile m_k prortipi arasındaki Euclidean uzaklığının karesi şu şekilde ifade edilir:

$$d_k^2(x) = \|x - m_k\|^2 = (x - m_k)'(x - m_k) = x'x - m_k'x - x'm_k + m_k'm_k \quad (1.130)$$

w_k sınıfını, $d_k^2(x)$ değerini en küçük yapan m_k değerine göre belirliyoruz. Terimler m_k 'ya göre gruplanırsa (1.131) eşitliği elde edilir.

$$d_k^2(x) = -2(m_k'x - 0.5m_k'm_k) + x'x \quad (1.131)$$

$c=2$ olduğu varsayılın. Sınıflar arasındaki karar sınırı (1.132)'ye karşı düşer:

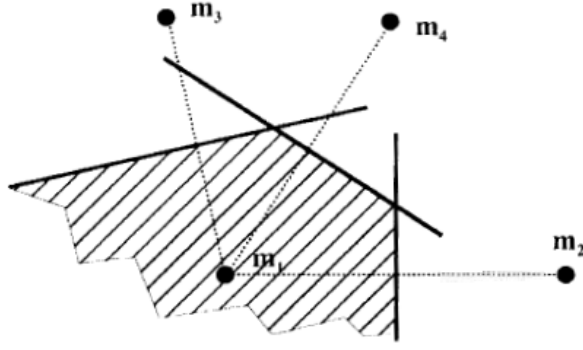
$$d_1^2(x) = d_2^2(x) \quad (1.132)$$

Böylece (1.131) kullanılarak (1.133) sonucu elde edilir.

$$\begin{aligned} m_1'x - 0.5\|m_1\|^2 &= m_2'x - 0.5\|m_2\|^2 \\ \Rightarrow (m_1 - m_2)'[x - 0.5(m_1 + m_2)] &= 0 \end{aligned} \quad (1.133)$$

x' 'de doğrusal olan bu son eşitlik $(m_1 - m_2)'$ doğrultusuna dik ve ortalamaların ortasını ifade eden $0.5(m_1 + m_2)$ noktasından geçen bir hiper düzlemi temsil eder. Bu, w_1 ile w_2 'yi ayıran bir doğrusal ayırtaçtır.

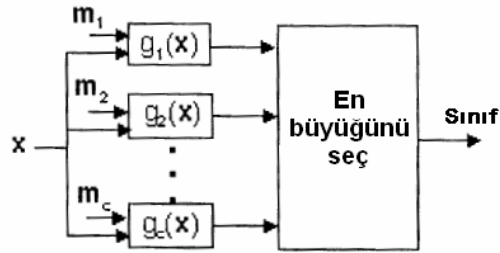
c sınıf için en küçük uzaklık ayırtacı, ikili olarak ayrılan hiper düzlem kesimlerinden oluşan parça parça doğrusaldır. Şekil 1.27., bu sınıflandırmayı şekilsel olarak göstermektedir.



Şekil 1.27. w_1 sınıfı için karar bölgesi doğrusal ayırtaçların diğer sınıflara bağlı olduğunu göstermektedir

$d_k^2(x)$ uzaklığını en küçük değere getirmek aşağıdaki karar fonksiyonunun en büyük değere getirilmesine denktir:

$$g_k(x) = m'_k x - 0.5 \|m_k\|^2 = w'_k x + w_{k,0}; \quad w_k = m_k, \quad w_{k,0} = -0.5 \|m_k\|^2 \quad (1.134)$$



Şekil 1.28. x giriş vektörü için en büyük karar fonksiyonu sınıflandırıcı

Şekil 1.28., en büyük karar fonksiyonu sınıflandırıcısının genel yapısını göstermektedir. $g_k(x)$ 'lerin, ortalama vektörü ağırlık vektörüne eşit olan ve yanlılık terimi ortalama vektörünün uzunluğuna bağlı olan doğrusal karar fonksiyonu olduğuna dikkat edin.

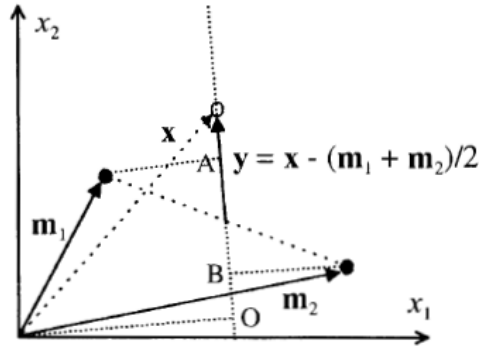
Önceki $c=2$ durumuna başvurarak doğrusal ayırma sistemini daha yakından inceleyebiliriz. Şimdi yeni bir öznitelik vektörü $y = x - 0.5(m_1 + m_2)$ ile ilgilendiğimizden koordinat ekseninin bir ötelemeye uğradığını varsayın. Doğrusal ayırma fonksiyonları (1.135) ile ifade edilir.

$$h_i(y) = m'_i y - 0.5 \|m_i\|^2, \quad i = 1, 2 \quad (1.135)$$

Burada m_1 ve m_2 'nin yeni koordinat sisteminde ifade edilmesiyle $\|m_1\| = \|m_2\|$ eşitliğinin elde edildiği açıkça görülmektedir. Bu yüzden ayırma fonksiyonu (1.136) ile ifade edilebilir.

$$h_i(y) = m_i' y \quad (1.136)$$

$m_i' y$, m_i' ve y arasındaki ilişki vektörü olduğundan Euclidean doğrusal ayırtaç, maksimum ilişki sınıflandırıcısı ya da minimum uzaklık sınıflandırıcısı olarak da bilinir. İlişki vektörleri, vektörler arasındaki açığa bağlı bir bilgi sağlar. Açık düştükçe artarlar ve sıfır derecede maksimuma ulaşırlar. Bu, benzerlik ölçüsünün alternatif olarak (vektörel izdüşümsel) yorumlanmasıdır. Bilinmeyen x örüntüsünün sınıf üyeliğini belirlemek amacıyla tanımladığımız teknik, şablon eşleme olarak adlandırılan en eski örüntü tanıma yöntemlerinden biridir. Her bir yeni örüntü, ilişki ölçüsüne göre kaydedilmiş şablonla eşleştirilir [36].



Şekil 1.29. x öznelik vektörünün maksimum ilişki yaklaşımı ile sınıflandırılması

1.4.2. En Yakın Komşuluk Kuralı

$D^n = \{x_1, \dots, x_n\}$, n tane etiketlenmiş prototip kümesini; $x' \in D^n$ de x test noktasına en yakın prototipi belirtsin. Bu durumda x' i sınıflandırmak için en yakın komşuluk kuralı, x' ile ilişkilendirilmiş etiketin atanması olarak ele alınabilir. En yakın komşuluk kuralı hemen hemen en uygun prosedürdür. Kullanımı çoğunlukla Bayes oranı ile elde edilen mümkün en

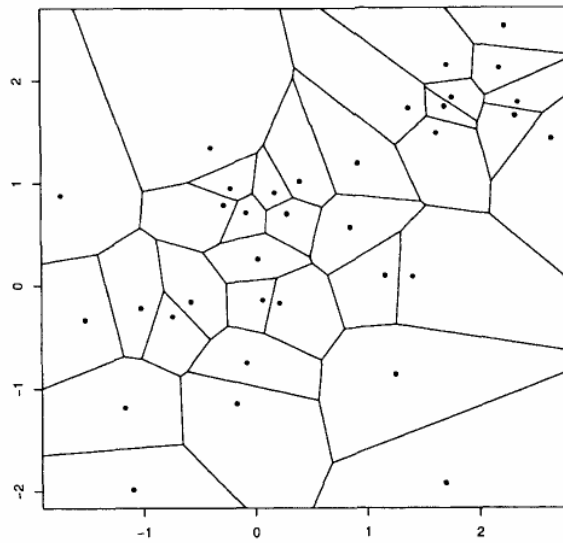
düşük değerden daha iyi hata oranı verir. Bununla beraber, sınırsız sayıda prototiple hata oranının Bayes oranının iki katından daha kötü olmayacağı görülebilir.

Ayrıntıya girmeden önce, en yakın komşuluk yönteminin neden bu kadar iyi çalıştığına sezgisel bir anlam verelim. Başlarken, en yakın komşu ile ilişkili θ' etiketinin rastgele değişken ve $\theta' = \omega_i$ olasılığının da sadece bir $P(\omega_i | x')$ deney sonrası olasılığı olduğuna dikkat edin. Örnek sayısı çok fazla olduğu zaman x' nün x e yeteri kadar yakın olduğu varsayımı ($P(\omega_i | x') \approx P(\omega_i | x)$) mantıklı bir yaklaşımdır. Bunun yapının ω_i durumunda olma olasılığı olmasından dolayı, en yakın komşuluk kuralı olasılıkları yapı ile verimli bir şekilde eşler.

Eğer ω_m yi (1.137)'deki şekilde tanımlarsak Bayes karar kuralı her zaman ω_m yi seçecektir.

$$P(\omega_m | x) = \max_i P(\omega_i | x) \quad (1.137)$$

Bu kural, öznitelik uzayını verilen bir eğitim noktasına diğer eğitim noktalarına göre daha yakın olan bütün noktaları içeren parçalara bölmemizi sağlar. Böyle bir hücredeki bütün noktalar eğitim noktasının sınıfı ile etiketlenir. Buna Voronoi mozaïği adı verilir (Şekil 1.30.).



Şekil 1.30. Düzlemde bir nokta kümesi ile tanımlanmış Voronoi mozaïğinin bir kısmı

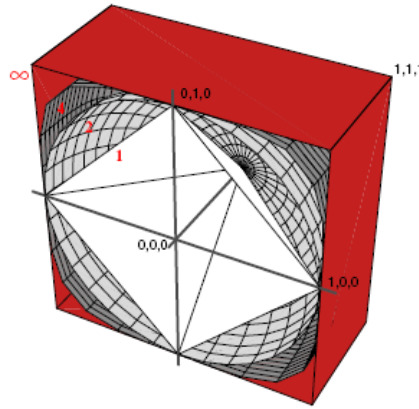
$P(\omega_m | x)$ olasılığı bire yakın olduğunda en yakın komşuluk seçimi yaklaşık olarak Bayes seçimine yakın olur. Bu, minimum hata olasılığın düşük olması durumunda en yakın komşuluk hata olasılığının da küçük olacağını belirtir. $P(\omega_m | x)$, $1/c$ değerine yakın olduğunda, yani bütün sınıflar eşit olasılığa sahip olduğunda Bayes karar kuralı ile en yakın komşuluk kuralının yaptığı seçimler nadiren aynı olur, fakat her ikisinde de hata olasılığı yaklaşık $1-1/c$ 'dir. Daha detaylı araştırma gerekmesine rağmen, bu gözlemler en yakın komşuluk kuralının performansının iyi olmasını, beklenmeyen bir durum olmaktan çıkarmaktadır.

En yakın komşuluk sınıflandırıcı, örüntüler arasında bir ölçüt ya da uzaklık fonksiyonuna bağlıdır. Genelde ölçüt olarak Euclidean uzaklığının kullanımı yaygınsa da problemimize göre alternatif başka ölçütlerin kullanımı ile daha iyi sonuçların alınabileceği unutulmamalıdır.

d boyutlu örüntüler için genel bir ölçüt sınıfı Minkowski ölçütüdür:

$$L_k(a, b) = \left(\sum_{i=1}^d |a_i - b_i|^k \right)^{1/k} \quad (1.138)$$

Bu ifade L_k norm olarak da adlandırılmaktadır. Buradan, Euclidean uzaklığı L_2 norma, Manhattan uzaklığı da L_1 norma karşı düşer. Şekil 1.31.'de değişik k değerleri için oluşan uzaylar gösterilmektedir.



Şekil 1.31. Minkowski ölçütünde değişik k değerleri için orijinden 1.0 uzaklığındaki noktaları kapsayan bölgeler

Şekil 1.31.'de beyaz yüzey Manhattan uzaklığını, açık gri de Euclidean uzaklığını göstermektedir.

Ayrıca Tanimoto ve tanjant uzaklıkları da kullanılan diğer ölçütler olarak gözükmemektedir [14].

Özellikle çok boyutlu veriler için iyi sonuç verebilecek, Euclidean uzaklığının özel bir biçimi (ağırlıklandırılmış Euclidean uzaklığı) daha vardır. Aşağıda formülü verilen bu biçimde, Euclidean uzaklığı standart sapma değerine bölünerek ağırlıklandırılır.

$$L_k(a, b) = \left(\sum_{i=1}^d \frac{|a_i - b_i|^2}{(\sigma_k)^2} \right)^{1/2} \quad (1.139)$$

1.4.3. Destek Vektör Makineleri

Destek Vektör Makinelerinin (Support Vector Machines (SVM)) altyapısı Vapnik [28] tarafından geliştirilmiş ve çekici bir çok özelliğinden dolayı popülerlik kazanmıştır. Formülasyon, klasik sinir ağları tarafından kullanılan geleneksel Deneysel Risk Minimizasyonundan (Experimental Risk Minimization (ERM)) daha iyi olan Yapısal Risk Minimizasyonu (Structural Risk Minimization (SRM)) prensibini şekillendirir. SRM, eğitim verilerindeki hatayı en küçük değerine getiren ERM'nin aksine genelleştirme hatasındaki üst sınırı en küçük değerine getirir. Bu, istatistiksel öğrenmedeki hedefimiz olan, SVM'ye genelleştirme yeteneğinde artış sağlayan farktır. SVM'ler sınıflandırma problemini çözmek için geliştirilmişlerdir, fakat son zamanlarda regresyon problemlerine de genişletilmeye başlanmıştır [37].



Şekil 1.32. İki adet doğrusal ayrılabilen veri kümesi. (a) Rastgele seçilmiş ayırma düzlemi (b) En büyük marjini verecek ayırma düzlemi

Destek Vektör Makineleri çok basit bir fikir üzerine kurulmuştur: Örüntü vektörlerinin en iyi ayırma hiper düzlemini (en büyük marjın hiper düzlemi) oluşturacak çok boyutlu öznitelik uzayına eşlenmesi ve sınıflandırılmasıdır.

Bu bölümde destek vektör modelinin arkasındaki temel düşünceleri açıklayacağız. Destek vektör sınıflandırıcıların büyük kısmı ikili sınıflandırma üzerinde durmaktadır. Çok sınıflı sınıflandırıcılar bir kaç ikili sınıflandırıcının birleştirilmesi ile elde edilmektedir.

1.4.3.1. Doğrusal Olarak Ayrılabilen Veriler

Bir grup eğitim örneğinin $(\{x_i, i = 1, \dots, n\})$, $y_i = \pm 1$ etiketleriyle w_1 veya w_2 sınıfına ait olduğu ikili bir sınıflandırma işlemini ele alalım. Doğrusal ayırma fonksiyonunu (1.140)'daki gibi belirtelim.

$$g(x) = w^T x + w_0 \quad (1.140)$$

Bu durumda karar kuralı (1.141)'deki şekli alır.

$$w^T x + w_0 \begin{cases} > 0 \\ < 0 \end{cases} \Rightarrow x \in \begin{cases} w_1, \text{ karşı düşen sayısal değer } y_i = +1 \\ w_2, \text{ karşı düşen sayısal değer } y_i = -1 \end{cases} \quad (1.141)$$

Böylece, bütün eğitim noktaları (1.142) koşuluna göre doğru sınıflandırılmış olmaktadır.

$$y_i(w^T x_i + w_0) > 0 \quad (1.142)$$

Şekil 1.32. iki ayrılabilir veri kümesini ayırma hiper düzlemleri (A) ile göstermektedir. Mümkün bir çok ayırma hiper düzleminin mevcut olduğu açıkça görülmektedir. En büyük marjın sınıflandırıcı, verileri ayıran A hiper düzleminin iki tarafındaki paralel hiper düzlemlere olan uzaklık olan marjini en büyük değere getirecek şekilde hiper düzlemi belirler. Varsayım, marjinin büyüklüğü nispetinde ayırma hiper

düzlemi ile tanımlanan doğrusal sınıflandırıcının genelleştirme hatasının da o kadar küçük olacağıdır.

Perceptron kuralının bir çeşidi $b > 0$ marjini ortaya koymakta ve çözümü (1.143)'teki şekilde aramaktadır.

$$y_i(w^T x_i + w_0) \geq b \quad (1.143)$$

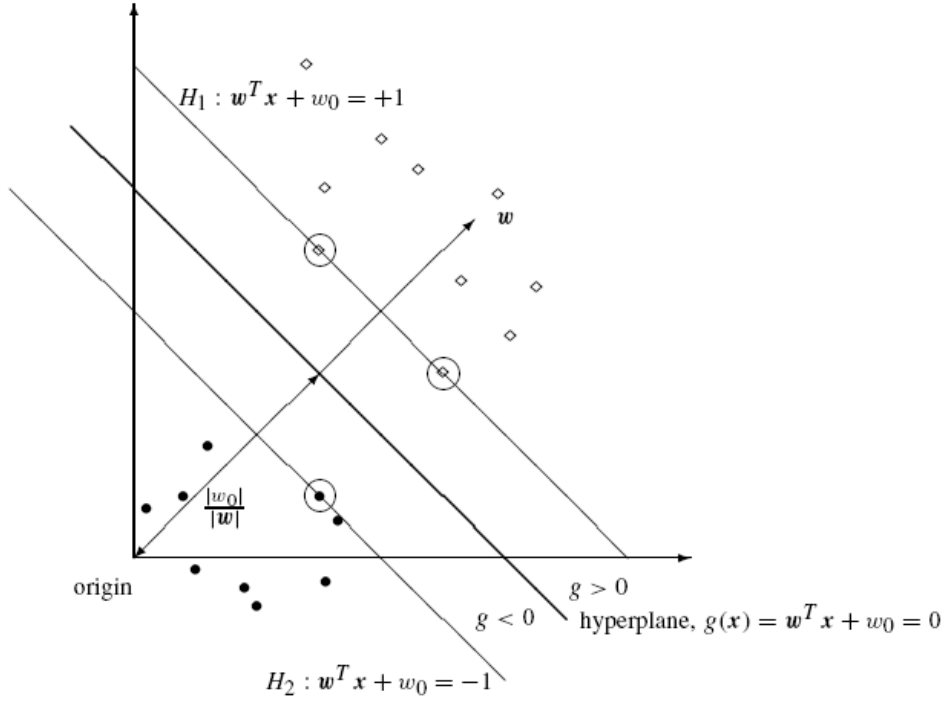
Perceptron algoritması bütün x_i noktalarının ayırma hiper düzleminden $b/|w|$ değerinden daha büyük uzaklıklarda olduğu bir çözüm elde etmektedir. b , w_0 ve w 'nin ölçeklenmesi koşulunu değiştirilmemiş kılar ve koşul sağlanmaya devam eder. Bu yüzden genelleştirmeden ödün vermeden $b=1$ değeri alınabilir ve kuralsal hiper düzlemler olarak adlandırılan hiper düzlemler tanımlanır:

$$\begin{aligned} H_1 : w^T x + w_0 &= +1 \quad \text{ve} \quad H_2 : w^T x + w_0 = -1 \\ w^T x_i + w_0 &\geq +1, \quad y_i = +1 \text{ için} \\ w^T x_i + w_0 &\leq -1, \quad y_i = -1 \text{ için} \end{aligned} \quad (1.144)$$

Bu iki hiper düzlemin her birinin ayırma hiper düzlemi $g(x) = 0$ ile arasındaki uzaklık $1/|w|$ 'dur ve marjin olarak adlandırılır. Şekil 1.33, ayırma hiper düzlemini ve her iki veri kümesi için kuralsal hiper düzlemleri göstermektedir. Kuralsal hiper düzlemlerin üzerine düşen, Şekil 1.33.'te yuvarlak içine alınan noktalar destek vektörler olarak adlandırılır. Şekilde H_1 ve H_2 kuralsal hiper düzlemlerdir. Marjin, ayırma hiper düzlemi ile en yakın noktadan geçen hiper düzlem arasındaki dik uzaklıktır.

Bu yüzden, marjini en büyük değere getirmek, $|w|$ 'yu (1.145) koşuluna göre en küçük değere getirmek demektir.

$$C1: \quad y_i(w^T x_i + w_0) \geq 1, \quad i = 1, \dots, n \quad (1.145)$$



Şekil 1.33. Kural sal hiper düzlemlerle marjinin grafiksel gösterimi

Eşitlik ve eşitsizlik sınırlamalarına sahip optimizasyon problemlerine standart bir yaklaşım, bizi aşağıda verilen L_p amaç fonksiyonunun birincil biçimine götüren Lagrange biçimciliktir.

$$L_p = \frac{1}{2} w^T w - \sum_{i=1}^n \alpha_i (y_i (w^T x_i + w_0) - 1) \quad (1.146)$$

Burada $\{\alpha_i, i = 1, \dots, n; \alpha_i \geq 0\}$ Lagrange çarpıcılarıdır. Birincil parametreler w ve w_0 iken parametre sayısı $p+1$ 'dir (p =öznitelik uzayının boyutu).

(1.145) koşulu altında $w^T w$ değerinin minimizasyonu problemine çözüm, L_p fonksiyonunun eyer noktasının belirlenmesi şeklindedir. L_p noktası, w ve w_0 'a göre minimize edilirken α_i 'ye göre de en büyük değerine getirilir. L_p 'yi w ve w_0 'a göre ayırmak ve sıfıra eşitlemekle (1.147) sonucu elde edilir.

$$\begin{aligned} \sum_{i=1}^n \alpha_i y_i &= 0 \\ w &= \sum_{i=1}^n \alpha_i y_i x_i \end{aligned} \quad (1.147)$$

(1.147)'yi (1.146)'de yerine yazarsak Lagrangian'ın ikili biçimini verir.

$$L_D = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j x_i^T x_j \quad (1.148)$$

Bu, (1.149) koşuluna bağlı olarak α_i 'ye göre en büyük değerine getirilmelidir.

$$\alpha_i \geq 0 \quad \sum_{i=1}^n \alpha_i y_i = 0 \quad (1.149)$$

Verilen sınırlamalara göre eyer noktasını bulmada yaygın olarak kullanılan algoritmalar dahili nokta algoritması, sıralı minimal optimizasyon ve dağınık açgözlü matris yaklaşımıdır. Daha fazla ayrıntıya sahip olmak için çekirdek yöntemlerin geniş biçimde anlatıldığı kitaplarla optimizasyon kaynaklarından yararlanılabilir. İkili biçimin önemi, optimizasyon ölçütünü x_i örüntülerinin iç çarpımı olarak ifade etmesindendir. Bu, anahtar kavram olup, doğrusal olmayan destek vektör makineleri için önem arz etmektedir. İkili değişkenler Lagrange çarpıcılarıdır (α_i) ve bu yüzden parametre sayısı örüntü sayısına (n) eşittir.

1.4.3.2. Karush-Kuhn-Tucker Koşulları

Yukarıda birincil problem, sayısal olarak daha kolay çözülebilecek ikili biçime dönüştürüldü. Kuhn-Tucker koşulları, amaç fonksiyonunun eşitlik ve eşitsizlik koşullarına göre en küçük değere indirgendiğinde sağlanan, gerekli ve yeterli koşulları sunmaktadır. Amaç fonksiyonunun birincil biçiminde bunlar (1.150)'de belirtilen şekilde ifade edilir.

$$\begin{aligned} \frac{\partial L_p}{\partial w} &= w - \sum_{i=1}^n \alpha_i y_i x_i = 0 & \frac{\partial L_p}{\partial w_0} &= - \sum_{i=1}^n \alpha_i y_i = 0 \\ y_i (x_i^T w + w_0) - 1 &\geq 0 \\ \alpha_i &\geq 0 \\ \alpha_i (y_i (x_i^T w + w_0) - 1) &= 0 \end{aligned} \quad (1.150)$$

Özellikle Karush-Kuhn-Tucker tamamlayıcılık koşulu olarak bilinen $\alpha_i (y_i (x_i^T w + w_0) - 1) = 0$ koşulu, etkin sınırlamalar için $\alpha_i \geq 0$, etkin olmayan sınırlamalar için $\alpha_i = 0$ olduğunu gösterir. Etkin sınırlamalar için Lagrange çarpıcı, L_p 'nin en uygun değerinin özel sınırlamalara olan duyarlılığını gösterir. Sıfırdan farklı Lagrange çarpıcıya sahip bu veri noktaları kuralsal hiper düzlemlerin üzerine düşer. Bunlar destek vektörler olarak bilinen veri kümesindeki en fazla bilgi verici noktalardır. Eğer diğer $\alpha_i=0$ koşulunu sağlayan örüntülerden herhangi birinin yeri değiştirilirse (dıştaki kuralsal hiper düzlemlerden birini kesmemek koşuluyla) ayırma hiper düzleminin çözümünde hiç bir etki yapılmamış olacaktır.

1.4.3.3. SVM ile Sınıflandırma

Sınırlandırılmış optimizasyon problemini ikili biçim ile yeniden biçimlendirmek sayısal ikinci dereceden programlama çözümlerini kullanmamızı sağlar. Lagrange çarpıcılar (α_i) elde edildiğinde w_0 değeri herhangi bir destek vektörünü veya bütün destek vektörlerinin ortalamasını (1.151) kullanarak (1.152) eşitliğinden bulunabilir.

$$n_{SV} w_0 + w^T \sum_{i \in SV} x_i = \sum_{i \in SV} y_i \quad (1.151)$$

(1.151)'de n_{SV} , destek vektörlerinin sayısı olup toplamlar bütün destek vektörleri kümesi SV üzerinde yapılmaktadır.

$$\alpha_i (y_i (x_i^T w + w_0) - 1) = 0 \quad (1.152)$$

(1.152)'de kullanılan w için çözüm, diğer örüntüler için $\alpha_i=0$ olduğundan (1.153)'de verilmektedir:

$$w = \sum_{i \in SV} \alpha_i y_i x_i \quad (1.153)$$

Böylece, destek vektörleri ayırma hiper düzlemini tanımlar.

Yeni bir x örüntüsü, $w^T x + w_0$ ifadesinin işaretine göre sınıflandırılacaktır.

w ve w_0 'ı (1.151)'de yerine koyarsak aşağıdaki x 'i w_1 sınıfına dahil eden doğrusal sınıflandırıcıyı elde ederiz.

$$\sum_{i \in SV} \alpha_i y_i x_i^T x - \frac{1}{n_{SV}} \sum_{i \in SV} \sum_{j \in SV} \alpha_i y_i x_i^T x_j + \frac{1}{n_{SV}} \sum_{i \in SV} y_i > 0 \quad (1.154)$$

1.4.3.4. Doğrusal Olarak Ayrılamayan Veriler

Bir çok sahici problemde sınıfları birbirinden ayıran doğrusal sınır bulunmayacak ve en uygun ayırma hiper düzlemini araştırmak anlamsız olacaktır. Verileri sınıfların birbirinden doğrusal olarak ayrılabilirdiği çok boyutlu öznitelik uzayına dönüştürmek için karmaşık öznitelik vektörlerini $\phi(x)$ kullansak bile bu, verilerin aşırı yüklenmesine ve bu yüzden düşük genelleştirme yeteneğine sahip olmasına neden olacaktır.

Bununla birlikte yukarıdaki düşünceleri, ayrılamayan verileri işlemek amacıyla sınırlamaları gevşeterek genişletebiliriz. Bunu pozitif pay değişkenini (ξ_i , $i=1, \dots, n$) sınırlamalara (1.155) sonucunu üretecek şekilde dahil ederek gerçekleştiriyoruz:

$$\begin{aligned} w^T x_i + w_0 &\geq +1 - \xi_i, & y_i = +1 \text{ için} \\ w^T x_i + w_0 &\leq -1 + \xi_i, & y_i = -1 \text{ için} \\ \xi_i &\geq 0, & i = 1, \dots, n \end{aligned} \quad (1.155)$$

Ayırma hiper düzlemi tarafından yanlış sınıflandırılan bir nokta için $\xi_i > 1$ olmalıdır. Ayrılamamazlıktan dolayı ek maliyeti hesaba katmanın uygun yolu, $w^T w/2$ ifadesini $w^T w/2 + C \sum_i \xi_i$ ile değiştirerek maliyet fonksiyonuna fazladan maliyet terimi eklemektir (L1 yumuşak-marjin). Burada C , düzenlileştirme parametresidir. $C \sum_i \xi_i$ terimi yanlış sınıflandırma miktarının ölçümü olarak düşünülebilir. C değeri düşük oldukça aykırı değerlerin cezaları küçük olacak ve marjin daha yumuşak olacaktır. Doğrusal durum ise $C \rightarrow \infty$ iken geçerli olacaktır. [38] Diğer ceza terimleri de mümkündür, örneğin $C \sum_i \xi_i^2$ gibi (L2 yumuşak-marjin).

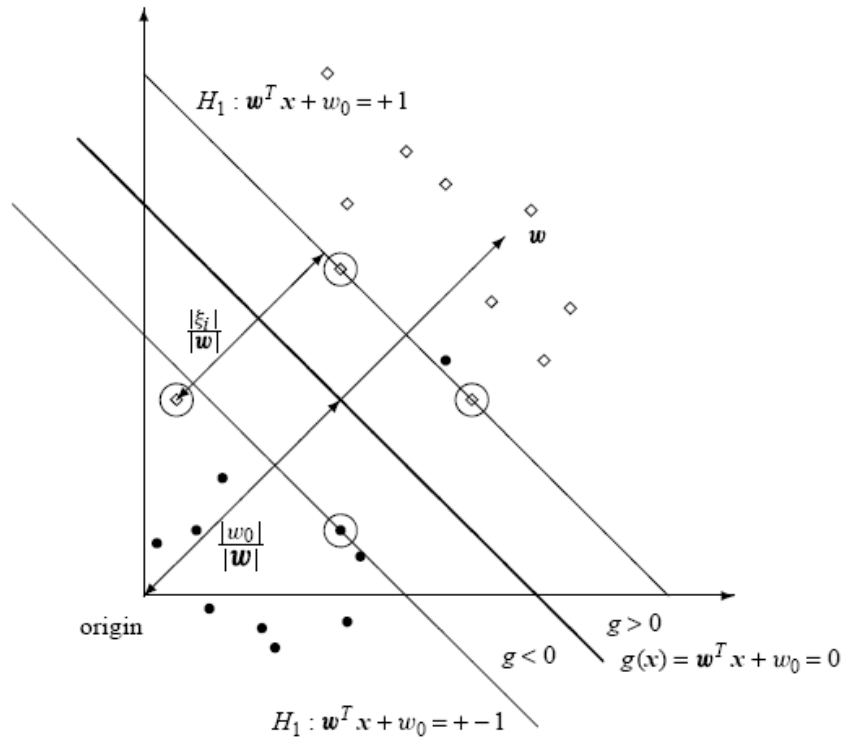
Böylece (1.155)'daki sınırlamalara göre (1.156) ifadesini minimize ediyoruz.

$$\frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_i \xi_i \quad (1.156)$$

Lagrangian'ın birincil biçimi (1.157)'deki şekle dönüşmüş olmaktadır.

$$L_p = \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_i \xi_i - \sum_{i=1}^n \alpha_i (y_i (\mathbf{w}^T \mathbf{x}_i + w_0) - 1 + \xi_i) - \sum_{i=1}^n r_i \xi_i \quad (1.157)$$

(1.157)'deki $\alpha_i \geq 0$ ve $r_i \geq 0$ Lagrange çarpıcılarıdır; r_i , ξ_i 'nin pozitifliğinden emin olunması için eklenmiştir.



Şekil 1.34. Ayrılamayan veriler için doğrusal ayırma hiper düzlemi

$\mathbf{w}$ ve w_0 'a göre ayırmak (1.158) sonucunu çıkartır.

$$\sum_{i=1}^n \alpha_i y_i = 0$$

$$w = \sum_{i=1}^n \alpha_i y_i x_i$$
(1.158)

ξ_i 'ye göre ayırarak (1.159) elde edilir:

$$C - \alpha_i - r_i = 0$$
(1.159)

Yukarıdaki sonuçlar birincil biçimde yerine konulduğunda, en büyük marjın sınıflandırıcı ile aynı biçimde olan Lagrangian'ın ikili biçimi elde edilir.

$$L_D = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j x_i^T x_j$$
(1.160)

Bu, (1.161)'daki denklemlere bağlı α_i 'ye göre en büyük değere getirilmiştir:

$$\sum_{i=1}^n \alpha_i y_i = 0$$

$$0 \leq \alpha_i \leq C$$
(1.161)

Son koşul (1.159)'ün sonucudur ve $r_i \geq 0$ 'dır. Böylece, en büyük değere çıkartma problemindeki tek değişiklik α_i 'nin üst sınırında olmaktadır.

Karush-Kuhn-Tucker tamamlayıcılık koşulları (1.162)'de verilmektedir.

$$\alpha_i (y_i (x_i^T w + w_0) - 1 + \xi_i) = 0$$

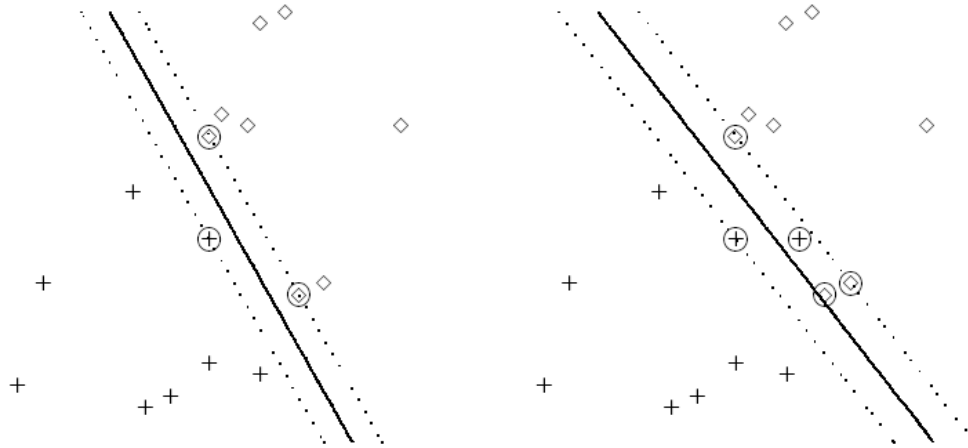
$$r_i \xi_i = (C - \alpha_i) \xi_i = 0$$
(1.162)

$\alpha_i > 0$ örüntüleri destek vektörleri olarak adlandırılır. $0 < \alpha_i < C$ koşulunu sağlayan, ayırma hiper düzleminin $1/|w|$ uzaklığındaki kural sal hiper düzlemlerden birinin üzerine düşen $\xi_i = 0$ değerine sahip olmalıdır. Sıfırdan farklı pozitif pay değişkeni sadece $\alpha_i = C$ durumlarında ortaya çıkar. Bu durumda, x_i noktaları eğer $\xi_i > 1$ ise yanlış

sınıflandırılmışlardır. Eğer $\xi_i < 1$ ise, doğru sınıflandırılmışlardır, fakat ayırma hiper düzlemine $1/|w|$ 'dan daha yakındırlar. Ayrılabilir durumda olduğu gibi, w_0 değeri yukarıdaki ilk koşul ve herhangi bir destek vektörü kullanılarak ya da $\xi_i = 0$, $0 < \alpha_i < C$ olan örneklerin toplanmasıyla belirlenir ve w , (1.153) ile verilir. Bu (1.163) eşitliğini oluşturur.

$$w_0 = \frac{1}{N_{\widetilde{SV}}} \left\{ \sum_{i \in SV} y_i - \sum_{i \in SV, j \in \widetilde{SV}} \alpha_i y_i x_i^T x_j \right\} \quad (1.163)$$

(1.163)'de SV , $0 < \alpha_i < C$ koşulunu sağlayan α_i değerleri ile ilişkili destek vektörleri kümesi, $\widetilde{SV}$ de $0 < \alpha_i < C$ koşulunu sağlayan $N_{\widetilde{SV}}$ adet destek vektörünün kümesidir.



Şekil 1.35. Doğrusal olarak ayrılabilir veriler (solda) ve ayrılamayan veriler (sağda)

Şekil 1.35., doğrusal olarak ayrılabilen ve ayrılamayan veriler için en uygun ayırma hiper düzlemini göstermektedir. Destek vektörleri yuvarlak içine alınmıştır. Destek vektörü olmayan bütün noktalar marjın şeridinin dışına düşmektedir. Sağdaki resimde destek vektörlerinden biri yanlış sınıflandırılmıştır.

Tek serbest parametre olan düzenlilik parametresi C 'dir. C 'yi belirli bir aralıkta değiştirerek ve sınıflandırıcının performansı gözlenerek, ya da çapraz geçerlilik sınaması uygulanarak uygun C değeri belirlenebilir.

1.4.3.5. Çok Sınıflı Destek Vektör Makineleri

Destek vektör makineleri, bire karşı bir veya bire karşı hepsi yaklaşımları ile ikili sınıflandırıcıları kullanarak ya da C adet doğrusal ayırma fonksiyonunu eş zamanlı olarak oluşturarak çok sınıflı problemlere de uygulanabilir.

(1.164)'da verilen doğrusal bir sınıflandırıcıyı ele alalım.

$$g_k(x) = (w^k)^T x + w_0^k, \quad k = 1, \dots, C \quad (1.164)$$

$\{(w^k, w_0^k), k = 1, \dots, C\}$ için bir çözüm aranmaktadır. Öyle ki karar kuralının, eğer (1.165) kuralı eğitim verilerini hatasız ayırabiliyorsa x 'i w_i sınıfına ataması gerekir.

$$g_i(x) = \max_j g_j(x) \quad (1.165)$$

$\{(w^k, w_0^k), k = 1, \dots, C\}$ için (1.166) koşulunu sağlayan çözümler vardır:

$$(w^k)^T x + w_0^k - ((w^j)^T x + w_0^j) \geq 1, \quad \text{bütün } x \in w_k \text{ ve } j \neq k \text{ için} \quad (1.166)$$

Bu, bütün sınıf çiftlerinin ayrılabilir olduğu anlamına gelir. Eğer bir çözüm mevcutsa en küçük çözümü araştıracağız. Eğer eğitim verileri ayrılamıyorsa, pozitif pay değişkenleri ortaya atılır ve (1.167) ifadesi minimize edilir.

$$L = \sum_{k=1}^C (w^k)^T w^k + C \sum_{i=1}^n \xi_i, \quad \text{bütün } x_i \in w_k \text{ ve } j \neq k \text{ için} \quad (1.167)$$

$$(w^k)^T x_i + w_0^k - ((w^j)^T x_i + w_0^j) \geq 1 - \xi_i \quad \text{sınırlaması altında}$$

Eşitsizlik sınırlamasına bağlı olarak L değerinin minimize edilmesi işlemi iki sınıflıdaki durum ile aynıdır.

1.4.3.6. Doğrusal Olmayan Destek Vektör Makineleri

Destek vektör algoritması dönüştürülmüş öznitelik uzayı $\phi(x)$ 'e doğrusal olmayan bir ϕ fonksiyonu ile uygulanabilir. Aslında bu, birçok örüntü sınıflandırma yönteminin temelindeki ilkedir. Bu yaklaşımı SVM bağlamında inceleyeceğiz.

İkili sınıflandırma problemi için (1.168) biçimindeki ayırma fonksiyonu araştırılmaktadır.

$$g(x) = w^T \phi(x) + w_0 \quad (1.168)$$

Karar kuralı (1.169)'da gösterilmektedir.

$$w^T \phi(x) + w_0 \begin{cases} > 0 \\ < 0 \end{cases} \Rightarrow x \in \begin{cases} w_1, \text{ karşı düşen sayısal değer } y_i = +1 \\ w_2, \text{ karşı düşen sayısal değer } y_i = -1 \end{cases} \quad (1.169)$$

SVM yöntemi en büyük marjin çözümünü Lagrangian'ı en büyük değerine getirerek belirler. Lagrangian'ın ikili biçimi (1.170) şeklini alır.

$$L_D = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \phi^T(x_i) \phi(x_j) \quad (1.170)$$

Burada $y_i = \pm 1$, $i = 1, \dots, n$, sınıf belirteç değerleri; α_i , $i = 1, \dots, n$ de (1.171) koşullarını sağlayan Lagrange çarpıcılarıdır.

$$\begin{aligned} 0 &\leq \alpha_i \leq C \\ \sum_{i=1}^n \alpha_i y_i &= 0 \end{aligned} \quad (1.171)$$

w 'nın çözümü: $w = \sum_{i \in SV} \alpha_i y_i \phi(x_i)$ ve yeni veri örneği x 'in sınıflandırması $g(x)$

fonksiyonunun işaretine göredir.

$$g(x) = \sum_{i \in SV} \alpha_i y_i \phi^T(x_i) \phi(x) + w_0 \quad (1.172)$$

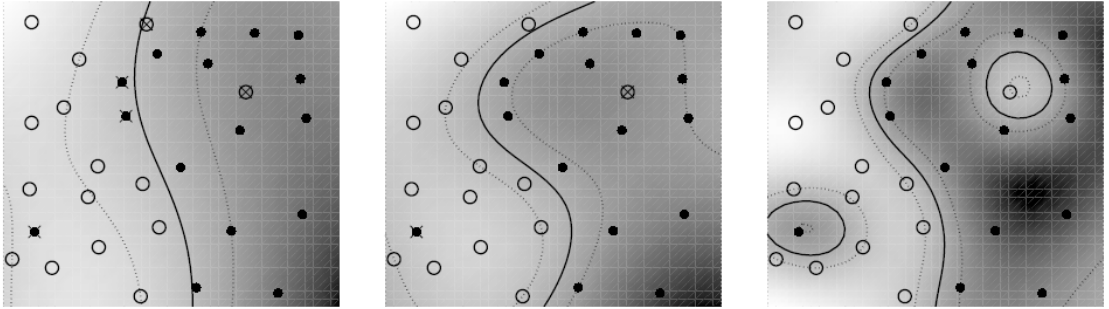
$$w_0 = \frac{1}{N_{\widetilde{SV}}} \left\{ \sum_{i \in \widetilde{SV}} y_i - \sum_{i \in SV, j \in \widetilde{SV}} \alpha_i y_i \phi^T(x_i) \phi(x_j) \right\} \quad (1.173)$$

L_D 'nin optimizasyonu ve onu izleyen örneğin sınıflandırılması sadece çekirdek fonksiyonu ile değiştirilebilecek olan dönüştürülmüş uzaydaki öznitelik vektörleri arasındaki iç çarpıma bağlıdır.

Böylece, $\phi(x)$ dönüşümünün açıkça hesaplanmasından kaçınarak iç çarpımı $K(x,y)$ ile değiştirebiliriz. Ayırma fonksiyonu (1.174) biçimine gelir.

$$g(x) = \sum_{i \in SV} \alpha_i y_i K(x_i, x) + w_0 \quad (1.174)$$

Çekirdek gösterimin avantajı, eğitim algoritmasında sadece K 'nın kullanımına ihtiyaç duyarak ϕ 'nin açıkça bilinmesine gerek duyulmamasıdır. Bazı durumlarda (Gaussian çekirdek gibi) öznitelik uzayı sonsuz boyutta olur ve bu yüzden çekirdeğin kullanımı daha verimli olur [39].



Şekil 1.36. Yumuşak marjın SVM kullanılarak çözülen ikili sınıflandırma problemi örneği

Şekil 1.36.'da Gaussian çekirdeklerin kullanıldığı yumuşak marjın sınıflandırma problemi ele alınmıştır. Soldan sağa doğru çekirdeğin genişliği küçültülmüştür. Büyük genişlikler için karar sınırının doğrusala yakın olduğuna ve verilerin hatasız sınıflandırılmadığına dikkat edin. Şekilde kesiksiz çizgiler karar sınırlarını, kesikli çizgiler de marjının kenarlarını göstermektedir.

1.4.3.7. SVM'nin Avantajları

Destek vektör makinelerinin, başarılı bir sınıflandırıcı olan çok katmanlı sinir ağlarına göre avantajları şöyledir:

1. Genelleştirme yeteneğinin maksimizasyonu: Çok katmanlı bir sinir ağı sınıflandırıcının eğitiminde çıkışlar ve istenen eğitim çıkışları arasındaki karelerin toplam hatası minimize edilir. Böylece, sınıfın sınırları başlangıç ağırlıkları değiştiğinde değişir. Bu yüzden genelleştirme yeteneği de etkilenir. Özellikle eğitim verileri az ve doğrusal olarak ayrılabilirdiği durumlarda genelleştirme yeteneği oldukça kötüye gider. Fakat destek vektör makineleri marjini en büyük değere çıkartmaya çalıştığı için genelleştirme yeteneği böyle bir koşulda bile çok fazla bozulmaz [40].
2. Yerel minimum olmayışı: Çok katmanlı sinir ağı sınıflandırıcı, bir çok yerel minimum noktasına sahip olabilir. Bu durumdan nasıl kurtulacağı hakkında çok sayıda araştırma yapılmaktadır. Fakat, destek vektör makineleri ikinci dereceden programlama problemi şeklinde gösterilebildiğinden sadece bir tane global optimum sonucu vardır.
3. Aykırı değerlere dayanıklılık: Çok katmanlı sinir ağı sınıflandırıcılar, karelerin toplamı hatasını kullandığından aykırı değerlerden kolayca etkilenebilirler. Aykırı değerlerin etkisini önlemek için aykırı değerler eğitimden önce bir şekilde elenmelidir. Destek vektör makinelerinde C marjin parametresi yanlış sınıflandırma hatasını kontrol eder. Eğer C büyük değerlere ayarlanırsa yanlış sınıflandırma bastırılabilir. Eğer küçük değer kullanılırsa, genel kümeden uzaktaki eğitim verileri yanlış sınıflandırılabilir. Böylece C değerinin uygun seçimi ile aykırı değerleri bastırabiliriz.

1.4.3.8. SVM'nin Dezavantajları

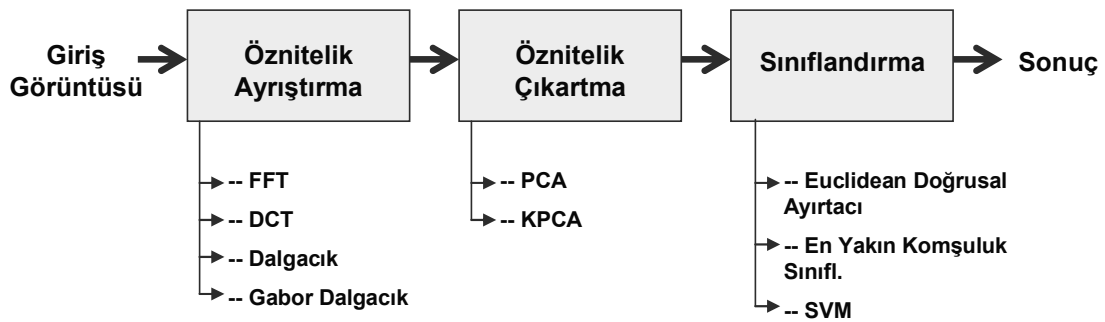
1. Çok sınıflı probleme genişletilmesi: Çok katmanlı sinir ağlarından farklı olarak destek vektör makineleri, doğrudan karar fonksiyonunu kullanır. Bu yüzden çok sınıflı problemlere genişletilmesi basit olmayıp bir kaç formülasyona ihtiyaç duyar.

2. Uzun eğitim zamanı: Destek vektör makinesinin eğitimi, ilgili ikili problemin çözümü ile gerçekleştiğinden değişkenlerin sayısı eğitim verilerinin sayısına eşittir. Bundan dolayı, çok sayıdaki eğitim verisi için ikili problemin çözümü bellek büyüklüğü ve eğitim zamanı açısından zor olmaktadır.
3. Parametrelerin seçimi: Destek vektör makinelerinin eğitiminde uygun bir çekirdek, onun parametreleri ve C marjın parametresini seçmemiz gerekir. Verilen bir problem için en uygun parametrelerin seçimi model seçimi olarak adlandırılır. Bu, sinir ağı sınıflandırıcıdaki durumla aynıdır. Şöyle ki, gizli birimlerin sayısının, başlangıç değerlerinin, ağırlıkların vs. ayarlanması gerekir. Destek vektör makinelerinde model seçimi, eğitim verileri tekrar tekrar kullanılarak genelleştirme yeteneğinin hesaplanmasıyla belirlenir. Fakat bu zaman alan bir işlem olduğundan genelleştirme yeteneğinin tahmini için başka yöntemler ileri sürülmüştür [29].

2. YAPILAN ÇALIŞMALAR, BULGULAR VE İRDELEME

2.1. Giriş

Çalışmada, genel amaçlı görünüşe dayalı bir öğrenme sistemi gerçekleştirilmeye çalışılmıştır. Bu öğrenme sisteminde giriş olarak ham görüntü alınmaktadır. Bu ham görüntü üzerinde öncelikle FFT, DCT, ortogonal dalgacık ve Gabor dalgacık dönüşümleri uygulanarak veriler zaman domeninden tayfsal domene aktarılmıştır. Böylece veriyi en iyi niteleyebilecek frekans ya da yerleşmiş frekans bilgileri elde edilmiş olup istenilen bilgilerin alınması, istenmeyen gürültülerin kaldırılması sağlanmıştır. Sonra, nesnelerin birbirinden daha az öz nitelikle daha iyi ayırt edilebilmesi için öz nitelik çıkartma yöntemleri uygulanmıştır. Burada, temel bileşen analizi ve çekirdek temel bileşen analizi yöntemleri üzerinde durulmuştur. Bu yöntemlerde, verilen eğitim örneklerinin öz nitelik uzayında birbirlerinden mümkün olduğunca uzak yerleşmesini sağlayacak bir dönüşüm gerçekleştirilir. Böylece verilen eğitim örnekleri birbirlerinden en iyi şekilde ayırt edilebilecektir. Son aşamada da her bir görüntünün öz nitelik çıkartma algoritmaları ile elde edilen öz nitelik vektörleri sınıflandırma algoritması ile sınıflandırılır. Sınıflandırmada Euclidean doğrusal ayırtıcı (en küçük uzaklık) ile en yakın komşuluk sınıflandırıcının yanında günümüzde popüler olan destek vektör makineleri de uygulanmıştır.



Şekil 2.1. Görünüşe dayalı tanıma sisteminin temel işlem adımları

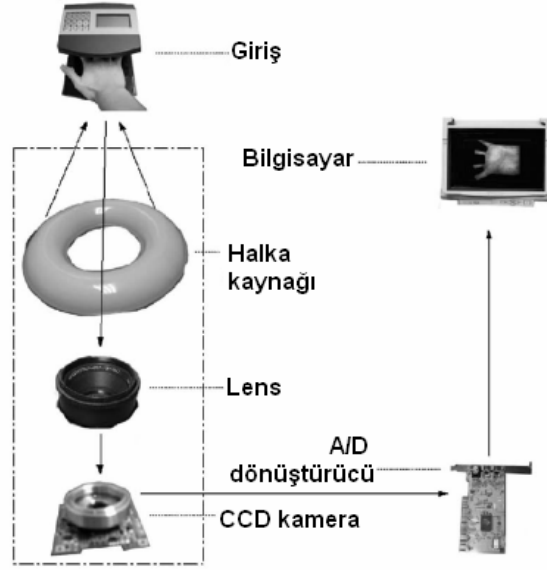
Görünüşe dayalı sınıflandırma sistemimizde avuç izi görüntülerini ele aldık. Bu veriler üzerinde yöntemimizi uyguladık ve başarılı sayılabilecek sonuçlar elde ettik.

2.2. Avuç İzine Dayalı İnsan Tanıma

Avuç izine dayalı kişilerin ayırt edilmesi son yıllarda araştırmacıların etkin biçimde ilgilendiği konulardan biridir. Bu konu ile ilgili yapısal [55, 56] ve görünüşe dayalı [46, 47] değişik yaklaşımlar ileri sürülmüştür. Yapısal yaklaşımlar görüntüdeki değişik özelliklerden yararlanıp o probleme has öznitelikler çıkartmaktadır. Görünüşe dayalı yaklaşımlar ise resmi bir bütün olarak ele alıp bütün problemler için aynı algoritmayı kullanabilmektedir. Bunlardan bizim de üzerinde çalıştığımız görünüşe dayalı yaklaşımlar diğerlerine göre daha iyi sonuç vermektedir.

Sunulan yöntemin performansı, PolyU-I [63] ve PolyU-II [62] avuç içi veritabanları üzerinde değerlendirilmiştir. PolyU-I avuç izi veritabanındaki örnekler CCD kamera ile alınmıştır. Veritabanı, 100 değişik avuçtan alınan 600 görüntüyü içermektedir. Her bir avuçtan alınan 6 örneğin ilk üçü bir oturumda, diğer üçü ise başka bir oturumda alınmıştır. İkinci oturum ile ilk oturum arasındaki süre yaklaşık olarak iki aydır. PolyU-II avuç izi veritabanındaki görüntüler de benzer şekilde elde edilmiştir. Bu veritabanında 386 değişik avuçtan alınan 7752 adet görüntü bulunmaktadır. Her bir avuca ait ilk oturumda 10, ikinci oturumda da 10 adet görüntü bulunmaktadır. İki oturum arasındaki süre 69 gündür. Bütün görüntüler 384x284 boyutlarında 75 dpi çözünürlükte elde edilmiştir. Ayrıca, ışık kaynağı ve CCD kameranın odak ayarı değiştirilerek ilk ve ikinci oturumlarda alınan görüntülerin farklı avuç izi aygıtlarından alındığı hissi verilmiştir.

Avuç izi görüntü yakalama sisteminde kararlı bir avuç izi görüntüsü elde etmek için, yarı kapalı bir ortam oluşturmak amacıyla bir mahfaza ve bir kapak, ayrıca avuç izi görüntüsü alınırken ışık koşullarının değişmemesini sağlayacak bir halka kullanılmaktadır. Platformdaki kontrol noktaları olarak hizmet eden altı kanca, kullanıcının ellerini düzgün yerleştirmesini sağlamaktadır. A/D dönüştürücü CCD kamera ile alınmış görüntüleri doğrudan bilgisayara iletir. Şekil 2.2., bu avuç izi görüntüsü üreten sistemin şematik diyagramını göstermektedir.



Şekil 2.2. Avuç izi yakalama sisteminin tasarım ilkesi

Avuç izi tanıma yöntemimizde öncelikle her bir avuç içi görüntüsünden ilgilendiğimiz 128x128'lik bölgenin çıkartılması gerçekleştirilir. Sonra öznitelik ayrıştırma, öznitelik çıkartma ve sınıflandırma işlemleri ile tanıma süreci tamamlanır.

2.2.1. Önışlemler

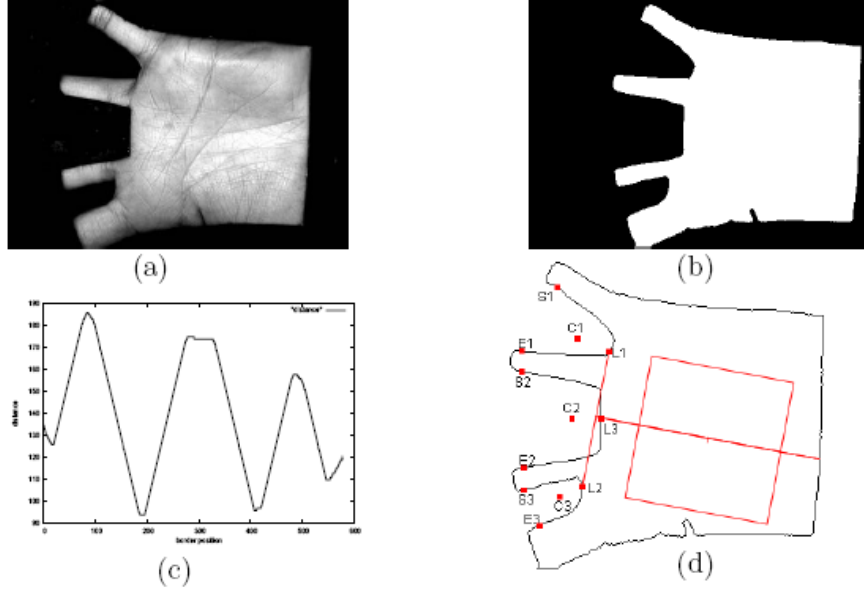
Avuç içi görüntülerinden ilgilenilen bölgenin seçiminde 2 çeşit yöntem uygulanmıştır. Her iki bölge seçiminde de öncelikle avuç görüntüsü eşik değeri ile ikili görüntüye çevrilmektedir. Eşik değerinin seçimi için uyguladığımız yöntem, bulanıklığın minimizasyonu [57] yöntemidir. İkili görüntü üzerinde sınır takibi algoritması ile parmak aralarındaki sınırlar bulunur. Sonraki adımlar ise biraz farklılık göstermektedir.

1. bölge seçimi yöntemi [44] işlem adımları şöyledir:

- Orijinal gri seviye görüntü ikili görüntüye çevrildikten sonra parmak arasındaki boşluklar görüntünün sol tarafından olan uzaklıklar kullanılarak belirlenir. Sonra işaretteki en uç değerler başlangıç noktası (Sx_i , Sy_i) ve bitiş noktası (Ex_i , Ey_i) olarak işaretlenir.
- Parmak aralarının ağırlık merkezi (Cx_i , Cy_i) hesaplanır. Ağırlık merkezi ile başlangıç ve bitiş noktalarının ortasından geçen doğrunun parmağı kestiği yer bulunur. Böylece üç nokta (L_1 , L_2 , L_3) elde edilmiş olur.

- L_1 ve L_2 noktalarından geçen doğru çizilir ve bu doğru Y eksenini oluşturacak şekilde görüntü döndürülür. Yeni oluşturulan Y eksenine dik ve L_3 'ten geçen doğrunun orta noktası, ilgilenilen bölgenin merkezine gelecek şekilde 128x128'lik görüntü seçilir.

Şekil 2.3.'te 1. bölge seçimi yönteminin temel işlem adımları gösterilmiştir.



Şekil 2.3. 1. Bölge seçme yönteminin temel aşamaları (a) Orijinal görüntü, (b) İkili görüntü, (c) Görüntünün sol tarafından olan uzaklıkla oluşturulan işaret, (d) Avucun orta bölgesi olarak üretilen avuç izi görüntüsünün çıkartılması

2. Bölge seçimi yönteminin [43] işlem adımları ise şu şekildedir:

- Görüntünün ikili karşılığı bulunduğundan sonra sınır izleme ile parmak aralıkları bulunur.
- İki parmak arasının (işaret parmağı ile ortanca parmak ve yüzük parmağı ile serçe parmak arası) tanjantı bulunur. Bunun için (x_1, y_1) ve (x_2, y_2) sırasıyla (F_1x_j, F_1y_j) ve (F_2x_j, F_2y_j) parmak aralıklarındaki herhangi iki nokta olsunlar. Bu durumda $F_1y_j \leq mF_1x_j + c$ bütün i ve j ler için sağlanıyorsa $y=mx+c$ doğrusu iki parmak arasının tanjantı olacaktır.
- (x_1, y_1) ve (x_2, y_2) noktaları bir doğru ile birleştirilir ve bu doğruya göre döndürme işlemi gerçekleştirilir. Döndürme işleminden sonra bu doğrunun orta noktasından

dik bir doğru daha çizilir ve orta noktası merkeze gelecek şekilde seçilebilecek en büyük alan olan 128x128'lik bir alan avuç izi görüntüsü olarak belirlenir.

Avuç görüntülerinden ilgilendiğimiz bölge olan avuç izi görüntüleri çıkartıldıktan sonra görüntü üzerinde sıfır ortalamalı birim varyans sağlanmaktadır. Böylece görüntümüz sonraki aşamalar için en uygun hale getirilmiş olmaktadır. (2.1) eşitliğinde N boyut sayısı, μ x dizisinin ortalaması, σ^2 de varyansı belirtmektedir.

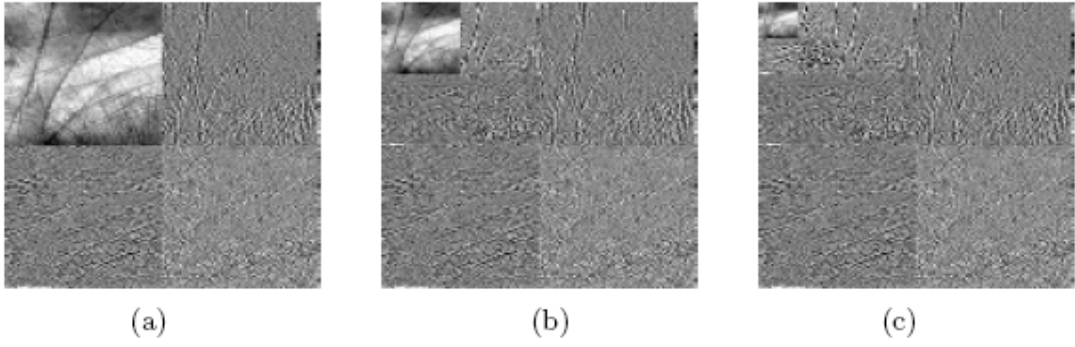
$$x'_i = \frac{x_i - \mu}{\sigma^2}, \quad \text{bütün } i \in N \text{ için} \quad (2.1)$$

Önişlem olarak histogram eşitleme ve kenar algılama yöntemleri de denenmiş olup negatif bir etki oluşturduğundan kaldırılmıştır.

2.2.2. Öznitelik Ayırıştırma

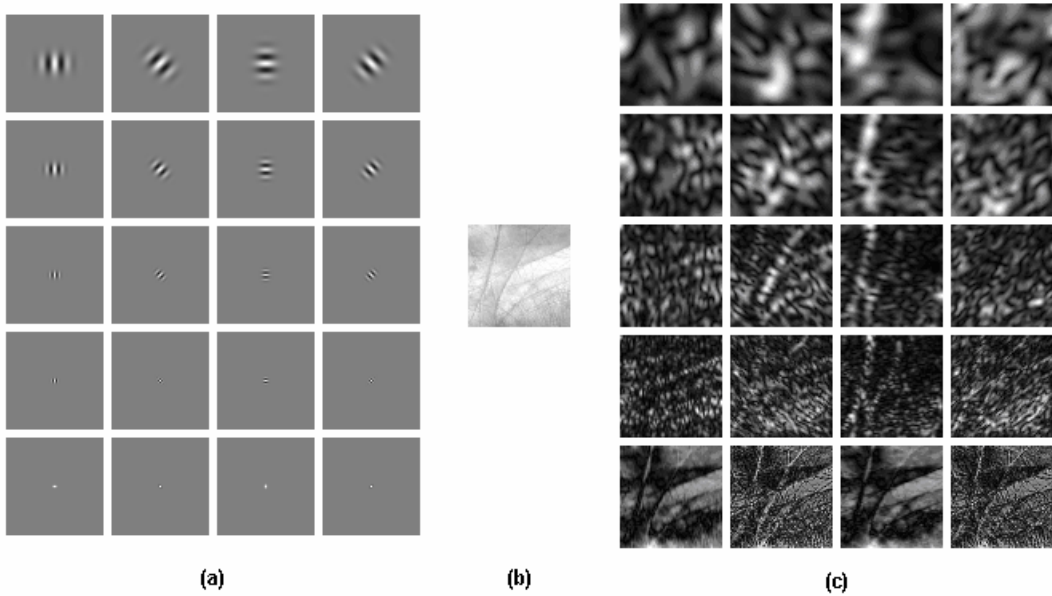
Elde ettiğimiz görüntüler üzerinde öznitelik ayırıştırma yöntemleri olarak FFT, DCT, ortogonal dalgacık dönüşümleri ve Gabor dalgacık dönüşümü uygulanmıştır. 2-D FFT ve 2-D DCT dönüşümleri ile uzaysal domenden ayrılıp frekans domenine geçilmiştir. FFT uygulamasında 3x3'den küçük ve 16x16'dan büyük katsayılara karşı düşen frekanslar elenmiştir. DCT'de ise 32x32'den büyük alan elenerek düşük frekans bilgileri kullanılmıştır. Burada amaç, gürültü sayılabilecek, tanımada gereksiz bilgilerin kaldırılması ve öznitelik çıkartma algoritmasına verilecek veri boyutunun düşürülmesidir.

Yerelleştirilmiş frekans bilgilerini sağlayan güçlü çoklu çözünürlük ayırıştırma tekniği ayrık dalgacık dönüşümü (DWT), değişik temel fonksiyonlar kullanılarak uygulanmıştır. Haar, Daubechies-4, Daubechies-8 ve Coiflet-6 temel fonksiyonları ilk veritabanında karşılaştırılmış, daha sonraki testlerde ise en uygun sonucu veren D-4 filtresi kullanılmıştır. Ayrık dalgacık dönüşümü uygulanırken hız ve kolaylık açısından filtre bankaları yaklaşımından yararlanılmıştır. 1. ve 2. veritabanlarında sırasıyla 4 ve 6. seviyelerdeki görüntüler hesaplanmış bunlardan 16x16 ve 32x32'lik alanlar alınmıştır. Bu seçimler deneylerle belirlenmiş olup avuç izini en iyi karakterize eden sonuçları vermektedirler. Şekil 2.4. örnek bir avuç izi görüntüsüne dalgacık dönüşümünün uygulanmasını göstermektedir.



Şekil 2.4. Değişik dalgacık alt bandlarındaki avuç izi görüntüleri (a) Birinci seviye dalgacık ayrıştırması (b) İkinci seviye dalgacık ayrıştırması (c) Üçüncü seviye dalgacık ayrıştırması

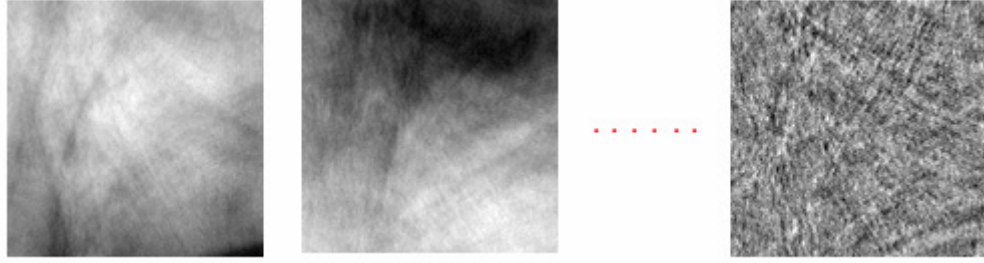
Gabor dalgacıkları yönteminde, Gabor çekirdeği değişik oryantasyon ve ölçeklerde görüntüyle katlanmaktadır. Burada, 1.2.4.6.'da anlatılan yöntem [10] uygulanmıştır. 5 ölçek ve 4 oryantasyon kullanılmıştır. Gabor dalgacıklarının görüntünün boyutunu büyütmesinden (görüntü boyu * oryantasyon * ölçek) dolayı görüntüler yarı yarıya küçültülmüşlerdir. Filtre tasarımı DC bileşen kaldırılmıştır. Böylece parlaklık değişmezliği sağlanmıştır. Gabor filtresinde kullanılan bandın yüksek ve düşük frekansları deneysel sonuçlarla belirlenmiştir. Gabor dalgacıklarının avuç izine uygulanmasına bir örnek Şekil 2.5.'te verilmiştir.



Şekil 2.5. Gabor dalgacıklarının avuç izine uygulanmasına bir örnek (a) Uygulanan Gabor dalgacıklarının reel bileşenleri (b) Örnek avuç izi görüntüsü (c) Filtrelenmiş avuç izi görüntülerinin genlik değerleri

2.2.3. Öznitelik Çıkartma

Avuç izi görüntülerine öznitelik çıkartma amacıyla PCA ve KPCA yöntemleri uygulanmıştır. Birinci veritabanında PCA ve KPCA uygulanırken bütün eğitim görüntüleri kullanılmış ve değişik öznitelik sayılarına göre çıkan başarı oranları hesaplanmıştır. İkinci veritabanının eğitim örneği sayısı çok büyük olduğundan her sınıfa ait sadece 1 örnek alınarak PCA ve KPCA'nın dönüşüm matrisi hesaplanmıştır. Böylece işlemler kabul edilebilir bir hızla uygulanmıştır. KPCA'da daha iyi sonuç verdiği için Gaussian çekirdek kullanılmıştır. Şekil 2.6.'da birinci, ikinci ve sonuncu özavuç karşılık gelen görüntüler verilmektedir.



Şekil 2.6. İlk avuç izi veritabanına göre ilk 2 ve sonuncu özavuç görüntüleri

2.2.4. Sınıflandırma

Görünüşe dayalı otomatik tanıma sistemimizin son aşaması sınıflandırmadır. Elde ettiğimiz özavuç görüntüleri 3 çeşit sınıflandırma yönteminden geçirilmiştir. Bunlardan ilki Euclidean doğrusal ayırtacıdır. Bu yöntem doğrusal olarak ayırma gerçekleştirdiğinden, dolayısıyla da düşük başarı verdiği için ileriki aşamalarda uygulamalarımızdan kaldırılmıştır. İkinci yöntem ağırlıklandırılmış Euclidean uzaklığına bağlı en yakın komşuluk yöntemidir. Bu yöntem basit olmasına karşın VC boyutunun [58] sonsuz olması gibi özelliklerinden dolayı iyi sınıflandırma yapan bir yöntemdir. Son olarak uygulanan yöntem ise Destek Vektör Makineleridir. Bu yöntemi çekirdek hile kullanarak doğrusal olmayan şekilde uyguladığımızdan iyi sonuçlar üretmiştir. Çekirdek olarak Gaussian çekirdeğin kullanıldığı bu yöntem, KPCA ile beraber etkin bir şekilde kullanılamamıştır. Çünkü, çekirdek yöntemlerde kullanılan parametrelerin (standart sapma, polinom derecesi, C sabiti gibi) otomatik belirlenmesi günümüzde üzerinde çok sayıda araştırma yapılan bir

konudur. Henüz bu konuda çok başarılı bir yöntem ortaya atılamadığından parametreler denemelerle seçilmiş olup parametre sayısının artması çok boyutlu uzayda gezinmemize ve bu yüzden de uygun sonucun bulunmasının zorlaşmasına neden olmaktadır.

2.2.5. Deneysel Sonuçlar ve Karşılaştırma

Avuç izi görüntüleri üzerindeki ilk test, 1. Bölge seçimi yöntemine göre ilk veritabanı üzerinde gerçekleştirilmiştir. İkinci oturumun ele alınmadığı bu testte ilk oturumdaki görüntüler'den bir veya ikisi eğitim, geri kalanlar ise test olarak alınmıştır. PCA, KPCA, DCT+KPCA, FFT+KPCA ve Dalgacık+KPCA yöntemlerinin karşılaştırıldığı testte değişik öznitelik boylarına göre tanımadaki başarılar verilmiştir. Sınıflandırma yöntemi olarak Euclidean doğrusal ayırtacının kullanıldığı bu testin karşılaştırmalı sonuçları Tablo 2.1.'de verilmektedir.

Tablo 2.1. PolyU-I veritabanında 1. bölge seçimi ile elde edilmiş ilk oturumdaki görüntüler arasında LED sınıflandırmasının karşılaştırmalı sonuçları

Eğitim Sayısı	Yöntem		Seçilen Öznitelik Sayısı				
			10	30	50	80	90
1	PCA		23	72	79.5	84	84
	KPCA		32	74	76.5	82	84.5
	DCT+KPCA		52	83.5	87.5	90	91.5
	FFT+KPCA		39	55	72	92	95
	WT+KPCA	D-4	47	83	86	90	93
		D-8	35	84	83.5	90	93.5
		Haar	40	80.5	82	86.5	90.5
		Coiflet6	52	84.5	86	89.5	92.5
2			Seçilen Öznitelik Sayısı				
			20	60	100	160	180
	PCA		66	79	81	83	83
	KPCA		63	77	80	85	86
	DCT+KPCA		74	81	89	95	95
	FFT+KPCA		59	80	91	93	93
	WT+KPCA	D-4	74	82	90	94	94
		D-8	76	80	91	95	95
		Haar	54	69	82	90	90
		Coiflet6	73	80	91	95	95

Sonuçlar incelendiğinde öznitelik sayısının artmasıyla tanıma oranlarının arttığı dikkat çekmektedir. Örneğin tek eğitim örneğinin kullanıldığı D-8 filtresinin sonuçlarına bakarsanız 10 öznitelik kullanıldığında %35 olan başarının 90 örnek kullanılması halinde %93.5 ile en yüksek değere çıktığı görülmektedir. Diğer bir dikkat çeken husus ise tayfsal domenin uygulandığı test sonuçlarının diğerlerine göre daha iyi sonuç vermesidir. Sonuçları incelediğinizde uzaysal domendeki başarının tayfsal domendekilere göre yaklaşık %10 civarında daha az olduğu görülmektedir. Testler aynı oturumdaki görüntüler arasında yapıldığı için eğitim örneği sayısının artmasının genel itibarı ile başarıda artış sağlamasına rağmen çok büyük getirisinin olduğu da söylenemez. Elde edilen sonuçlara göre dalgacık dönüşümünün en iyi ve en tutarlı sonuçları verdiği söylenebilir (örneğin FFT+KPCA tek eğitim örneğinde iyi sonuç vermesine rağmen iki eğitim örneğinde daha düşük sonuç vermiştir).

İkinci yapılan testte işlem adımları aynı olup, eğitim ve test örnekleri iki ayrı oturumdan alınmaktadır. Tablo 2.2.'deki sonuçlara bakıldığında eğitim örneği sayısının artmasıyla başarının logaritmik olarak arttığı görülmektedir. En yüksek başarı oranı %92 ile FFT+KPCA'dan elde edilmiştir. Bu testte de görülmektedir ki KPCA, PCA'ya göre; tayfsal domen de uzaysal domene göre azımsanmayacak derecede başarı sağlamaktadır. KPCA'nın PCA'ya göre başarılı olmasının belki de en önemli nedeni çekirdek hile ile doğrusal ayırtacın doğrusal olmayan ayırtaç haline getirilmesidir.

Tablo 2.2. PolyU-I veritabanında 1. bölge seçimi ile elde edilmiş her iki oturum görüntüleri arasında LED sınıflandırmasının karşılaştırmalı sonuçları

Eğitim Sayısı:	1	2	3	4
PCA	% 66	% 74.25	% 77.3	% 70
KPCA	% 73.6	% 83.75	% 84	% 82
FFT+KPCA	% 82	% 92.25	% 90.6	% 92
DCT+KPCA	% 80	% 89.5	% 90	% 90.5
DWT+KPCA	Haar	% 81.6	% 86.75	% 86.3
	D4	% 82.8	% 80.5	% 90.3
	D8	% 86.4	% 91.5	% 91.3
	Coiflet	% 82.4	% 89.5	% 89.3

Diğer yapılan iki test (Tablo 2.3. ve Tablo 2.4.) ise bu yapılan testlerde kullanılan Euclidean doğrusal ayırtacının yerine ağırlıklandırılmış Euclidean uzaklığına dayalı en yakın komşuluk sınıflandırıcının kullanılması ile elde edilen başarıdaki artışı göstermektedir.

Tablo 2.3.'teki sonuçlara bakıldığında en yakın komşuluk (WED) yönteminin doğrusal ayırmaya (LED) göre çok daha iyi sonuçlar verdiği görülmektedir. Bunun sebepleri olarak, verilerin örnek sayısının az olmasından dolayı ağırlık merkezinin hatalı örneklerle saptırılabilceği ve örneklerimizin öznelilik uzayında doğrusal olarak ayıramayacak karakteristiğe sahip olabilecekleri düşünülebilir. Bu sonuçlara bakıldığında en iyi sonuç %99.5 başarı ile FFT+KPCA'da görülmektedir. Ayrıca dikkat çeken diğer bir husus ise çekirdek hilenin LED'de WED'ye göre daha fazla yarar sağladığıdır.

Tablo 2.3. PolyU-I veritabanında 1. bölge seçimi ile elde edilmiş her iki oturum görüntüleri arasında LED ve WED sınıflandırmasının karşılaştırmalı sonuçları

Eğitim Sayısı:		1	2	3	4
PCA	LED	% 66.0	% 74.25	% 77.3	% 70.0
	WED	% 80.2	% 93.5	% 95.0	% 97.5
KPCA	LED	% 73.6	% 83.75	% 84.0	% 82.0
	WED	% 80.6	% 94.0	% 96.67	% 98.0
FFT+KPCA	LED	% 82.0	% 92.25	% 90.6	% 92.0
	WED	% 82.8	% 98.0	% 98.67	% 99.5
DCT+KPCA	LED	% 80.0	% 89.5	% 90.0	% 90.5
	WED	% 82.4	% 95.75	% 97.33	% 98.0
DWT+KPCA	LED	% 82.8	% 89.5	% 90.3	% 91.0
	WED	% 82.0	% 95.75	% 97.67	% 98.0

Tablo 2.4.'te sonuçları verilen testte ilgilenilen bölge seçimi olarak 2. yöntem uygulanmıştır. Özellikle eğitim örneği sayısı fazla olduğunda sonuçlar %100'lere yaklaştığı için uygulanan algoritmaların fazla bir öneminin kalmadığı görülmektedir. Fakat, eğitim örneği sayısı azaldıkça fark artmaktadır. Tek eğitim örneği sonuçlarına bakıldığında DWT'un en iyi sonucu verdiği gözükmemektedir. Bu testten çıkardığımız sonuç 2. bölge seçimi yönteminin daha iyi olduğudur.

Tablo 2.4. PolyU-I veritabanında 2. bölge seçimi ile elde edilmiş her iki oturum görüntüleri arasında WED sınıflandırmasının karşılaştırmalı sonuçları

Eğitim Sayısı:	1	2	3	4
PCA	% 92.4	% 99	% 99.67	% 100
KPCA	% 92.6	% 99.25	% 99.67	% 100
FFT+KPCA	% 91.4	% 99.5	% 99.67	% 100
DCT+KPCA	% 93.4	% 99.25	% 99.67	% 100
DWT+KPCA	% 93.8	% 99.25	% 99.67	% 100

Diğer bir testte ise, ilk oturum örnekleri eğitim, ikinci oturum örnekleri ise test olarak alınmaktadır. Bu test, diğerlerine göre daha gerçekçi olmaktadır. Bu amaçla 1. bölge seçimi ile LED uygulandığında en fazla %76 olan başarı oranı, WED'a dayalı en yakın komşuluk yöntemi kullanıldığında Tablo 2.5.'ten görüldüğü üzere %81.33'lere kadar çıkmaktadır. Bu sonuçlar da göstermektedir ki KPCA, PCA'dan, dalgacık dönüşümü de diğer dönüşümlerden daha iyi sonuç vermektedir. Şunu da not etmek gerekir ki DWT yönteminde filtre bankasında kullandığımız filtre D-4 filtresidir.

Tablo 2.5. PolyU-I veritabanında 1. bölge seçimi ile elde edilmiş görüntüler için oturumlar arası WED sınıflandırmasının karşılaştırmalı sonuçları

Yöntem	Öznitelik uzunluğu				
	25	75	125	200	300
PCA	% 70.66	% 77	% 76.33	% 75.66	% 77
KPCA	% 49.66	% 62	% 74.66	% 73.66	% 77.66
DCT+KPCA	% 72.66	% 77	% 80	% 80	% 80.66
FFT+KPCA	% 64.66	% 75.33	% 80	% 81.33	% 80.66
DWT+KPCA	% 71.66	% 78	% 79	% 80.66	% 81.33

Tablo 2.5.'te sonuçları verilen yöntemde sadece ilgilenilen bölge seçimini 2. tekniğe dayandırırsak Tablo 2.6.'da verilen sonuçlar elde edilmektedir. Bu sonuçlara bakarsanız en yüksek başarının %95 ile DWT+KPCA'da olduğu görülmektedir. Dolayısıyla ikinci bölge seçiminin ilgilenilen bölgeyi en iyi şekilde belirlediği sonucuna varılabilir. Bu tabloda dikkat çeken diğer bir husus ise sade PCA'nın, örnek sayısı belirli bir değeri aştığında hızlı iniş yapmasıdır. Bu sonuç PCA'nın karakteristiğinden kaynaklanmaktadır. Öznitelik sayısı belirli bir değeri aştığında anlamsız (yanlış sınıflandırmaya yol açacak) bir öznitelik hesaba katılabilir ve sınıflandırma performansı büyük ölçüde düşer. KPCA'da böyle bir sonuçla

karşılaşmamamızın sebebi ise muhtemelen, örüntülerdeki bilgilerin bir çok özneliliğe yayılmasıdır.

Tablo 2.6. PolyU-I veritabanında 2. bölge seçimi ile elde edilmiş görüntüler için oturumlar arası WED sınıflandırmasının karşılaştırmalı sonuçları

Yöntem	Öznitelik uzunluğu				
	25	75	125	200	300
PCA	% 81.66	% 86.33	% 85.66	% 85	% 29
KPCA	% 81.66	% 85	% 85.33	% 85.66	% 87.33
DCT+KPCA	% 88.33	% 94	% 94	% 94	% 94.33
FFT+KPCA	% 79	% 90	% 91.33	% 91.33	% 91.33
DWT+KPCA	% 88.66	% 94	% 95	% 95	% 95
DWT+PCA	% 87.33	% 92.66	% 14.33	% 39.33	% 39.33

İlk veritabanında elde ettiğimiz sonuçlardan dolayı ikinci veritabanında sadece ikinci tür bölge seçimi yöntemi kullanılarak ilgilenilen bölge belirlenmiştir. Bu veritabanında örnek sayısı çok daha fazla ($386 \times 10 \times 2$) olduğundan algoritmanın gerçek gücü daha iyi anlaşılabilir. Bununla birlikte işlem yükü de artmakta ve hesaplama yükünün azaltılması gündeme gelmektedir. Bu amaçla bu veritabanında yapılan testlerde PCA'nın dönüşüm matrisinin, KPCA'nın da Gram matrisinin hesaplanmasında her bir sınıftan sadece tek örnek alınmıştır. Böylece özvektör hesabı için gerekli süre kabul edilebilir seviyeye düşürülmüştür.

Tablo 2.7. PolyU-II veritabanında ilk oturumdaki görüntüler arasında LED ve WED sınıflandırmasının karşılaştırmalı sonuçları

Yöntem		Öznitelik uzunluğu				
		50	100	200	300	380
PCA	LED	% 60.66	% 71.80	% 74.56	% 74.39	% 74.13
	WED	% 98.74	% 99.17	% 99.09	% 99.05	% 98.96
DWT+PCA	LED	% 59.54	% 71.45	% 87.30	% 87.73	% 87.73
	WED	% 98.83	% 99.30	% 99.35	% 99.35	% 99.39
KPCA	LED	% 63.55	% 73.66	% 75.82	% 74.69	% 73.92
	WED	% 98.87	% 99.22	% 99.05	% 99.00	% 98.92
DWT+KPCA	LED	% 83.46	% 86.01	% 86.01	% 87.43	% 88.03
	WED	% 98.74	% 99.30	% 99.56	% 99.654	% 99.654

Bu veritabanında ilk yapılan test ilk oturumdaki 10 görüntüden 4'ünü eğitim, diğer 6'sını da test görüntüsü olarak almaktadır. Bu testte çekirdek yöntemin PCA'ya etkisi ve ayrık dalgacık dönüşümünün yararı araştırılmaktadır. Ayrıca WED sınıflandırıcısının LED sınıflandırıcıya göre daha iyi sonuç verdiği de gösterilmektedir. Tablo 2.7.'de görüldüğü üzere KPCA, PCA'dan daha iyi sonuç vermekte, DWT da pozitif etki oluşturmaktadır. En iyi sonuçlar DWT+KPCA ile elde edilmiş olup %99.654 başarı sağlamıştır.

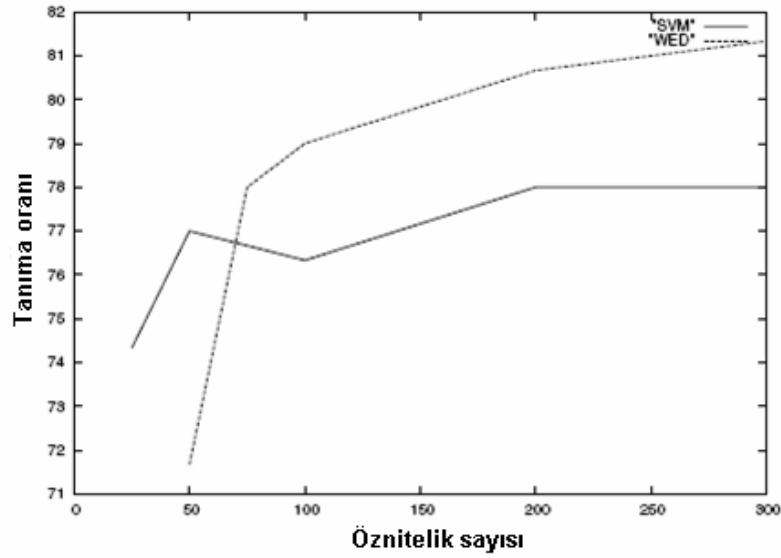
İkinci veritabanında yapılan diğer testte, ilk oturumdaki her bir kişiye ait 10 görüntü eğitim, diğer oturumdaki 10 görüntü de test olarak alınmaktadır. Bu testte WED'ye dayalı en yakın komşuluk sınıflandırması kullanılmıştır. Tablo 2.8.'de verilen sonuçlara göre en iyi sonuç %93.168 ile ayrık dalgacık dönüşümünde elde edilmiştir. Bu tabloda elde ettiğimiz sonuçlar daha önce elde ettiğimiz sonuçlara paraleldir. Dikkat çeken bir durum ise FFT+KPCA'nın %84.493'te kalmasıdır. Bu da daha önce belirttiğimiz DWT'un FFT'ye göre daha kararlı olduğu düşüncemizi doğrulamaktadır.

Tablo 2.8. PolyU-II veritabanında oturumlar arası WED sınıflandırmasının karşılaştırmalı sonuçları

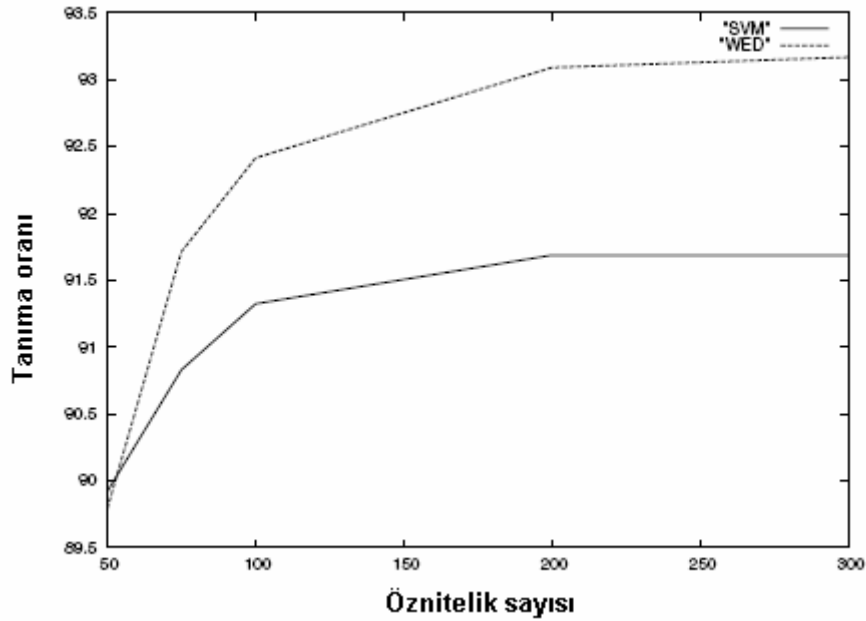
Yöntem	Öznitelik uzunluğu				
	50	75	100	200	300
PCA	% 88.59	% 90.31	% 90.85	% 91.24	% 91.24
DWT+PCA	% 89.45	% 91.24	% 92.10	% 91.72	% 92.67
KPCA	% 88.59	% 90.41	% 90.85	% 91.11	% 91.16
FFT+KPCA	% 71.32	% 76.18	% 78.80	% 82.44	% 84.49
DWT+KPCA	% 89.79	% 91.71	% 92.41	% 93.09	% 93.168

Görünüşe dayalı otomatik tanıma sistemimizde sınıflandırma algoritması olarak WED yerine SVM de denenmiştir. SVM, özellikle daha düşük sayıda öznitelik seçildiğinde başarılı olurken, öznitelik sayısı artırıldığında WED'dan daha kötü sonuçlar vermektedir. Şekil 2.7.'de 1. veritabanı üzerinde 1. bölge seçiminin kullanılmasında, Şekil 2.8.'de de 2. veritabanı üzerinde 2. bölge seçiminin kullanılmasında DWT+KPCA uygulanarak SVM ile WED'nin performans ölçümü yapılmaktadır. Şekil 2.7.'de 60, Şekil 2.8.'de de 50'den az öznitelik kullanıldığında SVM, WED'den daha iyi sonuç vermektedir. Bunun en önemli sebebi olarak, iki tane çekirdek yöntem kullanıyor olmamızın parametre seçimini uygun yapamamamıza neden olması görülebilir. Belirlenmesi gereken parametre sayısı arttıkça arama yapacağımız uzayın boyutu da artacaktır. Bu örneklerde KPCA parametreleri WED

için en iyi sonucu veren değerler olarak belirlenmektedir. Fakat, SVM ile KPCA'yı uygulamamız halinde KPCA için başka parametrelerin seçilmesi başarıyı arttırabilmektedir.



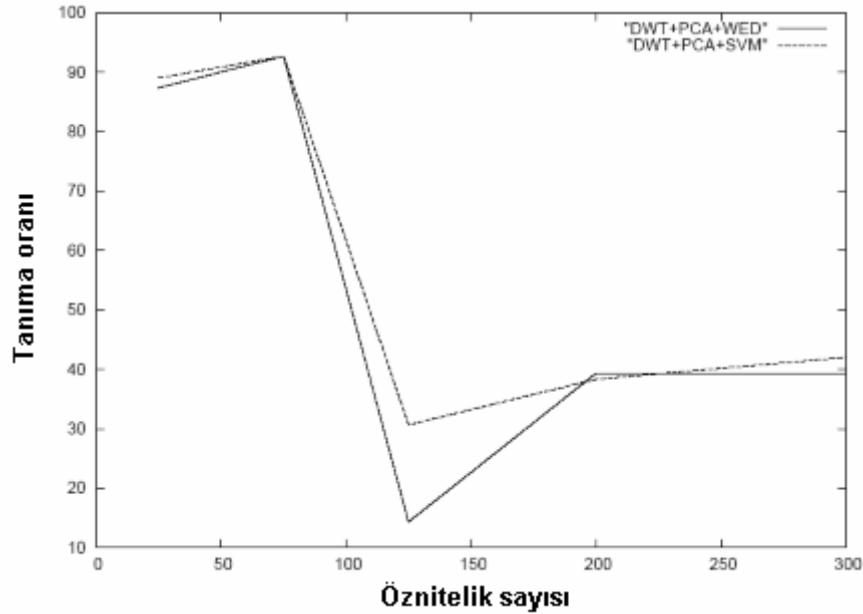
Şekil 2.7. PolyU-I veritabanında KPCA+ SVM ve WED'nin karşılaştırılması



Şekil 2.8. PolyU-II veritabanında KPCA+ SVM ve WED'nin karşılaştırılması

Şekil 2.9.'a baktığımızda görmekteyiz ki DWT+PCA'ya SVM'nin uygulanması WED'nin uygulanmasına yakın ama daha iyi sonuç vermektedir. Bu da bizim daha önceki

parametre seçiminin önem arz ettiği açıklamamızı doğrulamaktadır. Çünkü burada sadece SVM çekirdeğinin parametresi belirlenmektedir. Ayrıca SVM'nin, PCA ile çıkartılan öznitelik sayısının belirli bir değeri aşması durumunda oluşabilen ani düşüşü yumuşattığı da görülmektedir.



Şekil 2.9. PolyU-I veritabanında PCA+ SVM ve WED'nin karşılaştırılması

Son olarak uygulanan testler, Gabor dalgacıklarının etkisini araştırmaktadır. Bu amaçla 2. tür bölge seçimi ile elde edilen görüntüler üzerinde ilk oturum eğitim, ikinci oturum test olarak alınmış ve ilk veritabanı için Tablo 2.9., ikinci veritabanı için de Tablo 2.10.'da gösterilen sonuçlar elde edilmiştir. Tablo 2.9.'a baktığımızda 64x64'e küçültülmüş görüntü üzerine uygulanan Gabor dalgacıklarının, daha önce Tablo 2.6.'da DWT+KPCA ile elde edilen %95'lik başarıyı %100'e çıkarttığı görülmektedir. Benzer şekilde Tablo 2.10., Tablo 2.8. ile kıyaslanırsa ikinci veritabanından DWT+KPCA ile elde edilen %93.168'lik başarı, Gabor+KPCA ile %95.168'e çıkarılmıştır. Gabor dalgacıkları sonuçları bakımından çok uygun gözükse de bazı dezavantajları da vardır. Özellikle verinin boyutunun şişirilmesi ve işlem yükünün artması en büyük sorunlar olarak gözükmemektedir. Bunları çözmek için değişik yaklaşımlar ileri sürülmüş olmakla beraber henüz geliştirme aşamasındadırlar.

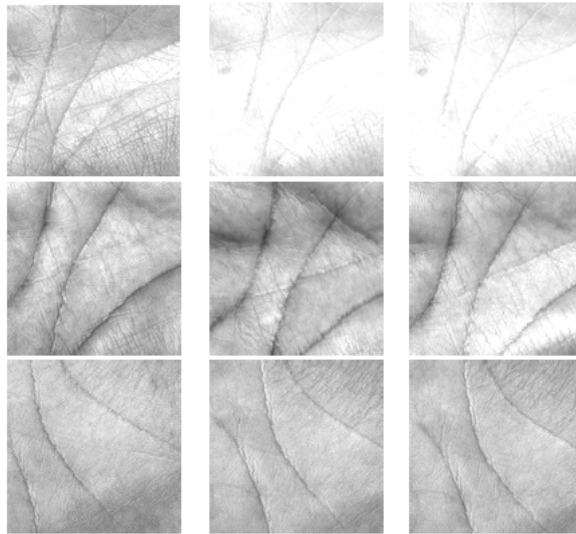
Tablo 2.9. PolyU-I veritabanında Gabor için farklı oturum testleri

Yöntem	Öznitelik uzunluğu				
	25	75	125	200	300
Gabor+KPCA	% 88.33	% 99.33	% 100	% 100	% 100

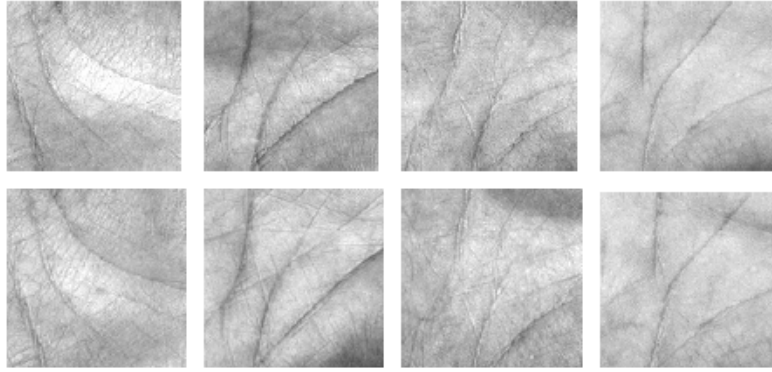
Tablo 2.10. PolyU-II veritabanında Gabor için farklı oturum testleri

Yöntem	Öznitelik uzunluğu				
	50	75	100	200	300
Gabor+KPCA	% 86.1	% 90.25	% 92.05	% 94.25	% 95.168

Genel olarak test sonuçlarına bakıldığında başarının yüksek olduğu görülmektedir. Başarıyı etkileyen faktörlerden biri olarak aynı avuca ait görüntülerin birbirlerinden farklı olması düşünülebilir. Görüntüler incelendiğinde farklı oturumlarda alınan görüntülerde parlaklık değişimlerinden dolayı bilgi kaybı, daha önce belirgin olmayan çizgilerin belirmesi ve rotasyon, öteleme farklılıkları olduğu görülür. Şekil 2.10.'da en soldaki görüntüler ile sağdakiler farklı oturumlardan alınmış olup bu sorunlar gösterilmektedir. Bu değişimler sonucu olumsuz etkilemektedir. Hatta, aynı oturumda alınan görüntüler arasında bile bu tür sorunların olduğu ve sınıflandırmada negatif etkiye neden olduğu söylenebilir. Şekil 2.11.'de verilen görüntülerden üst satırdaki görüntüler eğitim olarak verilirken alt satırdakiler de test olarak verilmiş ve DWT+KPCA sonuçlarına göre yanlış sınıflandırılmışlardır.



Şekil 2.10. Aynı oturumlardaki görüntüler arasındaki farklılıklar



Şekil 2.11. Aynı oturumdaki görüntüler arasındaki farklılıklar

İkinci veritabanında başarıyı düşüren diğer bir neden olarak da PCA ya da KPCA için kullandığımız eğitim örneği sayısını sınırlamamız görülebilir. Her bir sınıftan tek eğitim örneği seçmemizin sebebi eğitim süresini kısa tutmaktır. Birden çok örnek seçildiğinde özvektörleri hesaplanacak matrisimizin boyu büyümekte, dolayısıyla hesaplama süresi üstel olarak artmaktadır. Bununla beraber tek örnek seçilmesi başarıyı belirli bir oranda düşürmektedir. Bunu göstermek için ilk veritabanında KPCA için 3 eğitim örneğini ele alarak hesapladığımız Gabor testleri, tek ve iki örnek alarak tekrar hesaplanmıştır. Tablo 2.11.'deki sonuçlardan da görüldüğü üzere iki örnek alındığındaki sonuçlar üç örnek alındığına yakın sonuçlar verirken, tek örnek alındığında tanıma oranında düşüş yaşanmıştır.

Tablo 2.11. PCA'da eğitim örneği sayısının önemi

Yöntem	KPCA için alınan eğitim örneği sayısı		
	1	2	3
Gabor+KPCA	% 98	% 100	% 100
Gabor+PCA	% 97.33	% 99.66	% 99.66

Burada son olarak şunu da belirtmek gerekir ki, çekirdek yöntemlerin parametre seçiminin de sonuçlara etkisi olmaktadır. Parametreler otomatik olarak seçilemediğinden en iyi değerler bulunamamış olabilir. Bu durumdan dolayı da tanıma oranlarının düşebileceği göz ardı edilmemelidir.

Uygulanılan yöntemlerin gerçekleştirilme zamanları incelendiğinde Tablo 2.12.'deki sonuçlar ortaya çıkmıştır. Bu sonuçlar 3.0 GHz P4 işlemci ve 1 GB belleğe sahip bir bilgisayarda optimize edilmemiş kodlarla testler yapılarak elde edilmiştir. Optimize kodlarla çok değişik sonuçların çıkabileceği de unutulmamalıdır. Testlerde ilk veritabanında 300 örnek test, 300 örnek de eğitim olarak alınmıştır. PCA ve KPCA sonuçları öznitelik ayrıştırmadan geçirilmemiş 128*128 boyutlu ham görüntü üzerinde gerçekleştirilmiştir. VT-II'de ise 3850 görüntü eğitim 3850 görüntü test olarak alınmıştır. PCA ve KPCA için her sınıftan tek örnek alındığından 385 örnek ile çekirdek matrisi oluşturulmuş ve çıkartılan temel bileşen sayısı 380 olarak belirlenmiştir. PCA ve KPCA bu testlerde uygulanılan ham görüntü yerine dalgacık dönüşümü ile elde edilmiş 16*16 boyutlu görüntü üzerinde uygulanırsa PCA süresi 80.8 saniyeye (42.5 sn. eğitim, 38.3 sn. test), KPCA süresi de 95.4 saniyeye (50.4 sn. eğitim, 45 sn. test) düşmektedir. Bu sonuç da dikkate alınması gereken bir husustur. Genel olarak testlerin süreleri incelendiğinde DWT+KPCA testlerinin gerçek zamanlı olarak uygulanabileceği görülmektedir. Gabor için ise iyi bir optimizasyon tekniğine ihtiyaç vardır.

Tablo 2.12. Uygulanılan yöntemlerin gerçekleştirilme süreleri (saniye olarak)

Yöntem	Kullanılan veritabanı	
	VT-I	VT-II
FFT	4.4	70.3
DCT	27.4	438
DWT.	0.9	17.2
Gabor	112.6	1689
PCA	35.6 (eğitim=28, test=17.6)	666.9 (eğitim=350.4, test=326.5)
KPCA	62.9 (eğitim=38.4, test=24.5)	896.3 (eğitim=454.8, test=441.5)
LED	0.15	12.3
WED	0.9	57.1
SVM	1.7 (eğitim=1, test=0.7)	299.2 (eğitim=126.7, test=172.5)

Elde edilen sonuçlar yayınlanan diğer avuç izi tanıma yöntemleriyle karşılaştırılmıştır. Tablo 2.13.'de verilen yöntemler, testlerinde eğitim ve test görüntülerini aynı oturumdan

almaktadırlar. Burada önerdiğimiz yöntemler DWT+KPCA'nın PolyU-I ve II veritabanlarına uygulansıdır (Önerilen 1 ve Önerilen 2). Tablodaki Ö1 ve Ö2 bizim önerdiğimiz yöntemlerin sırasıyla birincisi ve ikincisidir.

J. Tao vd. çalışmalarında [42] PolyU-I veritabanına 2-D PCA'yı uygulamıştır. 4 örnek eğitim, 2 örnek de test olarak alınarak %98.4 başarı elde edilmiştir. J. Lu [43] ise çalışmasında 2-D PCA'yı dalgacık dönüşümü ile birleştirmeyi önermiştir. PolyU-I veritabanının kullanıldığı bu testlerde %97 başarı elde edilmiştir. Dalgacık ile ICA'yı birlikte uygulayan Guan-Ming Lu [44], 400 avuçtan 10'ar adet görüntü almakta ve bunlardan 4'ünü eğitim görüntüsü olarak kullanmaktadır. Elde ettiği başarı ise %98.44'tür. Lei Zhang [45], dalgacık dönüşümü ile yönsel içerik modellemeyi birleştirerek 50 avucu ayırt etme problemini ele almıştır. Her bir avuca ait tek görüntü eğitim için alınarak 150 görüntü test edilmiş ve %98 oranında tanıma sağlanmıştır. Xiangqian Wu vd. [46] çalışmalarında FLD yöntemini kullanarak 300 kişiye ait aynı oturumda alınmış 10 görüntüden 6'sını eğitim için ayırmış ve %99.2 başarı elde etmiştir. Özavuç yaklaşımının ele alındığı Guangming Lu'nun [47] çalışmasında 382 kişiye ait 8 avuç görüntüsünden 4'ü eğitimde kullanılarak %99.149 başarı sağlanmıştır. Wang Y.'nin [48] çalışmasında da KFDA uygulanarak 5 görüntü eğitim 5 görüntü de test olarak ele alınmış, %97.5 başarı sağlanmıştır. DCT ile FLD'ı birleştiren Xiao-Yuan Jing [49] 20 örnekten 5'ini eğitim için kullanarak 190 avucu %98.13 başarı oranı ile tanımıştır. Görüldüğü üzere yöntemimiz performans açısından diğer yöntemlerden daha iyi sonuç vermektedir.

Tablo 2.13. Karışık oturum testlerinin diğer yöntemlerle kıyaslanması

Yöntemler	Ö1	Ö2	[42]	[43]	[44]	[45]	[46]	[47]	[48]	[49]
avuç sayısı	100	386	100	100	400	50	300	382	160	190
örnek sayısı	600	3860	600	600	4000	200	3000	3056	1600	3040
Tanıma oranı	100	99.654	98.4	97.0	98.44	98.0	99.2	99.149	97.5	98.13

Tablo 2.14.'de ise karşılaştırılan bütün sonuçlar PolyU-I veritabanında 1. oturumun eğitim, 2. oturumun da test olarak alınmasıyla elde edilmiştir. Wei Jiang [50] çalışmasında PCA ile FLD'ı birleştirmiş ve %97.5 başarı elde etmiştir. Ajay Kumar [51] ise istatistiksel olarak değişik yöntemleri denemiş ve %95.8 ile en iyi sonucunu elde etmiştir. Dewen Hu'nun [52] 2DLPP yaklaşımı ise pek de iyi sayılamayacak %84.7 başarı oranında

kalmıştır. Jian Yang [53] UDP ismini verdiği yöntemi önermiş ve %99.7 başarı ile bizim önerdiğimiz yaklaşıma en yakın sonucu elde etmiştir. 2-D PCA ile AMD yaklaşımını birleştiren Wang-Meng Zuo [54] da %97.67 başarı elde etmiştir. Görüldüğü üzere Gabor+KPCA yaklaşımımız diğer bütün yöntemlerden daha iyi sonuç vermiştir. Karşılaştırmalara 2. veritabanının konulmamasının sebebi ise bu veritabanında yapılmış böyle bir testin bulunamamasıdır.

Tablo 2.14. Aynı oturum testlerinin diğer yöntemlerle kıyaslanması

	Yöntem					
	Önerilen	[50]	[51]	[52]	[53]	[54]
Tanıma oranı	100	97.5	95.8	84.7	99.7	97.67

3. SONUÇLAR

Son zamanlarda örüntü tanımada uygulanan popüler bir yaklaşım görünüşe dayalı tanımadır. Çalışmamızda bu yaklaşım üzerinde durulmuş ve performansının model tabanlı yaklaşımlardan bile daha iyi olduğu görülmüştür. Bu yaklaşımın diğer bir yararı, başka tür problemlere de genelleştirilebilir olmasıdır.

Uygulamamızda, ön işlem olarak öznitelik ayrıştırmanın etkileri incelenmiştir. Bu işlemlerin, frekans bilgilerini ortaya çıkartarak tanıma oranını pozitif yönde etkilediği görülmüştür. Geleneksel FFT ve DCT yaklaşımlarının haricinde frekansta yerelliği sağlayan ortogonal dalgacık dönüşümleri ve Gabor dalgacıkları da incelenmiş ve en iyi sonuçlar bu dönüşümlerden sağlanmıştır. Ayrık dalgacık dönüşümünün bir yararı olarak da filtre bankası kullanımından ötürü hız kazancı olduğu söylenebilir. Gabor dalgacıkları ise veri boyutunu büyütmede ve işlem hızını düşürmekte olmasına rağmen performans bakımından göz ardı edilemeyecek sonuçlar vermiştir. Dolayısıyla görünüşe dayalı yaklaşımlarda öznitelik ayrıştırma tekniklerinin vazgeçilmemesi gereken bir işlem adımı olduğu sonucuna varılmıştır.

Görünüşe dayalı yaklaşımın ana iskeletini oluşturan öznitelik çıkartımı için geleneksel PCA yönteminin yanında çekirdek PCA da uygulanmıştır. Çekirdek PCA, PCA'nın doğrusal olmayan biçimi olup ikinci dereceden istatistik yerine çok daha yüksek derecelerden istatistik sağlamaktadır. Yaptığımız araştırmalardan görmekteyiz ki çekirdek kavramı son yıllarda bir çok doğrusal yönteme uygulanarak bu yöntemleri doğrusal olmayan yöntemlere başarı ile dönüştürmüştür. Çekirdek yöntemlerinin en büyük yararı, doğrusal olmayan uzaya doğrudan değil, dolaylı olarak dönüşüm yapmasıdır. Yani, veriler doğrusal uzaya aktarılmayıp sadece o uzaydaki işlemlerinin sonuçları elde edilmektedir. Böylece sonsuz boyuttaki bir uzayda (Gaussian gibi) bile işlemler kolaylıkla yapılabilir. Çalışmamızda çekirdek yöntemlerin doğrusal yöntemlerle kıyaslandığında başarı açısından daha iyi sonuçlar verdiği görülmüştür.

Çekirdek yöntemlerde karşılaştığımız tek sorun olarak parametre seçiminin otomatik bulunmasındaki güçlükler düşünülebilir. Parametre seçimi için günümüzde değişik yaklaşımlar denenmekte olup en yaygın kullanılan bağımsız geçerlilik sınavmasının bile her zaman çok iyi sonuç vermediği ve büyük veritabanları için fazla işlem yüküne neden olduğu görülmektedir. Parametre seçiminin elle yapılması ise hem yöntemin gerçek

problemlere uygulanabilirliğine engel olmakta hem de en iyi parametrenin bulunamamasına sebep olabilmektedir. Bu yüzden parametre seçiminin uygun ve otomatik olarak yapılmasının önem arz ettiği görülmüştür.

Yöntemimizde son aşama olarak uygulanan sınıflandırmada, dikkat çeken en önemli sonuç en yakın komşuluk yönteminin basit olmasına karşın performans açısından çok iyi olduğudur. SVM'nin de benzer sonuçlar verdiği bu çalışmada doğrusal bir sınıflandırıcı olan LED zayıf kalmıştır. Burada şunu da unutmamak gerekir ki, bir sınıfa ait örnek sayısının arttığı problemlerde en yakın komşuluk yöntemi test süresi bakımından iyi sonuç verememekte ve SVM yönteminin etkinliği daha iyi anlaşılmaktadır.

Sonuç olarak önerdiğimiz yöntemi diğer yöntemlerle kıyasladığımızda, performans ve genelleştirilebilme açısından gelecek vaad eden bir yöntem olduğunu görmekteyiz.

Çalışma sırasında, geliştirilen yaklaşımın avuç izi tanıma problemine uygulanmasına dayanan 3 adet bildiri SIU 2007, Canadian AI 2007 ve MLDM 2007 konferanslarına sunulmuş ve bildiri kitapçıklarında basılmıştır [59, 60, 61].

4. ÖNERİLER

1. Çalışmamızda uyguladığımız Gabor dalgacıkları, görüntünün boyutunu arttırdığı için hız kaybına neden olmaktadır. Bu yüzden gerçek zamanlı uygulamalar için Gabor dalgacıklarının sonuç kümesinin düzensiz örnekleme gibi yöntemlerle küçültülmesi gerekir. Böylece performans artışına neden olan Gabor dalgacığının rahatlıkla uygulanabilmesi sağlanacaktır.
2. Üzerinde çalışılması gereken diğer bir husus ise çekirdek parametrelerinin en uygun değerlerinin otomatik olarak etkin bir biçimde bulunabilmesinin sağlanmasıdır. Bu gerçekleştirildiğinde bir çok çekirdek yöntem kolaylıkla birleştirebilecek duruma gelecektir. Ayrıca performans iyileştirme amacıyla her bir boyut için farklı çekirdek parametresi kullanımı üzerinde de durulabilir.
3. Çalışmada kullanılan öznitelik ayırıştırma teknikleri daha da genişletilebilir. KFDA, KCCA, KICA gibi değişik çekirdek yöntemlerin de uygulanıp karşılaştırılmasında yarar vardır. Bu yöntemlerden özellikle KFDA, biometrik tanımda PCA'dan sonra kullanılan en yaygın teknik olarak görülmektedir.
4. İstatistiğin ilgi alanlarından olan sınıflandırma yöntemleri üzerinde de daha çok durulabilir. Örneklerin uzaydaki dağılımları üzerinde olasılıksal bir yaklaşım olan Bayes öğrenme kuralının yanında, karar ağaçları, HMM ve YSA üzerinde de araştırmalar yapıp algoritmaların güçlülüğü denenebilir.
5. Son yıllarda üzerinde çok araştırmalar yapılan SVM yönteminin değişik versiyonları üzerinde araştırmalar yapıp performans artışı sağlanılmaya çalışılmalıdır.
6. Yöntemimiz genel amaçlı bir yöntem olduğu için başka veritabanlarında denenmesi yöntemimizin başarısını daha iyi gösterebilir. Bu amaçla yöntemin, biometriklerin tanınmasında, bakteri tanımda, ve genel nesne tanımda da uygulanmasında yarar vardır.
7. Uyguladığımız yöntemde üzerinde durmadığımız, fakat araştırılması gereken diğer bir konu ise rotasyon, öteleme ve ölçekleme değişmezliklerinin sağlanabilmesidir. Bu konular uyguladığımız avuç izi görüntüleri için önem arz etmese de bakteri tanıma gibi genel amaçlı uygulamalarda sonucu büyük ölçüde etkileyecektir.

8. Son olarak şunun da belirtilmesi gerekir ki, yöntemler geliştirilirken birbirlerine uygunluğu düşünülmelidir. Matematiksel alt yapının incelenmesi ve istatistiksel ve geometriksel yorumlar sonucunda yararı olabileceği düşünülen yöntemlerin gerçekleştirilmesi gerekir. Karmaşıklıktan ziyade bu hususa dikkat edilmesi gerekir. Unutulmamalıdır ki, bazen çok basit bir yaklaşım, çok daha karmaşık bir yaklaşımdan daha iyi sonuç verebilmektedir.

5. KAYNAKLAR

1. Acharya, T. ve Ray, A. K., Image Processing, Principles and Applications, Wiley Publishing, New Jersey, 2005.
2. Ahmed, N., Natarajan, T. ve Rao, K. R., On Image Processing and a Discrete Cosine Transform, IEEE Trans. on Computers, 23, 1 (1974) 90-93.
3. Crane, R., A Simplified Approach to Image Processing, Prentice Hall, New Jersey, 1997.
4. Vidakovic, B., Statistical Modeling by Wavelets, Wiley Publishing, New York, 1999.
5. Bow, S.T., Pattern Recognition and Image Preprocessing, Second Edition, Marcel Dekker Inc., New York, 2002.
6. Gabor, D., Theory of Communication, J. IEE, 93 (1946) 429-457.
7. Osadebey, M. E., Integrated Content-Based Image Retrieval Using Texture, Shape and Spatial Information, Yüksek Lisans Tezi, Umea University, Umea, 2006.
8. Daugman, J. G., Complete discrete 2-D Gabor transforms by neural Networks for image analysis and compression, IEEE Trans. Acoust. Speech Signal Process, 36 (1988) 1169-1179.
9. Marcelja, S., Mathematical description of the responses of simple cortical cells, Journal of the Optical Society of America 70 (1980) 1297-1300.
10. Manjunath, B. S. ve Ma, W.Y., Texture Features for Browsing and Retrieval of Image Data, IEEE Transaction on Pattern Analysis and Machine Intelligence, 18, 8 (1996) 837-842.
11. Daugman, J.G., Uncertainty relations for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters, Journal of the Optical Society of America A, 2 (1985) 1160-1169.
12. Jones, J. ve Palmer, L., An evaluation of the two-dimensional Gabor filter model of simple receptive fields in cat striate cortex, Journal of Neurophysiology, 58, 6 (1987) 1233-1258.
13. Porat, M. ve Zeevi, Y.Y., The generalized Gabor scheme of image representation in biological and machine vision, IEEE Trans. Pattern Analysis and Machine Intelligence, 10, 4 (1988) 452-468.
14. Duda, R. O., Hart, P. E. ve Stork, D. G., Pattern Classification, Second Edition, Wiley Publishing, New York, 2001.

15. Sirovich, L. ve Kirby, M., Low-dimensional procedure for the characterization of human faces, Journal of the Optical Society of America, 2 (1987) 519-524.
16. Turk, M. ve Pentland, A., Eigenfaces for Recognition, Journal of Cognitive Neuroscience, 3, 1 (1991) 71-86.
17. Penev, P. S. ve Sirovich, L., The Global Dimensionality of Face Space, Fourth IEEE International Conference on Automatic Face and Gesture Recognition, March 26–30 2000, Grenoble, France, Bildiri Kitabı, 264-270.
18. Zhao, L. ve Yang, Y., Theoretical analysis of illumination in PCA-based vision systems, Pattern Recognition, 32, 4 (1999) 547-564.
19. Wiskott, L., Fellous, J. M., Krüger, N. ve Malsburg, C., Face Recognition by Elastic Bunch Graph Matching, 7th International Conference on Computer Analysis of Images and Patterns, 1997, Kiel, Bildiri Kitabı, 456-463.
20. Pres, W. H., Teukolsky, S. A., Vetterling, W. T. ve Flannery, B. P., Numerical Recipes in C, Cambridge University Press, New York, 1997.
21. Alpaydın, E., Introduction to Machine Learning, The MIT Press, London, 2004.
22. Yambor, W. S., Analysis of PCA-Based And Fisher Discriminant-Based Image Recognition Algorithms, Yüksek Lisans Tezi, Colorado State University, Colorado, 2000.
23. Cristianini, N. ve Taylor, J. S., An Introduction to Support Vector Machines and Other Kernel-based Learning Methods, Cambridge University Press, UK, 2000.
24. Schölkopf, B. ve Smola, A. J., Learning with Kernels, The MIT Press, London, 2002.
25. Taylor, J. S. ve Cristianini, N., Kernel Methods for Pattern Analysis, Cambridge University Press, UK, 2004.
26. Boser, B. E., Guyon, I. M. ve Vapnik, V., A training algorithm for optimal margin classifiers, 5th Annual ACM Workshop on Computational Learning Theory, July 1992, Pittsburgh, Bildiri Kitabı, 144-152.
27. Guyon, I., Boser, B. ve Vapnik, V., Automatic capacity tuning of very large VC-dimension classifiers, Advances in Neural Information Processing Systems, 5 (1993) 147-155.
28. Vapnik, V., The Nature of Statistical Learning Theory, Springer-Verlag, New York, 1995.
29. Abe, S., Support Vector Machines for Pattern Classification, Springer, London, 2005.

30. Romdhani, S., Gong, S. ve Psarrou, A., A multi-view nonlinear active shape model using kernel PCA, British Machine Vision Conf., 1999, Nottingham, UK, Bildiriler Kitabı, 483–492.
31. Huang, M. H., Kernel Eigenfaces vs. Kernel Fisherfaces: face recognition using kernel methods, Fifth IEEE International Conference on Automatic Face and Gesture Recognition, May 2002, Washington, Bildiriler Kitabı, 215-220.
32. Scholkopf, B., Smola, A. ve Muller, K. R., Nonlinear Component Analysis as a Kernel Eigenvalue Problem, Neural Computation, 10 (1998) 1299-1319.
33. Yang, M. H., Ahuja, N. ve Kriegman, D., Face Recognition Using Kernel Eigenfaces, IEEE International Conference on Image Processing, 2000, Vancouver, Bildiriler Kitabı, 37-40.
34. Kim, K. I., Jung, K. ve Kim, H. J., Face recognition using kernel principal component analysis, IEEE Signal Processing Letters, 9, 2 (2002) 40-42.
35. Schölkopf, B., Burges, C. J. C. ve Smola, A. J., Advances in kernel methods: support vector learning, The MIT Press, London, 1999.
36. Marques de Sa, J. P., Pattern Recognition Concepts, Methods and Applications, Springer, Portugal, 2001.
37. Vapnik, V., Golowich, S. ve Smola, A., Support Vector Method for Function Approximation, Regression Estimation and Signal Processing, Neural Information Processing Systems, 9 (1997) 281-287.
38. Theodoridis, S. ve Koutroumbas, K., Pattern Recognition, Second Edition, Academic Press, London, 2003.
39. Webb, A. R., Statistical Pattern Recognition, Second Edition, Wiley Publishing, U.K., 2002.
40. Abe, S., Pattern Classification: Neuro-Fuzzy Methods and Their Comparison, Springer-Verlag, London, 2001.
41. Murase, H. ve Nayar, S. K., Visual learning and recognition of 3-D objects from appearance, International Journal of Computer Vision, 14 (1995) 5–24.
42. Tao, J., Jiang, W., Gao, Z., Chen, S. ve Wang, C., Palmprint Recognition Based on 2-Dimensional PCA, First International Conference on Innovative Computing, Information and Control, 2006, Beijing, Bildiriler Kitabı, 326-330.
43. Lu, J., Zhang, E., Kang, X., Xue, Y. ve Chen, Y., Palmprint Recognition Using Wavelet Decomposition and 2D Principal Component Analysis, Int. Conf. on Communication, Circuits, and Systems, 2006, Guilin, Bildiriler Kitabı, 2133-2136.

44. Lu, G., Wang, K. ve Zhang, D., Wavelet Based Independent Component Analysis for Palmprint Identification, 3rd. Int. Conf. on Machine Learning and Cybernetics, 2004, Shanghai, Bildiriler Kitabı, 3547-3550.
45. Zhang, L. ve Zhang, D., Characterization of Palmprints by Wavelet Signatures via Directional Context Modeling, IEEE Transactions on Systems, Man, and Cybernetics, 34, 3 (2004) 1335-1347.
46. Wu, X., Zhang, D. ve Wang, K., Fisherpalms Based Palmprint Recognition, Pattern Recognition Letters, 24, 15 (2003) 2829–2838.
47. Lu, G., Zhang, D. ve Wang, K., Palmprint Recognition Using Eigenpalms Features, Pattern Recognition Letters, 24 (2003) 1463–1467.
48. Wang, Y. ve Ruan, Q., Kernel Fisher Discriminant Analysis for Palmprint Recognition, The 18th Int. Conference on Pattern Recognition, 2006, Hong Kong, Bildiriler Kitabı, 457–460.
49. Jing, X. Y. ve Zhang, D., A Face and Palmprint Recognition Approach Based on Discriminant DCT Feature Extraction, IEEE Trans. on Systems, Man, and Cybernetics, 34, 6 (2004) 2405–2415.
50. Jiang, W., Tao, J. ve Wang, L., A Novel Palmprint Recognition Algorithm Based on PCA and FLD, Int. Conference. on Digital Telecommunications. 2006, Cote d'Azur, Bildiriler Kitabı, 28-28.
51. Kumar, A. ve Zhang, D., Personal Recognition Using Hand Shape and Texture, IEEE Transactions on Image Processing, 5, 8 (2006) 2454–2460.
52. Hu, D., Feng, G. ve Zhou, Z., Two-Dimensional Locality Preserving Projections with Its Application to Palmprint Recognition, Journal of the Pattern Recognition Soc., 40 (2007) 339-342.
53. Yang, J., Zhang, D., Yang, J. ve Niu, B., Globally Maximizing, Locally Minimizing: Unsupervised Discriminant Projection with Applications to Face and Palm Biometrics, IEEE Transactions on Pattern Aanalysis and Machine Intelligence, 29, 4 (2007) 650-664.
54. Zuo, W. M., Wang, K. Q. ve Zhang, D., Assembled Matrix Distance Metric for 2DPCA-Based Face and Palmprint Recognition, Fourth International Conference on Machine Learning and Cybernetics, 18-21 August 2005, Guangzhou, Bildiriler Kitabı, 210-216.
55. Zhang, D. ve Shu, W., Two novel characteristics in palmprint verification: Datum point invariance and line feature matching, Pattern Recognition, 32 (1999) 691-702.
56. You, J., Li, W. ve Zhang, D., Hierarchical palmprint identification via multiple feature extraction, Pattern Recognition, 35, 4 (2002) 847-859.

57. Parker, J. R., Algorithms for Image Processing and Computer Vision, Wiley Publishing, New York, 1996.
58. Vapnik, V. ve Chervonenkis, A., On the uniform convergence of relative frequencies of events to their probabilities, Theory of Probability and its Applications, 16, 2 (1971) 264-280.
59. Aykut, M. ve Ekinci, M., Dalgacık Tabanlı Kernel PCA Uygulayarak Avuç İzi Tanıma, IEEE 15. Sinyal İşleme ve İletişim Uygulamaları Kurultayı, 11-13 Haziran 2007, Eskişehir.
60. Ekinci, M. ve Aykut, M., A Novel Approach for Automatic Palmprint Recognition, 20th Conference of the Canadian Society for Computational Studies of Intelligence, Canadian AI 2007, 28-30 May 2007, Montreal, Bildiriler Kitabı, 122-133.
61. Ekinci, M. ve Aykut, M., Palmprint Recognition by Applying Wavelet Subband Representation and Kernel PCA, 5th International Conference on Machine Learning and Data Mining MLDM'2007, 18-20 July 2007, Leipzig, Bildiriler Kitabı, 628-642.
62. Zhang, D., Kongi, W., You, J. ve Wong, M., Online Palmprint Identification, IEEE Trans. on Pattern Analysis and Machine Intelligence, 25, 9 (2003) 1041–1049.
63. www.comp.polyu.edu.hk/biometrics/ Palmprint Database. 29 Mayıs 2004.

ÖZGEÇMİŞ

1981 yılında Trabzon'da doğdu. İlk öğrenimini Mimar Sinan İlkokulu ve Cumhuriyet Ortaokulu'nda, orta öğrenimini Trabzon Lisesi'nde tamamladı. 2001 yılında Karadeniz Teknik Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü'nde lisans programına başladı ve 2005 yılında bu bölümden bölüm birincisi olarak mezun oldu. Aynı yıl Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı'nda yüksek lisans programına ve aynı bölümde araştırma görevlisi olarak çalışmaya başladı. Halen bu görevini sürdürmektedir. Yabancı dil olarak İngilizce bilmektedir.