

YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

VIDEO GÖRÜNTÜLERİNDEN YÜZ TANIMA

Matematik Mühendisi İbrahim Saygın TOPKAYA

**FBE Matematik Mühendisliği Anabilim Dalı Matematik Mühendisliği Programında
Hazırlanan**

YÜKSEK LİSANS TEZİ

Tez Danışmanı : Yrd. Doç. Dr. Nilgün Güler BAYAZIT (Y.T.Ü.)

İSTANBUL, 2008

İÇİNDEKİLER

	Sayfa
SİMGE LİSTESİ.....	iii
KISALTMA LİSTESİ	iv
ŞEKİL LİSTESİ.....	v
ÇİZELGE LİSTESİ	vi
ÖNSÖZ	vii
ÖZET	viii
ABSTRACT.....	ix
1. GİRİŞ VE TANIM.....	1
2. KURULAN SİSTEMİN GENEL MİMARİSİ.....	4
2.1 Eğitim Aşaması	4
2.2 Sınıflandırma / Test Aşaması	5
3. HAAR BENZERİ ÖZELLİKLER VE YÜZ ÇIKARTMA İŞLEMİ.....	6
3.1 Haar-Benzeri Özellikler	6
3.2 Kullanılan Karelerin Seçilmesi ve Yüz Bulmanın İyileştirilmesi.....	10
3.3 Bulunan Yüzlerde Hizalama	14
3.4 Normalizasyon ve Maskeleye	16
4. TEMEL BİLEŞENLER VE DOĞRUSAL DİSKRİMİNANT ANALİZLERİ	19
4.1 Temel Bileşenler Analizi	19
4.2 Doğrusal Diskriminant Analizi	22
5. DESTEK VEKTÖR MAKİNELERİ	26
5.1 Doğrusal Ayırma.....	26
5.2 Doğrusal Olmayan Ayırma ve Kernel Yöntemi	28
5.3 Çok Sınıflı Destek Vektör Makineleri	31
5.4 Videoları Sınıflandırmak için Destek Vektör Makineleri'nin Kullanımı	31
6. UYGULAMA VE DENEY	35
6.1 Kullanılan Veri Seti	35
6.2 Geliştirme Ortamı ve Kütüphaneler (C++ / OpenCV ve MATLAB / PRTOOLS).....	36
6.3 Yüz Bulma, İyileştirme, Ön İşlemler ve Oluşturulan Deney Setleri	37
6.4 TBA ve DDA ile Boyut Düşürme.....	39
6.5 DVM ile Sınıflandırma	40
7. SONUÇ VE YORUMLAR.....	43
KAYNAKLAR	45
ÖZGEÇMİŞ	46

SİMGE LİSTESİ

$i(.)$	Resmin nokta değeri
$K[.]$	Kovaryans matris
$Ker(.)$	Çekirdek fonksiyonu
$S_i[.]$	Sınıf içi saçılma matrisi
$S_A[.]$	Sınıflar arası saçılma matrisi
$V[.]$	Özvektör matrisi
w	Yüzey normal
y_i	Etiket değerleri
X	Eğitim seti
X_i	Veri değerleri
$Z[.]$	Dönüşüm matrisi
$\Phi(.)$	Dönüşüm fonksiyonu
$\lambda[.]$	Özdeğer dizisi

KISALTMA LİSTESİ

DDA	Doğrusal Diskriminant Analizi
DTBA	Döndürülmüş Toplam Alan Tablosu
DVM	Destek Vektör Makinesi
HBÖ	Haar-Benzeri Özellikler
TAT	Toplam Alan Tablosu
TBA	Temel Bileşen Analizi

ŞEKİL LİSTESİ

Şekil 2.1 Eğitim Aşaması.....	4
Şekil 2.2 Sınıflandırma / Test Aşaması	5
Şekil 3.1a Örnek Haar Benzeri Özellikler (Lienhart ve Maydt, 2002).....	7
Şekil 3.1b Örnek Haar Benzeri Özellikler (Lienhart ve Maydt, 2002).....	7
Şekil 3.1c Örnek Haar Benzeri Özellikler (Lienhart ve Maydt, 2002).....	7
Şekil 3.2 Toplam Alan Tablosu	8
Şekil 3.3 Döndürülmüş Toplam Alan Tablosu	9
Şekil 3.4 Örnek Seçilmiş Haar Benzeri Özellikler (Viola ve Jones, 2001)	10
Şekil 3.5 Yüz üzerinde ilk gözün konumunu bulmak.....	11
Şekil 3.6 İkinci gözün bulunması için yüzün ikiye ayrılması.....	12
Şekil 3.7 Farklı yarılarda bulunmuş iki göz.....	12
Şekil 3.8 Ağızın bulunması için yüzün ikiye ayrılması	13
Şekil 3.9 Beklenen konumlarda bulunan gözler ve ağız.....	13
Şekil 3.10 Gözler ve ağız oluşturduğu üçgenin yüz karesindeki konumu.....	14
Şekil 3.11 Gözleri birleştiren eksenin yatay eksenle yaptığı θ açısı.....	15
Şekil 3.12 θ açısı kadar döndürülmüş resim	15
Şekil 3.13 Döndürme sonrası yüzü çevreleyen karenin tekrar hesaplanması	16
Şekil 3.14 Yüzün maskelenmesi için kullanılan noktalar	17
Şekil 3.15 Bir video karesinde bulunan, hizalanan ve maskelenen yüz	18
Şekil 4.1 Örnek veri seti için hesaplanan yeni koordinatlar	19
Şekil 4.2 n özellikli m adet eleman için oluşturulmuş matris	20
Şekil 4.3 Şekil 4.2'deki X matrisinin ortalanmış (X') hali	21
Şekil 4.4 n x n kovaryans matris için, n uzunlukta n adet özvektör	21
Şekil 4.5 İlk z adet özdeğere karşılık gelen, n uzunlukta z adet özvektör	22
Şekil 4.6 Örnek veri seti için hesaplanan yeni koordinat.....	23
Şekil 4.7 r örnekli n boyutlu s sınıf için örnek matris.....	23
Şekil 4.8 Şekil 4.6'daki X matrisinin ortalanmış (X_A') hali	24
Şekil 4.9 Şekil 4.6'daki X matrisinin ortalanmış (X_i') hali	24
Şekil 5.1 Doğrusal olarak ayrılabilen veri setinde uygulanan iki farklı en uygun ayırıcı (yeşil çizgi) ve destek vektörlerin bulunduğu ayırıcılar (kesikli çizgiler)	26
Şekil 5.2 En uygun ayırıcı ve w vektörü	27
Şekil 5.3 Doğrusal ayrılamayan bir veri seti.....	28
Şekil 5.4 Üçüncü koordinat, X^2-Y^2 düzlemine dik bir koordinat olmak üzere yeni koordinat sistemi.....	29
Şekil 5.5 Eğitim Aşaması.....	33
Şekil 5.6 Test Aşaması.....	34
Şekil 6.1 Honda/UCSD Video Veritabanı'ndan Örnek Görüntüler	35
Şekil 6.2 OpenCV ile bulunmuş bir yüz bölgesi	36
Şekil 6.3 Başarı Oranları.....	42

ÇİZELGE LİSTESİ

Çizelge 3.1 Lienard ve Maydt'ın, Şekil 3.1'de gösterilen özelliklerden kullandıkları (Lienhart ve Maydt, 2002)	7
Çizelge 6.1 Eğitim veri setlerinin büyüklük ve oluşturulma süreleri	38
Çizelge 6.2 Test veri setlerinin bir video için ortalama büyüklük ve oluşturulma süreleri	38
Çizelge 6.3 Eğitim setleri için TBA ve DDA hesaplamaları sonuçları	40
Çizelge 6.4 Sınıflandırıcı süreleri ve başarı oranları	41
Çizelge 6.5 Sistemin toplam eğitim ve bir video için sınıflandırma süreleri (dakika)	42

ÖNSÖZ

Lisans eğitimimin son senesinde beni Veri Madenciliği ve Makine Öğrenmesi alanlarına yönlendirerek bu çalışmanın bugünlere gelmesindeki yolu açan ve bu çalışma boyunca da beni destekleyen ve yönlendiren Yrd. Doç. Dr. Nilgün Güler Bayazıt'a,

Çalışma boyunca manevi desteklerini benden esirgemeyen aileme,

Yüz tanıma konusundaki birikimlerini paylaştığı için Prof. Dr. Muhittin Gökmen'e,

Çalışmanın gerçekleştirilmesi esnasında sabırlı yardımlarını eksik etmeyen Sezin Karlı'ya,

Yüksek lisans ve çalışma süresince desteğini ve yardımını her zaman gördüğüm Deniz Adil Demirci'ye

Teşekkürü bir borç bilirim.

İbrahim Saygın TOPKAYA

ÖZET

Önemi ve kullanım alanı gün geçtikçe artan bir çalışma alanı olan yüz tanıma, önceden sisteme tanıtılan insanların sistem tarafından daha sonra otomatik olarak tanınmasını amaçlamaktadır.

Bu tezde ise, video görüntüleri üzerinde çalışan bir yüz tanıma sistemi kurulmuştur. Her bir videoda bir kişi olacak şekilde poz, açı ve dönme kısıtı olmadan elde edilen videolarla eğitilen sistem, daha sonra yine aynı şekildeki farklı videoların kime ait olduğunu tanımaya çalışmaktadır.

Çalışmanın temel amacı, videolardan yüz tanınırken tüm video karelerinin kullanılması yerine otomatik çalışabilecek bir sistem kurarak, bu sistemin seçeceği belli karelerin, belli ön işlemlerden geçerek kullanılması durumunda başarı oranı ve çalışma sürelerinin nasıl etkileneceğini incelemektir.

Tez metninde, sistem kurulurken yüz ve yüz öğelerini bulma amaçlı kullanılan Haar-Benzeri Özellikler, boyut dönüştürme amaçlı kullanılan Temel Bileşenler Analizi ile Doğrusal Diskriminant Analizi ve sınıflandırma amaçlı kullanılan Destek Vektör Makineleri algoritmalarıyla beraber bu algoritmaları temel alarak geliştirilen sistem tanıtılmış ve deney sonuçları ile başarı durumu ölçülmüştür.

Anahtar kelimeler: Haar-Benzeri Özellikler, Destek Vektör Makineleri, Yüz Tanıma, Yüz Bulma, Temel Bileşenler Analizi, Doğrusal Diskriminant Analizi

ABSTRACT

Face recognition is an application in computer science, importance and practical usage of which is rapidly increasing day by day. In simplest terms, face recognition aims to build a trained system which recognizes people, who are first introduced to system for training.

In this work, a system working on video records is constructed. First, system is trained with videos consisting only one person per each, however having no restriction in pose, angle and rotation. Then different videos with same situations are presented to system for recognition.

The basic aim in this study is to analyze; how -instead of using all video frames- using only a subset of frames, automatically selected and preprocessed by a system, affects success rate and processing time.

In thesis text, first algorithms for object and face detection, dimension transform and classification -which are Haar-Like Features, Principal Components Analysis & Linear Discriminant Analysis and Support Vector Machines respectively- are introduced. Following that, a system based on these algorithms is constructed and experimental results are presented.

Keywords: Haar-Like Features, Support Vector Machines, Face Recognition, Face Detection, Principal Components Analysis, Linear Discriminant Analysis

1. GİRİŞ VE TANIM

Yüz tanıma, yapay zeka çalışmalarında oldukça popüler olan ve pratik uygulamalardaki önemi gün geçtikçe artan bir çalışma alanıdır. En yalın haliyle yüz tanıma, *önceden sisteme tanıtılan insanların sistem tarafından daha sonra otomatik olarak tanınmasını* amaçlamaktadır.

Bilgisayarla görü, görüntü işleme ve örüntü tanıma gibi temel bilgisayar bilimleri dallarıyla ilişkili olan yüz tanımanın, yüzlerin sabit fotoğraflar veya hareketli görüntülerden alınması, görüntü alındığı esnada yüzlerin pozunu veya açısı, ortamın ışığı, yüzlerde sakal/bıyık benzeri doğal farklılaştırıcılar veya gözlük, atkı gibi aksesuarların bulunması gibi bir çok etkene bağlı olmasından dolayı her bir etkeni ele almaya yönelik farklı teknikler ve yöntemler geliştirilmiştir.

Günümüzde güvenlik uygulamalarında kullanımı yaygınlaşmış olan ve biyometrik fotoğraf olarak anılan, belli özel koşullar altında çekilen vesikalık fotoğraflar gibi şartların daha önceden sabitlendiği ve yüzün tüm tanımlayıcı öğelerinin (göz, burun, ağız vb.) net bir şekilde erişilebilir olduğu fotoğraflardan yüz tanıma karşın hareketli görüntülerden yüz tanıma halen daha, bir çok zorlukla karşılaşılacak bir uygulamadır. Çünkü bir önceki paragrafta bahsedilen neredeyse tüm etkenlerin karşılaşıldığı hareketli görüntülerden yüz tanıma, bu zorluklara karşın güvenlikten eğlenceye kadar bir çok konuda uygulama alanı bulabilir. Örneğin kameralardan çekilen güvenlik kayıtlarındaki kişilerin otomatik olarak tanınması, bir filmdeki oyuncuların otomatik olarak listelenmesi veya insan-makine etkileşimi gibi uygulamalar temelde hep hareketli görüntülerden yüz tanıma dayanır.

Kaynak ne olursa olsun (video, canlı kayıt, gerçek zamanlı görüntü vs.) hareketli görüntülerden yüz tanıma temelde iki farklı yaklaşımla incelenir (Hadid ve Pietikainen, 2004):

- **Her bir karenin bağımsız incelendiği yaklaşım:** Bu yaklaşım klasik fotoğraflardan yüz tanımadan farksız olup, her bir kare birbirinden bağımsız birer resim olarak ele alınır ve resimdeki kişi veya kişiler tanınmaya çalışılır. Her bir karenin bir gözlem olarak ele alındığı bu yapılarda, tekli gözlemler üzerinde çalışan sınıflandırıcılar (En Yakın K Komşu, Destek Vektör Makineleri vb.) kullanılır.
- **Karelerin Aynı Anda İncelendiği Yaklaşım:** Bu yaklaşımda ise kareler arasındaki ilişki de göz önüne alınarak karelerin, dolayısıyla karelerdeki aynı kişiye ait yüzlerin zamana göre konumları ve değişimleri incelenir. Burada Gizli-Markov Modeli gibi klasik makine

öğrenme yöntemlerinin yanında Yayılma Yaklaşımı (Condensation Approach) gibi özgün yöntemler veya yüzün üç boyutlu modellenmesi gibi çalışmalar gerçekleştirilmektedir.

Bu tez çalışmasında ise birinci yaklaşım temel alınarak bir model geliştirilmesi ve *“Her video bir kişiye ait olmak üzere; hareket ve pozda bir kısıtlama olmadan çekilen videolardaki kişileri tanımak için, videolardaki karelerin tümünü kullanmak yerine kareler arasında bir seçim ve yüz bileşenlerinde hizalama/normalizasyon işlemlerini otomatik yapan bir sistem nasıl kurulabilir?”* sorusunun cevabının bulunması amaçlanmaktadır.

Sistem, klasik bir öğreticili öğrenme sistemi gibi, öncelikle eğitim videoları olarak adlandırılan videoları kullanarak kişileri öğrenmeyi ve ardından test videoları olarak adlandırılan bir başka grup videodan ise hangi videonun kime ait olduğunu bulmayı hedeflemektedir. Bunun için öncelikle videolardaki her kare tek tek işlenerek insan yüzleri çıkartılmakta, daha sonra bulunan insan yüzleri arasında yüzün gerekli tüm öğelerinin (gözler ve ağız) açıkça bulunup bulunmamalarına göre bir seçim yapılmaktadır. Sistemin tamamen kendi kendine çalışabilir olması amaçlandığından bu seçim işlemi belli bir algoritma çerçevesinde yapılmakta ve sistemin çalışması esnasında işleme müdahale edilmemektedir.

Videodan elde edilen tüm insan yüzleri içeren kareler yerine, sistemin gerçekleştirdiği bu seçim sonunda elde edilen kareler kullanıldığı için, aynı video daha az sayıda ama daha rafine bilgi içeren sayıda kare ile temsil edilir hale getirilmektedir. Ayrıca elde edilen bu karelerde yüz özelliklerinin hizalanması, arka planın etkisinin kaldırılması ve ışıklandırmadan kaynaklanan farkların giderilmesi gibi bir takım görüntü işleme ön işlemleri gerçekleştirilmekte ve böylece bir önceki adımda bahsedilen ayırt edici karelerin bir standart içerisinde ele alınması sağlanmaktadır.

Elde edilen bu yüz kareleri ile bir sınıflandırıcı sistem eğitilmekte, yine aynı seçim sistemi sonucunda test videolarından elde edilen ayırt edici kareler de bu sınıflandırıcı tarafından tanınmaya çalışılmaktadır. Bu aşamada karelerin hangi videodan elde edildiği bilgisi de sistem tarafından kullanılmakta ve karelerin birbirlerinden bağımsız işlenmesi yerine bir videodaki kareler bir bütün halinde incelenebilmektedir.

Tez metni şu şekilde düzenlenmiştir; sistemin mimarisi hakkında genel bir bilgi verildikten sonra (Bölüm 2), öncelikle videolardaki karelerde yüz bölgelerinin bulunması sırasında kullanılan Haar Benzeri Özellikler algoritması ve bu algoritma kullanılarak video karelerinde yüz bölgesinin bulunması esnasında yapılan seçim ve ön işlemler (Bölüm 3), ardından veri setlerinde boyut düşürme için kullanılan Temel Bileşenler ve Doğrusal Diskriminant

Analizleri (Bölüm 4) ve son olarak da sınıflandırma aşamasında kullanılan Destek Vektör Makineleri algoritması ve bu algoritmaya dayalı sınıflandırıcı (Bölüm 5) kısaca tanıtılmıştır.

Daha sonra ise testte yapılan çalışma anlatılmış (Bölüm 6), öncelikle belirlenen bir deney setinden Bölüm 3'deki yöntemler ile çıkartılan yüzler ve oluşturulan deney setleri anlatılmıştır. Aynı sıra takip edilerek bir önceki aşamada elde edilen veri setinin nasıl daha düşük boyutta bir uzaya indirildiği açıklanmış ve ardından kurulan sınıflandırıcı sunulmuştur. Önceki iki aşama sonunda elde edilen farklı setler ile elde edilen farklı sonuçlar bu bölümde gösterilmiş ve sonuçların yorumlanması da sonuç bölümüne (Bölüm 7) bırakılmıştır.

2. KURULAN SİSTEMİN GENEL MİMARİSİ

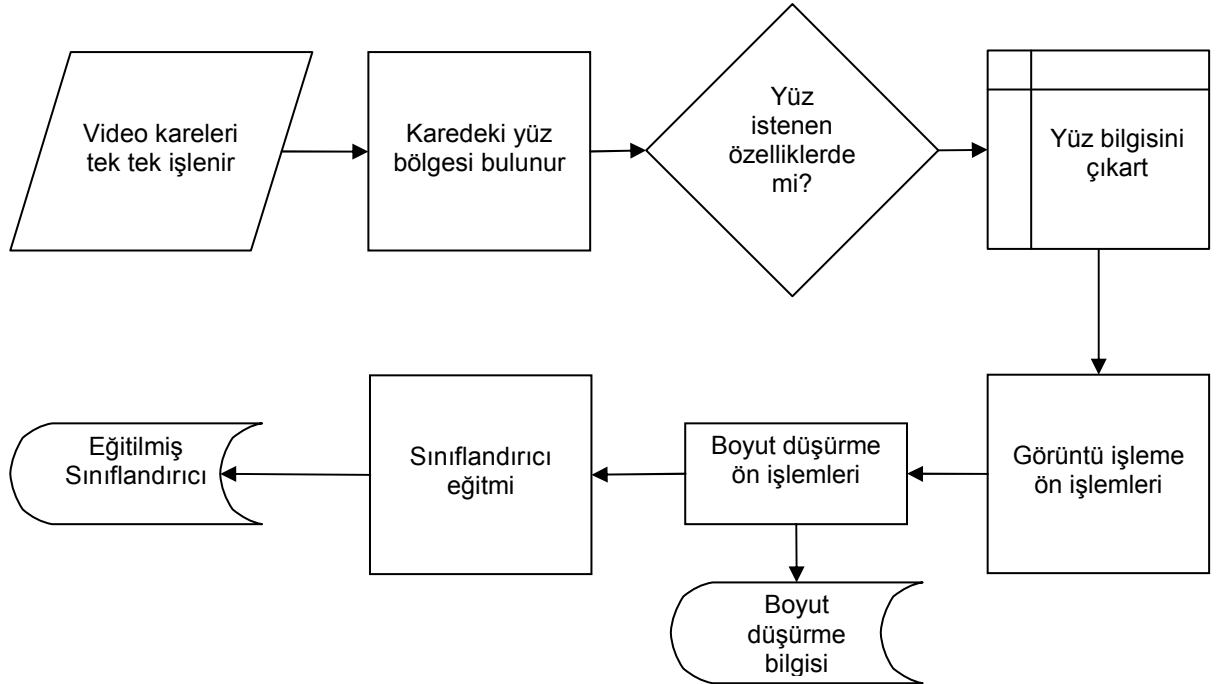
Kurulan sistem, eğitim ve test aşamasında benzer adımları gerçekleştirerek, bir sınıflandırıcı eğitir ve bu sınıflandırıcı ile yeni videoların tanınması gerçekleştirilir. Kullanılan algoritmaların detaylı anlatımları sonraki bölümlere bırakılmak üzere, sırasıyla eğitim ve test aşamalarının çalışma düzenlerine bu bölümde kısaca değinilecektir.

2.1 Eğitim Aşaması

Eğitim aşaması, sistemde daha önceden kime ait oldukları bilinen kişilerin videolarının işlenmesi ve bu videolardan elde edilerek ön işlemlerden geçirilen veriler ile bir sınıflandırıcı eğitilmesi işlemlerini kapsamaktadır. Bu aşamanın işlemleri kısaca şu şekilde özetlenebilir:

1. Tüm eğitim videoları için tüm kareler tek tek işlenir. Her karede varsa yüz bölgesi bulunur ve ayırt edici bir kare ise ön işlemlerden geçirilir.
2. Seçilen ve ön işlemlerden geçirilen yüz resimlerinden oluşturulan veri setinde boyut dönüşümü için gerekli değerler hesaplanır ve dönüşüm uygulanır.
3. Boyut dönüşümü uygulanmış eğitim seti üzerinde bir sınıflandırıcı eğitilir.

Aynı işlemler bir akış şemasında ifade edilirse, aşağıdaki gibi bir şema elde edilir:



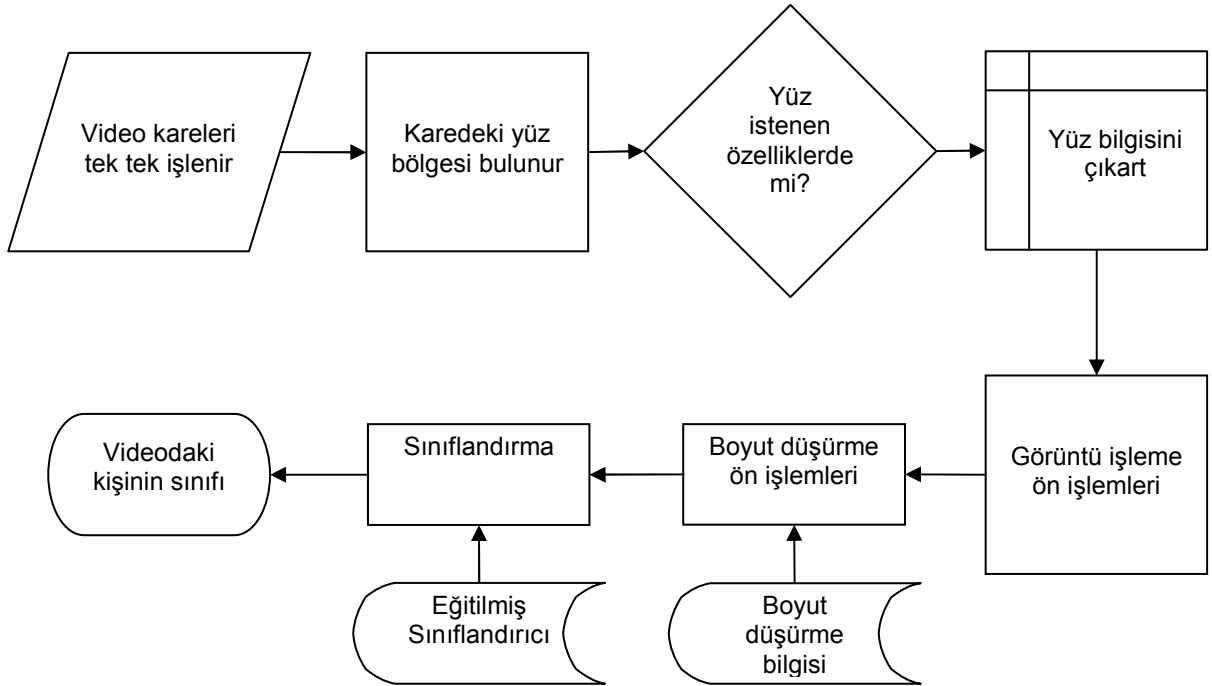
Şekil 2.1 Eğitim Aşaması

2.2 Sınıflandırma / Test Aşaması

Sınıflandırıcı eğitimi ile tamamlanan eğitim aşaması sonrasında, sistem tarafından bir videonun tanınması ise aşağıdaki şekilde gerçekleştirilir:

1. Videonun her karesini işleyerek varsa ayırt edici yüz kareleri seçilir ve ön işlemler uygulanır.
2. Eğitim aşamasında hesaplanan dönüşüm değerleri, seçilen ve ön işlemlerden geçirilen karelerden oluşturulan veri setine uygulanır.
3. Her kare tek tek sınıflandırılır ve tüm karelerin sınıf bilgileri kullanılarak videonun sınıfı hesaplanır.

Bu aşamanın işlemleri bir akış şemasında ifade edilirse, aşağıdaki gibi bir şema elde edilir:



Şekil 2.2 Sınıflandırma / Test Aşaması

Birinci adımdaki yüz ve yüz öğeleri bulma aşamasında Haar-Benzeri Özellikler, boyut düşürme aşamasında Temel Bileşenler ile Doğrusal Diskriminant Analizleri ve sınıflandırıcı aşamasında da Destek Vektör Makineleri algoritmaları tercih edilmiştir. Takip eden bölümlerde öncelikle bu algoritmalar hakkında bilgi verilmekte ve sistemde nasıl kullanıldıkları anlatılmaktadır.

3. HAAR BENZERİ ÖZELLİKLER VE YÜZ ÇIKARTMA İŞLEMİ

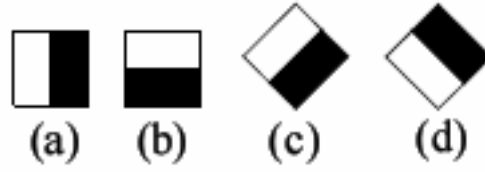
Paul Viola ve Michael Jones'un, resimlerde istenen bir nesneyi bulma amacıyla geliştirdikleri ve Haar-Benzeri Özellikler (HBÖ) (Haar-Like Features) olarak adlandırdıkları özelliklerin çıkartılmasına dayalı nesne bulma yöntemi (Viola ve Jones, 2001), özel olarak insan yüzünü bulunması amaçlanan bir nesne olarak kabul edip, insan yüzü örnekleri ile eğitildiğinde, resimlerdeki insan yüzlerini bulmak amacıyla kullanılabilir. Yöntem daha sonra Intel Laboratuvarları'ndan Rainer Lienhart ve Jochen Maydt tarafından iyileştirilerek (Lienhart ve Maydt, 2002), tezde kullanılmış olan OpenCV kütüphanesindeki haline getirilmiştir. Çalışmanın bu aşamasında ise yüz bulma aşamasında HBÖ temelli bir yüz bulma ve bulunan yüzde yine HBÖ kullanarak bulunan yüz özelliklerini (göz, ağız vs.) hizalayarak normalize eden bir sistem geliştirilmesi amaçlanmaktadır.

3.1 Haar-Benzeri Özellikler

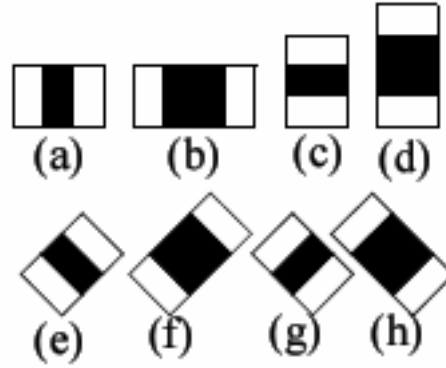
Bir resmin komşu nokta bölgeleri için Haar tabanlı fonksiyonların çıktılarına benzer çıktılar üreten bu yöntem, bu çıktılardan ürettiği özellikleri kullanarak bir sınıflandırıcı eğittikten sonra, bu sınıflandırıcı verilen bir resmin (veya resim bölgesinin) aranan nesne –bu çalışma için insan yüzü ve insan yüzündeki öğeler- olup olmadığına karar vermektedir. Dolayısıyla büyük bir resimde belli bir boyutta bir pencere dolaştırılarak, pencerenin o an için bulunduğu konumun insan yüzü olup olmadığı kararı ile, resimdeki yüz bölgelerinin sınırları, daha sonra bulunan yüz bölgesinde daha küçük boyutta pencereler dolaştırılarak da yüz bölgesindeki yüze ait öğeler bulunabilmektedir.

HBÖ üretilirken, birbirlerine komşu iki veya üç nokta bölgesinin içinde kalan noktaların değerlerinin toplamının aralarındaki fark alınmaktadır. Örneğin iki bölgeli bir özellik (Şekil 3.1a) hesaplanırken her iki bölgedeki noktaların değerlerinin toplamının farkı, üç bölgeli bir özellik (Şekil 3.1b) hesaplanırken dışta kalan iki bölgedeki noktaların toplamının aradaki bölge içerisindeki noktaların toplamına olan farkı alınmaktadır.

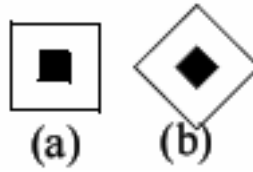
Viola ve Jones'un kullandığı yatay ve dikey komşu bölgelerin (Şekil 3.1a: a,b ve Şekil 3.1b: a,b,c,d) dışında, Lienard ve Maydt bu bölgelerin 45° döndürülmüş hallerini (Şekil 3.1a: c,d ve Şekil 3.1b: e,f,g,h) ve biri diğerinin içinde kalan bölgeleri (Şekil 3.1c) de kullanmaktadır. Örneğin 24x24 boyutlarındaki bir resimde Çizelge 3.1'de görüldüğü gibi 117.000'den fazla özellik bulunabilmektedir.



Şekil 3.1a Örnek Haar Benzeri Özellikler (Lienhart ve Maydt, 2002)



Şekil 3.1b Örnek Haar Benzeri Özellikler (Lienhart ve Maydt, 2002)



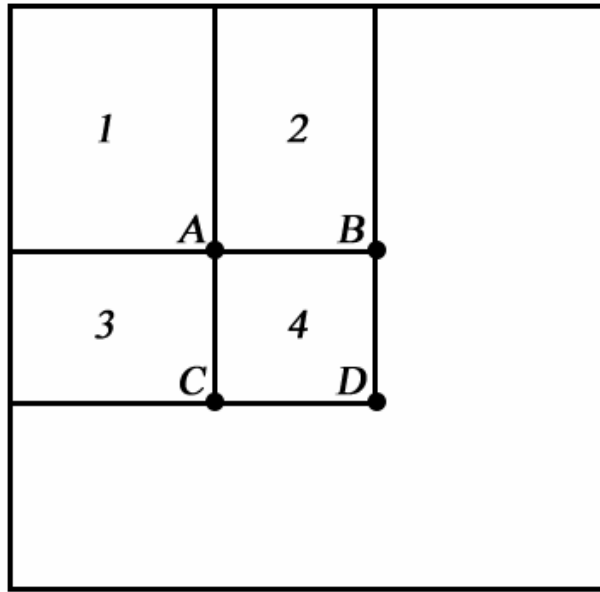
Şekil 3.1c Örnek Haar Benzeri Özellikler (Lienhart ve Maydt, 2002)

Çizelge 3.1 Lienard ve Maydt'in, Şekil 3.1'de gösterilen özelliklerden kullandıkları (Lienhart ve Maydt, 2002)

Şekil 3.1a		Şekil 3.1b	
a,b	43.200	a,c	27.600
c,d	8.464	b,d	20.736
Şekil 3.1c		e,g	4.356
a	8.464	f,h	3.600
b	1.521	TOPLAM	117.941

Komşu bölgelerin nokta değerlerinin farkları hesaplanırken, Toplam Alan Tablosu (TAT) adı verilen bir yöntemden faydalanılması toplamların bulunmasında büyük fayda sağlamaktadır. Bir resmin TAT'nda, o resimdeki her bir nokta için değer olarak o noktanın solunda ve yukarısında kalan noktaların değerlerinin toplamları atanmaktadır.

$$TA(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y') \quad (3.1)$$



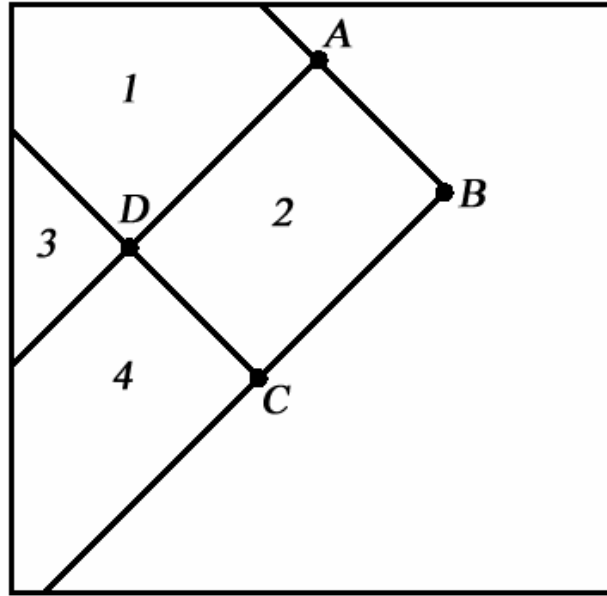
Şekil 3.2 Toplam Alan Tablosu

Şekil 3.2'deki örneğe bakılırsa, A noktası için Toplam Alan değeri 1. bölgedeki noktaların toplamı, D bölgesi içinse 1., 2., 3. ve 4. bölgedeki noktaların toplamı olarak hesaplanmaktadır.

Bu şekilde bir resim için bir kere TAT hesaplanarak, daha sonraki HBÖ hesaplanırken, farkı alınmak istenen bölgelerin sağ alt köşelerindeki noktalara ait Toplam Alan değerleri kullanılarak, tekrar tekrar noktaların değerlerini toplamak gereksinimi ortadan kaldırılmaktadır. Tekrar Şekil 3.2 örneğine bakılırsa, bir kez TAT hesaplandıktan sonra 1. ve 2. bölgedeki noktaların değerlerinin toplamlarının farkı alınmak istenirse, TAT'ndaki A ve (B – A) nokta değerlerinin farkını almak yeterli olmaktadır.

TAT'nun yatay ve dikey komşu bölgelerinin kullanıldığı özellikleri hesaplarken kullanışlı olması gibi, benzeri şekilde bir resim için Döndürülmüş Toplam Alan Tablosu (DTAT)

kullanılarak da, 45° döndürülmüş komşu bölgelerin kullanıldığı özelliklerin kolay bir şekilde hesaplanabilir.



Şekil 3.3 Döndürülmüş Toplam Alan Tablosu

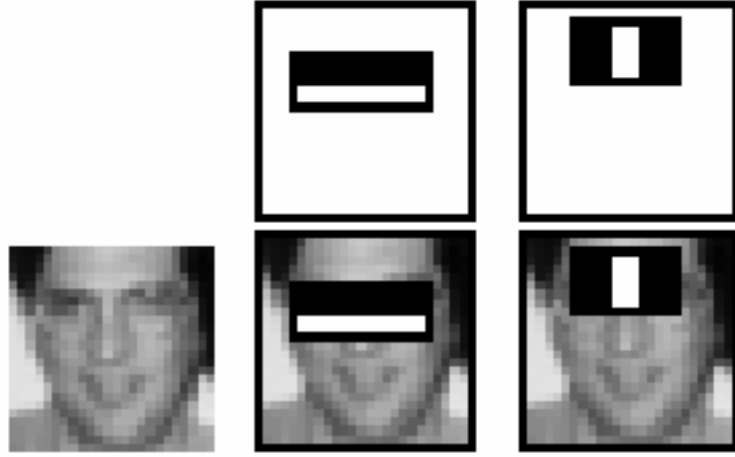
Şekil 3.3'teki örneğe bakılırsa, D noktası için Döndürülmüş Toplam Alan değeri 3. bölgedeki noktaların toplamı, B bölgesi içinse 1., 2., 3. ve 4. bölgedeki tüm noktaların toplamı olarak hesaplanmaktadır.

Bu şekilde bir resim için bir kere DTAT hesaplanarak, daha sonraki HBÖ hesaplanırken, farkı alınmak istenen bölgelerin sağ köşelerindeki noktalara ait Döndürülmüş Toplam Alan değerleri kullanılarak, tekrar tekrar noktaların değerlerini toplamak gereksinimi ortadan kaldırılmaktadır. Tekrar Şekil 3.3 örneğine bakılırsa, bir kez DTAT hesaplandıktan sonra 1. ve 3. bölgedeki noktaların değerlerinin toplamlarının farkı alınmak istenirse, DTAT'ndaki D ve (A – D) nokta değerlerinin farkını almak yeterli olmaktadır.

Daha önce de söz edildiği gibi, ufak bir resimde bile çok sayıda HBÖ hesaplanabilmektedir. Bu yüzden bulunan tüm bu özellikleri kullanmak yerine bunlar içerisinde ayırt edici olan çok daha düşük sayıda özelliğin kullanılması işlemin hızı açısından önemlidir.

Hangi özelliklerin kullanılacağı belirlenmesinde, bulunması istenen nesneye ait (olumlu örnek) ve farklı nesnelere ait (olumsuz örnek) bir çok resim kullanılarak, bir sınıflandırıcı ile özellikler arasında eleme yapılabilir. Viola ve Jones çalışmalarında Karar Ağaçları'ndan oluşan bir AdaBoost sınıflandırıcı ile özelliklerden 6.601 tanesini kullanılır olarak tespit

etmişlerdir. Örneğin bu özelliklerden iki tanesi Şekil 3.4’te görüldüğü gibi göz ve üst burun bölgesindeki nokta değerlerini kullanarak ayırt etme işlemini gerçekleştirmektedir.



Şekil 3.4 Örnek Seçilmiş Haar Benzeri Özellikler (Viola ve Jones, 2001)

Yöntemin asıl tanıma gerçekleştirdiği kısım ise, bir önceki aşamada bulunan özellikleri kullanarak, incelenen resmin aranmakta olan nesneye ait olup olmadığına karar verme aşamasıdır. Bu noktada yine AdaBoost yöntemi ile kurulan bir ardışık zayıf sınıflandırıcılar dizisi ile, olumsuz örnekleri sistemi fazla meşgul etmeyecek şekilde daha ilk aşamalarda eleyerek tanıma aşamasında hız kazanma amaçlanmaktadır. Viola ve Jones’un çalışmasında, en az iki yaprağa sahip karar ağaçları kullanılarak kurulan bir AdaBoost sınıflandırıcı tercih edilmiştir.

3.2 Kullanılan Karelerin Seçilmesi ve Yüz Bulmanın İyileştirilmesi

Denemelerde görülmüştür ki, sadece yüz nesnesini bulma amaçlandığında, doğru olumlu tespit sayısı oldukça yüksek olsa da, özellikle arka planda karmaşık görüntüler içeren ortamlarda çekilen resimlerde yanlış olumlu tespit sayısı da yüksek olabilmektedir. Yüz bulma aşamasının ürettiği çıktı (belli işlemlerden geçtikten sonra) yüz tanıyan sınıflandırıcının girdi verisi olarak kullanıldığı için yüz bulma aşamasının iyileştirilmesi sistemin genel başarısı için önemli bir gerekliliktir. Ancak sunulan sistemin, bulunan tüm yüzleri kullanmayıp bunlar arasında bir seçimde bulunması, aslında bulunan yüzlerin gerçekten yüz olup olmadığına dair bir kontrol adımı da olmakta ve zaten bu adım sayesinde yanlış olumlu tespitler de azalmaktadır.

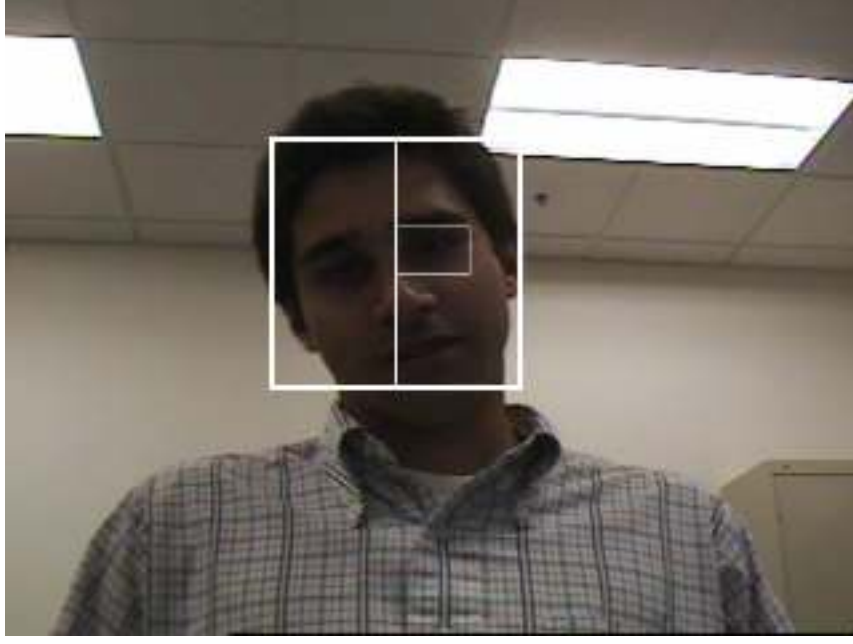
Bir insan yüzü göz önüne alındığında, bir yüz nesnesinin üzerinde karşılıklı iki göz, ve bu gözlerin merkezlerini birleştiren eksene dik bir eksende ve gözleri birleştiren eksenle karşılıklı

bir ağız nesnesi bulunması gerektiği açıktır. Ekran düzlemi X-Y düzlemi olarak kabul edilirse, yüzün bu düzlem içerisindeki hareketleri ve dönmeleri ağız ve gözün görünülebilirliğini etkilememektedir. Ancak ekrana dik bir Z ekseninde yüzün döndüğü durumlarda bu öğelerin görünüşleri bozulabilir, hatta profil resimlerinde olduğu gibi kimi zaman bazı yüz öğelerinin tamamen görünemediği durumlar bile ortaya çıkabilir.

İşte sistem de tüm bu öğelerin yüz üzerinde tam olarak görüldüğü kareleri seçmeyi hedeflemekte ve yüz bulmanın yanı sıra, benzer mantıkla göz ve ağız bulma işlemleri de sistem tarafından gerçekleştirilmektedir. Bu yüzden, bir resim parçası yüz olarak tespit edildikten sonra bu yüzün üzerinde bir göz nesnesi aranmakta, bulunduğu takdirde sistem bu gözü bir yarıda kalmak üzere yüzü ikiye ayırarak bulunan gözün olduğu kısmın diğer yarısında ikinci bir göz aramaktadır.



Şekil 3.5 Yüz üzerinde ilk gözün konumunu bulmak



Şekil 3.6 İkinci gözün bulunması için yüzün ikiye ayrılması

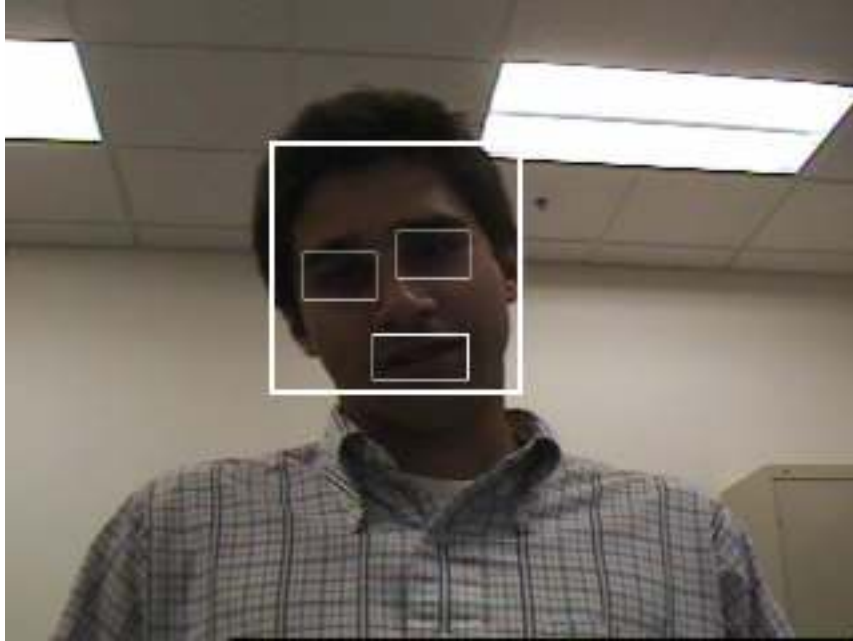


Şekil 3.7 Farklı yarılarında bulunmuş iki göz

İkinci göz bulunduktan sonra ise sistem yüzü, her iki göz de aynı yarıda kalmak üzere ikiye ayırarak, bulunan gözlerin olduğu kısmın diğer yarısında ağzı aramaktadır.

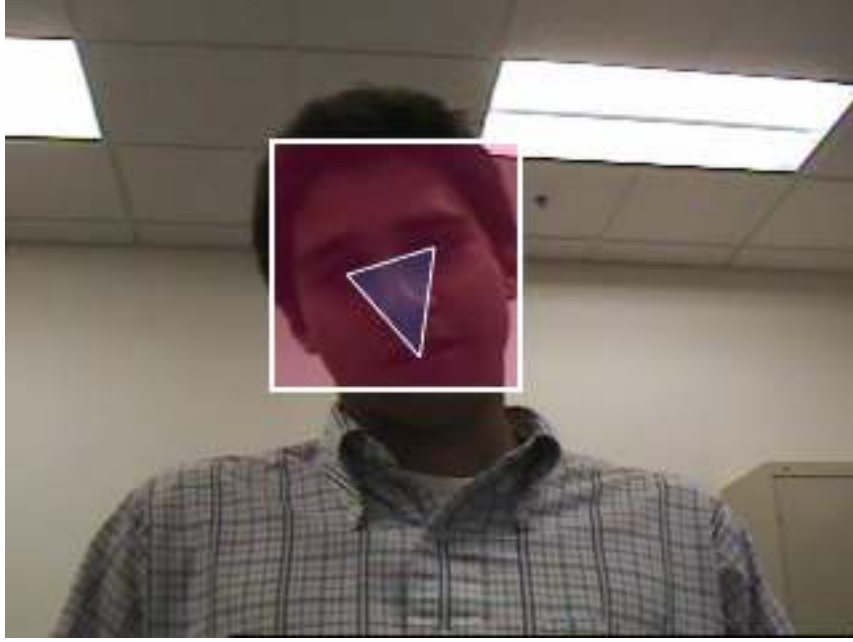


Şekil 3.8 Ağızın bulunması için yüzün ikiye ayrılması



Şekil 3.9 Beklenen konumlarda bulunan gözler ve ağız

En sonunda ise bulunan iki göz ve ağız merkezlerinin oluşturduğu üçgenin, yüze ait bulunan bölgenin belli bir kısmını kapsayıp kapsamadığı kontrol edilmekte ve belli bir oranı karşılamadığı durumdaki kareler de elenmektedir.



Şekil 3.10 Gözler ve ağızın oluşturduğu üçgenin yüz karesindeki konumu

Sonuç olarak bir videodan, tanıma aşamasına olumlu katkıda bulunmayan karelerle beraber yanlış şekilde yüz olarak tespit edilen resimleri içeren kareler de elenmelerine rağmen, yüzle ilgili tüm bilgileri içeren ve doğru şekilde yüz olarak tespit edilen kimi karelerin de elenebildiği öngörülebilmektedir. Ancak bir videodaki ardışık karelerden hepsinin eğitim ve tanımda kullanılmasının gerekmediği, yanlış olumlu tespitlerden ve yüz hakkında eksik bilgi içeren karelerden kurtulabilme adına, ardışık karelerden bazılarının doğru olumlu tespit edilmesinden feragat edilebileceği ve seçilen az sayıdaki karenin bile sınıflandırıcı eğitimi ve tanınmasında yeterince olumlu etkisinin olduğu düşünülmektedir. Ayrıca bu esnada yüz üzerindeki iki gözün merkez konumlarının bulunması da ileride görüleceği gibi yüz resminin hizalanması ve maskelenmesi aşamasında kullanılmaktadır.

3.3 Bulunan Yüzlerde Hizalama

Sistem yukarıda anlatılmış olan sağlama işlemi sonucunda, anlatıldığı şekilde önden görünen yüz olduğu onaylanan resimlerde, gözler aynı hizada olmak üzere bir döndürme gerçekleştirilmektedir.

İki gözü birleştiren eksenin, resmin tabanı ile yaptığı açı θ olarak kabul edilirse, sistem tüm resmi $-\theta$ açısı kadar döndürmektedir. Döndürme işlemi sırasında yüz sınırlarını belirleyen bölge de döndüğünden, bu bölgenin de ufak bir hesaplamayla düzgün bir kare olmak üzere yeniden hesaplanması gerekmektedir.

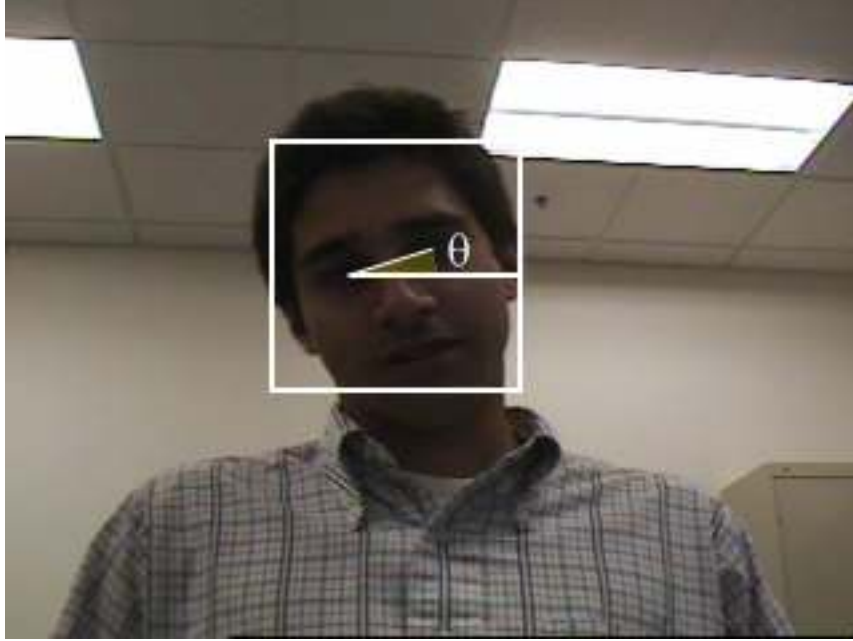
Yüz sınırlarını belirleyen bölgenin, döndürüldükten sonraki halinin köşeleri sırası ile (x_1, y_1) , (x_2, y_2) , (x_3, y_3) ve (x_4, y_4) ise, yeni karenin sol üst köşe koordinatları:

$$X = \min(x_1, x_2)$$

$$Y = \min(y_1, y_2)$$

(3.2)

denklemleri ile hesaplanmaktadır.



Şekil 3.11 Gözleri birleştiren eksenin yatay eksenle yaptığı θ açısı



Şekil 3.12 θ açısı kadar döndürülmüş resim

Kare bölgenin bir kenarı ise:

$$a = \max(\max(x_3, x_4) - X, \max(y_3, y_4) - Y) \quad (3.3)$$

denklemleri ile hesaplanmaktadır.



Şekil 3.13 Döndürme sonrası yüzü çevreleyen karenin tekrar hesaplanması

Burada, bu bölgenin döndürülmeden önceki kenarına a_0 denirse, döndürüldükten sonraki kenar uzunluğunun:

$$a_1 = a_0 * \cos(\theta) \quad (3.4)$$

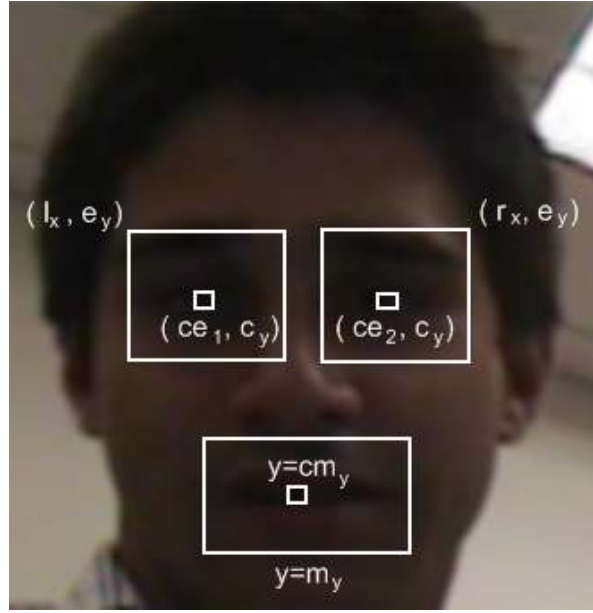
olması gerektiğinden, hesaplanan (döndürüldükten sonra düzgün kare haline getirilen) bölgenin kenarları a_1 olmak üzere uzatılmakta veya kısaltılmaktadır.

3.4 Normalizasyon ve Maskeleye

Sistem, bir yüz üzerinde gözleri ve ağzı anlatıldığı gibi bulduktan sonra, resme histogram eşitleme uygulayarak aydınlanma farkından kaynaklanan olumsuzlukları ortadan kaldırmayı amaçlamaktadır. Histogram eşitleme işleminden sonra ise sistem resmi, gözleri ve ağzı vurgulayan bir elips ile çevrelemektedir. Çalışmanın ilk aşamalarında gözlerin merkezlerini birleştiren doğrunun tam orta noktasını merkez ve eksen uzunluklarını sırası ile gözler arasındaki mesafe ve gözler ile ağız bölgesi arasındaki mesafe olarak alan bir elips

düşünülmekte idiysen de, bu elipsin ağız çevresinde çok az bir bölgeyi taradığı ve büyük oranda bilgi kaybına yol açtığı farkedildiğinden, elips bölgesi hesaplanmadan önce bulunan noktalardan türetilen bir takım yardımcı noktalar kullanılmaktadır.

Sol gözün bulunduğu bölgenin sol üst köşesi (l_x, e_y) , sağ gözün bulunduğu bölgenin sağ üst köşesi (r_x, e_y) , gözlerin merkezleri sıra ile (ce_1, c_y) and (ce_2, c_y) , ağız bölgesinin alt sınırının Y koordinat değeri m_y ve ağız merkezinin Y koordinat değeri cm_y olarak tanımlanırsa, çizilen elipsin denklemi:



Şekil 3.14 Yüzün maskelenmesi için kullanılan noktalar

$$x_1 = l_x - \frac{(ce_2 - ce_1)}{4}$$

$$x_2 = r_x + \frac{(ce_2 - ce_1)}{4}$$

$$y_1 = e_y - \frac{(cm_y - c_y)}{4}$$

$$y_3 = m_y + \frac{(cm_y - c_y)}{4}$$

$$h = x_1 + ((x_2 - x_1)/2)$$

$$k = y_1$$

$$a = \frac{x_2 - x_1}{2}$$

$$b = y_3 - y_1$$

$$(X, Y): \frac{(X - h)^2}{a^2} + \frac{(Y - k)^2}{b^2} > 1$$

(3.5)

olmaktadır. Bu şekilde çizilen bir elips bölgesi özellikle burun-ağız arasındaki bölgedeki yumuşak geçişi sağlamak amacıyla işe yaramasına rağmen, gözler çevresindeki bölgede gereğinden fazla bir alanı tarayabilmektedir. Bu yüzden, ikinci bir sınır koşulu da verilerek burundan yukarıdaki bölgede elipsden daha dar bir dikdörtgen bölge sınır olarak kabul edilmektedir. Bu bölgenin üst sınırları ise bulunan gözleri belirleyen bölgelerin kenarlarına bağlı olmakta ve:

$$(X, Y): X \geq l_x \wedge X \leq r_x \wedge Y \geq y_1 \quad (3.6)$$

şeklinde ifade edilmektedir. Sonuçta çene etrafında bir elips, gözler ve burun etrafında ise bir dikdörtgenle sınırlanmış bir yüz elde edilmektedir.

Sistem burada dikey eksen uzunluğu yatay eksen uzunluğundan kısa olan resimleri de eleyerek, yatay eksen uzunluğunu dikey eksene eşitleyip bir kare elde etmek üzere resmin her iki yanına eşit genişlikte siyah alan eklemektedir. Bu koşullar altında, gözlerin konumları ve aralarındaki mesafeye göre, yüz bölgesi sadece ayırt edici kısımları kalmış ve arka plandaki gereksiz bilgi de temizlenmiş şekilde çıkartılmış olmaktadır.



Şekil 3.15 Bir video karesinde bulunan, hizalanan ve maskelenen yüz

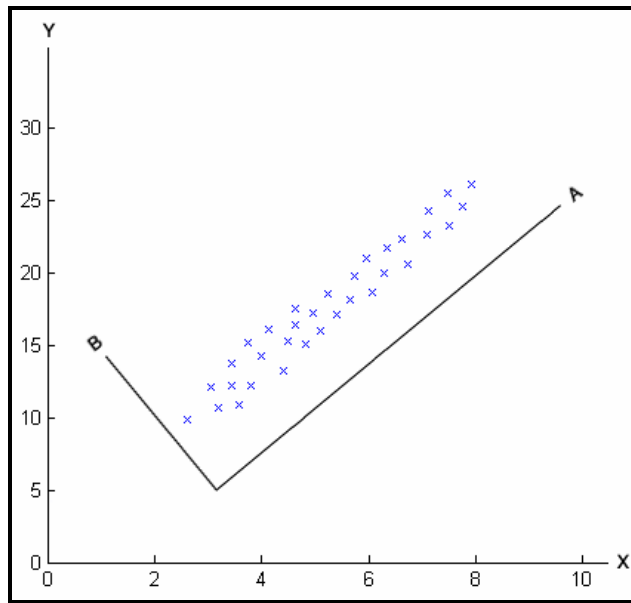
4. TEMEL BİLEŞENLER VE DOĞRUSAL DİSKRİMİNANT ANALİZLERİ

Modern makine öğrenme yöntemleri, yüksek yoğunlukta matematiksel işlemler gerçekleştirdikleri için, özellikle görüntü gibi yüksek boyutlu verilerin kullanıldığı durumlarda veri setinin, boyut sayısının düşürülmesine yönelik bir ön işlemden geçirilmesi, sınıflandırıcı eğitimi aşamasını büyük ölçüde kısaltmaktadır. Özellikle bu çalışmada da kullanılmakta olan Destek Vektör Makineleri gibi en uygun parametre değerlerinin daha önceden kestirilemediği ve aynı eğitim sürecinin farklı parametrelerle tekrar tekrar denendiği yöntemlerde bu zaman kazancının önemi büyüktür.

Çalışmada popüler boyut düşürme yöntemleri olan Temel Bileşenler Analizi (TBA) (Principal Component Analysis / PCA) ve Doğrusal Diskriminant Analizi (DDA) (Linear Discriminant Analysis / LDA) kullanılmaktadır.

4.1 Temel Bileşenler Analizi

Aynı zamanda Karhunen-Loève Dönüşümü veya Hotelling Dönüşümü olarak anılan TBA, yüksek boyutlu bir veri setini, doğrusal olarak ayrıştırarak, aynı boyut sayısında ancak farklı koordinat sistemine sahip bir uzayda tanımlamayı amaçlar. Yöntemin en önemli özelliği, tanımladığı yeni koordinat sistemindeki boyutların veri setinin varyansı ile doğru orantılı olması olup, boyutlardan az bir kısmının veri setinin varyansının büyük bir kısmını kapsamamasından dolayı düşük varyanslı boyutların kullanılmayarak veri setinde büyük ölçüde boyut kazancı sağlanmasıdır (Smith 2002).



Şekil 4.1 Örnek veri seti için hesaplanan yeni koordinatlar

Örneğin Şekil 4.1'e bakılırsa, X-Y koordinat sisteminde tanımlı noktalar, (X,Y) şeklinde ifade edilebilirken, dönüştürülmüş A-B koordinat sisteminde (A,B) koordinatları ile ifade edilebilmektedir. Ancak dikkat edilirse A-B koordinat sisteminde noktaların B eksenindeki değişimleri A eksenine göre oldukça azdır. Dolayısıyla bu setin varyansının büyük bir kısmı A eksenini tarafından kapsanmakta olup, noktaların sadece A koordinatıyla bile tanımlanması durumunda veri setinde büyük bir bilgi kaybı olmamaktadır.

Sonuç olarak, daha önceden (X,Y) gibi iki boyutta ifade edilebilen set, sadece (A) gibi tek bir boyutta ifade edilebilir hale gelmekte, boyut sayısı yarıya düşürülmesine rağmen verideki kayıp boyut sayısındaki kazancın yanında oldukça düşük kalmaktadır. İşte TBA de, yüksek boyutlu bir veri setini örnekteki gibi yeni koordinatlar cinsinden ifade etmeyi, ve düşük varyanslı koordinatların uygulamada dikkate alınmayarak boyut düşürülmesini sağlamayı amaçlamaktadır.

TBA hesaplanırken, öncelikle veriyi, her satırda bir eleman ve her sütunda bir özellik/boyut olmak üzere bir matriste ifade etmek gerekmektedir.

$$X = \begin{bmatrix} x_{11} & x_{12} & x_{13} & \dots & x_{1n} \\ x_{21} & x_{22} & x_{23} & \dots & x_{2n} \\ x_{31} & x_{32} & x_{33} & \dots & x_{3n} \\ \dots & \dots & \dots & \dots & \dots \\ x_{m1} & x_{m2} & x_{m3} & \dots & x_{mn} \end{bmatrix}$$

Şekil 4.2 n özellikli m adet eleman için oluşturulmuş matris

Satır formunda bir matriste toplanan veride, her bir sütun verideki elemanların bir özelliğini temsil etmektedir. Dolayısıyla, her bir sütun/özellik için tüm veri setindeki o özelliğin ortalama değeri hesaplanabilmektedir:

$$\overline{X}_i = \frac{\sum_{j=1}^m X_{ji}}{m} \quad (4.1)$$

Her özelliğin ortalaması bulunduktan sonra, tüm elemanlarda, her özellikten o özelliğin ortalaması çıkartılarak, veri seti sıfıra ortalananmış hale getirilir:

$$X' = \begin{bmatrix} \overline{x_{11} - x_1} & \overline{x_{12} - x_2} & \overline{x_{13} - x_3} & \dots & \overline{x_{1n} - x_n} \\ \overline{x_{21} - x_1} & \overline{x_{22} - x_2} & \overline{x_{23} - x_3} & \dots & \overline{x_{2n} - x_n} \\ \overline{x_{31} - x_1} & \overline{x_{32} - x_2} & \overline{x_{33} - x_3} & \dots & \overline{x_{3n} - x_n} \\ \dots & \dots & \dots & \dots & \dots \\ \overline{x_{m1} - x_1} & \overline{x_{m2} - x_2} & \overline{x_{m3} - x_3} & \dots & \overline{x_{mn} - x_n} \end{bmatrix}$$

Şekil 4.3 Şekil 4.2'deki X matrisinin ortalanmış (X') hali

Sıfıra ortalanmış matris, kendi devriği ile çarpılarak, kovaryans matrisi hesaplanır:

$$K = X'^T \cdot X' \quad (4.2)$$

n özellikten oluşan bir veri setinde, yukarıdaki adımlardan sonra elde edilen kovaryans matrix, n x n boyutlarında olmaktadır. Bir sonraki adımda ise, kovaryans matrisin özdeğer (λ) ve özvektör (V) matrisleri hesaplanır:

$$V \cdot K = \lambda \cdot K \quad (4.3)$$

λ sütun matrisi, K matrisinin özdeğerlerini içerirken, V matrisi de her sütunda bir özvektör olmak üzere özvektörleri ve dolayısıyla yeni koordinatları içermektedir.

$$V = \begin{bmatrix} a_{11} & a_{21} & a_{31} & \dots & a_{n1} \\ a_{12} & a_{22} & a_{32} & \dots & a_{n2} \\ a_{13} & a_{23} & a_{33} & \dots & a_{n3} \\ \dots & \dots & \dots & \dots & \dots \\ a_{1n} & a_{2n} & a_{3n} & \dots & a_{nn} \end{bmatrix}$$

Şekil 4.4 n x n kovaryans matris için, n uzunlukta n adet özvektör

Toplamı belli bir eşik değerini sağlayan özdeğerlere karşılık gelen özvektörler seçilerek toplamda n tane olan koordinat yerine $z < n$ olmak üzere daha az sayıda koordinat kullanılabilir. Burada seçimi özdeğeri en büyük özvektörden başlayarak büyüklük sırasına göre seçim yapmanın, varyansı koruma adına gerekli bir hareket olduğu aşikardır:

$$\sum_{i=1}^n \lambda_i = \lambda_T \quad (4.4)$$

$$\sum_{i=1}^z \lambda_i = \lambda_Z \rightarrow \max\left(\frac{\lambda_Z}{\lambda_T}\right) \quad (4.5)$$

$$V' = \begin{bmatrix} a_{11} & a_{21} & a_{31} & \dots & a_{z1} \\ a_{12} & a_{22} & a_{32} & \dots & a_{z2} \\ a_{13} & a_{23} & a_{33} & \dots & a_{z3} \\ \dots & \dots & \dots & \dots & \dots \\ a_{1n} & a_{2n} & a_{3n} & \dots & a_{zn} \end{bmatrix}$$

Şekil 4.5 İlk z adet özdeğere karşılık gelen, n uzunlukta z adet özvektör

Elde edilen z adet özvektör, yeni koordinat sistemine bir dönüşüm matrisi olarak düşünülürse, bunun veri setine uygulanması halinde, veri seti istenen koordinat sistemine dönüştürülmüş olmaktadır. Ancak tüm özvektörler yerine sadece belli kısmının alınması, dönüşüm esnasında bir miktar veri kaybına yol açsa da, amaç veri kaybını mümkün mertebe az tutarak, dönüşüm sonrası koordinat/boyut sayısında olabildiğince azaltma sağlamaya çalışmaktır:

$$Z = X \cdot V' \quad (4.6)$$

Bu dönüşüm sonrasında, m satır (eleman) ve n sütundan (boyut/özellik) oluşan veri, m satır (eleman) ve z sütundan (boyut/özellik) oluşur hale gelmektedir. Boyut sayısı n-z kadar düşmüş olmasına karşın, seçilen özvektörlere karşılık gelen özdeğerlerin, tüm özdeğerlere olan toplamı oranında bilgi de korunmuş olmaktadır.

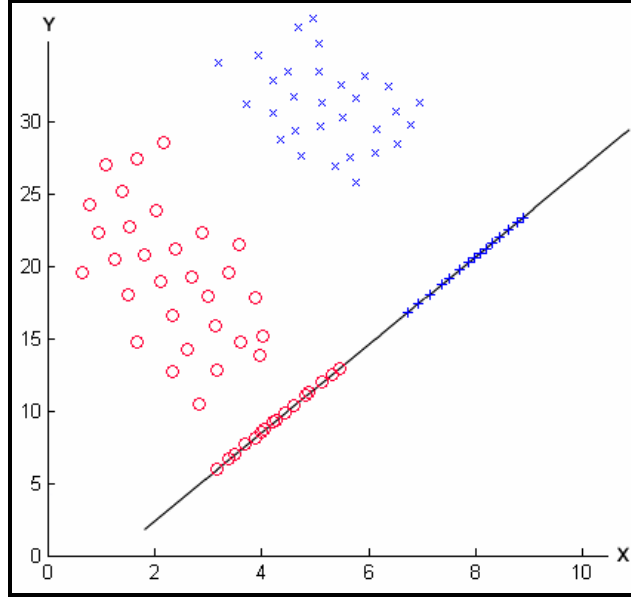
4.2 Doğrusal Diskriminant Analizi

Veri setindeki her elemanı birbirlerinden bağımsız inceleyen TBA'nin aksine, DDA verideki aynı sınıfa ait elemanları bir bütün olarak inceleyerek, bir sınıfa ait birden fazla elemanın bilgisini kullanarak ve elemanların sınıf içi ve sınıfların kendi aralarındaki saçılmalarının bilgisini çıkartarak sınıflar arası bir ayrıma gitmeyi amaçlar. Böylece hesaplanan yeni koordinat sisteminde sınıflar arasındaki bilgi değişimi de korunmaya çalışılır ve sınıflandırma algoritmalarında kullanılırken daha iyi sonuç verebilen koordinatlar elde edilebilir (Belhumeur vd., 1997).

Aynı sınıftaki elemanların birbirlerine göre olan konumlarına Sınıf İçi Saçılma (S_i), farklı sınıfların birbirlerine göre olan konumlarına Sınıflar Arası Saçılma (S_A) denirse, DDA:

$$\max \left(\frac{W^T \cdot S_A \cdot W}{W^T \cdot S_i \cdot W} \right) \quad (4.7)$$

koşulunu sağlayan W matrisini bulmayı hedefler.



Şekil 4.6 Örnek veri seti için hesaplanan yeni koordinat

Örneğin Şekil 4.6 incelenirse bulunan yeni tek bir koordinat üzerinde her iki sınıfın rahatlıkla ayrılabilirdiği görülebilmektedir, ancak sadece X ve Y koordinatları göz önüne alınırsa bunlar üzerinde iki sınıftaki elemanların değerlerinin çakıştığı görülebilmektedir.

DDA'nde aynı TBA'nde olduğu gibi veri seti belli bir satır formuna getirilir, ancak bu sefer, veri setindeki hangi satırın hangi sınıfa ait olduğu bilgisi de kullanılmaktadır. Örneğin her bir sınıfa ait r örneğin olduğu s sınıflı ve n boyutlu bir veri düşünülürse aşağıdaki yapıda bir matris elde edilmektedir:

$$X = \begin{bmatrix} A_{11} & A_{12} & A_{13} & \dots & A_{1n} \\ A_{21} & A_{22} & A_{23} & \dots & A_{2n} \\ \dots & \dots & \dots & \dots & \dots \\ A_{r1} & A_{r2} & A_{r3} & \dots & A_{rn} \\ B_{11} & B_{12} & B_{13} & \dots & B_{1n} \\ \dots & \dots & \dots & \dots & \dots \\ S_{r1} & S_{r2} & S_{r3} & \dots & S_{rn} \end{bmatrix}$$

Şekil 4.7 r örnekli n boyutlu s sınıf için örnek matris

Matris formuna getirilen veri setinde, aynı TBA'nde olduğu gibi her özellik için tüm veri setinde bir ortalama değer hesaplanabilir. Ayrıca, ayrı ayrı her bir sınıf için de özelliklerin ortalaması hesaplanmaktadır. Şekil 4.6'daki matris için bu hesap incelenirse:

$$\overline{X}_i = \frac{\sum_{j=a}^s \sum_{k=1}^r j_{ki}}{S \cdot R} \quad (4.8)$$

$$\overline{A}_i = \frac{\sum_{j=1}^r A_{ji}}{R}, \overline{B}_i = \frac{\sum_{j=1}^r B_{ji}}{R} \dots \overline{S}_i = \frac{\sum_{j=1}^r S_{ji}}{R} \quad (4.9)$$

şeklinde olduğu görülür. Bu iki bilgi, öncelikle veri setinin genelini sıfıra ortalananmış hale getirmek için kullanılır:

$$X_A' = \begin{bmatrix} A_{11} - \overline{X}_1 & A_{12} - \overline{X}_2 & A_{13} - \overline{X}_3 & \dots & A_{1n} - \overline{X}_n \\ A_{21} - \overline{X}_1 & A_{22} - \overline{X}_2 & A_{23} - \overline{X}_3 & \dots & A_{2n} - \overline{X}_n \\ \dots & \dots & \dots & \dots & \dots \\ A_{r1} - \overline{X}_1 & A_{r2} - \overline{X}_2 & A_{r3} - \overline{X}_3 & \dots & A_{rn} - \overline{X}_n \\ B_{11} - \overline{X}_1 & B_{12} - \overline{X}_2 & B_{13} - \overline{X}_3 & \dots & B_{1n} - \overline{X}_n \\ \dots & \dots & \dots & \dots & \dots \\ S_{r1} - \overline{X}_1 & S_{r2} - \overline{X}_2 & S_{r3} - \overline{X}_3 & \dots & S_{rn} - \overline{X}_n \end{bmatrix}$$

Şekil 4.8 Şekil 4.6'daki X matrisinin ortalananmış (X_A') hali

Daha sonra ise her bir sınıf, kendisi için hesaplanan ortalama özellikler kullanılarak kendi içinde sıfıra ortalananmış hale getirilir:

$$X_i' = \begin{bmatrix} A_{11} - \overline{A}_1 & A_{12} - \overline{A}_2 & A_{13} - \overline{A}_3 & \dots & A_{1n} - \overline{A}_n \\ A_{21} - \overline{A}_1 & A_{22} - \overline{A}_2 & A_{23} - \overline{A}_3 & \dots & A_{2n} - \overline{A}_n \\ \dots & \dots & \dots & \dots & \dots \\ A_{r1} - \overline{A}_1 & A_{r2} - \overline{A}_2 & A_{r3} - \overline{A}_3 & \dots & A_{2n} - \overline{A}_n \\ B_{11} - \overline{B}_1 & B_{12} - \overline{B}_2 & B_{13} - \overline{B}_3 & \dots & B_{1n} - \overline{B}_n \\ \dots & \dots & \dots & \dots & \dots \\ S_{r1} - \overline{S}_1 & S_{r2} - \overline{S}_2 & S_{r3} - \overline{S}_3 & \dots & S_{rn} - \overline{S}_n \end{bmatrix}$$

Şekil 4.9 Şekil 4.6'daki X matrisinin ortalananmış (X_i') hali

X_i' matrisindeki her bir örnek (sıra) kendi devriği ile çarpılarak elde edilen matrislerin toplamı ile Sınıf İçi Saçılma (S_i) matrisi hesaplanır. $[X_i']_j$ matristeki j. satıra işaret ediyorsa, Sınıf İçi Saçılma:

$$S_i = \sum_{j=1}^s [X_i']_j^T \cdot [X_i']_j \quad (4.10)$$

olmaktadır. Sınıflar arası saçılma ise, X_A' matrisinde her bir sınıfa ait tüm satırların oluşturduğu matrisin kendi devriği ile çarpılması sonucu elde edilen matrislerin toplamını bularak hesaplanır. X_A' matrisinde A sınıfına ait örnekler $[X_A']_A$ matrisini oluşturuyorsa, Sınıflar Arası Saçılma:

$$S_A = \sum_{j=A}^S [X_A']_A^T \cdot [X_A']_A + [X_A']_B^T \cdot [X_A']_B \dots + [X_A']_S^T \cdot [X_A']_S \quad (4.11)$$

olmaktadır. Daha önce de belirtildiği gibi, DDA

$$\max \left(\frac{W^T \cdot S_A \cdot W}{W^T \cdot S_I \cdot W} \right) \quad (4.7)$$

koşulunu sağlayan W matrisini bulmayı amaçlar. Bu koşulu sağlayan W matrisi:

$$S = S_I^{-1} \cdot S_A \quad (4.12)$$

matrisinin özvektörleridir:

$$V \cdot S = \lambda \cdot S \quad (4.13)$$

Hesaplanan özvektörler (V) aynı TBA'nde olduğu gibi, X matrisini yeni bir koordinat sisteminde ifade eden bir dönüşüm matrisi olarak kullanılır:

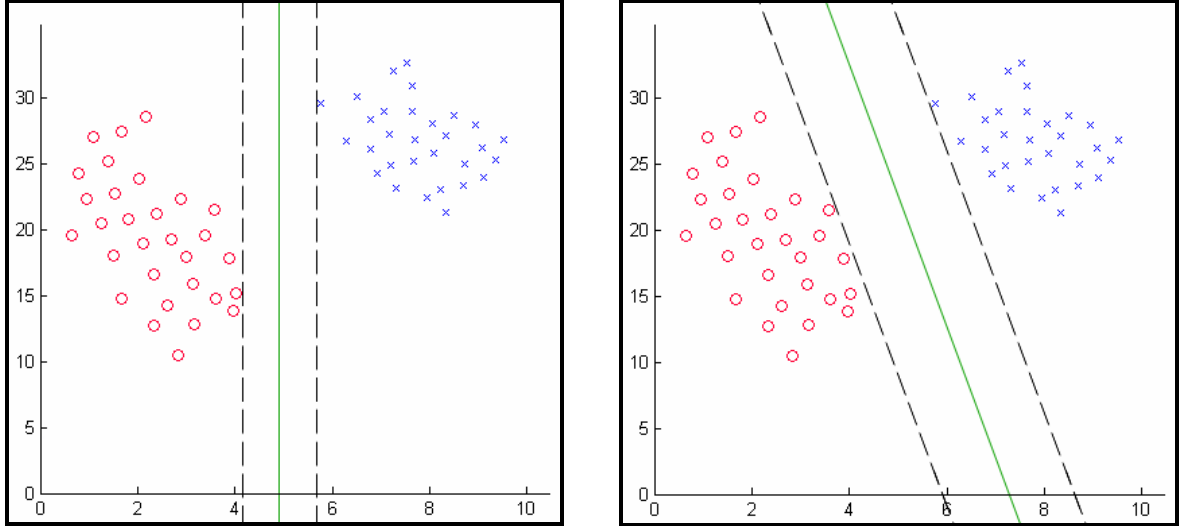
$$Z = X \cdot V \quad (4.14)$$

5. DESTEK VEKTÖR MAKİNELERİ

1960'lı yıllarda Vladimir Vapnik tarafından geliştirilen Destek Vektör Makineleri (DVM) (Support Vector Machines / SVM), Vapnik-Chervonenkis Öğrenme Teorisi'ne dayalı bir istatistiksel ve öğreticili öğrenme algoritmasıdır. İlk bulunduğu dönemde doğrusal bir ayırmaya giderek sınıflandırma yapan algoritma, Kernel yöntemlerin geliştirilmesi ile doğrusal-olmayan ayırmalarda da oldukça başarılı şekilde uygulanabilir hale gelmiştir.

5.1 Doğrusal Ayırma

Temelde iki sınıflı bir öğrenme sağlayan algoritma, sınıflara ait örnekler arasında en iyi şekilde ayırımı sağlayan bir hiper-düzlem bulmayı amaçlar. Örnek olarak doğrusal olarak ayrılabilen iki sınıflı bir veri seti göz önüne alınırsa:

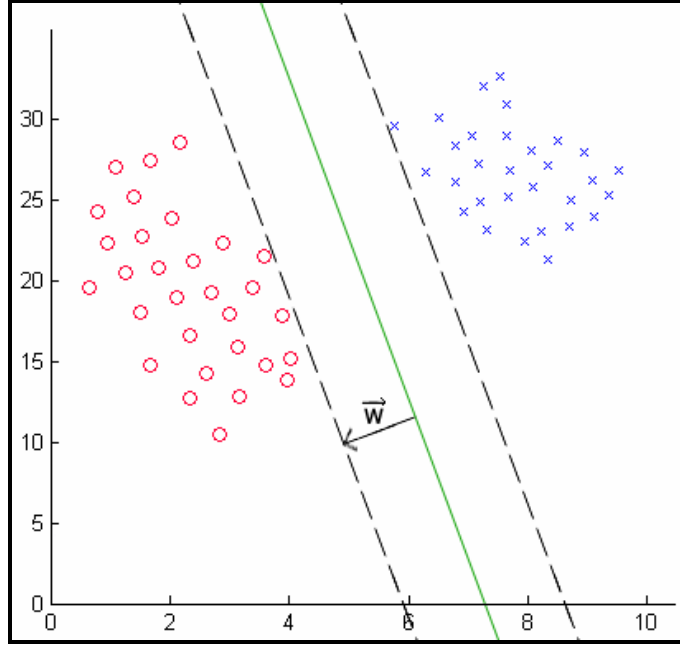


Şekil 5.1 Doğrusal olarak ayrılabilen veri setinde uygulanan iki farklı en uygun ayırıcı (yeşil çizgi) ve destek vektörlerinin bulunduğu ayırıcılar (kesikli çizgiler)

bulunan iki farklı –en uygun aday- ayırıcı düzlem (yeşil çizgiler) ve bu düzlemlere göre Destek Vektörleri (kesikli çizgiler) görülebilmektedir. Her iki ayırıcı da seti doğrusal olarak ikiye ayırır da, ikinci ayırıcının Destek Vektörleri arasındaki mesafeyi daha uzun tuttuğu, dolayısıyla daha iyi bir ayırma gerçekleştirebildiği görülebilmektedir.

En uygun ayırıcı doğruya dik ve en uygun ayırıcı ile destek vektörlerinden biri arasındaki mesafe büyüklüğünde bir vektöre w , ayırıcı doğrunun orijine olan uzaklığına ise $b(|w|)^{-1}$ denirse ayırıcının Hessian normal formu:

$$w \cdot x + b = 0 \quad (5.1)$$



Şekil 5.2 En uygun ayırıcı ve w vektörü

olmaktadır. Şekil 5.2'deki sınıflardan birine $+1$, diğerine ise -1 dendiğini varsayalım. $+1$ sınıfının destek vektörünün bulunduğu doğru:

$$w \cdot x + b = +1 \quad (5.2)$$

-1 sınıfının destek vektörünün bulunduğu doğru ise:

$$w \cdot x + b = -1 \quad (5.3)$$

koşullarını sağlamaktadır. Bu durumda, sınıfları belirleyen koşullar sırası ile:

$$w \cdot x + b \geq +1 \quad \text{ve} \quad w \cdot x + b \leq -1 \quad (5.4)$$

olmaktadır. Her bir x_i değeri için sınıfı belirten bir $y_i \in \{-1, 1\}$ değeri olduğundan bu iki koşul tek bir koşul olarak yazılabilir (Burges, 1998):

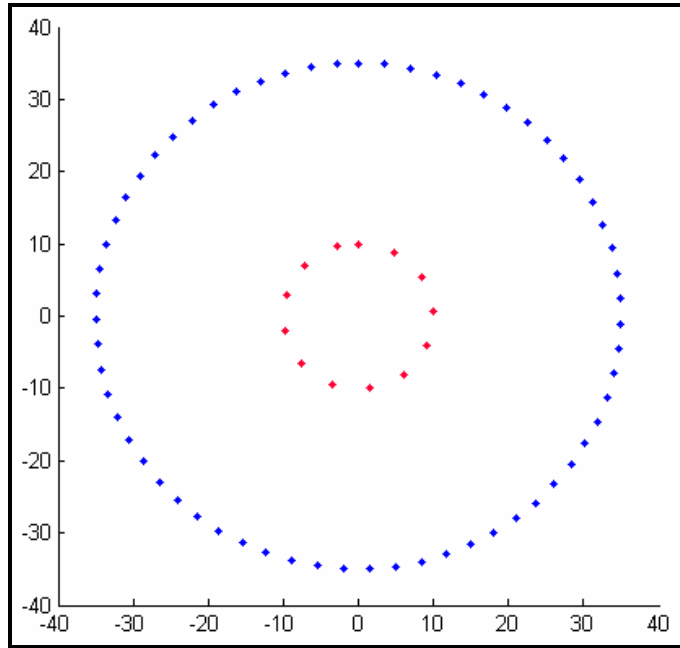
$$y_i \cdot (w \cdot x_i + b) \geq +1 \quad (5.5)$$

İki sınıfı birbirinden ayıran en uygun ayırıcı, iki sınıfa ait destek vektörlerinin bulunduğu ayırıcılara eşit uzaklıkta olup, bu uzaklığın $(|w|)^{-1}$ olmasından dolayı, destek vektörler arasındaki uzaklık $2 \cdot (|w|)^{-1}$ 'dir. Algoritma, Şekil 5.1'de görüldüğü gibi bu uzaklığı maksimize etmeyi amaçlamakta bir diğer ifade ile her i değeri için $y_i \cdot (w \cdot x_i + b) \geq +1$ koşulu altında $\min(2 \cdot (|w|)^{-1})$ değerini bulmayı hedeflemektedir (Vapnik, 1998; Burges, 1998). Aslen bir

kuadratik programlama problemi olan bu problem, modern uyarlamalarda Ardışık Minimal Optimizasyon yöntemleri ile çözülmektedir (Burges, 1998; Chang ve Lin 2001).

5.2 Doğrusal Olmayan Ayırma ve Kernel Yöntemi

DVM, doğrusal olarak ayrılabilen setlerde başarı ile uygulanabilir olmalarına rağmen, gerçek hayattaki gözlemlerin genelde doğrusal olmadığı bilinmektedir. Bu durumda, doğrusal olarak ayrılması mümkün olmayan veri setini, daha yüksek boyutlu bir uzaya taşıyarak, o uzayda gerçekleştirilen doğrusal bir ayırım, veri setinin kendi uzayına doğrusal-olmayan bir ayırım olarak izdüşürülebilmektedir (Schölkopf, 2000).

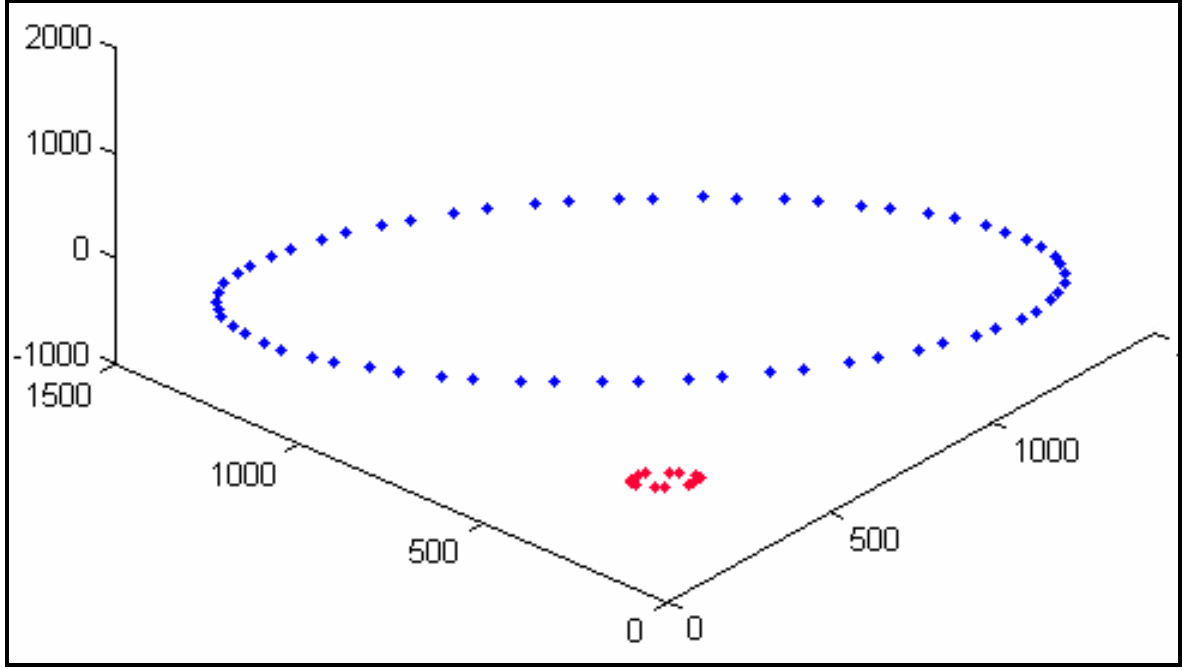


Şekil 5.3 Doğrusal ayrılamayan bir veri seti

Şekil 5.3'te görülen ve (x,y) koordinat sisteminde yer alan veri seti, x,y değerlerinden türetilen bir başka koordinat sistemine taşınabilir. Yeni koordinat sistemine dönüşü sağlayan fonksiyona ϕ denirse, $\phi(x,y)$ olarak ifade edilebilen bu koordinat sistemi, örnek olarak üç boyutlu olarak kabul edilir ve

$$\phi(x,y) = (x^2, \sqrt{2} \cdot x \cdot y, y^2) \quad (5.6)$$

olarak ifade edilirse, veri setinin bu yeni koordinat sistemindeki halini Şekil 5.4'te görmek mümkün olabilmektedir:



Şekil 5.4 Üçüncü koordinat, X^2-Y^2 düzlemine dik bir koordinat olmak üzere yeni koordinat sistemi

Bu yeni koordinat sisteminde ise veri setinin oldukça basit bir şekilde doğrusal olarak ayrılabilirdiği görülmektedir. Bu koordinat sisteminde kolaylıkla bulunan bu ayırım, orijinal koordinat sistemine geri izdüşürüldüğünde, doğrusal olmayan bir ayırım elde edilmektedir.

Verilen örnekte yüksek boyutlu uzaya taşınmak kolay olsa da, genelde incelenen örneklerin orijinal hallerinin bile çok boyutlu olması ve taşınmak istenen uzayların da karışık yapıda olması, noktaların tek tek yüksek boyuta taşınmasını imkansızla yakın kılmaktadır. Ancak literatürde Kernel Hilesi olarak anılan bir yöntem ile, noktaları tek tek taşımaya gerek kalmadan da istenen sonuç elde edilmektedir.

Doğrusal ayırımın yüksek boyutlu uzayda daha farklı sonuç verebilmesi, bu uzayda noktasal çarpımların daha farklı sonuç vermesiyle alakalıdır. Dolayısıyla, yüksek boyutlu uzaydaki noktasal çarpımların, orijinal uzaydaki karşılıkları ifade edilebilirse, orijinal verinin doğrusal olmayan ayırımını bulmak için veri setini yüksek boyutlu uzaya taşıyıp noktasal çarpım yapmaya gerek kalmadan, yüksek boyutlu uzaydaki noktasal çarpımın etkisi orijinal uzaydaki veriye uygulanabilmektedir.

Bu yöntemin kolay anlaşılır bir örneğini yine Şekil 5.3 ve 5.4'te görülen koordinat dönüşümünde gözlenebilir. Bunun için üç boyutlu a-b-c uzayında \vec{r} ve \vec{s} olarak iki vektör,

$$\vec{r} = (r_a, r_b, r_c) \text{ ve } \vec{s} = (s_a, s_b, s_c) \quad (5.7)$$

şeklinde tanımlansın. Bunların iki boyutlu x-y uzayının koordinatları cinsinden ifadeleri 5.6'da gösterilen ϕ fonksiyonu göz önüne alınırsa,

$$\vec{r} = (r_x^2, \sqrt{2} \cdot r_x \cdot r_y, r_y^2) \text{ ve } \vec{s} = (s_x^2, \sqrt{2} \cdot s_x \cdot s_y, s_y^2) \quad (5.8)$$

olmaktadır. İki vektörün üç boyutlu a-b-c uzayındaki noktasal çarpımları:

$$[a,b,c] \vec{r} \cdot \vec{s} = (r_a \cdot s_a) + (r_b \cdot s_b) + (r_c \cdot s_c) \quad (5.9)$$

x-y koordinatları cinsinden ifade edilirse:

$$[a,b,c] \vec{r} \cdot \vec{s} = (r_x^2 \cdot s_x^2) + (2 \cdot r_x \cdot r_y \cdot s_x \cdot s_y) + (r_y^2 \cdot s_y^2) \quad (5.10)$$

elde edilir. Bu ifadenin, iki vektörün iki boyutlu uzaydaki noktasal çarpımları olan:

$$[x,y] \vec{r} \cdot \vec{s} = (r_x \cdot s_x) + (r_y \cdot s_y) \quad (5.11)$$

ifadesinin tam karesi olduğu görülmektedir:

$$[(r_x \cdot s_x) + (r_y \cdot s_y)]^2 = (r_x^2 \cdot s_x^2) + (2 \cdot r_x \cdot r_y \cdot s_x \cdot s_y) + (r_y^2 \cdot s_y^2) \quad (5.12)$$

$$[a,b,c] \vec{r} \cdot \vec{s} = ([x,y] \vec{r} \cdot \vec{s})^2 \quad (5.13)$$

Sonuç olarak, uygulamada kullanılmakta olan yüksek boyutlu uzaydaki noktasal çarpım işlemlerinin sonuçları, ϕ dönüşümünü kullanmaya gerek kalmadan, düşük boyutlu uzaydaki noktasal çarpım cinsinden ifade edilebilir olmuştur (Burges, 1998; Schölkopf, 2000). Burada 5.13'teki ilişkiyi kuran fonksiyona (bu örnek için basit bir tam kare fonksiyonu) Kernel Fonksiyonu adı verilmektedir.

Tahmin edilebileceği gibi sonsuz sayıda farklı ϕ dönüşümü ve buna bağlı sonsuz sayıda kernel fonksiyonu tanımlanabilmektedir. Ancak doğal olarak bir kernel fonksiyonunun pratik anlamda kullanılabilir olması, pratik uygulamalarda işe yarar sonuçlar vermesi ile alakalıdır. Uygulamalarda en yaygın kullanılan ve doğrusal olmayan ayırım sağlayan kernel fonksiyonları aşağıdaki gibidir (Vapnik, 1998):

- Polinomal Kernel: $\text{Ker}(x_i, x_j) = (\gamma x_i^T \cdot x_j + r)^d$
- Radyal Temelli (Radial Basis Function) Kernel: $\text{Ker}(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2)$
- Sigmoid Kernel: $\text{Ker}(x_i, x_j) = \tanh(\gamma x_i^T \cdot x_j + r)$

Burada γ , d ve r ilgili kernelin parametrelerini ifade etmektedir.

DVM, kernel yöntemlerin de kullanım alanı bulması ile doğrusal olmayan ayrımlarda da başarılı sonuçlar elde edilmesini sağlamış ve oldukça üstün bir sınıflandırma algoritması olarak pratik uygulamalarda önemli bir yer edinmiştir.

5.3 Çok Sınıflı Destek Vektör Makineleri

DVM çalışma prensipleri gereği iki sınıflı bir ayırım gerçekleştirildiklerinden, daha fazla sayıda sınıf içeren gerçek hayat problemlerinin sınıflandırılmasında bir takım ek çalışmalara ihtiyaç duyulmaktadır.

Aslen ikili sınıflandırma yapan DVM, ardışık veya paralel çalışır şekilde bir çok DVM'nden oluşan bir sistem kurularak çoklu sınıflandırma yapabilir hale getirilebilir (Abe, 2003).

Genelde bu şekilde kullanılan üç tane yöntem söz konusudur:

- **Bire Karşı Bir:** Bire Karşı Bir Yöntemi'nde, her sınıf ile kalan her bir sınıf arasında ikili çiftler oluşturulur ve her bir çift için bir sınıflandırıcı eğitilir. Sonuçta n sınıf için $n \cdot (n-1) / 2$ adet sınıflandırıcı eğitilir. Bilinmeyen bir örneğin sınıflandırılması aşamasında ise tüm sınıflandırıcılardan elde edilen sonuçlar kıyaslanır ve en uygun olan sonuç seçilir.
- **Bire Karşı Hepsi:** Bire Karşı Hepsi Yöntemi'nde, her sınıf ile kalan tüm sınıfların oluşturduğu bir üst-sınıf arasında bir eğitim gerçekleştirilir. Dolayısıyla sistem n sınıf için n adet sınıflandırıcı oluşturur ve örneğin A sınıfı için oluşturulan sınıflandırıcı bir örneğin A sınıfına mı yoksa diğer tüm sınıfların oluşturduğu üst-sınıfa mı ait olduğunu, yani A sınıfına ait olup olmadığını hesaplar. Sınıflandırma aşamasında yine n adet sınıflandırıcının tamamından sonuç elde edilir ve en uygun olanı seçilir.
- **Döngüsel Olmayan Yönlü Graf:** Bu yöntem ise Bire Karşı Bir Yöntemi'nin sınıflandırma aşamasındaki sürecinin iyileştirilmiş bir haline dayanır. Sınıflandırma aşamasının ilk safhalarında örneğin ait olmayacağına karar verilen sınıfları içeren sınıflandırıcılar ileriki safhalarda çalıştırılmaz, böylece $n \cdot (n-1) / 2$ sınıflandırıcının tamamı sınanmadığından işlem yoğunluğu azalmaktadır.

5.4 Videoları Sınıflandırmak için Destek Vektör Makineleri'nin Kullanımı

1. Bölümde anlatıldığı gibi videoların sınıflandırılmasında her bir kare ayrı ayrı incelenebilir veya kareler arasında zaman-konum ilişkisi kurularak kareler bir bütün halinde incelenebilir. Bu çalışma da kareleri ayrı ayrı inceliyor olmasına rağmen, videonun sınıflandırılması aşamasında videodaki tüm karelerin sınıf bilgisi kullanılmaktadır.

Video kareleri, sınıflandırıcı için eğitim bilgisi olarak kullanıldığında, kullanılan kernel ve seçilen parametre değerleri için eğitici çalışmakta ve o anki parametreler için eğitim setini baz alarak bir sınıflandırıcı eğitmektedir. Bu aşamada video bilgisi kullanılmamakta olup, her resim birbirlerinden bağımsız olarak eğitim için bir girdi niteliğinde olmaktadır.

Ancak aynı yapıdaki test videolarına ait verilerde, her videoya ait tüm kareler eğitilen sınıflandırıcı tarafından sınıflandırılmakta ve bir videodaki kareler arasında en çok bulunan sınıf o videonun sınıfı olarak kabul edilmektedir. Böylece arada yanlış olarak sınıflandırılan karelerin adetleri, doğru olarak sınıflandırılan karelerin adetlerinden az ise, tanıma işlemini olumsuz etkileyen bu karelerin etkisinin kaldırılması gerçekleştirilmektedir.

Bu aşamada bir videodaki kareler arasında birden fazla sınıf arasında eşitlik olduğu durumda ise, eşit sayıda bulunan sınıflar arasında ayrı bir sınıflandırıcı tekrar eğitilmekte ve videonun sınıfı aynı şekilde tekrar bulunmaya çalışılmaktadır. Eşitlik durumunun bozulmadığı durumda ise video artık tahmin edilememiş sayılmaktadır.

Bu aşamayı bir örnekle anlatmak gerekirse bir kişiye ait ve n adet temsili kareye sahip bir videonun $S = \{A, B, C, D, E\}$ sınıflarından biri olarak sınıflandırıldığı durum göz önüne alınabilir. Tüm sınıfları sınıflandırabilen bir DVM ile n adet karenin, her kare için tek bir sınıf olacak şekilde sınıflandırıldığı varsayılırsa, her bir sınıf için o sınıf olarak sınıflandırılmış farklı kare sayıları sırası ile A_n, B_n, C_n, D_n, E_n ve

$$A_n + B_n + C_n + D_n + E_n = n \quad (5.14)$$

olmaktadır. Her karenin tek tek sınıflandırılması sonucu elde edilen adetlerden tek bir tanesinin diğerlerinin tümünden büyük olduğu durumda, örneğin:

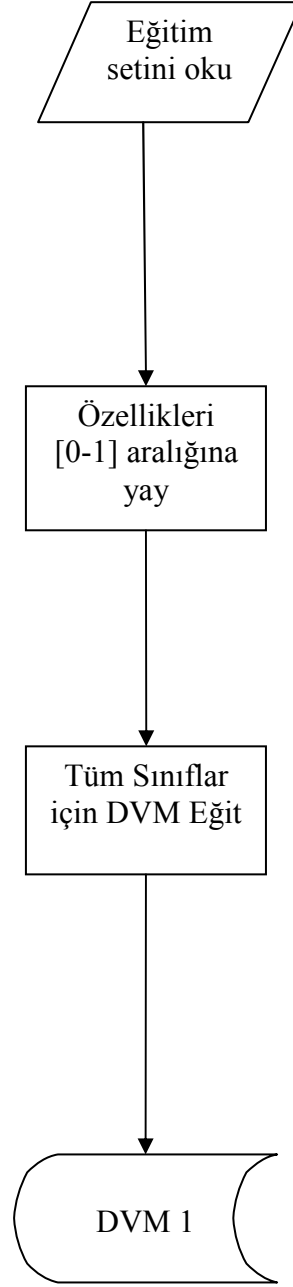
$$B_n > \forall \{ A_n, C_n, D_n, E_n \} \quad (5.15)$$

durumunda, video bu tek sınıf olarak -örnek için B sınıfı- sınıflandırılmaktadır. Ancak, birden fazla sınıfa ait kare sayılarında eşitlik var ise, örneğin:

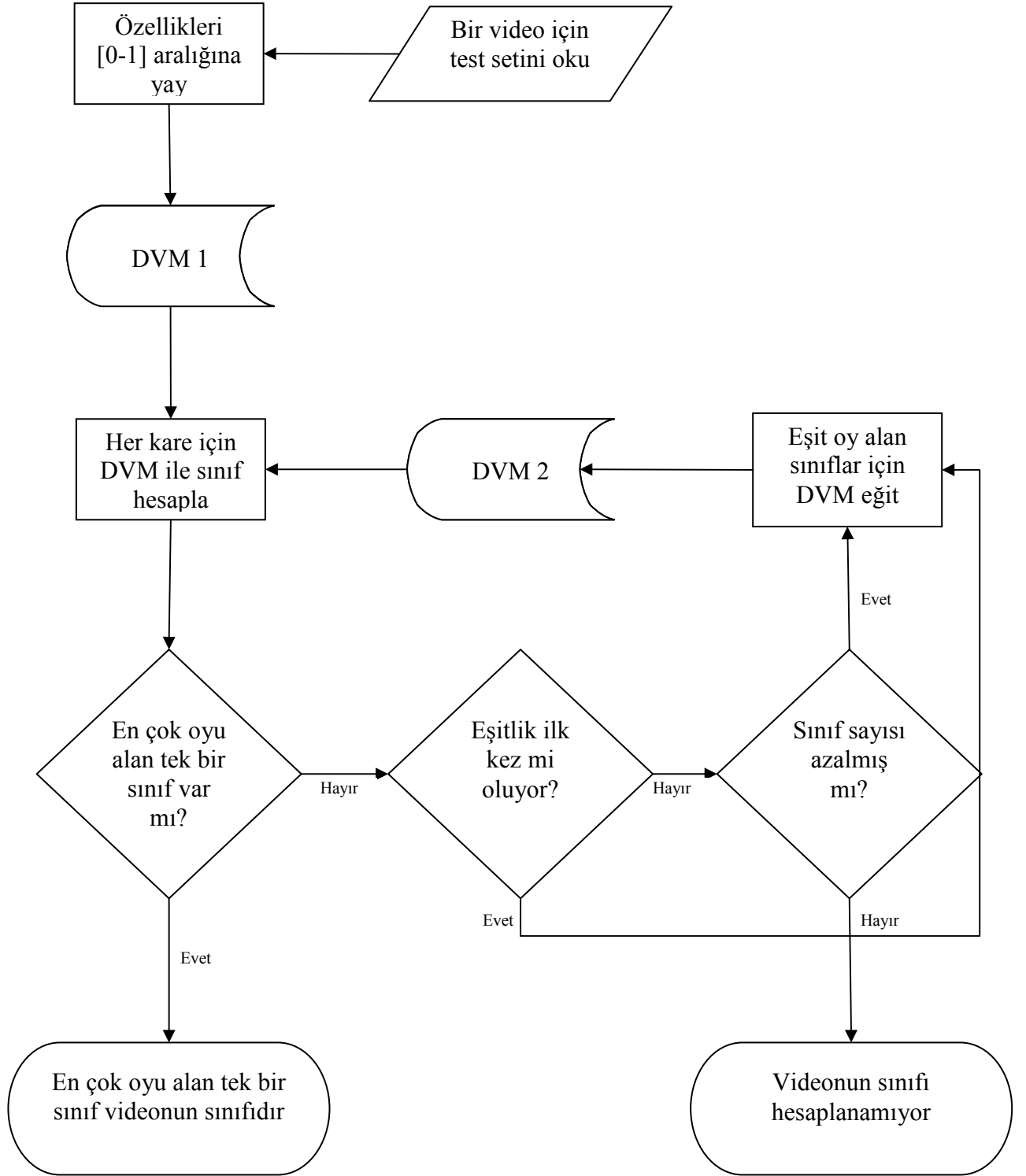
$$B_n = D_n > \forall \{ A_n, C_n, E_n \} \quad (5.16)$$

durumunda, kare sayıları eşit olan sınıflar için -örnek için B ve D sınıfları- yeni bir DVM eğitilmektedir. Daha önce de belirtildiği gibi bu döngü eşit kareye sahip sınıf sayısı azalmayana veya tek bir sınıf baskın çıkana kadar devam ettirilmektedir.

Bir parametre grubu için yapılan işlemler bir şema halinde gösterilirse:



Şekil 5.5 Eğitim Aşaması



Şekil 5.6 Test Aşaması

şemaları elde edilir.

6. UYGULAMA VE DENEY

6.1 Kullanılan Veri Seti

Tez çalışmasında, California Üniversitesi tarafından hazırlanan ve araştırmacılara ücretsiz olarak dağıtılan Honda/UCSD Video Veritabanı kullanılmıştır (Lee vd., 2003, 2005). 19 farklı kişiye ait, 640x480 çözünürlüğünde ve görece temiz bir arka plan önünde kaydedilen videolarda, kişilerin yüzlerinin farklı pozlardan çekilecek şekilde hem düzlemsel hem de üç boyutlu hareket etmeleri sağlanmıştır. Genelde 15'er saniye süren kayıtlarda kişinin farklı pozlarda ve büyüklüklerde bir çok yüz görüntüsü elde edilebilmiştir.



Şekil 6.1 Honda/UCSD Video Veritabanı'ndan Örnek Görüntüler

Deneyleerde iki ayrı set halinde dağıtılan veritabanının birinci seti kullanılmıştır. Daha önceden de belirtildiği gibi 19 farklı kişiye ait videolardan oluşan bu sette, her kişinin birer tane eğitim videosunun olmasına karşın, test video adedi kişi başına 1-4 arası adetlerde değişmektedir. Deneyleerde bu videoların tamamı, boyutları değiştirilmeden ve elle herhangi bir ön işlemden geçirilmeden kullanılmıştır.

6.2 Geliştirme Ortamı ve Kütüphaneler (C++ / OpenCV ve MATLAB / PRTOOLS)

Tez çalışmasında, videolardan yüz kısımları çıkartılırken kullanılan Intel'in OpenCV kütüphanesi, C++ ortamında kullanılmak üzere, Lienard ve Maydt'in geliştirdiği yöntemeye dayanan nesne bulma fonksiyonları içermektedir. Intel tarafından ücretsiz dağıtılan bu kütüphane, sunduğu bilgisayarla görü amacıyla kullanılabilir bir çok nesne ve fonksiyon ile, görüntü kaynaklarından (video, kamera, resim vs.) görüntülerin dijital ortama alınarak işlenmesini ve çeşitli sınıflandırıcılarla makine öğrenmesi sistemleri kurulabilmesini sağlamaktadır.

Kütüphane ile beraber gelen bir HBÖ eğitici programı ile, ilk olarak çok sayıda olumlu ve olumsuz örnek resimleri sisteme verilerek, öncelikle hangi HBÖ'in kullanılabilirliği, daha sonra bu özelliklerin nesne bulmada hangi ardışık kurallar ile kullanılacağı hesaplanmaktadır. Program, bulduğu ardışık kuralları Cascade isimli XML formatındaki dosyalarda saklayarak, daha sonra kütüphanenin kullanıldığı programlar içerisinde bu dosyalardaki HBÖ bilgisi ile nesne bulma fonksiyonlarının kullanılabilmesini sağlamaktadır.



Şekil 6.2 OpenCV ile bulunmuş bir yüz bölgesi

Bu fonksiyonlar ile, istenen bir resimdeki insan yüzü olarak tespit edilen kare şeklindeki bölgeler bulunarak ayrı olarak incelenebilmektedir. Ayrıca Intel, kütüphane ile beraber örnek XML dosyaları dağıtmakta, bu örnek dosyalar ile çalışmada kullanılan yüz bulma işlemleri gerçekleştirilebilmektedir.

Yapılan çalışmanın ilk etabında, OpenCV ile dağıtılan örnek XML dosyalarından yüz nesnesine ait olanlardan iki tanesi (frontalface_alt2 ve profileface) kullanılarak, video

karelerinin içerisindeki önden ve profilden görünen yüz kısımlarının çıkartılması amaçlanmıştır. Ayrıca gözler ve ağızın bulunması için de Modesto Castrillón Santana tarafından hazırlanmış olan XML dosyaları (Santana, 2007) kullanılmıştır.

TBA ve DDA içinse MATLAB programlama ortamı ve aynı ortam için geliştirilmiş olan PRTOOLS isimli paket kullanılmıştır. Paket, bir metin dosyasından sınıf ve özellik bilgileri okunduktan sonra bu bilgiler ile veri setinin boyutu istenilen boyuta veya istenilen varyansı korumak üzere belirlenen boyuta indirgeyebilen TBA dönüşümü yapmayı sağlamaktadır. Ayrıca yine aynı paket ile, yine sınıf ve özellik bilgileri okunan bir veri setinde sınıf sayısından az olmak üzere istenen adet boyuta düşürebilmek üzere DDA dönüşümü yapılması mümkündür.

Yüz bölgesini çıkartma amacıyla kullanılan OpenCV kütüphanesi, sunduğu görüntü işleme fonksiyonlarının yanında, başlıca Makine Öğrenmesi algoritmaları için de sınıflandırıcılar ve fonksiyonlar sunmaktadır. Bunlardan birisi olan ve aslen ünlü LIBSVM (Chang ve Lin, 2001) kütüphanesine dayalı DVM sınıflandırıcısı, deneyin sınıflandırıcı eğitimi ve testi aşamasında kullanılmış ve bu fonksiyonlara dayalı bir eğitim ve test sistemi kurulmuştur.

Boyut dönüşümü yapılan kısımlar MATLAB , diğer tüm kısımlar ise Microsoft Visual C++ 2005 ortamında geliştirilmiştir. Deneylerin tamamı 704 MB hafızaya sahip bir AMD Sempron 2200 işlemcili bilgisayarda, Windows XP üzerinde gerçekleştirilmiştir.

6.3 Yüz Bulma, İyileştirme, Ön İşlemler ve Oluşturulan Deney Setleri

HBÖ yardımıyla, bir videodaki insan yüzüne ait olduğu tespit edilen resimler kareler halinde çıkartılarak, anlatılan yöntemlerle üç farklı deney seti olmak üzere kaydedilmiştir. Burada, deneylerde kullanılan veri setlerinde bir videoda sadece bir kişi bulunduğu için, sistem bulunan yüzlerden sadece birincisini kaydetmektedir ki bu da OpenCV'nin çalışma mantığına göre sistemde bulunan en büyük (yani tek kişiye ait) yüz olmaktadır.

Bu aşamada, sistemin tamamen otomatik çalışması hedeflenmiş, bir video sisteme verildiğinde içerisinde o videodaki karelerde insan yüzünün istenen şekilde geçtiği en fazla sayıda resim çıkartılmaya çalışılmıştır. Karşıt olarak da, insan yüzüne ait olmayan veya insan yüzüne ait olmasına rağmen istenildiği kadar bilgi içermeyen resim sayısının da en az sayıda –tercihen sıfır- olması doğal olarak hedeflenmektedir.

Sistem bir video dosyasını okumaya başladığında, öncelikle her kareyi tek tek inceleyerek, varsa karede yüz olarak tespit ettiği bir bölgeyi çıkartarak 50x50 boyutlarında bir kare olarak

kaydetmektedir. Çalışmanın devamında, bulunduğu haliyle hiç bir ön işlemten geçmeden kaydedilen bu yüzlerden oluşan deney seti *Normal Set* olarak anılmaktadır.

Önden görünen yüz olarak tespit edilen ve 3.2. bölümdeki yöntemle onaylanan yüz bölgesi, 3.3. bölümde anlatılan şekilde gözleri aynı hizada olmak üzere döndürüldükten sonra 50x50 boyutlarında bir kare olarak kaydedilmiştir. Çalışmanın devamında bu resimlerden oluşan deney seti *Hizalanmış Set* olarak anılmaktadır.

Son olarak ise 3.4. bölümdeki yöntemler uygulanarak, bir önceki adımda elde edilen settaki resimlerden uygun olanları ön işlemlerden geçirilerek elips bir maske ile çevrelenmiştir. Çalışmanın devamında bu resimlerden oluşan deney seti *Maskeli Set* olarak anılmaktadır.

Önceden de belirtildiği gibi C++ ortamında geliştirilen bu kısımdaki tüm işlemler ve seçimler dışarıdan müdahale edilmeden sistemin kendisi tarafından gerçekleştirilmiştir. Sonuçta her üç deney seti için de elde edilen yüz bölgesi adedi ve elde edilme süreleri aşağıdaki gibi olmuştur:

Çizelge 6.1 Eğitim veri setlerinin büyüklük ve oluşturulma süreleri

	Normal Set	Hizalanmış Set	Maskeli Set
Süre (dk)	30,03	69,46	77,02
Kare Adedi	7.692	3.190	3.090

Eğitim setinin hesaplanması sistemin oluşturulması için gerekli olduğundan, videolar sınıflandırılmadan önce sistemin kurulması aşamasında tüm eğitim videolarının mutlaka işlenmesi gerekmektedir. Ancak bu aşama bir kez geçilmesi gereken bir aşama olup, sınıflandırılan her video için tekrar etmemektedir. Test videoların işlenme hızı ise, tek bir test videosu için ortalama değerler ile ifade edilebilir:

Çizelge 6.2 Test veri setlerinin bir video için ortalama büyüklük ve oluşturulma süreleri

	Normal Set	Hizalanmış Set	Maskeli Set
Süre (dk)	1,02	2,56	3,12
Kare Adedi	272	131	127

Her üç deney seti de, sistem tarafından ayrı ayrı resim dosyaları olarak kaydedilmelerinin yanında, makine öğrenmesi yöntemlerinde kullanılmak üzere, her bir satırı bir resmi temsil

eden bir metin dosyasına da yazılmıştır. Metin dosyasının formatı incelenen videonun ait olduğu kişiye ait bir değeri (Sınıf Bilgisi), sadece incelenen videoya ait özgün bir değeri (Video ID) ve kaydedilen resimdeki nokta değerlerinin (50x50 bir resim için 2.500 nokta) satırlar yan yana olmak üzere hallerini içermektedir.

6.4 TBA ve DDA ile Boyut Düşürme

Bir önceki bölümde anlatılan yöntemlerle elde edilen, 50x50 boyutlarındaki yüz resimleri, 2.500 noktadan oluşan özellik ve sınıf/video bilgilerini de içermek üzere metin dosyalarına yazılmıştı. İkinci aşamada ise, 2.500 boyutlu bu veri, 4. Bölüm’de anlatılan yöntemler kullanılarak daha düşük boyutta temsil edilmeye çalışılmaktadır.

TBA dönüşümü esnasında 2.500 boyutlu veri setinin kaç boyuta indirgeneceği, uygulayıcının vermesi gereken bir karardır. Tez uygulamasında boyut sayısına karar verilmesi yerine, varyansın korunmasına karar verilmiş, PRTOOLS’un TBA hesaplayıcı fonksiyonuna parametre olarak varyansın %99’unun korunması gerektiği verilmiştir.

İlk aşamada bir önceki bölümde sözü edilen üç farklı veri setine de, birbirlerinden bağımsız olarak TBA uygulanmıştır. Burada, veri setinin dönüşüm matrisi hesaplanırken, sadece eğitim videolarından elde edilen veriler kullanılmış, test videolarından elde edilen verilere dönüşüm uygulanırken eğitim videolarından elde edilen matris kullanılmıştır. Her üç tipteki veri seti de hem resimlerin yapısı hem de adetleri açısından birbirlerinden farklı olduğundan, üçüne uygulanan TBA hesaplaması farklı sonuçlar vermiştir.

Bir sonraki aşamada ise, TBA ile daha düşük boyutlu bir uzaya indirgenen eğitim setine, DDA uygulanmış ve 19 sınıftan oluşan veri seti 18 boyutlu bir uzaya indirgenmiştir. Burada, yine TBA’nde olduğu gibi dönüşüm matrisleri eğitim videolarından elde edilen setlerden hesaplanmış, test videoları bu eğitim videolarından elde edilen dönüşüm matrisleri kullanılarak indirgenmiştir.

Her üç eğitim seti için ayrı ayrı uygulanan TBA ve DDA hesaplamalarının süre ve elde edilen temel bileşen adetleri açısından incelemeleri Çizelge 6.3’te görülebilir. Bir önceki adımda olduğu gibi, eğitim seti kullanılarak TBA ve DDA hesaplamalarının gerçekleştirilmesi sistemin oluşturulması için gerekli olduğundan, videolar sınıflandırılmadan önce sistemin kurulması aşamasında bu hesaplamaların mutlaka bir kere yapılması gerekmektedir. Sınıflandırma aşamasında ise sistem sadece bu analizler sonucu elde edilen dönüşüm matrislerini kullanarak her bir video için tek bir matris çarpımı gerçekleştirmektedir.

Çizelge 6.3 Eğitim setleri için TBA ve DDA hesaplamaları sonuçları

	Normal Set	Hızlanmış Set	Maskeli Set
TBA Hesaplama Süresi (sn)	401	270	246
Temel Bileşen Sayısı	548	359	494
DDA Hesaplama Süresi (sn)	4	2	3
Toplam Hesap Süresi (sn)	405	272	249

Sonuç olarak, başlangıçta 2.500 noktadan/boyuttan/özellikten oluşan veri seti, 18 özellekle tanımlanabilir hale getirilmiştir. Deneyin son aşamasında elde edilen 18 özellikli üç farklı (Normal, Hızlanmış ve Maskeli) veri setindeki eğitim videoları için DVM sınıflandırıcısı eğitilmiş, elde edilen sınıflandırıcıların test videoları üzerindeki tanıma oranları karşılaştırılmıştır.

6.5 DVM ile Sınıflandırma

Bir önceki aşamada elde edilen sonuçlar, bir DVM temelli sınıflandırıcı için girdi niteliği taşımaktaydı. Bu aşamada sözü edilen sınıflandırıcı eğitilmiş ve farklı tipteki veri setleri için sonuçlar irdelenmiştir.

Bu veri setindeki eğitim videolarına ait verilerden, 18 adet özellik ve sınıf bilgisini kullanarak bir DVM sınıflandırıcısı eğitilmektedir. LIBSVM'nin geliştiricileri, eğitim ve sınıflandırma aşamasında daha başarılı sonuçlar elde etmek için eğitim verisindeki sürekli özelliklerin [0-1] aralığına yayılmasını önermektedir (Chang ve Lin, 2001). Bununla beraber aynı şekilde test verisi de, eğitim verisi yayılırken kullanılan katsayılar kullanılarak ilgili aralığa yayılmalıdır. Buna göre uygulamadaki sistem de öncelikle 18 adet özelliği kendi arasında [0-1] aralığına yaymaktadır.

5. Bölüm'de de gösterildiği gibi popüler uygulamalarda farklı tipte kernel kullanımı mümkündür. Ayrıca her kernel için farklı sayıda ve aralıkta kullanılabilen parametreler de mevcut olduğundan, deney yapılırken parametre değerlerinin önceden kestirilmesi imkansızdır. Bu çalışmada, en çok kullanılan kernel tiplerinden birisi olan Radyal Temelli (Radial Basis Function) Kernel kullanılmıştır. Radyal Temelli Kernel, parametre olarak sadece γ parametresini beklemektedir, ancak bir de DVM'nde en iyi ayırıcı düzlem bulunurken denklem çözümede kullanılan C parametresi söz konusu olduğundan, bu iki parametre için en uygun kombinasyonun bulunması ve sistemin bu iki parametrenin en iyi sonucu verdiği durumda sınanması gerekmektedir.

Bahsedilen bu en uygun durumun bulunmasına dair bir takım çalışmalar olsa da, bunlar bu tez çalışmasının konusu dışında olduğundan, bu tezde en uygun parametre kombinasyonu iki parametre için de belirli aralıklarda ve belirli adımlarda sistemin tekrar tekrar çalıştırılması ile bir ızgara tarama (grid search) işlemi sonucu bulunmuştur.

Sistemin bu aşamasında Bölüm 5.4'teki işlemler her bir parametre çifti için gerçekleştiriliyor olup, bahsedildiği gibi sistem;

$$\gamma : 0,001 \leq \gamma \leq 100 \text{ ve } C : 1 \leq C \leq 100 \quad (6.1)$$

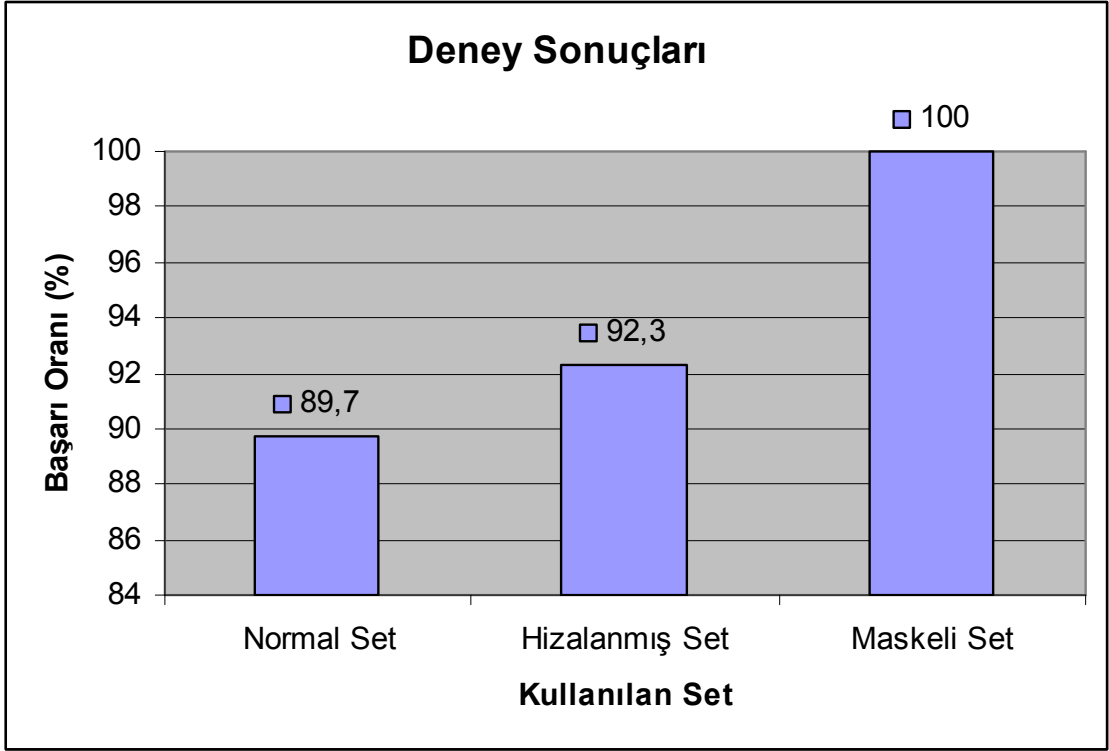
parametreleri için bu işlemleri tekrar etmektedir. Sonuçta en uygun parametre çifti ile elde edilen en yüksek sonuç, sistemin genel başarı sonucu olarak kabul edilmiştir.

Sınıflandırıcının performansının eğitim süresi olarak ölçümü de bahsedilen toplam ızgara arama süresi olarak kabul edilmiş, bu süre sonunda elde edilen en uygun parametrelere sahip destek vektörleri, sınıflandırma aşamasında kullanılmıştır. Önceki adımlarda da olduğu gibi, bu adımda da eğitim aşaması sistemin kurulması esnasında bir kereliğine gerçekleşmekte olup, sınıflandırma işlemi sadece elde edilen destek vektörlerinin uygulanması ile sağlanmaktadır.

6.3. Bölüm'de sözü edilen üç farklı veri seti, 6.4. Bölüm'de birbirlerinden bağımsız olarak DDA ile indirgenmiş ve sonuçta üçer tane eğitim ve test seti elde edilmişti. Bu aşamada, yukarıda anlatılan şekilde her üç eğitim seti için ayrı ayrı sınıflandırıcılar hesaplanmış ve her üç sınıflandırıcı da ilgili test seti ile sınanmıştır. Sonuçta görsel olarak farklılaştırılan eğitim setlerinin, sınıflandırıcı başarısını da önemli ölçüde değiştirdiği görülmüştür:

Çizelge 6.4 Sınıflandırıcı süreleri ve başarı oranları

	Normal Set	Hızalanmış Set	Maskeli Set
Toplam Izgara Arama Süresi (sn)	706	143	125
Toplam Test Video Adedi	39	39	39
Doğru Sınıflandırılan Video	35	36	39
Başarı Oranı (%)	% 89,7	% 92,3	% 100
En İyi Oranı Veren C	2	1	1
En İyi Oranı Veren γ	8,192	4,096	0,256



Şekil 6.3 Başarı Oranları

Tüm adımlardaki sürelerin toplam olarak ifade edilmesi ise sistemin genel olarak çalışma performansı hakkında bir bilgi verebilir. Yüz bulma ve ilgili ön işlemleri, boyut düşürme ve sınıflandırma adımlarının eğitim süreleri toplanırsa, sistemin genel olarak eğitiminin ne kadar süreceği görülebilir. Bir kez gerçekleştirilen bu adımın ardından ise, bir tane videonun sınıflandırılması da süre olarak o videodaki yüzlerin bulunması ve ön işlemlerden geçirilmesi kadar sürmekte, boyut düşürme ve ardından bu düşük boyutlu sette destek vektörlerin uygulanması ise basit birer matris ve vektör çarpımından ibaret olduğundan süreyi etkilememektedir.

Çizelge 6.5 Sistemin toplam eğitim ve bir video için sınıflandırma süreleri (dakika)

	Normal Set	Hizalanmış Set	Maskeli Set
Yüz Bulma ve Ön İşlemler	30,03	69,46	77,02
Boyut Düşürme	6,75	4,53	4,15
Sınıflandırıcı Eğitimi	11,77	2,38	2,08
Toplam Eğitim Süresi	48,55	76,37	83,25
Bir Video Sınıflandırılması	1,02	2,56	3,12

7. SONUÇ VE YORUMLAR

Yüz tanımanın bir alt dalı olan hareketli görüntülerden yüz tanıma alanına ait videolardan yüz tanıma, kendi kendine başarıyla çalışan bir sistem kurulabildiğinde gerçek hayatta geniş bir pratik kullanım alanına sahip olabilecek bir araştırma konusudur.

Girişte de anlatıldığı gibi bu çalışma, *videolardaki karelerin tümünü kullanmak yerine kareler arasında eleme ve yüz bileşenlerinde hizalama/normalizasyon işlemlerinin* videodan yüz tanıma başarısına olan etkisini incelemektedir. Bunun için 6. Bölüm'de bahsedildiği gibi bir veri seti üzerinde çalışma gerçekleştirilmiş ve üç aşamalı bir sistem kurulmuştur.

Sistem ilk aşamada videolardaki kareleri tek tek inceleyerek yüz olarak tespit ettiği kısımları çıkartmış, ikinci aşamada bu yüzleri bir veri seti haline getirip bu veri setinin boyut sayısını düşürmüş, son olarak da tanıma aşamasında aynı videolardan elde edilen yüzler arasındaki ilişkiyi göz önüne alan bir sınıflandırıcı eğitmiştir.

Çalışmanın cevap aradığı soruya yönelik farklılaştırma, bu aşamalardan birincisinde gerçekleştirilmiş ve

1. Videolarda yüz olarak bulunan tüm bölgelerden oluşan
2. Sadece 3.2. Bölüm'de anlatılan şartlara uyan yüzlerin gözlerinin hizalanmış hallerinden oluşan,
3. Bu yüzlerin gözleri arasındaki mesafeyi kullanarak bir elips maske ile çevrelenmiş ve histogram eşitlenmiş hallerinden oluşan

üç farklı veri setinin sistem tarafından otomatik olarak oluşturması sağlanmıştır.

İkinci aşamada tüm veri setlerine de aynı şekilde TBA ve DDA uygulanmış, son aşamada da her biri ile ayrı ayrı DVM sınıflandırıcıları eğitilmiş ve her sınıflandırıcı kendi test seti üzerinde tanıma gerçekleştirmiştir.

Sonuçta görüldüğü gibi sadece eleme sonucu oluşturulan gözleri hizalanmış set, büyüklük olarak normal setin yarısı kadar olsa da, başarıda az miktarda artış sağlamıştır. Asıl artış ise eleme üzerine normalizasyon da yapıldığı zaman söz konusu olmuştur. Ancak normalizasyon için de yüz üzerinde öncelikle gözlerin ve ağız konumunun doğru şekilde bulunması gerektiğinden bu iki adımın birbirinden bağımsız veya birbirlerinin alternatifi gibi düşünülmesi yanlış olacaktır.

Veri setlerinin oluşturulma süreleri ve nicel büyüklükleri karşılaştırıldığında ise, ilk aşamada normal setin oluşturulması diğer iki sete göre üç kata yakın daha hızlı gerçekleştirilmektedir. Ek işlemlerin tek dezavantajı bu en baştaki setlerin oluşturulma süresi olarak görülebilir.

Ancak aynı çizelgede görüldüğü gibi sonuçta elde edilen set büyüklükleri ise, hizalı ve normalize edilmiş setlerde yarı yarıya düşmüştür. Bunun etkisi ise ikinci ve üçüncü aşamalardaki boyut düşürme ve sınıflandırıcı aşamalarında süre kazancı olarak görülmüştür.

Sonuçta en başta amaçlandığı ve iddia edildiği gibi, videolardan yüz tanımada tüm karelerin kullanılması yerine uygun bir seçim yapıldığında daha az –yarı yarıya yakın- sayıda kare ile daha yüksek başarı oranı elde etmenin mümkün olduğu görülebilmektedir. Burada önemli olan kareler arasında seçim yaparken kullanılan eleme mantığını iyi seçmek olduğundan, daha hızlı hesaplanabilen eleme yöntemleri (örneğin görüntü tanıma yöntemleri yerine veri seti üzerinde çalışan kümeleme algoritmaları gibi) ile elde edilen sonuçları incelemek gelecek çalışmalar olarak düşünülebilir. Ayrıca kullanılan videolarda bir videoda bir kişi olduğu için ek bir nesne takibi vs. işlemi gerçekleştirilmemiştir, ancak daha gelişmiş bir sistem kurulmak istenirse bu tip bir özellik eklenerek de birden fazla kişiye ait videolar da sistem tarafından işlenebilir hale getirilebilir.

KAYNAKLAR

- Abe, S., (2003) "Analysis of Multiclass Support Vector Machines", Proc. International Conference on Computational Intelligence for Modelling Control and Automation (385-396)
- Belhumeur, P. N., Hespanha, J. P., Kriegman, D. J. (1997) "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection", Tran. IEEE Pattern Analysis and Machine Intelligence (Vol.19 Iss.7 711 – 720)
- Burges, C. (1998) "A Tutorial on Support Vector Machines for Pattern Recognition", Data Mining and Knowledge Discovery 2 (121 - 167)
- Chang, C-C., Lin, C-J. (2001) "A Library for Support Vector Machines", LIBSVM Web
- Chang, C-C., Lin, C-J. (2001) "A Practical Guide to Support Vector Classification", LIBSVM Web
- Hadid, A., Pietikainen, P., (2004) "From Still Image to Video-Based Face Recognition An Experimental Analysis", Proc. Sixth IEEE Int. Conf. on Automatic Face and Gesture Recognition (813 - 818)
- Lee, K. C., Ho, J., Yang, M. H., Kriegman, D. (2005) "Visual Tracking and Recognition Using Probabilistic Appearance Manifolds", Computer Vision and Image Understanding
- Lee, K. C., Ho, J., Yang, M. H., Kriegman, D. (2003) "Video-Based Face Recognition Using Probabilistic Appearance Manifolds", Proc. IEEE Computer Society Conf. on Comp. Vision and Patt. Rec. (Vol.1 I-313- I-320)
- Lienhart, R., Maydt, J., (2002) "An Extended Set of Haar-like Features for Rapid Object Detection", Proc. IEEE Int. Conf. on Image Processing (Vol.1 I-900 - I-903)
- Ognjen, A., Andrew, Z. (2005) "Automatic Face Recognition for Film Character Retrieval in Feature-Length Films", Proc. IEEE Computer Society Conf. on Comp. Vision and Patt. Rec. (Vol.1 I-860 - I-867)
- Santana, M. C., Suárez, O. D., Tejera, M. H., Artal, C. G. (2007) "ENCARA2: Real-time Det. of Multiple Faces at Different Resolutions in Video Streams", Journal of Visual Communication and Image Representation
- Schölkopf, B. (2000) "Statistical Learning and Kernel Methods", Microsoft Research
- Smith, L. I. (2002) "A Tutorial on Principal Components Analysis", Cornell University Web
- Turk, M., Pentland, A. (1991) "Face Recognition Using Eigenfaces", Proc. IEEE Computer Society Conf. on Comp. Vision and Patt. Rec. (586 - 591)
- Vapnik, V. N. (1998) "Statistical Learning Theory", John Wiley & Sons Inc., New York
- Viola, P., Jones, M., (2001) "Rapid Object Detection Using A Boosted Cascade Of Simple Features", Proc. IEEE Computer Society Conf. on Comp. Vision and Patt. Rec. (Vol.1 I-511 - I-518)

ÖZGEÇMİŞ

Doğum tarihi 15.04.1981

Doğum yeri İstanbul

Lise 1991-1998 Kartal Burak Bora Anadolu Lisesi

Lisans 1998-2004 Yıldız Teknik Üniversitesi Kimya-Metalurji Fak.
Matematik Mühendisliği Bölümü

Yüksek Lisans 2005-2008 Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü
Matematik Mühendisliği Anabilim Dalı

Çalıştığı kurumlar

2002-2004 Enersis Enerji Sis. Müh. Hizm. San. Tic. A.Ş.
2004-2006 TelcoTurk
2006-Devam Promesa Bilgi Teknolojileri