

Görüntü Tanıma Uygulamalarında Tensörel Ortak Bileşen Yöntemleri

Hasan Serhan Yavuz

DOKTORA TEZİ

Elektrik-Elektronik Mühendisliği Anabilim Dalı

Eylül 2008

Tensor Based Common Component Methods in Image Recognition Applications

Hasan Serhan Yavuz

DOCTORAL DISSERTATION

Department of Electrical and Electronics Engineering

September 2008

Görüntü Tanıma Uygulamalarında Tensörel Ortak Bileşen Yöntemleri

Hasan Serhan Yavuz

Eskişehir Osmangazi Üniversitesi
Fen Bilimleri Enstitüsü
Lisansüstü Yönetmeliği Uyarınca
Elektrik-Elektronik Mühendisliği Anabilim Dalı
Telekomünikasyon Bilim Dalında
DOKTORA TEZİ
Olarak Hazırlanmıştır

Danışman: Yrd. Doç. Dr. M. Atif Çay

Eylül 2008

ONAY

Elektrik-Elektronik Mühendisliđi Anabilim Dalı Doktora öđrencisi Hasan Serhan Yavuz'un DOKTORA tezi olarak hazırladıđı "Görüntü Tanıma Uygulamalarında Tensörel Ortak Bileşen Yöntemleri" başlıklı bu çalışma, jürimizce lisansüstü yönetmeliđin ilgili maddeleri uyarınca deđerlendirilerek kabul edilmiştir.

Danışman : Yrd. Doç. Dr. M. Atıf Çay

İkinci Danışman : ———

Doktora Tez Savunma Jürisi:

Üye : Yrd. Doç. Dr. M. Atıf Çay

Üye : Prof. Dr. Atalay Barkana

Üye : Prof. Dr. İdris Dađ

Üye : Prof. Dr. Abdurrahman Karamancıođlu

Üye : Doç. Dr. Ömer Nezih Gerek

Fen Bilimleri Enstitüsü Yönetim Kurulu'nun tarih ve sayılı kararıyla onaylanmıştır.

Prof. Dr. Nimetullah BURNAK

Enstitü Müdürü

ÖZET

Bu tez çalışmasında, sayısal görüntü tanıma uygulamalarında kullanılacak, ortak bileşen öznitelikleri üzerinden, vektörel yöntemlerden daha iyi tanıma başarımlarını gerçekleştirebilen yeni yöntemler sunulmuştur. Tensörel ortak bileşen yöntemleri olarak adlandırılan bu yeni yöntemlerde, tensör biçiminde temsil edilen veri örneklerinin orijinal yapısını değiştirme zorunluluğu yoktur. Çalışmada, üç farklı tensörel ortak bileşen yöntemi önerilmiştir. Yöntemlerden birincisi, yetersiz veri durumu için ortak vektör yaklaşımının tensör cebirine doğal bir genişlemesidir. İkinci yöntem, yüksek dereceden tekil değer ayrıştırmasına dayanan farklı bir yöntemdir. Üçüncü yöntemse, yüksek dereceden tekil değer ayrıştırması temeline dayanan tensörel ortak bileşen yönteminin multilineer diskriminant analizi kavramlarıyla birleştirilmiş olan yeni bir biçimidir. Yöntemlerin sayısal görüntü tanıma başarımları, çeşitli yüz veritabanları ve bir nesne veritabanı üzerinde yapılan deneylerle test edilmiş ve uygulamalar sonucunda genelde başarılı sonuçlar elde edilmiştir.

Anahtar Kelimeler: Görüntü Tanıma, Öznitelik Çıkarımı, Sınıflandırma, Tensör

SUMMARY

In this thesis study, new methods which can be used in digital image recognition applications on common component features to achieve better recognition efficiencies than vector based methods have been presented. For these new methods, called as tensor based common component methods, there is no obligation of conversion of the form of the data samples which are represented by tensors. In the present study, three tensor based common component methods have been proposed. The first method is the natural extension of the common vector method for the insufficient case into the tensor algebra. The second method is a different method which is based on higher order singular value decomposition. The third method is a new form of the higher order singular value decomposition based common component method in connection with multilinear discriminant analysis concepts. Digital image recognition efficiencies of the methods have been tested by making the experiments on some face databases and an image database. It has been observed that the application results have been found to be successful in general.

Keywords: Image Recognition, Feature Extraction, Classification, Tensor

TEŞEKKÜR

Uzun, zorlu ve yorucu olan doktora çalışmam sırasında, bana danışmanlık ederek, bana olan güvenini, moral desteğini, yardımlarını ve anlayışını esirgemeyen danışman hocam Yrd. Doç. Dr. M. Atıf Çay'a sonsuz teşekkürlerimi sunarım.

Doktora çalışmalarımın başından bugüne kadar engin birikimiyle, tez konusunun belirlenmesinden çalışma sonuçlanıncaya kadar bilgisini ve yönlendirici önerilerini benden esirgemeyen, titizlik ve kaliteli akademisyenlik adına kendisini her zaman örnek alacağım, saygıdeğer hocam Prof. Dr. Atalay Barkana'ya en içten teşekkürlerimi sunmaktan onur duyarım.

Çalışmalarımı takdir edip, benim için çok kıymetli övgü dolu sözlerini ve yüreklendirici desteğini her fırsatta dile getirmekten kaçınmayan, değerli bir akademisyen ve benim için değerli bir insan olan sayın hocam Prof. Dr. İdris Dağ'a en içten teşekkürlerimi sunarım.

Son olarak da aileme, ne yaptığımı hiç anlamasalar da bana sonuna kadar güvendikleri ve benim için yaptıkları sonsuz fedakarlıklar için ve sevgili eşime, en çok ihtiyacım olduğu zamanlarda yanımda olup beni cesaretlendirdiği için teşekkür ederim.

İÇİNDEKİLER

	<u>Sayfa</u>
ÖZET	v
SUMMARY	vi
TEŞEKKÜR	vii
ŞEKİLLER DİZİNİ	xi
ÇİZELGELER DİZİNİ	xii
SİMGELER VE KISALTMALAR DİZİNİ	xiii
1. GİRİŞ	1
2. ORTAK VEKTÖR YÖNTEMLERİ	5
2.1 Ortak Vektör Yaklaşımları.....	5
2.1.1 Yetersiz veri durumu için ortak vektör yaklaşımı.....	8
2.1.1.1 Gram-Schmidt süreci kullanarak fark altuzayı üzerinden ortak vektör belirleme.....	8
2.1.1.2 Kovaryans ölçüsü üzerinden ortak vektör belirleme.....	10
2.1.2 Yeterli veri durumu için ortak vektör yaklaşımı.....	11
2.1.3 Ortak vektör yaklaşımı sınıflandırıcısı.....	12
2.2 Ayırt Edici Ortak Vektör Yöntemi.....	13
2.2.1 Gram-Schmidt süreci kullanarak ayırt edici ortak vektör belirleme.....	13
2.2.2 Sınıf içi saçılım üzerinden ayırt edici ortak vektör belirleme.....	15
2.2.3 Ayırt edici ortak vektör yöntemi sınıflandırıcısı.....	16
3. MATRİS TABANLI YÖNTEMLER	18
3.1 İki Boyutlu Temel Bileşen Analizi.....	19
3.2 İki Boyutlu Altuzay Sınıflandırıcıları.....	23
3.2.1 İki boyutlu CLAFIC (2D-CLAFIC) yöntemi.....	24
3.2.2 Ortalama çıkarılmış iki boyutlu CLAFIC (2D-CLAFIC- μ) yöntemi.....	25
3.2.3 İki boyutlu ALS (2DALS) yöntemi.....	26

İÇİNDEKİLER (devam)

	<u>Sayfa</u>
3.3 Yeni Bir Ortak Matris Yaklaşımı: Ortak Matris-2.....	28
3.3.1 Yetersiz veri durumu için Ortak Matris-2 yöntemi.....	31
3.3.2 Yeterli veri durumu için Ortak Matris-2 yöntemi.....	34
3.3.3 Ortak Matris-2 yöntemi sınıflandırıcı kararı.....	36
4. YÜKSEK DERECEDEKİ TENSÖR TABANLI YÖNTEMLER.....	38
4.1 Multilineer Cebir Temelleri.....	38
4.1.1 Tensör – matris – vektör dönüşümleri.....	39
4.1.2 Tensör – matris çarpımı.....	41
4.1.3 Skaler çarpım.....	42
4.1.4 Tensör ayrıştırılmaları.....	42
4.2 Multilineer Temel Bileşen Analizi Yöntemi.....	44
4.3 Multilineer Diskriminant Analizi Yöntemi.....	47
5. TENSÖREL ORTAK BİLEŞEN YÖNTEMLERİ.....	52
5.1 Gram-Schmidt Birimdikleştirme Temeline Dayanan Tensörel Ortak Bileşen Yöntemi.....	52
5.2 Yüksek Dereceden Tekil Değer Ayrıştırması Temeline Dayanan Tensörel Ortak Bileşen Yöntemi.....	55
5.3 Tensör Diskriminant Analizi Temeline Dayanan Tensörel Ortak Bileşen Yöntemi.....	60
6. SAYISAL GÖRÜNTÜ TANIMA DENEYLERİ.....	64
6.1 Yale Veritabanı Yüz Tanıma Deneyleri.....	65
6.2 ORL Veritabanı Yüz Tanıma Deneyleri.....	68
6.3 AR Veritabanı Yüz Tanıma Deneyleri.....	70
6.4 COIL100 Veritabanı Nesne Tanıma Deneyleri.....	77
7. SONUÇ VE ÖNERİLER.....	81

İÇİNDEKİLER (devam)

	<u>Sayfa</u>
8. KAYNAKLAR DİZİNİ.....	86
ÖZGEÇMİŞ.....	91

ŞEKİLLER DİZİNİ

<u>Sekil</u>	<u>Sayfa</u>
4.1 Üçüncü dereceden bir tensör örneği.....	39
4.2 Üçüncü dereceden bir \mathcal{A} tensörünün matris dilimleri.....	40
4.3 Üçüncü dereceden bir \mathcal{A} tensörünün mod- n vektörleri.....	40
6.1 Yale veritabanından seçilen 4 bireyin görüntü örnekleri.....	66
6.2 ORL veritabanından seçilen 4 bireyin görüntü örnekleri.....	69
6.3 AR veritabanından bir bireyin birinci oturum görüntüleri.....	71
6.4 AR veritabanından ön-işlenmiş çeşitli görüntü örnekleri.....	72
6.5 AR veritabanı 50 sınıflı deney için seçilen sınıflar.....	72
6.6 COIL100 veritabanındaki bir nesnenin görüntüleri.....	77
6.7 COIL100 veritabanından seçilen 40 nesne.....	78

ÇİZELGELER DİZİNİ

<u>Cizelge</u>	<u>Sayfa</u>
6.1 Yale yüz veritabanı deney sonuçları.....	67
6.2 ORL yüz veritabanı deney sonuçları.....	69
6.3 AR yüz veritabanı 50 sınıf gri seviyeli görüntüler için deney sonuçları.....	73
6.4 AR yüz veritabanı 50 sınıf renkli görüntüler için deney sonuçları.....	74
6.5 AR yüz veritabanı 117 sınıf gri seviyeli görüntüler için deney sonuçları.....	75
6.6 AR yüz veritabanı 117 sınıf renkli görüntüler için deney sonuçları.....	76
6.7 COIL100 veritabanı gri seviyeli 40 nesne görüntüleri için deney sonuçları.....	78
6.8 COIL100 veritabanı renkli 40 nesne görüntüleri için deney sonuçları.....	79

SİMGELER VE KISALTMALAR DİZİNİ

<u>Kısaltmalar</u>	<u>Açıklama</u>
a, b, \dots	Skaler
$\mathbf{a}, \mathbf{b}, \dots$	Vektör
$\mathbf{A}, \mathbf{B}, \dots$	Matris
$\mathcal{A}, \mathcal{B}, \dots$	Tensör
2D-ALS	İki boyutlu ALS
2D-CLAFIC	İki boyutlu CLAFIC
2D-CLAFIC $-\mu$	Ortalama çıkarılmış iki boyutlu CLAFIC
2DLDA	İki boyutlu LDA
2DPCA	İki boyutlu PCA
ALS	Averaged Learning Subspace
AMD	Assembled Matrix Distance measure (Birleştirilmiş Matris Uzaklık ölçüsü)
AOVY	Ayırt Edici Ortak Vektör Yöntemi
CANDECOMP	CANonical DECOMPosition
CLAFIC	Class-Featuring Information Compression
CLAFIC $-\mu$	Ortalama çıkarılmış CLAFIC
CP	CANDECOMP-PARAFAC ayrıştırması
EKK	En Küçük Kareler
HOSVD	Higher Order Singular Value Decomposition (Yüksek dereceden tekil değer ayrıştırması)
IMPCA	Image-PCA
LDA	Linear Discriminant Analysis (Doğrusal Diskriminant Analizi)
MDA	Multilinear Discriminant Analysis (Multilinear Diskriminant Analizi)
MPCA	Multilinear Principal Component Analysis (Multilinear Temel Bileşen Analizi)
OM2	Ortak Matris-2
OVY	Ortak Vektör Yaklaşımı
PARAFAC	PARAllel FACTors

SİMGELER VE KISALTMALAR DİZİNİ (devam)

<u>Kısaltmalar</u>	<u>Açıklama</u>
PCA	Principal Component Analysis (Temel Bileşen Analizi)
SVD	Singular Value Decomposition (Tekil Değer Ayrıştırması)
TOBY	Tensörel Ortak Bileşen Yöntemi
TOBY(GSO)	Gram-Schmidt Birindikleştirme Temeline Dayanan Tensörel Ortak Bileşen Yöntemi
TOBY(HOSVD)	Yüksek Dereceden Tekil Değer Ayrıştırması Temeline Dayanan Tensörel Ortak Bileşen Yöntemi
TOBY(HOSVD) – 1	Tek Ortak Bileşen Tahminli TOBY(HOSVD) Yöntemi
TOBY(HOSVD) – 2	İki Ortak Bileşen Tahminli TOBY(HOSVD) Yöntemi
TOBY(MDA)	Tensör Diskriminant Analizi Temeline Dayanan Tensörel Ortak Bileşimi Yöntemi

BÖLÜM 1

GİRİŞ

Son yıllardaki teknolojik gelişmeler, makine kullanımını endüstriyel alanlardan evlere, hatta kişisel kullanımlara kadar yaygınlaştırmıştır. Yarı iletken endüstrisinin elektronik devre tasarımlarını minyatürleştirilmesi kolaylığından sonra, tasarlanan sayısal elektronik devrelerin programlanabilirlik avantajı sayısal bilgisayar gelişimini tetiklemiştir. Bunun sonucu olarak, bilgisayar teknolojisi büyük bir hızla gelişmiştir ve günümüzde de gelişimini sürdürmektedir. Halen kişisel bilgisayarlar ve çevrebirimi cihazları sayesinde yaşam konforunu arttırıcı ve birçok alanda gerekli uygulamaları kolaylaştırıcı sistemler tasarlayıp kullanıcıların hizmetine sunmak mümkündür. Günümüzde yapılan çalışmalar çevreyi algılayabilen, düşünebilen, insanlarla sözlü ve görsel iletişim kurabilen akıllı makineler tasarlama amacına yöneliktir.

Makine algılamasını gerçekleştirmek için geliştirilen yöntemler genelde örüntü tanıma (pattern recognition) yöntemleri olarak adlandırılır. Örüntü tanımanın biyometrik, diyagnostik, askeri amaçlı uygulamalar gibi birçok uygulamaları mümkündür. El yazısı karakterleri tanıma, insan yüzlerini, yüz ifadelerini tanıma, potansiyel düşman hedeflerini tanıma uygulamaları örnek olarak verilebilir. Örüntü bir nesne, bir süreç veya isimlendirilebilen herhangi bir olay olabilir (Duda, et al., 2001). Örüntü sınıfları, birbirleriyle benzer özellikler taşıyan örüntülerin oluşturduğu gruplardır. Herbir örüntü sınıfı içindeki örüntüler birbiriyle benzeşir, farklı örüntü sınıfları içindeki örüntüler benzeşmez. Eğitici bir örüntü tanıma uygulamasında örüntü sınıfları genellikle önceden tanımlanır ve tasarlanan örüntü tanıma mekanizması, eğitim verisi kullanılarak eğitilir. Tanıma işlemi, mekanizmanın son basamağıdır. Eğitilen sisteme test verisi gönderildiğinde test örüntüsü, eğitilen örüntü sınıflarından birisine atanarak işlem tamamlanır.

Öznitelik çıkarma (feature extraction) işlemi örüntü tanımanın en önemli aşamasıdır. Veri uzayı, genellikle çok boyutlu olmakla birlikte, yalın haliyle

incelendiğinde örüntü sınıflarının birbirinden ayrılmasının basit olmadığı bir biçimdedir. Başarılı bir örüntü tanıma işlemi gerçekleştirmek için aynı örüntü sınıfındaki verilerin belirleyici özellikleri, diğer örüntü sınıflarının belirleyici özelliklerinden kolaylıkla ayırt edilebilecek bir uzaya haritalanmalıdır. Veri uzayını öznelik (feature) uzayına haritalama işlemi öznelik çıkarma işlemi olarak adlandırılır (Fukunaga, 1990). Öznelik uzayı, genellikle veri uzayından çok daha az boyuta sahiptir. Bu nedenle literatürde boyut indirgeme biçiminde anılan yöntemler (Burges, 2004), uygun şekilde işlendiği takdirde, öznelik çıkarma yöntemi olarak kullanılabilir.

Karhunen-Loeve dönüşümü olarak da adlandırılan temel bileşen analizi (PCA), örüntü tanıma uygulamalarında, öznelik çıkarma amacıyla kullanılan en temel yöntemdir. Yüz tanıma uygulamalarında PCA kullanımı ilk kez Turk ve Pentland (1991) tarafından yapılmıştır. Bundan sonra PCA esasına dayalı veya PCA ile ilişkisi olan birçok yöntem önerilmiştir. Bu yöntemlerin arasında Fisher doğrusal diskriminant analizi (LDA) kullanan yöntemler de başarılı sonuçlar verir (Belhumeur, et. al, 1997).

Temel bileşen analizi ilkeleriyle vektörel altuzay yöntemlerini ilişkilendirmek de mümkündür. Bu bakış açısından, Ortak Vektör Yaklaşımı (OVY) (Gulmezoglu, et al., 2001), bir diğer önemli vektörel örüntü tanıma yöntemidir. OVY ilk olarak ses tanıma uygulamaları için önerilmiş (Gulmezoglu, et al., 1999), daha sonra çeşitli örüntü tanıma uygulamalarında denenmiştir (Çokar, 2000; Günel, 2003; Demirkaya, 2004).

Vektörel örüntü tanıma yöntemlerinin görüntü tanıma uygulamalarında kullanılabilmesi için, özgün olarak matris biçiminde temsil edilen sayısal görüntü verilerinin vektöre dönüştürülme zorunluluğu vardır. Matris biçimindeki verilere uygulanan iki boyutlu temel bileşen analizi (2DPCA) yöntemi (Yang, et al., 2002; 2004) bu zorunluluğu ortadan kaldıran ilk çalışmalardandır. Buna paralel olarak iki boyutlu doğrusal diskriminant analizi (2DLDA) (Li and Yuan, 2005) yöntemi ve bu yaklaşımların türevlerinin, görüntü tanıma uygulamalarında başarılı sonuçlar verdiği belirtilmiştir (Chen, et al., 2005; Kong, et al., 2005; Zhang and Zhou, 2005; Jing, et al., 2006).

Daha sonra, sinyal işleme problemlerinin büyük bir kısmının ikiden daha fazla sayıda indislerle temsil edilen veri biçimleriyle uğraşmak durumunda kalması sebebiyle, tek indisle betimlenen vektörler ve iki indisle ifade edilen matrislere ilave olarak, ikiden fazla sayıda indisle temsil edilen yüksek dereceli tensör verilerini işleyen yöntemler yaygınlık kazanmıştır. Yüksek dereceden tensörler kullanarak işlem yapabilmek için bilinen vektör ve matris cebiri yetersiz kalmaktadır. Vektör kavramları üzerine kurulu olan lineer cebirin vektör uzayları tanımlarından genişletilerek oluşturulan, yüksek dereceden tensör temellerini içeren ve tensör uzayları tanımlarının yapıldığı cebir, multilineer cebir olarak adlandırılır (Lathauwer, et al., 2004). Görüntü tanımda tensör verilerin kullanımı, yüksek dereceden tekil değer ayrıştırması ve diğer bazı tensör ayrıştırmaları (Lathauwer, et al., 2000; Vasilescu and Terzopoulos, 2002; Bader and Kolda, 2006) sayesinde hayat bulmuştur. Tekil değer ayrıştırması kullanan PCA tabanlı birçok yöntemin yüksek dereceden tensör verileri için genişletilmesi bu sayede mümkündür. Örnek olarak, klasik temel bileşen analizinin tensör uyarlaması olan multilineer temel bileşen analizi (Lu, et al., 2006), klasik diskriminant analizinin multilineer genişlemesi olan multilineer diskriminant analizi (Yan, et al., 2007) gösterilebilir.

Bu tez çalışmasında, sayısal görüntü tanıma uygulamalarında kullanılacak, ortak bileşen öznelikleri üzerinden, vektörel yöntemlerden daha iyi tanıma başarımlarını gerçekleştirecek yeni yöntemler geliştirmek hedeflenmiştir. Ortak bileşen fikrinin esin kaynağı önceki paragraflarda bahsedilen ortak vektör yaklaşımıdır. Bu yüzden ikinci bölümde ortak vektör yöntemleri ayrıntılı biçimde anlatılmıştır. Vektörel yöntemlerden yüksek dereceden tensör verilerine uygulanabilecek yöntemlere geçiş yapabilmek için, 2DPCA yönteminin ışığı altında, çeşitli matris tabanlı yöntemler incelenmiş ve ilk çalışmalar matris tabanlı yöntemler üzerine yapılmıştır. Bu çalışmalar sırasında iki boyutlu altuzay sınıflandırıcıları önerilmiş ve yayımlanmıştır (Cevikalp, et al., 2008). Geliştirdiğimiz matris tabanlı yöntemler ve iki boyutlu yaklaşım temeline dayandırdığımız yeni bir ortak matris yaklaşımı, iki boyutlu yöntemlerin çıkış noktası olan 2DPCA yöntemiyle birlikte üçüncü bölümde verilmiştir. Dördüncü bölümde, yüksek dereceden tensörler için tanımlanan temel multilineer cebir kavramları tanıtılmış ve multilineer cebir esasına dayalı, günümüzde yoğun olarak ilgi çeken çeşitli yöntemler anlatılmıştır. Multilineer cebir tanımlarından yararlanarak ve ortak vektör

yaklaşımı teorileri göz önünde bulundurularak geliştirdiğimiz, vektör, matris veya daha yüksek dereceden tensör verilerine uygulanabilen ortak bileşen yöntemleri ise beşinci bölümde anlatılmıştır. Altıncı bölümde, tez çalışmasında geliştirilen yöntemlerin ve diğer klasik yöntemlerin görüntü tanıma performanslarının ölçülmesi için yapılan görüntü tanıma deneylerine yer verilmiştir. Çalışmanın son bölümü olan yedinci bölümde elde edilen sonuçlar irdelenerek çeşitli önerilerde bulunulmuştur.

Tez çalışmasında skaler, vektör, matris ve tensör niceliklerinin karışmaması için formülasyonlarda skalerler küçük yatık harflerle (a, b, \dots), vektörler küçük koyu harflerle ($\mathbf{a}, \mathbf{b}, \dots$), matrisler büyük koyu harflerle ($\mathbf{A}, \mathbf{B}, \dots$) ve daha yüksek dereceden tensörler ise büyük süslü harflerle ($\mathcal{A}, \mathcal{B}, \dots$) temsil edilecektir.

BÖLÜM 2

ORTAK VEKTÖR YÖNTEMLERİ

Bu bölüm, Gulmezoglu, et al. (1999; 2001; 2007) tarafından sunulan ortak vektör yaklaşımları ve Cevikalp, et al., (2005) tarafından sunulan ayırt edici ortak vektör yönteminin bizim bakış açımızdan özetini içermektedir. Ortak vektör yaklaşımları Bölüm 2.1’de, ayırt edici ortak vektör yöntemi ise Bölüm 2.2’de verilmiştir.

2.1 Ortak Vektör Yaklaşımları

Aynı sınıfa ait olduğu bilinen eğitim verisi kümesi, \mathcal{R}^d uzayında tanımlanan M adet gözlem vektöründen oluşsun. Gözlem vektörleri ($\mathbf{a}_k \in \mathcal{R}^d$, $k=1,2,\dots,M$) bir matrisin sütunlarını oluşturacak biçimde dizilerek \mathbf{D} sembolü ile gösterilen sınıf eğitim verisi matrisi (bundan sonra devam eden yazıda kısaca eğitim verisi olarak kullanılacaktır), Eşitlik 2.1’de verildiği gibi oluşturulur.

$$\mathbf{D} = [\mathbf{a}_1 \quad \mathbf{a}_2 \quad \cdots \quad \mathbf{a}_M] = \begin{bmatrix} a_{11} & \cdots & a_{M1} \\ \vdots & \ddots & \vdots \\ a_{1d} & \cdots & a_{Md} \end{bmatrix} \quad (2.1)$$

Ortak vektör yaklaşımının amacı, aynı sınıfa ait olduğu bilinen gözlem vektörlerini tek bir ortak vektörle temsil edip, sınıf özniteliği olarak sınıf ortak vektörünü kullanmaktır. Bunun için her bir gözlem vektörü, sınıf ortak vektörü ve sınıf farklılık vektörlerinin birleşiminden oluşmalıdır (Gulmezoglu, et al., 1999; 2001). Hata metriği ilave edilerek oluşturulan doğrusal toplamsal model Eşitlik 2.2’de gösterilmiştir.

$$\mathbf{a}_k = \mathbf{a}_{com} + \mathbf{a}_{k,dif} + \boldsymbol{\varepsilon}_k, \quad k=1,2,\dots,M \quad (2.2)$$

Burada \mathbf{a}_{com} ortak vektördür, $\mathbf{a}_{k,dif}$ k 'inci gözlem vektörünün ilgili sınıftaki farklılığını ifade eden fark vektörüdür ve $\boldsymbol{\varepsilon}_k$ ise hata minimizasyon metriği tanımlayabilmek için ilave edilen hata vektörüdür. Buna göre sırasıyla hata vektörü Eşitlik 2.3'de ve minimize edilmesi gereken en küçük kareler (EKK) hata ölçüsü Eşitlik 2.4'de verilmiştir.

$$\boldsymbol{\varepsilon}_k = \mathbf{a}_k - \mathbf{a}_{k,dif} - \mathbf{a}_{com}, \quad k = 1, 2, \dots, M \quad (2.3)$$

$$F = \sum_{\forall k} \|\boldsymbol{\varepsilon}_k\|^2 = \sum_{k=1}^M \|\mathbf{a}_k - \mathbf{a}_{k,dif} - \mathbf{a}_{com}\|^2 \quad (2.4)$$

Yapılabilecek en küçük hata sıfır olabilir. Bunun için gözlem vektörü Eşitlik 2.5'deki gibi yazılmalıdır.

$$\mathbf{a}_k = \mathbf{a}_{com} + \mathbf{a}_{k,dif}, \quad k = 1, 2, \dots, M \quad (2.5)$$

Denklem 2.5'de gösterilen eşitlik ancak gözlem vektörleri tarafından gerilen (spanned) uzayda, birbirine dik iki altuzayın tanımlanabilmesiyle sağlanabilir. Öyle ki, \mathbf{a}_{com} vektörünün gerdiği (span) altuzay gözlemlerin ortak özelliklerini; $k = 1, 2, \dots, M$ için $\mathbf{a}_{k,dif}$ vektörlerinin gerdiği altuzaysa gözlemlerin farklı olan özelliklerini biçimlendiren altuzaylar olmalı ve bu iki altuzayın birleşiminden oluşan uzaysa, $k = 1, 2, \dots, M$ için \mathbf{a}_k vektörleri tarafından gerilmelidir.

B uzayı, $k = 1, 2, \dots, M$ için $\mathbf{a}_{k,dif}$ vektörlerinin gerdiği uzay olsun. O halde B uzayı \mathfrak{R}^d 'nin bir altuzayı olur. B^\perp uzayı ise B altuzayının tümleyeni olsun. Bu durumda da $B + B^\perp = \mathfrak{R}^d$ olur. Fark vektörlerinin gerdiği bir B altuzayı bulunursa, bu fark vektörleri gözlem vektörünün B altuzayına olan izdüşümüyle hesaplanabilir. \mathbf{P}_{cov} , B altuzayına olan izdüşüm matrisi olarak tanımlanırsa, sütun gözlem vektörlerinin izdüşümü Eşitlik 2.6'da gösterilen fark vektörlerini verir.

$$\mathbf{a}_{k,dif} = \mathbf{P}_{cov}^T \mathbf{a}_k, \quad k = 1, 2, \dots, M \quad (2.6)$$

Fark vektörleri belirlendikten sonra, sınıf ortak vektörü, herhangi bir gözlem vektörünün kendisinden fark vektörünün çıkarılmasıyla Eşitlik 2.7’de verildiği gibi hesaplanabilir.

$$\mathbf{a}_{com} = \mathbf{a}_k - \mathbf{a}_{k,dif}, \quad k = 1, 2, \dots, M \quad (2.7)$$

Sınıfın ortak vektörü tek ve değişmezdir (Gulmezoglu, et al., 1999; 2001). Bir başka ifadeyle tüm $k = 1, 2, \dots, M$ değerleri için aynı ortak vektör bulunmalıdır.

Aslında, B altuzayı belirlendikten sonra, ortak vektör bulunması B ’nin tümleyen uzayı üzerinden de yapılabilir. B altuzayı, farklılıkları ölçülendirdiği için farklılık altuzayı olarak adlandırılırken, onun tümleyeni B^\perp altuzayı da, farksızları ölçülendiren farksızlık altuzayı olur. B^\perp ve B tümleyen altuzaylar olduğu için, bu iki uzaya izdüşümü alınan herhangi bir vektörün izdüşümleri birbirine diktir. B^\perp altuzayına izdüşüm matrisi de \mathbf{P}_{cov}^\perp şeklinde gösterilirse, Eşitlik 2.8’deki gibi gözlem vektörünün B^\perp altuzayına olan izdüşümüyle hesaplanan vektör, doğrudan ortak vektörü verir.

$$\mathbf{a}_{com} = \left(\mathbf{P}_{cov}^\perp \right)^T \mathbf{a}_k, \quad k = 1, 2, \dots, M \quad (2.8)$$

Ortak vektör yaklaşımındaki en önemli nokta, farklılık – farksızlık altuzay tanımlamaları ve bu altuzaylara olan izdüşümlerin belirlenmesidir. \mathfrak{R}^d uzayında, birbirine tümleyen olan farklılık ve farksızlık altuzayları ancak gözlem sayısı M ’nin vektör boyutu d ’den küçük veya eşit olduğu durumda bulunabilir. Bu durum, yetersiz veri durumu olarak adlandırılır (Gulmezoglu et al., 2007). Yetersiz veri durumunda Eşitlik 2.5 geçerlidir. Gözlem sayısı M ’nin vektör boyutu d ’den daha büyük olduğu durum ise yeterli veri durumudur (Gulmezoglu et al., 2007) ve bu durumda ortak vektör çözümü Eşitlik 2.2 üzerinden yapılabilir. Tezin ilerleyen kısımlarında, yeterli ve yetersiz veri durumunda ortak vektör yaklaşımı yöntemleri farklı alt başlıklar altında incelenecektir.

2.1.1 Yetersiz veri durumu için ortak vektör yaklaşımı

Yetersiz veri durumu, aynı sınıftaki gözlem sayısının gözlem vektörü boyutuna eşit veya daha az ($M \leq d$) olduğu durumdur. Bu durumda birbirine tümleyen fark ve farksızlık altuzayları bulunabilir. Eşitlik 2.4’de verilen hata ölçüsü sıfırdır ve gözlem vektörü Eşitlik 2.5’de yazıldığı gibi tanımlanabilir. Yetersiz veri durumuna uygun olan eğitim verisi için ortak vektör belirleme süreci, (Gulmezoglu, et al., 1999) kaynakçasındaki gibi Gram-Schmidt birimdikleştirme (orthonormalization) yöntemi kullanılarak fark bileşenlerinin yok edilmesi ile mümkün olabildiği gibi, (Gulmezoglu, et al., 2001) kaynakçasındaki gibi PCA ile ilişkilendirilerek kovaryans matrisinin tekil değer ayrıştırması sonucunda da bulunabilir. Her iki yöntem de aynı ortak vektörü verir ancak izlenen süreç farklıdır. Bu yüzden iki yöntem farklı alt başlıklarda özetlenmiştir.

2.1.1.1 Gram-Schmidt süreci kullanarak fark altuzayı üzerinden ortak vektör belirleme

Bu yöntemde, sınıftaki gözlem vektörlerinden herhangi biri alınıp geriye kalan diğer gözlem vektörlerinden çıkarılarak $M-1$ adet fark vektörü ($\mathbf{b}_j \in \mathcal{R}^d$, $j=1,2,\dots,M-1$) bulunur. Örneğin \mathbf{a}_1 vektörü diğerlerinden çıkarıldığında fark vektörleri Eşitlik 2.9’daki gibi olur.

$$\left. \begin{array}{l} \mathbf{b}_1 = \mathbf{a}_2 - \mathbf{a}_1 \\ \mathbf{b}_2 = \mathbf{a}_3 - \mathbf{a}_1 \\ \vdots \\ \mathbf{b}_{M-1} = \mathbf{a}_M - \mathbf{a}_1 \end{array} \right\} \Rightarrow \mathbf{B} = [\mathbf{b}_1 \quad \mathbf{b}_2 \quad \dots \quad \mathbf{b}_{M-1}] = \begin{bmatrix} b_{11} & \dots & b_{(M-1)1} \\ \vdots & \ddots & \vdots \\ b_{1d} & \dots & b_{(M-1)d} \end{bmatrix} \quad (2.9)$$

Fark vektörlerinin farklılık altuzayını gerebilmesi için bu vektörler Gram-Schmidt süreci kullanılarak birimdikleştirilir. Gram-Schmidt birimdikleştirme süreci kısaca aşağıdaki gibi özetlenmiştir (Moon and Stirling, 2000).

Gram-Schmidt birimdikleştirme süreci

1. İlk vektör birimleştirilir: $\mathbf{z}_1 = \frac{\mathbf{b}_1}{\|\mathbf{b}_1\|}$

2. \mathbf{b}_2 vektörünün \mathbf{z}_1 üzerine olan izdüşümü kendinden çıkarılır; \mathbf{e}_2 olarak adlandırılan bu vektör, ortogonalite teoremine göre \mathbf{z}_1 'e diktir.

$$\mathbf{e}_2 = \mathbf{b}_2 - \langle \mathbf{b}_2, \mathbf{z}_1 \rangle \mathbf{z}_1$$

3. Eğer \mathbf{e}_2 sıfırdan farklıysa normleştirilerek \mathbf{z}_2 bulunur; \mathbf{e}_2 sıfıra eşitse $\mathbf{z}_2 \in \text{span}(\mathbf{z}_1)$ 'dir, bu durumda \mathbf{e}_2 ihmal edilir.

$$\mathbf{z}_2 = \frac{\mathbf{e}_2}{\|\mathbf{e}_2\|} \quad (\mathbf{e}_2 \neq 0 \text{ ise})$$

4. 2 ve 3'üncü basamak son \mathbf{b} vektörüne kadar tekrarlanır ve süreç tamamlanır.

$$\mathbf{e}_3 = \mathbf{b}_3 - \langle \mathbf{b}_3, \mathbf{z}_1 \rangle \mathbf{z}_1 - \langle \mathbf{b}_3, \mathbf{z}_2 \rangle \mathbf{z}_2, \quad \mathbf{z}_3 = \frac{\mathbf{e}_3}{\|\mathbf{e}_3\|} \quad (\mathbf{e}_3 \neq 0 \text{ ise})$$

$$\mathbf{e}_4 = \mathbf{b}_4 - \langle \mathbf{b}_4, \mathbf{z}_1 \rangle \mathbf{z}_1 - \langle \mathbf{b}_4, \mathbf{z}_2 \rangle \mathbf{z}_2 - \langle \mathbf{b}_4, \mathbf{z}_3 \rangle \mathbf{z}_3, \quad \mathbf{z}_4 = \frac{\mathbf{e}_4}{\|\mathbf{e}_4\|} \quad (\mathbf{e}_4 \neq 0 \text{ ise})$$

⋮

$$\mathbf{e}_j = \mathbf{b}_j - \sum_{i=1}^{j-1} (\langle \mathbf{b}_j, \mathbf{z}_i \rangle \mathbf{z}_i), \quad \mathbf{z}_j = \frac{\mathbf{e}_j}{\|\mathbf{e}_j\|} \quad (\mathbf{e}_j \neq 0 \text{ ise})$$

Süreç sonunda bulunan \mathbf{z}_j vektörleri, fark altuzayı B 'nin taban vektörleridir. Bundan sonra, fark altuzayına izdüşürülerek bulunan fark vektörleri ve ortak vektör sırasıyla Eşitlik 2.10 ve Eşitlik 2.11 kullanılarak belirlenebilir.

$$\mathbf{a}_{k,dif} = \langle \mathbf{a}_k, \mathbf{z}_1 \rangle \mathbf{z}_1 + \langle \mathbf{a}_k, \mathbf{z}_2 \rangle \mathbf{z}_2 + \cdots + \langle \mathbf{a}_k, \mathbf{z}_{M-1} \rangle \mathbf{z}_{M-1}, \quad k = 1, 2, \dots, M \quad (2.10)$$

$$\mathbf{a}_{com} = \mathbf{a}_k - \mathbf{a}_{k,dif} \quad \forall k = 1, 2, \dots, M \quad (2.11)$$

Sınıf içindeki tüm gözlemler için Eşitlik 2.11 aynı ortak vektörü verir (Gulmezoglu, et al., 1999).

2.1.1.2 Kovaryans ölçüsü üzerinden ortak vektör belirleme

Yetersiz veri durumu için fark ve farklılık altuzayları, gözlem vektörlerinin kovaryans matrisine özdeğer-özvektör ayrıştırması yapılarak da belirlenebilir (Gulmezoglu, et al., 2001). Bunun için öncelikle Eşitlik 2.12’de \mathbf{m}_a sembolüyle gösterilen sınıf gözlemlerinin ortalama vektörü, daha sonra sütun gözlem vektörleri için Eşitlik 2.13’de verilen $\mathbf{C} \in \Re^{d \times d}$ kovaryans matrisi hesaplanır.

$$\mathbf{m}_a = \frac{1}{M} \sum_{k=1}^M \mathbf{a}_k \quad (2.12)$$

$$\mathbf{C} = \frac{1}{M-1} \sum_{k=1}^M (\mathbf{a}_k - \mathbf{m}_a)(\mathbf{a}_k - \mathbf{m}_a)^T \quad (2.13)$$

Bir sonraki aşamada kovaryans matrisine özdeğer-özvektör ayrıştırması yapılır. Özdeğer-özvektör probleminin özelliği gereği, veri kovaryans matrisinin özvektörleri, lineer bağımsız ve birbirine diktir. Ayrıca $M \leq d$ olduğu için özdeğerler pozitifdir ve bazıları sıfıra eşittir. Sıfıra eşit olan özdeğerlere karşılık gelen özvektörler, farksızlık altuzayının taban vektörleri iken, sıfırdan farklı olan özdeğerlere karşılık gelen özvektörlerse farklılık altuzayının taban vektörleridir. Özdeğerlerin büyükten küçüğe doğru dizildiği kabul edildiğinde sıfırdan büyük olan ilk r özdeğere karşılık gelen özvektörler B altuzayının taban vektörleri, geriye kalan $(d - r)$ adet sıfıra eşit özdeğere karşılık gelen özvektörse B^\perp altuzayının taban vektörleridir. Kovaryans matrisinin

özdeğere göre büyükten küçüğe sıralı özvektörleri \mathbf{u}_j ($j=1,2,\dots,d$) ve farklılık altuzayına olan izdüşüm matrisi de \mathbf{P}_{cov} sembolleriyle gösterilirse, özvektörlerle izdüşüm matrisi arasındaki bağıntı Eşitlik 2.14'de verilmiştir.

$$\mathbf{P}_{\text{cov}} = \sum_{j=1}^r \mathbf{u}_j \mathbf{u}_j^T \quad (2.14)$$

İzdüşüm belirlendikten sonra gözlem fark vektörleri Eşitlik 2.6'dan, sınıf ortak vektörü ise Eşitlik 2.7'den hesaplanabilir.

Ayrıca ortak vektör, Eşitlik 2.8'de gösterildiği gibi herhangi bir gözlem vektörünün doğrudan B^\perp altuzayına olan izdüşümü alınarak da belirlenebilir. Farksızlık altuzayına olan izdüşüm ise Eşitlik 2.15'de verilmiştir.

$$\mathbf{P}_{\text{cov}}^\perp = \sum_{j=r+1}^d \mathbf{u}_j \mathbf{u}_j^T \quad (2.15)$$

2.1.2 Yeterli veri durumu için ortak vektör yaklaşımı

Yeterli veri durumu, aynı sınıftaki gözlem sayısının gözlem vektörü boyutundan daha fazla ($M > d$) olduğu durumdur. Yeterli veri durumunda, fark altuzayı ve farksızlık altuzayını tümleyen altuzaylar olarak tanımlamak mümkün değildir. Gulmezoglu, et al., (2007) çalışmalarında, kovaryans yöntemi üzerinden yaklaşık olarak farksızlık altuzayını belirlemiş ve Eşitlik 2.4'deki EKK hata ölçüsünü enküçüklemek (minimization) için eğitim verisi ortalama vektörünü yaklaşık olarak belirledikleri farksızlık altuzayına izdüşürerek ortak vektörü tanımlamışlardır. Bu yöntem kısaca aşağıdaki gibi özetlenebilir.

Eğitim verisi kovaryans matrisi 2.13'de verilen eşitlikten hesaplanır ve kovaryans matrisine özdeğer-özvektör ayrıştırması yapılır. Yeterli veri durumunda sıfıra

eşit olan özdeğer yoktur. Özdeğerlerin hepsi sıfırdan büyük ve pozitiftir. Özdeğerler, bu kez küçükten büyüğe doğru sıralanır ve en küçük ilk q tane özdeğere karşılık gelen özvektörler yaklaşık olarak farksızlık altuzayını geren taban vektörleri olarak kabul edilir. Kovaryans matrisinin, özdeğerleri küçükten büyüğe doğru sıralanmış özvektörleri \mathbf{v}_j , $j=1,2,\dots,d$ sembolü ile gösterilirse, bu durumda farksızlık altuzayına olan izdüşüm Eşitlik 2.16'da verildiği gibidir.

$$\mathbf{P}_{\text{suff}}^\perp = \sum_{j=1}^q \mathbf{v}_j \mathbf{v}_j^T \quad (2.16)$$

Sınıf ortak vektörü de, Eşitlik 2.12'de verilen gözlem vektörleri ortalamasının doğrudan farksızlık altuzayına izdüşürülmesiyle, Eşitlik 2.17'de verildiği gibi belirlenir.

$$\mathbf{a}_{\text{com}} = (\mathbf{P}_{\text{cov}}^\perp)^T \mathbf{m}_a \quad (2.17)$$

2.1.3 Ortak vektör yaklaşımı sınıflandırıcısı

Çok sınıflı örüntü tanıma problemleri için ortak vektör yaklaşımında her bir sınıfın, sınıf ortak vektörleri ve ortak vektör belirleyici izdüşüm matrisleri, yeterli veya yetersiz veri durumuna göre yukarıda anlatılan yöntemlerden herhangi birisi kullanılarak belirlenebilir. Toplam N_C adet sınıf olduğu varsayıldığında, sınıf ortak vektörleri $\mathbf{a}_{\text{com}}^{(c)}$ ve farksızlık altuzayına izdüşüm matrisi $\mathbf{P}_{(c)}^\perp$ ile gösterildiğinde ($c=1,2,\dots,N_C$), sınıflandırılmak istenen bir test vektörü $\mathbf{a}_{\text{test}} \in \mathfrak{R}^d$, tüm sınıflar için öncelikle Eşitlik 2.18'de verildiği gibi ortak vektör uzayına izdüşürülür.

$$\mathbf{a}_{\text{com},\text{test}}^{(c)} = (\mathbf{P}_{(c)}^\perp)^T \mathbf{a}_{\text{test}}, \quad \forall c \quad (2.18)$$

Daha sonra sınıf ortak vektörleri ile test ortak vektörü arasındaki Öklid uzaklıkları tüm sınıflar için hesaplanır. Test ortak vektörü, Öklid uzaklığı cinsinden, hangi sınıfın ortak

vektörüne daha yakınsa (hangi sınıfın ortak vektörü ile test ortak vektörü arasındaki uzaklık en küçükse) sınıflandırma kararı ilgili sınıf olarak atanır. Bu sınıflandırıcı, ortak vektör yaklaşımı sınıflandırıcısıdır (Gulmezoglu, et al., 1999; 2001; 2007). Sınıflandırıcı karar kriteri Eşitlik 2.19’da gösterilmiştir.

$$\text{Karar sınıfı: } c^* = \arg \min_{\forall c} \left\{ \left\| \mathbf{a}_{com, test}^{(c)} - \mathbf{a}_{com}^{(c)} \right\| \right\} \quad (2.19)$$

2.2 Ayırt Edici Ortak Vektör Yöntemi

Cevikalp, et al., (2005) tarafından sunulan Ayırt edici Ortak Vektör Yöntemi (AOVY), ortak vektör yaklaşımı ve doğrusal diskriminant analizi teorilerinden hareket edilerek oluşturulan farklı bir yöntemdir. Örnek uzayı boyutunun, tüm sınıf verilerini içeren eğitim verisi sayısından daha büyük olduğu durumlar için tanımlanan yöntem, sınıf içi saçılımlara (scatter) bakarak sınıf ortak vektörlerini belirleyip, ortak vektörlerin sınıflar arası saçılımlarını enbüyükleyen (maximization) bir izdüşüm belirlemeyi hedefler. Bu izdüşüm kullanılarak çıkarılan öznitelikler de ayırt edici ortak vektörler olarak isimlendirilir. Yöntemin uygulanabilmesi için örnek uzayı boyutu $d > (M - 1)N_c$ koşulunu sağlamalıdır ve bu koşula uygun olmayan veriler için yöntem kullanılamaz.

Cevikalp, et al., (2005) makalesinde, ayırt edici ortak vektör belirleyen iki farklı algoritma önermiştir. Birincisi Gram-Schmidt birimdikleştirme sürecine dayalı bir algoritma iken, diğeri sınıf içi saçılım matrisinin özdeğer-özvektör ayrıştırmasına dayanır. Her iki algoritma, sırasıyla ilerleyen alt bölümlerde anlatılmıştır.

2.2.1 Gram-Schmidt süreci kullanarak ayırt edici ortak vektör belirleme

Her bir sınıf için sınıftaki gözlem vektörlerinden herhangi biri alınıp geriye kalan diğer gözlem vektörlerinden çıkarılarak $M - 1$ adet fark vektörü

$\{\mathbf{b}_j^c \in \mathfrak{R}^d \mid \mathbf{b}_j^c = \mathbf{a}_{j+1}^c - \mathbf{a}_1^c, c=1,2,\dots,N_C, j=1,2,\dots,M-1\}$ bulunur. Burada $\mathbf{a}_k^c \in \mathfrak{R}^d$, c 'inci sınıfın ($c=1,2,\dots,N_C$) k 'inci gözlem vektörüdür ($k=1,2,\dots,M$). Bulunan fark vektörleri, Bölüm 2.1.1.1'de verilen Gram-Schmidt birimdikleştirme süreci kullanılarak birimdikleştirilir. Birimdikleştirme sonucu $\mathbf{z}_l \in \mathfrak{R}^d$ ($l=1,2,\dots,r$) olacak biçimde ve toplamda en fazla $r = N_C(M-1)$ adet birimdik vektör elde edilir. Birimdik vektörlerin gerdiği uzay farklılık altuzayıdır. Bu durumda sınıf ortak vektörleri, Eşitlik 2.20'de verildiği gibi, sınıfa ait herhangi bir gözlem vektörünün farklılık bileşenin kendisinden çıkarılmasıyla bulunabilir.

$$\mathbf{a}_{com}^c = \mathbf{a}_k^c - \sum_{l=1}^r \langle \mathbf{a}_k^c, \mathbf{z}_l \rangle \mathbf{z}_l, \quad \forall c \quad (2.20)$$

Bir sonraki aşamada, sınıf ortak vektörlerinin fark vektörleri Eşitlik 2.21'de gösterildiği gibi belirlenir.

$$\mathbf{b}_{com}^s = \mathbf{a}_{com}^{s+1} - \mathbf{a}_{com}^1, \quad s=1,2,\dots,N_C-1 \quad (2.21)$$

Sınıf ortak vektörlerinin fark vektörleri, yeniden Gram-Schmidt birimdikleştirme süreci kullanılarak aşağıda gösterildiği gibi birimdikleştirilir.

$$\mathbf{b}_{com}^s \xrightarrow{\text{Gram-Schmidt birimdikleştirme}} \mathbf{w}_s$$

Birimdikleştirilen vektörler, bir matris sütununa yerleştirilir. Sütunlarında \mathbf{w}_s ($s=1,2,\dots,N_C-1$) birimdik vektörlerini içeren bu matris \mathbf{W} sembolüyle gösterildiğinde, sınıf ayırt edici ortak vektörleri Eşitlik 2.22'de gösterildiği gibi belirlenir.

$$\mathbf{\Omega}_c = \mathbf{W}^T \mathbf{a}_k^c, \quad c=1,2,\dots,N_C \quad (2.22)$$

2.2.2 Sınıf içi saçılım üzerinden ayırt edici ortak vektör belirleme

Sınıf içi saçılım kullanarak da ayırt edici ortak vektörler belirlenebilir. Gözlem vektörlerinin sütun vektörleri olduğu kabul edildiğinde sınıf içi saçılım matrisi \mathbf{S}_W , Eşitlik 2.23’de verildiği gibi hesaplanır.

$$\mathbf{S}_W = \sum_{c=1}^{N_c} \sum_{k=1}^M (\mathbf{a}_k^c - \boldsymbol{\mu}_c)(\mathbf{a}_k^c - \boldsymbol{\mu}_c)^T \quad (2.23)$$

Burada $\boldsymbol{\mu}_c$, c ’inci sınıf gözlemlerinin ortalama vektörüdür ve Eşitlik 2.24’de gösterilmiştir.

$$\boldsymbol{\mu}_c = \frac{1}{M} \sum_{k=1}^M \mathbf{a}_k^c, \quad c = 1, 2, \dots, N_c \quad (2.24)$$

Sınıf içi saçılım matrisine özdeğer-özvektör ayrıştırması yapılır. Veri boyutu $d > (M-1)N_c$ koşulunu sağladığı için özdeğerlerin bazıları sıfıra eşit olup diğerleri pozitiftir. Sıfıra eşit olan özdeğerlere karşılık gelen özvektörler farksızlık altuzayını geren taban vektörleri, sıfırdan büyük olan özdeğerlere karşılık gelen özvektörlerse farklılık altuzayını geren taban vektörleridir. Özdeğerler büyükten küçüğe doğru dizildiğinde ve \mathbf{S}_W matrisinin rankının r olduğu kabul edildiğinde, ilk r özdeğere karşılık gelen özvektörler farklılık altuzayının taban vektörleridir. Bu özvektörler $\boldsymbol{\alpha}_l$, $l = 1, 2, \dots, r$ biçiminde gösterildiğinde, sınıf ortak vektörleri Eşitlik 2.25’deki gibi hesaplanır.

$$\mathbf{a}_{com}^c = \mathbf{a}_k^c - \sum_{l=1}^r \langle \mathbf{a}_k^c, \boldsymbol{\alpha}_l \rangle \boldsymbol{\alpha}_l, \quad \forall c \quad (2.25)$$

Bir sonraki aşama, sınıf ortak vektörleri arasındaki saçılımı belirlemektir. Sınıf ortak vektör saçılımı \mathbf{S}_{com} ile gösterilir ve Eşitlik 2.26’da verildiği gibi bulunur.

$$\mathbf{S}_{com} = \sum_{c=1}^{N_C} (\mathbf{a}_{com}^c - \boldsymbol{\mu}_{com})(\mathbf{a}_{com}^c - \boldsymbol{\mu}_{com})^T \quad (2.26)$$

Burada $\boldsymbol{\mu}_{com}$, sınıf ortak vektörlerinin ortalaması olup Eşitlik 2.27’de verilmiştir.

$$\boldsymbol{\mu}_{com} = \frac{1}{N_C} \sum_{c=1}^{N_C} \mathbf{a}_{com}^c \quad (2.27)$$

Son olarak, \mathbf{S}_{com} matrisine özdeğer-özvektör ayrıştırması yapılır. En fazla $(N_C - 1)$ sayıda özdeğer pozitif olup, bu özdeğerlere karşılık gelen özvektörler, sınıf ortak vektörlerinin farklılık bileşenlerini belirleyen altuzayın taban vektörleridir. Her bir özvektör \mathbf{W} matrisinin bir sütununa yerleştirilerek, veri örneklerini en iyi öznitelik uzayına haritalayan \mathbf{W} matrisi oluşturulur. Ayırt edici ortak vektörlere Eşitlik 2.22 kullanılarak hesaplanabilir.

2.2.3 Ayırt edici ortak vektör yöntemi sınıflandırıcısı

Ayırt edici ortak vektör yönteminde sınıflandırma işlemi, hangi sınıfa ait olduğu bilinmeyen test vektörünün (\mathbf{a}_{test}) eğitim aşamasında belirlenen izdüşüm kullanılarak ayırt edici ortak vektör uzayında özniteliğinin çıkarılması ve bu özniteliğin uzaklık bakımından sınıf ayırt edici vektörlerle karşılaştırılıp, en küçük uzaklığı veren sınıfa atanmasıyla gerçekleştirilir. Ayırt edici ortak vektör uzayı taban vektörleri \mathbf{v}_s sembolüyle gösterildiğinde (\mathbf{v}_s vektörleri, Bölüm 2.2.1’deki algoritma kullanıldığında \mathbf{w}_s vektörleridir) sınıflandırıcı basamağı Eşitlik 2.28 – 2.30’da özetlenmiştir.

$$\mathbf{V} = [\mathbf{v}_1 \quad \mathbf{v}_2 \quad \cdots \quad \mathbf{v}_{N_C-1}] \quad (2.28)$$

$$\mathbf{\Omega}_{test} = \mathbf{V}^T \mathbf{a}_{test} \quad (2.29)$$

$$\text{Karar sınıfı: } c^* = \arg \min_{\forall c} \left\{ \left\| \mathbf{\Omega}_c - \mathbf{\Omega}_{test} \right\| \right\} \quad (2.30)$$

BÖLÜM 3

MATRİS TABANLI YÖNTEMLER

Bu bölümde, gri seviyeli görüntü tanıma uygulamalarında kullanılan bazı matris tabanlı yöntemler verilmektedir. Önceki bölümde verilen yöntemler vektör tabanlı yöntemler olup, sayısal görüntü tanıma uygulamalarında kullanılabilmesi için, orijinalde matris biçiminde temsil edilen sayısal görüntü verisinin vektöre dönüştürülme zorunluluğu vardır. Vektöre dönüştürme basamağı, genellikle görüntü matrisinin satırları yan yana eklenerek bir satır vektörü veya sütunları alt alta eklenerek bir sütun vektörü oluşturularak gerçekleştirilir. Satır vektörü veya sütun vektörü tercihi görüntü tanıma sonucunu etkilemez. Vektörel işlemler sonrasında görüntü matrisi, matris vektörleştirilirken yapılan işlemin tam tersi biçiminde matrisleştirilerek geri elde edilebilir.

Ortak vektör yönteminin matris işlemlerine dayandırılarak matris tabanlı yöntem geliştirme çalışmaları ilk olarak Turhal, et al., (2005) tarafından denenmiştir. Çalışmada ortak vektör yöntemiyle veya diğer yöntemlerle karşılaştırma yapılmamıştır ancak Turhal'ın denediği ortak matris yaklaşımı, verinin matris biçimini bozmadan işlem yapar ve yöntemde bulunan ortak matris, görüntü verisini vektöre dönüştürüp ortak vektör yaklaşımı uygulandıktan sonra bulunan ortak vektörü geri matrise çevirmeyle bulunan sonuca denktir. Bu yüzden, ortak matris yaklaşımının tanıma performansı, ortak vektör yaklaşımının tanıma performansına bire bir eşit olup, temelde ortak vektör yönteminden farklı bir yöntem değildir.

Matris tabanlı yöntemler başlığı altında verilen yöntemler, sayısal görüntü verisinin orijinal matris biçimini muhafaza ederek, vektör dönüşümü gerektirmeden veriyi işleyen yöntemlerdir. Yang and Yang (2002) tarafından sunulan IMPCA yöntemi bilinen ilk matris tabanlı yöntemlerdendir. Bu yöntem aynı yazarlar tarafından geliştirilerek (Yang, et al., 2004) kaynakçasında iki-boyutlu PCA veya kısaca 2DPCA isimleriyle verilmiştir. Bu noktadan sonra 2DPCA yöntemine dayalı birçok yöntem

(Chen, et al., 2005; Kong, et al., 2005; Li and Yuan, 2005; Zhang and Zhou, 2005; Jing, et al., 2006) sunulmuş olup, literatürde genellikle iki boyutlu (2D) yöntemler olarak anılırlar ve görüntü tanıma uygulamalarında vektörel yöntemlerle karşılaştırıldıklarında, tanıma başarısı ve hesap yükü bakımından vektörel yöntemlerden çoğu zaman bazı üstünlükleri vardır.

Tez çalışmasında, vektörel yöntemlere olan üstünlüklerine dayanarak, öncelikle iki boyutlu yöntemler yoğun olarak incelenmiştir. Çalışmada, altuzay sınıflandırıcıları üzerine gidilerek iki boyutlu yöntem teorilerine dayandırılan yeni iki boyutlu altuzay sınıflandırıcı yöntemleri (Cevikalp, et al., 2008) önerilmiştir. Daha sonra iki boyutlu yöntemlerin teorileri ortak vektör yaklaşımı üzerinde geliştirilerek Turhal, et al., (2005) kaynakçasındaki ortak matris yaklaşımından teorik anlamda farklı olan yeni bir ortak matris yaklaşımı geliştirilmiştir.

Tez çalışmasında önerilen matris tabanlı yöntemlere geçmeden önce, iki boyutlu yöntemlerin en temeli olan 2DPCA yöntemi birinci alt bölümde verilmiştir. İkinci alt bölümde, önerdiğimiz iki boyutlu altuzay sınıflandırıcıları, üçüncü alt bölümde ise geliştirdiğimiz yeni bir ortak matris yaklaşımı, Ortak Matris-2 ismiyle sunulmuştur.

3.1 İki Boyutlu Temel Bileşen Analizi

İki boyutlu temel bileşen analizi, klasik temel bileşen analizinin vektörel veri ile işlem yapması zorunluluğunu ortadan kaldırma amacını güden bir yaklaşımdır. Yöntemin en önemli özelliği, özellikle görüntü tanıma uygulamaları gibi matris verisi içeren uygulamalarda matris biçimini bozmadan bir kovaryans ölçüsü belirleyip, tüm işlemleri matris formunda gerçekleştirmesidir.

Görüntü tanıma probleminin eğitim verisi kümesinin herbiri M görüntü örneği içeren N_C adet farklı sınıftan oluşan $d_1 \times d_2$ boyutlarında gri seviyeli sayısal görüntüler topluluğu olduğu kabul edilsin. İki boyutlu temel bileşen analizi yönteminde eğitim

verisi kümesini oluşturan $\{\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_{MN_C}\}$; $\mathbf{A}_j \in \mathfrak{R}^{d_1 \times d_2}$ ($j = 1, 2, \dots, MN_C$) görüntüleri orijinal matris gösterimleriyle temsil edilir.

İki boyutlu temel bileşen analizinin amacı, herhangi bir görüntü örneği matrisini (\mathbf{A}), Eşitlik 3.1'de verildiği gibi bir $\mathbf{x} \in \mathfrak{R}^{d_2}$ vektörü üzerine izdüşürüp $\mathbf{y} \in \mathfrak{R}^{d_1}$ öznitelik vektörlerini elde etmektir (Yang, et al., 2004).

$$\mathbf{y} = \mathbf{A}\mathbf{x} \quad (3.1)$$

Buradaki hedef, öznitelik vektörlerinin ayırt edici kapasitesini en üst seviyeye çıkaracak olan izdüşüm vektörlerini belirlemektir. İzdüşüm vektörlerinin, ayırt edici öznitelik çıkarma ölçüsü olarak dönüşüm yapılmış $\mathbf{y} \in \mathfrak{R}^{d_1}$ vektörlerinin toplam saçılımı kullanılır. Bahsedilen toplam saçılım ölçüsü $\mathbf{y} \in \mathfrak{R}^{d_1}$ vektörlerinin kovaryans ölçüsünün izine (trace) eşittir. Matris izi bulma işleci $tr(\cdot)$ biçiminde temsil edildiğinde ve ölçü kriteri olarak kullanılacak olan amaç fonksiyonu $J(\mathbf{x})$ biçiminde gösterilirse amaç fonksiyonu Eşitlik 3.2'deki gibi ifade edilir.

$$J(\mathbf{x}) = tr(\mathbf{S}_x) = tr(\text{cov}(\mathbf{y})) \quad (3.2)$$

Burada verilen \mathbf{S}_x , $\mathbf{A} \in \mathfrak{R}^{d_1 \times d_2}$ matrisinin $\mathbf{x} \in \mathfrak{R}^{d_2}$ üzerine olan izdüşümün kovaryans matrisidir ve aşağıdaki gibi bulunur.

$$\mathbf{S}_x = \text{cov}(\mathbf{x})$$

$$\mathbf{S}_x = E\left[(\mathbf{y} - E[\mathbf{y}])(\mathbf{y} - E[\mathbf{y}])^T\right]$$

$$\mathbf{S}_x = E\left[(\mathbf{A}\mathbf{x} - E[\mathbf{A}\mathbf{x}])(\mathbf{A}\mathbf{x} - E[\mathbf{A}\mathbf{x}])^T\right]$$

$$\mathbf{S}_x = E\left[(\mathbf{A}\mathbf{x} - E[\mathbf{A}]\mathbf{x})(\mathbf{A}\mathbf{x} - E[\mathbf{A}]\mathbf{x})^T\right]$$

$$\mathbf{S}_x = E \left[(\mathbf{A} - E[\mathbf{A}]) \mathbf{x} \mathbf{x}^T (\mathbf{A} - E[\mathbf{A}])^T \right] \quad (3.3)$$

Amaç fonksiyonu da Eşitlik 3.4’de verildiği gibidir.

$$J(\mathbf{x}) = \text{tr}(\mathbf{S}_x) = \mathbf{x}^T E \left[(\mathbf{A} - E[\mathbf{A}])^T (\mathbf{A} - E[\mathbf{A}]) \right] \mathbf{x} = \mathbf{x}^T \mathbf{G}_t \mathbf{x} \quad (3.4)$$

\mathbf{G}_t , görüntü saçılım (kovaryans) matrisi olarak tanımlanır ve eğitim verisi kullanarak Eşitlik 3.5’deki gibi hesaplanabilir.

$$\mathbf{G}_t = E \left[(\mathbf{A} - E[\mathbf{A}])^T (\mathbf{A} - E[\mathbf{A}]) \right] = \frac{1}{MN_C} \sum_{j=1}^{MN_C} (\mathbf{A}_j - \bar{\mathbf{A}})^T (\mathbf{A}_j - \bar{\mathbf{A}}) \quad (3.5)$$

Burada $\bar{\mathbf{A}}$, eğitim verisi görüntü örneklerinin ortalamasıdır, $\bar{\mathbf{A}} = \frac{1}{MN_C} \sum_{j=1}^{MN_C} \mathbf{A}_j$ şeklinde elde edilir.

Bu koşullar altında en iyi $\mathbf{x} \in \mathcal{R}^{d_2}$ izdüşüm vektörü, $J(\mathbf{x}) = \mathbf{x}^T \mathbf{G}_t \mathbf{x}$ amaç fonksiyonunu enbüyükleyen vektördür. Böyle bir $\mathbf{x} \in \mathcal{R}^{d_2}$ vektörü, *optimal izdüşüm eksenini* olarak adlandırılır. Amaç fonksiyonunu enbüyükleyen optimal vektör, görüntü saçılım matrisi \mathbf{G}_t ’nin özvektörlerinin en büyük özdeğere karşılık olanıdır.

Genelde, sadece bir optimal izdüşüm eksenini bulmak iyi tanıma sonuçları için yeterli değildir. Bu nedenle, sadece bir tane değil, birkaç tane; örneğin r tane özvektör optimal izdüşüm eksen grubu olarak seçilebilir. Optimal izdüşüm eksenleri görüntü saçılım matrisinin en büyük ilk r özdeğerine karşılık gelen özvektörlerdir.

$$\left\{ \begin{array}{l} \{ \mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_r \} = \arg \max J(\mathbf{x}) \\ \mathbf{x}_k^T \mathbf{x}_l = 0, \quad k \neq l \quad k, l = 1, 2, \dots, r \end{array} \right\} \quad (3.6)$$

Eşitlik 3.6’da verilen optimal izdüşüm vektörleri kullanılarak en iyi izdüşüm, Eşitlik 3.7’de verildiği gibidir.

$$[\mathbf{y}_1 \ \mathbf{y}_2 \ \cdots \ \mathbf{y}_r] = \mathbf{A} [\mathbf{x}_1 \ \mathbf{x}_2 \ \cdots \ \mathbf{x}_r] \quad (3.7)$$

Optimal izdüşüm vektörleri, \mathbf{X}_{opt} izdüşüm matrisinin sütununa yerleştirildiğinde bu izdüşüm, matris öznelikleri verir. Tüm eğitim veri kümesi görüntü örnekleri için genelleme yapmak gerekirse, $\mathbf{A}_j \in \mathfrak{R}^{d_1 \times d_2}$ ($j=1,2,\dots,MN_C$) görüntü örneklerinin iki boyutlu temel bileşen analizi yöntemi kullanılarak çıkarılan öznelikleri (r adet optimum izdüşüm vektörü kullanıldığı için) matris biçimindedir ve öznelik matrisleri $\mathbf{Y} \in \mathfrak{R}^{d_1 \times r}$ Eşitlik 3.8’de verildiği gibi hesaplanır.

$$\mathbf{Y}_j = \mathbf{A}_j \mathbf{X}_{opt} , \quad j = 1, 2, \dots, MN_C \quad (3.8)$$

İki boyutlu temel bileşen analizi yönteminde öznelikleri sınıflandırmada en yakın komşu sınıflandırıcısı (nearest-neighbor classifier) kullanılır (Yang, et al., 2004). Hangi sınıfa ait olduğu bilinmeyen test görüntüsü $\mathbf{A}_{test} \in \mathfrak{R}^{d_1 \times d_2}$ olsun. Öncelikle, eğitim aşamasında belirlenen optimal izdüşüm matrisi kullanılarak, test vektörü öznelik matrisi Eşitlik 3.9’da verildiği gibi çıkarılır.

$$\mathbf{Y}_{test} = \mathbf{A}_{test} \mathbf{X}_{opt} , \quad \mathbf{Y}_{test} = [\mathbf{y}_1^{test} \ \mathbf{y}_2^{test} \ \cdots \ \mathbf{y}_r^{test}] \quad (3.9)$$

Test matrisinin sütun vektörleri ile eğitim verisinin her bir örneğinin öznelik matrisinin sütun vektörleri arasındaki Euclid uzaklıkları ayrı ayrı hesaplanıp toplanarak uzaklık ölçüsü bulunur. Eşitlik 3.10’da verilen bu uzaklık ölçüsü, (Zuo, et al., 2006) kaynakçasında 2DPCA uzaklık ölçüsü veya Yang uzaklık ölçüsü olarak adlandırılır.

$$d(\mathbf{Y}_{test}, \mathbf{Y}_j) = \sum_{k=1}^r \| \mathbf{y}_k^{test} - \mathbf{y}_k^j \| , \quad \forall j \quad (3.10)$$

Test görüntü matrisi, kendisine yukarıdaki uzaklık ölçüsüne göre en yakın olan öznitelik hangi sınıfın elemanı ise o sınıfa dahil edilir.

3.2 İki Boyutlu Altuzay Sınıflandırıcıları

İki boyutlu altuzay sınıflandırıcıları, CLAFIC (Class-Featuring Information Compression), ortalama çıkarılmış CLAFIC ($CLAFIC - \mu$) ve ALS (Averaged Learning Subspace) yöntemleri olarak bilinen vektörel altuzay sınıflandırıcı yöntemlerine iki boyutlu yaklaşım teknikleri kullanılarak oluşturulan yeni yöntemlerdir (Cevikalp, et al., 2008).

Vektörel altuzay sınıflandırıcı yöntemlerinin temel modeli, veri kümesinin her bir sınıfını, Öklid örnek uzayının doğrusal bir altuzayı biçiminde betimlemekten geçer (Laaksonen, 1997). Alt-uzay yöntemlerinde, vektörel verilerden oluşan bir sınıfın, tüm sınıf verilerinden oluşan toplam örnek uzayının bir altuzayını kapsadığı varsayılır (Oja, 1983). Bu yaklaşımlarda, öncelikle her bir sınıfın kapsadığı altuzayların birimlik taban vektörleri belirlenir. Sınıflandırılacak olan test vektörünün sınıfına karar verme işlemi ise ilgili test vektörünün her bir sınıf altuzayına olan izdüşüm büyüklükleri karşılaştırılarak veya doğrudan test vektörünün her bir sınıf alt-uzayına olan uzaklıkların hesaplanmasıyla gerçekleştirilir.

Altuzay yöntemlerinin görüntü tanıma problemlerinde kullanımı, matris biçimindeki (iki indisli) görüntü verisinin vektör biçimine dönüştürülmesiyle başlar. Görüntü verisindeki değişken sayısı genellikle çok büyük olduğu için altuzay sınıflandırıcı uygulanabilirliği sağlanır. Bu durum avantaj gibi gözükse de, görüntü vektör boyutlarının çok büyük olması, korelasyon veya kovaryans matrislerinin hesaplanmasında ağır bir işlem yükü getirir. İki boyutlu temel bileşen analizi yönteminde hesaplanan kovaryans ölçüsünün hesap yükü ise daha hafiftir. İki boyutlu yöntemlerden yola çıkılarak geliştirilen iki boyutlu altuzay sınıflandırıcıları özetlenerek

ilerleyen alt bölümlerde verilmiştir. Ayrıntılı bilgi için bkz. (Yavuz, et al., 2006; Cevikalp, et al., 2006; Cevikalp, et al., 2008).

3.2.1 İki boyutlu CLAFIC (2D-CLAFIC) yöntemi

Görüntü tanıma probleminin eğitim kümesinin, herbir sınıfında M adet $d_1 \times d_2$ boyutlarında gri seviyeli sayısal görüntü örneği içeren N_C adet farklı sınıftan oluştuğu kabul edilsin ve c 'inci sınıfın k 'inci görüntü örneği $\mathbf{A}_k^c \in \mathfrak{R}^{d_1 \times d_2}$ ($k = 1, 2, \dots, M$) sembolüyle gösterilsin. Sınıfların korelasyon matrisleri, görüntü örneklerinin matris biçimleri bozulmadan Eşitlik 3.11'de verildiği gibi tanımlanır.

$$\mathbf{R}_{2d}^{(c)} = \frac{1}{M} \sum_{k=1}^M (\mathbf{A}_k^c)^T \mathbf{A}_k^c, \quad c = 1, 2, \dots, N_C \quad (3.11)$$

İki boyutlu korelasyon matrisi $\mathbf{R}_{2d}^{(c)} \in \mathfrak{R}^{d_2 \times d_2}$, vektörel korelasyon matrisinden ($\mathbf{R}^{(c)} \in \mathfrak{R}^{d \times d}$, $d = d_1 d_2$) çok daha küçük boyutlardadır. $\mathbf{R}_{2d}^{(c)}$ yarı kesin (semi definite) artı (positive) bir matris olduğu için özdeğerleri sıfırdan büyük veya sıfıra eşittir. Öncelikle özdeğerler büyükten küçüğe doğru dizilir, daha sonra en büyük ilk r_c özdeğere karşılık gelen özvektörler dönüşüm vektörü olarak seçilir. Bu dönüşüm vektörleri, c 'inci sınıf altuzayını geren birimdik vektörlerdir. Bu birimdik vektörler \mathbf{w}_l^c sembolüyle gösterilip, bir matrisin sütunu olarak yazıldığında, $\mathbf{W}_c = [\mathbf{w}_1^c \ \mathbf{w}_2^c \ \dots \ \mathbf{w}_{r_c}^c]$, dönüşüm matrisini oluşturur. İki boyutlu yaklaşım geleneğine göre, herhangi bir test görüntü matrisi $\mathbf{A}_{test} \in \mathfrak{R}^{d_1 \times d_2}$, seçilen özvektörler üzerine izdüşürülerek görüntü matrisinin öznitelik vektörleri elde edilir (Yang, et al., 2004). Bu izdüşüm Eşitlik 3.12'de gösterilmiştir.

$$\mathbf{Y}_{test}^c = \mathbf{A}_{test} \mathbf{W}_c, \quad c = 1, 2, \dots, C \quad (3.12)$$

Elde edilen $\mathbf{Y}_{test}^c \in \mathfrak{R}^{d_1 \times r_c}$ matrisi, test görüntü matrisinin öznitelik matrisidir. İki-boyutlu CLAFIC yönteminde sınıflandırma için özniteliklerin *Frobenius* norm (Moon and Stirling, 2000) kareleri Eşitlik 3.13’de verildiği gibi tüm sınıflar için hesaplanır. Buna göre test görüntüsü, en büyük uzaklık değerini veren sınıfa atanır.

$$\|\mathbf{Y}_{test}^c\|^2 = \sum_{j=1}^{d_1} \sum_{k=1}^{r_c} |y_{jk}|^2 = \text{trace}(\mathbf{Y}_{test}^c \mathbf{Y}_{test}^{cT}), \quad c=1,2,\dots,N_C \quad (3.13)$$

2D-CLAFIC sınıflandırıcısı, Eşitlik 3.14’de özetlenmiştir.

$$g(\mathbf{A}_{test}) = \arg \max_{c=1,2,\dots,N_C} (\|\mathbf{Y}_{test}^c\|^2) \quad (3.14)$$

3.2.2 Ortalama çıkarılmış iki boyutlu CLAFIC (2D-CLAFIC- μ) yöntemi

Vektörel CLAFIC yönteminin bir değişik uyarlaması olan CLAFIC- μ yöntemi, sınıflandırma işleminde sınıf ortalamalarını kullanır (Laaksonen, 1997). Bu yaklaşımda, test verisi sınıf ortamları civarında ortalanır. Bu yüzden, sınıf alt-uzay taban vektörleri sınıf korelasyon ölçüsü yerine sınıf kovaryans ölçüsünden elde edilir. Ayrıca, sınıflandırma karar sınırı, diğer yöntemden farklı olarak, ortalanmış test vektörünün altuzaylara olan uzaklıklarının en küçüğü alınarak belirlenir.

Vektörel CLAFIC yönteminden benzetim yapılarak, sınıf ortalamaları kullanıldığında iki boyutlu CLAFIC yönteminin değişik bir uyarlaması olan iki boyutlu CLAFIC- μ yöntemi geliştirilebilir. Sınıfların kovaryans matrisleri, görüntü verisinin matris biçimini bozmadan Eşitlik 3.15’de gösterildiği gibi hesaplanır.

$$\Sigma_{2d}^{(c)} = \frac{1}{M} \sum_{k=1}^M (\mathbf{A}_k^c - \bar{\mathbf{A}}^c)^T (\mathbf{A}_k^c - \bar{\mathbf{A}}^c), \quad c=1,2,\dots,N_C \quad (3.15)$$

Burada $\bar{\mathbf{A}}^c = \frac{1}{M} \sum_{k=1}^M \mathbf{A}_k^c$, c 'inci sınıf görüntülerinin ortalama matrisidir. Kovaryans matrisinin büyükten küçüğe doğru dizilmiş ilk r_c özdeğerlerine karşılık gelen özvektörler, sınıf altuzayının birimdik taban vektörleridir. Bu birimdik vektörler, \mathbf{W}_c izdüşüm matrisinin sütunlarını oluşturur. Sınıflandırılacak test görüntüsü \mathbf{A}_{test} için uzaklık ölçüsü, görüntü verisinin matris biçimi muhafaza edilerek ve matris norm hesaplamalarında *Frobenius* norm kareleri kullanılarak Eşitlik 3.16'daki gibi bulunur.

$$f_c = \|\mathbf{A}_{test} - \bar{\mathbf{A}}^c\|^2 - \left\| (\mathbf{A}_{test} - \bar{\mathbf{A}}^c) \mathbf{W}_c \right\|^2, \quad c = 1, 2, \dots, N_C \quad (3.16)$$

2D-CLAFIC- μ yönteminde, test görüntüsü, tüm sınıflar için f_c uzaklık değerlerinin en küçük olduğu sınıfa atanır. Sonuç olarak 2D-CLAFIC- μ sınıflandırıcısı Eşitlik 3.17'deki gibidir.

$$g(\mathbf{A}_{test}) = \arg \min_{c=1,2,\dots,N_C} \left(\|\mathbf{A}_{test} - \bar{\mathbf{A}}^c\|^2 - \left\| (\mathbf{A}_{test} - \bar{\mathbf{A}}^c) \mathbf{W}_c \right\|^2 \right) \quad (3.17)$$

3.2.3 İki boyutlu ALS (2DALS) yöntemi

İki boyutlu ALS yöntemi, 2D-CLAFIC yönteminin iteratif öğrenmeli biçimi gibi düşünülebilir. Bu yöntemde, yanlış sınıflandırma oranını düşürecek biçimde altuzayı geren taban vektörleri iteratif bir algoritma kullanılarak güncellenir. Yapılan güncelleme, klasik ALS yönteminde olduğu gibi veri örneklerinin normalize edilmemiş korelasyon matrisleri üzerinedir. İki boyutlu ALS yönteminde, iki boyutlu yöntemlerdeki korelasyon ölçüsü kullanılır. Bu durum, veri uzayı boyutu büyük olduğu durumlarda, klasik ALS yöntemine göre ciddi bir hesap kolaylığı sağlamaktadır. İki boyutlu ALS algoritması aşağıda özetlenmiştir.

İki boyutlu ALS algoritması:

- *Basamak 1:* Döngü sayısını sıfırla ($epoch = 0$), en çok döngü sayısını sabitle ($maxepoch = T$) ve ilk eğitim kümesi sınıf korelasyon matrislerini Eşitlik 3.18'deki gibi hesapla.

$$\mathbf{S}_0^{(c)} = \sum_{k=1}^M (\mathbf{A}_k^c)^T \mathbf{A}_k^c, \quad c = 1, 2, \dots, N_C \quad (3.18)$$

Eşitlik 3.18'de verilen korelasyon matrisi normalize edilmemiştir. Bunu kullanarak, 2D-CLAFIC yöntemindeki gibi altuzay taban vektörlerini belirle.

- *Basamak 2:* Eşitlik 3.14'de verilen 2D-CLAFIC sınıflandırıcısını kullanarak eğitim kümesi veri örneklerini sınıflandır ve doğru tanıma oranını belirle (DTO_{epoch}). Hatalı sınıflandırılan örnekleri iki altkümede topla: $epoch$ 'uncu döngüde $\Delta_{epoch}^{(c)}$ kümesi eğitim verisi sınıflandırmasında aslında c 'inci sınıfa ait olduğu halde yanlışlıkla bir başka sınıfa atanan veri örneklerini içeren küme; $\Gamma_{epoch}^{(c)}$ kümesi ise c 'inci sınıfa ait olmadığı halde yanlışlıkla c 'inci sınıf elemanı olarak karar verilen veri örneklerini içeren kümeler olsun. Buna göre korelasyon matrisini Eşitlik 3.19'da verildiği gibi güncelle.

$$\mathbf{S}_{epoch+1}^{(c)} = \mathbf{S}_{epoch}^{(c)} + \alpha \sum_{\mathbf{A}_j^c \in \Delta_{epoch}^{(c)}} (\mathbf{A}_j^c)^T \mathbf{A}_j^c - \beta \sum_{\mathbf{A}_l^c \in \Gamma_{epoch}^{(c)}} (\mathbf{A}_l^c)^T \mathbf{A}_l^c, \quad c = 1, 2, \dots, N_C \quad (3.19)$$

Burada, α ve β pozitif katsayıları, hata düzeltme keskinliğini belirleyen parametreler olup klasik ALS yöntemindeki gibi uygun biçimde ayarlanmalıdır.

- *Basamak 3:* Güncellenen korelasyon matrislerinden altuzay tabanlarını belirleyip doğru tanıma oranlarını çıkar ve döngü sayısını bir arttırarak 2'nci basamağa dön; $DTO_{epoch+1} < DTO_{epoch}$ veya $epoch = maxepoch$ ise dur.

3.3 Yeni Bir Ortak Matris Yaklaşımı: Ortak Matris-2

Bu bölümde, ortak vektör yaklaşımını temel alarak, iki boyutlu yöntem prensipleriyle uyumlu olarak geliştirdiğimiz ve veri matris biçimini koruyarak işlem yapan yeni bir ortak matris yaklaşımı, Ortak Matris-2 (OM2) ismiyle sunulmuştur. Sunulan yöntem, Turhal, et al., (2005) kaynakçasında ortak matris yaklaşımı olarak adlandırılan yöntemden farklıdır.

Ortak matris yaklaşımlarının amacı, her bir veri sınıfı için, sınıftaki görüntü verilerinin ortak özelliklerini taşıyan bir ortak matris belirlemektir. Belirlenen ortak matris, sınıf temsil edicisi niteliğini taşır. Bu özelliği sebebiyle ortak matris, görüntü tanıma probleminin sınıflandırıcı karar kuralında kullanılabilir.

Görüntü tanıma probleminin eğitim kümesinin, herbiri M_c örnek içeren ve $\omega^{(1)}, \omega^{(2)}, \dots, \omega^{(N_c)}$ biçiminde gösterilen N_c adet sınıftan oluştuğu kabul edilsin. $\mathbf{A}_k^c \in \mathfrak{R}^{d_1 \times d_2}$ ise $d_1 \times d_2$ boyutlarında, c 'inci sınıfın k 'inci görüntü örneği olsun. Sınıf ortak matrisi \mathbf{A}_{com}^c , Eşitlik 3.20'de verilen ifadeyi sağlamalıdır.

$$\mathbf{A}_k^c = \mathbf{A}_{com}^c + \mathbf{A}_{k,dif}^c, \quad c = 1, 2, \dots, N_c; \quad k = 1, 2, \dots, M_c \quad (3.20)$$

Burada $\mathbf{A}_{k,dif}^c$, c 'inci sınıfın k 'inci görüntü örneğinin ortak matristen farkını ifade eden fark matrisidir. Ortak Vektör yaklaşımında, bu noktada izlenen kabullenme, ilgili örneğin ortak bileşeni ve fark bileşeninin, birbirine dik olan iki alt-uzay üzerine ilgili örneğin izdüşümünün alınarak belirlenmesidir (Gulmezoglu, et al., 1999; 2001). Ortak Matris yaklaşımında da aynı kabullenme üzerinden yola çıkılmıştır. Aradaki fark, Ortak Matris yönteminde iki boyutlu yöntem prensiplerine göre matrislerin altuzaya izdüşürülmesidir.

Sadece $k = 1, 2, \dots, M$ örnek içeren bir sınıf için durumu betimlemek gerekirse, $\mathbf{A}_k = \mathbf{A}_{com} + \mathbf{A}_{k,dif}$ eşitliğini sağlayan \mathbf{A}_{com} ortak matrisi, \mathbf{A}_k görüntü matrisinin

farksızlık altuzayı olarak tanımlanacak bir altuzay üzerine izdüştürülmesiyle; $\mathbf{A}_{k,dif}$ fark matrisi ise, \mathbf{A}_k görüntü matrisinin *fark altuzayı* olarak tanımlanacak bir altuzay üzerine izdüştürülmesiyle bulunacaktır. Buradaki farksızlık altuzayı ve fark altuzayı, tümleyen uzaylar olmalıdır. Farksızlık altuzayına olan izdüştüm matrisi $\mathbf{P}_{indif} \in \mathfrak{R}^{d_1 \times d_1}$, fark altuzayına olan izdüştüm matrisi de $\mathbf{P}_{dif} \in \mathfrak{R}^{d_1 \times d_1}$ olarak tanımlanırsa, fark ve farksızlık altuzaylarına hesaplanan izdüştümler aşağıda 3.21-3.26 eşitliklerinde verilen koşulları sağlamalıdır.

$$\mathbf{P}_{indif}^T \mathbf{A}_k = \mathbf{A}_{com} \quad (3.21)$$

$$\mathbf{P}_{indif}^T \mathbf{A}_{com} = \mathbf{A}_{com} \quad (3.22)$$

$$\mathbf{P}_{indif}^T \mathbf{A}_{k,dif} = \mathbf{0} \quad (3.23)$$

$$\mathbf{P}_{dif}^T \mathbf{A}_k = \mathbf{A}_{k,dif} \quad (3.24)$$

$$\mathbf{P}_{dif}^T \mathbf{A}_{k,dif} = \mathbf{A}_{k,dif} \quad (3.25)$$

$$\mathbf{P}_{dif}^T \mathbf{A}_{com} = \mathbf{0} \quad (3.26)$$

Bu koşulları sağlayan \mathbf{P}_{dif} ve \mathbf{P}_{indif} izdüştüm matrisleri, birimcil (unitary) matris olmalıdır. Herhangi bir \mathbf{A}_k görüntü matrisinin \mathbf{P}_{indif} üzerinden dönüşümü, doğrudan \mathbf{A}_{com} ortak matrisini verir. \mathbf{A}_k görüntü matrisinin \mathbf{P}_{dif} üzerinden dönüşümü de $\mathbf{A}_{k,dif}$ fark matrisini verir. \mathbf{P}_{dif} ve \mathbf{P}_{indif} arasındaki ilişki Eşitlik 3.27’de verilmiştir.

$$\mathbf{P}_{indif} = \mathbf{I} - \mathbf{P}_{dif} \quad \text{veya} \quad \mathbf{P}_{dif} = \mathbf{I} - \mathbf{P}_{indif} \quad (3.27)$$

$\mathbf{B}_j \in \mathfrak{R}^{d_1 \times d_2}$ ($j = 1, 2, \dots, M-1$) matrisleri de herhangi bir \mathbf{A}_k örneğinin, diğer tüm örneklerden çıkarılmasıyla bulunan fark matrisleri olsun. Örneğin, \mathbf{A}_1 diğerlerinden çıkarıldığında bulunan fark matrisleri Eşitlik 3.28’de verilmiştir

$$\begin{aligned}
\mathbf{B}_1 &= \mathbf{A}_2 - \mathbf{A}_1 \\
\mathbf{B}_2 &= \mathbf{A}_3 - \mathbf{A}_1 \\
&\vdots \\
\mathbf{B}_{M-1} &= \mathbf{A}_M - \mathbf{A}_1
\end{aligned} \tag{3.28}$$

$\mathbf{B}_j^T \mathbf{v} = \mathbf{0}$ değerini sağlayan $\mathbf{v} \in \mathfrak{R}^{d_1}$ vektörleri, \mathbf{B}_j matrisinin sol sıfır uzayını (left nullspace) oluşturan vektörlerdir. \mathbf{B}_j ($j=1,2,\dots,M-1$) matrislerinin sol sıfır uzayının olabilmesi için aynı \mathbf{v} vektörü 3.29 eşitliğini sağlamalıdır.

$$\mathbf{B}_1^T \mathbf{v} = \mathbf{0} \text{ ve } \mathbf{B}_2^T \mathbf{v} = \mathbf{0} \text{ ve } \dots \text{ ve } \mathbf{B}_{M-1}^T \mathbf{v} = \mathbf{0} \tag{3.29}$$

Eşitlik 3.29, aşağıdaki biçimde de yazılabilir:

$$[\mathbf{B}_1 \ \mathbf{B}_2 \ \dots \ \mathbf{B}_{M-1}]^T \mathbf{v} = [\mathbf{0} \ \mathbf{0} \ \mathbf{0} \ \mathbf{0}] \tag{3.30}$$

Bu durumda, Eşitlik 3.30'daki sol matris Eşitlik 3.31'de verildiği gibi tanımlanırsa, Eşitlik 3.30, Eşitlik 3.32'deki gibi sadeleşir.

$$[\mathbf{B}_1 \ \mathbf{B}_2 \ \dots \ \mathbf{B}_{M-1}] = \mathbf{B} \tag{3.31}$$

$$\mathbf{B}^T \mathbf{v} = \mathbf{0} \tag{3.32}$$

Buradaki $\mathbf{B} \in \mathfrak{R}^{d_1 \times d_2(M-1)}$ matrisi, $(M-1)$ adet fark matrisinin yan yana dizilerek oluşturduğu bileşik fark matrisidir. Altuzay teknikleriyle fark altuzayı ve farksızlık altuzayı belirlemenin yolu \mathbf{B} 'nin sol sıfır (left null) altuzayı ve erim (range) altuzayının belirlenebilmesinden geçer. \mathbf{B} 'nin erim altuzayı her koşulda belirlenebilir ancak sıfır (null) altuzayı yalnızca özel koşullar altında mevcuttur (Moon and Stirling, 2000). Birbirine dik ve birbirinin tümleyeni olan erim ve sol sıfır uzayları, ortak vektör yaklaşımındaki fark ve farksızlık altuzaylarına karşılık gelir. Bileşik fark matrisinin sol sıfır uzayı varsa, bu altuzay farksızlık altuzayı olduğundan, herhangi bir sınıfa ait

görüntü örneklerinin bu altuzaya olan izdüşümü, ilgili sınıfın ortak matrisini verir. \mathbf{B} matrisinin sol sıfır uzayının tanımlanabilmesi durumu, $\mathbf{B}^T \in \mathfrak{R}^{d_2(M-1) \times d_1}$ matris boyutlarının $d_2(M-1) < d_1$ koşulunu sağladığı zaman mümkündür. Bu durum, veri sayısı M , (d_1/d_2+1) oranından daha küçükse ortaya çıkar ve OM2 yaklaşımının yetersiz veri durumu olarak adlandırılır. Bunun tam tersi, $d_2(M-1) \geq d_1$ koşulunu sağlayan durum ise OM2 yaklaşımının yeterli veri durumudur. Yeterli ve yetersiz veri durumu için OM2 yöntemi ayrı alt başlıklarda verilmiştir.

3.3.1 Yetersiz veri durumu için Ortak Matris-2 yöntemi

Yetersiz veri durumunda, birbirinin tümleyeni olan fark ve farksızlık altuzaylarını belirlemek mümkündür. Dolayısıyla ortak matris, bileşik fark matrisinin erim uzayını veya sol sıfır uzayını kullanarak bulunabilir. Bir matrisin erim uzayının taban vektörleri, bu matrisinin sütunlarını oluşturan vektörlerin birimdikleştirilmesi sonucunda belirlenir (Moon and Stirling, 2000). \mathbf{B} matrisinin erim uzayı, \mathbf{A} matrislerinin fark altuzayına karşılık geldiğinden, bu altuzaya olan izdüşüm matrisi \mathbf{P}_{dif} , birimdikleştirme sonucunda bulunan birimdik taban vektörleri ile belirlenebilir. İzdüşüm matrisi bulunduktan sonra, tümleyen uzayın izdüşüm matrisi Eşitlik 3.27'den; ortak matris ise doğrudan Eşitlik 3.21'den belirlenebilir. Başka bir biçimde, fark altuzayı izdüşümü kullanılarak fark bileşeni Eşitlik 3.24'den; ortak bileşen de Eşitlik 3.20'den hesaplanabilir. Bileşik fark matrisinin erim uzayı kullanılarak ortak matris belirleme süreci aşağıda özetlenmiştir.

Yetersiz veri durumunda B'nin erim uzayı kullanılarak ortak matris belirleme algoritması

- *Basamak 1:* Bileşik fark matrisi sütun vektörlerini birimdikleştir:

$$\mathbf{W} = \text{Gram-Schmidt birimdikleştirme} (\mathbf{B})$$

$$\mathbf{W} = \begin{bmatrix} \mathbf{w}_1 & \mathbf{w}_2 & \cdots & \mathbf{w}_{d_2(M-1)} \end{bmatrix} \quad (\mathbf{w}_l : \text{fark altuzayı birimdik taban vektörleri})$$

- *Basamak 2*: Fark altuzayına izdüşüm matrisini belirle:

$$\mathbf{P}_{dif} = \sum_{l=1}^{d_2(M-1)} \mathbf{w}_l \mathbf{w}_l^T \quad (\mathbf{P}_{dif} : d_1 \times d_1 \text{ 'lik simetrik izdüşüm matrisi})$$

- *Basamak 3*: Ortak matrisi (a) veya (b) şıklarındaki eşitliklerle hesapla:

- *3a*:

$$\mathbf{A}_{k,dif} = \mathbf{P}_{dif}^T \mathbf{A}_k = \mathbf{P}_{dif} \mathbf{A}_k$$

$$\mathbf{A}_{com} = \mathbf{A}_k - \mathbf{A}_{k,dif}$$

- *3b*:

$$\mathbf{P}_{indif} = \mathbf{I} - \mathbf{P}_{dif}$$

$$\mathbf{A}_{com} = \mathbf{P}_{indif}^T \mathbf{A}_k = \mathbf{P}_{indif} \mathbf{A}_k$$

Bileşik fark matrisinin sol sıfır uzayı kullanılarak da doğrudan farksızlık altuzayına olan izdüşüm matrisi \mathbf{P}_{indif} belirlenebilir. Farksızlık altuzayına olan izdüşüm kullanılarak ortak matris belirleme algoritması da aşağıda özetlenmiştir.

Yetersiz veri durumunda B'nin sol sıfır uzayı kullanılarak ortak matris belirleme algoritması

- *Basamak 1*: Bileşik fark matrisi sol sıfır uzayı taban vektörlerini aşağıdaki eşitlikten bul:

$$\mathbf{B}^T \mathbf{v}_h = \mathbf{0} \quad \forall h$$

$$\mathbf{V} = [\mathbf{v}_1 \quad \mathbf{v}_2 \quad \cdots \quad \mathbf{v}_{d_1-d_2(M-1)}] \quad (\mathbf{v}_h : \text{farksızlık altuzayı birimlik taban vektörleri})$$

- *Basamak 2*: Farksızlık altuzayına izdüşüm matrisini belirle:

$$\mathbf{P}_{indif} = \sum_{h=1}^{d_1-d_2(M-1)} \mathbf{v}_h \mathbf{v}_h^T \quad (\mathbf{P}_{indif} : d_1 \times d_1 \text{ 'lik simetrik izdüşüm matrisi})$$

- *Basamak 3*: Ortak matrisi (a) veya (b) şıklarındaki eşitliklerle hesapla:

- *3a*:

$$\mathbf{A}_{com} = \mathbf{P}_{indif}^T \mathbf{A}_k = \mathbf{P}_{indif} \mathbf{A}_k$$

o 3b:

$$\mathbf{P}_{dif} = \mathbf{I} - \mathbf{P}_{indif}$$

$$\mathbf{A}_{k,dif} = \mathbf{P}_{dif}^T \mathbf{A}_k = \mathbf{P}_{dif} \mathbf{A}_k$$

$$\mathbf{A}_{com} = \mathbf{A}_k - \mathbf{A}_{k,dif}$$

Bileşik fark matrisi \mathbf{B} kullanılmadan da ortak matris belirlenebilir. Bunun yolu, görüntü verisi örneklerinin kovaryans matrisini kullanmaktır. Kovaryans matrisinin özdeğer-özvektör ayrıştırması sonucu, fark ve farksızlık altuzayları belirlemek mümkündür. Kovaryans matrisi Σ , görüntünün matris biçimini bozmadan, iki boyutlu yaklaşım tekniklerine uyumlu olarak Eşitlik 3.33'de verildiği gibi hesaplanır.

$$\Sigma = \frac{1}{M} \sum_{k=1}^M (\mathbf{A}_k - \bar{\mathbf{A}})(\mathbf{A}_k - \bar{\mathbf{A}})^T \quad (3.33)$$

Burada $\bar{\mathbf{A}} = \frac{1}{M} \sum_{k=1}^M \mathbf{A}_k$, sınıf görüntülerinin ortalama matrisidir. Kovaryans matrisi yarı kesin pozitif bir matris olduğu için özdeğerleri sıfırdan büyük veya sıfıra eşittir. Veri durumu yetersiz veri durumuna uygun olduğu için, en az bir tane sıfır olan özdeğer mevcuttur. Sıfıra eşit olan özdeğerlere karşılık gelen özvektörler sol sıfır uzayını temsil eden taban vektörleri, sıfırdan büyük olan değerlere karşılık gelen özvektörlerse erim uzayını temsil eden taban vektörleridir. Sol sıfır uzayı farksızlık altuzayına, erim uzayı da fark altuzayına karşılık geldiği için, fark ve farksızlık altuzayları belirlenmiş olur. Bu noktadan sonra izlenecek yol, izdüşümlerin hesap edilerek ortak matrisin önceki bölümlerdeki basamaklara benzer biçimde belirlenmesidir. Kovaryans ölçüsü kullanarak ortak matris belirleme algoritması aşağıda özetlenmiştir.

Yetersiz veri durumunda kovaryans ölçüsü kullanılarak ortak matris belirleme algoritması

- *Basamak 1:* Eşitlik 3.33'de verilen veri kovaryans matrisine özdeğer-özvektör ayrıştırması yap ve sıfırdan büyük olan özdeğerlere karşılık gelen özvektörleri seç:

$$\begin{array}{ccccccc} \lambda_1 & \geq & \lambda_2 & \geq & \dots & \geq & \lambda_r & > & 0 \\ \downarrow & & \downarrow & & & & \downarrow & & \\ [\mathbf{w}_1 & \mathbf{w}_2 & \dots & \mathbf{w}_r] & = & \mathbf{W} \end{array}$$

(\mathbf{w}_l : fark altuzayı birimdik taban vektörleridir)

- *Basamak 2:* Fark altuzayına olan izdüşüm matrisini belirle:

$$\mathbf{P}_{dif} = \sum_{l=1}^r \mathbf{w}_l \mathbf{w}_l^T \quad (\mathbf{P}_{dif} : d_1 \times d_1 \text{ 'lik simetrik izdüşüm matrisi})$$

- *Basamak 3:* Ortak matrisi (a) veya (b) şıklarındaki eşitliklerle hesapla:

○ 3a:

$$\mathbf{A}_{k,dif} = \mathbf{P}_{dif}^T \mathbf{A}_k = \mathbf{P}_{dif} \mathbf{A}_k$$

$$\mathbf{A}_{com} = \mathbf{A}_k - \mathbf{A}_{k,dif}$$

○ 3b:

$$\mathbf{P}_{indif} = \mathbf{I} - \mathbf{P}_{dif}$$

$$\mathbf{A}_{com} = \mathbf{P}_{indif}^T \mathbf{A}_k = \mathbf{P}_{indif} \mathbf{A}_k$$

3.3.2 Yeterli veri durumu için Ortak Matris-2 yöntemi

Yeterli veri durumunda $d_2(M-1) \geq d_1$ olduğu için bileşik fark matrisinin sol sıfır uzayı yoktur. Sol sıfır uzayı olmadığı için erim uzayının tümleyeni olan bir altuzay bulunamaz. Bu durumda, altuzay teknikleriyle ortak matris belirlenemez. Ortak matris belirlemek için kovaryans yöntemlerinden yararlanılmalıdır. İki-boyutlu yöntemlerdeki kovaryans matrisinin özdeğer-özvektör ayrıştırmasından fark altuzayını temsil edeceği kabul edilen yaklaşık bir fark altuzayı belirlenebilir. Seçilen özvektör sayısı

belirlenen fark altuzayı, farklı seçimler için farklı sonucu verecektir. Seçilmesi gereken özvektör sayısı ise yöntemin bir giriş parametresi olarak kabul edilir ve en iyi tanıma performansını veren parametre iteratif deneylerle belirlenir.

Yeterli veri durumu için ortak matris belirleme işlemi sınıf kovaryans matrisinin hesaplanmasıyla başlar. Sınıf kovaryans matrisi Eşitlik 3.33'de verildiği gibidir. Kovaryans matrisine özdeğer-özvektör ayrıştırması yapılır. Yetersiz veri durumundan farklı olarak, kovaryans matrisinin tüm özdeğerleri sıfırdan büyüktür ve yeterli veri durumunda sıfır olan özdeğer yoktur. Büyükten küçüğe doğru dizilmiş ilk r özdeğere karşılık gelen özvektörler, yaklaşık olarak fark altuzayını belirleyen birimlik taban vektörleridir. Kullanıcı seçimine bağlı olan r sayısı, fark altuzayını belirleyen tek faktördür. Buna göre belirlenen fark altuzayı üzerine izdüşürülen örneklerin fark bileşenleri kendilerinden çıkarılarak farksızlık matrisleri bulunur. Yetersiz veri durumunda her bir örneğin farksızlık matrisi ortak matrise eşitken, yeterli veri durumunda her bir örneğin farksızlık matrisi farklıdır. Bu yüzden ortak matris, en küçük hata karesi kestiricisine göre tüm örneklerin farksızlık bileşenlerinin ortalamasına eşit olarak alınır. Benzer biçimde, örneklerin ortalamasının farksızlık bileşeni de aynı ortak matrisi verir. Yeterli veri durumunda OM2 algoritmasının basamakları aşağıda özetlenmiştir.

Yeterli veri durumunda ortak matris belirleme algoritması

- *Basamak 1:* Eşitlik 3.33'de verilen veri kovaryans matrisine özdeğer-özvektör ayrıştırması yap ve özdeğerleri büyükten küçüğe sırala. Özdeğerlere karşılık gelen özvektörleri de aynı sıraya dizerek ilk r özvektörü seç.

$$\begin{array}{ccccccc} \lambda_1 & \geq & \lambda_2 & \geq & \dots & \geq & \lambda_r & \geq & \dots & \geq & \lambda_{d_1} \\ \downarrow & & \downarrow & & & & \downarrow & & & & \\ [\mathbf{w}_1 & \mathbf{w}_2 & \dots & \mathbf{w}_r] & = & \mathbf{W} \end{array}$$

(\mathbf{w}_l : yaklaşık olarak seçilen fark altuzayı birimlik taban vektörleridir)

- *Basamak 2:* Fark altuzayına olan izdüşüm matrisini belirle.

$$\mathbf{P}_{dif} = \sum_{l=1}^r \mathbf{w}_l \mathbf{w}_l^T \quad (\mathbf{P}_{dif} : d_1 \times d_1 \text{ 'lik simetrik izdüşüm matrisi})$$

- *Basamak 3:* Ortak matrisi fark altuzayı üzerinden (a) veya (b) şıklarında verildiği gibi hesapla.

- *3a:*

$$\mathbf{A}_{k,dif} = \mathbf{P}_{dif}^T \mathbf{A}_k = \mathbf{P}_{dif} \mathbf{A}_k$$

$$\mathbf{A}_{com} = \frac{1}{M} \sum_{k=1}^M (\mathbf{A}_k - \mathbf{A}_{k,dif}) = \frac{1}{M} \sum_{k=1}^M \mathbf{A}_k - \mathbf{P}_{dif} \left(\frac{1}{M} \sum_{k=1}^M \mathbf{A}_k \right)$$

- *3b:*

$$\mathbf{P}_{indif} = \mathbf{I} - \mathbf{P}_{dif}$$

$$\mathbf{A}_{com} = \frac{1}{M} \sum_{k=1}^M (\mathbf{P}_{indif} \mathbf{A}_k) = \mathbf{P}_{indif} \left(\frac{1}{M} \sum_{k=1}^M \mathbf{A}_k \right)$$

3.3.3 Ortak Matris-2 yöntemi sınıflandırıcı kararı

OM2 yöntemi kullanılarak bulunan sınıf ortak matrisleri, sınıfların ayırt edici öznelikleri gibi kullanılabilir. Ortak vektör yaklaşımından farkı, bu özneliklerin matris biçiminde olmasıdır. Ortak matris sınıflandırıcı kararı için, ortak vektör yaklaşımına benzeyen bir sınıflandırıcı kullanılabilir. Eğitim aşamasında, tüm $\mathbf{A}_k^c \in \mathfrak{R}^{d_1 \times d_2}$ örnekleri kullanılarak her bir sınıfın ortak matrisi \mathbf{A}_{com}^c ve ortak matris hesaplanması için gereken izdüşüm matrisi \mathbf{P}_{indif}^c veya \mathbf{P}_{dif}^c ($c = 1, 2, \dots, N_C$) belirlenir. Bu işlem, eğitim verisinin yeterli veri veya yetersiz veri durumuna uygun olmasına göre yukarıda anlatılan yöntemlerden ve izdüşümlerden herhangi birisi kullanılarak yapılabilir. Test aşaması, hangi sınıfa ait olduğu bilinmeyen $\mathbf{A}_{test} \in \mathfrak{R}^{d_1 \times d_2}$ test görüntüsünün sınıflandırma aşamasıdır. Bu aşamada test görüntüsü, eğitim aşamasında bulunan tüm \mathbf{P}_{indif}^c izdüşümleri üzerine izdüşürülür. Test görüntüsünün c 'inci sınıfa olan izdüşümü $\mathbf{A}_{test,com}^c$ ile ilgili sınıf ortak matrisi arasındaki uzaklıklar karşılaştırılır ve

ortak matrisleri en yakın olan sınıf kararı verilir. Farklılık altuzayı üzerinden $\mathbf{A}_{test,com}^c$ Eşitlik 3.34’de gösterildiği gibi bulunur.

$$\mathbf{A}_{test,com}^c = \mathbf{P}_{indif}^c \mathbf{A}_{test}^c, \quad c = 1, 2, \dots, N_c \quad (3.34)$$

OM2 yaklaşımında öznitelikler matris biçiminde olduğu için, uzaklık ölçüsü olarak matris uzaklık ölçüleri kullanılabilir. (Zuo, et al., 2006) kaynakçasında, iki-boyutlu yöntemler için önerilen birleştirilmiş matris uzaklık ölçüsünün (AMD) iyi sonuçlar verdiği belirtilmiştir. Buna dayanarak OM2 yönteminde AMD uzaklık ölçüsünün kullanılması önerilmiştir. $\mathbf{A}_{test,com}^c$ ile \mathbf{A}_{com}^c matrisleri arasındaki AMD uzaklık ölçüsü Eşitlik 3.35’de verilmiştir.

$$d_{AMD}(\mathbf{A}_{test,com}^c, \mathbf{A}_{com}^c) = \left(\sum_{l=1}^{d_2} \left(\sum_{o=1}^{d_1} (a_{ol}^{(test,com)} - a_{ol}^{(com)})^2 \right)^{\frac{p}{2}} \right)^{\frac{1}{p}}, \quad p > 0 \quad (3.35)$$

Burada, d_{AMD} AMD uzaklık ölçüsü değeri olup, $p \in (0, 2]$ ise kullanıcı seçimine bırakılan sınıflandırıcı parametresidir. $p = 2$ değeri için AMD uzaklık ölçüsü matris Frobenius ölçüsüne denktir. $p = 1$ değeri ise 2DPCA yöntemindeki uzaklık ölçüsüne eşittir. Zuo, et al., (2006) ise, $p < 1$ koşulunu sağlayacak biçimde seçilecek olan p değerlerinin, tanıma oranlarını, diğer ölçülerden daha iyi çıkaracağını ifade etmektedir. OM2 yönteminde sınıf atamasına, test ortak matrisi ile sınıf ortak matrisleri arasındaki en küçük AMD uzaklığını veren sınıf olarak karar verilir. Sınıf kararı Eşitlik 3.36’da özetlenmiştir.

$$\mathbf{Karar\ sınıfi\ } c : d_{AMD}(\mathbf{A}_{test,com}^c, \mathbf{A}_{com}^c) = \arg \min_{\forall c} \{ d_{AMD}(\mathbf{A}_{test,com}^c, \mathbf{A}_{com}^c) \} \quad (3.36)$$

BÖLÜM 4

YÜKSEK DERECEDEN TENSÖR TABANLI YÖNTEMLER

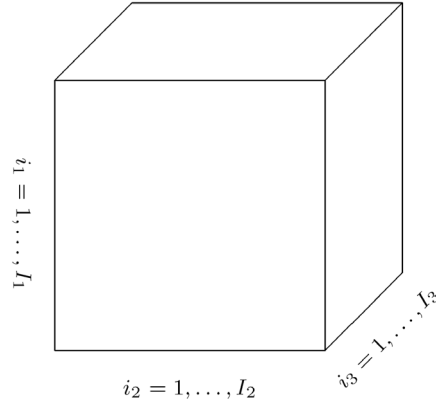
Son zamanlarda, sinyal işleme problemlerinin büyük bir kısmı ikiden daha fazla sayıda indislerle temsil edilen veri biçimleriyle uğraşmak durumunda kalmıştır. Literatürde, tek indisle betimlenen vektörler, iki indisle ifade edilen matrislere ilave olarak, ikiden fazla sayıda indisle temsil edilen nicelikler, yüksek dereceli tensörler olarak adlandırılır. Yüksek dereceden tensörler kullanarak işlem yapabilmek için bilinen vektör ve matris cebiri yetersiz kalmaktadır. Vektör kavramları üzerine kurulu olan lineer cebirin vektör uzayları tanımlarından genişletilerek oluşturulan, yüksek dereceden tensör temellerini içeren ve tensör uzayları tanımlarının yapıldığı cebir, multilineer cebir olarak adlandırılır (Vasilescu and Terzopoulos, 2002; Lathauwer, et al., 2004).

Bu bölümde, yüksek dereceden tensör biçiminde verilere uygulanabilen tensör tabanlı yöntemler ele alınmıştır. Yöntemlere adaptasyon sağlayabilmek için ilk altbaşlıkta, multilineer cebirin temel kavramlarına değinilmiştir. Daha sonra klasik temel bileşen analizi yönteminin multilineer genişlemesi olan multilineer temel bileşen analizi (MPCA) ve multilineer diskriminant analizi (MDA) yöntemleri özetlenmiştir.

4.1 Multilineer Cebir Temelleri

Bu bölümde, multilineer cebirin temel tanım ve gösterimleri verilmiştir. Bölümde verilen tanımlar (Lathauwer, et al., 2000; Vasilescu and Terzopoulos, 2002; Lathauwer, et al., 2004; Bader and Kolda, 2006) kaynaklarından derlenmiştir. Çalışmada yer alan değişken gösterimleri ise Bader and Kolda, (2006) kaynakçasındaki gibidir. Skaleraler küçük yatık harflerle (a, b, \dots), vektörler küçük koyu harflerle ($\mathbf{a}, \mathbf{b}, \dots$), matrisler büyük koyu harflerle ($\mathbf{A}, \mathbf{B}, \dots$) ve daha yüksek dereceden tensörler de büyük süslü harflerle ($\mathcal{A}, \mathcal{B}, \dots$) temsil edilmektedir.

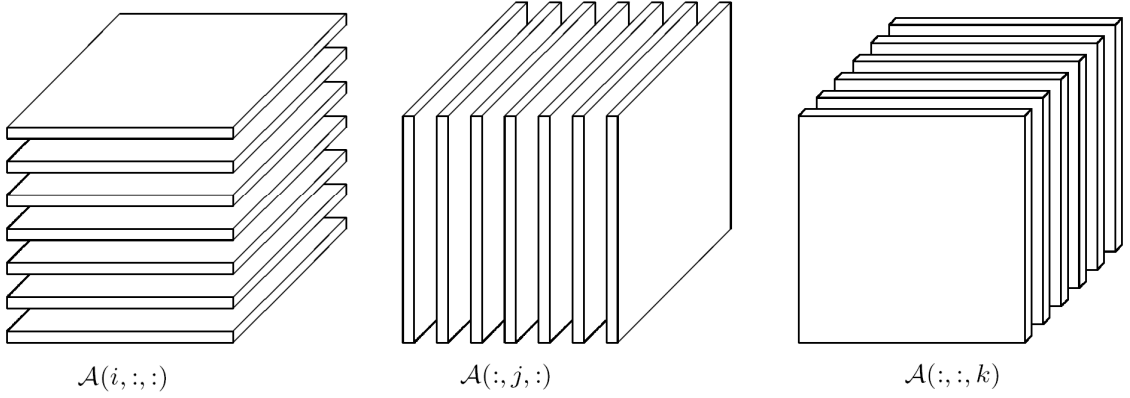
Bir tensör, çok boyutlu (multidimensional) veya N -yönlü (N -way) dizi olarak tanımlanır. Skaler sıfırıncı derece, vektör birinci derece, matris ise ikinci dereceden bir tensördür. $\mathcal{A} \in \mathfrak{R}^{I_1 \times I_2 \times \dots \times I_N}$ biçiminde tanımlanan bir \mathcal{A} tensörünün derecesi N 'dir. Bu tensörün bir elemanı $\mathcal{A}_{i_1 \dots i_n \dots i_N}$ veya $a_{i_1 \dots i_n \dots i_N}$ ile gösterilir ($1 \leq i_n \leq I_n$). \mathcal{A} 'nın n 'inci modu (veya yönü), I_n boyutudur. Şekil 4.1'de üçüncü dereceden bir tensör örneği gösterilmiştir.



Şekil 4.1 Üçüncü dereceden bir tensör örneği

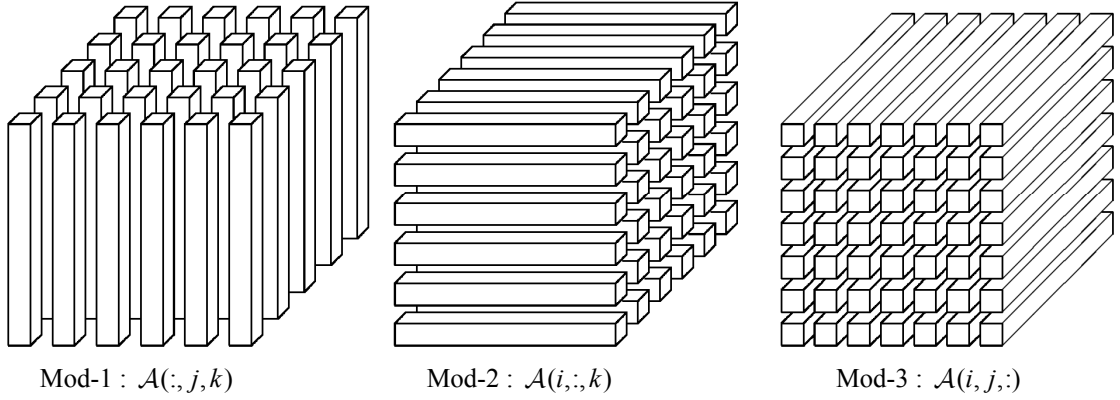
4.1.1 Tensör – matris – vektör dönüşümleri

Matrisleri bazı özel durumlarda vektöre çevirmek mümkün olduğu gibi, yüksek dereceden tensörleri de matrise veya vektöre dönüştürmek mümkündür. Üçüncü dereceden bir \mathcal{A} tensörünün bir indisi sabit tutulup diğer iki indisi değiştirildiğinde matrisler oluşur. Bu işlem, üç boyutlu gösterimde üçüncü dereceden tensör küpünü dilimlemeye benzer. Oluşan matris dilimleri, sabit tutulan indise göre Şekil 4.2'de verildiği gibidir.



Şekil 4.2 Üçüncü dereceden bir \mathcal{A} tensörünün matris dilimleri

Benzer biçimde, tensörleri vektöre dönüştürmek için bir indisi değiştirip diğer indisleri sabit tutmak gerekir. Değiştirilen indis n 'inci indis olursa, bulunan vektörler, ilgili tensörün mod- n vektörleri olarak adlandırılır. Üçüncü dereceden \mathcal{A} tensörünün mod- n vektörleri Şekil 4.3'de verilmiştir.



Şekil 4.3 Üçüncü dereceden bir \mathcal{A} tensörünün mod- n vektörleri

Bazı özel durumlarda tensörleri matris veya vektör biçiminde temsil etmek gerekebilir. Tensörleri matrise dönüştürme işlemi *düzleştirme* (flattening) ya da *matrisleştirme* (matricizing) olarak adlandırılır. Matrisleştirmenin ilk aşaması ilgili tensörü matris dilimlerine bölmekten geçer. Son aşama ise bölmelenen matrisleri matris blokları olarak kabul edip, uygun biçimde daha büyük bir matris içine yerleştirmektir. Matrisleştirmenin de farklı modları mevcuttur. N 'inci dereceden bir tensör her bir yönüne doğru (N farklı modda) matrisleştirilebilir.

Benzer biçimde, tensörler vektör biçimine de dönüştürülebilirler. Matrislerin vektöre dönüştürülmesi mod-1 düzeyinde sütun vektörlerini alt alta koyarak vektörleştirme, mod-2 düzeyinde satır vektörlerini yan yana koyarak vektörleştirmeye karşılık gelir. Matrisleştirilen veya vektörleştirilen tensörleri orijinal formunda yazmak gerektiğinde, hangi mod üzerinden düzleştirildiği bilinmelidir. Buna göre ilgili elemanlar yerlerine koyularak tensöre geri dönüştürülebilir. Örnek olarak, $\mathcal{A} \in \mathfrak{R}^{I_1 \times I_2 \times I_3}$ olan bir tensörün matrisleri $\mathbf{A}_{(1)} \in \mathfrak{R}^{I_1 \times I_2 I_3}$, $\mathbf{A}_{(2)} \in \mathfrak{R}^{I_2 \times I_1 I_3}$, $\mathbf{A}_{(3)} \in \mathfrak{R}^{I_3 \times I_1 I_2}$ olup, vektörü de $\text{vec}(\mathcal{A}) \in \mathfrak{R}^{I_1 I_2 I_3}$ 'dür.

4.1.2 Tensör – matris çarpımı

Bir tensör ile matris çarpımı işlemi için tensörün hangi modunun matrisin sütunlarıyla çarpılacağı özelleştirilmesi gerekir. Bunun için tensor matris çarpımı mod- n çarpımı olarak tanımlanır (Lathauwer, et al., 2000). \mathcal{A} tensörü $I_1 \times I_2 \times \dots \times I_N$ boyutlarında bir tensörse ve \mathbf{U} matrisi de $J_n \times I_n$ 'lik bir matrisse, \mathcal{A} tensörü ile \mathbf{U} matrisinin mod- n ($1 \leq i_n \leq I_N$) çarpımı $\mathcal{A} \times_n \mathbf{U}$ biçiminde gösterilir ve $I_1 \times I_2 \times \dots \times I_{n-1} \times J_n \times I_{n+1} \times \dots \times I_N$ boyutlarında olur. Bu mod- n çarpımı, aslında tam olarak \mathbf{U} matrisiyle \mathcal{A} tensörünün uygun modda düzleştirilmişinin matris çarpımına eşittir. Mod- n tensör çarpımı aşağıdaki iki özelliği sağlar (Vasilescu and Terzopoulos, 2002):

1. $\mathcal{A} \in \mathfrak{R}^{I_1 \times \dots \times I_n \times \dots \times I_m \times \dots \times I_N}$, $\mathbf{U} \in \mathfrak{R}^{J_m \times I_m}$ ve $\mathbf{V} \in \mathfrak{R}^{J_n \times I_n}$ iken Eşitlik 4.1 sağlanır.

$$\begin{aligned} \mathcal{A} \times_m \mathbf{U} \times_n \mathbf{V} &= (\mathcal{A} \times_m \mathbf{U}) \times_n \mathbf{V} \\ &= (\mathcal{A} \times_n \mathbf{V}) \times_m \mathbf{U} \\ &= \mathcal{A} \times_n \mathbf{V} \times_m \mathbf{U} \end{aligned} \quad (4.1)$$

2. $\mathcal{A} \in \mathfrak{R}^{I_1 \times \dots \times I_n \times \dots \times I_N}$, $\mathbf{U} \in \mathfrak{R}^{J_n \times I_n}$ ve $\mathbf{V} \in \mathfrak{R}^{K_n \times J_n}$ iken Eşitlik 4.2 sağlanır.

$$(\mathcal{A} \times_n \mathbf{U}) \times_n \mathbf{V} = \mathcal{A} \times_n (\mathbf{V}\mathbf{U}) \quad (4.2)$$

4.1.3 Skaler çarpım

İki tensörün skaler çarpımı $\mathcal{A}, \mathcal{B} \in \mathfrak{R}^{I_1 \times I_2 \times \dots \times I_N}$ Eşitlik 4.3’de verildiği gibi tanımlanır (Lathauwer, et al., 2004):

$$\langle \mathcal{A}, \mathcal{B} \rangle \stackrel{def}{=} \sum_{i_1} \sum_{i_2} \dots \sum_{i_N} a_{i_1 i_2 \dots i_N} b_{i_1 i_2 \dots i_N} \quad (4.3)$$

Skaler çarpımları sıfıra eşit olan iki tensör birbirine diktir. Bir tensörün Frobenius normu da Eşitlik 4.4’deki gibi tanımlanır (Lathauwer, et al., 2004).

$$\|\mathcal{A}\| \stackrel{def}{=} \sqrt{\langle \mathcal{A}, \mathcal{A} \rangle} \quad (4.4)$$

4.1.4 Tensör ayrıştırmaları

Bir matrisin tekil değer ayrıştırması SVD (singular value decomposition), lineer cebirde çok önemli anlamlar ifade eden bir ayrıştırma. SVD, matrisin satır uzayı ve sütun uzayını dikleştirerek, ilgili matrisi rankı 1 olan en az sayıdaki matrislerin toplamı biçiminde ifade eden bir ayrıştırma. SVD, Eşitlik 4.5’deki gibi gösterilebilir.

$$\mathbf{A} = (\mathbf{u}_1 \circ \mathbf{v}_1) + (\mathbf{u}_2 \circ \mathbf{v}_2) + \dots + (\mathbf{u}_r \circ \mathbf{v}_r) \quad (4.5)$$

Burada, $i = 1, 2, \dots, r$ için $\mathbf{u}_i \in \mathfrak{R}^I$, $\mathbf{v}_i \in \mathfrak{R}^J$, $\mathbf{A} \in \mathfrak{R}^{I \times J}$ boyutlarında olup “ \circ ” işlemcisi de dış çarpım (outer product) işlemcisidir.

Matrisin ikinci dereceden bir tensör olduğu düşünülürken, yukarıdaki tekil değer ayrıştırması üç veya daha fazla dereceden tensörler için genelleştirilebilir. Yüksek dereceden tensörler için yapılan tekil değer ayrıştırması HOSVD (Higher Order Singular Value Decomposition) veya mod- n SVD olarak adlandırılır (Lathauwer, et al., 2000; Vasilescu and Terzopoulos, 2002). HOSVD'nin amacı, SVD de olduğu gibi, ilgili tensörü en az sayıda rank-1 tensörlerin toplamı olarak yazabilmek veya bir başka deyişle herhangi bir tensörü rank-1 tensörlerine ayrıştırılabilmektir. Örneğin, üçüncü dereceden $\mathcal{A} \in \mathfrak{R}^{I \times J \times K}$ tensörü, Eşitlik 4.6'daki gibi $\mathbf{u}_i \in \mathfrak{R}^I$, $\mathbf{v}_i \in \mathfrak{R}^J$, $\mathbf{w}_i \in \mathfrak{R}^K$ vektörlerinin dış çarpımlarından oluşan r adet rank-1 tensörün toplamı biçiminde yazılmalıdır.

$$\mathcal{A} = (\mathbf{u}_1 \circ \mathbf{v}_1 \circ \mathbf{w}_1) + (\mathbf{u}_2 \circ \mathbf{v}_2 \circ \mathbf{w}_2) + \dots + (\mathbf{u}_r \circ \mathbf{v}_r \circ \mathbf{w}_r) \quad (4.6)$$

Uygulamaların birçoğunda iki önemli tensör ayrıştırması yöntemi kullanılmaktadır. Bu yöntemlerden birincisi, iki farklı yazarın birbirinden bağımsız ve eş zamanlı olarak farklı isimlerle tanıttığı bir yöntemdir. Yazarlardan biri bu yöntemi CANDECOMP (CANonical DECOMPosition) yöntemi olarak adlandırırken, diğeri PARAFAC (PARAllel FACTors) yöntemi olarak adlandırmaktadır. İkinci ayrıştırma yöntemi ise yazarının ismiyle adlandırılan TUCKER modelidir (Lathauwer, et al., 2000; Lathauwer, et al., 2004).

$\mathcal{A} \in \mathfrak{R}^{I \times J \times K}$ tensörü için CP modeli Eşitlik 4.7'de gösterilen biçimdedir.

$$\mathcal{A} = \sum_{i=1}^R (\mathbf{u}_i \circ \mathbf{v}_i \circ \mathbf{w}_i) \quad (4.7)$$

Burada $i = 1, 2, \dots, R$ için $\mathbf{u}_i \in \mathfrak{R}^I$, $\mathbf{v}_i \in \mathfrak{R}^J$ ve $\mathbf{w}_i \in \mathfrak{R}^K$ 'dir. Vektörler \mathbf{u}_i , \mathbf{v}_i ve \mathbf{w}_i için diklik gibi hiçbir kısıt söz konusu değildir.

$\mathcal{A} \in \mathfrak{R}^{I \times J \times K}$ tensörü için TUCKER modeli ise Eşitlik 4.8'de verildiği gibidir.

$$\mathcal{A} = \sum_{i=1}^{R_1} \sum_{j=1}^{R_2} \sum_{k=1}^{R_3} \sigma_{ijk} (\mathbf{u}_i \circ \mathbf{v}_j \circ \mathbf{w}_k) \quad (4.8)$$

Burada $R_1 \leq I$, $R_2 \leq J$, $R_3 \leq K$ ve tüm i, j, k için $\mathbf{u}_i \in \mathfrak{R}^{R_1}$, $\mathbf{v}_j \in \mathfrak{R}^{R_2}$, $\mathbf{w}_k \in \mathfrak{R}^{R_3}$ 'dır. $\mathcal{S} = (\sigma_{ijk})$ tensörü *çekirdek tensör* (core tensor) olarak adlandırılır. Çekirdek tensör \mathcal{S} , orijinal tensör \mathcal{A} ile aynı boyutlarda olmak zorunda değildir. Aslında CP ayrıştırması, TUCKER ayrıştırmasının özel bir durumudur.

TUCKER ayrıştırmasında da vektörlerin dikliği kısıtı mevcut değildir. Ancak vektörler dikse, \mathbf{u}_i , \mathbf{v}_i ve \mathbf{w}_i sırasıyla \mathbf{U} , \mathbf{V} ve \mathbf{W} matrislerinin sütunlarını oluşturacak biçimde yerleştirildiğinde TUCKER modeli yüksek dereceli tekil değer ayrıştırması HOSVD olarak adlandırılır. HOSVD'nin mod- n tensör çarpımları cinsinden ifadesi Eşitlik 4.9'daki gibidir.

$$\mathcal{A} = \mathcal{S} \times_1 \mathbf{U} \times_2 \mathbf{V} \times_3 \mathbf{W} \quad (4.9)$$

Burada $\mathcal{A}, \mathcal{S} \in \mathfrak{R}^{I \times J \times K}$, $\mathbf{U} \in \mathfrak{R}^{I \times I}$, $\mathbf{V} \in \mathfrak{R}^{J \times J}$ ve $\mathbf{W} \in \mathfrak{R}^{K \times K}$ 'dır. HOSVD, her bir moda göre \mathcal{A} tensörünün düzleştirilmesinden elde edilen matrislerin tekil değer ayrıştırması alınarak hesaplanır. Bu nedenle HOSVD her zaman yapılabilir. SVD ile HOSVD arasındaki en önemli fark HOSVDnin tensör rankı bilgisini diyagonal bir çekirdek tensörü olmadıkça açığa çıkarmamasıdır. Hangi koşullarda diyagonal bir çekirdek tensörü elde edildiği ise açık değildir (Vasilescu and Terzopoulos, 2002).

4.2 Multilineer Temel Bileşen Analizi Yöntemi

Multilineer temel bileşen analizi (multilinear principal component analysis – MPCA), klasik temel bileşen analizinin multilineer cebir yöntemleriyle genişletilmiş biçimidir. Klasik temel bileşen analizinde giriş verisi vektör iken, MPCA yönteminin

giriş verisi N 'inci dereceden tensör olabilir. MPCA, temel bileşen analizi yönteminin yüksek dereceden verilere uygulanabilen doğal bir genişlemesidir (Lu, et al., 2006).

PCA yöntemi, izdüştürülen vektörlerin varyansını enbüyükleme amacını taşıyan bir boyut indirgeme yöntemidir. Yöntem, büyük boyutlu uzayda tanımlanan giriş vektörlerini çok daha küçük boyutlu uzayda tanımlanan öznelik vektörlerine haritalar. Gri seviyeli sayısal görüntü tanıma uygulamasında eğitim verisinin, önceki bölümlerde gösterildiği gibi $\mathbf{A}_k^c \in \mathfrak{R}^{d_1 \times d_2}$ biçiminde temsil edilen herbiri M_c örnek içeren N_C adet sınıftan oluştuğu kabul edilsin. Klasik temel bileşen analizinde, görüntü matrisleri öncelikle vektöre dönüştürülür ve bu vektörlerden bir eğitim verisi oluşturulur. Temel bileşenler, eğitim verisinin kovaryans matrisine özdeğer-özvektör ayrıştırması yaparak veya sıfır ortalamalı eğitim verisine doğrudan tekil değer ayrıştırması yaparak belirlenir. Eğitim veri örnekleri, $\mathbf{a}_k^c \in \mathfrak{R}^{d_1 d_2}$ biçiminde gösterilirse, eğitim veri ortalaması Eşitlik 4.10'da gösterilir.

$$\bar{\mathbf{a}} = \frac{1}{N_C} \sum_{c=1}^{N_C} \left(\frac{1}{M_c} \sum_{k=1}^{M_c} \mathbf{a}_k^c \right) \quad (4.10)$$

Sıfır ortalamalı eğitim verisi matrisi \mathbf{X} ile temsil edildiğinde Eşitlik 4.11'de verildiği gibi oluşturulur.

$$\mathbf{X} = \begin{bmatrix} \mathbf{a}_1^1 - \bar{\mathbf{a}} & \mathbf{a}_2^1 - \bar{\mathbf{a}} & \cdots & \mathbf{a}_{M_1}^1 - \bar{\mathbf{a}} & \mathbf{a}_1^2 - \bar{\mathbf{a}} & \cdots & \mathbf{a}_{M_{N_C}}^{N_C} - \bar{\mathbf{a}} \end{bmatrix} \quad (4.11)$$

Bu durumda klasik PCA, tensörel gösterimle Eşitlik 4.12'deki gibi ifade edilir.

$$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T = \mathbf{U}^{(1)}\mathbf{S}\mathbf{U}^{(2)} = \mathbf{S} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \quad (4.12)$$

Burada, \mathbf{U} , \mathbf{X} 'in tekil değer ayrıştırması sonucu bulunan sol matris olup $\mathbf{U}^{(1)}$ biçiminde; \mathbf{V} , \mathbf{X} 'in tekil değer ayrıştırması sonucu bulunan sağ matris olup $\mathbf{V}^T = \mathbf{U}^{(2)}$ biçiminde gösterilmiştir. \mathbf{S} ise, herbir vektöre karşılık gelen tekil değerlerin oluşturduğu diyagonal

matristir. Ortalama çıkarılmış eğitim verisinin temel bileşenleri, $\mathbf{U}^{(1)}$ sol matrisinin sütun vektörleridir.

PCA tabanlı altuzay analizinde, $\mathbf{U}^{(1)}$ matrisinin, büyükten küçüğe dizilmiş tekil değerlerden sadece ilk birkaç tanesine karşılık gelen sütun vektörü kullanılır, yani, $\mathbf{U}^{(1)} \in \mathfrak{R}^{d \times d}$ ise, ilk r tekil değere karşılık gelen $\tilde{\mathbf{U}}^{(1)} \in \mathfrak{R}^{d \times r}$ matrisi izdüşüm matrisi olarak kullanılır. Öznitelik vektörü $\tilde{\mathbf{y}}$, ortalama çıkarılmış bir test vektörünün ($\mathbf{x} = \mathbf{a}_{test} - \bar{\mathbf{a}}$), $\tilde{\mathbf{U}}^{(1)}$ üzerine izdüşürülmesiyle Eşitlik 4.13'de verildiği gibi bulunur.

$$\tilde{\mathbf{y}} = \tilde{\mathbf{U}}^{(1)T} \mathbf{x} = \mathbf{x} \times_1 \tilde{\mathbf{U}}^{(1)T} \quad (4.13)$$

Sınıflandırma işlemi, test vektörlerinin öznitelikleri ile eğitim vektörlerinin öznitelikleri arasındaki uzaklıklara bakılarak gerçekleştirilir. En yakın uzaklığa sahip olan vektörler aynı sınıfa atanır.

MPCA ise klasik PCA'nın multilineer duruma genişletilmesiyle oluşur. Eğitim veri tensörünün, genel gösterimde N 'inci dereceden bir tensör olduğu kabul edilsin. Bu tensör, her bir örneği $(N-1)$ 'inci dereceden olan verilerin her birinin bir üst boyuta yerleştirilmesiyle oluşturulur. Yine, ortalama çıkarılmış verilerin eğitim tensörünü oluşturduğu kabul edilirse, sıfır ortalamalı eğitim verisi elde edilir. Sıfır ortalamalı eğitim tensörü \mathcal{X} sembolüyle gösterildiğinde, HOSVD kullanarak, N 'inci dereceden ayrıştırma Eşitlik 4.14'de verildiği gibi yapılır (Lu, et al., 2006).

$$\mathcal{X} = \mathcal{S} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \dots \times_N \mathbf{U}^{(N)} \quad (4.14)$$

\mathcal{X} 'nin mod- n tekil değerleri $\sigma_i^{(n)}$ ile gösterilir ve bu değerler \mathcal{X} tensörünün mod- n düzleştirilmesinin tekil değer ayrıştırması sonucunda bulunur. Elde edilen tekil değerler \mathcal{S} tensörünün mod- n düzleştirilmiş biçimine karşılık gelir. Klasik temel bileşen analizinde taban vektörleri sol matrisin sütun vektörlerine karşılık geldiği halde MPCA de taban tensörlerinin ne ifade ettiği açık biçimde görülemez. PCA den

genelleme yapıldığında, taban tensörlerinin $(N-1)$ 'inci dereceden olduğu söylenebilir. Tıpkı PCA yöntemindeki gibi, izdüşüm almada, mod- n SVD (HOSVD) gerçekleştirilerek belirlenen $\mathbf{U}^{(n)} \in \mathfrak{R}^{d_n \times d_n}$ matrislerinin ilk r_n ($r_n \leq d_n$) sütun vektörleri alınarak indirgenmiş $\tilde{\mathbf{U}}^{(n)} \in \mathfrak{R}^{d_n \times r_n}$ izdüşüm matrisleri kullanılır. İzdüşüm alınırken, yine PCA yönteminden yapılan genellemeye göre $(N-1)$ tane izdüşüm matrisi kullanılmalıdır. Bu durumda toplam taban vektörü sayısı $\prod_{n=1}^{N-1} r_n$ olur. İzdüşüm tensörü $\tilde{\mathcal{P}}_{-N}$ biçiminde gösterilir ve Eşitlik 4.15'de verildiği gibidir.

$$\tilde{\mathcal{P}}_{-N} = \times_1 \mathbf{U}^{(1)T} \times_2 \mathbf{U}^{(2)T} \dots \times_{N-1} \mathbf{U}^{(N-1)T} \quad (4.15)$$

MPCA yönteminde, sınıflandırılacak test verisi, $(N-1)$ 'inci derecedendir. Ortalama çıkarılmış test verisi $\mathcal{T} = \mathcal{A}_{test} - \bar{\mathcal{A}}$ ile gösterilirse, öznitelik tensörü $\tilde{\mathcal{Y}}$, \mathcal{Z} 'nin $\tilde{\mathcal{P}}_{-N}$ üzerine izdüşürülmesiyle aşağıdaki gibi bulunur.

$$\tilde{\mathcal{Y}} = \mathcal{T} \times \tilde{\mathcal{P}}_{-N} = \mathcal{T} \times_1 \mathbf{U}^{(1)T} \times_2 \mathbf{U}^{(2)T} \dots \times_{N-1} \mathbf{U}^{(N-1)T} \quad (4.16)$$

Sınıflandırma aşamasında, test veri örneğinin öznitelik tensörü, eğitim veri örneğinin öznitelik tensörlerinden hangisine daha yakınsa, ilgili test tensörü o sınıfa atanır. Lu and Venetsanopoulos'un (2006) çalışmasında uzaklık ölçüsü olarak Eşitlik 4.4'deki tensör Frobenius norm ölçüsü kullanılmıştır.

4.3 Multilinear Diskriminant Analizi Yöntemi

Multilinear diskriminant analizi (MDA), önceki bölümde verilen MPCA yönteminden farklı olarak, altuzayların ayırt edici özelliklerini arttırmayı hedefleyen bir ayırt edici yöntemdir. Yüz tanıma problemleri için Yan, et al., (2007) tarafından sunulan yöntem, bu bölümde özet olarak verilmiştir.

Yöntemde, öncelikle ayırt edici tensör kriteri sunulmuştur. Kendi içlerinde ilintili, birbiri arasında mümkün olduğunca ilintisiz olan ayırt edilebilir altuzaylar, m -mod optimizasyon¹ olarak tanıtilan yeni bir iteratif yöntemle öğrenilir. MDA yönteminin üç temel özelliği bulunmaktadır. Birincisi, çoklu ilintilendirilen altuzaylar birlikte kullanılarak farklı sınıfları başarılı bir şekilde ayırt edebilir. İkincisi, yüksek dereceden tensör içeren sınıflandırma problemlerinde büyük boyutsallık (curse of dimensionality) ve küçük örnek yığını (small sample size) problemleri olarak bilinen sorunlarla başedebilir. Üçüncüsü de eğitim aşamasında hesap kolaylığı sağlayan düşük boyuttaki özniteliklerle çalışır.

MDA yönteminde tensör diskriminant kriteri Eşitlik 4.17’de verildiği gibidir. Bu diskriminant kriterinin amacı tensör metrikleri kullanarak, sınıf içi saçılımı enküçükleyip, aynı zamanda sınıflar arası saçılımı enbüyükleyen ilintili izdüşüm matrislerini araştırmaktır.

$$\mathbf{U}_m^* |_{m=1}^N = \arg \max_{\mathbf{U}_m |_{m=1}^N} \left(\frac{\sum_c N_c \left\| \bar{\mathcal{A}}^c \times_1 \mathbf{U}_1 \cdots \times_N \mathbf{U}_N - \bar{\mathcal{A}} \times_1 \mathbf{U}_1 \cdots \times_N \mathbf{U}_N \right\|^2}{\sum_i \left\| \mathcal{A}_i \times_1 \mathbf{U}_1 \cdots \times_N \mathbf{U}_N - \bar{\mathcal{A}}^{c_i} \times_1 \mathbf{U}_1 \cdots \times_N \mathbf{U}_N \right\|^2} \right) \quad (4.17)$$

Burada $\bar{\mathcal{A}}^c$, c 'inci sınıfa ait olan örneklerin ortalama tensörü, $\bar{\mathcal{A}}$, tüm örneklerin ortalama tensörü, N_c , c 'inci sınıftaki örnek sayısıdır. Eşitlik 4.17’de verilen kriter, yüksek dereceden doğrusal olmayan bir optimizasyon problemine eşit olup bir kapalı çözüm bulmak oldukça zordur. MDA yönteminde optimum çözüm, m -mod optimizasyon adı verilen ve tensörün sadece bir modu üzerinden amaç fonksiyonunu optimize etmeye çalışan iteratif bir algoritma kullanarak bulunur.

¹ Yöntem orijinal kaynakçada k -mod optimizasyon olarak adlandırılmış olmasına rağmen k sembolü önceki bölümlerde veri örneğini temsil etmek için kullanılmış olduğundan m sembolüyle değiştirilmiştir.

Sadece tensörün bir modu üzerinden optimizasyon Eşitlik 4.18'de gösterildiği gibidir ve özel bir diskriminant problemi gibi düşünülebilir. Veri örneği tensörleri, m 'inci modda matrisleştirilir ve herbir matrisin sütun vektörü aynı sınıfa ait olan bir nesneymiş gibi düşünülebilir. Bu durumda yeni optimizasyon problemi Eşitlik 4.19'daki gibi ifade edilir.

$$(\mathbf{U}_m^*) = \arg \max_{\mathbf{U}_m} \frac{\sum_c N_C \|\bar{\mathcal{A}}^c \times_k \mathbf{U}_k - \bar{\mathcal{A}} \times_k \mathbf{U}_k\|^2}{\sum_i \|\mathcal{A}_i \times_k \mathbf{U}_k - \bar{\mathcal{A}}^{c_i} \times_k \mathbf{U}_k\|^2} \quad (4.18)$$

$$\mathbf{U}_m^* = \arg \max_{\mathbf{U}_m} \frac{Tr(\mathbf{U}_m^T \mathbf{S}_B \mathbf{U}_m)}{Tr(\mathbf{U}_m^T \mathbf{S}_W \mathbf{U}_m)} \quad (4.19)$$

Burada \mathbf{S}_B sınıflar arası saçılım, \mathbf{S}_W sınıf içi saçılım olup, sırasıyla aşağıdaki eşitliklerdeki gibidir.

$$\mathbf{S}_B = \prod_{j=1}^{\prod_{o \neq m} d_o} \left(\sum_{c=1}^{N_C} N_C (\bar{\mathbf{A}}_{m,j}^c - \bar{\mathbf{A}}_{m,j}) (\bar{\mathbf{A}}_{m,j}^c - \bar{\mathbf{A}}_{m,j})^T \right) \quad (4.20)$$

$$\mathbf{S}_W = \prod_{j=1}^{\prod_{o \neq m} d_o} \left(\sum_{i=1}^{N_T} (\mathbf{A}_{m,j}^i - \bar{\mathbf{A}}_{m,j}^{c_i}) (\mathbf{A}_{m,j}^i - \bar{\mathbf{A}}_{m,j}^{c_i})^T \right) \quad (4.21)$$

Eşitliklerde verilen sembollerden örneğin $\mathbf{A}_{m,j}^i$, i 'inci veri örneği tensörü $\mathcal{A}_i \in \mathfrak{R}^{d_1 \times d_2 \times \dots \times d_N}$ 'nin mod- m matrisleştirilerek elde edilen \mathbf{A}_m^i matrisinin j 'inci sütun vektörünü temsil eder. Benzer biçimde, $\bar{\mathbf{A}}_{m,j}^c$, c 'inci sınıf ortalama tensörü $\bar{\mathcal{A}}^c$ 'nin mod- m matrisleştirilmiş biçiminin j 'inci sütun vektörü; $\bar{\mathbf{A}}_{m,j}$ tüm veri örneklerinin ortalama tensörü $\bar{\mathcal{A}}$ 'nin mod- m matrisleştirilmiş biçiminin j 'inci sütun vektörünü temsil eder. N_T ise tüm sınıflardaki örnekleri içeren eğitim kümesi toplam örnek sayısıdır.

Her bir iterasyonda, $\mathbf{U}_1, \dots, \mathbf{U}_{m-1}, \mathbf{U}_{m+1}, \dots, \mathbf{U}_N$ matrislerinin bilindiği varsayıldığında, diskriminant tensör kriteri Eşitlik 4.22'ye değiştirilir.

$$\mathbf{U}_m^* = \arg \max_{\mathbf{U}_m} \left(\frac{\sum_c N_C \|\bar{\mathcal{A}}^c \times_1 \mathbf{U}_1 \cdots \times_N \mathbf{U}_N - \bar{\mathcal{A}} \times_1 \mathbf{U}_1 \cdots \times_N \mathbf{U}_N\|^2}{\sum_i \|\mathcal{A}_i \times_1 \mathbf{U}_1 \cdots \times_N \mathbf{U}_N - \bar{\mathcal{A}}^{c_i} \times_1 \mathbf{U}_1 \cdots \times_N \mathbf{U}_N\|^2} \right) \quad (4.22)$$

Öznitelik tensörü, $\mathcal{Y}_i = \mathcal{A}_i \times_1 \mathbf{U}_1 \cdots \times_{k-1} \mathbf{U}_{k-1} \times_{k+1} \mathbf{U}_{k+1} \cdots \times_N \mathbf{U}_N$ biçiminde temsil edilirse Eşitlik 4.22, Eşitlik 4.23'e dönüşür.

$$\mathbf{U}_m^* = \arg \max_{\mathbf{U}_m} \frac{\sum_c N_C \|\bar{\mathcal{Y}}^c \times_k \mathbf{U}_k - \bar{\mathcal{Y}} \times_k \mathbf{U}_k\|^2}{\sum_i \|\mathcal{Y}_i \times_k \mathbf{U}_k - \bar{\mathcal{Y}}^{c_i} \times_k \mathbf{U}_k\|^2} \quad (4.23)$$

Eşitlik 4.23'de gösterilen ifade, Eşitlik 4.18'deki ifadenin \mathcal{A}_i yerine \mathcal{Y}_i yazılmış biçimindedir. Bu yüzden, m -mod optimizasyonda Eşitlik 4.23, Eşitlik 4.18 yerine kullanılabilir. MDA algoritması, aşağıda özetlenmiştir.

MDA Algoritma Basamakları

- *Giriş parametreleri:* Sıfır ortalamalı eğitim veri tensörü $\mathcal{X} \in \mathfrak{R}^{d_1 \times d_2 \times \cdots \times d_n \times N_T}$, sınıf etiketleri $c_i \in \{1, 2, \dots, N_C\}$ ve indirgenecek olan boyutlar : $\tilde{d}_1 \times \tilde{d}_2 \times \cdots \times \tilde{d}_N$

- *Basamak 1:* İlk izdüşüm matrislerini birim matris olarak ata.

$$\mathbf{U}_1^0 = \mathbf{I}_{d_1 \times d_1}, \quad \mathbf{U}_2^0 = \mathbf{I}_{d_2 \times d_2}, \quad \dots \quad \mathbf{U}_N^0 = \mathbf{I}_{d_N \times d_N}$$

- *Basamak 2:*

For $t = 1, 2, \dots, T_{\max}$ do

a) For $m = 1, 2, \dots, N$ do

$$\mathcal{Y}_i = \mathcal{X}_i \times_1 \mathbf{U}_1^t \cdots \times_{k-1} \mathbf{U}_{k-1}^t \times_{k+1} \mathbf{U}_{k+1}^t \cdots \times_N \mathbf{U}_N^t$$

$$\mathbf{Y}_m^i \xleftarrow{m\text{-mod matrisleřtirme}} \mathcal{Y}_i$$

$$\mathbf{S}_B = \prod_{j=1}^{d_o} \left(\sum_{c=1}^{N_C} N_C (\bar{\mathbf{Y}}_{m,j}^c - \bar{\mathbf{Y}}_{m,j}) (\bar{\mathbf{Y}}_{m,j}^c - \bar{\mathbf{Y}}_{m,j})^T \right)$$

$$\mathbf{S}_W = \prod_{j=1}^{d_o} \left(\sum_{i=1}^{N_T} (\mathbf{Y}_{m,j}^i - \bar{\mathbf{Y}}_{m,j}^{c_i}) (\mathbf{Y}_{m,j}^i - \bar{\mathbf{Y}}_{m,j}^{c_i})^T \right)$$

$$\mathbf{S}_B \mathbf{U}_m^t = \mathbf{S}_W \mathbf{U}_m^t \mathbf{\Lambda}_m, \quad \mathbf{U}_m^t \in \mathfrak{R}^{d_m \times \tilde{d}_m}$$

b) If $t > 2$ and $\|\mathbf{U}_m^t - \mathbf{U}_m^{t-1}\| < \tilde{d}_m d_m \varepsilon$, $m = 1, 2, \dots, N$, break;

- *Basamak 3: Çıkış izdüřümleri:* $\mathbf{U}_m = \mathbf{U}_m^t \in \mathfrak{R}^{d_m \times \tilde{d}_m}$, $m = 1, 2, \dots, N$.

MDA algoritması son basamağında optimum çıkış izdüřümleri belirlendikten sonra, eğitim verisindeki her bir örneğın izdüřümü Eşitlik 4.24 kullanılarak belirlenir. Sınıflandırılacak olan test verisi $\mathcal{A}_{test} \in \mathfrak{R}^{d_1 \times d_2 \times \dots \times d_N}$ 'in izdüřümü de Eşitlik 4.25'deki gibi hesaplanır ve eğitimdeki tüm özniteliklerle karşılaştırılır. Eşitlik 4.26'da gösterilen en yakın komşu sınıflandırıcısı kararına göre sınıf ataması yapılır.

$$\mathcal{Z}_i = \mathcal{A}_i \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \cdots \times_N \mathbf{U}_N, \quad \forall i \quad (4.24)$$

$$\mathcal{Z}_{test} = \mathcal{A}_{test} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \cdots \times_N \mathbf{U}_N \quad (4.25)$$

$$c_i^* = \arg \min_{\forall i} \|\mathcal{Z}_i - \mathcal{Z}_{test}\| \quad (4.26)$$

BÖLÜM 5

TENSÖREL ORTAK BİLEŞEN YÖNTEMLERİ

Bu bölümde, geliştirdiğimiz tensörel ortak bileşen yöntemlerine yer verilmiştir. Tez çalışması süresince 3 çeşit tensörel ortak bileşen yöntemi (TOBY) geliştirilmiştir. Bunlardan birinci yöntem, Gram-Schmidt temeline dayalı ortak vektör yaklaşımının doğrudan tensör gösterimlerine genişletilmesidir ve Gram-Schmidt birimdikleştirme temeline dayanan tensörel ortak bileşen yöntemi başlığı altında verilmiştir. İkinci yöntem, yüksek dereceden tekil değer ayrıştırmasına dayanır ve HOSVD temeline dayanan TOBY başlığı altında verilmiştir. Üçüncü yöntemse, multilineer diskriminant analizi yöntemine dayanır ve tensör diskriminant analizi temeline dayanan TOBY başlığı altında verilmiştir.

5.1 Gram-Schmidt Birimdikleştirme Temeline Dayanan Tensörel Ortak Bileşen Yöntemi

Gram-Schmidt birimdikleştirme temeline dayanan tensörel ortak bileşen yöntemi (TOBY(GSO)) Bölüm 2.1.1’de verilen ortak vektör yaklaşımının tensör işlemlerine genişletilmiş biçimidir. Bu yöntemde Gram-Schmidt süreci vektörlere değil, tensör verisine uygulanacak biçimde uyarlanmıştır. Yöntemde, ortak vektör yaklaşımındaki gibi öncelikle aynı sınıfa ait olan eğitim verisi örneklerinin fark bileşenleri hesaplanır. Yüksek dereceden tensör verisi biçiminde ifade edilen fark bileşenleri birimdikleştirilerek fark altuzayı belirlenir. Herhangi bir veri örneğinin fark altuzayına olan izdüşümü farklılık bileşenini, bu bileşenin kendisinden çıkarılması ise ortak tensör bileşenini verir.

Örüntü tanıma probleminin eğitim kümesinin, herbir sınıfında M_c adet N ’inci dereceden tensör verisinden oluşan ve toplamda N_c adet sınıfa ait olan örneklerden

oluşturduğu kabul edilsin ve c 'inci sınıfın k 'inci örneği $\mathcal{A}_k^c \in \mathfrak{R}^{d_1 \times d_2 \times \dots \times d_N}$ sembolüyle gösterilsin. Böyle bir veri, ortak vektör yaklaşımının yetersiz veri koşulunu sağlarsa, Bölüm 2.1.1'de verilen teknik, yüksek dereceden tensör verilerine genişletilerek uygulanabilir. Bu koşullar altında, yetersiz veri durumu, $M_c \leq \left(\prod_{n=1}^N d_n \right)$, $\forall c$ eşitsizliği sağlandığında oluşur. Bu eşitlik sağlanmışsa, Gram-Schmidt süreci tensörel veriler için uyarlanarak TOBY(GSO) yöntemi çalıştırılabilir. Algoritma basamakları aşağıda verilmiştir.

TOBY(GSO) algoritma basamakları

- *Basamak 1:* Sınıf sayısı $c = 1$ ata.
- *Basamak 2:* İlk veri örneğini diğer tüm veri örneklerinden çıkararak sınıf fark bileşenlerini oluştur.

$$\mathcal{B}_{h-1}^c = \mathcal{A}_h^c - \mathcal{A}_1^c, \quad h = 2, 3, \dots, M_c$$

- *Basamak 3:* İlk fark bileşenini kendi normuna böl.

$$\mathcal{Z}_1^c = \frac{\mathcal{B}_1^c}{\|\mathcal{B}_1^c\|}$$

- *Basamak 4:* $j = 2, t = 2$ ata.
- *Basamak 5:* Aşağıdaki döngüyü tüm fark bileşenleri için tekrarla:

while $j \leq M_c - 1$,

do {

$$\mathcal{E}_j^c = \mathcal{B}_j^c - \sum_{i=1}^{j-1} (\langle \mathcal{B}_j^c, \mathcal{Z}_i^c \rangle \mathcal{Z}_i^c)$$

$$\text{if } \|\mathcal{E}_j^c\| \neq 0 \rightarrow \{ \mathcal{Z}_t^c = \frac{\mathcal{E}_j^c}{\|\mathcal{E}_j^c\|}, t = t + 1 \}$$

$$j = j + 1$$

}

- *Basamak 6:* Herhangi bir örnek için fark bileşeni aşağıdaki gibidir.

$$\mathcal{A}_{k,dif}^c = \sum_{\forall t} (\langle \mathcal{A}_k^c, \mathcal{Z}_t^c \rangle \mathcal{Z}_t^c)$$

- *Basamak 7*: Fark bileşenini kullanarak sınıf ortak bileşenini hesapla:

$$\mathcal{A}_{com}^c = \mathcal{A}_k^c - \mathcal{A}_{k,dif}^c \quad \forall k$$

- *Basamak 8*: Sınıf sayısını bir arttır ($c = c + 1$) ve basamak 2'ye dön. Tüm sınıflar tamamlanınca dur.

Algoritma basamaklarından görüldüğü gibi, TOBY(GSO) yönteminde tüm işlemler tensör verisi üzerinden gerçekleştirilir. Tensör iç-çarpım işleminde Eşitlik 4.3, tensör norm işleminde Eşitlik 4.4 kullanılır. Her bir sınıf için basamak 7'de bulunan ortak bileşen tektir ve sınıf ayırt edici özneliği olarak kullanılabilir.

Eğitim aşamasında sınıf ortak bileşenleri ve üzerlerine izdüşüm alınan \mathcal{Z}_t^c ($\forall t, \forall c$) tensörleri test aşamasında kullanılmak üzere elde edilir. Sınıflandırma aşamasında ise, hangi sınıfa ait olduğu bilinmeyen test tensörünün ($\mathcal{A}_{test} \in \mathfrak{R}^{d_1 \times d_2 \times \dots \times d_N}$) ortak bileşeni eğitim aşamasındaki izdüşümler kullanılarak Eşitlik 5.1'deki gibi çıkarılır ve sınıf ortak bileşeni ile karşılaştırılır. Sınıf kararı, test ortak bileşeni ile tüm sınıfların ortak bileşenleri arasındaki uzaklık değerlerinin en küçüğünün elde edildiği sınıf olarak verilir. TOBY(GSO) sınıflandırıcı eşitliği Eşitlik 5.2'de gösterilmiştir. Uzaklık ölçüsü olarak Eşitlik 5.3 kullanılabilir.

$$\mathcal{A}_{test,com}^c = \mathcal{A}_{test}^c - \sum_{\forall t} (\langle \mathcal{A}_{test}^c, \mathcal{Z}_t^c \rangle \mathcal{Z}_t^c) \quad (5.1)$$

$$d(\mathcal{A}_{test,com}^c, \mathcal{A}_{com}^c) = \arg \min_{\forall c} \{d(\mathcal{A}_{test,com}^c, \mathcal{A}_{com}^c)\} \quad (5.2)$$

$$d(\mathcal{A}_{test,com}^c, \mathcal{A}_{com}^c) = \|\mathcal{A}_{test,com}^c - \mathcal{A}_{com}^c\| \quad (5.3)$$

TOBY(GSO) yöntemi, Eşitlik 5.3'de yer alan norm ölçüsü olarak Eşitlik 4.4'de verilen Frobenius norm kullanıldığında, 1'inci dereceden tensör verisi (vektör) için

Bölüm 2.1.1’de verilen ortak vektör yaklaşımına denktir. TOBY(GSO) yöntemi, ortak vektör yaklaşımının tensörel veri için multilineer cebir işlemlerine genişletilmiş genel bir biçimidir. Ortak vektör yaklaşımında, 2’nci veya daha yüksek dereceden tensör verilerinin işlenmesi, verinin vektöre dönüştürülmesiyle mümkündür. TOBY(GSO) yönteminde ise veri orijinal biçiminde muhafaza edilerek işlem yapılır ve aynı sonuçlar elde edilebilir. Eşitlik 5.3’deki uzaklık ölçüsünde Frobenius normdan farklı bir uzaklık ölçüsü kullanılırsa, yöntemin performansı ortak vektör yaklaşımından farklı olacaktır. Örneğin, gri seviyeli sayısal görüntüler 2’nci dereceden tensör biçiminde olduğu için Eşitlik 3.35’de verilen AMD matris uzaklık ölçüsü kullanılarak TOBY(GSO) yönteminin tanıma başarısı OVY tanıma başarımından daha iyi bir biçime getirilebilir. Sayısal görüntü tanıma deneyleri bölümünde, AMD uzaklığı kullanan TOBY(GSO) yönteminin bazı veritabanlarında OVY’den daha iyi tanıma oranları verdiği belirtilmiştir.

5.2 Yüksek Dereceden Tekil Değer Ayrıştırması Temeline Dayanan Tensörel Ortak Bileşen Yöntemi

Bu bölümde, ortak vektör yaklaşımı ile temel bileşen analizi arasındaki ilişkiyi yola çıkarak ve dördüncü bölümde bahsedilen multilineer temel bileşen analizi yöntemindeki yüksek dereceden tekil değer ayrıştırmasından yararlanarak geliştirdiğimiz tensörel ortak bileşen yöntemi sunulmuştur. Yüksek dereceden tekil değer ayrıştırması temeline dayanan tensörel ortak bileşen yöntemi (TOBY(HOSVD)) adını verdiğimiz bu yeni yöntem, yüksek dereceden tensör verilerine uygulanabilir. Önerdiğimiz yöntem aynı zamanda üçüncü bölümde verilen iki boyutlu yöntemlerin daha yüksek dereceden tensörlere uygulanabilen bir genellemesi özelliğini taşımaktadır.

Önceki bölümlerde olduğu gibi görüntü tanıma probleminin eğitim kümesi, herbir sınıfında M_c adet N ’inci dereceden tensör verisinden oluşan ve toplamda N_c adet sınıfa ait olan örneklerden oluşsun. Bu durumda c ’inci sınıfın k ’inci örneği $\mathcal{A}_k^c \in \mathfrak{R}^{d_1 \times d_2 \times \dots \times d_N}$ sembolüyle gösterilsin. İlk aşamada, herbir sınıf için eğitim verisi sıfır

ortalamalı olacak biçimde bir veri tensörü oluşturulur. Eğitim verisi c 'inci sınıf ortalama tensörü, Eşitlik 5.4'de gösterilmiştir.

$$\bar{\mathcal{A}}^c = \frac{1}{M_c} \sum_{k=1}^{M_c} \mathcal{A}_k^c \quad (5.4)$$

Sıfır ortalamalı sınıf tensörü de, sınıftaki her bir örneğin sınıf ortalamasından çıkarılarak yeni bir tensörün $(N+1)$ 'inci yönüne gömülmesiyle oluşturulur. Sıfır ortalamalı eğitim örneklerinden oluşturulan bu tensör $\mathcal{D}^c \in \mathfrak{R}^{d_1 \times d_2 \times \dots \times d_N \times M_c}$ biçiminde gösterilsin. Eğitim verisi olarak adlandırılan bu tensöre MPCA yöntemine benzeyen bir işlemle yüksek dereceden tekil ayrıştırması yapılarak mod izdüşüm matrisleri elde edilebilir. Bu mod izdüşüm matrislerinin sütun vektörleri de fark ve farksızlık altuzaylarını betimleyen taban vektörleri olarak kullanılabilir.

Yüksek dereceden tekil değer ayrıştırması, (Lathauwer, et al., 2000) kaynakçasında ayrıntılı olarak verilmiştir. Kısaca özetlemek gerekirse, N 'nci dereceden herhangi bir $\mathcal{A} \in \mathfrak{R}^{I_1 \times I_2 \times \dots \times I_N}$ tensörü $\mathcal{A} = \mathcal{S} \times_1 U_1 \times_2 U_2 \dots \times_N U_N$ çarpımı biçiminde ayrıştırılabilir. Burada $\mathcal{S} \in \mathfrak{R}^{I_1 \times I_2 \times \dots \times I_N}$ çekirdek tensör, $U_n \in \mathfrak{R}^{I_n \times I_n}$ ($n=1, \dots, N$) ise birimcil (unitary) matrislerdir. Çekirdek tensörü aşağıdaki iki özelliği sağlar.

1) $\mathcal{S}_{i_n=\tau}$ tensörü \mathcal{S} tensörünün (n 'inci indisini τ değerine sabitleyerek bulunan) bir alt-tensörü olarak tanımlandığında, \mathcal{S} tensörünün herhangi iki alt-tensörü $\mathcal{S}_{i_n=\alpha}$ ve $\mathcal{S}_{i_n=\beta}$, $\alpha \neq \beta$ koşulu altında mümkün olan tüm n , α ve β değerleri için birbirine diktir ($\langle \mathcal{S}_{i_n=\alpha}, \mathcal{S}_{i_n=\beta} \rangle = 0, \alpha \neq \beta$).

2) \mathcal{S} tensörünün alt-tensörleri aşağıdaki sıralamaya göre dizilmiştir:

$$\|\mathcal{S}_{i_n=1}\| \geq \|\mathcal{S}_{i_n=2}\| \geq \dots \geq \|\mathcal{S}_{i_n=I_n}\| \geq 0.$$

Örüntü tanıma problemindeki sıfır ortalamalı veri tensörüne yüksek dereceden tekil değer ayrıştırması uygulandığında Eşitlik 5.5 elde edilir.

$$\mathcal{D}^c = \mathcal{S}^c \times_1 \mathbf{U}_1^c \times_2 \mathbf{U}_2^c \times \cdots \times_N \mathbf{U}_N^c \times_{N+1} \mathbf{U}_{N+1}^c \quad (5.5)$$

Burada, \mathcal{S}^c çekirdek tensörü, \mathbf{U}_j^c ($j=1,2,\dots,N+1$) ise sütunları birimdik olan taban vektörlerinden oluşan birimcil matrislerdir. Aslında, ortak bileşen analizi için çekirdek tensörünü belirlemeye gerek yoktur. Ortak bileşen, fark ve farksızlık altuzayları üzerine olan izdüşümden bulunduğu için, birimcil matrislerin belirlenmesi yeterlidir. Lathauwer, et al., (2000)'e göre birimcil matrisler, veri matrisinin ilgili moda düzleştirilip, tekil değer ayrıştırması sonucu belirlenen sol matrise eşittir. Özetlemek gerekirse, $\mathbf{U}_1^c \in \mathfrak{R}^{d_1 \times d_1}$ matrisi $\mathcal{D}_{(1)}^c \in \mathfrak{R}^{d_1 \times d_2 d_3 \cdots d_N M_c}$ matrisinin, $\mathbf{U}_2^c \in \mathfrak{R}^{d_2 \times d_2}$ matrisi

$\mathcal{D}_{(2)}^c \in \mathfrak{R}^{d_2 \times d_1 d_3 \cdots d_N M_c}$ matrisinin, benzer biçimde $\mathbf{U}_n^c \in \mathfrak{R}^{I_n \times I_n}$ matrisi de $\mathcal{D}_{(n)}^c \in \mathfrak{R}^{I_n \times \prod_{\forall m, m \neq n} I_m}$ matrisinin tekil değer ayrıştırması sonucunda bulunan sol matrislerdir. Veri örneği $\mathcal{A}_k^c \in \mathfrak{R}^{d_1 \times d_2 \times \cdots \times d_N}$ boyutlarında olduğu için, veri tensörü ayrıştırmasında $\mathbf{U}_{N+1}^c \in \mathfrak{R}^{M_c \times M_c}$ kullanılmaz. Her bir mod yönündeki izdüşüm, ilgili modun birimcil matrisiyle belirlenir.

Birimcil matrisler, ortak bileşen hesabında kullanılabilirler. Birimcil matrislerin sütun vektörleri, tekil değer ayrıştırması sonucu bulunan özvektörler olduğuna göre birimdik vektörlerdir. Ortak vektör yaklaşımı prensiplerinden benzetim yapıldığında, her bir birimcil matrisin sıfır olan özdeğere karşılık gelen özvektörleri farksızlık altuzayını, sıfırdan büyük özdeğerlere karşılık gelen özvektörler ise fark altuzayını temsil eden taban vektörleri olmalıdır. Yine önceki bölümlerde verilen bilgiler dahilinde, ayrıştırılan matrisin sıfıra eşit özdeğer içermesi için sol sıfır uzayının bulunması gerekir, bir başka

ifadeyle n 'inci mod matrisi için $\left(\prod_{\forall m, m \neq n} I_m \right) \leq I_n$ koşulu sağlanmalıdır. Bu koşul, I_n

boyutları genellikle birbirine yakın olduğu için hiçbir zaman sağlanmaz. Bu yüzden, birbirinin tümleyeni olan fark ve farksızlık altuzayları yoktur. Bu durumda, yaklaşık olarak fark altuzayı ve yaklaşık olarak farksızlık altuzayını belirleyecek olan özvektör

sayısı giriş parametresi olarak seçilmelidir. Tekil değerler büyükten küçüğe doğru dizildiğinde, ilk r_n özvektör fark altuzayı taban vektörlerini, geriye kalan özvektörlere farksızlık altuzayı taban vektörlerini temsil ettiği kabul edilir. Genel olarak mod- n birimcil matris ve sütun vektörleri Eşitlik 5.6'da verildiği gibi gösterilebilir.

$$\mathbf{U}_n^c = [\mathbf{u}_{n,1}^c \quad \mathbf{u}_{n,2}^c \quad \cdots \quad \mathbf{u}_{n,r_n}^c \quad \mathbf{u}_{n,r_n+1}^c \quad \cdots \quad \mathbf{u}_{n,I_n}^c], \quad n = 1, 2, \dots, N \quad (5.6)$$

Bir sonraki aşama fark ve farksızlık altuzayları üzerine olan izdüşümlerin belirlenmesidir. Birim matrisler, her bir mod yönünde belirlendiği için, izdüşüm matrisleri de herbir mod için ayrıca hesaplanmalıdır. Eşitlik 5.6'da gösterilen mod- n birimcil matrisinin birimcik sütun vektörleri kullanılarak, c 'inci sınıfta n 'inci mod için fark altuzayına olan izdüşüm matrisi $\mathbf{P}_{n,dif}^c$ sembolüyle, farksızlık altuzayına olan izdüşüm matrisi de $\mathbf{P}_{n,indif}^c$ biçiminde gösterildiğinde, altuzaylara olan izdüşümler sırasıyla Eşitlik 5.7 ve 5.8'de verildiği gibi belirlenir.

$$\mathbf{P}_{n,dif}^c = \sum_{j=1}^{r_n} (\mathbf{u}_{n,j}^c)(\mathbf{u}_{n,j}^c)^T, \quad n = 1, 2, \dots, N \quad (5.7)$$

$$\mathbf{P}_{n,indif}^c = \sum_{j=r_n+1}^{I_n} (\mathbf{u}_{n,j}^c)(\mathbf{u}_{n,j}^c)^T, \quad n = 1, 2, \dots, N \quad (5.8)$$

Buna göre, fark altuzayına olan tensör izdüşümü \mathcal{P}_{dif}^c , farksızlık altuzayına olan tensör izdüşümü \mathcal{P}_{indif}^c biçiminde gösterildiğinde, tüm modları içeren tensör izdüşümleri de Eşitlik 5.9 ve 5.10'daki gibidir.

$$\mathcal{P}_{dif}^c = \times_1 \mathbf{P}_{1,dif}^c \times_2 \mathbf{P}_{2,dif}^c \times \cdots \times_N \mathbf{P}_{N,dif}^c \quad (5.9)$$

$$\mathcal{P}_{indif}^c = \times_1 \mathbf{P}_{1,indif}^c \times_2 \mathbf{P}_{2,indif}^c \times \cdots \times_N \mathbf{P}_{N,indif}^c \quad (5.10)$$

Fark ve farksızlık altuzayları yaklaşık olarak belirlendiği için ortak bileşen hesabı fark altuzayı üzerinden yapılacak olan tahminle veya farksızlık altuzayı üzerinden yapılacak olan tahminle veya hem fark hem farksızlık uzayı tahminlerinden yapılacak olan ortak bir tahminle hesaplanabilir. Altuzaylar yaklaşık olarak belirlendiği için, ortak matris hesabında sınıf ortalaması altuzaylar üzerine izdüştürülmelidir. Sadece fark altuzayı üzerinden ortak bileşen tahmini $\mathcal{A}_{com,(Pdif)}^c$ biçiminde temsil edilirse Eşitlik 5.11'de verildiği gibi belirlenebilir. Benzer şekilde sadece farksızlık altuzayı üzerinden yapılan ortak bileşen tahmini $\mathcal{A}_{com,(Pindif)}^c$ biçiminde temsil edilirse Eşitlik 5.12'de verildiği gibi belirlenebilir. Her iki tahmini de içeren ortak bileşen tahmini ise $\mathcal{A}_{com,(Pdif+Pindif)}^c$ biçiminde temsil edildiğinde Eşitlik 5.13'de gösterildiği gibi belirlenebilir.

$$\mathcal{A}_{com,(Pdif)}^c = \bar{\mathcal{A}}^c - (\bar{\mathcal{A}}^c \times \mathcal{P}_{dif}^c) = \bar{\mathcal{A}}^c - (\bar{\mathcal{A}}^c \times_1 \mathbf{P}_{1,dif}^c \times_2 \mathbf{P}_{2,dif}^c \times \cdots \times_N \mathbf{P}_{N,dif}^c) \quad (5.11)$$

$$\mathcal{A}_{com,(Pindif)}^c = \bar{\mathcal{A}}^c \times \mathcal{P}_{indif}^c = \bar{\mathcal{A}}^c \times_1 \mathbf{P}_{1,indif}^c \times_2 \mathbf{P}_{2,indif}^c \times \cdots \times_N \mathbf{P}_{N,indif}^c \quad (5.12)$$

$$\mathcal{A}_{com,(Pdif+Pindif)}^c = \frac{\mathcal{A}_{com,(Pdif)}^c + \mathcal{A}_{com,(Pindif)}^c}{2} \quad (5.13)$$

Fark ve farksızlık altuzayları, tümleyen uzaylar olmadığı için 5.11-5.13 eşitliklerinde verilen tensörel ortak bileşenler birbirine eşit değildir. Bu yüzden, TOBY(HOSVD) yöntemiyle ortak bileşen çıkarmada, 5.11-5.13 eşitliklerinden herhangi biri kullanılmalıdır. Yapılan deneysel çalışmalarda Eşitlik 5.12'de verilen, sadece \mathcal{P}_{indif}^c izdüşümü üzerinden hesaplanan ortak bileşen eğitim verisi için genellikle %100 tanıma oranları başarımına rağmen test verisi söz konusu olduğunda aynı başarıyı sürdüremediği görülmüştür. Eşitlik 5.11'de verilen, sadece \mathcal{P}_{dif}^c izdüşümü üzerinden yapılan ortak bileşen tanıma sonuçları hem eğitim hem test verisinde genelde başarılıdır. Bu tahminin tanıma başarısı Eşitlik 5.13'de gösterilen en küçük hata kareleri kestiricisi tahminine yakındır. Bu yüzden, TOBY(HOSVD) yönteminin uygulanacağı

problemdede hesap karmaşıklığı sorunu olmadığı sürece ortak bileşen tahmin edicisi olarak Eşitlik 5.13 kullanılabilir. Hesaplama zamanından tasarruf etmek gerekirse de Eşitlik 5.11'deki ortak bileşen tahmininin kullanılması önerilir.

TOBY(HOSVD) yöntemi sınıflandırıcısı, önceki bölümde verilen sınıflandırıcılarla aynı prensipte sınıflandırma gerçekleştirir. Sınıflandırılacak olan test verisinin ortak bileşenleri, eğitim aşamasında belirlenen izdüşümler kullanılarak, her bir sınıf için belirlenir ve eğitim aşamasında bulunan sınıf ortak bileşeni ile karşılaştırılır. Test örneğine sınıf ataması, test verisinin ortak bileşeni ile sınıf ortak bileşeni arasındaki uzaklıkların en küçüğüne göre yapılır. $d(\mathcal{A}, \mathcal{B})$ işleci \mathcal{A} ve \mathcal{B} tensörleri arasındaki uzaklığı belirtir ve TOBY(HOSVD) sınıflandırıcı kararı Eşitlik 5.14'de verildiği gibidir.

$$\mathbf{Karar\ sınıfı\ } c : d(\mathcal{A}_{test,com}^c, \mathcal{A}_{com}^c) = \arg \min_{\forall c} \{d(\mathcal{A}_{test,com}^c, \mathcal{A}_{com}^c)\} \quad (5.14)$$

Özel bir durum olarak gri seviyeli görüntü tanıma problemlerinde, Eşitlik 3.35'de verilen AMD uzaklık ölçüsü kullanılabilir. Renkli sayısal görüntü tanıma deneylerinde ise yine AMD uzaklık ölçüsü her bir renk bileşeni için uygulanıp, renk bileşenlerinin uzaklıkları toplamı da toplam uzaklık olarak elde edilebilir.

5.3 Tensör Diskriminant Analizi Temeline Dayanan Tensörel Ortak Bileşen Yöntemi

Ortak vektör yaklaşımından ayırt edici ortak vektör yöntemine yapılan genellemeler göz önünde bulundurulduğunda, Bölüm 4.3'de verilen MDA yöntemi uyarlanarak, bir önceki bölümde TOBY(HOSVD) yöntemiyle hesaplanan ortak tensör bileşenleri, sınıf ayırt edici öznelikleri gibi kullanılabilir. Bu şekilde tensörel ortak bileşenler, sınıf içi benzerlikleri arttırıp, sınıflar arası benzerlikleri azaltmayı hedefleyen diskriminant özellikleri kazanır. Bu bölümde, MDA algoritması içinde denenen tensörel ortak bileşen yöntemi, tensör diskriminant analizi temeline dayanan tensörel ortak bileşimi yöntemi (TOBY(MDA)) ismiyle sunulmuştur.

TOBY(MDA) yönteminde, MDA yönteminin m -mod optimizasyon basamakları kullanılır. MDA yöntemine yapılan değişiklik, her bir mod- m izdüşüm hesaplarından sonra, bu izdüşümlerin TOBY(HOSVD) yöntemindeki gibi kullanılarak tensörel ortak bileşenler çıkarılmasıdır. Sınıf öznitelikleri saçılım matrisleri yerine sınıf ortak bileşen saçılımları kullanılır. Tensör diskriminant kriteri sınıf içi ortak bileşen saçılımını birbirine yaklaştırırken, sınıflar arası ortak bileşen saçılımını uzaklaştırmayı hedefler. Sonuç olarak, her bir modda ortak bileşenler üzerinden sınıflar arası saçılımı en büyüklerken, sınıf içi saçılımları en küçükleyen optimum izdüşüm matrisleri belirlenir. Böylece sınıflandırma bu izdüşümler üzerinden gerçekleştirilir. MDA yönteminden uyarlanan TOBY(MDA) algoritma basamakları aşağıda verilmiştir.

TOBY(MDA) Algoritma Basamakları

- *Giriş parametreleri:* Sıfır ortalamalı eğitim veri tensörü $\mathcal{X} \in \mathfrak{R}^{d_1 \times d_2 \times \dots \times d_n \times N_T}$, sınıf etiketleri $c_i \in \{1, 2, \dots, N_C\}$ ve indirgenecek olan boyutlar: $\tilde{d}_1 \times \tilde{d}_2 \times \dots \times \tilde{d}_N$

- *Basamak 1:* İlk izdüşüm matrislerini $\mathbf{P}_{n,dif}^0 \in \mathfrak{R}^{d_n \times d_n}$ olacak biçimde aşağıdaki gibi ata.

$$\mathbf{U}_1^0 = \mathbf{I}_{d_1 \times d_1 - 1}, \quad \mathbf{U}_2^0 = \mathbf{I}_{d_2 \times d_2 - 1}, \quad \dots \quad \mathbf{U}_N^0 = \mathbf{I}_{d_N \times d_N - 1}$$

$$\mathbf{P}_{1,dif}^0 = (\mathbf{U}_1^0)(\mathbf{U}_1^0)^T, \quad \mathbf{P}_{2,dif}^0 = (\mathbf{U}_2^0)(\mathbf{U}_2^0)^T, \quad \dots \quad \mathbf{P}_{N,dif}^0 = (\mathbf{U}_N^0)(\mathbf{U}_N^0)^T$$

- *Basamak 2:*

For $t = 1, 2, \dots, T_{\max}$ do

a) For $m = 1, 2, \dots, N$ do

$$\mathcal{P}_{dif}^t = \times_1 \mathbf{P}_{1,dif}^t \times_2 \mathbf{P}_{2,dif}^t \dots \times_N \mathbf{P}_{N,dif}^t$$

$$\mathcal{Y}_i = \mathcal{A}_i - (\mathcal{A}_i \times \mathcal{P}_{dif}^t)$$

$$\bar{\mathcal{Y}}^c = \bar{\mathcal{A}}^c - (\bar{\mathcal{A}}^c \times \mathcal{P}_{dif}^t)$$

$$\begin{aligned}\bar{\mathcal{Y}} &= \bar{\mathcal{A}} - (\bar{\mathcal{A}} \times \mathcal{P}_{dif}^t) \\ \mathbf{Y}_m^i &\xleftarrow{m\text{-mod matrisleřtirme}} \mathcal{Y}_i \\ \bar{\mathbf{Y}}_m^c &\xleftarrow{m\text{-mod matrisleřtirme}} \bar{\mathcal{Y}}^c \\ \bar{\mathbf{Y}}_m &\xleftarrow{m\text{-mod matrisleřtirme}} \bar{\mathcal{Y}} \\ \mathbf{S}_B &= \prod_{j=1}^{\prod_{o \neq m} d_o} \left(\sum_{c=1}^{N_C} N_C (\bar{\mathbf{Y}}_{m,j}^c - \bar{\mathbf{Y}}_{m,j}) (\bar{\mathbf{Y}}_{m,j}^c - \bar{\mathbf{Y}}_{m,j})^T \right) \\ \mathbf{S}_W &= \prod_{j=1}^{\prod_{o \neq m} d_o} \left(\sum_{i=1}^{N_I} (\mathbf{Y}_{m,j}^i - \bar{\mathbf{Y}}_{m,j}^{c_i}) (\mathbf{Y}_{m,j}^i - \bar{\mathbf{Y}}_{m,j}^{c_i})^T \right) \\ \mathbf{U}_m^t &= \text{eig}(\mathbf{S}_W^{-1} \mathbf{S}_B), \mathbf{U}_m^t \in \mathfrak{R}^{d_m \times \tilde{d}_m} \\ \mathbf{P}_{m,dif}^t &= (\mathbf{U}_m^t) (\mathbf{U}_m^t)^T\end{aligned}$$

b) If $t > 2$ and $\|\mathbf{U}_m^t - \mathbf{U}_m^{t-1}\| < \tilde{d}_m d_m \varepsilon$, $m = 1, 2, \dots, N$, break;

- *Basamak 3:* Çıkış izdüřümleri: $\mathbf{U}_m = \mathbf{U}_m^t \in \mathfrak{R}^{d_m \times \tilde{d}_m}$, $m = 1, 2, \dots, N$.
- *Basamak 4:* Sınıflandırıcı:

$$\mathcal{Z}_i = \mathcal{A}_i \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \cdots \times_N \mathbf{U}_N, \quad \forall i$$

$$\mathcal{Z}_{test} = \mathcal{A}_{test} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \cdots \times_N \mathbf{U}_N$$

$$c_i^* = \arg \min_{\forall i} \|\mathcal{Z}_i - \mathcal{Z}_{test}\|$$

TOBY(MDA) yönteminin, MDA yönteminden en önemli farkı, öznitelik olarak tensörel ortak bileşenler kullanmasıdır. İzdüřümler, TOBY(HOSVD) yöntemindeki gibi alınarak ortak bileşen öznitelikleri elde edilir. Bu durum, TOBY(HOSVD) yönteminin sadece bir sınıflandırıcı olarak değil, aynı zamanda öznitelik çıkarıcı olarak kullanılabileceğini de göstermektedir. Bu öznitelikler MDA algoritmasında kullanılıp, algoritmanın diğer basamakları izlendiğinde MDA yönteminin yeni bir uyarlaması olan TOBY(MDA) oluşur.

TOBY(MDA) yönteminin MDA yönteminden bir diğer farkı da 1'inci basamaktaki rastgele izdüşüm belirleme aşamasıdır. MDA yönteminde birim matrisler tercih edilirken, TOBY(MDA) yönteminde birim matris tercih edilemez. Bunun nedeni, başlangıç izdüşümü birim matris alındığında ortak bileşen özneliğinin sıfır olmasıdır. Bu yüzden, TOBY(MDA) algoritmasının 1'inci basamağında, birim matris dışındaki herhangi bir matris alınmalıdır.

TOBY(MDA) sınıflandırıcısı, MDA yöntemi sınıflandırıcısıyla aynıdır. Ortak tensör bileşeni öznelikleri sınıflar arası uzaklaştırılacak, sınıf içi yakınlaştırılacak biçimde bir optimum dönüşüm belirlenir. Bu dönüşüm kullanarak sınıflandırmada kullanılacak olan ve boyut olarak indirgenmiş öznelikleri çıkarılır.

TOBY(MDA) yöntemi, özet olarak, MDA sınıflandırıcısı ile TOBY(HOSVD) yöntemi özneliklerini birleştiren bir yöntemdir. TOBY(MDA) yöntemi iteratif bir yöntem olduğu için eğitim aşamasının hesapsal yükü en iyi izdüşümün hangi iterasyonda bulunacağına bağlıdır. Eğitim aşaması tamamlandıktan sonra, sınıflandırma aşamasında bir boyutu indirgenmiş tensör öznelikleri kullandığından ve zaten eğitim aşamasında hesaplanan özneliklerle en yakın komşu karşılaştırması yapıldığından, bu yöntem test aşamasında hızlıdır.

BÖLÜM 6

SAYISAL GÖRÜNTÜ TANIMA DENEYLERİ

Bu bölümde, tez çalışmasında önerilen yöntemlerin verimliliğini test etmek için yapılan sayısal görüntü tanıma deneylerine yer verilmiştir. Deneylerde sayısal görüntü verisi içeren hazır veritabanları kullanılmıştır. Çalışmada ele alınan 3 veritabanı, insan yüzü portresi resimlerini içeren yüz veritabanları olup yüz görüntülerinden kimlik tanımlama amacıyla yüz tanıma deneylerinde kullanılmıştır. Dördüncü veritabanı ise nesne veritabanıdır. Bu veritabanı, farklı açılardan çekilen çeşitli nesnelere içermektedir. Çalışmada, bu veritabanı üzerinde nesne tanıma deneyleri yapılmıştır.

Deneylerde, tez çalışmasında önerilen iki boyutlu altuzay sınıflandırıcıları ve tensörel ortak bileşen yöntemlerinin yanı sıra, karşılaştırma yapabilmek için ortak vektör ve ayırtedici ortak vektör yöntemleri de çalıştırılmıştır. Yöntemlerin tanıma başarımları karşılaştırılırken, doğru tanıma oranlarına bakılmıştır. Doğru tanıma oranları hesaplanırken, tüm veritabanları deneyleri için aşağıdaki süreç izlenmiştir. Öncelikle, veritabanındaki her bir sınıfa ait olan örnekler rastgele seçimlerle eşit sayılar içeren iki gruba bölünür. Bu gruplardan birincisi eğitim verikümesi, diğeri test verikümesi olarak kullanılır. Giriş parametresi içermeyen yöntemler için, rastgele seçimler 10 kez tekrarlanır ve her tekrarda test verikümesi doğru tanıma oranları hesaplanır. Sonuç olarak, 10 adet test verikümesi doğru tanıma oranının ortalaması, yüzde cinsinden yöntemin doğru tanıma başarısı olarak verilir. Bu işlem, test aşaması olarak adlandırılır. Kullanıcının seçimine bırakılan giriş parametresi içeren yöntemlerde durum biraz daha karmaşıktır. En iyi giriş parametre değerinin belirlenebilmesi için yöntemler üzerinde test aşamasından önce eğitim aşaması uygulanmalıdır. Eğitim aşaması da test aşaması gibi uygulanır. Aynı sınıfa ait olan örneklerden yarısı rastgele seçilerek eğitim aşamasının eğitim verikümesini oluşturur ve geriye kalanları ise eğitim aşamasının test verikümesini oluşturur. Rastgele seçim 10 kez tekrarlanır ve 10 farklı eğitim ve test verikümesi elde edilir. Giriş parametresi (veya parametreleri) değiştirilerek test verikümesi doğru tanıma oranlarının ortalamaları elde edilir. En yüksek doğru tanıma

oranını veren parametre, optimum giriş parametresi olarak seçilir. Böylece eğitim aşaması tamamlanır. Test aşaması da benzer biçimde yürütülür. Her bir veri sınıfı için örneklerin yarısı rastgele seçilerek test aşamasının eğitim verikümesini, geriye kalan örnekler de test aşamasının test verikümesini oluşturur. Rastgele seçim işlemi 10 kez tekrarlanır. Eğitim aşamasında belirlenen giriş parametresi (veya parametreleri) kullanılarak, test aşamasının test verikümesi üzerinden doğru tanıma oranları hesaplanır. Doğru tanıma oranlarının ortalaması, yöntemin doğru tanıma başarısı olarak yüzde cinsinden verilir.

Bu tez çalışmasında yer alan yüz tanıma deneylerinde, karşılaştırma yapabilmek için sadece önerilen yöntemlerin verimliliği belirlenmekle kalmamış, aynı zamanda diğer bazı klasik yöntemlerin de verimliliği belirlenmiştir. Deneysel çalışmada kullanılan yöntemler, gri seviyeli görüntüler için 3 grupta toplanmıştır. Birinci grupta vektör tabanlı yöntemlerden temel bileşen analizi, ortak vektör yöntemi ve ayırt edici ortak vektör yöntemi denenmiştir. İkinci grup matris tabanlı yöntemler olup, 2DPCA ve 2DLDA yöntemlerinin yanı sıra 2DCLAFIC, 2DCLAFIC- μ ve 2DALS yöntemlerinin verimliliği elde edilmiştir. Üçüncü grup ise tensör tabanlı yöntemlerden oluşmaktadır ve tez çalışmasında önerilen TOBY(GSO), TOBY(HOSVD), TOBY(MDA) yöntemleriyle birlikte MDA yönteminin verimliliği belirlenmiştir. Renkli görüntülerin kullanıldığı deneylerde ise sadece vektör tabanlı ve tensör tabanlı yöntemler karşılaştırılmıştır.

Sayısal görüntü tanıma deneyleri sonuçları farklı veritabanları için farklı altbölümlerde verilmiştir. Her bir altbölümde ilgili veritabanı açıklanmış ve yöntemlerin doğru tanıma başarımları çizelgeler biçiminde özetlenmiştir.

6.1 Yale Veritabanı Yüz Tanıma Deneyleri

Yale yüz veritabanı (<http://cvc.yale.edu/projects/yalefaces/yalefaces.html>), orijinalde 15 kişinin farklı aydınlatma koşulları altında çekilen 11 poz gri seviyeli yüz resimlerinden oluşur. Veritabanında pozlar cepheden çekilmiş olup, mimik içermeyen doğal pozlarda normal, soldan aydınlatma, sağdan aydınlatma, cepheden aydınlatma

gibi aydınlatma farklılıkları içerdiği gibi, kişi gözlük takmıyorsa gözlüklü; gözlük takıyorsa gözlüksüz; şaşırmış, uykulu, göz kırpan, mutlu ve üzgün yüz mimiklerini içermektedir. Veritabanındaki resimlere, gözler yaklaşık olarak aynı konumlara yerleşecek ve yüz resmi 152×126 piksellik bir çerçeveye sığdırılacak biçimde bir önişleme uygulanmıştır. Eğitim ve test kümelerinin eşit sayıda görüntü içermesi için her bir kişinin 10 görüntüsü kullanılmıştır. Bu görüntülerden rastgele seçilen 5 tanesi eğitim verikümesini, geriye kalanlarsa test verikümesini oluşturur. Yale veritabanı için yüz tanıma deneyi, rastgele seçimlerle oluşturulan eğitim ve test verikümeleri kullanılarak bölüm girişinde anlatılan sürece uygun biçimde gerçekleştirilmiştir. Deneyde kullanılan görüntü örneklerinden rastgele seçilen 4 kişiye ait olan yüz görüntüleri Şekil 6.1’de gösterilmiştir.



Şekil 6.1 Yale veritabanından seçilen 4 bireyin görüntü örnekleri

Yale veritabanı 152×126 ’lık gri seviyeli görüntülerden oluştuğu için, vektör tabanlı yöntemler için her bir görüntü örneği $\mathbf{a}_j \in \mathcal{R}^{19152}$ boyutlarındadır. Vektörel yöntemlerden olan PCA giriş parametresi olarak özvektör sayısını kullanmaktadır ve bu giriş parametresi deneyin başında seçilmelidir. En iyi giriş parametresini belirlemek için eğitim aşaması yapılmış ve eğitim aşamasında en yüksek tanıma başarısını veren değer test aşamasında kullanılmıştır. Veri örneği boyutu, ortak vektör yaklaşımının yetersiz veri koşulunu sağlamaktadır ve ayırt edici ortak vektör yönteminin uygulanması için de uygundur. Hesap kolaylığı sağladığı için OVY ve AOVY yöntemlerinde Gram-Schmidt

birimdikleştirme süreci kullanan algoritma tercih edilmiştir. Matris tabanlı yöntemlerde, seçilmesi gereken bileşen sayısı veya altuzay boyutu mevcuttur. Bu yöntemler de önce eğitim aşamasında mümkün olan tüm giriş parametreleri için çalıştırılmış ve eğitim aşamasında en yüksek tanıma başarısını veren parametre test aşamasında kullanılmıştır. Tensörel yöntemlerden TOBY(GSO) için özelleştirilmesi gereken giriş parametresi mevcut değilken, TOBY(HOSVD), MDA ve TOBY(MDA) için gri seviyeli görüntü matrisinin her bir indisi için birer tane olmak üzere toplamda 2 adet bileşen belirleyici parametre vardır. Bu parametreler, çeşitli konfigürasyonlarla eğitim aşamasında denenmiş ve en yüksek tanıma başarısını verenler test aşamasında kullanılmıştır. Çizelge 6.1’de deneyde çalıştırılan yöntemlerin varsa giriş parametre değerleri, test aşaması doğru tanıma oranları ortalaması ve standart sapması verilmiştir.

Çizelge 6.1 Yale yüz veritabanı deney sonuçları

Yöntem	r_1	r_2	p	Test Aşaması doğru tanıma oranları ortalaması (%)	Test Aşaması doğru tanıma oranları standart sapması
PCA	15	–	–	72,80	4,03
OVY	–	–	–	75,73	4,44
AOVY	–	–	–	92,67	2,45
2DPCA	12	–	–	76,13	4,10
2DLDA	8	–	–	83,20	3,99
2DCLAFIC	7	–	–	77,73	4,95
2DALS	2	–	–	79,07	4,12
2DCLAFIC $-\mu$	2	–	–	81,73	4,49
TOBY(GSO)	–	–	2	75,73	4,44
TOBY(HOSVD)–1	3	20	1	79,87	4,10
TOBY(HOSVD)–2	3	20	2	81,87	3,22
MDA	5	5	–	70,80	7,26
TOBY(MDA)	10	5	–	82,13	4,32

Çizelgede gösterilen r_1 sembolü birinci indis yönünde seçilen özvektör sayısı, r_2 sembolü ikinci indis yönünde seçilen özvektör sayısı, p sembolü ise Bölüm 3.3.3’de verilen AMD uzaklık ölçüsü parametresidir. Bu parametreleri içeren yöntemlerin karşılığında ilgili parametre değeri yazılmış, içermeyen yöntemlerde ise çizgi çekilmiştir.

Yöntemlerden TOBY(HOSVD)–1 biçiminde kısaltılan yöntem, Eşitlik 5.11 kullanılarak hesaplanan tek ortak bileşen tahminli TOBY(HOSVD) yöntemi, TOBY(HOSVD)–2 biçiminde kısaltılan yöntemse, Eşitlik 5.13 kullanılarak hesaplanan iki ortak bileşen tahminli TOBY(HOSVD) yöntemini temsil etmektedir.

Çizelge 6.1’den görüldüğü üzere, Yale veritabanında doğru tanıma oranları cinsinden en yüksek başarıyı büyük farkla ayırt edici ortak vektör yöntemi gerçekleştirmiştir. Bu yöntemi ikinci sırada 2DLDA, üçüncü sırada TOBY(MDA), dördüncü sırada TOBY(HOSVD)–2 ve beşinci sırada 2DCLAFIC– μ yöntemi, % 80’in üzerinde ve birbirlerine yakın başarılı tanıma oranlarıyla takip etmiştir. En düşük tanıma oranını MDA yöntemi verirken PCA yöntemi de tanıma başarısında sondan ikincidir. önerilen TOBY(GSO) yöntemi ortak vektör yaklaşımıyla eşit başarı oranına sahiptir. Yale veritabanı için AOVY yönteminin başarısı açıktır. Ancak önerilen yöntemlerin hepsi ortak vektör yaklaşımından daha iyi tanıma oranlarına sahiptir. TOBY(MDA) yönteminin MDA yönteminden çok daha başarılı tanıma oranları vermesi bir diğer önemli nokta olarak göze çarpmaktadır. Yale veritabanı yüz tanıma deneyinde AOVY yöntemiyle kıyaslandığında çok yüksek başarılı tanıma oranları elde edilememesine rağmen, önerilen yöntemler AOVY ve 2DLDA haricindeki diğer yöntemlerden genellikle daha başarılı sonuçlar vermiştir.

6.2 ORL Veritabanı Yüz Tanıma Deneyleri

ORL veritabanı, Cambridge Üniversitesi AT&T Laboratuvarında oluşturulan (<http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html>) bir yüz veritabanıdır. Veritabanı, 40 bireyin 10’ar adet farklı poz ve farklı yüz ifadelerini içeren 92×112 boyutlarında gri seviyeli yüz görüntülerinden oluşmaktadır. Diğer veritabanlarından farklı olarak ORL veritabanında kişilere yüzlerini belirli yönlerde hareket ettirme serbestliği tanınmıştır ve kameranın açısı sabit değildir. Bu yüzden pozların tümü sadece cepheden çekilen görüntülerden oluşmaz. Deneylerde, ORL veritabanındaki yüz görüntüleri herhangi bir ön işleme yapılmadan kullanılmıştır. ORL veritabanından rastgele seçilen 4 bireyin görüntü örnekleri Şekil 6.2’de gösterilmiştir.



Şekil 6.2 ORL veritabanından seçilen 4 bireyin görüntü örnekleri

Bir önceki veritabanında yapılan deneylerde de ele alınan yöntemler kullanılarak, ORL veritabanı üzerinde yüz tanıma deneyleri yapılmıştır. Deney sonuçları benzer biçimde Çizelge 6.2’de özetlenmiştir.

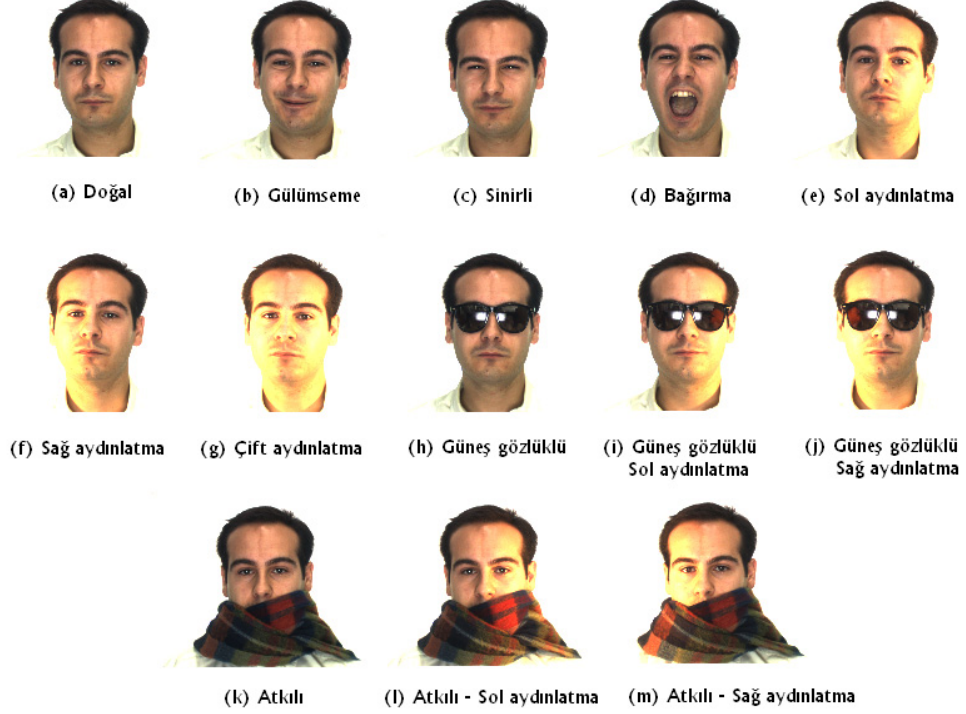
Çizelge 6.2 ORL yüz veritabanı deney sonuçları

Yöntem	r_1	r_2	p	Test Aşaması doğru tanıma oranları ortalaması (%)	Test Aşaması doğru tanıma oranları standart sapması
PCA	47	–	–	95,15	1,38
OVY	–	–	–	96,15	1,81
AOVY	–	–	–	96,20	1,48
2DPCA	11	–	–	95,70	1,40
2DLDA	8	–	–	96,40	1,70
2DCLAFIC	9	–	–	92,25	1,84
2DALS	9	–	–	92,25	1,84
2DCLAFIC- μ	2	–	–	93,90	1,41
TOBY(GSO)	–	–	0,125	96,35	1,51
TOBY(HOSVD)-1	5	5	0,5	96,30	1,55
TOBY(HOSVD)-2	8	3	0,125	95,35	1,23
MDA	5	10	–	95,60	1,71
TOBY(MDA)	8	10	–	96,85	1,58

ORL veritabanı için hem klasik hem de önerilen yöntemlerin tümü genelde yüksek tanıma oranları vermiştir. 2DCLAFIC, 2DCLAFIC- μ ve 2DALS haricinde geri kalan yöntemler % 95'in üzerinde başarılı tanıma gerçekleştirmiştir. En yüksek tanıma oranını önerilen yöntemlerden TOBY(MDA) verirken, bu yöntemi sırasıyla 2DLDA, TOBY(GSO), TOBY(HOSVD)-1, AOVY ve OVY takip etmiştir. ORL veritabanında, tensörel ortak bileşen yöntemleri, OVY ve AOVY yöntemlerinden daha iyi sonuçlar vermiştir. Ayrıca, $p = 0,125$ AMD uzaklık parametresini kullanan TOBY(GSO) yöntemi, OVY yönteminin tanıma başarısını geliştirmiştir. Normalde $p = 2$ için ortak vektör yaklaşımına denk olan yöntem, $p = 0,125$ değeri için OVY'den daha başarılıdır. HOSVD tabanlı TOBY yönteminin bir ortak bileşen tahminli ve iki ortak bileşen tahminli biçimleri karşılaştırıldığında, ORL veritabanı için tek ortak bileşen tahminli TOBY(HOSVD)-1 yönteminin, 2 ortak bileşen tahminli TOBY(HOSVD)-2 yönteminden daha başarılı olduğu göze çarpmaktadır. TOBY(MDA) yöntemi de ORL veritabanında MDA yönteminden daha yüksek doğru tanıma oranları vermiştir. Elde edilen sonuçları kısaca özetlemek gerekirse, önerilen tensörel ortak bileşen yöntemleri ORL veritabanında başarılı sonuçlar vermiştir.

6.3 AR Veritabanı Yüz Tanıma Deneyleri

AR yüz veritabanı (http://cobweb.ecn.purdue.edu/~aleix/alexi_face_DB.html), Aleix Martinez ve Robert Benavente (Martinez and Benavente, 1998) tarafından oluşturulmuştur. Veritabanı, 64 erkek, 53 bayan olmak üzere toplam 117 kişinin farklı aydınlatma ve farklı mimik koşullarında cepheden çekilmiş, 24-bit renkli, 768×576 'lık yüz fotoğraflarından oluşur. Veritabanında, her bir birey için 14 gün arası bulunan 13'er pozluk iki farklı oturumda çekilen toplam 26 görüntü bulunmaktadır. Şekil 6.3'de birinci oturumda çekilen 13 görüntü verilmiştir. Ondört gün arası bulunan ikinci oturumda yine aynı pozlar fotoğraflanmıştır.

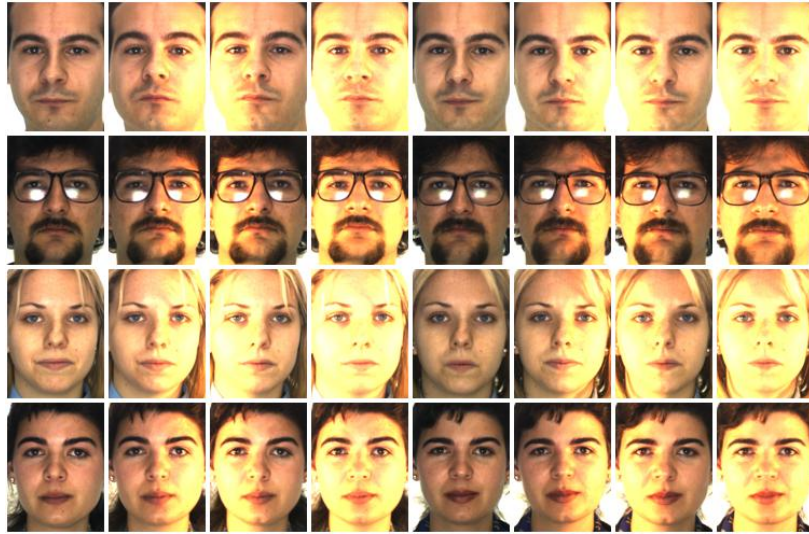


Şekil 6.3 AR veritabanından bir bireyin birinci oturum görüntüleri

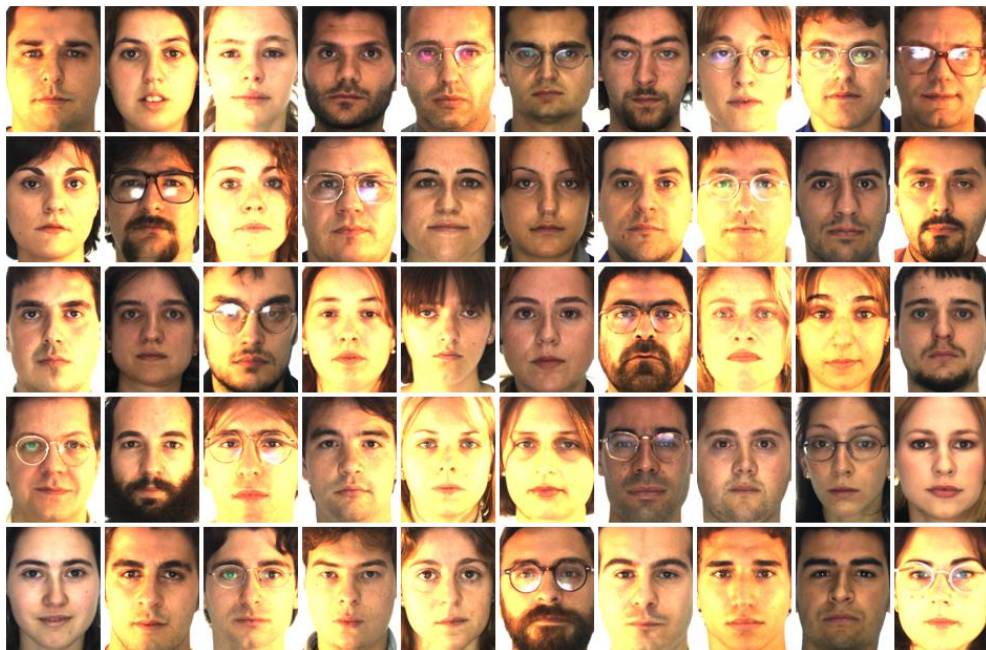
Deneyde kullanılan görüntüler, Şekil 6.3 de her iki oturumun (a), (e), (f) ve (g) pozlarıdır. Bu pozlar farklı aydınlatma şiddeti altında çekilen ve mimik içermeyen doğal yüz görüntüleridir. Görüntülere, öncelikle ön-işleme yapılmıştır. Ön-işleme basamağında, tüm bireylerin yüzleri aynı çerçeve içine sığacak ve özellikle gözler arası uzaklık yaklaşık olarak aynı kalacak biçimde yerleştirilip çerçeve dışında kalan bölgeler kesilmiş ve yüksek boyut sorunu da aşağı örneklemeyle (downsampling) giderilerek deneyde kullanılan veritabanı biçimlendirilmiştir. Ön-işleme sonucu her bir bireyin 60×45 boyutlarında 8 adet renkli görüntüsü elde edilmiştir. Bunlardan rastgele seçilen 4 örnek eğitim verikümesinde kullanılırken, geriye kalan 4 örnek de test verikümesi için kullanılır. AR veritabanında ön-işleme yapılan görüntü örnekleri Şekil 6.4'de gösterilmiştir.

AR veritabanı kullanılarak, renkli görüntüler ve gri seviyeli görüntüler için iki farklı deney yapılmıştır. Birinci deneyde sınıf sayısı azaltılarak 50'ye düşürülmüştür.

Rastgele seçilen 50 kişinin renkli ve gri seviyeli görüntüleri ile yüz tanıma deneyi gerçekleştirilmiştir. İkinci deneyde ise veritabanındaki 117 kişinin tüm görüntüleri kullanılmış ve yöntemlerin sınıflandırıcı performansları karşılaştırılmıştır. Birinci deney için seçilen 50 kişinin görüntüleri de Şekil 6.5’de gösterilmiştir.



Şekil 6.4 AR veritabanından ön-işlenmiş çeşitli görüntü örnekleri



Şekil 6.5 AR veritabanı 50 sınıflı deney için seçilen sınıflar

AR veritabanı üzerinde deneyler, önceki bölümlerdeki diğer deneylere benzer biçimde gerçekleştirilmiştir. Veritabanındaki renkli görüntüler, öncelikle gri seviyeli biçimine dönüştürülmüş ve gri seviyeli görüntüler üzerinde yüz tanıma deneyleri Bölüm 6.1 ve Bölüm 6.2’deki gibi yapılmıştır.

Gri seviyeli görüntüler matris biçiminde temsil edilirken renkli görüntü örnekleri, 3’üncü dereceden tensör biçiminde ifade edilir. Bu yüzden matris tabanlı yöntemler sadece gri seviyeli görüntülere uygulanabilir. Renkli görüntü deneylerinde, renkli görüntü örnekleri vektöre dönüştürülüp vektörel yöntemler ve 3’üncü dereceden tensör biçimleri korunarak tensörel yöntemler denenmiştir. Tensörel yöntemlerde, giriş parametresi olarak mod izdüşümlerini belirlemek için seçilmesi gereken özvektör sayıları renkli görüntüler 3’üncü derece tensör olduğundan 3 tanedir. AR veritabanında, Şekil 6.5’de gösterilen 50 sınıf için yapılan gri seviyeli görüntü tanıma deney sonuçları Çizelge 6.3’de, renkli görüntü tanıma deney sonuçları ise Çizelge 6.4’de verilmiştir.

Çizelge 6.3 AR yüz veritabanı 50 sınıf gri seviyeli görüntüler için deney sonuçları

Yöntem	r_1	r_2	p	Test Aşaması doğru tanıma oranları ortalaması (%)	Test Aşaması doğru tanıma oranları standart sapması
PCA	185	–	–	57,20	2,94
OVY	–	–	–	90,00	2,79
AOVY	–	–	–	99,80	0,26
2DPCA	43	–	–	86,70	2,52
2DLDA	13	–	–	99,55	0,64
2DCLAFIC	7	–	–	97,45	1,92
2DALS	7	–	–	97,45	1,92
2DCLAFIC $-\mu$	6	–	–	99,30	0,67
TOBY(GSO)	–	–	0,125	91,90	2,88
TOBY(HOSVD)–1	8	33	0,125	99,70	0,42
TOBY(HOSVD)–2	8	33	0,5	99,70	0,42
MDA	12	20	–	99,55	0,55
TOBY(MDA)	15	10	–	99,00	0,71

Çizelge 6.4 AR yüz veritabanı 50 sınıf renkli görüntüler için deney sonuçları

Yöntem	r_1	r_2	r_3	p	Test Aşaması doğru tanıma oranları ortalaması (%)	Test Aşaması doğru tanıma oranları standart sapması
OVY	–	–	–	–	90,25	2,51
AOVY	–	–	–	–	99,85	0,24
TOBY(GSO)	–	–	–	0,125	92,20	2,29
TOBY(HOSVD)–1	12	35	2	2	99,70	0,35
TOBY(HOSVD)–2	12	35	2	2	99,70	0,35
MDA	50	15	2	–	98,85	1,03
TOBY(MDA)	20	25	1	–	97,70	0,95

Çizelgelerden görüleceği üzere, 50 sınıf için AR veritabanı cepheden çekilmiş ancak aydınlatma farklılıkları içeren 8'er doğal poz üzerinde yapılan deneylerde, yöntemlerin bazıları % 99'un üzerinde doğru tanıma oranları vermiştir. Gri seviyeli görüntüler kullanılarak yapılan deneylerde en iyi tanıma oranını % 99,80 ile AOVY verirken, önerilen yöntemlerden TOBY(HOSVD) yöntemi hem bir hem de iki ortak bileşen tahminli yapısıyla % 99,70 doğru tanıma oranıyla AOVY yöntemini izlemiştir. Bu yöntemi de sırasıyla 2DLDA, MDA, 2DCLAFIC- μ ve TOBY(MDA) % 99'a büyük veya eşit olan başarılı tanıma oranlarıyla takip etmiştir. Bu yöntemlerin arkasından iki boyutlu yöntemler sıralanırken, bunları TOBY(GSO) ve OVY izlemiştir. AR veritabanı görüntülerinde de $p = 0,125$ AMD uzaklık parametresi kullanan TOBY(GSO) yönteminin doğru tanıma oranı, ortak vektör yaklaşımının doğru tanıma oranından daha yüksektir. Doğru tanıma oranları bakımından son iki sırayı 2DPCA ve PCA almıştır. Özellikle PCA yöntemi, diğer yöntemlerle kıyaslandığında oldukça başarısız bir tanıma performansı göstermiştir. Aydınlatma koşulları farkını yoğun miktarda içeren bir veritabanı söz konusu olduğunda önerilen yöntemlerin, en kötü başarıyı gerçekleştiren PCA ve 2DPCA yöntemlerinden oldukça başarılı sonuçlar verdiği söylenebilir.

Renkli görüntüler söz konusu olduğunda ise, MDA ve TOBY(MDA) yöntemlerinin doğru tanıma oranlarındaki küçük bir düşüş göze çarpmaktadır. TOBY(HOSVD) yönteminin gri seviyeli görüntülerdeki tanıma oranı aynı görüntülerin

renklileri için değişmezken, diğer yöntemlerde çok küçük bir miktarda iyileşme söz konusudur. Renkli görüntüler için yöntemler doğru tanıma oranlarına göre sıralandığında sıralama AOVY, TOBY(HOSVD)–1, TOBY(HOSVD)–2, MDA, TOBY(MDA), TOBY(GSO) ve OVY biçimindedir. Önerilen yöntemlerin tanıma başarımlarının ortak vektör yönteminin tanıma başarısından çok daha iyi olduğu söylenebilir. Normalde, $p = 2$ için OVY verimliliğine denk olan TOBY(GSO), $p = 0,125$ için OVY'den daha başarılıdır.

AR veritabanı üzerindeki ikinci deney, yine cepheden çekilmiş doğal pozlardan aydınlatma değişikliği içeren 8 tanesini kullanacak biçimde, ancak bu kez toplamda tüm sınıfları içeren (117 sınıf) verikümesi üzerinde uygulanmıştır. Böylece, sınıf sayısı arttığı zaman yöntemlerin verimliliğindeki değişimin gözlenmesi amaçlanmıştır. 117 bireyin gri seviyeli görüntüleri üzerine yapılan deney sonuçları Çizelge 6.5'de, renkli görüntüleri üzerine yapılan deney sonuçları Çizelge 6.6'da verilmiştir.

Çizelge 6.5 AR yüz veritabanı 117 sınıf gri seviyeli görüntüler için deney sonuçları

Yöntem	r_1	r_2	p	Test Aşaması doğru tanıma oranları ortalaması (%)	Test Aşaması doğru tanıma oranları standart sapması
PCA	450	–	–	48,93	1,81
OVY	–	–	–	86,73	3,05
AOVY	–	–	–	99,42	0,52
2DPCA	59	–	–	82,80	1,61
2DLDA	15	–	–	98,57	0,68
2DCLAFIC	10	–	–	97,37	0,75
2DALS	10	–	–	97,37	0,75
2DCLAFIC– μ	6	–	–	98,42	0,78
TOBY(GSO)	–	–	0,125	89,17	2,27
TOBY(HOSVD)–1	10	45	0,125	99,06	0,38
TOBY(HOSVD)–2	10	45	0,125	99,06	0,38
MDA	20	15	–	99,02	0,52
TOBY(MDA)	25	25	–	98,38	0,47

Çizelge 6.6 AR yüz veritabanı 117 sınıf renkli görüntüler için deney sonuçları

Yöntem	r_1	r_2	r_3	p	Test Aşaması doğru tanıma oranları ortalaması (%)	Test Aşaması doğru tanıma oranları standart sapması
OVY	–	–	–	–	86,24	2,67
AOVY	–	–	–	–	99,42	0,43
TOBY(GSO)	–	–	–	0,125	88,95	1,87
TOBY(HOSVD)–1	10	45	3	0,125	98,78	0,48
TOBY(HOSVD)–2	10	45	3	0,125	98,78	0,48
MDA	20	15	2	–	98,55	0,73
TOBY(MDA)	25	25	1	–	97,14	0,81

Deney sonuçlarında, sınıf sayısı artırıldığı zaman, yöntemlerin doğru tanıma oranlarındaki bir miktar düşüş göze çarpan ilk noktadır. Bunun sebebi, çoklu sınıflarda birbirine benzeşen örneklerin sayısının artması yüzünden yanlış sınıflandırma olasılığının artmasıdır. Çizelge 6.5 ve Çizelge 6.3 karşılaştırıldığında, 117 sınıf ve 50 sınıf için gri seviyeli görüntüleri doğru tanıma sıralaması aynıdır ancak doğru tanıma oranlarındaki kayıp farklı olabilir. Sınıflararası ayırt edici saçılım kullanan yöntemlerin tanıma kaybı yüzdesi, kullanmayanlarınkinden biraz daha düşüktür. Örneğin 2DPCA ve OVY yöntemlerinin 117 sınıf durumunda doğru tanıma oranı, 50 sınıf doğru tanıma oranından % 3'den daha fazla düşükken, 2DLDA ve AOVY yöntemlerinin tanıma oranlarındaki düşüş % 1'den daha azdır. 117 sınıflı görüntülerde AOVY en başarılı tanımayı gerçekleştirirken, TOBY(HOSVD) yakın bir yüzdeyle AOVY'yi takip eder. AR veritabanı için MDA tanıma başarımı TOBY(MDA) yönteminden biraz daha yüksektir. Fakat TOBY(MDA) 117 sınıf gri seviyeli görüntülerde % 98,38 ve renkli görüntülerde % 97,14'lük doğru tanıma oranlarıyla yine de başarılıdır. Önerilen tensörel ortak bileşen yöntemlerinin hepsi, ortak vektör yönteminin doğru tanıma başarısından daha yüksek oranlar vermiştir. AMD uzaklık ölçüsü kullanan TOBY(GSO) yöntemi de OVY tanıma başarısını geliştirmiştir. 117 sınıflı gri seviyeli ve renkli görüntü tanıma deneyleri karşılaştırıldığında, renkli görüntü tanıma başarımlarının gri seviye tanıma başarımlarından biraz daha kötü olduğu göze çarpmaktadır. Bunun temel sebebi, seçilen pozlardaki aydınlatma şiddetlerinin farklı olmasıdır. Renk bileşenleri de eklendiğinde aydınlatma şiddeti farkları tanıma oranını iyileştirme etkisi gösterememiştir.

6.4 COIL100 Veritabanı Nesne Tanıma Deneyleri

Tez çalışmasında önerilen yöntemler, sadece yüz tanıma değil, değişik görüntü tanıma uygulamalarında da kullanılabilirler. Yüz adet değişik nesnenin çeşitli görüntülerinden oluşan COIL100 veritabanı (Nene, et al., 1996) bir nesne veritabanıdır (<http://www1.cs.columbia.edu/CAVE/software/softlib/coil-100.php>). Tez çalışmasında denenen yöntemlerin verimliliği, COIL100 veritabanı görüntüleri kullanılarak da test edilmiştir.

Orijinalde, COIL100 nesne veritabanı, 128×128'lik renkli görüntülerden oluşmaktadır. Her bir nesne, motorla dönen koyu renkli bir platform üzerine yerleştirilip tam tur atıncaya kadar kendi eksenini etrafında 5'er derece döndürülerek fotoğraflanmıştır. Böylece her nesnenin 72 adet görüntüsü mevcuttur. COIL100 veritabanındaki bir zımba nesnesinin tüm pozları Şekil 6.6'da gösterilmiştir.



Şekil 6.6 COIL100 veritabanındaki bir nesnenin görüntüleri

COIL100 veritabanı deneylerinde, hesap kolaylığı sağlaması açısından görüntülere aşağı örnekleme uygulanarak boyutları 64×64'e indirgenmiştir. Deneyde, 100 nesne arasından rastgele seçilen 40 tanesi kullanılmıştır. Seçilen görüntüler Şekil 6.7'de gösterilmiştir.

Her bir nesnenin toplam 72 görüntüsü olduğu için, bunlardan rastgele seçilen 36 tanesinin eğitim verikümesini, geriye kalan 36 tanesinin ise test verikümesini oluşturduğu kabul edilmiş ve bölüm girişinde anlatılan sürece uygun olarak nesne tanıma deneyleri yapılmıştır. Önceki deneylerde olduğu gibi, veritabanındaki görüntüler gri seviyeli ve renkli olarak ayrı ayrı denenmiş ve sonuçları karşılaştırılmıştır. Gri

seviyeli görüntüler üzerinde yapılan deney sonuçları Çizelge 6.7’de, renkli görüntüler üzerinde yapılan deney sonuçları Çizelge 6.8’de verilmiştir.



Şekil 6.7 COIL100 veritabanından seçilen 40 nesne

Çizelge 6.7 COIL100 veritabanı gri seviyeli 40 nesne görüntüleri için deney sonuçları

Yöntem	r_1	r_2	p	Test Aşaması doğru tanıma oranları ortalaması (%)	Test Aşaması doğru tanıma oranları standart sapması
PCA	40	–	–	99,45	0,23
OVY	–	–	–	99,81	0,15
AOVY	–	–	–	93,28	0,76
2DPCA	11	–	–	99,26	0,23
2DLDA	10	–	–	98,94	0,26
2DCLAFIC	7	–	–	91,67	0,70
2DALS	9	–	–	95,51	2,13
2DCLAFIC- μ	2	–	–	94,88	0,50
TOBY(GSO)	–	–	2	99,81	0,15
TOBY(HOSVD)-1	5	15	2	94,58	0,85
TOBY(HOSVD)-2	5	15	2	94,99	0,75
MDA	15	15	–	97,71	0,68
TOBY(MDA)	10	15	–	98,58	0,40

Çizelge 6.8 COIL100 veritabanı renkli 40 nesne görüntüleri için deney sonuçları

Yöntem	r_1	r_2	r_3	p	Test Aşaması doğru tanıma oranları ortalaması (%)	Test Aşaması doğru tanıma oranları standart sapması
OVY	–	–	–	–	99,93	0,10
AOVY	–	–	–	–	99,03	0,34
TOBY(GSO)	–	–	–	2	99,93	0,10
TOBY(HOSVD)–1	5	15	1	2	98,82	0,41
TOBY(HOSVD)–2	5	15	1	2	98,81	0,41
MDA	15	15	–	–	99,33	0,32
TOBY(MDA)	10	15	2	–	99,65	0,24

COIL100 veritabanı gri seviyeli görüntü deneylerinde genelde tüm yöntemler başarılı tanıma oranları vermiştir. En yüksek doğru tanıma oranını (% 99,81) ortak vektör yaklaşımıyla birlikte TOBY(GSO) yöntemi vermiştir. Daha önceden de ifade edildiği gibi, $p = 2$ değeri için TOBY(GSO) yöntemi verimliliği ortak vektör yaklaşımı yöntemine denktir. Tanıma oranları bakımından bu yöntemleri PCA ve 2DPCA % 99'un üzerinde başarımlarla takip etmiştir. Sıralama 2DLDA, TOBY(MDA), MDA, 2DALS, TOBY(HOSVD)–2, 2DCLAFIC– μ , TOBY(HOSVD)–1, AOVY ve 2DCLAFIC biçiminde devam etmektedir. En düşük performansı veren 2DCLAFIC yöntemi bile % 91,67 başarımla sağlamıştır. Örnek sayısı COIL veritabanında fazla sayıda olduğu için, 2DCLAFIC yönteminin iteratif bir uyarlaması olan 2DALS yöntemi, tanıma başarımını oldukça iyileştirmiştir.

Deneyde, ortak vektör yaklaşımının başarısının yanı sıra bir başka dikkat çeken nokta, diğer deneylerde genelde en düşük tanıma oranlarını veren PCA ve 2DPCA yöntemlerinin, COIL100 veritabanında oldukça iyi sonuçlar vermesidir. Önerilen tensörel ortak bileşen yöntemlerinden TOBY(MDA), % 98,58 doğru tanıma oranıyla MDA'dan daha iyi sonuç vermiştir. TOBY(HOSVD) ise tek ortak bileşen tahmini ile % 94,58, iki ortak bileşen tahmini ile % 94,99 doğru tanıma gerçekleştirilmesine rağmen diğer yöntemlerin yanında düşük bir başarımla gerçekleştirmiş gibi gözükmektedir.

Renkli görüntü deneylerinde tanıma başarımı sıralaması, gri seviyeli görüntü deney sonuçlarının sıralamasına benzer biçimdedir. OVY ve TOBY(GSO) en yüksek tanıma oranlarını yakalarken TOBY(MDA), MDA, AOV ve TOBY(HOSVD) bu yöntemleri izlemektedir. Renkli görüntü deneylerinde göze çarpan önemli bir ayrıntı, tüm yöntemlerin tanıma başarısının gri seviyeli görüntü deneylerindeki kadar yüksek olmasıdır. Nesnelere ayırt etmekte renk bileşenleri önemli bir ayırt edici özelliktir. Bu yüzden, deneylerde en düşük tanıma oranına sahip olan TOBY(HOSVD)-2 yöntemi, gri seviyeli görüntülerde % 94,99 başarımla göstermişken renkli görüntüler için % 98,81 doğru tanıma oranını yakalamıştır. Renkli görüntü tanıma deneylerinde COIL100 veritabanı için tüm yöntemlerin oldukça başarılı tanıma oranları verildiği söylenebilir.

BÖLÜM 7

SONUÇ VE ÖNERİLER

Bu tez çalışmasında, görüntü tanıma uygulamalarında kullanılmak üzere sayısal görüntülerin orijinal matris veya daha yüksek dereceden tensör gösterimlerini bozmadan öznitelik çıkarıcı ve/veya sınıflandırıcı özellikleri olan yeni yöntemler önerilmiştir. Önerilen yöntemler, matris tabanlı yöntemler ve tensörel ortak bileşen yöntemleri başlıkları altında verilmiştir. Sayısal görüntülerin aydınlatma şiddeti değerleri gri seviyeli görüntüler için matris gösterimiyle, renkli görüntüler için 3'üncü dereceden tensör gösterimiyle ifade edilir. Görüntü tanıma amacıyla kullanılan birçok yöntem vektör temellerine dayandığından, sayısal görüntü aydınlatma şiddeti değerlerinin vektörel yöntemlerde kullanılabilmesi için, sayısal görüntü gösteriminin öncelikle vektöre dönüştürülme zorunluluğu vardır. Oysa ki, görüntüdeki her bir pikselin konumu önemlidir ve piksellerin konuşlanma biçimleri bütünde görüntüyü oluşturur. Bu yüzden, sayısal görüntü piksellerinin konumlarını değiştirmeden işlem yapacak olan yöntemlerin, vektörel yöntemlere göre, piksellerin konumlarını bozmadıkları için daha verimli tanıma başarımları gerçekleştirmesi beklenmiştir. Çeşitli yüz veritabanları ve COIL100 nesne veritabanı üzerinde yapılan görüntü tanıma deneylerinde, önerilen yöntemler genellikle başarılı sonuçlar vermiştir.

Tensörel ortak bileşen yöntemi fikrinin temel dayanağı ortak vektör yaklaşımıdır. Vektörel bir yöntem olan ortak vektör yönteminin multilineer cebir temelleri altında yüksek dereceden tensör verilerine genişletilmiş biçiminin geliştirilmesi hedeflenmiştir. Ortak vektör yönteminin, tensör cebirine doğal bir genişlemesi olan ilk yöntem, Gram-Schmidt birimdikleştirme temelli tensörel ortak bileşen yöntemidir. TOBY(GSO) yöntemi, ortak vektör yaklaşımının yetersiz veri durumuna uygun olan veriler üzerine, verinin tensör biçimini bozmadan uygulanabilir. TOBY(GSO) yöntemiyle bulunan tensörel ortak bileşenler, veri tensörlerini vektör biçimine dönüştürüp ortak vektör yaklaşımı kullanılarak bulunan ortak vektörlerin yeniden tensör biçimine dönüştürülmesiyle elde edilen ortak vektörlerin tensör

gösterimine eşittir. Bu yüzden, TOBY(GSO) yönteminin, görüntü tanıma uygulamalarında sınıflandırıcı uzaklığı olarak Frobenius norm kullanıldığı zaman doğru tanıma başarımı ile ortak vektör yaklaşımının doğru tanıma başarımları aynı sonucu verir. Aralarındaki fark, TOBY(GSO) yönteminde tüm işlemlerin yüksek dereceden tensör verileri üzerinde gerçekleştirilmesidir. İkinci dereceden tensör verisi biçiminde verilen sayısal görüntülerde, Frobenius norm yerine AMD uzaklık ölçüsü kullanılabilir. Sayısal görüntü tanıma deneylerinde, AMD uzaklık ölçüsü kullanan TOBY(GSO) yönteminin doğru tanıma başarısının, bazı veritabanlarında ortak vektör yaklaşımının doğru tanıma başarısından daha yüksek olduğu gözlemlenmiştir. AMD uzaklık ölçüsünün kullanılabilmesi için verinin matris biçimi bozulmamalıdır. Bu yüzden, AMD uzaklık ölçüsü sadece TOBY(GSO) yöntemiyle birlikte kullanılabilir.

TOBY(GSO) yönteminin görüntü tanıma uygulamalarındaki başarısı, AMD uzaklığı kullanılmadığında, ortak vektör yaklaşımının tanıma başarısına eşit olduğu için, sayısal görüntü tanıma uygulamalarında, OVY'den daha başarılı sonuçlar verecek olan yöntemler araştırılmaya devam edilmiştir. TOBY(HOSVD), tez çalışmasında geliştirilen ve görüntü tanıma başarısı OVY'den genellikle daha yüksek olan bir yöntemdir. TOBY(HOSVD), üçüncü bölümde verilen ortak matris-2 yönteminin yüksek dereceden tensör verilerine genişletilmiş biçimidir. Yüksek dereceden tekil değer ayrıştırması temeline dayanan yöntem, sayısal görüntü tanıma deneylerinde başarılı sonuçlar vermiştir. TOBY(HOSVD) de ortak bileşenler 3 farklı varsayım üzerinden tahmin edilebilir. Birinci varsayım, yaklaşık olarak fark altuzayını belirlemekten geçer ve fark altuzayı üzerinden ortak bileşen tahmini yapılabilir. İkinci varsayım yaklaşık olarak faksızlık altuzayını belirleyip farksızlık altuzayı üzerinden ortak bileşen tahmini üzerinedir. Üçüncü varsayım ise yaklaşık olarak fark ve farkızlık altuzaylarını belirleyip, her iki altuzayı kullanarak ortak bileşen tahmini yapmaya dayanır. Bu tahminler arasından sadece fark altuzayı üzerinden yapılan tahmin ve her iki altuzayı kullanarak yapılan tahmin, sadece farksızlık altuzayı kullanarak yapılan tahminden daha başarılı sonuçlar vermişlerdir. Bu yüzden, TOBY(HOSVD) yönteminde, sadece fark altuzayı üzerinden veya her iki altuzayı üzerinden yapılan ortak bileşen tahmininin kullanılması önerilir.

TOBY(HOSVD) yöntemi, sadece sınıf içi saçılımları değerlendirerek işlem yapan bir yöntemdir. Görüntü tanıma verimliliği genelde başarılı olan yöntemin tanıma başarımını arttırabilmek için özniteliklerin sınıf içi ve sınıflararası saçılımlarına bakan MDA yöntemi temel alınarak, TOBY(HOSVD) ile tahmin edilen ortak bileşenler üzerinde sınıflar arası ortak bileşen saçılımını enbüyükleyecek değişiklikler yapılmış ve tensör diskriminant analizi temeline dayanan ortak bileşen yöntemi geliştirilmiştir. TOBY(MDA), Yale, ORL ve COIL100 veritabanlarında TOBY(HOSVD) verimliliğini biraz yükseltmiştir. Sayısal görüntü tanıma deney sonuçlarına bakıldığında, TOBY(MDA) yönteminin de, vektörel yöntemlerle kıyaslandığında başarılı bir yöntem olduğu söylenebilir.

Tez çalışmasında önerilen ortak bileşen yöntemlerinden TOBY(GSO)'da her bir sınıf için üretilen ortak bileşen tektir. Aynı sınıfa ait olan örneklerin ortak bileşen uzayına olan izdüşümleri aynı ortak bileşeni verir. Ortak bileşen tensörünün derecesi ne olursa olsun, yöntem başarıyla çalışabilir. Birinci dereceden tensör veri örnekleri söz konusu ise yöntemin başarımı ortak vektör yaklaşımına denktir. Yöntem ayrıca, daha yüksek dereceden tensör verisi örneklerine de doğrudan uygulanabilir. Sınıf ortak bileşeni, sınıf özniteliği olarak kabul edilir. Sınıflandırma aşamasında test örneğinin ortak bileşen uzayına olan izdüşümü ile sınıf ortak bileşenleri karşılaştırılır, test örneği, test ortak bileşeni hangi sınıfın ortak bileşenine daha yakınsa o sınıfa atanır.

TOBY(HOSVD) yöntemi, TOBY(GSO) yönteminden farklı teorilere sahiptir. TOBY(HOSVD) yönteminde, aynı sınıfa ait olan veri örneklerinin ortak bileşen uzayına izdüşümleri farklı birer öznitelik verir. Bu özelliği sayesinde, yöntem bir öznitelik çıkarıcı yöntem gibi kullanılabilir. Sınıf ortak bileşeni, en küçük kareler kestiricisine göre ortak bileşen uzayına alınan izdüşümlerin ortalaması olarak kabul edilir. Ortak bileşen uzayı da yaklaşık olarak fark altuzayı belirlenerek, yaklaşık olarak farksızlık altuzayı belirlenerek veya her iki altuzay üzerinden yapılan tahminlerin ortalaması alınarak gerçekleştirilebilir. Ortak bileşen tahmini yapıldıktan sonra, sınıflandırıcı aşaması TOBY(GSO) yöntemine benzer biçimde gerçekleştirilir. Test vektörünün ortak bileşen tahmini yapılır ve sınıf ortak bileşen tahminleriyle karşılaştırılır. Test vektörünün ortak bileşen tahmini hangi sınıfın ortak bileşenine daha yakınsa, ilgili sınıf

kararı verilir. Tez çalışmasında, TOBY(HOSVD) yönteminin sınıflandırıcıyla birlikte denemeleri yapılmıştır. Yöntemin sadece öznelik çıkarıcı aşaması uygulanarak başka sınıflandırıcılarla birlikte kullanılabilir. TOBY(HOSVD) yönteminin, sınıflandırıcı verimliliği yüksek olan sınıflandırıcılarla birlikte kullanıldığında, görüntü tanıma uygulamalarında tanıma başarısını arttıracığı beklenebilir.

TOBY(MDA) yöntemi, TOBY(HOSVD) yönteminde çıkarılan özneliklerin, bir tensör diskriminant yöntemi olan MDA algoritması temel alınarak sınıflar arası ayırt edici özelliklerinin güçlendirildiği, diskriminant temelli bir tensör ortak bileşen yöntemidir. TOBY(MDA) yöntemi, iteratif bir algoritmaya sahiptir ve en iyi izdüşümü bulmayı hedefler. Bu yöntemin sınıflandırıcısı, en yakın komşu sınıflandırıcısıdır. Deneysel sonuçlara bakıldığında, ORL, Yale ve COIL100 veritabanlarında TOBY(MDA) yönteminin MDA yönteminden daha yüksek tanıma oranları verdiği gözlemlenmiştir. Yöntem, aynı veritabanlarında TOBY(HOSVD) yönteminden de daha başarılı sonuçlar vermiştir.

Deneysel sonuçlar göz önünde bulundurularak yöntemler karşılaştırıldığında, önerilen tensörel yöntemlerin genelde başarılı doğru tanıma oranları verdiği söylenebilir. Klasik yöntemlerden PCA yöntemi, COIL100 veritabanı hariç, diğer tüm veritabanlarında doğru tanıma oranları bakımından sıralamada son sıralarda yer almıştır. Matris tabanlı yöntemlerden 2DPCA de genellikle PCA'den biraz daha yüksek tanıma oranları vermesine rağmen, sıralamada PCA yöntemine yakın sıralardadır. Matris tabanlı yöntemler arasında önerilen iki boyutlu altuzay sınıflandırıcılarının doğru tanıma başarımlarının da genellikle yüksek olduğu söylenemez. Ancak bu yöntemlerin kovaryans hesabı vektörel ve tensörel yöntemlere göre daha az işlemsel güç gerektirdiğinden, iki boyutlu yöntemler diğer yöntemlerden daha kısa sürede tanıma gerçekleştirebilir. İki boyutlu altuzay sınıflandırıcılarından 2DCLAFIC- μ yönteminin tanıma başarısı genellikle 2DCLAFIC ve 2DCLAFIC'in iteratif uyarlaması olan 2DALS yönteminin tanıma oranlarından daha yüksektir. İki boyutlu yöntemlerden 2DLDA ise matris tabanlı yöntemler arasında en başarılı tanıma oranını veren yöntemdir. Ortak vektör yaklaşımı, COIL100 veritabanında TOBY(GSO) ile birlikte en başarılı tanıma oranını veren yöntemdir. Diğer veritabanlarında ise önerilen tensör

yöntemlerden daha düşük tanıma oranları vermiştir. Önerilen yöntemlerden TOBY(GSO) yönteminin doğru tanıma başarısı, her durumda ortak vektör yaklaşımının doğru tanıma başarısına eşit veya daha yüksektir. TOBY(HOSVD) ve TOBY(MDA) yöntemleri ise COIL100 veritabanı hariç diğer veritabanlarının tümünde OVY'den daha başarılı doğru tanıma oranları vermiştir. AOVY, Yale ve AR veritabanlarında oldukça başarılı tanıma oranları vermesine rağmen, ORL ve COIL100 veritabanlarında önerilen yöntemlerden daha düşük doğru tanıma oranları vermiştir. Yale veritabanı deneylerinde, AOVY yönteminin başarısı kesindir. ORL veritabanı için TOBY(MDA), TOBY(GSO) ve TOBY(HOSVD)-1, hem OVY, hem de AOVY yöntemlerinden daha yüksek doğru tanıma oranlarına sahiptir. AR veritabanında en başarılı tanıma oranını AOVY sağlamasına rağmen ikinci sırada çok küçük bir farkla TOBY(HOSVD) takip eder. COIL100 veritabanında ise AOVY yönteminin doğru tanıma başarısı çok yüksek değildir. Bu yüzden, özellikle OVY ve AOVY yöntemlerinin doğru tanıma başarısıyla karşılaştırıldığında, önerilen tensör ortak bileşen yöntemlerinin genellikle başarılı sonuçlar verdiği söylenebilir.

Sonuç olarak, tez çalışmasında, sayısal görüntü tanıma uygulamalarında başarılı sonuçlar veren tensörel ortak bileşen yöntemleri önerilmiştir. Önerilen yöntemler yüksek dereceden tensör verisi örneklerine doğrudan uygulanabilir. Verinin yerleşke yapısını bozmadan işlem yapan yöntemlerin uygulaması Yale, ORL ve AR yüz veritabanları üzerinde yapılan yüz tanıma deneyleriyle ve COIL100 veritabanı üzerinde yapılan nesne tanıma deneyleriyle test edilmiş ve genelde yüksek doğru tanıma oranları elde edilmiştir. Tensörel verilere doğrudan uygulanabilen yöntem geliştirme çalışmaları oldukça yeni bir araştırma konusudur. Gelişen teknoloji ile birlikte, veri ifade biçimlerinin yüksek dereceden tensörler biçiminde gösteriminin de yaygınlaşacağı tahmin edilerek, önerilen yöntemlerin sadece görüntü tanıma alanında değil, farklı görüntü tanıma alanlarında da kullanılabileceği düşünülmektedir.

KAYNAKLAR DİZİNİ

- Bader, B.W. and Kolda, T.G., 2006, Algorithm 862: MATLAB tensor classes for fast algorithm prototyping, *ACM Transactions on Mathematical Software* 32, 4, 635–653.
- Belhumeur, P.N., Hespanha, J.P. and Kriegman, D.J., 1997, Eigenfaces vs. Fisherfaces: recognition using class specific linear projection, *IEEE Trans. Pattern Analysis and Machine Intelligence* 19, 7, 711-720.
- Burges, C.J.C., 2004, Geometric methods for feature extraction and dimensional reduction: a guided tour, Technical Report, 34 p. (unpublished).
- Cevikalp, H., Wilkes, M. and Barkana, A., 2005, Discriminative common vectors for face recognition, *IEEE Trans. Pattern Analysis and Machine Intelligence* 27, 1, 1-10.
- Cevikalp, H., Yavuz, H.S. and Barkana, A., 2006, Two dimensional (2D) subspace classifiers for image recognition, in proceedings of the European Signal Processing Conference EUSIPCO2006, Italy, 4 p.
- Cevikalp, H., Yavuz, H.S., Cay, M.A. and Barkana, A., 2008, Two-dimensional subspace classifiers for face recognition, *Neurocomputing*, doi:10.1016/j.neucom.2008.02.015. (in press)
- Chen, S., Zhu, Y., Zhang, D. and Yang, J.Y., 2005, Feature extraction approaches based on matrix pattern: MatPCA and MatFLDA, *Pattern Recognition Letters* 26, 8, 1157-1167.

KAYNAKLAR DİZİNİ (devam ediyor)

- Çokar, O., 2000, Ortak vektörün iki boyutlu örüntü tanımada kullanımı, Yüksek Lisans tezi, Eskişehir Osmangazi Üniversitesi, Fen Bilimleri Enstitüsü, 20 s. (yayımlanmamış).
- Demirkaya, O., 2004, Örüntü tanımada ortak vektör yaklaşımının kullanılması, Yüksek Lisans tezi, Eskişehir Osmangazi Üniversitesi, Fen Bilimleri Enstitüsü, 75 s. (yayımlanmamış).
- Duda, R.O., Hart, P.E. and Stork, D.G., 2001, Pattern classification, J. Wiley, New York Inc., 654 p.
- Fukunaga, K., 1990, Introduction to statistical pattern recognition, Boston : Academic Press, 591 p.
- Gulmezoglu, M.B., Dzhafarov, V., Keskin M. and Barkana, A., 1999, A novel approach to isolated word recognition, IEEE Trans. Speech and Audio Proc. 7, 6, 620-628.
- Gulmezoglu, M.B., Dzhafarov, V. and Barkana, A., 2001, The common vector approach and its relation to principal component analysis, IEEE Trans. Speech and Audio Proc. 9, 6, 655-662.
- Gulmezoglu, M.B., Dzhafarov, V., Edizkan, R., Barkana, A., 2007, The common vector approach and its comparison with other subspace methods in case of sufficient data, Computer Speech and Language 21, 266-281.
- Günel, S., 2003, Ortak vektör yaklaşımı yöntemiyle TI TMS320C6711 DSK platformunda konuşmacıdan bağımsız gerçek zamanlı rakam tanıma, Yüksek Lisans tezi, Eskişehir Osmangazi Üniversitesi, Fen Bilimleri Enstitüsü, 102 s. (yayımlanmamış)

KAYNAKLAR DİZİNİ (devam ediyor)

- Jing, X.Y., Wong, H.S. and Zhang, D., 2006 Face recognition based on 2D Fisherface approach, *Pattern Recognition* 39, 4, 707-710.
- Kong, H., Li, X., Wang, L., Teoh, E.K., Wang, J. and Venkateswarlu, R., 2005, Generalized 2D principal component analysis”, in proceedings of IEEE International Joint Conference on Neural Networks IJCNN'05, 1, 108-113.
- Laaksonen, J., 1997, Subspace classifiers in recognition of handwritten digits, Ph. D. thesis, Helsinki University of Technology, (unpublished).
- Lathauwer, L.D., Moor, B.D. and Vandewalle, J., 2004, Computation of the canonical decomposition by means of a simultaneous generalized Schur decomposition, *Siam J. Matrix Appl.* 26, 2, 295-327.
- Lathauwer, L.D., Moor, B.D. and Vandewalle, J., 2000, A multilinear singular value decomposition, *Siam Journal of Matrix Analysis and Applications* 21, 4, 1253-1278.
- Li, M. and Yuan, B., 2005, 2D-LDA: A statistical linear discriminant analysis for image matrix, *Pattern Recognition Letters* 26, 5, 527-532.
- Lu, H., Plataniotis, K.N. and Venetsanopoulos, A.N., 2006, Multilinear principal component analysis of tensor objects for recognition, in proceedings of 18th International Conference on Pattern Recognition ICPR'06.
- Martinez, A.M. and Benavente, R., The AR face database, CVC Technical Report #24, 1998.

KAYNAKLAR DİZİNİ (devam ediyor)

- Moon, T.K. and Stirling, W.C., 2000, Mathematical methods and algorithms for signal processing, Prentice Hall, New Jersey, 937 p.
- Nene, S.A., Nayar S.K. and Murase, H., Columbia object image library (COIL-100), Technical Report CUCS-006-96, Columbia University, 1996.
- Oja, E., 1983, Subspace methods of pattern recognition. Research Studies Press, 200 p.
- Turhal, U.C., Gulmezoglu M.B. and Barkana, A., 2005, Face recognition using common matrix approach, in proceedings of 13th European Signal Processing Conference, EUSIPCO 2005.
- Turk, M. And Pentland, A.P., 1991, Eigenfaces for recognition, J. Cognitive Neuroscience 3, 1, 71-86.
- Vasilescu, M.A.O. and Terzopoulos, D., 2002, Multilinear analysis of image ensembles: TensorFaces, in proceedings of the European Conference on Computer Vision ECCV'02, 447–460.
- Yan, S., Xu, D., Yang, Q., Zhang, L., Tang, X. and Zhang, H.J., 2007, Multilinear discriminant analysis for face recognition, IEEE Trans. On Image Processing 16, 1, 212-220.
- Yang, J. and Yang, J.-Y., 2002, From image vector to matrix: a straightforward image projection technique – IMPCA vs. PCA, Pattern Recognition 35, 9, 1997–1999.
- Yang, J., Zhang, D., Frangi, A.F. and Yang, J.-Y., 2004, Two-dimensional PCA: a new approach to appearance–based face representation and recognition,” IEEE Transactions on Pattern Analysis and Machine Intelligence 26, 1, 131-137.

KAYNAKLAR DİZİNİ (devam ediyor)

Yavuz, H. S., Cevikalp, H. and Barkana, A., 2006, Görüntü tanımada iki boyutlu CLAFIC yöntemleri, IEEE 14. Sinyal İşleme, İletişim ve Uygulamaları Kurultayı, 17-19 Nisan 2006, 4 p.

Zhang, D. and Zhou, Z.H., 2005, (2D)2PCA: Two-directional two-dimensional PCA for efficient face representation and recognition, Neurocomputing 69, 224-231.

Zuo, W., Zhang, D. and Wang, K., 2006, An assembled matrix distance metric for 2DPCA – based image recognition, Pattern Recognition Letters 27, 3, 210-216.

<http://cvc.yale.edu/projects/yalefaces/yalefaces.html>

<http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html>

http://cobweb.ecn.purdue.edu/~aleix/alexi_face_DB.html

<http://www1.cs.columbia.edu/CAVE/software/softlib/coil-100.php>

ÖZGEÇMİŞ

1978 yılında Eskişehir’de doğan Hasan Serhan Yavuz, ilk ve orta öğrenimini Eskişehir’de sırasıyla Murat Atılğan İlkokulu, Atatürk Ortaokulu ve Atatürk Lisesi’nde tamamladıktan sonra, 1999 yılında Eskişehir Osmangazi Üniversitesi Elektrik-Elektronik Mühendisliği Bölümü’nden mezun olmuştur. Aynı sene Eskişehir Osmangazi Üniversitesi Elektrik-Elektronik Mühendisliği Bölümü’nde araştırma görevlisi olarak akademik hayata başlamış olup, yüksek lisans derecesini 2002 yılında Eskişehir Osmangazi Üniversitesi Fen Bilimleri Enstitüsü Elektronik Anabilim Dalı’ndan almıştır.