

**Çok Yüzlü Konik Fonksiyonlar Temelli  
Sınıflandırma Yaklaşımları ile Hareket Tanıma**

Emre Çimen  
Yüksek Lisans Tezi

Endüstri Mühendisliği Ana Bilim Dalı

Ağustos 2013

**Bu tez çalışması Anadolu Üniversitesi Bilimsel Araştırma Projeleri  
Komisyonu Başkanlığı tarafından desteklenmiştir. Proje No: 1304F063**

## JÜRİ VE ENSTİTÜ ONAYI

**Emre Çimen'in "Çok Yüzlü Konik Fonksiyonlar Temelli Sınıflandırma Yaklaşımları ile Hareket Tanıma "** başlıklı **Endüstri Mühendisliği** Anabilim Dalındaki, Yüksek Lisans Tezi 06.08.2013 tarihinde, aşağıdaki jüri tarafından Anadolu Üniversitesi Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliğinin ilgili maddeleri uyarınca değerlendirilerek kabul edilmiştir.

	<u>Adı Soyadı</u>	<u>İmza</u>
Üye (Tez Danışmanı) :	Yard. Doç Dr. GÜRKAN ÖZTÜRK	.....
Üye :	Prof. Dr. REFAİL KASIMBEYLİ	.....
Üye :	Doç. Dr. ADİL BAGİROV	.....

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

Enstitü Müdür

## ÖZET

Yüksek Lisans Tezi

### ÇOK YÜZLÜ KONİK FONKSİYONLAR TEMELLİ SINIFLANDIRMA YAKLAŞIMLARI İLE HAREKET TANIMA

Emre ÇİMEN

Anadolu Üniversitesi

Fen Bilimleri Enstitüsü

Endüstri Mühendisliği Anabilim Dalı

Danışman: Yard. Doç. Dr. Gürkan ÖZTÜRK

2013, 66 Sayfa

İnsan vücut hareketlerinin algılanması ve anlamlandırılması ile makinelere komut verme, özellikle cihazları uzaktan kontrol etmek için kullanılan önemli bir teknoloji haline dönüşmektedir. Bu çalışmada kümeleme temelli Çok Yüzlü Konik Fonksiyonlar algoritmaları ile sınıflandırıcıların oluşturulması, özellikle görüntü üzerinde işlem yapılan problemler için büyük öneme sahip öznelik seçimi yaklaşımlarının incelenmesi ve yeni bir yaklaşım önerilmesi, son olarak da Çok Yüzlü Konik Fonksiyonlar temelli algoritmaların hareket tanıma problemine uygulanması gerçekleştirilmiştir. Uygulama aşamasında hareket tanıma için kullanılan veri kümeleri kameradan alınan görüntüler kullanılarak oluşturulmuştur. Elde edilen veriler kullanılarak çok yüzlü konik fonksiyonlar temelli sınıflandırıcılar elde edilmiş ve uygulama probleminin çözümünde başarı ile kullanılmıştır.

**Anahtar Kelimeler:** Matematiksel Programlama, Veri Madencilği, Sınıflandırma, Hareket Tanıma, Öznelik Seçimi

## **ABSTRACT**

**Master of Science Thesis**

### **GESTURE RECOGNITION WITH POLYHEDRAL CONIC FUNCTIONS BASED CLASSIFIERS**

**Emre ÇİMEN**

**Anadolu University**

**Graduate School of Sciences**

**Industrial Engineering Program**

**Supervisor: Assoc. Prof. Dr. Gürkan ÖZTÜRK**

**2013, 66 pages**

Controlling a machine via detecting and recognising human gestures, becomes important technology especially using for remote controlling a device. In this study we carried about generating clustering based polyhedral conic functions algorithms, investigating and proposing a new method for the feature selection problem, which is very important for the applications like video processing, and finally implementing generated polyhedral conic functions based algorithms to the gesture recognition problem. In implementation step data needed for the classifiers are collected from the camera. The data obtained are used for generating classifiers based polyhedral conic functions and these classifiers are used successfully in the application.

**Keywords:** Mathematical programming, Data Mining, Classification, Gesture Recognition, Feature Selection

## TEŞEKKÜR

Öğrencisi olduğumdan dolayı kendimi çok şanslı hissettiren, benim daha iyi bir insan ve ileride başarılı bir akademisyen olmam için büyük emekler sarfeden, sadece hocam olarak nitelendiremeyeceğim Sayın Yrd.Doç.Dr.Gürkan ÖZTÜRK'e;

Bir tek tez çalışması konusunda değil birçok alanda bana yol gösteren, birlikte çalışmaktan büyük zevk aldığım ve bana çok şey katan Sayın Prof.Dr. Ömer Nezih GEREK 'e; sadece başarılarıyla değil insani özellikleriyle de her açıdan örnek aldığım değerli hocalarım Sayın Prof.Dr. Refail KASIMBEYLİ ve Sayın Doç.Dr.Adil BAGİROV 'a;

Benden desteklerini esirgemeyen, her birini ailemden bir fert olarak saydığım arkadaşlarım ve hocalarıma;

Son olarak beni ülkemize hayırlı bir insan olarak yetiştirmek için ellerinden geleni yapan, beni bu günlere getiren annem Nermin ÇİMEN ve babam Mecit ÇİMEN'e;

Teşekkürü bir borç bilir, sevgilerimi sunarım ...

## İÇİNDEKİLER

ÖZET.....	i
ABSTRACT .....	ii
TEŞEKKÜR .....	iii
İÇİNDEKİLER .....	iv
ŞEKİLLER DİZİNİ .....	vi
ÇİZELGELER DİZİNİ .....	viii

1. GİRİŞ.....	1
---------------	---

## 2. SINIFLANDIRMA PROBLEMLERİNİN ÇÖZÜMÜ İÇİN

KÜMELEME TEMELLİ YAKLAŞIMLAR.....	4
2.1. Giriş .....	4
2.2. Literatür Özeti .....	5
2.3. Sınıflandırma Problemlerinin Çözümünde Kullanılan Yöntemler .....	7
2.4. Kümeleme .....	8
2.4.1. <i>K</i> -ortalamalar algoritması .....	9
2.4.2. <i>EM</i> algoritması .....	11
2.4.3. Geliştirilmiş bütünsel <i>k</i> -ortalamalar algoritması .....	16
2.5. Çok Yüzlü Konik Fonksiyonlar ile Sınıflandırma .....	17
2.5.1. Çok yüzlü konik fonksiyonlar algoritması .....	18
2.5.2. Kümeleme temelli sınıflandırma yaklaşımları .....	19
2.6. Hesapsal Sonuçlar .....	26
2.7. Sonuçlar .....	28

3. ÖZNETELİK SEÇİMİ İÇİN YENİ BİR YAKLAŞIM .....	30
--	----

3.1. Giriş .....	30
3.2. Literatür Özeti .....	33
3.3. Öznitelik Seçimi Yöntemleri .....	34
3.4. Önerilen Yaklaşım .....	41
3.5. Hesapsal Sonuçlar .....	44
3.6. Sonuçlar .....	45

<b>4. HAREKET TANIMA .....</b>	<b>46</b>
4.1. Giriş .....	46
4.2. Literatür Özeti .....	46
4.3. Hareket Tanımada Kullanılan Yöntem ve Cihazlar .....	51
4.4. Matematiksel Programlama ile Hareket Tanıma Üzerine Bir Uygulama .....	54
4.4.1. Elin bulunması .....	54
4.4.2. Hareket vektörlerinin çıkarılması ve anlamlandırılması .....	56
4.5. Hesapsal Sonuçlar .....	57
4.6. Sonuçlar .....	59
<b>5. SONUÇ VE ÖNERİLER .....</b>	<b>61</b>
<b>KAYNAKÇA .....</b>	<b>62</b>

## ŞEKİLLER DİZİNİ

2.1 Bir kümeleme örneği.....	5
2.2 Voronoi Diyagramı .....	10
2.3 $k$ -Ortalamalar kümeleme yöntemi adımları .....	10
2.4 Gerçek kümeler ve $k$ -ortalamalar algoritmasıyla elde edilen kümeler .....	11
2.5 Karma normal dağılım .....	12
2.6 İki boyutta karma normal dağılım.....	12
2.7 $EM$ algoritması adımları.....	15
2.8 İki boyutlu açıklayıcı örnek dağılımı .....	22
2.9 $EM$ -ÇKF algoritması ile yeşil sınıfı ayıran fonksiyondan bir görünüm .....	22
2.10 $EM$ -ÇKF algoritması ile yeşil sınıfı ayıran fonksiyondan bir görünüm .....	23
2.11 $EM$ -ÇKF algoritması ile yeşil sınıfı ayıran fonksiyondan bir görünüm .....	23
2.12 $GB$ $k$ -ort-ÇKF algoritması ile yeşil bir kümeyi ayıran fonksiyon.....	23
2.13 $GB$ $k$ -ort-ÇKF algoritması ile yeşil bir kümeyi ayıran fonksiyon.....	24
2.14 $k$ -ort-ÇKF algoritması ile yeşil sınıfı ayıran fonksiyondan bir görünüm.....	24
2.15 $k$ -ort-ÇKF algoritması ile yeşil sınıfı ayıran fonksiyondan bir görünüm.....	25
2.16 $k$ -ort-ÇKF algoritması ile yeşil sınıfı ayıran fonksiyondan bir görünüm.....	25
2.17 $k$ -ort-ÇKF algoritması ile mor sınıfı ayıran fonksiyondan bir görünüm.....	25
2.18 Farklı küme sayılarına göre sınıflandırma başarıları .....	27
3.1 Levrek ve somon balıkları için iki öznitelik .....	31
3.2 En ve boy özniteliklerinin dağılım grafikleri .....	31
3.3 Öznitelik dağılımlarına iki öznitelik .....	32
3.4 AdaBoost tarafından seçilen iki öznitelik .....	37
3.5 Ayrılması amaçlanan iki küme .....	39
3.6 İlk adımda ayırıcı zayıf sınıflandırıcının seçimi .....	39
3.7 Ağırlıkların değişmesi ve ikinci zayıf sınıflandırıcı seçimi .....	40
3.8 Ağırlıkların değişmesi ve üçüncü zayıf sınıflandırıcı seçimi.....	40
3.9 Ağırlıkların değişmesi ve dördüncü zayıf sınıflandırıcı seçimi .....	40
3.10 Ağırlıkların değişmesi ve beşinci zayıf sınıflandırıcı seçimi.....	41
3.11 Son sınıflandırıcının oluşturulması .....	41
3.12 Seçilen sınıflandırıcıya göre doğru ve yanlış ayrılan nesnelere.....	43
4.1 HOG öznitelikleri .....	47



4.2 Dardas ve Georganas tarafından kullanılan hareketler .....	49
4.3 Sarrafzadeh ve ark. tarafından kullanılan hareketler.....	49
4.4 Top yakalama uygulamasından bir görüntü .....	51
4.5 <i>toF</i> kamera ve bu kameradan elde edilmiş bir görüntü.....	52
4.6 Nasa tarafından kullanılan bir stereo kamera.....	52
4.7 <i>Wii</i> uzaktan kumanda .....	53
4.8 Farklı <i>Y</i> değerlerine göre <i>YCbCr</i> renk uzayı .....	55
4.9 Renk ile ayırma, orjinal görüntü ve hareket ile ayırma.....	55
4.10 Çıkarılan bir hareket vektörü .....	56
4.11 Halka hareketinin tanınması.....	56

## ÇİZELGELER DİZİNİ

2.1 Iris, wine, glass ve vehicle veri setleri sınıflandırma doğrulukları .....	27
2.2 Coverttype veri seti sınıflandırma doğrulukları .....	27
2.3 Bölüm 2.5.2 açıklayıcı örnek sınıflandırma doğrulukları .....	28
3.1 Literatürde farklı algoritmalar için sonuçlar .....	44
4.1 El bulmada kullanılan yöntemler .....	48
4.2 Lie ve ark. tarafından kullanılan hareketler ve tanıma oranları .....	50
4.3 Halka hareketi test başarıları .....	57
4.4 İki sınıflı bir problem için hatalı sınıflandırma matrisi .....	58
4.5 Halka hareketi için kesinlik değerleri .....	58
4.6 Halka hareketi için duyarlılık değerleri.....	59
4.7 Yöntemler için hesaplanan $f$ değerleri.....	59

## 1. GİRİŞ

Bir cihazı uzaktan kontrol etmek bilim insanlarının çok uzun zamandır üzerinde çalıştığı bir alandır ve bu konuda bugüne kadar değişik çalışma prensiplerine sahip birçok kontrol cihazı geliştirilmiştir. İnsan doğası sürekli olarak işlerini kolaylaştırmak eğilimindedir. Günümüz teknolojisi basitleşme, kullanıcılara en üst düzeyde kullanım rahatlığı sağlama yolunda ilerlemektedir. Fütüristler teknolojinin gelişimi konusunda tahminler yürütürlerken elle rahatça kontrol edilebilen, bir hareketle kayan, açılan, kapanan dijital ekranlardan bahsetmektedirler. Bilim kurgu filmlerinde de yıllar sonrasının dünyası anlatılmak istendiğinde ya da çok gelişmiş bir milleti canlandırmaya çalışırken, onların gelişmişlikleri etraftaki sanal ekranların el-kol hareketleri ile kontrol edebilmesi ile vurgulanmaktadır. Akıllı evlerde insanların yerlerinden kalkmadan buzdolabı, fırın, çamaşır makinesi gibi cihazlarını kontrol edebilecek olması uzak bir gelecek değildir ve bu tip cihazların kontrolü için herhangi bir kontrol cihazına ihtiyaç duyulmayacaktır. Bu amaçlar ile günümüzde araştırmacılar daha etkin şekilde makineleri kontrol edebilmek için çeşitli insan-bilgisayar ara yüzleri (human-computer interface) üzerine çalışmalar yürütmektedir. Parmaklar, eller, kollar, baş ve yüz gibi insan vücut hareketlerinin vücuda takılan ya da elle tutulan bir cihaz olmaksızın; algılanması ve anlamlandırılması ile makinelere komut verme, özellikle önemli bir teknoloji haline dönüşmektedir. Teknolojinin ilerlemesi cihazı kontrol etme konusunda sadece kumandaya ya da düğmelere bağlı kalmaya gerek olmadığını göstermektedir. Özellikle Apple firmasının ürünlerinden klavyeyi çıkartarak öncülüğünü ettiği akılla beraber, mekanik tuş takımlarının kullanılması günden güne azalmaktadır.

Cumhuriyetimizin 100. yılı için Tübitak'ın öncülüğünde yapılan 2023 vizyon çalışmasında, "Türkiye'nin Bilim, teknoloji ve yenilikte yetkinleşmiş, üreten, net katma değerini kendi beyin gücüne dayanarak artırabilen bir ülke olması", ülkemizin vizyonlarından biri olarak benimsenmiştir. Bu vizyon doğrultusunda "belirli alanlarda Türkiye'nin rekabet üstünlüğü kazanarak uluslararası ticaretten ciddi bir pay alır hale gelmesi" hedeflenmektedir. Buna göre öncelik verilmesi gereken teknolojik faaliyet konuları belirlenmiş, bunların

arasında “Tüketici elektroniğinde yeni kuşak ürünler tasarlayıp üretebilmek” te yer almıştır. Yapılan çalışma sonucunda elde edilen sistemin farklı ürünlere entegre edilmesi halinde, bu ürünlerin piyasada büyük bir rekabet üstünlüğü sağlayacağı düşünülmektedir. Hareket ile kontrolü destekleyen ürünleri piyasaya süren öncü firmalar büyük avantajlar elde etmişlerdir. Kasım 2006’da Nintendo firmasının geliştirdiği, bir kumanda kullanarak kullanıcı hareketlerini tanıyabilen Wii oyun konsolu bu konudaki beklentileri arttırmıştır. Yine oyun konsolu pazarında 2010 yılında böyle bir teknolojiyi kullanmayı mümkün kılan bir cihaz geliştiren Microsoft firması pazar payını ciddi şekilde arttırmış ve cihazın piyasaya çıkışının ilk 25 gününde 2.5 milyon adet satmıştır. Benzer şekilde, çok yakın zamanda televizyonlarının *akıllı* unsurlarını hareket ile kontrol etmeye başlayan Samsung firması da ciddi ilgi uyandırmıştır.

Tez kapsamında temel olarak sınıflandırma probleminin çözümü üzerine çalışılmıştır. 1930’lu yıllarda Fisher’in doğrusal ayırma kavramını sunmasından bu yana sınıflandırma problemleri üzerine yoğun şekilde çalışılmaktadır. Sınıflandırma işlemi hangi sınıfa ait oldukları bilinen eldeki veriler ile bir sınıflandırıcının eğitilmesi ve daha sonra sınıfı bilinmeyen verilerin elde edilen sınıflandırıcı ile sınıf etiketlerinin belirlenmesi adımlarından oluşmaktadır. Sınıflandırma problemi hayatımızın bir çok alanında karşımıza çıkan önemli bir problemdir. Bu problemleri çözmek için geliştirilen yöntemler yüz tanıma, ses tanıma, hastalık teşhisi, e-posta filtreleme, kredi değerlendirme gibi alanlarda günlük hayatta uygulanır hale gelmiştir. Literatürde farklı yaklaşımlara dayanan bir çok sınıflandırma algoritması bulunmaktadır. Bayes sınıflandırma, karar ağaçları, sinir ağları ve destek vektör makineleri literatürde en çok karşılaşılan yöntemlerdir [1,2,3]. Bu çalışmada Gasimov ve Öztürk [4] tarafından ortaya atılmış olan çok yüzlü konik fonksiyonlar temelli sınıflandırıcılardan yararlanılmıştır. Yeni bir sınıflandırma yaklaşımı olarak Çiftçi tarafından [5]  $k$ -ortalamlar temelli çok yüzlü konik fonksiyonlar yaklaşımı önerilmiş ve büyük boyutlu problemler için önerilen yöntemin başarılı sonuçlar verdiği gösterilmiştir. Bu çalışmanın ikinci bölümünde, literatürde yaygın şekilde kullanılan beklenti maksimizasyonu ( $EM$ ) ve geliştirilmiş bütünsel  $k$ -ortalamlar kümeleme algoritmalarına dayanan yeni yaklaşımlar önerilmiştir. Bu yaklaşımlarda da  $k$ -ort

$\mathcal{CKF}$  yaklaşımına benzer şekilde, oluşturulacak  $\mathcal{CKF}$ 'lerin merkezleri, kümeleme algoritmalarının ( $EM$  ve geliştirilmiş bütünsel  $k$ -ortalamalar) bir sonucu olarak elde edilir. Yeni önerilen yaklaşımların başarısı bazı test problemleri üzerinden karşılaştırmalı sonuçlar ile beraber sunulmuştur.

Sınıflandırma problemleri için önemli bir sorun sınıflandırılacak verileri tanımlayan öznitelik uzayının çoğu zaman çok büyük olmasıdır. Elde var olan öznitelik uzayının bir kısmı sınıflandırma için anlamsız olabilir. Bu nedenle öznitelik uzayından uygun özniteliklerin seçilmesi sınıflandırma için oldukça önemli bir işlemdir. Bu kapsamda çok yüzlü konik fonksiyonlar kullanılarak *boosting* temeline dayanan bir öznitelik seçimi yöntemin geliştirilmesi üçüncü bölümde anlatılmıştır.

Dördüncü bölümde  $k$ -ortalamalar temelli çok yüzlü konik sınıflandırıcılar ile hareketlerin tanınması ele alınmıştır. Temel girdi bir kamera yardımıyla alınan görüntülerdir. Hareketlerin tanınabilmesi için hareket vektörlerinin sürekli olarak düzenli bir şekilde hesaplanması gerekmektedir. Bu nedenle tezin uygulamasında izlenen adımlar; elin bulunması, elin izlenmesi, hareket vektörünün çıkarılması, tanımlanan hareketler için sınıflandırıcıların oluşturulması ve hareketlerin tanınması olarak sıralanmıştır.

Bu çalışma ile ilgili genel değerlendirme ve gelecekte yapılabilecek çalışmalar ile ilgili açıklama ve tartışmalar, son bölüm olan sonuç ve öneriler bölümünde sunulmuştur.

## 2. SINIFLANDIRMA PROBLEMLERİNİN ÇÖZÜMÜ İÇİN KÜMELEME TEMELLİ YAKLAŞIMLAR

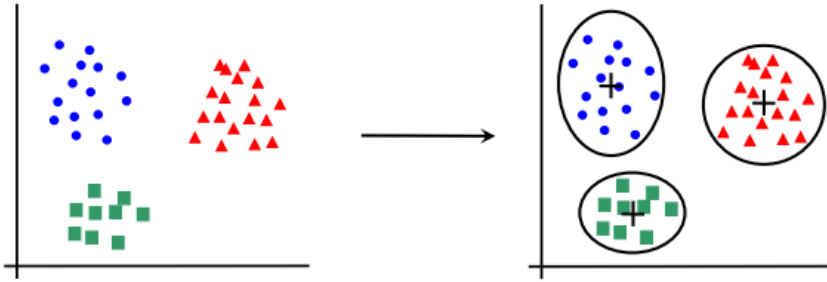
Bu bölümde genel olarak sınıflandırma problemi ve literatürdeki sınıflandırma yaklaşımlarına değinilmiştir. Ayrıca kullanılan üç kümeleme algoritması  $k$ -ortalamalar, geliştirilmiş bütünsel  $k$ -ortalamalar ve  $EM$  hakkında bilgiler verilmiştir. Daha sonra çok yüzlü konik fonksiyonlar temelli algoritmalar anlatılıp, sonuçlar sunulmuştur.

### 2.1.Giriş

Günümüzde verilerin elde edilmesi, saklanması ve ulaşılabilirliği teknolojideki gelişmelere paralel olarak kolaylaşmış ve ucuzlamıştır. Büyük boyutlarda ve hızlı bir şekilde toplanan verilerin çeşitli analizler sonucunda anlamlı bilgilere dönüştürülmesi süreci olarak tanımlanan veri madenciliğinin günümüzde en çok kullanılan teknikleri kümeleme ve sınıflandırma problemleridir [6].

Sınıflandırma, daha önceden karşılaşılmamış bir nesnenin öznitelik değerlerinin hesaplanması ve elde edilen değerlerin sınıflandırıcı bir fonksiyona sokularak nesnenin önceden tanımlanmış bir sınıfa atanmasıdır. Sınıflandırma işlemi kullanılan yöntem ne olursa olsun temel olarak iki adımdan oluşmaktadır. İlk adım karar verici kuralların-fonksiyonların elde edildiği eğitim adımıdır. Literatürde eğitim adımının denetimli (supervised) ve yarı denetimli (semi-supervised) gibi tipleri bulunmaktadır. Denetimli sınıflandırmada eğitimde kullanılacak verilerin öznitelik değerleri ve sınıfları bellidir. Yarı denetimli sınıflandırmada ise eğitimde kullanılacak verilerin sınıflarını elde etmek ya zor ya da maliyetli bir işlemdir. Dolayısıyla her sınıf etiketi belli değildir. Eğitim adımında sınıflandırıcı kural elde edildikten sonra test adımına geçilir. Testteki verilerin sınıf etiketinin bilinmediği varsayımı ile eğitimde elde edilen kurala göre bu verilerin gerçek sınıfları tahmin edilmeye çalışılır. Sınıflandırıcı ile doğru tahmin edilen sınıf sayısı, sınıflandırıcının başarısını gösteren doğruluk değerinin hesaplanmasında kullanılır.

Veri madenciliğinin uygulamada kullanılan en önemli kollarından birisi ise kümeleme analizidir. Uygulanması ve yorumlanması kullanılan algoritmaya göre değişmekle birlikte genel olarak kolaydır. Amacı var olan verilerin hangi kümeye ait olduğu bilinmeden, birbirine olan benzerliklerine bağlı olarak benzer özellikler gösteren verilerin bir araya toplanmasıdır. Kümeleme analizi temelde istatistik ve matematik bilimlerinden yararlanmaktadır. Makine öğrenmesi ve yapay zeka gibi konularda oldukça ihtiyaç duyulan bir analizdir.



Şekil 2.1 Bir kümeleme örneği [7]

Şekil 2.1’de 3 kümeden oluşan veriler ve bu verilerin kümelere ayırımı ile merkez noktaları gösterilmektedir.

## 2.2.Literatür Özeti

Literatürde sınıflandırma ve kümeleme problemlerinin çözümü için ortaya atılmış birçok yöntem bulunmaktadır. İzleyen bölümde en sık kullanılan yöntemlere değinilecektir. Bu bölümde literatürde bulunan bazı çalışmalar özetlenmiştir.

Sınıflandırma problemlerinin çözümü için geliştirilmiş yöntemlerin kullanılabilirliği sergiledikleri başarılarla, uygulanabildikleri problem boyutlarının esnek olmasına, uygulanabildikleri problem tiplerinin fazla olmasına, hızlarına ve işlemsel yüklerinin durumuna göre değişmektedir. Literatürde bahsedilen kriterlere göre başarılı olan ve en sık kullanılan yöntemler ayırma analizi, Bayes sınıflandırma, sinir ağları, karar ağaçları ve destek vektör makineleri olarak sayılabilir [1,2,3]. Bunların yanı sıra genetik algoritmalar, en yakın komşu ve bulanık mantık gibi yöntemler de mevcuttur. Sınıflandırma problemlerinin temeli 1930’ların ortalarında ortaya atılan Fisher’in doğrusal ayırma tekniğine dayanır.

1980'lerin ortalarında Rumerhart, Hintom ve Wiliams tarafından geri yayımlı sinir ağları bulunmuştur. İnsan sinir sistemi yapısından esinlenen yöntem, sinir sistemini matematiksel olarak modellemeyi amaçlamaktadır. Ortaya atılışından itibaren sinir ağları araştırmacılar tarafından büyük ilgi görmüş ve yaygın olarak kullanılmıştır. Literatürde en sık kullanılan ve en ünlü yöntemlerden biri Cortes ve Vapnik [8] tarafından oluşturulmuş Destek Vektör Makineleri olarak adlandırılan bir öğrenme tipidir.

1960'lı yıllardan itibaren matematiksel programlama temellerine dayanan birçok sınıflandırma algoritması ortaya atılmıştır. Dışbükey olmayan ve tam sayılı matematiksel programlama yöntemlerinden başarıyla uygulanan yöntemler olsa da literatürde var olan yöntemlerin büyük bir bölümü doğrusal programlama temellidir.

Doğrusal ayırma için Bennet ve Mangasarian gürbüz (robust) doğrusal programlama yaklaşımını geliştirmişlerdir [9]. Astorino ve Gaudioso [10] ise belirlenen sayıda hiper düzlem oluşturarak iki kümeyi birbirinden ayırmayı amaçlamışlar ve  $h$ -polihedral ayırma yaklaşımını sunmuşlardır. Bagirov [11] tarafından sunulan enb-enk ayırma yöntemi hiperdüzlemleri kullanan başarılı yöntemlerden bir diğeridir. Üney ve Türkay [12] çok sınıflı problemlerin çözümü için tüm sınıflara ait verileri ayıracak çok boyutlu kutuların kullanılmasına dayalı karma tamsayı matematiksel bir model geliştirmişlerdir.

Kümeleme analizi için yine çok geniş bir literatürden bahsetmek mümkündür. Farklı temellere dayanan birçok kümeleme algoritması bulunmaktadır. Genel olarak hiyerarşik yöntemler, ağırlık merkezi temelli yöntemler, yoğunluk temelli yöntemler ve dağılım temelli yöntemler olarak gruplanabilirler. Hiyerarşik kümeleme yöntemlerinden *SLINK* [13] Sibson tarafından 1973 yılında ortaya atılmıştır. Yine hiyerarşik ayırma yöntemlerinden *CLINK* [14] 1977 yılında Defays tarafından sunulmuştur. Bu iki yöntem araştırmacılar tarafından genel olarak görece eski yöntemler olarak nitelendirilseler de teoriler ile yoğunluk temeline dayanan kümeleme analizi yöntemlerine ilham kaynağı olmuşlardır. Kümeleme analizinde en ünlü yöntemlerden biri MacQuen [15] tarafından literatüre kazandırılmış ve  $k$ -ortalamlar olarak bilinen yöntemdir. Bu yöntem için Lloyd [16] tarafından pratik



bir algoritma geliştirilmiştir. Problem doğası gereği *NP-Zor* bir problemdir ve *k*-ortalamalar algoritması bu probleme yaklaşık bir sonuç sunar. Bagirov [17] 2008 yılında geliştirilmiş bütünsel *k*-ortalamalar algoritması ile başarılı sonuçlar elde etmiştir. Ayrıca literatürde *k*-ortalamalar algoritmasının *k*-ortalamalar++, *k*-medyanlar, bulanık *c*-ortalamalar gibi çeşitli varyasyonlarına rastlamak mümkündür. Dağılım temelli algoritmalar ise istatistik biliminden yoğun şekilde yararlanır. Bu tip algoritmaların en bilineni Dempster ve ark. [18] tarafından geliştirilen *EM* algoritmasıdır. Yoğunluk temelli algoritmalar kümeleri en fazla nokta yoğunluğunun bulunduğu yerlere atamaya çalışırlar. Ester ve ark [19] yoğunluk temelli algoritmalarından *DBSCAN*'i 1996 yılında başarılı bir şekilde uygulamışlardır. Ankerst ve ark. [20] tarafından 1999 yılında *OPTICS* adıyla *DBSCAN* algoritması genelleştirilmiştir.

### 2.3.Sınıflandırma Problemlerinin Çözümünde Kullanılan Yöntemler

Karar ağaçları, akış diyagramı bir ağaca benzeyen ve çok sık kullanılan sınıflandırma yöntemlerinden bir tanesidir. Hesapsal yükünün az olması, gürültülere karşı duyarlı olmaması ve gereksiz öznitelikler ile ilgilenmemesi nedenleriyle diğer sınıflandırma yöntemlerine göre avantaj sağlar [21]. Karar ağaçları yapısı gereği ağacın en üst noktasına en iyi ayırıcı olan özniteliği koyar. Öznitelikler düğüm noktalarında bulunur ve model bu özniteliğe dayanarak bir karar kuralı belirlemiştir. Geline düğümde test edilen verinin değerine göre gidilecek yön belirlenir. Yöntemin detayları Barros ve ark. 'nın [21] çalışmasında sunulmuştur.

Ayırma analizi sınıflandırma yöntemlerinden en eskilerinden birisidir. Verilerin öznitelikleri arasında bulunan farklılıkları araştırır. Ayırma analizinin amaçlarından biri en ayırıcı özniteliklere göre uygun ve iyi ayıran fonksiyonları bulmaktır. Diğer amaç ise verileri en iyi şekilde sınıflara atayabilecek kuralların çıkarılmasıdır. Bu noktada bu iki amacı karıştırmamak gerekmektedir [6].

Destek vektör makineleri ikili sınıflar için ayırıcı doğrusal yüzeyler elde etmek amacıyla geliştirilmiş bir yöntemdir. Kavramsal olarak girdi vektörü yüksek boyutlu bir uzaya taşıma fikri gerçekleştirilir. Bu yüksek boyutlu uzayda doğrusal ayırıcı yüzeyler elde edilir. *DVM* özel özellikleri sayesinde yüksek

genelleme yeteneğine sahiptir [8]. Çok başarılı ve yaygın kullanılır bir yöntemdir. Pradhan [22] 2012 yılında destek vektör makineleri ile ilgili bir genel inceleme (survey) makalesi yayınlamış ve bu konudaki son gelişmeler ile yöntemin temellerini aktarmıştır.

Yapay sinir ağları beyinden esinlenilmiş bir algılayıcıdır. Bilişsel bilim ve sinir bilim konusunda çalışan bilim insanları beynin nasıl çalıştığını anlamak amacıyla çalışmalar yapmaktadır. Bu amaç doğrultusunda beyindeki sinir ağlarının modelleri oluşturulur ve benzetim çalışmaları yapılır. Beyin, görüntü ve konuşma tanıma, öğrenme gibi yetenekleri açısından mühendislik ürünlerinden çok üstündür [23]. Yapay sinir ağları ile beynin matematiksel olarak modellenerek bu üstünlüklerin makinelere aktarılması için çalışılmaktadır.

Bayes sınıflandırmanın temeli aşağıdaki denklemde de gösterilen Thomas Bayes'in ortaya attığı Bayes teoremine dayanır. İstatistiksel bir sınıflandırıcıdır. Eğitim verilerine dayanarak verilerin hangi sınıflara ait olabileceği konusunda olasılık hesabı yapar. Özniteliklerin birbirinden bağımsız olduğu varsayımıyla modelin çözümü kolaylaşır ve böyle durumlarda yöntem naif bayes yöntemi olarak adlandırılır.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Genetik algoritmalar yöntemi ise yine doğadan esinlenerek ortaya çıkarılmış bir yöntemdir. Doğada bulunan canlıların ileriki nesillere gen aktarımını örnek almış bir yöntemdir. Farklı farklı sonuçlar elde edebilmek için DNA aktarımında var olan mutasyon ve çaprazlama olayları modelde mevcuttur. Doğada güçlü canlıların hayatta kaldığı gibi bu yöntemde de iyi sonuçlar elde tutulur ve bir sonraki işlem adımına aktarılır. Sezgisel bir algoritmadır.

#### 2.4.Kümeleme

Çok yüzlü konik fonksiyonlar temelli sınıflandırma yaklaşımlarında, fonksiyonların merkez noktalarının belirlenmesi oluşturulan sınıflandırıcıların başarılarını etkilemektedir. Bu bölümde kümeleme temelli farklı merkez belirleme

stratejileri ile bütünleşik yöntemler üzerinde durulmaktadır. Bu anlamda kullanılan kümeleme yöntemleri ile ilgili bilgiler verilmektedir.

#### 2.4.1. K-ortalamlar algoritması

$k$ -ortalamlar algoritması küme bilgisi elde olmayan verilerin  $k$  sayıda kümeye atanması işlemini gerçekleştirir. Ağırlık merkezi tabanlı kümeleme algoritmalarından en sık kullanılan ve uygulanması oldukça basit bir algoritmadır. Algoritmanın temel amacı ağırlık merkezleri ile veriler arasındaki uzaklıkların karesini en küçüklemektir. Algoritma başlamadan önce küme sayısı  $k$  girdi olarak verilmelidir. Algoritma, başlangıç merkez noktalarını belirlemek amacıyla, verilen  $k$  küme sayısı kadar, kümelenecek veri noktaları arasından rassal olarak başlangıç noktaları seçer. Daha sonra tüm noktalar kendilerine öklid uzaklığına göre en yakın başlangıç noktasına atanır. Ardından yeni küme merkezleri o kümeye atanan noktaların merkezi olarak güncellenir. Bu iki adım küme merkezleri değişmeyinceye kadar birbirini izleyerek tekrar eder.

Eldeki veriler  $(x_1, x_2, \dots, x_n)$  ile gösterilsin ve  $n$  adet  $d$  boyutlu olsun.  $k$  küme sayısı  $k \leq n$  olmak üzere kümeler  $S = \{S_1, S_2, \dots, S_k\}$  ile gösterilsin. Böyle bir durumda en küçüklenmek istenen amaç fonksiyonu aşağıdaki gibidir:

$$\text{enk} \sum_{i=1}^k \sum_{x_j \in S_i} \|x_j - \mu_i\|^2$$

$\mu_i, S_i$  kümesine ait noktaların ortalamasıdır.

***k-ortalamlar algoritması*** : Öncelikle başlangıç adımı gerçekleştirilir. Küme sayısı kadar rassal olarak farklı noktalar küme merkezi olarak seçilir.

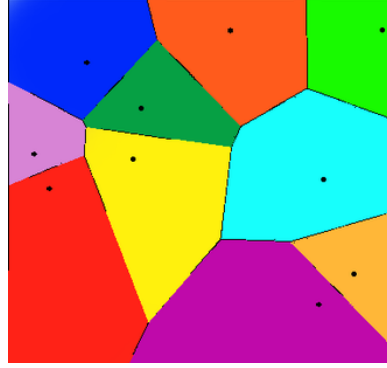
- *Atama adımı*: Öklid uzaklığına göre her noktayı en yakın merkeze ata.

$$S_i^{(t)} = \{x_p : \|x_p - m_i^{(t)}\| \leq \|x_p - m_j^{(t)}\| \forall 1 \leq j \leq k\}$$

- *Güncelleme adımı*: Atama adımında oluşturulan kümelerin yeni merkezleri hesaplanır.

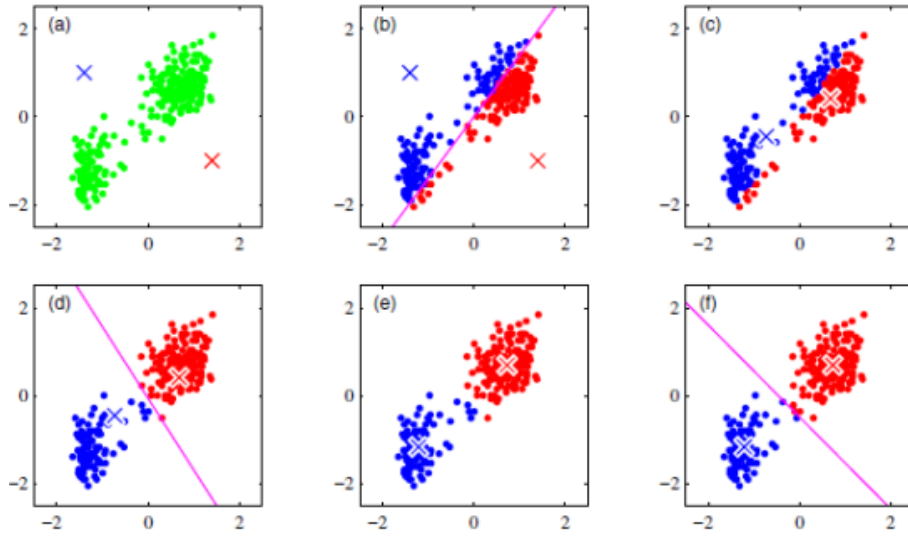
$$m_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j$$

Şekil 2.2 'de öklid uzaklıklarına göre atamaları gösteren Voronoi Diyagramı görülmektedir.

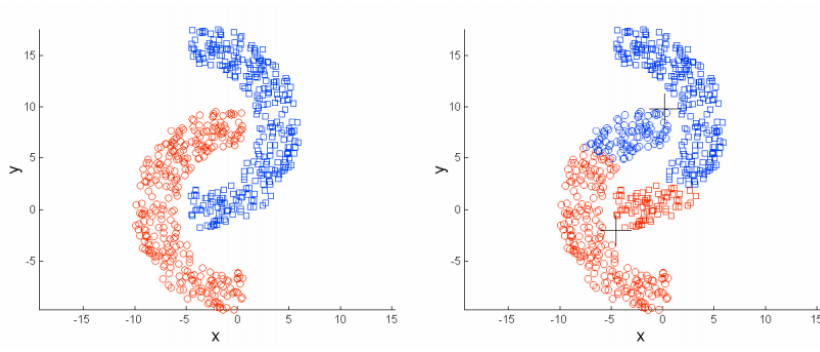


Şekil 2.2 Voronoi Diyagramı [24]

Şekil 2.3 'te adım adım  $k$ -ortalamalar algoritmasının kümeleri ayrışı gösterilmektedir:



Şekil 2.3  $k$ -Ortalamlar kümeleme yöntemi adımları [7]



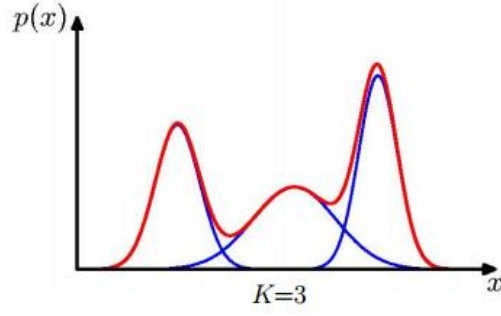
Şekil 2.4 Gerçek kümeler ve  $k$ -ortalamalar algoritmasıyla elde edilen kümeler [7]

$k$ -ortalamalar algoritması en iyi ayırmayı garanti etmese de büyük veri kümelerinde hızlı olarak çalışır. Bir çok kümeleme algoritması gibi  $k$  küme sayısını tespit edememesi olumsuz yönlerinden bir tanesidir. Veriler Şekil 2.4'te olduğu gibi dağıldığında  $k$ -ortalamalar kümeleme algoritması iyi sonuçlar verememektedir.

#### 2.4.2. EM algoritması

Temeli istatistik bilimine dayanan bu yöntem istatistiksel uygulamalarda, örüntü tanıma tekniklerinde, kayıp – eksik bilgilerin tahmininde ve kümeleme analizinde yoğun olarak kullanılmaktadır. Temeli karma gauss modellerine ( $GMM$ - Gaussian Mixture Models) dayanır.  $k$ -ortalamalar kümeleme yönteminde olduğu gibi her nokta %100 bir sınıfa atanacak diye bir şart yoktur. Her nokta ve her küme için noktanın hangi kümeye ne kadar olasılıkla ait olabileceği hesaplanır. Dolayısıyla nokta sadece bir kümeye atanmaz, noktalar kümelere belli olasılıklarla atanmaktadır.

Tek boyutta üçlü normal dağılım karma gauss modelini incelersek 3 sınıf için dağılım grafiği Şekil 2.4'teki gibi görünür.



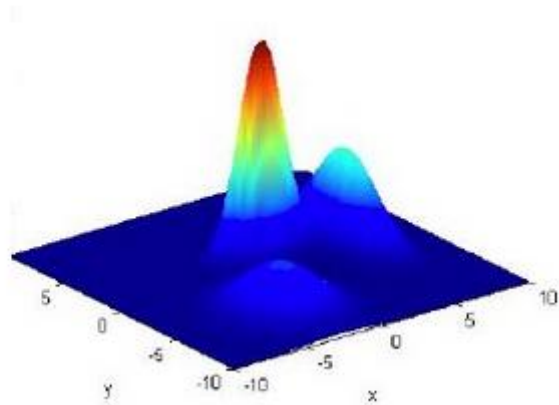
Şekil 2.5 Karma normal dağılım [7]

Böyle bir durumda  $p(x)$  olasılık hesabı her bir normal dağılım değerinin belli bir katsayı ile çarpımlarının toplamına eşittir.

$$p(x) = \sum_{k=1}^K \pi_k N(x|\mu_k, \Sigma_k)$$

$$\forall k : \pi_k \geq 0, \quad \sum_{k=1}^K \pi_k = 1$$

Tek boyutta değişkenlik değerini gösteren  $\sigma$  tek bir değer iken Şekil 2.5'in iki boyutlu hali olan Şekil 2.6 için değişkenlik tek bir değer değil  $d$  boyutlu uzay için  $d \times d$  boyutta  $\Sigma$  ile gösterilen bir kovaryans matrisidir.



Şekil 2.6 İki boyutta karma normal dağılım

$d$  boyutta karma dağılımlar için çok değişkenli normal dağılım olasılık hesabı aşağıdaki şekilde yapılmaktadır:

$$f_x(x_1, \dots, x_d) = \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \exp\left(-\frac{1}{2}(x - \mu)^T \Sigma^{-1} (x - \mu)\right)$$

Bağımsızlık varsayımı altında karma Gauss modelinin benzerliği aşağıdaki formülle ifade edilir :

$$\prod_{i=1}^N p(x_i) = \prod_{i=1}^N \sum_{k=1}^K \pi_k N(x_i | \mu_k, \Sigma_k)$$

*EM* algoritması Beklenti (*E-Expectation*) ve Maksimizasyon (*M-Maximization*) adımlarından oluşmaktadır. Algoritma başlangıçta rassal olarak ya da  $k$ -ortalamalar algoritması yardımıyla kümelenecek veri noktaları arasından merkezlerin seçilmesiyle başlatılır. Eldeki veriler  $(x_1, x_2, \dots, x_n)$  ile gösterilsin ve  $n$  adet  $d$  boyutlu olsun.  $k$  küme indisi olmak üzere:

***E Adımı:*** Elde bulunan parametreler  $\mu_k$  ve  $\Sigma_k$  ' e göre hangi noktanın hangi kümeye ait olduğunu gösteren aidiyet değerleri hesaplanır.  $\gamma_{ik}$   $i$ . noktanın  $k$ . kümeye ait olma olasılığını ifade eder. Küme için farklı bir merkez  $\mu$  ve kovaryans matrisi  $\Sigma$  olduğuna dikkat etmek gereklidir.

$$\gamma_{ik} = \frac{\pi_k N(x_i | \mu_k, \Sigma_k)}{\sum_{j=1}^K \pi_j N(x_i | \mu_j, \Sigma_j)}$$

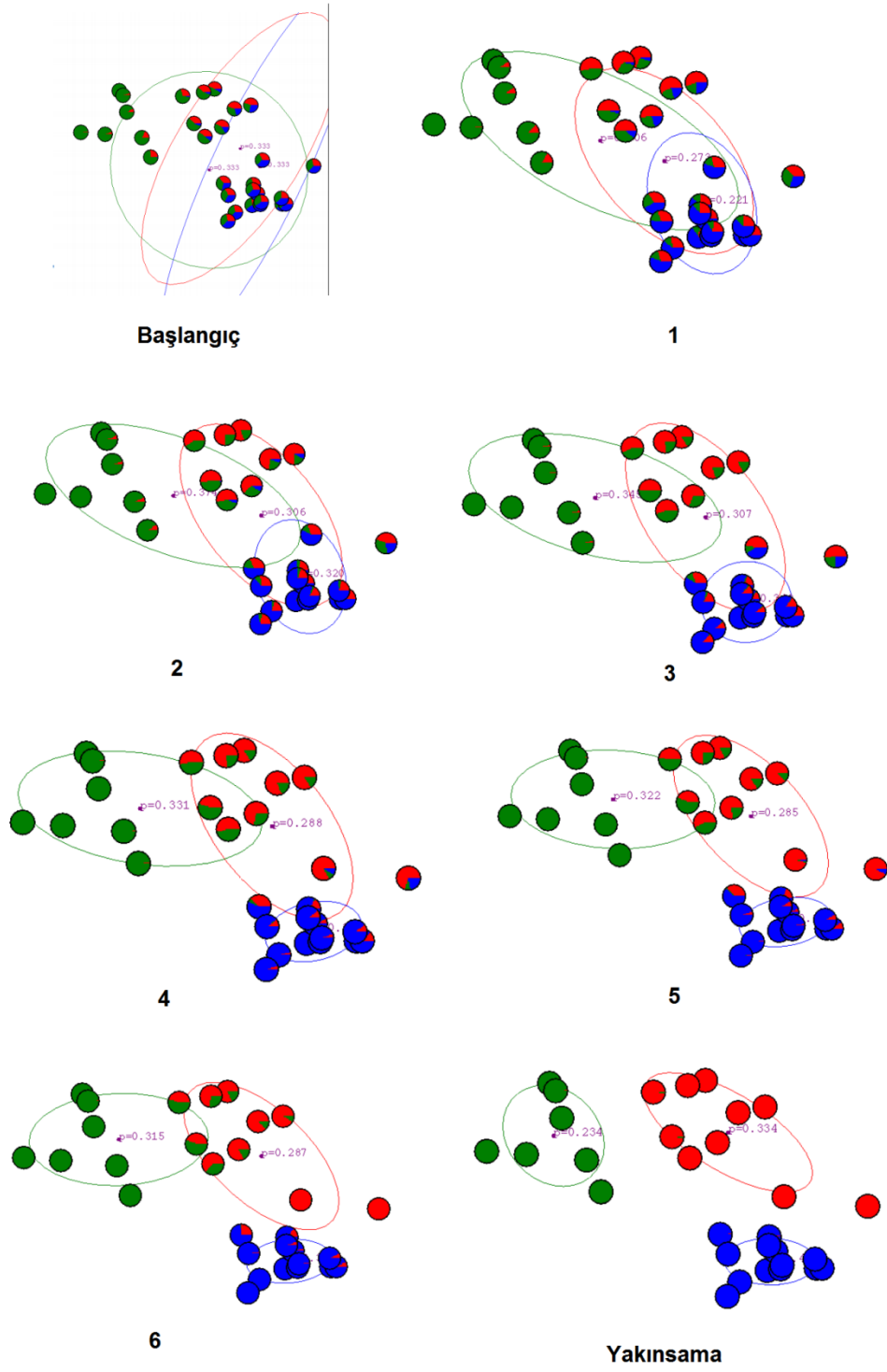
***M Adımı:*** *E* adımından sonra algoritmayı bitirme şartı sağlanmıyorsa *M* adımına geçilir. Elde olan noktaların kümelere ait olma olasılıklarına göre yeniden merkez  $\mu$ , kovaryans matrisi  $\Sigma$  ve küme ağırlık katsayıları  $\pi$  hesaplanır.

$$\mu_k = \frac{1}{N_k} \sum_{i=1}^N \gamma_{ik} x_i$$

$$\Sigma_k = \frac{1}{N_k} \sum_{i=1}^N \gamma_{ik} (x_i - \mu_k) (x_i - \mu_k)^T$$

$$\pi_k = \frac{N_k}{N}, \quad N_k = \sum_{i=1}^N \gamma_{ik}$$





Şekil 2.7 EM algoritması adımları [25]

$E$  ve  $M$  adımları algoritma sonlanıncaya dek birbirini izler. Şekil 2.7  $EM$  algoritmasının adım adım nasıl çalıştığını göstermektedir.

### 2.4.3. Geliştirilmiş bütünsel $k$ -ortalamalar algoritması

$k$ -ortalamalar algoritmasının başlangıç küme merkez noktaları rassal olarak seçilmektedir. Başlangıç merkez noktalarının rassal olarak seçilmesinin kümeleme hata oranına etkisi büyüktür. Geliştirilmiş bütünsel  $k$ -ortalamalar algoritması en iyi başlangıç noktalarının seçilmesini amaçlar ve başlangıç noktalarının seçilmesinden sonra  $k$ -ortalamalar algoritmasını kullanır.

Algoritma kümeleme problemini iki küme için çözümden başlar ve her adımda küme sayısını bir arttırarak devam ettirir. Her adımda bütün noktaların kendi küme merkezlerine olan uzaklıkları bulunur. Tüm uzaklıkların toplamında bir önceki uzaklık toplamına göre anlamlı bir iyileşme olmadıysa işlem durdurulur, en iyi küme sayısı bulunmuş olunur.

$k$ . küme merkezinin hesaplanması için başlangıç noktasını bulan algoritmanın adımları izleyen şekilde verilmiştir [17].

**Adım 1:** Her  $a^i \in D_0 \cap A$  için merkezi  $c^i$  merkezli  $s_2(a^i)$  kümesini hesapla,  $c^i$  noktasındaki  $\bar{f}_k$  fonksiyonunun değerini  $\bar{f}_{k,a^i} = \bar{f}_k(c^i)$ .

**Adım 2:**  $c^j$  merkezi  $s_2(c^j)$  kümesi için aşağıdaki değerleri hesapla:

$$\bar{f}_{k,min} = \min_{a_i \in D_0 \cap A} \bar{f}_{k,a^i}$$

$$a^j = arg \min_{a_i \in D_0 \cap A} \bar{f}_{k,a^i}$$

**Adım 3:**  $s_2(c^j)$  kümesi ve merkezi küme elemanları değişmeyinceye kadar tekrar hesapla.

Kümeleme problemleri için **arttırımlı algoritma** ise şu şekilde sunulmaktadır:

**Adım 1:**  $\epsilon > 0$  toleransı seç,  $A$  kümesinin merkezini hesapla,  $x^1 \in R^n$ , kümeleme fonksiyonunu hesapla  $f^1$ ,  $k = 1$  yap.

**Adım 2:** Bir sonraki küme merkezinin hesaplanması için;  $k = k + 1$  yap,  $x^1, \dots, x^{k-1}$   $k-1$  bölütleme (*partition*) problemi için merkez olsun,  $k$ . küme merkezi için bir başlangıç noktası bul (başlangıç noktası bulma algoritması ile),  $\bar{y} \in R^n$ .

**Adım 3:** Yeni başlangıç noktası olarak  $(x^1, \dots, x^{k-1}, \bar{y})$  seç,  $k$ -ortalamlar algoritmasını uygula ve karşı gelen çözümü  $(y^1, \dots, y^k)$  ve amaç fonksiyonu  $f^k$  değerini bul.

**Adım 4:**  $\frac{f^{k-1} - f^k}{f^1} < \epsilon$  ise dur, değil ise  $x^i = y^i$ ,  $i = 1, \dots, k$  yap, Adım 2'ye git.

Bu tez çalışmasında geliştirilmiş bütünsel  $k$ -ortalamlar algoritması kullanılırken en iyi küme sayısı bulunmamış, tıpkı  $k$ -ortalamlar algoritmasında olduğu gibi kümeleme problemi çözülmeden önce küme sayısı belirlenmiştir.

## 2.5.Çok Yüzlü Konik Fonksiyonlar ile Sınıflandırma

Sınıflandırma problemi temel olarak aynı noktayı içermeyen iki kümeyi en iyi şekilde ayırabilecek yapıların elde edilmesi problemi olarak tanımlanabilir. Böyle bir ayırma ikili sınıflandırma problemini ifade etmektedir. Çok yüzlü konik fonksiyonlar temelli algoritmalar ile sınıflandırmada amaç farklı iki kümeyi en iyi şekilde ayırabilen ve problemin yapısına göre doğrusal olmayan yüzeylerin elde edilmesidir. Böylelikle fonksiyonun aldığı değere göre noktaların hangi kümeye ait olduğu tespit edilebilir. Çoklu sınıflandırma ise ikili sınıflandırma probleminin ardışık bir hali olarak tanımlanmaktadır. Her adımda bir sınıf seçilir ve bu sınıfı diğer tüm sınıfların birleşiminden oluşan noktalar kümesinden ayrılmasını sağlayan fonksiyonlar hesaplanır. Bu şekilde ayrılacak sınıf karşısında tüm sınıfların seçilmesi 1'e karşı hepsi algoritması olarak tanımlanmaktadır. Çok yüzlü konik fonksiyonlar ile sınıflandırma için fonksiyonlar ile aynı isimde olan Çok yüzlü konik fonksiyonlar algoritması [6] ve  $k$ -ort-ÇKF [5] gibi çeşitli algoritmalar geliştirilmiştir.

Öztürk'ün çalışmasında [6] tanımlanan ve çok yüzlü konik fonksiyonlar ile sınıflandırma algoritmalarının temelini oluşturan ayırma fonksiyonu  $g_{(w,\xi,\gamma,a)} : R^n \rightarrow R$  denklem ile tanımlanmaktadır :

$$g_{(w,\xi,\gamma,a)}(x) = w(x - a) + \xi \|x - a\|_1 - \gamma$$

$w \in R^n, \xi \in R_+ = [0, +\infty), \gamma \geq 1, wx = w_1x_1 + \dots + w_nx_n$  ifadesi  $w$  ile  $x$  vektörlerinin skaler çarpımı ve  $\|x\|_1 = |x_1| + \dots + |x_n|$  ise  $x$  vektörünün 1 normudur. Fonksiyonun tepe noktası  $(a, -\gamma)$ , seviye kümesi dış bükey çok yüzlü ve grafiği konidir.

Bahsedilen algoritmalar sınıflandırma problemini bir doğrusal programlama probleminin çözümü olarak  $w, \xi$  ve  $\gamma$  parametrelerini elde eder. Bu parametreler ayırıcı fonksiyonun oluşturulmasında kullanılır.

### 2.5.1. Çok yüzlü konik fonksiyonlar algoritması

Çok yüzlü konik fonksiyonlar algoritması Öztürk'ün [6] çalışmasında sunulmuştur.  $A$  kümesinin  $B$  kümesinden ayrılmak istendiğini varsayalım. Algoritma öncelikle rassal olarak  $A$  kümesinden bir tepe noktası seçer. Daha sonra fonksiyonun alt seviye kümesi olan dış bükey çok yüzlü mümkün olduğunca  $B$  noktasından olan küme dışarıda kalacak şekilde ve  $A$  kümesi noktaları içeride kalacak şekilde fonksiyonlar oluşturulur. Her adımda  $A$  kümesinden bir kısım nokta  $B$ 'den ayrılır.  $A$  kümesinden olan tüm noktalar ayrılıncaya kadar adımlar tekrarlanır. Nihai sınıflandırıcı fonksiyon elde edilen tüm fonksiyonların noktasal en küçüğü olarak hesaplanır.

$A$  ve  $B$  kümeleri  $R^n$ 'de verilmiş iki küme iken algoritma adımları şu şekildedir:

$$A = \{a^i \in R^n : i \in I\}, \quad I = \{1, \dots, m\}$$

$$B = \{b^j \in R^n : j \in J\}, \quad J = \{1, \dots, p\}$$

**Adım 0:**  $l = 1, l_l = I, A_l = A$  atamalarını yap. Adım 1 e git.

**Adım 1:**  $a^l$  noktası,  $A_l$  kümesinin herhangi bir noktası olsun.  $P_l$  problemini çöz.

$$\omega(\alpha^i - \alpha^l) + \xi \|\alpha^i - \alpha^l\|_1 - \gamma + 1 \leq y_i, \quad \forall_i \in I_l,$$

$$-\omega(\alpha^i - \alpha^l) - \xi \|\alpha^i - \alpha^l\|_1 + \gamma + 1 \leq 0, \quad \forall_j \in J,$$

$$y = (y_1, \dots, y_m) \in R_+^m, \quad \omega \in R^n, \quad \xi \in R, \gamma \geq 1$$

kısıtları altında;

$$(P_l) \quad \text{enk} \left( \frac{ye_m}{m} \right)$$

$(P_l)$  probleminin bir çözümü  $w^l, \xi^l, \gamma^l, y^l$  'yi bul. Bu çözüme karşı gelen çok yüzlü konik fonksiyon aşağıdaki şekilde oluşturulur.

$$g_l(x) = g_{\omega^l, \xi^l, \gamma^l, y^l}(x)$$

**Adım 2:**  $I_{l+1} = \{i \in I_l : g_l(\alpha^i) + 1 > 0\}$ ,  $A_{l+1} = \{\alpha^i \in A_l : i \in I_{l+1}\}$ ,  $l = l + 1$  güncellemelerini yap.  $A_l \neq \emptyset$  ise adım 1'e git.

**Adım 3:**  $A$  ve  $B$  kümelerini ayıran  $g(x)$  fonksiyonu denklem ile tanımlanır. Durulur.

### 2.5.2. Kümeleme temelli sınıflandırma yaklaşımları

Çok yüzlü konik fonksiyonlar algoritmasında rassal olarak seçilen fonksiyon tepe noktalarının sınıflandırma başarısı üzerine etkisi oldukça yüksektir. Algoritmanın doğası gereği eğitimde tüm noktalar ayrılmakta ve %100 eğitim başarısı sağlanmaktadır. Ancak test aşamasında bu algoritmanın literatür ile rekabet edebilir düzeyde olduğu gösterilmiştir [26]. Eğitim aşamasında elde edilen başarıları arttırmak amacıyla Çiftçi 'nin [5] çalışmasında merkez

noktalarının seçiminin rassal olarak gerçekleştirilmesi yerine  $k$ -ortalamalar kümeleme algoritması ile seçilmesi öne sürülmüştür. Bu aşırı uyum (overfitting) sorununa çözüm getirmektedir. Modelde aşırı uyum sorununu gidermek amacıyla modelleme aşamasında  $A$  kümesine ait noktaların tamamını ayırmak yerine, “çok yüzlü dışındaki  $A$  kümesi noktalarının ve çok yüzlü içindeki  $B$  kümesi noktalarının toplam sayısını” en küçükleme amaçlanmıştır.  $A$  kümesi çok yüzlü konik fonksiyonlar algoritmasında tek iken  $k$ -ort-ÇKF algoritmasında  $A$  kümesi alt kümeler bölünür ve her adımda bu alt kümeler  $B$  kümesinden ayrılır.

Literatürde  $k$ -ortalamalar kümeleme yöntemi en iyi ve en sık kullanılan algoritma olmakla birlikte bu bölümün önceki başlıklarında bahsedildiği üzere farklı disiplinlere dayanan bir çok kümeleme algoritması bulunmaktadır. Bir algoritmanın diğerine üstün olduğunu söylemek mümkün değildir. Kümelemenin uygulanacağı veri kümesinin dağılımı; algoritmanın işlem karmaşıklığı ve zamanı gibi kriterler göz önüne alındığında her bir kümeleme algoritmasının diğerine göre üstün – zayıf yönleri mevcuttur. Bu sebeple Çiftçi'nin çalışmasında [5] gösterilen başarılı sonuçların motivasyonu ile farklı kümeleme algoritmalarının kullanılmasının sınıflandırma başarısı üzerine etkili olabileceği düşünülmüştür. Sınıflandırma başarılarını, program çalışma performanslarını karşılaştırmak amacıyla bu tez çalışmasında  $EM$  ve geliştirilmiş bütünsel  $k$ -ort kümeleme algoritmaları merkezlerin ve kümelerin belirlenmesinde kullanılmıştır.

İki algoritmanın eğitim ve test adımları  $f$  kez çapraz doğrulama için izleyen şekildedir:

$A$  veri kümesi  $R^n$  'de  $n$  boyutlu uzayda tanımlı olsun ve  $m$  adet veriye sahip olsun.

$$A = \{\alpha^i \in R^n : i \in I\}, I = \{1, 2, \dots, m\}$$

Sınıflandırma probleminin çözümü için veri kümesi *eğitim* ve *test* veri kümesi olarak ikiye ayrılmaktadır. Sırasıyla:

$$\hat{A} = \{\alpha^i \in R^n : i \in \hat{I} \subset I\} \text{ ve } \bar{A} = A/\hat{A}$$

**Adım 0:**  $j := 0$  yap ve  $k$ 'yi küme sayısına eşitle.

**Adım 1:**  $j := j + 1$  yap ve  $j$  sınıfı için eğitim kümesinden bir alt küme seç,  $\hat{A}_j$ .

**Adım 2:** Seçilen alt kümeye seçilen kümeleme algoritmasını uygula. Kümelenen verileri,  $\hat{A}_{jr}$ , ve her küme için merkez noktaları elde et.

$$\bar{x}_{jr} \in R^n, r = 1, \dots, k$$

**Adım 3:**  $k$ -ort-ÇKF'leri  $g_{jr}(x), r = 1, \dots, k$  'i  $(\omega^{jr}, \xi_{jr}, \gamma_{jr})$  parametreleri ile  $j$  sınıfı için aşağıdaki  $P_{jr}$  doğrusal programlama modelini çözerek bul.  $P_{jr}$  çözerken  $\hat{A}_{jr}$  kümesini ve merkezler  $\bar{x}_{jr} \in R^n, r = 1, \dots, k$  'i kullan.

$$(P_{jr}) \left\{ \begin{array}{l} \min \frac{1}{|\hat{A}_{jr}|} \sum_{i \in \hat{A}_{jr}} y_i + \frac{1}{|\hat{B}_j|} \sum_{l \in \hat{B}_j} z_l \\ \text{k. a} \\ \omega^{jr} (\alpha_i - \bar{x}_{jr}) + \xi_{jr} \|\alpha_i - \bar{x}_{jr}\|_1 - \gamma_{jr} + 1 \leq y_i, \quad \forall i \in \hat{A}_{jr} \\ -\omega^{jr} (b_l - \bar{x}_{jr}) - \xi_{jr} \|b_l - \bar{x}_{jr}\|_1 - \gamma_{jr} + 1 \leq z_l, \quad \forall l \in \hat{B}_j \\ y_i > 0, z_l > 0 \end{array} \right.$$

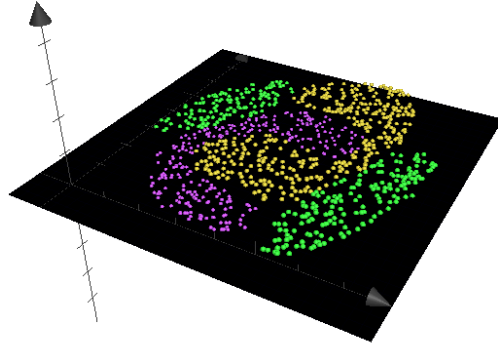
$$\hat{A}_{jr} = \{i: \alpha_i \in \hat{A}_j\}, \hat{B}_j = \cup_{s=1}^c \hat{A}_s, s \neq j \text{ ve } \hat{B}_j = \{i: \alpha_i \in \hat{B}_j\}$$

**Adım 4:** Ayırıcı fonksiyonu elde et.

$$g_j(x) = \min_{r=1, \dots, k} g_{jr}(x)$$

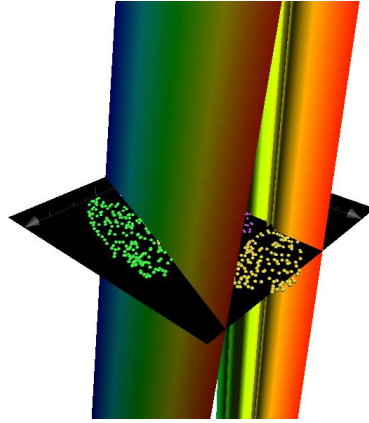
**Adım 5:**  $i$ . Eğer  $j < c$  ise adım 1'e git, değilse dur.

Bu üç algoritma ile oluşacak fonksiyonları göstermek amacıyla üç sınıflı bir örnek oluşturmuştur. Sınıflardan ikisi iç içe geçmiş ve dış bükey değildir. Diğer sınıf (yeşil) ise ayırık ve iki kümeye bölünmüştür. Şekil 2.8'de üç sınıf görülmektedir.



Şekil 2.8 İki boyutlu açıklayıcı örnek dağılımı

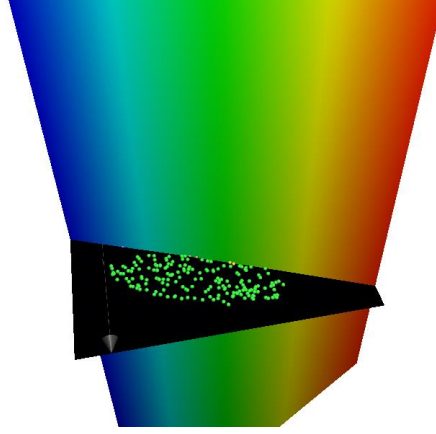
Şekil 2.9'da *EM-ÇKF* ile çözülmüş ve yeşil sınıfı ayıran fonksiyon görülmektedir.



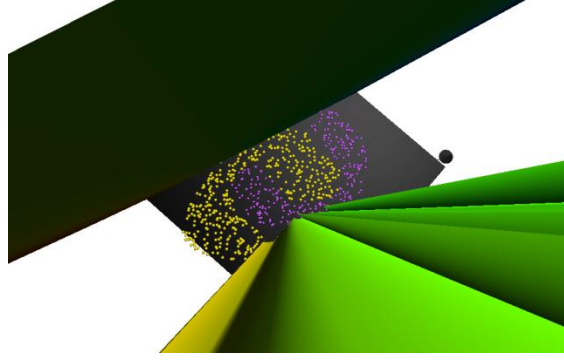
Şekil 2.9 *EM-ÇKF* algoritması ile yeşil sınıfı ayıran fonksiyondan bir görünüm

Şekil 2.10 ve Şekil 2.11'de *EM-ÇKF* ile çözülmüş ve yeşil sınıfı ayıran fonksiyon farklı bir açıdan görülmektedir.



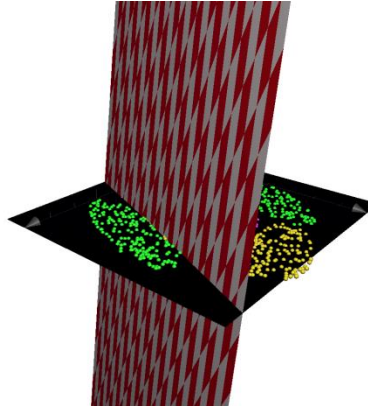


Şekil 2.10 *EM-ÇKF* algoritması ile yeşil sınıfı ayıran fonksiyondan bir görünüm

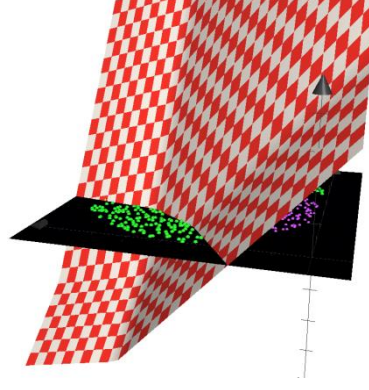


Şekil 2.11 *EM-ÇKF* algoritması ile yeşil sınıfı ayıran fonksiyondan bir görünüm

Şekil 2.12 ve Şekil 2.13'te ise geliştirilmiş bütünsel *k*-ort-ÇKF ile çözülmüş ve yeşil sınıf kümelerini ayıran fonksiyonlar farklı bir açıdan görülmektedir.

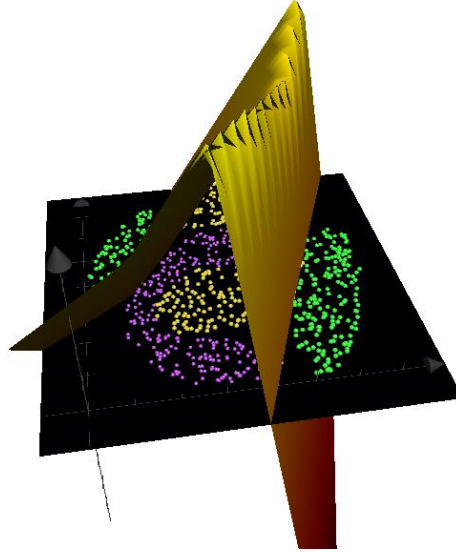


Şekil 2.12 *GB k-ort-ÇKF* algoritması ile yeşil bir kümeyi ayıran fonksiyon



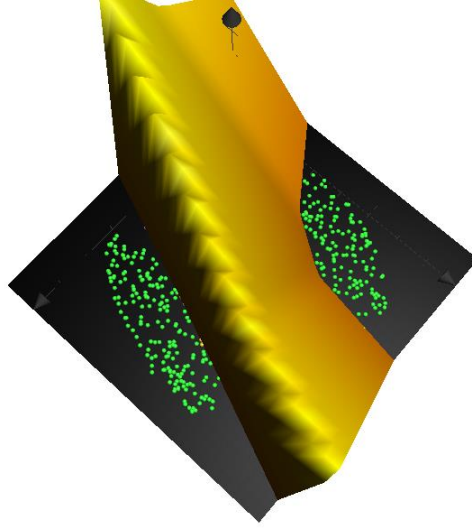
Şekil 2.13 GB  $k$ -ort-ÇKF algoritması ile yeşil bir kümeyi ayıran fonksiyon

Aşağıdaki Şekil 2.14'te  $k$ -ort-ÇKF ile çözülmüş ve yine yeşil sınıfı ayıran fonksiyon görülmektedir.

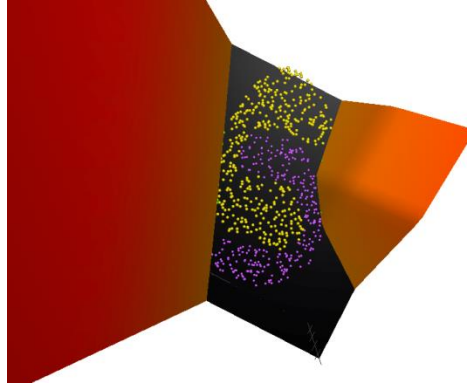


Şekil 2.14  $k$ -ort-ÇKF algoritması ile yeşil sınıfı ayıran fonksiyondan bir görünüm

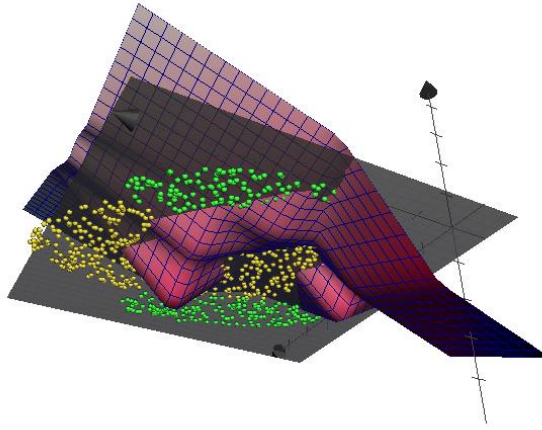
Şekil 2.15 ve Şekil 2.16'da  $k$ -ort-ÇKF ile çözülmüş ve yeşil sınıfı ayıran fonksiyon farklı açılardan görülmektedir.



Şekil 2.15  $k$ -ort-ÇKF algoritması ile yeşil sınıfı ayıran fonksiyondan bir görünüm



Şekil 2.16  $k$ -ort-ÇKF algoritması ile yeşil sınıfı ayıran fonksiyondan bir görünüm



Şekil 2.17  $k$ -ort-ÇKF algoritması ile mor sınıfı ayıran fonksiyondan bir görünüm

$k$ -ort-ÇKF ile elde edilmiş, mor sınıfı ayıran fonksiyon aşağıdaki Şekil 2.17’de görülmektedir.

## 2.6. Hesapsal Sonuçlar

$k$ -ort-ÇKF, geliştirilmiş bütünsel  $k$ -ort-ÇKF ve  $EM$ -ÇKF algoritmaları farklı özelliklere sahip çeşitli veri setlerinde denenmiş ve sonuçlar sunulmuştur. Veri setleri Bölüm 2.5.2’de gösterilen örnek hariç *UCI* Makine Öğrenmesi Veri Tabanları’ndan [27] elde edilmiştir. Test edilen veri setleri iris plant, wine, glass, vehicle, coverytype ve Bölüm 2.5.2’de gösterilen örnek veri setidir.

***Iris Plant Veri Seti:*** Bu veri seti iris bitkisini sınıflandırmak amacıyla oluşturulmuştur. Örnekler 3 sınıftan oluşmaktadır. Veri setinde 4 çok değişkenli gerçek öznitelikten oluşan 150 örnek bulunmaktadır. Ağustos 2013 tarihi itibarıyla 459785 gibi çok yüksek bir atıf sayısına sahiptir.

***Wine Veri Seti:*** Wine veri seti üç farklı şarap tipinin sınıflandırılması amacıyla oluşturulmuştur. 13 öznitelikli 178 örnekten oluşmaktadır.

***Glass Veri Seti:*** Cam veri seti farklı cam türlerinin sınıflandırılması amacıyla oluşturulmuştur. 10 öznitelikten oluşan 214 örnek mevcuttur. 6 sınıf mevcuttur.

***Vehicle Veri Seti:*** Bu veri seti 946 örnekten oluşur. Örnekler 18 adet özniteliğe sahiptir.

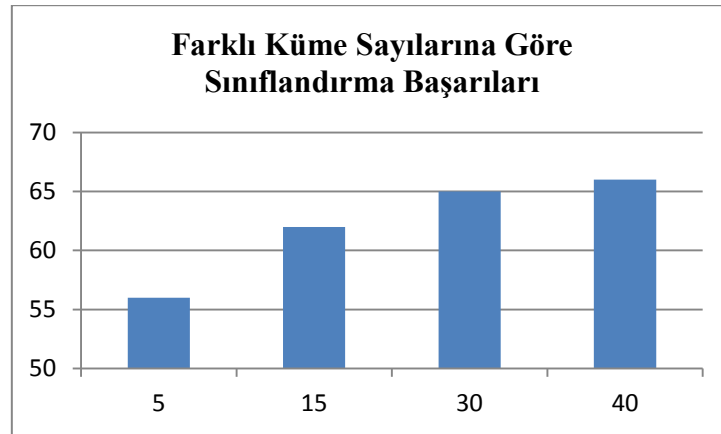
***Coverytype Veri Seti:*** Orman bitki örtülerini sınıflandırmak amacıyla oluşturulmuştur. 54 öznitelikli 581012 örneğe sahip büyük bir veri setidir.

Çizelge 2.1 Iris, wine, glass ve vehicle veri setleri sınıflandırma doğrulukları

Yöntem	Iris	Wine	Glass	Vehicle
1e1 DVM [28]	98.67	94.40	64.8	-
Kernel Algaç [29]	95.33	97.74	-	-
1eh DVM [30]	97.33	99.43	71.49	86.64
1eh ITS [6]	97.33	96.63	70.60	79.79
1eh ÇKF [6]	96.67	96.63	65.42	78.28
<i>k</i> -ort-ÇKF	98.0	96.60	75.2	82.7
EM-ÇKF	96.67	97.78	-	-
GB- <i>k</i> -ort-ÇKF	96.0	95.5	73.3	81.7

Çizelge 2.2 Coverttype veri seti sınıflandırma doğrulukları

Yöntem	Coverttype
Yapay sinir ağları	70
LDA	58
<i>k</i> -ort-ÇKF(40 küme)	66
<i>k</i> -ort-ÇKF(2 küme)	57
GB- <i>k</i> -ort-ÇKF(2 küme)	60



Şekil 2.18 Farklı küme sayılarına göre sınıflandırma başarıları

**Çizelge 2.3** Bölüm 2.5.2 açıklayıcı örnek sınıflandırma doğrulukları

Yöntem	Açıklayıcı örnek
SMO	56.21
<i>EM-ÇKF</i>	96.96
<i>k-ort-ÇKF</i>	98.98
<i>GB-k-ort-ÇKF</i>	98.98
J48	96.69
Naif Bayes	76.45
Logistics	53.20
Bagging	96.79
IBK	98.39

Tez çalışmasında elde edilen sınıflandırma başarıları ve literatürdeki bazı yöntemlerle karşılaştırmaları çizelgelerde gösterilmiştir. Sonuçlar glass veri seti hariç 10 kez çapraz doğrulama ile elde edilmiştir. Glass veri seti ise sınıfların bazılarında yetersiz veri olması sebebiyle 2 kez çapraz doğrulama ile sınanmıştır. Algoritmaların tamamı cpp ile kodlanmış ve doğrusal programlama problemlerinin çözümü için IBM Cplex kütüphaneleri kullanılmıştır.

## 2.7. Sonuçlar

Bu bölümde Çiftçi'nin çalışmasında [5] geliştirilen *k-ort-ÇKF* algoritması, bu tez çalışmasında geliştirilen *EM-ÇKF* algoritması, geliştirilmiş bütünsel *k-ort-ÇKF* algoritması ve diğer literatür yöntemleri karşılaştırılmıştır. *k-ort-ÇKF* algoritması ile sınıflandırma problemi için başarılı sonuçlar elde edilmiştir [5]. Ancak her kümeleme algoritmasının belli avantajları ve dezavantajları mevcuttur. Bu avantaj ve dezavantajları karşılaştırmak amacıyla başarılı küme algoritmalarından *EM-ÇKF* ve geliştirilmiş bütünsel *k-ort-ÇKF* algoritmalarına yönelinmiştir. Yapılan testler sonucunda üç algoritmanın birbirlerine yakın ve literatür değerlerinin bir çoğundan iyi olmakla birlikte, rekabet edebilir oldukları gözlemlenmiştir. Algoritmaların test doğruluk oranları açısından birbirlerine büyük bir üstünlükleri gözlemlenmemiştir. Ancak

algoritmaların kullanılabilirlikleri açısından ciddi farklar bulunmaktadır.  $k$ -ort ve geliştirilmiş bütünsel  $k$ -ort kümeleme algoritmaları,  $EM$  kümeleme algoritmasına hız açısından büyük üstünlük sağlamaktadır. Hesapsal sonuçlarda görüleceği üzere  $EM$  kümeleme algoritması ile sadece küçük veri setlerinde sonuçlar elde edilebilmiştir.

### 3. ÖZNETELİK SEÇİMİ İÇİN YENİ BİR YAKLAŞIM

Bu bölümde sınıflandırma problemi için önemli işlemler olan öznitelik çıkarma ve öznitelik seçme yöntemlerine değinilmiştir. Konu hakkında genel bilgilere, literatür özetine, kullanılan yöntem detaylarına ve sonuçlara bu bölüm ile ulaşılabilir.

#### 3.1.Giriş

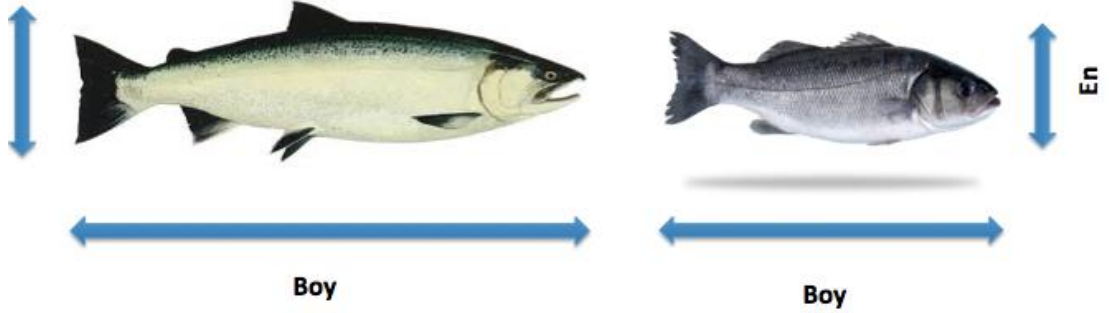
Makine öğrenmesi ve istatistik biliminde öznitelik seçimine; özellik seçimi, nitelik seçimi, değişken seçimi gibi isimlerle de karşılaşılabilir. Makine öğrenmesi ve istatistik bilimi gibi alanlarda veriler, öznitelikler ile ifade edilirler. Ancak öznitelik uzayı her zaman en iyi özniteliklerden oluşmaz.

Sınıflandırma problemi ele alınırsa, sınıflandırma yapmak için öncelikle sınıflandırılacak nesnelerin bazı öznitelikler ile tanımlanması sağlanır. Öznitelik sayısı bazen çok az olsa da bazı durumlarda yüzbinlerle ifade edilen sayılara çıkabilir. Her iki durumda da özniteliklerin sınıflandırma için ne kadar anlamlı olduğu kestirilemez. Öznitelik uzayında mutlaka gereksiz ve anlamsız öznitelikler olacaktır. Bu özniteliklerin çeşitli algoritmalar yardımıyla belirlenerek sınıflandırma problemi için daha anlamlı olan özniteliklerin seçimi, kısaca öznitelik seçimi problemini tanımlar. Literatürde var olan sınıflandırma algoritmalarının büyük bir bölümü çok yüksek sayıda öznitelik ile çalışmak için uygun değildir. Öznitelik uzayı boyutunun yüksek olması sınıflandırma başarısını düşürecek, hesapsal yükü artıracak ve dolayısıyla kabul edilebilir bir zamanda çözüme ulaşmayı engelleyecektir. Ayrıca büyük boyutlu veriler ile işlem yapmak bilgisayar hafızasından büyük kaynaklar ayırmayı gerektirir.

Bu tez çalışmasında öznitelik seçimi problemi üzerine yönelinmesinin sebebi tez çalışmasına konu olan hareket tanıma gibi gerçek zamanlı sistemlerde algoritma hızının çok önemli olması ve öznitelik sayısının da algoritma hızı için anahtar faktör olmasıdır. Bir hareket tanıma sisteminde bir saniyede 15-30 çerçeve görüntü işlenmesi gerekmektedir. Örneğin tez çalışmasının bir sonraki adımında el yeri tespitinde kullanılacak şekilsel faktörler arasından öznitelik seçimi yapmak kaçınılmazdır.

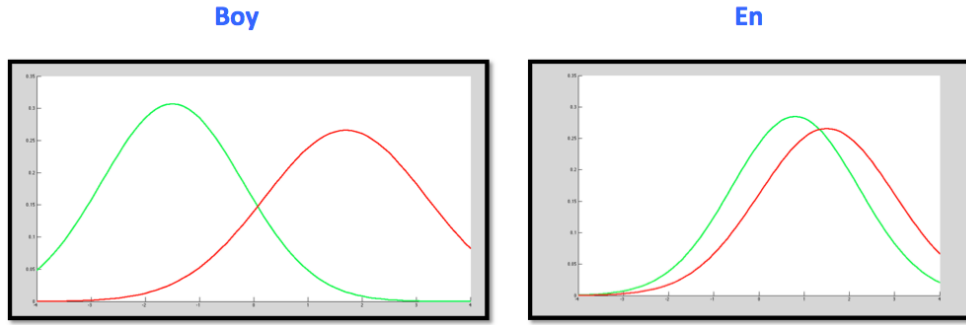


Basit bir örnek üzerinden kabaca öznitelik seçimi işlemini açıklamak için levrek ve somon balıklarının sınıflandırmak istenildiğini varsayalım. Öznitelikler balıkların enleri ve boyları olarak seçilsin. Hangi özneliğin balıkları sınıflandırma için daha anlamlı olduğunu belirleyelim. Şekil 3.1’de öznitelikler görülmektedir.



Şekil 3.1 Levrek ve somon balıkları için iki öznitelik

En ve boy özniteliklerinin balıklar için dağılımları incelenirse Şekil 3.2’ye benzer bir dağılım elde edilecektir.

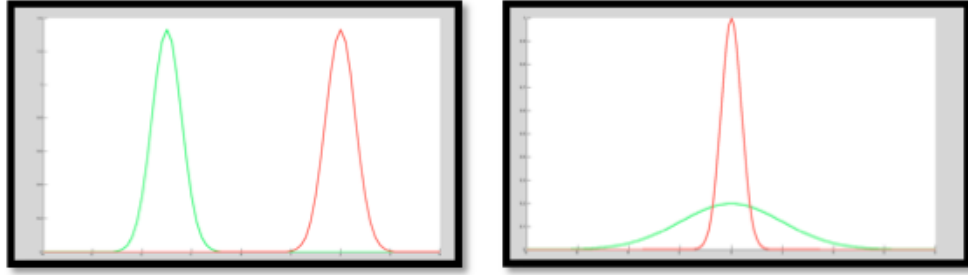


Şekil 3.2 En ve boy özniteliklerinin dağılım grafikleri

Kırmızı grafik levrek, yeşil grafik somon balığının öznitelik dağılımı iken; görüldüğü üzere boy özneliğinin ene göre sınıflandırma açısından daha ayırıcı olduğu söylenebilir.

Sınıflandırma açısından iyi ayırıcı öznitelik ortalamalarının her sınıf için anlamlı derecelerde farklı ve sınıf içi standart sapmalarının küçük olması gerektiği söylenebilir. Şekil 3.3’te solda iki sınıf için verilen özneliğın çok iyi bir ayırıcı

olduğu ve sınıfların bu özneliğe göre bir eşik değeri ile ayrılabilceği görülmektedir. Ancak böyle bir özneliği bulmak pratikte oldukça zordur. Şekil 3.3'te sağda ise bu özneliğe göre sınıflandırma yapmanın pek mümkün olmadığı görülmektedir.



Şekil 3.3 Öznelik dağılımlarına iki öznelik

Boyut indirgemenin özel bir hali ise öznelik çıkarma olarak tanımlanabilir. Öznelik çıkarma yöntemleri öznelikler arasından bir alt uzay seçmek yerine büyük boyutlu veriyi, daha küçük boyutta başka bir uzaya taşır. Böylece daha ayırt edici öznelikler elde ederken aynı zamanda yeni öznelikler ile boyutu da indirgemiş olur.

Öznelik seçimini çok güzel anlatan bir hikaye, çiftçi ve iki oğlunun hikyesidir. Zamanın birinde çiftçi iki oğlunu çağırır ve oğullarına karşıda gördükleri şeyin ağaç olup olmadığını sorar. Çocuklardan biri diğerine göre daha zekidir. Ancak daha az zeki olan çocuk çok kısa bir zamanda “ağaç” diye cevaplar. Zeki olansa uzunca bir süre cevap vermez ve bekler. Babası zeki olana neden cevap vermediğini sorduğunda, çocuk ağacın her yaprağını iyi bildiğini ve tüm yaprakları kontrol ettiğini söyler. Diğer çocuk ise nesnenin yeşil olduğu için ağaç olarak cevapladığını söyler. Bir nesnenin ağaç olup olmadığını söylemek için ne yeşil olduğunu görmek yeterlidir ne de tüm yaprakları kontrol etmek gereklidir. Problem için en iyi öznelik sayısı belirlenmelidir. Kısaca öznelik seçiminin temel amacı en fazla sınıflandırma başarısını sağlayacak, en az öznelik sayısını ve hangi özneliklerin seçilmesi gerektiğini belirlemektir.

### 3.2.Literatür Özeti

Sun ve ark. [31] Watanabe'nin [32] iki farklı örüntünün gereksiz öznitelikler ile benzer hale geldiğini gösterdiğini belirtmişlerdir. Öznitelik seçimi ile yapılmak istenen işlem aslında tam tersidir.

Öznitelik seçimi yöntemleri uygulama alanları oldukça geniştir. Dash ve Liu [33] makine öğrenmesi algoritmaları için öznitelik seçimi yöntemlerini kullanmışlardır. Özelde ise yüz bulma [34,35], araç bulma [35,36], yüz tanıma [37], takip etme [38] gibi problemlere uygulanmıştır.

Öznitelik seçimi algoritmaları temel olarak iki adıma ayrılır: Öznitelik seçim adımı, değerlendirme adımı . Değerlendirme yöntemleri ise temel olarak üç gruba bölünebilir. Bunlardan ilki sarıcı (wrapper) yöntemler olarak nitelendirilen en iyi öznitelik sonucu garantileyen yöntemlerdir. Diğerleri ise filtre ve gömülü (embedded) yöntemlerdir. Öznitelik seçim yöntemleri ise en iyi, sezgisel ve rassal yöntemler olarak üçe ayrılmaktadır. Bu yöntemlerden en basiti tam kapsamlı arama yöntemidir [39]. Ancak işlemsel yükün çok fazla olması sebebiyle çok tercih edilmez. En iyi çözümü garanti eden ve tam kapsamlı arama yöntemine göre daha hızlı olan bir yöntem de dal sınır algoritmasıdır [33,40]. Bu yöntemin, amaç fonksiyonunun monoton olması gibi bir kısıtı vardır ki; bu kısıt yöntemin kullanılabilirliğini azaltmaktadır. Ardışık ileri yönde seçim (sequential forward selection – *SFS*) ve ardışık geri yönde seçim (sequential backward selection – *SBS*) en iyi bilinen sezgisel yöntemlerden ikisidir [41]. Bu iki yöntem başarı ile uygulanabilen algoritmalarıdır. Bu algoritmaların bir genelleştirmesi olan ekle  $-l$  çıkar  $r$  (plus  $-l$  take away  $-r$ ) yöntemine temel oluşturmuşlardır [42]. Tüm bu yöntemlerin yerel en iyi sonucu buldukları, en iyi sonucu garantilemedikleri unutulmamalıdır. Öznitelik seçimi problemine adapte edilmiş bir diğer yöntem ise genetik algoritmalarıdır [43]. Rassal arama temeline dayanan öznitelik seçimi yöntemi destek (relief) [44] ve onun uzantısı destek- $f$  (relief- $f$ ) [45] öznitelik seçime uygulanmakta olan diğer yöntemlerdir. Öznitelik seçimi yöntemleri detaylarına Molina ve ark.'nın [46] literatürü araştırdığı çalışmalarından ulaşılabilir.

Literatürde öznitelik seçimi yöntemleri yoğun şekilde çalışılsa da araştırmacılar daha çok öznitelik çıkarma yöntemlerine ilgi duymaktadırlar. Öznitelik uzayı boyut sayısını indirgemek yerine tüm uzayı yeni bir az boyutlu uzaya transfer etmeyi amaçlamaktadırlar. Öznitelik seçimi konusunda en sık kullanılan yöntemlerden biri temelli 1901 yılında Pearson [47] tarafından ortaya atılan *Temel Eksenler Teoremi*'ne dayanan Temel Bileşenler Analizidir (Principal Component Analysis). Bu yöntem 1930'lu yıllarda Hotelling [48] tarafından geliştirilmiştir. Doğrusal ayırma analizi (Linear Discriminant Analysis) Fisher [49] tarafından ortaya atılan ve çok sık kullanılan bir diğer yöntemdir. Doğrusal ayırma analizi, temel bileşenler analizi ve faktör analizi benzerdir. Veriyi en iyi şekilde ifade edecek özniteliklerin doğrusal bileşimini bulur. Bağımsız bileşen analizi (Independent Component Analysis) [50] diğer yöntemlere göre daha yeni bir algoritmadır. Amaç doğrusal bir ayrıştırma ile istatistiksel olarak bağımsız faktörler elde etmektir. Kısmi en küçük kareler yöntemi [51] ise öznitelik seçimi için en yeni yöntemlerden biridir. Temel bileşen analizine dayanır ve geniş bir uygulama alanına sahiptir. Dalgacık dönüşümü ise sinyallerde gürültü azaltma, öznitelik çıkarma ve sinyal sıkıştırma için kullanılan yaygın bir yöntemdir. Öznitelik seçimi yöntemleri ile ilgili detaylı bilgiye Ding ve ark. [52] tarafından yapılan literatür araştırmasından ulaşılabilir.

### 3.3.Öznitelik Seçimi Yöntemleri

Öznitelik seçimi yöntemleri literatür araştırmasında da bahsedildiği üzere iki temel aşamadan oluşur. İlki özniteliklerin seçildiği öznitelik seçme adımı, ikincisi ise seçilen özniteliklerin iyiliğinin değerlendirildiği değerlendirme aşaması. Öznitelik seçim yöntemleri de kendi arasında en iyi, sezgisel ve rassal olarak üçe ayrılır. Yöntemlerin detayları izleyen kısımda verilecektir. Değerlendirme yöntemleri de kendi arasında üçe ayrılır. En iyi sonuçları sağlayan sarıcı (wrapper) yöntemler, filtre yöntemler ve gömülü (embedded) yöntemler. Her yöntemin bazı avantajları ve dezavantajları vardır. En iyi sonucu veren yöntemler gerçek öğrenme algoritmasını çözer ancak oldukça zaman alıcıdır. En iyi sonucu belirlenmiş bir öğrenme algoritması için verebilirler. Her adımda yapılması gereken çapraz doğrulama ile birlikte çözüm zamanı çok uzar. Filtre

yöntemleri daha hızlıdır fakat öğrenme algoritmasına entegre değildirler. Gömülü (embedded) yöntemlerde ise sarıcı (wrapper) ve filtre yöntemlerin tersine öznitelik seçimi aşaması öğrenme aşamasından ayrılamaz. Sınıflandırıcı fonksiyonun yapısı işlem üzerinde büyük öneme sahiptir.

Öznitelik seçim yöntemlerinden bir kısmı izleyen şekilde açıklanmaktadır: Tam kapsamlı arama yöntemi oluşabilecek tüm seçim olasılıklarını tek tek deneyerek seçilecek en iyi öznitelik kümesini bulur. Ancak bu, hem işlemsel yük açısından hem de zaman açısından oldukça maliyetlidir. Problem doğası gereği zaten yüksek boyuttaki verilerle çalışılmaktadır. Olası tüm kombinasyonları denemek bu durumlarda kabul edilebilir değildir.

“Dal ve sınır algoritması en iyi sonuca tam kapsamlı arama yönteminden daha kısa zamanda ulaşmaktadır. Ancak yöntemin en iyi sonuca ulaşabilmesi için ölçüt fonksiyonu monoton olmalıdır. Başka bir ifadeyle öznitelik alt kümeleri  $X$  ve  $Y$  için,

$$X < Y \Leftrightarrow J(x) < J(Y)$$

durumu sağlanmalıdır. Buna göre bir alt kümeye bir özneliğin eklenmesiyle elde edilen daha yüksek boyutlu bir alt kümeye ait ölçüt fonksiyonu değeri mutlaka daha büyük olmalıdır” [53].

Ardışık ileri yönde seçim, ardışık geri yönde seçim ve *ekle -l çıkar -r* yöntemleri birbirlerine çok benzer yöntemlerdir. Ardışık ileri yönde seçimde başlangıçta seçilmiş hiçbir öznitelik yoktur. Her adımda ölçüt fonksiyonuna göre en iyi öznitelik seçilir ve bir öznitelik seçilmiş uzaya eklenir. Seçilmek istenen öznitelik sayısı kadar adımlar tekrarlanır. Ardışık geri yönde seçimde ise ardışık ileri yönde seçim algoritmasının tam tersidir. Başlangıçta tüm uzay seçili durumdadır. Her adımda ölçüt fonksiyonuna göre en kötü özellik seçilir ve uzaydan çıkarılır. İstenilen sayıda öznitelik elde edilinceye kadar bu işlem devan ettirilir. *ekle -l çıkar -r* yöntemi ise bu iki algoritmanın bir genelleştirilmiş halidir. “Bu yöntem iç içelik etkisini azaltmak için öznitelik kümesi üzerinde  $l$  ve  $r$  parametreleriyle belirlenen miktarda ardışık ekleme ve çıkarma işlemi yapar” [53]. Esnek bir yapıya sahip şekilde öznelikleri seçer.

Tüm öznitelik çıkarma yöntemlerinde olduğu gibi temel bileşenler analizinde (*PCA*) de amaç çok fazla bilgi kaybetmeden verileri daha küçük boyutlu uzaylara taşımaktır. Temel bileşenler analizi basitçe özdeğerler vektörünü temel alan çok değişkenli istatistiksel yöntem olarak tanımlanabilir. Öncelikle ortalamadan sapmalar bulunur ve kovaryans matrisi hesaplanır. Kovaryans matrisinin özdeğerleri ve özdeğer vektörleri elde edilir. Özdeğerler en yüksekten en düşük değere sıralanacak şekilde özdeğerler ve özdeğer vektörleri yeniden sıralanır. Daha sonra bu vektörler ile veri seti yeni uzaya yansıtılır.

Bağımsız bileşen analizi, çok boyutlu istatistiksel verilerde saklı faktörlerin bulunması için geliştirilmiş olan oldukça yeni istatistiksel bir yöntemdir. *BBA* modelinde saklı faktörler ya da bağımsız bileşenler doğrusal veya doğrusal olmayan şekilde bilinmeyen bir karıştırma mekanizması ile karıştırılarak gözlem verilerini oluştururlar [54].

Doğrusal ayırma analizi (*LDA*) ise verileri en iyi ayırmayı sağlayacak öznitelikleri doğrusal ayrılabilmeyi en büyükleyecek şekilde seçer. Sınıf içi değişkenlikleri en aza indirirken sınıflar arası değişkenlikleri fazlalaştırmayı amaçlayan bir yöntemdir.

Viola ve Jones [34] tarafından sunulan nesne bulma algoritmasına burada ayrı bir parantez açmak gerekir. Nesne tanıma sistemleri için çığır açmış ve bir kilometre taşı olmuştur. İlk gerçek zamanlı nesne tanıma sistemini geliştirmişler ve saniyede 15 çerçeve görüntü işleme hızıyla (15 *fps*) nesnelerin görüntüdeki yerini tespit etmişlerdir. Nesnelerin yerini bulmak amacıyla Haar benzeri öznitelikler (Haar like features) kullanılır. Bahsedilen öznitelikler sınıflandırılacak her bir küçük çerçeve için yaklaşık 150 000 adettir. Bu durum bir öznitelik seçimi yöntemini kullanmayı zorunlu kılmıştır. Sınıflandırma ve öznitelik seçimini birlikte barındıran AdaBoost yöntemi nesneleri sınıflandırmak için kullanılmıştır. Çalışmalarını 2004 yılındaki yayınlarında [55] yüz bulma problemine uygulamışlardır. Bu tez çalışmasının ileriki adımlarında elin Haar benzeri öznitelikler ile tespit edilmesi amaçlanmaktadır. Bu nedenle AdaBoost yöntemiyle benzerlikler gösteren ancak çok yüzlü konik fonksiyonlar ve matematiksel programlama temeline dayanan bir öznitelik seçimi ve sınıflandırma yöntemi sunulmuştur. Algoritma adımlarına bir sonraki bölümde ulaşılabilir.



Şekil 3.4 AdaBoost tarafından seçilen iki öznelik [34]

AdaBoost yönteminin temel prensibi zayıf sınıflandırıcılar (%50'den biraz daha başarılı) elde edip bu zayıf sınıflandırıcıları belli ağırlıklar ile tek bir güçlü sınıflandırıcıda birleştirmektir.

$$H(x) = a_1 h_1(x) + a_2 h_2(x) + a_3 h_3(x) + \dots$$

$$H(x) = \sum_{t=1}^T a_t h_t(x)$$

$(x_1 y_1, \dots, x_m y_m)$ ;  $x_i \in X$ ;  $y_i \in \{+1, -1\}$   $x$ 'ler öznelik vektörü iken,  $y$ 'ler iki sınıf için sınıf bilgileridir. Algoritma adımları izleyen şekildedir:

$t = 1, \dots, T$  kez tekrarla:

**Adım 0:** Her nesnenin ağırlığını (olasılık dağılımı) eşit olarak hesapla.

$$D_1(i) = 1/m$$

**Adım 1:** Hata en az olacak şekilde  $t$ .adım için  $D$  ağırlıklarını kullanarak zayıf sınıflandırıcı fonksiyonu çağır.

$$h_t(x): X \rightarrow \{+1, -1\}$$

**Adım 2:** Seçilen sınıflandırıcı için hatayı hesapla.

$$\varepsilon_t = \frac{1}{2} \sum_{i=1}^m D_t(x_i) \cdot |h_t(x_i) - y_i|$$

**Adım 3:**  $h_t$  için ağırlık değeri olan  $a_t$ 'yi hesapla.

$$a_t = \frac{1}{2} \ln \left( \frac{1 - \varepsilon_t}{\varepsilon_t} \right)$$

**Adım 4:** Nesnelere için ağırlıkları güncelle.

$$D_{t+1}(x_i) = \frac{D_t(x_i) e^{-a_t y_i h_t(x_i)}}{Z_t}$$

$Z_t$  burada normalleştirme için kullanılır.

$$\sum_{i=1}^m D_{t+1}(x_i) = 1$$

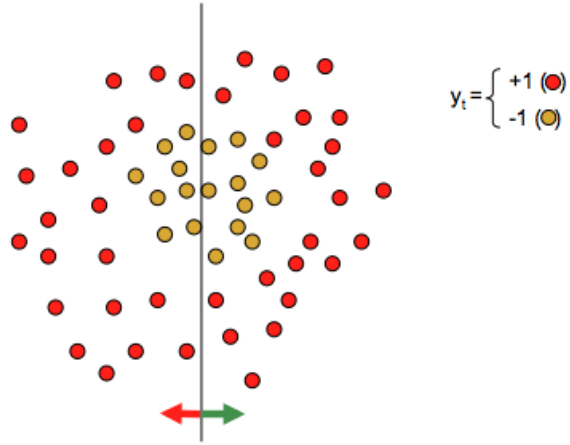
**Adım 5:**  $\varepsilon_t < 0.5$  ise ve  $t \neq T$  ise adım 1'e dön.

**Adım 6:** Güçlü sınıflandırıcıyı oluştur.

$$H(x) = \text{sign} \left( \sum_{t=1}^T a_t h_t(x) \right)$$

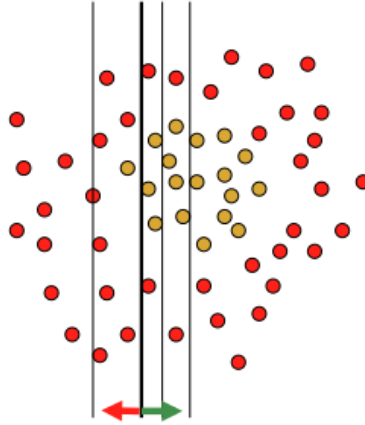
Aşağıdaki örnek algoritmayı görsel bir şekilde açıklamaktadır: Veri kümesi aşağıdaki şekilde dağılmaktadır ve her adımda doğru ile gösterilen basit sınıflandırıcılar elde edilecektir [56].





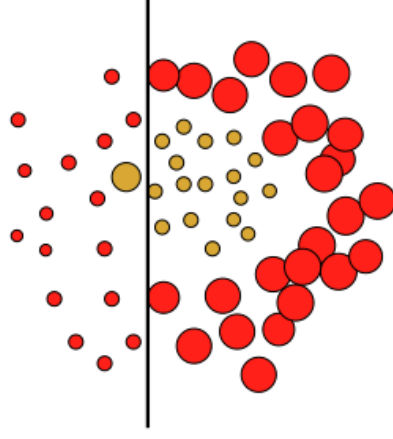
Şekil 3.5 Ayrılması amaçlanan iki küme

En iyi basit sınıflandırıcı seçilir.

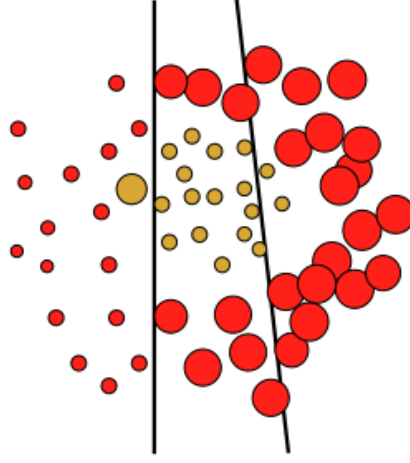


Şekil 3.6 İlk adımda ayırıcı zayıf sınıflandırıcının seçimi

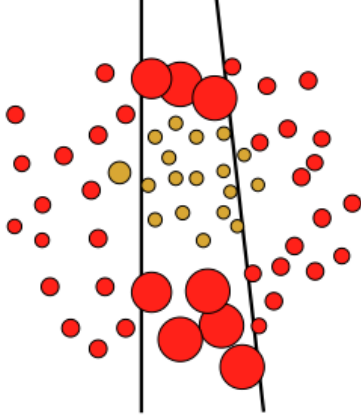
$\epsilon_t$  ve  $a_t$  hesaplanır. Ağırlıklar güncellenir. *Adım 4* 'te verilen formüle göre bir önceki adımda yanlış sınıflandırılan nesnelerin ağırlığı artar. Her adımda yeni ağırlıklara göre basit sınıflandırıcılar elde edilir.



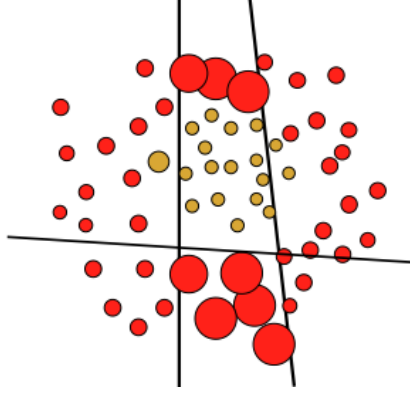
Şekil 3.7 Ağırlıkların değişmesi ve ikinci zayıf sınıflandırıcı seçimi



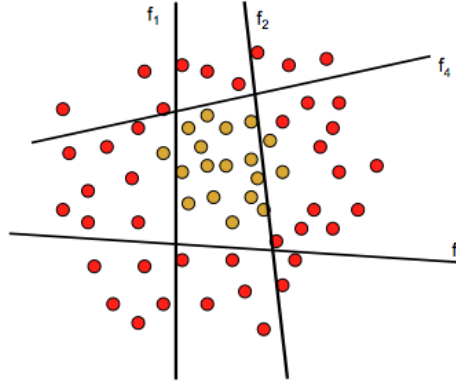
Şekil 3.8 Ağırlıkların değişmesi ve üçüncü zayıf sınıflandırıcı seçimi



Şekil 3.9 Ağırlıkların değişmesi ve dördüncü zayıf sınıflandırıcı seçimi



Şekil 3.10 Ağırlıkların değişmesi ve beşinci zayıf sınıflandırıcı seçimi



Şekil 3.11 Son sınıflandırıcının oluşturulması

Son güçlü sınıflandırıcı tüm fonksiyonların hesaplanan ağırlık değerleriyle çarpımlarının toplamıdır.

### 3.4.Önerilen Yaklaşım

AdaBoost yöntemi bir çok yönden avantajlı bir algoritma olsa da çok uzun eğitim zamanları gerektirir. Tez çalışmasında uygulanan yöntem sayesinde daha kısa eğitim zamanları ve daha başarılı bir algoritma geliştirmek amaçlanmıştır. AdaBoost algoritmasından farklı olarak her adımda zayıf sınıflandırıcılar bulunurken tek öznelik kullanan çok yüzlü konik fonksiyonlar kullanılmıştır. Öznelik sayısı kadar elde edilen çok yüzlü konik fonksiyonlara arasından en iyi ayrılan öznelik

seçilir ve doğru-yanlış ayrılan noktalar tutulur. Bu işlem nokta ağırlıkları güncellenerek tekrarlanır ve tüm adımlar bittiğinde belirlenen en iyi öznitelikler arasından aşağıdaki algorithmada verilen iki amaçlı modele göre öznitelikler tekrar seçilir. Model ayrıntıları aşağıdaki algorithmada verilmiştir.

$(x_1y_1, \dots, x_my_m)$ ;  $x_i \in X$ ;  $y_i \in (+1, -1)$   $x$ 'ler öznitelik vektörü iken,  $y$ 'ler iki sınıf için sınıf bilgileridir.

$t = 1, \dots, T$  kez tekrarla:

**Adım 0:** Her nesnenin ağırlığını (olasılık dağılımı) eşit olarak hesapla.

$$D_1(i) = 1/m$$

**Adım 1:** Hata en az olacak şekilde  $t$ .adım için  $D$  ağırlıklarını kullanarak çok yüzlü konik fonksiyonları oluştur. Öznitelik sayısı kadar çok yüzlü konik fonksiyon oluştur. En iyi özniteliğe ait çok yüzlü konik fonksiyon seç.

$$h_t(x): X \rightarrow \{+1, -1\}$$

**Adım 2:** Doğru ayrılan ve yanlış ayrılan noktaları tut.

**Adım 3:** Seçilen sınıflandırıcı için hatayı hesapla.

$$\varepsilon_t = \frac{1}{2} \sum_{i=1}^m D_t(x_i) \cdot |h_t(x_i) - y_i|$$

**Adım 4:**  $h_t$  için ağırlık değeri olan  $a_t$ 'yi hesapla.

$$a_t = \frac{1}{2} \ln \left( \frac{1 - \varepsilon_t}{\varepsilon_t} \right)$$

**Adım 5:** Nesnelere için ağırlıkları güncelle.

$$D_{t+1}(x_i) = \frac{D_{t+1}(x_i)e^{-a_t y_i h_t(x_i)}}{Z_t}$$

$Z_t$  burada normalleştirme için kullanılır.

$$\sum_{i=1}^m D_{t+1}(x_i) = 1$$

**Adım 6:**  $\varepsilon_t < 0.5$  ise ve  $t \neq T$  ise adım 1'e dön.

**Adım 7:** Matematiksel modeli çöz. Matematiksel modelde amaçlardan biri öznitelik seçimleriyle toplamda ayrılmış olan noktaları en büyükmek iken diğeri ise en az sayıda özniteliği seçebilmektir.

Şekil 3.2'de satırlar seçilen öznitelikleri, sütunlar ise veri setindeki nesnelere göstermektedir. Hangi nesnenin hangi sınıflandırıcıyla ayrılabilirdiği görülmektedir.

433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450	452	453	454	455	457	458	459	460	461	462	463	464	465	466	
1	1	1	1	1	1	1	1	1																1	1	1	1	1	1	1	1	1
1	1	1	1		1		1			1	1			1	1	1		1				1	1	1	1	1				1	1	1
						1	1	1		1	1			1	1	1		1			1	1										1
1					1	1			1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	1	1		1		1		1						1		1								1			1	1			1	1

Şekil 3.12 Seçilen sınıflandırıcıya göre doğru ve yanlış ayrılan nesnelere

$i = 1, \dots, m$  ve  $l = 1, \dots, T$ ;

$$y_l = \begin{cases} 1; & \text{eğer ilgili öznitelik seçildiyse} \\ 0; & \text{diğer durum} \end{cases}$$

$$x_i = \begin{cases} 1; & \text{eğer ilgili nesne doğru ayrıldıysa} \\ 0; & \text{diğer durum} \end{cases}$$

$$p_{il} = \begin{cases} 1; & \text{eğer } x_i \text{ l. öznitelik ile doğru ayrıldıysa} \\ 0; & \text{diğer durum} \end{cases}$$

$$x_i \leq \sum_l p_{il} y_l \quad i = 1, \dots, m; \quad l = 1, \dots, T$$

$$1 \leq \sum_l y_l \quad i = 1, \dots, m; \quad l = 1, \dots, T$$

Kısıtları altında;

$$f_1 = enb \sum_i x_i$$

$$f_2 = enk \sum_l y_l$$

**Adım 8:** Matematiksel model tarafından seçilen öznitelikler ile  $k$ -ort-ÇKF algoritmasını çöz.

### 3.5. Hesapsal Sonuçlar

Algoritmanın sınanması için *UCI* Musk veri seti kullanılmıştır. Musk bir çeşit moleküldür. 2 sınıflı bir veri setidir ve amaç test edilecek molekülün musk molekülü olup olmadığını sınamaktır. Bu veri kümesinde, her birinde 166 öznitelik olan 476 örnek vardır. Bu öznitelikler molekülün şekli, bağ yapısı vb. ile ilgili bilgileri içermektedir.

**Çizelge 3.1** Literatürde farklı algoritmalar için sonuçlar [57]

	<b>LPBoost</b>	<b>CRB</b>	<b>AdaBoost</b>	<b>C4.5</b>
C4.5 ile boost	0.882	0.906	0.929	0.834
Karar ağaçları ile boost	0.882	-	0.891	-

Tez çalışmasında geliştirilen algoritma sonucu 166 öznitelik arasında 21' i seçilmiştir. 21 öznitelik kullanılarak 10 kez çapraz doğrulama sonucu elde edilen test başarısı ise % 90.1'dir.

### 3.6.Sonuçlar

Bu bölümde yeni bir öznitelik seçme yöntemi sunulmuştur. AdaBoost vb. yöntemlerin eğitim sürelerinin çok yüksek olması nedeniyle; hızlı, özgün ve başarılı bir yöntem geliştirmek amaçlanmıştır. Yöntem 166 öznitelikli musk veri seti üzerinde sınanmış; 21 öznitelik seçilerek bu öznitelikler ile % 90.1 sınıflandırma başarısı elde edilmiştir. Demiriz ve ark. [57] sunduğu çalışmada bulunan yöntemlerle karşılaştırıldığında iyi değerlerden birini elde edildiği gözlemlenmiştir.

## 4. HAREKET TANIMA

Bu bölümde, 2.bölümde değinilen sınıflandırma algoritması  $k$ -ort-ÇKF algoritmasının hareket tanıma problemine bir uygulaması yapılmıştır. Problemin önemi anlatılmış, son yıllarda geliştirilen yöntemler sunulmuştur. Daha sonra uygulamanın detayları ve sonuçlar verilmiştir.

### 4.1.Giriş

Son dönemde piyasada bulunan birçok ürün klasik işlevlerinin yanında akıllı olarak tabir edilen çeşitli özellikler kazanmışlardır. Gelecekte cihazların akıllı özelliklerinin hızla artması beklenmektedir. Tez çalışmasında ilk adımı gerçekleştirilen hareket tanıma problemi çözümünün ileride entegre bir sistem halinde, bahsedilen cihazlarda kullanılabileceği düşünülmektedir.

Hareketin tanınabilmesi için öncelikle kamera görüş açısında kontrolü gerçekleştirecek elin varlığının kontrol edilmesi gerekmektedir. Eğer ortamda kontrolü gerçekleştirecek el bulunuyorsa, elin arka plandan ayrılması sağlanmalı ve daha sonra elin koordinatları tespit edilmelidir. El hareket ettirildiğinde hareketler izlenerek elin koordinatlarına göre hareket vektörü çıkarılmalıdır. Bu sayede ön işlemler tamamlanmış olacaktır. Elde edilen hareket vektörleri gerçek zamanlı olarak, belirli hareketleri tespit etmek üzere oluşturulmuş sınıflandırcılara iletilmektedir. Tez çalışmasında el-kol hareketleriyle görüntü kontrolünü sağlayacak, yeni bir yöntem araştırılmıştır.  $k$ -ort-ÇKF algoritması daha önce bu problem üzerine uygulanmamıştır. Sınıflandırma işlemi sonrasında elde edilen çıktıya göre tanımlanmış olayın (ekrana halka çizdirilmesi) gerçekleşmesi sağlanmıştır. Yöntem cihazlarda kullanılması halinde ürünler için piyasada büyük bir rekabet gücü – üstünlük sağlayacaktır.

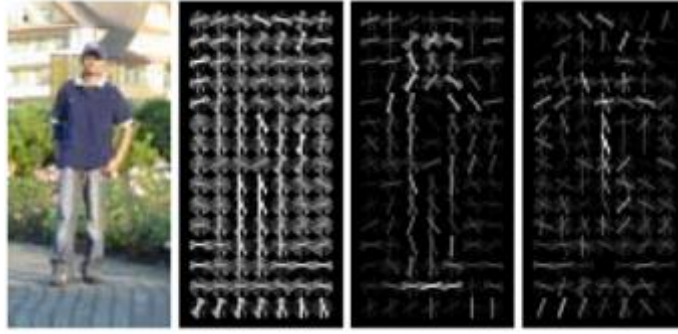
### 4.2.Literatür Özeti

Bir sisteme görüntü tabanlı girdi vermek bilgisayar görmesi araştırma konularının en önemlilerinden ve en çok çalışma yapılan alanlarından biridir. Yapılmış çalışmalar renk bölümleme, görüntü farklılaştırma, korelasyon, dağılım



analizleri, hareket analizleri, sınıflandırma gibi konuları içermektedir. Hareket tanıma sisteminde kullanılan elemanlara göre doğal olarak tanıma yöntemi de değişmektedir. Literatürde tezde izlenen elin bulunması, elin izlenmesi, hareket vektörünün çıkarılması, tanımlanan hareketler için sınıflandırıcıların oluşturulması gibi adımlara ait ayrı ayrı çalışmalar bulunduğu gibi bir kaç tanesinin bir arada bulunduğu çalışmalarda bulunmaktadır.

Viola ve Jones [34,55]'un çalışmaları gerçek zamanlı nesne bulma sistemleri için en önemli adımlardan biri olmuştur. Yöntem detaylarına Bölüm 3.2'den ulaşılabilir. Nesne tanıma konusunda literatürdeki bir diğer önemli çalışma ise 2005 yılında Delal ve Triggs tarafından yayınlanan çalışmadır [58]. Kısaca *HOG* olarak isimlendirilen öznelikler kullanılarak trafikteki yayalar tespit edilmiştir. Yerleri tespit için Destek Vektör Makinaları kullanılmıştır. *MIT* yaya veritabanı (*MIT* pedestrian database) üzerinde yapılan testlerde neredeyse mükemmel sonuçlar elde edilmiştir.



Şekil 4.1 HOG öznelikleri [58]

Mahmoudi ve Parviz [59] görüntüdeki eli bulan yöntemleri anlatmışlar ve bunların üstün özelliklerini incelemişlerdir. Yapılan çalışmada kullanılan algoritmaların gerçek zamanlı olarak kullanılmasının önemine değinilmiş ve gerçek zamanlı bir uygulamada saniyede 30 görüntünün işlenmesi gerektiği belirtilmiştir. Ten renginin elin yerini bulmada kullanışlı bir özellik olduğu belirtilmiştir. Literatürdeki çalışmalarda YUV, HSV ve YCbCr renk uzaylarının kullanıldığı görülmüştür. Çizelge 4.1'de el bulmada kullanılan yöntemler ve baskın özellikleri verilmektedir:

**Çizelge 4.1** El bulmada kullanılan yöntemler [59]

Yöntem	Kullanılan Özellik	Yöntemin Niteliği
CAMShift	Renk	Yanıltıcı varlıklı ve gürültülü bir ortamda izleme
Kalman Filtresi	Renk, Hareket	Sınıflandırma Başarı Oranı %86
Condensation	Dağılım(Kontur), Hareket	Eklemli ve kamufle olmuş bir objenin hızlı şekilde takibi
Icondensation	Renk, Dağılım(Kontur) Hareket	Karışık ve çeldirici el renklerinin bulunduğu ortamda güçlü ve hızlı izleme
Bayes Ağı	Renk, Yön, Hareket	Üst üste binen objelerin (el-yüz) güçlü şekilde ayrılması
Ortak Bayes Filtresi	Hareket, Şekil	Zayıf işaretlerin kullanılması

2009 yılında Mao ve ark. [60] çalışmalarında Viola ve Jones tarafından geliştirilen nesne bulma yöntemini [34,55] kullanarak karmaşık bir arka plana sahip görüntüdeki elin yerini tespit etmişlerdir. Elin yerinin bulunmasında ten renkleri kullanılmıştır. Saniyede 15 çerçeve görüntü işlenerek gerçek zamana yakın bir hızla el bulunmuştur. HSV ve YCbCr renk uzaylarıyla çalışılmış ve YCbCr renk uzayında daha iyi sonuçlar elde edilmiştir.

Mitra ve Acharya [61] literatür taramalarında bu problem için Saklı Markov Modellerinin, sonlu durum makinelerinin (finite state machines), ten renklerinin, parçacık filtreleme, optik akış, birleştirici yöntemler ve condensation gibi yöntemlerin kullanıldığından bahsetmişler ve bunlar hakkında genel bilgiler vermişlerdir.

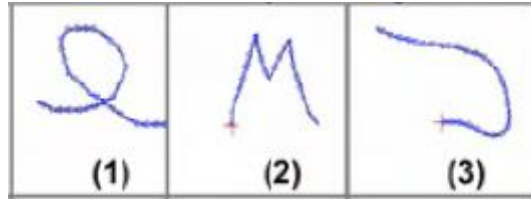
2011 yılında Dardas ve Georganas [62] çalışmalarında el hareketleriyle bir uygulamayı kontrol etmeye çalışmışlardır. Ten rengi ile ayırma ve görüntüdeki yüzün çıkarılması işlemlerinden sonra el yeri tespit edilmiştir. Çok sınıflı destek vektör makineleri ve *bag of features* yöntemleriyle sistem eğitilmiştir. Eğitim esnasında *k*-ortalama kümeleme yöntemi de kullanılmıştır. Daha sonra eğitilen sisteme görüntüden elde edilen el hareketleri verilmiş ve tanıma yapılarak istenen bazı belirli çıktılar elde edilmiştir.



Şekil 4.2 Dardas ve Georganas tarafından kullanılan hareketler [62]

Liu N. ve ark. [63] çalışmalarında 1 hareketleri ile harf tanıma yapmışlardır. Görüntüden otomatik olarak elin yeri belirlenmiştir. Elin yeri belirlenirken YUV renk uzayı kullanılmıştır. Daha sonra hareket vektörü çıkarılmış ve Saklı Markov Modeli (Hidden Markov Model- *HMM* ) ile harf tanıma yapılmıştır. 30 videoluk bir veri tabanı kullanılarak A'dan Z ye 26 harf %90 oranıyla tanınmıştır.

Sarrafzadeh A. ve ark [64] çalışmada yapay sinir ağları kullanmışlardır. Gerçek zamanlı olarak geliştirilen sistemde %98.71 'lik bir başarı elde etmişlerdir. Şekil 4.3'teki 3 hareketi tanımayı amaçlamışlardır.



Şekil 4.3 Sarrafzadeh ve ark. tarafından kullanılan hareketler

Liu L. ve ark. [65] çalışmalarında el hareketlerini tanıyan ucuz ve basit bir sistem geliştirmişlerdir. Sistemleri pasif ve aktif olmak üzere iki duruma sahiptir. Pasif durumdayken aktive edici hareketin gelmesi beklenmektedir. Aktive edici hareket geldiğinde sistem aktif duruma geçer ve önce eli bulur. Daha sonra önceden belirlenmiş hareketlerden biri yapıldığında sistem istenilen bir çıktı verir. Aktif durumda 3 dakika boyunca bir hareket tanınmazsa sistem pasif duruma

geçer. Aktive edici hareket olarak saat yönünde çember çizme hareketi kullanılmıştır. İnsan varlığının görüntüde olup olmadığı ten rengi kullanılarak belirlenmiştir. Çalışmada 8 adet hareket tanınmaya çalışılmıştır. Hareketler ve tanıma oranları aşağıdaki gibidir.

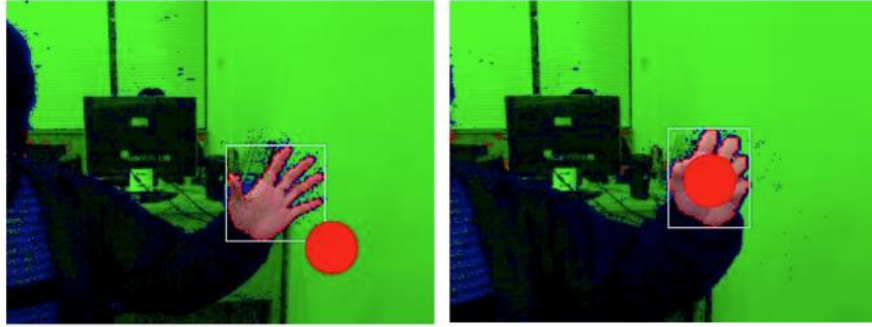
**Çizelge 4.2** Lie ve ark. tarafından kullanılan hareketler ve tanıma oranları

<b>Hareket</b>	<b>Tanıma Oranı</b>
Aktivasyon	86 %
Sol	91 %
Sağ	88 %
Yukarı	85 %
Aşağı	87 %
Giriş	79 %
Çıkış	76 %
Yakınlaşma	96 %
Uzaklaşma	94 %

Burande ve ark. [66] el hareketlerini bir web kamera ile tanımaya çalışmışlardır. Kalman filtreleme, Saklı Markov Modeli ve grafik eşleme yöntemleri kullanılmıştır. Sisteme öncelikle tanınacak veriler girilir. Daha sonra web kameradan görüntüler elde edilir. Elde edilen görüntü düzgünleştirilmiş, tene ait pikseller etiketlenmiş ve gürültüden arındırılmıştır. Bir sonraki adımda tanıma ve tanımaya göre yapılacak çıktı işlemi gerçekleştirilmiştir. Teni bulmak için *HSI* (Hue-Saturation-Illumination) değerleri kullanılmıştır. Yapay kırmızı ışık kaynağı kullanılmış ve bazı normalleştirmeler yapılmıştır.

Yeh ve ark. [67] el hareketlerini tanımayı amaçlayan bir çalışma yayınlamışlardır. Test için basit bir top tutma-sürükleme oyununda denemeler yapılmıştır. Her çalışmada olduğu gibi öncelikle görüntüdeki elin yeri tespit edilmiştir. İşlem kolaylığı için öndeki ve arkadaki objeler ayrılmış; daha sonra öndeki objelerden elin yeri bulunmuştur. Daha sonra tanıma yapılmıştır. Sistem

320x240 çözünürlükte Microsoft LifeCam VX-6000 marka tek kameraya sahiptir. Uygulama Pentium IV Core 2 Duo 3.4 GHz CPU, 1 GB Ram, Windows XP işletim sistemli bir bilgisayarda test edilmiştir. Sistem saniyede ortalama 27 görüntü ile çalışmıştır ve CPU kullanım oranı % 30 seviyesindedir. Tanıma başarısı ise normal ışık altında %90 seviyesindedir. Halojen lambayla aydınlatılmış ortamda başarı düşmektedir.



Şekil 4.4 Top yakalama uygulamasından bir görüntü [67]

Segen ve Kumar [68] aynı noktaya odaklanmış 2 kamerayla elde edilen 3 boyutlu yakın çekimden 3 el hareketini 1/500'den daha az hata ile tanımışlardır. Çalışmalarında 2 adet 200Mhz SGI Indy bilgisayar kullanılmıştır. Her iki CPU' da %30 oranla kullanarak saniyede 60 çerçeve görüntü işlenmiştir.

### 4.3. Hareket Tanımadaki Kullanılan Yöntem ve Cihazlar

Hareket tanıma sistemlerinde bir çok farklı cihaz ve farklı girdi tipleri kullanılmaktadır. Bu durum sistemin kullanımını ve geliştirilen hareket tanıma algoritmalarının yapısını etkilemektedir.

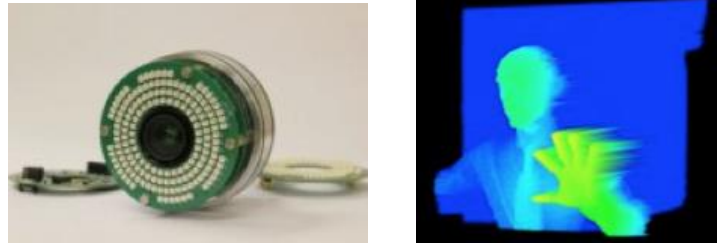
Temelde iki tip hareket tanımlanmaktadır:

- 1- **Çevrim dışı Hareketler:** Sistemi aktive etmekte kullanılan hareketlerdir.
- 2- **Çevrim içi Hareketler:** Sistem aktif olduktan sonra kullanılan hareketlerdir.

Hareketlerin bilgisayarlar tarafından algılanmasında kullanılan genel bazı girdi araçları ise şunlardır [69] :

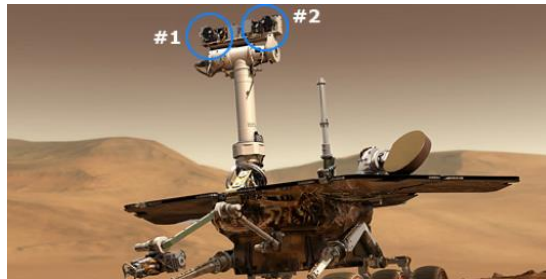
• **Kablolu Eldiven:** Manyetik ve eylemsizlik izleme cihazları kullanılarak bilgisayara girdi sağlanır.

• **Derinlik Duyarlı Kameralar:** 3 boyutlu görüntü oluşturmak için yapısal ışık (structured light) ya da time of flight (*tof*) kameralar kullanılır. *tof* kameralar lazerin bir objeden çarpıp, geri dönmesi arasındaki süreyi esas alarak 3 boyutlu modeli çıkartırlar. Bir kaç metreden 60 m ' ye kadar çalışabilmektedir ve uzaklık duyarlılığı 1 cm'dir. Ticari olanlarında 320 x 240 piksel çözünürlüğünde ve saniyede 100 çerçeve görüntü verebilenleri mevcuttur. Yapısal ışık ise lazere alternatif bir uygulamadır. Tüm objede lazer gezdirmek yerine bir ışık deseni gönderilerek 3 boyutlu görüntü elde edilir. Hızlı, hassas ve hareketli cisimleri algılayabilirler.



Şekil 4.5 *tof* kamera ve bu kameradan elde edilmiş bir görüntü [70]

• **Stereo Kameralar:** Aynı yere bakan iki kamera kullanılarak 3 boyutlu görüntüyü tahmin etmek amacıyla kullanılır. Buradaki yöntem insanda derinlik algısının aynı noktaya bakan iki göz ile elde edilen iki resmin beyinde birleştirilmesi ile aynı mantığa sahiptir.



Şekil 4.6 Nasa tarafından kullanılan bir stereo kamera

- **Kontrolör Tabanlı Cihazlar:** Bilgisayar tarafından algılanan ve vücudun bir parçası gibi kullanılan cihazlardır. Son zamanlardaki en popüler örneği Nintendo *Wii* için kullanılan uzaktan kumandadır.



Şekil 4.7 *Wii* uzaktan kumanda

- **Tek Kamera:** Tek kamera ile tanıma görüntü tanıma teknikleri kullanılarak yapılmaktadır. Derinlik algılaması olmadığından tek kamera ile hareket tanıma sistemlerinin derinlik algılaması yapan sistemlere göre etkinliği azdır.

Tanıma başarısı girdinin ve tanıma yönteminin kalitesine göre değişmektedir. Yöntemler iki temel gruba ayrılabilir [69]:

**3 Boyutlu Model Tabanlı Yöntemler:** 3 boyutlu yüzeyler yardımıyla hacimsel bir model oluşturulur. Vücudun belli bölgeleri bu hacimsel modelden çıkartılır ve eklemlerin açısı, elin pozisyonu vb. bilgiler elde edilir.

**Görüntü Tabanlı Modeller:** Fotoğraf ya da videodan elde edilen görüntülerin işlenmesiyle yapılan yöntemlerdir. Tek ya da çift görüntü kullanılabilir.

Arka plandaki farklılıklar, ışık değişimi, görüntüdeki gürültüler hareket tanımayı zorlaştıran en önemli etmenlerdir. Ayrıca uzaklığın değişmesi, kamera çözünürlüğü tanımayı önemli derecede etkilemektedir. Kullanılan yöntemler her kamerada çalışmayabilir. Bu nedenle sistemdeki elemanların özellikleri oldukça önemlidir.

#### 4.4. Matematiksel Programlama ile Hareket Tanıma Üzerine Bir Uygulama

Hareketi tanıyabilmek için öncelikle elin izlenmesi, hareket vektörünün çıkarılması, tanımlanan hareketler için sınıflandırıcıların oluşturulması adımları izlenmelidir.

El hareketlerinin anlamlandırılabilmesi için öncelikle kameranın görüş açısında kontrolü gerçekleştirecek elin varlığının kontrol edilmesi gerekmektedir. Eğer ortamda kontrolü gerçekleştirecek el bulunuyorsa, elin arka plandan ayrılması sağlanmalı ve daha sonra elin koordinatları tespit edilmelidir. El hareket ettirildiğinde hareketler izlenerek elin koordinatlarına göre hareket vektörü çıkarılması ile ön işlemler tamamlanmaktadır. Elde edilen hareket vektörleri gerçek zamanlı olarak, belirli hareketi tespit etmek üzere oluşturulmuş sınıflandırıcılara iletilmektedir. Sınıflandırma işlemi sonrasında elde edilen çıktıya göre tanımlanmış ekran çıktısı verilmektedir.

##### 4.4.1. Elin bulunması

Önerilen sistemde bir nesnenin ilk anda el olarak tanınması için iki kritere sahip olması beklenmektedir. Bunlar ten rengi ve harekettir. Görüntüde ten renginde olan, hareket eden nesnelere el olarak tanımlanmaktadır. Uygulamada elin şekilsel faktörleri ile ilgilenilmemiştir.

Eli diğer nesnelere ayıran en önemli özellik avuç içinin renk değeridir. Avuç içinin renk özünü (*hue*) değeri herkes için, tüm ırklar dahil, birbirine çok yakındır. Eli tanıyabilmek için tüm avuç içinin renk özünü değerlerini kapsayacak bir renk aralığı tanımlanmıştır. Görüntü sistemlerinde farklı ihtiyaçlara göre birçok renk uzayı kullanılmaktadır. *TSL* (Tint, Saturation, Luma), *HSV* (Hue, Saturation, Value) ve *YCbCr* (Parlaklık, Mavi Renklilik, Kırmızı Renklilik) renk uzaylarının yaygın olarak kullanıldığı görülmektedir. Bu çalışmada *YCbCr* renk uzayı kullanılmıştır. Ten rengini ayırmak için aşağıdaki değer aralıkları seçilmiştir. *Y* parlaklık değeri olduğundan dikkate alınmamıştır.

$$\begin{aligned} 90 &< Cb < 130 \\ 140 &< Cr < 210 \end{aligned}$$





**Şekil 4.8** Farklı  $Y$  değerlerine göre  $YCbCr$  renk uzayı

Eli bulmada kullanılan bir diğer özellik olan hareket, el bulma işleminin başarıyla gerçekleştirilmesi için oldukça önemlidir. Hareket özelliğinin kullanılmasıyla görüntü üzerindeki sadece hareketli noktalara odaklanılmıştır. Bu sayede hem sistemin eli doğru olarak tespit etme yüzdesi artmış hem de işlemsel yük büyük ölçüde azalmıştır. Şekil 4.9'da yapılan denemeler sonucu elde edilen görüntü örnekleri gösterilmektedir. Soldaki görüntü ten rengine ait pikselleri göstermektedir. Ortadaki gerçek görüntü ve sağ taraftaki hareketli piksellerin gösterildiği görüntüdür.



**Şekil 4.9** Renk ile ayırma, orjinal görüntü ve hareket ile ayırma

Görüntü üzerinde arama yapılırken sadece hareketli ve ten renginde olan kısımlarda arama yapılmış, hem ten renginde hem de hareketli noktalar takip edilmiştir.

Bu aşamadan sonra, daha önceden tanımlanmış belirli hareketi tanıyabilmek için hareket vektörünün çıkarılması aşamasına gelinir.

#### 4.4.2. Hareket vektörlerinin çıkarılması ve anlamlandırılması

Hareketin uygulama tarafından tanınabilmesi için hareket vektörü sınıflandırıcı fonksiyona girdi olarak verilmiştir. Kameradan sisteme sürekli olarak belli bir hızda görüntü akışı gerçekleşir. Yeri bulunan elin iki boyutlu uzayda  $x$  ve  $y$  koordinatları hesaplanmıştır. Ardarda gelen görüntülerdeki  $x$  ve  $y$  koordinatları belli sayıda görüntü için tutulur. Bu sayede ayırt edilecek hareket vektörleri hesaplanır. Bu çalışmada son 30 görüntü için  $x$  ve  $y$  koordinatları tutulmuş, dolayısıyla hareket vektörleri 60 öznitelikten oluşturulmuştur. Şekil 4.10'da kırmızı renk ile gösterilen vektör oluşturulan hareket vektörlerine bir örnektir.



Şekil 4.10 Çıkarılan bir hareket vektörü

Hareket vektörlerinin anlamlandırılmasında  $k$ -ort-ÇKF algoritması kullanılmıştır. Her yeni gelen görüntüde güncellenen hareket vektörü sınıflandırıcıya sokulmuştur. Tanınan hareket saat yönünün tersindeki halka hareketidir. Halka hareketi olan ve olmayan hareket vektörleri ile eğitim yapılarak sınıflandırıcı fonksiyonlar elde edilmiş ve halka hareketini tanıdığına ise ekrana yeşil renkli bir halka çizdirilmiştir.



Şekil 4.11 Halka hareketinin tanınması

#### 4.5. Hesapsal Sonular

Tek hareket iin elde edilen bu sonular, nerilen yntemin geliřtirilen sistemde bařarı ile kullanılabilir olduėunu gstermektedir. Ayrıca sınıflandırıcı yne duyarlıdır ve saat ynnde izilen halka hareketine cevap vermemektedir. Bu da farklı ynlerde yapılan aynı hareketlere farklı iřlemler atanabileceėini gstermektedir. Grnt iřleme hızı uygun ışık altında 28-30 *fps*'dir.

**izelge 4.3** Halka hareketi test bařarıları

Yntem	Halka hareketi
SMO	99.72
IBK	99.8
J48	99.7
Naif Bayes	96.07
<i>k</i> -ort-KF	99.66

Oluřturulan veri setinde 9619 adet saat yn tersi halka olmayan hareket ve 21 adet saat yn tersi halka hareketi mevcuttur. Dolayısıyla iki sınıflı bir problem oluřturulmuřtur. Veri setinin yapısından dolayı test doėruluėu yanında kesinlik ve duyarlılık gibi deėerlerin de incelenmesi yararlı olacaktır. yle ki SMO algoritması tm verileri halka hareketi deėil olarak nitelendirmiřtir. Ancak hesaplanan bařarı %99.72 gibi olduka yksek bir deėerdir. Fakat bu sınıflandırıcı gerek sistemde denendiėi takdirde hibir halka hareketini tanıyamayacaktır.

**Çizelge 4.4** İki sınıflı bir problem için hatalı sınıflandırma matrisi

		Gerçek Sınıf	
		A	B
Tahmin Edilen Sınıf	A	tp	fp
	B	fn	tn

Kesinlik ve duyarlılık değerleri hatalı sınıflandırma matrisi Çizelge 4.4' teki gibiyken aşağıdaki şekilde hesaplanır:

$$Kesinlik = \frac{tp}{tp + fn}$$

$$Duyarlılık = \frac{tp}{tp + fp}$$

**Çizelge 4.5** Halka hareketi için kesinlik değerleri [71]

Yöntem	Halka hareketi
SMO	0.0
IBK	57.6
J48	45
Naif Bayes	5.2
k-ort-ÇKF	37.2

**Çizelge 4.6** Halka hareketi için duyarlılık değerleri [71]

Yöntem	Halka hareketi
SMO	0.0
IBK	71.4
J48	42.8
Naif Bayes	100
<i>k</i> -ort-ÇKF	76.19

Kesinlik pozitif olarak tahminlenmiş pozitif sayısının pozitif olarak atanmış tüm örneklere oranıyken, duyarlılık pozitif olarak atanmış pozitif sayısının tüm pozitiflere oranıdır. Duyarlılık ve kesinliği birlikte ifade edebilmek amacıyla *f* değeri olarak adlandırılan bir ölçüt geliştirilmiştir.

$$f = \frac{2 * duyarlılık * kesinlik}{duyarlılık + kesinlik}$$

**Çizelge 4.7** Yöntemler için hesaplanan *f* değerleri

Yöntem	Halka hareketi
SMO	0.0
IBK	63.7
J48	43.8
Naif Bayes	9.8
<i>k</i> -ort-ÇKF	49.99

Elde edilen *f* değerleri Çizelge 4.7 'de gösterilmektedir.

#### 4.6. Sonuçlar

Bu bölümde ise *k*-ort-ÇKF sınıflandırma yöntemi çok güncel bir konu olan hareket tanıma problemine uygulanmıştır. Yöntemin her saniyede 30 adet hareket vektörünü test edebilecek hıza sahip olduğu gözlemlenmiştir (Kameradan alınan maksimum görüntü sayısı saniyede 30 çerçeve görüntüdür). Yöntem

başarılı bir şekilde uygulamaya dökülmüş ve halka hareketinin tanınabilir olduğu gösterilmiştir. Oluşturulan hareket veri seti 2 sınıflı olmakla birlikte, dengesiz bir veri setidir. Bu nedenle sınıflandırma başarısının yanında kesinlik, duyarlılık ve  $f$  değerleri de incelenmiştir. Tüm yöntemler sınıflandırma başarısı açısından iyi sonuçlar verdiği hesaplanmıştır. Fakat bu değer aldatıcıdır. Örneğin SMO algoritması tüm verileri tek sınıfa atmasına rağmen test başarısı % 99.72 'dir. Ancak kesinlik ve duyarlılık değerleri bu algoritma için % 0.0 olarak hesaplanmıştır.  $k$ -ort-ÇKF algoritması duyarlılık değeri %76.19 olarak hesaplanmış ve iyi bir değer elde edilmiştir.  $k$ -ort-ÇKF algoritması kesinlik değeri ise % 37.2'dir. Algoritmaların kesinlik değerlerinin düşük olması kamera karşısında halka hareketi yapılmadığında da sistemin halka hareketi yapıldığını varsayabileceğini göstermektedir. Duyarlılık değerleri ise halka hareketi yapıldığında tanınıp tanınmayacağını ifade eder. Duyarlılık ve kesinlik değerlerinin bir karması olan  $f$  değeri ise  $k$ -ort-ÇKF için % 49.99 olarak hesaplanmıştır.

## 5. SONUÇ VE ÖNERİLER

Tez çalışmasında kümeleme temelli sınıflandırma yöntemlerinin karşılaştırılması, öznitelik seçimi ve hareket tanıma probleminin çözümü üzerinde durulmuş ve sonuçlar bölüm sonlarında sunulmuştur.

Tüm sonuçlar gelecek vaad etmektedir. *EM-ÇKF* algoritmasının hız nedeniyle çok pratik olmadığı anlaşılmıştır. Fakat iyi sınıflandırma değerleri elde etmek için eğitimin uzun sürmesi göze alınabilir. Geliştirilmiş bütünsel *k-ort-ÇKF* ise hem hız hem de sınıflandırma başarısı açısından *k-ort-ÇKF* algoritması ile benzer sonuçlar elde etmiştir. Ancak küme sayısı arttığında *k-ort-ÇKF* algoritması hızı geliştirilmiş bütünsel *k-ort-ÇKF* algoritması hızından daha iyidir. Tez çalışmasının ileriki adımında 3. bölümde sunulan yöntem ile el şekil faktörlerinin seçilerek el yerinin bulunmasında bu şekil faktörlerinin de kullanılması amaçlanmaktadır. 4. Bölümde sunulan halka hareketine başka hareketlerde eklenerek bunların da tanınması amaçlanmaktadır ve yöntemin entegre bir sisteme dönüştürülmesi için çalışılacaktır.

## KAYNAKÇA

- [1] Rastogi R. ve Shim K., PUBLIC:A decision tree classifier that integrates, *Data Mining and Knowledge Discovery*, 4, 315-344, 2000.
- [2] Han J. ve M. Kamber,. San Francisco: Morgan Kaufmann, 2001.
- [3] Gehrke J., Ramakrishnan R., ve Ganti V., RainForest a framework for fast decision tree, *Data Mining and Knowledge Discovery*, 4(2000) , 127-162.
- [4] Öztürk G. ve Gasimov R., Separation via Polyhedral Conic Functions, *Optimization Methods and Software*, 21(4) , 2006.
- [5] Çiftçi M. T., Büyük boyutlu sınıflandırma problemleri için matematiksel programlama yaklaşımları, Yüksek Lisans Tezi, Anadolu Üniversitesi, FBE, Eskişehir, 2011.
- [6] Öztürk G., Sınıflandırma problemleri için yeni bir matematiksel programlama yaklaşımı, Doktora Tezi, Osmangazi Üniversitesi, Fen Bilimleri Enstitüsü, Eskişehir, 2007.
- [7] Zisserman A., Clustering and mixture models, 2013, C19 Machine Learning, Ders Notları.
- [8] Cortes C. ve Vapnik V., Support vector networks, *Machine Learning*, 20, 173-297, 1995.
- [9] Bennett K.P. ve Mangasarian O.L., Robust linear programming discrimination of two linearly inseparable sets, *Optimization Methods and Software*, 1, 23-34, 1992.
- [10] Asorino A. ve Gaudioso M., Polyhedral separability thorough Succesive LP, *Journal of Optimization Theory and Applications*, 112(2) , 265-293, 2002.
- [11] Bagirov A. M., Max-min separability, *Optimization Methods and Software*, 20(2-3) , 271-290, 2005.
- [12] Üney F. ve Türkay M., A mixed-integer programming approach to multi-class data classification problem, *European Journal Of Operational Research*, 173(3) , 910-920, 2006.
- [13] Sibson R., SLINK: an optimally efficient algorithm for the single-link cluster method, *The Computer Journal (British Computer Society)* , 16(1) , 30-34, 1973.
- [14] Defays D., An efficient algorithm for a complete link method, *The Computer Journal (British Computer Society)* , 20(4) , 364-366, 1977.
- [15] MacQueen J.B., Some methods for classification and analysis of multivariate observations, Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability 1. University of California Press.(281-297) , 1967.
- [16] Lloyd S., Least squares quantization in PCM, *IEEE Transactions on Information Theory*, 29(2) , 129-137, 1982.
- [17] Bagirov A.M., Modified global k-means algorithm for minimum sum-of-squares clustering problems, *Pattern Recognition*, 41(10) , 3192-3199, 2008.



- [18] Dempster A.P., Laird N.M., ve Rubin D.B., Maximum likelihood from incomplete data via EM algorithm, *Journal of the Statistical Society*, 39(1), 1-38.
- [19] Ester M. ve ark., A density-based algorithm for discovering clusters in large spatial databases with noise, *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, 226-231, 1996.
- [20] Ankerst M., Breunig M., Kriegel H.P., ve Sander J., OPTICS: Ordering points to identify the clustering structure, *ACM SIGMOD international conference on Management of data ACM Press*, 49-60, 1999.
- [21] Barros R.C., Basgalupp P.M., Carvalho C.P.L.F., ve Freitas A., A survey of evolutionary algorithms for decision-tree induction, *IEEE Transactions on Systems, Man and Cybernetics-Part C: Applications and Reviews*, 42(3), 2012.
- [22] Pradhan A., Support Vector Machine - A Survey, *International Journal of Emerging Technology and Advanced Engineering*, 2(8), 2012.
- [23] Alpaydın E., *Yapay Öğrenme*, 1st ed. İstanbul: Boğaziçi Üniversitesi Yayınevi, 2011.
- [24] (2013, Temmuz) Wikipedia Voronoi diagram. [Online]. [http://en.wikipedia.org/wiki/Voronoi\\_diagram](http://en.wikipedia.org/wiki/Voronoi_diagram)
- [25] Ricco S., Clustering and the EM algorithm, 2007, CPS 271, Ders Notu.
- [26] Gasimov R. ve Öztürk G., Separation via polyhedral conic functions, *Optimization Methods and Software*, 21(4), 2006.
- [27] (2013, Ağustos) UCI machine learning repository. [Online]. <http://archive.ics.uci.edu/ml/>
- [28] Zhong P. ve Fukushima M., A new multiclass support vector algorithm, *Optimization methods and software*, 21(3), 359 - 372, 2006.
- [29] Xu J. ve Zhang X., A multiclass kernel perceptron algorithm, *International Conference on Neural Networks and Brain*, 717 -721, 2005.
- [30] Hsu C.W. ve Lin C.J., A comparison of methods for multiclass support vector machines, *IEEE Transactions on Neural Networks*, 13(2), 415 - 425, 2002.
- [31] Sun Z., Bebis G., ve Miller R., Object detection using feature subset selection, *The Journal of the Pattern Recognition Society*, 37, 2004.
- [32] Watanabe S., *Pattern recognition: human and mechanical*. New York: Wiley, 1985.
- [33] Dash M. ve Liu H., Feature selection for class, *Intelligent Data Anal*, 1(3), 131-156, 1997.
- [34] Viola P. ve Jones M., Rapid object detection using a boosted cascade of simple features, *Proceedings on Computer Vision and Pattern Recognition*, 2001.
- [35] Sun Z., Bebis G., ve Miller R., Boosting object detection using feature selection, *IEEE International Conference on Advanced Video and Signal Based Surveillance*, 290-296, 2003.

- [36] Sun Z., G. Bebsi, ve Miller R. , Evolutionary gabor filter optimization with application to vehicle detection, *IEEE International Conference on Data Mining*, 307-314, 2003.
- [37] Gyaourova A. ve Pavlidis I. Bebis G., Infrared and visible image fusion for face recognition, *European Conference on Computer Vision*, 2004.
- [38] Collins R. ve Liu Y., On-line selection of discriminative tracking features, *IEEE International Conference on Computer Vision*, 2003.
- [39] Jain A., Duin R., ve Mao J., Statistical pattern recognition: a review, *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(1) , 4 –37, 2000.
- [40] Siedlecki W. ve Sklansky J., On automatic feature selection, *Int. J. Pattern Recognition Artif. Intell.*, 2(2) , 197-220, 1988.
- [41] Marill T. ve Green D., On the effectiveness of receptors in recognition systems, *IEEE Trans. Inform. Theory*(9) , 11-17, 1963.
- [42] Stearns S., On selecting features for pattern classifiers, *The Third International Conference of Pattern Recognition*, 71-75, 1976.
- [43] Siedlecki W. ve Sklansky J., A note on genetic algorithm for large-scale feature selection, *Pattern Recognition Lett.* , 10, 335-347, 1989.
- [44] Kira K. ve Rendell L., A practical approach to feature selection, *In Proc. of the 9th Int. Conf. on Machine Learning*, 249-256, 1992.
- [45] Kononenko I., Estimating attributes: analysis and extensions of relief, *In Proc. of the European Conf. on Machine Learning*, 171 -182, 1994.
- [46] Molina L.C., Belanche L., Nebot A., ve Girona J., Feature selection algorithms: A survey and experimental evaluation, *IEEE International Conference on Data Mining, Proceedings*, 306 - 313, 2003.
- [47] Pearson K., On lines and planes of closest fit to systems of points in space, *Philosophical Magazine*, 2(11) , 559–572, 1901.
- [48] Hotelling H., Analysis of a complex of statistical variables into principal components, *Journal of Educational Psychology*, 417 - 441 ve 498 - 520, 1933.
- [49] Fisher R.A., The use of multiple measurements in taxonomic problems, *Annals of Eugenics*, 7(2) , 179–188, 1936.
- [50] Comon P., Independent component analysis: A new concept, *Signal Processing*, 3, 287 - 314, 1994.
- [51] Ding S.F., Jin F.X., ve Shi Z.Z., Information feature compression based on partial least squares, *Journal of Computer Aided Design & Computer Graphics*, 2, 368–371, 2005.
- [52] Ding S., Jia W., Su C., Jin F., ve Shi Z., A survey on statistical pattern feature extraction, *Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence*, 5227, 701 - 708, 2008.
- [53] Günel S., Örüntü tanıma uygulamalarında alt uzay analiziyle öznetelik seçimi ve sınıflandırma, Doktora Tezi, Eskişehir Osmangazi Üniversitesi, FBE, Elektrik ve Elektronik Mühendisliği Anabilim Dalı, Telekomünikasyon Bilim Dalı, 2008.

- [54] Sezer O.G., Ertüzün A., ve Erçil A., Bağımsız bileşenler analizi ile otomatik kumaş kalite kontrolü.
- [55] Viola P. ve Jones M., Robust real-time face detection, *International Journal of Computer Vision*, 57(2) , 137–154, 2004.
- [56] Xu ve Arun, Adaboost and object detection, Ders Sunumu.
- [57] Demiriz A., Bennett K.P., ve Taylor J.S., Linear programming boosting via column generation, *Machine Learning*, 46(1-3) , 225 - 254, 2002.
- [58] Delal N. ve Triggs B., Histogram of oriented gradients for human detection, Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1, 2005.
- [59] Mahmoudi F. ve Parviz M., Visual hand tracking algorithms, *Geometric Modeling and Imaging--New Trends*, 228 -232, 2006.
- [60] Mao G.Z., Wu Y.L., Hor M.K., ve Tang C.Y., Real-time hand detection and tracking against complex background, *Intelligent Information Hiding and Multimedia Signal Processing, 2009. IIH-MSP '09*, 2009.
- [61] Mitra S. ve Acharya T., Gesture recognition: A survey, *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 37(3) , 311 - 324, 2007.
- [62] Dardas N.H. ve Georganas N.D., Real-Time hand gesture detection and recognition using bag-of-features and support vector machine techniques, *IEEE Transactions on Instrumentation and Measurement*, 60(11) , 3592 - 3607, 2011.
- [63] Liu N., Lovell B.C., ve Kootsokos P.J., Evaluation of HMM training algorithms for letter hand gesture recognition, *International Symposium on Signal Processing and Information Technology*, 648 - 651, 2003.
- [64] Sarrafzadeh A., Shanbahzadeh J., Messom C., ve Johnson M., Innovative technologies for the creative industries: Advanced human-machine interfaces for dynamic performance effects, *4th International Conference on Innovations in Information Technology*, 223 - 227, 2007.
- [65] Liu L., Sang N., ve Yang S., A low-cost hand gesture human-computer interaction system, *IEEE International Conference on Consumer Electronics (ICCE)*, 229 - 300, 2012.
- [66] Burande A., Tugnaya M.R., ve Choudhary N.H., Advanced recognition techniques for human computer interaction, *International Conference on Computer and Automation Engineering - ICCAE* , 2010.
- [67] Yeh C.H. ve ark., Vision-based virtual control mechanism via hand gesture recognition, *Journal of Computers*, 21(2) , 55 - 66, 2010.
- [68] Segen J. ve Kumar S., Human-computer interaction using gesture recognition and 3D hand tracking, *International Conference on Image Processing*, 3, 188 - 192, 1998.
- [69] (2013, Ağustos) Wikipedia gesture recognition. [Online]. [http://en.wikipedia.org/wiki/Gesture\\_recognition](http://en.wikipedia.org/wiki/Gesture_recognition)

- [70] (2013, July) Peacock Technology Ltd. [Online].  
<http://www.peacocktech.co.uk/Case%20Studies/time%20of%20flight%20camera>
- [71] (2013, Ağustos) Wikipedia precision and recall. [Online].  
[https://en.wikipedia.org/wiki/Precision\\_and\\_recall](https://en.wikipedia.org/wiki/Precision_and_recall)