

ÖNSÖZ

Bu tez, Karadeniz Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı, Bilgisayar Bilimleri Yüksek Lisans Programı’nda yapılan bir çalışmadır. “Dokunmatik Ekranlarda Kaydırma Biyometrisine Dayalı Kimlik Doğrulama Çalışmaları” adlı çalışmada, 3 farklı veri setine ait kullanıcı topluluklarının dokunmatik ekran üzerinde yaptıkları kaydırma hareketleri çeşitli makine öğrenmesi yöntemleriyle tanınmaya çalışılmıştır.

Bu tez çalışması boyunca gerek konu seçimi ve gerekse çalışmaların yürütülmesi sırasında bilgi ve deneyimlerini bana aktaran, danışman hocam Dr. Öğr. Üyesi Murat AYKUT’a ilgi, destek ve tecrübelerini esirgemediğinden dolayı teşekkür ediyorum.

Son olarak hayatım boyunca varlıkları ve destekleriyle bana güç veren ve her zaman yanımda olan anneme, babama ve kardeşlerime şükranlarımı sunarım.

Orhan SİVAZ
Trabzon 2021

TEZ ETİK BEYANNAMESİ

Yüksek Lisans Tezi olarak sunduğum “Dokunmatik Ekranlarda Kaydırma Biyometrisine Dayalı Kimlik Doğrulama Çalışmaları” başlıklı bu çalışmayı baştan sona kadar danışmanım Dr. Öğr. Üyesi Murat AYKUT’un sorumluluğunda tamamladığımı, çalışmaları kendim yaptığımı, başka kaynaklardan aldığım bilgileri metinde ve kaynakçada eksiksiz olarak gösterdiğimi, çalışma sürecinde bilimsel araştırma ve etik kurallara uygun olarak davrandığımı ve aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul ettiğimi beyan ederim. 14/12/2020

Orhan SİVAZ

İÇİNDEKİLER

	<u>Sayfa No</u>
ÖNSÖZ.....	III
TEZ ETİK BEYANNAMESİ.....	IV
İÇİNDEKİLER.....	V
ÖZET.....	VIII
ŞEKİLLER DİZİNİ.....	X
TABLolar DİZİNİ.....	XI
SEMBOLLER DİZİNİ.....	XII
1. GENEL BİLGİLER.....	1
1.1. Giriş.....	1
1.2. Dokunmatik Ekranlarda Kimlik Doğrulama Sistemleri.....	2
1.2.1. Fizyolojik Biyometri.....	3
1.2.2. Davranışsal Biyometri.....	4
1.3. Dokunmatik Ekranlarda Kaydırma Biyometrisi.....	5
1.4. Literatür Taraması.....	6
1.5. Veri Setleri.....	10
1.5.1. Serwadda Veri Seti.....	10
1.5.2. JSS18 Veri Seti.....	11
1.5.3. Umdaa Veri Seti.....	12
1.6. Ön İşlemler.....	12
1.7. Formüle Dayalı Özellik Çıkartma.....	13
1.8. Öğrenmeye Dayalı Özellik Çıkartma.....	15
1.8.1. Doğrusal Özellik Çıkartma Yöntemleri.....	15
1.8.1.1. Temel Bileşen Analizi.....	15
1.8.1.2. Doğrusal Diskriminant Analizi.....	17
1.8.1.3. Spektral Regresyon - Doğrusal Diskriminant Analizi.....	19
1.8.1.4. Yerellik Koruyucu İzdüşüm.....	20
1.8.2. Doğrusal Olmayan Özellik Çıkartma Yöntemleri.....	23
1.8.2.1. Çekirdek Kavramı.....	23
1.8.2.2. Çekirdek Temel Bileşen Analizi.....	24
1.8.2.3. Çekirdek Diskriminant Analizi.....	26
1.8.2.4. Spektral Regresyon - Çekirdek Diskriminant Analizi.....	28

1.8.2.5.	Çekirdek Yerellik Koruyucu İzdüşüm.....	29
1.9.	Sınıflandırma Yöntemleri	31
1.9.1.	Destek Vektör Makineleri	31
1.9.2.	En Küçük Kareler - Destek Vektör Makineleri	35
1.9.3.	Gauss Karışım Modeli.....	36
1.9.4.	K-En Yakın Komşu Algoritması	38
1.9.5.	Yapay Sinir Ağları.....	39
1.9.6.	Evrişimsel Sinir Ağları.....	42
1.9.6.1.	Evrişim Katmanı.....	43
1.9.6.2.	Havuzlama Katmanı	43
1.9.6.3.	Düzleştirme Katmanı	44
1.9.6.4.	Tam Bağlantılı Katmanı	44
1.9.6.5.	Transfer Öğrenme.....	44
1.9.7.	Füzyon İşlemi.....	45
1.10.	Performans Ölçüm Metriği	46
2.	YAPILAN ÇALIŞMALAR, BULGULAR VE İRDELEME	47
2.1.	Giriş.....	47
2.2.	Özellik Belirleme Çalışmaları	48
2.2.1.	Serwadda Veri Seti Üzerinde Özellik Belirleme Çalışmaları	48
2.2.1.1.	Serwadda Veri Seti Üzerinde Eklenen Özelliklerin Performansa Etkisi.....	49
2.2.2.	JSS18 Veri Seti Üzerinde Kullanılan Özellikler	50
2.2.3.	Umdaa Veri Seti Üzerinde Kullanılan Özellikler.....	51
2.3.	Özellik Çıkartma ve Sınıflandırma Yöntemleri ile Performans İyileştirme.....	51
	Çalışmaları.....	51
2.3.1.	Serwadda Veri Seti Üzerinde Sınıflandırma Çalışmaları	52
2.3.1.1.	Serwadda Veri Setinin EKK-DVM ve GMM ile Sınıflandırılması	53
2.3.2.	JSS18 Veri Seti Üzerinde Sınıflandırma Çalışmaları.....	58
2.3.2.1.	JSS18 Veri Setinin YSA ve EKK-DVM ile Sınıflandırılması	58
2.3.3.	Umdaa Veri Seti Üzerinde Sınıflandırma Çalışmaları	62
2.3.3.1.	Umdaa Veri Setinin EKK-DVM ile Sınıflandırılması	64
2.3.3.2.	Umdaa Veri Setinin ESA ile Sınıflandırılması.....	66
3.	SONUÇLAR.....	70
4.	ÖNERİLER	70
5.	KAYNAKLAR.....	71

ÖZGEÇMİŞ

Yüksek Lisans Tezi

ÖZET

DOKUNMATİK EKRANLARDA KAYDIRMA BİYOMETRİSİNE DAYALI KİMLİK
DOĞRULAMA ÇALIŞMALARI

Orhan SİVAZ

Karadeniz Teknik Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı
Danışman: Dr. Öğr. Üyesi Murat AYKUT
2021, 75 Sayfa

Bilgi güvenliğini sağlamak için günümüzde çeşitli biyometrik doğrulama sistemleri kullanılmaktadır. Bunlardan biri de son zamanlarda popüler olan dokunmatik ekran biyometrilerinden kaydırma biyometrisidir. Kaydırma biyometrisi diğer fizyolojik biyometrilere farklı olarak ek bir donanım gerektirmez ve kullanıcıyı dokunmatik ekranlar üzerinde yaptığı kaydırma hareketlerinden arka planda sürekli olarak tanımayı amaçlar.

Bu çalışmada, kaydırma hareketlerindeki karakteristik farklılıklarını ön plana çıkartacak yaklaşımların geliştirilmesi hedeflenmiştir. İlk olarak, ham kaydırma verilerinden elde edilen istatistiksel özelliklerin artırılmasının performans üzerindeki etkisi araştırılmıştır. Daha sonra, ÇYKİ, SR-ÇDA, ÇTBA ve ÇDA gibi en güncel özellik çıkartma ve EKK-DVM, GKM, ÇKP ve AK-EYK gibi yaygın kullanılan sınıflandırma yöntemlerinin kaydırma biyometrisinin doğrulama çalışmaları üzerindeki etkisi incelenmiştir. Ayrıca, skor seviyesinde sınıflandırma yöntemlerinin birleştirilmesinin; tek bir kaydırma hareketi yerine çoklu kaydırmaya dayalı doğrulamanın ve ham verilerden farklı tekniklerle görüntü oluşturarak ESA ile görüntü bazlı doğrulamanın etkileri incelenmiştir. Performans analizleri için literatürde yaygın kullanılan farklı karakteristiklere sahip 3 farklı veri setinden (Serwadda, JSS18 ve Umdaa) yararlanılmıştır.

Anahtar Kelimeler: Aktif Kimlik Doğrulama, Kaydırma Biyometriği, Özellik Çıkartma

Master Thesis

SUMMARY

AUTHENTICATION STUDIES BASED ON SWIPE BIOMETRICS ON
TOUCHSCREENS

Orhan SİVAZ

Karadeniz Technical University
The Graduate School of Natural and Applied Sciences
Computer Engineering Graduate Program
Supervisor: Asst. Prof. Murat AYKUT
2021, 75 Pages

Various biometric verification systems are used today to provide information security. One of them is swipe biometrics, one of the recently popular touchscreen biometrics. Swipe biometrics, unlike other physiological biometrics, does not require any additional hardware and aims to continuously recognize the user from the swipe movements user makes on touch screens in the background.

In this study, it is aimed to develop approaches that will highlight the characteristic differences in swipe movements. First, the effect of increasing the statistical properties obtained from the raw swipe data on the performance was investigated. Afterwards, the effect of the most up-to-date feature extraction methods such as KLPP, SR-KDA, KDA, and PCA and commonly used classification methods such as LS-SVM, GMM, MLP and WK-NN on swipe biometrics validation studies were examined. In addition, the effects of combining classification methods at the score level, verification based on multiple swiping instead of a single swipe movement and image based verification with ESA by creating images from raw data with different techniques were examined. For performance analysis, 3 different data sets (Serwadda, JSS18 and Umdaa) with different characteristics which are widely used in the literature were used.

Key Words: Active Authentication, Swipe Biometrics, Feature Extraction

ŞEKİLLER DİZİNİ

	<u>Sayfa No</u>
Şekil 1.1. Şifre (a) ve desen (b) temelli doğrulama sistemleri	3
Şekil 1.2. İki farklı kullanıcıya ait kaydırma hareketi [1]	6
Şekil 1.3. (a) Düzgün ve (b) düzgün olmayan kaydırma hareketlerine örnekler.....	13
Şekil 1.4. Örnek bir kaydırma hareketinden elde edilen bazı özellikler [23].....	15
Şekil 1.5. Veri kümesi üzerinde TBA ve DDA'nın ana izdüşümleri [28]	17
Şekil 1.6. (a) TBA ve (b) DDA izdüşüm yöntemleriyle MNIST veri setindeki örneklerin izdüşümü.....	19
Şekil 1.7. (a) Üç boyutlu veri seti, (b) TBA izdüşümü ve (c) YKİ izdüşümü (c)	20
Şekil 1.8. TBA (a), Danışmansız YKİ (b) ve Danışmanlı YKİ izdüşümü (c) yöntemleriyle MNIST veri setindeki örneklerin izdüşümü	22
Şekil 1.9. Doğrusal olarak ayıramayan verilerin ayrıştırılabilir uzaya taşınması [33]	24
Şekil 1.10. TBA (a) ve ÇTBA (b) izdüşüm yöntemleriyle MNIST veri setindeki örneklerin izdüşümü.....	26
Şekil 1.11. (a) FDA ve (b) ÇFA izdüşüm yöntemleriyle MNIST veri setindeki örneklerin izdüşümü.....	28
Şekil 1.12. ÇYKİ (a) ve YKİ (b) izdüşüm yöntemleriyle MNIST veri setindeki örneklerin izdüşümü.....	31
Şekil 1.13. Maksimum marjinle ayrılmış iki sınıflı bir veri seti örneği	31
Şekil 1.14. DVM yönteminde C parametresine göre marjinin değişiminin bir örnekle görselleştirilmesi, (a) $C = 0.1$, (b) $C = 10$	33
Şekil 1.15. Doğrusal olarak ayıramayan sınıflar için DVM yöntemiyle sınıflandırmaya bir örnek	34
Şekil 1.16. Gauss Karışım Modelinin Örnek Verileri Üzerinde Görselleştirilmesi [41]	37
Şekil 1.17. K-EYK ile örnek bir test verisinin sınıflandırılması [42].....	39
Şekil 1.18. Basit bir yapay sinir ağının yapısı [48]	40
Şekil 1.19. Girdi, ara ve çıktı katmanlarından oluşan bir ağ yapısı [48]	41
Şekil 1.20. Örnek Matris Üzerinde Maksimum Havuzlama İşleminin Uygulanması [50] .	44
Şekil 1.21. VGG16 Modeli Yapısı [51].....	45
Şekil 2.1. Serwadda veri setinde uygulanan işlemler [1]	47
Şekil 2.2. Kaydırma hareketlerinin görüntüye aktarılmasına örnekler.....	64
Şekil 2.3. Umdaa üzerinde ESA Modelinin devir sayısı artışına göre sağladığı eğitim ve doğrulama veri seti üzerindeki doğruluk oranları.....	68

TABLOLAR DİZİNİ

Sayfa No

Tablo 1.1. Örnek Sayısına Göre Kullanıcı Sayısı.....	11
Tablo 1.2. Serwadda veri setinde örnek bir kaydırma hareketi verisi	11
Tablo 1.3. Umdaa Veri Setinde Örnek Sayısına Göre Kullanıcı Sayısı	12
Tablo 2.1. Serwadda veri setinde farklı sayıda örneklerin skor seviyesinde birleştirilmesinin EER değerine etkisi	49
Tablo 2.2. Serwadda veri setinde eklenen özelliklerin EER değerine etkisi.....	50
Tablo 2.3. Umdaa veri setinde eklenen özelliklerin EER değerine etkisi	51
Tablo 2.4. Özellik Çıkartma Algoritmalarının Serwadda Veri Setine Uygulanmasından Sonra Yeni Uzay Boyutları	51
Tablo 2.5. Serwadda veri setinde özellik vektörlerinin tek bir sınıflandırıcıyla sınıflandırılma sonuçları	54
Tablo 2.6. Serwadda veri setinde EKK-DVM ve DVM sınıflandırmasından elde edilen EER değerlerinin karşılaştırılması.....	55
Tablo 2.7. Doğrusal özellik çıkartma algoritmalarıyla ve EKK-DVM ile elde edilen EER sonuçları	55
Tablo 2.8. Çekirdek özellik çıkartma ve EKK-DVM ile elde edilen EER sonuçları	56
Tablo 2.9. Serwadda veri setinde farklı parametrelerle EKK-DVM ve GKM füzyonu gerçekleştirilmesinin EER üzerindeki etkisi	57
Tablo 2.10. YSA modellerinin JSS18 veri seti üzerindeki EER sonuçları	59
Tablo 2.11. İki sınıflı yapıyla JSS18 veri seti üzerinde elde edilen EER sonuçları	60
Tablo 2.12. Özellik çıkartma algoritmalarının JSS18 veri setindeki performans sonuçları	60
Tablo 2.13. İki sınıflı yapıyla JSS18 veri seti üzerinde elde edilen EER sonuçları	61
Tablo 2.14. Umdaa veri seti üzerinde kullanım yönü sayısı farklılığında EKK-DVM yöntemiyle elde edilen EER değerleri	65
Tablo 2.15. Umdaa veri seti üzerinde farklı sınıflandırma yöntemleri ile elde edilen EER değerleri.....	65
Tablo 2.16. Farklı özellik çıkartma yöntemlerinin Umdaa veri setindeki EER sonuçları...	66
Tablo 2.17. Umdaa veri seti üzerinde 4 farklı ESA ile elde edilen EER değerleri	67
Tablo 2.18. Umdaa veri seti üzerinde ESA ile elde edilen EER değerleri	67

SEMBOLLER DİZİNİ

AK-EYK	: Ağırlıklı K-En Yakın Komşuluk
CMIM	: Mutual Information Maximization
ÇDA	: Çekirdek Diskriminant Analizi
ÇKP	: Çok Katmanlı Perseptron
ÇTBA	: Çekirdek Temel Bileşen Analizi
ÇYKİ	: Çekirdek Yerellik Koruyucu İzdüşüm
DDA	: Doğrusal Diskriminant Analizi
DVM	: Destek Vektör Makineleri
EER	: Eşit Hata Oranı
EKK-DVM	: En Küçük Kareler – Destek Vektör Makineleri
EM	: Expectation Maximization
ESA	: Evrişimsel Sinir Ağları
FAR	: False Acceptance Rate
FIS	: Fuzzy Interface System
FRR	: False Reject Rate
GE	: Gaussian Estimation
GKM	: Gauss Karışım Modeli
IVDD	: Import Vector Domain Description
LR	: Lojistik Regresyon
MCNN	: Meta Cognitive Neural Network
MVP	: Medians Vector Proximity
PSO-RBFN	: Particle Swarm Optimization – based Radial Basis Function Network
RF	: Random Forest
SFFS	: Sequential Forward Feature Selection
SR-ÇDA	: Spektral Regresyon – Çekirdek Diskriminant Analizi
SR-DDA	: Spektral Regresyon – Doğrusal Diskriminant Analizi
TBA	: Temel Bileşen Analizi
TRF	: Temporal Regression Forest
YKİ	: Yerellik Koruyucu İzdüşüm
YSA	: Yapay Sinir Ağları

1. GENEL BİLGİLER

1.1. Giriş

Dokunmatik ekranlarda güvenlik çeşitli sistemlerle sağlanmaktadır. Bunlardan biri de son zamanlarda popüler olmaya başlayan dokunmatik ekran biyometrisidir. Dokunmatik ekran biyometrilерinden biri olan kaydırma biyometrisi kullanıcının dokunmatik ekran üzerinde yaptığı kaydırma hareketlerinin karakteristiğini analiz ederek kullanıcıları doğrulayan bir davranışsal biyometridir. Kaydırma biyometrisi diğer biyometrilерden farklı olarak ek bir donanım gerektirmez ve kullanıcının günlük yaptığı kaydırma hareketlerinden kullanıcıyı arka planda sürekli olarak tanımayı amaçlar.

Diğer biyometri sistemleri kullanıcıyı oturumun başında bir kez doğrularken, kaydırma biyometrisi ise kullanıcıyı arka planda yaptığı kaydırma hareketlerinden sürekli doğrulamaktadır. Böylece oturumun başından sonuna kadar geçen sürede cihazın güvenliği kontrol edilmektedir. Diğer biyometrilерde kullanıcı oturumun başında kullanılan biyometriye göre parmak izi tanıtmak veya yüzünü göstermek gibi bir mecburiyet duymaktadır. Kaydırma biyometrisinde oturumun başında veri toplanırken kullanıcıdan spesifik bir görev yapması genelde istenmez, test verisi kullanıcının cihazla normal etkileşiminden elde edilmektedir.

Her kullanıcı dokunmatik ekranı kullanırken farklı karakteristik ortaya koymaktadır. Karakteristiklerin farklı olmasından faydalanarak veri setinden hız ve ivme gibi çeşitli özellikler çıkarılıp sınıflandırma yapılmaktadır. Kaydırma hareketleri sınıflandırma yapılırken dokunmatik ekranın konumlandırma biçimine (yatay/dikey) ve genellikle kaydırma hareketlerinin yönlerine göre ayrı ayrı değerlendirilmektedir.

Bir kaydırma hareketi tipi için gürültü olarak nitelendirilen bir özellik diğer bir kaydırma hareketi tipi için başarıyı önemli derecede yükselten bir özellik olabilir. Bu nedenle gürültü olarak nitelendirilen kısmı ortadan kaldırmak için, kullanılan sınıflandırma algoritmasına bağlı olmak şartıyla sınıflandırmadan önce çeşitli özellik çıkartma algoritmaları kullanılabilir.

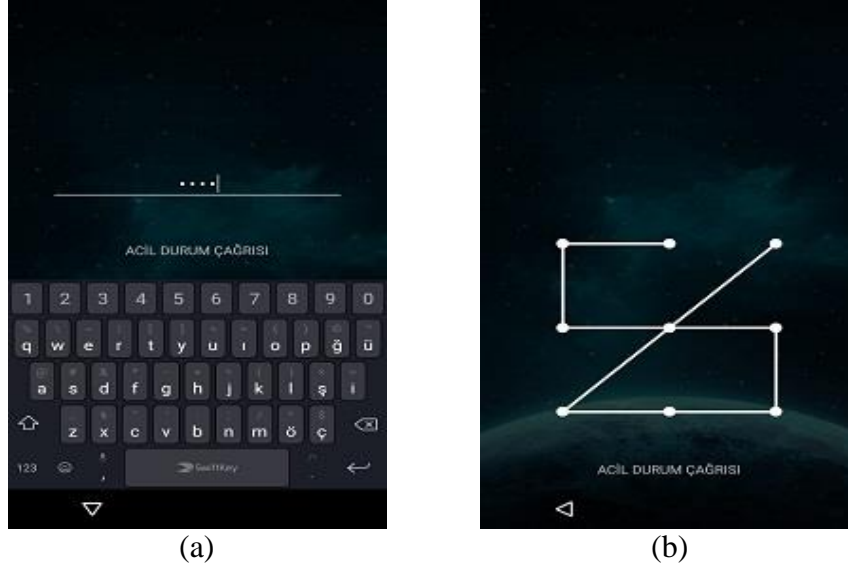
Bu tez çalışmasında kaydırma hareketlerine dayalı doğrulama sistemlerinde performans artışı sağlayacak yaklaşımlar üzerinde durulmuştur. Performans analizleri genel kullanıma açık, farklı karakteristikteki üç farklı uluslararası veri seti üzerinde

gerçekleştirilmiştir. İlk olarak kaydırma hareketlerinden çıkarılan özelliklerin sayısı artırılmış ve sınıflandırmadan önce hem doğrusal hem de doğrusal olmayan yapıya kavuşturulan biçimleriyle özellik çıkartma algoritmaları uygulanmış ve performans katkıları araştırılmıştır. Sınıflandırıcı olarak EKK-DVM, AK-EYK ve ÇKP gibi güncel yöntemler kullanılmış ve performansları karşılaştırılmıştır. Son olarak eğitimde kullanılacak örnek sayısı artırılarak performans ölçülmüş ve çeşitli tekniklerle görüntüye aktarılan kaydırma hareketleri ESA ile de sınıflandırılmıştır. Deneyler kodlanırken programlama dili olarak MATLAB ve Python kullanılmıştır.

1.2. Dokunmatik Ekranlarda Kimlik Doğrulama Sistemleri

Günümüzde dokunmatik ekranlar bir çok günlük aktivitede kullanılır. Sadece arama, mesaj alma ve gönderme değil ayrıca e-bankacılık, sosyal ağlar ve e-ticaret uygulamaları da bu cihazlar üzerinden kullanılmaktadır. Bu sebeple kullanıcılar yüksek güvenli kimlik doğrulama sistemlerine ihtiyaç duymaktadır. Dokunmatik ekranlarda kullanıcının kimliğini doğrulama işlemi kolay, hızlı ve kullanıcı dostu olmalıdır. Geleneksel kimlik doğrulama sistemleri şifre ya da desen bazlı olup, her iki yöntemin doğrulama arayüzleri Şekil 1.1’de gösterilmiştir. Bu iki yöntemin ortak eksikliklerinden biri kullanıcının gözlemlenmesiyle şifre ve desenlerin kolayca elde edilmesidir. Ayrıca kullanıcılar hatırlanması zor olacağı için genellikle daha kısa şifre ve daha basit desen seçme eğilimindedirler. Bu da cihazın güvenliğini tehlikeye atmaktadır.

Kimlik doğrulama için kullanılacak yöntem kullanıcının bireysel özelliklerini kullanan bir sistem olmalıdır. Böylece her cihaz kendi kullanıcısının özellikleriyle bağdaştırılarak cihazın güvenliği sağlanmalıdır. Bu sebeple, biyometrik özelliklerin benzersiz olmasından faydalanılarak dokunmatik ekranlarda kimlik doğrulaması için biyometri tabanlı yöntemlere yönelik araştırmalar yapılmaktadır. Bu sistemler kullanıcının fizyolojik ya da davranışsal özelliklerinden gerçek ya da sahte kullanıcı olup olmadığını tespit etmektedir. Genel olarak biyometrik kimlik doğrulama sistemleri fizyolojik ve davranışsal olmak üzere iki gruba ayrılır.



Şekil 1.1. Şifre (a) ve desen (b) temelli doğrulama sistemleri

1.2.1. Fizyolojik Biyometri

Fizyolojik biyometri sistemleri parmak izi tanıma, yüz tanıma, iris tanıma, retina tanıma ve el/avuç izi tanıma gibi alt başlıklardan oluşmaktadır.

Parmak iziyle kimlik doğrulama yöntemi akıllı telefonlarda en çok kullanılan yöntemdir. Parmak izi tabanlı yöntemlerin kullanılması kolay ve ucuz olduğundan dolayı tercih edilmektedir. Ama kesik, kir, aşınma ve yıpranma gibi nedenlerden başarı negatif yönde etkilenebilmektedir. Ayrıca yüksek kaliteli parmak izi görüntüleri elde etmek maliyetli bir iştir.

Yüz tanıma, yüz özelliklerini kullanarak bir resimden veya videodan kullanıcıyı tanıma işlemidir. Ayrıca endüstri ve akademide çok dikkat çeken popüler bir biyometrik tekniktir. Yüz tanıma sistemleri düşük ışık, uzun saç, kısmen yüzü kaplayan diğer nesneler ve düşük çözünürlüklü görüntüler altında düzgün çalışmayabilir.

İris gözbebeğinin büyüklüğünü ayarlayan elastik ve pigmentli bir bağ dokusudur. Gözden göze ve kişiden kişiye benzersiz bir desene sahiptir. İris tanımanın temel amacı kullanıcının iris yapısının matematiksel analiziyle kullanıcının kimliğini belirlemektir. Yüksek başarılı sonuçlar vermesine rağmen iris tanımanın zayıflıkları yüksek maliyet ve büyük zaman gereksinimidir.

İnsan retinası gözün arka kısmında bulunan ve sinir hücrelerinden oluşan ince bir dokudur. Genellikle retina iris ile karıştırılır. İris gözün ön kısmında bulunurken, retina

arka kısmında bulunur. Retina çok sayıda duyuşal doku katmanından ve görevi ışık ışınlarını elektriksel uyarılara dönüştürmek olan milyonlarca fotoreseptörden oluşur. Retinadaki kan damarı modeli sayesinde tanıma işlemi yapılabilir. Tıpkı iriste olduğu gibi retina tanımada da yüksek başarılı sonuçlar alınmasına rağmen yöntemin zayıflıkları yüksek maliyet ve büyük zaman gereksinimidir.

Bir insanın eli belirli bir yaştan sonra değişim göstermez ama insan eli benzersiz değildir. Avuç izi ise tek yumurta ikizlerinde bile farklılık gösterir. Avuç izi kimlik doğrulama için çok sayıda ayırt edici özellik içerir. El ve avuç izleri büyüklüğü sebebiyle akıllı telefonlara yaygın olarak kullanılmamaktadır.

Ayrıca fizyolojik biyometride avuç damar ve parmak eklemi gibi biyometrik sistemler de bulunmaktadır.

Fizyolojik biyometri sistemleri, özellikle iris tanımada yüksek başarılı sonuçlar vermesine rağmen iki büyük dezavantaja sahiptir. Bunlardan biri fizyolojik özellikleri taramak ya da tanımak için ek donanım gerektirmesidir ve bu da maliyeti önemli derecede arttırmaktadır. Bir diğeri ise sadece oturumun başında kimlik doğrulamanın yapılması ve oturum sonlanana kadar cihazın güvenliğinin tehlikeye atılmasıdır. Bu sorunlar da ek bir donanıma ihtiyaç duymayan ve belli aralıklarla kimlik doğrulamanın yapıldığı davranışsal biyometriye olan ilgiyi arttırmaktadır.

1.2.2. Davranışsal Biyometri

Davranışsal biyometri sistemleri ses tanıma, imza tanıma, yürüyüş modeli tanıma, tuş vuruşu dinamikleri ve dokunmatik hareketlerden oluşmaktadır.

Ses biyometrisi ile insan sesi karakterize edilerek konuşan kişi tanınmaya çalışır. Sesli kimlik doğrulama yönteminin kullanımı kolaydır ve kullanıcılar tarafından yaygın olarak kabul edilir. Ek olarak hem uygulama maliyeti hem de depolama alanı küçüktür. Ancak en büyük dezavantajı hastalanma gibi durumlarda insan sesinin değişmesinden dolayı yanlış sonuçlar vermesidir.

İmza biyometrisinde kullanıcıların akıllı telefonlar üzerinde imzalarına göre tanıma yapılır. Bu biyometrik teknik, kullanıcının imzalama davranışını tamamen taklit etmek zor olduğundan sahte kullanıcılara karşı oldukça dayanıklıdır. Bu yöntem, kullanıcıların her zaman aynı şekilde imzalaması söz konusu olmadığı için yüksek hata oranına sahiptir.

Yürüyüş tanıma, insanların yalnızca yürüme biçimlerinin analizini içeren bir biyometrik teknolojidir. Çoğu akıllı telefonlarda ivme ölçer olduğundan uygulanması kolaydır. Ancak arazi, ayakkabı çeşidi, yaralanma ve yorgunluk benzeri durumlar sistemin doğruluğunu etkilemektedir.

Tuş vuruşu ile kimlik doğrulama yöntemi, bir klavye veya tuş takımına karakter yazarken bireyin tarzını ve ritmini kullanarak kimlik doğrulama yapmaya çalışır. Bu sistemin avantajı özel bir donanıma ihtiyaç duyulmaması iken dezavantajı ise kullanıcı kararlılığının zamana göre değişmesi yani kullanıcı karakteristiğinin tutarlı olmamasıdır.

Dokunmatik ekran biyometrisinde ise kullanıcının dokunmatik ekran üzerinde yaptığı hareketlerden (tıklama, yakınlaştırma, uzaklaştırma, kaydırma) tanıma işlemi gerçekleştirilir. Bu biyometrik, dokunmatik ekranların yaygınlaşmasıyla popüler olmuş bir biyometri çeşitidir. Ek bir donanıma ihtiyaç duyulmaz. Tanıma işlemi arka planda yapıldığından kullanıcının kimlik doğrulama işleminden haberi olmayabilir. Bu biyometrinin en büyük dezavantajları uygulamadan uygulamaya yapılan hareketlerin değişmesi, ekran boyutuna göre farklı performans elde edilmesi, zaman içinde kullanıcı kararlılığının önemli derecede değişmesi ve her zaman aynı parmağın kullanılmaması durumunda kullanıcıyı tanıyamamasıdır. Ayrıca kullanıcının duygu durumuna göre de kararlılığı değişebilmektedir.

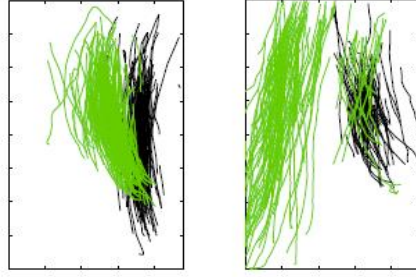
1.3. Dokunmatik Ekranlarda Kaydırma Biyometrisi

Dokunmatik ekran tabanlı akıllı cihazlar kullanıcı arayüzüyle etkileşim için girdi olarak dokunmatik tıklamalar ve dokunmatik hareketler kullanır. Dokunmatik ekran biyometrisinde ek bir donanıma ihtiyaç duyulmaz. Veri, kullanıcının dokunmatik ekranla olan normal ilişkisinden elde edilir. Veriyi elde etmek için herhangi bir spesifik görevin yapılmasına gerek yoktur. Her kullanıcı dokunmatik ekranla etkileşime geçtiğinde farklı davranışlar gösterdiği için her kullanıcı için farklı desenler oluşacaktır. Bu desenler her kullanıcı için ayırt edici özelliklere sahip olduğundan ve sınıflar arası varyans sunduğundan dolayı kullanıcıların bu desenler yardımıyla ayırt edilmesini sağlar.

Bu çalışmada sadece kullanıcının yaptığı kaydırma hareketleri ele alınmıştır. Kullanıcının ekran üzerine yaptığı kaydırma hareketleri analiz edilerek, oturumun başında bir kez istenen şifre ve desenlerin aksine sürekli kimlik doğrulaması yapılmaktadır. Yani

kullanıcının önceden kaydedilmiş kaydırma hareketleri profiline dayanılarak oturum boyunca yapılan kaydırma hareketleri aktif olarak doğrulanmaktadır.

Şekil 1.2’de iki farklı kullanıcıya ait iki farklı günde kaydedilmiş örnek kaydırma hareketleri gösterilmiştir. Farklı renkler farklı günlerde kaydedilmiş kaydırma hareketlerini göstermektedir. Kaydırma hareketleri incelendiğinde yüksek sınıf içi çeşitlilik göze çarpmaktadır. Örneğin, kullanıcı tarafından ekran üzerinde kullanılan alan değişmektedir. Sonuç olarak kullanıcının yaptığı kaydırma hareketlerinin kararlılığı zamana göre değişmektedir. Kullanıcılar arası kıyaslama yapıldığında ise kaydırma hareketlerinin farklı eğim ve uzunluğa sahip oldukları gözükmemektedir. Böylece yüksek sınıflar arası değişkenlik elde edilmektedir.



Şekil 1.2. İki farklı kullanıcıya ait kaydırma hareketi [1]

Kullanıcıların yaptığı kaydırma hareketlerinden özellik çıkartma aşamasından önce, kaydırma hareketinin hangi yöne yapıldığı bulunur. Bazı çalışmalarda [1] kaydırma hareketleri başlangıç ve bitiş noktalarına göre 4 gruba (yukarı, aşağı, sol ve sağ) ayrılırken, bazı çalışmalarda [2] ise kaydırma hareketinin başlangıç ve bitiş noktaları arasındaki doğrunun x eksenine yaptığı açıya göre bu sayı 3’e (dikey, yatay ve çapraz) düşmektedir.

Açıya göre kaydırma tipi belirlenirken x eksenine yapılan açı 30 derecenin küçük ve 150 dereceden büyük ise kaydırma hareketi yatay olarak, 30-60 ve 120-150 derece aralıklarındaysa çapraz ve son olarak 60-120 derece aralığında ise dikey olarak kabul edilmektedir.

1.4. Literatür Taraması

Dokunmatik ekranlarda biyometri ile sürekli kimlik doğrulama son zamanlarda popüler araştırma alanlarından biri haline gelmiştir. Yapılan çalışmalar incelenirken

kullanılan veri setinin kullanıcı sayısı, veri setinden çıkarılan özelliklerin sayısı, kullanılan yöntemler ve elde edilen sonuçlar dikkate alınmıştır. Bu kısımda ilk olarak dokunma biyometrisi ile ilgili çalışmalardan bahsedilmiş sonraki aşamada ise sadece kaydırma biyometrisi ile ilgili çalışmalar özetlenmiştir.

Wu vd. [3] kaydırma hareketlerinden 40, dokunma hareketlerinden ise 15 özellik çıkararak toplamda 55 özellikli ve 10 kullanıcı bir veritabanı oluşturmuşlardır. Her kullanıcıdan 15 örnek rastgele seçilmiş ve örneklerin %40'ı eğitim, %10'u parametreleri ayarlamak, %50'si de test için kullanılmıştır. Destek Vektör Makineleri (DVM) ile %96.8 doğruluk oranı elde edilmiştir.

Alariki vd. [4] dokunma hareketlerinden 6 özellik çıkarıp, 18 kullanıcı bir veritabanı oluşturmuş ve Random Forest (RF) ile sınıflandırarak %98.14 doğruluk oranı elde etmişlerdir.

Teh vd. [5] 150 kullanıcıdan PIN girilme anında veri toplamışlardır. Veri seti, 50, 100 ve 150 kullanıcı 3 kısıma ayrılarak kullanıcı sayısının performansa katkısı incelenmiştir. Her kullanıcıdan 4 ve 16 karakter uzunluğundaki PIN değerini 10 kere girmeleri istenmiş ve 10 örneğin 7'si eğitim için kullanılırken geri kalan 3 tanesi ise test için kullanılmıştır. Toplamda 6 özellik çıkarılmış ve 150 kullanıcı veri seti kullanıldığında Gaussian Estimation (GE) ile 4 karakter uzunluğunda PIN için %8.55, 16 karakter uzunluğundaki PIN için ise %5.49 EER elde edilmiştir.

Ali vd. [6] 4 farklı dokunma hareketiyle (dokunma, çift dokunma, uzun dokunma ve kaydırma hareketi) 6 kullanıcıdan veri toplamış, 6 özellik çıkarılmış ve her kullanıcının 4 parmağı farklı bir birey olarak düşünülerek $6 \times 4 = 24$ kullanıcı veri seti oluşturulmuş ve DVM ile sınıflandırma yapılarak ortalama %98.42 başarı elde edilmiştir.

Soni vd. [7] 10 kullanıcıdan 4 farklı yere 20 kere dokunmaları istenerek 6 özellikli veri seti oluşturmuş ve veri seti çeşitli sınıflandırma algoritmaları ile sınıflandırarak, Yapay Sinir Ağları (YSA) ile %89.37 doğruluk oranı elde etmişlerdir.

Alghamdi vd. [8] dokunma (tapping (text tapping and num tapping)), kaydırma (scrolling), sürüklemeye (dragging) ve yakınlaştırma (zooming) hareketlerinin verilerini gelişmiş bir android uygulaması kullanarak 5 oturumda elde etmiş ve 20 kullanıcı veri setinden toplamda 42 özellik çıkarmış. 5 oturumdan 3'ü eğitim için kullanılırken 2'si test için kullanılmıştır. %5.25 EER hata oranının yakalandığı en iyi sonuç Medians Vector Proximity (MVP) yöntemi ile elde edilmiştir.

Zou vd. [9], 40 kullanıcıdan 5 farklı cihaz üzerinde veri toplamış ve bu verilerden 32 özellik çıkarmıştır. Sonraki aşamada ise Mutual Information Maximization (CMIM) yöntemi ile sınıflandırma performansını en çok arttıran 5 özellik sınıflandırıcıya verilmiş ve örneklerin yarısı eğitim için kullanılırken diğer yarısı ise test için kullanılmıştır. Sınıflandırma için tek sınıflı Import Vector Domain Description (IVDD) yöntemi kullanılmış ve %3.56 FAR ve %2.14 FRR değerleri elde edilmiştir.

Alpar vd. [10], kullanıcıların PIN girerken yaptığı dokunma hareketlerini incelemiş ve şifreyi bilme durumunda bile farklı lokasyon tıklamaları gerçekleştireceğini savunmuşlardır. Kullanıcılardan Orcan90* şifresini girmeleri istenmiş ve 64 gerçek ve 64 sahte şifre girişi analiz edilmiştir. Kullanıcının aynı noktaya tıklamasının imkansız olmasından dolayı esnek bir sistem olan Fuzzy Interface System (FIS) kullanmış ve 1.61 EER değeri elde edilmiştir.

Meng vd. [11], 48 kullanıcıdan 24'ünün bir cihazı istediği gibi kullanmasını isterken diğer 24'nün ise sadece web üzerinde arama yapmasını istemiş ve 20 oturumda dokunma verileri toplamışlardır. Elde edilen verilerden 21 özellik çıkarılmış, 20 oturumun 12'si eğitim için geri kalanı ise test için kullanılmıştır. Çeşitli sınıflandırma algoritmaları kullanılmış olup en iyi sonuç Particle Swarm Optimization-based Radial Basis Function Network (PSO-RBFN) ile elde edilmiştir. Cihazı istediği gibi kullananlar için %3.67 FAR ve %4.13 FRR elde edilirken, cihazı sadece web üzerinde arama için kullanan kullanıcılar için %2.22 FAR ve %2.54 FRR elde edilmiştir.

Meng vd. [12], diğer bir çalışmalarında 60 kullanıcıdan 5 gün boyunca 30 oturumda dokunma verilerini toplamış ve bu verilerden 9 özellik çıkarmışlardır. Çeşitli sınıflandırma yöntemleri kullanılmış ve en iyi sonuç DVM ile elde edilmiştir. 30 oturumun 20'si eğitim için kullanılırken geri kalan 10 oturum ise test için kullanılmıştır. Sonuç olarak DVM ile 4.66 EER elde edilmiştir.

Yang vd. [2], 45 kullanıcıdan 2 hafta boyunca dokunma hareketleri toplamış ve dokunma hareketlerini 4 gruba (dokunma, dikey kaydırma, yatay kaydırma ve çapraz kaydırma) ayırmışlardır. Dokunma için 4, yatay ve dikey kaydırma için 10'ar ve son olarak çapraz kaydırma hareketleri için ise 16 özellik çıkarılmıştır. Sınıflandırıcı olarak tek sınıflı DVM ve Isolation Forest kullanılmış ve ortalama %95.85 doğruluk elde edilmiştir.

Bundan sonraki kısımda sadece kaydırma biyometrisi ile ilgili çalışmalardan bahsedilecektir.

Frank vd. [13], 41 kullanıcıli veri setinden 30 özellik çıkartmış, bu özelliklerin sadece 27 tanesi kullanılmış ve K-En Yakın Komşuluk (K-EYK) ve DVM ile sınıflandırma yapmıştır. Dikey ve yatay kaydırma hareketleri ayrı ayrı sınıflandırılmıştır. Oturum içi senaryoda %2-%3 arası, oturumlar arası senaryoda ise %0-%4 arasında EER elde etmiştir.

Serwadda vd. [14], 197 kullanıcıli veri setinden 28 özellik çıkartmış ve en popüler 10 makine öğrenmesi yöntemleriyle sınıflandırma yapmıştır. Sınıflandırıcılar arasında en iyi performansı Lojistik Regresyon (LR) vermiştir. Veri seti dokunmatik ekranın hem yatay hem de dikey kullanımında elde edilen verilerden oluşmaktadır. LR ile yatay kullanımda yatay ve dikey kaydırma hareketleri için sırasıyla %10.5 ve %13.7 EER elde edilirken, bu sayılar dikey kullanımda ise %13.8 ve %17.2'dir.

SenthilPrabha vd. [15], 60 kullanıcıli veri setinden 30 özellik çıkartıp bu özellikleri Meta Cognitive Neural Network (MCNN) ile sınıflandırmıştır. %84.52-88.61 arası doğruluk elde edilmiştir.

Fierrez vd. [1], 197, 41, 71 ve 48 kullanıcıli veri setlerinden biri 28 bir diğeri 5 boyutlu iki özellik vektörü çıkarmış, 28 boyutlu özellik vektörünü En Küçük Kareler-Destek Vektör Makineleri (EKK-DVM) ile 5 boyutlu özellik vektörünü Gauss Karışım Modeli (GKM) ile sınıflandırmıştır. Kaydırma hareketleri başlangıç ve bitiş noktalarına göre 4 gruba ayrılarak ayrı ayrı değerlendirilmiştir. Daha sonra ise EKK-DVM ve GKM sınıflandırıcılarının skorlarının ortalaması alınarak nihai skorlar elde edilmiş ve performans ölçülmüştür. 197 kullanıcıli veri setinde oturumlar arası senaryoda dikey kullanımda %13.9-17.4, yatay kullanımda ise %10.7-12.2 arası EER elde edilmiştir.

Shen vd. [16], 71 kullanıcıli veri setinden 4 tane konum , 15 tane uzunluk , 12 tane açı, 1 tane zaman, 8 tane hız, 6 tane ivme, 6 tane açısal hız ve 6 tane de basınç ile ilgili toplamda 58 özellik çıkartmış ve bu özelliklere özellik seçme uygulayarak en iyi sınıflandırma performansını veren özellikleri seçmiştir. Sınıflandırma aşamasında RF, DVM, K-EYK ve YSA kullanılmıştır. Bu çalışmada diğeri çalışmalardan farklı olarak uygulamaya bağlı performans ölçümü yapılmıştır. Örnek olarak döküman okunurken RF ile %5.76-%7.21 FAR, %6.82-%7.98 FRR elde edilmiştir.

Mahbub vd. [17], 48 kullanıcıli veri setinden 4 tane konum, 2 tane zaman, 4 tane hız, 4 tane ivme, 7 tane uzunluk ve 1 tane de basınç ile ilgili 22 özellik çıkarıp popüler makine öğrenmesi algoritmalarını kullanarak sınıflandırma yapmışlardır. En iyi performansı %22.1 EER ile RF yöntemi vermiştir.

Syed vd. [18], 31 kullanıcıli veri setinden 4 tane konum, 3 tane basınç, 4 tane hız, 3 tane uzunluk, 2 tane zaman ve 1 tane açı ile ilgili 17 özellik çıkartıp RF ile sınıflandırma yapmıştır. Bu çalışmada, dokunmatik ekranın boyutuna ve ekranın farklı tutuş senaryolarına göre performans ölçülmüştür. Ekran boyutu arttıkça performansın arttığı vurgulanmıştır. Ekran boyutuna ve tutuş senaryolarına göre %3.80-8.81 arası EER'ler elde edilmiştir.

Tse vd. [19], 31 kullanıcıli veri setinden 14 özellik çıkarıp çeşitli makine öğrenmesi algoritmalarıyla sınıflandırma yapmıştır. En iyi performansı %66.63 doğruluk ile Çok Katmanlı Perseptron (ÇKP) vermiştir.

Ooi vd. [20], Serwadda ve Frank veri setleri üzerinde skor birleştirme yönteminin aksine özellik birleştirme yaparak Serwadda ve Frank veri setleri için sırasıyla ortalama %4 ve %2.5 EER elde etmişleridir. Yöntem olarak Temporal Regression Forest (TRF) kullanılmıştır.

1.5. Veri Setleri

1.5.1. Serwadda Veri Seti

Serwadda veri seti [14] 197 kullanıcının yaptığı kaydırma hareketlerinden oluşmaktadır. Kaydırma hareketleri toplanırken tek bir cihaz kullanılmıştır. Kullanıcıya resim ve text bazlı birden çok seçenekli sorular sorularak gözden geçirme ve okuma yapması istenmiştir. Ayrıca kullanıcıya dokunmatik ekranla serbest etkileşim kurma izni verilmiştir. Her kullanıcı en az bir gün arayla olmak üzere 2 oturumda veri girişi sağlamıştır. Dokunmatik ekranı 138 kullanıcı dikey olarak kullanmışken, yatay olarak ise 59 kullanıcı kullanmıştır. Dikey kullanımda sınıflandırılmaya dahil olan toplam kaydırma hareketi sayısı 84,420 iken bu sayı yatay kullanımda 35,490'dır. Tablo 1.1'de eğitim aşamasında kullanılacak minimum örnek sayısına göre eğitim verisindeki kullanıcı sayısı verilmiştir. Bazı kullanıcılara ait örnek sayısı az olduğundan tabloda belirtilen minimum eğitim örneği sayısını sağlayan kişi sayısı da değişiklik göstermektedir.

Tablo 1.1. Örnek Sayısına Göre Kullanıcı Sayısı

Yön / Örnek Sayısı	Dikey			Yatay		
	40	60	80	40	60	80
Yukarı	124	74	36	54	44	25
Aşağı	132	124	99	55	53	51
Sol	104	73	38	39	22	9
Sağ	118	97	67	45	36	22

Veri toplama için kullanılan dokunmatik ekran üzerinde kaydırma hareketi yapıldığında, arka planda x ve y kordinatları, ekrana yapılan ve normalize edilmiş basınç bilgisi, zaman damgası (timestamp), kaydırma hareketinin yapıldığı parmak ile ekran arasında kalan alan ve akıllı telefonun hangi şekilde kullanıldığı (dikey, yatay) bilgisi kaydedilmektedir. Tablo 1.2’de ilk kullanıcının örnek bir kaydırma hareketi verisi gösterilmektedir. Kaydırma hareketi 7 nokta kaydı içermektedir.

Tablo 1.2. Serwadda veri setinde örnek bir kaydırma hareketi verisi

Kullanıcı No	Kaydırma No	X	Y	Basınç	Kaplanan Alan	Olay Zamanı
1	1	366.56	500.00	0.75	0.26	5294067
1	1	374.53	457.81	0.75	0.23	5294098
1	1	382.96	420.31	0.75	0.26	5294126
1	1	385.78	404.68	0.75	0.23	5294141
1	1	390.93	388.28	0.75	0.20	5294156
1	1	393.28	371.09	0.75	0.20	5294171
1	1	396.56	357.81	0.75	0.20	5294185

1.5.2. JSS18 Veri Seti

JSS18 veri seti [21], [22] 31 kullanıcının ekran boyutları farklı 4 akıllı cihaz üstünde ve her cihazın 3 farklı pozisyonuna göre 8 oturumda toplanan kaydırma hareketlerinden oluşmaktadır. İlk pozisyonda kullanıcı dokunmatik ekranı masanın üzerinde ve dikey olarak kullanırken, ikinci pozisyonda kullanıcı akıllı cihazı sağ elde tutup dokunmatik ekranı dikey olarak kullanmaktadır. Üçüncü pozisyonda ise kullanıcı akıllı cihazı yine sağ elde tutarken, dokunmatik ekranı yatay olarak kullanmaktadır. Her kullanıcı, her cihaz ve her pozisyon için veri girişi yapmıştır. Kaydırma hareketleri bir resim eşleştirme oyunu ile elde edilmiştir. Kullanıcı ekranda gördüğü resmi galeride bulmak için yatay ve dikey

kaydırmalar yapmıştır. Her kullanıcıdan eşit örnek sayısı alındığında kullanıcı başına örnek sayısı 13,080 olmaktadır. Yatay kaydırmaların sayısı dikey kaydırmaların sayısının 5 katıdır. Toplamda 4 cihaz ve 3 farklı pozisyon olduğu için 12 farklı durum oluşmaktadır. Her durum için kullanıcı başına ortalama 1,090 kaydırma hareketi bulunurken, bunların 908 tanesi yatay, 182 tanesi ise dikey kaydırma hareketleridir.

1.5.3. Umdaa Veri Seti

Umdaa veri seti [17] 48 kullanıcıdan iki ay içerisinde toplanan kaydırma hareketlerinden oluşmaktadır. Serwadda veri setinde olduğu gibi kaydırma hareketleri tek bir dokunmatik ekran üzerinde elde edilmiştir. Veriler elde edilirken kullanıcıya kaydırma hareketi yapması için spesifik bir görev ya da soru verilmemiş, dokunmatik ekranla serbest etkileşim kurmasına izin verilmiştir. Diğer veri setlerine göre ortalama kullanıcı başı kaydırma hareketi sayısı oldukça fazladır.

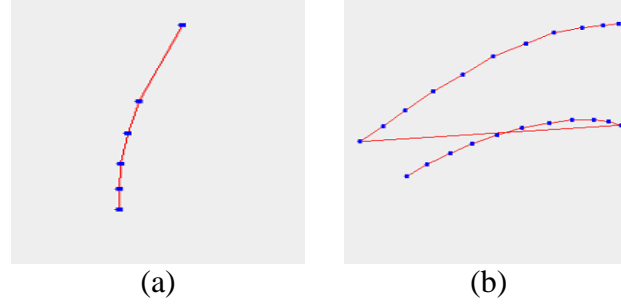
Tablo 1.3.'de eğitimde kullanılacak minimum örnek sayısına göre eğitimdeki kullanıcı sayısı verilmiştir. Yatay kullanımda kullanıcı sayısı az olduğundan dolayı sadece dikey kullanım değerlendirilmiştir. Dikey kullanımdaki toplam kaydırma hareketi sayısı 106,060'dır. Toplam kaydırma hareketi sayısının yarısından fazlasını aşağıya doğru yapılan kaydırma hareketleri oluşturmaktadır (54,620).

Tablo 1.3. Umdaa Veri Setinde Örnek Sayısına Göre Kullanıcı Sayısı

Yön / Örnek Sayısı	40	60	80
Yukarı	28	26	20
Aşağı	31	30	30
Sol	21	20	14
Sağ	26	25	19

1.6. Ön İşlemler

Bir kaydırma hareketinin değerlendirilmeye alınması için bazı ön işlemlerden geçmesi gerekmektedir. İlk olarak bir kaydırma hareketi en az 4 nokta kaydı içermelidir [1]. Bir sonraki aşamada ise kaydırma hareketinin düzgün olması beklenir. Şekil 1.3'te düzgün ve düzgün olmayan örnek kaydırma hareketleri gösterilmektedir.



Şekil 1.3. (a) Düzgün ve (b) düzgün olmayan kaydırma hareketlerine örnekler

Şekil 1.3'teki resimlerde referans noktalar kaydırma hareketinin kaydedildiği anları ifade etmektedir. Bir kaydırma hareketinin düzgün olup olmadığının kontrolü şöyle yapılmaktadır: ilk olarak başlangıç noktasından her referans noktaya uzaklıklar bulunur. Bu uzaklıklar sürekli artarak gidiyorsa, kaydırma hareketi düzgün bir kaydırma hareketidir [2]. Eğer uzaklıklar artarak gitmiyorsa kaydırma hareketi düzgün bir kaydırma hareketi değildir ve aykırı örnek olduğu için değerlendirmeye alınmaz. Uzaklıklar küçük bir çıkıntı dahil artarak gitmiyorsa örnek aykırı örnek sınıfına dahil edilmektedir.

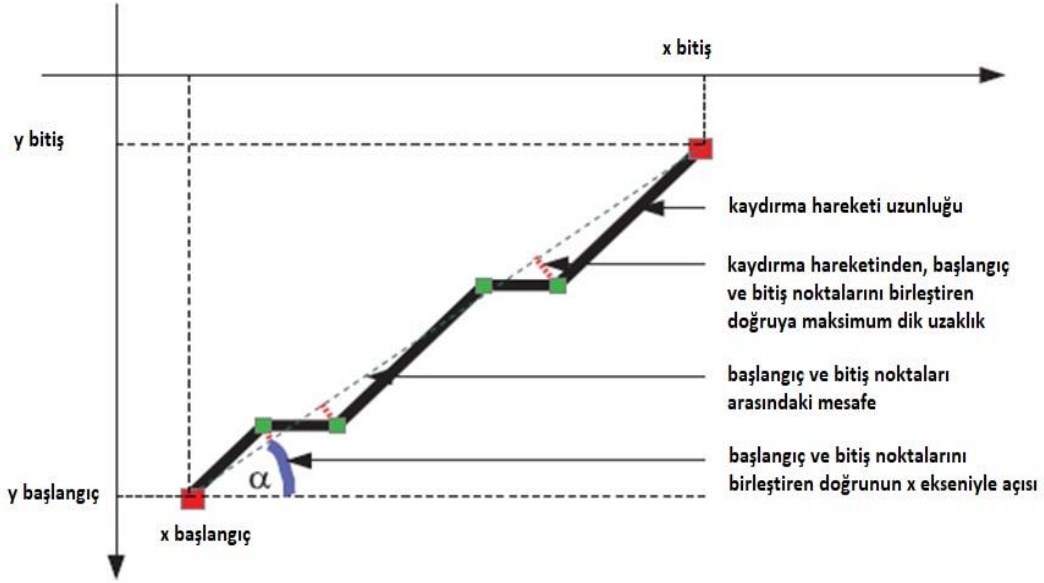
1.7. Formüle Dayalı Özellik Çıkartma

Tablo 1.2'de belirtilen kaydırma hareketi verilerinden aşağıdaki özellikler çıkarılmıştır.

- (1-5) Kaydırma hareketinin $1 \times n$ boyutlarındaki hız dizisinin ortalaması, standart sapması, birinci, ikinci ve üçüncü çeyrek bilgisi [14], [18]. Birinci çeyrek bilgisi $1 \times n$ boyutlarındaki hız dizisinin %25'lik kısmına denk gelen skalar değeri ifade etmektedir. İkinci çeyrek bilgisi %50'ye denk gelen, üçüncü çeyrek bilgisi de %75'e denk gelen skalar değeri ifade etmektedir.
- (6-10) Kaydırma hareketinin $1 \times n$ boyutlarındaki ivme dizisinin ortalaması, standart sapması, birinci, ikinci ve üçüncü çeyrek bilgisi [14].
- (11-15) Kaydırma hareketinin basınç dizisinin ortalaması, standart sapması, birinci, ikinci ve üçüncü çeyrek bilgisi [14].

- (16-20) Kaydırma hareketinin dokunmatik ekran üzerinde 1xn boyutlarındaki kaplanan alan dizisinin ortalaması, standart sapması, birinci, ikinci ve üçüncü çeyrek bilgisi [14].
- (21-24) Kaydırma hareketin yatay olması durumunda en sol ve sağ noktanın kordinatları, dikey olması durumunda ise en aşağı ve üst noktaların koordinatları (min X, max X, min Y, max Y) [14].
- (25) Başlangıç bitiş noktaları arasındaki en kısa mesafe [14], [18].
- (26) Başlangıç ve bitiş noktalarını birleştiren doğrunun x ekseniiyle oluşturduğu açı [14], [18].
- (27) Kaydırma hareketinin toplam süresi [14], [18].
- (28) Her ardışık nokta çiftleri arasındaki mesafelerin toplamı (Kaydırma hareketinin uzunluğu) [14], [18].
- (29) Kaydırma hareketinin uzunluğunun başlangıç ve bitiş noktaları arasındaki mesafeye oranı [18].
- (30-32) Kaydırma hareketinden, kaydırma hareketinin başlangıç ve bitiş noktaları arasındaki doğruya olan maksimum dik uzaklık ve bu maksimum uzaklığa sahip doğrunun kaydırma hareketi üzerindeki koordinatları [23].
- (33-37) Kaydırma hareketinden başlangıç ve bitiş noktaları arasındaki doğruya olan dik uzaklık dizisinin ortalaması, standart sapması, birinci, ikinci ve üçüncü çeyrek bilgisi [23].
- (38-39) Kaydırma hareketinin başlangıç ve bitiş noktalarında ekrana yapılan basınç [18].
- (40) Ardışık iki kaydırma hareketi arasındaki süre [18].
- (41) Kaydırma hareketinin orta noktasındaki basınç [18].
- (42-45) Kaydırma hareketinin başlangıç ve bitiş noktaları [18].

Şekil 1.4.'te örnek bir kaydırma hareketi üzerinden çıkarılan bazı özellikler gösterilmiştir.



Şekil 1.4. Örnek bir kaydırma hareketinden elde edilen bazı özellikler [23]

1.8. Öğrenmeye Dayalı Özellik Çıkartma

Herhangi bir konuda oluşturulan veri setleri için çıkarılan özelliklerin bazıları sınıflandırmaya negatif katkıda bulunabilmektedir. Bu negatif etkiyi ortadan kaldırmak için kullanılacak sınıflandırıcının yapısına bağlı olmak şartıyla özellik seçme ya da çıkartma işlemleri yapılabilir. Özellik çıkartma algoritmaları belirli bir amacın (verilerin saçılımını maksimize etmek, sınıf ayırt edilebilirliğini arttırmak gibi) optimizasyonuna yönelik olarak verileri genellikle daha düşük boyutlu yeni uzayaya eşleyen bir dönüşüm sağlar. Böylece sınıflandırma performansı artırılabilir.

1.8.1. Doğrusal Özellik Çıkartma Yöntemleri

1.8.1.1. Temel Bileşen Analizi

Temel Bileşen Analizi (TBA), giriş uzayının boyutunu indirgemeyi amaçlayan ve örüntü tanımda yaygın olarak kullanılan bir algoritmadır [24]. TBA danışmansız bir yöntemdir. TBA'da ilk olarak eğitim verisiyle daha küçük boyutlu uzay bulunur ve ardından test verisi bu uzay üzerine izdüşürülür.

İzdüşüm uzayı eğitim verisiyle elde edileceğinden eğitim verisi, bütün veriyi iyi temsil etmelidir. TBA’da ilk olarak eğitim verisinin ortalaması eşitlik (1)’deki gibi bulunur. Daha sonra örneklerin ortalaması orjine yerleştirilecek şekilde eşitlik (2)’deki gibi öteleme işlemi gerçekleştirilir. Öteleme işlemi eğitim verisindeki her örnekten ortalamanın çıkarılmasıyla bulunur.

$$\boldsymbol{\mu} = [\mu_1, \mu_2, \dots, \mu_p], \quad \mu_k = \frac{1}{n} \sum_{i=1}^n x_{ik}, \quad k=1,2,\dots,p \quad (1)$$

$$\bar{\mathbf{x}}_i = \mathbf{x}_i - \boldsymbol{\mu}, \quad i = 1,2,\dots,n \quad (2)$$

Burada \mathbf{x}_i , eğitim kümesindeki örnekleri, n örnek sayısını, p giriş uzayı boyutunu, $\boldsymbol{\mu}$ ise örneklerin ortalamasını ifade etmektedir.

Bir sonraki adımda veriler arasındaki saçılımı ifade eden kovaryans matrisi eşitlik (3) kullanılarak bulunur.

$$\mathbf{S}_t = \sum_{i=1}^n \bar{\mathbf{x}}_i^T \bar{\mathbf{x}}_i \quad (3)$$

Kovaryans matrisi bulunduktan sonra problem eşitlik (4)’teki gibi genel özdeğer özvektör problemine dönüştürülür.

$$\mathbf{S}_t \mathbf{e} = \lambda \mathbf{e} \quad (4)$$

Yukarıdaki denklemde λ özdeğerleri ifade ederken, \mathbf{e} aynı özdeğerlere karşılık gelen özvektörleri ifade etmektedir. Özdeğer ve özvektörler bulunduktan sonra, özdeğerler büyükten küçüğe doğru sıralanır. Sıralanan özdeğerlerin bazıları gürültüyü temsil edebileceğinden dolayı, özdeğerlerin toplamının %95’ini en az sayıda özdeğerlerin toplamıyla sağlayan özdeğerlere karşılık gelen özvektörler seçilir. Bu aşama eşitlik (5)’te verilmiştir.

$$\sum_{i=1}^m |\lambda_i| = 0.95 * \sum_{i=1}^n |\lambda_i| \quad (5)$$

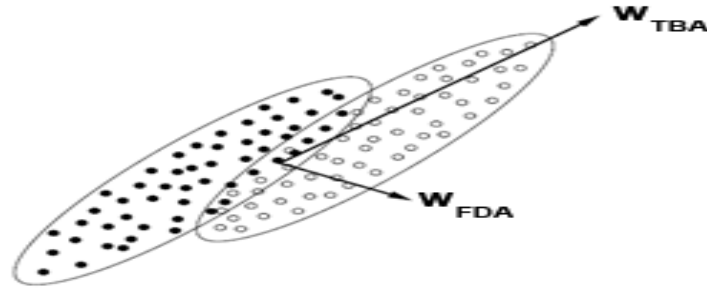
Yukarıdaki denklemde m toplam özdeğer sayısını ifade ederken, n ise 0.95 şartını sağlayan özdeğerlerin sayısını ifade etmektedir.

Son olarak yeni bir test örneğini eğitim aşamasında elde edilen uzaya izdüşürmek için eşitlik (6)'daki gibi, test örneğinden ortalama çıkarılır ve özvektörlerden oluşan matris ile çarpılır [25].

$$\mathbf{y}_{\text{yeni uzay}} = (\mathbf{y} - \mu)\mathbf{e} \quad (6)$$

1.8.1.2. Doğrusal Diskriminant Analizi

TBA, veriyi daha küçük boyutlu uzaya izdüşürürken her zaman en uygun izdüşümü bulamaz [26]. Bu durum özellikle az sayıda öznitelik kullanıldığında daha dikkat çekmektedir (Şekil 1.5.). Bunun nedeni TBA'nın danışmansız olmasında yatar. Etiket bilgisinin de eğitime katılması gerektiği düşünülerek Doğrusal Diskriminant Analizi (DDA) ortaya atılmıştır [27]. Bu yöntemin amacı eşitlik (7)'deki amaç fonksiyonunu maksimize etmektir.



Şekil 1.5. Veri kümesi üzerinde TBA ve DDA'nın ana izdüşümleri [28]

$$J(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{S}_B \mathbf{w}}{\mathbf{w}^T \mathbf{S}_w \mathbf{w}} \quad (7)$$

Yukarıdaki denklemde \mathbf{S}_B sınıflar arası dağılımı matrisini ifade ederken, \mathbf{S}_w de sınıf içi dağılımı ifade etmektedir. Aşağıdaki denklemlerde \mathbf{S}_B ve \mathbf{S}_w 'nin nasıl hesaplandığı ifade edilmektedir.

$$\mathbf{S}_B = \sum_{k=1}^c l_i (\mu_k - \mu) (\mu_k - \mu)^T \quad (8)$$

$$\mathbf{S}_w = \sum_{k=1}^c \sum_{x \in \mathcal{C}_k} (\mathbf{x} - \mu_k)(\mathbf{x} - \mu_k)^T, \quad \mu_k = \frac{1}{l_k} \sum_{x \in \mathcal{C}_k} \mathbf{x}, \quad \mu = \frac{1}{l} \sum_{k=1}^c l_k \mu_k \quad (9)$$

Yukarıdaki denklemlerde l_i i sınıfına ait örnek sayısını ifade ederken, l tüm örnek sayısını ifade etmektedir. DDA'da genel mantık sınıf içi varyansı minimize edip sınıflar arası varyansı maksimize etmektir. Son olarak eşitlik (7)'deki amaç fonksiyonu birtakım optimizasyon işlemlerinden sonra genelleştirilmiş özdeğer özvektör problemine eşitlik (10)'daki gibi dönüştürülür [25] .

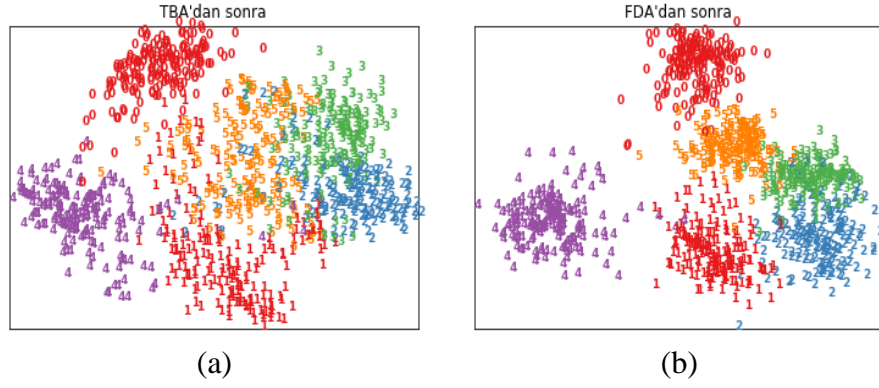
$$\mathbf{S}_B \mathbf{w} = \lambda \mathbf{S}_w \mathbf{w} \quad (10)$$

Bazı çalışmalarda \mathbf{S}_w yerine optimizasyon amacıyla \mathbf{S}_T ifadesi de kullanılmaktadır. Aşağıdaki ifadede \mathbf{S}_T 'nin formülü ve yöntemin son hali bulunmaktadır. Bu çalışmada deneyler koşullurken sadece \mathbf{S}_T ifadesinin olduğu denklem kullanılmıştır. \mathbf{W} , her sınıf için $1/l_j$ (l_j j. sınıftaki örnek sayısını ifade etmektedir.) katsayılarını içermektedir. içermektedir. Eğer örnekler aynı sınıfta değilse aralarında ilişkisi olmadığı düşünülerek \mathbf{W} matrisinde o değer sıfıra çekilir.

$$\mathbf{S}_B \mathbf{w} = \lambda \mathbf{S}_T \mathbf{w}, \quad \mathbf{S}_T = \mathbf{X} \mathbf{X}^T, \quad \mathbf{X} \mathbf{W} \mathbf{X} \mathbf{w} = \lambda \mathbf{X} \mathbf{X}^T \mathbf{w} \quad (11)$$

Aşağıdaki şekilde sınıf bilgisini kullanmayan ve danışmansız bir yöntem olan TBA ile sınıf bilgisini kullanan ve danışmanlı bir yöntem olan DDA'nın 8x8 boyutlarında rakam görüntülerinden oluşan MNIST [29] veri setine uygulanması sonucu elde edilen 2 boyutlu yeni uzaydaki dağılımları gösterilmektedir.

Veri seti toplam on sınıftan oluşup, sıfırdan dokuz kadar olan sayılar içermektedir. Boyut indirgeme algoritmasından sonra ayırımın daha detaylı gözükmesi için ilk altı rakamın (0, 1, 2, 3, 4, 5) veri seti kullanılmıştır. Görselliğin sağlanması amacıyla, özdeğerler ve özvektörler bulunduktan sonra hem TBA hem de DDA için en büyük iki özdeğere karşılık gelen özvektörler alınarak dönüşüm işlemi yapılmıştır.



Şekil 1.6. (a) TBA ve (b) DDA izdüşüm yöntemleriyle MNIST veri setindeki örneklerin izdüşümü

DDA, TBA'dan farklı olarak danışmanlı bir yöntemdir. TBA'nın danışmansız olması her zaman uygun izdüşümü bulmasına engel olmaktadır. Yukarıdaki şekilde de görüldüğü gibi DDA'da TBA'ya göre sınıf içi örnekler birbirlerine daha yakın durmaktadırlar. Bunun nedeni DDA'nın sınıf içi varyansı azaltıp sınıflar arası varyansı arttırmasıdır.

1.8.1.3. Spektral Regresyon - Doğrusal Diskriminant Analizi

DDA yöntemi örnek sayısı çok olduğunda hesaplama açısından uzun zaman alabilmekte ve aşırı bellek kullanımına sebep olmaktadır. Bu sebeple, alternatif bir yöntem olan Spektral Regresyon - Doğrusal Diskriminant Analizi (SR-DDA) [30] yöntemi geliştirilmiştir. Hatırlanacağı üzere DDA yöntemi aşağıdaki problemi çözmeye çalışır.

$$\mathbf{XW}\mathbf{X}^T\boldsymbol{\alpha} = \lambda\mathbf{X}\mathbf{X}^T\boldsymbol{\alpha} \quad (12)$$

SR-DDA yöntemi ise şu teoremi kullanarak problemi çözmektedir. $\mathbf{W}\mathbf{y} = \lambda\mathbf{y}$ özdeğer özvektör probleminde \mathbf{y} özvektör, λ ise özdeğer olsun. Eğer $\mathbf{X}^T\boldsymbol{\alpha} = \mathbf{y}$ ise eşitlik (12)'deki aynı özdeğer ve özvektör elde edilir. Aşağıdaki denklemde teoremin ispatı yapılmaktadır.

$$\mathbf{XW}\mathbf{X}^T\boldsymbol{\alpha} = \mathbf{XW}\mathbf{y} = \mathbf{X}\lambda\mathbf{y} = \lambda\mathbf{X}\mathbf{y} = \lambda\mathbf{X}\mathbf{X}^T\boldsymbol{\alpha} \quad (13)$$

Böylece eşitlik (12)'deki denklemi çözmek yerine $\mathbf{W}\mathbf{y} = \lambda\mathbf{y}$ ifadesi çözülerek daha kısa zamanda sonuç elde edilir. Yukarıda bahsedilen ifadeyi çözmek için ilk olarak aşağıdaki ifade minimize edilmeye çalışılır.

$$\arg \min_{\alpha} ((\mathbf{X}^T \alpha - \mathbf{y})^T (\mathbf{X}^T \alpha - \mathbf{y}) + \beta \alpha^T \alpha) \quad (14)$$

Yukarıdaki ifadenin α 'ya göre türevi alınıp sıfıra eşitlenirse aşağıdaki ifade elde edilir. İfade de \mathbf{I} birim matrisi ifade ederken, β değeri ise düzenleme katsayısıdır. Son olarak eşitlik (15) çözülerek özdeğerler ve özvektörler elde edilmiş olur. SR-DDA yöntemi DDA yöntemine göre hem hız hem de bellek açısından önemli fayda sağlamaktadır.

$$(\mathbf{X}\mathbf{X}^T + \beta \mathbf{I})\alpha = \mathbf{X}\mathbf{y} \quad (15)$$

1.8.1.4. Yerellik Koruyucu İzdüşüm

Yerellik Koruyucu İzdüşüm (YKİ) yöntemi TBA yöntemine alternatif olarak ortaya konulan doğrusal bir boyut indirgeme yöntemidir [31]. TBA yönteminden farklı olarak, boyutu azaltılacak olan verinin komşuluk ilişkisini koruyacak şekilde izdüşüm yapan bir yöntemdir. Aşağıdaki şekilde de görüldüğü gibi 3 boyutlu ve iki sınıflı veri setine TBA ve YKİ özellik çıkartma algoritmaları uygulanarak, 2 boyutlu uzaydaki izdüşümleri gösterilmiştir. TBA verinin orjinal uzaydaki geometrik yapısını korumadan izdüşüm yaparken, YKİ yöntemi ise izdüşümü verinin geometrik yapısını koruyarak yapmaktadır.



Şekil 1.7. (a) Üç boyutlu veri seti, (b) TBA izdüşümü ve (c) YKİ izdüşümü (c)

YKİ ve TBA arasındaki bir diğer fark aykırı değerlerin izdüşüm üzerindeki etkisidir. TBA varyans tabanlı bir yöntem olduğu için aykırı değerlerin varlığından güçlü şekilde etkilenmektedir. YKİ ise verinin komşuluk ilişkisini korumaya odaklandığı için aykırı değerlerin varlığından güçlü bir şekilde etkilenmemektedir. Yöntemin analizi yapılırken ilk olarak eşitlik (16) minimize edilmeye çalışılır.

$$\min_p \sum_{i,j=1}^n \|y_i - y_j\|^2 S(i,j), \quad S(i,j) = e^{\frac{-\|x_i - x_j\|^2}{t}} \quad (16)$$

Yukarıdaki denklemde y_i ve y_j , x_i ve x_j 'nin yeni uzaya izdüşürülmüş biçimlerini ifade ederken $S(i,j)$ ise verinin komşuluk ilişkisini ifade etmektedir. Komşuluk matrisi, YKİ yönteminin kullanım şekline göre farklılık gösterebilir. Etiket bilgisi kullanıldığında, x_i ve x_j aynı sınıfta ise $S(i,j)$ yukarıdaki formüldeki gibi hesaplanır. Eğer aynı sınıfta değilse ise $S(i,j)$ 0'a setlenir. Etiket bilgisi kullanılmadığında ise, x_i x_j 'ye en yakın k komşudan biriyse $S(i,j)$ yine formüldeki gibi bulunur. Aksi durumda ise $S(i,j)$ 0'a setlenir. Minimize edilecek denklem daha da basitleştirilerek eşitlik (17) elde edilir.

$$\begin{aligned} & \frac{1}{2} \sum_{i,j=1}^n \|w^T x_i - w^T x_j\|^2 S(i,j) \\ &= \frac{1}{2} (\sum_{i,j=1}^n w^T x_i S_{ij} x_j^T w - 2 \sum_{i,j=1}^n w^T x_i S_{ij} x_j^T w + \sum_{i,j=1}^n w^T x_j S_{ij} x_i^T w) \\ &= \sum_{i,j=1}^n w^T x_i S_{ij} x_i^T w - \sum_{i,j=1}^n w^T x_i S_{ij} x_j^T w \\ &= \sum_i w^T x_i D_{ii} x_i^T w - w^T X S X^T w \\ &= w^T X D X^T w - w^T X S X^T w \\ &= w^T X (D - S) X^T w \end{aligned} \quad (17)$$

Eşitlik (17)'deki D matrisi diagonal bir matristir ve $D_{ii} = \sum_j S(i,j)$ eşitliğiyle hesaplanır.

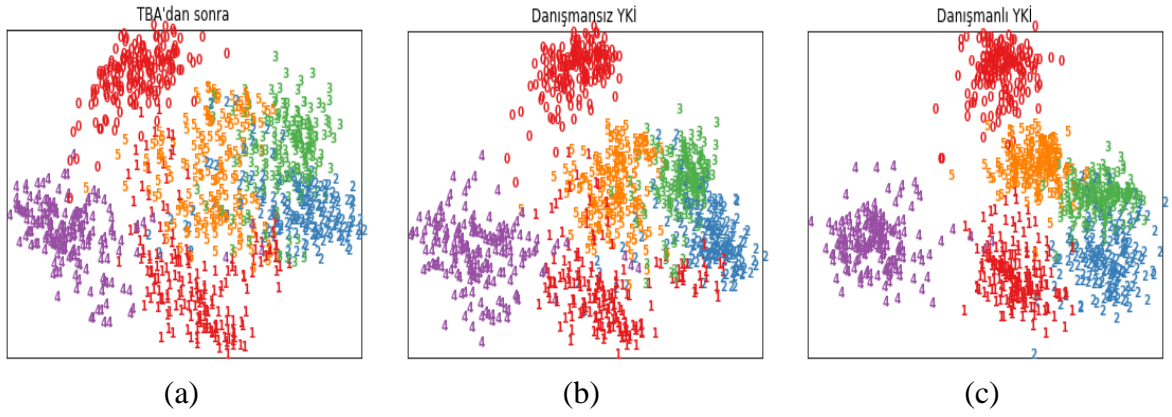
Minimizasyon problemi $w^T X D X^T w = 1$ kısıt koşulu altında eşitlik (18)'deki gibi genelleştirilmiş özdeğer özvektör problemine dönüştürülür. En düşük özdeğerlere karşılık gelen özvektörler aranan sonuçlardır.

$$\mathbf{X}(\mathbf{D} - \mathbf{S})\mathbf{X}^T\mathbf{w} = \lambda\mathbf{XDX}^T\mathbf{w} \quad (18)$$

Eşitlik (18)'de, en küçük özdeğerlere karşılık düşen özvektörler, eşitlik (19)'daki en büyük özdeğerlere karşılık düşen özvektörlerle aynıdır.

$$\mathbf{XSX}^T\mathbf{w} = \lambda\mathbf{XDX}^T\mathbf{w} \quad (19)$$

YKİ yöntemi komşuluk matrisinin hesaplanma türüne göre danışmanlı ya da danışmansız olabilir. Aşağıdaki şekillerde MNIST [29] veri setine TBA, danışmansız ve danışmanlı YKİ yöntemleri uygulanarak iki boyutlu uzaydaki izdüşümleri gösterilmektedir.



Şekil 1.8. TBA (a), Danışmansız YKİ (b) ve Danışmanlı YKİ izdüşümü (c) yöntemleriyle MNIST veri setindeki örneklerin izdüşümü

Yukarıda da görüldüğü gibi YKİ yöntemi danışmansız olarak da kullanıldığında TBA'dan daha iyi sonuç vermektedir. Bunun nedeni YKİ'nin verinin komşuluk ilişkisini korumaya çalışmasından kaynaklanmaktadır.

YKİ yöntemi danışmansız olarak kullanıldığında komşuluk matrisi her veriye en yakın k tane örnek için denklem eşitlik (16)'daki gibi hesaplanır. En yakın k örnek içinde olmayanlar ise sıfıra setlenir. k parametresinin seçimi deneysel olarak belirlenmektedir. YKİ yöntemi danışmanlı olarak kullanıldığında ise sınıf içi geometrik yapı korunmaya çalışılır. Danışmanlı yöntemde bir örnek sadece aynı sınıftaki örneklerle eşitlik (16)'dan geçirilir. Sonuç olarak orjinal uzayda bir örnek aynı sınıftaki bir örnekle yakınsa yeni uzayda da yakın olması, farklı bir sınıftaki örneklerle ise uzak olması amaçlanır.

1.8.2. Doğrusal Olmayan Özellik Çıkartma Yöntemleri

1.8.2.1. Çekirdek Kavramı

Çekirdek kavramı, Aizerman vd. [32] tarafından ortaya atılmıştır. Çekirdek kavramı doğrusal öğrenme yöntemlerini yüksek boyutlu öznitelik uzayına taşıyarak alternatif bir çözüm sunar. Orjinal uzayda ayrılamayan sınıflar çeşitli çekirdek fonksiyonlarıyla daha yüksek boyutlu uzaya taşınarak ayrıştırılabilir hale getirilir.

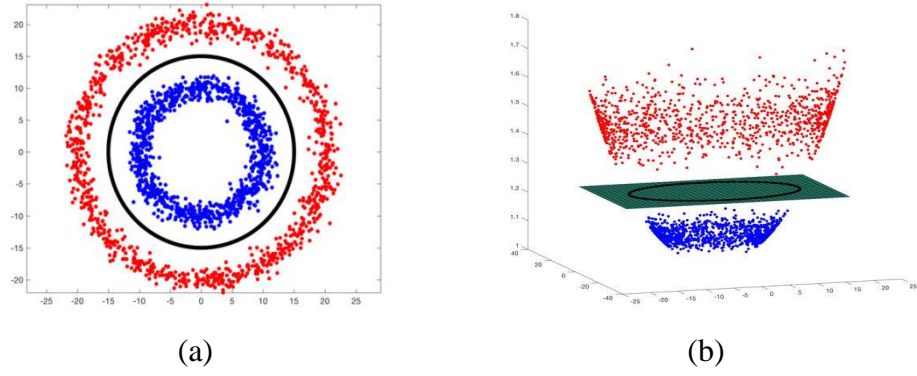
$$\mathbf{x} = (x_1, \dots, x_n) \rightarrow \phi(\mathbf{x}) = (\phi_1(\mathbf{x}), \dots, \phi_n(\mathbf{x})) \quad (20)$$

Çekirdek yöntemlerde, izdüşüm uzayına doğrudan açıkça bir izdüşüm işlemi yapılmamakta, izdüşürülecek uzaydaki işlemlerin sonucu giriş uzayı ve çekirdek fonksiyonları yardımıyla elde edilmektedir. Örneğin, giriş verilerimiz iki boyutlu uzayda tanımlansın: $x = (x_1, x_2)$. Kullanılan çekirdek fonksiyonu ise $k(\mathbf{x}, \mathbf{x}') = (1 + \mathbf{x}^T \mathbf{x}')^2$ şeklinde bir ikinci dereceden fonksiyon olsun. Eğer yeni uzaydaki işlemler örneklerin iç çarpımı şeklinde ifade edilebiliyorsa çekirdek fonksiyonu yardımıyla o uzaydaki sonuç giriş verilerinden elde edilebilir:

$$k(\mathbf{x}, \mathbf{x}') = (1 + x_1 x'_1 + x_2 x'_2)^2 = 1 + 2x_1 x'_1 + 2x_2 x'_2 + 2x_1 x'_1 x_2 x'_2 + x_1^2 x'^2_1 + x_2^2 x'^2_2 \quad \text{Bu}$$

sonuç, $\phi(x) = (1, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_1 x_2, x_1^2, x_2^2)$ dönüşümüyle elde edilecek yeni uzaydaki iç çarpımla aynıdır.

Şekil 1.9'da görüldüğü gibi doğrusal olarak ayrılamayan 2 boyutlu veri ϕ dönüşümü ile yeni uzayda hiper düzlemle ayrılabilir hale getirilebilir. Şekilde sınıfların giriş uzayında doğrusal olarak ayrılamayacağı ve karar sınırının daire olduğu gözükmemektedir. Giriş verileri doğrusal olmayan $\phi(x, y) = (z_1, z_2, z_3) = (x, y, e^{-\gamma(x^2+y^2)})$ eşleşmesi ile yeni uzaya taşındığında karar sınırı hiper düzlem halini alır.



Şekil 1.9. Doğrusal olarak ayrılamayan verilerin ayrıştırılabilir uzaya taşınması [33]

Literatürde çeşitli çekirdek fonksiyonları bulunmaktadır. En yaygın kullanılan çekirdekler polinomsal ve Gaussian çekirdekleridir. Gaussian çekirdeği eşitlik (21)'de verildiği şekilde tanımlanmaktadır.

$$k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma^2}\right) \quad (21)$$

Örnek sayısının çok fazla olması durumunda çekirdek fonksiyonun hesaplanması zaman alabilmekte ve güçlü bilgisayarlar gerektirmektedir. Bu gibi problemlerin üstesinden gelebilmek için her kullanıcının verisinden yeterli sayıda ve dengeli örnekleme yapılarak çekirdek fonksiyonunun hesaplanması kolaylaştırılmaktadır.

1.8.2.2. Çekirdek Temel Bileşen Analizi

Çekirdek Temel Bileşen Analizi (ÇTBA), Schölkopf vd. [34] tarafından geliştirilmiştir. Çekirdek kavramı ile TBA doğrusal olmayan yapıya dönüştürülmüş olur. İlk olarak veri seti yüksek boyutlu uzaya çekirdek fonksiyonuyla taşınır ve TBA'nın adımları bu yüksek boyutlu uzayda yapılır. TBA'daki özdeğer problemi aşağıdaki şekli alır.

$$\mathbf{S}_t^\phi \mathbf{e}^\phi = \lambda \mathbf{e}^\phi \quad (22)$$

$$\mathbf{S}_t^\phi = \frac{1}{N} \sum_{j=1}^N \phi(\mathbf{x}_i) \phi^T(\mathbf{x}_i) \quad (23)$$

Yukarıda \mathbf{S}_t^ϕ kovaryans matrisini ifade etmektedir. \mathbf{e}^ϕ özellik vektörlerinin doğrusal kombinasyonları cinsinden yazıldığında eşitlik (24) elde edilir.

$$\mathbf{e}^\phi = \sum_{i=1}^N \alpha_i \phi(\mathbf{x}_i) \quad (24)$$

(23) ve (24) eşitlikleri, eşitlik (22)'de yerine yazılır ve her iki taraf $\phi^T(\mathbf{x}_k)$ ile çarpılırsa eşitlik (25) elde edilir.

$$\sum_{i=1}^N \sum_{j=1}^N \alpha_j k(\mathbf{x}_k, \mathbf{x}_i) k(\mathbf{x}_i, \mathbf{x}_j) = N \sum_{j=1}^N \alpha_j k(\mathbf{x}_k, \mathbf{x}_j) \quad (25)$$

Burada $k(\mathbf{x}_i, \mathbf{x}_j)$ (26)'da tanımlanan iç çarpım çekirdeğidir.

$$k(\mathbf{x}_i, \mathbf{x}_j) = \phi^T(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j) \quad (26)$$

Eşitlik (25) matris şeklinde yazılırsa daha basit bir gösterim olan eşitlik (27) elde edilir.

$$\mathbf{K}^2 \boldsymbol{\alpha} = N \lambda \mathbf{K} \boldsymbol{\alpha} \quad (27)$$

Eşitlik (27)'de $\boldsymbol{\alpha}$ özvektör matrisini, λ çekirdek matrisinin özdeğerlerini ve \mathbf{K} da çekirdek matrisini ifade etmektedir. Formülün her iki tarafı \mathbf{K} ile sadeleştirilirse eşitlik (28) elde edilir.

$$\mathbf{K} \boldsymbol{\alpha} = \lambda \boldsymbol{\alpha} \quad (28)$$

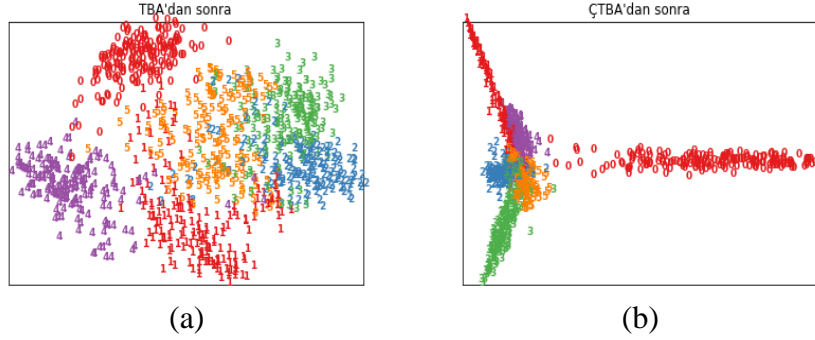
Eşitlik (28)'deki özdeğer özvektör problemi çözüldükten sonra aynı TBA'da olduğu gibi özdeğerler büyükten küçüğe sıralanır ve bütün özdeğerlerin toplamının %99'unu en az sayıda özdeğerlerin toplamıyla sağlayan özdeğerlere karşılık gelen özvektörler dönüşüm matrisi olarak belirlenir.

$$\sum_{i=1}^m |\lambda_i| = 0.99 * \sum_{i=1}^n |\lambda_i| \quad (29)$$

Son olarak eşitlik (30)'daki gibi her özvektör ona karşılık gelen özdeğerin kareköküne bölünerek normalize edilir [25].

$$\mathbf{a}_i = \mathbf{a}_i / \sqrt{\lambda_i} \quad (30)$$

Aşağıdaki şekilde MNIST [29] veri setine TBA ve Gaussian çekirdek fonksiyonuna sahip ÇTBA uygulanarak yeni izdüşümler elde edilmiştir.



Şekil 1.10. TBA (a) ve ÇTBA (b) izdüşüm yöntemleriyle MNIST veri setindeki örneklerin izdüşümü

Hem TBA hem de ÇTBA için izdüşüm yapılırken sadece en büyük iki özdeğere karşılık gelen özvektörler alınarak izdüşüm yapıldığından bazı sınıfların örnekleri iç içe geçmektedir. İki boyutlu uzayda iç içe geçen sınıflar daha fazla özdeğere karşılık gelen özvektörler hesaba katıldığında birbirinden ayrılabilir.

1.8.2.3. Çekirdek Diskriminant Analizi

Çekirdek kavramı ile DDA doğrusal olmayan yapıya dönüştürülmüş ve Çekirdek Diskriminant Analizi (ÇDA) [35] adını almıştır. Böylece giriş uzayında karmaşık ve birbirine geçmiş örneklerin ayırt edilebilmesi sağlanmıştır. İlk olarak eşitlik (31)'deki amaç fonksiyonu doğrusal olmayan ϕ eşlemesi ile F özellik uzayına taşınır.

$$J(\mathbf{v}) = \frac{\mathbf{v}^T \mathbf{S}_B^\phi \mathbf{v}}{\mathbf{v}^T \mathbf{S}_W^\phi \mathbf{v}}, \quad \mathbf{v} = \sum_{i=1}^l \alpha_i \phi(\mathbf{x}_i) \quad (31)$$

$$\mathbf{S}_B^\phi = (\mu_1^\phi - \mu_2^\phi)(\mu_1^\phi - \mu_2^\phi)^T \quad (32)$$

$$\mathbf{S}_W^\phi = \sum_{i=1}^2 \sum_{\mathbf{x} \in C_i} (\phi(\mathbf{x}) - \mu_i^\phi)(\phi(\mathbf{x}) - \mu_i^\phi)^T, \quad \mu_i^\phi = \frac{1}{l_i} \sum_{\mathbf{x} \in C_i} \phi(\mathbf{x}) \quad (33)$$

Burada $\mathbf{v} \in F$; \mathbf{S}_B^ϕ ve \mathbf{S}_W^ϕ , F özellik uzayında sınıflar arası ve sınıf içi dağılım matrisleridir. Amaç fonksiyonunun pay kısmındaki ifade açılırsa eşitlik (34) elde edilir.

$$\mathbf{v}^T \mathbf{S}_B^\phi \mathbf{v} = \mathbf{v}^T (\mu_2^\phi - \mu_1^\phi)(\mu_2^\phi - \mu_1^\phi)^T \mathbf{v} \quad (34)$$

Çekirdek kavramının kullanılmasıyla aşağıdaki eşitlik elde edilir.

$$\mathbf{v}^T \mu_i^\phi = \frac{1}{l_i} \sum_{j=1}^{l_i} \sum_{k=1}^{l_i} \alpha_j k(\mathbf{x}_j, \mathbf{x}_k^i) = \mathbf{a}^T \mathbf{M}_i, \quad (\mathbf{M}_i)_j = \frac{1}{l_i} \sum_{k=1}^{l_i} k(\mathbf{x}_j, \mathbf{x}_k^i) \quad (35)$$

İfade daha sade hale getirilirse eşitlik (36) elde edilir.

$$\mathbf{v}^T \mathbf{S}_B^\phi \mathbf{v} = \mathbf{a}^T \mathbf{M} \mathbf{a}, \quad \mathbf{M} = (\mathbf{M}_2 - \mathbf{M}_1)(\mathbf{M}_2 - \mathbf{M}_1)^T \quad (36)$$

Aynı işlemler amaç fonksiyonunun payda kısmı için yapıldığında eşitlik (37) elde edilir.

$$\mathbf{v}^T \mathbf{S}_W^\phi \mathbf{v} = \sum_{j=1,2} \mathbf{a}^T \mathbf{K}_j (\mathbf{I} - \mathbf{1}_{l_j}) \mathbf{K}_j^T \mathbf{a} \quad (37)$$

Eşitlik (37)'de \mathbf{I} birim matrisi, $\mathbf{1}_{l_j}$ bütün elamanları $1/l_j$ olan matris olmak üzere payda kısmının en yalın hali eşitlik (38)'deki gibi olur.

$$\mathbf{v}^T \mathbf{S}_W^\phi \mathbf{v} = \mathbf{a}^T \mathbf{N} \mathbf{a}, \quad \mathbf{N} = \sum_{j=1,2} \mathbf{K}_j (\mathbf{I} - \mathbf{1}_{l_j}) \mathbf{K}_j^T \quad (38)$$

Sonuç olarak amaç fonksiyonu aşağıdaki ifadeye dönüşür.

$$J(\mathbf{a}) = \frac{\mathbf{a}^T \mathbf{M} \mathbf{a}}{\mathbf{a}^T \mathbf{N} \mathbf{a}}, \quad \mathbf{a} = \mathbf{N}^{-1} \mathbf{M} \text{ nin özvektörleri} \quad (39)$$

Yaygın olarak, TBA'da %95, ÇTBA'da ise %99 kuralı ile özvektör sayısı belirlenirken, ÇDA'da özvektör sayısı sınıf sayısının bir eksiği kadardır. Yani 10 sınıflı bir veri setine ÇDA uygulanırsa, 9 tane özvektör elde edilir.

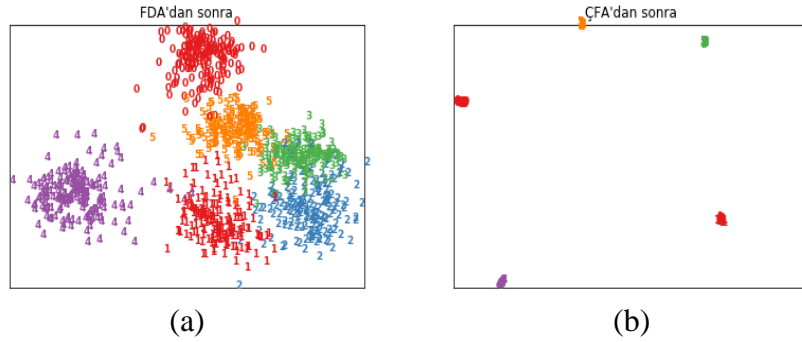
Yeni bir x örneği giriş olarak verildiğinde izdüşümü aşağıdaki eşitlikle bulunur.

$$(\mathbf{v} \cdot \phi(\mathbf{x})) = \sum_{i=1}^l \mathbf{a}_i k(\mathbf{x}_i, \mathbf{x}) \quad (40)$$

\mathbf{N} matrisinin boyut sayısı arttıkça genelde tekil olduğu için tersi alınamaz. Bu nedenle bir düzenleştirme uygulanması gerekir. En yaygın kullanılan düzenleştirme yöntemi, birim matrisin küçük bir sayıyla çarpılarak \mathbf{N} 'ye eklenmesi işlemidir [25].

$$\mathbf{N}' = \mathbf{N} + c\mathbf{I} \quad (41)$$

Aşağıdaki şekilde MNIST [29] veri setine sırasıyla DDA ve ÇDA yöntemleri uygulanıp, en büyük iki özdeğere karşılık gelen özvektörler alınarak iki boyutlu uzayda dağılımları gösterilmiştir. Şekilden de görüldüğü gibi ÇDA'da çekirdek kavramı sayesinde sınıflar sadece iki özdeğer kullanıldığında bile tamamen birbirinden ayrılmış gözükmemektedir.



Şekil 1.11. (a) FDA ve (b) ÇFA izdüşüm yöntemleriyle MNIST veri setindeki örneklerin izdüşümü

1.8.2.4. Spektral Regresyon - Çekirdek Diskriminant Analizi

DDA'da olduğu gibi ÇDA'da da örnek sayısı çok olduğunda aşırı bellek tüketimi ve zaman kaybı olmaktadır. Bu sebeple Spektral Regresyon - Çekirdek Diskriminant Analizi (SR-ÇDA) [36] yöntemi ileri sürülmüştür. Hatırlanacağı üzere DDA yöntemi aşağıdaki problemi çözmeye çalışır.

$$\mathbf{S}_B \boldsymbol{\alpha} = \lambda \mathbf{S}_T \boldsymbol{\alpha}, \quad \mathbf{K} \mathbf{W} \mathbf{K}^T \boldsymbol{\alpha} = \lambda \mathbf{K} \mathbf{K}^T \boldsymbol{\alpha} \quad (42)$$

SR-ÇDA yönteminde SR-DDA yönteminden farklı olarak kovaryans matrisi yerine çekirdek matrisi kullanılmaktadır. Eğer \mathbf{K} çekirdek matrisi tekil değilse özdeğerler aşağıdaki ifade ile elde edilir.

$$\boldsymbol{\alpha} = \mathbf{K}^{-1}\mathbf{y} \quad (43)$$

Eğer tekil ise aşağıdaki gibi düzenleştirme yapılır ve özdeğerler öyle bulunur.

$$\boldsymbol{\alpha} = (\mathbf{K} + \beta\mathbf{I})^{-1}\mathbf{y} \quad (44)$$

1.8.2.5. Çekirdek Yerellik Koruyucu İzdüşüm

YKİ doğrusal bir yöntemdir ve doğrusal olmayan ilişkileri temsil etmekte yetersizdir. Bu yüzden çekirdek kavramı kullanılarak yöntem doğrusal olmayan yapıya dönüştürülmüş ve Çekirdek Yerellik Koruyucu İzdüşüm (ÇYKİ) [37] adını almıştır.

Giriş örnekleri $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n]$ olsun. İlk olarak doğrusal olmayan ϕ fonsiyonuyla veri yüksek boyutlu F özellik uzayına haritalanır $F: \phi(\mathbf{X}) = [\phi(\mathbf{x}_1), \phi(\mathbf{x}_2), \dots, \phi(\mathbf{x}_n)]$. Daha sonra F özellik uzayında, örneklerin ağırlıklı mesafelerinin toplamını minimize ederek, $\phi(\mathbf{X})$ verilerinin sınıf içi geometrik yapısını koruyabilen \mathbf{P}_ϕ izdüşümü bulunur. İzdüşümü bulmak için ilk olarak eşitlik (45) minimize edilir.

$$\min_{\mathbf{P}_\phi} \sum_{i,j=1}^n \|\mathbf{z}_i - \mathbf{z}_j\|^2 \mathbf{W}(i,j), \quad \mathbf{z}_i = \mathbf{P}_\phi^T \phi(\mathbf{x}_i) \quad (45)$$

Bu denklemdeki $\mathbf{z}_i = \mathbf{P}_\phi^T \phi(\mathbf{x}_i)$, \mathbf{P}_ϕ üzerinde $\phi(\mathbf{x}_i)$ 'nin izdüşümünü ifade etmektedir. $\mathbf{W}(i, j)$ ise \mathbf{x}_i ve \mathbf{x}_j arasındaki komşuluk ilişkisini temsil etmektedir. Minimize edilecek denklem daha da basitleştirilerek eşitlik (46) elde edilir.

$$\sum_{i,j=1}^n \|\mathbf{P}_\phi^T \phi(\mathbf{x}_i) - \mathbf{P}_\phi^T \phi(\mathbf{x}_j)\|^2 \mathbf{W}(i,j) = \mathbf{Z} \mathbf{P}_\phi^T \phi(\mathbf{X}) (\mathbf{D} - \mathbf{W}) \phi(\mathbf{X})^T \mathbf{P}_\phi \quad (46)$$

Eşitlik (46)'daki \mathbf{D} matrisi diagonal bir matrisdir ve $\mathbf{D}_{ii} = \sum_j \mathbf{W}(i,j)$ eşitliğiyle hesaplanır.

$$\mathbf{P}_\phi = \sum_{i=1}^n \alpha_i \phi(\mathbf{x}_i) = \phi(\mathbf{X})\alpha \quad (47)$$

Eşitlik (47), eşitlik (46)'da yerine koyulursa eşitlik (48) elde edilir.

$$\begin{aligned} & 2\mathbf{P}_\phi^T \phi(\mathbf{X})(\mathbf{D} - \mathbf{W})\phi(\mathbf{X})^T \mathbf{P}_\phi \\ &= 2\alpha^T \phi(\mathbf{X})^T \phi(\mathbf{X})(\mathbf{D} - \mathbf{W})\phi(\mathbf{X})^T \phi(\mathbf{X})\alpha \\ &= 2\alpha^T \mathbf{K}(\mathbf{D} - \mathbf{W})\mathbf{K}\alpha \end{aligned} \quad (48)$$

Eşitlik (48)'de $\mathbf{K}(i, j) = \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$ denklemiyle ifade edilir. Çekirdek kavramına göre, F 'de iki vektörün noktasal çarpımı, doğrusal olmayan ϕ haritalamasını açıkça bilmeden bir çekirdek fonksiyonuyla $k(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{x}) \cdot \phi(\mathbf{y})$ hesaplanır.

Minimizasyon problemi $\alpha^T \mathbf{K} \mathbf{D} \mathbf{K} \alpha = 1$ kısıt koşulu altında eşitlik (49)'daki gibi genelleştirilmiş özdeğer problemine dönüştürülür. En düşük özdeğerlere karşılık gelen özvektörler aranan sonuçlardır.

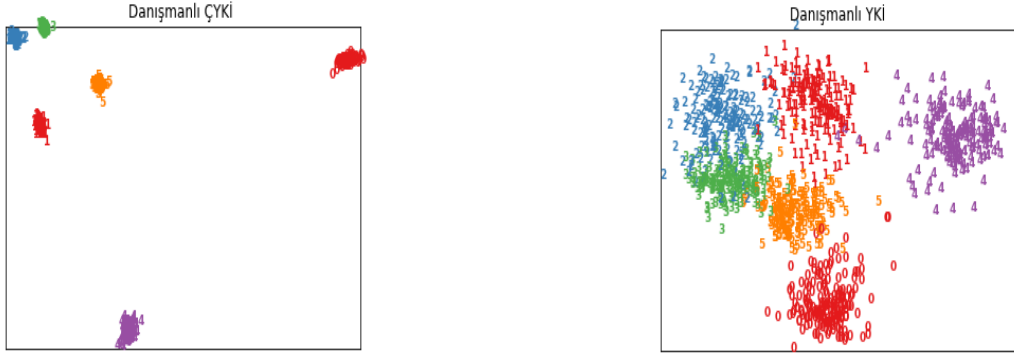
$$\mathbf{K}(\mathbf{D} - \mathbf{W})\mathbf{K}\alpha = \lambda \mathbf{K} \mathbf{D} \mathbf{K} \alpha \quad (49)$$

$$\mathbf{K} \mathbf{W} \mathbf{K} \alpha = \lambda \mathbf{K} \mathbf{D} \mathbf{K} \alpha \quad (50)$$

Eşitlik (49)'da en küçük özdeğerlere karşılık düşen özvektörler ile eşitlik (50)'deki en büyük özdeğerlere karşılık düşen özvektörler aynıdır. Bu çalışmada özvektörler bulunurken eşitlik (50) kullanılmıştır.

TBA'da %95 kuralı, ÇTBA'da %99 kuralı, DDA ve ÇDA'da sınıf sayısının bir eksiği kuralı ile çıkış uzayının boyutu hesaplanırken YKİ ve ÇYKİ için kesin bir kural yoktur. Bu sebeple çıkış uzayının boyutu bu çalışmada deneysel çalışmalarla ile belirlenmiştir.

Aşağıdaki şekilde MNIST veri setine sırasıyla YKİ ve ÇYKİ yöntemleri uygulanıp, en küçük iki özdeğere karşılık gelen özvektörler alınarak iki boyutlu uzayda dağılımları gösterilmiştir. Şekilden de görüldüğü gibi çekirdek kavramının kullanılmasının sınıfları birbirinden ayırmada başarılı olduğu gözükmektedir.



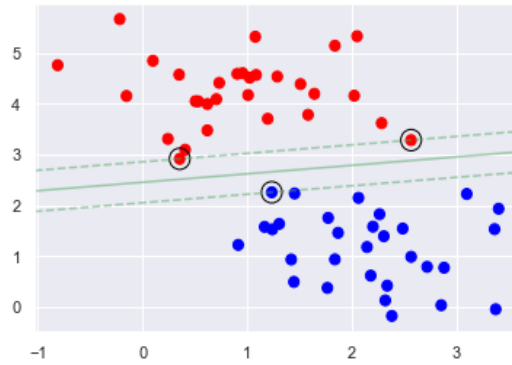
Şekil 1.12. ÇYKİ (a) ve YKİ (b) izdüşüm yöntemleriyle MNIST veri setindeki örneklerin izdüşümü

1.9. Sınıflandırma Yöntemleri

1.9.1. Destek Vektör Makineleri

DVM, Vapnik tarafından geliştirilmiş [38] bir sınıflandırma yöntemidir. Amacı iki veri kümesini birbirinden ayırabilecek birçok hiperdüzlemden maksimum marjine sahip olanı bulmaktır. Yöntemde, marjinin büyüklüğü arttıkça ayırma hiper düzlemi ile tanımlanan doğrusal sınıflandırıcının genelleştirme hatasının daha küçük olacağı varsayımı yapılmaktadır.

Şekil 1.13'te iki sınıflı bir veri setini maksimum marjinle ayıran, ayırma hiper düzlemi gösterilmektedir. Siyah daire içine alınan örnekler destek vektörlerini ifade etmektedir.



Şekil 1.13. Maksimum marjinle ayrılmış iki sınıflı bir veri seti örneği

Bütün örneklerin ayırma hiperdüzleminden belirli bir uzaklıkta olduğu çözüme ulaşmak için eşitlik (51)'nin sağlanması gerekir.

$$y_i(\mathbf{w}^T \mathbf{x}_i + w_0) \geq 1, i = 1, \dots, n \quad (51)$$

Eşitlik (51)'de y_i sınıf etiketini ifade etmektedir. Sınıf etiketleri yerine konulursa marjinin uzunluğu $2/\|\mathbf{w}\|$ olarak bulunur.

$$\begin{aligned} (1) (\mathbf{w}^T \mathbf{x}_i + w_0) - 1 &= 0, \quad \mathbf{w}^T \mathbf{x}_i = 1 - w_0 \\ (-1) (\mathbf{w}^T \mathbf{x}_i + w_0) - 1 &= 0, \quad \mathbf{w}^T \mathbf{x}_i = -1 - w_0 \\ ((1 - w_0) - (-1 - w_0)) \frac{1}{\|\mathbf{w}\|} &= \frac{2}{\|\mathbf{w}\|} \end{aligned} \quad (52)$$

Marjini geniş tutmak için eşitlik (52)'nin maksimize edilmesi gerekir. Diğer bir deyişle $\|\mathbf{w}\|$ ifadesi minimize edilmelidir. Optimizasyon problemi, Lagrange biçimlilik ile L_p amaç fonksiyonunun ilk haline dönüştürülür.

$$L_p = \frac{1}{2} \mathbf{w}^T \mathbf{w} - \sum_{i=1}^n \alpha_i (y_i (\mathbf{w}^T \mathbf{x}_i + w_0) - 1) \quad (53)$$

Birincil parametreler \mathbf{w} ve w_0 'dır. Eşitlik (53)'de $\alpha_i \{i = 1, \dots, n; \alpha_i \geq 0\}$ Lagrange çarpıcılarıdır. Amaç fonksiyonu indirgendiğinde eşitlik (54) ve (55) elde edilir.

$$\frac{\partial L_p}{\partial \mathbf{w}} = \mathbf{w} - \sum_i^n \alpha_i y_i \mathbf{x}_i = 0, \quad \mathbf{w} = \sum_i^n \alpha_i y_i \mathbf{x}_i \quad (54)$$

$$\frac{\partial L_p}{\partial w_0} = - \sum_i^n \alpha_i y_i = 0, \quad \sum_i^n \alpha_i y_i = 0 \quad (55)$$

Eşitlik (54), eşitlik (53)'te yerine yazılırsa eşitlik (56) elde edilir.

$$\begin{aligned} & \frac{1}{2} (\sum_i^n \alpha_i y_i \mathbf{x}_i) (\sum_i^n \alpha_i y_i \mathbf{x}_i) - \sum_i^n \alpha_i y_i \mathbf{w}^T \mathbf{x}_i - \sum_i^n \alpha_i y_i w_0 + \sum_i^n \alpha_i \\ & \frac{1}{2} \sum_i^n \sum_j^n \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j - \sum_i^n \sum_j^n \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j + 0 + \sum_i^n \alpha_i \\ & L_p = \sum_i^n \alpha_i - \frac{1}{2} \sum_i^n \sum_j^n \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \end{aligned} \quad (56)$$

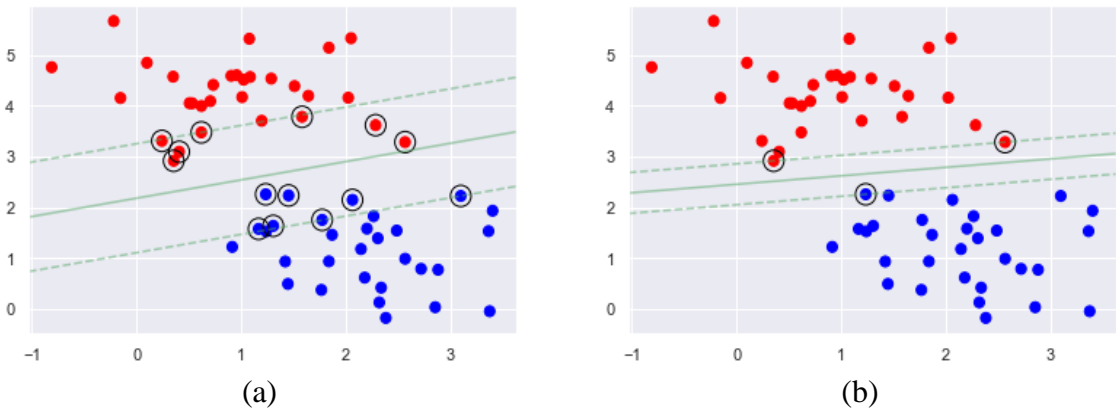
Lagrange çarpıcılar (a_i) elde edildiğinde w_0 değeri, herhangi bir destek vektörünün veya bütün destek vektörlerinin ortalamasının kullanımıyla bulunabilir. Buraya kadar anlatılanlar katı marjin için geçerlidir. Yumuşak marjin de minimize edilecek ifadeye ξ_i hata değişkeni eklendiğinde minimize edilecek ifade ve kısıt koşulları aşağıdaki ifadelere dönüşmektedir.

$$L_p = \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^N \xi_i, \quad y_i(\mathbf{w}^T \mathbf{x}_i + w_0) - 1 + \xi_i = 0, \quad \xi_i \geq 0 \quad (57)$$

Yukarıdaki amaç fonksiyonu indirgenildiğinde yine eşitlik (56) elde edilmektedir. Sonuç olarak, katı marjin sınıflandırıcıyla yumuşak marjin sınıflandırıcı arasındaki tek farkın a_i sınırlamasında olduğu görülmüştür [25].

$$0 \leq a_i \leq C \quad (58)$$

Eşitlik (58)'de C düzenleştirme parametresidir. Aşağıdaki şekilde C parametresinin sınıflandırmadaki görsel etkisi gösterilmiştir. C değeri küçük olduğunda, marjin daha büyük olurken, C değeri arttıkça marjinin küçüldüğü gözükmemektedir. C değerinin küçük olması bazı örneklerin kesik çizgilerden oluşan iki doğru arasında olmasına sebep olurken, C değeri arttıkça iki doğru arasında örnek kalmamaktadır.



Şekil 1.14. DVM yönteminde C parametresine göre marjinin değişiminin bir örnekle görselleştirilmesi, (a) $C = 0.1$, (b) $C = 10$

Doğrusal olarak ayrılamayan örnekler için tıpkı özellik çıkartma algoritmalarında olduğu gibi çekirdek kavramı kullanılabilir. Örnekler ϕ fonksiyonuyla doğrusal olmayan

uzaya izdüşürülmektedir. Doğrusal DVM'deki amaç fonksiyonu, doğrusal olmayan ϕ fonksiyonu ile eşitlik (59)'a dönüşür.

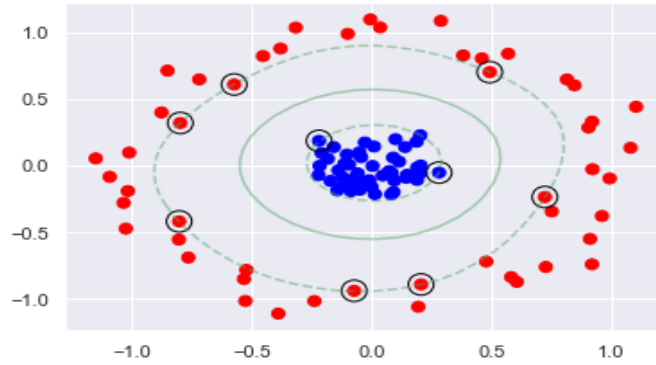
$$L_p = \sum_i^n a_i - \frac{1}{2} \sum_i^n \sum_j^n \alpha_i \alpha_j y_i y_j \phi^T(\mathbf{x}_i) \phi(\mathbf{x}_j) \quad (59)$$

Test örneği \mathbf{x} 'in sınıflandırılması, eşitlikler yeniden düzenlendiğinde aşağıdaki şekli alır.

$$g(x) = \sum_{i \in DV} a_i y_i \phi^T(\mathbf{x}_i) \phi(x) + w_0 \quad (60)$$

L_p 'nin optimizasyonu ve onu izleyen örneğin sınıflandırılması sadece çekirdek fonksiyonu ile değiştirilebilecek olan, doğrusal olmayan uzaydaki öznitelik vektörleri arasındaki iç çarpıma bağlıdır. Sonuç olarak $\phi(x)$ dönüşümünün hesaplanmasından kaçınılarak formülasyondaki iç çarpımlar $k(x, y)$ ile değiştirilmektedir.

Aşağıdaki şekilde doğrusal olarak ayrılamayan sınıflar, ϕ fonksiyonuyla doğrusal olmayan uzaya izdüşürüldükten sonra elde edilen destek vektörleri ve karar sınırı orjinal uzayda gösterilmiştir.



Şekil 1.15. Doğrusal olarak ayrılamayan sınıflar için DVM yöntemiyle sınıflandırmaya bir örnek

DVM, genelleştirme yeteneğinin maksimizasyonu, optimizasyonda yerel minimumunun olmayışı ve düzenleştirme parametreleri sayesinde aykırı değerlere dayanıklılık gibi önemli avantajlara sahiptir. Bunun yanında, eğitim zamanının ve işlem

yükünün fazla olması, parametrelerinin seçiminin zorluğu gibi bazı dezavantajları da mevcuttur.

Çalışmamızda DVM'nin daha da basitleştirilmiş bir hali olan EKK-DVM yöntemi kullanılmıştır.

1.9.2. En Küçük Kareler - Destek Vektör Makineleri

Doğrusal olmayan sınıflandırma ve regresyon problemlerinin dışbükey ikinci dereceden programlama ile çözülmesi doğrusal olmayan DVM'nin avantajlarından biridir. Bu ikinci dereceden programlama (quadratic programming - QP) probleminin sonucu olarak seyreklik de elde edilir. Burada şu soru önem kazanmıştır: DVM formülasyonu, hiçbir avantajından ödün vermeden ne kadar basit düzeye indirgenebilir?

Vapnik'in DVM sınıflandırıcısı formülasyonunda yapılan değişiklik, kullanımı QP çözücülerden daha kolay olan doğrusal denklem kümesinin çözümünü sağlamaktadır. Aşağıdaki DVM değişikliği Suykens tarafından [39] ortaya atılmıştır.

$$\begin{aligned} \min_{\mathbf{w}, b, e} J_p(\mathbf{w}, \mathbf{e}) &= \frac{1}{2} \mathbf{w}^T \mathbf{w} + \gamma \frac{1}{2} \sum_{k=1}^N e_k^2, \\ y_k(\mathbf{w}^T \phi(\mathbf{x}_k) + w_0) &= 1 - e_k, k = 1, \dots, N, \\ y(\mathbf{x}) &= \text{sign}(\mathbf{w}^T \phi(\mathbf{x}) + w_0) \end{aligned} \quad (61)$$

Vapnik'in formülasyonu iki noktada değişime uğramıştır. İlki, eşitsizlik sınırlamaları yerine eşitlik sınırlamalarının kullanılmasıdır. Eşitlik (51)'de eşitsizliğin sağ tarafında eşik değeri olarak kullanılan '1', EKK-DVM için eşitlik (61)'de hedef değer olarak kullanılmaktadır. Bu hedef değerine e_k hata değişkeniyle ulaşılmasına izin verilmektedir. İkincisi ise, hata değişkeni için karesel kayıp fonksiyonunun kullanılmasıdır. Bu değişiklikler problemi oldukça kolay hale getirmektedir.

Genellikle \mathbf{w} 'nin boyutunun çok büyük olmasından dolayı, EKK-DVM için doğrusal olmayan sınıflandırma formülasyonunu ikincil problem biçimine dönüştürmek gerekir. Problem için Lagrangian aşağıdaki eşitliğe dönüşür:

$$L(\mathbf{w}, w_0, \mathbf{e}, \boldsymbol{\alpha}) = \frac{1}{2} \mathbf{w}^T \mathbf{w} + \gamma \frac{1}{2} \sum_{k=1}^N e_k^2 - \sum_{k=1}^N \alpha_k [y_k(\mathbf{w}^T \phi(\mathbf{x}_k) + w_0) - 1 + e_k] \quad (62)$$

Yukarıdaki eşitlikte α_k değerleri Lagrange çarpıcılarıdır. Optimizasyondan sonra aşağıdaki kısıt koşulları elde edilir.

$$\frac{\partial L}{\partial \mathbf{w}} = 0 \rightarrow \mathbf{w} = \sum_{k=1}^N \alpha_k y_k \phi(\mathbf{x}_k) \quad (63)$$

$$\frac{\partial L}{\partial w_0} = 0 \rightarrow \sum_{k=1}^N \alpha_k y_k = 0 \quad (64)$$

$$\frac{\partial L}{\partial e_k} = 0 \rightarrow \alpha_k = \gamma e_k, \quad k = 1, \dots, N \quad (65)$$

$$\frac{\partial L}{\partial \alpha_k} = 0 \rightarrow y_k(\mathbf{w}^T \phi(\mathbf{x}_k) + w_0) - 1 + e_k = 0, \quad k = 1, \dots, N \quad (66)$$

$\mathbf{y} = [y_1; \dots; y_N]$, $\mathbf{1}_v = [1; \dots; 1]$, $\mathbf{e} = [e_1; \dots; e_N]$, $\mathbf{a} = [a_1; \dots; a_N]$ olarak tanımlansın. \mathbf{w} 'yu ve \mathbf{e} 'yi kaldırmak için eşitlik (63) ve (65), eşitlik (66)'da yerine koyulursa sistem koşulları ikiye indirgenmiş olur.

$$y_k[(\sum_{i=1}^N \alpha_i y_i \phi(\mathbf{x}_i))^T \phi(\mathbf{x}_k) + w_0] - 1 + \frac{\alpha_k}{\gamma} = 0, \quad k = 1, \dots, N$$

$$\sum_{k=1}^N \alpha_k y_k = 0 \quad (67)$$

Burada, çekirdek kavramı uygulanıp minimumu alınacak ifadede eşitlik (63) ve (65) yerine yazılırsa optimizasyon problemi aşağıdaki şekli alır:

$$\min \frac{1}{2} (\sum_{k=1}^N \alpha_k y_k \phi(\mathbf{x}_k))^T (\sum_{k=1}^N \alpha_k y_k \phi(\mathbf{x}_k)) + \gamma \frac{1}{2} \sum_{k=1}^N (\frac{\alpha_k}{\gamma})^2 \quad (68)$$

İkili biçimdeki sınıflandırıcı ise eşitlik (63)'ün yerine yazılmasıyla standart DVM'deki gibi hesaplanır [25]:

$$y(x) = \text{sign}(\sum_{k=1}^N \alpha_k y_k k(\mathbf{x}, \mathbf{x}_k) + w_0) \quad (69)$$

1.9.3. Gauss Karışım Modeli

Gözetimsiz öğrenme algoritmalarından biri olan GKM tüm veri noktalarının farklı parametrelere sahip Gauss dağılımlarının bir karışımından geldiği varsayımına

dayanmaktadır. Gerçek veri dağılımını öğrenmeye çalışan parametrik üretken modellerdir [40].

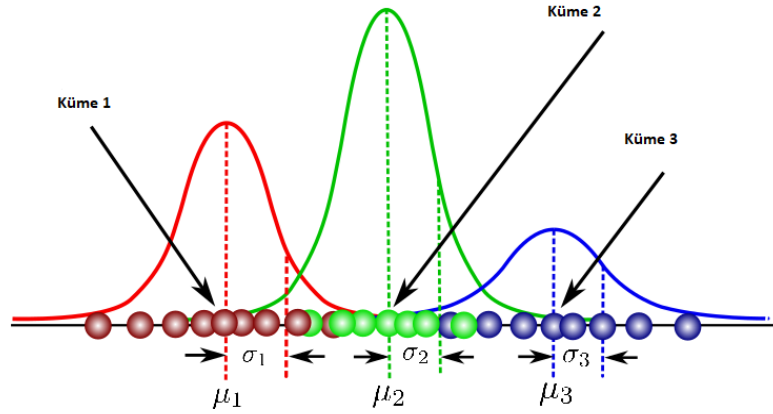
GKM'ler k-Ortalamlar kümeleme algoritmasının yumuşak genellemesi olarak düşünülebilir. k-Ortalamlar gibi, GKM'ler de öğrenme algoritmasına girdi olarak k kümelerinin sayısını talep eder. Ancak, ikisi arasında önemli bir fark, k-Ortalamlar sadece eşit büyüklükteki dairesel formdaki kümeleri öğrenebilirken GKM'ler ise herhangi bir eliptik şekle sahip kümeleri öğrenebilirler.

Ayrıca, k-Ortalamlar bir örneğin sadece bir kümeye ait olmasına izin verirken, GKM'ler her örneği bütün kümelerle belirli oranlarda ilişkilendiren olasılıklar döndürür. Genel olarak, GKM'ler her kümeyi farklı bir Gauss dağılımı olarak öğrenmeye çalışır.

Karışımındaki her Gauss bileşeni aşağıdaki parametrelerden oluşmaktadır.

- Küme ortalaması μ_k
- Küme kovaryans matrisi Σ_k
- Gauss işlevinin ne kadar büyük veya küçük olacağını tanımlayan bir karıştırma olasılığı π_k

Yukarıdaki parametreler aşağıdaki şekilde görselleştirilmiştir.



Şekil 1.16. Gauss Karışım Modelinin Örnek Verileri Üzerinde Görselleştirilmesi [41]

Öncelikle Beklenti Büyütme (Expectation Maximization - EM) algoritması kullanılarak Maksimum Olasılık Optimizasyonu (Maximum Likelihood Optimization) başlatılır. EM algoritması 2 aşamadan oluşmaktadır. Beklenti ve maksimizasyon adımları:

Beklenti adımımda, tahmini parametreleri kullanarak her bir x_n örneğinin benzerliği aşağıdaki formülle hesaplanır. D her örneğin boyutunu belirtmektedir.

$$N_{nk}(x_n|\mu_k, \Sigma_k) = \frac{1}{(2\pi)^{D/2} |\Sigma_k|^{1/2}} \exp\left(-\frac{1}{2} (x - \mu_k)^T \Sigma^{-1} (x - \mu_k)\right) \quad (70)$$

Daha sonra, verilen her x_n örneğinin k . kümeye ait olma olasılığı hesaplanır.

$$P_{nk} = \frac{N_{nk}(x_n|\mu_k, \Sigma_k)\pi_k}{\sum_{k=1}^K N_{nk}(x_n|\mu_k, \Sigma_k)\pi_k}, \quad \mu_k = \frac{\sum_{n=1}^N P_{nk}x_n}{\sum_{n=1}^N P_{nk}}$$

$$\Sigma_k = \frac{\sum_{n=1}^N P_{nk}(x - \mu_k)(x - \mu_k)^T}{\sum_{n=1}^N P_{nk}}, \quad \pi_k = \frac{1}{N} \sum_{n=1}^N P_{nk} \quad (71)$$

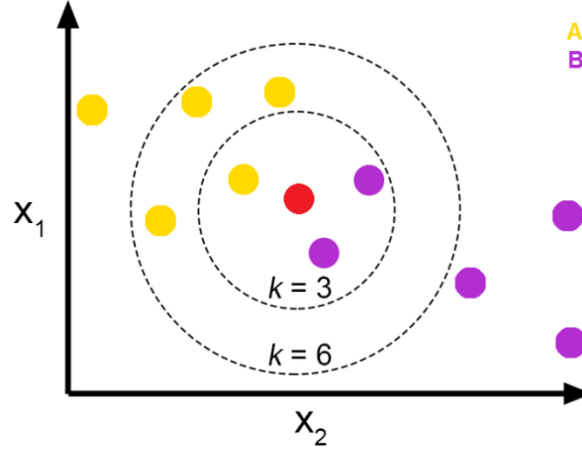
K kümeden her biri güncellediğinde ortalama, kovaryans ve π_k 'lar tekrar hesaplanır. Yakınsamaya kadar adımlar tekrar edilir ve belli bir eşik değeri altında değişimler olduğu noktada algoritma durdurulur.

1.9.4. K-En Yakın Komşu Algoritması

K-EYK yöntemi diğer sınıflandırma yöntemlerine göre basit ve anlaşılması daha kolay olan danışmanlı bir yöntemdir. Yöntem için tek parametre k parametresidir. K-EYK yöntemi sınıflandırılmak istenen bir veriyi daha önceki verilerle olan yakınlık ilişkisine göre sınıflandıran bir algoritmadır. Yeni gelen verinin eğitim verilerine uzaklıkları bulunur. En yakın k örnek içerisinde en çok hangi sınıfa ait veri varsa yeni gelen veri o sınıfa dahil edilir.

Aşağıdaki şekilde sarı ve mor dairelerden oluşan örnekler iki ayrı A ve B sınıfını ifade etsin. Merkezdeki kırmızı örnek sınıflandırılırken, k değerine göre çevresindeki örneklere olan uzaklıklara bakılır. Örneğin $k = 3$ seçildiğinde en yakın 3 örneğin ikisi mor olduğundan kırmızı örnek B sınıfına dahil edilir.

k değeri 6 seçildiğinde ise en yakın 6 örnekten 4 tanesi sarı olduğundan örnek A sınıfına dahil edilir. k değerinin küçük seçilmesi eğitilen modelin aykırı değerlere karşı daha duyarlı olmasına sebep olurken büyük seçilmesi de diğer sınıflardan çok sayıda örnek içermesine neden olacaktır.



Şekil 1.17. K-EYK ile örnek bir test verisinin sınıflandırılması [42]

Yöntem kullanılırken özellik vektörlerinin birbirlerine olan mesafelerinin hesaplanmasında çeşitli uzaklık ölçütleri kullanılmaktadır. En yaygın olanı Öklid uzaklığıdır. K-EYK yöntemi modifiye edilerek bu çalışmada kullanılan Ağırlıklı K-En Yakın Komşu (AK-EYK) yöntemi elde edilmiştir. AK-EYK yönteminde herhangi bir test örneğinin eğitim verisindeki en yakın k örneğe uzaklıkları bulunduktan sonra, her sınıf için ağırlıklar şu formülle hesaplanır: $toplam_c = \sum_{i=1}^{n_c} 1/uzaklik_i$. Burada C sınıf sayısını ifade ederken, n_c de o sınıfa ait örnek sayısını ifade etmektedir. Son olarak test örneği ağırlığı en büyük olan sınıfa dahil edilir [43].

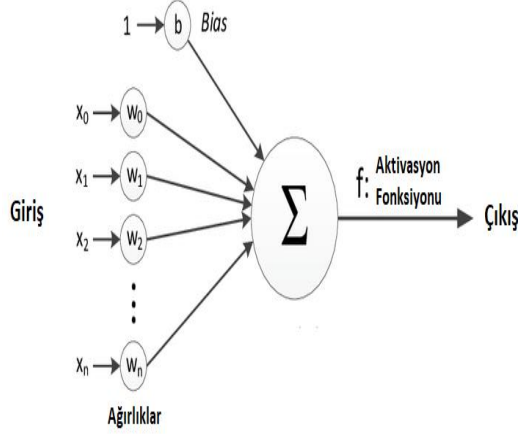
1.9.5. Yapay Sinir Ağları

YSA makine öğrenmesinin hemen hemen her alanında kullanılan ve çok ilgi gören bir yöntemdir. İnsan beyninin çalışma mantığı dikkate alınarak matematiksel bir model geliştirilmiştir. Yapay sinir hücrelerinin çeşitli ağırlıklarla birbirine bağlanması sonucunda model meydana gelir.

YSA'nın gelişimine bakıldığında, ilk hesaplama modelleri olan Threshold Logic Unit 1943 yılında McCulloch ve Piits tarafından [44] önerilmiştir. 1949 yılında Hebb [45] ağırlıkların her kullanışında güçlendiğini göstermiş ve Hebb öğrenme kuralını öne sürmüştür. 1959 yılında Frank Rosenblatt [46] hesaplama işlemlerine dayanan ve örüntü tanımada kullanılabilen tek katmanlı perseptron modelini geliştirmiştir. 1984 yılı ve sonrasında Boltzman Makinası adı verilen yapay nöronlar arasında geri besleme döngülerinin de kullanılabildiği ilk YSA modelleri önerilmeye başlanmıştır. Rumelhart vd. 1986 yılı ve

sonrası yaptıkları çalışmalarla [47] ileri beslemeli yapay sinir ağları için hatanın geriye yayılma algoritmasını geliştirerek YSA'nın gelişmesinde önemli katkıda bulunmuşlardır.

Basit bir yapay sinir ağı girişlerden, girişlere karşılık düşen ağırlıklardan ve aktivasyon fonksiyonlarından oluşmaktadır. Aşağıda şekilde basit bir yapay sinir ağının yapısı gösterilmektedir.



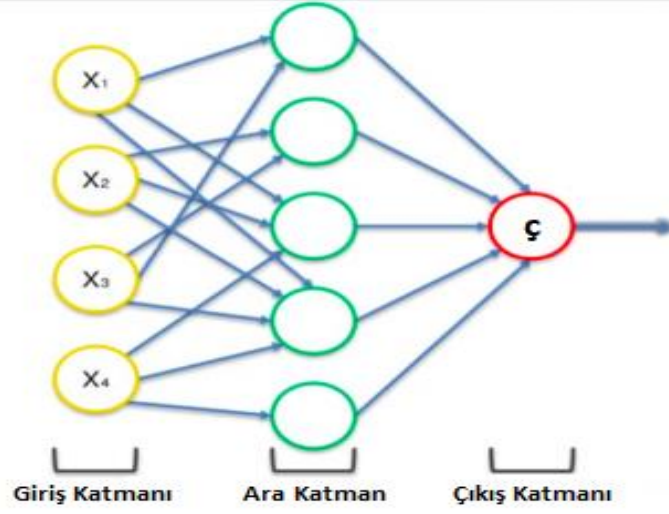
Şekil 1.18. Basit bir yapay sinir ağının yapısı [48]

Girişler ağırlıklarla çarpılarak bias değeri ile toplandıktan sonra elde edilen sonuç aktivasyon fonksiyonundan geçirilir ve çıkış elde edilir. Matematiksel olarak çıkış aşağıdaki eşitlikle hesaplanır.

$$\text{çıkış} = f\left(\sum_{i=0}^n w_i x_i + b\right) \quad (72)$$

Yukarıdaki eşitlikte f aktivasyon fonksiyonunu ifade etmektedir. Aktivasyon fonksiyonunun amacı giriş ve ağırlıkların çarpılıp toplanmasıyla elde edilen değeri çıkışa aktarmadan önce belirli değerler arasında sınırlandırmaktır (ReLU aktivasyon fonksiyonu hariç. ReLU için alttan sınır varken ama üstten sınır yoktur.). Literatürde çeşitli aktivasyon fonksiyonları bulunmaktadır. Bunlardan en popüler olanları Signum, Sigmoid ve Tanh'dır.

Katman ve nöron sayısında değişiklikler yapılarak çeşitli YSA modelleri elde edilebilir. Aşağıdaki şekilde bir giriş, bir ara ve bir de çıkış katmanından oluşan bir model gösterilmiştir.



Şekil 1.19. Girdi, ara ve çıktı katmanlarından oluşan bir ağ yapısı [48]

Katman sayısı ve her bir katmandaki nöron sayısı problemden probleme değişiklik gösterebilir. Çıkış ve giriş katmanındaki nöron sayılarını sırasıyla ağa verilen problemin sınıf sayısı (iki sınıflı hariç) ve her bir örneğin boyutu belirler. Ara katmandaki nöron sayısı ise problemin çözülebilirliğine göre değişmektedir.

Eğitim örnekleri ağa girdi olarak verilerek üretilen çıktıların etiket değerlerine yakınsaması sağlanır ve böylece en iyi performansı verecek ağırlıklar bulunur. Danışmanlı yapay sinir ağlarında etiket değerleri ile çıktı değerleri arasındaki farkların karesel toplamı bize hata fonksiyonunu verir. Bu hata fonksiyonu aşağıdaki eşitlikle hesaplanır.

$$E_{toplam} = \sum \frac{1}{2} (x_{çıkış} - x_{etiket})^2 \quad (73)$$

YSA'da amaç hatayı minimize etmektir. Bu sebeple hatanın minimumuna yakınsamak gereklidir. Bir fonksiyonun lokal minimumuna yakınsamak için en çok kullanılan yöntemlerden biri Gradyan İniş (Gradient Descent) algoritmasıdır. Gradyan İniş, yerel minimuma yakınsamak için birinci türevi kullanır. Aşağıdaki eşitlikte Gradyan İniş ile hesaplanan birinci türev ile ağırlıkların nasıl güncellendiği gösterilmektedir.

$$\mathbf{w}_{ij}^{k+1} = \mathbf{w}_{ij}^k + \Delta \mathbf{w}_{ij}, \quad \Delta w_{ij} = -\alpha \frac{\partial E}{\partial v_{ij}} \quad (74)$$

Örnek olarak bir giriş bir ara bir de çıkış katmanından oluşan bir ağ yapısına sahip olduğumuzu düşünelim. İlk katmandaki ağırlıklar W ile ifade edilirken, ikinci katmandaki ağırlıklar V ile ifade edilsin. Son katmanda sigmoid aktivasyon fonksiyonu kullanıldığı varsayıldığında son katmandaki ağırlıkların güncellemesi aşağıdaki gibi yapılmaktadır.

$$f(x) = \frac{1}{1+\exp(-x)}, \quad f'(x) = f(x)(1 - f(x)) \quad (75)$$

Yukarıda $f(x)$ sigmoid aktivasyon fonksiyonunu ifade ederken, $f'(x)$ $f(x)$ 'in türevini ifade etmektedir. Giriş katmanına giriş olarak X verildiğinde ilk olarak sırasıyla aşağıdaki gibi her katmandaki çıktılar elde edilmektedir. Daha sonra ise gradyan hesaplanmaktadır.

$$X' = f(WX), \quad X'' = f(VX'), \quad \frac{\partial E}{\partial V} = (d - X'')f'(VX')X' \quad (76)$$

d her bir örneğin sınıf etiket bilgisini ifade etmektedir. α ise öğrenme katsayısıdır ve 0 ile 1 arasında bir değer almaktadır. Öğrenme katsayısı küçük seçilirse yerel minimuma ulaşma ihtimali artar ama çok daha fazla sürede minimuma yakınsanır. Bu da işlem yükünü arttırır. Öğrenme katsayısı büyük seçilirse yerel minimum yakalanmayabilir ve yanlış noktalara yakınsama olabilir.

Bir yapay sinir ağının eğitiminin sonlanması için genellikle iki yöntem kullanılır. Bunlardan ilki hatanın belirli bir değerden düşük olmasını beklemektir. Diğerisi ise eğitim sürecini belirli bir devirde (epoch) tekrarlamaktır [49].

1.9.6. Evrişimsel Sinir Ağları

Evrişimsel Sinir Ağları (ESA) görüntüyü bir kaç katmanda işleyen günümüzde kullanılan en popüler sinir ağı modellerinden biridir. Bu sinir ağı modeli görüntü, video gibi görsel verileri sınıflandırma, analiz etme ve tanıma gibi çeşitli alanlarda yüksek doğruluk oranı vermesi sebebiyle kullanılmaktadır.

ESA ile klasik sinir ağlarından farklı olarak veri setinden ilk önce kullanılan filtrelerle amaca uygun özellikler çıkarılır ve daha sonra çıkarılan bu özelliklerle sinir ağlarındaki gibi eğitim gerçekleştirir. ESA'da önemli olan en iyi performansı verecek olan özellikleri bulmaktır [49]. ESA dört adımdan oluşmaktadır.

- Evrişim Katmanı
- Havuzlama (Pooling) Katmanı
- Düzleştirme
- Tam Bağlantılı (Fully Connected) Katmanı

1.9.6.1. Evrişim Katmanı

Bu katman ESA'nın temelidir. Bu aşamada görüntünün özellikleri, katsayıları eğitim boyunca güncellenen filtrelerle algılanmaya çalışılır. Örneğin, bu filtre kenarları algılayacak bir filtre olabilir. Bu filtreler genellikle çok boyutludur ve piksel değerleri içerirler. Gerçekçi örnekler kullanıldığında, örneğin 128x128x3 (genişlik, yükseklik, derinlik) boyutlarındaki veri setine 64 tane 3x3x3 boyutlarında filtre uygulanırsa çıktı olarak 64 tane 128x128 boyutlarında Feature Map elde edilecektir.

Evrişim katmanından sonra çoğunlukla ReLU aktivasyon fonksiyonu kullanılır. ReLU aktivasyon fonksiyonunun kullanılmasının nedeni hız bakımından en iyi sonucu vermesidir. ReLU aktivasyon fonksiyonunun tanımı aşağıdaki gibidir.

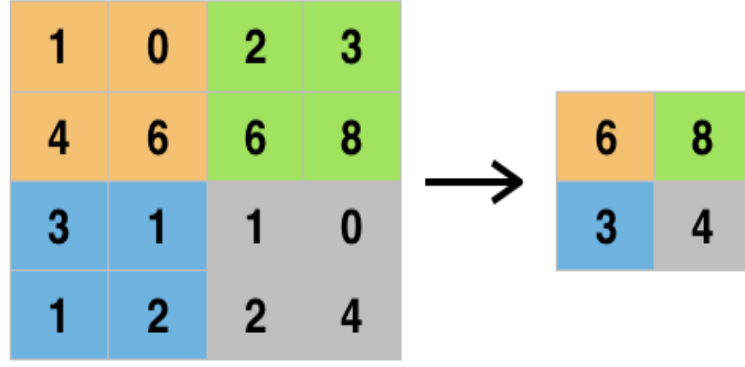
$$f(x) = \max(0, x) \quad (77)$$

1.9.6.2. Havuzlama Katmanı

Elde edilen çok sayıda özellik haritası ağırlık parametre sayısını artırdığından dolayı yüksek bellek boyutu gerektirir. Bu sebeple havuzlama işlemi yapılır. Bir çok havuzlama işlemi mevcuttur. Bunlardan en popüler olanları Maksimum Havuzlama (Max Pooling) ve Ortalama Havuzlama (Average Pooling)'dır.

Birçok uygulamada havuzlama katmanını kullanmak tercih edilmez. Bunun yerine evrişim katmanında daha büyük adımlarla filtreyi kaydırma işlemi tercih edilir.

Aşağıdaki şekilde 4x4 boyutlarındaki matrise 2x2 boyutlarında Maks. Havuzlama uygulandığında elde edilen sonuçlar gösterilmiştir.



Şekil 1.20. Örnek Matris Üzerinde Maksimum Havuzlama İşleminin Uygulanması [50]

1.9.6.3. Düzleştirme Katmanı

Bu katmanda tam bağlantılı katmanın girişleri hazırlanır. Sinir ağı verileri tek boyutlu bir diziden aldığı için evrişim ve havuzlama katmanından gelen veriler tek boyutlu matrise çevirilir.

1.9.6.4. Tam Bağlantılı Katmanı

Bu katmanda elde edilen tek boyutlu özellik vektörü için bir sinir ağı geliştirilir. Doğrusal olmayan özelliklerin çıkarılması için çok katmanlı sinir ağı kullanılır. Eğitim işlemi bu katmanda başlar ve elde edilen hata geriye doğru ilk katmana kadar iletilir.

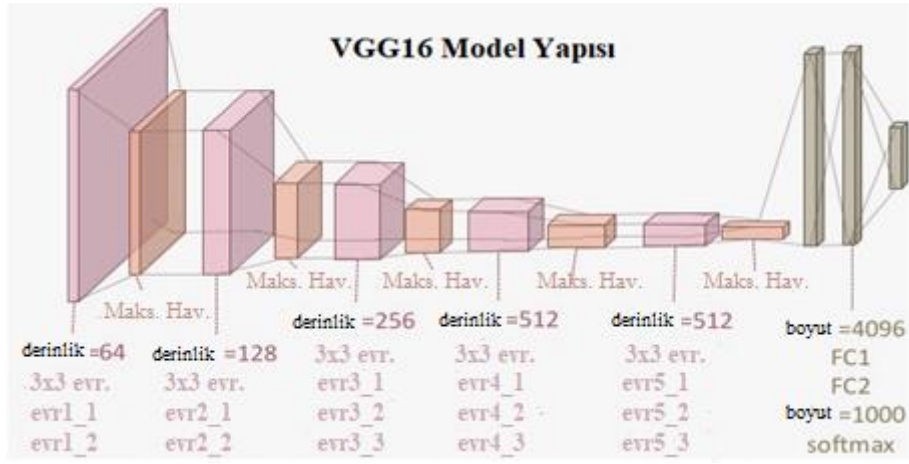
Sınıf etiketli verilerle beslenen ESA çıkış katmanında elde edilen hatayı geriye doğru iletir ve filtrelerin ağırlıklarının güncellenmesi ve hangi filtrelerin seçileceği hata miktarına göre yapılır.

1.9.6.5. Transfer Öğrenme

Probleme göre oluşturulan bazı ESA modelleri veri setinin azlığı sebebiyle aşırı öğrenmeye düşebilir. Bunun üstesinden gelmek için milyonlarca görüntüyle eğitilmiş olan ağı modellerinden yararlanılır. Bu modellerdeki katsayıların sadece son katmanlarındakiler veri setindeki az örneklerle güncellenir. Transfer öğrenme adını verdiğimiz bu yapı için en popüler eğitilmiş ağılar VGG16, VGG19, Inception V3, Xception ve ResNet 50'dir.

Transfer öğrenmede genellikle tam bağlantılı katmandan önceki ağırlık bilgileri korunarak sadece tam bağlantılı katmanındaki ağırlıklar güncelleştirilmektedir. Böylece çok kısa zamanda modelin öğrenmesi sağlanmaktadır. Performansın düşük olduğu durumlarda tam bağlantılı katmandan önceki bazı katmanlar da eğitilerek performansta iyileşme sağlanabilmektedir.

Aşağıdaki şekilde VGG16 modelinin yapısı gösterilmektedir. VGG16 modeli, görüntü tanıma ve sınıflandırma amacıyla oluşturulan ve 1000 sınıfa sahip ImageNet veritabanı ile eğitilmiştir. VGG16 modeli toplamda 16 (13 evrişim ve 3 tam bağlantılı katman) tane katmandan oluşmaktadır.



Şekil 1.21. VGG16 Modeli Yapısı [51]

VGG16 Model yapısı incelendiğinde peş peşe evrişim / katlama işlemlerinin yapıldığı gözükmemektedir. Transfer öğrenme işleminin başarılı olması için çıkarılan ilk özelliklerin genel olması gerekmektedir.

1.9.7. Füzyon İşlemi

Bu kısımda özellik seviyesi, skor seviyesi ve karar seviyesi füzyon işlemlerinden bahsedilmektedir. Özellik seviyesi füzyonda iki modele ait özellikler birleştirilerek tek bir sınıflandırıcı modeline verilmektedir. Skor seviyesi füzyonda ise iki sınıflandırıcıdan elde edilen skorların ortalamasının alınmasıyla yeni skorlar elde edilmektedir. Son olarak, karar seviyesi füzyonu için, AND ve OR mantığı ile iki yöntemin nihai kararında gerçekleştirilir. Bu çalışmada skor seviyesinde füzyon işlemi kullanılmıştır.

Skor füzyon işleminde herhangi iki sınıflandırıcıdan elde edilen skorlar sınıflandırma performanslarına göre deneysel sonuçlarla belirlenmiş katsayılarla çarpılıp toplandıktan sonra nihai skor elde edilmektedir. Performansı daha iyi olan yöntemin katsayısı daha büyük tutularak nihai skorda etkisi daha fazla olması sağlanmaktadır.

$$Skor_{nihai} = W_{dvm}Skor_{dvm} + W_{gkm}Skor_{gkm} \quad (78)$$

Yukarıdaki eşitlikte örnek olarak DVM ve GKM sınıflandırıcılarının füzyonu gösterilmektedir. W_{dvm} ve W_{gkm} deneysel sonuçlarla belirlenmiş olup en iyi performans gösteren katsayıları ifade etmektedir.

1.10. Performans Ölçüm Metriği

Biyometrik sistemlerde güvenlik genel olarak Equal Error Rate (EER) metriği ile ölçülmektedir. EER hesaplamak için test aşamasından sonra elde edilen skarlardan minimum ve maksimum değerler alınarak eşik değerinin alt ve üst limiti belirlenir. Daha sonra skorlar, gerçek kullanıcının (pozitif) ve sahte kullanıcının skorları (negatif) olmak üzere ikiye ayrılır.

Gerçek kullanıcının yaptığı kaydırma hareketlerinin reddedilme olasılığı False Reject Rate (FRR) olarak ifade edilirken, sahte kullanıcının yaptığı kaydırma hareketlerinin kabul oranı ise False Acceptance Rate (FAR) olarak ifade edilir. Eşik değerinin değişimine göre FAR arttıkça FRR azalmakta, FAR azaldıkça FRR artmaktadır. FAR ve FRR'nin eşit olduğu nokta ise EER'yi ifade etmektedir.

FAR ve FRR, minimum değerden başlayıp çok küçük bir adım değeriyle artarak maksimum değere ulaşana kadar her eşik değeri için hesaplanır. FAR, skor değeri eşik değerinden büyük olan sahte kullanıcı örnek sayısının tüm sahte kullanıcı örnek sayısına oranıyla hesaplanır. FRR ise, eşik değerinden küçük olan gerçek kullanıcı örnek sayısının, tüm gerçek kullanıcı-örnek sayısına oranıyla hesaplanır.

$$FAR = \frac{\text{gerçek olarak sınıflandırılan sahte örnek sayısı}}{\text{toplam sahte örnek sayısı}} \quad (79)$$

$$FRR = \frac{\text{sahte olarak sınıflandırılan gerçek örnek sayısı}}{\text{toplam gerçek örnek sayısı}} \quad (8)$$

2. YAPILAN ÇALIŞMALAR, BULGULAR VE İRDELEME

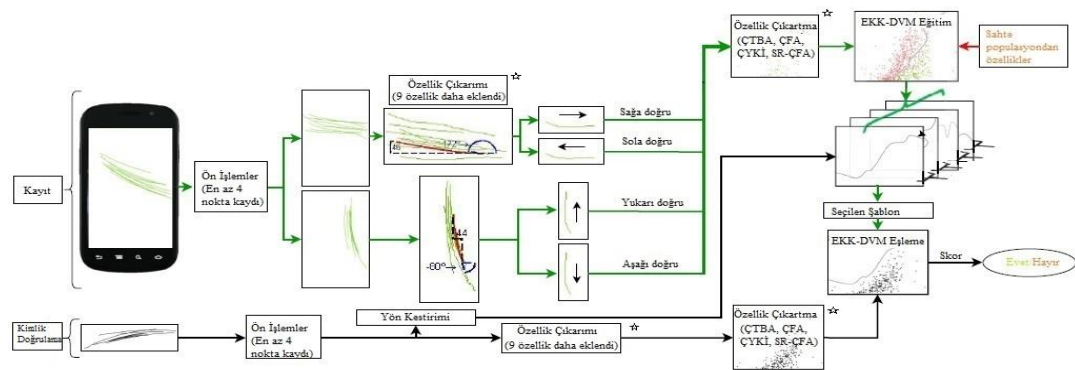
2.1. Giriş

Kullanılan üç veri setinden ilk olarak Serwadda veri setinden [14] biri 37 bir diğeri 5 boyutlu iki özellik vektörü çıkarılmış ve ilk özellik vektörüne özellik çıkartma algoritmaları uygulanmıştır. Özellik çıkartma uygulanan özellik vektörü EKK-DVM ile, 5 boyutlu ikinci özellik vektörü ise GKM ile sınıflandırılmıştır. Her iki sınıflandırıcıdan elde edilen skorlar deneysel sonuçlarla belirlenmiş katsayılarla çarpılarak nihai performans elde edilmiştir.

Daha sonra ise JSS18 veri seti [18] diğer veri setlerinden farklı olarak özellikleri çıkarılmış halde mevcut olduğundan DVM, EKK-DVM, AK-EYK ve YSA ile sınıflandırma yapılarak performans ölçülmüştür.

Son olarak Umdaa veri setinde [17] Serwadda'ya benzer şekilde çıkarılan 28 özelliğe 9 özellik daha eklenmiş ama performansa negatif yönde etki ettikleri için sadece ilk 28 özellik kullanılmıştır. Daha sonra özellik çıkartma algoritmaları uygulanmış ve EKK-DVM ile sınıflandırma yapılmıştır. Ayrıca Umdaa veri seti için kaydırma hareketleri kordinatları yardımıyla görüntüye aktarılmış ve ESA ile de sınıflandırma yapılarak performans ölçülmüştür.

Anlatımı basitleştirmek amacıyla Serwadda veri setinde uygulanan adımlar aşağıdaki şekilde gösterilmektedir. Görselde veri setinden çıkarılan ilk özellik vektörüne (37 boyutlu) uygulanan adımlar gösterilmektedir.



Şekil 2.1. Serwadda veri setinde uygulanan işlemler [1]

Şekil 2.1.'de gözüktüğü gibi ilk olarak kaydırma hareketleri kaydediliyor ve daha sonra sınıflandırma için kullanılacak özellikler çıkarılıyor. Sonraki aşamada ise başlangıç ve bitiş noktalarına göre kaydırma hareketinin yönü kestiriliyor. Son aşamada ise sırasıyla özellik çıkartma ve sınıflandırma işlemleri yapılıyor. Eğitim modeli oluşturulduktan sonra aynı adımlar takip edilerek test örneği sınıflandırıcıya verilerek örneğin hangi sınıfa dahil olacağı kestiriliyor.

2.2. Özellik Belirleme Çalışmaları

2.2.1. Serwadda Veri Seti Üzerinde Özellik Belirleme Çalışmaları

Serwadda veri seti için Bölüm 1.7'de belirtilen ilk 37 özellik kullanılmıştır. Ön işlem olarak ise sadece kaydırma hareketinin en az 4 noktaya sahip olması istenmiştir. Bu çalışmada Bölüm 1.7'deki ilk 28 özelliğe (29-37) numaralı diğer çalışmalardan toplanan 9 özellik daha eklenerek özellik sayısı 37'ye yükseltilmiştir.

Serwadda veri setinden 37 özellikle çalışmayacak şekilde daha önce imza biyometriği için de kullanılan 100 özellik [52] çıkarılmıştır. Bu 100 özellikten 39 tanesi kaydırma biyometriği ile ilgili olmadığından geri kalan 61 özellik kullanılmıştır. Kaydırma biyometriği ile ilgili olan bu 61 özelliğe Sequential Forward Feature Selection (SFFS) algoritması [53] uygulanarak özellik seçme yapılmış ve böylece en iyi sınıflandırma performansını sağlayacak 5 özellik belirlenmiştir. Sonuç olarak Serwadda veri setinden biri 37, bir diğeri 5 boyutlu iki özellik vektörü elde edilmiştir.

Test aşamasında performans ölçülürken her kaydırma hareketi bireysel olarak değerlendirilmektedir. Bir kaydırma hareketi ile kullanıcının kimliğini doğrulamak zor olduğundan dolayı belirli sayıda kaydırma hareketi ile kullanıcıyı tanıma durumu ortaya atılmıştır [12]. Kullanıcı n sayıda kaydırma hareketi girdikten sonra, n sayıda kaydırma hareketinden elde edilen skorların ortalaması alınarak nihai skor elde edilmektedir. Böylece performans artmaktadır.

$$yeniSkor_s = (\sum_{s=1}^{s+n-1} skor_s) / n, \quad s = 1, 2, \dots, (S-n+1) \quad (81)$$

Skor ortalama yöntemi, eşitlik (81) ile ifade edilmektedir. S skor sayısını ifade ederken, n ise skor ortalaması alınacak kaydırma hareketi sayısını ifade etmektedir. Örnek

olarak 40 pozitif ve 100 negatif skordan, $n = 10$ olduğunda ve bir skor kayarak ortalama alınır 31 pozitif skor ve 91 negatif skor elde edilmektedir. Daha sonra bu skordardan EER hesaplanmaktadır.

Kullanıcıdan beklenen kaydırma hareketi sayısının (n) fazla olması performansı artırırken, o andaki kullanıcının dokunmatik ekranla olan etkileşim süresini arttırmaktadır. Bu sebeple n 'yi belirlemek bir ödünleşim problemine dönüşmektedir.

Aşağıdaki tabloda Bölüm 1.7'deki ilk 28 özelliğe EKK-DVM uygulanarak kaydırma hareketlerinin bireysel ve çeşitli sayılarda grup olarak değerlendirilmesinden elde edilen performans sonuçları gösterilmektedir. Tabloya bakıldığında n sayısı arttıkça skor ortalama yönteminin hem dikey hem de yatay kullanımda bütün kaydırma hareket tipleri için performansı arttırdığı gözükmemektedir. Örnek olarak dikey kullanımda yukarıya doğru yapılan kaydırma hareketi için her kaydırma hareketi bireysel değerlendirildiğinde %28.07 EER elde edilirken, 10 kaydırma hareketinin skor ortalaması alındığında ise %18.11 EER elde edilmektedir.

Tablo 2.1. Serwadda veri setinde farklı sayıda örneklerin skor seviyesinde birleştirilmesinin EER değerine etkisi

Yöntemler / Kullanım Yönü		n=1	n=3	n=6	n=10	n=15
Dikey	Yukarı	28.07	22.07	20.03	18.11	17.25
	Aşağı	26.38	20.45	18.35	16.70	15.64
	Sola	23.89	19.67	18.06	16.66	15.79
	Sağa	24.71	20.38	18.73	17.31	16.68
Yatay	Yukarı	22.22	15.36	13.50	11.57	10.29
	Aşağı	22.65	15.98	13.97	12.56	11.64
	Sola	17.65	11.17	9.05	7.03	6.09
	Sağa	17.92	12.09	10.16	8.99	8.16

Bu çalışmada aynı veri setini kullanan diğer çalışmalarla eşit şartlar altında karşılaştırma yapmak amacıyla bundan sonraki çalışmalarda $n = 10$ olarak alınmıştır.

2.2.1.1. Serwadda Veri Seti Üzerinde Eklenen Özelliklerin Performansa Etkisi

Serwadda veri seti için iki özellik vektörü oluşturulmuştur. İlk özellik vektörüne [13] nolu çalışmada kullanılan 28 özellikten farklı 9 özellik daha eklenmiştir. Bu 9 özelliğin performansa katkısı aşağıda verilmiştir. İlk olarak eklenen 9 özellik gruplara ayrılarak

değerlendirilmiştir. Eklenen ilk özellik (29 nolu özellik) yatay kullanımda yukarıya yapılan kaydırma hareketleri hariç diğer bütün kaydırma hareketleri için performansı arttırmıştır. Daha sonra ise 3 özellik (30, 31 ve 32 nolu özellikler) daha eklenerek performans değerlendirilmiş ve her kaydırma hareketi tipi için gelişim sağlanmıştır. Son olarak geriye kalan 5 özellik (33, 34, 35, 36 ve 37 nolu özellikler) daha eklenip performans ölçülmüştür.

Tablo 2.2. Serwadda veri setinde eklenen özelliklerin EER değerine etkisi

Yöntemler / Kullanım Yönü		28 Özellik + EKK-DVM	29 Özellik + EKK-DVM	32 Özellik + EKK-DVM	37 Özellik + EKK-DVM
Dikey	Yukarı	18.11	17.94	17.32	16.51
	Aşağı	16.70	16.51	15.97	15.05
	Sola	16.66	16.49	15.68	14.90
	Sağa	17.31	17.15	16.36	15.44
Yatay	Yukarı	11.57	11.65	10.98	10.16
	Aşağı	12.56	12.39	11.93	11.25
	Sola	7.03	6.86	6.57	6.19
	Sağa	8.99	8.85	8.27	7.45

Tablo 2.2.'de görüldüğü gibi ilk 28 özelliğe eklenen 9 özelliğin, 28 özelliğin performansına göre performansa pozitif yönde katkıda bulunduğu görülmüştür. Dikey kullanımda performansta özelliklerin eklenmesiyle elde edilen gelişim yatay kullanıma göre daha fazladır. En fazla gelişim dikey kullanımda sağa doğru yapılan kaydırma hareketlerinden elde edilirken, en az gelişim ise yatay kullanımda sola doğru yapılan kaydırma hareketlerinden elde edilmiştir. Bölüm (1.7)'deki geriye kalan 38 ve 45 arasındaki özellikler (38 ve 45 dahil) bu veri seti için performansa pozitif yönde katkı sağlamadıkları için değerlendirmeye alınmamışlardır.

2.2.2. JSS18 Veri Seti Üzerinde Kullanılan Özellikler

Bu veri seti Serwadda ve Umdaa veri setlerinden farklı olarak özellikleri çıkarılmış halde mevcuttur. Her kaydırma hareketi için 17 tane özellik çıkarılmıştır. Bu veri seti için kullanılan ve Bölüm 1.7'de belirtilen özelliklerin numaraları şunlardır: (1), (3-5), (25-29), (38-45).

2.2.3. Umdaa Veri Seti Üzerinde Kullanılan Özellikler

Serwadda veri setine eklenen 9 özellik aşağıdaki tabloda görüldüğü gibi Umdaa veri seti üzerinde hiçbir kaydırma hareketi türü için performansa katkıda bulunmadığından dolayı 28 olan özellik sayısında bir değişiklik yapılmamıştır.

Tablo 2.3. Umdaa veri setinde eklenen özelliklerin EER değerine etkisi

Yöntemler / Kullanım Yönü		28 Özellik + EKK-DVM	32 Özellik + EKK-DVM	37 Özellik + EKK-DVM
Dikey	Yukarı	7.56	8.36	8.34
	Aşağı	10.02	10.59	10.53
	Sola	9.06	9.95	9.75
	Sağa	7.96	8.65	9.00

2.3. Özellik Çıkartma ve Sınıflandırma Yöntemleri ile Performans İyileştirme Çalışmaları

Özellik çıkartma algoritmaları yeni uzaya izdüşüm yaptıkları için örnek uzayının boyutu değişmektedir. ÇTBA’da çıkış boyutu %99 kuralı ile belirlenirken, ÇDA’da ise çıkış boyutu sınıf sayısının bir eksigidir.

Tablo 2.4.’te giriş boyutu 37 olan Serwadda veri setine sınıflandırmadan önce ÇTBA, ÇDA ve ÇYKİ özellik çıkartma yöntemleri uygulandığında elde edilen çıkış boyutları verilmiştir.

Tablo 2.4. Özellik Çıkartma Algoritmalarının Serwadda Veri Setine Uygulanmasından Sonra Yeni Uzak Boyutları

Yöntemler / Kullanım Yönü		Çıkış Boyutu		
		ÇTBA	ÇDA	ÇYKİ
Dikey	Yukarı	29	123	18
	Aşağı	29	131	17
	Sola	29	103	26
	Sağa	29	117	15
Yatay	Yukarı	29	53	40
	Aşağı	29	54	100
	Sola	29	38	40
	Sağa	29	44	33

ÇTBA uygulandığında veri setinin yeni boyutu 29 olmaktadır. ÇDA'da ise çıkış boyutu tabloda görüldüğü gibi eğitim verisinin sınıf sayısının bir eksikidir. Son olarak ÇYKİ'de çıkış boyutunu belirleyen herhangi bir kural olmamasından dolayı çıkış boyutu deneysel olarak belirlenmektedir. Bu çalışmada ÇYKİ için en uygun özellik sayısı 20, 30, 40, 50, 60, 70, 80, 90, 100, 110 ve 120 değerleri arasından seçilmiştir.

Çıkış boyutunun büyük olması sınıflandırma probleminin alacağı süreyi arttırmaktadır. Bu sebeple çıkış boyutu ne kadar az olursa sınıflandırma o kadar az zaman almaktadır. Özellikle her kullanıcı için bir sınıflandırma modeli oluşturulan sistemlerde sınıf sayısı kadar eğitim modeli oluşturulacağından dolayı çıkış boyutu daha da önem kazanmaktadır.

2.3.1. Serwadda Veri Seti Üzerinde Sınıflandırma Çalışmaları

Serwadda veri seti 2 oturumda kaydedilmiştir. İki oturum arasındaki zaman farkından dolayı kullanıcı kararlılığı sabit kalmamaktadır. Kullanıcı kararlılığının zamana göre değiştiği düşünülerek sınıflandırma yapılırken sadece oturumlar arası senaryo incelenmiştir. Oturumlar arası senaryoda ilk oturumdaki veriler eğitim için kullanılırken, ikinci oturumdaki veriler test için kullanılmıştır.

Serwadda veri setinden çıkarılan birbirinden farklı iki özellik vektörü bulunmaktadır. Bunlardan ilki olan 37 boyutlu özellik vektörüne sınıflandırmadan önce çeşitli özellik çıkartma algoritmaları uygulanıp daha sonra EKK-DVM ile sınıflandırma yapılmıştır. Özellik çıkartma algoritmalarının sınıflandırmaya ne kadar katkıda bulunduğunu görebilmek için EKK-DVM'den önce özellik çıkartma algoritmalarının kullanılmadığı senaryo ile kullanıldıkları senaryoya arasında kıyaslama yapılmıştır. Serwadda veri setinden çıkarılan 5 boyutlu diğer özellik vektörüne ise GKM uygulanmıştır.

Sınıflandırma yapılırken kullanıcı bazlı sınıflandırma yapılmıştır. Yani her kullanıcı sırasıyla gerçek kullanıcı olarak düşünülüp sınıflandırma yapılmıştır. Bir kullanıcının sınıflandırmaya dahil edilmesi için en az 40 kaydırma hareketine sahip olması istenmektedir. Kullanıcıdan 40 tane kaydırma hareketi rastgele seçilerek gerçek kullanıcının eğitim verisi oluşturulmaktadır. Ardından gerçek kullanıcı hariç diğer bütün kullanıcılar arasından rastgele 4 tane kullanıcı seçilip, seçilen bu 4 kullanıcının her birinden 10 kaydırma hareketi alınarak eğitim için kullanılacak sahte kullanıcının verisi oluşturulmaktadır.

EKK-DVM ile eğitim yapılırken yukarıda bahsedilen 2 sınıflı yapı kullanılmıştır. İlk sınıf gerçek kullanıcının verisi olurken, ikinci sınıf sahte kullanıcının verisi olmaktadır. EKK-DVM ile eğitim modeli oluşturulduktan sonra, gerçek kullanıcının test verisi ile diğer bütün kullanıcıların test verisi eğitim modeline girdi olarak verilerek performans ölçümü için kullanılacak olan skorlar elde edilmektedir.

EKK-DVM sınıflandırıcısı için iki parametre bulunmaktadır. Bunlardan ilki Gauss genişliğini ifade eden gamma parametresidir. Bir diğeri ise marjin genişliğini ayarlayan C parametresidir. EKK-DVM'nin kullanıldığı bütün deneylerde en uygun parametreler aşağıdaki değerlerin kombinasyonu ile elde edilen eğitim modellerinden minimum EER değerini veren modelin seçilmesiyle elde edilmiştir. 4 gamma ve 4 C parametresi olduğundan dolayı toplam 16 model oluşturulmaktadır. Modellerden en küçük EER değerini veren model seçilmektedir.

$$gamma = [1, 10, 100, 1000] \text{ ve } C = [0.1, 1, 10, 100] \quad (82)$$

GKM ile sınıflandırma yapılırken ise sahte kullanıcının eğitim verisine gerek duyulmamaktadır. GKM sadece gerçek kullanıcının eğitim verisinden model oluşturmaktadır. Yine aynı şekilde gerçek kullanıcının test verisi ve diğer bütün kullanıcıların test verisi GKM ile oluşturulan eğitim modeline girdi olarak verilerek performans ölçülmektedir.

Son olarak EKK-DVM ve GKM'nin skorları, toplamaları bire eşit olan deneysel sonuçlarla belirlenmiş katsayılarla çarpıldıktan sonra toplanarak nihai skor elde edilmektedir. Her iki sınıflandırıcı için farklı katsayı kullanılmasının nedeni bireysel performansı iyi olanın etkisini nihai performansta daha etkili hale getirmektir.

2.3.1.1. Serwadda Veri Setinin EKK-DVM ve GMM ile Sınıflandırılması

Biri 37, diğeri 5 boyutlu iki özellik vektörü sırasıyla EKK-DVM ve GKM ile sınıflandırılarak füzyon bir sistem oluşturulmuştur. Mevcut sistemin uygunluğunu teyit etmek için 5 boyutlu özellik vektörü 37 boyutlu özellik vektörüne eklenerek 42 boyutlu özellik vektörü oluşturulmuş ve bu 42 boyutlu özellik vektörü hem EKK-DVM hem de GKM ile sınıflandırılmıştır. Böylece özellikleri tek bir sınıflandırıcıyla eğitmenin füzyon sistemle karşılaştırması yapılmıştır.

Aşağıdaki tabloda görüldüğü gibi 37 boyutlu özellik vektörüne 5 özellik daha eklenerek EKK-DVM ile yapılan sınıflandırmanın sonuçları, 37 boyutlu özellik vektörünün EKK-DVM ile sınıflandırılmasından elde edilen sonuçlardan daha iyi olmasına rağmen EKK-DVM (37) ve GKM (5)'in skor füzyonundan ($0.7 \cdot \text{EKK-DVM} + 0.3 \cdot \text{GKM}$) yatay kullanımda sola ve sağa yapılan kaydırma hareketleri hariç daha başarılı sonuçlar elde edilmiştir. Ayrıca iki özellik vektörünün birleştirilmesinden elde edilen 42 boyutlu özellik vektörü GKM ile de sınıflandırılmış, 5 özelliğin GKM ile sınıflandırılmasından elde edilen sonuçlardan yatay kullanımda sola ve sağa doğru yapılan kaydırma hareketleri hariç daha kötü sonuçlar elde edilmiştir. Sonuç olarak tüm özellikleri birleştirip tek bir sınıflandırıcı kullanmak yerine füzyon sistem kullanmanın daha yararlı olduğu gözlemlenmiştir.

Tablo 2.5. Serwadda veri setinde özellik vektörlerinin tek bir sınıflandırıcıyla sınıflandırılma sonuçları

Yöntemler / Kullanım Yönü		EKK-DVM (37)	EKK-DVM (42)	GKM (5)	GKM (42)	EKK-DVM + GKM
Dikey	Yukarı	16.51	15.54	23.31	31.40	14.71
	Aşağı	15.05	14.10	21.96	29.03	13.10
	Sola	14.90	13.86	18.82	26.48	12.37
	Sağa	15.44	13.93	22.15	34.30	13.73
Yatay	Yukarı	10.16	9.84	18.97	22.14	9.52
	Aşağı	11.25	10.01	20.56	22.16	9.61
	Sola	6.19	5.84	20.70	20.03	6.47
	Sağa	7.45	6.93	20.76	18.95	7.33

Bir sonraki adımda DVM ve EKK-DVM arasında karşılaştırma yapılmıştır. Aşağıdaki tabloda 37 boyutlu özellik vektörüne uygulanan DVM ile EKK-DVM'nin sonuçları gösterilmektedir. Aşağıdaki tabloda da görüldüğü gibi EKK-DVM bütün kaydırma hareketleri tipleri için DVM'den daha başarılı sonuçlar vermektedir. İki yöntem içinde hem aynı parametreler taranmış hem de aynı programlama dili kullanılmıştır. Zaman açısından kıyaslama yapıldığında ise bütün kaydırma hareketleri için DVM, EKK-DVM'den daha kısa sürede sonuç vermektedir.

Tablo 2.6. Serwadda veri setinde EKK-DVM ve DVM sınıflandırmasından elde edilen EER değerlerinin karşılaştırılması

Yöntemler / Kullanım Yönü		EKK-DVM	DVM
Dikey	Yukarı	16.51	17.34
	Aşağı	15.05	16.04
	Sola	14.90	16.04
	Sağa	15.44	16.61
Yatay	Yukarı	10.16	10.47
	Aşağı	11.25	12.57
	Sola	6.19	6.51
	Sağa	7.45	7.93

Daha sonra direkt olarak doğrusal olmayan özellik çıkartma algoritmalarını 37 boyutlu özellik vektörüne uygulamak yerine sınıflandırmadan önce doğrusal özellik çıkartma algoritmaları uygulanarak performans ölçülmüştür. Aşağıdaki tabloda sınıflandırmadan önce uygulanan doğrusal özellik çıkartma algoritmalarının katkısı incelenmiştir. TBA dikey kullanımda performansı iyileştirirken yatay kullanımda ise kötüleştirmektedir. TBA'nın dikey kullanımda yukarı ve sola yapılan kaydırma hareketleri için EER değerlerini sırasıyla %1.39 ve %3.03 düşürdüğü gözükmemektedir. DDA kullanıldığında ise performans hem dikey hem de yatay kullanımda bütün kaydırma hareketleri tipleri için kötüleşmektedir. Sınıflandırmadan önce danışmanlı YKİ kullanmak TBA'da olduğu gibi performansı dikey yönde artırırken yatay kullanımda kötü yönde etkilemektedir.

Tablo 2.7. Doğrusal özellik çıkartma algoritmalarıyla ve EKK-DVM ile elde edilen EER sonuçları

Yöntemler / Kullanım Yönü		EKK-DVM	TBA + EKK-DVM	DDA + EKK-DVM	YKİ + EKK-DVM
Dikey	Yukarı	16.51	15.12	16.60	15.01
	Aşağı	15.05	14.62	16.94	14.78
	Sola	14.90	11.87	15.25	12.62
	Sağa	15.44	14.17	16.84	14.98
Yatay	Yukarı	10.16	11.23	10.85	10.79
	Aşağı	11.25	12.28	12.78	12.46
	Sola	6.19	6.98	8.15	7.57
	Sağa	7.45	8.99	10.05	8.65

Aşağıdaki tabloda ise doğrusal olmayan boyut indirgeme algoritmalarının sınıflandırmadan önce kullanılmasının EER değerlerine etkisi gösterilmektedir. Tablo incelendiğinde ÇTBA yöntemi dikey kullanımda performansı artırırken yatay kullanımda ise çok az kötüleştirdiği gözükmemektedir. ÇDA yönteminde yarı yarıya olmak üzere bazı kaydırma hareketleri için performans arttarken bazılarında ise kötüleşmektedir. ÇYKİ'ye gelindiğinde ise sınıflandırıcı olarak EKK-DVM kullanıldığında yatay kullanımda sağa yapılan kaydırma hareketleri hariç diğer bütün kaydırma hareketleri tipleri için başarılı sonuçlar elde edilmiştir. Doğrusal olmayan özellik çıkartma yöntemleri uygulanırken Gauss genişliğini ifade eden gamma parametresi için 1, 2, 5, 8, 15, 35, 50, 75, 120 değerleri kullanılmıştır.

Tablo 2.8. Çekirdek özellik çıkartma ve EKK-DVM ile elde edilen EER sonuçları

Yöntemler / Kullanım Yönü		EKK- DVM	ÇTBA + EKK- DVM	ÇDA + EKK- DVM	ÇYKİ-1 + EKK- DVM	ÇYKİ-2 + EKK- DVM	ÇYKİ-2 + AK- EYK	EKK- DVM [18]
Dikey	Yukarı	16.51	14.24	16.43	14.31	14.65	18.82	23.8
	Aşağı	15.05	13.89	15.70	13.71	13.94	18.55	22.9
	Sola	14.90	11.71	14.63	11.31	11.71	15.54	21.6
	Sağa	15.44	14.51	16.58	13.79	13.69	18.28	22.9
Yatay	Yukarı	10.16	10.33	9.94	9.65	9.66	12.55	14.7
	Aşağı	11.25	11.65	10.36	10.16	10.12	12.10	16.1
	Sola	6.19	6.59	5.86	5.88	5.50	7.55	13.2
	Sağa	7.45	8.23	8.04	7.77	7.92	10.47	11.8

Tablo 2.8.'de ÇYKİ-1 danışmansız yöntemi belirtirken ÇYKİ-2 ise danışmanlı olanı göstermektedir. İkiyi kıyaslandığında ve sınıflandırma için EKK-DVM kullanıldığında ise çoğunlukla danışmansız ÇYKİ daha iyi sonuçlar vermektedir. Sınıflandırıcı olarak EKK-DVM'nin kullanıldığı ve danışmansız ÇYKİ ve ÇTBA'nın karşılaştırıldığı durumda ise danışmansız ÇYKİ'nin daha iyi sonuçlar ürettiği gözükmemektedir. Ayrıca, EKK-DVM ve AK-EYK sınıflandırıcıları arasında kıyaslama yapıldığında bütün kaydırma hareketleri için EKK-DVM daha iyi performans göstermektedir. Son olarak [18] nolu çalışmaya göre kıyaslama yapıldığında bütün kaydırma hareketlerinde daha başarılı sonuçlar elde edilmiştir.

Danışmansız ÇYKİ için bir diğer parametre ise komşuluk matrisi oluşturulurken kullanılacak olan k parametresidir. k parametresi her örneğin diğer bütün örneklerle uzaklıkları bulunduğundan ve küçükten büyüğe sıralandıktan sonra bu uzaklıkların kaç

tanisinin komşuluk matrisinde kullanılacağını belirlemektedir. Herhangi bir örneğe en yakın k örnek dışında kalan örnekler komşuluk matrisinde sıfıra setlenecektir. 20, 30, 40, 50, 60 değerleri içerisinde en uygun k değeri 40 olarak bulunmuştur. Danışmanlı ÇYKİ’de komşuluk matrisi oluşturulurken uzaklıklar sadece sınıf içi hesaplandığından böyle bir parametreye ihtiyaç duyulmamıştır.

Son olarak EKK-DVM ve GKM sınıflandırıcılarından elde edilen skorlar toplamaları bir olan belirli ağırlıklarla çarpılarak en iyi sınıflandırma performansını verecek olan ağırlıklar aranmış ve skor füzyonu yapılmıştır. Aşağıdaki tabloda her iki modelin skorları katsayılarla çarpılarak elde edilen yeni skarlardan üretilen EER sonuçları gösterilmektedir. Tablo incelendiğinde yatay kullanımda sağa doğru yapılan kaydırma hareketleri hariç diğer bütün kaydırma hareketleri tiplerinde skor füzyonunun kullanılmasıyla performans artmıştır. Ayrıca skor füzyonu EKK-DVM’den elde edilen performansı ortalama %0.8 arttırmaktadır.

Tablo 2.9. Serwadda veri setinde farklı parametrelerle EKK-DVM ve GKM füzyonu gerçekleştirilmesinin EER üzerindeki etkisi

Yöntemler / Kullanım Yöntü		ÇYKİ-1 + EKK-DVM	GKM	0.6* EKK-DVM + 0.4* GKM	0.7* EKK-DVM + 0.3* GKM	0.8* EKK-DVM + 0.2* GKM	0.9* EKK-DVM + 0.1* GKM	EKK-DVM + GMM [18]
Dikey	Yukarı	14.31	23.31	13.94	13.60	13.59	13.88	17.4
	Aşağı	13.71	21.96	13.07	12.92	13.01	13.28	15.9
	Sola	11.31	18.82	10.75	10.52	10.51	10.95	13.9
	Sağa	13.79	22.15	13.26	12.96	13.03	13.38	16.1
Yatay	Yukarı	9.65	18.97	9.70	9.57	9.48	9.53	10.7
	Aşağı	10.16	20.56	8.82	8.71	8.91	9.32	12.2
	Sola	5.88	20.70	7.39	6.56	6.04	5.70	12.1
	Sağa	7.77	20.76	9.18	8.40	8.06	7.94	12.0

EKK-DVM ve GKM kullanan ve her iki yöntemin skorlarını da 0.5 değerleriyle çarpan [18] nolu çalışmaya göre kıyaslama yapıldığında hem yatay hem de dikey kullanım için bütün kaydırma hareketlerinde performans önemli derecede artmıştır. Performansta en büyük artış yatay kullanımda sola doğru yapılan kaydırma hareketlerinde olmuştur. Bu da sınıflandırmadan önce özellik çıkartma algoritması kullanmanın performansı arttırdığını göstermektedir.

2.3.2. JSS18 Veri Seti Üzerinde Sınıflandırma Çalışmaları

JSS18 veri seti özellikleri çıkarılmış halde paylaşıldığından dolayı sınıflandırma için kullanılacak özellikler Serwadda'da kullanılan özelliklerden farklılık göstermektedir. Kaydırma hareketleri 8 oturumda kaydedilmiştir. Eğitim ve test verisi oluşturulurken kullanıcı kararlılığının değişimini minimize etmek için, her oturumdaki kaydırma hareketleri sıralı olmak üzere beşerli gruplara ayrılmıştır. Daha sonra tek numaralı gruplar eğitim, çift numaralı gruplar ise test için olmak üzere ayrılarak eğitim ve test verisi oluşturulmuştur.

JSS18 veri setinin kullanıldığı çalışmada kaydırma hareketleri yönlerine göre gruplara ayrılmamış olup bu çalışmada gruplara ayırmanın performansa katkısı da incelenmiştir. Kaydırma hareketleri yönlerine göre gruplandıktan sonra, her yön için elde edilen kaydırma hareketi sayısı değişmektedir. Örneklerin yarısı eğitim için kullanılırken diğer yarısı test için kullanılmaktadır.

JSS18 veri setinde tek bir özellik vektörü olduğundan dolayı sınıflandırma sadece DVM, EKK-DVM, YSA ve AK-EYK ile yapılmıştır. Serwadda veri setinde olduğu gibi sınıf modeli oluşturulurken kullanıcı bazlı eğitim modeli (iki sınıflı) oluşturulmuştur. İki sınıflı modelin yanında YSA için çok sınıflı yapı da kullanılmıştır. Sahte kullanıcı oluşturulurken toplam örnek sayısının gerçek kullanıcıdaki örnek sayısına yakın olmasına ve her kullanıcıdan alınan örnek sayısının dengeli olmasına özen gösterilmiştir. Sınıflandırıcı olarak karşılaştırma amacıyla EKK-DVM yönteminin yanında AK-EYK yöntemi de kullanılmıştır. Sınıflandırmadan önce çeşitli özellik çıkartma algoritmaları kullanılarak performansa katkıları da incelenmiştir.

2.3.2.1. JSS18 Veri Setinin YSA ve EKK-DVM ile Sınıflandırılması

Tıpkı Serwadda veri setinde olduğu gibi bu veri setinde de kaydırma hareketleri yönlerine göre 4 gruba ayrılıp öyle sınıflandırılmıştır. İlk olarak YSA ile sınıflandırma yapılmıştır. Bu deneylerde her iki kullanım yönünde de aşağı ve yukarıya doğru yapılan kaydırma hareketleri için örnek sayıları az olduğundan bu kaydırma hareketleri için sınıflandırma yapılmamıştır. Aşağıdaki tabloda ilk olarak YSA ile sınıflandırmanın sonuçları verilmektedir. Tabloda YSA - 1 iki sınıflı kısmi verinin kullanıldığı yapıyı (sahte kullanıcıyı oluşturmak için gerçek kullanıcı hariç, diğer kullanıcılardan bir miktar verinin

alınması) gösterirken, YSA - 2 bütün verinin kullanıldığı çok sınıflı YSA yapısını, YSA - 3 yine iki sınıflı yapı olup bütün verilerin kullanıldığı yapıyı göstermektedir. YSA - 4 ise iki sınıflı yapı olup, iki sınıf arasındaki örnek dengesizliğini ortadan kaldırmak için her gerçek kullanıcının tüm örnek sayısının, tüm sahte kullanıcıların örnek sayısına eşit olana kadar tekrarlandığı durumdur.

Aşağıdaki tablodaki veriler incelendiğinde çok sınıflı (YSA - 2) yapıdaki sonuçların diğer üç YSA modeline göre daha iyi olduğu gözükmemektedir. YSA sınıflandırma yöntemi kullanılırken örnek sayısı önemli bir parametredir. Tablo incelendiğinde örnek sayısının az olduğu YSA - 1 modelinin en düşük performansa sahip olması YSA'da örnek sayısının önemini göstermektedir. Tablo incelendiğinde YSA - 4 modelindeki örneklerin tekrarlanmasının YSA - 3 modeline göre performansa olumlu yönde katkı sağlamadığı da gözükmemektedir.

Tablo 2.10. YSA modellerinin JSS18 veri seti üzerindeki EER sonuçları

Yöntemler / Kullanım Yönü		YSA - 1	YSA - 2	YSA - 3	YSA - 4
Dikey	Sola	9.76	6.24	6.85	7.08
	Sağa	10.42	6.96	7.80	8.14
Yatay	Sola	8.09	5.41	5.77	5.89
	Sağa	8.28	5.62	6.04	6.31

Sonraki aşamada JSS18 veri seti iki sınıflı eğitim mantığıyla DVM, EKK-DVM ve AK-EYK ile sınıflandırılmıştır. Aşağıdaki tablo incelendiğinde ise ilk olarak kaydırma hareketlerinin yönlerine göre gruplandırılmasının bütün kaydırma hareketleri türlerinde performansa pozitif yönde katkıda bulunduğu gözükmemektedir. Ayrıca sınıflandırma algoritmaları karşılaştırıldığında en iyi performansı EKK-DVM sınıflandırıcısı vermektedir.

DVM ve EKK-DVM yöntemleri için Serwadda veri setinde kullanılan aynı 4 gamma ve 4 C parametresi kullanılmıştır. AK-EYK sınıflandırıcısı için ise 6 (1, 3, 5, 7, 9, 11) k değeri kullanılmıştır. Aşağıdaki tabloda belirtilen YSA modelinin parametrelerine bakıldığında ise, 4 katmanlı bir ağ (iki gizli katman, sırasıyla 60 ve 80 nörona sahip) kullanılmış olup, optimize edici olarak Adam kullanılmıştır. Ayrıca iki gizli katman için, hızlı olması ve daha iyi performans vermesi sebebiyle reLU, son katman için ise sigmoid aktivasyon fonksiyonu kullanılmıştır.

Tablo 2.11. İki sınıflı yapıyla JSS18 veri seti üzerinde elde edilen EER sonuçları

Yöntemler / Kullanım Yönü		YSA	DVM	AK-EYK	EKK-DVM	Grupsuz EKK-DVM
Dikey	Yukarı	-	10.34	8.71	8.82	10.89
	Aşağı	-	9.06	9.13	8.56	
	Sola	9.76	8.72	9.43	8.20	
	Sağa	10.42	9.48	9.25	8.73	
Yatay	Yukarı	-	9.71	8.72	8.42	8.85
	Aşağı	-	8.37	7.05	7.07	
	Sola	8.09	7.77	7.60	7.15	
	Sağa	8.28	7.66	7.64	7.01	

Aşağıdaki tabloda EKK-DVM sınıflandırıcısından önce 4 özellik çıkartma algoritması kullanılarak performanslar ölçülmüştür. Tablo incelendiğinde sınıflandırmadan önce ÇTBA algoritması kullanmanın performansa pozitif yönde katkı yaptığı gözükmemektedir. Sınıflandırmadan önce ÇDA kullanıldığında ise performans ÇTBA'ya göre daha iyi olmaktadır. Serwadda veri setinde en iyi performansı ÇYKİ yöntemi vermesine rağmen bu veri setinde ise SR-ÇDA ve ÇDA yöntemlerinin gerisinde kalmaktadır. Bu veri setinde danışmansız yöntemin çok düşük performans vermesinden dolayı sadece danışmanlı ÇYKİ yönteminin sonuçları gösterilmiştir. Dört algoritma karşılaştırıldığında ise en iyi performans büyük çoğunlukla SR-ÇDA algoritması göstermektedir.

Tablo 2.12. Özellik çıkartma algoritmalarının JSS18 veri setindeki performans sonuçları

Yöntemler / Kullanım Yönü		EKK-DVM	ÇTBA + EKK-DVM	ÇDA + EKK-DVM	ÇYKİ + EKK-DVM	SR-ÇDA + EKK-DVM
Dikey	Yukarı	8.82	7.47	6.20	5.97	6.34
	Aşağı	8.56	7.18	5.20	5.56	5.22
	Sola	8.20	7.39	6.23	6.53	6.07
	Sağa	8.73	7.92	6.73	6.76	6.65
Yatay	Yukarı	8.42	7.09	6.04	6.08	5.82
	Aşağı	7.07	5.99	4.42	4.46	4.36
	Sola	7.15	6.32	5.25	5.42	5.16
	Sağa	7.01	6.39	5.33	5.44	5.26

Kullanılan dört özellik çıkartma algoritması da doğrusal olmayan şekilde kullanıldığından gamma parametresine ihtiyaç duyulmuştur. Dört algoritma için de 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.75, 0.8, 1, 2 ve 4 değerleri gamma seçiminde kullanılmıştır. ÇYKİ yönteminin bir diğer parametresi olan boyut sayısı için ise 20, 25, 30, 35, 40 ve 45

değerleri kullanılmıştır. Danışmanlı ÇYKİ kullanıldığı için k parametresine gerek duyulmamıştır.

Tablo 2.11.'deki EKK-DVM ve AK-EYK sonuçları ile aşağıdaki tabloda bulunan SR-ÇDA yöntemi uygulanmış aynı yöntemler kıyaslandığında EKK-DVM ve AK-EYK sınıflandırma yöntemlerinden önce kullanılan SR-ÇDA yönteminin performansı her iki yöntem için önemli derecede artırdığı gözükmemektedir. Aşağıdaki tabloda EKK-DVM ve AK-EYK karşılaştırıldığında ise dikey kullanımda yukarıya doğru yapılan kaydırma hareketleri hariç EKK-DVM daha iyi performans göstermektedir. SR-ÇDA ve EKK-DVM uygulandıktan sonra elde edilen EER sonuçlarına bakıldığında yatay kullanımdaki performansın dikeye göre daha iyi olduğu gözükmemektedir. SR-ÇDA ve EKK-DVM yöntemlerinden elde edilen sonuçlara uygulanan skor ortalama ve skor maks. yöntemlerinin sonuçlarına bakıldığında en iyi performans dikey kullanımda aşağıya doğru yapılan kaydırma hareketleri için elde edilmiştir.

Serwadda veri setinde olduğu gibi kaydırma hareketlerinin bireysel değerlendirilmesi yeterli performans sağlamadığı için n skor ortalama burada da kullanılmıştır. Bu deneyde aynı veri setini kullanan çalışmayla karşılaştırma yapmak amacıyla $n = 5$ olarak alınmıştır. Örnek olarak dikey kullanımda aşağıya doğru yapılan kaydırma hareketleri bireysel olarak değerlendirildiğinde %5.22 EER elde edilirken, bir kaydırma hareketi kayarak, her beşli kaydırma hareketinin skor ortalaması alınıp yeni skorlar oluşturulduğunda %0.94 EER elde edilmektedir.

Tablo 2.13. İki sınıflı yapıyla JSS18 veri seti üzerinde elde edilen EER sonuçları

Yöntemler / Kullanım Yönü		SR-ÇDA + AKEYK	SR-ÇDA + EKK-DVM	5 Skor Ortalama	5 Maks.	Syed vd. [20]
Dikey	Yukarı	6.14	6.34	1.11	1.69	5.16
	Aşağı	6.17	5.22	0.94	1.58	
	Sola	6.62	6.07	2.71	2.89	
	Sağa	6.97	6.65	3.03	3.37	
Yatay	Yukarı	6.09	5.82	1.15	1.29	3.80
	Aşağı	4.26	4.36	1.00	1.31	
	Sola	5.59	5.16	2.64	2.52	
	Sağa	5.61	5.26	1.99	2.44	

Eğer n sayısı artarsa başarı daha da artacaktır. n değerinin büyük olmasının önemli bir dezavantajı ise kullanıcının oturumun başında doğrulanmasına kadar geçecek süreyi

arttırmasıdır. Çünkü kullanıcıdan n sayıda kaydırma hareketi yapması beklenecek ve bu da gerçek ya da sahte kullanıcının dokunmatik ekranla daha fazla etkileşimde kalmasına neden olacaktır. Bu süre içinde cihaz içindeki önemli bilgiler tehlikeye girecektir. Bu sebeple n değerinin seçilmesi önemlidir.

Skor ortalama yönteminden farklı olarak kullanılan bir diğer yöntem ise 5 adet skorun belirli bir eşik değerinden geçirilerek dâhil oldukları sınıflardan maksimum sayıda skor içeren sınıfa atanmasıdır. Tablo 2.13 incelendiğinde skor ortalama yönteminin yatay kullanımda sola yapılan kaydırma hareketleri hariç diğer bütün kaydırma hareketleri tiplerinde diğer yöntem (skor maks.) göre daha başarılı olduğu gözükmemektedir.

Aynı veri setini kullanan çalışmada kaydırma hareketleri 4 gruba ayrılmayarak sınıflandırma yapılmıştır. Dikey kullanımda %5.16 ve yatay kullanımda ise %3.8 EER elde edilmiştir. Bu çalışmada ise kaydırma hareketleri 4 gruba ayrılarak sınıflandırılmıştır ve skor ortalama yöntemi için dikey kullanımda en kötü performans %3.03 iken yatay kullanımda bu sayı %2.64'dür. Bu da bize kaydırma hareketlerini sınıflandırmadan önce başlangıç ve bitiş konumlarına göre gruplayıp ve sınıflandırmadan önce özellik çıkartma algoritması kullanmanın performansı arttırdığını göstermektedir.

2.3.3. Umdaa Veri Seti Üzerinde Sınıflandırma Çalışmaları

Umdaa veri setinin diğer veri setlerinden en önemli farkı kullanıcı başına örnek sayısının diğer veri setlerine göre çok olmasıdır. Örnek sayısının fazla olması öğrenme sürecinde örnek sayısının az olmasından kaynaklanan dezavantajları ortadan kaldırmaktadır. Eğitim verisi için her kullanıcıdan 40 örnek alarak oluşturulan model sınıflandırıcıların öğrenmesi için yetersiz kalabilmektedir.

Ön işlem olarak kaydırma hareketinin en az 4 nokta kaydına sahip olması, düzgün kaydedilmesi ve kaydırma hareketinin uzunluğunun 5'ten büyük olması istenmiştir. Serwadda ve JSS18 veri setlerinde kaydırma hareketleri başlangıç ve bitiş noktalarına göre 4 gruba (yukarı, aşağı, sol ve sağ) ayrılırken bu veri setinde dörtlü gruplandırmanın yanında kullanıcı başı örnek sayısı fazla olmasından faydalanılarak açığa göre 6 gruba ayrılarak da performans ölçülmüştür.

Diğer veri setlerinde olduğu gibi kullanıcının eğitime dahil olması için test verisi hariç eğitim setinin en az 40 örneğe sahip olması gerekmektedir. Diğer veri setlerinde eğitim modeli oluşturulurken her kullanıcıdan eşit sayıda örnek alınırken bu veri seti için

bu gözardı edilmiştir. Kullanıcının sahip olduğu bütün kaydırma hareketleri sınıf başı 1000 örneği aşmayacak şekilde eğitime dâhil edilmektedir. 1000 örneği geçen durumlarda ise kullanıcıdan rastgele 1000 örnek seçilmektedir.

Veri setinin ilk %70'lik kısmı eğitim için kullanılırken geri kalan %30'u test için kullanılmıştır. Özellik çıkartma algoritmaları uygulanırken örnek sayısının çok olması ve bunun sonucunda çekirdek ve komşuluk matrisinin hesaplanması uzun süre alacağından dolayı özellik çıkartma algoritmalarının kullanıldığı deneylerde eğitim verisi üzerinde kullanıcı başı örnek sayısı 400'den fazla olan kullanıcılardan 400 örnek rastgele seçilmiştir.

Serwadda ve JSS18 veri setlerinin performanslarının ölçümünde olduğu gibi Umdaa veri seti için de skor ortalama işlemi yapılmıştır. Aynı veri setini kullanan çalışma ile kıyaslama yapmak amacıyla $n = 16$ alınmıştır.

Son olarak kaydırma hareketlerinin görünümünün ayırıcı yapısından faydalanmak amacıyla kaydırma hareketleri koordinatlarından yararlanılarak görüntüye aktarılmıştır. Her görüntü için kullanılan kaydırma hareketi sayısı 16 olarak belirlenmiş olup ve bir sonraki görüntü için ise bir kaydırma hareketi atlayarak görüntüler elde edilmektedir.

Bir kaydırma hareketi boş bir siyah tuval üzerine aktarılırken kaydırma hareketinin piksel değeri iki ayrı duruma göre belirlenmektedir. İlk durumda kaydırma hareketlerinin piksel değerleri direkt olarak 255 değerine setlenip 3 kanalın (RGB) da aynı değere sahip olduğu resimler üretilmektedir.

İkinci durumda ise RGB piksel değerleri sırasıyla o resimdeki kaydırma hareketlerinin normalize edilmiş hız, ivme ve basınç özelliklerinin ortalamasına göre belirlenmektedir. Normalize edilmiş bilgiler 255 değeriyle çarpılarak piksel değerleri elde edilmektedir. Böylece her üç kanalın da farklı piksel değerlerine sahip olduğu 270x480 boyutlarında görüntüler elde edilmektedir. Daha sonra oluşturulan görüntülerin boyutları ölçeklenerek 32x56 yapılmaktadır.

Aşağıdaki şekilde aynı kullanıcıya ait kaydırma hareketlerinden elde edilen ardışık iki örnek görüntü gösterilmektedir.



Şekil 2.2. Kaydırma hareketlerinin görüntüye aktarılmasına örnekler

Son olarak elde edilen görüntüler ESA ile sınıflandırılmıştır. Ayrıca veri setinin az örnek içermesi sebebiyle ESA ile sınıflandırma yapılırken gerçekleştirilecek aşırı öğrenme gibi durumlardan dolayı VGG16 modeli üzerinde Transfer Learning işlemi de yapılmıştır.

2.3.3.1. Umdaa Veri Setinin EKK-DVM ile Sınıflandırılması

İlk olarak kaydırma hareketleri başlangıç-bitiş noktaları ve açığa göre sınıflandırıldığında elde edilen performanslar karşılaştırılmıştır. Açığa göre gruplama yapılırken çapraz kaydırma hareketleri için çok az örnek olduğundan değerlendirmeye alınmamışlardır. Tablo 2.14. incelendiğinde kaydırma hareketleri açığa göre gruplandırıldığında elde edilen sonuçlar, başlangıç ve bitiş noktalarına göre yapılan gruplamaya göre aşağıya doğru yapılan kaydırma hareketleri hariç daha başarılı sonuçlar vermiştir.

Tabloda kaydırma hareketlerinin performansı ölçülürken skor ortalama yöntemi kullanılmış ($n = 16$) ve Umdaa veri seti üzerinde EKK-DVM ile elde edilen EER performansları gösterilmiştir.

Aynı veri setini kullanan çalışmayla karşılaştırma yapmak amacıyla bundan sonraki çalışmalarda başlangıç ve bitiş noktalarına göre yapılan gruplama sonucunda elde edilen sonuçlarla devam edilecektir.

Tablo 2.14. Umdaa veri seti üzerinde kullanım yönü sayısı farklılığında EKK-DVM yöntemiyle elde edilen EER değerleri

Yöntemler / Kullanım Yönü		EKK-DVM 4 Grup	EKK-DVM 6 grup
Dikey	Yukarı	7.56	7.36
	Aşağı	10.02	10.30
	Sola	9.06	6.07
	Sağa	7.96	7.29

Daha sonraki aşamada farklı sınıflandırma algoritmalarının Umdaa veri seti üzerindeki performansları karşılaştırılmıştır. Aşağıdaki tablo incelendiğinde ve üç sınıflandırma yöntemi karşılaştırıldığında en kötü performans AK-EYK yöntemi ile elde edilmektedir. DVM ve EKK-DVM karşılaştırıldığında ise diğer veri setlerinde olduğu gibi EKK-DVM bu veri setinde de daha iyi sonuçlar vermektedir. DVM ve EKK-DVM yöntemleri için Serwadda ve JSS18 veri setlerinde kullanılan 4 gamma ve 4 C parametresi kullanılmıştır. AK-EYK sınıflandırıcısı için ise 6 k değeri (1, 3, 5, 7, 9, 11) kullanılmıştır. Farklı parametrelerle oluşturulan sınıflandırma modellerinden en düşük EER değerine sahip olan model test için kullanılmaktadır.

Tablo 2.15. Umdaa veri seti üzerinde farklı sınıflandırma yöntemleri ile elde edilen EER değerleri

Yöntemler / Kullanım Yönü		AK-EYK	DVM	EKK-DVM
Dikey	Yukarı	10.65	8.56	7.56
	Aşağı	14.27	11.29	10.02
	Sola	14.07	10.62	9.06
	Sağa	10.24	9.16	7.89

Son olarak EKK-DVM yönteminden önce çeşitli özellik çıkartma yöntemleri uygulanarak performans arttırılmaya çalışılmıştır. Tablo incelendiğinde EKK-DVM yönteminden önce kullanılan dört algoritma içerisinde en kötü performans ÇTBA yöntemi ile elde edilirken, en iyi performans ise SR-ÇDA yöntemi ile elde edilmiştir. Özellik çıkartma yöntemi olarak SR-ÇDA kullanılıp sınıflandırıcı olarak ise EKK-DVM ve AK-

EYK yöntemleri kullanıldığında ve performanslar karşılaştırıldığında EKK-DVM yöntemi bütün kaydırma hareketleri için daha iyi sonuçlar vermektedir.

Kullanılan dört özellik çıkartma algoritması için de gamma parametreleri 0.4, 0.45, 0.5, 0.8, 1, 2, 4, 8, 16, 32, 64 ve 128 değerleri arasından seçilmiştir. ÇYKİ yönteminde özellik sayısı 20-90 arasındaki değerlerden belirlenmiştir.

Tablo 2.16. Farklı özellik çıkartma yöntemlerinin Umdaa veri setindeki EER sonuçları

Yöntemler / Kullanım Yönü		ÇTBA + EKK-DVM	ÇDA + EKK-DVM	ÇYKİ + EKK-DVM	SR-ÇDA + EKK-DVM	SR-ÇDA + AKEYK	[18]
Dikey	Yukarı	7.54	6.84	6.70	6.13	7.82	16.3
	Aşağı	10.25	9.41	9.20	8.76	10.67	17.8
	Sola	9.97	8.11	8.46	8.06	10.11	18.2
	Sağa	7.22	6.67	7.13	6.75	8.61	18.5

Aynı veri setini kullanan çalışmayla kıyaslama yapıldığında sınıflandırmadan önce ön işlemlerin yapılması ve özellik çıkartma yöntemlerinin kullanılması performansı önemli derecede yükseltmiştir. [18] nolu çalışmada elde edilen EER'ler EKK-DVM ve GKM'nin skor füzyonu olmasına rağmen %16.3-18.5 EER arasındadır.

2.3.3.2. Umdaa Veri Setinin ESA ile Sınıflandırılması

Kaydırma hareketlerinin görüntülere aktarılması sonucu elde edilen veri seti içerisinde daha önce MNIST veri setine de uygulanan [49] ESA modelleriyle sınıflandırılmıştır. Uygulanan ESA modellerinin (sırasıyla model 1, model 2, model 3 ve model 4) yapıları aşağıdaki gibidir. BN Batch Normalization'ı ifade ederken, MH ise Maksimum Havuzlamayı ifade etmektedir.

$\text{Evr}(32,(3,3)) \rightarrow \text{ReLU} \rightarrow \text{BN} \rightarrow \text{Evr}(32,(3,3)) \rightarrow \text{ReLU} \rightarrow \text{MH}(2,2) \rightarrow \text{BN} \rightarrow \text{Flatten} \rightarrow \text{BN} \rightarrow \text{Dense}(512) \rightarrow \text{ReLU} \rightarrow \text{BN} \rightarrow \text{Dropout}(0.3) \rightarrow \text{Dense}(\text{Sınıf Sayısı})$

$\text{Evr}(32,(3,3)) \rightarrow \text{ReLU} \rightarrow \text{BN} \rightarrow \text{Evr}(32,(3,3)) \rightarrow \text{ReLU} \rightarrow \text{MH}(2,2) \rightarrow \text{BN} \rightarrow \text{Evr}(64,(3,3)) \rightarrow \text{ReLU} \rightarrow \text{BN} \rightarrow \text{Flatten} \rightarrow \text{BN} \rightarrow \text{Dense}(512) \rightarrow \text{ReLU} \rightarrow \text{BN} \rightarrow \text{Dropout}(0.3) \rightarrow \text{Dense}(\text{Sınıf Sayısı})$

$\text{Evr}(32,(3,3)) \rightarrow \text{ReLU} \rightarrow \text{BN} \rightarrow \text{Evr}(32,(3,3)) \rightarrow \text{ReLU} \rightarrow \text{MH}(2,2) \rightarrow \text{BN} \rightarrow \text{Evr}(64,(3,3)) \rightarrow \text{ReLU} \rightarrow \text{MH}(2,2) \rightarrow \text{Flatten} \rightarrow \text{BN} \rightarrow \text{Dense}(512) \rightarrow \text{ReLU} \rightarrow \text{BN} \rightarrow \text{Dropout}(0.3) \rightarrow \text{Dense}(\text{Sınıf Sayısı})$

$\text{Evr}(32,(3,3)) \rightarrow \text{ReLU} \rightarrow \text{BN} \rightarrow \text{Evr}(32,(3,3)) \rightarrow \text{ReLU} \rightarrow \text{MH}(2,2) \rightarrow \text{BN} \rightarrow \text{Evr}(64,(3,3)) \rightarrow \text{ReLU} \rightarrow \text{MH}(2,2) \rightarrow \text{Flatten} \rightarrow \text{BN} \rightarrow \text{Dense}(512) \rightarrow \text{ReLU} \rightarrow \text{BN} \rightarrow \text{Dropout}(0.3) \rightarrow \text{Dense}(\text{Sınıf Sayısı})$

Aşağıdaki tabloda kullanılan ESA modellerinin performansları gösterilmektedir. En iyi sonuç model 4 ile elde edilmiştir.

Tablo 2.17. Umdaa veri seti üzerinde 4 farklı ESA ile elde edilen EER değerleri

Yöntemler / Kullanım Yönü		Model 1	Model 2	Model 3	Model 4
Dikey	Aşağı	16.89	16.18	16.14	15.72

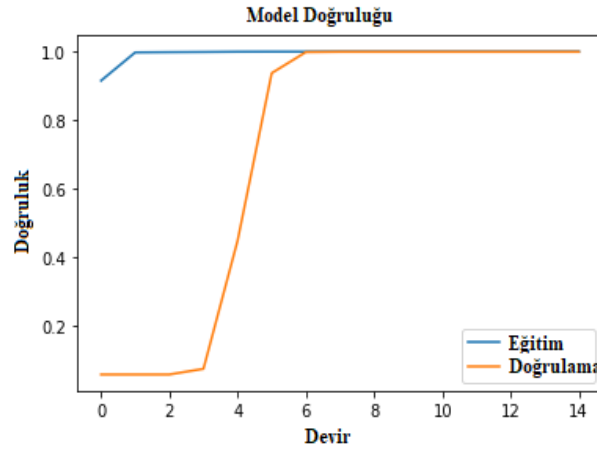
Kullanılan ESA modellerinde optimize edici olarak en iyi sonucu veren Adam kullanılmış olup sonuçlar 15 epoch'ta elde edilmiştir. Son katmanda Softmax aktivasyon fonksiyonu kullanılmıştır. Batch Size için 50, 100, 200 ve 400 değerleri denenmiş, en iyi sonuç 100 için elde edilmiştir.

Aşağıdaki tabloda ESA sınıflandırıcıları ve transfer öğrenmenin performansları gösterilmektedir. Sonuçların bulunduğu ilk sütunda (ESA Tek Kanal 1), kaydırma hareketlerinin uzunluğunun 10 birimden büyük olması sağlanarak uzunluğu 10 birimden küçük olan kaydırma hareketleri değerlendirmeye alınmamıştır. Değerlendirmeye alınmayan kaydırma hareketleri toplam kaydırma hareketlerinin %1.8'ni oluşturmaktadır. En sağdaki üç sütunda ise kaydırma hareketinin değerlendirilmeye alınması için uzunluğunun 20 birimden büyük olması istenmektedir. Uzunluğu 20 birimden düşük kaydırma hareketleri bütün kaydırma hareketlerinin %4.7'sini oluşturmaktadır.

Tablo 2.18. Umdaa veri seti üzerinde ESA ile elde edilen EER değerleri

Yöntemler / Kullanım Yönü		ESA Tek Kanal 1	ESA Tek Kanal 2	ESA 3 Kanal	Transfer Öğrenme
Dikey	Aşağı	16.96	15.72	24.43	26.11

Yukarıdaki tabloda görüldüğü kaydırma hareketi uzunluğunun 10 (ESA Tek Kanal 1) ve 20 (ESA Tek Kanal 2) birimden fazla olması istendiği durumlar karşılaştırıldığında kaydırma hareketinin uzunluğu arttıkça performansın arttığı gözlemlenmektedir. Hız, ivme ve basınç bilgilerine göre RGB piksel parlaklıklarının ayarlandığı 3 kanallı modelde ESA ile sınıflandırma yapıldığında tek kanala göre performansın kötüleştiği ve hız, ivme ve basınç bilgilerinin piksel parlaklığı olarak eklenmesinin performansa katkıda bulunmadığı gözükmemektedir. Aşağıdaki şekilde ESA Tek Kanal 2'nin eğitim ve doğrulama veri seti üzerindeki başarı durumları gösterilmektedir. Şekilde de görüldüğü gibi yaklaşık 2 epoch'dan sonra eğitim başarısı %100'e ulaşırken, doğrulama verisi üzerindeki başarısı ise 4 epoch'tan sonra %100'de sabit kalmaktadır.



Şekil 2.3. Umdaa üzerinde ESA Modelinin devir sayısı artışına göre sağladığı eğitim ve doğrulama veri seti üzerindeki doğruluk oranları

Ayrıca veri setinin yetersiz olması sebebiyle gerçekleştirilecek aşırı öğrenme durumunu da değerlendirmek için VGG16 modeli üzerinde transfer öğrenme işlemi yapılmış ve %26.11 EER değeri elde edilmiştir. Kullanılan ESA modellerinin transfer öğrenme işleminden daha iyi sonuçlar verdiği gözükmemektedir.

3. SONUÇLAR

Bu tez çalışmasında, kaydırma hareketlerinden oluşan üç veri seti üzerinde çeşitli deneyler yapılmıştır. İlk olarak sınıflandırma için kullanılacak özellik sayısı artırılmış ve Serwadda veri seti üzerinde performansa pozitif yönde katkı sağladığı gösterilmiştir. Daha sonra Serwadda veri seti üzerinde sınıflandırmadan önce özellik çıkartma algoritmaları kullanmanın faydası araştırılmış ve performansa pozitif yönde katkıda bulunduğu gözlemlenmiştir.

JSS18 veri seti üzerinde ise kaydırma hareketlerinin bütün olarak değerlendirilmesi yerine 4 gruba (yukarı, aşağı, sol ve sağ) ayrılıp değerlendirilmesinin performansa pozitif yönde katkıda bulunduğu saptanmıştır. Ayrıca Serwada veri setinde olduğu gibi bu veri seti için de özellik çıkartma yöntemi kullanmanın performansa pozitif yönde katkı sağladığı gözlemlenmiştir.

Umdaa veri seti üzerinde diğer iki veri setine göre kullanıcı başı ortalama örnek sayısının fazla olmasından faydalanıp kaydırma hareketleri açığa göre 6 gruba ayrılmış ve başlangıç bitiş noktalarına göre gruplandırma ile karşılaştırılmıştır. Ayrıca diğer veri setlerinde sabit olan eğitimde kullanılacak örnek sayısı bu veri seti için sabit tutulmamıştır. İlk iki veri setinde olduğu gibi bu veri setinde de özellik çıkartma yöntemleri performansı yükselmiştir. Son olarak Umdaa veri setindeki kaydırma hareketleri resim üzerine aktararak ESA sınıflandırılmış ve Umdaa veri setinin EKK-DVM ile sınıflandırıldığı duruma göre daha kötü sonuçlar elde edilmiştir.

4. ÖNERİLER

Bu tez çalışmasında üç veri seti kullanılarak çeşitli sınıflandırma algoritmalarıyla kaydırma hareketleri sınıflandırılmıştır. Sınıflandırmadan önce bazı ön işlemler yapılmış, çeşitli özellik çıkartma algoritmaları kullanılmış ve karşılaştırılmıştır. Bu kapsamda bu çalışmayla ilgili aşağıdaki öneriler dikkate alınabilir.

Zaman içinde kullanıcı kararlılığının değişmesine karşı daha dayanıklı sınıflandırıcılar ve algoritmalar geliştirilebilir.

Sadece kaydırma biyometriği değil, diğer biyometrilere (dokunma, hareket) katılarak multibiyometrik sistemler oluşturularak sistemin güvenilirliği artırılabilir.

ESA ile sınıflandırma yapılırken daha çok bilgi barındıran farklı görüntü oluşturma teknikleri geliştirilebilir.

YSA ve ESA gibi sınıflandırıcıların çok örnekli veri setlerinde daha iyi performans gösterdiği göz önünde bulundurularak kullanıcı başı ortalama örnek sayısı yüksek olan veri setleri oluşturulabilir. Büyük veri setlerinin oluşturulması durumunda eğitim işlemleri uzun süre ve güçlü makineler gerektirebileceğinden dolayı eğitim süreci bulut sunucular üstünde yapılabilir.

Veri setleri uygulama bazlı toplanarak veri setinin kararlılığı artırılabilir. Ayrıca kullanılan el ve parmak bilgisi de veri setine katılarak daha detaylı sınıflandırma yapılabilir.

5. KAYNAKLAR

1. Fierrez, J., Pozo, A., Diaz, M.M., Galbally, J. ve Morales, A., Benchmarking Touchscreen Biometrics for Mobile Authentication, IEEE Transactions on Information Forensics and Security, 13, 11 (2018) 2720-2733.
2. Yang, Y., Guo, B., Wang, Z., Li, M., Yu, Z. ve Zhou, X., BehaveSense: Continuous Authentication for Security-Sensitive Mobile Apps Using Behavioral Biometrics, Ad Hoc Networks, 84, (2019) 9-18.
3. Wu, J.S., Lin, W.C., Lin, C.H. ve Wei, T.E., Smartphone continuous authentication based on keystroke and gesture profiling, International Carnahan Conference on Security Technology (ICCST 2015), Eylül 2015, Taipei, 191-197.
4. Alariki, A.A., Manf, A.B.A. ve Khan, S., A Study of Touching Behavior for Authentication in Touch Screen Smart Devices, International Conference on Intelligent Systems Engineering (ICISE 2016), Islamabad, Ocak 2016, 1-6.
5. Teh, P.S., Zhang, N., Teoh, A.B.J. ve Chen, K., TDAS: a touch dynamics based multi-factor authentication solution for mobile devices, International Journal of Pervasive Computing and Communications, 12, 1(2016) 127-153.
6. Ali, Z., Payton, J. ve Sritapan, V., At Your Fingertips: Considering Finger Distinctness in Continuous Touch-Based Authentication for Mobile Devices, IEEE Security and Privacy Workshops (SPW 2016), San Jose, Mayıs 2016, 272-275.
7. Soni, D., Hanmandlu, M. ve Saini, H.C., A Machine Learning Approach for User Authentication Using Touchstroke Dynamics, First International Conference on Smart System, Innovations and Computing (SSIC 2017), Jaipur, April 2017, 391-410.
8. Alghamdi, S.J. ve Elrefaei, L.A., Dynamic Authentication of Smartphone Users Based on Touchscreen Gestures, Arabian Journal for Science and Engineering, 43, 2(2018) 789-810.
9. Zou, B. ve Li, Y., Touch-based Smartphone Authentication Using Import Vector Domain Description, IEEE 29th International Conference on Application-specific Systems, Architectures and Processors (ASAP 2018), Milan, Temmuz 2018, 1-4.
10. Alpar, O., Biometric touchstroke authentication by fuzzy proximity of touch locations, Future Generation Computer Systems, 86, (2018) 71-80.
11. Meng, W., Wang, Y., Wong, D.S., Wen, S. ve Xiang, Y., TouchWB: Touch behavioral user authentication based on web browsing on smartphones, Journal of Network and Computer Applications, 117, (2018) 1-9.

12. Meng, W., Li, W. ve Wong, D.S., Enhancing touch behavioral authentication via cost-based intelligent mechanism on smartphones, Multimedia Tools and Applications, 77, 23(2018) 30167-30185.
13. Frank, M., Biedert, R., Ma, E., Matinovic, I. ve Song, D., Touchalytics: On the Applicability of Touchscreen Input as a Behavioral Biometric for Continuous Authentication, IEEE Transactions on Information Forensics and Security, 8, 1 (2013) 136-148.
14. Serwadda, A., Phoha, V.V. ve Wang, Z., Which Verifiers Work?: A Benchmark Evaluation of Touch-based Authentication, IEEE Sixth International Conference on Biometrics Theory, Applications and Systemd (BTAS 2013), Eylül-Ekim 2013, Arlington, 1-8.
15. SenthilPrabha, R., Vidhyapriya, R., ve RavithaRajalakshni, N., Performans Analysis for a Touch Dynamic Authentication System with Reduced Feature Set Using Neural Networks, IETE Journal of Research, 62, (2016) 198-204.
16. Shen, C., Zhang, Y., Guan, X. ve Maxion, R.A., Performance Analysis of Touch-Interaction Behavior for Active Smartphone Authentication, IEEE Transactions on Information Forensics and Security, 11, 3 (2016) 498-513.
17. Mahbub, U., Sarkar, S., Patel, V.M. ve Chellappa, R., Active User Authentication for Smartphones: A challenge Data Set Benchmark Results, IEEE Eighth International Conference on Biometrics Theory, Applications and Systems (BTAS 2016) , Eylül 2016, Niagara Falls, 1-8.
18. Syed, Z., Helmick, J., Banerjee, S. ve Cukic, B., Touch Gesture-Based Authentication on Mobile Devices: The Effects of User Posture, Device Size, Configuration, and Inter-Session Variability, The Journal of Systems and Software, 149 (2019) 158-173.
19. Tse, K.W. ve Hung, K., Behavioral Biometrics Scheme with Keystroke and Swipe for User Authentication on Mobile Platform, IEEE 9th Symposium on Computer Applications & Industrial Electronics (ISCAIE 2019), Nisan 2019, Malaysia, 125-130.
20. Ooi, S.Y. ve Teoh, A.B.J., Touch-Stroke Dynamics Authentication Using Temporal Regression Forest, IEEE SIGNAL PROCESSING LETTERS, 26, 7(2019) 1001 - 1005.
21. Syed, Z., Helmick, J., Banerjee, S., Cukic, B., Effect of User Posture, Device Size and Configuration on the Performance of Touch-based Authentication Systems, Journal of Systems and Software.
22. Syed, Z., Helmick, J., Banerjee, S., Cukic, B., Effects of User Posture and Device Size on the Performance of Touch-based Authentication Systems, IEEE 16th International Symposium on High Assurance Systems Engineering (HASE 2015), Ocak 2015, USA 10–17.

23. Antal, M., Bokor, Z. ve Szabo, L.Z., Information Revealed from Scrolling interactions on Mobile Devices, Pattern Recognit. Lett., 56, (2015) 7-13
24. Duda, R. O., Hart, P. E. ve Stork, D. G., Pattern Classification, Second Edition, Wiley Publishing, New York, 2001.
25. Aykut, M., Esnek Kullanımlı Avuç İzi Bölgesine Dayalı Doğrulama Sistemlerinin Tasarım Çalışması, Doktora Tezi, Karedeniz Teknik Üniversitesi, Trabzon, 2013.
26. Welling, M., Max Welling's Class Notes on FLD. www.ics.uci.edu/~welling/classnotes/papers_class/Fisher-LDA.pdf. Son güncelleme tarihi : 1 Aralık 2020.
27. Fisher, R. A., The Use of Multiple Measurements in Taxonomic Problems, Annals of Eugenics, 7 (1936) 179-188.
28. Schölkopf, B. ve Smola, A. J., Learning with Kernels, The MIT Press, London, 2002.
29. The Courant Institute of Mathematical Sciences New York University. <http://yann.lecun.com/exdb/mnist/>. Son güncelleme tarihi : 12.12.2020.
30. Cai, D., He, X. ve Han, J., SRDA: An efficient algorithm for large-scale discriminant analysis, IEEE Transactions On Knowledge And Data Engineering, 20, 1(2008) 1-12.
31. He, X., Yan, S., Hu, Y. ve Zhang H. J., Learning a Locality Preserving Subspace for Visual Recognition, Proceedings 9th IEEE International Conference on Computer Vision (ICCV 2003), Ekim 2003, France, 385-392.
32. Aizerman, M., Braverman, E. ve Rozonoer, L., Theoretical Foundations of The Potential Function Method in Pattern Recognition Learning, Automation and Remote Control, 25 (1964) 821-837.
33. Support Vector Machines, web: <https://www.learnopencv.com/support-vector-machines-svm/>. Son güncelleme tarihi: 15.04.2020
34. Schölkopf, B., Smola, A. ve Müller, K., Nonlinear Component Analysis as a Kernel Eigenvalue Problem, Neural Comput., 10,5 (1998) 1299-1319.
35. Mika, S., Ratsch, G., Weston, J., Scholkopf, B. ve Muller, K., Fisher Discriminant Analysis with Kernels, IEEE Neural Networks for Signal Processing IX, Ağustos 1999, Madison, Bildiriler Kitabı, 41-48.

36. Cai, D., He, X. ve Han, J., Efficient kernel discriminant analysis via spectral regression, Proceedings of the Seventh Ieee International Conference On Data Mining (ICDM 2007), Omaha, Ekim 2007, 427-432.
37. Cheng, J., Liu, O., Lu, H. ve Chen, Y. W., Supervised Kernel Locality Preserving Projections for Face Recognition, Neurocomputing, 67 (2005), 443-449
38. Vapnik, V., The Nature of Statistical Learning Theory, Springer-Verlag, New York, 1995.
39. Suykens, J. ve Vandewalle, J., Least Squares Support Vector Machines Classifiers, Neural Processing Letters, 9, 3 (1999) 293-300.
40. Gaussian Mixture Models, web: <https://bengio.abracadoudou.com/lectures/gmm.pdf>. Son güncelleme tarihi: 21.04.2020
41. Gaussian Mixture Models, web: <https://towardsdatascience.com/gaussian-mixture-models-explained-6986aaf5a95>. Son güncelleme tarihi: 21.04.2020
42. K - Nearest Neighbors, web: <https://towardsdatascience.com/knn-k-nearest-neighbors-1-a4707b24bd1d>. Son güncelleme tarihi : 20.08.2020.
43. Weighted K-Nearest Neighbors, web: <https://www.geeksforgeeks.org/weighted-k-nn/>. Son güncelleme tarihi : 20.08.2020.
44. McCulloch, W., S. ve Pitts, W., A Logical Calculus of Ideas Immanent in Nervous Activity, The Bulletin of Mathematical Biophysics, 5, 4 (1943) 115-133.
45. Hebb, D., O., The Organization of Behavior, Journal of Clinical Psychology, 6, 3 (1950) 307.
46. Rosenblatt, F., The Perceptron: A Probabilistic Model for Information Storage and Organization In The Brain, Psychological Review, 65, 6 (1958) 386-408.
47. Rumelhart, D., E. ve McClelland, J., L, Parallel Distributed Processing, 1, 1986.
48. Introduction to Artificial Neural Networks, web: <https://towardsdatascience.com/introduction-to-artificial-neural-networks-ann-1aea15775ef9>. Son güncelleme tarihi: 21.04.2020
49. Introduction to Machine Learning & Deep Learning in Python, web: <https://globalsoftwaresupport.teachable.com/>. Son güncelleme tarihi: 21.04.2020
50. Convolution Neural Networks, web: <https://medium.com/@tuncerergin/convolutional-neural-network-convnet-yada-cnn-nedir-nasil-calisir-97a0f5d34cad>. Son güncelleme tarihi: 21.04.2020

51. Transfer Learning, web: <https://towardsdatascience.com/a-comprehensive-hands-on-guide-to-transfer-learning-with-real-world-applications-in-deep-learning-212bf3b2f27a>. Son güncelleme tarihi: 21.04.2020
52. Diaz, M.M., Fierrez, J., Krish, R.P. ve Galbally, J., Mobile signature verification: feature robustness and performance comparison, IET Biometrics, 3, 4 (2014), 267–277.
53. Pudil, P., Novovicova, J. ve Kittler, J., Floating search methods in feature selection, Pattern Recognition Letters, 15, 11 (1994), 1119-1125.

ÖZGEÇMİŞ

Orhan SİVAZ 1991 yılında Trabzon'un Çaykara ilçesinde doğdu. İlköğretim ve liseyi Trabzon'da tamamlamış olup, 2014 yılında Karadeniz Teknik Üniversitesi Bilgisayar Mühendisliği Bölümünden mezun olmuştur. 2017 yılında Karadeniz Teknik Üniversitesi Bilgisayar Mühendisliği Bölümünde Bilgisayar Bilimleri Anabilim dalında yüksek lisans eğitimine başlamıştır. 2018 yılı itibariyle Karadeniz Teknik Üniversitesi Bilgisayar Bilimleri Anabilim Dalında Araştırma Görevlisi olarak çalışmaktadır.