

KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

İŞARET DİLİ KARAKTERLERİNİN KARMAŞIK DEĞERLİ SİNİR
AĞLARI İLE TANINMASI

DOKTORA TEZİ

Selda BAYRAK

KASIM 2024

TRABZON



**KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

**İŞARET DİLİ KARAKTERLERİNİN KARMAŞIK DEĞERLİ SINIR AĞLARI
İLE TANINMASI**

Selda BAYRAK

**Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsünce
"DOKTOR (BİLGİSAYAR MÜHENDİSLİĞİ)"
Unvanı Verilmesi İçin Kabul Edilen Tezdir.**

Tezin Enstitüye Verildiği Tarih : 20 / 09 / 2024

Tezin Savunma Tarihi : 07 / 11 / 2024

Tez Danışmanı : Doç. Dr. Vasif NABİYEV

Trabzon 2024

ÖNSÖZ

İşaret dili, sağır ve dilsiz bireylerin hem birbirleriyle hem de diğer insanlarla etkili bir şekilde iletişim kurmalarına olanak tanıyan bir yöntemdir. Bu çalışmada, yeni bir yöntem kullanılarak işaret dili tanıma üzerine bir araştırma yapılmıştır. İşaret dili görüntülerinde hedeflenen tanıma oranına ulaşılmış ve gerçek zamanlı bir tanıma sistemi geliştirilmiştir.

Doktora çalışması sürecinde bilgi, rehberlik ve destekleriyle her zaman yanımda olan değerli danışman hocam Doç. Dr. Vasif NABİYEV'e içten teşekkürlerimi sunarım. Ayrıca, görüş ve önerileriyle bu çalışmaya katkıda bulunan değerli jüri üyeleri Prof. Dr. Mustafa ULUTAŞ ve Prof. Dr. Ali GANGAL'a da minnettarım.

Bu çalışmayı gerçekleştirdiğim süreç boyunca her konuda yardımını esirgemeyen sevgili aileme, eşime ve manevi desteğiyle bana daima güç ve ilham veren kızıma en derin şükranlarımı sunarım.

Bu çalışmanın ilgili alanlara katkı sunmasını temenni ederim...

Selda BAYRAK

Trabzon 2024

TEZ ETİK BEYANNAMESİ

Doktora Tezi olarak sunduğum " İşaret Dili Karakterlerinin Karmaşık Değerli Sinir Ağları İle Tanınması" başlıklı bu çalışmayı baştan sona kadar danışmanım Doç. Dr. Vasif NABİYEV'in sorumluluğunda tamamladığımı, verileri/örnekleri kendim topladığımı, deneyleri/analizleri ilgili laboratuvarlarda yaptığımı/yaptırdığımı, 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. 07/11/2024

Selda BAYRAK

İÇİNDEKİLER

	<u>Sayfa No</u>
ÖNSÖZ.....	III
TEZ ETİK BEYANNAMESİ.....	IV
İÇİNDEKİLER.....	V
ÖZET	VII
SUMMARY	VIII
ŞEKİLLER DİZİNİ	IX
TABLolar DİZİNİ.....	XII
SEMBOLLER VE KISALTMALAR DİZİNİ	XIV
1. GENEL BİLGİLER	1
1.1. Giriş.....	1
1.2. İşaret Dili.....	2
1.3. İşaret Dili Tanıma Sistemlerinde Kullanılan Yöntemler.....	5
1.4. Tezin Kapsamı ve Amacı	6
1.5. Literatürde İşaret Dili Tanıma Üzerine Gerçekleştirilen Çalışmalar	6
1.6. Tez Kapsamında Kullanılacak Veri Setleri	12
1.6.1. MNIST Veri Seti	12
1.6.2. MUD Veri Seti	13
1.6.3. Model Doğrulama ve Değerlendirme Teknikleri.....	13
1.7. MNIST Veri Seti Üzerinde Gerçekleştirilen Literatür Çalışmaları.....	14
1.8. MUD Veri Seti Üzerinde Gerçekleştirilen Literatür Çalışmaları.....	17
1.9. Özellik Çıkarımı	20
1.9.1. Momentler	21
1.9.2. Uygulamalı Bilimlerde Moment Kavramı	23
1.9.3. Yapay Zeka ve Görüntü İşleme Alanında Moment Analizi	26
1.9.4. Görüntü Momentlerinin Tanımı.....	27
1.9.5. Moment Türleri	27
1.9.6. Zernike Momentleri	31
1.9.7. Zernike Momentlerinin Hesaplanması.....	32
1.9.8. Karmaşık Sayılar	33
1.9.9. Karmaşık Zernike Momentleri.....	44

1.10. Sınıflandırma	50
1.10.1. Makine Öğrenmesi	51
1.10.2. Derin Öğrenme ve Mimarileri	53
1.10.3. Derin Sinir Ağları	57
1.11. Karmaşık Değerli Derin Sinir Ağları	58
1.11.1. Optimizasyon Algoritmaları	59
1.11.2. CVDNN Eğitim Algoritması	62
1.11.3. Derin Öğrenme Modelinde Hiperparametreler	67
1.11.4. Karmaşık Değerli Aktivasyon Fonksiyonları	69
1.11.5. Seyreltme Oranı	72
1.11.6. Softmax Çıkış Fonksiyonu	73
1.11.7. Karmaşık Değerli XOR Problemi	74
2. YAPILAN ÇALIŞMALAR	76
2.1. Karmaşık Zernike Momentleri Kullanılarak Özellik Çıkarımı	76
2.2. Öznitelik Vektöründen Görüntünün Yeniden Oluşturulması	78
2.3. CVDNN Mimarisi	80
2.4. Model Mimarisinin Detayları	82
2.5. Karmaşık Veriler İçeren XOR Probleminin Çözümü	85
2.6. Performans Analiz Metrikleri	86
2.7. Önerilen Modelin MNIST Veri Seti Üzerindeki Performansının Değerlendirilmesi	88
2.8. Önerilen Modelin MUD Veri Seti Üzerindeki Performansının Değerlendirilmesi	89
2.9. Önerilen Modelin Kullanıldığı Akıllı Sistemin Geliştirilmesi	100
3. BULGULAR VE TARTIŞMA	105
3.1. Model Mimarisi ve Akıllı Sistem Üzerindeki Ablasyon Çalışmaları	105
3.2. Geliştirilen Modelin Diğer Çalışmalarla Karşılaştırılması	106
4. SONUÇLAR	111
5. ÖNERİLER	114
6. KAYNAKLAR	116
ÖZGEÇMİŞ	

Doktora Tezi

ÖZET

İŞARET DİLİ KARAKTERLERİNİN KARMAŞIK DEĞERLİ SINIR AĞLARI İLE TANINMASI

Selda BAYRAK

Karadeniz Teknik Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı
Danışman: Doç. Dr. Vasif NABİYEV
2024, 121 Sayfa

Teknolojik gelişmeler, sağır ve dilsiz bireylerin topluma entegrasyonunda önemli bir rol oynamaktadır. Bu nedenle, işaret dili tanıma sistemlerindeki iyileştirmeler büyük önem taşımaktadır. İşaret dilleri üzerine yapılan birçok çalışma gerçek sayılar kullanılarak gerçekleştirilmiştir. Bu çalışmada, görüntülerden öznitelik çıkarımı ve işaret dili alfabesi tanıma işlemlerini karmaşık sayılar kullanarak gerçekleştiren yeni bir yaklaşım sunulmaktadır. Bu bağlamda Amerikan işaret dili tanımaya yönelik bir model geliştirilmiştir. Geliştirilen modelde, karakter görüntülerinin öznitelik vektörünü elde etmek için karmaşık Zernike momentleri kullanılmıştır. Karmaşık sayılardan oluşan öznitelik vektörünü katmanlar arasında işleyebilen karmaşık değerli bir derin sinir ağı da geliştirilmiştir. Model, Sign Language MNIST veri setinde %89.01, Massey Üniversitesi veri setinde ise holdout tekniği ile %98.67, bir deneği dışarda bırakma tekniği ile %81.22 tanıma oranlarına veri ön işleme olmadan ulaşmıştır. Önerilen model aynı veri setlerini kullanan birçok çalışma ile ayrı ayrı karşılaştırılmış, iki veri seti bir arada değerlendirildiğinde en iyi performansı göstermiştir. Ayrıca önerilen yöntemi bünyesinde barındıran bir akıllı sistem de geliştirilmiştir.

Anahtar Kelimeler: Karmaşık değerli derin sinir ağı, Karmaşık zernike momentleri, Öznitelik çıkarımı, İşaret dili tanıma modeli

PhD. Thesis

SUMMARY

RECOGNITION OF SIGN LANGUAGE CHARACTERS USING COMPLEX-VALUED NEURAL NETWORKS

Selda BAYRAK

Karadeniz Technical University
The Graduate School of Natural and Applied Sciences
Computer Engineering Graduate Program
Supervisor: Assoc. Prof. Vasif NABIYEV
2024, 121 Pages

Technological advancements play a significant role in the integration of deaf and mute individuals into society. Therefore, further improvements in sign language recognition systems are of great importance. Many studies on sign languages have been conducted using real numbers. In this paper, a new approach is presented for performing feature extraction from images and sign language alphabet recognition using complex numbers. In this context, a model is developed for recognizing American sign language. In the developed model, complex Zernike moments are used to obtain the feature vector of character images. A complex valued deep neural network capable of processing the feature vector composed of complex numbers across layers is also developed. The model achieves recognition rates of 89.01% on the Sign Language MNIST dataset and 98.67% for the holdout and 81.22% for the leave-one-subject-out on the Massey University dataset, respectively, without any preprocessing. The proposed model, which is compared separately with many studies using the same datasets, shows the best performance when the two datasets are considered together. An intelligent system incorporating the proposed method has also been developed.

Key Words: Complex valued deep neural network, Complex zernike moments, Feature extraction, Sign language recognition model

ŞEKİLLER DİZİNİ

Sayfa No

Şekil 1.	Amerikan işaret dilinde 'üzgünüm' ifadesinin mimik ve el hareketiyle anlatılması.....	2
Şekil 2.	Amerikan işaret dilindeki harfleri ifade etmek için kullanılan MNIST veri seti görüntüleri.....	3
Şekil 3.	Amerikan işaret dilindeki harfleri ve rakamları ifade etmek için kullanılan MUD veri seti görüntüleri	4
Şekil 4.	Genel olarak işaret dillerinde kullanılan hareketlerin hiyerarşik akışı	7
Şekil 5.	Karami ve arkadaşları tarafından geliştirilen modelin mimarisi	8
Şekil 6.	Rao ve ekibi tarafından geliştirilen modelin mimarisi	9
Şekil 7.	Sharma ve ekibi tarafından geliştirilen modelin mimarisi.....	10
Şekil 8.	Rastgoo ve ekibi tarafından geliştirilen modelin mimarisi.....	11
Şekil 9.	MNIST eğitim veri seti içindeki her bir sınıfa ait görüntü sayısı	12
Şekil 10.	MNIST test veri seti içindeki her bir sınıfa ait görüntü sayısı.....	13
Şekil 11.	Fizikte kuvvetlerin moment etkisi	22
Şekil 12.	$f(x) = e^{-x^2}$ fonksiyonu ve ilk dört momentinin hesaplanmış değerleri.....	24
Şekil 13.	Bir olasılık dağılım fonksiyonu ve ilk dört momentinin değerleri	25
Şekil 14.	Sayı Kümeleri	35
Şekil 15.	(a) $f(z)$ fonksiyonunun holofonik bir örneği (b) $f(z)$ fonksiyonunun holofonik olmayan bir örneği	38
Şekil 16.	(a) Euler formülünün birim çember üzerindeki temsili (b) Euler formülünün, trigonometrik gösterimi	40
Şekil 17.	Karmaşık düzlemde verilen beş karmaşık sayı (b) İlgili beş karmaşık sayının $f(z) = z^2$ fonksiyonuna göre dönüşümünü.....	41
Şekil 18.	(a) Ayrık noktalardan oluşan karesel bir görüntü (b) İlgili görüntünün $f(z) = z^2$ fonksiyonuna göre dönüşümünü	41
Şekil 19.	(a) Renklendirilmiş karmaşık düzlemde verilen dört karmaşık sayı (b) İlgili karmaşık sayıların ifade edilen düzlemde $f(z) = z^2$ fonksiyonuna göre dönüşümünü.....	42
Şekil 20.	(a) Renklendirilmiş karmaşık bir düzlem (b) Belirtilen düzlemin $f(z) = e^{(\sin(z))^2}$ fonksiyonuna göre dönüşümünü	43
Şekil 21.	(a) Renklendirilmiş karmaşık bir düzlemin $f(z) = z^2$ fonksiyonuna göre dönüşümü (b) İlgili düzlemin $f(z) = z^2$ fonksiyonuna göre dönüşümünün 3D olarak gösterilmesi	43
Şekil 22.	Zernike radyal polinomlarının ($m = 0, 1$) için ilk beş derecesi	45

Şekil 23.	Bağıntı (33)'deki $e^{-jm\theta}$ açısai bileşenin m = 2 için sinüs ve kosinüs grafikleri	46
Şekil 24.	$(V_{nm}(r, \theta))$ Zernike polinomunun farklı n ve m değeri için hesaplanmış halinin kutupsal formda gerçek ve sanal bileşenlerini renk kodlaması	46
Şekil 25.	(a) $V_{nm}(r, \theta)$ Zernike polinomunun (n = 2, m = 2 için) kutupsal olarak renklendirilmiş formu (b) Aynı noktaların ayrı bir görüntüde grid yapısında nasıl konumlandığının gösterilmesi.....	47
Şekil 26.	Farklı derecelerdeki Zernike polinomları	47
Şekil 27.	(a) NxN boyutlu ayrı bir görüntü (b) Birim disk (c) Birim diskin normalize edilmiş görüntü içine iz düşürülmüş hali.....	48
Şekil 28.	Örnek bir fotoğraf izdüşümü	48
Şekil 29.	Zernike polinomlarının eşleniğinin iz düşürülmüş görüntüye uygulanması.....	49
Şekil 30.	Karmaşık öznitelik vektörü grafiği.....	49
Şekil 31.	Öznitelik vektörünün genlik grafiği.....	50
Şekil 32.	Derin öğrenmenin, yapay zekâ ve makine öğrenmesiyle olan ilişkisi.....	51
Şekil 33.	CNN'lerin genel yapısı	54
Şekil 34.	RNN, LSTM ve GRU'nun genel yapısı	55
Şekil 35.	RBM'nin genel yapısı.....	55
Şekil 36.	GAN'ın genel yapısı.....	56
Şekil 37.	Oto-kodlayıcıların genel yapısı.....	57
Şekil 38.	Karmaşık değeri derin sinir ağı (CVDNN).....	63
Şekil 39.	Aktivasyon fonksiyonları.....	71
Şekil 40.	Yetersiz öğrenme, aşırı öğrenme ve ideal öğrenmeyi temsil eden grafikler.....	72
Şekil 41.	Seyreltme tekniğine bağıli olarak devre dışı bırakılan nöronlar ve ağı yapısı.....	73
Şekil 42.	Özellik çıkarımı için işlem adımları	77
Şekil 43.	(a) 200x200 piksel orijinal giriş görüntüsü (b) orijinal görüntü için farklı derecelerde yeniden oluşturulan görüntüler.....	79
Şekil 44.	(a) MNIST veri setinden 28x28 piksel orijinal giriş görüntüsü (b) orijinal görüntü için farklı derecelerde yeniden oluşturulan görüntüler.....	79
Şekil 45.	Geliştirilen modelin mimarisi	82
Şekil 46.	(a) MNIST veri seti için ilk iterasyon ve alt küme iterasyonları (b) MNIST veri seti için son ana iterasyon ve alt küme iterasyonları.....	83
Şekil 47.	MUD veri seti için ilk iterasyon ve alt küme iterasyonları (b) MUD veri seti için son ana iterasyon ve alt küme iterasyonları	84

Şekil 48.	(a) MNIST veri seti için hata eğrisi-iterasyon grafiği (b) MUD veri seti için hata eğrisi-iterasyon grafiği	84
Şekil 49.	Karmaşık sayılar kullanan XOR probleminin çözümü.....	86
Şekil 50.	Önerilen modelin MNIST test veri seti için oluşturduğu karmaşıklık matrisi	86
Şekil 51.	Önerilen modelin Holdout tekniğine göre MUD test veri seti için oluşturduğu karmaşıklık matrisi	90
Şekil 52.	Önerilen modelin LOSO tekniğine göre MUD veri setinde Denek 1 için oluşturduğu karmaşıklık matrisi	92
Şekil 53.	Önerilen modelin LOSO tekniğine göre MUD veri setinde Denek 2 için oluşturduğu karmaşıklık matrisi	94
Şekil 54.	Önerilen modelin LOSO tekniğine göre MUD veri setinde Denek 3 için oluşturduğu karmaşıklık matrisi	94
Şekil 55.	Önerilen modelin LOSO tekniğine göre MUD veri setinde Denek 4 için oluşturduğu karmaşıklık matrisi	97
Şekil 56.	Önerilen modelin LOSO tekniğine göre MUD veri setinde Denek 5 için oluşturduğu karmaşıklık matrisi	97
Şekil 57.	Sağ el için Mediapipe tarafından üretilen anahtar noktalar ve açıklamaları.....	100
Şekil 58.	Gerçek zamanlı sistemden bir kesit görüntüsü	102
Şekil 59.	ASL veri setinin %70 eğitim %30 test verisi olarak ayrıldığı durumda önerilen model tarafından elde edilen hata analiz sonuçları.....	109
Şekil 60.	ASL veri setinde birbirine oldukça benzeyen çıkış sınıfları.....	109

TABLolar DİZİNİ

	<u>Sayfa No</u>
Tablo 1. Li ve arkadaşları tarafından MNIST veri seti üzerinde gerçekleştirilen çalışmalarının sonuçları	15
Tablo 2. Bhushan ve arkadaşları tarafından MNIST veri seti üzerinde gerçekleştirilen çalışmalarının sonuçları	16
Tablo 3. MNIST veri seti üzerinde gerçekleştirilen akademik çalışmaların test verisi üzerindeki tanıma oranları	17
Tablo 4. MUD veri seti üzerinde gerçekleştirilen akademik çalışmaların Holdout ve LOSO tekniklere göre tanıma oranları	19
Tablo 5. Belirtilen kurallara göre karmaşık değerli XOR için eğitim örnekleri	74
Tablo 6. XOR problemi için giriş ve çıkışlar	85
Tablo 7. Karmaşıklık matrisinin genel temsili	87
Tablo 8. Önerilen modelin MNIST test veri setinin her bir sınıfı için ürettiği performans metrik değerleri	89
Tablo 9. Holdout tekniğine göre MUD test verisindeki her bir sınıf için önerilen model tarafından üretilen performans metriği değerleri	91
Tablo 10. Önerilen model tarafından LOSO tekniği kullanılarak MUD veri setindeki tüm denekler için hesaplanan ortalama değerler ile birlikte, her bir denek için görüntü sayısı, doğru tahmin sayısı, ortalama tanıma oranı, ortalama kesinlik, ortalama geri çağırma ve ortalama F1 puanı	92
Tablo 11. LOSO tekniği kullanılarak MUD veri setindeki Denek 1'e göre her bir sınıf için önerilen model tarafından üretilen performans metrik değerleri	93
Tablo 12. LOSO tekniği kullanılarak MUD veri setindeki Denek 2'ye göre her bir sınıf için önerilen model tarafından üretilen performans metrik değerleri	95
Tablo 13. LOSO tekniği kullanılarak MUD veri setindeki Denek 3'e göre her bir sınıf için önerilen model tarafından üretilen performans metrik değerleri	96
Tablo 14. LOSO tekniği kullanılarak MUD veri setindeki Denek 4'e göre her bir sınıf için önerilen model tarafından üretilen performans metrik değerleri	98
Tablo 15. LOSO tekniği kullanılarak MUD veri setindeki Denek 5'e göre her bir sınıf için önerilen model tarafından üretilen performans metrik değerleri	99
Tablo 16. Web kamerasından alınan görüntü akışında elin kare içine alınmasını sağlayan kod parçacığı	102

Tablo 17.	Önerilen model ile diğer çalışmaların MNIST test veri setindeki tanıma oranları	107
Tablo 18.	Holdout tekniğine göre önerilen model ve diğer çalışmaların MUD test veri setindeki tanıma oranları	107
Tablo 19.	LOSO tekniğine göre önerilen model ve diğer çalışmaların MUD test veri setindeki tanıma oranları	108



SEMBOLLER VE KISALTMALAR DİZİNİ

ASL	: Amerikan işaret dili
MUD	: Massey Üniversitesi veri seti
CNN	: Evrişimli sinir ağları
DNN	: Derin sinir ağları
RBM	: Kısıtlı Boltzmann makinesi
LSTM	: Uzun kısa süreli bellek
RNN	: Tekrarlayan sinir ağları
3DCNN	: 3D evrişimli sinir ağı
SVM	: Destek vektör makineleri
HMM	: Gizli Markov modeli
RF	: Rastgele orman
XGBoost	: Ekstrem gradyan artırma
DT	: Karar ağaçları
KNN	: K-en yakın komşu
SGD	: Stokastik gradyan inişi
NBG	: Naive Bayes Gaussian
ANN	: Yapay sinir ağları
CZM	: Karmaşık Zernike momentleri
CVDNN	: Karmaşık değerli derin sinir ağı
MLP	: Çok katmanlı algılayıcı
LOSO	: Bir deneği dışarda bırakma
RFC	: Rastgele orman sınıflandırıcısı
SGDC	: Stokastik gradyan iniş sınıflandırıcısı
HOG	: Yönlü gradyanların histogramı
LBP	: Yerel ikili desenler
SIFT	: Ölçek-bağımsız özellik dönüşümü
GAN	: Üretken karşıt ağlar
GRU	: Kapı özyinelemeli geçitler
CVNN	: Kompleks değerli sinir ağı
MSE	: Ortalama Kare Hatası

SSIM : Yapısal Benzerlik İndeksi
TP : Doğru Pozitif
TN : Doğru Negatif
FP : Yanlış Pozitif
FN : Yanlış Negatif



1. GENEL BİLGİLER

1.1. Giriş

Sağır ve dilsiz bireyler, toplumsal yaşamın önemli bir kesimini oluşturmaktadır. Dünya genelinde milyonlarca insan işitme kaybı yaşamakta, yalnızca işitme yetilerini kaybetmekle kalmayıp aynı zamanda iletişim ve toplumsal katılım açısından da önemli zorluklarla karşılaşmaktadır. Dünya Sağlık Örgütü'nün verilerine göre dünya nüfusunun yaklaşık %5'i (430 milyon kişi) işitme kaybına sahiptir ve bu sayının 2050 yılına kadar iki katına çıkacağı öngörülmektedir (URL1, 2024). Bu bireylerin büyük bir kısmı ciddi işitme kaybı yaşamaktadır ve tamamen sağır olanlar da bu gruba dahildir. İşitme kaybı, çoğunlukla dilsizlikle ilişkilendirilir ve bu durum konuşma engelini doğurabilir. Dilsizlik, bazen doğuştan gelen ya da nörolojik sorunlar gibi farklı faktörlerden kaynaklanabilir. Ancak, işitme kaybı yaşayan bireylerin büyük bir kısmının aynı zamanda konuşma engeli yaşadığı gözlemlenmektedir. Bu durum, yalnızca sağlık hizmetleri açısından değil, toplumsal katılım, eğitim, istihdam gibi birçok alanda da ciddi bir sorunu beraberinde getirmektedir. İşitme kaybı ve konuşma engeli olan bireylerin yaşam kalitesini artırmak, toplumun bu bireylerle daha iyi bir iletişim ve etkileşim kurmasını sağlamak için daha fazla toplumsal farkındalığa ve kapsayıcı politikalara ihtiyaç vardır. Bu bireylerin ihtiyaçları karşılanamadığında, toplumdaki dışlanma ve yalnızlaşma gibi ciddi sosyal sorunlarla karşı karşıya kalma riskleri de artmaktadır.

Toplumsal farkındalığın artırılması, sağır ve dilsiz bireylerin toplumsal yaşama tam anlamıyla katılabilmelerinin sağlanması açısından kritik bir öneme sahiptir. Eğitim ve iş olanaklarının sağlanması, bu bireylerin kendi potansiyellerini gerçekleştirmelerine ve bağımsız bir yaşam sürdürebilmelerine olanak tanıyacaktır. Sağır ve dilsiz bireylerin yaşadığı zorlukların giderilmesi, toplumsal eşitlik ve adaletin sağlanması açısından da büyük önem taşımaktadır.

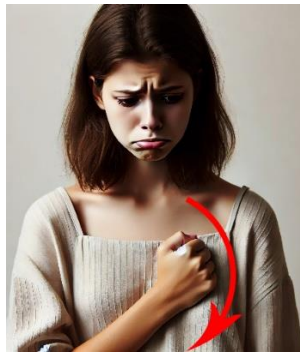
Dünya Sağlık Örgütü'nün (URL1, 2024) verileri, işitme kaybı ve dilsizlik konusunun giderek daha büyük bir toplumsal sorun haline geleceğini göstermektedir. Bu durum, yalnızca sağlık hizmetleri ile sınırlı kalmamalı, aynı zamanda toplumsal bilinçlenme ve kapsayıcı politikalarla desteklenmelidir. Ayrıca, işaret dili gibi alternatif iletişim araçlarının yaygınlaştırılması ve bu dilin herkes tarafından öğrenilmesi teşvik edilmelidir. İşaret dili,

işitme kaybı ve konuşma engeli olan bireyler için en temel iletişim aracıdır ve bu dili bilen bireyler aracılığıyla daha kapsayıcı bir toplum oluşturulabilir. İşaret dilinin herkes tarafından öğrenilmesi teşvik edilse bile, bunun istenilen başarıyı tam anlamıyla sağlayamayacağı açıktır. Bu doğrultuda, sağır ve dilsiz bireylerle toplum arasında köprü kuracak alternatif araçların veya uygulamaların geliştirilmesi büyük fayda sağlayacak ve istenen başarıya ulaşmayı mümkün kılacaktır.

1.2. İşaret Dili

İşaret dili, genellikle işitme engelli bireylerin kendi aralarında ya da tercümanlar aracılığıyla başkalarıyla iletişim kurmak için kullandığı bir yöntemdir. Ancak, tercüman sayısının sınırlı olması ve işaret dilinin konuşma dilinden bağımsız bir yapıya sahip olması nedeniyle, işitme engelli bireyler çevreleriyle her zaman etkili bir şekilde iletişim kurma fırsatına sahip olamayabilirler. Bu sebeple, işaret dili tanıma sistemlerinin ve teknolojilerinin geliştirilmesi, sağır ve dilsiz bireylerin daha bağımsız ve etkili iletişim kurabilmeleri açısından büyük bir önem taşımaktadır.

Sağır bireyler, kelimeleri ve harfleri ifade etmek için parmak alfabesi ve jestler gibi işaret dili yöntemlerini kullanırlar. İşaret dilinin temel bileşenleri arasında parmak alfabesi ve çeşitli jestler önemli bir rol oynar. Parmak alfabesi, her bir harf ve rakamı belirli bir işaretle temsil ederken, jestler ve akışa dayalı el işaretleri kelimeleri ifade etmek için kullanılır. Ayrıca, yüz ifadeleri ve baş hareketleri de işaret dilinin önemli unsurlarıdır. Bu unsurlar, iletişimi güçlendirir ve işaret dilindeki ifadelerin daha anlamlı ve anlaşılır olmasını sağlar. Şekil 1, el ve mimik hareketlerinden oluşan bir ifadeyi göstermektedir.



Şekil 1. Amerikan işaret dilinde 'üzgünüm' ifadesinin mimik ve el hareketiyle anlatılması

Dünyanın birçok bölgesinin kendine özgü işaret dili bulunmaktadır. Her ülkenin ya da bölgenin işaret dili, yerel kültür ve dil yapısına bağlı olarak farklılık gösterebilir. Türk işaret dili, Amerikan işaret dili, İngiliz işaret dili, Alman işaret dili, İtalyan işaret dili, Japon işaret dili, Fransız işaret dili, İspanyol işaret dili, Hint işaret dili, Rus işaret dili ve Çek işaret dili bunlardan bazılarıdır. Örneğin, Amerikan işaret dili (ASL), Amerika Birleşik Devletleri ve Kanada'da yaygın olarak kullanılan bir işaret dilidir. ASL, 1817'de Thomas Gallaudet tarafından kurulan Amerikan sağır okulunda bir dil olarak ortaya çıkmıştır. ASL, eski Fransız işaret dili, çeşitli köy işaret dilleri ve ev işaret sistemlerinin etkileşimi sonucunda oluşmuştur.

İşaret dillerinin kendine özgü dil yapıları ve gramer kuralları vardır. ASL, bu bağlamda bir dil olarak, sözdizimi (dil bilgisi), sözcük dağarcığı ve anlamlı işaretlerin birleşimi ile ifade edilir. Ayrıca, ASL'nin İngilizce'den farklı bir gramer yapısı ve sözcük sıralaması vardır. Örneğin, ASL'de cümle yapısı genellikle nesne-yüklem-özne şeklindedir, bu da İngilizce'nin özne-yüklem-nesne yapısından farklıdır.

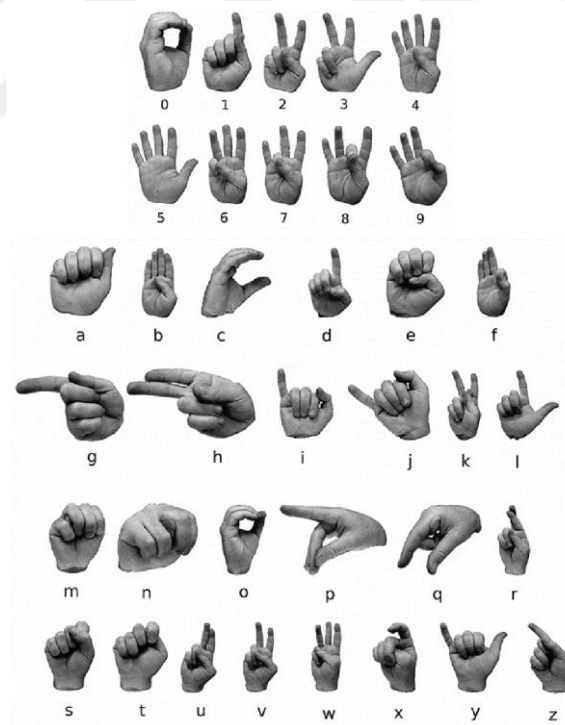
Şekil 2, ASL için kullanılan parmak alfabesini gösteren Sign Language MNIST veri setini temsil eder (Tecperson, 2023). Bu veri seti, ASL alfabesindeki her bir harfi temsil eden işaretlerin tanımlanmasını sağlar. İlgili veri seti, hem eğitim hem de test verisi açısından çok sayıda görüntü içerdiğinden önemli bir kaynaktır.



Şekil 2. Amerikan işaret dilindeki harfleri ifade etmek için kullanılan MNIST veri seti görüntüleri (Tecperson, 2023)

Massey Üniversitesi tarafından Amerikan işaret dili alfabesini ve rakamlarını içeren bir diğer veri seti, Massey Üniversitesi veri seti (MUD) geliştirilmiştir (Barzack vd., 2011). MUD, kişisel farklılıkları da dikkate alan bir veri setidir. Şekil 3, ASL'deki harflerin ve rakamların tanımlanmalarını sağlayan işaretlerin yer aldığı MUD veri setini göstermektedir (Barzack vd., 2011). İlgili veri seti Kaggle web sitesi üzerinden indirilebilir (Thakur, 2023). Kaggle web sitesinde, MUD veri setinin çeşitli ön işlemlerden geçirilerek (gürültü eklenerek, döndürülerek ve kaydırılarak) elde edilen varyasyonlarını içeren başka veri setleri de bulunmaktadır (Dima, 2024).

MNIST ve kişisel farklılıkları dikkate alarak oluşturulan MUD, ASL tanıma ve analizleri için önemli kaynaklardır. Araştırmacılar ve geliştiriciler, bu veri setlerini ASL tanıma sistemlerini eğitmek ve test etmek amacıyla kullanabilirler. Ayrıca, her iki veri setini birleştirerek geliştirilen bir tanıma sisteminin, genellikle daha yüksek bir doğruluk oranına sahip olması beklenmektedir. Bu da, daha geniş veri çeşitliliği ve farklı özelliklerin daha iyi bir şekilde öğrenilmesine olanak tanır.



Şekil 3. Amerikan işaret dilindeki harfleri ve rakamları ifade etmek için kullanılan MUD veri seti görüntüleri (Thakur, 2023)

Kaggle web sitesi üzerinden çok farklı işaret dillerine yönelik veri setleri ve çeşitli varyasyonları elde edilebilir.

1.3. İşaret Dili Tanıma Sistemlerinde Kullanılan Yöntemler

İşaret dili verilerini elde etmek için çeşitli yöntemler kullanılmaktadır. Bunlar arasında, el hareketlerini ve yüz ifadelerini ayrıntılı bir şekilde tespit etmek için yaygın olarak kullanılan kamera tabanlı sistemler, Kinect gibi derinlik sensörleri ve sensörlerle donatılmış eldivenler gibi giyilebilir cihazlar yer almaktadır. Kamera tabanlı sistemler, genellikle video analizi ile işaretleri tanımak için kullanılır ve doğru tanıma için ışık ve açı gibi faktörlere duyarlıdır. Derinlik sensörleri ise, 3D veriler sağlayarak el ve parmak hareketlerinin daha doğru şekilde takip edilmesini sağlar. Bu sensörler, el hareketlerinin yanı sıra ortamın derinliğini de algılayarak, işaret tanıma doğruluğunu artıran ek bir bilgi katmanı sunar. Giyilebilir sensörler ise daha doğrudan etkileşimli veri toplama sağlar ve kullanıcının elleriyle yaptığı hareketleri yüksek doğrulukla kaydedebilir. Bu yöntemler, işaret dili tanıma sistemlerinin verimliliğini ve doğruluğunu artırmak için farklı avantajlar sunar.

Dünyada en yaygın kullanılan işaret dillerinden bazıları ABD, Almanya, Hindistan, Türkiye, Çin, Arjantin, Kore ve İran'a aittir. Bu alanda ise ABD, öne çıkan ülke olarak dikkat çekmektedir. İşaret dili tanıma sistemlerinde, genellikle statik görüntüler en yaygın olarak kullanılır. Bu tür görüntüler için en sık tercih edilen mimariler arasında evrişimli sinir ağları (CNN), derin sinir ağları (DNN) ve kısıtlı Boltzmann makineleri (RBM) yer alır. Dinamik görüntülerin işlenmesinde ise uzun kısa süreli bellek (LSTM), tekrarlayan sinir ağları (RNN) ve 3D evrişimli sinir ağları (3DCNN) gibi mimariler öne çıkmaktadır. Geleneksel sınıflandırma yöntemlerinde, destek vektör makineleri (SVM) ve kümülatif büyüklük histogramları yaygın olarak kullanılırken, geleneksel tanımlayıcılarda ise en çok tercih edilen yöntem gizli Markov modeli (HMM) olmaktadır. İşaret dili tanıma süreçlerinde, el, yüz ve vücut özelliklerinin birleştirilmesi (özellik füzyonu) sıklıkla kullanılsa da, el özellikleri en yaygın veri türü olarak tercih edilmektedir (Rastgoo vd., 2021).

İşaret dili tanımda, rastgele orman (RF) ve ekstrem gradyan artırma (XGBoost) gibi karar ağacı tabanlı topluluk yöntemleri yaygın olarak kullanılmaktadır. RF, karar ağaçlarının (DT) temel tahminci olarak kullanıldığı ve bootstrap örnekleme ile birleştiği popüler bir homojen topluluk örneğidir. Basit yapısına rağmen, RF, güçlü performansı ile tanınan bir algoritmadır. Diğer bir karar ağacı tabanlı topluluk yöntemi olan XGBoost ise, gradyan güçlendirme yapısı kullanarak modelin doğruluğunu artırır. Bunun dışında, k-en yakın komşu (KNN), stokastik gradyan inişi (SGD), naive Bayes Gaussian (NBG) ve yapay sinir

ağları (ANN) gibi makine öğrenimi algoritmaları da işaret dili tanıma sistemlerinde etkin bir şekilde kullanılmaktadır (Pham vd., 2021).

1.4. Tezin Kapsamı ve Amacı

İşaret dili tanıma sistemleri ve görüntü işleme alanlarında pek çok uygulama geliştirilmiş olup, bu yöntemler çeşitli çözüm önerileri sunmaktadır. Ancak, bu yöntemler ölçekleme, gürültüye karşı dayanıklılık ve ayrıntılı özellik verilerinin elde edilmesi gibi belirli sınırlamalarla karşılaşabilmektedir. Bu çalışmada, söz konusu sınırlamaların üstesinden gelmek ve daha etkili bir özellik çıkarımı sağlamak amacıyla karmaşık Zernike momentlerinin (CZM) kullanımı önerilmektedir. Bununla birlikte, çıkarılan bu karmaşık değerli özellik verilerini işleyebilmek için, karmaşık değerli derin sinir ağı (CVDNN) tabanlı bir yöntemin geliştirilmesinin gerekliliği vurgulanmaktadır.

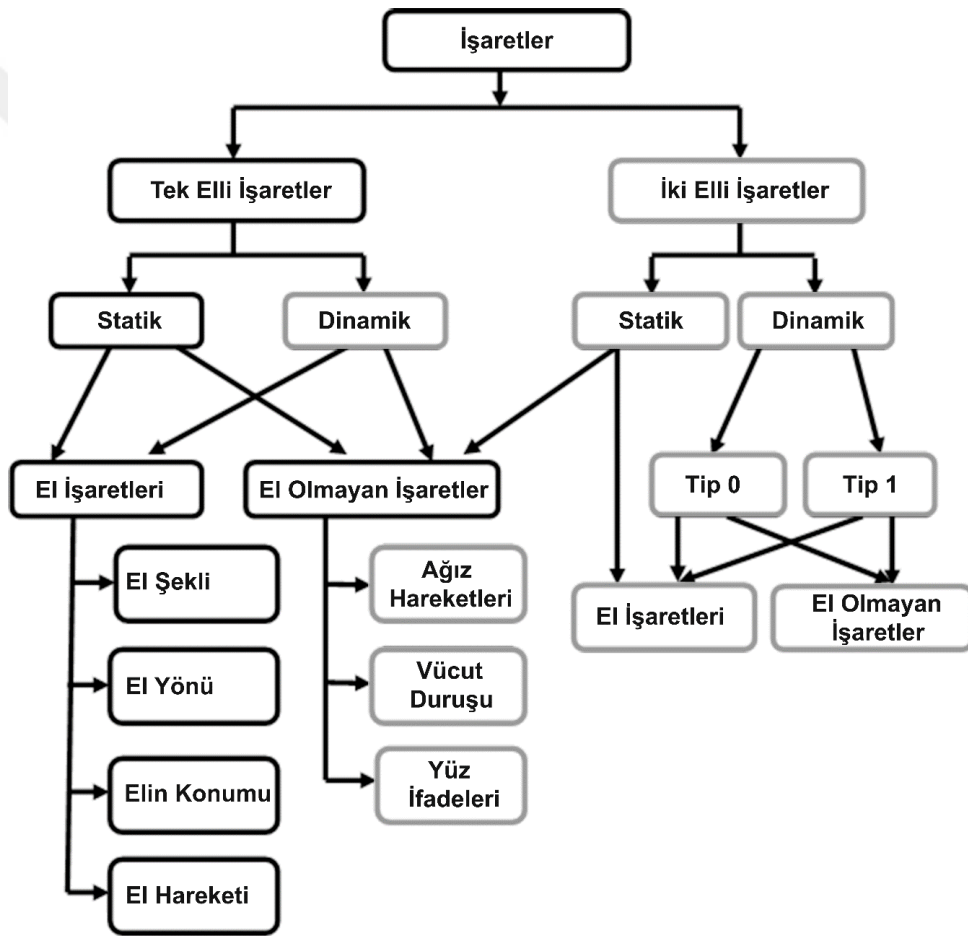
Bu çalışmanın temel amacı, el işareti görüntülerinden elde edilen karmaşık değerli öznitelik vektörünü girdi olarak kullanan karmaşık değerli derin sinir ağı geliştirmek ve bu ağı kullanarak ASL'yi tanımaya yönelik bir sınıflandırıcı model tasarlamaktır. Önerilen model, hızlı özellik çıkarımı ve düşük kaynak tüketimi sağlamakla birlikte, aynı zamanda gerçek zamanlı ve mobil sistemler için de uyumlu olacak şekilde tasarlanmıştır. Ayrıca, bu geliştirilmiş modelin, işaret dili tanıma için gerçek zamanlı bir sistemde etkin bir şekilde kullanılabilmesi amacıyla uygulamaya alınması hedeflenmektedir.

Bu bağlamda, çalışmada öncelikle CZM ile ilgili temel kavramlar ve daha önce yapılan çalışmalar ele alınacaktır. Ardından, CVDNN tasarım sürecine dair ayrıntılar sunulacak ve bu tasarımın gerçekleştirilme adımları detaylı olarak incelenecektir. Bu süreç boyunca, her iki konu başlığı altında elde edilen bulgular ve uygulamaya yönelik aşamalar sistematik bir şekilde değerlendirilecektir.

1.5. Literatürde İşaret Dili Tanıma Üzerine Gerçekleştirilen Çalışmalar

İşaret dili tanıma sistemleri, işaretleri statik ve dinamik olarak iki ana kategoriye ayırır. Statik işaretler, genellikle el şekilleri veya parmak konumlarıyla ifade edilen harfler ve semboller gibi durağan ifadeleri içerir. Bu tür işaretler, genellikle bir anlık görüntü üzerinden tanınır çünkü hareket içeren bir akış yerine sabit bir pozisyonu temsil ederler. Örneğin, parmak alfabesindeki harfler, çoğunlukla statik işaretlere dayanır ve bazı işaret dillerinde

kelimeler de statik işaretlerle ifade edilir. Dinamik işaretler ise belirli bir hareket dizisini takip eder ve genellikle cümleleri ya da daha karmaşık kelimeleri temsil eder. Bu işaretlerin tanınması, hareketlerin doğru bir şekilde takip edilmesi ve analiz edilmesi gerektiği için daha zordur (Wadhawan & Kumar, 2021). Şekil 4, işaret dillerinde yaygın olarak kullanılan işaretlerin hiyerarşik akışını göstermektedir. Ayrıca, bu şekil, her iki elin kullanıldığı işaretlerde baskın ve baskın olmayan el durumlarını belirtmek için Tip 0 ve Tip 1 sınıflandırmalarını içermektedir. Tip 0, her iki elin de aktif olduğu işaretleri ifade ederken, Tip 1, bir elin diğerine göre daha fazla aktif olduğu durumları belirtir.

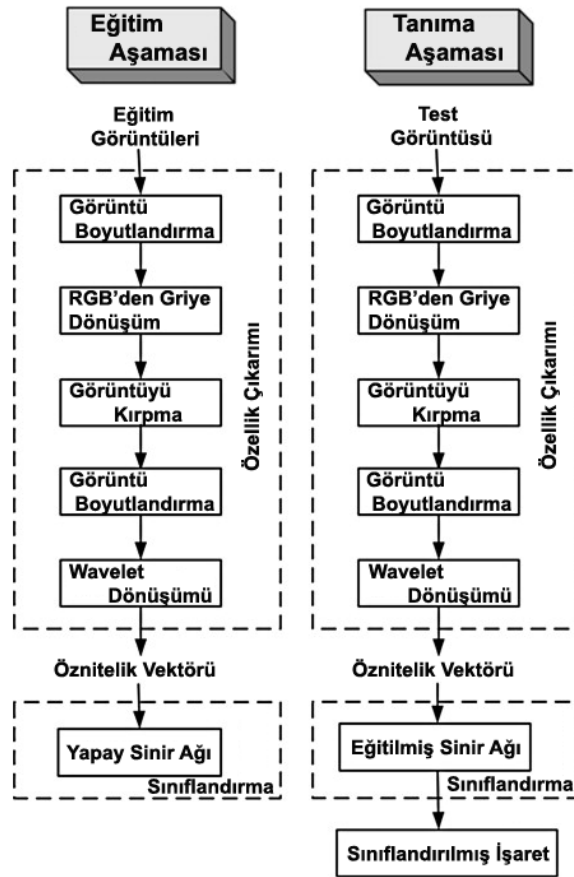


Şekil 4. Genel olarak işaret dillerinde kullanılan hareketlerin hiyerarşik akışı (Wadhawan & Kumar, 2021)

Karami ve ekibi, İran işaret dili alfabesindeki 32 statik işaretin tanınmasına yönelik bir çalışma gerçekleştirmektedir. İlgili çalışmada, özellik çıkarımı için dalgacık dönüşümü (wavelet transform) kullanılmakta, sınıflandırma ise çok katmanlı yapay sinir ağları (MLP)

ile yapılmaktadır. Otomatik segmentasyon yerine, veriler manuel olarak seçilip işlenmiştir. (Karami vd., 2011). Şekil 5, geliştirdikleri modelin mimarisel tasarımını göstermektedir.

Başka bir çalışmada Kelly ve ekibi, Hu momentleri ve öz uzay boyut fonksiyonunun kombinasyonunu kullanarak statik tek elle işaret tanıma için SVM tabanlı bir sistem geliştirmiştir (Kelly vd., 2010). Yöntemlerinin, el işaretlerini kişiden bağımsız olarak tanımda diğer çalışmalara göre daha etkili olduğunu belirtmişlerdir. Bu çalışmada, elin şekli, pozisyonu, yönelimi ve hareketi gibi dört temel bileşen kullanılarak el duruşları tanımlanmakta ve bu bileşenler üzerinden el postürleri sınıflandırılmaktadır.

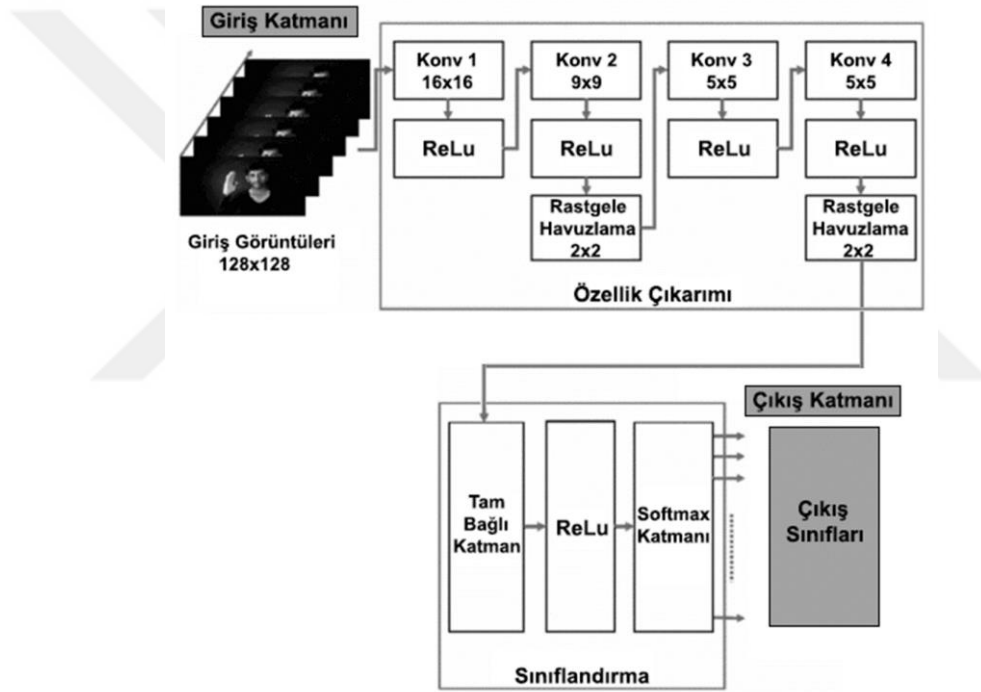


Şekil 5. Karami ve arkadaşları tarafından geliştirilen modelin mimarisi (Karami vd., 2011)

Nandy ve ekibi, Hint işaret dilinde iki elin kullanıldığı dinamik işaretleri tanımaya yönelik bir çalışma gerçekleştirmiştir. Yazarlar, Hint işaret dilinde çeşitli işaretleri içeren bir video veritabanı oluşturmuş ve jest sınıflandırması için yönlü histogramlar, Öklid mesafesi ve KNN gibi yenilikçi teknikler kullanmışlardır. İlgili çalışma Hint işaret dili jestlerinin

tanınmasını geliştirmeyi ve iletişimi daha erişilebilir hale getirmeyi amaçlamaktadır (Nandy vd., 2010).

Rao ve ekibi, selfie modunda elde edilen sürekli (dinamik) işaret dili verilerini kullanarak Hint işaret dilini tanımak için CNN tabanlı bir çalışma gerçekleştirmiştir. Çalışmanın amacı, mobil platformda çalışabilecek bir sistem için algoritmalar simüle etmektir (Rao vd., 2018). 128x128 piksel büyüklüğündeki giriş görüntülerini alan sistem aktivasyon fonksiyonu olarak ReLU, çıkış katmanını yorumlamak amacıyla ise Softmax fonksiyonunu kullanmaktadır. Özellik çıkarımı ve sınıflandırma modülleri de dahil olmak üzere ilgili sistemin mimarisi Şekil 6'da gösterilmektedir.



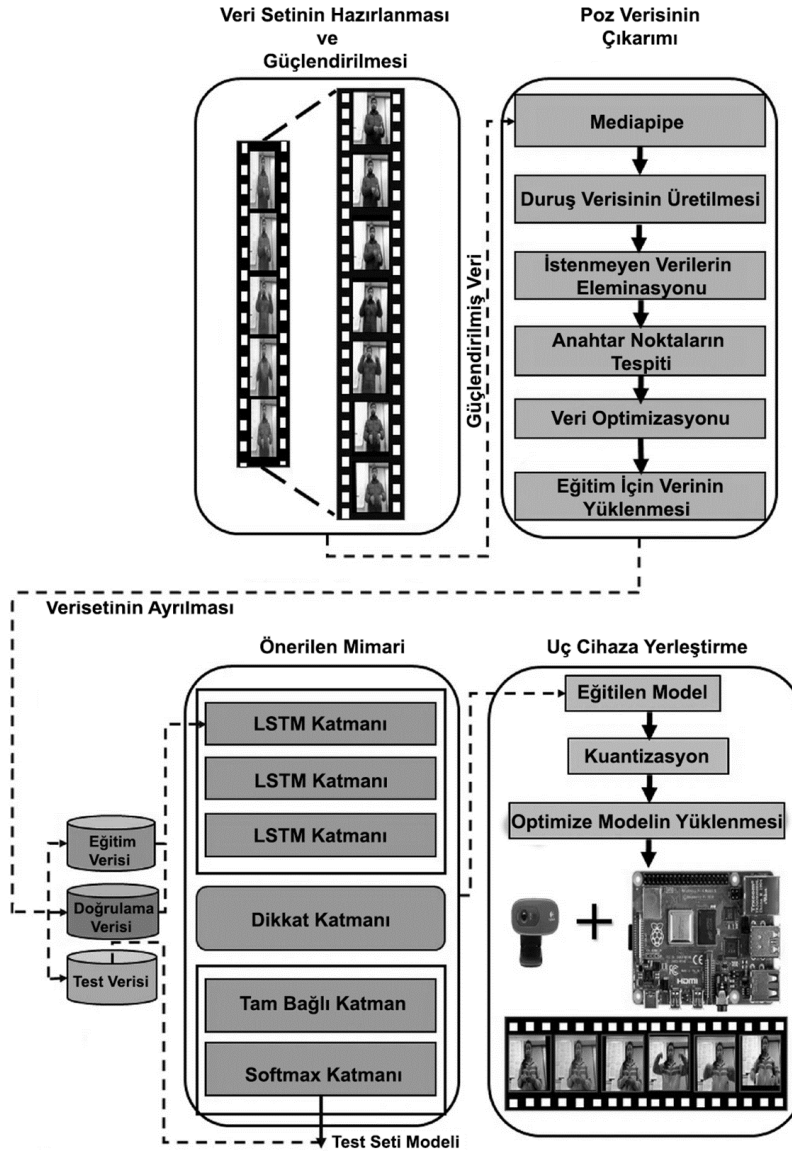
Şekil 6. Rao ve ekibi tarafından geliştirilen modelin mimarisi (Rao vd., 2018)

Koller ve ekibi Alman işaret dilinde sürekli işaret tanıma için HMM ve CNN'i birleştiren hibrit bir model sunmuştur. Bu model, CNN'lerin güçlü ayrıştırıcı yeteneklerini HMM'lerin sıralı veri modelleme yetenekleriyle birleştirir. İlgili çalışma, eğitim ve değerlendirme için sıralı verilerle başa çıkmanın önemini tartışmaktadır. Hibrit yaklaşım, kelime hata oranında %20'ye kadar iyileşmeler sağlamaktadır (Koller vd., 2018).

Sreemathy ve ekibi Hint işaret dilinde iki elin kullanıldığı işaretlerin otomatik tanınması için bir yöntem önermiştir. Bu yöntem veriler üzerinde ön işleme, özellik çıkarımı

ve sınıflandırmayı içermektedir. Önerilen teknoloji, özel gereksinimleri olan bireyler için bir öğretim asistanı olarak kullanılmış ve çocukların bilişsel yeteneklerinde %60–70'lik bir artış göstermiştir (Sreemathy vd., 2023).

Sharma ve ekibi, dinamik işaret dilini gömülü cihazlarda tanımak için bellek açısından verimli LSTM tabanlı bir derin öğrenme mimarisi geliştirmiştir. Çalışmada, takvim aylarını temsil eden bir Hint İşaret Dili veri seti oluşturmuştur. Toplamda 480 video kaydı toplanmış ve bu veriler üzerinden model eğitilmiştir. Model boyutunu küçültmek ve uç cihazlarda kaynak tüketimini azaltmak için eğitim sonrası kuantizasyon uygulamışlardır (Sharma vd., 2024). Geliştirilen modelin mimarisi Şekil 7’de gösterilmiştir.

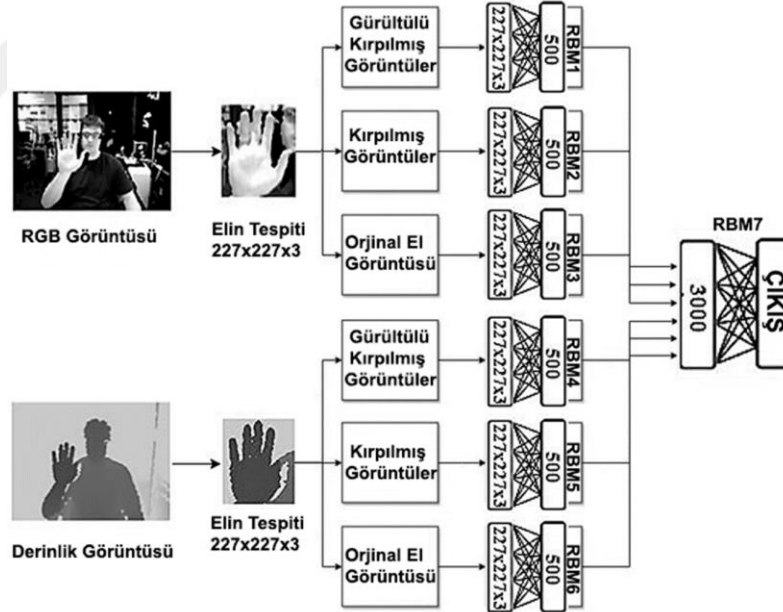


Şekil 7. Sharma ve ekibi tarafından geliştirilen modelin mimarisi (Sharma vd., 2024)

Cui ve ekibi, Alman işaret dilinde sürekli işaret tanıma için CNN ve Bi-LSTM kombinasyonunu kullanarak bir sistem geliştirmiştir. Sıra öğrenme modülü olarak RNN kullanarak, bu yaklaşımın zaman içindeki bağımlılıkları öğrenmede HMM'leri geçtiğini belirtmişlerdir (Cui vd., 2019).

Zhou ve meslektaşları, video tabanlı işaret dili anlayışını artırmak için çoklu ipucu öğrenme yaklaşımını kullanmış ve görsel tabanlı sıra öğrenme problemini çözmek için bir mekânsal-zamansal çoklu ipucu ağı önermişlerdir (Zhou vd., 2022). Biri zamansal biri mekânsal olmak üzere iki modülden oluşan ağ, birden fazla ipucunun işbirliğini keşfetmeyi amaçlamaktadır.

Rastgoo ve arkadaşları tarafından yapılan bir diğer çalışmada, RGB ve derinlik modlarını kullanarak görsel verilerden otomatik işaret dili tanıma sürecinde RBM kullanılmıştır. Çalışmanın hedeflerinden biri, önerinin gürültüye karşı dayanıklılığını değerlendirmektir (Rastgoo vd., 2018). Şekil 8, geliştirdikleri modelin mimarisini temsil etmektedir.



Şekil 8. Rastgoo ve ekibi tarafından geliştirilen modelin mimarisi (Rastgoo vd., 2018)

Samaan ve ekibi, dinamik işaret dili tanıma sorunlarını çözmek için MediaPipe ve RNN modellerini bir arada kullanmaktadır. MediaPipe ellerin, vücudun ve yüzün ana noktalarını çıkarmak için kullanılır ve bu sayede konumun, şeklin ve yönelimin belirlenmesini sağlar (Samaan vd., 2022).

1.6. Tez Kapsamında Kullanılacak Veri Setleri

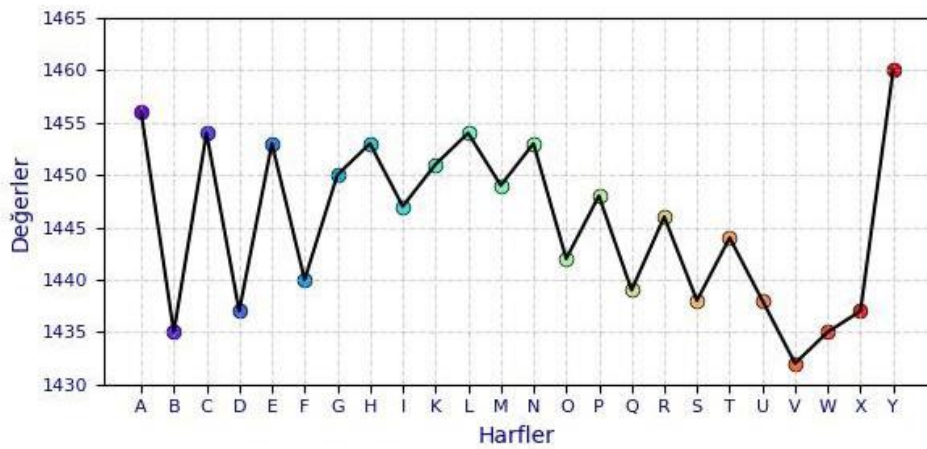
Bir önceki bölümde işaret dili ile ilgili yapılan çeşitli çalışmalar incelenmiştir. Bu bölümde ise doktora çalışmasında kullanılan veri setleri hakkında ayrıntılı bilgiler sunulacaktır. Veri setleri tanıtıldıktan sonra, literatürde bu veri setlerini kullanan bazı çalışmalara da yer verilecektir.

1.6.1. MNIST Veri Seti

İşaret dili MNIST veri seti, ASL harflerinin el görüntülerini içeren bir veri setidir. Bu veri seti, işaret dili tanıma sistemlerini eğitmek ve test etmek amacıyla kullanılmaktadır. CSV formatında sunulan veri seti, Kaggle web sitesi üzerinden temin edilebilir (Tecperson, 2023). Veri seti, eğitim ve test olmak üzere iki ana kısımdan oluşmaktadır.

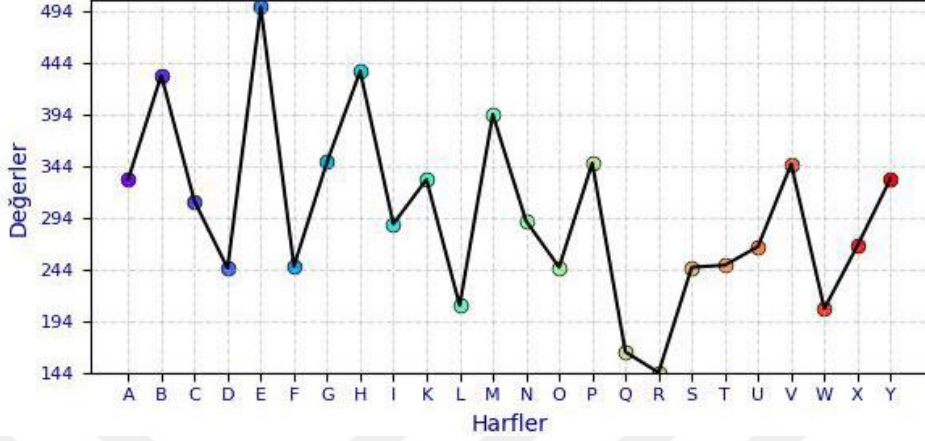
Veri seti, hareketle ifade edilen J ve Z harfleri hariç, İngiliz alfabesindeki tüm harfleri kapsayan 24 sınıftan oluşmaktadır. 28x28 çözünürlüğündeki görüntülerden oluşan bu set, eğitim ve test verisi olarak toplamda 34.627 gri tonlama görüntü içermektedir. Bu görüntülerin 27.455'i eğitim verisi, 7.172'si ise test verisi olarak ayrılmıştır. Diğer bir deyişle, veri setinin yaklaşık %80'i eğitim, %20'si ise test verisi olarak kullanılmıştır. Eğitim ve test verilerindeki sınıf başına örnek sayılarında farklılıklar bulunmaktadır.

Şekil 9, MNIST eğitim veri seti içindeki her bir sınıfın içerdiği görüntülerin sayısını göstermektedir. İlgili grafikten görüleceği üzere, veri seti içindeki görüntüler sınıf başına genel olarak düzenli bir dağılım göstermektedir.



Şekil 9. MNIST eğitim veri seti içindeki her bir sınıfa ait görüntü sayısı

Şekil 10 ise MNIST test veri seti içindeki her bir sınıfın içerdiği görüntülerin sayısını göstermektedir.



Şekil 10. MNIST test veri seti içindeki her bir sınıfa ait görüntü sayısı

1.6.2. MUD Veri Seti

Çalışmalarda kullanılan bir diğer veri seti ise MUD'dur (Thakur, 2023). Bu veri seti, İngiliz alfabesindeki harfler (A-Z) ve sayılar (0-9) dahil olmak üzere toplam 36 sınıf içermektedir. Veri setinde 65 görüntü içeren T sınıfı hariç her sınıf 70 görüntü içerirken, toplamda 2.515 işaret bulunmaktadır. Görüntüler 400x400 boyutlarında ve RGB formatındadır. Beş denek tarafından oluşturulan bu veri seti, hem Holdout hem de bir deneği dışarda bırakma (LOSO) doğrulama teknikleri için uygundur.

1.6.3. Model Doğrulama ve Değerlendirme Teknikleri

Holdout doğrulama tekniği, makine öğrenimi modellerini değerlendirmek ve doğrulamak için kullanılan temel bir yöntemdir (Nacar & Erdebilli, 2021). Bu teknikte, veri seti genellikle eğitim, doğrulama ve test olmak üzere iki veya üç ayrı alt kümeye bölünür. Eğitim alt kümesi modelin öğrenmesi için kullanılırken, test alt kümesi modelin performansını değerlendirmek için ayrılır. Eğer doğrulama kümesi de kullanılıyorsa, modelin hiperparametre ayarlarını optimize etmek amacıyla eğitim ve doğrulama verileri arasında bir denge sağlanır. Bazı çalışmalar, veri setini %70 eğitim ve %30 test olarak

ayırırken, diğerleri %80 eğitim ve %20 test oranını tercih etmektedir. Holdout tekniği, özellikle büyük veri setlerinde hızlı ve basit bir değerlendirme yöntemi sunarken, küçük veri setlerinde veri israfına yol açabilir. Bu nedenle, küçük veri setlerinde çapraz doğrulama tekniklerine kıyasla daha az güvenilir olabilir.

LOSO tekniği, özellikle insan tabanlı veri setlerinde kullanılan bir çapraz doğrulama yöntemidir (Maleki vd., 2020). Bu teknikte, veri setindeki her denek sırasıyla test kümesi olarak ayrılırken, geri kalan deneklerin verileri eğitim kümesi olarak kullanılır. Böylece, her denek bir kez test verisi olarak kullanılır ve model, her denek için bağımsız bir şekilde eğitilip test edilir. LOSO çapraz doğrulama tekniğinde, veri setindeki toplam denek sayısı K ise, bir denegin test için bırakıldığı ve geri kalan $(K - 1)$ denegin eğitim için kullanıldığı anlamına gelir. Bu süreç K kez, her denek için bir kez tekrar edilir. Elde edilen başarı oranlarının aritmetik ortalaması nihai sonuç olarak alınır. LOSO tekniği, özellikle kişisel farklılıkların önemli olduğu biyometrik tanıma, sağlık verileri ve davranışsal analiz gibi alanlarda kullanışlıdır. Kişilerarası genellemeyi test etmek için güçlü bir yöntem sunar, çünkü modelin her birey için ne kadar iyi performans gösterdiği detaylı şekilde analiz edilebilir. Ancak bu teknik, veri seti büyükse zaman ve işlem maliyeti açısından oldukça yükündür.

1.7. MNIST Veri Seti Üzerinde Gerçekleştirilen Literatür Çalışmaları

Çok sayıda işaret dili ve bu diller için mevcut olan çeşitli veri setleri nedeniyle, gerçekleştirilen çalışmaların performans testlerini karşılaştırmak zorlaşmaktadır. Bu bölümde bahsedilen tüm çalışmalar, MNIST veri setinin tamamı üzerinde gerçekleştirilmiştir (Tecperson, 2023). Bu durum, ilgili çalışmaların birbiriyle karşılaştırılmasını kolaylaştırmaktadır.

Li ve arkadaşları, ASL'yi tanımak amacıyla MNIST veri seti üzerinde bir çalışma yürütmektedir. Çalışmada, ham veriler ve bazı ön işleme adımları (el tespiti, arka plan çıkarma, gürültü azaltma vb.) uygulanarak elde edilen veriler üzerinde çeşitli sınıflandırma algoritmalarının performansı test edilmiştir (Li. vd, 2022). Ham veriler üzerinde yapılan çalışmalar sonucunda, SVM, rastgele orman sınıflandırıcısı (RFC), KNN, SGD ve NBG gibi makine öğrenmesi ve sınıflandırma algoritmaları kullanılarak elde edilen doğrulama değerleri Tablo 1'de listelenmiştir. Ayrıca, doktora çalışmaları kapsamında geliştirilecek modelde kullanılacak olan optimizasyon algoritması olan SGD için tanıma oranını %66.02

olarak elde etmişlerdir. Çalışmalarında, temel bileşen analizi ve manifold algoritması gibi farklı boyut indirgeme tekniklerini kullanarak, Tablo 1'deki her makine öğrenmesi ve sınıflandırma algoritması üzerindeki etkilerini de incelemişlerdir. Elde edilen sonuçlar, farklı boyut indirgeme tekniklerinin her algoritmada aynı şekilde sonuç vermediğini ve KNN için en uygun algoritmanın manifold, diğerleri için ise temel bileşen analizi olduğunu ortaya koymaktadır.

Tablo 1. Li ve arkadaşları tarafından MNIST veri seti üzerinde gerçekleştirilen çalışmalarının sonuçları

Makine Öğrenmesi veya Sınıflandırma Algoritması	Veri Seti	Tanıma Oranı
SVM	MNIST	84.19
RFC	MNIST	81.61
KNN	MNIST	78.17
SGD	MNIST	66.02
NBG	MNIST	38.90

Son yıllarda, tanıma sistemlerinde yüksek başarı oranları nedeniyle derin öğrenme ağları tercih edilmektedir. Her aşamada, probleme özgü stratejiler belirlenmekte ve doğruluğu artırmak için çeşitli çabalar gösterilmektedir. Örneğin, CNN'ler ön işleme ve özellik çıkarma aşamalarını otomatik olarak gerçekleştirmektedir (Bhatt vd., 2021). LeNet, başarılı ilk evrişimli sinir ağı modellerinden biri olarak öne çıkmaktadır. Bilgin ve Mutludoğan, LeNet-5 modelini MNIST veri seti üzerinde kullanmaktadır. LeNet-5 modelinin test verisi üzerindeki tanıma oranını %82.19 olarak rapor etmektedirler. Bir diğer derin öğrenme ağı olan CapsNet'te ise MNIST veri seti üzerinde %88.93 tanıma oranına ulaşmaktadırlar (Bilgin & Mutludoğan, 2019). Çalışmalarında, kapsül ağlarının işaret dili karakter tanıma için faydalı olduğunu ve LeNet'ten daha başarılı olduğunu belirtmektedirler.

Bhushan ve çalışma arkadaşları, Amerikan işaret dilini tanımaya yönelik olarak MNIST veri seti üzerinde CNN tabanlı bir çalışma yapmışlardır. Bu çalışmada, KNN, stokastik gradyan iniş sınıflandırıcısı (SGDC) ve NBG gibi çeşitli makine öğrenmesi algoritmaları incelenmekte ve karşılaştırılmaktadır. Çalışmalarında herhangi bir ön işleme yapılmadan veri seti üzerinde yapılan analizler sonucunda, CNN, RF, XGBoost, KNN,

SGDC ve NBG gibi makine öğrenmesi algoritmaları ile elde edilen doğrulama değerleri Tablo 2’de sunulmuştur (Bhushan vd., 2022). Gerçekleştirdikleri çalışmada, SGD algoritmasının sınıflandırma işlemleri için uyarlanmış bir versiyonu olan SGDC kullanmışlar ve tanıma oranını 59.80% olarak elde etmişlerdir. En iyi sonuçlar ise CNN ile elde edilmiştir.

Tablo 2. Bhushan ve arkadaşları tarafından MNIST veri seti üzerinde gerçekleştirilen çalışmalarının sonuçları

Makine Öğrenmesi veya Sınıflandırma Algoritması	Veri Seti	Tanıma Oranı
CNN	MNIST	91.41
RF	MNIST	84.43
XGBoost	MNIST	81.35
KNN	MNIST	80.46
SGDC	MNIST	59.80
NBG	MNIST	38.90

Kameralar aracılığıyla elde edilen görüntülerdeki birçok sorunun yanında, ışık yoğunluğundaki değişim bu tür görüntülerden hareketlerin çıkarılmasını zorlaştırmaktadır. Kamera görüntülerdeki gürültü gibi ekstra bilgiler, hesaplama süresini etkilemekte ve engellemektedir. Ayrıca karmaşık arka planlar, hareketlerin çıkarılmasını zorlaştırmaktadır. Ansar ve ekibi, ASL’yi tanımak için bir CNN mimari modeli geliştirmişler ve ham veri seti üzerinde bazı ön işlemler gerçekleştirmişlerdir. Tanıma süreci beş ana aşamaya ayrılmaktadır. İlk aşama, görüntülerdeki gürültüyü azaltmak ve ışık yoğunluğunu ayarlamaktır. İkinci aşama, elin tespit edilmesi ve arka planın kaldırılmasıdır. Üçüncü aşama, elde edilen görüntüdeki ana noktaların (landmark tespiti) belirlenmesidir. Sonraki aşamada, özellik çıkarımı yapılır. Son aşamada ise çıkarılan özellikler CNN sınıflandırıcısından geçirilir. Geliştirdikleri modeli MNIST veri seti üzerinde iki farklı doğrulama işlemine tabi tutmaktadırlar. İlk doğrulama için veri setinin üçte biri test için kullanılırken, geri kalanı eğitim için kullanılmaktadır. İkinci doğrulama için ise veri setinin üçte ikisi eğitim için, geri kalanı ise test için kullanılmaktadır. İlk değerlendirme için %93.2

ve ikinci değerlendirme için %91.6 doğruluk oranlarına ulaşmaktadırlar. Ancak, bu değerlerin veri setinin ilgili bölümlerinde (tüm test verisi üzerinde değil) elde edildiğini belirtmekte fayda bulunmaktadır (Ansar vd., 2023). Bu sebeple, Ansar ve ekibinin gerçekleştirdiği çalışmaya ait sonuçlar, MNIST veri seti üzerinde gerçekleştirilen diğer akademik çalışmalarla ve doktora tez çalışmaları kapsamında elde edilen sonuçlarla karşılaştırılmayacaktır.

Tablo 3, MNIST veri seti üzerinde tanıma oranı baz alınarak gerçekleştirilen akademik çalışmalardan elde edilen sonuçları toplu olarak listelemektedir.

Tablo 3. MNIST veri seti üzerinde gerçekleştirilen akademik çalışmaların test verisi üzerindeki tanıma oranları

Grup	Model/Metot	Tanıma Oranı (%)
(Li. vd, 2022) / (Bhushan vd., 2022)	NBG	38.9
(Bhushan vd., 2022)	SGDC	59.8
(Li. vd, 2022)	SGD	66.02
(Li. vd, 2022) / (Bhushan vd., 2022)	KNN / KNN	78.17 / 80.46
(Bhushan vd., 2022)	XGBoost	81.35
(Li. vd, 2022) / (Bhushan vd., 2022)	RFC / RF	81.61 / 84.43
(Bilgin & Mutludoğan, 2019)	LeNeT-5	82.19
(Li. vd, 2022)	SVM	84.19
(Bilgin & Mutludoğan, 2019)	CapsNet	88.93
(Bhushan vd., 2022)	CNN	91.41

1.8. MUD Veri Seti Üzerinde Gerçekleştirilen Literatür Çalışmaları

Bu bölümde, bir başka ASL veri seti olan MUD ile ilgili çalışmalara yer verilmektedir. MUD veri seti, eğitim ve test verilerine ayrılmadığından, bu veri setini kullanan çalışmalarda model başarısını değerlendirmek için genellikle Holdout ve LOSO gibi doğrulama teknikleri kullanılmaktadır.

Barbhuiya ve çalışma arkadaşları, CNN özelliklerini geleneksel Zernike moment tabanlı özelliklerle birleştiren yeni ve etkili bir görüntü tanıma yaklaşımı önermektedir. Bu

hibrit yaklaşımın, özellikle benzer el duruşları arasındaki farkları ayırt etme kabiliyetini artırması hedeflenmektedir. Önerilen model, CNN'nin (özellikle AlexNet) son evrimsel katmanından elde edilen özellikleri, Zernike moment özellikleriyle birleştiren bir mimari sunmaktadır.

Bu yaklaşım, MUD veri setinde %98.41 tanıma oranı ile üstün bir performans göstermiştir. Çalışmada, veri seti Holdout tekniği kullanılarak %80 eğitim ve %20 test olacak şekilde bölünmüştür. Ancak, eğitim sırasında her adımda yapılan rastgele seçimler nedeniyle veri setindeki bazı görüntülerin hem eğitim hem de test setlerinde yer alabileceği belirtilmiştir (Barbhuiya vd., 2022).

Bhaumik ve ekibi, el hareketlerini doğru bir şekilde tanımak amacıyla, mekansal özellikleri öğrenen ve çok ölçekli filtreler kullanarak zengin mekansal bilgileri çıkaran CNN tabanlı taşınabilir bir ağ olan SpAtNet'i önermektedirler. SpAtNet, çevresel koşullar, döndürme, ölçekleme ve aydınlatma gibi zorlukları aşmak için tasarlanmış hafif bir CNN tabanlı ağdır. Model, çok ölçekli dikkatli özellik birleştirme ve iç içe geçmiş modül olmak üzere iki ana bileşenden oluşmaktadır. Çok ölçekli dikkatli modül, 1x1, 3x3 ve 5x5 boyutlarındaki çok ölçekli filtreler kullanarak zengin mekansal bilgileri çıkarmaktadır. Küçük filtreler, daha ince özellikleri; büyük filtreler ise daha kaba özellikleri çıkarmaktadır. İç içe geçmiş modül, dört evrim katmanının ardışık olarak dizilmesiyle yapılandırılmış olup, yüksek seviyeli bağlamsal özelliklerin öğrenilmesi amacıyla kullanılmaktadır. MUD veri seti üzerinde LOSO tekniği kullanarak yapılan değerlendirmede, %80.44 tanıma oranına ulaşmaktadırlar (Bhaumik vd., 2024).

Chevtchenko ve ekibi, gerçek zamanlı el duruşu tanıma amacıyla, geleneksel özellik çıkarıcıları birleştiren CNN mimarisi tabanlı bir model geliştirirler. Model, Gabor filtreleri, Zernike momentleri ve Hu momentleri gibi klasik öznitelik tanımlayıcılarını, CNN'ler ile birleştirerek tanıma doğruluğunu artırmayı amaçlamaktadır. Modelin performansını artırmak için hiperparametrelerin seçimi, ağaç yapılı Parzen tahminleme algoritması kullanılarak gerçekleştirilmiştir. MUD veri seti üzerinden her iki doğrulama tekniğiyle değerlendirilen çalışmada, Holdout yöntemiyle %80 eğitim ve %20 test olarak bölündüğünde %98.05 tanıma oranı, LOSO yöntemiyle ise %84.02 tanıma oranı elde edilmiştir (Chevtchenko vd., 2018).

Barbhuiya ve ekibinin bir başka çalışmasında, SVM sınıflandırıcısı ile el işaretlerini tanımak için modifiye edilmiş VGG16 ve AlexNet modellerine dayanan evrimsel bir sinir ağı mimarisi geliştirilmiştir. Bu modeller, özellik çıkarımı için kullanılmış ve ardından çok

sınıflı bir destek vektör makinesi sınıflandırıcısı ile birleştirilmiştir. Her iki model de aynı sonuçları elde etmiş ve LOSO yöntemiyle %70 tanıma oranına ulaşmıştır (Barbhuiya vd., 2021).

Tablo 4, MUD veri seti üzerinde tanıma oranı baz alınarak gerçekleştirilen akademik çalışmalardan elde edilen sonuçları, ilgili tekniklere göre toplu olarak listelemektedir. Holdout tekniği için en yüksek değer %98.41, LOSO tekniği için ise %84.02 olarak elde edilmiştir.

Tablo 4. MUD veri seti üzerinde gerçekleştirilen akademik çalışmaların Holdout ve LOSO tekniklere göre tanıma oranları

Grup	Teknik	Model/Metot	Tanıma Oranı (%)
(Barbhuiya vd., 2022)	Holdout	Inception-v3	93.5
(Barbhuiya vd., 2022)	Holdout	Squeezenet	95.23
(Barbhuiya vd., 2022)	Holdout	Vgg19	96.68
(Barbhuiya vd., 2022)	Holdout	AlexNet	97.01
(Barbhuiya vd., 2022) / (Chevtchenko vd., 2018)	Holdout	Vgg16	97.48 / 92.20
(Chevtchenko vd., 2018)	Holdout	Hibrit	98.05
(Barbhuiya vd., 2022)	Holdout	Hibrit	98.41
(Chevtchenko vd., 2018)	LOSO	Vgg16	61.42
(Chevtchenko vd., 2018) / (Bhaumik vd., 2024)	LOSO	MobileNet	62.53 / 73.25
(Barbhuiya vd., 2021)	LOSO	Modified Vgg16 + SVM	70.00
(Barbhuiya vd., 2021)	LOSO	Modified AlexNet + SVM	70.00
(Bhaumik vd., 2024)	LOSO	MobileNetV2	72.28
(Bhaumik vd., 2024)	LOSO	ResNet50	74.42
(Chevtchenko vd., 2018)	LOSO	CNN	73.86/ 78.51 / 79.04
(Bhaumik vd., 2024)	LOSO	CNN	80.44
(Chevtchenko vd., 2018)	LOSO	Hibrit	84.02

1.9. Özellik Çıkarımı

Özellik çıkarma ve sınıflandırma, makine öğrenmesi ve görüntü işleme gibi birçok alanda kritik bir adımdır. Özellik çıkarma, bir veri kümesindeki anlamlı bilgileri seçip, bu bilgileri kullanarak sınıflandırma yapmak amacıyla veri setinin daha düşük boyutlu bir temsiline dönüştürme işlemidir. Bu sayede verinin daha iyi anlaşılması ve modellenmesi sağlanır. Görüntü işleme gibi alanlarda bu süreç, hem sınıflandırma doğruluğunu artırmak hem de hesaplama maliyetlerini düşürmek için hayati önem taşır.

Temel özellik çıkarma tekniklerinden biri, yönlü gradyanların histogramı (HOG)'dur. HOG, görüntüdeki yerel kenar bilgilerini çıkarmak için kullanılır ve özellikle nesne tespitinde oldukça etkilidir. Görüntüdeki her pikselin gradyan yönü hesaplanarak yönlü histogramlar oluşturulur. Bu yöntem, özellikle insan tespiti ve nesne tanıma gibi uygulamalarda başarıyla kullanılmaktadır.

Yerel ikili desenler (LBP) ise dokusal bilgileri çıkarmada yaygın olarak kullanılan bir tekniktir. Bu yöntem, her pikselin komşularıyla olan ilişkisini analiz ederek görüntüdeki dokuları tanımlar. LBP, yüz tanıma ve dokusal analiz gibi alanlarda sıkça tercih edilir.

Gelişmiş özellik çıkarma yöntemlerinden biri, ölçek-bağımsız özellik dönüşümü (SIFT)'tir. SIFT, görüntüdeki önemli anahtar noktaları tespit eder ve bu noktaların ölçekten bağımsız özelliklerini çıkarır. Böylece, farklı ölçeklerdeki veya açılardaki nesneler doğru şekilde tanınabilir.

Bir diğer gelişmiş yöntem olan momentler, görüntüdeki global şekil özelliklerini çıkaran matematiksel bir tekniktir. Momentler, özellikle biyometrik tanıma ve görüntü sınıflandırma gibi uygulamalarda kullanılır. Gabor filtreleri ise belirli bir frekanstaki dokusal bilgiyi çıkarmak için kullanılan doğrusal filtrelerdir. Bu filtreler, yerel frekans bilgilerini analiz ederek dokusal bilgiyi tanır ve doku analizi ile yüz tanıma gibi alanlarda oldukça etkilidir.

Derin öğrenme tabanlı özellik çıkarma tekniklerinden en yaygın olanı CNN'dir. CNN'ler, görüntü işleme alanında otomatik özellik çıkarımı yapan derin öğrenme modelleridir. Geleneksel yöntemlerden farklı olarak, özelliklerin manuel olarak belirlenmesi gerekmez; CNN'ler, veriden bu özellikleri kendisi öğrenir. Özellikle büyük veri setlerinde ve karmaşık sınıflandırma problemlerinde üstün performans sergiler. CNN'ler, görüntü sınıflandırma ve nesne algılama gibi birçok farklı alanda yaygın olarak kullanılır.

Özellik çıkarma işlemi tamamlandıktan sonra, bu özellikleri kullanarak sınıflandırma yapmak gerekmektedir. SVM, verileri ayırmak için bir hiper-düzlem kullanarak sınıflandırma yapan güçlü bir algoritmadır ve özellikle yüksek boyutlu veri setlerinde başarılı sonuçlar elde eder. DT ise sınıflandırma için verileri ardışık şekilde bölerek karar veren daha basit bir yöntemdir ve anlaşılabilir modeller sunar. ANN, özellikle çok boyutlu ve karmaşık veri setlerinde güçlü sonuçlar elde edebilen daha gelişmiş bir sınıflandırma yöntemidir. ANN'ler, büyük veri kümeleri üzerinde çalışırken üstün performans sağlar.

Sonuç olarak, özellik çıkarma ve sınıflandırma teknikleri, görüntü işleme ve makine öğrenmesi alanlarında başarılı modeller oluşturma temel yapı taşlarıdır. Hem geleneksel yöntemler (HOG, SIFT, LBP) hem de CNN gibi derin öğrenme tabanlı yöntemler, farklı veri tiplerine ve uygulama alanlarına göre etkili çözümler sunar. Doğru özellik çıkarma ve sınıflandırma yöntemi, bir projenin başarısında belirleyici bir rol oynar (Wadhawan & Kumar, 2021).

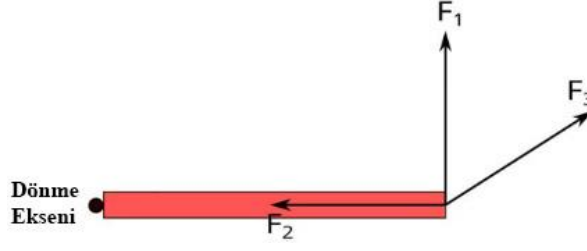
1.9.1. Momentler

Moment kavramı fizik, istatistik ve mühendislik gibi farklı disiplinlerde çeşitli anlamlar ve uygulamalara sahiptir.

Fizikte bir kuvvetin momenti, bir cismin belirli bir nokta veya eksen etrafında dönme eğilimini ölçen bir büyüklüktür. Bu durum cismin doğrusal hareketinden (ötelemesinden) farklı olarak yalnızca dönme hareketiyle ilgilidir. Bir momentin oluşabilmesi için kuvvetin cismin merkezinden geçmeyecek şekilde uygulanması gerekir. Moment, genellikle kuvvetin büyüklüğü ile kuvvetin uygulandığı noktanın dönme eksenine olan dik uzaklığının çarpımı olarak ifade edilir.

Bu döndürme etkisi, kuvvetin doğrultusunda zıt ve eşit bir kuvvetin bulunmaması durumunda ortaya çıkar ve döndürme eğiliminin büyüklüğü kuvvetin uygulama noktasına olan dik uzaklık arttıkça artar. Şekil 11, kuvvetlerin moment etkisini göstermektedir. İlgili moment etkisi bağıntı (1) kullanılarak hesaplanır. Bağıntı (1)'de M dönme etkisine, F uygulanan kuvvete ve d kuvvetin uygulama noktasının dönme eksenine olan uzaklığına karşılık gelir. Dolayısıyla Şekil 11'de F_2 kuvvetinin momenti sıfırdır.

$$M = F \cdot d \quad (1)$$



Şekil 11. Fizikte kuvvetlerin moment etkisi

Moment vektörel bir büyüklüktür ve momentin yönü kuvvetin döndürme yönüne göre pozitif ya da negatif olabilir. Bu durum, her bir kuvvetin etkisinin hangi yönde olduğunu belirler. Bir cisme etki eden birden fazla kuvvet olduğunda, bu kuvvetlerin her biri için hesaplanan momentler toplanarak net moment bulunabilir. Bu özellik, momentlerin süperpozisyonu olarak adlandırılır ve birden fazla kuvvetin etkisinin değerlendirilmesinde önemli bir rol oynar. Fizikte moment, özellikle dönen cisimlerin davranışlarını anlamak ve tasarlamak için temel bir kavramdır ve gerçek hayatta birçok sistemde kullanılır. Moment kavramının daha iyi anlaşılabilmesi için farklı sistemlerden aşağıdaki örnekler verilebilir.

Kapı açma işlemi, fiziksel momentin günlük hayatımızda nasıl işlediğini anlamamıza yardımcı olan iyi bir örnektir. Bu örnekte kapının menteşe noktası, kapının döneceği eksenini oluşturur. Kapıyı açmak için uyguladığınız kuvvet, kapının menteşesine olan uzaklığa göre farklı döndürme etkileri yaratır. Kuvveti kapı kolunun yakınına uygularsanız, kapıyı açmak çok daha kolay olur. Çünkü daha uzun bir mesafe, daha fazla moment üretir. Öte yandan, kuvveti menteşeye çok yakın bir noktada uygularsanız, kapıyı açmak zorlaşır çünkü moment daha düşük olur. Örneğin 10 Newton'luk bir kuvvet kapının kenarından 1 metre uzakta uygulandığında 10 Newton-metre tork üretilir. Özetle moment, kapının dönme eksenini etrafında ne kadar döneceğini gösterir. Yüksek bir moment kapının daha hızlı ve kolay açılmasını sağlar. Bununla birlikte sürekli yüksek momentlere maruz kalan kapı zamanla menteşe yerlerinde aşınma ve yıpranma yaşayabilir. Bu nedenle kapıların tasarımında moment hesapları, hem kullanım verimliliğini hem de yapısal dayanıklılığını sağlamak için önemlidir.

Robotik kollar, genellikle eklem ve motorlardan oluşur ve endüstriyel montaj, cerrahi uygulamalar gibi farklı alanlarda kullanılır. Robotik kolun hareketi, uygulanan kuvvet ve kol parçalarının uzunluğuna bağlıdır. Belirli bir yükü kaldırabilmek için gerekli momentin hesaplanması önemlidir. Kolun uzunluğu ve eklem yerleşimi doğru ayarlanmazsa, motorlar yeterli kuvvet üretemeyebilir. Ağırlığı ve boyutu büyük olan nesneler, kolun dönme eksenini

etrafında daha büyük bir moment oluşturur ve bu da motorların daha fazla güç sağlamasını gerektirir. Robotik kollar sıkça yüksek momentlere maruz kalır, bu yüzden mühendisler, motorlar ve eklemler için dayanıklılık artırıcı hesaplamalar yapar. Aşırı moment, aşınma ve hasara yol açabileceğinden, doğru moment hesapları robotun güvenli ve verimli çalışmasını sağlar. Bu hesaplamalar, robotik kol tasarımında kritik rol oynar ve daha güvenilir, uzun ömürlü sistemlerin geliştirilmesini sağlar.

1.9.2. Uygulamalı Bilimlerde Moment Kavramı

Matematikte moment, genellikle bir fonksiyonun veya dağılımın şekli, yayılımı, simetrisi gibi özelliklerini tanımlamak için kullanılan bir kavramdır. Bu kavram, özellikle analiz, istatistik ve olasılık teorisi gibi alanlarda önemli bir yer tutar.

Birinci moment yani ortalama, bir fonksiyonun ya da veri setinin merkezi eğilimini gösterir ve aritmetik ortalamasını temsil eder. Bu gösterge, fonksiyonun genel merkezini belirler ve fonksiyonun değerlerinin ortalamaya olan yakınlıklarını ifade eder. Matematiksel olarak bir $f(x)$ fonksiyonu için birinci moment, bağıntı (2) kullanılarak elde edilir.

$$\mu_1 = \int_{-\infty}^{\infty} xf(x)dx \quad (2)$$

İkinci moment ise varyans olarak bilinir ve veri setinin ne kadar yayıldığını ölçer. Varyans, verilerin ortalamadan ne kadar uzaklaştığını gösterir. Yüksek varyans, verilerin birbirinden uzak olduğunu, düşük varyans ise verilerin birbirine yakın olduğunu ifade eder. Matematiksel olarak varyans birinci momentin karesi olarak ifade edilebilir ve bağıntı (3) kullanılarak hesaplanır.

$$\mu_2 = \sigma^2 = \int_{-\infty}^{\infty} (x - \mu)^2 f(x)dx \quad (3)$$

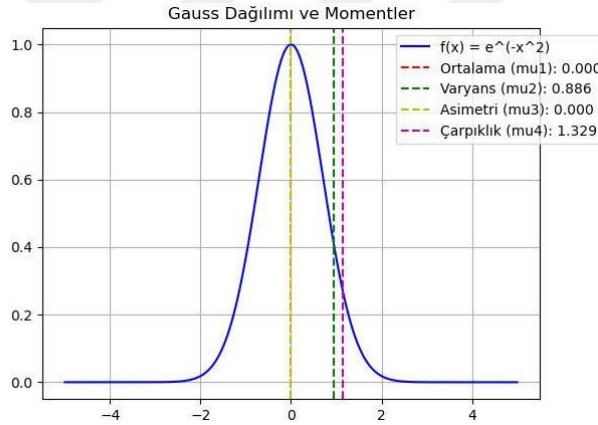
Üçüncü moment, bir fonksiyonun veya veri setinin çarpıklığını tanımlar ve değerlerinin ne kadar sağa veya sola eğilimli olduğunu gösterir. Pozitif çarpıklık, fonksiyonun veya verilerin sağa doğru uzun bir kuyruğa sahip olduğunu belirtirken, negatif çarpıklık ise sola doğru bir kuyruğa işaret eder. Matematiksel olarak üçüncü moment, bağıntı (4)'de gösterildiği gibi ifade edilir.

$$\mu_3 = \int_{-\infty}^{\infty} (x - \mu)^3 f(x) dx \quad (4)$$

Dördüncü moment basıklık olarak bilinir ve bir fonksiyonun veya veri setinin tepe noktası ile uç değerleri hakkında bilgi verir. Yüksek bir basıklık, fonksiyonun veya verilerin daha keskin bir tepeye sahip olduğunu ve uç değerlerin daha belirgin olduğunu gösterir. Matematiksel olarak dördüncü moment, bağıntı (5)'te gösterildiği gibi hesaplanır.

$$\mu_4 = \int_{-\infty}^{\infty} (x - \mu)^4 f(x) dx \quad (5)$$

Şekil 12, $f(x) = e^{-x^2}$ fonksiyonunu ve ilgili fonksiyonun ilk dört momentinin hesaplanmış değerlerini göstermektedir.



Şekil 12. $f(x) = e^{-x^2}$ fonksiyonu ve ilk dört momentinin hesaplanmış değerleri

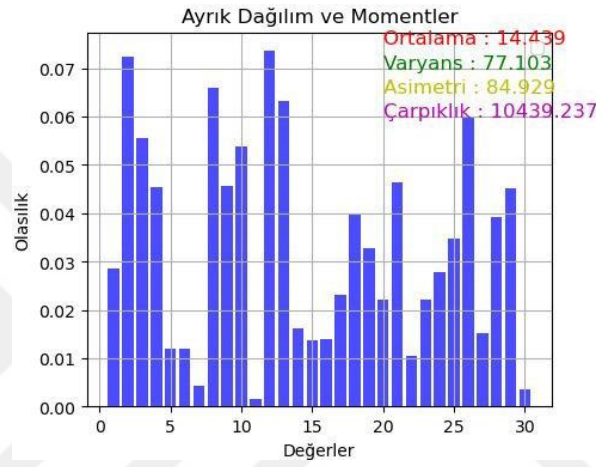
Bir sürekli $f(x)$ fonksiyonu için n. dereceden genelleştirilmiş moment, bağıntı (6)'daki gibi ifade edilir.

$$\mu_n = \int_{-\infty}^{\infty} (x - \mu)^n f(x) dx \quad (6)$$

Ayrık fonksiyonlar için genelleştirilmiş momentler, sürekli dağılımlar için ifade edilen açıklamalara benzer şekilde yapılır, ancak burada integral yerine toplamlar kullanılır. Ayrık bir $f(x)$ fonksiyonu için genelleştirilmiş n. dereceden moment, bağıntı (7)'deki gibi tanımlanır.

$$\mu_n = \sum_x (x - \mu)^n P(x) \quad (7)$$

Burada $P(x)$, olasılık kütlesi fonksiyonu olarak bilinir. Bir başka deyişle, bir rastgele değişkenin belirli bir değeri alması olasılığını gösteren fonksiyondur. Şekil 13, 0-30 değer aralığından rastgele seçilen bir veri kümesinin olasılık dağılım fonksiyonu ve bu dağılım fonksiyonunun ilk dört momentinin hesaplanmış değerlerini göstermektedir.



Şekil 13. Bir olasılık dağılım fonksiyonu ve ilk dört momentinin değerleri

Teorik açıklamaların ardından, farklı uygulamalar üzerinde bu kavramların pratikte nasıl kullanıldığını ele alabiliriz. Örneğin istatistik alanında veri analizi yapmak amacıyla bir eğitim kurumu öğrencilerin sınav sonuçları üzerinde çeşitli istatistiksel momentleri kullanabilir. Bu analizler, öğrencilerin başarı seviyelerini daha iyi anlamak ve iyileştirme stratejileri geliştirmek için faydalıdır. Öncelikle ortalama yani birinci moment, sınıfın genel başarı seviyesini belirlemek için hesaplanır. Ortalama not, tüm öğrencilerin başarı düzeyinin bir göstergesi olarak sınıfın genel performansı hakkında bilgi verir. İkinci moment olan varyans, notların ne kadar yayıldığını gösterir. Elde edilen varyans değeri, eğitimcilerin öğrenci grubu içindeki çeşitliliği anlamalarına yardımcı olur. Çarpıklığı temsil eden üçüncü moment, not dağılımının simetrik olup olmadığını inceleyerek sınıf notlarının nasıl bir dağılım gösterdiği hakkında bilgi verir. Pozitif çarpıklık, daha fazla öğrencinin düşük notlar aldığını, negatif çarpıklık ise daha fazla öğrencinin yüksek notlar aldığını gösterir. Son olarak dördüncü moment olan basıklık, sınav sonuçlarındaki uç değerlerin (çok yüksek veya çok düşük notlar) sıklığını belirler. Yüksek basıklık sınavın zorluk seviyesinin yüksek

olduğunu ve uç değerlerin daha sık görüldüğünü, düşük basıklık notların daha homojen bir şekilde dağıldığını belirtir. Bu momentler, eğitimde başarıyı anlamak ve iyileştirmek için önemli ipuçları sunar.

Bir diğer örnek olarak, bir şirketin yeni bir ürününün tüketici memnuniyetini ölçmek amacıyla gerçekleştirdiği çeşitli istatistiksel analizler verilebilir. İlk adım olarak birinci moment olan ortalama memnuniyet puanı hesaplanarak tüketicilerin genel memnuniyet seviyesi belirlenir ve ürünün genel algısı anlaşılmaya çalışılır. İkinci moment olan varyans, tüketici memnuniyetindeki farklılıkları inceleyerek ürünün farklı demografik gruplar üzerindeki etkisinin değerlendirilmesine olanak sağlar. Bu sayede, ürünün hangi tüketici grupları arasında daha fazla beğenildiği veya daha az ilgi gördüğü anlaşılabilir. Üçüncü moment olan çarpıklık, memnuniyet dağılımının simetrik olup olmadığını gösterir. Bu analiz, bazı tüketici gruplarının üründen daha fazla faydalandığını veya daha fazla memnuniyet sağladığını ortaya koyarak, şirketin hangi hedef kitleye daha fazla odaklanması gerektiğine yardımcı olur. Son olarak dördüncü moment olan basıklık, yüksek ve düşük memnuniyet puanlarının sıklığını değerlendirerek pazarlama stratejilerini daha verimli hale getirmeye olanak tanır. Bu analizler bütünü, tüketici deneyimlerini iyileştirmek için önemli bilgiler sunar ve şirketin pazarlama yaklaşımını optimize etmesine yardımcı olur.

İnşaat mühendisliğinde moment, yapının stabilitesini sağlamak açısından çok önemlidir. Yüklerin kolonlar üzerindeki etkisinin doğru bir şekilde hesaplanması ve her bir kolonun momentlere karşı dayanıklı olacak şekilde tasarlanması gerekmektedir. Kolonların üzerine yük bindikçe, bu yüklerin oluşturduğu momentlerin yapıya etkisi artar ve bu da yapısal sorunlara yol açabilir. Statik denge, momentlerin doğru bir şekilde hesaplanması ve yüklerin dengeli bir şekilde dağıtılması ile sağlanabilir.

1.9.3. Yapay Zeka ve Görüntü İşleme Alanında Moment Analizi

Görüntü analizi, dijital görüntülerden anlamlı bilgiler elde etmek için kullanılan tekniklerin bütünüdür. Bu süreçte momentler görüntülerdeki nesnelerin şekil, yapı ve konum özelliklerini tanımlamada kritik bir rol oynar.

Momentler, bir görüntüdeki piksellerin belirli bir referans noktasına göre ağırlıklandırılmış toplamları olarak tanımlanır ve bu özellikleriyle nesnelerin temel karakteristiklerini etkili bir şekilde belirlemeye olanak tanır.

Momentlerin önemli bir avantajı, görüntülerdeki nesneler döndüğünde veya boyutları değiştiğinde bile tutarlı özellikler sunabilmesidir. Bu özellik, görüntülerin farklı açılardan ve çeşitli boyutlarda etkili bir şekilde analiz edilmesine imkan sağlar (Flusser vd., 2009).

Örneğin yüz tanıma sistemlerinde kullanılan moment tabanlı yöntemler, yüz özelliklerinin çıkarılmasını kolaylaştırarak tanıma oranını artırır. Ayrıca otonom araçlarda çevresel nesnelerin tanınmasında, moment analizi önemli bir rol oynar ve bu sayede araçların çevrelerini daha iyi algılamalarına olanak tanır. Tıbbi görüntüleme alanında ise, hastalıkların teşhisi için görüntülerdeki yapısal özelliklerin analizi momentler kullanılarak gerçekleştirilebilir. Bu yöntem, hastalıkların daha doğru bir şekilde tanımlanmasına ve tedavi süreçlerinin iyileştirilmesine katkı sağlar. Bu örnekler, momentlerin görüntü analizi alanındaki geniş kapsamlı uygulamalarını ve sunduğu avantajları açıkça ortaya koymaktadır.

1.9.4. Görüntü Momentlerinin Tanımı

Görüntü işleme ve desen tanıma alanlarında, momentler bir görüntüdeki şekillerin özelliklerini anlamak ve analiz etmek için kullanılır. Moment tabanlı görüntü temsili, geometrik kararlılık ve dönüşüme bağımsızlık gibi değerli matematiksel nitelikleri sayesinde anlamlandırma için gerekli temel şartları yerine getirmede etkili olduğu vurgulanmıştır (Qi vd., 2021).

Moment tabanlı görüntü temsiliinde ortogonal polinomlar, görüntülerin özelliklerini çıkarmak için yaygın olarak kullanılan güçlü matematiksel araçlardır. Bu polinomlar, görüntü üzerinde tanımlanan ve ortogonalite sağlayan fonksiyonlarla çalışır ve böylece daha az veriyle görüntüdeki bilgileri etkili bir şekilde temsil ederler. Klasik ortogonal polinomlar ve radyal ortogonal fonksiyonlar, matematiksel analizin farklı alanlarda kullanılan iki önemli polinom türüdür. Legendre, Chebyshev ve Jacobi polinomları gibi klasik ortogonal polinomlar, genellikle dikdörtgen bölgelerde tanımlanır ve üç terimli yineleme ilişkisiyle ifade edilir. Bu polinomlar x ve y eksenlerinde ayrı ayrı ortogonalite sağlar ve genellikle gerçek sayılarla çalışır. Radyal ortogonal fonksiyonlar ise polar koordinat sisteminde tanımlanır ve genellikle dairesel bölgelerde kullanılır. Bu fonksiyonlar, hem radyal bileşen $R_{nm}(r)$ ile hem açısız bileşen $e^{-jm\theta}$ ile ifade edilir. Açısız bileşen, karmaşık değerli üstel bir ifade olduğundan radyal ortogonal fonksiyonlar karmaşık değerli veriler içerir. Klasik polinomlar genellikle gerçek sayılarla işlem yaparken, radyal fonksiyonlar döndürmeye

karşı değişmezliği ve karmaşık yapısı ile daha farklı matematiksel ve pratik uygulamalarda kullanılır (Corriou, 2022).

Momentlerin özellikleri ve performanslarının görüntü analizinde kapsamlı bir şekilde incelenmesi ilk olarak Teh ve Chin tarafından gerçekleştirilmiştir. Bu inceleme, ortogonal momentlerle ilgili gelecekte yapılacak tüm araştırmalar için bir temel oluşturmuş ve bir görüntüyü ortogonal polinomlar kullanarak tanımlamanın etkinliğini ortaya koymuştur (Teh & Chin, 1988). Teh ve Chin'in çalışmasının ardından, Zernike momentleri Khotanzad ve Hong tarafından desen tanıma uygulamalarında kullanılmıştır (Khotanzad & Hong, 1990). Ayrıca, görüntü analizi (Chen & Sun, 2010), multimedya damgalama (Kim & Lee, 2003), tıbbi görüntü analizi (Dai vd., 2010), adli bilimler (Mahdian & Saic, 2007) gibi uygulamalara momentlerin ve moment değişmezlerinin uygulanabilirliğinin arttığı gösterilmiştir.

Hacim momentleri, bir görüntüdeki piksel değerlerinin doğrudan kullanılmasıyla elde edilen temel momentlerdir. Bu momentler, görüntünün genel özelliklerini belirlemede yararlıdır. Hacim momentleri, nesnenin konumu ve yoğunluğu hakkında bilgi sağlar fakat dönüş ve ölçek değişikliklerine karşı hassastır. Örneğin, bir nesne döndüğünde veya boyutu değiştiğinde hacim momentlerinin değerleri de değişir. Matematiksel olarak, bir görüntünün hacim momentleri bağıntı (8)'de olduğu gibi ifade edilir. İlgili bağıntıda M_{pq} p ve q derecelerindeki momenti, $f(x, y)$ (x, y) koordinatlarındaki pikselin yoğunluğunu, N ve M ise görüntünün boyutlarını ifade eder.

$$M_{pq} = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} f(x, y) x^p y^q \quad (8)$$

Normalize edilmiş momentler, hacim momentlerinin merkezi momentlere dönüştürülmesiyle elde edilir. Bu momentler, nesnenin boyut ve pozisyon değişikliklerine karşı daha az hassas olup, daha tutarlı sonuçlar sağlar. Normalize edilmiş momentlerin hesaplanmasında, piksel değerlerinin ortalaması ve standart sapması dikkate alınır. Örneğin ikinci normalize edilmiş moment görüntüdeki nesnelerin yayılımını ölçerken, birinci normalize edilmiş moment nesnenin ortalama konumunu belirler. Normalize edilmiş momentler, görüntüdeki şekil ve özelliklerin daha doğru bir şekilde analiz edilmesine olanak tanır. Bu nedenle nesne tanıma, şekil analizi ve sınıflandırma gibi birçok uygulamada tercih edilmektedir. Sonuç olarak normalize edilmiş momentler, görüntü işleme alanında daha sağlam ve güvenilir analizler yapılmasını mümkün kılarak dönüş ve ölçek değişikliklerinden

kaynaklanabilecek olumsuz etkileri en aza indirir. Matematiksel olarak, bir görüntünün normalize edilmiş momentleri bağıntı (9)'daki gibi gösterilir. İlgili bağıntıda μ_{pq} p ve q derecelerindeki normalize edilmiş momenti, M_{pq} p ve q derecelerindeki hacim momenti, M_{00} ise görüntünün toplam yoğunluğunu ifade etmektedir.

$$\mu_{pq} = \frac{M_{pq}}{M_{00}^{(1+\frac{p+q}{2})}} \quad (9)$$

1.9.5. Moment Türleri

Geometrik momentler, bir görüntüdeki şekillerin geometrik yapısını, boyutunu ve yönelimini belirlemek için kullanılır. Bu temel momentler şekillerin alanını, ağırlık merkezini ve diğer temel geometrik dağılımlarını tanımlamak için basit ancak etkili araçlar olarak işlev görür. Öte yandan şekiller döndürüldüğünde, konumu veya boyutları değiştirildiğinde tanıma ve analiz işlemleri daha zor hale gelebilir.

Geometrik momentler, bir görüntüdeki piksel değerlerinin belirli polinomlarla çarpılıp toplanmasıyla hesaplanır. Ancak, kullanılan polinomlar genellikle tek terimli polinomlardır ve bu polinomların birbirleriyle ortogonal olmaması, hesaplanan momentlerin bilgi fazlalığına yol açmasına sebep olur. Bir başka deyişle, aynı veriyi birden fazla momentin temsil etmesi anlamına gelir. İşte bu noktada, Hu momentleri ve ortogonal momentler gibi daha gelişmiş momentler devreye girmektedir (Mukundan & Ramakrishnan, 1998).

1962 yılında Hu, mühendislik alanında iki boyutlu momentleri ilk kez tanıtmıştır. Hu, bir görüntünün iki boyutlu geometrik momentlerini "moment değişmezleri" olarak adlandırmış ve bu yapıyı, doğrusal dönüşümlere yani çevirme, döndürme, ölçekleme ve eğiklik gibi işlemlere karşı değişmeyen bir özellik olarak tanımlamıştır. İlgili çalışmada, cebirsel değişmezler teorisine dayanan yedi ortogonal değişmezi öne sürmüştür (Hu, 1962).

O tarihten itibaren yapılan araştırmalar, momentler ve moment değişmezleri teorisinde önemli ilerlemeler kaydetmiştir. Bu teorik gelişmeler, momentlerin çok farklı alanlarda kullanılmasını mümkün kılmış ve giderek daha fazla araştırmacının bu konuyu çalışmalarının odak noktası haline getirmesine yol açmıştır.

Hu momentlerinin döndürme dayanıklılığı sayesinde bir nesne döndürüldüğünde aynı nesne olarak tanınabilir. Ölçekleme dayanıklılığı, nesnenin boyutları değiştiğinde bile boyut

farkını göz ardı ederek nesneyi tanımaya devam eder. Yer değiştirme dayanıklılığı ise, nesne farklı bir konuma taşındığında bile özelliklerini doğru bir şekilde yakalar. Bu özellikleriyle Hu momentleri, optik karakter tanıma, plaka tanıma ve şekil sınıflandırma gibi uygulamalarda yaygın olarak kullanılır. Örneğin, bir plakanın farklı açılardan fotoğrafı çekildiğinde bile bu gelişmiş momentler sayesinde plaka üzerindeki yazılar aynı şekilde tanınabilir. Bu sayede, şekiller üzerindeki karmaşık dönüşüm ve değişiklikler sırasında bile doğru analizler yapılabilir. Hu momentlerinin kullanım alanları geniştir; nesne tanıma süreçlerinde görüntülerdeki nesnelerin tanınması ve sınıflandırılması için yaygın olarak kullanılırken, mühendislik, robotik ve biyomedikal uygulamalarda nesnelerin şekil özelliklerini analiz etmek için de tercih edilmektedir. Ayrıca, biyolojik görüntüleme hücrelerin ve dokuların analizinde Hu momentleri önemli bir rol oynamaktadır.

Hu momentlerinin hesaplanması, önce görüntüdeki hacim momentlerinin hesaplanması ile başlar. Bu momentler, piksel yoğunlukları kullanılarak belirlenir. Akabinde, hacim momentleri normalleştirilir ve merkezi momentler elde edilir. Hu momentleri, yedi temel normalize edilmiş momentten oluşur ve bu momentler nesnenin geometrik özelliklerini tanımlamak için kullanılır. Ayrıca, görüntüdeki nesnelerin geometrik bilgilerini sağlamanın yanı sıra gürültü ve küçük değişikliklere karşı da oldukça sağlam bir yapıya sahiptirler. Hu momentleri, bağıntı (10)'da gösterildiği gibi hesaplanmaktadır.

$$\begin{aligned}
 \phi_1 &= \mu_{20} + \mu_{02} \\
 \phi_2 &= (\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2 \\
 \phi_3 &= (\mu_{30} - 3\mu_{12})^2 + (3\mu_{21} - \mu_{03})^2 \\
 \phi_4 &= (\mu_{30} + \mu_{12})^2 + (\mu_{21} + \mu_{03})^2 \\
 \phi_5 &= (\mu_{30} - 3\mu_{12})(\mu_{30} + \mu_{12})((\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2) \\
 \phi_6 &= (\mu_{20} - \mu_{02})((\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2) \\
 \phi_7 &= (3\mu_{21} - \mu_{03})(\mu_{30} + \mu_{12})((\mu_{30} - \mu_{12})^2 - (\mu_{21} - \mu_{03})^2)
 \end{aligned} \tag{10}$$

Ortogonal momentler, genel moment hesaplamalarına kıyasla daha yüksek doğruluk ve sıkıştırma kapasitesi sağlayan momentlerdir. Bu momentlerin temel avantajı, çekirdekleri ortogonal polinomlar üzerine inşa edildiği için birbirlerinden bağımsız bileşenler oluşturabilmesidir. Bu ortogonalite, momentlerin minimum bilgi fazlalığına sahip olmasını sağlar; yani her bir moment derecesi, görüntünün farklı bir yönünü temsil eder. Düşük dereceli momentler, bir desenin genel şeklini temsil ederken yüksek dereceli momentler daha ayrıntılı bilgileri sunar.

1980 yılında Teague tarafından tanıtılan ortogonal momentler, görüntü analizi alanında önemli bir gelişme olarak kabul edilir (Teague, 1980). Teague, Legendre ve Zernike ortogonal momentlerini önererek geometrik momentlerin ve Hu değişmezlerinin yüksek bilgi fazlalığı sorununa çözüm bulmayı amaçlamıştır.

Legendre momentleri, iki boyutlu görüntülerin tanımlanması için kullanılan bir başka ortonormal moment çeşididir. Legendre momentleri, görüntü üzerindeki piksel yoğunlukları baz alınarak hesaplanır ve özellikle hızlı hesaplanabilmeleri ve düşük bilgi kaybı nedeniyle tercih edilirler. Bu momentler, belirli bir görüntü üzerindeki piksel verilerini kompakt bir biçimde temsil eder ve bu nedenle sıkıştırma algoritmalarında kullanılabilir. Legendre momentleri, kare veya dikdörtgen simetrilere sahip nesneler üzerinde oldukça etkilidir ve belirli bir simetriye sahip görüntülerin tanınmasında kullanılır.

Legendre momentleri, görüntü yoğunluk fonksiyonu $f(x, y)$ üzerinden hesaplanır ve Legendre polinomları kullanılarak tanımlanır. Hesaplama süreci, belirli bir bölgedeki yoğunluğu temsil eden entegrasyonları içerir. Farklı derecelerde hesaplanabilmesi, daha karmaşık şekil bilgilerini elde etmeyi mümkün kılar, düşük dereceli momentler daha genel ve temel bilgileri sağlar. Ayrıca, görüntü restorasyon süreçlerinde de önemli bir rol oynar; burada görüntülerin yeniden yapılandırılmasında moment bilgisi kullanılır (Zhang vd., 2010).

1.9.6. Zernike Momentleri

Zernike momentleri, bir görüntünün geometrik özelliklerini yüksek hassasiyetle yakalayabilen ve özel polinomlar üzerinden tanımlanan momentlerdir. Bu momentler, özellikle döndürmelere karşı oldukça dayanıklıdır. Bir başka deyişle, bir nesne döndürüldüğünde dahi Zernike momentleri nesnenin tanınmasını mümkün kılar. Bu özellikleri sayesinde, simetrik ya da döndürülmüş yapıların analizi için sıklıkla tercih edilir.

Zernike momentlerinin bir diğer önemli özelliği, nesnenin ince geometrik yapılarını detaylı bir şekilde yakalayabilmesidir. Bu özellik, onları yüksek tanımlama kabiliyetine sahip bir araç yapar. Bu sayede, Zernike momentleri özellikle tıbbi görüntüleme, astronomik veri analizleri ve biyometrik tanıma sistemleri gibi hassasiyet gerektiren alanlarda sıklıkla kullanılır (Zernike & Stratton, 1934).

Zernike momentlerinin duyarlılığı da dikkat çekicidir. Görüntülerdeki gürültüye karşı dayanıklı olmaları, onları çok çeşitli uygulamalarda kullanılabilir hale getirir. Ayrıca,

yüksek dereceli Zernike momentleri, görüntüdeki karmaşıklığı etkili bir şekilde temsil edebilir. Bu durum, görüntülerin daha düşük boyutlu bir temsil ile ifade edilmesi imkanı sunarken veri boyutunun azaltılmasına da yardımcı olur.

Tıbbi görüntüleme alanında, Zernike momentlerinin kullanımı büyük bir rol oynamaktadır. Özellikle tıbbi görüntüleme gibi gürültüye duyarlı alanlarda, MR ve CT taramalarında lezyonlar ya da diğer anormalliklerin tanımlanmasında Zernike momentlerinin katkısı büyüktür. Bu özellikleri, görüntülerdeki yapıları etkili bir şekilde analiz etmeye olanak tanır (Kumar vd., 2018).

1.9.7. Zernike Momentlerinin Hesaplanması

Zernike momentleri ve alt grupları arasında belirgin farklılıklar bulunmaktadır; bu farklılıklar her bir türün kullanım alanını ve uygulama yeteneklerini etkiler. Klasik Zernike momentleri, döndürmeye dayanıklı ve genellikle şekil tanıma uygulamalarında etkili iken, logaritmik Zernike momentleri yüksek dinamik aralığa sahip görüntüleri analiz etmekte kullanılır. Karmaşık Zernike momentleri, hem gerçek hem de sanal bileşenleri içererek iki boyutlu sinyallerin işlenmesinde fayda sağlar. Sürekli Zernike momentleri, pürüzsüz ve sürekli yüzeylerin analizi için tasarlanmıştır. Harmonik Zernike momentleri ise görüntüdeki harmonik bileşenleri analiz etmeye yöneliktir. Her bir tür, belirli bir uygulama veya görüntü özelliğine odaklanarak çeşitli avantajlar ve dezavantajlar sunar.

Zernike momentleri, görüntünün yoğunluk fonksiyonu $f(x, y)$ kullanılarak hesaplanır. Zernike momentleri, bağıntı (11)'de belirtilen formülü ile tanımlanır.

$$Z_n^m = \frac{n+1}{\pi} \int_x \int_y f(x, y) V_{nm}^*(x, y) dy dx \quad (11)$$

Bağıntı (11)'de integral, görüntünün dairesel alanı boyunca hesaplanır. Bir başka deyişle, görüntü üzerinde belirli bir noktanın yoğunluk değeri, Zernike polinomları $V_{nm}(x, y)$ ile çarpılarak toplamsal bir değer elde edilir. Bu hesaplama, görüntünün farklı açı ve ölçeklerdeki özelliklerini ortaya çıkarmaya yardımcı olur ve özellikle dairesel simetrik nesnelerin analizi için idealdir.

Logaritmik Zernike momentleri, Zernike momentlerinin logaritmik modifikasyonu sonucunda elde edilir. Logaritmik Zernike momentleri ile ilgili farklı yaklaşımlar mümkündür ve logaritma işlemi, Zernike momentlerinin hesaplanmasında sadece görüntü

üzerinde değil, Zernike polinomunun kendisi üzerinde de yapılabilir. Belirtilen Logaritmik Zernike momentlerinin Zernike momentlerine uygulanmış modifikasyonu bağıntı (12)'de ifade edilmiştir (Lu & Yang, 2023).

$$Z_n^m = \frac{n+1}{\pi} \int_x \int_y f(x,y) \log(V_{nm}^*(x,y)) dy dx \quad (12)$$

Logaritmik Zernike momentleri, yüksek dinamik aralığa sahip görüntülerin analizi için özel olarak geliştirilmiş bir yöntemdir. Bu momentler, görüntülerin yoğunluk dağılımlarını daha etkili bir şekilde temsil eder ve özellikle zayıf aydınlatma koşullarında veya yüksek kontrast gerektiren durumlarda önemli avantajlar sunar.

1.9.8. Karmaşık Sayılar

Matematiksel problemlerin çözümünde, bazı sayı kümeleri tek başına yeterli olmayabilir. Bu durumda, mevcut sayı kümelerini genişleterek daha kapsamlı sistemler oluşturmak gerekebilir ve buna bağlı olarak farklı sayı kümeleri geliştirilmiştir.

Doğal sayılar (0,1,2,3,...) kümesi, matematiğin temel yapı taşlarından biridir. Pozitif tam sayılardan oluşan bu sınıf, günlük yaşamda sayma işlemlerinde kullanılır aynı zamanda matematiksel işlemlerin temelini oluşturur ve toplama ile çarpma işlemleri bakımından kapalı bir küme olarak tanımlanır. Bir başka ifadeyle, iki doğal sayının toplamı veya çarpımı sonucunda elde edilen sayının yine bir doğal sayı olduğu anlamına gelir. Örneğin, $3+5=8$ ve $4 \times 6=24$ işlemleri, doğal sayıların toplama ve çarpma işlemlerinde kapalı olduğunu gösterir. Ancak çıkarma ve bölme işlemlerinde farklı sınıfların tanımlanmasını gerektirir.

Tam sayılar (...,-3,-2,-1,0,1,2,3,...), doğal sayıları ve negatif tam sayıları içerir. Matematikte çıkarma işlemiyle ilişkili olarak tanımlanan tam sayılar, doğal sayıların eksikliklerini giderir ve simetrik bir yapı sağlayarak temel aritmetik işlemleri genişletir. Bu yapı, hem pozitif hem de negatif değerlerin işlenmesini mümkün kılar ve böylece matematiğin daha geniş uygulama alanlarına hizmet eder.

Rasyonel sayılar, iki tam sayının oranı olarak ifade edilir ve genellikle (a/b) formunda yazılır; burada a ve b tam sayılar olup, $b \neq 0$ koşulu sağlanır. Bu sayıların ondalık açılımları ya sonlu bir değerle bitmekte ya da belirli bir örüntüyü tekrar eden bir yapıya sahip olmaktadır. Rasyonel sayılar, kesirlerin matematiksel ifadesini sağlar ve bölme işlemini

tanımlamada kritik bir rol oynar. Ayrıca, reel sayıların önemli bir alt kümesini oluşturur ve sayısal analizde temel bir yapı taşını temsil eder.

İrrasyonel sayılar, kesir olarak ifade edilemeyen ve ondalık açılımı sonsuza kadar devam eden, ancak herhangi bir örüntü tekrarı göstermeyen sayılardır. Örneğin, $\sqrt{2}$, π ve e gibi sayılar irrasyonel sayıların klasik örneklerindendir. Bu sayılar, matematikte önemli bir yere sahiptir ve özellikle geometrik, trigonometrik ve analitik hesaplamalarda kullanılır. İrrasyonel sayılar, rasyonel sayılarla birlikte reel sayı kümesini tamamlayarak matematiğin temel yapı taşlarından birini oluşturur.

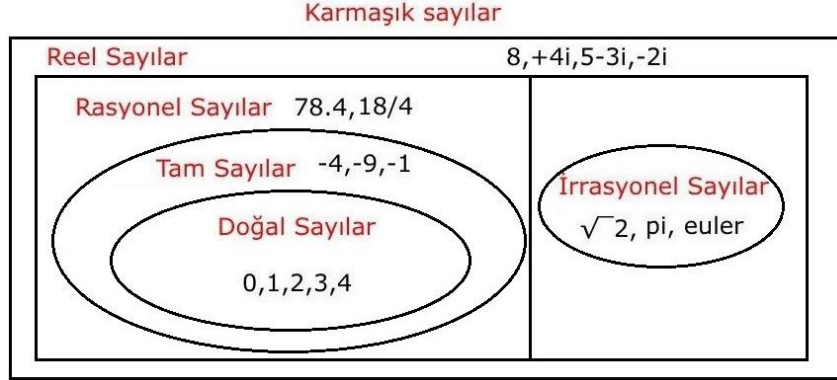
Reel (gerçek) sayılar, rasyonel ve irrasyonel sayıların birleşiminden oluşan ve sayı doğrusu üzerindeki tüm noktaları kapsayan bir sayı kümesidir. Bu sınıf, reel analiz, trigonometri ve geometri gibi matematiğin pek çok dalında temel bir yapı taşıdır. Reel sayılar, sürekli bir sayı doğrusu kavramı sunarak kesintisiz bir sayı sistemi sağlar ve bu da analitik hesaplamaların gerçekleştirilmesine olanak tanır. Bu özellikleri sayesinde reel sayılar, hem teorik hem de uygulamalı matematikte kritik bir öneme sahiptir.

Karmaşık sayılar, gerçek ve sanal bileşenlerin birleşiminden oluşmakta ve genellikle $z = x + iy$ formunda ifade edilmektedir; burada a gerçek kısmı, bi ise sanal kısmı temsil eder ve i , $i^2 = -1$ özelliğine sahiptir. Bu sayı sistemi, reel sayıların da dışında matematikte daha genel bir yapı sağlar ve özellikle cebir, mühendislik, fizik ve sinyallerin analizi gibi alanlarda kullanılmaktadır. Karmaşık sayılar, iki boyutlu bir düzlemde görselleştirilebilme özelliği sayesinde, hem geometrik hem de analitik problemlerin çözümünde güçlü bir araç olarak ortaya çıkmaktadır. Bu nedenle, matematiksel analizden diferansiyel denklemlere kadar geniş bir uygulama alanında temel yapı taşıdır. Örneğin, $x^2 + 1 = 0$ denkleminin reel sayılar arasında çözümü yoktur, çünkü hiçbir reel sayının karesi negatif bir değer almaz. Böylece $x^2 + 1 = 0$ denkleminin çözümleri karmaşık birim i ve $-i$ olarak tanımlanır; burada i sanal birim olup $x^2 = -1$ özelliğine sahiptir.

Karmaşık sayılar özellikle dalga hareketleri, elektromanyetik alanlar, kuantum mekaniği ve elektronik gibi alanlarda önemli bir rol oynamaktadır. Elektrik devrelerinde alternatif akımın modellenmesi veya dalgaların sinüzoidal hareketi gibi olaylar, en iyi şekilde karmaşık sayılarla ifade edilir. Ayrıca, dönme, titreşim ve salınımlar gibi olayların matematiksel olarak modellenmesinde de karmaşık sayılar büyük kolaylık sağlar. Bu sayede, karmaşık sayıların gerçek dünyadaki etkileri daha net bir şekilde gözlenmektedir. Karmaşık analiz ve Fourier dönüşümleri gibi matematiksel teknikler de karmaşık sayılar kullanılarak geliştirilmiş olup, buna bağlı olarak sinyal işleme ve sistem analizi gibi alanlarda önemli

avantajlar sağlanmıştır. Sonuç olarak karmaşık sayılar, reel sayılar kümesinin ötesine geçerek daha geniş bir çözüm alanı sunar ve matematiksel denklemlerin çözümünü daha kapsamlı bir şekilde mümkün kılar.

Şekil 14 sayı kümelerini, birbirlerini kapsama gibi aralarındaki ilişkiler dahil olmak üzere bir arada göstermektedir.



Şekil 14. Sayı Kümeleri

İki karmaşık sayı $u = a + ib$ ve $v = c + id$ olsun. Sıfırdan farklı her karmaşık sayı, birinci dereceden bir polinomdur. Karmaşık eşlenik, bir karmaşık sayının gerçek kısmını aynı bırakırken, sanal kısmının işaretini değiştirerek elde edilen karmaşık sayıdır. $u = a + ib$ sayısının eşleniği, $\bar{u} = u^* = a - ib$ şeklinde gösterilir.

Karmaşık sayıların toplamı gerçek kısımların toplamı ve sanal kısımların toplamı olarak ifade edilir. Karmaşık sayıların farkı, benzer şekilde gerçek kısmın farkı ve sanal kısmın farkı olarak ifade edilir. İlgili işlemler bağıntı (13) gösterildiği şekilde gerçekleştirilir.

$$\begin{aligned} u + v &= (a + ib) + (c + id) = (a + c) + i(b + d) \\ u - v &= (a + ib) - (c + id) = (a - c) + i(b - d) \end{aligned} \quad (13)$$

Karmaşık sayıların çarpımı ise dağılım özelliği kullanılarak yapılmaktadır. İlgili işlemler bağıntı (14)'de gösterilmiştir.

$$\begin{aligned} uv &= (a + ib)(c + id) \Rightarrow a(c + id) + ib(c + id) \Rightarrow ac + iad + ibc + i^2bd \\ i^2 &= -1 \text{ olduğundan } uv = ac + iad + ibc - bd \Rightarrow (ac - bd) + i(ad + bc) \end{aligned} \quad (14)$$

Nihayetinde karmaşık sayıların çarpımı, gerçekte kısmı $(ac - bd)$ ve sanal kısmı $(ad + bc)$ olan bir karmaşık sayıdır. Karmaşık sayıların bölümü, pay ve paydanın eşlenikleri kullanılarak gerçekleştirilir. İlgili işlemler bağıntı (15)'de gösterilmiştir.

$$\begin{aligned} \frac{u}{v} = \frac{a + ib}{c + id} &\Rightarrow \frac{(a + ib)(c - id)}{(c + id)(c - id)} \Rightarrow \frac{ac + bd + i(bc - ad)}{c^2 + d^2} \\ &\Rightarrow \frac{ac + bd}{c^2 + d^2} + i \frac{bc - ad}{c^2 + d^2} \end{aligned} \quad (15)$$

Bir karmaşık sayının büyüklüğü, orijin ile olan uzaklık olarak tanımlanmaktadır ve $z = x + iy$ karmaşık sayısı için $|z|$ şeklinde gösterilir. İlgili büyüklük değeri bağıntı (16) kullanılarak hesaplanır.

$$r = |z| = \sqrt{x^2 + y^2} \quad (16)$$

Argüman ise karmaşık sayının pozitif gerçekte eksen ile yaptığı açıyı ifade eder. Bu iki özellik (büyüklük ve argüman), karmaşık sayıların kutupsal formda gösterilmesini mümkün kılar ve çeşitli mühendislik uygulamalarında kullanılır. Bağıntı (17), argümanın elde edilmesine yönelik formülü ve bir karmaşık sayının kutupsal formda temsilini ifade etmektedir.

$$\begin{aligned} \theta = \arg(z) &= \tan^{-1} \left(\frac{y}{x} \right) \\ z &= r(\cos \theta + i \sin \theta) \end{aligned} \quad (17)$$

Karmaşık değişkenli fonksiyonlar, karmaşık sayıların bir fonksiyonudur ve gerçekte fonksiyonların karmaşık sayı uzayına genişletilmiş halidir. Karmaşık fonksiyonlar, analitik özellikleri, türevlenebilirlik koşulları ve farklı davranışları ile karmaşık analizde önemli bir yer tutar. Karmaşık değişkenli fonksiyon genellikle eğer $f(z): \mathbb{C} \rightarrow \mathbb{C}, z = x + iy$ ise $\exists u, v: \mathbb{R}^2 \rightarrow \mathbb{R}$ olmak üzere bağıntı (18)'de gösterildiği gibi ifade edilir.

$$f(z) = f_{(x+iy)} = u(x, y) + iv(x, y) \quad (18)$$

İlgili bağıntıda $u(x, y)$ fonksiyonun gerçek, $v(x, y)$ ise sanal kısmına karşılık gelirken her ikisi de x ve y 'nin bir fonksiyonu olarak tanımlanır.

Karmaşık analizde diferansiyel kuralı, karmaşık diferansiyel tanımını ifade etmektedir. Hem karmaşık değişken z hem de z 'nin karmaşık eşleniği z^* ile ilgili türevleri içeren karmaşık bir fonksiyonun diferansiyeli ∂f , bağıntı (19)'da ifade edilmiştir.

$$\partial f = \frac{\partial f}{\partial z} \partial z + \frac{\partial f}{\partial z^*} \partial z^* \quad (19)$$

Karmaşık analizde türevlerin karmaşık eşlenikleri bağıntı (20)'de verilmiştir. Bu tür türev özellikleri, karmaşık fonksiyon teorisinde, Cauchy-Riemann denklemlerinin analizinde ve holomorfik fonksiyonların incelenmesinde yaygın olarak kullanılır.

$$\left(\frac{\partial f}{\partial z}\right)^* = \frac{\partial f^*}{\partial z^*}, \quad \left(\frac{\partial f}{\partial z^*}\right)^* = \frac{\partial f^*}{\partial z} \quad (20)$$

Eğer $f(z): \mathbb{C} \rightarrow \mathbb{R}$ ise türevlerin karmaşık eşlenikleri bağıntı (21)'deki gibi olmaktadır.

$$\left(\frac{\partial f}{\partial z}\right)^* = \frac{\partial f}{\partial z^*}, \quad \left(\frac{\partial f}{\partial z^*}\right)^* = \frac{\partial f}{\partial z} \quad (21)$$

Holomorfik bir fonksiyon, bir veya daha fazla karmaşık değişkene bağlı olan ve tanım kümesinin her noktasında karmaşık türevlenebilir olan bir karmaşık değerli fonksiyondur. Holomorfik fonksiyonlar aynı zamanda analitik fonksiyon olarak adlandırılır ve Cauchy-Riemann denklemleri sağlanır.

$f(z): \mathbb{C} \rightarrow \mathbb{C}$ fonksiyonu ve $\Omega \subset \mathbb{C}$ açık bir alt kümesinde karmaşık türevi bağıntı (22)'deki limitin var olmasıyla tanımlanır. İlgili bağıntıda $\Delta z = \Delta x + i\Delta y$ şeklindedir ve $\Delta z \rightarrow 0$ hem $\Delta x \rightarrow 0$ hem de $\Delta y \rightarrow 0$ olur.

$$f'(z) = \lim_{\Delta z \rightarrow 0} \frac{f(z + \Delta z) - f(z)}{\Delta z} \quad (22)$$

Bu limitin var olabilmesi için Δz farklı yollarla sifıra yaklaşırsa bile aynı sonucu vermelidir. Yani hem gerçek eksen ($\Delta y = 0$) hem de sanal eksen ($\Delta x = 0$) boyunca aynı olmalıdır.

Gerçek eksen boyunca türev ($\Delta y = 0$) bağıntı (23)'de verilmektedir.

$$\frac{\partial f}{\partial z} = \frac{\partial u}{\partial x} + i \frac{\partial v}{\partial x} \quad (23)$$

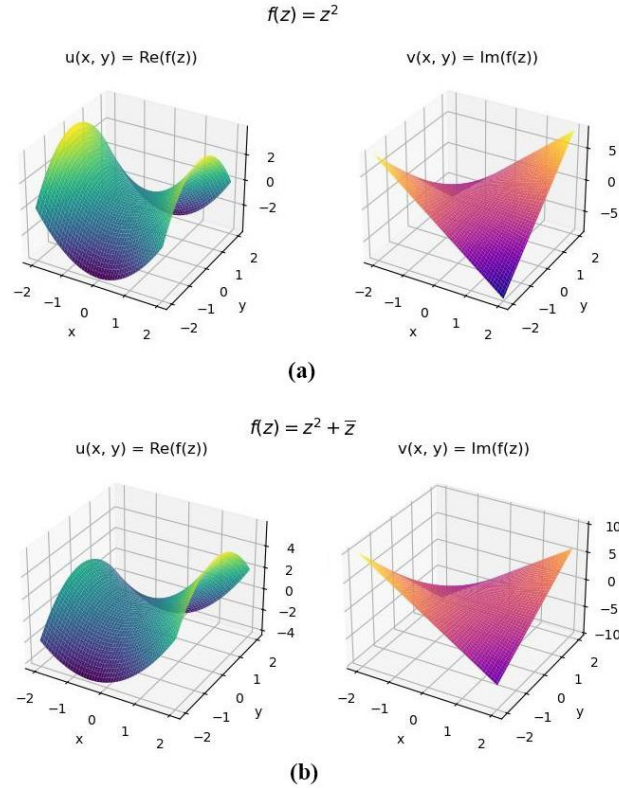
Sanal eksen boyunca türev ($\Delta x = 0$) bağıntı (24)'de verilir.

$$\frac{\partial f}{\partial z} = \frac{\partial v}{\partial y} - i \frac{\partial u}{\partial y} \quad (24)$$

Bu iki türev eşitliğinden Cauchy-Riemann denklemleri bağıntı (25)'de ifade edilir.

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y}, \quad \frac{\partial u}{\partial y} = -\frac{\partial v}{\partial x} \quad (25)$$

Bu durumda, $\partial f / \partial z^* = 0$ olmalıdır. $f(z)$ fonksiyonunun holofomik bir örneği Şekil 15. (a)'da, holofomik olmayan ise Şekil 15. (b)'de gösterilmektedir.



Şekil 15. (a) $f(z)$ fonksiyonunun holofomik bir örneği (b) $f(z)$ fonksiyonunun holofomik olmayan bir örneği

Wirtinger hesaplaması, Wilhelm Wirtinger tarafından 1927 yılında ortaya atılan ve karmaşık türev kavramını genelleştiren bir yöntemdir. Wirtinger hesabı, özellikle karmaşık analiz, diferansiyel geometri ve sinyal işleme gibi alanlarda sıkça kullanılır. Bu hesaplama, karmaşık fonksiyonların türevini ve gradyanını hesaplamak için geliştirilmiştir ve özellikle karmaşık sayılarla çalışan fonksiyonların optimizasyonunda kullanılır. Wirtinger hesaplamasının temel avantajı, karmaşık değerli fonksiyonlarla çalışırken türev almayı daha sistematik ve verimli hale getirmesidir.

Wirtinger hesaplaması, holomorfik olmayan fonksiyonlarla çalışmayı mümkün kılar ve aynı zamanda gradyan hesaplaması için alternatif bir yöntem sunar. Bu yöntem, eğitim sürecinin kararlılığını da artırarak daha güvenilir sonuçlar elde edilmesine yardımcı olur (Fischer, 2002).

Wirtinger hesaplamasında karmaşık fonksiyonların türevini almak için özel türev operatörleri kullanılır. Wirtinger operatörleri olarak adlandırılan bu operatörler, karmaşık sayıların gerçekte ve sanal kısımlarına bağlıdır. Wirtinger operatörleri bağıntı (26)'da ifade edilmektedir.

$$\frac{\partial}{\partial z} = \frac{1}{2} \left(\frac{\partial}{\partial x} - i \frac{\partial}{\partial y} \right) , \quad \frac{\partial}{\partial z^*} = \frac{1}{2} \left(\frac{\partial}{\partial x} + i \frac{\partial}{\partial y} \right) \quad (26)$$

Karmaşık değerli değişkenlerle çalışan sinir ağlarında, optimizasyon süreçlerinde kullanılan özel bir zincir kuralı önemli bir rol oynamaktadır. Bu kural, değişkenler karmaşık olsa bile maliyet fonksiyonunun her zaman gerçekte bir değer almasını sağlayarak işlemleri daha verimli hale getirir. Karmaşık değişkenler için türev almayı genelleştiren bu yöntem, hem teorik hem de pratik uygulamalarda kritik bir öneme sahiptir (Simon, 2014).

$f(z): \mathbb{C} \rightarrow \mathbb{C}$, $z \in \mathbb{C}$ ve $x(z), y(z): \mathbb{C} \rightarrow \mathbb{R}$ için gerçekte ve sanal kısımlar üzerinde karmaşık zincir kuralı bağıntı (27)'de ifade edilir.

$$\frac{\partial f}{\partial z} = \frac{\partial f}{\partial x} \frac{\partial x}{\partial z} + \frac{\partial f}{\partial y} \frac{\partial y}{\partial z} \quad (27)$$

$f, g: \mathbb{C} \rightarrow \mathbb{C}$ fonksiyonlardır ve $z \in \mathbb{C}$ bir karmaşık değişkendir. Buna bağlı olarak, karmaşık zincir kuralı bağıntı (28)'de ifade edilmektedir. Burada f fonksiyonu g 'ye, g fonksiyonu da z 'ye bağlıdır.

$$\frac{\partial h(g)}{\partial z} = \frac{\partial f}{\partial g} \frac{\partial g}{\partial z} + \frac{\partial f}{\partial g^*} \frac{\partial g^*}{\partial z} \quad (28)$$

$g(z)$ 'nin z, z^* ile türevi ayrı ayrı hesaplanır. Aynı durum g^* için de geçerli olmaktadır. İlgili formüller bağıntı (29)'da verilmektedir.

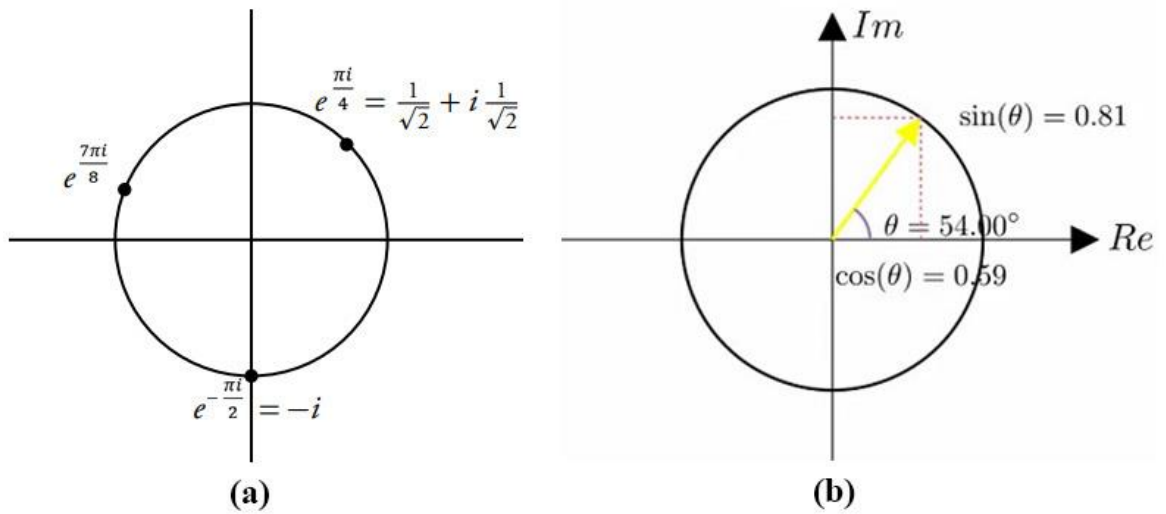
$$\frac{\partial g}{\partial z} = g'(z) \quad , \quad \frac{\partial g}{\partial z} = (g^*)'(z) \quad , \quad \frac{\partial g}{\partial z} = \frac{\partial g}{\partial x} \frac{\partial x}{\partial z} + \frac{\partial g}{\partial y} \frac{\partial y}{\partial z} \quad (29)$$

Karmaşık sayılar genel olarak dikdörtgensel, kutupsal (polar) ve üstel formda olmak üzere sırasıyla $z = x + iy$, $z = r(\cos \theta + i \sin \theta)$ ve $z = re^{i\theta}$ şeklinde ifade edilir.

Euler formülü, karmaşık analizde hem üstel fonksiyonların hem de trigonometrik fonksiyonların temel bir birleşimini sağlayarak, bu iki önemli matematiksel yapı arasında güçlü bir ilişki kurar. İlgili formül ve açılımı bağıntı (30)'da ifade edilmektedir.

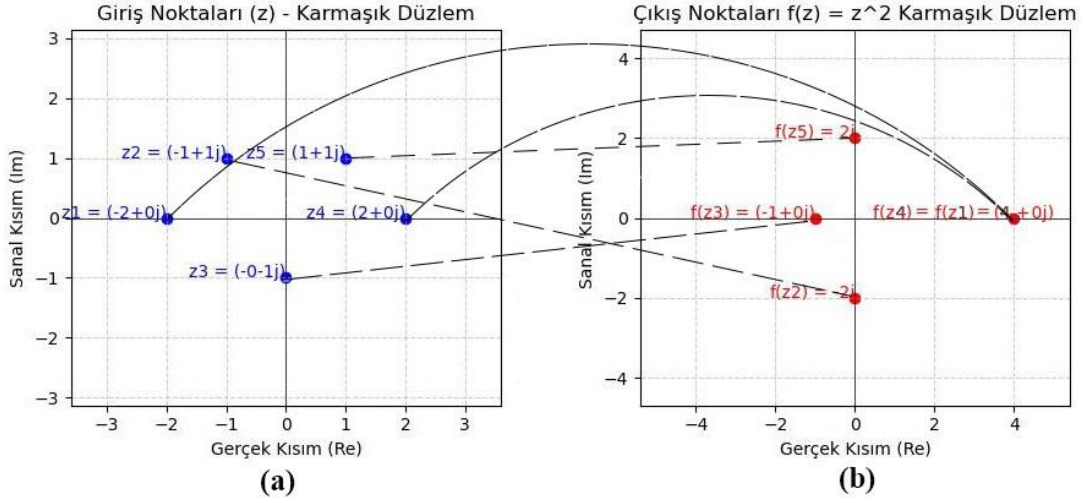
$$e^{i\theta} = \cos \theta + i \sin \theta \quad (30)$$

Bu formül, karmaşık sayılar düzleminde üstel fonksiyonların döndürme ve büyüklük ilişkilerini sinüs ve kosinüs fonksiyonlarıyla ifade edebileceğimizi gösterir. Euler formülünün, Şekil 16. (a)'da birim çember üzerindeki temsili, Şekil 16. (b)'de ise trigonometrik gösterimi belirtilmiştir.



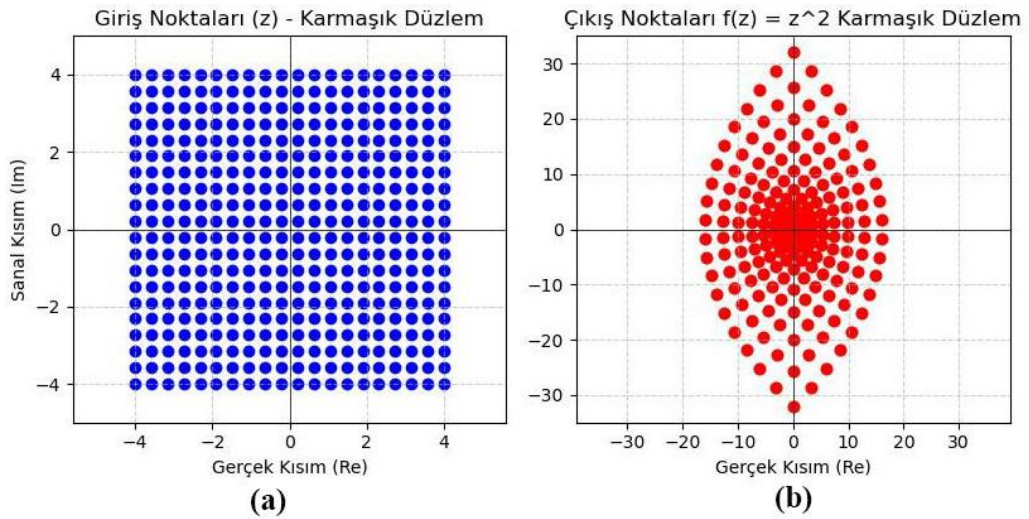
Şekil 16. (a) Euler formülünün birim çember üzerindeki temsili (b) Euler formülünün, trigonometrik gösterimi

Karmaşık düzlem, karmaşık sayıları görselleştirmek ve üzerinde işlem yapmak için kullanılan iki boyutlu bir yüzeydir. Bu düzlemde her karmaşık sayı bir nokta olarak temsil edilir ve düzlemdeki yatay eksen karmaşık sayıların gerçekte kısmını, dikey eksen ise sanal kısmını gösterir. Şekil 17. (a) karmaşık düzlemde verilen beş karmaşık sayıyı, Şekil 17.(b) ilgili beş karmaşık sayının $f(z) = z^2$ fonksiyonuna göre dönüşümünü göstermektedir.



Şekil 17. (a) Karmaşık düzlemde verilen beş karmaşık sayı (b) İlgili beş karmaşık sayının $f(z) = z^2$ fonksiyonuna göre dönüşümünü

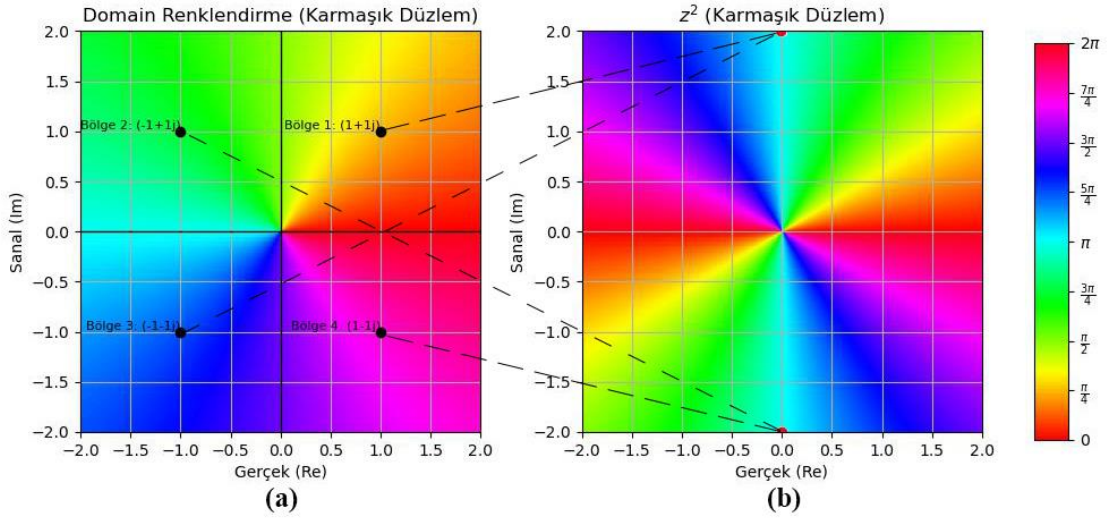
Şekil 18. (a) ayrıık noktalardan oluşan karesel bir görüntüyü, Şekil 18. (b) ise ilgili ayrıık karesel görüntünün $f(z) = z^2$ fonksiyonuna göre dönüşümünü belirtir.



Şekil 18. (a) Ayrıık noktalardan oluşan karesel bir görüntü (b) İlgili görüntünün $f(z) = z^2$ fonksiyonuna göre dönüşümünü

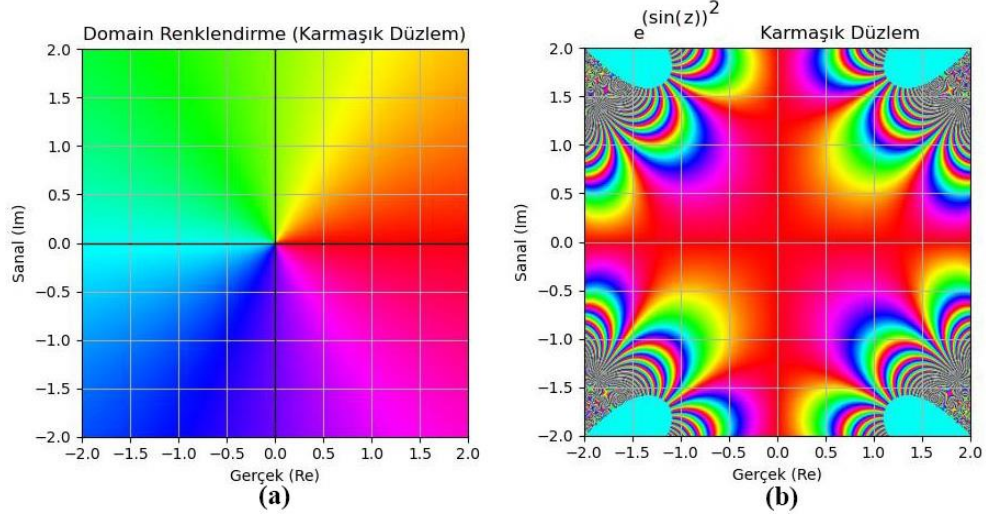
Domain renklendirme yöntemi, karmaşık fonksiyonların görselleştirilmesinde kullanılan bir tekniktir. Bu yöntem, karmaşık bir fonksiyonun değerlerini karmaşık düzlemdeki her bir noktaya renk atayarak temsil eder. Böylece karmaşık fonksiyonların davranışları, özellikle genlik ve faz bilgisi açısından daha anlaşılır hale gelir. Bu görselleştirme yöntemi, karmaşık sayılarla çalışan fonksiyonların analizini kolaylaştırmak için oldukça etkilidir.

Şekil 19. (a) renklendirilmiş $(0 - 2\pi)$ aralığında karmaşık düzlemde verilen dört karmaşık sayıyı, Şekil 19. (b) ise ilgili karmaşık sayıların ifade edilen düzlemde $f(z) = z^2$ fonksiyonuna göre dönüşümünü göstermektedir.



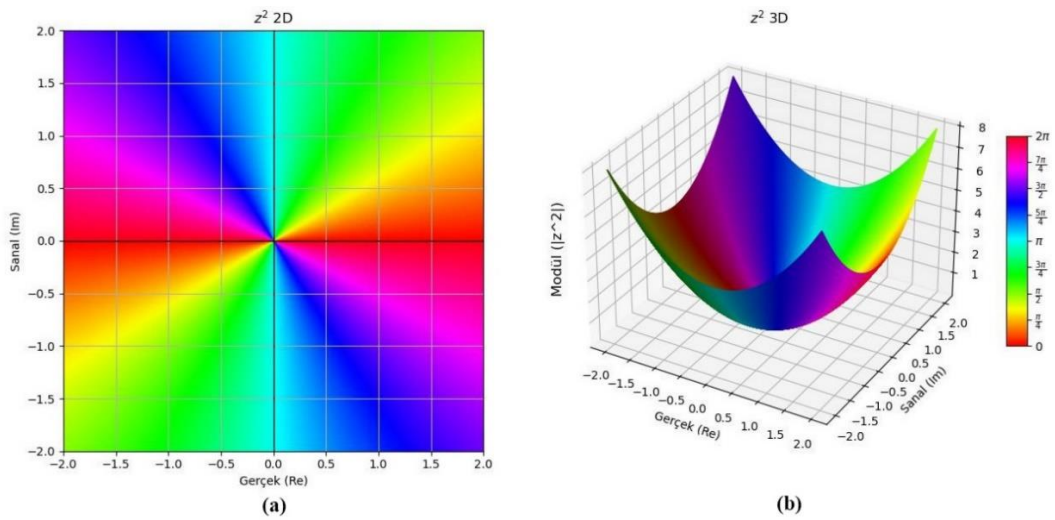
Şekil 19. (a) Renklendirilmiş karmaşık düzlemde verilen dört karmaşık sayı (b) İlgili karmaşık sayıların ifade edilen düzlemde $f(z) = z^2$ fonksiyonuna göre dönüşümünü

Şekil 20. (a) renklendirilmiş karmaşık bir düzlemi, Şekil 20. (b) ise belirtilen düzlemin $f(z) = e^{(\sin(z))^2}$ fonksiyonuna göre dönüşümünü göstermektedir. Karmaşık düzlemin üzerindeki her noktada, sinüs fonksiyonu dalgalanma yaratır. Bu durum, karmaşık düzlemde çok katmanlı ve tekrarlı bir dalga yapısı meydana getirir. Sinüs değerinin karesi, dalgalanmanın büyüklüğünü daha da artırır ve simetrik bir yapı oluşturur. Eksponansiyel fonksiyon ise bu dalgalanmaları şiddetlendirir ve karmaşık düzlemde renklerde yoğunlaşma, simetri ve fraktal benzeri yapılar meydana getirir. Düzlemin genelinde, fonksiyonunun etkisiyle renkler düzenli aralıklarla dalgalanmış ve tekrarlayan simetrik desenler oluşturmuştur. Bu durum, fonksiyonun büyüklük ve faz üzerinde eş zamanlı etkisiyle ortaya çıkmıştır.



Şekil 20. (a) Renklendirilmiş karmaşık bir düzlem (b) Belirtilen düzlemin $f(z) = e^{(\sin(z))^2}$ fonksiyonuna göre dönüşümünü

3D gösterim, verilerin veya fonksiyonların üç boyutlu bir uzayda görselleştirilmesidir. Bu görselleştirme, özellikle karmaşık fonksiyonların, yüzeylerin veya dinamik sistemlerin analizinde oldukça etkilidir. Üçüncü boyutun kullanımı, verilerdeki gizli desenlerin veya özelliklerin daha belirgin hale gelmesini sağlar. Böylece 3D gösterim, verilerin ve ilişkilerin daha ayrıntılı olarak incelenmesini ve anlaşılmasını kolaylaştırır. Şekil 21. (a) renklendirilmiş karmaşık bir düzlemin $f(z) = z^2$ fonksiyonuna göre dönüşümünü, Şekil 21. (b) ise ilgili düzlemin $f(z) = z^2$ fonksiyonuna göre dönüşümünü 3D olarak göstermektedir.



Şekil 21. (a) Renklendirilmiş karmaşık bir düzlemin $f(z) = z^2$ fonksiyonuna göre dönüşümü (b) İlgili düzlemin $f(z) = z^2$ fonksiyonuna göre dönüşümünün 3D olarak gösterilmesi

1.9.9. Karmaşık Zernike Momentleri

Karmaşık Zernike momentleri, hem gerçek hem de sanal bileşenleri bir araya getirerek iki boyutlu sinyallerin işlenmesinde önemli avantajlar sunar. Bu momentler, geleneksel Zernike momentlerine göre daha zengin bilgi sağlar ve görüntülerin karmaşık yapısını daha iyi yansıtmaya kapasitesine sahiptir.

Karmaşık Zernike momentlerinin gerçek bileşeni görüntünün yoğunluk değerlerini temsil ederken, sanal bileşen görüntüdeki faz bilgilerini içerir. Bu iki bileşenin birleşimi, nesnelerin şekilsel ve yapısal özelliklerini daha kapsamlı bir şekilde değerlendirmeyi mümkün kılar. Bu özellikleri sayesinde karmaşık Zernike momentler, özellikle tıbbi görüntüleme, yüz tanıma ve endüstriyel görüntü analizi gibi alanlarda, karmaşık şekil ve yapıların incelenmesinde değerli bir araç haline gelmektedir. Karmaşık Zernike momentleri, görüntülerin daha iyi tanınması ve sınıflandırılmasına katkıda bulunarak, görüntü işleme alanında önemli bir gelişme sağlamaktadır.

Diğer yöntemlerle karşılaştırıldığında, moment formülasyonu, gürültüye dayanıklılığı, bilgi yoğunluğu ve yeniden yapılandırma yeteneği açısından üstün performans gösterdiği için daha popülerdir (Teh & Chin, 1988). Mevcut yöntemlerin çoğu, tanıma sürecinde değişmeyen özellikler olarak yalnızca momentlerin büyüklük bileşenini kullanmaktadır. Bununla birlikte, Singh ve ekibi, görüntü temsili için faydalı bilgiler sunan momentlerin faz bileşenini vurgulamaktadır. Çalışmalarında, büyüklük ve faz katsayılarının performansını kapsamlı bir şekilde analiz etmişlerdir (Singh vd., 2011). Li ve ekibi ise Zernike momentlerinin faz bileşeninin, görüntü yeniden yapılandırmasında önemli bilgiler içerdiğini belirlemektedir. Bu nedenle, hem büyüklük hem de faz katsayılarını birleştirerek yeni bir şekil betimleyicisi oluşturmayı önermektedirler (Li vd., 2009).

Karmaşık Zernike momentleri, bağıntı (31)'de olduğu gibi ifade edilir.

$$A_n^m = \frac{n+1}{\pi} \int_0^{2\pi} \int_0^1 f(r, \theta) V_{nm}^*(r, \theta) r dr d\theta \quad (31)$$

$N \times N$ boyutundaki bir ayrık görüntü için, n (0, 1, 2, ..., ∞ olabilir) derecesindeki ve m (pozitif ya da negatif tam sayı olabilir) tekrarlarındaki CZM'ler bağıntı (32)'de gösterildiği gibi hesaplanmaktadır.

$$A_{nm} = \frac{n+1}{\pi} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} f(x_i, y_j) V_{nm}^*(x_i, y_j) \quad (32)$$

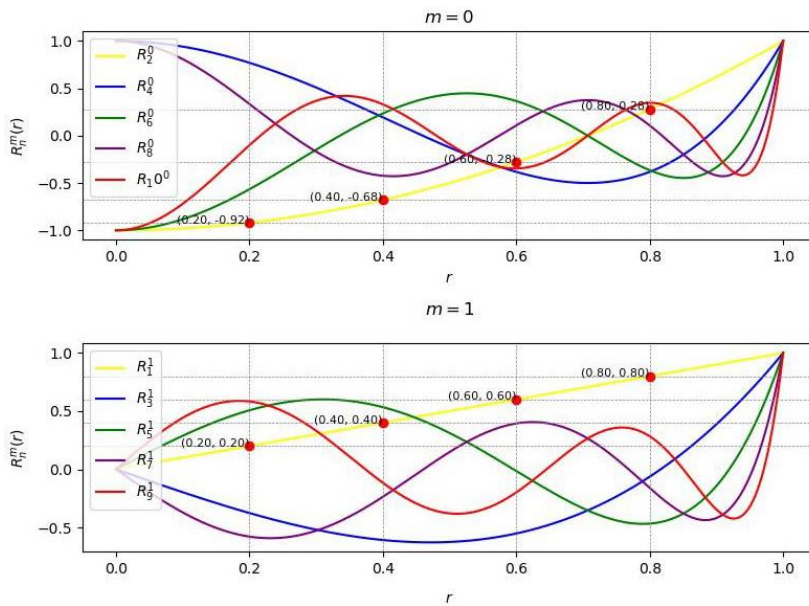
$$x_i = \frac{2i - (N-1)}{N-1}, \quad y_j = \frac{2j - (N-1)}{N-1}$$

(32) bağıntısında $|m| \leq n$ ve $n - |m|$ çift olmalıdır. $f(x_i, y_j)$ görüntünün yoğunluğunu ifade eder ve orijinal koordinatlar birim disk içine uyacak şekilde ölçeklendirilmiştir, yani $x_i^2 + y_j^2 \leq 1$. Ayrıca, * karmaşık eşlenik anlamına gelir ve Zernike polinomu $V_{nm}(x_i, y_j)$, polar koordinatlarda bağıntı (33)'de $V_{nm}(r, \theta)$ olarak ifade edilir.

$$V_{nm}(r, \theta) = R_{nm}(r) e^{-jm\theta} \quad (33)$$

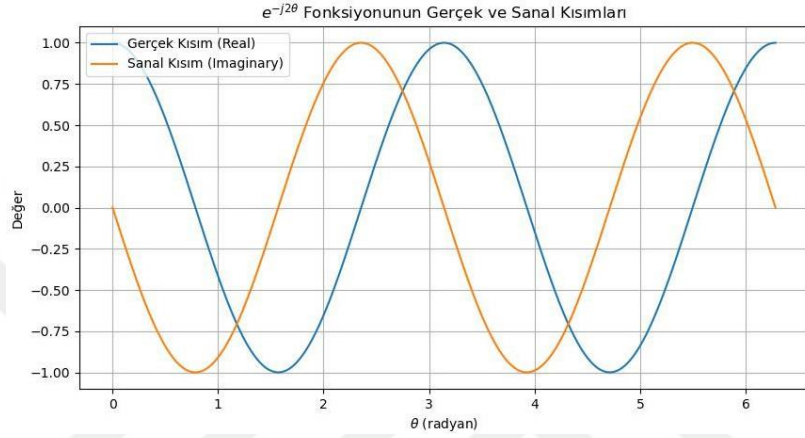
Bağıntı (33)'de, $0 \leq r \leq 1$ ve j sanal birim olmak üzere, ortogonal radyal polinom $R_{nm}(r)$ bağıntı (34)'de gösterildiği gibi ifade edilir. Şekil 22, Zernike radyal polinomlarının ($m = 0, 1$) için ilk beş derecesini göstermektedir.

$$R_{nm}(r) = \sum_{s=0}^{\frac{n-|m|}{2}} (-1)^s \frac{(n-s)!}{s! \left(\frac{n+|m|}{2} - s\right)! \left(\frac{n-|m|}{2} - s\right)!} r^{n-2s} \quad (34)$$



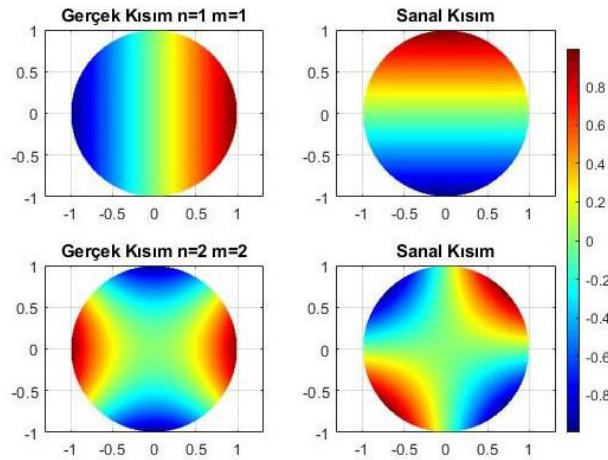
Şekil 22. Zernike radyal polinomlarının ($m = 0, 1$) için ilk beş derecesi

Şekil 23, bağıntı (33)'deki $e^{-jm\theta}$ açısıl bileşeninin $m = 2$ için sinüs ve kosinüs grafiklerini ve karmaşık düzlemdeki dönüşlerin trigonometrik temelde nasıl ifade edildiğini gösterir. İlgili şekilde, gerçek eksen (real) üzerinde kosinüs bileşeni, sanal eksen (imaginary) üzerinde ise sinüs bileşeni yer alır. θ açısının değişimine bağlı olarak kosinüs bileşeni yatayda, sinüs bileşeni ise dikeyde salınım gösterir.



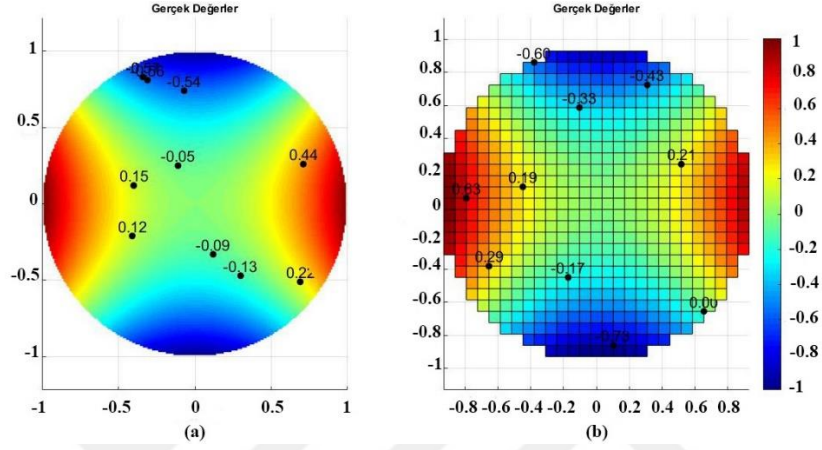
Şekil 23. Bağıntı (33)'deki $e^{-jm\theta}$ açısıl bileşeninin $m = 2$ için sinüs ve kosinüs grafikleri

Şekil 24, $V_{nm}(r, \theta)$ Zernike polinomunun farklı n ve m değerleri için hesaplanmış halinin kutupsal formda gerçek ve sanal bileşenlerini renk kodlaması ile göstermektedir. Böylece, Zernike polinomunun geometrik yapısı ve açısıl frekanslarına bağlı salınım özellikleri daha kolay anlaşılır.



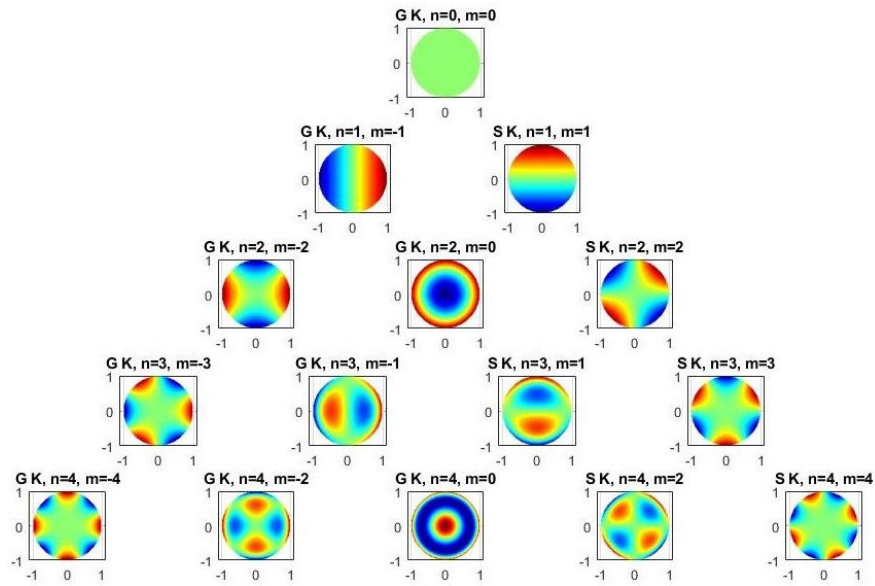
Şekil 24. $V_{nm}(r, \theta)$ Zernike polinomunun farklı n ve m değerleri için hesaplanmış halinin kutupsal formda gerçek ve sanal bileşenlerini renk kodlaması

Şekil 25. (a), $V_{nm}(r, \theta)$ Zernike polinomunun ($n = 2, m = 2$ için) kutupsal olarak renklendirilmiş formu üzerinde rastgele seçilen noktaların sayısal karşılıklarını içerir. Şekil 25. (b) ise, aynı noktaların ayırık bir görüntüde grid yapısında nasıl konumlandığını gösterir. İlgili grid yapısı, Zernike polinomunun her bir noktası için ayırık sayısal değerlerin görsel olarak nasıl düzenlendiğini daha net bir şekilde ortaya koymaktadır.



Şekil 25. (a) $V_{nm}(r, \theta)$ Zernike polinomunun ($n = 2, m = 2$ için) kutupsal olarak renklendirilmiş formu (b) Aynı noktaların ayırık bir görüntüde grid yapısında nasıl konumlandığının gösterilmesi

Şekil 26, Zernike polinomlarının farklı derecelerdeki (farklı n ve m değerleri için) yapılarını kutupsal formda renklendirilmiş şekillerle gösterir.



Şekil 26. Farklı derecelerdeki Zernike polinomları

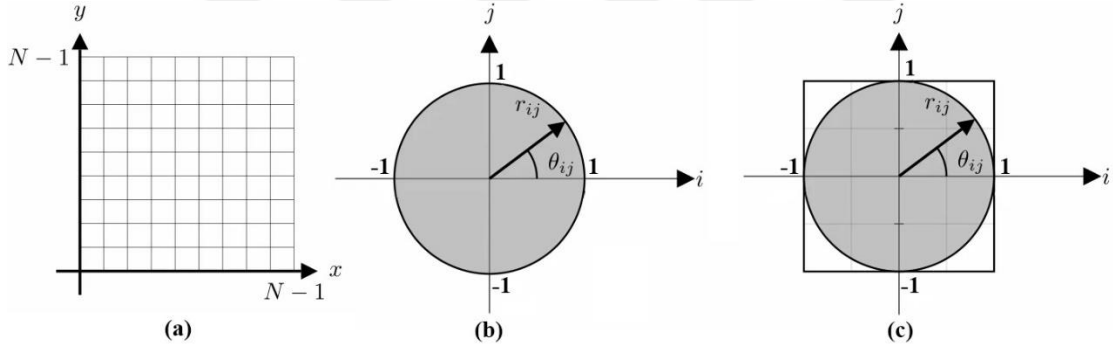
CZM'yi hesaplamak için, görüntü (veya ilgi bölgesi) öncelikle polar koordinatlar kullanılarak birim diske haritalanır ve görüntünün merkezi birim diskin orijini olarak kabul edilir. Bağntı (35), normalize edilmiş görüntü koordinatlarını (x_i, y_j) birim daire alanında polar koordinatlara (r, θ) dönüştürür.

$$x_i = r \cos \theta, \quad y_j = r \sin \theta$$

$$r = \sqrt{x_i^2 + y_j^2} = \sqrt{\left(\frac{2i - (N - 1)}{N - 1}\right)^2 + \left(\frac{2j - (N - 1)}{N - 1}\right)^2} \quad (35)$$

$$\theta = \tan^{-1}\left(\frac{y_j}{x_i}\right) = \tan^{-1}\left(\frac{(2j - (N - 1))/(N - 1)}{(2i - (N - 1))/(N - 1)}\right)$$

$N \times N$ boyutlu ayırık bir görüntünün orijinali ve birim diskin görüntü içine iz düşürülmüş hali Şekil 27'de gösterilmiştir. Örnek bir fotoğraf izdüşümü Şekil 28'de verilmektedir.

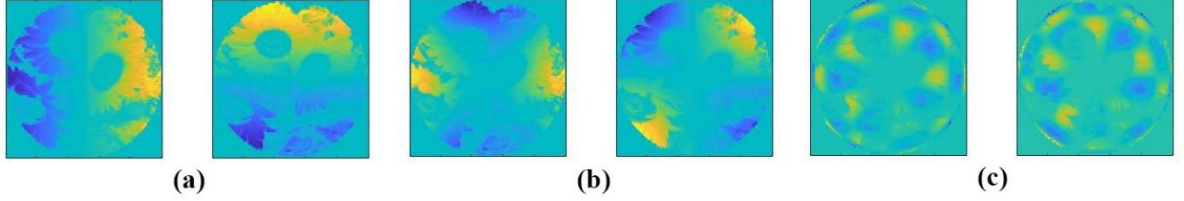


Şekil 27. (a) $N \times N$ boyutlu ayırık bir görüntü (b) Birim disk (c) Birim diskin normalize edilmiş görüntü içine iz düşürülmüş hali



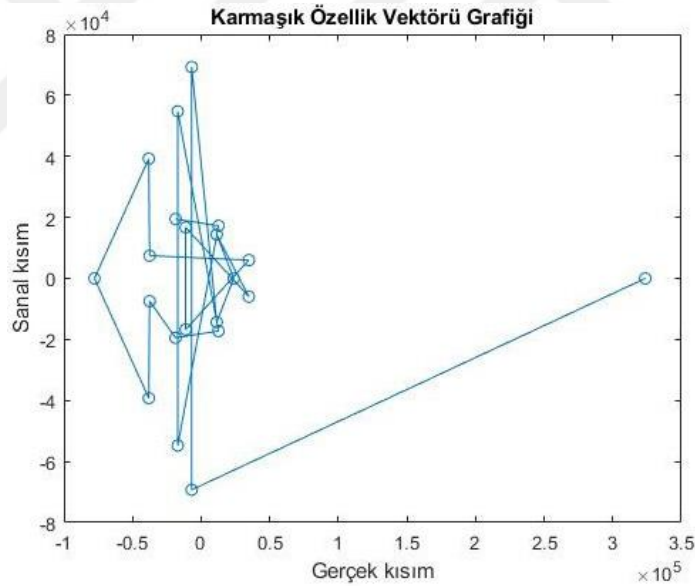
Şekil 28. Örnek bir fotoğraf izdüşümü

Zernike polinomlarının eşleniğinin $(V_{nm}^*(x_i, y_j))$ iz düşürülmüş görüntüye uygulama sonuçları gerçek ve sanal bileşenleri sırasıyla, $n = 1$ ve $m = 1$ için Şekil 29.(a)'da, $n = 2$ ve $m = 2$ için Şekil 29.(b)'de, $n = 9$ ve $m = 5$ için Şekil 29.(c)'de gösterilmiştir.



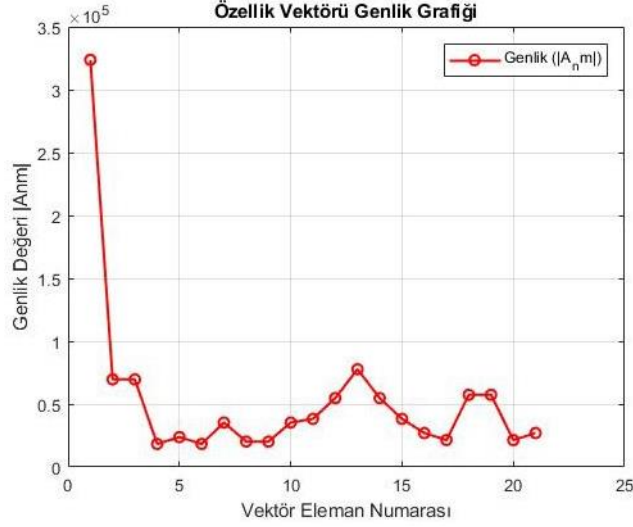
Şekil 29. Zernike polinomlarının eşleniğinin iz düşürülmüş görüntüye uygulanması

Bağıntı (32)'ye göre elde edilmiş öznitelik vektörünün karmaşık öznitelik vektörü grafiği Şekil 30'da verilmiştir.



Şekil 30. Karmaşık öznitelik vektörü grafiği

Bağıntı (32)'ye göre $A_{n,-m} = A_{n,m}^*$ olduğundan, $A_{n,m}$ 'nin karmaşık eşleniği $A_{n,-m}$ 'ye eşittir. Bu durum da genliklerin birbirine eşit olduğu $(|A_{n,m}| = |A_{n,-m}|)$ anlamına gelmektedir. Bir başka deyişle, karmaşık sayıların eşleniği alındığında genlik değeri değişmez. Şekil 31, öznitelik vektörünün genlik grafiğini $(|A_{n,m}|)$ belirtmektedir.



Şekil 31. Öznitelik vektörünün genlik grafiği

Orijinal görüntü, ters dönüşüm uygulanarak (öznitelik vektörü üzerinden) yeniden oluşturulabilir. Görüntünün öznitelik vektörü üzerinden yeniden oluşturulmasını ifade eden bağıntı (36)'da gösterilmiştir.

$$\hat{f}(x_i, y_j) = \sum_{n=0}^{n_{max}} \sum_{m=-n}^n A_{nm} V_{nm}(x_i, y_i) \quad (36)$$

İlgili bağıntıda n_{max} en yüksek moment derecesini temsil etmektedir. İkili (binary) görüntüler için yeniden yapılandırma nispeten az sayıda terimle gerçekleştirilebilir. Ancak, gri tonlamalı (gray level) görüntüler için yeniden yapılandırma genellikle daha yüksek bir derecede yapılır. Ayrıca, gri tonlamalı görüntüler için Zernike momentlerinin değeri, ikili görüntüler için olanlardan daha yüksektir. Bu durum, gri tonlama görüntülerdeki daha fazla bilgi ve daha fazla renk tonu nedeniyle, görüntüdeki karmaşıklığın artmasından kaynaklanır.

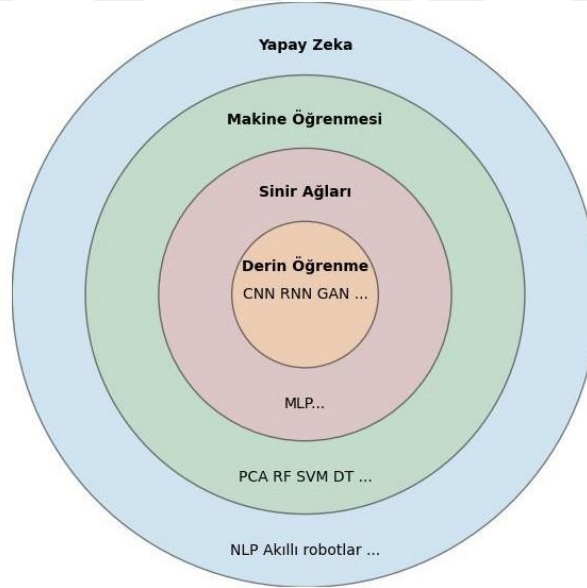
1.10. Sınıflandırma

Yapay zeka, bilgisayarların insan benzeri düşünme, karar verme ve problem çözme yeteneklerini kazanmasını hedefleyen bir bilim dalıdır. Bilgisayarların yalnızca veri işlemekle sınırlı kalmayıp öğrenme ve akıl yürütme gibi karmaşık süreçleri de gerçekleştirebileceği fikrine dayanır. Yapay zekanın temel amacı, insanların üstlendiği zorlu

bilişsel görevleri otomatikleştirerek günlük yaşamdan endüstriye kadar geniş bir alanda kullanılabilir çözümler sunmaktır.

Makine öğrenimi, yapay zekanın bir alt dalı olup, bilgisayar sistemlerinin deneyim ve verilerden öğrenerek kendi performanslarını geliştirmelerini sağlar. Geleneksel programlamadan farklı olarak, makine öğrenimi modelleri sabit kurallar ve algoritmalar yerine, verilerden çıkarım yaparak çalışır. Özellikle büyük veri miktarlarının artışı ve güçlü hesaplama kaynaklarının gelişimi, makine öğreniminin hızlı bir şekilde ilerlemesine ve yaygınlaşmasına katkı sağlamıştır. DT, SVM ve Bayes ağları (BN), makine öğreniminin önemli yöntemlerine örnek olarak gösterilebilir.

Derin öğrenme, makine öğrenmesinin bir alt kümesi olarak, çok katmanlı makine öğrenimi ile aynı anda özellik seçimi ve model uyumunu gerçekleştiren bir özelliğe sahiptir. Örneğin, CNN, görüntü tanıma ve analizinde yaygın olarak kullanılırken; RNN, dil işleme ve zaman serisi analizlerinde kullanılır. Derin öğrenmenin, yapay zekâ ve makine öğrenmesiyle olan ilişkisi Şekil 32’de gösterilmektedir.



Şekil 32. Derin öğrenmenin, yapay zekâ ve makine öğrenmesiyle olan ilişkisi

1.10.1. Makine Öğrenmesi

Makine öğrenmesi, bilgisayarların örnek verilerden öğrenerek tahminlerde bulunmasını veya kararlar almasını sağlayan bir yapay zeka dalıdır. Geleneksel

programlamadan farklı olarak, makine öğrenmesi sistemlerinde bilgisayara açık kurallar yazmak yerine veriyle eğitilen bir model oluşturulur. Bu model, geçmiş verilere dayanarak gelecekteki durumları tahmin edebilir veya belirli örüntüleri tanıyabilir. Makine öğrenmesi; matematik, istatistik, bilgisayar bilimi ve optimizasyon yöntemlerinin birleşimi olarak çalışır ve sağlık, finans, tarım, üretim gibi pek çok alanda önemli uygulamalara sahiptir.

Makine öğrenmesi algoritmalarının temel amacı, bir hedef fonksiyonu optimize ederek en iyi sonuçları elde etmektir. Bu süreç, önce bir model oluşturmayı, ardından bu modelin performansını artırmak için uygun bir hedef fonksiyonu tanımlamayı içerir. Hedef fonksiyon belirlendikten sonra, sayısal veya analitik optimizasyon yöntemleri kullanılarak çözüm bulunur. Örneğin bir e-posta hizmeti, spam mesajları ayırt etmek için makine öğrenmesi algoritmalarını kullanırken, bir otonom araç ise çevresinden gelen verilerle hareket stratejileri oluşturabilir.

Makine öğrenmesi algoritmaları, çözülmesi gereken problem türüne ve modelleme amacına göre dört ana kategoriye ayrılır (Shiliang vd., 2019).

Denetimli öğrenme, modelin etiketlenmiş veriler kullanılarak eğitildiği bir makine öğrenmesi yöntemidir. Bu yaklaşım, sınıflandırma ve regresyon olarak iki ana alt gruba ayrılır. Sınıflandırma, verileri belirli kategorilere ayırmayı amaçlar. Örneğin, bir cümleyi uygun bir kategoriye yerleştirmek veya bir görüntünün içeriğini sınıflandırmak gibi problemler sınıflandırma kapsamında ele alınır. Regresyon ise sürekli bir değeri tahmin etmeye yönelik bir yaklaşımdır. Örneğin, bir evin fiyatını tahmin etmek ya da bir ürünün gelecekteki satış rakamlarını öngörmek gibi durumlarda regresyon yöntemleri kullanılır.

Yarı denetimli öğrenme yönteminde, hem etiketli hem de etiketlenmemiş veriler bir arada kullanılır. Etiketlenmemiş verilerin fazla olduğu durumlarda, model performansını artırmak için etkili bir yaklaşım sağlar.

Denetimsiz öğrenme, etiketli veri olmaksızın verilerdeki gizli örüntüleri ve ilişkileri keşfetmeye odaklanır. Bu yaklaşım, verilerdeki yapıların anlaşılmasına ve gruplandırılmasına olanak tanır. Denetimsiz öğrenmenin temel kullanım alanlarından biri kümeleme olup, bu süreçte veriler benzer özelliklere göre gruplandırılır. Diğer bir önemli kullanım alanı ise boyut indirgeme olup, verilerin daha az boyutta temsil edilmesini sağlar. Bu yöntem, özellikle büyük veri setlerinin görselleştirilmesi ve işlenmesinde önemli bir rol oynar.

Pekiştirmeli öğrenme yönteminde, bir ajan çevresinden aldığı geri bildirimlere (ödülleri ya da cezalar) göre hareket etmeyi öğrenir. Bu bağlamda asıl amaç, uzun vadede en

yüksek ödülü sağlayacak stratejiyi geliştirmektir. Bu yöntem, robotik sistemlerden oyun stratejilerine kadar geniş bir alanda uygulanır.

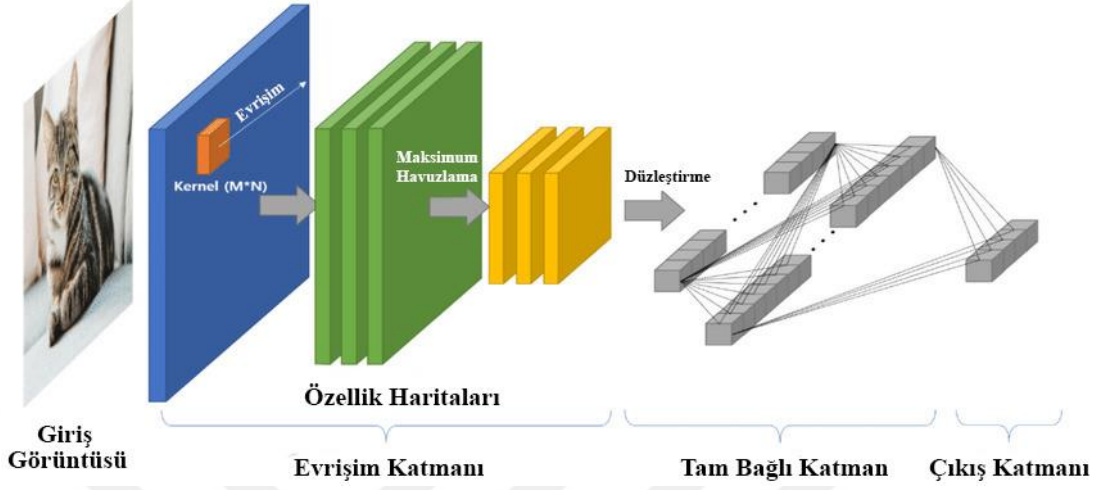
Her bir kategori, belirlenen hedef fonksiyonunu optimize etmek için farklı algoritmalar ve teknikler kullanır. Sonuç olarak makine öğrenmesi algoritmaları, çözmek istedikleri problem türüne uygun en iyi çözümü üretmeyi hedefler. Geniş bir kullanım alanına sahip olan makine öğrenmesi günümüz teknolojisinin temel yapı taşlarından biri olarak hızla gelişmeye devam etmektedir.

1.10.2. Derin Öğrenme ve Mimarileri

Derin öğrenme, verilerdeki karmaşık desenleri ve ilişkileri anlamak için çok katmanlı yapay sinir ağlarını kullanan bir makine öğrenmesi dalıdır. Bu katmanlar, verileri hiyerarşik olarak işler ve basit özelliklerden başlayarak daha karmaşık yapıları ortaya çıkarır. Böylece, büyük ve karmaşık veri kümelerinde bile etkili bir şekilde öğrenme ve tahmin yapabilme yeteneği sunar. CNN, LSTM, RNN, RBM, üretken karşıt ağlar (GAN) ve derin oto-kodlayıcılar, bu mimarilere örnek olarak verilebilir.

CNN ilk olarak 1980'lerde önerildi (Jiuxiang vd., 2018). CNN'lerde, standart yapay sinir ağlarında kullanılan matris çarpımı yerine evrişim işlemi kullanılır. Bu değişiklik, ağdaki ağırlık sayısını azaltarak ağırlık karmaşıklığını düşürür. Ayrıca, görüntüler doğrudan ham giriş olarak ağa verilebilir, böylece standart algoritmalarındaki özellik çıkarma sürecine gerek kalmaz. CNN'ler, hiyerarşik katmanların etkin eğitimi sayesinde derin öğrenmede gerçek anlamda ilk başarılı mimari olarak kabul edilir. CNN topolojisi, mekansal ilişkileri kullanarak ağdaki parametre sayısını azaltır ve bu da geri yayılım algoritmalarıyla performansın artmasını sağlar. Grafik işlemci birimi teknolojisinin gelişmesiyle birlikte Krizhevsky ve arkadaşları 2012 yılında ImageNet problemini çözmek için ekran kartı destekli bir program kullanmış ve bu durum CNN'leri tekrar popüler hale getirmiştir (Krizhevsky vd., 2012). Derin ağların en büyük sorunlarından biri, ağdaki birçok gizli düğüm nedeniyle eğitim süresinin uzun olmasıydı. Ancak ekran kartı işlemcilerinin paralel işlem kapasitesinin artmasıyla bu sorun aşılmıştır. CNN'lerin başarısındaki bir diğer önemli faktör, veri setlerindeki artış ve büyük miktarda veriyi etkili bir şekilde işleyebilme kapasitesidir. Ayrıca, transfer öğrenme yöntemleri sayesinde CNN'ler, önceden eğitilmiş modeller kullanılarak farklı problemler için yeniden uyarlanabilir ve bu da daha az veriyle yüksek doğruluk oranları elde edilmesini sağlar. Günümüzde CNN'ler, ses verisi analizi ve

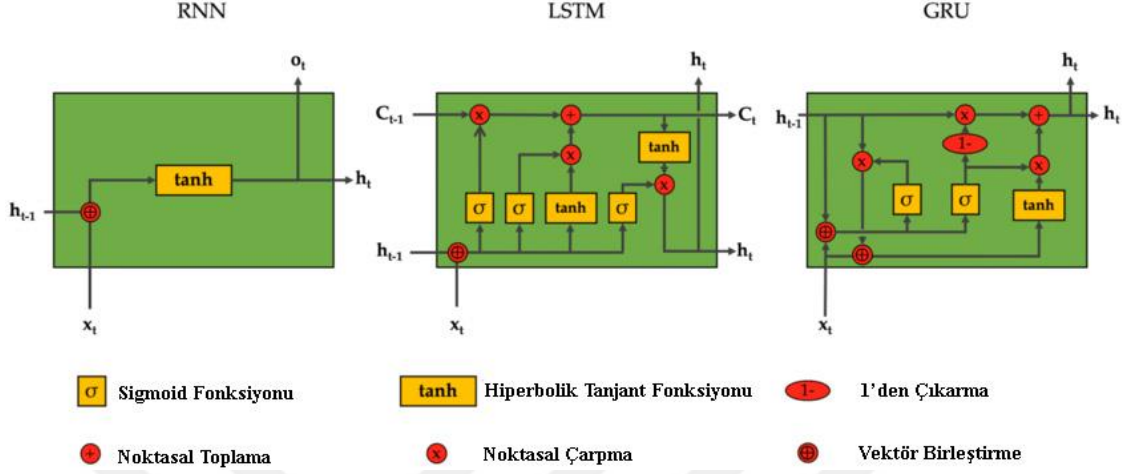
görüntü tanıma alanlarında oldukça popülerdir. Şekil 33, CNN'lerin genel yapısını göstermektedir.



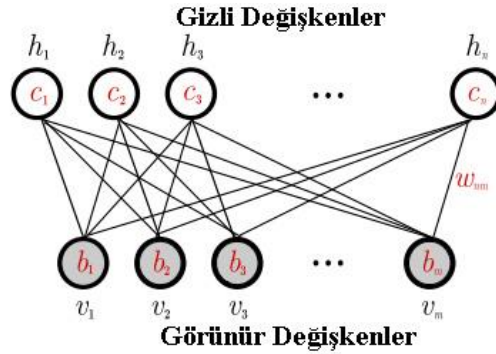
Şekil 33. CNN'lerin genel yapısı (Kim vd., 2020)

RNN ve LSTM, zaman serisi verilerini ve sıralı verileri analiz etmek için kullanılan güçlü derin öğrenme modelleridir. Bu modeller, özellikle geçmiş verilerdeki desenleri anlamada ve gelecekteki olayları tahmin etmede büyük başarı göstermektedir. LSTM, RNN'nin bir uygulamasıdır ve ilk olarak 1997'de Hochreiter ve arkadaşları tarafından önerilmiştir. LSTM, daha önce tanımlanan ileri besleme ağ yapılarından farklı olarak, önceki durumların bilgisini saklayabilir ve hafıza veya durum farkındalığı gerektiren işler için eğitilebilir. LSTM, RNN'nin önemli bir sınırlamasını yani kaybolan gradyanlar problemini, gradyanların değiştirilmeden geçmesine izin vererek kısmen çözer. LSTM'ler, kapı mekanizmaları sayesinde bilgiyi seçici bir şekilde güncelleyebilir ve uzun süreli bağımlılıkları daha iyi öğrenebilir. Bu özellik, özellikle uzun dizilerdeki ilişkileri kavramak için büyük bir avantaj sağlar. Bununla birlikte, hem LSTM hem de RNN modelleri, büyük veri kümeleri ve yüksek işlem gücü gereksinimleri nedeniyle eğitim aşamasında zaman ve kaynak açısından maliyetli olabilir. Son yıllarda, çift yönlü RNN ve kapı özyinelemeli geçitler (GRU) gibi türev modeller de geliştirilmiş ve farklı uygulama senaryolarında daha verimli sonuçlar elde edilmiştir. Bu sayede oluşabilecek problemlerin nerede ve ne zaman olabileceğini öngörerek proaktif önlemler alınmasını sağlar. RNN, genellikle dil çevrimlerinde kullanılmakla birlikte, zaman serileri gibi sıralı verilerle bir sonraki noktayı tahmin etme işlemi için idealdir. Örneğin, finansal hareketlerin verilerini kullanarak gelecekteki durumları tahmin edebiliriz. Ayrıca, cümlelerde art arda gelen kelimelere göre

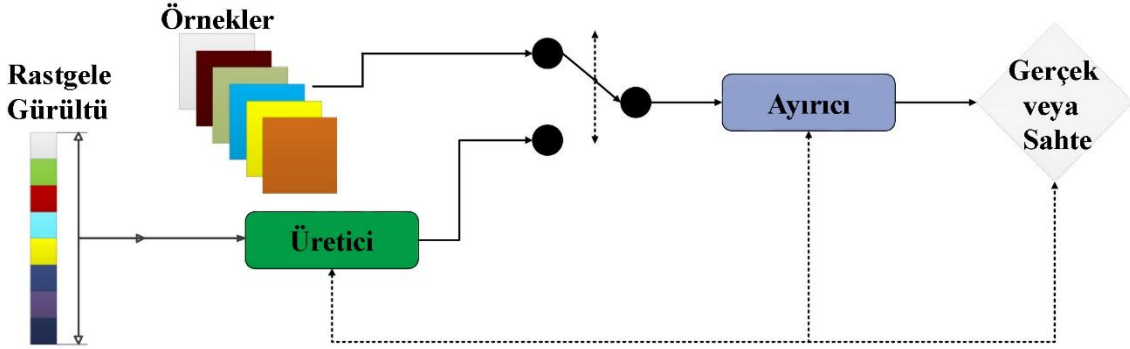
bir sonraki kelimeyi tahmin etmek de RNN'lerin yaygın kullanım alanlarındandır (Sherstinsky, 2020). RNN, LSTM ve GRU'nun genel yapısı Şeki 34'de gösterilmektedir.



RBM, etiketlenmemiş verilerden doğrusal olmayan üretici modeller oluşturmak için kullanılan bir yapay sinir ağıdır. Bu ağ, denetimsiz öğrenme algoritmalarını kullanarak giriş verisi üzerinde özellikle Boltzmann dağılımına tabi olan bir olasılık dağılımını öğrenir ve girişin olasılıksal olarak yeniden inşa edilmesini sağlar. RBM, görünür ve gizli olarak iki katmadan oluşan bir yapıya sahiptir. Görünür katmanındaki her bir birim, gizli katmandaki tüm birimlere bağlanırken, aynı katmandaki birimler arasında bağlantılar yoktur. Bu özellik, RBM'yi kısıtlı bir yapı olarak tanımlar ve bu nedenle adını "kısıtlı" Boltzmann makinesi alır. Regresyon, sınıflandırma, boyut indirgeme, zaman serisi modelleme ve özellik çıkarımı gibi bir dizi uygulama alanında etkilidir (Teh, 2000). n gizli, m görünür değişkenli bir RBM'nin grafi Şekil 35'de gösterilmektedir.



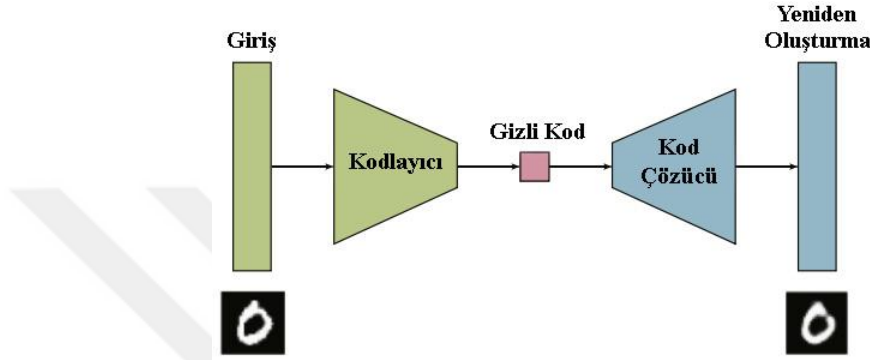
GAN'lar, oyun teorisine dayalı üretici modelleri öğrenmek için kullanılan güçlü bir yapay zeka aracıdır. GAN'lar veri dağılımını doğrudan modellemek yerine, bu verilerden örnekler üretmeyi hedefler ve bunu gerçekleştirmek için derin sinir ağlarını kullanır. Rastgele gürültü verisi üretici ağ tarafından işlenerek, gerçek verilere benzer sahte örnekler üretir. Bu süreçte ayırıcı ağ, üretici ağından gelen sahte verilerle gerçek verileri ayırt etmeye çalışır. Üretici ve ayırıcı arasındaki karşılıklı etkileşim, her iki ağın ağırlıklarını ve sapmalarını güncelleyerek kayıp fonksiyonlarını minimize etmelerini sağlar. Böylece, üretici daha gerçekçi veriler üretmeye ayırıcı ise bu verileri daha doğru ayırt etmeye başlar. GAN'ların eğitimi, yüksek boyutlu parametrelerle yapılan bir Nash dengeleme süreci gerektirir (Nash, 1950). Son dönemde, GAN'ların yakınsama sürecini iyileştirmeye yönelik tekniklerin kullanımı artmıştır. Bu teknikler, daha iyi yarı denetimli öğrenme performansı ve daha gerçekçi örnekler üretmeye imkan tanımaktadır. GAN'lar, özellikle görüntü üretimi, video oluşturma ve ses sentezi gibi yaratıcı alanlarda devrim yaratmış ve fotoğraf düzenleme, deepfake üretme ve sanat üretme gibi pek çok alanda başarılı bir şekilde uygulanmıştır (Goodfellow vd., 2024). Şekil 36, GAN'ın genel yapısını ifade etmektedir (Dong vd., 2021).



Şekil 36. GAN'ın genel yapısı (Dong vd., 2021)

Derin oto-kodlayıcılar, etiketlenmemiş verilerle çalışan ve veriyi sıkıştırarak boyut indirgeme işlemi gerçekleştiren denetimsiz öğrenme tabanlı yapay sinir ağlarıdır. Bu ağlar, gizli katmandaki temsiller aracılığıyla veriyi daha verimli işler ve özellik çıkarımı yapar. Oto-kodlayıcılar, giriş verisini yeniden oluşturmak için öğrenilen bir temsili kullanır ve bu süreç geri yayılım algoritmasıyla gerçekleştirilir. Oto-kodlayıcılar, temel bileşen analizi gibi lineer boyut indirgeme yöntemlerini genişleterek, doğrusal olmayan temsiller elde eder. Giriş katmanı, gizli katman ve çıkış katmanı olmak üzere genellikle üç katmanda oluşur.

Gizli katmanda yer alan nöron sayısı, giriş ve çıkış katmanlarındaki nöronlardan daha azdır, bu da verinin sıkıştırılmasını ve ağda daha az veriyle işlem yapılmasını sağlar. Bu yapı, ağın daha verimli çalışmasını ve verilerin daha etkili bir şekilde temsil edilmesini sağlar. Derin öğrenme ve transfer öğrenme gibi alanlarda önemli bir rol oynar ve etiketlenmemiş verilerle çalışarak güçlü ve etkili çözümler sunar (Baldi, 2011). Şekil 37, oto-kodlayıcıların genel yapısını gösterir (Pinaya vd., 2020).



Şekil 37. Oto-kodlayıcıların genel yapısı (Pinaya vd., 2020)

1.10.3. Derin Sinir Ağları

ANN, insan beyninin çalışma prensiplerinden esinlenerek geliştirilmiş bir hesaplama modelidir. Bu ağlar, verilerin işlenmesini ve öğrenilmesini sağlamak için çeşitli katmanlardan oluşur ve her bir katman, veriler üzerinde belirli bir işlevi yerine getirir. ANN, derin öğrenmenin temel yapı taşlarını oluşturur. ANN'lerin kökenleri, 1940'lara dayanır. İlk olarak, bilim insanları, beynin nöronlar arasındaki bağlantıların öğrenme ve bilgi işleme üzerindeki etkilerini modellemeye çalıştılar. Bu çalışmaların sonucu olarak, ilk ANN'ler ortaya çıktı.

Sinir ağlarının çalışma prensibi, biyolojik nöronların işleyişine benzer şekilde tasarlanmıştır. Bir nöron, aldığı girişleri ağırlıklı bir şekilde toplar. Bu ağırlıklı toplam, nöronun girişlerine bağlanan sinapslar aracılığıyla yapılır ve bu süreç, sinapsların her birine bir ağırlık değeri atanarak gerçekleştirilir. Bu ağırlıklı toplam, aslında nöronun aldığı verilerin bir tür ölçeklendirilmesidir. Ancak nöronlar sadece bu toplamı kullanarak bir çıktı üretmezler. Eğer sadece ağırlıklı toplam alınsaydı, nöronun işlemi basit bir doğrusal cebirsel işlemle farksız olurdu. Gerçek nöronlar, aldıkları girişlerin ağırlıklı toplamını doğrudan

kullanmak yerine, bu toplam üzerinde genellikle doğrusal olmayan bir işlem uygular. Bu doğrusal olmayan işlem, nöronun çıktısını belirler. Bir nöron belirli bir eşik değerini aşan girişlere tepki verir ve çıktıyı üretir. Yani ağırlıklı toplamın kendisi sadece ham veriyi temsil eder, ancak nöronun aktif olup olmayacağına karar veren asıl işlem doğrusal olmayan bir fonksiyondur. Doğrusal olmayan fonksiyonlar, sinir ağlarının güçlü bir özellik kazanmasını sağlar. Bu bağlamda ilgili fonksiyonlar, karmaşık ve doğrusal olmayan ilişkileri öğrenme ve modelleme yeteneği sunar. Eğer sinir ağları sadece doğrusal bir fonksiyon kullanarak çalışsaydı, ağ yalnızca doğrusal ilişkileri öğrenebilir ve daha karmaşık problemlerin üstesinden gelemezdi.

Öğrenme ağıdaki ağırlıkların değerlerini belirlemeyi içerir ve buna ağı eğitilmesi denir. Eğitim tamamlandıktan sonra, ağın çıktısını hesaplamak için eğitim sürecinde belirlenen ağırlıklar kullanılır ve bu süreç test işlemi olarak adlandırılır.

Sinir ağları alanında, derin öğrenme adı verilen bir alan bulunmaktadır. Bu alanda, sinir ağları üç katmandan daha fazla katmana sahiptir, yani birden fazla gizli katman bulunur. Böylece DNN'ler, daha karmaşık ve soyut yüksek seviyeli özellikleri öğrenme kapasitesine sahiptir. Doktora çalışmaları kapsamında, derin öğrenme alanında kullanılan sinir ağlarına genel olarak DNN'ler denilecektir (Sze vd., 2017).

1.11. Karmaşık Değerli Derin Sinir Ağları

Makine öğrenmesi ve derin öğrenme modelleri (MLP, DNN, CNN, RNN vb.), işlemlerinde karmaşık sayılar yerine gerçek sayılar kullanmaktadır. Bu modeller, gerçek değerli ağırlıklar ve gerçek değerli aktivasyon fonksiyonları kullanılarak standart geri yayılım algoritması ile eğitilmektedir. Öte yandan, karmaşık değerli sinir ağları (CVNN), karmaşık değerli ağırlıklar, karmaşık değerli aktivasyon fonksiyonları ve karmaşık değerli geri yayılım algoritması kullanmaktadır. CVNN, özellikle karmaşık veri analizi ve ayrıntılı bilgi gerektiren özel uygulamalarda tercih edilirken, gerçek değerli sinir ağları daha genel uygulamalarda yaygın olarak kullanılmaktadır.

Son yıllarda, karmaşık sayılar kullanarak yapay sinir ağları oluşturma ve karmaşık değerli sinir ağlarının gerçek değerli muadillerine kıyasla olası avantajlarını keşfetme konusunda artan bir ilgi vardır. Ayrıca, CVNN'ler birçok pratik sistemi verimli bir şekilde modellemek için kullanılmaktadır. Bu nedenlerle, CVNN'lere yönelik araştırmalar artmıştır (Chanthorn vd., 2020).

Karmaşık değerli sinir ağları, bir sinir ağı için ağırlıklar, eşikler, giriş ve çıkış verilerinin karmaşık sayılarla temsil edilebildiği yapılardır. Böylece, karmaşık değerli veriler, ağına girişine herhangi bir ön işleme tabi tutulmadan doğrudan verilebilir (Lee vd., 2022). CVNN'ler özellikle telekomünikasyon, askeri sistemler, görüntü işleme, robotik sistemler, otonom sistemler, sinyal işleme ve radar teknolojilerinde kullanılır. Bu alanlarda sinyaller sıklıkla karmaşık sayılarla temsil edilir ve bu sinyallerin kompleks değerli sinir ağları ile doğrudan işlenmesi daha verimli sonuçlar sağlar (Virtue vd., 2017).

1.11.1. Optimizasyon Algoritmaları

Derin öğrenme modellerinin temel amacı, ağına ağırlıklarını optimize ederek kayıp fonksiyonunu minimize etmektir. Bu süreç, genellikle gradyanlara dayalı bir yaklaşımı kullanır ve gradyan inişi gibi optimizasyon algoritmalarını içerir. Gradyan inişi algoritması, ağırlık vektörlerinin en uygun değerlerini belirlemek için önemli bir yöntem sunar. Ancak bu süreçte, özellikle derin ağlarda karşılaşılan “gradyan kaybolması” ve “gradyan patlaması” gibi problemler optimizasyonun etkinliğini olumsuz etkileyebilir. Gradyanların çok küçük değerlere düşmesi öğrenme sürecinin yavaşlamasına, çok büyük değerlere ulaşması ise ağırlıkların kararsız hale gelmesine yol açabilir. Bu problemlerin üstesinden gelmek için hem ileri seviye optimizasyon algoritmaları geliştirilmiş hem de bu algoritmalara gradyan problemlerini hafifletecek çözümler entegre edilmiştir.

Optimizasyon algoritmalarının başarısı, bir probleme ait sınıfları ayırıştırabilmek için en uygun ağırlık vektörlerini bulma kabiliyetiyle doğrudan ilişkilidir. Bu bağlamda optimizasyon algoritmaları, derin öğrenme modellerinin etkinliği ve başarısı için vazgeçilmezdir. Bazı optimizasyon algoritmaları aşağıdaki gibi sıralanabilir (Sun vd., 2020) (Shrestha, 2019).

- Gradyan inişi algoritması, makine öğrenimi ve derin öğrenme algoritmalarının temelini oluşturan bir yöntemdir. Bu yöntem, bir fonksiyonun minimum değerine ulaşmak için türev veya eğim bilgisini kullanır. Bu süreçte, rastgele bir başlangıç noktası seçilir ve bu noktadaki türevin pozitif veya negatif olmasına göre fonksiyonun değerini azaltacak yönde ilerlenir. Burada amaç fonksiyonun minimum noktasına ulaşmak ve bu noktada işlemi durdurmaktır.

Bu yöntem bazı zorluklar içermektedir. Şöyle ki, fonksiyonun karmaşık bir yapıya sahip olduğu durumlarda, yöntem yerel minimumlarda takılabilir. DNN’lerde kullanılan

gradyan inişi yöntemi, çok boyutlu bir alanda uygulanır. Bu alan ağıdaki her ağırlık için bir boyut içerir ve karmaşıklık arttıkça doğru çözümü bulmak zorlaşabilir.

Geriye yayılım algoritması, gradyan inişini kullanarak ağırlıkların optimize edilmesini sağlar. Bu yöntemde, ağırlık çıktısı ile beklenen çıktı arasındaki hata hesaplanır ve bu hata ağırlıklar üzerinde geri yayılır. Her bir ağırlık, hatayı azaltacak şekilde düzenlenir ve bu işlem veri setindeki tüm örnekler işlenene kadar devam eder. Süreç, hata daha fazla azalamadığında tamamlanır.

Bu yöntem, her katmandaki hatanın bir sonraki katmana nasıl yansıtılacağını hesaplamak için kullanılır. Böylece her ağırlık ve bağlantı, sistemin genel performansını artıracak şekilde optimize edilir.

- SGD algoritması, gradyan inişinin en yaygın kullanılan varyasyonlarından biridir ve ağırlıkların güncellenmesini daha hızlı ve verimli hale getirir. Gradyan inişinde tüm eğitim veri kümesi işlendikten sonra ağırlıklar güncellenirken, SGD'de ise ağırlıklar, eğitim veri kümesindeki küçük bir örnek grubunun (mini-batch) işlenmesinden sonra revize edilir. Buna bağlı olarak daha sık yapılan güncellemeler, modelin global minimuma daha hızlı ulaşmasını sağlayabilir. Ancak özellikle derin sinir ağlarında, gradyan kaybolması problemi nedeniyle gradyanlar çok küçük değerlere düşerek öğrenme sürecini yavaşlatabilir ve modelin yakınsamasını zorlaştırabilir. Yanlış ayarlanmış bir öğrenme oranı da bu sorunu daha da kötüleştirebilir. Bu problemleri hafifletmek için başlangıç ağırlıklarının optimizasyonu ve aktivasyon fonksiyonu seçimi gibi stratejiler kullanılabilir. Örneğin, Xavier veya He başlatma yöntemleriyle ağırlıkların başlatılması, gradyanların daha dengeli bir şekilde yayılmasını sağlayarak gradyan kaybolma sorununu azaltabilir (Kumar, 2017). Ayrıca, ReLU gibi sabit türevlere sahip aktivasyon fonksiyonları, sigmoid veya tanh gibi gradyanları sıfıra yaklaştıran fonksiyonlara kıyasla daha etkili bir öğrenme süreci sunabilir.

- SGD yönteminde öğrenme oranı, ağırlık güncellemelerinin büyüklüğünü belirleyen sabit bir çarpan olarak kullanılır. Ancak bu yaklaşım, bazı durumlarda minimum noktayı aşan güncellemeler ve gradyanlardaki gürültü nedeniyle yavaş bir yakınsama süreci gibi iki temel soruna yol açabilir.

Momentum algoritması, bu sorunların üstesinden gelmek için fiziğin momentum kavramından esinlenir. Algoritma, *momentum* adı verilen bir değişken kullanır ve bu değişken gradyanların üstel olarak azalan bir ortalamasına dayanır. Böylece, sistemin yanlış yönlere doğru gereksiz iniş yapması engellenir. Öğrenme oranı ve momentum parametresi, ağırlık güncellemelerinin hem yönünü hem de değişimini kontrol etmek için birlikte çalışır.

Doğru şekilde ayarlanmış bir momentum faktörü, öğrenme oranının yüksek olduğu durumlarda yakınsama sürecinde oluşabilecek salınımları azaltmada önemli bir rol oynar. Ancak momentum faktörünün seçimi kritik bir konudur. Faktör çok küçük olduğunda, algoritmanın yakınsama hızını artırma etkisi yeterince hissedilemez. Öte yandan çok büyük bir faktör değeri, güncellemelerin en uygun noktayı aşmasına ve dengesiz bir öğrenme sürecine neden olabilir.

Yapılan birçok deney, momentum faktörü için en uygun değer 0.9 olduğunu göstermiştir. Bu değer, algoritmanın yakınsama hızını artırırken kararlılığı koruyarak öğrenme sürecini optimize eder.

- SGD yönteminde öğrenme oranının doğru bir şekilde ayarlanması, modelin etkinliğini büyük ölçüde etkiler. Ancak uygun bir öğrenme oranı belirlemek zorlu bir süreçtir ve bu problemi çözmek için öğrenme oranını otomatik olarak ayarlayan çeşitli uyarlanabilir yöntemler geliştirilmiştir. Bu yöntemler, parametre ayarına gerek duymadan hızlı bir şekilde yakınsama sağlayarak genellikle başarılı sonuçlar elde eder. SGD'nin en yaygın kullanılan iyileştirmelerinden biri olan AdaGrad algoritması, her ağırlık için farklı bir öğrenme oranı belirler ve öğrenme oranını geçmiş gradyanları dikkate alarak dinamik bir şekilde ayarlar. Bu özellik, öğrenme oranını manuel olarak belirleme ihtiyacını ortadan kaldırırken özellikle seyrek veri yapılarında etkili sonuçlar sağlar.

AdaGrad'ın bazı dezavantajları da vardır. Eğitim süreci ilerledikçe biriken gradyanlar, öğrenme oranını sıfıra yaklaştırarak güncellemelerin etkisiz hale gelmesine neden olabilir. Bu durum, öğrenme sürecini yavaşlatarak modelin yeterince hızlı öğrenmesini engeller. Ayrıca, AdagGrad'ın öğrenme oranını sürekli olarak küçültmesi gradyan kaybolması problemini daha da artırabilir. Bu sorunların üstesinden gelmek için bazı stratejiler geliştirilmiştir. Örneğin gradyan kırpma yöntemi, büyük gradyanlar için bir eşik değeri belirleyerek gradyan patlaması problemini kontrol altına alırken, ReLU ve türevleri gibi aktivasyon fonksiyonları gradyanların daha verimli bir şekilde yayılmasına yardımcı olabilir.

AdaGrad'ın sınırlamalarına çözüm olarak geliştirilen RMSProp algoritması, geçmişteki tüm gradyanları biriktirmek yerine belirli bir zaman dilimindeki gradyanlara odaklanır. RMSProp, üstel azalan hareketli ortalamalar kullanarak öğrenme oranını daha stabil hale getirir ve sıfıra yaklaşmasını engeller. Bu sayede, optimizasyon süreci daha verimli bir şekilde devam eder ve DNN'lerde karşılaşılan temel problemlerden biri olan gradyan yönetimi daha etkin bir şekilde gerçekleştirilir.

- Adam algoritması, her parametre için uyarlanabilir öğrenme oranları sunan ve momentum tabanlı bir yaklaşımla gradyanların geçmiş değerlerini dikkate alan gelişmiş bir SGD yöntemidir. Bu yöntem, AdaDelta ve RMSProp gibi algoritmalarından ilham alarak geçmiş gradyanların ve karelerinin üstel olarak azalan ortalamalarını hesaplar. Ayrıca, momentum yöntemine benzer şekilde, geçmiş gradyanların üstel ortalamasını da tutar. Bu süreçte kullanılan $\beta_1=0.9$, $\beta_2 = 0.999$ ve $\epsilon = 10^{-8}$ gibi standart parametreler, gradyanların dengeli bir şekilde güncellenmesini sağlar ve hızlı yakınsama ile etkili bir öğrenme süreci sunar.

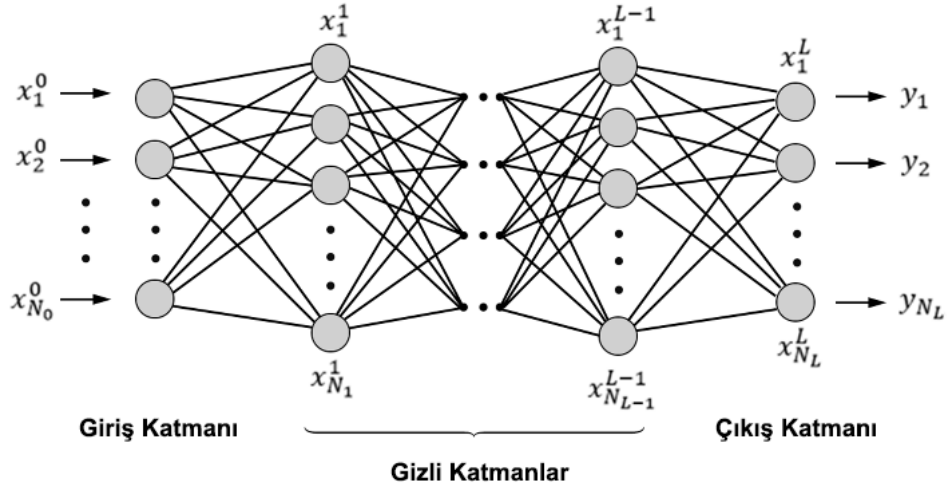
Adam algoritması, özellikle ReLU gibi aktivasyon fonksiyonlarıyla birlikte kullanıldığında iyi performans gösterir ve gradyan patlaması gibi problemlerin yönetiminde başarılıdır. Ancak bu gibi durumlarda gradyanların büyüklüğünün kontrol altına alınması gerektiğinden, Adam ile birlikte gradyan kırılması gibi tekniklerin kullanılması önerilir. Ayrıca başlangıç ağırlıklarının Xavier veya He başlatma gibi yöntemlerle optimize edilmesi, gradyanların düzgün yayılmasını sağlayarak gradyan kaybolması problemini engelleyebilir.

DNN'lerde, Adam algoritmasının etkinliği artık bağlantılar ile artırılabilir. Artık yapılar, bir katmanın çıktısına önceki katmanların çıktısını ekleyerek gradyanların bir katmandan diğerine daha etkili bir şekilde geçmesini sağlar. Bu mekanizma, derin ağlarda öğrenme sürecini kolaylaştırır ve Adam'ın sunduğu avantajları tamamlayarak optimizasyonu daha da verimli hale getirir. Adam, bu güçlü özellikleri sayesinde diğer uyarlanabilir algoritmalara kıyasla pratikte oldukça başarılı sonuçlar verir.

1.11.2. CVDNN Eğitim Algoritması

CVDNN mimarisinin öğrenme sürecinde gradyan tabanlı öğrenme kullanır; ileriye doğru yayılım sırasında hatayı hesaplar ve ardından bu hatayı her bir nörona geri yayarak geri yayılım sırasında ağırlıkları günceller.

Geri yayılım algoritması, yapay sinir ağlarındaki ağırlık ve yanlılığı optimize etmek için kullanılan temel bir algoritmadır. Bu algoritma, hatayı en aza indirmede ve sinir ağlarının eğitimi sırasında öğrenme sürecini etkili bir şekilde yürütmede kritik bir rol oynar. Şekil 38, bir CVDNN'yi temsil etmektedir.



Şekil 38. Karmaşık değerli derin sinir ağı (CVDNN)

Derin sinir ağı, $0 \leq l \leq L$ ile indekslenen L adet katmana sahiptir. İlgili eşitlikte $l = 0$ olduğunda giriş katmanını, $l = L$ olduğunda ise çıkış katmanını temsil eder. Her l katmanı, $1 \leq n \leq N_l$ ile indekslenen N_l nörona sahiptir. w_{nm}^l parametresi, l katmanındaki n . nörondan, $l - 1$ katmanındaki m . nörona olan bağlantının ağırlığını ifade eder. Bu nedenle, l katmanındaki n . nöronun çıktısı ve $(l + 1)$ katmanındaki m . nöronun girdisi, bağıntı (37)'de gösterildiği gibi ifade edilir ve ileri beslemeli hesaplama olarak bilinir. İlgili bağıntıda, b terimi n . nöronun eşik değerini ve f ise kompleks aktivasyon fonksiyonunu temsil etmektedir.

$$\begin{aligned} net_n^l &= \sum_{m=1}^{N_{l-1}} w_{nm}^l x_m^{l-1} + b_n^l \\ x_n^l &= f(net_n^l) \end{aligned} \quad (37)$$

$0 \leq l \leq L$ eşitliğinde $l = L$ olduğunda, ilgili katmanın çıktısı x_n^L aynı zamanda karmaşık değerli sinir ağının da çıktısı olmaktadır. Karmaşık değerli sinir ağının çıktısı Şekil 38'de gösterildiği gibi y_n olarak ifade edilir. Karmaşık değerli geri yayılımında yaygın olarak kullanılan hata fonksiyonu bağıntı (38)'de gösterilmiştir. Bu denklemde, $d_n(t)$, $y_n(t)$ ve e^* sırasıyla beklenen çıktıyı, gerçek çıktıyı ve t zamanındaki karmaşık eşlenik hatayı temsil eder.

$$E(t) = \frac{1}{2} \sum_{n=1}^{N_L} e_n(t) e_n^*(t) \quad (38)$$

$$e_n(t) = d_n(t) - y_n(t)$$

Herhangi bir ağırlığın (w_{nm}^l) güncellenmesi gerektiğinde, Wirtinger hesaplamasına göre ($\partial E / \partial w_{nm}^l$ ve $\partial E / \partial (w_{nm}^l)^*$), hata fonksiyonu $E(t)$ 'nin ilgili ağırlığa göre kısmi türevlerinin hesaplanması gerekir (Abdalla, 2024). Ayrıca, $E(t)$ analitik olmayan bir gerçek değerli fonksiyondur; bu nedenle, $E(t)$ 'nin w_{nm}^l 'nin gerçek ve sanal kısımlarına göre kısmi türevlerinin ayrı ayrı hesaplanması gerekir (Leung & Haykin, 1991). w_{nm}^l 'nin formülü ve çıkış katmanının ($l = L$) güncelleme kuralı bağıntı (39)'da verilmiştir. İlgili bağıntı, η öğrenme sabitini temsil eder.

$$\begin{aligned} w_{nm}^l(t) &= w r_{nm}^l(t) + i w i_{nm}^l(t) \\ w_{nm}^l(t+1) &= w_{nm}^l(t) - \frac{1}{2} \eta \frac{\partial E(t)}{\partial w_{nm}^l(t)} \\ \frac{\partial E(t)}{\partial w_{nm}^l(t)} &= \frac{\partial E(t)}{\partial w r_{nm}^l(t)} + i \frac{\partial E(t)}{\partial w i_{nm}^l(t)} \end{aligned} \quad (39)$$

Bağıntı (39)'u takiben, zincir kuralı hem gerçek hem de kompleks değişkenlerle kompleks fonksiyonlara uygulanabilir. Bağıntı (40)'da gösterilen zincir kuralında, karmaşık değişken $z = x + iy$ olarak verildiğinde, h ve g karmaşık değerli fonksiyonları için, h fonksiyonu g 'ye bağlıdır. Benzer şekilde g fonksiyonu da z^* 'ye bağlıdır. Bu nedenle tüm ilişkiler $h(g(z))$ şeklinde ifade edilebilir (Abdalla, 2024).

Karmaşık değerli geri yayılım ve CVDNN optimizasyonu, karmaşık değerli aktivasyon fonksiyonlarının gerçek ve sanal kısımlarına göre kısmi türevleri hesaplayarak uygulanabilir. Bu, dolayısıyla hem gerçek hem de karmaşık fonksiyonlar için karmaşık zincir kuralı uygulamasını kolaylaştırır.

$$\begin{aligned} \frac{\partial h(g)}{\partial z} &= \frac{\partial h}{\partial g} \frac{\partial g}{\partial z} + \frac{\partial h}{\partial g^*} \frac{\partial g^*}{\partial z} \\ \frac{\partial h(g)}{\partial z^*} &= \frac{\partial h}{\partial g} \frac{\partial g}{\partial z^*} + \frac{\partial h}{\partial g^*} \frac{\partial g^*}{\partial z^*} \end{aligned} \quad (40)$$

z' nin gerçek ve hayali kısımlarına göre, karmaşık zincir kuralı bağıntı (41)'de gösterildiği şekilde elde edilir.

$$\frac{\partial g}{\partial z} = \frac{\partial g}{\partial x} \frac{\partial x}{\partial z} + \frac{\partial g}{\partial y} \frac{\partial y}{\partial z} \quad (41)$$

Karmaşık zincir kuralı, bağıntı (39)'daki kısmi türevleri genişletmek için kullanılmıştır. İlgili formüller bağıntı (42)'de gösterilmiştir.

$$\begin{aligned} \frac{\partial E(t)}{\partial wr_{nm}^l} = & \frac{\partial E(t)}{\partial y_n} \frac{\partial y_n}{\partial net_n^l} \frac{\partial net_n^l}{\partial wr_{nm}^l} + \frac{\partial E(t)}{\partial y_n} \frac{\partial y_n}{\partial (net_n^l)^*} \frac{\partial (net_n^l)^*}{\partial wr_{nm}^l} + \\ & \frac{\partial E(t)}{\partial y_n^*} \frac{\partial y_n^*}{\partial net_n^l} \frac{\partial net_n^l}{\partial wr_{nm}^l} + \frac{\partial E(t)}{\partial y_n^*} \frac{\partial y_n^*}{\partial (net_n^l)^*} \frac{\partial (net_n^l)^*}{\partial wr_{nm}^l} \\ = & -\frac{1}{2} (d_n^* - y_n^*) \frac{\partial f(net_n^l)}{\partial net_n^l} x_m^{l-1} - \frac{1}{2} (d_n^* - y_n^*) \frac{\partial f(net_n^l)}{\partial (net_n^l)^*} (x_m^{l-1})^* - \\ & \frac{1}{2} (d_n - y_n) \frac{\partial f(net_n^l)^*}{\partial net_n^l} x_m^{l-1} - \frac{1}{2} (d_n - y_n) \frac{\partial f(net_n^l)^*}{\partial (net_n^l)^*} (x_m^{l-1})^* \end{aligned} \quad (42)$$

$y_n = f(net_n^l)$, olmak üzere net_n^l bağıntı (37)'de $E(t)$ ise bağıntı (38)'de verilmiştir.

Gerçek kısım için bağıntı (42)'de verilen formüle benzer olarak, sanal kısma karşılık gelen formül bağıntı (43)'de verilmiştir. Bağıntı (42) ve (43)'ün bağıntı (39)'a yerleştirilmesiyle, çıkış katmanı için hata fonksiyonunun türevinin son hali bağıntı (44)'de gösterildiği gibi elde edilir. Benzer şekilde, bağıntı (44), $l \leq L - 1$ olduğunda gizli katmanlar için güncelleme kuralında kullanılan formülü gösterir.

$$\begin{aligned}
i \frac{\partial E(t)}{\partial w_{nm}^l} = & \\
i \left[-\frac{1}{2} i (d_n^* - y_n^*) \frac{\partial f(net_n^l)}{\partial net_n^l} x_m^{l-1} + \right. & \\
\frac{1}{2} i (d_n^* - y_n^*) \frac{\partial f(net_n^l)}{\partial (net_n^l)^*} (x_m^{l-1})^* - & \\
\frac{1}{2} i (d_n - y_n) \frac{\partial f(net_n^l)^*}{\partial net_n^l} x_m^{l-1} + & \\
\left. \frac{1}{2} i (d_n - y_n) \frac{\partial f(net_n^l)^*}{\partial (net_n^l)^*} (x_m^{l-1})^* \right] &
\end{aligned} \tag{43}$$

$$w_{nm}^l(t+1) = w_{nm}^l(t) + \eta \delta_n^l (x_m^{l-1})^*$$

burada

$$\delta_n^l =>$$

l = L olduğunda

$$\frac{1}{2} (d_n^l - y_n^l) \frac{\partial f(net_n^l)^*}{\partial (net_n^l)^*} + \frac{1}{2} (d_n^l - y_n^l)^* \frac{\partial f(net_n^l)}{\partial (net_n^l)^*} \tag{44}$$

l ≤ L - 1 olduğunda

$$\left(\sum_k (w_{km}^{l+1})^* \delta_k^{l+1} \right) \frac{\partial f(net_n^l)^*}{\partial (net_n^l)^*} + \left(\sum_k (w_{km}^{l+1}) (\delta_k^{l+1})^* \right) \frac{\partial f(net_n^l)}{\partial (net_n^l)^*}$$

Karmaşık gradyan iniş algoritması için bağıntı (44)'de verilen güncelleme kuralları, hem tam hem de ayırık aktivasyon fonksiyonları için geçerlidir. İlgili formüllere momentum terimi de eklenebilir. Momentum, modelin öğrenme sürecinde önceki adımlardaki ağırlık değişimlerini dikkate almasını sağlar, böylece güncellemeler daha kararlı bir şekilde yapılır ve hızlıca optimize edilir (Sutskever vd., 2013).

1.11.3. Derin Öğrenme Modelinde Hiperparametreler

Derin öğrenme modelinde kullanılan hiperparametreler aşağıdaki gibi sıralanabilir.

- Derin öğrenme uygulamalarında veri setinin büyüklüğü ve çeşitliliği, modelin öğrenme başarısını doğrudan etkileyen önemli faktörlerdir. Genel olarak, veri seti büyüdükçe modelin performansı da artar, ancak bu artış sonsuza kadar doğru orantılı şekilde devam etmez. Belirli bir noktadan sonra, veri seti boyutunun artırılması sistemin başarısına sınırlı katkı sağlar ve hatta modelin karmaşıklığı yeterli değilse performans düşüşü yaşanabilir. Ayrıca yalnızca veri setinin büyük olması yeterli değildir; verilerin çeşitliliği de en az boyut kadar önemlidir. Çeşitlilik, modelin daha genelleştirilebilir özellikler öğrenmesine olanak tanır ve başarıyı artırır.

Küçük veri setleri ile çalışılırken, sentetik veri üretimiyle veri seti genişletilebilir veya transfer öğrenme gibi yöntemler kullanılarak büyük ölçekli veri setlerinde eğitilmiş modellerden öznitelik transferi yapılabilir. Bu yöntemler, özellikle kaynakların sınırlı olduğu veya yeterince paylaşımın bulunmadığı durumlarda etkili bir çözüm sunar. Ancak, görsel veriler gibi çok özel alanlarda her sınıf için binlerce örnek gereklidir.

Sonuç olarak veri setinin büyüklüğü ve çeşitliliği arasındaki dengeyi sağlamak, eğitim sürecini etkili bir şekilde yönetmek için kritik bir adımdır. Eğitimin sık yapılmayacağı ve depolama alanının sorun olmadığı durumlarda öğrenim başarısı öncelikli olurken, mobil ortamlar gibi depolama alanının problem olduğu durumlarda veri seti boyutunun daha dikkatli değerlendirilmesi gerekir.

- Öğrenme oranı, modelin her iterasyon sırasında ağırlıklarda ne kadar değişiklik yapılacağını belirleyen kritik bir hiperparametredir. Çok büyük bir öğrenme oranı modelin en uygun çözümünü aşarak salınma neden olabilirken, çok küçük bir değer ise eğitim sürecini yavaşlatabilir ve modelin lokal optimum bir değere takılarak global optimuma ulaşamamasına yol açabilir. Bu nedenle öğrenme oranının doğru bir şekilde ayarlanması, modelin eğitim başarısı için büyük önem taşır. Genellikle varsayılan olarak 0.01 gibi bir değer kullanılır ve belli bir eğitim adımından sonra bu değer kademeli olarak azaltılır. Bu yöntem, modelin daha genel bir çözüm bulmasına yardımcı olurken aşırı öğrenmeyi engeller ve eğitim sürecinin verimliliğini artırır.

- Momentum katsayısı, gradyan inişi gibi optimizasyon yöntemlerinde kullanılan önemli bir hiperparametredir. Momentum, gradyan inişini hızlandırmak ve yerel minimumlara takılmadan daha iyi bir global minimuma ulaşmak için kullanılan bir tekniktir.

Momentum, önceki adımlardan gelen gradyan bilgilerini kullanarak güncellemeyi yapar. Bir başka deyişle, her iterasyonda ağırlıklar sadece mevcut gradyanla değil, aynı zamanda önceki iterasyonlarda elde edilen gradyanlarla da güncellenir. Bu, modelin daha hızlı ve daha kararlı bir şekilde öğrenmesini sağlar.

Momentum katsayısı, genellikle 0 ile 1 arasında bir değer alır ve önceki iterasyonların katkısının ne kadar olacağını belirler. Eğer momentum katsayısı 0'a yakınsa, model daha geleneksel bir gradyan inişi gibi çalışır ve sadece mevcut gradyan bilgisine dayanır. Eğer momentum katsayısı 1'e yakınsa, önceki gradyan bilgileri daha fazla dikkate alınır ve bu da daha hızlı bir öğrenme süreci sağlayabilir.

Momentum katsayısının doğru seçilmesi, modelin eğitim sürecinin kararlılığını artırır ve daha hızlı bir optimizasyon sağlar. Yanlış bir momentum değeri, modelin eğitimi sırasında salınımlar veya aşırı adımlar gibi problemlere yol açabilir. Bu yüzden, momentum katsayısının doğru seçilmesi, öğrenme sürecinde kritik bir rol oynar.

Momentum sabiti genellikle 0.9 gibi bir değere ayarlanır, ancak bu değer problem ve veri setine göre değiştirilebilir.

- Derin öğrenme modellerinde eğitim verisi, tüm veri seti olarak modele verildiğinde parametrelerin güncellenmesi süreci zaman alıcı ve maliyetli olabilir. Bu problemi çözmek için veri seti küçük parçalara bölünür. Bu parçalara *mini-batch* denir ve her mini-batch, modelin parametrelerini güncellemek için kullanılır. Bu parçaların büyüklüğü, *batch size* olarak bilinen bir hiperparametreyle belirlenir.

Eğitim sürecinde, parametreler her mini-batch üzerinde yapılan geri yayılım işlemiyle güncellenir. Her iterasyonda gradyan inişi kullanılarak ağırlıklar değiştirilir. Eğer veri seti çok büyükse, tüm veri üzerinde işlem yapmak çok daha fazla zaman ve kaynak gerektirir. Bu nedenle veriler küçük parçalara ayrılır ve her bir parçanın işlenmesi sonrasında modelin parametreleri güncellenir.

Batch size seçimi, eğitim sürecinin verimliliğini doğrudan etkiler. Batch size 1 olarak seçildiğinde, model her örnekle birlikte parametrelerini günceller ve bu süreç SGD olarak adlandırılır. Ancak veriler arasındaki büyük farklar tahmin sonuçlarında önemli değişimlere yol açabilir, bu da genellikle istenmeyen bir durumdur. Öte yandan, batch size tüm veri setinin büyüklüğü kadar seçildiğinde modelin güncellenmesi çok daha maliyetli hale gelir.

Batch size belirlenirken veri setinin büyüklüğü, veri dağılımı ve kullanılan makinenin işlem gücü gibi faktörler göz önünde bulundurulmalıdır. Uygun batch size seçimi, eğitim

süresini kısaltarak modelin daha verimli öğrenmesini sağlar ve veri setinin büyüklüğüne uygun şekilde ayarlanması eğitimin başarısını artırır. (Kandel & Castelli, 2020).

- İterasyon, eğitim verisinin model tarafından bir kez işlenmesi anlamına gelir. Eğitim sürecinde genellikle birden fazla iterasyonla model eğitilir. İterasyon sayısının fazla olması modelin daha fazla öğrenmesine olanak tanır, ancak aşırı öğrenmeye de neden olabilir. Aşırı öğrenmeyi önlemek için erken duraklama mekanizmaları veya çapraz doğrulama gibi teknikler kullanılabilir. İterasyon sayısının uygun bir şekilde belirlenmesi, modelin genelleme yeteneğini, doğruluğunu artırır ve eğitim süresini etkileyebilir.

- Derin öğrenme ağlarında gizli katman sayısı ve her katmandaki nöron sayısının artırılması, modelin kapasitesini yükselterek daha karmaşık problemlerin öğrenilmesine olanak tanır. Ancak, katman sayısının artmasıyla geri yayılımın etkisi ilk katmanlara daha az ulaşabilir ve bu durum belirli bir noktadan sonra ek katmanların sağladığı katkının azalmasına neden olabilir. Benzer şekilde, nöron sayısının fazla olması modelin hesaplama zamanı ve bellek ihtiyacını artırırken, özellikle GPU olmayan sistemlerde bu durum problem yaratabilir. Öte yandan yetersiz nöron sayısı, modelin yetersiz uyum göstermesine yol açabilir. Bu nedenle, modelin performansını optimize etmek için gizli katmanların ve nöron sayısının dikkatlice ayarlanması kritik öneme sahiptir. İlk katmanlarda fazla, sonraki katmanlarda ise giderek azalan nöron sayısı kullanmak gibi düzenleme teknikleri, aşırı öğrenme riskini azaltırken modelin genel başarısını artırabilir.

- Modelin başlangıç ağırlıkları genellikle küçük rastgele değerlerle başlatılır. Başlangıç ağırlıkları, modelin eğitim sürecindeki kararlılığı ve doğru çözümü bulma yeteneğini etkileyebilir. Bu bağlamda, modelin doğru çözüme ulaşmasını hızlandırabilir ve genel model performansını artırabilir.

1.11.4. Karmaşık Değerli Aktivasyon Fonksiyonları

Karmaşık değerli sinir ağlarında kullanılan aktivasyon fonksiyonları iki ana kategoriye ayrılabilir. Bu iki aktivasyon fonksiyonu tipi, CVNN'lerin çalışma prensiplerini belirleyen temel unsurlardandır (Lee vd., 2022).

İlk kategori, ayrık aktivasyon fonksiyonlardır. Bu fonksiyonlar, karmaşık sayıların gerçek ve sanal bileşenlerini birbirinden bağımsız olarak işler. Bir başka deyişle, her bileşen ayrı ayrı işlenir ve aktivasyon fonksiyonuna uygulanır. Ayrık aktivasyon fonksiyonları Tip-A ve Tip-B olarak da iki alt kategoriye ayrılır. İlgili alt kategoriler için formüller bağıntı

(45)'de ifade edilmiştir. Bağntı (45)'de f_A, f_B karmaşık değerli fonksiyonlar, $f_{Re}, f_{Im}, f_r, f_\theta$ ise geleneksel türdeki (sigmoid, tanh, ReLU gibi) gerçek değerli aktivasyon fonksiyonlarıdır.

$$Tip - A \Rightarrow f_A(z) = f_{Re}(x) + if_{Im}(y)$$

$$Tip - B \Rightarrow f_B(z) = f_r(|z|) + if_\theta(\arg(z)) \quad (45)$$

$$z = x + iy = |z|\exp(i \arg(z)) , z \in \mathbb{C} \text{ and } x, y \in \mathbb{R}$$

İkinci kategori ise bütünleşik aktivasyon fonksiyonlarıdır. Bu fonksiyonlar, karmaşık sayıların gerçek ve sanal bileşenlerini tek bir bütün olarak ele alır ve işlem yaparken bu bileşenleri ayırmaz. Bu sayede, karmaşık sayıların tam yapısından yararlanarak daha bütüncül bir yaklaşım sergiler.

Sigmoid aktivasyon fonksiyonu genellikle yapay sinir ağlarında kullanılan bir aktivasyon fonksiyonudur. Matematiksel olarak sigmoid fonksiyonu bağntı (46)'da ifade edilmiştir (Benvenuto & Piazza, 1992).

$$\text{sigmoid}(x) = \frac{1}{1 + e^{-x}} \quad (46)$$

İlgili bağntıda x girdi değerini temsil eder. Sigmoid fonksiyonu, girdi değerini (genellikle $-\infty$ ile $+\infty$ arasında) 0 ile 1 arasına sıkıştırır. Bu özellikle sınıflandırma problemlerinde kullanışlıdır, çünkü çıktı değeri olasılık olarak yorumlanabilir.

Sigmoid fonksiyonunun dezavantajlarından biri, büyük girdi değerleri için gradyanın çok küçük olmasıdır, bu da geriye yayılım sürecinde gradyanın kaybolmasına yol açabilir. Bu nedenle, çok katmanlı ağlarda veya derin öğrenme modellerinde tercih edilmeyebilir.

tanh aktivasyon fonksiyonu, yapay sinir ağlarında sıklıkla kullanılan bir başka aktivasyon fonksiyonudur. Fonksiyonun matematiksel karşılıkları bağntı (47)'de ifade edilmiştir (Qiumei vd., 2019).

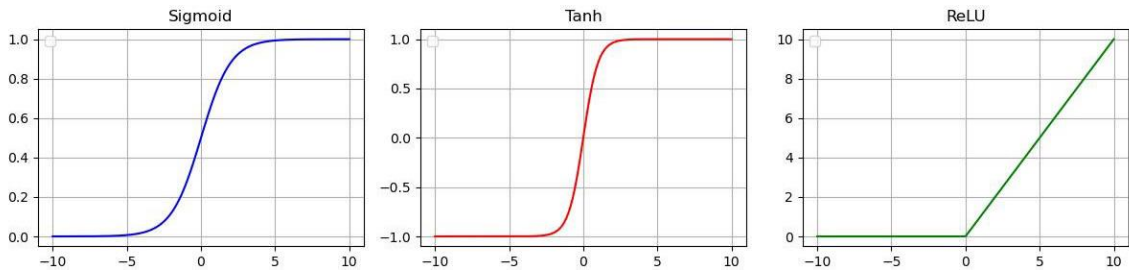
$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad \text{veya} \quad \tanh(x) = \frac{2}{1 + e^{-2x}} - 1 \quad (47)$$

tanh fonksiyonu, girdi değerini -1 ile 1 arasına dönüştürür. Böylece fonksiyon sıfır merkezli bir çıktı aralığı sağlar, bu da bazı durumlarda daha hızlı ve verimli öğrenmeyi destekleyebilir. tanh fonksiyonu, büyük pozitif veya negatif girişler için gradyanların çok küçük hale gelmesine neden olabilir. Ancak sigmoid fonksiyonundan farklı olarak, tanh daha geniş bir gradyan aralığına sahiptir ve sıfır merkezli olması gradyanların daha dengeli bir şekilde yayılmasına yardımcı olabilir. Sigmoid fonksiyonunun aksine, tanh sıfır merkezli bir çıktıya sahip olduğundan özellikle bazı aktivasyon fonksiyonları arasında geçiş yaparken öğrenme süreçlerini iyileştirebilir.

ReLU aktivasyon fonksiyonu, derin öğrenme ve yapay sinir ağlarında oldukça popüler bir aktivasyon fonksiyonudur. Matematiksel olarak, ReLU fonksiyonu bağıntı (48)'de gösterildiği şekilde ifade edilir (LeCun vd., 2015).

$$ReLU(x) = \max(0, x) \quad (48)$$

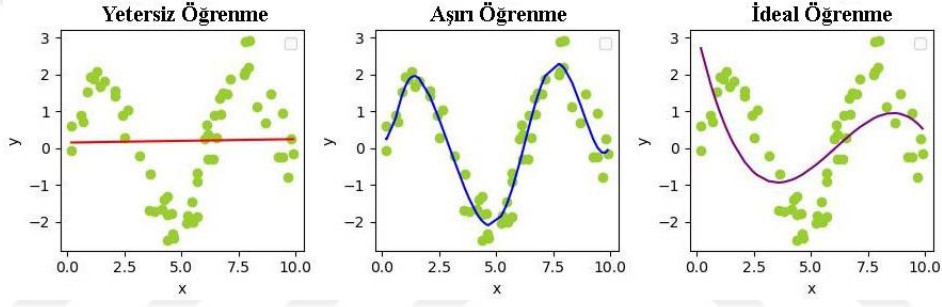
İlgili bağıntı, giriş değeri x pozitifse x 'in kendisini, negatifse 0'ı döndürür. ReLU fonksiyonu oldukça basit ve hesaplaması hızlıdır, çünkü sadece bir karşılaştırma ve maksimum seçimi gerektirir. Bu durum, büyük ağlarda hesaplama verimliliğini artırabilir. ReLU fonksiyonu, negatif bölgedeki gradyanları sıfırlar. Ancak pozitif bölgedeki gradyanlar sabit kalır ve bu durum derin ağlarda gradyanların daha etkili bir şekilde yayılmasına yardımcı olabilir. ReLU sıfır merkezli değildir, çıkış değeri yalnızca sıfır ya da pozitif olabilir. Bu nedenle, ReLU genellikle derin öğrenme modellerinde tercih edilen bir aktivasyon fonksiyonudur. ReLU'nun dezavantajları arasında "ölü nöron" problemi bulunmaktadır. Bazı nöronlar negatif girişler nedeniyle sürekli olarak sıfır çıkış üretir ve bu durum modelin öğrenme yeteneğini sınırlayabilir. Bu tür sorunları aşmak için Leaky ReLU, Parametric ReLU gibi varyasyonlar geliştirilmiştir (Liang & Hu, 2021).



Şekil 39. Aktivasyon fonksiyonları

1.11.5. Seyreltme Oranı

Seyreltme veya *dropout* oranı, derin öğrenme modellerinde aşırı öğrenmeyi önlemek amacıyla kullanılan etkili bir düzenleme tekniğidir. Modelin sadece eğitim verisine değil, genel verilere de iyi genelleme yapmasını sağlar. Bu yöntem, modelin veri setini ezberlemesi yerine genelleştirme yeteneğini artırarak, yeni veriler üzerinde daha iyi performans göstermesini sağlar. Aynı zamanda, seyreltme oranı dikkatli bir şekilde ayarlanmadığında yetersiz öğrenme riski oluşturabileceği için doğru kullanımına özen gösterilmelidir. Yetersiz öğrenme, modelin eğitim verisindeki desenleri doğru şekilde öğrenememesi durumudur ve modelin çok basit veya yetersiz bir yapıya sahip olmasından kaynaklanır. Şekil 40, yetersiz öğrenme, aşırı öğrenme ve ideal öğrenmeyi temsil eden grafikleri göstermektedir.

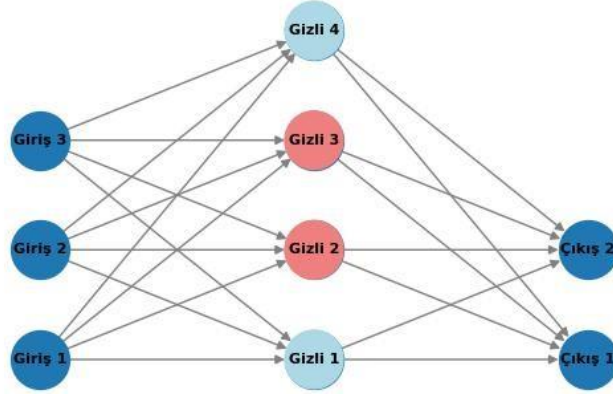


Şekil 40. Yetersiz öğrenme, aşırı öğrenme ve ideal öğrenmeyi temsil eden grafikler

Seyreltme yöntemi, her bir iterasyonda sinir ağındaki bazı nöronları rastgele devre dışı bırakır, bir başka deyişle o nöronların çıktıları sifıra eşitlenir. Bu işlem, modelin bir alt kümesinin her defasında eğitilmesini sağlar bu da modelin belirli özelliklere aşırı bağımlı olmasını önler (Srivastava vd., 2014). Şekil 41, seyreltme tekniğine bağlı olarak devre dışı bırakılan nöronları ve ağ yapısını göstermektedir. Gizli2 ve Gizli3 isimli nöronlar eğitimin bir adımında devre dışı kalan nöronları ifade etmektedir.

Eğitim sırasında, teknik gereğince her katmandaki nöronların belirli bir oranı rastgele seçilerek geçici olarak sıfırlanır. Bu oran genellikle %20-%50 arasında olur. Test aşamasında ise seyreltme uygulanmaz. Bunun yerine, eğitim sırasında devre dışı bırakılan nöronların etkisini telafi etmek için nöronların aktivasyonları seyreltme oranına göre ölçeklendirilir.

Seyreltme, modelin daha fazla çeşitlilikte öğrenmesine neden olduğu için eğitim süresini uzatabilir. Her bir eğitim adımında bazı nöronlar devre dışı bırakıldığından, bu süreç sırasında hesaplama yükü artabilir.



Şekil 41. Seyreltme tekniğine bağlı olarak devre dışı bırakılan nöronlar ve ağ yapısı

1.11.6. Softmax Çıkış Fonksiyonu

Softmax, genellikle sınıflandırma problemlerinde kullanılan bir aktivasyon fonksiyonudur. Çok sınıflı sınıflandırmalarda, modelin her bir sınıfa ait olasılığını belirlemek için kullanılır. Softmax fonksiyonu, giriş vektöründeki her değeri pozitif bir olasılık değeri haline getirir ve tüm çıktılarının toplamı 1 olacak şekilde normalize edilir. Softmax fonksiyonu bağıntı (49)'da olduğu gibi ifade edilir (Kagalkar & Raghuram, 2020). İlgili bağıntıda x_i giriş vektöründeki i . öğeyi, K sınıf sayısını temsil eder. e^{x_i} giriş değerlerinin üstel fonksiyonla dönüştürülmüş halidir.

$$Softmax(x_i) = \frac{e^{x_i}}{\sum_{j=1}^K e^{x_j}} \quad (49)$$

Softmax, sınıflandırma problemlerinde her bir sınıf için tahmin edilen değeri olasılık olarak yorumlamayı mümkün kılar. Bu olasılıklar 0 ile 1 arasında olur ve tüm sınıfların olasılıklarının toplamı 1'e eşittir. Softmax, genellikle sinir ağının son katmanda kullanılır ve birden fazla sınıfın bulunduğu sınıflandırma problemlerinde etkin bir çözüm sunar. Model, her bir sınıfa ait olasılığı üretir ve en yüksek olasılığa sahip sınıf, tahmin edilen sınıf olarak seçilir. Softmax, büyük giriş değerleri ile küçük giriş değerleri arasındaki farkı büyütür. Yani, en büyük giriş değeri daha yüksek bir olasılıkla temsil edilirken, diğer sınıfların olasılıkları küçülür. Softmax, sinir ağları ve lojistik regresyon gibi algoritmalarda, birden fazla sınıfa ait verilerin sınıflandırılmasında yaygın olarak kullanılır. Doktora çalışmaları

kapsamında geliştirilen modelde, Şekil 38'in son katmanı olan çıkış katmanında softmax fonksiyonu kullanılmıştır.

1.11.7. Karmaşık Değerli XOR Problemi

Bu çalışmada, karmaşık değerli verilerin sınıflandırılmasını sağlayacak sistemin doğruluğunu test etmek amacıyla, Nitta tarafından tanımlanan karmaşık değerli XOR problemi verileri kullanılmıştır (Nitta, 1997). Giriş verisi, karmaşık değerli veri türü olarak $0, j, 1, 1 + j$ şeklinde değer kümesine sahiptir. Tablo 5, iki girişli ve tek çıkışlı kompleks bir ağ için kompleks değerli giriş verilerini ve bunlara karşılık gelen çıkış değerlerini listeler.

Tablo 5. Belirtilen kurallara göre karmaşık değerli XOR için eğitim örnekleri

Giriş 1	Giriş 2	Çıkış
0	0	1
0	j	j
j	j	$1+j$
j	1	j
1	1	$1+j$
j	0	0
$1+j$	$1+j$	1
$1+j$	j	j

Çıkış verisinin üretilmesinde kullanılan iki kural maddeler halinde aşağıda ifade edilmiştir.

- Giriş olarak alınan iki karmaşık değerli veri eşitse, karmaşık değerli çıkış verisinin gerçek kısmı 1, aksi takdirde 0 olur.
- İkinci karmaşık değerli veri girişi 1 ya da j ise, karmaşık değerli çıkış verisinin sanal kısmı 1, aksi takdirde 0 olur.

Standart XOR işleminde tek çıkışla iki sınıf temsil edilebilirken Tablo 5’den görüleceği üzere karmaşık değerli XOR tek çıkışla dört sınıfı temsil edilebilmektedir. Bu sayede çok sayıda farklı sınıf için gereken çıkış sayısı oldukça azaltılabilir.



2. YAPILAN ÇALIŞMALAR

Bu bölümde doktora süreci boyunca gerçekleştirilen çalışmalar ayrı ayrı ele alınarak incelenecektir. İlgili çalışmalar; CZM'ler ile özellik (karmaşık değerli öznitelik vektörünün elde edilmesi) çıkarımı, öznitelik vektöründen görüntünün yeniden oluşturulması, bir CVDNN'in geliştirilmesi, XOR probleminin geliştirilen karmaşık değerli derin sinir ağı ile çözülmesi, MNIST veri seti üzerinde gerçekleştirilen çalışmalar, MUD veri seti üzerinde gerçekleştirilen çalışmalar ve önerilen yöntemi içeren akıllı bir sistemin geliştirilme aşamaları olmak üzere yedi alt başlıkta incelenecektir. Gerçekleştirilen çalışmalar sırasında görüntüler üzerinde herhangi bir ön işlem (el tespiti, arka plan çıkarımı, gürültü azaltma, veri güçlendirme vs.) uygulanmamıştır. Elde edilen başarı oranlarının doğrudan ham görüntüler üzerinden elde edildiğinin vurgulanması gerekmektedir.

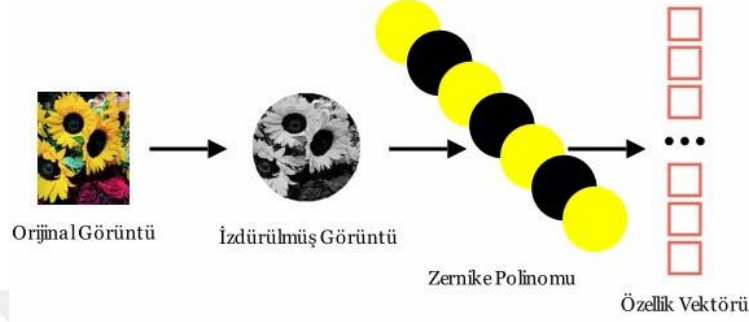
2.1. Karmaşık Zernike Momentleri Kullanılarak Özellik Çıkarımı

Karmaşık Zernike momentleri, özellik çıkarımında kullanılan son derece etkili yöntemlerden biridir. Öyle ki, herhangi bir görüntü için, CZM'ler kullanılarak elde edilen öznitelik vektöründen görüntünün yeniden oluşturulması mümkündür. Öznitelik vektörünün boyutunu belirleyen ve görüntünün yeniden oluşturulmasında kullanılan iki önemli parametre sırasıyla CZM'lerin derecesi n ve tekrar sayısıdır m . Her iki parametrenin de belirlenmesi gerekmektedir. Bu bağlamda n derecesine bağlı olarak tekrar sayısı m belirlenir.

Derece n , hem çıkarılan özniteliklerin sayısını (öznitelik vektörünün boyutunu) hem de ASL tanıma için geliştirilen karmaşık değerli derin sinir ağının performansını doğrudan etkiler. Derece arttıkça elde edilen öznitelik vektörünün boyutu ve işlem süresi orantılı olarak artar. Öznitelik çıkarımına fazla zaman harcamak, gerçek zamanlı uygulamalarda performans düşüşüne yol açabilir. Öte yandan daha fazla özneliğe sahip olmak, görüntünün yeniden oluşturulmasında olumlu bir etki yaratır.

Tekrar sayısı m , 0 ile n arasında değişir. Negatif tekrar değerleri, pozitif değerlerin eşlenikleri olduğu için hesaplama yapılmadan elde edilebilir. Negatif değerler sınıflandırma modellerinde değil genellikle görüntünün yeniden oluşturulmasında kullanılır.

Geliştirilen sistemde, MNIST veri setinden rastgele seçilen her bir görüntü için farklı derecelerde özellik çıkarımı ve görüntü yeniden oluşturma işlemleri gerçekleştirilerek en uygun dereceyi belirlemek hedeflenmiştir. Şekil 42, özellik çıkarımı için işlem adımlarını göstermektedir.



Şekil 42. Özellik çıkarımı için işlem adımları

Orijinal görüntü ile yeniden oluşturulan temsiller üzerinde öncelikli olarak Ortalama Kare Hatası (MSE) ve Yapısal Benzerlik İndeksi (SSIM) gibi benzerlik test prosedürleri uygulanmıştır (Palubinskas, 2016). Ek kontroller sağlamak amacıyla Histogram da ayrıca kullanılmıştır (Bhuiyan & Khan, 2018). MNIST veri seti için elde edilen sonuçların, genellikle düşük derecelerde (12 ile 15 arasında) yüksek benzerlik oranına sahip oldukları tespit edilmiştir. MNIST veri seti üzerindeki çalışmalarda, her görüntü için 64 öznitelik üretecek şekilde derece değeri 14 olarak belirlenmiştir.

Özellik çıkarımı sürecinde, MNIST veri seti için ilgili CSV dosyası verileriyle aynı (kayıpsız) değere sahip PNG formatında görüntüler oluşturulmuştur. n ve m değerleri belirlendikten sonra, eğitim ve test görüntüleri için bağıntı (32), (33), (34) ve (35)'ü kullanarak karmaşık değerli verilerden ($z = x + iy, z \in \mathbb{C}$ ve $x, y \in \mathbb{R}$) oluşan öznitelik vektörleri elde edilmiştir. Benzer şekilde aynı işlemler MUD veri seti üzerinde uygulanmıştır. MNIST veri setindeki her bir öge için özellik çıkarımında harcanan süre yaklaşık 0.082 saniyedir. MNIST veri setindeki görüntülerin 28x28 piksel gibi oldukça küçük boyutlarda olduğunu yeniden hatırlatmak gerekmektedir. MUD veri setindeki her bir öge için özellik çıkarımı ise yaklaşık 2.33 saniye sürmektedir. Ayrıca, MUD veri setindeki görüntülerin 400x400 piksel boyutunda nispeten büyük olduğunu yeniden belirtmek gerekmektedir.

Karmaşık Zernike momentleri nesnelerin daha karmaşık özelliklerini temsil etmek ve ek bilgi sağlamak için kullanılır. CZM'ler, Zernike momentlerinin genliği kullanmaktan daha kapsamlı ve ayrıntılı bir özellik temsili sunar. Esasında karmaşık Zernike momentleri olarak ifade edilen terim, Zernike momentlerinin büyüklüğünü kullanmaktan farklı olarak nesnelerin dairesel simetri bilgisinin yanı sıra aynı zamanda faz bilgisini de dikkate alarak daha ayrıntılı bir analiz sağlamasından ileri gelmektedir. Bu özellikleri nedeniyle, yüz tanıma gibi karmaşık desen tanıma görevlerinde genellikle daha etkili olurlar. Geliştirilen modelin (özellik çıkarımı ve CVDNN ile sınıflandırma) genel yapısını koruyarak gerçek sayılarla çalışacak şekilde (gerçek değerli derin sinir ağı) yapılandırılmasıyla karşılaştırıldığında, karmaşık sayılarla çalışmanın performans üzerinde yaklaşık %20 oranında olumlu bir etki yarattığı gözlemlenmiştir. Bu etki oranı, verilerin karmaşık değerli olarak kullanılmasının başarıyı önemli ölçüde etkilediğini açıkça ortaya koymaktadır.

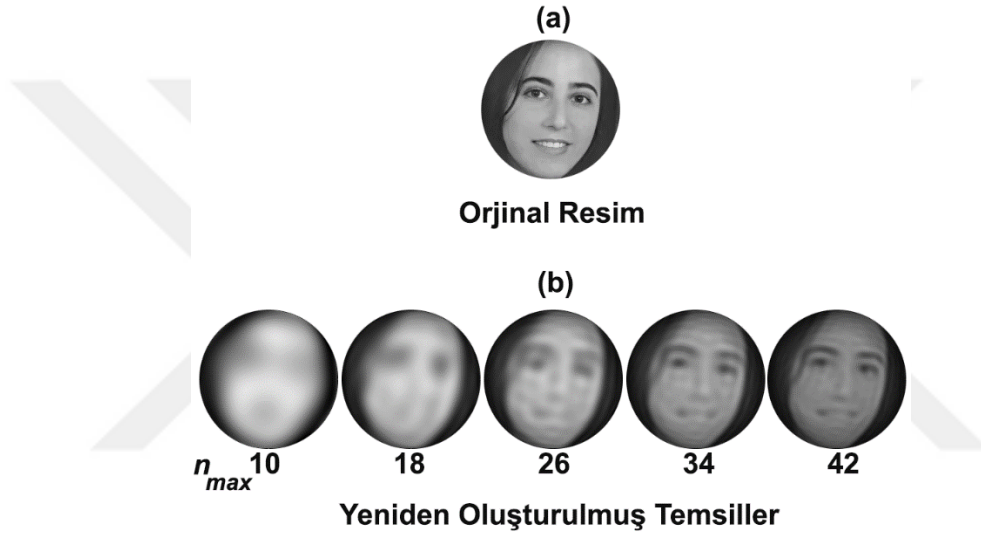
2.2. Öznitelik Vektöründen Görüntünün Yeniden Oluşturulması

Bir görüntünün yeniden oluşturulabilmesi için öncelikle görüntüden özellik çıkarımı yapılması gerekmektedir. Bu nedenle, özellik çıkarımında kullanılacak yöntem büyük önem taşımaktadır. Karmaşık Zernike momentlerinin, görüntünün yeniden oluşturulmasında oldukça etkili olduğu genel bilgiler bölümünde vurgulanmıştır. Görüntünün yeniden oluşturulmasındaki bu başarısı, aynı zamanda oldukça etkili bir özellik çıkarımı gerçekleştirdiğini de göstermektedir.

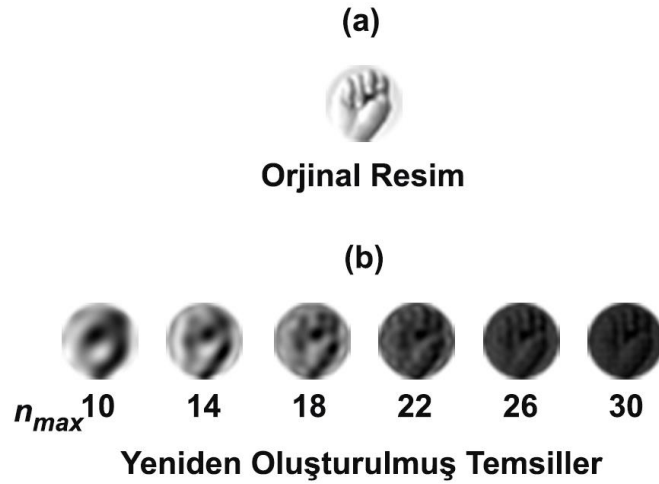
Sınıflandırma amaçlı geliştirilecek bir model ile görüntünün yeniden üretilmesi amaçlı geliştirilecek model arasında, özellik çıkarımında kullanılacak moment derecesinin seçimi konusunda bir zıtlık bulunmaktadır. Şöyle ki, daha önce belirtildiği üzere sınıflandırma amaçlı geliştirilen modelde moment derecesinin düşük olması hem hız hem de tespit açısından olumlu sonuçlar verirken, görüntünün yeniden oluşturulmasında moment derecesinin düşük olması üretilen görüntünün kalitesini düşürmektedir. Bu bağlamda, eğer görüntünün yeniden oluşturulması birincil hedefse moment derecesi yüksek bir değer olarak seçilmelidir. Ancak, moment derecesinin artması zaman ve kaynak tüketimini de yükseltmektedir. Görüntünün yeniden oluşturulması aşamasında, negatif m tekrarlarına bağlı olarak elde edilen karmaşık değerli veriler de hesaba katılmalıdır.

Şekil 43, 200x200 piksel bir görüntüyü ve farklı moment dereceleri baz alınarak yeniden oluşturulan temsillerini göstermektedir. Şekil 44 ise MNIST veri setinden rastgele

seçilen 28x28 piksel bir görüntüyü ve ilgili görüntü için dereceye bağlı olarak yeniden oluşturulmuş temsillerini göstermektedir. Her iki şekilden de görüleceği üzere moment derecesi arttıkça, üretilen temsillerin kalitesi (netliği) de giderek artmaktadır. Zernike momentler birim disk üzerinde tanımlandığı için, hem özellik çıkarımı hem de özniteliklerden oluşturulan temsiller birim disk üzerinde ifade edilir. Bu nedenle, her iki şekilden de görüleceği üzere hem orijinal görüntüler hem de temsilleri birim disk olarak gösterilmiştir. Karmaşık Zernike momentlerinin görüntünün yeniden oluşturulmasındaki başarısı, yüz ve desen gibi detay gerektiren alanlarda rahatlıkla kullanılabileceğini gösterir.



Şekil 43. (a) 200x200 piksel orijinal giriş görüntüsü (b) orijinal görüntü için farklı derecelerde yeniden oluşturulan görüntüler



Şekil 44. (a) MNIST veri setinden 28x28 piksel orijinal giriş görüntüsü (b) orijinal görüntü için farklı derecelerde yeniden oluşturulan görüntüler

2.3. CVDNN Mimarisi

Genel bilgiler başlığında, karmaşık Zernike momentlerle elde edilen öznitelik vektörünü işleyebilecek bir sinir ağı geliştirilmesi gerektiği belirtilmiştir. Geliştirilen sinir ağı, en az iki gizli katmana sahip olduğu için karmaşık değerli bir derin sinir ağı olarak adlandırılır. Derin sinir ağının geliştirilmesi sürecinde, üç gizli katman da denenmiş ancak iki gizli katmana kıyasla fark edilecek düzeyde bir iyileşme sağlamadığı için tercih edilmemiştir. Ağın içindeki tüm katmanlar karmaşık değerli verilerle çalışabilecek şekilde tasarlanmıştır.

Özellik çıkarımında, seçilen moment derecesi 14'e bağlı olarak 64 uzunluklu bir vektör elde edildiği daha önce belirtilmiştir. Elde edilen öznitelik vektörüne bağlı olarak 64 giriş alan derin sinir ağı, ilk gizli katmanda 700 nöron, ikinci gizli katmanda 400 nöron kullanmaktadır. Veri setlerine bağlı olarak, çıkış katmanında MNIST veri seti için 25, MUD veri seti için ise 36 çıkış üretilmektedir. MNIST veri setinde, J hariç olmak üzere [A-Y] harfleri için normalde 24 çıkış bulunmaktadır. J ve Z harfleri ASL'de hareketli görüntü olduğundan çıkış kümesine dahil edilmezler. Veri setinde J harfi bulunmamasına rağmen, harfler A'dan başlayarak Y'ye kadar 0'dan 24'e kadar numaralandırılmıştır. Dolayısıyla, normalde 24 sınıf için 25 çıkış oluşturulmuştur. Z harfinin çıkarılmasının, J harfi gibi bir etki yaratmayacağı açıktır.

Bir CVDNN'nin en kritik bölümlerinden biri uygun aktivasyon fonksiyonunun belirlenmesidir. Liouville teoremi, her noktada analitik ve sınırlı olan bir karmaşık değerli fonksiyonun sabit olduğunu belirtir (Hansen, 2008). CVDNN aktivasyon fonksiyonları ayrık ve tam olmak üzere iki kategoriye ayrılır. Ayrık aktivasyon fonksiyonları, karmaşık bileşenlere bağımsız olarak gerçek değerli geleneksel aktivasyon fonksiyonlarını uygular ve sonuçları birleştirerek karmaşık değerli bir nöron çıktısı üretir. Bu nedenle, bu aktivasyon fonksiyonları sınırlı olabilir ancak analitik olmayabilir. Ayrık aktivasyon fonksiyonları, Tip-A ve Tip-B olarak iki alt türe ayrılır (Bassey vd., 2024). Gerçekleştirilen doktora çalışmasında, bağıntı (50)'de gösterilen Tip-A ayrık aktivasyon fonksiyonu kullanılmaktadır. İlgili bağıntıda, f_A karmaşık değerli bir fonksiyon olup f_{Re} ve f_{Im} gerçek değerli aktivasyon fonksiyonlarıdır. Çalışmada, gerçek değerli aktivasyon fonksiyonu olarak ReLU kullanılmaktadır (Liang & Hu, 2021). Çıkış sınıfları dışında, her iki veri seti için de aynı CVDNN yapısı kullanılmaktadır.

$$f_A(z) = f_{Re}(x) + if_{Im}(y) \quad (50)$$

Aşırı öğrenmeyi önlemek amacıyla, ikinci gizli katmanda seyreltme tekniği uygulanmaktadır (Srivastava vd., 2014). Seyreltme tekniği sabit olarak uygulanmış ve en iyi sonuçlar 0.38 değeri ile elde edilmiştir. Eğitim adımına bağlı olarak değişen (azalarak) bir seyreltme tekniği denenmiş olsa da, dikkate değer bir etki sağlamadığından tercih edilmemiştir.

Karmaşık değerli geri yayılım algoritması, mini-batch eğitim yaklaşımını kullanır. Böylece, modelin genelleştirme yeteneğinin artırılması amaçlanmaktadır. MNIST veri seti için batch size değeri 1000, MUD veri seti için ise 100 olarak belirlenmiştir.

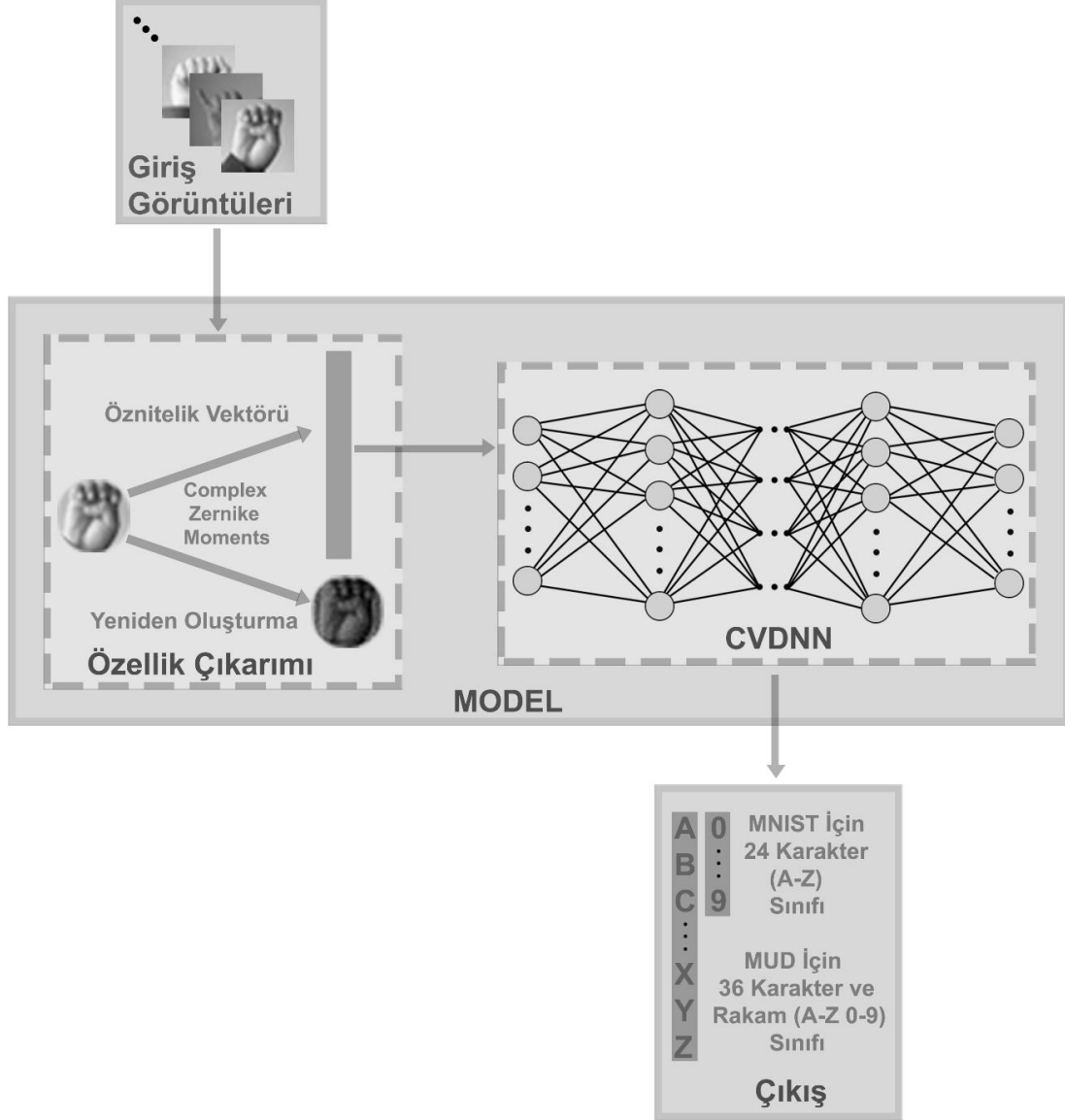
Çıkış katmanını yorumlamak için torch kütüphanesinin softmax fonksiyonu kullanılmaktadır (URL2, 2023).

Karmaşık değerli verilerle çalışabilen optimizasyon algoritmaları arasında Adam, SGD ve RMSProp bulunmaktadır (Kingma & Ba, 2024). Optimizasyon algoritması olarak momentum'lu SGD (SGD with Momentum) tercih edilmiştir. Diğer iki optimizasyon algoritması da iyi ve yakın sonuçlar verse de, en iyi sonuçlar momentum'lu SGD ile elde edilmiştir. Öğrenme oranı 0.01, momentum katsayısı ise 0.9 olarak belirlenmiştir.

MNIST veri setindeki her test maddesinin öznitelik vektöründen çıktı sınıfını belirleme süresi yaklaşık 0.002 saniyedir. Dolayısıyla, modelin bir görüntüyü giriş olarak alıp çıktı sınıfını belirlemesi (özellik çıkarımı ve sınıflandırma dahil olmak üzere) toplamda yaklaşık 0.084 (0.082+0.002) saniye sürmektedir. MUD veri setinde de benzer şekilde, öznitelik vektöründen çıktı sınıfının belirlenmesi yaklaşık 0.002 saniyedir. Böylece MUD veri setindeki bir görüntüyü giriş olarak alıp çıktı sınıfını belirleme süresi toplamda yaklaşık 2.332 (2.33+0.002) saniye olmaktadır. İlgili değerlerden görüleceği üzere, modelin en fazla zaman alan bölümü özellik çıkarım işlemidir.

Modelin başarısını etkileyen önemli etkenlerden biri de özellik çıkarımı sonucunda elde edilen öznitelik vektörünün nasıl normalize edileceğinin belirlenmesidir. Standart normalizasyon tekniklerinden biri olan verilerin 0 ile 1 arasına ölçeklenmesi işlemi beklenen sonuçları vermemiştir. Bu nedenle, doktora çalışmaları kapsamında geliştirilen farklı bir normalizasyon tekniği kullanılmıştır. İlgili teknik gereğince, öncelikli olarak veri kümesindeki en büyük ve en küçük değerler belirlenmiştir. Bu teknik gereğince, tüm veriler -5 ile +5 aralığına ölçeklenecek şekilde sabit bir sayıya bölünmüştür.

Mimarinin belirtilen özellikleri, kapsamlı ve uzun bir deneysel çalışmanın sonucunda tespit edilmiştir. Şekil 45, geliştirilen modelin mimarisini göstermektedir.



Şekil 45. Geliştirilen modelin mimarisi

2.4. Model Mimarisinin Detayları

Bu bölümde, MNIST ve MUD veri setleri için geliştirilen modellerin üretim aşamalarındaki bazı önemli kısımlar incelenecektir. MNIST için geliştirilen modelin eğitimi, veri setinin eğitim kısmının (27455 görüntü) büyüklüğüne bağlı olarak 300 iterasyon

sonunda tamamlanmaktadır. Ancak eğitim kümesi batch size değeri 1000 olacak şekilde alt kümelere ayrıldığında, toplam iterasyon sayısı $(27455/1000) * 300 \Rightarrow 8100$ olarak hesaplanır. MNIST veri seti üzerinden geliştirilen model için, ilk ve son ana iterasyonlar ve bunların alt kümelerindeki iterasyonlar Şekil 46’da belirtilmiştir. Ayrıca, Şekil 46 her bir alt küme için eğitimin başlangıcına ve bitişine dair hata oranlarını göstermektedir.

Train Epoch: 0 [0/ 27455 (0%)]	Loss: 3.257881	Train Epoch: 299 [0/ 27455 (0%)]	Loss: 0.457386
Train Epoch: 0 [1000/ 27455 (4%)]	Loss: 3.219559	Train Epoch: 299 [1000/ 27455 (4%)]	Loss: 0.699468
Train Epoch: 0 [2000/ 27455 (7%)]	Loss: 3.247825	Train Epoch: 299 [2000/ 27455 (7%)]	Loss: 0.604586
Train Epoch: 0 [3000/ 27455 (11%)]	Loss: 3.267477	Train Epoch: 299 [3000/ 27455 (11%)]	Loss: 0.705835
Train Epoch: 0 [4000/ 27455 (15%)]	Loss: 3.271223	Train Epoch: 299 [4000/ 27455 (15%)]	Loss: 0.577954
Train Epoch: 0 [5000/ 27455 (18%)]	Loss: 3.310870	Train Epoch: 299 [5000/ 27455 (18%)]	Loss: 0.585892
Train Epoch: 0 [6000/ 27455 (22%)]	Loss: 3.400144	Train Epoch: 299 [6000/ 27455 (22%)]	Loss: 0.703725
Train Epoch: 0 [7000/ 27455 (25%)]	Loss: 3.511385	Train Epoch: 299 [7000/ 27455 (25%)]	Loss: 0.551020
Train Epoch: 0 [8000/ 27455 (29%)]	Loss: 3.554481	Train Epoch: 299 [8000/ 27455 (29%)]	Loss: 0.597556
Train Epoch: 0 [9000/ 27455 (33%)]	Loss: 3.565644	Train Epoch: 299 [9000/ 27455 (33%)]	Loss: 0.658587
Train Epoch: 0 [10000/ 27455 (36%)]	Loss: 3.588221	Train Epoch: 299 [10000/ 27455 (36%)]	Loss: 0.699735
Train Epoch: 0 [11000/ 27455 (40%)]	Loss: 3.742552	Train Epoch: 299 [11000/ 27455 (40%)]	Loss: 0.705166
Train Epoch: 0 [12000/ 27455 (44%)]	Loss: 3.706336	Train Epoch: 299 [12000/ 27455 (44%)]	Loss: 0.647626
Train Epoch: 0 [13000/ 27455 (47%)]	Loss: 3.758722	Train Epoch: 299 [13000/ 27455 (47%)]	Loss: 0.539945
Train Epoch: 0 [14000/ 27455 (51%)]	Loss: 3.826790	Train Epoch: 299 [14000/ 27455 (51%)]	Loss: 0.687416
Train Epoch: 0 [15000/ 27455 (55%)]	Loss: 3.756641	Train Epoch: 299 [15000/ 27455 (55%)]	Loss: 0.666957
Train Epoch: 0 [16000/ 27455 (58%)]	Loss: 3.694682	Train Epoch: 299 [16000/ 27455 (58%)]	Loss: 0.700340
Train Epoch: 0 [17000/ 27455 (62%)]	Loss: 3.616764	Train Epoch: 299 [17000/ 27455 (62%)]	Loss: 0.596461
Train Epoch: 0 [18000/ 27455 (66%)]	Loss: 3.600613	Train Epoch: 299 [18000/ 27455 (66%)]	Loss: 0.630230
Train Epoch: 0 [19000/ 27455 (69%)]	Loss: 3.645375	Train Epoch: 299 [19000/ 27455 (69%)]	Loss: 0.606959
Train Epoch: 0 [20000/ 27455 (73%)]	Loss: 3.536862	Train Epoch: 299 [20000/ 27455 (73%)]	Loss: 0.612925
Train Epoch: 0 [21000/ 27455 (76%)]	Loss: 3.569918	Train Epoch: 299 [21000/ 27455 (76%)]	Loss: 0.644997
Train Epoch: 0 [22000/ 27455 (80%)]	Loss: 3.508972	Train Epoch: 299 [22000/ 27455 (80%)]	Loss: 0.607796
Train Epoch: 0 [23000/ 27455 (84%)]	Loss: 3.477903	Train Epoch: 299 [23000/ 27455 (84%)]	Loss: 0.589094
Train Epoch: 0 [24000/ 27455 (87%)]	Loss: 3.456936	Train Epoch: 299 [24000/ 27455 (87%)]	Loss: 0.601419
Train Epoch: 0 [25000/ 27455 (91%)]	Loss: 3.422732	Train Epoch: 299 [25000/ 27455 (91%)]	Loss: 0.635184
Train Epoch: 0 [26000/ 27455 (95%)]	Loss: 3.429300	Train Epoch: 299 [26000/ 27455 (95%)]	Loss: 0.607988
Train Epoch: 0 [27000/ 27455 (98%)]	Loss: 3.379064	Train Epoch: 299 [27000/ 27455 (98%)]	Loss: 0.692295

Şekil 46. (a) MNIST veri seti için ilk iterasyon ve alt küme iterasyonları (b) MNIST veri seti için son ana iterasyon ve alt küme iterasyonları

MUD veri setinin eğitim ve test verisi olarak ayrılmadığı daha önce belirtilmiştir. Benzer şekilde, MUD veri seti üzerinden geliştirilen modelde veri setinin %70’i eğitim, %30’u test olarak ayrıldığı daha önce ifade edilmiştir. Böylece modelin eğitimi, veri setinin eğitim kısmının (1762 görüntü) büyüklüğüne bağlı olarak 100 iterasyon sonunda tamamlanmaktadır. MUD veri setinin eğitim kümesi için batch size değeri 100 olarak belirlenmiştir. Bu nedenle, toplam iterasyon sayısı $(1762/100) * 100 \Rightarrow 1762$ olarak hesaplanmaktadır. MUD veri seti üzerinden geliştirilen model için, ilk ve son ana iterasyonlar ile bunların alt kümelerindeki iterasyonlar ve eğitimin başlangıcına ve bitişine dair hata oranları Şekil 47’de belirtilmiştir.

Şekil 48, hem MNIST hem de MUD veri setleri üzerinden eğitilen modeller için ayrı ayrı olmak üzere eğitimin başından sonuna kadar oluşan hata oranlarının grafiğini gösterir.

Train Epoch: 0	[0/ 1761 (0%)]	Loss: 4.126262
Train Epoch: 0	[100/ 1761 (6%)]	Loss: 19.696672
Train Epoch: 0	[200/ 1761 (11%)]	Loss: 12.835375
Train Epoch: 0	[300/ 1761 (17%)]	Loss: 6.255947
Train Epoch: 0	[400/ 1761 (23%)]	Loss: 5.832349
Train Epoch: 0	[500/ 1761 (28%)]	Loss: 5.084908
Train Epoch: 0	[600/ 1761 (34%)]	Loss: 4.237368
Train Epoch: 0	[700/ 1761 (40%)]	Loss: 3.926234
Train Epoch: 0	[800/ 1761 (45%)]	Loss: 3.885735
Train Epoch: 0	[900/ 1761 (51%)]	Loss: 4.691839
Train Epoch: 0	[1000/ 1761 (57%)]	Loss: 4.143806
Train Epoch: 0	[1100/ 1761 (62%)]	Loss: 4.025784
Train Epoch: 0	[1200/ 1761 (68%)]	Loss: 4.930043
Train Epoch: 0	[1300/ 1761 (74%)]	Loss: 4.119783
Train Epoch: 0	[1400/ 1761 (80%)]	Loss: 4.083418
Train Epoch: 0	[1500/ 1761 (85%)]	Loss: 3.888967
Train Epoch: 0	[1600/ 1761 (91%)]	Loss: 4.128245
Train Epoch: 0	[1700/ 1761 (97%)]	Loss: 3.802180

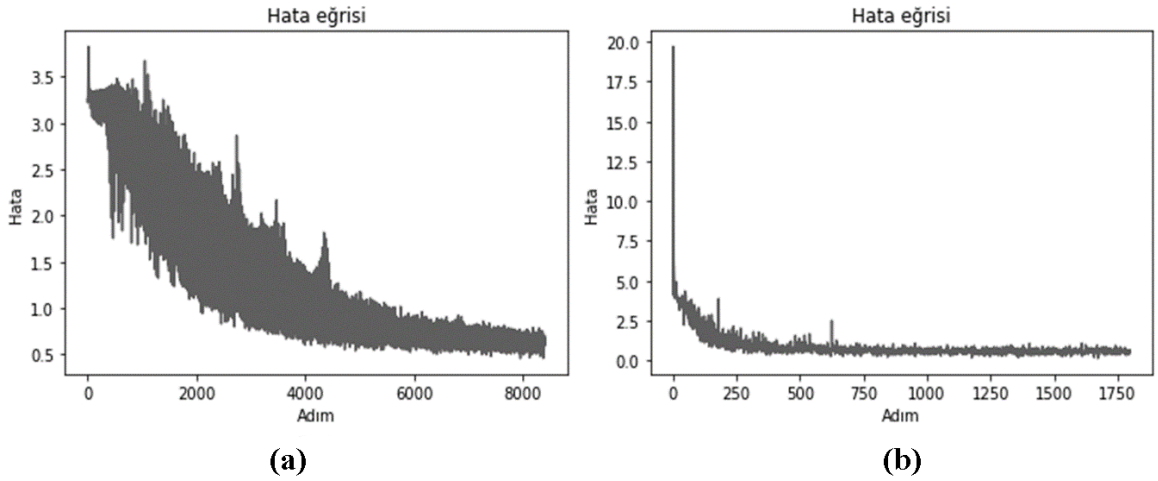
(a)

Train Epoch: 99	[0/ 1761 (0%)]	Loss: 0.355284
Train Epoch: 99	[100/ 1761 (6%)]	Loss: 0.458963
Train Epoch: 99	[200/ 1761 (11%)]	Loss: 0.349081
Train Epoch: 99	[300/ 1761 (17%)]	Loss: 0.500410
Train Epoch: 99	[400/ 1761 (23%)]	Loss: 0.564292
Train Epoch: 99	[500/ 1761 (28%)]	Loss: 0.604224
Train Epoch: 99	[600/ 1761 (34%)]	Loss: 0.610402
Train Epoch: 99	[700/ 1761 (40%)]	Loss: 0.482233
Train Epoch: 99	[800/ 1761 (45%)]	Loss: 0.481522
Train Epoch: 99	[900/ 1761 (51%)]	Loss: 0.591458
Train Epoch: 99	[1000/ 1761 (57%)]	Loss: 0.339178
Train Epoch: 99	[1100/ 1761 (62%)]	Loss: 0.538059
Train Epoch: 99	[1200/ 1761 (68%)]	Loss: 0.529163
Train Epoch: 99	[1300/ 1761 (74%)]	Loss: 0.646374
Train Epoch: 99	[1400/ 1761 (80%)]	Loss: 0.654236
Train Epoch: 99	[1500/ 1761 (85%)]	Loss: 0.627425
Train Epoch: 99	[1600/ 1761 (91%)]	Loss: 0.415438
Train Epoch: 99	[1700/ 1761 (97%)]	Loss: 0.625114

(b)

Şekil 47. (a) MUD veri seti için ilk iterasyon ve alt küme iterasyonları (b) MUD veri seti için son ana iterasyon ve alt küme iterasyonları

Şekil 48’de, ilgili grafiklerde belirtilen adım sayıları, yukarıda belirtilen durumlardan kaynaklanmaktadır. Her iki modelde de hata oranlarının eğitim sonunda 0’a yakınsamamasının nedeni, aşırı öğrenmeyi engellemek amacıyla kullanılan seyreltme tekniğidir. Bu durum, seyreltme tekniğinin bilinçli olarak uygulanmasından kaynaklanmaktadır.



(a)

(b)

Şekil 48. (a) MNIST veri seti için hata eğrisi-iterasyon grafiği (b) MUD veri seti için hata eğrisi-iterasyon grafiği

2.5. Kompleks Veriler İçeren XOR Probleminin Çözümü

Gerçekleştirilen doktora çalışmasının ana hedefinin işaret dilinin tanınmasına yönelik akıllı bir sistemin geliştirilmesi olduğu daha önce ifade edilmiştir. Bu hedefe ulaşmak için sistemde kullanılacak modelin geçerliliği, çeşitli veri setleri üzerinde test edilmelidir. Çalışmada bu veri setleri MNIST ve MUD olarak belirlenmişti. Ancak veri setleriyle çalışırken, “CVDNN Mimarisi” bölümünde ifade edilen çeşitli faktörler nedeniyle karmaşık sayılarla çalışmanın etkisinin yanlış yorumlanabileceği açıktır. CVDNN, uzun süreli bir deneysel çalışmanın sonucunda nihai halini almıştır. Bu bağlamda, başlangıçta ağ tasarımı nedeniyle başarının düşük çıkma olasılığı göz önünde bulundurulmuştur. Bu nedenle, veri setleri ile çalışmaya geçilmeden önce daha basit bir CVDNN ile temel bir problemin çözümüne odaklanmak gerekmektedir. Bu doğrultuda, ilk çalışma giriş ve çıkış verileri karmaşık sayılar olacak şekilde XOR probleminin çözümü üzerine gerçekleştirilmiştir.

XOR problemin çözümünde kullanılan CVDNN’in ilk prototipi, gerçek değerli aktivasyon fonksiyonu olarak sigmoid kullanmaktadır. XOR probleminin çözümünde başarılı olan bu fonksiyon, veri setleri üzerinde iyi sonuçlar vermemiştir.

Tablo 6, XOR problemi için hem eğitim hem de test verilerini içermektedir. Toplamda 16 farklı durumu kapsayan bu problem için 12 durum eğitim verisi olarak kullanılmıştır.

Tablo 6. XOR problemi için giriş ve çıkışlar

Veri Türü	Giriş 1	Giriş 2	Çıkış
Eğitim	0	0	1
Eğitim	0	j	j
Eğitim	j	0	0
Eğitim	j	j	1+j
Eğitim	j	1	j
Eğitim	1	1	1+j
Eğitim	1+j	j	j
Eğitim	1+j	1+j	1
Eğitim	0	1	j
Eğitim	0	1+j	0
Eğitim	j	1+j	0
Eğitim	1	0	0
Test	1	j	j
Test	1	1+j	0
Test	1+j	0	0
Test	1+j	1	j

Şekil 49, eğitim ve test verileri için elde edilen sonuçları göstermektedir. İlgili şekilde virgülle ayrılan ilk kısımdaki veri giriş 1'e, ikinci kısımdaki veri ise giriş 2'ye karşılık gelmektedir. Çıkış verisi ise, karmaşık sayı olarak ilgili şeklin sağ tarafında yer almaktadır. Tablo 6'daki çıkış verileri ile modelden elde edilen çıkış verilerinin eşleşmesi, XOR probleminin prototip CVDNN ile tamamen çözüldüğünü ortaya koymaktadır. Bu çalışma, veri setleri üzerinde çalışmaya geçiş aşamasında önemli bir adım olmuştur.

```
Test Predictions:
tensor([0.+0.j, 0.+0.j]) ----> [0.9911 + 0.0073.j]
tensor([0.+0.j, 0.+1.j]) ----> [0.0046 + 0.9920.j]
tensor([0.+1.j, 0.+0.j]) ----> [0.0081 + 0.0035.j]
tensor([0.+1.j, 0.+1.j]) ----> [0.9894 + 0.9888.j]
tensor([0.+1.j, 1.+0.j]) ----> [0.0068 + 0.9997.j]
tensor([1.+0.j, 1.+0.j]) ----> [0.9883 + 0.9911.j]
tensor([1.+1.j, 0.+1.j]) ----> [0.0082 + 0.9983.j]
tensor([1.+1.j, 1.+1.j]) ----> [0.9917 + 0.0134.j]
tensor([0.+0.j, 1.+0.j]) ----> [0.0007 + 0.9995.j]
tensor([0.+0.j, 1.+1.j]) ----> [0.0112 + 0.0001.j]
tensor([0.+1.j, 1.+1.j]) ----> [0.0023 + 0.0021.j]
tensor([1.+0.j, 0.+0.j]) ----> [0.0012 + 0.0060.j]
tensor([1.+0.j, 0.+1.j]) ----> [0.0272 + 0.9995.j]
tensor([1.+0.j, 1.+1.j]) ----> [0.0089 + 0.0005.j]
tensor([1.+1.j, 0.+0.j]) ----> [0.0078 + 0.0001.j]
tensor([1.+1.j, 1.+0.j]) ----> [0.0006 + 0.9776.j]
```

Şekil 49. Karmaşık sayılar kullanan XOR probleminin çözümü

2.6. Performans Analiz Metrikleri

Bu bölümde, MNIST ve MUD veri setleri üzerinde gerçekleştirilen çalışmalardan elde edilen sonuçların analizinde kullanılan metrikler tanıtılmaktadır. Bu metriklerin tamamını bünyesinde barındıran karmaşıklık matrisi (confusion matrix), bir sınıflandırma modelinin performansını değerlendirmede kullanılan önemli bir araçtır. Bu matris, modelin tahminlerini gerçek etiketlerle karşılaştırarak tanıma (recognition), kesinlik (precision), hatırlama (recall) ve F1 skoru gibi performans metriklerini hesaplamak için temel bir yapı sağlar. Karmaşıklık matrisinin temel bileşenleri maddeler halinde aşağıda ifade edilmiştir.

- Doğru Pozitif (TP): Karmaşıklık matrisinde doğru bir şekilde pozitif olarak sınıflandırılmış örneklerin sayısı.
- Doğru Negatif (TN): Karmaşıklık matrisinde doğru bir şekilde negatif olarak sınıflandırılmış örneklerin sayısı.

- Yanlış Pozitif (FP): Karmaşıklık matrisinde yanlış bir şekilde pozitif olarak sınıflandırılmış negatif örneklerin sayısı.
- Yanlış Negatif (FN): Karmaşıklık matrisinde yanlış bir şekilde negatif olarak sınıflandırılmış pozitif örneklerin sayısı.

Karmaşıklık matrisi genellikle Tablo 7’de belirtildiği üzere ifade edilir.

Tablo 7. Karmaşıklık matrisinin genel temsili

	Gerçek Pozitif	Gerçek Negatif
Tahmin Pozitif	TP	FP
Tahmin Negatif	FN	TN

Bu matris, her sınıf için TP, TN, FP ve FN değerlerini içeren bir tablo oluşturur. Bu bilgiler, sınıflandırma modelinin çeşitli performans metriklerini hesaplamak için kullanılır.

İlgili metriklere ait değerler, tanıma oranı için bağıntı (51), kesinlik için bağıntı (52), hatırlama için bağıntı (53) ve F1 skoru için bağıntı (54) kullanılarak hesaplanmaktadır.

$$\text{Tanıma Oranı} = \frac{\text{Doğru Sınıflandırılmış Örnek Sayısı}}{\text{Toplam Örnek Sayısı}} \quad (51)$$

$$\text{Kesinlik} = \frac{\text{Doğru Pozitif}}{\text{Doğru Pozitif} + \text{Yanlış Pozitif}} \quad (52)$$

$$\text{Hatırlama} = \frac{\text{Doğru Pozitif}}{\text{Doğru Pozitif} + \text{Yanlış Negatif}} \quad (53)$$

$$\text{F1 Score} = 2 \times \frac{(\text{Kesinlik} \times \text{Hatırlama})}{(\text{Kesinlik} + \text{Hatırlama})} \quad (54)$$

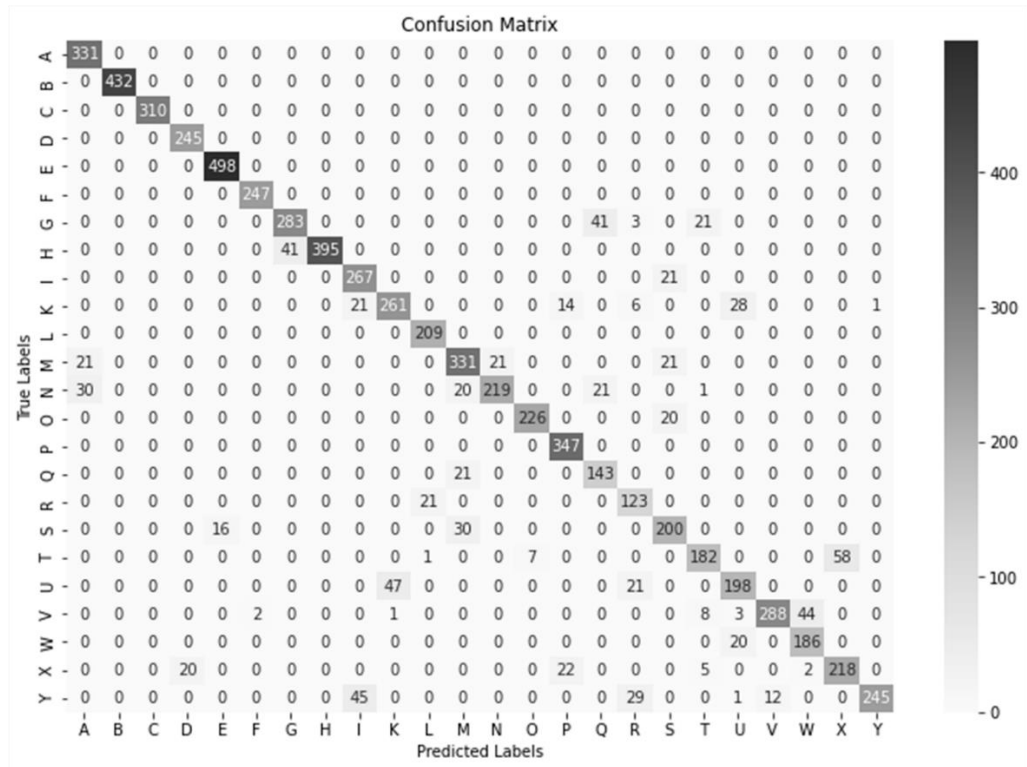
İlgili bağıntılarda belirlenen performans metriklerine göre, önerilen modelin MNIST ve MUD veri setlerinde elde ettiği değerler kendi alt başlıklarında sırasıyla incelenmektedir.

Gerçekleştirilen birçok akademik çalışmada, bağıntı (51)'deki tanıma yerine doğruluk olarak ifade edilen farklı bir bağıntı kullanılmasına rağmen, doktora çalışmaları kapsamında başarıyı daha net ve gerçekçi bir yaklaşımla göstermek amacıyla ilgili bağıntı tercih edilmiştir.

2.7. Önerilen Modelin MNIST Veri Seti Üzerindeki Performansının Değerlendirilmesi

MNIST veri setinden elde edilen karmaşık değerli verilerden oluşan eğitim ve test öznitelik vektörleri, geliştirilen CVDNN'e giriş olarak verilmiştir. Model, MNIST test verilerinde %89,01 tanıma oranı elde etmiştir. Şekil 50, karmaşık değerli veriler kullanılarak MNIST test veri setinde yapılan çalışmanın karmaşıklık matrisini göstermektedir. Ortalama kesinlik değeri 0.8793, ortalama hatırlama değeri 0.8856 ve ortalama F1 skoru 0.8791 olarak hesaplanmıştır; bu hesaplamalar sırasıyla bağıntı (52), bağıntı (53) ve bağıntı (54) kullanılarak yapılmıştır.

Tablo 8, MNIST test verilerindeki her bir sınıf (yirmi dört sınıf) için kesinlik, hatırlama ve F1 skorlarını belirtmektedir.



Şekil 50. Önerilen modelin MNIST test veri seti için oluşturduğu karmaşıklık matrisi

Tablo 8. Önerilen modelin MNIST test veri setinin her bir sınıfı için ürettiği performans metrik değerleri

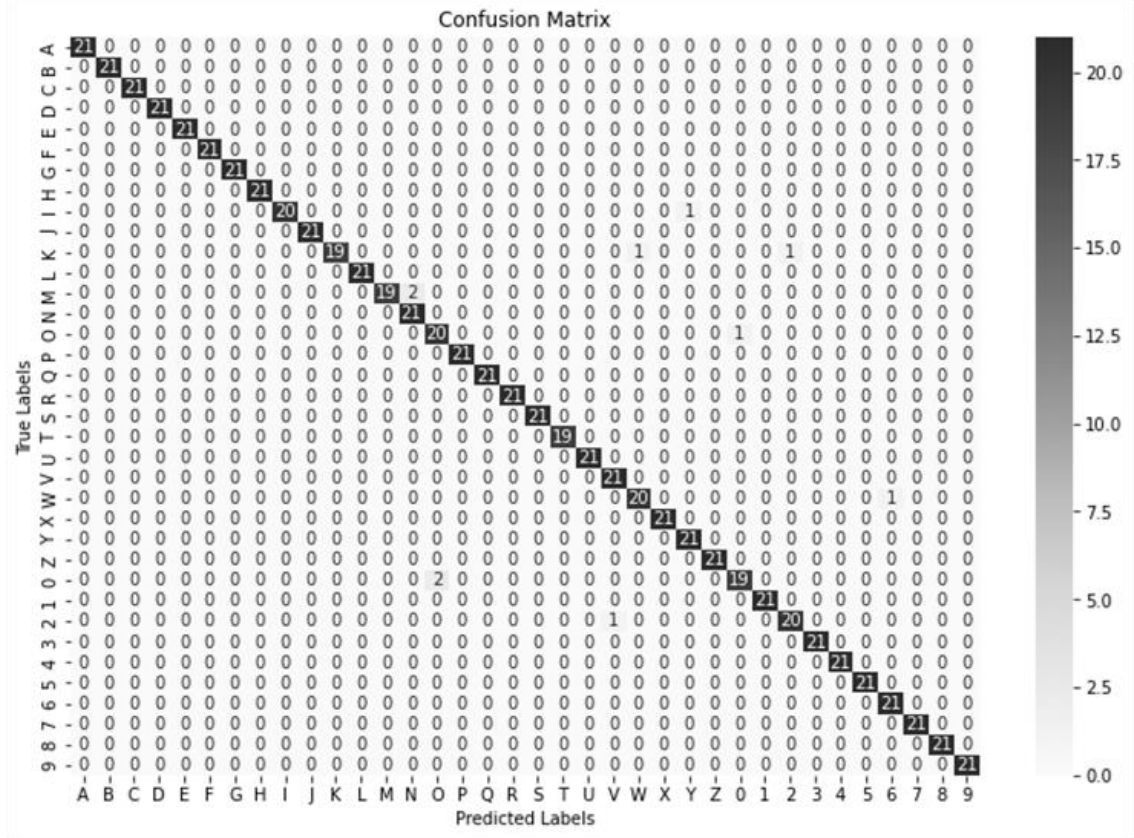
Karakter Sınıfları	Kesinlik	Hatırlama	F1 Score
A	0.8665	1.0	0.9285
B	1.0	1.0	1.0
C	1.0	1.0	1.0
D	0.9245	1.0	0.9608
E	0.9689	1.0	0.9842
F	0.992	1.0	0.996
G	0.8735	0.8132	0.8423
H	1.0	0.906	0.9507
I	0.8018	0.9271	0.8599
K	0.8447	0.7885	0.8156
L	0.9048	1.0	0.95
M	0.8234	0.8401	0.8317
N	0.9125	0.7526	0.8249
O	0.97	0.9187	0.9436
P	0.906	1.0	0.9507
Q	0.6976	0.872	0.7751
R	0.6758	0.8542	0.7546
S	0.7634	0.813	0.7874
T	0.8387	0.7339	0.7828
U	0.792	0.7444	0.7674
V	0.96	0.8324	0.8916
W	0.8017	0.9029	0.8493
X	0.7899	0.8165	0.8029
Y	0.9959	0.738	0.8478
Ortalama	0.8793	0.8856	0.8791

2.8. Önerilen Modelin MUD Veri Seti Üzerindeki Performansının Değerlendirilmesi

Önerilen model, MUD veri seti üzerinde Holdout ve LOSO olmak üzere iki farklı doğrulama tekniği kullanılarak değerlendirilmiştir. İlk olarak Holdout tekniğine göre elde edilen sonuçlar, ardından LOSO tekniği kullanılarak elde edilen sonuçlar sunulmaktadır.

Holdout tekniği kullanılarak gerçekleştirilen çalışmada, veri seti rastgele %70 (1761 görüntü) eğitim ve %30 (754 görüntü) test olarak bölünmüştür. Literatürdeki birçok çalışmada eğitim ve test verisi için %80-%20 ayırımının kullanıldığı düşünüldüğünde, önerilen sistemin elde ettiği başarı ayrı bir anlam taşımaktadır. Önerilen model MUD veri

setinde %98.67 tanıma oranı elde etmiştir. Şekil 51, MUD veri setinde Holdout tekniğine göre oluşturulan karmaşıklık matrisini gösterir. Tablo 9, MUD test verilerindeki her bir sınıf (otuz altı sınıf) için Holdout tekniğine göre kesinlik, hatırlama ve F1 skorlarını listeler.



Şekil 51. Önerilen modelin Holdout tekniğine göre MUD test veri seti için oluşturduğu karmaşıklık matrisi

LOSO tekniğini uygulayabilmek için MUD veri seti, el işaretlerinin kişilere göre ayrıldığı beş klasöre dönüştürülür. Bu klasörler "Denek1", "Denek2", "Denek3", "Denek4" ve "Denek5" olarak adlandırılmış olup, sırasıyla 900, 895, 180, 180 ve 360 görüntü içermektedir. Çıkış sınıfı ise dosya adında tanımlanmıştır. Önerilen model, LOSO tekniğini kullanarak MUD veri setinde ortalama %81.22 tanıma oranı elde etmektedir. Tablo 10, ilgili teknik kullanılarak tüm denekler için hesaplanan ortalama değerleri, görüntü sayısını, doğru tahmin sayısını, ortalama tanıma oranını, ortalama kesinlik, ortalama geri çağırma ve her bir denek için ortalama F1 puanını listelemektedir. "Denek1", "Denek2", "Denek3", "Denek4" ve "Denek5" için üretilen karışıklık matrisleri sırasıyla Şekil 52, Şekil 53, Şekil 54, Şekil 55 ve Şekil 56'de gösterilmektedir. Benzer şekilde "Denek1", "Denek2", "Denek3", "Denek4"

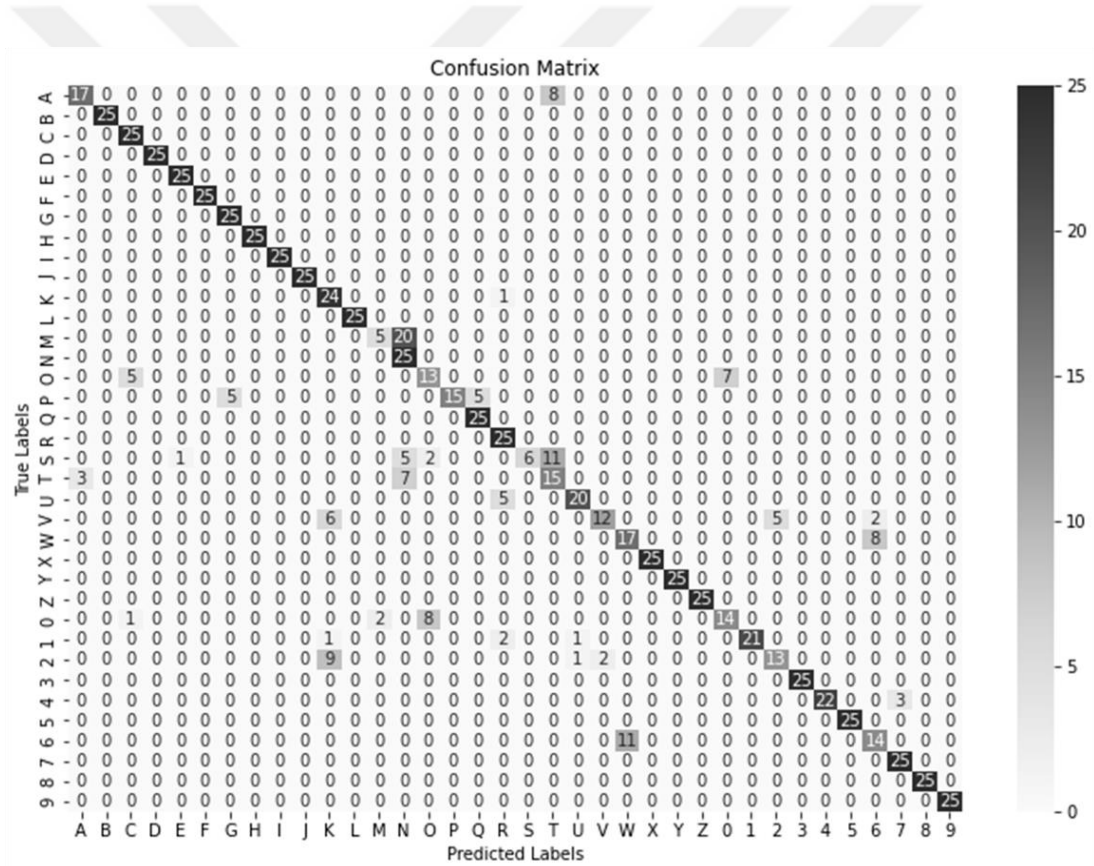
ve "Denek5" için kesinlik, hatırlama ve F1 skorları sırasıyla olmak üzere Tablo 11, Tablo 12, Tablo 13, Tablo 14 ve Tablo 15’de listelenmektedir.

Tablo 9. Holdout tekniğine göre MUD test verisindeki her bir sınıf için önerilen model tarafından üretilen performans metriği değerleri

Karakter Sınıfları	Kesinlik	Hatırlama	F1 Skoru
A	1.0	1.0	1.0
B	1.0	1.0	1.0
C	1.0	1.0	1.0
D	1.0	1.0	1.0
E	1.0	1.0	1.0
F	1.0	1.0	1.0
G	1.0	1.0	1.0
H	1.0	1.0	1.0
I	1.0	0.9524	0.9756
J	1.0	1.0	1.0
K	1.0	0.9048	0.95
L	1.0	1.0	1.0
M	1.0	0.9048	0.95
N	0.913	1.0	0.9545
O	0.9091	0.9524	0.9302
P	1.0	1.0	1.0
Q	1.0	1.0	1.0
R	1.0	1.0	1.0
S	1.0	1.0	1.0
T	1.0	1.0	1.0
U	1.0	1.0	1.0
V	0.9545	1.0	0.9767
W	0.9524	0.9524	0.9524
X	1.0	1.0	1.0
Y	0.9545	1.0	0.9767
Z	1.0	1.0	1.0
0	0.95	0.9048	0.9268
1	1.0	1.0	1.0
2	0.9524	0.9524	0.9524
3	1.0	1.0	1.0
4	1.0	1.0	1.0
5	1.0	1.0	1.0
6	0.9545	1.0	0.9767
7	1.0	1.0	1.0
8	1.0	1.0	1.0
9	1.0	1.0	1.0
Ortalama	0.9872	0.9868	0.9867

Tablo 10. Önerilen model tarafından LOSO tekniği kullanılarak MUD veri setindeki tüm denekler için hesaplanan ortalama değerler ile birlikte, her bir denek için görüntü sayısı, doğru tahmin sayısı, ortalama tanıma oranı, ortalama kesinlik, ortalama geri çağırma ve ortalama F1 puanı

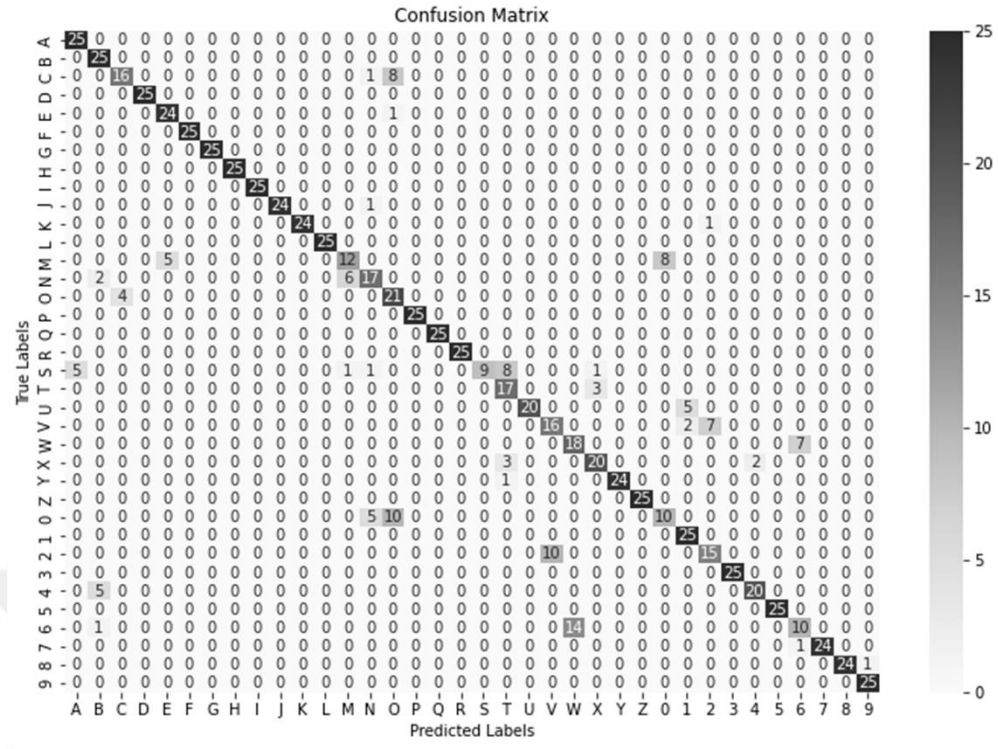
Denek	Görüntü Sayısı	Doğru Tahmin Sayısı	Kesinlik	Hatırlama	F1 Skoru	Tanıma Oranı
Denek 1	900	753	0.8622	0.8367	0.8290	0.8367
Denek 2	895	765	0.8630	0.8547	0.8498	0.8547
Denek 3	180	151	0.8211	0.8389	0.8153	0.8389
Denek 4	180	162	0.9295	0.9000	0.8915	0.9000
Denek 5	360	227	0.6551	0.6306	0.6265	0.6306
Ortalama	-	-	0.8262	0.8122	0.8024	0.8122



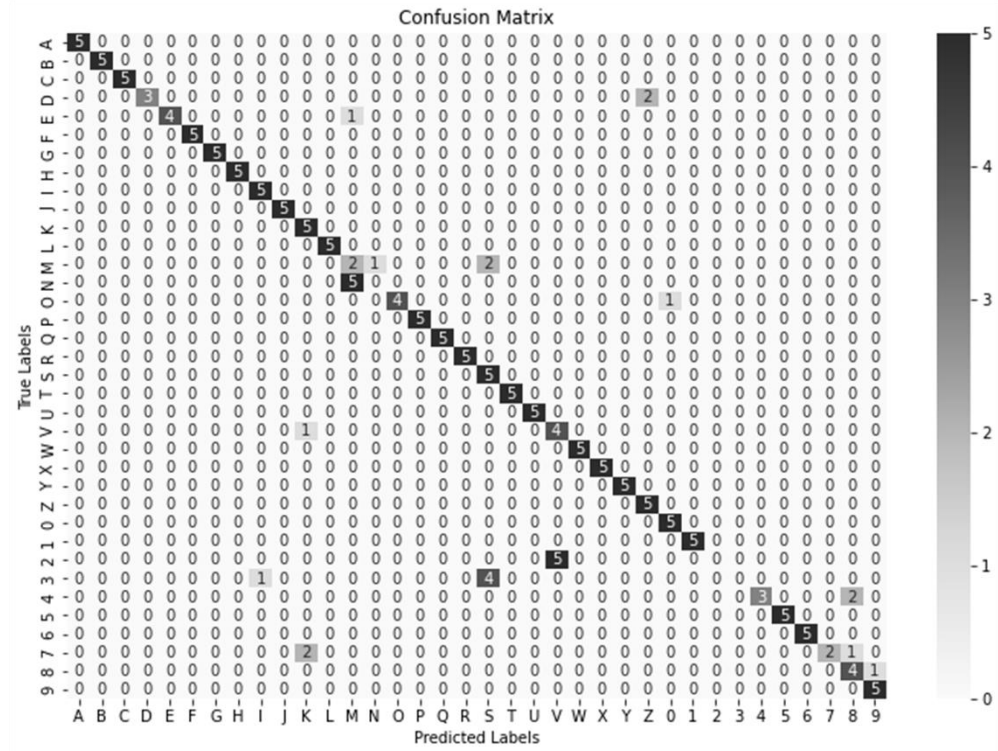
Şekil 52. Önerilen modelin LOSO tekniğine göre MUD veri setinde Denek 1 için oluşturduğu karmaşıklık matrisi

Tablo 11. LOSO tekniği kullanılarak MUD veri setindeki Denek 1'e göre her bir sınıf için önerilen model tarafından üretilen performans metrik değerleri

Karakter Sınıfları	Kesinlik	Hatırlama	F1 Skoru
A	0.85	0,68	0.7556
B	1.0	1.0	1.0
C	0.8065	1.0	0.929
D	1.0	1.0	1.0
E	0.9615	1.0	0.9804
F	1.0	1.0	1.0
G	0.8333	1.0	0.9091
H	1.0	1.0	1.0
I	1.0	1.0	1.0
J	1.0	1.0	1.0
K	0.6	0.96	0.7385
L	1.0	1.0	1.0
M	0.7143	0.2	0.3125
N	0.4386	1.0	0.6098
O	0.5652	0.52	0.5417
P	1.0	0.6	0.75
Q	0.833	1.0	0.9091
R	0.7576	1.0	0.8621
S	1.0	0.24	0.3871
T	0.4412	0.6	0.5085
U	0.9091	0.8	0.8511
V	0.8571	0.48	0.6154
W	0.6071	0.68	0.6415
X	1.0	1.0	1.0
Y	1.0	1.0	1.0
Z	1.0	1.0	1.0
0	0.6667	0.56	0.6087
1	1.0	0.84	0.913
2	0.7222	0.52	0.6047
3	1.0	1.0	1.0
4	1.0	0.88	0.9362
5	1.0	1.0	1.0
6	0.5833	0.56	0.5714
7	0.8929	1.0	0.9434
8	1.0	1.0	1.0
9	1.0	1.0	1.0
Ortalama	0.8622	0.8367	0.829



Şekil 53. Önerilen modelin LOSO tekniğine göre MUD veri setinde Denek 2 için oluşturduğu karmaşıklık matrisi



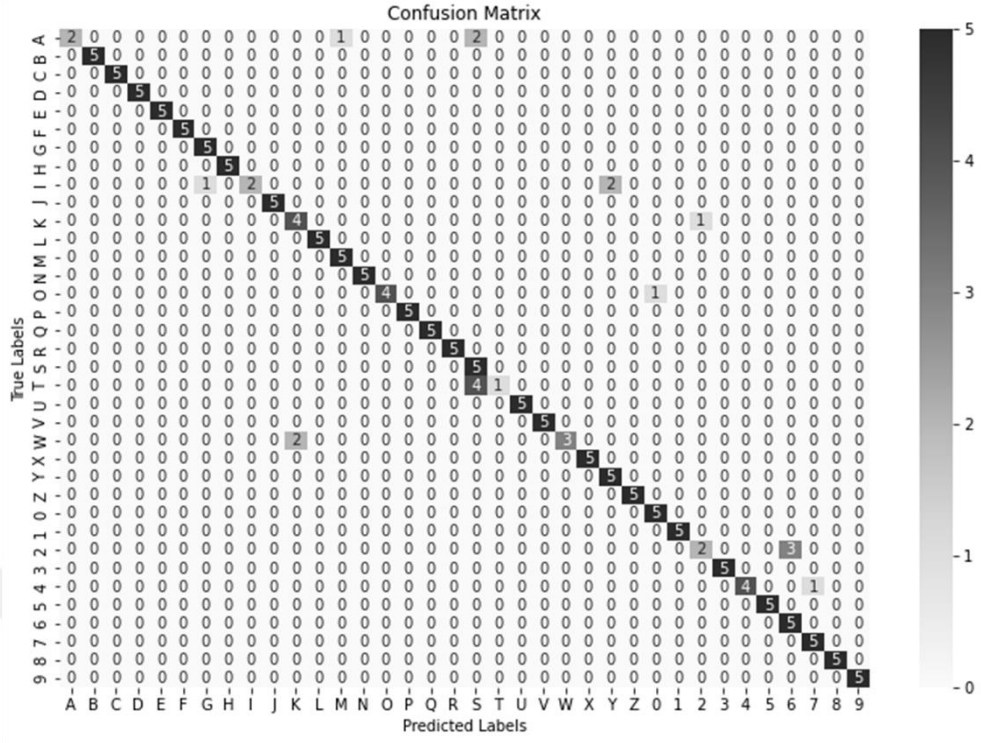
Şekil 54. Önerilen modelin LOSO tekniğine göre MUD veri setinde Denek 3 için oluşturduğu karmaşıklık matrisi

Tablo 12. LOSO tekniği kullanılarak MUD veri setindeki Denek 2'ye göre her bir sınıf için önerilen yöntem tarafından üretilen performans metrik değerleri

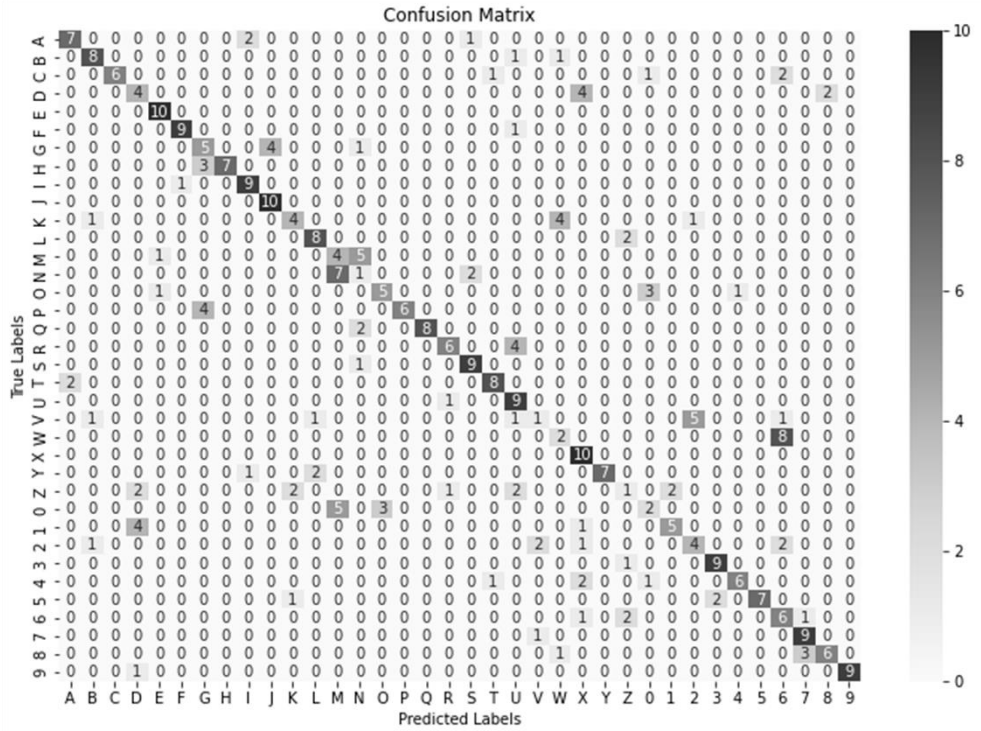
Karakter Sınıfları	Kesinlik	Hatırlama	F1 Skoru
A	0.8533	1.0	0.9091
B	0.7576	1.0	0.8621
C	0.8	0.64	0.7111
D	1.0	1.0	1.0
E	0.8276	0.96	0.8889
F	1.0	1.0	1.0
G	1.0	1.0	1.0
H	1.0	1.0	1.0
I	1.0	1.0	1.0
J	1.0	0.96	0.9796
K	1.0	0.96	0.9796
L	1.0	1.0	1.0
M	0.6316	0.48	0.5455
N	0.68	0.68	0.68
O	0.525	0.84	0.6462
P	1.0	1.0	1.0
Q	1.0	1.0	1.0
R	1.0	1.0	1.0
S	1.0	0.36	0.5294
T	0.5862	0.85	0.6939
U	1.0	0.8	0.8889
V	0.6154	0.64	0.6275
W	0.5625	0.72	0.6316
X	0.8333	0.8	0.8163
Y	1.0	0.96	0.9796
Z	1.0	1.0	1.0
0	0.5556	0.4	0.4651
1	0.7812	1.0	0.8772
2	0.6522	0.6	0.625
3	1.0	1.0	1.0
4	0.9091	0.8	0.8511
5	1.0	1.0	1.0
6	0.5556	0.4	0.4651
7	1.0	0.96	0.9796
8	1.0	0.96	0.9796
9	0.9615	1.0	0.9804
Ortalama	0.863	0.8547	0.8498

Tablo 13. LOSO tekniği kullanılarak MUD veri setindeki Denek 3’e göre her bir sınıf için önerilen model tarafından üretilen performans metrik değerleri

Karakter Sınıfları	Kesinlik	Hatırlama	F1 Skoru
A	1.0	1.0	1.0
B	1.0	1.0	1.0
C	1.0	1.0	1.0
D	1.0	0.6	0.75
E	1.0	0.8	0.8889
F	1.0	1.0	1.0
G	1.0	1.0	1.0
H	1.0	1.0	1.0
I	0.8333	1.0	0.9091
J	1.0	1.0	1.0
K	0.625	1.0	0.7692
L	1.0	1.0	1.0
M	0.25	0.4	0.3077
N	0.0	0.0	0.0
O	1.0	0.8	0.8889
P	1.0	1.0	1.0
Q	1.0	1.0	1.0
R	1.0	1.0	1.0
S	0.4545	1.0	0.625
T	1.0	1.0	1.0
U	1.0	1.0	1.0
V	0.4444	0.8	0.5714
W	1.0	1.0	1.0
X	1.0	1.0	1.0
Y	1.0	1.0	1.0
Z	0.7143	1.0	0.8333
0	0.8333	1.0	0.9091
1	1.0	1.0	1.0
2	0.0	0.0	0.0
3	0.0	0.0	0.0
4	1.0	0.6	0.75
5	1.0	1.0	1.0
6	1.0	1.0	1.0
7	1.0	0.4	0.5714
8	0.5714	0.8	0.6667
9	0.8333	1.0	0.9091
Ortalama	0.8211	0.8389	0.8153



Şekil 55. Önerilen modelin LOSO tekniğine göre MUD veri setinde Denek 4 için oluşturduğu karmaşıklık matrisi



Şekil 56. Önerilen modelin LOSO tekniğine göre MUD veri setinde Denek 5 için oluşturduğu karmaşıklık matrisi

Tablo 14. LOSO tekniđi kullanılarak MUD veri setindeki Denek 4'e gre her bir sınıf iin nerilen model tarafından retilen performans metrik deđerleri

Karakter Sınıfları	Kesinlik	Hatırlama	F1 Skoru
A	1.0	0.4	0.5714
B	1.0	1.0	1.0
C	1.0	1.0	1.0
D	1.0	1.0	1.0
E	1.0	1.0	1.0
F	1.0	1.0	1.0
G	0.8333	1.0	0.9091
H	1.0	1.0	1.0
I	1.0	0.4	0.5714
J	1.0	1.0	1.0
K	0.6667	0.8	0.7273
L	1.0	1.0	1.0
M	0.8333	1.0	0.9091
N	1.0	1.0	1.0
O	1.0	0.8	0.8889
P	1.0	1.0	1.0
Q	1.0	1.0	1.0
R	1.0	1.0	1.0
S	0.4545	1.0	0.625
T	1.0	0.2	0.3333
U	1.0	1.0	1.0
V	1.0	1.0	1.0
W	1.0	0.6	0.75
X	1.0	1.0	1.0
Y	0.7143	1.0	0.8333
Z	1.0	1.0	1.0
0	0.8333	1.0	0.9091
1	1.0	1.0	1.0
2	0.6667	0.4	0.5
3	1.0	1.0	1.0
4	1.0	0.8	0.8889
5	1.0	1.0	1.0
6	0.625	1.0	0.7692
7	0.8333	1.0	0.9091
8	1.0	1.0	1.0
9	1.0	1.0	1.0
Ortalama	0.9295	0.9	0.8915

Tablo 15. LOSO tekniđi kullanılarak MUD veri setindeki Denek 5’e gre her bir sınıf iin nerilen model tarafından retilen performans metrik deđerleri

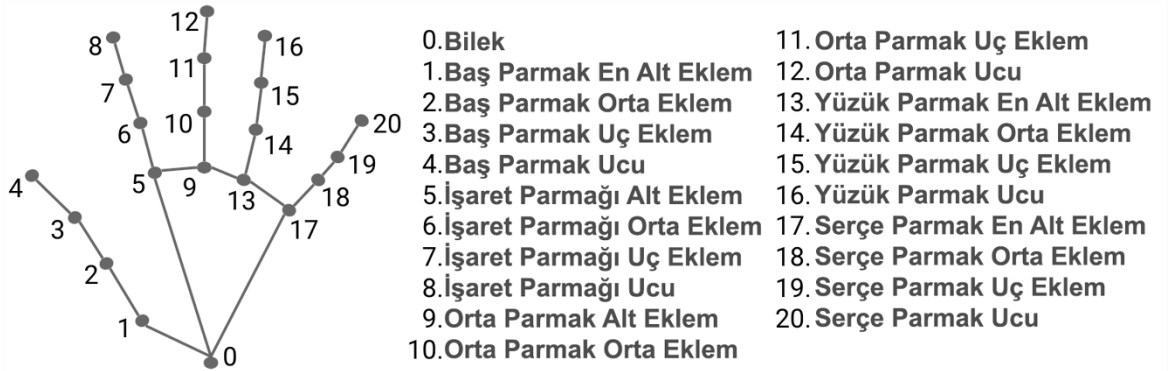
Karakter Sınıfları	Kesinlik	Hatırlama	F1 Skoru
A	0.7778	0.7	0.7368
B	0.7273	0.8	0.7619
C	1.0	0.6	0.75
D	0.3636	0.4	0.381
E	0.8333	1.0	0.9091
F	0.9	0.9	0.9
G	0.4167	0.5	0.4545
H	1.0	0.7	0.8235
I	0.75	0.9	0.8182
J	0.7143	1.0	0.8333
K	0.5714	0.4	0.4706
L	0.7273	0.8	0.7619
M	0.25	0.4	0.3077
N	0.1	0.1	0.1
O	0.625	0.5	0.5556
P	1.0	0.6	0.75
Q	1.0	0.8	0.8889
R	0.75	0.6	0.6667
S	0.75	0.9	0.8182
T	0.8	0.8	0.8
U	0.5	0.9	0.6429
V	0.25	0.1	0.1429
W	0.25	0.2	0.2222
X	0.5263	1.0	0.6897
Y	1.0	0.7	0.8235
Z	0.1667	0.1	0.125
0	0.2857	0.2	0.2353
1	0.7143	0.5	0.5882
2	0.4	0.4	0.4
3	0.8182	0.9	0.8571
4	0.8571	0.6	0.7059
5	1.0	0.7	0.8235
6	0.3158	0.6	0.4138
7	0.6923	0.9	0.7826
8	0.75	0.6	0.6667
9	1.0	0.9	0.9474
Ortalama	0.6551	0.6306	0.6265

2.9. Önerilen Modelin Kullanıldığı Akıllı Sistemin Geliştirilmesi

Geliştirilen akıllı sistem, ASL'nin tanınması amacıyla oluşturduğumuz modeli kullanarak gerçek zamanlı işaret dili tanıma amacı taşımaktadır. Bu bağlamda ilgili sistem kullanıcıdan işaret dili görüntülerinin gerçek zamanlı olarak alınmasını sağlayacak bir dış cihaza ihtiyaç duymaktadır. Dış cihazın kaliteli olması, elde edilecek görüntüyü ve dolayısıyla modelin performansını doğrudan etkileyecektir. Dış cihaz olarak dizüstü bilgisayar üzerinde bütünleşik olarak sunulan standart bir web kamerası kullanılmıştır.

Geliştirilen gerçek zamanlı işaret dili tanıma sistemi için başlangıçta karşılaşılan iki temel zorluk, görüntüde elin tespit edilmesi gerekliliği ve arka plan farklılıklarına bağlı olarak özellik çıkarımı ve sınıflandırmada oluşabilecek hatalardır.

Web kamerasından alınan görüntü akışında elin tespit edilmesi amacıyla Mediapipe kütüphanesi kullanılmıştır (URL3, 2024). Mediapipe bir görüntüdeki yüz, poz, sol el ve sağ el bilgilerinin elde edilmesini sağlayan Google tarafından üretilmiş bir yapıdır. Geliştirilen sistem şu anda sadece el görüntüsü üzerinden çalıştığı için sol ve sağ el bilgilerinin elde edilmesi yeterli olacaktır. İlgili kütüphane, görüntü üzerindeki tespit edilen her bir el için 21 adet anahtar nokta tanımlamaktadır. Sağ el için tespit edilen anahtar noktalar ve açıklamaları Şekil 57'de gösterilmiştir. Sol el için de benzer şekilde olmak kaydıyla y eksenine göre simetri durumu mevcuttur.



Şekil 57. Sağ el için Mediapipe tarafından üretilen anahtar noktalar ve açıklamaları

Gerçekleştirilen tüm çalışmalarda görüntülere herhangi bir ön işleme uygulanmadığı daha önce belirtilmiştir. Bu nedenle, hem MNIST hem de MUD veri setleri üzerinde gerçekleştirilen çalışmalarda arka planla ilgili herhangi bir işlem yapılmaksızın modeller geliştirilmiştir. Başka bir deyişle, her iki veri seti için geliştirilen modeller eğitim sürecinde

görüntüleri arka planlarıyla birlikte analiz etmiştir. MNIST veri setinin düşük çözünürlüğe sahip olması, ayrıca görüntülerdeki arka planların farklı renk tonlarına ve nesne veya nesne parçalarına sahip olması nedeniyle gerçek zamanlı sistem için uygun görülmemiştir. Buna karşılık, MUD veri seti yüksek çözünürlüğe sahip olup tamamen siyah bir arka plan üzerinde el görüntüleri içermektedir. Bu nedenle, arka plana bağlı oluşabilecek hataları en aza indirmek amacıyla MUD veri seti, geliştirilen modelin tanıma sisteminde kullanılmıştır. Böylece sistem kullanılırken arka plan olarak siyah bir zemin oluşturulup, arka plan kaynaklı hataların en aza indirgenmesi hedeflenmiştir.

Arka plan sorunu kısmen çözüme ulaştırıldığına göre, sıradaki süreç Mediapipe'in sağ ve sol el için ürettiği anahtar noktaları kullanarak geliştirilecek sistem için elin uygun şekilde tespit edilmesidir. Zernike momentlerin birim disk üzerinde çalıştığı daha önce ifade edilmiştir. Birim disk, karesel bir görüntü üzerinde oluşturabilir. Bu nedenle, anahtar noktalar kullanılarak eli çerçeveleyen en küçük kare oluşturulmuştur. Eli çerçeveleyen karesel görüntü, öncelikle 400x400 piksele ölçeklenerek özellik çıkarımı aşamasına yönlendirilir. Bu ölçekleme işlemi, görüntünün MUD veri seti üzerinde geliştirilen modelle uyumlu olması amacıyla gerçekleştirilir. Eğer bu işlem yapılmazsa sistemden beklenen başarı elde edilemez. Özellik çıkarımından elde edilen öznitelik vektörü CVDNN'e yönlendirilerek çıkış sınıfı tespit edilir. Elde edilen sınıf ara yüzün sol üst köşesinde gösterilir. Ayrıca eli çerçeveleyerek elde edilen görüntü de 100x100 piksel boyutuna ölçeklenerek ara yüzün sağ üst köşesine yerleştirilmiştir.

Özellik çıkarımı süreci çözünürlük yüksek olduğundan gerçek zamanlı bir sistem için fazla zaman almaktadır. Bu sebeple her beş görüntüde (frame) bir işlem yapılmaktadır. Fakat buradaki ilk amaç önerilen yöntemin başarısını test etmektir. Bu sebeple de yüksek çözünürlüklü görüntüler üzerinden işlem yapılmaktadır.

Gerçek zamanlı sistem, eğitiminde kullanılmayan bir el üzerinden yani aslında LOSO tekniğine göre test edilmiştir. El işaretlerinin düzgün ve doğru bir şekilde yapılabilmesi, ayrıca arka planın ve ışığın optimum ayarlanması sonucunda geliştirilen sistem, tanıma oranı olarak %90 ile %95 arasında bir performans değeri elde etmektedir. Şekil 58 gerçek zamanlı sistemden bir kesiti göstermektedir.

Tablo 16 ise web kamerasından alınan görüntü akışında elin kare içine alınmasını sağlayan kod parçasığını içermektedir.



Şekil 58. Gerçek zamanlı sistemden bir kesit görüntüsü

Tablo 16. Web kamerasından alınan görüntü akışında elin kare içine alınmasını sağlayan kod parçasığı

Kod Parçasığı

```
import cv2
import mediapipe as mp
import numpy as np
from kostur import ozellikcikar
from progakis import sinifbelirle

mp_drawing = mp.solutions.drawing_utils
mp_drawing_styles = mp.solutions.drawing_styles
mpHands = mp.solutions.hands

cap = cv2.VideoCapture(0)
hands = mpHands.Hands()
margin = 20
frame=0

while cap.isOpened():
    frame+=1
    if frame%5==0:
        sinifimiz=""
```

```

with mpHands.Hands(static_image_mode=True, max_num_hands=2,
min_detection_confidence=0.5) as hands:
    success, image = cap.read()
    image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
    results = hands.process(image)
    if results.multi_hand_landmarks:
        for hand_landmarks in results.multi_hand_landmarks:
            h, w, _ = image.shape
            x_coords = []
            y_coords = []

            for lm in hand_landmarks.landmark:
                x, y = int(lm.x * w), int(lm.y * h)
                x_coords.append(x)
                y_coords.append(y)

            x_min, x_max = min(x_coords), max(x_coords)
            y_min, y_max = min(y_coords), max(y_coords)

            x_min_extended = max(x_min - margin, 0)
            y_min_extended = max(y_min - margin, 0)
            x_max_extended = min(x_max + margin, w)
            y_max_extended = min(y_max + margin, h)

            side_length = max(x_max_extended - x_min_extended, y_max_extended -
y_min_extended)

            x_min_square = max(x_min_extended - (side_length - (x_max_extended -
x_min_extended)) // 2, 0)
            y_min_square = max(y_min_extended - (side_length - (y_max_extended -
y_min_extended)) // 2, 0)
            x_max_square = min(x_min_square + side_length, w)
            y_max_square = min(y_min_square + side_length, h)

            mask = np.zeros(image.shape[:2], dtype=np.uint8)
            cv2.rectangle(mask, (x_min_square, y_min_square), (x_max_square,
y_max_square), 255, -1)

            hand_area = cv2.bitwise_and(image, image, mask=mask)

            hand_area_cropped = hand_area[y_min_square:y_max_square,
x_min_square:x_max_square]
            hand_area_resized = cv2.resize(hand_area_cropped, (400, 400))
            hand_area_view=cv2.resize(hand_area_cropped, (100, 100))
            dosyaismi="hand_area_cropped.png"
            cv2.imwrite(dosyaismi, cv2.flip(hand_area_resized, 1))

```

```
ozellikcikar(dosyaismi)
sinifimiz=sinifbelirle()

hand_area_height, hand_area_width = hand_area_view.shape[:2]
top_left_x = 0
top_left_y = 0

if hand_area_height > 0 and hand_area_width > 0:
    image[top_left_y:top_left_y + hand_area_height, top_left_x:top_left_x +
hand_area_width] = hand_area_view

    cv2.rectangle(image, (x_min_square, y_min_square), (x_max_square,
y_max_square), (0, 255, 0), 2)
    frame=0
    imageflip=cv2.flip(image, 1)
    cv2.putText(imageflip, f'{sinifimiz}', (0, 30), cv2.FONT_HERSHEY_SIMPLEX, 1,
(0, 255, 0), 2, cv2.LINE_AA)
    cv2.imshow('HandTracker',imageflip)
    if cv2.waitKey(5) & 0xFF == 27:
        break

cap.release()
cv2.destroyAllWindows()
```

3. BULGULAR VE TARTIŞMA

Bu bölüm, çalışmalar gerçekleştirilirken karşılaşılan zorlukların nasıl çözüldüğünü, farklı seçeneklerin olduğu durumlarda ilgili seçeneklerden elde edilen sonuçları, diğer çalışmalarla karşılaştırmaları ve önerilen yöntemin etkisinin kanıtlanmasını içeren çeşitli işlemlerden oluşmaktadır.

3.1. Model Mimarisi ve Akıllı Sistem Üzerindeki Ablasyon Çalışmaları

İlk olarak, özellik çıkarımı aşamasında moment derecesinin nasıl belirlendiği daha detaylı bir şekilde incelenecektir. Moment derecesinin belirlenmesi işlemi, öncelikli olarak 10. dereceden 30. dereceye kadar, 4'er adımlık bir ilerlemeyle (10, 14, 18, 22, 26 ve 30) elde edilen öznitelik vektörlerinin CVDNN'ye verilmesi ve sonuçların analiz edilmesiyle başlar. Elde edilen sonuçları incelediğinde, en iyi değerlerin benzerlik algoritmaları MSE ve SSIM'in sonuçlarına bağlı olarak 10. ve 14. moment derecelerinde elde edildiği tespit edilmiştir. Aynı işlemler, 10. dereceden 15. dereceye kadar (10, 12, 13, 14 ve 15) adım adım gerçekleştirilmiş ve en iyi sonucun 14. derecede olduğu belirlenmiştir. Gerçekleştirilen deneyler sonucunda, doğru moment derecesinin seçilmesinin sistemin performansını yaklaşık %8-10 oranında etkilediği gözlemlenmiştir.

Başlangıçta, arka planın performans üzerindeki etkisini netleştirebilmek amacıyla MNIST veri seti üzerinde çeşitli arka plan çıkarım teknikleri denenmiş, ancak arzu edilen başarıya ulaşılamamıştır. Bu noktada, veri setinin düşük çözünürlüğe sahip olması ve görüntülerin net olmaması önemli bir etkiye sahiptir. MNIST veri setinin başlangıçta seçilmesinin nedeni, birçok çalışmanın bu veri seti üzerinde gerçekleştirilmiş olması ve dolayısıyla karşılaştırma imkanı sunmasıdır. MNIST veri setinde elde edilen başarıyı takiben, MUD veri setinde sağlanan sonuçlar yöntemin etkinliğini kanıtlamış ve doğrudan akıllı sistem tasarımına geçilmiştir. Akıllı sistemin test aşamalarında, arka planın farklı olduğu durumlarda sonuçların olumsuz olduğu tespit edilmiştir. MUD veri setindeki tüm görüntülerin arka planının tamamen siyah olması ve akıllı sistemin de benzer bir ortam kurulduğunda yüksek başarı elde edilmesi, arka planın etkisinin ne denli önemli olduğunu göstermektedir. MUD veri setinde arka planın çıkarılması işleminin başarılı olacağı

rahatlıkla söylenebilir. Aynı şekilde, akıllı sistemde de arka planın kolayca çıkarılabileceği bir ortam hazırlanarak siyah arka plan bağımlılığından kurtulmak mümkündür.

CVDNN modülünde, katman sayısını belirlemek için öncelikle 4-5 katmanlı yapılar test edilmiş ve sonuçların benzer olduğu gözlemlenmiştir. CVDNN'in karmaşık girişlere sahip XOR problemini çözmek amacıyla üretilen ilk prototipinin gerçek değerli aktivasyon fonksiyonu sigmoid kullandığı daha önce ifade edilmiştir. Sigmoid dışında, sıkça kullanılan ReLU ve tanh gibi gerçek değerli aktivasyon fonksiyonları da mevcuttur. Çalışmalar kapsamında her üç fonksiyon da kullanılmış ve ReLU'nun en iyi seçenek olduğu tespit edilmiştir. tanh fonksiyonu da başarılı sonuçlar vermiş olmakla birlikte, ReLU'nun performansına ulaşamamıştır. ReLU fonksiyonun performansa etkisi, sigmoid'e kıyasla yaklaşık %9 ile %11 arasında değişmektedir. Seyreltme tekniği, aşırı öğrenmeyi önleyerek sistemin performansını artırmada kritik bir rol oynar. Seyreltme oranının belirlenmesi ve uygulanmasının detayları daha önce ifade edilmiştir. Seyreltme tekniğinin sistem performansına etkisi yaklaşık %6-8 arasında değişmektedir.

Önerilen yöntemin etkinliğini kanıtlamak amacıyla bir başka çalışma daha gerçekleştirilmiştir. Bu çalışmada, karmaşık verilerin gerçek değerli verilere kıyasla daha iyi sonuçlar verdiğini göstermek için, ilk çalışma gerçek verilerle yapılmıştır. Bu amaçla, karmaşık verilerin genlik $|z|$ bilgileri kullanılarak eğitim ve test öznitelik vektörleri oluşturulmuştur. Benzer şekilde, CVDNN'in yapısı aynı kalmak şartıyla sadece gerçek değerli verilerle çalışabilecek şekilde DNN'e dönüştürülmüştür. Sonuçta elde edilen DNN, MNIST test verilerinde %72.15 tanıma oranı elde etmiştir. Literatürde, benzer optimizasyon algoritmasını (momentumlu SGD) kullanan sistemlerin, gerçek değerli veriler üzerinde SGD'nin %66.02 ve SGDC'nin %59.80 tanıma oranı elde ettiği belirtilmektedir. Ayrıca, en yaygın kullanılan optimizasyon algoritmalarından biri olan Adam'ın, momentumlu SGD ile benzer başarı oranlarına sahip olduğu gözlemlenmiştir (Kingma & Ba, 2024). Sonuç olarak, karmaşık sayılarla çalışmanın, modelin yapısını korunarak gerçek sayılarla çalışabilecek şekilde yapılandırılmasına kıyasla yaklaşık %20 olumlu bir performans etkisi sağladığı gözlemlenmiştir.

3.2. Geliştirilen Modelin Diğer Çalışmalarla Karşılaştırılması

Bu bölümde, önerilen yaklaşımı içeren model literatürdeki diğer çalışmalarla karşılaştırılır. Öncelikle önerilen modelin MNIST veri setinde kullanılan diğer modellerle

karşılaştırılması ele alınmakta, ardından benzer bir karşılaştırma MUD veri seti için Holdout ve LOSO tekniklerine bağlı olarak ayrı ayrı yapılmaktadır.

Adil bir karşılaştırma sağlamak amacıyla, MNIST test veri setinin tamamı üzerinde yapılan çalışmalara odaklanılmaktadır. Tablo 17, önerilen model ile MNIST test veri setinin tamamında gerçekleştirilen çalışmaların karşılaştırılmasını ifade etmektedir. Ayrıca, tabloda listelenen çalışmaların hiçbirisi veri seti üzerinde herhangi bir ön işleme gerçekleştirmemiştir.

Tablo 17. Önerilen model ile diğer çalışmaların MNIST test veri setindeki tanıma oranları

Model/Metot	Tanım Oranı (%)
NBG (Li. vd, 2022) (Bhushan vd., 2022)	38.9
SGDC (Bhushan vd., 2022)	59.8
SGD (Li. vd, 2022)	66.02
KNN (Li. vd, 2022) / KNN (Bhushan vd., 2022)	78.17 / 80.46
XGBoost (Bhushan vd., 2022)	81.35
RFC (Li. vd, 2022) / RF (Bhushan vd., 2022)	81.61 / 84.43
LeNeT-5 (Bilgin & Mutludoğan, 2019)	82.19
SVM (Li. vd, 2022)	84.19
CapsNet (Bilgin & Mutludoğan, 2019)	88.93
Önerilen Model	89.01
CNN (Bhushan vd., 2022)	91.41

Önerilen modelin MUD veri seti için diğer çalışmalarla karşılaştırılması, Holdout ve LOSO tekniklerine göre sırasıyla Tablo 18 ve Tablo 19'da listelenmiştir. Tüm çalışmaların LOSO tekniğinde aynı veriler üzerinde gerçekleştirildiği göz önüne alındığında, karşılaştırmalar daha güvenilirdir.

Tablo 18. Holdout tekniğine göre önerilen model ve diğer çalışmaların MUD test veri setindeki tanıma oranları

Model/Metot	Tanım Oranı (%)
Inception-v3 (Barbhuiya vd., 2022)	93.50
Squeezenet (Barbhuiya vd., 2022)	95.23
Vgg19 (Barbhuiya vd., 2022)	96.68
AlexNet (Barbhuiya vd., 2022)	97.01
Vgg16 (Barbhuiya vd., 2022) / Vgg16 (Chevtchenko vd., 2018)	97.48 / 92.20
Chevtchenko (Chevtchenko vd., 2018)	98.05
Barbhuiya (Barbhuiya vd., 2022)	98.41
Önerilen Model	98.67

Tablo 19. LOSO tekniğine göre önerilen model ve diğer çalışmaların MUD test veri setindeki tanıma oranları

Model/Metot	Tanıma Oranı (%)
Vgg16 (Chevtchenko vd., 2018)	61.42
MobileNet (Chevtchenko vd., 2018) / MobileNet (Bhaumik vd., 2024)	62.53 / 73.25
Modified VGG16 + SVM (Barbhuiya vd., 2021)	70.00
Modified AlexNet+SVM (Barbhuiya vd., 2021)	70.00
MobileNetV2 (Bhaumik vd., 2024)	72.28
ResNet50 (Bhaumik vd., 2024)	74.42
CNN (Chevtchenko vd., 2018)	73.86 / 78.51 / 79.04
SpAtNet (Bhaumik vd., 2024)	80.44
Önerilen Model	81.22
Chevtchenko (Chevtchenko vd., 2018)	84.02

Geliştirilen modelin başarı durumu incelenirken, çıkış sınıflarında ne tür hatalar yapıldığına dair analizler de gerçekleştirilmiştir. Bu analizlerin sonuçları aslında karmaşıklık matrisi üzerinden de elde edilebilse de, çok fazla çıkış sınıfının olması sebebiyle hata yapılan çıkışların tespitini daha rahat sağlamak amacıyla yapılmıştır. Bu yönde yapılan çalışmalarda, doğru sınıf çıktısı ile tahmini sınıf çıktıları farklı olan öğeler listelenmiştir. Şekil 59, ASL veri setinin eğitim için %70, test için %30 olarak ayrıldığı (Holdout tekniği) durumda önerilen modelin test verisi üzerinden ürettiği çıkış sınıflarının hata analiz sonuçlarını göstermektedir. İlgili şekilde, boş kümeler ilgili sınıf için hata olmadığını gösterirken, diğer durumlarda o sınıf için yapılan yanlış tahminleri ve sayısını virgülle ayrılmış şekilde sınıf bazında göstermektedir.

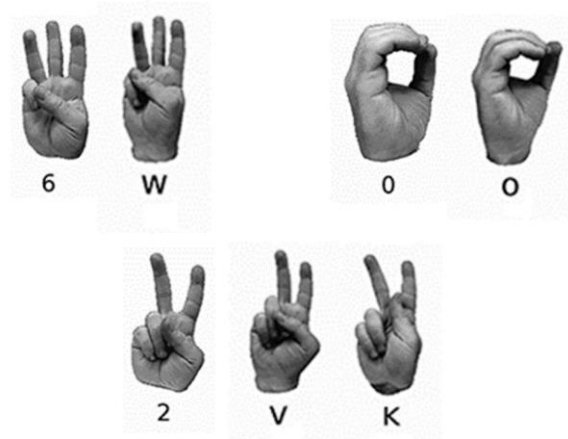
Şekil 60, önerilen modelin ilgili tekniğe göre hata yaptığı durumlarda, ayırt etmede en fazla zorluk yaşadığı sınıflara ait görüntüleri birlikte göstermektedir. İlgili şekilden görülebileceği üzere, önerilen model 6 ile W, 0 ile O ve 2 ile V sınıflarının bazı örneklerinde sorun yaşamaktadır. Bu belirtilen sınıflarla ilgili olarak, gerçek zamanlı denemelerde de sorun yaşanabileceği görülmüştür çünkü Şekil 60'dan anlaşılabileceği üzere, bu sınıflar birbirine oldukça benzemektedir. Ancak burada dikkat edilmesi gereken durum ilgili sınıfların birbirinden ayrılmasındaki çok ince detaylardır. Bu durum, gerçek zamanlı bir sistemde yapılan deneylerde, ilgili kişinin işaret dilini çok iyi bilmesinin önemli olduğunu ortaya koymaktadır. İlgili veri seti Holdout tekniğine göre %80-%20 olarak ayrıldığında, önerilen model %99.60 tanıma oranına ulaşmakta ve 503 görüntüde 501 doğru tahmin yapmaktadır.

```

A      Hata Analizi: {}
B      Hata Analizi: {}
C      Hata Analizi: {}
D      Hata Analizi: {}
E      Hata Analizi: {}
F      Hata Analizi: {}
G      Hata Analizi: {}
H      Hata Analizi: {}
I      Hata Analizi: {Y:1}
J      Hata Analizi: {}
K      Hata Analizi: {W:1,iki:1}
L      Hata Analizi: {}
M      Hata Analizi: {N:2}
N      Hata Analizi: {}
O      Hata Analizi: {sifir:1}
P      Hata Analizi: {}
Q      Hata Analizi: {}
R      Hata Analizi: {}
S      Hata Analizi: {}
T      Hata Analizi: {}
U      Hata Analizi: {}
V      Hata Analizi: {}
W      Hata Analizi: {alti:1}
X      Hata Analizi: {}
Y      Hata Analizi: {}
Z      Hata Analizi: {}
sifir  Hata Analizi: {0:2}
bir    Hata Analizi: {}
iki    Hata Analizi: {V:1}
uc     Hata Analizi: {}
dort   Hata Analizi: {}
bes    Hata Analizi: {}
alti   Hata Analizi: {}
yedi   Hata Analizi: {}
sekiz  Hata Analizi: {}
dokuz  Hata Analizi: {}

```

Şekil 59. ASL veri setinin %70 eğitim %30 test verisi olarak ayrıldığı durumda önerilen model tarafından elde edilen hata analiz sonuçları



Şekil 60. ASL veri setinde birbirine oldukça benzeyen çıkış sınıfları

Son olarak řunu ifade edebiliriz ki web kamerasından alınan görüntüde özellik çıkarımı aşamasında kullandığımız karmaşık Zernike moment tekniđi CNN gibi hazır sinir ađlarına kıyasla donanım bazlı kaynak tüketimini oldukça azaltmıştır. Bu sayede model çok daha düşük kaynak içeren sistemlerde de çalışabilmektedir.



4. SONUÇLAR

Bu çalışmada, Amerikan işaret dilini tanımaya yönelik bir araştırma gerçekleştirilmiştir. Bu çalışmada, özellik çıkarma aşamasında Zernike momentlerin büyüklükleri yerine karmaşık bileşenler ve faz bilgisinin kullanılması daha uygun bir yöntem olarak değerlendirilmiştir. Bu sayede, elde edilen öznitelik vektörünün görüntüyü daha iyi temsil edeceği ve dolayısıyla geliştirilen uygulamanın daha başarılı sonuçlar üreteceği öngörülmüştür.

Literatürde karmaşık Zernike momentler olarak adlandırılan bu teknikle, öncelikli olarak görüntünün yeniden oluşturulması amaçlanmıştır. Üretilen yeniden oluşturulmuş temsillerin, yüksek moment derecelerinde orijinal görüntüye oldukça benzediği tespit edilmiş ve bu durum, yöntemin özellik çıkarımındaki etkinliğini kanıtlamıştır.

Elde edilen karmaşık öznitelik vektörünü işleyebilecek karmaşık değerli bir sinir ağının geliştirilmesi gerekmektedir. Bu bağlamda, katmanlar üzerinde karmaşık verilere ayrı ayrı aktivasyon fonksiyonu uygulanır ve elde edilen veriler üzerinden tekrar karmaşık veri üretilerek ağın katmanları boyunca karmaşık verilerle çalışması sağlanır.

Görüntü analizi gibi karmaşık süreçlere geçilmeden önce, karmaşık değerli sinir ağının oluşturulması ve işlevselliğini test etmek amacıyla basit bir problemin çözümüne odaklanılmıştır. Bu bağlamda, karmaşık giriş değerlerine sahip XOR probleminin çözümü tasarlanan sinir ağı üzerinden gerçekleştirilmiştir. 16 farklı durumu için problem için 12 durum eğitim amacıyla kullanılmış ve toplam 16 durum test olarak verilmiştir. Geliştirilen ağın ilk prototipi tüm durumları doğru olarak tespit etmiştir.

XOR problemiindeki başarıdan sonra, asıl hedef olan işaret dili tanıma sisteminin oluşturulmasına geçilmiştir. İki farklı veri seti olan MNIST ve MUD üzerinde gerçekleştirilen işlemler doğrultusunda, özellik çıkarım sürecinde çeşitli düzenlemeler yapılmıştır. Bu düzenlemelerden en önemlisi, moment derecesinin ayarlanmasıdır. Ancak, kapsamlı bir analiz ve geliştirme süreci, birçok düzenleme ile birlikte karmaşık değerli sinir ağı modülünde gerçekleştirilmiştir. Bu düzenlemelerden bazıları; katman sayısının ve katmanlardaki nöron sayısının belirlenmesi, seyreltme oranının ve tekniğinin tespit edilmesi, öznitelik vektörü değerlerinin normalizasyonu, karmaşık aktivasyon fonksiyonun netleştirilmesi, öğrenme adımı sırasında verilerin hangi oranda ayrılarak kullanılacağına kararlaştırılması, çıkış katmanının yorumlanmasında ve hatanın geri yayılımında

kullanılacak fonksiyonların belirlenmesi, optimizasyon algoritmasının saptanması ve parametrelerinin ayarlanması olarak ifade edilebilir. Bu düzenlemeler gerçekleştirilirken karşılaşılan en önemli problem, bütün parametrelerin en uygun şekilde ayarlanması durumunda dahi tek bir parametrenin yanlış ayarlanmasının sonuçları ciddi şekilde olumsuz yönde etkilemesidir.

Önerilen model MNIST ve MUD veri tabanları üzerinde ayrı ayrı test edilmiştir. MNIST veri setinin test veri kümesinde %89.01 tanıma doğruluğu elde ederek önemli bir başarı sağlamıştır. Ayrıca, önerilen model aynı veri seti üzerinde yapılan diğer çalışmalarla karşılaştırıldığında birçok bilinen çalışma ve modeli geride bırakarak, CNN'in ardından ikinci olmuştur.

MUD veri seti, eğitim ve test kümeleri olarak ayrılmadığından, önerilen model Holdout ve LOSO tekniklerine göre ayrı ayrı değerlendirilmiştir. Holdout tekniğine uygun olarak veri seti %70 eğitim, %30 test kümesi olacak şekilde ayrıldığında, test kümesinde %98.67 tanıma oranı elde edilmiştir. Aynı veri seti üzerinde yapılan birçok çalışmayla kıyaslandığında, önerilen model en yüksek tanıma oranını elde etmiştir. İlgili tekniğe göre yapılan çalışmalar incelendiğinde, birçok çalışmanın veri setini %80 eğitim, %20 test verisi olarak ayırdığı görülmüştür. Benzer şekilde, MUD veri seti %80 eğitim, %20 test verisi olarak ayrıldığında tanıma oranı %99.60 seviyesine yükselmiştir. Bu değerlerin karmaşıklık matrisinden elde edilen doğruluk oranı değil, tanıma oranı olduğunu tekrar hatırlatmakta fayda bulunmaktadır.

LOSO tekniğine bağlı olarak MUD veri seti üzerinde gerçekleştirilen çalışmalarda, önerilen model %81.22 tanıma oranı elde etmiş ve CNN tabanlı birçok çalışmayı geride bırakarak ikinci olmuştur. Elde edilen sonuçlar üzerinden yapılan analizlerde, Denek 5'e ait skorun (eğitimde Denek 5 yok, test kümesinde var) düşük olduğu ve dolayısıyla ana skoru olumsuz etkilediği görülmüştür. Literatürdeki bazı çalışmalarda da Denek 5'e ait sonuçların düşük olduğu gözlemlenmiş ve bu deneğin diğer deneklere göre zorlu ve etkili bir değerlendirme aracı olduğu, elde edilen sonuçlardan da tespit edilmiştir.

Geliştirilen modelin entegre edildiği bir gerçek zamanlı uygulama da oluşturulmuştur. Bu uygulama, web kamerasından gelen görüntü akışını anlık olarak işleyerek öncelikle el tespiti ve takibini gerçekleştirir. El tespiti önerilen modele uygun şekilde (birim disk sebebiyle karesel) yapıldıktan sonra, özellik çıkarım modülü ile tespit edilen alanın öznitelik vektörü elde edilir. Bu vektör, CVDNN modülüne yönlendirilir ve Holdout (70-30) tekniğiyle eğitilen model üzerinden değerlendirilir. Ancak burada önemli bir hususa dikkat

çekmek gerekir. Holdout tekniğine dayalı bir model kullanılıyor olmasına rağmen, uygulamanın işleyişi aslında LOSO tekniğine benzerdir. Bunun nedeni, test sırasında kullanılan el görüntülerinin, eğitim verisi içerisinde yer almamasıdır. İlgili uygulama üzerinde yapılan çalışmalar sırasında, ışık ve arka plan faktörlerinin oldukça etkili olduğu gözlemlenmiştir. MUD veri setinin arka planının tamamen siyah olması avantajından yararlanarak, uygulama için siyah arka planlı bir ortam hazırlanarak geliştirilen sistem test edilmiştir. Ayrıca, ışık koşulları uygun şekilde ayarlandığında ve el işaretlerini düzgün bir şekilde gerçekleştirildiğinde, %90'ın üzerinde bir tanıma oranı elde edilmiştir.



5. ÖNERİLER

Karmaşık Zernike momentler kullanılarak önerilen yöntemle geliştirilen model, başarılı sonuçlar elde etmiş olsa da gerçek zamanlı uygulamada karşılaşılan bazı durumlar, gelecekte yapılacak geliştirmeler için yol gösterici olmuştur. Bu bağlamda, ışık ve arka plan problemlerinin çözülmesi hedeflenmektedir. Her ne kadar iki ayrı faktör gibi görünseler de, temelde bu durum bütünlük olarak değerlendirilebilir. Işık problemi, arka planın varlığına bağlı olarak artmaktadır. Arka planın çıkarılabileceği bir durumda ışığın etkisi çok daha düşük olacaktır. Bu nedenle, gelecekteki en önemli çalışmalar arasında arka planın çıkarılması öne çıkmaktadır. Belirtilen ön işleme adımlarını görüntülere uygulamanın başarı oranlarını önemli ölçüde artırabileceği düşünülmektedir.

Görüntü akışında el tespiti gerçekleştirildikten sonra veya sabit bir görüntü kullanıldığında, özellik çıkarımı modülünde ilgili görüntü üzerinde birim çember oluşturulmaktadır. Bu bağlamda, bazı resimlerde hedeflenen verinin çemberin dışında kalma olasılığı oldukça yüksektir. Bu durum, hem öğrenme hem de test aşamalarında dolaylı olarak olumsuz bir etki yaratmaktadır. Bu noktada hedeflenen veriyi çemberin içine düşürecek şekilde bir karesel alanın seçilmesinin olası veri kaybını azaltarak sistemin performansını artıracığı düşünülmektedir.

Gerçek zamanlı sistemin daha hızlı tepki verebilmesi için, makul başarı (tanıma) oranlarını koruyarak daha düşük çözünürlüklü veri veya veri setleri ile eğitilen modeller kullanılmalıdır. Bu doğrultuda, en uygun değere ulaşılan kadar çeşitli çalışmalar yapılması gerekmektedir.

Gerçekleştirilen akıllı tanıma sistemi sabit görüntüler üzerinde oldukça yüksek başarı oranları elde etmiştir. Arka plan sorunlarının çözülmesinin ardından, bir sonraki aşamada hareketli görüntüler (çoğunlukla kelimeler) üzerinde çalışabilecek bir sistemin geliştirilmesi hedeflenmektedir. Belirtilen öneriler sisteme dahil edildiğinde, ilgili sistem mobil cihazlar üzerinde çalışabilecek bir yapıya sahip olacaktır. Bu sayede, bu alanda işlevsel bir mobil uygulama geliştirmek mümkün olacaktır.

Bütün belirtilen hedeflerin dışında, önerilen yöntem literatürde belirtilen bazı çalışmalara benzer şekilde, diğer model veya yöntemlerle birleştirilerek hibrit bir yaklaşım olarak da denenebilir.

Karmaşık Zernike momentleri ve karmaşık değerli derin sinir ağıları, görüntü tanıma (özellik çıkarımı ve sınıflandırma), şekil tanıma (tıbbi görüntüleme), biyometrik sistemler (parmak izi ve retina tarama), görüntü restorasyonu, radar teknolojileri, sinyal işleme, askeri sistemler, telekomünikasyon, robotik, otonom sistemler ve makine öğrenmesi gibi geniş bir uygulama yelpazesinde kullanılmaktadır. Genellikle gerçekleştirilen akademik çalışmalar elektrik-elektronik ve alt bilim dallarını kapsadığından, diğer alanlar üzerinde çeşitli akademik çalışmalara da yoğunlaşılabilir.



6. KAYNAKLAR

- Abdalla, R. (2024, 12 Nisan). Complex-valued Neural Networks -- Theory and Analysis: <https://arxiv.org/pdf/2312.06087> adresinden alınmıştır.
- Ansar, H., Mudawi, N. A., Alotaibi, S. S., Alazeb, A., Alabdullah, B. I., Alonazi, M., & Park, J. (2023). Hand Gesture Recognition for Characters Understanding Using Convex Hull Landmarks and Geometric Features. *IEEE Access*, 11, 82065-82078.
- Thakur, A. (2023, 14 Eylül). *American Sign Language Dataset*: <https://www.kaggle.com/datasets/ayuraj/asl-dataset/> adresinden alınmıştır.
- Baldi, P. (2011). Autoencoders, unsupervised learning and deep architectures. *UTLW'11: Proceedings of the 2011 International Conference on Unsupervised and Transfer Learning workshop*, s. 37-50, USA.
- Barczak, A.L.C., Reyes, N.H., Abastillas, M., Piccio, A., & Susnjak, T. (2011). A new 2D static hand gesture colour image dataset for ASL gestures. *Res. Lett. Inf. Math. Sci.*, 15, 12–20.
- Bassey, J., Li, X., & Qian, L. (2024, 15 Mayıs). A Survey of Complex-Valued Neural Networks: <https://arxiv.org/pdf/2101.12249> adresinden alınmıştır.
- Benvenuto, N., & Piazza, F. (1992). On the complex backpropagation algorithm. *IEEE Transactions on Signal Processing*, 40(4), 967-969.
- Bhatt, D., Patel, C., Talsania, H., Patel, J., Vaghela, R., Pandya, S., Modi, K., & Ghayvat, H. (2021). CNN Variants for Computer Vision: History, Architecture, Application, Challenges and Future Scope. *Electronics*, 10(20), 2470.
- Bhaumik, G., & Govil, M.C. (2024). SpAtNet: a spatial feature attention network for hand gesture recognition. *Multimedia Tools and Applications*, 83, 41805–41822.
- Bhuiyan, A-A., & Khan, A. R. (2018). Image Quality Assessment Employing RMS Contrast and Histogram Similarity. *The International Arab Journal of Information Technology*, 15(6), 983-989.
- Bhushan, S., Alshehri, M., Keshta, I., Chakraverti, A. K., Rajpurohit, J., & Abugabah, A. (2022). An Experimental Analysis of Various Machine Learning Algorithms for Hand Gesture Recognition. *Electronics*, 11(6), 968.
- Bilgin, M., & Mutludoğan, K. (2019). American Sign Language Character Recognition with Capsule Networks. *2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, s. 1-6, Türkiye.
- Barbhuiya, A.A., Karsh, R.K., & Jain, R.. (2021). CNN based feature extraction and classification for sign language. *Multimedia Tools and Applications*, 80, 3051-3069.

- Barbhuiya, A.A., Karsh, R.K., & Jain, R. (2022). A convolutional neural network and classical moments-based feature fusion model for gesture recognition. *Multimedia Systems*, 28, 1779–1792.
- Chanthorn, P., Rajchakit, G., Thipcha, J., Emharuethai, C., Sriraman, R., Lim, C. P., & Ramachandran, R. (2020). Robust Stability of Complex-Valued Stochastic Neural Networks with Time-Varying Delays and Parameter Uncertainties. *Mathematics*, 8(5), 742.
- Chen, Z., & Sun, S.K. (2010). A Zernike moment phase-based descriptor for local image representation and matching. *IEEE Transactions on Image Processing*, 19(1), 205-219.
- Chevtchenko, S. F., Vale, R. F., Macario, V., & Cordeiro, F. R. (2018). A convolutional neural network with feature fusion for real-time hand posture recognition. *Applied Soft Computing Journal*, 73, 748-766.
- Corriou J-P. (2022). *Numerical Methods and Optimization Theory and Practice for Engineers*. Switzerland: Springer.
- Cui, R., Liu H., & Zhang, C. (2019). A Deep Neural Framework for Continuous Sign Language Recognition by Iterative Training. *IEEE Transactions on Multimedia*, 21(7), 1880-1891.
- Dai, X.B., Shu, H.Z., Luo, L.M., Han, G.N., & Coatrieux, J.L. (2010). Reconstruction of tomographic images from limited range projections using discrete Radon transform and Tchebichef moments. *Pattern Recognition*, 43(3), 1152-1164.
- Dima. (2024, 15 Temmuz). *ASL Benchmark Dataset (YOLOv5 PyTorch Format)*: <https://www.kaggle.com/datasets/tasnimdima/datazip> adresinden alınmıştır.
- Dong, S., Wang P., & Abbas K. (2021). A survey on deep learning and its applications. *Computer Science Review*, 40, 100379.
- Flusser, J., Suk, T., & Zitová, B. (2009). *Moments and Moment Invariants in Pattern Recognition*. Great Britain: John Wiley & Sons Ltd.
- Fischer, A., & Igel, C. (2012). An Introduction to Restricted Boltzmann Machines. *17th Iberoamerican Congress, CIARP 2012*, s. 14-36, Argentina.
- Fischer, R. F. H., (2002). *Appendix A: Wirtinger Calculus*. USA: John Wiley & Sons Ltd.
- Goodfellow, I. J., Pouget-Abadie J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2024, 22 Nisan). Generative Adversarial Networks: <https://arxiv.org/abs/1406.2661> adresinden alınmıştır.
- Hansen, W. (2008). A Strong Version of Liouville's Theorem. *The American Mathematical Monthly*, 115(7), 583–595.
- Hu, M.K. (1962). Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, 8(2), 179-187.

- Idrees, H. (11 Haziran 2024). RNN vs. LSTM vs. GRU: A Comprehensive Guide to Sequential Data Modeling: <https://medium.com/@hassaanidrees7/rnn-vs-lstm-vs-gru-a-comprehensive-guide-to-sequential-data-modeling-03aab16647bb> adresinden alınmıştır.
- Jiuxiang, G., Zhenhua, W., Jason, K., Lianyang, M., Amir, S., Bing, S., Ting, L., Xingxing, W., Gang, W., Jianfei, C., & Tsuhan, C. (2018). Recent advances in convolutional neural networks. *Pattern Recognition*, 77, 354-377.
- Kagalkar, A., & Raghuram, S. (2020). CORDIC Based Implementation of the Softmax Activation Function. *2020 24th International Symposium on VLSI Design and Test (VDATE)*, s. 1-4, India.
- Kandel, I., & Castelli, M. (2020). The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset. *ICT Express*, 6(4), 312-315.
- Karami, A., Zanj, B., & Sarkaleh, A. K. (2011). Persian sign language (PSL) recognition using wavelet transform and neural networks. *Expert Systems with Applications*, 8(3), 2661-2667.
- Kelly, D., McDonald, J., & Markham, C. (2010). A person independent system for recognition of hand postures used in sign language. *Pattern Recognition Letters*, 31(11), 1359-1368.
- Kim, H.S., & Lee, H.K. (2003). Invariant image watermark using Zernike moments. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(8), 766-775.
- Kim, S.-H., Geem, Z.W., & Han, G.-T. (2020). Hyperparameter Optimization Method Based on Harmony Search Algorithm to Improve Performance of 1D CNN Human Respiration Pattern Recognition System. *Sensors*, 20(13), 3697.
- Kingma, D. P., & Ba, J. (2024, 13 Mayıs). Adam: A Method for Stochastic Optimization: <https://arxiv.org/pdf/1412.6980> adresinden alınmıştır.
- Khotanzad, A., & Hong, Y.H. (1990). Invariant image recognition by Zernike moments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(5), 489-497.
- Koller, O., Zargaran, S., Ney, H., & Bowden, R. (2018). Deep Sign: Enabling Robust Statistical Continuous Sign Language Recognition via Hybrid CNN-HMMs. *International Journal of Computer Vision*, 126, 1311-1325.
- Krizhevsky, A., Sutskever, I., & Hinton, G.,E. (2012). ImageNet classification with deep convolutional neural networks. *NIPS'12: Proceedings of the 25th International Conference on Neural Information Processing Systems*, s. 1097-1105, USA.
- Kumar, S. K. (2024, 23 Mart). On weight initialization in deep neural networks: <https://arxiv.org/pdf/1704.08863> adresinden alınmıştır.
- Kumar, Y., Aggarwal, A., Tiwari, S., & Singh, K. (2018). An efficient and robust approach for biomedical image retrieval using Zernike moments. *Biomedical Signal Processing and Control*, 39, 459-473.
- Liang, X., & Xu, J. (2021). Biased ReLU neural networks. *Neurocomputing*, 423, 71-79.

- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep Learning. *Nature*, 521 (7553), 436.
- Lee, C., Hasegawa, H., & Gao, S. (2022). Complex-Valued Neural Networks: A Comprehensive Survey. *IEEE/CAA Journal of Automatica Sinica*, 9(8), 1406-1426.
- Leung, H., & Haykin, S. (1991). The complex backpropagation algorithm. *IEEE Transactions on Signal Processing*, 39(9), 2101-2104.
- Li, L., Liu, D., Shen, C., & Sun, J. (2022). American Sign Language Recognition Based on Machine Learning and Neural Network. *2022 International Conference on Machine Learning and Intelligent Systems Engineering (MLISE)*, s. 452-457, China.
- Li, S., Lee, M. -C. & Pun, C. -M. (2009). Complex Zernike Moments Features for Shape-Based Image Retrieval. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 39(1), 227-237.
- Lu, X., & Yang, J. (2023). Image analysis with logarithmic Zernike moments. *Digital Signal Processing*, 133.
- Mahdian, B., & Saic, S. (2007). Detection of copy-move forgery using a method based on blur moment invariants. *Forensic Science International*, 171(2-3), 180-189.
- Maleki, F., Muthukrishnan, N., Ovens, K., Reinhold, C., & Forghani, R. (2020). Machine Learning Algorithm Validation: From Essentials to Advanced Applications and Implications for Regulatory Certification and Deployment. *Neuroimaging Clin N Am.*, 30(4), 433-445.
- Mukundan, R., & Ramakrishnan, K. R. (1998). *Moment Functions in Image Analysis: Theory and Applications*, Singapore: World Scientific Publishing Co. Pte. Ltd.
- Nacar, E. N., & Erdebilli B. (2021). Makine öğrenmesi algoritmaları ile satış tahmini. *Journal of Industrial Engineering*, 32(2), 307-320.
- Nandy, A., Prasad, J. S., Mondal, S., Chakraborty, P., & Nandi, G. C. (2010). Recognition of Isolated Indian Sign Language Gesture in Real Time. *Information Processing and Management*, 102-107.
- Nash, J. F. (1950). Equilibrium points in n-person games. *Mathematics*, 36(1), 48-49.
- Nitta, T. (1997). An Extension of the Back-Propagation Algorithm to Complex Numbers. *Neural Networks*, 10(8), 1391-1415.
- Palubinskas, G. (2016). Image similarity/distance measures: what is really behind MSE and SSIM. *International Journal of Image and Data Fusion*, 8(1), 32-53.
- Pham, K., Kim, D., Park, S., & Choi, H. (2021). Ensemble learning-based classification models for slope stability analysis. *CATENA*, 196.
- Pinaya, W. H. L., Vieira, S., Garcia-Dias, R., & Mechelli A. (2020). *Machine Learning Chapter 11 - Autoencoders*. USA:Academic Press.

- Qi, S., Zhang, Y., Wang, C., Zhou, J., & Cao, X. (2021). A Survey of Orthogonal Moments for Image Representation: Theory, Implementation, and Evaluation. *ACM Comput. Surv.* 55(1),1-35.
- Qiumei, Z., Dan, T., & Fenghua, W. (2019). Improved Convolutional Neural Network Based on Fast Exponentially Linear Unit Activation Function. *IEEE Access*, 7, 151359-151367.
- Rao, G. A., Syamala, K., Kishore, P. V. V., & Sastry, A. S. C. S. (2018). Deep convolutional neural networks for sign language recognition. *2018 Conference on Signal Processing and Communication Engineering Systems (SPACES)*, s. 194-197, India.
- Rastgoo, R., Kiani, K., & Escalera, S. (2018). Multi-Modal Deep Hand Sign Language Recognition in Still Images Using Restricted Boltzmann Machine. *Entropy*, 20(11), 809.
- Rastgoo, R., Kiani, K., & Escalera, S. (2021). Sign Language Recognition: A Deep Survey. *Expert Systems with Applications*, 164.
- Samaan, G. H., Wadie, A.R., Attia, A. K., Asaad, A. M., Kamel, A. E., Slim, S. O., Abdallah M. S., & Cho Y. -I. (2022). MediaPipe's Landmarks with RNN for Dynamic Sign Language Recognition. *Electronics*, 11(19), 3228.
- Sharma, V., Sharma, A., & Saini, S. (2024). Real-time attention-based embedded LSTM for dynamic sign language recognition on edge devices. *Journal of Real-Time Image Processing*, 21, 53.
- Sherstinsky, A. (2020). Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network. *Physica D: Nonlinear Phenomena*, 404, 132306.
- Shrestha, A., & Mahmood, A. (2019). Review of Deep Learning Algorithms and Architectures, *IEEE Access*, 7, 53040-53065.
- Shiliang, S., Cao, Z., Zhu, H., & Zhao, J. (2024, 03 Mart). A Survey of Optimization Methods from a Machine Learning Perspective: <https://arxiv.org/abs/1906.06821> adresinden alınmıştır.
- Simon, H. (2014). *Adaptive Filter Theory Fifth Edition*, USA: Pearson Education Limited.
- Singh, C., Mittal, N., & Walia, E. (2011). Face recognition using Zernike and complex Zernike moment features. *Pattern Recognition and Image Analysis*, 21, 71–81.
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *The Journal of Machine Learning Research*, 15(1), 1929-1958.
- Sreemathy, R., Turuk, M., Kulkarni, I., & Khurana, S. (2023). Sign language recognition using artificial intelligence. *Education and Information Technologies*, 28, 5259–5278.
- Sun S., Cao Z., Zhu H. and Zhao J. (2020). A Survey of Optimization Methods From a Machine Learning Perspective, *IEEE Transactions on Cybernetics*, 50(8), 3668-3681.

- Sutskever, I., Martens, J., Dahl, G., and Hinton, G. (2013). On the importance of initialization and momentum in deep learning. *Proceedings of the 30th International Conference on Machine Learning*, s. 1139-1147, USA.
- Sze, V., Chen, Y. -H., Yang, T. -J., & Emer, J. S. (2017). Efficient Processing of Deep Neural Networks: A Tutorial and Survey, *Proceedings of the IEEE*, 105(12), 2295-2329.
- Teague, M.R. (1980). Image analysis via the general theory of moments. *J. Opt. Soc.Am.*, 70(8), 920-930.
- Tecperson. (2023, 20 Ağustos). *Sign Language MNIST*: <https://www.kaggle.com/datasets/datamunge/sign-language-mnist> adresinden alınmıştır.
- Teh, C. -H., & Chin, R. T. (1988). On image analysis by the methods of moments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(4), 496-513.
- Teh, Y. W., & Hinton, G. E. (2000). Rate-coded restricted Boltzmann machines for face recognition. *NIPS'00: Proceedings of the 13th International Conference on Neural Information Processing Systems*, s. 872-878, USA.
- URL1. (2024, Şubat 24). *Deafness and hearing loss*: <https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss> adresinden alınmıştır..
- URL2. (2023, Aralık 18). *torch.nn.functional.nll_loss*: https://pytorch.org/docs/stable/generated/torch.nn.functional.nll_loss.html adresinden alınmıştır.
- URL3. (2024, Şubat 08). *Hand landmarks detection guide*: https://ai.google.dev/edge/mediapipe/solutions/vision/hand_landmarker adresinden alınmıştır.
- Virtue, P., Yu, S. X., & Lustig, M. (2017). Better than real: Complex-valued neural nets for MRI fingerprinting. *2017 IEEE International Conference on Image Processing (ICIP)*, s. 3953-3957, China.
- Wadhawan, A., & Kumar, P. (2021). Sign Language Recognition Systems: A Decade Systematic Literature Review. *Archives of Computational Methods in Engineering*, 28, 785–813.
- Zernike, F., & Stratton, F.J.M. (1934). Diffraction Theory of the Knife-Edge Test and its Improved Form, The Phase-Contrast Method. *Monthly Notices of the Royal Astronomical Society*, 94(5), 377-384.
- Zhang, H., Shu, H., Han, G.N., Coatrieux, G., Luo, L., & Coatrieux, J. L. (2010). Blurred Image Recognition by Legendre Moment Invariants. *IEEE Transactions on Image Processing*, 10(3), 596-611.
- Zhou, H., Zhou, W., Zhou, Y., & Li, H. (2022). Spatial-Temporal Multi-Cue Network for Sign Language Recognition and Translation. *IEEE Transactions on Multimedia*, 24, 768-779.

ÖZGEÇMİŞ

Selda Bayrak İlk ve orta öğrenimini Trabzon'da tamamlamıştır. 1999 yılında Fatih Süper Lisesi'nden mezun olduktan sonra 2000 yılında Karadeniz Teknik Üniversitesi Bilgisayar Mühendisliği Bölümünde lisans eğitimine başlamıştır. İngilizce hazırlık sınıfı da dahil olmak üzere 2006 yılında lisans eğitimini tamamladıktan sonra aynı yıl Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalında yüksek lisansına başlamış ve 2007 yılı içinde Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği bölümüne araştırma görevlisi olarak atanmıştır. 2009 yılında Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalında doktora eğitimine başlamıştır. 2017 yılında Karadeniz Teknik Üniversitesi Of Teknoloji Fakültesi Yazılım Mühendisliğinde öğretim görevlisi olarak çalışmaya başlamış ve halen ilgili bölümde çalışmaya devam etmektedir.

Yabancı dil olarak İngilizce bilmektedir ve SCI, SSCI, AHCI indeksi kapsamındaki yayını aşağıda verilmiştir.

✉ Selda Bayrak, Vasif Nabiye, Celal Atalar, American Sign Language Recognition Model Using Complex Zernike Moments and Complex-Valued Deep Neural Networks, IEEE Access, 2024.

<https://doi.org/10.1109/ACCESS.2024.3461572>