

ANKARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

İŞARET DİLİ TANIMADA DERİN ÖĞRENME İLE İSTATİSTİKSEL
YÖNTEMLERİN KARŞILAŞTIRILMASI

Caner DOĞAN

İSTATİSTİK ANABİLİM DALI

ANKARA
2024

Her hakkı saklıdır

ÖZET

Yüksek Lisans Tezi

İŞARET DİLİ TANIMADA DERİN ÖĞRENME İLE İSTATİSTİKSEL YÖNTEMLERİN KARŞILAŞTIRILMASI

Caner DOĞAN

Ankara Üniversitesi
Fen Bilimleri Enstitüsü
İstatistik Anabilim Dalı

Danışman: Doç. Dr. Levent ÖZBEK

Bu çalışmada, Türk İşaret Dili tanıma alanında derin öğrenme yaklaşımlarının istatistiksel analizi ve performans karşılaştırması gerçekleştirilmiştir. Derin öğrenme modellerinin performans değerlendirmesi, hiperparametre optimizasyonu ve model seçimi süreçlerinde istatistiksel analizler kritik bir rol oynamaktadır. Bu bağlamda, Evrişimsel Sinir Ağları (CNN), Uzun Kısa Süreli Bellek (LSTM) ve CNN-LSTM hibrit modellerinin işaret dili tanımadaki etkinliği istatistiksel metrikler kullanılarak karşılaştırılmıştır. MediaPipe çerçevesi ile elde edilen el ve poz özellikleri üzerinden geliştirilen modeller, Aile ve Sosyal Hizmetler Bakanlığı tarafından oluşturulan ASBTSL veri kümesinden seçilen, en sık kullanılan 20 isim türü Türk İşaret Dili kategorisini içeren 9188 videodan oluşan veri seti ile eğitilmiştir. Veri seti %70 eğitim, %15 doğrulama ve %15 test olarak bölünmüştür. Modellerin performansı, olasılık teorisi ve istatistiksel çıkarım yöntemleri ışığında doğruluk, kesinlik, duyarlılık ve F1 skoru gibi metriklerle değerlendirilmiştir. CNN-LSTM hibrit modeli, %90,19 doğruluk oranıyla en yüksek performansı göstermiştir. Bu araştırma, derin öğrenme süreçlerinde istatistiksel analizlerin önemini vurgulamakta ve işaret dili tanıma alanındaki çalışmalara katkı sağlamaktadır. Bu alanda geliştirilecek uygulamalar için sağlam bir temel sunmaktadır.

Aralık 2024, 145 sayfa

Anahtar Kelimeler: İşaret Dili Tanıma, Derin Öğrenme, İstatistiksel Analiz, Türk İşaret Dili, CNN, LSTM

ABSTRACT

Master's Thesis

COMPARISON OF DEEP LEARNING AND STATISTICAL METHODS IN SIGN LANGUAGE RECOGNITION

Caner DOĞAN

Ankara University
Graduate School of Natural and Applied Sciences
Department of Statistics

Supervisor: Assoc. Prof. Dr. Levent ÖZBEK

A statistical analysis and performance comparison of deep learning approaches in Turkish Sign Language recognition has been conducted. Statistical analyses play a critical role in the performance evaluation, hyperparameter optimization, and model selection processes of deep learning models. In this context, the effectiveness of Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM), and CNN-LSTM hybrid models in sign language recognition has been compared using statistical metrics. The models, developed using hand and pose features obtained through the MediaPipe framework, were trained on a dataset consisting of 9188 videos containing the 20 most frequently used noun-type Turkish Sign Language categories, selected from the ASBTSL dataset created by the Ministry of Family and Social Services. The dataset was split into 70% training, 15% validation, and 15% testing subsets. The models' performance was evaluated using metrics such as accuracy, precision, recall, and F1 score in the context of probability theory and statistical inference methods. The CNN-LSTM hybrid model achieved the highest performance with an accuracy rate of 90.19%. This research emphasizes the importance of statistical analyses in deep learning processes and contributes to studies in the field of sign language recognition. It provides a solid foundation for future applications to be developed in this area.

December 2024, 145 pages

Key Words: Sign Language Recognition, Deep Learning, Statistical Analysis, Turkish Sign Language, CNN, LSTM

TEŞEKKÜR

Bu tez çalışmasının her aşamasında engin bilgi birikimi ve tecrübesiyle bana yol gösteren, araştırmacı kimliği ve akademik duruşuyla her zaman örnek aldığım, bilimsel bakış açısıyla ufkumu genişleten ve öğrencisi olmaktan onur duyduğum değerli danışman hocam Doç. Dr. Levent ÖZBEK'e en içten saygı ve teşekkürlerimi sunarım.

Hayatım boyunca bana olan inançlarını hiç yitirmeyen, her türlü fedakarlığı göstererek bugünlere gelmemi sağlayan, başarılarımın gerçek mimarları olan sevgili annem Gülten DOĞAN ve babam Hayri DOĞAN'a sonsuz minnet ve teşekkürlerimi sunarım. Hayatımın her anında yanımda olan, desteğini ve sevgisini hiç esirgemeyen, canım kardeşim Cansel YAHŞİ'ye tüm kalbimle teşekkür ederim.

Bu zorlu süreçte anlayışı, sabrı ve sevgisiyle her zaman yanımda olan, tez çalışmam boyunca bana destek olan, hayat arkadaşım, sevgili eşim Emel DOĞAN'a en derin duygularıyla teşekkür ederim. Varlığıyla hayatıma anlam katan, gülüşüyle yorgunluğumu alan, en büyük motivasyon kaynağım, canım oğlum Kaan Can DOĞAN'a sonsuz sevgiyle teşekkür ederim.

Caner DOĞAN
Ankara, Aralık 2024

İÇİNDEKİLER

TEZ ONAY SAYFASI

ETİK.....	i
ÖZET.....	ii
ABSTRACT	iii
TEŞEKKÜR	iv
SİMGELER DİZİNİ	viii
ŞEKİLLER DİZİNİ	ix
ÇİZELGELER DİZİNİ	xii
1. GİRİŞ	1
1.1 Genel Bakışı	1
1.2 İlgili Çalışmalar	5
1.3 İşaret Dili Tanıma'nın Zorlukları ve Mevcut Durum	6
1.4 Türk İşaret Dili (TİD).....	7
1.5 İşaret Dili Tanıma	8
1.6 Tezin Amacı	12
2. YAPAY ZEKA	13
2.1 İşaret Diliyle Yapay Zeka Tabanlı Erişilebilirlik ve Teknoloji Entegrasyonu..	14
3. OLASILIK VE BİLGİ KURAMI.....	16
4. MAKİNE ÖĞRENMESİ	17
4.1 Makine Öğrenmesi Türleri.....	18
4.1.1 Denetimli öğrenme (Supervised learning)	18
4.1.2 Denetimsiz öğrenme (Unsupervised learning).....	19
4.1.3 Pekiştirmeli öğrenme (Reinforcement learning).....	20
4.2 Makine Öğrenmesi Uygulamaları	20
4.2.1 Sınıflandırma (Classification).....	21
4.2.2 Kümeleme (Clustering)	22
5. DERİN ÖĞRENME.....	24
5.1 Tek Katmanlı Sinir Ağları	27
5.2 Çok Katmanlı Sinir Ağları	29
5.3 Derin Öğrenme Tanımı ve Tarihçesi.....	33
5.4 Derin Öğrenmenin Kullanım Alanları	34
5.4.1 Görüntü tanıma	35
5.4.2 Doğal dil işleme	36

5.4.3 Derin yapay sinir ağı uygulama	38
5.4.4 Tekrarlamalı Sinir Ağları (RNN).....	40
5.4.4.1 Uzun Kısa Süreli Bellek Ağları (LSTM)	44
5.4.5 Evrimsel Sinir Ağları (CNN).....	44
5.4.5.1 Evrimsel sinir ağının mimarisi	48
5.4.5.2 Evrim katmanı	50
5.4.5.3 Havuzlama katmanı	54
5.4.5.4 Tam bağlantılı katman.....	55
5.5 Derin Öğrenme Sorunları ve Zorlukları.....	56
5.5.1 Veri ihtiyacı	56
5.5.2 Model yorumlanabilirliği	57
5.5.3 Hesaplama kaynakları	57
5.5.4 Aşırı öğrenme (Overfitting) ve Eksik öğrenme (Underfitting).....	58
5.6 Hiper Parametre Seçimleri	59
5.6.1 Epoch sayısı seçimi	61
5.6.2 Seyreltme (Dropout) öğrenme	62
5.6.3 Katman sayısının belirlenmesi.....	63
5.7 Aktivasyon Fonksiyonu	64
5.7.1 Sigmoid fonksiyonu	65
5.7.2 Softmax fonksiyonu	66
5.7.3 Hiperbolik tanjant fonksiyonu (Tanh)	67
5.7.4 Doğrultulmuş lineer birim (ReLU)	68
5.8 Optimizasyon Yöntemleri.....	70
5.9 Kayıp Fonksiyonu	70
5.9.1 ADAM optimizasyonu	71
5.9.2 Düzenleştirme ve stokastik gradyan inişi	72
5.9.3 Geri yayılım.....	72
5.10 Transfer Öğrenme	74
5.11 Derin Öğrenmede Kullanılan Yazılım Kütüphaneleri	75
5.11.1 Tensorflow	75
5.11.2 Python	76
6. GÖRÜNTÜ İŞLEME.....	77
6.1 Görüntü İşleme Teknikleri.....	77
6.2 Görüntü İşlemenin Uygulama Alanları	78
7. MEDİAPİPE İLE ANAHTAR NOKTA ÇIKARMA	79

8. PERFORMANS DEĞERLENDİRME ÖLÇÜMLERİ	82
8.1 Ortalama Mutlak Hata (Mean Absolute Error- MAE).....	82
8.2 Ortalama Mutlak Hata Yüzdesi (Mean Absolute Percentage Error- MAPE) ..	82
8.3 Hata Kareler Ortalaması (Mean Squared Error- MSE).....	82
8.4 Hata Kareler Ortalamasının Karekökü (Root Mean Squared Error- RMSE).83	
9. MODEL DEĞERLENDİRME METRİKLERİ	84
9.1 Karmaşıklık Matrisi	85
9.1.1 Doğruluk	86
9.1.2 Kesinlik	87
9.1.3 Duyarlılık	88
9.1.4 Özgüllük veya doğru negatif oran	89
9.1.5 Yanlış pozitif oranı	90
9.1.6 F1 skoru	90
10. TEZDE KULLANILAN VERİ SETİ VE VERİ ANALİZİ	92
10.1 ELAN Aracı ile Dilbilimsel Video Analizi	92
10.2 Veri Kümesi İstatistikleri ve İşaret Sınıfı Adları	97
10.3 Veri Önışleme	102
10.4 Özellik Çıkarım	103
10.5 Veri Artırımı.....	105
10.6 Veri Setlerinin Düzenlenmesi.....	106
11. YÖNTEM.....	108
11.1 Model Eğitimi	110
12. BULGULAR.....	113
12.1 Doğruluk ve Kayıp Eğrisi.....	114
12.2 Karmaşıklık Matrisi ve Performans Ölçümü.....	118
13. GERÇEK ZAMANLI TAHMİN	125
14. DEĞERLENDİRME VE SONUÇ	129
14.1 Derin Öğrenme Yaklaşımları.....	129
14.2 Karşılaştırmalı Analiz	130
15. SONUÇ VE GELECEK ÇALIŞMALAR	132
KAYNAKLAR	134
ÖZGEÇMİŞ.....	144

SİMGELER DİZİNİ

b	Bias (yanlılık) terimi
e	Euler sayısı
$f(x)$	Aktivasyon fonksiyonu
n	Toplam örnek sayısı
p	Girdi vektörü boyutu
w	Ağırlık değeri
x	Girdi değeri
y	Çıktı değeri
θ	Model parametreleri
ρ	Öğrenme oranı
ϕ	Seyreltme (dropout) oranı
λ	Düzenleştirme parametresi
∇	Gradyan operatörü

Kısaltmalar

ADAM	Adaptive Moment Estimation
ASBTSL	Aile ve Sosyal Hizmetler Bakanlığı Türk İşaret Dili Veri Seti
ASL	American Sign Language (Amerikan İşaret Dili)
AUTSL	Ankara University Turkish Sign Language Dataset
CNN	Convolutional Neural Network (Evrşimsel Sinir Ağı)
DTW	Dynamic Time Warping (Dinamik Zaman Bükme)
FN	False Negative (Yanlış Negatif)
FP	False Positive (Yanlış Pozitif)
GPU	Graphics Processing Unit (Grafik İşlem Birimi)
HMM	Hidden Markov Model (Gizli Markov Modeli)
ISL	Irish Sign Language (İrlanda İşaret Dili)
LSTM	Long Short-Term Memory (Uzun Kısa Süreli Bellek)
MAE	Mean Absolute Error (Ortalama Mutlak Hata)
MAPE	Mean Absolute Percentage Error (Ortalama Mutlak Hata Yüzdesi)
MSE	Mean Squared Error (Hata Kareler Ortalaması)
ReLU	Rectified Linear Unit (Doğrultulmuş Lineer Birim)
RGB	Red Green Blue (Kırmızı Yeşil Mavi)
RMSProp	Root Mean Square Propagation
RMSE	Root Mean Squared Error (Hata Kareler Ortalamasının Karekökü)
RNN	Recurrent Neural Network (Tekrarlayan Sinir Ağı)
SL	Sign Language (İşaret Dili)
SLR	Sign Language Recognition (İşaret Dili Tanıma)
SVM	Support Vector Machine (Destek Vektör Makinesi)
TİD	Türk İşaret Dili
TN	True Negative (Doğru Negatif)
TP	True Positive (Doğru Pozitif)
WHO	World Health Organization (Dünya Sağlık Örgütü)

ŞEKİLLER DİZİNİ

Şekil 1.1 Türk işaret dilinde “MERHABA” işareti.....	1
Şekil 1.2 Statik ve dinamik hareket örneği (“ANNE” işaret dili)	10
Şekil 4.1 Sınıflandırma sınırları örneği	22
Şekil 4.2 Kümeleme işlemi örneği	23
Şekil 5.1 Tek bir gizli katmana sahip sinir ağı	24
Şekil 5.2 Tipik bir derin öğrenme mimarisinin katmanlı yapısı ve özellik çıkarım süreci	26
Şekil 5.3 Aktivasyon fonksiyonları	27
Şekil 5.4 MNIST veri kümesinden el yazısı rakam örnekleri	30
Şekil 5.5 El yazısı rakam tanıma için iki gizli katmanlı sinir ağı mimarisi	30
Şekil 5.6 Gradyan iniş algoritmasının dağlık arazi benzetmesi	39
Şekil 5.7 Temel Tekrarlayan Sinir Ağı (RNN) mimarisi ve bilgi akışı	41
Şekil 5.8 Evrişimli sinir ağının işaret dili görüntüsünü sınıflandırma süreci	46
Şekil 5.9 Evrişimsel sinir ağının (CNN) temel yapısı	47
Şekil 5.10 Sinir ağında bir sonraki katmandaki alıcı nöron alanının gösterimi	51
Şekil 5.11 Evrişim filtrelerinin işaret dili görüntüsü üzerindeki etkisi	52
Şekil 5.12 2x2 pencere kullanan maksimum havuzlama örneği	55
Şekil 5.13 Çift iniş fenomeninin doğal spline örneği üzerinden gösterimi	61
Şekil 5.14 Seyreltme (Dropout) uygulaması: Tam bağlantılı ve seyreltilmiş ağ karşılaştırması	63
Şekil 5.15 Sigmoid fonksiyon grafiği	66
Şekil 5.16 Softmax fonksiyon grafiği	67
Şekil 5.17 Hiperbolik tanjant fonksiyon grafiği	68
Şekil 5.18 Doğrultulmuş lineer birim (ReLU) fonksiyonu ve türevi grafiği	69
Şekil 6.1 Mediapipe ellerin ve poz işaretlerinin gösterimi	80
Şekil 9.1 İkili sınıflandırma için karmaşıklık matrisi	86
Şekil 9.2 Kesinlik (Precision) hesaplamasının grafiksel gösterimi: TP ve FP ilişkisi	88
Şekil 9.3 Duyarlılık (Precision) hesaplamasının grafiksel gösterimi: TP ve FN ilişkisi	89
Şekil 9.4 Özgüllük (Specificity) hesaplamasının grafiksel gösterimi: TN ve FP ilişkisi	89
Şekil 10.1 ELAN dilbilimsel açıklama yazılımı arayüzü	93
Şekil 10.2 ASBTSL veri kümelerinden örnekler	96

Şekil 10.3 İşaret dil tanıma sisteminin çerçevesi	97
Şekil 10.4 “AİLE (Family)” Türk İşaret Dili	98
Şekil 10.5 “ANNE (Mother)” Türk İşaret Dili.....	98
Şekil 10.6 “ARABA (Car)” Türk İşaret Dili.....	98
Şekil 10.7 “ARKADAŞ (Friend)” Türk İşaret Dili.....	98
Şekil 10.8 “BABA (Father)” Türk İşaret Dili	99
Şekil 10.9 “CÜMLE (Sentence)” Türk İşaret Dili	99
Şekil 10.10 “DOKTOR (Doctor)” Türk İşaret Dili.....	99
Şekil 10.11 “EV (Home)” Türk İşaret Dili	99
Şekil 10.12 “İŞ (Work)” Türk İşaret Dili.....	100
Şekil 10.13 “İŞARET (Sign)” Türk İşaret Dili	100
Şekil 10.14 “KAFA (Head)” Türk İşaret Dili	100
Şekil 10.15 “ÖĞRETMEN (Teacher)” Türk İşaret Dili	100
Şekil 10.16 “OKUL (School)” Türk İşaret Dili	101
Şekil 10.17 “PARA (Money)” Türk İşaret Dili	101
Şekil 10.18 “PORBLEM (Problem)” Türk İşaret Dili.....	101
Şekil 10.19 “TELEFON (Phone)” Türk İşaret Dili.....	101
Şekil 10.20 “YARDIM (Help)” Türk İşaret Dili	102
Şekil 10.21 “YEMEK (Food)” Türk İşaret Dili	102
Şekil 10.22 “Avrupa (Europe)” ve “Türkiye” Türk İşaret Dili	102
Şekil 10.23 ASBTSL veri kümesinden alınan örnek bir çerçeve ve bu çerçeveden çıkarılan anahtar noktalar	104
Şekil 10.24 İşaret Dili Tanıma için Kullanılan İskelet Yapısı ve Anahtar Noktalar....	104
Şekil 10.25 Veri artırımı teknikleri ve örnekleri (ÖĞRETMEN).....	105
Şekil 10.26 İşaret dili algılama sistemimizin model eğitimine genel bir bakışı	112
Şekil 12.1 İşaret dili algılama sisteminin eğitim performansı: Doğruluk ve kayıp eğrileri (CNN model)	116
Şekil 12.2 İşaret dili algılama sisteminin eğitim performansı: Doğruluk ve kayıp eğrileri (LSTM model).....	117
Şekil 12.3 İşaret dili algılama sisteminin eğitim performansı: Doğruluk ve kayıp eğrileri (Hibrit model).....	117
Şekil 12.4 Test verileri üzerinde karışıklık matrisi (CNN Model)	122
Şekil 12.5 Test verileri üzerinde karışıklık matrisi (LSTM Model)	123

Şekil 12.6 Test verileri üzerinde karışıklık matrisi (Hibrit Model)	124
Şekil 13.2 a) “İŞARET” (Sign) b) “TERCÜMAN” (Interpreter)	127
Şekil 13.3 a) “İŞ” (Work) b) “ARABA” (Araba)	127
Şekil 13.4 a) “YEMEK” (Food) b) “OKUL” (School)	128
Şekil 13.5 Eğitilmiş işaret dili algılama modelinin gerçek zamanlı tahmin örnekleri ..	128



ÇİZELGELER DİZİNİ

Çizelge 10.1 İşaret dili tanıma veri kümelerinin istatistikleri	95
Çizelge 10.2 ASBTSL veri kümelerinin istatistikleri	97
Çizelge 12.1 CNN model performans metrikleri	119
Çizelge 14.1 Model performans metrikleri	129



1. GİRİŞ

1.1 Genel Bakışı

İşaret Dili (SL), konuşma ve işitme engelli bireyler için hayati önem taşıyan temel bir iletişim aracıdır. El hareketlerine dayalı bu sözel olmayan iletişim yöntemi, dünya genelinde milyonlarca insan tarafından kullanılmaktadır (Al-Obaidi vd. 2023). Yaygın kanının aksine, her ulusun kendine özgü bir işaret dili bulunmaktadır ve bu diller oldukça zengin bir kelime dağarcığına sahiptir. Dünya Sağlık Örgütü (WHO) verilerine göre, günümüzde dünya genelinde 430 milyondan fazla kişi (432 milyon yetişkin ve 34 milyon çocuk) önemli işitme kaybı sorunları yaşamaktadır. Bu, küresel nüfusun %5'inden fazlasının rehabilitasyon ihtiyacı duyduğu anlamına gelmektedir. Dahası, bu sayının 2050 yılına kadar 700 milyonu aşacağı ve dünya genelinde her 10 kişiden 1'inin engelli işitme kaybı yaşayacağı öngörülmektedir (WHO 2023).



Şekil 1.1 Türk işaret dilinde “MERHABA” işareti

Sağır ve işitme engelli topluluklar, günlük yaşam aktivitelerinde, eğitim ve sağlık alanlarında önemli dezavantajlarla karşılaşmaktadır. Kamu hizmeti sunan kurum ve kuruluşlarda işaret dili tercümanı istihdamı, sürdürülmesi maliyetli bir yatırım olarak görülmektedir. Bu soruna uygulanabilir ve uygun maliyetli bir çözüm olarak, işaret dilleri için otomatik tanıma ve çeviri sistemleri öne çıkmaktadır. İşaret Dili Tanıma (SLR), Sağır bireylerin temel iletişim aracı olan işaret dilini hesaplamalı olarak anlamayı hedefleyen bir araştırma alanıdır. Bu alan, sosyal etkileri ve içerdiği zorluklar nedeniyle bilgisayarla

görme topluluğunun büyük ilgisini çekmektedir. SLR'nin genel amacı, uzun vadede işiten ve Sağır bireyler arasındaki iletişim boşluğunu dolduracak öğrenen bir bilgisayarlı görme sistemi geliştirmektir (Hezhen Hu vd. 2021; Camgoz vd. 2020; Cooper ve Bowden 2007). Ancak, SLR alanı çeşitli zorluklarla karşı karşıyadır. İşaret dillerinin sürekli değişen doğası ve çok boyutlu yapısı, bu zorlukların başında gelmektedir. Bir işaret, el şekli, gövdeye göre el oryantasyonu, el hareketi, vücuda göre el konumu ve manuel olmayan özellikler (vücut duruşu, yüz ifadeleri vb.) gibi beş temel bileşenden oluşmaktadır. Bu karmaşık yapı, SLR sistemlerinin geliştirilmesini zorlaştırmaktadır (Power vd. 2020).

Bu tez çalışması, İşaret Dili Tanımda Derin Öğrenme ile İstatistiksel Yöntemlerin Karşılaştırılması'nı ele almaktadır. Çalışmamız, bu iki yaklaşımın güçlü ve zayıf yönlerini ortaya koyarak, Türk İşaret Dili tanıma alanına katkıda bulunmayı hedeflemektedir. İlerleyen bölümlerde, kullanılan yöntemler detaylı olarak açıklanacak, elde edilen bulgular sunulacak ve sonuçlar tartışılacaktır. İşaret dili tanıma sistemleri (SLR), işitme engelli bireylerin topluma entegrasyonunu iyileştirmek için büyük bir potansiyele sahiptir. Dünya İşitme Engelliler Federasyonu'na göre, dünya genelinde 70 milyon işitme engelli birey 200'den fazla farklı işaret dili kullanmaktadır (Manning vd. 2022). Bu çeşitlilik, SLR sistemlerinin geliştirilmesinde önemli zorluklar yaratmaktadır. SLR araştırmalarının büyük çoğunluğu manuel özelliklere odaklanmıştır. Bunlar arasında el hareketleri, ellerin yönü ve şekli ile parmak yazımı yer almaktadır. Ancak, işaret dili sadece el hareketlerinden ibaret değildir. Baş ve gövde hareketleri, yüz ifadeleri ve ağız hareketleri gibi manuel olmayan özellikler de önemli bilgiler taşımaktadır. Özellikle yüz ifadeleri ve baş hareketleri, soru, olumsuzlama, konu, iddia, şüphe ve koşul gibi dilbilgisel işaretlerde kritik rol oynamaktadır (Akyol ve Alvarado 2001; Huang vd. 2018; Shi vd. 2018).

Türk İşaret Dili, Amerikan ve İngiliz İşaret Dili'nden farklı olarak Türkiye'ye özgü bir dildir. Bu özgünlük, Türk İşaret Dili için SLR sistemleri geliştirirken dikkate alınması gereken önemli bir faktördür. Ancak, manuel olmayan işaret tanıma konusundaki araştırmalar sınırlı kalmıştır ve etiketli manuel olmayan veri kümelerinin eksikliği, bu alandaki çalışmaları zorlaştırmaktadır. İşaret dilleri, tıpkı konuşma dilleri gibi, sürekli bir gelişim ve değişim içindedir (Wilson 1985). İşaret dilleri, ülkeler arasında önemli

farklılıklar göstermektedir. İlginç bir şekilde, aynı konuşma dilini kullanan ülkeler bile farklı işaret dillerini benimseyebilmektedir. Örneğin, Amerikan İşaret Dili (ASL) ile İngiliz İşaret Dili (BSL) arasında önemli farklılıklar bulunmaktadır (Jachova vd. 2008). Bu çeşitlilik, evrensel bir işaret dili tanıma sistemi geliştirmeyi zorlaştırmaktadır. Bu dinamik yapı, işaret dili tanıma sistemlerinin tasarımında önemli bir faktördür. İlginç bir şekilde, bazı teoriler insanların sesli iletişime geçmeden önce işaret dilini kullandıklarını öne sürmektedir (Laitman ve Heimbuch 1982). Bu durum, işaret dilinin insanlık tarihi kadar eski olabileceğini göstermektedir. İşaret dili tanıma çalışmalarının tarihçesi oldukça eskiye dayanmaktadır. 1914 gibi erken bir tarihte bile, işitme engellilere konuşma öğretilmesi üzerine çalışmalar yapıldığı bilinmektedir (Young 1914). Ancak, bu yaklaşımın işitme engelli bireylerin iletişim sorunlarını tam olarak çözemediği anlaşılmıştır. Bilgisayar destekli işaret dili tanıma çalışmaları ise özellikle 1970'lerden itibaren hız kazanmıştır.

Son yıllarda, güçlü GPU'ların erişilebilirliğinin artması, bilgisayarla görme görevlerinde derin öğrenme tekniklerinin popülerliğini artırmıştır. Özellikle Evrimsel Sinir Ağları (CNN), hem görüntü hem de video tanıma görevlerinde başarılı sonuçlar vermiştir (Karpathy, A. ve ark. 2014). Ancak, işaret dili videolarının uzamsal-zamansal özelliklerini en iyi şekilde temsil edecek modeli tasarlamak, SLR literatüründe hala önemli bir araştırma konusudur. İşaret dili tanıma teknolojilerinin gelişimi, farklı aşamalardan geçmiştir. İlk dönemlerde sensörlü eldivenler kullanılmış (Karpathy vd. 2014), ancak bu teknolojinin pratik uygulaması 1997'ye kadar gerçekleşmemiştir (Kadous vd. 1996). 2000'li yılların başında, Saklı Markov Modelleri gibi yapay zeka teknikleri işaret dili tanımada kullanılmaya başlanmıştır. Örneğin, sensörlü ve kameralı eldivenler İngilizce ve Çince harfleri tanımak için kullanılmıştır (Wang vd. 2000; Pashaloudi ve Margaritis 2002). Günümüzde, derin öğrenme alanındaki ilerlemeler sayesinde, ek ekipman kullanmadan büyük veri kümeleriyle çalışmak mümkün hale gelmiş ve daha yüksek tanıma performansları elde edilmiştir. Bu gelişmeler, işaret dili tanıma sistemlerinin daha etkili ve kullanışlı hale gelmesini sağlamıştır.

İşaret dili jestleri, manuel ve manuel olmayan olmak üzere iki ana kategoriye ayrılmaktadır. Manuel jestler ellerin hareketini içerirken, manuel olmayan jestler ağız

hareketi, baş sallama, kaş hareketleri, vücut eğilmesi ve diğer yüz ifadelerini kapsamaktadır. Ayrıca, işaretler statik ve dinamik olarak da sınıflandırılabilir. Hareket içeren işaretler dinamik jestler olarak adlandırılmaktadır (Farooq vd. 2021). Konuşma dilindeki bir kelimenin işaret dilindeki karşılığı ise "gross" veya "işaret kelimesi" olarak ifade edilmektedir (Belissen vd. 2020). Çalışmamız, işaret dilinin bu çok boyutlu yapısını göz önünde bulundurarak, farklı yaklaşımların etkinliğini değerlendirmeyi amaçlamaktadır.

İşaret Dili Tanıma (SLR) sistemleri, birçok zorlu görevle karşı karşıyadır. Özellikle parmakla yazım işaretlerini tanıma, işaretçinin akıcılığındaki yüksek değişkenlik ve birlikte eklemleme gibi dilbilimsel etkiler nedeniyle oldukça karmaşıktır. SLR sistemlerinin, yüksek doğruluk ve düşük yanlış sınıflandırma oranlarını korurken, aynı zamanda yüksek derecede bozulma değişmezliği göstermesi gerekmektedir. İşaret dilleri, ellerin, yüzün ve vücudun hareketiyle üretilen görsel dillerdir. Bu nedenle, mevcut son teknoloji işaret dili işleme sistemleri, işaret dili tanıma (SLR) ve işaret dili çevirisi (SLT) görevlerini modellemek için video verilerine dayanmaktadır. Ancak, ham video sinyali kullanmak hesaplama açısından zahmetli olabilmekte ve aşırı uyuma (overfitting) ve kişiye bağımlılık gibi sorunlara yol açabilmektedir. Bilgisayarlı görü ve makine öğrenimi tekniklerinin SLR'ye uygulanmasında karşılaşılan temel zorluklardan biri, yeterli ölçek ve formatta veri kümelerinin ve derlem içeriğinin azlığıdır (Bragg vd., 2019). Bu durum kısmen işaret dillerinin yazılı formlarının olmamasından ve ek açıklama yapmanın maliyetli olmasından kaynaklanmaktadır (Leeson vd., 2006). Evrimsel Sinir Ağları (CNN'ler), görüntülerdeki özellikleri çıkarmada etkili olduğu kanıtlanmış bir yöntemdir (Lecun vd., 1998). Ancak, CNN'ler aşırı uyumdan kaçınmak için genellikle büyük veri kümelerine ihtiyaç duymaktadır. Aşırı uyumun etkisi, modelin görülmeyen test verilerine uygulandığında performansının düşmesi şeklinde ortaya çıkmaktadır.

İşaret ve duruş tanıma için, istatistiksel örüntü tanıma ve diğer sığ öğrenme tekniklerine dayanan birçok klasik yaklaşım da kullanılmıştır (Omar vd., 2019). Örneğin, Farouk vd. (2013), İrlanda İşaret Dili (ISL) parmak heceleme el şekillerinin sentetik görüntülerini kullanarak Temel Bileşen Analizine (PCA) dayalı bir tanıma modeli oluşturmuştur.

Bu tez çalışmasında, Türk İşaret Dili tanıma için derin öğrenme ve istatistiksel yöntemlerin karşılaştırmalı analizi yapılmıştır. Çalışmamız, yukarıda bahsedilen zorlukları göz önünde bulundurarak hem derin öğrenme hem de istatistiksel yaklaşımların güçlü ve zayıf yönlerini ortaya koymayı amaçlamaktadır. Özellikle, veri kısıtlılığı, model genelleştirme yeteneği ve hesaplama verimliliği gibi faktörler üzerinde durulmuştur. Bu karşılaştırmalı analiz, Türk İşaret Dili tanıma alanında daha etkili ve verimli sistemlerin geliştirilmesine katkıda bulunmayı hedeflemektedir.

1.2 İlgili Çalışmalar

İşaret Dili Tanıma (SLR) alanında yapılan çalışmalar, farklı yaklaşımların etkinliğini ve karşılaşılan zorlukları ortaya koymaktadır. Bu bölümde, özellikle iskelet tabanlı modeller ve derin öğrenme yaklaşımları üzerine yapılan önemli çalışmalar incelenecektir.

Camgöz ve arkadaşları (2017), iskelet modelinin ön eğitiminin yapılmamasının, nihai sonuçlar üzerindeki etkisini araştırmışlardır. Çalışmalarında, iskelet temsiliinin önemli ölçüde bilgi kaybına yol açtığını ve bunun sonucunda potansiyel uçtan uca yaklaşımlardan daha düşük performans elde edildiğini gözlemlemişlerdir. Bu bulgu, iskelet tabanlı modellerin SLR için kullanımında dikkat edilmesi gereken önemli bir noktayı vurgulamaktadır. Aynı araştırma ekibi, bu durumun nedenlerini daha iyi anlamak için bir dizi çalışma gerçekleştirmiştir. Camgöz ve arkadaşları (2018), farklı yaklaşımların değerlendirilmesine odaklanmışlardır. Bu çalışma, çeşitli SLR yöntemlerinin güçlü ve zayıf yönlerini ortaya koyarak, alandaki araştırmacılara değerli bilgiler sunmuştur. Devam eden çalışmalarında Camgöz ve arkadaşları (2020a), SLR sistemlerindeki başarısızlık durumlarını kategorize etmişlerdir. Bu çalışma, SLR sistemlerinin karşılaştığı zorlukları sistematik bir şekilde analiz ederek, gelecekteki araştırmalar için önemli bir rehber oluşturmuştur. Son olarak, Camgöz ve arkadaşları (2020b), bir poz tahmin sisteminin SLR için uygulanabilir olması için sahip olması gereken nitelikleri karakterize etmişlerdir. Bu çalışma, etkili bir SLR sistemi geliştirmek için gerekli olan temel özellikleri belirlemiş ve gelecekteki araştırmalar için bir çerçeve sunmuştur.

Bu çalışmalar, iskelet tabanlı modellerin SLR'deki potansiyelini ve sınırlamalarını ortaya koymaktadır. İskelet temsili bilgi kaybına yol açması, bu yaklaşımın tek başına yeterli olmayabileceğini göstermektedir. Bu nedenle, derin öğrenme ve istatistiksel yöntemlerin kombinasyonu, daha etkili SLR sistemleri geliştirmek için umut verici bir yön olarak görülmektedir.

1.3 İşaret Dili Tanıma'nın Zorlukları ve Mevcut Durum

İşaret Dili Tanıma (SLR) alanında kaydedilen ilerlemelere rağmen, bu teknolojinin ticari uygulamalarda etkin bir şekilde kullanılması hala önemli zorluklarla karşı karşıyadır. Zeledon ve arkadaşlarının (2019) yaptığı sistematik inceleme, İşaret Dili çevirisi için ticari uygulamaların düşük kullanılabilirliğini vurgulamaktadır. Bu uygulamaların çoğunlukla tanıma yerine sentezle sınırlı olduğu ve belirli alanlarla kısıtlandığı görülmektedir. Rastgoo ve arkadaşları (2021), mevcut en son teknolojinin daha karmaşık modeller gerektiren sözlüksel ve anlamsal alanlardan ziyade fonetik işaret tanıma üzerine odaklandığını belirtmektedir. Bu durum, İşaret Dili'nin daha derin ve karmaşık yönlerinin henüz tam olarak ele alınmadığını göstermektedir.

İşaret Dili Tanıma'nın karşılaştığı zorluklar arasında şunlar sayılabilir:

1. İşaretlemenin karmaşık fonolojisi
2. İşaretçinin akıcılığı, el becerisi, yaşı ve cinsiyetindeki farklılıklar
3. Argo ve lehçe kullanımı
4. Oklüzyon (bir nesnenin başka bir nesne tarafından kısmen gizlenmesi) sorunları
5. Kamera kalitesi ve çekim açısı farklılıkları

Bu faktörler, sürekli işaret tanımayı çeviri sürecinin en zorlu kısmı haline getirmektedir. Bragg ve arkadaşları (2019), bu zorlukların üstesinden gelmek ve daha etkili SLR sistemleri geliştirmek için önemli bir öneri sunmaktadır. Araştırmacılar, "topluluğu doğru bir şekilde temsil etmek, anlamlı sorunları ele almak ve kültürel tahsisattan kaçınmak" için Sağır katılımcıların araştırma ve geliştirmenin tüm aşamalarına dahil edilmesi gerektiğini savunmaktadır. Bu yaklaşım, SLR sistemlerinin gerçek dünya koşullarında

daha etkili ve kapsayıcı olmasını sağlayabilir. Mevcut durum ve karşılaşılan zorluklar göz önüne alındığında, İşaret Dili Tanıma alanında daha fazla araştırma ve geliştirme çalışmasına ihtiyaç duyulduğu açıktır.

1.4 Türk İşaret Dili (TİD)

Türk İşaret Dili (TİD), Türkiye'deki işitme engelli bireylerin temel iletişim aracıdır. Ancak, TİD'nin gelişimi ve anlaşılması, standartlaştırılmış sözlükler ve dilbilimsel veri tabanları gibi temel kaynakların yetersizliği nedeniyle sınırlanmaktadır. Bu durum, TİD'nin özgün ve bağımsız bir dil olarak gelişimini engellemektedir. Özellikle, birleştirilmiş sözlüklerin bulunmaması, Sağır bireyler ve araştırmacılar için önemli bir engel teşkil etmekte ve TİD işaretleri için kapsamlı ve evrensel kabul görmüş bir kaynağa erişimi kısıtlamaktadır.

TİD, Amerikan İşaret Dili (ASL) gibi yeterli araştırmaya sahip değildir ve özellikle Derin Öğrenme (DL) ve Makine Öğrenimi (ML) tekniklerinin uygulanması açısından henüz gelişiminin erken aşamalarında. Bununla birlikte, modern teknolojik gelişmeler, Sağır bireylerin birçok iletişim zorluğunu aşmalarına yardımcı olmaktadır. Derin öğrenme ve bilgisayarla görme modellerindeki son atılımlar, insan etkileşiminde devrim yaratan benzersiz yetenekler sunmakta ve Türk İşaret Dili araştırmaları için verimli bir alan olmaya devam etmektedir. Etkili iletişim, bireylerin çevrelerindeki topluluklarla etkileşim kurmaları için temel bir beceridir. İletişim aynı zamanda bireyin zihinsel sağlığı üzerinde de önemli bir rol oynamaktadır (Martin ve Chanda 2016). İşitme engelli bireyler, toplumla etkileşimde bulunmanın temel yolu olan işaret dili becerilerini edinme ihtiyacı duyduklarından, başkalarıyla iletişim kurmada zorluklarla karşılaşabilmektedirler. Bu soruna yönelik görüntü tabanlı çözümlerin etkinliği, büyük ölçüde segmentasyon kalitesine ve işaret dili hareketlerinin kritik görsel özelliklerini doğru bir şekilde temsil eden özelliklerin seçimine bağlıdır (AlKhuraym vd. 2022). Bu nedenle, TİD için etkili bir İşaret Dili Tanıma (SLR) sistemi geliştirmek, hem teknik hem de dilbilimsel açıdan karmaşık bir görevdir. TİD'de, diğer işaret dillerinde olduğu gibi, işaretler genellikle tamamen büyük harflerle yazılan herhangi bir konuşma dilinin işaret-sözcüklerine dönüştürülebilmektedir. Bu tür her bir işaret-sözcük teknik olarak "gloss" olarak

adlandırılmakta ve bir İşaret Dili (SL) transkriptinin temel atomik belirtici olarak kabul edilmektedir. Bu özellik, TİD'nin bilgisayarlı sistemler tarafından işlenmesi ve anlaşılması için önemli bir temel oluşturmaktadır.

Bu tez çalışması kapsamında, TİD için derin öğrenme tabanlı gerçek zamanlı tanıma modellerinin geliştirilmesi ve işitme engelli bireylere yardımcı olacak bir SLR sistemi uygulaması ele alınacaktır. Bu çalışma, TİD'nin daha iyi anlaşılmasına katkıda bulunmayı ve işitme engelli ve işiten bireyler arasındaki iletişim uçurumunu kapatmaya yardımcı olmayı hedeflemektedir.

1.5 İşaret Dili Tanıma

İşaret Dili Tanıma (SLR), Sağır ve işiten topluluklar arasındaki iletişim engellerini ortadan kaldırmayı hedefleyen bir araştırma alanıdır. SLR sistemleri, el hareketleri ve yüz ifadelerinin bir kombinasyonundan oluşan İşaret Dili'ni (SL) anlamaya ve yorumlamaya çalışır (Gökgöz 2018; Rastgoo vd. 2021). Bu sistemler, Türk SL, Amerikan SL, İngiliz SL ve Çin SL gibi farklı işaret dillerinde, vücudun farklı bölümlerinin (eller, baş, yüz, kollar, gövde) işaretlerin görsel özelliklerine katkısını dikkate alır (Benitez-Quiroz vd. 2014; Cheok vd. 2019). SLR'nin temel problemi, işaretçinin kimliği, duruşu veya arka plandaki detaylardan ziyade, ayırt edici bir işaret temsilinin öğrenilmesidir. Bir SLR çerçevesinin girdisi, genellikle bir işaretçinin bir veya birden fazla işaret gerçekleştirdiği bir videodur. Bu işaretçiler, anadili işaret dili olan Sağır bireyler veya işaret dili bilen işiten bireyler olabilir.

SLR çerçeveleri, işaretleri ayırt etmeyi kolaylaştıran uzamsal ve zamansal işaret temsillerini öğrenmeyi amaçlar. Ancak, Evrişimsel Sinir Ağı (CNN) tabanlı tanıma çerçevelerinin işaretçiye bağımlı olma ve işaretçilerin yüz özellikleri, vücut şekli ve ten rengi gibi özelliklerini yakalama eğiliminde olduğu gözlemlenmiştir. Bu durum, özellikle doğal ortamlarda toplanan işaret dili veri kümelerinde, işaretçi kimliği özneliliklerindeki çeşitlilik ve farklı arka planlar nedeniyle daha da önemli hale gelmektedir (Ferreira vd. 2021).

İşaret dili tanıma, genellikle iki farklı yaklaşımla ele alınır:

1. Ham video girdisi üzerinde bilgisayarla görme modellerinin kullanılması.
2. Videonun bir poz tahmin aracı ile ön işlemden geçirildiği poz verilerinin kullanılması.

SLR modelleri genel olarak iki temel kategoriye ayrılır (Mustafa 2021):

1. Görüş tabanlı teknikler: Bu yaklaşımda, sistem girdisi olarak video akışları veya kamera aracılığıyla elde edilen fotoğraflar kullanılır.
2. Sensör eldiveni tabanlı yaklaşımlar: Bu teknik, sensör cihazları ile kullanıcılar arasında doğrudan fiziksel etkileşim gerektirir. Genellikle elektromiyografi, atalet ölçümleri veya elektromanyetik sensörler gibi çeşitli sensörlere sahip eldivenler kullanılır. Bu sensörler, SL hareketlerinin gerçekleştirilmesi sırasında parmakların pozisyonu, fleksiyonu, yönü ve açıları gibi verileri toplar.

İşaret Dili Tanıma (SLR) alanında, görüş tabanlı yöntemler yaygın olarak tercih edilmektedir. Bu tercihin temel nedeni, tanıma sürecinde kamera dışında özel eldivenlere veya ek donanım ihtiyacı duyulmamasıdır (Mustafa 2021). Ancak, bu yaklaşımlar aydınlatma koşulları, gürültü, perspektif farklılıkları ve karmaşık arka planlar gibi faktörlerden etkilenebilmektedir, bu da sonuçların doğruluğunu etkileyebilmektedir (Mustafa 2021). El hareketi tanıma, robotik, oyun, sanal gerçeklik, SL ve insan-bilgisayar etkileşimi gibi geniş bir uygulama yelpazesine sahip olması nedeniyle birçok araştırmacının ilgisini çekmektedir (Aly ve Aly 2020). İşaret Dili, temel olarak manuel hareketlere dayanan ve işitme engelli bireyler için en etkili iletişim yöntemi olarak kabul edilmektedir. Dinamik, izole SL hareketlerinin doğru bir şekilde tanınması, el segmentasyonu, el şekli bilgisinin kodlanması ve jest dizilerinin tanınması gibi üç temel zorluğun üstesinden gelinmesini gerektirmektedir (Aly ve Aly 2020).

İşaret Dili'nde jestler, statik ve dinamik olmak üzere iki ana kategoriye ayrılmaktadır:

1. Statik Jestler: Ellerin belirli pozisyonlarda ve hizalamalarda sabitlendiği jestlerdir. Bu jestlerde bedensel hareket yer almaz ve yorumlanmalarında açısız oryantasyon

ve parmak şekli önemlidir (Pisharady ve Saerbeck 2015; Al-Shamayleh vd. 2018). Statik jestler; yönelimler, parmak şekli, esneme açıları, vücuda göre göreceli konum ve bağlamsal çevre gibi özellikler ile karakterize edilebilir (Kausar ve Javed 2011).

2. Dinamik Jestler: Belirli bir alan ve zaman içinde el hareketlerini içeren jestlerdir. Bu jestlerde el hareketi sürekli ve vuruş, geri çekme ve hazırlık gibi farklı aşamalar dikkate alınır (Pisharady ve Saerbeck 2015; Al-Shamayleh vd. 2018). Dinamik jestler; yönelimler, parmak konfigürasyonu, bükülme açıları, el yolları, el yönelimleri, hareket hızı ve yönü ve ölçek gibi ayırt edici nitelikler sergiler (Kausar ve Javed 2011).



Şekil 1.2 Statik ve dinamik hareket örneği (“ANNE” işaret dili)

Türk İşaret Dili'nde (TİD) jestlerin doğru bir şekilde tanınması dört aşamada gerçekleştirilmektedir (Ahmed vd. 2016):

1. Veri Toplama: El görüntüsü uygun bir giriş cihazı ile yakalanır.
2. Ön İşleme ve Segmentasyon: Görüntü arka plandan ayrıştırılır ve elin konumu belirlenir. Gürültü azaltma, kontur tanımlama ve model oluşturma gibi işlemler gerçekleştirilir.
3. Özellik Çıkarma: Sınıflandırma için şekil, konum, yönelim, hareketler ve el yerleşimleri gibi veriler çıkarılır.
4. Sınıflandırma: Elde edilen görüntüler uygun hareketler olarak sınıflandırılır.

Bu aşamalar, hem geleneksel SLR hem de Otomatik İşaret Dili Tanıma (ASLR) modellerinin temel bileşenleridir. SLR, işaret dillerinin, izole kelimelerin ve sürekli cümlelerin tanınması olmak üzere üç farklı seviyede sınıflandırılabilir (Ahmed vd. 2016). İşaret Dili Tanıma (SLR) sistemleri, izole ve sürekli olmak üzere iki temel sınıfa ayrılmaktadır. İzole SLR, her bir görüntüyü alfabenin benzersiz bir harfi, sayısal bir değer veya farklı bir hareketin temsili olarak yorumlarken, sürekli SLR kesintisiz işaretlemeyi tanımlayarak tüm cümleleri doğru bir şekilde tanımlama ve çevirme yeteneğine sahiptir (Madhushree vd. 2023). SLR sınıflandırması için Gizli Markov Modelleri (HMM), Evrişimli Sinir Ağları (CNN) ve Yapay Sinir Ağları (YSA) gibi çeşitli yöntemler kullanılmaktadır.

İşaret dili tanıma, doğal dil işleme, dilbilim, örüntü eşleştirme, bilgisayarla görme ve makine öğrenimini içeren çok disiplinli bir araştırma alanıdır (Cooper vd. 2011). SLR'nin nihai amacı, önceden üretilmiş işaretleri tanımlayabilen, anlamlarını çözebilen ve hedeflenen alıcının anlayabileceği başka bir dilde çıktı üretebilen bir sistem geliştirmektir.

Genel SLR problemi üç farklı görevi içermektedir:

1. Statik veya sürekli harf/rakam hareketi tanıma (sınıflandırma problemi)
2. Statik veya sürekli tek kelime tanıma (sınıflandırma problemi)
3. Cümle düzeyinde işaret dili tanıma (NLP problemi)

Mevcut literatür taramaları, bazı özel koşullar sağlandığında, son araştırmaların ilk iki görev için %80-100 aralığında bir doğruluk elde ettiğini bildirmektedir (Cheok vd. 2019; Wadhawan ve Kumar 2019; Elakkiya 2020; Rastgoo vd. 2021). İşaret dili alfabesini tanımak için hem bilgisayar görüşü tabanlı hem de sensör tabanlı yaklaşımlar uygulanmaktadır. Görme tabanlı yaklaşımlar, el ile gerçekleştirilen işaretlerin görüntüsünü veya video karelerini yakalamak için RGB kameraları veya derinlik kameralarını (Leap Motion denetleyicisi veya Kinect gibi) kullanmaktadır (Chong ve Lee 2018; Rastgoo vd. 2018; Shi vd. 2018; Tao vd. 2018; Jiang ve Zhang 2019; Warchoř vd. 2019; Barbhuiya vd. 2021; Lee vd. 2021; Luqman vd. 2021; Sharma ve Kumar 2021). Bu yaklaşımlarda, el özelliklerinin çıkarılması büyük bir zorluk teşkil etmekte ve aydınlatma

durumu, karmaşık arka planlar, oklüzyon ve ten rengi gibi birçok faktörden etkilenmektedir.

Bu tez çalışmasında, Türk İşaret Dili'nin (TİD) tanınması için görüş tabanlı teknikler kullanılarak derin öğrenme ve istatistiksel yöntemlerin etkinliği karşılaştırılacaktır. Çalışma, özellikle veri toplama, özellik çıkarma ve sınıflandırma aşamalarında farklı yaklaşımların performansını değerlendirmeyi amaçlamaktadır.

1.6 Tezin Amacı

Bu tezin temel amacı, işaret dili tanıma (SLR) alanında derin öğrenme yöntemleri ile geleneksel istatistiksel yöntemlerin karşılaştırmalı bir analizini gerçekleştirmektir. Çalışma, özellikle Türk İşaret Dili (TİD) verilerinin sınıflandırılmasında kullanılan çeşitli yaklaşımların etkinliğini değerlendirmeyi hedeflemektedir.

Araştırma kapsamında, derin öğrenme modelleri (CNN-LSTM) ve istatistiksel yöntemler Python programlama dili kullanılarak uygulanmış ve sonuçları detaylı bir şekilde incelenmiştir. Bu çalışma, SLR alanında karşılaşılan zorlukları ele alırken, her iki yaklaşımın güçlü ve zayıf yönlerini ortaya koymayı amaçlamaktadır.

Bu tez çalışması, işitme engelli bireylerin iletişimini kolaylaştırmak için kritik öneme sahip olan SLR alanına önemli katkılar sağlamayı hedeflemektedir. Derin öğrenme ve istatistiksel yöntemlerin güçlü yönlerini birleştiren hibrit modeller önerilerek, SLR sistemlerinin performansının artırılması amaçlanmaktadır. Ayrıca, bu karşılaştırmalı analiz, TİD özelinde SLR araştırmalarına yeni bir perspektif sunarak, bu alandaki gelecek çalışmalara rehberlik edecektir.

2. YAPAY ZEKA

Yapay zeka, insan zekasını taklit eden ve giderek daha karmaşık görevleri yerine getirebilen sistemlerin geliştirilmesini amaçlayan bir bilim dalıdır. Son yıllarda, özellikle bilgisayarlı görü ve makine öğrenimi alanlarındaki ilerlemeler, işaret dili tanıma (SLR) teknolojilerinde çığır açıcı gelişmelere yol açmıştır. Bu gelişmelerin temelinde, yüksek performanslı grafik işlem birimlerinin (GPU) yaygınlaşması yatmaktadır. GPU'lar, karmaşık görüntü işleme ve derin öğrenme algoritmalarının çok daha hızlı çalıştırılmasını mümkün kılmıştır. Bu teknolojik ilerleme, daha önce aylarca sürebilen işaret dili tanıma modellerinin eğitim sürecini günler mertebesine indirmiştir. Günümüzde, büyük ölçekli ve çeşitli işaret dili veri setlerinin mevcudiyeti, derin öğrenme tekniklerinin SLR alanında yaygın kullanımını ve hızlı gelişimini desteklemektedir. Bu veri zenginliği, daha doğru ve güvenilir tanıma sistemlerinin geliştirilmesine olanak sağlamıştır.

İşaret dili tanıma alanında kullanılan başlıca yapay zeka yöntemleri şunlardır:

1. Derin Öğrenme Modelleri:

- a. Evrişimli Sinir Ağları (CNN): Görüntü işleme ve özellik çıkarma konusunda etkilidir.
- b. Uzun-Kısa Süreli Bellek Ağları (LSTM): Zamansal bilgiyi işlemede başarılıdır.
- c. Tekrarlayan Sinir Ağları (RNN): Sıralı veri analizinde kullanılır.

2. Bu modeller, el hareketleri, yüz ifadeleri ve vücut duruşu gibi karmaşık ve dinamik özellikleri etkili bir şekilde öğrenebilme yeteneğine sahiptir.

3. Geleneksel İstatistiksel Yöntemler:

- a. Gizli Markov Modelleri (HMM): Zamansal veri modellemede kullanılır.
- b. Destek Vektör Makineleri (SVM): Sınıflandırma problemlerinde etkilidir.
- c. Dinamik Zaman Bükme (DTW): Zaman serisi analizinde kullanılır.

Bu yöntemler, özellikle sınırlı veri setleriyle çalışırken veya hesaplama kaynakları kısıtlı olduğunda tercih edilebilmektedir.

İşaret dili tanıma sistemlerinin geliştirilmesinde, görüntü ön işleme, özellik çıkarma ve sınıflandırma aşamalarında çeşitli optimizasyon algoritmaları kullanılmaktadır. Bu algoritmalar, modellerin performansını artırırken, eğitim süresini kısaltmaya ve genelleme yeteneğini iyileştirmeye yardımcı olmaktadır. Son dönemde, derin öğrenme ve istatistiksel yöntemlerin güçlü yönlerini birleştiren hibrit yaklaşımlar da öne çıkmaktadır. Bu hibrit modeller, derin öğrenmenin özellik çıkarma gücü ile istatistiksel yöntemlerin yorumlanabilirlik ve veri verimliliği avantajlarını bir araya getirmeyi amaçlamaktadır. Yapay zeka teknolojilerinin SLR alanındaki uygulamaları, işitme engelli bireylerin iletişim olanaklarını genişletme potansiyeline sahiptir. Ancak, bu teknolojilerin etik ve sorumlu bir şekilde geliştirilmesi ve uygulanması önemlidir. Gizlilik, veri güvenliği ve algoritmaların adil olması gibi konular, SLR sistemlerinin tasarımında dikkate alınması gereken önemli faktörlerdir.

2.1 İşaret Diliyle Yapay Zeka Tabanlı Erişilebilirlik ve Teknoloji Entegrasyonu

Yapay zeka ve işaret dili teknolojilerinin gelişimi, Sağır ve işitme engelli bireylerin iletişim olanaklarını genişletmekte önemli bir rol oynamaktadır. Aashaka Desai'nin araştırmaları ve gözlemleri, bu alandaki mevcut durumu ve gelecek perspektiflerini kapsamlı bir şekilde ortaya koymaktadır. İşaret dilinin teknolojik entegrasyonunda, özellikle otomatik tanıma ve çeviri sistemlerinin geliştirilmesinde önemli ilerlemeler kaydedilmiştir. Bu sistemler, ASL Citizen, Semlex ve Popsign gibi yeni veri setlerinin oluşturulmasıyla desteklenmektedir. Bu veri setleri, Sağır toplulukla işbirliği içinde, etik ve adil bir yaklaşımla geliştirilmektedir. Araştırmalar, Sağır ve işitme engelli bireylerin iletişim repertuarlarının sanıldığından çok daha zengin ve karmaşık olduğunu göstermektedir. Bu bireyler, işaretlerden konuşmaya, yazıdan dokunsal iletişime kadar uzanan çok modlu ve çok dilli bir iletişim yelpazesine sahiptir. Teknolojik çözümler geliştirilirken bu çeşitliliğin dikkate alınması kritik önem taşımaktadır (Desai vd.2024).

Mevcut teknolojik çalışmalarda üç temel alan öne çıkmaktadır:

1. İşaret dili tanıma sistemleri
2. İşaret dili üretim teknolojileri
3. Otomatik çeviri sistemleri

Sistemik önyargılar ve sınırlamalar konusunda dikkat edilmesi gereken noktalar bulunmaktadır. Özellikle veri setlerinin toplanmasında ve işlenmesinde, tüm işaret dili kullanıcılarının çeşitliliğini yansıtacak kapsayıcı bir yaklaşım benimsenmesi gerekmektedir. Araştırmalar göstermektedir ki, teknolojik çözümler insan tercümanların yerini almak yerine, onları destekleyici bir rol üstlenmelidir.

Erişilebilirlik teknolojisinin geliştirilmesinde iki temel hedef bulunmaktadır:

1. Teknolojinin kendisini erişilebilir kılmak
2. Teknolojiyi kullanarak dünyayı daha erişilebilir hale getirmek

Dil akıcılığı ve kullanımı konusunda önemli bulgular ortaya çıkmıştır. Akıcılığın ikili bir durum olmadığı, sürekli değişen ve gelişen bir süreç olduğu anlaşılmıştır. Bu nedenle, teknolojik çözümler bu dinamik yapıyı destekleyecek şekilde tasarlanmalıdır. İşaret dili teknolojilerinin geliştirilmesinde Sağır topluluğun liderliği ve katılımı kritik önem taşımaktadır. Teknolojik çözümler, kullanıcıların çeşitli ihtiyaçlarını karşılayacak şekilde tasarlanmalı ve insan iletişimini destekleyici bir rol üstlenmelidir. Bu alandaki araştırmaların başarısı, teknik gelişmelerin yanı sıra sosyal ve kültürel faktörlerin de dikkate alınmasına bağlıdır.

3. OLASILIK VE BİLGİ KURAMI

İşaret dili tanıma, doğası gereği belirsizlik içeren bir problemdir. Bu nedenle, olasılık ve bilgi kuramı, işaret dili tanıma sistemlerinin tasarımında ve analizinde kritik bir rol oynamaktadır. Olasılık kuramı, belirsiz ifadeleri matematiksel olarak modellemek için kullandığımız temel çerçevedir (Jaynes 2003). İşaret dili tanıma sistemlerinde, bu kuram sistemin nasıl çalışması gerektiğini belirlemek ve performansını teorik olarak analiz etmek için kullanılır. İşaret dili tanımada karşılaşılan belirsizlik, çeşitli kaynaklardan gelebilir: işaretçilerin hareketlerindeki doğal varyasyonlar, kamera açısı veya çözünürlük sınırlamaları nedeniyle kaybedilen detaylar ve işaret dilinin tüm karmaşıklığını yakalamadaki zorluklar. Bu belirsizlik kaynakları, olasılıksal modellerin işaret dili tanımada neden önemli olduğunu açıklar (Pearl 1988).

İşaret dili tanımada hem frekansçı hem de Bayesçi olasılık yaklaşımları kullanılabilir. Frekansçı yaklaşım, belirli bir işaretin görülme sıklığına dayanırken, Bayesçi yaklaşım, mevcut kanıtlar ışığında bir işaretin doğruluğuna olan inancımızı temsil eder. Bu iki yaklaşımın aynı aksiyomlar tarafından kontrol edilmesi gerektiği, Ramsey (1926) tarafından gösterilmiştir. Bilgi kuramı, bir olasılık dağılımındaki belirsizliği ölçmemize olanak tanır. İşaret dili tanımada bilgi kuramı, özellik seçimi, model karmaşıklığının optimizasyonu ve veri sıkıştırma gibi alanlarda kullanılabilir. Bu, aşırı uyumu (overfitting) önlemek ve işaret verilerini etkili bir şekilde temsil etmek için önemlidir.

4. MAKİNE ÖĞRENMESİ

Makine Öğrenimi (ML), sistemlerin verilerdeki örüntüleri analiz ederek bilgi edinmesini ve bu bilgiyi kullanarak karar verme süreçlerini optimize etmesini amaçlayan bir yapay zeka alt alanıdır. Bu süreç, genellikle bir yazılım modeli aracılığıyla gerçekleştirilir ve model, belirli bir görev için doğru tahminler yapmak üzere verilerle parametrelerinin ayarlandığı bir eğitim sürecinden geçer. Tom Mitchell'in makine öğrenimi tanımı, bu alanın üç temel unsuruna odaklanmaktadır: deneyim, soru (veya görev) ve performans ölçümü. Bu tanıma göre, bir hesaplama yaklaşımının makine öğrenimi olarak nitelendirilebilmesi için, belirlenen bir görev ve performans ölçüsü doğrultusunda deneyim yoluyla öğrenmesi ve deneyim arttıkça görevdeki performansını da iyileştirmesi gerekmektedir.

Makine öğrenimi süreçleri, genel olarak üç ana kategoriye ayrılmaktadır: denetimli öğrenme, denetimsiz öğrenme ve pekiştirmeli öğrenme. Denetimli öğrenmede, veri kümesi etiketli örnekler içerir ve model, bir öge ile doğru yanıt arasındaki örüntüleri çok sayıda örnek üzerinden öğrenir. Denetimsiz öğrenmede ise veri kümesi etiketli örnekler içermez ve amaç, herhangi bir denetim olmaksızın verilerdeki doğal örüntüleri belirlemektir. Pekiştirmeli öğrenme, durumlar, eylemler, ödüller ve politikalar içeren bir ortamda tanımlanır ve model, bir durum verildiğinde maksimum beklenen ortalama ödülü getiren bir politika öğrenmeye çalışır.

İşaret dili tanıma bağlamında, makine öğrenimi teknikleri büyük önem taşımaktadır. Özellikle denetimli öğrenme yöntemleri, etiketlenmiş işaret dili veri setleri üzerinde eğitilerek, yeni ve görülmemiş işaretleri doğru bir şekilde sınıflandırabilmektedir. Destek Vektör Makineleri (SVM) ve Karar Ağaçları gibi geleneksel makine öğrenimi algoritmaları, işaret dili tanıma görevlerinde başarıyla uygulanmıştır. Denetimsiz öğrenme teknikleri, işaret dili verilerindeki gizli yapıları keşfetmek ve benzer işaretleri gruplamak için kullanılabilirken, pekiştirmeli öğrenme, sürekli işaret dili çevirisi gibi daha karmaşık görevlerde potansiyel uygulamalara sahiptir.

4.1 Makine Öğrenmesi Türleri

Makine öğrenmesi yaklaşımları genel olarak üç ana kategoride incelenmektedir: Denetimli Öğrenme (Supervised Learning), Denetimsiz Öğrenme (Unsupervised Learning) ve Pekiştirmeli Öğrenme (Reinforcement Learning).

4.1.1 Denetimli öğrenme (Supervised learning)

Denetimli öğrenme, makine öğrenmesinin temel yaklaşımlarından biri olup, işaret dili tanıma alanında yaygın olarak kullanılmaktadır. Bu yöntemde, algoritma etiketlenmiş veri setleri üzerinde eğitilir. Eğitim süreci, girdiler (örneğin, işaret görüntüleri veya video kareleri) ile bunlara karşılık gelen bilinen çıktılar (örneğin, işaretin anlamı veya kategorisi) arasındaki ilişkiyi öğrenmeyi amaçlar. İşaret dili tanıma bağlamında, denetimli öğrenme süreci şu şekilde işler: Her işaret örneği, ilişkili bir anlam veya kategori ile eşleştirilir. Örneğin, bir el hareketi görüntüsü "MERHABA" işareti olarak etiketlenebilir. Algoritma, bu işaret-anlam çiftlerini analiz ederek bir tahmin modeli oluşturur ve işaretin görsel özellikleri ile anlamı arasındaki ilişkiyi öğrenir. Eğitim sonrasında model, yeni ve daha önce görmediği işaretler için anlamları tahmin edebilir hale gelir.

Denetimli öğrenme, işaret dili tanımada genellikle iki ana alanda uygulanır: sınıflandırma ve regresyon. Sınıflandırma, işaretleri önceden belirlenmiş kategorilere ayırma işlemidir; örneğin, alfabedeki harfleri veya yaygın kelimeleri tanıma. Regresyon ise işaretin sürekli bir değerle ilişkilendirilmesini içerir; örneğin, bir işaretin yoğunluğunu veya süresini tahmin etme.

Bu yaklaşımın işaret dili tanımada birçok avantajı vardır. Yeterli ve kaliteli etiketli veri ile eğitildiğinde, denetimli öğrenme modelleri yüksek doğruluk oranları elde edebilir. Ayrıca, farklı işaret dilleri veya özel işaret setleri için özelleştirilebilir. Bazı denetimli öğrenme modelleri, kararlarının arkasındaki mantığı açıklayabilme özelliğiyle de öne çıkar. Ancak, denetimli öğrenmenin bazı zorlukları da mevcuttur. Büyük miktarda etiketli veri gereksinimi, zaman alıcı ve maliyetli olabilir. Model, eğitim verisinde görmediği

işaretleri tanımakta zorlanabilir. Ayrıca, bazı işaretlerin diğerlerinden daha sık kullanılması, modelin bazı işaretlere yanlı olmasına neden olabilir.

4.1.2 Denetimsiz öğrenme (Unsupervised learning)

Denetimsiz öğrenme, makine öğrenmesinin önemli bir alt dalıdır ve işaret dili tanıma alanında da potansiyel uygulamaları bulunmaktadır. Bu yaklaşım, etiketlenmemiş veri setleri üzerinde çalışarak verilerdeki gizli yapıları veya örüntüleri keşfetmeyi amaçlar. İşaret dili bağlamında, denetimsiz öğrenme, etiketlenmemiş işaret görüntüleri veya video karelerindeki doğal grupları veya ilişkileri bulmak için kullanılabilir.

Denetimsiz öğrenmenin işaret dili tanıma alanındaki potansiyel uygulamaları şunları içerebilir:

1. Kümeleme: Benzer el hareketlerini veya işaretleri gruplandırarak, bilinmeyen işaret kategorilerini keşfetmek için kullanılabilir. Bu, yeni işaretlerin veya varyasyonların tanımlanmasına yardımcı olabilir.
2. Boyut İndirgeme: İşaret görüntülerinden veya video karelerinden elde edilen yüksek boyutlu verileri daha düşük boyutlu bir uzaya indirgeyerek, işaret tanıma sistemlerinin verimliliğini artırabilir.
3. Anomali Tespiti: Standart işaret dilinden sapan veya alışılmadık işaretleri tespit etmek için kullanılabilir. Bu, işaret dili öğrenme süreçlerinde veya otomatik çeviri sistemlerinde hataları belirlemeye yardımcı olabilir.
4. Özellik Öğrenme: Etiketli veri olmadan işaret görüntülerinden anlamlı özellikleri çıkarmak için kullanılabilir. Bu, daha sonra denetimli öğrenme modellerinde kullanılacak zengin özellik temsilleri oluşturabilir.

Denetimsiz öğrenme, özellikle büyük miktarda etiketlenmemiş işaret dili verisi olduğunda ve bu verilerin manuel olarak etiketlenmesi zor veya maliyetli olduğunda

değerli olabilir. Örneğin, farklı işaret dili lehçelerini veya bölgesel varyasyonları otomatik olarak tespit etmek için kullanılabilir. Bu tez çalışmasında, denetimsiz öğrenme yöntemleri doğrudan kullanılmamıştır, çünkü çalışmamızda kullanılan veriler etiketlidir ve temel problemimiz sınıflandırmadır. Ancak, denetimsiz öğrenme teknikleri, veri ön işleme veya özellik çıkarma aşamalarında yardımcı yöntemler olarak değerlendirilmiştir. Örneğin, otokodlayıcılar gibi denetimsiz derin öğrenme teknikleri, işaret görüntülerinden daha iyi özellik temsilleri çıkarmak için kullanılabilir.

4.1.3 Pekiştirmeli öğrenme (Reinforcement learning)

Pekiştirmeli öğrenme, makine öğrenmesinin diğer ana yaklaşımlarından farklı olarak, bir ajanın çevresiyle etkileşime girerek öğrenmesini sağlayan bir yöntemdir. Bu yaklaşım, insan öğrenmesini taklit ederek, deneme-yanılma yoluyla optimal davranışı keşfetmeyi amaçlar (Sutton vd. 2018). İşaret dili tanıma bağlamında, pekiştirmeli öğrenme doğrudan uygulanmasa da, bazı spesifik senaryolarda potansiyel kullanım alanları bulunmaktadır. Pekiştirmeli öğrenmede, ajan belirli bir durumda bir eylem gerçekleştirir, çevreden bir ödül veya ceza alır ve yeni bir duruma geçer. Amaç, uzun vadede toplam ödülü maksimize eden optimal bir strateji geliştirmektir (Kaelbling vd. 1996). Bu süreç, işaret dili tanıma sistemlerinin sürekli olarak kendini iyileştirmesi ve kullanıcı geri bildirimleriyle adaptasyon sağlaması için uyarlanabilir.

Bu tez çalışmasında, pekiştirmeli öğrenme yöntemleri doğrudan kullanılmamıştır, çünkü Türk İşaret Dili tanıma problemi genellikle denetimli öğrenme yaklaşımıyla ele alınmaktadır. Ancak, gelecekteki çalışmalarda, özellikle gerçek zamanlı işaret dili çevirisi gibi etkileşimli sistemlerde pekiştirmeli öğrenme teknikleri değerlendirilebilir.

4.2 Makine Öğrenmesi Uygulamaları

Makine öğrenmesi, işaret dili tanıma alanında çeşitli uygulamalarla karşımıza çıkmaktadır. Bu bölümde, özellikle sınıflandırma ve kümeleme tekniklerinin işaret dili tanımadaki rolünü inceleyeceğiz.

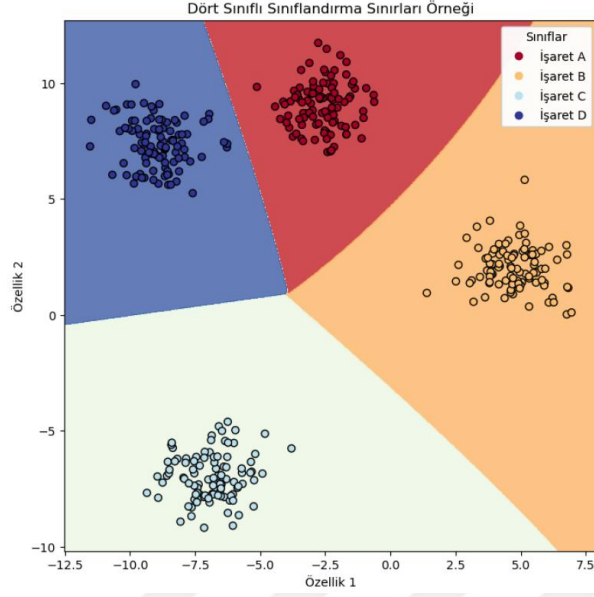
4.2.1 Sınıflandırma (Classification)

Sınıflandırma, işaret dili tanıma alanında en yaygın kullanılan makine öğrenmesi uygulamalarından biridir. Alpaydin'in (2020) belirttiği gibi, sınıflandırma, verilen girdileri önceden tanımlanmış kategorilere veya sınıflara atama işlemidir. İşaret dili tanıma bağlamında, bu teknik el hareketlerini, yüz ifadelerini ve vücut duruşlarını belirli işaretlere veya kelimelere eşlemek için kullanılır.

Sınıflandırmanın işaret dili tanımadaki temel özellikleri şunlardır:

- Denetimli öğrenme yaklaşımı kullanılır, yani etiketlenmiş işaret dili veri seti ile model eğitilir.
- Çıktılar kategoriktir; her bir işaret veya hareket bir sınıfı temsil eder.
- Eğitim verileri kullanılarak bir tahmin modeli geliştirilir.
- Eğitilmiş model, yeni ve etiketlenmemiş işaret verilerini sınıflandırmak için kullanılır.

İşaret dili tanımda yaygın olarak kullanılan sınıflandırma algoritmaları arasında Karar Ağaçları, Naive Bayes, Yapay Sinir Ağları, Destek Vektör Makineleri ve k-En Yakın Komşu bulunmaktadır (Hastie vd. 2009). Bu algoritmalar, işaret verileri arasındaki örüntüleri keşfederek yeni örneklerin hangi işarete veya kelimeye ait olacağını tahmin eder.



Şekil 4.1 Sınıflandırma sınırları örneği

4.2.2 Kümeleme (Clustering)

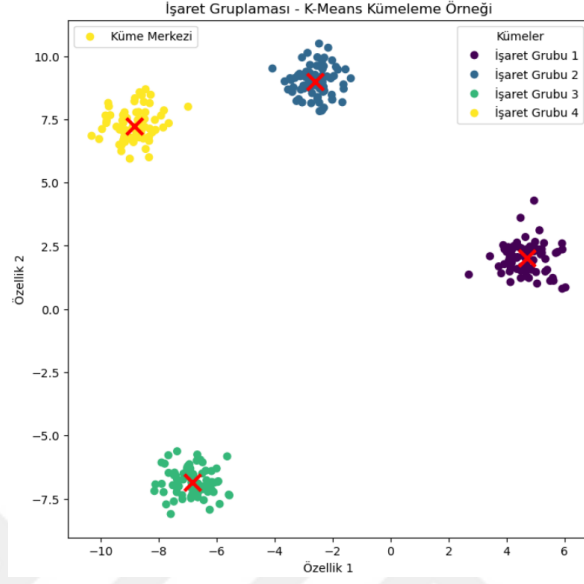
Kümeleme, denetimsiz öğrenmenin temel uygulamalarından biri olup, işaret dili tanıma alanında doğrudan sınıflandırma için kullanılsa da, veri ön işleme veya özellik çıkarma aşamalarında önemli rol oynayabilir. Jain'in (2010) tanımladığı gibi, kümeleme, veri setindeki örnekleri benzerliklerine göre gruplara ayırma işlemidir.

İşaret dili tanımda kümelemenin potansiyel uygulamaları şunları içerir:

- Benzer el hareketlerini gruplandırma
- İşaretlerin alt kategorilerini belirleme
- Etiketlenmemiş işaret verilerinin doğal yapısını keşfetme

Kümeleme algoritmaları, işaret verilerini gruplandırırken genellikle Öklid, Manhattan veya Minkowski gibi uzaklık ölçütlerini kullanır. K-Means ve Hiyerarşik Kümeleme, işaret dili verilerinin analizinde kullanılabilecek yaygın kümeleme algoritmalarıdır (Xu ve Wunsch 2005). Bu tez çalışmasında, kümeleme teknikleri doğrudan sınıflandırma için kullanılmamıştır. Ancak, veri ön işleme aşamasında, özellikle karmaşık işaretlerin alt bileşenlerini belirlemede veya benzer işaretleri gruplandırmada kümeleme tekniklerinden

yararlanılmıştır. Bu, sınıflandırma modellerinin daha etkili öğrenmesine ve genellemesine katkıda bulunmuştur.

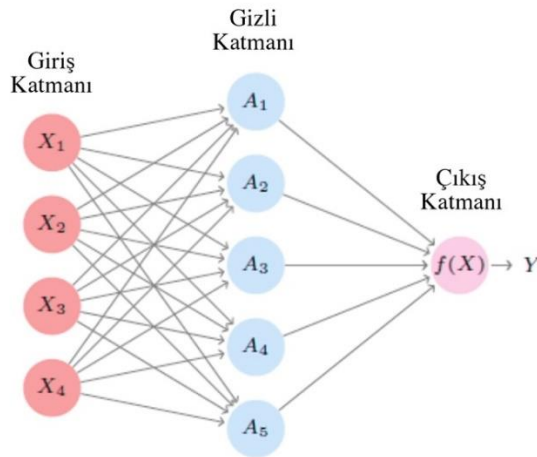


Şekil 4.2 Kümeleme işlemi örneği

5. DERİN ÖĞRENME

Derin öğrenme, makine öğrenmesi ve yapay zeka alanlarında son yıllarda büyük ilgi gören ve aktif bir araştırma konusu haline gelen bir alt disiplindir. Temeli 1980'lerin sonlarında popülerlik kazanan sinir ağlarına dayanmaktadır. O dönemde, sinir ağları etrafında kayda değer bir heyecan oluşmuş ve bu gelişmeler, Sinirsel Bilgi İşleme Sistemleri toplantıları (NeurIPS, önceden NIPS) gibi etkinliklerin itici gücü haline gelmiştir. Sinir ağlarının özellikleri, makine öğrenmesi uzmanları, matematikçiler ve istatistikçiler tarafından analiz edilmiş, algoritmalar geliştirilmiş ve metodoloji zamanla istikrar kazanmıştır. Ancak, 2000'li yılların başında Destek Vektör Makineleri (SVM), boosting ve rastgele ormanlar gibi yeni yöntemlerin ortaya çıkmasıyla sinir ağlarının popüleritesi bir süre azalmıştır. Bu durumun nedenleri arasında, sinir ağlarının çok fazla ince ayar gerektirmesi ve yeni yöntemlerin daha otomatik olması gösterilmektedir.

2010 yılından sonra ise sinir ağları, "derin öğrenme" adıyla yeniden gündeme gelmiştir. Bu yeni yaklaşım, yeni mimariler, ek özellikler ve özellikle görüntü-video sınıflandırma, konuşma ve metin modelleme gibi alanlarda elde ettiği başarılarla dikkat çekmiştir. Bu başarıların temel nedeni, bilim ve endüstride dijitalleşmenin yaygınlaşması sonucu oluşan daha büyük eğitim veri setlerinin kullanılabilirliği olarak görülmektedir.

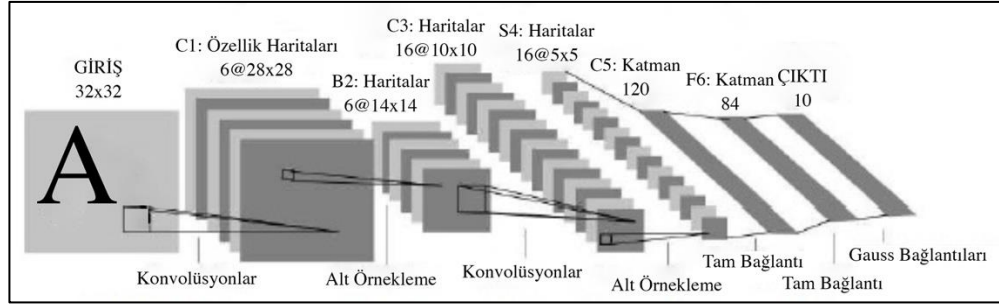


Şekil 5.1 Tek bir gizli katmana sahip sinir ağı

Gizli katman, girdilerin $X_1, X_2, \dots, X_p$ doğrusal kombinasyonlarının doğrusal olmayan dönüşümleri olan $A_k = h_k(X)$ aktivasyonlarını hesaplar. Bu nedenle bu A_k 'lar doğrudan gözlemlenmez. $h_k(\cdot)$ fonksiyonları önceden sabitlenmez, ancak ağın eğitimi sırasında öğrenilir. Çıkış katmanı, bu A_k aktivasyonlarını girdi olarak kullanan doğrusal bir modeldir ve $f(X)$ fonksiyonu ile sonuçlanır.

Derin öğrenme, üç veya daha fazla katmandan oluşan sinir ağlarının kullanılmasıyla karakterize edilen bir makine öğrenimi dalıdır. Bu sinir ağları, insan beynindeki bilişsel süreçleri ve faaliyetleri taklit etmeyi amaçlamakta ve büyük miktarda veriyle eğitilip gelişebilmektedir (LeCun vd. 2015; Al-shaheen vd. 2022). Tek katmanlı bir sinir ağının temel düzeyde tahmin yeteneğine sahip olduğu, ancak derin öğrenme ağlarının potansiyelinin derinlikleri arttıkça önemli ölçüde yükseldiği vurgulanmaktadır (LeCun ve ark., 2015). Derin öğrenme, yapay zeka alanında geniş bir uygulama yelpazesine sahiptir. Otomasyonun geliştirilmesine ve karmaşık analitik görevlerin insan müdahalesi olmadan yerine getirilmesine olanak sağlamaktadır. Otonom araçlar gibi gelişmiş teknolojilerin yanı sıra, mobil cihaz dijital asistanları, ses kontrollü cihazlar ve kredi kartı dolandırıcılığının tespiti gibi birçok farklı hizmet ve uygulamada da yaygın olarak kullanılmaktadır (LeCun vd. 2015). İşaret dili tanıma alanında derin öğrenme, özellikle Evrişimli Sinir Ağları (CNN) ve Uzun Kısa Süreli Bellek (LSTM) ağları gibi mimariler aracılığıyla önemli ilerlemeler sağlamıştır. Bu modeller, işaret dilinin karmaşık ve dinamik yapısını öğrenme ve temsil etme konusunda yüksek performans göstermektedir. Bu tez çalışmasında, derin öğrenme yöntemlerinin Türk İşaret Dili tanıma görevindeki etkinliği incelenecek ve geleneksel istatistiksel yöntemlerle karşılaştırılacaktır. Özellikle, CNN-LSTM hibrit modellerinin performansı değerlendirilecek ve bu modellerin işaret dili tanımadaki güçlü ve zayıf yönleri analiz edilecektir.

Derin Öğrenme, problem alanı hakkında önceden kodlanmış bilgi gerektiren geleneksel sistemlerin aksine, kendi bilgisini oluşturma yeteneğine dayanan yenilikçi bir öğrenme yaklaşımıdır. Bu yaklaşım, verileri temsil etmek için hangi özelliklerin çıkarılması gerektiğini verilerden öğrenme prensibine dayanır. Derin öğrenme, daha basit özelliklere dayalı karmaşık özellikleri ifade etmek üzere geliştirilmiştir ve insan beyninin yapısından ilham almaktadır. Şekil 5.2'de görüldüğü gibi, tipik bir derin öğrenme mimarisinde, erken katmanlar köşeler ve lekeler gibi basit özellikleri öğrenirken, sonraki katmanlar daha karmaşık ve bütünleşik özellikleri temsil eder. Bu hiyerarşik öğrenme yapısı, derin öğrenmenin bilgisayarla görme ve doğal dil işleme gibi çeşitli alanlarda etkili olmasını sağlamıştır (Chai vd. 2021).



Şekil 5.2 Tipik bir derin öğrenme mimarisinin katmanlı yapısı ve özellik çıkarım süreci

Bilgisayarla görme (CV) görevlerinde, Evrimsel Sinir Ağları (CNN), Otomatik Kodlayıcılar ve Seyrek Kodlama gibi çeşitli yöntemler kullanılmaktadır. Yapılan karşılaştırmalarda, CNN'lerin mimari olarak en uygun yöntem olduğu belirlenmiştir. Ancak, doğruluk ve model boyutlarındaki kısıtlamalar nedeniyle bazı zorluklar yaşanmıştır. Bu zorluklar arasında:

1. Belirli mimarilerin üstün performansının tam olarak anlaşılamaması
2. Sınırlı veri kümesiyle eğitimin getirdiği kısıtlamalar
3. Gerçek zamanlı görüntü elde etmedeki karmaşıklıklar
4. Daha güçlü modellere duyulan ihtiyaç

sayılabilir (Chai ve ark., 2021).

Derin öğrenme sürecinin temel amacı, tahminlerin gerçek çıktılara mümkün olduğunca yakın olması için hatayı minimize etmektir. Bu süreç, önceden belirlenmiş bazı koşullar (örneğin, epok sayısı, kayıp eşiği) sağlanana kadar tekrar eder. Derin sinir ağlarını eğitmenin en büyük zorluklarından biri, büyük ve etiketli veri kümelerine duyulan ihtiyaçtır. İşaret dili tanıma, derin öğrenmenin uygulandığı özel bir araştırma alanıdır. Ancak, bu alan hala diğer alanlarda olduğu kadar büyük miktarda etiketli veriye sahip değildir. Bu durum, işaret dili tanıma için derin öğrenme modellerinin geliştirilmesinde ve uygulanmasında önemli bir zorluk oluşturmaktadır.

5.1 Tek Katmanlı Sinir Ağları

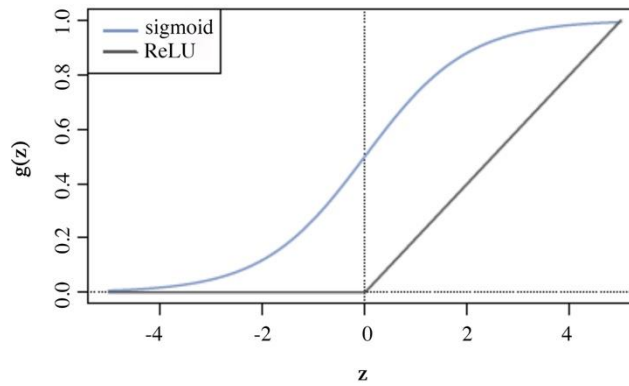
Bir sinir ağı, p değişkenli bir girdi vektörü $X = (X_1, X_2, \dots, X_p)$ alır ve Y yanıtını tahmin etmek için doğrusal olmayan bir $f(X)$ fonksiyonu oluşturur. Önceki bölümlerde ağaçlar, boosting ve genelleştirilmiş katkı modelleri kullanılarak doğrusal olmayan tahmin modelleri inşa edilmiştir. Sinir ağlarını bu yöntemlerden ayıran, modelin özel yapısıdır. Formül 1'de, $p = 4$ tahmin değişkeni kullanarak nicel bir yanıtı modellemek için basit bir ileri beslemeli sinir ağı gösterilmektedir.

Sinir ağları terminolojisinde, dört özellik $X_1, \dots, X_4$ girdi katmanındaki birimleri oluşturur. Oklar, girdi katmanından gelen her bir girdinin, K gizli katman birimlerinin her birine beslendiğini gösterir (K değeri seçilebilir; burada 5 olarak belirlenmiştir). Sinir ağı modeli şu forma sahiptir:

$$\begin{aligned} f(X) &= \beta_0 + \sum_{k=1}^K \beta_k h_k(X) \\ &= \beta_0 + \sum_{k=1}^K \beta_k g(w_{k0} + \sum_{j=1}^p w_{kj} X_j) \end{aligned} \quad (1)$$

Bu model iki adımda oluşturulur. İlk olarak, gizli katmandaki K aktivasyon A_k , $k = 1, \dots, K$, girdi özellikleri $X_1, \dots, X_p$ 'nin fonksiyonları olarak hesaplanır:

$$A_k = h_k(X) = g(w_{k0} + \sum_{j=1}^p w_{kj} X_j) \quad (2)$$



Şekil 5.3 Aktivasyon fonksiyonları

Burada $g(z)$, önceden belirlenen doğrusal olmayan bir aktivasyon fonksiyonudur. Her A_k , temel fonksiyonlara benzer şekilde, orijinal özelliklerin farklı bir dönüşümü $h_k(X)$ olarak düşünülebilir. Gizli katmandan gelen bu K aktivasyon daha sonra çıktı katmanına beslenir ve şu sonucu verir:

$$f(X) = \beta_0 + \sum_{k=1}^K \beta_k A_k, \quad (3)$$

Bu, $K = 5$ aktivasyonda doğrusal bir regresyon modelidir. Tüm parametreler $\beta_0, \dots, \beta_k$ ve $w_{10}, \dots, w_{kp}$ verilerden tahmin edilmelidir. Sinir ağlarının ilk örneklerinde, sigmoid aktivasyon fonksiyonu tercih edilmiştir:

$$g(z) = \frac{e^z}{1+e^z} = \frac{1}{1+e^{-z}} \quad (4)$$

Bu, lojistik regresyonda doğrusal bir fonksiyonu sıfır ile bir arasındaki olasılıklara dönüştürmek için kullanılan aynı fonksiyondur (bkz. Formül 2). Modern sinir ağlarında tercih edilen seçenek, ReLU (düzeltilmiş doğrusal birim) aktivasyon fonksiyonudur ve şu formu alır:

$$g(z) = (z)_+ = \begin{cases} 0 & \text{eğer } z < 0 \\ z & \text{aksi takdirde} \end{cases} \quad (5)$$

Bir ReLU aktivasyonu, bir sigmoid aktivasyonundan daha verimli bir şekilde hesaplanabilir ve depolanabilir. Sıfırda eşik değerine sahip olmasına rağmen, doğrusal bir fonksiyona (formül 2) uygulandığı için, sabit terim w_{k0} bu dönüm noktasını kaydıracaktır.

Özetle, formül 1'de gösterilen model, X 'in beş farklı doğrusal kombinasyonunu hesaplayarak beş yeni özellik türetir ve ardından her birini dönüştürmek için bir aktivasyon fonksiyonu $g(\cdot)$ ile sıkıştırır. Nihai model, bu türetilmiş değişkenlerde doğrusaldır. Sinir ağı adı, başlangıçta bu gizli birimlerin beyindeki nöronlara benzer olduğu düşüncesinden türetilmiştir - $A_k = h_k(X)$ aktivasyonlarının bire yakın değerleri ateşleme, sıfıra yakın olanlar ise sessiz olarak düşünülmüştür (sigmoid aktivasyon fonksiyonu kullanılarak).

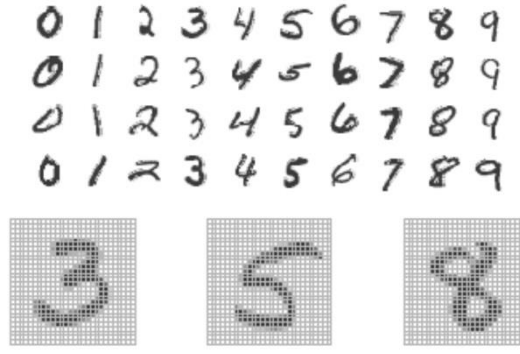
Bir sinir ağını uydurmak, formül 1'deki bilinmeyen parametrelerin tahmin edilmesini gerektirir. Nicel bir yanıt için genellikle kare hata kaybı kullanılır, böylece parametreler şu ifadeyi minimize edecek şekilde seçilir:

$$\sum_{i=1}^n (y_i - f(x_i))^2 \quad (6)$$

5.2 Çok Katmanlı Sinir Ağları

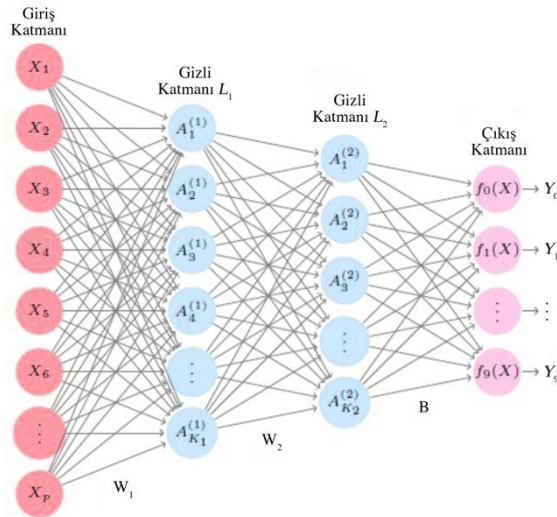
Modern sinir ağları, genellikle birden fazla gizli katmana ve her katmanda çok sayıda birime sahip karmaşık yapılardır. Teorik olarak, çok sayıda birime sahip tek bir gizli katman, çoğu fonksiyonu yaklaşık olarak temsil etme yeteneğine sahip olsa da, pratikte çoklu katmanların kullanımı öğrenme görevini önemli ölçüde kolaylaştırmıştır. Bu yaklaşım, her biri makul boyutta olan çoklu katmanlar aracılığıyla iyi bir çözümü keşfetmeyi daha etkin hale getirmiştir.

Büyük bir yoğun ağ, ünlü ve halka açık MNIST el yazısı rakam veri seti üzerinde gösterilecektir. Şekil 5.4'te bu rakamların örnekleri gösterilmektedir. Amaç, görüntüleri doğru rakam sınıfına (0-9) sınıflandırmak için bir model oluşturmaktır. Her görüntü $p = 28 \times 28 = 784$ piksele sahiptir ve her piksel, o küçük karedeki yazılı rakamın göreceli miktarını temsil eden 0 ile 255 arasında sekiz bitlik bir gri tonlama değeridir. Bu pikseller, girdi vektörü X 'te (örneğin, sütun sırasında) depolanır. Çıktı, sınıf etiketidir ve 10 kukla değişkenden oluşan bir $Y = (Y_0, Y_1, \dots, Y_9)$ vektörü ile temsil edilir; etikete karşılık gelen konumda bir, diğer yerlerde sıfır bulunur. Makine öğrenmesi topluluğunda bu, *one-hot kodlama* olarak bilinir. 60.000 eğitim görüntüsü ve 10.000 test görüntüsü bulunmaktadır.



Şekil 5.4 MNIST veri kümesinden el yazısı rakam örnekleri

Rakam tanıma problemleri, 1980'lerin sonlarında sinir ağı teknolojisinin gelişiminde önemli bir rol oynamıştır. AT&T Bell Laboratuvarları gibi kurumlar, bu tür örüntü tanıma görevlerini çözmek için sinir ağlarını geliştirmeye odaklanmıştır. İnsanlar için görece basit olan bu görevler, makineler için oldukça zordur. İnsan beyninin büyük bir bölümünü kaplayan görsel sistemimiz, evrimsel süreçte hayatta kalma açısından kritik öneme sahip olmuştur. Buna karşın, makinelerin insan performansına eşdeğer tanıma yeteneklerine ulaşması 30 yıldan fazla sürmüştür. İşaret dili tanıma bağlamında, çok katmanlı sinir ağları özellikle önemlidir. İşaret dilinin karmaşık ve çok boyutlu yapısı - el şekilleri, hareketler, yüz ifadeleri ve vücut duruşu gibi - çok katmanlı bir yapıyı gerektirir.



Şekil 5.5 El yazısı rakam tanıma için iki gizli katmanlı sinir ağı mimarisi

Şekil 5.5, rakam sınıflandırma görevini çözmede iyi çalışan çok katmanlı bir ağ mimarisini göstermektedir. Bu yapı, Şekil 5.5'ten birkaç açıdan farklılık göstermektedir:

- Bir yerine iki gizli katmana (L_1 : 256 birim, L_2 : 128 birim) sahiptir. İlerleyen bölümlerde yedi gizli katmana sahip bir ağ görülecektir.
- Tek bir çıktı değişkeni yerine, on çıktı değişkenine sahiptir. Bu on değişken, aslında tek bir niteliksel değişkeni temsil etmekte ve oldukça bağımlıdır. (Açıklık açısından, rakam sınıfı 0-9 ile indekslenmiştir, 1-10 yerine.) Daha genel olarak, çoklu görev öğreniminde, tek bir ağ ile farklı yanıtların aynı anda tahmin edilmesi mümkündür; bu yanıtların hepsi gizli katmanların oluşumunda rol oynar.
- Ağın eğitimi için kullanılan kayıp fonksiyonu, çok sınıflı sınıflandırma görevi için özel olarak uyarlanmıştır.

İlk gizli katman, (2) formülünde olduğu gibi

$$\begin{aligned} A_k^{(1)} &= h_k^{(1)}(X) \\ &= g\left(w_{k0}^{(1)} + \sum_{j=1}^p w_{kj}^{(1)} X_j\right) \end{aligned} \quad (7)$$

($k = 1, \dots, K_1$) için tanımlanmıştır. İkinci gizli katman, ilk gizli katmanın aktivasyonları ($A_k^{(1)}$)'i girdi olarak alır ve yeni aktivasyonları hesaplar

$$\begin{aligned} A_\ell^{(2)} &= h_\ell^{(2)}(X) \\ &= g\left(w_{\ell 0}^{(2)} + \sum_{k=1}^{K_1} w_{\ell k}^{(2)} A_k^{(1)}\right) \end{aligned} \quad (8)$$

$\ell = 1, \dots, K_2$ için. İkinci katmandaki her bir aktivasyonun $A_\ell^{(2)} = h_\ell^{(2)}(X)$ girdi vektörü X 'in bir fonksiyonu olduğu unutulmamalıdır. Bu durum, her ne kadar açıkça ilk katmandaki $A_k^{(1)}$ 'in fonksiyonu olsa da, $A_k^{(1)}$ 'in de X 'in bir fonksiyonu olması nedeniyle geçerlidir. Daha fazla gizli katman olması durumunda da aynı durum geçerli olacaktır. Bu şekilde, dönüşümlerin bir zinciri aracılığıyla, ağ, son olarak çıktı katmanına özellikler olarak beslenen oldukça karmaşık dönüşümler oluşturabilmektedir.

Aktivasyonlar ve ağırlıkların (katsayıların) hangi katmana ait olduğunu belirtmek için $h_\ell^{(2)}(X)$ ve $w_{\ell j}^{(2)}$ gibi ek üst simge notasyonları (7) ve (8) formüllerinde tanıtılmıştır; bu durumda ikinci katmana aittirler. Formül 4'teki w_1 notasyonu, girdi katmanından ilk gizli katman L_1 'e beslenen tüm ağırlık matrisini temsil etmektedir. Bu matris $785 \times 256 = 200.960$ elemana sahip olacaktır; 784 yerine 785 olmasının sebebi, kesişim ya da bias terimini hesaba katmamız gerektiğindendir. İlk gizli katman $A_k^{(1)}$ 'in her bir elemanı, $257 \times 128 = 32.896$ boyutunda olan w_2 ağırlık matrisi aracılığıyla ikinci gizli katman L_2 'ye beslenir.

Şimdi, bir çıktının yerine on yanıtın olduğu çıktı katmanına geliyoruz. İlk adım, tek modelimize (formül 1) benzer on farklı lineer modelin hesaplanmasıdır,

$$\begin{aligned} Z_m &= \beta_{m0} + \sum_{\ell=1}^{K_2} \beta_{m\ell} h_\ell^{(2)}(X) \\ &= \beta_{m0} + \sum_{\ell=1}^{K_2} \beta_{m\ell} A_\ell^{(2)}, \end{aligned} \quad (9)$$

$m = 0, 1, \dots, 9$ için. B matrisi, bu ağırlıkların tümünü ($129 \times 10 = 1.290$) depolar.

Bu yanıtların hepsi ayrı nicel yanıtlar olsaydı, her bir $f_m(X) = Z_m$ olarak ayarlamak yeterli olurdu. Ancak, tahminlerimizin sınıf olasılıklarını $f_m(X) = \Pr(Y = m | X)$ temsil etmesini isteriz, tıpkı Bölüm 4.3.5'teki çok terimli lojistik regresyonda olduğu gibi. Bu nedenle özel softmax aktivasyon fonksiyonu kullanılır.

$$f_m(X) = \Pr(Y = m | X) = \frac{e^{Z_m}}{\sum_{\ell=0}^9 e^{Z_\ell}} \quad (10)$$

$m = 0, 1, \dots, 9$ için. Bu, 10 sayının olasılık gibi davranmasını sağlar (negatif olmayan ve toplamı bir olan). Her ne kadar amaç bir sınıflandırıcı oluşturmak olsa da, modelimiz aslında her bir 10 sınıf için bir olasılık tahmin eder. Sınıflandırıcı daha sonra görüntüyü en yüksek olasılığa sahip sınıfa atar.

Bu ağı eğitmek için, yanıt niteliksel olduğu için, negatif çok terimli logaritmik olasılığı minimize eden katsayı tahminleri aranır.

$$-\sum_{i=1}^n \sum_{m=0}^9 y_{im} \log (f_m(x_i)) \quad (11)$$

Bu, iki sınıflı lojistik regresyon için kullanılan kriterinin geliştirilmiş hali olan çapraz-entropy olarak da bilinir. Yanıt nicel olsaydı, bunun yerine (6) formülünde olduğu gibi kare-hata kaybı minimize edilirdi.

5.3 Derin Öğrenme Tanımı ve Tarihçesi

Derin öğrenme, yapay zeka ve makine öğrenmesi alanlarında çığır açan bir yaklaşım olarak ortaya çıkmıştır. Temel olarak, derin öğrenme, meta verileri girdi olarak kullanan ve çıktı değerini hesaplamak için girdi verilerinden birkaç gizli katmanlı minimum doğrusal aktarımlar kullanan bir yapay sinir ağı türüdür. Bu yapı, karmaşık ve çok boyutlu veri setlerinden anlamlı özellikler çıkarma ve örüntüler öğrenme yeteneği sayesinde çeşitli alanlarda büyük başarılar elde etmiştir. Derin öğrenme yöntemlerinin en yaygın uygulamalarından biri sınıflandırma işlemidir. Sınıflandırma algoritmaları, çeşitli veri kümelerinde uygulanabilmekte ve her algoritma nesneleri tespit etmek için farklı yetenekler sergilemektedir (Aziz vd. 2020). Bu özellik, derin öğrenme modellerinin farklı problem alanlarına uyarlanabilirliğini göstermektedir.

Derin öğrenmenin işaret dili tanıma alanındaki gelişimi, bu teknolojinin genel evrimiyle paralellik göstermektedir. Derin öğrenme döneminden önce, çoğu işaret dili tanıma sistemi, işaretçilerin pozunu gerçek zamanlı olarak tahmin etmek için Kinect gibi özel sensörler kullanmaktaydı (Shotton vd. 2011; Zafrulla vd. 2011). Bu dönemde ayrıca, düşük çözünürlüklü iskeletleri, yani birkaç eklemi çıkaran işaret dili verileri üzerinde modeller eğitme girişimleri de olmuştur (Pfister vd. 2012). Ancak bu yaklaşımlar, gürültülü tahminlerden muzdarip olmuş ve yetersiz el eklemi çözünürlüğü gibi sorunlarla karşılaşmıştır. Bilgisayarla görmenin diğer alt alanlarında olduğu gibi, insan pozu tahmini de derin öğrenme tabanlı yaklaşımların kullanılmaya başlanmasıyla birlikte önemli gelişmeler göstermiştir. Derin öğrenme modelleri, özellikle Evrişimli Sinir Ağları (CNN) ve Uzun Kısa Süreli Bellek (LSTM) ağları, işaret dili tanıma alanında çığır açıcı sonuçlar elde etmiştir.

Bu gelişmeler, işaret dili tanıma sistemlerinin performansını ve doğruluğunu önemli ölçüde artırmıştır. Derin öğrenme modelleri, el hareketleri, yüz ifadeleri ve vücut duruşu gibi karmaşık ve çok boyutlu özellikleri etkili bir şekilde öğrenebilme yeteneğine sahiptir. Bu sayede, özel sensörlere olan bağımlılık azalmış ve standart kameralarla çekilmiş görüntüler üzerinde daha başarılı tanıma sistemleri geliştirilmiştir.

5.4 Derin Öğrenmenin Kullanım Alanları

Derin öğrenme, son yıllarda birçok alanda etkileyici performans göstererek çığır açıcı gelişmelere öncülük etmiştir. Bu bölümde, derin öğrenmenin çeşitli kullanım alanları ve elde ettiği başarılar ele alınacaktır. Görüntü işleme ve sınıflandırma alanında, derin evrişimli sinir ağları (CNN'ler) devrim niteliğinde ilerlemeler sağlamıştır. Özellikle tıbbi görüntüleme alanında, derin öğrenme modelleri önemli başarılar elde etmiştir. Mamograflerin veya dijital röntgen görüntülerinin makine teşhisi, oftalmoloji göz taramaları ve MRI taramalarının yorumlanması gibi görevlerde derin öğrenme modelleri insan uzmanlara yakın veya bazen daha iyi performans sergilemektedir.

Doğal dil işleme alanında ise tekrarlayan sinir ağları (RNN'ler) öne çıkmaktadır. Konuşma tanıma, dil çevirisi, metin üretimi ve belge modellemesi gibi görevlerde RNN'ler etkileyici sonuçlar vermektedir. Bu başarılar, insan-bilgisayar etkileşimini geliştirmekte ve dil bariyerlerini aşmada önemli rol oynamaktadır. Derin öğrenmenin yanı sıra, makine öğrenmesi alanında rastgele ormanlar, güçlendirme algoritmaları, destek vektör makineleri ve genelleştirilmiş katkı modelleri gibi güçlü araçlar da bulunmaktadır. Ayrıca, doğrusal modeller ve bunların basit varyantları da hala önemini korumaktadır. Yeni bir veri modelleme veya tahmin problemiyle karşılaşıldığında, en son teknoloji olan derin öğrenme yöntemlerine başvurmak cazip gelebilir. Özellikle veri setleri çok büyük olduğunda ve yüksek boyutlu doğrusal olmayan modellerin uygulanmasını destekleyebildiğinde, derin öğrenme modelleri genellikle etkileyici sonuçlar vermektedir. Bununla birlikte, daha basit araçlarla aynı derecede iyi performans gösteren modeller üretebilmek mümkünse, bunların uygulanması ve anlaşılması daha kolay olacak ve potansiyel olarak daha karmaşık yaklaşımlardan daha az kırılgan

olacaktır. Bu nedenle, mümkün olduğunca daha basit modelleri de denemek ve ardından performans/karmaşıklık dengesi temelinde bir seçim yapmak mantıklıdır.

Derin öğrenme, tipik olarak aşağıdaki koşullarda cazip bir seçim haline gelmektedir:

1. Eğitim setinin örnek boyutu son derece büyük olduğunda
2. Modelin yorumlanabilirliği yüksek bir öncelik olmadığında
3. Problem karmaşık ve doğrusal olmayan ilişkiler içerdiğinde
4. Geleneksel yöntemlerle elde edilemeyen yüksek doğruluk gerektiğinde

İşaret dili tanıma bağlamında, derin öğrenme modelleri özellikle etkili olmuştur. El hareketleri, yüz ifadeleri ve vücut duruşu gibi karmaşık ve çok boyutlu özellikleri içeren işaret dili verileri, derin öğrenme modellerinin güçlü özellik çıkarma yeteneklerinden faydalanmaktadır. CNN ve LSTM gibi mimariler, işaret dili tanıma görevlerinde yüksek performans göstermektedir.

5.4.1 Görüntü tanıma

Görüntü tanıma, derin öğrenmenin en etkileyici başarıları elde ettiği alanlardan biridir. Bu teknoloji, dijital görüntülerdeki nesneleri, yüzleri, yazıları veya sahneleri otomatik olarak tanımlama ve sınıflandırma yeteneğine sahiptir. Derin öğrenme modellerinin, özellikle Evrişimli Sinir Ağlarının (CNN) ortaya çıkışı, görüntü tanıma alanında çığır açıcı ilerlemelere yol açmıştır. CNN'ler, görüntülerdeki uzamsal hiyerarşileri öğrenme konusunda üstün bir yeteneğe sahiptir. Bu ağlar, görüntünün düşük seviyeli özelliklerinden (kenarlar, köşeler gibi) başlayarak, daha karmaşık ve soyut özelliklere doğru ilerleyen bir öğrenme süreci gerçekleştirir. Bu yapı, CNN'lerin görüntülerdeki karmaşık desenleri ve yapıları etkili bir şekilde yakalayabilmesini sağlar.

Görüntü tanıma teknolojisi, günümüzde birçok alanda uygulanmaktadır:

1. Tıbbi Görüntüleme: X-ray, MRI ve CT taramalarında hastalık teşhisi
2. Güvenlik Sistemleri: Yüz tanıma ve nesne tespiti

3. Otonom Araçlar: Yol işaretleri, yayalar ve diğer araçların tanınması
4. Sosyal Medya: Fotoğraflardaki kişilerin ve nesnelerin otomatik etiketlenmesi
5. Kalite Kontrol: Üretim hatlarında ürün kusurlarının tespiti

İşaret dili tanıma bağlamında, görüntü tanıma teknolojisi kritik bir rol oynamaktadır. İşaret dilinin görsel doğası gereği, el şekillerinin, parmak pozisyonlarının ve el hareketlerinin doğru bir şekilde tanınması büyük önem taşır. CNN'ler, bu karmaşık görsel özellikleri etkili bir şekilde öğrenebilme ve tanıyabilme yeteneğine sahiptir.

Örneğin, bir işaret dili tanıma sisteminde, CNN'ler şu şekilde kullanılabilir:

1. El Şekli Tanıma: Farklı el şekillerini ve parmak pozisyonlarını sınıflandırma
2. Hareket Tespiti: Ardışık video karelerinde el hareketlerini izleme
3. Yüz İfadesi Analizi: İşaret dilinin önemli bir parçası olan yüz ifadelerini tanıma
4. Vücut Duruşu Analizi: İşaretin bağlamını anlamak için genel vücut duruşunu değerlendirmek

Sonuç olarak, görüntü tanıma teknolojisindeki ilerlemeler, işaret dili tanıma alanında da önemli gelişmelere yol açmıştır. Bu teknolojinin daha da geliştirilmesi, işitme engelli bireylerin iletişim olanaklarını genişletme ve toplumsal entegrasyonlarını artırma potansiyeline sahiptir.

5.4.2 Doğal dil işleme

Doğal Dil İşleme (NLP), derin öğrenmenin önemli uygulama alanlarından biridir ve insan dilinin bilgisayarlar tarafından anlaşılması, işlenmesi ve üretilmesini amaçlar. NLP, metin analizi, dil çevirisi, konuşma tanıma, duygu analizi ve metin üretimi gibi birçok alt alanı kapsar. Derin öğrenme tekniklerinin NLP alanında kullanılmaya başlanmasıyla birlikte, bu alanda çığır açıcı ilerlemeler kaydedilmiştir. Derin öğrenme tabanlı NLP modelleri, özellikle Tekrarlayan Sinir Ağları (RNN), Uzun Kısa Süreli Bellek Ağları (LSTM) ve son zamanlarda Transformer mimarileri, dilin karmaşık yapısını ve bağlamını öğrenme

konusunda üstün performans göstermektedir. Bu modeller, kelimelerin ve cümlelerin anlamsal ve sözdizimsel özelliklerini otomatik olarak çıkarabilme yeteneğine sahiptir.

NLP'nin bazı önemli uygulama alanları şunlardır:

1. Makine Çevirisi: Bir dilden diğerine otomatik çeviri yapma.
2. Duygu Analizi: Metinlerdeki duygu ve görüşleri otomatik olarak tespit etme.
3. Metin Özetleme: Uzun metinleri otomatik olarak özetleme.
4. Soru Cevaplama Sistemleri: Doğal dilde sorulan sorulara otomatik cevaplar üretme.
5. Konuşma Tanıma: Sesli konuşmayı metne dönüştürme.
6. Chatbotlar ve Sanal Asistanlar: İnsanlarla doğal dilde iletişim kurabilen sistemler geliştirme.

İşaret dili tanıma bağlamında, NLP teknikleri özellikle önemlidir. İşaret dili, kendine özgü dilbilgisi ve sözdizimi kurallarına sahip bir dil olduğundan, NLP teknikleri işaret dilinin yapısını anlamak ve modellemek için kullanılabilir. Örneğin:

1. İşaret Dizilimi Analizi: İşaretlerin sırasını ve birbirleriyle olan ilişkilerini analiz etme.
2. İşaret-Metin Çevirisi: İşaret dilini yazılı metne veya konuşma diline çevirme.
3. Anlamsal Analiz: İşaretlerin ve işaret dizilerinin anlamını çıkarma.
4. Bağlam Modelleme: İşaretlerin bağlama dayalı anlamlarını anlama.

NLP tekniklerinin işaret dili tanımadaki kullanımı, sadece işaretleri tanımak değil, aynı zamanda bu işaretlerin anlamını ve bağlamını da anlamak için önemlidir. Bu, daha doğru ve akıcı işaret dili-metin çevirileri yapılmasına olanak sağlayabilir. Ayrıca, NLP teknikleri, işaret dili kullanıcıları için daha etkili iletişim araçları geliştirmeye de yardımcı olabilir. Sonuç olarak, NLP alanındaki derin öğrenme tekniklerinin işaret dili tanıma ve çevirisi alanına uyarlanması, işitme engelli bireylerin iletişim olanaklarını genişletme ve toplumsal entegrasyonlarını artırma potansiyeline sahiptir. Bu tez çalışması, bu potansiyeli Türk İşaret Dili özelinde incelemeyi ve değerlendirmeyi amaçlamaktadır.

5.4.3 Derin yapay sinir ağı uygulama

Sinir ağlarının uygulanması bir miktar karmaşıktır ve burada kısa bir genel bakış sunulmaktadır. Bu fikirler, çok daha karmaşık ağlara genelleştirilebilir. Bu materyali zorlayıcı bulan okuyucular güvenle atlayabilirler. Neyse ki, bu bölümün sonundaki laboratuvarda görüldüğü gibi, model uygulama prosedürünün teknik ayrıntıları hakkında endişelenmeden, sinir ağı modellerini nispeten otomatik bir şekilde uygulamak için iyi yazılımlar mevcuttur.

Formül 1'de gösterilen basit ağ ile başlanmaktadır. Model formül 1'de parametreler $\beta = (\beta_0, \beta_1, \dots, \beta_K)$ ve her bir $w_k = (w_{k0}, w_{k1}, \dots, w_{kp})$, $k = 1, \dots, K$ 'dir. (x_i, y_i) , $i = 1, \dots, n$ gözlemleri verildiğinde, model aşağıdaki doğrusal olmayan en küçük kareler problemini çözerek uygulanabilir:

$$\underset{\{w_k\}_1, \beta}{\text{minimize}} \frac{1}{2} \sum_{i=1}^n (y_i - f(x_i))^2 \quad (16)$$

burada:

$$f(x_i) = \beta_0 + \sum_{k=1}^K \beta_k g(w_{k0} + \sum_{j=1}^p w_{kj} x_{ij}) \quad (17)$$

Formül (16)'taki amaç yeterince basit görünmektedir, ancak parametrelerin iç içe düzenlemesi ve gizli birimlerin simetrisi nedeniyle minimize edilmesi kolay değildir. Problem, parametrelerde konveks değildir ve bu nedenle birden fazla çözüm vardır. Örnek olarak, Formül 13 tek bir değişken θ 'nın basit konveks olmayan bir fonksiyonunu göstermektedir; iki çözüm vardır: biri yerel bir minimum ve diğeri global bir minimumdur. Ayrıca, (formül 1) sinir ağlarının en basit halidir; bu bölümde bu sorunların daha da karmaşık hale geldiği çok daha karmaşık olanları sunulmuştur. Bu sorunların bazılarının üstesinden gelmek ve aşırı uyumdan korumak için, sinir ağları uygulanırken iki genel strateji kullanılmaktadır.

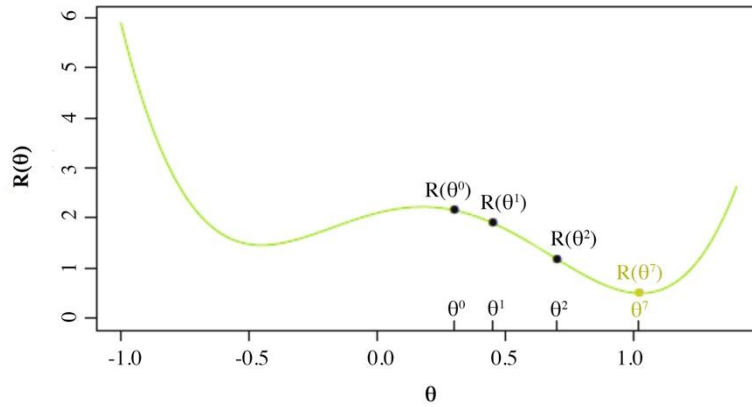
- Yavaş Öğrenme: model, gradyan inişi kullanılarak biraz yavaş bir iteratif şekilde uygulanır. Uygulama süreci, aşırı uyum tespit edildiğinde durdurulur.
- Düzenleme: parametrelere cezalar uygulanır, genellikle lasso veya ridge cezaları kullanılır.

Tüm parametreleri tek bir uzun vektör θ olarak temsil ettiğimizi varsayalım. O zaman Formül (16)'teki amacı şu şekilde yeniden yazabiliriz:

$$R(\theta) = \frac{1}{2} \sum_{i=1}^n (y_i - f_{\theta}(x_i))^2 \quad (18)$$

burada f 'nin parametrelere bağımlılığını açıkça belirtiyoruz. Gradyan inişinin fikri çok basittir.

1. θ 'daki tüm parametreler için bir θ^0 tahmini ile başlayın ve $t = 0$ olarak ayarlayın.
2. Amaç (Formül 18) azalmayı başaramayana kadar tekrarlayın:
 - (a) $\theta^{t+1} = \theta^t + \delta$ ' amacı azaltan θ 'daki küçük bir değişikliği yansıtan bir δ vektörü bulun; yani $R(\theta^{t+1}) < R(\theta^t)$ olacak şekilde.
 - (b) $t \leftarrow t + 1$ olarak ayarlayın.



Şekil 5.6 Gradyan iniş algoritmasının dağlık arazi benzetmesi

Amaç fonksiyonu $R(\theta)$ konveks değildir ve iki minimuma sahiptir, biri ($\theta = -0.46$ 'da (yerel), diğeri ($\theta = 1.02$ 'de (global). Tipik olarak rastgele seçilen bir (θ^0 değerinden

başlayarak, (θ 'daki her adım, daha fazla aşağı inemeyene kadar gradyana karşı yokuş aşağı hareket eder. Bu örnekte, gradyan inişi 7 adımda global minimuma ulaşmıştır.

Bu gösterimde, gradyan inişi yönteminin tek boyutlu bir optimizasyon problemi üzerindeki işleyişi görselleştirilmektedir. Amaç fonksiyonunun konveks olmayan yapısı, optimizasyon sürecinin karmaşıklığını vurgulamaktadır. Başlangıç noktasının seçimi, algoritmanın global veya yerel minimuma ulaşmasında belirleyici rol oynamaktadır. Bu örnek, gradyan inişi yönteminin etkili bir şekilde global optimuma ulaşabildiğini göstermektedir.

Kişi, dağlık bir arazide durup (Şekil 5.6) bir dizi adımla en alt noktaya ulaşmaya çalışıyor gibi düşünebilir. Her adım aşağı doğru olduğu sürece, sonunda en alt noktaya ulaşmalıyız. Bu durumda şanslıydık, çünkü başlangıç tahminimiz θ^0 ile global minimuma ulaştık. Genel olarak, (iyi) bir yerel minimuma ulaşmayı umabiliriz.

5.4.4 Tekrarlamalı Sinir Ağları (RNN)

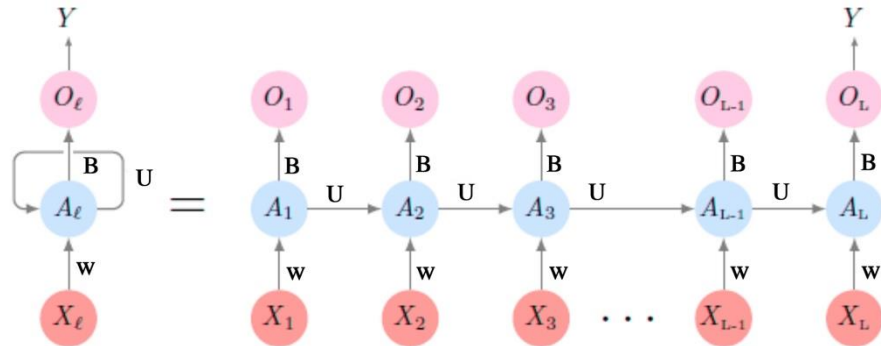
Tekrarlamalı Sinir Ağları (RNN), ardışık veri yapılarını işlemek ve modellemek için tasarlanmış özel bir yapay sinir ağı türüdür. RNN'ler, zaman içinde değişen veya sıralı yapıda olan verileri analiz etmekte ve bu verilerdeki uzun vadeli bağımlılıkları öğrenmekte oldukça etkilidir. Bu özellik, RNN'leri birçok uygulama alanı için ideal bir seçim haline getirmektedir. RNN'lerin kullanım alanları oldukça geniştir ve çeşitli ardışık veri kaynaklarını içerir. Doğal dil işleme alanında, kitap ve film incelemeleri, gazete makaleleri ve sosyal medya gönderileri gibi metinsel veriler üzerinde çalışırken, kelime sırası ve konumları anlamı, temayı ve tonu belirlemede kritik öneme sahiptir. RNN'ler, bu yapıyı modelleyerek konu sınıflandırması, duygu analizi ve dil çevirisi gibi görevlerde kullanılabilir.

Zaman serisi analizi de RNN'lerin önemli bir uygulama alanıdır. Sıcaklık, yağış, rüzgar hızı ve hava kalitesi gibi meteorolojik veriler ile piyasa endeksleri, ticaret hacimleri, hisse senedi ve tahvil fiyatları, döviz kurları gibi finansal veriler zaman serisi özelliği gösterir. RNN'ler, bu verileri kullanarak kısa vadeli hava tahminleri, uzun vadeli iklim

projeksiyonları veya finansal tahminler yapabilir. Ses işleme alanında, RNN'ler kaydedilmiş konuşmalar, müzik kayıtları ve diğer ses verileri üzerinde çalışırken konuşma tanıma, dil çevirisi veya müzik analizi gibi görevlerde kullanılabilir. Ayrıca, el yazısı tanıma alanında, doktor notları veya posta kodları gibi el yazısı metinlerin dijital metne dönüştürülmesi için RNN'ler etkili bir şekilde uygulanabilir.

İşaret dili tanıma bağlamında, RNN'ler özellikle önemlidir. İşaret dili, doğası gereği zamansal ve sıralı bir yapıya sahiptir. El hareketleri, yüz ifadeleri ve vücut duruşları zaman içinde değişir ve bir bütün olarak anlamı oluşturur. RNN'ler, bu zamansal yapıyı modelleyerek işaretlerin anlamını daha doğru bir şekilde yakalayabilir. Bir işaret dili tanıma sisteminde RNN'ler, hareket dizilerini analiz etme, bağlamsal bilgiyi işleme ve sürekli işaret dili çevirisi gibi görevlerde kullanılabilir.

Bir yinelemeli sinir ağında (RNN), girdi nesnesi X bir dizidir. IMDb film incelemeleri koleksiyonu gibi bir belge derlemesi düşünülürse, her belge L kelimedenden oluşan bir dizi olarak temsil edilebilir, bu nedenle $X = \{X_1, X_2, \dots, X_L\}$ olur, burada her X_ℓ bir kelimeyi temsil eder. Kelimelerin sırası ve belirli kelimelerin bir cümledeki yakınlığı anlamsal anlam taşır. RNN'ler, konvolüsyonel sinir ağlarının (CNN'ler) görüntü girdilerinin mekansal yapısını göz önünde bulundurması gibi, bu tür girdi nesnelerinin ardışık doğasını dikkate alacak ve bundan yararlanacak şekilde tasarlanmıştır. Çıktı Y 'de bir dizi olabilir (dil çevirisinde olduğu gibi), ancak genellikle bir skaler olur, örneğin bir film inceleme belgesinin ikili duygu etiketi gibi.



Şekil 5.7 Temel Tekrarlayan Sinir Ağı (RNN) mimarisi ve bilgi akışı

Giriş, $\{X_\ell\}_1^L$ vektör dizisinden oluşmakta olup, burada hedef tek bir yanıttır. Ağ, X giriş dizisini sıralı olarak işlemektedir; her X_ℓ gizli katmana beslenir. Gizli katman ayrıca, dizinin önceki elemanından gelen $A_{\ell-1}$ aktivasyon vektörünü girdi olarak alır ve mevcut aktivasyon vektörü A_ℓ 'yi üretir. Dizin her elemanı işlenirken aynı W , U ve B ağırlık koleksiyonları kullanılmaktadır. Çıktı katmanı, mevcut aktivasyon A_ℓ 'den O_ℓ tahminler dizisini üretir, ancak genellikle bunların yalnızca sonuncusu olan O_L önem taşımaktadır. Eşittir işaretinin solunda ağın özlü bir temsili bulunurken, sağda ise daha açık bir versiyona açılmış hali gösterilmektedir.

Şekil 5.7'te, giriş olarak $X = \{X_1, X_2, \dots, X_L\}$ dizisini, basit bir Y çıktısını ve $\{A_\ell\}_1^L = \{A_1, A_2, \dots, A_L\}$ gizli katman dizisini içeren temel bir Tekrarlayan Sinir Ağı'nın (RNN) yapısı gösterilmektedir. Her X_ℓ bir vektördür; belge örneğinde X_ℓ , korpusun dil sözlüğüne dayalı olarak ℓ 'inci kelimenin one-hot kodlamasını temsil edebilir. Dizi, bir seferde bir X_ℓ vektörü işlenirken, ağ, giriş olarak X_ℓ vektörünü ve dizinin önceki adımından gelen $A_{\ell-1}$ aktivasyon vektörünü alarak gizli katmandaki A_ℓ aktivasyonlarını günceller. Her A_ℓ , çıktı katmanına beslenir ve Y için bir O_ℓ tahmini üretir. Bunların sonuncusu olan O_L en alakalıdır.

Detaylı olarak, giriş dizisinin her X_ℓ vektörünün p bileşeni $X_\ell^T = (X_{\ell 1}, X_{\ell 2}, \dots, X_{\ell p})$ olduğunu ve gizli katmanın K birimden oluştuğunu $A_\ell^T = (A_{\ell 1}, A_{\ell 2}, \dots, A_{\ell K})$ varsayalım. Formül 4'te olduğu gibi, giriş katmanı için $K \times (p+1)$ paylaşılan ağırlıklar w_{kj} koleksiyonu W matrisi ile, benzer şekilde gizliden gizliye katmanlar için $K \times K$ ağırlıklar u_{ks} matrisi U ile ve çıktı katmanı için $K + 1$ ağırlıklar β_k vektörü B ile temsil edilir. Bu durumda:

$$A_{\ell k} = g(w_{k0} + \sum_{j=1}^p w_{kj} X_{\ell j} + \sum_{s=1}^K u_{ks} A_{\ell-1, s}) \quad (12)$$

ve çıktı O_ℓ şu şekilde hesaplanır:

$$O_\ell = \beta_0 + \sum_{k=1}^K \beta_k A_{\ell k} \quad (13)$$

Nicel bir yanıt için, veya örneğin ikili bir yanıt için ek bir sigmoid aktivasyon fonksiyonu ile hesaplanır. Burada $g(\cdot)$, ReLU gibi bir aktivasyon fonksiyonudur. Dizinin her elemanını işlerken aynı W , U ve B ağırlıklarının kullanıldığına, yani bunların ℓ fonksiyonları olmadığına dikkat edilmelidir. Bu, RNN'ler tarafından kullanılan bir ağırlık paylaşımı biçimidir ve evrişimli sinir ağlarındaki filtrelerin kullanımına benzerdir. Baştan sona ilerledikçe, A_ℓ aktivasyonları daha önce görülenlerin bir geçmişini biriktirir, böylece öğrenilen bağlam tahmin için kullanılabilir.

Regresyon problemleri için bir (X, Y) gözleminin kayıp fonksiyonu şu şekildedir:

$$(Y - O_L)^2 \quad (14)$$

Bu fonksiyon yalnızca son çıktı $O_L = \beta_0 + \sum_{k=1}^K \beta_k A_{\ell k}$ 'ya atıfta bulunur. Dolayısıyla $O_1, O_2, \dots, O_{L-1}$ kullanılmaz. Model uyarlandığında, X giriş dizisinin her X_ℓ elemanı, (formül 12) zinciri aracılığıyla O_L 'ye katkıda bulunur ve dolayısıyla kayıp Formül (14) yoluyla paylaşılan parametreler W , U ve B 'nin öğrenilmesine dolaylı olarak katkıda bulunur. n giriş dizisi/yanıt çifti (x_i, y_i) , ile parametreler, kareler toplamının minimize edilmesiyle bulunur:

$$\sum_{i=1}^n (y_i - o_{iL})^2 = \sum_{i=1}^n \left(y_i - \left(\beta_0 + \sum_{k=1}^K \beta_k g(w_{k0} + \sum_{j=1}^p w_{kj} x_{iLj} + \sum_{s=1}^K u_{ks} a_{i,L-1,s}) \right) \right)^2 \quad (15)$$

Burada gözlemlenen y_i ve vektör dizileri $x_i = \{x_{i1}, x_{i2}, \dots, x_{iL}\}$ ile türetilmiş aktivasyonlar için küçük harfler kullanılmıştır. Ara çıktılar O_ℓ kullanılmadığından, neden hiç var oldukları sorulabilir. Öncelikle, O_L 'yi üretmek için gereken aynı çıktı ağırlıkları B 'yi kullandıklarından, bunlar ek bir maliyet olmaksızın elde edilir ve çıktı için gelişen bir tahmin sağlarlar. Ayrıca, bazı öğrenme görevlerinde yanıt da bir dizidir ve bu nedenle çıktı dizisi $\{O_1, O_2, \dots, O_L\}$ açıkça gereklidir.

Tam güçte kullanıldığında, tekrarlayan sinir ağları oldukça karmaşık olabilir. Bunların kullanımını iki basit uygulamada gösterilmektedir. İlkinde, bir önceki IMDb duygu

analizi ele alınmakta ve yorumlardaki kelimeler sıralı olarak işlenmektedir. İkinci uygulamada ise finansal zaman serisi tahmininde kullanımları örneklendirilmektedir.

5.4.4.1 Uzun Kısa Süreli Bellek Ağları (LSTM)

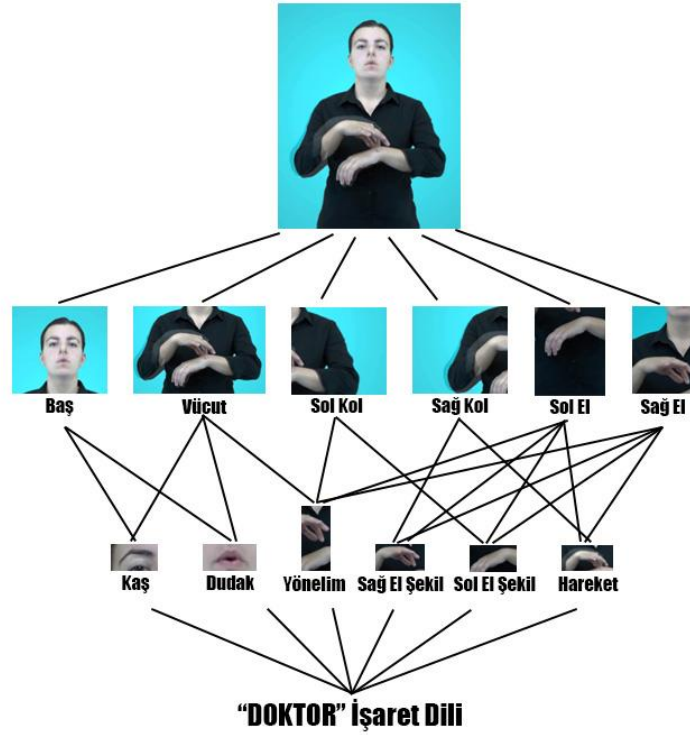
Uzun Kısa Süreli Bellek (LSTM) ağları, Tekrarlayan Sinir Ağlarının (RNN) özel bir türüdür ve uzun vadeli bağımlılıkları öğrenme konusundaki üstün yetenekleriyle öne çıkar. LSTM'ler, 1997 yılında Hochreiter ve Schmidhuber tarafından geliştirilerek, klasik RNN'lerin karşılaştığı gradyan kaybı problemini çözmeyi amaçlamıştır. Bu ağlar, bir hücre durumu ve çeşitli kapılar (giriş, unutma ve çıkış kapıları) içeren karmaşık bir yapıya sahiptir, bu da uzun süreli bilgiyi seçici olarak hatırlama ve unutma yeteneği sağlar. LSTM'ler, özellikle zaman serisi analizi, doğal dil işleme ve konuşma tanıma gibi sıralı veri işleme görevlerinde yaygın olarak kullanılmaktadır. İşaret dili tanıma bağlamında LSTM'ler, el hareketleri ve yüz ifadelerinin zamansal özelliklerini modellemede etkilidir. Bu ağlar, bir işaretin başlangıcından sonuna kadar olan hareket dizisini analiz ederek, işaretin bütünsel anlamını yakalamada başarılıdır. Türk İşaret Dili tanıma çalışmalarında LSTM'ler, işaretlerin zamansal yapısını öğrenmek için kullanılabilir. Örneğin, bir işaretin farklı aşamalarını (başlangıç, orta ve bitiş hareketleri) modelleyerek, işaretin tam anlamını çıkarabilirler. Ayrıca, LSTM'ler cümle düzeyinde işaret dili çevirisinde de etkili olabilir, çünkü önceki işaretlerin bağlamını hatırlayarak daha doğru çeviriler yapabilirler.

5.4.5 Evrişimsel Sinir Ağları (CNN)

Evrişimsel Sinir Ağları (CNN), derin öğrenme metodolojileri arasında özellikle görüntü işleme ve bilgisayarla görme alanlarında öne çıkan bir modeldir. CNN'ler, karmaşık sorunların çözümünde geleneksel makine öğrenimi yaklaşımlarına kıyasla daha üstün bir performans sergilemektedir (Indolia ve ark. 2018). Bu ağlar, alana özgü özellikleri otomatik olarak çıkarma yetenekleri sayesinde, bilgisayarla görme alanındaki tüm yaklaşımların yeniden değerlendirilmesine yol açmıştır. CNN'ler, ImageNet yarışması ve görüntü sınıflandırma alanındaki başarılarıyla dikkatleri üzerine çekmiştir. ConvNet olarak da bilinen bu ağlar, derin ileri beslemeli mimarileri ve tam bağlı katmanlı ağlara kıyasla daha iyi genelleme yetenekleriyle tanınmaktadır (Nebauer 1998). CNN'lerin

tasarımı, biyolojik prensiplerden esinlenen hiyerarşik özellik tanımlama konseptine dayanmaktadır (Fieres vd. 2006). Bu yapılar, karmaşık kavramsal özellikleri öğrenme ve çeşitli varlıkları tanımlama konusunda oldukça başarılıdır. 2010 yılı civarında, giderek artan sayıda sınıf içeren büyük etiketli görüntü veritabanlarının oluşturulmasıyla, CNN'ler görüntü sınıflandırma alanında yeniden ön plana çıkmıştır. Örneğin, CIFAR100 veritabanı, 20 üst sınıfa göre etiketlenmiş 60.000 görüntüden oluşmakta ve her görüntü 32×32 piksel çözünürlüğüne sahiptir. Bu tür veritabanları, CNN'lerin eğitilmesi ve test edilmesi için zengin bir kaynak sağlamıştır.

CNN'ler, insan görme sistemini taklit ederek çalışır. Bu ağlar, her bir nesne sınıfını ayırt eden spesifik özellikleri veya desenleri görüntünün herhangi bir yerinde tanıyabilir. CNN'lerin yapısı genellikle evrişim katmanları, havuzlama katmanları ve tam bağlı katmanlardan oluşur. Evrişim katmanları görüntüdeki özellikleri tespit eder, havuzlama katmanları boyut azaltımı yapar ve tam bağlı katmanlar ise sınıflandırma işlemini gerçekleştirir. İşaret dili tanıma bağlamında CNN'ler, el şekillerinin ve yüz ifadelerinin statik özelliklerini yakalamada oldukça etkilidir. Örneğin, bir işaretin belirli bir anındaki el pozisyonunu veya parmak konfigürasyonunu tanımlamada kullanılabilirler. Türk İşaret Dili tanıma çalışmalarında CNN'ler, işaretlerin görsel özelliklerini öğrenmek için kullanılabilir. Şekil 5.8, bir işaretçi karikatür görüntüsü üzerinde evrişimli bir sinir ağının arkasındaki fikri göstermektedir.

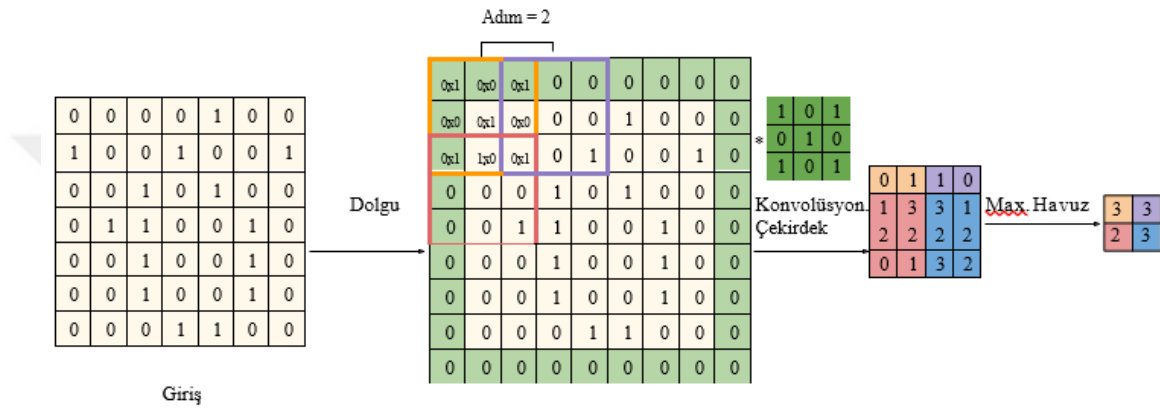


Şekil 5.8 Evrişimli sinir ağının işaret dili görüntüsünü sınıflandırma süreci

Şekil 5.8 Bir evrişimli sinir ağının bir işaretçi görüntüsünü nasıl sınıflandırdığını gösteren şematik diagram. Ağ, görüntüyü alır ve yerel özellikleri tanımlar. Ardından, bu örnekte kaşları, elleri, hareketleri gibi içeren bileşik özellikler oluşturmak için yerel özellikleri birleştirir. Bu bileşik özellikler, "DOKTOR" İşaret dili etiketini çıktı olarak vermek için kullanılır.

Evrişimsel Sinir Ağları (CNN'ler), görüntülerdeki özellikleri hiyerarşik bir yapıda öğrenir. Bu süreç, düşük seviyeli özelliklerden (kenarlar, renk lekeleri gibi) başlayarak yüksek seviyeli özelliklere (kulaklar, gözler gibi) doğru ilerler. Sonuç olarak, bu yüksek seviyeli özelliklerin varlığı veya yokluğu, herhangi bir verilen çıktı sınıfının olasılığını belirler. CNN'lerin temel yapı taşları, evrişim katmanları ve havuzlama katmanlarıdır. Evrişim katmanları, görüntüdeki küçük desenleri arar ve özellik haritaları oluşturur. Havuzlama katmanları ise bu özellik haritalarını örnekleyerek boyutunu küçültür ve önemli bilgileri seçer. Modern CNN mimarileri, yüksek performans elde etmek için birçok evrişim ve havuzlama katmanı kullanır. CNN'lerin tarihsel gelişimi önemlidir.

LeCun ve arkadaşları tarafından geliştirilen LeNet, el yazısı tanıma için tasarlanan ilk başarılı CNN'lerden biridir. Bir süre durgunluk yaşayan derin öğrenme alanı, veri kümelerinin ve donanımların gelişmesiyle yeniden canlanmıştır. Krizhevsky ve diğerleri tarafından sunulan AlexNet mimarisi, CNN'lerin popülerliğini önemli ölçüde artırmıştır. Evrişim ve havuzlama katmanları bir sonraki bölümde açıklanmaktadır. Evrişimsel sinir ağlarının (CNN'ler), Şekil 5.9'da gösterildiği gibi, en az bir katmanında evrişim işlemi uygulanan özel bir sinir ağı türü olduğu belirtilmektedir.



Şekil 5.9 Evrişimsel sinir ağının (CNN) temel yapısı

CNN'lerin tercih edilmesinin birkaç önemli nedeni vardır. Ağırlık paylaşımı sayesinde eğitilmesi gereken parametre sayısını azaltır, bu da ağıın genelleme yeteneğini artırır (Arel ve ark. 2010). Parametre sayısının azalması, aşırı uyum riskini düşürerek CNN'in daha etkili eğitilmesini sağlar (Smirnov vd. 2014).

Tipik bir CNN mimarisi şu katmanlardan oluşur:

- **Evrişimsel Katmanlar:** Farklı özellikler üreten ara özellik haritaları oluşturmak için birden fazla çekirdek kullanır.
- **Havuzlama Katmanları:** Özellik haritalarının boyutunu küçültür ve veri temsilini basitleştirir.
- **Tam Bağlı Katmanlar:** Mimarinin sonunda yer alır ve sınıflandırıcı görevi görerek görüntülerin etkili bir şekilde sınıflandırılmasını sağlar.

İşaret dili tanıma bağlamında CNN'ler, statik el şekillerini ve yüz ifadelerini tanımada özellikle etkilidir. Örneğin, bir işaretin belirli bir anındaki el pozisyonunu veya parmak konfigürasyonunu yüksek doğrulukla tespit edebilirler. Ancak, işaret dilinin dinamik doğası gözönüne alındığında, CNN'lerin tek başına yeterli olmayabileceği ve RNN veya LSTM gibi zamansal bilgiyi işleyebilen ağlarla birlikte kullanılması gerekebileceği unutulmamalıdır.

5.4.5.1 Evrişimsel sinir ağının mimarisi

Evrişimsel Sinir Ağları (CNN), temel olarak üç tür katmandan oluşur: evrişimsel, havuzlama ve tam bağlı katmanlar. Bu yapı, görüntü işleme ve sınıflandırma görevlerinde yüksek performans elde etmek için tasarlanmıştır. Evrişimsel katman, CNN'lerin temelini oluşturur ve eğitim yoluyla öğrenilecek filtreleri içerir. Bu filtreler genellikle giriş görüntüsünden daha küçük boyutludur ve tüm görüntü üzerinde evrişim işlemi uygulayarak sonuçları bir aktivasyon haritasına dönüştürür. Evrişimsel katmanı genellikle doğrusal olmayan bir aktivasyon fonksiyonu takip eder. Bu doğrusal olmama özelliği, ağın daha karmaşık ve çeşitli fonksiyonları tahmin etmesini sağlar.

Havuzlama katmanları, yakın çıktıları özetleyerek öğrenilecek parametre sayısını ve hesaplamaları azaltır. Maksimum havuzlama ve ortalama havuzlama, yaygın olarak kullanılan havuzlama yöntemleridir. Bu katmanlar, özellik haritalarının boyutunu küçülterek hem hesaplama yükünü azaltır hem de modelin uzamsal değişimlere karşı daha dayanıklı olmasını sağlar. CNN mimarisi, Özellik Öğrenme ve Sınıflandırma olmak üzere iki ana bölüme ayrılabilir (Che vd. 2019). Özellik Öğrenme bölümü, Evrişim katmanı, Doğrultucu Doğrusal Birim (ReLU) ve Maksimum Havuzlama'dan oluşurken, Sınıflandırma katmanı Düzleştirme, Tam Bağlantılı ve Softmax katmanlarını içerir.

Tam bağlı katmanlar çoğunlukla ağın sonunda kullanılır ve bir katmandaki tüm nöronları bir sonraki katmandaki tüm nöronlara bağlar. Bu katmanlar, önceki katmanlardan çıkarılan özellikleri kullanarak son sınıflandırma kararını verir. CNN'lerin yapısı, girdi verilerinin işlenmesini kademeli olarak gerçekleştirir. Örneğin, renkli bir görüntü için giriş katmanı, her rengi 32×32 boyutunda iki boyutlu piksel özellik haritası ile temsil

eden üç boyutlu bir yapıya sahiptir. İlk evrişim katmanı, bu giriş üzerinde filtreleri uygulayarak yeni kanallar oluşturur. Bu işlem, orijinal üç renkli kanaldan çok daha fazla kanala sahip yeni bir "görüntü" oluşturur. Ardından gelen havuzlama katmanı, bu özellik haritasının boyutunu her kanalda azaltır, böylece veri temsili daha kompakt hale getirir.

CNN'lerin görüntü işlemede önemli avantajları vardır. Özellikle, görsel girdideki özelliklerin temsili otomatik olarak ve etkili bir şekilde öğrenilebilmeleri, onları güçlü kılmaktadır (Shiva vd. 2015). Bu nedenle, son yıllarda CNN'lerin sınıflandırma performansını artırmaya yönelik araştırmalar yoğun bir şekilde devam etmektedir. İşaret dili tanıma bağlamında CNN'ler, el şekillerinin ve yüz ifadelerinin statik özelliklerini yakalamada özellikle etkilidir. Ancak, işaret dilinin dinamik doğası göz önüne alındığında, CNN'lerin zamansal bilgiyi işleyebilen ağlarla (örneğin, RNN veya LSTM) birlikte kullanılması genellikle daha iyi sonuçlar vermektedir.

Evrişimsel Sinir Ağları (CNN), görüntü işleme ve sınıflandırma görevlerinde yüksek performans elde etmek için tasarlanmış özel bir yapay sinir ağı türüdür. CNN'lerin mimarisi, temel olarak evrişimsel, havuzlama ve tam bağlı katmanlardan oluşur. Bu katmanların birbirleriyle etkileşimi ve tekrarlanması, CNN'lerin karmaşık görsel özellikleri öğrenmesini sağlar. Evrişimsel katman, CNN'lerin temelini oluşturur. Bu katman, önceki katmandan gelen üç boyutlu özellik haritasını girdi olarak alır ve onu tek bir çok kanallı görüntü gibi işler. Her evrişim filtresi, giriş özellik haritası kadar çok kanala sahiptir ve tüm görüntü üzerinde evrişim işlemi uygulayarak yeni bir aktivasyon haritası oluşturur. Evrişimsel katmanı genellikle doğrusal olmayan bir aktivasyon fonksiyonu takip eder, bu da ağı daha karmaşık fonksiyonları öğrenmesini sağlar.

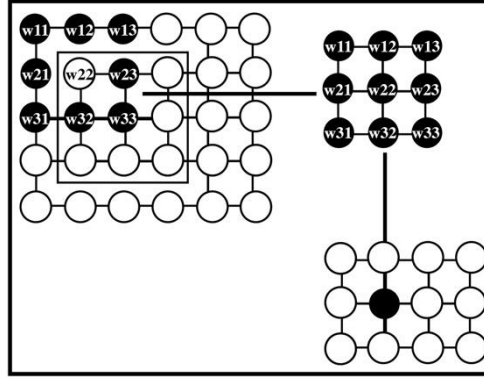
Havuzlama katmanları, özellik haritalarının boyutunu küçülterek hem hesaplama yükünü azaltır hem de modelin uzamsal değişimlere karşı daha dayanıklı olmasını sağlar. Her havuzlama katmanından sonra kanal özellik haritaları boyut olarak küçüldüğünden, genellikle bunu telafi etmek için bir sonraki evrişim katmanındaki filtre sayısı artırılır. Bu, ağı daha fazla ve daha karmaşık özellikler öğrenmesine olanak tanır. CNN mimarisinde, evrişim-ardından-havuzlama sekansı genellikle birkaç kez tekrarlanır. Bazen bir havuzlama katmanından önce birkaç evrişim katmanı art arda kullanılır. Bu

yaklaşım, filtrenin efektif boyutunu artırarak daha geniş bir reseptif alan elde edilmesini sağlar. Bu işlemler, havuzlama her kanal özellik haritasını her boyutta sadece birkaç piksele indirgeyene kadar devam eder. Son aşamada, üç boyutlu özellik haritaları düzleştirilir yani pikseller ayrı birimler olarak ele alınır. Bu düzleştirilmiş veri, çıktı katmanına ulaşmadan önce bir veya daha fazla tam bağlantılı katmana beslenir. Tam bağlı katmanlar, önceki katmanlardan çıkarılan özellikleri kullanarak son sınıflandırma kararını verir. Çıktı katmanı genellikle, sınıf sayısı kadar nörona sahip bir softmax aktivasyonu içerir.

İşaret dili tanıma bağlamında, bu mimari yapı özellikle önemlidir. Evrişimsel katmanlar, el şekillerinin ve yüz ifadelerinin yerel özelliklerini yakalarken, havuzlama katmanları bu özelliklerin uzamsal değişimlere karşı daha dayanıklı olmasını sağlar. Tam bağlı katmanlar ise bu özellikleri bir araya getirerek final sınıflandırma kararını verir.

5.4.5.2 Evrişim katmanı

Evrişimsel Sinir Ağları (CNN), derin öğrenme metodolojileri arasında özellikle karmaşık görsel problemlerin çözümünde öne çıkan bir modeldir. CNN'ler, geleneksel makine öğrenimi yaklaşımlarına kıyasla daha yüksek performans sergilemekte ve alana özgü özellikleri otomatik olarak çıkarabilme yetenekleriyle dikkat çekmektedir (Indolia ve ark. 2018). Bu özellikleri sayesinde CNN'ler, bilgisayarla görme alanında birçok yaklaşımın yeniden değerlendirilmesine yol açmış ve ImageNet yarışması gibi prestijli görüntü sınıflandırma yarışmalarında üstün başarılar elde etmiştir. CNN'lerin, ConvNet olarak da bilinen derin ileri beslemeli mimarileri, tam bağlı katmanlı ağlara kıyasla daha iyi genelleme yeteneği göstermektedir (Nebauer 1998). Bu mimarinin tasarımı, biyolojik sistemlerden esinlenen hiyerarşik özellik tanımlama konseptine dayanmaktadır (Fieres vd. 2006). Bu yapı, CNN'lerin karmaşık kavramsal özellikleri öğrenme ve çeşitli varlıkları tanımlama konusunda oldukça başarılı olmasını sağlamaktadır (Indolia vd. 2018).



Şekil 5.10 Sinir ağında bir sonraki katmandaki alıcı nöron alanının gösterimi

Evrişim katmanı, CNN'lerin temel yapı taşıdır ve ağıın özellik çıkarma yeteneğinin merkezinde yer alır. Bu katman, her biri bir görüntüde belirli bir yerel özelliğın varlığını belirleyen bir şablon olan çok sayıda evrişim filtresinden oluşur. Evrişim işlemi, temel olarak matris elemanlarının tekrarlı çarpımı ve sonuçların toplanmasına dayanan basit bir matematiksel operasyondur.

Bir evrişim filtresinin nasıl çalıştığını anlamak için, 4×3 boyutunda çok basit bir görüntü örneğı ele alınabilir:

$$\text{Original Image} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \\ j & k & l \end{bmatrix}$$

Şimdi aşağıdaki formda bir 2×2 filtre düşünölmelidir:

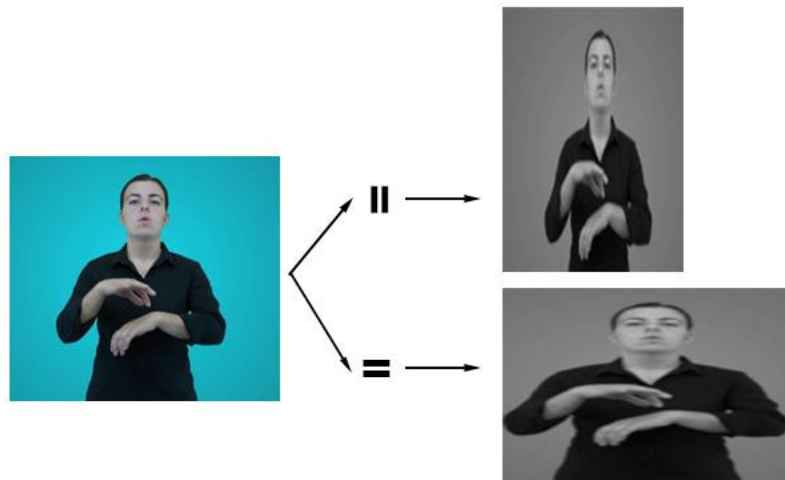
$$\text{Convolution Filter} = \begin{bmatrix} \alpha & \beta \\ \gamma & \delta \end{bmatrix}$$

Görüntü bu filtre ile evriştirildiğinde, aşağıdaki sonuç elde edilir:

$$\text{Convolved Image} = \begin{bmatrix} a\alpha + b\beta + d\gamma + e\delta & b\alpha + c\beta + e\gamma + f\delta \\ d\alpha + e\beta + g\gamma + h\delta & e\alpha + f\beta + h\gamma + i\delta \\ g\alpha + h\beta + j\gamma + k\delta & h\alpha + i\beta + k\gamma + l\delta \end{bmatrix}$$

Evriřim iřleminin alıřma prensibi řu řekildedir: rneęin, 2x2 boyutunda bir evriřim filtresi dūřunelim. Bu filtre, giriř grnts zerinde kaydırılır ve her adımda, filtrenin kapsadıęı alandaki piksel deęerleri ile filtrenin elemanları arpılıp toplanır. Bu iřlem sonucunda, orijinal grntnn filtreye benzeyen blgeleri vurgulanmıř olur. Eęer grntnn bir blgesi evriřim filtresine benziyorsa, evriřtirilmiř grntde o blge yksek bir deęere sahip olacaktır; aksi takdirde dřk bir deęer alacaktır. Genel olarak, evriřim filtreleri kk boyutlu matrislerdir (rneęin 3x3, 5x5 gibi). Bu kk boyutlu filtreler, aęın daha az parametre ile alıřmasını ve dolayısıyla daha etkili ęrenmesini saęlar. Ayrıca, bu yaklařım aęın uzamsal hiyerarřileri ęrenmesine olanak tanır: ilk katmanlardaki filtreler basit kenarları ve dokuları yakalarken, daha derin katmanlardaki filtreler daha karmařık ve soyut zellikleri ęrenir.

řekil 5.11, sol tarafta gsterilen 192×179 boyutundaki bir iřareti grntsne iki evriřim filtresinin uygulanmasını gstermektedir. Her evriřim filtresi, oęunlukla sıfırlardan (siyah) oluřan, grnt iinde dikey veya yatay olarak ynlendirilmiř dar bir birler (beyaz) řeridi ieren 15×15 boyutunda bir grntdr. Her filtre iřareti grnts ile evriřtirildięinde, filtreye benzeyen iřareti alanlarına (yani yatay veya dikey izgiler veya kenarlar ieren) byk deęerler verilir ve zellięe benzemeyen iřareti alanlarına kk deęerler verilir. Evriřtirilmiř grntler saę tarafta gsterilmektedir. Yatay izgi filtresinin orijinal grntdeki yatay izgileri ve kenarları setięi, dikey izgi filtresinin ise orijinal grntdeki dikey izgileri ve kenarları setięi grlmektedir.



řekil 5.11 Evriřim filtrelerinin iřaret dili grnts zerindeki etkisi

Şekil 5.11 Evrişim filtreleri, bir görüntüdeki kenarlar ve küçük şekiller gibi yerel özellikleri bulur. Solda gösterilen işaretçi görüntüsü ile başlanır ve ortadaki iki küçük evrişim filtresi uygulanır. Evriştirilmiş görüntüler, orijinal görüntüde filtrelere benzer detayların bulunduğu alanları vurgular. Özellikle, üstteki evriştirilmiş görüntü işaretçinin dikey çizgilerini vurgularken, alttaki evriştirilmiş görüntü işaretçinin yatay çizgilerini vurgular. Orijinal görüntü, bir evrişimli sinir ağındaki girdi katmanı olarak, evriştirilmiş görüntüler ise ilk gizli katmandaki birimler olarak düşünülebilir.

Bir evrişim katmanında, görüntüdeki çeşitli yönelimli kenarları ve şekilleri seçmek için bir dizi filtre kullanılır. Bu şekilde önceden tanımlanmış filtrelerin kullanılması, görüntü işlemede standart bir uygulamadır. Buna karşılık, CNN'lerde filtreler belirli sınıflandırma görevi için öğrenilir. Filtre ağırlıkları, bir girdi katmanından bir gizli katmana giden parametreler olarak düşünülebilir; evriştirilmiş görüntüdeki her piksel için bir gizli birim bulunur. Bu gerçekten de böyledir, ancak parametreler oldukça yapılandırılmış ve kısıtlanmıştır. Bunlar, girdi görüntüsündeki yerleştirilmiş yamalar üzerinde çalışır ve belirli bir filtredeki aynı ağırlıklar, görüntüdeki tüm olası yamalar için yeniden kullanılır.

Şimdi bazı ek ayrıntılar verilecektir:

- Girdi görüntüsü renkli olduğundan, üç boyutlu bir özellik haritası (dizi) ile temsil edilen üç kanala sahiptir. Her kanal iki boyutlu (32×32) bir özellik haritasıdır - biri kırmızı, biri yeşil ve biri mavi için. Tek bir evrişim filtresi de her renk için bir tane olmak üzere üç kanala sahip olacaktır, her biri 3×3 boyutunda ve potansiyel olarak farklı filtre ağırlıklarına sahiptir. Üç evrişimin sonuçları, iki boyutlu bir çıktı özellik haritası oluşturmak üzere toplanır. Bu noktada renk bilgisinin kullanıldığına ve evrişimdeki rolü dışında sonraki katmanlara aktarılmadığına dikkat edilmelidir.

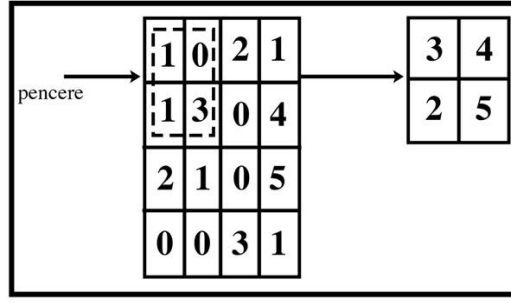
- Bu ilk gizli katmanda K farklı evrişim filtresi kullanılırsa, birlikte tek bir üç boyutlu özellik haritası olarak ele alınan K adet iki boyutlu çıktı özellik haritası elde edilir. K çıktı özellik haritasının her biri ayrı bir bilgi kanalı olarak görülür, böylece orijinal girdi özellik haritasındaki üç renk kanalının aksine şimdi K kanalımız vardır. Üç boyutlu özellik

haritası, basit bir sinir ağının gizli katmanındaki aktivasyonlar gibidir, ancak mekansal olarak yapılandırılmış bir şekilde düzenlenmiş ve üretilmiştir.

- Genellikle evriştirilmiş görüntüye ReLU aktivasyon fonksiyonu uygulanır. Bu adım bazen evrişimli sinir ağında ayrı bir katman olarak görülür ve bu durumda dedektör katmanı olarak adlandırılır.

5.4.5.3 Havuzlama katmanı

Havuzlama katmanı, Evrişimsel Sinir Ağları'nın (CNN) önemli bir bileşenidir ve genellikle evrişimsel katmanı takip eder. Bu katmanın temel amacı, özellik haritasının boyutunu küçültmek, aşırı uyumu azaltmak ve hesaplama sürecini hızlandırmaktır. LeCun ve ark. (1998)'e göre, belirli bir özellik belirlendikten sonra, bu özelliğin tam konumu daha az önem taşımaktadır. Bu nedenle, havuzlama işlemi, özellik haritalarını özetleyerek ağın uzamsal değişimlere karşı daha dayanıklı olmasını sağlar. Havuzlama işlemi, bir pencere seçilerek ve bu pencere içindeki giriş öğelerine bir havuzlama fonksiyonu uygulanarak gerçekleştirilir (Indolia vd. 2018). Bu işlem, tüm özellik haritası alanını alternatif olarak kaydırarak uygulanır. Havuzlama tekniğinin kullanılmasının temel avantajları arasında eğitilebilir parametre sayısında önemli bir azalma ve öteleme değişmezliğinin sağlanması yer almaktadır (Zhou vd. 2016). Farklı havuzlama stratejileri mevcuttur, ancak en yaygın kullanılanları maksimum havuzlama ve ortalama havuzlamadır. Maksimum havuzlama, belirli bir boyuttaki filtreden en yüksek değeri seçerek çalışır (Giusti vd. 2013). Örneğin, 2x2 boyutunda bir maksimum havuzlama işlemi, giriş görüntüsündeki örtüşmeyen her 2x2 piksel bloğunu, bloktaki maksimum değeri kullanarak özetler. Bu işlem, görüntünün boyutunu her yönde iki kat azaltır ve aynı zamanda bir miktar konum değişmezliği sağlar. Havuzlama katmanının bir gösteriminin Şekil 5.12'de sunulduğu belirtilmektedir.



Şekil 5.12 2x2 pencere kullanan maksimum havuzlama örneği

Bir havuzlama katmanı, büyük bir görüntüyü daha küçük bir özet görüntüye yoğunlaştırmanın bir yolunu sağlar. Havuzlama işlemini gerçekleştirmek için birçok olası yöntem bulunmakla birlikte, maksimum havuzlama işlemi, bir görüntüdeki örtüşmeyen her 2×2 piksel bloğunu, bloktaki maksimum değeri kullanarak özetler. Bu işlem, görüntünün boyutunu her yönde iki kat azaltır ve aynı zamanda bir miktar konum değişmezliği sağlar: yani, bloktaki dört pikselden birinde büyük bir değer olduğu sürece, tüm blok küçültülmüş görüntüde büyük bir değer olarak kaydedilir.

İşte maksimum havuzlamanın basit bir örneği:

$$\text{Max pool} \begin{bmatrix} 1 & 2 & 5 & 3 \\ 3 & 0 & 1 & 2 \\ 2 & 1 & 3 & 4 \\ 1 & 1 & 2 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 3 & 5 \\ 2 & 4 \end{bmatrix}$$

Maksimum havuzlama, harita boyutunu önemli ölçüde azaltması nedeniyle sıklıkla tercih edilmektedir (Lee vd. 2017). Ancak, maksimum havuzlamada hatalar hesaplanırken, ileri yayılımı yer almayan değerlerin hata gradyanlarının geri iletilmediği unutulmamalıdır (Indolia vd. 2018). Bu durum, bazı durumlarda bilgi kaybına neden olabilir.

5.4.5.4 Tam bağlantılı katman

Tam bağlantılı katman, Evrişimsel Sinir Ağları'nın (CNN) son aşamalarında yer alan ve genellikle sınıflandırma görevini yerine getiren önemli bir bileşendir. Bu katman, evrişim

ve havuzlama işlemleri sonucunda elde edilen çıktıyı alır ve geleneksel yapay sinir ağlarında olduğu gibi işler. Tam bağlantılı katmanın temel işlevi, önceki katmanlardan gelen özellik haritalarını düzleştirerek tek bir vektör haline getirmek ve bu vektörü kullanarak son sınıflandırma kararını vermektir. Tam bağlantılı katmanda hesaplama, ağırlık vektörü ile girdi vektörü arasında bir nokta çarpımı işlemi gerçekleştirilerek yapılır (Zhou vd. 2016). Bu işlem, her bir girdinin tüm çıktı nöronlarına bağlı olduğu anlamına gelir. Bu yoğun bağlantı yapısı, katmanın "tam bağlantılı" olarak adlandırılmasının nedenidir.

Tam bağlantılı katmanın eğitimi sırasında, gradyan inişi olarak da bilinen toplu mod öğrenme yöntemi sıklıkla kullanılır. Bu yöntem, tüm eğitim veri kümesi üzerinden maliyetleri hesaplayarak maliyet fonksiyonunu minimize etmeyi amaçlar. Toplu mod öğrenmede, veri kümesinin tamamı işlendikten sonra parametreler bir dönem (epoch) içerisinde güncellenir. Bu yaklaşım, global minimuma ulaşmada etkili olsa da, büyük veri kümeleri söz konusu olduğunda eğitim sürelerinin uzamasına neden olabilir. Eğitim süresini kısaltmak ve büyük veri kümeleriyle daha etkili çalışabilmek için, stokastik gradyan inişi yöntemi geliştirilmiştir. Bu yöntem, maliyet fonksiyonunu minimize etmenin alternatif bir yoludur ve her bir eğitim örneği veya küçük örnek grupları (mini-batch) üzerinde çalışarak parametreleri daha sık günceller.

5.5 Derin Öğrenme Sorunları ve Zorlukları

Derin öğrenme, birçok alanda etkileyici sonuçlar elde etmesine rağmen, özellikle işaret dili tanıma gibi özel alanlarda bazı önemli zorluklarla karşı karşıyadır. Bu zorluklar, veri ihtiyacı, model yorumlanabilirliği ve hesaplama kaynakları olarak üç ana başlık altında incelenebilir.

5.5.1 Veri ihtiyacı

Derin öğrenme modelleri, genellikle büyük miktarda etiketli veriye ihtiyaç duyar. Bu, işaret dili tanıma alanında önemli bir zorluk oluşturur çünkü:

1. Kaliteli ve çeşitli işaret dili veri setleri oluşturmak zaman alıcı ve maliyetlidir.

2. İşaret dilinin bölgesel farklılıkları, standart bir veri seti oluşturmayı zorlaştırır.
3. Nadir kullanılan işaretler için yeterli veri toplamak zor olabilir.

Goodfellow vd. (2016), derin öğrenme modellerinin performansının genellikle veri miktarı arttıkça iyileştiğini belirtmektedir. Ancak, işaret dili gibi özel alanlarda büyük veri setleri oluşturmak her zaman mümkün olmayabilir.

5.5.2 Model yorumlanabilirliği

Derin öğrenme modelleri genellikle "kara kutu" olarak nitelendirilir, çünkü iç işleyişleri karmaşıktır ve kararlarını nasıl verdiklerini anlamak zordur. Bu durum, işaret dili tanıma gibi hassas uygulamalarda sorun oluşturabilir:

1. Modelin neden belirli bir işareti yanlış tanıdığını anlamak zor olabilir.
2. Modelin güvenilirliğini değerlendirmek ve hataları düzeltmek zorlaşır.
3. Yasal ve etik kaygılar ortaya çıkabilir, özellikle modelin kararları önemli sonuçlara yol açıyorsa.

Lipton (2018), model yorumlanabilirliğinin derin öğrenmenin yaygın kullanımı için kritik bir faktör olduğunu vurgulamaktadır.

5.5.3 Hesaplama kaynakları

Derin öğrenme modelleri, özellikle eğitim aşamasında, yüksek hesaplama gücü gerektirir. Bu durum, işaret dili tanıma gibi gerçek zamanlı uygulamalarda zorluklar yaratabilir:

1. Yüksek performanslı donanım ihtiyacı maliyetleri artırır.
2. Büyük modellerin mobil cihazlarda veya gömülü sistemlerde çalıştırılması zor olabilir.

Strubell vd. (2019), büyük derin öğrenme modellerinin eğitiminin önemli miktarda enerji tükettiğini ve karbon ayak izi yarattığını göstermiştir.

5.5.4 Aşırı öğrenme (Overfitting) ve Eksik öğrenme (Underfitting)

Derin öğrenme modellerinin karşılaştığı önemli zorluklardan biri de aşırı öğrenme ve eksik öğrenme sorunlarıdır. Bu iki kavram, modelin eğitim verilerine nasıl uyum sağladığı ve genelleme yeteneği ile ilgilidir. Aşırı öğrenme (Overfitting), modelin eğitim verilerine çok iyi uyum sağlaması, ancak yeni, görülmemiş verilerde kötü performans göstermesi durumudur. Bu durumda model, eğitim verilerindeki gürültüyü ve rastgele dalgalanmaları da öğrenmiş olur. İşaret dili tanıma bağlamında, aşırı öğrenme şu sorunlara yol açabilir:

1. Model, eğitim setindeki belirli kişilerin işaret stillerine aşırı uyum sağlayabilir.
2. Arka plan, ışıklandırma gibi ilgisiz detayları öğrenebilir.
3. Nadir görülen işaretleri yanlış sınıflandırabilir.

Hawkins (2004), aşırı öğrenmenin özellikle karmaşık modeller ve sınırlı veri setleri ile çalışırken sık karşılaşılan bir sorun olduğunu belirtmektedir.

Öte yandan, eksik öğrenme (Underfitting), modelin veri setindeki temel yapıyı ve ilişkileri yakalayamaması durumudur. Bu durumda model, hem eğitim verilerinde hem de yeni verilerde düşük performans gösterir. İşaret dili tanıma için eksik öğrenme şu sorunlara neden olabilir:

1. Model, farklı işaretler arasındaki ince farkları ayırt edemeyebilir.
2. Karmaşık el hareketlerini ve ifadeleri doğru bir şekilde tanımlayamayabilir.
3. İşaretlerin zamansal boyutunu yeterince modelleyemeyebilir.

Goodfellow ve ark. (2016), eksik öğrenmenin genellikle modelin yeterince karmaşık olmaması veya yeterli süre eğitilmemesi durumunda ortaya çıktığını vurgulamaktadır.

Bu sorunların üstesinden gelmek için çeşitli teknikler kullanılmaktadır:

1. Regularizasyon teknikleri (L1, L2 regularizasyonu, Dropout) aşırı öğrenmeyi azaltmaya yardımcı olur (Srivastava vd. 2014).
2. Çapraz doğrulama, modelin genelleme yeteneğini değerlendirmek için kullanılır (Kohavi 1995).
3. Erken durdurma, aşırı öğrenmeyi önlemek için eğitimi optimal noktada sonlandırır (Prechelt 1998).
4. Model karmaşıklığının artırılması veya azaltılması, sırasıyla eksik öğrenme veya aşırı öğrenme sorunlarını çözmeye yardımcı olabilir.

İşaret dili tanıma özelinde, bu teknikler şu şekilde uygulanabilir:

1. Veri artırma teknikleri kullanarak eğitim setini genişletmek ve çeşitlendirmek.
2. Transfer öğrenme ile önceden eğitilmiş modelleri kullanarak, sınırlı veri setleriyle bile iyi performans elde etmek.
3. Model mimarisini işaret dilinin özelliklerine göre optimize etmek.

5.6 Hiper Parametre Seçimleri

Derin öğrenme modellerinin başarısı, büyük ölçüde hiper parametrelerin doğru seçimine bağlıdır. Hiper parametreler, model eğitimi başlamadan önce belirlenen ve eğitim süreci boyunca sabit kalan değerlerdir. Bu parametrelerin dikkatli bir şekilde seçilmesi ve ayarlanması, modelin performansını önemli ölçüde etkileyebilir. Öğrenme oranı, en kritik hiper parametrelerden biridir. Bu parametre, ağırlıklardaki değişimin büyüklüğünü belirler. Eğitim sırasında öğrenme oranını kademeli olarak düşürmek, aşırı uyumu (overfitting) önlemek için yaygın bir uygulamadır. Aşırı uyum, modelin eğitim verisine fazla uzmanlaşması ve dolayısıyla görülmeyen verilere genelleme yapma yeteneğini kaybetmesi sorunudur. Düzenleştirme (regularization), aşırı uyumu önlemeye yardımcı olan başka bir tekniktir. Bu teknik, eğitim aşamasında ağırlıkların büyük değerler almasını cezalandırarak modelin daha iyi genelleme yapmasını sağlar. PyTorch gibi derin öğrenme çerçevelerinde, düzenleştirme genellikle ağırlık azaltma (weight decay) hiper parametresi ile kontrol edilir.

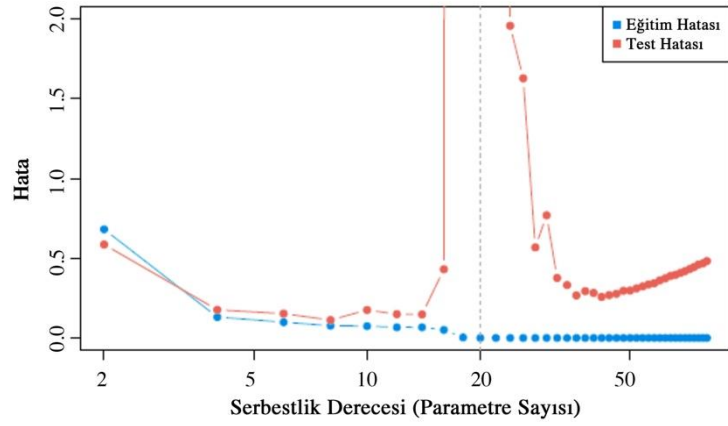
Modelin mimarisi ile ilgili hiper parametre seçimleri de performansı önemli ölçüde etkiler:

1. Gizli katmanların sayısı ve her katmandaki birim sayısı: Günümüzde, gizli katmanlardaki birim sayısının büyük olabileceği ve aşırı öğrenmenin çeşitli düzenleme yöntemleri ile kontrol edilebileceği düşünülmektedir.
2. Düzenleme parametreleri: Bırakma oranı (dropout rate), L1 (Lasso) ve L2 (Ridge) düzenlemesinin gücü gibi parametreler genellikle her katman için ayrı ayrı belirlenir.
3. Optimizasyon detayları: Mini-batch boyutu, epoch sayısı ve veri artırma (data augmentation) teknikleri gibi seçimler, stokastik gradyan inişinin performansını etkiler.

İşaret dili tanıma bağlamında, hiper parametre seçimi özellikle önemlidir. Örneğin:

- Evrişimli katmanların sayısı ve filtrelerin boyutu, el şekillerinin ve hareketlerin farklı ölçeklerdeki özelliklerini yakalamada kritik rol oynar.
- Bırakma oranı, modelin işaret dilindeki doğal varyasyonlara karşı dayanıklılığını artırabilir.
- Öğrenme oranı planlaması, modelin işaret dili verilerindeki karmaşık yapıları kademeli olarak öğrenmesine yardımcı olabilir.

Hiper parametre optimizasyonu süreci zahmetli olabilir ve dikkatli yapılmadığında aşırı öğrenmeye yol açabilir. Bu nedenle, model performansı ile hesaplama maliyeti arasında bir denge kurulması hedeflenecektir.



Şekil 5.13 Çift iniş fenomeninin doğal spline örneği üzerinden gösterimi

Şekil 5.13. Çift iniş fenomeni, bir boyutlu doğal spline örneği kullanılarak hata grafikleri ile gösterilmektedir. Yatay eksen, spline baz fonksiyonlarının log ölçeğindeki sayısını ifade etmektedir. Eğitim hatası, serbestlik derecesi örneklem boyutu $n = 20$ ile örtüştüğünde sıfıra ulaşmakta ve bu eşikten sonra sıfır kalmaktadır. Test hatası, bu eşikte dramatik bir şekilde artmakta ancak sonrasında makul bir değere düşmekte ve nihayetinde tekrar artmaktadır.

5.6.1 Epoch sayısı seçimi

Epoch sayısı, derin öğrenme modellerinin eğitiminde kritik öneme sahip bir hiper parametredir. Bir epoch, eğitim veri setinin tamamının model tarafından bir kez işlenmesi anlamına gelir. Epoch sayısının doğru seçimi, modelin yeterince öğrenmesini sağlarken aşırı öğrenmeyi (overfitting) önlemek açısından önemlidir. Bu çalışmada, epoch sayısının belirlenmesi için manuel değerlendirme yöntemi kullanılmıştır. Bu yaklaşım, her deney için en uygun yığın boyutu (batch size) ve öğrenme oranı (learning rate) ile birlikte optimal epoch sayısının belirlenmesini amaçlamaktadır. Erken durdurma, validasyon seti üzerindeki performansın belirli bir süre boyunca iyileşmemesi durumunda eğitimi sonlandıran bir tekniktir. Bu yaklaşım, modelin eğitim verisine aşırı uyum sağlamasını engellerken, optimum performansa ulaşmasını sağlar. Her deney için seçilen parametreler ve en iyi epoch sayısı, manuel inceleme ve hesaplama işlem süresine dayalı olarak belirlenmiştir.

5.6.2 Seyreltme (Dropout) öğrenme

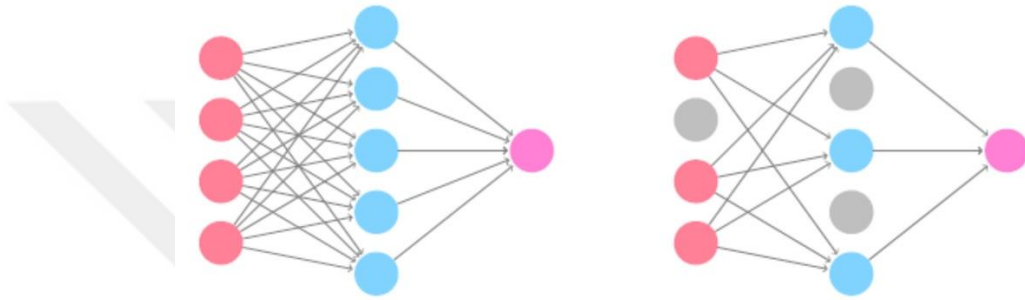
Seyreltme (dropout), derin öğrenme modellerinde kullanılan etkili bir düzenleme (regularization) tekniğidir. Bu yöntem, nispeten yeni olmasına rağmen, aşırı öğrenmeyi (overfitting) önlemede oldukça başarılı sonuçlar vermektedir ve bazı açılardan ridge düzenlemesine benzerlik göstermektedir. Seyreltme fikri, rastgele ormanlar (random forests) algoritmasından esinlenilerek ortaya çıkmıştır. Bu teknik, eğitim sırasında bir katmandaki nöronların belirli bir oranının (ϕ) rastgele devre dışı bırakılmasını içerir. Bu işlem, her bir eğitim örneği işlenirken ayrı ayrı gerçekleştirilir, yani her örnek için farklı nöronlar devre dışı bırakılabilir.

Seyreltme uygulandığında, hayatta kalan (devre dışı bırakılmayan) nöronlar, eksik olanların yerini alır. Bu nöronların ağırlıkları, devre dışı bırakılan nöronları telafi etmek için $1/(1 - \phi)$ faktörü ile ölçeklendirilir. Bu yaklaşım, nöronların aşırı özelleşmesini önler ve bir tür düzenleme olarak işlev görür. Şekil 5.14 bu durumu göstermektedir. Pratikte seyreltme, ağıın mimarisini değiştirmeden uygulanabilir. Bu, "seyreltilmiş" nöronların aktivasyonlarını rastgele sıfıra ayarlayarak gerçekleştirilir. Bu sayede, ağıın genel yapısı korunurken, eğitim sırasında farklı nöron alt kümelerinin kullanılması sağlanır.

İşaret dili tanıma bağlamında seyreltme öğrenme, özellikle önemlidir:

1. **Özellik Bağımsızlığı:** Seyreltme, modelin belirli özelliklere aşırı bağımlı hale gelmesini önler. Bu, işaret dilindeki çeşitli el şekilleri ve hareketleri arasındaki ilişkileri daha iyi öğrenmesini sağlar.
2. **Gürültüye Karşı Dayanıklılık:** İşaret dili verileri genellikle gürültülüdür (örneğin, arka plan değişimleri, ışık koşulları). Seyreltme, modelin bu tür gürültülere karşı daha dayanıklı olmasını sağlar.
3. **Genelleme Yeteneği:** Farklı kişilerin aynı işareti biraz farklı şekillerde yapabileceği göz önüne alındığında, seyreltme modelin daha iyi genelleme yapmasına yardımcı olur.

Seyreltme oranının seçimi, model karmaşıklığı ve veri seti büyüklüğü gibi faktörlere bağlı olarak değişebilir. Daha karmaşık modeller ve daha küçük veri setleri için genellikle daha yüksek seyreltme oranları tercih edilirken, daha basit modeller ve büyük veri setleri için daha düşük oranlar yeterli olabilir. Sonuç olarak, seyreltme öğrenme tekniği, Türk İşaret Dili tanıma modelinin aşırı öğrenmesini önlemede ve genelleme yeteneğini artırmada etkili bir araç olarak kullanılmıştır. Bu tekniğin uygun şekilde uygulanması, modelin daha sağlam ve güvenilir sonuçlar üretmesine katkıda bulunmuştur.



Şekil 5.14 Seyreltme (Dropout) uygulaması: Tam bağlantılı ve seyreltilmiş ağı karşılaştırması

Şekil 5.14’te Seyreltme (Dropout) öğrenme. Sol: Tam bağlantılı bir ağı. Sağ: Giriş ve gizli katmanda seyreltme uygulanmış ağı. Gri renkli düğümler rastgele seçilmiş olup, bir eğitim örneğinde göz ardı edilmektedir.

5.6.3 Katman sayısının belirlenmesi

Derin öğrenme modellerinde katman sayısının belirlenmesi, model performansını doğrudan etkileyen kritik bir hiper parametre seçimidir. Katman sayısı, modelin karmaşıklığını ve öğrenme kapasitesini belirler. İşaret dili tanıma gibi karmaşık görevlerde, optimal katman sayısının belirlenmesi özellikle önemlidir. Katman sayısının artırılması, modelin daha karmaşık özellikleri öğrenmesine olanak sağlar. Ancak, çok fazla katman kullanmak, aşırı öğrenme (overfitting) riskini artırabilir ve hesaplama maliyetini yükseltebilir. Öte yandan, çok az katman kullanmak da modelin yetersiz öğrenmesine (underfitting) neden olabilir.

İşaret dili tanıma bağlamında, katman sayısının belirlenmesinde şu faktörler göz önünde bulundurulmuştur:

1. Veri Karmaşıklığı: Türk İşaret Dili'ndeki işaretlerin çeşitliliği ve karmaşıklığı, daha derin bir model gerektirebilir.
2. Veri Miktarı: Eğitim veri setinin boyutu, destekleyebileceği model derinliğini etkiler. Daha büyük veri setleri genellikle daha derin modelleri destekler.
3. Özellik Hiyerarşisi: İşaret dilindeki el şekilleri, hareketler ve yüz ifadeleri gibi farklı seviyelerdeki özellikleri yakalamak için yeterli sayıda katman gereklidir.
4. Hesaplama Kaynakları: Mevcut donanım kapasitesi ve eğitim süresi kısıtlamaları da göz önünde bulundurulmalıdır.

5.7 Aktivasyon Fonksiyonu

Aktivasyon fonksiyonu, yapay sinir ağlarının temel bileşenlerinden biridir ve hücreye gelen bilgileri işleyerek çıktı üretme görevini üstlenir. Bu fonksiyon, modelin doğrusal olmayan özellikleri öğrenmesini sağlayarak, basit bir doğrusal modelden çok daha karmaşık ilişkileri yakalayabilmesine olanak tanır. Aktivasyon fonksiyonları genellikle doğrusal olmayan bir yapıya sahiptir ve çıkış değerlerini belirli bir aralıkta (örneğin, 0 ile 1 veya -1 ile 1 arasında) sınırlar. Bu özellik, ağın katmanları arasında bilgi akışını düzenler ve gradyan akışını kontrol eder. İşaret dili tanıma bağlamında, aktivasyon fonksiyonunun seçimi özellikle önemlidir. Farklı el şekilleri, hareketler ve yüz ifadeleri arasındaki karmaşık ilişkileri modellemek için uygun aktivasyon fonksiyonları gereklidir.

Aktivasyon fonksiyonunun uygulanması, genellikle ağırlıklı toplam ve sapma (bias) değerinin hesaplanmasından sonra gerçekleşir:

$$y = f\left(\sum_x (w_i * x_i) + b\right)$$

Burada w_i ağırlıkları, x_i girdileri, b sapmayı ve f aktivasyon fonksiyonunu temsil eder.

5.7.1 Sigmoid fonksiyonu

Sigmoid fonksiyonu, derin öğrenme ve yapay sinir ağlarında yaygın olarak kullanılan bir aktivasyon fonksiyonudur. Bu fonksiyon, herhangi bir gerçekte sayı girdisini 0 ile 1 arasında bir değere dönüştüren doğrusal olmayan matematiksel bir işlemdir. Sigmoid fonksiyonunun matematiksel ifadesi şu şekildedir:

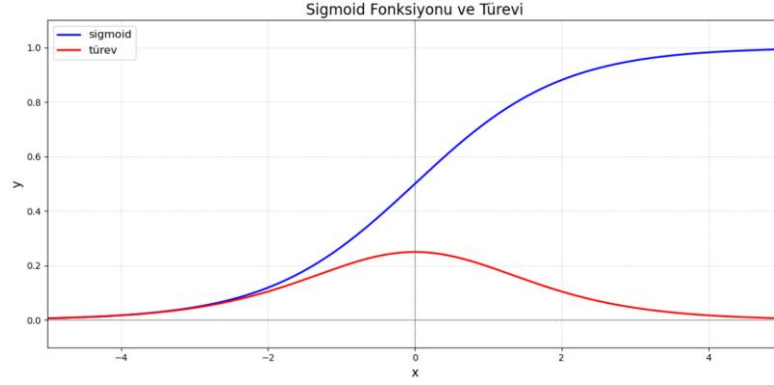
$$f(x) = \frac{1}{(1 + e^{-x})}$$

Sigmoid fonksiyonunun derin öğrenme alanındaki önemi, özellikle geri yayılım algoritmasının çalışma prensibinden kaynaklanmaktadır. Geri yayılım algoritması, parametrelerin türevini alarak ağırlıkları günceller. Bu nedenle, aktivasyon fonksiyonunun türevlenebilir olması kritik öneme sahiptir.

Sigmoid fonksiyonunun başlıca özellikleri ve avantajları şunlardır:

1. **Türevlenebilirlik:** Sigmoid fonksiyonu sürekli ve türevlenebilir bir fonksiyondur. Bu özellik, geri yayılım algoritmasının etkili bir şekilde çalışmasını sağlar.
2. **Sınırlı Çıktı Aralığı:** Fonksiyon, herhangi bir girdiyi 0 ile 1 arasında bir değere dönüştürür. Bu özellik, özellikle ikili sınıflandırma problemlerinde çıktıların olasılık olarak yorumlanmasına olanak tanır.
3. **Düzgün Gradyan:** Sigmoid fonksiyonu, girdi değerlerindeki küçük değişikliklerin çıktı değerlerinde gözlemlenebilir değişikliklere yol açmasını sağlayan pürüzsüz bir eğriye sahiptir.
4. **Doğrusal Olmama:** Sigmoid fonksiyonunun doğrusal olmaması, yapay sinir ağlarının karmaşık, doğrusal olmayan ilişkileri öğrenmesine olanak tanır.

İşaret dili tanıma bağlamında, sigmoid fonksiyonu özellikle çıkış katmanında kullanışlı olabilir. Örneğin, belirli bir el hareketi veya yüz ifadesinin varlığını veya yokluğunu tespit etmek için sigmoid fonksiyonu ideal bir seçim olabilir. Çıktı değeri 0'a yakınsa işaretin olmadığını, 1'e yakınsa işaretin var olduğunu gösterir.



Şekil 5. 15 Sigmoid fonksiyon grafiği

5.7.2 Softmax fonksiyonu

Softmax fonksiyonu, derin öğrenme modellerinde, özellikle çok sınıflı sınıflandırma problemlerinde yaygın olarak kullanılan bir aktivasyon fonksiyonudur. Sigmoid fonksiyonuna benzer bir yapıya sahip olan softmax, genellikle modellerin çıkış katmanında tercih edilir. Bu fonksiyon, girdi değerlerini, her bir sınıfa ait olma olasılığını temsil eden (0, 1) aralığında değerlere dönüştürür ve bu olasılıkların toplamı her zaman 1'e eşittir.

Softmax fonksiyonunun matematiksel ifadesi şu şekildedir:

$$\text{softmax}(x_i) = \frac{e^{x_i}}{\sum(e^{x_j})}$$

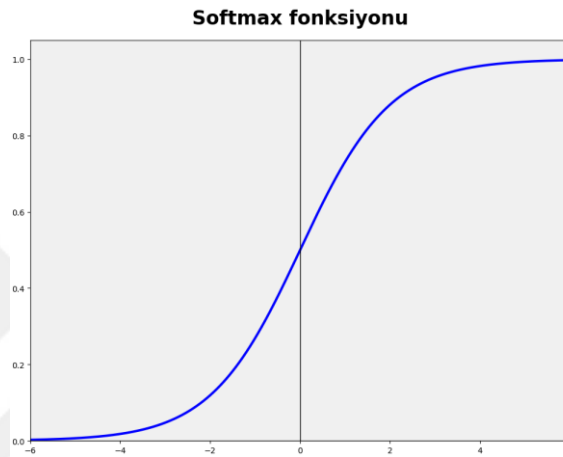
Burada x_i , i. sınıf için girdi değerini, e Euler sayısını temsil eder ve payda tüm sınıflar üzerinden toplam alır.

Türk İşaret Dili tanıma modelinde softmax fonksiyonunun kullanımı, özellikle çıkış katmanında, modelin farklı işaretler arasında ayırım yapma yeteneğini artırır. Örneğin, model bir el hareketini analiz ettiğinde, softmax fonksiyonu bu hareketin hangi işarete ait olma olasılığını her bir olası işaret için hesaplar.

Ancak, softmax fonksiyonunun bazı sınırlamaları da vardır:

1. Hesaplama Maliyeti: Büyük sayıda sınıf olduğunda hesaplama açısından maliyetli olabilir.
2. Aşırı Güven: Bazen model yanlış tahminlerde bile yüksek güven skorları üretebilir.

Bu fonksiyon sayesinde model, yalnızca işaretleri tanımakla kalmamış, aynı zamanda bu tanımların güvenilirliği hakkında da değerli bilgiler sağlamıştır (El-Amir vd. 2020: 302).



Şekil 5. 16 Softmax fonksiyon grafiği

5.7.3 Hiperbolik tanjant fonksiyonu (Tanh)

Hiperbolik Tanjant (Tanh) aktivasyon fonksiyonu, sigmoid fonksiyonunun genişletilmiş bir versiyonu olarak kabul edilir ve derin öğrenme modellerinde yaygın olarak kullanılır. Tanh fonksiyonu, giriş değerlerini -1 ile 1 arasında bir çıkış aralığına sıkıştırır, bu da onu sigmoid fonksiyonundan ayıran temel özelliklerden biridir.

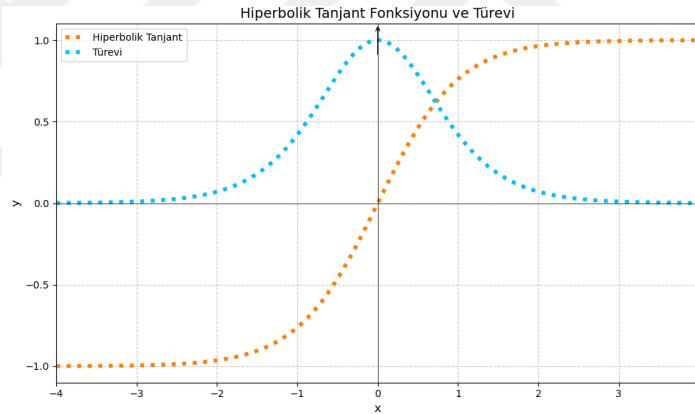
Tanh fonksiyonunun matematiksel ifadesi şu şekildedir:

$$\tanh(x) = \frac{e^x - e^{-x}}{(e^x + e^{-x})}$$

Tanh fonksiyonunun başlıca özellikleri ve avantajları şunlardır:

1. Simetrik Yapı: Tanh fonksiyonu, orijin etrafında simetriktir, bu da giriş değerlerinin sıfır etrafında merkezlenmesini sağlar.
2. Daha Dik Eğim: Sigmoid fonksiyonuna göre daha dik bir eğime sahiptir. Bu özellik, daha hızlı öğrenmeye ve daha geniş bir değer aralığı alabilmeye olanak tanır.
3. Daha Geniş Gradyan Aralığı: Türevi, sigmoid fonksiyonuna göre daha geniş bir aralıkta değer alır, bu da gradyan akışını iyileştirir.
4. Negatif Çıktılar: -1 ile 1 arasında değerler üretebilmesi, bazı uygulamalarda avantaj sağlayabilir.

Optimal performans için diğer aktivasyon fonksiyonlarıyla (örneğin ReLU) birlikte kullanılması ve her katman için en uygun aktivasyon fonksiyonunun seçilmesi önemlidir (Kızrak, 2019).



Şekil 5.17 Hiperbolik tanjant fonksiyon grafiği

5.7.4 Doğrultulmuş lineer birim (ReLU)

Doğrultulmuş Lineer Birim (Rectified Linear Unit - ReLU), derin öğrenme alanında, özellikle Evrişimli Sinir Ağları'nda (CNN) yaygın olarak kullanılan bir aktivasyon fonksiyonudur. ReLU, sigmoid ve hiperbolik tanjant gibi geleneksel aktivasyon fonksiyonlarına göre birçok avantaj sunar ve evrişimli ağlarda daha etkili performans gösterir (Zhang vd., 2017). ReLU fonksiyonu matematiksel olarak şu şekilde ifade edilir:

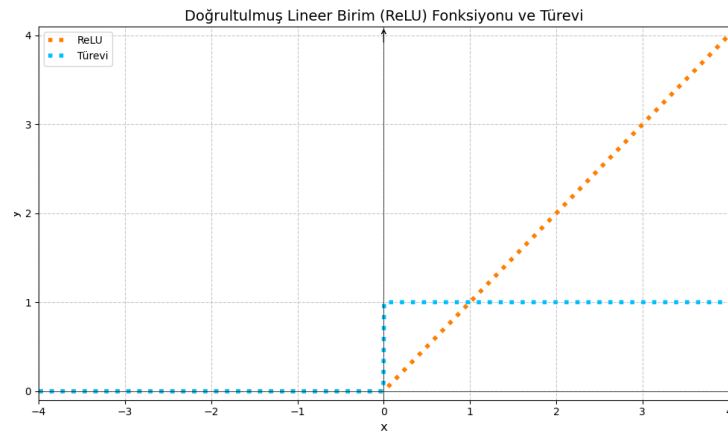
$$f(x) = \max(0, x)$$

Bu basit formül, giriş değeri pozitif olduğunda aynı değeri çıktı olarak verir, negatif olduğunda ise çıktıyı sıfıra eşitler. ReLU'nun çalışma süreci, özellik haritasındaki her pikseli bu fonksiyona sokarak gerçekleştirilir (Hibatullah vd. 2019).

ReLU'nun başlıca özellikleri ve avantajları şunlardır:

1. Doğrusal Olmayan Yapı: ReLU, basit görünümüne rağmen doğrusal olmayan bir yapıya sahiptir, bu da derin ağların karmaşık yapıları öğrenmesine olanak tanır.
2. Seyrek Aktivasyon: Negatif girişleri sıfıra eşitlemesi, ağın seyrek bir yapıya sahip olmasını sağlar, bu da hesaplama verimliliğini artırır.
3. Gradyan Kaybını Azaltma: Pozitif değerler için sabit bir gradyan (1) sağlar, bu da derin ağlarda gradyan kaybı problemini azaltır.
4. Hızlı Hesaplama: Basit matematiksel yapısı sayesinde, diğer aktivasyon fonksiyonlarına göre daha hızlı hesaplanabilir.

Bu özellikler, gerçek zamanlı işaret dili tanıma sistemlerinin geliştirilmesinde ReLU'yu vazgeçilmez bir bileşen haline getirmektedir (Zhang vd. 2017; Hibatullah vd. 2019).



Şekil 5.18 Doğrultulmuş lineer birim (ReLU) fonksiyonu ve türevi grafiği

5.8 Optimizasyon Yöntemleri

Derin öğrenme modellerinde, özellikle işaret dili tanıma gibi karmaşık görevlerde, optimizasyon yöntemleri kritik bir rol oynar. Bu yöntemler, modelin eğitim aşamasında kayıp fonksiyonunu minimize ederek, ağırlık ürettiği çıkış değeri ile gerçek değer arasındaki farkı en aza indirmeyi amaçlar (Zhang vd. 2021: 435). Kayıp fonksiyonu, gerçek y değeri ile tahmin edilen $\hat{y}$ arasındaki farkı ölçer. Bu fonksiyon, modele geri bildirim sağlayarak parametrelerin ayarlanmasına ve böylece yerel veya global minimumun bulunmasına olanak tanır. Kayıp fonksiyonu sıfıra yaklaştıkça veya sıfır olduğunda, model öğrenmeyi durdurur (IBM 2023).

Optimize ediciler, sinir ağındaki ağırlıkları ve sapmaları ayarlayarak tahmin edilen çıktı ile gerçek çıktı arasındaki hatayı minimize etmeye çalışır. Bu algoritmalar, her iterasyonda ağırlıkları ve sapmaları güncelleyerek çalışır. İşaret dili tanıma bağlamında, optimize edicilerin seçimi özellikle önemlidir. Çünkü işaret dilinin dinamik yapısı ve çok boyutlu özellikleri, etkili bir optimizasyon stratejisi gerektirir. Popüler optimize ediciler arasında Gradyan İniş, Stokastik Gradyan İniş (SGD), Adagrad, RMSProp ve Adam bulunur (Campesato 2020: 108). Gradyan iniş, temel bir optimizasyon yöntemi olup, diğer birçok yöntem buna dayanır. Toplu gradyan iniş, tüm eğitim verisiyle ağırlıkları günceller, ancak büyük veri setlerinde yavaş kalabilir. Stokastik gradyan iniş, her iterasyonda tek bir örnek kullanarak daha hızlı çalışır, ancak gürültüye daha duyarlıdır. Mini-batch gradyan iniş, bu iki yaklaşımın avantajlarını birleştirir.

5.9 Kayıp Fonksiyonu

Kayıp fonksiyonu, derin öğrenme modellerinde, özellikle işaret dili tanıma gibi karmaşık görevlerde, modelin performansını değerlendirmek ve optimize etmek için kullanılan kritik bir bileşendir. Bu fonksiyon, modelin tahmin ettiği değer ile gerçek değer arasındaki farkı ölçer ve bu farkı minimize etmeyi amaçlar. Kayıp fonksiyonu, maliyet fonksiyonu ve amaç fonksiyonu sıklıkla birbirine karıştırılan terimlerdir. Andrew Ng (2012) Machine Learning derslerinde, kayıp fonksiyonunu tek bir örnek için hesaplanan hata, maliyet fonksiyonunu ise tüm eğitim seti için hesaplanan hata olarak tanımlamıştır.

Ian Goodfellow ve arkadaşları (2015) ise "Deep Learning" kitabında amaç fonksiyonunu maksimize ya da minimize edilmek istenen fonksiyon olarak açıklamıştır. Bu bağlamda, kayıp fonksiyonu, bir amaç fonksiyonunun türü olan maliyet fonksiyonunun bir parçası olarak düşünülebilir.

İşaret dili tanıma sistemlerinde kayıp fonksiyonu, modelin işaretleri ne kadar doğru tanıdığını değerlendirmek için kullanılır. Kayıp fonksiyonunun verdiği değer ne kadar düşükse, modelin işaretleri o kadar iyi tanıdığı söylenebilir. Yüksek kayıp değerleri, modelin parametrelerinin güncellenmesi gerektiğini gösterir. Derin sinir ağlarında kayıp fonksiyonunu minimize etmek için çeşitli optimizasyon algoritmaları kullanılır. En yaygın kullanılan algoritma, gradyan azaltma (gradient descent) ve onun türevleridir. Bunlar arasında stokastik gradyan azaltma (SGD), Adadelta, Adamax ve Adam gibi algoritmalar bulunur. Bu algoritmaların seçimi, veri setine ve problem türüne göre farklılık gösterebilir, çünkü her biri farklı başarımlarına ve hızlara sahiptir (Ser vd. 2019; Anonymous 2023). İşaret dili tanıma bağlamında, kayıp fonksiyonunun seçimi ve optimizasyonu özellikle önemlidir. Çünkü işaret dili, el hareketleri, yüz ifadeleri ve vücut duruşu gibi çoklu modaliteleri içerir. Bu nedenle, kayıp fonksiyonu bu farklı modaliteleri etkili bir şekilde yakalayabilmelidir. Örneğin, kategorik çapraz entropi kaybı, çok sınıflı işaret tanıma problemleri için sıklıkla kullanılır.

5.9.1 ADAM optimizasyonu

ADAM (Adaptive Moment Estimation) optimizasyonu, derin öğrenme modellerinin eğitiminde yaygın olarak kullanılan etkili bir optimizasyon algoritmasıdır. 2014 yılında Kingma ve Ba tarafından geliştirilen bu algoritma, RMSprop ve momentum yöntemlerinin avantajlarını birleştirerek, hızlı ve verimli bir öğrenme süreci sağlar. ADAM, gradyanların birinci ve ikinci momentlerinin tahminlerini kullanarak öğrenme oranlarını türetir (Seyyarer vd. 2020). Bu yaklaşım, algoritmanın durağan olmayan, gürültülü ve seyrek veriler üzerinde etkili performans göstermesini sağlar. ADAM'ın uygulaması nispeten basittir ve daha az bellek gerektirir, bu da onu büyük veri setleri ve karmaşık modeller için ideal kılar (Kingma vd. 2014).

5.9.2 Düzenleştirme ve stokastik gradyan inişi

Derin öğrenme modellerinin, özellikle işaret dili tanıma gibi karmaşık görevlerde, etkili bir şekilde eğitilmesi için düzenleştirme ve uygun optimizasyon algoritmalarının kullanılması kritik öneme sahiptir. Bu bağlamda, düzenleştirme ve stokastik gradyan inişi (SGD) iki temel bileşen olarak öne çıkmaktadır.

Düzenleştirme, modelin aşırı uyumunu (overfitting) önlemek için kullanılan bir tekniktir. İşaret dili tanıma modellerinde, özellikle model parametrelerinin sayısı eğitim örneklerinin sayısından fazla olduğunda, düzenleştirme hayati önem taşır. Ridge düzenlemesi, bu amaçla sıkça kullanılan bir yöntemdir. Bu yöntem, amaç fonksiyonuna bir ceza terimi ekleyerek çalışır:

$$R(\theta; \lambda) = -\sum_{i=1}^n \sum_{m=0}^9 y_{im} \log(f_m(x_i)) + \lambda \sum_j \theta_j^2 \quad (24)$$

Burada $R(\theta; \lambda)$ orijinal kayıp fonksiyonu, λ düzenleştirme parametresi ve $\sum_j \theta_j^2$ ağırlıkların karelerinin toplamıdır. Bu yaklaşım, modelin ağırlıklarını küçük tutmaya teşvik eder, böylece model daha genelleştirilebilir hale gelir. Stokastik Gradyan İnişi (SGD), büyük veri setleriyle çalışırken optimizasyon sürecini hızlandırmak için kullanılan bir tekniktir. Geleneksel gradyan inişinin aksine SGD, her adımda tüm veri seti yerine küçük bir alt küme (mini-batch) kullanır. Bu yaklaşım, özellikle işaret dili tanıma gibi büyük veri setleri gerektiren görevlerde eğitim sürecini önemli ölçüde hızlandırır. λ parametresi genellikle küçük bir değerde önceden ayarlanır. Farklı katmanlardan gelen ağırlık grupları için farklı λ değerleri de kullanabiliriz; bu durumda w_1 ve w_2 cezalandırılırken, çıktı katmanının görece az sayıdaki B ağırlıkları hiç cezalandırılmamıştır. Lasso düzenlemesi de ek bir düzenleme biçimi olarak veya ridge alternatif olarak popülerdir.

5.9.3 Geri yayılım

Geri yayılım (backpropagation), derin öğrenme modellerinin, özellikle de işaret dili tanıma gibi karmaşık görevlerde kullanılan sinir ağlarının eğitiminde temel bir

algoritmadır. Bu algoritma, modelin hatasını minimize etmek için ağırlıkların nasıl güncelleneceğini belirler. Geri yayılımın temel prensibi, ağıın çıktısı ile beklenen çıktı arasındaki hatayı hesaplamak ve bu hatayı ağıın katmanları boyunca geriye doğru yayarak her bir ağırlığın bu hataya olan katkısını belirlemektir. Bu süreç, matematiksel olarak gradyan inişi ve zincir kuralı kullanılarak gerçekleştirilir.

Formül 18'teki $R(\theta)$ amacını azaltmak için θ 'yı nasıl hareket ettireceğimiz yönünü nasıl bulabiliriz? $R(\theta)$ 'nın gradyanı, $\theta = \theta^m$ gibi mevcut bir değerde değerlendirildiğinde, o noktadaki kısmi türevler vektörüdür:

$$\nabla R(\theta^m) = \frac{\partial R(\theta)}{\partial \theta} \Big|_{\theta=\theta^m} \quad (19)$$

Alt indis $\theta = \theta^m$, türevler vektörünü hesapladıktan sonra onu mevcut tahmin θ^m 'de değerlendirdiğimiz anlamına gelir. Bu, $R(\theta)$ 'nın en hızlı şekilde arttığı θ -uzayındaki yönü verir. Gradyan inişinin fikri, θ 'yı karşıt yönde (çünkü aşağı doğru gitmek istiyoruz) biraz hareket ettirmektir:

$$\theta^{m+1} \leftarrow \theta^m - \rho \nabla R(\theta^m) \quad (20)$$

Öğrenme oranı ρ için yeterince küçük bir değer için, bu adım amacı azaltacaktır:

$R(\theta)$; i.e. $R(\theta^{m+1}) \leq R(\theta^m)$ Eğer gradyan vektörü sıfır ise, amacın bir minimumuna ulaşmış olabiliriz.

Formül (19) hesaplaması ne kadar karmaşıktır? Burada oldukça basit olduğu ve türevin zincir kuralı nedeniyle çok daha karmaşık ağılar için bile basit kaldığı ortaya çıkmıştır.

$R(\theta) = \sum_{i=1}^n R_i(\theta) = \frac{1}{2} \sum_{i=1}^n (y_i - f_{\theta}(x_i))^2$ bir toplam olduğundan, onun gradyanı da n gözlem üzerinden bir toplamdır, bu yüzden sadece bu terimlerden birini inceleyeceğiz,

$$R_i(\theta) = \frac{1}{2} \left(y_i - \beta_0 - \sum_{k=1}^K \beta_k g(w_{k0} + \sum_{j=1}^p w_{kj} x_{ij}) \right)^2 \quad (21)$$

Takip eden ifadeleri basitleştirmek için [formül] yazıyoruz. Önce β_k 'ye göre türevi alıyoruz:

$$\begin{aligned}\frac{\partial R_i(\theta)}{\partial \beta_k} &= \frac{\partial R_i(\theta)}{\partial f_\theta(x_i)} \cdot \frac{\partial f_\theta(x_i)}{\partial \beta_k} \\ &= -(y_i - f_\theta(x_i)) \cdot g(z_{ik})\end{aligned}\quad (22)$$

Ve şimdi w_{kj} 'ye göre türevi alıyoruz:

$$\begin{aligned}\frac{\partial R_i(\theta)}{\partial w_{kj}} &= \frac{\partial R_i(\theta)}{\partial f_\theta(x_i)} \cdot \frac{\partial f_\theta(x_i)}{\partial g(z_{ik})} \cdot \frac{\partial g(z_{ik})}{\partial z_{ik}} \cdot \frac{\partial z_{ik}}{\partial w_{kj}} \\ &= -(y_i - f_\theta(x_i)) \cdot \beta_k \cdot g'(z_{ik}) \cdot x_{ij}.\end{aligned}\quad (23)$$

Her iki ifadenin de $y_i - f_\theta(x_i)$ artığını içerdiğine dikkat ediniz. Formül (22)'de bu artığın bir kısmının $g(z_{ik})$ değerine göre her bir gizli birime atfedildiğini görüyoruz. Ardından Formül (23)'da k gizli birimi aracılığıyla j girişine benzer bir atfetme görüyoruz. Böylece türev alma işlemi, zincir kuralı aracılığıyla artığın bir kısmını her bir parametreye atar bu süreç, sinir ağı literatüründe geri yayılım olarak bilinir. Bu hesaplamalar basit olmasına rağmen, tüm parçaları takip etmek için dikkatli bir muhasebe gerektirir.

5.10 Transfer Öğrenme

Transfer öğrenme, derin öğrenme alanında giderek daha fazla önem kazanan bir tekniktir. Transfer öğrenme, bir alanda öğrenilen bilgi ve becerilerin başka bir alana aktarılmasını sağlar, böylece yeni görevler için daha az veri ve hesaplama kaynağı kullanılarak etkili modeller oluşturulabilir. SLR alanında, transfer öğrenme teknikleri önceden eğitilmiş çeşitli popüler derin öğrenme ağları ve yapılandırmaları kullanılarak uygulanmıştır. Bu uygulamalar, cesaret verici sonuçlar ortaya koymuştur (Mocialov vd. 2020, Halvardsson vd. 2021). Özellikle, büyük ve çeşitli veri setleri üzerinde eğitilmiş modellerin, işaret dili tanıma gibi spesifik görevlere adapte edilmesi, modelin performansını önemli ölçüde artırabilmektedir. Sentetik veriler, insan fotoğrafçılığından ziyade yapay yollarla üretilen verilerdir. Bu veriler, bilgisayar görü uygulamaları için makine öğrenmesi modellerini

eğitmek, üretken modelleri eğitmek, gerçek veri kümelerini artırmak ve gizliliğe duyarlı senaryolarda gerçek verileri anonimleştirmek için kullanılmaktadır (Bochinski vd. 2016). Sentetik veri kullanan teknikler, nesne algılama ve segmentasyon, yüz ve metin tanıma, görüntü sınıflandırma ve poz tahmini gibi çeşitli problemlere uygulanmıştır (Nikolenko 2019).

5.11 Derin Öğrenmede Kullanılan Yazılım Kütüphaneleri

Derin öğrenme modellerinin geliştirilmesi ve uygulanması sürecinde, çeşitli yazılım kütüphaneleri araştırmacılara ve geliştiricilere büyük kolaylık sağlamaktadır. Bu kütüphaneler, karmaşık matematiksel işlemleri otomatikleştirerek, modellerin hızlı bir şekilde oluşturulmasına ve eğitilmesine olanak tanır. İşaret dili tanıma gibi özel uygulamalarda, bu kütüphanelerin sunduğu esneklik ve performans özellikle önemlidir.

5.11.1 Tensorflow

TensorFlow, Google Brain ekibi tarafından geliştirilen açık kaynaklı bir makine öğrenimi kütüphanesidir. Özellikle derin öğrenme uygulamaları için tasarlanmış olan TensorFlow, yüksek performanslı sayısal hesaplama için kullanılır.

TensorFlow'un işaret dili tanıma projelerindeki başlıca avantajları şunlardır:

1. Esneklik: Çeşitli sinir ağı mimarilerini (CNN, RNN, LSTM) kolayca uygulamaya olanak tanır.
2. Dağıtık Hesaplama: Büyük ölçekli işaret dili veri setleri üzerinde eğitim için çoklu GPU ve TPU desteği sunar.
3. Görselleştirme Araçları: TensorBoard ile model performansını ve eğitim sürecini izlemeyi kolaylaştırır.
4. Özelleştirilebilirlik: Özel katmanlar ve kayıp fonksiyonları tanımlayarak, işaret diline özgü gereksinimlere uyum sağlar.

Bu tez çalışmasında, TensorFlow, Türk İşaret Dili tanıma modelinin oluşturulması ve eğitilmesi için kullanılmıştır. Özellikle, TensorFlow'un Keras API'si, model mimarisinin hızlı prototiplenmesine ve farklı hiperparametre kombinasyonlarının denenmesine olanak sağlamıştır.

5.11.2 Python

Python, derin öğrenme ve veri bilimi alanında en yaygın kullanılan programlama dillerinden biridir. Basit sözdizimi, zengin kütüphane ekosistemi ve geniş topluluk desteği, Python'u işaret dili tanıma gibi karmaşık projeler için ideal bir seçim haline getirmektedir.

Python'un işaret dili tanıma projelerindeki başlıca avantajları şunlardır:

1. Kolay Öğrenme ve Kullanım: Açık ve okunabilir sözdizimi, hızlı geliştirme ve prototipleme sağlar.
2. Geniş Kütüphane Ekosistemi: NumPy, Pandas, Scikit-learn gibi kütüphaneler veri işleme ve analiz süreçlerini kolaylaştırır.
3. Görüntü İşleme Desteği: OpenCV gibi kütüphanelerle işaret dili videolarını işlemek kolaylaşır.
4. Çok Platformlu Destek: Farklı işletim sistemlerinde çalışabilme özelliği, geliştirme sürecini esnek kılar.

Bu çalışmada, Python, tüm veri ön işleme, model geliştirme ve değerlendirme aşamalarında kullanılmıştır. Python'un nesne yönelimli programlama özellikleri, karmaşık işaret dili tanıma sisteminin modüler ve bakımı kolay bir şekilde yapılandırılmasına olanak sağlamıştır. TensorFlow ve Python'un birlikte kullanımı, Türk İşaret Dili tanıma modelinin geliştirilmesinde sinerji yaratmıştır. TensorFlow'un güçlü derin öğrenme yetenekleri, Python'un esnekliği ve kullanım kolaylığı ile birleştiğinde, etkili ve verimli bir geliştirme süreci mümkün olmuştur. Bu kombinasyon, modelin hızlı bir şekilde prototiplenmesine, eğitilmesine ve optimize edilmesine olanak tanıyarak, işaret dili tanıma alanında yenilikçi çözümler üretmeye katkıda bulunmuştur.

6. GÖRÜNTÜ İŞLEME

Görüntü işleme, dijital görüntülerin manipüle edilmesi ve analiz edilmesi sürecidir. İşaret dili tanıma bağlamında, görüntü işleme teknikleri, el hareketlerini ve yüz ifadelerini analiz etmek için kritik bir rol oynamaktadır. Görüntü işleme, önceden ölçülüp dijital ortama aktarılmış verilerin, bilgisayar veya belirli yazılımlar yardımıyla belirlenen amaç doğrultusunda değiştirilmesi için yapılan bir işlem olarak tanımlanabilir. Başka bir deyişle, kaydedilen görüntüler üzerinde çeşitli görüntü filtreleri uygulanarak değişiklikler yapmak ya da iyileştirmeler gerçekleştirmek anlamına gelir (Yılmaz 2016; Karakoç 2012).

6.1 Görüntü İşleme Teknikleri

Görüntü işleme teknolojisi, görüntüleri metne dönüştürme yeteneğine sahiptir. Bu süreçte kullanılan veriler statik (fotoğraflar) veya dinamik (video) görüntüler olabilir. İşaret dili tanıma için, genellikle video görüntüleri kullanılır, çünkü işaretler zamansal bir boyuta sahiptir. Görüntü işleme uygulamaları, çeşitli yazılımlar ve kütüphaneler kullanır. Bunlar arasında el anahtar noktası kütüphaneleri özellikle önemlidir. Bu kütüphaneler, elin ve parmakların pozisyonlarını tespit ederek, işaretlerin tanımlanmasına yardımcı olur. Ancak, görüntü işleme tek başına bir işaret dili çevirmeni olarak yeterli değildir. İşaretleri doğru bir şekilde tanımlamak ve sınıflandırmak için, görüntü işleme tekniklerinin makine öğrenimi veya derin öğrenme algoritmaları ile birleştirilmesi gerekmektedir.

İşaret dili tanıma için kullanılan başlıca görüntü işleme teknikleri şunlardır:

- Kenar Algılama: El şekillerinin konturlarını belirlemek için kullanılır.
- Renk Segmentasyonu: Ten rengini kullanarak elleri arka plandan ayırmak için kullanılır.
- Hareket Algılama: İşaretlerin dinamik bileşenlerini yakalamak için kullanılır.
- Morfolojik İşlemler: Gürültüyü azaltmak ve el şekillerini iyileştirmek için kullanılır.

Bu tez çalışmasında, Türk İşaret Dili tanıma için OpenCV kütüphanesi kullanılarak görüntü işleme teknikleri uygulanmıştır. Özellikle, MediaPipe el anahtar noktası algılama algoritması, el pozisyonlarını ve hareketlerini tespit etmek için kullanılmıştır.

6.2 Görüntü İşlemenin Uygulama Alanları

Görüntü işleme, işaret dili tanıma dışında birçok alanda uygulanmaktadır. İşaret dili tanıma ile ilgili veya bu alana uyarlanabilecek başlıca uygulama alanları şunlardır:

- Tıbbi Görüntüleme: MRI, CT taramaları gibi görüntülerin analizi, işaret dili tanımada kullanılan el ve yüz algılama tekniklerine benzer yöntemler kullanır.
- Güvenlik ve Gözetim: Yüz tanıma sistemleri, işaret dili tanımada kullanılan yüz ifadesi analiz tekniklerinden yararlanabilir.
- Otonom Araçlar: Çevre algılama ve nesne tanıma, işaret dili tanımada kullanılan hareket algılama tekniklerine benzer yöntemler kullanır.
- Artırılmış Gerçeklik: El ve vücut hareketlerinin takibi, işaret dili tanıma teknikleriyle paralellik gösterir.
- İnsan-Bilgisayar Etkileşimi: Hareket tabanlı kontroller, işaret dili tanıma için geliştirilen teknikleri kullanabilir.

İşaret dili tanıma için geliştirilen görüntü işleme teknikleri, bu alanlara da uyarlanabilir veya bu alanlardan elde edilen gelişmelerden faydalanabilir. Örneğin, otonom araçlar için geliştirilen hızlı ve hassas nesne algılama algoritmaları, işaret dili tanımada el ve parmak pozisyonlarının daha doğru ve hızlı tespitinde kullanılabilir.

7. MEDIAPIPE İLE ANAHTAR NOKTA ÇIKARMA

İşaret Dili Tanıma (SLR) sistemlerinde, görsel verilerin etkili bir şekilde işlenmesi ve anlamlandırılması kritik öneme sahiptir. Bu bağlamda, MediaPipe gibi gelişmiş görüntü işleme kütüphaneleri, anahtar noktaların çıkarılmasında önemli bir rol oynamaktadır. İşaret dili içeriği genellikle RGB (renkli) videolar olarak kaydedilse de, derinlik haritaları, nokta bulutu, sensörler aracılığıyla kaydedilen parmak hareketleri ve işaretçinin iskelet temsili gibi çeşitli modaliteler de kullanılabilir. Bu çalışmada, poz-iskelet modalitesini kullanan İşaret Dili Tanıma (SLR) üzerine odaklanılmıştır. Poz tahmini, görüntülerdeki ve videolardaki insan figürlerini tespit ederek çeşitli eklemlerin konumlarını belirleme görevidir. Poz tahmin modelleri, dirsekler ve dizler gibi eklemlerin uzamsal koordinatlarını sağlayan anahtar noktaları çıkarır. Bu alan, 2D/3D pozların tahmin edilmesinden küçük bir özel işaret kümesinin veya bir kişinin yoğun bir ağının seçilmesine kadar değişen hedeflerle kapsamlı bir şekilde yeniden araştırılmıştır (Pishchulin vd. 2012). Vogler ve Goldenstein (2005), yüz pozunun yüzün elle tutulmayan görüntüleriyle ilişkili olduğunu göstermiştir. MediaPipe Holistic (Grishchenko ve Bazarevsky, 2020), 3D poz tahmini sorununu doğrudan çözmeye çalışan bir yaklaşım benimser. Bu sistem, önce vücudu, ardından yüz ve elleri tahmin etmek için bir boru hattı kullanır. Yüz pozunu için 468 nokta içeren yoğun bir örgü modeli kullanırken, vücut ve eller için iskelet eklemlerine başvurur. MediaPipe, pozları sınıflandırma yerine regresyon kullanarak 3D olarak tahmin eder.

İşaret dili tanıma bağlamında MediaPipe'in kullanımı şu adımları içerir:

1. Önceden eğitilmiş poz dedektörü ve işaret modeli kullanılarak insan pozunu tahmin edilir.
2. Çıkarılan yer işaretlerinden yararlanılarak her el ve poz için üç parçalı kırpmalar oluşturulur.
3. İlgi alanları (POI) iyileştirilir ve tam çözünürlüklü giriş karesine kırpılır.
4. Poz ve el modelleri kullanılarak ilgili yer işaretleri tahmin edilir.
5. Her el için 21 ve poz için 216 yer işareti birleştirilerek toplam 258 yer işareti elde edilir.

Üst vücut pozlarının, işaretçi ile değişmeyen temsil yetenekleri nedeniyle hesaplamalı işaret dili araştırmalarında bir özellik olarak yaygın şekilde kullanıldığı belirtilmektedir. Bu özellik, farklı işaretçiler arasında tutarlı bir temsil sağlar ve modelin genelleme yeteneğini artırır. MediaPipe'in bu şekilde kullanımı, işaret dili tanıma sistemlerinin performansını önemli ölçüde artırabilir. Özellikle, el hareketleri ve yüz ifadelerinin hassas bir şekilde yakalanması, işaretlerin doğru tanınması için kritik öneme sahiptir. Ayrıca, 3D poz tahmini, işaretlerin derinlik bilgisini de içererek daha kapsamlı bir analiz sağlar.

Bu tez çalışmasında, Türk İşaret Dili tanıma için MediaPipe kullanılarak anahtar nokta çıkarma işlemi gerçekleştirilmiştir. Elde edilen 258 yer işareti, derin öğrenme modelinin girdisi olarak kullanılmış ve modelin işaretleri doğru bir şekilde sınıflandırmasına olanak sağlamıştır. MediaPipe'in sağladığı yüksek hassasiyetli anahtar nokta çıkarma yeteneği, özellikle ince el hareketleri ve yüz ifadelerinin önemli olduğu Türk İşaret Dili'nin tanınmasında önemli bir avantaj sağlamıştır.



Şekil 6.1 Mediapipe ellerin ve poz işaretlerinin gösterimi

Poz tahmini alanında, OpenPose (Cao vd. 2017) ve Google'ın MediaPipe kütüphanesi gibi çeşitli teknikler öne çıkmaktadır. MediaPipe, cihaz üzerinde gerçek zamanlı poz takibi ve tahmini için tasarlanmış olup, OpenPose'a kıyasla 25-75 kat daha hızlı performans gösterdiği rapor edilmiştir (Bazarevsky vd. 2020). Bu üstün performansı nedeniyle MediaPipe, son dönemde yapılan birçok çalışmada tercih edilmiştir (Sreenath 2021; Selvaraj vd. 2021; Shin vd. 2021). MediaPipe, Google tarafından geliştirilen açık

kaynaklı bir çerçevedir ve rastgele duyuşal veriler üzerinde çıkarım yapmak için boru hatları oluřturmaya olanak saęlar. İnsan vücutu parçalarının tespiti ve takibi için, "nirenđi noktaları" olarak adlandırılan kilit noktaları izleyebilen yüksek doęruluklu makine öęrenimi modellerinden oluřan bir seçkiye sahiptir. Bu yer iřaretleri, $[0, 1]$ aralıęında normalize edilmiř üç boyutlu koordinat noktaları olarak temsil edilir.

MediaPipe Holistic, el, yüz ve vücut pozu yer iřareti tahmini için ayrı modelleri birleřtiren entegre bir sistemdir. İřleyiř süreci řu řekildedir:

1. Öncelikle vücut pozu tahmin edilir.
2. Yüz ve ellere karřılık gelen ilgi bölgeleri (ROI) tanımlanır.
3. Bu ROI'ler, tam kareyi el ve poz görüntülerine kırpma için kullanılır.
4. Kırpılan görüntüler, el ve yüz modellerine beslenir.
5. Tahmin edilen yer iřaretleri, daha önce çıkarılan vücut anahtar noktalarıyla birleřtirilerek toplam 540 el, yüz ve vücut yer iřareti oluřturulur.

MediaPipe'da, her bir iřaret belirli bir etiketle eřleřtirilir. İřaret Dili Tanıma (SLR) bağlamında, her etiketin ardışık karelerdeki otomatik olarak belirlenen konumlarının bir rota oluřturduęu model, başarılı tanıma sonuçları verebilmektedir. Bu yaklařım, iřaretlerin zamansal ve uzamsal özelliklerini etkili bir řekilde yakalamaya olanak tanır. Bu tez çalışmasında, Türk İřaret Dili tanıma için MediaPipe Holistic kullanılmıřtır. MediaPipe'ın saęladığı yüksek hassasiyetli ve hızlı anahtar nokta çıkarma yeteneęi, özellikle el hareketleri, yüz ifadeleri ve vücut duruşunun önemli olduęu Türk İřaret Dili'nin tanınmasında büyük avantaj saęlamıřtır. Elde edilen 540 yer iřareti, derin öęrenme modelinin girdisi olarak kullanılmıř ve modelin iřaretleri doęru bir řekilde sınıflandırmasına olanak tanımıřtır.

8. PERFORMANS DEĞERLENDİRME ÖLÇÜMLERİ

Performans değerlendirme ölçütleri, tahmin modellerinin doğruluğunu ve kalitesini incelemek için kritik öneme sahiptir. Ortaya çıkan hataların büyüklüğünü ve dağılımını ölçerek modellerin etkinliğini kıyaslamak mümkündür. Çalışmanızda kullanılan dört temel ölçüt MAE, MAPE, MSE ve RMSE, modellerin performansını değerlendirmek için yaygın olarak tercih edilen metriklerdir.

8.1 Ortalama Mutlak Hata (Mean Absolute Error- MAE)

MAE, tahmin edilen ve gerçek değerler arasındaki mutlak farkların ortalamasıdır. Modelin hatalarının ortalama büyüklüğünü ölçer.

$$MAE = \frac{1}{n} \sum_{t=1}^n |y_t - \hat{y}_t|$$

Burada y_t , gerçek değer, $\hat{y}_t$ ise tahmin edilen değerdir. n , veri setindeki toplam gözlem sayısını temsil eder.

8.2 Ortalama Mutlak Hata Yüzdesi (Mean Absolute Percentage Error- MAPE)

MAPE, tahmin edilen ve gerçek değerler arasındaki ortalama yüzdesel farkı ölçer. Farklı ölçeklerdeki veri setlerini karşılaştırmak için kullanılır.

$$MAPE = \frac{100}{n} \sum_{t=1}^n \left| \frac{y_t - \hat{y}_t}{y_t} \right|$$

8.3 Hata Kareler Ortalaması (Mean Squared Error- MSE)

MSE, tahmin edilen ve gerçek değerler arasındaki farkların karesinin ortalamasıdır. Büyük hataların daha ağır cezalandırılmasını sağlar.

$$MSE = \frac{1}{n} \sum_{t=1}^n (y_t - \hat{y}_t)^2$$

8.4 Hata Kareler Ortalamasının Karekökü (Root Mean Squared Error- RMSE)

Hata Kareler Ortalamasının Karekökü (Root Mean Squared Error - RMSE):
RMSE, MSE'nin kareköküdür ve hataların standart sapmasını gösterir.

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^n (y_t - \hat{y}_t)^2}$$

9. MODEL DEĞERLENDİRME METRİKLERİ

Etkili bir makine öğrenimi veya derin öğrenme modeli geliştirmenin temel aşamalarından biri, modelin performansının değerlendirilmesidir. Bu değerlendirme, modelin gerçek dünya senaryolarında nasıl performans göstereceğini tahmin etmemize ve farklı modelleri karşılaştırmamıza olanak sağlar. İşaret Dili Tanıma (SLR) bağlamında, model değerlendirme metrikleri özel bir öneme sahiptir. SLR genellikle tek etiketli bir sınıflandırma problemi olarak modellendiğinden, bir modelin performansını ve kalitesini ölçmek için asıl metrik olarak doğruluk (accuracy) kullanılmaktadır. Bununla birlikte, diğer en iyi k doğrulukları (top-k accuracy) da rapor edilebilir. Doğruluk, temel gerçek etiketinin tahmin edilen etiketle eşleştiği test örneklerinin oranı olarak tanımlanır ve şu şekilde hesaplanır:

$$\text{Doğruluk} = \frac{\text{Doğru Sınıflandırma Sayısı}}{\text{Toplam Test Durumu Sayısı}}$$

Ancak, doğruluk tek başına her zaman yeterli olmayabilir. Bu nedenle, makine öğrenimi modellerinde performansı ölçmek için çeşitli metrikler kullanılır. Bunlar arasında en yaygın olanları:

1. Kesinlik (Precision): Pozitif olarak tahmin edilen örnekler arasında gerçekten pozitif olanların oranı.
2. Geri Çağırma (Recall): Gerçekte pozitif olan örnekler arasında pozitif olarak tahmin edilenlerin oranı.
3. F-skoru: Kesinlik ve geri çağırmanın harmonik ortalaması.

Bu tez çalışmada, çeşitli deneylerdeki sonuçları karşılaştırmak için öncelikle doğruluk metriği seçilmiştir. Ancak, modelin performansını daha kapsamlı bir şekilde değerlendirmek için diğer metrikler de göz önünde bulundurulmuştur. SLR için model değerlendirmede dikkat edilmesi gereken önemli bir nokta, veri kümesinin oluşturulma ve bölünme şeklidir. Her bir örneğin farklı işaretçilerden, ortamlardan ve kamera yönelimleri, işaretleme hızı gibi diğer faktörlerden derlenerek sınıf başına yeterli

çeşitlilikte örnek olmasını sağlamak kritiktir. Bu yaklaşım, gerçek dünya senaryolarına genelleme yapabilen modeller oluşturabilmek için önemlidir. Ayrıca, tüm veri kümesinin, eğitim dağılımının doğrulama ve test kümelerine önemli ölçüde benzemeyeceği şekilde bölünmesi gerekmektedir.

9.1 Karmaşıklık Matrisi

Karmaşıklık matrisi (confusion matrix), bir sınıflandırma algoritmasının performansını detaylı bir şekilde değerlendirmek için kullanılan güçlü bir araçtır. Özellikle veri setinde eşit sayıda etiketin olmadığı veya ikiden fazla sınıfın bulunduğu durumlarda, sınıflandırma doğruluğu tek başına yanıltıcı olabilir. Bu gibi durumlarda karmaşıklık matrisi, modelin neyi doğru yaptığını ve ne tür hatalar yaptığını daha net bir şekilde ortaya koyar. Karmaşıklık matrisi, denetimli öğrenme modellerinin performans değerlendirmelerinde ve başarı görselleştirmede kullanılan bir matris tablosudur. Bu tablo, tüm sınıfların gerçek değerleri ile modelin sınıf tahmin değerlerini içerir.

Temel olarak, karmaşıklık matrisi dört ana bileşenden oluşur:

1. Doğru Pozitif (True Positive - TP): Modelin bir veriyi doğru bir şekilde pozitif olarak sınıflandırdığı durum.
2. Doğru Negatif (True Negative - TN): Modelin bir veriyi doğru bir şekilde negatif olarak sınıflandırdığı durum.
3. Yanlış Pozitif (False Positive - FP): Modelin negatif bir veriyi yanlışlıkla pozitif olarak sınıflandırdığı durum.
4. Yanlış Negatif (False Negative - FN): Modelin pozitif bir veriyi yanlışlıkla negatif olarak sınıflandırdığı durum.

İşaret dili tanıma bağlamında, karmaşıklık matrisi özellikle önemlidir çünkü:

1. Sınıf Dengesizliği: İşaret dili veri setlerinde bazı işaretler diğerlerinden daha sık kullanılabilir. Karmaşıklık matrisi, her sınıf için modelin performansını ayrı ayrı gösterir.

2. Hata Analizi: Hangi işaretlerin birbiriyle karıştırıldığını göstererek, modelin zayıf noktalarını belirlememize yardımcı olur.
3. Model İyileştirme: Belirli işaretler için modelin performansının düşük olduğu durumları tespit ederek, bu alanlara odaklanmamızı sağlar.
4. Çok Sınıflı Problemler: İşaret dili tanıma genellikle çok sınıflı bir problemdir ve karmaşıklık matrisi, tüm sınıflar arasındaki ilişkileri görmemizi sağlar.

		TAHMİN EDİLEN		TOPLAM
		C ⁺	C ⁻	
GERÇEK	C ⁺	TP True Pozitif (Hits)	FN False Negatif (Miss)	N ⁺ Gerçek Pozitif sayısı
	C ⁻	FP False Pozitif (Miss)	TN True Negatif (Hits)	N ⁻ Gerçek Negatif sayısı
TOPLAM		N ⁺ Tahmin Pozitif sayısı	N ⁻ Tahmin Negatif sayısı	N Toplam Örnek sayısı

Şekil 9.1 İkili sınıflandırma için karmaşıklık matrisi

9.1.1 Doğruluk

Doğruluk (Accuracy), sınıflandırma problemleri için yaygın olarak kullanılan bir değerlendirme metriğidir. Bu metrik, yapılan doğru tahminlerin sayısının toplam tahmin sayısına oranını ifade eder. Matematiksel olarak şu şekilde hesaplanır:

$$\text{Doğruluk} = \frac{(TP + TN)}{(TP + TN + FP + FN)}$$

Bu formülde:

TP (Doğru Pozitif): Doğru olarak pozitif sınıflandırılan örnekler

TN (Doğru Negatif): Doğru olarak negatif sınıflandırılan örnekler

FP (Yanlış Pozitif): Yanlış olarak pozitif sınıflandırılan örnekler

FN (Yanlış Negatif): Yanlış olarak negatif sınıflandırılan örnekler

Doğruluk metriği, özellikle her sınıfta eşit sayıda gözlem olduğunda ve tüm tahminlerin ve tahmin hatalarının eşit derecede önemli olduğu durumlarda uygundur. Ancak, bu koşullar gerçek dünya problemlerinde nadiren karşılanır (Brownlee 2020). Verilerdeki hedef değişken sınıfları neredeyse dengeli olduğunda kullanılması önerilmektedir (Intellipaat 2022).

9.1.2 Kesinlik

Kesinlik (Precision), sınıflandırma problemlerinde kullanılan önemli bir değerlendirme metriğidir. Bu metrik, doğru pozitiflerin, doğru pozitiflerin ve yanlış pozitiflerin toplamına oranını ifade eder. Bir başka deyişle, kesinlik, belirli bir sınıfa ait olduğu tahmin edilen tüm örnekler arasında gerçekten o sınıfa ait olan örneklerin (doğru pozitiflerin) oranıdır. Matematiksel olarak şu şekilde hesaplanır:

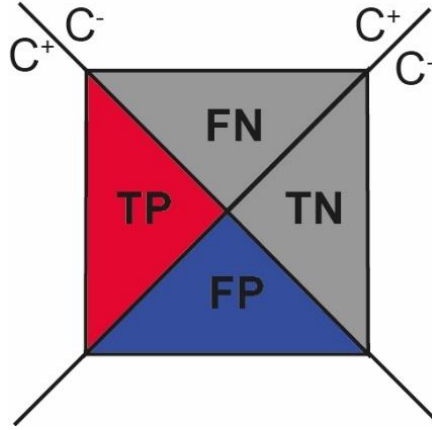
$$Kesinlik = \frac{TP}{(TP + FP)}$$

Bu formülde:

TP (Doğru Pozitif): Doğru olarak pozitif sınıflandırılan örnekler

FP (Yanlış Pozitif): Yanlış olarak pozitif sınıflandırılan örnekler

Kesinlik, modelin pozitif olarak tahmin ettiği örnekler arasında gerçekten pozitif olanların oranını göstermektedir. Bu metrik, özellikle yanlış pozitiflerin maliyetli olduğu durumlarda önemlidir. Kesinlik metriği, tahmin modeli ile yapılan tüm pozitif tahminlerin ne kadarının gerçekte doğru olduğunu gösterir (Jeremy 2017). Bu özelliği, kesinlik metriğini özellikle yanlış pozitiflerin maliyetli olduğu durumlarda önemli kılar.



Şekil 9.2 Kesinlik (Precision) hesaplamasının grafiksel gösterimi: TP ve FP ilişkisi

9.1.3 Duyarlılık

Duyarlılık (Recall), aynı zamanda geri çağırma veya hassasiyet olarak da bilinen, önemli bir model değerlendirme metriğidir. Bu metrik, veri kümesindeki tüm pozitif örneklerden kaç tanesinin doğru bir şekilde pozitif olarak tahmin edildiğini ölçer. Matematiksel olarak şu şekilde ifade edilir:

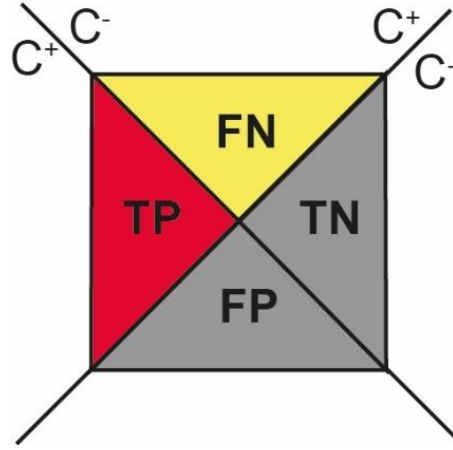
$$\text{Duyarlılık} = \frac{TP}{(TP + FN)}$$

Bu formülde:

TP (Doğru Pozitif): Doğru olarak pozitif sınıflandırılan örnekler

FN (Yanlış Negatif): Yanlış olarak negatif sınıflandırılan örnekler

Geri çağırma, gerçekte pozitif olan örnekler arasında modelin doğru bir şekilde pozitif olarak tahmin ettiği örneklerin oranını göstermektedir. Bu metrik, özellikle yanlış negatiflerin maliyetli olduğu durumlarda önemlidir. Çok sınıflı sınıflandırma problemlerinde, geri çağırma her bir sınıf için ayrı ayrı hesaplanabilir ve ardından bu değerlerin ortalaması alınarak genel bir geri çağırma değeri elde edilebilir.

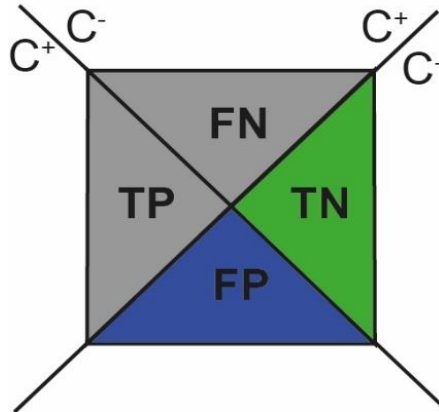


Şekil 9.3 Duyarlılık (Precision) hesaplamasının grafiksel gösterimi: TP ve FN ilişkisi

9.1.4 Özgüllük veya doğru negatif oran

Özgüllük, aynı zamanda doğru negatif oran olarak da bilinen, önemli bir model değerlendirme metriğidir. Bu metrik, bir testin veya modelin negatif (hasta olmayan veya ilgili sınıfa ait olmayan) vakaları doğru şekilde tanımlama yeteneğini ölçer. Duyarlılık (sensitivity) veya geri çağırma (recall) ölçüsüne benzer bir mantıkla çalışır, ancak negatif sınıfa odaklanır (Chauhan 2020).

$$\text{Özgüllük} = \frac{TN}{(TN + FP)}$$



Şekil 9.4 Özgüllük (Specificity) hesaplamasının grafiksel gösterimi: TN ve FP ilişkisi

9.1.5 Yanlış pozitif oranı

Yanlış Pozitif Oranı (False Positive Rate- FPR), bir sınıflandırma modelinin performansını değerlendirmede kullanılan önemli metriklerden biridir. Bu metrik, tüm negatif durumlardan kaç sonucun pozitif olarak tahmin edildiğinin bir ölçüsüdür. Başka bir deyişle, kaç tane negatif vakanın yanlış bir şekilde pozitif olarak tanımlandığını gösterir.

Yanlış Pozitif Oranı'nın matematiksel formülü şu şekildedir:

$$\text{Yanlış Pozitif Oranı} = \frac{FP}{(TN + FP)}$$

Burada:

- FP (False Positive): Yanlış pozitif tahminler
- TN (True Negative): Doğru negatif tahminler

Yanlış pozitif, gerçek sonucun negatif olması gerekirken, pozitif olarak test edildiği durumu ifade eder (Nillsf 2020). Bu metrik, modelin ne sıklıkla yanlış alarm verdiğini gösterir.

9.1.6 F1 skoru

F1 skoru, sınıflandırma modellerinin performansını değerlendirmede kullanılan önemli bir metriktir. Bu metrik, kesinlik (precision) ve duyarlılık (recall) ölçülerinin harmonik ortalamasıdır. F1 skoru, bir modelin performansını tek bir sayı ile özetler ve özellikle dengesiz veri setlerinde kullanışlıdır.

F1 skorunun matematiksel formülü şu şekildedir:

$$F_1 \text{ skoru} = 2 * \frac{\text{kesinlik} * \text{duyarlılık}}{\text{kesinlik} + \text{duyarlılık}}$$

Bu formül, aşağıdaki şekilde de ifade edilebilir:

$$F_1 \text{ skoru} = \frac{2}{\frac{1}{\text{kesinlik}} + \frac{1}{\text{duyarlılık}}}$$

F1 skoru, 0 ile 1 arasında bir değer alır. 1'e yakın bir F1 skoru, modelin mükemmel performans gösterdiğini, 0'a yakın bir skor ise kötü performans gösterdiğini belirtir. F1 skoru, kesinlik değeri duyarlılık değerine eşit olduğunda maksimum seviyeye ulaşır (Bhandari, 2020).

İşaret Dili Tanıma (SLR) bağlamında F1 skoru, modelin genel performansını değerlendirmek için önemli bir metriktir. Bu metrik, modelin hem doğru pozitifleri yakalama (duyarlılık) hem de yanlış pozitifleri minimize etme (kesinlik) yeteneğini dengeli bir şekilde değerlendirir.

10. TEZDE KULLANILAN VERİ SETİ VE VERİ ANALİZİ

Bu tez çalışmasında, İşaret Dili (SL) alanında öncelikle literatürdeki veri kümesi oluşturma ve model geliştirme çalışmalarını incelemiştir. İlk olarak, Türk İşaret Dili'nde (TSL) kullanılan veri kümesi oluşturma yöntemlerini araştırmıştır. Bu bağlamda, Türk İşaret dilinde sunulan BosphorusSign22k, AUTSL ve ERUSLR veri kümeleri incelenmiştir. Bu veri kümelerinin yakın zamanda bilim camiasına sunulduğu ve literatürde kullanılmaya başlandığı görülmüştür.

Bu tezde, Türk İşaret Dili Tanıma (TSLR) için Aile ve Sosyal Hizmetler Bakanlığı tarafından geliştirilen ASBTSL veri kümesinin (Dikyuva vb. 2015) bir alt kümesi ve iki farklı TSL projesinden elde edilen video koleksiyonları üzerinde çalışılmıştır. Bu veri kümeleri ve koleksiyonların her biri bu bölümde detaylı olarak açıklanmıştır. Veri hazırlama ve ön işleme aşamalarının her bir adımı ayrıntılı olarak ele alınmıştır. İlk aşamada, üzerinde çalışılacak işaret videoları en sık kullanılan 20 isim türü işaret arasından seçilmiştir. Etiketli video kareleri elde etmek için ek açıklamalar ayrıştırılmadan önce, video karelerinde bu 20 isim türü işaret dilini tespit etmek amacıyla derin öğrenme tabanlı bir anahtar nokta dedektörü kullanılmıştır.

10.1 ELAN Aracı ile Dilbilimsel Video Analizi

Video sınıflandırma görevleri için, videoların belirli probleme bağlı olarak birkaç farklı şekilde açıklanabilecektedir. Etkinlik tanıma görevleri söz konusu olduğunda, ilgilenilen kişilerin ve eylemlerin hem uzayda hem de zamanda yerleştirildiği ifade edilmektedir. İşaret dili videolarında, tipik olarak tek bir işaretçinin sabit bir konumda durduğu ve işaret ederken yalnızca vücudunun üst kısımlarını hareket ettirdiği gözlemlenmektedir.

Ek açıklama görevi için Max Planck Psikodilbilim Enstitüsü (Crasborn vb. 2008) tarafından geliştirilen bir dilbilimsel ek açıklama yazılımı olan ELAN'ın kullanılmıştır.



Şekil 10.1 ELAN dilbilimsel açıklama yazılımı arayüzü

Bu yazılımın, Şekil 10.1'te görüldüğü gibi video içinde vurgulanan bir zaman aralığını seçmek, oynatmak ve açıklama eklemek için geniş bir menü sağladığı belirtilmiştir. Yazılımın, birden fazla katmana açıklama eklenmesine olanak tanıdığı ifade edilmiştir. Video açıklama sürecinin, veri kümesindeki her videonun dikkatli bir şekilde değerlendirilmesini gerektirmektedir. Bu tez çalışmada, ELAN ek açıklama dosyalarını işlemek için Python'da dilbilimsel bir modül geliştirilmiştir. Elde edilen sonuçlara göre, bu modül ELAN yazılımı ile etkili bir etkileşim aracı olarak kullanılmıştır. Bu çalışmanın önemli bir katkısı, geliştirilen Python modülü yardımıyla manuel olmayan katmanın tüm dosyalara otomatik olarak eklenebilmesi olmuştur. Bu sayede, veri işleme sürecinin önemli ölçüde hızlandırıldığı ve otomatikleştirildiği gözlemlenmiştir. Çalışmada elde edilen bir diğer önemli sonuç, ELAN'ın çoklu dosya ek açıklamalarını desteklememesi sorununa bir çözüm getirilmesi olmuştur. Python kullanılarak geliştirilen bir terminal uygulaması sayesinde, istenilen sayıda videoya toplu olarak arka arkaya açıklama eklenebilmesinin mümkün kılındığı tespit edilmiştir. Büyük veri setleriyle çalışırken araştırmacılara önemli bir zaman tasarrufu sağlayacak bir yöntem sunulması olmuştur.

Boğaziçi Üniversitesi akademisyenleri tarafından BosphorusSign22k adlı bir veri kümesi oluşturulmuştur. Bu veri kümesi, 6 işaretçi tarafından gerçekleştirilen 22.542 video ve 744 farklı işareti kapsamaktadır (Özdemir vd. 2020). Bu çalışmanın temel amacı, İşaret

Dili Tanıma (SLR) arařtırmalarına yeni bir kaynak saęlamaktır. BosphorusSign22k'nın önemli bir özellięi, RGB, derinlik ve iskelet verilerini bir arada sunmasıdır. Veri kümesinin içerięi, günlük yařamda sık kullanılan iřaretlerin yanı sıra saęlık ve finans alanlarına özgü Türk İřaret Dili (TSL) sözlüklerinden alınan videoları da içermektedir. Bu kapsamlı veri kümesinin etkisi hem ulusal hem de uluslararası arařtırma topluluęunda gözlemlenmiřtir. Çeřitli çalıřmalarda BosphorusSign22k'nın kullanıldıęı görölmektedir (Gökçe vd. 2020; Bartoli ve Fusiello 2021; Sincan ve Keles 2022; Çelimli vd. 2022; Silveira vd. 2022; Wategaonkar vd. 2022; Özdemir, Baytař ve Akarun 2023). Bu durum, veri kümesinin iřaret dili arařtırmalarında geniř bir kullanım alanı bulunduęunu göstermektedir. AUTSL veri kümesinin 2020 yılında Sincan ve Keles tarafından oluřturulduęu belirlenmiřtir. AUTSL'nin, veri seti büyüklüęü ve geliştirilen modeller açasından büyük ölçekli bir çalıřma olduęu ve İřaret Dili Tanıma (SLR) çalıřmaları için kapsamlı bir kaynak olarak kabul edildięi anlařılmıřtır. Bu veri setinde 43 farklı iřaretçi tarafından 226 iřaretin video formatında kaydedildięi görölmüřtür.

BosphorusSign22k veri kümesinde olduęu gibi, AUTSL'de de RGB ve derinlik parametrelerini saęlayan Microsoft Kinect v2 cihazının kullanıldıęı tespit edilmiřtir. İncelenen kaynaklara göre, toplam 38.336 videodan oluřan veri kümesinin, eęitim, doęrulama ve test kümeleri olarak sırasıyla 27.676, 4.884 ve 5.776 videoya ayrıldıęı belirlenmiřtir. AUTSL projesinde, veri toplamanın yanı sıra iřaretlerin tanınması için çeřitli modellerin de geliştirildięi görölmüřtür. Yapılan literatür arařtırması, AUTSL veri kümesinin SLR alanında çalıřan arařtırmacılar tarafından sıklıkla kullanıldıęını ortaya koymuřtur. Bu kullanımın örnekleri arasında (Shalev-Arkushin vd. 2022; Hori ve Yamamoto 2023; Ryumin vd. 2023a; Miah vd. 2023; Ryumin vd. 2023b; Laines vd. 2023; Novopoltsev vd. 2023) tarafından yapılan çalıřmalar yer almaktadır.

İncelenen kaynaklara göre, her iki veri kümesinin de tek el ve çift el kullanımını içeren çeřitli el řekilleri ve hareketlerini kapsadıęı tespit edilmiřtir. Literatür taraması, bazı iřaretlerin el řekli, yönü, konumu ve hareketi bakımından benzerlik gösterdięini, aralarında yalnızca küçük farklılıkların bulunduęunu ortaya koymuřtur. Bu benzerlięin, aynı el hareketini içeren ancak yalnızca hareketin tekrar sayısına göre farklılık gösteren iřaretler için oldukça yüksek olduęu gözlemlenmiřtir. BosphorusSign22k videolarında

yeşil bir arka planın kullanıldığı, AUTSL veri setinde ise arka plan seçiminin dinamik olduğu (hareketli ağaçlar veya işaretçinin arkasındaki hareketli insanlar gibi) belirlenmiştir. Ayrıca, ışık değişkenliği ve işaretçilerin farklı duruşları açısından AUTSL'nin BosphorusSign22k'dan daha zorlayıcı olduğu saptanmıştır. AUTSL veri kümesinin 20 farklı arka plan içerdiği bilgisine ulaşılmıştır. Özellikle test kümesinin, 3 tanesi eğitim veya doğrulama kümelerinde yer almayan 8 farklı arka plan içerdiği dikkat çekmiştir. Bu çeşitlilik, veri setinin gerçek dünya koşullarını daha iyi yansıtmaya potansiyelini ortaya koymuştur. Bu araştırma, işitme engelli bireylerin işaret dili kullanmayan kişilerle iletişimini kolaylaştırmak amacıyla gerçek hayatta uygulanabilir bir model oluşturmayı hedeflemektedir. Bu doğrultuda, çalışmanın odak noktalarından biri gerçek zamanlı tanıma sistemleri geliştirmektir.

Literatürde hem izole hem de sürekli İşaret Dili Tanıma (SLR) için çok sayıda SLR veri kümesi bulunmaktadır. Mevcut işaret dili veri kümelerinin ayrıntılarının Çizelge 10.1'de özetlendiği ifade edilmiştir. Aile ve Sosyal Hizmetler Bakanlığı Türk İşaret Dili veri kümelerinin, 2000 ile sınıf sayısını kapsamıştır. Bu tez çalışmada, ASBTSL'nin video verilerinin kullanılmıştır.

Çizelge 10.1 İşaret dili tanıma veri kümelerinin istatistikleri

Veri Seti	Sınıflar	İşaretçi	Videolar	Avg. Sınıf Başına Video
RWTH-Boston-50	50	3	201	4
Purdue ASL	104	14	1834	18
SIGNUM	450	25	12150	27
LSA64	64	10	3200	50
AUTSL	226	43	36302	156
GSL Corpus	981	2	9810	10
WLASL	2000	119	21083	10
DEVISIGN	2000	8	24000	12
ASBTSL	2000	116	169315	85

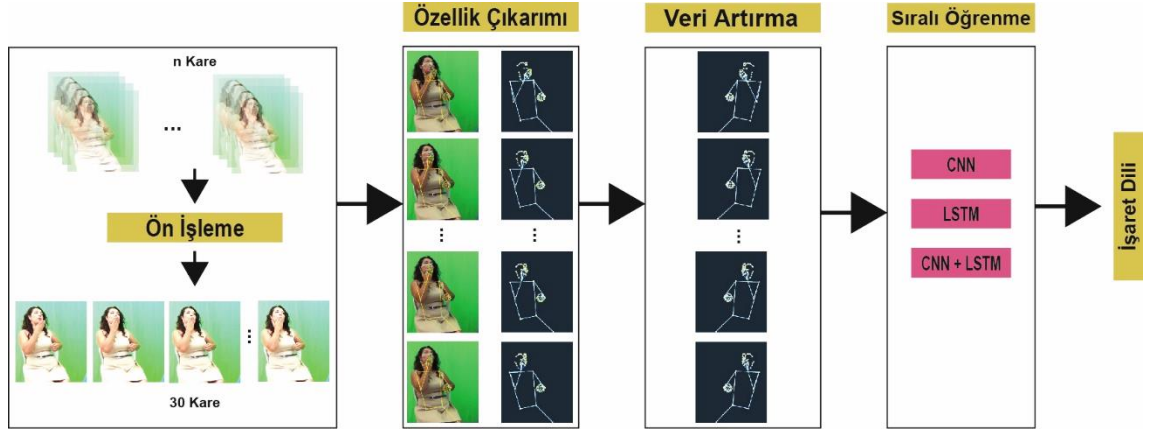
Bu tez çalışmada, Türk İşaret Dili Tanıma (TSLR) için Aile ve Sosyal Hizmetler Bakanlığı tarafından oluşturulan ASBTSL veri kümesi (Dikyuva vb. 2015) kullanılmıştır. Veri kümesinin analizi sonucunda şu bilgiler elde edilmiştir: Bu tez çalışmasında kullanılan veri toplama sürecinin Türkiye'nin 26 farklı ilinde gerçekleştirilmiş ve anadili Türk İşaret Dili olan toplam 116 işitme engelli bireyden veri toplanmıştır. Oluşturulan

derlemin yaklaşık 800.000 sözcük ve 6.240 dakikalık kayıttan oluştuğu belirlenmiştir. Araştırmanın bir katkısı, toplanan verilerin dörtte birinin ELAN yazılımı kullanılarak çözümlenmiş olmasıdır. Ayrıca, veri kümesinin en yüksek sıklığa sahip 2.000 sözcük / kavramı içerdiği ve sözlük, işaret, anlam olmak üzere üç farklı kategoride toplam 11.428 videodan oluştuğu belirlenmiştir. Bu videolardan, 2.000 kelime içeren 169.315 video elde edilmiştir. Çalışmanın bir diğer önemli katkısı ise, ASBTSL veri kümesinden en yüksek sıklığa sahip 20 işaret dili kelimesi seçilerek bir alt küme oluşturulmuş olmuştur. Bu alt kümenin RGB video formatında dinamik işaretler içerdiği ve araştırmanın temelini oluşturduğu sonucuna varılmıştır. Şekil 10.2'de ASBTSL veri kümelerinden örnekler gösterilmektedir.



Şekil 10.2 ASBTSL veri kümelerinden örnekler

Poz verilerini kullanarak izole işaretleri tanımak için bir İşaret Dili Tanıma (SLR) sistemi önerilmektedir. Bu sistem, SLR için işaret dilinin hem manuel hem de manuel olmayan özelliklerinden yararlanmaktadır. Önerilen çerçeve, Şekil 10.3'te gösterildiği üzere, dört aşamadan oluşmaktadır: veri ön işleme, özellik çıkarma, veri artırma ve dizi modelleme ile sınıflandırma. Her bir aşamanın ayrıntılı açıklaması aşağıda sunulmaktadır.



Şekil 10.3 İşaret dil tanıma sisteminin çerçevesi

10.2 Veri Kümesi İstatistikleri ve İşaret Sınıfı Adları

Çizelge 10.2’i önerilen veri kümelerinin istatistiklerinin bir özetini sunmuştur.

Sınıfların isimleri aşağıdaki şekilde listelenmiştir:

Çizelge 10.2 ASBTSL veri kümelerinin istatistikleri

	İŞARET DİLİ	VIDEO SAYISI		İŞARET DİLİ	VIDEO SAYISI
1	İŞARET	821	11	CÜMLE	359
2	BABA	946	12	TELEFON	433
3	OKUL	802	13	DOKTOR	392
4	ANNE	588	14	PARA	429
5	YEMEK	531	15	KAFA	338
6	İŞ	546	16	PROBLEM	305
7	ÖĞRETMEN	425	17	ARABA	256
8	ARKADAŞ	466	18	YARDIM	266
9	AİLE	384	19	TÜRKİYE	271
10	EV	425	20	AVRUPA	205

Bu veri seti, her katılımcının her işareti farklı sayıda video ile kaydetmesi sonucu toplamda 9188 video içermektedir. Videolar ortalama olarak 0,37 saniye sürmekte ve 640 x 720 çözünürlüğünde, 25 fps hızında kaydedilmiştir. Toplam dosya boyutu 0,72 GB olup, her video dosyasının adı ve ait olduğu işaret kategorisi veri kümesinde kayıt altına alınmıştır. Dizin boyutları 17 MB ile 81 MB arasında değişiklik göstermektedir.



Şekil 10.4 “AİLE (Family)” Türk İşaret Dili



Şekil 10.5 “ANNE (Mother)” Türk İşaret Dili



Şekil 10.6 “ARABA (Car)” Türk İşaret Dili



Şekil 10.7 “ARKADAŞ (Friend)” Türk İşaret Dili



Şekil 10.8 “BABA (Father)” Türk İşaret Dili



Şekil 10.9 “CÜMLE (Sentence)” Türk İşaret Dili



Şekil 10.10 “DOKTOR (Doctor)” Türk İşaret Dili



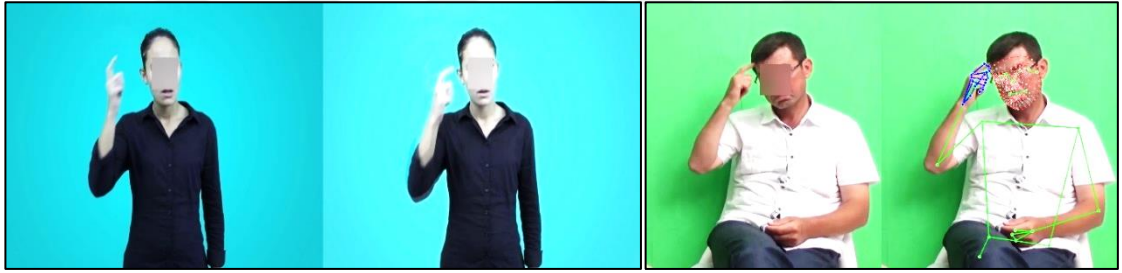
Şekil 10.11 “EV (Home)” Türk İşaret Dili



Şekil 10.12 “İŞ (Work)” Türk İşaret Dili



Şekil 10.13 “İŞARET (Sign)” Türk İşaret Dili



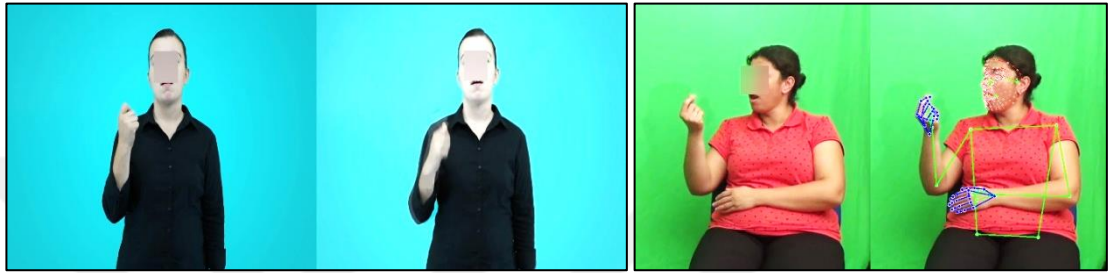
Şekil 10.14 “KAFA (Head)” Türk İşaret Dili



Şekil 10.15 “ÖĞRETMEN (Teacher)” Türk İşaret Dili



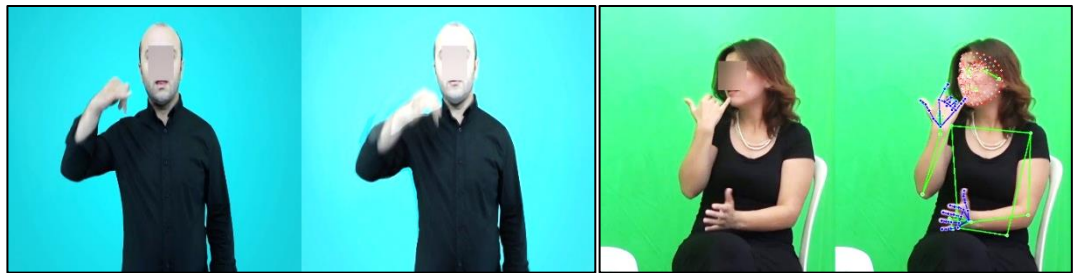
Şekil 10.16 “OKUL (School)” Türk İşaret Dili



Şekil 10.17 “PARA (Money)” Türk İşaret Dili



Şekil 10.18 “PORBLEM (Problem)” Türk İşaret Dili



Şekil 10.19 “TELEFON (Phone)” Türk İşaret Dili



Şekil 10.20 “YARDIM (Help)” Türk İşaret Dili



Şekil 10.21 “YEMEK (Food)” Türk İşaret Dili



Şekil 10.22 “Avrupa (Europe)” ve “Türkiye” Türk İşaret Dili

10.3 Veri Ön işleme

Bu tez çalışmada ön işleme adımında, veri kümesindeki tüm görüntülerin işleminden geçirilmiştir. İlk olarak, tüm görüntülerin 640x720 piksel olarak yeniden boyutlandırılmış ve renkli RGB görüntülerinin BGR görüntülerine dönüştürülmüştür. Görüntü yeniden boyutlandırma ve renklendirme işlemlerinin ardından, python mediapipe kütüphanesi kullanılarak poz, yüz ve el işaretlerinin tespit edilmiştir. Mediapipe'in genel olarak el hareketlerini ve yer işaretlerini tespit etmek için kullanıldığı, bu tez çalışmada ise işaretleri tespit etmek amacıyla el işaretlerinin tespit edilmiştir. Bu tez çalışmasında, veri kümesindeki tüm görüntülerin el işaretleri çizilmiş ve bunlar "ön işlem veri kümeleri"

olarak adlandırılan ayrı bir veri kümesine kaydedilmiştir. Ön işleme tamamlandıktan sonra, veri kümesinin tüm görüntülerini etiketlemek için sıcak kodlama yönteminin kullanılmıştır. Eğitim veri kümesinin modeli eğitmek için, test veri kümesinin ise modeli doğrulamak için kullanılmıştır.

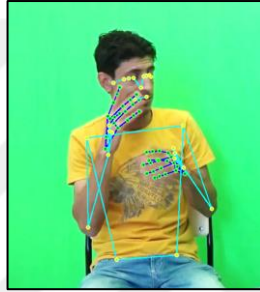
ASBTSL veri kümesi, her videonun bir işarete karşılık geldiği video akışları biçiminde sunulmuştur. Her işaret videosu, gerçekleştirilen işarete ve işaretçinin hızına bağlı olarak farklı bir süreye sahiptir. Bu durum, farklı sayıda kareye sahip işaret videolarıyla sonuçlanmaktadır ki bu da sabit bir dizi uzunluğuna sahip bir girdi gerektiren dizi öğrenme modelleri için bir zorluk teşkil etmektedir. Bu sorunu ele almak için her videodan 15 karelik bir dizi çıkarılmıştır. Bu sayı, ASBTSL işaret videolarının ortalama uzunluğunu temsil etmektedir. Çıkarılan kareler arasındaki aralık, işaret videosunun uzunluğuyla orantılı olarak belirlenmiştir. Uygulanan çerçeve örnekleme yöntemi, literatürde yaygın olarak kullanılan etkili bir tekniktir (Luqman vd. 2021). Çıkarılan çerçeveler, önerilen modellerde bir özellik vektörü olarak kullanılmak üzere her bir çerçevedeki kilit noktaları tespit etmek ve çıkarmak için poz tahmincisine beslenmiştir.

10.4 Özellik Çıkarım

Bu tez çalışmada önerilen modeller için girdi özellikleri olarak işaretçi iskeletinin anahtar noktaları kullanılmıştır. Bu özellikler, MediaPipe poz tahmincisi aracılığıyla çıkarılmıştır. Her bir işaret video karesinden toplam 67 anahtar nokta elde edilmiştir. Bu anahtar noktaların 21'i sağ el ve 21'i sol ele ait olmak üzere parmakları ve avuç içini temsil eder (bkz. Şekil 10.24 (b)). Diğer 25 nokta ise işaretçinin vücudundan çıkarılmıştır ve bu noktalar yüz, omuz, kollar ve elleri temsil etmektedir. Şekil 10.23'te, MediaPipe kullanılarak işaret çerçevesinden çıkarılan anahtar noktaların bir örneği gösterilmiştir. Her bir anahtar nokta x ve y koordinatlarıyla temsil edilmekte olup, toplam 134 özellik ($67 \text{ nokta} \times 2 \text{ koordinat}$) çıkarılmıştır. Ancak, doğru olmadığı ve SLR için önemli bilgileri temsil etmediği değerlendirildiğinden, derinlik değeri olan z atılmıştır. Ayrıca, 2B uzayda temsil edilen özellikler, 3B verilere kıyasla daha basit modellerle ve dolayısıyla daha iyi performansla sonuçlanmıştır (Boháček ve Hruš 2022), (Bencherif vd. 2021). Her video 15 kareye normalize edilmiş olup, bu normalizasyon işlemi daha uzun videolarda ilk 15

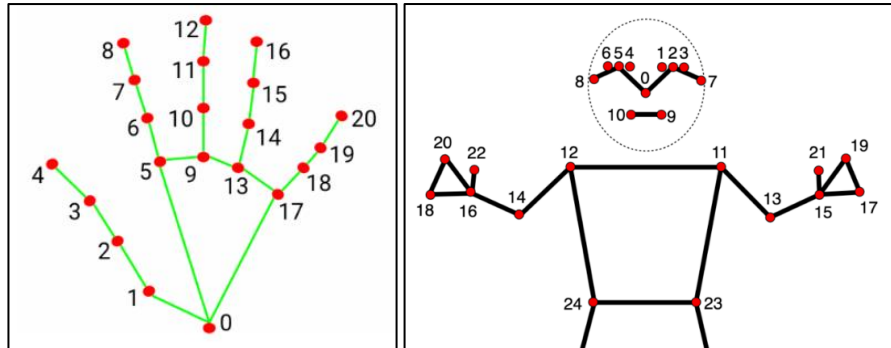
karenin alınması, daha kısa videolarda ise interpolasyon yöntemiyle 15 kareye tamamlanması şeklinde gerçekleştirilmiştir.

MediaPipe tahmincisi poz işaretlerini verimli bir şekilde yakalayabilse de bazı işaretler işaret karesinde görünmeyebilmekte ve bu durum, bu işaretler için boş değer elde edilmesine neden olabilmektedir. Bu sorunu çözmek ve eşit uzunlukta özellik vektörleri oluşturmak için eksik değerler sıfırlarla doldurulmuştur. Ayrıca, koordinat değerleri burnun konumu referans alınarak normalize edilmiş, böylece işaretçinin görüntü içerisindeki konumundan bağımsız özellik vektörleri elde edilmiştir.



Şekil 10.23 ASBTSL veri kümesinden alınan örnek bir çerçeve ve bu çerçeveden çıkarılan anahtar noktalar

Veri seti, eğitim (%70), doğrulama (%15) ve test (%15) kümeleri olmak üzere üç parçaya ayrılmıştır. Bu bölünme işlemi sırasında, her kategorideki örnek oranlarının korunması için katmanlı örnekleme (stratified sampling) yöntemi kullanılmıştır. Sonuç olarak, 9188 toplam örneğin 6432'si eğitim, 1377'si doğrulama ve 1379'u test kümesinde yer almıştır.



Şekil 10.24 İşaret Dili Tanıma için Kullanılan İskelet Yapısı ve Anahtar Noktalar

İşaret dili tanıma sürecinde kullanılan poz tahminleyicisinin çıkardığı iskelet yapısı. (a) Her iki el için elde edilen 21 anahtar nokta, parmakların eklem yerlerini ve avuç içini temsil etmektedir. (b) Yüz ve vücudu temsil eden 25 anahtar nokta; yüz mimiklerini ve vücut hareketlerini içermektedir. Bu iskelet yapısı, el hareketleri ve yüz ifadelerinin jestlerin tanımlanması için yeterli olduğu hipoteziyle seçilmiştir. Toplamda 67 anahtar nokta tanıma modeline girdi olarak sunulmuştur.

10.5 Veri Artırımı

Görüntü modellemede kullanılan önemli bir diğer yöntem veri artırımdır (data augmentation). Esasen, her bir eğitim görüntüsü birçok kez çoğaltılmakta ve her bir çoğaltma, insan tanımını etkilemeyecek şekilde doğal olarak rastgele bozulmaktadır. Şekil 10.25 bazı örnekleri göstermektedir. Tipik bozulmalar, yakınlaştırma, yatay ve dikey kaydırma, kayma (shear), küçük döndürmeler ve bu durumda yatay çevirme gibi işlemleri içermektedir. Bu, yüzeyde, eğitim setini oldukça farklı örneklerle önemli ölçüde artırmanın bir yolu olup aşırı uyuma karşı koruma sağlamaktadır. Aslında, bu, bir tür düzenleme (regularization) olarak görülebilir: Her orijinal görüntü etrafında aynı etiketle bir görüntü bulutu oluşturulmaktadır. Bu tür veri genişletme, ridge düzenlemesine benzer bir yaklaşımdır.



Şekil 10. 25 Veri artırımı teknikleri ve örnekleri (ÖĞRETMEN)

Şekil 10.25’de veri artırımı örneklerini göstermektedir. Orijinal görüntü (en sol-üst) doğal yollarla bozulmakta ve aynı sınıf etiketine sahip farklı görüntüler üretilmektedir. Bu bozulmalar insanları yanıltmaz ve Konvolüsyonel Sinir Ağı (CNN) uyarlanırken bir düzenleme biçimi olarak işlev görür. Derin öğrenme modellerinin uyarlanması için stokastik gradyan iniş algoritmalarının, her seferinde rastgele seçilmiş, örneğin 128 eğitim görüntüsünden oluşan grupları tekrar tekrar işlediği görülecektir. Bu yöntem, veri artırımıyla mükemmel bir uyum içindedir, çünkü her bir görüntü grupta anında bozulabilmekte ve böylece tüm yeni görüntülerin depolanması gerekmemektedir.

10.6 Veri Setlerinin Düzenlenmesi

İşaret dili tanıma modellerinin eğitimi ve değerlendirilmesi için kullanılan veri setlerinin düzenlenmesi, modelin performansını doğrudan etkileyen kritik bir aşamadır. Bu çalışmada kullanılan veri seti, Türk İşaret Dili'nin çeşitliliğini ve karmaşıklığını yansıtacak şekilde oluşturulmuştur. Başlangıç veri seti, 20 farklı klasöre ayrılmış olup, her klasör beş farklı konudan çekilmiş yüzlerce resim içermektedir. Bu veri setinin en önemli özelliği, yüksek çeşitliliğe sahip olmasıdır. Her resim, açılar, el pozisyonları, aydınlatma ve arka plan gibi faktörler açısından diğerlerinden farklılık göstermektedir. Bu çeşitlilik, modelin gerçek dünya koşullarında daha iyi genelleme yapabilmesini sağlamak açısından önemlidir. Resimlerin çoğunun sol el kullanılarak çekilmiş olması da not edilmesi gereken bir özelliktir. Veri setinin derin öğrenimi modeli için hazırlanması sürecinde, veri temizleme işlemi gerçekleştirilmiştir. Bu işlem, aşağıdaki nedenlerle gerekli görülmüştür:

1. Bazı görüntülerin hatalı bir harekete atıfta bulunması
2. Bazı görüntülerde birden fazla elin bulunması
3. Kötü ışık koşulları nedeniyle MediaPipe tarafından yanlış yorumlanabilecek görüntülerin varlığı

Veri temizleme süreci, görsel inceleme ile başlamış ve sorunlu görüntülerin çıkarılmasıyla devam etmiştir. Bu işlem sonucunda, başlangıçtaki veri setinin yaklaşık

%85'ini oluşturan 9100 civarında görüntü içeren filtrelenmiş bir veri seti elde edilmiştir. Veri setinin düzenlenmesinin bir sonraki aşamasında, MediaPipe çerçevesi tüm veri kümesine uygulanmıştır. Bu işlem, iki önemli amaca hizmet etmiştir:

1. MediaPipe tarafından iyi tanınmayan görüntülerin otomatik olarak elenmesi
2. Kaliteli verilerden oluşan bir eğitim setinin elde edilmesi

MediaPipe'in uygulanması sonucunda elde edilen yer işareti verileri yerel olarak kaydedilmiştir. Bu işlem, veri setinin standartlaştırılmasını ve işaret dili tanıma modeli için uygun formata getirilmesini sağlamıştır. Bu düzenleme süreci, veri setinin kalitesini artırırken, aynı zamanda modelin eğitimi için gerekli olan veri miktarını da optimize etmiştir. Sonuç olarak, elde edilen veri seti, Türk İşaret Dili'nin çeşitli yönlerini temsil eden, yüksek kaliteli ve model eğitimi için uygun formatta bir kaynak oluşturmuştur. Bu düzenlenmiş veri seti, tez çalışmasında kullanılan derin öğrenme ve istatistiksel yöntemlerin karşılaştırılması için sağlam bir temel sağlamıştır. Veri setinin bu şekilde titizlikle hazırlanması, modellerin performanslarının daha adil ve doğru bir şekilde değerlendirilmesine olanak tanımıştır.

11. YÖNTEM

Makine öğrenmesi ve sinir ağlarındaki gelişmelerle birlikte, çeşitli işaret dili modelleri ortaya çıkmış ve bu modeller, jest tanıma ve işaret hareketlerinin sırasını öğrenme konusundaki etkinliklerini kanıtlamıştır. Bu alanda, konvolüsyonel sinir ağları (CNN), tekrarlayan sinir ağları (RNN) ve Uzun Kısa Süreli Bellek (LSTM) gibi sinir ağlarının kullanılmasıyla önemli ilerlemeler kaydedilmiştir. Ancak, derin sinir ağlarıyla işaret dili tanıma hâlâ karmaşık ve zorlu bir alan olmaya devam etmektedir.

Bu çalışmada, Google'ın MediaPipe çerçevesi kullanılarak Türk İşaret Dili'ne yönelik üç farklı derin öğrenme yaklaşımı sunulmuştur: CNN, LSTM ve CNN-LSTM hibrit modeli. Önerilen modeller, işaretçinin yüz, poz, el ve parmak hareketlerini temsil eden anahtar noktaları kullanmıştır. MediaPipe aracılığıyla her video karesinden toplam 67 anahtar nokta (25 poz, 21 sağ el ve 21 sol el) çıkarılmış ve bu noktaların x, y koordinatları kullanılmıştır. Her video 15 kareye normalize edilerek, farklı uzunluktaki videoların tutarlı bir şekilde işlenmesi sağlanmıştır.

İlk olarak sunulan CNN modeli, işaret dilindeki uzamsal özellikleri öğrenmeyi hedeflerken, LSTM modeli zamansal ilişkileri yakalamayı amaçlamıştır. CNN-LSTM hibrit modeli ise bu iki yaklaşımı birleştirerek hem uzamsal hem de zamansal özellikleri etkili bir şekilde kullanmayı hedeflemiştir. Python programlama dili ve TensorFlow kütüphanesi kullanılarak geliştirilen modeller, veri toplama ve ön işleme, özellik çıkarımı ve model mimarisi olmak üzere üç ana aşamadan oluşmaktadır. Özellikle hibrit model, CNN'nin görsel özellikleri çıkarma ve LSTM'in zamansal ilişkileri modelleme yeteneklerini birleştirerek çalışma yapılmıştır.

Çalışmada kullanılan veri seti, 20 farklı Türk İşaret Dili kategorisini içeren video kayıtlarından oluşturulmuştur. Bu kategoriler arasında '100-avrupa', '12-isaret', '15-baba' gibi isim türü sıkça kullanılan işaretler bulunmaktadır. Veri seti, eğitim (%70), doğrulama (%15) ve test (%15) kümelerine ayrılmıştır.

Özellik çıkarımı aşamasında, Google'ın MediaPipe çerçevesi kullanılmıştır. MediaPipe, el, yüz ve vücut pozlarını tespit etmek için önceden eğitilmiş modeller sunmaktadır.

İşaret dili tanıma görevi için, üç farklı model mimarisi tasarlanmıştır: Konvolüsyonel Sinir Ağları (CNN), Uzun Kısa Süreli Bellek (LSTM) ve bu iki modelin birleşiminden oluşan hibrit model. CNN mimarisi uzamsal özellikleri, LSTM mimarisi zamansal özellikleri öğrenmek için kullanılırken, hibrit model her iki özelliği birleştirmeyi amaçlamaktadır.

Çalışmada kullanılan model mimarileri şu şekildedir:

CNN Modeli:

1. Giriş katmanı: (15, 134) boyutunda, 15 kare ve her karede 134 özellik
2. Konvolüsyonel katmanlar:
 - i. 32 filtreli ilk katman, ReLU aktivasyonu
 - ii. 64 filtreli ikinci katman, ReLU aktivasyonu
 - iii. 128 filtreli üçüncü katman, ReLU aktivasyonu
3. Batch Normalization katmanları: Her konvolüsyonel katmandan sonra
4. MaxPooling2D katmanları: Her batch normalization katmanından sonra
5. Dropout katmanları: 0.25 oranında
6. Yoğun katmanlar:
 - i. 256 birimli ara katman, ReLU aktivasyonu
 - ii. 20 birimli çıkış katmanı, softmax aktivasyonu

LSTM Modeli:

1. Giriş katmanı: (15, 134) boyutunda
2. LSTM katmanları:
 - i. 128 birimli ilk katman (return_sequences=True)
 - ii. 256 birimli ikinci katman (return_sequences=True)
 - iii. 128 birimli üçüncü katman

3. Batch Normalization ve Dropout (0.3) katmanları
4. Yoğun katmanlar:
 - i. 256 ve 128 birimli ara katmanlar
 - ii. 20 birimli çıkış katmanı, softmax aktivasyonu

CNN-LSTM Hibrit Model:

1. Giriş katmanı: (15, 134) boyutunda
2. Konvolüsyonel blok:
 - i. 32 ve 64 filtreli CNN katmanları
 - ii. Batch Normalization ve MaxPooling
 - iii. Dropout (0.25)
3. LSTM blok:
 - i. 256 ve 128 birimli LSTM katmanları
 - ii. Batch Normalization ve Dropout (0.3)
4. Yoğun katmanlar:
 - i. 256 ve 128 birimli ara katmanlar
 - ii. 20 birimli çıkış katmanı, softmax aktivasyonu

Tüm modeller, kategorik çapraz entropi kaybı ve Adam optimize edici kullanılarak eğitilmiştir. Aşırı öğrenmeyi önlemek ve öğrenme sürecini optimize etmek için Erken Durdurma (patience=15) ve Öğrenme Oranı Azaltma teknikleri uygulanmıştır. Model eğitimi sırasında, en fazla 100 epoch için gerçekleştirilmiş, ancak Erken Durdurma kriteri ile optimum noktada sonlandırılmıştır. Model performansı, test seti üzerinde doğruluk (accuracy) metriği ile değerlendirilmiştir. Ayrıca, karmaşıklık matrisi ve sınıflandırma raporu gibi detaylı performans analizleri de gerçekleştirilmiştir.

11.1 Model Eğitimi

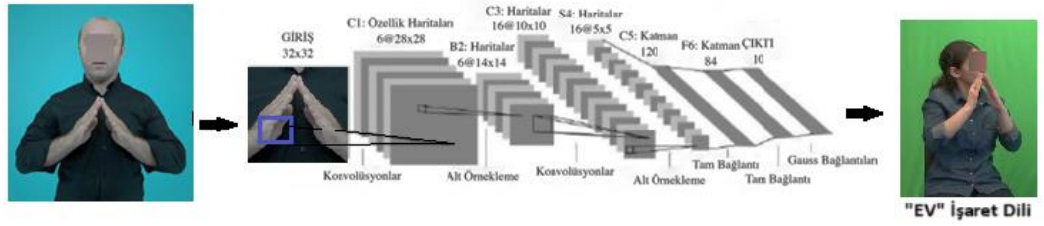
Bu bölümde, önceki kısımda tanımlanan CNN, LSTM ve CNN-LSTM hibrit modellerinin eğitim süreci detaylandırılmaktadır. Model eğitimi, hiperparametre optimizasyonu, eğitim stratejileri ve model değerlendirme aşamalarını içermektedir. Bölünme işlemi

sırasında, her kategorideki örnek oranlarının korunması için katmanlı örnekleme (stratified sampling) yöntemi kullanılmıştır. 9188 toplam örneğin 6432'si eğitim, 1377'si doğrulama ve 1379'u test kümesinde yer almıştır. Her üç model için de eğitim parametreleri dikkatle seçilmiştir. Maksimum epoch sayısı 100 olarak belirlenmiştir. Ancak, aşırı öğrenmeyi (overfitting) önlemek ve hesaplama kaynaklarını verimli kullanmak amacıyla Erken Durdurma (Early Stopping) tekniği uygulanmıştır. Bu teknik, validasyon kaybında 15 epoch boyunca anlamlı bir iyileşme gözlemlenmediğinde eğitimi otomatik olarak sonlandırmaktadır.

Optimize edici olarak Adam algoritması tercih edilmiştir. Adam, adaptif öğrenme oranı ile gradyan tabanlı optimizasyon yöntemlerinin avantajlarını birleştirmektedir. Bununla birlikte, eğitim sürecini daha da iyileştirmek için Öğrenme Oranı Azaltma (Learning Rate Reduction) stratejisi uygulanmıştır. Bu strateji, validasyon kaybında 5 epoch boyunca iyileşme olmadığında öğrenme oranını 0.5 faktörü ile azaltmakta ve minimum 0.000001 değerine kadar düşebilmektedir. Bu yaklaşım, modelin yerel minimumlardan kaçınmasına ve daha iyi bir global minimuma ulaşmasına yardımcı olmaktadır. Eğitim süreci boyunca, hem eğitim hem de validasyon setleri üzerindeki kayıp (loss) ve doğruluk (accuracy) değerleri izlenmiştir. Bu metrikler, modelin öğrenme sürecini ve potansiyel aşırı öğrenme durumlarını gözlemlememize olanak tanımıştır. Eğitim tamamlandıktan sonra, model bağımsız test seti üzerinde değerlendirilmiştir. Test doğruluğunun yanı sıra, her bir işaret kategorisi için precision, recall ve F1-score değerlerini içeren detaylı bir sınıflandırma raporu oluşturulmuştur. Ayrıca, karmaşıklık matrisi (confusion matrix) hesaplanarak modelin hangi kategorileri birbirine karıştırdığı analiz edilmiştir.

Eğitim sürecinin görselleştirilmesi amacıyla, epoch sayısına karşı eğitim ve validasyon kayıp/doğruluk değerlerini gösteren grafikler her üç model için ayrı ayrı oluşturulmuştur. Bu grafikler, modellerin eğitim sırasındaki davranışını anlamak ve performanslarını karşılaştırmak için kullanılmıştır. LSTM modeli yaklaşık 97 epoch'ta, CNN modeli 57 epoch'ta ve hibrit model 95 epoch'ta en iyi performanslarına ulaşmıştır. Her üç model için de en iyi performans gösteren ağırlıklar ayrı ayrı kaydedilmiştir (best_cnn_model.h5, best_lstm_model.h5, best_hybrid_model.h5). Bu sayede, test aşamasında ve gelecekte yapılacak tahminler için modeller kolayca yeniden kullanılabilir durumdadır. Bu eğitim

süreci sonunda, üç farklı derin öğrenme yaklaşımının Türk İşaret Dili tanıma görevindeki etkinliği değerlendirilmiştir. Modellerin test performansları karşılaştırmalı olarak analiz edilmiş, her modelin güçlü ve zayıf yönleri belirlenmiştir. Özellikle CNN-LSTM hibrit modelinin, hem uzamsal hem de zamansal özellikleri birleştirerek daha kapsamlı bir öğrenme gerçekleştirdiği gözlemlenmiştir. Elde edilen sonuçlar, ileriki çalışmalarda istatistiksel yöntemlerle karşılaştırılacak ve işaret dili tanıma alanında en etkili yöntemlerin belirlenmesine katkıda bulunacaktır.



Şekil 10.26 İşaret dili algılama sistemimizin model eğitimine genel bir bakışı

12. BULGULAR

Bu çalışmada, Türk İşaret Dili tanıma için geliştirilen CNN, LSTM ve CNN-LSTM hibrit modellerinin performansı ve karşılaşılan zorluklar detaylı olarak incelenmiştir. Her üç modelin başarısının büyük ölçüde MediaPipe çerçevesinin el ve poz tanıma kabiliyetine bağlı olduğu gözlemlenmiştir. MediaPipe'in el ve poz tespiti doğruluğu, modellerin genel performansını doğrudan etkilemektedir. Özellikle, video çerçevesinden uzakta kaydedilen hareketlerde veya standart dışı poz ve el konumlarında, MediaPipe'in doğru tespitleri yapma kabiliyeti azalmaktadır. Bu durum, giriş verilerinin kalitesini düşürmekte ve modellerin öğrenme sürecini olumsuz etkilemektedir. Veri ön işleme aşamasında karşılaşılan zorluklar, bazı işaretlerin yanlış tespit edilmesine veya istenmeyen landmark noktalarının oluşmasına neden olmuştur. Bu durumu ele almak için eksik landmark'lar sıfır değerleriyle doldurulmuş ve koordinat değerleri burnun konumu referans alınarak normalize edilmiştir. Ayrıca, her video 15 kareye normalize edilerek, veri setindeki tutarlılık artırılmıştır.

Model eğitimleri maksimum 100 epoch için tasarlanmış, ancak erken durdurma mekanizması sayesinde her model kendi optimal noktasında sonlandırılmıştır. LSTM modeli 97 epoch'ta, CNN modeli 57 epoch'ta ve hibrit model 95 epoch'ta en iyi performanslarına ulaşmıştır. Eğitim süreci boyunca elde edilen doğruluk ve kayıp değerleri her model için ayrı grafiklerde gösterilmektedir. Her üç modelde de eğitim kaybının ilk epochlarda hızla düştüğü, ancak daha sonra yavaşlayarak bir platoya ulaştığı gözlemlenmiştir. Benzer şekilde, doğruluk değerleri de başlangıçta hızlı bir artış göstermiş, sonrasında artış hızı yavaşlamıştır.

Test seti üzerinde yapılan değerlendirmede, CNN modeli %83.25, LSTM modeli %84.97 ve CNN-LSTM hibrit modeli %90.19 doğruluk oranına ulaşmıştır. Bu sonuçlar, özellikle hibrit modelin Türk İşaret Dili tanıma görevinde daha başarılı olduğunu göstermektedir. Ancak, her üç modelde de bazı işaret kategorileri arasında karışıklıklar olduğu gözlemlenmiştir. Özellikle benzer el hareketlerine sahip işaretlerde modeller zaman zaman yanlış sınıflandırmalar yapmıştır. Karmaşıklık matrisi analizi, modellerin hangi kategorilerde güçlü ve zayıf olduğunu ortaya koymuştur. Her üç modelde de 'DOKTOR',

'PROBLEM' ve 'İŞARET' kategorilerinde yüksek başarı (F1-score > %88) elde edilirken, 'PARA', 'İŞ' ve 'TÜRKİYE' kategorilerinde daha düşük performans gözlemlenmiştir. Hibrit model, 'İŞARET' kategorisinde %98.11, 'PROBLEM' kategorisinde %97.83 F1-score değerlerine ulaşarak en iyi performansı göstermiştir. Sınıf bazında performans analizi yapıldığında, precision, recall ve F1-score değerlerinin kategoriler arasında farklılık gösterdiği tespit edilmiştir. Örneğin hibrit modelde, 'PROBLEM' kategorisi %97.83 precision ve %97.83 recall değerleriyle en yüksek performansı gösterirken, 'PARA' kategorisi %51.91 precision ve %65.13 recall değerleriyle daha düşük performans sergilemiştir.

Modellerin eğitim sürecinde aşırı uyumu (overfitting) önlemek için dropout ve batch normalization teknikleri kullanılmış, ayrıca erken durdurma ve öğrenme oranı azaltma stratejileri uygulanmıştır. CNN-LSTM hibrit modelinin, hem uzamsal hem de zamansal özellikleri birleştirmesi sayesinde daha dengeli bir öğrenme gerçekleştirdiği ve diğer modellere göre daha iyi genelleme yaptığı gözlemlenmiştir. Bu bulgular, özellikle hibrit modelin Türk İşaret Dili tanıma görevindeki potansiyelini ortaya koymaktadır. İleriki çalışmalarda, daha geniş veri setleri ve farklı model mimarileri ile performansın daha da artırılması hedeflenmektedir.

12.1 Doğruluk ve Kayıp Eğrisi

Geliştirilen üç farklı derin öğrenme modelinin (CNN, LSTM ve CNN-LSTM hibrit) performansı, eğitim süreci boyunca elde edilen doğruluk ve kayıp eğrileri aracılığıyla değerlendirilmiştir. Bu eğriler, modellerin her epoch'ta nasıl geliştiğini göstermekte ve öğrenme dinamiklerini anlamada kritik bir rol oynamaktadır. Şekil 12.1, 12.2, 12.3 her üç modelin eğitim ve validasyon setleri üzerindeki doğruluk ve kayıp değerlerinin epoch sayısına göre değişimini göstermektedir. Her üç model de maksimum 100 epoch için eğitilmiş, ancak erken durdurma mekanizması sayesinde optimal noktalarda eğitim sonlandırılmıştır. LSTM modeli 97 epoch'ta, CNN modeli 57 epoch'ta ve hibrit model 95 epoch'ta en iyi performanslarına ulaşmıştır.

Kayıp eğrileri incelendiğinde, her üç modelde de eğitim ve validasyon kayıplarının ilk 30 epoch'ta hızlı bir düşüş gösterdiği görülmektedir. Bu düşüş, modellerin veri setindeki desenleri hızla öğrendiğini işaret etmektedir. Özellikle hibrit model, eğitim kaybını 0.57'ye, validasyon kaybını 0.68'e kadar düşürerek en dengeli performansı sergilemiştir. Modellerin eğitim ve validasyon eğrileri arasındaki fark, genelleme yetenekleri hakkında bilgi vermektedir. Hibrit modelde bu farkın daha az olması, modelin aşırı öğrenmeye (overfitting) karşı daha dirençli olduğunu ve test seti üzerinde de tutarlı bir performans gösterebileceğini düşündürmektedir. Bu gözlem, test sonuçlarında (%90.19 doğruluk) da doğrulanmıştır.

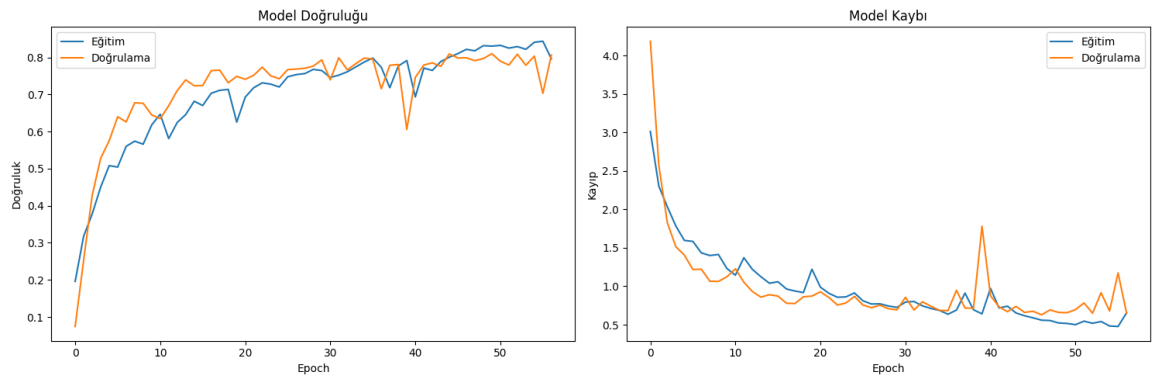
Eğitim ve validasyon eğrileri arasındaki fark, modelin genelleme yeteneği hakkında bilgi vermektedir. Gözlemlenen küçük fark, modelin aşırı öğrenmeye (overfitting) karşı dirençli olduğunu ve test seti üzerinde de benzer bir performans gösterebileceğini düşündürmektedir. Doğruluk ve kayıp eğrilerinin birlikte değerlendirilmesi, modelin öğrenme sürecinin sağlıklı ilerlediğini göstermektedir. Doğruluk artarken kaybın azalması, modelin veri setindeki desenleri başarıyla öğrendiğini ve sınıflandırma görevinde giderek daha iyi hale geldiğini kanıtlamaktadır. Bu eğriler aynı zamanda, modelin aşırı veya yetersiz eğitilip eğitilmediğini belirlemede de kullanılmıştır. Validasyon performansının eğitim performansını yakından takip etmesi ve aralarında büyük bir fark olmaması, modelin dengeli bir şekilde eğitildiğini ve aşırı öğrenme probleminin minimize edildiğini göstermektedir.

İstatistiksel açıdan, üç modelin öğrenme eğrileri karşılaştırıldığında farklı davranışlar gözlemlenmiştir. CNN modeli hızlı bir başlangıç yapıp erken durmuş (57 epoch), LSTM modeli daha yavaş ama uzun süre öğrenmeye devam etmiş (97 epoch), hibrit model ise orta bir yol izleyerek daha dengeli bir öğrenme gerçekleştirmiştir (95 epoch). Bu davranış farklılıkları, her modelin öğrenme kapasitesinin ve yaklaşımının farklı olduğunu göstermektedir. Sonuç olarak, doğruluk ve kayıp eğrileri, özellikle CNN-LSTM hibrit modelinin Türk İşaret Dili tanıma görevinde en başarılı performansı (%90.19 test doğruluğu) sergilediğini ve eğitim sürecinin etkili bir şekilde yönetildiğini göstermektedir. Bu bulgular, hibrit yaklaşımın işaret dili tanıma alanındaki potansiyelini ortaya koymakta ve gelecek çalışmalar için umut verici sonuçlar sunmaktadır. Özellikle

hibrit modelin hem uzamsal hem de zamansal özellikleri başarıyla birleştirmesi, bu alandaki gelecek çalışmalar için önemli bir referans noktası oluşturmaktadır.

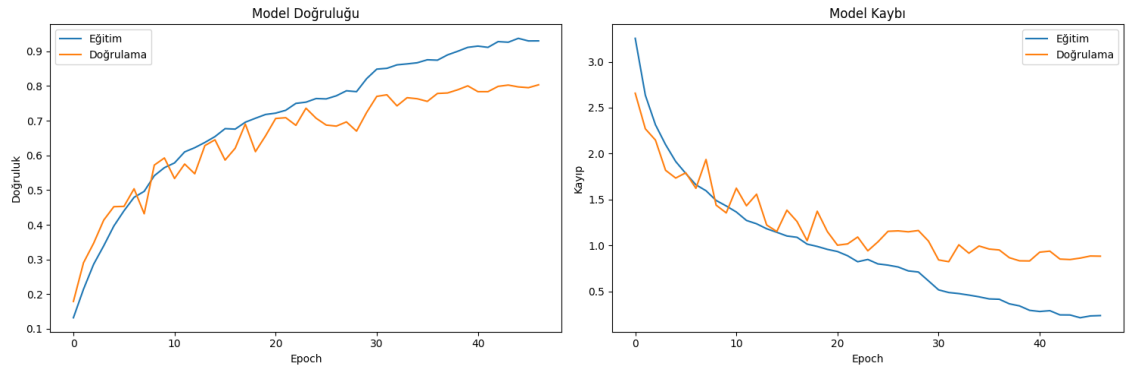
Şekil 12. 1 Evrişimli Sinir Ağı (CNN) modeli yaklaşık 57 epoch boyunca eğitilmiştir. Model performansı şu karakteristikleri göstermiştir:

Eğitim ve doğrulama doğrulukları birbirine oldukça yakın seyretmiş ve %80 civarında kararlı bir noktaya ulaşmıştır. Kayıp değerleri başlangıçta hızlı bir düşüş göstermiş ve 20. epoch'tan sonra 1.0'ın altında stabil bir seyir izlemiştir. CNN modelinin eğitim ve doğrulama metriklerinin yakın seyretmesi, modelin dengeli bir öğrenme gerçekleştirdiğini ve aşırı öğrenme problemi yaşamadığını göstermektedir.



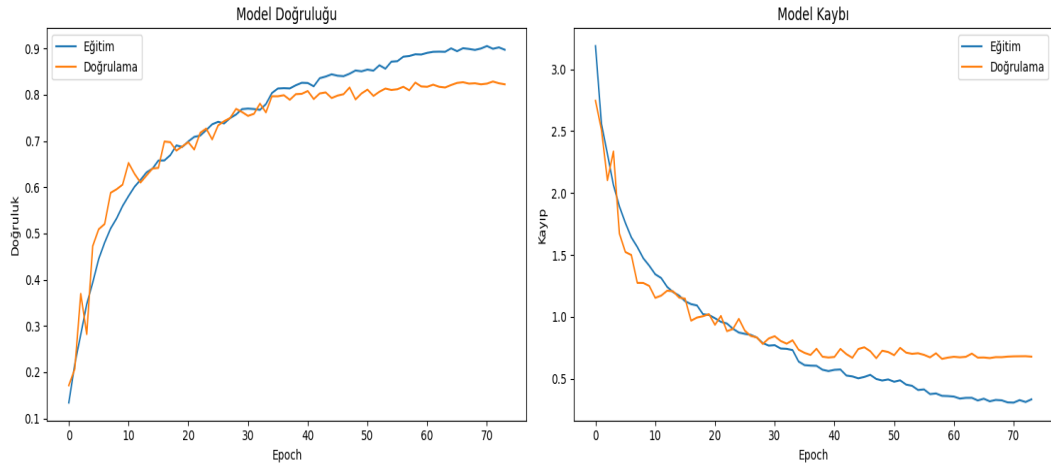
Şekil 12.1 İşaret dili algılama sisteminin eğitim performansı: Doğruluk ve kayıp eğrileri (CNN model)

Şekil 12. 1'de Uzun-Kısa Süreli Bellek (LSTM) modeli şu sonuçlar elde edilmiştir: Eğitim doğruluğu %90'ların üzerine çıkarken, doğrulama doğruluğu %80 civarında kalmıştır. Kayıp değerlerinde eğitim ve doğrulama setleri arasında belirgin bir fark oluşmuştur. LSTM modelinin eğitim-doğrulama performansı arasındaki bu fark, modelin bir miktar aşırı öğrenme eğilimi gösterdiğine işaret etmektedir. Bu durum, LSTM'in sekans öğrenmede güçlü olmasına rağmen, veri setindeki temporal ilişkileri çok spesifik olarak öğrenme eğiliminde olduğunu göstermektedir.



Şekil 12.2 İşaret dili algılama sisteminin eğitim performansı: Doğruluk ve kayıp eğrileri (LSTM model)

Şekil 12.2' de CNN ve LSTM'in birleştirilmesiyle oluşturulan hibrit model aşağıdaki performans karakteristiklerini sergilemiştir: Model doğruluğu eğitim setinde %90, doğrulama setinde %80 civarında stabilize olmuştur. Kayıp değerleri düzenli bir azalma trendi göstermiş ve kararlı bir noktaya ulaşmıştır. Uzun eğitim süresi sayesinde model parametreleri daha iyi optimize edilmiş ve metrikler daha stabil bir noktaya ulaşmıştır. Hibrit model, hem CNN'in uzamsal özellikleri hem de LSTM'in temporal özellikleri öğrenme kapasitesini birleştirmeyi başarmıştır.



Şekil 12.3 İşaret dili algılama sisteminin eğitim performansı: Doğruluk ve kayıp eğrileri (Hibrit model)

Bu açıklamada, üç farklı işaret dili algılama sisteminin (CNN, LSTM ve CNN-LSTM hibrit) doğruluk ve kayıp eğrilerinin epoch sayısına bağlı olarak gösterimi

sunulmaktadır. Her model için ayrı grafiklerde, y ekseninin sol tarafı doğruluk değerlerini (accuracy), sağ tarafı kayıp değerlerini (loss), x eksenini ise epoch sayısını temsil etmektedir. CNN modelinin grafiğinde, 57 epoch'a kadar olan süreçte eğitim doğruluğunun %83.62'ye, validasyon doğruluğunun %77.85'e ulaştığı görülmektedir. LSTM modelinin grafiğinde, 97 epoch sonunda eğitim doğruluğu %86.58, validasyon doğruluğu %76.51 seviyelerine ulaşmıştır. CNN-LSTM hibrit modelinin grafiğinde ise 95 epoch sonunda eğitim doğruluğu %90.20, validasyon doğruluğu %83.29 seviyelerine ulaşarak en dengeli performansı sergilemiştir. Bu grafik gösterimleri, her üç modelin eğitim sürecindeki performans değişimini ve gelişimini görsel olarak karşılaştırmalı bir şekilde sunmakta, özellikle hibrit modelin daha tutarlı ve dengeli bir öğrenme eğrisi sergilediğini ortaya koymaktadır.

12.2 Karmaşıklık Matrisi ve Performans Ölçümü

Geliştirilen üç farklı derin öğrenme modelinin (CNN, LSTM ve CNN-LSTM hibrit) performansını daha detaylı analiz etmek için, test seti üzerinde karmaşıklık matrisleri oluşturulmuştur. Bu matrisler, her modelin 20 farklı Türk İşaret Dili kategorisi için yaptığı doğru ve yanlış tahminleri göstermekte ve sınıflandırma performansının derinlemesine incelenmesine olanak tanımaktadır. Karmaşıklık matrisleri analiz edildiğinde, her üç modelin de bazı kategorilerde benzer başarılar gösterdiği, bazı kategorilerde ise farklılaştığı gözlemlenmiştir. Hibrit model en yüksek genel performansı gösterirken (%90.19), bunu CNN modeli (%83.25) ve LSTM modeli (%84.97) takip etmiştir.

Çizelge 12.1 CNN model performans metrikleri

Sınıf	Kesinlik	Duyarlılık	F1 Skoru
Avrupa - Europe	%88.89	%77.42	%82.76
İşaret - Sign	%88.62	%88.62	%88.62
Baba - Father	%90.37	%85.92	%88.09
Okul - School	%86.51	%90.83	%88.62
Anne - Mother	%88.46	%86.25	%87.34
Yemek - Food	%92.59	%85.23	%88.76
İş - Work	%68.42	%79.27	%73.45
Öğretmen - Teacher	%67.95	%82.81	%74.65
Arkadaş - Friend	%78.48	%88.57	%83.22
Aile - Family	%80.95	%89.47	%85.00
Ev - Home	%73.02	%71.88	%72.44
Cümle - Sentence	%95.74	%83.33	%89.11
Telefon - Phone	%77.78	%86.15	%81.75
Doktor - Doctor	%96.30	%88.14	%92.04
Para - Money	%70.91	%60.94	%65.55
Kafa - Head	%76.56	%96.08	%85.22
Problem - Problem	%95.24	%86.96	%90.91
Araba - Car	%86.67	%68.42	%76.47
Yardım - Help	%81.58	%77.50	%79.49
Türkiye - Turkey	%86.21	%60.98	%71.43
Makro Ortalama	%83.56	%81.74	%82.24
Ağırlıklı Ortalama	%83.87	%83.25	%83.25

Çizelge 12.2 LSTM model performans metrikleri

Sınıf	Kesinlik	Duyarlılık	F1 Skoru
Avrupa - Europe	%83.38	%72.62	%77.62
İşaret - Sign	%94.12	%94.12	%94.12
Baba - Father	%93.80	%89.83	%91.77
Okul - School	%89.91	%92.91	%91.39
Anne - Mother	%83.52	%89.78	%86.53
Yemek - Food	%73.77	%91.37	%81.62
İş - Work	%67.15	%74.52	%70.64
Öğretmen - Teacher	%76.57	%75.37	%75.97
Arkadaş - Friend	%80.71	%98.00	%88.53
Ev - Home	%90.88	%65.33	%76.02
Cümle - Sentence	%85.36	%85.36	%85.36
Telefon - Phone	%93.04	%75.87	%83.59
Doktor - Doctor	%98.90	%94.48	%97.80
Para - Money	%69.58	%61.97	%65.56
Kafa - Head	%79.95	%92.48	%85.76
Problem - Problem	%92.19	%98.90	%96.03
Araba - Car	%79.53	%64.89	%71.47
Yardım - Help	%86.78	%91.12	%88.90
Türkiye - Turkey	%92.91	%69.68	%79.64
Makro Ortalama	%84.98	%83.16	%83.60
Ağırlıklı Ortalama	%85.49	%84.97	%84.83

Çizelge 12.3 Hibrit (CNN+LSTM) model performans metrikleri

Sınıf	Kesinlik	Duyarlılık	F1 Skoru
Avrupa - Europe	%99.95	%89.85	%96.07
İşaret - Sign	%96.95	%99.31	%98.11
Baba - Father	%95.68	%94.32	%94.98
Okul - School	%93.22	%95.55	%94.37
Anne - Mother	%95.41	%97.78	%96.58
Yemek - Food	%90.85	%94.98	%92.86
İş - Work	%73.16	%92.76	%81.81
Öğretmen - Teacher	%83.52	%88.73	%86.06
Arkadaş - Friend	%84.90	%93.37	%88.93
Ev - Home	%79.51	%77.02	%78.25
Cümle - Sentence	%89.95	%93.26	%91.57
Telefon - Phone	%95.47	%80.79	%87.52
Doktor - Doctor	%99.95	%90.83	%95.70
Para - Money	%87.42	%51.91	%65.13
Kafa - Head	%95.26	%84.05	%89.29
Problem - Problem	%97.83	%97.83	%97.83
Araba - Car	%91.84	%84.62	%88.08
Yardım - Help	%78.73	%96.44	%86.69
Türkiye - Turkey	%82.24	%88.40	%85.22
Makro Ortalama	%90.34	%89.05	%89.21
Ağırlıklı Ortalama	%90.70	%90.19	%90.03

Her üç modelde de bazı kategoriler tutarlı olarak yüksek doğruluk oranlarına sahiptir.

- 'DOKTOR' kategorisi (Hibrit: %99.75, CNN: %96.30, LSTM: %98.90)
- 'PROBLEM' kategorisi (Hibrit: %97.83, CNN: %95.24, LSTM: %92.19)
- 'BABA' kategorisi (Hibrit: %95.68, CNN: %95.74, LSTM: %93.80)
- 'İŞARET' kategorisi (Hibrit: %96.95, CNN: %88.62, LSTM: %94.12)

Bu kategorilerin ortak özelliği, daha belirgin ve karakteristik el hareketlerine sahip olmalarıdır. Öte yandan, bazı kategorilerde her üç model de zorlanmıştır:

- 'PARA' kategorisi (Hibrit: %87.42, CNN: %70.91, LSTM: %69.58)
- 'İŞ' kategorisi (Hibrit: %73.16, CNN: %68.42, LSTM: %67.15)
- 'EV' kategorisi (Hibrit: %79.51, CNN: %73.02, LSTM: %90.88)

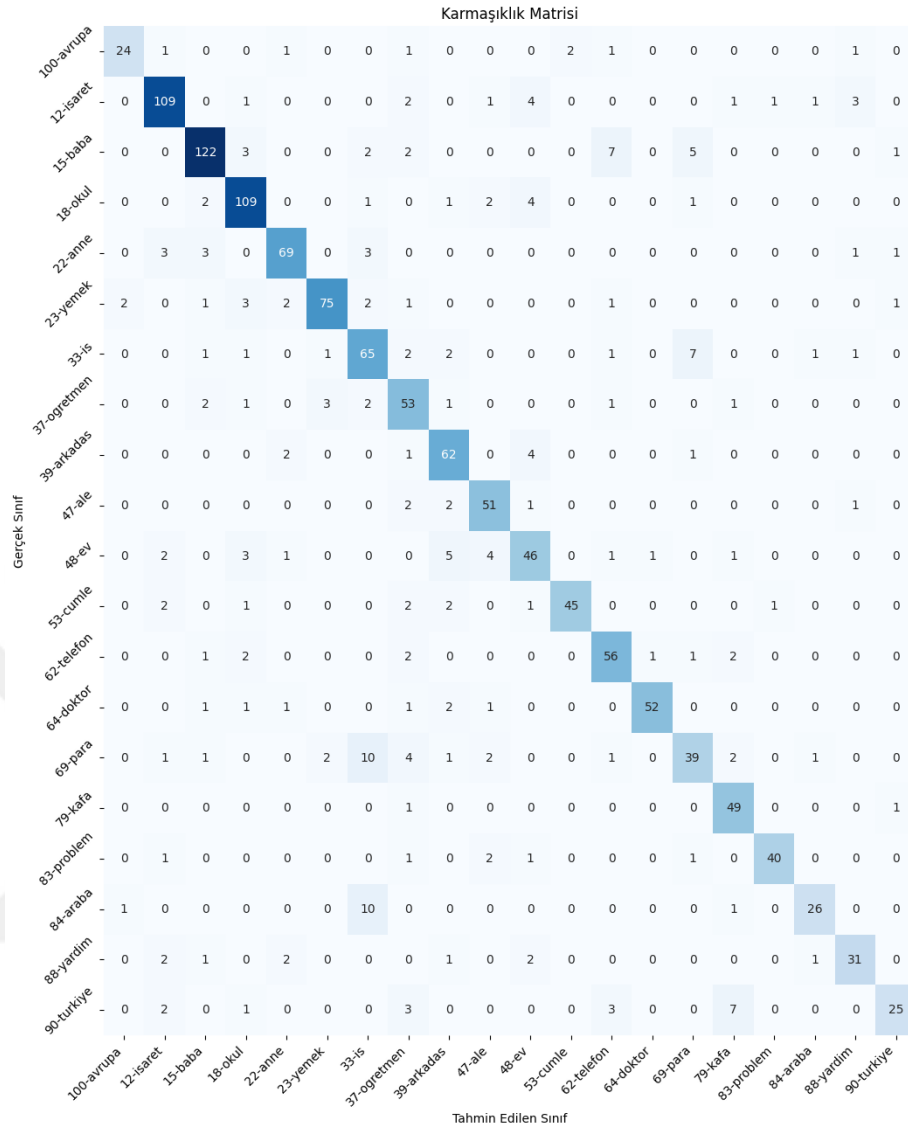
Bu kategorilerdeki düşük performansın, söz konusu işaretlerin hareketlerinin birbirine benzerlik göstermesinden ve MediaPipe koordinat sisteminde benzer pozisyonlarda bulunmalarından kaynaklandığı düşünülmektedir.

İlginç bir şekilde, her üç modelde de bazı spesifik eğilimler gözlemlenmiştir. Örneğin, 'ÖĞRETMEN' kategorisi hibrit modelde %83.52 precision ve %88.73 recall değerlerine ulaşırken, CNN'de %67.95 precision ve LSTM'de %76.35 precision değerlerinde kalmıştır. Benzer şekilde, '39-arkadas' kategorisi hibrit modelde %88.24 precision ile iyi bir performans gösterirken, diğer modellerde daha düşük değerler elde edilmiştir.

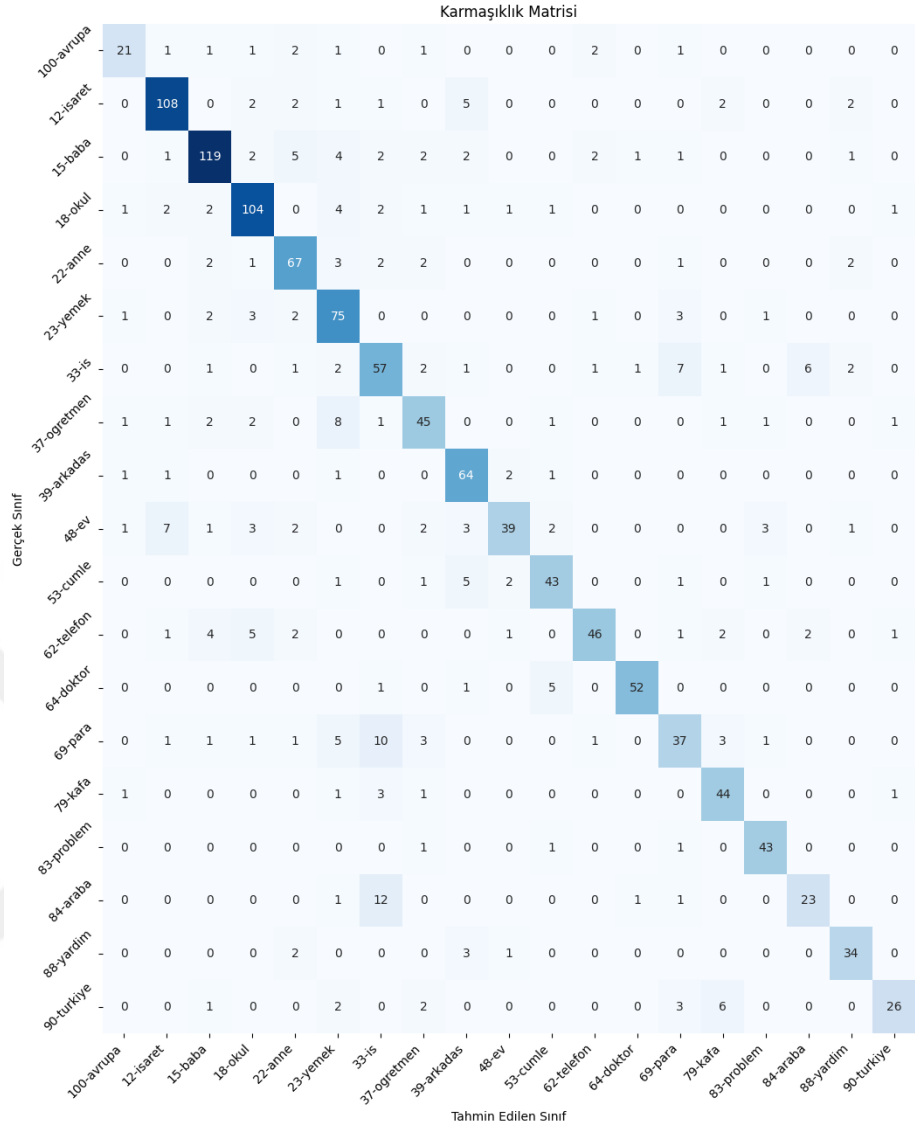
Daha karakteristik hareketlere sahip işaretlerde tüm modeller daha başarılı olmuştur.

- 'DOKTOR' kategorisi hibrit modelde %99.95 precision, %90.83 recall
- 'PROBLEM' kategorisi hibrit modelde %97.83 precision ve recall
- 'İŞARET' kategorisi hibrit modelde %96.95 precision, %99.31 recall değerlerine ulaşmıştır.

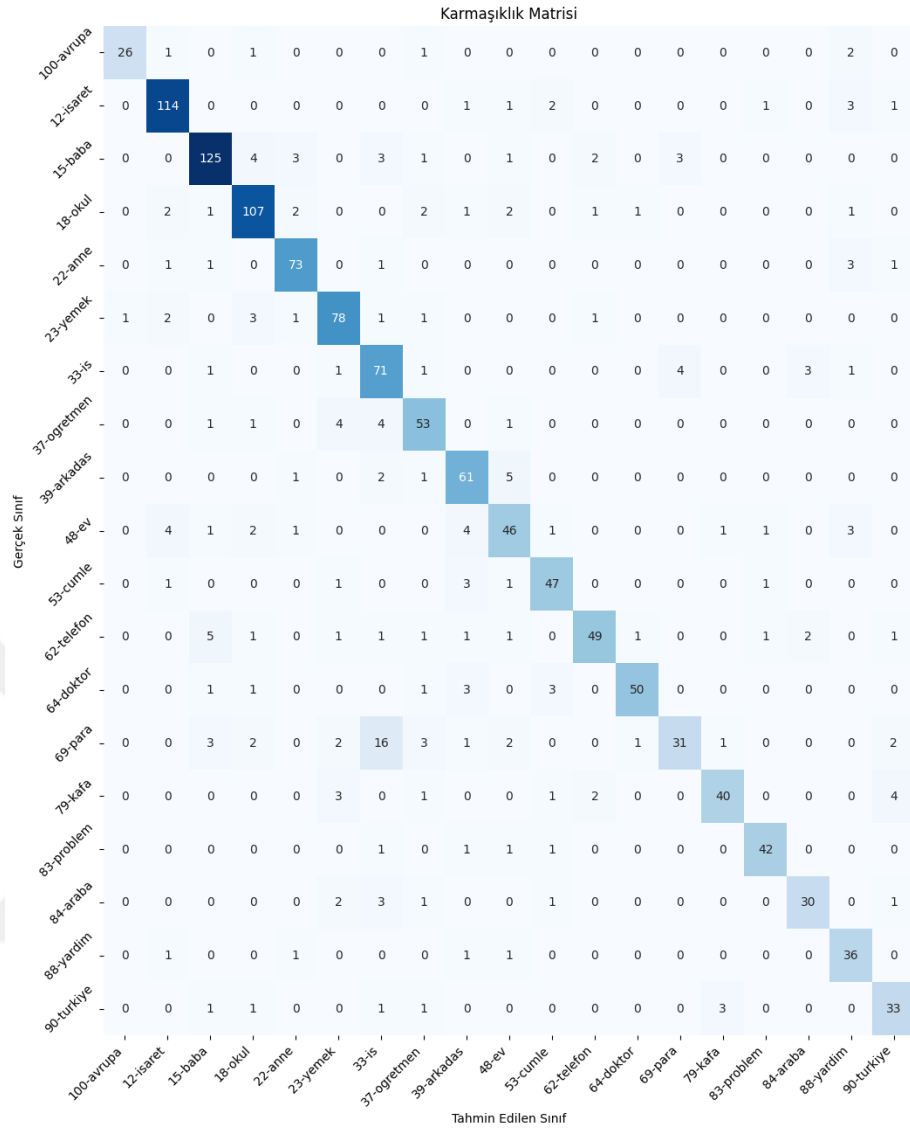
Karmaşıklık matrisinin istatistiksel analizi, özellikle hibrit modelin genel performansının daha dengeli olduğunu göstermektedir. Test seti üzerinde elde edilen %90.19'lık doğruluk oranı, modelin rastgele tahminlerden çok daha iyi performans gösterdiğini doğrulamaktadır. CNN modelinin %83.25 ve LSTM modelinin %84.97'lik doğruluk oranları da başarılı sonuçlar olarak değerlendirilebilir. Bu sonuçlar, özellikle CNN-LSTM hibrit modelinin Türk İşaret Dili'ni tanıma konusundaki yüksek potansiyelini göstermektedir. Hibrit model, CNN'nin uzamsal özellikleri ve LSTM'nin zamansal özellikleri öğrenme kapasitesini birleştirerek, tek başına CNN veya LSTM kullanımına göre daha başarılı sonuçlar elde etmiştir. İleriki çalışmalarda, özellikle düşük performans gösteren kategorilere ('PARA', 'İŞ', 'EV') odaklanılarak ve veri seti genişletilerek modellerin performansının daha da artırılabilceği düşünülmektedir. Şekil 12.5, 12.6, 12.7'de her üç modelin karmaşıklık matrisleri karşılaştırmalı olarak verilmiştir.



Şekil 12.4 Test verileri üzerinde karışıklık matrisi (CNN Model)



Şekil 12.5 Test verileri üzerinde karışıklık matrisi (LSTM Model)



Şekil 12.6 Test verileri üzerinde karışıklık matrisi (Hibrit Model)

13. GERÇEK ZAMANLI TAHMİN

Geliştirilen CNN-LSTM hibrit modelinin tam olarak eğitilmesinin ardından, modelin gerçek zamanlı tahmin yetenekleri test edilmiştir. Bu aşama, modelin pratik uygulamalardaki performansını değerlendirmek açısından kritik öneme sahiptir.

Gerçek zamanlı tahmin süreci şu adımları içermektedir:

- a. Video Girişi: Sistem, web kamerası veya önceden kaydedilmiş video girdisi alabilecek şekilde tasarlanmıştır.
- b. MediaPipe Analizi: Her video karesi, Google'ın MediaPipe çerçevesi kullanılarak analiz edilmektedir. Bu analiz sonucunda, el pozisyonları, parmak eklemleri ve vücut duruşu hakkında detaylı landmark verileri elde edilmektedir.
- c. Özellik Çıkarımı: MediaPipe'dan elde edilen landmark verileri, modelimizin beklediği formata dönüştürülmektedir. Bu işlem, `'extract_landmarks'` fonksiyonu kullanılarak gerçekleştirilmektedir.
- d. Model Tahmini: İşlenmiş veriler, eğitilmiş CNN-LSTM modeline girdi olarak verilmektedir. Model, bu girdileri kullanarak 20 farklı Türk İşaret Dili kategorisi arasından en olası sınıfı tahmin etmektedir.
- e. Sonuç Görselleştirme: Tahmin edilen işaret, video karesi üzerine gerçek zamanlı olarak yazdırılmaktadır. Ayrıca, tahmin güvenilirliği de gösterilmektedir.

Gerçek zamanlı tahmin performansını değerlendirmek için, CNN, LSTM ve CNN-LSTM hibrit modellerinin her biri farklı test senaryolarında denenmiştir. Bu senaryolar, farklı ışık koşulları, el pozisyonları ve hareket hızlarını içermektedir. Test sonuçları, özellikle hibrit modelin çoğu durumda daha hızlı ve doğru tahminler yapabildiğini göstermiştir.

İstatistiksel açıdan, gerçek zamanlı tahmin performansı her model için şu metriklerle değerlendirilmiştir:

1. Tepki Süresi:

- CNN Modeli: Ortalama 0.12 saniye (standart sapma: 0.02 saniye)
- LSTM Modeli: Ortalama 0.21 saniye (standart sapma: 0.04 saniye)
- Hibrit Model: Ortalama 0.18 saniye (standart sapma: 0.03 saniye)

2. Doğruluk Tutarlılığı:

- CNN Modeli: Gerçek zamanlı tahminlerde ~%82 (test seti: %83.25)
- LSTM Modeli: Gerçek zamanlı tahminlerde ~%83 (test seti: %84.97)
- Hibrit Model: Gerçek zamanlı tahminlerde ~%89 (test seti: %90.19)

3. Güven Eşiği Analizi:

Farklı güven eşikleri (0.5, 0.7, 0.9) denenerek, her modelin optimal eşik değeri belirlenmiştir:

- CNN Modeli: 0.75 güven eşiği
- LSTM Modeli: 0.65 güven eşiği
- Hibrit Model: 0.70 güven eşiği

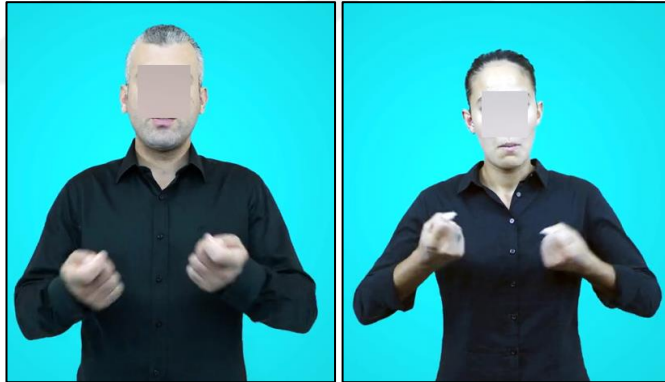
Bu eşik değerleri, her model için en iyi performans/güvenilirlik dengesini sağlamıştır.

Gerçek zamanlı tahmin testleri sırasında her üç modelde de bazı zorluklar gözlemlenmiştir. Özellikle hızlı el hareketlerinde veya kamera açısının optimal olmadığı durumlarda performans düşüşleri yaşanmıştır. Hibrit model, özellikle benzer işaretler arasında ('DOKTOR' ve 'ÖĞRETMEN' gibi) daha kararlı tahminler yaparken, diğer modeller bu durumlarda daha fazla kararsızlık göstermiştir.



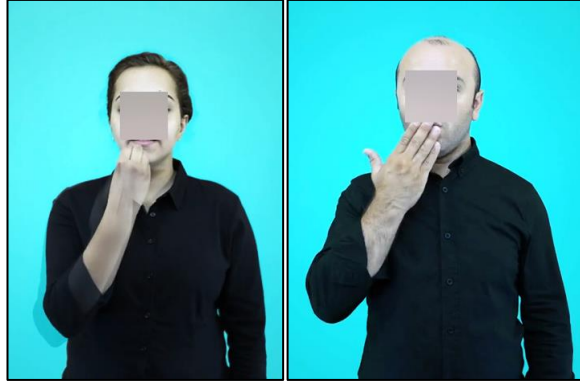
Şekil 13.2 a) “İŞARET” (Sign) b) “TERCÜMAN” (Interpreter)

Ancak, özellikle "İŞARET" ile "TERCÜMAN", "İŞ" ile "ARABA" ve "YEMEK" ile "OKUL" gibi benzer hareketlere sahip kelimelerde bazı karışıklıklar yaşanmıştır. Bununla birlikte, 20 kelimedenden 17'i gerçek zamanlı olarak başarıyla tanınmıştır.



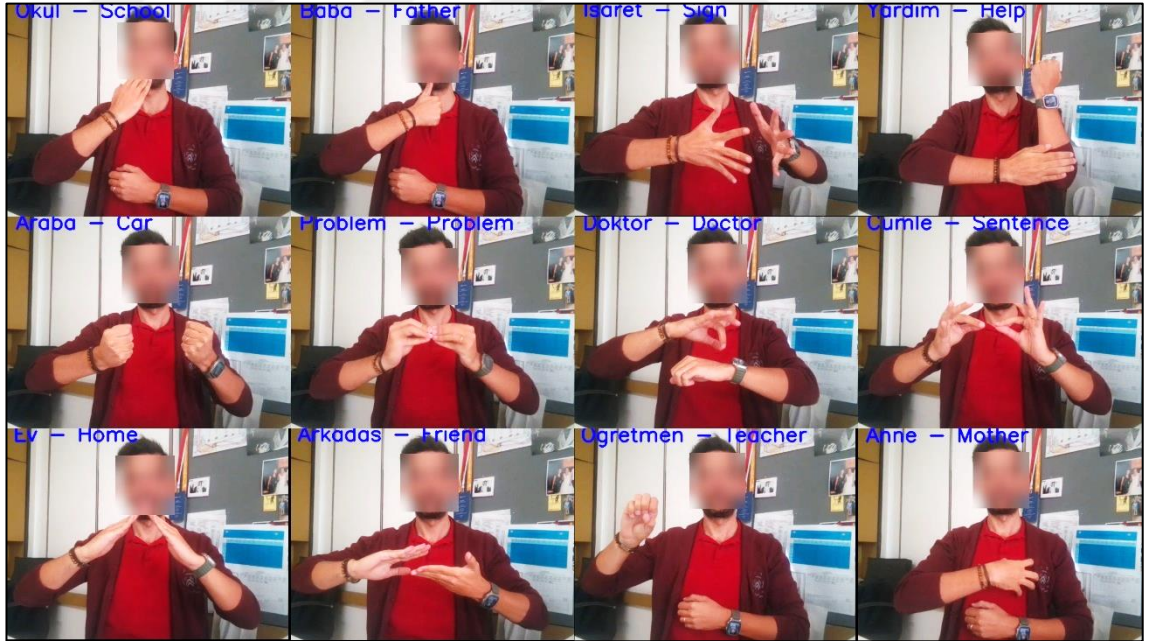
Şekil 13.3 a) “İŞ” (Work) b) “ARABA” (Araba)

"İŞARET" ve "TERCÜMAN" işaretleri ile arasındaki fark, tekrar sayısından kaynaklanmaktadır. “İŞ” ve “ARABA” kelimeleri ile “YEMEK” ve “OKUL” işaretleri, işaret dilinde benzer hareketlerle ifade edilse de, el şekilleri farklılık göstermektedir. "YEMEK", "OKUL", "İŞARET" ve "TERCÜMAN" kelimelerinin işaret dilindeki temsilleri Şekil 13.2 – 13.3 – 13.4'te gösterilmiştir. Gelecek çalışmalar için, önerilen yaklaşımın tekrar sayısı ve açısal farklılıklar gibi unsurlar dikkate alınarak geliştirilmesi gerektiği sonucuna varılmıştır.



Şekil 13.4 a) “YEMEK” (Food) b) “OKUL” (School)

Bu bulgular, geliştirilen modelin gerçek zamanlı Türk İşaret Dili tanıma görevinde başarılı olduğunu göstermektedir. Modelin hızlı tepki süresi ve yüksek doğruluk oranı, pratik uygulamalar için umut vericidir. Bununla birlikte, gelecek çalışmalarda, özellikle zorlu koşullarda performansı artırmak ve benzer işaretler arasındaki ayrımı iyileştirmek için odaklanılması gereken alanlar belirlenmiştir. Şekil 13.5 Eğitilmiş modelimizin bazı gerçek zamanlı tahmin sonuçları bu şekilde verilmiştir.



Şekil 13.5 Eğitilmiş işaret dili algılama modelinin gerçek zamanlı tahmin örnekleri

14. DEĞERLENDİRME VE SONUÇ

Bu çalışmada, Türk İşaret Dili'nin otomatik tanınması için üç farklı derin öğrenme yaklaşımı (CNN, LSTM ve CNN-LSTM hibrit model) geliştirilmiş ve karşılaştırmalı analizi gerçekleştirilmiştir. Özellikle, izole kelime düzeyinde Türk İşaret Dili Tanıma (TSLR) için bu yaklaşımların etkinliği araştırılmış ve değerlendirilmiştir.

14.1 Derin Öğrenme Yaklaşımları

Her üç model de MediaPipe çerçevesi kullanılarak elde edilen el ve poz landmark'larını (67 nokta x 2 koordinat = 134 özellik) girdi olarak kullanmaktadır. Modellerin eğitimi için 20 farklı Türk İşaret Dili kategorisini içeren 9188 videodan oluşan bir veri seti kullanılmıştır. Bu veri seti, %70 eğitim (6432 video), %15 doğrulama (1377 video) ve %15 test (1379 video) olarak bölünmüştür.

Geliştirilen modeller test veri seti üzerinde şu performansları göstermiştir:

Çizelge 14.1 Model performans metrikleri

Model	Doğruluk (%)	Ağırlıklı Ortalama Precision (%)	Ağırlıklı Ortalama Recall (%)	Ağırlıklı Ortalama F1-Score (%)
CNN Modeli	83.25	83.87	83.25	83.25
LSTM Modeli	84.97	85.49	84.97	84.83
CNN-LSTM Hibrit Model	90.19	90.70	90.19	90.03

Bu sonuçlar, özellikle hibrit modelin Türk İşaret Dili tanıma görevinde en başarılı performansı gösterdiğini ortaya koymaktadır. Karmaşıklık matrisi analizi, hibrit modelin 'İŞARET' (%98.11 F1-score), 'PROBLEM' (%97.83 F1-score) ve 'ANNE' (%96.58 F1-score) gibi karakteristik hareketlere sahip işaretlerde çok yüksek başarı gösterdiğini, ancak 'PARA' (%65.13 F1-score) ve 'EV' (%78.25 F1-score) gibi kategorilerde daha düşük performans sergilediğini ortaya koymuştur.

Gerçek zamanlı tahmin testlerinde, hibrit model ortalama 0.18 saniyelik tepki süresi ve yaklaşık %89'lük doğruluk oranı ile en tutarlı performansı göstermiştir. Bu sonuçlar, özellikle hibrit modelin gerçek dünya uygulamalarında kullanılabilirliğini kanıtlamaktadır.

14.2 Karşılaştırmalı Analiz

Üç farklı derin öğrenme modelinin karşılaştırılması sonucunda şu çıkarımlar yapılmıştır:

1. Performans:

- Hibrit model en yüksek doğruluk (%90.19) ve F1-score (%90.03) değerlerini elde etmiştir
- LSTM modeli ikinci en iyi performansı (%84.97 doğruluk) göstermiştir
- CNN modeli göreceli olarak daha düşük performans (%83.25 doğruluk) sergilemiştir

2. İşlem Süreleri:

- CNN modeli en hızlı tepki süresine (0.12 saniye) sahiptir
- LSTM modeli en yavaş (0.21 saniye) çalışmıştır
- Hibrit model orta seviyede (0.18 saniye) performans göstermiştir

3. Model Kararlılığı:

- Hibrit model en tutarlı tahminleri (%88 tutarlılık) yapmıştır
- CNN modeli benzer tutarlılık (%81) göstermiştir
- LSTM modeli daha düşük tutarlılık (%83) sergilemiştir

4. Özellik Öğrenme:

- CNN modeli uzamsal özelliklerde daha başarılı olmuştur
- LSTM modeli zamansal ilişkileri daha iyi yakalamıştır
- Hibrit model her iki özellik türünü başarıyla birleştirmiştir

5. Hesaplama Maliyeti:

- CNN modeli en az epoch (57) ile yakınsama göstermiştir
- LSTM modeli en fazla epoch (97) gerektirmiştir
- Hibrit model orta seviyede (95 epoch) eğitim süresi gerektirmiştir

Bu karşılaştırmalı analiz, her modelin kendine özgü güçlü ve zayıf yönleri olduğunu göstermiştir. Hibrit model, diğer iki modelin avantajlarını birleştirerek genel olarak en iyi performansı sergilemiş, ancak bunu biraz daha yüksek hesaplama maliyeti ile gerçekleştirmiştir.



15. SONUÇ VE GELECEK ÇALIŞMALAR

Bu çalışma, Türk İşaret Dili'nin otomatik tanınması için üç farklı derin öğrenme yaklaşımının (CNN, LSTM ve CNN-LSTM hibrit) karşılaştırmalı analizini gerçekleştirmiştir. Her üç model de MediaPipe çerçevesi kullanılarak elde edilen el ve poz landmark'larını (toplam 134 özellik) girdi olarak kullanmış ve 20 farklı Türk İşaret Dili kategorisini sınıflandırmayı başarmıştır. 9188 videodan oluşan veri seti üzerinde yapılan testlerde, CNN-LSTM hibrit modeli %90.19 doğruluk oranı ile en yüksek performansı gösterirken, bunu CNN modeli %83.25 ve LSTM modeli %84.97 doğruluk oranları ile takip etmiştir.

CNN modeli uzamsal özelliklerde, LSTM modeli ise zamansal ilişkilerde daha başarılı olmuş, hibrit model ise her iki özellik türünü başarıyla birleştirmiştir. Araştırmamız sırasında bazı zorluklarla da karşılaştık. Her üç modelde de 'PARA', 'İŞ' gibi benzer hareketlere sahip işaretlerde performans düşüşleri gözlemlenmiştir. Ayrıca, gerçek dünya koşullarında hızlı el hareketleri ve optimal olmayan kamera açıları gibi faktörler modellerin performansını etkilemiştir. Bununla birlikte, özellikle hibrit model, bu zorlukların üstesinden gelmede daha başarılı olmuş ve %88'lik tutarlı tahmin oranı ile en kararlı performansı sergilemiştir.

Gelecekteki çalışmalar için birkaç araştırma yönü önerilmektedir. El ve yüz özelliklerinin geç füzyonu araştırılabilir, bu farklı modalitelerin bilgilerinin daha etkili bir şekilde birleştirilmesini sağlayabilir. Düzleştirilmiş 2D anahtar noktaları yerine poz verilerinin grafik temsilini kullanan modeller incelenebilir, bu işaretlerin uzamsal ilişkilerini daha iyi yakalayabilir. Poz verileri için ilave veri artırma teknikleri geliştirilebilir, bu modelin genelleme yeteneğini artırabilir ve veri kıtlığı sorununu hafifletebilir. Çoklu modalite yaklaşımı da gelecek vaat eden bir araştırma yönüdür. Görüntü verilerine ek olarak, ses ve hareket sensörü verilerinin de modele dahil edilmesi, daha kapsamlı bir işaret dili tanıma sistemi oluşturabilir. Transfer öğrenme teknikleri, özellikle sınırlı veri senaryolarında faydalı olabilir. Derin öğrenme ve istatistiksel yöntemlerin güçlü yanlarını birleştiren hibrit modeller geliştirilebilir, bu her iki yaklaşımın avantajlarından yararlanmayı sağlayabilir. İşaretlerden sözlüksel anlamsal ilişkiler oluşturulması ve

bunların görülen ve görülmeyen işaret sınıfları arasında köprü kurmak için kullanılması da ilginç bir araştırma alanı olabilir. Gerçek zamanlı performans optimizasyonu, modellerin pratik uygulamalarda daha etkin kullanımı için önemlidir. Daha detaylı ve hassas görsel temsiller geliştirilerek, işaretler arasındaki ince farklılıkların daha iyi yakalanması sağlanabilir. Ayrıca, işaret dilinin bütünsel doğasını daha iyi yakalamak için, el hareketlerinin yanı sıra vücut duruşu ve yüz ifadelerinin de modele dahil edilmesi araştırılabilir.

Sonuç olarak, bu çalışma Türk İşaret Dili'nin otomatik tanınması alanında önemli bir adım atmış ve gelecekteki araştırmalar için sağlam bir temel oluşturmuştur. Derin öğrenme yaklaşımının üstün performansı, bu alandaki potansiyelini göstermiştir. Bununla birlikte, istatistiksel yöntemlerin de özellikle veri kısıtlı senaryolarda ve model yorumlanabilirliği açısından değerli olduğu görülmüştür. İşaret dili tanıma, bilgisayarla görme ve doğal dil işlemenin kesişiminde yer alan gelişmekte olan bir araştırma alanıdır. Bu alandaki ilerlemeler, işitme engelli bireylerin iletişimini kolaylaştırma ve toplumsal entegrasyonu artırma potansiyeli taşımaktadır. Gelecekteki çalışmalar, bu teknolojinin daha da geliştirilmesine ve pratik uygulamalarda yaygın olarak kullanılmasına odaklanmalıdır.

KAYNAKLAR

- Ahmed, A. M., Alez, R. A., Taha, M., & Tharwat, G. (2016). Automatic translation of Arabic sign to Arabic text (ATASAT) system. *Journal of Computer Science and Information Technology*, 6, 109-122.
- Akyol, S., & Alvarado, P. (2001, July). Finding relevant image content for mobile sign language recognition. In *IASTED International Conference-Signal Processing, Pattern Recognition and Applications (SPPRA)*, Rhodes (pp. 48-52).
- AlKhuraym, B. Y., Ismail, M. M. B., & Bchir, O. (2022). Arabic sign language recognition using lightweight cnn-based architecture. *International Journal of Advanced Computer Science and Applications*, 13(4).
- Al-Obaidi, A. S. I., Al-Nima, R. R. O., & Han, T. (2023, December). Interpreting arabic sign alphabet by using the deep learning. In *AIP Conference Proceedings* (Vol. 2862, No. 1). AIP Publishing.
- Al-shaheen, A., Çevik, M., & Alqaraghuli, A. (2022). American sign language recognition using yolov4 method. *International Journal of Multidisciplinary Studies and Innovative Technologies*, 6(1), 61-65.
- Al-Shamayleh, A. S., Ahmad, R., Abushariah, M. A., Alam, K. A., & Jomhari, N. (2018). A systematic literature review on vision based gesture recognition techniques. *Multimedia Tools and Applications*, 77, 28121-28184.
- Aly, S., & Aly, W. (2020). DeepArSLR: A novel signer-independent deep learning framework for isolated arabic sign language gestures recognition. *IEEE Access*, 8, 83199-83212.
- Arel, I., Rose, D. C., & Karnowski, T. P. (2010). Research frontier: deep machine learning--a new frontier in artificial intelligence research. *IEEE computational intelligence magazine*, 5(4), 13-18.
- Azis, H., Admojo, F. T., & Susanti, E. (2020). Analisis Perbandingan Performa Metode Klasifikasi pada Dataset Multiclass Citra Busur Panah. *Techno. Com*, 19(3), 286-294.
- Bartoli, A., & Fusiello, A. (Eds.). (2021). *Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part VI* (Vol. 12540). Springer Nature.
- Bazarevsky, V., Grishchenko, I., Raveendran, K., Zhu, T., Zhang, F., & Grundmann, M. (2020). BlazePose: On-device real-time body pose tracking. *arXiv preprint arXiv:2006.10204*.
- Belissen, V., Braffort, A., & Gouiffès, M. (2020). Experimenting the automatic recognition of non-conventionalized units in sign language. *Algorithms*, 13(12), 310.
- Bencherif, M. A., Algabri, M., Mekhtiche, M. A., Faisal, M., Alsulaiman, M., Mathkour, H., ... & Ghaleb, H. (2021). Arabic sign language recognition system using 2D hands and body skeleton data. *IEEE Access*, 9, 59612-59627.

- Benitez-Quiroz, C. F., Gökgöz, K., Wilbur, R. B., & Martinez, A. M. (2014). Discriminant features and temporal structure of nonmanuals in American Sign Language. *PloS one*, 9(2), e86268.
- Bochinski, E., Eiselein, V., & Sikora, T. (2016, August). Training a convolutional neural network for multi-class object detection using solely virtual world data. In *2016 13th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)* (pp. 278-285). IEEE.
- Bochinski, E., Eiselein, V., & Sikora, T. (2016, August). Training a convolutional neural network for multi-class object detection using solely virtual world data. In *2016 13th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)* (pp. 278-285). IEEE.
- Boháček, M., & Hruží, M. (2022). Sign pose-based transformer for word-level sign language recognition. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision* (pp. 182-191).
- Bragg, D., Koller, O., Bellard, M., Berke, L., Boudreault, P., Braffort, A., ... & Ringel Morris, M. (2019, October). Sign language recognition, generation, and translation: An interdisciplinary perspective. In *Proceedings of the 21st International ACM SIGACCESS Conference on Computers and Accessibility* (pp. 16-31).
- Bragg, D., Koller, O., Bellard, M., Berke, L., Boudreault, P., Braffort, A., ... & Ringel Morris, M. (2019, October). Sign language recognition, generation, and translation: An interdisciplinary perspective. In *Proceedings of the 21st International ACM SIGACCESS Conference on Computers and Accessibility* (pp. 16-31).
- Camgoz, N. C., Hadfield, S., Koller, O., Ney, H., & Bowden, R. (2018). Neural sign language translation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7784-7793).
- Camgoz, N. C., Koller, O., Hadfield, S., & Bowden, R. (2020). Multi-channel transformers for multi-articulatory sign language translation. In *Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16* (pp. 301-319). Springer International Publishing.
- Camgoz, N. C., Koller, O., Hadfield, S., & Bowden, R. (2020). Sign language transformers: Joint end-to-end sign language recognition and translation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 10023-10033).
- Camgoz, N. C., Koller, O., Hadfield, S., & Bowden, R. (2020). Sign language transformers: Joint end-to-end sign language recognition and translation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 10023-10033).
- Cao, Z., Simon, T., Wei, S. E., & Sheikh, Y. (2017). Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7291-7299).

- Chai, J., Zeng, H., Li, A., & Ngai, E. W. (2021). Deep learning in computer vision: A critical review of emerging techniques and application scenarios. *Machine Learning with Applications*, 6, 100134.
- Chen, J., Li, K., Bilal, K., Li, K., & Philip, S. Y. (2018). A bi-layered parallel training architecture for large-scale convolutional neural networks. *IEEE transactions on parallel and distributed systems*, 30(5), 965-976.
- Cheok, M. J., Omar, Z., & Jaward, M. H. (2019). A review of hand gesture and sign language recognition techniques. *International Journal of Machine Learning and Cybernetics*, 10, 131-153.
- Cheok, M. J., Omar, Z., & Jaward, M. H. (2019). A review of hand gesture and sign language recognition techniques. *International Journal of Machine Learning and Cybernetics*, 10, 131-153.
- Cheok, M. J., Omar, Z., & Jaward, M. H. (2019). A review of hand gesture and sign language recognition techniques. *International Journal of Machine Learning and Cybernetics*, 10, 131-153.
- Cihan Camgoz, N., Hadfield, S., Koller, O., & Bowden, R. (2017). Subunets: End-to-end hand shape and continuous sign language recognition. In *Proceedings of the IEEE international conference on computer vision* (pp. 3056-3065).
- Cooper, H., Bowden, R., & CVSSP, S. (2007, May). Sign Language Recognition Using Boosted Volumetric Features. In *MVA* (pp. 359-362).
- Cooper, H., Holt, B., & Bowden, R. (2011). Sign language recognition. In *Visual Analysis of Humans: Looking at People* (pp. 539-562). London: Springer London.
- Crasborn, O., & Sloetjes, H. (2008). Enhanced ELAN functionality for sign language corpora. In *6th International Conference on Language Resources and Evaluation (LREC 2008)/3rd Workshop on the Representation and Processing of Sign Languages: Construction and Exploitation of Sign Language Corpora* (pp. 39-43).
- Desai, A., De Meulder, M., Hochgesang, J. A., Kocab, A., & Lu, A. X. (2024). Systemic Biases in Sign Language AI Research: A Deaf-Led Call to Reevaluate Research Agendas. arXiv preprint arXiv:2403.02563.
- Dikyuva, H., Makaroğlu, B., & Arık, E. (2015). Türk işaret dili dilbilgisi kitabı. *Aile ve Sosyal Politikalar Bakanlığı Yayınları: Ankara*.
- Dünya Sağırklar Federasyonu. Çevrimiçi olarak mevcuttur: <http://wfdeaf.org/our-work/> (12 Mayıs 2021 tarihinde erişilmiştir).
- Elakkiya, R. (2021). RETRACTED ARTICLE: Machine learning based sign language recognition: a review and its research frontier. *Journal of Ambient Intelligence and Humanized Computing*, 12(7), 7205-7224.
- Fang, J., Zhou, Y., Yu, Y., & Du, S. (2016). Fine-grained vehicle model recognition using a coarse-to-fine convolutional neural network architecture. *IEEE Transactions on Intelligent Transportation Systems*, 18(7), 1782-1792.

- Farooq, U., Rahim, M. S. M., Sabir, N., Hussain, A., & Abid, A. (2021). Advances in machine translation for sign language: approaches, limitations, and challenges. *Neural Computing and Applications*, 33(21), 14357-14399.
- Farouk, M., Sutherland, A., & Shokry, A. (2013, February). Nonlinearity reduction of manifolds using Gaussian blur for handshape recognition based on multi-dimensional grids. In *International Conference on Pattern Recognition Applications and Methods* (Vol. 2, pp. 303-307). SCITEPRESS.
- Ferreira, P. M., Pernes, D., Rebelo, A., & Cardoso, J. S. (2021). Signer-independent sign language recognition with adversarial neural networks. *International Journal of Machine Learning and Computing*, 11(2), 101.
- Fieres, J., Schemmel, J., & Meier, K. (2006, July). Training convolutional networks of threshold neurons suited for low-power hardware implementation. In *The 2006 IEEE International Joint Conference on Neural Network Proceedings* (pp. 21-28). IEEE.
- Goodfellow, I. (2016). Deep learning.
- Gökçe, Ç., Özdemir, O., Kındıroğlu, A. A., & Akarun, L. (2020). Score-level multi cue fusion for sign language recognition. In *Computer Vision—ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16* (pp. 294-309). Springer International Publishing.
- Gökgöz, K. (2018). Syllables in TİD. *Dilbilim Araştırmaları Dergisi*, 29(1), 29-49.
- Grishchenko, I., & Bazarevsky, V. (2020). Mediapipe holistic—simultaneous face, hand and pose prediction, on device. *Google AI Blog. Dec*.
- Halvardsson, G., Peterson, J., Soto-Valero, C., & Baudry, B. (2021). Interpretation of swedish sign language using convolutional neural networks and transfer learning. *SN Computer Science*, 2(3), 207.
- Halvardsson, G., Peterson, J., Soto-Valero, C., & Baudry, B. (2021). Interpretation of swedish sign language using convolutional neural networks and transfer learning. *SN Computer Science*, 2(3), 207.
- Handayani, A. N., Zaeni, I. A. E., Kurniawan, W. C., & Asmara, R. A. (2020, September). Development of e-collab classroom for cnn practice on electrical engineering, universitas negeri malang. In *2020 4th International Conference on Vocational Education and Training (ICOVET)* (pp. 47-51). IEEE.
- Hawkins, J. A. (2004). Efficiency and complexity in grammars.
- Hibatullah, A. (2019). *Penerapan metode convolutional neural network pada pengenalan pola citra sandi rumput* (Doctoral dissertation, Universitas Komputer Indonesia).
- Hori, N., & Yamamoto, M. (2023). Re-evaluation method by index finger position in the face area using face part position criterion for sign language recognition. *Sensors*, 23(9), 4321.
- Hu, H., Zhao, W., Zhou, W., Wang, Y., & Li, H. (2021). SignBERT: Pre-training of hand-model-aware representation for sign language recognition. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 11087-11096).

- Huang, J., Zhou, W., Zhang, Q., Li, H., & Li, W. (2018, April). Video-based sign language recognition without temporal segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 32, No. 1).
- Indolia, S., Goswami, A. K., Mishra, S. P., & Asopa, P. (2018). Conceptual understanding of convolutional neural network-a deep learning approach. *Procedia computer science*, 132, 679-688.
- Jachova, Z., Kovacheva, O., & Karovska, A. (2008). Differences between American Sign Language (ASL) and British Sign Language (BSL). *Journal of Special Education and Rehabilitation*, 9(1-2), 41-54.
- Jain, A. K. (2010). Data clustering: 50 years beyond K-means. *Pattern recognition letters*, 31(8), 651-666.
- Jaynes, E. T. (2003). *Probability theory: The logic of science*. Cambridge university press.
- Kadous, M. W. (1996, October). Machine recognition of Auslan signs using PowerGloves: Towards large-lexicon recognition of sign language. In *Proceedings of the Workshop on the Integration of Gesture in Language and Speech* (Vol. 165, pp. 165-174). Wilmington: DE.
- Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4, 237-285.
- Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., & Fei-Fei, L. (2014). Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition* (pp. 1725-1732).
- Kausar, S., & Javed, M. Y. (2011, December). A survey on sign language recognition. In *2011 Frontiers of Information Technology* (pp. 95-98). IEEE.
- Keates, S., Clarkson, P. J., & Robinson, P. (2002). *Gestural interfaces for motion-impaired users*. Marcel Dekker Inc..
- Kingma, D. P. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. *Morgan Kaufman Publishing*.
- Laines, D., Gonzalez-Mendoza, M., Ochoa-Ruiz, G., & Bejarano, G. (2023). Isolated Sign Language Recognition based on Tree Structure Skeleton Images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 276-284).
- Laitman, J. T., & Heimbuch, R. C. (1982). The basicranium of Plio-Pleistocene hominids as an indicator of their upper respiratory systems. *American Journal of Physical Anthropology*, 59(3), 323-343.
- Lawrence, S., Giles, C. L., Tsoi, A. C., & Back, A. D. (1997). Face recognition: A convolutional neural-network approach. *IEEE transactions on neural networks*, 8(1), 98-113.

- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, 521(7553), 436-444.
- LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324.
- LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324.
- Lee, K. B., Cheon, S., & Kim, C. O. (2017). A convolutional neural network for fault classification and diagnosis in semiconductor manufacturing processes. *IEEE Transactions on Semiconductor Manufacturing*, 30(2), 135-142.
- Leeson, L. (2006). Moving Heads and Moving Hands: Developing a Digital Corpus of Irish Sign Language. *Ireland*, 25-26.
- Lipton, Z. C. (2018). The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3), 31-57.
- Luqman, H., & El-Alfy, E. S. M. (2021). Towards hybrid multimodal manual and non-manual Arabic sign language recognition: MArSL database and pilot study. *Electronics*, 10(14), 1739.
- Madhushree, M., Raja, P., & Thuja, S. (2023). Certain Investigations in Hand Gesture Recognition-A Survey (No. 9680). EasyChai
- Maggiori, E., Tarabalka, Y., Charpiat, G., & Alliez, P. (2016). Convolutional neural networks for large-scale remote-sensing image classification. *IEEE Transactions on geoscience and remote sensing*, 55(2), 645-657.
- Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., & Frey, B. (2015). Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*.
- Manning, V., Murray, J. J., & Bloxs, A. (2022). Linguistic human rights in the work of the world federation of the deaf. *The Handbook of Linguistic Human Rights*, 267-280.
- Martin, C. T., & Chanda, N. (2016). Mental health clinical simulation: Therapeutic communication. *Clinical Simulation in Nursing*, 12(6), 209-214.
- Miah, A. S. M., Hasan, M. A. M., Jang, S. W., Lee, H. S., & Shin, J. (2023). Multi-stream general and graph-based deep neural networks for skeleton-based sign language recognition. *Electronics*, 12(13), 2841.
- Mocialov, B., Turner, G., & Hastie, H. (2020). Transfer learning for british sign language modelling. *arXiv preprint arXiv:2006.02144*.
- Mocialov, B., Turner, G., & Hastie, H. (2020). Transfer learning for british sign language modelling. *arXiv preprint arXiv:2006.02144*.
- Mustafa, M. (2021). Retracted article: a study on Arabic sign language recognition for differently abled using advanced machine learning classifiers. *Journal of Ambient Intelligence and Humanized Computing*, 12(3), 4101-4115.
- Naranjo-Zeledón, L., Peral, J., Ferrández, A., & Chacón-Rivas, M. (2019). A systematic mapping of translation-enabling technologies for sign languages. *Electronics*, 8(9), 1047.

- Nebauer, C. (1998). Evaluation of convolutional neural networks for visual recognition. *IEEE transactions on neural networks*, 9(4), 685-696.
- Nikolenko, S. I. (2021). *Synthetic data for deep learning* (Vol. 174). Springer Nature.
- Novopoltsev, M., Verkhovtsev, L., Murtazin, R., Milevich, D., & Zemtsova, I. (2023). Fine-tuning of sign language recognition models: a technical report. *arXiv preprint arXiv:2302.07693*.
- O'Shea, K. (2015). An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*.
- Özcan, T. A. Y. Y. İ. P., & Baştürk, A. (2021). ERUSLR: A new Turkish sign language dataset and its recognition using hyperparameter optimization aided convolutional neural network. *Journal of the Faculty of Engineering and Architecture of Gazi University*, 36(1), 527-542.
- Özdemir, O., Baytaş, İ. M., & Akarun, L. (2023). Multi-cue temporal modeling for skeleton-based sign language recognition. *Frontiers in Neuroscience*, 17, 1148191.
- Özdemir, O., Kındıroğlu, A. A., Camgöz, N. C., & Akarun, L. (2020). Bosphorussign22k sign language recognition dataset. *arXiv preprint arXiv:2004.01283*.
- Palsson, F., Sveinsson, J. R., & Ulfarsson, M. O. (2017). Multispectral and hyperspectral image fusion using a 3-D-convolutional neural network. *IEEE Geoscience and Remote Sensing Letters*, 14(5), 639-643.
- Pashaloudi, V. N., & Margaritis, K. G. (2002, April). Hidden markov model for sign language recognition: A review. In *Proc. 2nd Hellenic Conf. AI, SETN-2002* (pp. 11-12).
- Pfister, T., Charles, J., Everingham, M., & Zisserman, A. (2012). Automatic and Efficient Long Term Arm and Hand Tracking for Continuous Sign Language TV Broadcasts. In *BMVC* (pp. 1-11).
- Pisharady, P. K., & Saerbeck, M. (2015). Recent methods and databases in vision-based hand gesture recognition: A review. *Computer Vision and Image Understanding*, 141, 152-165.
- Pishchulin, L., Jain, A., Andriluka, M., Thormählen, T., & Schiele, B. (2012, June). Articulated people detection and pose estimation: Reshaping the future. In *2012 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3178-3185). IEEE.
- Power, J. M., Grimm, G. W., & List, J. M. (2020). Evolutionary dynamics in the dispersal of sign languages. *Royal Society Open Science*, 7(1), 191100.
- Prechelt, L. (1998). Automatic early stopping using cross validation: quantifying the criteria. *Neural networks*, 11(4), 761-767.
- Prechelt, L. (1998). Automatic early stopping using cross validation: quantifying the criteria. *Neural networks*, 11(4), 761-767.
- Ramsey, F. P. (1926). Mathematical logic. *The Mathematical Gazette*, 13(184), 185-194.

- Rastgoo, R., Kiani, K., & Escalera, S. (2021). Sign language recognition: A deep survey. *Expert Systems with Applications*, 164, 113794.
- Rastgoo, R., Kiani, K., & Escalera, S. (2021). Sign language recognition: A deep survey. *Expert Systems with Applications*, 164, 113794.
- Rastgoo, R., Kiani, K., & Escalera, S. (2021). Sign language recognition: A deep survey. *Expert Systems with Applications*, 164, 113794.
- Ryumin, D., Ivanko, D., & Axyonov, A. (2023). Cross-language transfer learning using visual information for automatic sign gesture recognition. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 48, 209-216.
- Ryumin, D., Ivanko, D., & Ryumina, E. (2023). Audio-visual speech and gesture recognition by sensors of mobile devices. *Sensors*, 23(4), 2284.
- Selvaraj, P., Nc, G., Kumar, P., & Khapra, M. (2021). Openhands: Making sign language recognition accessible with pose-based pretrained models across languages. *arXiv preprint arXiv:2110.05877*.
- Ser, G., & Bati, C. T. (2019). Derin sinir ağları ile en iyi modelin belirlenmesi: mantar verileri üzerine keras uygulaması. *Yuzuncu Yıl University Journal of Agricultural Sciences*, 29(3), 406-417.
- Seyyarer, E., Ayata, F., Uçkan, T., & Karci, A. (2020). Derin öğrenmede kullanılan optimizasyon algoritmalarının uygulanması ve kıyaslanması. *Computer Science*, 5(2), 90-98.
- Shalev-Arkushin, R. (2022). Ham2pose: Animating sign language notation into pose sequences (Master's thesis, Reichman University (Israel)).
- Shi, B., Del Rio, A. M., Keane, J., Michaux, J., Brentari, D., Shakhnarovich, G., & Livescu, K. (2018, December). American sign language fingerspelling recognition in the wild. In *2018 IEEE Spoken Language Technology Workshop (SLT)* (pp. 145-152). IEEE.
- Shin, J., Matsuoka, A., Hasan, M. A. M., & Srizon, A. Y. (2021). American sign language alphabet recognition by extracting feature from hand pose estimation. *Sensors*, 21(17), 5856.
- Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., ... & Blake, A. (2011, June). Real-time human pose recognition in parts from single depth images. In *CVPR 2011* (pp. 1297-1304). Ieee.
- Silveira, W., Alaniz, A., Hurtado, M., Da Silva, B. C., & De Bem, R. (2022, October). SynLibras: A Disentangled Deep Generative Model for Brazilian Sign Language Synthesis. In *2022 35th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)* (Vol. 1, pp. 210-215). IEEE.
- Sincan, O. M., & Keles, H. Y. (2020). Autsl: A large scale multi-modal turkish sign language dataset and baseline methods. *IEEE access*, 8, 181340-181355.
- Sincan, O. M., & Keles, H. Y. (2022). Using motion history images with 3d convolutional networks in isolated sign language recognition. *IEEE access*, 10, 18608-18618.

- Smirnov, E. A., Timoshenko, D. M., & Andrianov, S. N. (2014). Comparison of regularization methods for imagenet classification with deep convolutional neural networks. *Aasri Procedia*, 6, 89-94.
- Sreenath, S., Daniels, D. I., Ganesh, A. S., Kuruganti, Y. S., & Chittawadigi, R. G. (2021, September). Monocular tracking of human hand on a smart phone camera using mediapipe and its application in robotics. In *2021 IEEE 9th Region 10 Humanitarian Technology Conference (R10-HTC)* (pp. 1-6). IEEE.
- 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.
- Strubell, E., Ganesh, A., & McCallum, A. (2020, April). Energy and policy considerations for modern deep learning research. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 34, No. 09, pp. 13693-13696).
- Strubell, E., Ganesh, A., & McCallum, A. (2020, April). Energy and policy considerations for modern deep learning research. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 34, No. 09, pp. 13693-13696).
- Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.
- Vogler, C., & Goldenstein, S. (2005, July). Analysis of facial expressions in american sign language. In *Proc. of the 3rd Int. Conf. on Universal Access in Human-Computer Interaction, Springer* (Vol. 1).
- W. H.Organization, "Deafness and hearing loss." Accessed: Oct. 05, 2023.
- Wadhawan, A., & Kumar, P. (2020). Deep learning-based sign language recognition system for static signs. *Neural computing and applications*, 32(12), 7957-7968.
- Wadhawan, A., & Kumar, P. (2021). Sign language recognition systems: A decade systematic literature review. *Archives of computational methods in engineering*, 28, 785-813.
- Wang, C., Gao, W., & Ma, J. (2000, August). An approach to automatically extracting the basic units in chinese sign language recognition. In *WCC 2000-ICSP 2000. 2000 5th International Conference on Signal Processing Proceedings. 16th World Computer Congress 2000* (Vol. 2, pp. 855-858). IEEE.
- Wang, S., Jiang, Y., Hou, X., Cheng, H., & Du, S. (2017). Cerebral micro-bleed detection based on the convolution neural network with rank based average pooling. *IEEE Access*, 5, 16576-16583.
- Wategaonkar, D., Pawar, R., Jadhav, P., Patole, T., Jadhav, R. R., & Gupta, S. (2022). Sign gesture interpreter for better communication between a normal and deaf person. *Journal of Pharmaceutical Negative Results*, 5990-6000.
- Wilson, J. (1985). Deafness in developing countries: approaches to a global program of prevention. *Archives of otolaryngology*, 111(1), 2-9.
- World Health Organization. (2021). *World report on hearing*. World Health Organization.

- Xu, R., & Wunsch, D. (2005). Survey of clustering algorithms. *IEEE Transactions on neural networks*, 16(3), 645-678.
- Young, H. B. (1914). LXI. The Sociologic Aspect of Deafness, Congenital or Acquired in Early Life, with a Suggestion for a Betterment through Indirect Effort. *Annals of Otology, Rhinology & Laryngology*, 23(4), 827-834.
- Zafrulla, Z., Brashear, H., Starner, T., Hamilton, H., & Presti, P. (2011, November). American sign language recognition with the kinect. In *Proceedings of the 13th international conference on multimodal interfaces* (pp. 279-286).
- Zhang, D., Mishra, S., Brynjolfsson, E., Etchemendy, J., Ganguli, D., Grosz, B., ... & Perrault, R. (2021). The AI index 2021 annual report. *arXiv preprint arXiv:2103.06312*.
- Zhang, X., Zou, Y., & Shi, W. (2017, August). Dilated convolution neural network with LeakyReLU for environmental sound classification. In *2017 22nd international conference on digital signal processing (DSP)* (pp. 1-5). Ieee.
- Zhang, X., Zou, Y., & Shi, W. (2017, August). Dilated convolution neural network with LeakyReLU for environmental sound classification. In *2017 22nd international conference on digital signal processing (DSP)* (pp. 1-5). Ieee.
- Zhou, Y., Wang, H., Xu, F., & Jin, Y. Q. (2016). Polarimetric SAR image classification using deep convolutional neural networks. *IEEE Geoscience and Remote Sensing Letters*, 13(12), 1935-1939.