T.C.
KASTAMONU ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
MALZEME BİLİMİ VE MÜHENDİSLİĞİ ANA BİLİM DALI

EVRİŞİMSEL SİNİR AĞLARINI KULLANARAK İŞARET DİLİ GÖRÜNTÜLERİNİN SINIFLANDIRILMASI

KHALED MOHAMED ABUBAKER ELBAYOUDI

YÜKSEK LİSANS TEZİ

DR. ÖĞR. ÜYESİ ÜMİT TOKEŞER

HAZİRAN-2020
KASTAMONU

TEZ ONAYI

KHALED MOHAMED ABUBAKER ELBAYOUDI tarafından hazırlanan "**EVRİŞİMSEL SİNİR AĞLARINI KULLANARAK İŞARET DİLİ GÖRÜNTÜLERİNİN SINIFLANDIRILMASI**" adlı tez çalışması **26.06.2020 tarihinde** aşağıdaki jüri üyeleri önünde savunulmuş ve **oy birliği** ile Kastamonu Üniversitesi Fen Bilimleri Enstitüsü MALZEME BİLİMİ VE MÜHENDİSLİĞİ ANA BİLİM DALI'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Danışman	Dr. Öğr. Üyesi Ümit TOKEŞER Kastamonu Üniversitesi	……..
Jüri Üyesi	Dr. Öğr. Üyesi Cevat RAHEBİ Altınbaş Üniversitesi	……..
Jüri Üyesi	Doç. Dr. Oytun Emre SAKICI Kastamonu Üniversitesi	……..

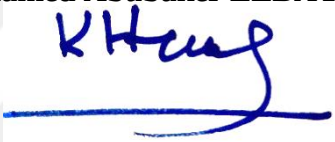
Jüri üyeleri tarafından kabul edilmiş olan bu tez Kastamonu Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulunca onanmıştır.

Enstitü Müdürü	Prof. Dr. İzzet ŞENER	……

TAAHHÜTNAME

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildirir ve taahhüt ederim.

Khaled Mohamed Abubaker ELBAYOUDI



ÖZET

YÜKSEK LİSANS TEZİ

EVRİŞİMSEL SİNİR AĞLARINI KULLANARAK İŞARET DİLİ GÖRÜNTÜLERİNİN SINIFLANDIRILMASI

KHALED MOHAMED ABUBAKER ELBAYOUDI

KASTAMONU ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ
MALZEME BİLİMİ VE MÜHENDİSLİĞİ ANA BİLİM DALI
DANIŞMAN: DR. ÖĞR. ÜYESİ ÜMİT TOKEŞER

Bu tezde, Türk İşaret Dili üzerine bir derin öğrenme (deep learning) uygulaması yapılmaktadır. Dünya genelinde işaret dili tanıma üzerine yapılmış ve halen de devam etmekte olan pek çok çalışma bulunmaktadır. İşaret dili tanıma üzerine yapılan çalışmalar değerlendirildiğinde pek çok problemle karşılaşıldığı görülmüştür. Görüntü işleme açısından değerlendirildiğinde söz konusu problemlerin başlıcaları ortamın aydınlığı ve ışık, arkaplanın karmaşıklığı, kameranın bulunduğu konum ve pozisyonu, oldukça zor olan ve pek çok soruna yol açan uygun alanın çıkarılması işlemi, segmentasyon uygulamalarının sonucu bazı hareketlerin kaybı ve alfabeyi temsil eden bazı parmak hareketlerinin birbirine çok benzemesidir. İşaretlerin kurallara uygun şekilde ifade edilmemesi, peşpeşe gelen işaretler için doğru görüntülerin/karelerin seçilmemesi ve öznitelik çıkarımında doğru yöntemi seçme yeteneğinin eksikliği gibi sebeplerden kaynaklanan yanlış çıkarımlarda bulunulması türünden problemler de mevcuttur. Son zamanlarda yapılan çalışmalara bakıldığında, görüntü işlemeden kaynaklanan problemlerin çözülmesinde Kinect teknolojisinin kullanıldığı görülmektedir. Bu sistemler fazladan donanım gerektirdiği için çok maliyetli olabilmekte ve bu yüzden fazla tercih edilmemektedir. Bu nedenle daha az hatayı netice veren işaret dili tanıma sistemleri geliştirmek için çalışmalar devam etmektedir.

ANAHTAR KELİMELER: Türk işaret dili, yapay sinir ağları, derin öğrenme.

2020, 69 Sayfa
Bilim Kodu:91

ABSTRACT

MSC THESIS

IMAGE CLASSİFİCATİON FOR SİGN LANGUAGE USİNG CONVOLUTİONAL NEURAL NETWORK

KHALED MOHAMED ABUBAKER ELBAYOUDI

KASTAMONU UNIVERSITY INSTITUTE OF SCIENCE
DEPAERTMENT OF MATERİAL SCİENCE AND ENGİNEERİNG
SUPERVISOR: ASSİST. PROF. ÜMİT TOKEŞER

In this thesis, the deep learning method is used to Turkish Sign Languages. Many studies on sign language recognition have been and are still being conducted around the world. When the studies on sign language recognition were evaluated, it was seen that many problems were encountered. From the perspective of image processing, the main problems are the light in the environment, the complexity of the background, the position of the camera, the difficult and troublesome removal of the relevant region, the loss of some signs as a result of segmentation applications and the movements of the finger alphabet are very similar. There are also problems such as incorrect results when the signs are not made in accordance with the rule, difficulty in selecting the right frames for the recognition of successive signs, and inability to choose the correct property extraction method. When we look at the latest studies, it is seen that Kinect technology is used to solve the problems caused by image processing. Such systems are very costly as they require additional equipment and are therefore not preferred much. Therefore, studies are still going on to develop sign language recognition systems with the least error rate.

KEYWORDS: Turkish sign languages, artificial neural network, deep learning.

2020, 69 Pages
Science Code:91

TEŞEKKÜR

Öncelikle bu çalışma boyunca bana yol göstericilik yapan Dr. Ümit Tokeşer'e teşekkür etmek isterim. Bilgisayar Mühendisliği Bölümü hocalarına ve araştırma görevlilerine bu araştırmayla ilgili pek çok gerekliliğin yerine getirilmesi için yaptıkları yardımlardan dolayı minnettarım. Kastamonu Üniversitesi'ndeki arkadaşlarıma ve Libyalı topluluğuna destekleri için teşekkür etmek isterim. Beni bu bursa layık gören Libya Yüksek Öğretim Bakanlığı'na ve Ankara'da bulunan Libya Büyükelçiliği bünyesinde yer alan Kültür İşleri Ateşeliğine de takdir ve teşekkürlerimi ifade etmek isterim.

Son olarak ve son derece önemli olarak; eşim Boshra ve sevgili oğullarım Mohamed ve Ahmed'e, çalışmalarıma devam etmem ve bu tezi bitirmem için gerekli olan özgüveni kazanmamda etkili olan manevi desteği sağladıkları için minnettarım.

Babama, anneme, kız kardeşlerime ve erkek kardeşlerime de aynı şekilde müteşekkirim. Duaları ve inançları bu tezi bitirmem için bana hem cesaret verdi hem de ilham kaynağı oldu. En büyük abim Abubaker'e de, her zaman yanımda olduğu ve başarıyla çalışmalarımı tamamlamamda beni cesaretlendirdiği için teşekkür borçluyum.

Çalışmamın bu alanda çalışan başkaları için kullanışlı olmasını ve ileride yapılacak yeni çalışmalara katkı sağlamasını ümit ediyorum.

KHALED MOHAMED ABUBAKER ELBAYOUDI

Kastamonu, 2020

İÇİNDEKİLER

Sayfa

TEZ ONAYI ... **ii**
TAAHHÜTNAME ... **iii**
ÖZET ... **iv**
ABSTRACT ... **v**
TEŞEKKÜR ... **vi**
İÇİNDEKİLER ... **vii**
SİMGELER VE KISALTMALAR DİZİNİ ... **viii**
ŞEKİLLER DİZİNİ ... **ix**
TABLOLAR DİZİNİ ... **x**
1. GİRİŞ ... **1**
1.1 Arkaplan ... 1
1.2 Motivasyon ... 1
1.3 Tezin Amacı ... 7
1.4 Tezin Yapısı ... 8
2. LİTERATÜR TARAMASI ... **9**
2.1 Arka Plan ... 9
2.2 Evrişimli Sinir Ağı ve İşaret Dilleri ... 9
2.3 Görüntü İşleme Teknikleri ve İşaret Dilleri ... 13
2.4 Sensör Eldivenleri Kullanımıyla İşaret Dillerini Tanıma ... 14
2.5 Bulanık Mantık Yöntemi ve İşaret Dilleri ... 18
2.6 Destek Vektör Makinesi İle İşaret Dilleri ... 25
3. ARAÇ VE YÖNTEMLER ... **29**
3.1 Arkaplan ... 29
3.2 Evrişimli Sinir Ağları ... 29
3.2.1 Evrişimli Sinir Ağlarının İşleyişinin Genel Açıklaması ... 30
3.2.2 Evrişim Adımları ... 31
3.2.3 ReLU ... 40
3.2.4 Birleştirme Adımı ... 42
3.2.5 Tam Bağlantılı Katman ... 44
3.3 Görüntü Sınıflandırma ... 47
3.4 Eğitim ve Test Amaçlı Ayrıştırma Yöntemi ... 47
4. SİMÜLASYON SONUÇLARI ... **48**
4.1 Deneysel Sonuçlar ... 48
4.2 Türk İşaret Dili İçin Veri Kümesi ... 48
4.3 İşin Akış Şeması ... 49
4.4 Python Açıklaması ... 51
4.4.1 Python Geliştirme Ortamları ... 51
4.4.2 NumPy, SciPy ve Matplotlib Kütüphaneleri ... 52
4.5 Simülasyon Sonuçları ... 52
4.6. Diğer Çalışmalarla Karşılaştırma ... 54
5. SONUÇ VE TARTIŞMA ... **60**
5.1 Sonuç ... 60
5.2 Gelecek Çalışmalar ... 61
KAYNAKLAR ... **62**
ÖZGEÇMİŞ ... **69**

SİMGELER VE KISALTMALAR DİZİNİ

Kısaltmalar

TSL : Türk İşaret Dili
ASL : Amerikan İşaret Dili
GSL : Alman İşaret Dili
CSL : Çin İşaret dili
CNN : Evrişimli Sinir Ağı
Auslan : Avustralyan İşaret dili
ArSL : Arapça İşaret dili
CTC : Bağlantıcı Geçici Sınıflandırma
MKNN : Modifiye k. En Yakın Komşu
SL : İşaret Dilleri
SVD : Tekil Değer Ayrışması
SVM : Destek Vektör Makinesi
SASL : Güney Afrikan İşaret dili
ACC : Accuracy

ŞEKİLLER DİZİNİ

Sayfa

Şekil 1.1 Türk İşaret Dili için Parmak Alfabesi 1
Şekil 2.1 Amerikan işaret dili rakamları 10
Şekil 2.2 Amerikan işaret dili alfabesi 10
Şekil 2.3 Aşama eğitimi yaklaşımına genel bakış 12
Şekil 2.4. Tamil İşaret Dili Alfabesi için Önerilen Metodoloji 14
Şekil 2.5 İşaret dilini konuşulan dile tam olarak çeviren bir uygulama modeli 15
Şekil 2.6. Veri toplama ortamı ve bölümlere ayrılmış örnek cümle 17
Şekil 2.7 IT2FHMM model eğitiminde kullanılan görüntü örnekleri 19
Şekil 2.8 PSL tanıyıcı sisteminin blok şeması 19
Şekil 2.9 PSL'de görüntü girdisi 20
Şekil 2.10 Girdi görüntüsüne Akıllı Kenar Algılama işleminin uygulanması süreci 25
Şekil 3.1 Genel sistem adımları 29
Şekil 3.2 Evrişimli Sinir Ağlarının Genel Modeli (Bantupalli ve Xie, 2019, Matsugu, 2003) 30
Şekil 3.3 ConvNet'in TSL'de yer alan (A) harfini tanıması süreci (Saha., 2018) 31
Şekil 3.4 Evrişim katmanında öznitelik haritasının elde edilişi (Bantupalli ve Xie, 2019) 32
Şekil 3.5 5×5×1'lik görüntünün 3×3×1'lik çekirdekle evriştirilip 3×3×1'lik özniteliğin elde edilmesi (1) (Bantupalli ve Xie, 2019) 33
Şekil 3.6 5×5×1'lik görüntü matrisiyle 3×3×1'lik çekirdeğin evrişim işlemine sokulması ve 3×3×1'lik özniteliğin elde edilmesi (2) (Bantupalli ve Xie, 2019) 35
Şekil 3.7 CNN'de öznitelik haritası 36
Şekil 3.8 Adım uzunluğu 1 (Albawi vd., 2017) 37
Şekil 3.9 Adım uzunluğu 2 (Albawi vd., 2017) 38
Şekil 3.10 Adım uzunluğu 2, sıfır dolgusu 1'ken evrişim (Albawi vd., 2017) 39
Şekil 3.11 Sıfır dolgusu (Albawi vd., 2017) 40
Şekil 3.12 ReLU aktivasyon fonksiyonu (Bantupalli ve Xie, 2019) 41
Şekil 3.13 ReLU İşlemi (Bantupalli ve Xie, 2019) 41
Şekil 3.14 CNN'de maksimum havuzlama işlemi 43
Şekil 3.15 CNN'de havuzlama ortalaması 44
Şekil 3.16 Düzleştirme 45
Şekil 3.17 Tam bağlantılı katman (Saha., 2018) 46
Şekil 4.1 Önerilen metodun ana adımları 48
Şekil 4.2 Türk İşaret Dilinden örnekler (Ezel ve Baykan, 2017) 49
Şekil 4.3 Önerilen yöntemin özeti 50
Şekil 4.4 CNN katmanının eğitim süreci 53
Şekil 4.5 Her bir harf için doğruluk performansı 53
Şekil 4.6 Python tarafından okunan bazı görüntüler 54
Şekil 4.7 Çalışmamızda kullanılan Python geliştirme ortamı 54

TABLOLAR DİZİNİ

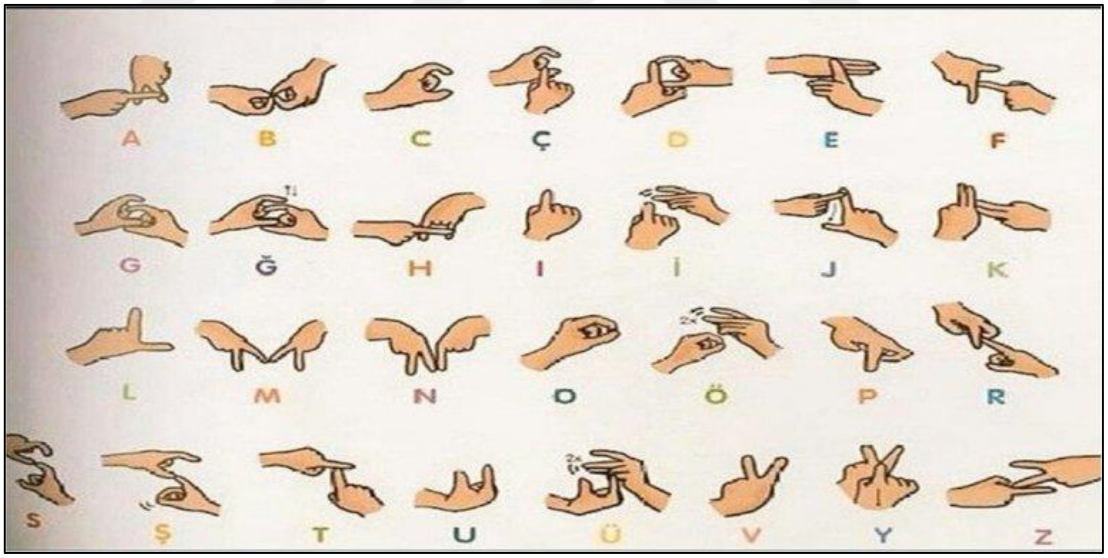
Sayfa

Tablo 1.1 Dünyadaki başlıca işaret dilleri (Sahoo vd., 2014) 3
Tablo 1.2 İşaret dili/Yüz ifadesi için standart veritabanları (Sahoo vd., 2014) 4
Tablo 1.3 Farklı araştırmacılar tarafından işaret dili tanımak için oluşturulan veritabanları 5
Tablo 2.1 Çeşitli araştırmacılar tarafından kullanılan özellik çıkarma yöntemleri 21
Table 2.2. Özel olarak tasarlanmış aygıtlar kullanarak özellik çıkarma 22
Tablo 2.3 Video kamera kullanarak öznitelik çıkarma 23
Tablo 2.4 Araştırmacılar tarafından kullanılan sınıflandırma yöntemleri 27
Tablo 4.1 Sonuçların Karşılaştırılması 55
Tablo 4.2 Araştırmacıların kendi veri kümelerinden alınan sonuçlar 57

1. GİRİŞ

1.1 Arkaplan

Konuşulan dil insanlar arası iletişimi sağlayan başlıca yöntemdir; bu şekilde iletişim kurduğumuzda sadece konuşmanın gereklerini yerine getirmiş olmayız, bu aynı zamanda sosyal bağların tesis edilmesine de yarar. Konuşmayı engelleyen fiziksel anomalilerin neredeyse tamamı sözlü olarak iletişim kurmayı imkansız hale getirmektedir. Bu sorunun üstesinden gelebilmek için, işaret dili olarak adlandırılan görsel bir iletişim sistemi geliştirilmiştir, bu sistemde konuşma dilindeki tabirler vücut hareketleriyle (çoğunlukla eller) ifade edilir. Türk İşaret Dili (TSL), Türk alfabesindeki harflerin el hareketleriyle temsil edildiği standart işaret dillerinden biridir. Şekil 1.1'de Türk İşaret Dilindeki harflerin el parmaklarıyla nasıl ifade edildiği görülmektedir.



Şekil 1.1 Türk İşaret Dili için Parmak Alfabesi

1.2 Motivasyon

İşaret dili, işitme ve konuşma bozukluğu olan kişilerin kullandığı bir iletişim şeklidir ve şu an işitme bozukluğu, birçok insanın dünya çapında yaşamlarının kısıtlanmasını netice vermekte, bu insanların benzeri kısıtlamaları olmayanlara göre kolay olmayan bir hayat sürmelerine neden olmaktadır (Garg, 2012). Sağır ve dilsiz insanlar

birbirleriyle işaret dili denen bir dille iletişim kurarlar; bu dil el hareketlerinden, pozisyonlardan, jestlerden, yüz ifadesiden, ağız ve kafa hareketlerinden oluşur. İşaret dilleri günlük yaşamda farklı ifade araçlarını kullanan doğal dillerdir. İşaretler bir veya iki elle ifade edilir. Şu an itibariyle gelişmiş olan işaret dili konuşma dilinden farklıdır, dolayısıyla işaret dili dilbilgisi temel olarak konuşma dilinkinden farklıdır. Konuşma dilinde cümle yapısı tek boyutludur; bir kelime bir diğerini takip ederken, işaret dilinde ise zamansal ve mekansal bir konfigürasyonda eşzamanlı bir yapı paralel olarak mevcuttur. Çok az insan işaret dilini bilmektedir. Dahası, işaret dili yaygın kanaatin aksine uluslararası bir dil değildir (Pigou vd., 2014). Aslında, bugün dünya çapında 138 ila 300 arasında farklı işaret dili kullanılmaktadır. İşitme engelli olan çocuklar ve yetişkinler arasında sıklıkla yeni işaret dilleri gelişmektedir. Örneğin Amerikan İşaret Dili (ASL), Amerika Birleşik Devletleri'nin neredeyse tüm işitme engelli topluluklarınca benimsenen işaret diliyken, coğrafi konuma bağlı olarak GSL (Alman İşaret Dili), CSL (Çin İşaret Dili), TSL (Türk İşaret Dili), Auslan (Avustralya İşaret Dili), ArSL (Arapça İşaret Dili) gibi farklı işaret dilleri gelişmiş durumdadır (Garg, 2012). İşaret dili, teknoloji ve çok modlu iletişim alanında en iyi anlaşılan alanlardan biridir. Derin öğrenme alanındaki güncel gelişmeler sayesinde sinir ağlarının işaret dili çevirisi yapabileceği kapsamlı uygulamalar bulunmaktadır. Düşüncelerimizi paylaşmanın veya mesajlarımızı başkalarına iletmenin tek yolu iletişimdir, ancak konuşamayan (sağır ve dilsiz) biri için normal insanlarla iletişim zordur (Bheda ve Radpour, 2017). Bu zorluğun üstesinden gelmek, anlık olarak işaret dilinin herkesin anlayabileceği şekilde çevirilmesiyle ve iletişimin sağlanmasıyla mümkündür, bu iletişim ve iletişim için gerekli olan çeviri de makine öğrenimi ve görüntü işleme tekniklerini kullanarak icra edilebilir. Söz konusu çeviri işleminde, Evrişimli Sinir Ağı (CNN) görüntü sınıflandırması kullanılacaktır. Otomatik görüntü sınıflandırma sistemlerinde, evrişimli sinir ağları yaygın olarak kullanılmaktadır (Yim vd., 2015).

Çalışmamız için bu konunun seçilmesini netice veren bazı zorluklar ve problemler şu şekilde sıralanabilir:

- Dünya üzerinde çok fazla sayıda sağır ve/veya dilsiz insan konuşma ve/veya işitme problemlerinden muzdariptir.

- Bütün sosyal boyutlarda iletişim temel bir gerekliliktir.
- İşaret dilleriyle kurulan iletişimde aşılması gereken zorluklar bulunmaktadır.
- Konuşma engelli insanlarla kurulan dialoglarda sıklıkla yanlış anlaşılmalar söz konusu olmaktadır.
- Sağır ve dilsiz insanlar toplumdan izole olarak yaşamaktadır.
- Konuşma engelli insanlar günlük rutinlerini tamamlamakta zorluk çekmektedir.
- Çocuğunda konuşma veya işitme engeli olan ebeveynler çocuklarını yanlış anlamaktadır.
- Sağır ve dilsiz insanların bir iletişim çözümüne ne derece ihtiyaç duyduğunu diğer insanlar anlayamamaktadır.
- Büyük bir kitlenin günlük hayata olan ilgisinin artması bu yolla mümkün olabilir.
- El hareketlerini ve kullanılan işaretleri tercüme edecek bir yazılım yukarıda sayılan sorunları çözmede etkili olacaktır.

Dünya üzerindeki başlıca işaret dilleri Tablo 1.1'de listelenmektedir.

Tablo 1.1 Dünyadaki başlıca işaret dilleri (Sahoo vd., 2014)

Sıra No	Ülke/Coğrafi Bölge	İşaret Dili	Kısaltması	Makale Sayısı
1	Birleşik Krallık	İngiliz İşaret Dili	BSL	NIL
2	Amerika Birleşik Devletleri	Amerikan İşaret Dili	ASL	13
3	Avustralya	Avustralya İşaret Dili	Auslan	NIL
4	Japonya	Japon İşaret Dili	JSL	NIL/1
5	Çin Halk Cumhuriyeti	Çin İşaret Dili	CSL	5
	Tayvan	Tayvan İşaret Dili	TSL	1
6	Orta Doğu	Arapça İşaret Dili	ArSL	2
	İran İslam Cumhuriyeti ve diğer Körfez ülkeleri	Farsça İşaret Dili	PSL	2
7	Hindistan Cumhuriyeti	Hint İşaret Dili	ISL	NIL/2
8	Vietnam Sosyalist Cumhuriyeti	Vietnam İşaret Dili	VSL	1
9	Ukrayna	Ukrayna İşaret Dili	UKL	1
10	Sri Lanka Demokratik Sosyalist Cumhuriyeti	Sri Lanka İşaret Dili	SLTSL	1

Tablo 1.1 Devamı

11	Brezilya Federatif Cumhuriyeti	Brezilya İşaret Dili (Lingua Brasileira de Sinais)	Libras	1
12	Polonya Cumhuriyeti (Rzeczpospolita Polska)	Polonya İşaret Dili (Polski Jezyk Migowy)	PJM	1
13	Hollanda (Nederland)	Nederlandse Gebarentaal veya Hollanda İşaret Dili	NGT/ SLN	1

Farklı araştırmacılar tarafından kullanılan standart veritabanları farklı kütüphaneler halinde mevcuttur. Bu çalışmada Tablo 1.2'de listelenen kütüphane verileri kullanılacaktır.

Tablo 1.2 İşaret dili/Yüz ifadesi için standart veritabanları (Sahoo vd., 2014)

Kütüphane Veritabanı	**İşaret dili**
Lifeprint Fingerpell Library	ASL
CAS-PEAL	CSL
Extended Yale B Yale B frontal (Extended Yale B'nin alt kümesi)	Face Database
Weizmann face database	Face Database
Amerikan İşaret dili Çevriyazı (transcription) için SignStream'in kullanıldığı Linguistik Araştırma Projesi	ASL
ASL Lexicon Video Dataset	ASL
eNTERFACE	ASL
PETS 2002	PSL'ye benzer
RWTH-BOSTON-104 Database	ASL

Tablo 1.3'te farklı araştırmacılar tarafından oluşturulan veri tabanlarının karakteristik özellikleri verilmektedir.

Tablo 1.3 Farklı araştırmacılar tarafından işaret dili tanımak için oluşturulan veritabanları

İşaret Dili	Açıklama	Örnek küme	Kaynak
ASL	ALS alfabesi, ASL'de kullanılan tek haneli sayılar ve sadece elle ifade edilen kelimelerin bir alt grubu.	örn. A, B, D… örn. 3, 5, 7 örn. sevgi, karşılaşmak, fazla.	(Munib vd., 2007)
CSL	Çin manuel alfabesi, her biri bir Pinyin harfine tekabül eden 30 el hareketi.	A-Z, ZH, CH, SH, NG	(Quan ve Jinye, 2008)
Belirtilmemiş	Deneysel veri, 183 kelime içeren 25 cümlenin işaret dilinden oluşmaktadır.	Örnek küme belirtilmemiş	(Maebatake vd., 2008)
Belirtilmemiş	Veritabanının kelime hazinesi, isim-fiil-sıfat gibi on farklı kelime türünden kelimeleri temsil eden 262 işaretten oluşur.	Örnek küme belirtilmemiş	(Grobel ve Assan, 1997)
VSL	Latin kökenli Vietnam işaret dilinin alfabesi 23 harften oluşur.	A, B, C, D, Đ, E, G, H, I, K, L, M, N, O, P, Q, R, S, T, U, V, X, Y.	(Bui ve Nguyen, 2007)
ASL	26 harfi ifade etmek için 26 el hareketi.	A, B, C, …, Z	(Wang vd., 2010)
UKL	UKL'nin 85 hareketi, el işaretlerinin örnek görüntüleri.	Neden? Ödül, Ne için?, Ne kadar?, vb.	(Davydov vd., 2010)
ASL	Statik alfabe işaretleri ve ikinci veritabanı Amerikan İşaret dilinin dinamik alfabe dizilerinden oluşur.	A, B, C, …, Z örn. JOHN (J-O-H-N) ismi	(Rokade vd., 2009)
ASL	Eğitim kümesi 26 harften oluşur.	A, B, C, …, Z	(Mekala vd., 2011)
TSL	TSL'den 15 farklı jest.	Bir madeni para, sigara, çiçek, isteksizce, sıra, almak, hemen, anlamak, nefret, sol, yedi, ay, sekiz, yürümek, vicdan.	(Pahlevanzadeh vd., 2007)

Tablo 1.3 devamı

ArSL	İşaretlerle ifade edilen 23 Arapça kelime/deyimden oluşan bir veritabanı.	Arkadaş, Bir şey yemek, Komşu, Uyumak, Misafir, İçmek, Hediye, Uyandırmak, Düşman, Dinlemek, Selam olsun, Konuşmayı bırakmak, Hoşgeldin, Koklamak, Teşekkür ederim, Yardım etmek, İçeri gel, Dün, Utanç, Gitmek, Ev, Gelmek ve Ben.	(Shanableh ve Assaleh, 2007)
ArSL	30 farklı el hareketini temsil etmek için 900 adet renkli görüntü kullanılmıştır.	elif , ted, be, zad, te, ayn, ta, gayn, cim, fe, he, qaf, kha, kef, dal, lam, zel, mim, re, nun, ze, he, sin, vav, şin, le, sad, ye, dat, te.	(Maraqa vd., 2012)
ASL	26 KARAKTER	A, B, C, …, Z	(Nadgeri vd., 2010)
SLTSL	Sri Lanka Tamil parmak yazım sembolleri.	On iki ünlü, on sekiz ünsüz, dört Grantha ünsüzü ve bir özel karakter ‘bu nedenle’.	(Vanjikumaran ve Balachandran, 2011)
PSL	PSL'den sekiz işaret.	İyi, kase, izin, çocuk, tarihi, dur, cümle, elektrik	(Sarkaleh vd., 2009)
ASL	Gerçek zamanlı İşaret Dili Hareket Tanıma Sisteminde (SemiosiS) ASL hareketleri	İyi geceler, tünaydın, günaydın, eş (hanım), kız çocuk, kız kardeş, artış, Tanrı, İsa, doğum günü, temel, dün.	(Kumarage vd., 2011)
Libras	Brezilya İşaretleri (kelimeler).	Ağaç, ananas, dün, dünden önceki gün, vb	(Antunes vd., 2011)
PSL	PSL alfabesinden 32 seçili harf/hareket.	B, OO, …, Y, H.	(Karami vd., 2011)
ASL	36 hareketin 30 farklı görüntüsü.	A-Z, 0-9	(Binh vd., 2005)

Tablo 1.3 devamı

PJM	PJM'den 48 el duruşu.	22 pozisyon şu karakterlere karşılık gelir: A, B, C,…, Z. 5 pozisyon tam sayılara karşılık gelir: 1, 2, 4, 5, 100. 7 pozisyon Uluslararası İşaret dili pozisyonlarına karşılık gelir: Bm, Cm, ..., Xm 10 pozisyon PSL'nin modifikasyonlarıdır: Aw, Bk, Bz,., Yk 4 pozisyon, sayı pozisyonlarının değiştirilmiş haline karşılık gelir: 1z, 4z, 5s, 5z.	(Flasiński ve Myśliński, 2010)
CSL	CSL alfabesinden 30 harf	A, B, C, …, Z. ZH, CH, SH, NG.	(Teng vd., 2005)
ASL	Amerikan İşaret Diline (ASL) dayanır ve İngilizce'nin net ve eksiksiz bir görsel sunumunu elde etmek için kelimelerle, kiplerle, son ekler ve ön ekler ile zenginleştirilmiştir.	Sahne 1 (her iki parmak da bükülür): A, C, D, E, G, I, J, L, M, N, O, Q, S, T, X, Y, Z, küçük-C, I-L-HAND, BENT-V, L-1 HAND; Sahne 2 (yüzük parmağı bükülüken orta parmak düzdür): H, K, P, R, U, V; Sahne 3 (her iki parmak da düz): B, F, W, BENT, FLAT.	(Kong ve Ranganath, 2008)

1.3 Tezin Amacı

Bu tezin amacı işitme engelli insanlarla normal insanlar arasındaki iletişime imkan verecek teknolojinin geliştirilmesidir. Burada sistemin el hareketlerini ve işaretlerini içeren görüntüleri herkesin anlayabileceği dile çevirmesi beklenmektedir. Bu doğrultuda sistem, Türk İşaret Dili alfabesindeki harfleri otomatik olarak tanımak durumundadır. İzlenecek yöntem Türk İşaret Dili alfabesini tanımak için görüntüleri sınıflandırmak ve doğruluğu yüksek sonuçlar elde etmek için Evrişimli Sinir Ağlarından (Convolutional Neural Network-CNN) yararlanacaktır.

1.4 Tezin Yapısı

- Birinci bölümde, bu çalışmanın amacı ve bunun başlıca sebepleri açıklanmaktadır.
- İkinci bölümde işaret dillerinin tanınması üzerine yapılan farklı alanlardaki çalışmalar ve bu alanda uygulanan farklı yöntemler tartışılmaktadır.
- Üçüncü bölüm bu tezde kullanılan yöntemleri açıklamaktadır. Ayrıca, evrişimli sinir ağları, görüntü işleme, ve veri kümesinin eğitim ve test amaçlı olarak nasıl sınıflandırılacağına dair açıklamalara bu bölümde yer verilecektir.
- Dördüncü bölüm Türk İşaret Dilini içeren veri kümesini sunmaktadır. Bu bölüm ayrıca iş akış şemasını ve Python programlama diline dair açıklamaları içermektedir. Son olarak, simülasyondan elde edilen sonuçların hızları ve doğruluk yüzdeleri sunulmaktadır.
- Beşinci bölüm elde edilen sonuçları sunar ve gelecekte yapılabilecek olası çalışmalara değinir.

2. LİTERATÜR TARAMASI

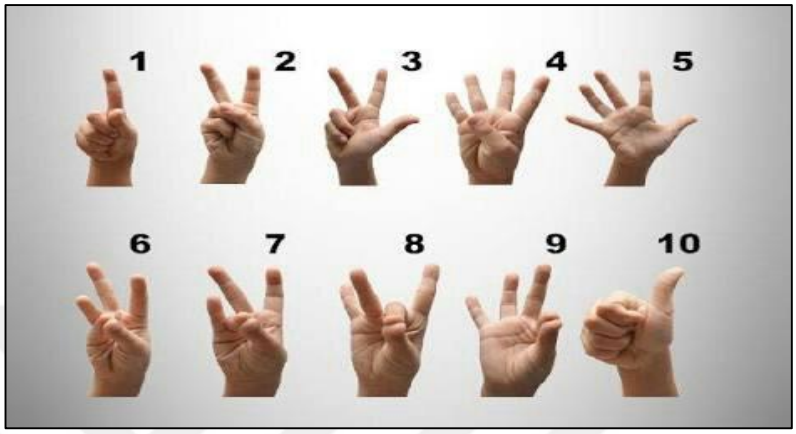
2.1 Arka Plan

İşaret dilini yorumlamak ya da tercüme etmekle kasdedilen ellerin veya yüz hareketlerinin, jestlerin ayırt edilmesi, bu hareketlerin anlamının da herkes için anlaşılabilir olacak şekilde ortaya konulmasıdır. Pek çok araştırmacı bu gereksinime farklı çözümler geliştirmeye çalıştı. İşaret dili tanıma genellikle üç ana aşamaya ayrılır. İlk olarak, el hareketleri veya vücut parçalarının tespiti ve kaydedilmesi gerekir, bu da kısaca veri toplama ve izleme anlamına gelir. İkinci adımda, el hareketlerinin veya jestlerin bilgilerini algılayan fonksiyonlar tanımlanmalı ve girdi verilerinden bunun çıkarımı yapılmalıdır. İcra edilen işaret dili anlamı, son adımda, özniteliğin tanımlayıcılarından toplanan veriler kullanılıp sınıflandırılarak elde edilir (Garg, 2012). Bazı çalışmalara göre, veri toplama yöntemleri temel olarak vizyona/görmeye veya eldiven kullanımına dayanan bazı yaklaşımlardan oluşmaktadır; vizyona/görmeye dayalı yöntemlerde kamera işaretleri yani hareketleri/jestleri yakalar, görüntüleri tanır ve doğru şekilde sınıflandırır (Raheja vd., 2016).

2.2 Evrişimli Sinir Ağı ve İşaret Dilleri

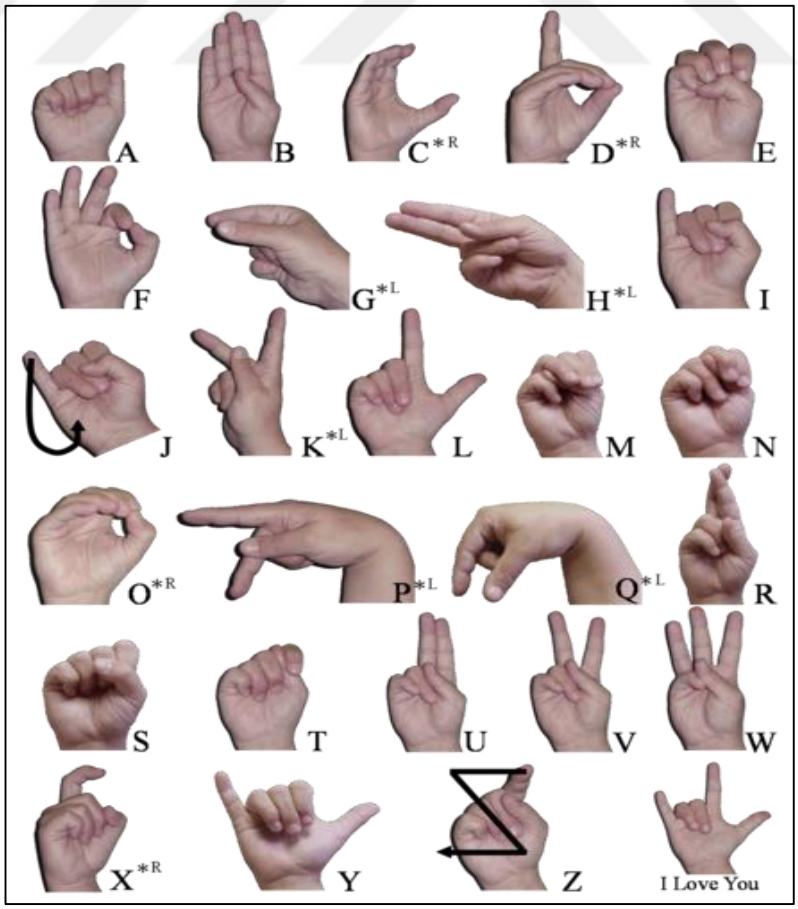
Bheda ve Radpour (2017) tarafından yapılan araştırmalarda Amerikan işaret dilinde hem harflerin hem de rakamların resimlerini sınıflandırmak için yoğun bir şekilde evrişimsel sistemleri kullanan bir teknik önerdiler. Evrişimli Sinir Ağları, görüntü tanıma ve gruplama konularında inanılmaz derecede verimli olmuş ve son zamanlarda insan işaretlerinin tanınması için etkili bir şekilde kullanılmıştır. Genel stratejileri, mini-grup stokastik gradyan azalmasını basit denetimli öğrenme tekniklerinden biri olarak kullanmaktı. Amaçları ASL'deki her harf ve rakamı (0-9) derin evrişimli sinir ağları kullanarak tanımlamaktı. Görüntüler, yüksek çözünürlükte sabit boyutlu 200x200 veya 400x400, 200x200'e kadar dolgulanmış durumdaydı. Veri kümesi, her işaret için alınan görüntülerden oluşuyordu. Uygun teknikler kullanılarak, arka planlar görüntülerden çıkarıldı. Veri kümesi eğitim ve test için ikiye bölündüğünde doğruluk oranı yüksekti. Derinliği ve şekli algılayan kameraların

kullanılmasıyla, işaretleme hareketi ile her iletişim için derinlik ve hareket profilleri oluşturarak prosedür çok daha basit hale getirilmektedir (Bheda ve Radpour, 2017) (Hisham ve Hamouda, 2018). Amerikan işaret dili rakamlarının örnekleri şekil 2.1'de gösterilmektedir.



Şekil 2.1 Amerikan işaret dili rakamları

Amerikan işaret dili alfabe örnekleri şekil 2.2'de gösterilmiştir.



Şekil 2.2 Amerikan işaret dili alfabesi

Başlangıçta, otomatik olarak oluşturulan bir dizi görüntüyü eğitip test ettiler. Bu bilgilendirme koleksiyonu, 5 kişi ve rakamdan oluşan her harf için 25 görüntüden oluşan bir derlemeydi. Veri kümesi koleksiyonu kontrollü bir ortamda çalışmadığından, özellikle ışık, ten gölgeleme ve görüntülerin yakalandığı diğer ekolojik kontrastlara duyarlıdır. Araştırmacılar, alfabenin hareketlerinde yüzde 82,5 doğruluk ve NZ ASL veri kümesi kullanılırken rakamlarda yüzde 97 doğruluk tespit ettiler. Alfabenin harfleri yüzde 67, rakamlar yüzde 70 doğruydu. Harf hareketlerinin eğitimi, geçen süre anlamında yaklaşık 25 dakikaya yaklaştı, ve rakamlar içinse 10 dakika içinde aynı sonuca ulaşılmıştı. Araştırmacılar, alfabe hareketlerinde% 82.5 doğruluk ve NZ ASL veri kümesini kullanırken rakamlarda %97 doğruluk doğruluğunu gözlemlediler. Kendi oluşturdukları veri kümesinde çok daha düşük doğruluk ölçümleri vardı. Doğruluk, alfabedeki harflerde %67 ve rakamlarda %70 seviyesindeydi. Zaman karmaşıklığı açısından, harflerin hareketleri yaklaşık 25 dakikada, rakamlar ise yaklaşık 10 dakikada çözümlenmiştir (Bheda ve Radpour, 2017).

Zhang vd. (2017) tarafından derin sinir ağları ile işaret dilinin görüş tabanlı sürekli tanınması için zayıf denetimli bir sistem sundular; bu sistemde organize açıklama işaretleri vardı ancak işaret cümleleri görüntülerinde zaman ayrımı yoktu, ve etiketli eğitim cümlelerinin sayısı azaltılmıştı. Metodolojileri, uzamsal ve zamansal özelliklerin tespiti ve sekans eğitimi için tekrarlayan evrişimli bir sinir ağı dahil ederek video parçalarının açıklamalarla eşleştirmesini incelemektir. Tasarım mimarisi için üç aşamalı bir optimizasyon süreci tasarladılar. Daha sonra, sekans öğrenme için uçtan uca bir öğrenme şeması geliştirdiler ve uyum önerisi için objektif mekanizma olarak bağlantıcı zamansal sınıflandırmayı (CTC) kullandılar. Bundan sonra, çıkarıcı fonksiyonunu değiştirmek için daha güçlü kontrol olarak uyumlaştırma önerisini aldılar. Son olarak, gelişmiş öznitelik temsilleri ile, sekans öğrenme modeline ince ayar yaparak zayıf denetimli bir düzenleme algılama ağı oluşturdular.

İşaret dilinin anlık anlaşılması için önerilen çözümü gerçek dünyadaki bir duruma uyguladılar ve geliştirdikleri yöntem herhangi bir ek denetim olmaksızın en son teknolojiye eşdeğer sonuçlar üretti. Vizyona dayalı anlık işaret dili tanıma sistemleri

genellikle işaretçilerin başarısının görüntü dizilerini girdi olarak alır ve açıklama etiketlerini otomatik olarak doğru sırayla çıkarmayı öğrenir. Önerilen çözüm, özellik dizilerinin açıklamalı dizilere nasıl eşleneceğini öğrenmek için CNN'yi geçici evrişim ve video kliplerden uzamsal-zamansal temsil öğrenimi ve uzun süreli bellek (LSTM) modülüyle RNN kullanmaktır. Uygulama aşamasında, derin mimariyi etkili bir şekilde eğitmek için üç aşamalı bir optimizasyon süreci vardır.

1) Tüm mimariyi çıkarım fonksiyonu ve öğrenme dizisi bölümü ile eğiterek uyumlaştırma çözümünü bulmak.
2) Öznitelik çıkarımı bölümünü ayarlamak için açıklama düzeyi segmentleri ile uyumlaştırma önerisinin kategorik olasılıkları arasındaki yazışmayı kullanmak.
3) Gelişmiş fonksiyon temsilleri ile öğrenmeye çalışan dizi elemanını girdi olarak birbirine bağlamak.

Eğitim yaklaşımının aşamaları şekil 2.3'te gösterilmektedir.



Şekil 2.3 Aşama eğitimi yaklaşımına genel bakış

İşaret dilinin anlık anlaşılması için veri kümesinin bir grubu olan RWTH-PHOENIX-Weather çoklu işaretçisi 2014 (Forster vd., 2014) üzerinde yöntemlerini değerlendirdiler. Bu veri kümesi Alman işaret dilinde, eğitim için 5.672 cümle, toplam 65.227 işaret açıklaması, ve 799.006 görüntü içerir. Bu tür videolar dokuz işaretçi tarafından yapılır ve her video tek bir açıklama cümlesi içerir. Sonuçları nicel olarak ölçmek için, işaret dilinin anlık tanınmasında yaygın olarak kullanılan metrik olarak kelime hata oranı (WER) kullanılır. Referans diziyi hipoteze dönüştürmek

için, WER en az sayıdaki değişiklik, silme (del) ve yerleştirme (ins) işlemlerini test eder. Uçtan uca tanıma testinin sonuçları, RWTH-PHOENIX-Weather 2014'ün çoklu işaretçi veri kümesinde eğitim kümesi için del=45.6, ins=2.5 ve C3d + BLSTM için WER=76.8; test verilerinde del=46.9, ins=2.8, WER=77.6; ConvTC + RNN eğitim verisindeyken del=19.5, ins=6.8, WER=53.8; test verilerinde del=18.9, ins=7.6, WER=53.7, ancak +öneğitim doğrulama verileri del=16.3, ins=6.7, WER=46.2; test verilerinde del=15.1, ins=7.4, WER=46.9 idi.

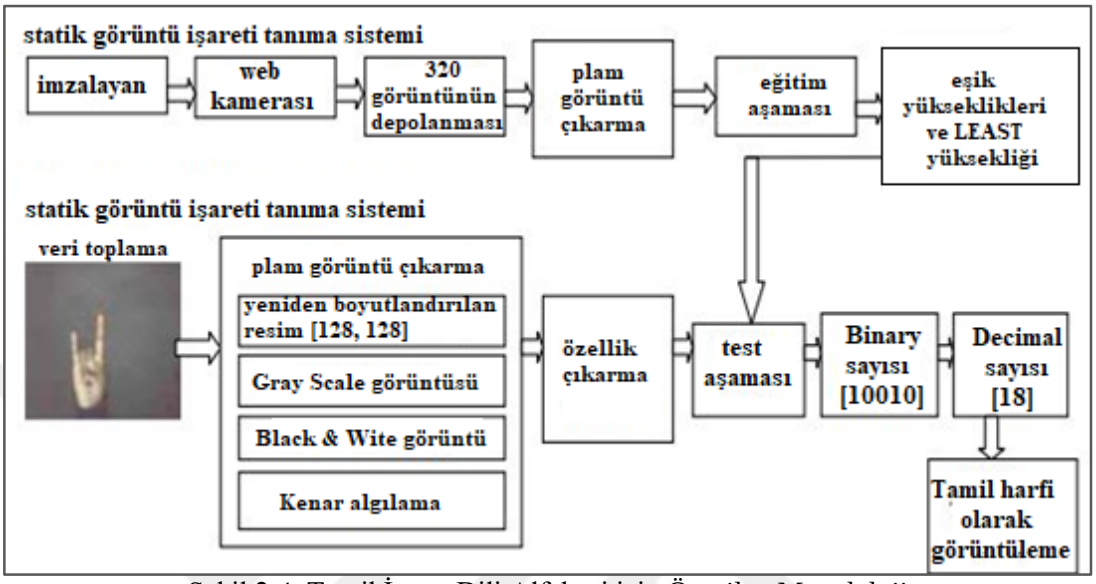
Ancak uçtan uca sonuçlar doğrulama verilerinde del=16.3, ins=6.7, WER=46.2 iken test verilerinde del=15.1, ins=7.4, WER=46.9 idi.

2.3 Görüntü İşleme Teknikleri ve İşaret Dilleri

Subha Rajam ve G. Balakrishnan, çalışmalarında görüntü işleme tekniğini, bir eldeki beş parmağın her birinin YUKARI veya AŞAĞI doğru olmasıyla ifade edilebilecek farklı kombinasyonlardan doğacak 32 farklı ikili (binary) sayıyı ondalık sayılara çevirmek için kullandılar. Çalışmalarında El (Palm) Çıkarma işlemi, öznitelik nokta çıkarma yöntemi, eğitim aşaması ve test aşaması ikili sayıları onluk sisteme dönüştürmek için kullanıldı. El Görüntüsü Çıkartma, çekilen görüntülerin 128x128 piksel çözünürlüğe yeniden boyutlandırıldığı anlamına gelir, ayrıca RGB renkli görüntüler siyah beyaz, gri tonlamalı (grayscale) görüntülere dönüştürülür. Öznitelik Nokta Çıkarma, görüntünün el çıkarma yöntemine uygulanmasından sonra, görüntüler nihayetinde sadece dış kenar görüntülerine, yani ayrıt imgelerine dönüşür. Görüntüdeki her piksel 0 veya 1'dir. Sol-Sağ ve Sağ-Sol tarama yöntemi, kenar görüntüsünün düz parmak konumunu belirlemek için kullanılır. Bu tür yöntemler statik ve dinamik nesneleri dönüştürebilir. Bundan sonra, ikili sayılar dönüştürme algoritması kullanılarak ondalık sayılara dönüştürülür; ve son olarak ondalık sayılar 12 ünlü, 18 ünsüz harfe ve 1 Aayutha Ezhuthu'ya karşılık gelen Tamil harflerine dönüştürülür.

Önerilen yöntem, 12 sesli harf, 1 Aayutha Ezhuthu ve 18 Tamil ünsüzünde her biri YUKARI ve AŞAĞI 5 parmak konumunu temsil eden 32 ikili görüntü kombinasyonu almaktı. Bu görüntüler, bir web kamerası aracılığıyla farklı

ortamlarda 10 kez tekrar alınırve toplam 320 fotoğrafa ortaya çıkar. Görüntüleri ayrı bir klasörde sakladılar. Tamil İşaret Dili Alfabesi için önerdikleri yöntem şekil 2.4'te gösterilmektedir.



Şekil 2.4. Tamil İşaret Dili Alfabesi için Önerilen Metodoloji

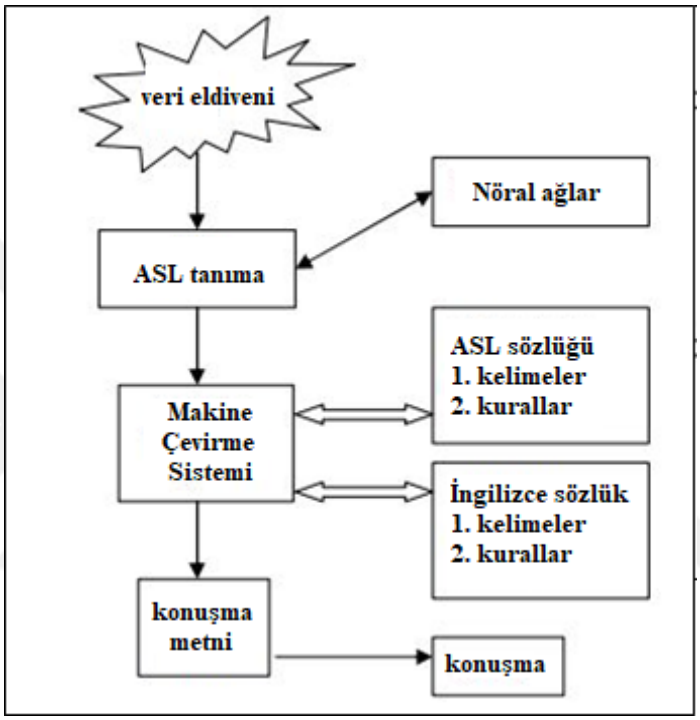
Bu iki farklı şekilde kullanılabilir; Statik görüntü işaret tanıma sistemi veya dinamik görüntü işaret tanıma sistemi. Bu iki yöntem veri toplama, el görüntüsü çıkarma, öznitelik nokta çıkarma, eğitim aşaması ve test aşamasını gerektirir. Statik görüntü tanıma sistemi kullanılarak test edildiğinde, 320 görüntüde %96.87 işaret tanıma doğruluğu elde edildi, sadece 10 görüntü doğru şekilde sınıflandırılamadı. 32 görüntünün gerçek zamanlı çekilmesi durumunda ise (5'er görüntü) toplam 160 görüntü'de, 2 sembol dışında tüm görüntüler doğru sınıflandırıldı, yani %98.75'lik doğru sonuç elde edilmiştir (Swetha ve Mary, 2019).

2.4 Sensör Eldivenleri Kullanımıyla İşaret Dillerini Tanıma

Mehdi ve Khan (2002) tarafından yapılan araştırmalarda işaret dili hareketlerini yorumlamak için sensör eldivenleri kullanma fikrini tartıştılar.

Bu, "Konuşan Eller (Talking Hands)" kampanyası başlatılarak ve etkileri analiz edilerek mümkün olmuştur. Projede, Amerikan İşaret Dili sinyallerini tespit etmek ve İngilizce cümlelere çevirmek için bir sensör eldiveni kullanıldı. Bu değerler 24 İngilizce alfabe içinde gruplandırılmıştır ve araştırmacılar ayrıca iki noktalama işareti

eklediler. Programları, hareket algılamayı pratik yapmadan optimize etmeyi amaçlamaktadır, böylece sistemin halka açık yerlerde, uzun eğitim oturumları gerektirmeksizin kullanılması mümkün olacaktır. Hareket yakalama ve tanıma hızı, yavaş veya hızlı bir performansla ASL'nin işaretlerini sergileyenlere göre değiştirilebilir. Bir işaret dilini konuşulan dile tam olarak çevirebilen bir uygulamanın modeli şekil 2.5'te gösterilmektedir.



Şekil 2.5 İşaret dilini konuşulan dile tam olarak çeviren bir uygulama modeli

Sistem, bacaklar, dirsekler, kafa vb. gibi vücudun diğer bölümlerinin şeklini veya hareketini değil, yalnızca elin ana hatlarını yakalayarak çalışır. Bu sistemde yazılımın doğruluk oranı % 88'dir. Ortaya çıkan materyal, tren istasyonları, havaalanları ve oteller gibi halka açık yerlerde kullanılabilir. Sağır ve dilsiz insanlar da bunu kullanarak bir konuşma, seminer veya ders verebilirler. Amerikan İşaret Dili'nin tamamını İngilizce'nin günlük konuşma diline çevirebildiklerini varsayarak, bu çip yapılı çeviri ünitesi ile kullanışlı ve taşınabilir bir donanım cihazı oluşturulabileceği belirtilmektedir. Ayrıca, yerleşik hoparlörleri ve gerekli sensörlere sahip olan bu donanım cihazını kullanarak bir çift veri eldiveni ile dilsiz bir kişi herhangi bir normal kişiyle her yerde konuşabilir. Bu nedenle, uygun yerlerde gerekli

sayıda sensörle özel bir elbise de donatılabilir. Bu neredeyse sağır topluluk ile sıradan dil dünyası arasındaki boşluğu dolduracaktır (Abreu vd., 2016).

Tubaiz vd. (2015) tarafından Arapça işaret dili için yeni bir veri sınıflandırma tekniği kullanarak tanıma sistemi oluşturmak için sensör eldiveni kullandılar. Araştırmalarında, seksen kelimelik bir sözlük kullanılan grup cümlelerinden oluşan sensör tabanlı bir veri kümesi derlediler. 80 kelimelik bir kelime dağarcığı, düzensiz dilbilgisi ve cümle uzunluğu olan 40 cümle içinde kullanıldı. Veri kümesini eldiven tabanlı bir sistem kullanarak kaydettiler. Algılanan ifadeler bölümlere ayrılmış ve etiketlenmiştir. Vizyon tabanlı anlık bir ArSL yönteminde aynı cümleler kullanılmış ve bu çalışmada iki sistemi karşılaştırmışlardır. El hareketleri veri kümesine DG5-VHand'den iki veri eldiveni kullanılarak kaydedilir. Verilerin etiketlenmesi, el hareketlerini işaret dilindeki aynı kelimelerle senkronize etmek için bir kamera kullanılarak yapılır. Verilerin zamansal bağımlılığını yakalamak ve vurgulamak için düşük karmaşıklıktaki ön işleme ve öznitelik çıkarma teknikleri kullanılır. Şekil 2.6 veri toplama sürecini ve bölümlere ayrılmış bir cümleyi göstermektedir.



Şekil 2.6. Veri toplama ortamı ve bölümlere ayrılmış örnek cümle

Sınıflandırma amacıyla daha sonra modifiye bir En Yakın k-Komşuluğu (MKNN) yöntemi kullanılır. Doğru sınıflandırma amacıyla, önerilen MKNN öznitelik vektörünü kullanır. Önerilen çözüm %98.9'luk bir cümle tanıma puanı elde etti. Bulgular, aynı cümleleri kullanan görmeye dayanan yerleşik bir yaklaşımla karşılaştırılmaktadır.

Bu kıyaslamalarda önerilen çözümün sınıflandırma oranlarında oldukça üstün olmasının yanısıra, görüş/vizyona dayalı sistem kısıtlamalarını da ortadan kaldırdığı bulunmuştur. Sağ el tarafından yedi cümle kullanılmış, ancak geri kalan 33 cümle her iki el tarafından da kullanılmıştır. 24 yaşında sağ elini kullanan bir kadın çalışmalarında her cümle için 10 tekrar yaptı. Veri kümesini algılamak için iki DG5-VHand cihazı eldiveni kullanılmıştır. DG5-VHand eldiveni, beş entegre büküm sensörü ve el hareketlerinin ve el yönünün algılanmasını sağlayan gömülü 3 eksenli ivme ölçere sahiptir. Eldivenler kablosuz operasyonlar için idealdir ve pille çalışır. Eldivenlerle iletişim kurmak ve sensör verilerini toplamak için Bluetooth bağlantısı

olan bir bilgisayar kullanılır. Cümleler, fiziksel olarak ayrılmamış anlık işaret dilinde bir kelime akışından oluşur. Böyle bir durum veya sorun, işaret dilinin görmeye dayalı anlık anlayışında daha az öneme sahiptir, ihmal edilebilir. Bunun nedeni, video görüntülerinin görsel inceleme kullanılarak yüksek derecede doğrulukla manuel olarak etiketlenebilmesidir. Farklı kelimelerin sınırları bir işaret dili uzmanı tarafından cümle içinde görsel olarak tanımlanabilir. Bununla birlikte, eldivene dayalı anlık işaret dili için durum böyle değildir. Bu durumda, sensör değerlerini manuel işaretleme açısından görsel olarak incelemek mümkün değildir. Eldiven tabanlı sensör okumaları manuel etiketlemesi için bir çözüm, kameranın işaretleri kaydetmek üzere konumlandırılmasıdır. İşaretler tamamlandıktan sonra, video kayıtları kelime sınırlarını tanımlamak için sensör okumalarıyla senkronize edilebilir.

2.5 Bulanık Mantık Yöntemi ve İşaret Dilleri

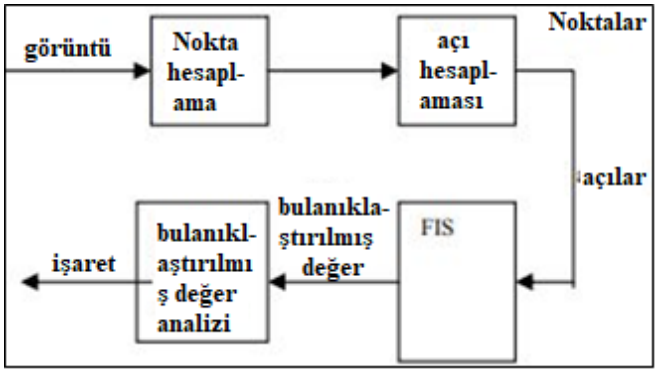
Darwish vd. (2016), araştırmalarında, el ile iletişimin en gelişmiş yolunun İşaret Dilleri (SL) olduğunu belirtirler. Bu araştırmada, Tip-2 Bulanık HMM (T2FHMM) türevli algoritma kullanılarak statik el hareketi tanıma üzerine birçok bulgu sunulmuştur. Hem öğrenme hem de tanıma aşamalarında gözlemlenebilir olarak kullanılan özellikler, matrisin geometrik yapısını en iyi şekilde gösteren Tekil Değer Ayrışmasına (SVD) dayanmaktadır. Hareket tanıma yaklaşımları görme/vizyon tabanlı bir tekniktir ve sistem el hareketlerini tanımlamak için sensörlü eldivenleri veya boyalı eldivenleri kullanmaz. T2FHMM tabanlı bir model, eğitim veri kümesinin her harf için 100'den fazla görüntü olduğu bireysel tanıma birimlerini güçlü bir şekilde tahmin etmek için elde edilen SVD fonksiyon serisinden kullanılır.Veri kümesi, böyle bir ileri geri algoritma tarafından eğitilir, daha sonra Viterbi algoritması, en yüksek hareket olasılığını seçmek için girdi görüntüsünü modele sığdırmak için veritabanını oluşturur. Sistem tamamen otomatiktir ve gerçek zamanlı çalışır. Önerilen yöntemin bulguları, belirli bir işaretteki elin yönüne bakılmaksızın sistemin çok çeşitli işaretleri doğru bir şekilde yorumlama yeteneğini göstermektedir. IT2FHMM model eğitiminde kullanılan görüntü örnekleri şekil 2.7'de gösterilmektedir.



Şekil 2.7 IT2FHMM model eğitiminde kullanılan görüntü örnekleri

Tip2 bulanık mantık, işaret dili sorunlarında bulunan normal bulanıklığı ortadan kaldırır ve çok yüksek bir hassasiyet seviyesi sağlar. Deneyler tek elli bir veritabanı üzerinde gerçekleştirildi ve farklı el hareketi en boy oranları ve karmaşık arka planlarla % 96'ya kadar tanıma doğruluğu sağlandı. Önerilen sistemin tanımlama seviyesi standart el görüntüleri için %100 ve karmaşık el görüntüleri için %95.5'tir (Darwish vd., 2016).

Ayrıca, başka bir çalışmada, Kausar vd. (2008) sağır insanlar tarafından etkili bir iletişim aracı olarak kullanılabilecek bir otomatik işaret dili tanıma sistemi oluşturmak için bulanık mantık yöntemi önerdiler. Pakistan işaret dili alfabelerini tanımlamak için çalışmalarında bulanık bir sınıflandırıcı önerilmektedir. Önerilen yöntemde her parmak ucunu ve eklemi ayırt etmek için renkli eldivenler kullanılır. Parmağın ucu ile karşılık gelen parmak eklemi arasındaki açı ölçülür ve her bir parmağın konumunu tanımlamak için kullanılır ve daha sonra bu pozisyonlar işaret dili alfabesini tanımak için kullanılır. Bulanık çıkarım sistemine girdi olarak açılar verilir ve daha sonra sistem her bir parmağın konumunu belirler ve bu konumlara göre durulaştırma (defuzzification) yapılır. PSL tanıyıcı sisteminin blok şeması şekil 2.8'de gösterilmektedir.



Şekil 2.8 PSL tanıyıcı sisteminin blok şeması

PSL'deki örnek görüntü girdisi şekil 2.9'da gösterilmiştir.



Şekil 2.9 PSL'de görüntü girdisi

Önerilen PSL tanıma yöntemini uygulamak için MATLAB 7.0 kullanıldı. Önerilen yöntemin bulguları oldukça umut vericidir. Gerekli doğruluk seviyesinde 37 işaretten sadece iki işaret tanınmamıştır. Diğer 35 işaretin tamamı yüzde 100 doğruluk sağlar. Önerilen sistem %95 doğruluk oranını yakalamıştır. Önerilen yöntemin bulguları, belirli bir işaretteki elin pozisyonuna bakılmaksızın sistemin çok çeşitli işaretleri doğru bir şekilde yorumlama yeteneğini göstermektedir. Bulanık mantık, işaret dili probleminin bünyesinde yer alan doğal bulanıklığa karşı çok yüksek bir doğruluk oranı sunar. Sistemin doğru yorumlayamadığı iki sinyal birbirine çok yakındır.

Çeşitli araştırmacılar tarafından kullanılan öznitelik çıkarma yöntemleri tablo 2.1'de gösterilmiştir.

Tablo 2.1 Çeşitli araştırmacılar tarafından kullanılan özellik çıkarma yöntemleri

(a) Dijital fotoğraf makinesi kullanarak öznitelik çıkarımı		
Kaynak	**Yöntem**	**Açıklama**
(Rezaei vd., 2008)	Kontur (çevresel hatlar)	Mesafe vektörü, hareket parametrelerini hesaplamak için bazı kontrol noktalarını çıkarmak için kullanıldı.
(Munib vd., 2007)	Gizli Markov Modeli	Mükemmel görüntü işleme ve sinir ağları ile Hough dönüşümü (Montiel vd., 2000) kullanılmıştır.
(Quan ve Jinye, 2008)(Peng ve Yulong, 2009)	Hu moments, Gabor dalgacıkları, ölçekten bağımsız öznitelik dönüşümü	Öznitelik çıkarma işlemi şunları içerir: 16 bölmeli renk histogramı; Hu momenti; 48 boyutlu Gabor dalgacıkları ve bir dizi ilgi noktası ve SIFT öznitelikleri, görüntülerin hem genel görsel özellikleri hem de yerel görsel özelliklerini karakterize eder.
(Rezaei vd., 2008)	Eliptik Fourier temsili	Bir elin Eliptik Fourier temsili kullanımı, karmaşık deforme olabilen durumlarda çok yararlı olabilir. Önerilen yöntem hem sentetik veriler hem de gerçek veriler üzerinde test edilmiştir.
(Maraqa vd., 2012)	30 öznitelik	Renk katmanlarına göre bölümlere ayrılmış renk bölgeleri genişletildi. Otuz öznitelik çıkarılır ve tek bir hareketi temsil eden bir vektörde gruplandırılır. Her vektör otuz girişten oluşur; parmak uçları ile kendileri ve bilek arasındaki açıları ifade etmek için on beş girdi ve parmak uçları arasındaki mesafeleri temsil eden on beş girdi; ve parmak uçları ile bilek arasında. Daha sonra renk alanını ifade etmek için en iyi bölgeyi belirlemek için bir küme yöntemi kullanmaya ihtiyaç vardı. Gürültü veya bir parmak ucu gibi bir nedenden ötürü rengin eksik olması durumunda bile başka bir renkle kaplanmış olsa bile, görüntünün ortasındaki varsayılan konum seçilerek bu durum çözülmüştür.

Tablo 2.1. Devamı

(Sarkaleh vd., 2009)	Haar dalgacık dönüşümü	DWT, seçilen PSL sözcüğünün görüntülerine uygulanır ve dalgacık katsayılarından bazı öznitelikler çıkarılır.
(Zhou vd., 2011)	8 öznitelik	Yazarlar bu hareketlerden sekiz öznitelik çıkarırlar: alan, çevre, elipsin hareket bölgesinin sığması için iki eksenin uzunluğu ve bunların türevleri.

Tablo 2.1 devamı

(Karami vd., 2011)	2D DWT (dalgacık fönüşümü) HAAR dalgacık	Dijital filtreler ve aşağı örnekleyiciler kullanılarak iki boyutlu (2D) DWT uygulandı. Orijinal verilere düşük ve yüksek geçişli filtrelerin konveksiyonu uygulandı ve görüntü her bir ayrışma seviyesinde belirli katsayı kümelerinde ayrıştırılabilir. Önerilen sistem için, Haar dalgacık dönüşümü, insan vücudunun yumuşak kısımlarını tanımlamadaki sadeliği ve etkinliği nedeniyle her görüntüden özellikleri türetmek için seçildi. Her görüntü için yaklaşık değerleri ve ayrıntıları elde etmek amacıyla, dokuz seviyeli ayrışma kullanılmıştır.

Özel olarak tasarlanmış cihazlar kullanılarak özellik çıkarma, tablo 2.2'de gösterilmiştir.

Table 2.2. Özel olarak tasarlanmış aygıtlar kullanarak özellik çıkarma

(Kong ve Ranganath, 2008)	Fourier Analizi	Yazarlar periyodiklik tespiti için bir Fourier analiz yaklaşımı ve yörünge tanıma için VQPCA kümeleme yöntemini kullanmışlardır. VQPCA, temel veri dağılımı ile ilgili herhangi bir varsayımda bulunmayan ve verilerin yerel olarak doğrusal temsilini bulan hibrit bir kümeleme ve yerel PCA yöntemidir.

Video kamera kullanarak öznitelik çıkarma, tablo 2.3'te gösterilmiştir.

Tablo 2.3 Video kamera kullanarak öznitelik çıkarma

(Kshirsagar ve Doye, 2010)	Kontur eşleme	Öznitelik çıkarımı için ağırlık merkezi (Centroid), Sonlu Durum Makinesi, akıllı kenar algılama kullanıldı.
(Rokade vd., 2009)	Kontur eşleme, VOP	Video karelerinden öznitelik elde etmek için Video Nesne Düzlemi (VOP) üretimi kullanıldı. Kontur eşleme kullanılarak VOP'u çıkarmak için çerçeve arası değişiklik algılama algoritması kullanıldı.
(Mekala vd., 2011)	İlgi alanı (Points of Interest), Fourier Tanımlayıcı.	Çıkarılan öznitelikler el şekli ve el hareketi olmak üzere iki alt kategoriye ayrılmıştır. Ellerin İlgi Çekici Nokta (POI) adı verilen el hareketi özellikleri kullanıldı. Öznitelik vektörü 55 öznitelikten oluşur.
(Dreuw ve Ney, 2008)	PCA-Bozulma olmaksızı el öznitelikleri	Önerilen Görsel Konuşmacı Uyumlaştırma (VSA) yöntemi, görünüme dayalı çerçeve özellikleri için sonuçları önemli ölçüde geliştirir. Sanal Eğitim Örnekleri (VTS) tarafından ek eğitim verilerinin kullanılması her zaman iyileştirmelere yol açmaz. Ancak her iki yöntemin bir kombinasyonu, literatürde bu veriler için bildirilen en iyi sonucu verir. Tanıyıcı çıkışının kelime güvenlerini kullanarak, çoklu tanıma sistemleri, sistem çıktıları üzerinde "geçiş" yapılarak kolayca birleştirilebilir. Dört farklı sistem birleştirildi: PCA Çerçeveleri üzerinde kayan pencere, PCA Çerçeveleri üzerinde kayan pencere ve el yörüngesi (HT) özellikleri, PCA Çerçeveleri üzerinde kayan pencere ve el hızı (HV) özellikleri ve Görünüm tabanlı PCA-el yamaları
(Pahlevanzadeh vd., 2007)	GFD/GCD	GFD, polar-raster örneklemeli görüntünün 2-D Fourier dönüşümünden çeşitli ölçekli Fourier katsayılarının çıkarılmasıyla elde edilmiştir.
(Shanableh ve Assaleh, 2007)	Tahmin hataları veya görüntü farklılıklarının toplamı (AD)	Ağırlıklı ve yönlü birikmiş tahmin hataları kullanılarak video tabanlı hareketlerin zamansal boyutunu ortadan kaldırmak için önerilen öznitelik çıkarma şemaları. Girdi videosu elimine edildi ve tüm sekans, video sekansındaki hareketi temsil eden bir mutlak AD görüntüsü üzerine aktarıldı. Farklı işaret hareketleri çok benzer AD görüntülerine sahip olabilir. Yazarlar, ağırlıklı yönlü birikmiş görüntü farklılıklarının kullanılmasını önermişlerdir. Ağırlıklandırma, tahmin hatalarının bir AD görüntüsünde birikme şeklini ifade eder. Bu tür farklılıklar pozitif ve negatif birikmiş tahmin hatalarına (sırasıyla AD + ve AD-) kategorize edilebilir.
(Nadgeri vd., 2010)	Oryantasyon histogramı	Dönüşümsel değişmezlik öznitelikleri çıkarıldı.

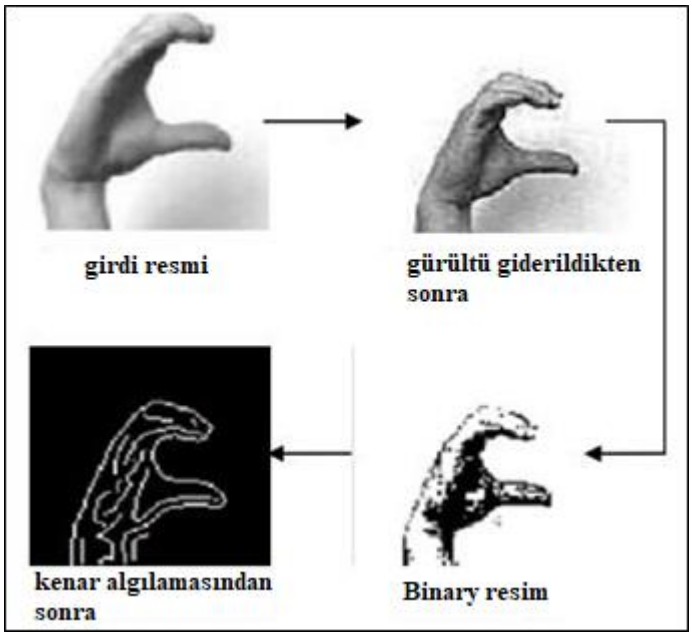
Tablo 2.3 devamı

(Dreuw ve Ney, 2008)	Damla/imge bölgesi çıkarımı	Sistemin üçüncü aşamasında, bir işareti sınıflandırmak için bir öznitelik vektörü oluşturmak için elden öznitelik çıkarımı gerçekleştirildi. Bu artışın gelişimi, damla çıkarımı görüntü işleme tekniklerini uygular ve görüntünün ikili değerine dayalı bir vektör oluşturur.
(Jinye, 2008)	Normalize edilmiş eylemsizlik anı (NMI)	Görüntünün NMI değeri, tonlanma önleyici bozulma, çeviri değişmezliği, değişmez boyut ve benzeri özniteliklere sahiptir.
(Maraqa vd., 2012)	Histogram	En net görüntüyü tanımlamak için Histogram Analizi kullanılmıştır.
(Flasiński ve Myśliński, 2010)	Grafiği öğelerine ayırma (parsing)	El duruşlarını temsil eden bir dizi Endeksli Kenar Belirsiz –PSL (IE-PSL) grafiği, bir ETPL grafik dilbilgisi ile oluşturulan biçimsel bir dil olarak ele alındı. Standart (dize) dilbilgisi ve grafik dilbilgisi arasındaki temel fark, ikincisinin üretiminde gömme dönüşümü adı verilen bir oluşumdan oluşur. Gömme dönüşümü, bir türev adım sırasında sağ taraftaki grafiği dönüştürülmüş bir grafiğe gömmeye izin verir.
(Athitsos vd., 2010)	Dinamik Zaman Çözgü(DTW)	DTW mesafe ölçüm özniteliği çıkarımı kullanıldı. Sistemin bir sorgu videosu ile en benzer veritabanı eşleşmelerini tanımlaması için işaret videoları arasında bir mesafe ölçüsü tanımlandı
(Tsechpenakis vd., 2006)	4 öznitelik kullanılmıştır	Yansıtma bilgileri, Görünen parmak sayısı ve çoklu çerçeve özellikleri, Gömülü CSS işlevleri, Gömülü kenar yönlendirme histogramları
(Teng vd., 2005)	Yerel Doğrusal Gömme	PCA ve Denetimli LLE ile lokal olarak doğrusal gömme özellik çıkarma için kullanılmıştır.
(Aran ve Akarun, 2010)	Kalman Filtresi	Manuel bileşen ve çıkarılan öznitelikler sadece el hareketi, şekli ve konumundan kaynaklanıyordu. Videolar önce el ve yüz algılama ve segmentasyon için işlendi. Daha sonra, manuel işaretler (el hareketi, el şekli, yüze göre el pozisyonu) için işaret öznitelikleri çıkarıldı. El hareketi analizi için, her bir elin kütle merkezi (CoM) takip edildi ve bir Kalman filtresi ile süzüldü. Öznitelikler, bir elipsin genişlik, yükseklik ve yönlendirme parametrelerini ve ikili el görüntüsünde hesaplanan yedi Hu momentini içerir. El pozisyonu öznitelikleri, el merkezinin yüz merkezine normalleştirilmiş yatay ve dikey mesafeleridir. Öznitelik vektörü boyutsallığı görüntü başına 32'dir (her bir el için; dört el hareketi, 10 el şekli ve iki el konumu özelliği).

2.6 Destek Vektör Makinesi İle İşaret Dilleri

Rahman ve Afrin (2013) el hareketleri ile bir bilgisayar ve insanlar arasında etkileşim kurmak için kullanılabilecek bir el işaretleri tanıma yöntemi sundu. Araştırmalarında el hareketlerini tanımak için Destek Vektör Makinesi (SVM) kullanmayı önerdiler.

Sistemlerinde, Akıllı Kenar Algılama Algoritmasıyla (Canny Edge Detection Algorithm) bir görüntünün kenarlarını tespit etmişlerdi; çünkü bu kenar algılama yöntemi karşılaştırılan diğer algoritmalarından daha iyi çalışıyordu. Bu algoritmada, hata oranını en aza indirerek, konumu en üst düzeye çıkarmak için kenarları gerçek kenarlara yakınlaştırarak ve minimum yanıt için tek bir kenar olduğunda kenarları yalnızca bir kez işaretleyerek kenarları bulmak da dahil olmak üzere belirli kriterlere göre optimum kenar algılanır. Bir Gauss fonksiyonunun ilk türevini kullanarak gerekli şartları sağlayan optimal filtre bulunur. İlk aşamada, bir Gauss filtresi ile görüntü evrişimleştirilir. Şekil 2.10, girdi görüntüsü üzerinde Akıllı Kenar Algılama işleyişinin ana fikrini göstermektedir.



Şekil 2.10 Girdi görüntüsüne Akıllı Kenar Algılama işleminin uygulanması süreci

Bundan sonra, görüntü gradyanı sıkıştırılmış görüntüyü besleyerek hesaplanır. Görüntü kenarlarının algılanması üzerine, kaynak 0° ila 180° arasında döndürülerek

görüntünün merkezi etrafında farklı açılardan çoklu paralel ışın projeksiyonlarının ölçümü gerçekleştirildi.
Önerdikleri sistem, bilgi iletmek için belirli el hareketlerini tanır ve kullanır, öznitelik vektörleri, sınıflandırıcının veri olarak çıkarılan bu öznitelikleri kullandığı sistemde Çiftdikgen Dalgacık Dönüşümü tarafından seçilmiştir. Çok sınıflı SVM kullanarak el hareketleri on kategoride gruplanır: A, B, C, D, G, H, I, L, V, Y. Sistem hareketleri anlamada başarılı şekilde çalışır. Sınırlı bir hareket dizisi üzerinde %92'lik doğru sonuçlar elde edilmiştir Rahman ve Afrin (2013), Li vd. (2017).

Ayrıca, başka bir çalışmada Naidoo vd. (1998) makalelerinde, tek özelliklerin yerel renk histogramları gibi düşük seviyeli hesaplama olduğu Destek Vektör Makineleri'ni kullanarak statik el hareketlerini sınıflandırmak için bir yöntem önerdiler. Ayrıca makalede, Güney Afrika İşaret Dili'ne (SASL) dayalı statik el görüntülerini karmaşık arka planlara karşı tanıyan bir sistem geliştirmeye odaklanmıştır. Sınıflandırma amaçlı olarak çeşitli makine öğrenme algoritmaları arasından örneğin sinir ağları, sınıflandırma ağaçları tercihi yapılmıştır. Bunun nedeni, girdi alanı boyutu çok küçük olsa bile, öncelikli bilgi eklemeden yüksek genelleme performansından kaynaklanmaktadır. Bu araştırmanın temel amacı, sağır ve işitme toplulukları arasındaki iletişime yardımcı olacak bir sistem geliştirmektir. Görme/vizyon tabanlı sistemler, jest tanıma üzerine araştırma için umut verici bir seçenek sunar. Görüntü tabanlı el formlarını belirlemeye yönelik mevcut girişimler oldukça iyi bir performans üretmiştir, ancak bunun ana nedeni özelleşmiş araçların çok büyük bir hesaplama kullanımıdır. 2D el damlacık temsilleriyle güçlü tanıma seviyelerine ulaşan Pavlovic, Sharma ve Huang'ın girişimi buna örnektir. Sistemlerinin amacı nispeten iyi sonuçlar elde etmektir, ancak buna paralel olarak zaman gereksinimi ve doğruluk yüzdesi arasında bir denge gözetilmelidir. Bununla birlikte, %90'dan fazla doğruluk elde edilmesi amaçlanmalıdır (Sharrma vd., 2017).

Önceki çalışmalar, işaret dili alanında halen çok sayıda çalışmaya ve araştırmaya ihtiyaç duyulduğunu göstermektedir. Birçok sağır ve dilsiz insan, özellikle gelişmekte olan ülkelerde, bu deneyimlerden faydalanamamıştır. Uzun vadeli hedef, herkes için hem kullanımı hem her anlamda maliyeti kolay sisteme gelmektir.

Araştırmacılar tarafından kullanılan sınıflandırma yöntemleri tablo 2.4'te gösterilmiştir.

Tablo 2.4 Araştırmacılar tarafından kullanılan sınıflandırma yöntemleri

Yöntem	Açıklama	Makale Sayısı	Kaynaklar
Sinir Ağları	Çeşitli sinir ağı sınıflandırıcıları kullanılır	6	(Munib vd., 2007) (Munib vd., 2007) (Maraqa vd., 2012) (Sarkaleh vd., 2009) (Karami vd., 2011)(Mekala vd., 2011) (Vanjikumaran ve Balachandran, 2011)
SVM	Destek Vektör Makinesi sınıflandırması	1	(Rezaei vd., 2008)
HMM	Varyantları ile HMM sınıflandırıcı	7	(Kshirsagar ve Doye, 2010) (Davydov vd., 2010) (Grobel ve Assan, 1997) (Aran ve Akarun, 2010) (Binh vd., 2005) (Zhou vd., 2011) (Davydov vd., 2010)
Bulanık Kümeler	Kullanılan diğer sınıflandırıcılarla Bulanık Kümeler.	1	(Bui ve Nguyen, 2007)
Tensör analizi	Tensör esaslı sınıflandırma önerildi.	2	(Rana vd., 2009) (Wang vd., 2010)
FSM ve DTW	Sonlu Durum Makinesi ve Dinamik Zaman Çözgü Algoritmaları.	1	(Rokade vd., 2009)
ROVER	Tanıma çıktısı oylama hatası azaltma	1	(Dreuw ve Ney, 2008)
GCD ile Öklid uzaklığı sınıflandırıcısı	Anahtar görüntülerde el şekillerinin GCD özellikleri ve Öklid uzaklık sınıflandırıcısı.	1	(Pahlevanzadeh vd., 2007)

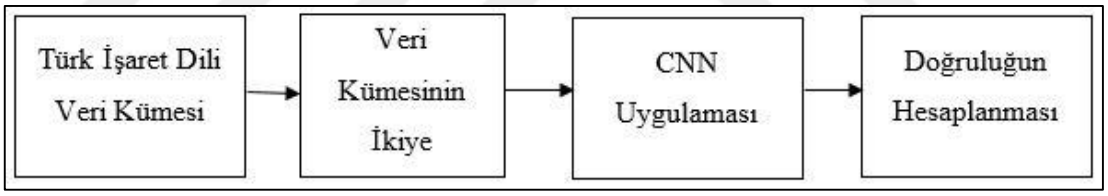
Tablo 2.4 devamı

I., II. derece Polinom sınıflandırıcısı	KNN, 1. derece polinom sınıflandırıcı ve 2. derece polinom sınıflandırıcı kullanılarak önerilen çözümün kullanıcıdan bağımsız sınıflandırılması.	1	(Shanableh ve Assaleh, 2007)
CAMSHIFT Algoritması	Sürekli Uyarlamalı Ortalama SHIFT Algoritması	1	(Nadgeri vd., 2010)
HSBN	El Şekilleri Bayes Ağı	1	(Thangali vd., 2011)
BoostMap	Optimizasyonu gömmek için bir ikili sınıflandırıcı ve artırma yöntemi Ada Boost kullanılır. Bu yaklaşım Boost Map yönteminde benimsenmiştir.	1	(Athitsos vd., 2010)
SVR	Destek Vektör Regresyon tekniği	1	(Tsechpenakis vd., 2006)
VQPCA	Vektör Niceleme Temel Bileşen Analizi (PCA)	1	(Kong ve Ranganath, 2008)

3. ARAÇ VE YÖNTEMLER

3.1 Arkaplan

Bu bölüm, etkili bir tanıma ve öngörü sisteminin oluşturmak için kullanılabilecek yöntemleri açıklamaktadır. Bu sistemin temel amacı, Türk İşaret Dili alfabesini ve bazı sembolleri otomatik olarak tanımak ve öngörmektir. Görüntüler evrişimli sinir ağı yöntemi kullanılarak sınıflandırılır. Sistemin uygulaması farklı aşamalardan oluşmaktadır. İlk olarak, veri kümesi (TSL-Türk İşaret Dili) eğitim ve test verileri olarak sistem tarafından ikiye ayrılır. Bir sonraki adımda, veri kümesi görüntüleri evrişimli sinir ağı katmanları tarafından işlenir. CNN aşaması da dört katmandan oluşur; evrişimli katman, doğrusal olmayanlık katmanı (ReLU), birleştirme katmanı ve tamamen bağlantılı katman. Veri kümesini işledikten sonra, sistem Türk İşaret Dili alfabesindeki her bir harf için doğruluğu yada doğruluk yüzdesini verir. CNN uygulamasının da dahil olduğu genel sistemin adımları şekil 3.1'de gösterilmiştir.

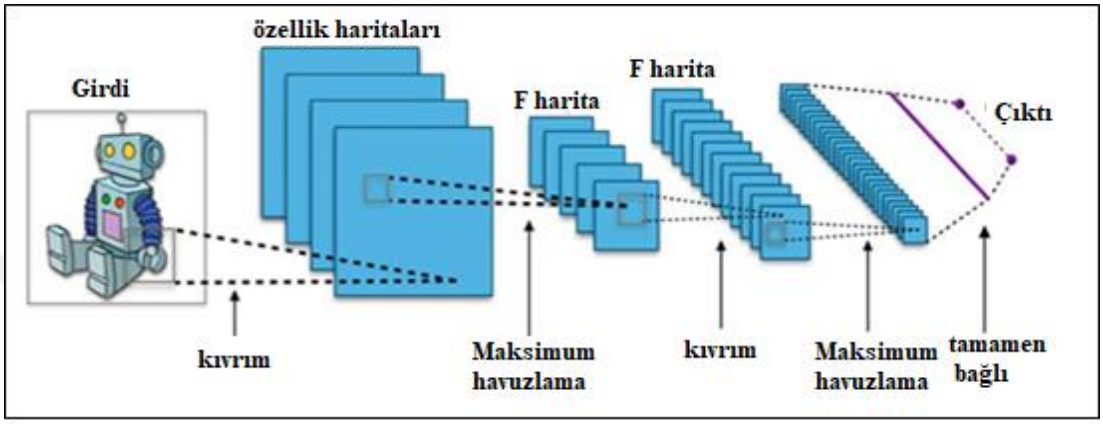


Şekil 3.1 Genel sistem adımları

3.2 Evrişimli Sinir Ağları

Evrişimli sinir ağı (CNN), görüntü tanıma ve işleme amaçlı kullanılan ve piksel bilgilerini işlemek üzere özel olarak tasarlanmış bir tür yapay sinir ağıdır. CNN oldukça güçlü bir görüntü işleme yöntemi ve derin öğrenmeye dayanan, hem tanımlayıcı hem üretici görevleri yerine getiren bir yapay zeka (AI) türüdür; nesne ve video tanıma gibi makine vizyonuyla, tanıma sistemlerinde ve doğal dil işlemede kullanılır. CNN daha az işlem süresi için tasarlanmış çok katmanlı algılayıcı gibi bir sistem kullanmaktadır. CNN girdi katmanı, saklı bir katman, birkaç evrişimli katmandan oluşan çıktı katmanı, Doğrusal Olmayan (ReLU) katman, birleştirme katmanları ve tamamen bağlantılı katman gibi katmanlardan oluşur, Şekil 3.2'de evrişimli sinir ağı katmanlarının genel çalışma şekli gösterilmektedir (Kim, 2014).

Ayrıca, CNN'ler veya ConvNet'ler, görüntü sınıflandırma ve tanıma alanında ağırlığı olan sinir ağlarıdır (Bantupalli ve Xie, 2019). CNN'ler, hayvanların görsel korteksinde nöronlar arasındaki iletişim örüntülerinin işleyişi bakımından biyolojik süreçler gibi bir performans vermesi amacıyla modellenmiştir (Bantupalli ve Xie, 2019; Matsugu vd., 2003).

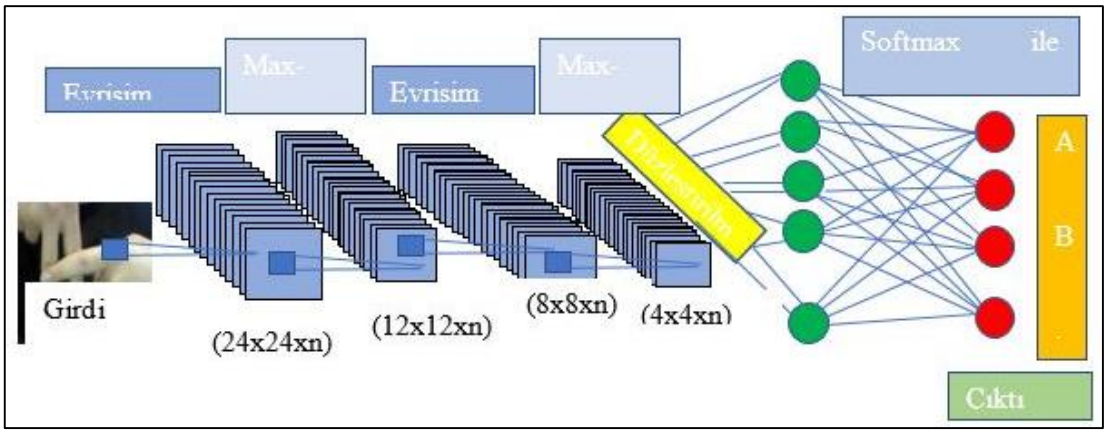


Şekil 3.2 Evrişimli Sinir Ağlarının Genel Modeli (Bantupalli ve Xie, 2019, Matsugu, 2003)

3.2.1 Evrişimli Sinir Ağlarının İşleyişinin Genel Açıklaması

Evrişimli Sinir Ağı (ConvNet/CNN), girdi olarak bir görüntüyü algılayabilen, görüntüyü farklı yönlerden, örneğin görüntüde yer alan nesnelere önem atfederek ve bu nesneleri birbirinden ayırarak değerlendirebilen bir tür derin öğrenme algoritmasıdır. ConvNet'te ön işleme, diğer sınıflandırma algoritmalarından çok daha kısa sürer. ConvNet bu filtreleri/karakteristikleri öğrenebilirken, daha önce yaygın olarak kullanılmakta olan yöntemlerde filtreler yeterli eğitim ile manuel olarak üretilmiştir. Bir ConvNet'in yapısı nöronların insan beyni içindeki iletişim modeline benzer ve bu yapı görsel korteksin organizasyonundan esinlenilerek ortaya çıkarılmıştır. Görme alanının sadece Alıcı Alan olarak bilinen sınırlı bir bölgesinde, bireysel nöronlar uyaranlara tepki verir. Bu alanların veya bu alanlarda oluşan imgelerin biraraya gelen bir koleksiyonu, tüm görme alanını kapsar (Saha., 2018).

Bu tezde, Türk İşaret Dili görüntülerini sınıflandırmada kullanılan ana yöntem, evrişimli sinir ağıdır. Örneğin şekil 3.3'te, Türk İşaret Dili harflerinden birini içeren görüntüyü tanımada izlenen adımları gösteren basit bir örnek verilmektedir.



Şekil 3.3 ConvNet'in TSL'de yer alan (A) harfini tanıması süreci (Saha., 2018)

Şekil 3.2'deki CNN, 28x28 boyutlarındaki bir görüntüyü (Türk İşaret Dilinden bir görüntü) girdi olarak almaktadır.

Şekil 3.2'de yer alan görüntü üzerinde dört temel işlem uygulanmaktadır:

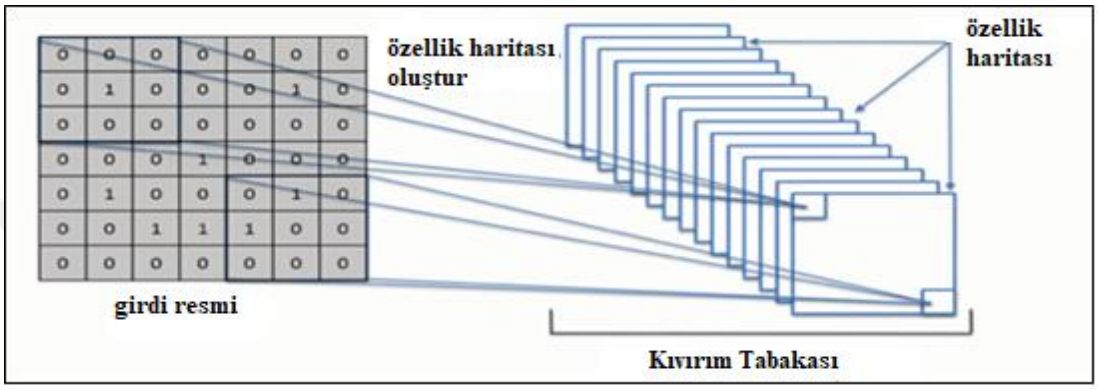
1. Evrişim
2. Doğrusal Olmayanlık (ReLU)
3. Birleştirme veya Alt Örnekleme
4. Sınıflandırma (Tam Bağlantılı Katman)

Bu işlemler herhangi bir CNN'nin temel bloklarını oluşturur. Evrişimli sinir ağındaki her aşamanın belirli bir işlemi yani fonksiyonu vardır.

3.2.2 Evrişim Adımları

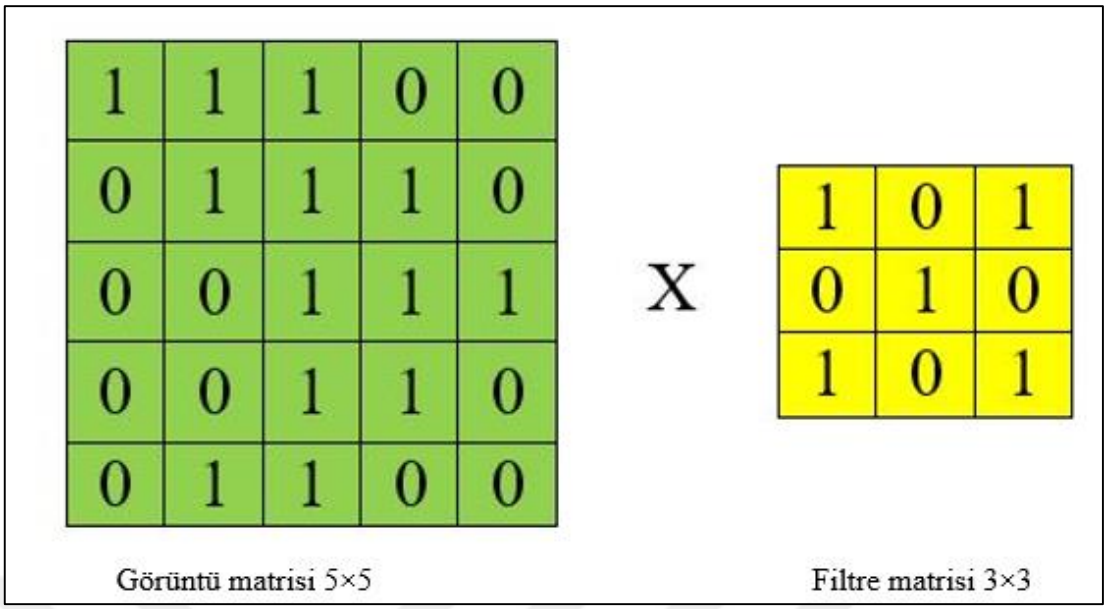
ConvNet adını "evrişim" (*İng.* convolution) işleminden alır. Evrişimin ConvNet bünyesindeki temel fonksiyonu, görüntü verilerinden öznitellikler çıkarımı yapmasıdır. Evrişim, küçük kareler üzerindeki girdi verilerini kullanarak nesne özniteliklerini öğrenir ve pikseller arasındaki uzamsal ilişkiyi gözetir (Bantupalli ve Xie, 2019). Evrişimin amacı, girdi verilerinden görüntü, video veya sıralı bilgi elde etmektir. Evrişim, görüntü özniteliklerini öğrenerek ve pikseller arasındaki uzamsal ilişkileri hesaba alarak veya muhafaza ederek çalışır. Her görüntüyü bir piksel matrisi olarak ele alacak olursak, evrişim, veri bloklarını verili matristen daha küçük bir boyuta indirgemektedir. CNN terminolojisinde 3x3'lük bir matrise "filtre" veya

"öznitelik detektörü/algılayıcısı" denir ve aynı filtrenin görüntünün üzerinde gezdirilmesiyle oluşturulan matrise ise "Evriştirilmiş Öznitelik" veya "Öznitelik Haritası" denir. Aynı işlem, girdi görüntüsü bir dizi öznitelik haritasına dönüştürülene kadar tekrarlanır (Bantupalli ve Xie, 2019). Şekil 3.4'te evrişim katmanından istenilen çıktıyı elde etmek için girdi görüntüsünden öznitelik haritalarının nasıl elde edileceğini model olarak gösterir.



Şekil 3.4 Evrişim katmanında öznitelik haritasının elde edilişi (Bantupalli ve Xie, 2019)

Evrişimin işleyişini daha net anlatabilmek için, bir sonraki örnekte bu işleyiş detaylı bir şekilde açıklanacaktır. Bir görüntünün piksel değerlerinin sadece 0 ve 1'lerden oluştuğunu ve görüntü boyutlarının da 5×5 olduğunu ve 3×3 boyutlarında bir filtre matrisimiz olduğunu varsayalım. Aşağıdaki şekillerde, evrişimin nasıl çalıştığını net ve basit bir şekilde göstermeye çalışacağız: Evrişim işleminde, $(5 \times 5 \times 1)$ boyutlarına sahip A görüntü girdi matrisi (genellikle önceki katmandan aldığı değerleri taşır), ve genellikle girdiden daha küçük boyutlarda $(3 \times 3 \times 1)$ olan K filtre matrisi (çekirdek/kernel olarak da adlandırılır) ve çıktı da B adı verilen yeni bir matristir (evriştirilen öznitelik). Bu süreçte her işlemde (n) A'nın değiştiği, ancak K'nın sabit kaldığı görülecektir. $5 \times 5 \times 1$ görüntü matrisini $3 \times 3 \times 1$ boyutlarında bir filtre/çekirdek matrisiyle evriştirdiğimizde $3 \times 3 \times 1$ boyutlarında evrişim öznitelikli yeni bir matris elde edilişi şekil 3.5'de gösterilmiştir.



Şekil 3.5 5×5×1'lik görüntünün 3×3×1'lik çekirdekle evriştirilip 3×3×1'lik özniteliğin elde edilmesi (1) (Bantupalli ve Xie, 2019)

Evrişim işleminin matris işlemleri dizisi olarak işleyişi şu şekildedir:

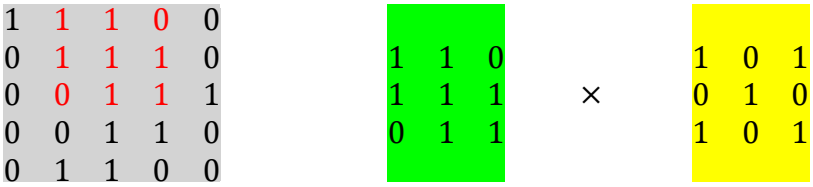
K 3 × 3 × 1'lik bir matrisse, B matrisinin ilk girdisi şöyle hesaplanır:

- A'nın 3 × 3'lük ilk alt matrisi seçilir.
- Her bir girdi karşılık gelen K'daki ağırlığıyla çarpılır.
- Tüm çıktılar (çarpımlar) toplanır.

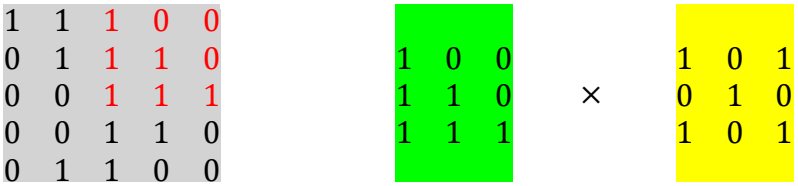
Bu son iki adım, A'nın alt matrisiyle K'nin iç çarpımına eşittir, aslında çok boyutlu iki vektörden bir skaler yani bir tam sayı elde edilmektedir. Bundan sonra K matrisi bir sonraki elemanı elde etmek için sağa doğru kaydırılır ve bu şekilde işlem, A'nın her bir alt matrisi ile tekrarlanır.

A= girdi görüntüsü	A'nın alt matrisleri	×	K
1 1 1 0 0 0 1 1 1 0 0 0 1 1 1 0 0 1 1 0 0 1 1 0 0	1 1 1 0 1 1 0 0 1	×	1 0 1 0 1 0 1 0 1

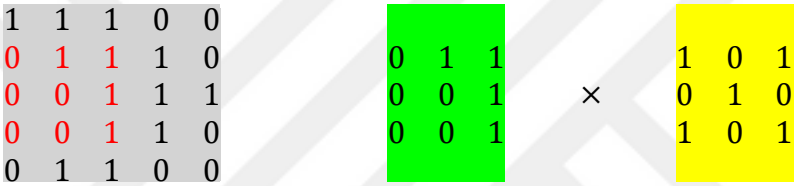
(1×1) +(1×0) +(1×1) +(0×0) +(1×1) +(1×0) +(0×1) +(0×0) +(1×1)=4



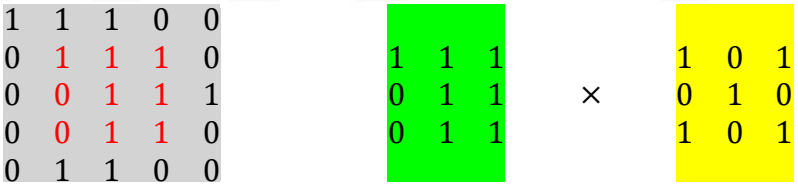
(1×1) +(1×0) +(0×1) +(1×0) +(1×1) +(1×0) +(0×1) +(1×0) +(1×1)=3



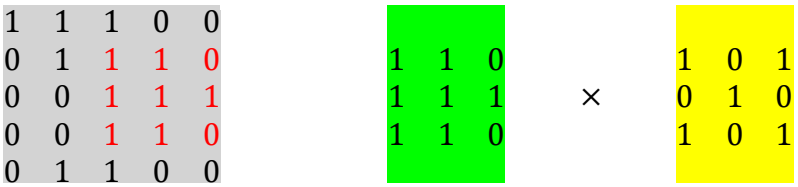
(1×1) +(0×1) +(0×1) +(1×0) +(1×1) +(0×1) +(1×1) +(1×0) +(1×1)=4



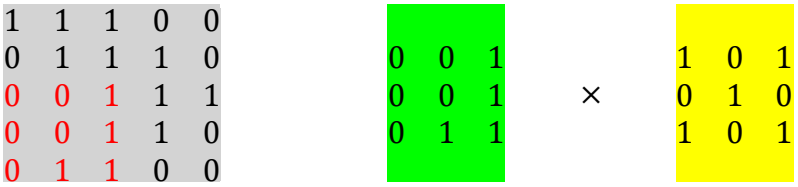
(0×1) +(1×0) +(1×1) +(0×0) +(0×1) +(1×0) +(0×1) +(0×0) +(1×1)=2



(1×1) +(1×0) +(1×1) +(0×0) +(1×1) +(1×0) +(0×1) +(1×0) +(1×1)=4



(1×1) +(1×0) +(0×1) +(1×0) +(1×1) +(1×0) +(1×1) +(1×0) +(0×1)=3



(0×1) +(0×0) +(1×1) +(0×0) +(0×1) +(1×0) +(0×1) +(1×0) +(1×1)=2

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

0	1	1
0	1	1
1	1	0

×

1	0	1
0	1	0
1	0	1

(0×1) +(1×0) +(1×1) +(0×0) +(1×1) +(1×0) +(1×1) +(1×0) +(0×1)=3

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

1	1	1
1	1	0
1	0	0

×

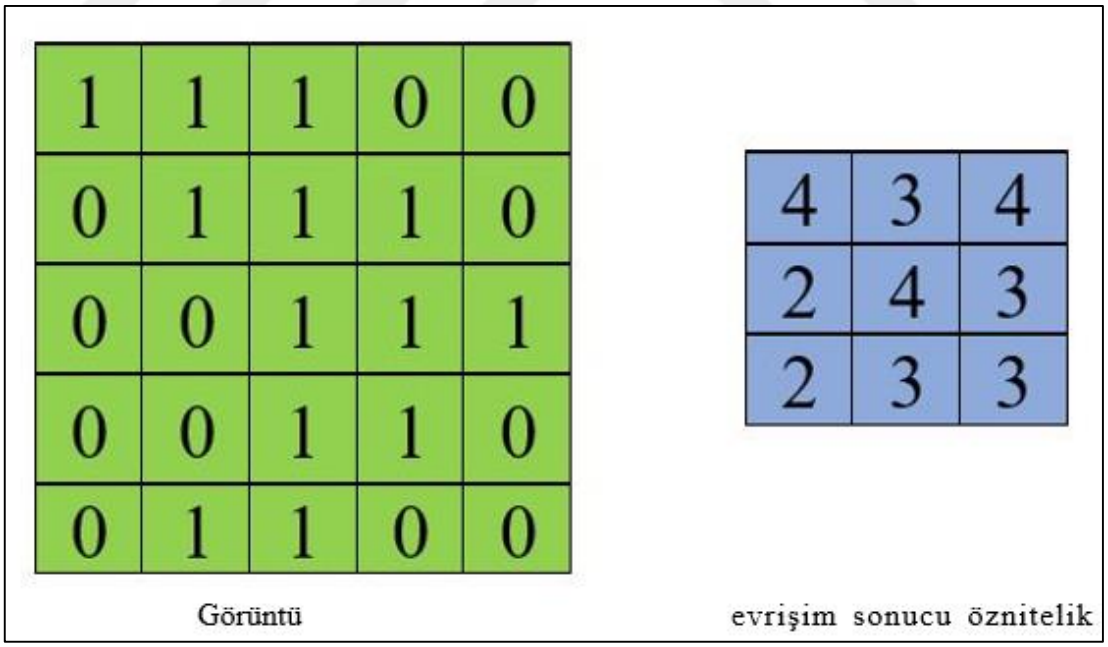
1	0	1
0	1	0
1	0	1

(1×1) +(1×0) +(1×1) +(1×0) +(1×1) +(0×0) +(1×1) +(0×0) +(0×1)=4

B (evrişim sonrası)=

4	3	4
2	4	3
2	3	4

Görüntü matrisini filtre matrisi ile çarptıktan sonra, evrişim işleminin sonucu olarak şekil 3.6'daki matris elde edilir:



Şekil 3.6 5×5×1'lik görüntü matrisiyle 3×3×1'lik çekirdeğin evrişim işlemine sokulması ve 3×3×1'lik özniteliğin elde edilmesi (2) (Bantupalli ve Xie, 2019)

Eğitim sürecinde CNN bu filtrelerin prensiplerini kendi kendine öğrenir. Ne kadar çok filtre kullanılırsa, o kadar fazla nesne özniteliği çıkarılır ve görüntülerdeki saklı örüntüleri tanımlamada ağımız o kadar başarılı olur. Öznitelik Haritasının boyutunu (Evrişim Özniteliği) üç parametre belirler; bu parametrelerin, evrişim adımı gerçekleşmeden önce belirlenmesi gerekir:

Derinlik (Depth): Derinlik, evrişim işlemi için kullandığımız filtre sayısıdır. Şekil 3.7'de gösterilen ağda, orijinal A harfini dönüştürmek için üç ayrı filtre kullanılır. Bu nedenle, gösterildiği gibi üç ayrı öznitelik haritası oluşturulur. Bu şekilde elde edilen üç öznitelik haritası bir dizi 2 boyutlu matris olarak düşünülebilir, dolayısıyla öznitelik haritasının 'derinliği' üç olacaktır.



Şekil 3.7 CNN'de öznitelik haritası

Adım Uzunluğu (Stride): Stride yada adım uzunluğu, filtre matrisinin giriş matrisi üzerinde bir adım sonra kaç piksel atlayacağıdır. Adım uzunluğu yada stride 1'se, filtreler her bir adımda yalnızca bir piksel hareket ederler. Adım uzunluğu 2 olduğunda, filtreler her seferinde 2 piksel atlar. Daha büyük bir adım uzunluğu kullanıldığında daha küçük boyutlu öznitelik haritaları oluşturulacaktır. Şekil 3.8 ve şekil 3.9'da adım uzunluğu 1 veya 2 olarak alındığında uygulamayla elde edilen öznitelik haritaları gösterilmektedir.

CNN'de belirlenecek adım uzunluğu, aynı anda hem avantajları hem de dezavantajları olan seçeneklerden biridir. Alanlara bakıldığında, bir sonraki alt matrisin komşularıyla birçok örtüşmesinin yada üst üste binmesinin olacağı görülebilir, adım uzunluğunun kontrol edilmesiyle üst üste binmeler de kontrol edilebilir. Örneğin; 7×7 matrisinde adım uzunluğu, sadece bir olursa 5×5 lik bir matris elde edilir, ancak adım uzunluğu 2 ise 3×3 lük bir matris ortaya çıkacaktır.

Basitçe, daha büyük bir adım uznluğu sadece üst üste binmeyi azaltmaz, aynı zamanda çıktı boyutunu da azaltır (Albawi vd., 2017).

Şu ana kadar ele aldığımız verileri formülize edecek olursak, $A \times A$ boyutlarındaki görüntünün ve $K \times K$ boyutlarındaki filtrenin işleme sokulması durumunda, çıktının boyutu O şu şekilde hesaplanır:

$$O(\text{Çıktı}) = 1 + \frac{A-K}{S} \tag{3.1}$$

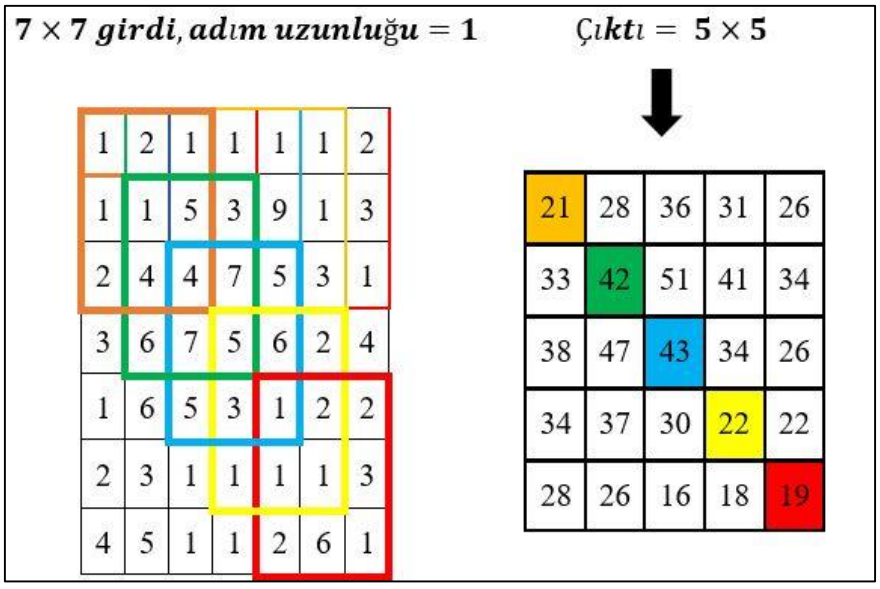
$$O(\text{Çıktı}) = 1 + \frac{\text{görüntü boyutu} - \text{çekirdek}}{\text{adım uzunluğu}}$$

Burada, A girdi boyutu, K filtre boyutu ve S adım uzunluğudur.

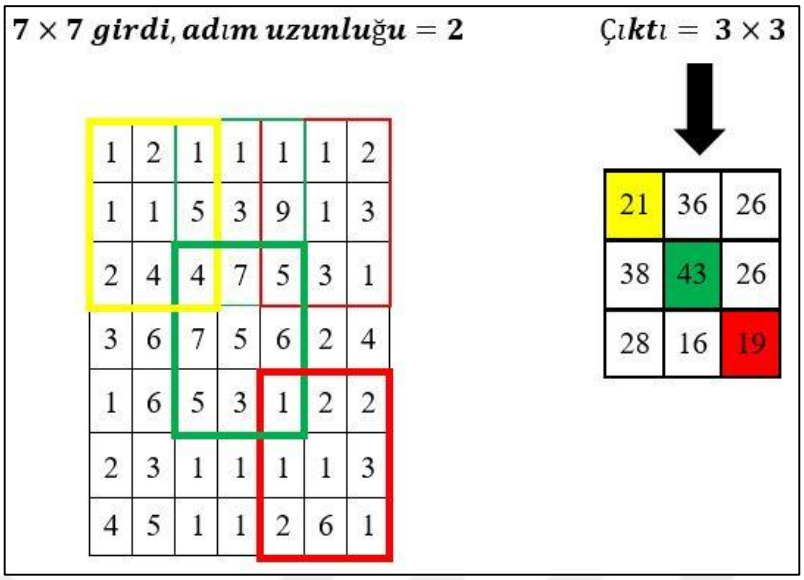
Şekil 3.8'de girdi görüntüsü boyutu $7 \times 7'$dir ve adım uzunluğuysa 1'dir, bu nedenle çıktı matrisi 5×5 boyutlarındadır.

$O(\text{Çıktı}) = 1 + \frac{7-3}{1} = 5$ Şekil 3.9'da ise girdi görüntüsü boyutu $7 \times 7'$dir ve adım uzunluğu 2'dir, bu nedenle çıktı matrisi 3×3 boytlarındadır.

$$O(\text{Çıktı}) = 1 + \frac{7-3}{2} = 3$$



Şekil 3.8 Adım uzunluğu 1 (Albawi vd., 2017)



Şekil 3.9 Adım uzunluğu 2 (Albawi vd., 2017)

Sıfırla doldurma/dolgulama (Zero-padding): Genellikle girdi matrisinin çevresini sıfırlarla doldurmak kolaydır, bu sayede filtre girdi görüntü matrisinin sınır değerlerine de uygulanabilir. Sıfırla doldurmanın güzel bir özelliği, öznitelik haritasının boyutlarını kontrol edebilmesidir. Sıfır dolgusu kullanımına geniş evrişim denir ve sıfır dolgusu kullanmamak küçük bir evrişim olacaktır. Evrişim aşamasının dezavantajlarından biri, görüntünün sınırlarında ortaya çıkabilecek bilgi kaybıdır. Sadece filtre slaytları alındığında, hiçbir zaman görüntünün tamamını görme şansı elde edilemeyecektir. Dolayısıyla sıfır dolgusu kullanmak, sözkonusu sorunu çözmek için basit ve etkili bir yöntemdir. Sıfır dolgusunun avantajlarından biri çıktı boyutunu yönetebilmektir. Bununla birlikte, sıfır dolgusu uygulanarak başlangıç girdisiyle aynı olan 5×5 boyutlarında bir çıktı elde edilecektir. Ancak sıfır dolgusu yönteminin temel amacı çıktı boyutunun adım uzunluğuyla birlikte küçülmesini önlemektir. Sıfır dolgusunun dahil olduğu denklem şöyledir:

K filtre boyutuysa

$$P(sıfır\ dolgusu) = \frac{K-1}{2} \tag{3.2}$$

$K = 3 \times 3$ ise o zaman: $P(sıfır\ dolgusu) = \frac{3-1}{2} = 1$

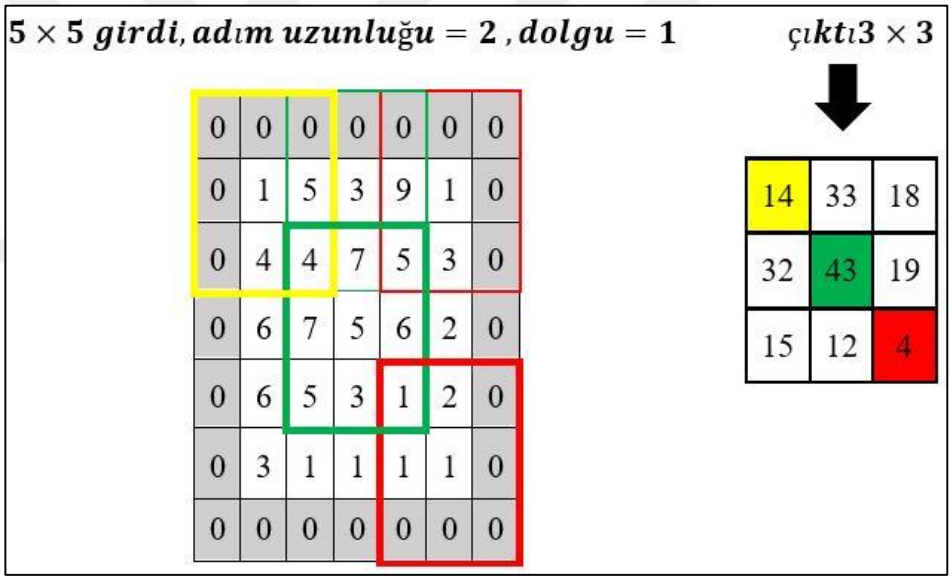
$$O(\text{Çıktı}) = 1 + \frac{A+2P-K}{S} \tag{3.3}$$

$$Çıktı\ görüntüsü = 1 + \frac{(görüntü\ boyutu + (2 \times dolgu)) - filtre\ boyutu}{adım\ uzunluğu}$$

Yukarıda verilen denklemlerde A girdi görüntüsünün boyutu; K filtre boyutu; P sıfır dolgusunun katman sayısı (evrişim sıfır dolgusu boyutu); ve S adım uzunluğudur (Albawi vd., 2017).

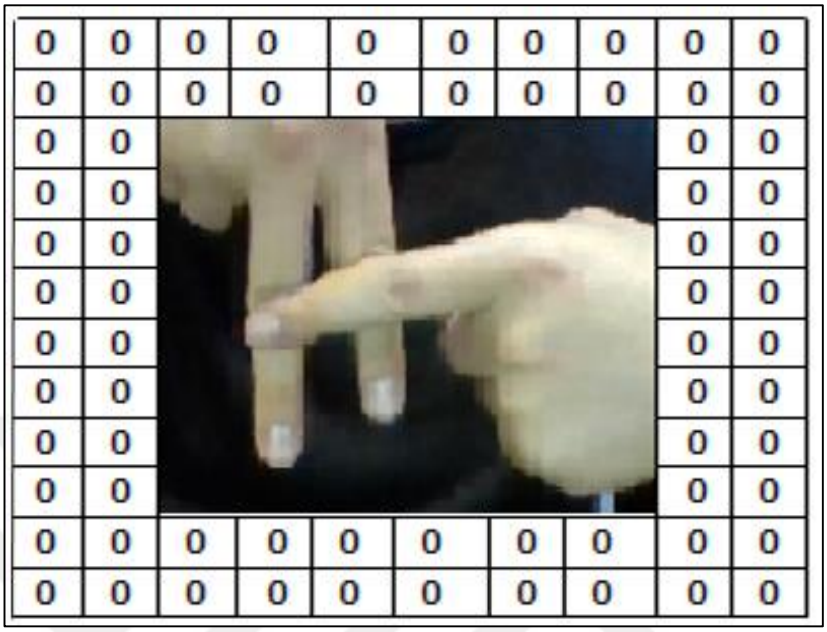
Şekil 3.10'da girdi görüntü boyutu 5×5 için uygulama görülmektedir, burada adım uzunluğu=2 ve dolgu=1 ise, çıktı 3×3 olur, bunların hepsi aşağıdaki denkleme göre belirlenir:

$$O(çıktı)1 + \frac{5+(2\times1)-3}{2} = 3$$



Şekil 3.10 Adım uzunluğu 2, sıfır dolgusu 1'ken evrişim (Albawi vd., 2017)

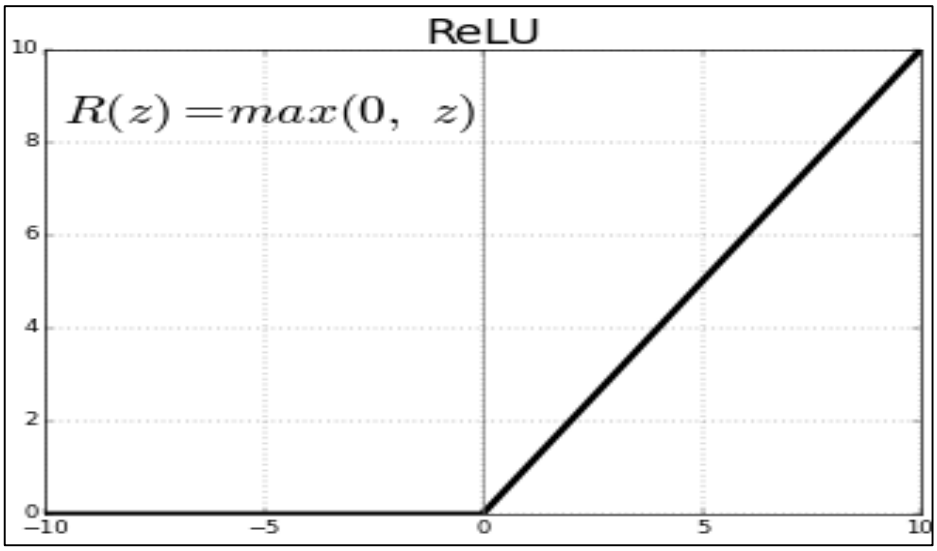
Şekil 3.11'de Türk İşaret Dilinde A için sıfır dolgusu görülmektedir, burada sıfır dolgusu çıktı boyutunun küçülmesini engeller.



Şekil 3.11 Sıfır dolgusu (Albawi vd., 2017)

3.2.3 ReLU

ReLU İngilizce'de Rectified Linear Unit (Rektifiye Doğrusal Birim) kelime grubunun kısaltmasıdır ve doğrusal olmayan bir işlemdir (Bantupalli ve Xie, 2019; Krizhevsky vd., 2012). ReLU, evrişim işleminden sonraki aşamadır. Her öğeye ayrı ayrı uygulanır ve öznitelik haritasında bütün negatif piksel değerlerini sıfır olarak değiştirir. Gerçek dünyadaki eğitim doğrusal olmadığı ve CNN'nin de bunu modellemesi beklendiği için, ReLU'nun amacı doğrusal olmamayı sisteme dahil etmektir. ReLU Aktivasyon Fonksiyonu şekil 3.12'de gösterilmiştir.

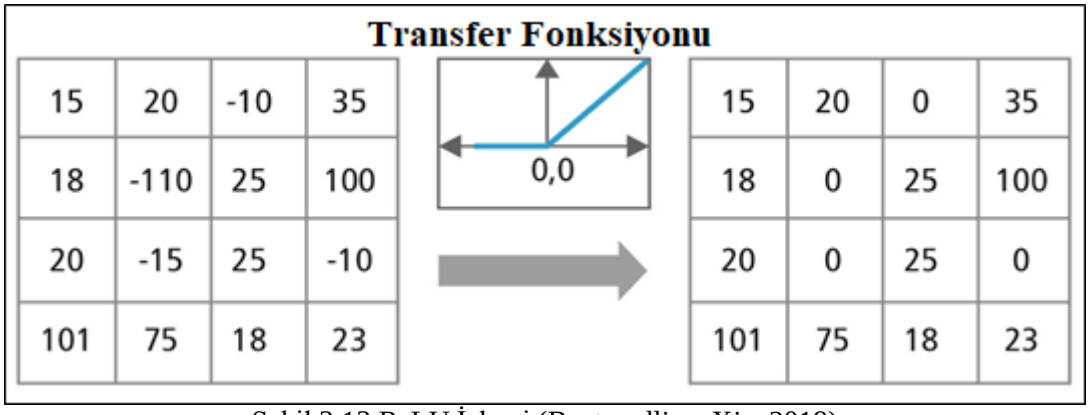


Şekil 3.12 ReLU aktivasyon fonksiyonu (Bantupalli ve Xie, 2019)

ReLU fonksiyonu, girdinin sayı değeriyle sıfır sayısı arasında büyük olanı seçer.

Çıktı= Max (Sıfır, Girdi)

Sigmoid ve tanh gibi diğer aktivasyon fonksiyonlarıyla karşılaştırıldığında, ReLU daha basit matematiksel işlemler gerektirdiği için, hesap gereksinimi bağlamında düşük maliyetli bir aktivasyon fonksiyonudur, bu husus bir sinir ağını tasarlarken dikkate alınması gereken önemli bir noktadır (Bantupalli ve Xie, 2019). ReLU işlemi şekil 3.13'de gösterilmiştir.



Şekil 3.13 ReLU İşlemi (Bantupalli ve Xie, 2019)

3.2.4 Birleştirme Adımı

Birleştirme katmanının amacı, öznitelik haritalarının çözünürlüğünü küçülterek uzamsal değişmezlik elde etmektir. Burada, birleştirilen her öznitelik haritası önceki katmanın öznitelik haritalarından biriyle eşleştirilir. Birimler Şekil 3.3'te gösterildiği gibi küçük bir birim yamasındaki verileri entegre eder. Bu birleştirme penceresi rastgele olabilir ve pencereler üst üste gelebilir. Birleştirme ayrıca diğer katmanların karmaşıklığını azaltmanın yanı sıra hesaplamaları hızlandırmayı ve tespit ettiği bazı öznitelikleri biraz daha sağlam hale getirmeyi de amaçlar (Scherer vd., 2010). Birinci evrişimsel katmanın amacı, veri kümesi boyunca yer alan girdi görüntülerinden yerel bölgelerde bulunan ortak olan örüntüleri çıkarmaktır. Bu, bir şablonun veya filtrenin giriş görüntüsünün pikselleri üzerinde gezdirilmesi, şablonun görüntüdeki her konumda iç çarpımının hesaplanması ve katmandaki her filtre için bir öznitelik haritası *c* olarak çıkarılmasıyla elde edilir. Bu, görüntünün her bir bölümünün modele ne kadar iyi uyduğunun bir ölçüsüdür. Daha sonra her özellik haritası *c*'ye, her bir birim elemanı için *a=f (c)* doğrusal olmayan fonksiyon *f ()* uygulanır. Daha sonra elde edilen aktivasyonlar birleştirme katmanına aktarılır. Bu da verileri bir dizi küçük yerel alan R'de bir araya getirir, bunun çıktısı da birleştirilmiş bir fonksiyon haritası *s*'dir (daha küçük boyutlu). Her bir öznitelik haritası *c* için toplama fonksiyonunu *pool ()* olarak gösterecek olursak (Zeiler ve Fergus, 2013), elde edeceğimiz formül şu olur:

$$S_i = pool\big(f(c_i)\big)\ \forall_i \in R_j \tag{3.4}$$

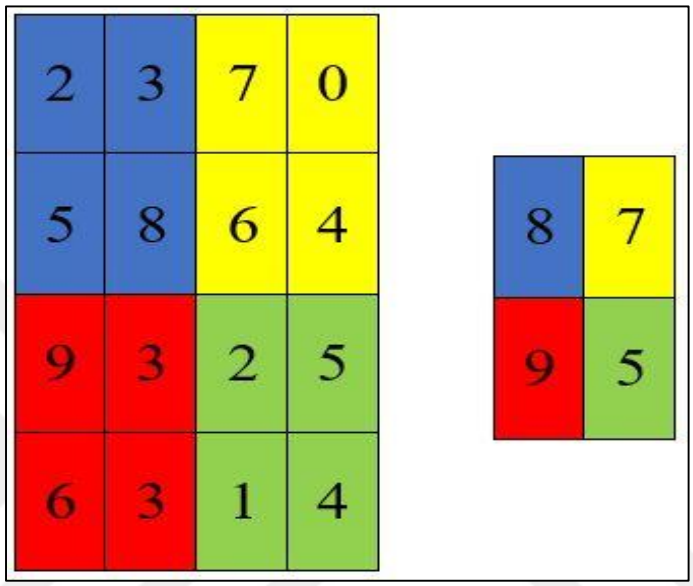
Burada S_i birleştirme işleminin sonucudur, Rj ise (c) öznitelik haritasında j birleştirmesidir, i ise bunun içinde yer alan her bir elemanın indisidir.

İki Tür Birleştirme vardır:

Max Pooling(Maksimum Birleştirme): Max pooling, öznitelik haritasının belirlenen her bir parçasında en yüksek, yani maksimum değeri alarak oluşturulur (Zeiler ve Fergus, 2013).

$$S_i = \max_{i \in R_j} a_i \tag{3.5}$$

Burada S_i birleştirme işleminin sonucudur, a_i filtre boyutudur, Rj ise (c) öznitelik haritasında birleştirmenin yapıldığı j alanıdır. Şekil 3.14'te 2×2'lik filtre, adım uzunluğu =2, ve dolgu=0 için Max Birleştirme işlemini göstermektedir.



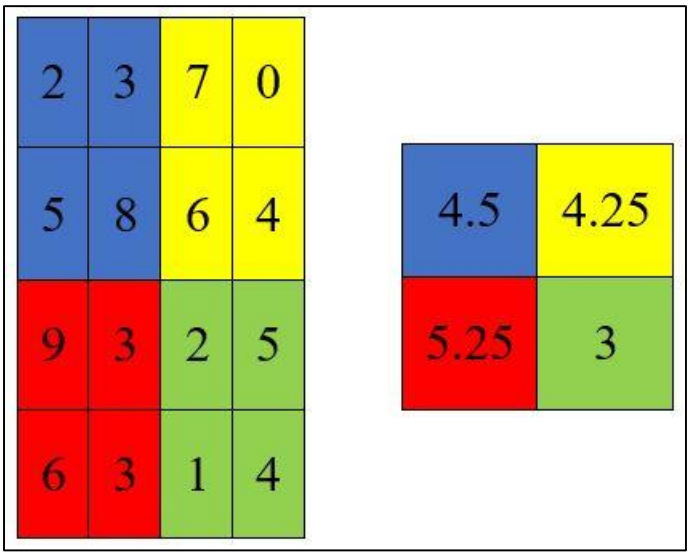
Şekil 3.14 CNN'de maksimum havuzlama işlemi

Her bir S_i için max pooling sonucu S_i = [(2,3,5,8) =8, (7,0,6,4) =7, (9,3,6,3) =9, (2,5,1,4) =5]

Average Pooling (Ortalama Birleştirme): Average Pooling işlemi, öznitelik haritasının belirlenen her bir parçasındaki, yani görüntünün çekirdek tarafından kaplanan kısmındaki ortalama değeri hesaplanarak gerçekleştirilir (Zeiler ve Fergus, 2013).

$$S_i = \frac{1}{[R_j]} \sum_{i \in R_j} a_i \tag{3.6}$$

Burada S_i birleştirme işleminin sonucudur, a_i filtre boyutudur, Rj ise (c) öznitelik haritasında birleştirmenin yapıldığı j alanıdır. Şekil 3.15'te 2×2 filtre, adım uzunluğu=2, ve dolgu=0 için bir Avarage Poling (Ortalama Birleştirme) örneği görülmektedir.



Şekil 3.15 CNN'de havuzlama ortalaması

$S_i = \left(\frac{1}{16}\right) \times 18 \times 4 = 4.5$ $S_i = \left(\frac{1}{16}\right) \times 17 \times 4 = 4.25$

$S_i = \left(\frac{1}{16}\right) \times 21 \times 4 = 5.25$ $S_i = \left(\frac{1}{16}\right) \times 12 \times 4 = 3$

Her bir S_i için Average pooling sonucu S_i = [(2,3,5,8) =4.5, (7,0,6,4) =4.25, (9,3,6,3) =5.25, (2,5,1,4) =3]

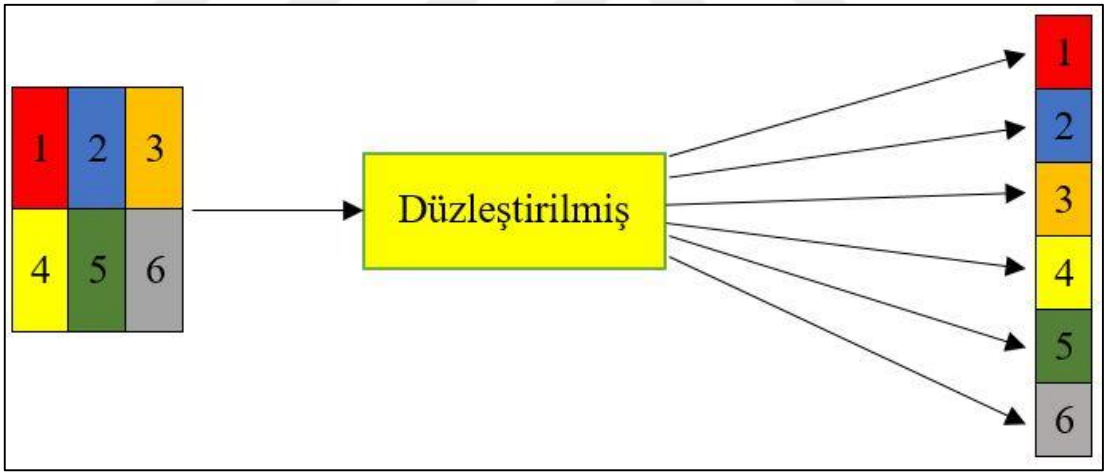
3.2.5 Tam Bağlantılı Katman

Bu adım, CNN ağının son aşamasıdır. Tam bağlantılı katman, geleneksel bir sinir ağının nöronları düzenleme şekline çok benzer. Bu nedenle, Şekil 3.16'da gösterildiği gibi, tam bağlantılı bir katmandaki her bir düğüm, hem önceki hem de sonraki katmandaki her düğüme doğrudan bağlıdır. Tam bağlantılı bir katmanın başlıca dezavantajı, eğitim örneklerinde karmaşık hesaplama gerektiren çeşitli parametrelere ihtiyaç duymasıdır (Albawi vd., 2017). Ancak yine de, geleneksel CNN modellerinde, tam bağlantılı (FC) katman yaygın olarak kullanılmaktadır (Zhang vd., 2017). Dahası; evrişimsel katmanın veya nihai birleştirmenin düzleştirilmiş çıktısı, daha sonra tam bağlantılı katmana girdi olarak iletilir. Burada; düzleştirilmişin (flattened) anlamı, birleştirmenin ve evrişimli sinir ağının son (veya herhangi bir) çıktısının yani 3 boyutlu bir matrisin, tüm değerlerinin bir vektöre dönüştürüleceğidir. Tam bağlantılı katman, geleneksel bir çok katmanlı algılayıcıdır,

ve çıkış katmanında bir softmax aktivasyon fonksiyonu kullanır. Dolayısıyla, çıktı matrisi elde edildikten ve düzleştirildikten sonra, çıktı sınıflarını tahmin etmek için softmax fonksiyonu kullanılır. Softmax fonksiyonunun ifadesi şu şekildedir:

$$f(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{N} ex_j} \tag{3.7}$$

burada x_i ifadesi i. çıktının boyutunu, N ise toplam boyut sayısını, ki genellikle sınıf sayısına eşittir, simgeler. Softmax'ın aktivasyon özelliği genellikle çıktı katmanına entegre edilir. Bu, genellikle çok sınıflı öğrenme problemlerinde yaygın olarak uygulanır. Softmax fonksiyonu, her bir grubun çıktılarını 0 ile 1 arasına sıkıştıracaktır. İdeal olarak, softmax fonksiyonu, her bir girdinin sınıfının belirlenmeye çalışıldığı sınıflandırıcı çıktı katmanında kullanılır. Çıktı matrisi elde edildikten ve bu matris düzleştirildikten sonra; softmax fonksiyonu, evrişimli sinir ağı uygulamasında tam bağlantılı katmanın son aşamasıdır (Li vd., 2018).



Şekil 3.16 Düzleştirme

Softmax fonksiyonuna sayısal bir örnek:

$$f(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{N} ex_j}$$

$x_{i=[1,\ 2,\ 3,\ 4,\ 5,\ 6]}$ $e^{x_{i=[2.72,\ 7.39,\ 20.08,\ 54.60,\ 148.41,\ 403.43]}}$

$$\sum_{j=1}^{N} ex_i = 2.72+7.39+20.08+ 54.60+ 148.41+403.43=636.63$$

$$f(x_1) = \frac{2.72}{636.63} = 0.004$$

$$f(x_2) = \frac{7.39}{636.63} = 0.011$$

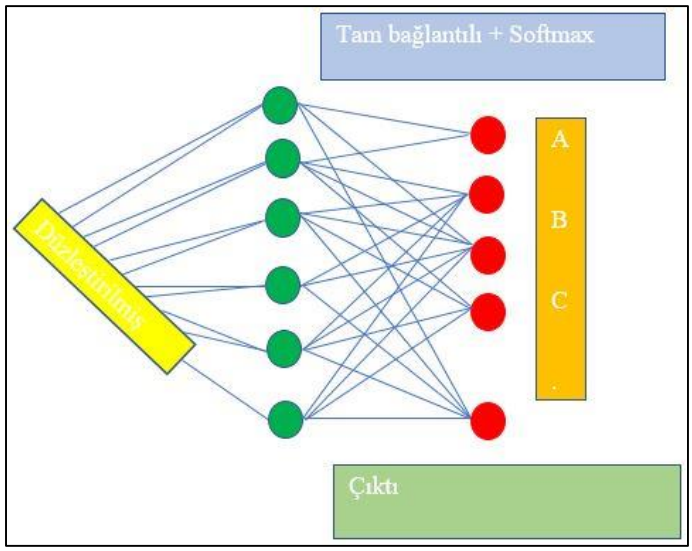
$$f(x_3) = \frac{20.08}{636.63} = 0.031$$

$$f(x_4) = \frac{54.60}{636.63} = 0.085$$

$$f(x_5) = \frac{148.41}{636.63} = 0.233$$

$$f(x_6) = \frac{403.43}{636.63} = 0.634$$

Tam bağlantılı katman şekil 3.17'de gösterilmektedir.



Şekil 3.17 Tam bağlantılı katman (Saha., 2018)

3.3 Görüntü Sınıflandırma

Görüntü sınıflandırması, çok spektrumlu istatistiksel karar kurallarının veya mantıksal uzamsal karar kurallarının uygulanmasına dayalı olarak pikselleri farklı alan kullanımı/alan örtüsü (LULC) sınıflarına ayırmak için bir yöntemdir. Ayrıca görüntü sınıflandırması, güvenilirliği temel olarak veri kümesi özellikleri, analiz edilen sorunun karmaşıklığı ve sınıflandırma algoritmasının sağlamlığı ile ilişkili karmaşık bir süreçtir (Sinha, 2016).

Sınıflandırma, makine öğreniminin birincil unsurlarındandır. Tüm makine öğrenimi uygulamalarında olduğu gibi amaç, ilgili bir çıkarım yapma algoritması ile bir araya getirildiğinde, yeni veya daha önce değerlendirilmemiş veriler hakkında bir çıkarım yapabilen, bir şeyler tahmin edebilen bir model geliştirmektir (Price vd., 2019).

3.4 Eğitim ve Test Amaçlı Ayrıştırma Yöntemi

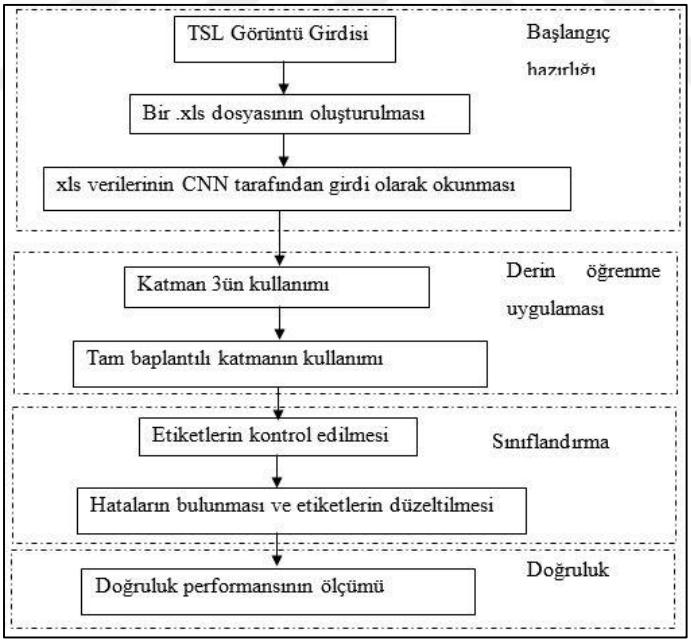
Veri kümesi belirlendikten sonra, eğitim ve test amaçlı iki alt kümeye ayrılır. %70'e %30'luk bir ayrıştırma makul görünmektedir (eğitim için %70).

Örneklemin yani veri kümesinin ikiye bölünmesi yaklaşımı, yüksek boyutlu çalışmalarda yaygın olarak kullanılan bir tasarımdır. Sınıflandırma doğruluğunu hesaplamanın bir yolu olarak, bu tasarım eldeki verileri bir eğitim kümesi ve bir test kümesi olarak ikiye ayırır. Eğitim kümesinden bir sınıflandırıcı geliştirilir, ve bu sınıflandırıcı test kümesindeki her bir örneğe tek tek uygulanır. İstatistiksel tahmin modelleri pratikte verileri bahsettiğimiz iki gruba ayrıştırmadan geliştirilir, yani model geliştirme için kullanılan verilerle tahmin doğruluğu için kullanılan veriler birbirinden ayrılmazlar (Considine vd., 2018; Kurian vd., 2017).

4. SİMÜLASYON SONUÇLARI

4.1 Deneysel Sonuçlar

Bu bölümde önerilen sistemin performansı incelenmektedir ve evrişimli sinir ağı yöntemi kullanıldığında Türk İşaret Dili için elde edilen görüntülerin ne derece doğru sınıflandırıldığı değerlendirilmektedir. Doğruluk hesaplamalarının yanı sıra, karışıklık matrisi, test verilerinin ve eğitim verilerinin sonuçlarını açıklığa kavuşturan bazı grafikler de hazırlanmıştır. Tüm bu veriler bir dizi aşama neticesinde ortaya çıkmıştır. İlk aşamada Türk İşaret Dilinin harfleri toplanmıştır. İkinci olarak, görüntüler CNN modeli kullanılarak sınıflandırılmıştır. Bundan sonra CNN modeli, harfleri ve sembolleri tanıyabilmesi ve tahmin edebilmesi amacıyla bir kısım girdi verisiyle eğitilmiştir. Akış şeması şekil 4.1 bu aşamaları gösterir. Önerilen yöntemin ana adımları şekil 4.1'de gösterilmektedir.

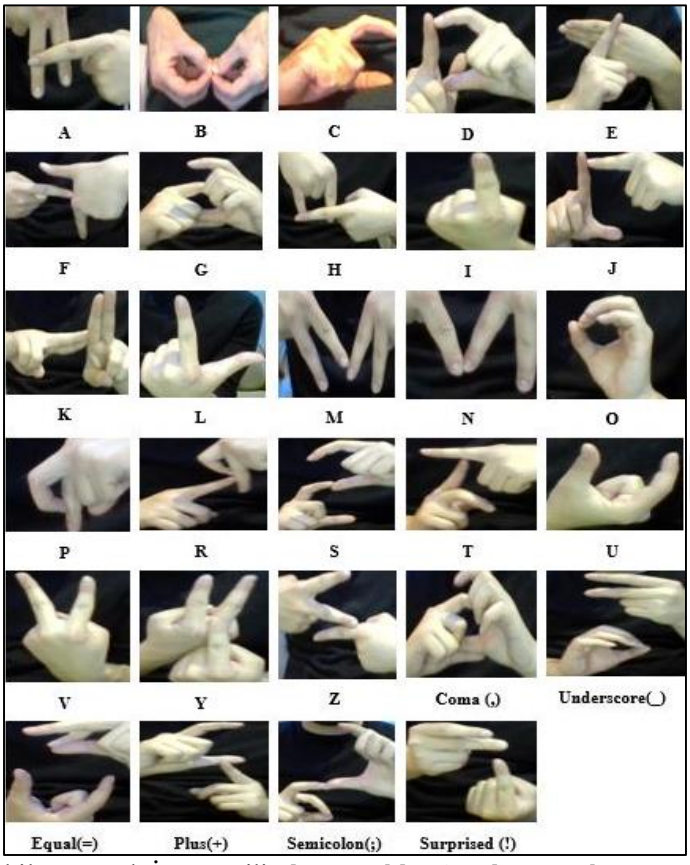


Şekil 4.1 Önerilen metodun ana adımları

4.2 Türk İşaret Dili İçin Veri Kümesi

Türk İşaret Diline ait görüntüler Kaggle web sitesinden alınmıştır (Ezel ve Baykan, 2017). Veri kümesi 2976 öğeyi içeriyordu ve toplam 29 sınıf vardı, her sınıfta 100-104 arası görüntü bulunmaktaydı. Veri kümesi eğitim için 2082 öğe ve test için 894

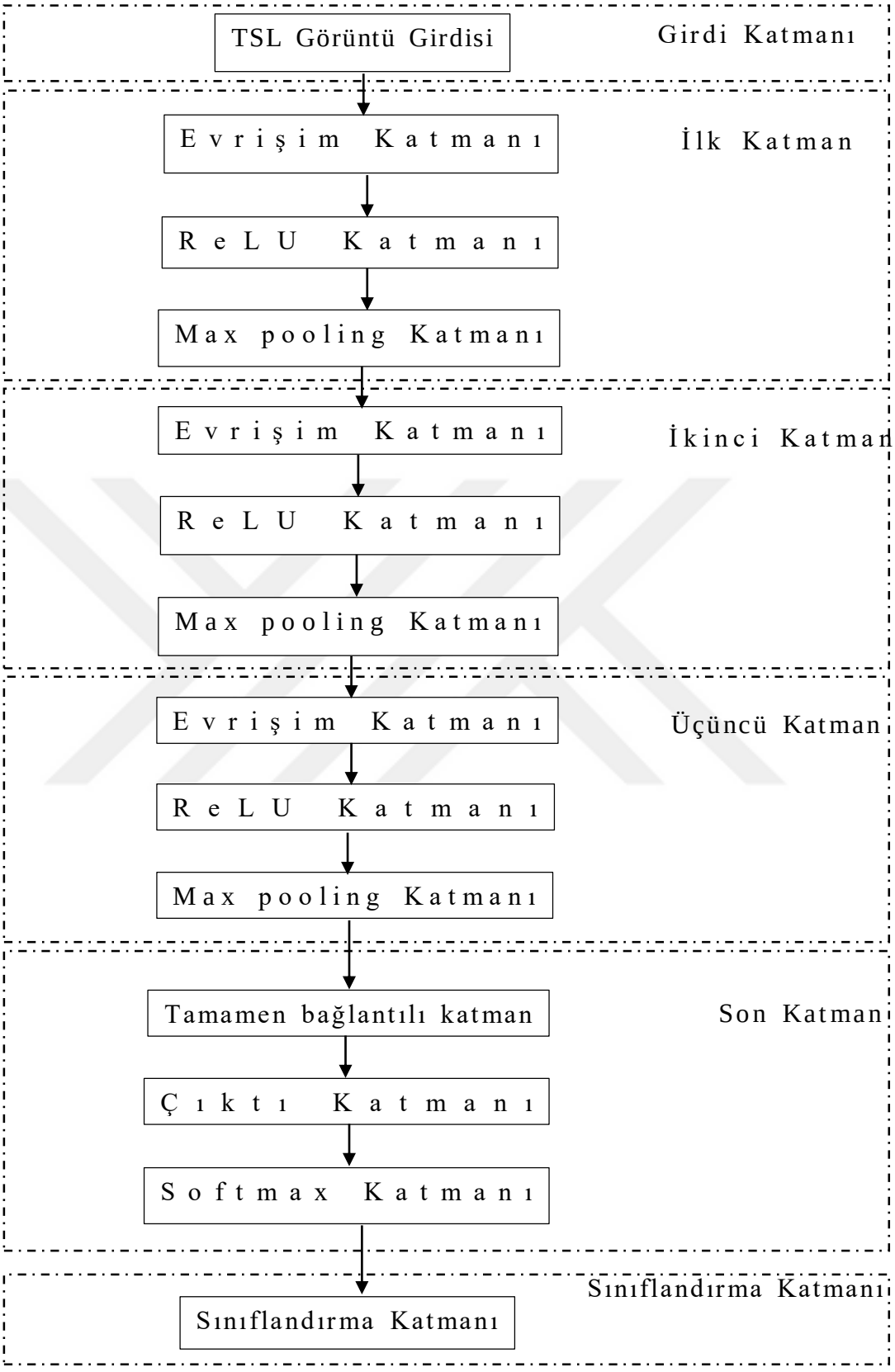
öğe içerecek şekilde ayrıştırılmıştır. İşaret diline ait veri kümeleri genellikle vücut hareketlerini içerir, bu hareketler parmaklarla, ellerle, kollarla, kafayla, yüz ifadesiyle veya vücudun tümden fiziksel hareketleriyle ifade edilen bir anlamı veya bir karşılığı olan hareketlerdir. Bu verilerin kullanılmasındaki amaç Türk İşaret Diline ait bu işaretlerin tanınmasıdır, bu işaretler alfabe harflerinden (A'dan Z'ye) veya [(Virgül ,)) (Alt çizgi _) (Eşit =) (Artı +) (Noktalı virgül ;) (Ünlem !)] gibi sembollerden oluşur. Şekil 4.2 Türk İşaret Dilinden bazı örnekleri göstermektedir.



Şekil 4.2 Türk İşaret Dilinden örnekler (Ezel ve Baykan, 2017)

4.3 İşin Akış Şeması

Önerilen yöntemin özeti şekil 4.3'te verilmektedir.



Şekil 4.3 Önerilen yöntemin özeti

4.4 Python Açıklaması

Python üst düzey programlama dillerinden biridir. Pythonla sayısal hesaplama, görüntü işleme, ağ programlama, web geliştirme işlemleri rahatlıkla yapılabilir. Python kullanıcı ve geliştiricileri topluluğu, programcıların makine öğrenmesi modellerini uygulamalarına yardımcı olan başlıca topluluklardan biridir (Li vd., 2018). Makine öğrenmesi (ML), bilgisayar yazılımlarına görevleri kendi başlarına nasıl gerçekleştireceklerini öğretir. Python dilinin başlıca özelliklerini şöyle listeleyebiliriz:

- Ücretsizdir.
- Windows, Mac veya Linux gibi yaygın işletim sistemlerinin tamamında bulunur.
- Python birçok platformda çalışabilir.
- Python dilinde yazılmış programların, çeşitli işletim sistemleri arasında çok az değişiklikle veya hiçbir değişiklik yapılmaksızın transferi mümkündür.
- Python topluluğu herhangi bir yazılımı kolayca öğrenebilir/öğretebilir, ayrıca bu dille programlama yeteneği daha hızlı kazanılabilir.
- Güçlü etkileşimli Python yorumlayıcısı, gerçek zamanlı kod ve canlı denemelerin oluşturulmasını sağlar, bu da zaman tasarrufu sağlar ve üretkenliği arttırır (Raschka, 2018).

4.4.1 Python Geliştirme Ortamları

Herhangi bir modern programlama dilinde olduğu gibi Python'un da, paketlere erişmek, saklamak ve çözüm üretmek için kendine has bir yolu vardır. Burada, Python sanal ortamlarının temel amacı, bünyesinde Python projeleri için yalıtılmış bir ortam yaratmaktır. Her bir projenin, diğer projelerin bağımlılıklarından bağımsız olarak kendi bağımlılıklarına sahip olması amaçlanır. Birçok farklı Python geliştirme ortamı veya derleyicisi arasında seçim yapmak mümkündür. Windows, Linux ve Unix gibi her bir işletim sisteminin otomatik olarak içerdiği, yerleşik bir yorumlayıcısı vardır. Bilim topluluğunun ihtiyaçlarına cevap verebilmek amacıyla

Python'da özel paketler oluşturulmuş durumdadır ve bu paketler Python'un belirli bilimsel modülleri üzerine inşa edilmiştir (Chityala ve Pudipeddi, 2014).

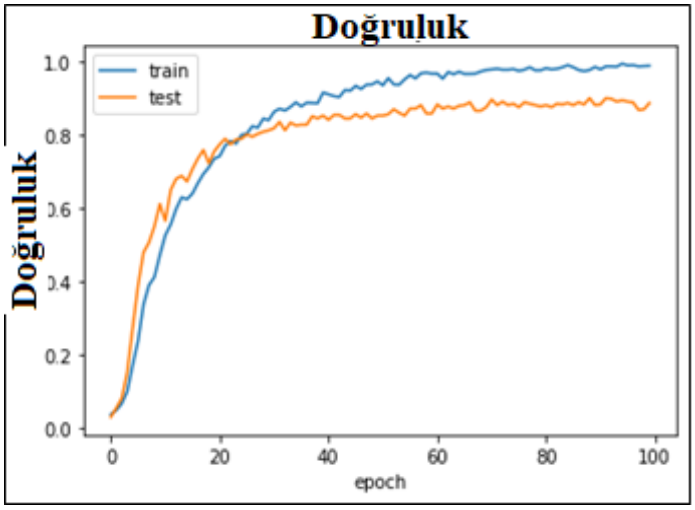
4.4.2 NumPy, SciPy ve Matplotlib Kütüphaneleri

Python dilinin görüntü işlemek için kullanılabilecek bir dizi kütüphanesi mevcuttur; örneğin NumPy, bir dizi kullanışlı dizilim nesneleriyle görüntüleri ve matrisleri temsil eder, ve doğrusal cebir işlemlerini yerine getirir, ayrıca görüntüleri sınıflandırmak için NumPy'in kullanılması mümkündür Solem (2012), Oliphant (2006). SciPy kütüphanesi ise bir dizi hızlı sayısal işlem koleksiyonu elde etmek için dizilimleri kullanır. Ancak çok yönlü olan Matplotlib kütüphanesi, Python kullanarak yüksek kaliteli grafikler elde etmek için muhtemelen en fazla özelliğe sahip kütüphanedir (Richert, 2013).

4.5 Simülasyon Sonuçları

Değerlendirme sonuçlarının en güvenilir şekilde elde edilebilmesi için döngü değeri (epoch) 100 olarak; minimum örneklem büyüklüğü (mini batch size) 5 olarak alınmıştır. El bölgesinin 28x28 piksel boyutundaki sadece siyah-beyaz ikili görüntülerinden oluşan veri kümesin 5 kat çapraz doğrulama yöntemi ile test edilmesiyle %82.75'lik bir başarı elde edilmiştir. Döngü (epoch) değeri 100 olarak alındığında, CPU'nun eğitimin her bir katı için yaklaşık bir buçuk saate ihtiyaç duyduğu görülmüştür.

Şekil 4.4'te, 2 kat için eğitim sürecinin grafiği gösterilmektedir. Burada mavi ve turuncu çizgiler sırasıyla eğitim ve test değerleri için doğrulama değerleridir; siyah çizgiler yani sırasıyla *y* ve *x* eksenleri ise testin doğruluk ve devir sayısını temsil eder.



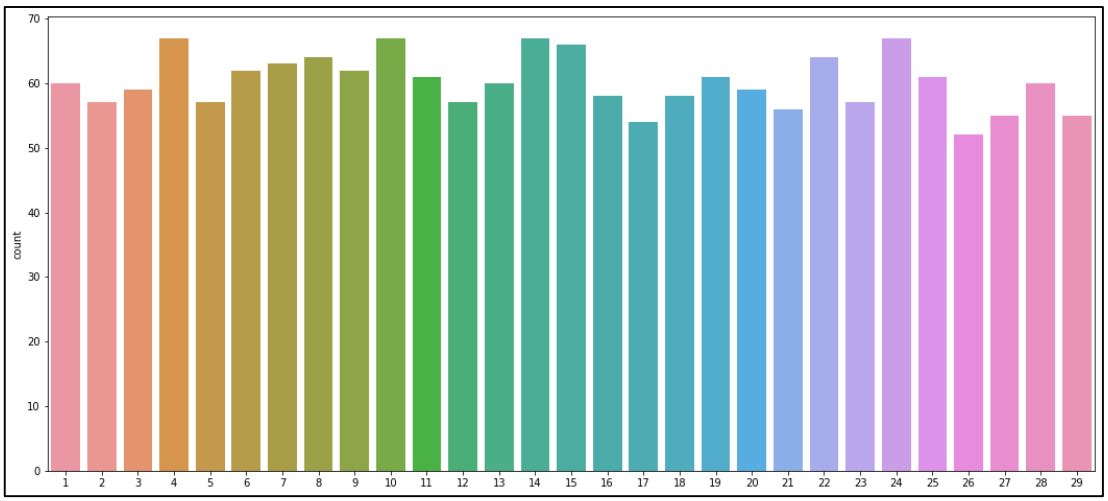
Şekil 4.4 CNN katmanının eğitim süreci

Birinci evrişim katmanı uygulanarak elde edilen veriler, hem pozitif hem de negatif aktivasyon sahalarını, yani ReLU aktivasyon fonksiyonunun uygulanmasını içerir.

CNN yapısının 4. katmanında elde edilen veriler 2x2 büyüklüğündeydi ve hem satır hem de sütunlarda maximum pooling yapıldı. Bu katman neticesinde, özniteliğin boyutu 28x28 piksele düşürüldü.

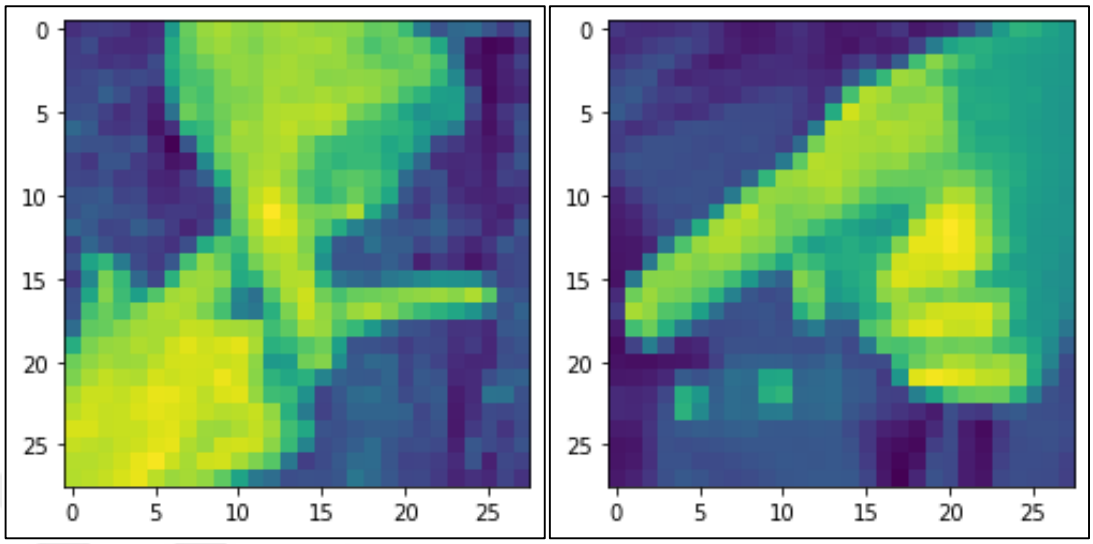
İkinci evrişim tabakasında 24 adet 4x4 filtre kullanıldı ve dolgulama işlemi, görüntülerin boyutunun korunmasını sağlamak için uygulandı.

Her bir harf için doğruluk performansı şekil 4.5'de gösterilmektedir.



Şekil 4.5 Her bir harf için doğruluk performansı

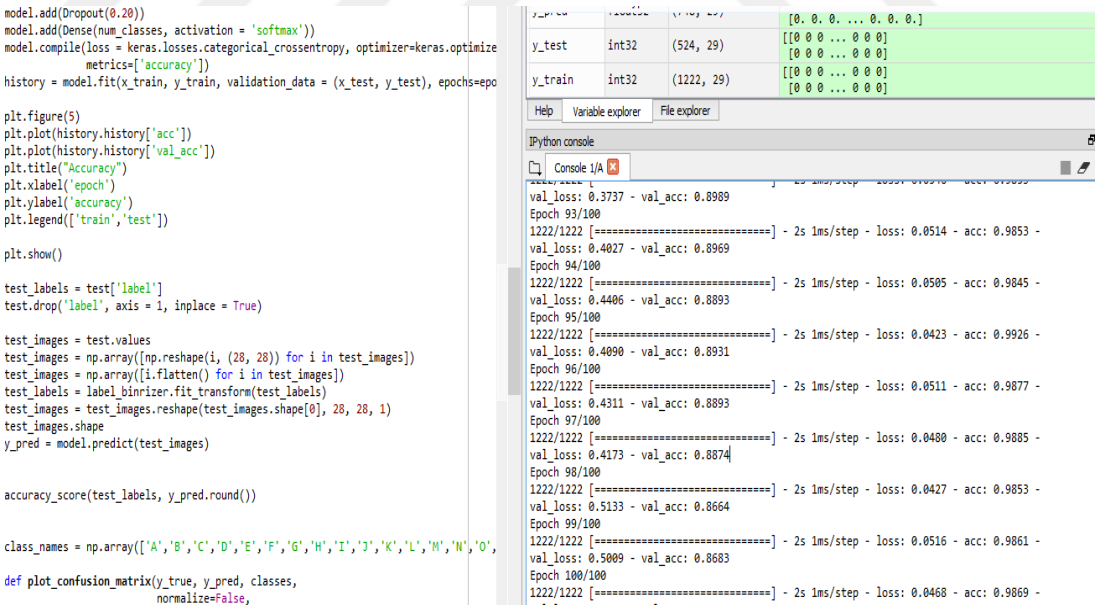
Python tarafından okunan bazı görüntüler şekil 4.6'da gösterilmektedir.



Şekil 4.6 Python tarafından okunan bazı görüntüler

Görüntüler 28x28 boyutlarındadır.

Bu tezde kullanılan Python geliştirme ortamı şekil 4.7'de gösterilmektedir.



Şekil 4.7 Çalışmamızda kullanılan Python geliştirme ortamı

4.6. Diğer Çalışmalarla Karşılaştırma

Çeşitli bilimsel makalelerde elde edilen sonuçlar Tablo 4.1'de özetlenmiştir. Tablo 4.2, araştırmalar için mevcut standart veri kümelerinden elde edilen sonuçları

göstermektedir. Benzer şekilde, içerik oluşturucuların kendi veri kümelerinden elde edilen sonuçlarsa Tablo 4.3'te özetlenmiştir. Sonuç, İşaret dili girdisi, Veri kümesinin boyutu, Eğitim kümesi, Test kümesi, standart veri kümesi/içerik oluşturucuların kendi veri kümesi, sınıflandırma yöntemleri ve son olarak tanıma oranı gibi parametreleri içerir Ong ve Ranganath (2005), Mitra ve Acharya (2007).

Tablo, sinir ağlarının ve HMM varyasyonlarının (Moni ve Ali, 2009) tanıma yüzdesi bağlamındaki popülariteleri nedeniyle araştırmacılar tarafından yaygın olarak kullanıldığını göstermektedir.

Tablo 4.1 Sonuçların Karşılaştırılması

Standart veri kümelerinden elde edilen sonuçlar								
Kaynak	İşaret dili	Kullanılan Veri kümesi			Kullanılan Standart veri kümesi	Sınıflandırma yöntemi	Tanıma oranı	
		Veri boyutu	Eğitim kümesi	Test kümesi			öznitelik	
							yok	var
(Kshirsagar ve Doye, 2010)	ASL	26 130	? ?	? ?	Lifeprint Fingerespell Library	Dynamic Time Wrapping	85.77	92.82
						Statik hareketler	82.97	87.64
						Dinamik hareketler		
(Rokade vd., 2009)	(Rezaei vd., 2008) ile aynı					(Rezaei vd., 2008) ile aynı		
(Rana vd., 2009)	Yüz verisi	150	üç farklı ışık seviyesi	1040	CAS-PEAL Database	Tanıma Yöntemi PCA MPCA-ML MPCA-LV MPCA-JS MPCA-PS PCA+LDA PCA+LPP	74.79 74.04(83.08) 75.10 92.79 87.79 89.89 88.94	
(Rana vd., 2009)	Yüz verisi	38	3040	9120	Extended Yale B	Tanıma Yöntemi PCA MPCA-ML MPCA-LV MPCA-JS MPCA-PS PCA+LDA PCA+LPP	67.48 75.95 (79.77) 49.89 84.20 85.03 85.03 84.30	

Tablo 4.1 devamı

(Karami vd., 2011)	Yüz verisi	28	140	140	Weizmann face database	Tanıma Yöntemi PCA MPCA-ML MPCA-LV MPCA-JS MPCA-PS PCA+LDA PCA+LPP	86.71 84.52 (86.31) 69.64 77.18 98.21 97.82 91.67
(Thangali vd., 2011)	ASL	419	?	?	Lexicon Video Verikümesi SignStream'den linguistik Ek açıklamalarla	El şekli Bayes ağı (HSBN) Ranked handshapes 1 5 10 15 20 25	HSBN yok / HSBN var 32.1 / 26.0 61.3 / 55.1 75.1 / 71.4 81.0 / 80.2 85.9 / 84.5 89.6 / 88.7
(Aran ve Akarun, 2010)	ASL				eNTERFACE		
(Athitsos vd., 2010)	ASL	193	?	710	Lexicon Video Verikümesi	BoostMap embedding	(33.1)iptidai hesaplamayla aynı, fakat 800 kat daha hızlı.
(Flasiński ve Myśliński, 2010)	PSL	48 48	240 144	? ?	Own Database PETS	ETPL(k) graph parsing model	94.31 85.40
(Dreuw ve Ney, 2008)	ASL	201	161	40	RWTH-BOSTON-104	Recogniser Output Voting Error Reduction (ROVER)	12.9 Word error Rate (WER)
Önerilen Yöntem	TSL	2974	2081	893	Türk İşaret Dili	Evrişimli Sinir Ağı	94.4%

Bu tabloda görüldüğü gibi, önerilen yöntem diğer yöntemlerden daha yüksek bir doğruluk oranı vermektedir. Bu araştırmada, öznitelik çıkarımı için evrişimli sinir ağını kullandık ve toplam 3 katman söz konusuydu. Önerilen yöntem ağın eğitiminde kullanılan verilerden öznitelikleri doğru bir şekilde buldu. Bu sonuç, söz konusu

eğitim verilerinin kullanılmasıyla test veri kümesinde %94.4'lük bir doğruluk oranı elde edildiğini göstermektedir.

Tablo 4.2 Araştırmacıların kendi veri kümelerinden alınan sonuçlar

Kaynak	işaret Dili	Kullanılan Veri Kümesi			Sınıflandırma Yöntemi	Tanıma oranı	
		Boyut	Eğitim	Test			
(Munib vd., 2007)	ASL	20	200	100	ANN(feed-forward BPN) Akıllı eşik yok Akıllı eşik var(0.15) Akıllı eşik var(0.25)	 77.72 91.53 92.33	
(Quan ve Jinye, 2008)(Peng ve Yulong, 2009)	CSL				SVM sınıflandırıcı	95.0256	
(Maebatake vd., 2008)	?	183	75%	25%	Gizli Markov Modeli El pozisyonu (0.0) ve hareket yok El pozisyonu (1.0) ve hareket yok El pozisyonu (0.5) ve hareket var (0.5) El pozisyonu (0.2) ve hareket var(0.8)	 49.3 70.2 70.6 75.6	
(Grobel ve Assan, 1997)	SLN	262	43	43	Gizli Markov Modeli Eğitim 1 Eğitim 2 Eğitim 3	Test1 98.8 86.6 98.3	Test 2 91.1 95.8 100
			150	150	Eğitim 1 Eğitim 2 Eğitim 3	93.7 58.5 93.2	64.4 90.7 92.5
			262	262	Eğitim 1 Eğitim 2 Eğitim 3	91.1 47.6 89.8	56.2 93.0 94.3
(Bui ve Nguyen, 2007)	VSL	23	?	?	Bulanık kural tabanlı sistem 'R', 'U' ve 'V' dışında bütün harfler 'R', 'U' ve 'V' İki eksenli MEMS hızlandırıyla(muğlaklık) Vietnamca heceleme kuralları uygulandıktan sonra	100 90, 79, ve 93 94, 90, ve 96	

Tablo 4.2 devamı

(Wang vd., 2010)		26	80%	20%	Bakış Açısı Bakış 1 Bakış 2 Tensör alt uzay analizi Bakış 3 Bakış 4 Bakış 5 Ortalama	Gri 76.9 73.1 100 92.3 92.3 86.9	İkili 69.2 80.8 92.3 92.3 88.5 84.6
(Davydov vd., 2010)	UKL	12 85	? ?	240	Gizli Markov Modeli Statik İşaretler P2DIDM Görüntü bozulması, çevreleyen alanın haç şeklinde olması Görüntü bozulması, çevreleyen alanın kare şeklinde olması Piksel-piksel Dinamik Hareketler	94 84 74 88 91.7	
(Mekala vd., 2011)	ASL	26	26	26	Bileşimli Sinir Ağları Gürültüye bağışıklık yok Gürültüye bağışıklık var	100 48	
(Pahlevanzadeh vd., 2007)	TSL	450	450	-	Genel Kosinüs Tanımlayıcısı (GCD) 3D Hopfield NN kullanarak GCD	Trained 96 -	Test 91 100
(Shanableh ve Assaleh, 2007)	ArSL	3450	2300	1150	KNN ve polinomsal ağlar	87	
(Maraqa vd., 2012)	ArSL	30 Gest.	900	300	Yapay Sinir Ağları Elman Ağı Tamamen Tekrarlayan Ağ	89.66 95.11	
(Vanjikumaran ve Balachandran, 2011)	SLTSL	300	?	?	Yapay Sinir Ağları Test Sonuçları Ünsüzler için sonuçlar Ünlüler için sonuçlar		73.76 74.72 71.50
(Sarkaleh vd., 2009)	PSL	8	160	80	Çok Katmanlı Algılayıcı Sinir Ağları Gizli nöron sayısı 10 11 12		 98.75 97.08 97.50

Tablo 4.2 devamı

(Zhou vd., 2011)	CSL	10	960	240	Hipotez Karşılaştırma Kılavuzlu Çapraz Onaylama (HC-CV)	88.5
(Karami vd., 2011)	PSL	20	416	224	Çok Katmanlı Algılayıcı Sinir Ağları	94.06
(Binh vd., 2005)	ASL	36	1080	?	Yarı iki-boyutlu gizli Markov modelleri (P2-DHMMs)	98
(Teng vd., 2005)	CSL	30	2475	1650	Yerel lineer gömme	92.2
(Kong ve Ranganath, 2008)	ASL	27	9072	3888	Fisher'in doğrusal diskriminantıyla doğrusal bir karar ağacı (FLD)	96.1
					Vektörel Nicemleme PCA (VQPCA)	
					Periodik Olmayan İşaretler	97.30
					Periodik İşaretler	97.00
					Toplam	86.80

5. SONUÇ VE TARTIŞMA

5.1 Sonuç

Bu tezde, Türk İşaret Dilinin parmak alfabesi hareketlerinin başarılı bir şekilde tanınmasını sağlamak için derin öğrenme kullanılmıştır. Son yıllarda, nesne tanıma ve görüntüdeki nesnelerin algılanması gibi çoğu uygulamanın başarısını artırmak için derin öğrenme yapısı tercih edilmektedir. Bu çalışmanın temel amacı, derin öğrenmenin Türk İşaret Dilinin parmak alfabesini tanımada ne kadar başarılı olabildiğini değerlendirmektir. Yapılan değerlendirmeler ışığında vardığımız sonuç; bu çalışmada kullanılan her iki derin öğrenme yapısının, özellikle de gerekli eğitimden sonra gösterdikleri sınıflandırma performansı göz önünde bulundurulduğunda, gerçek zamanlı olarak kullanılabileceğidir.

Türkiye'de son yıllarda Türk İşaret Dilini tanıma sistemleri üzerine çok sayıda çalışma yapılmış olsa da, parmak alfabesi üzerine çok az çalışma mevcuttur. Türk İşaret Dili hakkında kapsamlı bir çalışmanın ortaya konabilmesi için, kelime tabanlı işaretlerin yanı sıra harf tabanlı işaretlerin tanınması çalışmalara dahil edilmelidir. Parmak alfabesine özellikle özel isimlerin ve soneklerin ifade edilmesinde ihtiyaç duyulmaktadır. Bu çalışmada, Türk İşaret Dilinin parmak alfabesi hareketlerini doğru bir şekilde sınıflandırabilmek için farklı derin öğrenme yapıları kullanılmıştır. Ayrıca bir öznitelik çıkarımı yapmaksızın gerekli özniteliklerin öğrenilmesi derin öğrenme yöntemi ile mümkün olmuştur. Derin öğrenme yapıları dışında, YSA ve SVM sınıflandırma yöntemlerinin Türk İşaret Dilinin parmak alfabesi hareketlerini tanıma başarısı da kıyaslanmıştır. Bu mukayeseler neticesinde, derin öğrenme yapıları ile birlikte veri geliştirme yöntemlerinin uygulanması sonucu elde edilen veri kümelerinin kullanılmasının daha başarılı sonuçlar verdiği görülmüştür. Bu çalışmanın sonucu olarak elde edilen çıkarımlar, bu konuda gelecekte yapılacak çalışmalara ışık tutacaktır.

Bu tezde geliştirilen sistemin gerçek zamanlı olarak etkili ve doğru bir şekilde çalışabilmesi için, ortamın ışık koşullarının iyi ayarlanması zorunluluğu bulunmaktadır. Ortamdaki aşırı ışık, el bölgesinin ayrıştırılmasını zorlaştırmakta ve

sınıflandırma sürecinin yanlış neticeler vermesine neden olmaktadır. Türk İşaret Dilinin parmak alfabesindeki bazı harfler arasındaki mevcut benzeşme, sistemin başarısını olumsuz etkilemektedir. Fakat; Türk İşaret Diliyle ilişkili sistemler üzerine az sayıda çalışma yapılmış olması, sistemin çalışabilmesi için ekstra donanıma ihtiyaç duyulmaması ve bu nedenle sistemin düşük maliyetli olması, renkli eldiven gibi araçlara gerek kalmaksızın pratik tanımanın sistemce yapılabilmesi, el bölgesinden kaynaklanan sorunları en aza indirmek için gerekli çalışmaların yapmış olması, sınıflandırma yöntemi olarak derin öğrenme yapılarını kullanıyor olması, önceden herhangi bir öznitelik çıkarma yöntemi uygulamaksızın eğitim verilerini doğrudan sınıflandırabilmesi gibi hususlar çalışmamızı orijinal yapan faktörler olarak sıralanabilir.

5.2 Gelecek Çalışmalar

Gelecekte yapılacak çalışmaların şu sıralayacağımız konulara odaklanması olası görünmektedir: önerilen sistemin, hem harf hem de kelime bazında çalışabilecek şekilde geliştirilmesi ve gerçek zamanlı uygulamalarda kullanılması; ortam koşullarından kaynaklanan sistem kısıtlamalarının kısmen veya tamamen ortadan kaldırılması; sistem tarafından sadece el hareketlerinin değil, aynı zamanda yüz ifadesinin ve dudakların durumunun da algılanması. Böylesi çalışmaların yapılması işitme sorunu olan bireyler için bir umut ışığı olacaktır ve bu kişilerin topluma entegrasyonuna katkıda bulunacaktır.

KAYNAKLAR

Abreu, J. G., Teixeira, J. M., Figueiredo, L. S. & Teichrieb, V., (2016). "Evaluating Sign language recognition using the myo armband," in *2016 XVIII Symposium on Virtual and Augmented Reality (SVR)*, pp. 64–70.

Albawi, S., Mohammed, T. A. & Al-Zawi, S., (2017). "Understanding of a convolutional neural network," in *2017 International Conference on Engineering and Technology (ICET)*, pp. 1–6.

Antunes, D. R., Guimaraes, C., García, L. S., Oliveira, L. E. S. & Fernandes, S., (2011). "A framework to support development of Sign language human-computer interaction: Building tools for effective information access and inclusion of the deaf," in *2011 Fifth International Conference on Research Challenges in Information Science*, pp. 1–12.

Athitsos, V., Wang, H. & Stefan, A., (2010). "A database-based framework for gesture recognition," *Pers. Ubiquitous Comput.*, vol. 14, no. 6, pp. 511–526.

Bantupalli, K. & Xie, Y., (2019). "American İşaret dili Recognition Using Machine Learning and Computer Vision,".

Bheda, V. & Radpour, D., (2017) "Using deep convolutional networks for gesture recognition in American Sign language," *arXiv Prepr. arXiv1710.06836*.

Binh, N. D., Shuichi, E. & Ejima, T., (2005). "Real-time hand tracking and gesture recognition system," *Proc. GVIP*, pp. 19–21.

Bui, T. D. & Nguyen, L. T., (2007). "Recognizing postures in Vietnamese Sign language with MEMS accelerometers," *IEEE Sens. J.*, vol. 7, no. 5, pp. 707–712.

Chityala, R. & Pudipeddi, S., (2014). *Image Processing and acquisition using Python*. Chapman and Hall/CRC.

Considine, E. C., Thomas, G., Boulesteix, A. L., Khashan, A. S. & Kenny, L. C., (2018). "Critical review of reporting of the data analysis step in metabolomics," *Metabolomics*, vol. 14, no. 1, p. 7.

Darwish, S. M., Madbouly, M. M. & Khorsheed, M. B., (2016). "Hand gesture recognition for Sign language: a new higher order fuzzy HMM approach," *Hand*, vol. 1, p. 18565.

Davydov, M. V., Nikolski, I. V. & Pasichnyk, V. V., (2010). "Real-time Ukrainian Sign language recognition system," in *2010 IEEE International Conference on Intelligent Computing and Intelligent Systems*, vol. 1, pp. 875–879.

Dreuw, P. & Ney, H., (2008). "Visual modeling and feature adaptation in Sign language recognition," in *ITG Conference on Voice Communication [8. ITG-Fachtagung]*, pp. 1–4.

Ezel, E. & Baykan, Ö. K., (2017). "Vision-Based Türk İşaret Dili Recognition Using Convolutional Neural Networks," in *International Conference on Theoretical and Applied Computer Science and Engineering 2017 (ICTACSE, 2017)*.

Flasiński, M. & Myśliński, S., (2010). "On the use of graph parsing for recognition of isolated hand postures of Polish Sign language," *Pattern Recognit.*, vol. 43, no. 6, pp. 2249–2264.

Forster, J., Schmidt, C., Koller, O., Bellgardt, M. & Ney, H., (2014). "Extensions of the İşaret dili Recognition and Translation Corpus RWTH-PHOENIX-Weather.," in *LREC*, pp. 1911–1916.

Garg, A., (2012). "Converting American İşaret dili To Voice Using RBFNN." *Sciences*.

Grobel, K. & Assan, M., (1997). "Isolated Sign language recognition using hidden Markov models," in *1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation*, vol. 1, pp. 162–167.

Hisham, B. & Hamouda, A., (2018). "Arabic Sign language Recognition using Microsoft Kinect and Leap Motion Controller," *Available SSRN 3389799*.

Jinye, P., (2008). "Application of improved Sign language recognition and synthesis technology in IB," in *2008 3rd IEEE Conference on Industrial Electronics and Applications*, pp. 1629–1634.

Karami, A., Zanj, B. & Sarkaleh, A. K., (2011). "Persian Sign language (PSL) recognition using wavelet transform and neural networks," *Expert Syst. Appl.*, vol. 38, no. 3, pp. 2661–2667.

Kausar, S., Javed, M. Y. & Sohail, S., (2008). "Recognition of gestures in Pakistani Sign language using fuzzy classifier," in *Proceedings of the 8th conference on Signal processing, computational geometry and artificial vision*, pp. 101–105.

Kim, Y., (2014). "Convolutional neural networks for sentence classification," *arXiv Prepr. arXiv1408.5882*.

Kong, W. W. & Ranganath, S., (2008). "Signing exact english (SEE): Modeling and recognition," *Pattern Recognit.*, vol. 41, no. 5, pp. 1638–1652.

Krizhevsky, A., Sutskever, I. & Hinton, G. E., (2012). "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105.

Kshirsagar, K. P. & Doye, D., (2010). "Object Based Key Frame Selection for Hand Gesture Recognition," in *2010 International Conference on Advances in Recent Technologies in Communication and Computing*, pp. 181–185.

Kumarage, D., Fernando, S., Fernando, P., Madushanka, D. & Samarasinghe, R., (2011). "Real-time Sign language gesture recognition using still-image comparison & motion recognition," in *2011 6th International Conference on Industrial and Information Systems*, pp. 169–174.

Kurian S. M. vd., (2017). "Biomarker guidelines for high-dimensional genomic studies in transplantation: adding method to the madness," *Transplantation*, vol. 101, no. 3, pp. 457–463.

Li, O., Liu, H., Chen, C. & Rudin, C., (2018). "Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions," in *Thirty-Second AAAI Conference on Artificial Intelligence*.

Li, W. J., Hsieh, C. Y., Lin, L. F. & Chu, W. C., (2017). "Hand gesture recognition for post-stroke rehabilitation using leap motion," in *2017 International Conference on Applied System Innovation (ICASI)*, pp. 386–388.

Maebatake, M., Suzuki, I., Nishida, M., Horiuchi, Y. & Kuroiwa, S. (2008). "Sign language recognition based on position and movement using multi-stream HMM," in *2008 Second International Symposium on Universal Communication*, pp. 478–481.

Maraqa, M., Al-Zboun, F., Dhyabat, M. & Zitar, R. A., (2012). "Recognition of Arabic Sign language (ArSL) using recurrent neural networks," *J. Intell. Learn. Syst. Appl.*, vol. 4, no. 01, p. 41.

Matsugu, M., Mori, K., Mitari, Y. & Kaneda, Y., (2003). "Subject independent facial expression recognition with robust face detection using a convolutional neural network," *Neural Networks*, vol. 16, no. 5–6, pp. 555–559.

Mehdi, S. A. & Khan, Y. N., (2002). "İşaret dili recognition using sensor gloves," in *Proceedings of the 9th International Conference on Neural Information Processing, 2002. ICONIP'02*, vol. 5, pp. 2204–2206.

Mekala, P., Gao, Y., Fan, J. & Davari, A., (2011). "Real-time Sign language recognition based on neural network architecture," in *2011 IEEE 43rd Southeastern Symposium on System Theory*, pp. 195–199.

Mitra, S. & Acharya, T., (2007). "Gesture recognition: A survey," *IEEE Trans. Syst. Man, Cybern. Part C (Applications Rev.*, vol. 37, no. 3, pp. 311–324.

Moni, M. A. & Ali, A. B. M. S., (2009). "HMM based hand gesture recognition: A review on techniques and approaches," in *2009 2nd IEEE International Conference on Computer Science and Information Technology*, pp. 433–437.

Montiel, E., Aguado, A. S. & Nixon, M. S., (2000). "On resolving ambiguities in arbitrary-shape extraction by the hough transform," in *Proc. of the British Machine Vision Conference (BMVC'00)*.

Munib, Q., Habeeb, M., Takruri, B. & Al-Malik, H. A., (2007) "American Sign language (ASL) recognition based on Hough transform and neural networks," *Expert Syst. Appl.*, vol. 32, no. 1, pp. 24–37.

Nadgeri, S. M., Sawarkar, S. D. & Gawande, A. D., (2010). "Hand gesture recognition using CAMSHIFT algorithm," in *2010 3rd International Conference on Emerging Trends in Engineering and Technology*, pp. 37–41.

Naidoo, S., Omlin, C. W. & Glaser, M., (1998). "Vision-based static hand gesture recognition using support vector machines," *Univ. West. Cape, Bellv*.

Oliphant, T. E., (2006). *A guide to NumPy*, vol. 1. Trelgol Publishing USA.

Ong, S. C. W. & Ranganath, S., (2005). "Automatic Sign language analysis: A survey and the future beyond lexical meaning," *IEEE Trans. Pattern Anal. Mach. Intell.*, no. 6, pp. 873–891.

Pahlevanzadeh, M., Vafadoost, M. & Shahnazi, M., (2007). "February 2007. Sign language recognition," in *9th International Symposium on Signal Processing and Its Applications*, pp. 1–4.

Peng, J. & Yulong, L., (2009). "Chinese Sign language recognition based on gray-level co-occurrence matrix and other multi-features fusion," in *2009 4th IEEE Conference on Industrial Electronics and Applications*, pp. 1569–1572.

Pigou, L., Dieleman, S., Kindermans, P. J. & Schrauwen, B., (2014). “Sign language recognition using convolutional neural networks,” in *European Conference on Computer Vision*, pp. 572–578.

Price, M., Hoover, J., Dagley, G., Wylie, S. & Tang, Q., (2019). “Object detection using image classification models.” Google Patents, 05-Mar.

Quan, Y. & Jinye, P. (2008). “Chinese Sign language recognition for a vision-based multi-features classifier,” in *2008 International Symposium on Computer Science and Computational Technology*, vol. 2, pp. 194–197.

Raheja, J. L., Mishra, A. & Chaudhary, A., (2016). “Indian Sign language recognition using SVM,” *Pattern Recognit. Image Anal.*, vol. 26, no. 2, pp. 434–441.

Rahman, M. H. & Afrin, J., (2013). “Hand gesture recognition using multiclass support vector machine,” in *International Journal of Computer Applications*, vol. 74, no. 1, published by Foundation of Computer Science, pp. 39–43.

Rana, S., Liu, W., Lazarescu, M. & Venkatesh, S., (2009). “A unified tensor framework for face recognition,” *Pattern Recognit.*, vol. 42, no. 11, pp. 2850–2862.

Raschka, S., (2018). “MLxtend: Providing machine learning and data science utilities and extensions to Python’s scientific computing stack.,” *J. Open Source Softw.*, vol. 3, no. 24, p. 638.

Rezaei, A., Vafadoost, M., Rezaei, S. & Daliri, A., (2008). “3D pose estimation via elliptical fourier descriptors for deformable hand representations,” in *2008 2nd International Conference on Bioinformatics and Biomedical Engineering*, pp. 1871–1875.

Rezaei, A., Vafadoost, M., Rezaei, S. & Shekofteh, Y., (2008). “A feature based method for tracking the 3-D trajectory and the orientation of a signer’s hand,” in *2008 International Conference on Computer and Communication Engineering*, pp. 346–351.

Richert, W., (2013). *Building machine learning systems with Python*. Packt Publishing Ltd.

Rokade, U. S., Doye, D. & Kokare, M., (2009). “Hand gesture recognition using object based key frame selection,” in *2009 International Conference on Digital Image Processing*, pp. 288–291.

Saha, S., (2018). “A comprehensive guide to convolutional neural networks–the ELI5 way.” Medium.

Sahoo, A. K., Mishra, G. S. & Ravulakollu, K. K., (2014) “Sign language recognition: state of the art,” *ARPN J. Eng. Appl. Sci.*, vol. 9, no. 2, pp. 116–134.

Sarkaleh, A. K., Poorahangaryan, F., Zanj, B. & Karami, A., (2009). “A neural network based system for persian Sign language recognition,” in *2009 IEEE International Conference on Signal and Image Processing Applications*, pp. 145–149.

Scherer, D., Müller, A. & Behnke, S., (2010). “Evaluation of pooling operations in convolutional architectures for object recognition,” in *International conference on artificial neural networks*, pp. 92–101.

Shanableh T. & Assaleh, K., (2007). “Arabic Sign language recognition in user-independent mode,” in *2007 International Conference on Intelligent and Advanced Systems*, pp. 597–600.

Sharrma, A., Khandelwal, A., Kaur, K., Joshi, S., Upadhyay, R. & Prabhu, S., (2017). “Vision based static hand gesture recognition techniques,” in *2017 International Conference on Communication and Signal Processing (ICCSP)*, pp. 705–709.

Sinha, A., (2016). “Hadoop in the Cloud to Analyze Climate Datasets,” in *Cloud Computing in Ocean and Atmospheric Sciences*, Elsevier, pp. 245–276.

Solem, J. E., (2012). Programming Computer Vision with Python: Tools and algorithms for analyzing images. “ O’Reilly Media, Inc.,”.

Swetha, S. J. & Mary, S. R., (2019). “Tamil İşaret dili Recognition using Machine Learning to Aid Deaf and Dumb People,”.

Teng, X., Wu, B., Yu, W. & Liu, C., (2005). “A hand gesture recognition system based on local linear embedding,” *J. Vis. Lang. Comput.*, vol. 16, no. 5, pp. 442–454.

Thangali, A., Nash, J. P., Sclaroff, S. & Neidle, C., (2011). “Exploiting phonological constraints for handshape inference in ASL video,” in *CVPR 2011*, pp. 521–528.

Tsechpenakis, G., Metaxas, D. & Neidle, C., (2006). “Learning-based dynamic coupling of discrete and continuous trackers,” *Comput. Vis. Image Underst.*, vol. 104, no. 2–3, pp. 140–156.

Tubaiz, N., Shanableh, T. & Assaleh, K., (2015). "Glove-based continuous Arabic Sign language recognition in user-dependent mode," *IEEE Trans. Human-Machine Syst.*, vol. 45, no. 4, pp. 526–533.

Vanjikumaran, S. & Balachandran, G., (2011). "An automated vision based recognition system for Sri Lankan Tamil Sign language finger spelling," in *2011 International Conference on Advances in ICT for Emerging Regions (ICTer)*, pp. 39–44.

Wang, S. J., Zhang, D. C., Jia, C. C., Zhang, N., Zhou, C. G. & Zhang, L. B., (2010). "A Sign language recognition based on tensor," in *2010 Second International Conference on Multimedia and Information Technology*, vol. 2, pp. 192–195.

Yim, J., Ju, J., Jung, H. & Kim, J., (2015). "Image classification using convolutional neural networks with multi-stage feature," in *Robot Intelligence Technology and Applications 3*, Springer, pp. 587–594.

Zeiler, M. D. & Fergus, R., (2013). "Stochastic pooling for regularization of deep convolutional neural networks," *arXiv Prepr. arXiv1301.3557*.

Zhang, C. L., Luo, J. H., Wei, X. S. & Wu, J., (2017). "In defense of fully connected layers in visual representation transfer," in *Pacific Rim Conference on Multimedia*, pp. 807–817.

Zhou, Y., Yang, X., Lin, W., Xu, Y. & Xu, L., (2011). "Hypothesis comparison guided cross validation for unsupervised signer adaptation," in *2011 IEEE International Conference on Multimedia and Expo*, pp. 1–4.

ÖZGEÇMİŞ



Adı Soyadı	: Khaled Mohamed Abubaker ELBAYOUDI
Doğum Yeri ve Yılı	: 19.03.1987
Medeni Hali	: Evli
Yabancı Dili	: İngilizce ve başlangıç seviyesinde Türkçe
E-posta	: albauodi@gmail.com
Mesleği	: Bilgisayar Mühendisi

Eitim Durumu

Lise	: Ibn Ghalboun Lisesi, 2006
Lisans	: Misurata Üniversitesi, 2011

Yayın Listesi :

O.S.S. Alsharif1, K.M. Elbayoudi1, A.A.S. Aldrawi1, K. Akyol. Evaluation of Different Machine Learning Methods for Caesarean Data Classification. International Journal of Information Engineering and Electronic Business(IJIEEB). no. 2, cilit. 5, 2019.