

**ANKARA ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**YÜKSEK LİSANS TEZİ**

**DERİN ÖĞRENME KULLANARAK İŞARET DİLİ RAKAM TANIMA**

**Hacer GÜLER**

**ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI**

**ANKARA  
2021**

**Her hakkı saklıdır**

## ÖZET

Yüksek Lisans Tezi

### DERİN ÖĞRENME KULLANARAK İŞARET DİLİ RAKAM TANIMA

Hacer GÜLER

Ankara Üniversitesi  
Fen Bilimleri Enstitüsü  
Elektrik Elektronik Mühendisliği Anabilim Dalı

Danışman: Prof. Dr. Murat Hüsnü SAZLI

El hareketleri insan yaşamının birçok alanında büyük rol oynamaktadır. İnsanlar iletişim esnasında sıklıkla el hareketlerinden yararlanır. Temelde el hareketlerinden oluşan işaret dili ise dünya nüfusunun %5'inden fazlasını oluşturan işitme engelli bireyler arasındaki tek iletişim aracıdır. İşaret dili yüz ifadesi, göz, kaş, dil ve dudak hareketlerinin beraberindeki el ve kol hareketleriyle oluşan bir tür görsel dildir. İşaret dili rakam tanıma problemi ise işaret dilinin temelini oluşturur. Farklı ülkeler ve hatta zaman zaman bir ülkenin içindeki bölgelerin bile kendi işaret dili vardır. Ancak birkaç istisnai durum dışında işaret dillerinde rakamlar aynı el hareketleri ile aktarılmaktadır. Bu tez çalışması kapsamında farklı veri setlerindeki görüntülerin bir araya getirilip ön işleme tekniklerinin uygulanmasıyla bu problem için yeni bir veri setinin oluşturulması ve derin öğrenme mimarilerinin kullanılmasıyla işaret dili rakamlarının sınıflandırılması amaçlanmıştır. Bu amaç doğrultusunda 19653 görüntü içeren ve daha önce herhangi bir çalışmada kullanılmayan bir veri seti oluşturulmuştur. Ardından önerilen modellerden 3 farklı evrişimli sinir ağı modeli sunulmuştur. Eğitilen ağların test aşaması gerçekleştirilerek performans analizi yapılmıştır ve modellerin başarımları oranları kıyaslanmıştır. Nihai olarak oluşturulan modelde eğitim veri setinde %100, test veri setinde ise %99,3 sınıflandırma doğrulama başarımları elde edilmiştir.

**Haziran 2021, 70 Sayfa**

**Anahtar Kelimeler:** Derin öğrenme, işaret dili tanıma, el işaretleri tanıma, işaret dili rakam tanıma, evrişimli sinir ağı.

## ABSTRACT

MSc. Thesis

### SIGN LANGUAGE DIGIT RECOGNITION USING DEEP LEARNING

Hacer GÜLER

Ankara University  
Graduate School of Natural and Applied Sciences  
Electrical and Electronics Engineering

Supervisor: Prof. Dr. Murat Hüsnü SAZLI

Hand gestures play a big role in many areas of human life. People often make use of gestures during communication with the others. Sign language, which basically consists of hand gestures, is the only way of communication between hearing impaired individuals who consist more than 5% of the world population. Sign language is a kind of visual language formed by hand and arm movements accompanying facial expression, eye, eyebrow, tongue and lip movements. The problem of recognition of digits in sign language forms the basis of sign language. Different countries, and sometimes even different regions within a country, have their own sign language. However, except for a few exceptions, numbers in sign languages are transferred with the same hand gestures. Within the scope of this thesis, it was aimed to create a new data set for this problem by combining the images in different data sets and applying pre-processing techniques and to classify the sign language numbers by using deep learning architectures. For this purpose, a data set containing 19653 images and not previously used in any study was created. Then, 3 different convolutional neural network models from the proposed models are presented. Performance analysis was made by performing the test phase of the trained networks and the models were compared. In the final model, 100% classification verification success was achieved in the training data set and 99.3% in the test data set.

**June 2021, 70 pages**

**Key Words:** Deep learning, sign language recognition, hand gesture recognition, sign language digit recognition, convolutional neural network.

## ÖNSÖZ ve TEŞEKKÜR

Yüksek lisans tez çalışmam süresince kıymetli bilgi, birikim ve tecrübesini esirgemeyen değerli danışman hocam Prof. Dr. Murat Hüsnu SAZLI'ya teşekkürü bir borç biliyor ve şükranlarımı sunuyorum. Çalışma süresince bilgi, öneri ve yardımlarıyla beni yönlendiren ve bana zaman ayıran Haluk TANRIKULU'ya teşekkür ederim.

Bu süreçte beni destekleyen, her koşulda destekçim olan başta abim Mustafa GÜLER, ablam Eda ORAN, babam Erdoğan GÜLER ve yeğenim Eymen ORAN olmak üzere sevgili aileme, değerli arkadaşlarıma ve yol arkadaşım Cenk MÜLAZIMOĞLU'na gönülden teşekkür ederim.

Hacer GÜLER  
Ankara, Haziran 2021

## İÇİNDEKİLER

### TEZ ONAY SAYFASI

ETİK.....	i
ÖZET.....	ii
ABSTRACT .....	iii
ÖNSÖZ ve TEŞEKKÜR.....	iv
KISALTMALAR DİZİNİ .....	vii
ŞEKİLLER DİZİNİ .....	viii
ÇİZELGELER DİZİNİ .....	x
1. GİRİŞ .....	1
2. KURAMSAL TEMELLER.....	3
2.1 Yapay Sinir Ağları .....	4
2.1.1 Aktivasyon fonksiyonu .....	5
2.1.2 Yapay sinir ağının yapısı .....	8
2.2 Derin Öğrenme .....	9
2.2.1 İleri doğru yayılma.....	10
2.2.2 Kayıp fonksiyonu .....	11
2.2.3 Geriye doğru yayılma .....	12
2.2.4 Optimizasyon .....	12
2.2.5 Derin öğrenme tekniklerinin iyileştirilmesi .....	13
2.2.6 Derin öğrenme mimarileri.....	15
2.3 Evrişimli Sinir Ağları.....	16
2.3.1 Evrişim katmanı .....	18
2.3.2 Havuzlama katmanı .....	23
2.3.3 Tam bağımlı katman.....	23
2.4 İşaret Dili .....	24
3. LİTERATÜR TARAMASI .....	25
4. MATERYAL VE YÖNTEM.....	31
4.1 Veri Seti.....	32
4.1.1 Veri Seti – 1.....	32
4.1.2 Veri Seti – 2.....	35
4.1.3 Veri Seti – 3.....	37
4.1.4 Veri Seti – 4.....	39
4.1.5 Veri Seti – 5.....	41

<b>4.1.6 Veri Seti – 6.....</b>	<b>43</b>
<b>4.1.7 Tez veri seti.....</b>	<b>46</b>
<b>4.2 Yazılım ve Donanım.....</b>	<b>46</b>
<b>4.3 Model Yapısı.....</b>	<b>48</b>
<b>4.4 Model Mimarileri.....</b>	<b>49</b>
<b>4.4.1 Model 1.....</b>	<b>49</b>
<b>4.4.2 Model 2.....</b>	<b>51</b>
<b>4.4.3 Model 3.....</b>	<b>53</b>
<b>5. ARAŞTIRMA BULGULARI.....</b>	<b>56</b>
<b>6. SONUÇ VE DEĞERLENDİRME.....</b>	<b>64</b>
<b>KAYNAKLAR.....</b>	<b>67</b>
<b>ÖZGEÇMİŞ.....</b>	<b>70</b>

## KISALTMALAR DİZİNİ

### Kısaltmalar

CNN	Evrişimli Sinir Ağı
CPU	Merkezi İşlem Birimi
GPU	Grafik İşlem Birimi
GUI	Grafiksel Kullanıcı Arayüzü
HCI	İnsan Bilgisayar Etkileşimi
IDE	Tümleşik Geliştirme Ortamı
LSTM	Uzun Kısa Süreli Hafıza Ağları
MSE	Ortalama Kare Hatası
ReLU	Düzeltilmiş Doğrusal Birim
RGB	Kırmızı-Yeşil-Mavi
RNN	Tekrarlayan Sinir Ağı

## ŞEKİLLER DİZİNİ

Şekil 2.1 Sinir hücresi(nöron) yapısı.....	4
Şekil 2.2 k ile etiketlenen bir nöronun doğrusal olmayan modeli.....	5
Şekil 2.3 Sigmoid fonksiyonu.....	6
Şekil 2.4 Hiperbolik tanjant fonksiyonu.....	7
Şekil 2.5 ReLU fonksiyonu.....	7
Şekil 2.6 İki katmanlı yapay sinir ağı.....	8
Şekil 2.7 Ivakhnenko A. tarafından eğitilen bilinen ilk derin ağ mimarisi.....	10
Şekil 2.8 Yetersiz uyum ve aşırı uyum durumlarının gösterimi.....	13
Şekil 2.9 Geleneksel sinir ağı ile seyreltme uygulanmış sinir ağının karşılaştırılması...14	
Şekil 2.10 Evrişimli sinir ağının genel yapısı.....	17
Şekil 2.11 Evrişim işlemi için girdi görüntüsü ve çekirdek.....	19
Şekil 2.12 Evrişim işleminin ilk adımı.....	19
Şekil 2.13 Evrişim işleminin ikinci adımı.....	20
Şekil 2.14 Girdi görüntüsü ve öz nitelik haritası.....	20
Şekil 2.15 Adım uzunluğunun iki olduğu durumda çekirdeğin gezdirilmesi.....	21
Şekil 2.16 Sıfır doldurma yapılan girdi görüntüsü.....	22
Şekil 2.17 Maksimum havuzlama.....	23
Şekil 4.1 Veri seti – 1’deki 0 sınıfına ait bir örnek.....	33
Şekil 4.2 Veri seti – 1’deki her bir sınıfa ait beş örnek görüntü.....	34
Şekil 4.3 Veri seti – 2’deki her bir sınıfa ait beş örnek görüntü.....	36
Şekil 4.4 Veri seti – 2’de ön işleme teknikleri uygulanmış örnek görüntüler.....	37
Şekil 4.4 Veri seti – 3’teki her bir sınıfa ait dört örnek görüntü.....	38
Şekil 4.5 Veri seti – 3’te ön işleme teknikleri uygulanmış örnek görüntüler.....	39
Şekil 4.6 Veri seti – 4’teki her sınıfa ait bir örnek görüntü.....	40



Şekil 4.7 Veri seti – 4’te ön işleme teknikleri uygulanmış görüntü örnekleri.....	41
Şekil 4.8 Veri seti – 5’teki her bir sınıfa ait bir örnek görüntü.....	42
Şekil 4.9 Veri seti – 5’te ön işleme teknikleri uygulanmış görüntü örnekleri.....	43
Şekil 4.10 Veri seti – 6’daki her bir sınıfa ait beş örnek görüntü.....	44
Şekil 4.11 Veri seti – 11’de ön işleme teknikleri uygulanmış görüntü örnekleri.....	45
Şekil 4.12 Evrişimli sinir ağı Model 1.....	49
Şekil 4.13 Evrişimli sinir ağı Model 2.....	51
Şekil 4.14 Evrişimli sinir ağı Model 3.....	53
Şekil 5.1 Model 1 için doğruluk değerleri değişimi.....	57
Şekil 5.2 Model 1 için hata matrisi.....	58
Şekil 5.3 Model 2 için doğruluk değerleri değişimi.....	60
Şekil 5.4 Model 2 için hata matrisi.....	60
Şekil 5.5 Model 3 için doğruluk değerleri değişimi.....	62
Şekil 5.6 Model 3 için hata matrisi.....	62

## ÇİZELGELER DİZİNİ

Çizelge 3.1 MVGG-5 ve MVGG-9 mimari konfigürasyonları.....	25
Çizelge 3.2 Çalışmada önerilen mimari konfigürasyonu.....	26
Çizelge 3.3 Çalışmada önerilen SCNN konfigürasyonu.....	27
Çizelge 3.4 Çalışmada önerilen MCNN konfigürasyonu.....	28
Çizelge 3.5 Çalışmada önerilen TCNN konfigürasyonu.....	28
Çizelge 3.6 Çalışmada önerilen CNN modeli konfigürasyonu.....	29
Çizelge 4.1 Veri Seti – 1’deki sınıflara göre örnek sayısı.....	33
Çizelge 4.2 Veri Seti – 2’deki sınıflara göre örnek sayısı.....	35
Çizelge 4.3 Veri Seti – 3’teki sınıflara göre örnek sayısı.....	38
Çizelge 4.4 Veri Seti – 4’teki sınıflara göre örnek sayısı.....	40
Çizelge 4.5 Veri Seti – 5’teki sınıflara göre örnek sayısı.....	41
Çizelge 4.6 Veri Seti – 6’daki sınıflara göre örnek sayısı.....	44
Çizelge 4.7 Oluşturulan veri setinde sınıflara göre örnek sayısı.....	46
Çizelge 4.8 Donanım konfigürasyonu.....	47
Çizelge 4.9 Yazılım konfigürasyonu.....	47
Çizelge 4.10 Veri setindeki sınıflara karşılık etiket vektörleri.....	48
Çizelge 4.11 Evrişimli sinir ağı Model 1 konfigürasyonu.....	50
Çizelge 4.12 Evrişimli sinir ağı Model 1 özeti.....	51
Çizelge 4.13 Evrişimli sinir ağı Model 2 konfigürasyonu.....	52
Çizelge 4.14 Evrişimli sinir ağı Model 2 özeti.....	53
Çizelge 4.15 Evrişimli sinir ağı Model 3 konfigürasyonu.....	54
Çizelge 4.16 Evrişimli sinir ağı Model 3 özeti.....	55
Çizelge 5.1 Model 1 için doğruluk başarımı.....	57

Çizelge 5.2 Model 1 için kesinlik, duyarlılık ve F1 skor değerleri.....	59
Çizelge 5.3 Model 2 için doğruluk başarımı.....	59
Çizelge 5.4 Model 2 için kesinlik, duyarlılık ve F1 skor değerleri.....	61
Çizelge 5.5 Model 3 için doğruluk başarımı.....	61
Çizelge 5.6 Model 3 için kesinlik, duyarlılık ve F1 skor değerleri.....	63
Çizelge 6.1 Modellerin literatürdeki çalışmalarla karşılaştırılması.....	65
Çizelge 6.2 Tez veri setinin önerilen diğer çalışmalardaki modellerde kullanılması ile çıkan başarımlar oranları .....	66



## 1. GİRİŞ

El hareketleri, günlük yaşamımızda insanlar arasındaki bilgi aktarımının büyük bir bölümünü oluşturan iletişim kanalıdır. El hareketleri insanların birbirleriyle iletişim kurduklarında doğal etkileşim yoludur ve bu nedenle el hareketleri sözel olmayan bir iletişim şekli olarak değerlendirilebilir. El hareketi tanıma işlemi ise insan eli ile tanımlanan hareketleri anlama ve sınıflandırma işlemidir.

El hareketi tanıma sisteminin kullanım alanlarından biri işaret dili tanıma uygulamalarıdır. Çoğu insan herhangi bir işaret dilini anlamamakta ve genel olarak işitme engelli insanlar hakkında çok az şey bilmektedir. Dünyadaki milyonlarca işitme engelli insanların varlığı çevirmen olmadan işaret dilini metne dönüştüren işaret dili tanıma sistemlerinin geliştirilmesinin önemini göstermektedir. Aynı zamanda el hareketi tanıma probleminin çözümü kimlik tanıma uygulamaları için de oldukça önemlidir. Günümüzde evimize, ofisimize girerken belirlenen bir şifrenin girilmesi gerekmektedir. Ancak Covid-19 salgınının olduğu bu günlerde herhangi bir yere temas etmek sağlıklı bir durum değildir. Rakamların temassız bir şekilde tanındığı sistemler günümüz probleminde oldukça faydalıdır.

İşaret dili, evrensel olduğunu varsayan genel görüşün aksine yereldir. Her işaret dili alfabesindeki harfler farklı bir şekilde tanımlanmaktadır. Bazı ülkelerin işaret dili alfabetesi tek el ile statik tüm harfleri tanımlarken bazı ülkelerde bu harfler iki el kullanılarak veya dinamik bir şekilde tanımlanmaktadır. Ancak Hint İşaret Dili, Arap İşaret Dili gibi birkaç istisnai durum dışındaki işaret dillerindeki rakamlar aynı statik el hareketleriyle aktarılmaktadır. Tez çalışması kapsamında çoğu işaret dili için evrensel olan işaret dili rakamlarının derin öğrenme kullanarak sınıflandırılması amaçlanmaktadır.

İşaret dili tanıma için sistem mimarileri, girdilerine göre iki ana sınıfa ayrılabilir. Birincisi girişini sensör aracılığıyla eldivenlerden alan, eldiven verisi tabanlı mimaridir. Bu yaklaşımın avantajı yüksek doğruluğa sahip olması iken zayıf tarafı kullanan kişinin hareket alanını sınırlamasıdır. İkincisi ise girişini bir kamera ile sağlayan görme temelli

mimaridir. Bu yaklaşımın avantajı eldiven verisi tabanlı yaklaşıma göre daha ucuz ve daha az sınırlayıcı olması iken zayıf tarafı daha fazla hesaplama gücü tüketmesidir. Bu tez çalışmasında görme temelli yaklaşım kullanılmaktadır.

İşaret dili veya el hareketi tanıma problemi için çalışmalarda genellikle tek bir senaryoda bir veya birden fazla kişi tarafından gerçekleştirilen görüntüleri içeren veri setleri kullanılmaktadır. Bu görüntüler genellikle aynı arka planda, aynı ışıklandırma ile el konumunun kameraya olan uzaklığı aynı olacak şekilde oluşturulmaktadır. Tez çalışması kapsamında farklı veri setlerinden toplanan görüntülerin bir araya getirilmesi ile hem çeşitliliği hem de veri miktarının artırılması, işaret dili tanıma problemi için yeni, herhangi bir çalışmada kullanılmayan bir veri setinin oluşturulması amaçlanmaktadır.

Tez çalışması kapsamında oluşturulan 10 sınıflı veri setinin derin öğrenme yöntemi ile sınıflandırılması amaçlanmaktadır. Derin öğrenme, karmaşık problemler için gözlem yaparak, analiz ederek, öğrenerek ve karar vererek insan beynini taklit eder. Derin öğrenmenin en önemli özelliği, öğrenmenin verinin kendisinden yapılmasıdır. Probleme uygun öz nitelikler elle çıkarılmak yerine etkili algoritmalar kullanılarak çıkarılmaktadır.

Çalışmada derin öğrenme mimarileri içinde en çok kullanılan sinir ağı olan evrişimli sinir ağı kullanılmaktadır. İşaret dili rakam tanıma problemi için farklı evrişimli sinir ağı modelleri analiz edilmekte ve önerilen modellerden analiz çıktılarını en iyi temsil eden 3 model tez çalışmasında sunulmaktadır.

Tez çalışmasında izlenen yöntemler hakkında kavramsal bilgiler ve işaret dili hakkında genel bilgiler Bölüm 2’de, literatürdeki çalışmalardan bazıları Bölüm 3’te, çalışma kapsamında oluşturulan veri seti ve önerilen evrişimli sinir ağı modelleri Bölüm 4’te, önerilen modellerin analiz edilmesi Bölüm 5’te, çalışma sonuçlarının yorumlanması ve incelenen çalışmalarla karşılaştırılması Bölüm 6’da verilmiştir.

## 2. KURAMSAL TEMELLER

Teknolojinin gelişmesiyle makinelerin insan gibi düşünebilmesi fikri ortaya atılmıştır. Yapay zekâ, bir bilgisayarın veya bilgisayar kontrolündeki bir robotun çeşitli faaliyetleri zeki canlılara benzer şekilde yerine getirme kabiliyetidir (Copeland 2015). McCulloch ve Pitts 1943 yılında sinir ağları fikrini ortaya koymuşlardır. Hebb 1949 yılında ilk kendi kendine öğrenme kuralını öne sürmüştür. 1958 yılında Rosenblatt tarafından denetimli öğrenmenin ilk modeli olarak algılayıcı (perceptron) önerilmiştir (Haykin 2009). Birçok problemin çözümünde kullanılan yapay zekaya dayalı çalışmalar donanımsal yetersizlikten dolayı durma noktasına gelmiştir. GPU (Graphics Processing Unit) ve diğer donanımsal gelişmeler sayesinde 2000'li yıllarda yapay sinir ağlarının kullanımı tekrar ön plana çıkmıştır (Şeker 2017).

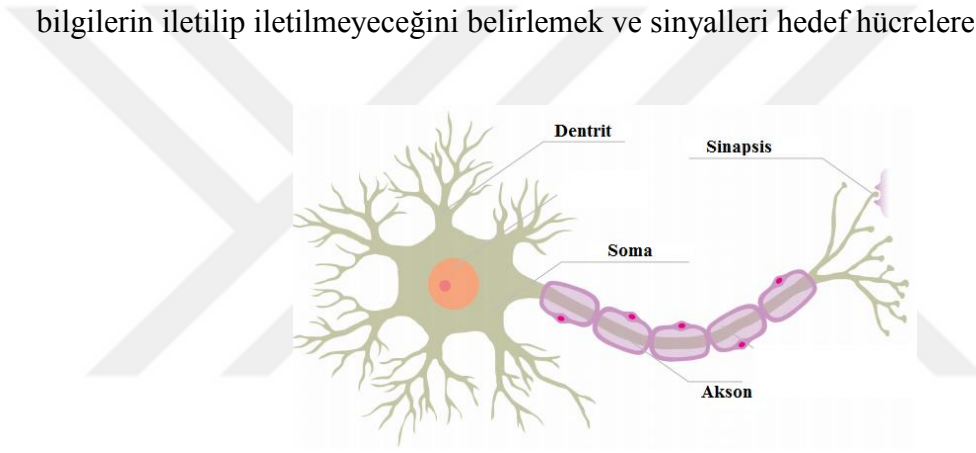
Derin öğrenme, yapay zekânın bir alt dalı olan makine öğrenmesinin özel bir halidir ve en popüler yaklaşımdır. Derin öğrenme, karmaşık problemler için gözlem yaparak, analiz ederek, öğrenerek ve karar vererek insan beynini taklit eder. Büyük miktardaki verileri alarak denetimli veya denetimsiz olarak öz nitelik çıkarma, dönüştürme ve sınıflandırma yeteneklerine sahiptir (Kayaalp ve Süzen 2018). Derin öğrenmenin en önemli özelliği, öğrenmenin verinin kendisinden yapılmasıdır. Probleme uygun öznitelikler elle çıkarılmak yerine etkili algoritmalar kullanılarak çıkarılmaktadır. Yani eski algoritmalar insana bağımlı çalışmaktayken, derin öğrenme algoritmaları kendi kendine öğrenebilmektedir (Kızrak ve Bolat 2018).

Derin öğrenme, birçok uygulama alanına sahiptir. Doğal dil işleme, görüntü ve video işleme alanlarında oldukça fazla kullanılmaktadır ve yüksek başarımlar elde edilmektedir. Sık kullanılan uygulama alanları dışında ilaç endüstrisi için ilaçların olumsuz sonuçların tespit edilmesi, uyku kalitesi tahmini, müşteri ilişkileri yönetimi, metin filtreleme, araç otonom sistemleri için multimedya yönetimi, akıllı trafik planlaması derin öğrenme mimarilerinin kullanıldığı problemlerdir. (Şeker 2017)

## 2.1 Yapay Sinir Ağları

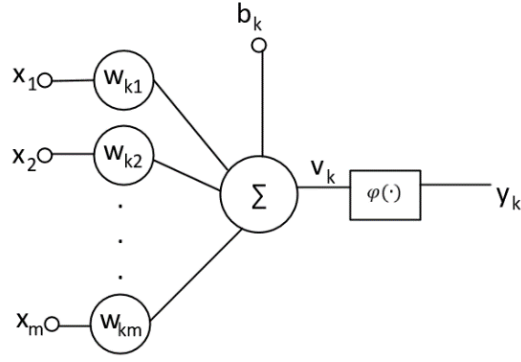
Yapay sinir ağları adından da anlaşılacağı gibi biyolojik beyin ve sinir sisteminden ilham alır (Sazlı 2006). Beyin oldukça karmaşık, doğrusal olmayan ve paralel çalışan bir bilgisayar gibidir ve nöron olarak bilinen sinir hücrelerini, bugün var olan en hızlı bilgisayarlardan oldukça hızlı kullanma yeteneğine sahiptir (Haykin 2009).

Bilim insanları, nöronların davranışlarını modelleyerek yapay sinir ağlarını geliştirmişlerdir. Sinir hücresinin basit bir modeli **şekil 2.1** ile gösterilmiştir. Nöronlar üç temel fonksiyonu gerçekleştirmektedir. Bunlar sinyalleri yani bilgiyi almak, bilgilerin iletilip iletilmeyeceğini belirlemek ve sinyalleri hedef hücelere iletmektir.



Şekil 2.1 Sinir hücresi(nöron) yapısı (Fountas 2011)

Yapay sinir ağlarının temelini oluşturan nöron modeli **şekil 2.2** ile gösterilmiştir. Burada sinir ağının üç temel unsuru belirlenebilir. Her biri kendi ağırlığı ile karakterize edilen bir dizi sinaps ilk unsurdur. Burada  $k$  nöronuna bağlı olan bir  $j$  sinapsının girişindeki  $x_j$  sinyali, sinaptik ağırlık olan  $w_{kj}$  ile çarpılır. İkinci unsur ise ağırlıklandırılan giriş sinyallerinin toplanmasıdır. Son olarak bir nöron çıktısının genliğinin sınırlandırılması için aktivasyon fonksiyonunun kullanılmasıdır. **Şekil 2.2** ile verilen nöron modeli aynı zamanda  $b_k$  ile gösterilen yanlılık ile aktivasyon fonksiyon girdisini, pozitif veya negatif olmasına bağlı olarak artırılmasını veya azaltılmasını sağlar (Haykin 2009).



Şekil 2.2 k ile etiketlenen bir nöronun doğrusal olmayan modeli

Şekil 2.2 ile gösterilen  $k$  nöronu için aktivasyon fonksiyonu girdisi matematiksel olarak Eşitlik (2.1), nöron çıkışı ise Eşitlik (2.2) ile verilir.

$$v_k = \sum_{j=1}^m w_{kj} x_k + b_k \quad (2.1)$$

$$y_k = \varphi(v_k) \quad (2.2)$$

Burada  $x_1, x_2, \dots, x_m$  giriş sinyalleri,  $w_{k1}, w_{k2}, \dots, w_{km}$  sinaptik ağırlıklar,  $b_k$  yanlılık,  $\varphi(\cdot)$  aktivasyon fonksiyonu ve  $y_k$  nöronun çıkış sinyalidir.

### 2.1.1 Aktivasyon fonksiyonu

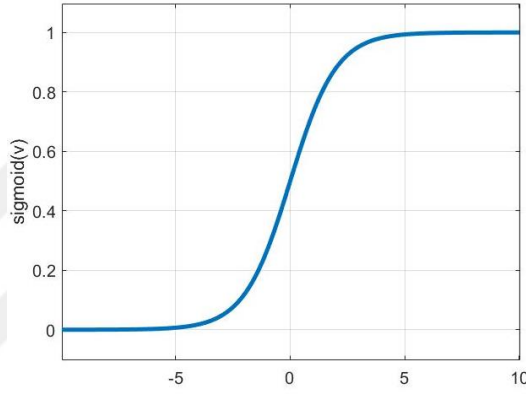
Aktivasyon fonksiyonları sinir ağlarının çıkışını oluşturan matematiksel eşitliklerdir. Bir yapay sinir ağında aktivasyon fonksiyonu kullanılmadığı durumda çıkış sinyali doğrusal bir fonksiyon sonucunda oluşturulur. Ancak gerçek hayatta görüntü, video, ses ve yazı gibi karmaşık girdiler kullanıldığından doğrusal olmayan durumlarda da sinir ağının öğrenme yapabilmesi gerekmektedir. Aktivasyon fonksiyonunun kullanımı ile doğrusal olmayan bir yapı elde edilir. Ayrıca aktivasyon fonksiyonlarının seçimi, yapay sinir ağlarının performansı için önemli bir role sahiptir. Çok katmanlı algılayıcılar gibi bazı modeller bu fonksiyonun türevinin alınabilir bir fonksiyon olmasını gerektirmektedir. Bu nedenle kullanılacak aktivasyon fonksiyonunun avantaj ve dezavantajları değerlendirilmelidir. Bu bölümde yalnızca en fazla kullanılan aktivasyon fonksiyonlarının tanımları verilmektedir.



Sinir ađlarında sıklıkla kullanılan doğrusal olmayan aktivasyon fonksiyonlarından biri Sigmoid fonksiyonudur ve Eşitlik (2.3) ile verilir.

$$\varphi(v) = \frac{1}{1+e^{-v}} \quad (2.3)$$

Sigmoid fonksiyonu **şekil 2.3** ile gösterilmiştir (Goodfellow vd. 2016). Gerçek bir sayı olarak 0 ile 1 arasında bir değere dönüştürür. Bu nedenle çıkış olarak bir olasılık değeri elde edilen modellerde kullanılır. Aynı zamanda bu fonksiyon süreklidir ve türevi alınabilir.

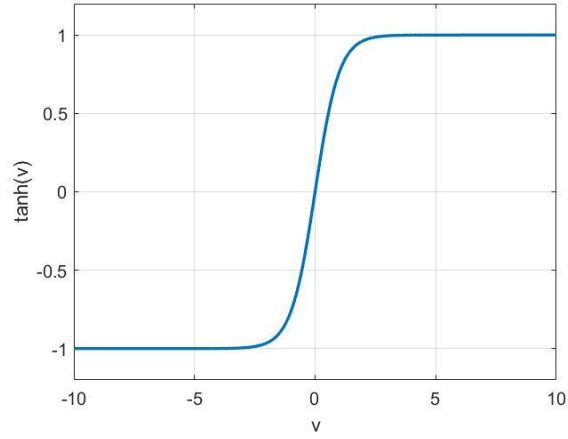


Şekil 2.3 Sigmoid fonksiyonu

En çok tercih edilen bir diğer aktivasyon fonksiyonu hiperbolik tanjant fonksiyonudur ve Eşitlik (2.4) ile verilir.

$$\varphi(v) = \frac{e^v - e^{-v}}{e^v + e^{-v}} \quad (2.4)$$

Hiperbolik tanjant fonksiyonu **şekil 2.4** ile gösterilmiştir. Sürekli ve türevi alınabilir olan hiperbolik tanjant fonksiyonu değerleri -1 ile 1 arasındadır (Sharma 2017). Bu fonksiyon genellikle iki sınıf arasındaki sınıflandırma problemlerinde kullanılır.

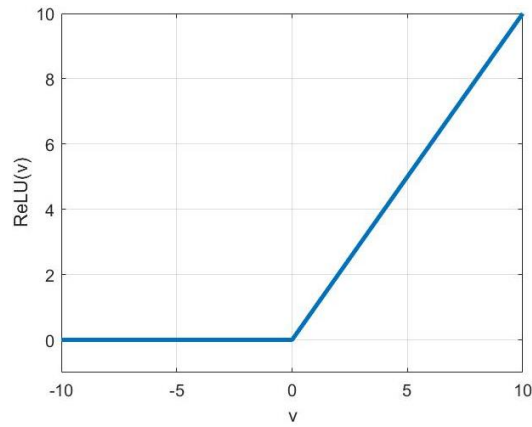


Şekil 2.4 Hiperbolik tanjant fonksiyonu

Geçtiğimiz yıllarda oldukça fazla kullanılmaya başlanan bir diğer aktivasyon fonksiyonu ise ReLU (Rectified Linear Unit) fonksiyonudur ve Eşitlik (2.5) ile tanımlanır.

$$\varphi(v) = \begin{cases} 0, & v \leq 0 \\ v, & v > 0 \end{cases} \quad (2.5)$$

ReLU fonksiyonu **şekil 2.5** ile gösterilmiştir. ReLU fonksiyonunun kullanılmasıyla tüm nöronlar aynı anda etkinleştirilmemektedir. Yani bir nöronun yalnızca doğrusal dönüşüm çıktısı sıfır olduğunda devre dışı bırakılmaktadır (Sharma 2017). Sigmoid ve hiperbolik tanjant fonksiyonları ile karşılaştırıldığında öğrenme hızının arttığı belirlenmiştir.



Şekil 2.5 ReLU fonksiyonu

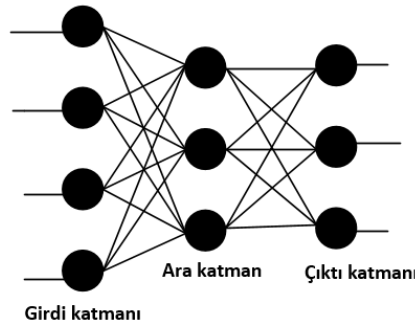
Çoklu sınıflandırma problemleri için kullanılan bir diğer aktivasyon fonksiyonu ise Softmax aktivasyon fonksiyonudur. Bu aktivasyon fonksiyonu verilen her bir girdinin bir sınıfa ait olma olasılığını gösteren 0 ile 1 arasında çıktılar üretir. Sigmoid fonksiyonunun genelleştirilmiş halidir. Bu aktivasyon fonksiyonu genellikle sınıflandırma işlemi yapan yapay sinir ağlarının çıkış birimi olarak kullanılır.  $N$  sınıf sayısı olmak üzere girdinin  $i$  sınıfına ait olma olasılığı Eşitlik (2.6) ile hesaplanır.

$$p_i = \frac{e^{v_i}}{\sum_{j=1}^N e^{v_j}} \quad (2.6)$$

### 2.1.2 Yapay sinir ağının yapısı

1950'li yıllarda Rosenblatt tarafından tüm model sinir ağlarının kaynağı olan algılayıcı tanımı yapılmıştır (Haykin 2009). Temel olarak değiştirilebilir sinaptik ağırlıklara ve bir yanlılık değerine sahip tek bir nörondan oluşur.

Yapay sinir hücrelerinin bir araya gelmesiyle yapay sinir ağı oluşturulur. Genel olarak bir yapay sinir ağı girdi katmanı, ara katmanlar ve çıktı katmanından oluşur. Dış dünyadan girdi katmanı ile alınan bilgiler ara katmanlara iletilir. Bu bilgiler ara katmanlarda işlenerek çıktı katmanına iletilir. Çıktı katmanında ise ara katmanlardan gelen bilgiler işlenerek girdi bilgisi için üretilmesi gereken çıktı üretilir. Bir girdi, bir ara katman ve bir çıktı katmanından oluşan yapay sinir ağı **şekil 2.6** ile gösterilmiştir. Bu yapay sinir ağı iki katmanlı bir sinir ağıdır, girdi katmanı ağı katman sayısına dahil edilmemektedir.



Şekil 2.6 İki katmanlı yapay sinir ağı

Ağa verilen girdilere karşı doğru çıktılarının oluşturulması istenmektedir. Girdiler bir vektöre dönüştürülerek ağa verilir ve ağın parametre değerleri doğru çıktı oluşturacak şekilde düzenlenir. Başlangıçta rastgele olarak belirlenen ağırlık değerlerinin, ağa örnekler verildikçe doğru çıktılar üretecek şekilde güncellenmesine ağın eğitilmesi denir (Öztemel 2006). Örnekler ile genelleme yapabilecek ağırlık değerlerinin bulunması ile ağın öğrenmesi sağlanır. Ağın eğitiminin ardından daha önce hiç ağa verilmemiş örneklerden oluşan test veri seti ile performans ölçümü yapılır. Bu öğrenme stratejisine gözetimli (supervised) öğrenme denir. Sisteme sadece girdilerin verildiği, veri setindeki örneklerin parametreler arasındaki ilişkinin kendiliğinden öğrenildiği öğrenme stratejisine ise gözetimsiz (unsupervised) öğrenme denir.

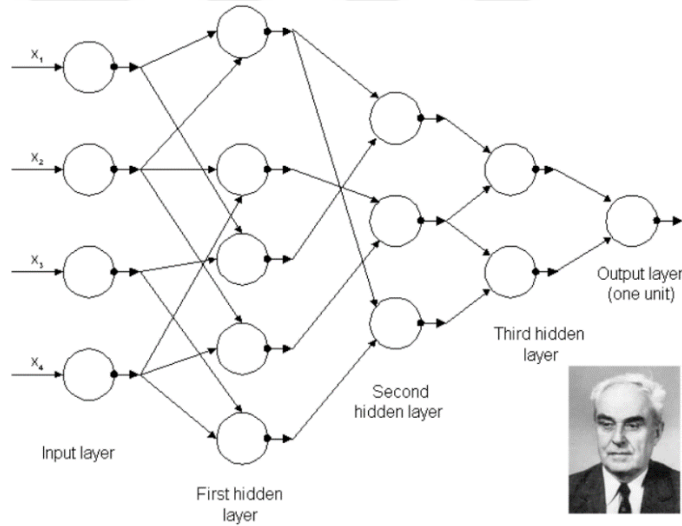
## 2.2 Derin Öğrenme

Tarihsel olarak, derin öğrenme kavramının kaynağı yapay sinir ağı araştırmalarıdır. Girdi ve çıktı katmanları arasında en az 1 gizli katmanı bulunan yapı çok katmanlı sinir ağı olarak tanımlanır (Sazlı 2006). İleri beslemeli sinir ağları veya birçok gizli katmandan oluşan çok katmanlı algılayıcılar derin sinir ağları olarak adlandırılır (Bengio 2009). Her ardışık katman, önceki katmandaki çıktıyı girdi olarak almaktadır. Mimarinin ilk katmanlarında alt seviye öz nitelikler çıkarılır. Bu öz nitelikler sonraki katmanlarda girdi olarak alınarak daha yüksek seviyedeki öz niteliklerin öğrenilmesi sağlanır.

Gözetimli derin beslemeli çok katmanlı algılayıcılar için ilk genel, öğrenme algoritması Ivakknenko ve Lapa tarafından 1965 yılında yayınlanmıştır (Şeker 2017). Çalışmada, katmanlardaki en iyi öz nitelikler seçilerek bir sonraki katmanlara iletilmektedir. Bilinen en eski derin ağ mimarisi **şekil 2.7** ile gösterilmiştir.

Derin sinir ağlarının çalışmasında ilk olarak çözülmesi istenen problem için eğitim, doğrulama ve test veri setleri oluşturulur. Gözetimli derin sinir ağlarına, girdiler ile birlikte ağın üretmesi beklenen çıktılarının bir arada bulunduğu eğitim veri seti verilir. Temel olarak ağın eğitilmesi iki aşamadan oluşmaktadır. Bunlar ağ çıktısının hesaplandığı ileriye doğru yayılma (forward propagation) ve ağ ağırlıklarının

güncellendiği geriye doğru yayılma (backward propagation) aşamalarıdır. İleriye doğru yayılma aşamasında eğitim aşamasındaki örnekler ağa girdi katmanı ile verilir ve ağırlık değerleri ile çarpılıp, aktivasyon fonksiyonundan geçirilerek ara katmanlar boyunca iletilir. Çıktı katmanında ağın çıktılarının elde edilmesi ile ileriye doğru yayılma sonlanır. Ağa verilen girdi değerlerine karşılık üretilen çıktı değerleri ile beklenen çıktı değerlerinin karşılaştırılarak aradaki fark, hata olarak hesaplanır ve geriye doğru yayılım aşaması başlar. Geriye doğru yayılım aşamasının amacı bu hatanın azaltılmasıdır ve hata ağın ağırlık değerlerine dağıtılarak ağırlık değerleri güncellenir. Modelin gizli katman sayısı, katman boyutları gibi hiperparametrelerini ayarlarken, eğitim veri setine göre oluşturulan modelin değerlendirilmesinde doğrulama veri seti kullanılır. Son olarak eğitim sırasında hiç kullanılmayan verilerden oluşan test veri seti, eğitim sonunda oluşturulan nihai modelin değerlendirilmesinde kullanılır.



Şekil 2.7 Ivakhnenko A. tarafından eğitilen bilinen ilk derin ağ mimarisi (Dev 2017)

### 2.2.1 İleri doğru yayılma

Girdi katmanından alınan örneklerin, ara katmanlar boyunca ilerlemesi ardından çıktı katmanına ulaşması ve verilen örneğe ilişkin çıkışın hesaplanması ile sonlanan adımdır. Bir  $l$  katmanındaki  $j$  nöronuna ait aktivasyon  $(l - 1)$  katmanındaki aktivasyon değerleri ile ilişkilidir ve Eşitlik (2.7) ile verilir (Janocha ve Czarnecki 2017).

$$a_j^l = \sigma(\sum_k w_{jk}^l a_k^{l-1} + b_j^l) \quad (2.7)$$

Burada  $w_{jk}^l$ ,  $(l - 1)$  katmanındaki  $k$  nöronu ile  $l$  katmanındaki  $j$  nöronu arasındaki ağırlık değeri,  $b_j^l$ ,  $l$  katmanındaki  $j$  nöronuna ait yanlılık değeri,  $a_j^{l-1}$ ,  $(l - 1)$  katmanındaki  $j$  nöronuna ait aktivasyon değeridir.

### 2.2.2 Kayıp fonksiyonu

Kayıp fonksiyonu belirlenen ağ parametrelerinin veri kümesine olan uygunluğunu değerlendirmek için kullanılan bir hata ölçüsüdür. Derin öğrenme modellerinin amacı, bu kayıp fonksiyon değerini en aza indirmektir. Kayıp fonksiyon değerini en aza indirme sürecine optimizasyon denir (Dao 2020). Veri setindeki tüm örnekler ya da örnek kümeleri için kayıp değerlerinin ortalamasına ise maliyet fonksiyonu (cost function) denir.

Kayıp fonksiyonu seçimi modellerin oluşturulması ve performansın iyileştirilmesinde önemli bir rol oynamaktadır. Sınıflandırma kayıp fonksiyonları, ikili sınıflandırma (binary classification) ve çok sınıflı sınıflandırma (multi-class classification) olarak ikiye ayrılabilir (Wang vd. 2020). İkili sınıflandırma, çıktının sıfır ve bir ile gösterilen iki öğeden biri olabileceği problemler iken çok sınıflı sınıflandırma ise çıktının ikiden daha fazla sınıftan birine ait olduğu problemdir.

İkili Çapraz Entropi Kaybı (Binary Cross Entropy Loss), hesaplanan değer beklenen değerden ne kadar uzakta olduğunu ölçerek nihai kaybı elde etmek için bu sınıfsal hataların ortalamasıdır.  $\hat{y}_i$ , ağ sonunda hesaplanan değer ve  $y_i$ , beklenen değer olmak üzere Eşitlik (2.8) ile verilir (Janocha ve Czarnecki 2017).

$$L = -y_i \log(\hat{y}_i) - (1 - y_i) \log(1 - \hat{y}_i) \quad (2.8)$$

İkili Çapraz Entropi Kaybının çok sınıflı problemlere uyarlanmış hali olan Kategorik Çapraz Entropi Kaybı, sınıf sayısı  $N$  olmak üzere Eşitlik (2.9) ile verilir.

$$L = \sum_{i=1}^N -y_i \log(\hat{y}_i) \quad (2.9)$$

### 2.2.3 Geriye doğru yayılma

İleriye Doğru Yayılma aşamasından sonra hesaplanan çıktı değerleri ile beklenen çıktı değerlerinin karşılaştırılmasıyla bir hatanın hesaplanması ile başlayan Geriye Doğru Yayılma aşaması, girdi katmanına kadar olan ağırlık ve yanlılık değerlerinin hesaplanan hatayı azaltmak için güncellenmesiyle son bulur.

Geri yayılım algoritması (backpropagation algorithm) ileri beslemeli sinir ağlarının eğitilmesinde yaygın olarak kullanılan bir algoritmadır (Goodfellow vd. 2016). Temel olarak geriye yayılımda, maliyet fonksiyonunun ağırlık değerlerine veya yanlılık değerine göre kısmi türevi hesaplanır. Bu türev değeri; ağırlık ve yanlılık değerleri değiştiği durumda maliyet fonksiyonunun ne kadar değiştiğini ifade etmektedir (Janocha ve Czarnecki 2017). Böylece geriye yayılım hem hızlı bir algoritmadır hem de ağırlık ve yanlılık değerlerinin değiştirilmesinin ağın genel davranışını nasıl değiştireceğine dair ayrıntılı bilgi sunmaktadır.

### 2.2.4 Optimizasyon

Optimizasyon bir  $f(x)$  fonksiyonu verildiğinde bu fonksiyonun en küçük veya en büyük değerini bulmaya yarayan bir algoritmadır. Derin öğrenme bağlamında, kayıp fonksiyonu optimize ederek sinir ağını eğitmek için optimizasyon algoritmaları kullanılır.

Geriye Doğru Yayılım aşamasında hatanın ağda geriye doğru yayılması için  $\frac{\partial C}{\partial w_{jk}^l}$  ve  $\frac{\partial C}{\partial b_j^l}$  kısmi integraller hesaplanır. Seçilen optimizasyon yöntemine göre bu parametrelerin güncellenmesi sağlanır (Nielsen 2015).

Hesaplanan çıktı değerleri, beklenen değerlere yaklaşık olarak tüm girdiler için eşit olduğunda hesaplanan maliyet fonksiyonunun değeri oldukça küçük bir değere sahiptir; yani sıfıra yaklaşır. Eğer hesaplanan maliyet fonksiyonu değeri büyük ise hesaplanan çıktı değerleri, fazla sayıdaki girdi için beklenen çıktı değerine yakın değildir. Buradaki

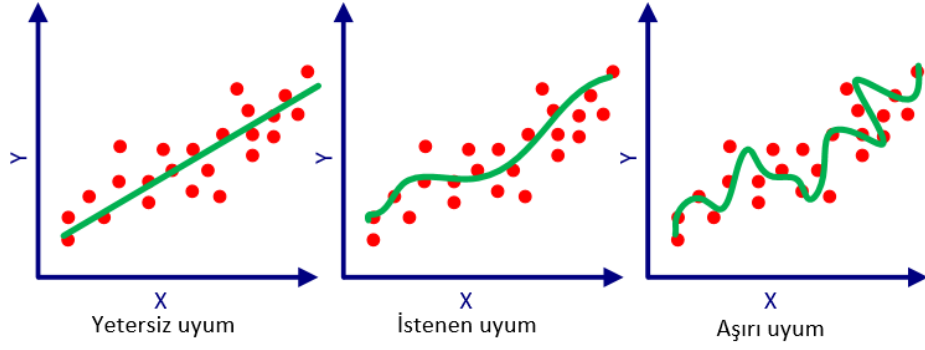
öğrenme algoritmasının amacı, ağırlık ve yanlılık değerlerinin fonksiyonu olan maliyeti minimum yapmaktır. Yani maliyeti olabildiğince minimum yapabilecek bir ağırlık ve yanlılık değerleri kümesi aranır.

### 2.2.5 Derin öğrenme tekniklerinin iyileştirilmesi

Derin öğrenme algoritmalarında eğitim aşamasında karşılaşılan aşırı uyum (overfitting) ve yetersiz uyum (underfitting) gibi birkaç temel problem bulunmaktadır. Denetimli derin öğrenme algoritmalarının görevi girdi ile çıktı verileri arasındaki ilişkiyi en doğru şekilde modellemektir. Ancak gerçek dünya uygulamalarında ikisi arasındaki gerçek ilişkiyi yakalamak için çok daha fazla veriye ihtiyaç duyulmaktadır. Dolayısıyla derin öğrenme modelinin en önemli özelliği genelleme kabiliyetidir. Modelin genelleme kabiliyeti ne kadar yüksek ise daha önce ağa verilmemiş verilerden daha anlamlı sonuçlar üretilebilir. Modelin, problemin karmaşıklığını uygun bir şekilde yakalayamamasına yetersiz uyum denir. Çok fazla parametresi olan modelin temel yapıyı ve genellemeyi ortaya çıkarmadan eğitim verisine uyum sağlamasına aşırı uyum ya da ezberleme denir.

Aşırı uyum ve yetersiz uyum problemlerinin her ikisi için de genelleme yeteneği zarar görmektedir. Bu durumlar **şekil 2.8** ile gösterilmiştir. Yetersiz uyum durumunda model problemin gerçek karmaşıklığını yakalayamaz ve istenen çıktıyı güvenilir bir şekilde tahmin edemez. Aşırı uyum durumunda ise gürültü gibi problemin rastgele doğasının bir sonucu olabilecek bir veri karmaşıklığı bile yakalanır. Burada kırmızı ile gösterilen noktalar verileri gösterirken yeşil çizgi bu verilere karşılık oluşturulan modelleri göstermektedir.





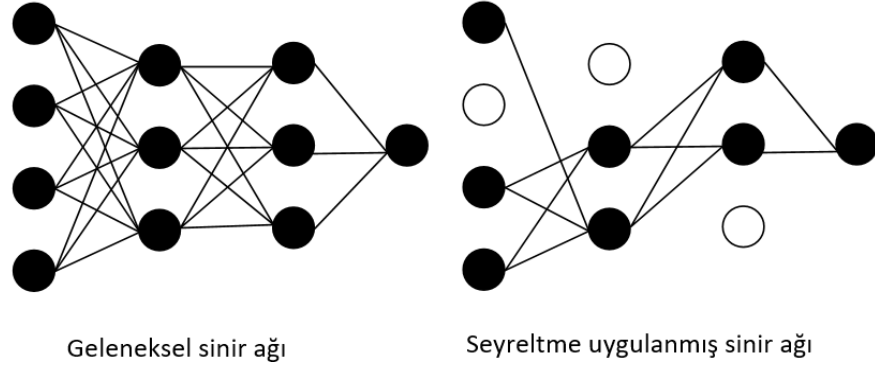
Şekil 2.8 Yetersiz uyum ve aşırı uyum durumlarının gösterimi

Aşırı uyum problemi için en basit çözüm erken durma (early stopping) yöntemidir. Bu yöntemle temel olarak eğitim veri seti için hesaplanan hata ile test veri seti için hesaplanan hata arasındaki fark artmaya başladığında ağın eğitimi durdurulur.

Aşırı uyum için problemi için bir diğer çözüm regülasyon (regularization) yöntemidir. Bu yöntemle ağın genelleştirme kabiliyeti artırılmaya çalışılır. Buradaki temel fikir maliyet fonksiyonuna bir terim ekleyerek model karmaşıklığını artırmaktır.

En sık kullanılan regülasyon tekniği veri setini artırma (dataset augmentation) yöntemidir. Derin öğrenme uygulamalarında genellikle ne kadar çok veri mevcutsa öğrenme performansı o kadar yüksektir. Nesne tanıma veya sınıflandırma problemlerinde genellikle orijinal resme döndürme (rotation), öteleme (translation), ölçekleme (scaling), gürültü ekleme (add noise), resmi belli bölgelerden kırma (cropping) gibi farklı teknikler uygulanmaktadır (Tan 2019).

Aşırı uyumu engellemek için yaygın olarak kullanılan bir diğer yöntem ise seyreltme (dropout) tekniğidir. Bu yöntem ile belli bir eşik seviyesi altındaki düğümler devre dışı bırakılır yani zayıf bilgilerin unutulması sağlanır. Geleneksel bir sinir ağı ile seyreltme uygulanmış sinir ağı **şekil 2.9** ile gösterilmiştir.



Şekil 2.9 Geleneksel sinir ağı ile seyreltme uygulanmış sinir ağının karşılaştırılması

Derin öğrenme uygulamalarında kullanılan veri kümeleri çok büyük olduğundan ağ girdileri önemli ölçüde değişiklik gösterir. Yığın normalleştirme (batch normalization) veri setinin tümüne ön işleme uygulama yerine veriyi gruplara ayırarak normalleştirme işleminin uygulanmasıdır ve derin öğrenme uygulamalarında ağın öğrenme hızını artıran bir yöntemdir (Baydar 2018).

### 2.2.6 Derin öğrenme mimarileri

Çözülmesi gereken probleme göre farklı derin öğrenme mimarileri bulunmaktadır. Sınıflandırma, tespit, tanı, tahmin, teşhis vb. olmak üzere problemlerin yapısına göre tercih edilecek olan mimari değişiklik göstermektedir. Aynı zamanda görüntü, ses, sinyal olması gibi verinin türüne göre de seçilen mimari değişmektedir.

Evrişimli sinir ağı, derin öğrenme mimarileri içinde en çok kullanılan sinir ağıdır (Doğan ve Türkoğlu 2017). Evrişimli sinir ağı mimarisi, özellikle görüntü ve video analizlerinin yapılmasında kullanılmaktadır. Hayvan görü sistemine dayanan bir yaklaşımdır. Sınıflandırma problemlerinde oldukça başarılı sonuçlar elde edilmektedir. Adını aldığı evrişim katmanı en temel katmandır ve girdinin öz niteliklerinin çıkarılması sağlanır. Evrişim katmanı dışında aktivasyon katmanı, havuzlama katmanı, tam bağımlı katman gibi katmanlar bulunmaktadır.

Tekrarlayan sinir ağı (Recurrent Neural Network), sıklıkla kullanılan bir diğer derin öğrenme mimarisidir. Sadece ağa girdi olarak verilen örnekler değil daha önce zaman

serisi içerisindeki giriş örneklerini de alır (Doğan ve Türkoğlu 2017). Tekrarlayan sinir ağlarındaki temel düşünce sıralı bilgilerin kullanılmasıdır. Tekrarlayan sinir ağları genellikle doğal dil işleme uygulamalarında kullanılmaktadır. Metin çevirisi, resimler için altyazı oluşturma, gürültü verilerden gürültüsüz verilerin elde edilmesi, metin özetlerinin çıkarılması, el yazısı tanıma, konuşma tanıma gibi problemlerin çözümünde sıklıkla kullanılmaktadır. Ayrıca RNN ve CNN yapılarının bir arada kullanıldığı da birçok çalışma bulunmaktadır (Kayaalp ve Süzen 2018).

Uzun Kısa Süreli Hafıza Ağları (Long-Short Term Memory – LSTM) ise RNN yapısının hafıza bakımından yetersiz geldiği problemlerde kullanılmak üzere geliştirilmiştir. LSTM kullanılan birçok uygulama alanı vardır. Resimlerden otomatik başlık çıkarma, sessiz videolara otomatik ses ekleme, ilişkili metinlerden kelime üretme LSTM kullanılan problemlere örnek olarak verilebilir (Kayaalp ve Süzen 2018).

Sınırlı Boltzmann makineleri (Restricted Boltzmann Machines) sınıflandırma, regresyon ve öz nitelik öğrenimi gibi problemlerde kullanılır ve girdi olarak kullanılan veri seti üzerinden olasılık dağılımlarını öğrenebilen bir sinir ağıdır. Derin inanç ağı (Deep Belief Network) ise sınırlı Boltzmann makinelerinin bir yığını olarak tanımlanır. Görüntü tanıma ve üretme problemlerinde kullanılmaktadır.

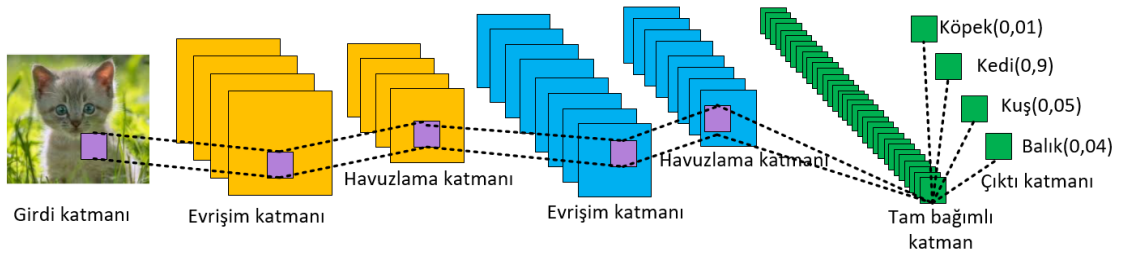
### **2.3 Evrişimli Sinir Ağları**

Geleneksel yapay sinir ağları, biyolojik sinir sistemlerinin çalışma şekline büyük ölçüde ilham alarak nihai çıktıyı optimize etmek için girdiden toplu olarak öğrenmek üzere dağıtılmış bir şekilde çalışan yüksek sayıda birbirine bağlı hesaplama düğümlerinden oluşur. Evrişimli sinir ağları, öğrenme yoluyla kendi kendini optimize eden nöronlardan oluşmaları bakımından geleneksel yapay sinir ağlarına benzemektedir (O'Shea ve Nash 2015). Evrişimli sinir ağlarının öz nitelik çıkarmak için görüntünün yerel özelliklerine bakması, geleneksel sinir ağları ile arasındaki temel farkı oluşturur. Ardından gelen tam bağımlı katmanlar ise global veriyi çıkarmak için kullanılır.

Evrişimli sinir ağları, hayvanların görsel sisteminden ilham almaktadır (Baydar 2018). Kedi görme sistemi için basit hücreler ve karmaşık hücreler olmak üzere iki tip hücre tanımı yapılmıştır. Fukushima tarafından tasarlanan Neocognitron adlı çok katmanlı yapay sinir ağı mimarisinde bu iki tip hücreden ilham alınmıştır. Bu mimaride ilk olarak temel öz nitelikler basit hücreler tarafından bulunur. Daha sonra bu yerel öz nitelikler karmaşık hücreler tarafından birleştirilerek daha karmaşık öz nitelikler elde edilir (Fukushima ve Miyake 1982). Bu mimari, evrişimli sinir ağları için önemli bir ilham kaynağı olmuştur.

Evrişimli sinir ağları, görüntü ve video işleme uygulamalarında sıklıkla kullanılmaktadır ve diğer derin öğrenme mimarilerine göre yüksek başarı oranları sağlamaktadır. Yüz tanıma, görüntü sınıflandırma, sahne etiketleme, hareket tanıma, insan duruşu tahmini, doküman analizi problemleri bu alandaki uygulamalara örnek olarak gösterilebilir (Yamashita vd. 2018). Evrişimli sinir ağları bilgisayarla görü problemlerini temel olarak tasarlanmış olsa da ham sinyallerden bilgi çıkarmak için kullanıldığından doğal dil işleme uygulamalarında da sıklıkla kullanılmaktadır. Konuşma tanıma ve metin sınıflandırma gibi problemler, bu alana örnek gösterilebilir.

Evrişimli sinir ağı bir girdi katmanından, bir çıktı katmanından ve aradaki birden fazla gizli katmandan oluşur. Evrişimli sinir ağının gizli katmanları temel olarak evrişim katmanları, havuzlama katmanları ve tam bağımlı katmanlardan oluşmaktadır. Tipik bir evrişimli sinir ağı mimarisi, **şekil 2.10** ile gösterildiği gibi birkaç evrişim ve havuzlama katmanından oluşan yığının tekrarlanmasıyla bir veya daha fazla tam bağımlı katmandan oluşur.



Şekil 2.10 Evrişimli sinir ağının genel yapısı

Evrişimli sinir ağlarının katmanlarındaki temel işlevler şu şekilde sıralanabilir:

- Girdi katmanı, evrişimli sinir ağlarının ilk katmanını oluşturmaktadır. Girdi olarak verilen veri genellikle ham haldedir, yani herhangi bir ön işleme yapılmamıştır.
- Evrişim katmanında, girdinin yerel bölgelerine bağlı olan nöronların çıkışı belirlenir. Bu çıkış, ağırlıklar ile girdi hacmine bağlı bölge arasındaki sayısal çarpımın hesaplanması ile belirlenir. Doğrusal olmayan bir yapı elde etmek için bir aktivasyon fonksiyonu uygulanır.
- Havuzlama katmanında, verilen girdiye alt örnekleme uygulanarak bir sonraki katman için girdi boyutu azaltılır. Böylece parametre sayısındaki azalma ile hesap yükü azaltılır ve sistemin ezberlemesi engellenir.
- Evrişimli sinir ağlarında, birbirini takip eden evrişim ve havuzlama katmanlarından sonra tam bağımlı katman gelir. Buradaki işlemler geleneksel sinir ağlarındaki gibidir.
- Son olarak çıktı katmanı ise sınıflandırma işleminin yapıldığı katmandır.

### 2.3.1 Evrişim katmanı

Evrişim katmanı, tipik olarak evrişim işlemi ve aktivasyon işlevi gibi doğrusal ve doğrusal olmayan işlemlerin bir kombinasyonundan oluşan, öz nitelik çıkarımı gerçekleştiren evrişimli sinir ağlarının temel bileşenidir (Yamashita vd. 2018). Evrişim, çekirdek (kernel) adı verilen küçük bir sayı dizisinin tensör adı verilen bir sayı dizisi olan girişe uygulanarak öz nitelik çıkarmak için kullanılan özel bir doğrusal işlem türüdür. Evrişim işlemi Eşitlik (2.9) ile verildiği gibi yıldız işareti ile gösterilir (Goodfellow vd. 2016).

$$s(t) = x(t) * w(t) \quad (2.9)$$

Verilen bir  $x$  fonksiyonunun,  $w$  çekirdeği ile bir boyutlu evrişim işlemi Eşitlik (2.10) ile verilir (Gummeson 2016).

$$s(t) = \int x(a)w(t - a)da \quad (2.10)$$

Fonksiyonun ve çekirdeğin, zaman indeksinin belli değerlerinde tanımlı olduğu durumda ayrık evrişim Eşitlik (2.11) ile verilir.

$$s(t) = \sum_{a=-\infty}^{\infty} x(a)w(t - a) \quad (2.11)$$

Derin öğrenme uygulamalarında, girdi genellikle çok boyutlu bir veri dizisi iken çekirdek öğrenme algoritması tarafından uyarlanan çok boyutlu bir parametre dizisidir. İki boyutlu bir görüntü,  $I$  girdi olarak ve iki boyutlu çekirdek  $K$  kullanılarak iki boyutlu evrişim işlemi Eşitlik (2.12) ile verilir.

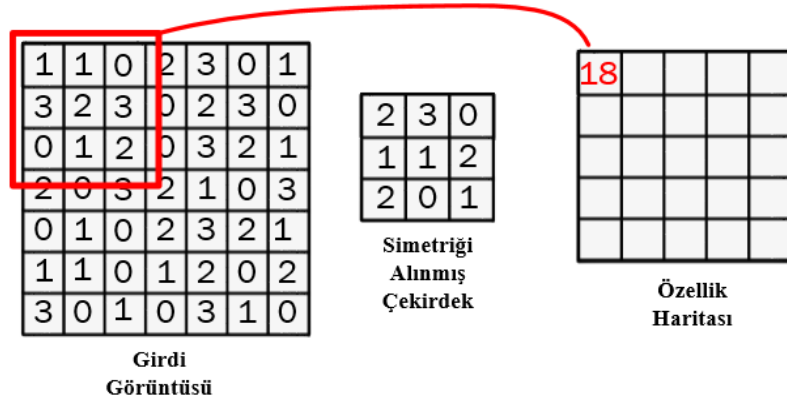
$$S(i, j) = I(i, j) * K(i, j) = \sum_m \sum_n I(m, n)K(i - m, j - n) \quad (2.12)$$

Yani evrişim işlemi ile kullanılan çekirdeğin  $x$  ve  $y$  eksenlerine göre simetriği alınarak, girdi görüntüsü üzerinde gezdirilmesi sağlanır. Örnek olarak **şekil 2.11** ile gösterildiği gibi  $7 \times 7$  boyutundaki bir girdi görüntüsü,  $3 \times 3$  boyutunda bir çekirdek olsun. **Şekil 2.11** ile aynı zamanda çekirdeğin  $x$  ve  $y$  eksenine göre simetriği alınmış hali de gösterilmiştir.

1	1	0	2	3	0	1	1	0	2	2	3	0
3	2	3	0	2	3	0	2	1	1	1	1	2
0	1	2	0	3	2	1	0	3	2	2	0	1
2	0	3	2	1	0	3	0	3	2	2	0	1
0	1	0	2	3	2	1	2	0	1	2	0	1
1	1	0	1	2	0	2	2	0	1	2	0	1
3	0	1	0	3	1	0	2	0	1	2	0	1
							<b>Çekirdek</b>			<b>Simetriği Alınmış Çekirdek</b>		
<b>Girdi Görüntüsü</b>												

Şekil 2.11 Evrişim işlemi için girdi görüntüsü ve çekirdek

Çekirdeğin  $x$  ve  $y$  ekseninin simetriği alınarak bulunan evrişim filtresiyle ilk olarak **şekil 2.12** ile gösterildiği gibi girdi görüntüsünün sol üst köşesi üzerine denk getirilerek eleman çarpımları yapıldıktan sonra toplanarak girdi görüntüsünün bu bölümüne karşılık bir değer elde edilir.

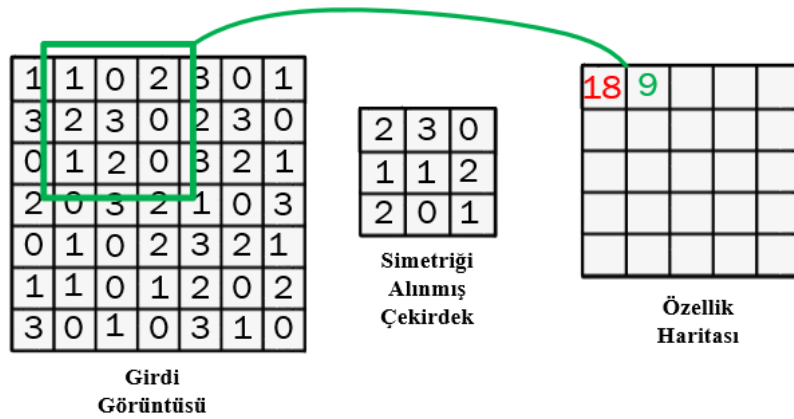


Şekil 2.12 Evrişim işleminin ilk adımı

Burada aynı pozisyondaki elemanlar çarpılarak toplanır. Girdi görüntüsün sol üst köşesine denk gelen bölüme karşılık gelen değer Eşitlik (2.13) ile hesaplanır.

$$1 \times 2 + 1 \times 3 + 0 \times 0 + 3 \times 1 + 2 \times 1 + 3 \times 2 + 0 \times 2 + 1 \times 0 + 2 \times 1 = 18 \quad (2.13)$$

İlk değer bulunmasından sonra, **şekil 2.13** ile gösterildiği gibi girdi görüntüsü üzerinde işlem yapılan çerçevenin bir yana kaydırılması ile aynı işlem tekrarlanarak ikinci değer elde edilir.



Şekil 2.13 Evrişim işleminin ikinci adımı

Bu şekilde girdi görüntüsünün her konumu için çekirdeğin her bir üyesi ile girdi arasındaki eleman bazlı bir hesaplama yapılarak çıktının karşılık gelen konumundaki çıkış değeri hesaplanır ve öz nitelik haritası (feature map) olarak adlandırılır (Yamashita

vd. 2018). Örnek olarak verilen girdi ve çekirdeğe karşılık oluşturulan öz nitelik haritası **şekil 2.14** ile gösterilmiştir. Öz nitelik haritası, orijinal görüntüdeki eşsiz özelliklerin vurgulanmasını sağlamaktadır.

1	1	0	2	3	0	1
3	2	3	0	2	3	0
0	1	2	0	3	2	1
2	0	3	2	1	0	3
0	1	0	2	3	2	1
1	1	0	1	2	0	2
3	0	1	0	3	1	0

**Girdi Görüntüsü**

18	9	15	23	18
24	18	21	17	24
11	19	14	18	26
7	17	22	18	15
12	5	16	17	24

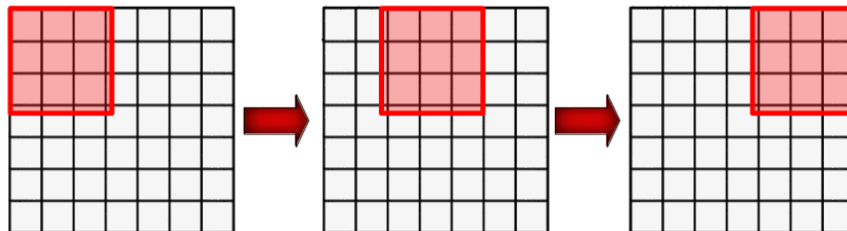
**Özellik Haritası**

Şekil 2.14 Girdi görüntüsü ve öz nitelik haritası

**Şekil 2.14**'de görüldüğü gibi  $7 \times 7$  boyutundaki girdi,  $3 \times 3$  boyutundaki bir çekirdek için oluşturulan öz nitelik haritası  $5 \times 5$  boyutundadır.  $N \times N$  boyutunda bir girdi,  $M \times M$  boyutundaki bir çekirdek için oluşturulan öz nitelik haritasının boyutu Eşitlik (2.14) ile hesaplanır.

$$O = N - M + 1 \quad (2.14)$$

Buraya kadar anlatılan evrişim işleminde çekirdek, girdi görüntüsünün üzerinde bir adım kaydırılarak gezdirilmiştir. Evrişimli sinir ağlarında hesaplanan parametre sayısını azaltmak için bu adım sayısı farklı bir değer seçilebilir. Adım değerinin iki olduğu durumda çekirdeğin girdi üzerinde gezdirilme aşaması **şekil 2.15** ile gösterilmiştir.



Şekil 2.15 Adım uzunluğunun iki olduğu durumda çekirdeğin gezdirilmesi



**Şekil 2.15** ile görüldüğü gibi adım uzunluğunun iki olduğu durumda  $7 \times 7$  boyutundaki bir girdi ile  $3 \times 3$  boyutundaki bir çekirdek ile oluşturulan öz nitelik haritası  $3 \times 3$  boyutundadır. Böylece adım uzunluğunun bir olduğu duruma göre daha küçük boyutta bir öz nitelik haritası elde edilmiştir. Eşitlik (2.14) ile verilen öz nitelik haritasının boyutu, adım uzunluğu birden farklı ve  $S$  olduğu durumda Eşitlik (2.15) ile verilir.

$$O = \frac{N-M}{S} + 1 \quad (2.15)$$

Yukarıda açıklanan evrişim işlemi, her bir çekirdeğin merkezinin girdinin en dıştaki elemanlarıyla örtüşmesine izin vermez. Görüntünün sınırlarında var olabilecek bilginin kaybedilmesi evrişim aşamasının dezavantajlarından biridir. Ayrıca evrişim işlemi sonucunda oluşturulan öz nitelik haritasının boyutu da girdi boyutundan küçüktür. Sıfır doldurma (zero padding) bu sorunu ele alan tekniklerden biridir (Albawi vd. 2017). Bu yöntemle çekirdeğin merkezinin girdinin en dıştaki elemanına denk gelmesini sağlayacak şekilde **şekil 2.16** ile gösterildiği gibi sıfırlardan oluşan elemanlar eklenir.

0	0	0	0	0	0	0	0	0
0	1	1	0	2	3	0	1	0
0	3	2	3	0	2	3	0	0
0	0	1	2	0	3	2	1	0
0	2	0	3	2	1	0	3	0
0	0	1	0	2	3	2	1	0
0	1	1	0	1	2	0	2	0
0	3	0	1	0	3	1	0	0
0	0	0	0	0	0	0	0	0

Şekil 2.16 Sıfır doldurma yapılan girdi görüntüsü

Sıfır doldurma işleminin ardından oluşturulan öz nitelik haritası  $7 \times 7$  boyutundadır ve girdi görüntüsünün boyutuna eşittir. Eşitlik (2.15) ile verilen öz nitelik haritasının boyutu, sıfır ile doldurulan sıra sayısı  $P$  olmak üzere Eşitlik (2.16) ile verilir.

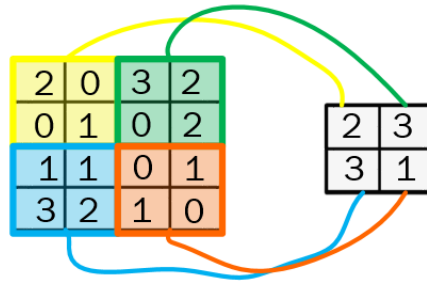
$$O = \frac{N+2P-M}{S} + 1 \quad (2.16)$$

Doğrusal bir işlem olan evrişim işleminden sonra öz nitelik haritası, eleman bazlı olarak doğrusal olmayan bir aktivasyon fonksiyonundan geçirilir.

### 2.3.2 Havuzlama katmanı

Evrişim katmanından sonra gelen havuzlama katmanı, öğrenilebilir parametre sayısını azaltmak için öz nitelik haritalarına alt örnekleme yaparak boyutunun azaltılmasını sağlar. Aynı zamanda sistemin ezberlemesi de engellenir. Evrişim katmanına benzer olarak havuzlama katmanı için de filtre boyutu, adım uzunluğu, sıfır doldurma parametreler tanımlanmaktadır.

Havuzlama katmanı için kullanılan en sık yöntem maksimum havuzlamadır. Burada bir filtrenin girdi üzerinde gezdirildiği düşünülür. Bu filtrenin denk geldiği elemanlardan maksimum değerde olanı, o bölge için çıktı değerini oluşturmaktadır. Genellikle adım uzunluğu, filtre boyutu da  $2 \times 2$  seçilmektedir (O'Shea ve Nash 2015). **Şekil 2.17** ile gösterildiği gibi  $4 \times 4$  boyutundaki bir öz nitelik haritası için  $2 \times 2$  boyutundaki maksimum havuzlama filtresi adım uzunluğunun 2 olduğu durumda  $2 \times 2$  boyutunda bir çıktı elde edilmektedir.



Şekil 2.17 Maksimum havuzlama

### 2.3.3 Tam bağımlı katman

Evrişimli sinir ağları mimarilerinde birbiri ardına gelen evrişim ve havuzlama katmanlarından sonra tam bağımlı katman gelir. Nihai evrişim ve havuzlama katmanından oluşturulan öz nitelik haritaları tek boyutlu bir sayı dizisine dönüştürülür ve bir veya birden fazla tam bağımlı katmana bağlanılarak ağın son çıkışına eşlenir.

Örneğin bu çıkış bir sınıflandırma problemi için her bir sınıfın olasılığını içerebilir. Burada çıktı katmanı için kullanılan aktivasyon fonksiyonu genellikle diğerlerinden farklıdır (Yamashita vd. 2018). Problemlere göre uygun bir etkinleştirme işlevinin seçilmesi gereklidir. Örneğin çok sınıflı sınıflandırma problemi için softmax fonksiyonu kullanılabilir. Böylece her bir sınıfa ait çıktı değeri 0 ile 1 arasında bir değer alır.

## 2.4 İşaret Dili

İşaret dili, işitme engelli insanlar arasındaki temel iletişim yöntemidir. Sıradan bir kişinin işitme engelli insanlarla iletişim kurabilmesi için, işaret dilini doğal dile çevirmek ve tersi için genellikle bir tercümana ihtiyaç duyulur (International Bibliography of Sign Language).

İşaret dilleri kendi dilbilgisi ve sözcükleriyle tam teşekküllü doğal dillerdir. Bu, işaret dillerinin evrensel olmadığı ve işaret dilleri arasında çarpıcı benzerlikler olmasına rağmen karşılıklı olarak anlaşılabilir olmadıkları anlamına gelmektedir. İşaret dilleri kullanılan ülkeye göre farklılık göstermektedir; hatta aynı ülkenin farklı yörelerinde bile farklı işaret dilleri kullanılmaktadır. Her işaret dilinin kendi alfabesi tanımlıdır. Tanımlanan bu harfler ve rakamlar sadece el hareketlerinden oluşurken, tek veya iki elin kullanıp kullanılmayacağı ve hareketlerin durağan veya dinamik olup olmayacağı yine işaret diline göre değişen unsurlardır.

İşaret dilleri ile kelimelerin anlatımı ve cümle kurulması ise genellikle dinamik hareketlerden oluşmaktadır. Hatta sadece el hareketlerinin kullanılması yerine kol, vücut ve yüz hareketleri de anlatıma dahil olmaktadır.

İşaret dili tanıma sistem mimarileri, girdilerine göre iki ana sınıfa ayrılmaktadır. Birincisi girişini sensör aracılığıyla eldivenlerden alan, eldiven verisi tabanlı mimaridir. Bu yaklaşımın avantajı yüksek doğruluğa sahip olması iken zayıf tarafı ise insan hareket alanını kısıtlayan bir yapıya sahip olmasıdır. İkinci yaklaşım ise girişini bir kamera ile sağlayan görme temelli mimaridir. Bu yaklaşımın avantajı eldiven verisi tabanlı yaklaşıma göre daha ucuz ve daha az sınırlayıcı olması iken zayıf tarafı daha fazla hesaplama gücü tüketmesidir.

### 3. LİTERATÜR TARAMASI

Arda Mavi çalışmasında yeni oluşturulan işaret dili rakam veri setini tanıtmakta ve bu veri setinin kullanıldığı sınıflandırma problemi için bir CNN mimarisi önermektedir (Mavi 2020). Çalışmada aynı zamanda önerilen model, popüler CNN modelleri ile karşılaştırılmaktadır. Çalışmada sunulan veri seti, tez çalışması kapsamında oluşturulan veri setine dahil edilmiştir ve 4.1.1 bölümünde veri seti tanıtılmıştır. Hazırlanan veri setiyle kullanılmak üzere seçilen iki popüler CNN mimarisi, MVGG-5 ve MVGG-9 mimarileridir. Bu mimarilerin konfigürasyonları **çizelge 3.1** ile verilmiştir.

Çizelge 3.1 MVGG-5 ve MVGG-9 mimari konfigürasyonları

<b>MVGG-5</b>	<b>MVGG-9</b>
Evrişim katmanı, 16 filtre, $3 \times 3$ çekirdek, ReLU	Evrişim katmanı, 16 filtre, $3 \times 3$ çekirdek, ReLU
Evrişim katmanı, 16 filtre, $3 \times 3$ çekirdek, ReLU	Evrişim katmanı, 16 filtre, $3 \times 3$ çekirdek, ReLU
Maksimum havuzlama katmanı, $2 \times 2$ filtre	Maksimum havuzlama katmanı, $2 \times 2$ filtre
Evrişim katmanı, 48 filtre, $3 \times 3$ çekirdek, ReLU	Evrişim katmanı, 32 filtre, $3 \times 3$ çekirdek, ReLU
Maksimum havuzlama katmanı, $2 \times 2$ filtre	Evrişim katmanı, 32 filtre, $3 \times 3$ çekirdek, ReLU
Tam bağımlı katman, 128 nöron, ReLU	Maksimum havuzlama katmanı, $2 \times 2$ filtre
Tam bağımlı katman, 10 nöron, Softmax	Evrişim katmanı, 48 filtre, $3 \times 3$ çekirdek, ReLU
	Evrişim katmanı, 48 filtre, $3 \times 3$ çekirdek, ReLU
	Maksimum havuzlama katmanı, $2 \times 2$ filtre
	Evrişim katmanı, 64 filtre, $3 \times 3$ çekirdek, ReLU
	Evrişim katmanı, 64 filtre, $3 \times 3$ çekirdek, ReLU
	Maksimum havuzlama katmanı, $4 \times 4$ filtre
	Tam bağımlı katman, 128 nöron, ReLU
	Tam bağımlı katman, 10 nöron, Softmax

Çalışmada önerilen CNN mimarisinin konfigürasyonu ise **çizelge 3.2** ile verilmiştir.

Çizelge 3.2 Çalışmada önerilen mimari konfigürasyonu

Evrişim katmanı, 32 filtre, $3 \times 3$ çekirdek, ReLU
Evrişim katmanı, 64 filtre, $3 \times 3$ çekirdek, ReLU
Maksimum havuzlama katmanı, $2 \times 2$ filtre
Evrişim katmanı, 64 filtre, $3 \times 3$ çekirdek, ReLU
Maksimum havuzlama katmanı, $2 \times 2$ filtre
Evrişim katmanı, 128 filtre, $3 \times 3$ çekirdek, ReLU
Maksimum havuzlama katmanı, $2 \times 2$ filtre
Tam bağımlı katman, 526 nöron, ReLU
Seyreltme katmanı, 0,5 seyreltme oranı
Tam bağımlı katman, 128 nöron, ReLU
Seyreltme katmanı, 0,5 seyreltme oranı
Tam bağımlı katman, 10 nöron, Softmax

Veri setinin %80'i eğitim, %20'si test veri seti olarak ayrılmıştır. AdaDelta optimizasyon fonksiyonu ve Kategorik çapraz entropi fonksiyonu kullanılmıştır. Test veri setinde doğruluk başarımı MVGG-5 mimarisi için %95, MVGG-9 mimarisi için %96 ve çalışmada önerilen model için %97'dir.

Çalışmada sunulan veri seti derin öğrenme uygulamalarında sıklıkla kullanılan veri setlerinden biri haline gelmiştir. Bir CNN mimarisi önerilmiş ve 2 farklı mimari ile karşılaştırılmıştır. Seçilen mimariler bu veri seti için en yüksek başarımları elde edilen MVGG mimarileridir. Önerilen model MVGG-5 mimarisinden daha çok, MVGG-9 mimarisinden daha az evrişim katmanı içermektedir ve doğruluk başarımı bu iki mimariden fazladır. Ancak önerilen model sadece bir veri seti üzerinde çalıştırılmıştır. Veri setindeki görüntüler aynı senaryo, ışıklandırma, arka planda gerçekleştirilen el hareketi görüntülerini içermektedir. Test görüntülerinin, çok az

değişişe sahip başka senaryolardaki görüntülerden oluşması durumunda oluşturulan modelin başarıml oranında düşüş görülecektir. Çünkü model bu veri setine göre oluşturulmuştur. Ayrıca farklı bir optimizasyon tekniğı seçilerek başarıml oranı da artırılabilir.

Hossain, Adhikary ve Soheli çalışmalarında işaret dili rakamlarının tanınması için CNN tabanlı bir model geliştirmeyi planlamışlardır (Hossain vd. 2020). Çalışmalarında Arda Mavi tarafından sunulan veri setini kullanmışlardır. Eğitim ve test aşaması için 3 farklı CNN ağı modeli ele alınmıştır ve bu modeller SCNN, MCNN ve TCNN olarak adlandırılmıştır.

2 gizli katmana sahip olan CNN genel modeli, SCNN olarak adlandırılmıştır ve bu model konfigürasyonu **çizelge 3.3** ile verilmiştir.

Çizelge 3.3 Çalışmada önerilen SCNN konfigürasyonu

Evrişim katmanı, 32 filtre, $3 \times 3$ çekirdek, ReLU
Maksimum havuzlama katmanı, $2 \times 2$ filtre
Evrişim katmanı, 64 filtre, $3 \times 3$ çekirdek, ReLU
Maksimum havuzlama katmanı, $2 \times 2$ filtre
Tam bağımlı katman, 128 nöron, ReLU
Tam bağımlı katman, 10 nöron, Softmax

SCNN modeline göre daha fazla gizli katmana sahip ikinci model MCNN olarak adlandırılmıştır ve konfigürasyonu **çizelge 3.4** ile verilmiştir.

Çalışmada transfer öğrenmenin kullanıldığı, yani önceden eğitilen Inception V3 ağıının dahil edildiğı model TCNN olarak adlandırılmıştır ve konfigürasyonu **çizelge 3.5** ile verilmiştir.

Çizelge 3.4 Çalışmada önerilen MCNN konfigürasyonu

Evrişim katmanı, 32 filtre, $3 \times 3$ çekirdek, ReLU
Maksimum havuzlama katmanı, $2 \times 2$ filtre
Normalleştirme katmanı, yığın normalleştirme
Seyreltme katmanı, 0,25 seyreltme oranı
Evrişim katmanı, 64 filtre, $3 \times 3$ çekirdek, ReLU
Maksimum havuzlama katmanı, $2 \times 2$ filtre
Normalleştirme katmanı, yığın normalleştirme
Seyreltme katmanı, 0,25 seyreltme oranı
Evrişim katmanı, 64 filtre, $3 \times 3$ çekirdek, ReLU
Maksimum havuzlama katmanı, $2 \times 2$ filtre
Normalleştirme katmanı, yığın normalleştirme
Seyreltme katmanı, 0,25 seyreltme oranı
Evrişim katmanı, 64 filtre, $3 \times 3$ çekirdek, ReLU
Maksimum havuzlama katmanı, $2 \times 2$ filtre
Normalleştirme katmanı, yığın normalleştirme
Seyreltme katmanı, 0,25 seyreltme oranı
Tam bağımlı katman, 256 nöron, ReLU
Tam bağımlı katman, 10 nöron, ReLU

Çizelge 3.5 Çalışmada önerilen TCNN konfigürasyonu

Inception V3 modeli
Tam bağımlı katman, 256 nöron, ReLU
Tam bağımlı katman, 10 nöron, Softmax

Veri setinin %70'i eğitim, %20'si doğrulama ve %10'u test verisi için kullanılmıştır. Kullanılan 3 model arasında TCNN %92'lik doğruluk başarımı ile daha iyi performans göstermiştir. Çalışmada aynı senaryoda oluşturulan görüntüleri kapsayan bir veri seti kullanılmıştır. Veri setindeki görüntü sayısının az olmasından transfer öğrenme tercih edilmiştir ve en çok başarı bu modelle sağlanmıştır.

Sevli ve Kemalöglu çalıřmalarında Türk İřaret Dili rakamlarından oluřan veri setindeki 2062 grnty sınıflandıran bir CNN modeli geliřtirmiřtir (Sevli ve Kemalglu 2020). Farklı optimizasyon teknikleri kullanılarak bu tekniklerin bařarıları sınıflandırılmıřtır. SGD, RMSProp, Adam ve Adamax kullanılan optimizasyon teknikleridir.

Çalıřmada geliřtirilen CNN modeli 16 katmanlıdır ve konfigrasyonu **çizelge 3.6** ile verilmiřtir.

Çizelge 3.6 Çalıřmada nerilen CNN modeli konfigrasyonu

Evriřim katmanı, 8 filtre, $5 \times 5$ çekirdek, ReLU
Maksimum havuzlama katmanı, $2 \times 2$ filtre
Seyreltme katmanı, 0,25 seyreltme oranı
Evriřim katmanı, 16 filtre, $3 \times 3$ çekirdek, ReLU
Maksimum havuzlama katmanı, $2 \times 2$ filtre
Seyreltme katmanı, 0,25 seyreltme oranı
Evriřim katmanı, 32 filtre, $3 \times 3$ çekirdek, ReLU
Maksimum havuzlama katmanı, $2 \times 2$ filtre
Seyreltme katmanı, 0,25 seyreltme oranı
Evriřim katmanı, 64 filtre, $3 \times 3$ çekirdek, ReLU
Maksimum havuzlama katmanı, $2 \times 2$ filtre
Seyreltme katmanı, 0,25 seyreltme oranı
Tam bağımlı katman, 128 nron, ReLU
Tam bağımlı katman, 64 nron, ReLU
Tam bağımlı katman, 10 nron, ReLU

Veri setinin %80'i eđitim, %20'si test verisi olarak ayrılmıřtır. Geliřtirilen CNN modeli farklı optimizasyon teknikleri ile drt kez eđitilmiřtir. Her eđitim iin devri sayısı 100 seilmiřtir. SGD optimizasyon tekniđinin kullanılmasıyla elde edilen eđitim dođruluk bařarımı %13, test dođruluk bařarımı %7,7'dir. Bu optimizasyon tekniđi ile olduka dřk bir sınıflandırma dođrulama bařarısı elde edilmiřtir. RMSProp optimizasyon tekniđinin kullanıldıđı durumda elde edilen eđitim dođruluk bařarımı %95,45 ve test



doğruluk başarımı %96,85'tir. Sınıflandırma başarısı SGD optimizasyon tekniğine göre oldukça fazladır. Adam optimizasyon tekniği kullanıldığında %98,42'lik eğitim ve %98,55'lik test doğruluk başarımları elde edilmiştir. Son olarak Adamax optimizasyon tekniğinin kullanıldığı eğitim aşamasında elde edilen eğitim doğruluk başarımı %89,81 ve test doğruluk başarımı %91,53'tür.

Çalışmada 4 farklı optimizasyon tekniği karşılaştırılmıştır ve en yüksek başarımları Adam optimizasyon tekniği ile elde edilmiştir. Çalışmada tek bir veri seti kullanılmıştır ve tek bir model üzerinden tüm optimizasyon teknikleri karşılaştırılmıştır. Farklı bir model daha eklenerek optimizasyon tekniklerinin her iki modelde de benzer davranışlar sergileyip sergilemediği analiz edilebilirdi.

#### 4. MATERYAL VE YÖNTEM

İşaret dili, farklı ülkelere ve hatta zaman zaman bir ülkenin içindeki farklı bölgelere göre değişiklik göstermektedir. Her işaret dili alfabesindeki harfler farklı bir şekilde tanımlanmaktadır. Bazı ülkelerin işaret dili alfabesi tek el ile statik tüm harfleri tanımlarken bazı ülkelerde bu harfler iki el kullanılarak veya dinamik bir şekilde tanımlanmaktadır. Ancak Hint İşaret Dili, Arap İşaret Dili gibi birkaç istisnai durum dışındaki işaret dillerindeki rakamlar aynı statik el hareketleriyle aktarılmaktadır. Tez çalışması kapsamında çoğu işaret dili için evrensel olan işaret dili rakamlarının derin öğrenme kullanarak sınıflandırılması amaçlanmaktadır.

İşaret dili tanıma problemi için literatürdeki çalışmalar genellikle bir veri seti kullanarak bir ya da birden fazla model önermekte ve bu modelleri tartışmaktadır. Kullanılan veri setindeki örnekler genellikle aynı senaryoda oluşturulmuş görüntülerdir. Yani görüntülerdeki ışıklandırma, arka plan, elin kameraya olan uzaklığı, görüntü boyutu gibi değişkenler aynıdır. Tez çalışması kapsamında farklı veri setlerinden toplanan görüntülerin bir araya getirilmesi ile hem çeşitliliğin hem de veri miktarının artırılması amaçlanmıştır.

Daha önce herhangi bir çalışmada kullanılmayan bir veri setinin oluşturulması ardından bir CNN modeli önerilerek bu veri setindeki örneklerin sınıflandırılması amaçlanmıştır. Bu amaç doğrultusunda farklı CNN modelleri oluşturulmuştur. Bu modeller oluşturulurken toplam katman sayısının artırılması, evrişim katmanındaki filtre sayısının ve çekirdek boyutunun değiştirilmesi, seyreltme katmanının eklenmesi, normalleştirme katmanının eklenmesi, tam bağımlı katman sayısının artırılması gibi değişikliklerin doğruluk başarımlarına etkileri gözlemlenmiştir. Ağ mimarisinde yapılan bu değişimlerin etkilerinin en iyi şekilde görülebildiği 3 CNN modeli tez çalışmasında sunulmuştur. Nihai olarak oluşturulan model en yüksek doğruluk başarımına sahip modeldir.

## 4.1 Veri Seti

Tez çalışması kapsamında kullanılan veri seti toplamda 6 veri setinin birleştirilmesi ile oluşturulmuştur. Veri setinde 0, 1, 2, 3, 4, 5, 6, 7, 8 ve 9 rakamlarının işaret dilindeki karşılıkları bulunmaktadır. Böylece çalışmada 10 sınıflı bir sınıflandırma problemi bulunmaktadır. Bu çalışma kapsamında oluşturulan veri seti, daha önce işaret dili rakam tanıma çalışmalarında kullanılmamıştır ve el görüntüleri açısından en fazla çeşitliliğe sahip veri setidir. Oluşturulan veri setinde bazı sınıflara ait görüntüler diğer sınıflardaki görüntülere göre daha azdır.

Kullanılan veri setlerine bazı ön işleme teknikleri uygulanarak sınıflandırma yapmadan önce görüntülerin birbirleri ile uyumlu olması sağlanmıştır. Nihai olarak toplamda 19653 görüntüden oluşan bir veri seti oluşturulmuştur ve Tez Veri Seti olarak adlandırılmıştır. Kullanılan veri setleri ve bu veri setlerine uygulanan ön işleme teknikleri sırayla aşağıdaki bölümlerde verilmektedir.

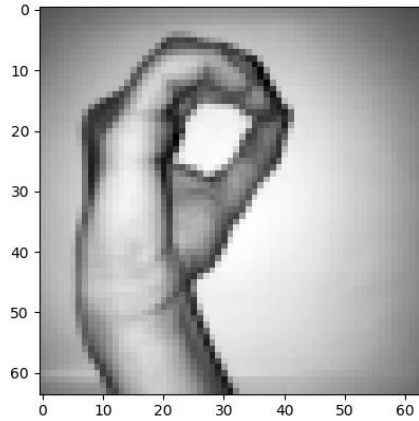
### 4.1.1 Veri Seti – 1

Tez çalışması kapsamında kullanılan ilk veri seti Ankara Ayrancı Anadolu Lisesi öğrencileri tarafından oluşturulmuştur. İşaret dili rakamları toplamda 218 öğrenci tarafından gerçekleştirilmiştir. Veri seti kullanıma açıktır ve indirilebilirdir (Mavi 2020). Veri setinde her bir öğrenciden alınan 0, 1, 2, 3, 4, 5, 6, 7, 8 ve 9 rakamlarının işaret dilindeki karşılıkları bulunmaktadır. Böylece bu veri seti tüm sınıflara ait örnekleri bulundurmaktadır. Veri setinde toplamda 2062 görüntü bulunmaktadır. Her bir sınıfa ait görüntü sayısı **çizelge 4.1** ile verilmiştir.

Sayısal görüntü 3 boyutlu nesnenin 2 boyutlu yüzey üzerine örneklenmiş halidir. 2 boyutlu bir dizi olarak aktarılan görüntünün en temel elemanı pikseldir. Her piksel ilgili konumdaki yeşinlik veya gri seviyesi bilgisini tutmaktadır. Veri setindeki görüntüler  $64 \times 64$  boyutundadır ve gri seviyeli görüntülerdir. Veri seti – 1'deki 0 sınıfına ait bir örnek **şekil 4.1** gösterilmiştir.

Çizelge 4.1 Veri Seti – 1’deki sınıflara göre örnek sayısı

Sınıf	Örnek Sayısı
0	205
1	206
2	206
3	206
4	207
5	207
6	207
7	206
8	208
9	204
<b>Toplam</b>	<b>2062</b>



Şekil 4.1 Veri seti – 1’deki 0 sınıfına ait bir örnek

Veri seti *numpy* kütüphanesine uygun *np*y formatındadır. Her sınıfa ait 5 farklı örnek **şekil 4.2** ile gösterilmiştir.



Şekil 4.2 Veri seti – 1’deki her bir sınıfa ait beş örnek görüntü

#### 4.1.2 Veri Seti – 2

Tez çalışması kapsamında kullanılan ikinci veri seti Amerikan İşaret Dili için hazırlanmış statik el hareketlerinden oluşan bir veri setidir. İşaret dili alfabesi ve rakamları toplamda 5 birey tarafından gerçekleştirilmiştir. Görüntüler oluşturulurken farklı ışıklandırma ve el hareketleri senaryoları kullanılmıştır. Veri seti toplamda 2425 görüntü içermektedir. Tez çalışması kapsamında işaret dili rakam görüntülerinin olduğu 700 görüntü alınmıştır ve Veri Seti – 2 olarak adlandırılmıştır. Veri seti kullanıma açıktır ve indirilebilirdir (Barczak vd. 2011)

Veri Seti – 2’de 0, 1, 2, 3, 4, 5, 6, 7, 8 ve 9 rakamlarının işaret dilindeki karşılıkları bulunmaktadır. Böylece bu veri seti tüm sınıflara ait örnekleri bulundurmaktadır. Her bir sınıfa ait görüntü sayısı **çizelge 4.2** ile verilmiştir.

Çizelge 4.2 Veri Seti – 2’deki sınıflara göre örnek sayısı

Sınıf	Örnek Sayısı
0	70
1	70
2	70
3	70
4	70
5	70
6	70
7	70
8	70
9	70
<b>Toplam</b>	700

Veri seti RGB görüntülerden oluşmaktadır. Veri setindeki görüntüler birbirlerinden farklı boyutlardadır ve *png* formatındadır. Her sınıfa ait 5 farklı örnek **şekil 4.3** ile gösterilmiştir.



Şekil 4.3 Veri seti – 2’deki her bir sınıfa ait beş örnek görüntü

Veri Seti – 2’deki görüntüler, siyah bir arka plandaki el hareketlerinden oluşmaktadır. Tez çalışması için oluşturulan veri setine uyarlamak her bir görüntüde elin bulunduğu kısım arka plandan ayrılarak beyaz bir arka plana yerleştirilmiştir. Bu adımda her bir görüntünün her bir piksel değeri için eşik seviyesi uygulanmış, bu eşik seviyesinin altındaki piksel değerlerine, beyaz arka plan sağlamak için 255 değeri atanmıştır. Ardından bu görüntülerin hepsi  $64 \times 64$  boyutuna ölçeklendirilmiş ve gri seviyeli görüntülere dönüştürülmüştür. Ön işleme teknikleri uygulanan görüntüler her bir sınıftan bir örnek verilecek şekilde **şekil 4.4** ile gösterilmiştir.



Şekil 4.4 Veri seti – 2’de ön işleme teknikleri uygulanmış örnek görüntüler

### 4.1.3 Veri Seti – 3

Tez çalışması kapsamında kullanılan üçüncü veri seti el hareketlerini sınıflandırmak için hazırlanmış statik görüntülerden oluşmaktadır. El hareketleri farklı bireyler tarafından gerçekleştirilmiştir. Veri Seti – 3’te 0, 1, 2, 3, 4 ve 5 rakamlarının işaret dilindeki karşılıkları bulunmaktadır. Diğer sınıflara ait görüntüler bulunmamaktadır. Veri setinde toplamda 1200 görüntü bulunmaktadır. Veri seti kullanıma açıktır ve indirilebilirdir (Anonymous 2019a). Her bir sınıfa ait görüntü sayısı **çizelge 4.3** ile verilmiştir.

Veri seti RGB görüntülerden oluşmaktadır. Veri setindeki görüntüler  $64 \times 64$  boyutlardadır ve *png* formatındadır. Her sınıfa ait 4 farklı örnek **şekil 4.5** ile gösterilmiştir.



Çizelge 4.3 Veri Seti – 3’teki sınıflara göre örnek sayısı

Sınıf	Örnek Sayısı
0	200
1	200
2	200
3	200
4	200
5	200
6	0
7	0
8	0
9	0
<b>Toplam</b>	1200



Şekil 4.4 Veri seti – 3’teki her bir sınıfa ait dört örnek görüntü

Veri Seti 3'teki görüntüler beyaz bir arka planda gerçekleştirilen el hareketlerinden oluşmaktadır. Böylece tez kapsamında oluşturulan veri setine uygun olduğundan herhangi bir arka plan değişikliği yapılmamıştır. Bu veri setindeki tüm görüntüler gri seviyeli görüntülere dönüştürülerek tez kapsamında oluşturulan veri setine eklenmiştir. Gri seviyeye dönüştürülen görüntülere örnekler **şekil 4.5** ile gösterilmiştir.



Şekil 4.5 Veri seti – 3'te ön işleme teknikleri uygulanmış örnek görüntüler

#### 4.1.4 Veri Seti – 4

Tez çalışması kapsamında kullanılan dördüncü veri seti Hint İşaret Dili için hazırlanmış statik el hareketlerinden oluşan bir veri setidir. Hint İşaret Dili, diğer işaret dillerine göre farklı bir yapıya sahiptir. 2 elin kullandığı birçok harf bulunmaktadır. Veri seti kullanıma açıktır ve indirilebilirdir (Anonymous 2020a). Bu veri setinden, tez kapsamında oluşturulan veri seti için uygun olan görüntüler alınarak Veri Seti 4 olarak adlandırılmıştır. 0, 1, 2, 3, 4, 5 ve 6 sınıflarına ait toplam 1997 görüntü dahil edilmiştir. Her bir sınıfa ait görüntü sayısı **çizelge 4.4** ile verilmiştir.

Veri seti RGB görüntülerden oluşmaktadır. Veri setindeki görüntüler birbirlerinden farklı boyutlardadır ve *png* formatındadır. Her sınıfa ait bir örnek görüntü **şekil 4.6** ile gösterilmiştir.

Çizelge 4.4 Veri Seti – 4’teki sınıflara göre örnek sayısı

Sınıf	Örnek Sayısı
0	300
1	300
2	296
3	273
4	288
5	295
6	245
7	0
8	0
9	0
<b>Toplam</b>	1997



Şekil 4.6 Veri seti – 4’teki her sınıfa ait bir örnek görüntü

Veri Seti 4’te 1, 2, 3, 4, 5 ve 6 sınıflarına ait görüntüler beyaz bir arka plan önünde gerçekleştirilmiştir ve tez kapsamında oluşturulan veri setine uygundur. Ancak 0 sınıfına ait görüntüler siyah bir arka planda gerçekleştirilmiştir. Tez çalışması için oluşturulan veri setine uyarlamak 0 sınıfındaki her bir görüntüde elin bulunduğu kısım arka plandan ayrılarak beyaz bir arka plana yerleştirilmiştir. Bu adımda her bir görüntünün her bir piksel değeri için eşik seviyesi uygulanmış, bu eşik seviyesinin altındaki piksel değerlerine, beyaz arka plan sağlamak için 255 değeri atanmıştır. Ardından bu görüntülerin hepsi  $64 \times 64$  boyutuna ölçeklendirilmiş ve gri seviyeli görüntülere dönüştürülmüştür. Ön işleme teknikleri uygulanan görüntüler her bir sınıftan bir örnek verilecek şekilde **şekil 4.7** ile gösterilmiştir.



Şekil 4.7 Veri seti – 4’te ön işleme teknikleri uygulanmış görüntü örnekleri

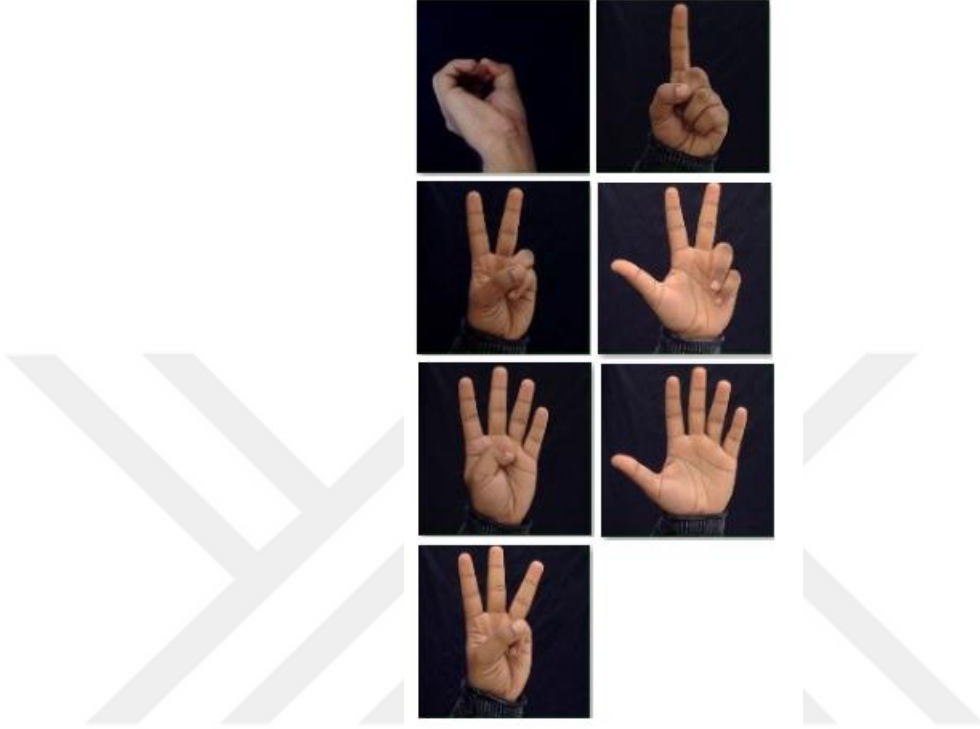
#### 4.1.5 Veri Seti – 5

Tez çalışması kapsamında kullanılan beşinci veri seti Hint İşaret Dili için hazırlanmış statik el hareketlerinden oluşan bir veri setidir. Veri setinde hem harf hem de rakam görüntüleri bulunmaktadır. Veri seti kullanıma açıktır ve indirilebilirdir (Anonymous 2020b). Bu veri setinden, tez kapsamında oluşturulan veri seti için uygun olan görüntüler alınarak Veri Seti 5 olarak adlandırılmıştır. 0, 1, 2, 3, 4, 5 ve 6 sınıflarına ait toplam 8400 görüntü dahil edilmiştir. Her bir sınıfa ait görüntü sayısı **çizelge 4.5** ile verilmiştir.

Çizelge 4.5 Veri Seti – 5’teki sınıflara göre örnek sayısı

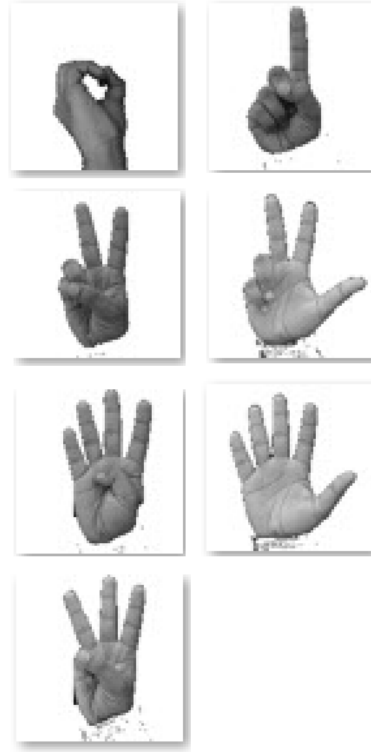
Sınıf	Örnek Sayısı
<b>0</b>	1200
<b>1</b>	1200
<b>2</b>	1200
<b>3</b>	1200
<b>4</b>	1200
<b>5</b>	1200
<b>6</b>	1200
<b>7</b>	0
<b>8</b>	0
<b>9</b>	0
<b>Toplam</b>	8400

Veri seti RGB görüntülerden oluşmaktadır. Veri setindeki görüntüler  $128 \times 128$  boyutundadır ve *jpg* formatındadır. Her sınıfa ait bir örnek görüntü **şekil 4.8** ile gösterilmiştir.



Şekil 4.8 Veri seti – 5'teki her bir sınıfa ait bir örnek görüntü

Veri Seti – 5'teki görüntüler siyah bir arka plan önündeki statik el hareketlerinden oluşmaktadır. Tez kapsamından kullanılan veri setine uygun hale getirmek için siyah arka plan önündeki elin sınırları belirlenerek elin olduğu kısmın görüntüden ayrılması sağlanmış ve beyaz düz bir arka plana yerleştirilmiştir. Bu adımda her bir görüntünün her bir piksel değeri için eşik seviyesi uygulanmış, bu eşik seviyesinin altındaki piksel değerlerine, beyaz arka plan sağlamak için 255 değeri atanmıştır. Ardından bu görüntülerin hepsi  $64 \times 64$  boyutuna ölçeklendirilmiş ve gri seviyeli görüntülere dönüştürülmüştür. Ön işleme teknikleri uygulanan görüntüler her bir sınıftan bir örnek verilecek şekilde **şekil 4.9** ile gösterilmiştir.



Şekil 4.9 Veri seti – 5’te ön işleme teknikleri uygulanmış görüntü örnekleri

#### 4.1.6 Veri Seti – 6

Tez çalışması kapsamında kullanılan altıncı veri seti statik el hareketlerinden oluşan veri setidir ve içerisinde birçok el hareketi görüntüsü bulundurmaktadır. Veri seti kullanıma açıktır ve indirilebilir (Anonymous 2019b). Bu veri setinden, tez kapsamında oluşturulan veri seti için uygun olan görüntüler alınarak Veri Seti 6 olarak adlandırılmıştır. 0, 1, 2, 3, 4, 5, 6, 7, 8 ve 9 rakamlarının işaret dilindeki karşılıkları bulunmaktadır. Böylece bu veri seti tüm sınıflara ait örnekleri bulundurmaktadır. Veri setinden toplamda 5294 görüntü tez için oluşturulan veri setine eklenmiştir. Her bir sınıfa ait görüntü sayısı **çizelge 4.2** ile verilmiştir.

Veri seti gri seviyeli görüntülerden oluşmaktadır. Veri setindeki görüntüler birbirlerinden farklı boyutlardadır ve *png* formatındadır. Her sınıfa ait iki örnek görüntü **şekil 4.10** ile gösterilmiştir.

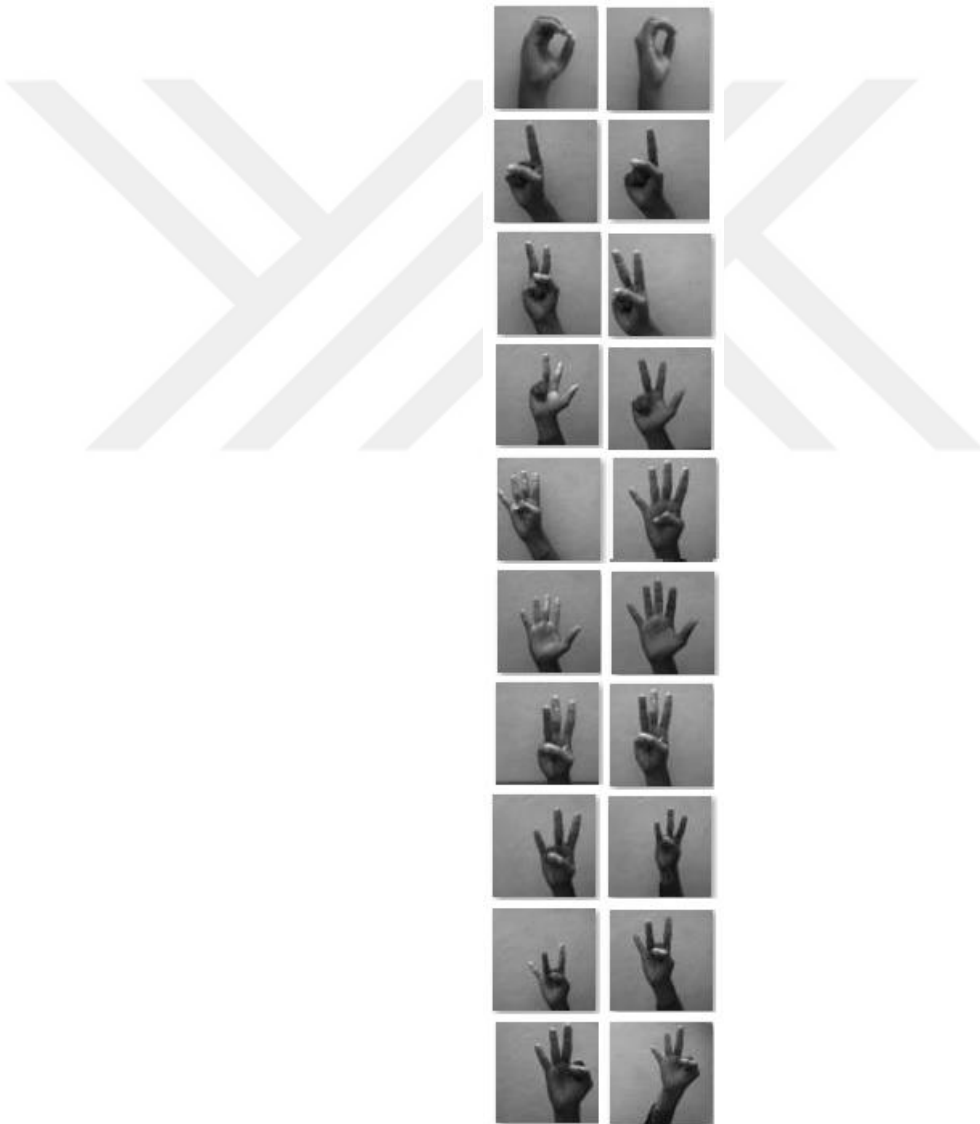
Çizelge 4.6 Veri Seti – 6’daki sınıflara göre örnek sayısı

Sınıf	Örnek Sayısı
0	500
1	500
2	500
3	500
4	500
5	523
6	318
7	778
8	449
9	726
<b>Toplam</b>	<b>5294</b>



Şekil 4.10 Veri seti – 6’daki her bir sınıfa ait beş örnek görüntü

Veri Seti – 6'daki görüntüler, aydınlık bir arka plandaki el hareketlerinden oluşmaktadır, böylece tez kapsamında kullanılan veri setine eklemek için görüntülerin arka planında herhangi bir değişiklik yapılmamıştır. Tez kapsamında oluşturulan veri setindeki görüntüler  $64 \times 64$  boyutundadır. Veri Seti – 6'daki görüntülerin genişlikleri yüksekliklerinden daha büyüktür. Bu görüntüleri oluşturulan tez çalışmasına uyarlamak için her bir görüntüde bulunan el kısmını almak üzere görüntüler kırpılmıştır ve  $64 \times 64$  boyutunda gri seviyeli görüntüler elde edilmiştir. Veri setine eklenen örnek görüntüler **şekil 4.11** ile gösterilmiştir.



Şekil 4.11 Veri seti – 11'de ön işleme teknikleri uygulanmış görüntü örnekleri



#### 4.1.7 Tez veri seti

Veri setlerindeki görüntülere ön işleme teknikleri uygulanarak tüm görüntülerin düz, aydınlık bir arka plana sahip olması sağlanmıştır. Aynı zamanda tüm görüntüler  $64 \times 64$  boyutuna dönüştürülmüştür. Oluşturulan veri setindeki sınıflara ait görüntü sayıları **çizelge 4.7** ile verilmiştir.

Çizelge 4.7 Oluşturulan veri setinde sınıflara göre örnek sayısı

Sınıf	Örnek Sayısı
0	2475
1	2476
2	2472
3	2449
4	2465
5	2495
6	2040
7	1054
8	727
9	1000
<b>Toplam</b>	19653

#### 4.2 Yazılım ve Donanım

Tez çalışması kapsamında kullanılan kütüphaneler Python programlama dili temellidir. Python nesne yönelimli, yüksek seviyeli ve genel amaçlı bir programlama dilidir. Tez çalışması kapsamında Python 3,7 versiyonu kullanılmıştır.

Tez çalışması kapsamında Anaconda platformu kullanılmıştır. Anaconda ücretsiz ve açık kaynaklı, veri bilimi ve benzeri bilimsel uygulamalar için Python ve R programlama dillerini kullanmak için hazırlanmış tümleşik bir dağıtımdır. Anaconda dağıtımı, grafiksel bir alternatif olarak bir grafiksel kullanıcı arayüzü (Graphical User Interface – GUI) olan Anaconda Navigator içerir. Anaconda Navigator, uygulamaları başlatmaya ve komut satırı komutlarını kullanmaya gerek kalmadan Anaconda paketlerini, ortamları ve kanalları kolayca yönetmeyi sağlar. Anaconda Navigator ile birçok program yönetilebilir, tez çalışması kapsamında Spyder kullanılmıştır. Python ile

yazılmış tümleşik geliştirme ortamı (Integrated Development Environment – IDE) olan Spyder, Python geliştirmek için kullanılır. Gelişmiş düzenleme, etkileşimli test, hata ayıklama ve iç gözlem özellikleriyle Python dili için güçlü bir etkileşimli ve açık kaynak geliştirme ortamıdır. Ek olarak Spyder IPython, NumPy, SciPy veya matplotlib gibi popüler Python kütüphanelerinin desteği sayesinde sayısal bir hesaplama ortamıdır.

Tez çalışması kapsamında veri setinin yüklenmesi, bazı sayısal hesaplamaların ve görselleştirmelerin yapılması için numpy, pandas, matplotlib ve seaborn hazır kütüphaneleri kullanılmıştır. Evrişimli sinir ağı modelleri ise *keras* kütüphanesi kullanılarak oluşturulmuştur. Keras, derin öğrenme algoritmaları için yazılmış ve yaygın bir biçimde kullanılan Python temelli bir kütüphanedir.

Derin sinir ağlarının eğitimi, birçok matris çarpımı içerdiğinden hesaplama açısından oldukça maliyetlidir. Modern CPU'lar bu tür hesaplamaları yapmak için yeterli değildir ve bu nedenle çok verimli değildir. Modern GPU'lar ise tam olarak bu işlemleri gerçekleştirmek için tasarlanmıştır. CUDA, Nvidia tarafından ortaya konulan, bilgisayarın işlem performansına yüksek oranda katkı yapan bir paralel programlama platformudur ve bu çalışma kapsamında kullanılmıştır. Temel olarak C, C++, C#, Fortran, Java, Python gibi programlama dilleri ile yazılmış algoritmaların GPU üzerinde çalışmasını sağlar. Kullanılan donanım konfigürasyonu **çizelge 4.8** ile verilmiştir. Kullanılan yazılım konfigürasyonu ise **çizelge 4.9** ile verilmiştir.

Çizelge 4.8 Donanım konfigürasyonu

GPU	NVIDIA GeForce 940MX 2GB
CPU	Intel Core i5 7200U

Çizelge 4.9 Yazılım konfigürasyonu

Keras	2.3.1
TensorFlow	2.1.0
CUDA	10.0
Python	3.7
İşletim Sistemi	Windows 10

### 4.3 Model Yapısı

Veri seti Numpy kütüphanesi kullanılarak yüklenir. Böylece girdi görüntüleri  $19653 \times 64 \times 64$  boyutundadır. Burada 19653 veri setindeki görüntü sayısıdır ve her bir görüntü  $64 \times 64$  boyutundadır. Bu matrisin elemanları 0 ile 1 arasındadır, böylece girdi görüntülerinin gri seviyeleri 0 ile 1 arasına normalleştirilmiştir. Yine Numpy kütüphanesi kullanılarak bu girdi görüntülerin hangi sınıfa ait olduğu bilgisini tutan etiket dizisi de yüklenir. Bu etiket dizisi  $19653 \times 10$  boyutundadır. Yine 19653 görüntü sayısı iken 10 sınıf sayısıdır. Her bir görüntü için hangi sınıfa ait ise sınıf sayısı indeksli eleman 1 iken diğer elemanlar 0'dır. Her bir sınıfa ait etiket vektörü **çizelge 4.10** ile verilmiştir.

Çizelge 4.10 Veri setindeki sınıflara karşılık etiket vektörleri

Sınıf	Etiket vektörü
0	1 0 0 0 0 0 0 0 0 0
1	0 1 0 0 0 0 0 0 0 0
2	0 0 1 0 0 0 0 0 0 0
3	0 0 0 1 0 0 0 0 0 0
4	0 0 0 0 1 0 0 0 0 0
5	0 0 0 0 0 1 0 0 0 0
6	0 0 0 0 0 0 1 0 0 0
7	0 0 0 0 0 0 0 1 0 0
8	0 0 0 0 0 0 0 0 1 0
9	0 0 0 0 0 0 0 0 0 1

Veri seti ve etiket dizisinin yüklenmesi ardından veri seti eğitim verisi, doğrulama verisi ve test verisi olarak rastgele olarak ayrılmaktadır. Veri setinin %70'i eğitim, %20'si doğrulama ve %10'u test veri seti olarak kullanılmıştır. Böylece veri setinden 13707 görüntü eğitim, 3980 görüntü doğrulama ve 1966 görüntü test için ayrılmıştır.

Tez çalışması kapsamında evrişimli sinir ağı modelleri için ileri beslemeli bir ağın modellenmesine uygun Keras sıralı (Sequential) modeli kullanılmıştır. Bir model, bu

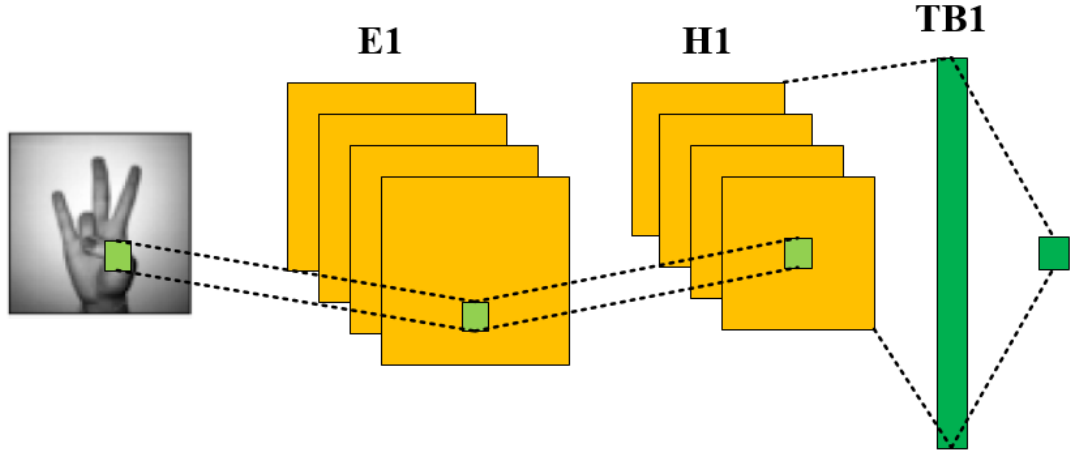
sıralı modelin çağırılması ile oluşturulur. Ardından sırasıyla modeldeki katmanlar tanımlanır. Tez çalışması kapsamında kullanılan katmanlar ve bu katmanların tanımlanması için gerekli parametreler bu bölümde tanımlanmıştır.

#### 4.4 Model Mimarileri

Bu bölümde oluşturulan evrişimli sinir ağlarının mimarileri verilmektedir.

##### 4.4.1 Model 1

Tez çalışması kapsamında ilk olarak 1 evrişim katmanı, 1 maksimum havuzlama katmanı ve 2 tam bağımlı katmandan oluşan bir sinir ağı kullanılmıştır. Kullanılan model **şekil 4.12** ile gösterilmiştir.



Şekil 4.12 Evrişimli sinir ağı Model 1

Model 1 için ilk evrişim (E1) katmanında 64 filtre kullanılmıştır ve çekirdek  $3 \times 3$  boyutundadır. Ardından  $2 \times 2$  boyutunda filtre ve  $2 \times 2$  adım uzunluğu ile maksimum havuzlama (H1) katmanı eklenmiştir. Son olarak 128 ve 10 nörondan oluşan tam bağımlı (TB1, TB2) katmanlar eklenmiştir. Model 1 konfigürasyonu **çizelge 4.11** ile verilmiştir.

Çizelge 4.11 Evrişimli sinir ağı Model 1 konfigürasyonu

Evrişim katmanı, 64 filtre, $3 \times 3$ çekirdek, ReLU aktivasyon, aynı doldurma
Maksimum havuzlama katmanı, $2 \times 2$ filtre, $2 \times 2$ adım uzunluğu
Tam bağımlı katman, 128 nöron, ReLU aktivasyon
Tam bağımlı katman, 10 nöron, Softmax aktivasyon

$64 \times 64$  boyutunda girdi görüntüsünün alındığı evrişim katmanında aynı doldurma uygulandığından oluşturulan öz nitelik haritası da  $64 \times 64$  boyutundadır. Kullanılan filtre sayısı 64 olduğundan evrişim katmanı sonucunda 64 öz nitelik haritası oluşturulur ve böylece çıktı  $64 \times 64 \times 64$  boyutundadır. Evrişim katmanında eğitilebilir parametre sayısı toplam ağırlık ve yanlılık değerleri sayısına eşittir.  $M$  çekirdek büyüklüğü,  $C$  girdinin kanal sayısı,  $F$  filtre sayısı olmak üzere ağırlık değeri sayısı, yanlılık değeri sayısı ve toplam eğitilebilir parametre sayısı sırasıyla Eşitlik (4.1), Eşitlik (4.2) ve Eşitlik (4.3) ile verilir.

$$W_C = M^2 CF \quad (4.1)$$

$$B_C = F \quad (4.2)$$

$$P_C = W_C + B_C \quad (4.3)$$

Evrişim katmanının ardından gelen, öz nitelik haritalarının alındığı havuzlama katmanında  $2 \times 2$  boyutunda filtre kullandığından her bir öz nitelik haritasına alt örnekleme uygulanarak  $32 \times 32$  boyutunda 64 öz nitelik haritası oluşturulur. Böylece havuzlama katmanında oluşturulan nihai çıktı  $32 \times 32 \times 64$  boyutundadır. Havuzlama katmanlarında sadece alt örnekleme yapıldığından eğitilebilir parametre sayısı sıfıra eşittir. Tam bağımlı katman öncesinde öz nitelik haritası tek boyutlu bir diziye dönüştürülür. Böylece ilk tam bağımlı katman öncesinde oluşturulan dizi  $1 \times 65536$  boyutundadır.  $G$  tam bağımlı katman girişinin büyüklüğü,  $N$  nöron sayısı olmak üzere

tam bağımlı katmanlarda toplam ağırlık değeri sayısı, yanlılık değeri sayısı ve toplam eğitilebilir parametre sayısı sırasıyla Eşitlik (4.4), Eşitlik (4.5) ve Eşitlik (4.6) ile verilir.

$$W_{TB} = GN \quad (4.4)$$

$$B_{TB} = N \quad (4.5)$$

$$P_{TB} = W_{TB} + B_{TB} \quad (4.6)$$

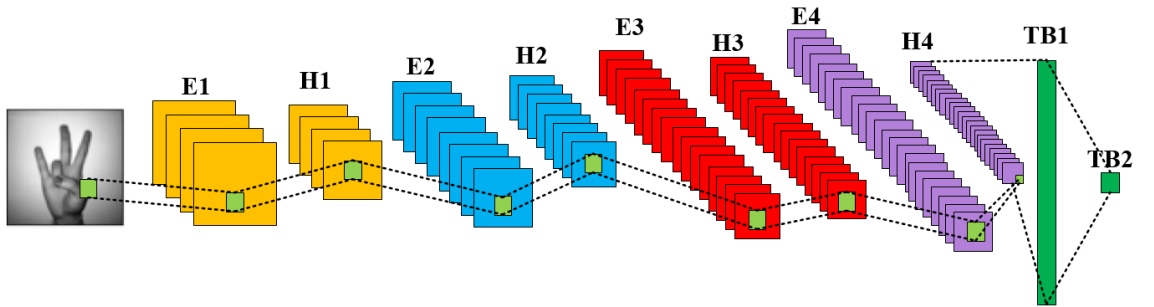
Model 1 katmanlarının çıktı boyutlarının ve eğitilebilir parametre sayılarının verildiği Model 1 özeti **çizelge 4.12** ile verilmiştir. Model 1 için toplam eğitilebilir parametre sayısı 8.390.666'dır.

Çizelge 4.12 Evrişimli sinir ağı Model 1 özeti

Katman	Çıktı boyutu	Parametre Sayısı
E1	$64 \times 64 \times 64$	640
H1	$32 \times 32 \times 64$	0
TB1	$1 \times 128$	8.388.736
TB2	$1 \times 10$	1.290
<b>Toplam Parametre Sayısı</b>		<b>8.390.666</b>

#### 4.4.2 Model 2

Model 1'deki evrişim ve havuzlama katman sayısının artırılmasıyla Model 2 oluşturulmuştur. Oluşturulan model **şekil 4.13** ile verilmiştir.



Şekil 4.13 Evrişimli sinir ağı Model 2

Model 2 için ilk evrişim (E1) katmanında 8 tane  $3 \times 3$  boyutunda çekirdek, ikinci evrişim (E2) katmanında 16 tane  $3 \times 3$  boyutunda çekirdek, üçüncü evrişim (E3) katmanında 32 tane  $3 \times 3$  boyutunda çekirdek ve dördüncü evrişim (E4) katmanında 64 tane  $3 \times 3$  çekirdek kullanılmıştır. Tüm evrişim katmanlarında sonra eklenen maksimum havuzlama (H1, H2, H3 ve H4) katmanlarında  $2 \times 2$  boyutunda filtre ve  $2 \times 2$  adım uzunluğu kullanılmıştır. Son olarak 128 ve 10 nörondan oluşan tam bağımlı (TB1, TB2) katmanlar eklenmiştir. Model 2 konfigürasyonu **çizelge 4.13** ile verilmiştir.

Çizelge 4.13 Evrişimli sinir ağı Model 2 konfigürasyonu

Evrişim katmanı, 8 filtre, $3 \times 3$ çekirdek, ReLU aktivasyon, aynı doldurma
Maksimum havuzlama katmanı, $2 \times 2$ filtre, $2 \times 2$ adım uzunluğu
Evrişim katmanı, 16 filtre, $3 \times 3$ çekirdek, ReLU aktivasyon, aynı doldurma
Maksimum havuzlama katmanı, $2 \times 2$ filtre, $2 \times 2$ adım uzunluğu
Evrişim katmanı, 32 filtre, $3 \times 3$ çekirdek, ReLU aktivasyon, aynı doldurma
Maksimum havuzlama katmanı, $2 \times 2$ filtre, $2 \times 2$ adım uzunluğu
Evrişim katmanı, 64 filtre, $3 \times 3$ çekirdek, ReLU aktivasyon, aynı doldurma
Maksimum havuzlama katmanı, $2 \times 2$ filtre, $2 \times 2$ adım uzunluğu
Tam bağımlı katman, 128 nöron, ReLU aktivasyon
Tam bağımlı katman, 10 nöron, Softmax aktivasyon

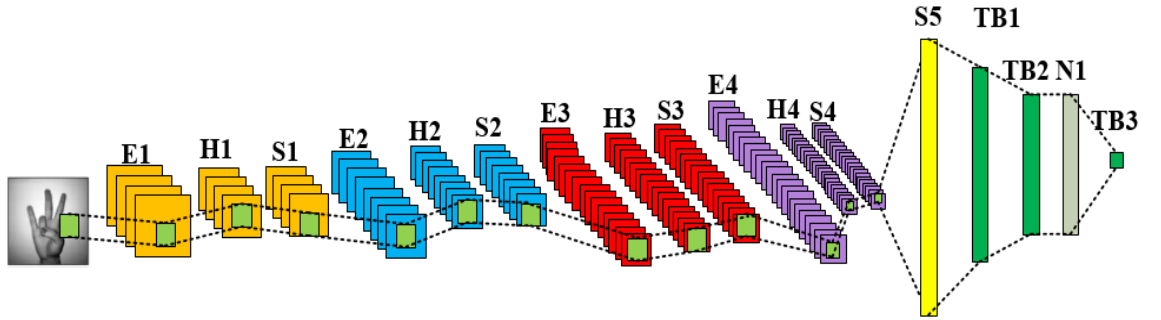
Evrişim katmanlarında aynı doldurma kullanıldığından oluşturulan öz nitelik haritalarının boyutları girdileri ile aynı boyuttadır. Maksimum havuzlama katmanlarından geçirilen öz nitelik haritalarının ise yükseklik ve genişlikleri yarıya inmektedir. Model 2 katmanlarının çıktı boyutlarının ve eğitilebilir parametre sayılarının verildiği Model 2 özeti **çizelge 4.14** ile verilmiştir. Model 2 için toplam eğitilebilir parametre sayısı 156.874'tür.

Çizelge 4.14 Evrişimli sinir ağı Model 2 özeti

Katman	Çıktı boyutu	Parametre Sayısı
E1	$64 \times 64 \times 8$	80
H1	$32 \times 32 \times 8$	0
E2	$32 \times 32 \times 16$	1168
H2	$16 \times 16 \times 16$	0
E3	$16 \times 16 \times 32$	4640
H3	$8 \times 8 \times 32$	0
E4	$8 \times 8 \times 64$	19.496
H4	$4 \times 4 \times 64$	0
TB1	$1 \times 128$	131.200
TB2	$1 \times 10$	1.290
<b>Toplam Parametre Sayısı</b>		<b>156.874</b>

#### 4.4.3 Model 3

Son olarak Model 2'deki havuzlama katmanlarının ardından seyreltme katmanları, bir tam bağlantı katman, bu katmanlardan önce bir seyreltme katmanı ve son tam bağımlı katmandan önce normalleştirme katmanı eklenerek Model 3 oluşturulmuştur. Model 3 mimarisi **şekil 4.14** ile verilmiştir.



Şekil 4.14 Evrişimli sinir ağı Model 3

Model 3 konfigürasyonu **çizelge 4.15** ile verilmiştir.



Çizelge 4.15 Evrişimli sinir ağı Model 3 konfigürasyonu

Evrişim katmanı, 8 filtre, 3×3 çekirdek, ReLU aktivasyon, aynı doldurma
Maksimum havuzlama katmanı, 2×2 filtre, 2×2 adım uzunluğu
Seyreltme katmanı, 0,25 seyreltme oranı
Evrişim katmanı, 16 filtre, 3×3 çekirdek, ReLU aktivasyon, aynı doldurma
Maksimum havuzlama katmanı, 2×2 filtre, 2×2 adım uzunluğu
Seyreltme katmanı, 0,25 seyreltme oranı
Evrişim katmanı, 32 filtre, 3×3 çekirdek, ReLU aktivasyon, aynı doldurma
Maksimum havuzlama katmanı, 2×2 filtre, 2×2 adım uzunluğu
Seyreltme katmanı, 0,25 seyreltme oranı
Evrişim katmanı, 64 filtre, 3×3 çekirdek, ReLU aktivasyon, aynı doldurma
Maksimum havuzlama katmanı, 2×2 filtre, 2×2 adım uzunluğu
Seyreltme katmanı, 0,25 seyreltme oranı
Seyreltme katmanı, 0,5 seyreltme oranı
Tam bağımlı katman, 256 nöron, ReLU aktivasyon
Tam bağımlı katman, 128 nöron, ReLU aktivasyon
Normalleştirme katmanı, yığın normalleştirme
Tam bağımlı katman, 10 nöron, Softmax aktivasyon

Evrişim katmanlarında aynı doldurma kullanıldığından oluşturulan öz nitelik haritalarının boyutları girdileri ile aynı boyuttur. Maksimum havuzlama katmanlarından geçirilen öz nitelik haritalarının ise yükseklik ve genişlikleri yarıya inmektedir. Havuzlama katmanlarından sonra gelen seyreltme katmanlarında ise boyutsal bir değişiklik olmamaktadır. Model 3 katmanlarının çıktı boyutlarının ve eğitilebilir parametre sayılarının verildiği Model 3 özeti **çizelge 4.16** ile verilmiştir. Model 3 için toplam parametre sayısı 321.226'dır.

Çizelge 4.16 Evrişimli sinir ağı Model 3 özeti

<b>Katman</b>	<b>Çıktı boyutu</b>	<b>Parametre Sayısı</b>
<b>E1</b>	64 × 64 × 8	80
<b>H1</b>	32 × 32 × 8	0
<b>S1</b>	32 × 32 × 8	0
<b>E2</b>	32 × 32 × 16	1168
<b>H2</b>	16 × 16 × 16	0
<b>S2</b>	16 × 16 × 16	0
<b>E3</b>	16 × 16 × 32	4640
<b>H3</b>	8 × 8 × 32	0
<b>S3</b>	8 × 8 × 32	0
<b>E4</b>	8 × 8 × 64	18.496
<b>H4</b>	4 × 4 × 64	0
<b>S4</b>	4 × 4 × 64	0
<b>S5</b>	1 × 1024	0
<b>TB1</b>	1 × 256	262.400
<b>TB2</b>	1 × 128	32.896
<b>N1</b>	1 × 128	512
<b>TB3</b>	1 × 10	1.290
<b>Toplam Parametre Sayısı</b>		321.226

## 5. ARAŞTIRMA BULGULARI

Oluşturulan 3 evrişimli sinir ağı modelleri için RMSProp optimizasyon yöntemi kullanılmıştır. Bu optimize edici tanımlanırken öğrenme oranı olarak 0,001 seçilmiştir. Modeller derlenirken maliyet fonksiyonu olarak Kategorik Çapraz Entropi fonksiyonu kullanılmıştır. Tüm modeller için devir sayısı 100 seçilmiştir.

Modellerin performans analizi için öncelikle eğitim ve doğrulama veri setleri için doğruluk değerleri hesaplanmıştır. Bu doğruluk değerleri, doğru tahmin edilen örneklerin veri setlerindeki örnek sayısına oranıdır. Ardından devir sayılarına karşılık bu doğruluk değerlerinin değişimleri çizdirilmiştir. Eğitim aşaması tamamlanan modeller kaydedildikten sonra test veri seti ile değerlendirilmiştir. Tüm modellerin test veri seti için hata matrisleri (confusion matrix) verilmiştir. Hata matrisi, örneklerin gerçek sınıfları ile tahmin edilen sınıflarını karşılaştırmaya yarayan bir tablodur. Son olarak tüm modeller için sınıflandırma raporu oluşturulmuştur. Bu raporda tüm sınıflar için kesinlik (*precision*), duyarlılık (*recall*) ve F1 Skor değerleri hesaplanmıştır. Doğruluk değeri, bir veri seti için toplam doğru tahmin sayısının yapılan toplam tahmin sayısına oranı olduğundan bir performans ölçüsü olarak, sınıflardaki örnek sayılarının eşit dağılımlı olmadığı problemler için tek başına yeterli değildir. Tahmin edilen sınıflar gerçek sınıflarla karşılaştırılarak doğru pozitif, yanlış pozitif, doğru negatif ve yanlış negatif olarak tanımlarından biri atanır (Hatibaruah vd. 2020). Kesinlik değeri pozitif tahmin edilen değerlerden gerçekte ne kadarının pozitif olduğunu göstermektedir. Duyarlılık, pozitif olarak tahmin edilmesi gereken örneklerin ne kadarının pozitif olarak tahmin edildiğini göstermektedir. F1 Skor değeri ise Kesinlik ve Duyarlılık değerlerinin harmonik ortalamasını göstermektedir. Kesinlik, duyarlılık ve F1 Skor değerleri Eşitlik (5.1), Eşitlik (5.2) ve Eşitlik (5.3) ile hesaplanır.

$$Kesinlik = \frac{Doğru\ pozitif}{Doğru\ pozitif + Yanlış\ pozitif} \quad (5.1)$$

$$Duyarlılık = \frac{Doğru\ pozitif}{Doğru\ pozitif + Yanlış\ negatif} \quad (5.2)$$

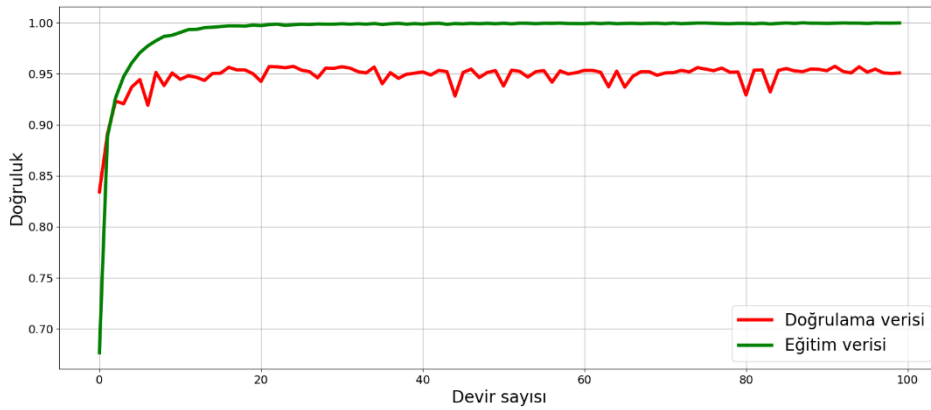
$$F1\ Score = \frac{2 \times Kesinlik \times Duyarluluk}{Kesinlik + Duyarluluk} \quad (5.3)$$

Kurulan evrişimli sinir ağı Model 1 için eğitim ve doğrulama verileri için hesaplanan doğruluk başarımı **çizelge 5.1** ile verilmiştir.

Çizelge 5.1 Model 1 için doğruluk başarımı

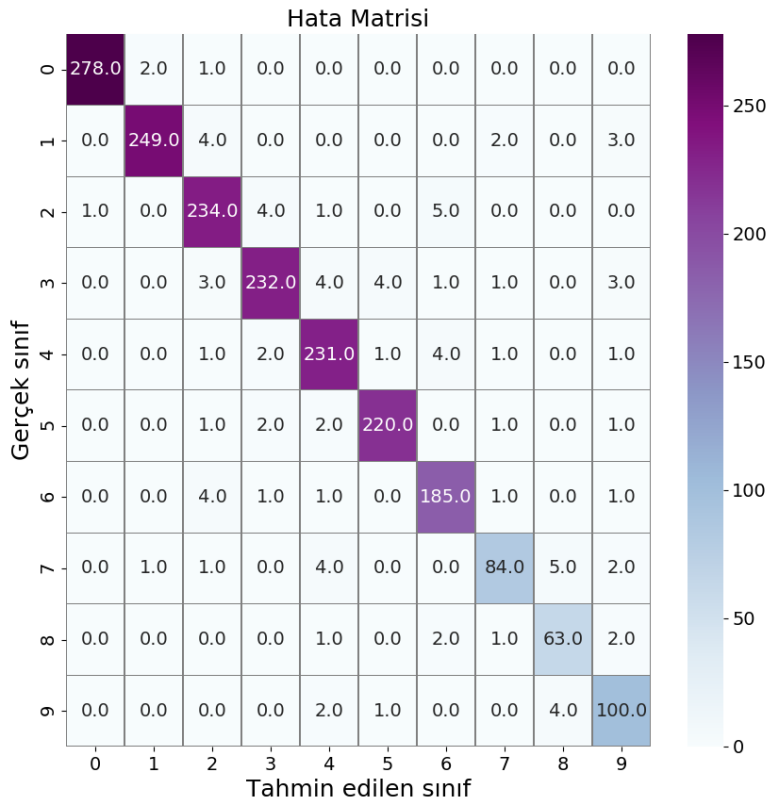
<b>Eğitim verisi</b>	%100
<b>Doğrulama verisi</b>	%95,1

Model 1 için devir sayısına karşılık doğruluk değerlerinin değişimleri **şekil 5.1** ile gösterilmiştir. Burada değişimler eğitim verisi için yeşil, doğrulama verisi için kırmızı ile gösterilmiştir. Hem eğitim hem de doğrulama verileri için devir sayısı arttıkça doğruluk değerleri artma eğilimindedir. Aynı zamanda devir sayısı arttıkça eğitim ve doğrulama verileri eğrileri arasındaki fark artmaktadır. Bu da aşırı uyum probleminin var olduğunu göstermektedir. Doğrulama verisi için doğruluk değerleri sürekli artış göstermemektedir ve ani değişkenlik göstermektedir. Bu da doğrulama sonuçlarının sağlamlığının çok yüksek olmadığını göstermektedir.



Şekil 5.1 Model 1 için doğruluk değerleri değişimi

Model 1 kullanılarak test verisi için hata matrisi **şekil 5.2** ile gösterilmiştir. Bu matriste gerçek sınıf değerleri solda ve tahmin edilen sınıf değerleri altta görülmektedir. Örneğin gerçek sınıf değeri 0 olup Model 1 ile tahmin edilen sınıf değeri 0 olan 278 örnek vardır. Böylece bu matrisin diyagonal elemanları doğru tahmin edilen örnek sayılarını göstermektedir. Model 1 ile 1966 test görüntüsünden 1876'sının sınıfı doğru tahmin edildiğinden test verisi için doğruluk başarımı %95,4'tür.



**Şekil 5.2 Model 1 için hata matrisi**

Test verisi için sınıflara ait kesinlik, duyarlılık ve F1 skor değerleri **çizelge 5.2** ile verilmiştir. Örnek sayısının diğer sınıflara göre görece az olduğu 7, 8 ve 9 sınıfları diğer sınıflara göre daha düşük kesinlik, duyarlılık ve F1 skor değerlerine sahiptir.

Çizelge 5.2 Model 1 için kesinlik, duyarlılık ve F1 skor değerleri

	0	1	2	3	4	5	6	7	8	9
<b>Kesinlik</b>	0,996	0,988	0,939	0,962	0,939	0,973	0,939	0,923	0,875	0,884
<b>Duyarlılık</b>	0,989	0,965	0,955	0,935	0,958	0,969	0,958	0,865	0,913	0,934
<b>F1 Skor</b>	0,992	0,976	0,947	0,948	0,948	0,971	0,948	0,893	0,893	0,909

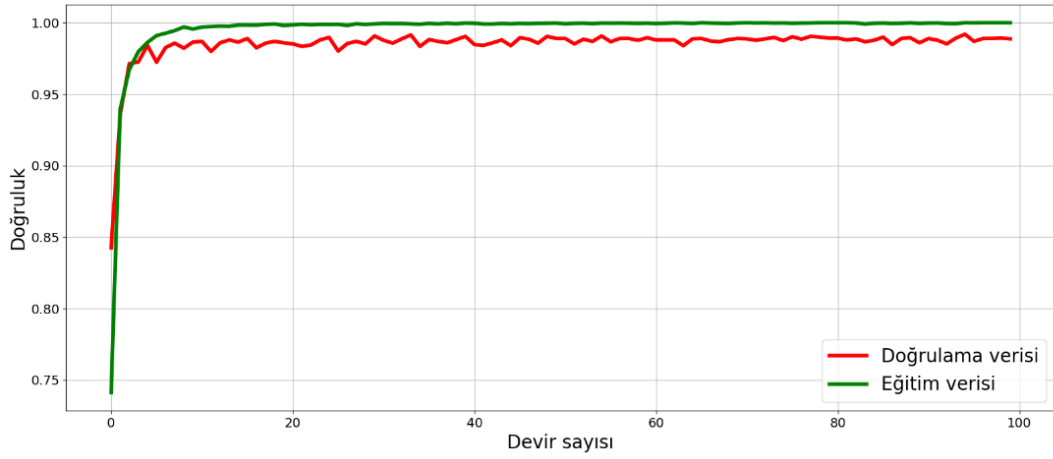
Model 1'deki evrişim ve havuzlama katman sayısının artırılmasıyla oluşturulan Model 2 için eğitim ve doğrulama verileri için hesaplanan doğruluk başarımı **çizelge 5.3** ile verilmiştir. Model 1 ile karşılaştırıldığında katman sayısının artırılması ile doğrulama verisi için doğruluk başarımı oldukça artmıştır. Aynı zamanda eğitim verisi ve doğrulama verisindeki doğruluk başarımları arasındaki fark da azalmıştır.

Çizelge 5.3 Model 2 için doğruluk başarımı

<b>Eğitim verisi</b>	% 100
<b>Doğrulama verisi</b>	%98,9

Model 2 için devir sayısına karşılık doğruluk değerlerinin değişimleri **şekil 5.3** ile gösterilmiştir. Model 1 ile benzer şekilde hem eğitim hem de doğrulama verileri için devir sayısı arttıkça doğruluk değerleri artma eğilimindedir. Model 1 ile karşılaştırıldığında eğitim ve doğrulama verileri arasındaki fark azalmıştır. Yani Model 2 ile daha az aşırı uyum gözlemlenmektedir. Aynı zamanda doğrulama eğrisindeki ani değişiklikler de Model 1'deki duruma göre daha azdır. Bu da doğrulama sonuçlarının sağlamlığını artırdığını göstermektedir.

Model 2 kullanılarak test verisi için hata matrisi **şekil 5.4** ile gösterilmiştir. Hata matrisi incelendiğinde 1966 test görüntüsünden 1942'sinin doğru sınıflandırıldığı görülmektedir. Böylece test verisi için doğruluk başarımı %98,8'dir. Model 1'e göre başarımları oldukça artmıştır.



Şekil 5.3 Model 2 için doğruluk değerleri değişimi



Şekil 5.4 Model 2 için hata matrisi

Test verisi için sınıflara ait kesinlik, duyarlılık ve F1 skor değerleri çizelge 5.4 ile verilmiştir. Model 1'e göre sınıflara ait değerler artış göstermiştir.

Çizelge 5.4 Model 2 için kesinlik, duyarlılık ve F1 skor değerleri

	0	1	2	3	4	5	6	7	8	9
<b>Kesinlik</b>	0.989	0.977	0.988	0.988	0.996	1	0.985	0.970	0.985	0.991
<b>Duyarlılık</b>	0.989	0.996	0.988	0.972	0.992	0.991	0.995	0.990	0.957	0.991
<b>F1 Skor</b>	0.989	0.987	0.988	0.98	0.994	0.996	0.99	0.98	0.971	0.991

Son olarak Model 2'deki havuzlama katmanlarının ardından seyreltme katmanları, bir tam bağlantı katman, bu katmanlardan önce bir seyreltme katmanı ve son tam bağımlı katmandan önce normalleştirme katmanı eklenerek oluşturulan Model 3 için eğitim ve doğrulama verileri için doğruluk başarımları **çizelge 5.5** ile verilmiştir. Modeller arasında doğrulama verisi için en yüksek doğruluk başarımları Model 3 ile elde edilmiştir.

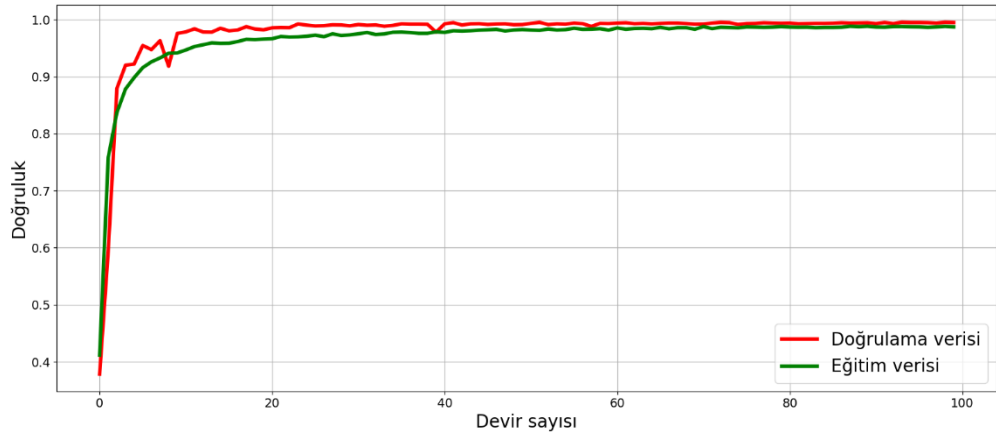
Çizelge 5.5 Model 3 için doğruluk başarımları

<b>Eğitim verisi</b>	%100
<b>Doğrulama verisi</b>	%99,5

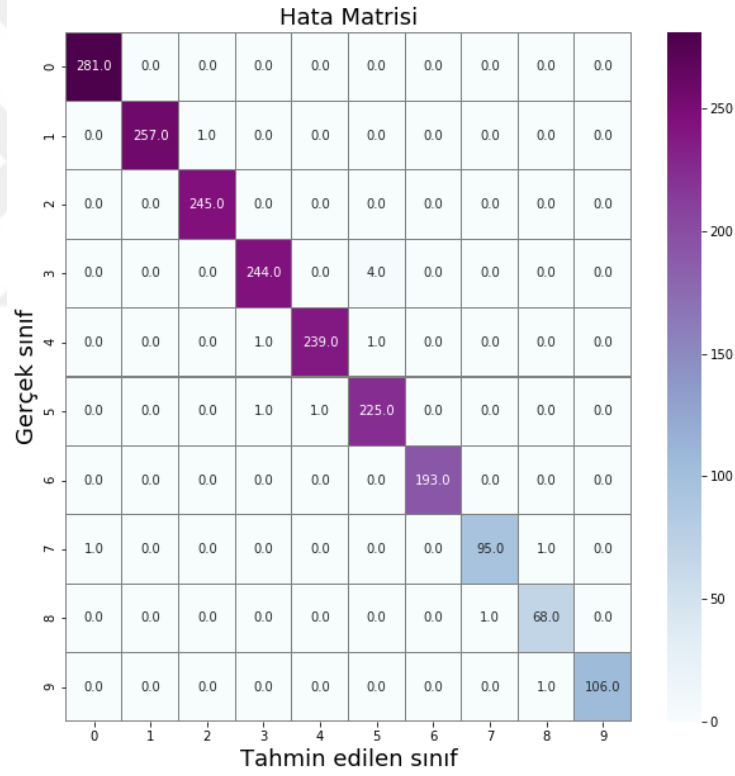
Model 3 için devir sayısına karşılık doğruluk değerlerinin değişimi **şekil 5.5** ile gösterilmiştir. Devir sayısı arttıkça eğitim ve doğrulama eğrileri arasındaki fark oldukça azalmıştır. Model 1 ve Model 2 ile karşılaştırıldığında aşırı uyum probleminin ortadan kalktığı gözlemlenmektedir. Ayrıca doğrulama eğrisindeki ani değişiklikler oldukça azalmıştır.

Model 3 kullanılarak test verisi için hata matrisi **şekil 5.6** ile gösterilmiştir. Bu model ile 1966 test görüntüsünden 1953'ü doğru sınıflandırılmıştır. Böylece test verisi için doğruluk başarımları %99,3'tür.





Şekil 5.5 Model 3 için doğruluk değerleri değişimi



Şekil 5.6 Model 3 için hata matrisi

Test verisi için sınıflara ait kesinlik, duyarlılık ve F1 skor değerleri çizelge 5.6 ile verilmiştir. Model 2'ye göre sınıflara ait değerler artış göstermiştir.

Çizelge 5.6 Model 3 için kesinlik, duyarlılık ve F1 skor değerleri

	0	1	2	3	4	5	6	7	8	9
Kesinlik	0,997	1	0,996	0,992	0,996	0,979	1	0,990	0,972	1
Duyarlılık	1	0,997	1	0,984	0,992	0,992	1	0,979	0,985	0,991
F1 Skor	0,998	0,998	0,998	0,988	0,994	0,985	1	0,985	0,979	0,996



## 6. SONUÇ VE DEĞERLENDİRME

Tez çalışması kapsamında literatürdeki çalışmalardan farklı olarak tek bir senaryoda elde edilen görüntülerden oluşan bir veri seti kullanmak yerine farklı senaryolarda oluşturulmuş veri setlerinin bir araya getirilmesi ile oluşturulan yeni bir veri seti oluşturulmuştur. Böylece ilk defa bu çalışmada kullanılan bu veri seti farklı ışıklandırma, düz ve aydınlık olacak şekilde farklı arka planda, farklı konumlandırma koşullarında, ten rengi gibi farklı el yapısına sahip bireylerin hem sağ hem de sol elleri ile gerçekleştirdikleri görüntüleri içermektedir. Veri setindeki görüntü çeşitliliğinin artırılması yanında toplamda 19653 görüntü kullanılarak literatürdeki işaret dili tanıma problemi çalışmalarından birçoğuna kıyasla çok daha büyük bir veri seti ile çalışma gerçekleştirilmiştir.

Veri setindeki bazı görüntülere gerekli ön işleme adımlarının uygulanmasından sonra sınıflandırma problemi için derin öğrenme mimarilerinden evrişimli sinir ağları tercih edilmiştir. Sınıflandırma problemi için basit bir evrişimli sinir ağı modelinden başlanarak katman sayısının artışının, başka katmanların eklenmesinin, ağ parametrelerinin değiştirilmesinin başarı oranına etkisi analiz edilmiştir. Tez çalışması kapsamında 3 model sunulmuştur. Bu modeller Model 1, Model 2 ve Model 3 olarak adlandırılmıştır.

Model 1 mimarisi, 1 evrişim ve 1 havuzlama katmanı ardından 2 tam bağımlı katmandan oluşan en basit mimaridir. Model 1 ile eğitimin tamamlanması ardından eğitim ve doğrulama verileri için doğruluk başarımları incelendiğinde çok basit bir model olmasına rağmen oldukça yüksek olduğu görülmüştür. Eğitim ve doğrulama verileri için devir sayısına karşılık doğruluk değerlerinin değişimi incelendiğinde 2 eğri arasında bir açıklık gözlemlenmiştir. Bu Model 1 için aşırı uyum probleminin olduğunu göstermektedir. Ayrıca doğrulama eğrisinde ani değişiklikler görüldüğünden doğrulama sonuçlarının sağlamlığı düşüktür. Model 1'in eğitim süresi diğer modellere kıyasla oldukça fazla sürmüştür, bunun nedeni eğitilebilir parametre sayısının diğer modellere kıyasla oldukça fazla olmasıdır. Ancak veri setinin büyüklüğünden test verisi için %95,4'lük bir doğruluk elde edilmiştir.

Model 1'deki katman sayısının artırılmasıyla oluşturulan Model 2'nin eğitim aşamasında elde edilen doğruluk başarımında artış olduğu gözlemlenmiştir. Devir sayısına karşılık doğruluk değerleri grafiği incelendiğinde Model 1'e göre oldukça iyileşme olduğu görülmüştür. Eğitim ve doğrulama eğrileri arasındaki fark oldukça azalmış ve doğrulama eğrisindeki ani değişiklikler azalmıştır. 1966 test görüntüsünden 1942'si doğru tahmin edilmiştir. Böylece Model 2 ile test verisi için %98,8'lik bir doğruluk başarımı elde edilmiştir.

Son olarak Model 2'ye eklenen katmanlar ile mimarinin daha da iyileştirilmesi amaçlanmıştır. Model 3 ile eğitim aşaması sonucunda eğitim verisi için %100'lük, doğrulama verisi için %99,5'lik doğruluk başarımları elde edilmiştir. Aynı zamanda eğitim ve doğrulama eğrileri arasındaki fark oldukça azalmıştır. Aynı zamanda doğrulama eğrisindeki ani değişiklikler yok denecek kadar azalmıştır. Model 3, diğer modeller arasında başarımları en yüksek, eğitim süresi en kısa olan modeldir.

Modellerin test verisi üzerindeki başarımlarının literatürdeki bazı çalışmalar ile karşılaştırılması **çizelge 6.1** ile verilmiştir. Veri setinin büyütülmesiyle nihai olarak önerilen Model 3 ile diğer çalışmalara kıyasla daha yüksek bir başarımları oranı elde edilmiştir.

Çizelge 6.1 Modellerin literatürdeki çalışmalarla karşılaştırılması

Model	Test Doğruluğu (%)
Mavi, 2020	97
Hossain, Adhikary ve Soheli, 2020	92
Sevli ve Kemaloğlu, 2020	98,55
<b>Model 1</b>	95,4
<b>Model 2</b>	98,8
<b>Model 3</b>	99,3

Son olarak tez çalışması kapsamında Mavi, Hossain ve arkadaşları, Sevli ve Kemaloğlu çalışmalarında önerilen mimariler tez veri seti ile eğitilmesi ardından test edilmiştir.

Test verisi üzerindeki başarımlar oranları **çizelge 6.2** ile verilmiştir. Çalışmalarda önerilen mimarilerin tez kapsamında oluşturulan veri seti ile eğitilmesi ile başarımlar oranları artmıştır. Bu durum oluşturulan veri setinin bu problem için başarılı bir veri seti olduğunu göstermektedir. Aynı zamanda tez çalışması kapsamında oluşturulan Model 3 halen diğer çalışmalarda önerilen modellere göre daha yüksek başarımlar oranına sahiptir.

Çizelge 6.2 Tez veri setinin önerilen diğer çalışmalardaki modellerde kullanılması ile çıkan başarımlar oranları

Kullanılan mimari	Test Doğruluğu (%)
Mavi, 2020	97,8
Hossain, Adhikary ve Soheli, 2020	99
Sevli ve Kemaloğlu, 2020	98,7

Kullanılan veri setindeki görüntülerde el hareketleri düz bir arka planda gerçekleştirilmektedir. Ancak gerçek hayatta arka plan çoğunlukla daha karmaşık bir yapıdadır. Daha karmaşık arka planda gerçekleştirilen veri setleri kullanmak veya oluşturmak gerçek hayattaki işaret dili tanıma problemine daha başarılı bir çözüm sağlayacaktır. Karmaşık arka plana sahip el hareket görüntüleri ham halde ağa verilebileceği gibi görüntülere bir ön işleme uygulanarak görüntüde el olan bölgenin arka plandan ayrılması ile sınıflandırılması da bir başka çözüm tekniği olacaktır. İşaret dili rakam tanıma probleminin bir ileriki adımı işaret dili alfabe tanımadır. Ancak bazı işaret dillerinde alfabedeki bazı harfler statik değil dinamik olarak tanımlanmaktadır. Gerçek hayattaki işaret dili tanıma problemleri de sürekli değişen görüntülerin tanınması ve geçmişte tanınan işaret ile anlam bütünlüğü sağlanması gerektiğinden video işleme yöntemlerinden ve farklı derin öğrenme mimarilerinden faydalanmak gerekecektir.

## KAYNAKLAR

- Albawi, S., Mohammed, T. A. ve Al-Zawi, S. 2017. Understanding of a convolutional neural network. In 2017 International Conference on Engineering and Technology (ICET) (pp. 1-6). IEEE.
- Anonymous. 2019a. Web Sitesi: [https://github.com/intelav/cnn-exploration/blob/master/sign\\_language\\_nn.py](https://github.com/intelav/cnn-exploration/blob/master/sign_language_nn.py) Erişim Tarihi: 16.02.2021
- Anonymous. 2019b. Web Sitesi: <https://www.kaggle.com/muhammadkhalid/sign-language-for-numbers/metadata> Erişim Tarihi: 01.03.2021
- Anonymous. 2020a. Web Sitesi: <https://www.kaggle.com/kartik2112/indian-sign-language-translation-letters-n-digits/metadata> Erişim Tarihi: 20.02.2021.
- Anonymous. 2020b. Web Sitesi: <https://www.kaggle.com/vaishnaviasonawane/indian-sign-language-dataset/metadata> Erişim Tarihi: 21.02.2021.
- Barczak, A. L. C., Reyes, N. H., Abastillas, A., Piccio, A. ve Susnjak, T. 2011. A new 2D static hand gesture colour image dataset for ASL gestures.
- Baydar, B. 2018. Convolutional Neural Network Based Brain MRI Segmentation. Yüksek Lisans Tezi, Orta Doğu Teknik Üniversitesi, Uygulamalı Bilimler Enstitüsü, Elektrik Elektronik Mühendisliği Bölümü, 117, Ankara.
- Bengio, Y. 2009. Learning Deep Architectures for AI. Now Publishers Inc.
- Copeland, J. 2015. Artificial intelligence: A philosophical introduction. John Wiley & Sons.
- Dao, H. 2020. Image classification using convolutional neural networks. Lisans tezi, Oulu University of Applied Sciences, Information Technology, 31, Finlandiya.
- Dev, D. 2017. Deep Learning with Hadoop. Packt Publishing Ltd.
- Doğan, F. ve Türkoğlu, İ. 2017. Derin Öğrenme Modelleri ve Uygulama Alanlarına İlişkin Bir Derleme. Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi, 10(2), 409-445.
- Fountas, Z. 2011. Spiking neural networks for human-like avatar control in a simulated environment. Computing Science of Imperial College London.
- Fukushima, K. ve Miyake, S. 1982. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In Competition and cooperation in neural nets (pp. 267-285). Springer, Berlin, Heidelberg.
- Goodfellow, I., Bengio, Y. ve Courville, A. 2016. Deep Learning. MIT Press, 802, Cambridge.

- Gummesson, A. 2016. Prostate Cancer Classification using Convolutional Neural Networks. Yüksek Lisans Tezi, Lund University, Centre for Mathematical Sciences, Faculty of Engineering, 54, İsveç.
- Hatibaruah, D., Talukdar, A. K., ve Sarma, K. K. 2020. A Static Hand Gesture Based Sign Language Recognition System using Convolutional Neural Networks. In 2020 IEEE 17th India Council International Conference (INDICON) (pp. 1-6). IEEE.
- Haykin, S. 2009. Neural Networks and Learning Machines. Prentice Hall, 938, Canada.
- Hosoe, H., Sako, S. ve Kwolek B. 2017. Recognition of JSL Finger Spelling Using Convolutional Neural Networks. 2017 Fifteenth IAPR International Conference on Vision Applications.
- Hossain, M. B., Adhikary, A., ve Soheli, S. J. 2020. Sign Language Digit Recognition Using Different Convolutional Neural Network Model. Asian Journal of Research in Computer Science, 16-24.
- International Bibliography of Sign Language, 2005; International Journal of Language & Communication Disorders, 2005.
- Janocha, K. ve Czarnecki, W. M. 2017. On loss functions for deep neural networks in classification. arXiv preprint arXiv:1702.05659.
- Kayaalp, K. ve Süzen, A. A. 2018. Derin Öğrenme ve Türkiye'deki Uygulamaları. Yayın Yeri: IKSAD International Publishing House, Basım sayısı, 1.
- Kızrak, M. A. ve Bolat, B. 2018. Derin öğrenme ile kalabalık analizi üzerine detaylı bir araştırma. Bilişim Teknolojileri Dergisi, 11(3), 263-286.
- Mavi, A. 2020. A New Dataset and Proposed Convolutional Neural Network Architecture for Classification of American Sign Language Digits. arXiv preprint arXiv:2011.08927.
- Nielsen, M. A. 2015. Neural networks and deep learning (Vol. 2018). CA: Determination press, 224, San Francisco.
- O'Shea, K. ve Nash, R. 2015. An introduction to convolutional neural networks. arXiv preprint arXiv:1511.08458.
- Öztemel, E. 2006. Yapay Sinir Ağları. Papatya Yayıncılık, 230, Türkiye.
- Sazlı, M. H. 2006. A brief review of feed-forward neural networks. Commun. Fac. Sci. Univ. Ank. Series A2-A3, V.50(1) pp 11-17
- Sevli, O., ve Kemaloğlu, N. 2020. Turkish Sign Language digits classification with CNN using different optimizers. International Advanced Researches and Engineering Journal, 4(3), 200-207.

- Sharma, S. 2017. Activation functions in neural networks. Towards Data Science, 6.
- Şeker, A. 2017. Derin öğrenme yöntemleri ve uygulamaları hakkında bir inceleme. Doktora Semineri, Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı, 49, İstanbul.
- Tan, Z. 2019. Derin Öğrenme Yardımıyla Araç Sınıflandırma. Yüksek Lisans Tezi, Fırat Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Bölümü, 75, Elâzığ.
- Yamashita, R., Nishio, M., Do, R. K. G. ve Togashi, K. 2018. Convolutional neural networks: an overview and application in radiology. Insights into imaging, 9(4), 611-629.
- Wang, Q., Ma, Y., Zhao, K. ve Tian, Y. 2020. A Comprehensive Survey of Loss Functions in Machine Learning. Annals of Data Science, 1-26.