T.C.
NECMETTİN ERBAKAN NİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

DERİN ÖĞRENME YÖNTEMİ İLE EL YAZISI TANIMA

Ayşe AYVACI ERDOĞAN

YÜKSEK LİSANS TEZİ

Bilgisayar Mühendisliği Anabilim Dalı

Temmuz-2021
KONYA

TEZ BİLDİRİMİ

Bu tezdeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

DECLARATION PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

İmza

Ayşe AYVACI ERDOĞAN

Tarih:

ÖZET

YÜKSEK LİSANS TEZİ

DERİN ÖĞRENME YÖNTEMİ İLE EL YAZISI TANIMA

Ayşe AYVACI ERDOĞAN

Necmettin Erbakan Üniversitesi Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Abdullah Erdal TÜMER

2021, 83 Sayfa

Jüri
Doç. Dr. Abdullah Erdal TÜMER
Prof. Dr. Sabri KOÇER
Dr. Öğr. Üyesi Vahit TONGUR

Günümüzde teknolojinin ilerlemesi, formlar ve dilekçeler gibi belgelerin bilgisayar ve dijital ortamda doldurulmasına neden oldu. Ancak bazı durumlarda, belgeler hala baskıda geleneksel tarzda korunmaktadır. Bununla birlikte, belgelerin farklı büyüklükleri (kapladığı yer) nedeniyle, depolanması, paylaşılması ve dosyalanması gibi bazı zorluklar bulunmaktadır. Bu nedenle, yazılı belgelerin dijital ortama taşınması büyük önem taşımaktadır. Bu ve benzeri nedenlerden dolayı, bu çalışma el yazısıyla yazılmış belgelerin sayısallaştırılmasına ilişkin metodolojileri inceleyerek Konvolüsyon Sinir Ağ yöntemi ile el yazılarının sayısallaştırılması amaçlanmıştır. Bunun için, görüntü formatına dönüştürülen belgeler görüntü işleme yöntemleri kullanılarak önceden işlenmiştir. Bu işlemler, belgenin satırlarını görüntü formatına bölmeyi, daha sonra karakterlere bölünen kelimelere bölmeyi ve son olarak karakterler üzerinde bir sınıflandırma işlemini içerir. Sınıflandırma aşamasında, derin öğrenme yöntemlerinden biri olan ve görüntü tanımada kullanılan Konvolüsyon Sinir Ağı yöntemi kullanılmıştır. Model, EMNIST veri kümesi kullanılarak ve eldeki belgelerden oluşturulan karakter veri kümesinde eğitilmiştir. Oluşturulan veri kümesi %88.72'lik bir başarı oranını yakalamıştır.

Anahtar Kelimeler: Derin Öğrenme, El Yazısı Tanıma, EMNIST, Görüntü İşleme, Karakter Tanıma, Konvolüsyon Sinir Ağı,

ABSTRACT

MS THESIS

DEEP LEARNING METHOD FOR HANDWRITING RECOGNITION

Ayşe AYVACI ERDOĞAN

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE OF NECMETTİN ERBAKAN UNIVERSITY
THE DEGREE O MASTER OF SCIENCE IN COMPUTER ENGINEERING

Advisor: Assoc. Prof. Dr. Abdullah Erdal TÜMER

2021, 83 Pages

Jury
Assoc. Doç. Dr. Abdullah Erdal TÜMER
Prof. Dr. Sabri KOÇER
Assist. Prof. Vahit TONGUR

The advancement of technology nowadays resulted into documents, such as forms and petitions, being filled out in computer and digital environment. Yet in some cases, documents are still preserved in traditional style, on print. Due to its distinct proportions, however, its storage, sharing and filing has become a complication. The relocation of these written documents to digital environment is therefore of great significance. In this view, this study aims to explore methodologies of digitizing handwritten documents. In this study, the documents converted to image format were pre-processed using image processing methods. These operations includes dividing lines of the document into image format, dividing into words which then divided into characters, and finally, a classification operation on the characters. As classification phase, one of the deep learning methods is the Convolution Neural Network method is used in image recognition. The model was trained using the EMNIST dataset, and in the character dataset created from the documents at hand. The dataset created had a success rate of 88.72.

Keywords: Character recognition, convolutional neural network, deep learning, handwriting recognition, EMNIST, image processing,

ÖNSÖZ

Tez çalışmam sırasında yaptığı katkı ve desteklerden dolayı danışmanım Necmettin Erbakan Üniversitesi Bilgisayar Mühendisliği öğretim üyesi Doç. Dr. Abdullah Erdal TÜMER'e teşekkür ederim.

Bu tezi hazırlamam sürecinde beni maddi ve manevi destekleyen aileme ve eşime sevgi ve saygılarımı sunarım.

Ayşe AYVACI ERDOĞAN
KONYA-2021

İÇİNDEKİLER

ŞEKİLLER DİZİNİ

SİMGELER VE KISALTMALAR

Kısaltmalar

CNN	Convolutional Neural Network(Evrişimli Sinir Ağları)
KNN	K-Nearest Neighbors (En Yakın Komşuluk)
SVM	Support Vector Machine(Destek Vektör Makineleri)
YSA	Yapay Sinir Ağları
RNN	Recurrent Neural Network (Tekrarlayan Sinir Ağı)
LSTM	Long / Short Term Memory (Uzun/ Kısa Süreli Bellek)
RBM	Restricted Boltzmann Machines (Kısıtlı Boltzmann Makineleri)
DBN	Deep Belief Networks (Derin İnanç Ağları)
AE	Autoencoders (Oto Kodlayıcılar)
DAE	Denoising Autoencoders (Derin Oto Kodlayıcılar)
Acc	Accuracy
Val	Valudation

1. GİRİŞ

Günümüzde karakter tanıma ile ilgili birçok çalışma bulunmaktadır. Bu çalışmalar genellikle optik karakter tanıma adı altında incelenmektedir. Optik karakter tanıma, basılı ve el yazısı metin karakterlerin tanınması ve bilgisayar tarafından düzenlenebilir bir metine çevrilmesidir. Optik karakter tanıma konusu basılı metinler üzerinde oldukça başarılı olmasına rağmen el yazısı ile yazılmış metinlerde başarı oranı daha düşüktür. El yazısı karakterleri normal karakterlere göre daha zor ayırt edilmektedir.

Günümüzde gelişen teknoloji ile birçok belge bilgisayar ortamında doldurulmaktadır. Ancak hala kâğıt üzerinde olan birçok işlem mevcuttur. Bunların bilgisayar ortamına aktarılması bir veya daha fazla kişinin görevlendirilmesi ile yapılmaktadır. Belgelerin otomatik olarak bilgisayar ortamına aktarılması için el yazısı tanıma sistemleri geliştirilmiştir. El yazısı tanıma, kâğıt, tablet ve telefon gibi ortamlara yazılan harf, rakam ve sembollerin bilgisayar sistemleri tarafından tanımlanıp anlamlı hale getirilmesidir.

El yazısı karakterleri optik karakterlere göre daha zor ayırt edilebilmektedir. Bu yüzden mevcut çalışmalardaki doğruluk oranları % 90'ın altındadır. El yazılarının tanınmasını etkileyen birçok varyasyon vardır. Bu varyasyonlar:

- *Doğal Varyasyonlar:* Yazma işlemi, kinestetik, bilişsel ve algısal motor bileşenlerin bir araya gelmesiyle oluşan bir olaydır. (Engel-Yeger, Nagauker-Yanuv, & Rosenblum, 2009). El yazısı her bireye özel bir yetenektir. Yazı, bir kişinin eğitim, kariyer ve iş hayatı boyunca düşüncelerini ifade etmelerini, iş yerindeki bilgileri kayıt etmelerini ve dönüştürmelerini sağlayan en önemli iletişim araçlarından birisidir. El yazısı, uzun zaman önce, insan bilgisini genişletmek ve iletişimi kolaylaştırmak için geliştirilmiş bir araçtır (Plamondon & Srihari, 2000). Her birey kendi özgü bir el yazısı tarzını geliştirir ve bireyin el yazısı ile yazdığı yazılar bir örnekten diğerine varyasyonlar gösterir. Karakterlerin şekilleri, yazının eğimi ve eğriliği, bir taban çizgisine uygunluk, yazma hızı, kalem basıncı, vuruş sırası ve stili yazının farklı olmasındaki doğal nedenlerdir. Yazara bağlı olarak da yazılar farklılık gösterir. Aynı yazara ait yazılan yazılarda bile farklılık olabilir. Bunların nedeni yukarıda belirtilen nedenlerdendir.

  Tüm yazı sistemlerinde genel varyasyonlara ek olarak, karakter sayısı, karakterler arasındaki benzerlik, allograflar, bitişik harfler ve bir karakterdeki vuruşların yazma sırası karakterlerin ayırt edilmesindeki diğer zorluklardır.

- *Kelime Boyutu:* Kelime dağarcığının büyüklüğü sistem tasarımı üzerinde doğrudan bir etkiye sahiptir. Küçük bir kelime boyutuna sahip kapalı bir kelime gövdesi, bu kelimeleri doğrudan tanıma aşamasında modellenebilir. Bununla birlikte, kelime boyutu arttıkça, tek tek kelimelerin modellenmesinin hesaplama karmaşıklığı nedeniyle kelimeler yerine harflerin modellenmesi tercih edilir.

  Açık kelime gövdeleri, küçük ve orta ölçekli kapalı kelime gövdelerinden daha zor olarak kabul edilmekle birlikte, kapalı kelime yaklaşımı, büyük kelime boyutları ile eşit derecede zor olabilir. Bu bağlamda, Türk Dili, kelime oluşumunda diğer dillere göre ek zorluklar getirmektedir. Türkçe, kök kelimelerin sonuna eklerin eklenmesiyle yeni kelimelerin oluşturulduğu bir dildir. Hangi soneklerin, hangi sırayla takip edebileceğini düzenleyen dil bilgisi kuralları vardır, ancak soneklerin eklenmesiyle oluşturulabilecek olası kelimelerin sayısı neredeyse sonsuzdur. Bu nedenle, sonsuz boyutlu bir kelime hazinesine sahiptir. Bu yüzden Türkçe kelimeleri tanımada, karakter tabanlı tanıma yapılması önerilir.
- *Görüntüdeki Kusurlar:* Belge üzerindeki yazıları tarayıcı ve kamera yardımı ile sayısallaştırıp bilgisayar ortamına aktarıldığında görüntü üzerinde çok fazla gürültü oluşur. Bu gürültüler kelime ve karakterlerin tanınması açısından zorluklara neden olabilir.

  Yine yazarın karakterleri birleşik bir şekilde yazması görüntüdeki karakterleri ayırt ederken gürültülü bir şekilde ortaya çıkmasına neden olur.

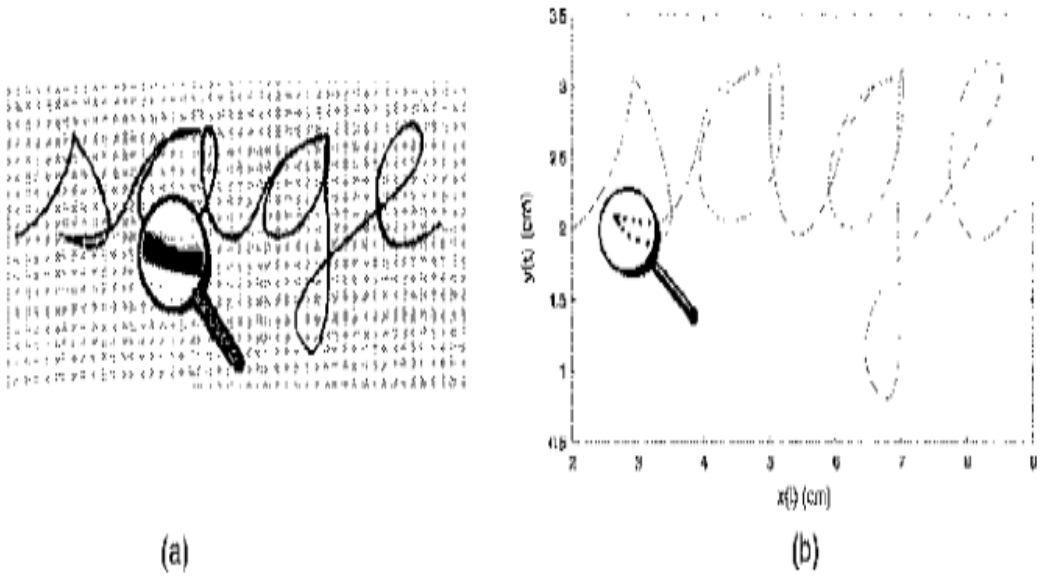
  Yazarın karakterleri birleşik bir şekilde yazması, karakterleri ayırırken görüntüde fazladan gürültüye neden olur. Bu da karakter tanımayı zor hale getirir.

Bu varyasyonlar kelime veya karakter tanımada zorluklara neden olurlar. Bu zorluklara rağmen birçok çalışma yapılmış ve bu alanda çalışmalar devam etmektedir.

Bununla birlikte Çince, Arapça ve Japonca gibi bazı dillerde, yapılan el yazısı tanıma sistemleri oldukça iyi düzeydedir. Fakat Türkçe el yazısı tanıma alanında yeterli düzeyde çalışma bulunmamaktadır. Yapılan çalışmalarda başarı seviyesi oldukça düşüktür.

El yazısı tanıma sistemlerinde iki yöntem vardır. Birincisi çevrimiçi yöntem (Şekil 1.1b) olarak adlandırılır. Bu yöntemde yazarın tablet veya özel ekranlara yazı yazarken vuruşlarının iki boyutlu koordinatları kullanılır. İkinci yöntem ise çevrimdışı yöntem (Şekil 1.1a) olarak adlandırılır. Bu yöntemde yazı resim olarak kullanılır. Çevrimiçi yöntemde görüntü, girişin uzamsal-zamansal bir temsili iken, çevrimdışı yöntemde görüntünün uzamsal parlaklığını içerir (Plamondon & Srihari, 2000). İki yöntemde de

ham veri depolama gereksinimleri farklıdır. Ortalama depolama gereksinimleri çevrimiçi yöntemde saniyede 100 örneklemede yüz bayt, çevrimdışı yöntemde ise inç başı 300 nokta da yüz kilo bayttır (Plamondon & Srihari, 2000).



Şekil 1.1. (a) Çevrimdışı el yazısı, (b) Çevrimiçi el yazısı

Tanınma oranlarına bakıldığında çevrimiçi yöntemler, çevrimdışı yöntemlere oranla daha yüksek bir tanıma oranına sahiptir. Çevrimdışı yöntemler 10,100 ve 1000'in üstü sözcükler için sırasıyla %95,%85 ve %78 tanıma oranına sahiptir (Plamondon & Srihari, 2000; Srihari, 1992). Çevrimiçi yöntemler ise daha büyük sözlükler için 21000 kelimede %80 doğruluk oranına sahiptir (Plamondon & Srihari, 2000; Seni, Srihari, & Nasrabadi, 1996).

Karakter tanıma teknolojilerinin sağladığı kolaylıklar, pek çok iş alanında kullanılabilmektedir:

- Banka çeklerin otomatik olarak taranıp gerekli hesap işlemlerinin elektronik ortamlarda gerçekleştirilmesi ve ilgili evrakların tasnif edilmesinde.
- Doktorların el yazılarının anlaşılır bir biçime getirilmesinde. Bu konuda yapılan bir çalışmada, doktorlardan alınan 500 reçete örneğini el yazısı tanıma ile doktorun el yazısından anahtar kelimeler belirlenmiştir. Bu belirlenen kelimeler ile doktorun yazdığı reçete raporunda istenilen ilacın ne olduğu anlaşılmıştır. Projede model olarak Hidden Markov Modeli kullanılmıştır. Performası arttırmak

için de çok katmanlı perceptron (MLP) uygulanmıştır (Roy, Bhunia, Das, Dhar, & Pal, 2017).

- Mektupların üzerinde yazan adreslerin posta koduna göre otomatik olarak tanınıp ayrıştırılması. CEDAR tarafından USPS (United States Postal Service) için yürütülen projede amaç postanelerde mektupların otomatik olarak gideceği adrese göre zarfların doğru şekilde ayırılmasıdır (Srihari, Cha, Arora, & Lee, 2001).
- Plaka tanıma sistemlerinde, araç plakalarının tanınıp otopark gibi yerlerde kullanılmasında.
- Elektronik kütüphaneler.
- SPAM mail engelleme.
- Yazılı metinleri farklı bir dile çevirme.
- İmza tanıma sistemleri.

Bu çalışmada çevrimdışı olarak el yazısı ile yazılmış metinlerin resim formatına çevrilerek üzerinde tanıma işlemi yapılmıştır. Türkçe sonsuz boyutlu bir kelime hazinesine sahip olduğu için, çalışmada karakter tabanlı tanıma yapılmıştır. Çalışmada literatürdeki çalışmalardan farklı olarak karakterlerin sadece kenar özelliklerine göre değil karakterin 2-boyutlu olarak tamamını kapsayan bir çalışma yapılmıştır. Bu işlemlerde, resimler üzerinde özellik çıkarmak için ayrı bir algoritma kullanılmamıştır. Ayrı bir algoritma yerine Konvolüsyon Sinir Ağları (Convolutional Neural Network-CNN) yöntemi ile hem karakterlere ait özellik çıkarma işlemi, hem de tanıma işlemi tek bir yöntem ile yapılmıştır. Çalışmada sırayla çevrimdışı olarak el yazı karakterleri üzerinde bir tanıma işlemi gerçekleştirilmiştir. Karakterlerin tanınması ve özelliklerinin çıkarılması için CNN yöntemi kullanılmıştır. Çalışma iki aşamadan oluşmaktadır;

Birinci aşama, tanıma için bir model oluşturup bu modelin eğitimin yapılması. Karakterlerin tanınması için model oluşturulmuş ve eğitimi yapılmıştır. Modelin eğitiminde EMNIST (NIST, 2017,April) veri seti kullanılmıştır. Çalışmada kullanılan veri seti Kaggle adlı siteden alınmıştır. Kaggle, Google LLC'nin bir yan kuruluşudur. Kaggle, veri bilimcilerden ve makine öğrenimi uygulayıcılardan oluşan çevrimiçi bir topluluktur. Bu topluluk sayesinde makine öğrenmesi alanında birçok model ve veri setinin depolamaktadır. EMNIST veri seti Kaggle sitesinden alınmıştır. EMNIST veri seti çok sayıda el yazısı ile yazılmış harf ve rakam içermektedir. CNN yöntemi ile EMNIST veri setindeki bütün karakterler 2-boyutlu olarak eğitime verilmiştir. Öncesinde

karakterler üzerinde bir özellik çıkarma işlemi yapılmadan veriler CNN modeline direk giriş olarak verilmiştir. Karakterlerin ayırt edici özellikleri modelin eğitimi sırasında CNN yöntemi ile yapılmış ve tanıma bu şekilde gerçekleştirilmiştir.

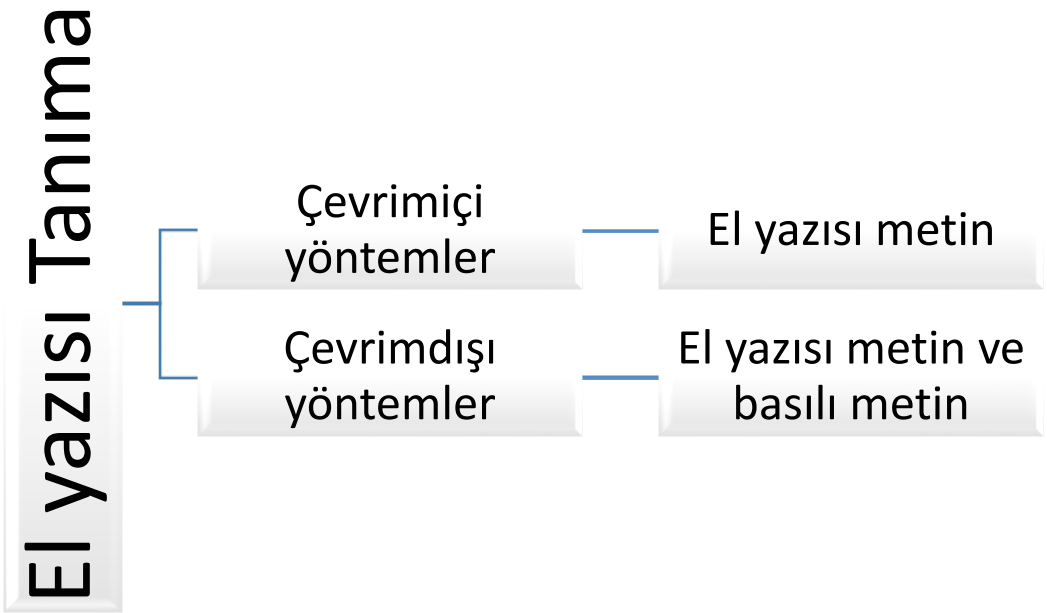
İkinci aşama, belgeler üzerinden karakterlerin elde edilmesi ve bu karakterler üzerinde tanıma işleminin yapılmasıdır. Tanıma işleminde kullanılan veriler çevremizdeki birçok kişi tarafından yazılan belgeler ile oluşturulmuştur. Daha sonra bu belgeler üzerinde bir ön işleme yapılarak karakterlerine ayrılma işlemi yapılmıştır. Ve sonrasında CNN yöntemi ile eğitilen model kullanılarak bir tanıma işlemi yapılmıştır.

2. KAYNAK ARAŞTIRMASI

Bu bölümde el yazısı tanıma ve derin öğrenme yöntemleri hakkında araştırma ve literatürdeki çalışmalar hakkında bilgi verilmiştir.

2.1. El Yazısı Tanıma

El yazısı tanıma, kâğıt belgeler, fotoğraflar, dokunmatik ekranlar gibi diğer cihazlarda el ile yazılmış yazıların anlaşılır bir hale dönüştürülmesidir. El yazısı tanımada Şekil 2.1. de göründüğü gibi çevrimiçi ve çevrimdışı el yazısı tanıma olarak iki ana başlıkta incelenir.

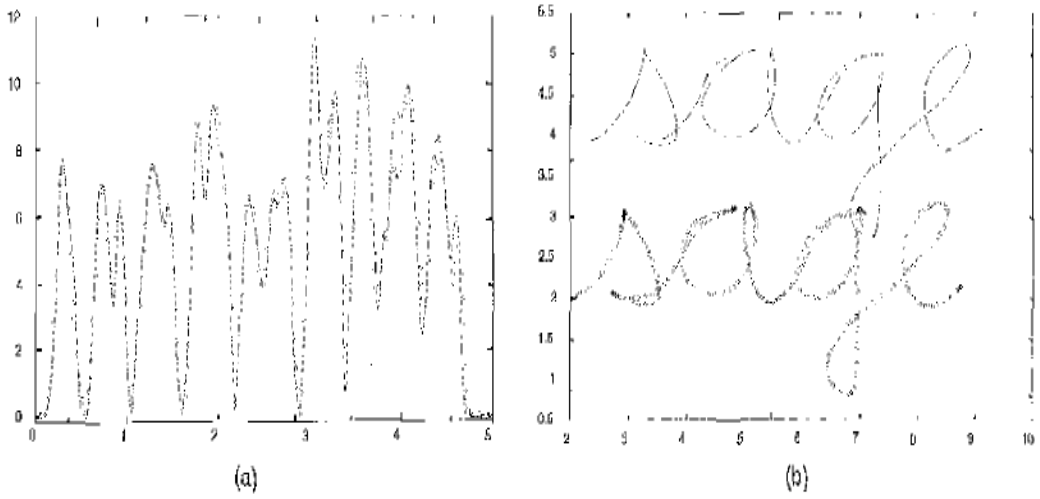


Şekil 2.1. El yazısı tanıma Kategorileri

2.1.1. Çevrimiçi Yöntemle El Yazısı Tanıma

Çevrimiçi yöntemle el yazısı tanıma (Online handwriting recognition - OHWR), tabletler veya özel ekranlar tarafından el veya kalem kullanılarak yazının otomatik olarak işlenmesiyle ilgili sistemlerdir. Bu sistemler kalem veya el hareketlerinin zamana göre hız, konum ve ivme bilgilerini barındırır. Bu bilgiler sayesinde bir tanıma işlemi gerçekleştirilir. OHWR' de kelime dağarcığının büyüklüğü sistemin performansında doğrudan bir etkiye sahiptir. Kelime sayısı arttıkça sistemin tanıma hızı ve performansı düşer (Bilgin Taşdemir, 2018). Şekil 2.2.'de çevrimiçi el yazısı tanıma örnekleri verilmiştir.

Bu sistem ile ilgili 1960'lı yılların başından günümüze kadar birçok yöntem ve yaklaşım bulunmuştur. Bu yöntemler üç sınıfta incelenebilir: Birinci sınıf bir ekran ve özel kalem ile yazılan metnin anlaşılması. İkinci sınıf el yazısı ile atılan imzalardan kişinin kimliğinin doğrulanması. Üçüncü sınıf eğitim ve rehabilitasyon amaçlı sistemler tasarlamak için el yazısının nöromotor özelliklerini kullanmaktır (Plamondon & Srihari, 2000).



Şekil 2.2. (a) Orginal kelimenin sinyal tablosu ,(b) çevrimiçi yazılmış bir kelime(Plamondon & Srihari, 2000)

2.1.1.1 Kalem Tabanlı Bilgisayarlar

Elektronik kalemler komut hareketlerinin ve el yazısı mesajlarının çevrimiçi tanımlanmasını sağlar. Çevrimiçi tanımada elde edilen sinyalleri tanımadan önce

gürültülerin azaltılması, izin yönlerini normalleştirmek ve sinyali anlamlı birimlere ayırmak için önceden bir işleme tabi tutulur. Gürültünün oluşmasında birçok neden vardır. Bunlar: sayısallaştırma işleminde ve el veya kalemin hareketlerinden kaynaklanan gürültüler olabilir. Gürültü azaltma yaklaşımları veriyi düzeltme ve sinyal filtreleme gibi işlemleri kapsar.

Tanıma işleminden önce yazılmış kelimenin normalleştirilmesi gerekir. Bu normalleştirme: taban çizgisinin düzeltilmesi, eğik yazıların düz yazıya çevrilmesi, karakter büyüklüklerinin ayarlanmasıdır.

Segmentasyon işlemi karakter tanımada temel yapıyı almak için kullanılır. İki yöntem kullanılmaktadır. Birinci yöntem, satır algılamadır (A Hennig, Sherkat, & Whitrow, 1996). Bu yöntem sayesinde kelime bölümleme ve metin içermeyen bölümlerin ayarlanması yapılır. İkinci yöntem ise girişin bireysel karakterlere hatta alt karakterlere bölünmesine odaklanır. Kelimelerin bölünmesiyle ilgili en büyük sorun, bireysel karakterlerin başlangıç ve bitişinin belirlenmesidir. Bunun için kullanılan en yaygın yaklaşımlar denetimsiz öğrenme (Hébert, Parizeau, & Ghazzali, 1999; Plamondon & Srihari, 2000) ve veriye dayalı bilgi tabanlı yöntemlerdir (Andreas Hennig, Sherkat, & Whitrow, 1997).

Sinyal vuruşları tanımlamak ve temsil etmek için çeşitli yöntemler önerilmiştir. Bu yöntemler maksimum eğrilik noktası segmentasyonu, ölçek uzayı yaklaşımı veya bileşen tabanlı yaklaşımıdır (Fujisaki, Chefalas, Kim, Tappert, & Wolf, 1991).

Daha sonrasında elde edilen kelimelerin veya karakterlerin sınıflandırılması işlemi yapılır. Sınıflandırma işlemi sırasında mesaj üretimindeki değişiklikleri unutmamak gerekir. Sınıflandırmayı etkileyen iki çeşit varyasyon vardır. Bunlar:
Geometrik Varyasyonlar: Bir mesaj üretirken vuruş koşullarına bağlı olarak konum, boyut, taban çizgisi yönelimi ve eğik olarak meydana gelen değişiklikleri ifade eder.
Allografik varyasyonlar: Farklı yazar popülasyonları tarafından tek bir karakterle ilişkilendirilen çeşitli modeller ile ilgilidir.

El yazısı vuruşlarının üretilme sırasındaki çeşitlilik de sorun kaynağı olabilir. Yazım hatalarının düzeltilmesi, kalemin kayması, harf ihmali veya ekleme, çevrimiçi tanıyıcıların görevini büyük ölçüde karmaşıklaştırır. Bu alanda el yazısı tanıma işlemleri iki sınıfta incelenebilir. Bunlar:

1) *Biçimsel, Yapısal ve Kural Tabanlı Yöntemler:* 1960 'larda kullanılan yöntemdir. Geniş bir veritabanının otomatikleştirilmesi zorunluğundan dolayı kullanılması bırakılmıştır. Son yıllarda bellirli özelliklerin kulllanımının artması sonucunda,

istatiksel bilgiler kullanılarak bulanık kuralların eklemesi yöntemi tekrar kullanılmaya başlanmıştır.

2) *İstatistiksel Sınıflandırma Yöntemleri*: Olasılık dağılımları ile farklı sınıfların tanımlandığı bir yöntemdir. Üç gruba ayrılır:

   a) *Açık Yöntemler:* Doğrudan türetilir veya dolaylı olarak lineer diskriminant analizinden, temel bileşen analizinden ve hiyerarşik küme analizinden dolayı matematiksel olarak iyi desteklenir.
   b) *Örtülü Yöntemler:* Yapay sinir ağlarına dayanan yöntemler anlamına gelir. Bu yöntemlerin sınıflandırma davranışı, eğitim veri setinin istatistiksel özellikleri ile tam olarak belirlenir.
   c) *Markov Modellemesi:* Gizli Markov Modelleme (HMM) süreci iki katına çıkaran bir yöntemdir. Bir vektör nicelleştirme algoritması kullanarak giriş özelliği vektörünün ayrı bir sembole dönüştürülmesini gerektirir. Sürgülü bir pencerede kontur şekilleri için bu sembollerin ortaya çıkma olasılıkları HMM algoritmasının temelini oluşturur. HMM algoritmasının amacı, belirli bir sınıfın meydana gelmesinin en muhtemel olma olasılığını bulmaktır (Plamondon & Srihari, 2000).

2.1.1.2. İmza Doğrulayıcılar

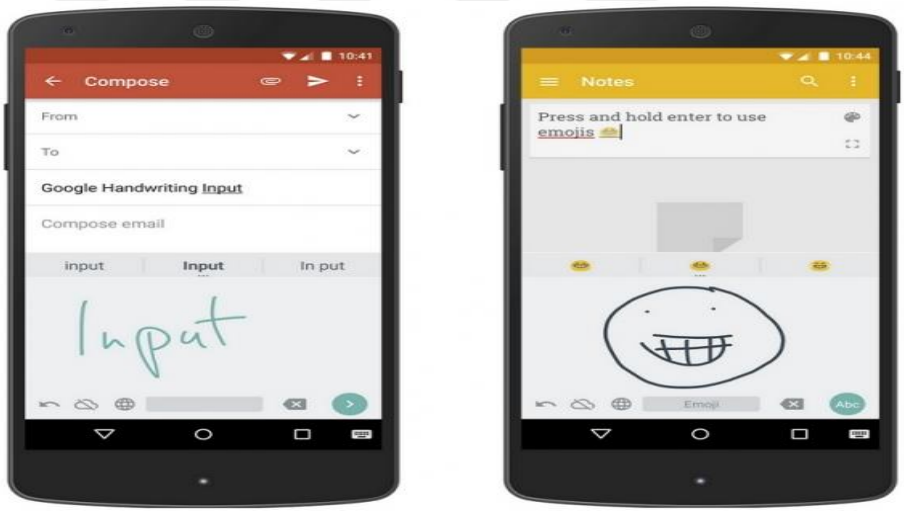
İmza doğrulayıcılar karakter tanımada çok farklı ve zor bir yöntemdir. İmza doğrulaması çoğunlukla yazının tekil, münhasır ve kişisel karakterinden faydalanmaya çalışır (Plamondon & Srihari, 2000). İmza doğrulama çifte zorluk sunar. Birincisi, imzalanmış olanın, yazılı olanı önemsemeden, bir bireyin benzersiz özelliklerine karşılık geldiğini doğrulamaktır. İkincisi, imzada sahteciliği ortaya çıkarmaktır. İmza doğrulayıcılarla bu iki zorluğun en aza indirgenmesi amaçlanmıştır.

İmza kişiye ait, kolayca değişmeyen ve uzun yıllar kullanılan bir yapıdır. Günümüzde imzanın elektronik ortamlarda kullanılması artmıştır. İmzanın tanınması için imza doğrulayıcı uygulamalar geliştirilmiştir.

2.1.1.3. Gelişim Araçları

El yazısı tanıma, insan-makine ilişkisini sağlar. İnsan-makine ilişki içinde çocukların karakterleri öğrenmesi, engelli kişilerin el yazısı ve çizim alıştırmalarını bir ara yüz yardımı ile kolay bir şekilde yapılmasına imkân verir. Bu alanda son yıllarda çocukların okuma ve yazmayı öğrenmeleri için birçok uygulama geliştirilmiştir. Çocuklar gördüğü karakterleri tablet ekranına ya da özel ekrana yazması ve sistem tarafından karakter kontrolünün sağlanması gibi birçok uygulama geliştirilmiştir.

Son yıllarda Android, ISO gibi mobil yazılımlar ile anlık olarak karakter tanıma uygulamaları geliştirilmeye başlanmıştır. Google, kullanıcıların el yazılarını klavye girişine dönüştüren yeni bir uygulama yayınlamıştır. 82 farklı dildeki baskı karakterlerini ve el yazısı şekillerini tanıyabilen uygulama, akıllı telefon kullanıcılarının daha hızlı bir şekilde yazı yazmalarına olanak tanımıştır. Şekil 2.3'te bu çalışmanın örneği verilmiştir.



Şekil 2.3. Google'nin el yazısı tanıma uygulaması

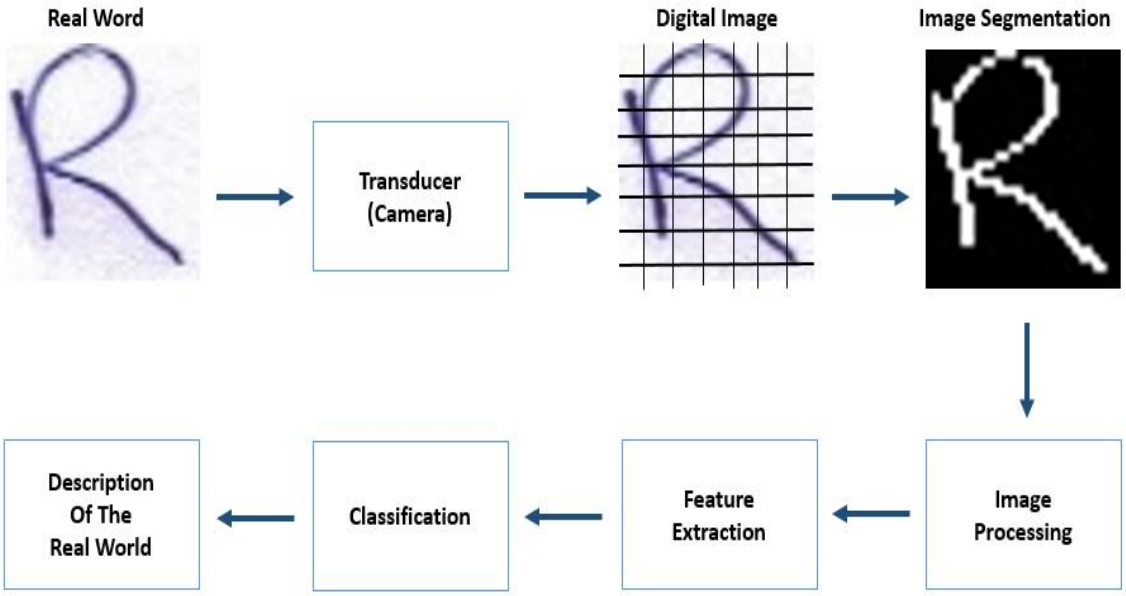
2.1.2. Çevrimdışı Yöntemle El Yazısı Tanıma

Çevrimdışı yöntemle el yazısı tanıma (Offline handwriting recognition), bir belge üzerindeki karakterlerin algılanmasını sağlayan sistemlerdir. Bu sistem kullanılırken önce belge bilgisayar ortamına atılarak sayısallaştırılır. Daha sonra belge sırasıyla paragraflara, cümlelere ve kelimelere bölünmesi sağlanır (Şekerci, 2007). Çevrimdışı olarak el yazısı tanıma bütünsel veya analitik strateji olarak ikiye ayrılır.

Bütünsel Strateji: Kelimenin tam anlamıyla tanınmasıdır. Dilimleme işlemi yapmadan direk olarak kelime üzerinde işlem yapılır. Kelimenin uzunluğu arttıkça tanınma oranı düşer. Kelime sadece bir ön işlemden geçirilerek görüntü üzerindeki gürültüler yok edilir. Sonra direk olarak kelime tabanlı bir tanıma işlemi yapılır.

Analitik Strateji: Belgenin paragraf, cümle, kelime, karakter olarak dilimlenmiş hali üzerinde tanınma işlemin uygulanmasıdır. Kelime tanıma işlemine geçmeden önce belirli adımlardan geçer ve en küçük yapı olan karakter üzerinden tanıma işlemi yapılır. Şekil 2.4'de çevrimdışı olarak karakter tanımanın adımları gösterilmiştir. Karakter tanıma da çevrimdışı yöntemler belirli bir işlem sırası izler. Bunlar (Srihari et al., 2001):

- Ön-işleme
- Dilimleme
- Öznitelik çıkarımı
- Sınıflandırma
- Son-işlem



Şekil 2.4. Çevrimdışı olarak karakter tanıma

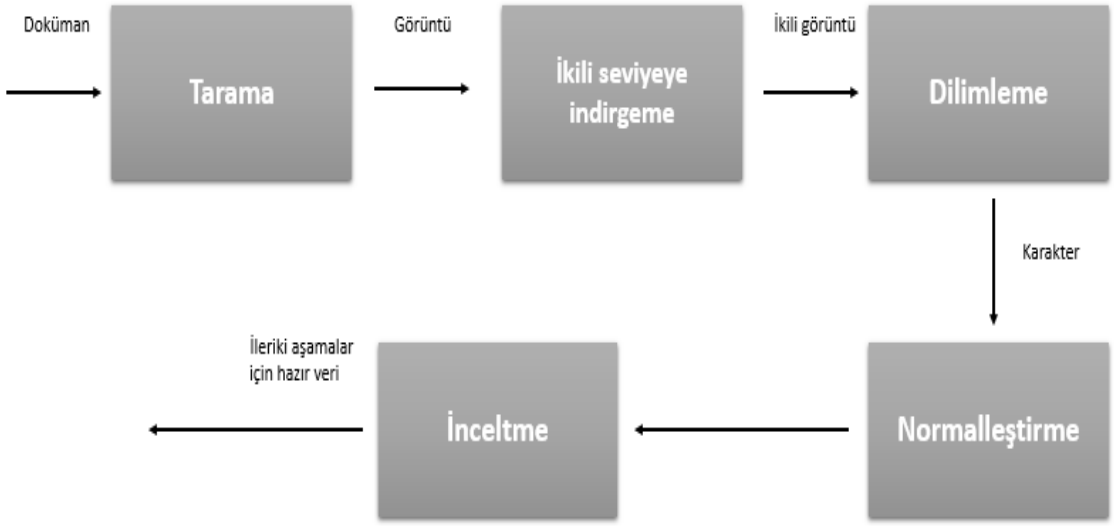
2.1.2.1. Ön-işleme

Ön işleme aşaması, normalleştirme, gürültünün giderilmesi, referans çizgisinin belirlenmesi gibi işlemlerden oluşur. Sayısallaştırma sırasında belge üzerinde oluşan gürültüler giderilmesi için filtreler, gürültü modelleri ve morfolojik işlemler

uygulanabilir. Normalleştirme, yazı karakterlerindeki farklılıkları gidererek karakterlerin standart bir hale getirilmesi işlemidir. Yazı düz bir şekilde yazılmamışsa yukarı- aşağı ya da sağ-sola doğru yazılmışsa yazı düz bir çizgi üzerine gelecek şekilde düzeltilmelidir. Referans çizgisi kullanılarak bütün karakterler bir çizgi üzerine oturtulur ve bu sayesinde bazı karakterlerin birbirleri ile karışması engellenir. Bu olaya örnek olarak 'g' ve '9' karakterleri birbirleriyle karışabilecekken, referans çizgisine göre konumlarına bakıldığında bu iki karakterin birbirleriyle karıştırılması ortadan kalkacaktır (Steinherz, Rivlin, & Intrator, 1999; Şekerci, 2007).

2.1.2.2. Dilimleme

Dilimleme aşaması, sözcüklerin harflere, karakterlere ve rakamlara ayırma işlemidir. Bu aşamada belge mümkün olduğu kadar küçük parçalara bölünür. Bu işlemde belge önce satırlara sonra kelimelere ve en son karakterlerine ayrılır. Bir başka kullanılan yöntem ise önce belgeyi mümkün olduğunca küçük parçalara bölmek, daha sonrada bunları genellikle gizli markov modelleri yöntemini (Hidden Markov Model-HMM) kullanarak birleştirmektir (Arica & Yarman-Vural, 2001; Şekerci, 2007). Şekil 2.5'de ön işleme ve dilimlemenin adımlarının şeması verilmiştir.



Şekil 2.5. Ön işleme ve dilimleme işleminin gösterimi

2.1.2.3. Öznitelik çıkarımı

Öznitelik görüntü üzerinde işimize yarayan önemli bilgiler olarak tanımlanır. El yazısı tanımada öznitelik çıkarımı; görüntü üzerinde önemli bilgi olan karakterlerin belirlenmesi için karakterlerin bulunduğu piksellerin daha belirgin olması açısından yapılan işlemlerdir.

Öznitelik çıkarmadaki amaç verinin daha kısıtlı bir uzayda daha belirgin özelliklerini tanımlamaktır. Bu aşamada farklı yöntemler kullanılabilir. Bu yöntemler: Fourier, Wavelet gibi dönüşümler uygulamak, histogram ya da iz düşüm tabanlı yöntemler, ya da harfleri çizgi, 8 eğri, köşe gibi basit şekiller bütünü olarak tanımlamak bunlardan bazılarıdır (Arica & Yarman-Vural, 2001).

2.1.2.4. Sınıflandırma

Öznitelik çıkarımından sonra elimizdeki verilerle karakter veya rakam sınıflandırması yaparız. Sınıflandırma işlemi için birçok algoritma kullanılmıştır. Bunlardan bazıları aşağıda verilmiştir:

- K-NN
- LibSVM
- Yapay Sinir Ağları
- Convolutional Neural Network(CNN)
- Bayes Sınıflandırıcı

K-NN, LibSVM, Yapay Sinir Ağı algoritmaları analitik strateji olarak işlem görmüş karakter tanıma için kullanılan algoritmalardır. CNN ise bütünsel strateji olarak hem de karakter tanıma için kullanılan algoritmadır.

2.1.2.4.1. K-NN

Yakın Komşuluk (K-NN) algoritması, kümeleme problemlerinde kullanılan bir makine öğrenme algoritmasıdır. Algoritmanın çalışma prensibi; m adet veriden oluşan bir veri kümesini, giriş parametresi olarak verilen k ($k < m$) adet kümeye ayırmaktır. Amaç benzer özelliklerdeki karakterleri kümeleme yapmaktır. Bunun için karakterlerin

özniteliklerine göre bir uzaklık formülü seçilerek karakterlerin kümelenmesi işlemi yapılır. Algoritmanın performansını, k küme sayısı, başlangıç olarak seçilen küme merkezlerinin değerleri ve benzerlik ölçümü kriterleri etkilemektedir (Zouhal & Denoeux, 1998).

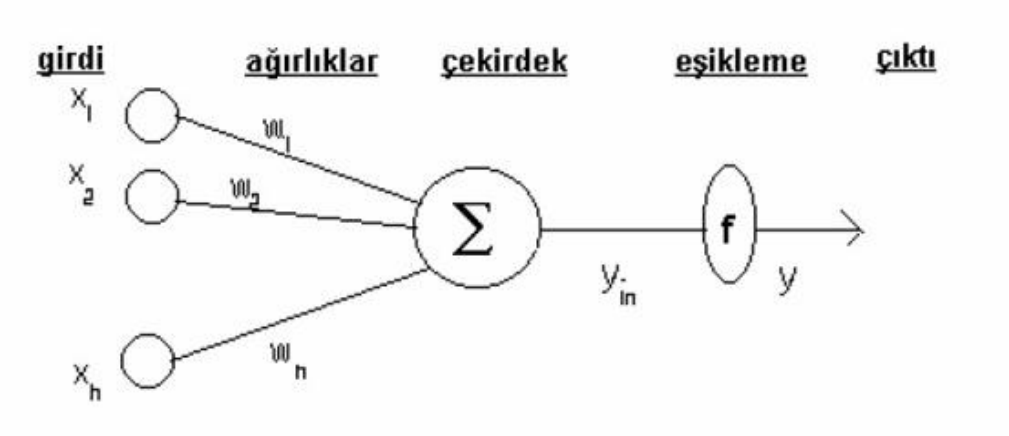
El yazısı tanıma sistemlerinde, karakter tanımada sıklıkla kullanılan bir algoritmadır. Öznitelik çıkarma adımından sonra elde edilen öznitelikler arasında kümeleme işlemi yapılır. K-NN algoritmasında k'nın değerinin seçimi önemlidir. K değeri genellikle 1 olarak seçilir. K değerinin 1 olması birbirine benzeyen karakterlerin tanınmasında sıkıntı olabilir. Buna 'e' ve 'c' harflerinin tanınması örnek verilebilir. Bu yüzden K değeri küçük harf kümelenmesi için 3, büyük harflerin kümelenmesinde 5 olarak seçilir (BEKTAŞ, BABUR, TURHAL, & KÖSE).

2.1.2.4.2. LibSVM

Destek Vektör Makineler (SVM) için geliştirilmiş bir kütüphanedir. İki sınıflı verilerin tahmininde kullanılan SVM' nin aksine çok sınıflı verilerin sınıflandırmasına olanak vermektedir (BEKTAŞ et al.). LibSVM, eğitim ve test kısımları olarak iki aşamada çalışır. Bu çalışmada LibSVM fonksiyonu, SVM sınıflandırıcısının Lineer çekirdeği kullanılarak oluşturulmuştur (BEKTAŞ et al.). Algoritma ilk önce el yazısından elde ettiği özniteliklerle bir eğitime tabi tutulur. Veri setinden eğitime ayrılmamış kısmı ile algoritma test edilir.

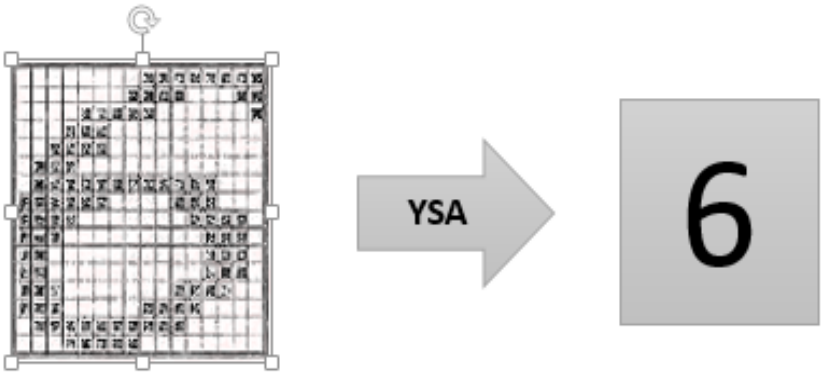
2.1.2.4.3. Yapay Sinir Ağları

İnsan beyninin çalışma mantığını referans alarak geliştirilmiş bir algoritmadır. Sınıflandırma, kestirim gibi işlevler için kullanılır. YSA'ların mimarisi, en az bir girdi ve bir çıktı katmanı olacak şekilde iki ya da daha fazla katmandan oluşur. Her katmanda nöronlar bulunur. Bu nöronlar ağırlık değerleri ile çarpıldıktan sonra çıkan değer bir sonraki katmana aktarılır, toplam fonksiyondan geçirilerek ardından aktivasyon fonksiyonuna gönderilir ve bir sonuç elde edilir. Bu işlemler iterasyonlar halinde yapılır. Ve her iterasyonda ağırlıklar güncellenerek öğrenme işlemi gerçekleştirilir. Şekil 2.6'da basit bir YSA modeli gösterilmiştir.



Şekil 2.6. Basit bir YSA modeli

Ağa giriş parametresi iki şekilde verilebilir. Birinci yöntem A'dan Z'ye kadar olan harflerin hem büyük hem de küçük hallerinin öznitelik matrislerinin verilmesidir. Şekil 2.7'ya bakıldığında bir karakterin matris halinin YSA'ya verilerek işlenmesine bir örnek verilmiştir. İkinci yöntem ise resim biçimindeki verilerin her bir pikseli tek tek okunur ve bir matrise atılır. Bu iki yöntemle elde ettiğimiz matrisler, giriş parametresi olarak YSA'ya verilir. Çıkış değeri olarak her karaktere denk gelen ASCII kodlar üretilir. YSA eğitim aşamasından sonra bir model üretir. Modele test için ayırdığımız değerler verilerek modelin doğruluk oranı belirlenir YSA karakter tanımada en çok kullanılan algoritmadır.

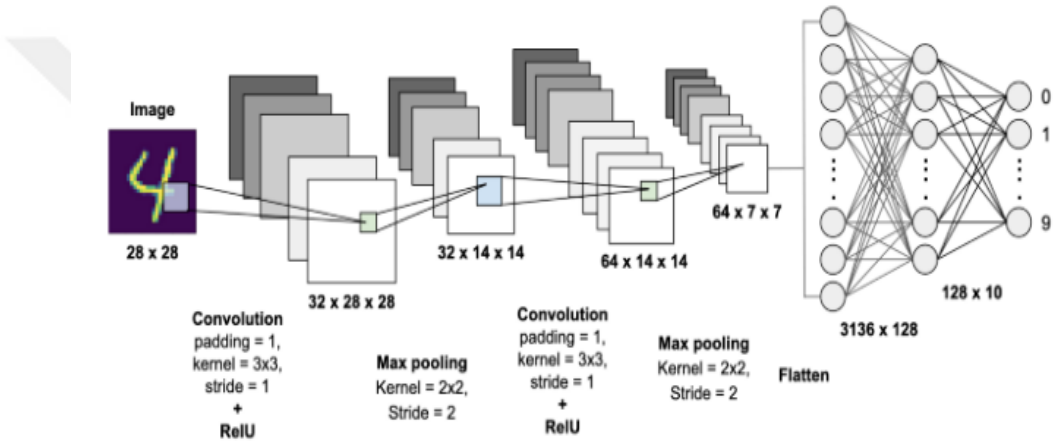


Şekil 2.7. YSA ile karakter tanıma

2.1.2.4.4. Konvolüsyon Sinir Ağları

Konvolüsyon sinir ağları (Convolution Neural Network – CNN) bir derin öğrenme mimarisidir. CNN genel olarak resimlerin sınıflandırılmasında kullanılan bir

kütüphanedir. Kelimeyi karakter ve bütün olarak tanıma yöntemidir. Algoritmaya giriş verisi olarak sınıflandırılacak olan belgeler resim formatında verilir. Bütün resimler gri seviyeye çevrilir. Ağın eğitimi için bütün resimler aynı boyutta ve derinlikte olmalıdır. Algoritma, iki aşamada gerçekleşir. İlk aşamada algoritmaya giriş değeri olarak verilen görüntü, konvolüsyon katmanında işlenir ve görüntüye ait önemli özellikler bu katmanda çıkarılarak bir matrise atılır. İkinci aşamada ise konvolüsyon katmanında elde edilen matris üzerinde çoklu yapay sinir ağı algoritması kullanılarak sınıflandırma işlemi yapılır. Algoritma çıktı olarak verilen kelimeye veya karaktere yakın ya da o kelimenin, karakterin kendisini üretir. Şekil 2.8.'de CNN algoritmasının karakter tanıması adımları gösterilmiştir.



Şekil 2.8. CNN algoritması ile karakter tanıma

2.1.2.4.4. Bayes Sınıflandırıcı

Naive Bayes sınıflandırıcı, Bayes olasılık teoremini örnek alarak geliştirilmiş bir sınıflandırma yöntemidir. Yeni bir sınıflandırma için veri kümesinin tamamını kullanarak yeni bir verinin mevcut sınıflardan hangisine ait olma olasılığını hesaplayan bir yaklaşımdır. Veri en yüksek olasılığa sahip olan sınıfa atanır.

Sınıf değeri olmayan yeni bir veri örneği X'in, Bayes sınıflandırıcı kullanılarak her sınıf için Denklem 2.1'i kullanılarak olasılıkları hesaplanır.

N adet sınıf olduğu düşünülürse, $S = (s_1, s_2, \ldots, s_n)$

Veri kümesindeki her veri için , veri m boyutlu nitelik vektörleri; $X = (x_1, x_2, \ldots, x_m)$ şeklinde gösterilir.

$$P(s_i/X) = \frac{P(X/s_i)*P(s_i)}{P(X)} \quad (2.1)$$

X verisinin hangi sınıfa ait olduğunu Denklem 2.2 kullanılarak bulunur.

$$X \in s_i \rightarrow P(s_i/X) = \max P(s_i/X); i \in |S| \quad (2.2)$$

Naive Bayes sınıflandırıcısı ile karakterlerin öznitelik değerlerine göre olasılıklarını hesaplayarak sınıflandırma işlemi yapılır.

2.1.2.5. Son-işlem

Son işlem sınıflandırmanın doğruluk oranını artırtmak için kullanılır. Doğrulama için birçok yöntem vardır. Bunlardan biri 2 veya 3 harften oluşan kombinasyonları içeren veri tabanları kullanılarak harflerde doğrulama işleminin yapılmasıdır. Diğer bir yöntem ise yüksek seviyeli formel gramatik model kullanılarak cümlenin doğruluğunun arttırılmasıdır.

2.1.3. El Yazısı Tanımanın Kullanıldığı Yerler

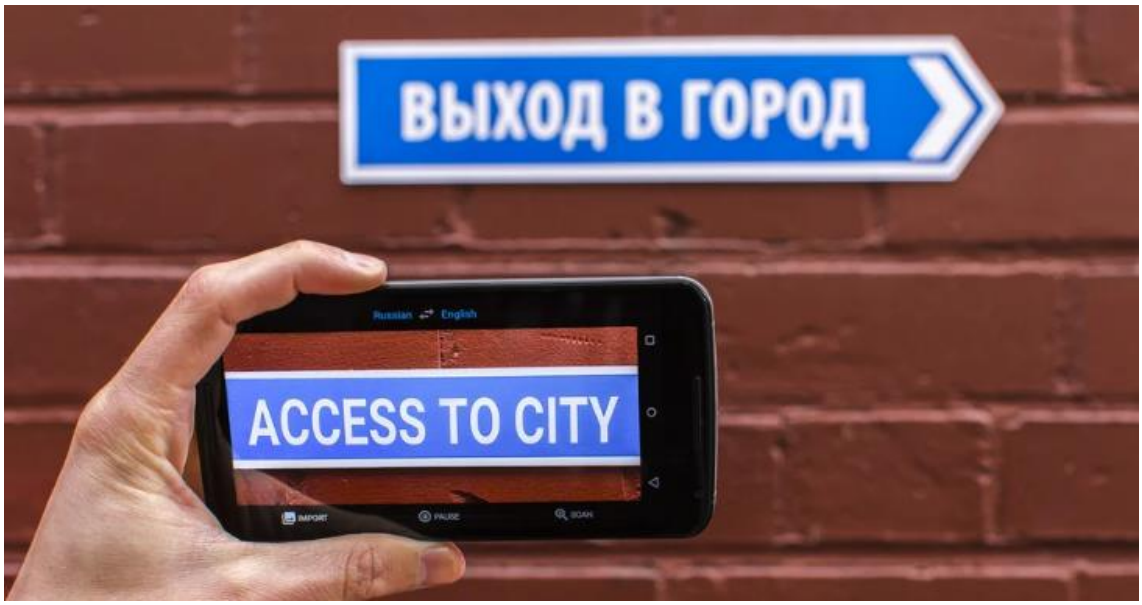
2.1.3.1. Eğitim Alanında

El yazısı tanıma ile ilgili eğitim alanında kullanılmak için birçok uygulama yapılmıştır. Bu uygulamalar, bireyin okuma yazmasını geliştirmek, geçmiş dönemdeki alfabelerin anlanması, yorumlanması ve günümüzdeki alfabelerle karşılaştırılması gibi amaçlar için geliştirilmiştir.

Öğrenmede güçlük çeken çocuklar için el yazısı tanıma ile öğrenmeyi kolaylaştırıcı bir mobil öğrenme ortamı geliştirilmiştir. Android işletim sistemiyle geliştirilmiş ara yüz ile kullanıcının el yazısı bilgileri alınır, alınan bu bilgi öncelikle cihaz üzerinde Ölçeklenebilir Vektör Grafikleri (Scalable Vector Graphics - SVG) dosyasına XML formatında harfin x ve y koordinat bilgileri kaydedilir, daha sonra SVG dosyası sunucudaki Tesseract'a tanıma işlemi için gönderilir. Ekrana yazılan karakter sunucudaki karakterlerle karşılaştırılarak bir öğrenme platformu sunar (Yılmaz, 2014).

Akıllı Mobil Öğrenme Etkileşim Sistemi (Intelligent Mobile Learning Interaction System - IMLIS), zihinsel engelliler için mobil öğrenme ortamı sunmaktadır. Kullanıcı yeteneklerini, öğrenme geçmişine göre tasarlanan karar motoruyla kişiselleştirmeyi amaçlamaktadır. Farklı seviyedeki kişilere göre farklı öğrenme ara yüzleri ve öğrenme aktivitelerini bir arada sunmaktadır (Yılmaz, 2014).

Google'nun Translate uygulaması da yazılı metinler üzerinde çeviri yapma imkânı sunmaktadır. Uygulamada, kamera kullanılarak anlık veya önceden çekilmiş görüntüdeki metni farklı dillere çevirir. Şekil 2.9'da uygulamanın kamera yardımı ile anlık çeviri yaparken ki ekran görüntüsü gösterilmiştir.



Şekil 2.9. Google Translatenin kamera ile metin çevirmesi

2.1.3.2. Adli Alanda

Adli alanda belgelerin incelenmesinde kullanılır. Adli Bilimler alanında, çeşitli yöntemler kullanılarak belgeler üzerinde yapılan sahteciliklerin tespit edilmesi amacıyla yapılan çalışmalardır. Adli alanda el yazısı tanıma, bir belgedeki imza sahteciliği, belgelerdeki karakterlerin karşılaştırılması, belgedeki yazıdan değişen karakterlerin bulunması, yazıya göre kişinin kimliğinin belirlenmesi gibi uygulamalarda kullanılır.

Yazıların incelenmesi, yazan kişinin karakterinin belirlenmesine Grafoloji denilmektedir. Bütün kişilerin el yazıları kendilerine özgü karakteristik özellikler taşırlar. Yazı kişiye özgü bir özelliktir. Kişinin el yazısından kişinin kim olduğuna dair projeler

yapılmıştır. Kişi tanıma alanında, bir metine bağımlı çalışmada, “d”, “y”, “f” ve “th” karakterlerinden mikro öznitelikler türetilmiştir (Pervouchine & Leedham, 2007). Bu öznitelikler türetilirken, adli grafoloji uzmanlarının kullandığı klasik özniteliklerden yararlanılmıştır. Türetilen özniteliklerden en uygun öznitelik kombinasyonlarını, genetik algoritma yöntemi ile öznitelik seçme çalışması yapılmıştır. Sınıflandırma yöntemi olarak sinir ağları kullanılmıştır. Deneysel çalışmalar, CEDAR mektuplarından (Srihari et al., 2001) 156 kişiye ait, ilgili karakterlerden kişi başına 15 ila 30 arasında seçilmesiyle oluşturulan bir veri tabanı üzerinde gerçekleştirilmiştir. En yüksek tanıma oranı, dört karakterden türetilen özniteliklerin kombinasyonlarının sınıflandırılmasıyla %58 olarak elde edilmiştir (Kırlı, 2011).

2.1.3.3. Tıp Alanında

Genel olarak doktorların el yazıları okunaksızdır. Bu nedenle doktorların hastanın tedavisinde kullandığı ilaç raporları, eczanede çalışanlar tarafından anlaşılmazı zordur. Hastaya uyguladığı tedavi yöntemlerini yazdığı raporların başka doktorlar tarafından ya da hasta tarafından anlaşılması zor olmaktadır. El yazısı tanıma ile doktorların el yazıları tanınarak, okunup anlaşılması kolaylaştırılmıştır. Bu konuda değişik çalışmalar yapılmıştır.

Doktorlardan alınan 500 reçete örneği kullanılarak, doktorların el yazısındaki anahtar kelimeler belirlenmiştir. Bu belirlenen kelimeler ile doktorun yazdığı reçete raporunda istenilen ilacın ne olduğu anlaşılmıştır. Projede model olarak Hidden Markov Modeli kullanılmıştır. Performansı arttırmak için de Çok Katmanlı Perceptron (MLP) uygulanmıştır (Roy et al., 2017).

2.1.3.4. Ticaret

Bankalarda senetlerin, çeklerin analizinde ve tanınmasında çeşitli zorluklar vardır. Çekin arka planının renkli ve karmaşık desenli olması çekin analizinde çeşitli sıkıntılara neden olur. Çeklerde önceden yazılmış bilgi alanlarının türü ve konumu ile kullanıcının dolduracağı kısımlar belirli olur analiz bu alan yerleştirilmesine göre yapılmaktadır. Alan yerleşimi analizi, görüntü filtreleme ve ikili hale getirilmesi metin bloklarının bölümlenmesi, kılavuz çizgileri ve gürültünün giderilmesini içerir (Liu et al., 1997).

Bu işlemlerden sonra çeklerin üzerinde yazan yazılar otomatik algılanmaktadır. Çek üzerindeki yazıya göre banka gerekli para çekme veya hesaplama işlemlerini yerine getirmektedir.

Yine bankalarda, müşterilerinin doldurdukları formları bilgisayar ortamına aktarmak için el yazısı tanıma sistemleri kullanılmaktadır. Bu sistemler yardımı ile form üzerinde el yazısı ile yazılmış yazıların bilgisayar ortamına aktarılması sağlanmıştır.

2.1.3.5. El Yazısı Tanıma İle Yapılan Diğer Çalışmalar

(Fanany, 2017) çalışmasında, bir form belgesindeki el yazı karakterlerini tanımak için bir iş akışı algoritması ve bir makine öğrenimi modeli kullanmıştır. Öğrenme modelinde, karakter özelliklerini çıkarmak için Evrişimli Sinir Ağına (CNN) ve sınıflandırma algoritması olarak Destek Vektör Makinelerinden (SVM) yaralanmıştır. Çalışma, on farklı test formu belgesinde % 83.37 doğruluk oranı vermektedir.
Çalışmada, eğitim ve test için NIST SD 19 2nd sürümü kullanılmıştır. Çalışmada karakterlerin özellikleri CNN algoritması kullanılarak çıkartılmış ve özellikler üzerinde SVM algoritması ile bir sınıflandırma işlemi gerçekleştirilmiştir.

(Mahapatra, Choudhury, & Karsh, 2020) de, EMNIST, Devanagari Handwritten Character ve Kannada-MNIST gibi farklı el yazısıyla yazılmış veri kümeleri üzerinde, CNN, hibrit KNN-SVM ve ν-SVM gibi KNN, SVM tabanlı sınıflandırıcılar kullanılmıştır. Geliştirilen modellerin başarıları karılaştırılmıştır (Mahapatra et al., 2020). EMNIST, Devanagari Handwritten Character ve Kannada-MNIST sırasıyla % 89.02,% 86.67 ve% 95,3 başarı oranı elde edilmiştir.

(Saha & Jaiswal, 2020) de, CNN sınıflandırma modeli ile EMNIST ve UCI Devanagari veri seti test edilmiştir.(Saha & Jaiswal, 2020). Bu modelde EMNIST veri seti için %79,3 başarı, UCI Devanagari veri seti için %93 başarı sağlamıştır. Çalışmada, farklı dillerde el yazısı ile yazılmış karakterlerin tanınması amaçlanmaktadır. El yazısı ile yazılmış metinlerde yazılar birbirine bağlıdır, bu nedenle bölümleme gereklidir. Bölümleme işlemi yazıdaki bazı önemli özellikleri çıkarma için kullanılır. Bunlar; cümleyi çıkarmak için satır bölümleme, sözcükleri çıkarmak için sözcük bölümleme ve tek tek harfleri çıkarmak için karakter bölümleme olarak bilinir. Çalışmada segmentasyon işlemi ile farklı çekirdek boyutu ayarlayarak konturlara göre bölme işlemini içerir. Bölümleme işleminden sonra veri kümeleri üzerinde test işlemleri gerçekleşmiştir.

(Cohen, Afshar, Tapson, & Van Schaik, 2017) çalışmasında EMNIST veri kümesinin farklı bölümlerini ve MNIST veri kümesinin detaylı olarak açıklamasını yapmıştır. Çalışmada görüntüleri oluşturmak için kullanılan dönüştürme sürecini ve veri kümesinin bölümleri için bir karşılaştırma sunmuştur.

Yine bu çalışmada EMNIST veri kümesinin bölümleri OPIUM-based sınıflandırma algoritması kullanılarak eğitilmiş ve başarı oranları verilmiştir. Çalışma incelendiğinde rakam tanımada kullanılan MNIST veri kümesi %96.22 başarı oranı ile rakam tanımada kullanılan en başarılı veri kümesidir. Sadece harf tanıma da en başarılı veri kümesi %85.15 oranı ile Letters veri kümesidir. Karakter tanıma olarak içerisinde hem harf hem de rakam bulunan Balanced veri kümesi %78.02 oranı ile karakter tanımada kullanılan en başarılı veri kümesidir.

(Baykal, Aktaş, & Yildiz, 2017)'de biyometrik karakteristiklerden olan imza karakteristiği seçilerek; etkin, çevrimdışı bir doğrulama sistemi yapılmıştır. 23 öznitelik ve DVM algoritması kullanılarak imzanın "Sahte mi, Gerçek mi?" olduğu %100 başarı oranı ile tespit edilmiştir. Çalışma, veri toplama, ön işleme, öznitelik çıkarma ve imza doğrulama olmak üzere 4 aşamadan oluşmaktadır. Ön işleme aşaması, görüntüyü grileştirme, normalizasyon, eşikleme ve görüntü iskeleti çıkarma işlemlerinden oluşmaktadır. Normalizasyon işlemi en yakın komşu interpolasyonu ile uygulanmıştır. Eşikleme işleminde Otsu algoritması ve iskelet çıkarmada Zhang algoritması kullanılmıştır. Çalışmanın üçüncü aşaması olan öznitelik çıkarmada, Hu'nun değişmez momentleri, bölgesel özellikler ve ayrık dalgacık dönüşümü kullanılmıştır. Çalışmada, imza doğrulama için en yüksek başarı gösteren; K En Yakın Komşu (KNN) algoritması ile beraber, YSA, radyal tabanlı fonksiyon (RTF), DVM algoritmaları kullanılmıştır. İmza doğrulama için kullanılan algoritmalardan en başarılı olan DVM olmuştur.

(Can & Yilmaz, 2019)'da NIST SD 19 2nd veri kümesini kullanarak aktarımlı derin öğrenme metodu ile yeni bir öğrenme işlemi geliştirmiştir. CNN'in tüm sınıflar için ürettiği olasılık değerlerini ve bu yöntemlerin başarı oranlarını incelemiştir. Çalışmada 3 tane yöntem kullanılmıştır. Bunlar: "En Yüksek Olasılık Değeri", "En Yüksek Matris Elemanı", ve "Matris Elemanı Ortalaması" dır. Sonuç olarak "Matris Elemanı Ortalaması" yöntemi %1,1 daha iyi bir doğruluk sağlayarak %76.12'lık başarı oranı ile hibrit bir model elde edilmiştir.

Çalışmada Imagenet veri kümesi ile önceden eğitilmiş olan Alexnet, VGGNet19, GoogLeNet ve ResNet50 modelleri, farklı el yazısı karakterleri içeren NIST19 veri kümesi kullanılarak aktarımlı öğrenme metoduyla tekrar eğitilmiştir. Çoğunlukla

CNN'ler için en iyi sonuç ürettiği tespit edilen hiper-parametre değerler kullanılmıştır. Eğitim sırasında kullanılan hiper-parametre değerleri; Öğrenme katsayısı, optimizasyon algoritması, moment değeri, yeniden eğitilen katman sayısı ve dondurulmuş katman sayısıdır. CNN ile yeni eğitim sonuçlarına bakıldığında, test kümesi ve mevcut hiper-parametreler ile yapılan deneyde en iyi sonuç üreten CNN modeli, Alexnet olmuştur. Bu model ile %75.02 doğruluk oranı elde edilmiştir.

(Salouhou, 2019) çalışmasında, Derin Sinir Ağı (DNN), Evrişimsel Sinir Ağı (CNN) ve Yinelemeli Sinir Ağı (RNN) gibi derin öğrenme mimarileri kullanarak, el yazısı karakter tanıma ve resim sınıflandırma işlemlerinin performanslarını arttırmak için ileri düzeydeki iyileştirici fonksiyon ve optimum hiper-parametrelerini belirleyerek aynı veri setleri üzerinde bir karşılaştırma yapmıştır. Derin öğrenmede en çok kullanılan DNN, CNN ve LSTM olan RNN mimarilerinin modellerini oluşturarak algoritmaların performanslarını karşılaştırmıştır. Modellerin eğitiltmesi ve test gerçekleştirmek üzere, Cifar-10, Fashion-Mnist, Mnist ve Arapça veri setleri kullanılmıştır. Elde edilen sonuçlara göre modeller arasında en iyi performanslı algoritma CNN modelidir. CNN algoritması, Mnist veri seti üzerinde %99.88 doğruluk ve 0.042 kayıp değeri elde edilmiş ve en iyi derin öğrenme modeli olarak belirlenmiştir.

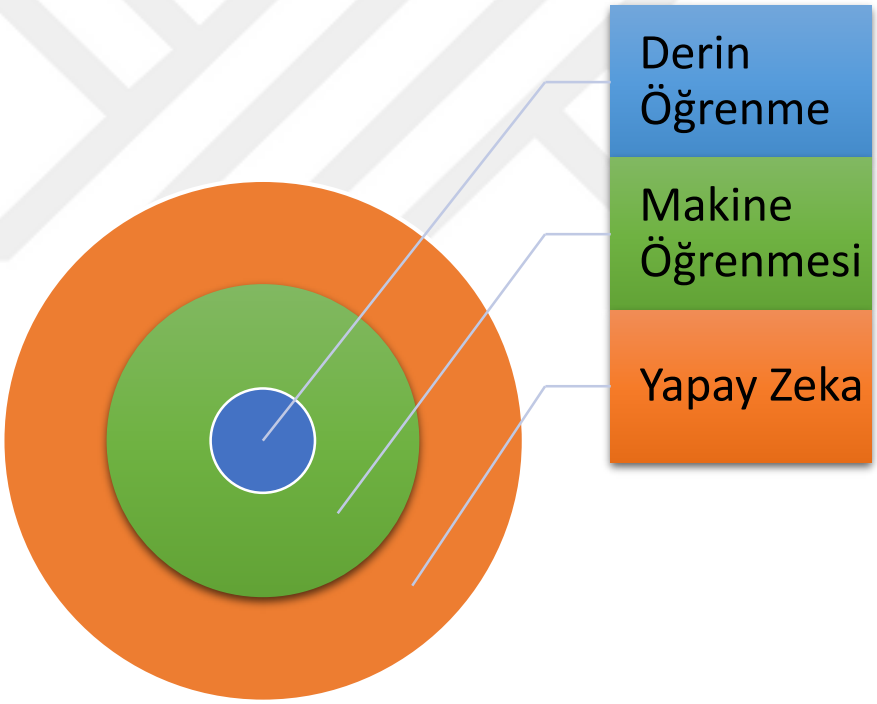
(Fatimah, Djamal, Ilyas, & Renaldi, 2019) çalışmasında, el yazısına göre bir kişilik tanıma uygulaması geliştirmiştir. Bilgisayar tabanlı Grafoloji, el yazısı görüntülerini inceleyerek ve her özelliğin modeline göre önerilen kişilik sonuçlarını verir. Bu çalışmada, 6 özellik 2 teknik kullanılarak yapılmıştır. Birinci teknik, çoklu yapı analizi kullanarak, kenar boşluğu, satırlar arası boşluk, kelimeler arasındaki boşluk, eğim ve baskın bölge özelliklileri ile kişilik tanıma yapılmasıdır. İkinci teknik, dört belirli harf ('a', 'g', 's', 't') Evrişimli Sinir Ağları (CNN) sınıflandırma yöntemini kullanılarak kişilik tanıma yapılmasıdır. Sonuç olarak, yapı analizinin doğruluğunun % 82,5 iken, CNN kullanan sembol yaklaşımının doğruluğunun % 98.03 olarak elde edilmiştir.

(Kırlı, 2011) çalışmasında, çevrim dışı olarak el yazısı satır görüntülerinden, metin içeriğinden bağımsız olarak otomatik kişi tanıma yapmayı amaçlamıştır. Deneysel çalışmalar, IAM veri tabanı üzerinde gerçekleştirilmiştir. Veri tabanındaki el yazısı satırları, el yazısı ile yazılmış formlardan ayıklanmış olup, her biri farklı içeriğe sahiptir. Veri tabanındaki el yazıları, herhangi bir taklitten uzak olup, kişilerin kendilerine ait en doğal yazım stilleriyle yazılmışlardır. Çalışmada el yazısı satır görüntülerinden kişiye özgü öznitelikler elde edebilmek için yeni yaklaşımlar önerilmiştir. Türetilen öznitelikler,

global ve lokal olarak ayrılmıştır. Bu öznitelikler çeşitli sınıflandırma yöntemleri ile test edilmişlerdir. Kişi tanıma modelimizde kullanılan sınıflandırma yöntemleri K-NN, GMM ve Normal Dağılımlı Fark Fonksiyonu Bayes 7 (Normal Density Discriminant Function (NDDF) Bayes) yöntemleridir.

2.2. Derin Öğrenme

Derin öğrenme, verilen giriş değerlerine göre sonuç değerleri üreten bir makine öğrenmesi yöntemidir. Derin öğrenme, makine öğrenmesi ve yapay zekâ kavramları birbirinden farklı fakat birbirleri ile ilişkili olan terimlerdir. Şekil 2.10' a baktığımızda derin öğrenme, makine öğrenmesinin; makine öğrenmesi ise yapay zekânın alt dalı olarak görünmektedir.



Şekil 2.10. Derin öğrenmenin kapsamı

Derin öğrenme, birden fazla doğrusal olmayan katman yardımı ile özellik çıkarma işlemi yapar. Ardışık olan tüm katmanlar önceki katmandaki çıktıyı girdi olarak alır (Deng & Yu, 2014; Şeker, Diri, & Balık, 2017). Derin öğrenme temel olarak veriyi en iyi temsil eden piksellerin değerlerine göre öğrenmeye dayalı bir yöntemdir. Bu yönteme

özellik çıkarma yöntemi denir. Derin öğrenmede, görüntü üzerindeki önemli piksellerin elle çıkarılması yerine farklı filtreler içeren algoritmalar kullanılarak çıkarılmaktadır.

Derin öğrenmenin tarihine bakıldığında denetimli derin beslemeli çok katmanlı perceptronlar için ilk algoritma Ivakhnenko ve Lapa tarafından 1965 yılında ortaya çıkmıştır (Ivakhnenko & Lapa, 1966). Bu çalışmada, katmanlardaki en iyi özellikler, istatistiksel yöntemlerle belirlenip bir sonraki katmana iletilmektedir. Ağlarını uçtan uca eğitmek için geri yayılım (backpropagation) kullanılmamıştır.

İkinci olarak derin öğrenme mimarisini 1979 yılında Fukushima önermiştir. Omurgalı canlıların sinir sisteminden esinlenerek geliştirilmiştir. Fukushima'nın ağları günümüzdeki ağlara benzer olarak çoklu bükülme ve havuz katmanları içermektedir (Fukushima, Miyake, & Ito, 1983).

Derin öğrenme mimarileri önceki yıllarda ortaya çıkmış olmasına rağmen ilk başarılı derin sinir ağı uygulamasını Yann LeCun ve arkadaşları tarafından posta kutuları üzerindeki, yazıları kullanarak geliştirmişlerdir (LeCun et al., 1989; Şeker et al., 2017). Bu çalışmadan sonra Yann LeCunn "LeNet" ağını kullanarak el yazısı rakamlarını (MNIST) sınıflandırmak için kıvrımlı ağlarla geri yayılımı birlikte uygulamıştır (Le Cun, Boser, et al., 1989). Bu çalışma ile karakter tanımadaki ilk işlemler yapılmaya başlamıştır.

Günümüzde gelişen teknoloji ile bu alana olan ilgide artmıştır. Derin öğrenme (deep learning) ifadesi ilk kez 2000 yılında Igor Aizenberg ve arkadaşları tarafından ortaya atılmıştır (Aizenberg, Aizenberg, & Vandewalle, 2000).

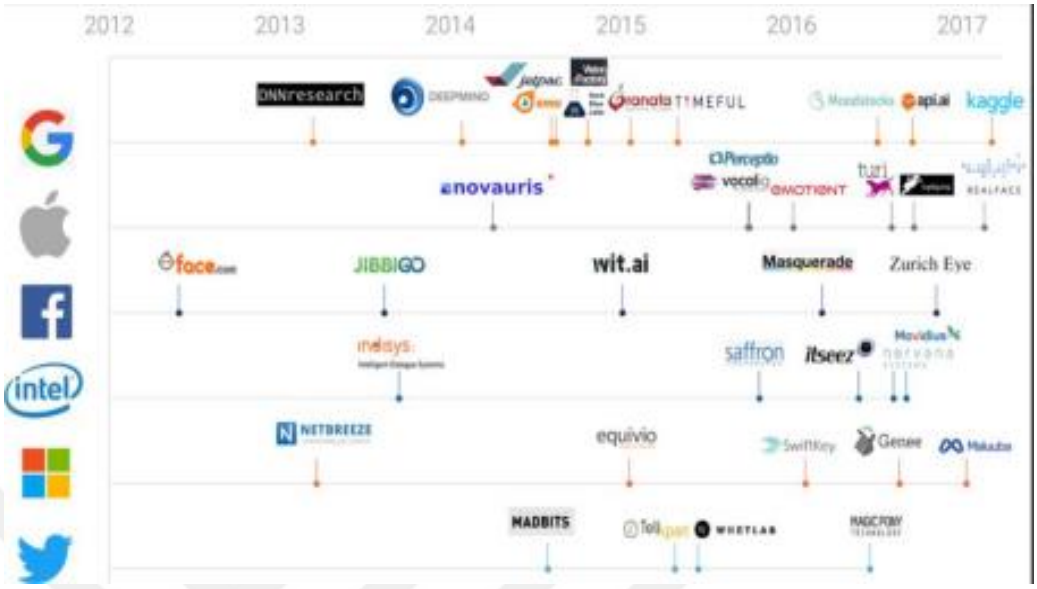
Daha sonra bu alanda Geoffrey Hinton 2006'da, çok katmanlı ileri beslemeli bir sinir ağının nasıl eğitileceğini ve denetimli bir geri yayılım yöntemini nasıl yapacağını makalesinde anlatmıştır (Hinton, 2007).

Bilgisayarların GPU hızlarının artmasıyla derin ağların ön-eğitim (pre-training) yapılmadan eğitilmesi ortaya çıkmıştır. Ciresan ve arkadaşları trafik işaretleri, medikal görüntüleme ve karakter tanıma gibi uygulamalarında bu yaklaşımı kullanmışlardır (Ciresan, Meier, Gambardella, & Schmidhuber, 2011).

Krizhevsky, Sutskever ve Hinton 2012 yılında benzer mimariler kullanmışlar ve GPU kullanarak yaptıkları çalışmalarda, ezberlemeyi azaltmak için "dropout" adı verilen katmanı geliştirmişler ve uygulamalarında kullanmışlardır (Hinton, Srivastava, Krizhevsky, Sutskever, & Salakhutdinov, 2012).

Derin öğrenmenin bu hızla gelişmesinin üzerine Google, Facebook ve Microsoft gibi büyük teknoloji firmaları derin öğrenme alanında yatırım yapmaya başlamışlardır. Şekil 2.11'e bakıldığında 2012-2017 tarihleri arasında derin öğrenmeye yatırım yapan

firmaların bu zaman aralıklarında kullanmaya başladıkları uygulamaların isimleri görünmektedir.



Şekil 2.11. Derin öğrenmeyi kullanan firmaların tarihlere göre uygulamaları (Şeker et al., 2017)

Geçmişten günümüze derin öğrenme ile birçok uygulama geliştirilmiş ve geliştirilmeye de devam etmektedir. Bu süreçte derin öğrenme alanında çok çalışma yapılmış ve değişik mimariler geliştirilmiştir. Derin öğrenmenin 6 tane mimarisi vardır. Bunlar;

- Konvolüsyon Sinir Ağları
- Tekrarlayan Sinir Ağı
- Uzun / Kısa Süreli Bellek
- Kısıtlı Boltzmann Makineleri
- Derin İnanç Ağları
- Derin Oto Kodlayıcılar ve Oto Kodlayıcılar

2.2.1. Derin Öğrenme Mimarileri

2.2.1.1 Konvolüsyon Sinir Ağları

Konvolüsyon sinir ağları (Convolution Neural Network - CNN) çok katmanlı algılayıcıların (Multi Layer Perceptron –MLP) bir türü olarak bilinir. Tarihte ilk CNN ağı

1988 yılında Yann LeCun tarafından bulunan LeNet mimarisidir (Le Cun, Jackel, et al., 1989).

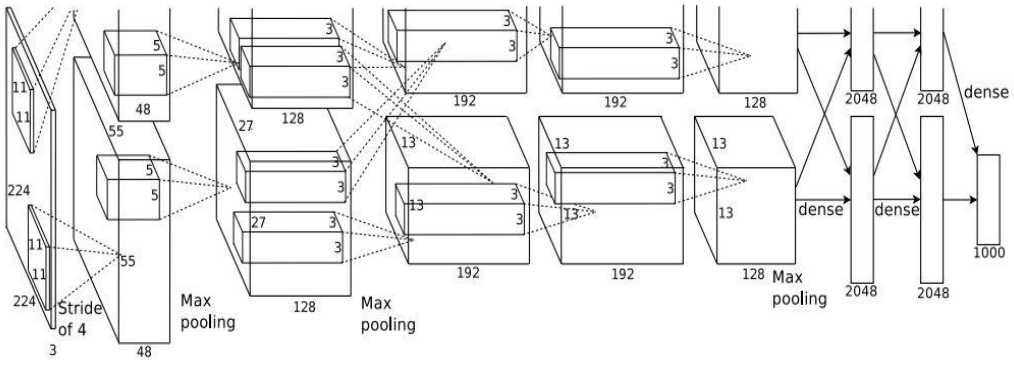
CNN algoritmaları görüntü ve ses işleme (Mushtaq, Su, & Tran, 2020; Su, Zhang, Wang, & Madani, 2019), doğal dil işleme (NLP) (Akhtyamova, Ignatov, & Cardiff, 2017; Sun et al., 2019), biyomedikal görüntü işleme (Cho, Ha, Park, & Park, 2020; Momeni, Thibault, & Gevaert, 2018) gibi alanlarda kullanılmaktadır. Görüntü işleme alanında en iyi sınıflandırma başarısına sahip olan derin öğrenme algoritmasıdır. Bu alanda çok fazla çalışma yapılmış ve birçok mimari geliştirilmiştir. Günümüzde sık kullanılan hazır CNN mimarileri vardır. Bu mimariler;

- LeNet
- AlexNet
- VGGNet
- GoogLeNet
- ResNet
- ZFNet

Olarak bilinmekte ve günümüzde birçok uygulamada kullanılmaktadır.

CNN mimarisi veriyi çeşitli katmanlarda işler. CNN modelinin katman yapısı Şekil 2.12'da olduğu gibidir. Bu katmanlar (Ergin, 2018,October);

- Convolutional layer- Görüntü üzerinde özellikleri çıkarmak için kullanılır.
- Non-Linearity layer- ReLU katmanı – Aktivasyon işleminin bulunduğu katmandır. Sistemde doğrusal olmayan (non-linearity) bir fonksiyon kullanır.
- Pooling(Downsampling) layer- Boyut azaltma ve uygunluk kontrolu yapar.
- Flattening layer – Klasik Sinir Ağı için tek boyutlu vektör verilerini hazırlar.
- Fully-Connected layer – Sınıflandırmada kullanılan Standart Sinir Ağı.



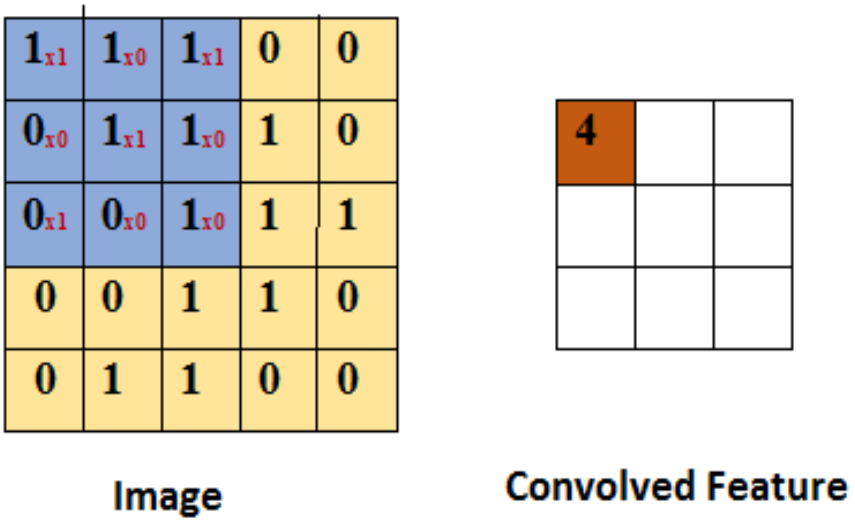
Şekil 2.12. CNN'nın katman yapısı

Convolution katmanı görüntü üzerindeki özellikleri çıkartmak için kullanılır. Bu katmanda görüntü üzerindeki özellikleri çıkarmak için görüntünün boyutundan daha küçük olan düşük veya yüksek seviyeli filtreler kullanılmaktadır. Bu filtreler genelde tek sayılardan oluşan matrislerdir. Bu filtreleri görüntü üzerinde gezdirerek ve matris çarpımı kullanarak, görüntü üzerindeki önemli özellikler tespit edilmeye çalışılır.

Şekil 2.13'e bakıldığında 5x5'lık orijinal bir görüntü üzerinde 3x3'lük bir filtre (mavi renkli kareler) gezdirilmektedir. Çıkan sonuçlar sağ taraftaki yeni matris olan colvolved feature matrisine yazılır. Filtrelerden sonra resmin orijinal boyutunu kaybetmemesi için sıfır değerleri eklenir.

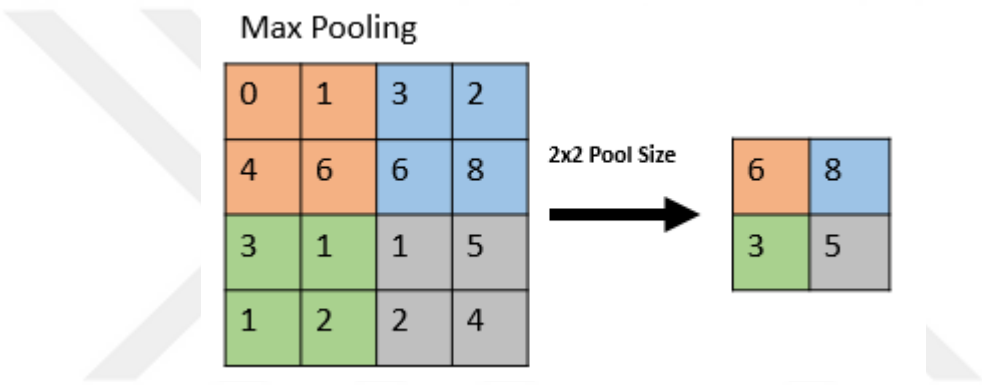
Convolution katmalarından sonra Non-Linearity katmanı gelir. Bu katman, aktivasyon katmanı olarak da bilinir ve aktivasyon fonksiyonlarından biri kullanılır. Bu katmanın amacı aktivasyon fonksiyonlarını kullanarak modelin negatif değerleri öğrenmesini ya da bazı özellikleri kavrayamamasını engellemek için kullanılmaktadır. Genelde aktivasyon fonksiyonu olarak; ReLU, tanh ve sigmoid gibi doğrusal olmayan fonksiyonlar kullanılır. Hız konusunda en iyi sonuçları Ractifier (Relu) fonksiyonu verdiği için eğitimlerde çoğunlukla bu fonksiyon kullanılır. ReLu fonksiyonu denklem 1 de gösterilmiştir.

$$Relu\ Fonksiyonu = -f(x) = max(0, x) \qquad (1)$$

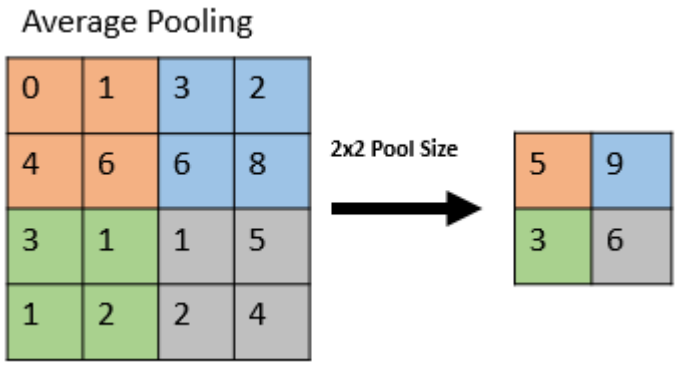


Şekil 2.13. Convolution filtresi

Pooling (havuzlama) katmanı, ardışık convolution katmanları arasına eklenir. Bu katman boyutsallığı azaltmak için kullanılmaktadır. Bu katman sayesinde çıkarılan gereksiz özellikler yok edilerek önemli olan özelliklere odaklanma sağlanır. CNN modellerde genellikle Max (Maksimum) ve Average (Ortalama) olarak iki farklı pooling tekniği kullanılır. Birinci teknik Maxpooling de öncelikle bir filtre oluşturulur ve bu filtre resim üzerinde gezdirilerek kapladığı alandaki en büyük sayıyı alır. İkinci teknik Averagepooling de ise yine bir filtre oluşturulur ve bu filtre resim üzerinde gezdirilerek kapladığı alandaki piksellerin ortalaması olan sayı alınır. Şekil 2.14'de maxpooling tekniğinin örneği gösterilmiştir. Şekil 2.15'de average pooling tekniğinin örneği gösterilmiştir.



Şekil 2.14. Maxpooling tekniği

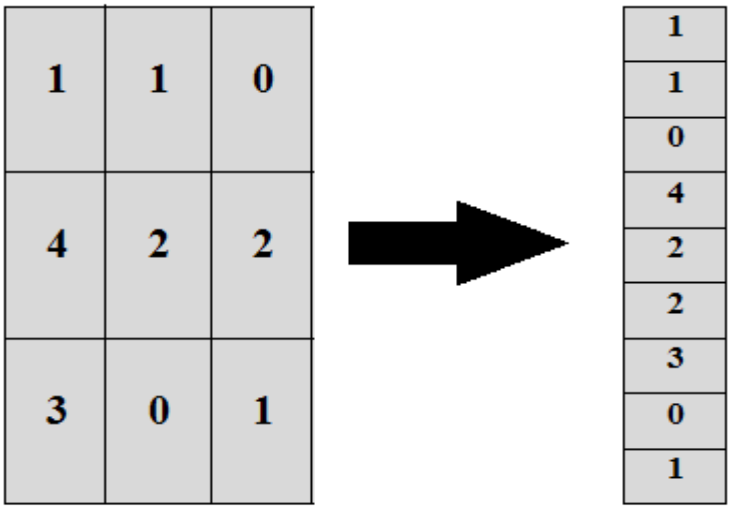


Şekil 2.15. Average pooling tekniği

Fatting katmanı, Fully Connected Layer için giriş verilerini hazırlayan katmandır. Sinir ağlarının giriş verilerini tek boyutlu bir dizi olarak alır. Bu veriler ise, Convolution ve Pooling katmanlarından gelen matrislerdeki verilerdir. Fatting katmanıda bu verileri

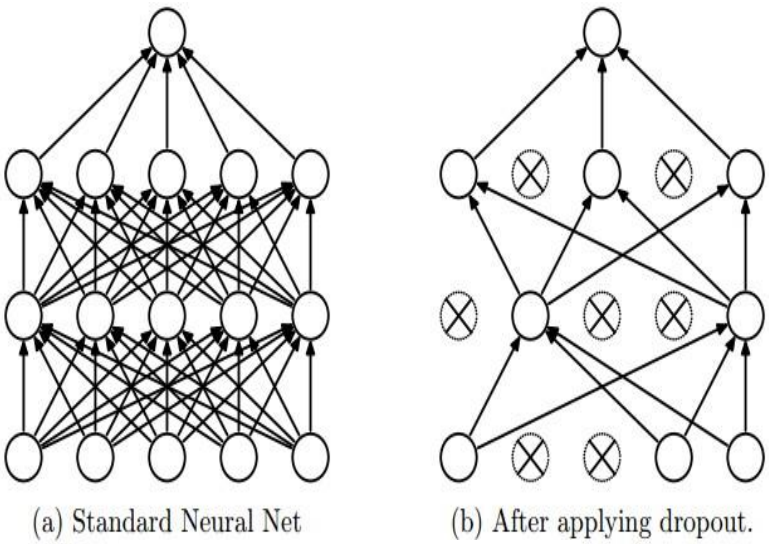
tek boyutlu diziye çevirme işlemini yapar. Şekil 2.16'de tek boyutlu diziye çevrilmiş bir son katman örneği verilmiştir.

Fully Connected Layer katmanı, verileri Flattening katmanında işlenip tek boyutlu diziye çevrilmiş şekilde alır ve klasik sinir ağı yöntemi kullanılarak öğrenme işlemini gerçekleştirir.



Şekil 2.16. Fatting katmanı

Modelin eğitimi sırasında under fitting , gereksiz ezberleme, over fitting gibi problemlerini önlemek için Dropout yöntemi kullanılır. Şekil 2.17'de olduğu gibi. Ağ içindeki bazı bağlantıların kaldırılmasıyla eğitim performansı artırılmıştır.

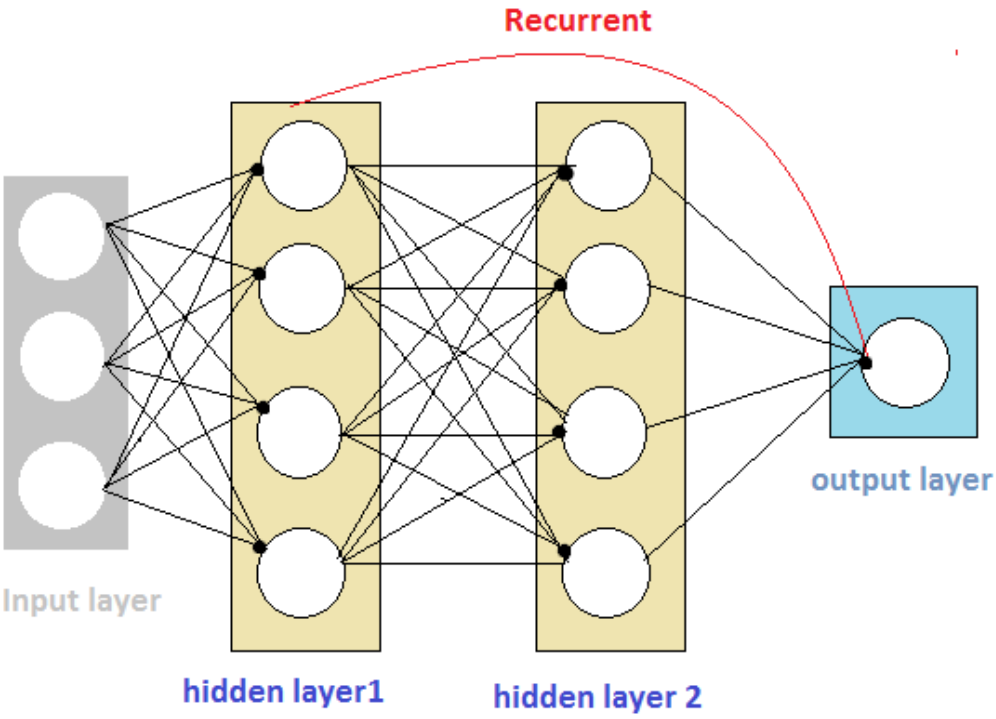


Şekil 2.17. Dropout katmanı

2.2.1.2. Tekrarlayan Sinir Ağı

Tekrarlayan Sinir Ağı (Recurrent Neural Network – RNN) ilk olarak 1980'lerde Simple Recurrent Network (Basit Tekrarlayan Ağ) olarak ortaya atılmıştır. Daha sonra 1990'lı yıllarda Jeff Elman tarafından RNN olarak ortaya atılmıştır.

RNN, düğümler arasında bağlantıların birbirine bağlı bir döngü şeklinde olduğu derin öğrenme mimarisidir. RNN'nın amacı ardışık bilgileri kullanarak çıktı değeri üretmektir. Geleneksel bir sinir ağında tüm girdi katmanları, gizli katmanlar ve çıktılar birbiriyle bağımsız olarak çalışmaktadır. RNN'de ise tüm katmalar birbirine bağlı bir şekilde döngü mantığıyla çalışmaktadır. Bu sebepten dolayı RNN mimarisi yinelenen olarak adlandırılmaktadır. RNN mimarisinin iki çeşit algoritması vardır. Bunlar; İki Yönlü RNN'ler (Bidirectional RNNs) ve Derin (iki yönlü) RNN'ler (Deep RNNs) dir. Şekil 2.18'de RNN mimarisinin katman yapısı görülmektedir.

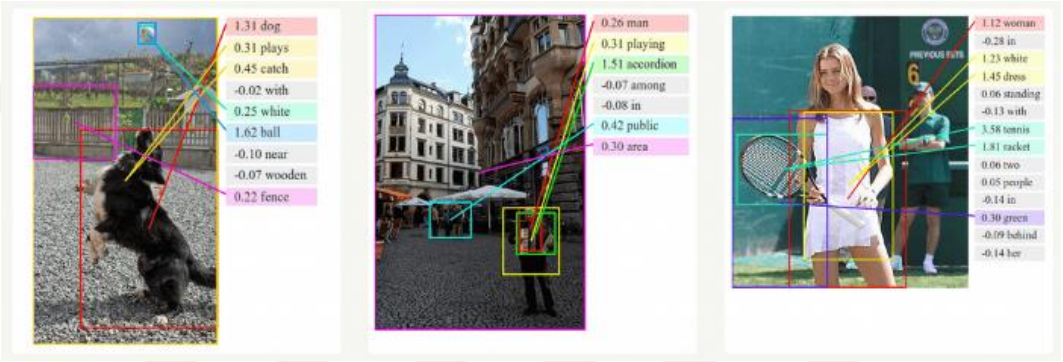


Şekil 2.18. RNN katman mimarisi

Günümüzde teknolojinin gelişmesiyle birlikte RNN algoritmaları birden fazla alanda kullanılmaya başlamıştır. Bu alanların başında ise kelime tahmini gelmektedir.

Kelime tanımadan farklı olarak kullanıldığı alanlar aşağıda verilmiştir. Şekil 2.19'a bakıldığında bu algoritma ile yapılan bazı uygulamaların görüntüleri gösterilmiştir. Bunlar:

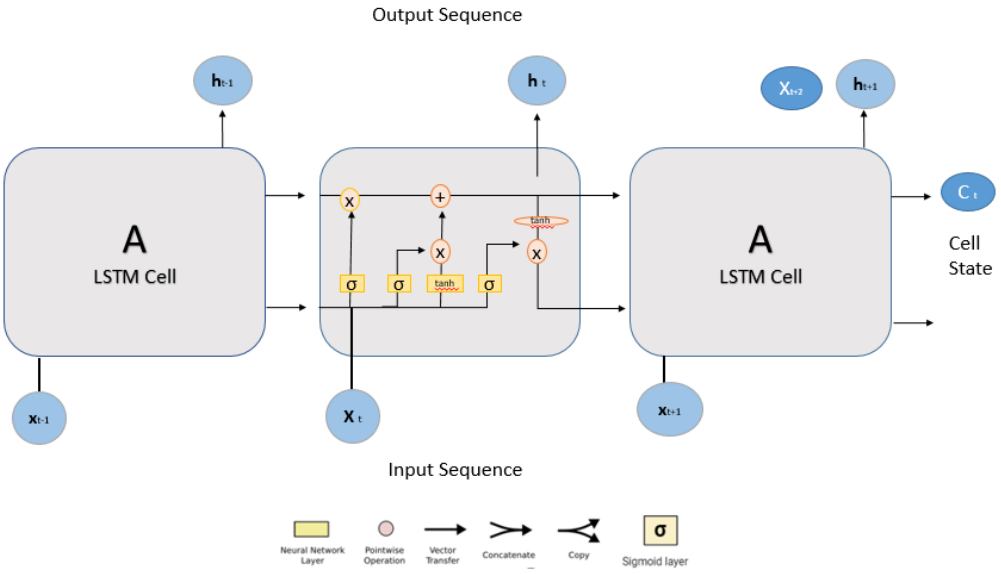
- Borsa Tahmini
- Zaman Serisi Anomali Tespiti
- Konuşma Tanıma
- Film Altyazı Tahmini ve Çevirisi
- Nesne Tanıma



Şekil 2.19. RNN ile yapılan uygulamalar

2.2.1.3. Uzun / Kısa Süreli Bellek

Uzun / kısa süreli belleğin (Long / short term memory - LSTM) RNN'ye benzeyen mimari bir yapısı vardır. RNN'den farklı olarak hafıza mantığının olmasıdır. Hafıza mantığı ile bağlamdaki boşlukları hatırlar. LSTM 1997 yılında Hochreiter ve Schmidhuber tarafından ortaya atılmıştır (Hochreiter & Schmidhuber, 1997). RNN'lerin eğitilmesindeki sorunlar bu mimari ile ortadan kalkmıştır. Şekil 2.20'de LSTM'nın yapısı ayrıntılı bir şekilde verilmiştir.



Şekil 2.20. LSTM yapısı

LSTM'in en önemli yapılarından biri cell state'dir. Bu yapıda bir hücreden diğer hücreye bilgi akışı ve gradient akışı sorunsuz bir şekilde gerçekleşir.

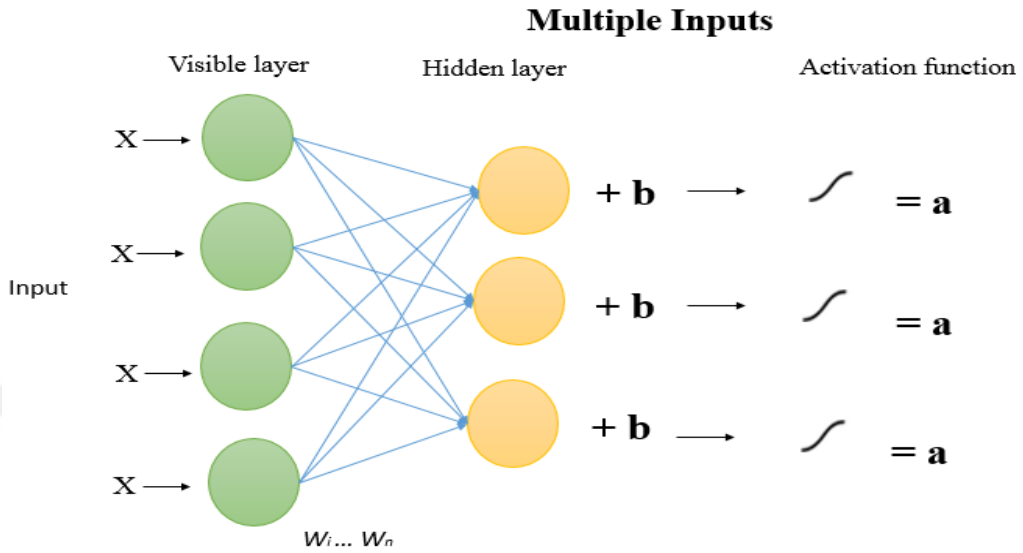
LSTM modellerinin en çok kullanıldığı uygulamalardan birisi konuşma ve metin işleme projeleridir. Bu projelerde oldukça iyi sonuçlar vermektedir. Bunların dışında LSTM mimarileri ile protein homolojisinin algılanması, robotik kalp cerrahi, müzik üretimi, dil çevirisi, düzensiz dillerde öğrenme, çevrimdışı el yazı tanınması gibi birçok alanda uygulamalar yapılmıştır.

2.2.1.4. Kısıtlı Boltzmann Makineleri

Kısıtlı Boltzmann Makineleri (Restricted Boltzmann Machines – RBM) ilk kez 1986 yılında ortaya çıkmış ve daha sonra Geoffrey Hinton tarafından geliştirilmiştir. RBM 'ler boyutsallık azaltma (Hinton & Salakhutdinov, 2006), sınıflandırma (Larochelle & Bengio, 2008), regresyon, işbirlikçi filtreleme (Salakhutdinov, Mnih, & Hinton, 2007), özellik öğrenme (Coates, Ng, & Lee, 2011) ve konu modelleme (Hinton & Salakhutdinov, 2009) gibi birçok alanda kullanılır. RBM'ler tarihe bakıldığında ele alacağımız ilk sinir ağıdır.

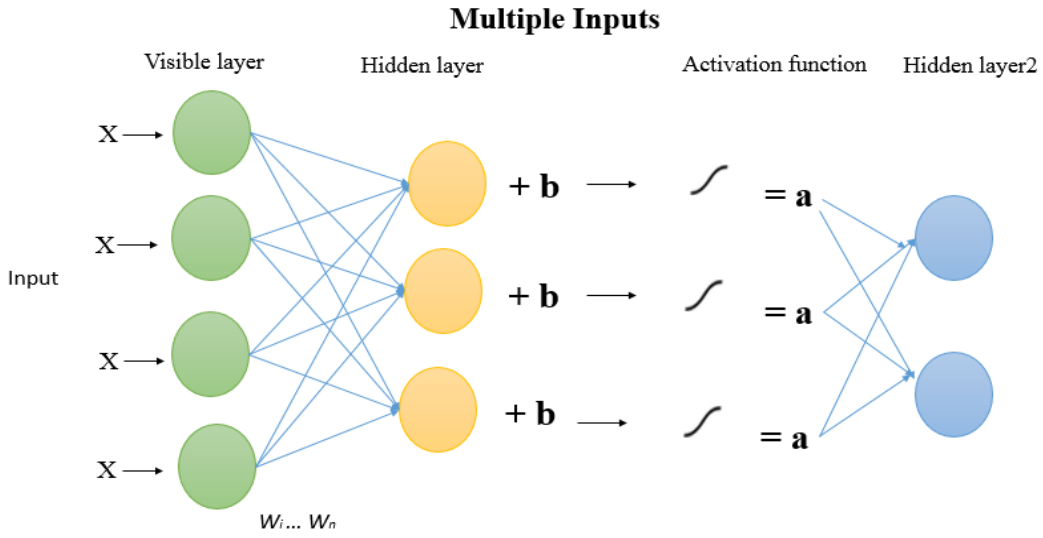
RBM' ler görünür (girdi) ve gizli olmak üzere aralarında bağlantılı katmalardan oluşur. Bu katmanlarda görülen her bir daireye düğüm adı verilir ve düğümler sayesinde basit hesaplama işlemleri yapılır. Düğümler birbirine katmanlar şeklinde bağlanır ancak

aynı katmandaki düğümler birbirine bağlanamaz. Bu işlem sayesinde katman içinde iletişim yoktur buda RBM' deki kısıtlamadır. Şekil 2.21' e bakıldığında RMB'de bir girdi ve bir gizli katman için bağlantı yapısı ve çalışma mantığı görülmektedir.



Şekil 2.21. Bütün girdi katmanları ve giriş katmanları için yapılan işlemler

RMB'lerde birden fazla gizli katman eklenerek daha derin bir eğitimle daha iyi sonuçlar alınabilir. Şekil 2.22'ye bakıldığında birden fazla gizli katmanın kullanımı ve düğümler arasındaki işlem yapısı verilmiştir.

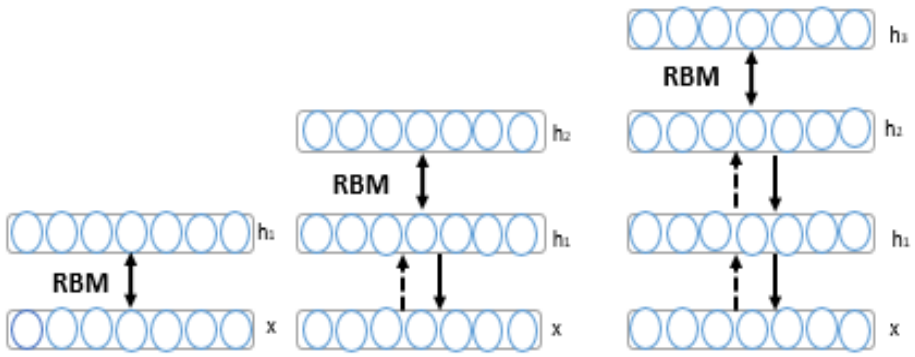


2.22. Birden fazla gizli katman için yapılan işlem

2.2.1.5. Derin İnanç Ağlar

Derin İnanç Ağlar (Deep Belief Networks – DBN), Geoffrey Hinton ve arkadaşları tarafından ortaya atılan çok katmanlı derin sinir ağıdır. Birbirleriyle bağlantılı ancak birimlerin olmadığı birden fazla gizli katmanlardan oluşurlar.

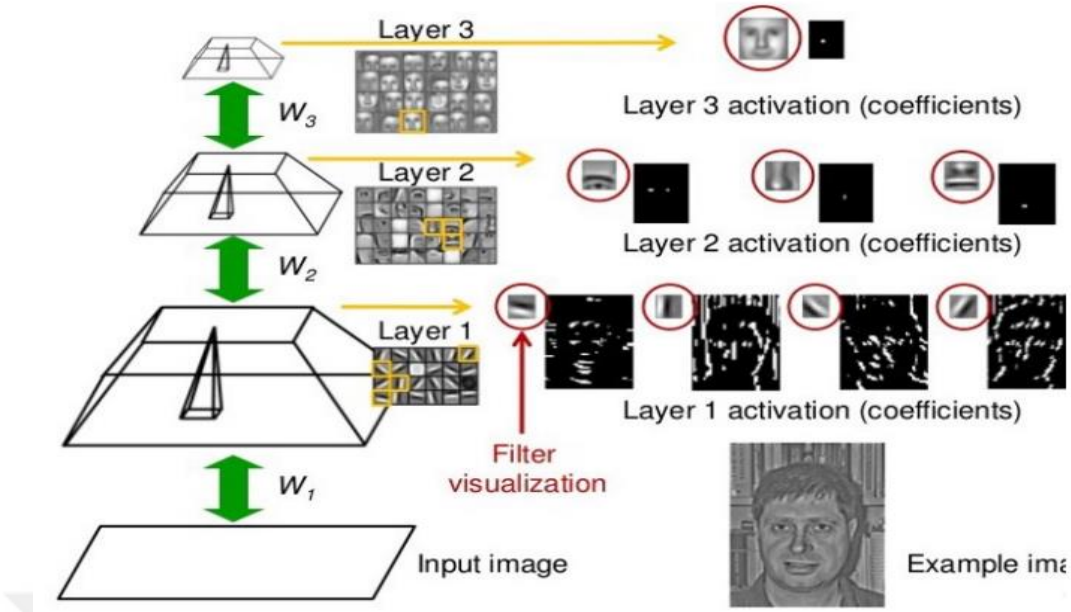
DBN'ler, Kısıtlı Boltzmann Makineleri (RBM'ler) veya Autoencoder gibi, denetlenmeyen ağların bir yapısı olarak görülebilir. DBN'ler RBM'lerin yığınları olarak kullanılmaktadır. DBM'lerin çalışma mantığı arka arkaya eklenen RBM'lerin katmanlarından oluşan bir sinir ağı yaklaşımıdır. RBM'ler sırayla eğitilerek DBN'lerin öğrenmesi gerçekleşir. Şekil 2.23'de bu yapıya örnek olarak arka arkaya eklenmiş RBM'ler ile oluşturulmuş bir DBN yapısı verilmiştir.



Şekil 2.23. RBM'lerden oluşmuş bir DBN yapısı

DBN'ler birden fazla alt dala ayrılmıştır ve bu dalın kullanım amaçları birbirinden farklı şekillerde yapılmaktadır. Bu yüzden DBN'ler çok farklı alanlarda farklı şekillerde kullanılmıştır. Şekil 2.24' da bu çalışmalara örnek bir uygulama gösterilmiştir. Ayrıca DBN ile yapılan çalışmalar aşağıda verilmiştir:

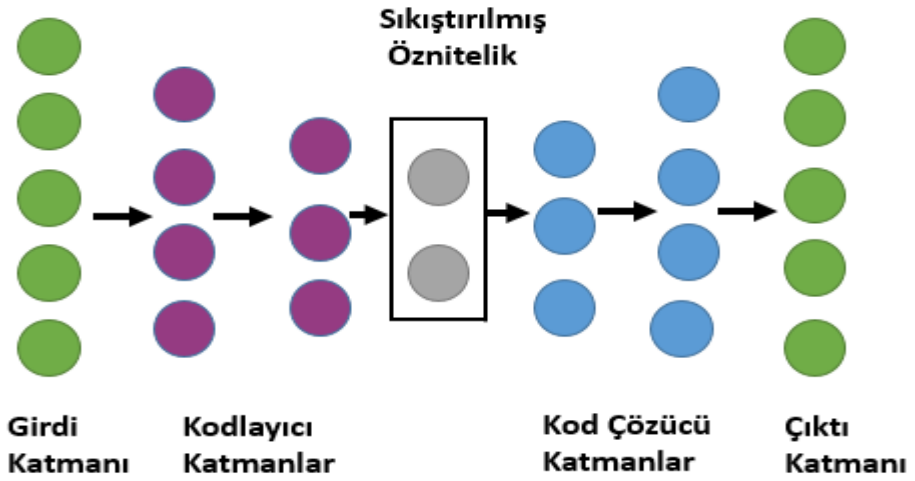
- Görüntü Tanımlama
- Video Dizileri
- Hareket Yakalama
- Konuşma Tanıma



Şekil 2.24. DBN uygulama örneği

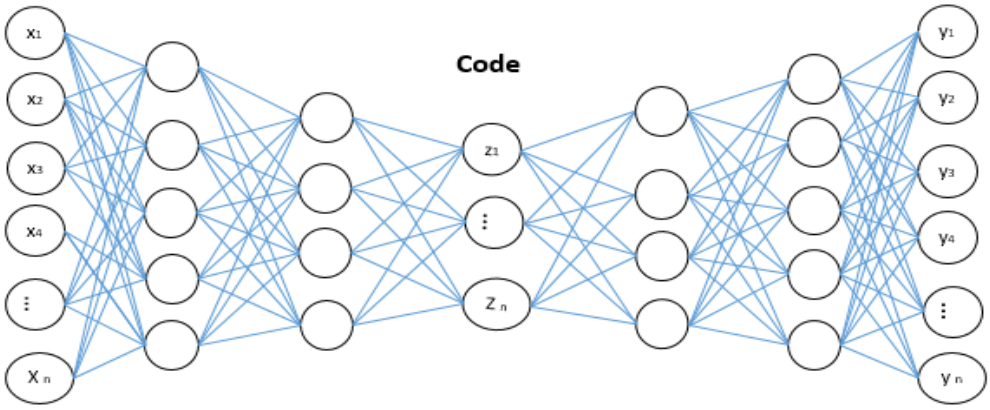
2.2.1.6. Derin Oto Kodlayıcılar ve Oto Kodlayıcılar

Oto kodlayıcılar (Autoencoders - AE), Diabolo ağı olarak adlandırılan özel yapay sinir ağıdır (Liou, Cheng, Liou, & Liou, 2014; Şeker et al., 2017). AE'ler, girdi katmanındaki verileri çıktı katmanına kopyalayan ileri beslemeli bir sinir ağıdır. Yani, sinir ağına girdi olarak verilen veriyi, çıktı katmanında tekrar oluşturur. Şekil 2.25'de verildiği üzere, AE, girdi verisi şifreleme-şifre çözme işleminden sonra çıktı olarak yine aynı girdiyi sağlayana kadar ağırlıklarını değiştirir. AE'lerin çalışma mantığı, hedefe ulaşana kadar ağırlıkların değişmesiyle girdi ve çıktıyı sağlamaya çalışmaktır.



Şekil 2.25. AE'nın katman yapısı

Derin oto kodlayıcırlar (Denoising Autoencoders - DAE), her bir katmandaki çıktıların bir sonraki katmanın girişlerine bağlandığı çok katmanlı AE'lerden oluşan sinir ağıdır. Şekil 2.26'de DAE yapısı verilmiştir.



Şekil 2.26. DAE'nin yapısı

DAE'ler ve AE'ler yönlü sinir ağları olarak bilinmektedir. Birden fazla kullanım alanları vardır. En başlıca kullanım alanları şu şekilde sıralanabilir:

- Doğrusal Olmayan Boyutların Azaltılması
- Tavsiye İnternet Motorlarının Tasarımı
- Özellikleri Sınıflandırma
- Görüntü Tanımlama

2.2.2. Derin Öğrenme Uygulamaları

Derin öğrenme, görüntü işleme, video işleme, doğal dil işleme, biyomedikal sinyal/görüntü işleme alanları sayesinde birçok uygulama geliştirilmiş bu uygulamaların çoğu günlük hayatta kullanılmaktadır.

Günümüzde teknolojinin gelişmesi ile birlikte derin öğrenmenin de kullanım alanları genişlemiştir. Derin öğrenme ile görüntü ayırma, video analizi, sınıflandırma, tanımlama gibi birçok alanda çalışmalar yapılmış ve hala bu çalışmalar devam etmektedir. Bu alanda yapılan çalışmaların bir kaçı aşağıda verilmiştir. Bunlar:

- Karakter tanıma
- Cinsiyet algılama
- Gerçek zamanlı olarak insan hareketlerinin tahmini
- Video ve resimdeki renklerin restorasyonu
- Düşük çözünürlüklü görüntülerin iyileştirilmesi
- Nesne Tanıma
- Spam mail engelleme
- Metin sınıflandırma
- Dil çevirme
- Resimlerin üzerindeki yazıların algılanıp başka bir dile çevrilmisi
- Gerçek zamanlı olarak insan vücudunun algılanıp analiz edilmesi
- Elektronik cihazlarda sesli yardım desteği
- Kendi kendine kullanılan otoman araçlar
- Oyunların bilgisayar tarafından oynanması
- Ünlü ressamların stillerine göre resim çizebilme
- Hava tahmini
- Seçim sonuçlarının tahmini
- Doğal afetlerin tahmin edilmesi
- Video üzerinden ses analizi ile otamatik olarak çeviri yapmak.
- Akıllı evler
- Akıllı ulaşım sistemleri
- Dil Çeviri (Machine Translation)
- Ses Tanıma (Speech Recognition)

2.2.4. Derin Öğrenme Kütüphaneleri ve Yazılımları

Derin öğrenme uygulamaların geliştirilebilmesi, algoritmalarının çalıştırılabilmesi ve problemlerin çözülebilmesi için kütüphanelere ihtiyaç vardır. Günümüzde mevcut kütüphaneler sürekli olarak geliştirmekte ve yeni kütüphaneler oluşturulmaktadır. Derin öğrenme için geliştirilen kütüphaneler aşağıda listelenmiştir;

TensorFlow: Makine öğrenmesi uygulamaları için Google tarafından geliştirilmiş, açık kaynaklı bir platformdur. TensorFlow, sunucularda, cihazlarda ve web gibi birçok platformda model oluşturup eğitme ve dağıtma imkânı sunar. Günümüzde Google, Intel, Twitter, CocoCola, airbnb, DeepMind, GE Healtcare, NeRSC gibi birçok büyük firmalar, TensorFlow kütüphanesini kullanarak uygulamalar geliştirmektedir.

Theano: Derin öğrenme uygulamaları için kullanılan bir Python kütüphanesidir. Direk derin öğrenme uygulamalarında kullanılabildiği gibi üzerine yazılmış Keras veya Lasagne gibi kütüphanelerle de kullanılabilir. Hem CPU'lu makinelerde hem de GPU'lu makineler üzerinde kullanılabilir. Dinamik olarak C kodu üretebilen bir kütüphanedir.

Caffe: Derin öğrenme uygulamalarında kullanılan, ifade, hız ve modülerlik bakımından geniş olan bir kütüphanedir. Berkeley AI Research (BAIR) ve katılımcılar tarafından geliştirilmiştir. Hem CPU'lu makinelerde hem de GPU'lu makinelerde çalışabilir ve aralarında kolay bir şekilde geçiş işlemi yapılabilir. Kullanışlı bir Python ara yüzü imkânı sunar. Önceden eğitilmiş modelleri kullanarak daha kolay bir şekilde derin öğrenme uygulamaları geliştirme imkânı sunar.

Keras: Derin öğrenme uygulamaları için Python dili ile birlikte kullanılan bir kütüphanedir. Keras, Tensorflow, CNTK veya Theano'nun üzerinde çalışabilen bir sinir ağı API'sidir. CPU'lu veya GPU'lu makineler üzerinde çalışmasını bu kütüphaneler üzerinden yapar. Kullanıcı ara yüzü kolay ve üst düzey olduğu için çok tercih edilen bir kütüphanedir.

PyTorch: GPU ve CPU kullanılarak derin öğrenme için kullanılan bir tensör kütüphanesidir. Python dili ile kullanılan bir kütüphanedir. Yeni bir sinir ağı katmanı eklemek için Python'da çeşitli kütüphaneler kullanılabilir ve Cython ile Numba gibi paketler uygulamaya dâhil edilebilir.

MXNet: GPU'ların ve bulut bilişimin tüm özelliklerini kullanan esnek bir derin öğrenme kütüphanesidir. MXNet, büyük ölçekli derin sinir ağları geliştirmeye imkân veren bir kütüphanedir. MXNet, GPU'lar ile Python ve R gibi yüksek seviyeli dillerde

optimize edilmiş, sayısal hesaplama için kullanılır. MXNet, zorunlu programlama (NDArray API tarafından desteklenir) ve sembolik programlama (Symbol API tarafından desteklenir.) adı altında iki programlama seçeneği sunar.

CNTK: Microsoft Bilişsel Araç Seti (Cognitive Toolkit-CNTK) açık kaynaklı bir derin öğrenme kütüphanesidir. CNTK, kullanıcının ileri beslemeli DNN'ler, CNN'ler ve tekrarlayan sinir ağları (RNN'ler / LSTM'ler) gibi günümüzde çok kullanılan model için kolayca oluşturulmasını ve birleştirilmesini sağlar. Python, C# veya C++ programlama dillerine kolayca dâhil edilebilir.

KNet: Yüksek seviyeli bir dil olan Julia üzerinde Deniz Yüret ve ortakları tarafından geliştirilen derin öğrenme kütüphanesidir. Oluşturulan modeller için dinamik hesaplamalı grafikler kullanarak GPU çalışmasını ve otomatik farklılaşma imkânı sağlayan bir kütüphanedir.

The VELES: Hızlı derin öğrenme uygulamaları geliştirmek için kullanılan akış tabanlı bir platformdur. Python'da yazılmış, Apache 2.0 altında OpenCL veya CUDA kullanır. Bulut üzerinde çalışma imkânı veren bir kütüphanedir.

Derin öğrenme için kullanılan kütüphaneler, farklı derin öğrenme mimarilerini desteklemektedir. Çizelge 2.1 'de kütüphanelerin desteklediği derin öğrenme algoritmaları verilmiştir.

Çizelge 2.1. Derin öğrenme kütüphaneleri ve desteklediği mimariler

	CNN	***RNN***	***RBM***	***LSTM***	***Derin Oto Kod.***
TensorFlow	+	+	+	+	+
Theano	+	+	+	+	+
Caffe	+	+	+	+	+
Torch	+	+	-	+	+

Bu çalışmada çevrimdışı olarak el yazısı ile yazılmış metinlerin resim formatına çevrilerek üzerinde tanıma işlemi yapılmıştır. Türkçe sonsuz boyutlu bir kelime hazinesine sahip olduğu için çalışma karakter tabanlı tanıma olarak yapılmıştır. Çalışmada, literatürdeki çalışmalardan farklı olarak karakterlerin sadece kenar özelliklerine göre değil karakterin 2-boyutlu olarak tamamını kapsayan bir çalışma yapılmıştır. Bu işlemlerde, resimler üzerinde özellik çıkarmak için ayrı bir algoritma

kullanılmamıştır. Ayrı bir algoritma yerine Konvolüsyon Sinir Ağları yöntemi ile hem karakterlere ait özellik çıkarma işlemi hem de tanıma işlemi tek bir yöntem ile yapılmıştır. Çalışmada sırayla çevrimdışı olarak el yazı karakterleri üzerinde bir tanıma işlemi gerçekleştirilmiştir. Karakterlerin tanınması ve özelliklerinin çıkarılması için CNN yöntemi kullanılmıştır.

3. MATERYAL VE YÖNTEM

3.1. Teknik Bilgiler

Çalışmada geliştirilen modelin eğitimi ve testi için kullanılan bilgisayarın donanım bilgileri Çizelge 3.1'de verilmiştir.

Çizelge 3.1. Bilgisayarın donamım bilgisi

İşlemci	: Intel Core i5- 4200U
İşlemci hızı	: 2.30 GHz
Ön bellek	: 6 GB
Bellek	: 8 GB
Dâhili disk kapasitesi	: 1 TB
Ekran kartı	: AMD
İşletim sistemi	: Ubuntu 14.04 LTS 64-bit

Yazılım kısmında programlama dili olarak Python 2.7 kullanılmıştır. Görüntü işleme için OpenCv Kütüphanesi, yapay zekâ için Tensorflow ve Keras kütüphaneleri kullanılmıştır. Sınıflandırma bölümünde kullanılan programlama dili ve kütüphaneler hakkında ayrıntılı bilgi sunulmuştur.

3.2. Veri Seti

3.2.1. Eğitim Veri Seti

Modelin eğitiminde kullanılan veri seti Kaggle adlı siteden alınmıştır. Kaggle, Google LLC'nin bir yan kuruluşudur. Kaggle, veri bilimcilerden ve makine öğrenimi uygulayıcılardan oluşan çevrimiçi bir topluluktur. Bu topluluk sayesinde makine öğrenmesi alanında birçok model ve veri seti depolamaktadır. EMNIST veri seti, el yazısı

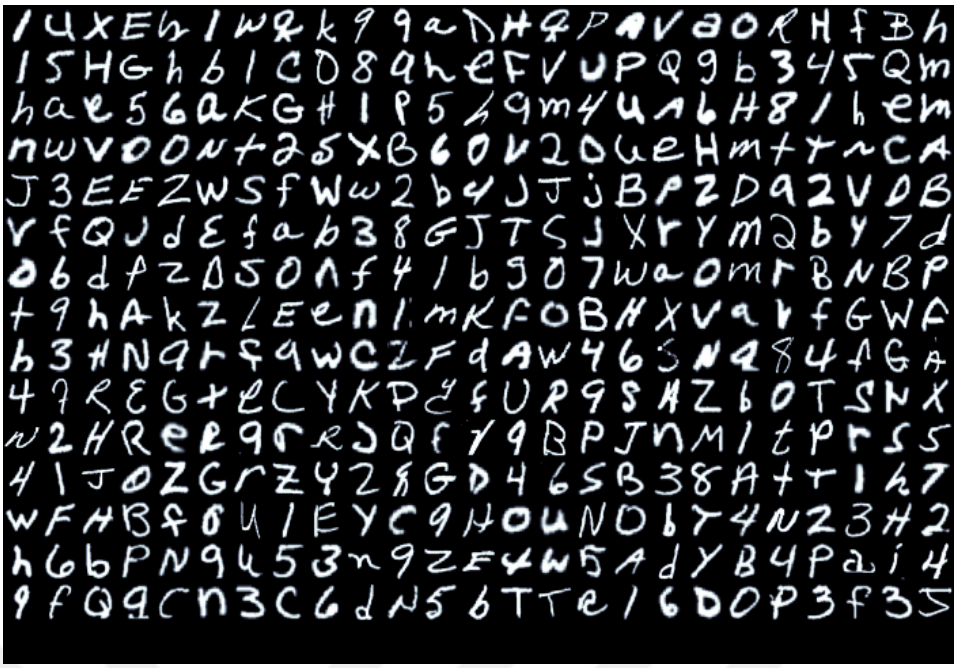
ile yazılmış 28 x 28 piksel görüntülerden oluşan harf ve rakamlar içerir. Şekil 3.1'de EMNIST veri kümesindeki örnek rakam ve harfler verilmiştir. Bu veri kümesinde sağlanan, altı farklı bölme vardır. Veri kümesini kısa özeti (NIST, 2017,April) aşağıdaki gibidir:

- EMNIST ByClass: 814,255 karakter. 62 sınıf değeri.
- EMNIST ByMerge: 814,255 karakter. 47 sınıf değeri.
- EMNIST Balanced: 131,600 karakter. 47 sınıf değeri.
- EMNIST Letters: 145,600 karakter. 26 sınıf değeri.
- EMNIST Digits: 280,000 karakter. 10 sınıf değeri.
- EMNIST MNIST: 70,000 karakter. 10 sınıf değeri.

Çizelge 3.2'e bakıldığında EMNIST veri kümesinin altı farklı bölmesinin veri özeti verilmiştir. Çizelgeye bakıldığında veri kümesinin test ve eğitim için ne kadar veri ayrıldığı, toplam veri sayısı ve doğrulama için kullanılabilirliği verilmiştir.

Çizelge 3.2. EMNIST veri kümesi ve organizasyonu

AD	SINIF	EĞİTİM	TEST	DOĞRULAMA	TOPLAM
BY_CLASS	62	697,932	116,323	Hayır	814,255
BY_MERGE	47	697,932	116,323	Hayır	814,255
BALANCED	47	112,800	18,800	Evet	131,600
DİGİTS	10	240,000	40,000	Evet	280,00
LETTERS	37	88,800	14,800	Evet	103,600
MNIST	10	60,000	10,000	Evet	70,000



Şekil 3.1. EMNIST veri kümesi

Çalışmada modelin eğitimi için EMNIST veri setinin Balanced kısmı kullanılmıştır. Balanced veri seti, 112,800 eğitim verisi, 18,800 test verisi olmak üzere toplamda 131,600 adet veriden oluşmaktadır. Bu veriler el yazısı ile yazılmış 47 adet harf ve rakamların görüntülerini içermektedir.

3.2.2. Test Veri Seti

Bu veri seti, çevremizdeki birçok kişi tarafından yazılan belgeler ile oluşturulmuştur. Kâğıt üzerindeki el yazısı ile yazılmış karakterlerin tanınması için oluşturulmuş veri setidir. Veri seti 10 adet belge içermektedir. Bu belgeler farklı kişiler tarafından el yazısı ile yazılmış, tarayıcı veya kamera yoluyla bilgisayar ortamına aktarılmış belgelerdir.

3.3. Geliştirilen Derin Öğrenme Algoritması ile El Yazısı Tanıma

Bu çalışmada gerçekleştirilen işlemler aşağıdaki adımlara bölünmüştür. Her bir adımdaki işlemler Python dili kullanılarak uygulaması geliştirilmiştir. Şekil 3.2'de geliştirilen uygulamanın adımları gösterilmiştir.

Şekil 3.2. Geliştirilen uygulamanın adımları

3.3.1. Model Oluşturma

Çalışmada karakterlerin sınıflandırılması için derin öğrenme metodu kullanılmıştır. Derin öğrenmenin Konvolüsyon sinir ağı mimarisi kullanılarak bir model oluşturulmuş ve sınıflandırma işlemi bu model kullanılarak yapılmıştır.

3.3.1.1. Derin Öğrenme

Derin öğrenme, birden fazla doğrusal olmayan katman yardımı ile özellik çıkarma işlemi yapar. Ardışık olan tüm katmanlar önceki katmandaki çıktıyı girdi olarak alır (Deng & Yu, 2014; Şeker et al., 2017). Derin öğrenme temel olarak veriyi en iyi temsil eden piksellerin değerlerine göre öğrenmeye dayalı bir yöntemdir. Bu yönteme özellik çıkarma yöntemi denir. Derin öğrenmede verilerin elle çıkarılması yerine görüntü üzerinde en iyi pikselleri çıkarmak için değişik algoritmalar kullanılmaktadır.

Derin öğrenmenin tarihine bakıldığında denetimli derin beslemeli çok katmanlı perceptronlar için ilk algoritma Ivakhnenko ve Lapa tarafından 1965 yılında ortaya çıkmıştır (Ivakhnenko & Lapa, 1966). Bu çalışmada, katmanlardaki en iyi özellikler istatistiksel yöntemlerle belirlenip bir sonraki katmana gönderilmektedir. Ağlarını uçtan uca eğitmek için geri yayılım (backpropagation) kullanılmamıştır.

İkinci olarak derin öğrenme mimarisini 1979 yılında Fukushima önermiştir. Omurgalı canlıların sinir sisteminden esinlenerek geliştirilmiştir. Fukushima'nın ağları günümüzdeki ağlara benzer olarak çoklu bükülme ve havuz katmanlarını içermektedir (Fukushima et al., 1983).

Derin öğrenme mimarileri önceki yıllarda ortaya çıkmış olmasına rağmen ilk defa başarılı bir derin sinir ağı uygulamasını Yann LeCun ve arkadaşları tarafından posta

kutusu yazıları üzerinde geliştirmişlerdir (LeCun et al., 1989; Şeker et al., 2017). Bu çalışmadan sonra Yann LeCunn "LeNet" ağını kullanarak el yazısı rakamlarını (MNIST) sınıflandırmak için kıvrımlı ağlarla geri yayılımı birlikte uygulamıştır (Le Cun, Boser, et al., 1989). Bu çalışma ile karakter tanımadaki ilk işlemler yapılmaya başlanmıştır.

Günümüzde gelişen teknoloji ile bu alana olan ilgide artmıştır. Derin öğrenme (deep learning) ifadesi ilk kez 2000 yılında Igor Aizenberg ve arkadaşları tarafından ortaya atılmıştır (Aizenberg et al., 2000).

Daha sonra bu alanda Geoffrey Hinton 2006'da, çok katmanlı ileri beslemeli bir sinir ağının nasıl eğitileceğini ve denetimli bir geri yayılım yöntemini nasıl yapacağını makalesinde anlatmıştır (Hinton, 2007).

Bilgisayarların gelişmesi GPU ve CPU'nun hızlanmasıyla derin ağların ön-eğitim (pre-training) yapılmadan eğitilmesi ortaya çıkmıştır. Ciresan ve arkadaşlarına trafik işaretleri, medikal görüntüleme ve karakter tanıma gibi uygulamalarda bu yöntemi kullanmıştır (Ciresan et al., 2011).

Krizhevsky, Sutskever ve Hinton 2012 yılında benzer mimariler kullanmışlar ve GPU kullanılarak yaptıkları çalışmalarda, ezberlemeyi azaltmak için "dropout" katmanını geliştirmişler ve uygulamalarında bu yöntemi kullanmışlardır (Hinton et al., 2012).

Derin öğrenmenin bu hızla gelişmesinin üzerine Google, Facebook ve Microsoft gibi büyük teknoloji firmaları derin öğrenme alanında yatırım yapmaya başlamışlardır.

Geçmişten günümüze derin öğrenme ile birçok uygulama geliştirilmiş ve geliştirilmeye de devam etmektedir. Bu süreçte derin öğrenme alanında çok çalışma yapılmış ve değişik mimariler geliştirilmiştir. Derin öğrenmenin 6 tane mimarisi vardır. Bunlar;

- Konvolüsyon Sinir Ağları
- Tekrarlayan Sinir Ağıları
- Uzun / Kısa Süreli Bellek
- Kısıtlı Boltzmann Makineleri
- Derin İnanç Ağlar
- Derin Oto Kodlayıcılar ve Oto Kodlayıcılar

Konvolüsyon sinir ağları: Konvolüsyon sinir ağları (Convolution Neural Network - CNN) çok katmanlı algılayıcıların (Multi Layer Perceptron –MLP) farklı bir türüdür. Tarihte ilk CNN ağı 1988 yılında Yann LeCun tarafından bulunan LeNet mimarisidir (Le Cun, Jackel, et al., 1989).

CNN algoritmaları görüntü ve ses işleme (Mushtaq et al., 2020; Su et al., 2019), doğal dil işleme (NLP) (Akhtyamova et al., 2017; Sun et al., 2019), biyomedikal görüntü işleme (Cho et al., 2020; Momeni et al., 2018) gibi birçok farklı alanda kullanılmaktadır. Özellikle görüntü işleme alanında en iyi sınıflandırma başarısına sahip olan derin öğrenme algoritmasıdır.

CNN mimarisi veriyi çeşitli katmanlarda işler. CNN modelinin katman yapısı Şekil 3.3'da olduğu gibidir. Bu katmanlar (Ergin, 2018,October):

- Convolutional layer - Görüntü üzerindeki özellikleri çıkarmak için kullanılır.
- Non-Linearity layer - ReLU katmanı – Aktivasyon işleminin bulunduğu katmandır. Sistemde doğrusal olmayan bir fonksiyon kullanır.
- Pooling(Downsampling) layer - Boyut azaltma ve uygunluk kontrolu yapar
- Flattening layer – Standart Sinir Ağı için tek boyutlu vektör verilerini hazırlar.
- Fully-Connected layer – Sınıflandırma için Standart Sinir Ağı kullanır.

Convolution katmanı görüntü üzerindeki özelliklerini algılamak için kullanılır. Bu katmanda görüntü üzerindeki özelliği çıkartmak için görüntünün boyutundan daha küçük olan düşük veya yüksek seviyeli filtreler kullanılarak yapılır. Bu filtreler genelde tek sayılardan oluşan matrislerdir. Bu Filtreleri görüntü üzerinde gezdirerek ve matris çarpımı kullanarak, özellikler tespit edilmeye çalışılır. Filtrelerden sonra resmin orijinal boyutunu kaybetmemesi için sıfır değerleri eklenir.

Convolution katmalarından sonra Non-Linearity katmanı gelir. Bu katman, aktivasyon katmanı olarak da bilinir ve aktivasyon fonksiyonlarından biri kullanılır. Bu katmanın amacı aktivasyon fonksiyonlarını kullanarak modelin negatif değerleri öğrenmesini ya da bazı özellikleri kavrayamamasını engellemektir. Genellikle aktivasyon fonksiyonu olarak; ReLU, tanh ve sigmoid gibi doğrusal olmayan fonksiyonlar kullanılır. Hızı konusunda en iyi sonuçları Ractifier (Relu) fonksiyonu verdiği için eğitimlerde çoğunlukla bu fonksiyon kullanılır.

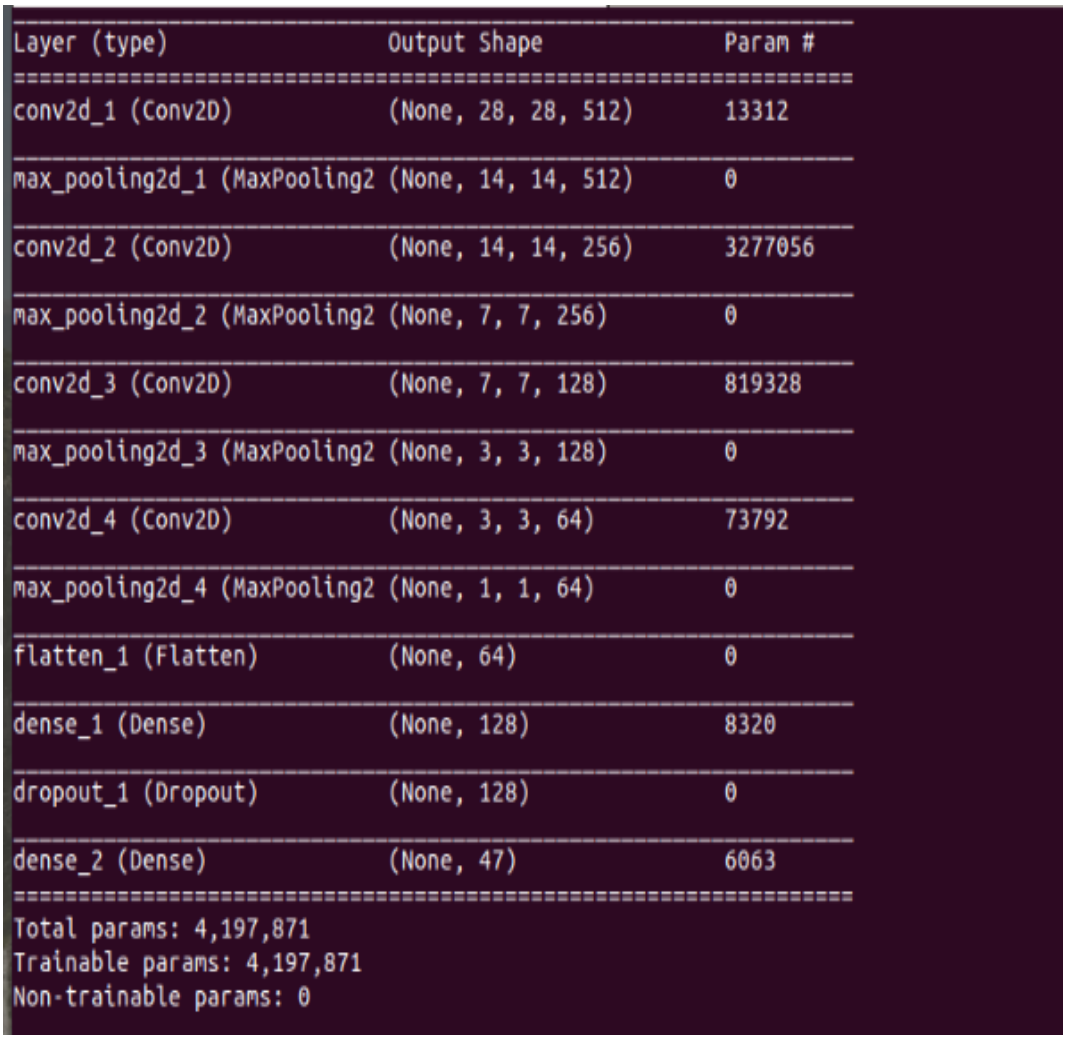
Pooling (havuzlama) katmanı, ardışık convolution katmanları arasına eklenir. Bu katman boyutsallığını azaltmak için kullanılmaktadır. Bu katman sayesinde çıkarılan gereksiz özellikler yok edilerek önemli olan özelliklere odaklanma sağlanır. CNN modellerde genellikle Max (Maksimum) ve Average (Ortalama) olarak iki farklı pooling tekniği kullanılır. Birinci teknik Maxpooling de öncelikle bir filtre oluşturulur ve bu filtre resim üzerinde gezdirilerek kapladığı alandaki en büyük sayıyı alır. İkinci teknik Averagepooling de ise yine bir filtre oluşturulur ve bu filtre resim üzerinde gezdirilerek kapladığı alandaki piksellerin ortalaması olan sayı alınır.

Fatting katmanı, Fully Connected Layer için giriş verilerini hazırlayan katmandır. Sinir ağlarının giriş verilerini tek boyutlu bir dizi olarak alır. Bu verileri ise, Convolution ve Pooling katmanlarından gelen matrislerdeki verilerdir. Fatting katmanıda bu verileri tek boyutlu diziye çevirme işlemini yapar.

Fully Connected Layer katmanı, verileri Flattening katmanında işlenip tek boyutlu diziye çevrilmiş şekilde alır ve klasik sinir ağı yöntemiyle öğrenme işlemini gerçekleştirir. Eğitim sırasında under fitting , gereksiz ezberleme, over fitting gibi problemlerini önlemek için Dropout yöntemini kullanılır.

3.3.1.2. Model Oluşturma ve Eğitim

Sınıflandırma adımında derin öğrenme algoritması için, Keras kütüphanesi kullanılarak Python dilinde geliştirilen bir uygulama yapılmıştır. Eğitim için EMNIST veri kümesinin Balanced kısmı kullanılmıştır. Modelin katmanları da, Konvolüsyon sinir ağı mimarisi kullanılarak oluşturulmuştur. Geliştirilen modelin katmanları; 4 adet CONV2D katmanı, 4 adet MaxPooling2 katmanı, 1 adet Flatten katmanı, 1 adet Dropout katmanı ve 2 adet Dense katmanından oluşmaktadır. Şekil 3.3'de geliştirilen uygulamada modelin katman yapısının çıktısı gösterilmiştir.



Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 28, 28, 512)	13312
max_pooling2d_1 (MaxPooling2	(None, 14, 14, 512)	0
conv2d_2 (Conv2D)	(None, 14, 14, 256)	3277056
max_pooling2d_2 (MaxPooling2	(None, 7, 7, 256)	0
conv2d_3 (Conv2D)	(None, 7, 7, 128)	819328
max_pooling2d_3 (MaxPooling2	(None, 3, 3, 128)	0
conv2d_4 (Conv2D)	(None, 3, 3, 64)	73792
max_pooling2d_4 (MaxPooling2	(None, 1, 1, 64)	0
flatten_1 (Flatten)	(None, 64)	0
dense_1 (Dense)	(None, 128)	8320
dropout_1 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 47)	6063

Total params: 4,197,871
Trainable params: 4,197,871
Non-trainable params: 0

Şekil 3.3. Çalışmada kullanılan modelin katmanların açıklaması

Bu yapılan çalışmada kullanılan modelin katmanların açıklamaları aşağıda verilmiştir:

CONV2D: Convolution katmanıdır. Görüntü üzerinde özellikleri saptamak için kullanılan katmandır.

MaxPooling2: Pooling (havuzlama) katmanı, ardışık convolution katmanları arasında eklenir. Bu katman boyutsallığı azaltmak için kullanılır.

Flatten: Fatting katmanı**,** Fully Connected katmanı için giriş verilerini hazırlayan katmandır.

Dropout: Eğitim sırasında under fitting, gereksiz ezberleme, over fitting promlemlerini önlemek için kullanılır.

Dense: Fully Connected katmanıdır.

Modelin eğitimi, EMNIST veri setinin Balanced kısmı kullanılarak yapılmıştır. Model farklı epochs ve batch_size değerlerine göre 4 kere eğitilmiştir. Eğitim setinin

%10’u validation (doğrulama) için ayrılmıştır. Modelin eğitiminde kullanılan epochs ve batch_size değerleri aşağıdaki gibidir;

- 5 epochs / 512 batch_size
- 5 epochs / 1024 batch_size
- 10 epochs / 512 batch_size
- 10 epochs / 1024 batch_size

Modelin katmaları ve eğitim kısmında kullanılan kodlar Şekli 3.4’de gösterilmiştir. Modellin oluşturulması ve eğitilmesi için gereken parametrelerin açıklaması aşağıda verilmiştir.

Sequential (): Sıralı bir model oluşturmak için kullanılır.

Add (): Modele katman ekler.

Compile (): Modeli derlemek için kullanılır. Model “categorical_crossentropy” olarak derlenmiştir.

Fit (): Modelin eğitimini yapar.

Epochs: Modelin kaç adımda eğitileceğini belirler.

Batch_size: Her bir adımda veri setinde kaç tane veri kullanacağını belirler.

Validation_data: Modelin doğrulamasında kullanılacak veriler.

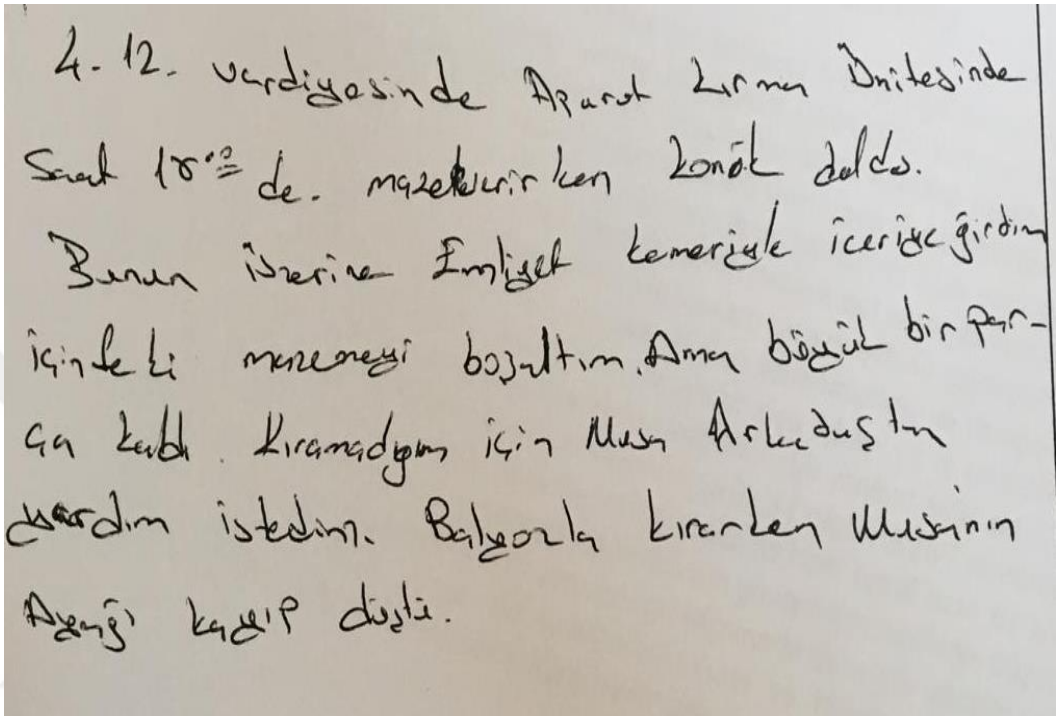
```
model = Sequential()
model.add(Conv2D(filters=512, kernel_size=(5,5), padding = 'same', activation='relu', \
    input_shape=(HEIGHT, WIDTH,1)))
model.add(MaxPooling2D(pool_size=(2,2), strides=(2,2)))

model.add(Conv2D(filters=256, kernel_size=(5,5), padding = 'same', activation='relu',\
    input_shape=(HEIGHT, WIDTH,1)))
model.add(MaxPooling2D(pool_size=(2,2), strides=(2,2)))

model.add(Conv2D(filters=128, kernel_size=(5,5), padding = 'same', activation='relu',\
    input_shape=(HEIGHT, WIDTH,1)))
model.add(MaxPooling2D(pool_size=(2,2), strides=(2,2)))

model.add(Conv2D(filters=64, kernel_size=(3,3) , padding = 'same', activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))

model.add(Flatten())
model.add(Dense(units=128, activation='relu'))
model.add(Dropout(.5))
model.add(Dense(units=num_classes, activation='softmax'))

model.summary()

model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
history = model.fit(train_x, train_y, epochs=1, batch_size=100, verbose=1, \
                    validation_data=(val_x, val_y))
model.save('my_model.h5')
model = keras.models.load_model('my_model.h5')
```

Şekil 3.4. Modelin katmanları ve eğitim için kullanılan parametrelerinin kodları

3.3.2. Ön İşleme

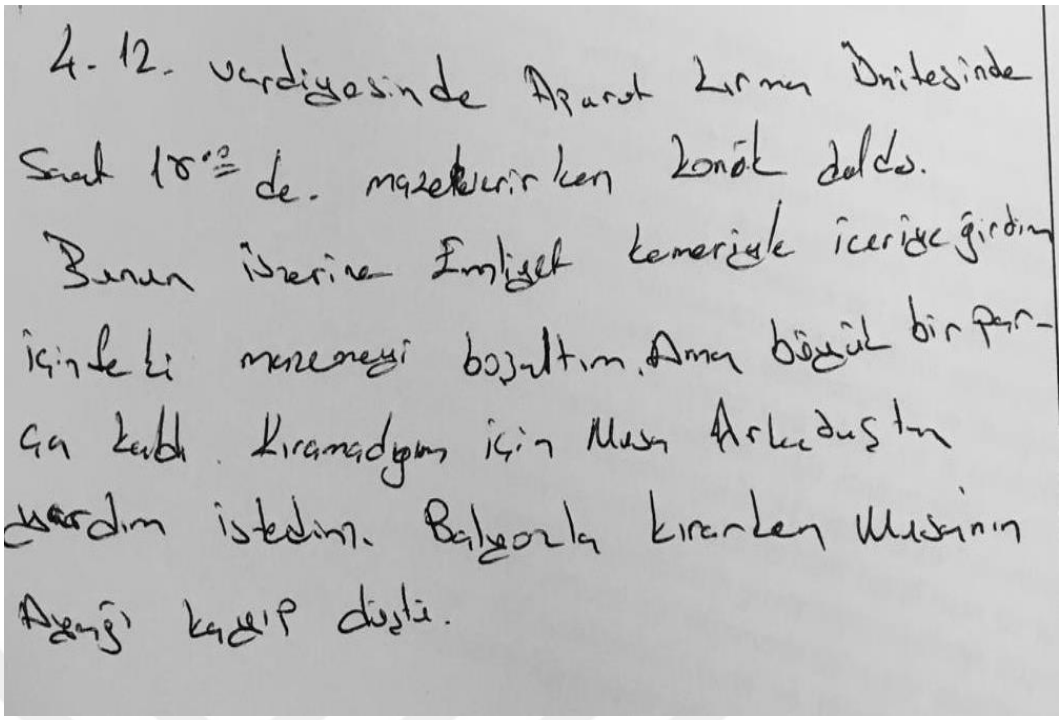
Bu çalışmada karekter tabanlı bir el yazısı tanıma sistemi yapılmıştır. Bu sebeple test için oluşturulan veri setindeki belgeleri karakterlerine ayırmak için bazı işlemler yapılmıştır. Ön işleme için OpenCV Kütüphanesi kullanılarak bir uygulama geliştirilmiştir. Ön işleme adımları sırası ile açıklanmıştır. Bu adımlar;

1. Kâğıt üzerindeki yazılar, tarayıcı veya kamera yardımı ile bilgisayar ortamına ".png" formatında kaydedilmiştir. PNG formatı, en az kayıpla sıkıştırma yaptığından tercih edilmiştir. Şekil 3.5'de kamera yardımı ile bilgisayar ortamına atılmış ve ".png" uzantılı orijinal görüntü gösterilmiştir.



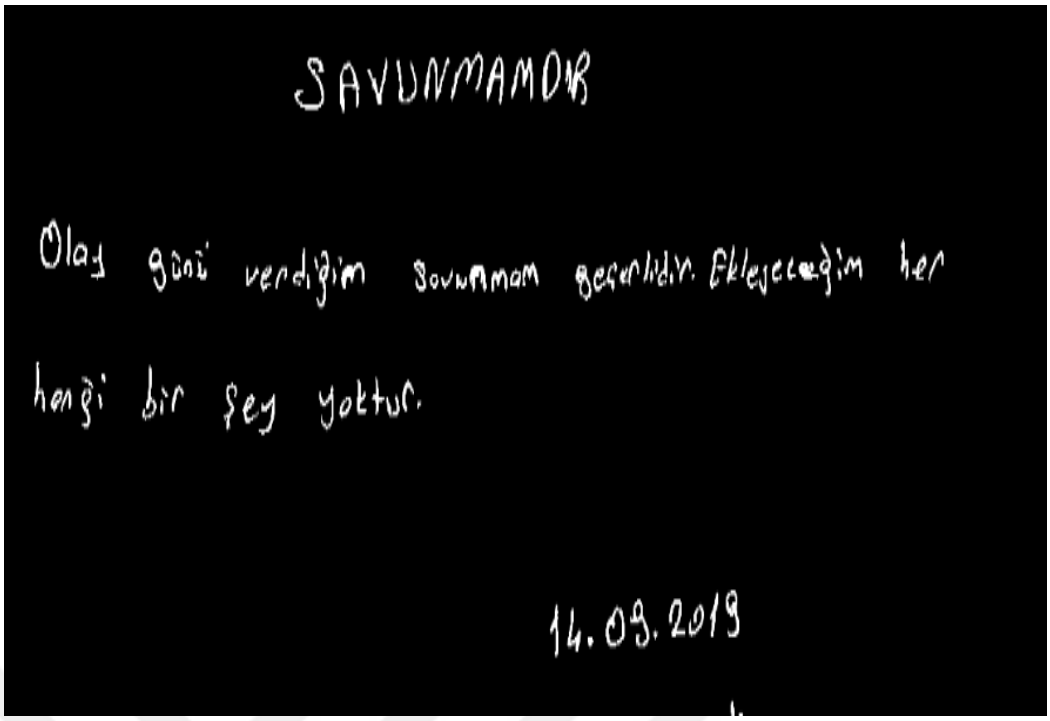
Şekil 2 **Şekil 3.5.** Kamera yardımı ile bilgisayara atılan belge

2. Orijinal görüntü ilk olarak cvtColor fonksiyonu kullanılarak gri seviyede bir görüntüye çevrilmiştir. CvtColor fonksiyonu, görüntüyü bir renk uzayından diğer renk uzayına çevirme işlemidir. Bu fonksiyon kullanırak RGB renk uzayındaki görüntü GREY renk uzayına yani tek kanallı 0 ile 255 arasında bir renk değeri alarak gri seviyeli bir görüntüye çevrilmiştir. Şekil 3.6 de gri seviyeye çevrilen bir görüntü örneği sunulmuştur.



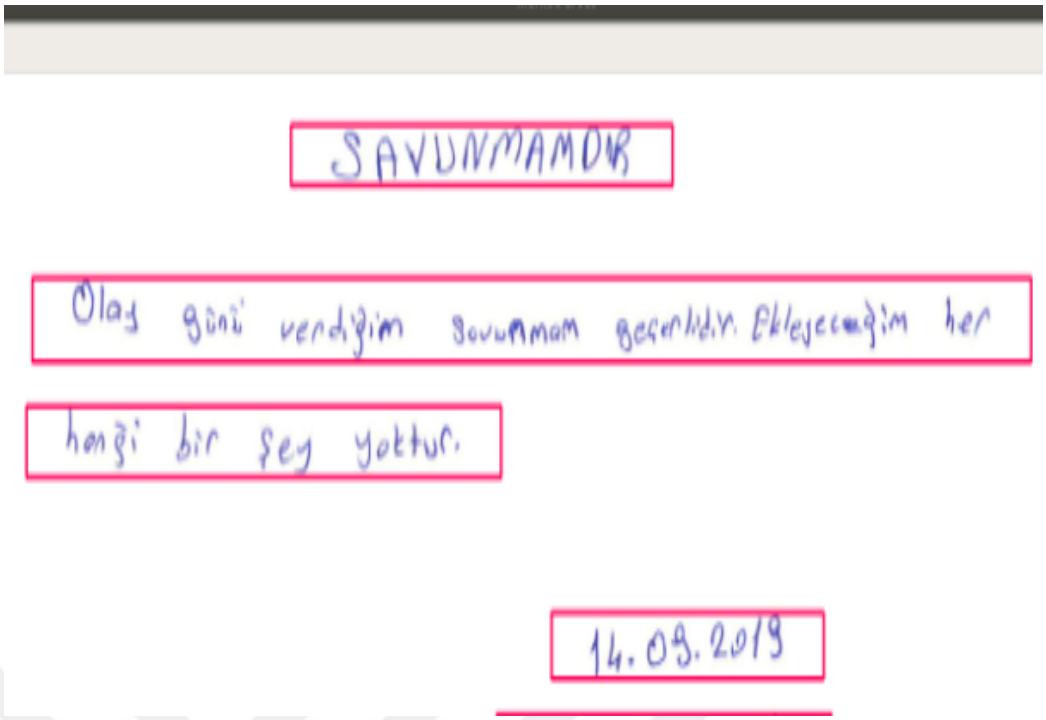
Şekil 3.6. Gri seviye görüntü

3. Gri seviyedeki görüntü threshold fonksiyonu uygulanarak ikili (binary) görüntüye yani siyah-beyaz görüntüye çevrilmiştir. Threshold işlemi pikselleri, verilen eşik değerine göre siyah (0) ya da beyaz (255) olarak güncelleme işlemidir. Bu işlem sayesinde görüntü üzerinde karakterlerin yazılı olduğu yerle arka plan ayrımı yapılmıştır. Bu işlem sonrası görüntü Şekil 3.7'de göründüğü gibi arka plan siyah, karakterler ise beyaz olarak ayrılmıştır.

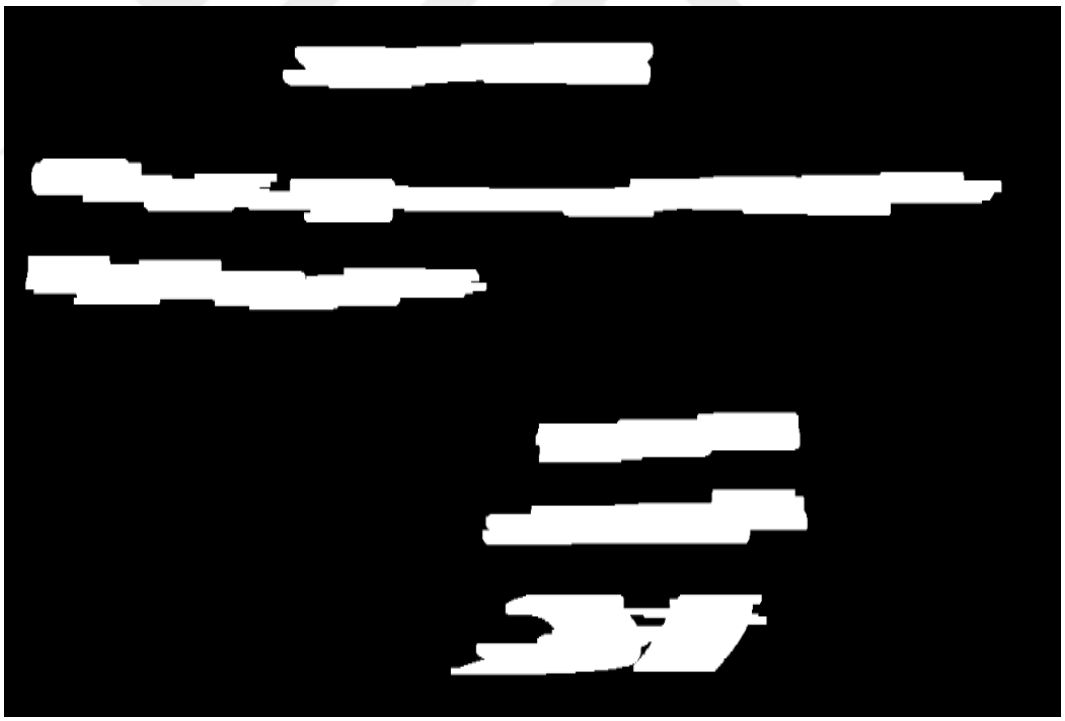


Şekil 3.7. Siyah-beyaz görüntü

4. Siyah-beyaz görüntü üzerinde morfolojik dönüşümler uygulanarak satırlar belirlenmiştir. Morfolojik dönüşümler, görüntü üzerindeki basit işlemlerdir. Erozyon ve genişleme olarak iki temel işlemi vardır. Satırları belirlemek için siyah-beyaz görüntü üzerinde genişleme (dilation) işlemi yapılmıştır. Dilation fonksiyonu kullanılarak metin üzerindeki yazılar genişletilmiştir. Bu işlemden sonra kontor çıkarma işlemi ile aynı renk ve yoğunluğa sahip olan tüm kesintisiz noktaları sınır boyunca birleştiren bir eğri çizilerek satırlar belirlenmiştir. Şekil 3.9'da siyah-beyaz görüntü üzerinde dilation işleminin sonucu gösterilmiştir. Bu işlemden sonra belirlenen kontor listesi kullanılarak orijinal görüntü üzerinde satırlar çizilmiştir. Şekil 3.8'de satırlara ayrılmış bir görüntü gösterilmiştir.



Şekil 3.8. Satırlarına ayrılmış görüntü

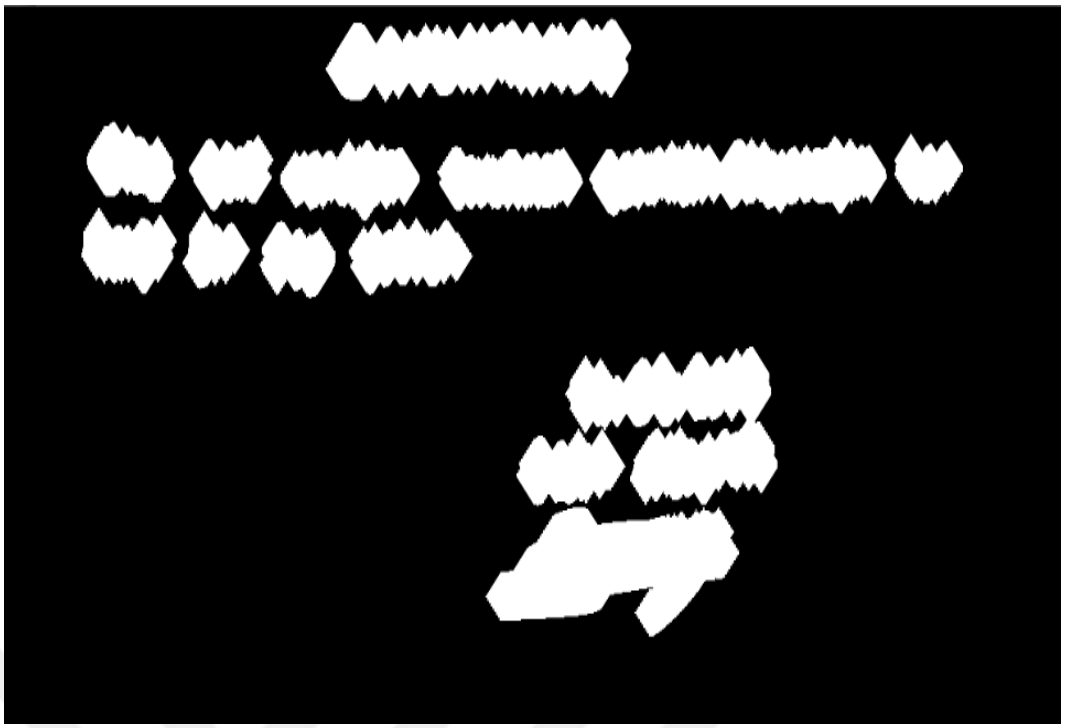


Şekil 3.9. Genişleme (dilation) işlemi uygulanmış görüntü

5. Satırlar üzerinde yeniden morfolojik dönüşüm işlemi uygulanarak belge kelimelere ayrılmıştır. Bu adımda kullanılan dilation fonksiyonunda kernel değeri farklı seçilerek satırlardaki kelimeler belirlenmiştir. Şekil 3.11' de örnek görüntü gösterilmiştir. Dilation işleminden sonra görüntü üzerinde 'findContours'

fonksiyonu kullanılarak kontur çıkarma işlemi uygulanmıştır. Kontur çıkarma işlemi, aynı yoğunluğa ve renk değerlerine sahip sürekli olan piksellerin belirlenmesidir. Bu sayede satırlardaki kelimeleri ayırma işlemi yapılmıştır. Şekil 3.10'da siyah-beyaz görüntü üzerinde kelimelerine ayrılmış bir görüntü gösterilmektedir.

Şekil 3.10. Kelimelerine ayrılmış görüntü



Şekil 3.11. Farklı kernel kullanılarak uygulanmış dilation uygulanmış görüntü

6. Ayrılan kelimeler üzerinde morfolojik dönüşüm ve kontur çıkarma işlemi yapmıştır. Bu işlem sayesinde kelimeler içindeki karakterin sınırları belirlenmiştir. Sınırları belirlenen karakter üzerinde ROI (Region of Interest-İlgi Bölgesi) işlemi yapılmıştır. Bu işlem, ilgili bölgenin piksel değerlerine göre sadece karakterin sınırları olan yeni bir görüntü elde edilme işlemidir. Elde edilen yeni görüntüler Şekil 3.12'de olduğu gibi siyah-beyaz formattaki karakterlerdir. Karakterler 28 x 28 boyutunda görüntüler şeklinde klasöre kaydedilmiştir.

Şekil 3.12. Test etmek için kullanılan karakterler

7. Mevcut karakterlerle oluşturulan veri setine normalizasyon işlemi uygulanmıştır. Normalizasyon işlemi, piksel yoğunluklarını ayarlayarak görüntüyü standart bir hale getirme işlemidir. Bu işlem sayesinde görüntüdeki piksel yoğunluğu standart hale getirilerek daha iyi sınıflandırma yapılması amaçlanmıştır.

Bu aşamada yukarıda açıklaması yapılan adımların Python programlama dili kullanılarak uygulaması geliştirilmiştir. Çizelge 3.3'de verilen Algoritma 1 yardımı ile bu kod çalıştırılmış; test veri setindeki tüm belgeler karakterlerine ayrılmıştır.

Çizelge 3.3 Ön işlem algoritması

Algoritma 1: (Ön işlem)
Girdi: Veri seti içerisindeki N adet görüntü **Çıktı:** Siyah- beyaz tonlamalı karakter görüntüleri içeren veri set
Adım 1. Başla **Adım 2.** Döngü (N tane görüntü) - Klasör içerisinden görüntü yükle. - Görüntüyü gri seviyeye çevir. - Gri seviyeli görüntüyü siyah- beyaz görüntüye çevir. - Morjolojik işlem uygulayarak satırları belirle. **Adım 3.** Döngü (Tespit edilen satır kadar)

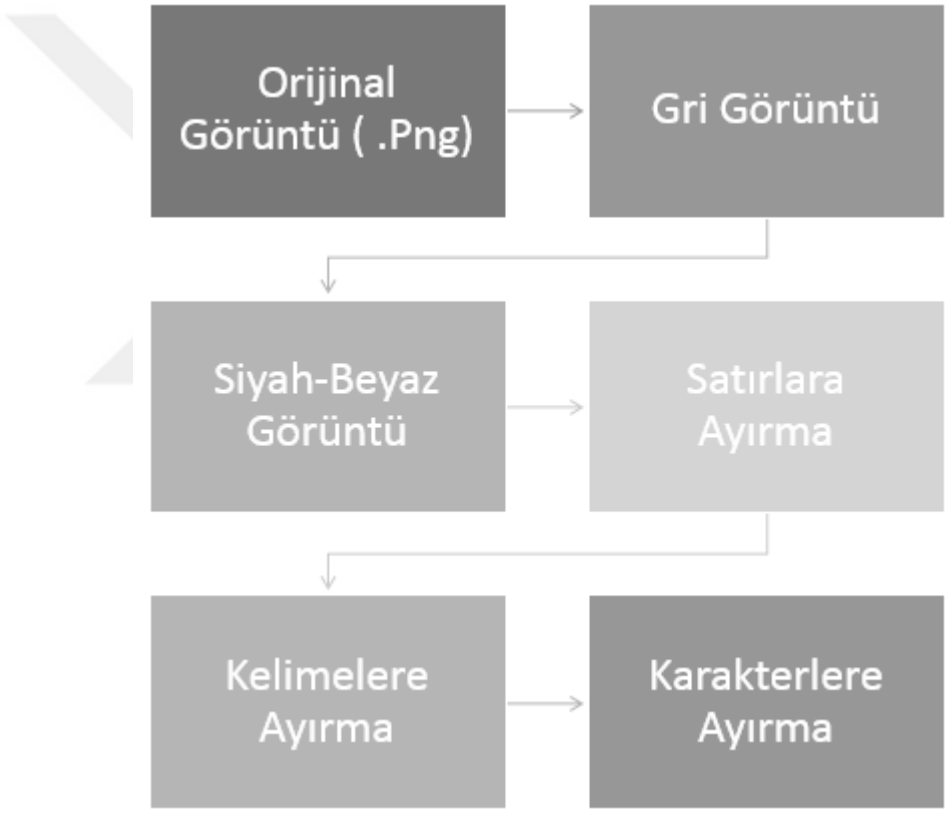
- Satırlara ayrılan görüntüde dilation işlemi uygulararak kelimelere ayır.

Adım 4. Döngü (Tespit edilen kelime kadar)

- Kelimelere ayrılan görüntüde morfolojik işlemler ve kontor çıkarma uygulayarak karakterleri tespit et.
- Yerleri tespit edilen karakterler üzerinde ROI işlemi uygula
- Karakter görüntülerine normalizyon uygula.

Adım 5. Bitir

Çalışmanın ön işleme adımına ait blok diyagram Şekil 3.13'de verilmiştir.

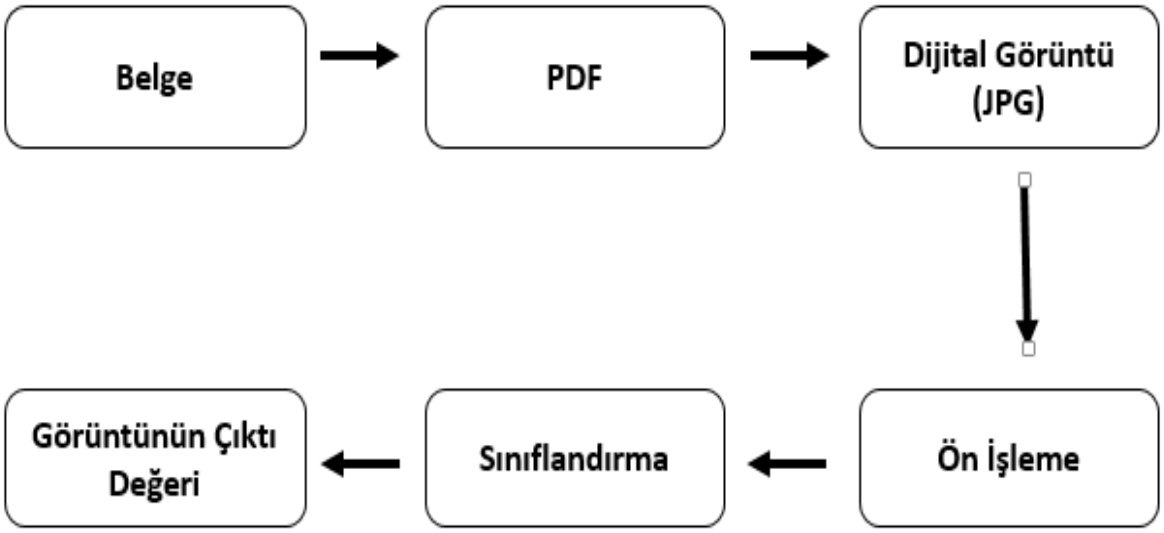


Şekil 4.13. Ön işlemenin blok diyagramı

3.3.3. Sınıflandırma

Bu aşamada, ön işlemden geçen karakterler üzerinde sınıflandırma işlemi yapılmıştır. Sınıflandırma uygulaması Python dili kullanılarak geliştirilmiştir. Uygulamada test veri setindeki belgeler ön işleme adımı uygulanarak karakterlerine ayrılmıştır. Daha sonra bu karakterler kullanılarak modellerin tahmin işlemleri

yapılmıştır. Şekil 3.14'te bu adımların blok diyagramı gösterilmiştir. Modelin "predict" fonksiyonu kullanılarak karakterler üzerinde tahmin işlemi yapılmıştır. Şekil 3.15'de modelin test veri seti üzerinde tahmin işleminin kodları verilmiştir.



Şekil 3.14. Sınıflandırma aşamasının blok diyagramı

```
np_load_old = np.load
np.load = lambda *a,**k: np_load_old(*a, allow_pickle=True, **k)
#test_data = process_test_data()

test_data = np.load('test_data.npy')

fig=plt.figure()

for num,data in enumerate(test_data[:12]):
    img_num = data[1]
    img_data = data[0]
    data = img_data.reshape(-1,IMG_SIZE,IMG_SIZE,1)
    model_out = model.predict([data])[0]
    y = fig.add_subplot(3,4,num+1)
    orig = img_data
    #str_label.append((model_out))
    y.imshow(orig,cmap='gray')
    #print str(model_out)
    plt.title(chr(mapp[model_out.argmax()]))
    y.axes.get_xaxis().set_visible(False)
    y.axes.get_yaxis().set_visible(False)
plt.show()
```

Şekil 3.14. Eğitilen modelde test etmek için kullanılan kod bloğu

4. ARAŞTIRMA SONUÇLARI VE TARTIŞMA

Oluşturulan model, EMNIST veri seti için farklı epochs ve batch_size değerleri kullanılarak eğitilmiş ve sonuçları elde edilmiştir. Çalışmalar yapılırken daha doğru bir el yazısı tanıma için model 4 kere eğitilmiştir. Test veri seti ile de modellerin başarı sonuçları bu bölümde paylaşılmıştır.

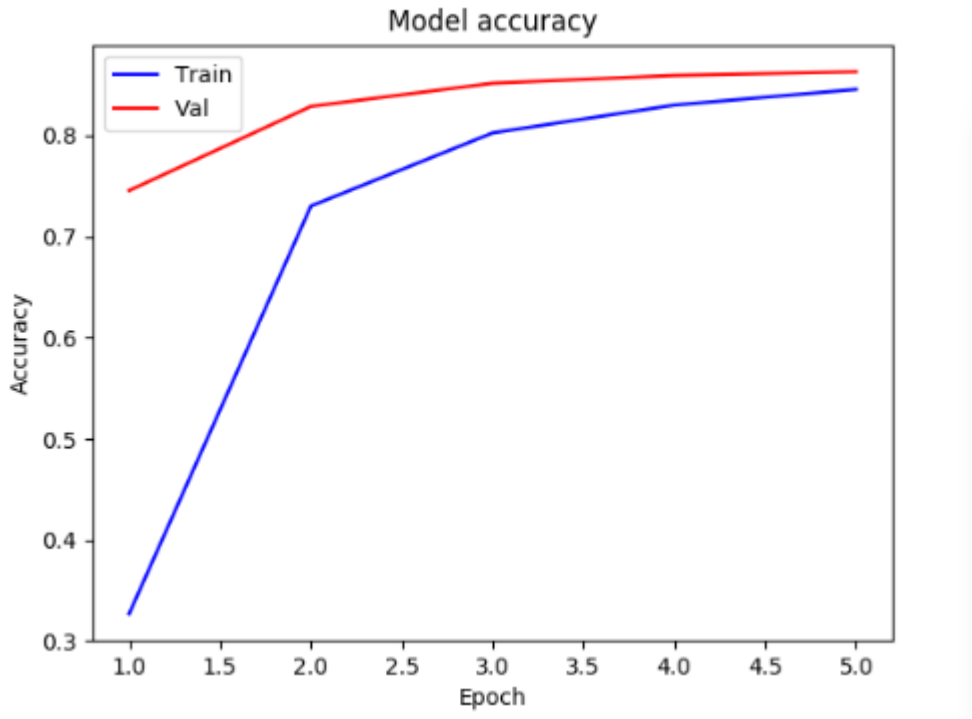
4.1. 5 Epoch ve 512 Batch_size

Modelin, EMNIST veri seti ile 5 epoch ve 512 batch_size değerleri kullanılarak ilk eğitimi yapılmıştır. Eğitimde istenilen başarıya ulaşılamamıştır.

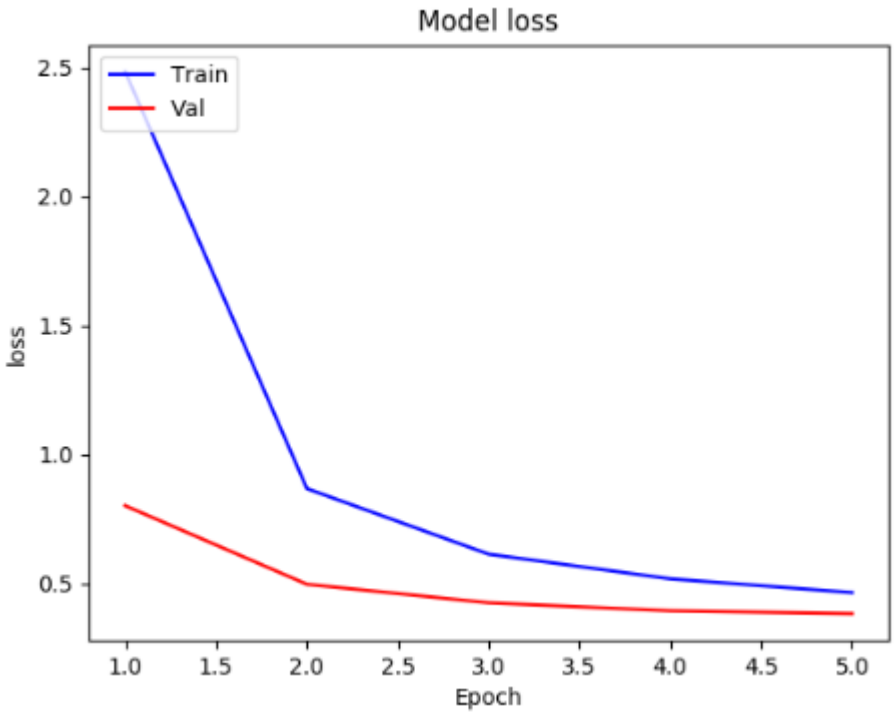
4.2. 5 Epoch ve 1024 Batch_size

Modelin, EMNIST veri seti kullanılarak 5 epoch ve 1024 batch_size değerleri ile ikinci kez eğitimi yapılmıştır. Eğitimin başarı (accuracy) ve kayıp (loss) grafiği Şekil 4.1 ve Şekil 4.2'de verilmiştir.

Şekil 4.1'e bakıldığında modelin train ve validation için ayrılan veriler üzerindeki başarı oranı verilmiştir. Modelin epoch değeri artıkça başarısının arttığı gözlemlenmiştir. Şekil 4.2'de modelin kayıp oranı verilmiştir. Epochs değeri arttıkça kayıp oranında bir azalma görülmüştür.



Şekil 4.1. Modelin 5 epochs ve 1024 batch_size değerlerine göre yapılan eğitimin başarı grafiği

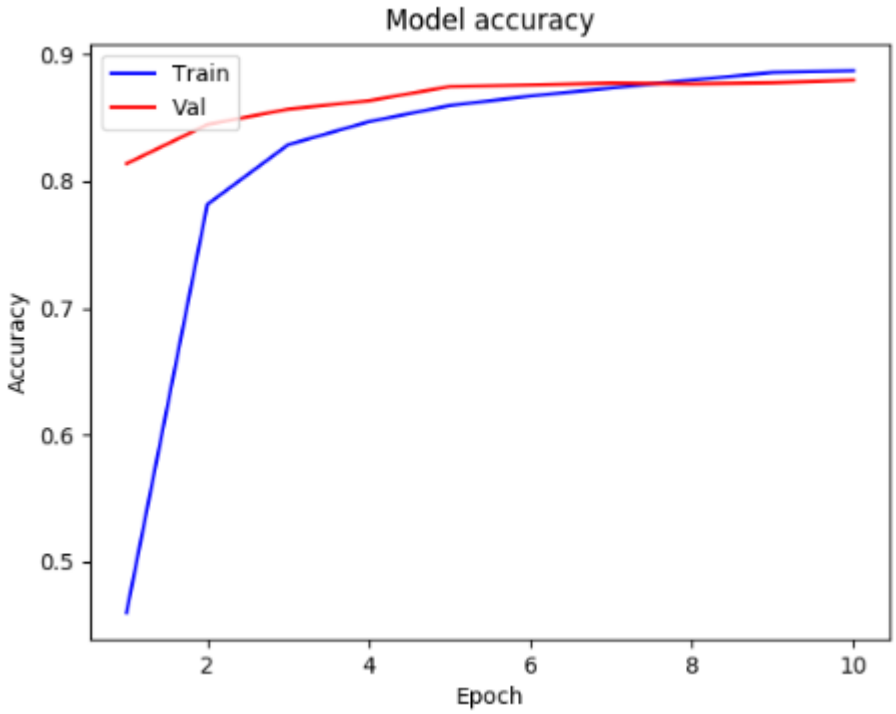


Şekil 4.2. Modelin 5 epochs ve 1024 batch_size değerlerine göre yapılan eğitimin başarı grafiği

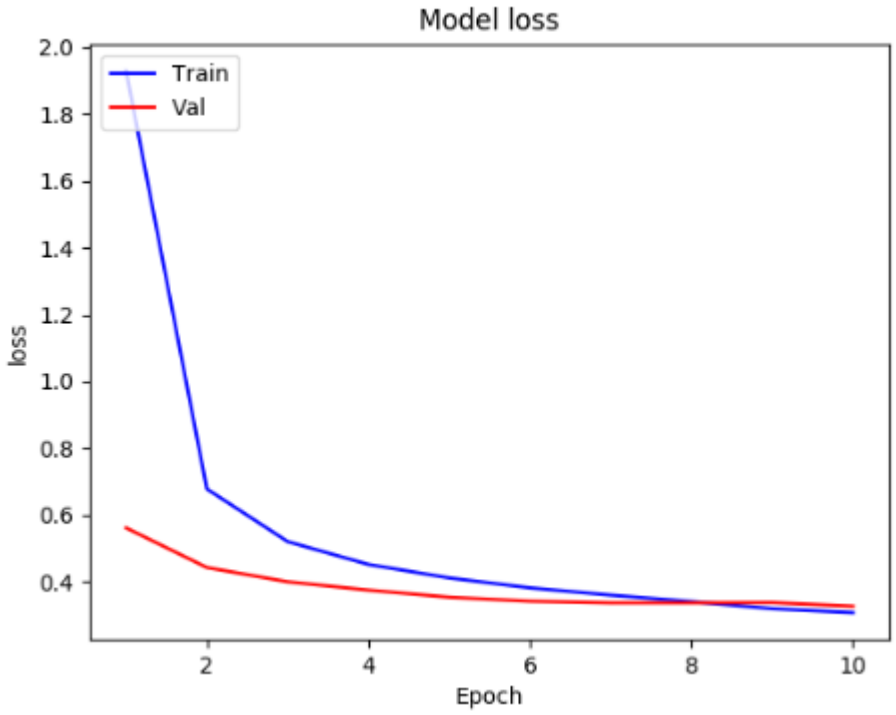
4.3. 10 Epoch ve 512 Batch_size

Modelin 10 epoch ve 512 batch_size değerine göre başarı ve kayıp oran grafikleri Şekil 4.3 ve Şekil 4.4'de verilmiştir.

Şekil 4.3'e de modelin train ve validation için ayrılan veriler üzerindeki başarı oranı gösterilmiştir. Modelin epoch değeri artıkça başarısının arttığı gözlemlenmiştir. Şekil 4.4'de modelin kayıp oranı sunulmuştur. Grafikte epochs değeri arttıkça kayıp oranında bir azalma görülmüştür.



Şekil 4.3. Modelin 10 epochs ve 512 batch_size değerlerine göre yapılan eğitimin başarı grafiği



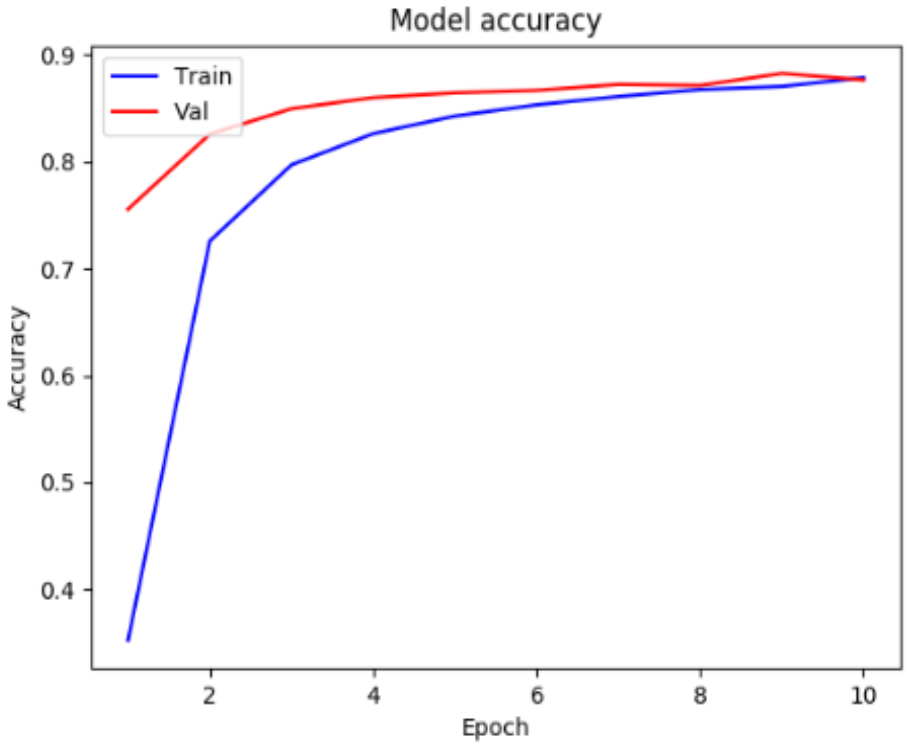
Şekil 4.4. Modelin 10 epochs ve 512 batch_size değerlerine göre yapılan eğitimin kayıp grafiği

4.4. 10 Epoch ve 1024 Batch_size

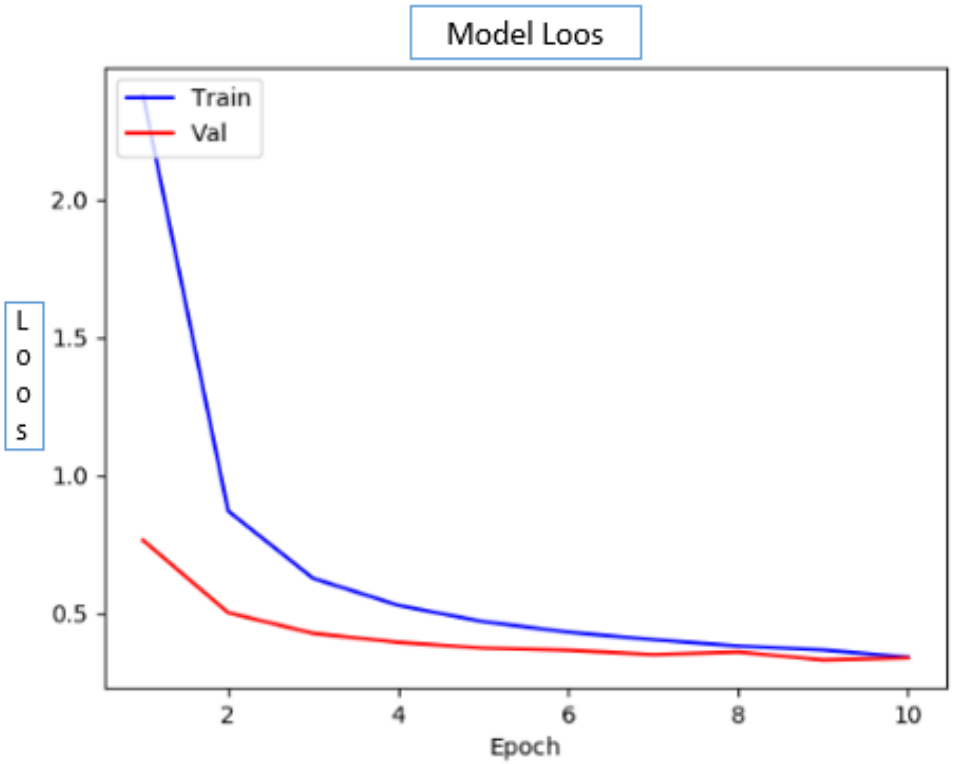
Modelin 10 epoch ve 1024 batch_size değerine göre başarı ve kayıp oran grafikleri Şekil 4.5 ve Şekil 4.6'da verilmiştir.

Şekil 4.5'de modelin train ve validation için ayrılan veriler üzerindeki başarı oranı görülmektedir. Modelin epoch değeri artıkça başarısının arttığı gözlemlenmiştir. Şekil 4.6 modelin kayıp oranını göstermektedir. Grafiğe bakıldığında epochs değeri arttıkça kayıp oranında bir azalma görülmüştür.

Grafiklere bakıldığında train için ayrılan veriler ile validation için ayrılan verilerin eğitimdeki başarı oranı ve kayıp oranında belirli bir zamandan sonra eşitlenme olmuştur. Bu da eğitimin daha sonraki epochs değerinde hep aynı başarı oranı ve kayıp oranı göstereceği anlamında yorumlanmıştır.



Şekil 4.5. Modelin 10 epochs ve 1024 batch_size değerlerine göre yapılan eğitimin başarı grafiği



Şekil 4.6. Modelin 10 epochs ve 1024 batch_size değerlerine göre yapılan eğitimin kayıp grafiği

Modelin eğitimi dört kere tekrarlanmıştır. Bu eğitimler aşağıdaki değerlere göre yapılmıştır.

- Birinci eğitim: 5 epochs ve 512 batch_size,
- İkinci eğitim: 5 epochs ve 1024 batch_size,
- Üçüncü eğitim: 10 epochs ve 512 batch_size ve
- Dörtücü eğitim: 10 epochs ve 1024 batch_size değerleri ile yapılmıştır.

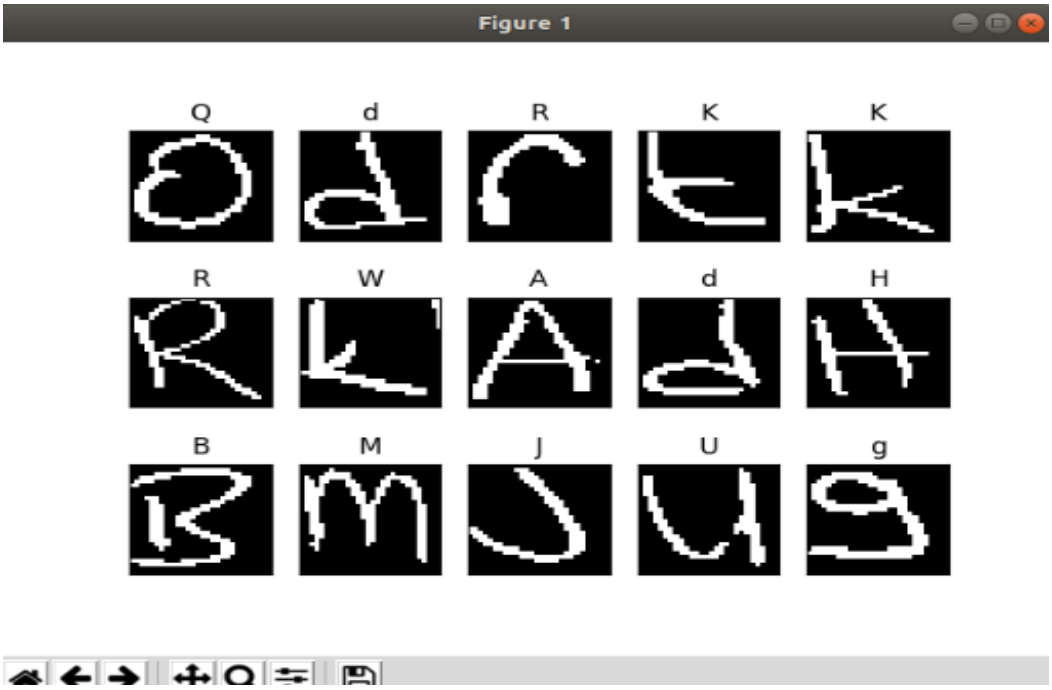
Bu dört eğitim sonucunda en iyi başarı oranı %88.72 oranı ile 10 epochs ve 512 batch_size ile yapılan eğitim olmuştur. Modelin dört eğitimden sonra başarı ve kayıp oranlarının değerleri Çizelge 4.1' de verilmiştir.

***Çizelge* 4.1.** Modelin dört farklı eğitimine göre sonuçları

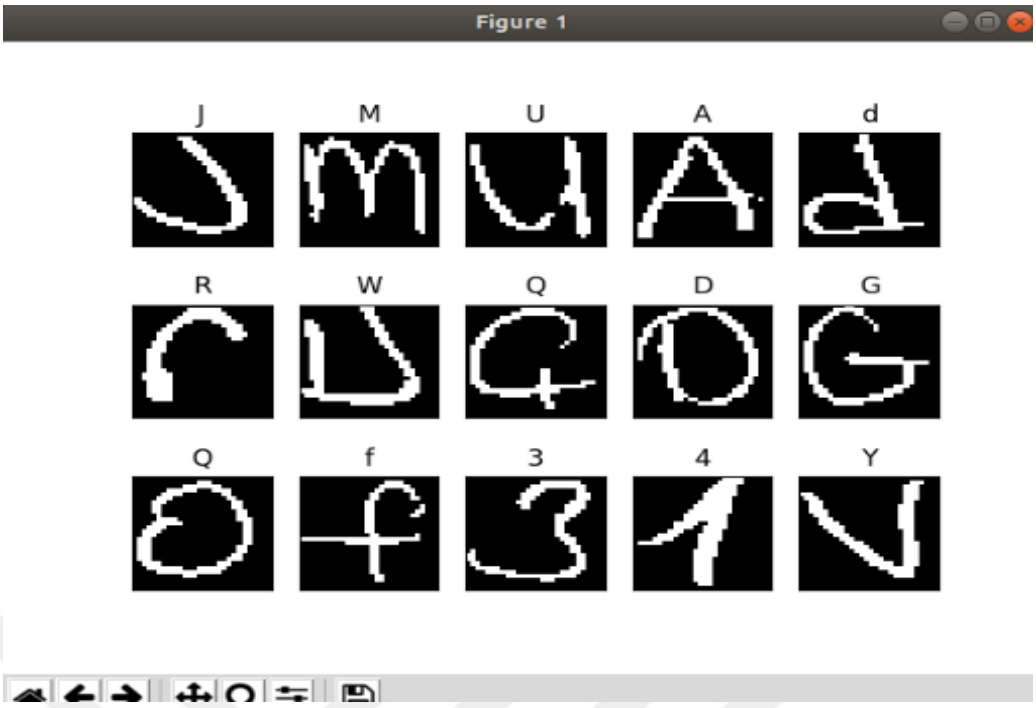
Epochs/ batch_size	**Acc**	**Loss**	**Val_acc**	**Val_loss**	**Test_acc**	**Test_loss**
5/512	%84.64	%37	%85.30	%35	%86	%36
5/2014	%84.54	%46	%86.29	%38.40	%86.48	%35.81
10/512	%88.72	%30.78	%87.98	%32.68	%87.50	%33.42
10/1024	%87.81	%34	%87.61	%34	%87	%35

4.5. Karakter Tanıma

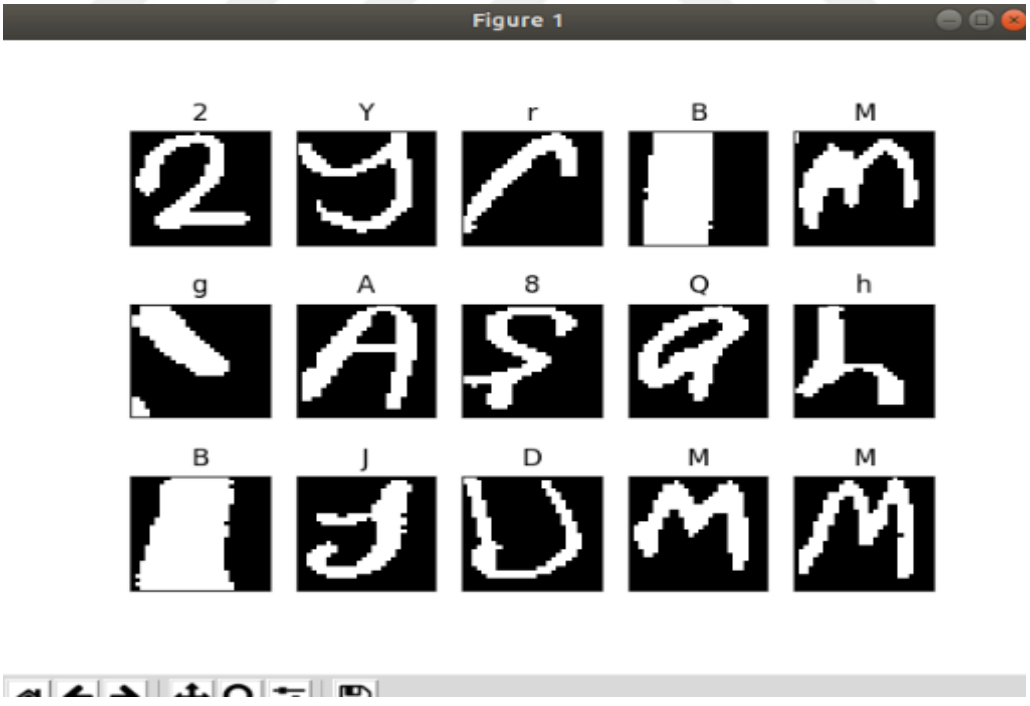
Bu çalışmada bağımsız olarak oluşturulan veri seti, eğitilmiş 4 modelde de ayrı ayrı test edilmiştir. Test işlemi, test veri setindeki veriler kullanılarak yapılmıştır. Bu veri setindeki veriler test edilmeden önce bir ön işleme tabi tutularak daha başarılı bir tanıma işlemi yapılmıştır. Şekil 4.7, Şekil 4.8 ve Şekil 4.9'da test veri setinden elde edilmiş karakterlerin tanınması verilmiştir. Bu tanıma işlemi, 10 epoch ve 512 batch_size ile eğitilmiş model ile yapılmıştır.



Şekil 4.7. Eğitilmiş model ile test edilen karakterler



Şekil 4.8. Eğitilmiş model ile test edilen karakterler



Şekil 4.9. Eğitilmiş model ile test edilen karakterler

5. SONUÇLAR VE ÖNERİLER

5.1. Sonuçlar

Çalışmada el yazısı ile yazılmış karakter görüntüleri üzerinde derin öğrenme algoritması kullanılarak, bir karakter tanıma uygulaması gerçekleştirilmiştir. Türkçe sonsuz boyutlu bir kelime hazinesine sahip olduğu için çalışmada karakter tabanlı bir tanıma işlemi yapılmıştır. Çalışmada karakterlerin sadece sınır bölgeleri yerine bir bütün halinde 2-boyutlu olarak bir sınıflandırma işlemi yapılmıştır. Karakterlerin tanınması için derin öğrenme mimarisi olan Konvolüsyonel Sinir Ağı algoritması kullanılmıştır.

Çalışma iki aşamada gerçekleştirilmiştir. Birinci aşama, karakterlerin sınıflandırılması için bir modelin oluşturulup eğitiminin yapılmasıdır. İkinci aşama, eğitilen modelin el yazıları üzerinde tanıma işlemini gerçekleştirmesidir. Bu iki aşamada gerçekleştirilen işlemler bölüm 3'de ayrıntı olarak verilmiştir.

Yapılan çalışmada model, Konvolüsyonel Sinir Ağı mimarisine göre oluşturulmuş ve eğitimi için EMNIST veri setinin Balanced kısmı kullanılarak yapılmıştır.

Konvolüsyon Sinir Ağı mimarisine göre oluşturulan model, EMNIST veri seti üzerinde 4 farklı değere göre eğitimi yapılmıştır. Modelin öğrenme sonuçları Çizelge 5.1 verilmiştir.

Çizelge 5.1. EMNIST veri seti üzerinde modelin başarı oranları

Epochs/ batch_size	**Acc**	**Loss**	**Val_acc**	**Val_loss**
5/512	%84.64	%37	%85.30	%35
5/2014	%84.54	%46	%86.29	%38.40
10/512	%88.72	%30.78	%87.98	%32.68
10/1024	%87.81	%34	%87.61	%34

Model farklı epoch ve batch_size değerlerine göre eğitilerek karakter sınıflandırma işlemi için en başarılı model oluşturulmaya çalışılmıştır. Çizelge 5.1 incelendiğinde aynı batch_size değerlerine sahip farklı epoch değerlerine göre yapılan eğitimde epoch değerinin artması modelin başarısının da artmasını sağlamıştır. Fakat aynı epoch değerlerine sahip farklı batch_size değerlerine göre yapılan eğitimde batch_size değerinin artması modelin başarı oranının düşmesine neden olmuştur.

Çizelgeye bakıldığında en başarılı eğitim % 88.72 oranla 10 epoch ve 512 batch_size değerlerine göre yapılmıştır.

Modelin eğitiminden sonra kâğıt üzerinde el yazısı ile yazılmış belgeler üzerinde test işlemi yapılmıştır. Bu aşamada kâğıt üzerindeki el yazıları bölüm 3'de anlatılan ön işlemeden geçerek karakterlerine ayrılmıştır. Ayrılan karakterler üzerinde, modelin en başarılı eğitimine göre bir tanıma işlemi yapılmıştır.

5.2. Öneriler

Bu çalışmada Konvolüsyon Sinir Ağı algoritması kullanılarak karakter tabanlı el yazısı tanıma işlemi yapılmıştır. Kullanılan veri seti üzende farklı derin öğrenme algoritmaları ve parametreleri kullanılarak daha başarılı bir tanıma işlemi yapılabilir.

Sonra ki çalışmalarda aynı veri seti üzerinde başka şekilde derin öğrenme mimarileri kullanılarak daha farklı bir tanıma işlemi yapılacaktır. Ve sınıflandırmanın doğruluk oranını artırtmak için 2 veya 3 harften oluşan kombinasyonları içeren veri tabanları kullanılarak harflerde doğrulama işlemi yapılıp, karakterleri birleştirilerek belgenin bir bütün halinde bilgisayar ortamına aktarılması işlemi yapılacaktır.

6. KAYNAKLAR

Aizenberg, I. N., Aizenberg, N. N., & Vandewalle, J. (2000). Multiple-Valued threshold logic and multi-valued neurons *Multi-Valued and Universal Binary Neurons* (pp. 25-80): Springer.

Akhtyamova, L., Ignatov, A., & Cardiff, J. (2017). *A Large-scale CNN ensemble for medication safety analysis.* Paper presented at the International Conference on Applications of Natural Language to Information Systems.

Arica, N., & Yarman-Vural, F. T. (2001). An overview of character recognition focused on off-line handwriting. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 31*(2), 216-233.

Baykal, B., Aktaş, T. Ö., & Yildiz, O. (2017). *MakİNe Öğrenmesİ yÖntemlerİ İle tomatİk ÇevrİmdiŞi İmza tanima ve doğrulama Sistemİ.* Paper presented at the 2017 International Artificial Intelligence and Data Processing Symposium (IDAP).

BEKTAŞ, B., BABUR, S., TURHAL, U., & KÖSE, E. MAKİNE ÖĞRENMESİ YARDIMIYLA OPTİK KARAKTER TANIMA SİSTEMİ OPTICAL CHARACTER RECOGNITION SYSTEM VIA MACHINE LEARNING.

Bilgin Taşdemir, E. F. (2018). *A large vocabulary online handwriting recognition system for Turkish.*

Can, F., & Yilmaz, A. (2019). *Hybrid handwriting character recognition with transfer deep learning.* Paper presented at the 2019 27th Signal Processing and Communications Applications Conference (SIU).

Cho, M., Ha, J., Park, C., & Park, S. (2020). Combinatorial feature embedding based on CNN and LSTM for biomedical named entity recognition. *Journal of Biomedical Informatics, 103*, 103381.

Ciresan, D. C., Meier, U., Gambardella, L. M., & Schmidhuber, J. (2011). *Convolutional neural network committees for handwritten character classification.* Paper presented at the 2011 International Conference on Document Analysis and Recognition.

Coates, A., Ng, A., & Lee, H. (2011). *An analysis of single-layer networks in unsupervised feature learning.* Paper presented at the Proceedings of the fourteenth international conference on artificial intelligence and statistics.

Cohen, G., Afshar, S., Tapson, J., & Van Schaik, A. (2017). *EMNIST: Extending MNIST to handwritten letters.* Paper presented at the 2017 International Joint Conference on Neural Networks (IJCNN).

Deng, L., & Yu, D. (2014). Deep learning: methods and applications. *Foundations and trends in signal processing, 7*(3–4), 197-387.

Engel-Yeger, B., Nagauker-Yanuv, L., & Rosenblum, S. (2009). Handwriting performance, self-reports, and perceived self-efficacy among children with dysgraphia. *American Journal of Occupational Therapy, 63*(2), 182-192.

Ergin, T. (2018,October). Convolutional Neural Network (ConvNet yada CNN) nedir, nasıl çalışır? .

Fanany, M. I. (2017). *Handwriting recognition on form document using convolutional neural network and support vector machines (CNN-SVM).* Paper presented at the 2017 5th international conference on information and communication technology (ICoIC7).

Fujisaki, T., Chefalas, T., Kim, J., Tappert, C. C., & Wolf, C. G. (1991). On-line run-on character recognizer: Design and performance. *International Journal of Pattern Recognition and Artificial Intelligence, 5*(01n02), 123-137.

Fukushima, K., Miyake, S., & Ito, T. (1983). Neocognitron: A neural network model for a mechanism of visual pattern recognition. *IEEE transactions on systems, man, and cybernetics*(5), 826-834.

Hébert, J.-F., Parizeau, M., & Ghazzali, N. (1999). *Learning to segment cursive words using isolated characters.* Paper presented at the Proc. of the Vision Interface Conference.

Hennig, A., Sherkat, N., & Whitrow, R. (1996). *Zone estimation for multiple lines of handwriting using approximating spline functions.* Paper presented at the Proceedings of the Fifth International Workshop on Frontiers in Handwriting Recognition IWFHR-5.

Hennig, A., Sherkat, N., & Whitrow, R. J. (1997). *Recognising letters in on-line handwriting using hierarchical fuzzy inference.* Paper presented at the Document Analysis and Recognition, 1997., Proceedings of the Fourth International Conference on.

Hinton, G. E. (2007). Learning multiple layers of representation. *Trends in cognitive sciences, 11*(10), 428-434.

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *science, 313*(5786), 504-507.

Hinton, G. E., & Salakhutdinov, R. R. (2009). Replicated softmax: an undirected topic model. *Advances in neural information processing systems, 22*, 1607-1614.

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation, 9*(8), 1735-1780.

Ivakhnenko, A. G. e., & Lapa, V. G. (1966). Cybernetic predicting devices: PURDUE UNIV LAFAYETTE IND SCHOOL OF ELECTRICAL ENGINEERING.

Kırlı, Ö. (2011). El Yazısı Görüntülerinden Kişi Tanıma.

Larochelle, H., & Bengio, Y. (2008). *Classification using discriminative restricted Boltzmann machines.* Paper presented at the Proceedings of the 25th international conference on Machine learning.

Le Cun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). *Handwritten digit recognition with a back-propagation network.* Paper presented at the Proceedings of the 2nd International Conference on Neural Information Processing Systems.

Le Cun, Y., Jackel, L. D., Boser, B., Denker, J. S., Graf, H. P., Guyon, I., . . . Hubbard, W. (1989). Handwritten digit recognition: Applications of neural network chips and automatic learning. *IEEE Communications Magazine, 27*(11), 41-46.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural computation, 1*(4), 541-551.

Liou, C.-Y., Cheng, W.-C., Liou, J.-W., & Liou, D.-R. (2014). Autoencoder for words. *Neurocomputing, 139*, 84-96.

Liu, K., Suen, C. Y., Cheriet, M., Said, J. N., Nadal, C., & Tang, Y. Y. (1997). Automatic extraction of baselines and data from check images. *International Journal of Pattern Recognition and Artificial Intelligence, 11*(04), 675-697.

Mahapatra, D., Choudhury, C., & Karsh, R. K. (2020). *Handwritten Character Recognition Using KNN and SVM Based Classifier over Feature Vector from Autoencoder*. Paper presented at the International Conference on Machine Learning, Image Processing, Network Security and Data Sciences.

Momeni, A., Thibault, M., & Gevaert, O. (2018). *Dropout-enabled ensemble learning for multi-scale biomedical data.* Paper presented at the International MICCAI Brainlesion Workshop.

Mushtaq, Z., Su, S.-F., & Tran, Q.-V. (2020). Spectral images based environmental sound classification using CNN with meaningful data augmentation. *Applied Acoustics, 172*, 107581.

NIST. (2017,April). The EMNIST Dataset.

Pervouchine, V., & Leedham, G. (2007). Extraction and analysis of forensic document examiner features used for writer identification. *Pattern Recognition, 40*(3), 1004-1013.

Plamondon, R., & Srihari, S. N. (2000). Online and off-line handwriting recognition: a comprehensive survey. *IEEE Transactions on pattern analysis and machine intelligence, 22*(1), 63-84.

Roy, P. P., Bhunia, A. K., Das, A., Dhar, P., & Pal, U. (2017). Keyword spotting in doctor's handwriting on medical prescriptions. *Expert Systems with Applications, 76*, 113-128.

Saha, P., & Jaiswal, A. (2020). Handwriting Recognition Using Active Contour *Artificial Intelligence and Evolutionary Computations in Engineering Systems* (pp. 505-514): Springer.

Salakhutdinov, R., Mnih, A., & Hinton, G. (2007). *Restricted Boltzmann machines for collaborative filtering.* Paper presented at the Proceedings of the 24th international conference on Machine learning.

Salouhou, A. (2019). *El Yazısı Karakter Tanıma ve Resim Sınıflandırmada Derin Öğrenme Yaklaşımları.* Fatih Sultan Mehmet Vakıf Üniversitesi, Lisansüstü Eğitim Enstitüsü.

Seni, G., Srihari, R. K., & Nasrabadi, N. (1996). Large vocabulary recognition of on-line handwritten cursive words. *IEEE Transactions on pattern analysis and machine intelligence, 18*(7), 757-762.

Srihari, S. N. (1992). High-performance reading machines. *Proceedings of the IEEE, 80*(7), 1120-1132.

Srihari, S. N., Cha, S.-H., Arora, H., & Lee, S. (2001). Handwriting identification: Research to study validity of individuality of handwriting and develop computer-assisted procedures for comparing handwriting. *Technical Report CEDAR-TR-01-1*.

Steinherz, T., Rivlin, E., & Intrator, N. (1999). Offline cursive script word recognition–a survey. *International Journal on Document Analysis and Recognition, 2*(2-3), 90-110.

Su, Y., Zhang, K., Wang, J., & Madani, K. (2019). Environment sound classification using a two-stream CNN based on decision-level fusion. *Sensors, 19*(7), 1733.

Sun, B., Yang, L., Zhang, W., Dong, P., Young, C., Dong, J., & Lin, M. (2019). *Demonstration of Applications in Computer Vision and NLP on Ultra Power-Efficient CNN Domain Specific Accelerator with 9.3 TOPS/Watt.* Paper presented at the 2019 IEEE International Conference on Multimedia & Expo Workshops (ICMEW).

Şeker, A., Diri, B., & Balık, H. H. (2017). Derin öğrenme yöntemleri ve uygulamaları hakkında bir inceleme. *Gazi Mühendislik Bilimleri Dergisi (GMBD), 3*(3), 47-64.

Şekerci, M. (2007). *Birleşik ve eğik Türkçe el yazısı tanıma sistemi.*

Yılmaz, B. (2014). ÖĞRENME GÜÇLÜĞÜ ÇEKEN ÇOCUKLAR İÇİN EL YAZISI TANIMA İLE ÖĞRENMEYİ KOLAYLAŞTIRICI BİR MOBİL ÖĞRENME UYGULAMASI TASARIMI.

Zouhal, L. M., & Denoeux, T. (1998). An evidence-theoretic k-NN rule with parameter optimization. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 28*(2), 263-271.