MAKİNE ÖĞRENME YÖNTEMLERİ İLE EL YAZISI KARAKTERLERİNİN TANINMASI

Özge TUNCER

Yüksek Lisans Tezi
Yönetim Bilişim Sistemleri Ana Bilim Dalı
Prof. Dr. Uğur YAVUZ
2022

T.C.
ATATÜRK ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ
YÖNETİM BİLİŞİM SİSTEMLERİ ANA BİLİM DALI

Özge TUNCER

MAKİNE ÖĞRENME YÖNTEMLERİ İLE EL YAZISI KARAKTERLERİNİN TANINMASI

YÜKSEK LİSANS TEZİ

TEZ YÖNETİCİSİ
Prof. Dr. Uğur YAVUZ

ERZURUM - 2022

SOSYAL BİLİMLER ENSTİTÜSÜ
Graduate School of Social Sciences

TEZ BEYAN FORMU

SOSYAL BİLİMLER ENSTİTÜSÜ MÜDÜRLÜĞÜNE

BİLDİRİM

Atatürk Üniversitesi Lisansüstü Eğitim ve Öğretim Uygulama Esaslarının ilgili maddelerine göre hazırlamış olduğum "MAKİNE ÖĞRENME YÖNTEMLERİ İLE EL YAZISI KARAKTERLERİNİN TANINMASI " adlı tezin/raporun tamamen kendi çalışmam olduğunu ve her alıntıya kaynak gösterdiğimi taahhüt eder, tezimin/raporumun kâğıt ve elektronik kopyalarının aşağıda belirttiğim koşullarda saklanmasına izin verdiğimi onaylarım.

Gereğini bilgilerinize arz ederim *.

☐ Tezimin/Raporumun tamamı her yerden erişime açılabilir.

☐ Tezimin/Raporumun makale için **altı ay,** patent için **iki yıl** süreyle erişiminin ertelenmesini istiyorum.

07.11.2022

Özge TUNCER

Aslı Islak İmzalıdır

---

* **LİSANSÜSTÜ TEZLERİN ELEKTRONİK ORTAMDA TOPLANMASI, DÜZENLENMESİ VE ERİŞİME AÇILMASINA İLİŞKİN YÖNERGE**

.....

ÜÇÜNCÜ BÖLÜM

Çeşitli ve Son Hükümler

Lisansüstü tezlerin erişime açılmasının ertelenmesi MADDE 6– (1) Lisansüstü tezle ilgili **patent başvurusu yapılması veya patent alma sürecinin devam etmesi durumunda**, tez danışmanının önerisi ve enstitü anabilim dalının uygun görüşü üzerine enstitü veya fakülte yönetim kurulu **iki yıl süre ile tezin erişime açılmasının ertelenmesine karar verebilir.**

(2) Yeni teknik, materyal ve metotların kullanıldığı, henüz **makaleye dönüşmemiş veya patent gibi yöntemlerle korunmamış** ve internetten paylaşılması durumunda 3. şahıslara veya kurumlara haksız kazanç imkanı oluşturabilecek bilgi ve bulguları içeren tezler hakkında tez danışmanının önerisi ve enstitü anabilim dalının uygun görüşü üzerine enstitü veya fakülte yönetim kurulunun gerekçeli kararı ile **altı ayı aşmamak üzere tezin erişime açılması engellenebilir**.

Gizlilik dereceli tezler MADDE 7– (1) Ulusal çıkarları veya güvenliği ilgilendiren, emniyet, istihbarat, savunma ve güvenlik, sağlık vb. konulara ilişkin lisansüstü tezlerle ilgili gizlilik kararı, tezin yapıldığı kurum tarafından verilir. Kurum ve kuruluşlarla yapılan işbirliği protokolü çerçevesinde hazırlanan lisansüstü tezlere ilişkin gizlilik kararı ise, ilgili kurum ve kuruluşun önerisi ile enstitü veya fakültenin uygun görüşü üzerine üniversite yönetim kurulu tarafından verilir. Gizlilik kararı verilen tezler Yükseköğretim Kuruluna bildirilir.

(2) Gizlilik kararı verilen tezler gizlilik süresince enstitü veya fakülte tarafından gizlilik kuralları çerçevesinde muhafaza edilir, gizlilik kararının kaldırılması halinde Tez Otomasyon Sistemine yüklenir.

SOSYAL BİLİMLER ENSTİTÜSÜ
Graduate School of Social Sciences

TEZ KABUL TUTANAĞI

SOSYAL BİLİMLER ENSTİTÜSÜ MÜDÜRLÜĞÜNE

Prof. Dr. Uğur YAVUZ danışmanlığında, Özge TUNCER tarafından hazırlanan bu çalışma 07/11/2022 tarihinde aşağıda isimleri yazılı jüri tarafından Yönetim Bilişim Sistemleri Anabilim Dalı'nda Yüksek Lisans Tezi olarak kabul edilmiştir.

Başkan : Prof. Dr. Uğur YAVUZ	İmza:	Aslı Islak İmzalıdır
Jüri Üyesi : Dr. Öğr. Üyesi M. Fatih ALAEDDİNOĞLU	İmza:	Aslı Islak İmzalıdır
Jüri Üyesi : Dr. Öğr. Üyesi İsmail AKGÜL	İmza:	Aslı Islak İmzalıdır

Prof. Dr. Sait UYLAŞ
Enstitü Müdürü

İÇİNDEKİLER

ÖZET

YÜKSEK LİSANS TEZİ

MAKİNE ÖĞRENME YÖNTEMLERİ İLE EL YAZISI KARAKTERLERİNİN TANINMASI

Özge TUNCER

Tez Danışmanı: Prof. Dr. Uğur YAVUZ

2022, 52 Sayfa

Jüri: Prof. Dr. Uğur YAVUZ
Dr. Öğr. Üyesi M. Fatih ALAEDDİNOĞLU
Dr. Öğr. Üyesi İsmail AKGÜL

Günümüzde hem kâğıda kalemle hem de tablette elektronik kalemle yazılan karakterlerin bilgisayarda tanınması ve sınıflandırılması en çok kullanılan uygulamalar arasında yer almaktadır. Ayrıca teknolojinin gelişmesi sayesinde bilgisayar donanımı ve yazılımı, görüntü tanıma ve görüntü işleme, el yazısı karakterlerin incelenmesine olan ilgiyi artırmaktadır. Bu çalışmada Türkçe el yazısı için bir tanıma sistemi tasarlanmıştır. Tasarlanan bu sistemde dokümanın taranması, karakterlerin ayrıştırılması, tanınması ve doğrulanması aşamaları kullanılmıştır. Tanıma sistemi büyük harf ve küçük harf olmak üzere iki tipteki karakteri ayrı ayrı tanıyacak şekilde tasarlanmıştır. Bu çalışma sırasında 71 kişiden 29 küçük harf ve 29 büyük harf olmak üzere toplam 4118 el yazısı karakter örneği toplanmış ve NIST veri seti ile birleştirilmiştir. Bu çalışmada önerilen yöntemin uygulanması sonucunda test setinde “0.016042793” hata oranı bulunmuş ve literatürdeki çalışmalarla karşılaştırıldığında elde edilen başarı görülmüştür. Hem benzer model-farklı veri seti hem de farklı model-benzer veri seti kullanılan çalışmalara göre bir eğitim ve test başarısı elde edilmiştir.

Anahtar Kelimeler: Makine Öğrenmesi, Görüntü Tanıma, El Yazısı Tanıma

ABSTRACT

MASTER THESIS

HANDWRITING WITH MACHINE LEARNING METHODS KNOWING YOUR CHARACTERS

Özge TUNCER

Advisor: Prof. Dr. Uğur YAVUZ

2022, 52 Pages

Jury: Prof. Dr. Uğur YAVUZ
Assist. Prof. Dr. M. Fatih ALAEDDİNOĞLU
Assist. Prof. Dr. İsmail AKGÜL

Today, recognition and classification of characters written both with a pen on paper and with an electronic pen on a tablet are among the most used applications. In addition, thanks to the development of technology, computer hardware and software, image recognition and image processing increase the interest in the study of handwritten characters. In this study, a recognition system was designed for Turkish handwriting. In this designed system, the stage of scanning the document, parsing the characters, recognizing and verifying are used. The recognition system is designed to recognize two types of characters separately, uppercase and lowercase letters. During this study, a total of 4118 handwritten character samples, 29 lowercase and 29 uppercase, were collected from 71 people and combined with the NIST dataset. As a result of the application of the proposed method in this study, the error rate of "0.016042793" was found in the test set and the success achieved is seen when compared with the studies in the literature. A training and test success was obtained according to studies using both similar model-different dataset and different model-similar dataset.

Keywords: Machine Learning, Image Recognition, Handwriting Recognition

KISALTMALAR DİZİNİ

ADAM	: Adaptive Moment Estimation
API	: Application Programming Interface
AQR	: Applied Quantitative Research
CNN	: Convolutional Neural Networks
CPU	: Central Process Unit
DNN	: Deep Neural Networks
GMM	: Gaussion Mixture Model
GPU	: Graphics Processing Unit
ICDAR	: International Conference on Document Analysis and Recognition
INRIA	: Institute for Research in Computer Science and Automation
LSTM	: Long Shor Term Memory
MLP	: Multi Layer Perception
MNIST	: Modifed National Institute of Standards and Techonology
NaN	: Not a Number
NIST	: National Institute of Standards and Techonology
OCR	: Optical Character Recognition
ReLU	: Rectified Linear Unit
RNN	: Recurrent Neural Networks
USBS	: Uyarlamalı Sinir Bulanık Sınıflayıcı
YSA	: Yapay Sinir Ağları

ŞEKİLLER DİZİNİ

TEŞEKKÜR

Yüksek lisans eğitimim boyunca değerli bilgi ve deneyimlerinden yararlandığım, her konuda bilgi ve desteğini almaktan çekinmediğim, araştırmanın planlanmasından yazılmasına kadar olan aşamalarda yardımlarını esirgemeyen ve beni yönlendiren değerli danışman hocam Prof. Dr. Uğur YAVUZ' a teşekkürlerimi sunarım.

Yüksek lisans tezimdeki verilerin hazırlanması ve düzenlenmesinde değerli bilgi ve deneyimlerinden yararlandığım ve her konuda bilgi ve desteğini almaktan çekinmediğim değerli hocam Dr. Öğr. Üyesi Muhammed Fatih ALAEDDİNOĞLU' na desteği ve sabrı için teşekkürlerimi ve saygılarımı sunarım.

Beni hayatım boyunca sevgi ve saygının kıymetini bilecek şekilde yetiştiren, tüm eğitim hayatım boyunca maddi ve manevi desteğini hissettiğim değerli ailem; babam ve annem Kadir – Fatma TUNCER'e, ablam ve erkek kardeşime sonsuz sevgilerimi ve teşekkürlerimi sunarım.

Tezin yazım sürecinde manevi desteklerinin yanı sıra fikir ve destekleriyle her zaman yanımda olan herkese teşekkürlerimi sunarım.

Erzurum – 2022 **Özge TUNCER**

GİRİŞ

Yazı, insanların hayatı boyunca düşüncelerini ifade etmeye, yaptıkları işleri kayıt etmeye ve dönüştürmeye yarayan araçtır (Yıldız,2013).

Kayıt edilen bazı yazılı belgelerin, bilgisayarların kullanabileceği formatlara dönüştürülmesi ihtiyacı ise günden güne artmaktadır. Bu durumda el yazısı ile yazılan birçok belgeyi okuyabilen ve tanıyabilen yazılımların geliştirilmesi talebi ortaya çıkmıştır.

Hızla büyüyen bir sektörde, teknolojinin gerisinde kalan belgelerin yeni sistemlere aktarılması ihtiyacı nedeniyle karakter tanıma teknolojilerinin önemi artmaktadır. El yazısı karakter tanıma teknolojileri birçok iş alanlarında hızla benimsenmeye başlamıştır. Bu alanların bazıları, bankalara gönderilen çeklerin otomatik olarak tanınarak gerekli hesap işlemlerinin elektronik ortamda gerçekleşmesi, trafik denetimi ve geçiş sistemlerinde plaka okunmasında karakter tanıma teknolojileri ile kullanılmaktadır.

El yazısı karakter tanıma da sıkça makine öğrenmesi kullanılır. Makine öğrenmesi, en basit haliyle, bir insan tarafından açıkça programlanma zorunluluğu olmadan kendi kendine "öğrenebilen" ve herhangi bir bilgisayar programını ifade eden bir çalışma alanıdır.

Son yıllarda, makine öğrenmesine bağlı olarak "Derin Öğrenme" alanı ortaya çıkmıştır. Derin öğrenme, birden çok katmana sahip olan yapay sinir ağı ve örnekler aracılığıyla öğrenir. Bu alan ile bir bilgisayar modeli, resimlerden, metinlerden veya seslerden bir sınıflandırma veya tanıma probleminin nasıl gerçekleştirileceğini öğrenir. Derin öğrenme modelleri ile en düşük kayıp (loss) ve en ileri doğruluk (accuracy) değerleri öğrenilebilmektedir (Salouhou 2019).

El yazısı karakter tanıma işlemi sırasında bazı zorluklar da yaşanabilmektedir. Kişinin kullandığı kâğıt ya da kalem, yazı yazarken hızına bağlı olarak harfleri birbirine bağlı yazması ya da harfleri çok değişik biçimlerde ve büyüklükte yazması, harflerin yön farklılıkları gibi durumlar el yazısı karakter tanıma işlemini etkilemekte ve büyük sorunlar oluşturmaktadır. Tarayıcı ile sayısallaştırılan bir belge çok fazla gürültü içerebilmektedir (Ahmet,2007).

El yazısı tanıma sistemleri bazı dillerde oldukça iyi çalışmaktadır. Fakat Türkçe el yazısı karakterleri hakkında yeterli yayın bulunmamaktadır.

Bu tezin amacı; karakter ayrıştırma ve tanıma alanında yapılmış olan çalışmalar incelenerek Türkçe el yazısı karakterlerini Keras kütüphanesini kullanarak bir tanıma sistemi gerçekleştirmektir. Bu çalışmanın diğer bir amacı, çok yönlü büyük veri kümeleri oluşturularak modelin eğitimi ve testini uygulamaktır. Bunun için tezde NIST veri seti, CNN algoritmasını ve makine öğrenmesi yöntemlerinden Pandas, Keras, Numpy, Tensorflow ve Scikit-learn kütüphaneleri kullanılmıştır.

Dört bölümden oluşan bu tezde giriş bölümünde El Yazısı Karakter Tanıma işleminin önemi, el yazısı karakter tanıma da kullanılan Makine Öğrenmesi ve bunun sonucunda gelişen Derin Öğrenme anlatılmıştır. Görüntü İşleme, Görüntü Tanıma ve El Yazısı Karakter Tanıma ile Optik Karakter Tanıma arasındaki bağlantı anlatılmıştır. Etkileşimsiz Karakter Tanıma yönteminden ve buna bağlı olan Tanıma Yönteminden bahsedilmiştir. İlaveten bu çalışmanın amacından bahsedilmiştir.

Birinci bölümde ise El Yazısı Karakter Tanıma üzerine yapılan birçok çalışma incelenmiştir.

İkinci bölümde yöntem ve bulgulardan bahsedilmiştir. Çalışmada kullanılan makine öğrenmesi kütüphanelerinden Python, Tensorflow, Pandas, Numpy, Keras ve Scikit-Learn hakkında detaylı bilgi verilmiştir. Derin öğrenme algoritmaları ve bileşenleri olan kayıp fonksiyonu (loss function), lineer fonksiyon ve optimizasyon algoritmaları (optimization algorithms) hakkında bilgi verilmiştir. Ayrıca NIST veri seti hakkında bilgi verilmiştir.

Sonuç bölümünde ise el yazısı karakter tanıma sisteminin diğer çalışmalarla kıyaslaması yapılmış ve öneri de bulunulmuştur.

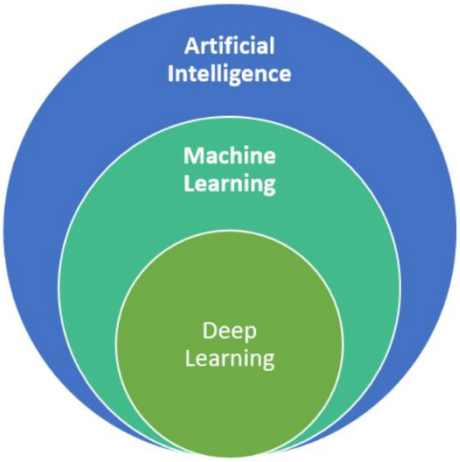
I. YAPAY ZEKA

Yapay Zekâ (Artificial Intelligence) kavramı ilk olarak 1956 yılında Dortmund Konferansında Stanford Üniversitesinde Profesör olan John McCarthy tarafından dile getirilmiştir. Günümüze kadar çok hızlı bir şekilde değişimler gösterdi ve yaşamımızın birçok alanında yoğun bir biçimde kullanılmaktadır (Coşkun ve Gülleroğlu, 2021). Yapay zeka ile akıllı tahminlerde bulunma, karmaşık problemleri çözme, değişen koşullara uyum sağlama gibi durumlara doğrudan ya da dolaylı olarak yardım etmektedir (Arslan, 2020).

Yapay Zeka bilimi içerisindeki alt alanlardan bazıları şunlardır(Aras, 2006; Özbilici, 2006):

- Bulanık Mantık (Fuzzy Logic)
- Genetik Algoritmalar (Genetic Algorithms)
- Görüntü İşleme (Image Processing)
- Konuşma İşleme (Speech Processing)
- Makine Öğrenmesi (Maching Learning)
- Örüntü Tanıma (Pattern Recognition)
- Yapay Sinir Ağları (Artificial Neural Network)
- Derin Öğrenme (Deep Learning)

Şekil 1’ de makine öğrenmesi, yapay zekâ ve derin öğrenme arasındaki ilişkiyi göstermektedir.



Şekil 1. Makine Öğrenmesi, Yapay Zekâ Ve Derin Öğrenme Arasındaki Bağlantı

A. MAKİNE ÖĞRENMESİ

Yeni bir terim gibi görünse de tanımı 1959 yılında bilgisayar oyunlarında ve yapay zekâ alanında Uzman Arthur Samuel yapmıştır. Tanıma göre, makine öğrenmesi, "Öğrenmek için programlamadan öğrenme yeteneği bir çalışma alanıdır." (Samuel, 1959). Makine öğrenmesi, en basit haliyle, bir insan tarafından açıkça programlanmak zorunda kalmadan kendi başına "öğrenebilen" ve herhangi bir bilgisayar programını ifade eden bir çalışma alanıdır (Salouhou, 2019).

Makine öğrenmesi, uygulama alanı fark etmeksizin çok büyük miktardaki geçmiş verileri analiz ederek gelecek ile ilgili öngörülerde bulunabilir ve karar vermemize yardımcı olmaktadır (Kutlugün, 2017).

Makine öğrenimi, yapay zekâ alanının bir alt dalıdır. Makine öğrenimi, model tanıma ve sayısal öğrenme çalışmalarından geliştirilen bir alt daldır (Alkan, 2019). Makine öğrenmesi; Arama Motorları, Nesne Tanıma, Doğal Dil İşleme, Bilgisayar Oyunları, Konuşma ve El Yazısı Tanıma gibi birçok alanda kullanılır (Anka, Kutlugün, ve Çakır, 2017) .

Makine öğrenmesinde üç önemli adım vardır. Birinci adım, öğrenme işleminin yapılacağı dokümanın hazır hale getirilmesidir. İkinci adım, öğrenme yöntemlerinin belirlenmesi ve uygulanmasıdır. Son adım ise öğrenme performansının ölçülmesidir (Uzun, 2007).

Makine öğrenmesi metotları aşağıda tanımları ile verilen işlemlerde kullanılmaktadır.

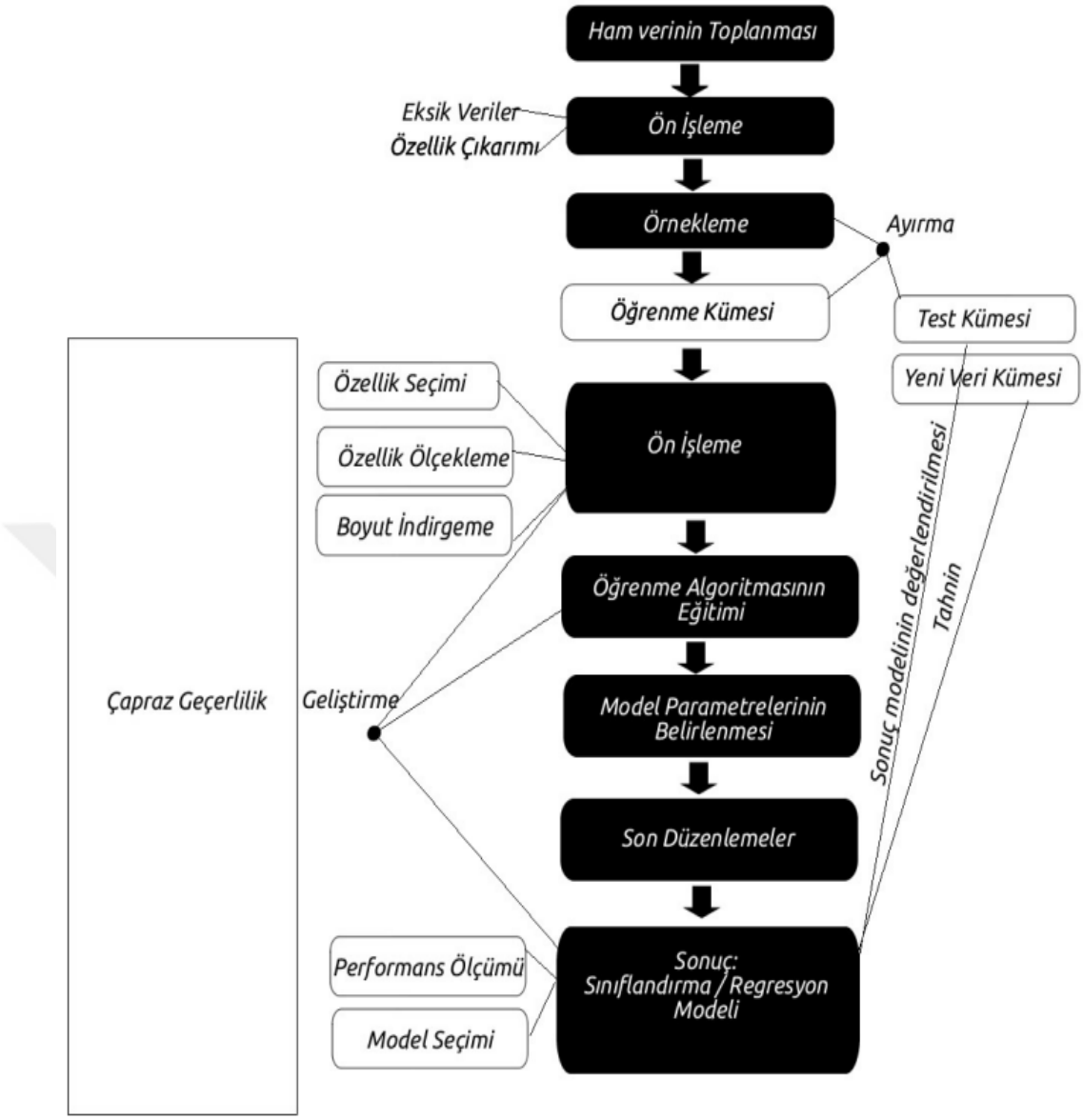
- **Sınıflandırma (Classification):** Belli bir sınıfı olan veriler üzerinden yeni bir sınıfın doğruluğunu tahmin edilebilmesidir.
- **Kümeleme:** Geçmiş bilgilerin bilinmediği verilerde birbirlerine benzerlik koşullarına göre bir kümeye ait olmasıdır.
- **Eğri Uydurma (Regresyon):** Geçmiş verilere karşılık olarak gelen sürekli değerlerin bulunduğu problemlerdir.
- **Özellik Seçimi ve Çıkarımı:** Veriye bazı özelliklerinden faydalanılarak verinin sınıfını ya da kümesini belirleyen özelliklerin belirlenmesidir.
- **İlişki Belirleme:** Oluşturulan sınıf ya da kümeler arasındaki ilişkilerin belirlenmesi işlemidir (Kutlugün, 2017).

1. Makine Öğrenmesinde Yöntemleri

a. Denetimli (Gözetimli) Öğrenme

Denetimli öğrenme, gözlenerek elde edilen etiketlenmemiş (girdi) verilerden faydalanarak bu veriler ve etiketlenmiş (çıktı) verileri kapsayan yeni bir çıkarımın elde edilmesi yöntemidir (Nizam ve Akın, 2014).

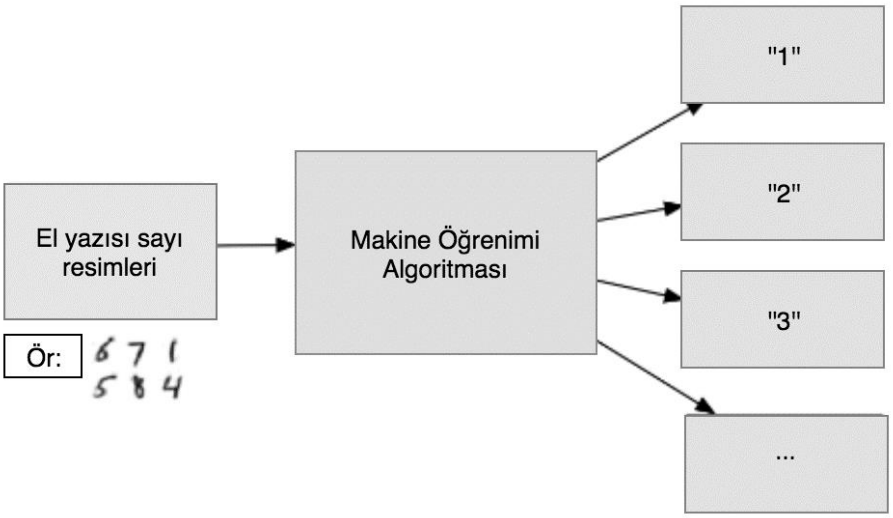
Girdi verileri ve sonuçlanması istenen çıktı verilerinden eğitim verisi elde edilmektedir. Sistem eğitilirken veri setinden her örneğe ait giriş ve çıkışlar verilir. Sistemin doğrulanmasında ise bilmediği bir test verisine eğitim verisinde bulunan uygun çıkışlardan biri atanır (Kotsiantis, Zaharakis, ve Pintelas, 2006). Denetimli öğrenme ve makine öğrenmesi sürecinin iş akış diyagramı Şekil 2'de gösterilmiştir (Saylan ve Gezer, 2017). Sık kullanılan denetimli öğrenme algoritmaları; Sınıflandırma, Regresyon algoritmalarıdır.



Şekil 2. Denetimli Öğrenme İle Makine Öğrenmesi Sürecinin İş Akış Diyagramı

a.1. Sınıflandırma

Geçmiş bilgilerle verinin hangi sınıfa ait olduğu biliniyorsa yeni gelen verilerin hangi sınıfa ait olacağının tespit edilmesidir (Diri, 2014) (Şekil 3). Sınıflandırma da eğitim seti etiketlenerek kullanım amacına göre farklı kategorilere ayrılabilir. Bu ayrılan kategoriler modellenebilir ve konuşma tanıma, DNA dizilimleri ve parmak izi gibi farklı sistem şeklinde tanımlanabilir (Therrien, 1989).



Şekil 3. Sınıflandırma İle İlgili Örnek

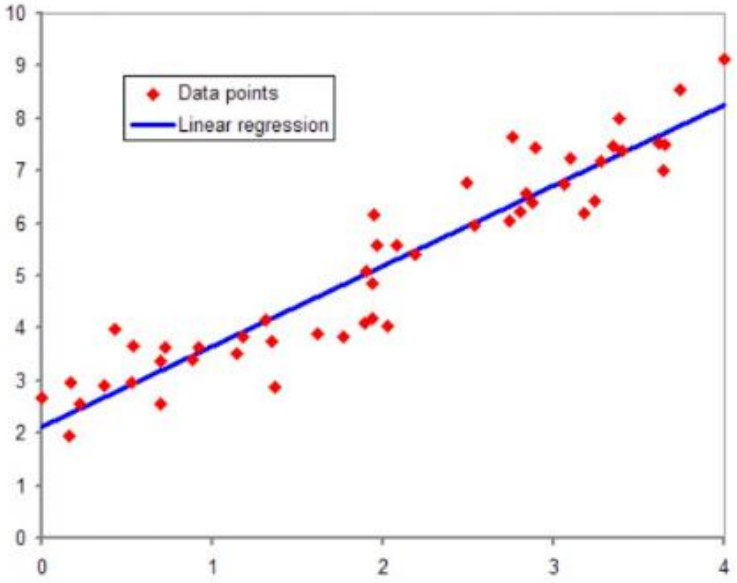
a.2. Regresyon

Girdi verileri X, sürekli bir çıktı verisi olan Y değerini tahmin etmemizi sağlamaktadır. Giriş ile çıkış arasında fonksiyonun eğrisi bulunmaktadır. Regresyon analizi, tek değişkenli veya çok değişkenli kullanılarak da gerçekleştirilebilmektedir (Akça, 2015). Sürekli olarak devam eden araba, ev gibi değerlerin tahmin edilmesinde kullanılır (Günçe, 2021).

a.2.1. Lineer (Doğrusal) Regresyon Algoritması

İki değişken arasındaki lineer ilişkinin çizgi denklemi olarak tanımlanır ve değişkenin değerlerinden biri bilindiğinde diğeri hakkında tahminde bulunmayı sağlar. Veriler arasında doğru bir tahmin yapabilmek için verilerin en iyi satırı oluşturması gerekmektedir. En iyi çizgi oluşturulurken tüm noktalara en yakın bölge tercih edilir. Lineer regresyon örneği Şekil 4'te gösterilmiştir. Lineer Regresyon formülü (1):

$$Y = a + bX \tag{1}$$

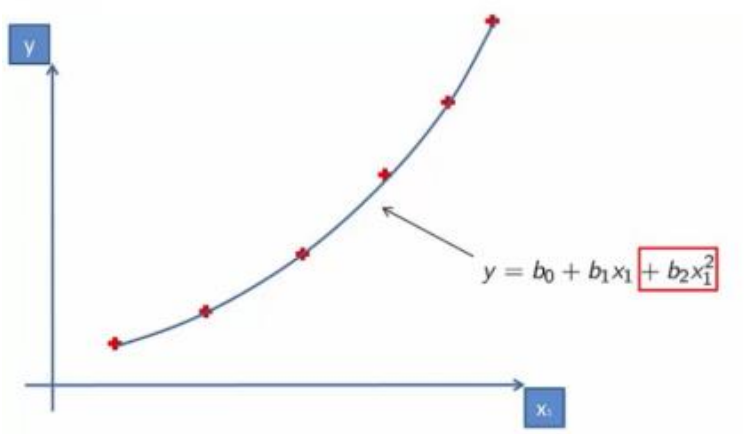


Şekil 4. Lineer Regresyon Örneği

a.2.2. Polinom Regresyon

Birden çok bağımsız değişken arasında bir polinom artışı olduğunda bir bağımlı tercih edilir ve Şekil 5'te örnek sonuç gösterilmiştir. Polinom Regresyon formülü (2):

$$Y = a + bX + cX^2 \tag{2}$$

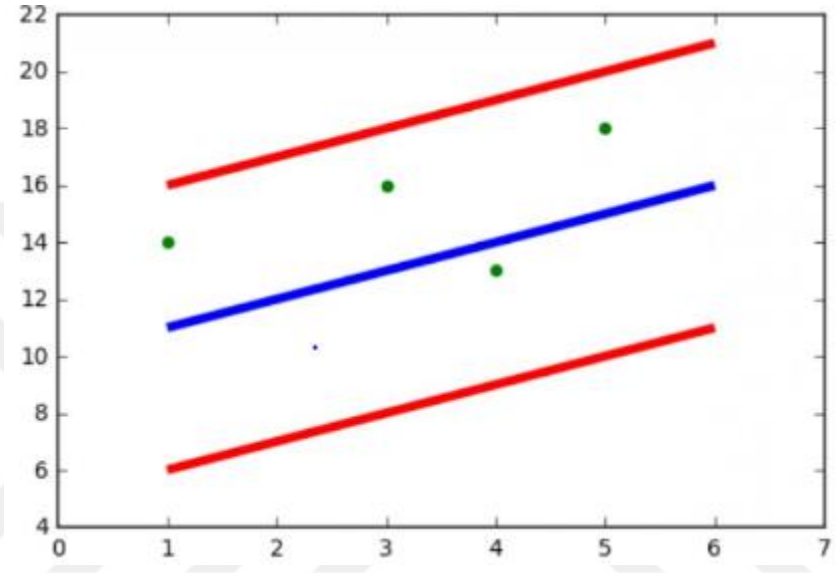


Şekil 5. Polinom Regresyon Örneği

a.2.3. Destek Vektör Regresyonu

Maksimum veri noktası sayısının bu sınır dahilinde olması maksimum marjlı bir hiperdüzlem tanımlanır ve şekil 6'da örnek gösterilmiştir. Destek Vektör Regresyon formülü (3):

$$-a < y - wx + b < a \tag{3}$$



Şekil 6. Destek Vektör Regresyon Örneği

a.2.4. Karar Ağaçları

Karar Ağaçları, regresyon ve sınıflandırma için kullanılır. Veri kümesini, bölme esnasında her seviyede küçük kısımlara bölerek tanımlar. Tahmin sırasında bu aralıktan bir değer istediğinde (eğitim sırasında öğrendiği) bu aralıktaki ortalamayı yanıtlar. Bundan dolayı da karar ağacı regresyonu diğer modeller gibi sürekli değil, kesiklidir. Başka bir deyişle, belirli bir aralıkta istenen tahminler için aynı sonuçları üretir.

Karar ağaçları mevcut durumun entropi (entropy) değerini (rastgelelik derecesini) azaltan seçimler yaparak bilgi kazanımını (information gain) maksimize etmeye uğraşır. Hata fonksiyonu yeniden hesaplanarak en düşük hataya sahip durumu/sorguyu bulur.

Entropy sonucu "0" olursa değerlerin düzenli/homojen olduğunu, eğer "1" olursa değerlerin eşit olduğunu gösterir. P_i veri setindeki i. veri sınıfının tüm sınıf içerisindeki bulunma olasılığını, c verinin miktarını ifade eder. Entropy formülü (4):

$$E = -\sum_{i=1}^{c} p_i \log_2 p_i \quad (4)$$

Bilgi kazanımı, durumlar/sorgular arası entropy farkından oluşur. A niteliği, S veri kümesini ve Sv ise S'nin alt kümesini ifade eder. Bilgi kazanımı formülü (5):

$$Gain(S, A) = Entropy(S) - \sum_{v \epsilon V_a lues(A)} \frac{|S_v|}{|S|} Entropy(S_v) \quad (5)$$

b. Denetimsiz (Gözetimsiz) Öğrenme

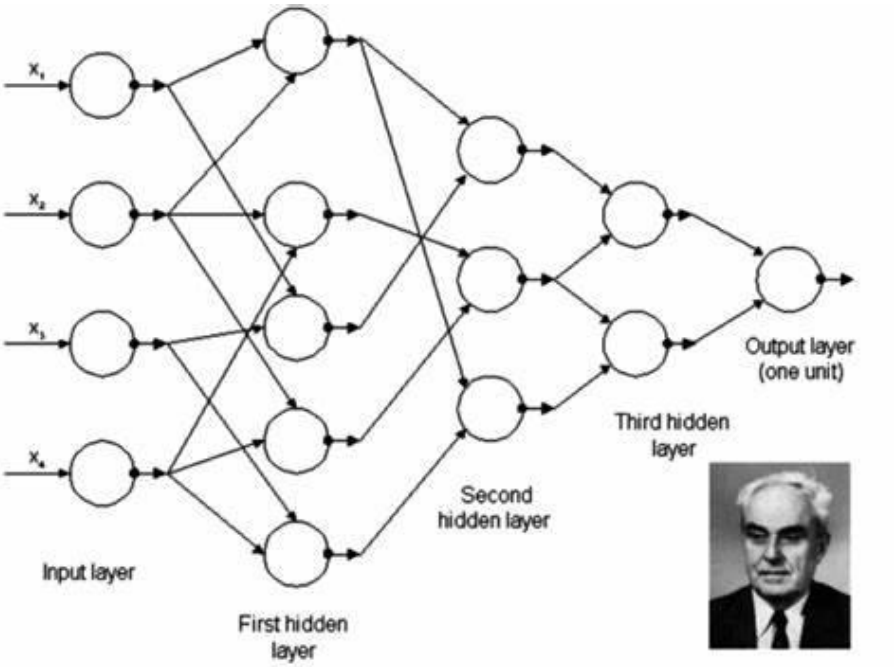
Çıktı verilerini kullanmadan yalnızca girdi verileri üzerinden kendi öğrenmeye çalışır (Hinton ve Sejnowski 1999).Veriler, belirli ortak özelliklere ve kümelere göre sınıflandırılır ve bu verilerden bir anlam çıkarmaya çalışır. Yapılan tahminlerin doğruluğu ile ilgili herhangi bir geri bildirim alınmamaktadır (Anonim, 2020).

b.1. Kümeleme

Geçmiş verilerin etiketlerinin verilmediği / sınıflarının / bilinmediği durumlarda, veriye yakın benzerlik gösteren kümelerin varlığıdır (Diri, 2014).

B. DERİN ÖĞRENME

Geoffrey Hinton, çalışmalarıyla birlikte yayınladığı makalesinde yapay sinir ağlarına yeni bir yaklaşım olan derin öğrenme (Deep Convolution Neural Network) tanıtmıştır (Hinton, Osindero, ve Teh, 2006). 1965 yılında, Ivakhnenko ve Lapa tarafından istatistiksel yöntemler kullanılarak her katman için en iyi özellikler seçilmiş ve bir sonraki katmana aktarılmıştır (Şeker, Diri, ve Balık, 2017) (Şekil 7).



Şekil 7. Ivakhnenko A. Tarafından Eğitildiği Bilinen İlk Derin Ağ Mimarisi

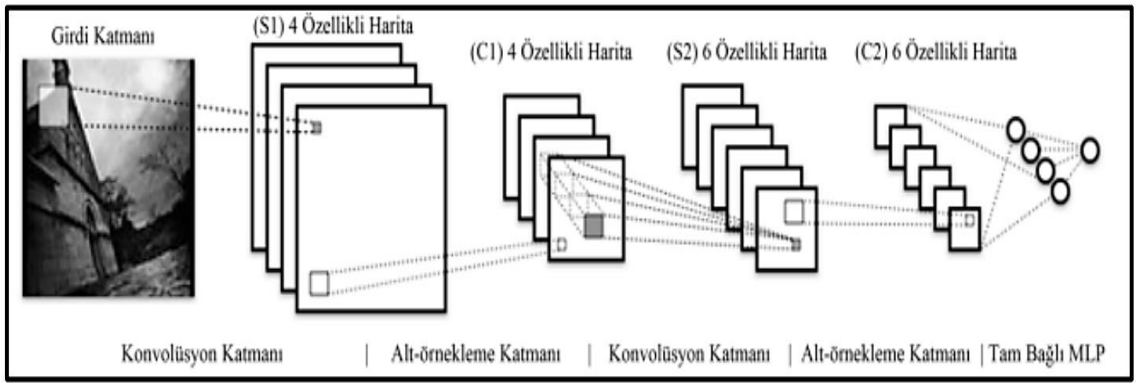
Yapay zekâ uygulamalarında yapay sinir ağları, kümele, makine öğrenmesi gibi birçok alt teknikten faydalanılmaktadır. Derin öğrenme bu tekniklerden biridir ve gelişmiş bir makine öğrenimidir. Gizli katmanlar, içerisinde özellik çıkarma işlemini etkin bir şekilde gerçekleştiren katmanlı bir yapısı vardır. Her katman bir önceki katmanın çıktısını girdi olarak alır (Deng ve Yu, 2014). Derin öğrenme, başlıca verileri tanımlayan kenar kümeleri, bir görüntüdeki piksel yoğunluk vektörü gibi seçili niteliklerden öğrenir.

Derin öğrenmenin çok katmanlı bir yapıya sahip olması, uygulamalardaki veri miktarının artması ve yüksek işlem gücü gerektirmesi nedeniyle 2000'li yılların ikinci yarısından itibaren GPU donanımına sahip bilgisayarların kullanılması için talep edilen yöntemlerden biri olmuştur. GPU donanımının özellikleri geliştikçe konuşma tanıma, görüntü sınıflandırma, video analizi ve doğal dil işleme gibi alanlarda daha başarılı uygulamalar ve sonuçlar elde edilmektedir. Ayrıca GPU paralel işleme yeteneği sayesinde geriye yayılım algoritmalarının hesaplanması hızlı bir şekilde yapılabilmektedir. GPU hızlarındaki artış sayesinde herhangi bir ön-eğitim almadan derin ağlarda eğitim vermek mümkün hale gelmiştir (Şeker vd., 2017). GPU tabanlı yapılan bir uygulamada sonuçların doğruluğunun yanı sıra yüksek öğrenme hızı da büyük bir avantajdır.

1. Derin Öğrenme Algoritmaları

a. Konvülasyonel Sinir Ağları (Convulutional Neural Network – CNN)

CNN, özellikle görüntü analizi için kullanılan bir veya daha fazla evrişim katmanından, bir altörneklemeden (subsampling) ve takiben standart birçok katmanlı sinir ağı gibi bir veya daha fazla tamamıyla bağlı katmandan oluşur (Şeker vd., 2017). CNN algoritması, hayvanların görme sistemlerinden ilham alınmıştır. İlk CNN ağı 1988'de Yann LeCun tarafından tanıtılan ve 1998'lere kadar geliştirilmeye devam eden LeNet adlı mimaridir (Şeker vd., 2017) (Şekil 8).



Şekil 8. Lenet Ağının Mimarisi

b. Tekrarlayan Sinir Ağları (Reccurrent Neural Network – RNN)

Başlangıçta canlı-cansız, insan-hayvan gibi ayrım yapan Basit Tekrarlayan Ağ (Simple Recurrent Network-SRN) Jeff Elman tarafından tasarlanmıştır. RNN ise yönlendirilmiş bir döngü ile birimler arasındaki bağlantıların dinamik zamansal davranışını sağlayan bir ağın dahili durumunu oluşturmak için sıralı bilgileri kullanan bir yapay sinir ağı sınıfıdır. RNN mimarisi, önceki çıktılara dayalı olarak bir dizinin her öğesi için (örneğin bir cümledeki sözcükler) aynı görevi yüklediği için tekrarlayan (recurrent) olarak adlandırılır (Mikolov vd., 2010).

c. Derin Sinir Ağları (Deep Neural Networks – DNN)

Derin Sinir Ağları, insan beyninin bilgi işleme metodundan ilham alınarak nöronlardan oluşan çok katmanlı yapay sinir ağıdır. DNN, bir giriş katmanı ile bir çıktı

katmanı arasında birden fazla doğrusal olmayan gizli katmana sahiptir ve her katmanın ağırlığı delta öğrenme yöntemiyle hesaplanır (Butuner ve Calp, 2022). DNN algoritmalarında, tüm veri kümesi çoğunlukla eğitim ve test veri kümesi olarak adlandırılan iki veri kümesine bölünür. Eğitim veri setinin rolü, sinir ağındaki ağırlıkların tahminlerini hesaplamaktır. Test veri setinin rolü ise, modelden gizlenen verilerden elde edilen ağırlık değerlerinin doğruluğunu test etmektir (Priddy ve Keller, 2005; Ser ve Bati, 2019).

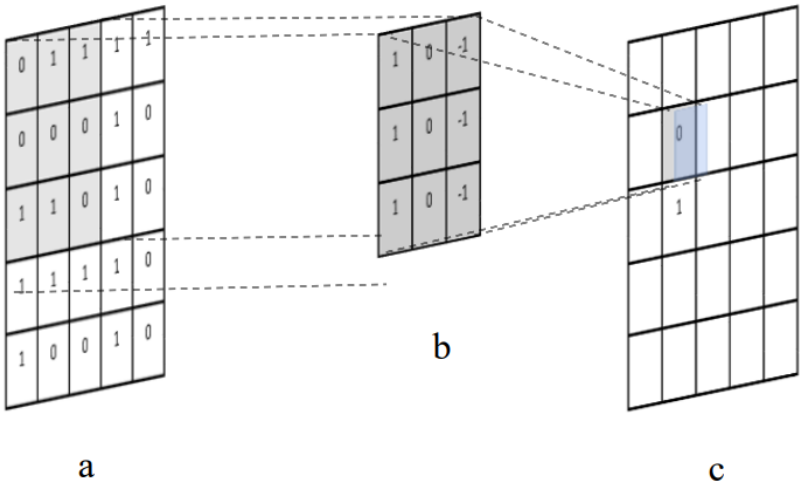
2. Derin Öğrenme Katmanları

a. Giriş (Input) Katmanı

Veriler herhangi bir işlem görmeden bu katmana ham olarak verilir. Veri setinin boyutu ağın hızını, test süresini ve bellek gereksinimini artırır. Yüksek bir girdi görüntü boyutu seçmek hem yüksek bellek gereksinimi hem de eğitim süresi ve test süresini uzatabilir. Ayrıca ağ başarısını da artırabilir. Görüntü boyutunu küçültmek ise daha başarısız sonuçlar verebilir.

b. Konvülasyon (Convolation) Katmanı

Belirtilen küçük boyutlu matrisler görüntü matrisinin tamamında hareket ettirilerek görüntüdeki özellikleri vurgulayan yeni bir görüntü matrisi elde edilir (Liu, Shen, ve Van Den Hengel, 2015) (Şekil 9).

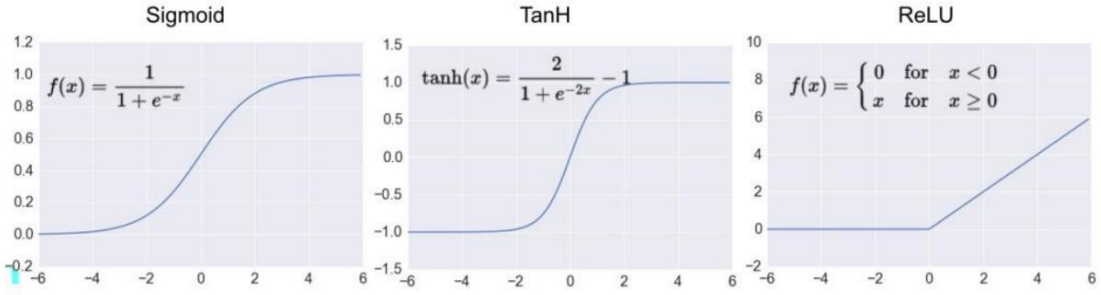


Şekil 9. Konvolüsyon İşlemi. A) Görüntü Matrisi B) Filtre C) Filtre Sonucunda Oluşan Yeni Görüntü Matrisi

c. Aktivasyon (Activation) Katmanı

Elde edilen özellik haritası üzerine bir aktivasyon fonksiyonu uygulanarak, özelliklerin düzenlenmesini sağlar. Derin ağ modellerinde en popüler üç aktivasyon fonksiyonu ve grafikleri şunlardır (Yavuz, 2021) (Şekil 10):

- **Tanh:** (-1) ile (1) arasındaki değerleri analiz eder.
- **Sigmoid:** (0) ile (1) arasındaki değerleri analiz eder.
- **ReLU:** Değer negatifse "0" olur, yoksa aynı kalır.

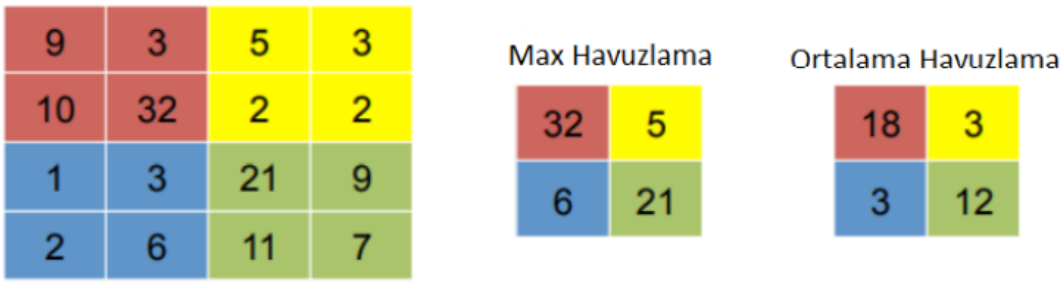


Şekil 10. En Popüler Üç Aktivasyon Fonksiyonunun Grafikleri

d. Havuzlama (Pooling) Katmanı

Matristeki kalan uygun değeri bulmak için veri havuzundan NxN boyutlu bir filtre geçilir. Ortalama (average pooling) ve maksimum değer (max pooling) işlemleri değer

elde etmek için en çok kullanılan işlemlerdir (Doğan ve Türkoğlu, 2019) (Şekil 11). Havuzlama katmanı, öznitelik haritalarının boyutunu küçültür ve elde edilen öznitelik haritasının güvenilirliğini artırır (Altuntaş, 2021).



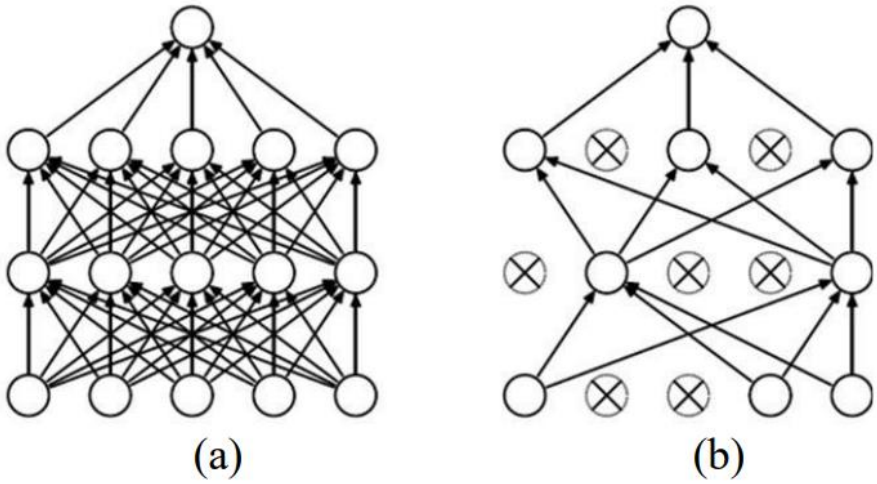
Şekil 11. Max ve Average Pooling İşlemi

e. Tam Bağlı (Full Connected) Katmanı

Bir önceki katmandaki verilere bağlı olarak bu katmanda tek boyutlu bir matris şeklinde bulunur. Nöronlar bu katmanda tamamen bağlantılıdır. Her nöron bir sonraki nöronla bağlıdır. Bu nedenle tam bağlı katman olarak bilinir (Adler, Elad, ve Zibulevsky, 2016).

f. Dropout Katmanı

Çok katmanlı yapay sinir ağlarında, sinir ağı eğitilirken aşırı öğrenme adı verilen ağın ezberlenmesi meydana gelir ve bunu önlemek için ezberleme yapan bazı düğümler ortadan kaldırılır (Srivastava vd., 2014) (Şekil 12).



Şekil 12. (A)Yapay Sinir Ağı (B)Dropout Uygulanmış Sinir Ağı (Çarpı Atılmış Nöronlar Ağdan Çıkarılmıştır)

g. Softmax Katmanı

Sınıflandırma işlemi içerisinde bir önceki katmandan değerleri alarak olasılıksal değer üretimi gerçekleştirir. Bu katmanda, her bir nöronun değerinin [0,1] arasına çekilmesini sağlayarak, nöronların sonuçlarında bir normalizasyon gerçekleştirir ve girdi görüntüsünün hangi sınıfa ait olma olasılığının belirlenmesine izin verir. Sınıflandırma yaparken hangi sınıfa daha yakın olduğuna dair değer üretir (Bishop, 2006).

II. GÖRÜNTÜ İŞLEME

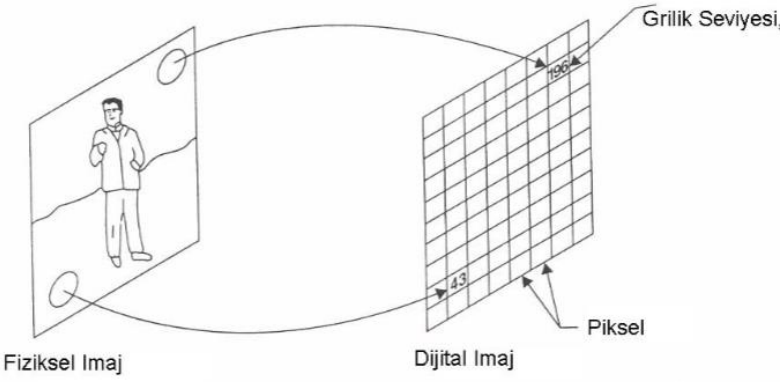
Görüntü işleme, elimizde bulunan bir görüntüden yararlı bilgiler çıkarmak ya da bir görüntüyü elde etmek için farklı algoritmalar kullanarak görüntü üzerinde bazı işlemlerin gerçekleştirilmesi yöntemine denilmektedir (Aydın, 2019).

Analog bir görüntü veya resimde I^1 (x, y) gibi bir fonksiyonla tanımlanmaktadır. Bu fonksiyonda, x ve y görüntünün yatay ve dikey eksendeki koordinatlarına karşılık gelen değişkenlerdir. Sayısal görüntü ise bu analog görüntünün N satır ve M sütun olarak örneklenmesiyle elde edilir. Bir satır ve sütunun kesiştiği her bölgeye piksel denir. Sayısal görüntüye dönüştürülen resimde N x M piksel var (Karakoç, 2012). N ×

[1] I'nın sayısal değeri görüntünün ilgili noktadaki gri seviye değeri veya parlaklık değeridir.

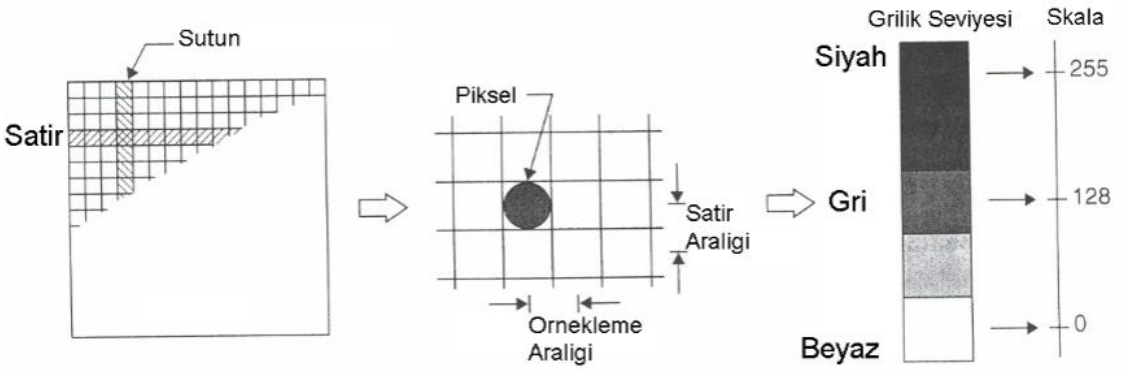
M pikselli iki boyutlu bir sayısal görüntü, N satır ve M sütundan oluşan bir matris olarak düşünülebilir (Yeşilyurt, 2021).

Fiziksel görüntü, piksellere bölümlenmiştir. Dijital görüntü ise piksellere bölümlenmiş görüntünün gri ton renk ölçeği kullanılarak sayısal değere dönüştürülmüştür (Şekil 13).



Şekil 13. Bir Fiziksel Görüntü ve Dijital Karşılığı

Her piksel için görüntünün parlaklığı örneklenir ve sayısal olarak değerlendirilir. İşlemin bu adımında her piksel için o noktadaki karanlığı ya da parlaklığı simgelemektedir. Bu işlem tüm pikseller için gerçekleştirildiğinde, bir pikseldeki gri tonun 0- 255 arası sayısal veri olarak tanımlanarak görüntü, dörtgen sıralı bir şekilde gösterilir (Şekil 14). Her piksel, satır, sütun numarası ya da konumu ve grilik seviyesi tam bir değere sahiptir. Bu sıralı dijital veri bilgisayarda işlenmek için uygundur (Castleman, 1996).



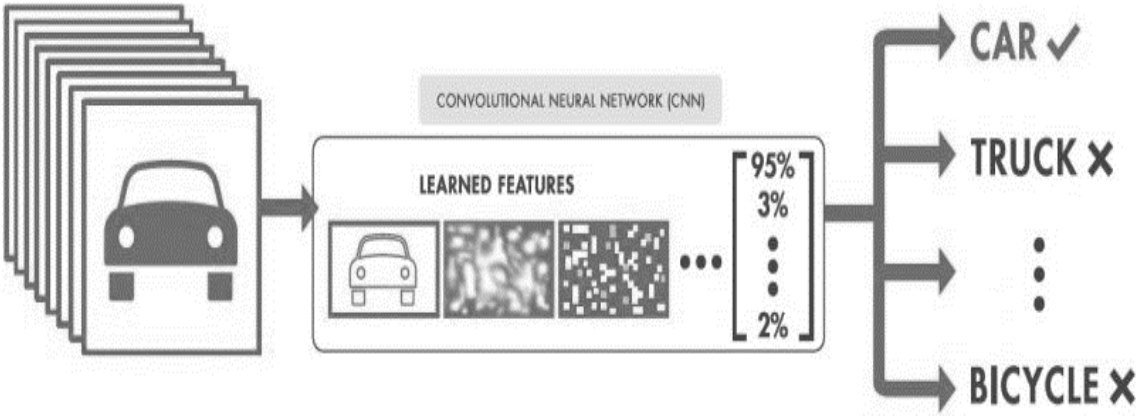
Şekil 14. Bir Görüntünün Sayısallaştırılması

Günümüzde teknolojinin gelişmesi ve yapay zekâ alt dalları kullanılarak pek çok farklı alanda görüntü işleme başarıyla uygulanmaktadır. Görüntü işlemenin kullanıldığı bazı alanlar şunlardır;

- Yüz tanıma ve güvenlik sistemleri
- Robot teknolojileri
- İnsansız hava araçları
- Google harita teknolojisi
- Radar, astronomi, trafik, gazete ve fotoğraf endüstrisi uygulamaları
- Duygu Analizi
- Snapchat, Instagram'da kullanılan gerçek zamanlı filtreler
- Radyoloji alanı (Tomografi, Ultrason vb.)

III. GÖRÜNTÜ TANIMA

Görüntü tanıma, bir görüntüdeki öğeleri etiketlemek ve sınıflandırmak için kullanılan algoritmalar ve yöntemler kümesidir. Görüntü tanıma modelleri, bir girdi görüntüsü almak için eğitilmektedir. Görüntüyü tanımlayan ise önceden sınıflandırılmış etiketleri çıkarmaktadır. Görüntü tanıma modelinin çalışma mantığı Şekil 15' te gösterilmiştir.



Şekil 15 . Görüntü Tanıma Convolutional Neural Network (CNN) Çalışma Mantığı

Bir görüntünün tek tek piksellerini işlemektedir. İşlenen bu verilerden anlamlı bir sonuç alınabilmesi için görüntüden belirli özelliklerin çıkarılması gerekmektedir. Bu işleme, özellik çıkarma adı verilmektedir. Özellik çıkarma, belirli desenlerin belirli

vektörlerle temsil edilmesini sağlamaktadır. Bu vektörlerin sınır aralığını belirlemek için derin öğrenme yöntemleri de kullanılmaktadır. Modeli eğitmek için bir veri seti kullanılmaktadır. Bunun sonunda model belirli nesneleri tahmin etmektedir ve yeni girdi görüntüsünü belirli bir sınıfa etiketlemektedir (Dilmegani, 2020).

IV. El YAZISI KARAKTER TANIMA

El ile yazılan rakam, harf ve sembollerin bilgisayar sistemleri tarafından tanınmasına denir (Dodurgalı, 2010).

Optik Karakter Tanıma (Optical Character Recognition) el yazısı karakter tanıma öncüsüdür. Birçok araştırmacı, OCR ile karakter tanımayı çözülmesi kolay bir problem olarak değerlendirmiştir. Ancak beklenin aksine, karakter tanımada büyük zorluklarla karşılaşmışlardır (Mori, Suen, ve Yamamoto, 1992).

1950'lere kadar harfleri ve sayıları okuyabilen makineler düşünülmüştür (Uğur ve Aydın, 2006). 1951 yılında M. Sheppard tarafından OCR'daki en eski çalışma kabul edilen okuma-yazma robotu GISMO icat edildi ve sadece 23 karakter tanımlayabildi. 1954' te J. Rainbow tarafından sadece büyük harfleri okuyabilen bir makine geliştirildi. 1967'de şirketler IBM birlikte Optik Karakter Tanıma sistemleri üretmeye başladı. Çok farklı karakter setlerinde, yazı türlerinde ve hatalı belgelerde problem devam etmektedir (Verma, Blumenstein, ve Kulkarni, 1998). El yazısı karakter tanıma probleminin çözülememesinin bazı nedenleri şunlardır (Dodurgalı, 2010):

- Görsel karakterde gürültü karışıklıklara sebep olmaktadır.
- Aynı karakterler biçim, yazı stili ve boyut olarak kişiden kişiye ya da aynı kişide zamanla değişmektedir.
- Görsel karakterlerin görünüşlerini tarif edebilecek kural yoktur. Sistem görsel karakterler için örneklerden faydalanarak sezgisel olarak çıkarmalıdır.

El yazısı karakter tanıma probleminin çözümünde kullanılan Etkileşimli (Online) ve Etkileşimsiz (Offline) Yöntemler vardır.

A. ETKİLEŞİMSİZ KARAKTER TANIMA YÖNTEMİ

Kâğıt üzerine yazılan metinler ilk olarak sayısallaştırılır ve bilgisayara aktarılır. Ardından tanıma algoritmasının çalıştıracağı gri seviye veya ikili karakter görüntüsü elde edilir.

Tanıma işlemi sırasında aşağıdan yukarıya şeklinde çalışan Etkileşimsiz Yöntem, benek seviyesi ile başlar anlamlı bir kelime bulmasıyla biter. Bu yaklaşımda hiyerarşik olarak ön işleme (pre-processing), bölütleme (segmentation), öznitelik çıkarma (feature exraction-representation) ve tanıma (recognition) işlemleri şeklinde gruplandırılır (Arica ve Yarman-Vural, 2001).

1. Tanıma (Recognition) Yöntemi

Karakter tanıma sistemlerinde çoğunlukla örüntü tanıma (pattern recognition) yöntemi kullanılır. Sayısallaştırılmış bir karakterin önceden belirlenmiş sınıflardan doğru olan ile eşleştirebilmesidir (Plamondon ve Srihari, 2000). Karakter tanıma da kullanılan bazı yöntemler şunlardır:

- İstatiksel Analiz Yöntemleri (Statistical Techniques)
- Şablon Eşleme Yöntemleri (Template Matching)
- Yapay Sinir Ağları (Artifical Neural Networks)

BİRİNCİ BÖLÜM

İLGİLİ ÇALIŞMALAR

Yapılan literatür araştırmasında, el yazısı karakterlerinin tanımlamak için pek çok farklı yöntem kullanıldığı görülmektedir.

Ayhan vd. (2005), çok katmanlı ileri beslemeli sinir ağı (YSA) mimarisi ve geri yayılım öğrenme yöntemi kullandı. Yapay sinir ağlarının önemli bir uygulaması olan karakter tanıma süreci ele alınmaktadır. A'dan Z'ye Times New Roman ve Arial formatında 29 Türkçe karakteri büyük ve küçük harflerle, el yazısı karakterlerle giriş vektörü olarak sunmuşlar ve bu karakterler üzerinden YSA'nın eğitim sürecini gerçekleştirmişlerdir. Bu karakterler kullanılarak yazılan metin içeren resme karakter algılama ve tanıma uygulandıktan sonra altyazıyı metne dönüştürdüler. Ayrıca C ++ Builder yazılım geliştirme ortamı ile resimlerden karakter tanıma ve tanıma yapan bir yazılım geliştirmiş ve uygulama yapmışlardır.

Perihan (2006), geri yayılım öğrenme algoritması kullanarak el yazısı karakter tanıma sistemi geliştirmiştir. Oluşturduğu geri yayılım, yapay sinir ağının parametre performanslarını grafiksel olarak incelemiş ve ağ performansını artıracak parametre değerlerini bulmuştur. İkinci olarak, Shashank'ın öğrenme algoritmasını kullanarak bir karakter tanıma sistemi tasarladı. Test aşamasında Shashank'ın aday puanı (candidate score), ideal ağırlık modeli puanı (ideal weight model) ve tanıma bölümü (recognition quotient) değerleri kullanılmıştır. Geliştirilen sistemde geri yayılım ağının performansı % 83 iken Shashank ağının performansı % 61'de kaldı.

Graves vd. (2012), Çok Boyutlu Yinelemeli Sinir Ağları (Multidimensional Recursive Neural Networks) ile çevrimdışı el yazısı tanımayı sundu. Geliştirilen çerçeve, girdi olarak işlenmemiş pikselleri alır. Alfabeye özel herhangi bir hazırlık gerektirmeyecek şekilde düşünülmüştür ve başka bir dili değiştirmeden kullanılabilir. Sistemleri, yetkili bir çevrimdışı komut dosyası tanıyıcı belirlemek için çok boyutlu Uzun-Kısa Süreli Bellek (LSTM) ile bağlantı kurarak geçici bir sınıflandırma ve hiyerarşik bir yapıyı birleştirdi. Sisteme İngilizce ve Arapça bir veritabanı uygulayarak güzel sonuçlara ulaştı. Sistem, Uluslararası Arap Tanıma Yarışması'nda % 91.4 doğruluk sağladı.

İhsan vd. (2013) el yazısıyla yazılan sayıların sekizinci mertebeden cebirsel denklemlerini elde ederek denklem katsayılarını özellik olarak kullanmıştır. Elde ettikleri katsayıların değişmez olması için sadece ölçekleme ve öteleme için normalize edilmişlerdir. Ek olarak, dil gücü ve Uyarlamalı Sinir Bulanık Sınıflayıcı (USBS) ile özellik seçimi yaptılar. Çalışmada, önerilen yöntemin Bayes ve yapay sinir ağları ile tanınma başarısı, Modified National Institute of Standards and Technology (MNIST) el yazısı sayı veritabanı kullanılarak ölçüldü. Araştırmada elde edilen % 92,87'lik tanınma oranı diğer yöntemlerle karşılaştırılabilir düzeyde değildir. Ancak yöntem geliştirilerek her karakterin bir denklem ile ifade edilebileceği açıktır. Böylelikle karakterleri görüntü formatında saklamak yerine katsayılarla saklayarak daha az bellek kullanımı sağlanabilir.

Musa vd. (2013) yaptıkları çalışmada, program yardımıyla el yazması Assamese dili karakterlerinin x, y koordinat değerlerini kaydetmişlerdir. Bu değerlerin boyutları Temel Bileşen Analizi ile küçültülmüş ve ardından minimum, maksimum, ortalama, medyan, varyans, standart sapma ve aralık değerleri alınarak özellikler elde edilmiştir. Bu özellikler, İleri Beslemeli Geri Yayılma Yapay Sinir Ağları ve Radyal Tabanlı Yapay Sinir Ağları olarak sınıflandırılmıştır. Sınıflandırmadan sonra test sonuçlarını karşılaştırdılar. Assamese dili çevrimiçi el yazısı veritabanı, California Üniversitesi Bilgi ve Bilgisayar Bilimleri Bölümü'ndeki başvurular için kullanıldı. İleri Beslemeli Geri Yayılımlı Yapay Sinir Ağı ile yapılan test sonucunda % 96 oranında, Radyal Tabanlı Yapay Sinir Ağında % 82 başarı oranına ulaşmıştır.

Durjoy vd. (2015), Standart 50 sınıf Bangla temel karakter veritabanını tanımak için bir CNN eğitimi aldıkları Çoklu Komutların El Karakter tanımasına CNN Tabanlı Ortak Yaklaşımı sundu. Ayrıca İngilizce, Telugu, Bangla, Devanagari ve Oriya dillerinin 5 farklı 10 sınıflı dijital tanınması için özellikler çıkarıldı. Bu çalışmanın amacı, farklı bir karakter tanıma sorununda uygulanabilmesi için bir nitelik çıkarma taktiği uygulamaktı. Çalışmaların sonuçları, Bangla dijital için % 98.375, Bangla temel karakterleri için % 95.6, Devanagari dijital için % 98.54, Oriya dijital için % 97.2 ve Telugu dijital için % 96.5 dijital oranlar elde etti.

İshak (2016) çalışmasında, yapay sinir ağları (YSA) ve Optik Karakter Tanıma (OCR) kullanarak modern Türkçeyi Osmanlı Türkçesine çevirmiştir. Bu tez iki süreçten

oluşmaktadır. İlk aşama, herhangi bir resimdeki karakterleri akıllı bir sistemle tanıma ve ayırma işlemidir. İkinci aşama, ayrılmış karakterlerin Osmanlı Türkçesine çevrilmesi sürecidir. YSA eğitimi için kullanılacak resim formatındaki karakterler elde edilmiş ve görüntü işleme teknikleri kullanılarak ölçeklenmiştir. Ardından, özellikleri kaldırıldı ve normalleştirildi. Ağ, normalleştirilmiş verileri ağa sunarak eğitildi. Eğitilen YSA test edildi ve performansı hesaplandı. Eğitimli ağ, sırayla farklı karakterleri tanır. Tanınan kelimelerin Osmanlı Türkçesine çevrilmesi sürecinde Osmanlıca gramer yapısına göre değiştirilmeleri gerekmektedir. Osmanlı Türkçesinin gramer yapısına göre, "a" ünlüsü için "ا" (elif), e ünlüsü için "ه" (o), o, ö, u, ü sesleri için "و" (vav) i, i "ى" (ye) sesleri için "و" (vav) kullanılır. Arapça ve Farsça kelimeler de Osmanlı Türkçesinde yaygın olarak kullanılmaktadır. Bu kelimeler Arapça ve Farsça gramer yapılarına göre tercüme edilmiştir.

Ahmet vd. (2017), evrişimsel sinir ağını (CNN) kullanarak Arapça el yazısı karakterlerinin tanıması üzerine bir çalışma yayınladı. El yazısı Arapça karakterler, eğitim amaçlı 13.440 resim ve test amaçlı 3360 resim içeren 16800 resimden oluşan bir veri setinin bulunduğu bir eğitim ve test veri seti hazırlandı. Çalışmadaki tanıma oranı % 94,9'dur.

Xu Yao vd. (2018) Çince karakterlerin hem tanıması hem de çizilmesi için farklı ve üretken modeller 21 olarak Yinelenen Sinir Ağını (RNN) kullanan bir sistem tanıtmıştır. Çerçeve, sıralı yapıyla başa çıkmak için oluşturulmuştur ve farklı bir alana özgü bilgiye ihtiyaç duyulmaz. Ayırt edici modelinde, çift taraflı RNN yöntemi, tanınma için hem GRU hem de LSTM ile entegre edilmiştir. Çince karakterleri üretirken, kalem yönünü şekillendirmek için Gaussian Harmanlama Model'ini (GMM) kullandılar, bu da başka el yazısı stilleri belirlerken çeşitliliği garanti ettiler. Eğitim ve test için, çevrimiçi el yazısı Çince karakter tanıma tekniğinin ICDAR 2013 Yarışması veritabanını kullandılar. Deneyim sonunda % 95.19 ile % 98.15 arasında değişen farklı veri setleri için ölçüm sonuçları aldılar.

Aoudou (2019) çalışmasında, Deep Neural Network (DNN), Convolutional Neural Network (CNN) ve Recursive Neural Network (RNN) Derin Öğrenme Algoritmaları, performansları artırmak için el yazısı karakter tanıma ve resim sınıflandırma problemleri, gelişmiş iyileştirme işlevi ve optimum hiper parametreler

elde edilecek sonuçlar belirlenerek önce modeller, daha sonra benzer bir çalışmada aynı veri seti ve aynı model dikkate alınarak karşılaştırma yapılmıştır. İlk aşamada modellerin sonuçları karşılaştırıldı. İkinci aşamada, kullanılan algoritmalar, literatürde tavsiye edilen benzer modellerin sonuçları ile karşılaştırılmıştır. Bu araştırmada gerekli teknik ve medyayı oluşturduktan sonra RNN, CNN ve DNN modelleri hazırlanarak hiper parametreleri belirledi. Modellerin karşılaştırılması için veri setlerine göre sütun grafikler çizdi. CNN modeli, el yazısıyla yazılmış karakterleri ve görüntüleri karşılaştırma yapmadan sınıflandırmanın en iyisi olduğunu belirtti.

İKİNCİ BÖLÜM

YÖNTEM VE BULGULAR

Bu bölümde, NIST veri kümesi, doğruluk ve kayıp performans metrikleri; Pandas, Keras, Numpy, Tensorflow ve Scikit-learn kullanılan araçlar ve kütüphaneler; CNN ile oluşturulan model ve hiper-parametre ayarlamaları; benzer çalışmalar ile sonuçlarımızın karşılaştırması ve bulgular sunulmuştur.

2.1. KULLANILAN VERİ KÜMESİ

Çalışma, Anaconda Tümleşik Geliştirme Ortamı (Integrated Development Enviroment – IDE) ve Jupyter Notebook yazılımı kullanılarak geliştirilmiştir. Geliştirme dili olarak Python, derin öğrenme kütüphaneleri olarak Tensorflow ve Keras kullanılmıştır.

Veri seti için Atatürk Üniversitesi Yönetim Bilişim Sistemleri bölümü öğrencilerden "Türkçe El Yazısı Örnek Formu" ile 71 kişiden el yazısı örneği toplanıldı (Şekil 17). Toplanan örnekler, tarayıcı yardımıyla görüntü dosyalarına dönüştürüldü. 142 adet görüntü dosyasına sırasıyla şu adımlar uygulanarak her bir görüntü ayrıştırıldı.

- Türkçe El Yazısı Formunu gri seviyeye çevrildi ve nesne bulmak için 2'li seviyeye dönüştürüldü.
- Görüntü üzerindeki harfleri barındıran karelerdeki nesneler resim üzerinde kırpılarak 28x28 boyutunda görüntülülere dönüştürüldü.
- Arka fon beyaz ve harfler siyah olacak şekilde [0-255] piksel değerlere dönüştürülerek "txt" dosyasına yazdırıldı. Bu işlemler tüm görüntü dosyaları için tek tek yapıldı.
- Her görüntü dosyası için oluşan txt dosyaları, tek txt dosyası olacak şekilde birleştirildi ve ardından arka fon siyah ve harfler beyaz olacak şekilde değiştirildi.
- Daha sonra her satırda ayrı bir önek için görüntü etiketi 0-57 arasında ve piksel değeri alacak şekilde kayıt yapıldı. Oluşturulan dosyanın yapısı aşağıdaki gibidir:

1.harf değeri,1.piksel değeri,2.piksel değeri, … ,784.piksel değeri

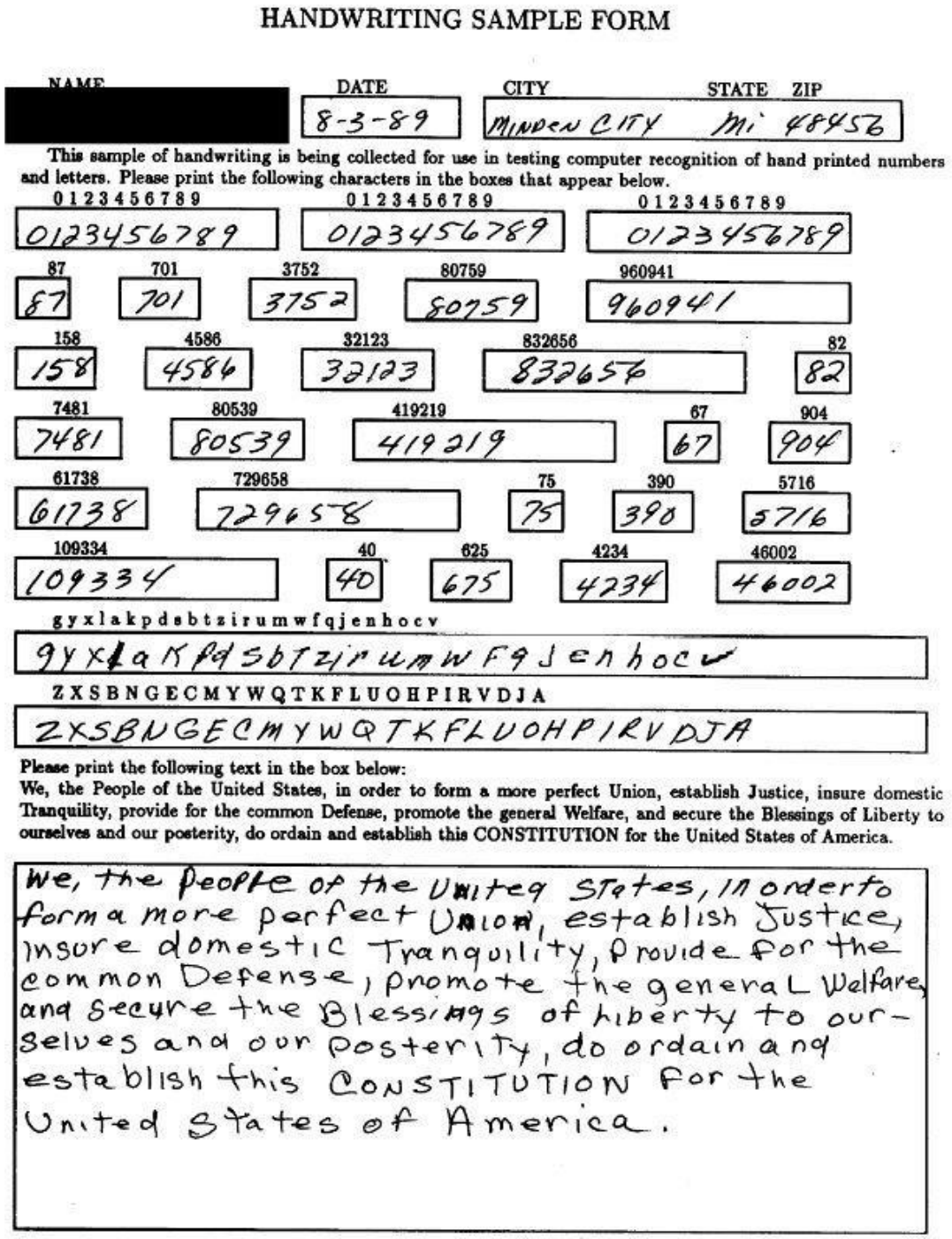
2.harf değeri,1.piksel değeri,2.piksel değeri, … ,784.piksel değeri

Bu şekilde veri kümesinde 4118 satır ve 785 sütundan, “A-Z” ve “a-z” karakterlerinden oluşmaktadır. Veri kümesi çalıştırıldığında başarılı sonuçlar alınamadı. Bu nedenle daha verimli sonuçlar elde etmek için bu veri kümesi NIST veri kümesi ile birleştirilerek 753138 satır ve 785 sütundan oluşan bir veri kümesi elde edilmiştir.

2.1.1. NIST Veri Kümesi

1901 yılında ABD Ticaret Bakanlığına bağlı olarak kurulan Amerikan Ulusal Standartlar ve Teknoloji Enstitüsü (National Institute for Standard and Technology - NIST) ülkenin en eski laboratuvarlarından biridir (Can, 2019).

2016 yılında NIST tarafından NIST Özel Veri Tabanı 19 (NIST Special Database 19) el ile yazılmış doküman ve karakter tanıma çalışmalarından oluşmaktadır (Anonim, 2016). Veri kümesi, “El Yazısı Örnek Formu” ile 3669 kişiden el yazısı ve rakam karakteri toplanarak elde edilmiştir ve bu formun düzenlenmesiyle 814.255 el yazısı ve rakam karakteri ayrıştırılmıştır (Şekil 16). Veri kümesi “0-9”, İngiliz alfabesi “A-Z” ve “a-z” karakterlerinden oluşmaktadır (Can, 2019).



HANDWRITING SAMPLE FORM

NAME [redacted] DATE 8-3-89 CITY MINDEN CITY STATE MI ZIP 48456

This sample of handwriting is being collected for use in testing computer recognition of hand printed numbers and letters. Please print the following characters in the boxes that appear below.

0123456789 0123456789 0123456789

0123456789 0123456789 0123456789

87 701 3752 80759 960941

87 701 3752 80759 960941

158 4586 32123 832656 82

158 4586 32123 832656 82

7481 80539 419219 67 904

7481 80539 419219 67 904

61738 729658 75 390 5716

61738 729658 75 390 5716

109334 40 625 4234 46002

109334 40 675 4234 46002

gyxlakpdsbtzirumwfqjenhocv

gyxlaKpdsbtzirumwFqJenhocv

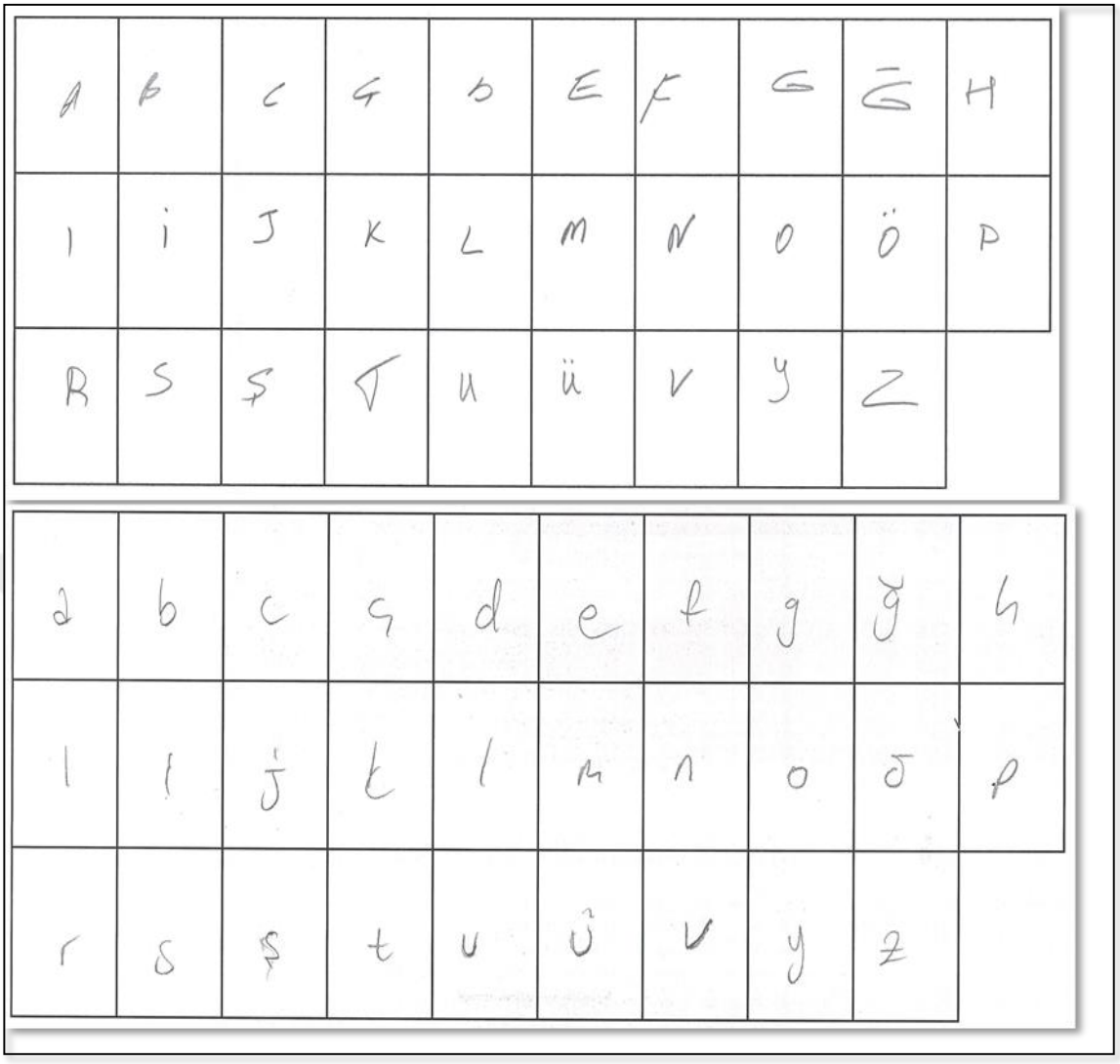
ZXSBNGECMYWQTKFLUOHPIRVDJA

ZXSBNGECMYWQTKFLUOHPIRVDJA

Please print the following text in the box below:
We, the People of the United States, in order to form a more perfect Union, establish Justice, insure domestic Tranquility, provide for the common Defense, promote the general Welfare, and secure the Blessings of Liberty to ourselves and our posterity, do ordain and establish this CONSTITUTION for the United States of America.

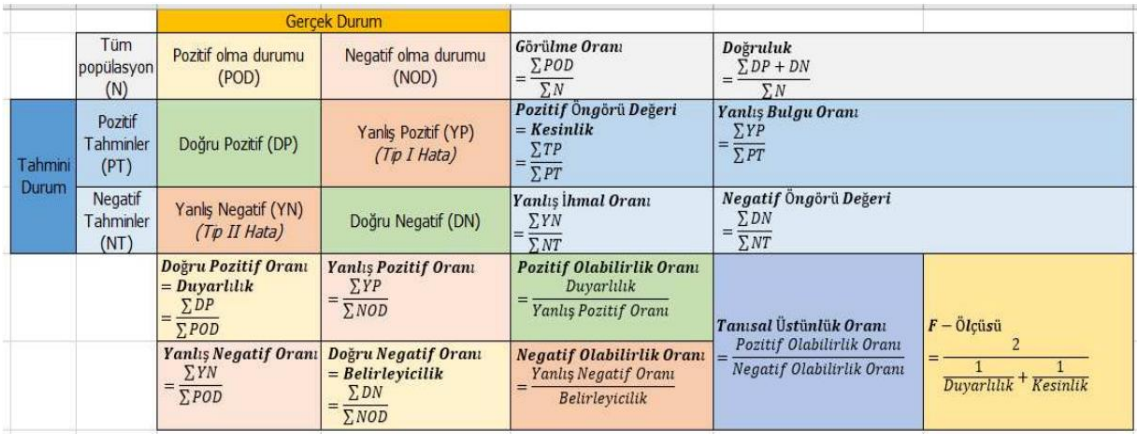
We, the People of the United States, In order to form a more perfect Union, establish Justice, insure domestic Tranquility, Provide for the common Defense, promote the general Welfare, and secure the Blessings of liberty to ourselves and our Posterity, do ordain and establish this CONSTITUTION for the United States of America.

Şekil 16. Nıst El Yazısı Örnek Formu



Şekil 17. Türkçe El Yazısı Örnek Formu

Diğer makine öğrenmesi yöntemlerinin veya aynı makine öğrenmesi yöntemi kullanılarak oluşturulan modellerin ve diğer değişkenlerin performansını karşılaştırmak için çeşitli performans değerlendirme yöntemleri ve ölçüleri geliştirilmiştir. En sık kullanılan performans metrikleri aşağıdaki şekilde gösterilmiştir (Şekil 18).



Şekil 18. Performans Değerlendirmesinde Kullanılan Çeşitli Ölçüler

Doğruluk (accuracy) değeri, algoritma tarafından doğru tahmin edilen örnek sayısının veri setindeki toplam örnek sayısına bölünerek bulunmaktadır. Genellikle doğruluk değeri, kurulan modellerin ve kullanılan algoritmaların performans karşılaştırması için kullanılmaktadır(Kartal ve Özen 2017). "DP" doğru pozitifi, "DN" doğru negatifi, "YP" yanlış pozitifi ve "YN" yanlış negatifi ifade etmektedir (Yılmaz, 2021). Doğruluk formülü (6):

$$Doğruluk = \frac{DP+DN}{DP+YP+YN+DN} \quad (6)$$

2.1.2. Kullanılan Araçlar ve Kütüphaneler

2.1.2.1. Python

1990'ların başında Guido Van Rossum tarafından Amsterdam'da geliştirilmeye başlandı. Python, öğrenmesi çok kolay, yüksek düzeyde basit sözdizimine sahip, nesne yönelimli, modülerliği ve okunabilirliği destekleyen, platformdan bağımsız, yorumlanabilir bir betik dilidir (Özdemir, 2014). Python, kullanıcıların yoğun ilgisiyle karşılaştıktan sonra sırasıyla kendi içinde daha da gelişti ve sırasıyla şu sürümleri yayınladı.

- Python 0.9
- Python 2.0
- Python 3.0
- Python 3.7.7

Python programlama dili yorumlanmış ve etkileşimlidir (Anonim). Python ile, ihtiyacınız olan işin çoğunu az sayıda kod satırı ile gerçekleştirebilirsiniz. Python'da bir program yazarken ihtiyaç duyabileceğiniz birçok veri yapısı ve işlevi hazırdır. Bu sayede diğer dillerde olduğu gibi bir problemi çözmek için en ince ayrıntısına kadar tasarım yapmanıza gerek kalmayacaktır (Python Programlama Dili Hakkında Bilinmesi Gerekenler). Python'un basit bir sözdizimi vardır. Bu sayede başkaları tarafından yazılan program ve programları yazmak daha kolay anlaşılır hale gelmektedir (Soytürk). Python'u kullanmak için gerekli eğitim bilgileri, kurulumu, hangi API'leri içerdiği ve gerekli öğretim bilgileri web sitesinde mevcuttur. Python programlama dilinin kullanım alanı geniştir ve şuralarda kullanılır;

- Görselleştirme uygulamaları
- Akademik alanlarda
- Web uygulamaları
- Oyun programlama
- Ağ programlama
- Yapay zekâ
- Veritabanı
- Güvenlik
- Sayısal alanlarda

Python arayüzündeki kütüphanelerin birçoğu veri bilimi ve makine öğrenimi üzerine elverişlidir. Bu kütüphanelerdeki yüksek kaliteli komutlar, makine öğrenimi kütüphanelerinin gelişmesine yardımcı oldu (Anonim).

Python programlama dilini güçlü kılan bir diğer özelliği ise Windows, Linux, Mac ve Unix gibi farklı platformlarda çalışabilmesidir.

Python dili, avantajları ile Google, Instagram, NASA, Youtube, Netflix, Uber ve IBM gibi firmaların ilgi odağında yer almaktadır.

2.1.2.2. Pandas

Pandas'ın gelişimi 2008 yılında Wes McKinney tarafından AQR Capital Management'ta başladı. 2009' un sonunda açık kaynaklı hale geldi (Anonim). Pandas'da kullanılan diller Python, CPython ve C'dir.

Pandas, "etiketli" veya "ilişkisel" verilerle çalışmayı hem kolay hem de sezgisel hale getirmek için tasarlanmış hızlı, esnek ve etkileyici veri yapıları sağlayan bir Python paketidir.

Pandas, birçok farklı veri türü için çok uygundur (Şahin, 2020):

- Bir SQL tablosu veya Excel elektronik tablosu gibi, heterojen olarak yazılmış sütunlara sahip tablo verileri
- Sıralı ve sırasız (sabit frekanslı olmayabilir) zaman serisi verileri
- Satır ve sütun etiketleri ile rastgele matris verileri (homojen veya heterojen olarak yazılmış)
- Herhangi bir biçimde istatistiksel / gözlemsel veri kümeleri. Verilerin bir Pandas veri yapısına yerleştirilmesi için etiketlenmesine gerek yoktur.

Pandas'da iki başlıca veri yapısı olan DataFrame ve Seriler, sosyal bilimler, finans, istatistik, birçok mühendislik alanındaki özgün kullanım şartlarının çoğunu ele almaktadır. Pandas, NumPy üzerine kurulu bir pakettir. Diğer birçok üçüncü taraf kütüphanesiyle bilgi işlem şartlarına iyi bir şekilde bütünleşik olmayı hedeflemektedir.

Pandas'ın özelliklerinden birkaçı şunlardır (Şahin, 2020) (Anonim):

- Kayıp verilerin kolay işlenmesini sağlar (Nan diye ifade edilir).
- Boyut değişebilirliği: DataFrame ve daha yüksek biçimli nesnelerden oluşan sütunlar silinebilir veya eklenebilir.
- Otomatik ve açık veri hizalama: Nesneler bir etiket kümesine açıkça hizalanabilir veya kullanıcı etiketleri görmezden gelebilir. Seri, DataFrame benzeri hesaplamalarda verileri sizin için otomatik olarak hizalamasına olanak tanır.
- Verilerin hem toplanması hem de dönüştürülmesi için veri kümeleri üzerinde bölme-uygulama-birleştirme işlemleri gerçekleştirmek için güçlü ve esnek bir "group by" özelliğine sahiptir.

- Diğer Python veya NumPy veri yapılarından farklı ve düzensiz indekslenmiş verilerin DataFrame nesnelerine dönüştürülmesini kolaylaştırır.
- Veri setlerinin rahatlıkla tekrardan şekillendirilmesi
- Hiyerarşik olarak eksenlerin etiketlenmesi (işaret/tik başına çok fazla etiket olabilir.)
- Zaman serisine mahsus fonksiyonellik: frekans değiştirme, tarih aralığı belirleme ve kayan pencere istatistikleri, tarih geciktirme ve kaydırma

2.1.2.3. NumPy

NumPy, Numeriacal Python (Sayısal Python)'un kısaltmasıdır. Jim Hugunin tarafından geliştirilen Numaray ve Numeric kütüphanelerin özellikleri kullanılarak Travis Oliphant tarafından 2005 yılında oluşturulmuştur (Anonim).

NumPy vektör, matris veya dizi hesaplamaları için özel bir kitaplık olmasına rağmen, büyük veri kümeleri üzerinde çalışmayı kolaylaştırır. Genellikle matematiksel işlemler yapılır. NumPy kullanılarak simülasyonlar ve istatistiksel işlemler gerçekleştirilebilir. Python'un dahili dizilerini kullanarak, daha verimli ve mümkün olandan daha az kodla yürütülür.

NumPy kütüphanesinin tabanındaki ndarray nesnesi, homojen veri türlerinin n boyutlu dizilerini içerir ve performans için derlenmiş kodda birçok işlem gerçekleştirilir. NumPy dizileri ile standart Python dizileri arasında bazı temel farklılıklar vardır ve şunlardır:

- NumPy dizileri oluşturulurken sabit bir boyuta sahip iken, standart Python listeleri ise dinamik olarak büyüyebilen dizilerdir. Bir ndarray nesnesinin boyutu değiştirildiğinde yeni bir dizi oluşturulur ve orijinali silinir.
- Bir NumPy dizisindeki öğelerin hepsinin aynı veri türüne sahip olması gerekmektedir(homojen) ve böylece bellekte aynı boyutta olacaktır. Fakat nesne dizileri farklı boyutlara sahip olabileceği istisnalar mevcuttur. Bu istisna durumlarda farklı boyutlu öğelerin dizilerine izin verilmiş olur ama temelde aynı veri tipi vardır ve homojendir.

NumPy'da array sınıfları "ndarray" olarak adlandırılmıştır. Python' daki array.array tipi numpy.array tipi ile aynı değildir. Dizilerin özellikleri (ndarray) şunlardır (İpek, 2018):

- **ndarray.ndim :** Aksis (boyut) sayısını vermektedir. Bu da "rank" olarak adlandırılmaktadır.
- **ndarray.shape :** Dizinin boyutunu vermektedir. Eğer nXm matris varsa, bu (n,m) lik bir tuple veri tipinde döndürecektir.
- **ndarray.size :** Dizideki elemanların sayısını vermektedir ve bu da n*m'e eşittir.
- **ndarray.itemsize:** Dizideki elemanların her birisi için byte sayısını vermektedir. Başka bir kullanımı da ndarray.dtype.itemsize' dir.
- **ndarray.data :** Dizinin elemanlarına ulaşmak için kullanılmaktadır. Ancak "index" kullanarak elemanlara ulaşıldığından ndarray.data özelliği kullanılmamaktadır.

NumPy'nin bazı ek veri çeşitleri vardır. Tamsayılar için "i" ve işaretsiz tamsayılar için "u" gibi tek karakterli veri çeşitlerini ifade eder (Balat, 2021).

2.1.2.4. Sklearn (Scikit-Learn)

Scikit-learn projesi 2007 yılında David Cournapeau tarafından bir Google Summer of Code projesi olarak başlamıştır. 2010 yılında, Fransız Bilgisayar Bilimi ve Otomasyon Araştırma Enstitüsü INRIA dahil oldu ve ilk halka açık sürüm (v0.1 beta) Ocak 2010 sonunda yayınlandı (Anonim). Sklearn kütüphanesinin API basittir ve iyi bir dokümantasyona sahiptir.

Yapay öğrenme alanında en çok kullanılan kütüphanelerden biri Scikit-Learn'dür. Bu kütüphane karar ağaçları, rastgele orman, doğrusal regresyon ve lojistik regresyon gibi birçok temel yöntemi içermektedir.

Scikit-Learn ile başka bir pakete ihtiyaç duymadan öznitelik seçmek, çapraz doğrulama yapmak, verideki eksik değerleri doldurmak ve sonuçları değerlendirilmektedir. "fit/predict" ya da "fit/transform" fonksiyonları sayesinde regresyon, kümeleme, karar ağaçları gibi yöntemleri uygulamak, veriyi ölçeklendirmek,

eksik değerleri doldurmak gibi farklı adımlarda benzer fonksiyonları kullanmak mümkündür (Yüceoğlu 2017).

Sklearn kullandığı diller şunlardır;

- Python
- CSS
- C++
- JavaScript
- HTML
- C
- Cython

2.1.2.5. TensorFlow

Google tarafından 2015 yılında geliştirilen derin öğrenme işlemlerinde kullanılan açık kaynaklı kodlu bir makine öğrenme kütüphanesidir. TensorFlow sistemi esnektir. Derin sinir ağı modelleri, eğitim ve test algoritmalarını kapsayan birçok farklı algoritmayı tanımlamak için kullanılabilir ve araştırılabilir. Konuşma tanıma, bir düzineden fazla bilgisayar biliminde ve bilgi edinme, bilgisayarla görme, coğrafi bilgi çıkarma, doğal dil işleme, robotik ve bilgisayarlı ilaç keşfi gibi diğer alanlarda araştırma yapmak ve üretime dağıtmak için kullanılmıştır (Anonim). TensorFlow'un kullanımı, kurulumu, içerdiği arayüzler ve gerekli bilgiler web sitesinde mevcuttur.

TensorFlow' un temel kullanım alanları şunlardır (Yektek, 2021);

- Dil Algılama (Language Detection)
- Ses Vasıtasıyla Arama (Voice Search)
- Metin Algılama (Text Detection)
- Görsel Algılama, Tanıma (Visual Recognition)
- Video Algılama (Video Detection)
- Zaman Serileri (Time Series)

TensorBoard, TensorFlow'da eğitilmiş modellerin görsellerini izlemek için kullanılır. TensorBoard, TensorFlow destekli modelin verileri ve nasıl davrandığını görselleştirmesini ve grafikleri analiz edip anlamasını sağlayan bir yardımcı programdır.

TensorBoard ayrıca farklı formatlarda girdiler alarak skaler, histogram, grafik, dağıtım, projektör, resim, ses ve metnin farklı görünümlerini de görüntüler.

- **Skaler**, Sınıflandırma doğruluğu gibi skaler değerleri görselleştirir.
- **Grafik,** Sinir ağı modeli gibi modellerin hesaplama grafiğini görselleştirir.
- **Dağılımlar,** Sinir ağının ağırlıkları gibi verilerin zaman içinde nasıl değiştiğini görselleştirir.
- **Histogramlar,** Dağılımları 3 boyutlu bir perspektifte gösteren dağılımın görüntüsüdür.
- **Projektör,** Kelime gömme işlemlerini görselleştirmek için kullanılabilir.
- **Resim,** Görüntü verilerini görselleştirir.
- **Ses,** Ses verilerini görselleştirir.
- **Metin,** Metin (dize) verilerini görselleştirir(Ganegedara, 2018).

TensorFlow' un temelinde Python kullanılarak geliştirilse de günümüzde C#, R, C++, Java gibi birçok dili desteklemektedir (Anonim). TensorFlow' u kütüphanesini CocaCola, airbnb, Lenovo, ebay, Intel, LinkedIn ve Bloomberg gibi şirketler kullanmaktadır (Anonim).

2.1.2.6. Keras

Keras, Python ile yazılmış açık kaynak kodlu, üst düzey bir sinir ağı kütüphanesidir. Keras, TensorFlow 2'nin üst düzey API olduğundan modern derin öğrenmeye odaklanmasını sağlayarak makine öğrenimi sorunlarını çözmek için ulaşılabilir bir arayüzdür. Keras'ın temel veri yapıları katmanlar ve modellerdir. En basit model türü olan doğrusal bir katman yığını **Sequentialmodeldir** . Daha karmaşık mimariler için ise rastgele katman grafikleri oluşturmanıza veya alt sınıflandırma yoluyla tamamen sıfırdan modeller yazmanıza olanak tanıyan **Keras işlevsel API'sini** kullanılmaktadır (Anonim).

Keras, dört temel ilkeyi kullanan bir Google mühendisi olan François Chollet tarafından geliştirildi ve sürdürüldü:

- **Modülerlik:** Bir model, bağımsız bir grafik veya dizi olarak anlaşılabilir. Bir derin öğrenme modelinin tüm çalışması, rastgele şekillerde birleştirilebilen ayrık bileşenlerdir.
- **Minimalizm:** Kütüphane gösterişsiz, yeterlidir ve bir sonuca ulaşmak için okunabilirliği en üst düzeye çıkarır.
- **Genişletilebilirlik:** Araştırmacıların yeni fikirleri denemesini ve keşfetmesini amaçlayan çerçeve içinde yeni bileşenler eklemek ve kullanmak daha kolaydır.
- **Python:** Özel dosya formatlarına sahip ayrı model dosyalar yoktur. Her şey yerel Python'dur.

Keras kütüphanesinin en büyük avantajları şunlardır (Doğan, 2020);

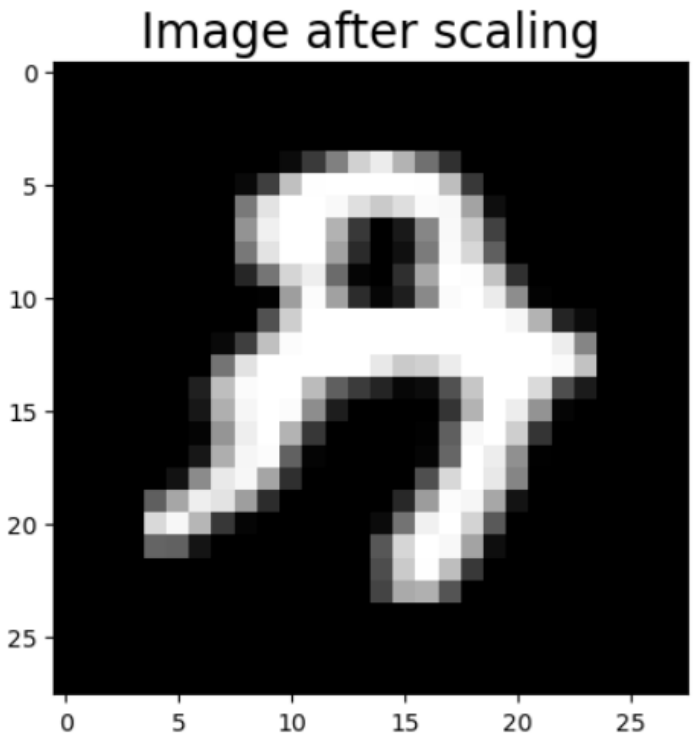
- Kolay ve hızlı bir şekilde model oluşturmamızı sağlar. Bu şekilde, yeni başlayanlar deneme yanılma yoluyla modellerdeki hangi değişiklikleri etkileyebileceklerini öğrenebilirler.
- Modelleri GPU ve CPU üzerinde sorunsuz şekilde çalıştırır. Bu sayede istediğiniz zaman GPU üzerinde işlem yaparak zamandan tasarruf sağlar.
- Sürekli veriler için Yinelemeli Sinir Ağlarını (RNN), bilgisayarlı görme modelleri için Evrişimsel Sinir Ağlarını (CNN) destekler.
- Geniş kaynak kütüphanesi ve bir topluluk sayesinde, bir soru olduğunda cevaba hızlı bir şekilde erişilebilir.

2.1.3. Oluşturulan Model ve Hiper-Parametre Ayarları

Tanımlanan algoritma Scikit Learn “fit” metodu kullanılarak veri seti eğitim verileri ile eğitilmiştir ve eğitim setinden bir örnek Şekil 19 ‘da gösterilmiştir.

```
xshuff=shuffle(xtrain)
a=xshuff[100,:]
a.shape
b=a.reshape(28,28)
b.shape
plt.imshow(b,cmap='gray')
plt.title('Image after scaling',fontsize=20)
```

```
Text(0.5, 1.0, 'Image after scaling')
```



Şekil 19. Eğitim Setinden Örnek

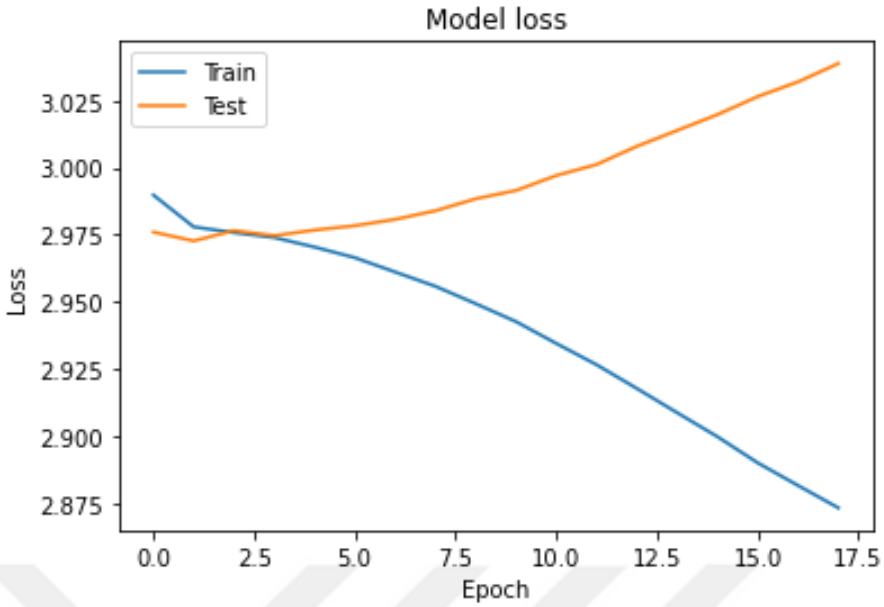
Evrişimsel Sinir Ağı (Convolutional Neural Network – CNN) için seyreltme (dropout) için [0.3] değer, dönem (epoch) sayısı 18, toptan boyutu (batch-size) 200, eniyileme algoritması (optimizer) ise "Adam" değerler Python programlama dili ile kodlanarak oluşturulmuştur. Model de gizli katman için aktivasyon fonksiyonu olarak ReLU, çıktı katmanında aktivasyon fonksiyonu olarak Softmax hiper-parametreleri belirlenmiştir (Şekil 20).

```
model=Sequential()
model.add(Conv2D(32,(5,5),input_shape=(28,28,1),activation='relu'))          #Convolutional layer
model.add(MaxPooling2D(pool_size=(2,2)))                                     #Max Pooling
model.add(Dropout(0.3))                                                       #Dropout
model.add(Flatten())                                                         #Flattening the layer
model.add(Dense(128,activation='relu'))                                      #First layer of NN
model.add(Dense(len(y.unique()),activation='softmax'))                       #Final layer of NN
model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])
his=model.fit(xtrain,ytrain,validation_data=(xtest,ytest),epochs=18,batch_size=200,verbose=2)
```

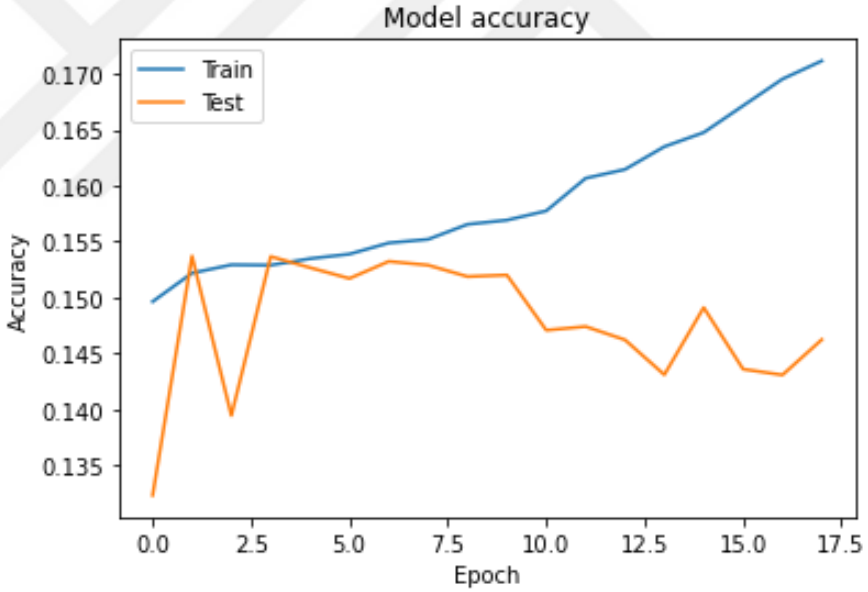
Şekil 20. CNN Modeli İçin Oluşturulan1. Katman Değerleri

```
Epoch 1/18
1413/1413 - 103s - loss: 2.9898 - accuracy: 0.1496 - val_loss: 2.9760 - val_accuracy: 0.1323
Epoch 2/18
1413/1413 - 98s - loss: 2.9779 - accuracy: 0.1522 - val_loss: 2.9727 - val_accuracy: 0.1537
Epoch 3/18
1413/1413 - 95s - loss: 2.9758 - accuracy: 0.1529 - val_loss: 2.9765 - val_accuracy: 0.1394
Epoch 4/18
1413/1413 - 97s - loss: 2.9739 - accuracy: 0.1529 - val_loss: 2.9746 - val_accuracy: 0.1537
Epoch 5/18
1413/1413 - 95s - loss: 2.9704 - accuracy: 0.1535 - val_loss: 2.9767 - val_accuracy: 0.1527
Epoch 6/18
1413/1413 - 93s - loss: 2.9664 - accuracy: 0.1539 - val_loss: 2.9783 - val_accuracy: 0.1517
Epoch 7/18
1413/1413 - 96s - loss: 2.9611 - accuracy: 0.1549 - val_loss: 2.9807 - val_accuracy: 0.1532
Epoch 8/18
1413/1413 - 96s - loss: 2.9558 - accuracy: 0.1552 - val_loss: 2.9839 - val_accuracy: 0.1529
Epoch 9/18
1413/1413 - 93s - loss: 2.9493 - accuracy: 0.1565 - val_loss: 2.9884 - val_accuracy: 0.1519
Epoch 10/18
1413/1413 - 96s - loss: 2.9426 - accuracy: 0.1569 - val_loss: 2.9915 - val_accuracy: 0.1520
Epoch 11/18
1413/1413 - 95s - loss: 2.9345 - accuracy: 0.1577 - val_loss: 2.9970 - val_accuracy: 0.1471
Epoch 12/18
1413/1413 - 93s - loss: 2.9265 - accuracy: 0.1607 - val_loss: 3.0011 - val_accuracy: 0.1474
Epoch 13/18
1413/1413 - 95s - loss: 2.9177 - accuracy: 0.1615 - val_loss: 3.0079 - val_accuracy: 0.1462
Epoch 14/18
1413/1413 - 94s - loss: 2.9087 - accuracy: 0.1635 - val_loss: 3.0138 - val_accuracy: 0.1431
Epoch 15/18
1413/1413 - 95s - loss: 2.8997 - accuracy: 0.1648 - val_loss: 3.0198 - val_accuracy: 0.1491
Epoch 16/18
1413/1413 - 96s - loss: 2.8900 - accuracy: 0.1672 - val_loss: 3.0265 - val_accuracy: 0.1436
Epoch 17/18
1413/1413 - 93s - loss: 2.8816 - accuracy: 0.1695 - val_loss: 3.0319 - val_accuracy: 0.1431
Epoch 18/18
1413/1413 - 96s - loss: 2.8732 - accuracy: 0.1712 - val_loss: 3.0388 - val_accuracy: 0.1462
```

Belirlenen model çalıştırıldığında epoch sonuçları elde edilmiştir ve bu sonuçlara göre eğitim ve test verilerinin doğrulama ve kayıp fonksiyonları grafikleri Şekil 21 ve Şekil 22’ de gösterilmiştir.



Şekil 21. CNN Modellinin İlk Katmanına Göre Kayıp Fonksiyonun Grafiği



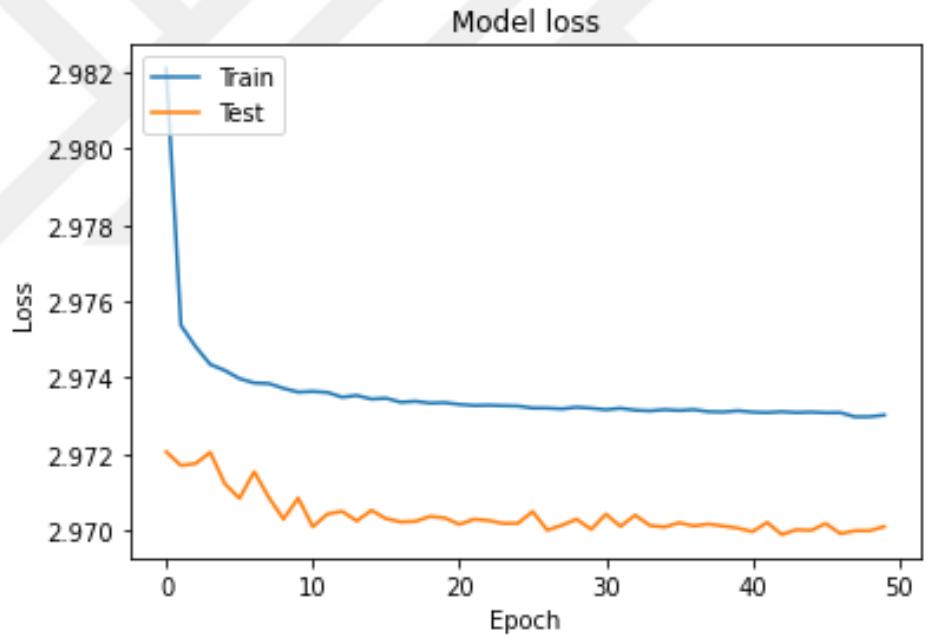
Şekil 22. CNN Modellinin İlk Katmanına Göre Doğrulama Fonksiyonun Grafiği

CNN modelinin ikinci ve üçüncü katmana göre seyreltme (dropout) için [0.3] değer, dönem (epoch) sayısı 50, toptan boyutu (batch-size) 20, eniyileme algoritması (optimizer) ise "Adam" değerler Python programlama dili ile kodlanarak oluşturulmuştur. Model de gizli katman için aktivasyon fonksiyonu olarak ReLU, çıktı katmanında aktivasyon fonksiyonu olarak Softmax hiper-parametreleri belirlenmiştir (Şekil 23).

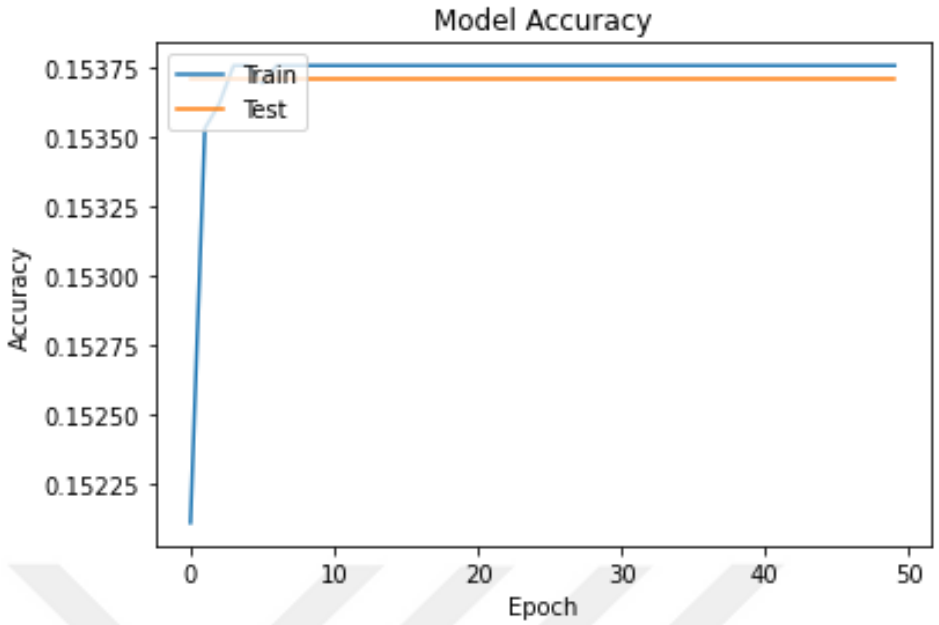
CNN modelinin ikinci ve üçüncü katmana göre eğitim ve test setinin doğrulama ve kayıp fonksiyon modelleri Şekil 24 ve Şekil 25'te gösterilmiştir.

```
model=Sequential()
model.add(Conv2D(32,(3,3),input_shape=(28,28,1),activation='relu'))          #Convolutional layer
model.add(MaxPooling2D(pool_size=(2,2)))                                     #Max Pooling
model.add(Conv2D(64,(3,3),activation='relu'))          #Second Convolutional layer
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Conv2D(128,(3,3),activation='relu'))          #Third Convolutional layer
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.3))                                                       #Dropout
model.add(Flatten())                                                         #Flattening the layer
model.add(Dense(128,activation='relu'))                                      #First layer of NN
model.add(Dense(len(y.unique()),activation='softmax'))                       #Final layer of NN
model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])
hist=model.fit(xtrain,ytrain,validation_data=(xtest,ytest),epochs=50,batch_size=20,verbose=1)
```

Şekil 23. CNN Modeli İçin Oluşturulan 2. ve 3. Katman Değerleri



Şekil 24. CNN Modellinin 2. ve 3. Katmanına Göre Kayıp Fonksiyonun Grafiği



Şekil 25. CNN Modellinin 2. ve 3. Katmanına Göre Doğrulama Fonksiyonun Grafiği

Tanımlanan algoritma Scikit Learn "predict" yöntemi kullanılarak veri setinin test verileri ile test edilmiştir. Test işlemi sırasında algoritmaya test görüntüleri verilmiştir. Algoritmanın ürettiği çıktıkların doğruluğu mevcut test etiketleri yardımıyla hesaplatılmıştır. Şekil 26 ve Şekil 27'de test setinin bir örneği verilmiştir.

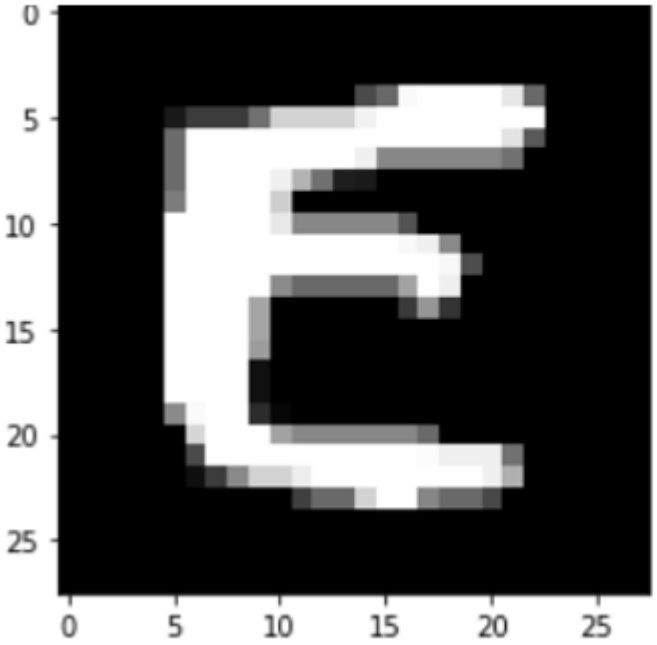
```
ypred=model.predict(xtest)
print(ypred)
```

```
[[0.03826693 0.02312997 0.06262583 ... 0.00018304 0.00016963 0.00018907]
 [0.03826693 0.02312997 0.06262583 ... 0.00018304 0.00016963 0.00018907]
 [0.03826693 0.02312997 0.06262583 ... 0.00018304 0.00016963 0.00018907]
 ...
 [0.03826693 0.02312997 0.06262583 ... 0.00018304 0.00016963 0.00018907]
 [0.03826693 0.02312997 0.06262583 ... 0.00018304 0.00016963 0.00018907]
 [0.03826693 0.02312997 0.06262583 ... 0.00018304 0.00016963 0.00018907]]
```

Şekil 26. Test Eğitim Setinin Tahmin Edilebilir Sonucu

```
fig=xtrain[100,:]
fig=fig.reshape(28,28)
plt.imshow(fig,cmap='gray')
print(ypred[650])
```

```
[3.82669270e-02 2.31299661e-02 6.26258329e-02 2.81849187e-02
 3.13586444e-02 3.22392001e-03 1.59326140e-02 1.95530523e-02
 3.24504101e-03 2.27003153e-02 1.53429322e-02 3.10763493e-02
 3.23992446e-02 5.15303649e-02 1.53911844e-01 5.11037633e-02
 1.48746027e-02 3.12806033e-02 1.24687985e-01 5.82752302e-02
 7.88757429e-02 1.19982054e-02 2.83941571e-02 1.70141216e-02
 2.82749329e-02 1.71943046e-02 1.86446297e-04 1.80931995e-04
 1.61893462e-04 1.85235462e-04 1.58544950e-04 1.83962387e-04
 1.80563802e-04 1.80594798e-04 1.62321390e-04 1.58913652e-04
 1.75173656e-04 1.64354336e-04 1.87708036e-04 1.73442430e-04
 1.69556777e-04 1.71269479e-04 1.64829326e-04 1.79114984e-04
 1.60685318e-04 1.76576854e-04 1.69983803e-04 1.64948593e-04
 1.68168306e-04 1.70493673e-04 1.63959034e-04 1.84419550e-04
 1.82395059e-04 1.99265720e-04 1.36852788e-04 1.83038399e-04
 1.69632709e-04 1.89069047e-04]
```



Şekil 27. Test Eğitim Setinin Tahmin Edilebilir Sonuç Örneği

SONUÇ VE DEĞERLENDİRME

Hızla büyüyen bir sektörde, teknolojinin gerisinde kalan belgelerin yeni sistemlere aktarılması ihtiyacı nedeniyle karakter tanıma teknolojilerinin önemi artmaktadır. El yazısı karakter tanıma teknolojileri birçok iş alanlarında hızla benimsenmeye başlamıştır. Bu alanların bazıları, bankalara gönderilen çeklerin otomatik olarak tanınarak gerekli hesap işlemlerinin elektronik ortamda gerçekleşmesi, trafik denetimi ve geçiş sistemlerinde plaka okunmasında karakter tanıma teknolojileri ile kullanılmaktadır.

Ayrıca teknolojinin gelişmesi sayesinde bilgisayar donanımı ve yazılımı, görüntü tanıma ve görüntü işleme, el yazısı karakterlerin incelenmesine olan ilgiyi artırmaktadır.

El yazısı karakter tanıma da sıkça makine öğrenmesi kullanılır. Makine öğrenmesi, en basit haliyle, bir insan tarafından açıkça programlanma zorunluluğu olmadan kendi kendine "öğrenebilen" ve herhangi bir bilgisayar programını ifade eden bir çalışma alanıdır.

El yazısı tanıma sistemleri bazı dillerde oldukça iyi çalışmaktadır. Fakat Türkçe el yazısı karakterleri ile ilgili yeterli çalışma bulunmamaktadır.

Bu çalışmanın amacı; karakter ayrıştırma ve tanıma alanında yapılmış olan çalışmalar incelenerek Türkçe el yazısı karakterlerini Keras kütüphanesini kullanarak bir tanıma sistemi gerçekleştirmektir. Bu çalışmanın diğer bir amacı, çok yönlü büyük veri kümeleri oluşturularak modelin eğitimi ve testini uygulamaktır. Bunun için tezde NIST veri seti, CNN algoritmasını ve makine öğrenmesi yöntemlerinden Pandas, Keras, Numpy, Tensorflow ve Scikit-learn kütüphaneleri kullanılmıştır.

Bu işlem derin öğrenmenin CNN yapısı ile yapılmıştır. CNN'nin karakter tanınmasındaki başarısını artırmak için NIST veri seti ve 71 kişiden Türkçe El Yazısı formuyla toplanan el yazısı örnekleri taranarak bilgisayar ortamına aktarılmıştır. Daha sonra aktarılan el yazısı örnekleri, aynı boyut ve aynı derinliğe sahip olması için 28x28 boyutunda ve siyah-beyaz görüntü dosyasına çevirdikten sonra eğitilmiştir. Veri kümesi çalıştırıldığında başarılı sonuçlar alınamadı. Bu nedenle daha verimli sonuçlar elde etmek için bu veri kümesi NIST veri kümesi ile birleştirilerek 753138 satır ve 785

sütundan oluşan bir veri kümesi elde edilmiştir. Ayrıca diğer çalışmalardan farklı olarak Keras kütüphanesi ile el yazısı karakter analizi yapılmıştır.

Önerilen yöntemin bu çalışmada uygulanmasında 1. katmandaki epoch değeri 18 olduğunda doğruluk (accuracy) ve kayıp (loss) grafiklerinde verimli sonuçlar alınamadığı görülmüştür. 2. ve 3. katmanlar için epoch değeri 50 olarak belirlendiğinde doğruluk (accuracy) ve kayıp (loss) grafiklerinde daha başarılı sonuçların olduğu görülmüştür. Bu çalışma sonucunda test setinde "0.016042793" hata oranı bulunmuştur. Literatürdeki çalışmalarla karşılaştırıldığında elde edilen başarı görülmüştür. Hem benzer model-farklı veri seti hem de farklı model-benzer veri seti kullanılan çalışmalara göre bir eğitim ve test başarısı elde edilmiştir.

KAYNAKÇA

About Keras (ty). Nisan 16, 2021 tarihinde Keras: https://keras.io/about/ adresinden alındı

About pandas (ty). Nisan 16, 2021 tarihinde Pandas: https://pandas.pydata.org/about/ adresinden alındı

ABOUT US (ty). Nisan 16, 2021 tarihinde NumPy: https://numpy.org/about/ adresinden alındı

About us Scikit-Learn (ty). Nisan 16, 2021 tarihinde scikit-learn: https://scikit-learn.org/stable/about.html adresinden alındı

Adler, A., Elad, M., & Zibulevsky, M. (2016). Compressed Learning: A Deep Neural Network Approach. arXiv preprint arXiv:1610.09615.

Akça, M. (2015, Temmuz 6). Regresyon Analizi Nedir? Nisan 15, 2021 tarihinde Mustafa Akça'nın Bloğu: https://mustafaakca.com/regresyon-analizi-nedir/ adresinden alındı

Alkan, M. A. (ty). Makine Öğrenimi. Nisan 15, 2021 tarihinde Siskon: https://www.siskon.com.tr/dosya/PDF/Makale/Makine_Ogrenimi_Machine_Learning.pdf adresinden alındı

Arslan, K. (2020). "Eğitimde Yapay Zeka ve Uygulamaları". *Batı Anadolu Eğitim Bilimleri Dergisi(BAEBD)*, 1(11), 71-88.

Aydın, Y. E. (2019, Kasım 14). Python ile Görüntü İşlemeye Giriş. Mayıs 2, 2021 tarihinde mobilhanem: https://www.mobilhanem.com/python-ile-goruntu-isleme-giris/ adresinden alındı

Balat, İ. (2021, Mart 20). Python NumPy Kullanımı - Nedir ve Nasıl Kullanılır? Nisan 16, 2021 tarihinde kerteriz: https://kerteriz.net/python-numpy-kullanimi-nedir-ve-nasil-kullanilir/ adresinden alındı

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Cambridge: Springer.

Castleman, K. (1996). *Digital Image Processing*. New Jersey: Prentice Hall.

Dilmegani, C. (2009). Image Recognition: In-depth Guide. Mayıs 15, 2022 tarihinde https://research.aimultiple.com/image-recognition/ adresinden alındı

Diri, B. (2014, Eylül 3). Makine Öğrenmesine Giriş. Nisan 15, 2021 tarihinde DOCPLAYER: https://docplayer.biz.tr/6389888-Makine-ogrenmesine-giris-machine-learning-ml.html adresinden alındı

Doğan, Ö. (2020, Eylül 5). Keras Kütüphanesi Nedir? Derin Öğrenme Modeli Oluşturma. Nisan 16, 2021 tarihinde teknoloji ORG: https://teknoloji.org/keras-kutuphanesi-nedir-derin-ogrenme-modeli-olusturma/ adresinden alındı

Ganegedara, T. (2018, Haziran 6). TensorBoard Tutorial For Beginners (article) - DataCamp. Nisan 16, 2021 tarihinde datacamp: https://www.datacamp.com/community/tutorials/tensorboard-tutorial adresinden alındı

General Python FAQ (ty). Nisan 16, 2021 tarihinde Python: https://docs.python.org/3/faq/general.html adresinden alındı

İpek, F. (2018, Nisan 24). NumPy Nedir, Neden ve Nasıl kullanılır? Nisan 16, 2021 tarihinde Kod Dağarcığı: http://www.firatipek.com/entry/15 adresinden alındı

Kantarcı, A. (2021, Ocak 1). Image Recognition in 2021: Indepth Guide. Mayıs 5, 2021 tarihinde AI Multiple: https://research.aimultiple.com/image-recognition/ adresinden alındı

Karakoç, M. (2012, Şubat). Görüntü işleme, Teknolojiler ve Uygulamlar. Uşak. Nisan 13, 2021 tarihinde https://ab.org.tr/ab12/sunum/21-goruntu_isleme-Karakoc.pdf adresinden alındı

Karakoç, M. (2012). Görüntü İşleme, Tekonojiler ve Uygulamaları. Akademik Bilişim, (s. 53-57). Uşak. Mayıs 2, 2021 tarihinde https://ab.org.tr/ab12/sunum/21-goruntu_isleme-Karakoc.pdf adresinden alındı

Kartal, E., & Özen, Z. (2017). “Dengesiz Veri Setlerinde Sınıflandırma”. *Mühendislikte Yapay Zeka Uygulamaları* (s. 109-131). içinde Sakarya: Sakarya Üniversitesi Kütüphanesi Yayınevi.

Kobayashi, K. (2002). “Birth of a Digital Phototelegraph-the Bartlane System”. *Journal of the Institute of Image Electronics Engineers of Japan*, 31(2), 244-249. doi: 10.11371/iieej.31.244

Kutlugün, M. A., Anka, F., & Çakır., M. (2017). Yapay Sinir Ağları ve K-En Yakın Komşu Algoritmalarının Birlikte Çalışma Tekniği (Ensemble) ile Metin Türü Tanıma. Haziran 23, 2021 tarihinde https://www.researchgate.net/publication/323990877_Yapay_Sinir_Aglari_ve_K-En_Yakin_Komsu_Algoritmalarinin_Birlikte_Calisma_Teknigi_Ensemble_ile_Metin_Turu_Tanima adresinden alındı

Liu, L., Shen, C., & Van Den Hengel, A. (2015). “The Treasure beneath Convolutional Layers: Cross-convolutional-layer Pooling for for Image Classificatio”. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (s. 4749-4757). Boston: Institute of Electrical and Electronics Engineers (IEEE).

Machine Learning (Makine Öğrenmesi) Nedir? | SmartPro Teknoloji. (2020). Nisan 15, 2021 tarihinde https://smartpro.com.tr/machine-learning-makine-ogrenmesi-nedir/ adresinden alındı

Mikolov, Tomas Karafiát, M., Burget, L., Jan, C., & Khudanpur, S. (2010). “Recurrent Neural Network Based Language Model”. *1th Annual Conference of the International Speech Communication Association* (s. 1045-1048). Japan: ISCA.

Mori, S., Suen, C. Y., & Yamamoto, K. (1992). Historical Review of OCR Research and Development. IEEE, 1029-1058. doi:10.1109/5.156468

Mukul, S. S. (2013). “The Origins of Digital Image Processing & Application fields in Digital Image Processing Medical Images”. *NCEAM*, 1(2), Mayıs 2, 2021 tarihinde alındı

NIST Special Database 19 (2016). Mart 12, 2022 tarihinde https://www.nist.gov/srd/nist-special-database-19 adresinden alındı

NumPy (ty). Nisan 16, 2021 tarihinde Wikipedia: https://en.wikipedia.org/wiki/NumPy#cite_note-Nature-5 adresinden alındı

Özdemir, H. (2014). Python nedir? Nisan 13, 2021 tarihinde https://www.pythontr.com/makale/python-nedir-235 adresinden alındı

Package overview (ty). Nisan 16, 2021 tarihinde Pandas: https://pandas.pydata.org/pandas-docs/stable/getting_started/overview.html adresinden alındı

Pandas (ty). Nisan 16, 2021 tarihinde Vikipedi: https://tr.wikipedia.org/wiki/Pandas#cite_note-5 adresinden alındı

Plamondon, R., & Srihari, S. N. (200). "On-Line and Off-Line Handwriting Recognition: A Comprehensive Survey". *1EEE Transactions on Pattern Analysis and Machine Intelligence*, 1(22), 63-83.

Priddy, K. L., & Keller, P. E. (2005). *Artificial Neural Networks: An Introduction* (Cilt 1). Washington: Spie Press.

Python Nedir? (ty). Nisan 16, 2021 tarihinde Mediaclick: https://www.mediaclick.com.tr/tr/blog/python-nedir adresinden alındı

Python nedir? (2016, Haziran 24). Nisan 16, 2021 tarihinde pythontr.com: https://www.pythontr.com/makale/python-nedir-235 adresinden alındı

Python Nedir? Python Hakkında Herşey (ty). Bilginç IT Akademi: https://bilginc.com/tr/blog/158/python-nedir-python-hakkinda-hersey adresinden alındı

Python Programlama Dili Hakkında Bilinmesi Gerekenler (ty). Nisan 16, 2021 tarihinde NedirNasılKimdir: https://www.nedirnasilkimdir.com/python-programlama-dili-hakkinda-bilinmesi-gerekenler/ adresinden alındı

Salouhou, Aoudou (2019). *El Yazısı Karakter Tanıma ve Resim Sınıflandırmada Derin Öğrenme Yaklaşımları*. (Yayımlanmış Yüksek Lisans Tezi). İstanbul: Fatih Sultan Mehmet Vakif Üniversitesi Lïsansüstü Eğitïm Enstitüsü

Samuel, A. L. (1959). "Some Studies in Machine Learning Using the Game of Checkers". *BM Journal of Research and Development*, 3(3), 210-229. doi:10.1147/rd.33.0210

Saylan, S., & Gezer, M. (2017). "Makine Öğrenmesi Kütüphaneleri". *Mühendislikte Yapay Zeka ve Uygulamaları* (s. 53-64). içinde Sakarya: Sakarya Üniversitesi Kütüphanesi Yayınevi.

scikit-learn (ty). Nisan 16, 2021 tarihinde Wikipedia: https://en.wikipedia.org/wiki/Scikit-learn#cite_note-wikidata-8841d81671926d02174fc8206c98988633ebc55e-v3-2 adresinden alındı

scikit-learn (2021, Ocak). Nisan 16, 2021 tarihinde Open HubBlack Duck Open Hub: https://www.openhub.net/p/scikit-learn/analyses/latest/languages_summary adresinden alındı

Ser, G., & Bati, C. T. (2019). "Derin Sinir Ağları ile En İyi Modelin Belirlenmesi: Mantar Verileri Üzerine Keras Uygulaması". *Yüzüncü Yıl Üniversitesi Tarım Bilimleri Dergisi Cilt,* 29(3), 406-417. doi:10.29133/YYUTBD.505086

SmartPro Bilgisayar Akademisi. (2020, Şubat 21). Nisan 15, 2021 tarihinde SmartPro: https://smartpro.com.tr/machine-learning-makine-ogrenmesi-nedir/ adresinden alındı

Soytürk, Z. (ty). Python Nedir? Nisan 13, 2021 tarihinde Mediaclick: https://www.mediaclick.com.tr/tr/blog/python-nedir adresinden alındı

Sponsors (ty). Nisan 16, 2021 tarihinde Pandas: https://pandas.pydata.org/uğabout/sponsors.html adresinden alındı

Srivastava, N., Hinton, G., Krizhevsky, A., & Salakhutdinov, R. (2014). "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". *Journal of Machine Learning Research*, (15), 1929-1958.

Şahin, O. (2020, Haziran 6). Pandas Nedir? Nasıl Kullanılır? – Python Kütüphanesi. Nisan 16, 2021 tarihinde teknoloji ORG: https://teknoloji.org/pandas-nedir-nasil-kullanilir-python-kutuphanesi/ adresinden alındı

Şeker, A., Diri, B., & Balık, H. H. (2017). "Derin Öğrenme Yöntemleri ve Uygulamaları Hakkında Bir İnceleme". *Gazi Mühendislik Bilimleri Dergisi*, 3(3), 47-64.

TensorFlow (ty). Nisan 16, 2021 tarihinde TensorFlow: https://www.tensorflow.org/ adresinden alındı

Therrien, C. W. (1989). *Decision, Estimation, and Classification : An Introduction to Pattern Recognition and Related Topics* (Cilt 1). Wiley.

Uğur, Aybars, ve Aydın, D. (2006). *Ant System Algoritmasının Java İle Görselleştirilmesi.* Denizli: Bilgi Teknolojileri Kongresi IV Akademik Bilişim.

Uzun, E. (2007). *İnternet Tabanlı Bilgi Erişimi Destekli Bir Otomatik Öğrenme Sistemi.* (Yayımlanmış Doktora Lisans Tezi) Edirne: Trakya Üniversitesi Fen Bilimleri Enstitüsü.

Verma, B., Kulkarni, S., & Blumenstein, M. (1998). "Recent Achievements in Off-line Handwriting Recognition Systems Author". *International Conference on Computational Intelligence and Multimedia Applications 1998*. Australia: Griffith University.

Why use Keras? (tarih yok). Nisan 16, 2021 tarihinde Internet Archive: https://web.archive.org/web/20190724090056/https://keras.io/why-use-keras/ adresinden alındı

Woods, R. C. (2002). *Digital Image Processing*. Pearson.

Yavuz, A. (2021). *Derin Öğrenme Algoritmaları İle Trafik İşaret ve Levhalarının Tanımlanması.* (Yayımlanmış Yüksek Lisans Tezi). Denizli: Pamukkale Üniversitesi Fen Bilimleri Enstitüsü.

Yediren, Ç. M. (2020, Kasım 3). Keras ile Python Derin Öğrenme Nedir? Nisan 16, 2021 tarihinde codernsoft: https://www.codernsoft.com/keras-ile-python-derin-ogrenme-nedir/ adresinden alındı

Yektek, K. (2021, Nisan 4). TensorFlow Nedir? Nisan 16, 2021 tarihinde coderspace: https://www.coderspace.io/blog/tensorflow-nedir/ adresinden alındı

Yeşilyurt, H. (2021, Nisan 15). OpenCV-Görüntü İşleme Nedir? Mayıs 2, 2021 tarihinde Mert Mekatronik: https://mertmekatronik.com/python-opencv-goruntu-%C4%B1sleme adresinden alındı

Yıldız, M. (2013). "Yazma Güçlüğü (Disgrafi) Olan Bir Ilkokul 2. Sınıf Öğrencisinin El Yazısı Okunaklılığının Geliştirilmesi: Eylem Araştırması". *Uşak Üniversitesi Sosyal Bilimler Dergisi*, 281-310.

Yılmaz, A. (2021). "Diagnosing Covid-19 From X-ray Images With Using Multi-channel Cnn Architecture". *Journal of the Faculty of Engineering and Architecture of Gazi University*, 4(36), 1761-1773.

Yüceoğlu, B. (2017, Kasım 23). Scikit-Learn İle Veri Analitiğine Giriş. Nisan 16, 2021 tarihinde Veri Defteri: http://www.veridefteri.com/2017/11/23/scikit-learn-ile-veri-analitigine-giris/ adresinden alındı

ÖZGEÇMİŞ

Kişisel Bilgiler	
Adı Soyadı	Özge TUNCER
Doğum Yeri ve Tarihi	
Eğitim Durumu	
Lisans Öğrenimi	
Y. Lisans Öğrenimi	
Bildiği Yabancı Diller	
Bilimsel Faaliyetleri	
İş Deneyimi	
Stajlar	
Projeler	
Çalıştığı Kurumlar	
İletişim	
E-Posta Adresi	
Tarih	