



Research Article

EVRIŞİMLİ SİNİR AĞLARI KULLANARAK OPTİK KARAKTER TANIMA

OPTICAL CHARACTER RECOGNITION USING CONVENTIONAL NEURAL NETWORK

Ahmet ÜNAL¹ | Esra ÜNAL² | Dilcan GÜLER^{3*}

¹ Bilg. Müh. İzmir Kâtip Çelebi Üniversitesi, Bilgi İşlem Daire Başkanlığı, İzmir, Türkiye, ORCID: 0000-0002-5634-123X

² Öğr. Gör. İzmir Kâtip Çelebi Üniversitesi, Bilgi İşlem Daire Başkanlığı, İzmir, Türkiye, ORCID: 0000-0002-2669-9498

³ Öğr. Gör. İzmir Kâtip Çelebi Üniversitesi, Bilgi İşlem Daire Başkanlığı, İzmir, Türkiye, ORCID: 0000-0002-8531-6842

Article Info:

Received : Feb 28, 2023

Revised : Mar 29, 2023

Accepted : Apr 07, 2023

Keywords:

Convolutional neural networks

Optical character recognition

(OCR)

Digitization

Anahtar Kelimeler:

Evrişimli sinir ağları

Optik karakter tanıma (OKT)

Sayısallaştırma

DOI: 10.46238/jobda.1257840

ABSTRACT

Technology has been an important part of human life. People can carry out all processes of their daily and business lives quickly and accurately via smart devices and computers anywhere at any time. This ensures that time is used more efficiently. All correspondence and documents in business life are done on computers. For this reason, the use of text recognition and character recognition technology saves cost and time. In this study, it is aimed to determine the text expressions in the picture by digitization method. In this direction, the steps of converting a scanned document to grayscale, removing the noise on the page, detecting the lines on the document, separating the detected lines into words, obtaining letters from the separated words and detecting the letter obtained by convolutional neural network algorithms will be examined through an example.

ÖZ

Teknoloji insan hayatının önemli bir parçası olmuştur. İnsanlar her an her yerde akıllı cihazlar ve bilgisayarlar aracılığı ile gerek gündelik gerekse iş hayatlarına ait tüm süreçleri hızlı ve doğru olarak gerçekleştirebilmektedir. Bu durum zamanın daha verimli kullanılmasını sağlamaktadır. İş hayatında tüm yazışmalar, belgeler bilgisayarlar üzerinden yapılmaktadır. Bu nedenle metin tanıma ve karakter tanıma teknolojisinin kullanılması maliyetten ve zamandan tasarruf sağlanmaktadır. Bu çalışmada resim içerisindeki metin ifadelerin sayısallaştırma yöntemi ile tespit edilmesi amaçlanmaktadır. Bu doğrultuda taranan bir dokümanın gri tona çevrilmesi, sayfa üzerindeki gürültünün temizlenmesi, doküman üzerinde satırların tespiti, tespit edilen satırların kelimelerine ayrılması, ayrılan kelimelerden harflerin elde edilmesi ve evrişimli sinir ağı algoritmaları ile elde edilen harfin tespitinin yapılması adımları bir örnek üzerinden incelenecektir.

© 2023 JOBDA All rights reserved

* Corresponding author.

E-Posta: dilcan.guler@ikcu.edu.tr

1 | GİRİŞ

Günümüzde yazı tanıma konusunda birçok teknolojik gelişme yaşanmaktadır. Bu geliştirmeler sayesinde gerek el yazısı gerekse makine yazılarının tanınmasında büyük ilerlemeler kaydedilmiştir. Böylece bilgisayar ya da daktiloda yazılmış yazıları otomatik olarak tanıyan uygulamalar innsa hayatının içinde yer almış ve özellikle iş yaşamında süreçlerin verimliliğini arttırıcı etki yaratmıştır. Bu bağlamda karakter tanıma çok araştırılan bir konu olmuş ve örüntü tanıma alanında pratik uygulamalar ortaya çıkarmıştır. Bu alandaki çalışmalar birçok yararlı yöntem ve bilginin oluşmasını sağlayarak hem teori hem de pratikte diğer alanların ilerlemesini büyük ölçüde etkilemiştir.

Karakter tanıma kısaca çekilen görüntüyü işler, karakterleri çıkartır ve bu karakterleri tanıır. Optik karakter tanıma, her türlü farklı karakteri tanıma sorunuyla ilgilenmektedir. Hem el yazısı hem de basılı karakterleri tanınabilir ve makine tarafından okunabilir dijital bir veri formatına dönüştürmektedir. Optik karakter tanıma geniş bir teknoloji yelpazesıyla birleştirilebildiği için optik karakter tanıma yazılımını kullanma olanakları yaygındır. Optik karakter tanıma kullanım alanları için doküman işleme, sipariş işleme, personel kayıt yönetimi, nüfus sayımı formlarının işlenmesi, çek işleme, ödeme işleme, emekli fonu işleme, CAPTCHA aşma gibi çalışmaların (Koyun ve Afşin, 2017) yanı sıra uygulamadan birkaç örnek aşağıdaki gibi verilebilir (Erdoğan ve Tümer, 2021; Koyun ve Afşin, 2017);

- Pasaportlar ve kimliklerin dijitalleştirilmesi ile kişilerin tanımlama ve kaydetme süreci hızlandırılır.
- Optik karakter tanıma yazılımı ile banka müşterileri Uluslararası Banka Hesap Numarası (IBAN)'larını tarayabilirler ve işlemlerini daha hızlı gerçekleştirebilirler.
- Bankalara gönderilen çekler otomatik olarak muhasebeleştirilir ve gerekli hesap işlemleri elektronik ortamda yapılır.
- Doktor el yazısının anlaşılır bir formata getirilmesi ile reçete örneklerindeki el yazısından anahtar kelimeler tespit edilmiştir. Belirtilen bu kelimeler ile yazılan reçetede istenilen ilacın ne olduğu anlaşılabilir.
- Mektuplardaki adreslerin tanınması ve posta koduna göre otomatik ayrıştırılması için postanelerde mektupların gideceği adrese göre zarfların otomatik olarak doğru şekilde ayrılması sağlanmaya çalışılmıştır.

Bilgisayarlar bugün insan gibi karar verebilir ve olaylar arasındaki ilişkiyi öğrenebilir durumdadır. Özellikle Endüstri 4.0 çalışmaları ile gündeme gelen yapay zeka kavramı, insan gibi davranışlar sergileme, sayısal mantık yürütme, hareket, konuşma ve ses algılama gibi birçok yeteneğe sahip yazılımsal ve donanımsal

sistemler bütünüdür. Kısaca yapay zeka ile bilgisayarlar, insanlar gibi düşünebilmekte ve karar verebilmektedir. Yapay zekanın en yaygın kullanımlarından birisi de görüntü işleme alanıdır. Bu kapsamda, yapay sinir ağları (YSA), yapay zeka araçlarından biridir. Derin öğrenme yöntemleri ise çok katmanlı yapay sinir ağı algoritmalarıdır. Sınıflandırma ve nesne tanıma işlemleri için en yaygın kullanılan algoritmalarından birisi evrişimli sinir ağı algoritmasıdır. Evrişimli sinir ağı, makine öğrenimi problemlerinde oldukça iyi bir performansa sahiptir.

2 | KARAKTER TANIMA

Karakter tanıma, elle ya da bir cihaz ile yazılmış yazıların bilgisayar uygulamaları ile tanınması işlemidir. Karakter tanıma teknolojilerinin gelişimiyle birlikte bir çok alanda kullanılabilirliği de ortaya çıkmıştır (Şekerci ve Kandemir, 2009). Bu alanlar ile ilgili; postanelerde mektupların posta koduna göre ayrıştırılması işlemlerinde, bankalarda çeklerin tanınması ve çevrimiçi ortamlarda işlemlerinin yapılabilmesinde, kütüphanelerde katalogların elektronik ortama kaydedilmesinde, araç plaka tanıma sistemlerinde, reklam panolarında yer alan afişlerin okunmasında kullanımı örnek olarak verilebilir.

Örneğin; mektupların, üzerinde bulunan posta koduna göre ayrıştırılması, bankalara yollanan çeklerin otomatik olarak tanınıp gerekli hesap işlemlerinin elektronik ortamlarda gerçekleştirilmesi, kütüphanelerdeki kitapların bilgisayar ortamına aktarılması, otoparklarda, geçiş kontrolünün olduğu alanlarda ve plaka tanıma sistemlerinde, reklam, afiş, market panolarının okunması gibi projelerde karakter tanıma teknolojileri yaygın olarak kullanılmaktadır (Bektaş, 2014; Bektaş vd, 2016)

Karakter tanıma, makine baskısı yazıların ve el yazılarının tanınması olarak iki kategoriye ayrılmaktadır. Makine baskısı ve el yazısı karakterler arasında büyük bir farklılık mevcuttur. El yazısında karakterler arasındaki farklılıklar makine baskısı karakterlere göre oldukça büyüktür. Makine baskısı yazılar, sabit ve belli bir karakter ölçüsüne sahipken, el yazıları kişiden kişiye farklılık göstermekle birlikte farklı tarzlarda farklı ölçülerde yazılabilir. Aynı kişinin bile farklı zamanlarda el yazısı değişikliği gösterebilmektedir. Burada karakterin büyüklüğü farklılaştığı gibi şekli de farklılık gösterebilmektedir (Şekerci ve Kandemir, 2009). Karakter tanıma işlemi, ilk olarak sinyal ve görüntü işleme algoritmalarıyla arka planın ayrıştırılması ile görüntüde var olan gürültünün temizlenmesi gibi işlemleri yapılır. Sonra bölümlenme yöntemi ile yazı karakterlerinin birbirinden ayrılması işlemleri yapılır (Şengür ve Türkoğlu, 2004)

2.1 | Karakter Tanıma Türleri

Karakter tanıma işlemi Çevirim içi karakter tanıma, Çevirim dışı karakter tanıma, Optik karakter tanıma, El

yazısı karakter tanıma, Makine yazısı karakter tanıma olmak üzere çeşitlere ayrılmaktadır (Şekerci ve Kandemir, 2009).

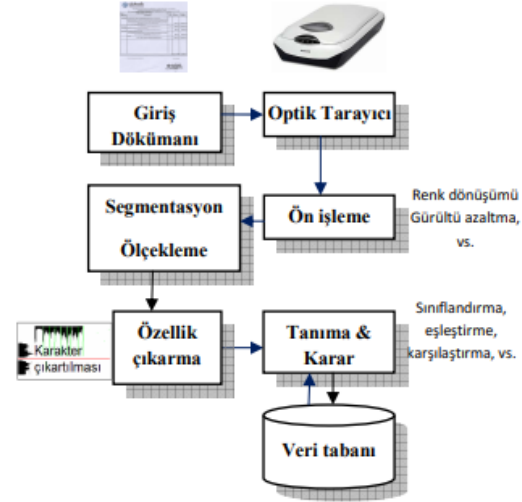
Çevirim içi karakter tanıma, karakterin yazılması anında çalışan tanıma yöntemidir. Bu yöntemi uygulayan uygulamalar, yazım esnasında elde edilen bilgileri kullanmaktadır. Karakterler özel bir kalemle hassas ekrana yazılır ve bilgisayar sistemi, bu kalemin üzerinden geçtiği pikselleri boyar. Bu yöntem ile karakter yazılırken, yazının hızı, yönü ve sırası gibi bilgiler elde edilir. Günümüzde, özellikle taşınabilir el bilgisayarlarında, tabletlerde ve cep telefonlarında, bu karakter tanıma yöntemini kullanan birçok uygulama vardır. Çevirim içi karakter tanımadada, kelimeler ve karakterler çalışma anında yazıldığı kadar hızlı tanınır. Çevirim içi yöntemler karakteri kalemin yüzeye değdiği anda oluşan pozisyonundan elde eder. Çevirim içi karakter tanımanın avantajı, kullanıcının işlem sırasında sisteme yardımcı olabilmesinin sağlanmasıdır. Bu yöntemlerden birisi kullanıcının harf bittiğinde bir işaretle bunu belirtmesidir. Diğer yöntemde karakterler arasındaki mesafenin dikkate alınmasıdır. Burada mesafe tespit edilirken kalemin yüzeyden kalktığı ve tekrar yüzeye konulduğu zaman arasındaki farkın belli bir değeri geçmesiyle yeni bir karaktere başlandığı varsayılmaktadır (Uzun ve Çağiltay, 2012)

Çevirim dışı karakter tanıma, fiziksel belge üzerindeki yazıların tanınması için kullanılmaktadır. Çevirim dışı yöntem de makine yazısı ve el yazısı tanıma şeklinde 2 kategori vardır. Her iki kategoride de tarayıcı ya da kameradan elde edilen metnin görüntüsü işlenir (Şekerci ve Kandemir, 2009). Elle veya makine ile yazılmış yazıların bilgisayar tarafından tanınmasına en genel haliyle Optik Karakter Tanıma denilmektedir. Optik karakter tanıma, "Optical character recognition", literatürde OCR olarak kullanılmaktadır. OCR, makine çıktısı, daktilo ya da bilgisayar gibi, olan metinlerin tarayıcı yardımıyla sisteme alınıp yazı karakterlerinin tespiti için kullanılmaktadır (Verma vd., 1998).

İlk bilgisayar destekli OCR Sistemi, GİSMO, 1950'li yıllarda David Shepard tarafından Amerika'da geliştirilmiş olup 23 harfi tanıyabilmekteydi. OCR sistemi alanında ilk patenti 1929 senesinde sahibi Alman Tauscheck almıştır (Musayev, 2004). 1997 yılında Mani ve Srinivasan, yapay sinir ağlarını kullanan bir OCR sistemini; ilki 1994 yılında Ahmed tarafından ve sonra 1997 yılında Alshebeili vd. tarafından Arap alfabesini tanıyan bir OCR sistemini; 1998 yılında Inoue vd. Japonca metinleri ve matematiksel formülleri ayrı ayrı tanıyan bir OCR sistemini; Singh vd. yapay sinir ağları yardımı ile Hint alfabesini tanıyan bir optik karakter tanıma sistemini geliştirmişlerdir. 1999 yılında, Gorski vd., bankalarda çekleri işlemek için çek tanıma sistemi, 2007 yılında Niise, Amerikan posta kodlarını okuyan yapay sinir ağı kullanan bir OCR sistemi geliştirmiştir (Kır vd., 2001).

Tipik bir OCR sistemi görüntü tarayıcı, OCR yazılımı ve donanımı ile çıkış arayüzü olma üzere üç mantıksal bileşen içerir. Görüntü tarayıcı ile metin görüntüleri optik olarak tanınır ve bu görüntüler OCR yazılım ve donanımları ile işlenir. Bu süreç belge analizi, tanıma ve bağlamsal işleme olarak üç aşamada gerçekleştirilir ve çıkış ara birimi ile sonuçlar paylaşımına açılır. (Srihari vd, 2003).

OCR sisteminin genel yapısı Şekil 1'de gösterilmiştir. Optik bir cihaz ile taranan basılı belgeye ait karakterler öncelikle sayısallaştırılmaktadır.



Şekil 1: OCR Genel Yapısı (Kır vd., 2001)

Karakterlerin şekillerinin karmaşıklığı ve gürültü, karakter tanımanın hızını ve doğruluğunu etkileyen faktörlerdendir. OCR sisteminde belge tarandıktan sonraki işlem adımları sırasıyla ön işleme, segmentasyon ve ölçekleme, özellik çıkartma ve karakter tanımadır (Kır vd., 2001).

OCR, taranan kağıt belgeler, PDF dosyaları veya dijital kamerayla çekilen görüntüler gibi farklı türde belgeleri düzenlenebilir ve aranabilir verilere dönüştürmenizi sağlayan bir teknolojidir. Dijital fotoğraf makinesiyle çekilen resimler, taranan belgelerden veya resimlerden farklıdır. Çoğu OCR uygulamasının metni doğru şekilde tanımasını zorlaştıran, genellikle kenarlarda bozulma ve loş ışık gibi kusurları vardır (Mithe vd, 2013). OCR sistemlerinin, hızlı veri girişi ve metin işlemeyi izin verme, metin girişinde insanların yazım hatalarını azaltma ve hızlı bilgi bulmaya imkan sağlama gibi avantajları vardır (Kır vd., 2001). OCR uygulamaları, makine öğrenim yöntemlerinden olan yapay sinir ağları algoritmalarını kullanarak karakter tanımak işlemi yaparlar (Koyun ve Afşin, 2017).

2.2 | Karakter Tanıma Süreci

Optik karakter tanımanın temel ve önemli aşamaları Kır vd. (2001)'ne benzer bir şekilde Tilki (2020) tarafından ön işleme, segmentasyon, normalizasyon, özellik çıkarımı ve sınıflandırma olarak

gruplandırılmıştır. Bu aşamalar maddeler halinde aşağıda özetle açıklanmaktadır (Tilki, 2020):

- Ön işleme ile elde edilen görüntüde oluşabilecek istenmeyen görüntüleri ve gürültüyü elimine etmek amaçlanmaktadır. Gürültü azaltmadaki gaye ise görüntüdeki istenmeyen herhangi bir bit örüntüsünü tespit ederek bunların veri olarak kullanılmamasını sağlamak ve görüntüyü saflaştırmaktır. Gürültüyü azaltmak için ise ikili ya da gri görüntüler kullanılmaktadır.
- Segmentasyon aşamasında ise ön işlemede istenmeyen parçaların elimine edilmesi ardından elde edilen saf görüntünün analizini kolaylaştırmak için, görüntüyü bileşenlerine ayrıştırma işlemi ifade etmektedir.
- Normalizasyon aşamasında, segmentasyon işleminden elde edilen ayrıştırılmış çıktı bir matris formatına getirilmekte ve veriler normalize edilerek özellik çıkarma aşamasında kullanımına olanak sağlamaktadır.
- Özellik Çıkarımı, normalizasyon aşamasında elde edilen matrislerin işleme alınmasıyla elde edilecek özellik vektörlerini ortaya çıkarmayı sağlayacak ilgili özellikleri ortaya çıkarma işlemidir. Özellik vektörleri sayesinde tanıma işlemini gerçekleştirmek daha pratik bir hale gelmektedir.
- Sınıflandırma aşamasında ise daha önceden tanımlanmış ve tasarlanmış çeşitli yöntemler sayesinde her bir görüntü örneği özelliklerine göre sınıflara ayrılmaktadır.
- Karakter tanıma işlemlerinin yapılabilmesi için öncelikle görüntünün ön işleme tabi tutulması gerekmektedir (Steinherz, T. vd, 1999). Yazarlar tarafından oluşturulan ve çalışmada kullanılan karakter tanıma süreci Şekil 2’de gösterilmiştir.



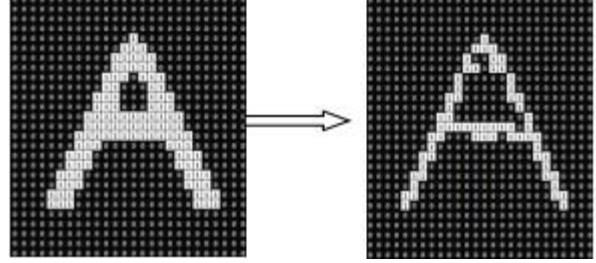
Şekil 2: Karakter Tanıma Süreci

Tarama: Fiziksel belgenin tarayıcı tarafından bilgisayarın tanıyabileceği formata dönüştürülmesidir. Bu dönüştürme tipi genelde jpg ya da pdf formatındadır. Bilgisayar ortamında görüntüler piksel haldedir. Piksel formatta, taranmış olan belgenin bütün görselinin eksiksiz aktarımı için bütün renklerin bilgisayar ortamına aktarılması gerekmektedir. Fiziksel belge sayısallaştırılarak bilgisayar tarafından tanınabilir bir yapıya dönüştürülmelidir. Bu işlem RGBA(R-kırmızı, G-yeşil, B-mavi, A alpha opacity) formatında olmaktadır. RGB, 3 rengin kullanımı ile bütün renkleri temsil edebilen bir yapıyı oluşturmaktadır (Ovatman, 2005).

Gri Format: Taranmış ve RGBA formatında olan görüntünün, beyaz ile siyahın ara tonları (Gri) ile temsil edileceği hale dönüştürülmesidir. (Şengür ve Türkoğlu, 2004)

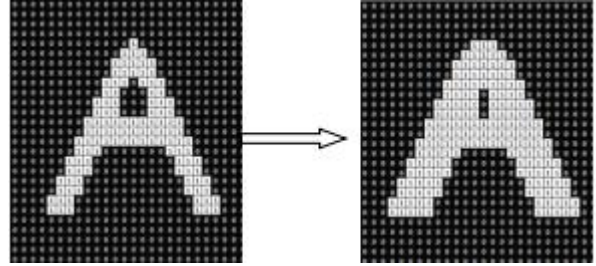
Siyah Beyaz: Gri tondaki görüntünün belirlenen bir eşik değerin kullanılmasıyla, görüntünün siyah-beyazla temsil edilmesi işlemidir. (Şengür ve Türkoğlu, 2004)

Aşındırma: Görüntü üzerindeki gürültülerin tespit edilmesi ve temizlenmesi için kullanılan bir yöntemdir. Görüntü üzerindeki siyah olarak görünen piksellerin yanında bulunan diğer siyah piksellerin daraltma yöntemi ile erozyon işlemidir (Atalı vd., 2016).



Şekil 3: Aşındırma İşlemi (Atalı vd., 2016).

Genişletme: Aşındırmaya tabi tutulan görüntünün daha net ve anlaşılabilir yapıya dönüştürülmesi için piksel tabanlı olarak yapılan bir işlemidir (Atalı vd., 2016).



Şekil 4: Genişletme İşlemi (Atalı vd., 2016).

OCR uygulamalarında yapay sinir ağının (YSA) kullanılması, kodu önemli ölçüde basitleştirebilir ve iyi performans elde ederken tanıma kalitesini iyileştirebilir. OCR'de sinir ağı kullanmanın bir başka yararı, sistemin genişletilebilirliğidir, yani başlangıçta tanımlanandan daha fazla karakter kümesini tanıyabilme yeteneğidir. Yapay sinir ağlarını kullanarak karakter tanıma algoritmasında, sinir ağının ağırlıkları geriye yayılım algoritması kullanılarak eğitilerek ayarlanmıştır. Her karakterin boyutu, giriş olarak 504 * 1 dizi verecek şekilde sütun şeklinde düzenlenmiş 28'e 18 pikseldir. Sinir ağını eğitmek için her biri 0'dan 9'a kadar rakamlar içeren farklı kümeler oluşturduk. Buna Blok eğitimi denir. Ağın tamamı, projeye statik olarak bağlı olan bir kitaplık olarak uygulandı. Bu, sinir ağı kodunu ön işleme ve bölümlenme kodunun geri kalanından ayırmaya yardımcı oldu. Ayrıca program için gereken belleğin azaltılmasına da yardımcı oldu. (Shah vd., 2009)

3 | YAPAY SİNİR AĞLARI ALGORİTMALARI

Yapay sinir ağları (YSA), yapay zekâ araçlarındandır. Yapay sinir ağları, örneklerle ilgili bilgiler toplamakta, genellemeler yapmakta ve daha sonra hiç görmediği örnekler ile karşılaştırılınca öğrendiği bilgileri kullanarak o örnekler hakkında karar verebilmektedir.

YSA temel olarak insan beyninin işlevlerini taklit eden çok büyük paralel hesaplama modelleridir. (Dongare et. al., 2012).Yapay sinir ağları, erişilebilir bilgilerdeki çok karmaşık eğilimleri karakterize edebilen bir teknolojidir ve örüntü tanıma ve eşleştirme, kümeleme ve sınıflandırma ile ilgili sorunları çözmek için daha etkilidir (Ataseven, 2013)

Yapay sinir ağları, paralel bağlantıya sahip olan bir yapıyla organize edilmiş nöron yada node katman gruplarından oluşmaktadır. Yapay sinir ağlarında öğrenme, eğitime yoluyla örneklerin kullanılmasıyla olmaktadır, girdi/çıkı verilerinin bir eğitime algoritması ile kullanarak bağlantı ağırlıkları arasında (weights of the synapses) bir yakınlık sağlanana kadar tekrar tekrar algoritmanın çalışmasıyla olmaktadır; öğrenme süreci matematiksel olarak modellenmektedir (Alpaslan vd., 2013).

Yapay sinir ağları (YSA), biyolojik sinir sistemlerinin matematiksel modellerinin geliştirilmesi olarak geliştirilmiştir. McCulloch ve Pitts tarafından basitleştirilmiş nöronların tanıtılmasından sonra sinir ağlarına (bağlantıcı modeller veya paralel dağıtılmış işleme olarak da bilinir) ilk ilgi dalgası ortaya çıkmıştır (Abraham, 2005). YSA uygulamasının iyi bir avantajı, büyük girdilere sahip karmaşık doğal sistemlerden modelleri kullanımı kolay ve daha doğru hale getirebilmesidir (Abiodun vd., 2018)

Yapay sinir ağlarının başlıca avantajları aşağıda belirtilmiştir (Yıldırım, 2021):

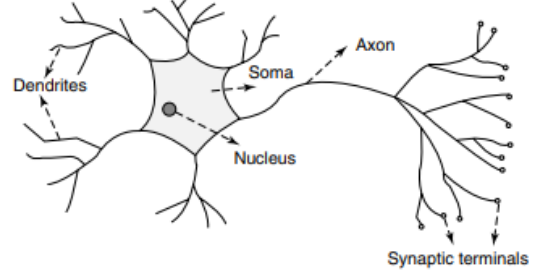
- Yapay Sinir Ağları birden fazla hücreden oluşmakla birlikte bu hücreler aynı anda çalışarak karmaşık işleri gerçekleştirebilmekte, gerçek zamanlı bilgiyi işleyebilmektedir.
- YSA, farklı öğrenme algoritmalarıyla öğrenme kabiliyeti vardır.
- Öğrenme ile bilinmeyen çıktılar için bilgi üretebilir.
- Örüntü tanıma, eksik örüntüleri tamamlama ve sınıflandırma yapabilir.
- Hata toleransına sahiptir.
- Eksik veya belirsiz bilgiyle çalışabilir. Hatalı durumlarda dereceli bozulma (graceful degradation) gösterir.

3.1 | Yapay Sinir Ağlarının Biyolojik Temelleri

Yapay sinir ağları nöron ismi verilen ve bilgi işleme özelliği olan sinir hücrelerinden oluşmaktadır. Bu nöronlar birbirleri ile bağlanarak fonksiyonları meydana getirmektedir. Burada ulaşılmak istenen; eğitilebilir, kendi kendine organize olup öğrenebilen ve değerlendirme yapabilen sinir ağları ile insan beyninin öğrenme ve uygulama yapısını modellemeye çalışmaktır (Yıldırım, 2021).

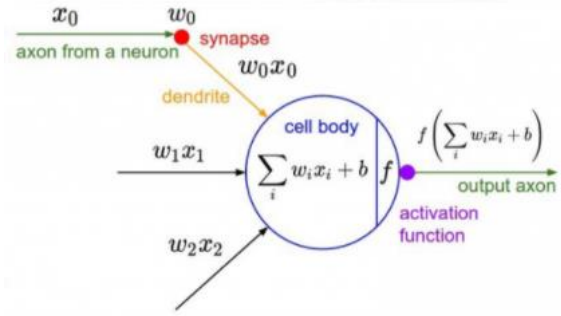
Sinir ağlarının temel işlem elemanlarına yapay nöronlar veya düğümler denir. Nöronun matematiksel modelinde, sinapsların etkileri, ilişkili giriş sinyallerinin etkisini oluşturan bağlantı ağırlıkları ile

temsil edilir. Nöron impulsu, transfer fonksiyonu tarafından dönüştürülen giriş sinyallerinin ağırlıklı toplamı olarak hesaplanır. Yapay bir nöronun öğrenme yeteneği, ağırlıkların seçilen öğrenme algoritmasına göre ayarlanmasıyla elde edilir (Abraham, 2005).



Şekil 5: Nöronun Gösterimi (Abraham, 2005)

Dentritte ağırlık(w) ve dentritlere giren bir başka nöronun da gelmiş olabilecek bir giriş değeri(x) vardır. Giriş değeri ve dentritteki ağırlık (w_0x_0) olarak nörona iletilir. Nöronun çarpma işlemi yapılır ve tüm dentritlerden gelen bu çarpımlar toplanır. Toplam bias(b) ile toplanır ve aktivasyon fonksiyonu çıkışa aktarılır. Bu, nihai çıkış olabileceği gibi bir başka hücrenin girişi de olabilir.

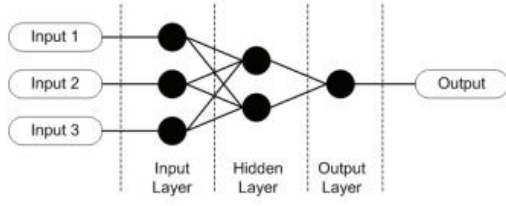


Şekil 6: Nöronun Matematiksel Modeli (Karpathy, 2015)

Yapay sinir ağlarının en temel işlemi; modelin en iyi sonucu vereceği ağırlık w ve bias değeri b parametrelerinin hesaplamaktır. Her bir sinir hücresi için hesap yapılır ve bunlar birbirine seri ya da paralel şekilde bağlanır (Yıldırım, 2021).

3.2 | Yapay Sinir Ağının Yapısı

Yapay sinir ağı, yapay sinir hücrelerinin birbirine bağlanmasıyla oluşur ve girdi, ara ve çıktı olarak 3 katmana sahiptir. Veri yapay sinir ağına girdi katmanında girer, gizli katmanda işlenir ve çıkış katmanından çıkar. Gizli birimdeki her birim, giriş katmanlarındaki tüm birimlere bağlanır ve çıktı katmanı tamamen gizli katmana bağlanır (Maind ve Wankar, 2014).



Şekil 7: Yapay Sinir Ağı (Krenker vd., 2011)

Yapay sinir ağında birden fazla gizli katman varsa çok katmanlı yapay sinir ağı, tek bir gizli katman varsa tek katmanlı yapay sinir ağı denilmektedir (Arı ve Berberler, 2017)

3.3 | Derin Öğrenme

Yapay zekâ, insan zekâsı gerektiren görevleri yerine getirebilen bilgisayar sistemlerinin teorisini ve geliştirilmesini ifade etmektedir. Derin öğrenme, gevşek bir şekilde, her katmanın bir önceki katmanın daha yüksek bir temsilini öğrenmesiyle, bir özellik algılayıcı hiyerarşisi eğitime girişimi olarak tanımlanabilir. Derin öğrenmenin ortaya çıkışı, sinir ağlarının (giderek daha büyük) kullanımında bir canlanma görmüştür (Szegedy vd., 2015). Derin öğrenme, makine öğrenmesinin bir tekniğidir ve çok katmanlı bir yapay sinir ağıdır. Birçok gizli katmandan oluşan sinir ağları, derin öğrenme diğer bir adı derin sinir ağları, hem akademik hem de endüstriyel sektörlerdeki zor problemleri çözmek için başarılı bir şekilde eğitilebilmektedir. Derin öğrenmenin mevcut uygulamaları, görüntü ve konuşma tanıma, gerçek zamanlı çeviri için görüntülerde metin tanıma, ilaç keşfi ile toksisite tahmini ve sağlık endüstrisindeki diğer birçok uygulamanın yanında arama, rezervuar karakterizasyonu, petrol üretimi sırasında izleme ve petrol sahası geliştirme dahil olmak üzere hidrokarbon endüstrisinde artan sayıda uygulama bulunmaktadır. Derin öğrenmenin güçlü noktalarından biri hiyerarşik yapısıdır. Bu katmanlı yapı, bilgilerin paylaşılmasına ve yeniden kullanılmasına izin verir. Belirli özellikleri seçmek ve gereksiz ayrıntıları reddetmek mümkündür. (Dell Aversana, 2019)

Günümüzde, 7 ile 50 arasında değişen bir dizi katman içeren derin öğrenme yaygın olarak kullanılmaktadır. 100 katmandan daha fazla derin ağlar, biraz daha iyi performanslara izin verir, ama hesaplama verimliliğinden azalır. Katman sayısı, derin öğrenmenin "hiper parametrelerinden" yalnızca birini temsil eder. Bir ağın karmaşıklığı, nöronların sayısı, bağlantıları ve ağırlıkları ile de verilir. Her bir ağırlık, öğrenilmesi gereken bir parametreyi temsil eder. Elbette, eğitimin karmaşıklığı bu ağırlıkların sayısına bağlıdır. (Dell Aversana, 2019) Derin öğrenme, bilgileri tıpkı bir insan beyni gibi sınıflandırır ve insan zekasına en çok benzeyen yapay zekanın oluşturulmasına olanak tanır. Verilerin birden fazla özellik seviyesinin öğrenilmesine dayanan bir yapı vardır.

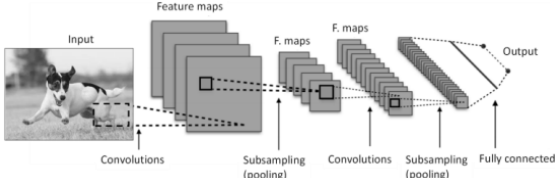
3.4 | Evrişimli Sinir Ağı

Derin öğrenme mimarilerinden biri olan Evrişimli Sinir Ağı (Convolutional Neural Network), çok katmanlı ileri yönlü bir yapay sinir ağıdır. Bu yapay sinir ağının en önemli avantajlarından biri büyük ağ modellerinde daha az sayıda eğitime ve daha az parametreye ihtiyaç duymasındır. Evrişimli Sinir Ağları, görüntü üzerinde nesne tanıma, görüntü analizi ve doğal dil işleme çalışmalarında kullanılmaktadır. Doğal dil ifadelerinden metin analizi, cümlelerin analizi ve sınıflandırılması gibi çalışmalar Evrişimli Sinir Ağlarının bu alandaki uygulanabilirliğini ortaya koymaktadır (Somuncu ve Atasoy, 2022).

Evrişimli sinir ağları, şebekenin yerel bölgelerinde güçlü uzamsal bağımlılıklara sahip olan şebeke yapılı girdilerle çalışmak üzere tasarlanmıştır. Izgara yapılı verilerin en bariz örneği 2 boyutlu bir görüntüdür. Bir görüntüdeki bitişik uzamsal konumlar genellikle tek tek piksellerin benzer renk değerlerine sahip olduğundan, bu tür veriler aynı zamanda uzamsal bağımlılıklar da sergiler. Ek bir boyut, 3 boyutlu bir giriş hacmi oluşturan farklı renkleri yakalar. Bu nedenle, evrişimli bir sinir ağındaki özellikler, uzamsal mesafelere dayalı olarak birbirleri arasında bağımlılıklara sahiptir. Metin, zaman serisi ve diziler gibi diğer sıralı veri biçimleri de, bitişik öğeler arasında çeşitli ilişki türlerine sahip ızgara yapılı verilerin özel durumları olarak kabul edilebilir. Evrişimli sinir ağlarının uygulamalarının büyük çoğunluğu görüntü verilerine odaklanır, ancak bu ağlar her tür zamansal, uzamsal ve uzamsal-zamansal veri için de kullanılabilir. (Aggarwal, 2018)

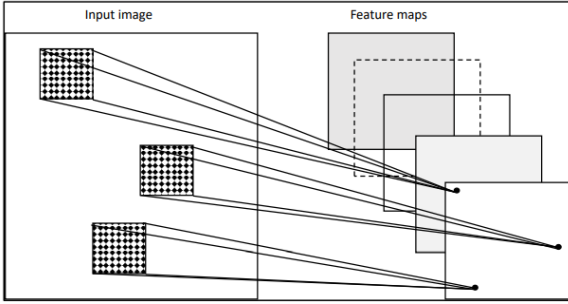
Evrişimli sinir ağlarında, her katmandaki durumlar, bir uzamsal ızgara yapısına göre düzenlenir. Bu uzamsal ilişkiler bir katmandan diğerine aktarılır, çünkü her özellik değeri bir önceki katmandaki küçük bir yerel uzamsal bölgeyi temel alır. Izgara hücreleri arasındaki bu uzamsal ilişkileri sürdürmek önemlidir, çünkü evrişim işlemi ve bir sonraki katmana dönüşüm kritik olarak bu ilişkilere bağlıdır. Evrişimli ağdaki her katman, yüksekliği, genişliği ve derinliği olan 3 boyutlu bir ızgara yapısıdır. Bir evrişimli sinir ağındaki bir katmanın derinliği, ağın kendisinin derinliği ile karıştırılmamalıdır. "Derinlik" sözcüğü (tek bir katman bağlamında kullanıldığında), giriş görüntüsündeki veya görüntüdeki ana renk kanallarının (örneğin mavi, yeşil ve kırmızı) sayısı gibi her katmandaki kanalların sayısını ifade eder. "Derinlik" kelimesinin hem her katmandaki öznitelik haritalarının sayısına hem de katman sayısına atıfta bulunmak için kullanılması, evrişimli ağlarda kullanılan terminolojinin talihsiz bir şekilde aşırı yüklenmesidir. Evrişimli sinir ağı, katmanlarındaki işlemlerin, katmanlar arasında seyrek (ve dikkatlice tasarlanmış) bağlantılarla uzamsal olarak organize edilmesi dışında, geleneksel bir ileri beslemeli sinir ağı gibi çalışır. Evrişimli bir sinir ağında yaygın olarak bulunan üç katman türü evrişim, havuzlama ve ReLU'dur. ReLU aktivasyonu, geleneksel

bir sinir ağından farklı değildir. Ek olarak, son bir katman seti genellikle tamamen bağlantılıdır ve uygulamaya özel bir şekilde bir dizi çıkış düğümüne eşlenir (Aggarwal, 2018). Çeşitli derin öğrenme türleri vardır. Denetimli öğrenmeye yönelik ağlar arasında en çok kullanılanlardan biri, LeCun ve diğerleri tarafından tanıtılmış Evrişimli Sinir Ağıdır (ConvNet). Bunlar, görüntü sınıflandırmanın karmaşık problemlerini çözmek için iyi performanslarından dolayı bilgisayarla görmeye popüler hale gelmiştir. (Dell Aversana, 2019).



Şekil 8: Evrişimli Sinir Ağı (Dell Aversana, 2019)

Sinir ağlarında, her nöron bir önceki katmandaki belirli sayıda konumdan girdi alır. Her nöron bir önceki katmanın her elementinden girdi aldığına buna "tamamen bağlı katman" denir. Evrişimli katman, nöronların yalnızca önceki katmanın sınırlı bir alt alanından girdi almaktadır. Bir nöronun bu girdi alanına "alıcı alan" denir. Tamamen bağlantılı bir katmanda alıcı alan, önceki katmanın tamamından oluşur. Ancak, evrişimli bir katmanda alıcı alan, önceki katmanın tamamından daha küçüktür. Bu kavram, bireysel nöronların kendi görsel uyarlarıyla sınırlı olan gerçek tepkisinin işleyişini taklit etmektedir. (Dell Aversana, 2019) Bu, hesaplama çabalarını azaltmak için etkili bir stratejidir.



Şekil 9: Evrişimli katmana bir örnek (Dell Aversana, 2019)

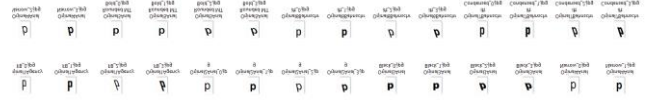
Örnek olarak, elde küçük bir imaj olduğunu varsayalım. YSA'da 100 x 100 boyutunda ilk nöral katmanı varsa, tamamen bağlı katmanlara sahip olmak için ikinci katmandaki her nöron için 10000 ağırlığa ihtiyaç vardır. YSA'nın değişken sayısını ve hesaplama çabalarını azaltmak için daha etkili bir strateji belirlenmelidir. Bunun için bir evrişim işlemi uygulanabilir. Bu durum, serbest parametre sayısını (ağırlıkları belirlenecek) azaltmaya izin verir. Aslında, her evrişimli nöron verileri yalnızca "alıcı alan" için işlemektedir. Bu adım, giriş katmanını bir özellik haritasına bağlayarak gerçekleştirilir. Bu alıcı alanlar, bir giriş görüntüsünün pikselleri üzerinde

kaydırdığımız "örtüşen pencereleri" temsil etmektedir. (Dell Aversana, 2019)

4 | EVRİŞİMLİ SİNİR AĞI İLE ÖRNEK BİR ÇALIŞMA

4.1 | Veri Setinin Hazırlanması

Evrişimli Sinir Ağı Algoritmasının eğitimini yapabilmek için c# dili kullanılarak Windows fontları ile bir örneklem oluşturulmuştur. Bu örneklem 153 font kullanılarak 29 büyük, 29 küçük harf ve 10 rakamın normal, kalın, italik, ve kalın italik olacak şekilde resim olarak kaydedilmesi yolu ile elde edilmiştir. Bu şekilde her bir ifade için 612 (153 * 4) örnek oluşturulmuştur.



Şekil 10: Örnek veri seti

Örnek veri seti Şekil 10'da gösterilmektedir. Örneklem dahil edilen font listesi ise aşağıdaki Tablo 1'de ifade edilmektedir.

Tablo 1: Kullanılan Font Listesi

Agency FB	Franklin Gothic Demi Cond	MS Gothic UI
Arial	Franklin Gothic Heavy	MV Boli
Arial Black	Franklin Gothic Medium	Myanmar Text
Arial Narrow	Franklin Gothic Medium Cond	Nirmala UI
Arial Rounded MT Bold	Gabriola	Nirmala UI Semilight
Bahnschrift	Gadugi	NsimSun
Bahnschrift Condensed	Georgia	OCR A Extended
Bahnschrift Light	Gill Sans MT	Palatino Linotype
Bahnschrift Light Condensed	Gill Sans MT Condensed	Perpetua
Bahnschrift Light SemiCondensed	Gill Sans Ultra Bold	PMingLiU-ExtB
Bahnschrift SemiBold	Gill Sans Ultra Bold Condensed	Poor Richard
Bahnschrift SemiBold Condensed	Gloucester MT Extra Condensed	Rockwell

Bahnschrift SemiBold SemiCondens	Haettenschweiler	Rockwell Condensed
Bahnschrift SemiCondensed	High Tower Text	Rockwell Extra Bold
Bahnschrift SemiLight	Impact	Segoe Print
Bahnschrift SemiLight Condensed	Imprint MT Shadow	Segoe Script
Bahnschrift SemiLight SemiConde	Informal Roman	Segoe UI
Berlin Sans FB	Ink Free	Segoe UI Black
Berlin Sans FB Demi	Leelawadee UI	Segoe UI Emoji
Bernard MT Condensed	Leelawadee UI Semilight	Segoe UI Historic
Bookman Old Style	Lucida Bright	Segoe UI Light
Britannic Bold	Lucida Console	Segoe UI Semibold
Calibri	Lucida Fax	Segoe UI Semilight
Calibri Light	Lucida Handwriting	Segoe UI Symbol
Calisto MT	Lucida Sans	SimSun
Cambria	Lucida Sans Typewriter	SimSun-ExtB
Cambria Math	Lucida Sans Unicode	Sitka Banner
Candara	Maiandra GD	Sitka Display
Candara Light	Malgun Gothic	Sitka Heading
Castellar	Malgun Gothic Semilight	Sitka Small
Century	Microsoft Himalaya	Sitka Subheading
Century Gothic	Microsoft JhengHei	Sitka Text
Century Schoolbook	Microsoft JhengHei Light	Stencil
Colonna MT	Microsoft JhengHei UI	Sylfaen

Comic Sans MS	Microsoft JhengHei UI Light	Tahoma
Consolas	Microsoft New Tai Lue	Tempus Sans ITC
Constantia	Microsoft PhagsPa	Times New Roman
Cooper Black	Microsoft Sans Serif	Trebuchet MS
Copperplate Gothic Bold	Microsoft Tai Le	Tw Cen MT
Copperplate Gothic Light	Microsoft YaHei	Tw Cen MT Condensed
Corbel	Microsoft YaHei Light	Tw Cen MT Condensed Extra Bold
Corbel Light	Microsoft YaHei UI	Verdana
Courier New	Microsoft YaHei UI Light	Wide Latin
Ebrima	Microsoft Yi Baiti	Yandex-UI-Icons-Private
Eras Bold ITC	MingLiU_HKSCS-ExtB	Yu Gothic
Eras Demi ITC	Mistral	Yu Gothic Light
Eras Light ITC	Mongolian Baiti	Yu Gothic Medium
Eras Medium ITC	Monotype Corsiva	Yu Gothic UI
Forte	MS Gothic	Yu Gothic UI Light
Franklin Gothic Book	MS Pgothic	Yu Gothic UI Semibold
Franklin Gothic Demi	MS Reference Sans Serif	Yu Gothic UI Semilight

Örneklem için dahil edilmeyen veri seti örnekleri Şekil 11'de gösterilmektedir.



Şekil 11: Kullanılmayan veri seti örneği

Örneklem için dahil edilmeyen fontlar aşağıdaki tabloda yer almaktadır. Bu fontlar farklı gösterim şekilleri olduğu için dahil edilememiştir..

Tablo 2: Kullanılmayan Font Listesi

Algerian	French Script MT	Niagara Solid
Baskerville Old Face	Garamond	Old English Text MT
Bauhaus 93	Gigi	Onyx
Bell MT	Gill Sans MT Ext Condensed Bold	Palace Script MT
Blackadder ITC	Goudy Old Style	Papyrus
Bodoni MT	Goudy Stout	Parchment
Bodoni MT Black	Harlow Solid Italic	Perpetua Titling MT
Bodoni MT Condensed	Harrington	Playbill
Bodoni MT Poster Compressed	HoloLens MDL2 Assets	Pristina
Book Antiqua	Javanese Text	Rage Italic
Bookshelf Symbol 7	Jokerman	Ravie
Bradley Hand ITC	Juice ITC	Script MT Bold
Broadway	Kristen ITC	Segoe MDL2 Assets
Brush Script MT	Kunstler Script	Showcard Gothic
Californian FB	Lucida Calligraphy	Snap ITC
Centaur	Magneto	Symbol
Chiller	Marlett	TeamViewer15
Curlz MT	Matura MT Script Capitals	Viner Hand ITC
Edwardian Script ITC	MingLiU-ExtB	Vivaldi
Elephant	Modern No. 20	Vladimir Script
Engravers MT	MS Outlook	Webdings
Felix Titling	MS Reference Specialty	Wingdings
Footlight MT Light	MT Extra	Wingdings 2

Freestyle Script	Niagara Engraved	Wingdings 3
------------------	------------------	-------------

Daha sonra bu örnekler python da geliştirilen kod ile, sağ ve soldaki boşluk silinerek, sonrasında da üst ve alttaki boşluk silinerek üç katına çıkarılmıştır. Örneklem toplamda 124.848 elemandan 25x40 boyutunda oluşturulmuştur.



Şekil 12: Artırılmış veriseti örneği

4.2 | Evrişimli Sinir Ağı Eğitim İşlemi

Bu aşamada tüm örnek resimlerin sayısallaştırılma işlemi gerçekleştirilmiştir. Bununla ilgili kodlamalar ve işlem özetleri ilerleyen şekillerde ifade edilmektedir.

```
#digitler = ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9']
for digit in digitler:
    label = digit
    training_directory = os.path.join(train_dir, str(label) + '/')
    if not os.path.exists(training_directory):
        os.mkdir(training_directory)
    if (filename.endswith('.png') or filename.endswith('.jpg')):
        training_digit = warpAffine(pil_img, loadad(training_directory + filename))
        features_list.append(training_digit)
        features_label_text.append(label)
        features_label.append(str(kategori))
    kategori = kategori + 1

#digitler = ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9']
for digit in digitler:
    label = digit
    training_directory = os.path.join(train_dir, str(label) + '/')
    if not os.path.exists(training_directory):
        os.mkdir(training_directory)
    if (filename.endswith('.png') or filename.endswith('.jpg')):
        training_digit = warpAffine(pil_img, loadad(training_directory + filename))
        features_list.append(training_digit)
        features_label_text.append(label)
        features_label.append(str(kategori))
    kategori = kategori + 1
```

Şekil 13: Sayısallaştırma Kodu

Normalleştirme işlemi için değerler 255 e bölünmüştür. Evrişim katmanında kernel size (5*5) olarak 80 filtresi kullanılmıştır. 3 adet hidden layer eklenmiş ve bu layer üzerinde aktivasyon fonksiyonu olarak “relu” tanımlanmıştır. Son katman için aktivasyon fonksiyonu olarak “softmax” tanımı yapılmıştır. Loss fonksiyonu olarak “crossentropy” ve epoch değeri olarak da 10 belirlenip eğitim işlemi gerçekleştirilmiştir.

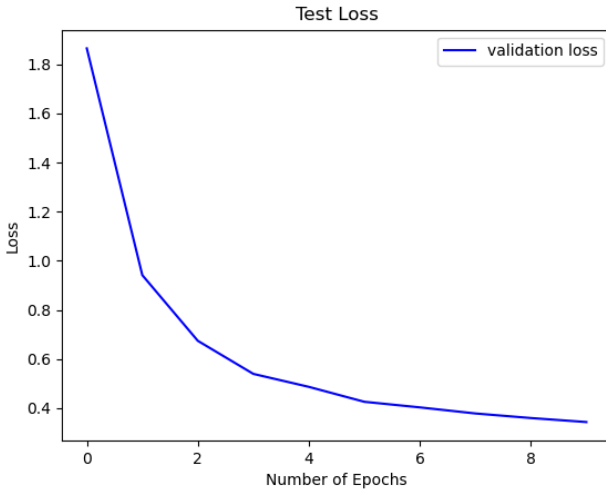
```
model = Sequential()
model.add(Conv2D(48, (5, 5), activation='relu', input_shape=(48, 28, 4)))
model.add(MaxPooling2D(pool_size=(2, 2)))
#model.add(Conv2D(88, (5, 5), activation='relu'))
#model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Flatten())
model.add(Dense(1000, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(500, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(250, activation='relu'))
model.add(Dense(kategori, activation='softmax'))
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
history = model.fit(x_train, y_train_one_hot, batch_size=256, epochs=10, validation_split=0.2)
```

Şekil 14: CNN yapısı kod örneği

Model: "sequential_1"

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 36, 21, 40)	4040
max_pooling2d_1 (MaxPooling2)	(None, 18, 10, 40)	0
flatten_1 (Flatten)	(None, 7200)	0
dense_1 (Dense)	(None, 1000)	7201000
dropout_1 (Dropout)	(None, 1000)	0
dense_2 (Dense)	(None, 500)	500500
dropout_2 (Dropout)	(None, 500)	0
dense_3 (Dense)	(None, 250)	125250
dense_4 (Dense)	(None, 68)	17068
Total params: 7,847,858		
Trainable params: 7,847,858		
Non-trainable params: 0		

Şekil 15: CNN Model Özeti



Şekil 16: Epoch – Loss Değeri grafiği

```

Epoch 1/10 [=====] - 78e 982ba/step - loss: 3.3141 - accuracy: 0.1664 - val_loss: 1.8644 - val_accuracy: 0.4755
Epoch 2/10 [=====] - 60e 1aa/step - loss: 1.5514 - accuracy: 0.5327 - val_loss: 0.9413 - val_accuracy: 0.7235
Epoch 3/10 [=====] - 79e 998aa/step - loss: 1.0304 - accuracy: 0.6790 - val_loss: 0.6739 - val_accuracy: 0.7973
Epoch 4/10 [=====] - 79e 1aa/step - loss: 0.8253 - accuracy: 0.7385 - val_loss: 0.5393 - val_accuracy: 0.8354
Epoch 5/10 [=====] - 75e 944aa/step - loss: 0.7066 - accuracy: 0.7730 - val_loss: 0.4865 - val_accuracy: 0.8469
Epoch 6/10 [=====] - 76e 960aa/step - loss: 0.6344 - accuracy: 0.7929 - val_loss: 0.4256 - val_accuracy: 0.8656
Epoch 7/10 [=====] - 87e 1aa/step - loss: 0.5856 - accuracy: 0.8079 - val_loss: 0.4029 - val_accuracy: 0.8707
Epoch 8/10 [=====] - 85e 1aa/step - loss: 0.5407 - accuracy: 0.8213 - val_loss: 0.3701 - val_accuracy: 0.8768
Epoch 9/10 [=====] - 75e 943aa/step - loss: 0.5100 - accuracy: 0.8263 - val_loss: 0.3595 - val_accuracy: 0.8797
Epoch 10/10 [=====] - 74e 930aa/step - loss: 0.4840 - accuracy: 0.8368 - val_loss: 0.3432 - val_accuracy: 0.8852

```

Şekil 17: Eğitim değerleri

4.3 | Resim işleme ve karakter belirleme

İlk olarak karakter çıkarımı yapılacak resim sayısalştırma işlemi yapılmıştır.

TEST METİN 12

DENEME 345

Şekil 18: Test yapılacak resim

Sonrasında resim gri formata çevrilip belirli bir eşik değere (127) göre siyah beyaz formata çevirilmiştir. Sonrasında (2*2) kernel değeri ile erosion ve dilation işlemleri uygulanmıştır.

TEST METİN 12

DENEME 345

Şekil 19: Ön işleme sonucunda oluşan örnek resim

Satırların tespit edilmesi, pixel bazlı temsil edilen görüntünün dikey olarak satırda ilk tespit edilen anlamlı yani çevresinde belirli bir eşik değerine göre siyah bulunan siyah nokta bulunur. Satır başlangıcının dikeyde tespitinden sonra bitişinin de dikeyde tespiti için dikeyde tespit edilen anlamlı yani çevresinde belirli bir eşik değerine göre beyaz bulunan beyaz nokta bulunur. Satır bitişi içinde dikeyde tespit edilen anlamlı yani çevresinde belirli bir eşik değerine göre beyaz bulunan beyaz nokta bulunur.

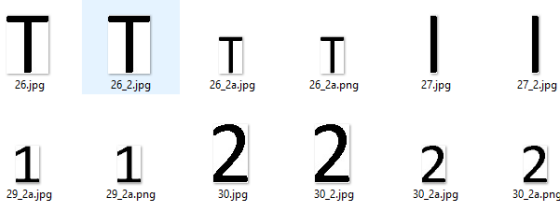
TEST METİN 12

Şekil 20: Satır belirleme sonucunda tespit edilen örnek satır

Kelimelerin tespiti, satır sınırları içerisinde pixel bazlı temsil edilen görüntünün dikey olarak satırda ilk tespit edilen anlamlı yani çevresinde belirli bir eşik değerine göre siyah bulunan siyah nokta bulunur. Kelime başlangıcının dikeyde tespitinden sonra bitişinin de dikeyde tespiti için dikeyde tespit edilen anlamlı yani çevresinde belirli bir eşik değerine göre beyaz bulunan beyaz nokta bulunur. Kelime bitişi içinde dikeyde tespit edilen anlamlı yani çevresinde belirli bir eşik değerine göre beyaz bulunan beyaz nokta bulunur.

Harflerin tespiti, Kelime sınırları içerisinde pixel bazlı temsil edilen görüntünün dikey olarak satırda ilk tespit edilen anlamlı yani çevresinde belirli bir eşik değerine göre siyah bulunan siyah nokta bulunur. Harf başlangıcının dikeyde tespitinden sonra bitişinin de dikeyde tespiti için dikeyde tespit edilen anlamlı yani çevresinde belirli bir eşik değerine göre beyaz bulunan

beyaz nokta bulunur. Kelime bitişi içinde dikeyde tespit edilen anlamlı yani çevresinde belirli bir eşik değerine göre beyaz bulunan beyaz nokta bulunur.



Şekil 13: Tespit edilen harf örnekleri

Bu işlemler sonucunda her harf için eğitimi yapılmış algoritmaya tahminleme işlemi yapılır.

I
TEST METİN 1z
DENEME 345

Şekil 14: Karakter tanıma sonucu

5 | SONUÇ

Gerçekleştirilen çalışma ile büyük oranda karakterler sorunsuz tespit edilebilmektedir. Çalışma ilk başladığında sadece windows fontları ile elde edilen veriseti kullanılmış ve sonuçlar elde edilmiştir. Bu deneme sonucunda resimde var olan 21 karakterden yaklaşık olarak yarısı tanımlanabilmiştir. Veriseti python kodu ile artırılarak 3 katına çıkarıldıktan sonra aynı resim ile deneme yapıldığında sadece 2 rakamının tanıma işlemi yapılamamıştır. Bunun nedeni ise rakamın "z" harfi olarak tanınmasıdır. Veriseti daha da artırılarak tanıma oranı artırılabilir.

Satır belirleme işleminde "İ" harfinin noktası da bir satır olarak algılanmıştır. Türkçe karakterlerde üstte ve altta yer alan noktalama işaretlerinin de daha düzgün algılanabilmesi için satır ve kelime belirleme algoritmasında iyileştirme yapılmalıdır.

KAYNAKÇA:

Abiodun, O. I., Jantan, A., Omolara, A. E., Dada, K. V., Mohamed, N. A., & Arshad, H. (2018). State-of-the-art in artificial neural network applications: A survey. *Heliyon*, 4(11), e00938.

Abraham, A. (2005). Artificial neural networks. *Handbook of measuring system design*.

Aggarwal, C. C. (2018). *Neural networks and deep learning*. Springer, 10(978), 3.

Alpaslan, F., Eğrioglu, E., Aladağ, Ç., İltter, D. ve Dalar, A. (2013). Tek Çarpımsal Sinir Hücreli Yapay Sinir Ağı Modelinin Eğitimi İçin ABC ve BP Yöntemlerinin Karşılaştırılması. *Eskişehir Technical University Journal of Science and Technology A - Applied Sciences and Engineering*. 14(3). 315-328.

Arı A. ve Berberler, M. E. (2017). Yapay sinir ağları ile tahmin ve sınıflandırma problemlerinin çözümü için arayüz tasarımı. *Acta Infologica*, 1(2), 55-73.

Atalı, G., Özkan, S. S., & Karayel, D. (2016). Image damage analysis with morphological image processing technique using artificial neural networks. *Academic Platform-Journal of Engineering and Science*, 4(1)

Ataseven, B. (2013). Yapay sinir ağları ile öngörü modellenmesi. *Öneri Dergisi*, 10(39), 101-115.

Bektaş, B. (2014). RFID ve XBEE Tabanlı Depo Yönetim Sistemi Tasarımı ve Gerçekleştirilmesi. İstanbul: Marmara Üniversitesi. Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi.

Bektaş, B., Babur, S., Turhal, U., & Köse, E. (2016). Makine öğrenmesi yardımıyla optik karakter tanıma sistemi. *5. Uluslar Arası Matbaa Teknolojileri Sempozyumu*, İstanbul.

Dell'Aversana, Paolo (2019) Artificial neural networks and deep learning. a simple overview. *Manager*. <https://doi.org/10.5121/ijsc.2012.3203> Google Scholar

Dongare, A. D., Kharde, R. R., & Kachare, A. D. (2012). Introduction to artificial neural network. *International Journal of Engineering and Innovative Technology (IJEIT)*, 2(1), 189-194.

Erdoğan, A. A., & Tümer, A. E. (2021). Deep Learning Method for Handwriting Recognition. *MANAS Journal of Engineering*, 9(1), 85-92.)

Karpathy, A. 2015. "Neural Networks Part 1: Setting Up the Architecture." Notes for CS231n Convolutional Neural Networks for Visual Recognition, Stanford University. <http://cs231n.github.io/neural-networks-1/>

Kır, B., Öz, C., & Gülbağ, A. (2001). Yapay Sinir Ağlarında Negative Correlation Learning Metodu Kullanarak Optik Karakter Tanıma. *Elektrik-Elektronik Bilgisayar Sempozyumu (FEEB 2011)*, Elazığ, 105-109.

- Krenker, A., Bešter, J., & Kos, A. (2011). Introduction to the artificial neural networks. *Artificial Neural Networks: Methodological Advances and Biomedical Applications*. InTech, 1-18.
- Koyun, A., & Afşin, E. (2017). Derin öğrenme ile iki boyutlu optik karakter tanıma. *Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi*, 10(1), 11-14.
- Maind, S. B., & Wankar, P. (2014). Research paper on basic of artificial neural network. *International Journal on Recent and Innovation Trends in Computing and Communication*, 2(1), 96-100.
- Musayev, E. (2004). Bilgisayar destekli karakter tanıma sistemi tasarımı. İstanbul Üniversitesi Fen Bilimleri Enstitüsü.
- Mithe, R., Indalkar, S., & Divekar, N. (2013). Optical character recognition. *International journal of recent technology and engineering (IJRTE)*, 2(1), 72-75.
- Ovatman, T. (2005). A Real-Time Optical Character Recognition System. İstanbul: İstanbul Technical University, Institute of Science and Technology Master Thesis.
- Shah, P., Karamchandani, S., Nadkar, T., Gulechha, N., Koli, K., & Lad, K. (2009, November). OCR-based chassis-number recognition using artificial neural networks. In *2009 IEEE International Conference on Vehicular Electronics and Safety (ICVES)* (pp. 31-34). IEEE.
- Somuncu, E., & ATASOY, N. A. (2022). Evrişimli tekrarlayan sinir ağı ile metin görüntüleri üzerinde karakter tanıma uygulaması gerçekleştirilmesi. *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*, 37(1), 17-28.
- Srihari, S. N., Shekhawat, A., & Lam, S. W. (2003). Optical character recognition (OCR). In *Encyclopedia of Computer Science* (pp. 1326-1333).
- 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.
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., & Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1-9).
- Şekerci, M., & Kandemir, R. (2009). Birleşik ve Eğik Türkçe El Yazısı Tanımada K-Nn Sınıflama Yöntemi Ve Sözlük Kullanımı. *Trakya Üniversitesi Fen Bilimleri Dergisi*, 10(1), 97-102.
- Şengür, A., ve Türkoğlu, İ. (2004). Değişmez momentlerle Türkçe karakter tanıma. *Fırat Üniversitesi Doğu Araştırmaları Dergisi*, 2(2), 114-117.
- Tilki, S. (2020). İngilizcede çember dolgu oranı yöntemi ile optik karakter tanıma (Master's thesis, İstanbul Sabahattin Zaim Üniversitesi)
- Uzun, E. & Çağıltay, K. (2012). Çevrimiçi El Yazısı Tanıma Sistemi Olan Graffiti'nin Kullanılabilirlik Açısından Değerlendirilmesi. *SDU International Journal Of Technological Science*, 4(1), 46-57.
- Verma, B., Blumenstein, M., & Kulkarni, S. (1998). Recent achievements in off-line handwriting recognition systems. In *Proceedings of the International Conference on Computational Intelligence and Multimedia Applications* (pp. 27-33).
- Yıldırım, Elif. Yapay Sinir Ağı (Artificial Neural Network) Nedir? Available from: <https://www.veribilimiokulu.com/blog/yapay-sinir-agiartificial-neural-network-nedir/> [Accessed 06th Februray 2021].