**T.C.**

**ANKARA ÜNİVERSİTESİ**

**MÜHENDİSLİK FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**(BLM3551)Yapay Zeka Dersi**

**Makine Öğrenmesi ile El Yazısı Tanıma Proje Raporu**

**Öznur Kandakoğlu - 19290250**

**Danışman**

**Doç. Dr. Mehmet Serdar Güzel**

**2021** – **Ankara**

İÇİNDEKİLER

[ŞEKİLLER TABLOSU 3](#_Toc61480258)

[ÖZET 4](#_Toc61480259)

[ABSTARCT 5](#_Toc61480260)

[1. GİRİŞ 6](#_Toc61480261)

[1.1 PROJE TANIMI 6](#_Toc61480262)

[1.2 PROJENİN AMACI VE ÖNEMİ 7](#_Toc61480266)

[2. YÖNTEM VE TEKNİKLER 8](#_Toc61480267)

[2.1 PROJENİN PLANLAMASI 8](#_Toc61480268)

[2.2 PROJENİN UYGULANMASI 8](#_Toc61480269)

[2.2.1 PROJE VERİ KÜMESİNİN OLUŞTURULMASI 9](#_Toc61480270)

[2.2.2 PROJE GÖRÜNTÜLERİNİN TEST EDİLMESİ 9](#_Toc61480271)

[3. BULGULAR 10](#_Toc61480272)

[4. SONUÇ VE TARTIŞMALAR 12](#_Toc61480273)

[5. ÖNERİLER 14](#_Toc61480274)

[6. EKLER 14](#_Toc61480275)

[KAYNAKLAR 15](#_Toc61480276)

# ŞEKİLLER TABLOSU

[Şekil 1 EMNSIT Veri Kümesi Rakam Örnekleri 7](#_Toc61480347)

[Şekil 2 Projenin Genel Çalışma Şeması 8](#_Toc61480348)

[Şekil 3 İlk Görüntü 9](#_Toc61480349)

[Şekil 4 Son Görüntü 9](#_Toc61480350)

[Şekil 5 Doğruluk değeri 10](#_Toc61480351)

[Şekil 6 Kayıp Değerler 11](#_Toc61480352)

[Şekil 7 Program çıktısı 11](#_Toc61480353)

[Şekil 8 Başarısız çıktı örneği 12](#_Toc61480354)

[Şekil 9 Başarılı çıktı örneği 12](#_Toc61480355)

[Şekil 10 Modelleme sonucu 13](#_Toc61480356)

# ÖZET

Bu proje Ankara Üniversitesi Bilgisayar Mühendisliği 2. Sınıf öğrencisi Öznur Kandakoğlu tarafından Yapay Zeka dersi final projesi olarak geliştirilen yapay sinir ağı tabanlı karakter tanıma uygulamasını içeren sistemin; tanımı, içeriği, amaçları, başlatılma nedenleri ve sonuçlarını içermektedir.

Yapay Sinir Ağları (YSA), giriş verilerinin yetersiz olduğu, mevcut verilerden hareketle bilinmeyen ilişkilerin ortaya çıkarılması ve algoritması veya kuralları tam olarak bilinmeyen durumlar için geliştirilmiş bir bilgi işleme sistemidir.[1] Bu çalışmada, yapay sinir ağlarının önemli bir uygulaması olan karakter tanıma işlemi ele alınmıştır.

Sistem, [Kaggle'daki EMNIST Veri Kümesi](https://www.kaggle.com/crawford/emnist) kullanılarak önceden eğitilmiş birden çok makine öğrenimi modelini birleştirerek el yazısını tanıyan ve metne dönüştüren React/ JavaScript ve Python / Django dilleri yardımıyla oluşturulmuş web uygulamasıdır. Tasarlanan sistemde, el yazısı tanıma sistemlerindeki amaçların çoğu kullanılmıştır. Bunlar dokümanın taranması, görüntüdeki gürültünün temizlenmesi, satırların ayıklanması, kelimelerin ayıklanması, kelime eğiminin düzeltilmesi, karakterlerin ayrıştırılması, tanıması ve doğrulaması aşamalarıdır. Tanıma sistemi büyük harf, küçük harf ve rakam olmak üzere üç tipteki karakteri ayrı ayrı tanıyacak şekilde tasarlanmıştır.

Anahtar Kelimeler: El yazısı tanıma, yapay sinir ağları, görüntü işleme, desen tanıma

# ABSTARCT

This project was developed by Öznur Kandakoğlu, a second year student of Computer Engineering at Ankara University as the final project of the Artificial Intelligence course, which includes artificial neural network based character recognition application that includes its definition, content, aims, reasons for initiation and results.

Artificial Neural Networks are input data systems which are developed for the cases in which are starting from the available data to disclose the unknown relationships and the algorithm or data processing system in which rules are not exactly known. In this study, the pattern recognition procedure which is an important application of neural network was considered.

A full stack React/JavaScript and Python/Django web application that recognizes handwriting and converts it into text, by incorporating multiple machine learning models that were pre-trained using the [EMNIST Dataset](https://www.kaggle.com/crawford/emnist) on Kaggle. The most of the aims used in handwritten recognition systems are used in this system. These aims are scanning document, clerance the image, appointing the satires of text, appointing words, smoothing the slant of the word, finding the characters,character recognitionand verification. Recognition system consists of three recognition modules which are designed for recognize uppercase characters, lowercase characters and numbers.

Keywords: Recognation of handwriting, artificial neural networks, image processing, pattern recognition.

# GİRİŞ

Yapay sinir ağları, insan beyninin özelliklerinden olan öğrenme yolu ile yeni bilgiler türetebilme, yeni bilgiler oluşturabilme ve keşfedebilme gibi yetenekleri herhangi bir yardım almadan otomatik olarak gerçekleştirmek amacı ile geliştirilen bilgisayar sistemleridir.[2] Yapay sinir ağları, olayların örneklerine bakmakta onlardan ilgili olaya hakkında genellemeler yapmakta, bilgiler toplamakta ve daha sonra hiç görmediği örnekler ile karşılaşınca öğrendiği bilgileri kullanarak o örnekler hakkında karar verebilmektedir.

Yapay sinir ağı modeli geliştirilmiş ve sayısız uygulama ortaya çıkmıştır. Bu çalışmalar aslında insan beyninin nasıl çalıştığı ve öğrenme olayını nasıl gerçekleştirdiğini merak etme sonucunda ortaya çıkmıştır. Yapılan çalışmalar ile bilgisayarların öğrenebildikleri ve başarılı sonuçlar ürettikleri görülmektedir [3]. Bu doğrultuda karakter tanıma, uzun yıllar boyunca üzerinde çokça çalışma yapılmış bir araştırma alanıdır. Karakter tanımanın amacı, insanoğlunun okuma özelliğini, insana göre çok daha hızlı çalışan bilgisayara taklit ettirmek olarak belirtilebilir [4]. Bu projede ise karakter tanıma probleminin özelleşmiş bir formu gerçekleştirilecektir. Karaktere ek olarak kelime, cümle ve rakam tanıma da gerçekleştirilecektir.

## PROJE TANIMI

Bu proje, İngilizce karakterlerin olduğu bir veri seti üzerinde (Şekil 1) çalışan bir karakter tanıma sisteminin tasarım ve implementasyonu gerçekleştirilmesini içerir. EMNSIT veri kümesinin kullanımın başlıca sebebi diğer veri kümelerine oranla fazla sayıda veriye sahip olması ve bu bağlamda elde edeceğimiz doğruluk oranını arttırmasıdır. Karakterlerinin tanımlanması için yapay sinir ağları kullanılacaktır. Yapay sinir ağı için hazır bir araç kullanılmayacak, yapay sinir ağı implementasyonu Python programlama dili kullanılarak yapılacaktır.

ok içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil 1 EMNSIT Veri Kümesi Rakam Örnekleri



## PROJENİN AMACI VE ÖNEMİ

El yazısı tanıma; el ile yazılan harf, rakam ve sembollerin bilgisayar sistemleri tarafından tanınmasıdır. İnsanlar için oldukça kolay olmasına rağmen, bir zemin üzerindeki çizgi ve eğrilerin otomatik olarak harf ve rakamlar, daha ileri aşamada da anlamlı sözcükler olarak algılanması oldukça zor bir problemdir. Fakat şu anki teknoloji el yazısı tanıma konusunda henüz kısıtlı bir düzeydedir ve hala tam olarak çözülmüş bir problem değildir. El yazısı tanımadaki zorluk, çok fazla sayıda değişik yazı karakteri olması ve kişiden kişiye farklılıklar göstermesinin yanında harflerin birbirine bağlı yazılmasından kaynaklanmaktadır.[5]

El yazısı tanıma yöntemleri çalışır hale geldiğinde, bu bizleri klavye kullanmaktan kurtaracak ve çok daha doğal bir şekilde yazmamızı ve çizmemizi sağlayacaktır. Bu tür sistemlere örnek olarak, elektronik ajanda ve diğer tablet bilgisayarlar gittikçe yaygınlaşmaya başlamıştır.

# YÖNTEM VE TEKNİKLER

Bu çalışmada el yazısı karakterler; modellenmek ve tanıma oranını arttırmak için çeşitli yöntemlerle işleme sokulmuşlardır.

## PROJENİN PLANLAMASI

Proje doğrultusunda, Python ve Node.js programlama dilleri kullanılarak bir web uygulaması geliştirilecektir. Kullanıcıdan alınan girdi aşağıdaki adımlar takip edilerek işlenecek.

Şekil 2 Projenin Genel Çalışma Şeması

## PROJENİN UYGULANMASI

Bu web uygulamasındaki önceden eğitilmiş beş model, el yazısını tahmin etmekten sorumludur. Öncelikle veri kümeleri hazırlanacak. Bu veri kümelerini Makine Öğrenmesi algoritması için hazırlayacağız. Hangi parametrelerin en iyi çalıştığını bulmak için modellerin birden çok permütasyonunu oluşturacağız.

### **2.2.1 PROJE VERİ KÜMESİNİN OLUŞTURULMASI**

Sinir ağı modelini geliştirmeden önce, modelin öğrenmesi için verilere ihtiyaç vardır. Bu kapsamda seçilen veri kümesi, Kaggle'daki EMNIST Veri Kümesidir. EMNSIT, en uygulanabilir olarak tanımlandığı için seçilmiştir. Başlamak için, seçilen veri setiyle birlikte gerekli tüm içe aktarmalar başlatılmalıdır.

### **2.2.2 PROJE GÖRÜNTÜLERİNİN TEST EDİLMESİ**

Proje görüntülerini adım adım inceleyelim. Şekil 3’de gösterildiği gibi, girdi ilk olarak train veri kümesinden rastgele görüntüyü çizmek için kullandığımız matplotlib yardımıyla alınır. Doğru girdi alınsa da görüntü ters olabilir. Bu yüzden Şekil 4’deki gibi doğru görüntüyü elde etmek için görüntünün yeri numpy yardımıyla değiştirilir.

|  |  |
| --- | --- |
| metin, küçük resim içeren bir resim  Açıklama otomatik olarak oluşturuldu  Şekil 3 İlk Görüntü | Şekil 4 Son Görüntü |

Bu işlemler her bir görüntü üzerinde gerçekleştirilecektir. Özetleyecek olursak her test görüntüsü için gerekli işlemler:

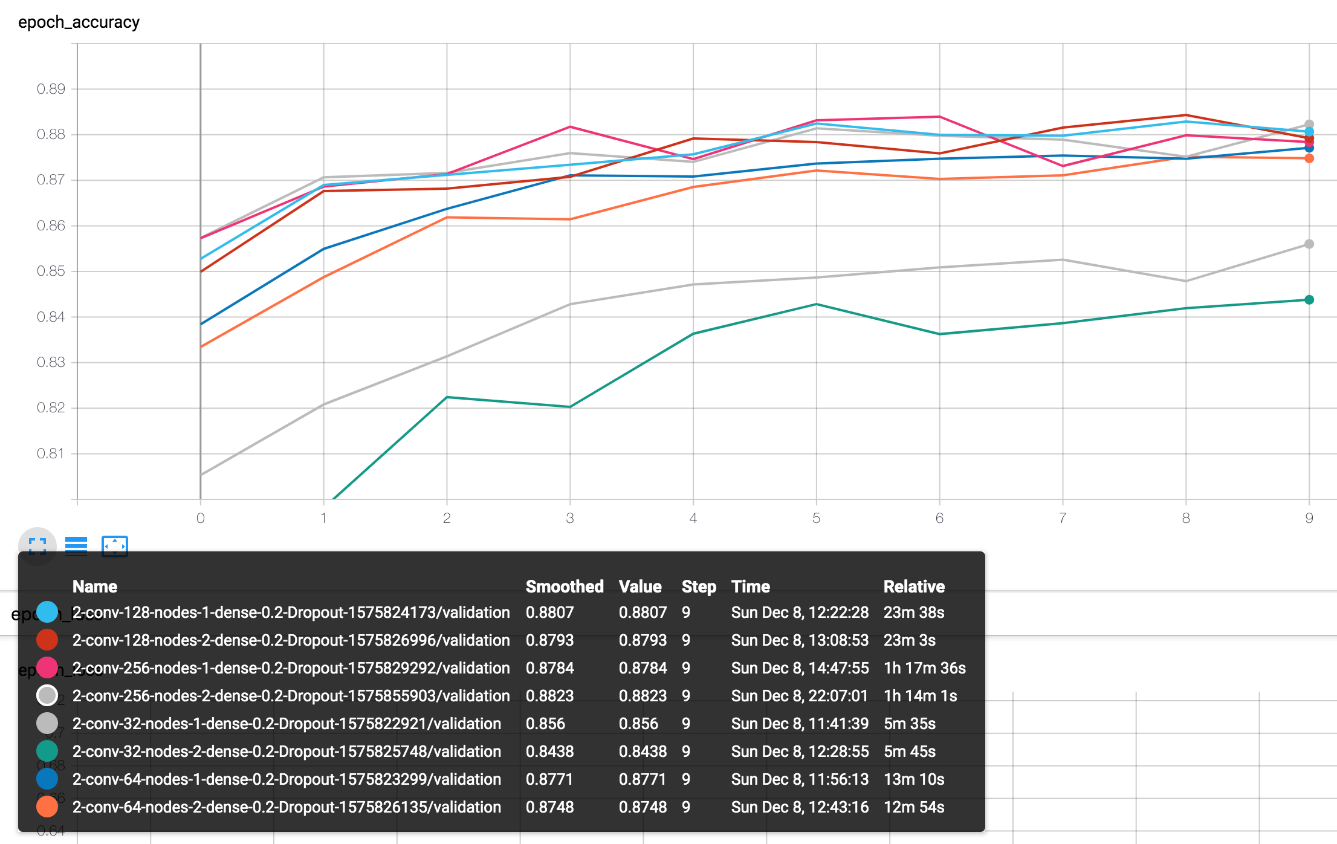
1. Train veri kümesinden rastgele görüntü seçilir.
2. Veriler 28x28 şeklinde yeniden şekillendirilir. Tamamlandığında piksel verileri normalleştirilir böylece piksel değerleri 0 ile 255 yerine 0 ile 1 arasında değişir. Bu normalleştirme, sinir ağının öğrenmesini kolaylaştırmaya yardımcı olur.
3. Veriler doğru konumuna aktarılır.

# BULGULAR

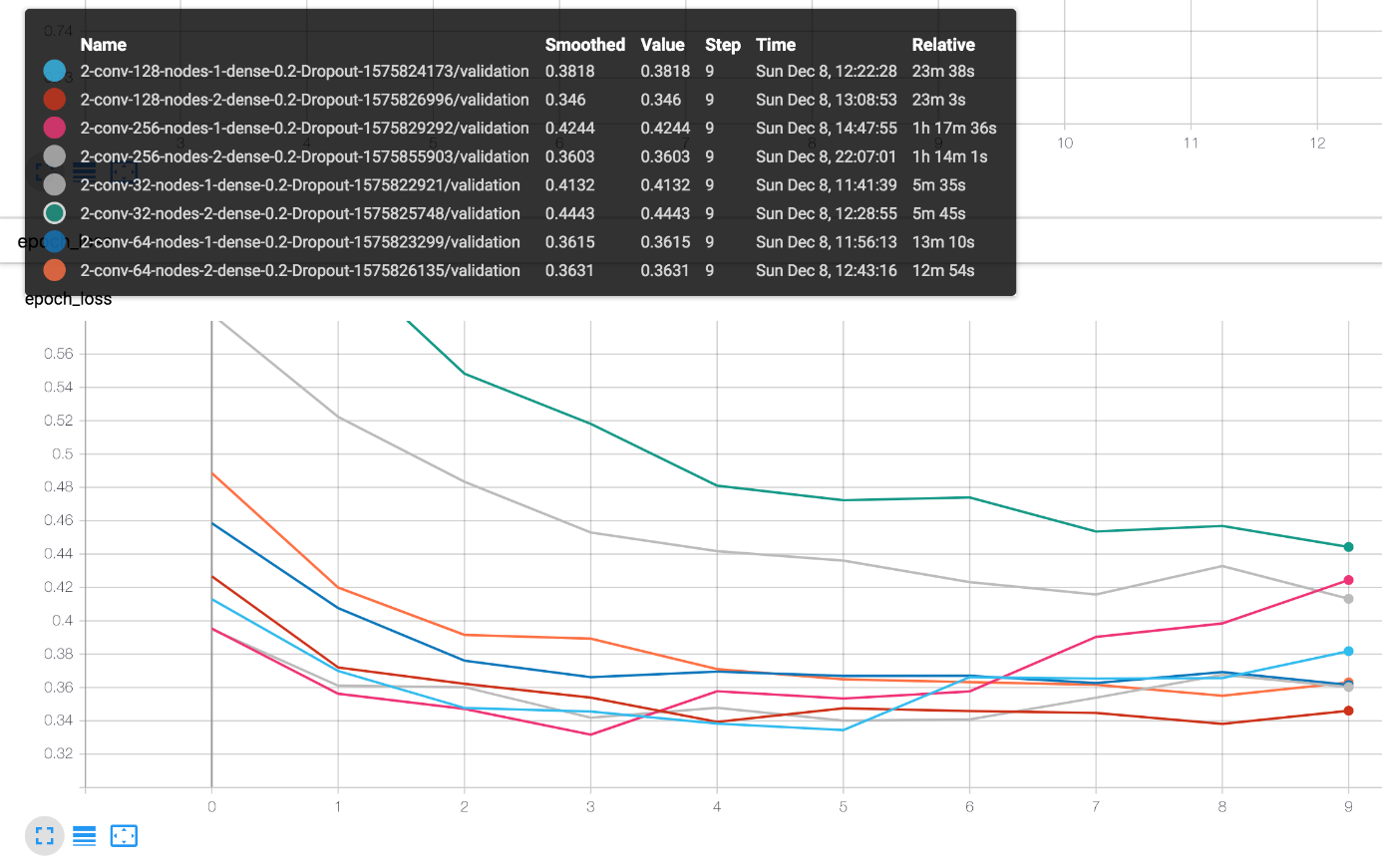
Projedeki önemli unsurlardan biri de maksimum düzeyde doğruluk oranı elde etmek ve minimum düzeyde değer kaybetmektir. Bu amaç doğrultusunda farklı kombinasyonlar denenmiştir. Farklı katman sayıları ve yoğunlukları, farklı nöron sayıları gibi birden çok durum değerlendirmeye alınmıştır. Bu değerlendirmeler sonucunda,

* 2 evrişimli katman
* 2 yoğun katman
* 0.2 bırakma
* Katman başına 128 düğüm

şeklinde bir model oluşturulmuştur. Oluşturduğumuz modelin son kombinasyondaki doğruluk ve kayıp değerleri Şekil 5 ve şekil 6’daki grafiklerde gösterilmiştir. Gösterilen grafikler, bırakma uygulanırken 2 evrişimli katmana sahip tüm modelleri temsil etmektedir.



Şekil 5 Doğruluk değeri



Şekil 6 Kayıp Değerler

Oluşturduğumuz 2 dönüşüm - 2 yoğun - 0,2 bırakma - 128 düğüm modeli sonucunda %87,42 doğruluk oranı elde edilmiş oldu. Doğruluk ve kayıp açısından, 32 nöronlu modeller grubun geri kalanına göre daha düşük performans vermiştir. Katman başına 64, 128 veya 256 nöron geçerli seçeneklerdir.

Programda “Hello World” cümlesi girdi olarak alındığında maksimum doğruluk oranıyla şekil 7’deki görüntü elde edilmiştir.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil 7 Program çıktısı

# SONUÇ VE TARTIŞMALAR

Bu raporda projenin ana fonksiyonu olan sözcük tanıma, rakam tanıma daha ileri seviyede de cümle tanıma işlemleri, veri kümesinin oluşturulması ve projenin nasıl test edildiği genel hatlarıyla açıklanmıştır. Yapay sinir ağları genelleme yeteneğine sahip olması, hatalara karşı toleranslı olması ve esneklikleri sayesinde eksik verilerle bile doğru çalışma yeteneğine sahip bir yöntemdir [6]. Bu sebeple, projede bu yönteme başvurulmuştur fakat yine de yapılan testler sırasında, her karakter için başarı oranının aynı olmadığı gözlemlenmiştir. Veri seti içinde benzeri bulunan harflerde başarın daha düşük olduğu gözlemlenmiştir. Şekil 8’de N harfinin W harfi olarak yanlış tanımlanması görülmektedir. Şekil 9’da ise başarılı bir tanımlanma örneği gösterilmiştir.



Şekil 8 Başarısız çıktı örneği



Şekil 9 Başarılı çıktı örneği

Bazı modeller, tamamen dikey bir çizgiyi “|”, "N" olarak tahmin etti. Bazıları harf olan "a" nın "Q" olduğunu tahmin etti ya da "1" veya "9" tahmininde bulunamadı. Bu problemi çözmek ve daha yüksek doğruluk elde edebilmek için projede web uygulaması içinde 2 dönüşüm - 2 yoğun - 0,2 bırakma - 128 düğümden oluşan 5 model oluşturuldu. Bu modeller yardımıyla “jüri metodu” kullanıldı. Jüri metodu, oluşturulan 5 modelin kararını içeren bir yapıdır. Her bir karakter için 5 farklı modelden sonuç alınmaktadır ve bu 5 sonuç içerisinden en çok aynı olan yanıt kullanıcıya çıktı olarak sunulmuştur. Şekil 7’de çıktısını gösterilen “Hello World” cümlesindeki “World” kelimesi tahmin edilirken nasıl bir karar mekanizması uygulandığı Şekil 10’da gösterilmiştir.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil 10 Modelleme sonucu

“W” harfi için tüm modeller W tahmininde bulundu.

“o” harfi için m1, m2, m3, m4 modelleri o tahmininde bulunurken m5, 0 tahmininde bulundu.

“r” harfi için m1, m4, m5 modelleri r tahmininde bulunurken m2, m3 T tahmininde bulundu.

“l” harfi için m1, m5 modelleri l tahmininde bulunurken m3, m4 I tahmininde bulundu, m2 1 tahmininde bulundu.

“d” harfi için tüm modeller d tahmininde bulundu.

# ÖNERİLER

Geliştirdiğim bu proje kapsamında alabileceğim maksimum doğruluk değerini almaya çalıştım ki bence bu proje için en önemli detaylardan biriydi. Yaptığım literatür araştırmasında da bu hususa dikkat edildiğini fakat hala kesin bir çözüm bulunamadığını gördüm. Eğer bu problem çözülebilirse günlük hayatta rahatlıkla kullanabileceğimiz bir mekanizma haline gelecektir. El yazısı tanıma sistemlerinin gelişmesi bilgisayarların her yerde kullanılarak günlük hayatımızın içine girmesini hızlandıracak, tarih boyunca var olan belgelerin ulaşılabilir olmasına olanak sağlayacaktır. Bu önemi dolayısıyla el yazısı tanıma hala çok çekici bir problem olmaya devam etmektedir.

El yazısı tanıma sistemlerin doruluğunu artırmak ve işlemi kolaylaştırmak için bazı çalışmalarda çeşitli kısıtlamalar getirilmektedir. Örneğin kişiye bağlı bir sistem tasarlanarak sadece tek bir kişinin yazı karakterinin öğrenilmesi sağlanabilir. Benzer şekilde sözlükten yararlanılarak kelime sayısı kısıtlanabilir. Bir başka yöntemse yazı şeklini sınırlamak ve örneğin kutular içine yazı yazılmasını istemek olabilir. Latin alfabesi kullanarak ve yazı sitili kısıtlanarak belli bir başarıya ulaşmak mümkündür ancak özellikle Çin ya da Arap alfabesi gibi alfabeler için bu işlem oldukça zordur. El yazısı tanıma sistemlerine benzer ve tamamlayıcı diğer sistemlere örnek olarak imza tanıma, denklem ve özel sembollerin tanınması verilebilir.

# EKLER

Projenin Youtube linki ektedir.

<https://www.youtube.com/watch?v=FdLywng3lOY&t=1s>

# KAYNAKLAR

**[1]** Chen, Xiangrong, and Alan L. Yuille. "Detecting and reading text in natural scenes." Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on. Vol. 2. IEEE, 2004.

**[2]** Jain, Anil K., Jianchang Mao, and K. Moidin Mohiuddin. "Artificial neural networks: A tutorial." IEEE computer 29.3 (1996): 31-44.

**[3]** Patil, Vijay, and Sanjay Shimpi. "Handwritten English character recognition using neural network." Elixir Comp. Sci. & Engg 41 (2011): 5587-5591.

**[4]** Sharma, Ankit, and Dipti R. Chaudhary. "Character recognition using neural network." International Journal of Engineering Trends and Technology (IJETT)- Volume4 (2013): 662-667.

**[5]** <http://www.cs.bilkent.edu.tr/~duygulu/papers/BilisimAnsiklopedisi.pdf>, 29.12.2020 tarihinde esinlenilmiştir.

**[6]** Ergezer, Halit, Mehmet Dikmen, and Erkan Özdemir. "Yapay sinir ağları ve tanıma sistemleri." PiVOLKA 2.6 (2003): 14-17.

**[7] “**Deep-Learning-Handwriting-Recognition”, MikeM711, Michael McCabe, “https://github.com/MikeM711/Deep-Learning-Handwriting-Recognition”

**[8]** Büşra Erkan, Yapay Sinir Ağı Tabanlı Karakter Tanıma (Lisans bitirme projesi, Gebze Teknik Üniversitesi, 2017)