**metin içeren bir resim

Açıklama otomatik olarak oluşturuldu**

[**Mühendislik Fakültesi**](https://www.konya.edu.tr/muhendislikvemimarlik)

[**Bilgisayar Mühendisliği Bölümü**](https://www.konya.edu.tr/bilgisayarmuhendisligi)

|  |
| --- |
| **PROJE BAŞLIĞI** |
| Yüksek Düzey Programlama Dönem Proje Ödevi |

|  |  |
| --- | --- |
| **Öğrenci Bilgileri** | |
| **ÖĞRENCİ NO** | 112113172049 |
| **ÖĞRENCİ AD SOYADI** | ABDULLAH OMAR SHUKRI SHUKRI |
| **CEP TELEFON NO** | 05386247800 |
| **EMAİL ADRESİ** | abdallhomar1973@gmail.com |

**DANIŞMAN: Doç.** Dr. Hasan Temurtaş

**HAZIRLAYAN:** ABDULLAH OMER

**KUTAHYA 30.11.2024**

|  |
| --- |
| **Proje Raporu İçeriği** |
| **1. Giriş**  **2. Projenin Amacı**  **3. Kullanılan Araçlar ve Kütüphaneler**  **4. Veri Seti: MNIST**  **5. Proje Adımları**   * **5.1. Kütüphanelerin İçe Aktarılması ve Veri Setinin Yüklenmesi** * **5.2. Verilerin İşlenmesi** * **5.3. Modelin Oluşturulması** * **5.4. Modelin Eğitilmesi** * **5.5. Modelin Değerlendirilmesi** * **5.6. GUI Geliştirilmesi**   **6. Sonuç** |

**1. Giriş**

Bu rapor,

Python kullanarak geliştirdiğim bir dıgıt projesine,

özellikle de el yazısı rakam tanıma uygulamasına odaklanmaktadır.

Proje, MNIST veri setini kullanarak bir Convolutional Neural Network (CNN) modeli oluşturmayı ve bu modeli bir Grafik Kullanıcı Arayüzü (GUI) ile entegre etmeyi amaçlamaktadır.

Uygulama, kullanıcının bir tuval üzerine çizdiği rakamları doğru bir şekilde tanımayı sağlar.

**2. Projenin Amacı**

El yazısı rakam tanıma,

bilgisayarların insanlar tarafından el yazısı ile yazılmış rakamları tanıyabilme yeteneğini geliştirmeyi amaçlar.

Bu, makine öğrenimi ve derin öğrenme alanındaki önemli bir uygulamadır, çünkü el yazısı genellikle düzensiz ve farklı varyasyonlar içerir. Bu proje ile, bu problemi çözebilen bir model geliştirilmiştir.

**3. Gereksinimler**

Proje için aşağıdaki kütüphaneler gereklidir:

* numpy
* tensorflow
* keras
* pillow
* tkinter

**Kütüphanelerin kurulumu:**

Pip install numpy tensorflow keras pillow

**4. Kullanılan Veri Seti: MNIST**

MNIST veri seti,

0-9 arasındaki rakamların el yazısı ile yazılmış toplamda 70.000 gri tonlamalı görüntüsünü içerir.

Görüntüler, her biri 28x28 piksel boyutunda olup, 60.000'i eğitim, 10.000'i ise test için ayrılmıştır.

**4. Proje Adımları**

4.1. Kütüphanelerin İçe Aktarılması ve Veri Setinin Yüklenmesi

Aşağıdaki kod parçası, MNIST veri setinin yüklenmesini sağlar:

from keras.datasets import mnist

from keras.models import Sequential

from keras.layers import Dense, Dropout, Flatten, Conv2D, MaxPooling2D

from keras import backend as K

# Veriler eğitim ve test kümeleri arasında ayrılır

(x\_train, y\_train), (x\_test, y\_test) = mnist.load\_data()

print(x\_train.shape, y\_train.shape)

**4.2. Verilerin İşlenmesi**

Veriler modele uygun hale getirilmiştir: Görüntü verileri modele uygun hale getirilecek şekilde yeniden boyutlandırılır ve normalleştirilir. Ayrıca, etiketler **one-hot encoding** yöntemiyle kategorik hale getirilir:

x\_train = x\_train.reshape(x\_train.shape[0], 28, 28, 1)

x\_test = x\_test.reshape(x\_test.shape[0], 28, 28, 1)

input\_shape = (28, 28, 1)

# Sınıf vektörlerini ikili sınıf matrislerine dönüştürme

num\_classes = 10

y\_train = keras.utils.to\_categorical(y\_train, num\_classes)

y\_test = keras.utils.to\_categorical(y\_test, num\_classes)

# Verileri normalleştirme

x\_train = x\_train.astype('float32') / 255 x\_test = x\_test.astype('float32') / 255

print(f"x\_train şekli: {x\_train.shape}") print(f"{x\_train.shape[0]} eğitim örneği")

print(f"{x\_test.shape[0]} test örneği")

**4.3. Modelin Oluşturulması**

CNN modeli, katmanlar eklenerek oluşturulmuştur:

model = Sequential()

model.add(Conv2D(32, kernel\_size=(3, 3), activation='relu', input\_shape=(28, 28, 1)))

model.add(Conv2D(64, (3, 3), activation='relu'))

model.add(MaxPooling2D(pool\_size=(2, 2)))

model.add(Dropout(0.25))

model.add(Flatten())

model.add(Dense(256, activation='relu'))

model.add(Dropout(0.5))

model.add(Dense(10, activation='softmax'))

model.compile(loss='categorical\_crossentropy', optimizer='Adadelta', metrics=['accuracy'])

**4.4. Modelin Eğitilmesi**

Modeli eğitmek için eğitim verileri kullanılır. Eğitilen model daha sonra kaydedilir.

hist = model.fit(x\_train, y\_train, batch\_size=batch\_size, epochs=epochs,

verbose=1, validation\_data=(x\_test, y\_test))

print("Model başarıyla eğitildi")

model.save('mnist.h5')

print("Model mnist.h5 olarak kaydedildi")

**4.5. Modelin Değerlendirilmesi**

Test veri seti üzerinde modelin başarısı ölçülmüştür:

score = model.evaluate(x\_test, y\_test)

print('Test loss:', score[0])

print('Test accuracy:', score[1])

**4.6. GUI Geliştirilmesi**

Rakamları Tahmin Etmek İçin GUI Oluşturma:

from keras.models import load\_model

from tkinter import \*

from PIL import ImageGrab

import numpy as np

model = load\_model('mnist.h5')

class App(Tk):

# GUI'nin oluşturulması ve çizim işlevleri

# ...

app = App()

app.mainloop()

**5. Ekran Görüntüleri :**

**metin, ekran görüntüsü, yazılım, bilgisayar içeren bir resim

Açıklama otomatik olarak oluşturuldu**

metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim

Açıklama otomatik olarak oluşturuldu

**metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim

Açıklama otomatik olarak oluşturuldu**

metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim

Açıklama otomatik olarak oluşturuldumetin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Proje Özeti**

Bu projede:

1. CNN modeli kullanılarak el yazısı rakam tanıma başarıyla gerçekleştirildi.
2. Model test verilerinde yaklaşık %99 doğruluk elde ettim.
3. Tahmin sonuçlarını görselleştirmek için Tkinter ile bir GUI oluşturuldum.