



Mühendislik Fakültesi

Bilgisayar Mühendisliği Bölümü

PROJE BAŞLIĞI
Yüksek Düzey Programlama Dönem Proje Ödevi

Öğrenci Bilgileri	
ÖĞRENCİ NO	112113172049
ÖĞRENCİ AD SOYADI	ABDULLAH OMAR SHUKRI SHUKRI
CEP TELEFON NO	05386247800
EMAIL ADRESİ	abdallhomar1973@gmail.com

DANIŞMAN: Doç. Dr. Hasan Temurtaş

HAZIRLAYAN: ABDULLAH OMER

KUTAHYA 30.11.2024

Proje Raporu İeriđi

1. Giriř

2. Projenin Amacı

3. Kullanılan Aralar ve Kütüphaneler

4. Veri Seti: MNIST

5. Proje Adımları

- 5.1. Kütüphanelerin İe Aktarılması ve Veri Setinin Yklenmesi
- 5.2. Verilerin İřlenmesi
- 5.3. Modelin Oluřturulması
- 5.4. Modelin Eđitilmesi
- 5.5. Modelin Deđerlendirilmesi
- 5.6. GUI Geliřtirilmesi

6. Sonu



KÜTAHYA DÜMLUPINAR ÜNİVERSİTESİ
2024 / 2025 EĞİTİM-ÖĞRETİM YILI
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
BİLGİSAYAR MÜHENDİSLİĞİNDE TEZ
VİZE RAPORU

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")
```



KÜTAHYA DÜMLUPINAR ÜNİVERSİTESİ
2024 / 2025 EĞİTİM-ÖĞRETİM YILI
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
BİLGİSAYAR MÜHENDİSLİĞİNDE TEZ
VİZE RAPORU

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

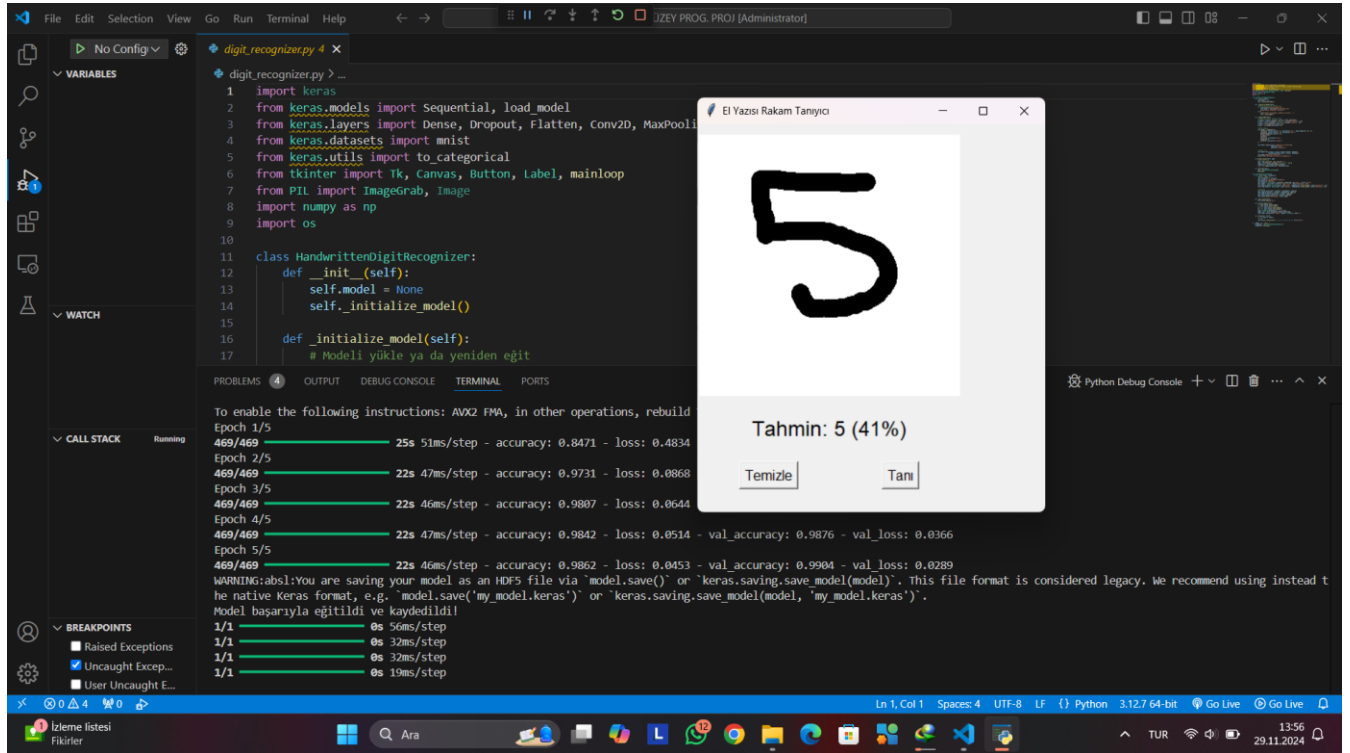
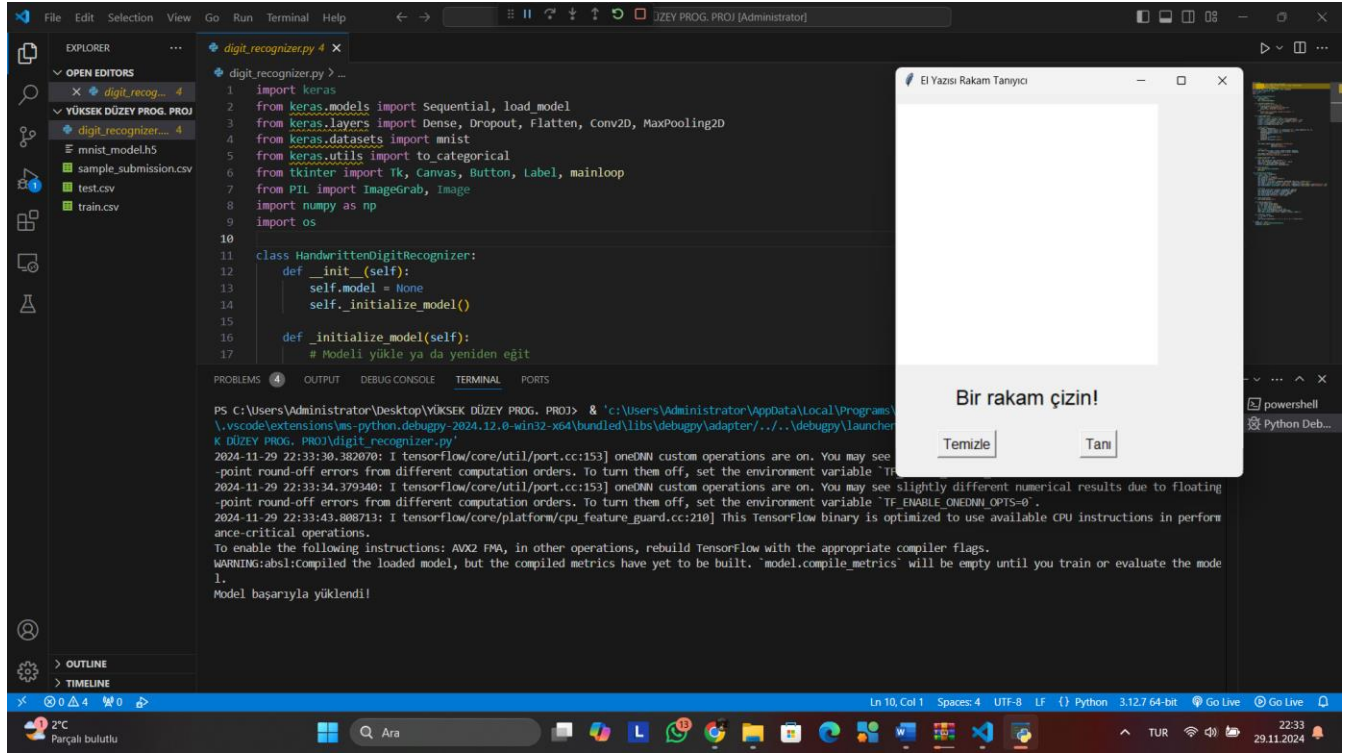
Rakamları Tahmin Etmek İçin GUI Oluşturma:

```
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
    # ...

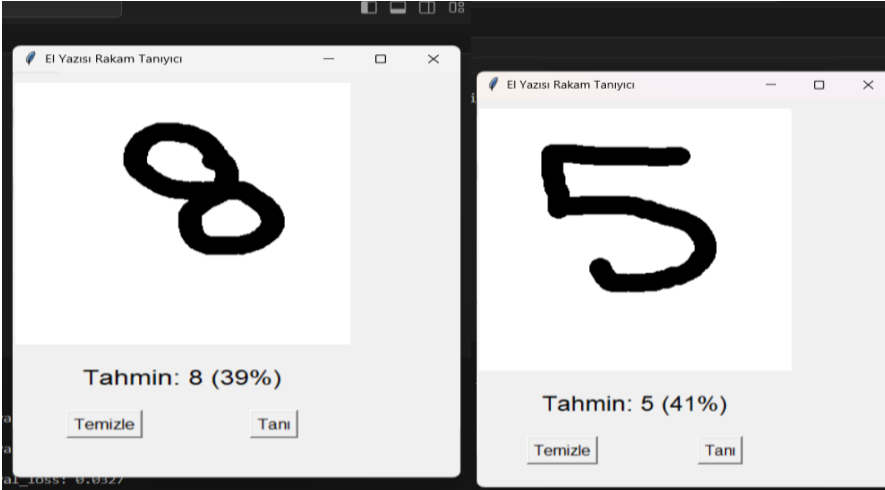
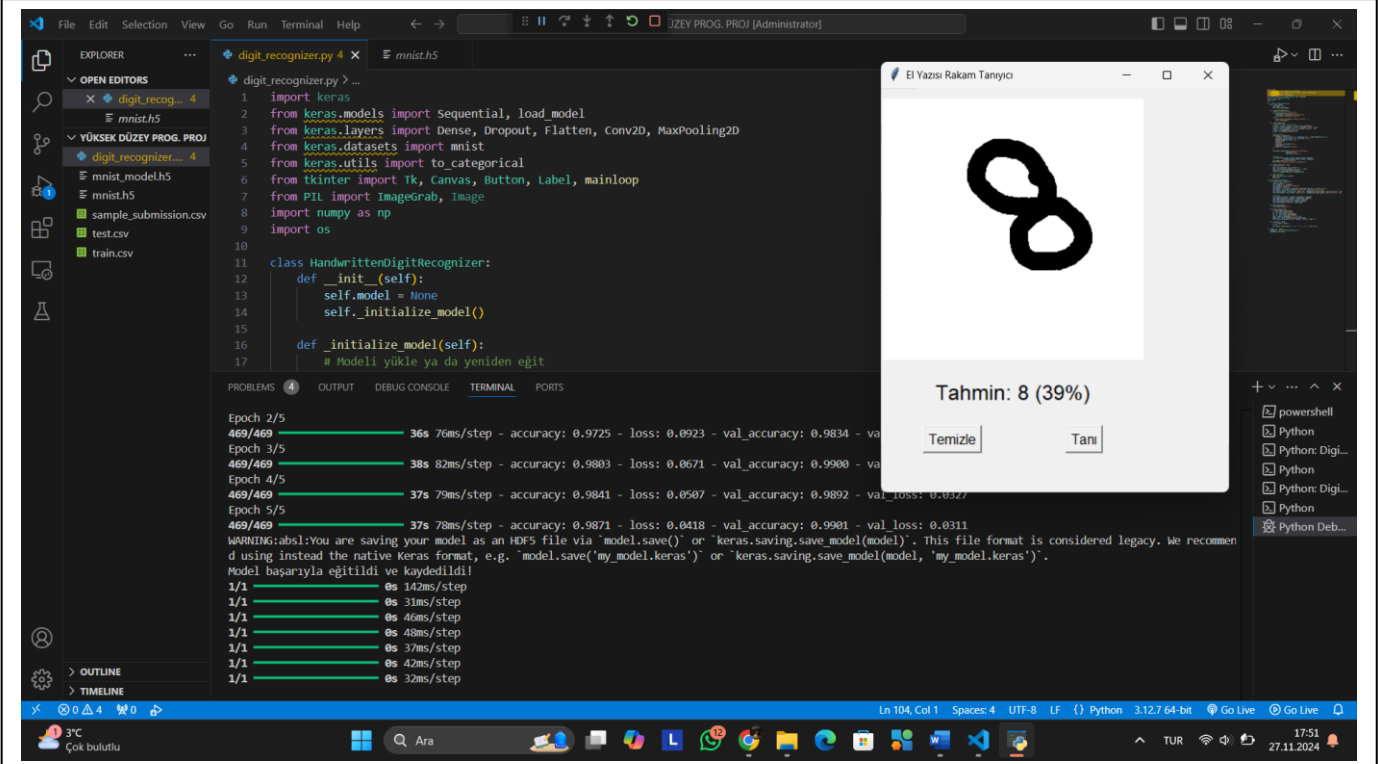
app = App()
app.mainloop()
```

5. Ekran Görüntüleri :





KÜTAHYA DUMLUPINAR ÜNİVERSİTESİ
2024 / 2025 EĞİTİM-ÖĞRETİM YILI
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
BİLGİSAYAR MÜHENDİSLİĞİNDE TEZ
VİZE RAPORU



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.