



**YILDIZ TEKNİK ÜNİVERSİTESİ ELEKTRİK
ELEKTRONİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ**

Yapay Zeka Dersi
Proje Raporu

20011626 - Yusuf Yemliha ÇELİK

17011085 - Saliha KOÇYİĞİT

Giriş

Bu proje, Python ile el yazısı rakamları tanımak ve sonrasında tahmin etmek için bir yapay zeka modeli geliştirmeyi amaçlamaktadır. İnsanlardan topladığımız verileri kullanarak bir makine öğrenme algoritması ile bir model oluşturduk. Sonrasında oluşturduğumuz modeli kullanmaya yönelik bir arayüz (GUI) tasarladık. Veri seti hazırlama, model oluşturma, eğitim, test ve tahmin adımlarını içeren proje, kullanıcı dostu bir arayüz sağlayarak derin öğrenme teknolojisini daha erişilebilir hale getirmeyi hedeflemektedir.

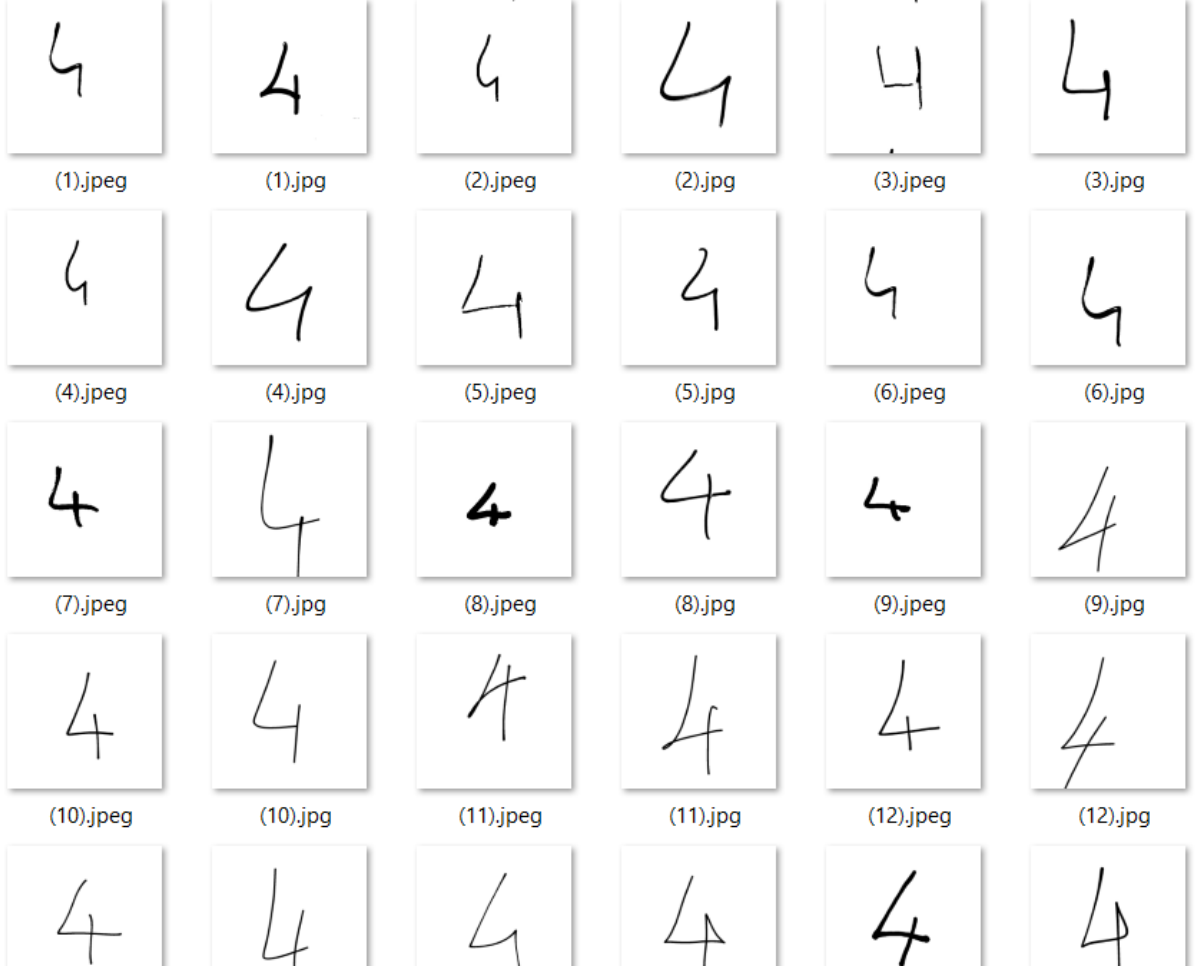
1. Veri Toplama

Proje sürecinde, insanlardan el yazısı rakamlarını içeren veri topladık. Veri toplama aşamasında, her bir rakamın çeşitli örneklerini topladık. Topladığımız verileri daha sonra kullanılabilir hale getirmek için işledik ve veri setimizi oluşturduk.



2. Veri Ön İşleme

Veri ön işleme aşamasında, topladığımız el yazısı rakamlarını gri tonlamalı görüntüleriye dönüştürdük. Görüntülerin boyutunu 28x28 piksel boyutuna yeniden boyuttlandırdık ve piksel değerlerini 0 ila 1 arasında normalize ettik.



3. Model Oluşturma ve Eğitim

Model oluşturma aşamasında, bir evrişimli sinir ağı (convolutional neural network) kullanmayı tercih ettik. Modelimiz, iki evrişimli katman, iki maksimum havuzlama katmanı, tam bağlantılı bir katman ve bir çıkış katmanından oluşuyor. Modeli derledik ve eğitmek için loss fonksiyonu olarak kategorik çapraz entropi (categorical cross-entropy) ve optimizasyon algoritması olarak Adam'ı kullandık.

Eğitim sürecinde, veri setimizi eğitim ve test veri setlerine ayırdık. Modeli eğitirken, veri setimizi kullanarak modelin parametrelerini güncelledik. Eğitim sürecini 25 epoch boyunca yürüttük ve her epoch sonunda modelin performansını değerlendirdik.

4. GUI Tasarımı

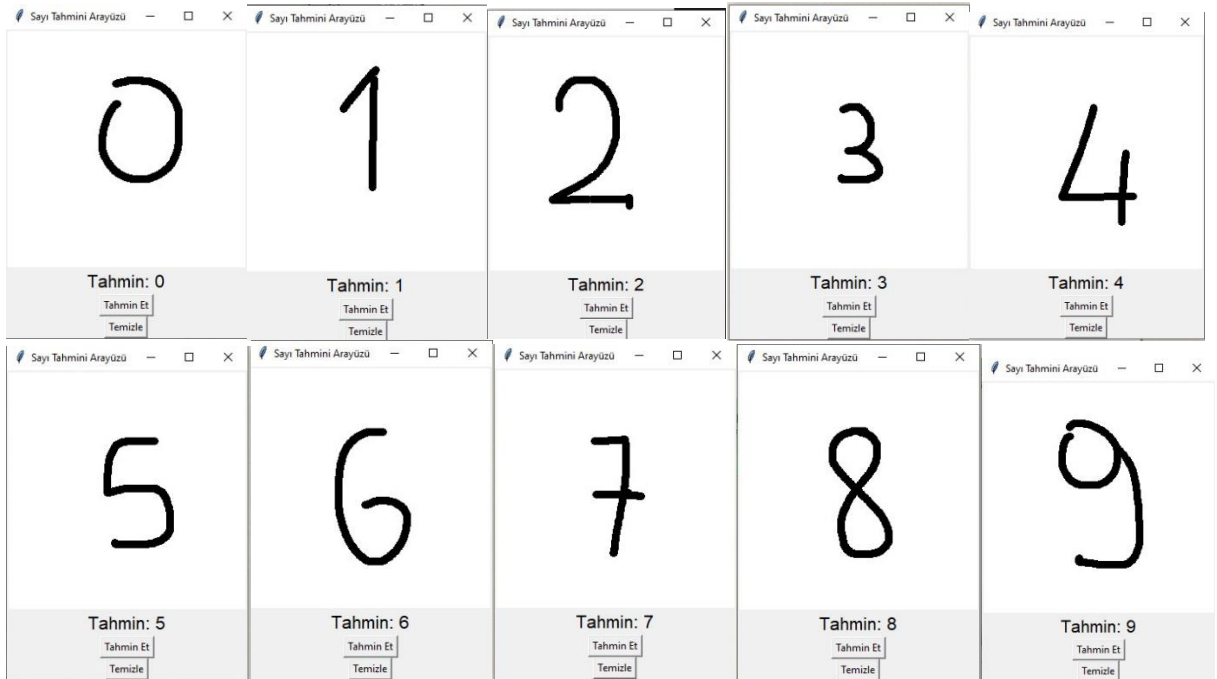
Eğittiğimiz modeli kullanarak bir grafik kullanıcı arayüzü (GUI) tasarladık. Kullanıcı, çizim alanında kendi el yazısı rakamlarını çizebilir ve modelimiz tarafından tahmin edilen rakamı görebilir. GUI, Tkinter kütüphanesi kullanılarak Python programlama diliyle oluşturuldu.

5. Sonuçlar ve Başarı Ölçümü

Eğitim sürecinin sonunda, modelin test veri setindeki başarı oranını değerlendirdik. Test kaybı ve doğruluk değerlerini hesapladık. Modelimiz test veri setinde 1.2737 loss ve 0.5937 doğruluk elde etti.

```
1/1 [=====] - 0s 148ms/step - loss: 1.2737 - accuracy: 0.5938  
Test loss: 1.273709774017334  
Test accuracy: 0.59375
```

Başarıyı ölçmek için, gerçek rakamların etiketleriyle tahmin edilen rakamları karşılaştırdık ve doğru tahminlerin yüzdesini hesapladık. Bu doğruluk oranı, modelimizin el yazısı rakamları tanımada ne kadar başarılı olduğunu göstermektedir.



Projemizin başarı oranını daha da artırmak için farklı yaklaşımlar denenebilir. Örneğin, daha fazla veri toplanabilir veya modelin mimarisi ve hiper parametreleri ayarlanabilir. Ayrıca, veri artırma teknikleri kullanarak daha çeşitli ve zorlu veri örnekleriyle modeli eğitebiliriz.

8. Kaynaklar

- TensorFlow: [tensorflow.org](https://www.tensorflow.org)
- OpenCV: opencv.org
- Tkinter: docs.python.org/3/library/tkinter.html