



Kahramanmaraş Sütçü İmam Üniversite  
Mühendislik ve Mimarlık Fakültesi  
Bilgisayar Mühendisliği Bölümü

BİL419 Yapay Zeka Dersi Dönem Ödevi

ABDURRAHMAN KARAOĞLU - 18110131307

AHMET ÖZBERK – 18110131310

## Özet

Yapay Zeka öğrenme, gerçekleştirme, problem çözme, yabancı bir dili anlama v.b. gibi insanoğlunun davranışlarını gösterebilen sistemlerle ilgilenen bir bilgisayar bilimidir. Yapay zekâ, insanın düşünme yöntemlerini analiz ederek bunların benzeri yapay yönergeleri geliştirmeye çalışmaktır.

Günümüzde yaygın olarak kullanılan başlıca yapay zeka yöntemleri; uzman sistemler, bulanık mantık ve yapay sinir ağlarıdır. İlk yapay zekanın temellerini 1943 yılında Warren ve Walter Pitts tarafından atılmıştır. Öğrenen bir sistemin oluşturulabileceğini iddia etmişlerdir. 1950 li yıllarda yapay zeka gelişmeye başlamış ve bir Genel Problem Çözücü yapılmaya çalışılmıştır. 1980 lere doğru teknolojinin gelişmesiyle yapay zeka çalışmalarına tekrar ağırlık verilmiştir. Genelden çok özele yönelilmiş ve Mycin'in temeli atılmıştır.[1] Projemiz kapsamında makine öğrenmesi ve yapay zeka teknikleri ile otonom araçların temelini oluşturan, trafik işaretlerini modelleyen ve tanıyan bir proje geliştirilmiştir.

## 1.Giriş

Bu çalışmada farklı trafik işaretleri verilerinin elde edildikten sonra, bu veriler sayesinde makine öğrenmesi algoritmamızı eğiterek, hangi resimdeki trafik işaretinin hangi anlama geldiği eğitilmiş modele gönderilir ve eğitilmiş olan model verilen yeni resimlerdeki trafik işaretlerinin hangi anlamlara geldiğini hızlı bir şekilde bulabiliyoruz. Verilerimizde belirtilen trafik işaretleri, 43 farklı sınıfa göre sınıflandırılır. Projede test amaçlı 12.631 adet test verisi bulunmaktadır. Eğitim verisi için de 39.210 adet veri bulunmaktadır.

## 2.Verilerin Eğitime Hazır Hale Getirilmesi

Bir modeli eğitirken, modeli farklı sınıflardan rastgele girdiler sağlamak, modelin daha iyi genellenebilmesi için önemlidir.

Bu nedenle, verileri rastgele olarak eğitim ve doğrulama kümesine bölecek olan sklearn `train_test_split()` işlemini kullanacağız.

```
In [4]: #Splitting eğitim ve test veri kümesi
X_train, X_test, y_train, y_test = train_test_split(data, labels, test_size=0.2, random_state=42)

print(X_train.shape, X_test.shape, y_train.shape, y_test.shape)

y_train = to_categorical(y_train, 43)
y_test = to_categorical(y_test, 43)

(31367, 30, 30, 3) (7842, 30, 30, 3) (31367,) (7842,)
```

### 3. Model Mimarisi

Evrişimli Sinir Ağları, görüntü sınıflandırma görevlerinde en son teknoloji olarak kanıtlamıştır ve modelimiz için kullanacağımız şey budur. Bir Evrişimsel Sinir Ağı (CNN), evrişim ve havuzlama katmanlarından oluşur. Her katmanda, görüntünün sınıflandırılmasına yardımcı olan görüntüden öznitelikler çıkarılır.

Modeli `categorical_crossentropy` ile derliyoruz çünkü veri kümemizde sınıflandırılacak birden fazla sınıf var.

```
In [ ]: #Modelin oluşturulması
model = Sequential()
model.add(Conv2D(filters=32, kernel_size=(5,5), activation='relu', input_shape=X_train.shape[1:]))
model.add(Conv2D(filters=32, kernel_size=(5,5), activation='relu'))
model.add(MaxPool2D(pool_size=(2, 2)))
model.add(Dropout(rate=0.25))
model.add(Conv2D(filters=64, kernel_size=(3, 3), activation='relu'))
model.add(Conv2D(filters=64, kernel_size=(3, 3), activation='relu'))
model.add(MaxPool2D(pool_size=(2, 2)))
model.add(Dropout(rate=0.25))
model.add(Flatten())
model.add(Dense(256, activation='relu'))
model.add(Dropout(rate=0.5))
model.add(Dense(43, activation='softmax'))

#Modelin derlenmesi
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
```

### 4. Modeli Eğitme ve Kaydetme

Artık model tanımlandı ve veriler hazır. Modelimizin eğitimini başlatmak için `model.fit()` işlevini kullanırız.

Modeli 15 devir için eğittikten sonra modeli bir `traffic_recognition.h5` dosyasına kaydedeceğiz.

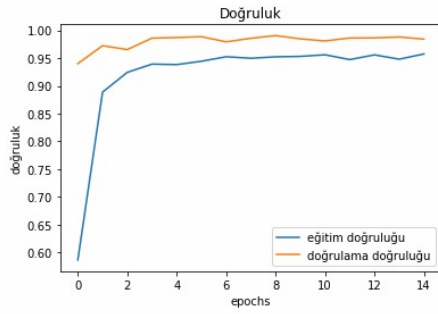
```
In [*]: epochs = 15
history = model.fit(X_train, y_train, batch_size=32, epochs=epochs, validation_data=(X_test, y_test))

Epoch 1/15
981/981 [=====] - 135s 136ms/step - loss: 1.6322 - accuracy: 0.5863 - val_loss: 0.2551 - val_accuracy: 0.9401
Epoch 2/15
981/981 [=====] - 125s 128ms/step - loss: 0.4191 - accuracy: 0.8887 - val_loss: 0.1155 - val_accuracy: 0.9723
Epoch 3/15
981/981 [=====] - 110s 112ms/step - loss: 0.2805 - accuracy: 0.9243 - val_loss: 0.1246 - val_accuracy: 0.9654
```

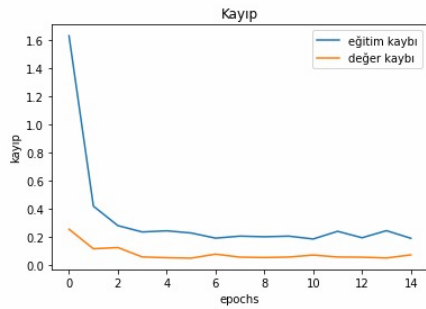
## 5. Accuracy Grafiği Hazırlanması

Matplotlib fonksiyonlarının yardımıyla eğitim ve doğruluk grafiği çizeceğiz.

```
In [8]: #doğruluk için grafikler çizme
plt.figure(0)
plt.plot(history.history['accuracy'], label='eğitim doğruluğu')
plt.plot(history.history['val_accuracy'], label='doğrulama doğruluğu')
plt.title('Doğruluk')
plt.xlabel('epochs')
plt.ylabel('doğruluk')
plt.legend()
plt.show()
```



```
In [9]: plt.figure(1)
plt.plot(history.history['loss'], label='eğitim kaybı')
plt.plot(history.history['val_loss'], label='değer kaybı')
plt.title('Kayıp')
plt.xlabel('epochs')
plt.ylabel('kayıp')
plt.legend()
plt.show()
```



## 6. Modeli Test Etme

Modelimizi test etmek için yaklaşık 12.000 görüntü içeren bir test klasörümüz var. 'test.csv' dosyası, sınıfın etiketiyle birlikte görüntünün yolunu içerir. Pandas, CSV dosyasından yol ve etiket çıkarmak için kullanacağımız kitaplıktır ve ardından sklearn accuracy\_score() işlevinin yardımıyla, gerçek değerleri modelimizin tahmin edilen değerleriyle karşılaştırabiliriz.

```
In [10]: #test veri kümesinde test doğruluğu
from sklearn.metrics import accuracy_score

y_test = pd.read_csv('Test.csv')

labels = y_test["ClassId"].values
imgs = y_test["Path"].values

data=[]

for img in imgs:
    image = Image.open(img)
    image = image.resize((30,30))
    data.append(np.array(image))
```

```
In [11]: X_test=np.array(data)

pred = np.argmax(model.predict(X_test), axis=-1)
```

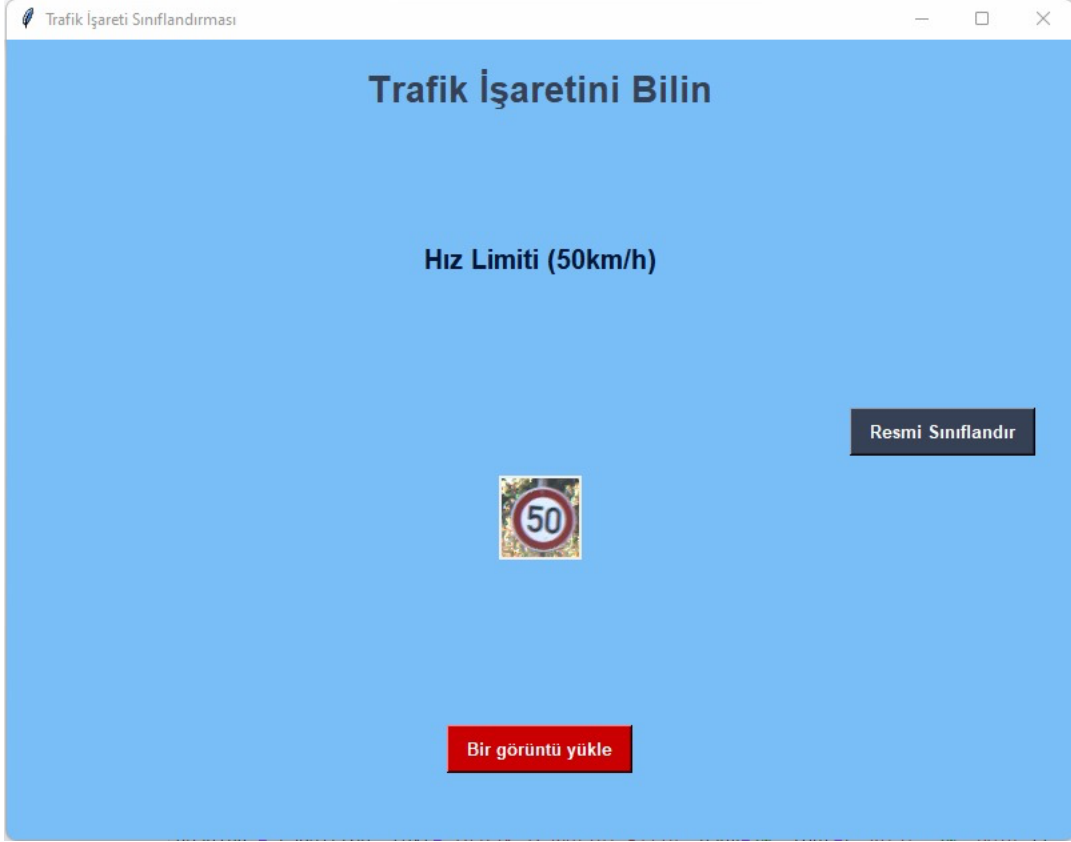
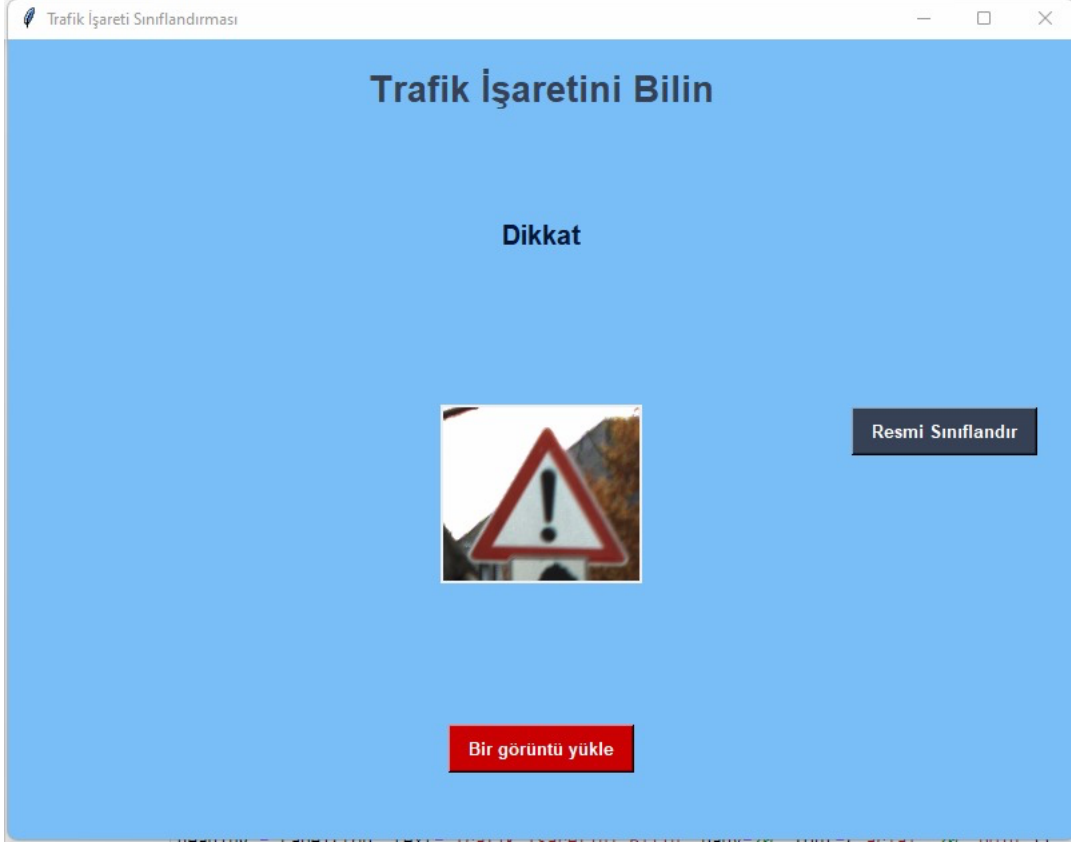
```
In [12]: #Test verileriyle doğruluk
accuracy_score(labels, pred)
```

```
Out[12]: 0.9471892319873317
```

Eğitilen model sonucunda % 94 gibi bir başarı oranına sahip oluyoruz.

## 7. Kullanıcı Arayüzü Oluşturma

Python programlama dilinin dahili olarak yüklü gelen Tkinter kitaplığı ile kullanıcı arayüzü oluşturulmuştur. Oluşturulan arayüzde, kullanıcıdan bir adet trafik işareti içeren resim dosyası eklenmesi istenmektedir. Eklenen resim dosyası modele gönderilerek, gelen değer ekrana basılır ve trafik işaretinin hangi anlama geldiği yazdırılır.



## KAYNAKLAR

[1] Öztemel E., “Yapay Sinir Ağları”, Papatya Kitapevi, İstanbul,2006.