



T.C

**KOCAELİ SAĞLIK VE TEKNOLOJİ ÜNİVERSİTESİ  
EĞİTİM  
MÜHENDİSLİK ve DOĞA BİLİMLERİ FAKÜLTESİ  
YAZILIM MÜHENDİSLİĞİ PROGRAMI**

**ÖDEV KONUSU**

**Yapay Zeka Destekli Hayvan Görüntü Sınıflandırıcı**

**Hazırlayan**

**Gürel Bilgin**

**220502041**

**GitHub Repo: <https://github.com/GurelBilgin/image-classifier>**

**30/12/2025**

## 1. Proje Tanımı:

Bu proje, bir görüntü sınıflandırma modeli geliştirerek kullanıcıların yüklediği hayvan görsellerini tanıyalıbilen bir yapay zekâ uygulamasıdır. Model, PyTorch ile eğitilmiş **ResNet18** tabanlı bir derin öğrenme mimarisi kullanır. Kullanıcı arayüzü Streamlit ile oluşturulmuş olup, görseller yüklenip işlendikten sonra sınıf tahmini ve güven oranı ekrana yansıtılır.

## Hedef:

- Görselleri sınıflandırabilen bir yapay zekâ modeli oluşturmak
- Kullanıcı dostu ve sade bir web arayüzü ile modelin kullanımını sağlamak

## 2. Kullanılan Teknolojiler

- Model:** ResNet18 (PyTorch)
- Arayüz:** Streamlit
- Veri İşleme:** torchvision, PIL, numpy
- Veri Seti:** Animals-10 Dataset (Kaggle)
- Geliştirme Ortamı:** Python 3.x, CPU/GPU desteği

## 3. Fonksiyonel Gereksinimler ve Gerçekleştirme

Gereksinim	Açıklama	Durum
Görüntü yükleme	Kullanıcı bilgisayarından resim yükleyebilir	<input checked="" type="checkbox"/> Gerçekleştirildi
Ön işleme	Boyutlandırma, kırpmalar, normalize	<input checked="" type="checkbox"/> Gerçekleştirildi (data_utils.py)
Model sınıflandırması	Görüntüyü sınıflandırıp metin ile gösterme	<input checked="" type="checkbox"/> Gerçekleştirildi (train.py + model_utils.py)
Kullanıcı arayüzü	Web arayüzü üzerinden tahmin ve görsel gösterimi	<input checked="" type="checkbox"/> Streamlit (app.py)
Tahmin güveni	Tahmin güven oranı gösterimi (%85 altı uyarısı)	<input checked="" type="checkbox"/> Gerçekleştirildi

#### 4. Model Eğitimi ve Parametreler

- **Model:** ResNet18 (önceki eğitilmiş ağırlıklar kullanılmadan)
- **Optimizer:** Adam, lr=3e-4
- **Batch Size:** 32
- **Epochs:** 15
- **Girdi Görüntü Boyutu:** 224x224
- **Ön İşleme:** Resize, CenterCrop, RandomHorizontalFlip, RandomRotation, Normalize

**Metrikler:** Accuracy, Precision, Recall, Confusion Matrix

#### 5. Veri Seti

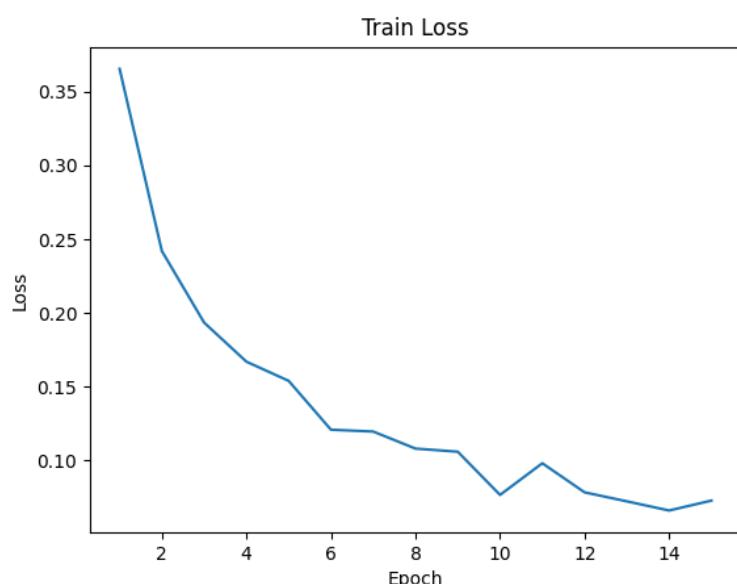
- **Kaynak:** [Animals-10 Dataset](#)
- **Sınıf Sayısı:** 8
- **Sınıflar:** Köpek, At, Fil, Tavuk, Kedi, İnek, Koyun, Sincap
- **Veri Ayrımı:** %80 eğitim, %20 test

**Not:** farfalla (kelebek) ve ragno (örümcek) sınıfları veri setinden çıkartıldı.

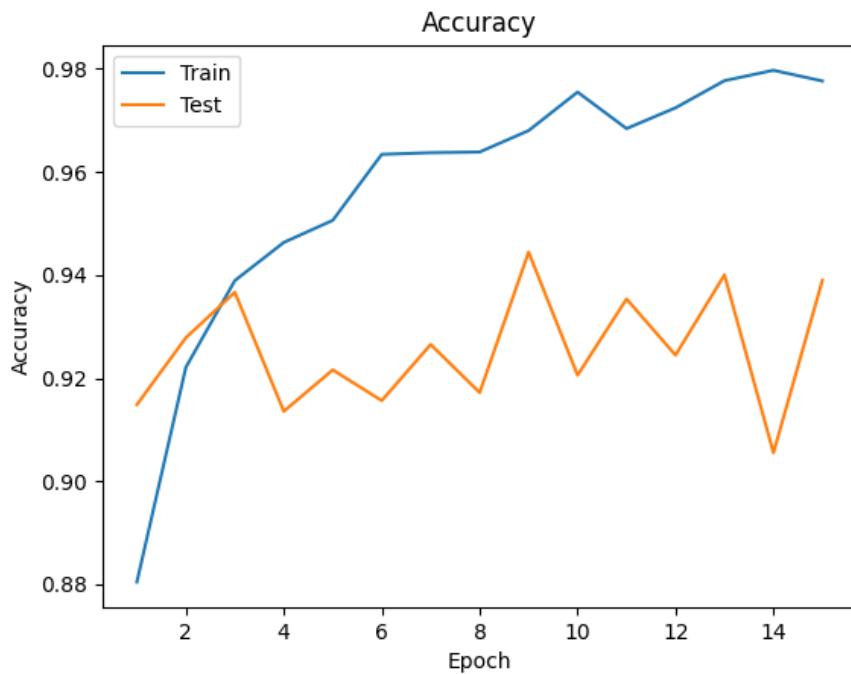
#### 6. Eğitim Sonuçları

- Eğitim sırasında loss, accuracy, precision ve recall değerleri loglanmıştır.
- Eğitim ve test performansları grafikleri metrics/ klasöründe kaydedilmiştir.

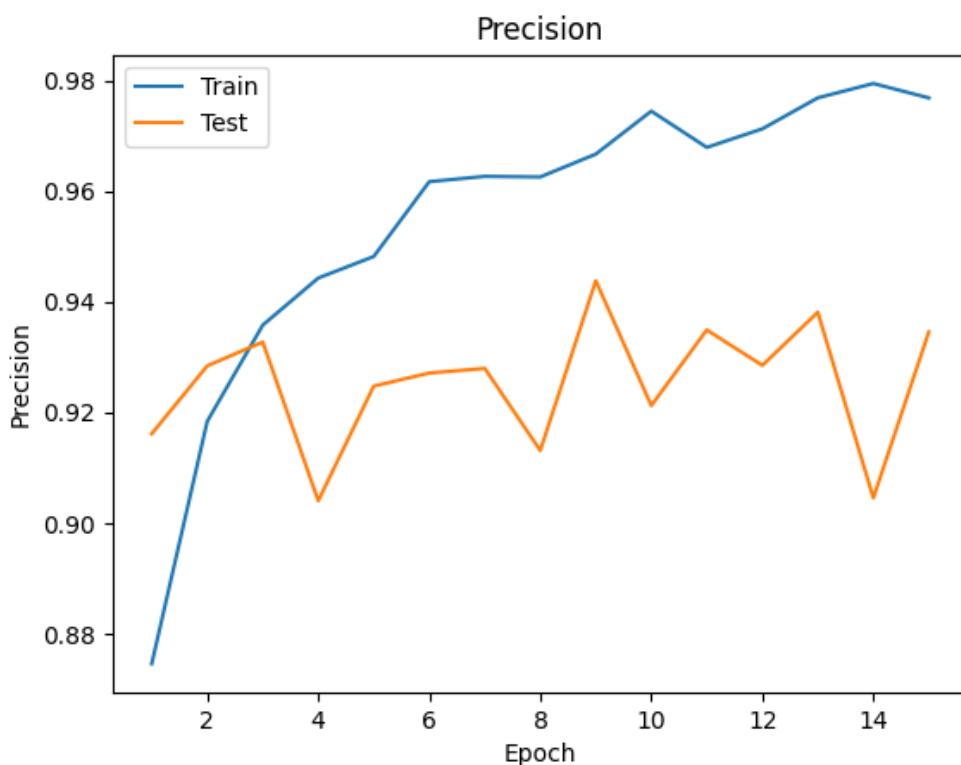
Eğitim kaybı (loss) her epoch için:



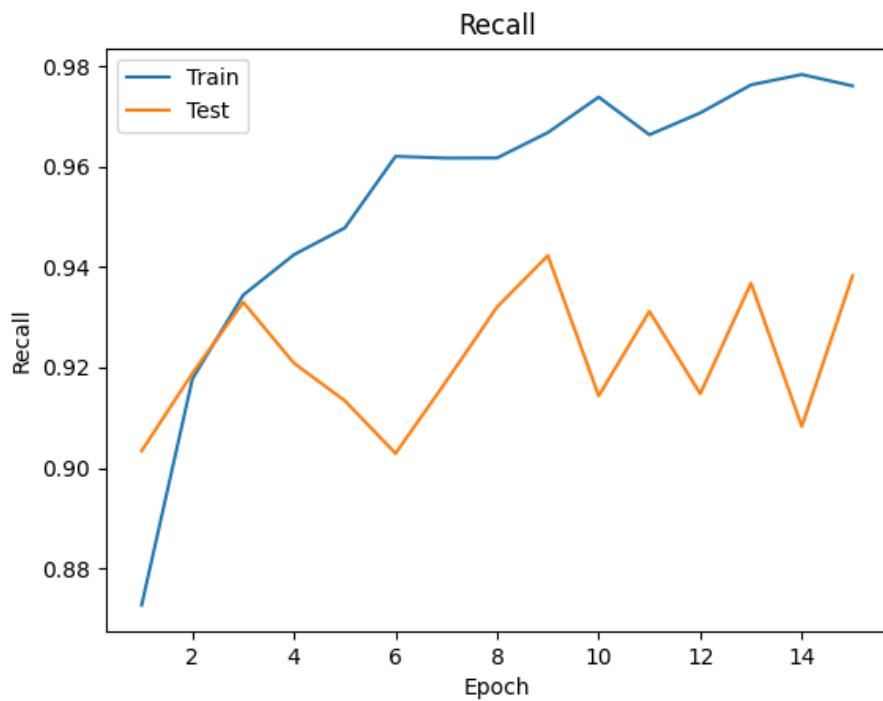
Eğitim ve test doğruluk oranı:



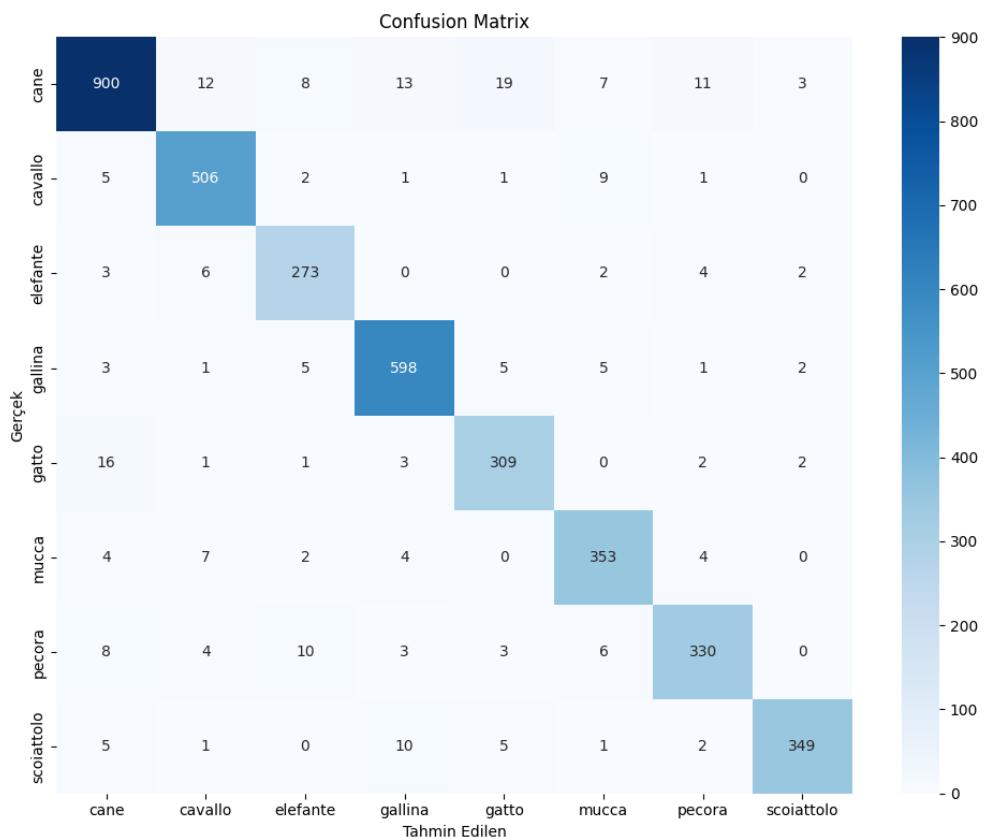
Eğitim ve test precision:



Eğitim ve test recall:



Sınıflar arası tahmin performansı:



## 8. Kullanıcı Arayüzü

- **Yükleme Butonu:** Kullanıcı resim seçebilir
- **Tahmin Et Butonu:** Model tahmin yapar
- **Tahmin Sonucu:** Sınıf ismi ve güven oranı (%85'in altında uyarı)
- **Desteklenen Sınıflar:** Kullanıcıya bir expander içinde listelenir

### Arayüz Görseli:

The screenshot shows the user interface of a Streamlit application for animal classification. At the top, there's a title section with an icon of a paw print and the text "Yapay Zeka Destekli Hayvan Görüntü Sınıflandırıcı". Below it, a descriptive text states: "Bu uygulama, hayvan görselinizi yapay zekâ ile değerlendirecek tahmin ettiği sınıfı ve güven oranını ekrana getirir." A "Nasıl Çalışır?" section provides instructions: 1. Bilgisayarınızdan bir hayvan görseli yükleyin, 2. Tahmin Et butonuna basın, 3. Yapay zekânın tahminini ve güven oranını görüntüleyin. It includes a "Desteklenen Hayvan Türleri" link. The main area has a "Bir hayvan resmi yükleyin" section with a file upload input labeled "Drag and drop file here" and "Limit 200MB per file • JPG, PNG, JPEG", and a "Browse files" button. At the bottom, there's a "Model Nasıl Çalışır?" link and a footer with the text "Model: ResNet18 | Framework: PyTorch | Arayüz: Streamlit".

## 6. Arayüz Tasarımı

- Streamlit tabanlı kullanıcı arayüzü
- Yüklenen görselin gösterimi
- Tahmin edilen sınıf ve güven oranı
- Desteklenen hayvan türlerinin listelenmesi

## 9. Dosya Yapısı

```
image_classifier/
├── dataset/
│   ├── test/
│   └── train/
└── metrics/
    ├── accuracy.png
    ├── confusion_matrix.png
    ├── precision.png
    ├── recall.png
    └── train_loss.png
└── src/
    ├── __init__.py
    ├── data_utils.py
    ├── model_utils.py
    └── translate.py
└── trained_models/
    ├── classes.json
    ├── model.pth
    └── training_log.csv
└── app.py
└── train.py
└── README.md
└── requirements.txt
└── screenshots/
```

## 11. Değerlendirme ve Sonuç

- **Model Performansı:** Eğitilen ResNet18 tabanlı model, yüklenen hayvan görsellerini yüksek doğrulukla sınıflandırabilmektedir. Eğitim ve test süreçlerinde elde edilen **accuracy, precision ve recall** değerleri modelin güvenilirliğini ve genellemeye yeteneğini göstermektedir. Confusion matrix analizleri, sınıflar arası karışıklıkların minimum olduğunu ortaya koymaktadır.
- **Kullanıcı Arayüzü:** Streamlit tabanlı arayüz, kullanıcıların görselleri kolayca yükleyebilmesine ve sonuçları hızlı bir şekilde görüntüleyebilmesine olanak tanır. Arayüz tasarıımı sade ve anlaşılır olup, kullanıcı deneyimi ön planda tutulmuştur. Desteklenen sınıfların listesi ve tahmin güven oranı gösterimi, kullanıcıya ek bilgi sunmaktadır.
- **Tahmin Güveni:** Model, %85'in altındaki tahminlerde kullanıcıyı uyararak yanlış sınıflandırmaların önüne geçmektedir. Bu mekanizma, kullanıcıya yalnızca güvenilir tahminleri sunarak uygulamanın doğruluk ve güvenilirliğini artırmaktadır.
- **Genel Sonuç:** Proje kapsamında geliştirilen yapay zekâ destekli görüntü sınıflandırıcı, hem teknik hem de kullanıcı deneyimi açısından başarılı bir şekilde çalışmaktadır. Eğitim metrikleri ve görsel arayüz entegrasyonu, modelin gerçek kullanım senaryolarında güvenle kullanılabileceğini göstermektedir.

## 12. Kaynaklar

- [Animals-10 Dataset](#)
- [PyTorch ResNet18 Tutorial](#)
- [PyTorch Transforms](#)
- [Streamlit Documentation](#)