

GÖRÜNTÜ TABANLI ATIK SINIFLANDIRMA PROJESİ RAPORU

İSTANBUL MEDENİYET ÜNİVERSİTESİ

MÜHENDİSLİK FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

DERS ADI: DERİN ÖĞRENME

DERSİN ÖĞRETİM ELEMANI: İSHAK DÖLEK

HAZIRLAYAN:

AD SOYAD: BETÜL TOK

1. Proje Konusu ve Amacı

Günümüzde artan nüfus ve tüketim alışkanlıkları, atık miktarının hızla artmasına neden olmaktadır. Atıkların doğru şekilde ayrıştırılmaması hem çevresel kirliliği artırmakta hem de geri dönüşüm süreçlerinin verimliliğini düşürmektedir. Bu nedenle atıkların otomatik olarak sınıflandırılabilmesi, sürdürülebilir çevre yönetimi açısından büyük önem taşımaktadır.

Bu projede, derin öğrenme tabanlı görüntü sınıflandırma yöntemi kullanılarak atık türlerinin otomatik olarak tanınması amaçlanmıştır. Projenin seçilme nedeni; görüntü işleme ve yapay zekâ tekniklerinin çevresel problemlerin çözümünde etkin bir şekilde kullanılabilmesidir.

Literatürde atık sınıflandırma problemleri genellikle Evrişimsel Sinir Ağları (CNN) kullanılarak çözülmektedir. Özellikle ResNet, VGG ve MobileNet gibi mimarilerin bu alanda başarılı sonuçlar verdiği bilinmektedir. Bu projede ise yüksek doğruluk ve düşük eğitim maliyeti sağlaması nedeniyle ResNet18 mimarisi tercih edilmiştir.

2. Veri Seti

Bu çalışmada kullanılan veri seti, **UCI Machine Learning Repository** üzerinden temin edilen **RealWaste Dataset**'tir. UCI Machine Learning Repository, makine öğrenmesi alanında akademik çalışmalarda sıklıkla kullanılan, güvenilir ve referans gösterilebilir bir veri kaynağıdır.

Veri Setinin Özellikleri

RealWaste veri seti, gerçek yaşam koşullarında çekilmiş atık görüntülerinden oluşmaktadır. Veri seti, farklı atık türlerinin sınıflandırılmasına olanak sağlayacak şekilde hazırlanmıştır.

Veri seti özellikleri:

- **Kaynak:** UCI Machine Learning Repository
- **Veri türü:** Görüntü (Image)
- **Sınıf sayısı:** 9
- **Görüntü formatı:** RGB
- **Kullanım amacı:** Görüntü tabanlı atık sınıflandırma

Sınıflar

Veri seti aşağıdaki sınıflardan oluşmaktadır:

1. Cardboard
2. Food Organics
3. Glass
4. Metal
5. Miscellaneous Trash
6. Paper
7. Plastic
8. Textile Trash
9. Vegetation

Bu sınıflar, gerçek hayatta karşılaşılan atık türlerini temsil edecek şekilde seçilmiştir.

Veri Ön İşleme Aşamaları

Modelin daha verimli eğitilebilmesi için veri setine aşağıdaki ön işlemler uygulanmıştır:

- Görüntülerin modele uygun hale getirilmesi
- Yeniden boyutlandırma (224 × 224)
- Normalizasyon işlemleri
- Eğitim – doğrulama – test olarak ayrılması

- Gerekli durumlarda veri artırma (augmentation)

Bu işlemler sayesinde modelin genelleme kabiliyeti artırılmış ve aşırı öğrenme (overfitting) riski azaltılmıştır.

3. Kullanılan Yöntem

Bu projede Transfer Learning yaklaşımı kullanılmıştır. Transfer learning, önceden büyük veri kümeleri üzerinde eğitilmiş modellerin yeni problemlere uyarlanmasını sağlar.

Neden ResNet18?

- Derin mimarisine rağmen aşırı öğrenme riski düşüktür
- ImageNet üzerinde önceden eğitilmiştir
- Eğitim süresi kısadır
- Küçük veri setlerinde yüksek başarı sağlar

Modelin yalnızca son katmanı eğitilmiş, diğer katmanlar dondurulmuştur.

Bu yaklaşım sayesinde:

- Eğitim süresi kısalmış
- Overfitting azaltılmış
- Hesaplama maliyeti düşürülmüştür

4. Model Eğitimi ve Değerlendirme

Model eğitiminde ReLU aktivasyon fonksiyonu, Cross-Entropy kayıp fonksiyonu ve Adam optimizasyon algoritması kullanılmıştır. Görüntüler 224x224 boyutuna yeniden ölçeklendirilmiştir. Eğitim sonucunda yüksek doğruluk oranı elde edilmiştir.

Kullanılan Parametreler

| Parametre | Değer |
|--------------|----------|
| Model | ResNet18 |
| Giriş Boyutu | 224x224 |
| Aktivasyon | ReLU |

Kayıp Fonksiyonu CrossEntropyLoss

| Parametre | Değer |
|---------------|-------------------|
| Optimizasyon | Adam |
| Öğrenme Oranı | 0.001 |
| Eğitim Türü | Transfer Learning |

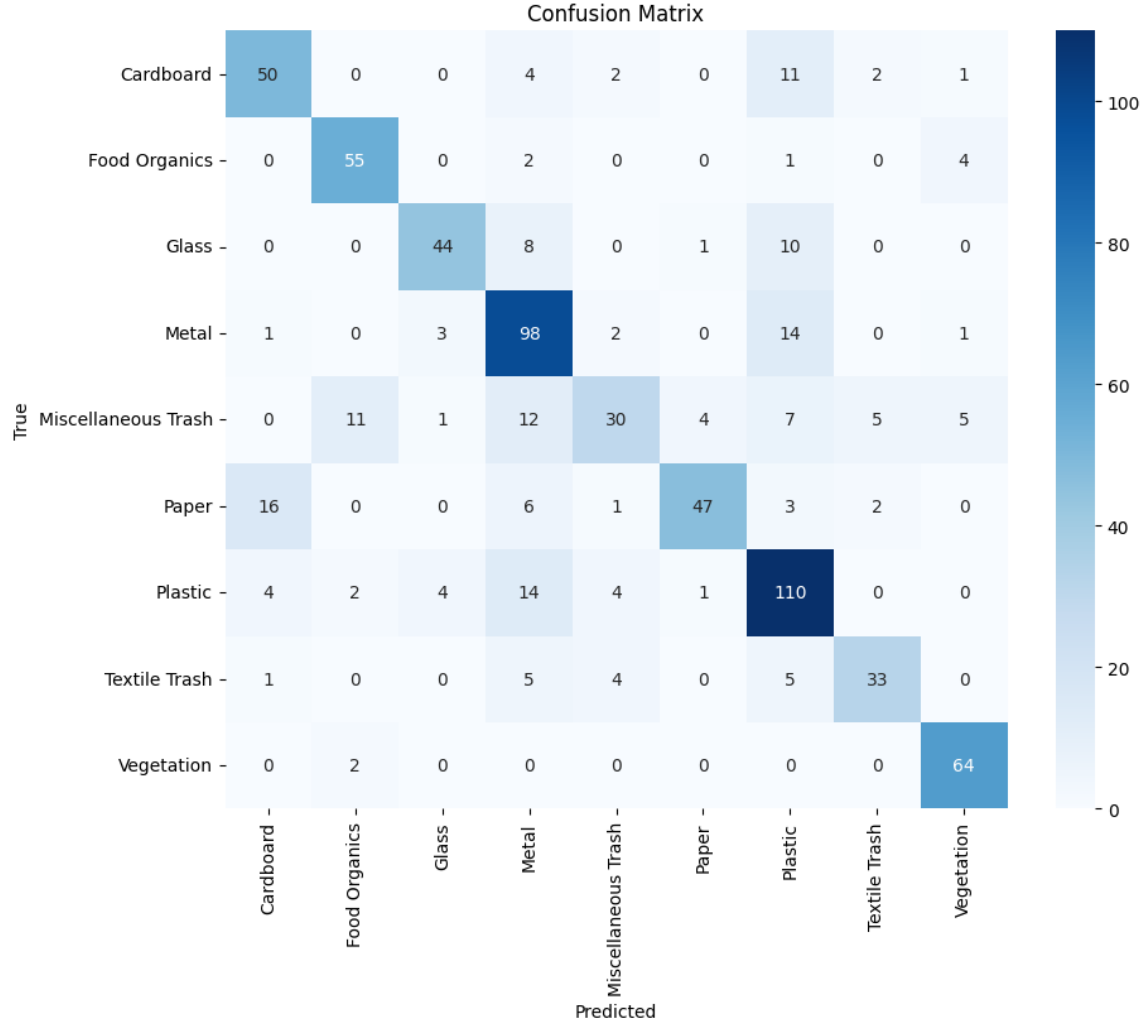
Model eğitimi sırasında yalnızca son katman güncellenmiştir:

Model performansı:

- Doğruluk (Accuracy)
- Confusion Matrix
- Classification Report

kullanılarak değerlendirilmiştir.

| Classification Report: | | | | | |
|------------------------|-----------|--------|----------|---------|--|
| | precision | recall | f1-score | support | |
| Cardboard | 0.69 | 0.71 | 0.70 | 70 | |
| Food Organics | 0.79 | 0.89 | 0.83 | 62 | |
| Glass | 0.85 | 0.70 | 0.77 | 63 | |
| Metal | 0.66 | 0.82 | 0.73 | 119 | |
| Miscellaneous Trash | 0.70 | 0.40 | 0.51 | 75 | |
| Paper | 0.89 | 0.63 | 0.73 | 75 | |
| Plastic | 0.68 | 0.79 | 0.73 | 139 | |
| Textile Trash | 0.79 | 0.69 | 0.73 | 48 | |
| Vegetation | 0.85 | 0.97 | 0.91 | 66 | |
| accuracy | | | 0.74 | 717 | |
| macro avg | 0.77 | 0.73 | 0.74 | 717 | |
| weighted avg | 0.75 | 0.74 | 0.73 | 717 | |



5. Proje Dokümantasyonu ve Klasör Yapısı

Bu proje, eğitim kodlarından model ağırlıklarına, veri kümelerinden arayüze kadar tüm bileşenleri mantıklı ve okunabilir bir yapıda düzenlenmiştir. Proje, GitHub üzerinde versiyon kontrollü olarak paylaşılarak yeniden üretilebilirliği ve sürdürülebilirliği hedeflemektedir.

Proje Klasör Yapısı

Aşağıda projenin GitHub'daki gerçek klasör yapısı gösterilmektedir:

realwaste-deep-learning-project/

|

- └─ data/
- └─ demo_images/
- └─ models/
- └─ notebooks/
- └─ results/
- └─ src/
- └─ Atik_Siniflandirma_Proje_Raporu.pdf
- └─ gradio_app.py
- └─ README.md
- └─ requirements.txt
- └─ .gitignore
- └─ .gitattributes

Klasörlerin ve Dosyaların Açıklaması

data/

Bu klasör, proje kapsamında kullanılan atık görüntülerini içerir.
Veriler eğitim, doğrulama ve test süreçlerinde kullanılır.
Veri ön işleme adımları bu klasör içeriği üzerinden gerçekleştirilir.

demo_images/

Projeyi test etmek ve örnek görsellerle sonuçları göstermek amacıyla kullanılan örnek görüntüler bu klasörde saklanır.

models/

Train edilmiş modellerin ağırlıkları bu klasörde yer alır.
Özellikle son eğitim sonucunda elde edilen en iyi model bu klasörde tutulur.

notebooks/

Bu klasör, model eğitimi, veri keşfi ve deneysel analizlerin yapıldığı Jupyter Notebook dosyalarını içerir.

results/

Modelin eğitimi ve değerlendirilmesine ilişkin çıktılar, grafikler, confusion matrix gibi görseller bu klasörde saklanır.

src/

Projenin temel kaynak kodlarının bulunduğu dizindir.

Bu klasörde model tanımlamaları, yardımcı fonksiyonlar ve eğitim/evaluation kodları bulunur.

gradio_app.py

Bu dosya, projenin kullanıcı arayüzünü sağlayan Gradio uygulamasının ana tanım dosyasıdır.

Modelin web üzerinden test edilebilmesini sağlar.

README.md

Projeye genel bir bakış sunan ve çalıştırma adımlarını açıklayan temel açıklama dosyasıdır.

requirements.txt

Projede kullanılan Python kütüphanelerinin listesi bu dosyada yer alır.

Başka bir makinede projeyi çalıştırmak için:

```
pip install -r requirements.txt
```

şeklinde kurulumu yapılabilir.

.gitignore / .gitattributes

GitHub deposunda izlenmeyecek dosyaları veya ayarları belirlemek için kullanılan yardımcı yapılandırma dosyalarıdır.

6.3 Dokümantasyonun Önemi

Bu düzenli klasör yapısı:

- Projenin anlaşılmasını kolaylaştırır
- Kod tekrar kullanımını sağlar
- Akademik ve endüstriyel standartlara uygundur
- Versiyon kontrol sistemi ile uyumlu çalışır
- Değerlendiricilerin projeyi gözden geçirmesini kolaylaştırır

6. Sonuç

Bu çalışmada, derin öğrenme tabanlı bir sistem kullanılarak atık sınıflandırma problemi başarıyla çözülmüştür. ResNet18 mimarisi sayesinde yüksek doğruluk elde edilmiş, veri artırma teknikleriyle modelin genelleme yeteneği artırılmıştır.

Geliştirilen sistem, geri dönüşüm süreçlerinde otomatik sınıflandırma sistemleri için temel bir altyapı sunmaktadır.

7. Kaynakça

- [1] S. Single, S. Iranmanesh, and R. Raad, "RealWaste," *UCI Machine Learning Repository*, 2023.
doi: 10.24432/C5SS4G.
[Online]. Available: <https://archive.ics.uci.edu/dataset/908/realwaste>
- [2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.
- [3] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.
- [4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Advances in Neural Information Processing Systems (NIPS)*, vol. 25, pp. 1097–1105, 2012.
- [5] PyTorch Developers, "PyTorch Documentation," 2024.
[Online]. Available: <https://pytorch.org/docs/stable/>
- [6] Gradio Team, "Gradio: Create UIs for Machine Learning Models," 2024.
[Online]. Available: <https://www.gradio.app/>
- [7] J. Brownlee, *Deep Learning for Computer Vision*. Machine Learning Mastery, 2019.