

Duruş Bozukluğu Tespiti İçin Veri Seti Oluşturma

Ensar Akbaş
Bilişim Sistemleri Mühendisliği
Kocaeli Üniversitesi
ensar.akbas@gmail.com

Berkay Yüce
Bilişim Sistemleri Mühendisliği
Kocaeli Üniversitesi
berkay.yce10@gmail.com

Özet— Bu rapor, makine öğrenimi ile duruş bozukluklarını sınıflandırmak için bir veri kümesi oluşturma ve ön işleme sürecini incelemektedir. Veri kümesi, Google Görseller’den otomatik veri çıkarımı ile oluşturulmuş ve ardından model performansını artırmak için ön işleme ve veri artırma teknikleri uygulanmıştır.

Anahtar Kelimeler— veri seti, veri ön işleme, veri artırma

I. Veri Seti Nedir ?

Veri seti, belirli bir amaç için toplanmış ve birbiriyle ilişkili olan verilerin bütünüdür. Çoğunlukla yapay zeka ve makine öğrenmesi projelerinde kullanılır ve modelin eğitilmesi için gerekli bilgileri içerir. Bir veri seti, sınıflandırma gibi görevlerde kullanılması için belirli özellikler taşıyan örnekleri veya kayıtları içerir. Bu veriler, doğru sınıflandırma yapılabilmesi ve modelin başarısının ölçülebilmesi için dikkatli bir şekilde seçilmelidir.

Projemizde, veri setimizi Google Görseller’den elde edilen görüntülerle oluşturduk ve daha sonra bu veriler üzerinde ön işleme ve veri artırma (data augmentation) işlemleri gerçekleştirdik.

II. Sınıflandırma Problemi Nedir ?

Sınıflandırma, makine öğrenmesinde yaygın olarak kullanılan ve veriyi belirli kategorilere veya sınıflara ayırmayı amaçlayan bir problem türüdür. Sınıflandırma problemlerinde bir model, verilen veri üzerinde eğitilir ve daha sonra yeni verileri doğru sınıflara ayırmak için kullanılır. Örneğin, projemizdeki sınıflandırma probleminde, pektus deformitelerinin farklı türlerini görüntüler aracılığıyla doğru şekilde tanıyacak bir model geliştirmek hedeflenmiştir. Sınıflandırma problemleri, sağlık, finans, güvenlik gibi birçok alanda farklı verilerle çalışmak için temel bir görev olarak kabul edilir.

III. Selenium Nedir Nasıl Kullandık ?

Selenium, web tarayıcılarını otomatikleştirmek için kullanılan bir araçtır ve Python gibi programlama dilleri ile entegre çalışabilir. Veri toplama süreçlerinde, Selenium’u kullanarak Google Görseller üzerinden belirli anahtar kelimeler ile arama yapıp, görselleri indirme işlemini gerçekleştirdik. Sayfa içindeki görselleri bulmak için CSS seçicileri ve XPATH gibi seçicilerden faydalandık. Ardından, görselleri sayfanın altına doğru kaydırarak yeni görsellerin yüklenmesini sağladık. Bu görsellerin URL’lerini topladık ve HTTP üzerinden indirdik. Böylece, belirli bir konuya dair geniş bir veri seti oluşturarak, sınıflandırma

projeleri için görsel veriyi hazırladık. Selenium için gerekli kütüphaneleri aşağıdaki gibi kurduk. Ayrıca dosya işlemleri için ‘os’ modülünü de projemiz içine dahil ettik.

```
from selenium import webdriver
from selenium.webdriver.common.by import By
import requests
import os
import time
```

IV. Pillow Nedir, Nasıl Kullandık ?

Pillow, Python’da görüntü işleme ve düzenleme için popüler bir kütüphanedir. Görsellerin boyutlandırılması, kontrast, parlaklık, keskinlik gibi özelliklerin ayarlanması ve filtrelerin uygulanması gibi işlemleri kolaylaştırır.

Projede, Pillow ile indirilen görseller üzerinde veri ön işleme (data preprocessing) yaptık. Görselleri yeniden boyutlandırarak model için daha uygun hale getirdik. Ayrıca, bu kütüphane sayesinde farklı renk ve parlaklık düzenlemeleri yaparak verilerimizi çeşitlendirdik. Pillow’un sağladığı işlevler sayesinde veri setimizin kalitesini ve homojenliğini artırdık. Pillow’u kullanabilmemiz için gerekli kütüphaneleri aşağıdaki gibi projemiz içine dahil ettik. Burada da dosyalama işlemlerini kullandığımız için ‘os’ modülünü tekrardan eklemiş olduk.

```
from PIL import Image, ImageEnhance, ImageColor, ImageFilter, UnidentifiedImageError
import os
```

V. Veri Ön İşleme Nedir ?

Veri ön işleme (data preprocessing), ham veriyi model eğitimi için hazırlama sürecidir. Görsellerdeki gürültüyü azaltmak, boyutları düzenlemek ve görsel formatı standardize etmek gibi işlemleri içerir.

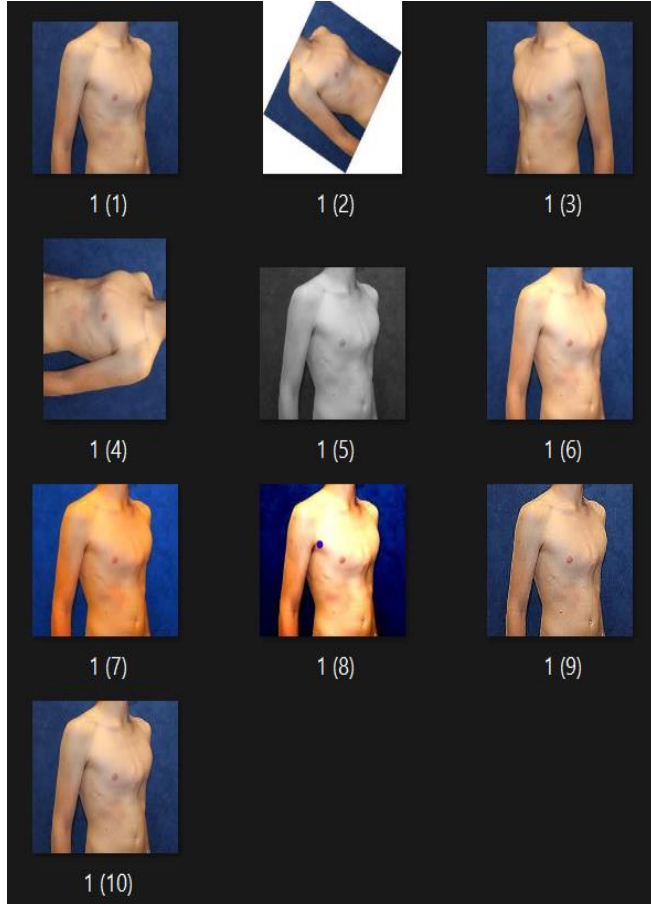
Bu projede, veri ön işleme ile görsellerin boyutlarını eşitleyerek modelin daha tutarlı bir şekilde öğrenmesini sağladık. Ayrıca, bu süreçte konuyla alakası olmayan verileri ayıklayarak veri setimizin doğruluğunu artırdık. Veri ön

işleme adımları, modelin performansını iyileştiren ve eğitim sürecini optimize eden temel adımlardan biridir.

VI. Veri Artırma (Data Augmentation) Nedir ? Nasıl Kullandık ? Önemi Nedir?

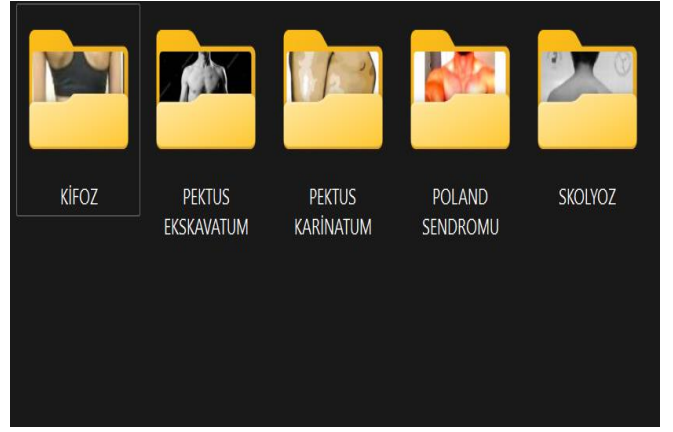
Veri artırma (data augmentation), mevcut veri setini artırmak için kullanılan bir tekniktir. Görseller üzerinde döndürme, çevirme, parlaklık ayarı, zoom gibi farklı işlemler uygulanarak yeni veriler oluşturulur.

Bu projede, Pillow kütüphanesi ile data augmentation işlemleri gerçekleştirdik ve görsellerin çeşitliliğini artırdık. Bu işlem, modelin veri seti üzerindeki öğrenme kapasitesini genişletir, aşırı öğrenme (overfitting) riskini azaltır ve modelin daha çeşitli verilerle eğitim almasını sağlar. Data augmentation, özellikle sınırlı sayıda veri ile çalışılan durumlarda modelin performansını büyük ölçüde artıran bir adımdır.



VII. Veri Setimiz

Bu projede, görsel verilerin düzenlenmesinde ve çeşitlendirilmesinde kullanılan işlemler, modelin doğruluk ve genelleme kapasitesini artırmak adına yapılan adımlardır. Bu aşamalarda kullanılan teknikler ve kütüphaneler sayesinde elde edilen veri seti, makine öğrenmesi modelinin daha doğru ve kapsamlı sonuçlar vermesine katkıda bulunacaktır.



KAYNAKÇA

- [1] [Selenium WebDriver with Python Tutorial - javatpoint](#)
- [2] [Selenium with Python — Selenium Python Bindings 2 documentation](#)
- [3] [Selenium İle İnternette Veri Çekme | by Ömer Şenol | Medium](#)
- [4] [Python Pillow Tutorial](#)
- [5] <https://www.youtube.com/watch?v=NB8OceGZGjA>
- [6] [Image Manipulation In Python Using Pillow | Edit Images Using Python | Python Tutorial | Simplilearn](#)
- [7] [Python ve Selenium ile web test otomasyonu-02: İnternet tarayıcı ile temel fonksiyonlar](#)
- [8] [Python Image Editor Using Python - GeeksforGeeks](#)
- [9] <https://www.youtube.com/watch?v=Xjv1sY630Uc&list=PLzMcBGfZo4-n40rB1XaJ0ak1bemv1qumQ>

- [Drive Linki](#)
- [Github Linki](#)