

# Sesli Duygu Analizi Projesi

Eren Şahyılmaz

Kocaeli Üniversitesi

Teknoloji Fakültesi

Bilişim Sistemleri Mühendisliği

201307003

erensahyilmaz@gmail.com

Murat Öztürk

Kocaeli Üniversitesi

Teknoloji Fakültesi

Bilişim Sistemleri Mühendisliği

211307082

murat5506furkan@gmail.com

## ÖZET

Bu araştırmada sesli duygu analizi için bir veri seti oluşturmak amaçlanmış olup çeşitli kaynaklardan toplanan ses kayıtlarını kullanarak kapsamlı bir veri hazırlığı süreci öngörülmektedir. Bu süreçte, toplanan ses kayıtlarının temizlenmesi, parçalara ayrılması gibi ön işleme adımları yer almaktadır. Python programlama dilinde yer alan kütüphaneler kullanılmıştır. Youtube'da bulunan videolarla hazırlanan veriler; mutluluk, öfke,üzgünlük ve nötr olarak sınıflara ayrılmıştır. Projenin veri toplama adımı tamamlandıktan sonra daha önce bahsedilen 4 sınıfın tespiti için bir makine öğrenmesi projesinin yapılması hedeflenmektedir.

**Anahtar kelimeler:**Python, sesli duygu analizi,Veri

## 1. VERİ KEŞFİ

4 adet sınıf için veri toplanmak amaçlanmıştır. Mutluluk, öfke, üzgünlük ve nötr. Öncelikle yapılması gereken hangi kaynaktan veri toplanacağını belirlemek gerekmektedir. Bu projede Youtube üzerinden veri toplanmıştır.

### 1.1 Kullanılan Kütüphaneler

#### 1.1.1 Selenium

Selenium, web tarayıcılarının otomasyonunu sağlayan ve destekleyen bir dizi araç ve kütüphanenin birleştirildiği bir çatı projesidir. Tarayıcılarla kullanıcı etkileşimini taklit etmek için uzantılar, tarayıcı dağıtımı için bir dağıtım sunucusu ve tüm büyük web tarayıcıları için değiştirilebilir kod yazmanıza olanak tanıyan W3C WebDriver spesifikasyonunun uygulamaları için altyapı sağlar. [1]

**from selenium import webdriver** kodu ile WebDriver kütüphanesini içe aktarır. Selenium WebDriver, web tarayıcılarını otomatikleştirmek için kullanılan güçlü bir araçtır.

**driver = webdriver.Chrome()** ile birlikte driver, Chrome WebDriver nesnesi oluşturulur. Bu nesne Chrome tarayıcısını kontrol eder. **driver.get** ile sayfa url'si alınır.

**driver.quit()** ile WebDriver nesnesi sonlandırılır.

### 1.1.2 BeautifulSoup

BeautifulSoup, HTML ve XML dosyalarından veri çekilmesini sağlayan bir Python kütüphanesidir. [2]

BeautifulSoup ( **driver.page\_source**, 'html.parser'): Bu satır, WebDriver nesnesi ile elde edilen sayfa kaynağını alır ve BeautifulSoup nesnesi olarak parse eder. Bu sayede, HTML yapısı içindeki verilere kolayca erişilebilir. **find\_all()** metodu sayesinde bir HTML veya XML belgesinde belirtilen özelliklere sahip tüm etiketleri bulunabilir **find()** ise belirli bir özellik istenir.

### 1.1.3 yt\_dlp

Bu kütüphane ile Youtube sitesi üzerinden medya indirme işlemleri yapılır.

## 1.2 Yöntem

Bu projede Youtube arama motoru Selenium ile yüklenmiştir. Ardından BeautifulSoup ile aramada çıkan her bir videoya ait başlık ve link alınmıştır. Ardından yt\_dlp kütüphanesi ile bu linke sahip videolar .webm formatında indirilmiştir.

## 2. VERİ ÖN İŞLEME

### 2.1 Kullanılan Kütüphaneler

#### 2.1.1 os

os modülü, işletim sistemine bağlı işlevsellikleri taşınabilir bir şekilde kullanmanın bir yolunu sağlar.. [3] Kısacası os modülü ile dosya işlemleri gerçekleşir. Dosya oluşturma veya dosyaya erişme gibi işlemler bu modül ile yönetilir. **makedirs()** fonksiyonu ile belirtilen yol (klasör/dizin) oluşturur. **listdir()** fonksiyonu, belirtilen bir dizindeki tüm dosya ve klasörlerin isimlerini bir liste olarak döndürür. **path.join()**, birden fazla dizin ve/veya dosya adını birleştirerek işletim sistemine uygun bir yol (path) oluşturur.

#### 2.1.2 pydub

Basit kolay ve üst seviye kullanımı ile ses dosyalarının manipülasyonunun yapıldığı kütüphanedir. [4] AudioSegment

ile ses dosyaları açılır ve dosya üzerinde pek çok düzenleme yapılabilir.

### 2.1.3 spleeter

Spleeter, Deezer tarafından geliştirilmiş, Python ile yazılmış ve Tensorflow kullanan, önceden eğitilmiş modellere sahip bir kaynak ayrıştırma kütüphanesidir.[5] Spleeter, ses dosyalarındaki enstrümanları ve vokalleri otomatik olarak ayırmak için kullanılır.

### 2.1.4 Shutil

shutil modülü, dosyalar ve dosya koleksiyonları üzerinde çeşitli üst düzey işlemler sunar. Özellikle, dosya kopyalama ve silme işlemlerini destekleyen işlevler sağlar. [6] shutil.copy() ile dosya kopyalama, shutil.move() ile taşıma yapılmaktadır; shutil.rmtree() ile bir dizin ve altındaki tüm dosyalar silinmektedir. Özellikle dosya ve klasör yönetimi gerektiren otomasyon görevlerinde veya yedekleme işlemlerinde çokça kullanılır.

### 2.1.5 subprocess

subprocess modülü ile yeni süreçler başlatılabilir, bu süreçler giriş/çıkış/hata akışlarına bağlanabilir ve dönüş kodlarını gösterir. [7] Bu modül, komut satırı komutlarının veya dış programların Python'dan çalıştırılmasını ve yönetilmesini sağlar.

### 2.1.6 ffmpeg

ffmpeg, multimedya verilerini işlemek için kullanılan açık kaynaklı ve ücretsiz bir yazılım kütüphanesidir. [8]

## 2.2 Yöntem

### 2.2.1 Ses Temizliği

spleeter modülü ile vokal sesleri ve enstrümanları ayırmak mümkündür. Daha önce belirtildiği gibi indirilen videolar öncelikle 'belirsiz' adı verilen bir klasöre kaydedilir. Daha sonra da vokal ve enstrüman seslerinin ayrılması işleminin yapılması gerekmektedir.

YoutubeDL kütüphanesi ile indirilen videolar '.webm' formatında inmiştir. İndirilen videolar çeşitli dizi/filmlerden alınmıştır, dolayısıyla bir dizi işleme tabi tutulması gerekmektedir.

ffmpeg ile bu ses dosyaları '.wav' formatına çevrilmiştir. subprocess ile klasörde bulunan tüm dosyalara aynı işlem uygulanmıştır. os ile kütüphaneler çekilmiştir ardından da spleeter içinde bulunan seperator fonksiyonu ile vokaller ve enstrümanlar ayrılmıştır. os ile aynı dizinde bir klasör açılmış ve vokal ve diğer sesler kaydedilmiştir.

### 2.2.2 Toplama

Klasörler içine kaydedilen vocal dosyalarını bir dosyada toplamak gerekmektedir. os ve shutil kütüphaneleri ile bu işlem mümkün olmaktadır.

for root, dirs, files in os.walk(source\_directory):

komutu ile dizin taranır. Root; dizinin yolunu, dirs; alt klasörlerin listesini ve files ;dizin içinde yer alan dosyaları liste halinde döndürür.

Bu sayede 'vocals.wav' dosyaları aranır ve isim olarak önünde klasörünün ismi de eklenecek şekilde geri döndürür.

### 2.2.3 Bölme

Elde edilen ses dosyaları önce süreleri 30sn'lik olmak üzere parçalanır. Bunun için pydub kütüphanesine ait AudioSegment kullanılır. AudioSegment, ses dosyalarını alır ve üzerinde işlem yapılmasına olanak sağlar. Burada sesler analiz edilecektir. Sesler analiz edilip 4 farklı kategoriye ayrılacaktır. Daha sonra 4 farklı kategoriye ait veriler 5 sn'lik daha küçük parçalara bölünecektir.

### 2.2.4 Temizlik

Bu aşamada, elde edilen ses dosyalarının kalite kontrolü ve gereksiz verilerin ayıklanması işlemi gerçekleştirilmiştir. pydub kütüphanesi ile ses dosyaları analiz edilerek, belirli bir sessizlik eşiği ve minimum sessizlik süresi kullanılarak seslerin içeriklerine göre ayrıştırılması yapılmıştır. Bu süreçte, tamamen sessiz olan veya ses içermeyen dosyalar tespit edilip farklı bir klasöre taşınmıştır. Bu sayede veri seti daha düzenli hale getirilmiş ve kullanılabilir veriler ayıklanarak içeriklerin doğruluğu sağlanmıştır. Dosyaların taşınması ve yeniden düzenlenmesi için os ve shutil kütüphaneleri kullanılmıştır. Bu işlemin sonucunda, ses dosyalarının analiz ve sınıflandırma süreçlerine daha verimli şekilde dahil edilmesi sağlanmıştır.

## 3. ZORLUKLAR VE ÇÖZÜMLERİ

### 3.1 Veri Toplama sürecindeki Zorluklar

Veri toplama aşamasında, özellikle YouTube üzerinden veri çekme işlemi sırasında çeşitli zorluklarla karşılaşıldı. YouTube'dan video verisi çekmek için kullanılan Selenium ve BeautifulSoup araçları, bazen dinamik içerik yükleme nedeniyle beklenmedik şekilde çalışabiliyordu. Ayrıca, video başlıkları ve bağlantılarını almak için doğru HTML etiketlerinin seçilmesi zaman alabiliyordu.

**Çözüm:** Bu sorunu aşmak için, Selenium ve BeautifulSoup'un birlikte kullanıldığı ve dinamik içeriklerin doğru şekilde yüklenmesini sağlayan bir çözüm geliştirilmiştir. Ek olarak, video başlıklarını ve bağlantılarını doğru şekilde elde etmek için HTML etiketlerinin özelleştirilmiş yolları kullanılmıştır.

### 3.2 Ses Dosyalarının Ayrıştırılması ve Temizlenmesi

Videolardan ses çıkarma işlemi sırasında, seslerin doğru şekilde ayrıştırılması ve temizlenmesi önemli bir zorluktu. Spleeter kütüphanesi ile vokaller ve enstrümanlar ayrılmaya çalışılsa da, bazı ses dosyalarında beklenmedik hatalar ortaya çıkabiliyordu. Özellikle, ses dosyalarındaki arka plan gürültüsü ve istenmeyen öğeler, analiz ve sınıflandırma süreçlerinde sıkıntılar yaratabiliyordu.

**Çözüm:** Bu sorunu çözmek için, ses dosyalarını temizlemek amacıyla ek filtreler ve ses eşiği kullanılmıştır. pydub kütüphanesi ile sesin içerik analizi yapılarak, düşük kaliteli ses dosyaları tespit edilip temizlenmiştir. Ayrıca, ffmpeg

kullanılarak ses dosyaları dönüştürülüp, ses kalitesinin daha yüksek olması sağlanmıştır.

### 3.3 Veri Setinin Düzenlenmesi ve Bölünmesi

Veri setinin farklı duygu sınıflarına ayrılması ve daha küçük parçalara bölünmesi sırasında, ses dosyalarının doğru bir şekilde etiketlenmesi ve parçalara ayrılması zorluk oluşturmıştır. Özellikle, her bir duygu için uygun ses segmentlerinin seçilmesi ve doğru etiketlemesi karmaşık bir işlem olabilmektedir.

**Çözüm:** Bu zorluk için, ses dosyalarını 30 saniyelik parçalara ayıran ve daha sonra bu parçaları 5 saniyelik daha küçük segmentlere bölen bir algoritma geliştirilmiştir. Bu şekilde, her bir duygu sınıfı için yeterli sayıda örnek sağlanarak, daha doğru bir analiz yapılması mümkün olmuştur. Ayrıca, veri temizliği sırasında pydub ve shutil kütüphaneleri ile gereksiz dosyalar ve hatalı veriler temizlenmiştir.

### 3.4 Çoklu Kütüphane ve Araç Kullanımı

Proje, farklı Python kütüphanelerinin entegrasyonunu gerektirdiği için, bu kütüphaneler arasında uyumsuzluklar yaşanabilmektedir. Özellikle, Spleeter, yt-dlp, ffmpeg, ve pydub gibi araçların bir arada çalıştırılması sırasında kütüphane bağımlılıkları ve sürüm uyumsuzlukları gibi sorunlar ortaya çıkmıştır.

**Çözüm:** Bu tür sorunları aşmak için, proje başında her bir kütüphanenin uyumlu sürümleri belirlenmiş ve sanal ortam kullanılarak bağımlılıklar izole edilmiştir. Bu sayede, her kütüphanenin doğru sürümü kullanılarak, tüm araçların uyumlu bir şekilde çalışması sağlanmıştır.

## 4. KAYNAKÇA

[1] “The Selenium Browser Automation Project,” *Selenium*. <https://www.selenium.dev/documentation/>

[2] L. Richardson, “Beautiful Soup Documentation — Beautiful Soup 4.4.0 documentation,” *Crummy.com*, 2019. <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>

[3] Python Software Foundation, “os — Miscellaneous operating system interfaces — Python 3.8.0 documentation,” *Python.org*, 2019. <https://docs.python.org/3/library/os.html>

[4] J. Robert, “pydub: Manipulate audio with an simple and easy high level interface,” *PyPI*. <https://pypi.org/project/pydub/>

[5] deezer, “deezer/spleeter,” *GitHub*, Jan. 27, 2020. <https://github.com/deezer/spleeter>

[6] “shutil — High-level file operations — Python 3.7.4 documentation,” *Python.org*, 2010. <https://docs.python.org/3/library/shutil.html>

[7] Python, “subprocess — Subprocess management — Python 3.8.5 documentation,” *docs.python.org*. <https://docs.python.org/3/library/subprocess.html>

[8] “FFmpeg,” *Ffmpeg.org*, 2024. <https://www.ffmpeg.org>