

# Rap Türlerinin Sınıflandırılması

Cengizhan Durmuş  
Kocaeli Üniversitesi  
Bilişim Sistemleri Mühendisliği  
201307087  
201307087@kocaeli.edu.tr

## I. GİRİŞ

Ses sınıflaması, makine öğrenimi ve uygulamalı bilgisayar bilimleri için en önemli araştırma konularından biridir. Bu çalışma sesleri kullanarak rap türlerini sınıflandırma yöntemi sunmaktadır.

Herkese açık sesler kullanılarak, boombap, jazz rap, old school, politic rap ve trap türlerine ait sesler YouTube üzerinde indirildi. Bu dosyalar farklı tarihlerde elde edilip indirilerek türdeki her ses dosyası için 15 saniyelik kesitler üretilmiştir. Bu şekilde, veri kümemizde toplam 2465 dosya elde edildi.

## II. VERİ KÜMESİNİN TOPLANMASI

YouTube'dan ses dosyalarını indirmek için öncelikle python kütüphanelerini yüklüyoruz. Youtube\_dl kütüphanesini video indirme işlemi için, Os kütüphanesini dosya üzerindeki işlemler için (mp3 veya wav türünde kaydetme gibi), Shutil kütüphanesini de ses dosyasını istenilen konuma kopyalama işlemi için kullanılacak.

```
import youtube_dl
from youtube_dl import YoutubeDL
import os
import shutil
```

5 farklı türde ki indirilecek ses dosyalarının linklerini her tür için farklı 'txt' dosyası içerisine kopyalayıp kaydediyoruz.

```
class youtube_download:
    def audio(destinationpath):
        file1 = open("olschool_list.txt", 'r')
        x = (file1.readlines())
```

Kod kısmında Youtube\_download isminde sınıf oluşturup, içerisine de audio adında fonksiyon tanımlıyoruz. Ardından oluşturduğumuz txt dosyalarından birini okumak için file isminde değişken tanımlamak için import ettiğimiz os kütüphanesini kullanıyoruz. Son satırda ise txt

dosyasında ki satırları tek tek okuyup x değişkenine aktarıyoruz.

İndirmek istediğimiz Youtube videosunun Kbps yani saniyede aktarılan bit sayısı, dosya formatı (mp3, wav vs.) gibi videonun tüm özelliklerini içeren params adında değişken oluşturduk.

```
params = {
    'format': 'bestaudio/best',
    'postprocessors': [{
        'key': 'FFmpegExtractAudio',
        'preferredcodec': 'wav',
        'preferredquality': '128',
    }],
}
```

Üstte oluşturulan params değişkeni ve import ettiğimiz youtube\_dl kütüphanesini kullanarak txt dosyasındaki linkleri wav dosyasına dönüştürecek döngü oluşturduk.

```
for y in x:
    print(y)
    if y != '\n':
        youtube =
youtube_dl.YoutubeDL(params)
youtube.download([str(y)])
```

Altındaki kod satırlarında ise tüm wav linklerini alıp, import edilen os.path kütüphanesi yardımıyla belirlenen dosya dizinine taşıyoruz.

```
sourcepath = os.getcwd()
sourcefiles = os.listdir(sourcepath)
for file in sourcefiles:
    if file.endswith('.wav'):
        shutil.move(os.path.join(sourcepath,
        file),
        os.path.join(destinationpath, file))
file1.close()
```

SourcePath isminde değişken oluşturduk ve dosyaları kaydetmek istediğimiz konumu aktardık.

### III. VERİLERİN KESİLMESİ

İndirilen ses dosyalarını içe aktarmak için AudioSegment sınıfındaki Pydub kütüphanesini projemize aktif ettik. Matematiksel işlevleri kullanmak için de Math kütüphanesini yükledik.

```
from pydub import AudioSegment
import math
```

Kesilen ses dosyasını export aşamasında isimlendirirken kullanmak için noun adında sayaç oluşturuyoruz. Kesilecek ses dosyalarını klasörde dolaşmak için for döngüsü oluşturduk. Yüklediğimiz AudioSegment kütüphanesi sayesinde ses dosyalarının bulunduğu klasör yolunu ilgili değişkene atadık.

```
noun = 1
for i in range(1, 530):
    song =
    AudioSegment.from_wav("./music_wav/p
    olitic/politic "+str(i)+").wav")
    duration = song.duration_seconds
```

Yol üzerinden ilgili ses dosyasına ulaşım song.duration\_seconds ile ses dosyasının uzunluğunu elde ettik ve değişkene atadık. 2. For döngüsünde her ses dosyasını 15 saniyeye bölmek için tüm dosyaları dolaştık.

```
for j in range(int(duration/15)):
    split_audio =
    song[(15*j)*1000:15*(j+1)*1000]

    split_audio.export("./music_wav/poli
    tic/cut/politic"+str(i)+".wav",
    format="wav")

    print("./music_wav/politic/cut/polit
    ic"+str(counter)+".wav")
    noun=noun+1
```

split\_audio.export ile de kesilen dosyaları çıkaracağımız yolu belirledik. Print ile de konsola yazdık ve en son sayacı 1 arttırdık.

Bu çalışmada rap türlerinin sınıflandırması için 1 adet veri kümesi kullanılmıştır. Aşağıdaki tabloda veri kümesindeki toplam ses dosyası sayısı Boom

bap, Old school, Jazz rap, Politik rap ve Trap türlerin de bulunan veri sayısı gösterilmiştir.

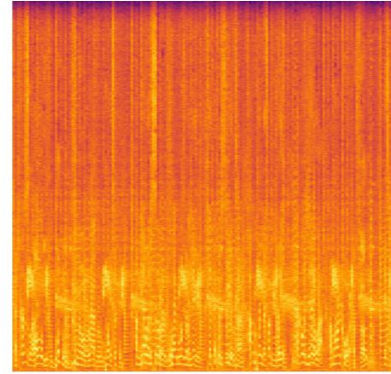
### Veri Kümesi

Sınıf	Dosya Sayısı
Boom Bap	480
Jazz Rap	504
Old School	493
Politic Rap	506
Trap	482
<b>Toplam</b>	<b>2465</b>

### IV. ÖZNİTELİK ÇIKARILMASI

#### A. Spektrogram

Spektrogram, belirli bir dalga biçiminde bulunan çeşitli frekanslarda zaman içinde bir sinyalin sinyal gücünü veya "yüksekliğini" temsil etmenin görsel bir yoludur. Bir spektrogram genellikle bir ısı haritası olarak, yani rengin veya parlaklığın değiştirilmesiyle gösterilen yoğunluğa sahip bir görüntü olarak tasvir edilir.



Şekil 1: boombap129.png dosyasına ait spektrogram görüntüsü

Spektrogram, belirli bir dalga formunda bulunan çeşitli frekanslarda bir sinyalin sinyal gücünü veya yüksekliğini temsil eden görseldir. Aynı zamanda enerji seviyelerini zaman içinde nasıl değiştiğini de gösterir.

#### B. Sıfır Geçiş Oranı (Zero Crossing Rate)

Bir sinyalin düzgünlüğünü ölçmenin çok basit bir yolu, o sinyalin bir segmenti içindeki sıfır geçiş sayısını hesaplamaktır. Zero crossing rate bir sinyalin sıfır çizgisinden geçişi yani işaret değişiminin oranıdır.

### C. Mel-Frekans Katsayılar (MFCC)

Mel frekans ölçeği, insan kulağının ses frekanslarındaki değişimi algılayışını gösteren bir ölçektir. MFCC (Mel-Frequency Cepstral Coefficients), ses sinyalinin kısa zamanlı güç spektrumunun Mel ölçeği üzerindeki ifadesidir. İnsan sesinin özelliklerini modeller.

### D. Spektral Merkez (Spectral Centroid)

Spektral Centroid, bir spektrumun enerjisinin hangi frekansta merkezlendiğini veya başka bir deyişle bir sesin “kütle merkezinin” nerede olduğunu gösterir. Spektrumun kütle merkezinin nerede olduğunu gösterir.

### E. Chroma Frekansı (Chroma Frequency)

Bir kroma özelliği veya vektörü , tipik olarak, sinyalde her perde sınıfının ne kadar enerji bulunduğunu gösteren 12 elemanlı bir özellik vektörüdür. Müzik parçaları arasındaki benzerlik ölçüsünü tanımlamayı sağlar

```
import librosa
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import os
from PIL import Image
import pathlib
import csv
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler
from keras import layers
from keras import layers
import keras
```

Ses dosyalarının spektrogram ve özelliklerini çıkartmak için yukarıdaki kütüphaneleri import ettik.

```
cmap = plt.get_cmap('inferno')
plt.figure(figsize=(8,8))
genres = 'boombap jazzrap oldschool
politic trap'.split()
for g in genres:

pathlib.Path(f'img_music/{g}').mkdir(
    parents=True, exist_ok=True)
    for filename in
os.listdir(f'./drive/My
Drive/genres/{g}'):
    songname = f'./drive/My
Drive/genres/{g}/{filename}'
    y, sr =
librosa.load(songname, mono=True,
duration=15)
    plt.specgram(y, NFFT=2048,
Fs=2, Fc=0, noverlap=128, cmap=cmap,
sides='default', mode='default',
scale='dB');
    plt.axis('off');

plt.savefig(f'img_music/{g}/{filename
e[:-3].replace(".", "")}.png')
plt.clf()
```

Yukarıdaki kodda genres dosyası altındaki tüm ses dosyalarının spektrogram görselleri işlenip png dosyası olarak img\_music dizini altında kaydedildi.

```
header = 'filename chroma_frequency
sp_centroid zero_crossing_rate mfcc'
header += ' label'
header = header.split()
```

Oluşturulacak olan .csv dosyasının başlıkları header değişkeniyle tanımlandı.

```
file = open('dataAudio.csv', 'w', ne
wline='')
with file:
    writer = csv.writer(file)
    writer.writerow(header)
    genres = 'boombap jazzrap oldschool
politic trap'.split()
    for g in genres:
        for filename in os.listdir(f'./d
rive/My Drive/genres/{g}'):
            songname = f'./drive/My Driv
e/genres/{g}/{filename}'
            y, sr = librosa.load(songnam
e, mono=True, duration=15)
```

```

        chroma_stft = librosa.feature.
e.chroma_stft(y=y, sr=sr)
        spec_cent = librosa.feature.
spectral_centroid(y=y, sr=sr)
        zcr = librosa.feature.zero_c
rossing_rate(y)
        mfcc = librosa.feature.mfcc(
y=y, sr=sr)
        to_append = f'{filename} {np
.mean(chroma_stft)} {np.mean(spec_ce
nt)} {np.mean(zcr)} {np.mean(mfcc)}
{g}'
        file = open('dataAudio.csv',
'a', newline='')
        with file:
            writer = csv.writer(file
)
            writer.writerow(to_appen
d.split())

```

Öncelikle .csv dosyası içerisine header değişkeninden başlık isimlerimizi alıp çekiyoruz. Google Drive da bulunan genres klasörünün altındaki türleri ilk for döngüsünde, türler altındaki ses dosyalarını ikinci for da dolaştık.

Sıfır geçiş oranı, Mel frekansı, Chroma frekansı, Spectral merkez özellikleri kaynakça da belirtilen site yardımıyla hesaplandı ve ilgili değişkene atandı.

Son olarak da .csv dosyası açılarak her ses dosyasına ait özellik sonuçlarına kod yardımıyla yazıldı.

## KAYNAKLAR

- [1] Sesin Öznelik Çıkarımı,  
<https://medium.com/datarunner/librosa-9729c09ecf7a>
- [2] Audio Classification Project Using Deep Learning,  
<https://www.youtube.com/watch?v=mHPpCXqQd7Y>, (Erişim Tarihi: 31 Mart 2021)