Kök Nota Analizi

ALPEREN OVALI

231307086

Alperen.oval@gmail.com

**Özet—** **Bu proje, müzikteki "la", "si", "fa" ve "sol" notalarının ses verileri kullanılarak otomatik olarak sınıflandırılmasını amaçlamaktadır. Müzik notalarının otomatik tanınması, müzik eğitimi, akor analizi, ses işleme ve diğer benzeri birçok farklı alanda kullanışlı olabilir.**

***Anahtar Kelimeler—Kök Nota, Looperman, Python, Librosa , Numpy, Pandas, Scikit-learn, WAV Files, Nota Sınıflandırma, Müzik işleme, Kök Nota Tespiti, Ses Analizi, Akustik Özellik Çıkarımı, MFCC değerleri , CSV, SVM***

# I. Tanıtım

Müziğin nota seslerini otomatik tanıma sistemi müzik eğitimi ve analizi ile ilgili yapay zekanın sunduğu yenilikçi olanağıdır. İlgili bu alandaki projelerin, müzik alanında öğrenim görmüş olan kişilerin desteklenmesinin yanı sıra, müzik programları ve ses işleme teknolojileri için de önemli bir araştırma alanı oluşturduğu söylenebilir.

Bu projede müzikte en çok kullanılan "la", "si", "fa", "sol" notalarının ses verilerinden otomatik tanıma ve sınıflandırma gerçekleştirilmiştir. MFCC tanım iki aşamalı mimari ile detaylı bir açıklamadan sonra Destek Vektör Makineleri (SVM) algoritmasına yönlendirilmiş bir makine öğrenim modeli geliştirilmiştir. Uygulama ses işleme ve sınıflandırma ile ilgili problemlerinde sıklıkla başvurulan yüksek doğruluk başarıları ile bilinen bir yöntemdir.

Projenin temel amacı bir sisteme, müzik notalarının tanınmasını sağlayıp bunu müzik eğitimine ve ses tanıma uygulamalarına kullanma imkanını araştırmaktır.

II. TEKNOLOJİLER VE KULLANILAN YÖNTEMLER

1.Python: Zengin kütüpaneleri ve veri işleme seçeneklerinden dolayı ana programlama dili olarak kullanılmıştır. Pip özelliği sayesinde çeşitli kütüphaneler indirilip Python syntax’ında yazılmıştır.

2.Librosa: Pip üzerinden indirilmiştir .Ses dosyalarından MFCC analizi çıkarılırken Librosa dokümatasyonundan faydalanılmıştır.

TOLGA BOZ

211307036

Tolgaboz350@gmail.com

3.Scikit-Learn: Veri ayırma, verileri test etmek amacıyla Scikit Learn kütüphanesinden SVM modeli kullanılmıştır.

4.SVM: Hem sınıflandırma hem de regresyon problemlerinde kullanılan güçlü bir denetimli makine öğrenme algoritmasıdır.

5.Pandas: Veri işleme ve CSV dosyalarına MFCC değerlerini yazmak amacıyla kullanılmıştır.

6-Numpy: Sayısal işlemler ve matris manipülasyonları için kullanılmıştır.

7-Visual Studio Code: Python kütüphanelerinin import edilmesi ve Python kodlarının yazılıp derlenmesi amacıyla kullanılmıştır.

8-Pip: Python programlama dili için kullandığımız standart paket yöneticisi. Kütüphanelerin indirilmesi ve bağımlılıkların kurulmasını sağlar.

9.Looperman.com: Bu web site üzerinden kök notası etiketlenmiş olan verileri indirip modeli eğitmek amacıyla kullandık.

III. PROJEDEKİ ENGELLER VE ÇÖZÜMLER

**Veri indirme kotası limiti:** Günde yaklaşık olarak sadece 100-120 arası veri çekmemize ve kaydetmemize neden oluyordu.

**Çözüm:** Tempmail.com gibi sitelerden tek kullanımlık hazır email hesabı aldık ve o şekilde elimizde biriken hesaplarla fazlasıyla veri toplayabildik.

**Dosya boyutunun büyük olması:** Verilerimizin boyutu fazla olduğu için verileri bir araya getirmekte zorlandık.

**Çözüm:** Ayrı bilgisayarlarda indirdiğimiz verileri harici bir HDD kullanarak verilerimizi orada depoladık.

**Kod ile İlgili Hatalar:** Librosa kütüphanesi ile ilgili sürüm hataları yaşadık.Ayrıca kodları taramak gayet uzun sürdü.

**Çözüm:** Kütüphaneyi Pip komutuyla güncelledik ve kodları taraması için bekledik.

IV. GELECEK GELİŞTİRMELER VE TAVSİYE EDİLENLER

## A. Talimat Setleri

Daha karmaşık işlem türlerini destekleyen talimat setleri eklenebilir. İşlemcinin daha uygulama alanına hitap etmesi sağlanabilir. Kesme yöntemi için geliştirmeler dinamik kesme önceliklendirme sistemleri yüklenebilir

## B. Daha Fazla Sınıf

. Daha fazla veri sınıfı eklenebilir (Do,Re,Mi). Modelin daha uygun çalışması veri bulunabilir. Ve ayrıca minör ve majör notalar eklenerek daha kesin ve etkili sonuçlar elde edilebilir.

.

## C. Model Optimizasyonu

Modelin daha uygun ve kararlı çalışabilmesi için hiperparametre gibi makine öğrenmesi modelini hızlandıracak talimatlar uygulanabilir.

## D. Bellek Yönetimi

Geliştirme için daha akıllı bellek yönetim algoritmaları kullanarak daha etkin ve hızlı sonuçlar elde edilmesi sağlanabilir. Bellek performansı artabilir ve kaynak kullanımı daha verimli hale getirilir.

1. Ölçeklendirilebirlik ve Yük Dengeleme

Veri hacmi ve kullanıcı sayısı arttıkça, uygulamanın daha büyük talepleri karşılayabilmesi için ölçeklendirilmesi gerekebilir. Yük dengeleme uygulayarak ve talepleri birden fazla sunucuya dağıtarak tıkanıklıkların önüne geçilebilir ve sistemin yüksek trafiği kaldırabilmesi sağlanabilir. AWS veya Google Cloud gibi bulut tabanlı bir

altyapı, projenin büyümesiyle birlikte gerekli ölçeklenebilirlik ve kaynakları sağlayarak destek verebilir..

1. Daha Uygun Makine Öğrenmesi Modeli Optimizasyonu

Projenin gelecekteki sürümleri için daha ileri düzey bir geliştirme, müzik piyasası trendlerini tahmin etmek veya finansal verilerdeki anormallikleri tespit etmek için makine öğrenmesi algoritmalarının entegrasyonunu içerebilir. Tarihsel veriler üzerinde bir model eğitilerek, sistem kullanıcılara içgörüler veya tahminler sunabilir ve projeye akıllı bir katman ekleyebilir. Bu amaçla scikit-learn veya TensorFlow gibi kütüphaneler kullanılabilir..

.

# REFERENCES

<https://librosa.org/doc>

<https://scikit-learn.org/stable/documentation.html>

<https://www.looperman.com/>

<https://pandas.pydata.org/docs/>

<https://numpy.org/doc/>