

ŞARKILARDAKİ ENSTRÜMANLARIN TESPİTİ

IDENTIFICATION OF INSTRUMENTS IN SONGS

BUSE YENER, NİSA AKSOY

BİLİŞİM SİSTEMLERİ MÜHENDİSLİĞİ

KOCAELİ ÜNİVERSİTESİ

buse.yener@gmail.com

BİLİŞİM SİSTEMLERİ MÜHENDİSLİĞİ

KOCAELİ ÜNİVERSİTESİ

nisaaksoy08@gmail.com

Özet

Bu dokümanda, şarkılardaki müzik aletlerini ayırt etmek için geliştirilen transformer tabanlı modellerin detayları yer almaktadır. . Proje, veri hazırlama, model eğitimi, performans değerlendirmesi ve sonuçların görselleştirilmesi aşamalarını içermektedir.

Abstract

This document includes details of transformer-based models developed to distinguish musical instruments in songs. The project includes data preparation, model training, performance evaluation and visualization of results.

1.Giriş

Projemizi yaparken Makine Öğrenmesi dalında en iyi verim alabildiğimiz Python programlama dilini ve Python dilinde yazılan kodları çalıştırabilmek için Jupyter Notebook geliştirme ortamını kullandık. Öncelikle Python yazılım dili nedir, ne işe yarar?

Python: Web uygulamaları, yazılım geliştirme, veri bilimi ve makine öğreniminde (ML) yaygın olarak kullanılan bir programlama dilidir. Geliştiriciler, etkili ve öğrenmesi kolay olduğu ve birçok farklı platformda çalıştırılabildiği için bu dili kullanır. Ücretsiz olarak indirilebilir, her türlü sistemle iyi bir entegrasyon sağlar ve geliştirme hızını artırır.

Avantajları:

- Geliştiriciler, basit ve İngilizceye benzer bir söz dizimine sahip olduğundan Python dilinde yazılmış programı kolayca okuyabilir ve anlayabilir.
- Geliştiriciler, diğer birçok dile kıyasla çok daha az kod satırıyla bir Python programı yazabildiğinden, geliştiricilerin daha üretken olmasını sağlar.

- Python, neredeyse her görev için yeniden kullanılabilir kodları içeren geniş bir standart kütüphane sahiptir. Geliştiricilerin kodu sıfırdan yazmasına gerek kalmaz.
- Geliştiriciler Python'ı Java, C ve C++ gibi diğer popüler programlama dilleriyle kolayca kullanabilir.
- Bir sorunla karşılaşırsanız Python topluluğundaki ekiplerden hızlıca destek alabilirsiniz.
- Python'ı öğrenmek için internette birçok yararlı kaynak bulunmaktadır. Kolayca videolar, dersler, belgeler ve geliştirici kılavuzları bulabilirsiniz.
- Windows, macOS, Linux ve Unix gibi farklı bilgisayar işletim sistemleri arasında taşınabilir.

Veri Bilimi ve Makine Öğrenimi: Veri bilimi, verilerdeki değerli bilgileri ayıklamaktır ve makine öğrenimi (ML), bilgisayarlara verilerden otomatik olarak öğrenmesini ve doğru tahminlerde bulunmasını öğretir. Veri bilimciler, aşağıdakiler gibi görevler için Python'ı kullanır:

- Veri temizleme, yanlış verileri düzeltme ve kaldırma
- Verileri ayıklama, verilerin çeşitli özelliklerini seçme
- Veri etiketleme, veriler için anlamlı adlar ekleme
- Verilerden farklı öngörüler çıkarma
- Çizgi grafikler, çubuk grafikler, sütun grafikler ve dilim grafikler gibi grafikler yardımıyla verileri görselleştirme

Veri bilimciler, Makine Öğrenmesi modellerini eğitmek ve verileri doğru şekilde sınıflandıran sınıflandırıcılar derlemek için Python dilinin Makine Öğrenmesi kütüphanelerini kullanır. Görüntü, metin ve ağ trafiği

sınıflandırması, konuşma tanıma ve yüz tanıma gibi sınıflandırma görevlerini yapması için çalışan kişiler Python tabanlı sınıflandırıcılar kullanır.

Jupyter Notebook: Veri bilimi, makine öğrenimi, veri analizi ve bilimsel hesaplamalar gibi alanlarda yaygın olarak kullanılan, etkileşimli bir not defteri uygulamasıdır. Bu araç, programcılar ve veri bilimcilerin verileri keşfetmelerine, analiz etmelerine, sonuçlarını görselleştirmelerine olanak tanır. Hem kod hem de metin içeriğini bir arada sunabilmesi, Jupyter Notebook'u diğer geliştirme araçlarından ayıran en önemli özelliklerden biridir.

2. Şarkılardaki Enstrümanların Tespiti

Bu proje, piyano, keman, gitar ve kanun gibi çeşitli müzik enstrümanlarına ait ses verilerinin toplanması, işlenmesi ve sınıflandırmaya uygun hale getirilmesinden oluşmaktadır. Projede, YouTube'dan indirilen ses dosyalarının sessizlik ve gürültüden arındırılması, ardından MFCC özelliklerinin çıkarılması ile veriler etiketlenerek makine öğrenmesi modelleri için uygun bir veri seti oluşturulmaktadır.

2.1 Veri Toplama

Proje için her enstrümana özel olarak YouTube çalma listeleri kullanılarak veri toplanmıştır. Ses dosyalarının düzenli bir şekilde saklanması ve işlenmesi için her enstrüman için ayrı bir klasör oluşturulmuştur. Ses verilerinin toplandığı bu işlemde:

Araçlar: Verileri toplamak için yt_dlp kütüphanesi kullanıldı.

İşlem: YouTube playlist'leri kullanılarak, her bir çalma listesindeki tüm videolar indirildi ve ses dosyası olarak kaydedildi. Kaydedilen dosyalar MP3 formatındadır ve her bir enstrüman için ilgili klasörde saklanmıştır.

2.2 Veri Ön İşleme

Toplanan ses verilerinin analiz edilebilir hale getirilmesi için ön işleme aşaması gerçekleştirilmiştir. Bu aşama, ses dosyalarındaki gürültü ve sessizlikleri azaltmak, verileri segmentlere ayırmak ve her segment için MFCC çıkarımı yapmak gibi işlemleri içerir.

Sessizlik Temizleme: Ses dosyalarındaki gereksiz sessiz bölümler, analizde istenmeyen gürültüye ve modelin doğruluğunu olumsuz etkileyebilecek unsurlara yol açabilir. Bu nedenle, pydub kütüphanesi kullanılarak her ses dosyasındaki sessizlik bölgeleri temizlenmiştir. Sessizliği giderilen ses dosyaları, işlenebilir duruma gelmiştir.

Gürültü Temizleme: Toplanan verilerde arka plan gürültüsü olabileceğinden, sesin netliği ve ayırt edici özellikleri korumak amacıyla gürültü temizleme işlemi yapılmıştır. noisereducer kütüphanesi kullanılarak her ses kaydının arka plan gürültüsü azaltılmıştır.

Normalizasyon : Gürültü temizleme işleminin ardından, ses sinyali, genlik dalgalanmalarını gidermek amacıyla

normalize edilmiştir. Bu adım, ses dosyasının sabit bir ses seviyesi sunmasını sağlamakta ve özellik çıkarımı adımıyla istikrarlı sonuçlar elde edilmesine katkıda bulunmaktadır.

2.3 Özellik Çıkarımı

Ön işleme adımlarından geçen ses verileri, sınıflandırma modeline uygun öznitelikler çıkarmak için hazırlanmıştır. Bu projede, her bir ses segmenti için MFCC (Mel Frekans Kepstral Katsayıları) çıkarımı yapılmıştır. MFCC, ses verisinin frekans özelliklerini temsil eden ve özellikle ses tanıma işlemlerinde yaygın olarak kullanılan bir tekniktir. Bu proje kapsamında, her ses parçası 3 saniyelik segmentlere ayrılmış ve her segment için 13 MFCC değeri hesaplanarak kaydedilmiştir.

2.4 Veri Segmentasyonu ve Etiketleme

Veri Segmentasyonu: Her bir ses kaydı, analiz sürecine uygun olacak şekilde 3'er saniyelik parçalara bölünmüştür. Bu işlemde, toplam süre boyunca her bir segment ayrı bir dosya olarak kaydedilmiştir. Her segment, gürültü temizleme ve MFCC çıkarımı işlemlerine tabi tutularak dosya sistemi içinde saklanmıştır.

Etiketleme: Veriler, ilerideki model eğitim aşamalarında kullanılmak üzere etiketlenmiştir. Her enstrümana ait segmentler, piyano, keman, gitar, kanun gibi etiketler ile ilişkilendirilmiştir. Bu sayede, model eğitimi sırasında her veriye hangi enstrümana ait olduğu bilgisiyle ulaşılabilecektir.

Veri toplama ve ön işleme süreçlerinin sonucunda, müzik enstrümanlarının ayırt edici özelliklerini yansıtan yüksek kaliteli bir veri seti oluşturulmuştur. Bu aşamada, ses verilerinden gürültü ve sessizlik temizlenmiş, ardından Mel-Frekans Kepstrum Katsayıları (MFCC) çıkarılarak veriler etiketlenmiştir. Bu şekilde hazırlanmış veri seti, makine öğrenmesi modellerinin eğitimi için uygun hale getirilmiş olup, modelin doğruluğunu artırmaya ve enstrümanları doğru bir şekilde sınıflandırmaya yönelik kritik bir temel oluşturmıştır.

```
Toplam veri sayısı: 14647
Örnek bir MFCC verisi: [[-528.546    -511.15637   -474.06735    ... -282.1321    -301.36893
  -323.31006 ]
 [  5.6669626   24.289175   50.58533    ...  110.64725   107.038704
  101.80883 ]
 [ -1.170081    5.1891623   11.636992    ... -50.12626   -49.600876
  -46.741325 ]
 ...
 [ -3.3576822   -3.465158   -1.8829875    ... -11.877163   -5.7145214
   5.3668895 ]
 [ -6.817239    -12.677327   -9.990316    ... -4.5545955  -10.044651
  -8.943605 ]
 [ -5.220399    -10.639755  -12.067743    ... -13.589108   -8.248816
  -5.7354994 ]]
```

Örnek bir etiket: gitar

Yukarıdaki resimde, proje kapsamında elde edilen toplam veri sayısının 14,647 olduğu görülmektedir. Her bir veri örneği, ses sinyalinin frekans özelliklerini temsil eden MFCC değerlerini içermektedir. MFCC özellikleri, her enstrümanın kendine özgü ses karakteristiklerini

yansıtarak makine öğrenmesi algoritmaları için öznelilik setini oluşturmaktadır. Görselde yer alan örnek MFCC verisi, bir gitar sesine ait olup, ses sinyalinin farklı zaman dilimlerindeki frekans bileşenlerini göstermektedir. Bu tür MFCC verileri, enstrüman sınıflandırma modelinin eğitimi için temel veri setini oluşturmaktadır.

3. Yapay Zeka Modelleri

Şarkılarda çalınan enstrümanları otomatik olarak tespit edebilmek amacıyla beş farklı transformatör modeli eğitilmiştir. Model eğitimi için MFCC özellikleri kullanılmış ve piyano, keman, gitar ile kanun enstrümanlarına ait ses verileri işlenmiştir. Her modelin başarımı, doğruluk, hassasiyet, özgüllük, F1 skoru gibi metrikler ile değerlendirilmiş, ayrıca eğitim süreçlerine ait epoch, kayıp grafikleri ve çıkarım süreleri analiz edilmiştir.

3.1 Audio Spectrogram Transformer (AST) Modeli

Audio Spectrogram Transformer (AST) modeli ile eğitilerek dört farklı enstrümanın (gitar, keman, kanun, piyano) sınıflandırılması gerçekleştirilmiştir.

Eğitim için, her enstrüman grubuna ait MFCC öznelilikleri içeren veri seti kullanılmış, veri dengesizliğini gidermek amacıyla SMOTE (Synthetic Minority Over-sampling Technique) uygulanmıştır. Veri seti 100 zaman adımı ve 13 MFCC katsayısı şeklinde sabitlenerek modelin eğitimi için uygun hale getirilmiştir.

Model, Transformer mimarisi kullanarak Multi-Head Attention ve Feed Forward Network katmanları üzerine inşa edilmiştir. Modelin çıktı katmanında softmax aktivasyonu ile dört sınıflı (gitar, keman, kanun, piyano) bir çıktı elde edilmiştir.

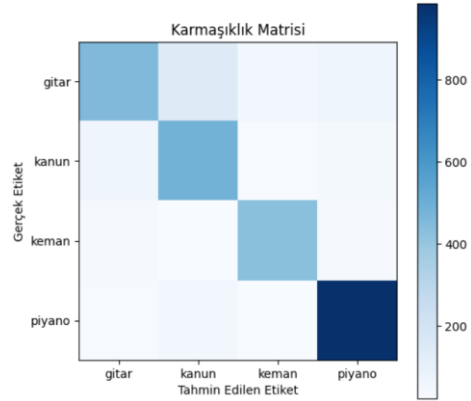
Performans Metrikleri:

- Accuracy: 0.8034
- Precision: 0.8046
- Recall: 0.8034
- F1-Score: 0.8014
- AUC: 0.8626
- Average Sensitivity (Duyarlılık): 0.7913
- Average Specificity (Özgüllük): 0.9338
- Çıkarım süresi: 0.8657 saniye
- Eğitim süresi: 1009.9507 saniye

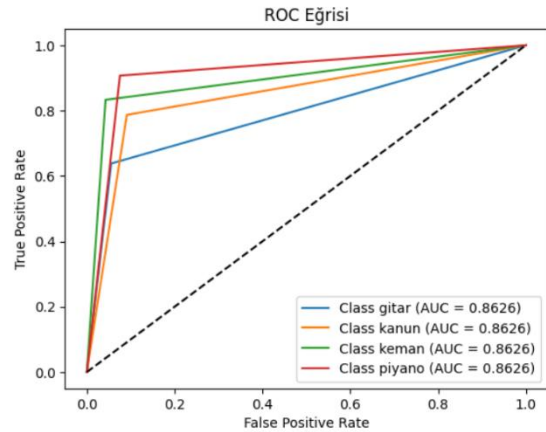
Model, enstrüman sınıflandırma görevinde yüksek doğruluk oranı sağlamıştır. AUC değeri 0.8626 olup, modelin genel sınıflandırma başarımının yüksek olduğunu göstermektedir. Ortalama özgüllük (Specificity) 0.9338 ile modelin yanlış pozitif oranının düşük olduğunu göstermektedir. Ortalama duyarlılık (Sensitivity) ise 0.7913 olup, modelin belirli enstrümanları doğru tespit etme oranını ifade etmektedir.

Karmaşıklık Matrisi: Aşağıdaki karmaşıklık matrisi, modelin tahmin ettiği sınıflar ile gerçek etiketler arasındaki ilişkiyi göstermektedir. Matrisin her satırı

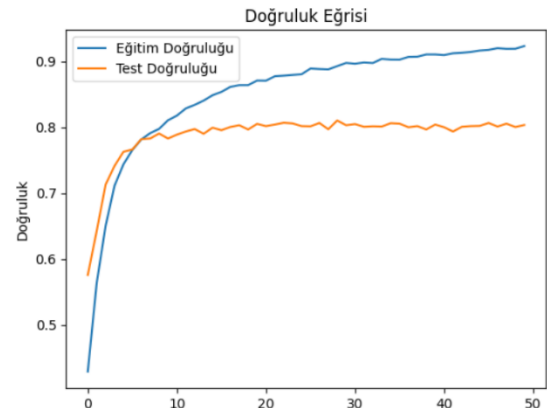
gerçek etiketleri, her sütunu ise modelin tahmin ettiği etiketleri temsil etmektedir.



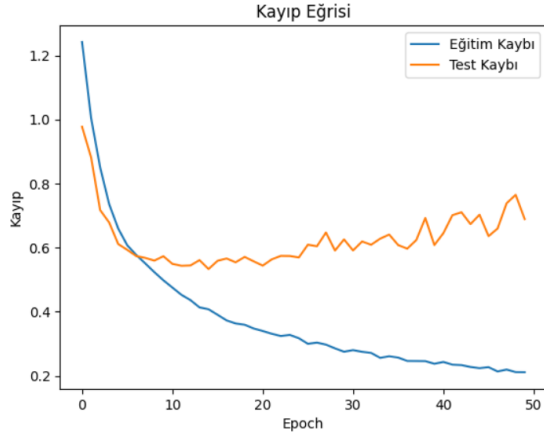
Roc Eğrisi: Aşağıdaki grafik, modelin her sınıf için ROC (Receiver Operating Characteristic) eğrisini ve AUC (Area Under the Curve) değerlerini göstermektedir. ROC eğrisi, modelin yanlış pozitif oranına (False Positive Rate) karşılık doğru pozitif oranını (True Positive Rate) göstererek sınıflandırma performansını değerlendirir. 0.8626'lık AUC değeri, modelin iyi bir sınıflandırma performansına sahip olduğunu göstermektedir.



Doğruluk Eğrisi: Grafik, modelin eğitim ve test veri kümeleri üzerindeki doğruluk oranlarının epoch (iterasyon) sayısına göre değişimini göstermektedir.



Kayıp Eğrisi: Grafik, modelin eğitim ve test veri kümeleri üzerindeki kayıplarının epoch (iterasyon) sayısına göre değişimini göstermektedir.



3.2 Data2Vec Modeli

Data2Vec modeli ile gitar, keman, kanun ve piyano olmak üzere dört farklı enstrümanın sınıflandırılması gerçekleştirilmiştir.

Veri hazırlığı için her enstrümana ait MFCC öznitelikleri kullanılmış ve veri setindeki dengesizliği gidermek amacıyla SMOTE (Synthetic Minority Over-sampling Technique) uygulanmıştır. Veriler, her biri 100 zaman adımı ve 13 MFCC katsayısı içerecek şekilde sabitlenerek modelin eğitimi için uygun hale getirilmiştir.

Modelin mimarisi Transformer tabanlı olup, Multi-Head Attention ve Feed Forward Network katmanları kullanılarak oluşturulmuştur. Çıktı katmanında ise softmax aktivasyonu ile dört sınıflı (gitar, keman, kanun, piyano) sınıflandırma yapılmıştır.

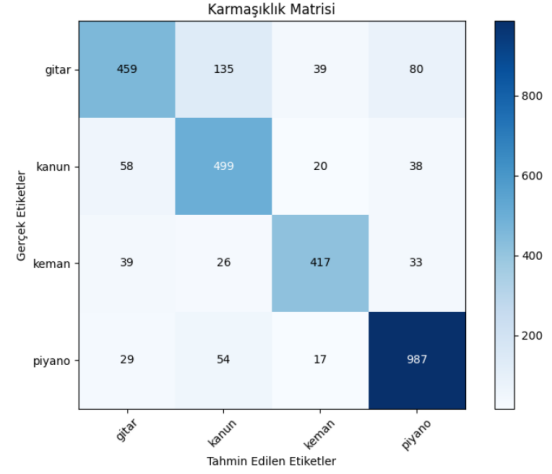
Performans Metrikleri:

- Accuracy: 0.8061
- Precision: 0.8081
- Recall: 0.8061
- F1-Score: 0.8043
- AUC: 0.8637
- Average Sensitivity (Duyarlılık): 0.7932
- Average Specificity (Özgüllük): 0.9342
- Çıkarım süresi: 0.9134 saniye
- Eğitim süresi: 1015.9630 saniye

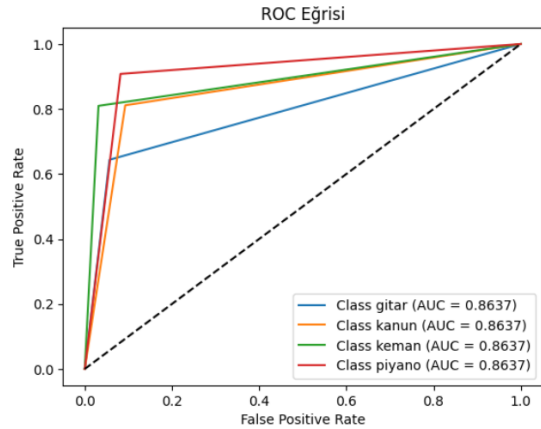
Bu modelin performansı, genellikle iyi sonuçlar vermektedir. Doğruluk (accuracy) %80.61 ile yüksek bir orana sahipken, Precision ve Recall değerleri de sırasıyla 0.8081 ve 0.8061, yani modelin pozitif sınıfları doğru şekilde tanıma ve yakalama konusunda başarılı olduğunu gösteriyor. F1-Score ise 0.8043 ile Precision ve Recall arasında iyi bir denge sağladığını ortaya koyuyor. AUC değeri 0.8637 ile modelin sınıfları ayırma yeteneğinin güçlü olduğunu gösterirken, duyarlılık (sensitivity) 0.7932 ve özgüllük (specificity) 0.9342 ile modelin pozitif ve negatif sınıfları doğru şekilde tanıma oranı oldukça yüksek. Çıkarım süresi 0.9134 saniye ile hızlı, eğitim

süresi ise 1015.96 saniye ile modelin eğitilmesi için geçen sürenin yönetilebilir olduğunu belirtmektedir. Genel olarak, model hem doğruluk hem de işlem hızları açısından tatmin edici bir performans sergilemektedir.

Karmaşıklık Matrisi: Aşağıda, modelin tahmin ettiği sınıflar ile gerçek etiketler arasındaki ilişkiyi gösteren karmaşıklık matrisi yer almaktadır. Satırlar gerçek etiketleri, sütunlar ise modelin tahmin ettiği etiketleri temsil etmektedir.

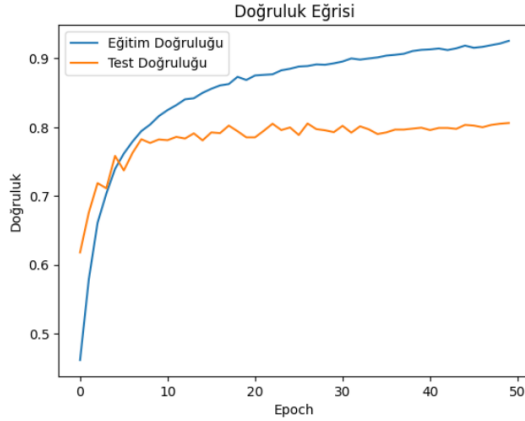


Roc Eğrisi: Grafik, modelin dört farklı sınıf (gitar, kanun, keman, piyano) için ROC (Receiver Operating Characteristic) eğrilerini göstermektedir. ROC eğrisi, bir sınıflandırma modelinin performansını değerlendirmek için kullanılır ve Doğru Pozitif Oranı (True Positive Rate) ile Yanlış Pozitif Oranı (False Positive Rate) arasındaki ilişkiyi gösterir.

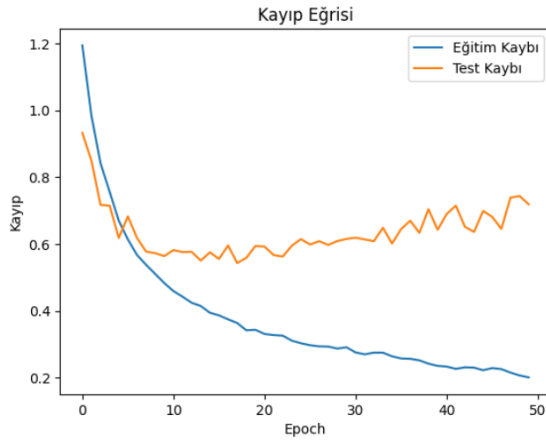


Grafikte her bir sınıf için AUC (Area Under the Curve) değeri 0.8637 olarak verilmiştir. AUC değeri, modelin sınıflandırma performansını özetleyen bir metriktir ve değerin 1'e ne kadar yakınsa, modelin performansı o kadar iyidir.

Doğruluk Eğrisi: Grafik, modelin eğitim ve test doğruluklarının epoch (iterasyon) sayısına göre değişimini göstermektedir.



Kayıp Eğrisi: Grafik, modelin eğitim ve test kayıplarının epoch (iterasyon) sayısına göre değişimini göstermektedir. Eğitim kaybı (mavi çizgi) epoch sayısı arttıkça sürekli azalırken, test kaybı (turuncu çizgi) başlangıçta azalıp ardından dalgalanmalar göstermektedir.



3.3 Wave2Vec Modeli

Wave2Vec tabanlı bir model kullanılarak dört farklı müzik enstrümanının sınıflandırılması gerçekleştirilmiştir. Her bir MFCC dosyası, modelin giriş katmanına uygun şekilde 100 zaman adımı ve 13 MFCC katsayısı ile yeniden şekillendirilmiştir. Veri setindeki dengesizliği gidermek amacıyla, SMOTE (Synthetic Minority Over-sampling Technique) tekniği uygulanmıştır. Bu teknik, daha az temsil edilen sınıflara yapay veriler ekleyerek modelin her sınıfı daha dengeli bir şekilde öğrenmesini sağlamaktadır.

Model, Wave2Vec mimarisine dayalı olarak tasarlanmıştır. Wave2Vec, temel olarak bir ses verisini işlerken kullanılan bir transformer tabanlı yapıdır. Bu yapının en önemli avantajı, dikkat (attention) mekanizmaları kullanarak ses verisinin zaman içindeki ilişkilerini daha iyi öğrenebilmesidir.

Wave2Vec blokları, her bir enstrüman grubunu temsil eden MFCC özniteliklerini işler ve sınıflandırmaya hazır hale getirir. Modeldeki ana yapı, her bir blokta bir konvolüsyon katmanı (Conv1D), normalizasyon

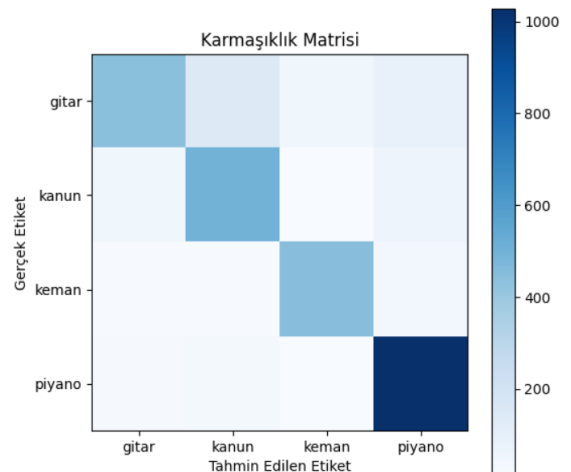
(BatchNormalization), aktivasyon fonksiyonu (ReLU) ve dropout katmanları içermektedir. Konvolüsyon katmanları, ses verisindeki önemli özellikleri çıkarmak için kullanılırken, BatchNormalization ve Dropout katmanları modelin genelleme yeteneğini artırarak aşırı öğrenmeyi (overfitting) engellemeyi hedeflemiştir. Son katmanlarda ise sınıflandırma için softmax aktivasyonu kullanılarak dört sınıf (gitar, keman, kanun, piyano) arasında seçim yapılmıştır.

Performans Metrikleri:

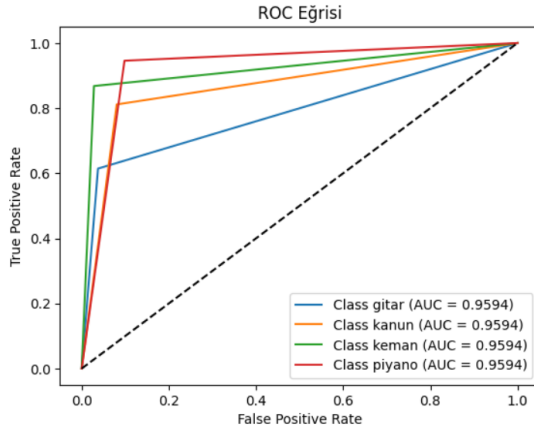
- Accuracy: 0.8232
- Precision: 0.8255
- Recall: 0.8232
- F1-Score: 0.8187
- AUC: 0.9594
- Average Sensitivity (Duyarlılık): 0.5112
- Average Specificity (Özgüllük): 0.5716
- Çıkarım süresi: 0.9453 saniye
- Eğitim süresi: 776.1240 saniye

Wave2Vec modelinin performansı, enstrüman sınıflandırma görevinde oldukça başarılı sonuçlar ortaya koymuştur. Model, %82.32 doğruluk oranı ve 0.9594 AUC değeriyle yüksek sınıflandırma başarısı sergilemiştir. Precision ve recall değerleri sırasıyla 0.8255 ve 0.8232 olarak hesaplanmış, modelin doğru sınıflandırma ve enstrümanları doğru tespit etme konusunda dengeli bir performans gösterdiği görülmüştür. Ortalama duyarlılık ve özgüllük değerlerinin daha düşük seviyelerde olması, bazı sınıflar için daha yüksek doğruluk sağlansa da, bazı sınıflarda iyileştirme yapılabileceğini işaret etmektedir. Model, 0.9453 saniyelik çıkarım süresiyle hızlı bir performans sergilerken, 776.1240 saniye eğitim süresiyle de verimli bir şekilde eğitilmiştir. Genel olarak, modelin performansı yüksek doğruluk, düşük yanlış pozitif oranı ve hızlı işlem süreleri ile olumlu değerlendirilmiştir.

Karmaşıklık Matrisi: Karmaşıklık matrisi, modelin tahmin ettiği sınıflar ile gerçek etiketler arasındaki ilişkiyi gösteren bir tablodur. Her satır, gerçek etiketleri; her sütun ise modelin tahmin ettiği etiketleri temsil etmektedir. Aşağıda, karmaşıklık matrisinin görseli yer almaktadır.

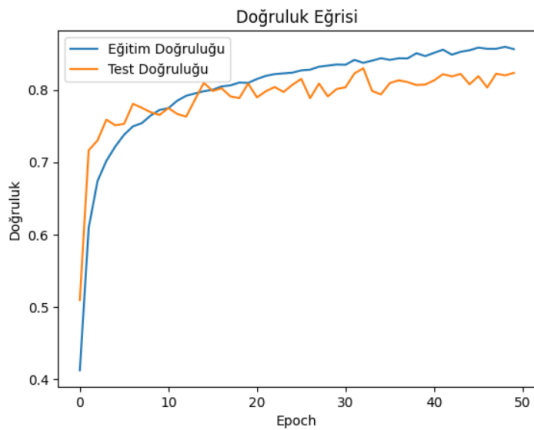


Roc Eğrisi: Grafik, modelin dört farklı sınıf (gitar, kanun, keman, piyano) için ROC (Receiver Operating Characteristic) eğrilerini göstermektedir. ROC eğrisi, bir sınıflandırma modelinin performansını değerlendirmek için kullanılır ve Doğru Pozitif Oranı (True Positive Rate) ile Yanlış Pozitif Oranı (False Positive Rate) arasındaki ilişkiyi gösterir.

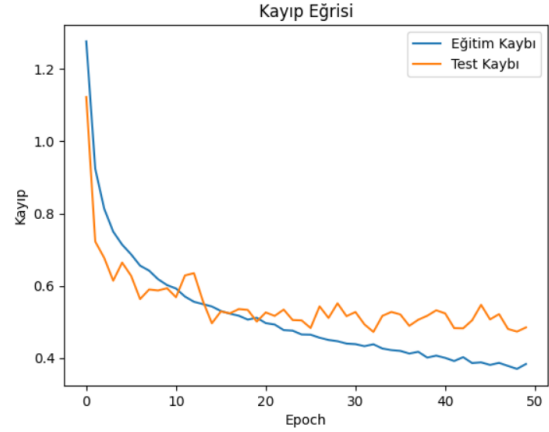


Grafikte her bir sınıf için AUC (Area Under the Curve) değeri 0.9594 olarak verilmiştir. AUC değeri, modelin sınıflandırma performansını özetleyen bir metriktir ve değerin 1'e ne kadar yakınsa, modelin performansı o kadar iyidir.

Doğruluk Eğrisi: Grafik, modelin eğitim ve test doğruluğunu epoch (dönem) sayısına göre göstermektedir. Yatay eksen epoch sayısını, dikey eksen ise doğruluk oranını temsil etmektedir.



Kayıp Eğrisi: Grafik, modelin eğitim ve test kayıplarının epoch (iterasyon) sayısına göre değişimini göstermektedir. Yatay eksen epoch sayısı, dikey eksen ise kayıp değerleri yer almaktadır.



3.4 Hubert Modeli

Hubert modeli, ses işleme ve doğal dil işleme (NLP) gibi alanlarda kullanılan bir derin öğrenme modelidir. Temelde, çok katmanlı bir konvolüsyonel ağ yapısı ile verilerin özelliklerini öğrenir ve bu özellikleri kullanarak çıktı üretir. Bu modelin temel bileşenleri, konvolüsyonel katmanlar, katman normalizasyonu, aktivasyon fonksiyonları ve eklemeli bağlantılar gibi bileşenler içerir.

Modelin özelleştirilmiş hali, ses verilerini (MFCC öznitelikleri gibi) işlemek için uyarlanmıştır. Kullanılan Hubert modelinde, ses sinyali verisinin analiz edilmesi ve sınıflandırılması için konvolüsyonel bloklar kullanılmıştır. Modelde katman normalizasyonu ve dropout gibi teknikler, modelin genelleme yeteneğini artırmak ve aşırı öğrenmeyi engellemek amacıyla eklenmiştir.

Her bir Hubert bloğunda, giriş verisi ilk olarak bir Conv1D katmanı aracılığıyla işlenir. Ardından, LayerNormalization ile normalizasyon yapılır ve aktivasyon fonksiyonu olarak ReLU uygulanır. Son olarak, shortcut bağlantısı eklenerek, daha derin katmanlardan öğrenilen bilgilerin daha önceki katmanlara aktarılması sağlanır. Bu bağlantı, daha hızlı öğrenme ve daha doğru sonuçlar elde edilmesini sağlar.

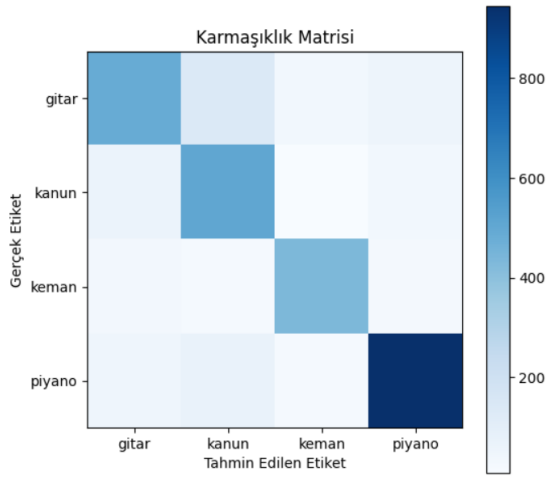
Performans Metrikleri:

- Accuracy: 0.8099
- Precision: 0.8153
- Recall: 0.8099
- F1-Score: 0.8104
- AUC: 0.9501
- Average Sensitivity (Duyarlılık): 0.5278
- Average Specificity (Özgüllük): 0.5490
- Çıkarım süresi: 1.2838 saniye
- Eğitim süresi: 924.0358 saniye

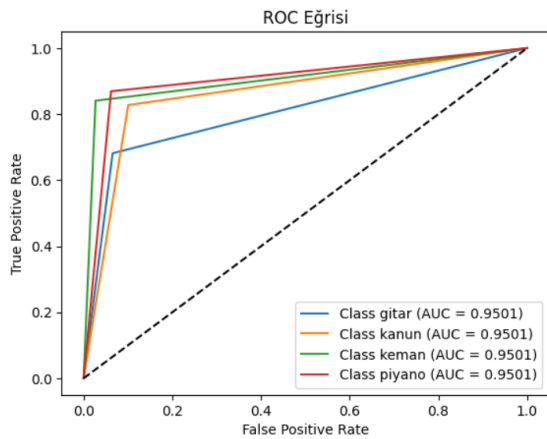
Modelin performansı, iyi bir genel başarıyı yansıtmaktadır. Accuracy (doğruluk) değeri 0.8099, modelin test verileri üzerinde doğru sınıflandırma oranını yüksek tutmakta olduğunu gösteriyor. Precision (kesinlik) ve Recall (duyarlılık) değerleri sırasıyla 0.8153 ve 0.8099, modelin hem yanlış pozitiflerden kaçınma hem de gerçek pozitifleri doğru şekilde tanıma yeteneğini iyi dengelediğini gösteriyor. F1-Score ise 0.8104 ile modelin

kesinlik ve duyarlılığı arasında iyi bir denge sağladığını vurguluyor. AUC (Area Under Curve) skoru 0.9501 ile modelin sınıflandırma doğruluğunun çok yüksek olduğunu gösteriyor. Ortalama duyarlılık 0.5278 ve özgüllük 0.5490 değerleri, modelin bazı sınıflar için zayıf performans sergileyebileceğini ancak genel olarak makul bir duyarlılık ve özgüllük dengelediğini gösteriyor. Eğitim süresi 924 saniye ile oldukça uzun olsa da, çıkarım süresi sadece 1.28 saniye, modelin gerçek zamanlı kullanım için uygun olduğunu ve hızlı tahmin yapabildiğini ortaya koyuyor.

Karmaşıklık Matrisi: Karmaşıklık matrisi, modelin tahmin ettiği sınıflar ile gerçek etiketler arasındaki ilişkiyi gösteren bir tablodur. Her satır, gerçek etiketleri; her sütun ise modelin tahmin ettiği etiketleri temsil etmektedir. Aşağıda, karmaşıklık matrisinin görseli yer almaktadır.

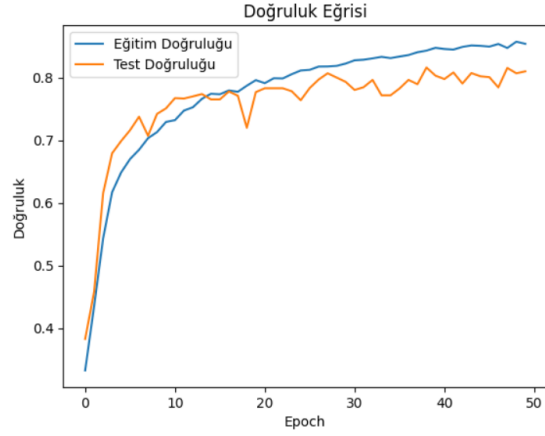


Roc Eğrisi: Grafik, modelin dört farklı sınıf (gitar, kanun, keman, piyano) için ROC (Receiver Operating Characteristic) eğrilerini göstermektedir. ROC eğrisi, bir sınıflandırma modelinin performansını değerlendirmek için kullanılır ve Doğru Pozitif Oranı (True Positive Rate) ile Yanlış Pozitif Oranı (False Positive Rate) arasındaki ilişkiyi gösterir.

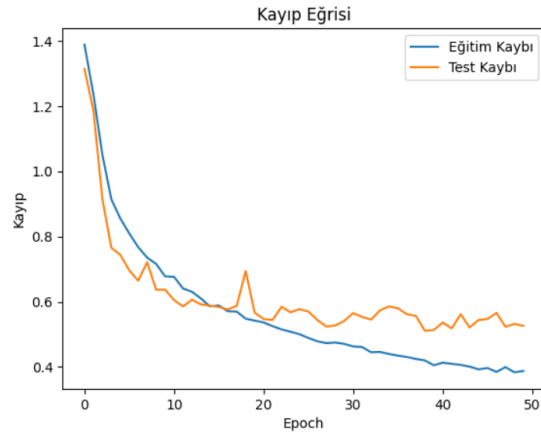


Grafikte her bir sınıf için AUC (Area Under the Curve) değeri 0.9501 olarak verilmiştir. AUC değeri, modelin sınıflandırma performansını özetleyen bir metriktir.

Doğruluk Eğrisi: Grafik, modelin eğitim ve test veri kümeleri üzerindeki doğruluk oranlarının epoch (iterasyon) sayısına göre değişimini göstermektedir.



Kayıp Eğrisi: Grafik, modelin eğitim ve test kayıplarının epoch (iterasyon) sayısına göre değişimini göstermektedir. Yatay ekseninde epoch sayısı, dikey ekseninde ise kayıp değerleri yer almaktadır.



3.5 Conformer Modeli

Model, Conformer (Convolutional Transformer) mimarisini temel alır. Conformer, dil modellemesinde yaygın olarak kullanılan Transformer yapısının, ses verisiyle daha etkili çalışabilmesi için bazı iyileştirmeler içeren bir modeldir. Conformer, özellikle ses tanıma ve sıralı verilerle çalışmada yüksek başarı elde etmiştir.

Modelin eğitiminde kullanılan veri seti, her bir enstrüman için ayrı ayrı MFCC öznelikleri ile temsil edilen ses dosyalarından oluşturulmuştur. MFCC, sesin frekans spektrumunun daha anlamlı bir şekilde temsil edilmesini sağlayarak, ses tanıma gibi görevlerde etkili bir özellik çıkarımı tekniğidir. Eğitim verisi, her enstrümana karşılık gelen etiketler ile birlikte modelin girişine verilmiştir.

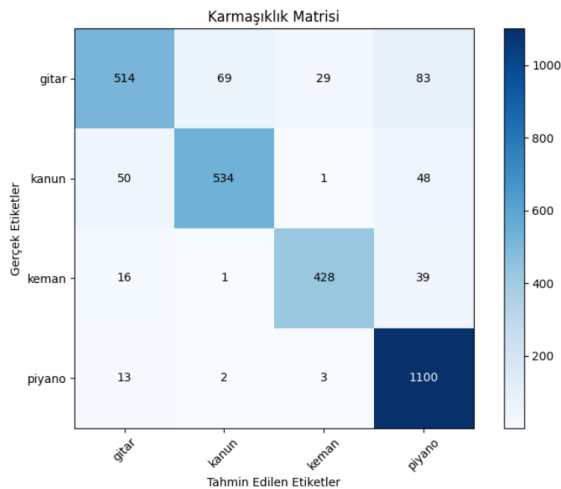
Conformer yapısının temel özelliklerini barındıran bir derin öğrenme ağından oluşmaktadır. Modelin ilk katmanı, giriş verisini uygun boyutlara şekillendirir. Ardından, bir dizi evrişimsel katman, ses verisindeki önemli özellikleri çıkarmak için kullanılır. Bu evrişimsel katmanlardan sonra, MaxPooling katmanları ile veri boyutu küçültülür. GlobalAveragePooling katmanı ile modelin daha genel özellikler öğrenmesi sağlanır ve en son katmanlarda, sınıflandırma işlemi için Dense katmanları ve Softmax fonksiyonu ile sonuçlar elde edilir. Modelin eğitim süreci, 50 epoch boyunca, 32 örnekten oluşan batch'ler ile gerçekleştirilmiştir.

Performans Metrikleri:

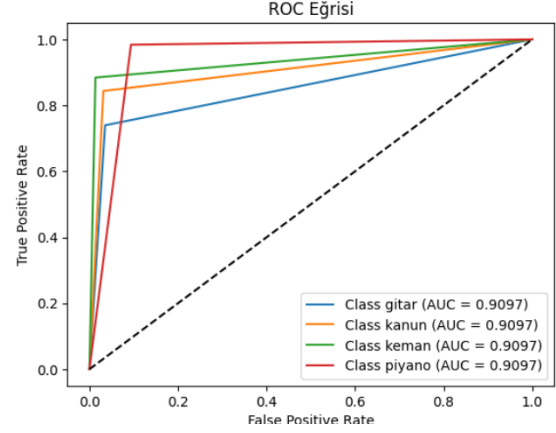
- Accuracy: 0.8792
- Precision: 0.8798
- Recall: 0.8792
- F1-Score: 0.8767
- AUC: 0.9097
- Average Sensitivity (Duyarlılık): 0.8628
- Average Specificity (Özgüllük): 0.9565
- Çıkarma süresi: 0.3522 saniye
- Eğitim süresi: 225.0066 saniye

Modelin performansını değerlendirdiğimizde, accuracy değeri 0.8792 ile genel doğruluğun yüksek olduğunu, precision ve recall değerlerinin sırasıyla 0.8798 ve 0.8792 olduğunu görürüz; bu da modelin doğru tahmin yapma ve pozitif sınıfları yakalama konusunda dengeli bir başarı sergilediğini gösterir. F1-Score 0.8767, modelin dengeyi sağladığını, AUC ise 0.9097 ile modelin doğru sınıfları ayırt etme başarısının yüksek olduğunu ortaya koyar. Average Sensitivity (0.8628) ve Average Specificity (0.9565) ise modelin pozitif ve negatif sınıfları doğru ayırt etme başarısını gösterir. Eğitim süresi 225 saniye iken, çıkarma süresi yalnızca 0.3522 saniye, modelin hızlı tahmin yapabildiğini gösterir.

Karmaşıklık Matrisi: Karmaşıklık matrisi, modelin tahmin ettiği sınıflar ile gerçek etiketler arasındaki ilişkiyi gösteren bir tablodur. Her satır, gerçek etiketleri; her sütun ise modelin tahmin ettiği etiketleri temsil etmektedir. Aşağıda, karmaşıklık matrisinin görseli yer almaktadır.

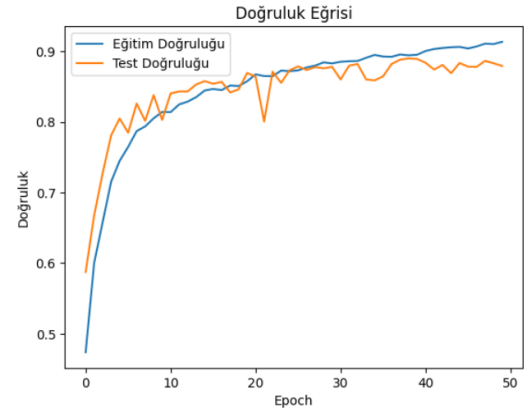


Roc Eğrisi: Grafik, modelin dört farklı sınıf (gitar, kanun, keman, piyano) için ROC (Receiver Operating Characteristic) eğrilerini göstermektedir. ROC eğrisi, bir sınıflandırma modelinin performansını değerlendirmek için kullanılır ve Doğru Pozitif Oranı (True Positive Rate) ile Yanlış Pozitif Oranı (False Positive Rate) arasındaki ilişkiyi gösterir.

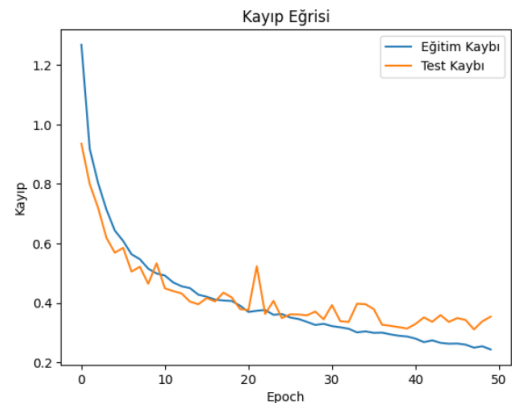


Grafikte her bir sınıf için AUC (Area Under the Curve) değeri 0.9097 olarak verilmiştir. AUC değeri, modelin sınıflandırma performansını özetleyen bir metriktir.

Doğruluk Eğrisi: Grafik, modelin eğitim ve test veri kümeleri üzerindeki doğruluk oranlarının epoch (iterasyon) sayısına göre değişimini göstermektedir.



Kayıp Eğrisi: Grafik, modelin eğitim ve test kayıplarının epoch (iterasyon) sayısına göre değişimini göstermektedir. Yatay eksen epoch sayısı, dikey eksen ise kayıp değerleri yer almaktadır.



4. Proje Linkleri

4.1 Github

211307048 Buse Yener –

<https://github.com/buseyener/Enstruman-Tanimlayici>

211307089 Nisa Aksoy –

<https://github.com/nisaaksoy/Enstruman-Tanimlayici>

4.2 Drive

211307048 Buse Yener –

<https://drive.google.com/file/d/1ph0Vx8MA63VMAK07Abw2as5MTBuyRRDv/view?usp=sharing>

211307089 Nisa Aksoy –

https://drive.google.com/drive/folders/1WhNSINogmJaaNLOU6CzXdg0eWzq_hef7

4.3 Google Colab

211307048 Buse Yener –

https://drive.google.com/drive/folders/1wIYjC_7hH1-W8DxMyQrlqTr0CJ_tDp_S

211307089 Nisa Aksoy –

https://drive.google.com/drive/folders/1WhNSINogmJaaNLOU6CzXdg0eWzq_hef7

Kaynakça

1. <https://aws.amazon.com/tr/what-is/python/#:~:text=Python%3B%20web%20uygulamalar%C4%B1%2C%20yaz%C4%B1%C4%B1m%20geli%C5%9Firme,%C3%A7al%C4%B1%C5%9F%C4%B1r%C4%B1labildi%C4%9Fi%20i%C3%A7in%20Python'%C4%B1%20kullan%C4%B1r>
2. <https://bilginc.com/tr/blog/python-nedir-python-hakkinda-hersey-158/>
3. <https://www.komtas.com/post/python-programlama-dili-ve-kullanim-alanlari>
4. <https://www.python.org/>
5. <https://jupyter.org/>
6. <https://medium.com/datarunner/jupyter-notebook-nedir-502c024c0076>
7. <https://pybit.es/articles/ast-intro/>
8. <https://www.geeksforgeeks.org/hubert-model/>
9. <https://neurosys.com/blog/wav2vec-2-0-framework>
10. <https://www.unite.ai/tr/data2vec/>