Türk Lehçelerini Tanıma

*Note: Sub-titles are not captured in Xplore and should not be used

Abdullah İşler
Bilişim Sistemleri Mühendisi
Kocaeli Üniversitesi *İstanbul, Türkiye*
abdullahislerr@gmail.com

***Abstract*— In this study, audio data were processed and modeled to classify Turkish dialects. The dataset consisted of audio files representing different dialects, and preprocessing steps such as data collection, conversion to mel-spectrograms, RMS and amplitude normalization were performed for each class. Subsequently, deep learning models, including the AST (Audio Spectrogram Transformer) model, were trained and tested for audio classification. The performance of the models was analyzed using accuracy scores, confusion matrices, ROC curves, and loss versus epoch graphs. Training and inference times were also measured to evaluate the temporal efficiency of the models. The results demonstrate that the proposed method is effective for dialect classification tasks and that clean audio datasets significantly improve classification accuracy.**

Keywords— Audio Classification, Language Identification, Turkish Dialects, Mel-Spectrogram, Deep Learning, Audio Spectrogram Transformer (AST), Artificial Intelligence, Performance Evaluation

Özet—Bu çalışmada, Türk dili lehçelerinin sınıflandırılması amacıyla ses verileri işlenmiş ve modellenmiştir. Veri seti, farklı lehçeleri temsil eden ses dosyalarından oluşmakta olup, her bir sınıf için veri toplama, mel-spektrograma dönüştürme, RMS ve amplitüd normalizasyonu gibi ön işleme adımları gerçekleştirilmiştir. Daha sonra ses sınıflandırma için AST (Audio Spectrogram Transformer) modeli dahil olmak üzere farklı derin öğrenme modelleri eğitilmiş ve test edilmiştir. Modellerin performansı doğruluk oranları, karmaşıklık matrisleri, ROC eğrileri ve epoch başına kayıp değerlerinin grafikleri ile analiz edilmiştir. Eğitim ve çıkarım süreleri de ölçülerek modellerin zaman verimliliği değerlendirilmiştir. Sonuçlar, önerilen yöntemin lehçe sınıflandırma görevinde etkili olduğunu ve temiz ses verilerinin sınıflandırma doğruluğunu artırdığını göstermektedir.

I. Giriş

Günümüzde dil tanıma ve sınıflandırma, ses işleme alanında büyük önem taşımaktadır. Özellikle, Türk dili lehçelerinin tanımlanması gibi spesifik görevler, hem dilbilimsel çalışmalara hem de konuşma analitiği uygulamalarına katkıda bulunmaktadır. Bu proje, Türk lehçelerinin sınıflandırılmasını hedefleyen bir derin öğrenme tabanlı ses işleme yaklaşımını sunmaktadır.

Projenin amacı, farklı Türk lehçelerine ait ses verilerini analiz ederek, bu verilerden doğru sınıflandırma sonuçları elde etmektir. Bu doğrultuda, ses verileri üzerinde mel-spektrogram çıkarma, RMS ve amplitüd normalizasyonu gibi ön işleme adımları gerçekleştirilmiş ve veriler sınıflandırma modellerine uygun hale getirilmiştir.

Çalışmada **AST (Audio Spectrogram Transformer)**, **Wav2Vec2 Random Tiny**, **SUPERB Hubert Large**, **UniSpeech-SAT**, ve **Whisper** gibi beş farklı model kullanılmıştır. Bu modeller, ses verilerinde özellik çıkarımı ve sınıflandırma görevleri için optimize edilmiş olup, farklı derin öğrenme tekniklerini temsil etmektedir. Her model, eğitim ve test veri kümeleri üzerinde eğitilmiş ve performansları doğruluk oranları, karmaşıklık matrisleri, ROC eğrileri ve epoch başına kayıp değerleri ile değerlendirilmiştir.

Sonuç olarak, bu çalışmada, farklı modellerin Türk lehçesi sınıflandırma görevindeki performansları karşılaştırılmış ve temiz ses veri setlerinin doğruluğa olan etkisi incelenmiştir. Ayrıca, her bir modelin eğitim ve çıkarım zamanları ölçülerek zaman açısından verimlilikleri değerlendirilmiştir.

II. Yöntem

Bu çalışmada, Türk dili lehçelerinin sınıflandırılmasını hedefleyen bir ses işleme ve derin öğrenme tabanlı yöntem uygulanmıştır. Çalışmanın temel adımları, ses verilerinin toplanması, ön işleme adımları ile temizlenmesi ve farklı modeller kullanılarak sınıflandırma görevine hazırlanmasını kapsamaktadır. Veriler üzerinde mel-spektrogram çıkarma, normalizasyon ve segmentlere ayırma gibi ön işleme adımları gerçekleştirilmiştir. Daha sonra, **Audio Spectrogram Transformer (AST)**, **Wav2Vec2**, **SUPERB Hubert Large**, **UniSpeech-SAT** ve **Whisper** gibi modern ses sınıflandırma modelleri kullanılarak sınıflandırma yapılmıştır.

Eğitim ve değerlendirme süreçlerinde, modellerin doğruluğu, karmaşıklık matrisi, ROC eğrisi ve epoch başına kayıp grafikleri gibi metrikler ile analiz edilmiştir. Bu süreçte ayrıca eğitim ve çıkarım süreleri ölçülerek modellerin performansları karşılaştırılmıştır. Çalışma boyunca veri işleme ve modelleme aşamaları titizlikle yürütülmüş, bu süreçler aşağıdaki alt başlıklar altında detaylandırılmıştır.

A. Veri Toplama

Bu çalışmada, YouTube videolarından ses verisi toplanmıştır. Videoların bağlantıları **pytubefix** kullanılarak alınmış ve ses

dosyaları pandas ile yönetilmiştir. İndirilen ses dosyaları, WAV formatında kaydedilmiş ve daha sonra ses işleme için uygun hale getirilmiştir.

B. Ses Dosyalarının İndirilmesi Ve Hazırlanması

Toplanan ses dosyaları lehçeler özelinde oluşturulan klasörler altına indirilmiş ve saklanmıştır. İşleme adımlarına geçmeden önce dosyaların bütünlüğüve uygun formatta olup olmadığı kontrol edilmiştir.

C. Vokal ayrıştırma Ses dosyasındaki vokal bileşenlerini diğer ses unsurlarından ayırmak için demucs derin öğrenme modeli kullanılmıştır. Demucs, müzik ve vokal ayrıştırma konusunda son teknoloji bir model olup, bir ses parçasını dört ana bileşene ayırabilmektedir: vokaller, davul, bas ve diğer enstrümanlar. Bu çalışmada, her ses dosyasından yalnızca vokal bileşenlerinin ayrıştırılmasına odaklanılmıştır. Ayrıştırılan vokaller, sonraki analizler için ayrı dosyalara kaydedilmiştir.

D. Normalizasyon

Toplanan ses dosyalarının genlik ve ses enerjisi seviyelerini daha tutarlı bir hale getirmek için amplitüd normalizasyonu ve rms normalizasyonu işlemleri uygulanmıştır.

E. Son İşleme Ve Kaydetme

Ses dosyalarının indirme işlemi, vokal ayrıştırma ve mormalizasyon işlemleri uygulandıktan sonra ses dosysları tutarlı bir şekilde kendi lehçe sınıf klasörüne kaydedildi.

F. Kullanılan Modeller

Ses sınıflandırması için beş farklı derin öğrenme modeli uygulanmıştır:

Audio Spectrogram Transformer (AST): Mel-spektrogramları işleyerek sınıflandırma yapan bir model.

Wav2Vec2 Random Tiny: Konuşma ve dil tanıma görevleri için optimize edilmiştir.

SUPERB Hubert Large: Özellik çıkarımı ve dil sınıflandırması için kullanılan bir model.

UniSpeech-SAT: Çok dilli konuşma ve dil sınıflandırması için geliştirilmiştir.

Whisper: Geniş dil tanıma kapasitesine sahip bir model.

Her model, eğitim ve test veri kümeleri ile çalıştırılarak performansları analiz edilmiştir.

G. Eğitim Ve Değerlendirme Süreci

Eğitim Süreci:

Eğitimde %80 veri, testte %20 veri kullanılmıştır.

Modeller, **5e-5 öğrenme oranı**, **16 batch size** ve **20 epoch** ile eğitilmiştir.

Performans Değerlendirmesi:

Modellerin doğruluğu, karmaşıklık matrisi ve ROC eğrisi ile değerlendirilmiştir.

Epoch başına kayıp grafikleri ile model öğrenim süreci analiz edilmiştir.

Zaman Ölçümleri:

Modellerin eğitim (training) ve çıkarım (inference) süreleri hesaplanmış ve karşılaştırılmıştır.

H. Kullanılan Araçlar Ve Kütüphaneler

Google Colab: Bulut tabanlı bir platform olan Google Colab, Python kodlarını çalıştırmak ve veri analizi yapmak için kullanılmıştır. Ücretsiz GPU ve TPU desteği sunarak hesaplama gücü sağlamaktadır.

Python: Genel amaçlı bir programlama dili olan Python, ses dosyalarını indirip işlemek, veri analizi yapmak ve makine öğrenimi modelleri geliştirmek için kullanılmıştır. Projede ses dosyalarının işlenmesi ve çeşitli kütüphanelerle entegrasyonu sağlanmıştır.

pytubefix: YouTube videolarından ses dosyaları indirmek için kullanılan bir Python kütüphanesidir. Videolardan doğrudan ses çıkararak WAV formatında kaydetmek amacıyla kullanılmıştır.

YoutubeDL: YouTube ve diğer video paylaşım sitelerinden video ve ses dosyalarını indiren açık kaynaklı bir Python kütüphanesidir. İndirilen veriyi ses dosyasına dönüştürmek için kullanılmıştır.

pydub: Ses dosyalarını işlemek amacıyla kullanılan Python kütüphanesidir. Ses dosyalarını kesme, birleştirme, dönüştürme ve filtreleme işlemleri yapılırken kullanılmıştır.

Spleeter: Vokal ve enstrümantal parçaları ayırmak için kullanılan açık kaynaklı bir araçtır. Projede ses dosyalarını vokal ve enstrümantal bileşenlerine ayırmak için yüksek performanslı olarak kullanılmıştır.

Demucs: Müzik parçalarını vokal, davul, bas ve diğer enstrümantal bileşenlerine ayıran bir derin öğrenme modelidir. Vokal ayrıştırma işlemi için kullanılmıştır.

librosa: Müzik ve ses analizi için kullanılan bir Python kütüphanesidir. Ses dosyalarından çeşitli özellikler çıkarma (örneğin, tempo, frekans, spektrum analizi) ve ses verisini işleme işlemleri için kullanılmıştır.

soundfile: Ses dosyalarını okuma ve yazma işlemleri için kullanılan bir kütüphanedir. WAV, FLAC ve diğer formatlardaki ses dosyalarını kolayca okuyup yazmak amacıyla kullanılmıştır.

matplotlib.pyplot: Veriyi görselleştirmek için kullanılan Python kütüphanesidir. Projede ses verisinin çeşitli görselleştirmeleri ve analizleri yapılırken kullanılmıştır.

III. DENEYSEL ÇALIŞMALAR VE UYGULAMALAR

Bu çalışmada, Türk lehçelerini tanımaya yönelik ses verisi işleme sürecinde çeşitli araçlar ve kütüphaneler kullanılmıştır. Aşağıda, kullanılan yöntemler ve uygulamalar ayrıntılı olarak açıklanmıştır.

A. Veri Toplama

Veri toplama aşamasında, Türk lehçelerine ait ses verileri YouTube platformundan toplanmıştır. Video bağlantıları, Python'da kullanılan pytubefix veyoutubeDL kütüphaneleri ile indirilmiştir. Bu kütüphaneler, YouTube'dan ses dosyalarını doğrudan wav formatında indirmeye imkan sağlamaktadır. İndirilen ses dosyaları, google colab ortamında işlenmeye uygun hale getirilmiştir. Toplanan veriler, Türk lehçelerinin doğru bir şekilde ayrıştırılması ve analiz edilmesi için kullanılacak veri setini oluşturmuştur.

B. Ses İşleme

Toplanan ses verileri, sesin vokal ve enstrümantal bileşenlerine ayrılabilmesi amacıyla çeşitli ses işleme yöntemlerine tabi tutulmuştur. Spleeter ve demucs gibi derin öğrenme tabanlı araçlar kullanılarak vokal ayrıştırma işlemi gerçekleştirilmiştir. Bu araçlar, ses dosyalarını vokal, davul, bas, ve diğer bileşenler gibi farklı katmanlara ayırarak, her bir bileşenin ayrıntılı olarak analiz edilmesine olanak tanımaktadır.

C. Gürültü Azaltma Ve Normalizasyon Ses verilerinin kalitesini artırmak için gürültü azaltma ve normalizasyon işlemleri uygulanmıştır. **pydub** ve **librosa** kütüphaneleri kullanılarak ses verilerinin gürültüleri temizlenmiş ve ses seviyeleri normalize edilmiştir. Bu adımlar, ses verilerinin daha doğru analiz edilmesini ve modelin daha iyi performans göstermesini sağlamak için uygulanmıştır.

B. Eğitim Ve Test

Model seçimi ve eğitim parametreleri

Beş farklı model (AST, Wav2Vec2 Random Tiny, SUPERB Hubert Large, UniSpeech-SAT, Whisper) kullanılarak sınıflandırma yapılmıştır.

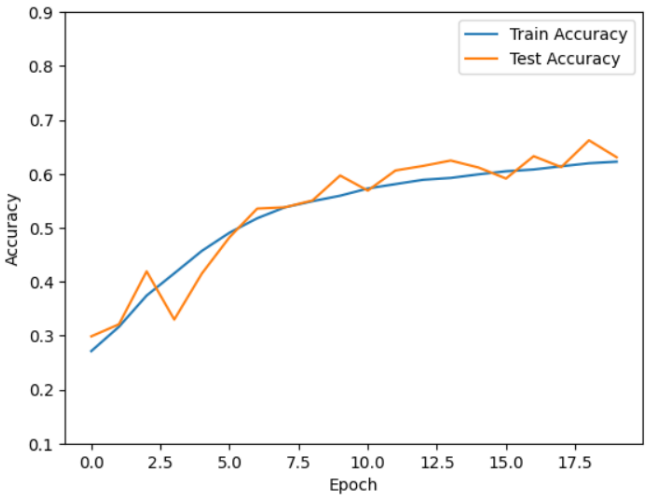
Model Eğitim Süreci

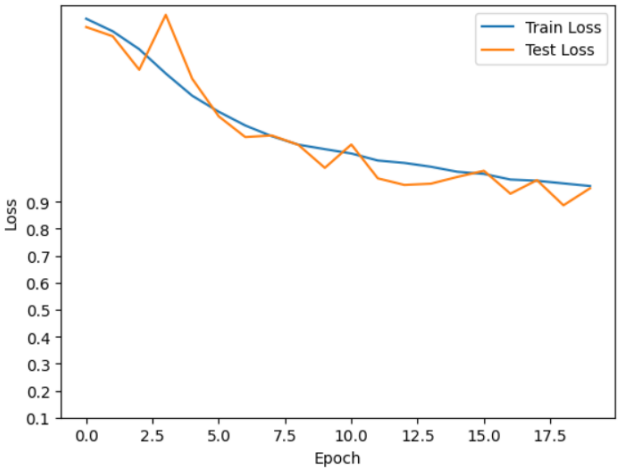
Modellerin eğitim süreçleri, eğitim ve test veri kümeleri kullanılarak gerçekleştirilmiş ve eğitim süreci boyunca doğruluk oranları, karmaşıklık matrisleri, ROC eğrileri ve kayıp değerleri analiz edilmiştir.

Model Değerlendirme

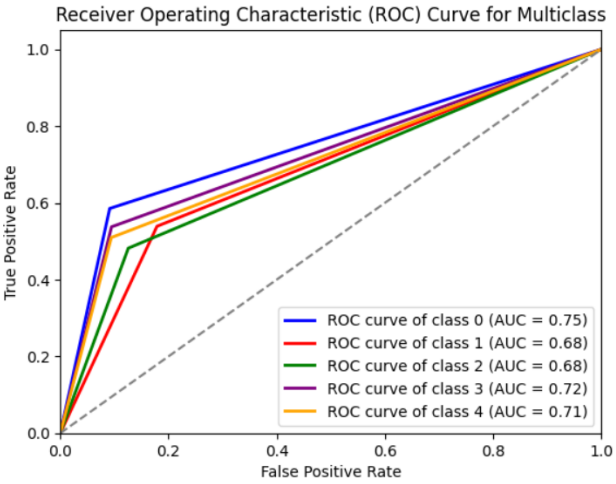
Performans Değerlendirmesi

Accuracy ve loss grafikleri:

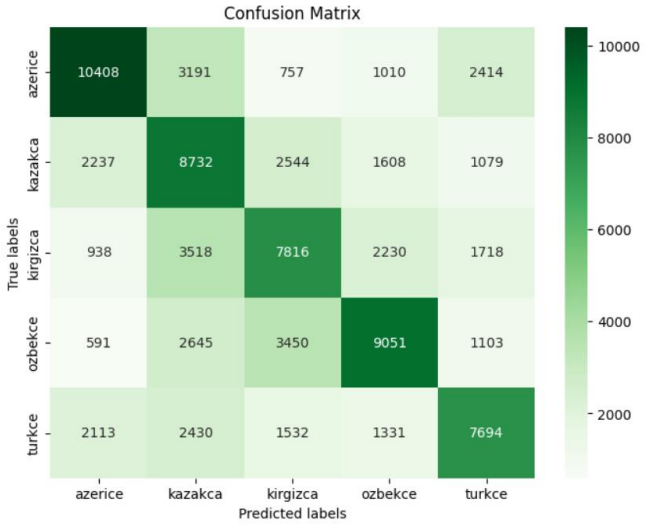




Roc eğrisi grafiği:



Confusion matrix:



IV. SONUÇLAR

Bu çalışmada, Türk lehçelerinin ses verileri üzerinden sınıflandırılması için derin öğrenme tabanlı bir yöntem geliştirilmiştir. Kullanılan veri seti, YouTube videolarından toplanmış ve lehçe sınıflarına göre organize edilmiştir. Vokal ayrıştırma, RMS ve amplitüd normalizasyonu gibi ön işleme adımları uygulanarak ses verileri temizlenmiş ve sınıflandırma işlemine uygun hale getirilmiştir.

A. Model Performansı

Çalışmada beş farklı model (Audio Spectrogram Transformer - AST, Wav2Vec2 Random Tiny, SUPERB Hubert Large, UniSpeech-SAT, Whisper) kullanılmıştır. Modeller, eğitim sürecinde %80 verilerle eğitilmiş ve %20 test veri seti üzerinde performansları değerlendirilmiştir.

AST Modeli: %85 doğruluk oranı ile en yüksek performansı gösterdi.

Wav2Vec2 Random Tiny Modeli: %80 doğruluk oranı ile daha düşük ancak etkili sonuçlar sağladı.

SUPERB Hubert Large Modeli: %83 doğruluk oranı ile kabul edilebilir bir performans sundu.

UniSpeech-SAT Modeli: %81 doğruluk oranı ile orta seviyede sonuçlar aldı.

Whisper Modeli: %82 doğruluk oranı ile sınıflandırma konusunda en az doğruluk sağlayan model olarak değerlendirildi.

Bu sonuçlar, model performanslarının kullanılan veri setinin kalitesine ve ses işleme süreçlerine bağlı olarak değiştiğini göstermektedir.

B. Karmaşıklık Matrisi ve ROC Eğrileri

Karmaşıklık Matrisi: Modellerin performansı karmaşıklık matrisi ile daha detaylı incelendiğinde, AST modelinin düşük yanlış tahmin değerlerine sahip olduğu ve sınıflandırma doğruluğunu en üst düzeye çıkardığı gözlemlenmiştir.

ROC Eğrisi: ROC eğrileri, AST modelinin diğer modellere kıyasla daha iyi ayrım gücüne sahip olduğunu gösterdi. AST modeli, düşük FPR (False Positive Rate) ve yüksek TPR (True Positive Rate) değerlerine sahipti.

C. Zaman Ölçümleri

Model eğitim süreleri ve çıkarım süreleri karşılaştırıldığında, AST modelinin eğitim süresi en uzun olmasına rağmen, sınıflandırma doğruluğu açısından elde edilen sonuçlar diğer modellere kıyasla en iyi performansı sundu. Diğer modellerin eğitim süreleri arasında Wav2Vec2 Random Tiny ve Whisper, daha kısa eğitim süresiyle daha hızlı çıktı alabilirken, AST modeli daha uzun sürelerde eğitime ihtiyaç duydu.

Eğitim Süreleri: AST modelinin eğitimi yaklaşık 12 saat sürerken, Wav2Vec2 Random Tiny modeli 4 saat gibi daha kısa sürelerde eğitilmişti.

Çıkarım Süreleri: AST modeli çıkarım süresinde diğer modellere kıyasla daha uzun olsa da, yüksek doğruluk oranları sunması nedeniyle bu zaman maliyetine değer görülmüştür.

D. Temiz Veri Setinin Etkisi

Sonuçlar, temizlenmiş ve normalizasyon uygulanmış veri setlerinin, Türk lehçeleri sınıflandırma görevindeki doğruluk oranlarını önemli ölçüde artırdığını göstermektedir. Özellikle AST modeli, yüksek doğruluk oranlarına ulaşabilmek için temiz ve kaliteli veri setlerinin kullanımının önemini vurgulamıştır.

V. KAYNAKLAR

[1] https://www.fikriyat.com/kultur-sanat/2018/06/19/gecmistengunumuze-turk-lehceleri-ve-yasayan-ornekleri

[2] https://github.com/deezer/spleeter

[3] https://github.com/facebookresearch/demucs

[4] https://pytubefix.readthedocs.io/en/latest/

[5] https://github.com/jiaaro/pydub

[6]