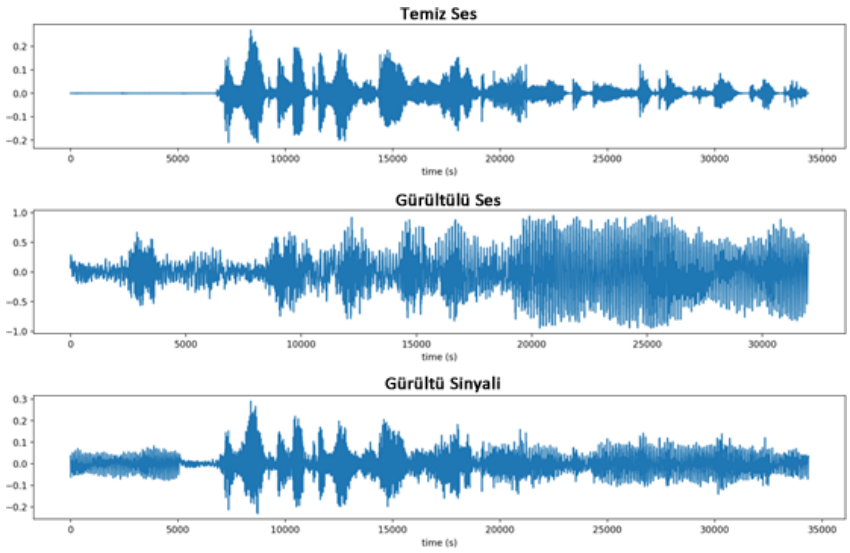
Projenin Adı: Sekte, Ses Sinyalindeki Gürültünün Derin Öğrenme Yöntemleriyle Azaltılması

DÖNEM İÇİ YAPILAN ÇALIŞMALARIN ÖZETİ

Gerçekleştirilen projede Evrişimsel Sinir Ağları (CNN) yöntemi ile eğitilmiş model kullanılarak gönderilen gürültülü bir ses kaydının gürültüsü azaltılmakta ve gürültünün içeriği ile ilgili bilgiler kullanan kişiye sunulmaktadır.

İlk olarak projenin yapılabilirliği konusunda araştırmalar ve benzer projeler incelenmiştir. Genellikle donanımsal ve sesin içeriğinden bağımsız olarak desibel değerlerinin değiştirilmesi ile gürültünün azaltıldığı gözlemlenmiştir. Makine öğrenmesi yöntemleri ile ses sinyalindeki gürültünün ayırt edilebilmesi için öncelikle gürültülü ve gürültüsüz seslerin özellikleri ayrıştırılmalı ve bunu ayırt edebilecek bir model ortaya çıkarılmalıdır.

Yapılan kaynak ve yöntem araştırmaları sonucunda gürültülü ve temiz iki farklı büyük veri setinin ayrı ayrı özelliklerinin ayrıştırılması gerektiği görülmüştür. Gürültülü sesler için 8.732 adet etiketlenmiş ve 10 gruba ayrılmış kısa ses kayıtlarından oluşan açık kaynak lisanslı bir veri seti indirilmiştir. Temiz sesler için ise 51.037 adet İngilizce gürültüsüz konuşma kaydından oluşan yine açık kaynak kodlu başka bir veri seti indirilmiştir.



Şekil 1. Temiz ses, gürültü ve iki sinyalin birleşimi

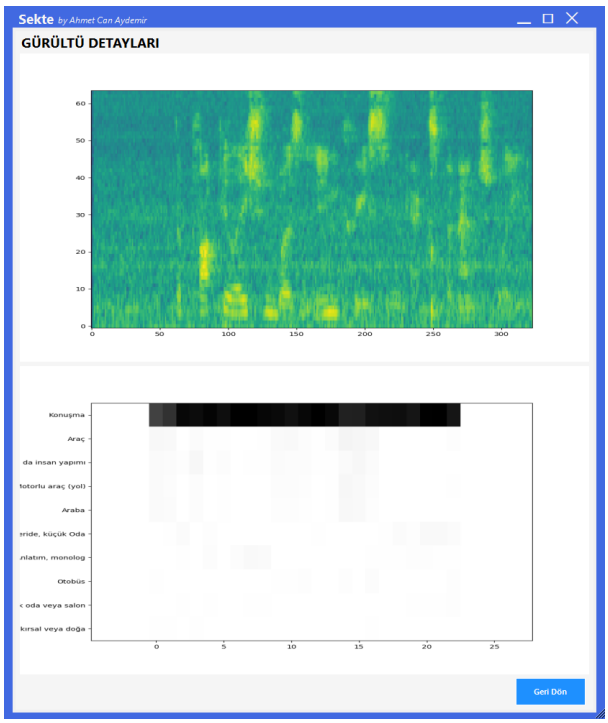
İndirilen veri setlerinde sessiz bölgeler silinmiş, aynı bit hızı oranına getirilmiş ve gereksiz alan harcaması önlenmiştir. Gürültülü ve gürültüsüz seslerin arasındaki farklılıklar ve özellikler short time fourier transform gibi matematiksel yöntemlerle ayrıştırılmış ve yaklaşık 90gb boyutunda model eğitiminde kullanılacak dosya oluşturulmuştur. Bu dosyalar train, validation ve test olmak üzere eğitimin daha doğru sonuç vermesi için ayrıştırılmış. Bölümlere ayrılan sesler farklı kategorilerden seçilmiş ve birleştirilmiştir.

Bu kadar büyük boyutlu verinin standart bir bilgisayarda işlenmesi çok fazla işlem gücü gerektirmektedir. Bu sebeple bu işlemin daha hızlı ve az maliyetli yapılabilme yolları araştırılmıştır. İnternet üzerinden ücretsiz olarak yüksek performanslı ekran kartları ile yapay zeka model eğitimleri yapılabilecek Google servisleri bulunmuş ve kullanılmıştır. Servis 90gb dosyanın kişisel bilgisayar üzerinden kullanılmasına izin vermediği için Google Drive üzerinden 200gb alan satın alınmış ve üniversite interneti kullanılarak tüm eğitim dosyaları bu bulut alanına yüklenmiştir.

Detayları materyal ve metot kısmında bahsedilecek olan model oluşturma işlemi projenin en çok zaman alan işlemidir. Daha önce özellikleri ayrıştırılıp dosyalara dönüştürülen gürültülü ve temiz seslerin birleşimleri, yüksek performanslı ekran kartı ile defalarca eğitilmiştir. Eğitimdeki kayıp-hata oranı %15 civarında olana kadar bu işlem tekrarlanmıştır.

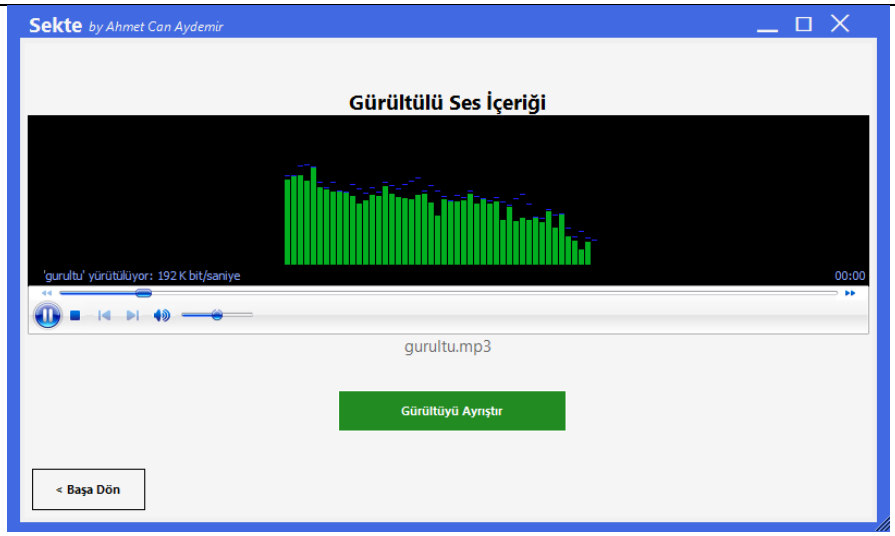
İşlem eğitim tamamlandıktan sonra eğitime dahil edilmemiş gürültülü ve temiz rastgele sesler birleştirilip modelin sonucu gözlemlenmiştir. Gözlem sonucunda modelin gürültüyü büyük oranda temizlediği görülmüştür. Fakat köpek havlaması gibi bazı spesifik seslerde başarı oranı değişkenlik göstermektedir.

Model dosyası tüm testlerin yapılmasının ardından kayıt edilmiştir. Bu model verilen gürültülü ses sinyalinden gürültüsü olabildiğince azaltılmış bir ses sinyali üretmek için üretilmiştir. Bunun dışında projede gürültülü sesin hangi konumunda ve hangi kategoride olduğunu da göstermesi beklenmektedir. Bunun için YouTube videolarındaki seslerin insanlar tarafından etiketlenmesi ile oluşturulmuş farklı bir veri seti kullanılmıştır. Bu veri setini kullanan ve tam olarak istediğimiz işi yapan bir model halihazırda bulunmakta ve açık kaynak lisansı ile kullanımına izin verilmektedir. Bu modelin kullandığı kategori tablosundaki 521 adet İngilizce kategori ismi Türkçe'ye çevrilmiş ve kayıt edilmiştir.



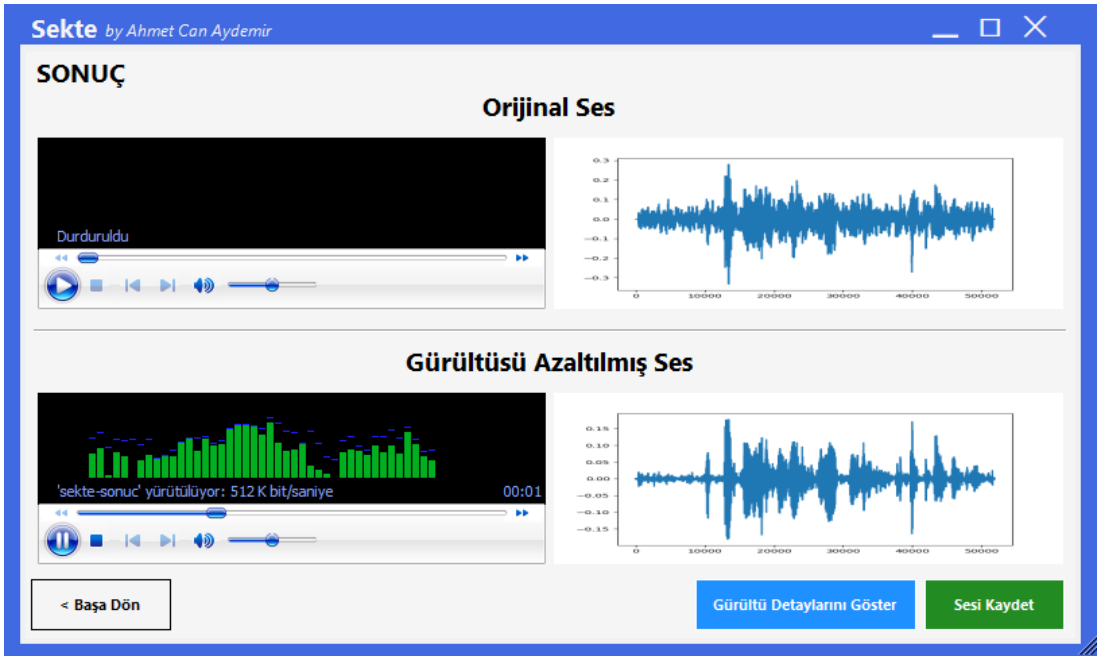
Şekil 2. Gürültünün detayları. Spektogram ve ses bölgelerinin kategorilere ayrılması

Elde edilen iki adet modelin farklı kaynaklardan birbirlerinden bağımsız olarak kullanılabilmesi hedeflenmiştir. Bunun için özellikle modellerin çalışması için gereken yapay zeka kütüphanelerinin sorunsuz çalıştığı Python programlama dilini kullanan çok basit düzeyde bir sunucu yazılmıştır. Sunucu gelen istek üzerindeki gürültülü mp3, wav gibi formatlardaki ses dosyasını almakta ve bu iki modelden ayrı ayrı geçirmektedir. Sonuç olarak gürültüsüz ses sinyali, gürültülü ve gürültüsüz dalga boyu görüntüleri, gürültülü sesin spektogramı ve son olarak gürültülü sesin hangi bölgesinde hangi kategoride gürültü olduğu gibi verileri yanıt olarak geri göndermektedir.



Şekil 3. Gürültülü sesi sunucuya yükleme arayüzü

Bu sunucu kullanılarak mobil uygulamalar, web uygulamaları ve masaüstü uygulamaları farklı alanlarda geliştirilebilecektir. Projede üretilen modeller ve sunucu kullanılarak, kullanıcının gönderdiği ses dosyalarındaki gürültüyü azaltmak için bir masaüstü uygulaması geliştirilmiştir. Uygulamada sesler sürüklendikten sonra yüklenen sesin ön izlemesi gösterilmektedir. Daha sonra ses sunucuya gönderilmekte ve sonucunda kullanıcıya gürültüsü azaltılmış ses dinletilmektedir. Kullanıcı dalga boylarındaki farklılıkları görebilmekte ve detayları görmek için resimleri büyütebilmektedir. Gürültülü ve gürültüsü azaltılmış hallerini ayrı ayrı dinleyip farkı görebilmektedir. Gürültüsüz ses dosyasını bilgisayarındaki herhangi bir klasöre kayıt edebilmekte ve gürültünün detaylarını ve spektogramını da program üzerinden inceleyebilmektedir.



Şekil 4. Sunucu tarafında modelin ürettiği sonucun kullanıcıya gösterilmesi

PROJENİN AMACI ve ÖNEMİ

Projenin Amacı:

Kullanıcının yüklediği gürültülü sesin, derin öğrenme yöntemleri ile oluşturulmuş model tarafından gürültüsü olabildiğince azaltılmış veya tamamen giderilmiş yeni ses dosyası haline getirilmesi ve kullanıcıya sunulmasıdır.

Projenin Önemi:

Konuşma sırasındaki gürültüyü azaltmak uzun süredir devam eden bir sorundur. Gürültülü bir giriş sinyali verildiğinde amaç, söz konusu gürültüyü ilgili sinyali bozmadan filtrelemektir. Arka planda bir parça müzik çalarken birinin video konferansta konuştuğu düşünülebilir. Bu durumda, bir konuşma sırasındaki gürültüyü azaltma sistemi, konuşma sinyalini iyileştirmek için arka plan gürültüsünü kaldırma görevine sahiptir. Diğer birçok kullanım durumunun yanı sıra bu uygulama, gürültünün konuşma anlaşılırlığını önemli ölçüde azaltabileceği video ve sesli konferanslar için özellikle önem arz etmektedir.

Konuşma sırasındaki gürültüyü azaltmak için klasik çözümler genellikle donanımsal çözümlerle veya Gauss karışımları gibi istatistiksel yöntemler ile ilgili gürültüyü tahmin etmekte ve daha sonra gürültüyü gideren sinyali geri kazanmayı amaçlamaktadır. Bununla birlikte son gelişmeler, verilerin mevcut olduğu durumlarda, derin öğrenmenin genellikle bu çözümlerden daha iyi performans gösterdiğini göstermiştir.

Sekte projesinde, Evrişimli Sinir Ağları (CNN) kullanarak sesteki gürültünün giderilmesi problemi ele alınmıştır. Gürültülü bir giriş sinyali verildiğinde, temiz sinyali çıkarabilen ve kullanıcıya geri gönderebilen istatistiksel bir model oluşturulmaktadır. Sekte ile düzgün konuşma sinyallerinin, sokak ortamında sıklıkla bulunan on farklı gürültü türünden ayrılmasına odaklanılmıştır.

Sekte projesinde gürültünün giderilmesinin yanında gürültünün detaylarını da incelenebilmesi de hedeflenmiştir. Sesin hangi kısımlarında araba sesi, hangi kısımlarında konuşma sesi var gibi kategorilere ayrıştırma işlemleri de sekte projesinde görülebilmektedir.

KAYNAK ARAŞTIRMASI

Proje genel olarak “krisp” adlı yazılım ile benzerlik göstermektedir. Bu yazılım yapay zeka kullanarak gerçek zamanlı olarak sesteki gürültüyü azaltmaktadır.

Google Colab, bu tür yüksek veri işleme gerektiren yapay zeka modelleri üretilmesinde ücretsiz olarak GPU gücünden faydalanılabilecek ücretsiz bir araçtır.

Common Voice projesi makinelere gerçek insanların nasıl konuştuklarını öğretmek için Mozilla’nın başlattığı bir girişimdir.

UrbanSound8K veri seti, 10 sınıftan 8732 etiketli küçük ses alıntısı (<= 4s) içerir. Etiketler klima, korna sesleri, çocuk sesleri, köpek havlaması, delme, araba motoru, silah sesi, delici çekiç, siren ve sokak müziğidir.

LibROSA, müzik ve ses analizi için bir python paketidir. Ses sinyali üzerinde dönüşümleri yapmak için gereken yapı taşlarını sağlar.

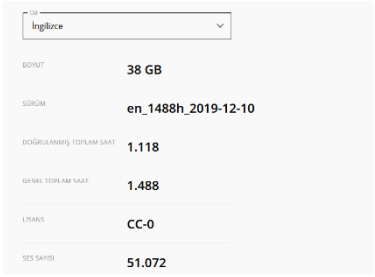
AudioSet, 632 sesli olay sınıfının genişleyen bir ontolojisinden ve YouTube videolarından çizilen 2.084.320 insan etiketli 10 saniyelik ses kliplerinden oluşan bir koleksiyondan oluşur. Yamnet ise bu veri setini kullanan açık kaynak kodlu bir kütüphanedir.

Flask, Python'da yazılmış bir mikro web çerçevesidir. Belirli bir araç veya kütüphane gerektirmediği için bir mikro çerçeve olarak sınıflandırılır. Herhangi bir veritabanı soyutlama katmanı, form doğrulaması veya önceden var olan üçüncü taraf kitaplıklarının ortak işlevler sağladığı diğer bileşenleri yoktur.

MATERYAL VE METOT

Projenin kodlaması farklı aşamalardan oluşmaktadır. Bu aşamalarda farklı teknolojiler ve programlama dillerinden faydalanılmıştır. Seslerin hazırlanması ve özelliklerinin çıkartılması konusunda Python programlama dilinden faydalanılmış ve librosa isimli ses işleme kütüphanesi Kısa Zamanlı Fourier Dönüşümü gibi ses sinyali işleme konusunda kullanılmıştır. Keras ve TensorFlow ise özelliklerin tfrecord dosyaları olarak kayıt edilmesi için tercih edilmiştir.

Gürültülü sesler için 8.732 adet etiketlenmiş ve 10 gruba ayrılmış ses dosyasından oluşan "UrbanSound8K" veri seti tercih edilmiştir. Gürültüsüz konuşma sesleri için ise 51.073 adet İngilizce konuşma kaydından oluşan "Mozilla CommonVoice" veri seti tercih edilmiştir. Kullanılan iki veri seti de açık kaynak kodludur ve herkes tarafından kullanılabilir ve değiştirilebilir. Bunun yanında gürültünün detayları için de Google'ın açık kaynak kodlu "AudioSet" veri setinden yararlanılmıştır.



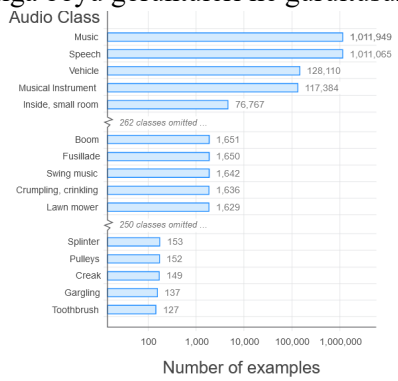
Şekil 5. Mozilla CommonVoice veri seti detayları

Bu gürültülü ve gürültüsüz ses kayıtlarının özellikleri çıkartılmış ve kayıt edilmiştir. Kayıt edilen dosyalar kullanılarak CNN ile model üretilmiştir. Model üretimi ve test işlemleri, model %15 civarında kayıp verene kadar tekrarlanmış ve eğitim devam ettirilmiştir. Sonucunda oluşan TensorFlow modeli kayıt edilmiştir.

Yaklaşık 90 GB gürültü özellik kayıt dosyalarının normal bir bilgisayarda işlenmesi çok zor olacağından bu dosyalar Google Drive'a yüklenmiştir. Daha sonra Google'ın ücretsiz sunduğu yüksek yapay zeka eğitim işlem kapasitesine sahip ve ücretsiz ekran kartı özelliklerinden faydalanılabilecek Google Colab kullanılmıştır. Google Colab üzerinde TensorFlow GPU üzerinde defalarca çalıştırılıp istenilen eğitim düzeyine gelinceye kadar model Google Drive içindeki dosyalar ile eğitilmiştir.

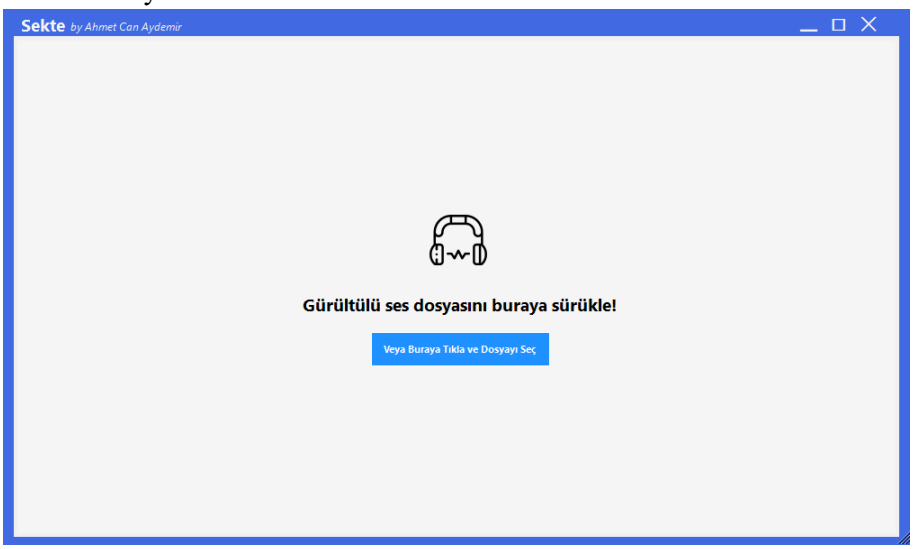
Gürültünün özelliklerinin ayrıştırılması için Google AudioSet veri setini kullanan Google'ın açık kaynak kodlu yamnet isimli modeli kullanılmıştır.

Kullanıcının gönderdiği gürültülü ses dosyalarının işlenmesi için flask sunucusu oluşturulmuştur. Bu flask sunucusunun amacı sekte modelini ve Google AudioSet'de paylaşılan yamnet modellerini çalıştırmaktır. Flask sunucusuna gönderilen gürültülü ses dosyasının özellikleri çıkarılıp modele girdi olarak verilmektedir. Model TensorFlow üzerinde çalışmakta ve çıktı olarak yeni bir ses dosyası üretmektedir. Sunucu üzerinden yanıt olarak gürültülü ve gürültüsü azaltılmış sesin dalga boyu görüntüleri ile gürültüsüz ses gönderilmektedir.



Şekil 6. Audioset veri seti sınıfları

Kullanıcı arayüzü için C# kullanılarak bir masaüstü uygulaması yazılmıştır. Bu uygulama kullanıcıdan gürültülü ses dosyasını almakta ve flask sunucusuna istek yapmaktadır. İstek sonucunda gelen yanıttaki görüntüleri ve ses dosyalarını ise tekrar kullanıcıya sunmaktadır.

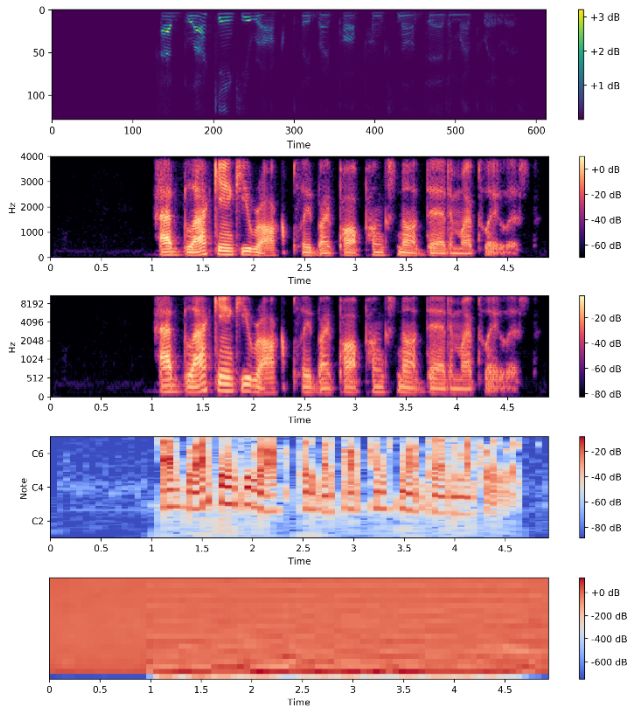


Şekil 7. Masaüstü uygulaması kullanıcı arayüzü

Ses Sinyali Özelliklerinin Çıkarılması

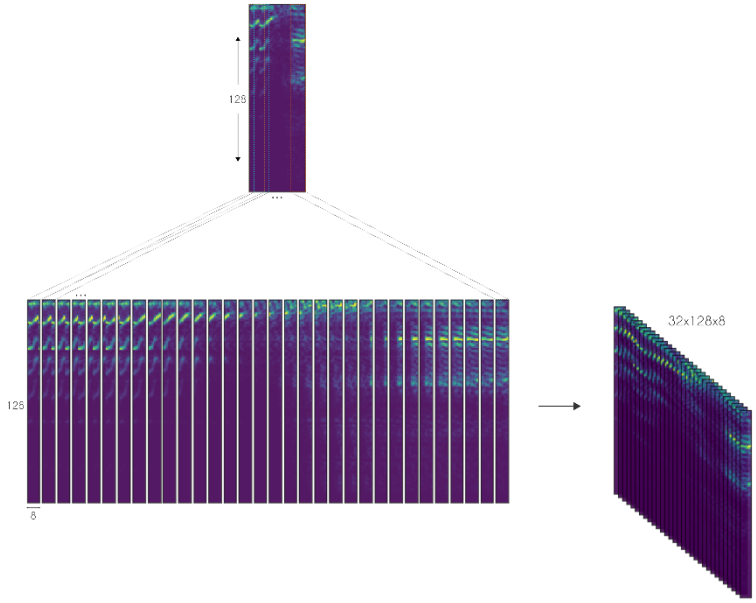
Sesleri kullanarak bir model oluşturmak için ilk olarak özelliklerinin çıkartılması gerekmektedir. Bunun için ilk olarak her iki veri setindeki ses sinyallerini 8KHz' indirip sessiz bölgeleri silinebilir. Bunun sonucunda hesaplama miktarı ve veri setinin boyutu azalmış olacaktır.

Ses sinyalleri genel olarak zaman/frekans olarak 2 boyutlu gösterilmelere dönüştürülürler. MFFC ve Q spektrumu ses uygulamalarında sıklıkla kullanılan gösterimlerdendir. 256 noktalı Kısa Süreli Fourier Dönüşümü (STFT) kullanarak hesaplanan spektral büyüklük vektörleri ile gürültülü sesin özellikleri çıkarılabilir. Bunun için Python'da librosa kütüphanesinden faydalanılmıştır.



Şekil 8. Ses verilerinin ortak 2B gösterimleri. Yukarıdan aşağıya:
(1) STFT büyüklük spektrumu; (2) Spektrogram; (3) Me-spektrogramı; (4) Sabit-q; (5) MFCC'ler

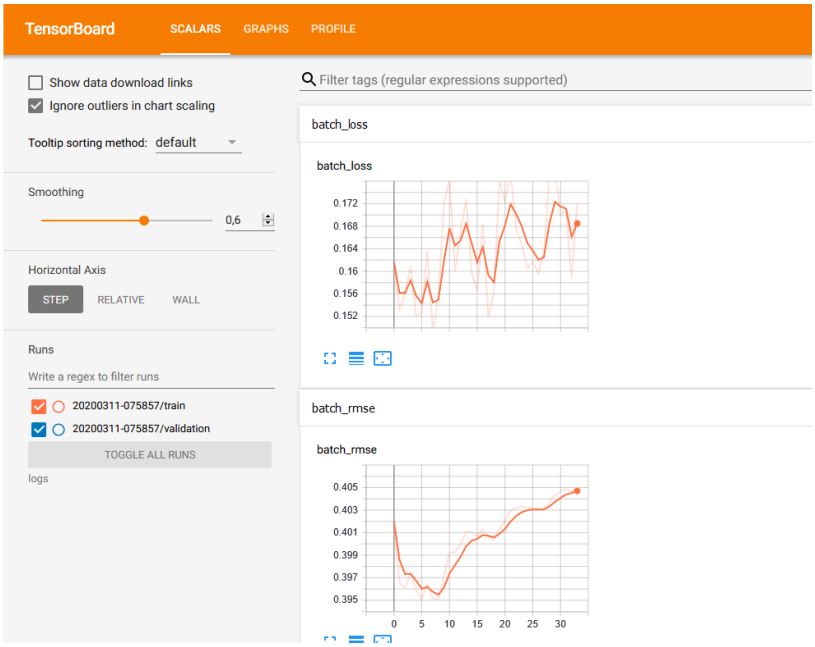
STFT vektörleri arasında %75 oranında çakışma olmamasını sağlamak için uzunluğu 256 ve atlama boyutu 64 olan periyodik bir Hamming Penceresi tanımlanmıştır. Sonuç olarak birbirini takip eden sekiz gürültülü STFT vektörü birleştirilmiştir ve bu sonuç girdi olarak kullanılmıştır. Dolayısıyla bir giriş vektörü (129,8) şekline sahiptir ve mevcut STFT gürültülü vektörü ile daha önceki 7 gürültülü STFT vektöründen oluşur. Başka bir deyişle, model geçmiş gözlemlere dayanarak mevcut sinyali tahmin eden bir sistemdir.



Şekil 9. Ses sinyalinin vektörlere ayrılıp ardışık olarak özelliklerinin çıkarılma prototipi

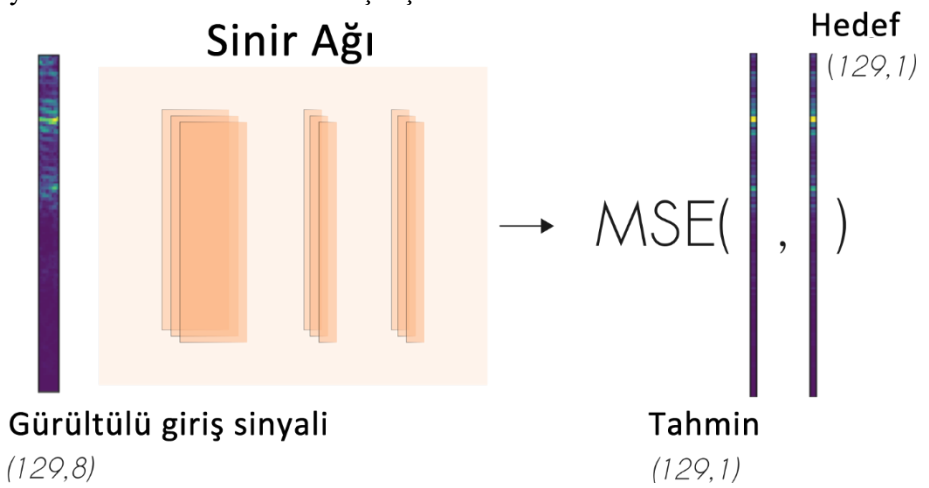
Modelin Üretilmesi

Projede kullanılan Derin Evrişimli Sinir Ağı, Yedekli Evrişimli Enkoder-Kod Çözücü Ağı'nı (CR-CED) temel almaktadır. Model, simetrik kodlayıcı-kod çözücü mimarilerine dayanmaktadır. Her iki bileşen de tekrarlanan Convolution, ReLU ve Toplu Normalizasyon bloklarını içerir. Toplamda, ağ bu tür bloklardan 16 tane içerir ve bu da 33 bine kadar parametre ekler.



Şekil 10. Model üretimi sırasındaki kök ortalama kare sapması ve kayıp oranı

Ayrıca, bazı kodlayıcı ve kod çözücü blokları arasında atlama bağlantıları vardır. Bu atlama bağlantılarında her iki bileşenden gelen özellik vektörleri toplama yoluyla birleştirilmektedir. Az sayıda eğitim parametresi ve model mimarisi sayesinde mobil cihazlarda bile çalışabilecek hafif bir model sonucu üretmektedir.



Şekil 11. Ses vektörlerinin sinir ağında eğitiminin prototipi

Sunucunun Çalıştırılması

Sunucunun çalıştırılabilmesi için öncelikle Windows işletim sistemi için "SET FLASK_APP=server.py" kodu ile flask sunucunun hangi dosya üzerinden çalıştırılacağı değişken ile kayıt edilmelidir. Daha sonra flask run komutu python modülü olarak çalıştığında sunucu istekleri almak için hazır olacaktır. Bu aşamadan sonra masaüstü yazılımı üzerinden gürültülü bir ses gönderilmesi sunucuya istek yapmak için yeterli olacaktır.

```
(base) E:\Google Drive\Sekte\Backend>SET FLASK_APP=server.py

(base) E:\Google Drive\Sekte\Backend>python -m flask run
 * Serving Flask app "server.py"
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
2020-06-11 22:40:31.973179: I tensorflow/core/platform/cpu_feature_guard.cc:142] Your CPU supports i
tions that this TensorFlow binary was not compiled to use: AVX AVX2
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Şekil 12. Çalışan sunucunun komut satırı arayüzü

KAYNAKLAR

1. DaitanGroup, 2019, How To Build a Deep Audio De-Noiser Using TensorFlow 2.0 https://medium.com/better-programming/how-to-build-a-deep-audio-de-noiser-using-tensorflow-2-0-1aea299
2. Se Rim Park, Jinwon Lee, 2016, A Fully Convolutional Neural Network for Speech Enhancement https://arxiv.org/abs/1609.07132
3. Google AudioSet https://research.google.com/audioset/dataset/index.html
4. YAMNet https://github.com/tensorflow/models/tree/master/research/audioset/yamnet
5. Cnn-audio-denoiser https://github.com/daitan-innovation/cnn-audio-denoiser
6. Davit Baghdasaryan, 2018, Real-Time Noise Suppression Using Deep Learning https://devblogs.nvidia.com/nvidia-real-time-noise-suppression-deep-learning/
7. Google Colab https://colab.research.google.com