Audio Classification of Normal Traffic and Siren Sounds: Spectrogram Encoding and Raw Audio Feature Extraction Approaches

Anushka Kulkarni [1] **Justin Zeng** [1] **Ka Fung Tjin** [1] **Priyanka Rose Varghese** [1] **Rahul Murugan** [1]

Abstract

As cities grow at an unprecedented rate, urbanization has led to increased traffic congestion, worsening delays for emergency responders, and putting lives at risk. A 2023 study, highlights how rapid urban growth is directly tied to traffic volume, creating critical challenges for public safety infrastructure to keep pace (1). In this context, developing an AI-powered adaptive traffic system is no longer a luxury; it is a necessity. The growing urgency caused by urbanization makes this an essential investment for smart cities to enhance public safety and manage traffic effectively (1)

We aim to create the best model that will predict if a siren audio segment is from an emergency vehicle or general road noise. In this experiment, we will look at two different ways of transforming audio data inputs: extracting acoustic features and converting the data to visual spectrograms. We will train and compare various models and their performance on a large real-world dataset.

1. Data Analysis

1.1. Audio Feature Dataset

For our audio data, we used the Large-Scale Audio Dataset for Emergency Vehicle Sirens and Road Noises dataset, consisting of 18,000 audio segments of 2.7 GB total. (2). It is well balanced, with the Ambulance class having (m=932) & Road class having (m=903) samples.Researchers constructed this dataset from existing datasets, including Google AudioSet which is collected from YouTube videos and UrbanSound which is collected from street sound field recordings. They also supplemented this data with voice-enabled camera node recordings and experimental setup recordings collected in Karachi, Pakistan. The strength of the data is that they are large and collected from many different sources, which means that they are generalizable. Features: The data is annotated with features collected from the Python Librosa library. Chroma_stft is the audio chromogram, mapping changes in pitch. Spectral centroid indicates the center of mass of the audio spectrum, highlighting the most prominent sounds. Roll-off rate refers to the cutoff frequency of a subset of data, removing outliers and provide a clearer sense of normal frequency. Spectrum bandwidth measures the range of frequencies within a continuous frequency band, reflecting frequency variation. Zero crossing rate represents the frequency of sign changes in the signal. Lastly, MFCCs capture the signal's frequencies at the first 21 Mel frequencies, conveying overall frequency shape.

We plotted the above graphs shown in **Figure 1 (Audio Feature Median Values For Ambulance vs. Road Noise (log scale))**and **Figure 2 (Boxplots of MFCC Feature Values For Ambulance vs. Road Noise)** to understand the distribution of features and important features. **Figure 1** shows the audio feature median values for ambulance vs. road noise in log scale. Here, we can see that ambulance noises seem to have a lower chroma sfft, mfcc5, mfcc6, mfcc13, mfcc15, mfcc19 and higher spectral centroid and mfcc8 means than road noise. Since features mapped in log scale, negative values were discounted. **Figure 2** shows boxplots of MFCC features for ambulance vs. road noises since some of these values are negative and would not be shown properly in the first graph. Outliers were also removed using the IQR method. Here, we observe that while the medians differ, the first to third quartile ranges for all features in road and ambulance noise overlap, indicating that the feature values for road and ambulance noise remain somewhat similar so the classification task is challenging without a ML.

1.2. Visual Dataset

For the visual representation of audio data, we processed each audio file to generate Mel spectrograms using the Librosa library. The dataset consists of 1,835 spectrogram images derived from 932 ambulance siren audio files and 903 road noise audio files. Each spectrogram encodes the frequency spectrum of the audio over time and serves as a visual representation of the sound, saved in high-resolution .png format to preserve acoustic feature details.

The spectrograms were generated using an FFT window size of 2048, a hop length of 512, and 128 Mel bands. Each spectrogram was normalized to a decibel scale to enhance the contrast between low- and high-intensity sounds, ensuring the patterns distinguishing sirens from road noise

are clearly visible. The axes represent frequency bands (y-axis in Hertz) and time intervals (x-axis in seconds), with amplitude intensity encoded as colors on the decibel scale.

Annotations link each spectrogram to its original audio file, corresponding label (ambulance or road), and additional features extracted using Librosa, such as MFCCs and spectral properties.

1.3. Data Analysis

The dataset contains 1,835 samples, evenly distributed between ambulance sirens (932 samples) and road noises (903 samples). Each .wav file was loaded with its original sampling rate, and the audio signals were transformed into Mel spectrograms using the short-time Fourier Transform (STFT). These spectrograms were then normalized to a decibel scale using Librosa's power_to_db function and visualized with Matplotlib for training purposes.

Ambulance sirens exhibit repetitive and periodic peaks in specific frequency bands, reflecting their oscillating nature. In contrast, road noise shows irregular and scattered patterns, indicative of its randomness. The high-resolution spectrograms preserve intricate audio details, making them suitable for training image-based models. This approach bridges audio data and visual classification tasks, providing a robust dataset for distinguishing between emergency sirens and general road noise.

2. Methods/Models

Our two approaches are based on different ways to preprocess data, which allows us to train different models.

2.1. Audio classification as an audio feature classification problem:

We will train two models: a tree-based Random Forest Model and a non-linear Kernel SVM on the extracted features.

Data-Preprocessing: Since there was no large class imbalance, we chose random train/test splitting. We also applied standard scaling normalization on the features, since some hard larger values than others. We had initially intended on collecting more features from the dataset, but upon evaluating the performance of our models and further research on important features for audio ML, we decided the existing feature set was sufficient.

Random Forest Model: Random forest models are ensemble methods that employ random bagging to split data into independent subsets and train decision trees from each subset. They work well with small datasets without fitting to noise and are interpretable. We trained a random forest model with an 80-20 split of the data. We tuned the following hyperparameters: n_estimators, max_depth, min_samples_split, min_samples_leaf, max_features, criterion, class_weight using both a Random and Grid Search approach. Both search methods had a similar performance, but we decided on the following model:

The following model was used:

```
random_forest_model =
↪  RandomForestClassifier(
    class_weight=None,
    criterion='entropy',
    max_depth=25,
    max_features='sqrt',
    min_samples_leaf=2,
    min_samples_split=5,
    n_estimators=250)
```

2.2. Audio classification as an image classification problem using Resnet18

Classifying images has been made easier with the release of ImageNet dataset (3) and several open-source pre-trained models, such as ResNets, EfficientNets, DenseNets, etc. Traditional audio classification (as demonstrated in the previous section) works on manually extracted numerical/categorical features from the audio, for example, Mel Frequency Cepstral Coefficients (MFCCs), spectral centroid. However, due to the success of image classification, one method to classify audio is by converting it into images, specifically Mel's Spectrograms (4).

For classifying these MEL's Spectrograms, we have selected the pretrained Resnet18. This model belongs to the family of ResNet models (1), pre-trained on the ImageNet dataset. ResNet-18 consists of a series of convolutional layers with skip connections to skip intermediate layers. The architecture of ResNet-18 is shown in **Figure 4: ResNet Architecture.** (5)

After the audio files are converted to Mel spectrograms and divided into train, validation and test datasets, we input them into data transformation pipelines with Pytorch's inbuilt transforms module. After data transformation, they are fed into the Resnet architecture in batches of 32.

The learning rate used for training this model is 1e-4 (without regularization). As this is a binary classification problem, Binary Cross-Entropy is used as the loss function. Our network training takes place in 2 phases. In Phase I, plotted in **Figure 5: Phase 1 - Training and Validation Loss by Epoch Curve** we freeze all the weights of the model except for the last fully connected layer and train for 20 epochs on this layer. This phase is often referred to as the feature extraction phase, where the fully connected layer captures important features from the images. In Phase II, plotted in **Figure 6: Phase 2 - Training and Validation Loss by**

Epoch Curve, we unfreeze the entire network's weights and train all the parameters (finetuning).

3. Results

3.1. SVM

We chose to use Sklearns' Linear Support Vector Classification (LinearSVC) for its ability to handle high-dimensional feature spaces and its robustness in linearly separable data. Given that the dataset contains a large number of features, some of which may not be directly interpretable, SVM was chosen over simpler linear models like logistic regression or linear classifiers, as SVMs are better equipped to handle datasets where the decision boundaries are not straightforward. By leveraging both dual and primal optimization modes, we aimed to explore computational efficiency and performance differences between the two approaches.

3.2. Random Forests

3.2.1. Classification Report

The classification report is given in (Figure 1)

	precision	recall	f1-score	support
Ambulance	0.96	0.97	0.97	187
Road	0.97	0.96	0.96	180
accuracy			0.96	367
macro avg	0.96	0.96	0.96	367
weighted avg	0.96	0.96	0.96	367

Figure 1. Classification Report-Random Forests

3.2.2. Analysis

The random forest model performed better than the SVM model, with a 0.96 accuracy and an f1-score of 0.96 for the negative class (Road). This is likely due to the fact that the random forest is an ensemble method that generates trees based on random bagging subsets on the data, allowing it to pick up on important features in the data while being less vulnerable to noise in the dataset even in small datasets. The tree is visualised in **Figure 3: Visualization of one tree in the Random Forest Model** The random forest model also allowed us to collect feature importances on the data, providing us key interpretability as we see the most important features are chroma stft, spectral centroid, and rolloff. chroma stft is by in large the most important. The feature importances from Random forests have been plotted in **Figure 7: Feature Importances for Random Forest Model**

3.3. ResNet

3.3.1. Classification Report

The classification report is shown in (Figure 2)

	precision	recall	f1-score	support
Ambulance	1.00	1.00	1.00	148
Road	1.00	1.00	1.00	139
accuracy			1.00	287
macro avg	1.00	1.00	1.00	287
weighted avg	1.00	1.00	1.00	287

Figure 2. Classification Report of Resnet18

3.3.2. Analysis

Resnet18 performs extremely well on image classification tasks due to the fact that skip connections improve the gradient flow and improves convergence. Since Resnet18 is pre-trained on ImageNet, it is efficient in feature extraction from images. It works effectively even on image classification tasks with limited training data.

As seen in the above classification report our fine tuned model is able to achieve a 100 percent accuracy on unseen data. It predicts all 148 ambulance samples and all 139 road samples correctly. This shows that our Resnet18 model performs exceptionally well even on unseen data. The confusion matrix of the Resnet 18 model can be seen in **Figure 8: Confusion Matrix of ResNet 18 model**

References

[1] Zhang X. Ren S. Sun J He, K. Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[2] Muhammad Asif, Muhammad Usaid, Sheikh Muhammad Rashid, Tabarka Rajab, Samreen Hussain, and Sarwar Wasi. Large-scale audio dataset for emergency vehicle sirens and road noises. *Scientific Data*, 9, 10 2022.

[3] R. Socher L. J. Li Kai Li J. Deng, W. Dong and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[4] J. Volkmann S. S. Stevens and E. B. Newman. A scale for the measurement of the psychological magnitude pitch. *Current Opinion in Behavioral Sciences*, 8:185–190.

[5] Farheen Ramzan, Muhammad Usman Khan, Asim Rehmat, Sajid Iqbal, Tanzila Saba, Amjad Rehman, and Zahid Mehmood. A deep learning approach for automated diagnosis and multi-class classification of alzheimer's disease stages using resting-state fmri and residual neural networks. *Journal of Medical Systems*, 44, 2019.