



university of
groningen

faculty of science
and engineering

VehicleNET: Predicting Vehicle Type based on Engine Sound

Applied ML Project Proposal

WBAI065-05.2024-2025

Title of Deliverable:	Project Proposal
Group Number:	7
Author(s):	Alexandru Baris Eroglu (S5539099) Bogdan Sandoiu (S5549329) Rares Medelet (S5608902) Gunes Saner (S5098041)
Delivery Date:	May 2, 2025

Contents

1 Preliminary Domain Knowledge	3
2 Preliminary Data Exploration	3
3 Proposed Preprocessing	5
4 Proposed Model + Baseline	5
5 Proposed evaluation	5
6 Model Usage	6
7 Risk Assessment	6
8 Individual Learning Outcomes	6
9 Relation to Grading Specifications	7
10 Member contributions	7

1 Preliminary Domain Knowledge

Sound is often the first, and sometimes the only, clue that a vehicle is approaching. A single microphone can flag a speeding car in thick fog for a smart-city traffic system, or give troops early warning of an incoming armored vehicle on the battlefield. Automatic vehicle sound classification turns those raw decibels into important intelligence, whenever cameras or radars fail. These classification algorithms are widely used for various purposes, mainly for urban planning and noise level detection (Damiano et al., 2024). Implementation in law enforcement is gaining attention as it can be implemented to detect non conforming vehicles and fine them (Cheng et al., 2023). Military use is also in discussion, starting to be used near bases to classify armored vs light vehicles in proximity of borders. Implementation is effective due to the lower costs of microphones compared to installation of CCTV systems.

IDMT-Traffic is a dataset released by Fraunhofer IDMT in 2021, composed by 2.5 hours of stereo audio recordings of 4718 vehicle passing events (Abeßer et al., 2021). The accompanying paper provides lightweight CNN baselines that reach roughly 93 % accuracy on a four-class classification task, giving future work a clear reference point. We aim to test if the combination of CNN (for feature extraction from spectrograms) and RNN (for capturing temporal dependencies in manually extracted features) provides a significant performance boost over using either model alone, particularly in terms of handling complex, noisy audio signals. We are confronted with data imbalance in our dataset among 8 different classes of vehicle types (Boubi, 2024), so we aim for a robust implementation with an F1 weighted of **minimum** 80 % for each class, including minority classes. For this project, we set two broad performance targets. First, the system should be dependable overall, reaching roughly 90 % weighted-F1 on a held-out test set. Second, we want to deliver clear added value over a traditional pipeline, so the final score should exceed the baseline by at least ten weighted-F1 points.

2 Preliminary Data Exploration

There are 8 classes in the dataset, each representing a different type of vehicle: Car, Truck, Bus, Train, Bicycle, Motorcycle, Airplane, and Helicopter. However, the dataset is pretty unbalanced and have a lot of noise in it(check the waveform and spectrogram plots). For example, Bus and Train classes dominate with 4,221 and 2,552 audio samples compared to Car (230), Truck (265). Some outliers are present in the dataset, particularly within the bus class. Because the audio appears to have been recorded from inside the vehicle, there are some samples in which you can hear background voices from passengers or the driver. There are no features yet in our dataset, only the audio samples, but we mentioned how to extract them in the following section.

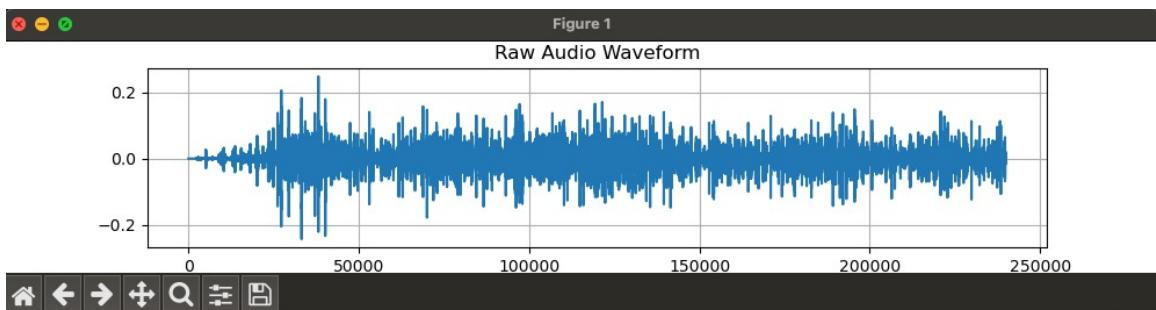


Figure 1: Raw audio waveform of an airplane. The signal showing the typical engine noise during flight with minimal variation.

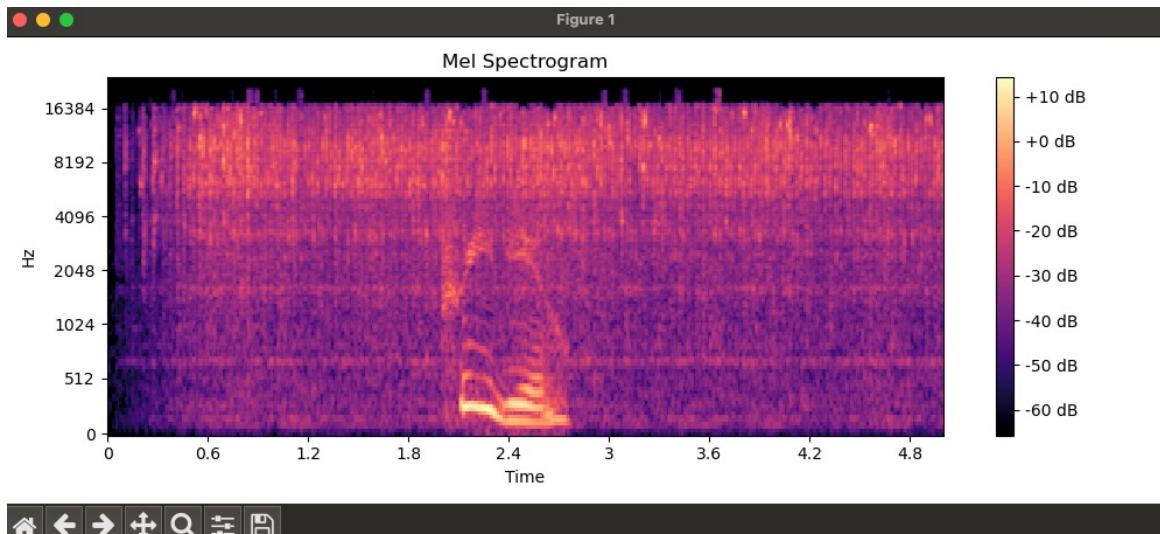


Figure 2: This Mel spectrogram shows the sound of an airplane over 5 seconds. Most energy is concentrated in the low to mid frequencies. A brief increase in intensity appears around 2–2.5 seconds, likely due to engine variation or background noise.

We applied t-SNE to reduce the high dimensional frequency-domain representations of our audio data into a 2D space for visualization.

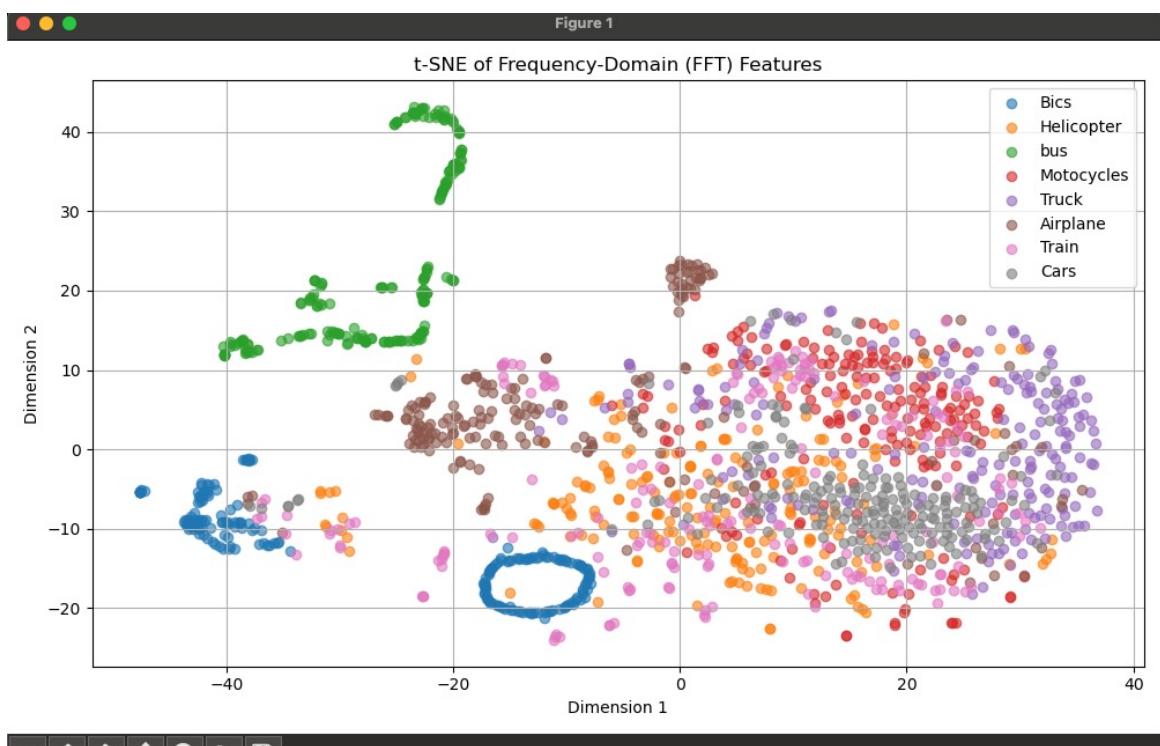


Figure 3: This t-SNE plot visualizes vehicle sounds in a 2D space using frequency-domain features. Some classes like Bus, Bics, and Airplane form somehow, separate clusters, indicating distinct frequency characteristics. Others like Train, Truck, Cars and Motorcycles show more overlap, suggesting similar sound patterns.

3 Proposed Preprocessing

We will find a sampling rate suitable for our whole dataset to resample all the audio samples. In terms of features extractions, we will use the **librosa** library to extract from each audio sample a 3-channel image tensor of shape (3, 128, 469) to serve as input for a Convolutional Neural Network(CNN). These three channels correspond to different time-frequency representations: the Mel spectrogram, which is a time-frequency representation of an audio signal that shows how the intensity of different frequencies changes over time; the Mel-Frequency Cepstral Coefficients(MFCC), which describes how the energy of the sound is distributed across frequencies; and the Delta MFCC, which represents the rate of change of the MFCCs over time and captures transitions in the audio signal. We will apply log normalization to the spectrograms (Mel, MFCC, Delta MFCC) to compress the range. Additionally, we will extract manually crafted features such as Root Mean Square , Zero Crossing Rate, spectral centroid, bandwidth, rolloff, and chroma from each audio sample. These features will be used alongside the spectrogram representations to provide additional insights into the sound and will represent the input of Recurrent Neural Network(RNN).

Furthermore, we will deal with the unbalanced dataset by applying weights to the classes. Moreover, the dataset will be split into training -70%, validation -15%, and test -15% sets based on the number of samples in each class. To reduce the noise of our dataset(that we observed in the plots of Question 2, both waveform and spectrogram) in our audio samples, we can apply a frequency filter within a specific range. However, the noise is good to some extent in order to generalize the performance.

4 Proposed Model + Baseline

For our baseline models, we decided to train Convolutional Neural Networks and Recurrent Neural Networks individually, using either spectrogram-based features (CNN) or manually extracted features (RNN). The CNN will learn local spatial patterns from the log spectrogram "images" created from the audio clips, while the RNN will process the temporal dependencies in the manually extracted features. The CNN will be modified with a softmax activation function in the final layer to classify the input into one of the 8 classes, similar to the RNN. These models will serve as the baseline for comparison against the main model, where we combine both CNN and RNN layers . The hybrid model will take advantage of the strengths of both approaches. After each network (CNN and RNN) processes the input data, we concatenate their outputs into a single combined feature vector, feed it into an Multi Layer Perceptron(MLP), and apply again a softmax function to classify into one of the 8 classes. We will grid-search hyperparameters including learning rate, batch size, CNN layer configs, RNN hidden size, kernel size, and dropout rate (0%, 20%, 40%) to control overfitting, especially on minority classes. The CNN and RNN models will be compared to the hybrid model, with the expectation that the hybrid model will deliver at least a 10% improvement in F1 score.

5 Proposed evaluation

We will initially use accuracy as the primary metric to evaluate the overall performance of the model on the test set and a confusion matrix to observe how often classes are confused with each other. Moreover, given the class imbalance, F1-score will be an important secondary metric because is evaluating the performance on minority classes, where accuracy might be misleading. However, we can also take into consideration the evaluation of the model's size to ensure that it's not too large for deployment or inference(including the number of parameters and overall memory usage).

In order to deal with overfitting and underfitting we will monitor the training and validation loss curves over time. Moreover, we will also use k-fold cross-validation to avoid overfitting.

6 Model Usage

Our lightweight eight class vehicle sound classifier works best when embedded in simple edge devices. These devices can use a small "before/after" setup: a pre-trigger to filter out non-vehicle sounds, and a post-processor to convert the soft-max output into an action. Imagine the code running on a solar-powered Raspberry Pi along a rural road, helping people that protect and preserve the environment to track traffic for wildlife studies 24/7. In a military context, at a remote base, a small acoustic node could filter out things like speech and gunfire, then pass the clip through the model. If the model is more than 70% sure the sound is an armored vehicle or truck, it sends an alert to point cameras in that direction, while harmless sounds are just logged. These are simple and effective setups that are easy to implement, especially since microphones are cheap and durable.

7 Risk Assessment

Our implementation of the model includes a classification process, which is done with the help of the recorded frequencies. The recordings in which the engine is not the only thing present in the audio(noise) are going to provoke changes in frequencies, with the risk of misclassification. We aim to address the issue of unwanted noise by applying frequency filtering. However, any noise that cannot be fully eliminated may actually help the model generalize better for real world environments. Moreover, class imbalance could also be an issue, but we will deal with it maybe by under sampling the dataset or adding weights to the classes. The deployment of the model can come at the cost of having even more noise in a real-life situation and not being able to classify properly for each class. In case of failure, we want to implement a model that can approve or reject a person that is applying for a loan based on multiple features such as income, loan amount, etc. . However, we are extremely confident that our main idea will work.

8 Individual Learning Outcomes

1. Rares: I want to apply what I learned so far from the courses Introduction to Machine Learning and Signal and Systems to a project that is based on different frequencies.
2. Bogdan: I want to learn how to use **librosa** library in order to extract different features from an audio signal and manipulate them for a specific task. Moreover, I want to expand my knowledge in terms of Deep Learning architectures throughout the usage of CNNs and RNNs. The last but not least, to learn applying pre-processing methods to different types of datasets like audios in our case.
3. Gunes: I believe this project will help me to apply machine learning models to real-world scenarios and problems, providing further practical experience. While doing so, it will also deepen my understanding of deep neural network (DNN) concepts and models, as we are going to be working with a DNN namely CNNs and RNNs in our case.

4. Baris: From this project I seek to learn how to apply in a practical setting all of the information I have learned in Introduction to Machine Learning and Signals and System. I want to gain more hands on experience in using libraries and gain experience in Pytorch specifically. I also want to improve my knowledge of Git and learn to work in a clean and structured manner.

9 Relation to Grading Specifications

Our project introduces a novel approach to vehicle classification models by combining CNN and RNN architectures, distinguishing it from commonly used approaches like individual CNN/RNN architecture.

Furthermore, we will implement a pre-processing pipeline to make sure that all the data is standardized. First, the whole dataset will be resampled with a unique suitable sampling rate. Then, we will apply a frequency filter within a specific range to avoid the noise as much as possible.

Finally, while optimizing the hyperparameters, we will tune learning rate, batch size, CNN layer configs, RNN hidden size and dropout rate using methods such as grid search.

In addition, to boost the grade, we consider the manually extracted features from our audios to be considered **feature engineering**, after that reducing the dimensionality throughout PCA. Moreover, if we are in time with the project, we tend to implement and out-of-distribution detection(training the main model on 7 classes and leaving one out) to observe the uncertainty of our model. We want to have clean and structured code.

10 Member contributions

We equally distributed the tasks of this project. Most of the time, we worked along at the same time on Discord.

References

- Abeßer, J., Gourishetti, S., Kátai, A., Clauß, T., Sharma, P., & Liebetrau, J. (2021). Idmt-traffic: An open benchmark dataset for acoustic traffic monitoring research. <https://arxiv.org/abs/2104.13620>
- Boubi, A. J. (2024). Vehicle Sounds Dataset [Last accessed 1 May 2025]. Retrieved May 1, 2025, from <https://www.kaggle.com/datasets/janboubiabderrahim/vehicle-sounds-dataset>
- Cheng, K. W., Chow, H. M., Li, S. Y., Tsang, T. W., Ng, H. L. B., Hui, C. H., Lee, Y. H., Cheng, K. W., Cheung, S. C., Lee, C. K., & Tsang, S. W. (2023). Spectrogram-based classification on vehicles with modified loud exhausts via convolutional neural networks. *Applied Acoustics*, 205, 109254. <https://doi.org/https://doi.org/10.1016/j.apacoust.2023.109254>
- Damiano, S., Bondi, L., Ghaffarzadegan, S., Guntoro, A., & van Waterschoot, T. (2024). Can synthetic data boost the training of deep acoustic vehicle counting networks? *Proceedings of the 2024 International Conference on Acoustics, Speech and Signal Processing (ICASSP) (accepted)*.