

# Emergency Vehicles Audio Detection and Localization in Autonomous Driving

Hongyi Sun, Xinyi Liu, Kecheng Xu, Jinghao Miao, Qi Luo\*

**Abstract**—Emergency vehicles in service have right-of-way over all other vehicles. Hence, all other vehicles are supposed to take proper actions to yield emergency vehicles with active sirens. As this task requires the cooperation between ears and eyes for human drivers, it also needs audio detection as a supplement to vision-based algorithms for fully autonomous driving vehicles. In urban driving scenarios, we need to know both the existence of emergency vehicles and their relative positions to us to decide the proper actions. We present a novel system from collecting the real-world siren data to the deployment of models using only two cost-efficient microphones. We are able to achieve promising performance for each task separately, especially within the crucial 10m to 50m distance range to react (the size of our ego vehicle is around 5m in length and 2m in width). The recall rate to determine the existence of sirens is 99.16%, the median and mean angle absolute error is  $9.64^\circ$  and  $19.18^\circ$  respectively, and the median and mean distance absolute error of 9.30m and 10.58m respectively within that range. We also benchmark various machine learning approaches that can determine the siren existence and sound source localization which includes direction and distance simultaneously within 50ms of latency.

## I. INTRODUCTION

### A. Motivation

We have entered an era where fully autonomous driving technology is evolving at a rapid pace. Over the past three years, a record amount of investment flows into the autonomous driving industry. More than 250 tech companies are competing on launching autonomous driving products and services, including robotaxis, robobuses, and self-driving trucks. A handful of automobile giants have already announced their plans to hit the market with L3 autonomy this year. [1] As a result, there is a surging demand for onboard software modules that empower and improve vehicle automation.

Knowing to yield emergency vehicles (EV) on public roads is one of the vital qualities for a fully autonomous vehicle (AV), especially that at L4 and beyond. An AV is expected to take appropriate actions even before an emergency vehicle with an active siren reaches its adjacency. The distance can be larger than the range of object identification by a vision-based perception. More importantly, there is a high chance that vision sensors of AVs are blocked by other vehicles, buildings or trees in urban driving scenarios. Hence,

audio detection for EV acts as a crucial role to complement vision-based object detection algorithms.

Additionally, the cost of hardware installation required for audio-based detection is reasonably low compared to the benefits it can provide. In our experiments, we use the audio data collected by general-purpose microphones that are economical yet still obtain satisfactory performance. Therefore, audio-based emergency vehicle detection is worth more exploration, while the literature in this area is relatively limited.

### B. Related Work

Siren detection and sound localization are separate topics in most of the previous and ongoing research.

1) **Siren Detection**: Some research focuses on the detection of EV siren sound based on machine learning or deep learning techniques. [2] uses waveform and hand-crafted features to train a binary classifier. [3] extracts the spectrum of the audio signal in their SVM-based siren detection model training. [4] uses the module difference function (MDF) then applies fully-connected layers for siren classification. [5] combines both time-domain features and frequency-domain features, and applies convolution neural networks to encode the signal history.

Even though these models achieve good performance on siren existence detection, they are not designed with the capability to detect the location of the sound source which is critical for an AV to determine appropriate actions.

2) **Sound Source Localization**: [6][7][8] apply rule-based algorithms based on the time difference of arrivals. These methods are only capable to determine to the direction of the sound source, but lack the capability of distance estimation. Plus, the performance is easily influenced by surrounding noise. Because of the inevitably large ambient noise in urban driving, they are not suitable for AVs. [8] only uses the siren data collected by their AVs when the AVs are parked still in the hospital without introducing much wind noise. [9][10][11] apply learning-based models. However, they focus on locating sounds in a relatively quiet indoor environment and the sound sources are close to the microphone devices within 10m distance range.

3) **Sequence Models**: Sequence models take sequential data as input or output. Sequential data, such as financial data, text and audio, are the data where the order matters. While Convolutional Neural Network (CNN) is mainly developed for computer vision applications, its ability to extract

\* Corresponding author

All Authors are with Apollo Autonomous Driving USA, a Baidu company, 1195 Bordeaux Dr, Sunnyvale, CA 94089 [luoqi06@baidu.com](mailto:luoqi06@baidu.com)

the local features from noisy data has attracted researchers and professionals to its applications with sequential data [5][12][13]. On the other hand, the local connectivity also comes with the limitation in capturing the global interaction. Therefore, transformers that directly handle the global information with the self-attention mechanism are gaining popularity in sequence models [14][15]. The downside of such a structure is the high memory cost, which is quadratic of the length of the sequence  $L$ .

As a result, modifications to transformers are proposed to reduce the space complexity in [16][17][18][19]. LogSparse Transformer proposed by [16] only attends to the previous tokens whose indexes' differences are powers of 2 with a combination of convolutions to enhance its locality-agnostic. Such modification reduces the space complexity to  $O(L(\log L)^2)$ , while limiting its capacity in exploiting the long-term dependencies. Reformer proposed by [18] utilizes locality-sensitive hashing and reversible transformer to further reduce the space complexity to  $O(L\log L)$ . It also uses identical queries and keys to reduce the number of parameters required for training. This could improve the model generalization, but potentially impede the model's expressivity.

### C. Contributions

Our proposed method is based on the backbone in [5]. Our main contributions are the following:

- 1) **Joint Model Training:** Our model is designed to perform three tasks simultaneously, including siren existence, siren sound source direction, and siren sound source distance detection. As shown in Figure 2, these three tasks share a unified backbone which consists of two streams that process the raw waveform and hand-craft features separately. For each task, we attach a block of fully-connected layers to the backbone and output the result accordingly. The latency to produce all three results online is approximately 50ms.
- 2) **Low Cost Devices:** To achieve the goal of siren sound source localization, we only use two microphone devices shown in Figure 1. Instead of using more expensive customized microphones, we employ more economical general-purpose devices. The price of each microphone device is less than 100 dollars.
- 3) **Auto-labeling Method:** We collect the siren data in real driving environments where the AVs with microphones and emergency vehicles with sirens are all moving at normal speed. The large number of data needed for the deep learning makes it even more difficult to manually label the data in such scenario. Therefore, we develop an auto-labeling method by taking advantage of the centimeter-level accurate localization module installed on Apollo autonomous driving system [20].
- 4) **Long-Distance Real-World data:** While most of the papers tackling audio challenges use synthetic data, we collect the audio data in real urban driving environments. The signal-noise ratio is more stable and

the noise is more stationary in the synthetic data. Data collected by holding microphones still can also result in a "reality gap" to the real autonomous driving situation, as the wind noise caused by the moving speed is a significant factor for model performance. In our data set, the distance and the ambient noise during each audio sample can vary in a wide spectrum. The sound sources can be as far as 100m in distances, so the trained models can be generalized to actual autonomous driving testing and operation.

## II. FRAMEWORK

### A. Hardware Integration

Two microphone devices are installed at the rear top of the autonomous driving vehicle (AV) as shown in Figure 1 after a trade-off between the sound quality and wind noise caused by the high-speed movement of AVs. Each microphone has 4 channels located at the front, back, left and right. There are 8 channels in total. These microphones are connected to the onboard computing unit through USB interfaces.

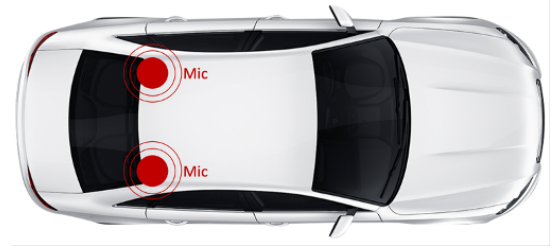


Fig. 1. Microphone devices integration.

### B. Onboard Architecture

In this section, we introduce the onboard software architecture of the siren detection module on Apollo autonomous driving platform as shown in Figure 2. It supports multiple-channel audio signal inputs, signal pre-processing and multi-functional model inference.

1) **Signal Input:** The signal input is collected from the 8 microphone channels as described in Section II-A. It takes 1.5 seconds of audio signals at 48k sample rate from every channel.

2) **Signal Pre-processing:** The pre-processing step applies a band-pass filter to filter out noise in the frequency range where the sound is less likely to be siren. This processed data are then dumped into the database for model training. The details will be described in Section II-C.

3) **Model Inference:** The model inference part takes the pre-processed signals from these 8 channels, and then applies a neural network model. The output consists of three parts: 1) if siren sound exists 2) siren sound source direction, and 3) siren sound source distance to the autonomous driving vehicle (AV). The details of the model structure can be found in Figure 2.

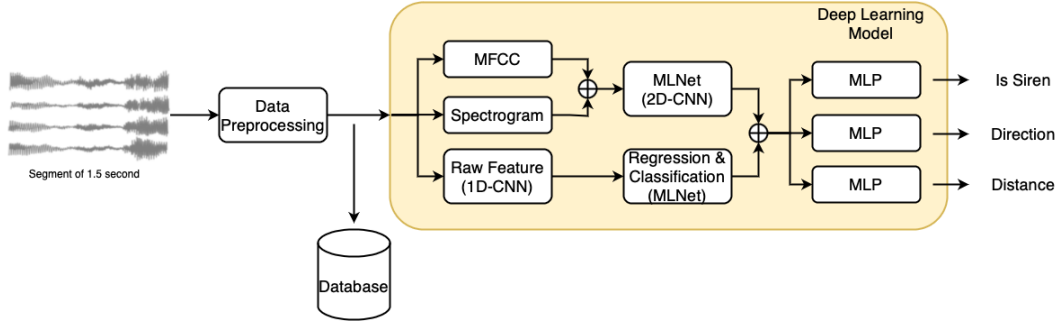


Fig. 2. Onboard architecture of the siren detection module on Apollo autonomous driving open-source platform.

### C. Offboard Architecture

The offboard architecture is described in Figure 3, which includes data auto labeling, feature extraction and model training.

1) *Data Auto Labeling*: One of the straightforward ways to collecting siren data is to follow an emergency vehicle with the siren on, and record the sound from different angles and distances. However, it is impossible (and possibly illegal) to measure the real-time positions of emergency vehicles and it is even harder for a person to recognize or manually label it offline. Deep learning models need hundreds of thousands of data points, therefore manual labeling is not feasible.

We design an automatic labeling algorithm to label is-siren binary classes, sound source direction and distance relative to the AV. A siren device is installed on another vehicle with Apollo open-source autonomous driving system that has a centimeter-level precise localization module. The locations of vehicles are recorded and later serve as ground truth labels. Vehicles are moving during our data collection, and their locations during the time frame for each data point are changing at different rates. We use the locations of the vehicles at the end of each data point to calculate the direction angle and the distance of the sound source relative to our AVs. The location message at the time of our interest can be either extract from the localization module or calculated by linear interpolation of locations from two nearest timestamps.

All siren data with distances above 100m are discarded because the siren sound is hardly perceivable in that range.

2) *Feature Extraction*: For the purpose of the data consistency between onboard and offboard, we use the data pre-processing part of the onboard workflow to dump the features needed in the model training. The extracted features consist of 8 channels of 1.5 seconds audio signals after the band-pass filter.

3) *Model Structure*: We deploy the SirenNet model proposed by Tran and Tsai [5] as our base model. There are two streams of the model to process two kinds of features. One stream processes raw waveform and the other processes the combination of Mel-frequency cepstral coefficients (MFCC) and log-mel spectrogram. These two streams work as the backbone of our model.

We improve the base model by doing the following

upgrades:

a) *Multi-Function MLP*: The representations of these features given by the two streams are concatenated to feed into multilayer perceptrons (MLP). While the SirenNet has only one final output that denotes the possibility of the audio input to be siren sound, our model generates more outputs for siren sound source localization: the direction angle and the distance to uniquely determine the position of the sound source. For the direction angle label, we do not use a single angle value on a scale of  $(0, 2\pi)$  directly, as the model itself cannot tell 0 and  $2\pi$  are the same. We use sine and cosine values of the direction angle for it to learn. Each MLP generates the output(s) for each task separately as shown in Figure 2.

b) *Attention Mechanism*: In addition to CNN deployed in the base model, we also experiment with transformer and reformer mentioned in [B.3] to process the features from the waveform. To overcome the memory constraint from the length  $(0.5s * 48k * 8)$  of each audio data point, we consider 1000 consecutive data points as one token in our sequence data. Since the self-attention mechanism ignores the order of the input data, we add the positional encoding as suggested by [14]. Instead of summing the positional encoding with the audio encoding, we concatenate them together by setting the length of positional encoding as 100. Note that we also experiment with the summing method but find that the concatenation method works consistently better with our data.

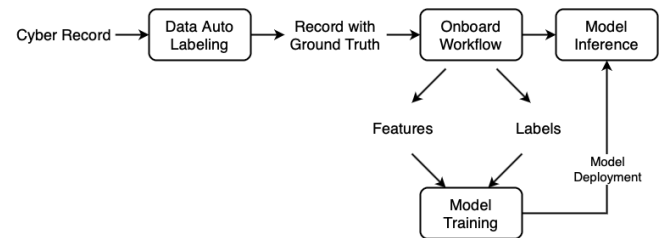


Fig. 3. Offboard architecture of the siren detection module on Apollo autonomous driving open-source platform.

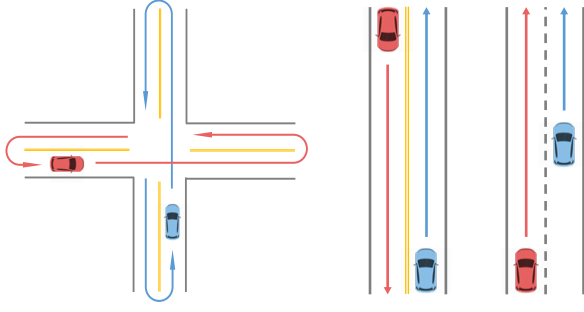


Fig. 4. Three data collection scenarios, where the blue one is the ego vehicle, the red one is the emergency vehicle. The left part is an intersection scenario where the two vehicles' trajectories are orthogonal to each other. The middle and right parts are the scenarios of regular road where the two vehicles' trajectories are parallel. The two vehicles' are in opposite directions in the middle part and in the same direction in the right part.

### III. EXPERIMENTS AND RESULTS

#### A. Data Collection

We set up three scenarios for data collection as shown by Figure 4. Under these settings, we are able to collect siren sound data from different directions at different distances with different ambient noise levels. Those collected records, including the siren sound recorded from the microphones and also the localization information of vehicles, are uploaded to our cloud platform. The auto labeling tool [II-C.1] is running in the cloud to generate machine learning-ready signal features and ground truth labels.

We collect a total of 20 hours of positive records with siren sound and 10 hours of negative records without siren sound in urban driving environments. Then we down-sample some positive data where the emergency vehicle with the siren on is strictly in front of and behind our AV to balance the direction distribution of our data. When data sets are split into training, validation, and testing set, we make sure that the data from the training set and the data from the valid/test set are not collected on the same day because the training data consists of consecutive sliding time windows of 1.5s interval of data. The model is validated and tested on data sets collected from different days from those in training. In this way, we decorrelate the data for better generalization. The samples from training, validation, and testing roughly follow a ratio of 8:1:1.

#### B. Experiment Setup

1) *Data Preparation*: The audio data is recorded at 48 kHz sampling rate by two 4-channel microphones. Every 1.5s length of audio is considered as one data point and we collect data points every 0.17s. We experiment with a time window interval of 1.5s, 1s, 0.5s and 0.25s as model input and find that 0.5s input achieves the best overall performance so we report all the results with 0.5s input. Band-pass filter is used to only keep the signal with frequency from 500Hz to 1800Hz so the model can focus on the part of the audio that is more likely to be a siren sound.

We consider the direction of the siren sound as 0 when it is right in front of our ego vehicle. When the siren sound

is at the left side of the ego vehicle, the direction angle is positive; when the siren sound is at the right side of the ego vehicle, the direction angle is negative. To ensure the sum of the square of the sine and cosine values of direction angle is one, we normalize these two outputs by dividing them with the square root of their sum of the square. The two normalized values are then used to calculate the direction angle output  $\hat{\theta}$ . When the absolute difference between the direction angle output  $\hat{\theta}$  and the ground truth angle  $\theta$  is larger than  $\pi$ , we use  $2\pi - \hat{\theta} + \theta$  in calculating the mean absolute error for the report purpose.

2) *Model Training*: We conduct our experiments on a server that runs the Ubuntu 18.04 operating system with the models implemented with 1.5.1 PyTorch framework. Adam is used as the optimizer and the initial learning rate is set as  $10^{-5}$ . The minimum learning rate is set as  $10^{-9}$ .

We use the binary cross entropy loss for is-siren binary classification. The sum of squared error is used for the sine and cosine of the direction angle and the distance detection. Since the loss calculated for our three tasks are at different magnitudes, the final loss of the whole model for the gradient computation is the weighted sum of these three tasks. The weights are subject to our experiments.

The accuracies are reported for the is-siren binary classification task. Mean absolute errors are calculated for both the direction angle and distance detection tasks. We train the model to detect the direction angle and distance of siren sound by masking the negative input data in calculating the loss for these two tasks. We choose the models with the best performance on the valid set and report their performance on the test set.

#### C. Performance Study

Table I shows that CNN models have better performance than the others in general. All models show comparable performances when it comes to the siren existence detection task. For the distance prediction task, the CNN model performs moderately worse than either the Transformer or the Reformer model. However, the CNN model performs significantly better than the Transformer and the Reformer model in siren direction angle prediction. The angle MAE of the CNN model is  $34.6^\circ$ , which is 25% better than the angle MAE,  $46.1^\circ$ , of the Transformer model and 29% better than the angle MAE,  $48.0^\circ$ , of the Reformer model.

The key factor in detecting the direction angle of the sound source is the difference time of arrivals among different channels, especially among different microphones for significant time differences. CNN is able to compute the relationship of the signals from different channels in each batch. Transformer and Reformer should be able to do that theoretically. However, they rely on positional encoding to understand the timestamp of the signals in each token input. This information requires more training of the model to capture aside from the signal information, thus requiring a much larger data set. We speculate that 264,506 positive training data points after data pre-processing result in the underfitting of Transformer-based models.



TABLE I  
PERFORMANCE COMPARISON BETWEEN CNN, TRANSFORMER AND REFORMER MODELS

| Model       | Weights         | Is_siren (Accuracy) | Angle Error (°) | Distance(m) |
|-------------|-----------------|---------------------|-----------------|-------------|
| CNN         | (1,0,0)         | 0.944               | NA              | NA          |
|             | (0,1,0)         | NA                  | 34.60°          | NA          |
|             | (0,0,1)         | NA                  | NA              | 14.226      |
|             | (10,10,0.008)   | 0.896               | 37.13°          | 15.221      |
|             | (10,5,0.008)    | 0.887               | 38.33°          | 14.761      |
|             | (10,5,0.005)    | 0.796               | 40.28°          | 14.432      |
| Transformer | (1, 0, 0)       | 0.940               | NA              | NA          |
|             | (0, 1, 0)       | NA                  | 46.12°          | NA          |
|             | (0, 0, 1)       | NA                  | NA              | 13.560      |
|             | (10, 10, 0.002) | 0.870               | 46.87°          | 15.335      |
|             | (10, 10, 0.008) | 0.832               | 50.25°          | 14.318      |
|             | (10, 5, 0.005)  | 0.839               | 51.97°          | 14.357      |
| Reformer    | (1, 0, 0)       | 0.938               | NA              | NA          |
|             | (0, 1, 0)       | NA                  | 47.96°          | NA          |
|             | (0, 0, 1)       | NA                  | NA              | 13.606      |
|             | (10, 10, 0.002) | 0.862               | 47.78°          | 14.812      |
|             | (10, 10, 0.008) | 0.834               | 51.45°          | 14.763      |
|             | (10, 5, 0.005)  | 0.856               | 50.82°          | 14.385      |

Table II shows the performance breakdown of CNN models according to the distance ranges in which the siren is from the AV. Since there is no distance label for a nonexistent siren, there is only the recall rate change of our model according to the distance between the siren sound source and our AV. When the sound source is less than 20m away from the AV, the recall rate is as high as 100%. When the sound source is 40m away from the ego vehicle, the recall rate can still remain over 99%. The recall rate is monotonically decreasing with the increase of the distance. The frequency of a siren sound is decreasing with increasing distance according to the Doppler effect, which can potentially lead to the overlapping of the siren sound frequency and noise frequency in urban driving. On the other hand, the siren sound intensity also decreases with the distance increases, thus resulting in a low signal-noise ratio.

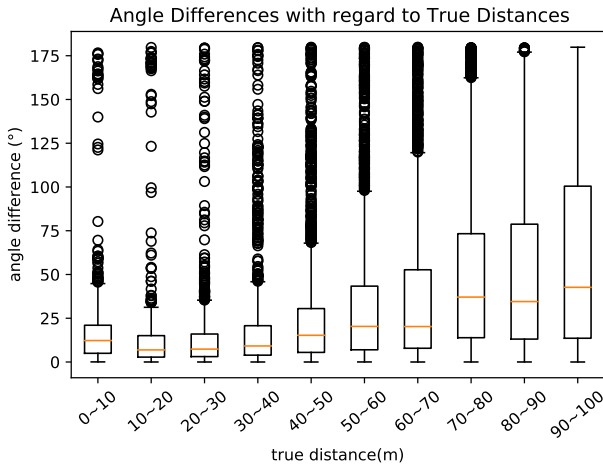


Fig. 5. CNN angle error analysis.

In contrast, the angle MAE is decreasing with the increase of the distance when the sound source is less than 30m away from the AV. This is because our data is collected when both

the sound source and the microphones are moving. Even though we only input 0.5s audio sample to the model, the relative angle of the siren to our ego vehicle can change dramatically during this time when the siren is close to the ego vehicle. As shown in Figure 5, the median of absolute angle error is only around 7° when the distance is from 10m to 30m. When the distance is larger than 30m, the angle MAE is increasing as the distance is increasing.

The distance detection MAE has an increasing trend with increasing distance overall, also shown by Figure 6. However, when the distance is between 20m to 70m, the distance MAE is relatively stable. When the distance is larger than 70m, the MAE of distance detection increases rapidly as the distance increases. This could be the result of the quality constraint of the microphones as we install general-purpose microphones instead of customized microphones that are more costly.

Table II also shows the performance of different models that are trained to produce the outputs for three different tasks concurrently. We choose the best model from CNN, Transformer and Reformer and fine-tune the last MLP layers associated with different tasks. The performance of these models after fine-tuning is reported in Table III. The model with the best performance is still a CNN model. The accuracy for this model is 89.4% while the angle MAE is 36.6° and the distance MAE is 14.2m.

#### IV. CONCLUSIONS

Our data collection system can collect the real-world sirens data in urban driving environments without tedious human labor. With cost-effective microphones, we are able to achieve excellent performance in three tasks when emergency vehicles are within a distance that is crucial for AV to react, 10m to 50, by training the model with the siren data within 100m distance range. This emergency audio detection system compensates for the limitation of vision-based object detection with state-of-the-art performance. We also test our system in actual autonomous driving scenarios and receive

TABLE II  
PERFORMANCE BREAKDOWN

| Metric \ Distance Range(m) | 0-10  | 10-20 | 20-30 | 30-40 | 40-50 | 50-60 | 60-70 | 70-80 | 80-90 | 90-100 |
|----------------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|--------|
| Recall (%)                 | 100   | 100   | 99.64 | 99.15 | 98.49 | 95.78 | 95.95 | 96.99 | 95.37 | 95.21  |
| Angle MAE (°)              | 23.54 | 15.39 | 13.56 | 17.55 | 26.30 | 32.98 | 35.59 | 50.36 | 51.83 | 59.14  |
| Distance MAE (m)           | 5.52  | 8.16  | 10.15 | 11.26 | 11.09 | 8.83  | 10.74 | 16.29 | 23.98 | 33.98  |

TABLE III  
PERFORMANCE COMPARISON BETWEEN CNN, TRANSFORMER AND REFORMER MODELS

| Model       | Weights         | Is_siren (Accuracy) | Angle Error ( $\pi$ ) | Distance(m) |
|-------------|-----------------|---------------------|-----------------------|-------------|
| CNN         | (10, 10, 0.008) | 0.894               | 0.639(36.61°)         | 14.184      |
| Transformer | (10, 10, 0.002) | 0.880               | 0.817(46.81°)         | 13.351      |
| Reformer    | (10, 10, 0.002) | 0.869               | 0.836(47.90°)         | 13.786      |

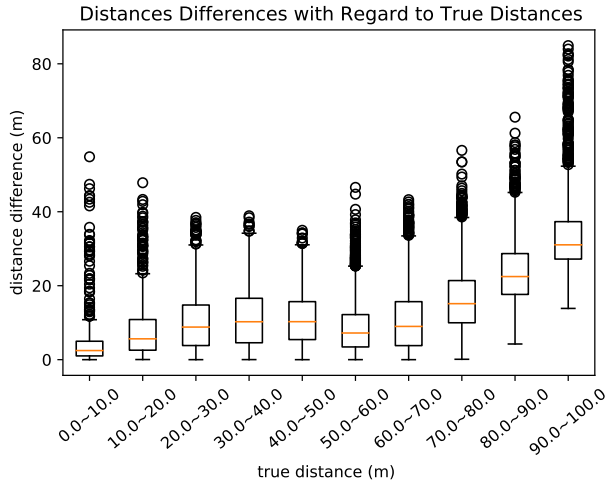


Fig. 6. CNN distance error analysis.

satisfactory feedback. The robustness of the model performance degrades when the siren distance is beyond a certain distance, i.e. 70m in our model. This can be potentially improved by utilizing a larger number of microphone arrays and more data collected.

## REFERENCES

- [1] T. GreyB, "Top 30 autonomous vehicle technology and car companies," Jun 2021. [Online]. Available: <https://www.greyb.com/autonomous-vehicle-companies/>
- [2] F. Beritelli, S. Casale, A. Russo, and S. Serrano, "An automatic emergency signal recognition system for the hearing impaired," in *2006 IEEE 12th Digital Signal Processing Workshop 4th IEEE Signal Processing Education Workshop*, 2006, pp. 179–182.
- [3] F. Meucci, L. Pierucci, E. Del Re, L. Lastrucci, and P. Desii, "A real-time siren detector to improve safety of guide in traffic environment," in *2008 16th European Signal Processing Conference*, 2008, pp. 1–5.
- [4] D. Carmel, A. Yeshurun, and Y. Moshe, "Detection of alarm sounds in noisy environments," in *2017 25th European Signal Processing Conference (EUSIPCO)*, 2017, pp. 1839–1843.
- [5] V.-T. Tran and W.-H. Tsai, "Acoustic-based emergency vehicle detection using convolutional neural networks," *IEEE Access*, vol. 8, pp. 75 702–75 713, 2020.
- [6] F. Grondin and F. Michaud, "Lightweight and optimized sound source localization and tracking methods for open and closed microphone array configurations," 2018.
- [7] R. Lee, M.-S. Kang, B.-H. Kim, K.-H. Park, S. Q. Lee, and H.-M. Park, "Sound source localization based on gcc-phat with diffuseness mask in noisy and reverberant environments," *IEEE Access*, vol. 8, pp. 7373–7382, 2020.
- [8] N. R. Shabtai and E. Tzirkel, "Detecting the direction of emergency vehicle sirens with microphones," in *EAA Spatial Audio Signal Processing Symposium*, 2019, pp. 137–142.
- [9] J. M. Vera-Diaz, D. Pizarro, and J. M. Guarasa, "Towards end-to-end acoustic localization using deep learning: from audio signal to source position coordinates," *CoRR*, vol. abs/1807.11094, 2018. [Online]. Available: <http://arxiv.org/abs/1807.11094>
- [10] N. Yalta, K. Nakadai, and T. Ogata, "Sound source localization using deep learning models," *Journal of Robotics and Mechatronics*, vol. 29, pp. 37–48, 02 2017.
- [11] C. Schymura, B. T. Bönninghoff, T. Ochiai, M. Delcroix, K. Kinoshita, T. Nakatani, S. Araki, and D. Kolossa, "PILOT: introducing transformers for probabilistic sound event localization," *CoRR*, vol. abs/2106.03903, 2021. [Online]. Available: <https://arxiv.org/abs/2106.03903>
- [12] X. Yan and J. Zhao, "Application of improved convolution neural network in financial forecasting," in *2019 IEEE 4th International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*. IEEE, 2019, pp. 321–326.
- [13] Y. Tay, M. Dehghani, J. Gupta, D. Bahri, V. Aribandi, Z. Qin, and D. Metzler, "Are pre-trained convolutions better than pre-trained transformers?" *arXiv preprint arXiv:2105.03322*, 2021.
- [14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.
- [15] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola, "Dive into deep learning," *arXiv preprint arXiv:2106.11342*, 2021.
- [16] S. Li, X. Jin, Y. Xuan, X. Zhou, W. Chen, Y.-X. Wang, and X. Yan, "Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting," *Advances in Neural Information Processing Systems*, vol. 32, pp. 5243–5253, 2019.
- [17] H. Wu, J. Xu, J. Wang, and M. Long, "Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting," *arXiv preprint arXiv:2106.13008*, 2021.
- [18] N. Kitaev, Ł. Kaiser, and A. Levskaya, "Reformer: The efficient transformer," *arXiv preprint arXiv:2001.04451*, 2020.
- [19] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *Proceedings of AAAI*, 2021.
- [20] Apollo-Platform, "https://github.com/apolloauto/apollo," Apr 2017.