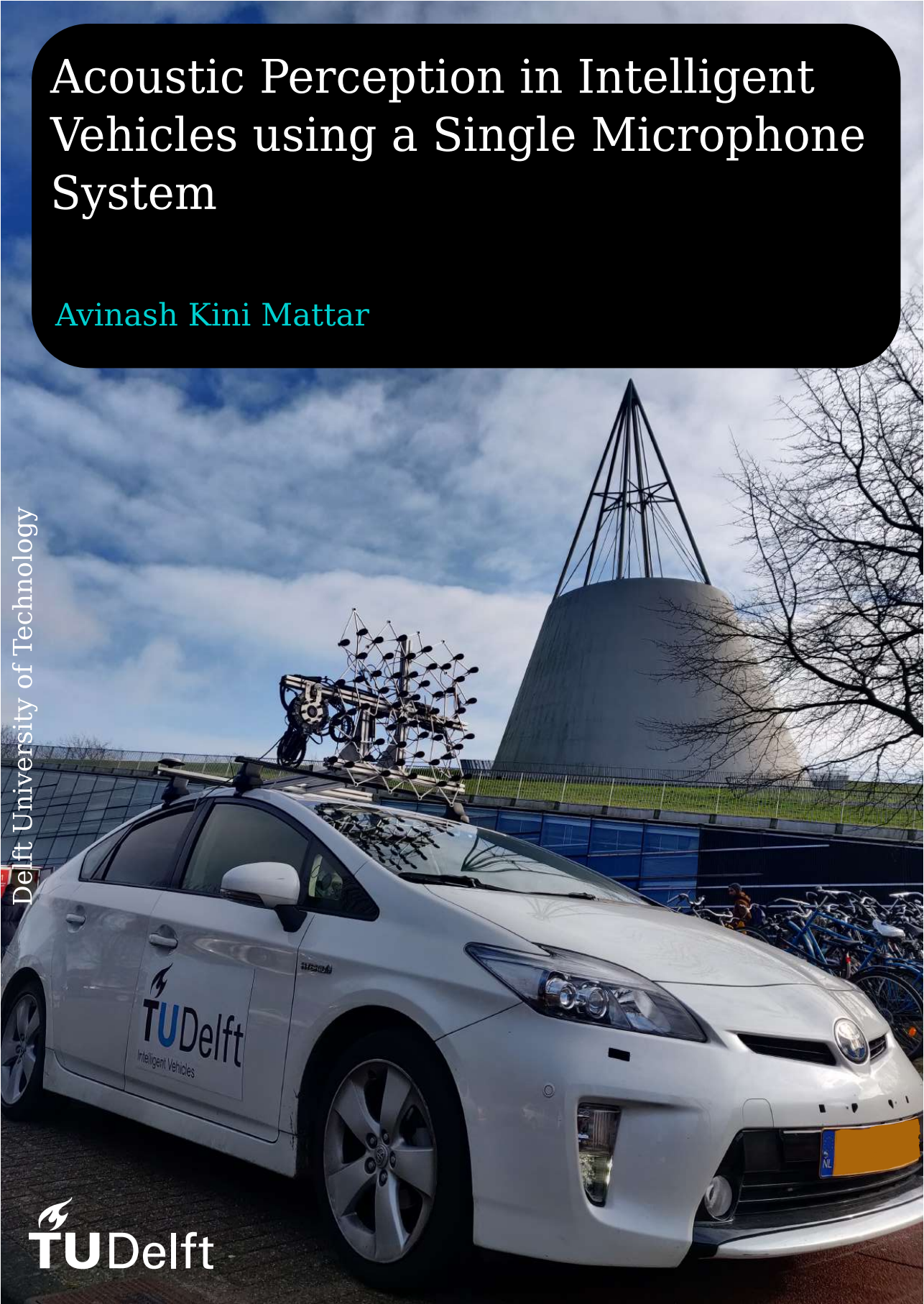
Acoustic Perception in Intelligent Vehicles using a Single Microphone System

Avinash Kini Mattar



Delft University of Technology

TUDelft

ACOUSTIC PERCEPTION IN INTELLIGENT VEHICLES USING A SINGLE MICROPHONE SYSTEM

by

AVINASH KINI MATTAR

For the degree of Master of Science in Mechanical Engineering
(Vehicle Engineering) at the Delft University of Technology

November 21, 2020

Faculty of Mechanical, Maritime and Materials Engineering (3ME) · Delft University of Technology

This thesis was done under the supervision of

Dr. J. F. P. Kooij,	CoR, Delft University of Technology
T. M. Hehn,	CoR, Delft University of Technology

The thesis committee consisted of

Prof. Dr. D. M. Gavrila,	Chair	CoR, Delft University of Technology
Dr. J. F. P. Kooij,	Supervisor	CoR, Delft University of Technology
T. M. Hehn,	Daily Supervisor	CoR, Delft University of Technology
Prof. Dr. G. C. H. E. de Croon,	External Member	LR, Delft University of Technology

Cognitive Robotics

Keywords: Acoustic Perception, Intelligent Vehicles, Sound Event Detection, Deep Learning, Domain Adaptation

An electronic version of this dissertation is available at
`http://repository.tudelft.nl/`.

ABSTRACT

Passive acoustic sensing utilizes the ability of sound to travel beyond the line-of-sight to understand the surroundings. This provides an advantage over the currently used sensors in Intelligent Vehicles that can sense obstacles within their line-of-sight only. Recently, a localization based approach has been implemented to take advantage of this sensing modality to predict approaching vehicles behind the blind corner in an urban scenario. While this approach shows a lot of promise, there is a difficulty in integrating the multi-microphone system. Additionally, the system would be unable to differentiate between the nature of two sound sources. This motivates the exploration of a classification based approach which uses audio data from only a single microphone to identify the sound sources present in them. This thesis investigates the possibility of having such a system on the Intelligent Vehicle to predict approaching vehicles from behind the blind corners.

A review of the literature revealed that techniques categorized under Sound Event Detection (SED) are suitable to implement a classification based approach. The prediction of the vehicle is treated as a binary classification problem and a Convolutional Recurrent Neural Network (CRNN) is used as the acoustic model to detect the presence of an approaching car in the audio sample represented by Log Mel Spectrogram features. Additionally, domain adaptation techniques were implemented to explore the possibility of improving the system performance with limited data collected while the ego-vehicle is driving. Experiments carried out indicate that when the ego-vehicle is static, the system performs well with the approaching vehicle predicted $1.4s$ before it is in line-of-sight and a balanced accuracy of 86.9% achieved for the classification task. However, the system achieved an accuracy of 68% on the samples recorded while the ego-vehicle was driving. Further experiments indicate that the acoustic model cannot generalize well to unseen situations in most cases and experiment with domain adaptation did not show any improvement in performance.

ACKNOWLEDGEMENTS

The past two years at TU Delft have exceeded my expectations by a good margin and I am grateful that I embarked on this journey. First, I am deeply indebted to my supervisor Dr. Julian Kooij for giving me an opportunity to work on the novel and challenging domain of Acoustic Perception for Intelligent Vehicles. It was exciting to be among the first few researchers to explore the feasibility of this sensing modality in a realistic scenario.

It would have been difficult to complete my thesis without the support and motivation from both of my supervisors Julian and Thomas Hehn. Their excellent guidance, recommendations and the crucial feedback during the various meetings we had, helped me structure my thesis well. It would be very remiss of me if I did not express my gratitude and appreciation to Thomas who ensured there were no disruptions in setting up of my remote workspace during the uncertain times of COVID-19 and for all the times he was available for impromptu discussions. I would also like to acknowledge the assistance of Yannick Schulz and Thomas in collecting and processing the acoustic dataset without which this work would not have been possible.

Next, I would like to thank my parents for their belief in me over the years and their support which allowed me to pursue my education with lesser worries. I would also like to thank my brother for the many technical discussions we had over the last two years which helped me gain a deeper understanding of various concepts. Lastly, I would like to thank my friends and housemates for the fun times we had and making my life in Delft more memorable.

Avinash Kini Mattar
Delft, November 2020

CONTENTS

1

INTRODUCTION

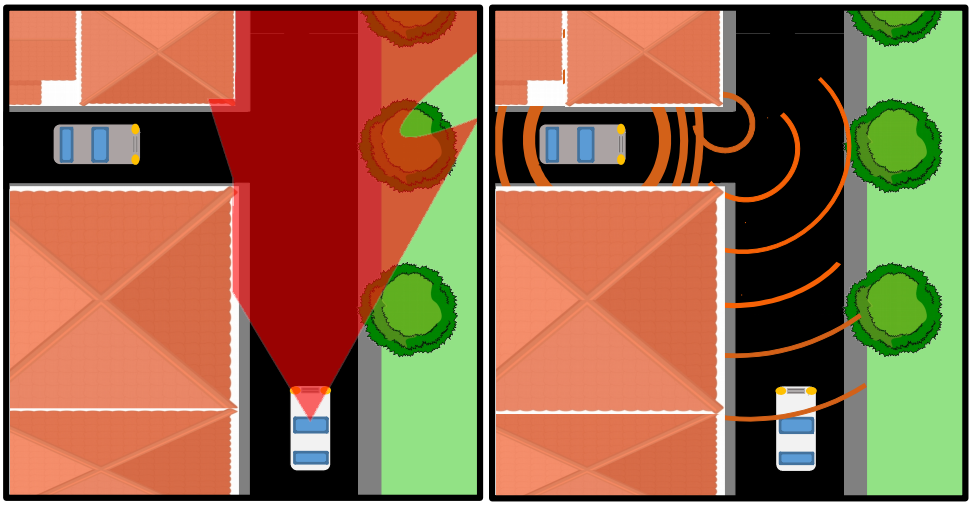
DRIVING is one among the complex tasks that many humans undertake in their daily life. This complexity is apparent as about 95% of the road accidents can be attributed to some level of human error [1]. Furthermore, the explosive growth in the number of vehicles on the road has led to increase in the likelihood of fatal accidents. A report by the World Health Organisation [2] highlights a grim reality of this scenario, wherein it is estimated that about 1.35 million people die annually in road accidents. This has motivated, both the academia and the industry, to focus developing technologies that would render the road vehicles less dependent on human drivers, and one of the terminologies used for such vehicles is Intelligent Vehicles (IVs).

Humans rely mainly on their vision to complete the complex task of driving. Taking this as a start point, lots of effort and money have been invested in the developments of line-of-sight sensor technology and the algorithms to process the data acquired from the same. With the driving performance of IVs now closer than ever to that of humans, it can be said that this approach has reaped large benefits. However, in order to truly capture human experience of driving, the IVs must employ sensors other than only those who emulate line-of-sight vision i.e. Camera, LiDAR, Radar etc. In addition to vision, humans also tend to use other sensor modalities while driving. For instance, while driving, humans can recognise an approaching emergency vehicle with loud sirens and react appropriately by recognizing the sirens through their distinct acoustic signature. As pedestrians and cyclists, humans are capable of detecting other wide array of traffic sounds, one instance would be the sound of an approaching vehicle from beyond their line-of-sight. However, these are difficult for the human to detect while driving because of the insulation modern vehicles offers against external sounds.

1.1. ACOUSTIC PERCEPTION FOR INTELLIGENT VEHICLES

Acoustic perception can be the key to unlocking super human levels of perception performance in IVs. As sound can travel beyond the line-of-sight of its source, using auditory cues can impart an IV the ability to predict impending obstacles well before they

1



(a) Line of sight sensing

(b) Acoustic sensing

Figure 1.1: The IV would be able to identify a vehicle approaching the blind intersection in case of (b) as opposed to that in case (a) where an emergency stop may be required putting the inhabitants of both the vehicles at risk. This thesis aims to evaluate the feasibility of a acoustic perception system that can augment safety of all participants in this scenario.

are within the sights of other sensors. For example, consider a traffic scenario where you have two vehicles approaching a blind intersection. Figure 1.1 provides a schematic representation of this scenario. Having acoustic sensing capabilities on the ego-vehicle would mean that the other vehicle approaching from behind the corner can be detected well before the intersection. The spatial layout presented in Figure 1.1 can be commonly found in urban areas. With the possibility of the presence of tall buildings and increased vehicular density, IVs navigating with only line-of-sight sensors through such areas are at a higher risk of encountering such dangerous scenarios.

To implement acoustic perception to address the above scenario, the ego-vehicle has to employ sensors that can capture sounds emanating from different sources in its surroundings and algorithms that can recognize the auditory cues of the other car approaching the intersection. These auditory cues available to the algorithm can vary based on the sensing approach that is adopted i.e. sound source localization or sound signal classification. In the localization based approach, a multiple microphone setup is used to acquire sound signal from the surroundings. The directional information for each of the sound sources can be extracted from the input data and those serve as the the cues from which the algorithms can learn to recognize relevant sources. Whereas in the classification based approach, the algorithm learns to detect the approaching car by recognizing its acoustic signature and audio signal from a single microphone setup is enough to achieve the required output.

During the development of this thesis, our acoustic research team implemented a localization based approach (see Appendix A) to detect a vehicle approaching the blind intersection similar to the one described in Figure 1.1. When the ego-vehicle was static, the system designed here was able to identify an approaching vehicle at an accuracy of 0.92 and achieve similar levels of accuracy $1s$ second ahead of a state-of-the-art visual detector thus increasing the reaction time available for the ego-vehicle. This indicates that by adopting this approach, one can expect the presence of clear auditory cues pointing towards an impending collision. Here, a multiple microphone setup is a must to implement this system and since microphones are relatively inexpensive, using more than one of these will not augment the hardware cost of the setup by a significant factor. However, the placement of these microphones in not trivial and there is no closed-form solution to obtain the optimal spatial configuration of such a setup [3]. And, since the microphones would have to be placed on the outer surface of the ego-vehicle, maintaining aerodynamic efficiency would add another dimension to this problem.

If a classification based acoustic perception is implemented, the complexity of microphone placement problem would be reduced as only one of them might be sufficient. In addition, localization based approaches cannot identify the sound source, which means that it is not possible to determine whether the sound is originating from a source that is of interest to the IV. This however, can be solved by having classification based perception working in tandem, which further strengthens the motivation to explore this approach for IVs.

1.2. COMPUTATIONAL ANALYSIS OF SOUND SCENES AND EVENTS

Sound event in an audio signal can be defined as a segment of the signal which can be identified by a textual label[4]. Following this definition, a typical sound event would mostly have a single source, be of short duration and well-defined. This could be the sound emanating from a car passing by, breaking of a glass etc. By contrast, *sound scene* refers to sound that arises from the mixture of the sounds from all the sources that are present in any situation. For example, the sound scene in the case of recordings made for IV application would be the combination of vehicle sounds, people interacting, their movements, weather related sounds etc.

The literature consists of many examples for which a classification based approach has been implemented for acoustic perception in different application scenarios, though only a few works address the IV setting. Similar methods have explored extensively and have been categorized under the term Computational analysis of sound scenes and events [5]. This is one among domains of technologies which enable machines to interpret sounds around them i.e. machine hearing. The goal here is to be able to detect these sound events, sound scenes or both from an audio signal.

Based on the information they extract from the signal, the techniques can be broadly categorized into Acoustic Scene Classification, Audio Tagging and **Sound Event Detection**. As the name suggests, Acoustic Scene Classification (ASC) (also known as Sound

1

Scene Classification) involves determining the sound scene of the given audio signal. Audio Tagging (AT) deals with identifying the sound events present in an audio recording, whereas **Sound Event Detection** (SED) deals with the classification of each sound event and additionally determining their temporal onset and offset. Figure 1.2 depicts the schematic of methods belonging to the above categories.

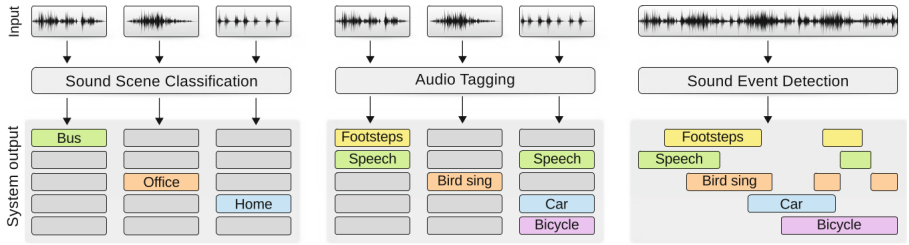


Figure 1.2: Three categories of computational analysis of sound scenes and events. The methods belonging to ASC and AT categories generally deal with audio recording long temporal length. Each block of input corresponds to different audio clips/recordings. However, those in SED operate on short segments of audio signal within a recording or continuously streaming input. Image from [5]

1.3. SOUND EVENT DETECTION

The environment in which IVs have to navigate, are comprised of a wide variety of sound events and their mixture in different contexts leads to variety of sound scenes (e.g, urban roads v/s rural roads). Understanding sound scenery as perceived by IV allows it to characterise the environment around it and thereby changing its nature of interaction during navigation. Part of this understanding is that the IV has to be able to differentiate between individual sound events for safer navigation. For instance, some of the events such as emergency vehicle sirens, tire noise from an fast approaching vehicle, auditory signals for visually-impaired pedestrians near the crossings etc. are very useful for the IV, while sound events such as an airplane passing above, crackling of thunder, among others are not very useful for the IV to navigate among the roads. Figure 1.3 depicts such a typical scenario wherein the IV will have to focus on identifying various sound events.

It would be prudent to focus on the task of identifying sound events, as characterising the environment can be achieved by other existing sensors commonly used in the IVs. For example, images captured by the camera can be potentially be used for this purpose [7]. Prioritizing identification of sound events then would help in avoiding scenarios as shown in Figure 1.1 and to identify them, techniques from two categories can be implemented: SED and AT. It can be observed from Figure 1.2 that the techniques in AT operate on audio recordings while those belonging to SED operates continuously streaming audio signal. Since the environment around IV is continuously changing, it makes techniques in SED better suitable to identify sound events as there would be a considerable delay in response by the IV if audio recordings with long temporal length are processed.

Figure 1.3: In the current infrastructure, IVs need to distinguish various sound events, for e.g engine/tyre noises from different vehicles, car honks, people talking etc. Image from [6]

SED can typically be broken into two sub tasks: *audio representation* and *classification*. In the audio representation stage, the incoming signal is divided into short segments of time length *t*. These segments are then processed to generate acoustic features and obtain a feature vector x_t. The classification stage consists of an *acoustic model* which can map the vector x_t to a set of pre-defined labels y_t. By aggregating the predictions over consecutive time frames, the onset and offset temporal locations of each sound event can then be determined. Figure 1.4 depicts the overview of a typical SED task.

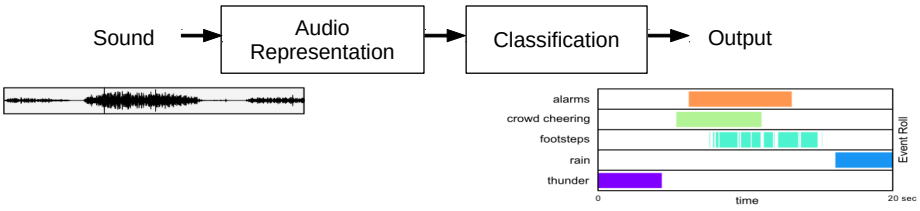


Figure 1.4: Stages in Sound Event Detection. Image adapted from [8]

1

1.4. DOMAIN ADAPTATION

In the traditional machine learning setting, the distribution of the data available for training the model and the target data would be same. Furthermore, the learning tasks for both model datasets will always be the same. However, there would be scenarios in real-world applications especially, wherein the data will be distributed differently. This would mean that models learnt through traditional machine learning techniques will have to be re-trained every time data from new domain is collected. Domain Adaptation is a sub-discipline of machine learning, which deals with models that would be trained on a source domain, but are used in the context of a target domain that is different but related to the source domain. The differences would be in the marginal probability distribution of the data, however, the feature spaces between the source and target domain would be the same. In the context of SED for IVs, the feature space can depend on the technique used to represent the incoming audio signal. The difference in probability distributions can arise from either using synthetic audio samples as source and real-life recordings as target domain samples, or source domain could be audio samples recorded while the IV is static and the target domain could be those recorded while the IV is driving. Techniques from domain adaptation have been developed and extensively explored in the visual applications as summarised by the survey here [9]. However, domain adaptation has been sparsely explored in SED in particular and the research in this direction is slowly starting to take shape.

Techniques from domain adaptation can be influential for acoustic perception in IVs. In the scenario described above (see Figure 1.1), both the IV and other vehicle on the perpendicular road are moving towards the intersection at roughly the same time. Collection of data under these circumstances should be done with proper care and hence it can be very tedious. On the other hand, if the ego-vehicle is not moving in this scenario, there is no risk of collision with the approaching car. Hence, the data collection does not require high levels of supervision and the approaching car can be oblivious to the process of data collection. An additional benefit is that if an acoustic perception model can be trained using only static ego-vehicle data, there would be no need of an IV to be present at the data collection and instead can be performed only by a microphone setup with a storage and processing unit. This could mean that data collection process can take place in parallel with a smaller setup and could drive the costs down in a large scale setting.

1.5. RESEARCH QUESTIONS

The goal of this thesis is to explore if classification based acoustic perception methods can assist the IV in predicting an approaching vehicle at the intersection as shown in Figure 1.1. The main research question that echoes this goal would be:

1. How well can a single microphone setup on an IV predict an approaching vehicle at a blind intersection?

When the acoustic model is deployed on IV, it can be said that the performance influenced by how well the training data is representative of the locations that an IV might

visit. However, it is impossible to collect training data from all of the locations that an IV might visit due to the sheer number of possibilities. Hence, it is important to investigate the performance of the acoustic model under these conditions. Another factor that can be investigated along with same lines is to see if the acoustic model trained only data collected from the static ego-vehicle can generalize on the data collected while it is driving. If the generalization is possible in this context, then it would ease the data collection and the implications of the same is described in Section 1.4. Both of the above mentioned investigative directions can be summarized in the research question below:

2. How well does the acoustic model generalize across different locations or different ego-motion modes?

As described in Section 1.4, domain adaptation techniques could be employed to potentially reduce the data collection effort. In the interest of exploring this line of research, the following question can be asked:

3. Is it possible to use techniques from domain adaptation by using static data to improve the performance on driving data which is limited in number?

1.6. THESIS OUTLINE

This thesis will start with Chapter 2 providing an overview of the techniques implemented in the SED literature, along with those that experiment with domain adaptation in tandem. These techniques covered here are implemented for non-IV based applications as SED has not yet been implemented for IV before this work. Further, datasets that might be relevant to the IV domain are also discussed here. Chapter 3 then details the techniques that would be implemented to answer the research questions set above. The results of the various experiments are reported and discussed in Chapter 4. Chapter 5 discusses the conclusions drawn and the recommendations for future research directions to take with respect to this work and also towards implementing an acoustic perception in IVs. The article elaborating the implementation of the localization based acoustic perception has been submitted to the IEEE Robotics and Automation Letters (RA-L) Journal and the IEEE International Conference on Robotics and Automation (ICRA), and is currently under review. Since the research conducted in this thesis has contributed to the implementation of the localization based approach, the aforementioned article has been included in Appendix A.

2

Related Work

Implementing a single microphone setup as the acoustic perception system for the IV would mean that sound classification based approaches have to be considered to understand the perceived audio signal. Amongst these sound classification techniques in the literature, methods categorized under Sound Event Detection (SED) are suitable for their application in IVs given how they process the input audio signal. Sections 2.1 and 2.2 discusses in-depth the various techniques used in the literature for the *audio representation* and *classification* stages of SED pipeline. Section 2.3 elaborates on the domain adaptation techniques implemented in SED and discusses the motivation behind the choice of a technique to be implemented in this work. Section 2.4 gives an overview of the datasets that are closely related to the application scenario considered here. In Section 2.5, the commonly used performance metrics used to evaluate different SED systems are discussed. Finally, Section 2.6 lists the contribution of this work.

2.1. Audio Representation

The sound as perceived by the microphones can be processed and be represented in different forms i.e. time domain, frequency domain and the time-frequency domain. The choice of audio representation is very crucial as it directly affects the performance of the acoustic model. In the literature, the time-frequency domain is the most common audio signal representation used for SED. The number of processing steps involved to obtain the vector x_t however, depends on this choice. For example, if raw waveform of the audio (time domain) is used, then the only processing step is the division of the same into segments of length t, while for time-frequency domain/frequency domain, in addition to the segmenting, Fourier transform has to be performed to extract the frequency information. Figure 2.1 depicts some of the different audio representations used in the literature.

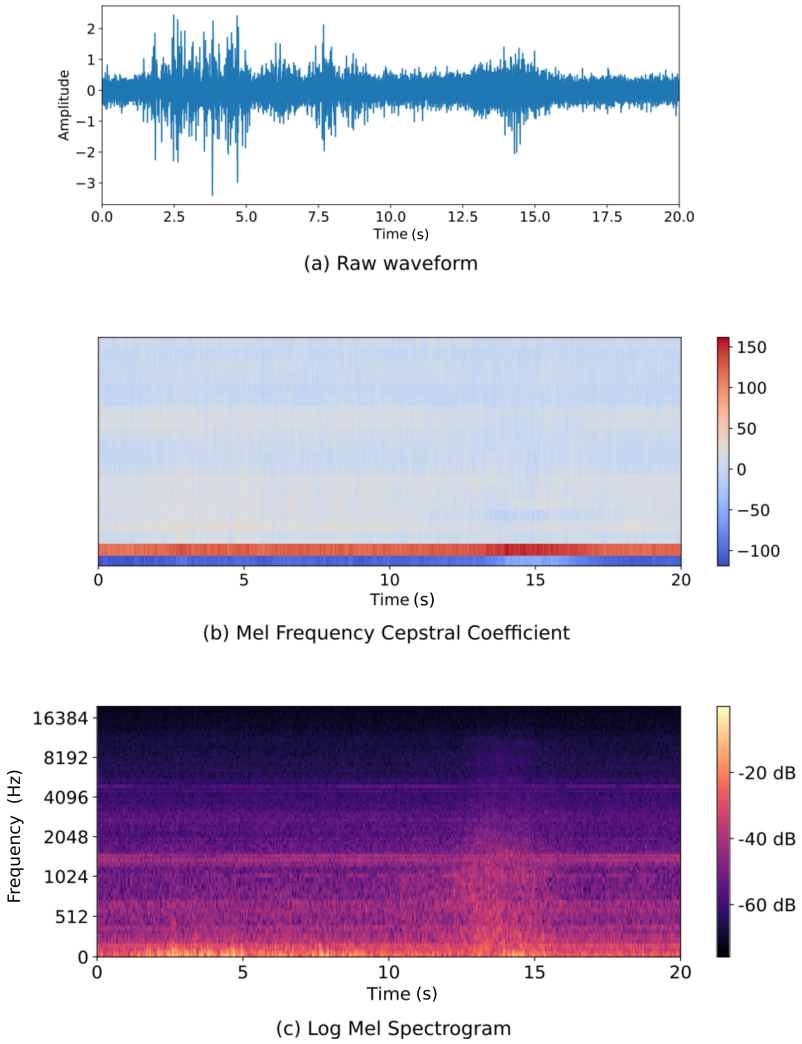


Figure 2.1: Different Audio Representations used in the literature. The audio is recorded from an microphone array mounted on top of an IV. Here the IV is in static state (with its engine on) has another car pass in front of it from its right to left. Higher energies in the Log Mel Spectrogram around 15s time mark is when the car is directly in front of the IV.

2.1.1. TIME DOMAIN

Time domain is the fundamental representation of the audio signal. These are raw waveform (see Figure 2.1 (a)) which represent the electrical voltage level as obtained from the microphones directly. The only processing step for time domain features is that they are divided into overlapping segments of time length t. This representation is not the most popular choice in the literature as it is not as robust to noise compared to others [10]. Moreover, raw waveform do not provide clear semantic differentiation between

different sound events as well other representation. In [11], the authors conclude that the frequency based representation offers more discriminative features for their acoustic model to learn and outperforms model based on raw waveform.However, some works take a different approach of utilising raw waveform for classification. For instance, authors in [12],[13],[10],[14] and [11] first learn a new representation using deep learning techniques such as Convolutional Neural Network (CNN) and then classify based on the learned representation. Here, [10] and [14] address the application of SED, whereas [12],[13] and [11] address an application of ASC. The main motivation behind implementing this approach for SED is that it makes no assumption about the optimal audio representation and is instead learned. All of these works have shown that they can closely match the performance of traditional time-frequency domain representation, however they require a lot of data just to learn the representation.

2.1.2. TIME-FREQUENCY DOMAIN

The most common form of representation used in the recent SED literature is in the time-frequency domain (see Figure 2.1 (c)). This is based on the reasoning that distribution of energy in frequency provides a lot of information that amplitude variations in raw waveform cannot provide [15]. Moreover, the time-frequency domain features are multi-dimensional and can be represented as an image. This implies that advancements in machine learning techniques for image classification tasks can also be adapted to SED.

The most basic time-frequency domain representation of an audio signal is called the *spectrogram*. Spectrogram has a real part (Magnitude spectrogram) and an imaginary part (Phase spectrogram), as it is the result of the Fourier transform which is a complex valued function. Some of the works use only the magnitude spectrogram [16],[17] while others use both magnitude and the phase spectrogram [18],[19],[20] for SED. However, there is no work that quantifies the effect of phase spectrogram for any of the acoustic classification tasks. There are other different variations of spectrogram features that employed for SED, ASC or AT tasks. Among them, the most popular choice is the Log mel spectrogram, where the frequency axis is transformed into Mel scale. The motivation behind this transformation is that in Mel scale, the pitches on the scale are adjusted such that the human listeners perceive them to be equidistant from one another [21]. Another variation spectrogram which is used for machine hearing tasks is the Gammatone spectrogram, also known as Gammatonegram [22]. Gammatonegrams are also inspired by the human auditory systems' ability to well differentiate between sounds of lower frequency than compared to those at higher frequencies. Furthermore, these features have been successfully applied in monophonic SED [23],[24]. To the best of my knowledge, there is no direct comparison between Gammatonegrams and Log mel Spectrograms for SED, ASC or AT. A possible reason for this might be that these features may be very similar in nature as they are both modelled after human auditory system and hence researchers go with the previously proved features.

2.1.3. FREQUENCY DOMAIN

Unlike time-frequency domain features, the audio representation falling under this categories do not encode temporal evolution of frequency information. Among frequency domain features, Mel Frequency Cepstral Coefficients (MFCC) (see Figure 2.1 (b)) is the most preferred choice for machine hearing related task, especially in the domain of Automatic Speech Recognition [25]. For recent works in polyphonic SED, it has been shown in [26] and [27] that MFCCs do not outperform Log mel spectrogram features. The authors in [27] mention that because MFCCs are derived through application of Discrete Cosine Transform on Log mel spectrogram, the loss of information is the reason for decrease in accuracy.

Humans can distinguish overlapping sounds with different pitches [28]. In [29], this attribute of sound is leveraged in the task of polyphonic SED, where three dominant pitch values are extracted per frame t. The resulting feature however is just supplements other features rather than used as a stand-alone feature. Moreover, [29] had only three target sound events to be detected. Increasing number of target sounds can restrict the distinctiveness this feature provides. Authors in [30] extend on this idea by introducing features computed by pitch values using Auto-correlation. As in [29], this is not the only input to their acoustic model.

2.1.4. OTHER REPRESENTATIONS

Transfer learning approach is commonly implemented in image classification task when the dataset available for the target task is small. A similar approach is implemented by authors of [31] to perform SED. First, a large model is trained to perform SED on a larger dataset - Youtube-100M [31] using Log mel spectrogram features. Then on a smaller dataset, *Audio Set* [32], the authors [31] train two fully-connected models, first of which takes in Log mel spectrogram as input and the other takes in the embedding from the larger model trained on Youtube-100M. It is shown that the latter model trained on the embeddings outperforms the former. Authors in [33] conducted a similar experiment, however they are not entirely successful in outperforming their baseline. However, they [33] conclude that the size of the two datasets should differ by a factor of 50, otherwise it will be counterproductive.

Most of the research addressing SED use audio signals only from a single microphone. However, some works explored the possibility of taking advantage of audio signals from multiple microphones. In [29], authors use a binaural microphone setup for recording sound and compute Time difference of Arrival (TDOA) for sounds in five mel-bands. They estimate TDOA using the generalized cross-correlation with phase-base weighting (GCC-PHAT)[34] and append these features to the Log mel spectrogram features. The details of computation can be found in [29]. Authors in [30] take a step back and append the GCC-PHAT values instead of computing TDOA and let the acoustic model figure out how to learn from this data. They found out that the performance was equivalent in both cases and were better than just using Log mel spectrogram. Additionally, authors in [35] found out by just using log-mel spectrogram and GCC-PHAT instead of magnitude and phase spectrogram features, the baseline is outperformed. The chal-

lenges involved in using spatial features for SED is to have to a database of sound events which is recorded by multiple microphones and the availability of the same microphone arrangement for experimentation.

2.2. CLASSIFICATION

With the growing ability to record and store information, data-driven techniques have found themselves to be in favour of traditional rule-based techniques across all domains of technology. For instance, authors in [36], use a rule-base scheme to classify between music, environment sounds and silence, which have highly distinct acoustic characteristics, however they use a data-driven technique to classify speech and the aforementioned non-speech classes as a pre-classification step, citing the reason that acoustic features of speech are sometimes similar to that of music or environment sounds. Extending this approach to include other classes while using a rule-based method to achieve high performance would be difficult. Moreover, the authors [36] tackled monophonic SED which is not a realistic scenario for an IV. Researchers from then on identified the proficiency of data-driven techniques for acoustic modelling.

2.2.1. MACHINE LEARNING FOR SOUND EVENT DETECTION

An important subset of these data-driven techniques include machine learning techniques which provides the ability to perform a certain task without being explicitly programmed. In the machine learning setting for SED, the task is to learn an acoustic model which can estimate probabilities $p(y_{c,t}|x_t)$ for each sound event c, given a input feature x_t extracted from an audio segment of length t. While deploying the learned model for the application it was trained, the estimated probabilities are binarized by thresholding ($y_t \in [0,1]$) and aggregated over time to determine onset and offset of each sound event predicted.

Early adopters of machine learning techniques for SED used Hidden Markov Models (HMM) [37][38] and Gaussian Mixture Models (GMM) [39]. This is because they these methods were successful for other machine hearing tasks such as speech recognition [40]. In the context of IVs, authors in [41] have classified 5 distinct but abnormal sound events that one might encounter on the roads. They implement a GMM based classifier which achieves 87.5% accuracy on their dataset generated from web-scraped data, but they only perform monophonic SED. Further, authors in [42] use a part-based model to detect sirens of emergency vehicles in traffic noise. They achieve high performance in scenarios with high SNR, however as the SNR reduced to more realistic scenarios the performance of their acoustic models prediction quickly reduces to chance level. To address polyphonic SED using HMMs, authors in [43] proposed a approach where in they would use consecutive passes of the Viterbi algorithm over the audio signal. From second iteration onwards, the model is prevented from entering into HMM states that have detected in the previous iteration. Thus, the framework would not allow real-time detection and also assumes that the maximum polyphony (overlapping sources) in the dataset is known. Support Vector Machines (SVM) have also found success in SED. Authors in [44], use SVM along with their proposed clustering scheme to

slightly outperform a GMM based classifier. Further, authors in [45] show that SVMs outperform HMMs in terms of classification of the audio segments. However, more recent works [46][47][38]have again used HMMs for SED as they enable to learn a language model i.e. temporal patterns, in sound events [33].

Another popular statistical machine learning technique used in SED is the non-negative matrix factorization (NMF)[48], as using this technique it is possible to explicitly model for polyphonic SED. In NMF, spectrogram of the incoming audio X is decomposed as the product of two matrices A and B by minimising an error function such as Kullback-Leiback divergence between X and the product AB. If shape of the spectrogram X is $f \times t$, where f is the number of frequency bins and t is the frame length of the audio segment, then the factors A is of the shape $f \times n$ and B is $n \times t$. Here, n is number of tracks the original audio track is separated into. In [49][4], the authors separate the audio track into 4 components and then use a HMM-based acoustic model to perform monophonic SED on all the four tracks and later aggregate results to get a polyphonic SED output. However, using the knowledge of maximum polyphony restricts the scalability of the system. Authors in [50] take a different approach with NMF. They consider the matrix A as a dictionary for the sound events, whereas B is considered their excitation in time. Additionally, this excitation matrix is assumed to be the same for both spectrogram and the annotation matrix. During training, an audio dictionary and annotation dictionary is learnt using the NMF technique. While testing, the excitation matrix is estimated using the audio dictionary of the training and annotations are estimated by multiplying this excitation matrix with the annotation dictionary. The advantage of this method as cited by the authors [50] is that its possibility of learning from a small amount of data.

2.2.2. DEEP LEARNING FOR SOUND EVENT DETECTION

Deep learning techniques have now become the mainstream machine learning techniques employed to tackle SED. This has been possible because of the developments in hardware as well as advancements in the research into such techniques for different domains. The idea that deep learning techniques can learn from raw or a low-level representation of the data was attractive, as for other machine learning techniques (as explained above), feature engineering was an important step to achieve high performance. For instance, deep learning techniques for SED have been the most preferred choice of participants in competitions like Detection and Classification of Acoustic Scenes and Events (DCASE), which promote research into various machine hearing tasks including SED. Moreover, to the best of my knowledge, most of the recent works in literature of SED have employed deep learning techniques to solve their problem.

Among the vast array of deep learning techniques, the most basic one is simply the interconnections of nodes in different layers stacked together and are commonly referred to as Deep Neural Network (DNN). The stacking and the non-linear activation units in the nodes enables them to learn complex mapping from input to the output in complicated tasks. As SED is one among such tasks, it is reasonable to expect deep learning techniques to solve it. Authors in [51] were one of the first works to address SED using DNNs. They show a 2-layer network with 70 units each outperforms a conventional

GMM-HMM based classifier and further with unsupervised pre-training step further pushes the performance of the DNN. However, they [51] evaluate the accuracy of their model over the entire audio recording by aggregating frame-wise predictions.Moreover, the acoustic model trained here could predict only one label for each frame input. This is later rectified by the authors in [27] by formulating a multi-label learning task without limiting the number of overlapping events and was the first work which addressed polyphonic SED using DNNs. The authors here [27] use a 2-layer network with 800 units each and later process the predictions in a 10-frame window using a median filter to smoothen the output. They achieve consistent accuracy for different levels of polyphony upto 5 and set the state-of-the-art for polyphonic SED. As employed in baseline method [4], authors [27] experimentally conclude that DNNs do not require source separation or any similar indication as to how many overlapping sound sources to expect.

As discussed earlier in Section 2.1, an audio signal can be represented as an image if time-frequency domain features are extracted. Taking inspiration from application of deep learning techniques in image classification tasks, Convolutional Neural Networks (CNN) were implemented for SED and time-frequency domain features were used as the input for the same in most cases. Moreover, using CNN with time-frequency features allows the acoustic model learn the correlations in that domain which a DNN with similar input cannot. For instance, authors in [52] show that by considering a large input time window, CNN with time-frequency (spectrogram) features as input can outperform DNN with same input by a significant margin. It should be noted here that the spectrogram features are smoothed in the frequency axis using a two-element window, downsampled and removed of noise by subtracting the minimum element. Hence, it is not a entirely raw/low-level representation of the signal. While their [52] method has its drawbacks that it requires a large input window and predicts only one label per frame, it is a starting point to prove the potential of CNNs in the domain of SED. Further,the authors in [52] selected a prominent feature on the temporal dimension by using a 1-max pooling scheme on these downsampled spectrogram features, like in [16], to further enhance the performance of the acoustic models using CNNs. Additionally, they show that their method is consistency for signals with varying SNR and that it is better than other models based on DNN, HMM-GMM and so on. As with [52], their method [16] also predicts one label per frame which makes it suitable only for monophonic SED. Authors in [53] address polyphonic SED using CNNs. Here, the spectrogram features are not processed like in [52]. The method here [53] allows multiple labels for each frame, however to compare their approach to previously published methods, they aggregate the labels for all frame to get one label for each sound clip. However, in their work they compare it to a DNN based method [54] but show that their model outperforms the same.

The temporal context that a CNN or DNN based method takes into account depends on the input frame length. To more automatically consider temporal evolution and create a language model of sound events, deep Recurrent Neural Networks (RNN) have been used in SED. This property of deep RNNs have helped them find success in related fields such as speech recognition [55]. Additionally, by having unlimited context, it is also possible to consider under what background context does the sound event usually occur.

Authors in [56] and [57] were among the first ones to employ deep RNN for tasks such as ASC and SED respectively. In [57], they normalize the raw waveform audio to account for different recording conditions and then extract log mel spectrogram. Further shift the mean of each frequency band to zero and impose unit variance. To map these features to output, they use a bidirectional long short-term memory (BLSTM)[58] and a layer of sigmoid activation functions. The RNN based acoustic model here [57] outperforms the DNN model used in [27]. Interestingly, the parameters used in the BLSTM model (850K) had only half the number of parameters as DNN based model (1.6M) used in [27]. Authors in [59] extend on this idea [57] to slightly enhance the event-wise predictions over time by having a hybrid BLSTM-HMM model on the grounds that RNN outputs can be further smoothened. This approach does improve event-based metrics as intended but segment-based metrics are equivalent to that of baseline RNN. However, the authors further apply a Sound Activity Detection (SAD) mask, which is a BLSTM based network that predicts whether sound event of any class is active in a given time frame. This reduces errors wherein any background noise is estimated as a sound event.

The advantage with CNNs was that it could learn local spectro-temporal dependencies whereas RNNs can model long-term temporal evolution and contextual information about the sound events. Current research in SED is mostly based on Convolutional and Recurrent Neural Networks (CRNN) which combines the individual advantages of both CNNs and RNNs. Authors in [8] first proposed such a network and thoroughly establish that their acoustic model as the state-of-the-art. They conduct various over 4 datasets for acoustic models based on all the statistical machine learning techniques and the deep learning techniques discussed above. Crucially, they highlight the advantage of CRNN over CNN or RNN by showing that, in a polyphonic scenario, CRNN can detect sound events with both short and long temporal length, where as CNN is more sensitive to those with shorter length and RNN to that with longer length. The success of CRNN method is apparent with most recent works [60][61][19] and others adopting CRNN as their base architecture and experimenting with different possible scenarios in SED.

More recently, a few works have experimented with different alternatives to overcome disadvantages of CNNs, RNNs and CRNNs. For instance, authors in [17] have shown CapsNet [62] which is more robust than CNNs towards affine transformations outperforms them in polyphonic SED. However, the authors [17] do suggest adding recurrent units and exploring techniques to cope with lack of generalization they observed with their approach. The CRNN approach has been further improved by authors in [63] by replacing CNN layers with dilated CNN [64]. The motivation behind this is to increase the receptive field of CNN layers without increasing the number of parameters, which generally causes overfitting. With dilated convolutions, longer temporal context can be modelled compared to that with conventional convolutions. The improvement was found to be between 2-6% in the metrics across three datasets. Authors in [65] take this idea one step ahead and replace RNNs with dilated convolutions as they model long temporal context and further replace CNNs with depthwise separated convolutions [66] to reduce the number of parameters of the acoustic model. Further, the authors [65] state that by replacing the RNNs one can parallelize the model. This could help in speeding up the training process by taking full advantage of the GPUs. Moreover, this also drastically

reduces the number of parameters (from 3.68M to 0.58M [65]) while slightly augmenting the performance (≈ 4% improvements in the metrics [65]) However, while this method is better than CRNN architecture it should be noted that it has been trained and tested only on a single dataset which is entirely synthetic i.e. contains only isolated sound events picked up from various recordings with no background context. Moreover, as this is a recently published work, it should be further investigated with different datasets involving real-life polyphonic scenarios.

Table 2.1: Summary of the popular methods used for polyphonic SED over the years. It should be noted that there are other works which base their acoustic model on the methods mentioned in the table and this is just to give an overview of how they have evolved over the years

Title	Year Published	Acoustic Model description	Real-time
Supervised model training for overlapping sound events based on unsupervised source separation [4]	2013	NMF to separate sound sources and HMM to classify sound events	Yes
Polyphonic Sound Event Detection Using Multi Label Deep Neural Networks [27]	2015	Fully connected neural network with 2 hidden layers	Yes (without post-processing)
Exploiting spectro-temporal locality in deep learning based acoustic event detection [53]	2015	CNN layers with pooling and fully connected layers	Yes
Recurrent Neural Networks For Polyphonic Sound Event Detection In Real Life Recordings [57]	2016	Bidirectional LSTM and sigmoid activation output layer	No
Convolutional Recurrent Neural Networks for Polyphonic Sound Event Detection [8]	2017	CNN layers with max pooling connected to Gated Recurrent Units and Fully connected layers at the output	Yes
Sound Event Detection via Dilated Convolutional Recurrent Neural Networks [63]	2020	Dilated CNN layers instead of CNN layers, which is connected to Fully connected layers at the output via BLSTM layers	No
Sound Event Detection with Depthwise Separable and Dilated Convolutions [65]	2020	Depthwise CNN layer instead of CNN layers and Dilated CNN layers instead of RNN as in [8]	Yes

2.3. DOMAIN ADAPTATION IN SED

Domain adaptation techniques have been used to account for the change data distribution and train a classifier that generalizes well on the target data and only a few works in SED literature have explicitly explored domain adaptation. Amongst those many techniques, based on the availability and annotation level of the target domain data, they can be divided into following categories [9]:

1. Supervised: Few labeled samples from target domain is assumed to be available. These samples would generally be not enough to train a classifier on their own.

2. Semi-supervised: Both limited labeled data and unlabeled data are present in the training stage.

3. Unsupervised: Adaptation is performed by the use of high number of unlabeled samples from target domain.

The current application of domain adapation techniques in SED are based on unsupervised domain adaptation. Authors in [67] make available the first dataset for research into domain adaptation techniques in SED. The dataset consists of sound event classes of "baby crying", "glass breaking" and "gunshot", which are then mixed such that they are overlapping. It is then optionally further mixed with an audio sample from an acoustic scene belonging to different categories i.e, Vehicle, Outdoor and Indoor or not i.e, Clean. The SNR of sound events mixture to the acoustic scene audio is varied from -3dB to -9dB. In the same work, they implement an adversarial approach to adapt their acoustic model to target domain data. They take a baseline CRNN [8] can train it as the source model from the source domain data. Adapting the target model to the domain is performed in an unsupervised manner by first initializing its weights from the source model. They further employ a FNN as a domain classifier, which will receive latent representations of the source and target data from the source and target model respectively to distinguish the domain of its input. The target model and the domain classifier are trained using an adversarial training scheme such that the target model learns an invariant representation of the audio samples irrespective of the domain and is able to fool the domain classifier. Authors show that if the source domain is one among Vehicle, Outdoor or Indoor, the performance is either same across domains or slightly lower, which is not quite promising. However, by using the Clean dataset as source domain it is difficult to obtain good generalization performance when adapting the model for different domains.

In [68], the authors combine adversarial learning and tri-training approaches to perform SED using strongly annotated synthetic data and weakly annotated or unlabeled real dataset. Figure 2.2 depicts the overview of their approach.

The baseline approach implemented here is that the feature extractor F and classifier F_t are trained using strongly annotated synthetic data and weakly labeled data. The adversarial approach is implemented by applying Gradient Reversal Layer (GRL) from the domain classifier D to the feature extractor F, and thus ensuring obtaining a invariant representation of both synthetic and real domain data. Unlabeled samples are also

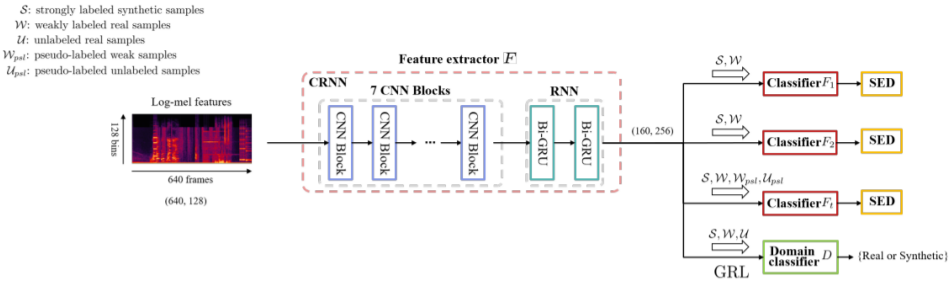


Figure 2.2: Domain adversarial approach as implemented by the authors in [68]

included along with the strong and weak samples in the training process by the use of tri-training method [69]. By just implementing the adversarial approach, they outperform the baseline and by including the tri-training process they further push ahead the performance of their system.

Authors in [70] also introduce the same approach as in [67] to use domain adaptation in SED, but unlike [67], the authors here [70] apply this technique on real recordings. Amongst the recordings they collect, they define domain gap between the source and target data as mismatch between recording equipment, mismatch in the characteristics of the sound events and finally mismatch of the background noise. Authors show that their approach is able to show improvements in relatively dificult to overcome domain gaps such as mismatch of both recording equipment and the sound events characteristics. However, in the five classes i.e. *car, children, large vehicle, people speaking, people walking* that the dataset consists of, only one class (*car*) shows improvement after adaptation while others degrade. While the authors posit that this could be due to the fact that *car* sounds are relatively simple, this could be due to the fact that the length of recordings belonging to *car* class are relatively higher than other classes in one of the datasets used and thus have more data available. However, this method is shown to partially work on source and target domain datasets of recording lengths of 1.5 and 9 hours respectively. This indicates that datasets relatively large and the unsupervised domain adaptation technique they have employed fits well, however it could mean that this approach might not work with smaller datasets.

As all of the above discussed methods belong to Unsupervised domain adaptation, the amount of data required from the target domain will be relatively high. Implementing any of these techniques would mean that we would not be able to answer of the questions defined for this research. Hence, this means that any techniques from supervised domain adaptation would be a better fit. Amongst the many techniques available in the literature, we look to adopt Augmented Cyclic Adversarial Learning (ACAL), introduced by the authors in [71] as a technique that can assist in domain adaptation in a low resource scenario. This technique builds on GAN-based CycleGAN [72], that employs a *cycle-consistency* constraint which enforces that when an example is mapped from source domain to the target and then back to the source domain again, it would re-

sult in the *same* example. As a result of using this constraint, CycleGAN can transfer the *style* of source domain data to match the distribution of target domain data, while preserving the *content* of the source domain examples. For instance, in the case of scenario that this work addresses, if the source domain data is the sound heard by the IV when it is stationary and the target domain data is the sound heard when it is driving, then content would refer to information in the features that point towards an approaching car which would have to be preserved and the style would refer to the ego-motion noise the generated examples from source domain data need to have. Authors who introduced ACAL [71], posit that if the data from target domain are scarce, the cycle-consistency constraint in cycleGAN might be too restrictive as the networks will not have enough data at hand to learn meaningful mappings to reconstruct the exact same source domain samples from its generated target domain example. Moreover, they [71] say that, during reconstruction of the source domain examples, if the content is preserved and the style matches the corresponding distribution, then it would be valid mapping and it is not necessary to obtain the exact same sample during reconstruction. By modifying the cycle-consistency constraint to match their above idea, the authors show significant gains over the baseline cycleGAN.

The reason to choose this technique (ACAL[71]) in our work is two fold. First, the technique is introduced to tackle scenarios where there is limited amount of training data in the target domain, which is similar to the scenario addressed in our work as we can expect there to be a limited amount of data collected when the IV is driving. Second, many of the domain adaptation techniques present in the literature address adaptation between visual domains only. However, the authors also have applied ACAL to an audio application wherein they adapt a speech recognition model trained on data from only one gender to the other, and have used time-frequency based audio representation in their implementation. Since this application (Automatic speech recognition) is a closely related task to SED, it would be interesting to see how ACAL performs for the same.

2.4. DATASETS FOR IV

Research into SED has seen a tremendous increase in popularity and the strong evidence of this is the increase in the number of publications and number of datasets that are publicly available for the researchers in this domain. Moreover, with machine learning techniques being extensively used in SED, the amount and quality of data is of paramount importance. The datasets available are also very diverse in nature and it is important to select the ones such that it closely relates to the application scenario.

There are few datasets which include sounds that an IV might hear during navigating amongst traffic. Authors in [73] have recorded a dataset that focuses on the traffic noise generated by the vehicles. The recordings were made at 4 different location across the city of Montevideo, Uruguay with varying background noise level. Most of the recordings here belong to sounds emanating from car passing by. Another similar dataset is the RoadCube[74] which was recorded for the purpose of vehicle classification. This dataset was recorded in Delft and it consists of 383 audio samples belonging to classes *bicycle, scooter, truck, van, car* and *no sound* i.e. sounds representing roadside noise. The record-

ing device here was 8-microphone array arranged in the shape of a cube and was placed at a distance close to the road. For the above mentioned datasets, the recordings are equivalent to the sounds perceived by the IV when it is not moving. Since it contains no recordings that include sounds generated when the IV is driving, it would not be suitable for our application. This lack of ego-motion related sounds has been addresssed by the dataset DriveSound [75]. This dataset was recorded from a microphone array mounted on top of an IV. The recordings were performed in both ego-motion conditions i.e. static and driving. However, the recordings capture sounds from vehicle that in the line-of-sight of the IV. While this is a valid application of acoustic perception in IVs, relying only on this dataset to design a system for our application might not be the way to go forward. This is because the recordings in DriveSound might not include cues that might exist only in non line-of-sight propagation of sound. Table 2.2 summarizes the details of the datasets discussed above.

Table 2.2: Datasets in the literature having sounds of interest to an IV.

Dataset	Recording Type	Number of Classes	Total Length (min)
MAVD-Traffic[73]	Real	21	233
RoadCube[74]	Real	6	25
DriveSound[75]	Real	12	20

2.5. SED EVALUATION

Evaluation of any system is critical in understanding the extent of its performance. However, care should also be taken in selecting the method of evaluation such that performance metrics obtained here directly indicates the real-life applicability of the system. Moreover, identifying such metrics is also crucial as it allows fair comparison to the other competing methods if any. Authors in [76] carry out an extensive survey of different metrics used in SED. This section summarizes their survey and highlights the metrics used by most researchers in SED literature.

Given the nature of SED prediction, the measurement of the performance can be performed in two different ways. To elaborate, the comparison of the system output and the reference annotations can be done either on a short fixed-length intervals (*Segment-based metrics*) or at each event level (*Event-based metrics*). Within the scope of each measurement choice, it is then necessary to define the nature of correct predictions and otherwise so that the metrics can be computed. The counts of such correct and the erroneous predictions are referred to as *intermediate statistics*. For most metrics in SED, these statistics are termed as True Positives (TP), False Positives (FP), False Negatives (FN), True Negatives (TN). Some metrics such as Error Rate, employ statistics derived from TP, FP, FN, TN and here they are termed as Substitutions (S), Insertions (I) and Deletions (D). Using the two measurement choices and the intermediate statistics listed above, many works in the literature mainly report F1-score and the Error Rate. Figures 2.3, 2.4 depict the computation of F1-score and Error Rate for both segment and event

based choices of measurement respectively.

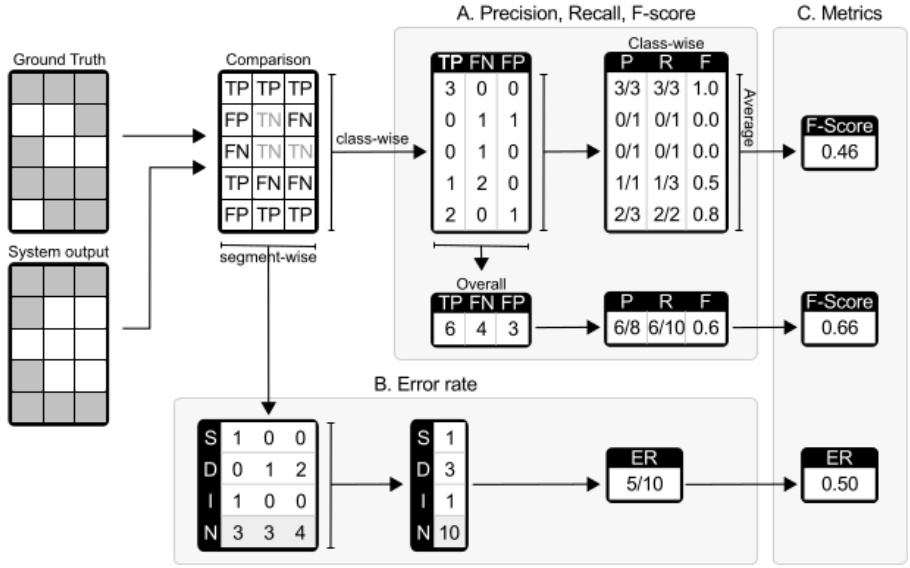


Figure 2.3: In *segment-based metrics*, the comparison between the system output and the reference is only for short time segments of the audio signal. If a shorter segment length is used, the evaluation would be precise. However, this dictates that the annotations for the dataset should be of very high quality. Image from [76]

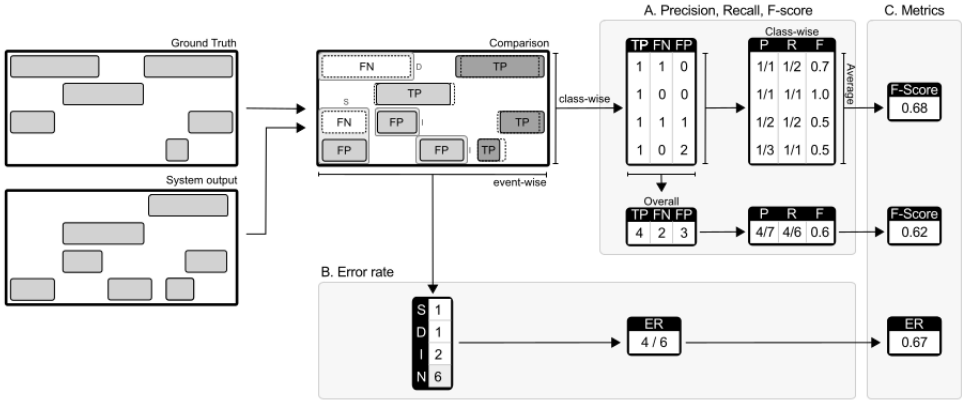


Figure 2.4: In *event-based metrics*, the comparison between the system output and the reference happens event by event. Here, true negatives are not defined as it is difficult to assess the number of such events towards the calculation of this metric. Image from [76]

Additionally, in a multi-class scenario which normally is the case in SED, aggregation of intermediate statistics can be performed in two ways: ① Instance-based averaging

or micro-averaging, which weighs all the individual decisions equally, regardless of the class they belong to. ② Class-based averaging or macro-averaging, which weighs each decision such that equal importance given to each of the classes present. The intermediate statistics are first aggregated class-wise to compute the metrics class-wise. The overall performance is considered to be the average of these class-wise metrics.

2.6. CONTRIBUTIONS

The review in this chapter has pointed that even though SED is a highly researched topic, no work has implemented SED techniques for acoustic perception in IVs. Consequently, there exists no dataset that can be used directly used to prepare an acoustic model to be deployed on an IV for acoustic perception. Additionally, currently implemented domain adaptation techniques in SED are unsuitable for application in IV domain. This thesis addresses these shortcomings with the following contributions:

- SED based techniques are more suitable to implement classification based acoustic perception for an IV. Deep learning based techniques are currently the state-of-the-art and are widely applied to different SED applications. Among them, the CRNN based architecture is more commonly implemented to design the acoustic model and hence has been evaluated on many audio datasets. The contribution of this work is the implementation and evaluation of the state-of-the-art CRNN based acoustic model for the IV domain.
- As highlighted in Section 2.4, the datasets available in the literature are not adequate for our application. However, as part of the developing a localization based approach (Appendix A), a dataset that captured the sounds perceived by an IV when coming across a scenario described by Figure 1.1 was collected. Evaluation of an classification based approach on this dataset and assistance in the data collection process are also among the contributions arising from this work.
- In the SED literature, domain adaptation techniques have been scarcely explored and in addition the current techniques applied would not be suitable for us as they would require a lot of data. The selected domain adaptation, ACAL, has not been applied for SED and the contribution of this work would be to explore if it is feasible to do so in the scope of the application scenario considered here.

3

METHODOLOGY

The main goal of the thesis is to evaluate whether a single microphone setup can identify an approaching vehicle behind an intersection. Given the setup, Sound Event Detection (SED) is the technique available in the literature that can help realize this approach. As mentioned earlier in Chapter 1, the existing SED pipeline in the literature consists of the following stages: *audio representation* and *classification*. Figure 3.1 depicts the overview of the proposed system in this work. The sound signal from a single microphone is served as the input to the audio representation stage which converts the audio in raw waveform to Log Mel Spectrogram features. The acoustic model in the classification stage takes in these features and outputs the labels which indicate the presence/absence of approaching vehicle for each short time frame. Finally, majority voting is performed to aggregate the frame-wise predictions to obtain the final decision on the presence/absence of approaching vehicle.

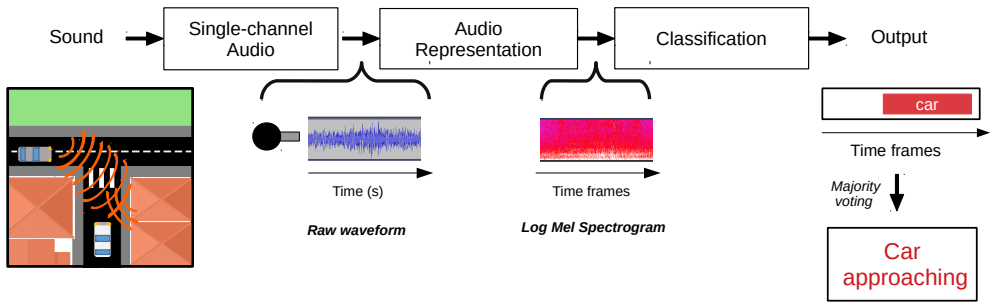


Figure 3.1: Overview of the acoustic perception proposed in this work

Sections 3.1 and 3.2 will describe in detail about the choices made in this work for both the stages of SED pipeline. Further, Section 3.3 explains the chosen domain adaptation technique, ACAL, which would allow us to train our acoustic model from few sam-

ples from the selected target domain. Finally, Section 3.4 elaborates the evaluation procedure and the performance metrics used in this work.

3.1. AUDIO REPRESENTATION

The sound captured by the microphone are represented as the time domain features. If these features are used, the only processing step before the classification is the division of incoming signal into segments of short temporal length. This implies that less time is consumed in the preparation of the feature vector for the classification stage. Hence, this makes it an attractive choice for our system, as for any perception system that is implemented on IV, real-time inference is very important. However, it has been found that deriving frequency information is useful and performance is better when such a representation is used [11].

As discussed in Section 2.1.2, many recent works prefer time-frequency representation over frequency only representation. While most of them do not compare or justify this preference, some of the recent works [26][27] have shown that MFCCs (frequency domain) do not outperform Log mel spectrogram features (time-frequency domain). Employing other representations such as localization based features (e.g. GCC-PHAT) is not feasible as they would require signal from than one microphone. Thus we would fail in answering the main research question. So, that leaves us with the time-frequency domain representation, which is chosen in this work to obtain the acoustic feature vectors for the classification stage of our system. In particular, the Log mel spectrogram features will be extracted from the raw waveform. The reason behind this choice is that these features are the most popular choice amongst the researchers in SED and even the state-of-the-art techniques employ these features. Moreover, since there is no prior work which addresses SED in the IV domain, this is a good starting point.

EXTRACTION OF LOG-MEL SPECTROGRAM

The extraction of time-frequency domain features is comprised of three stages. First, the signal it divided into short time frames of length. The selection of time length of these short frames is an important choice since it directly affects the frequency resolution. Having a higher frame length results in worse temporal resolution, however it means that the frequency resolution is increased. Second, these frames are then multiplied with a window function. This is done to reduce the effect of discontinuities at the borders of each frame. Typically, Hamming function is used the case of SED. Finally, frequency information of this short frame is computed by performing the Discrete Fourier Transform (DFT). To obtain a smoother representation over time, these short frames are overlapped in temporal dimension. The resulting feature matrix that is obtained by concatenation of the frequency domain vectors for the consecutive time frames of a sound clip is called the *spectrogram*.

From the spectrogram features, mel spectrogram is obtained by applying the *mel filterbank* at each time frame over the magnitude spectrogram. Mel filterbank utilizes the mel scale which consists of the triangular filters, whose bandwidths increase as central frequency for the filters increase. The frequency axis (f) of the spectrogram is trans-

formed into Mel scale (mel) by using the formula 3.1 as introduced in [77]. The transformation into mel scale results in higher resolution for the lower frequency range and vice versa, just as humans perceive it. The logarithm of the resulting time-frequency matrix is taken to obtain Log mel spectrogram.

$$mel = 2595 \cdot log_{10}\left(1 + \frac{f}{700}\right) \tag{3.1}$$

3.2. CLASSIFICATION

In the recent literature, acoustic models have primarily based on deep learning techniques. Amongst them, the Convolutional Recurrent Neural Networks (CRNN) have been extensively used for different applications. While there are some works [63][65] which propose other architectures that slightly outperform CRNN, they are only tested on limited number of datasets of which most of them are synthetic by nature. Hence, a decision is made to adopt CRNN based acoustic model for our application. The structure of the CRNN network adopted here closely follows to that work [8], which introduced it for the task of polyphonic SED.

3.2.1. CRNN ARCHITECTURE

The input data to the network is the time-frequency of an audio segment in the form of Log mel spectrogram features, $X \in \mathbb{R}^{N \times T}$, where N is the number of mel band energies and T is the resulting number of time frames of the spectrogram. These features are then fed into a Convolutional Neural Network (CNN) block. First, the input data to this block is fed to convolutional layers with two-dimensional filters. To introduce non-linearity in the network, the feature maps are passed through an activation function and in this work Rectified Linear unit (ReLU) is chosen to do so. Then, the output of the activation function is normalized using the Batch Normalization technique [78]. Following this, non-overlapping max pooling over frequency axis is performed to reduce the dimensionality of the data and enhance frequency invariance [8]. Dropout [79] is applied following the above operations. The combination of all these CNN blocks can be treated as a feature extractor for the next step that follows. After passing through L_c CNN blocks, the output of the CNN feature extractor is a tensor $H \in \mathbb{R}^{M \times F' \times T}$. Here, M is the number of feature maps for the convolutional layer in each of the CNN block, F' is the number of frequency bands remaining after the max pooling operations. The time dimension (T) is equal to the one in the input feature matrix as max-pooling is done only on the frequency axis and the inputs to the convolutional layers are padded by zeros.

The output tensor H from the CNN feature extractor are stacked over the frequency axis to yield $H \in \mathbb{R}^{(M.F') \times T}$. This is then fed into the RNN block which consists of L_r stacked recurrent layers. If the stacked output H is considered as a concatenation of frames $h_t^{L_c}$, where $t \in [0, T]$, then for each frame t a output hidden vector h_t is computed as,

$$
\begin{aligned}
h_t^{L_c+1} &= \mathscr{F}(h_t^{L_c},\ h_{t-1}^{L_c+1}) \\
h_t^{L_c+2} &= \mathscr{F}(h_t^{L_c+1},\ h_{t-1}^{L_c+2}) \\
&\vdots \\
h_t^{L_c+L_r} &= \mathscr{F}(h_t^{L_c+L_r-1},\ h_{t-1}^{L_c+L_r})
\end{aligned} \tag{3.2}
$$

Here, the function $\mathscr{F}$ represents a unit of the chosen architecture for RNN, which in this work is the Gated Recurrent Unit (GRU)[80]. GRU and its variant Bi-directional GRU (Bi-GRU) have been favored over another popular RNN architecture Long short term memory (LSTM). This is because the authors in state that [8] GRU utilizes lesser parameters while maintaining similar performance as LSTM for SED task. Among the GRU variants, Bi-directional GRU is not preferred as they require inputs from future time steps to make decision about the current one *i.e.* they are non-causal. For a real-time system like ours, this would be not ideal and hence uni-direction GRU is used in the RNN block.

The RNN block is followed by a single feedforward layer $\mathscr{G}$ with sigmoid activations, which serves as a output layer of the network. Each time frame ($h_t^{L_c+L_r}$) in the tensor S is then fed into these feed-forward layer to obtain:

$$
h_t^{L_c+L_r+1} = \mathscr{G}(h_t^{L_c+L_r}) \tag{3.3}
$$

The output $h_t^{L_c+L_r+1}$ are considered as the class-probabilities that indicate the presence of each sound event, $k = 1, 2, ..K$, in each time frame and can be expressed as:

$$
p(y_t(k)|x_{0:t}, \theta) = h_t^{L_c+L_r+1} \tag{3.4}
$$

where, K is the total number of sound event classes that the network is trained to recognize. To obtain the predictions $\hat{y} \in \{0,1\}^{K\times T}$ for each sound event k, this output is then thresholded over constant $C \in (0,1)$ as:

$$
\hat{y}(k) = \begin{cases} 1, & p(y_t(k)|x_{0:t} \geq C \\ 0, & \text{otherwise} \end{cases} \tag{3.5}
$$

The scenario that is addressed in this work, *i.e.* the task of predicting if there is an approaching vehicle behind the blind intersection, can be cast as a binary classification problem. To elaborate, when the IV is driving towards the intersection, the predictions can be $\hat{y}_t = 1$ or $\hat{y}_t = 0$, which indicates the presence or the absence of an approaching vehicles respectively. This means that the final predictions of the acoustic model would be $\hat{y} \in \{0,1\}^{1\times T}$ as $K = 1$. Figure 3.2 depicts the overview of the CRNN architecture used in this work.

3.2.2. LOSS FUNCTION

The acoustic model would be trained to predict an approaching vehicle in a supervised setting. Since training of deep networks is essentially an optimization problem, it can be formulated as follows:

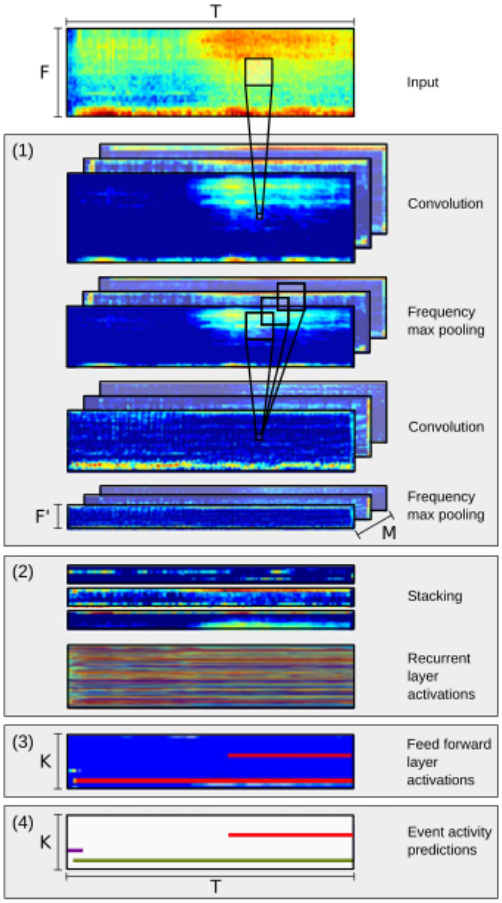


Figure 3.2: Overview of the CRNN architecture. (1): Convolutional operations are performed over the log mel spectrogram input along with max pooling only along frequency axis; (2) Activation maps are then stacked along frequency axis and then fed to Recurrent layers; (3) Classification of the extracted features are then done by a feed forward network (4) Event probabilities are then thresholded and binarized to complete SED. Image from [8]

$$\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} L(y^i, f_{crnn}(x^i, \theta)) \tag{3.6}$$

where f_{crnn} is the acoustic model and θ its parameters, L is the loss function, N is the total number of training samples, x^i is the feature vector of the ith sample (in this work, $x^i \in X$) and y^i is the corresponding label. Selection of loss function is key as it highly influences the effectiveness of the model to perform the learned task and hence it must be chosen appropriately. Since we are dealing with a binary classification problem, the loss function most commonly used for such problems is the Binary Cross-Entropy (BCE) loss. Additionally, the original proposal for CRNN in [8] and many other works

that implement this architecture, also employ BCE loss for training. Hence, it is decided that the same will be adopted as the loss function for this work.

3.3. DOMAIN ADAPTATION

The chosen Domain adaptation technique - Augmented Cyclic Adversarial Learning (ACAL) [71] is an extension of CycleGAN[72], a framework used to map inputs from one domain to another. In this section, these two techniques are detailed along with a brief introduction to Generative Adversarial Network[81], which is the core concept behind the same. Further, the training procedure of ACAL with the CRNN based acoustic model is also explained.

3.3.1. GENERATIVE ADVERSARIAL NETWORKS

Generative Adversarial Network (GAN) [81] is a framework which aims to learn the distribution of the data $P_{data}(X)$ using adversarial training. Here, the fake data samples are produced by a generator network $G(z)$ that is similar to the data sampled from $P_{data}(X)$. A discriminator network $D(x)$ then learns to determine whether a given sample is generated/fake or if it is sampled from $P_{data}(X)$. The networks are trained until the discriminator can no longer successfully distinguish the fake samples from the real ones. Training of the networks is performed by alternatively optimizing the objective function and this can be formulated as:

$$\min_G \max_D L_{adv}(G,D) = \mathbb{E}_{x\sim P_{data}(X)}[\log(D(x))] + \mathbb{E}_{z\sim P_z(Z)}[\log(1-D(G(z)))] \tag{3.7}$$

3.3.2. CYCLEGAN

The CycleGAN introduced by authors in [72], extends the GAN framework to multiple domains. Using this approach, mapping function between two domains S (source) and T (target) are learnt by having two generators $G_{S\to T}$ and $G_{T\to S}$ that are trained in an adversarial manner with two discriminators D_S and D_T. The adversarial objective function can be defined as:

$$\begin{aligned} L_{adv}(G_{S\to T}, D_T) &= \mathbb{E}_{x\sim P_T(X)}[\log(D_T(x))] + \mathbb{E}_{x\sim P_S(X)}[\log(1-D_T(G_{S\to T}(x))] \\ L_{adv}(G_{T\to S}, D_S) &= \mathbb{E}_{x\sim P_S(X)}[\log(D_S(x))] + \mathbb{E}_{x\sim P_T(X)}[\log(1-D_S(G_{T\to S}(x))] \end{aligned} \tag{3.8}$$

where $P_S(X)$ and $P_T(X)$ indicate the source and target domain distributions. Further, cycle-consistency loss is introduced so that any examples that was mapped to other domain can be inverted back to its original domain. According to the authors, the adversarial losses in Equation 3.8 alone cannot guarantee that the generators can learn to map an individual input from the source domain to the desired output as it could be mapped to any random permutations of data in the target domain. The cycle-consistency loss, according to the authors[72], reduces the space of possible mapping functions and thus improves performance. This loss can be defined as:

$$L_{cyc}(G_{S\to T}, G_{T\to S}) = \mathbb{E}_{x\sim P_S(X)}(\|G_{T\to S}(G_{S\to T}(x)) - x\|_1) + \mathbb{E}_{x\sim P_T(X)}(\|G_{S\to T}(G_{T\to S}(x)) - x\|_1) \tag{3.9}$$

The networks are then trained with the combination of the two losses whose relative importance is controlled by the weight λ and the training can be formulated as:

$$\min_{G_{S\to T}, G_{T\to S}} \max_{D_S, D_T} L(G_{S\to T}, G_{T\to S}, D_S, D_T) = L_{adv}(G_{S\to T}, D_T) + L_{adv}(G_{T\to S}, D_S) + \lambda L_{cyc}(G_{S\to T}, G_{T\to S}) \tag{3.10}$$

3.3.3. AUGMENTED CYCLIC ADVERSARIAL LEARNING

In the cycleGAN framework, the adversarial objective encourages generators to produce samples close to the true distribution, whereas the cycle-consistency loss encourages identity mapping. When the samples from both the domain are not limited in number, then the balancing of these objectives is not an issue and the training can converge well. However, when the target domain samples are less in number, the target discriminator D_T will overfit and it will act like a delta function on the samples from $P_T(X)$. This means that the feedback from D_T to the generator $G_{S\to T}$ would be limited (see Equation 3.8). Hence, in the final objective (Equation 3.10), the cycle-consistency loss would outweigh the adversarial loss.

According to the authors who introduced ACAL [71], the reasons for the above issue is two fold. First, the cycle-consistency loss which enforces exact reconstruction of the mapped sample back to original, is too strong. Second, discriminator alone is insufficient for the generator to learn effective mapping of samples to the target domain. The authors then use a task specific model in both cycle-consistency and the adversarial loss to address these problems. To elaborate, the losses in Equations 3.8 and 3.9 are now respectively transformed as:

$$\begin{aligned} L_{adv-ACAL}(G_{S\to T}, D_T, M_T) &= \mathbb{E}_{x\sim P_T(X)}[\log(D_T(x))] \\ &+ \mathbb{E}_{x\sim P_S(X)}[\log(1 - D_T(G_{S\to T}(x))] \\ &+ \mathbb{E}_{(x,y)\sim P_T(x,y)}[L_{task}(M_T(x,y))] \\ &+ \mathbb{E}_{(x,y)\sim P_S(x,y)}[L_{task}(M_T(G_{S\to T}(x),y))] \\ L_{adv-ACAL}(G_{T\to S}, D_S, M_S) &= \mathbb{E}_{x\sim P_S(X)}[\log(D_S(x))] \\ &+ \mathbb{E}_{x\sim P_T(X)}[\log(1 - D_S(G_{T\to S}(x))] \\ &+ \mathbb{E}_{(x,y)\sim P_S(x,y)}[L_{task}(M_S(x,y))] \\ &+ \mathbb{E}_{(x,y)\sim P_T(x,y)}[L_{task}(M_S(G_{T\to S}(x),y))] \end{aligned} \tag{3.11}$$

$$\begin{aligned} L_{cyc-ACAL}(G_{S\to T}, G_{T\to S}, M_S, M_T) &= \mathbb{E}_{(x,y)\sim P_S(X,Y)}[L_{task}(M_S(G_{T\to S}(G_{S\to T}(x))), y)] \\ &+ \mathbb{E}_{(x,y)\sim P_T(X,Y)}[L_{task}(M_T(G_{S\to T}(G_{T\to S}(x))), y)] \end{aligned} \tag{3.12}$$

Here, M_S and M_T are the task specific model that if given the training samples X and their corresponding labels Y from a particular domain (S or T), can learn to carry out the required task after its parameters are optimised in during training by using task-specific loss L_{task}. The cycle-consistency loss used here ($L_{cyc-ACAL}$) is less strict as it does not demand an exact reconstruction of the original example and the loss would not increase as long as the content of the reconstructed sample matches that of the original. From the adversarial loss ($L_{adv-ACAL}$), the generators can now utilize the feedback from the conditional probability distribution ($P_S(Y|X)$ or $P_T(Y|X)$ learnt by the task-specific model in addition to the feedback from the discriminator. Figure 3.3 compares the schematic of CycleGAN and the ACAL method used.

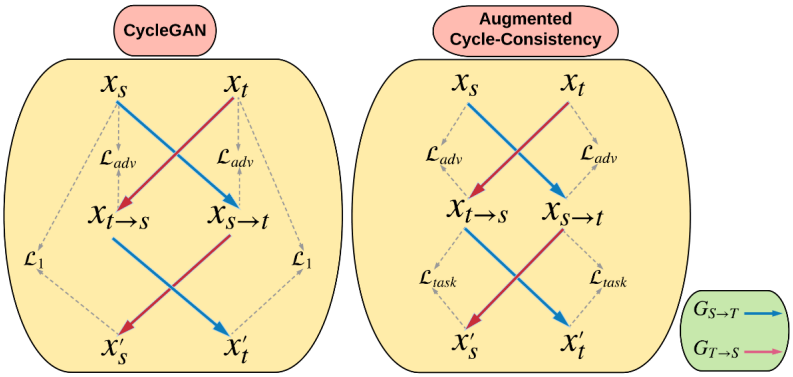


Figure 3.3: The comparison of cycleGAN [72] and ACAL methods[71]. In addition to helping out the generators produced higher quality samples, the task-specific model used in ACAL also would be trained by both fake and limited real samples in the target domain. In a sense, this can be considered as training a model with data augmentation without any manual intervention. Image from [71]

3.3.4. INTEGRATION OF CRNN AND ACAL

The main idea of introducing a domain adaptation technique in this work is to ease data collection process by having to collect majority of total data samples from when the ego-vehicle was `static`. In this setting, the source domain would be the audio samples collected when the ego-vehicle is `static` and the target domain would be those collected when the ego-vehicle is moving towards the intersection i.e. `dynamic`. Accordingly, M_S and M_T would be the CRNN based acoustic model that would be trained on `static` and `dynamic` domains and the task-specific loss L_{task} would be the BCE loss. The training procedure of the domain adaptation approach is outlined in Algorithm 1.

3.4. EVALUATION

In the literature, many of the applications that employ acoustic models to perform SED task commonly report F1-scores and Error Rate (ER) and contrast the performance of their approach to that of baseline using them. However, the computation of F1-score and the ER does not consider the True Negatives (TN). As a result, in a binary classification setting, if there is an imbalance in the dataset, wherein the either of the one class

Algorithm 1: Domain Adaptation using ACAL

Input : `static` data $P_S(x, y)$, `dynamic` data $P_T(x, y)$, pre-trained source task model M_S
Output: Target task model M_T
while *epochs* ≤ *total_epochs* **do**
  Sample (x_t, y_t) from P_T;
  Finetune source model M_S on (x_s, y_s) and $(G_{T\to S}(x_t), y_t)$);
  Train task model M_T on (x_t, y_t) and $(G_{S\to T}(x_s), y_s)$);
end

dominates in sample count, then these metrics would not be a true indicative of the performance. To address this particular shortcoming, the system implemented in this work would be evaluated by Balanced Accuracy. For further clarity on the predictions of the acoustic model, confusion matrix is also reported.

The predictions $\hat{y}$ of the acoustic model, as explained in Section 3.2.1, is of the shape $1 \times T$, where T is the temporal length of the input acoustic feature vector. In the real-time implementation of this system, the input segment will picked such that the frame $t = T$ will be the current time with respect to the IV. This means that as the IV is navigating through the environment, the acoustic model is making predictions on the audio segment that is heard since time that translates to T bins in the time-frequency representations in the feature vector. From IV's perspective, predictions made after *listening* to its current input segment are important in the further decision making process *i.e.* in this scenario that would be to either stop before or continue into the intersection. Hence, the frame-wise predictions are aggregated into a single prediction such that it would ease the further decision making process. This is realized by performing a majority voting on the final prediction vector $\hat{y}$ and thus obtaining a single output indicating the presence of car or its absence. These aggregated predictions will be compared with the ground truth to compute the metrics discussed in this section.

3.4.1. BALANCED ACCURACY

Accuracy (Acc) measures the classifiers ability to make the correct predictions and Balanced Accuracy (Bal_Acc) follows the same suit. However, unlike Accuracy, balanced accuracy is robust to class imbalance in the dataset. Following are the definition for the two metrics:

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.13}$$

$$Bal\_Acc = w.\frac{TP}{TP + FN} + (1 - w).\frac{TN}{TN + FP} \tag{3.14}$$

Here, TP, TN, FP, FN are the number of True Positives, True Negatives, False Positives and False Negatives respectively and w is the relative weight between the two terms. The reason to chose Balanced Accuracy as our main metric is that, in a binary-classification

setting, if the dataset is imbalanced, then evaluation via F1-score will present an unfair estimation of the performance of the classifier. To illustrate, suppose there are 8 positive examples and 3 negative examples in the test set, and if the classifier predicts every example as positive, which means $TP = 8, FP = 3, FN = 0, TN = 0$, then if $w = 0.5$,

$$F1 - score = 0.8421; \quad Acc = 0.7273; \quad Bal\_Acc = 0.5$$

A classifier predicting only one class suggests its performance is random and looking at the values of metrics above, only Balanced Accuracy indicates the classifier performance is random. Hence, in this work, Balanced Accuracy will be reported instead of the widely used F1-Score and ER.

3.4.2. CONFUSION MATRIX

A confusion matrix is a table which visualizes the performance of the classifier by reporting the TP, TN, FP and FN and in this work, the matrix is reported in the following format:

Table 3.1: Confusion Matrix for a binary-classification problem.

	Predicted positive	*Predicted negative*
Positive class	True Positive (TP)	False Negative (FN)
Negative class	False Positive (FP)	True Negative (TN)

3.4.3. CROSS VALIDATION

The collection of data for a data-driven learning process is always tedious as large amounts of data is usually required. If the dataset size is not large enough, then it would be incorrect to estimate the predictive power of the acoustic model by training and evaluating it on a simple train and test split of the data. This is mainly because, the small test set might not be entirely representative of the data and this might lead to over the top performance estimations of the acoustic model. Furthermore, it would be difficult to increase the size of this set as there would not be enough samples to train the acoustic model. Hence, in this work, the acoustic model will be evaluated using cross-validation technique as it can provide an less optimistic performance estimation of the same.

Specifically, k-fold cross validation techniques is employed to evaluate the performance of the acoustic model. The dataset or a given subset of the same, is divided into k folds, of which $k-2$ folds serve as train set, and the other two serve as validation and the test set. As the training of the acoustic model progress, the best performing model on the validation set, that has the lowest value of loss function, will be evaluated on the test set. This process will be repeated k times such that all the samples in the experiment will be present in the test set exactly once (see Figure 3.4).

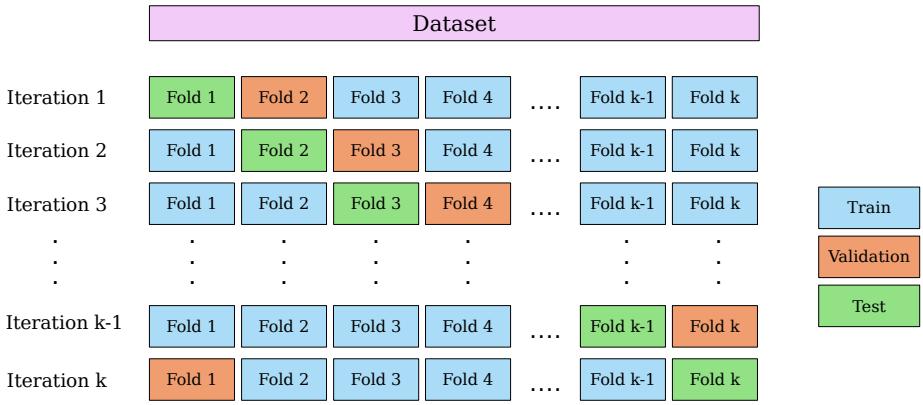


Figure 3.4: Overview of k-fold cross validation. The metrics averaged over the k test folds will be the performance estimate of the acoustic model.

4

EXPERIMENTS

This chapter describes and discusses the experiments that were performed to evaluate the feasibility of the single-microphone acoustic perception system. Section 4.1 details the dataset that is used in this work. The procedure to choose the different hyperparameters for the CRNN acoustic model is detailed in Section 4.2. In Section 4.3, the acoustic models predictions over the entire recordings is visualized and in Section 4.4 analyzes the quality of acoustic cues using these visualizations. These experiments give an insight as to how the model behaves when it is deployed on an IV that encounters a vehicle approaching from blind corner (see Figure 1.1). The experiment in Section 4.5 gives an idea as to how well the acoustic model generalizes when presented with samples from outside the training location. Finally, section 4.6 deals with the application of the chosen domain adaptation technique on the dataset used in this work.

4.1. DATASET

A new dataset was collected to validate the implementation of localization based acoustic perception (Appendix A) to predict an approaching vehicle at a blind intersection. This work also utilizes the same dataset to discern if a classification based approach with data from only a single microphone is feasible or not. Here in this section, the details of this dataset are presented.

4.1.1. HARDWARE SETUP

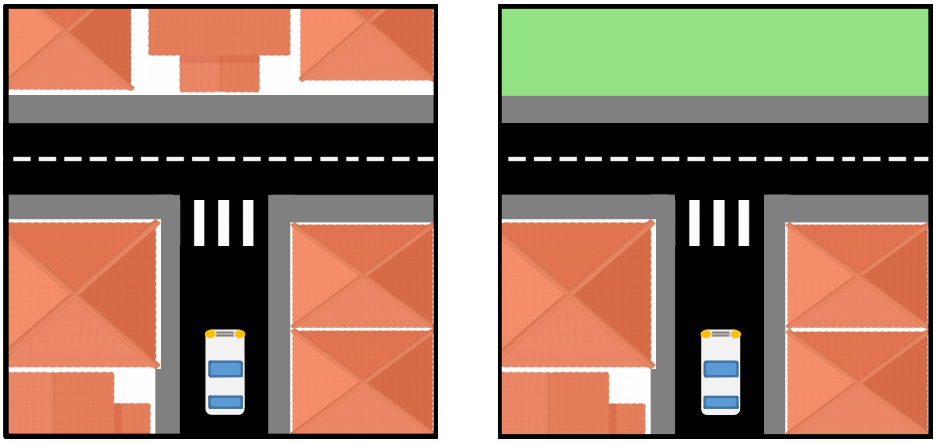
The recordings are captured from a custom built microphone array that is mounted on top of the IV, which is a hybrid Toyota Prius. The array is composed of 56 ADMP441 MEMS microphones which records audio at 48KHz. Further, they are positioned irregularly within a square of size $0.8m \times 0.7m$ and also houses a camera and a processing unit. Looking at the Figure 4.1, it can be seen that a single microphone system would be much easier to integrate than the multi-microphone system. The IV is also equipped with a stereo camera just behind the front windshield, which is used along with microphones

to get a complete audio-visual recording of the data points collected.

Figure 4.1: The microphone array used in the collection of dataset used in this thesis. Multiple microphones was used in the collection of this dataset as it was used in the implementation of localization based approach.

4.1.2. DATA COLLECTION

The collection of the recordings were performed in two ego-motion modes of the IV i.e. `static` and `dynamic` and this will be used to reference the two large subsets in this dataset. Recordings were captured at five different locations around the city of Delft and the locations were chosen such that layout is an intersection with blind corners. Table 4.1 reports the location name and the number of recordings captured at each location. Further categorization of these locations (type A and type B) can be made based on the similarities of the layout of buildings around them (see Figure 4.2). At all these locations, the IV would approach the intersection with blind corners on either side, such that another vehicle might approach it from the left or the right blind corner. As the data is collected using a multi-microphone setup, the audio files are multi channel by nature. During the experiments, data from a single channel alone is used. Finally, due to the hybrid nature of the IV, about 70% of the total `dynamic` recordings and about 18.5% of the total `static` recordings contain engine noises. The engine automatically switches on and off when the battery is to be charged.



(a) Type A (b) Type B

Figure 4.2: Schematic of two location categories. The vehicle depicted in both of the pictures is the IV with the microphone (not shown) mounted on top.

RECORDING PROCEDURE

- *Static Recordings:* In these recordings, at every location, the IV is positioned such that the corners are visible in the camera frame (see Figure 4.3) indicating that the approaching can only be seen when the IV is very close to the intersection. On average, the position where the IV is situated is about 7 – 10m away from the intersection. Each recording is started when an independent observer positioned at the intersection sees a vehicle approaching one of the blind corners and is stopped when the said vehicle passes beyond the other corner. As the IV is not moving, there is no risk to any of the other road users during the recordings. The recordings include different types of passing vehicles, in addition to the vehicle (2010 Škoda Fabia 1.2 TSI) driven by one of the team members.

- *Dynamic Recordings:* Capturing recordings when the IV is moving, and while there is a vehicle approaching the blind corner, is risky. Hence, it is decided that the other approaching vehicle would always be driven by one of the team members. As with the static recordings, there is an independent observer who would signal the drivers to start from their respective positions such that the two cars would meet at the intersection at the same time. Thus simulating the dangerous scenario introduced at the start of this report.

4.1.3. DATA PROCESSING

The recordings captured during the collection of the dataset were of variable length. Further, the entire length of the recordings cannot be used for training the acoustic model as it is not directly relevant to the scenario that is addressed in this work. For instance, at the start of `dynamic` recordings, the IV would be stationary and well away from the intersection. This is not representative of the scenario and annotation of the recordings

Table 4.1: Number of recordings captured at each location. The IDs assigned for each row indicate the ego-motion of the IV during recordings and the location along with its type. For e.g. in the ID SA1, the ego-vehicle is static and the recordings are captured at Annaboogerd, which is the first location in Type A category.

ID	Location Name	`car`	`no car`	Total
SA1	Annaboogerd	30	30	60
SA2	Kwekerij	41	49	90
SB1	Willem Dreeslaan	41	32	73
SB2	Vermeerstraat	55	43	98
SB3	Geerboogerd	45	45	90
DA1	Annaboogerd	38	37	75
DA2	Kwekerij	15	13	28
DB1	Willem Dreeslaan	35	36	71
DB2	Vermeerstraat	22	22	44
DB3	Geerboogerd	38	36	74
SAB	*Total Static*	212	199	411
DAB	*Total Dynamic*	148	144	292

Figure 4.3: View of the intersection at a *Type B* location (SB3) from the front camera of the IV during the static recordings.

is required to train the model with the relevant part of the recordings. For the `static` recordings, the time stamp at which an approaching vehicle is in direct line-of-sight has

been annotated. This is performed manually by going through the video captured by the front camera of the IV for each of the recordings. If the approaching vehicle is partially visible, then it is considered to be in line-of-sight. Since the camera frame rate is around 10Hz, a precise estimation of the time at which the vehicle is in line-of-sight is not possible. Hence, it has been decided (Appendix A) that the time stamp of the last image before the incoming vehicle is visible, would be annotated as t_0. In the `dynamic` recordings, the relevant time stamp in the recordings would be when the moving IV is closer to the intersection irrespective of an approaching vehicle behind the corner. Hence, the time stamp when the IV is at the same position as in the `static` recordings (as per location) is annotated as τ_0.

With a single microphone system, it is impossible to differentiate whether the vehicle is approaching from the left or right blind corner. Hence, the acoustic model will perform a binary-classification of which the two classes are identified as `car` and `no car`. Once the time-stamp annotation is complete, samples of length δT_s belonging to the four sub-classes *i.e.* `left, right, front` and `none` are extracted. The `left` and `right` samples are extracted when the approaching vehicle is beyond line-of-sight. For the `static` recordings, the start and the end time stamp of the extracted signal corresponds to $[t_0 - \delta T_s,\ t_0]$. For the `dynamic` recordings, these samples are extracted such that they are centered around τ_0 i.e. $[\tau_0 - 0.5\delta T_s,\ \tau_0 + 0.5\delta T_s]$. The `front` samples are extracted when the approaching vehicle is in line-of-sight. The extraction time window of these samples correspond to $[t_e - \delta T_s,\ t_e]$, wherein $t_e = t_0 + 1.5s$ for `static` recordings and $t_e = \tau_0 + 1.5s$ for `dynamic` recordings. The offset value of $1.5s$ has been manually estimated during the annotations of all the recordings. For the `none` sub-class, samples are extracted from recordings where there are no vehicles approaching the intersection. These samples can be extracted anywhere within the `static` recordings. Hence, the extraction time window corresponds to $[t_e - \delta T_s,\ t_e]$, where $t_e = t_{end} - 3s$ and t_{end} is the time stamp of the last camera frame of the recording. For the `dynamic` recordings, the `none` sub-class samples are extracted the same way as followed for `left` and `right` samples.

After the samples are extracted, the raw waveform is converted into Log mel Spectrogram features before it is fed into the acoustic model and the computation details of the same are mentioned in Section 3.1. The input audio sample is divided into short audio frames of $40ms$ with 50% overlap. Each short frame is then multiplied with Hamming function and Fourier transform of the same is computed to get a spectrogram of the input audio signal. Mel filterbank of $N = 128$ bands spanning from 0Hz to 23999Hz is used to convert the spectrogram into mel spectrogram. The upper limit on the frequency (23999Hz) is determined by the Nyquist frequency. The values for the other parameters, except the number of bands in mel filterbank, are the chosen based on their usage in the literature. The number of bands for mel filterbank generally used in the literature is 40, but to obtain a finer frequency resolution for a more detailed audio representation, we opt for 128 bands.

Table 4.2: Number of samples extracted for each sub-class: `front`, `left`, `none` and `right`.

ID	Name	`front`	`left`	`none`	`right`
SA1	Annaboogerd	30	14	30	16
SA2	Kwekerij	41	22	49	19
SB1	Willem Dreeslaan	41	17	32	24
SB2	Vermeerstraat	55	28	43	27
SB3	Geerboogerd	45	22	45	23
DA1	Annaboogerd	38	19	37	19
DA2	Kwekerij	15	7	13	8
DB1	Willem Dreeslaan	35	18	36	17
DB2	Vermeerstraat	22	10	22	12
DB3	Geerboogerd	38	19	36	19
SAB	*Total Static*	212	103	199	109
DAB	*Total Dynamic*	148	73	144	75

4.2. HYPERPARAMETER OPTIMIZATION

Hyperparameters refer to the parameters that are used to configure the overall learning process of a model. Tuning these hyperparameters is critical as they can drastically affect the performance of the model. Further, there is no single value or a narrow range of values that is universally applicable for any of these hyperparameters regardless of the application of a model. Hence, a sound strategy is required to tune these hyperparameters and this section deals with the same for the CRNN-based acoustic model.

To methodically arrive at the best possible architecture, a grid-search strategy is employed over certain hyperparameters. Different hyperparameter values have been used for each dataset by researchers to train their CRNN-based acoustic model. Further, this acoustic model has not been applied before on a dataset similar to the one used in this thesis. Hence, unlike the parameters for Log mel spectrogram extraction, it is difficult to derive the hyperparameter values directly from literature. Additionally, only some hyperparameters were included in the grid-search because for these particular parameters it is hard to conclude the best possible value through a manual search. The hyperparameters and their respective options for values in the grid-search process are as follows:

1. Number of CNN blocks (L_c): $\{1,2,3,4\}$.

2. The kernel and stride of the frequency max-pooling layers: $\{$(4), (8), (2,2), (4,2), (7,5,2), (5,3,1), (2,2,2,2), (4,4,4,2)$\}$.
   Here, the options are chosen such that the max-pooling reduces the number of frequency bands in the input spectrogram to one of the following number of bands (F'): (1,8,16,32). For instance, the max-pooling configuration of (7,5,2) reduces the original $N = 128$ bands to 1 band in three stages: 128 bands → 18 bands → 3 bands → 1 band.

3. Dropout value for CNN blocks and GRU: $\{0.1, 0.25, 0.5\}$. For GRU layers more than 1, the dropout for these layers is set to be equal to that of CNN blocks. For single layer GRU it is not possible to set the dropout to a value other than 0. This is because Pytorch implementation does not allow dropout value to be set for single GRU layers.

4. Number of GRU layers (L_r): $\{1, 2, 3\}$.

For each combination of the above hyperparameters in the grid-search, the performance of the acoustic model is estimated by the cross-validation approach (with $k = 10$ folds) described in Section 3.4.3. The details of other hyperparameters whose values are decided by a manual search are enlisted below:

- Number of feature maps (M): 16
- Temporal length of extracted samples (δT_s): $1s$
- Filter size in the convolutional layers: $(5, 5)$
- Learning rate: 1×10^{-3}
- Optimizer: Adam[82]
- Batch Size: 8
- Total number of training epochs: 150

Additionally, in the preliminary experiments it was found that if cross-validation experiments were repeated with the same configuration for all the hyperparameters, the average balanced accuracy would have considerable variation across the trials. This could be due to a combination of factors such as generally low number of samples in the data splits and the different initialization of model weights. Hence, to have a fair comparison during the grid search, the order of the samples in a batch and the initialization of the model weights were fixed with a specific random seed.

Tuning of the hyperparameters is performed separately on both `static` and `dynamic` subsets. The performance of all the combinations of hyperparameters in the grid-search are visualized with the help of a parallel coordinates plot (see Figures 4.4 and 4.5). Here, the parameters with their options are listed on the different axes of the plot and the lines joining the different parameter options depict the combinations that are evaluated in the grid-search. Additionally, the color of these lines indicate the average balanced accuracy in the cross-validation setting. Tables 4.3 and 4.4 present the top 5 best performing hyperparameter combinations on both the subsets.

From Figure 4.4, it can be seen that the color of most of the lines are towards the higher end of the accuracy variation spectrum. This indicates that the different hyperparameter combinations tried out in the grid-search, report metrics that are high and also close to each other. However, the same does not hold for the results from the `dynamic` subset (see Figure 4.5) as a trend of a few combinations outperforming others can be

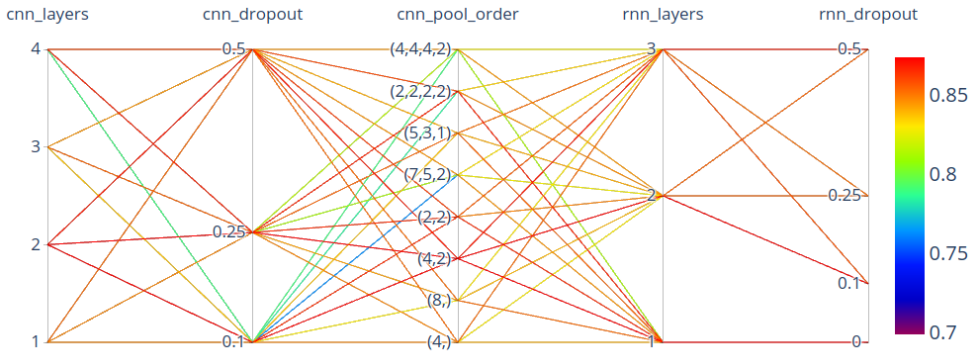


Figure 4.4: The results of all the combinations of the hyperparameters on the `static` subset. The color bar on the right depicts the color assigned to a particular balanced accuracy value. Most of the combinations here report similar performance as evidenced by the shade of red colors on most of the lines.

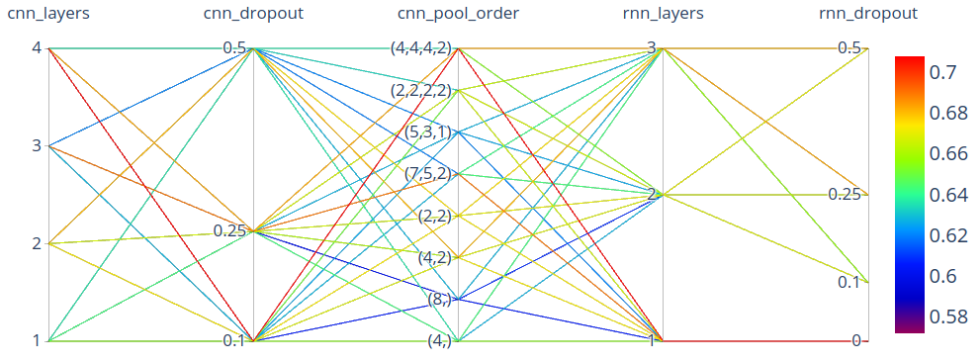


Figure 4.5: The results of all the combinations of the hyperparameters on the `dynamic` subset. Only few combinations here report metrics on higher end of the performance spectrum as evidenced by lack of red lines.

Table 4.3: Best performing hyperparameter combinations on the `dynamic` subset.

Rank	CNN Blocks	CNN Dropout	CNN Max Pooling Layers	GRU Layers	GRU Dropout	Balanced Accuracy
#1	4	0.1	(4, 4, 4, 2)	1	0	0.707 ± 0.05
#2	3	0.25	(7, 5, 2)	1	0	0.689 ± 0.07
#3	4	0.25	(4, 4, 4, 2)	3	0.25	0.683 ± 0.08
#4	4	0.25	(4, 4, 4, 2)	1	0	0.682 ± 0.09
#5	2	0.5	(4, 2)	3	0.5	0.680 ± 0.08

seen. A single acoustic model that performs well on both ego-motion modes of the IV would be easier to use as there would be no need to switch between different models dur-

Table 4.4: Best performing hyperparameter combinations on the `static` subset.

Rank	CNN Blocks	CNN Dropout	CNN Max Pooling Layers	GRU Layers	GRU Dropout	Balanced Accuracy
#1	2	0.25	(4, 2)	1	0	0.874 ± 0.04
#2	2	0.1	(4, 2)	2	0.1	0.873 ± 0.05
#3	2	0.5	(4, 2)	3	0.5	0.869 ± 0.03
#4	2	0.5	(4, 2)	1	0	0.867 ± 0.05
#5	2	0.25	(2, 2, 2, 2)	1	0	0.866 ± 0.04

ing real-time operation. Selecting the best performing combination on either `static` or `dynamic` subset would not be sufficient in this case because they do not perform well on the other subset. To elaborate, the best performing combination on `dynamic` subset ranks at #62 on the `static` subset (balanced accuracy of 0.789 ± 0.05), whereas the best one on `static` subset ranks at #15 on the `dynamic` subset (balanced accuracy of 0.661 ± 0.06). Hence, it was decided that the ranks of the all the combinations of both subsets will be summed and the one with the least sum would be carried forward for future experiments (see Table 4.5).

Table 4.5: Selected combination of hyperparameters and its performance on the both data subsets. Note that this combination ranks at #3 and #5 on the `static` and `driving` subset respectively.

Item	Value
CNN Blocks	2
CNN Dropout	0.5
CNN Max Pooling Layers	(4, 2)
GRU Layers	3
GRU Dropout	0.5
Balanced Accuracy- `static`	0.869 ± 0.03
Balanced Accuracy - `dynamic`	0.680 ± 0.08

4.3. PREDICTIONS ACROSS MULTIPLE TIME HORIZONS

The performance of the acoustic model is visualized across the entire length of the recordings. This is to give an idea of how the predictions of the acoustic model vary in different ego-motion modes of the IV and in the presence/absence of an approaching vehicle.

This experiment is again carried out in a cross-validation setting, wherein for each fold, the recordings from which test samples were picked out are identified. A window (of length δT_s) is initialized at the start of each recording and is slid towards its end in the steps of $0.1s$. The samples extracted from each window are then fed into the acoustic model to obtain the class-probabilities (see Equation 3.4), which indicate the gen-

eral confidence of the model for detecting the presence of a vehicle behind the corner. Here, for each class-probability vector, we take the median to represent the overall confidence of the model for a particular sample and therefore we can visualize the variations of prediction performance through the recordings. The time-stamp associated with this confidence value would be that at the end of each window. This visualization can be considered as analogous to the real-time operation, wherein the IV would have stored the last δT_s of the audio sample and used it for prediction. The confidence values across all recordings are first divided according to the sub-classes `left, right` and `none` and then the average confidence value and its standard deviation is plotted. For fair comparison within each sub-class, the plots are made with the recordings aligned around t_0 or τ_0. It should be noted that these predictions arise from $k = 10$ different acoustic models and variation of average confidence over every recording in the data subset is reported in the plots.

The variation of confidence of the acoustic model with time on both data subsets is depicted by the Figures 4.6 and 4.7. The dashed horizontal line in the plots indicate the decision threshold ($C = 0.5$) used to threshold the probability vector obtain the classes *i.e.* `car, no car`. On the `static` subset, the acoustic model is able to predict an approaching vehicle at an average of $1.4s$ before t_0. In the `driving` subset, the acoustic model on average is unable to distinguish between the presence and absence of the car. The average confidence quickly ramps up if there is a vehicle approaching the IV and within the next $0.5s$ after τ_0, the average confidence is clearly above the decision threshold. This could be an indication that the model is able to predict an approaching vehicle just before it is in view and not well in advance as seen in the `static` case.

Another observation is that the acoustic model seems to be slightly sensitive to the sounds due to the ego-motion of the IV. To elaborate, the average confidence for the `none` sub-class recordings gradually increases as the IV approaches τ_0 and then reduces quickly. This is similar to the motion profile of the IV during the data collection stage, where the IV accelerates from standstill towards the intersection and then stops when it reaches there. However, the model is not overly confident in associating ego-motion sounds to that of an approaching vehicle.

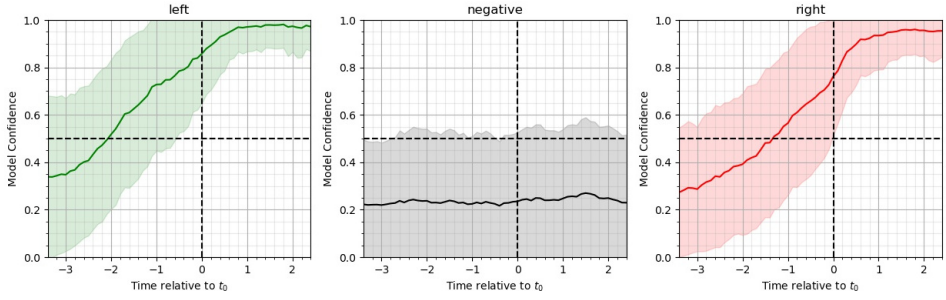


Figure 4.6: Variation of average confidence of the acoustic model with time across `static` recordings.

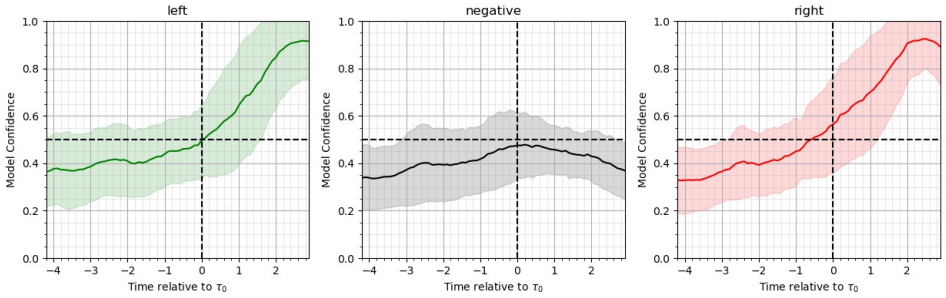


Figure 4.7: Variation of average confidence of the acoustic model with time across `dynamic` recordings

4.4. QUALITY OF ACOUSTIC CUES

In the `static` recordings, the acoustic model was able to detect an approaching car well before it was in view, but for the model trained and tested on `dynamic` data subset this was not the case. This can be attributed to the fact that the `left` and `right` samples, extracted from `dynamic` recordings, might not provide clear acoustic cues which point toward an approaching vehicle, as the ego-noise of the IV might mask the sounds of an approaching vehicle. To investigate this, the acoustic model is trained in two different settings. First, the samples belonging to `left` and `right` sub-class is removed from the train and validation split during the cross-validation experiment. The `front` samples have strong cues that point toward a vehicle passing by, so the scope of this experimental setting is to check if training with just `front` samples can help the model identify fainter cues found in `left/right` samples. Second, the samples from `front` is removed from train and validation split and `car` class therefore has only `left` and `right` samples. Relying only on the samples where the approaching vehicle is occluded, gives an idea as to how well the acoustic model learns from them.

Figures 4.8 and 4.9 depict the average confidence variation across different time horizons for the acoustic model trained only on `front` and `none` sub-class samples. It can observed from these plots that for both data subsets, the acoustic model cannot predict an approaching vehicle before it is in view. For both ego-motion modes, the model is now more confident about the absence of any approaching vehicles, as in the `none` sub-class plots, the average confidence of the model is well below the decision threshold. As described in Section 4.3, this was not the case especially with `dynamic` recordings, where until τ_0 the confidence of the model was very close to the decision threshold. The strong difference between the `front` and the `none` class samples in both ego-motion modes could be the key to this improvement. Further, inability of the acoustic model to predict the approaching vehicle before t_0 in the `static` subset highlights the importance of the `left` and `right` samples.

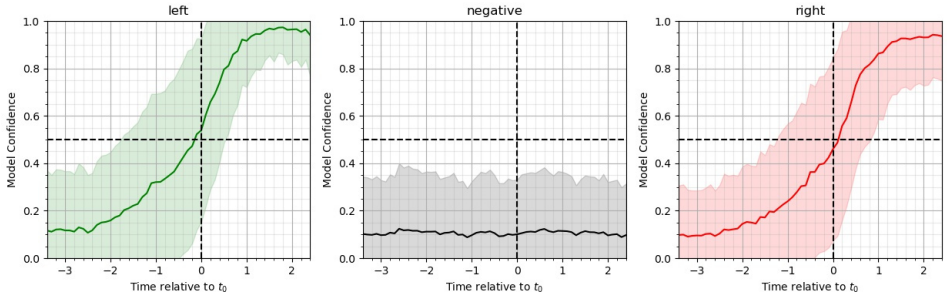


Figure 4.8: Variation of average confidence of the acoustic model trained only on `front` and `none` samples of the `static` subset. Balanced accuracy of 0.827 ± 0.03 is achieved during this experiment.

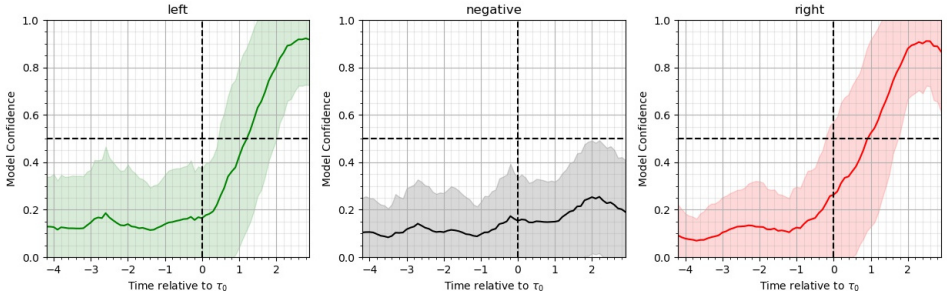


Figure 4.9: Variation of average confidence of the acoustic model trained only on `front` and `none` samples of the `dynamic` subset. Balanced accuracy of 0.743 ± 0.04 is achieved during this experiment.

Looking at the Figures 4.10 and 4.11, it can be said that training with only `left` and `right` samples works only in the case `static` recordings but not on `dynamic` ones. Here, the performance of the acoustic model on the `static` subset is similar to that when all the sub-class samples are included for training. The drop in average balanced accuracy is around ~ 1% and the prediction before t_0 drops by just $0.1s$. This indicates that the `front` samples are not as critical for the acoustic model to carry out the prediction task in case of `static` subset. Moreover, this assures that the acoustic cues are very prominent in the `left` and `right` samples of the `static` subset. This result also indicates that the acoustic features pointing towards an approaching vehicle from behind the corner (`left/right`) is quite different from the ones indicating the presence of the vehicle in line-of-sight (`front`). In the `dynamic` case, the average confidence is almost equal to the decision threshold at all time stamps. This means that the acoustic model does not have any predictive capability and therefore would make random predictions. This result indicates that the `left` and `right` samples of the `dynamic` subset are very similar to the `none` samples. This result indicates that there is almost no additional information in the `left` and `right` samples as compared to the `none` samples.

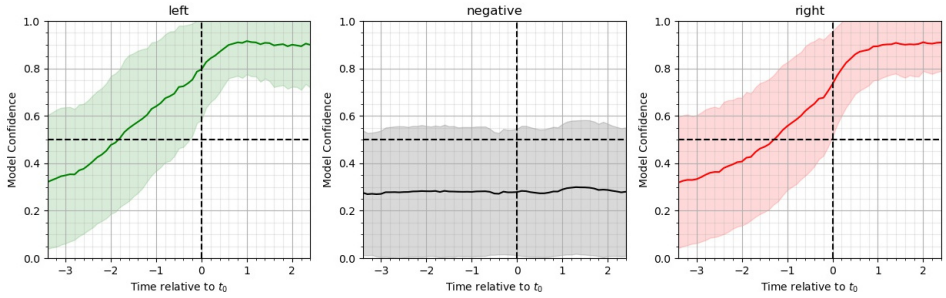


Figure 4.10: Variation of average confidence of the acoustic model trained on `left`, `right` and `none` samples of the `static` subset. Balanced accuracy of 0.855 ± 0.05 is achieved during this experiment.

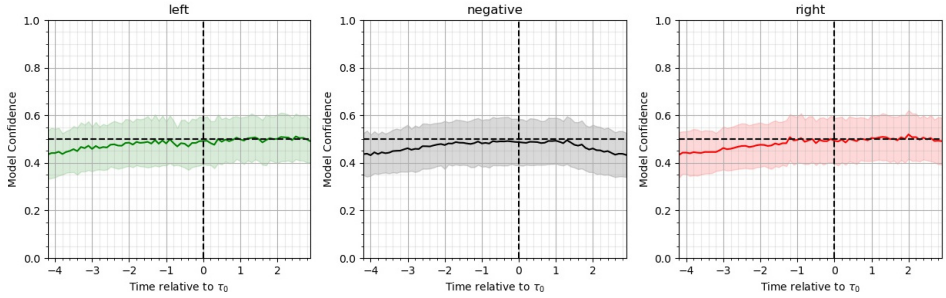


Figure 4.11: Variation of average confidence of the acoustic model trained on `left`, `right` and `none` samples of the `dynamic` subset. Balanced accuracy of 0.528 ± 0.03 is achieved during this experiment.

4.5. GENERALIZATION ON UNSEEN SAMPLES

The goal of this experiment is to investigate if the acoustic model can generalize on unseen samples. This is an important requirement for the acoustic model because it is impossible to record audio samples from every possible scenario that an IV might find itself in. Based on the data that is collected for this work, the generalization of the acoustic model can be investigated in two different ways: ① Location-based ② Ego-motion based. The location-based generalization implies that the samples from a single location (for e.g. SA1) are not simultaneously present in both train and the test split. Similarly, in the ego-motion based generalization, the samples are divided into the train and test split based on the ego-motion of the IV during their recordings.

Unlike the previous experiments, cross-validation is not performed to evaluate the performance of the acoustic model. This is because the samples considered for experimentation will be present in train, validation and test split during cross-validation. As a result, the ability of the acoustic model to generalize on unseen samples cannot be investigated. Hence, it is decided that subset(s) available for training the acoustic model will be split into two sets (for training and validation) according to three different random seeds. For the three acoustic models that are trained here, the average balanced

accuracy on the test set is reported. Additionally, the summed confusion matrix is also reported to give a better clarity on the predictions of the classifier.

Tables 4.6, 4.7 report the metrics for all the train and test combinations considered in the location-based generalization for both the data subsets. For the `static` subsets, it can be observed that the models trained and tested on *Type B* locations perform slightly better than those on *Type A* locations. The good test performance at location B1 can be attributed to the quietness of its surroundings. However, this cannot explain the larger drop in performance amongst the *Type A* combinations, as both of the locations here did have quiet surroundings. Additional investigation was performed to test the impact of the quiet surroundings on the performance of the acoustic model. From Table 4.8, it can be seen that *Type A* subsets individually report high metrics (see rows SA1 and SA2). However, when included together (see row SA), there is a considerable drop in the performance. The reasons behind this drop could also be the reason why the acoustic model struggles in generalize across SA1/SA2. However, the exact nature of this reason is still unclear. Further, again from Table 4.8, it can be seen that subsets that include samples from SB1 i.e. SB1, SB13 and SB12, all report high performance metrics. This could be because the test samples here would have included the quieter samples from SB1, which is also the reason why generalization on unseen SB1 samples reported high metrics.

On the `dynamic` subsets, the acoustic model is unable to generalize, especially in *Type B* location. Location specific trends as observed in this experiment for `static` subset cannot be observed here. This can be attributed to the fact that the ego-motion noises will always dominate the sounds from the surroundings and the quietness of the surroundings would not influence the performance as much as it did in the `static` subset. The generalization across DA1/DA2 both report balanced accuracies that are relatively higher than the rows (see Table 4.8). This performance is also close to the balanced accuracy of 0.68 that is obtained when all the `dynamic` samples are presented for training and evaluation. Further, from Table 4.8, it can be seen that combining DA1 and DA2 subsets for cross-validation (see row DA) significantly improves the balanced accuracy. This could mean that the samples in DA1 and DA2 are similar in nature. Hence, the generalization across the two subsets is relatively better and the cross-validation performance of the combined subset is superior.

Table 4.9 reports the results of different train/test combinations considered for the investigation of ego-motion based generalization of the acoustic model. It can be seen that the performance of the acoustic model is close to one-dimensional prediction in most cases. When the train subset is of `static` type, the acoustic model on the `dynamic` type subsets predominantly predicts presence of an approaching vehicle *i.e.* `car`, but when the train subset is `dynamic`, the trend is reversed. This could be an indication that volume of the audio samples play a role in the training of the acoustic model. To elaborate, the model trained with `dynamic` subset is exposed higher levels of noise due to ego-motion and absence of this noise in `static` subset causes the model to predict `no car` frequently. When the model is trained on the `static` subset, the loud ego-motion

Table 4.6: Generalization of the acoustic model across different locations amongst the `static` samples. For the *Type B* locations, as there are three locations available, the subsets are merged together for training the acoustic model.

Test	Train	Balanced Accuracy	Confusion Matrix
SA1	SA2	0.747 ± 0.02	$\begin{pmatrix} 97 & 83 \\ 4 & 86 \end{pmatrix}$
SA2	SA1	0.629 ± 0.02	$\begin{pmatrix} 246 & 0 \\ 109 & 38 \end{pmatrix}$
SA	SB	0.826 ± 0.02	$\begin{pmatrix} 393 & 33 \\ 64 & 173 \end{pmatrix}$
SB1	SB23	0.871 ± 0.01	$\begin{pmatrix} 244 & 2 \\ 24 & 72 \end{pmatrix}$
SB2	SB13	0.818 ± 0.03	$\begin{pmatrix} 315 & 15 \\ 41 & 88 \end{pmatrix}$
SB3	SB12	0.744 ± 0.02	$\begin{pmatrix} 248 & 22 \\ 58 & 77 \end{pmatrix}$
SB	SA	0.760 ± 0.05	$\begin{pmatrix} 795 & 51 \\ 151 & 209 \end{pmatrix}$

Table 4.7: Generalization of the acoustic model across different locations amongst the `static` samples.

Test	Train	Balanced Accuracy	Confusion Matrix
DA1	DA2	0.643 ± 0.06	$\begin{pmatrix} 213 & 15 \\ 72 & 39 \end{pmatrix}$
DA2	DA1	0.621 ± 0.09	$\begin{pmatrix} 61 & 29 \\ 17 & 22 \end{pmatrix}$
DA	DB	0.559 ± 0.07	$\begin{pmatrix} 309 & 9 \\ 128 & 22 \end{pmatrix}$
DB1	DB23	0.509 ± 0.00	$\begin{pmatrix} 202 & 8 \\ 102 & 6 \end{pmatrix}$
DB2	DB13	0.557 ± 0.04	$\begin{pmatrix} 91 & 41 \\ 38 & 28 \end{pmatrix}$
DB3	DB12	0.500 ± 0.00	$\begin{pmatrix} 228 & 0 \\ 108 & 0 \end{pmatrix}$
DB	DA	0.593 ± 0.03	$\begin{pmatrix} 337 & 233 \\ 114 & 168 \end{pmatrix}$

noise is probably misinterpreted as an approaching car and consequently the performance suffers. Both of these inferences are based on the confusion matrices reported for both the train/test combinations in Table 4.9. Moreover, the results here also reveal that the strong acoustic cues learnt by the acoustic model trained on `static` subset (see section 4.4) are not useful in predicting the approaching vehicle in the `dynamic` case.

Table 4.8: Cross-validation ($k = 10$) results on individual location based subsets.

Subset	Balanced Accuracy	Subset	Balanced Accuracy
SA1	0.900 ± 0.08	DA1	0.658 ± 0.18
SA2	0.866 ± 0.10	DA2	0.638 ± 0.23
SA	0.829 ± 0.08	DA	0.756 ± 0.13
SB1	0.912 ± 0.12	DB1	0.548 ± 0.14
SB2	0.808 ± 0.10	DB2	0.517 ± 0.08
SB3	0.765 ± 0.12	DB3	0.523 ± 0.11
SB23	0.803 ± 0.08	DB23	0.517 ± 0.08
SB13	0.904 ± 0.03	DB13	0.636 ± 0.09
SB12	0.908 ± 0.07	DB12	0.504 ± 0.03
SB	0.836 ± 0.07	DB	0.572 ± 0.08

Table 4.9: Ego-motion based generalization of the acoustic model. The subsets chosen for training mostly belong to `static` subset as it is easier to collect and hence can be assumed to be available before enough data is collected for `dynamic` situations for a particular location.

Test	Train	Balanced Accuracy	Confusion Matrix
`dynamic`	`static`	0.502 ± 0.00	$\begin{pmatrix} 887 & 1 \\ 430 & 2 \end{pmatrix}$
`static`	`dynamic`	0.642 ± 0.01	$\begin{pmatrix} 399 & 873 \\ 18 & 579 \end{pmatrix}$
DA	SA	0.539 ± 0.02	$\begin{pmatrix} 262 & 56 \\ 112 & 38 \end{pmatrix}$
DA1	SA1	0.516 ± 0.01	$\begin{pmatrix} 221 & 7 \\ 104 & 7 \end{pmatrix}$
DA2	SA2	0.620 ± 0.08	$\begin{pmatrix} 70 & 20 \\ 21 & 18 \end{pmatrix}$
DB	SB	0.500 ± 0.00	$\begin{pmatrix} 570 & 0 \\ 282 & 0 \end{pmatrix}$
DB1	SB1	0.500 ± 0.00	$\begin{pmatrix} 210 & 0 \\ 108 & 0 \end{pmatrix}$
DB2	SB2	0.496 ± 0.01	$\begin{pmatrix} 121 & 11 \\ 61 & 5 \end{pmatrix}$
DB3	SB3	0.500 ± 0.00	$\begin{pmatrix} 228 & 0 \\ 108 & 0 \end{pmatrix}$
DB12	SB12	0.500 ± 0.00	$\begin{pmatrix} 342 & 0 \\ 174 & 0 \end{pmatrix}$
DB13	SB13	0.500 ± 0.00	$\begin{pmatrix} 438 & 0 \\ 216 & 0 \end{pmatrix}$
DB23	SB23	0.500 ± 0.00	$\begin{pmatrix} 360 & 0 \\ 174 & 0 \end{pmatrix}$

4.6. DOMAIN ADAPTATION USING ACAL

In this section, we investigate if the chosen domain adaptation technique (ACAL)[71], can be used to help the acoustic model trained on `static` subset generalize on the `dynamic` subset. However, an open source implementation of the ACAL technique was unavailable. Hence, to verify if the reproduction of this technique was correct, one of the experiments performed by the authors [71] is carried out in Section 4.6.1. Subsequently, the results of the adaptation of the acoustic model is discussed in Section 4.6.2.

4.6.1. REPRODUCING ACAL ON MNIST-SVHN DATASETS

The experiment to be reproduced is the model ablation study, where the authors [71] study the contribution of each component of the model. For clarity, here we reproduce the results reported on their final chosen method - ACAL and the initial CycleGAN model from which this technique is modified from. The other ablations considered by the authors [71] have been ignored as they do not contribute towards verifying the reproduction of ACAL. Visual domain adaptation is performed in this experiment, where the source domain is the train split of Street View House Numbers (SVHN) dataset [83] and the target domain is a small part of the MNIST dataset [84]. Only 10 samples per class (i.e. total 100) from the train split of the MNIST dataset is used. This subset is denoted as MNIST-10 and contains only 0.17% of the full MNIST training data. The SVHN dataset used here consists of 73257 real-world images of digits.

The authors in [71] have not explicitly reported some of the details in their experiment, such as the architectures of generators ($G_{S\rightarrow T}$, $G_{T\rightarrow S}$), discriminators(D_T, D_S) and the value for the different hyperparameters used. Hence an exact reproduction of the results has been difficult. The generator and discriminator networks used by the authors in [72] for experimenting with cycleGAN are used in this experiment. The generator used here is the variant with the six residual blocks[85] as introduced by the authors here in [72] and for the discriminator, the PatchGAN[86] network is used. For the task-specific model (M_S, M_T), the authors in [71] used a modified version of LeNet[84] with two convolutional layers with 20 and 50 channels, followed by a dropout layer and two fully connected layers of 50 an 10 respectively. Other missing details have been finalized for the experiment here after manually searching for the best working combination of parameters and Table 4.10 reports the value for the same.

The results of this experiment is reported in Table 4.11. The low difference between the implementations of authors of ACAL[71] and this work suggests that reproduction is successful.

Table 4.10: Hyperparameters used for reproducing the ACAL technique for adapting from SVHN to MNIST dataset.

Item	Value
Batch Size	32
Epochs	5
# of discriminator layers	2
# of filters – last convolution layer of generator	16
# of filters – first convolution layer of discriminator	16
Relative importance weight – Cycle $S \rightarrow T \rightarrow S$ (λ_{STS})	1.0
Relative importance weight – Cycle $T \rightarrow S \rightarrow T$ (λ_{TST})	10.0
Dropout – Task specific model (M_S, M_T)	0.25
Optimizer & Learning Rate (GANs)	Adam[82] & 3×10^{-4}
Optimizer & Learning Rate (M_S, M_T)	SGD[87] & 1×10^{-2}

Table 4.11: Reproduction of the method Augmented Cyclic Adversarial Learning (ACAL). The testing performance is calculated on the full MNIST test set. The metrics reported is the standard Accuracy. Experiments were performed 4 times with random sampling to generate MNIST-10 dataset.

Domain Adaptation Model	Test Accuracy		
	Paper [71]	Reproduction	Difference
No Adaptation (trained on SVHN)	0.711	0.644	−0.067
Target Model (trained on MNIST-10)	0.792 ± 0.04	0.778 ± 0.02	−0.014
SVHN + MNIST-10	0.856 ± 0.01	0.756 ± 0.01	−0.1
CycleGAN	0.455 ± 0.01	0.468 ± 0.05	+0.013
ACAL	0.939 ± 0.00	0.894 ± 0.03	−0.045

4.6.2. TRAINING CRNN WITH ACAL ON ACOUSTIC IV DATASET

This section aims at addressing the domain adaptation on the acoustic IV dataset. In the reproduction experiment (Section 4.6.1), the ACAL technique used only a very small subset of samples from the target domain and large number of source domain samples to outperform the un-adapted model by a large margin. However, the acoustic dataset used here does not have a large number of samples in both the domains. Thus, using a small subset of samples of target domain (i.e. `dynamic`) might make learning impossible. Using only one large subset of the target domain would not be enough, as comparison with the cross-validation experiments carried out in the previous section would then be difficult to perform. Hence, the target domain samples are divided into $k = 10$ folds. The model (M_T) that has the lowest value of loss function on the validation fold is evaluated on the test fold. This best model selection procedure is the same as that followed in all of the previous experiments where there is no adaptation involved.

As opposed to the visual domain adaptation experiment, the authors [71] have provided details of the generators and discriminators used in their adaptation experiment

for audio domain experiment. The generators used are the same as the ones used in [71] which are based on U-net architecture with 4 layers of convolution and corresponding deconvolution layers. The discriminators are the same as the ones used in the visual domain experiment with the one modification to the output. The network is made to predict a single scalar instead of a matrix of real/fake probability. This modification was recommended by the authors in [88] and it greatly increased the stability of the training process. The tuning of the hyperparameters is relatively difficult compared to the visual domain adaptation. This is because the log mel spectrogram visualizations of source and target domain samples were similar to each other, unlike the numbers in MNIST and SVHN datasets. Table 4.12 presents the details of the hyperparameters used for the training.

Table 4.12: Hyperparameters used for the audio domain adaptation. The values are derived based on an exhaustive manual search. Task specific models use the same hyperparameters as the ones finalised in Table 4.5

Item	Value
Batch Size	8
Epochs	100 (last 50 epochs with decaying learning rate)
# of discriminator layers	3
# of filters – last convolution layer of generator	32
# of filters – first convolution layer of discriminator	16
Relative importance weight – Cycle $S \rightarrow T \rightarrow S$ (λ_{STS})	1.0
Relative importance weight – Cycle $T \rightarrow S \rightarrow T$ (λ_{TST})	1.0
Optimizer & Learning Rate (GANs)	Adam[82] & 1×10^{-3}

Table 4.13 reports the fold-wise and the mean performance for experiments with and without adaptation. From this table, it can be seen that 5 of the folds show improvement over un-adapted counterparts. But the improvement in overall performance is not significant as the rest of the folds depict a drop in performance after adaptation. If the confusion matrices are compared across the same fold of both experiments, it can be seen that the elements in the matrices are similarly distributed across the two matrices. For example, in Fold 1, the maximum difference between the corresponding elements of the two matrices is 1 and a similar trend can be seen for folds 2, 4, 5, 6, 9, 10. This indicates that the domain adaptation does not drastically change the predictive nature of the acoustic model. The reason for this could be the low quality of generated target domain samples which does not provide any new information for the acoustic model to learn from. However, these generated samples should similar to the real ones. Otherwise, the decision boundary would have been corrupted badly and consequently the performance would have dropped significantly during adaptation process.

To further investigate the quality of generated samples, the audio files were con-

structed from the spectrogram representation using the Griffin-Lim algorithm [89]. Upon hearing the generated samples, it could be noticed that the generator did capture the dominance of the low pitch background noise present in the `dynamic` samples due to ego-noise. Further, the generated `front` class samples were relatively louder than the other sub-class and this characteristic is also found in the real `dynamic` samples. However, it did not capture the high pitch engine noise and also contained some artifacts. Like the real `dynamic` samples, it was difficult to distinguish between `left`/`right` and the `none` samples.

Additional data in the source domain could have helped in training of the generators and discriminators in producing more convincing fake samples. The experiments with ACAL for audio domain adaptation in [71] are performed with much larger datasets (with total length of 5.4 hours [90]) than the one collected for this work (length of extracted `static` samples is 0.29 hours). Further, in the visual domain adaptation experiment (Section 4.6.1), the generated samples could be visualized and their quality can be accurately judged by a quick visual examination. This makes the manual hyperparameter tuning process easier unlike in the audio domain adaptation process where it is difficult to judge the quality of fake samples by looking at their spectrogram representation. The Griffin-Lim algorithm [89] is slow in reconstructing the audio samples and cannot be used in the tuning process effectively. Hence, it is possible that different hyperparameter values for the items listed in Table 4.12 could improve the results of the adaptation process.

Table 4.13: Comparison of the performance of the domain adaptation with that of no adaptation for acoustic IV dataset. The two learning settings in 2^{nd} and 5^{th} rows of Table 4.11 are respectively analogous to the No Adaptation and ACAL Adaptation experiments reported in this table. However, unlike in Table 4.11, similar gain in performance was not observed here. Highlighted metrics depict improvement in performance over un-adapted models.

# Fold	No Adaptation		ACAL Adaptation	
	Balanced Accuracy	Confusion Matrix	Balanced Accuracy	Confusion Matrix
Fold 1	0.800	$\begin{pmatrix}24 & 6\\3 & 12\end{pmatrix}$	0.750	$\begin{pmatrix}23 & 7\\4 & 11\end{pmatrix}$
Fold 2	0.750	$\begin{pmatrix}19 & 11\\2 & 13\end{pmatrix}$	0.700	$\begin{pmatrix}20 & 10\\4 & 11\end{pmatrix}$
Fold 3	0.598	$\begin{pmatrix}26 & 2\\11 & 4\end{pmatrix}$	**0.762**	$\begin{pmatrix}24 & 4\\5 & 10\end{pmatrix}$
Fold 4	0.493	$\begin{pmatrix}22 & 6\\12 & 3\end{pmatrix}$	**0.526**	$\begin{pmatrix}22 & 6\\11 & 4\end{pmatrix}$
Fold 5	0.664	$\begin{pmatrix}27 & 3\\8 & 6\end{pmatrix}$	**0.684**	$\begin{pmatrix}26 & 4\\7 & 7\end{pmatrix}$
Fold 6	0.659	$\begin{pmatrix}20 & 10\\5 & 9\end{pmatrix}$	0.657	$\begin{pmatrix}18 & 12\\4 & 10\end{pmatrix}$
Fold 7	0.743	$\begin{pmatrix}21 & 9\\3 & 11\end{pmatrix}$	**0.829**	$\begin{pmatrix}24 & 6\\2 & 12\end{pmatrix}$
Fold 8	0.688	$\begin{pmatrix}22 & 8\\5 & 9\end{pmatrix}$	0.593	$\begin{pmatrix}27 & 3\\10 & 4\end{pmatrix}$
Fold 9	0.709	$\begin{pmatrix}19 & 11\\3 & 11\end{pmatrix}$	**0.795**	$\begin{pmatrix}22 & 8\\2 & 12\end{pmatrix}$
Fold 10	0.702	$\begin{pmatrix}25 & 5\\6 & 8\end{pmatrix}$	0.667	$\begin{pmatrix}25 & 5\\7 & 7\end{pmatrix}$
Overall	0.680 ± 0.08		$\mathbf{0.696 \pm 0.08}$	

5

Conclusion

This thesis explored a single microphone setup as the acoustic perception system to predict if there is a vehicle approaching the road intersection from behind the blind corner. Methods that can identify the nature of sound sources, i.e. classification based approaches, were looked at and those relevant to the IV application were categorized under Sound Event Detection. A review of the literature revealed that the feature commonly used for audio representation stage was the Log-Mel Spectrogram and the Convolutional Recurrent Neural Network architecture was mostly preferred as the acoustic model. Furthermore, there was no dataset in the literature that could have been used to simulate the scenario of two vehicles approaching an intersection unbeknownst to each other and hence a new dataset had to be recorded for the evaluation. It was also found that the metrics generally used for SED were slightly biased towards the majority class in a binary classification setting (see Section 3.4.1) and the evaluation was performed with the metric - Balanced Accuracy. This metric can be carried over even if more classes are included in the future, however, if the temporal onset and offset of sound events is to be evaluated then a switch must be made to one of the traditional SED evaluation techniques as briefly described in Section 2.5. Additionally, domain adaptation approaches in SED were looked into to reduce the data collection effort by having to collect only a few samples from the target domain i.e. when the ego-vehicle is moving. The existing techniques in SED are based on unsupervised domain adaptation which would require a lot of data from the target domain. This would not be suitable for the application addressed in this work as it would require lot of effort to just collect data. Hence, a method that can learn from low number of target domain samples was selected for domain adaptation. Experiments were performed to explore the feasibility of the chosen system. Section 5.1 presents the results and conclusions that correspond to each of the research question posed at the beginning of this work. Section 5.2 presents some of the future directions to take with respect to this work and as well as towards implementing an acoustic perception system in IVs.

5.1. ADDRESSING RESEARCH QUESTIONS

1. **How well can a single microphone setup on an IV predict an approaching vehicle at a blind intersection?**

   The setup considered in this work was evaluated using a novel dataset that can be further divided into two subsets, one where the ego-vehicle is not moving (`static`) and the other where it is moving (`dynamic`). Experiments were mostly separately performed on the `static` and `dynamic` subsets. On the `static` subset, it was found that CRNN acoustic model achieves an balanced accuracy of 86.9%. Further, the performance of the acoustic model is visualized across the entire length of the test recordings. This experiment simulates the situation wherein the acoustic model is presumed to be deployed on the IV and its result gives an indication on how the model might perform in real-time scenarios. It is observed that the average confidence of the acoustic model crosses the decision threshold at $1.4s$ before the approaching vehicle is in line-of-sight. This is an ample amount of time for the ego-vehicle process the detection and make an appropriate decision moving forward.

   On the `dynamic` subset, the performance was not very impressive as a balanced accuracy of only 68% was achieved. During the visualizations of predictions across the `dynamic` test recordings, it was found that the average confidence moved above decision threshold right after τ_0 for positive cases. It should be noted that there is around 7-10 meters to the intersection from ego-vehicle perspective at τ_0 and within the next $0.5s$ (i.e $\tau_0 + 0.5s$) the average confidence is clearly above the decision threshold for positive cases. Hence, it could be said this is an indication of model being able to predict an approaching vehicle before it is in the field of view in `dynamic` scenarios. However, from a traffic safety perspective, this is not a robust indication and should be improved upon if this system is to be deployed in real-time.

2. **How well does the acoustic model generalize across different locations or different ego-motion modes?**

   Currently, the generalization capability of the acoustic model is not up to the mark across both the categories, i.e. location and ego-motion modes. The models trained on various `static` subsets perform very poorly when tested on the `dynamic` subsets. The prediction in this case by the model is almost always `car` and this could be because of the ego-vehicle noise that is present in the `dynamic` samples. Reverse of this trend can be observed when the model is trained only on `dynamic` subsets predict mostly `no car` when tested on `static` subsets. This trend can be observed by looking at the confusion matrices in Table 4.9. Only in few cases of location based generalization, the system generalizes quite well (e.g SB23 → SB1, see Table 4.6). However, the performance is entirely random in other cases and there is a slight bias towards predicting either one of the classes depending on the combination chosen. This inability to generalize well across different location implies that the model might be carrying out classification based on cues present in background noises specific to each location rather than the approaching vehicle.

3. **Is it possible to use techniques from domain adaptation by using static data to improve the performance on driving data which is limited in number?**

   The performance of the adapted model is more or less the same as the unadapted version. The generators were able to produce fake spectrogram samples from the both `static` and `dynamic` domains. However, the quality of those samples is clearly not enough to boost the performance. Before dismissing the selected domain adaptation, ACAL, as unsuitable for future work, it should be noted that the amount of data in the source domain was very less as compared to some of the experiments carried out the authors here [71].

5.2. FUTURE WORK

This section will describe future directions researchers can take to further explore into the field of acoustic perception for IVs. The first 3 points discussed below provide directions specific to this thesis and the fourth point is an opinion of the author on an alternative approach for the acoustic perception system that could be explored.

1. The dataset used in this thesis has mostly the same vehicle as the approaching the blind corner. While different vehicles have varied engine sounds, at higher speeds (> 20 km/h) the noise due to road-tyre contact is dominant [91]. This would mean that the performance drop due to unknown vehicles present in test data should be at minimum as the vehicles are currently detected by their tyre noises. If more passing by vehicles are included while recording `static` data, this could be analysed in further detail.

2. Other audio domain adaptation experiments using ACAL [71] had used higher amounts of source domain data. Hence, another motivation to collect more data would be to check if having more source domain (`static`) data would be helpful in successfully augmenting the performance on the `dynamic` samples as attempted by the domain adaptation experiment in Section 4.6.

3. In the audio domain adaptation experiment (Section 4.6), during the training of the networks, it was found that it was not intuitive to judge the quality of fake samples generated through the epochs and as a result the tuning of hyperparameters was difficult. However, this was not the case with the visual domain adaptation (Section 4.6.1), where just a visual inspection of the fake samples generated was enough to judge their quality. Using methods like Structural Similarity Index Measure (SSIM)[92], one can quantify the comparison to real and fake images. However, this approach did not yield consistent results for the spectrogram generated in the experiment here (Section 4.6). A technique which can provide similarity scores between spectrogram samples would provide further insights into the domain adaptation experiment.

4. In the current approach, only sounds related to approaching vehicles are present. Future data collection could extend the current dataset to include enough samples of other salient sounds in traffic such as emergency vehicle sirens, reversing beeps

of vehicles etc. Further, a study could be carried out to determine an exhaustive list of sounds that might be useful to IV while it is navigating through traffic.

5. The localization approach (Appendix A) relies on the reflection patterns, whereas the classification indirectly relies on the same hoping that reflections would enhance the audibility of the sound made by the approaching vehicle. Hence, combining the localization and classification methods could yield a better performing acoustic perception system. This could be taken forward in one of the two approaches mentioned below.

    - The disadvantage of the current system implemented in this work is that the microphone picks up sounds from all the directions including those that are irrelevant to the IV (for e.g. right behind the vehicle). The reflection patterns surrounding the ego-vehicle could be studied in more detail. By using the knowledge of these reflections, beamforming techniques [93] can be used to steer the microphone array to pick up sound signal in relevant directions only. This filtered audio signal can then be passed on to an acoustic classification based pipeline trained to determine the nature of the sound source, similar to this work. This approach could be particularly useful in `dynamic` scenarios, as picking up the sound signal from directions away from those producing ego-noise (e.g front engine, tyre noise below) could increase the audibility of the sound from the approaching vehicle. This in turn could lead to more prominent acoustic cues, especially in the `left` and `right` samples and thus improve the performance in `dynamic` scenarios.
    - The CRNN architecture can also be used with a multi-microphone setup to perform simultaneous localization and classification [19]. The simplest way to extend in this direction would be to extract multi-channel features such as GCC-PHAT and use them along with the Log-Mel spectrogram features, as implemented by the authors in [35].

Acoustic perception for IV has been relatively under explored as compared to perception with other line-of-sight based sensor modalities. A particular traffic scenario was considered in this work to demonstrate that it would be advantageous to add acoustic sensing to the existing sensing modalities in the IV. Even with limited amount of data, results from the experimental evaluation of the both the localization and classification approaches are promising. Future efforts should be largely focused on making the both the approaches, either individually or combined, robust to `dynamic` scenarios. Hopefully, this work serves as the foundation for further work in ultimately implementing a robust acoustic perception in IVs and thus augmenting the safety of all traffic participants.

A

Localization based Acoustic Perception

Hearing What You Cannot See: Acoustic Detection Around Corners

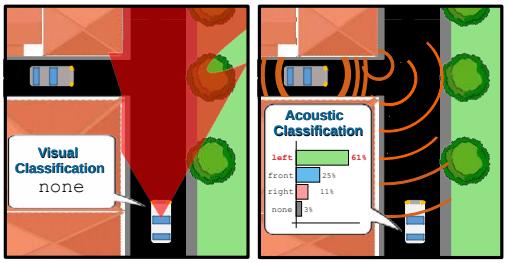
Yannick Schulz*[1] Avinash Kini Mattar*[1] Thomas M. Hehn*[1] Julian F. P. Kooij[1]

***Abstract*— This work proposes to use passive acoustic perception as an additional sensing modality for intelligent vehicles. We demonstrate that approaching vehicles behind blind corners can be detected by sound before such vehicles enter in line-of-sight. We have equipped a research vehicle with a roof-mounted microphone array, and show on data collected with this sensor setup that wall reflections provide information on the presence and direction of occluded approaching vehicles. A novel method is presented to classify if and from what direction a vehicle is approaching before it is visible, using as input Direction-of-Arrival features that can be efficiently computed from the streaming microphone array data. Since the ego-vehicle position within the local geometry affects the perceived patterns, we systematically study several environments types, and investigate generalization across these environments. With a static ego-vehicle, an accuracy of 0.92 is achieved on the hidden vehicle classification task. Compared to a state-of-the-art visual detector, Faster R-CNN, our pipeline achieves the same accuracy more than one second ahead, providing crucial reaction time for the situations we study. While the ego-vehicle is driving, we demonstrate positive results on acoustic detection, still achieving an accuracy of 0.85 within one environment type. We further study failure cases across environments to identify future research directions.**

I. INTRODUCTION

Highly automated and self-driving vehicles currently rely on three complementary main sensors to identify visible objects, namely camera, lidar, and radar. However, the capabilities of these conventional sensors can be limited in urban environments when sight is obstructed by narrow streets, trees, parked vehicles, and other traffic. Approaching road users may therefore remain undetected by the main sensors, resulting in dangerous situations and last-moment emergency maneuvers [1]. While future wireless vehicle-to-everything communication (V2X) might mitigate this problem, creating a robust omnipresent communication layer is still an open problem [2], and excludes road users without wireless capabilities. Acoustic perception does not rely on line-of-sight, and provides a wide range of complementary and important cues on nearby traffic: There are salient sounds with specified meanings, e.g. sirens, car horns, and reverse driving warning beeps of trucks, but also inadvertent sounds from tire-road contact and engine use.

In this work, we propose to use multiple cheap microphones to capture sound as an auxiliary sensing modality for early detection of approaching vehicles behind blind corners in urban environments. Crucially, we show that a data-driven pattern recognition approach can successfully identify such situations from the acoustic reflection patterns on building walls, and provide early warnings before conventional line-of-sight sensing is be able to, see Figure 1. While a vehicle should always exit a narrow street or walled garage cautiously, such early warnings would reduce the need for last-moment emergency braking.



(a) line-of-sight sensing (b) directional acoustic sensing

(c) Sound localization with a vehicle-mounted microphone array detects the wall reflection of an approaching vehicle behind a corner before it appears

Fig. 1. When an intelligent vehicle approaches a blind corner, (a) traditional line-of-sight sensors cannot determine if the corner is safe to pass until the vehicle is much closer, while (b) acoustic cues can provide early warnings of an approaching vehicle. Directional information confirms that the sound comes from around corner, and not a different source. (c) shows this effect using real-time beamforming in real outdoor conditions.

II. RELATED WORKS

Acoustic sensing is an active research topic in domains such as surveillance [3] and robotics [4], e.g. to localizing and separating dominant sound sources [5], [6], While mobile robotic platforms in outdoor environments may suffer from vibrations and wind, various works have demonstrated detection and localization of salient sounds on moving drones [7] and wheeled platforms [8], [9]. To reduce ego-noise of robots, nonnegative matrix factorization represents a widely used approach [10], [11].

Although acoustic cues are known to be crucial for traffic awareness by pedestrians and cyclist [12], only few works have explored passive acoustic sensing as a sensor for Intelligent Vehicles (IV). [13], [9], [14] focus on detec-

*) Shared first authors. 1) All authors are with the Intelligent Vehicles Group, TU Delft, The Netherlands. Primary contact: J.F.P.Kooij@tudelft.nl

tion and tracking in direct line-of-sight. [15], [16] address detection behind corners from a static observer. [15] only show experiments without directional estimation. [16] tries to accurately model wave refractions, but experiments in an artificial lab setup show limited success. Both [15], [16] rely on strong modeling assumptions, ignoring that other informative patterns could be present in the acoustic data. Acoustic traffic perception is furthermore used for road-side traffic monitoring, e.g. to counting vehicles and estimating traffic density [17], [18]. While the increase in Electric Vehicles (EVs) may reduce overall traffic noise, [19] shows that at 20-30km/h the noise levels for EV and internal combustion vehicles are already similar due to tire-road contact. [20] finds that at lower speeds the difference is only about 4-5 dB, though many EVs also suffer from audible narrow peaks in the spectrum. As low speed EVs can impact acoustic awareness of humans too [12], legal minimum sound requirements for EVs are being proposed [21], [22].

Direction-of-Arrival estimation is a key task for sound source localization, and over the past decades many algorithms have been proposed [23], such as the Steered-Response Power Phase Transform (SRP-PHAT) [24] which is well-suited for reverberant environments with possibly distant unknown sound sources. Still, in urban settings nearby walls, corners and surfaces distort sound signals through reflections and diffraction [25]. Accounting for such distortions has shown to improve localization [8], [26], but only in controlled indoor environments where accurate knowledge of the surrounding geometry is available.

Recently, data-driven methods have shown promising results in challenging real-world conditions for various acoustic tasks. For instance, learned sound models assist monaural source separation [27], and source localization from direction-dependent attenuations by fixed structures [28]. Increasingly, deep learning is used for audio classification [29], [30], localization [31], [32], and even sound wave generation [33]. Analogous to our work, [34] presents a first deep learning method for sensing around corners but with automotive radar. Thus, while the effect of occlusions on sensors measurements is difficult to model [16], data-driven approaches appear to be a good alternative.

This paper provides the following contributions: First, we successfully demonstrate in real-world outdoor conditions that a vehicle-mounted microphone array can detect approaching vehicles behind blind corners before line-of-sight detection is feasible. This is a key advantage for intelligent vehicles, where passive acoustic sensing is still a relatively under-explored topic. Our experiments investigate the impact on accuracy and detection time for various conditions, such as different locations and acoustic environments, driving versus static ego-vehicle, and compare to current visual and acoustic baselines.

Second, we propose a data-driven detection pipeline to efficiently address this task and show that it outperforms model-driven acoustic signal processing. We cast the detection task as a multi-class classification problem to identify if and from what corner a vehicle is approaching, and demonstrate that Direction-of-Arrival can provide robust and well known features as the input to a classifier, even without deep learning a feature extractor on large amounts of data.

Third, for our experiments we collected a new audio-visual dataset in real-world urban environments. To collect data, we mounted a front-facing microphone array on our research vehicle, which additionally has a front-facing cameras. This prototype setup facilitates qualitative and quantitative experimentation of different acoustic perception tasks.[1]

III. Approach

Ideally, an ego-vehicle driving through an area with occluding structures is able to early predict *if* and from *where* another vehicle is approaching, even if it is from behind a blind corner as illustrated in Figure 1. Concretely, in this work this task is studied by aiming to distinguish three situations as early as possible using ego-vehicle sensors only:

- an occluded vehicle approaches from behind a corner on the *left*, and only moves into view last-moment when the ego-vehicle is about to reach the junction,
- same, but vehicle approaches *right* behind a corner,
- no vehicle is approaching.

We propose to consider this task an online classification problem. As the ego-vehicle approaches a blind corner, the acoustic measurements made over short time spans should be assigned to one in a set of four classes, $\mathcal{C} = \{\texttt{left}, \texttt{front}, \texttt{right}, \texttt{none}\}$, where `left`/`right` indicates a still occluded (i.e. not yet in direct line-of-sight) approaching vehicle behind a corner on the left/right, `front` that the vehicle is already in direct line-of-sight, and `none` that no vehicle is approaching.

In Section III-A we shall first consider two line-of-sight baseline approaches for detecting vehicles. Section III-B then elaborates our proposed extension to acoustic non-line-of-sight detection. Section III-C provides details of our vehicle's novel acoustic sensor setup used for data collection.

A. Line-of-sight detection

We first consider how the task would be addressed with line-of-sight vehicle detection using either conventional cameras, or using past work on acoustic vehicle detection.

a) Visual detection baseline: Cameras are currently one of the de-facto choices for detecting vehicles and other objects within line-of-sight, as data-driven Convolutional Neural Networks have proven to be highly effective on images. However, visual detection can only detect vehicles that are already (partially) visible, and thus only distinguishes between the `front` and `none`. To demonstrate this, we use Faster R-CNN [35], a state-of-the-art visual object detector, as a visual baseline on the ego-vehicle's front-facing camera.

b) Acoustic detection baseline: Next, we consider that the ego-vehicle is equipped with an array of M microphones, and leverage robust beamforming to estimate the Direction-of-Arrival (DoA) of tire and engine sounds originating from

[1]Code and processed data will be released upon article acceptance.

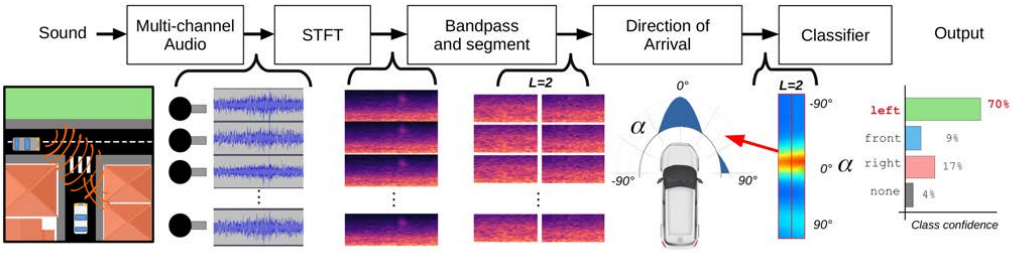


Fig. 2. Overview of our acoustic detection pipeline, see Section III-B for an explanation of the steps.

the approaching vehicle. DoA estimation directly identifies the presence and direction of such sound sources, and has been shown to work in unoccluded conditions [13], [9]. Since sounds can be heard around corners, and low frequencies diffract ("bend") around corners [25], one might wonder: Does the DoA of the sound of an occluded vehicle correctly identify from where a vehicle is approaching? To test this hypothesis for our target real-world application, our second baseline follows previous works [13], [9] and directly uses the salient DoA estimate.

Specifically, the implementation uses the Steered-Response Power-Phase Transform (SRP-PHAT) [24] for DoA estimation. SRP-PHAT relates the spatial layout of sets of microphone pairs and the temporal offsets of the corresponding audio signals with their relative distance to the sound source. To apply SRP-PHAT on M continuous synchronized signals, only the most recent δt seconds are considered. On each signal, a Short-Time Fourier Transform (STFT) is computed with a Hann windowing function, and a frequency bandpass only keeps responses in the $[f_{min}, f_{max}]$ Hz range. Using generalized cross-correlation of the M STFTs, SRP-PHAT computes the DoA energy $r(\alpha)$ for any given azimuth angle α around the vehicle, where $\alpha = -90°/0°/+90°$ indicates an angle towards the left/front/right of the vehicle respectively. Hence, the angle $\alpha_{\max}$ is obtained towards the most salient sound as $\alpha_{\max} = \arg\max r(\alpha)$. If the hypothesis holds that the salient sounds direction $\alpha_{\max}$ remains intact due to diffraction, one only needs to determine if this is beyond some sufficient threshold α_{th}. The baseline thus assigns class `left` if $\alpha_{\max} < -\alpha_{\text{th}}$, `front` if $-\alpha_{\text{th}} \leq \alpha_{\max} \leq +\alpha_{\text{th}}$, and `right` if $\alpha_{\max} > +\alpha_{\text{th}}$. We shall evaluate this baseline on the easier task of only separating these three classes, and ignore the `none` class.

B. Non-line-of-sight acoustic detection

We argue that in contrast to line-of-sight detection, DoA estimation alone is unsuited for occluded vehicle detection (and confirm this in Section IV-C). Salient sounds produce sound wave reflections on surfaces, such as walls (see Figure 1c), and thus the DoA does not reflect the actual location of the sound source. Further, modelling the sounds propagation [8] while driving through uncontrolled outdoor environments is challenging, especially as accurate models of the local geometry are missing. Instead we keep the robust DoA estimation using SRP-PHAT, but use the *full energy distribution* that captures all reflection patterns in front of the ego-vehicle. Rather than modeling these reflections, we take a data-driven approach and treat these as features to train a classifier on.

An overview of the proposed processing pipeline is shown in Figure 2. We again create M STFTs, using a temporal windows of δt seconds, Hann windowing function and a frequency bandpass of $[f_{min}, f_{max}]$ Hz. Notably, we do not apply any other form of noise filtering or suppression. To capture temporal changes in the reflection pattern, we split the STFTs along the temporal dimension into L non-overlapping segments. For each segment, we compute the DoA energy at multiple azimuth angles α in front of the vehicle. We distribute the azimuth range $[-90°, +90°]$ into B equal bins $\alpha_1, \cdots, \alpha_B$. From the original M signals, we thus obtain L response vectors $\boldsymbol{r}_l = [r(\alpha_1), \cdots, r(\alpha_B)]^\top$, containing the response for B angles of a segment l. Finally, these are concatenated to a $(L \times B)$-dimensional feature vector $\boldsymbol{x} = [\boldsymbol{r}_1, \cdots, \boldsymbol{r}_L]^\top$, for which a Support Vector Machine is trained to predict $\mathcal{C}$. Note that increasing the temporal resolution by having more segments L comes at the trade-off of a increased final feature vector size and reduced DoA estimation quality due to short windows.

C. Acoustic perception research vehicle

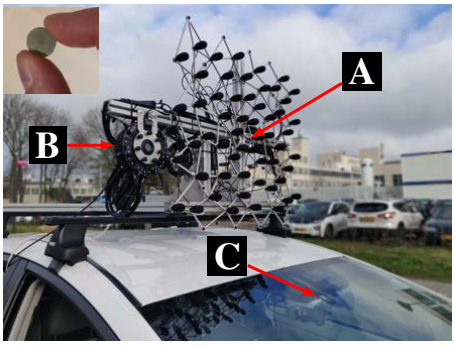


Fig. 3. Sensor setup of our test vehicle. A: Center of the 56 MEMS acoustic array. B: signal processing unit. C: front camera behind windscreen. Inset: the diameter of a single MEMS microphone is only 12mm.

To collect real-world data and demonstrate non-line-of-sight detection, a custom microphone array was mounted on the roof rack of our research vehicle [36], a hybrid electric Toyota Prius. The microphone array hardware consists of 56 ADMP441 MEMS microphones, support data acquisition at 48 kHz sample rate, 24 bits resolution, synchronous sampling, and was bought from *CAE Software & Systems GmbH* with a metal frame. On this metal frame the microphones are distributed semi-randomly in a $0.8m \times 0.7m$ square while the microphone density remains homogeneous. The general purpose layout was designed by the company through stochastic optimization to have large variance in inter-microphone distances and serve a wide range of acoustic imaging tasks. The center of the microphone array (see Figure 3) is about 1.78m above the ground, and 0.54m above and 0.50m behind the used front camera.

The vehicle is also equipped with a front-facing camera for data collection and processing. A signal processing unit receives the analog microphone signals, and sends the data to a PC through Ethernet running a custom software interface for the Robot Operating System (ROS). As depicted in the inset of the Figure 3, the microphones themselves are only 12mm wide and cost only about US$1. In the future, the array can be rearranged with fewer microphones and placed at different locations around the vehicle rather than on top, and integrated in a smaller form factor. Still, the large array allows to investigate the impact on the number of microphones, and the 2D planar arrangement provides both horizontal and vertical resolution such that Direction-of-Arrival responses (both, horizontal and vertical) can be overlaid as a 2D heatmap [37] on the front camera image.

Our implementation uses a custom ROS node to collect synchronized microphone signals together with other vehicle sensor data. Processing is done in python, using *pyroomacoustics* [23] for acoustic feature extraction, and *scikit-learn* [38] for classifier training.

IV. Experiments

To validate our method, we created a novel dataset with our acoustic research vehicle in real-world urban environments. We first illustrate the quality of acoustic beamforming in such conditions before turning to our main experiments.

A. Line-of-sight localization – qualitative results

As explained in Section III-C, the heatmaps of the 2D DoA results can be overlaid with the camera images. Figure 4 demonstrates some interesting qualitative findings observed while using the vehicle in urban traffic. The examples highlight that beamforming can indeed pick up various important acoustic events for autonomous driving in line-of-sight, such as the presence of vehicles and some vulnerable road users (e.g. strollers). Remarkably, even electric scooters, and oncoming traffic *while the ego-vehicle is driving* are recognized as salient sound sources. Even sound sources that are not yet in line-of-sight provide a salient signal due to reflections, see Figure 1c. Overall, these observations show the feasibility of acoustic detection of (occluded) traffic.

TABLE I

Samples per subset. In the ID, S/D indicates Static/Dynamic ego-vehicle, A/B the environment type (see figure 5).

ID	`left`	`front`	`right`	`none`	Sum
SA1 / DA1	14 / 19	30 / 38	16 / 19	30 / 37	90/113
SA2 / DA2	22 / 7	41 / 15	19 / 13	49 / 43	131/ 43
SB1 / DB1	17 / 18	41 / 35	24 / 17	32 / 36	114/106
SB2 / DB2	28 / 10	55 / 21	27 / 11	43 / 22	153/ 64
SB3 / DB3	22 / 19	45 / 38	23 / 19	45 / 36	135/112
SAB / DAB	103/ 73	212/148	109/ 75	199/144	*623/440*

B. Non-line-of-sight dataset and evaluation metrics

The quantitative experiments are designed to separately control and study various factors that could influence acoustic perception. We collected multiple recordings of the situations explained in Section III at five T-junction locations with blind corners in the inner city of Delft. The locations are categorized into two types of walled acoustical environments, namely types A and B (see Figure 5). At these locations common background noise, such as construction sites and other traffic, was present at various volumes. For safety and control, we did not record in the presence of other motorized traffic on the roads at the target junction.

The recordings can further be divided into Static data, made while is the ego-vehicle in front of the junction but not moving, and more challenging Dynamic data where the ego-vehicle reaches the junction at $\sim$15 km/h.[2] Static data is easily collected, and ensures that the main source of variance is the approaching vehicle's changing position.

For the static case, the ego-vehicle was positioned such that the building corners are still visible in the camera and occlude the view onto the intersecting road (on average a distance of $\sim$7-10m from the intersection). Different types of passing vehicles were recorded, although in most recordings the approaching vehicle was a Škoda Fabia 1.2 TSI (2010) driven by one of the authors. For the Dynamic case, coordinated recordings with the Škoda Fabia were conducted to ensure that encounters were relevant and executed in a safe manner. Situations with `left`/`right`/`none` approaching vehicles were performed in arbitrary order to prevent undesirable acoustic correlation with background noise to some classes. In $\sim$70% of the total Dynamic recordings and $\sim$19.5% of the total Static recordings, the ego-vehicle's noisy internal combustion engine was running to charge its battery.

a) Sample extraction: For each Static recording with an approaching target vehicle, the time t_0 is manually annotated as the moment when the approaching vehicle enters direct line-of-sight. Since the quality of our t_0 estimate is bound the ego-vehicle's camera frame rate (10 Hz), we conservatively regard the last image *before* the incoming vehicle is visible as t_0. Thus, there is no line-of-sight at $t \leq t_0$, and at $t > t_0$ the vehicle is considered visible (even though it might only be a fraction of the body). For the Dynamic data, this annotation is not feasible as the approaching car

[2]Please see animated results in supplementary video.

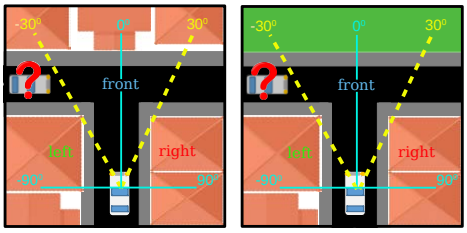
(a) Stroller at a distance (b) Electric scooter (c) Scooter overtaking (d) Car passing by (e) Oncoming car

Fig. 4. Qualitative examples of 2D Direction-of-Arrival estimation overlaid on the camera image (zoomed). (a): Stroller wheels are picked up even at a distance. (b), (c): Both conventional and more quiet electric scooters are detected. (d): The loudest sound of a passing vehicle is typically the road contact of the individual tires. (e): Even when the ego-vehicle drives at $\sim$ 30 km/h, oncoming moving vehicles are still registered as salient sound sources.

may be in direct line-of-sight, yet outside the limited field-of-view of the front-facing camera as the ego-vehicle has advanced into the intersection. Thus, annotating t_0 based on the camera images is not a representative for line-of-sight detection. To still compare our results across locations, we manually annotate the time τ_0, the moment when the ego-vehicle is at the same position as in the corresponding Static recordings. All Dynamic recordings are aligned to that time as it represents the moment where the ego-vehicle should make a classification decision, irrespective if an approaching vehicle is about to enter line-of-sight or still further away.

From the recordings, short $1s$ audio samples are extracted for our dataset. Let t_e, the end of the time window $[t_e - 1s, t_e]$, denote a sample's time stamp at which a prediction could be made. For Static recordings of the `left` and `right` class, at $t_e = t_0$ samples with the corresponding class label are extracted. For Dynamic recordings, `left` and `right` samples are extracted at $t_e = \tau_0 + 0.5$s to center them around τ_0, thus corresponding Static ego-vehicle position. In both cases, at $t_e = t_0 + 1.5$s also a sample for the `front` class is extracted from these recordings. Samples for the `none` class were from recordings with no approaching vehicles. Table I lists statistics of the number of samples per class in our dataset at each recording location.

b) Data augmentation: Table I shows that the data acquisition scheme produced imbalanced class ratios, with about half the samples for `left`, `right` compared to `front` and `none`. Our experiments therefore explore *data augmentation* for training. By exploiting the symmetry of the angular DoA bins, augmentation will double the `right` and `left` class samples by reversing the azimuth bin order in all $\boldsymbol{r}_l$, resulting in new features for the opposite label, i.e. as if additional data was collected at mirrored locations.



(a) **Type A**: completely walled (b) **Type B**: walled exit

Fig. 5. Schematics of considered environment types. The ego-vehicle approaches the junction from the bottom. Another vehicle might approach behind the left or right blind corner. Dashed lines indicate the camera FoV.

TABLE II

BASELINE COMPARISON AND HYPERPARAMETER STUDY W.R.T. OUR REFERENCE CONFIGURATION: SVM $\lambda = 1$, $\delta t = 1$, $L = 2$, DATA AUGMENTATION. RESULTS ON STATIC DATA. * DENOTES *our* PIPELINE.

Run	Accuracy	$J_{\texttt{left}}$	$J_{\texttt{front}}$	$J_{\texttt{right}}$	$J_{\texttt{none}}$
* *(reference)*	**0.92**	0.79	0.89	**0.87**	0.83
* wo. data augment.	**0.92**	0.75	0.91	0.78	0.83
* w. $\delta t = 0.5$s	0.91	0.75	0.89	**0.87**	0.82
* w. $L = 1$	0.86	0.64	0.87	0.73	0.79
* w. $L = 3$	**0.92**	0.74	0.92	0.82	0.81
* w. $L = 4$	0.90	0.72	0.90	0.77	0.83
* w. SVM $\lambda = 0.1$	0.91	0.78	0.89	0.81	0.82
* w. SVM $\lambda = 10$	0.91	**0.81**	0.86	0.84	0.83
DoA-only [13], [9]	0.64	0.11	0.83	0.28	-
Faster R-CNN	0.58	0.00	**0.94**	0.00	**0.91**

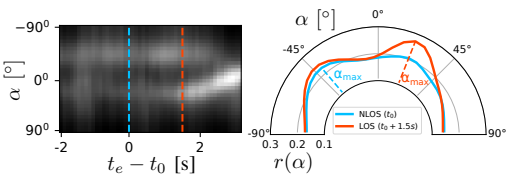
c) Metrics: We report the overall accuracy, and the per-class Jaccard index (a.k.a. Intersection-over-Union) as a robust measure of one-vs-all performance. First, for each class c the True Positives/Negatives (TP_c/TN_c), and False Positives/Negatives (FP_c/FN_c) are computed, considering target class c is positive and the other three classes are jointly negative. Given the total number of test samples N, the overall accuracy is then $\left(\sum_{c\in\mathcal{C}} TP_c\right)/N$ and the per-class Jaccard index is $J_c = TP_c/(TP_c + FP_c + FN_c)$.

C. Training and impact of classifier and features

First, the overall system performance and hyperparameters are evaluated on all Static data from both type A and B locations (i.e. subset ID 'SAB') using 5-fold cross-validation. The folds are fixed once for all experiments, with the training samples of each class equally distributed among folds.

We fix the frequency range to $f_{min} = 50$Hz, $f_{max} = 1500$Hz, and the number of azimuth bins to $B = 30$ (Section III-B). For efficiency and robustness, a linear Support Vector Machine (SVM) is used with $l2-$regularization weighted by hyperparameter λ. Other hyperparameters to explore include the sample length $\delta t \in \{0.5\text{s}, 1\text{s}\}$, the segment count $L \in \{1, 2, 3, 4\}$, and using/not using data augmentation.

Our final choice and reference is the SVM with $\lambda = 1$, $\delta t = 1$s, $L = 2$, and data augmentation. Table II shows the results of these parameter choices, and the impact of changing our choices. Note that without data augmentation, and with $L = 3$ segments, overall similar accuracy is



(a) DoA energy over time (b) DoA relative to ego-vehicle

Fig. 6. DoA energy for vehicle approaching `right` (true $\alpha > 45°$) behind a corner. While still occluded, the main DoA direction is incorrect ($\alpha_{max} < 30°$) until it is already in line-of-sight (LOS) after $t0 + 1.5s$.

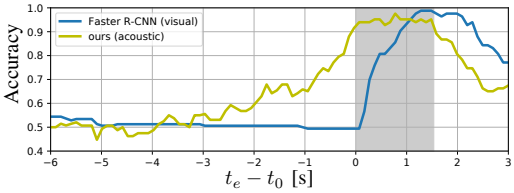


Fig. 7. Accuracy over test time t_e of our acoustic and the visual baseline on 83 Static recordings. Gray region indicates the other vehicle is half-occluded and two labels, `front` and either `left` or `right`, are considered correct.

achieved, but our reference choices perform better on `left` and `right`. The overall accuracy of the classifiers and hyperparameters on all these choices is similar, though SVM $\lambda = 1$ has a slight advantage. More importantly, it performs well on both `left` and `right`, so we keep these parameters as our main method for all future experiments.

The table also shows the results of the DoA-only baseline explained in Section III-A using $\alpha_{\text{th}} = 50°$, which was found through a grid search in the range $[0°, 90°]$. As expected, the DoA-only baseline [13], [9] shows weak performance for all metrics. While the sound source is occluded, the most salient sound direction does not represent its origin, but its reflections. Figure 6 shows how the full DoA energy develops over time for a car approaching right. When it is still occluded at t_0, there are multiple peaks and the most salient one is a reflection on the left ($\alpha_{max} \approx -40°$). Only once the car is in line-of-sight (t_0+1.5s) does the main mode clearly represent its true direction ($\alpha_{max} \approx +25°$).

The bottom row of the table shows the visual baseline, a Faster R-CNN R50-C4 model trained on the COCO dataset [39]. To avoid false positive detections, we set the score threshold of 75% and additionally required a bounding box height of 100 pixels to ignore cars far away in the background, which were not of interest. Generally this threshold is already exceeded once the hood of the approaching car is visible. While performing well on `front` and `none`, this visual baseline shows poor overall accuracy as it is physically incapable of classifying `left` and `right`.

D. Detection time before appearance

Ultimately, the goal is to know whether our acoustic method can be detected approaching vehicles earlier than the state-of-the-art visual baseline. For this purpose, their online performance is compared next.

Static recordings are divided into a fixed training (328 recordings) and test (83 recordings) split, stratified to adequately represent labels and locations. The training was conducted as in Section IV-C with `left` and `right` samples extracted at $t_e = t_0$. The visual baseline is evaluated on every camera frame (10 Hz). Our detector is evaluated on a sliding window of $1s$ across the 83 test recordings. To account for the transition period when the car can be partly occluded, `front` predictions by both methods are accepted as correct starting at $t = t_0$. For recordings of classes `left` and `right`, these classes are accepted as correct until $t = t_0 + 1.5$, allowing for temporal overlap with `front`.

Figure IV-D illustrates the accuracy on the test recordings for different evaluation times t_e. The overlap region is indicated by the gray area after $t_e = t_0$ and its beginning thus marks when a car enters the field of view. At $t_e = t_0$, just before entering the view of the camera, the approaching car can be detected with 0.94 accuracy by our method. This accuracy is achieved more than one second ahead of the visual baseline, showing that our acoustic detection gives the ego-vehicle additional reaction time. The accuracy of 0.8 at 0.24s before t_0 supports this.

Figure 8 shows the per-class probabilities as a function of extraction time t_e on the test set, separated by recording situations. The SVM class probabilities are obtained with the method in [40]. The probabilities for `left` show that the model initially predicts on average that no vehicle is approaching. Towards t_0, the `none` class becomes less likely and the model increasingly favors the correct `left` class. A short time after t_0, the prediction flips to the `front` class and switches to `right` after the vehicle passed. Similar (mirrored) behavior is observed for vehicles approaching from the right. The `none` class is constantly predicted as likeliest when no vehicle is approaching. Overall, the prediction matches the events of the recorded situations remarkably well. Still, note that the probabilities of `left` and `right` are only rising when the approaching vehicle is almost in line-of-sight, which corresponds to the extraction time of the training samples.

TABLE III

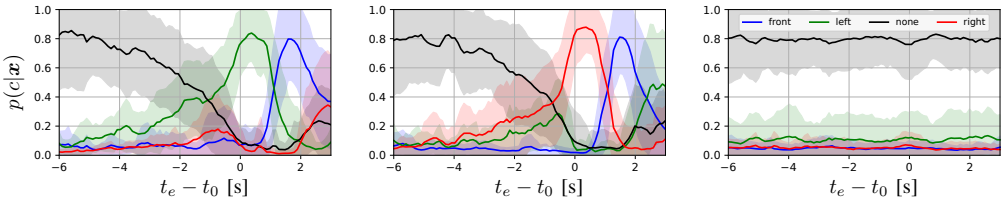
CROSS-VALIDATION RESULTS PER ENVIRONMENT ON DYNAMIC DATA.

Subset	Accuracy	$J_{\texttt{left}}$	$J_{\texttt{front}}$	$J_{\texttt{right}}$	$J_{\texttt{none}}$
DAB	0.76	0.42	0.79	0.42	0.65
DA	0.85	0.63	0.86	0.66	0.71
DB	0.74	0.29	0.80	0.41	0.64

E. Impact of the moving ego-vehicle

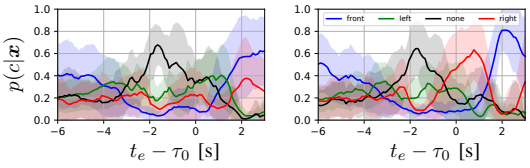
Next, we evaluate our classifier by cross-validation on the full Dynamic data and further distinguish between the different environment subsets. As for the Static data, 5-fold cross-validation is applied to each subset, keeping the class distribution balanced across folds.

Table III lists the corresponding metrics for each subset. On the full Dynamic data (DAB), the accuracy indicates



(a) `left` recordings (b) `right` recordings (c) `none` recordings

Fig. 8. Mean and std. dev. of predicted class probabilities at different times t_e on test set recordings of the Static data (blue is `front`, green is `left`, red is `right`, and black is `none`). Each figure shows recordings of a different situation. The approaching vehicle appears in views just after $t_e - t_0 = 0$.



(a) `left` recordings (b) `right` recordings

Fig. 9. Mean and std. dev. of predicted class probabilities at different times t_e on `left` and `right` test set recordings of the Dynamic data. The ego-vehicle reached location of training data when $t_e - \tau_0 = 0$.

TABLE IV

GENERALIZATION ACROSS LOCATIONS AND ENVIRONMENTS.

Training	Test	Accuracy	$J_{\texttt{left}}$	$J_{\texttt{front}}$	$J_{\texttt{right}}$	$J_{\texttt{none}}$
SB	SA	0.67	0.03	0.71	0.09	0.61
SA	SB	0.79	0.41	0.83	0.61	0.67
DB	DA	0.56	0.29	0.71	0.28	0.16
DA	DB	0.56	0.21	0.47	0.27	0.47

decent performance, but the metrics for `left` and `right` classes are much worse compared to the Static results in Table II. Separating subsets DA and DB reveals that the performance is highly dependent on the environment type. In fact, even with limited training data and large data variance from a driving ego-vehicle, we find already acceptable classification performance on type A environments, and notice that low `left` and `right` performance is mostly due to type B environments. We hypothesize that the less open type A environments reflect more target sounds, and are more shielded from potential noise sources.

We also analyze the temporal behavior of our method on Dynamic data. Unfortunately, a fair comparison with a visual baseline is not possible: the ego-vehicle often reaches the intersection early, and the approaching vehicle is unoccluded but still outside the front-facing camera's field of view (cf. τ_0 extraction in Section IV-B). Yet, the evolution of the predicted probabilities can be compared to those on the Static data in Section IV-D. Figure 9 illustrates the average predicted probabilities over 59 Dynamic test set recordings from all locations, after training on samples from the remaining 233 recordings. The classifier on average correctly predicts `right` samples (Figure 9b), between $t_e = \tau_0$ to $t_e = \tau_0 + 0.5$s. Of the `left` recordings at these times, many are falsely predicted as `none`, only few are confused with `right`. Furthermore, the changing ego-perspective of the vehicle results in a more switching DoA-energy features and thus class predictions, compared to the Static results in Figure 8. This indicates that it might help to include the ego-vehicle's relative position as an additional feature, and obtain more varied training data to cover the positional variations.

F. Generalization across acoustic environments

Training and testing on the samples from the same types of environment provides decent performance as seen in Sections IV-C and IV-E. To understand whether these results still hold when training and test set are from different environment types, our classifier is now trained on one environment type and evaluated on all samples of the other type.

In Table IV, combinations of training and test sets are listed. Compared to the results for Static and Dynamic data (see Tables II and III), the reported results in the table show a general trend. If the classifier is trained on one environment and tested on the environment, it performs worse than when samples of the same location are present. In particular, the classifier trained on SB and tested on SA is not capable to correctly classify samples of `left` and `right`, while inverse training and testing performs much better. On the Dynamic data, such pronounced effect are not visible, but overall shows decreased accuracy compared to the Static data. Interestingly, the `none` class seems to be much more difficult for the classifier than on the Static data.

In summary, the classifier does not generalize too well from one environment to another. Yet, for some combinations, the classifier may still show positive results, despite never seeing any sample from this environment before. While robustness might increase with more data from varied locations, a future research direction would be to make the detector conditional on the environmental map information.

G. Computational cost and array configuration

Finally, we investigate the computational cost for varying the number of microphones M. For a subset of M microphones we sample 100 random microphone configurations out of the $\binom{56}{M}$ possibilities, and keep the best performing one

on the Static data. For our unoptimized implementation, computation time is only 0.24/0.14/0.04s for $M = 56/28/14$, thus showing high computational efficiency. Interestingly, for up to $M \geq 7$ the overall accuracy remains above 90%.

V. Conclusions

We conclude that a vehicle mounted microphone array can be used to acoustically detect approaching vehicles behind blind corners, and may in the future also serve other acoustic sensing tasks. Our pipeline using 56 microphones achieved an accuracy of 0.92 on our 4-class hidden car classification task for a static ego-vehicle. In our experimental setup, an approaching vehicle was detected with the same accuracy as our visual baseline already more than one second ahead. This advantage in reaction time is crucial in the situations discussed in our work. When the ego-vehicle was moving, our method still performed well on one environment, but had difficulties on the other. In experiments across environments, differences between the environment types were also observed. This could be addressed by first classifying the environment from other sensors or map information. We are encouraged by our initial findings, though more experimentation is needed as we still used limited data and controlled conditions. Future work will focus on acquiring more training data to improve robustness across locations and enable classification of multiple simultaneous sources.

References

[1] C. G. Keller, T. Dang, H. Fritz, A. Joos, C. Rabe, and D. M. Gavrila, "Active pedestrian safety by automatic braking and evasive steering," *IEEE T-ITS*, vol. 12, no. 4, pp. 1292–1304, 2011.

[2] Z. MacHardy, A. Khan, K. Obana, and S. Iwashina, "V2X access technologies: Regulation, research, and remaining challenges," *IEEE Comm. Surveys & Tutorials*, vol. 20, no. 3, pp. 1858–1877, 2018.

[3] M. Crocco, M. Cristani, *et al.*, "Audio surveillance: A systematic review," *ACM CSUR*, vol. 48, no. 4, pp. 1–46, 2016.

[4] C. Rascon and I. Meza, "Localization of sound sources in robotics: A review," *Robotics & Autonomous Systems*, vol. 96, pp. 184–210, 2017.

[5] L. Wang and A. Cavallaro, "Acoustic sensing from a multi-rotor drone," *IEEE Sensors Journal*, vol. 18, no. 11, pp. 4570–4582, 2018.

[6] S. Argentieri, P. Danes, and P. Souères, "A survey on sound source localization in robotics: From binaural to array processing methods," *Computer Speech & Language*, vol. 34, no. 1, pp. 87–112, 2015.

[7] K. Okutani, T. Yoshida, K. Nakamura, and K. Nakadai, "Outdoor auditory scene analysis using a moving microphone array embedded in a quadrocopter," in *IEEE/RSJ IROS*. IEEE, 2012, pp. 3288–3293.

[8] I. An, M. Son, D. Manocha, and S.-e. Yoon, "Reflection-aware sound source localization," in *ICRA*. IEEE, 2018, pp. 66–73.

[9] Y. Jang, J. Kim, and J. Kim, "The development of the vehicle sound source localization system," in *APSIPA*. IEEE, 2015, pp. 1241–1244.

[10] K. Nakadai, T. Tezuka, and T. Yoshida, "Ego-noise suppression for robots based on semi-blind infinite non-negative matrix factorization," *J. of Robotics and Mechatronics*, vol. 29, no. 1, pp. 114–124, 2017.

[11] A. Schmidt, A. Brendel, *et al.*, "Motor data-regularized nonnegative matrix factorization for ego-noise suppression," *EURASIP J. on Audio, Speech, & Music Proc.*, vol. 2020, no. 1, pp. 1–15, 2020.

[12] A. Stelling-Kończak, M. Hagenzieker, and B. V. Wee, "Traffic sounds and cycling safety: The use of electronic devices by cyclists and the quietness of hybrid and electric cars," *Transport Reviews*, vol. 35, no. 4, pp. 422–444, 2015.

[13] M. Mizumachi, A. Kaminuma, N. Ono, and S. Ando, "Robust sensing of approaching vehicles relying on acoustic cues," *Sensors*, vol. 14, no. 6, pp. 9546–9561, 2014.

[14] A. V. Padmanabhan, H. Ravichandran, *et al.*, "Acoustics based vehicle environmental information," SAE, Tech. Rep., 2014.

[15] K. Asahi, H. Banno, O. Yamamoto, A. Ogawa, and K. Yamada, "Development and evaluation of a scheme for detecting multiple approaching vehicles through acoustic sensing," in *IV Symposium*. IEEE, 2011, pp. 119–123.

[16] V. Singh, K. E. Knisely, *et al.*, "Non-line-of-sight sound source localization using matched-field processing," *J. of the Acoustical Society of America*, vol. 131, no. 1, pp. 292–302, 2012.

[17] T. Toyoda, N. Ono, S. Miyabe, T. Yamada, and S. Makino, "Traffic monitoring with ad-hoc microphone array," in *Int. Workshop on Acoustic Signal Enhancement*. IEEE, 2014, pp. 318–322.

[18] S. Ishida, J. Kajimura, M. Uchino, S. Tagashira, and A. Fukuda, "SAVeD: Acoustic vehicle detector with speed estimation capable of sequential vehicle detection," in *ITSC*. IEEE, 2018, pp. 906–912.

[19] U. Sandberg, L. Goubert, and P. Mioduszewski, "Are vehicles driven in electric mode so quiet that they need acoustic warning signals," in *Int. Congress on Acoustics*, 2010.

[20] L. M. Iversen and R. S. H. Skov, "Measurement of noise from electrical vehicles and internal combustion engine vehicles under urban driving conditions," *Euronoise*, 2015.

[21] R. Robart, E. Parizet, J.-C. Chamard, *et al.*, "eVADER: A perceptual approach to finding minimum warning sound requirements for quiet cars." in *AIA-DAGA 2013 Conference on Acoustics*, 2013.

[22] S. K. Lee, S. M. Lee, T. Shin, and M. Han, "Objective evaluation of the sound quality of the warning sound of electric vehicles with a consideration of the masking effect: Annoyance and detectability," *Int. Journal of Automotive Tech.*, vol. 18, no. 4, pp. 699–705, 2017.

[23] R. Scheibler, E. Bezzam, and I. Dokmanić, "Pyroomacoustics: A python package for audio room simulation and array processing algorithms," in *ICASSP*. IEEE, 2018, pp. 351–355.

[24] J. H. DiBiase, *A high-accuracy, low-latency technique for talker localization in reverberant environments using microphone arrays*. Brown University Providence, RI, 2000.

[25] M. Hornikx and J. Forssén, "Modelling of sound propagation to three-dimensional urban courtyards using the extended Fourier pstd method," *Applied Acoustics*, vol. 72, no. 9, pp. 665–676, 2011.

[26] W. Zhang, P. N. Samarasinghe, H. Chen, and T. D. Abhayapala, "Surround by sound: A review of spatial audio recording and reproduction," *Applied Sciences*, vol. 7, no. 5, p. 532, 2017.

[27] K. Osako, Y. Mitsufuji, *et al.*, "Supervised monaural source separation based on autoencoders," in *ICASSP*. IEEE, 2017, pp. 11–15.

[28] A. Saxena and A. Y. Ng, "Learning sound location from a single microphone," in *ICRA*. IEEE, 2009, pp. 1737–1742.

[29] J. Salamon and J. P. Bello, "Deep convolutional neural networks and data augmentation for environmental sound classification," *IEEE Signal Processing Letters*, vol. 24, no. 3, pp. 279–283, 2017.

[30] A. Valada, L. Spinello, and W. Burgard, "Deep feature learning for acoustics-based terrain classification," in *Robotics Research*. Springer, 2018, pp. 21–37.

[31] W. He, P. Motlicek, and J.-M. Odobez, "Deep neural networks for multiple speaker detection and localization," in *ICRA*. IEEE, 2018, pp. 74–79.

[32] N. Yalta, K. Nakadai, and T. Ogata, "Sound source localization using deep learning models," *J. of Robotics and Mechatronics*, vol. 29, no. 1, pp. 37–48, 2017.

[33] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, *et al.*, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," in *ICASSP*. IEEE, 2018, pp. 4779–4783.

[34] N. Scheiner, F. Kraus, F. Wei, *et al.*, "Seeing around street corners: Non-line-of-sight detection and tracking in-the-wild using doppler radar," in *Proc. of IEEE CVPR*, 2020, pp. 2068–2077.

[35] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[36] L. Ferranti, B. Brito, E. Pool, Y. Zheng, *et al.*, "SafeVRU: A research platform for the interaction of self-driving vehicles with vulnerable road users," in *IV Symposium*. IEEE, 2019, pp. 1660–1666.

[37] E. Sarradj and G. Herold, "A python framework for microphone array data processing," *Applied Acoustics*, vol. 116, pp. 50–58, 2017.

[38] F. Pedregosa, G. Varoquaux, *et al.*, "Scikit-learn: Machine learning in python," *JMLR*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[39] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2," https://github.com/facebookresearch/detectron2, 2019.

[40] T.-F. Wu, C.-J. Lin, and R. C. Weng, "Probability estimates for multi-class classification by pairwise coupling," *JMLR*, vol. 5, no. Aug, pp. 975–1005, 2004.

BIBLIOGRAPHY

[1] Saving lives: Boosting car safety in the EU (2016). https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:52016DC0787&from=EN. Accessed: 01-03-2020.

[2] WHO global status report on road safety (2018). https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries. Accessed: 01-03-2020.

[3] Miloš Bjelić, Miodrag Stanojević, Dragana Šumarac Pavlović, and Miomir Mijić. Microphone array geometry optimization for traffic noise analysis. *The Journal of the Acoustical Society of America*, 141(5):3101–3104, 2017.

[4] T. Heittola, A. Mesaros, T. Virtanen, and M. Gabbouj. Supervised model training for overlapping sound events based on unsupervised source separation. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8677–8681, May 2013.

[5] Tuomas Virtanen, Mark D Plumbley, and Dan Ellis. *Computational analysis of sound scenes and events*. Springer, 2018.

[6] Dropping off passengers vexes self-driving cars, the last step is a doozie. https://www.forbes.com/sites/lanceeliot/2019/07/11/dropping-off-passengers-vexes-self-driving-cars-the-last-step-is-a-doozie/#3d814f123de6. Accessed: 10-03-2020.

[7] Lin Xie, Feifei Lee, Li Liu, Koji Kotani, and Qiu Chen. Scene recognition: A comprehensive survey. *Pattern Recognition*, page 107205, 2020.

[8] E. Çakır, G. Parascandolo, T. Heittola, H. Huttunen, and T. Virtanen. Convolutional recurrent neural networks for polyphonic sound event detection. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(6):1291–1303, June 2017.

[9] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018.

[10] E. Çakir and T. Virtanen. End-to-end polyphonic sound event detection using convolutional recurrent neural networks with learned time-frequency representation input. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7, July 2018.

[11] L. Hertel, H. Phan, and A. Mertins. Comparing time and frequency domain for audio event recognition using deep learning. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 3407–3411, July 2016.

[12] W. Dai, C. Dai, S. Qu, J. Li, and S. Das. Very deep convolutional neural networks for raw waveforms. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 421–425, March 2017.

[13] Y. Tokozume and T. Harada. Learning environmental sounds with end-to-end convolutional neural network. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2721–2725, March 2017.

[14] P. Zinemanas, P. Cancela, and M. Rocamora. End-to-end convolutional neural networks for sound event detection in urban environments. In *2019 24th Conference of Open Innovations Association (FRUCT)*, pages 533–539, April 2019.

[15] Toni Heittola, Emre Çakır, and Tuomas Virtanen. The machine learning approach for analysis of sound scenes and events. In *Computational Analysis of Sound Scenes and Events*, pages 13–40. Springer, 2018.

[16] Huy Phan, Lars Hertel, Marco Maass, and Alfred Mertins. Robust audio event recognition with 1-max pooling convolutional neural networks. *arXiv preprint arXiv:1604.06338*, 2016.

[17] Fabio Vesperini, Leonardo Gabrielli, Emanuele Principi, and Stefano Squartini. Polyphonic sound event detection by using capsule neural networks. *IEEE Journal of Selected Topics in Signal Processing*, 13(2):310–322, 2019.

[18] Wei Xia and Kazuhito Koishida. Sound event detection in multichannel audio using convolutionaltime-frequency-channel squeeze and excitation. *INTERSPEECH*.

[19] Sharath Adavanne, Archontis Politis, Joonas Nikunen, and Tuomas Virtanen. Sound event localization and detection of overlapping sources using convolutional recurrent neural networks. *IEEE Journal of Selected Topics in Signal Processing*, 13(1):34–48, 2018.

[20] Sławomir Kapka and Mateusz Lewandowski. Sound source detection, localization and classification using consecutive ensemble of crnn models. In *Acoustic Scenes and Events 2019 Workshop (DCASE2019)*, page 119, 2019.

[21] Stanley Smith Stevens, John Volkmann, and Edwin B Newman. A scale for the measurement of the psychological magnitude pitch. *The Journal of the Acoustical Society of America*, 8(3):185–190, 1937.

[22] Roy D Patterson, KEN Robinson, John Holdsworth, Denis McKeown, C Zhang, and Michael Allerhand. Complex sounds and auditory images. In *Auditory physiology and perception*, pages 429–446. Elsevier, 1992.

[23] P. Foggia, A. Saggese, N. Strisciuglio, and M. Vento. Cascade classifiers trained on gammatonegrams for reliably detecting audio events. In *2014 11th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 50–55, Aug 2014.

[24] Letizia Marchegiani and Ingmar Posner. Leveraging the urban soundscape: Auditory perception for smart vehicles. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6547–6554. IEEE, 2017.

[25] A. B. Nassif, I. Shahin, I. Attili, M. Azzeh, and K. Shaalan. Speech recognition using deep neural networks: A systematic review. *IEEE Access*, 7:19143–19165, 2019.

[26] Yong Xu, Qiuqiang Kong, Wenwu Wang, and Mark D Plumbley. Large-scale weakly supervised audio classification using gated convolutional neural network. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 121–125. IEEE, 2018.

[27] Emre Cakir, Toni Heittola, Heikki Huttunen, and Tuomas Virtanen. Polyphonic sound event detection using multi label deep neural networks. In *2015 international joint conference on neural networks (IJCNN)*, pages 1–7. IEEE, 2015.

[28] Albert S Bregman. *Auditory scene analysis: The perceptual organization of sound*. MIT press, 1994.

[29] Sharath Adavanne, Giambattista Parascandolo, Pasi Pertilä, Toni Heittola, and Tuomas Virtanen. Sound event detection in multichannel audio using spatial and harmonic features. *arXiv preprint arXiv:1706.02293*, 2017.

[30] Sharath Adavanne, Pasi Pertilä, and Tuomas Virtanen. Sound event detection using spatial features and convolutional recurrent neural network. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 771–775. IEEE, 2017.

[31] Shawn Hershey, Sourish Chaudhuri, Daniel PW Ellis, Jort F Gemmeke, Aren Jansen, R Channing Moore, Manoj Plakal, Devin Platt, Rif A Saurous, Bryan Seybold, et al. Cnn architectures for large-scale audio classification. In *2017 ieee international conference on acoustics, speech and signal processing (icassp)*, pages 131–135. IEEE, 2017.

[32] Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 776–780. IEEE, 2017.

[33] Yun Wang. Polyphonic sound event detection with weak labeling. *PhD thesis*, 2018.

[34] Charles Knapp and Glifford Carter. The generalized correlation method for estimation of time delay. *IEEE transactions on acoustics, speech, and signal processing*, 24(4):320–327, 1976.

[35] Yin Cao, Qiuqiang Kong, Turab Iqbal, Fengyan An, Wenwu Wang, and Mark D Plumbley. Polyphonic sound event detection and localization using a two-stage strategy. *arXiv preprint arXiv:1905.00268*, 2019.

[36] Lie Lu, Hong-Jiang Zhang, and Hao Jiang. Content analysis for audio classification and segmentation. *IEEE Transactions on speech and audio processing*, 10(7):504–516, 2002.

[37] Christian Zieger. An hmm based system for acoustic event detection. In *Multimodal Technologies for Perception of Humans*, pages 338–344. Springer, 2007.

[38] Xi Zhou, Xiaodan Zhuang, Ming Liu, Hao Tang, Mark Hasegawa-Johnson, and Thomas Huang. Hmm-based acoustic event detection with adaboost feature selection. In *Multimodal technologies for perception of humans*, pages 345–353. Springer, 2007.

[39] Emmanouil Benetos, Grégoire Lafay, Mathieu Lagrange, and Mark D Plumbley. Detection of overlapping acoustic events using a temporally-constrained probabilistic model. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6450–6454. IEEE, 2016.

[40] Julien Pinquier, Jean-Luc Rouas, and Régine André-Obrecht. Robust speech/music classification in audio documents. In *Seventh International Conference on Spoken Language Processing*, 2002.

[41] Mahesh Kumar Nandwana and Taufiq Hasan. Towards smart-cars that can listen: Abnormal acoustic event detection on the road. In *INTERSPEECH*, pages 2968–2971, 2016.

[42] Jens Schröder, Stefan Goetze, Volker Grützmacher, and Jörn Anemüller. Automatic acoustic siren detection in traffic noise by part-based models. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 493–497. IEEE, 2013.

[43] Toni Heittola, Annamaria Mesaros, Antti Eronen, and Tuomas Virtanen. Context-dependent sound event detection. *EURASIP Journal on Audio, Speech, and Music Processing*, 2013(1):1, 2013.

[44] A. Temko and C. Nadeu. Classification of meeting-room acoustic events with support vector machines and variable-feature-set clustering. In *Proceedings. (ICASSP '05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, volume 5, pages v/505–v/508 Vol. 5, March 2005.

[45] Andrey Temko, Robert Malkin, Christian Zieger, Dusan Macho, Climent Nadeu, and Maurizio Omologo. Acoustic event detection and classification in smart-room environments: Evaluation of chil project systems. *Cough*, 65(48):5, 2006.

[46] Andrey Temko, Climent Nadeu, and Joan-Isaac Biel. Acoustic event detection: Svm-based system and evaluation setup in clear'07. In *Multimodal Technologies for Perception of Humans*, pages 354–363. Springer, 2007.

[47] Annamaria Mesaros, Toni Heittola, Antti Eronen, and Tuomas Virtanen. Acoustic event detection in real life recordings. In *2010 18th European Signal Processing Conference*, pages 1267–1271. IEEE, 2010.

[48] Daniel D Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, pages 556–562, 2001.

[49] Toni Heittola, Annamaria Mesaros, Tuomas Virtanen, and Antti Eronen. Sound event detection in multisource environments using source separation. In *Machine Listening in Multisource Environments*, 2011.

[50] Onur Dikmen and Annamaria Mesaros. Sound event detection using non-negative dictionaries learned from annotated overlapping events. In *2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 1–4. IEEE, 2013.

[51] Oguzhan Gencoglu, Tuomas Virtanen, and Heikki Huttunen. Recognition of acoustic events using deep neural networks. In *2014 22nd European Signal Processing Conference (EUSIPCO)*, pages 506–510. IEEE, 2014.

[52] Naoya Takahashi, Michael Gygli, Beat Pfister, and Luc Van Gool. Deep convolutional neural networks and data augmentation for acoustic event detection. *arXiv preprint arXiv:1604.07160*, 2016.

[53] Miquel Espi, Masakiyo Fujimoto, Keisuke Kinoshita, and Tomohiro Nakatani. Exploiting spectro-temporal locality in deep learning based acoustic event detection. *EURASIP Journal on Audio, Speech, and Music Processing*, 2015(1):26, 2015.

[54] Miquel Espi, Masakiyo Fujimoto, Yotaro Kubo, and Tomohiro Nakatani. Spectrogram patch based acoustic event detection and classification in speech overlapping conditions. In *2014 4th Joint Workshop on Hands-free Speech Communication and Microphone Arrays (HSCMA)*, pages 117–121. IEEE, 2014.

[55] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. IEEE, 2013.

[56] Yun Wang, Leonardo Neves, and Florian Metze. Audio-based multimedia event detection using deep recurrent neural networks. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2742–2746. IEEE, 2016.

[57] Giambattista Parascandolo, Heikki Huttunen, and Tuomas Virtanen. Recurrent neural networks for polyphonic sound event detection in real life recordings. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6440–6444. IEEE, 2016.

[58] A. Graves and J. Schmidhuber. Framewise phoneme classification with bidirectional lstm networks. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 4, pages 2047–2052 vol. 4, July 2005.

[59] T. Hayashi, S. Watanabe, T. Toda, T. Hori, J. Le Roux, and K. Takeda. Blstm-hmm hybrid system combined with sound activity detection network for polyphonic sound event detection. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 766–770, March 2017.

[60] S. Kothinti, K. Imoto, D. Chakrabarty, G. Sell, S. Watanabe, and M. Elhilali. Joint acoustic and class inference for weakly supervised sound event detection. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 36–40, May 2019.

[61] A. Pankajakshan, H. L. Bear, and E. Benetos. Polyphonic sound event and sound activity detection: A multi-task approach. In *2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 323–327, Oct 2019.

[62] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In *Advances in neural information processing systems*, pages 3856–3866, 2017.

[63] Yanxiong Li, Mingle Liu, Konstantinos Drossos, and Tuomas Virtanen. Sound event detection via dilated convolutional recurrent neural networks, 2019.

[64] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions, 2015.

[65] Konstantinos Drossos, Stylianos I. Mimilakis, Shayan Gharib, Yanxiong Li, and Tuomas Virtanen. Sound event detection with depthwise separable and dilated convolutions, 2020.

[66] Jianbo Guo, Yuxi Li, Weiyao Lin, Yurong Chen, and Jianguo Li. Network decoupling: From regular to depthwise separable convolutions. In *British Machine Vision Conference (BMVC)*, 2018.

[67] Shayan Gharib, Konstantinos Drossos, Eemi Fagerlund, and Tuomas Virtanen. Voice: A sound event detection dataset for generalizable domain adaptation. *arXiv preprint arXiv:1911.07098*, 2019.

[68] Hyoungwoo Park, Sungrack Yun, Jungyun Eum, Janghoon Cho, and Kyuwoong Hwang. Weakly labeled sound event detection using tri-training and adversarial learning. *arXiv preprint arXiv:1910.06790*, 2019.

[69] Kuniaki Saito, Yoshitaka Ushiku, and Tatsuya Harada. Asymmetric tri-training for unsupervised domain adaptation. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2988–2997. JMLR. org, 2017.

[70] Wei Wei, Hongning Zhu, Emmanouil Benetos, and Ye Wang. A-crnn: A domain adaptation model for sound event detection. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 276–280. IEEE, 2020.

[71] Ehsan Hosseini-Asl, Yingbo Zhou, Caiming Xiong, and Richard Socher. Augmented cyclic adversarial learning for low resource domain adaptation. *International Conference on Learning Representations (ICLR)*, 2019.

[72] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.

[73] Pablo Zinemanas, Pablo Cancela, and Martín Rocamora. Mavd: A dataset for sound event detection in urban environments. 2019.

[74] T. M. M. P. V. Engelshoven A. S. V. D. Zwaag, J. O. Y. Huisman and Y. R. J. M. V. Engelshoven. Road surface and vehicle classification. 2018.

[75] P. R. van Laar. Acoustic recognition of motorized road vehicles. *Master Thesis, TU Delft*, 2019.

[76] Annamaria Mesaros, Toni Heittola, and Tuomas Virtanen. Metrics for polyphonic sound event detection. *Applied Sciences*, 6(6):162, 2016.

[77] Douglas O'shaughnessy. *Speech Communications: Human And Machine (ieee)*. Universities press, 1987.

[78] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[79] Yarin Gal and Zoubin Ghahramani. A theoretically grounded application of dropout in recurrent neural networks. In *Advances in neural information processing systems*, pages 1019–1027, 2016.

[80] Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder–decoder approaches. *Syntax, Semantics and Structure in Statistical Translation*, page 103, 2014.

[81] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[82] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[83] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[84] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[85] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[86] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[87] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010.

[88] Ehsan Hosseini-Asl, Yingbo Zhou, Caiming Xiong, and Richard Socher. A multi-discriminator cyclegan for unsupervised non-parallel speech domain adaptation. *arXiv preprint arXiv:1804.00522*, 2018.

[89] Daniel Griffin and Jae Lim. Signal estimation from modified short-time fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(2):236–243, 1984.

[90] John S Garofolo. Timit acoustic phonetic continuous speech corpus. *Linguistic Data Consortium, 1993*, 1993.

[91] Ulf Sandberg, Luc Goubert, and Piotr Mioduszewski. Are vehicles driven in electric mode so quiet that they need acoustic warning signals. In *20th International Congress on Acoustics*, 2010.

[92] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[93] Michael Brandstein and Darren Ward. *Microphone arrays: signal processing techniques and applications*. Springer Science & Business Media, 2013.