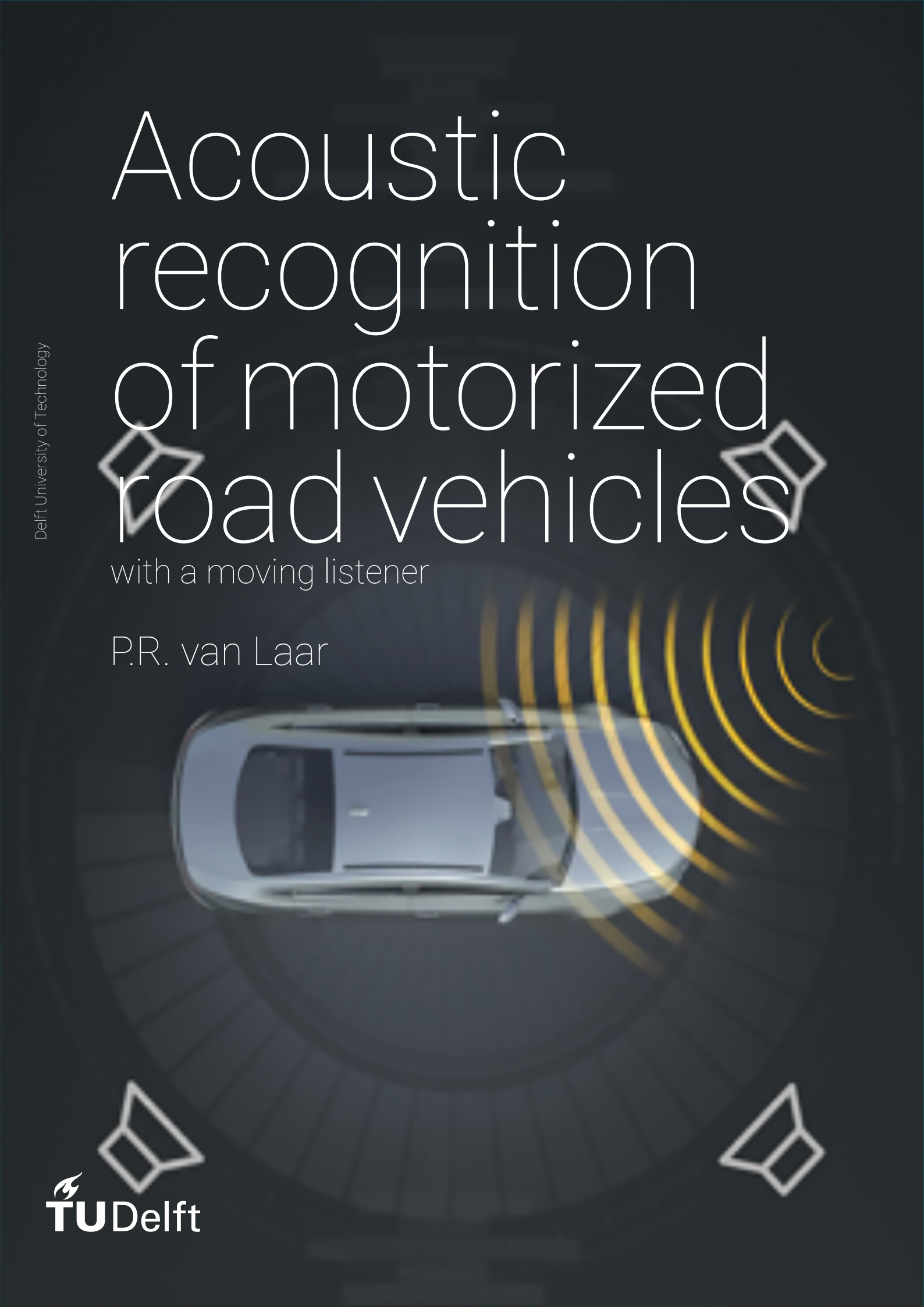
Acoustic recognition of motorized road vehicles

with a moving listener

P.R. van Laar

Delft University of Technology



TU Delft

Acoustic Recognition of Motorized road vehicles from a moving listener.

by

P.R. van Laar

The image on the coverpage is taken from [1]

Student number: 4147251

Supervisors: Dr. J. F. P. Kooij, TU Delft
MSc. T. Hehn TU Delft

TU Delft

Contents

Abstract

New measures have to be taken to combat fatalities caused by traffic accidents. Intelligent vehicles have the potential to increase safety, but depend heavily on their automated perception ability. Acoustic perception, an unused sensing modality in this field, has potential for the detection of nearby vehicles, an ability both human drivers and autonomous vehicles could use assistance with. In this thesis two existing datasets, AudioSet a large general purpose dataset and RoadCube a small dedicated vehicle recognition set, are evaluated. Furthermore commonly used acoustic features and classifier algorithm are evaluated. Special attention is given to the influence of a moving listener vehicle on the performance. For the evaluation a new dataset, DriveSound, is captured. It contains samples captured from a listener car, both when its moving or idle. Results show that RoadCube can be used for the detection of road vehicles, but only when the listener is idle. The best performing classifier from RoadCube, a Gaussian Mixture Model classifier surpassed classifiers trained on the evaluation dataset itself with a Matthews Correlation Coefficient (MCC) of 0.34. None of the classifiers performed well on the samples captured by a moving listener, except for the DriveSound-driving classifiers. The Support Vector Machine trained on this dataset attained a MCC of 0.56.

Acknowledgements

First of all I want to thank my supervisors, Julian Kooij and Thomas Hehn, for all the time and effort they spent guiding me. They were always available if I had a question or wanted to discuss something. Especially Thomas, who I bothered many times, was always available for a brainstorm or discussion. These sessions were one of the foremost reasons I learned many new things. Secondly I want to express my gratitude for the opportunity to do my thesis at the Intelligent Vehicles department. In this way I got a subject more suited to my interests, than I would have gotten at Biomechanical engineering. I also got the freedom to investigate the parts I find interesting. Of course it was a really bonus to get the opportunity to mount record sounds with a real intelligent vehicle.

Thanks to the PhD's and master students graduating at the Intelligent Vehicles section of the Cognitive Robotics department, where I worked on my thesis. You work on a thesis individually, but it often felt that we had a same goal. You kept me motivated. Finally thanks to Ronald Ensing and Frank Everdij who helped me multiple times setting up a remote simulation.

1 Introduction

Drivers on the road are more and more assisted by their intelligent vehicles, which are able to perceive the surroundings. It is a hot topic and many companies are working on driving automation. Many companies are researching fully autonomous vehicles for example, including traditional car manufacturers: Volvo [1], VW and Hyundai [2], Peugeot [3], Toyota [4] and others like Apple [5], Uber [6] and Alphabet's Waymo [2]. The intelligence in these modern road vehicles is mainly provided by ADASs and automated driving systems. The first type of system solely enhance the drivers situational awareness like a lane-departure warning system (example in [7]) and the other type automates one or more of the driving tasks like adaptive cruise control (example in [8]). There are thus large differences between the systems in the amount of automation provided (more details in section 2.1.1: *Autonomy levels of ADAS*). The driving tasks automated, currently or in the near future, range from pedestrian avoidance (example in [9]) to parking assistance (example in [10]). Automated perception is a crucial step for all these kinds of systems, regardless of task or level of automation.

Intelligent vehicles mainly aim to make driving safer, because road safety is still a big issue. Previous approaches focussed mainly on passive safety improvements like airbags, which lessen the impact of accidents [11]. Intelligent vehicles influence the actions to perform driving tasks, thus can prevent accidents. Despite the passive and previous active advancements on vehicle safety, traffic accidents still cause many deaths and injuries annually. There is thus still much to be gained by additional measures, including new types of automated driving systems. Each of these automated driving systems requires automated perception, which is the subject of this thesis. Its scope is limited to vehicle detection, because they are often involved in severe traffic accidents. These detections could thus be used to increase safety significantly. Section 1.1 goes further into detail on the current safety levels on the roads and the choice for a automated vehicle recognition system.

Computer vision is the most often used sensing modality, i.e. input type of the sensor, in existing vehicle detection systems and those currently under development [12], [13]. Passive acoustic sensing on the other hand, is not used yet in the context of intelligent vehicles, despite its key advantages like its ability to detect occluded vehicles. Examples of locations where vehicles could be occluded can be seen in figure 1.1. A complete list of benefits can be found in section 1.2. A passive acoustic motorized vehicle recognition system mounted on a listener vehicle, will be able to utilise the benefits of acoustics to reduce the amount of accidents involving vehicles, which in turn can improve safety significantly. This thesis evaluates this type of system and evaluates the components of this acoustic perception system: datasets, features and classifiers. Pedestrians and cyclists are not targeted, because it is assumed they produce too little sound consistently to be detected [13]. No acoustical detection systems like the one proposed here, is currently used on road vehicles [13] yet.

This chapter continues with more detailed statistics about traffic safety in section 1.1, followed by a section about the potential of acoustic perception 1.2. At the end of the chapter the topic of this thesis, the acoustic recognition of motorised vehicles, is formalised into research questions in section 1.3.

1.1. Traffic safety

Over 1.2 million people die each year on the roads in the world, while millions more are sustaining serious injuries [17]. Traffic accidents are a leading cause of death among young people and the main cause of death

[1] Volvo - https://www.volvocars.com/intl/about/our-innovation-brands/intellisafe/autonomous-driving
[2] Waymo - https://waymo.com/tech/

Figure 1.1: Examples of locations where visibility is limited
left: Mountainous road with a low visibility corner (Source: [14]), **middle:** Urban road with buildings blocking the view close to the intersection (Source: [15]), **right:** parked vehicle blocks visibility. (Source: [16]).

among those aged 15–29 years globally [17]. In the European Union (EU), where relative to other areas few traffic deaths occur, accidents still accounted for more than 25.000 fatalities and 213.000 serious injuries in 2018 [18], [19]. The current heavy usage of the roads and a possible future increase in road usage can exacerbate the amount of traffic accidents even further [20]. Improving road safety has thus the potential to save many lives and prevent many injuries. For the remainder of this thesis the focus will be on improving safety in Europe, but a solution there can probably also increase safety worldwide. For this reason ambitious goals where set by the EU to reduce the amount of road deaths by 50% by the year 2020, with respect to the 2010 levels [18], [19], [21]. Multiple safety initiatives, including additional safety systems on vehicles, have let to somewhat safer roads. Car manufacturers have made their vehicles safer by for example adding an electronic stability control system or a lane departure warning system to their vehicles [12]. Governments have enacted different policies to reach the EU goals, including introducing more low-speed enforcing infrastructure, examining high risk sites and enforcing alcohol limits [19], [21]. The amount of accidents however, which have resulted in deaths or serious injuries, is stagnating in the last years in large parts of the EU [18]. In the last four years only a total reduction of 3% of fatal accidents was achieved, while the goal was a decrease of more than 30% [18], [19]. This gap between reality and the reduction goal makes the 2020 goal almost unobtainable. The gap between the aimed reduction of road deaths and the real traffic fatalities can be seen in figure 1.2. Clearly current approaches are not effective enough thus new, different measures must be taken [18], [19].

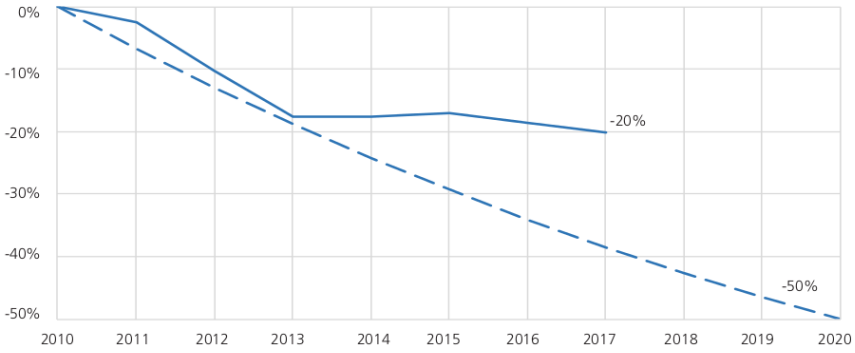


Figure 1.2: Reduction of the traffic fatalities in the EU. The dotted blue line depicts the 50 % reduction goal and the full blue line shows the actual reduction achieved. Source: [19]

In the EU motorized vehicles, especially cars, are involved in the majority of traffic accidents. This is indicated by reports of individual member states. In the Netherlands during 2015, for example about 74% of the traffic accidents and 76% of the fatal traffic accidents involved cars [22], [23]. Additionally Denmark stated that vehicle collisions are a major factor in the increase in deaths in 2016 [18]. A study examining the causes of traffic accidents in some of the EU countries [24] found that 72% of the investigated crashes involved causes related to the driver. This is consistent with earlier research [20] which stated that the behaviour and decision-making of drivers are key influences on road safety. At the same time 46% of the traffic fatalities were car occupants in 2015 in the whole EU, which translates to 12,090 people [25]. It is a significant number, but

if it is compared tot the percentage of accidents with cars involved, then it can be concluded that other traffic participants, like pedestrians and cyclists, are also often a victim to accidents caused by vehicles.

A machine perception system can be both be used to assist the driver or to automate tasks. There are three main components of driver's behaviour which can be assisted by a system like this: situational awareness, hazard perception and inattention. Firstly situational awareness is a crucial factor for a driver's performance [26], Hazard perception ability is a skill, which is only developed slowly with experience. Thus novice drivers in particular have difficulties with it [27]. As young drivers are a major cause of traffic accidents [18], enhancing this skill can increase safety significantly. Inattention is a also large problem with human drivers. A large study into driver behaviour ([28]) found that 93% of the rear-end crashes researched, involved inattention of the driver as a contributing cause. A machine perception system, which is able to detect vehicles nearby can assist the (automated) driver's situational awareness and hazard perception abilities, without suffering from inexperience or inattention. As vehicles are involved in the majority of accidents in the EU [18], an increased ability to detect them can improve safety significantly. A critical part of this awareness is knowing the location and type of other nearby, relevant actors. The relevance is in this case dependent on the speed, type and distance to the observer. A heavy truck approaching quickly is of more concern than a cyclist riding parallel to the observer. This thesis is about a system like the one proposed above. In the next section the potential of the acoustic sensing modality for this kind of system is presented.

1.2. The potential of passive acoustic perception

Passive acoustic perception aims to capture the sounds emitted by sources in the environment (shown in figure 1.3), as opposed to active acoustics (sonar), which emit a signal and catch their reflection. For this reason passive acoustic methods can reveal properties of the object emitting the sound, instead of only its location. Therefore the field can be divided into two types of problems: recognition and localisation. Recognition aims to detect certain types of sounds including vehicles. Often machine learning methods are used for this type of problem. Localisation on the other hand, aims to the estimate the direction and or distance to the sound source. This is necessary due to the omni directionality of the microphones. Array processing methods are often used for this type of problem, which combine the inputs from multiple microphones. There are some examples of machine learning being used for localisation as well, sometimes in conjunction with array processing [13]. In this thesis only the recognition problem is studied.

Figure 1.3: A car driving past the white listener vehicle. The sound emission from the blue target vehicle is captured by the microphones mounted on the listener vehicle.

Passive acoustic perception has some properties, which can be useful for road vehicle detection. Initial research has shown the following advantages in comparison with computer vision:

- omni-directional sensing [29]–[31]

- ability to detect occluded vehicles [32]
- robustness to diverse weather conditions [29], [33], [34]
- cheap sensors [35], [36]
- low data transmissions [37]

Firstly the ability to sense in every direction (omni-directionality) is important because vehicles could be coming from every direction. No additional sensors are required to detect vehicles from all directions, but to determine the location of the vehicle multiple sensors need to be used together. Secondly vehicles occluded by parked cars or buildings can be detected. On for example a cross-roads with limited visibility due to buildings, the sound of approaching vehicles could be detected without seeing it. Robustness against diverse weather conditions is also an important property, because this system needs to function outdoors. Microphones are furthermore cheap and send only a limited amount of data. Instead of using acoustics as sole perception modality, it can also be combined with other modalities to complement each other [38]. The robustness of passive acoustics to diverse weather conditions can for example be utilised in a combined modality system.

Humans already use their hearing in traffic by for example listening to approaching vehicles. For car drivers, as opposed to pedestrians and cyclists, this is much more difficult due to the insulation of in-car environments against external sounds [39]. An automated acoustic perception system is thus required for cars to use the earlier listed benefits. Passive acoustic vehicle detection is a unexplored modality for the use in vehicles in comparison to vision-based traffic perception, despite its potential advantages for this application. The lack of a clearly defined benchmark dataset, as the MNIST digits database for vision (proposed in [40]), is an indication of this. It makes it harder to compare algorithms and systems directly. This thesis evaluates commonly used acoustic perception methods for road vehicle recognition. The scope and research question will be expanded on in the next section.

1.3. Research questions

In this thesis commonly used machine learning methods, for other acoustic perception applications, are evaluated for vehicle detection. Both general purpose classifiers and acoustic feature detection algorithms are evaluated. The following question is central to the research:

How well can the acoustic signatures of motorized road vehicles, captured by vehicle mounted microphones, be discerned from other traffic and environmental sounds in an suburban environment using machine learning methods?

Different datasets, features and classifiers are evaluated for this problem. Two existing datasets are used to train the classifiers: Google's AudioSet and RoadCube. AudioSet is a large scale, general purpose dataset, while RoadCube solely contains specifically captured vehicle sounds at a much smaller scale. For the evaluation of these datasets a new dataset is captured: DriveSound. The aim is to determine which relevant information is present in the datasets and which tools are able to capture and use this information for the prediction of present vehicles. Due to the small size of the RoadCube and the fact that the problem is not well understood yet, a shallow classification approach is taken. Commonly used acoustic features and general purpose classifiers are use for this problem. As in many learning problems the art is to select the best performing tools, i.e. features and classifiers. Each method works in a different way and might be able to capture different pieces of information from the sounds. The performance difference between these methods is investigated. Another key aspect is the self motion of the listener vehicle. When the listener vehicle itself is moving, both the relative velocities of the targets relative to the the listener and the absolute velocities of the target vehicles are of interest. The acoustic signature of vehicle is namely dependent on the Vehicles travelling in opposite direction for example, have a high speed relative speed. The significant effects on the detection performance are investigated. It is expected that the general performance will be lower and that the self-motion will introduce additional noise, most notably wind noise.

The aim of this thesis is to find the dataset, feature space and classifier, which is be able to determine acoustically when there are vehicles being driven in the neighbourhood of the the listener vehicle. When a vehicle is being driven on a road, other road users, especially vehicles, within a distance R of the listener vehicle are of interest. This is shown in the drawing of figure 1.4, where this area of interest is drawn as a circle with radius R. Inside the circle vehicles can be detected by their acoustic signature, while sound emitted from outside the circle is considered noise. The target vehicles can have different types of motion inside the circle, including driving with constant speed, accelerating, braking and cornering. This can have an effect on the detection performance, because different motions produce different acoustic signatures. Additionally

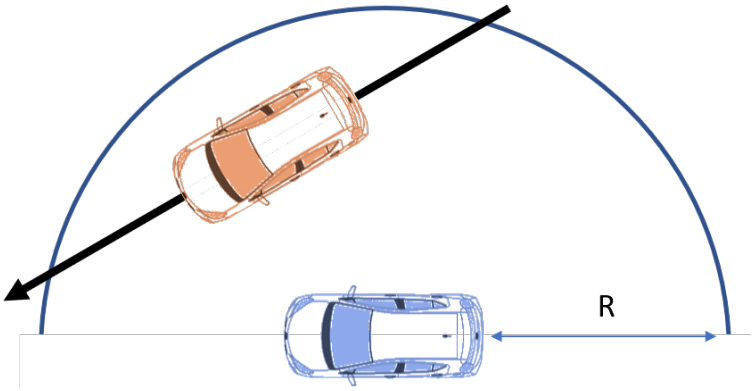


Figure 1.4: Drawing area around the listener vehicle. Target vehicles inside the circle with radius R should be detected, while vehicles outside the circle are considered as part of the environmental noise. In this thesis it is assumed that target vehicles are driven with constant speed. (The car graphic is taken from [41])

specific components may be used differently during different motion types. During cornering, the tyres will make a different contact with the road for example, which might produces a different sound. For now, the problem is limited to vehicles having a near constant speed and travel in a straight line, during their motion in the detectable radius R. The detection algorithm will aim to detect the moment a vehicle is closest to the listener vehicle.

Next steps would be to detect the vehicle on every moment in the circle, generalise to other types of motion of both the target vehicles and the listener vehicle. Additionally the type of the vehicle and the amount of vehicles present in R can be detected. Finally localisation methods can be added as pre-processing step to find the direction and ultimately the location of the target vehicles.

This report continues with an overview of already performed experiments of acoustic vehicle detection and some related fields (chapter 2: *Related work*). Subsequently the methods and procedures used in this thesis are discussed in the chapter 3: *Method*. Elaboration on the datasets is done in a separate chapter, namely chapter 4: *Datasets*. The results of the machine learning experiments are discussed in chapter 5: *Experiments* and the conclusions are drawn in chapter 6: *Conclusion*.

2

Related work

Intelligent vehicles are broadly researched and machine perception systems provide a crucial role. Acoustic perception however, is not used yet in commercial systems [13]. This chapter gives a reasoning why and how acoustic perception can be used for the detection of road vehicles. First and an overview of the current uses of traffic perception for intelligent vehicles systems is given in section 2.1: *Perception in automated driving*. It includes a listing of the specific challenges of traffic perception and a comparison is made of different sensing modalities with their advantages and disadvantages for this field. Here it is shown that acoustic perception has some useful properties. Secondly section 2.2: *Passive acoustic vehicle detection* contains more details on the characteristics of sound and sound emissions by road vehicles. Thirdly an overview is given on acoustic experiments performed, including vehicle detection but also robot audition. Finally the contribution of this thesis is elaborated. The author already did a literature review on the acoustic classification and localisation of road traffic in [13]. Relevant parts have been reused here.

2.1. Perception in automated driving

The systems employed in intelligent vehicle rely for a large part on machine perception [12]. Some of these ADASs send additional information to the human driver, which augments their situational awareness. These warning systems and other perception enhancers, allow the driver to perform better on the driving task. Current examples of employed in cars include: pedestrian detection (machine vision) [42], parking trajectories by using a rear view camera (machine vision) [10] and parking sensors (sonar) [43]. Another example are night driving systems [11], [12], which can improve the driver's collision avoidance at night. Other types of ADAS fully automate a certain driving task, for example in [44]. In this case the driver does not take the decisions and actions of that task any more. Collision avoidance systems, which brake when another vehicle or pedestrian comes too close, are an example of this. Autonomous vehicles automate all of the driving tasks normally performed by humans (example in [45]). An example of a semi-autonomous system is highway platooning, where vehicles use a control system to keep distance to each other on the highway. It is expected to reduce congestion, because they increase the throughput of the road and reduce the risk [46]. Regardless on the level of autonomy, all these types of systems are dependent on machine perception.

In the next sections the automation levels of the different systems is discussed. Afterwards subsection 2.1.2 compares the different sensing modalities for usage in traffic perception.

2.1.1. Autonomy levels of ADAS

Many different types of ADAS are employed in intelligent vehicles. An overview of the different types of systems can be seen in figure 2.1. One dimension these systems are differ in is the automation level they provide on a certain task. Two models are used here to describe the level of automation in a car: Sheridan's [47] and the Society of Automotive Engineers (SAE) [48]. Sheridan's modes of human automated control modes can be seen in figure 2.2. The control modes are for executing a single task. The spectrum ranges from no computer involvement in the feedback loop to fully automatic control. Manual control can be both without an computer or with computer aided perception. Supervisory control can be cooperative control between the human and the computer or the human only intervenes in certain scenarios. Finally with fully automated control the computer only displays the current state to the human, while there is no way

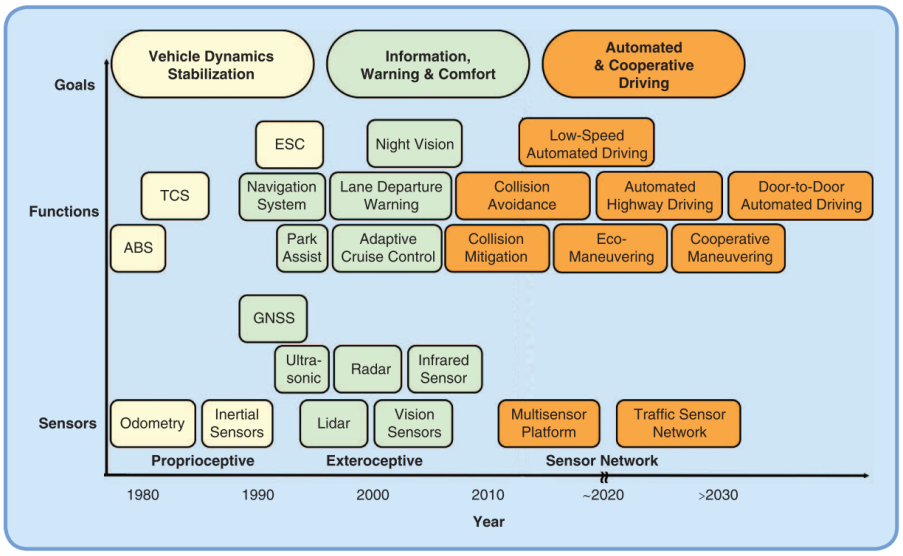


Figure 2.1: Timeline for the development of different types of ADAS systems. The development is moving towards cooperative and automated driving. The development predictions were made in 2014. Source: [12]

to intervene. The model from the SAE is focussed on the automation level of the dynamic driving task. It defines the autonomy levels of a vehicle. The different levels of autonomy range from no automation of this task to full automation where every aspect is automated in every condition.

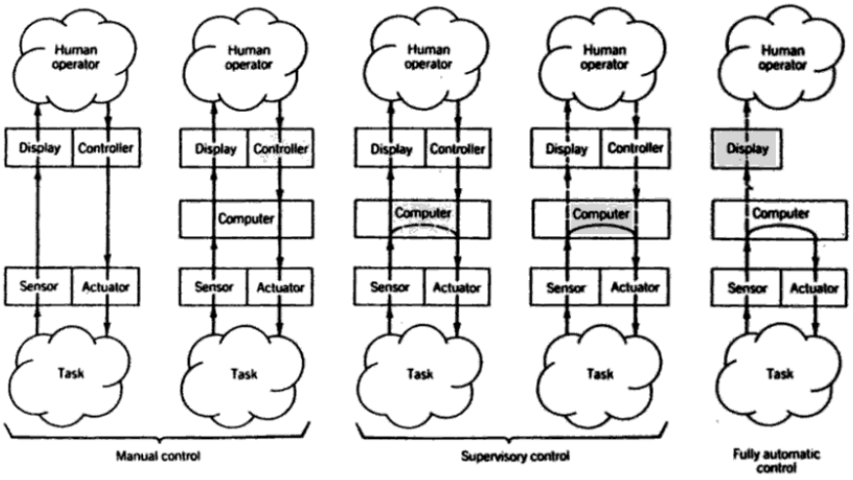


Figure 2.2: Different feedback modes for computer aided control. On the left side no computer is involved, while on the right side the process is fully automated. Source: [47]

As mentioned before ADAS [12] assist the driver in the execution of a task by augmenting their situational awareness. These systems correspond to the second version of manual control in the Sheridan model (figure 2.2) and SAE level zero (no automation, figure 2.3). These include warning systems and other perception enhancers. Another subset of ADAS can also intervene when necessary. This corresponds with the first level of supervisory control in the Sheridan model (figure 2.2) and SAE level zero (no automation, figure 2.3). ADAS which continually performs actions to assist the driver are the right model of supervisory control in the Sheridan model (figure 2.2) and SAE level 1 (driver assistance, figure 2.3) if they control brakes and/or steering.

In autonomous vehicles (example in figure 2.4) the driver is taken out of the loop and the vehicle will make decisions. This corresponds to fully automated control in the Sheridan model [47]. It would correspond to SAE levels 2 to 5 depending on the scenarios the automation enabled.

In systems perceiving the surroundings and conveying this information to the driver, the driver still takes

SAE level	Name	Narrative Definition	Execution of Steering and Acceleration/ Deceleration	*Monitoring* of Driving Environment	Fallback Performance of *Dynamic Driving Task*	System Capability *(Driving Modes)*
***Human driver* monitors the driving environment**						
0	No Automation	the full-time performance by the *human driver* of all aspects of the *dynamic driving task*, even when enhanced by warning or intervention systems	Human driver	Human driver	Human driver	n/a
1	Driver Assistance	the *driving mode*-specific execution by a driver assistance system of either steering or acceleration/deceleration using information about the driving environment and with the expectation that the *human driver* perform all remaining aspects of the *dynamic driving task*	Human driver and system	Human driver	Human driver	Some driving modes
2	Partial Automation	the *driving mode*-specific execution by one or more driver assistance systems of both steering and acceleration/ deceleration using information about the driving environment and with the expectation that the *human driver* perform all remaining aspects of the *dynamic driving task*	**System**	Human driver	Human driver	Some driving modes
***Automated driving system* ("system") monitors the driving environment**						
3	Conditional Automation	the *driving mode*-specific performance by an *automated driving system* of all aspects of the dynamic driving task with the expectation that the *human driver* will respond appropriately to a *request to intervene*	System	**System**	Human driver	Some driving modes
4	High Automation	the *driving mode*-specific performance by an automated driving system of all aspects of the *dynamic driving task*, even if a *human driver* does not respond appropriately to a *request to intervene*	System	System	**System**	Some driving modes
5	Full Automation	the full-time performance by an *automated driving system* of all aspects of the *dynamic driving task* under all roadway and environmental conditions that can be managed by a *human driver*	System	System	System	**All driving modes**

Figure 2.3: Levels of automation of the dynamic driving task according to the SAE. Source: Lecture slides from Dr. Meng Wang, adapted from [48]

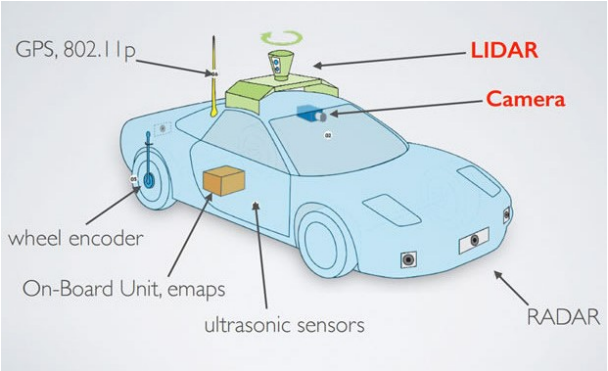


Figure 2.4: Examples of autonomous vehicle with various sensors. Source: [49]

the actions but the available information is augmented. The system will not take action by itself. They can be used to inform the driver continuously or only when a certain condition has been met. This can be an unexpected or changing situation or for example a dangerous incident. Drivers have a reaction time before they can act thus, especially for this kind of system, it is important to predict at least slightly into the future. Drivers mainly use their visual senses on the road. Continuous information is often conveyed visually, while auditory signal are more suited for high-priority warnings [50]. It is practical to keep the cue in the same perception domain as the original observation, because drivers expect it [39]. For an acoustic detection system this would mean that an acoustic cue is given when vehicle is detected very nearby or on collision course. In the remainder of this thesis the perception performance is discussed in isolation, thus without considering the feedback loop, because it will have impact on the design of the perception system.

2.1.2. Comparison of modalities

Many different sensing modalities are in use for traffic perception, including vision, LIght Detection And Ranging (LIDAR), radar, sonar [12]. They are used for various tasks:

- vision - pedestrian detection [42] or vehicle detection [51]
- LIDAR - autonomous parking [44]
- radar - vehicle classification [52]
- sonar - autonomous parking [43]

Out of these modalities computer vision is most widely used [12], [53]. Each of the modalities has very different properties, which can be helpful for some applications but a limitation in others. The detection range of the sensors is an example of this. These are shown in figure 2.9 along with a possible application. Combining multiple types of perception can complement the weaknesses of individual modalities and increase the robustness of the perception system. In the remaining paragraphs of this section the strengths and weaknesses are listed for each modality, including passive acoustic perception. An overview of these strengths and weaknesses is given in table 2.1.

	Active/Passive	directionality	Environmental robustness	Occluded objects
Acoustics	Passive	omni	robust against weather, light changes	yes, if the obstacle is not emitting sound
Lidar	Active	directional	very susceptible to weather and bright light	no
Radar	Active	omni	robust against weather, light changes	no
Sonar	Active	omni	robust against weather, light changes	no
Vision	Passive	directional	susceptible to weather and bright light	no

Table 2.1: Comparison of sensing modalities used for vehicle detection

Multiple advantages of acoustic perception exist for vehicle recognition. Microphones are omni-directional [29]–[31], cheap [35], [36], [54] and cost less energy to operate [55] in contrast to for example cameras [34]. Due to the omni-directionality, it is not necessary to know the source location to process sounds [34]. Occluded objects can be detected [29]–[31], [56], [57]. This last advantage over for example vision, can be helpful on a crossroads when nearby vehicles are occluded by a building so they cannot be seen, but can still be heard. This situation is depicted in figure 2.6. The obstacle, which blocks the view cannot have emit a sound themselves. Furthermore, this mode is robust against weather and light changes [29], [33], [34], [54], [58], which is helpful for operating in an outdoors environment. As it is a passive method, it is unintrusive to the surroundings [35], because no sounds are emitted which could be annoying to others. The final advantage of acoustic perception is that auditory processing will be less computationally demanding than vision [37]. On the other hand this modality is susceptible to random environmental noise and requires target vehicles to emit a sound.

LIDAR sensors emit beams of light and sense their reflection, thus it is a form of active sensing. It provides an accurate distance measurement It is very susceptible to bright light from for example the sun. It has a limited range, due to a relatively high attenuation of the light on contact [11]. Furthermore the light beams must be reflected back at the sensor instead of being absorbed or deflected [11]. A scanning laser can map the surroundings in a 3D pointcloud, usable for object detection.

Radar sensors emit radio waves, which typically have a frequency of 20 - 80 GHz, and capture the reflections. It is an active sensing method with omni-directional sensors [32]. It has a longer range, but a lower spatial

resolution than LIDAR [11]. They are unable to detect occluded obstacles [35]. A schematic example of radar used for the mapping of the environment can be seen in figure 2.5.

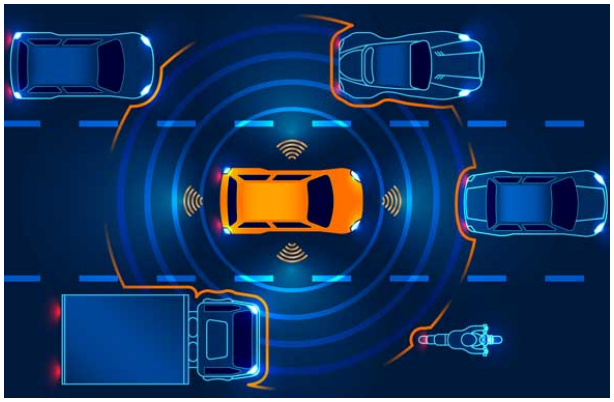


Figure 2.5: Intelligent vehicles which uses active sensing with radar to perceive the nearby traffic. Source: [59]

Sonar uses a sound emitter and captures the acoustic reflection. It is thus the active acoustical perception variant. This technique is use for parking sensors [12]. Sonar is able to detect objects which are not emitting sound themselves. It uses more energy than passive acoustics and has a lower range. Depending on which frequencies are emitted, the sound emittance can be a nuisance, especially when multiple vehicles are emitting at the same time, or it can affect other devices.

Vision is the most commonly used modality for traffic perception along with radar [12]. It is also a passive method, because camera's capture light coming from the environment. Vision has the primary advantages that much information can be gathered at once and that the data is well interpretable by humans. Cameras are directional however, so no information is gathered in the blind spot [32]. Furthermore, occluded objects cannot be detected, thus larger vehicles can mask smaller ones [35]. Weather conditions, lighting changes, caused by for example headlights, shadows and reflections can influence the observations [35]. Finally if colours and intensities match between objects and their surroundings, it can cause confusion [35]. Examples of vision based systems include night vision aids, rear view cameras and pedestrian detection systems.

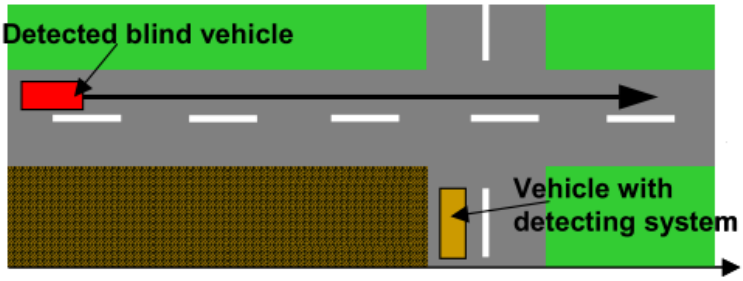


Figure 2.6: An example sketch of an urban crossroads, where a vehicle cannot be seen from the listener's perspective. Source: [32]

2.2. Passive acoustic vehicle detection

A sound wave is caused by a vibrating object producing waves which travel through the air or another medium. Subsequently these waves are captured by a microphone, which samples the air pressure relative to the atmosphere. In the case of road vehicles detection, the vibrating objects are multiple vehicle components, like wheels and the engine.

2.2.1. The characteristics of sound waves

To understand acoustic perception, relevant characteristics of sound waves and and some modelling assumptions must be understood. The relevant properties of sound waves are: the Doppler effect [60], diffraction [34], interference [61] and reverberation [62]. These are shown in figure 2.7. The Doppler effect, is caused by a difference in relative speed between the emitting source and the listener. The speed difference causes

acoustic waves to reach the listener slightly earlier or later, which shifts the frequency of the sound [60]. Sound is able to travel around obstacles. This is called diffraction and it is possible due to the wave characteristics of sound. It can change frequencies and energy contents of the signal [63]. Occluded objects can still be detected due to this characteristic [32]. Another wave characteristic is interference. When multiple waves meet at a location the the signals are summed. This can result in a strengthening or suppression of the signals [64]. If sound waves interact with a surface they can be reflected, which is called reverberation [65], [66]. Part of the energy is absorbed, so the reflection will be less potent than the original signal [67]. An environment with many reflective surfaces will result in many reflections. The listener will receive thus the same signal from multiple directions, which makes it more difficult to determine the direction of the source.

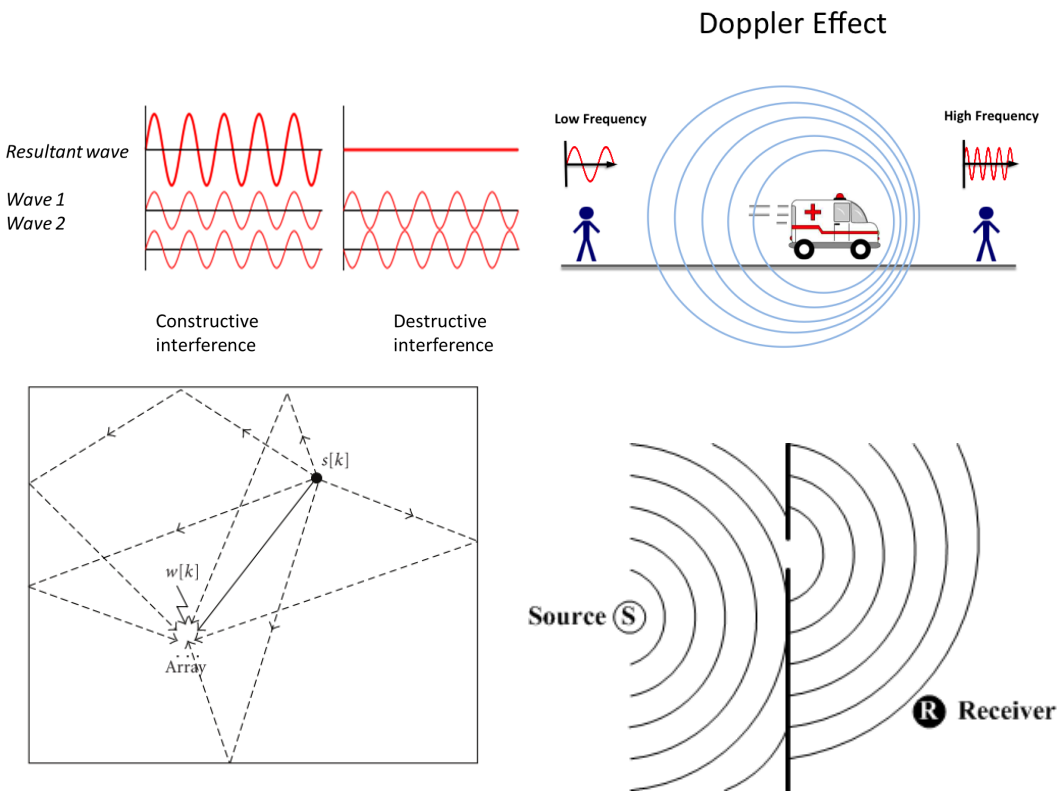


Figure 2.7: Images depicting different sound characteristics.
Top-left: Interference: when a wave reach a certain point simultaneously, the wave are summed. Source: [68],
Top-right: Doppler effect: a source moving away from the listener is heard at a lower frequency and vice versa for movement towards the listener. Source: [69],
Bottom-left: Reverberation: sounds can reflect of surfaces and travel in multiple paths to the target. Source: [66],
Bottom-right: Diffraction: Sound can circumvent obstacles and reach the listener when it cannot see the emitter. Source: [70]

Modelling assumptions can also be a factor in acoustic recognition. Common modelling assumptions about the wave propagation are the point source, the narrowband and the no attenuation assumptions. The first example, the point source assumption, simplifies the location of the sound source to be a single point. The narrowband assumption expects that a sound emission only has significant power around a certain frequency. This is called a narrowband signal, as opposed to broadband. Subsequently only the dominant frequency of the signal is considered in further processing steps. Finally no attenuation assumes that no energy of the sound wave is lost over its trajectory.

A microphone measures the pressure difference in the air. The microphone transforms the audio capture to a discrete signal. The sampling frequency f_s is a critical property of the signal, because it gives the relationship between the time and frequency dimension of the signal. As opposite to taking a picture, sound is captured over time and not at a specific moment. This means that a moment in time can only be approximated by making the sound fragments very small. Longer fragments can have non-stationary effects like acceleration.

Furthermore noise is important to evaluate, both measurement noise and from non relevant sources [34]. An important metric for the strength of a signal is the signal-to-noise ratio. It is defined as $\frac{\mu}{\sigma}$, where μ is the

mean and σ is the standard deviation [71].

2.2.2. Acoustic vehicle signatures

The sound produced by a motorized vehicle is composed out of several components, which are emitted from different parts of the vehicle. Most of the sound from a moving vehicle comes from the engine, tyres, aerodynamics [56], [58] and intake system and exhaust [64]. Sirens of emergency vehicles are a special case, which can be specifically targeted (as is done in [39]). The signals from the different components differ in power, frequency or directivity [64]. The emitted sound signal from different vehicles may significantly vary, due to the vehicle speed, vehicle type, vehicle orientation to the receiver, it's technical condition or it's load [58], [60]. Vehicles of the same type however are emitting similar acoustic signals under identical conditions [58]. Noise types from the engine and aerodynamics depend strongly on the vehicle type [58]. Tyre noise does not [56], [58], which means that tyre noise is similar to tyre noise from another vehicle when travelling on the same road conditions.

Depending on the situation one or more components will dominate the other. According to [72], when the vehicle is travelling faster than 40km/h the tyre/road noise dominates, while [73] uses the rule of thumb of 50 km/h and higher. Furthermore aerodynamic noise is significant for speed over 100 km/h [60]. Sandberg [74] mentioned, in 2001, that for some more modern cars the tyre noise always dominates, without naming hybrid or electric vehicles specifically. More modern sources such as [72], [73] (2013 and 2015 respectively), don't mention this. This undercuts the claim, because it is expected that this effect is larger on even more modern cars in the future. On dirt roads, tyre noise is always dominant [60]. In the low frequency spectrum, a vehicle noise is dominated by the intake and exhaust noise [60]. Motorbikes are a special case, where the tyre/road noise is dominated by the exhaust noise [75].

Engine noise is emitted due to the pressure changes created during combustion. The main energy of this noise lies between 800 and 3000 Hz, but this was range was found for military vehicles. Road vehicles probably emit in a subset of this range, due to limited variance in engines. In the noise the cycling rate can be discerned in the form of fundamental frequencies. Engine speed changes result in a change of these spectrum peaks, and spectrum distance between those peaks [60].

The contact noise between tyres and the road is caused by resonance between the grooves of tyres and the road [56]. According to Asahi [56] its spectrum is usually distributed around 1000 Hz, while Gorski [60] only says its a has a high frequency characteristic. In [35] a bandpass filter is used within 940 and 1060, which also accounts for the Doppler effect. The vehicle speed, type of tyres and the road surface influences the emitted sound [58], [60]. The condition of the road, for example dry, wet, icy or snowy conditions, also produces different acoustic signatures [76], [77]. Alonso mentiones that these conditions can be detected with passive acoustics by using frequency features [72]. Uneven roads add a low-frequency part to this type of noise [60].

These systems are functioning outdoors, so there are a lot of sources of additional noise. Here the common sources are discussed, but there can be many occasional ones as well, such as construction sites. Rain and wind are sources of broad-spectrum ambient noise [58], [60]. According to Asahi [56] the frequencies of these forms of noise are mainly less than 500 Hz, while [60] states that the wind noise is the most noticeable in low frequencies, which is different from for example tyre/noise [56]. Thus it makes sense that ambient noise can be filtered by a hi-pass filter [58] or band-pass filter [56]. It is expected that wind noise is more present in open spaced areas or around tall structures. In urban areas more surfaces are present than in open space, which can lead to an increase in reverberation.

No examples were found of systems targeting non motorized traffic, i.e. pedestrians and cyclists. It is assumed that their sound emittance is too little and too inconsistent. The lack of engine noise is not the only cause for the omission. Other noise types, including tyre and aerodynamic noise, are only recognisable at higher speeds [13]. This makes detecting cyclists infeasible, but electric cars a possibility.

2.2.3. Challenges of acoustic traffic perception

Passive acoustic traffic perception with a moving listener vehicle has some specific challenges, due to the outdoors environment and the motion of the listener vehicle. Operating outdoors requires handling unpredictable environments and weather conditions. Road vehicles have to function in all kinds of environments, ranging from plains to mountains and from forests to densely populated cities. Therefore the variety in environmental conditions can be large, including building density, temperature, humidity and type of road surface. The difficulty of uncertain environments is illustrated by the choice of predictable environment for current tests with self-driving cars [78], because the vehicles can not handle random environments well [12], [45]. The weather will make scenario's even more unpredictable (examples in figure 2.8). Rain for example affects

almost all sensors: it reduces visibility [79], it causes additional noise [60] and it disrupts lidar [33]. Acoustic perception can be made robust against many weather conditions, including rain and bright light [29], [33], [34]. Unpredictable noise sources, for example caused by roadside construction, could also be a problem. [34]. It can introduce noise which in turn lowers the signal to noise ratio, which can be problematic. Two possible solutions to this problem, segmenting the sounds from a single source and reducing the noise are currently open problems [13]. Finally when a sound source is obstructed by another sound emitter, it is difficult to discern them [58].

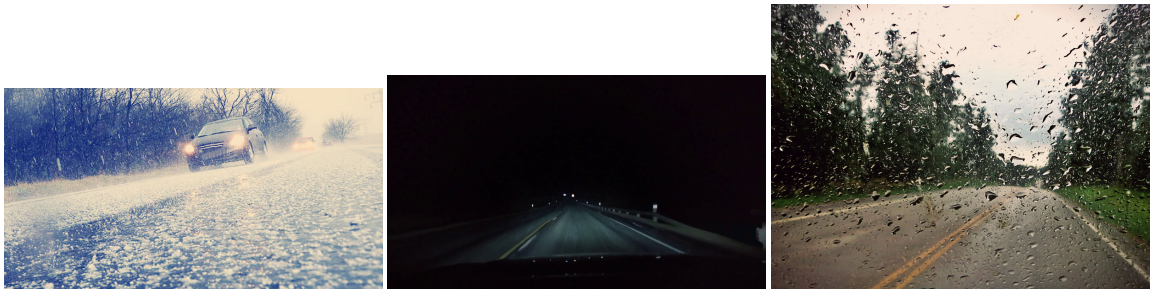


Figure 2.8: Examples of weather conditions, which affect the capabilities of sensors. **left:** hail. Source: [80], **middle:** darkness inside a tunnel and bright light at the end Source: [81], **right:** heavy rain. Source: [82]

Another big challenge is the motion of both the target and the listener vehicles. The captured acoustic signature of vehicles are dependent on both the target vehicle's velocity and the relative velocity with the listener vehicle [58], [60]. The absolute target velocity will influence the acoustic signature, while the relative velocity influences the transmission of the sound through the air. The relative velocity is important, because it will change the received sound waves [56], [65]. Due to the high speeds of vehicles it can thus be expected that an oncoming vehicle's signature will differ from an overtaking vehicle. The motion type of the target vehicles, i.e. driving with constant speed, braking, accelerating or cornering, influences the acoustic signature as well [83]. Another difficulty is that multiple sources must be detectable simultaneously. Apart that the sounds can interfere with each other, localisation algorithms must identify the amount of sources and might need to track them, which are no trivial tasks [38]. The speed differences put a strong real-time constraint on the solutions. If the perception system operates too slowly, it might miss vehicles or report them too slowly to act on the information. Additionally the system must be operational without manual calibration on the environment or use other prior knowledge. Due to the real-time constraint the length of the sound fragments used as input is limited. The hardware running the perception algorithm including sensors, must be implemented in or on a car. This restricts the size, power consumption and cost of the system.

2.3. Related experiments

Although no commercial acoustic vehicle perception systems mounted on a vehicle exists, a few experiments with these systems have been performed for research. In this section these experiments are discussed along with experiments from adjacent fields: fixed position vehicle detection, military vehicle detection and robot audition systems. First the road vehicle recognition systems are listed in subsection 2.3.1, starting with the those with microphones placed at a fixed position. Subsequently the vehicle mounted systems are discussed. In subsection 2.3.2 the recognition systems targeting military vehicles are listed. Finally a few robot audition examples are showed in subsection 2.3.3.

2.3.1. Road vehicle recognition

[83] aims to determine certain properties of the vehicle from their acoustic emission, including vehicle's length and width, number of cylinders and engine revolutions per minute. The feature vector obtained is specific to the make and model of the vehicle. The authors claim that it can be used for classification of the different vehicle models, but no classification experiment is performed. A single microphone is used for the recording and each vehicle is considered as a dipole sound source to take the interference between the tire emissions into account.

Nooralahiyan [37] uses a neural network for a four class classification problem. The classes were buses, cars, motorcycles and vans. Linear Predictive Coding (LPC) parameters are used as features. Fragments with 25 frames are processed at a time and a sliding window approach with 80% overlap is used to divide the

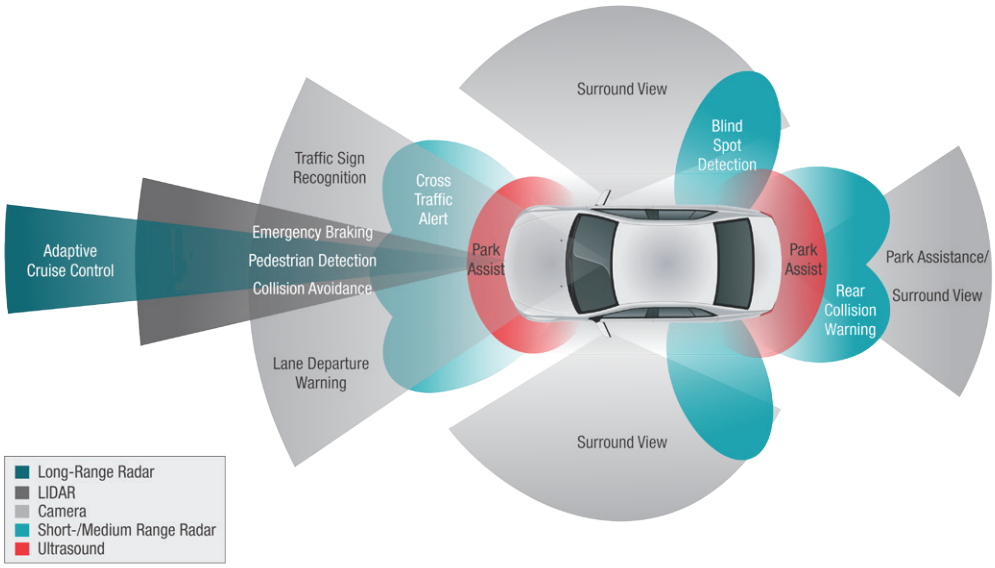


Figure 2.9: Examples of ADAS with the used modality and operating ranges. Source: [84]

signal into fragments. The audio was captured at an empty airstrip and on urban public roads. In the former scenario, much less noise was present. The urban area required more training iterations and performed worse on classification. It must be noted that this system was proposed in 1998 and might therefore be less effective than current methods.

Wu [85] uses frequency vectors as features. First the mean feature vector from the training set is substracted and then Principal Component Analysis (PCA) is used to reduce the dimensions of the remaining spectral variance vector. For each class the mean is calculated in the new projection space. Classification is then calculating which of the trained means is closest to the new sample.

The system from [86] uses classification to estimate the direction a vehicle is currently located. Four microphones are used to capture spectrograms. A SVM is used to classify the features calculated from the averages and medians of the spectrograms. It aims to perform classification and localisation is possible in a single step. It uses classes along a location-speed grid to estimate both. Adding vehicle type as a third dimensions to this grid can steeply increase the amount of involved classes, which in turn raise the amount training data required.

[57] aims to detect approaching vehicles. They utilise the principle that the energy level of the acoustic emission for approaching vehicles is positive due to the Doppler effect (more information about the Doppler effect in 2.2.2). The listener car, with a single microphone mounted on the right bottom wind shield, was idling on the side of the urban road during the experiments. It was not reported what kind of engine was in the listening vehicle and if it was turned on at that moment. Only vehicles separated by more than five signals from other vehicles were included. Five bandpass filters between 1kHz and 2kHz were used to create five channels. The signal was then compressed using a hair-cell function and a processed using a spike generator. A learning vector quantization neural network was used for classification. This type of network was chosen for its computational performance on specific hardware. The classification problem was binary with an approaching class and a negative class.

In [75] experiments were performed where the listener car was travelling between 50 and 100 km/h. Their program aimed to detect if sound frames contain sounds emitted by one or more vehicles. First the signals are filtered between 400 and 1200 Hz. Subsequently both frequency vector features and MFCC features are extracted. The extracted features are used by a neural network with ten hidden layers. Probably there is redundant information between these vectors, because MFCC features are also spectral features (see section 3.2 for more details) and in this case the frequency range is limited.

Instead of detection separate vehicles [36] is estimating the traffic density into three classes: light, medium and heavy. It uses Mel Frequency Cepstral Coefficients (MFCC) features and makes a comparison between a SVM-Radial Basis Function (RBF) and a GMM classifier. Alonso [72] aims to detect different road conditions, thus targets tyre/road noise. He uses the "relevant frequency components" as features. As tyre noise normally

has a frequency around 1000 Hz (see section 2.2.2), it is assumed that this is the area of interest in the frequency domain. The configuration of the SVM classifier is not mentioned. Afterwards the results are filtered by a spurious events filter to limit false positives. In the paper it is not explained what kind of filtering is used, but is probably uses some prior information about tire sounds. For example there tyre noise is located around 1000 Hz (see section 2.2.2), thus there must be a peak amplitude at that frequency. The system is specifically designed to be embedded in a car and runs in real-time.

2.3.2. Military vehicle detection

Military ground vehicles have different characteristics than regular road vehicles. Tracked vehicles are also present in this category for example. Additionally military vehicle can have tracks instead of wheels. They also operate mostly off-road, which changes the tire/ground contact noise significantly. Despite these differences there, method used for the acoustic detection of these vehicles can also be useful for road vehicles.

Gorski [60] compares three kinds of features in the context of vehicle detection: Harmonic line, Schur coeficients and MFCC features. Although it is not stated, it is assumed that they where extracted on a per frame basis. It concluded that Harmonic line features and MFCC feature perform well. It must be noted that the tests that the tests were done with military vehicles on dirt roads. This has two main effects. First of all engine noises can vary more between different military vehicles, than between road vehicles. As Harmonic line features are dependent on the engines. They might perform not as well with road vehicles. Secondly due to the gravel roads it could be assumed that the vehicles were travelling at a relatively low speeds. Therefore there is no dominant component source and there is a low frequency part added to the tyre noise (see section 2.2.2). The SVM classifier was used in a one versus all configuration (four classes). They were trained on undistorted signals of approaching vehicles.

2.3.3. Robot audition experiments

Passive acoustic perception is in use in other fields besides traffic detection. In robotics it is used to have conversations with humans [34] and is well-researched [13]. In robotic audition GMM classifiers are used successfully [34]. These might perform well for vehicle classification as well. In comparison to robot audition were, for example [87] uses 14 microphones, vehicle detection used only a few. Additionally more sophisticated pipelines are used, combining classification of speech and localisation of the source. An example is shown in 2.10. The algorithm in [75] first classifies a frame to contain vehicle sounds or not, with a neural network. Subsequently the location of these sounds is estimated by the localisation part of the algorithm. No information between the algorithms is shared, other than the raw signal. The classification is run on every frame, while the localisation is only run on positive classified frames.

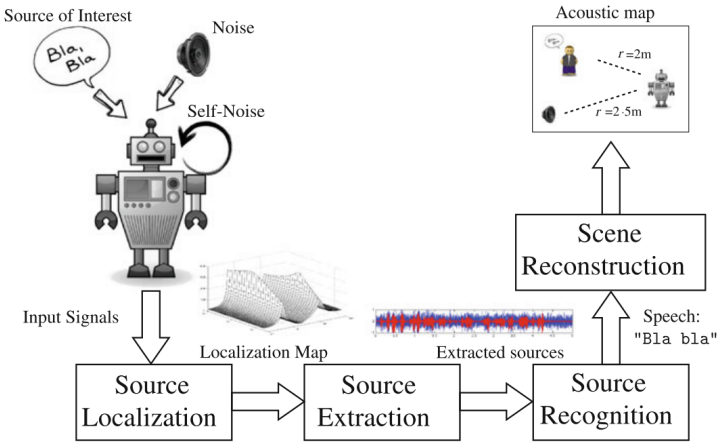


Figure 2.10: Flow diagram of a classical robot audition pipeline. First array processing techniques are used to localise and extract the sounds produced by a single source. This extracted source fragment is then fed into the recognition part of the system. Source: [88].

2.3.4. Experiment summary

Not many acoustic perception systems for the classification of vehicles exist. Five systems were examined, namely [75], [37], [85], [60] and [72]. Two of the five methods were created in the previous millennium and

are probably overtaken by more recent methods. The aims of the systems differed slightly. In one case [75] binary classification was used to determine if one or more vehicles were present. Three system [37], [60], [85] determined the vehicle type. The type classes here were broad, for example buses, motorcycles, sedans and vans. In general vehicle sound perception focusses mainly on regular cars, because they are the most prevalent.

In summary many different kinds of features are used for the classification of vehicles, namely: frequency bins, regular and PCA-decomposed ([72], [75]), Mel frequency cepstral coeficients (MFCC) [60], [75], Harmonic line [60], Schur coeficients [60] and plain LPC features [37]. Both of [60], [72] use a SVM classifier. The systems in [37], [75] use a neural network and [85] uses PCA in combination with a distance metric for classification.

2.4. My contribution

To make acoustic detection a valid alternative to vision-based vehicle detection system, its feasibility must be clearly proven. This thesis makes a first step by evaluating the feasibility of cars driving in a straight line on road with near constant speed. A key aspect of the acoustic vehicle detection challenge is the effect of movement of the listener vehicle on the acoustic detection of other road vehicles. Performances are compared between driving and idle scenarios and the effects of the driving scenario are further investigated. The goal is to reach an intuition on the effects of driving and how to deal with them.

In the process commonly used datasets, features and classifiers are evaluated on their suitability to this domain. Two existing datasets, Google's AudioSet and RoadCube are tried and compared. Furthermore the following feature types are investigated: energy, energy entropy, zero crossing rate, diverse spectral statistics, Chroma features, harmonic line, LPC features, and MFCC (more details in section 3.2). Many come from different fields and in this thesis their suitability for vehicle detection is evaluated. In the process of collecting and implementing feature extraction methods, a new python library for acoustic feature collection was created. It contains methods from LibRosa, pyAudioAnalysis, Scikit-talkbox and a few own implementations. The repository can be found on `https://github.com/pvanlaar/py_sound_feature`.

Additionally four classifier types were used, namely: SVM, GMM classifier, MLP and Random Forest. These classifiers are used in vehicle recognition (SVM, MLP) or robot audition GMM. Additionally the random forest classifier was added due to its feature selection and lack of overfitting characteristics.

For the evaluation of the existing datasets another dataset was captured, called DriveSound. First of all a new realistic dataset, , was created for the evaluation and comparison of the existing datasets. To the knowledge of the author no similar dataset, which is captured from a moving listener vehicle, exists yet. It aims to simulate the real world as closely as possible. Therefore it is captured without any preprocessing or other alteration of the data. DriveSound contains eight scenarios, each on a different location with maximum speeds of 30, 50 or 70 km/h. In three of those scenarios the listener vehicle is driving, while in the other cases it is parked adjacent to the road. For this reason there are samples which are easier to classify while others are very hard classify. Hard samples occur for examples when the listener vehicle is waiting for traffic light on a busy intersection and it is surrounded by both stationary and moving vehicles. In the dataset many different vehicle classes were discerned, which can be grouped at own insight for classification. The data is annotated to get all the moments a vehicle is at the closest point to the listener vehicle. Camera and 3D lidar scans are included, thus a re-annotation can be done when in another way when needed. Finally the dataset is captured by multiple microphones thus it is suited for pipelines which include a localisation step. More details about this dataset and the collection process can be found in de datasets section 4.3.

3

Method

The aim of this research is to evaluate different configurations of a machine learning pipeline, which are able to distinguish between sound fragments containing vehicles and those without. This chapter explains the methods and algorithms used for the classification pipeline and the evaluation of its performance. It starts with section 3.1 where the classification process is explained. Subsequently the acoustic feature types are listed in section 3.2, followed by the used classifier algorithms in section 3.3. Finally section 3.4 elaborates the methods to evaluate the performance of the pipeline.

3.1. Classification pipeline

The classification pipeline consists of three elements during operation, namely the microphone, feature extractor and the classifier. A schematic of this pipeline can be seen in figure 3.1. The microphone captures the sound waves (more detailed information in section 2.2.1). The signal is then per sound fragment, a small part of the signal of typically one to a few seconds, fed into the feature extractor. The feature extractor will quantify one or more properties of the fragment in a feature vector. This abstracted and compressed form is finally used by the classifier to discern between different classes. Classes are groups of sounds with a certain commonality, for example they all contain cars. A model, trained on a dataset, contained in the classifier is then used to categorise the feature vector belong to a certain known class. The output of the classifier and thus the pipeline is a class label prediction for the captured sound at that moment.

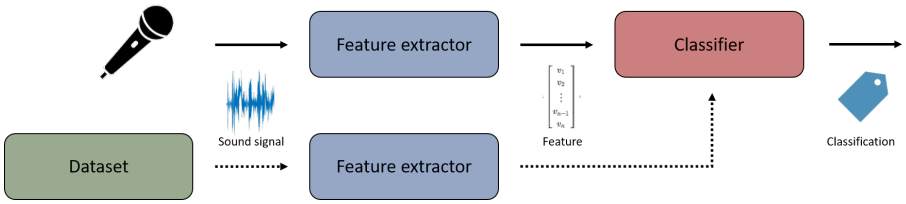


Figure 3.1: An schematic overview of the classification pipeline. On the left side the sound is captured by a microphone, which translates it to an one dimensional signal (with sample rate f_s). The feature extractor extracts certain properties of the signal. These properties are subsequently used by the classifier to classify the sound as for example a vehicle. The bottom half of the pipeline is used during the training phase. The samples from the dataset are used to train the classifier, which happens once. (The microphone is taken from [89])

Before the classification pipeline can function, the classifier must be trained on a dataset. The dataset contains independent sound samples of which all have a class label. The amount of classes and the sample sizes of those classes are important parameters of the classification problem. The samples of the dataset go into the same feature extractor as used when the pipeline is used for the prediction of new samples. In this stage the parameters of the classifier model are tuned to the data. Commonalities from the dataset, in feature representation, are captured in this model.

3.1.1. Feature extraction

Features are an abstraction of data, which aim to capture certain properties. For a feature to be useful, it must contain enough information and the information must be usable for classification. This means that the samples of different classes must differ in the feature space, otherwise the classes cannot be distinguished from each other by a classifier. Many different feature types exist and multiple types can used alongside each other. Each element of the feature vector is considered independent [90]. Features are a more compact form than an audio signal, which means that the amount of dimensions of the feature signal is lower. A higher dimensional problem is more complex, but also increases the solution space for the classifier. It increases the possibility that the best model exists in this space, but also increases the amount of required training samples. There is thus a trade-off between user more and higher dimensional features and limiting the amount of required training samples. The invariance of a feature type may also be a major factor for its suitability to the problem. Invariance is a measure of robustness and makes certain types more suitable for certain classification problem. For example when an acoustic feature is disrupted by wind noise, it probably will not perform well outdoors.

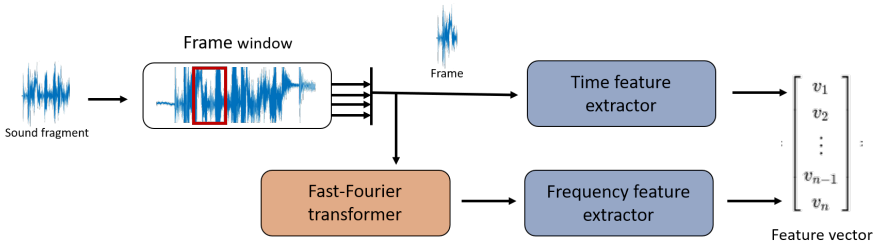


Figure 3.2: The pipeline to extract a feature out of a audio fragment. The fragment is first divided into frames with a size of 2048 elements and a overlap of 0.5. For each frame a feature vector is extracted. This can be done in the time and frequency domain. The feature vector can contain multiple types of features. For the frequency domain features, the signal a histogram with the frequency content is taken using the Fast-Fourier Transform (FFT). The fragment feature vector is obtained by averaging over each frame

An audio fragment is represented by a feature vector with length F, which is thus the dimensionality of the feature space. The feature extraction happens for each audio frame however, which is a small piece of the fragment. In this thesis each fragment is divided into frames of 2048 elements with an overlap ratio of 0.5, regardless of the fragments length. Different methods for the frame feature extraction exist, which can work in the time and frequency domain. For the frequency domain features the histogram is taken with the frequency content using the FFT. The fragment feature vector is created by averaging the frame features. Features from fragments with multiple lengths can be combined in this way. Depending on the feature type, the feature extraction for a frame is done in the time and the frequency domains. When the acoustic The acoustic feature types used in this thesis are discussed in section 3.2. After the extraction of all features the whole dataset, the features are normalised. First of all each sample is mean normalised to zero to remove the DC bias. Secondly the features are scaled in a way that each feature element has unit variance.

3.1.2. Classifiers

A classifier is a parametrized model used to predict the class of new, unseen samples. During the training phase the parameters of the model are optimized for the prediction performance on the training set. The amount of parameters of a classifier is an important property due to overfitting. When this effect occurs the model is only performing well on the training data, but does not generalize to new samples (example in figure 3.3). Generally the simple models are easier to fit and compute, but span only a limited part of the possible models or hypothesis. So there is a trade-off between computability and performance [90]. Another key property is the training, which is the time to fit all the model to the training samples. In this thesis the training time is not a key metric and the reported results should only be used for comparison of the algorithms. The classifier types used in this thesis are discussed in section 3.3. Each type has parameters, which are set during the training phase, and meta-parameters, which are chosen beforehand. Examples of the meta-parameters of the model are the model complexity or the outlier cost in the learning algorithm. Each type of classifier has very different meta-parameters.

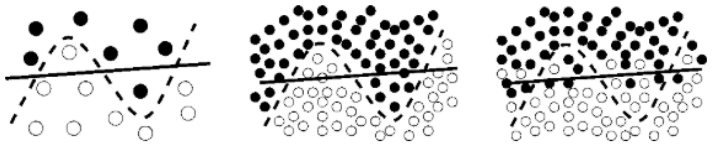


Figure 3.3: Schematic display of overfitting. On the left the training set is shown, which is a subset of the population. For the classification of the whole population one of those lines is correct, where the dotted line is more complex, but has a smaller training error. The middle picture shows the case where the dotted line is correct. In that case the straight line would underfit. In the right picture the straight line is correct and the dotted line proved too dependent on the training data. This is called overfitting. Source: [91].

3.2. Acoustic feature types

Different acoustic feature types are used to extract the feature from an audio frame. Many of them were used in earlier experiments (see section 2.3.4). Some of them extract from the time signal, while others are extracted in the frequency domain. An overview of the feature types used, along with the domain they are operating in, is given in table 3.1. This section will list the different types per the domain they are extracted in, starting with the time domain. The frequency domain part is preceded by the explanation about the transformation to the frequency domain. Many feature extraction functions are based on implementations from the libraries Librosa [92]. and pyAudioAnalysis [93]. When this is the case the library will be named in brackets after the feature name. In the remaining paragraphs the different feature types are discussed. Commonly used elements are the frame's time signal s, it's frequency bins S and the frame size K.

Time	Energy
	Energy entropy
	Zero crossing rate (pyAudioAnalysis)
	LPC features (Scikits: Talkbox)
Frequency	Frequency bin features (Librosa)
	Spectral centroid (Librosa)
	Spectral spread (Librosa)
	Spectral entropy
	MFCC (pyAudioAnalysis)
	Harmonic line
	Chroma features (Librosa)

Table 3.1: Overview of feature types, separated according to the domain the are extracted from.

3.2.1. Time domain features

Energy: Energy is a intensity measure in the time domain. The amount of energy in a frame weakly correlates to the perceived loudness of the sound [94]. Loudness also takes the receiver's sensitivity into account. The human ear is for example more sensitive to certain frequencies. The energy is calculated by taking the element-wise square of the time frame and normalising for the frame size, i.e.

$$\sum_i \frac{s_i^2}{K}. \tag{3.1}$$

Energy entropy: Entropy in the signal analysis sense is measure the likeliness to white noise as opposite to signal patterns. It originated from thermodynamics where it also is a measure of disorganisation. White noise is regarded as a system in equilibrium because of the constant output. This noise type therefore has the highest entropy, while other (periodic) parts of the signal have a lower entropy [95]. The calculation is done by first getting the power signal s^2. Afterwards the histogram is taken to get the probability density function

estimate, with a bin amount of 256. The entropy is then calculated by using

$$-\sum_k h(s_k^2)\log_2 h(s_k^2), \tag{3.2}$$

where $h(s^2)$ is the histogram of the squared time signal, which is normalised so its sum equals 1.

Zero crossing rate: The zero crossing rate measures the amount of sign changes between measuring points in a frame. It is normalised by the length of the frame, thus it is a value between zero and one.

LPC features: LPC parameters are used for speech recognition and to compress signals [96]. They are calculated by using an innovation filter, which decomposes a signal in LPC and white noise [60]. The amount of coefficients is equal to the order of the filter.

3.2.2. Frequency domain features

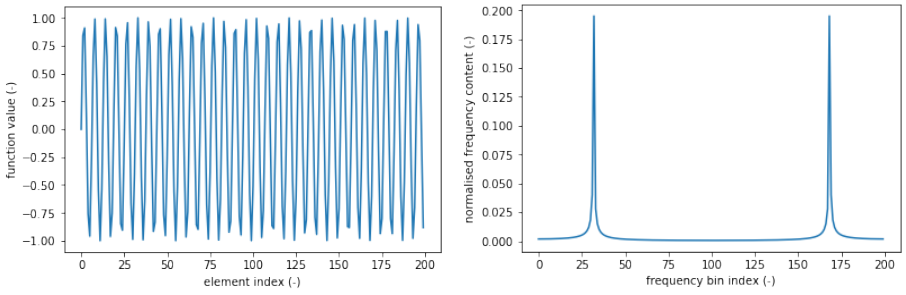


Figure 3.4: Example sine signal and its frequency histogram.
left: sine signal, **right:** magnitude of frequencies calculated by FFT with a frame size of 200. The first peak is the sine frequency, but the second peak is an alias (duplicate). Only frequencies until the nyquist frequency can be measured.

An audio signal can be converted to the frequency domain by the FFT. It models the audio signal as a sum of unit sine functions with a certain frequency. The transform results in a histogram of the frequencies in the signal (example of signal and histogram can be seen in figure 3.4). Only frequencies up to half the sampling frequency (Nyquist frequency) can be detected. The 2048 element frames used in this thesis will result thus in a 1024 histogram which ranges from 0 Hz to the Nyquist frequency. With a sampling rate of 48 kHz, as found in many of the used datasets, the resolution for each bin is 23.4 Hz. Increasing the frame size, and thus increasing the time duration of a frame, will thus result in a higher frequency resolution and a lower time resolution. If the sample rate is increased while the frame size is kept equal, the frequency resolution will decrease while the time resolution will increase [71]]. Before a frame is converted to the frequencies it is first multiplied by a windowing function, in this case the Hamming window. This type of windowing function is generally used [72], [75]. Windowing reduces the leakage of the transform, which occurs if the base function does not end or begins its period at the start of the frame.

Frequency bin features: Frequency bin feature simply uses the frequency histogram created by the Fourier transform. Due to the histogram size, 1024 in this case, only frequencies below a threshold of $3000\,Hz$ are kept. Road vehicles operate below this threshold (see section 2.2.2). The Doppler effect and diffraction (section 2.2.1) can shift frequencies, so this can effect frequency vector features or other spectral based features.

Spectral centroid: The spectral centroid is the mean frequency based on the Fourier histogram. It can be calculated by multiplying the centre frequencies f_k with the power in the corresponding bins S_k, i.e.

$$\frac{\sum_k f_k S_k}{\sum_k S_k}. \tag{3.3}$$

Spectral spread: Spectral spread is like the standard deviation of the frequency histogram. It is defined as the difference between a bin and the centroid weighted by the power amount in the bin, which results in

$$\sqrt[p]{\left(\frac{\sum_k S_k(f_k - Centroid)^p}{\sum_k S_k}\right)} \tag{3.4}$$

with $p = 2$ is the order.

Spectral entropy: Spectral entropy is similar to the earlier mentioned energy entropy, but now it is calculated based on the frequency content. Replacing the energy histogram $h(s^2)$ from equation 3.2 with the frequency variant yields

$$-\sum_k S_k^2 \log_2 S_k^2. \tag{3.5}$$

MFCC:

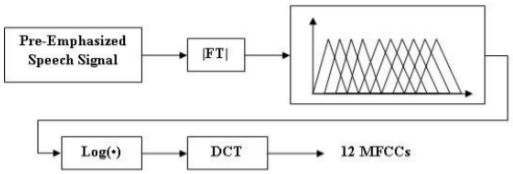


Figure 3.5: Pipeline showing the extraction of MFCC features. Source: [97]

MFCC are features developed for the purpose of speech recognition. They can be extracted from the frequency histogram of a frame. First Mel filters are used to convert the frequencies to the non linear Mel scale which is inspired by the human ear [96]. Subsequently the logarithm of the mel scale is taken and the discrete cosine transform is used to convert the Mel scale to the cepstral coefficients, i.e.

$$MFCC = dct(log10(dot(X, melBank))). \tag{3.6}$$

A more detailed explanation of the calculation procedure can be found in [96]. The MFCC extraction pipeline can be seen in 3.5.

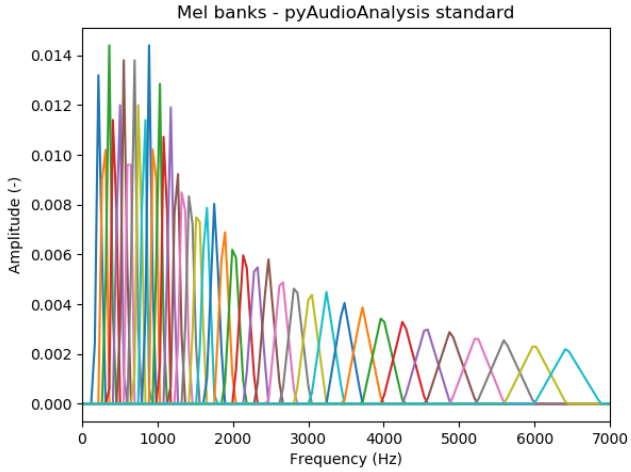


Figure 3.6: The mel-frequency bins of the MFCC filter bank. They are used to convert the frequencies of the signal to the mel-spectrum. This is done by taking a histogram using these unequal bins.

Harmonic line: Harmonic line features target the frequencies emitted by a vehicle's engine. Each engine has a fundamental frequency associated with the cycle rate, which manifests itself acoustically in the form of harmonic peaks. The harmonic peak frequencies f_p relate to the fundamental frequency f_e with

$$f_e = k f_p \tag{3.7}$$

, where k is an integer. Harmonic line Association is used to find the frequency. First the dominant peak of the frequency histogram is found. Because it is unknown which k corresponds to the dominant peak, multiple guesses are tried (with k varying between 2 and 11). For each of those guesses the f_p is calculated and then line is constructed for $k \in [1, 11]$. The line with the maximum total spectral energy found in the points is chosen. Only the first two points are saved as features.

Chroma features:
Chroma features aim to capture the musical tones found in the frame. A histogram is made on the tones without taking octaves into account. Thus the same note from different octaves are counted the same. For the calculation the implementation of librosa is used.

3.3. Classifier algorithms

In this thesis four types of classifiers are used: SVMs, GMM classifiers, MLPs and Random forest classifiers. The first three types are already being used for acoustic classification and the Random Forest is used for as a baseline because it requires little tuning to give good results. Additionally the state of this kind of classifiers is understandable by humans. All of the implementations come from the Scikit-learn library (version 19.1) [98]. Implementation details were taken from the online documentation.

3.3.1. SVM

SVMs are suited for two-class problems. They aim to estimate the boundary in the feature space, called a hyperplane, separating both classes best [66]. The feature space often spans multiple dimensions, thus the hyperplane must be multi-dimensional as well. The hyperplane is defined by the support vectors, which are samples lying close to this boundary. The original method creates a straight hyperplane (as seen in figure 3.7). This is a efficient method, but if the data is not separable by a straight plane curved hyperplanes have to be used. SVMs require no prior information about the problem other than training data [66].

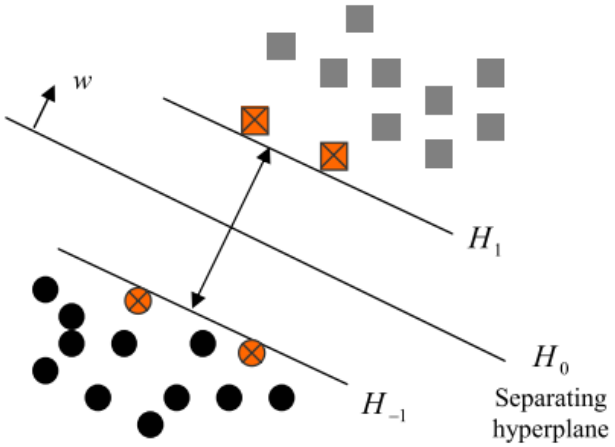


Figure 3.7: Example of a straight hyperplane created by a linear SVM. The circles and squares belong to positive and negative class respectively. The four shapes made orange are support vectors, which define the boundary hyperplanes H_{-1} *and* H_{+1}. The separating hyperplane H_0 is in middle the boundary hyperplanes. The space between the boundaries and thus the classes is the margin, indicated by the double sided arrow. Source: [66]

One of the key strengths of the original SVM is the simplicity of its decision function

$$f_{SVM-linear}[\boldsymbol{x}] = \boldsymbol{w} \cdot \boldsymbol{x} + b, \tag{3.8}$$

which contains only a single dot product. $\boldsymbol{w}$ is a vector of weights and b is a scalar bias Both are obtained during the training phase. $\boldsymbol{x}$ is the feature vector, which is to be classified. A label y will be given to the feature according to the sign of the decision function, i.e. $y = sgn\left(f_{SVM}[\boldsymbol{x}]\right)$, therefore a SVM is suitable for binary problems.

For some problems straight hyperplanes are not able to separate the classes adequately. In that case curved hyperplanes have to be used instead. To support these, the linear decision function 3.8) has to be modified by using the kernel trick. This means that the features are mapped to a higher dimensional space than the feature space where the classes are linearly separable by using a function $\phi(\boldsymbol{x})$. The decision function then becomes

$$f_{SVM-kernel}[\boldsymbol{x}] = \boldsymbol{w} \cdot \phi(\boldsymbol{x}) + b. \tag{3.9}$$

Instead of directly defining the mapping $\phi(\boldsymbol{x})$, the kernel trick defines the quadratic form $K(\boldsymbol{x_i}, \boldsymbol{x_j}) = \phi(\boldsymbol{x_i})\phi(\boldsymbol{x_j})$. Commonly used kernels include the polynomial kernel ($K_{poly}[\boldsymbol{x_i}, \boldsymbol{x_j}] = \gamma(\boldsymbol{x_i} \cdot \boldsymbol{x_j} + r)^d$) with degree d, positive multiplier γ and bias coefficient r and the RBF kernel ($K_{RBF}(\boldsymbol{x_i}, \boldsymbol{x_j}) = \exp^{-\gamma|x_i - x_j|^2}$) kernels [66], [91]. If the kernel is chosen as the dot product ($K_{linear}[\boldsymbol{x_i}, \boldsymbol{x_j}] = \boldsymbol{x_i} \cdot \boldsymbol{x_j}$), the linear classifier case is obtained.

During the training phase the optimal hyperplane is estimated. Optimal in this case case means that the distance between both classes, called the margin (see figure 3.7), is maximised while the amount of training errors is minimised. Increasing the margin will improve the generalisation characteristics of the classifier. [66]. Training errors are incorrect classifications made on the samples of the training set. A training set containing outliers, which is often the case in real world data, can be difficult to separate with an absolute margin as the above definition does. This means that estimating a hyperplane separating all samples correctly is not possible or only with a very complex model, which does not generalise well. Instead of an absolute margin a soft margin could be used, which discounts outliers. This means in mathematical terms that samples close to or on the wrong side of the boundary can be discounted by a ξ_i. The size of the discount ξ_i can vary for each sample individually. If the discounts can be chosen freely this would lead to trivial solutions, thus they must be penalized during training. The training has now become a optimization problem where the margin has to be maximized and the sum of the discounts has to be minimised. If positive labels are defined as $y \geq 1$ and negative ones as $y \leq -1$, the margin size is defined as $\frac{2}{||\boldsymbol{w}||}$. This means that to maximise the margin the length of the weight vector $||\boldsymbol{w}||$ must be minimized. Combining both objectives in a single problem yield the following minimisation problem

$$\min\left(\frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C\sum_{i=1}^{N}\xi_i\right) \tag{3.10}$$

subject to the constraints of the training samples

$$y_i(\boldsymbol{w}^T\phi(\boldsymbol{x}_i) + b) \geq 1 - \xi_i. \tag{3.11}$$

Here C is the regularization parameter or training parameter, which penalises the sum of sample discounts. A high C will permit fewer training errors thus the model will fit the training set better, but at the same time it may reduce the classifiers' generalisation ability. If $C = \infty$, the SVM has a hard margin again [66]. Using Lagrangian multipliers λ_i to include the constraints (equation 3.11) in the minimisation problem (equation 3.10) the problem can be written as a single equation. In this thesis a deconvolution method [99] is used to solve this problem, provided in the sklearn [98] SVC class, which is rooted in the liblinear [100] and libsvm [101] libraries for the linear and kernel SVMs respectively. When N is the sample amount in the training class and M is the amount of support vectors, the used algorithm has a computational complexity of $\mathcal{O}(N * M)$ or $\mathcal{O}(M^3)$, whichever is larger (which depends on C) [102], [103]. The computation of kernel functions accounts in practice often for more than half the total computing time [103].

It has been shown that the optimal solution is obtained when $\lambda_i \neq 0$ for all samples on the boundary or on the wrong side [91]. These are called the support vectors (the orange features in 3.7). Using the solution the decision function can be rewritten as only dependent on the support vectors, i.e

$$f_{SVM}[x] = \sum_{j=1}^{M}(\lambda_j y_j K[\boldsymbol{x_j}, \boldsymbol{x}]) + b. \tag{3.12}$$

The weights $\boldsymbol{w}$ are substituted by a summation of the multiplication of the Lagrange multiplier λ_i, label y_i and kernel $K[\boldsymbol{x_j}, \boldsymbol{x}]$ for each support vector (M is the amount of support vectors). This makes the classification only dependent on the support vectors, instead of all training samples. Retraining the SVM with only the support vectors using the regular method yields only approximate results. It is possible to get an exact result with incremental SVM training methods as is done in for example [104].

SVMs can also be extended for a multi class system. A possible option is to use the one-versus-all configuration. For each class a classifier is trained, where the negative class contains all other classes. All classifiers score the sample using the decision function (equation 3.12) or calculating the probability. The probability can be calculated by for example fitting a sigmoid function [105]. The highest positive score wins, which means that it is most belonging to that class. Other options include one-versus-one configurations or multi class extensions of the SVM method itself. These and other methods can be found in [106].

3.3.2. GMM classifier

A GMM classifier uses probability density functions to estimate the probability that a feature belongs to a certain class. Each class has a separate probability density function. During classification the class with the highest posterior probability wins and the corresponding label y is selected as classification. The probability density function is formed by a linear combination of Gaussian distributions. The complexity of the model can be adjusted by setting the amount of Gaussians to be used for each class. Including more Gaussians means that more parameters have to be estimated during the training phase, which requires more training data. Knowledge about the relative occurrence of a certain class can be incorporated into the classifier, by setting the prior probability $p(y_{class})$ for each class.

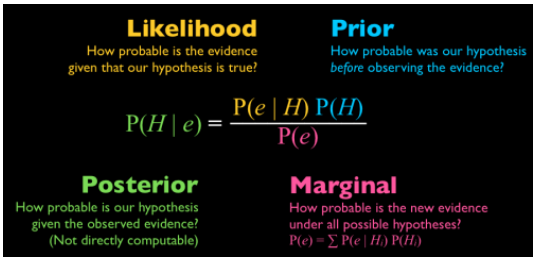


Figure 3.8: Explanation of Bayes rule and its element. Source: [107]

The classification is based on the posterior probability, where the class with highest score wins, i.e. $y = \arg\max_{class} p(y_{class}|\boldsymbol{x})$. These posterior probabilities are calculated using Bayes' rule which, formulated in terms of features $\boldsymbol{x}$ and labels y is

$$p(y_{class}|\boldsymbol{x}) = \frac{p(y_{class})p(\boldsymbol{x}|y_{class})}{\sum_{i=0}^{classes} p(y_i)p(\boldsymbol{x}|y_i)}. \tag{3.13}$$

An explanation of Bayes rule can be found in figure 3.8. The prior probability (before the classification) $p(y)$ can be set beforehand or can be derived from the training set, thus utilising prior knowledge. If it is so that each class occurs equally often, the decision function (equation 3.13) can be simplified by selecting the class with the maximum likelihood $p(\boldsymbol{x}|y_{class})$, instead of calculating the posterior probability $p(y_class|\boldsymbol{x})$.

The likelihoods $p(\boldsymbol{x}|y_{class})$ are estimated by a model consisting of a linear combination of Gaussian distributions. A Gaussian distribution

$$\mathcal{N}(\boldsymbol{x};\boldsymbol{\mu},\Sigma) = \frac{1}{(2\pi)^{\frac{||x||}{2}}\sqrt{||\Sigma||}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right) \tag{3.14}$$

is dependent on the mean $\boldsymbol{\mu}$ and covariance matrix Σ. The mean determines the centre of the Gaussian and the covariance matrix determines its shape. For a feature space with D dimensions, μ consists of D parameters, Σ of maximal $(D*D-D)/2+D$ and the mixing weights α_i give $D-1$ (because $\sum_{i=0} \alpha_i = 1$). When G Gaussian distributions are combined the likelihood $p(\boldsymbol{x}|y_{class})$ model becomes

$$p(\boldsymbol{x}_i|y_{class}) = \sum_{i=0}^{G} \alpha_i \mathcal{N}_i(\boldsymbol{x};\boldsymbol{\mu}_i,\Sigma_i) \tag{3.15}$$

with $G(\frac{1}{2}D*D+\frac{3}{2}D+1)-1$ parameters It is possible to reduce the amount of parameters in the covariance matrix, but this will limit the amount of possible shapes of the Gaussians. Possible reductions include making the shapes of all Gaussians similar and using a spherical shape, i.e. $\Sigma_{sphere} = I$.

The goal during training is to find the combination of parameters Θ, which maximises

$$p(\mathcal{X}|\boldsymbol{\Theta}) = \prod_{i=0} p(\boldsymbol{x}_i|\boldsymbol{\Theta}). \tag{3.16}$$

for the features set of a class $\mathcal{X}$. One of the possibilities for obtaining the parameters of the mixtures is the Expectation-Maximisation (EM) algorithm [66], which is an efficient way to solve the problem [36]. If enough Gaussians components are used in the mixture model, the density tends to converge to the true density [36]. Generally speaking a GMM classifier requires more samples than a SVM [36].

Instead of a single feature, a group of features can also be evaluated simultaneously by using a similar formulation to the training phase (equation 3.16) and multiplying feature 1 to V. i.e.

$$p(y_{class}|\boldsymbol{x}_0..\boldsymbol{x}_V) = \prod_{i=0}^{F}(p(y_{class}|\boldsymbol{x}_i)). \tag{3.17}$$

This means that it can also work on feature vectors of multiple frames simultaneously. Averaging is then not needed any more.

3.3.3. MLP

MLPs are, as the name implies, networks of linked neurons. The network is forward fully connected, which means that each neuron is connected to all neurons from the previous layer. A neuron can be seen as a function where inputs are weighted. If the outcome of the function exceeds a threshold, the neuron will activate and produce a signal. An example of a neural network structure can be seen in figure 3.9. The first layer is the input layer where the feature vector elements are put in. The final layer of the network is the output layer, which contains for each class the scores that a feature belongs it. In between these layers are hidden layers, which connect the neurons between the input and output layer. A key characteristic is the presence of hidden layers. In this thesis the implementation from scikit-learn [98] is used, which is CPU based.

The amount and size of the hidden layers are a parameter of the classifier. More neurons in a network requires a complexer model and more parameters to be trained. The type activation function of the neurons can be varied. Possible function are: identity $f(x) = x$, logistic $f(x) = \frac{1}{1+\exp(-x)}$, tanh $f(x) = tanh(x)$, rectified linear unit function $f(x) = max(0, x)$. Different solvers for the weight optimisation are used, namely adam [108] and Limited memory - Broyden-Fletcher-Goldfarb-Shanno (L-BFGS). According to the scikit-learn documentation adam performs better on large datasets, while L-BFGS performs better on smaller datasets. The stochastic gradient descent is not used, because adam is an improved version of it. For the adam solver the initial learning rate can be set, which controls the step size of updating the weights. The (L2) regularization parameter α is used to penalize large weights and combat overfitting. A lower value will result in a less complex decision boundary and vice versa.

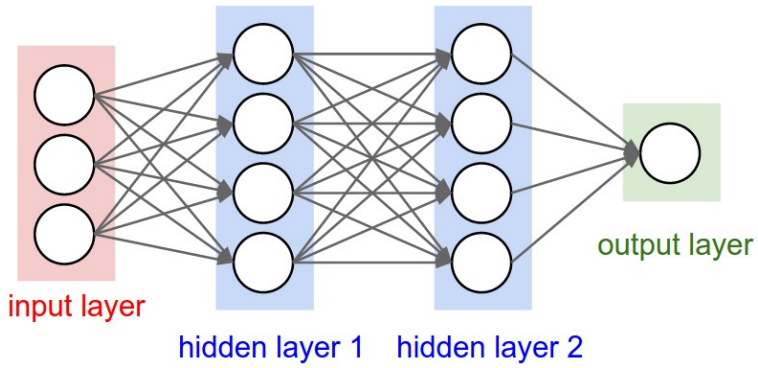


Figure 3.9: Example of a forward neural network. It contains two hidden layers and one output neuron. Source: [109]

3.3.4. Random forest classifier

A random forest classifier consists of an amount of decision trees, which work together to classify new samples. Adding more trees will improve the results, while the overfitting effect is limited due to the averaging over all of the trees [110]. A tree starts at the root and branches out to leaves. At each split a the samples are split into two ways.s

Each of the trees will estimate the probability that a sample belongs to a certain class. The probabilities are averaged over all of the trees and sample is given the label of the class with the highest probability. The used scikit-learn [98] implementation diverges here from the original method in [110], where each tree votes for a class and the class with the most votes wins.

During the training phase each tree is created on separate randomly selected subset of the training data. The training of the individual decision trees can be found in [111]. These subsets are sampled with replacement thus samples can be used for the creation of multiple tress. Additionally only a random selection of the feature

elements is used of the subset. This random feature selection is different for each tree. The tree will be the optimal classifier for that subset of features, not the whole training set. Therefore each tree is a randomization of the optimal tree. Due to this randomness, adding more trees will increase the bias, but reduces variance. Generally the variance decrease compensates for the bias increase, thus it is advantageous to add more trees to the forest [110].

Because each tree is trained on a subset of the training set, the remaining part can be used to evaluate the tree classifier. The forest classifier can then be evaluated by evaluating each sample only on those tree that weren't trained on it. The importance of each feature element can be estimated by randomizing the values of one element and then repeating the forest evaluation. The metric for importance is the gini importance, which is the percentage increase in misclassification rate between the general classification and the randomized set. Afterwards this value is peak normalised.

3.4. Performance evaluation

Different methods for feature extraction and classification can be used in the classification pipeline. Furthermore for the training on the classifier, different datasets can be used. For the performance evaluation these pipeline configurations need to be compared in a structured way. All of the configurations are evaluated by measuring the classifiers ability to classify unseen samples. This performance is captured by a performance metric, which are explained in section 3.4.1. The structure of these evaluation experiments is elaborated in sections 3.4.2 3.4.3 explaining the feature selection and the classifier optimisation respectively.

3.4.1. Performance metrics

For the evaluation of a machine learning algorithm, the choice for a performance metric is important [112]. A performance metric quantifies the correctness of the classification by comparing the classifications with the ground truth. For a binary classification problem, with a positive and negative class to choose from, there are four possible outcomes at each sample. It can be classified as positive or negative, and this classification can be correct or incorrect. All these possible outcomes are summarised in the confusion matrix, shown in table 3.2. The true positives are in the top left cell and the false positives or type 1 errors in the top right corner. On the bottom left the false negatives are found or type II errors and finally in the bottom right corner are the true negatives.

		Classified	
		Positives (CP)	Negatives (CN)
Actual	Positives (AP)	True Positives (TP)	False Negatives (FN)
	Negatives (AN)	False Positives (FP)	True Negatives (TN)

Table 3.2: Confusion matrix of a binary classification problem. It contains all possible outcomes of the classification of a sample.

The well interpretable metric is accuracy. It is defined as the amount of correct classifications divided by the total amount of samples. In terms of the confusion matrix it can be written as

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}. \tag{3.18}$$

A disadvantage of this metric is that if there are significantly more samples from one of the classes it can be unreliable. When there are for example more positive than negative samples, a classifier which classifies everything as positive will score well on accuracy. The amount of false negatives and true negatives will be zero ($FN = 0,\ TN = 0$), while the true positives and false positives become the actual positives and negatives respectively ($TP = AP,\ FP = AN$). Equation 3.18 will then become

$$accuracy = \frac{AP}{AP + AN}. \tag{3.19}$$

It can be seen that for large fraction of positive samples ($AP > AN$), the accuracy of this particular classifier is high, while its predictive ability remains limited. This effect is seen for both positive and negative imbalances.

Two other metrics, precision and recall are invariant to the unbalanced classes when used in combination [113]. Precision is the fraction of correct positive classifications to the total number of positive classifications.

It is useful when the positive class is more important than the negative class. The metric can be calculated by

$$precision = \frac{TP}{TP + FP}. \tag{3.20}$$

Recall or sometimes called sensitivity is the True Positive Rate (TPR), which is the correct positive classifications divided by all the actual positive samples. It is defined as

$$TPR = recall = \frac{TP}{TP + FN}. \tag{3.21}$$

Precision and recall can be combined in a f-score. This is a category of metrics, which define a weighted combination of both. One of those, the f1-score puts equal weight on precision as on recall. It is the harmonic mean of both metrics, which can be written as

$$f1\text{-}score = 2 \cdot \frac{precision \cdot recall}{precision + recall}. \tag{3.22}$$

The f-score changes depending on the choice for the positive class [113].

If the negative class is important to consider, the specificity or True Negative Rate (TNR) can be used as performance metric. It is similar to the recall, but for the negative class and is defined as

$$TNR = \frac{TN}{TN + FP}. \tag{3.23}$$

The False Negative Rate (FPR) is the opposite of this metric, with a relation $x - 1$. It can thus be written as

$$FPR = \frac{FP}{TN + FP}. \tag{3.24}$$

Another metric is a Receiver Operating Characteristic curve (ROC curve), which combines the TPR and FPR metrics. It is a plot showing the relation between the two at different thresholds of the decision function. This means that at first everything is classified negative, thus both the TPR and FPR are zero, because there are no postive classifications. Subsequently the threshold is increased gradually, until eventually everything is classified positive. A higher threshold often mean that a larger portion of the actual positives are classified (higher TPR), but also that more negatives are classified as positive (higher FPR). An ideal classifier will go through the top left corner, where the TPR is 1 and FPR is zero. The area under this curve could also be used as an performance metric, if the metric needs to be a single number. In the ROC curve a reference line is shown for comparison, which shows a classifier guessing randomly. For this reference the assumption is made that there are an infinite amount of samples, which results in the straight line. The line corresponds with an Area Under Curve (AUC) of 0.5. An example ROC curve can be ssen in figure 3.10.

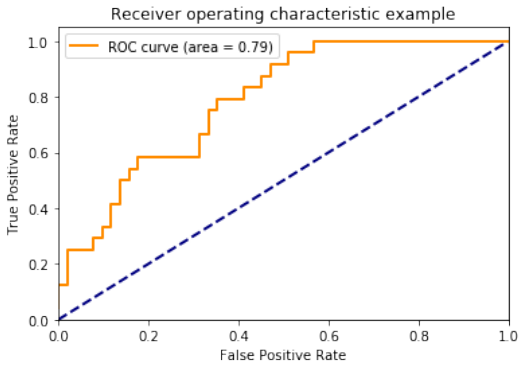


Figure 3.10: An example ROC curve. At different values of the decision function threshold the true and false positive rates are determined. An ideal classifier goes through the top left corner. The reference line represents a random guessing classifier. A compressed metric based on this curve is the AUC

Unlike the previous scores the Matthews correlation coefficient (MCC) is a balanced measure, because every part of the confusion matrix is used. Therefore it is well suited for datasets with unbalanced classes [114]. It is defined as

$$MCC = \frac{TP * TN - FP * FN}{((TP + FP) * (FN + TN) * (FP + TN) * (TP + FN))^{1/2}} \tag{3.25}$$

. The score ranges from -1 to 1. Scoring a 1 means that the classifier perfectly predicted the class and -1 means that everything is predicted wrongly. In the latter case reversing the prediction yields a perfect classifier again. A score of zero means that the predictions are done randomly. MCC is thus a measure for the predictability, regardless of class size [114].

3.4.2. Random forest feature selection

Selecting the best features for the machine learning problem is an important step. Minimizing the amount of features generally reduces the amount of parameters of classifiers which need to be trained. The random forest classifier already has a metric to evaluate the importance of each feature element (see 3.4.2). This property can be used to evaluate the features and select the best ones [115]. An added benefit is that the random forest classifier is relatively quick to train and doesn't overfit.

The same feature space is used for experiments on all datasets, but the feature importances are determined separately for each of the datasets. For each dataset first a classifier optimization, between random forest configurations, is performed. These random forest configurations differ in the amount of tree the classifiers contain. The detailed optimization procedure can be found in the next section (3.4.3). The best performing random forest classifier is used to measure the importances of the feature elements. These importances are peak normalised to the highest importance on a dataset.

After the normalised importances are estimated for each dataset, they are used to select a common feature space. The selection is made by setting a threshold on the importances of each dataset. This threshold is the same for each dataset and is chosen by hand. Many of the used features in this thesis have multiple elements. If one of these elements is selected all of the elements are included in the selected feature space. The selected feature space is the used for the remaining experiments.

3.4.3. Classifier configuration optimisation

Each classifier type has meta-parameters which must be chosen before the training phase. To select the best configuration of these meta-parameters, multiple configurations need to be evaluated. This evaluation is performed on a specific dataset with a fixed class division and feature space. Here the MCC metric (more details about the MCC in section 3.4.1) is used to make the performance based choice, but the f1 score and the confusion matrix are calculated as well. Different configuration settings are selected for each classifier type, randomly. Randomly sampling of the meta-parameters yields a better exploration of the configuration space than a grid search with the same amount of evaluations [116]. For discrete variables each option is sampled with an equal chance. Continuous integer distributions are converted in a discrete distribution with a minimum and maximum value. The other continuous variables are samples from an exponential distribution with a set scale. The Probability Density Function (PDF) of this distribution is given by $pdf = \lambda * \exp(-\lambda x)$, where the scale parameter is $scale = \frac{1}{\lambda}$

The evaluation of a classifier is done in two phases: the optimisation phase (80%) and the test phase (20%). Each of the phases uses a separated part of the training set. The division of samples is made in a stratified manner, thus although randomly selected the class balance of the dataset is preserved. In the optimisation phase k-fold cross-correlation is used to select the best classifier, because it produces more statistical valid results. It means that the optimisation part in divided into $k = 4$ parts and k repetitions of the training and scoring are done. Again the division keeps the class balance intact by using stratified selection. In each repetition one of the parts is selected as validation set, while the other parts are used to train the classifier. Each of the classifier configurations, which model's parameters are estimated with the use of the training parts, are scored on the validation set. These scores are averaged over the repetitions. The mean (MCC) scores are then used to select the best classifier. Afterwards the best configuration is retrained on the whole optimisation set and evaluated on the test part. The dataset division procedure is summarised in figure 3.11. This is necessary because by using the validation set for the selection of the meta-parameters these scores contain a bias. Comparing the validation score and the test score gives the generalisation loss, which gives some indication about the variance in the problem and the generalisation ability of the classifier.

In the cross-dataset evaluations the classifier is trained on one dataset and evaluated on another. The training datasets are AudioSet and RoadCube and the evaluation datasets are the idle and driving DriveSound sets in this thesis. The best classifier of each type is selected on the training dataset by using the classifier optimisation procedure described above. The trained classifiers are then evaluated on the held out test set of the evaluation dataset. The dataset division for the cross-dataset evaluation can be seen in figure 3.12. These held out set is the same as the held out test set used for the intra dataset classifier optimisation, thus the obtained test scores can be compared directly. In this way no information from the evaluation datasets is

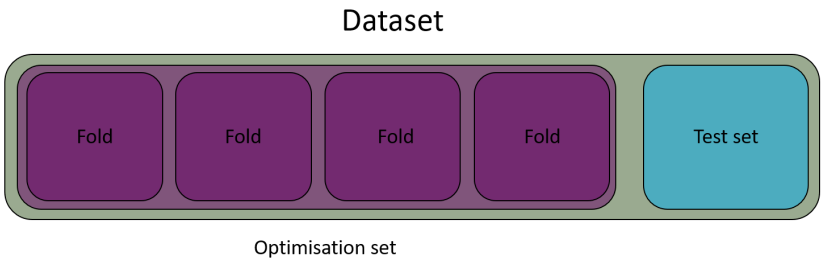


Figure 3.11: Division of dataset into parts for the intra or single dataset experiments. A test part (20%) is held out. The remaining optimisation set is used for the cross-validation with $k = 4$ folds. Each of the fold is once the validation set. The classifier is trained on the other folds and evaluated on the validation fold. The best classifier has the highest average validation score. The best classifier configuration is retrained on the whole optimisation set and evaluated on the test set.

used in the classifier.

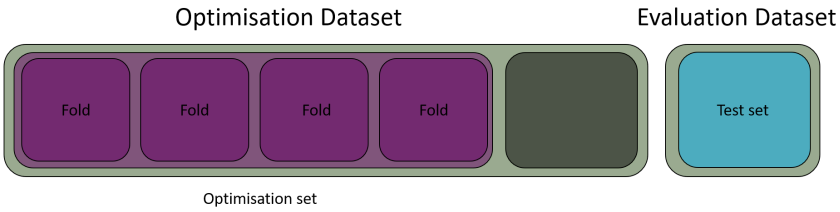


Figure 3.12: Division of dataset of the cross dataset evaluation. The best classifier is obtained through cross-validation on the optimisation set of the optimisation dataset. A test set of 20% is held out for both dataset, but the one in the optimisation dataset is not usedd (black block). The optimisation set is used for the cross-validation with $k = 4$ folds. Each of the fold is once the validation set. The classifier is trained on the other folds and evaluated on the validation fold. The best classifier has the highest average validation score. The best classifier configuration is retrained on the whole optimisation set and evaluated on the test set of the evaluation dataset (blue block).

4

Datasets

Training datasets are crucial for the performance of a classification pipeline. For vehicle detection two suitable datasets exist: AudioSet [117], a large general purpose audio dataset, and RoadCube [118] (name is given by this author), a a specialised dataset for vehicles. Both of these datasets will be used in this thesis. Another large general purpose audio dataset containing road vehicles, FreeSound [119], was not fully ready at the moment of writing, thus could not be used. UrbanSound [120], an acoustic dataset containing city sounds lacked usable vehicle classes. For the evaluation a new dataset is recorded, which aims to mirror reality as much as possible. It is called DriveSound and was recorded on the roads of Delft, The Netherlands. The data is split in idle and driving scenarios.

Dataset	author	properties	sample length	total vehicle samples	comments
Audioset	Google	large, general purpose	10 sec	78717	created using Youtube videos
RoadCube	Tu Delft students	specific, captured outdoors, multiple microphones	4 sec	383	
DriveSound	author of this thesis	captured from a car, both idle and driving scenario's	1 sec	1201	evaluation set

Table 4.1: Overview of datasets

In table 4.1 a summary is given of the two existing datasets and the newly captured one. The biggest differences are found in the amount of samples in each dataset. AudioSet has an enormous amount of samples and probably contains more variation for this reason. RoadCubecontains fewer sound fragments, but they are all known to be vehicles captured on the road. The same is true for DriveSound, but on a slighter larger scale. In each dataset the same class division is made for the experiments, namely the binary class division between the motorised vehicles and environment groups. As the different datasets have differently defined classes, which classes are part of which problem group can differ per dataset. The division of the samples can be seen in table 4.2. In both AudioSet and RoadCube more vehicles than other sounds are present. The opposite is true for the DriveSound datasets. The DriveSound-idle dataset for example has more than three times as much environment samples than vehicles. The sample length also differs between the datasets. Longer samples could means that the feature is averaged over more frames. This could smooth some in-stationary effects.

In the subsequent sections (4.1-4.3) the characteristics of the existing datasets is examined first, after which the collection and properties of the new dataset are discussed.

	AudioSet	RoadCube	DriveSound-idle	DriveSound-driving
Motorized	52795	233	149	122
Environment	25922	150	548	382

Table 4.2: The amount of samples in each class for the motorised vehicles versus environment class division. The samples are separated by dataset.

4.1. AudioSet

Audioset is a large, general purpose datasets with a many different sound classes. It was made by the sound understanding group in the machine perception research organization of Google [117]. All audio data originates from youtube.com videos and is manually annotated. As youtube.com videos are uploaded by users, it is assumed that the recordings were made with a large variation of microphones and under various conditions. Due to the datasets size and variety, it is expected that signals contain more noise on average, which might be useful for training purposes. The variety also ensures that there is little data bias, which a classifier could mistake for a pattern. The negative class is also adjacent to the vehicle classes, which could make the classification more precise. The downside is that due to the variety it is difficult to say which information generalises well. Additionally the boundary of the vehicle classes are less clear, especially the boundary between the car and car passing by class is unknown. Each of the samples is captured at 48 kHz. The average frequencies in the dataset can be seen in figure 4.1.

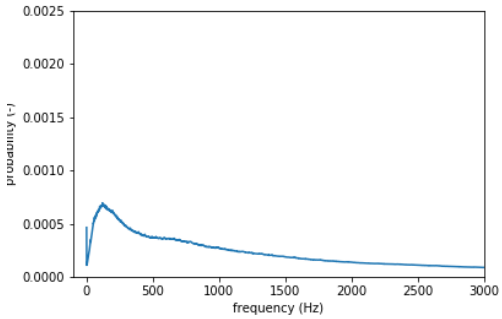


Figure 4.1: Probability density function of the average frequencies found in AudioSet.

The dataset contains at the moment of writing 2,084,320 audio fragments of 10 seconds divided over 527 classes. In the original dataset samples have multiple labels, but for this research only samples holding a single vehicle label were selected. This means that a sample containing speech and car sounds is included in the car class, but a sample containing both the car and motor vehicle (road) labels were excluded. Each sample is the extracted dataset belongs to a single class and all reported sample amounts are of this extracted set. It also means that the "outside, urban or man-made" class contains no distinguishable motorized road vehicles.

The relevant classes and samples contained in them are summarised in table 4.3. Two classes, the car and the "outside, urban or man-made classes, are much larger than the others. The specific divisions in problem groups as specified earlier can be seen in table 4.4. In the motorised vehicles versus environment division, more motorized vehicles (67.1%) than environment (32.9%) are present.

4.1.1. Preparation

Audioset was only available in VGG features from the original website, which are based on MFCC features. To get the original sound files the youtube.com video name was retrieved for each relevant sample. Each sample containing multiple labels from the classes selected here was excluded. Subsequently the audio of each of these videos was downloaded. Some of the youtube.com videos were not available any more, due to a variety of reasons like copyright issues, thus could not be downloaded. The sample amounts reported in this chapter are the samples which actually could be downloaded. Some of the sounds were in stereo (2 channel audio) or multi channel format. These samples were converted to mono (single channel) format before the feature extraction.

Class	Sample amount
Bus	4210
Car	28271
Car passing by	1852
Emergency vehicle	4191
Motorcycle	5361
Motor vehicle (road)	1135
Outside, urban or man-made	25922
Truck	7774

Table 4.3: Samples per class in AudioSet. The reported sample amounts only contains the samples which hold a single label from this list. Samples holding multiple of these class labels were excluded.

Group	Classes	Total samples
Motorized	Bus, Car, Car passing by, Emergency vehicle, Motorcycle, Motor vehicle (road), Truck	52795
Environment	Outside, urban or man-made	25922

Table 4.4: Class division into groups for the AudioSet including the total samples number found in each group.

4.2. RoadCube

RoadCube is a dataset specifically captured for road vehicle classification. It was captured alongside the road using eight microphones mounted in the shape of a cube. During recording the microphone cube was placed at $0.25m$ from the road. Fragments containing vehicles passing by were separated from the vehicle-less audio. The audio was captured from 8 different locations, with speed limits of 30 and $50km/h$ [118]. The dataset consist of different vehicle classes, namely bike, scooter, truck, van and car and captures were made on different road surfaces. Additionally there is a "no sound" class which consist of roadside noise, but is relatively quiet. This could mean that classifiers trained on this dataset will classify any sound as a vehicle. During the collection of this dataset and the DriveSound the same microphones were used although in a different spatial configuration. There could thus be some microphones specific bias in both datasets. Each sample is $4s$ long and is captured with a sampling frequency of 44.1 kHz. A sample always has a single label. The average frequencies found in the dataset can be seen in figure 4.2.

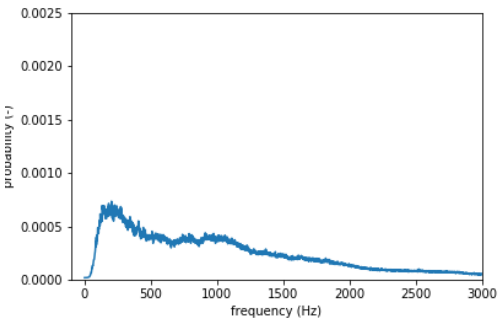


Figure 4.2: Probability density function of the average frequencies found in RoadCube.

In table 4.5 a summary of the classes in this dataset can be seen. The specific divisions in problem groups as specified earlier can be seen in table 4.6. There are much more cars in this dataset than other vehicles. This could be a realistic representation of the balance in vehicle types found on the roads (in the Netherlands at least). The motorised vehicles (60.8%) versus environment (39.2%) division is more balanced than AudioSet.

Class	Sample amount
Bike	77
Car	151
Scooter	31
Truck	21
Van	30
No Sound	73

Table 4.5: Samples per class in RoadCube.

Group	Classes	Total samples
Motorized	Car, Scooter, Truck, Van	233
Environment	Bike, No Sound	150

Table 4.6: The classes of the RoadCube grouped by the motorized vs environment division along with the total samples in each group.

4.3. DriveSound datasets

A new dataset, DriveSound was captured for the evaluation of the existing datasets. It aims to capture the real world situation as realistically as possible. The captured data was therefore not filtered or cleaned after collection. The balance in the vehicle classes is representative for the real balance as encountered on the road. Recordings were made during which the listener vehicle was idle and when it is driving. They are split into two datasets with the idle scenarios in one and the driving scenarios in the other set. These datasets are discussed in the upcoming sections. Details on the collection of the dataset can be found in section 4.3.3.

The recordings were captured with a sampling frequency of 48 kHz. Each fragment in the dataset has a duration of one second. The average frequency content in a samples can be seen in figure 4.3. Eight microphones were mounted on the listener vehicle, but only six functioned correctly, thus each moment in the dataset is captured in six sound fragments. The distance between the microphones is not taken into account, thus the moment of capture is assumed to be equal.

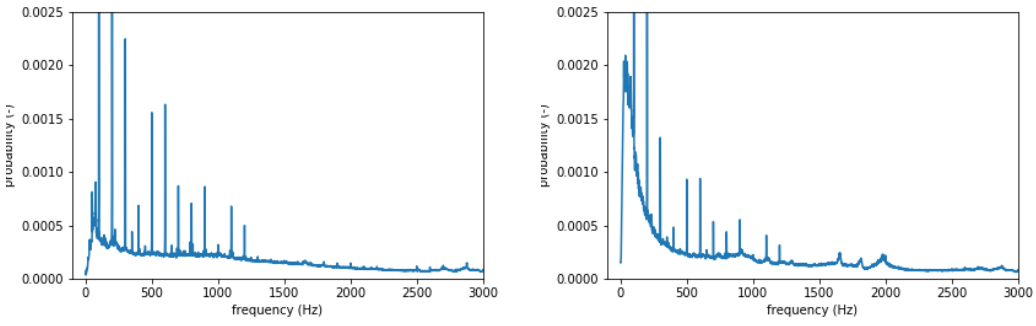


Figure 4.3: Probability density function of the average frequencies found in RoadCube. **left:** idle set, **right:** driving set

4.3.1. Idle dataset

To construct this dataset the data from the five idle scenarios were used. The environment class is the largest class of them all. By far most of the vehicles present in the dataset are cars, followed by vans. Many different vehicle classes are present. The motorized class (21.4 %) is much smaller than the environment (78.6 %) class.

4.3.2. Driving dataset

The driving dataset is smaller than the idle dataset, because only three scenarios were used. Still the environment class is much larger than other classes and the car class is the biggest vehicle class. Again the motorised (24.2

Class	Idle set	Driving set
Bus	1	2
Car	110	99
Motorcycle	3	-
Environment	548	382
Excavator	-	1
Mini-truck	-	3
Pick-up	1	-
Scooter	5	1
Touring car	1	-
Tricycle	1	-
Truck	6	1
Van	21	15

Table 4.7: The amount of samples per class in the DriveSound datasets.

Group	Classes	Total samples
Motorized	Bus, Car, Motorcycle, Pick-up, Scooter, Touring car, Tricycle, Truck, Van	149
Environment	Environment	548

Table 4.8: The classes of the DriveSound-idle dataset grouped by the motorized vs environment division along with the total samples in each group.

%) class is smaller than the environment (75.8 %).

Group	Classes	Total samples
Motorized	Bus, Car, Motorcycle, Excavator, Mini-truck, Scooter, Truck, Van	122
Environment	Environment	382

Table 4.9: The classes of the DriveSound-driving dataset grouped by the motorized vs environment division along with the total samples in each group

4.3.3. Collection

The recordings were done on eight different locations (excluding a pilot recording) in the remainder of this report the location including the circumstances will be called a scenario. In five of the eight scenarios the listener vehicle was idle in a parking spot adjacent to the road. In the other scenarios the listener vehicle was being driven along the road. The speed limits on the roads of the different scenarios varied between 30, 50 and 70 km/h. All of the scenarios were captured on the same day on which the weather was sunny and dry. A list of the scenarios including their characteristics can be found in table 4.10.

The listener vehicle was a prius hybrid car. It has a petrol engine, which acts as a generator. This generator will occasionally be turned on, even when the vehicle is idle. It will cause an engine sound at unpredictable moments, which must not be classified as a vehicle. Eight microphones are mounted on the roof of the vehicle (see figure 4.4). In this configuration the sounds captured by the microphones may contain a relatively high portion of wind noise.

For the experiments in this thesis the vehicles were annotated when they were the closest to the listener vehicle. This was done because of the targeted motions are constant motions and these are the most audible when the target vehicle is closest. Other variants of annotation aiming to capture other types of information are also possible. The annotation was done by using lidar system and two front facing camera's to establish the ground truth. Each time a vehicle passed by the time, vehicle type and vehicle location was noted down.

Figure 4.4: Recording one of the idle scenarios of the DriveSound. Eight microphones (white circles) are mounted on the roof of the vehicle along with a Velodyne lidar scanner, used to determine the ground truths.

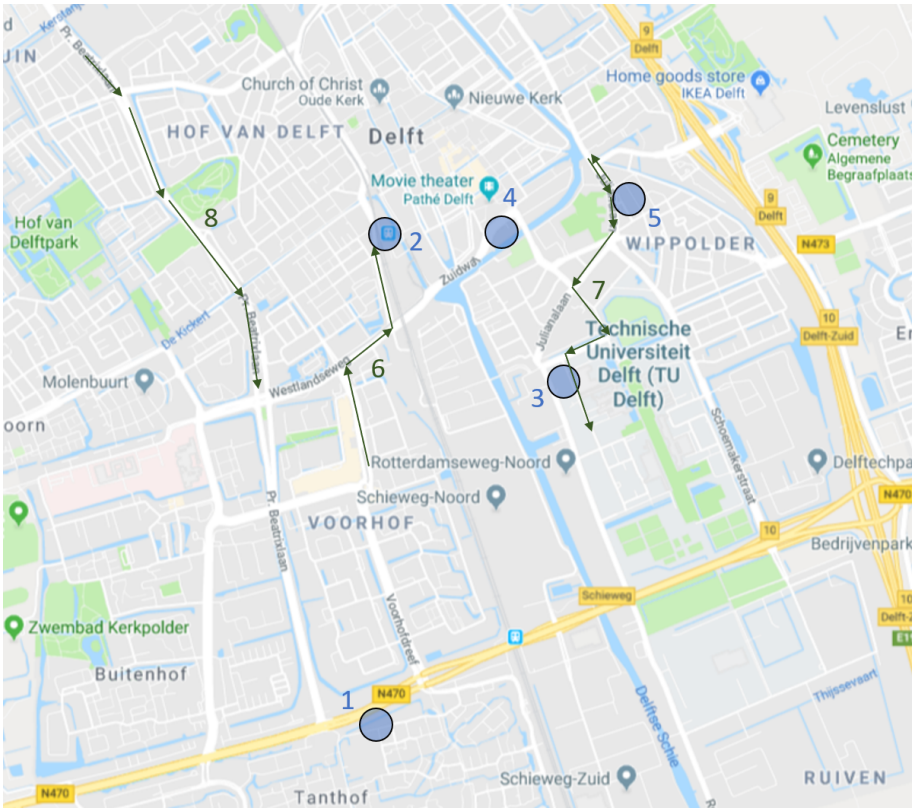


Figure 4.5: DriveSound scenario locations. There are five idle scenarios, which are denoted by the blue dots. The other three scenarios are with a moving listener vehicle. The driven route is presented with green arrows. The numbers correspond to the scenarios in table 4.10. The map is taken from `maps.google.com`

It is estimated by the author that the timing error is in the order of 0.1 second.

	Scenario	listener motion	duration (mm:ss)	speed limit	vehicles annotated
1	gas station	idle	06:05	70 km/h	105
2	behind station	idle	02:21	30 km/h	26
3	hague college	idle	05:51	50 km/h	13
4	near jumbo	idle	04:33	50 km/h	72
5	science centre	idle	04:41	50 km/h	64
6	past station	driving	03:05	50 km/h	50
7	to 3me	driving	03:30	30 km/h	46
8	subsidiary road	driving	01:19	70 km/h	30

Table 4.10: DriveSound datasets - Summary of the captured scenarios

Each of the scenarios is a recording, but for the classification pipeline small samples are required. The split into vehicles and no vehicles is done with the help of the annotations. A sample of 1s second is extracted around the annotated time. This sample belongs to the class which was annotated. Multiple class samples can overlap. After each of the vehicles is extracted, the environments samples are extracted. The parts of the signal which are not included in a class are divided into the environment samples of 1s. Between the vehicle class and a negative class is a margin of 2s to get a contrast. A schematic of this division can be found in figure 4.6.

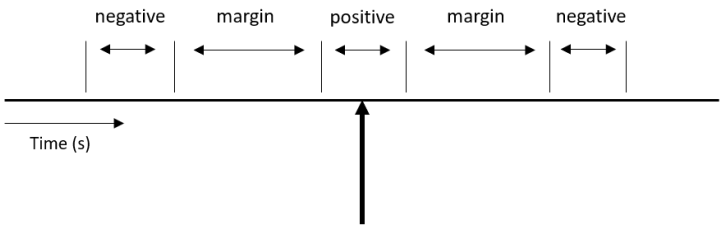


Figure 4.6: Recordings are divided into vehicle (positive) and environment (negative) samples of 1s. The time of the vehicle classes is annotated (big arrow). Vehicle and no-vehicle classes are separated by a margin of 2s. Vehicle samples can overlap with other vehicle samples.

5

Experiments

A classification pipeline can have different components, namely the dataset, the feature space and the classification algorithm. The usefulness of these components needs to be evaluated. First a feature selection is done using the random forest classifier (see section 5.1). Subsequently the performance of the classification pipeline will be measured by performing a classifier optimisations, both on each dataset as cross dataset. An overview of the performed classifier optimisation experiments can be found in table 5.1. First the the intra or single dataset optimisations are performed (diagonal of the table). These trained classifiers are then used to classify samples from the from the DriveSound datasets. More information about the datasets can be found in chapter 4: *Datasets*.

		Evaluation set			
		AudioSet	RoadCube	prius-idle	prius-driving
Optimisation set	AudioSet	1-intra		5	6
	RoadCube		2-intra	7	8
	DriveSound-idle			3-intra	9
	DriveSound-driving				4-intra

Table 5.1: Overview of the performed classifier optimisations. In each case the feature space is set by the feature selection and the class division is motorized versus environment.

The commonalities shared between all of the experiments are the class division, the performance metric, the feature extraction method and the classifier types. The motorized vehicles versus environment class division is used, which is a binary classification problem. The motorized vehicles class contains one or more simultaneous vehicles, which are close by. The environment class on the other hand only contains background noise. The former class consists of samples with one or more sound emissions of vehicles, while the environment class only contains background noise. Specifics on the composition of each class for each dataset can be found in chapter 4: *Datasets*. The same performance metric is used for all experiments, namely the MCC. It is chosen because it is well suited for unbalanced class divisions. More information on this metric can be found in section 3.4.1: *Performance metrics*. Each dataset consists of samples in the wav file format. These need to be transformed to features to be usable for classification. A sample is divided in frames of 2048 samples, regardless of the length of the signal. The feature are extracted for each frame and averaged over the whole signal afterwards. All types mentioned in table 3.1 were extracted. In the feature selection phase the feature types were included or excluded in the features space for the experiments afterwarsd

In this thesis four types of classifiers are used: SVM with kernels RBF, linear and polynomial, GMM classifier, MLP and Random forest classier. Each of them has meta-parameters which must be chosen beforehand and cannot be trained. Determining the optimal meta-parameters is the main aim of the classifier optimisation. In this thesis the possible classifier configurations are randomly samples from meta-parameter distributions. These distributions for each parameter for each type of classifier are shown in table 5.2. Three types of distributions exist: choice, integer and exponential. Both choice and integer distributions are discrete and

offer only a limited number of options. Each of the options has an equal chance of being sampled. The exponential distribution is continuous, thus an almost infinite amount of values can be sampled. The scale argument sets the order of magnitude, which is sampled most.

Classifier	parameter	distribution type	range/options
SVM	kernel type	choice	Linear, RBF, polynomial
	C - training parameter	exponential	scale=100
	γ - multiplier	exponential	scale=.1
	r - bias coefficient	exponential	scale=.1
	d - polynomial degree	integer	2-11
GMM	component amount	integer	1-20
	covariance matrix shape	choice	diagonal, symmetric, tied, full
MLP	hidden layer amount	integer	1-20
	nodes per hidden layer	integer	5-200
	activation function	choice	tanh, relu
	α - regularization penalty	exponential	scale=.0001
	solver	choice	L-BFGS, adam
	initial learning rate	exponential	scale=.001
Random forest	estimator amount	integer	2-200

Table 5.2: Overview of meta-parameters distributions. A classifier configuration is randomly sampled from the distributions corresponding to the classifier type. There discrete (choice, integer) and continuous (exponential) parameter distributions. In the case of the discrete distribution each of the

This chapter continues with sections presenting the results of the experiments. In section 5.1 the feature selection results are shown. Here the subset of feature types is chosen, that will be used in the classifier optimisation experiments afterwards. The included and excluded features are summarized in table 5.6. After the feature selection the intra or single dataset classifier optimisation results are shown in section 5.2. The classifiers trained on AudioSet and RoadCube are then used on the DriveSounds in section 5.3.

5.1. Feature selection

The first step in the optimisation of the classifier pipeline, is to to select the most useful features. In this step feature types are included and excluded, based on the importance measure from the random forest classifier. For a detailed description about the feature selection procedure see section 3.4.2. Frequency vector features, or frequency magnitudes, are excluded even before the selection, because they have too many elements. Another reason is their feature length which varies with the sample rate of the sound signal. Due to RoadCube having a different sample rate than the other datasets, this would be a problem for the classifiers. The features used in the selection, called abstract features because they represent a property of the signal, can be found in table 5.3. Each of the feature elements is given a general index in this experiment, which makes it easier to plot the results. Additional information about the different feature types can be found in 3.2. Afterwards the feature selection procedure is repeated with only the frequency vector features to get more insight in the important frequencies for vehicle classification.

5.1.1. Abstract feature selection

To select the most important features, the elements with a performance above the threshold of 0.75 are examined. This threshold means that these elements are within a 25% importance of the most important feature on that dataset. These most important feature elements are shown in 5.4. There it can be seen that MFCC elements are not only present in all of the most important feature elements, but one element has an importance within 90% of the maximum value. Furthermore the LPC features seem to be important, but only on the DriveSound sets. Both AudioSet and RoadCube are dependent on few features elements. AudioSet has fewer element with a high importance and peaks than RoadCube. AudioSet only has a single feature

Feature name	feature length	feature indices (0-based)
Energy	1	0
Energy entropy	1	1
LPC feature	8	2-9
Zero crossing rate	1	10
Chroma	12	11-22
Harmonic line	1	23
MFCC	13	24-36
Spectral centroid and spread	2	37-38
Spectral entropy	1	39
Spectral rollOff	1	40

Table 5.3: Overview of all abstract features extracted. The general index listed here is used for other graphs and tables in this section.

element above the threshold (MFCC (4)), with Energy entropy and MFCC (1) somewhat below it. RoadCube is dependent on relatively simple features. The most important element is the MFCC element which indicates power in the signal, followed by the spectral spread. The spectral centroid and the first LPC element are just below the threshold.

Both the DriveSound datasets have more varied values for the relatively unimportant elements. The MFCC (4) element has a large importance in the idle set, but is unimportant in the driving set. Multiple LPC elements are important in both the idle and driving case. The prime example is LPC (3) which has a large importance in both sets In DriveSound-driving more elements, especially of the LPC kind, are important. It is the only dataset, where the spectral spread is has an importance above the threshold. This probably is due to the lack of elements which are much more important than others. Finally in the driving set of DriveSound more features elements have a high importance. This includes elements of the Chroma type, which are only important here.

Dataset name	feature element	element index	importance
AudioSet	MFCC(4)	27	1.0
RoadCube	MFCC(1)	24	1.0
	Spectral centroid and spread(2)	38	0.99
DriveSound-idle	LPC feature(3)	4	.86
	MFCC(4)	27	1.0
DriveSound-driving	LPC feature(3)	4	.84
	LPC feature(4)	5	1.0
	LPC feature(7)	8	.81
	Chroma(8)	18	.93
	MFCC(6)	29	.90
	MFCC(11)	34	.77

Table 5.4: Summary of the most important feature elements as determined by the random forest classifier. Feature elements above the threshold of 0.75 are listed. The features are sorted per dataset on their feature index. After each feature name the 1-based feature type index is presented. The element index column is the general index used in for the experiment (for example in table 5.3). The importances are peak normalised per dataset, thus values range from 0 to 1.

The test scores of the best classifier for each dataset are shown in table 5.5. Abstract features score higher on each dataset except for the RoadCube set. In the last case the scores are close and very high, which makes it more likely that this difference is part of the possible variance. It can furthermore be seen that the scores on RoadCube are very high, on AudioSet scores are lower and on the DriveSound datasets the lowest scores

are attained.

Dataset	Abstract features	Frequency feature
AudioSet	.40	.32
RoadCube	.97	1.0
DriveSound-idle	.24	.20
DriveSound-driving	.27	.20

Table 5.5: Overview of the test scores obtained by the best random forest classifier on the different datasets both for the abstract features and the frequency vector features.

In the graphs in figure 5.1, the importances of all abstract features can be seen for each dataset. Each feature element is numbered to make the graphs more readable. The conversion from these indices to the feature names is found in table 5.3. Both Audioset and RoadCube only have a small amount of features with a high importance. RoadCube has many feature elements which have a importance close to zero, including the Chroma features, the Harmonic line, the MFCC features, apart from the first element, and the spectral entropy. The other datasets have a much higher minimum importance, which means that there exist much less dominant feature elements. The DriveSound-idle dataset depends only on a few features, while the DriveSound-driving feature importances are much closer together.

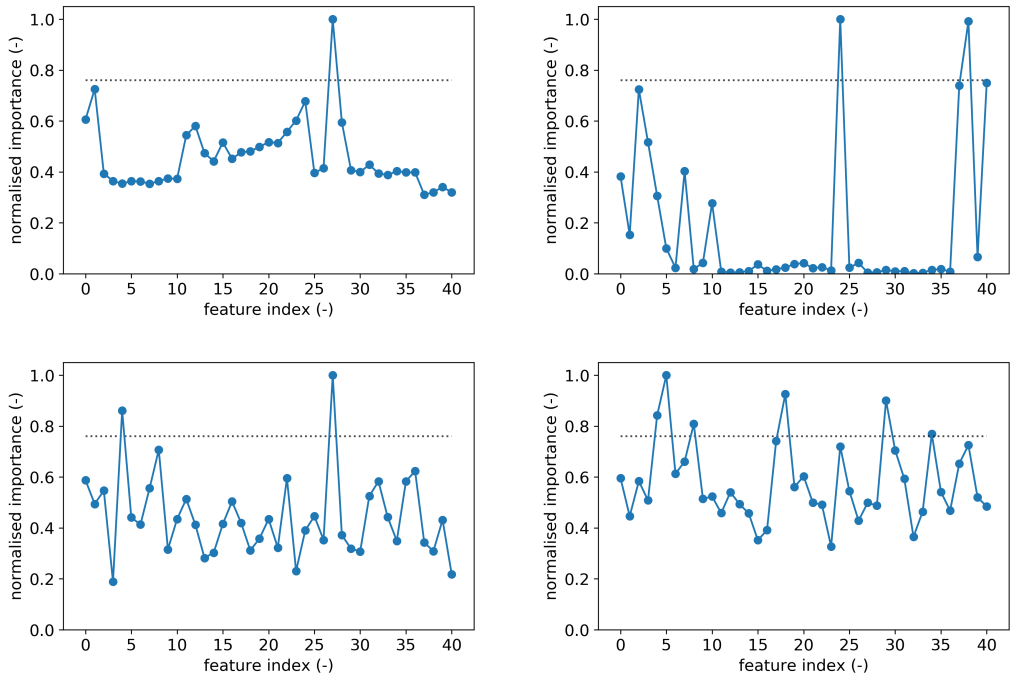


Figure 5.1: The relative importances of the abstract feature elements, captured per dataset. They are peak normalised per dataset. The lines connecting the dots are just a visual aid, the elements' importances have no (linear) relationship. The dotted line is the threshold of 0.75. The indices of the feature elements correspond to the names in table 5.3.
left-top: AudioSet, **right-top:** RoadCube, **left-bottom:** DriveSound-idle, **right-bottom:** DriveSound-driving

5.1.2. Frequency feature selection

To get more insight in the important frequencies for each dataset, feature selection is performed separately on the frequency bin features. The results of the frequency vector feature selection can be seen in figure 5.2. It can be seen that the classifications in AudioSet depend on low frequencies, while RoadCube depends a lot on frequencies around 1600 Hz. In the latter case it can also be seen that frequencies closely together can have high differences. The relatively low amount of samples can cause this, because the FFT is a histogram. With few samples an many histogram bins, this can mean that some bins are not present in the dataset. Both

of the DriveSound sets also show a dominant frequency around 1600 Hz, but the driving set contains more variation in the importances.

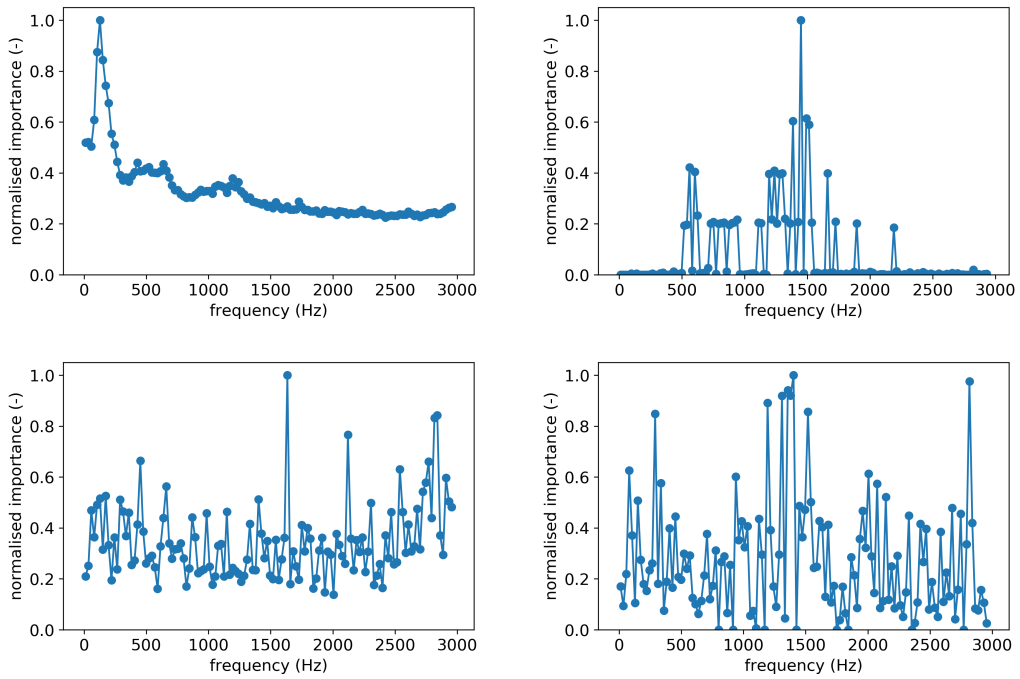


Figure 5.2: The relative importances of the frequency vector feature elements, captured per dataset. They are peak normalised per dataset. The lines connecting the dots are a linear interpolation of the results.
left-top: AudioSet, **right-top:** RoadCube, **left-bottom:** DriveSound-idle, **right-bottom:** DriveSound-driving

5.1.3. Conclusion

Abstract features quantify a property of the sound signal. Compared to the frequency histogram and especially the raw signal, their information density is high. The cost is the additional processing required to extract these features. For the datasets in this thesis this is worth the effort, because the random forest classifiers score higher with them and the features have more than three times less elements.

Combining the results form the abstract and frequency bin feature selection a few conclusions can be drawn. As mentioned before the LPC features are important for all the datasets, except for AudioSet. Additionally important frequencies in AudioSet are very different from the other sets. There are probably many samples containing engine noises or these are most important, because engines emit frequencies in the range of the low hundreds Hz (see section 2.2.2: *Acoustic vehicle signatures*). The frequencies show that lower frequencies have a higher importance, which indicates that there is relatively much engine noise present in the dataset. Another explanation is that the engine noise is the only discernible difference between the classes. If engine noise is the only usable property in the dataset, the low scores on the DriveSound would indicate that engine noise alone is not enough for vehicle detection. The selection also indicates that the classification on the RoadCube is a relative simple power based problem. The most important MFCC element on RoadCube, the first element, is namely a measure of the power of the signal. Additionally on RoadCube relatively simple features are important, while many sophisticated features have a relative importance close to zero. RoadCube has a low amount of noise in the negative and the high power contrast between the classes, which is probably the reason why simple features are preferred. The more sophisticated features could be eclipsed by the simpler frequency statistics (centroid, spread, roll-off). It is also shown in the frequency importance, where a single frequency is important.

In the idle set of DriveSound peak importances of both abstract and frequency feature elements are present. The most important abstract feature in the idle set, MFCC(4) is not The LPC features, which are time domain based, keep their importance. It is thus likely that frequency domain based features are susceptible to the motion of the vehicle, while time features are robust. The results from the other feature selection procedure with the frequency bin features, show importance peaks in the idle set near: 400 Hz, 1600 Hz, 2100

Hz and 2800 Hz. Th 1600 Hz and 2100 Hz peaks are not present in the driving set any more. In the driving sets multiple frequencies in the range from 1100 Hz to 1500 Hz are more important, which could indicate that the earlier peak of 1600 Hz has shifted due to the listener vehicle's motion.

Included	Excluded
LPC features, Chroma features, MFCC, Spectral centroid, Spectral spread	Energy, Energy entropy, Zero crossing rate, Frequency bin features, Spectral entropy, Harmonic line

Table 5.6: A list of the features included and excluded in the feature space for the classifier optimisation. When a single feature element was selected, the whole (multi-element) type list selected. The amount of elements of all included feature types is 35.

MFCC elements are one the most important in all datasets, although different elements are important for different datasets and are therefore included in the feature space. LPC features have a relatively high importance in all of the dedicated vehicle detection datasets (below threshold on RoadCube). This means that some part of the vehicle's acoustic signature is captured and can be used, thus it is included. More than one element of this type has a high importance, thus all of them are included. Chroma features are only important on the driving set of DriveSound. For the driving scenario's Chroma features might thus be important, thus they are included. The spectral centroid and spread have an importance near or over the theshold on the RoadCube and are therefore included. A overview of the included and excluded feature types is given in table 5.6. In the upcoming sections this optimised feature space is used for the experiments.

5.2. Intra experiments

After the feature space optimisation in the previous section (see table 5.6 for the included features), the performance of the classifier types should be evaluated. As a first step the classifier optimisation is performed for each of the datasets separately. These experiments can be found on the diagonal of the classifier optimisation index table (5.1). They act as a performance baseline, because it is unlikely that captured information from another dataset will be better suited for classification, than information about the current dataset. The amount of configurations evaluated are 300, 80, 200, 100 for the SVM, GMM, MLP and random forest classifiers, respectively. More information about the procedure for the classifier optimisation can be found in section 3.4.3: *Classifier configuration optimisation.*

5.2.1. Dataset comparison

The test scores of the best classifiers can be seen in table 5.7 along with the classifier type that attained it. It can be seen that there are large differences between the datasets. On RoadCube the classifiers are able to achieve a near perfect score, while the classifiers are only moderately able to correctly classify the test samples from the other datasets. This is due to a lack of generalisation, because the training scores of the best classifiers are high. It must be noted that the validation score of the best classifier on DriveSound-idle was much lower that the test score, namely 0.27 (see table 5.8).

	AudioSet	RoadCube	DriveSound-idle	DriveSound-driving
test score (MCC)	.41	.90	.43	.38
classifier type	SVM-RBF	MLP	GMM	GMM

Table 5.7: Test scores for each single dataset experiment along with the type of classifier that attained it. The classifier with the best validation score was selected to do the test. It was retrained on the whole optimisation part of the dataset. The test score for the DriveSound-idle is 0.15 higher than the mean validation score of 0.27. It could thus be an anomaly. The best classifier configuration can be found in Appendix 7

	Classifier type	validation score	test score	generalisation loss
AudioSet	SVM	**.40**	.41	.00
	GMM	.30	.31	.00
	MLP	**.40**	.41	-.01
	Random forest	.37	.38	-.01
RoadCube	SVM	.96	.97	-.01
	GMM	.96	.89	.07
	MLP	**.97**	.90	.07
	Random forest	.96	.95	.01
DriveSound-idle	SVM	.25	.22	.03
	GMM	**.27**	.43	-.15
	MLP	.25	.22	.03
	Random forest	.23	.19	.03
DriveSound-driving	SVM	.34	.56	-.21
	GMM	**.40**	.38	.03
	MLP	.36	.36	-.00
	Random forest	.33	.48	-.15

Table 5.8: Mean validation score, test score and generalisation loss (MCC) for the average best performing classifier on the validation set separated by classifier type and dataset. This best classifier's is retrained and used on the test set. The difference between the validation and test score is the generalisation loss.

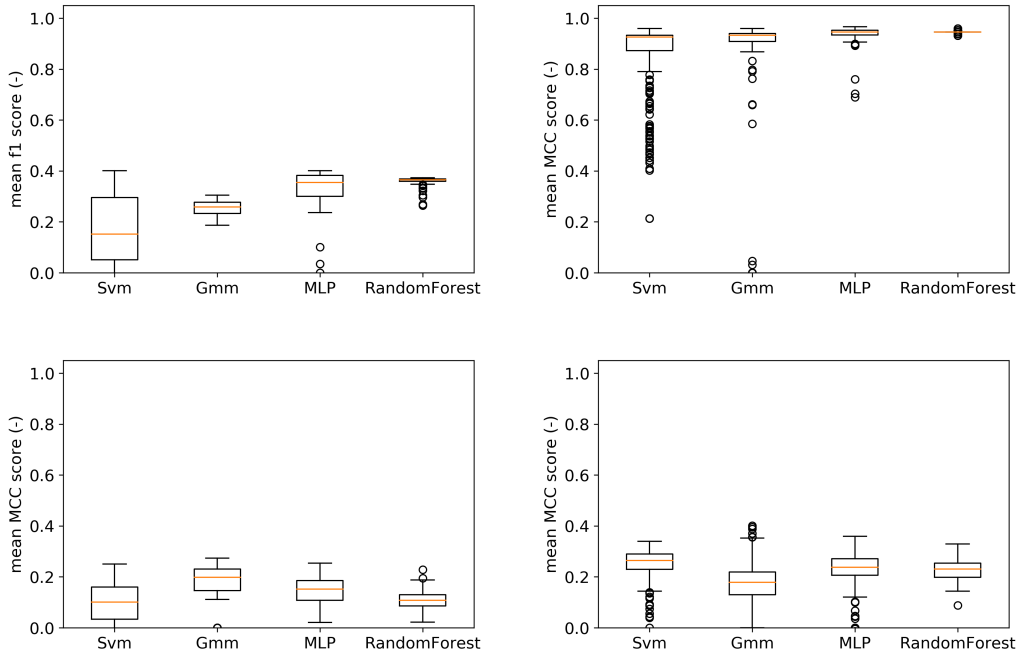


Figure 5.3: Mean validation score (MCC score) of each classifier configuration for each dataset with the motorized versus environment class division separated by classifier type. The mean score is taken over the $k = 4$ folds of the cross-validation.
Boxes represent 50 % of the data. The yellow line is the median value. The length of the whiskers is 1.5 times the length of the box lengths, but can be shorter if the minimum/maximum of the values is closer. Points are outliers.
left-top: AudioSet, **right-top:** RoadCube, **left-bottom:** DriveSound-idle, **right-bottom:** DriveSound-driving

5.2.2. Classifier types

In table 5.8 the scores of the best classifiers of each type are presented. It can be seen there that the MLP is often the best classier or in any case its test scores are near the best. The best SVMs score near the best scores on the best scores and it even one of the best types on AudioSet. On the opposite side, the GMM classifier also performs well, but not on AudioSet. The random forest classifier is performing worse than the other types on every dataset, although the GMM performs worse on AudioSet. The generalisation of a classifier differs between the types of classifiers. It is relatively low and consistent for the SVM and random forest classifiers, but the GMM and MLP perform more irregularly. The negative GMM loss on DriveSound-idle set and the one of the SVM on the driving set both seem anomalies. The first has a standard deviation on the cross-validation folds of 0.15, which explains the surge in performance. The SVM only has a standard deviation of 0.03, thus that surge must be just lucky. If not only the best classifiers, but also other configurations are considered (seen in figure 5.3), it can be seen that the random forest classifier performs the most consistent. On the other hand are many SVM configurations, which perform relatively poor. This is due to the linear kernels variants, which have scores near 0.

The training time can be a factor to consider, when training classifiers, especially on large datasets. The average training time per fold of the cross-validation can be seen in figure 5.4. The MLP classifier takes by far the longest to train. A positive note is that on the AudioSet the training time is about a factor of 200 bigger than on RoadCube, while 2000 times as many samples are used. Only the SVMs come close when they are trained on AudioSet. Their training times appears to increase exponentially with the size of the dataset.

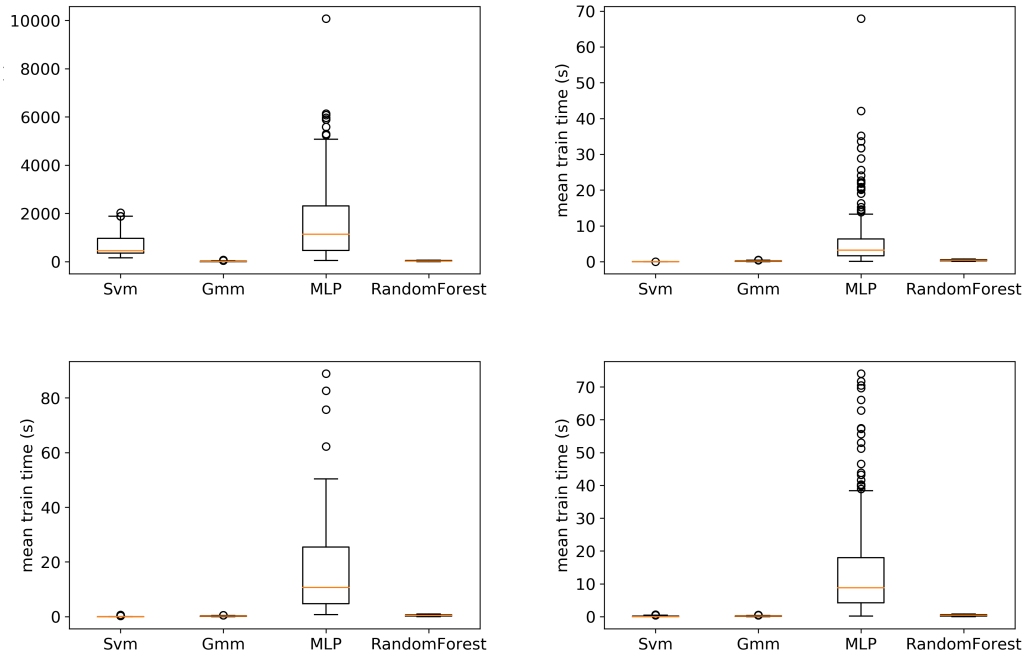


Figure 5.4: Mean training time in seconds for each classifier configuration for each dataset with the motorized versus environment class division separated by classifier type. The mean is taken over the $k = 4$ folds of the cross-validation.
Boxes represent 50 % of the data. The yellow line is the median value. The length of the whiskers is 1.5 times the length of the box lengths, but can be shorter if the minimum/maximum of the values is closer. Points are outliers.
left-top: AudioSet, **right-top:** RoadCube, **left-bottom:** DriveSound-idle, **right-bottom:** DriveSound-driving

5.2.3. Conclusion

The test scores (table 5.7) of the best classifiers give an indication of the difficulty of the classification task for each dataset. According to the test scores, AudioSet is a difficult to classify, while RoadCube is much easier. The DriveSound datasets are expected to be more difficult for a classifier than RoadCube, because it was captured in a less controlled environment, but less or equally difficult than AudioSet. This is true for the driving dataset, but not for the idle set, which performs worse. It can mean that environment noise has a large influence on the idle dataset, but not on the driving set. When the listener vehicle is driving, it is less

susceptible to a single external noise source, but more ego-noise is produced.

5.3. Cross-dataset experiments

In this section the results for the cross-dataset experiments are presented. Classifiers are optimised first on the training dataset and are afterwards evaluated on the test part of evaluation dataset (DriveSound: idle and driving). The procedural details can be found in section 3.4.2: *Random forest feature selection*. These best classifiers are reused from the earlier single dataset experiments.

The results for the cross-dataset evaluation on DriveSound-idle can be seen in table 5.9. All of the classifiers trained on AudioSet perform poorly, especially the GMM. The GMM classifier trained on the RoadCube on the other hand is the best performing classifier. It has few false negatives, while keeping the false positives limited. It performs better than the classifier trained on the evaluation dataset, DriveSound-idle. On the idle set, the cross dataset classifiers focus more on the motorized classes and limiting the false negatives, while the idle set itself focusses more on the environment classes. Both the GMM and the MLP classifiers perform well, when they are trained on the RoadCube, while the SVM and the random forest classifier have a reduced performance. On the DriveSound-idle this is the other way around. It could be that less noisy data is favoured by these kinds of classifiers.

	Classifier type	test score (MCC)	confusion matrix
AudioSet	SVM	**.19**	28 2 81 29
	GMM	.06	13 17 40 70
	MLP	.18	28 2 83 27
	Random forest	.15	30 0 99 11
RoadCube	SVM	.15	26 4 78 32
	GMM	**.34**	24 6 43 67
	MLP	.31	23 7 43 67
	Random forest	.26	19 11 36 74
DriveSound-idle	SVM	.28	13 17 17 93
	GMM	.23	14 16 24 86
	MLP	.14	7 23 13 97
	Random forest	**.29**	6 24 3 107

Table 5.9: The results of the cross-dataset experiment on DriveSound-idle. The confusion matrix shown here has the convention [TP, FN; FP, TN]. For more information see table 3.2

Table 5.10 shows the experimental results for the evaluation on the DriveSound-driving. It can be seen that all classifiers from both AudioSet and RoadCube are performing less or equally well than a random guess. The DriveSound-idle classifiers are simply guessing that almost everything is not a vehicle. The best DriveSound-driving classifier on the other hand is able to score well on its own test set. The classifier types from AudioSet and RoadCube are all performing below the random guessing score. Comparing them does not make that much sense. The SVM and in lesser extend the random forest classifier are performing well on the driving dataset. They are able to cause few false positives, while the SVM is able to classify more than half of the vehicles correctly. Again the DriveSound classifier is focussed on classifying the positives well.

5.3.1. Conclusion

The RoadCube dataset has shown its potential for the classification of vehicles, when the listener is stationary. The classifiers trained on this dataset perform much better than those trained on AudioSet and even those from DriveSound-idle. This indicates that a specific, less noisy dataset is useful for this purpose. When the listener is moving it changes the story however. Then none of the classifiers trained on other datasets is able to discern between the a vehicle or no vehicle. Especially the DriveSound-idle simply guessed almost everything to be negative. The bad performance could be explained with the feature importances found in the feature selection (figure 5.1). The other datasets are dependent on a few features, while the DriveSound-driving set dependent on multiple. Both the frequency shifts caused by the motion and the added noise are possible causes for the reduced performance.

The SVM classifier is performing well on the DriveSound datasets. Especially on the driving dataset it outperforms the others. This is another indication that this type of classifier handles noisy datasets relatively well. The GMM and the MLP classifier are working relatively well on the RoadCube when it is evaluated on the idle set. In this dataset more vehicles are present than environment samples, which cause the classifier to have a preference to classify a vehicle.

	Classifier type	test score (MCC)	confusion matrix
AudioSet	SVM	-.20	17 7 68 9
	GMM	-.13	7 17 34 43
	MLP	**-.10**	18 6 65 12
	Random forest	-.18	20 4 73 4
RoadCube	SVM	-.05	14 10 49 28
	GMM	-.09	11 13 43 34
	MLP	-.11	9 15 39 38
	Random forest	**.00**	11 13 35 42
DriveSound-idle	SVM	**.05**	3 21 7 70
	GMM	-.13	0 24 5 72
	MLP	-.07	0 24 2 75
	Random forest	-.05	0 24 1 76
DriveSound-driving	SVM	**.56**	13 11 4 73
	GMM	.39	11 13 8 69
	MLP	.36	7 17 3 74
	Random forest	.48	9 15 2 75

Table 5.10: The results of the cross-dataset experiment on DriveSound-driving. The confusion matrix shown here has the convention [TP, FN; FP, TN]. For more information see table 3.2

6 Conclusion

In this thesis the classification of acoustic signatures of motorized road vehicles was investigated. Two datasets, AudioSet and RoadCube, are evaluated along with commonly used acoustic feature and classifier algorithms. First the most important features were selected based on the random forest classifiers importance measure. Afterwards the different classifier configurations were evaluated, first on each dataset separately and afterwards on the evaluation dataset DriveSound.

6.1. Discussion

The set-up of the classifier optimisation can influence the results and almost always there are improvements possible. In this case the feature selection did not take into account that feature elements can influence the importance from other elements. The random forest feature selection does not take this into account. A better, but more computationally intensive method can be found in [121], where many feature type combinations are tried iteratively.

The validation scores on the DriveSound datasets were mediocre at best. It must be noted that the samples in the dataset are obtained from the real world under difficult classification conditions. Other reasons for the low scores can be the relatively short fragment lengths of 1s with respect to the other datasets. Increasing the length would have limited the real-time application of the classification pipeline and is thus not preferred. The short frames and the feature extraction methods could make classification extra difficult. The extracted frames of 2048 correspond to about 0.04 second, which might be too small. Furthermore the feature extraction averages over the whole frame, possibly smoothing interesting features. The placement of the microphones on the car might have influenced the measurements as well. They were placed placed on top of the car, where they might be more susceptible to aerodynamic noise. The hybrid vehicle is not producing a constant vehicle engine noise, because the engine will only be used on occasion. With the relatively small evaluation dataset DriveSound this might be a significant influence.

As the evaluation dataset is captured in medium sized city, the results are not necessarily generalizable to other environments, such as rural areas or city centres with high skyscrapers. Different road surfaces and external noise sources can alter the sounds in both classes. Additionally in areas with a high building density, reverberation might become a problem. It however expected that these effects are manageable. RoadCube for example already contains multiple pavement types.

6.2. Conclusion

Acoustic vehicle detection can be used in ADAS and autonomous vehicles. This thesis shows that is possible to detect vehicles, but not yet with the performance required to base a single modality warning system on. A system were sensor modalities are fused could be a solution.

Two existing datasets were examined, AudioSet a general purpose dataset and RoadCube a specialised small scale vehicle detection dataset. Out of the two existing datasets RoadCube is the most useful, while AudioSet is not suitable for the classification of vehicles in this form. This indicates that specialised smaller dataset is to be preferred above a general purpose dataset. It must be noted however that the features and classifiers were not able to utilise AudioSet fully. This can be seen in the lower validation and test scores in the AudioSet intra experiment (section 5.2.1), while the training scores are high. A possible reason is that

AudioSet contains too much noise or misclassified samples to be classified correctly. The usefulness of the RoadCube diminishes when the listener vehicle is moving. A possible explanation is the lack of noise in the negative class. Classifiers will then be trained on the existence of a sound instead of the type of sound. The only examined dataset capable of detecting vehicles is DriveSound-driving. are suited for the detection of vehicles while moving. For the application this dataset should thus be extended or a new dataset dedicated for vehicle detection with a moving listener should be captured. Another possibility is to investigate a filtering or frequency shift preprocessing step, which make the RoadCube usable for this kind of signal.

Additionally to the datasets also different feature types were used. The test scores in the feature selection indicate that abstract or more complex features perform better than using the raw frequency histogram (see section 5.1). This supports their raison d'être. Furthermore MFCC features were important in any of the datasets Frequency domain based features including the MFCCs were found to be the most important in three out of four datasets. In the other dataset, DriveSound-idle the LPC features were found to be the most important.

Four types of classifiers were evaluated, namely SVM, GMM, MLP and random forest. The MLP performs the best on the intra sets, but it takes relatively long to train. If AudioSet is not included the GMM classifier would perform best overal (see section 5.2.2). The SVM with a RBF kernel performs most robustly. Linear kernels score very low most of the time. The random forest classifiers most of the times perform less well than other types.

The cross-dataset evaluation (section 5.3) shows that AudioSet and RoadCube are not able to classify samples on the DriveSound-driving set. Even DriveSound-idle set does not generalise to the driving case. From the feature selection (section 5.1) it seems that the frequency based features are less viable on the driving set. The are most probably affected by the Doppler effect (section 2.2.1) caused by the motion of the listener vehicle. Furthermore the LPC features could be robust to the driving.

6.3. Future work

The research done in this thesis can be extended in multiple ways. First of all the experiments can be repeated with additional datasets, features and classifier. An interesting dataset to try would be FreeSound, because it can be considered a compromise between the many samples of AudioSet and the specific purpose of RoadCube. The DriveSound could also be used for more specific scenarios, for example a single speed limit. It can be insightful to compare the differences between the scenario's and evaluate them in terms of allowed speed and the distance between microphones and the road. For this purpose the dataset could be extended. Other feature types, like wavelets could be evaluated, but it altering the feature extraction pipeline could give more interesting results. For example hidden markov models, which contain multiple states, could be used to capture not only the average property, but also the change over the signal. A recurrent neural network could be used instead of the multi-frame feature vector. Another interesting approach is to use bag-of-instances clustering to target the components of the vehicle sound. Instead if hand crafted features, a deep learning approach can be taken as well. Features could then be learned by using a deep neural network directly on the sound signal or frequency histogram. These classifiers generally need many more samples, thus it would favour larger datasets. Additionally the problem could be extended to the multi class case, where vehicle types are discerned from each other or the amount of vehicles is predicted. RoadCubeprobably performs well for this type of problem. It will not suffer from the drawback that the classification depends on the power of the sound. It can be assumed that the emissions of all vehicles have a similar power, regardless of class. Then the lower noise present in the dataset can be helpful to clearly distinguish between vehicle classes. This dataset has disadvantages as well for the multiclass system. There are not many different vehicles and a low amount of samples per vehicles limited.

Currently vehicles are detected when they pass the listener vehicle. A next step would be to detect vehicles before this moment. Approaching vehicles could than be detected when there is still time to react. A possible way to investigate this is by using a sliding window, which continuously outputs the confidence that a vehicle is nearby Another way to extend the functionality is by combining the recognition of vehicles with localisation methods like beamforming, which is able to steer the microphone array in a certain direction. This is currently used for robot audition 2.3.3 and has the potential to improve performance of the recognition by reducing noise and add more functionality, like determining the location and speed of the target vehicle. A more advanced localisation method, geometric source separation could also be used to isolate vehicle sound candidates, which can subsequently be classified, but these advanced techniques probably require more microphones. The DriveSound datasets are already suitable for localisation methods, because they were captured with

multiple microphones.

Many factors exist which can affect the performance of the vehicle detection. These properties of acoustic vehicle detection must be further investigated to understand the problem better and potentially improve the solution. The first is to investigate the detection range and its influence on the detection accuracy. RoadCube was detected on a short range, which could be one of the reasons why it is easier, but it is helpful to know at what distance vehicles could be detected. Further research is also needed to determine the acoustic detection range under divers noise conditions. In the datasets only cars were detected with a constant speed. Accelerating and cornering vehicles have a different acoustic signature. Additionally the influence of the relative speed on the vehicle signature could be investigated. Finally a possible improvement could be reached when individual components of a vehicle's sound emission are targeted. This might give also more insight in the commonalities and differences between vehicle types. It would be helpful to investigate the detection of occluded vehicles (sound property: diffraction). If this is feasible it would be a major benefit of acoustic detection of road vehicles. Finally the optimal placement of the microphones on the vehicle can be researched.

I

Appendices

7

Best classifiers per experiment

7.1. (1) - AudioSet intra

Best classifier	MLP
test score (MCC)	.41
activation function	rectified linear unit
α	8.54e-04
hidden layers	1
neurons per layer	142
solver	adam
initial learning rate	.00020

Table 7.1: Test score of the classifier configuration attaining the highest MCC score on the intra AudioSetexperiment with motorized vehicles versus environment division.

7.2. (2) - RoadCube intra

Best classifier	MLP
test score (MCC)	.90
activation function	rectified linear unit
α	5.70e-05
hidden layers	18
neurons per layer	123
solver	L-BFGS
initial learning rate	7.78e-05

Table 7.2: Test score of the classifier configuration attaining the highest MCC score on the intra RoadCube experiment with motorized vehicles versus environment division.

7.3. (3) - DriveSound-idle intra

7.4. (4) - DriveSound-driving intra

Best classifier	GMM
test score (MCC)	.27
components	4
covariance type	tied

Table 7.3: Test score of the classifier configuration attaining the highest MCC score on the intra DriveSound-idle experiment with motorized vehicles versus environment division.

Best classifier	GMM
test score (MCC)	.40
components	2
covariance type	tied

Table 7.4: Test score of the classifier configuration attaining the highest MCC score on the intra DriveSound-driving experiment with motorized vehicles versus environment division.

List of Figures

List of Tables

Acronyms

ADAS Advanced Driver Assistance System 3, 9, 15, 16, 23, 61

AUC Area Under Curve 37

EU European Union 10, 11

FFT Fast-Fourier Transform 28, 30, 52, 69

FPR False Negative Rate 37

GMM Gaussian Mixture Model 3, 23–25, 32, 34, 49, 50, 54, 56–58, 62, 68

L-BFGS Limited memory - Broyden-Fletcher-Goldfarb-Shanno 35, 50, 67

LIDAR LIght Detection And Ranging 18, 19

LPC Linear Predictive Coding 22, 25, 29, 30, 50, 51, 53, 54, 62

MCC Matthews correlation coefficient 37, 38, 49

MFCC Mel Frequency Cepstral Coefficients 23, 25, 29, 31, 42, 50–54, 62, 69

MLP Multi-Layer Perceptron 3, 25, 32, 35, 49, 50, 54, 56–58, 62, 67

PCA Principal Component Analysis 23, 25

PDF Probability Density Function 38

RBF Radial Basis Function 23, 33, 49, 50, 62

ROC curve Receiver Operating Characteristic curve 37, 69

SVM Support Vector Machine 3, 23–25, 32–34, 49, 50, 54, 56–58, 62, 69

TNR True Negative Rate 37

TPR True Positive Rate 37

Symbols

C SVM training parameter 33

$\boldsymbol{x}$ Feature 32

k Cross-validation fold amount 38

y Ground truth 32

μ Mean of the sound signal amplitude 20

s A sound frame in the time domain 29

Bibliography

[1] F. Thibault. (Nov. 2017). Functional audio interfaces through interactive sound: Using auditory displays for automotive safety, guidance and entertainment applications. (Visited on: 21-02-2018), [not peer reviewed], [Online]. Available: https://blog.audiokinetic.com/en/functional-audio-interfaces-through-interactive-sound-using-auditory-displays-for-automotive-safety-guidance-and-entertainment-applications/.

[2] B. Chappel. (Jan. 2018). U.s. autonomous-car startup signs deal with vw and hyundai. (Visited on: 09-03-2018), [not peer reviewed], [Online]. Available: https://www.npr.org/sections/thetwo-way/2018/01/04/575612666/autonomous-car-startup-signs-deal-with-vw-and-hyundai.

[3] Reuters. (May 2017). Peugeot gears up with nutonomy for self-driving car test. (Visited on: 09-03-2018), [not peer reviewed], [Online]. Available: https://www.reuters.com/article/us-peugeot-nutonomy-singapore/peugeot-gears-up-with-nutonomy-for-self-driving-car-test-idUSKBN17Z06H.

[4] A. Hawkins. (Jan. 2018). Toyota's new self-driving car can 'see' up to 200 meters in every direction. (Visited on: 09-03-2018), [not peer reviewed], [Online]. Available: https://www.theverge.com/2018/1/4/16849422/toyota-self-driving-car-platform-3-lexus-lidar.

[5] ——, (Apr. 2017). Apple just received a permit to test self-driving cars in california. (Visited on: 09-03-2018), [not peer reviewed], [Online]. Available: https://www.theverge.com/2017/4/14/15303338/apple-autonomous-vehicle-testing-permit-california.

[6] D. Etherington. (Jan. 2018). Uber ceo hopes to have self-driving cars in service in 18 months. (Visited on: 09-03-2018), [not peer reviewed], [Online]. Available: https://techcrunch.com/2018/01/23/uber-ceo-hopes-to-have-self-driving-cars-in-service-in-18-months/.

[7] J. M. Clanton, D. M. Bevly, A. S. Hodel, and S. Member, "A Low-Cost Solution for an Integrated Multisensor Lane Departure Warning System", *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 1, pp. 47–59, 2009. DOI: 10.1109/TITS.2008.2011690.

[8] C. Wang and H. Nijmeijer, "String Stable Heterogeneous Vehicle Platoon Using Cooperative Adaptive Cruise Control", *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, vol. 2015-Octob, pp. 1977–1982, 2015. DOI: 10.1109/ITSC.2015.320.

[9] D. Ribeiro, A. Mateus, J. C. Nascimento, and P. Miraldo, "A Real-Time Pedestrian Detector using Deep Learning for Human-Aware Navigation", 2016. arXiv: 1607.04441. [Online]. Available: http://arxiv.org/abs/1607.04441.

[10] A. J. Lawrence, N. Avinash, and A. Yogaraj, "FPGA prototyping of vehicle trajectory display for reverse parking - A state of the art survey", *2015 International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials, ICSTM 2015 - Proceedings*, no. May, pp. 389–391, 2015. DOI: 10.1109/ICSTM.2015.7225447.

[11] A. Ziebinski, R. Cupek, H. Erdogan, and S. Waechter, "A Survey of ADAS Technologies for the Future Perspective", in *International Conference on Computational Collective Intelligence*, vol. 2, 2016, pp. 135–146, ISBN: 9783319452463. DOI: 10.1007/978-3-319-45246-3.

[12] K. Bengler, K. Dietmayer, B. Farber, M. Maurer, C. Stiller, and H. Winner, "Three decades of driver assistance systems: Review and future perspectives", *IEEE Intelligent Transportation Systems Magazine*, vol. 6, no. 4, pp. 6–22, 2014, ISSN: 19391390. DOI: 10.1109/MITS.2014.2336271.

[13] P. van Laar, "Acoustic traffic perception", 2018, [not peer reviewed].

[14] N. E. L. Center. (). Regulatory challenges. (Visited on: 12-02-2019), [not peer reviewed], [Online]. Available: https://www.nelconline.org/content/regulatory-challenges.

[15] (). (Visited on: 12-02-2019), [not peer reviewed], [Online]. Available: https://s.iha.com/1426900004126/Short-term-rentals-Rome-Apartment-Flat_4.jpeg.

[16] B. Aaron. (Feb. 2015). The new york city parking rule that makes intersections more dangerous. (Visited on: 12-02-2019), [not peer reviewed], [Online]. Available: https://nyc.streetsblog.org/2015/02/27/the-new-york-city-parking-rule-that-makes-intersections-more-dangerous/.

[17] World Health Organisation. (2015). Global status report on road safety 2015. (Visited on: 03-10-2018), [not peer reviewed], [Online]. Available: http://apps.who.int/iris/bitstream/10665/189242/1/9789241565066_eng.pdf?ua=1.

[18] European Transport Safety Council. (Jun. 2017). Ranking eu progress on road safety. (Visited on: 20-08-2017), [not peer reviewed], [Online]. Available: http://etsc.eu/wp-content/uploads/PIN_ANNUAL_REPORT_2017-final.pdf.

[19] ——, (Jun. 2018). Ranking eu progress on road safety. (Visited on: 02-10-2018), [not peer reviewed], [Online]. Available: https://etsc.eu/wp-content/uploads/PIN_AR_2018_final.pdf.

[20] A. A. M. Aljanahi, A. H. Rhodes, and A. V. Metcalfe, "Speed , speed limits and road traffic accidents under free flow conditions", *Accident Analysis and Prevention*, vol. 31, pp. 161–168, 1999.

[21] World Health Organisation. (2015). Global status report on road safety 2015. (Visited on: 15-03-2018), [not peer reviewed], [Online]. Available: http://www.who.int/violence_injury_prevention/road_safety_status/2015/en/.

[22] CBS Statline. (May 2017). Overledenen; doden door verkeersongeval in nederland, wijze van deelname. Dutch. (Visited on: 20-08-2017), [not peer reviewed], [Online]. Available: http://statline.cbs.nl/Statweb/publication/?DM=SLNL&PA=71936NED.

[23] M. van Infrastructuur en Milieu and Rijkswaterstaat. (Feb. 2018). Verkeersongevallen - bestand geregistreerde ongevallen nederland. (Visited on: 20-02-2018), [not peer reviewed], [Online]. Available: https://data.overheid.nl/data/dataset/verkeersongevallen-bestand-geregistreerde-ongevallen-nederland.

[24] P. Thomas, A. Morris, R. Talbot, and H. Fagerlind, "Identifying the causes of road crashes in Europe.", *Annals of advances in automotive medicine. Association for the Advancement of Automotive Medicine. Scientific Conference*, vol. 57, pp. 13–22, 2013, ISSN: 1943-2461. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/24406942%7B%5C%%7D5Cnhttp://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC3861814.

[25] European Commission. (Jun. 2017). Traffic safety basic facts 2017. (Visited on: 21-02-2018), [not peer reviewed], [Online]. Available: https://ec.europa.eu/transport/road_safety/sites/roadsafety/files/pdf/statistics/dacota/bfs2017_car_occupants.pdf.

[26] G. Underwood, A. Ngai, and J. Underwood, "Driving experience and situation awareness in hazard detection", *Safety Science*, vol. 56, pp. 29–35, 2013, ISSN: 0925-7535. DOI: 10.1016/j.ssci.2012.05.025. [Online]. Available: http://dx.doi.org/10.1016/j.ssci.2012.05.025.

[27] M. S. Horswill, M. Garth, A. Hill, and M. O. Watson, "The effect of performance feedback on drivers ' hazard perception ability and self-ratings", *Accident Analysis and Prevention*, vol. 101, pp. 135–142, 2017, ISSN: 0001-4575. DOI: 10.1016/j.aap.2017.02.009. [Online]. Available: http://dx.doi.org/10.1016/j.aap.2017.02.009.

[28] V. L. Neale, T. A. Dingus, S. G. Klauer, and M. Goodman, "An overview of the 100-car naturalistic study and findings", *Traffic Safety*, pp. 1–10, 2005. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.172.2366%7B%5C&%7Drep=rep1%7B%5C&%7Dtype=pdf.

[29] J.-M. Valin, F. Michaud, and J. Rouat, "Robust localization and tracking of simultaneous moving sound sources using beamforming and particle filtering", 2016. DOI: 10.1109/ICASSP.2006.1661100. arXiv: 1604.01642. [Online]. Available: http://arxiv.org/abs/1604.01642%7B%5C%%7D0Ahttp://dx.doi.org/10.1109/ICASSP.2006.1661100.

[30] N. Shimada, A. Itai, and H. Yasukawa, "A study on using linear microphone array-based acoustic sensing to detect approaching vehicles", *ISCIT 2010 - 2010 10th International Symposium on Communications and Information Technologies*, pp. 182–186, 2010. DOI: 10.1109/ISCIT.2010.5664832.

[31] J.-m. Valin, J. Rouat, and L. Dominic, "Robust Sound Source Localization Using a Microphone Array on a Mobile Robot", *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1228–1233, 2003. DOI: 10.1109/IROS.2003.1248813. arXiv: 1602.08213.

[32] S. Yoshizawa and Y. Nakatoh, "Blind vehicle's sound detecting technique for advanced safety-driving system", in *ICCE - International Conference on Consumer Electronics*, 2009, pp. 3–4, ISBN: 9781424425594.

[33] J. P. Kuhn, B. C. Bui, and G. J. Pieper, *Acoustic Sensor System For Vehicle Detection and Multi-lane Highway Monitoring*, 1998.

[34] S. Argentieri, P. Danès, and P. Souères, "A survey on sound source localization in robotics: From binaural to array processing methods", *Computer Speech and Language*, vol. 34, no. 1, pp. 87–112, 2015, ISSN: 10958363. DOI: 10.1016/j.csl.2015.03.003. [Online]. Available: http://dx.doi.org/10.1016/j.csl.2015.03.003.

[35] R. Peral-Orts, E. Velasco-Sanchez, N. Campillo-Davo, and H. Campello-Vicente, "Using Microphone Arrays to Detect Moving Vehicle Velocity", *Archives of Acoustics*, vol. 38, no. 3, pp. 407–415, 2013, ISSN: 0137-5075. DOI: 10.2478/aoa-2013-0048.

[36] V. Tyagi, S. Kalyanaraman, and R. Krishnapuram, "Vehicular Traffic Density State Estimation Based on Cumulative Road Acoustics", *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 3, pp. 1156–1166, 2012, ISSN: 15249050. DOI: 10.1109/TITS.2012.2190509.

[37] A. Y. Nooralahiyan, H. R. Kirby, and D. McKeown, "Vehicle classification by acoustic signature", *Mathematical and Computer Modelling*, vol. 27, no. 9-11, pp. 205–214, 1998, ISSN: 08957177. DOI: 10.1016/S0895-7177(98)00060-0.

[38] R. Chellappa, G. Q. G. Qian, and Q. Z. Q. Zheng, "Vehicle detection and tracking using acoustic and video sensors", *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, no. 4, pp. 793–796, 2004, ISSN: 1520-6149. DOI: 10.1109/ICASSP.2004.1326664. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1326664.

[39] B. Fazenda, H. Atmoko, F. Gu, L. Guan, and A. Ball, "Acoustic Based Safety Emergency Vehicle Detection for Intelligent Transport Systems", no. 1, pp. 4250–4255, 2009.

[40] Y. Lecun, L. Bottou, Y. Bengio, and H. P, "Gradient-based learning applied to document recognition", *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998, ISSN: 00189219. DOI: 10.1109/5.726791. arXiv: 1102.0183.

[41] (). (Visited on: 14-02-2019), [not peer reviewed], [Online]. Available: https://www.the-blueprints.com/vectordrawings/show/7603/toyota_prius_c/.

[42] D. Gerónimo, A. M. López, A. D. Sappa, and T. Graf, "Survey of pedestrian detection for advanced driver assistance systems", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 7, pp. 1239–1258, 2010, ISSN: 01628828. DOI: 10.1109/TPAMI.2009.122.

[43] N. N. Samani, J. Ghaisari, and M. Danesh, "Autonomous parallel parking of a vehicle in a limited space using a RBF network and a feedback linearization controller", *2012 2nd International eConference on Computer and Knowledge Engineering, ICCKE 2012*, pp. 117–122, 2012. DOI: 10.1109/ICCKE.2012.6395363.

[44] D. Pérez-Morales, S. Domínguez-Quijada, O. Kermorgant, and P. Martinet, "Autonomous parking using a sensor based approach", in *International Conference on Intelligent Transportation Systems*, 2016.

[45] J. Ziegler, P. Bender, M. Schreiber, H. Lategahn, T. Strauss, C. Stiller, T. Dang, U. Franke, N. Appenrodt, C. G. Keller, E. Kaus, R. G. Herrtwich, C. Rabe, D. Pfeiffer, F. Lindner, F. Stein, F. Erbs, M. Enzweiler, C. Knoppel, J. Hipp, M. Haueis, M. Trepte, C. Brenk, A. Tamke, M. Ghanaat, M. Braun, A. Joos, H. Fritz, H. Mock, M. Hein, and E. Zeeb, "Making bertha drive-an autonomous journey on a historic route", *IEEE Intelligent Transportation Systems Magazine*, vol. 6, no. 2, pp. 8–20, 2014, ISSN: 1939-1390. DOI: 10.1109/MITS.2014.2306552.

[46] D. Jia, K. Lu, J. Wang, X. Zhang, and X. Shen, "A survey on platoon-based vehicular cyber-physical systems", *IEEE Communications Surveys and Tutorials*, vol. 18, no. 1, pp. 263–284, 2016, ISSN: 1553877X. DOI: 10.1109/COMST.2015.2410831. arXiv: arXiv:1011.1669v3.

[47] T. B. Sheridan, *Telerobotics, Automation, and Human Supervisory Control*. Cambridge, Massachusetts, USA: MIT Press, 1992, ISBN: 9780262193160.

[48] On-Road Automated Driving Committee (ORAD), "Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles", SAE International, Tech. Rep., 2016, p. 30. DOI: https://doi.org/10.4271/J3016_201609.

[49] D. Storm. (Nov. 2015). Black hat europe: It's easy and costs only $60 to hack self-driving car sensors. (Visited on: 12-02-2019), [not peer reviewed], [Online]. Available: https://www.computerworld.com/article/3005436/cybercrime-hacking/black-hat-europe-it-s-easy-and-costs-only-60-to-hack-self-driving-car-sensors.html.

[50] Y. Cao, A. Mahr, S. Castronovo, M. Theune, C. Stahl, and C. Müller, "Local Danger Warnings for Drivers: The Effect of Modality and Level of Assistance on Driver Reaction", *Proceedings of the International Conference on Intelligent User Interfaces. International Conference on Intelligent User Interfaces (IUI-10), February 7-10, Hong Kong, China*, pp. 239–248, 2010. [Online]. Available: http://www.dfki.de/web/forschung/publikationen/renameFileForDownload?filename=cao%7B%5C_%7Det%7B%5C_%7Dal%7B%5C_%7D2010%7B%5C_%7Diui.pdf%7B%5C&%7Dfile%7B%5C_%7Did=uploads%7B%5C_%7D844.

[51] N. Ohta and K. Niijima, "Detection of Approaching Cars via Artificial Insect Vision", vol. 88, no. 10, pp. 1589–1596, 2005. DOI: 10.1002/ecjc.20195.

[52] H. Zhang, W. Yu, and X. Sun, "Adaptive traffic lane detection based on normalized power accumulation", *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, pp. 968–973, 2008. DOI: 10.1109/ITSC.2008.4732600.

[53] J. Horgan, C. Hughes, J. McDonald, and S. Yogamani, "Vision-Based Driver Assistance Systems: Survey, Taxonomy and Advances", *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, vol. 2015-Octob, pp. 2032–2039, 2015. DOI: 10.1109/ITSC.2015.329.

[54] Y. Na, Y. Guo, Q. Fu, and Y. Yan, "Cross Array and Rank-1 MUSIC Algorithm for Acoustic Highway Lane Detection", *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 9, pp. 2502–2514, 2016, ISSN: 15249050. DOI: 10.1109/TITS.2016.2521661.

[55] B. Malhotra, I. Nikolaidis, and J. Harms, "Distributed classification of acoustic targets in wireless audio-sensor networks", *Computer Networks*, vol. 52, no. 13, pp. 2582–2593, 2008, ISSN: 13891286. DOI: 10.1016/j.comnet.2008.05.008.

[56] 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", no. Iv, pp. 119–123, 2011.

[57] H. Tsuzuki, M. Kugler, S. Kuroyanagi, and A. Iwata, "A Novel Approach for Sound Approaching Detection", pp. 407–414, 2010.

[58] A. Severdaks and M. Liepins, "Vehicle counting and motion direction detection using microphone array", *Elektronika ir Elektrotechnika*, vol. 19, no. 8, pp. 89–92, 2013, ISSN: 13921215. DOI: 10.5755/j01.eee.19.8.5400.

[59] R. E. S. Automotive. (). Advanced driver assistance system (adas). (Visited on: 12-02-2019), [not peer reviewed], [Online]. Available: https://www.nelconline.org/content/regulatory-challenges.

[60] M. Górski and J. Zarzycki, "Feature Extraction in Vehicle Classification", in *International Conference on Signals and Electronic Systems - ICSES*, 2012, ISBN: 9781467317115.

[61] P. Gupta and S. P. Kar, "MUSIC and improved MUSIC algorithm to estimate direction of arrival", in *2015 International Conference on Communication and Signal Processing, ICCSP 2015*, 2015, pp. 757–761, ISBN: 9781479980819. DOI: 10.1109/ICCSP.2015.7322593.

[62] J. R. Jensen, J. K. Nielsen, R. Heusdens, and M. G. Christensen, "DOA estimation of audio sources in reverberant environments", *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. 2016-May, no. 1, pp. 176–180, 2016, ISSN: 15206149. DOI: 10.1109/ICASSP.2016.7471660.

[63] K. Youssef, S. Argentieri, and J. L. Zarader, "A learning-based approach to robust binaural sound localization", *IEEE International Conference on Intelligent Robots and Systems*, pp. 2927–2932, 2013, ISSN: 21530858. DOI: 10.1109/IROS.2013.6696771.

[64] F. Fahy and J. Walker, *Advanced Applications in Acoustics, Noise and Vibration*. London: Spon Press, 2004, ISBN: 0203645138.

[65] H. W. Löllmann, A. H. Moore, P. A. Naylor, B. Rafaely, R. Horaud, A. Mazel, and W. Kellermann, "Microphone array signal processing for robot audition", in *Hands-free Speech Communications and Microphone Arrays - HSCMA*, 2017.

[66] A. Temko and C. Nadeu, "Classification of acoustic events using SVM-based clustering schemes", *Pattern Recognition*, vol. 39, no. 4, pp. 682–694, 2006, ISSN: 00313203. DOI: 10.1016/j.patcog.2005.11.005.

[67] J. Chen, J. Benesty, and Y. Huang, "Time delay estimation in room acoustic environments: An overview", *Eurasip Journal on Applied Signal Processing*, vol. 2006, no. i, pp. 1–19, 2006, ISSN: 11108657. DOI: 10.1155/ASP/2006/26503.

[68] Allenai. (2017). Wave interference. (Visited on: 23-03-2018), [not peer reviewed], [Online]. Available: http://data.allenai.org/tqa/wave_interference_L_1068/.

[69] P. Malupillai. (Jan. 2018). What happens to the phenomenon of the doppler effect when the velocity of the source is greater than the velocity of sound? (Visited on: 23-03-2018), [not peer reviewed], [Online]. Available: https://www.quora.com/What-happens-to-the-phenomenon-of-the-doppler-effect-when-the-velocity-of-the-source-is-greater-than-the-velocity-of-sound.

[70] Wikimedia. (Nov. 2010). File:sound diffraction from a hole.png. (Visited on: 23-03-2018), [not peer reviewed], [Online]. Available: https://commons.wikimedia.org/wiki/File:Sound_Diffraction_from_a_Hole.png.

[71] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing*. London, UK: Prentice-Hall, 1996, ISBN: 0-13-394338-9.

[72] J. Alonso, J. M. López, I. Pavón, C. Asensio, and G. Arcas, "Platform for On-Board Real-Time Detection of Wet , Icy and Snowy Roads, using Tyre/Road Noise Analysis", in *ISCE - International Symposium on Consumer Electronics - IEEE*, 2015, pp. 1–2, ISBN: 9781467373654.

[73] P. Marmaroli, M. Carmona, J. M. Odobez, X. Falourd, and H. Lissek, "Observation of vehicle axles through pass-by noise: A strategy of microphone array design", *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 4, pp. 1654–1664, 2013, ISSN: 15249050. DOI: 10.1109/TITS.2013.2265776.

[74] U. Sandberg, "Tyre/ road noise - Myths and realities", *Inter-noise 2001*, no. 27-30, p. 22, 2001.

[75] Y. Jang, J. Kim, and J. Kim, "The development of the vehicle sound source localization system", *2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, APSIPA ASC 2015*, no. December, pp. 1241–1244, 2015. DOI: 10.1109/APSIPA.2015.7415472.

[76] G. Descornet, "Vehicle noise emission on wet road surfaces", *Internoise 2000. Proceedings of the 29Th International Congress on Noise C*, no. August, pp. 3325–3330, 2000. [Online]. Available: http://ovidsp.ovid.com/ovidweb.cgi?T=JS%7B%5C&%7DPAGE=reference%7B%5C&%7DD=tspt%7B%5C&%7DNEWS=N%7B%5C&%7DAN=E107873.

[77] W. Kongrattanaprasert, H. Nomura, T. Kamakura, and K. Ueda, "Detection of Road Surface States from Tire Noise Using Neural Network Analysis", *IEEJ Transactions on Industry Applications*, vol. 130, no. 7, pp. 920–925, 2010, ISSN: 0913-6339. DOI: 10.1541/ieejias.130.920.

[78] T. Lee. (Nov. 2017). What it's like to live in phoenix? 'waymo units all over the damn place'. (Visited on: 14-03-2018), [not peer reviewed], [Online]. Available: https://arstechnica.com/cars/2017/11/why-phoenix-is-becoming-the-self-driving-capital-of-the-world/.

[79] Z. Yang and L. S. C. Pun-cheng, "Vehicle detection in intelligent transportation systems and its applications under varying environments : A review", *Image and Vision Computing*, vol. 69, pp. 143–154, 2018, ISSN: 0262-8856. DOI: 10.1016/j.imavis.2017.09.008. [Online]. Available: https://doi.org/10.1016/j.imavis.2017.09.008.

[80] (). (Visited on: 12-10-2018), [not peer reviewed], [Online]. Available: https://ama.ab.ca/2017/05/24/tips-for-driving-in-hail.

[81] (). (Visited on: 12-10-2018), [not peer reviewed], [Online]. Available: https://goodstock.photos/bright-light-at-end-of-dark-tunnel/.

[82] (). (Visited on: 12-10-2018), [not peer reviewed], [Online]. Available: http://www.motorcyclephilosophy.org/2014/08/2014-sturgis-sd-day-5-august-5.html.

[83] V. Cevher, R. Chellappa, and J. McClellan, "Vehicle Speed Estimation Using Acoustic Wave Patterns", *IEEE Transactions on Signal Processing*, vol. 57, no. 1, pp. 30–47, 2009, ISSN: 1053-587X. DOI: 10.1109/TSP.2008.2005750. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4625947.

[84] P. Aaron. (Feb. 2015). Making safety systems safer with the right ldo. (Visited on: 12-02-2018), [not peer reviewed], [Online]. Available: https://e2e.ti.com/blogs_/b/behind_the_wheel/archive/2015/02/18/making-safety-systems-safer-with-the-right-ldo.

[85] H. Wu, M. Siegel, and P. Khosla, "Vehicle Sound Signature Recognition by Frequency Vector Principal Component Analysis", pp. 429–434, 1998.

[86] K. Kodera, A. Itai, and H. Yasukawa, "Estimation of speed and arrival time of approaching vehicles using sound", Japanese, *IEICE Technical Report*, pp. 13–18, 2007.

[87] C. T. Ishi, O. Chatot, H. Ishiguro, and N. Hagita, "Evaluation of a MUSIC-based real-time sound localization of multiple sound sources in real noisy environments", *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009*, pp. 2027–2032, 2009. DOI: 10.1109/IROS.2009.5354309.

[88] S. Argentieri, A. Portello, M. Bernard, P. Danès, and B. Gas, "Binaural Systems in Robotics", in *The Technology of Binaural Listening*, J. Blauert, Ed., Springer, Berlin, Heidelberg, 2013, pp. 225–253, ISBN: 9783642377624. DOI: 10.1007/978-3-642-37762-4_9.

[89] (). (Visited on: 14-02-2019), [not peer reviewed], [Online]. Available: https://www.shutterstock.com/image-vector/microphone-vector-icon-black-illustration-isolated-683105479.

[90] S. J. Russell and P. Norvig, *Artificial Intelligence - A Modern Approach*. New York, USA: Prentice Hall, 2010.

[91] K. R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf, "An introduction to kernel-based learning algorithms", *IEEE Transactions on Neural Networks*, vol. 12, no. 2, pp. 181–201, 2001, ISSN: 10459227. DOI: 10.1109/72.914517.

[92] B. Mcfee, C. Raffel, D. Liang, D. P. W. Ellis, M. Mcvicar, E. Battenberg, and O. Nieto, "librosa : Audio and Music Signal Analysis in Python", no. Scipy, pp. 18–24, 2015.

[93] T. Giannakopoulos, "pyAudioAnalysis : An Open-Source Python Library for Audio Signal Analysis", pp. 1–17, 2015. DOI: 10.1371/journal.pone.0144610.

[94] J.-M. Valin, J. Rouat, and F. Michaud, "Enhanced robot audition based on microphone array source separation with post-filter", *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 3, pp. 2123–2128, 2004. DOI: 10.1109/IROS.2004.1389723. arXiv: arXiv:1603.02341v1.

[95] S. Siddagangaiah, Y. Li, X. Guo, X. Chen, Q. Zhang, and K. Yang, "A Complexity-Based Approach for the Detection of", *entropy*, 2016. DOI: 10.3390/e18030101.

[96] N. Dave, "Feature Extraction Methods LPC, PLP and MFCC In Speech Recognition", *International Journal for Advance Research in Engineering and Technology*, vol. 1, no. Vi, pp. 1–5, 2013.

[97] M. B. Trawicki, M. T. Johnson, and T. S. Osiejuk, "Automatic song-type classification and individual identification of the ortolan bunting (Emberiza hortulana L) bird vocalizations", *The Journal of the Acoustical Society of America*, no. October 2015, 2004. DOI: 10.1121/1.4785529.

[98] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay, "Scikit-learn: Machine Learning in Python", *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2012, ISSN: 15324435. DOI: 10.1007/s13398-014-0173-7.2. arXiv: 1201.0490. [Online]. Available: http://dl.acm.org/citation.cfm?id=2078195%7B%5C%%7D5Cnhttp://arxiv.org/abs/1201.0490.

[99] R.-e. Fan, P.-h. Chen, and C.-j. Lin, "Working Set Selection Using Second Order Information for Training Support Vector Machines", vol. 6, pp. 1889–1918, 2005.

[100] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A Library for Large Linear Classification", *Journal of Machine Learning Research*, vol. 9, no. 2008, pp. 1871–1874, 2008, ISSN: 15324435. DOI: 10.1038/oby.2011.351.

[101] C.-c. Chang and C.-j. Lin, "LIBSVM : A Library for Support Vector Machines", *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, pp. 1–39, 2013, ISSN: 21576904. DOI: 10.1145/1961189.1961199. arXiv: 0-387-31073-8.

[102] A. Abdiansah and R. Wardoyo, "Time Complexity Analysis of Support Vector Machines (SVM) in LibSVM", *International Journal of Computer Applications*, vol. 128, no. 3, pp. 975–8887, 2015, ISSN: 09758887. DOI: 10.5120/ijca2015906480. [Online]. Available: https://pdfs.semanticscholar.org/a8b4/786c9128d4a94caeb67c858ab4f4288c49ff.pdf.

[103] L. Bottou and C.-J. Lin, "Support Vector Machine Solvers", *Large Scale Kernel Machines*, pp. 301–320, 2007. DOI: 10.1.1.127.511. [Online]. Available: http://leon.bottou.org/papers/bottou-lin-2006.

[104] G. Cauwenberghs and T. Poggio, "Incremental and Decremental Support Vector Machine Learning", *Learning*, vol. 13, no. 13, p. 409, 2001, ISSN: 10495258. DOI: 10.1.1.21.1720. [Online]. Available: http://books.google.com/books?hl=en%7B%5C&%7Dlr=%7B%5C&%7Did=Mgs2FwtgNxwC%7B%5C&%7Doi=fnd%7B%5C&%7Dpg=PA409%7B%5C&%7Ddq=Incremental+and+Decremental+Support+Vector+Machine+Learning%7B%5C&%7Dots=EJWO4nfu9A%7B%5C&%7Dsig=iQnXtiOz9MNVDtR7g-V0bcXrBTs.

[105] J. C. Platt, *Probabilistic Outputs for Support Vector Mahines and Comparisons to RegularizedLikelihoodMethods.pdf*, 1999.

[106] C. W. Hsu and C. J. Lin, "A comparison of methods for multiclass support vector machines", *IEEE Transactions on Neural Networks*, vol. 13, no. 2, pp. 415–425, 2002, ISSN: 10459227. DOI: 10.1109/72.991427.

[107] E. Geller. (Oct. 2012). Bayes' rule and bomb threats. (Visited on: 14-02-2019), [not peer reviewed], [Online]. Available: https://www.psychologyinaction.org/psychology-in-action-1/2012/10/22/bayes-rule-and-bomb-threats.

[108] R. Shindjalova, K. Prodanova, and V. Svechtarov, "Modeling data for tilted implants in grafted with bio-oss maxillary sinuses using logistic regression", *AIP Conference Proceedings*, vol. 1631, pp. 58–62, 2014, ISSN: 15517616. DOI: 10.1063/1.4902458. arXiv: 1412.6980v9.

[109] A. Karpathy. (2017). Cs231n convolutional neural networks for visual recognition. (Visited on: 21-03-2018), [not peer reviewed], [Online]. Available: https://cs231n.github.io/neural-networks-1/.

[110] L. Breiman, "Random Forests", pp. 1–33, 2001, ISSN: 1098-6596. DOI: 10.1017/CBO9781107415324.004. arXiv: arXiv:1011.1669v3.

[111] A. Averbuch, V. A. Zheludev, N. Rabin, and A. Schclar, "Wavelet-based acoustic detection of moving vehicles", *Multidimensional Systems and Signal Processing*, vol. 20, no. 1, pp. 55–80, 2009, ISSN: 09236082. DOI: 10.1007/s11045-008-0058-z.

[112] D. Powers, "Evaluation: from precision, recall and f-measure to ROC, informedness, markedness & correlation", *Journal of Machine Learning Technologies*, vol. 2, no. 1, pp. 37–63, 2011, ISSN: 2229-3981. DOI: 10.1.1.214.9232. arXiv: arXiv:1011.1669v3.

[113] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks", *Information Processing and Management*, vol. 45, no. 4, pp. 427–437, 2009, ISSN: 03064573. DOI: 10.1016/j.ipm.2009.03.002. [Online]. Available: http://dx.doi.org/10.1016/j.ipm.2009.03.002.

[114] G. Jurman, S. Riccadonna, and C. Furlanello, "A comparison of MCC and CEN error measures in multi-class prediction", *PLoS ONE*, vol. 7, no. 8, pp. 1–8, 2012, ISSN: 19326203. DOI: 10.1371/journal.pone.0041882. arXiv: 1008.2908.

[115] M. Sandri and P. Zuccolotto, "Variable Selection Using Random Forests", 1998.

[116] J. Bergstra and B. Yoshua(, "Random Search for Hyper-Parameter Optimization", *Journal of Machine Learning Research*, vol. 13, pp. 281–305, 2012, ISSN: 1532-4435. DOI: 10.1162/153244303322533223. arXiv: 1504.05070.

[117] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, C. Moore, M. Plakal, and M. Ritter, "Audio Set: An ontology and human-labeled dataset for audio events", *Icassp*, pp. 1–5, 2017. [Online]. Available: https://research.google.com/pubs/archive/45857.pdf.

[118] A. S. V. D. Zwaag, J. O. Y. Huisman, T. M. M. P. V. Engelshoven, and Y. R. J. M. V. Engelshoven, "Road Surface and Vehicle Classification", 2018.

[119] E. Fonseca, J. Pons, X. Favory, F. Font, D. Bogdanov, A. Ferraro, S. Oramas, A. Porter, and X. Serra, "Freesound Datasets: a Platform for the Creation of Open Audio Datasets", *ISMIR, International Society for Music Information Retrieval Conference*, pp. 486–493, 2017.

[120] C. J. Salamon and J. P. Bello, "Urban Sound Datasets", in *22nd ACM International Conference on Multimedia*, 2014, ISBN: 9781450330633. DOI: 10.1145/2647868.2655045. [Online]. Available: http://serv.cusp.nyu.edu/projects/urbansounddataset/.

[121] G. Chandrashekar and F. Sahin, "A survey on feature selection methods", *Computers and Electrical Engineering*, vol. 40, no. 1, pp. 16–28, 2014, ISSN: 00457906. DOI: 10.1016/j.compeleceng.2013.11.024. arXiv: 1606.03476. [Online]. Available: http://dx.doi.org/10.1016/j.compeleceng.2013.11.024.