Car, Bicycle and Pedestrian Detection in Adverse Weather Conditions using Lidar and Thermal Imaging

Project Report

844

Aalborg University
Electronics and IT

Electronics and IT
Aalborg University
http://www.aau.dk

AALBORG UNIVERSITY
STUDENT REPORT

Title:
Car, Bicycle and Pedestrian Detection in Adverse Weather Conditions using Lidar and Thermal Imaging

Theme:
Computer Vision

Project Period:
Spring Semester 2021

Project Group:
844

Participant(s):
Bjarne Johannsen

Supervisor(s):
Rikke Gade

Copies: 1

Page Numbers: 65

Date of Completion:
May 26, 2021

Abstract:

In the following work, a novel multimodal sensor data fusion on far infrared imaging and lidar is elaborated and its effects on avderse weather conditions are studied. Furthermore, a well-known problem in the thermal imaging domain is addressed, the existing large domain gap to existing datasets for the pre-training of deep neural networks. With the help of artificial thermal images the training of underrepresented object classes can be improved. The final analysis takes into account light fog, dense fog and snow during day and night. This information is finally evaluated, characterized and summarized with an outlook showing that infrared and far infrared wavelengths have strong challenges with atmospheric humidity. Both lidar and thermal imaging show degraded performance in these conditions.

agreement with the author.

Contents

List of Figures **vii**

Preface **1**

1 Introduction **3**
1.1 Motivation 3
1.2 Structure of the Thesis 4

2 Fundamentals **5**
2.1 Sensors for Vehicle Environmental Perception 5
2.2 Convolutional Neural Networks 11
2.3 Sensor Data Fusion 14

3 Requirements and Design **21**
3.1 Sensor Data Fusion Architecture 21
3.2 Object Detection 26
3.3 Pre Training Deep Neural Networks for Thermal Imaging 29

4 Transfer Learning for Thermal Imaging **31**
4.1 Generating Artificial Thermal Images 33
4.2 Evaluating MSCOCO Performance 37
4.3 Evaluating MSCOCO-thermal Performance 39

5 Evaluating Thermal and Lidar Imaging in Adverse Weather Conditions **41**
5.1 Dense Fog Daytime 43
5.2 Dense Fog Nighttime 48
5.3 Light Fog Daytime 51
5.4 Light Fog Night 53
5.5 Snow Day 55
5.6 Snow Night 57

6 Conclusion and Outlook **61**
6.1 Conclusion 61
6.2 Outlook 62

Bibliography **63**

List of Figures

2.1 Thermal Image and RGB Image from the Same Scene at Night. [4] . 6
2.2 Flir ADK - Automotive Thermal Camera. [9] 6
2.3 (a) Scanner frame axes. (b) Scanner layout. (c) Scanner Parameters in Vertical Plane. (d) Scanner Parameters in Horizontal Plane.[13] . . 8
2.4 Multi-echo measurement a lidar sensor[14] 8
2.5 Visualisation of a high resolution Lidar point cloud in foggy weather. 10
2.6 Principle of an Artificial Neuron. 11
2.7 Process chain of environmental perception with subsequent feature extraction 16
2.8 Low-Level-Fusion 17
2.9 Feature-Level-Fusion 18
2.10 High-Level Fusion 19

3.1 Sensor Data Fusion Architectures in Adverse Weather Conditions. [23] 21
3.2 Low Level Sensor Data Fusion of Lidar and Camera. [23] 22
3.3 High Level Sensor Data Fusion of Lidar and Camera. [23] 22
3.4 Weighted upsampling of the lidar point cloud. [17] 23
3.5 Weighted Depth Filling Algorithm. [17] 25
3.6 Complementary Weighted Depth Filling Algorithm. [17] 25
3.7 One-Stage Detector vs. Two-Stage Detector. [5] 26
3.8 Architecture of YOLO.[17] 27
3.9 MSCOCO Object Deetection Benchmark Evaluation.[17] 28

4.1 Example Image from the Dataset with Detected Objects. [10] 32
4.2 Example Image from the Dataset with the Corresponding Temperature. 33

4.3 Example Image from the Dataset with the Corresponding Semantic Segmentation. 33
4.4 Absolute Temperature Distribution of each class in the Thermal-World dataset. 34
4.5 Panoptical Labels and Corresponding Images from the MSCOCO Dataset. [18] 35
4.6 RGB Image, Semantic Segmenation, Instance Segmentation and Panoptic Segmentation of the same Scene. [18] 35
4.7 Absolute Background Temperature Distribution and Example Values. 36
4.8 Relative Temperature Distribution and Example Values of a Person. 36
4.9 MSCOCO Images and Corresponding Generated Thermal Images . 37
4.10 Average Precision of MSCOCO + Flir. 38
4.11 Precision vs. Recall of MSCOCO + Flir. 39
4.12 Average Precision of MSCOCO thermal + Flir. 40
4.13 Precision vs. Recall of MSCOCO thermal + Flir. 40

5.1 Geographical Diversity of the SeeingThroughFog Dataset. [4] 41
5.2 Sensor Setup on the Project Car. [4] 42
5.3 Image ID: 200, Thermal Imaging fails in dense fog. 43
5.4 Image ID: 200, Also lidar fails ins combination with thermal imaging in dense fog. 44
5.5 Image ID: 2190, Thermal Imaging fails in dense fog and can only correctly see close persons. 44
5.6 Image ID: 200, RGB shows a clear image of the envivroment compared to thermal imaging. 47
5.7 Image ID: 2190, RGB shows a clear image of the envivroment compared to thermal imaging. 47
5.8 Image ID: 2190, Lidar captures mostly noise in these conditions. . . 48
5.9 Image ID: 900, Lidar captures mostly noise in these conditions. . . . 50

Preface

Aalborg University, May 26, 2021

Bjarne Johannsen
bjohan20@student.aau.dk

Chapter 1

Introduction

1.1 Motivation

A major challenge in the development of autonomous vehicles is a highly accurate perception pipeline during all weather conditions. This is necessary to reliably implement fully autonomous vehicles. It is of great importance to gain extensive knowledge about the dynamic environment and other road users. Moving objects such as pedestrians, cyclists, cars are called dynamic. This information is essential to navigate safely in public traffic. Autonomous vehicles are equipped with a sensor set. The capturing and processing of the environment by high resolution sensors is called environmental perception. Such a sensor set usually contains lidars and cameras. Recently, thermal imaging was also incorporated by several studies. Each of these sensors has advantages and disadvantages, which should be combined profitably in the sensor fusion.

Autonomous vehicles have largely been developed and tested in warm sunny regions where visibility is usually near perfect. Adverse weather conditions such as those that occur in northern latitudes pose a major challenge to classic environmental perception systems. Therefore, in the present work an approach to incorporate lidar and thermal imaging into a fused perception is being developed and evaluated.

As part of the development of this multi-modal object detection, a forward-looking sensor configuration consisting of a automotive reference lidar and a high-resolution thermal camera was used. The novel fused perception pipeline has the goal to overcome adverse weather conditions and resolve bad visibility by incorporating daylight invariant optical measurements principles for autonomous driving.

1.2 Structure of the Thesis

This work is divided into five central chapters. First of all, the following chapter gives an explanation of the essential basics of the sensors used in this work. There, the underlying technology for capturing the environment is explained. On the one hand the lidar, on the other hand the thermal imaging. Subsequently, the sensor data fusion will be described and the processing chain from measurement to perception will be elaborated in more detail. The following section focuses on fusion architectures, highlighting high-level, feature-level, and low-level fusion systems and basic understanding of neural networks which will be used for feature extraction. The third chapter is used to define requirements and present the best possible solution. Chapter four focuses on the implementation and evaluation of transfer learning to overcome the domain gap in the thermal domain. The fifth chapter focuses on evaluation different weather conditions and showing the performance of thermal imaging in adverse weather. Chapter six summarizes the knowledge gained and gives an outlook on future expansions.

Chapter 2

Fundamentals

2.1 Sensors for Vehicle Environmental Perception

Thermal Imaging

Visual cameras that capture visible light have been the standard imaging device in most applications such including driver assistance systems. But there are some drawbacks to use these cameras. The colors and visibility of objects depend on an energy source such as the sun or artificial light. Therefore, the main challenge is that the images depend on the illumination, changing intensity, color balance, direction. In addition, nothing can be captured in complete darkness. In the mid- and long-wave infrared spectrum (3-14 μm), radiation is emitted by the objects themselves, with wavelength and intensity depending on temperature. They do not depend on an external energy source. Thermal imaging cameras take advantage of this property and measure radiation in parts of this spectrum. [11] Figure 2.1 shows the same scene at night captured with a far infrared camera and a standard rgb automotive camera.

(a) (b)

Figure 2.1: Thermal Image and RGB Image from the Same Scene at Night. [4]

For the lenses of thermal imaging cameras, a non-glass material has to be utilized to account for the very low transmissivity of glass to thermal radiation. Germanium is the most popular of these materials. Germanium is a metalloid material, grayish-white in color, that is nearly completely transmissive to the infrared spectrum and reflective to the visible spectral range. Germanium has a comparatively high price, in consequence of which the dimensions of the lens are very significant importance [11] and limiting the use of wide angle lenses with only up to 75 ° in automotive applications. [9]

Figure 2.2: Flir ADK - Automotive Thermal Camera. [9]

Lidar - Light Detection and Ranging

Lidar is a sensor based on an optical measurement method. The distance can be determined by means of light pulses emitted by the sensor itself. Compared to the Radar, the Lidar uses electromagnetic radiation in the infrared spectrum.

In the sensors used in the vehicle a measurement according to the "Time of Flight" principle is usually applied. The duration from the emission of the light pulse to the reception of the backscattered beams is proportional to the radial distance between the sensor and the detected object. Since the distance from measuring system to the object has to be completed both on the way there and on the way back, time t and the speed of light c_0 have to be divided by two as shown in equation 2.1.

$$d = \frac{c_0 \cdot t}{2} \tag{2.1}$$

The propagation properties of the wavelength with which the distance is measured in the Lidar sensor results in a comparable detection of object structures such as a camera or the human eye. Only non-hidden objects in the direct field of view can be detected. The big advantage of the short freuency at about 350THz is the possibility to detect small objects as long as a laser beam hits the object. The short wavelength enables a very high resolution, which is usually only mechanically limited by the scanning method.[26]

Figure 2.3 shows the mechanical construction of a lidar sensor. It contains 64 separate lasers, which cover a vertical field of view from 2°to -24.8°. This measurement setup rotates several times per second around the z-axis and thus allows the surrounding area to be captured. In this case, the horizontal resolution is determined by the sampling rate and the rotational speed. The vertical resolution of the the field of view divided by the number of lasers.

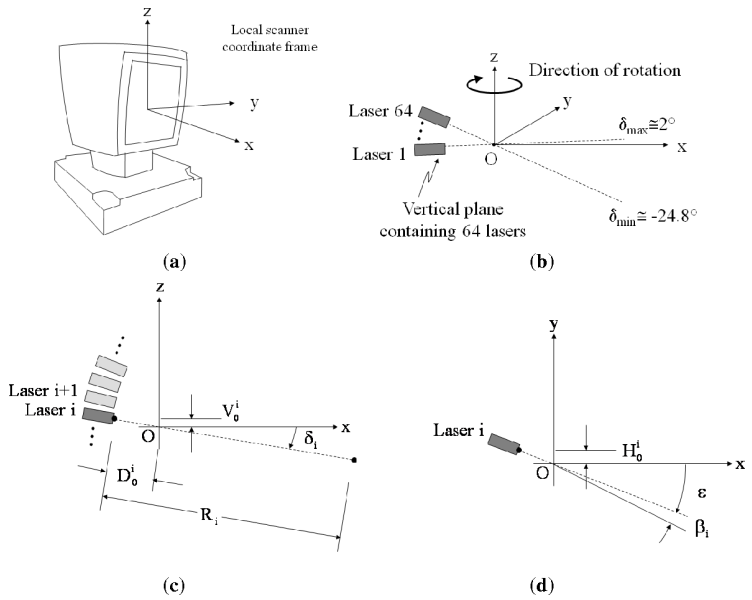


Figure 2.3: (a) Scanner frame axes. (b) Scanner layout. (c) Scanner Parameters in Vertical Plane. (d) Scanner Parameters in Horizontal Plane.[13]

Often there are several objects in one measuring channel, but with a sufficiently large distance, several objects can be detected by appropriate evaluation procedures. In this case we speak of a multi-echo measurement, which is shown in figure 2.4. This enables the multi-target capability of the sensor. At a high attenuation, e.g. due to fog, water droplets and dust, corresponding pulses are reflected at these, as shown in figure 2.4. The actual object (car) can still be detected correctly.

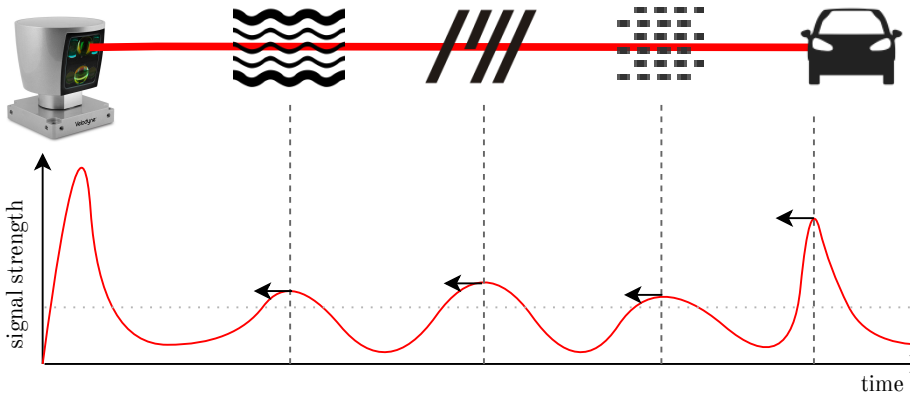


Figure 2.4: Multi-echo measurement a lidar sensor[14]

The proportion of transmitted radiation is referred to as the transmittance. The attenuation is generally composed of absorption, scattering, diffraction and reflection and depends on the wavelength. A major challenge in laser measurement technology is to detect the energy after reflection from an object, which is very limited due to the eye safety requirements. It should be noted that the object, similar to a Lambert reflector[1], usually radiates its energy diffusely into half the solid angle (180°).[26]

[1]The Lambertian law (also Lambertian cosine law) describes, formulated by Johann Heinrich Lambert, how the radiation strength decreases with a flattening beam angle due to the perspective effect. If a surface follows Lambert's law and the radiance of the surface is constant, the result is a circular distribution of the radiant intensity.

Figure 2.5 shows the resulting point cloud of a high-resolution rotation lidar. In this case, the sensor is mounted on the roof of the measuring vehicle, rotates mechanically and can thus completely detect the environment horizontally. The intensity of the reflection is represented by the color of the points.

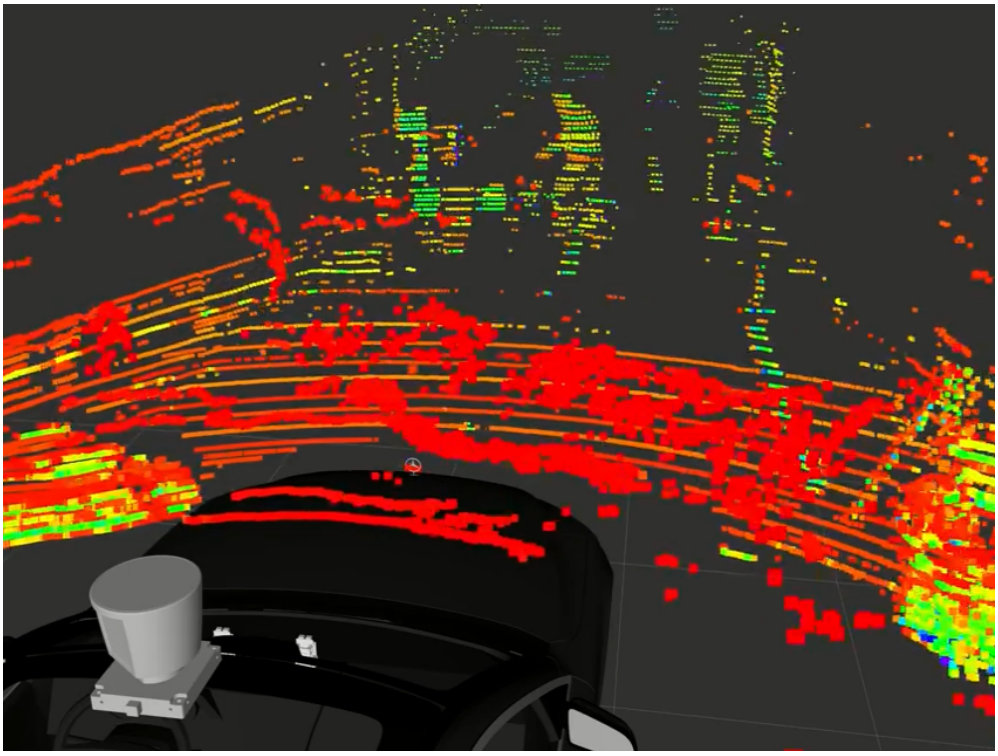


Figure 2.5: Visualisation of a high resolution Lidar point cloud in foggy weather.

2.2 Convolutional Neural Networks

Principles

Deep learning essentially refers to the use of neural networks with many hidden layers. In theory, a single hidden layer is sufficient to represent any function to any degree of accuracy, but may however require an infinite number of neurons. Using multiple hidden layers can be motivated by comparing neural networks to logic circuits. The number of units needed to represent some functions decreases as the depth of the circuit increases. More complex functions can therefore be represented by fewer neurons when the depth of the network is increased.

The artificial neuron

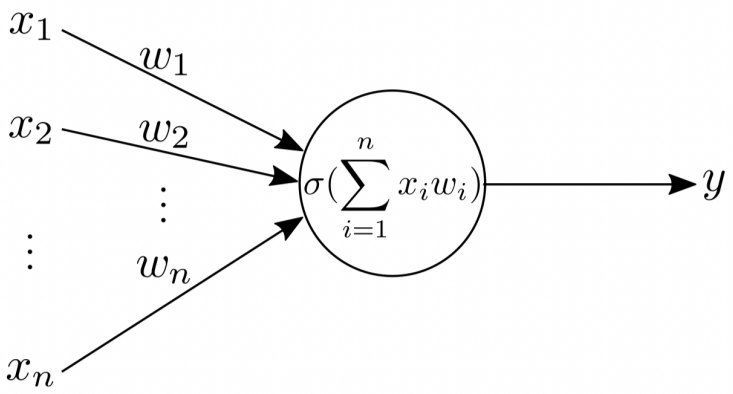


Figure 2.6: Principle of an Artificial Neuron.

The basic computational building block of a neural network is the artificial neuron, depicted in Figure 2.6. It takes a fixed number of scalar input values, represented as the elements of a vector x and outputs a single scalar value y. Each input position of the neuron is associated with a scalar parameter called a weight, represented here as the elements of the vector w. It is these weights that constitute the parameters of neuron, and are what is adjusted when the neuron is learning. The computation performed by the neuron consists of two steps. The first step computes a linear combination of the input values and their corresponding weights, which is equivalent to the dot product of these two vectors. The second step is performed by feeding the result of the first step through an activation function. The output

from the second step is then declared the final output of the neuron. Commonly used activation functions include the sigmoid shaped logistic and hyperbolic tangent functions. For deep neural networks, the ReLU ($y = max(x, 0)$) activation function is a popular choice. The activation function introduces a non-linearity to the output of the neuron, making it capable of non-linear approximations.

Layers

Several neurons can be used together when more than one output dimension is needed, such as in multi-class classification. Each neuron's output then corresponds to a single dimension in the output space. Which can be represented by a layer consisting of as many neurons as the desired output dimension. And each neuron is given an input vector and carries out independent computations to generate its output. With a fully connected layer, i.e., if each neuron in the layer receives inputs from all neurons in the previous layer, the computation performed can be described by a vector-matrix multiplication followed by an element-wise application of the activation function.

Convolutional Neural Networks

Convolutional neural networks represent by far the most popular type of neural network when it comes to image processing. While the underlying mechanisms are the same as for normal feed-forward neural networks, the convolutional network utilizes two different types of layers with several special properties that have made them good at the task of image processing. Namely, the convolutional layer. Within a convolutional layer, where each neuron is only given a small local window of input on which to carry out its calculations. The spatial window that is visible to a given neuron is called its receptive field, and the convolutional layer significantly reduces the receptive field of neurons in comparison to a fully connected layer. In a convolutional layer, the receptive fields of all neurons are tiling the input vector in overlapping windows. This kind of constraint is motivated by the realization that in a signal with spatial correlation, like an image, nearby locations are more likely to be correlated than locations farther apart. However, in the example of an image where the position of an entity within the image is random, the above implementation of the convolutional layer means that each neuron in the layer would have to be trained to recognize the same object. Doing so allows for an optimization that significantly reduces the number of parameters in the convo-

lutional layer. While each neuron in the same layer can share its weights with each other. In other words, the computation performed by a convolutional layer can be efficiently implemented as a convolutional operation, which convolves a filter of values representing the shared weights of the neurons in the layer with the input image. This convolution layer effectively targets a highly important issue in image processing, namely, translational invariance. That it is applied to all possible locations in the image means that the filter is able to identify patterns in the image independent of their location. Practically, this means that a convolutional layer carries out a convolution of the input with filters learned during training. However, a single convolutional layer typically contains many filters, each of which produces its own output. By doing this, it allows the layer to learn to extract several different types of features from the input. When applied to an image, the output of a convolutional layer is known as a feature map. Basically, this is a multi-channel image where each channel corresponds to the output of each filter.

Deep Neural Networks

A major difficulty in performing image analysis is the need to create features that extract unique features from the image that can be reliably used by a classifier. An extremely timeintensive task, and frequently the one that makes or breaks a good system. Nevertheless, neural networks with many hidden layers have shown that they can solve this problem by being able to automatically extract such features when adequate training data is available. Traditionally, these types of deep networks have been known as difficult to train because the gradient wanished as it propagates through many layers and eventually disappears due to the many parameters of the model. However, recent advances in hardware as well as architectural inventions have made it possible to train these deep networks. A technique that has demonstrated that this problem can be addressed is the rectified linear unit (ReLU) activation function, defined as $ReLU(x) = max(0, x)$. In contrast to sigmoidal activation functions, ReLU enables the gradient to propagate without being vanished by the activation function. Deep neural networks have demonstrated several highly interesting properties in the way information is extracted from the input. Within a trained network, the initial layers learn to detect very general features, which are assembled into higher level feature detectors by subsequent layers. Convolutional networks are applied to image data are particularly likely, as this information is easy to detect.

2.3 Sensor Data Fusion

Goals of sensor data fusion

The basic objective of sensor data fusion is to bring together different independent sensors in such a way that the advantages of different measuring principles and methods compensate each other's weaknesses.

Redundant Information

If several sensors provide information about the same object, this is referred to as redundancy. This additional information can be used to improve the known object properties. Different physical measuring principles have different occurring errors. This behaviour has to be considered by using respective measurement models. Redundant environmental sensors can lead to an increase in fault tolerance and increased system availability. If individual sensors fail, data of sufficient quality can still be made available.[26]

Complementarity

Environmental sensors, which provide different but additional information to the fusion, are called complementary. On the one hand, this can be done by different visual ranges of similar sensors, which improve the detection. This way, problems with the association at the edge of the sensor detection range can be solved[7]. If the data refer to the same object, the gathered information can be increased by detecting different properties. The use of different sensor measurement methods can lead to an increase in the robustness of the overall system, as different characteristics of the objects can be detected. For example, the Lidar is able to detect outer contours in the direct field of view, whereas the Lidar enables the detection of features beyond the contour edges.[26]

Process of Sensor Data Fusion

The sensor data fusion is divided into main components, which are explained below. These are generally valid for single sensor as well as multi sensor systems.

Signal Processing and Feature Extraction

In the signal processing step with subsequent Feature extraction, environmental information is captured from the sensors. In the first step of the measurement, the signals are recorded by the sensor, which also contains noise. The raw signals recorded this way are converted into physical quantities, which form the raw data of a sensor. The further steps of signal processing includes the interpretation under consideration of the physical properties. Afterwards, characteristics are extracted from the raw data. Feature hypotheses can then be derived to object hypotheses. However, this entails the risk of misinterpretations. Basically, the data of all sensors must be transformed into a common coordinate system. Possible errors in the alignment can lead to errors in the assignment. Different measurement methods of the sensors lead to different feature hypotheses. For example, the outer edge of a vehicle is detected by laser sensors, but only the hot wheels are detected by thermal camera. This can also occur with sensors of the same type, since different viewing angles provide different measured values. In a multisensor system, different sensors ideally map the same object. Due to unequal resolution and misinterpretations, the characteristics can differ from each other.[26] Figure 2.7 shows the described procedure.

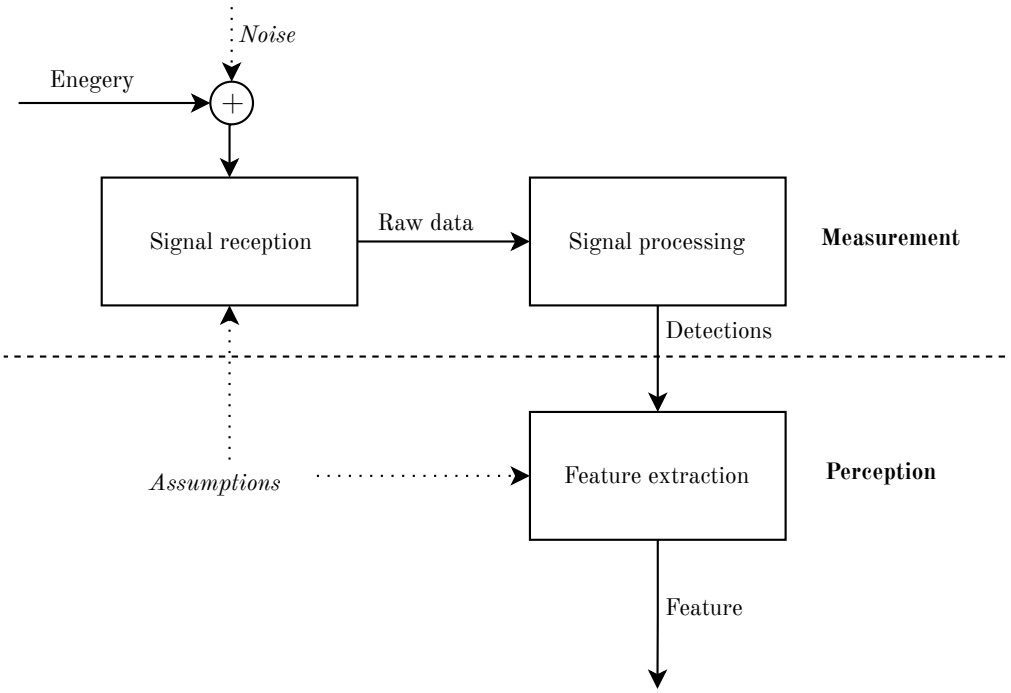


Figure 2.7: Process chain of environmental perception with subsequent feature extraction

Classification

In the classification step, the object hypotheses are assigned previously defined object classes based on given properties of the data. These classifications are used to distinguish between cars and cyclists and pedestrians, for example.

Analysis of the situation

When using the fused data to take over driving functions, an analysis of the situation becomes necessary. This is the link between the merged environment acquisition and the assistance functions. The analysis must then provide the basis for intervention decisions, taking into account the performance of sensors and fusion.[26]

Multi-sensor Data Fusion Architectures

Sensor fusion is generally divided into three different architectures. They are also referred to as central, decentral and hybrid fusion. These are based on the degree of processing of the sensor data and the level at which they are merged.

In a low-level fusion architecture, there is no pre-processing of raw data at the sensor level. Each sensor transmits its raw data to the fusion module, which then performs a low-level fusion of the raw data from all sensors. The merged raw data is then provided as input for a central tracking algorithm. This architecture is sometimes referred to as a centralized tracking architecture. A basic low-level fusion architecture is shown in Figure 2.8. The advantage of a low-level fusion is that all relevant data still exists and is taken into account in the further processing. Abstraction levels such as features result usually in a loss of information. Also it is possible to classify the data very early by merging raw data from different sources. The merged measurements selected for input into the tracking algorithm already have a high probability of being a valid measurement of a relevant object for a particular application. However, low-level fusion requires a large database and can be complex to implement in practice. Adding a new sensor to the architecture requires significant changes to the fusion module because the raw data comes in different formats and from different sensor types.

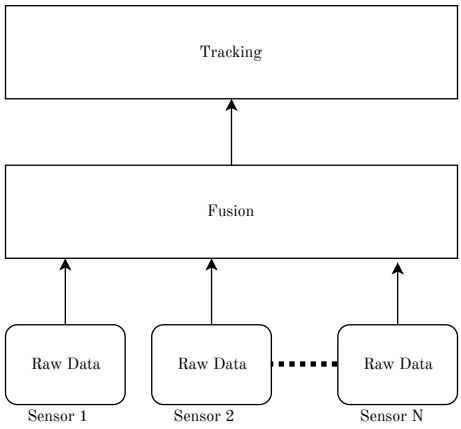


Figure 2.8: Low-Level-Fusion

In the case of fusion at feature level, certain characteristics are extracted from

the raw data by a preprocessing step before the data fusion is performed. The extracted characteristics from a particular object model are then used as input for the tracking algorithm. The architecture of the fusion is shown in Figure 2.9. The main advantage of feature-level fusion is its ability to reduce the bandwidth of sensor data for the fusion. This is possible because the extracted feature data is much smaller than the raw data of the sensor. However, feature-level fusion retains the ability to preprocess and classify sensor data such as a low-level fusion. This reduction in bandwidth results in a significantly reduced runtime of the entire fusion. In addition, parallel processing of the raw data is possible, which further decreases the runtime of the overall system.

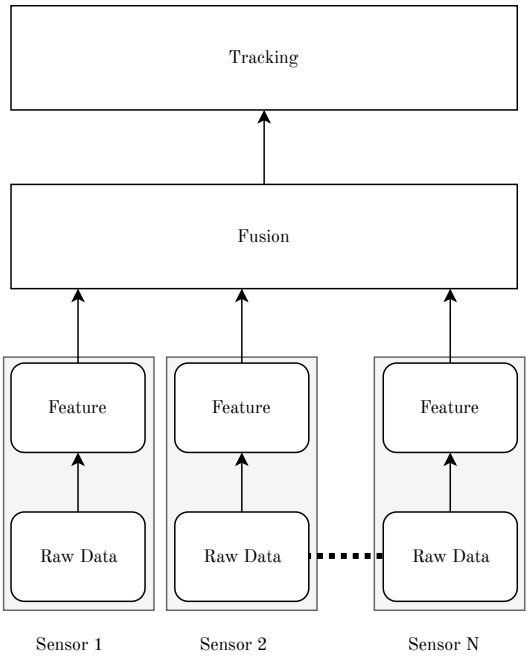


Figure 2.9: Feature-Level-Fusion

High-level fusion architecture is the opposite of low-level fusion. Each sensor independently performs the steps from the raw data to the tracking algorithm and generates an object list. In the fusion module, these objects are then linked to one of the following objects and lead to a track-to-track fusion of sensor-independent objects. A basic fusion at this level is shown in Figure 2.10. The main advantage of high-level fusion is the modularity and encapsulation of sensor-specific details.

All sensor relevant details are kept on the sensor level so that the fusion module can process the data abstractly. This makes the high-level fusion architecture advantageous for applications in which modularity and simplicity of design are of primary importance.[1]

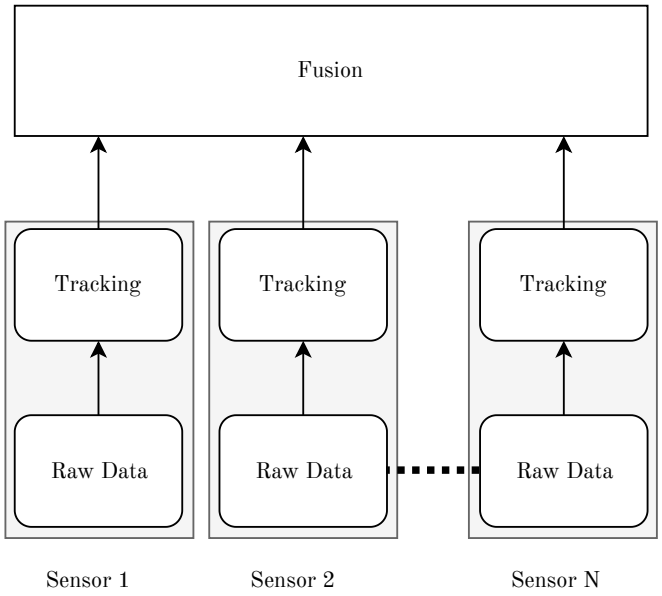


Figure 2.10: High-Level Fusion

Chapter 3

Requirements and Design

The development of Car, Bicycle and Pedestrian Detection in Adverse Weather Conditions by using Lidar and Thermal Imaging is subject to some requirements which will be presented in the next section. Thermal Imaging and Lidar have different raw data, Lidar provides a high resolution 3D point cloud. Since this work focuses on sensor domain, the complexity is reduced to a 2D projection.

3.1 Sensor Data Fusion Architecture

Sensor Data Fusion in adverse weather conditions is well studied in [23].

EVALUATION USING A NET TRAINED WITH ADVERSE-KITTI

approach			**classes**						
feature encoder	fusion approach	dataset	car	truck	tram	pedestrian	cyclist	van	**mAP**
VGG16m	Early Fusion	Kitti	0.732	0.784	0.588	0.390	0.535	0.621	**0.609**
		adverse-Kitti	0.689	0.657	0.433	0.343	0.413	0.522	**0.510**
	Middle Fusion	Kitti	0.738	0.775	0.606	0.414	0.541	0.628	**0.617**
		adverse-Kitti	0.697	0.667	0.474	0.363	0.431	0.539	**0.529**
	Late Fusion	Kitti	0.736	0.765	0.681	0.425	0.519	0.644	**0.628**
		adverse-Kitti	0.698	0.663	0.488	0.372	0.408	0.547	**0.529**
VGG16	Early Fusion	Kitti	0.795	0.899	0.888	0.538	0.752	0.806	**0.780**
		adverse-Kitti	0.766	0.826	0.745	0.498	0.679	0.719	**0.707**
	Middle Fusion	Kitti	0.801	0.922	0.915	0.522	0.794	0.817	**0.795**
		adverse-Kitti	0.772	0.856	0.778	0.495	0.705	0.732	**0.723**
	Late Fusion	Kitti	0.798	0.907	0.962	0.556	0.758	0.839	**0.805**
		adverse-Kitti	0.766	0.851	0.801	0.506	0.690	0.758	**0.729**

Figure 3.1: Sensor Data Fusion Architectures in Adverse Weather Conditions. [23]

The results shown in figure 3.1 and present similar performance with the late fusion having the highest mean average precision and the mid-level fusion the second highest. With this knowledge further studies of fusion architectures are not

considered to be necessary. The low-level fusion is used because of its simplicity and the know performance. Figure 3.2 shows the low level fusion of lidar and camera. The lidar point cloud is projected onto the camera and a multi-channel image is generated, which additionally contains all depth information of the lidar sensor. An additional upsampling of the point cloud is possible in this case and should be done. The concatenated image channels are fed into feature encoder network, which determines bounding boxes and object classes considering the information from both sensors.

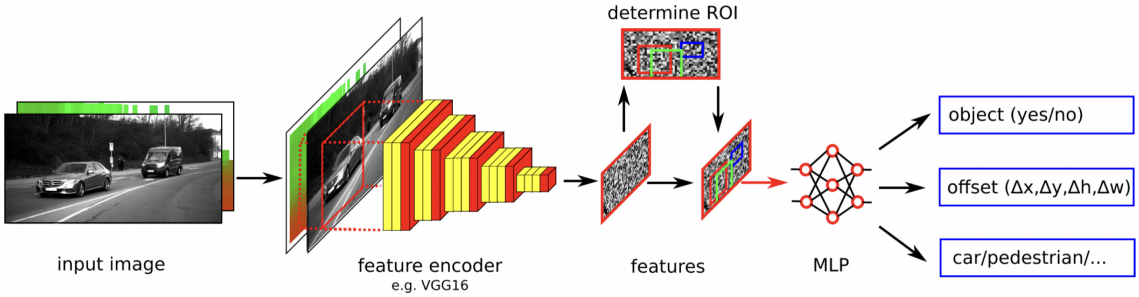


Figure 3.2: Low Level Sensor Data Fusion of Lidar and Camera. [23]

In the high-level fusion, shown in Figure 3.3 finder the feature extraction takes place completely independently. Only the final feature maps are fused in the last step to generate bounding boxes and their object classes.

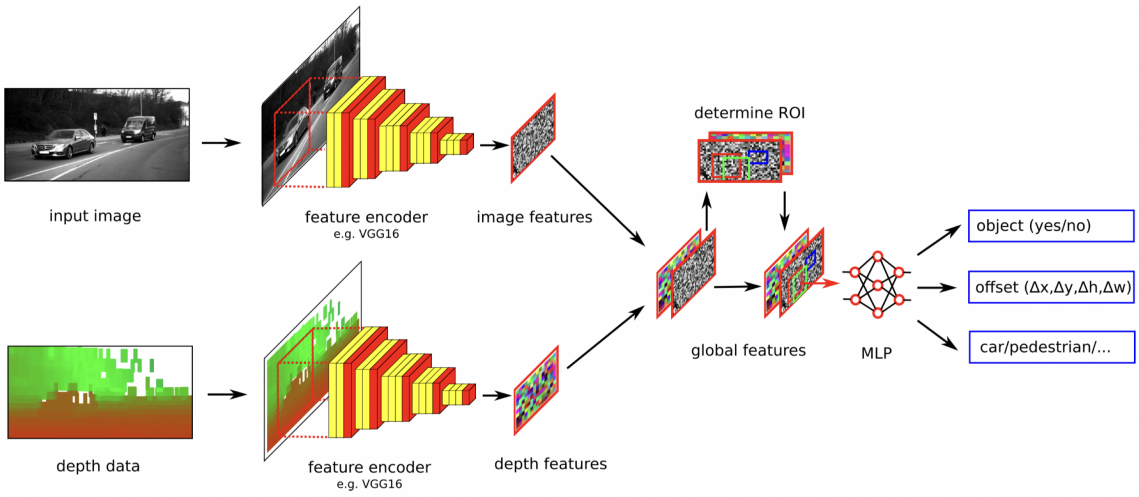


Figure 3.3: High Level Sensor Data Fusion of Lidar and Camera. [23]

The recently proposed solely upsampling based approach in [17] proves to be the most accurate low-level fusion and one of the most accurate methods overall in the autonomous driving KITTI vision benchmark [12] benchmark while still maintaining simplicity and is therefore selected for this work. Figure 3.4 shows the procedure of this method. In the first step, the raw data of the point cloud projected in the camera plane are visible. Which is very sparse.

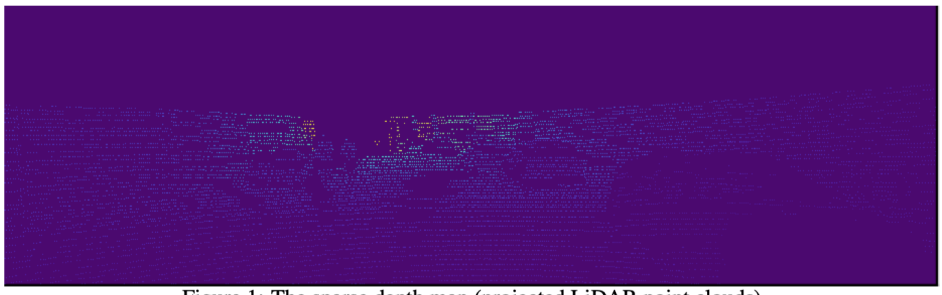


Figure 1: The sparse depth map (projected LiDAR point clouds)

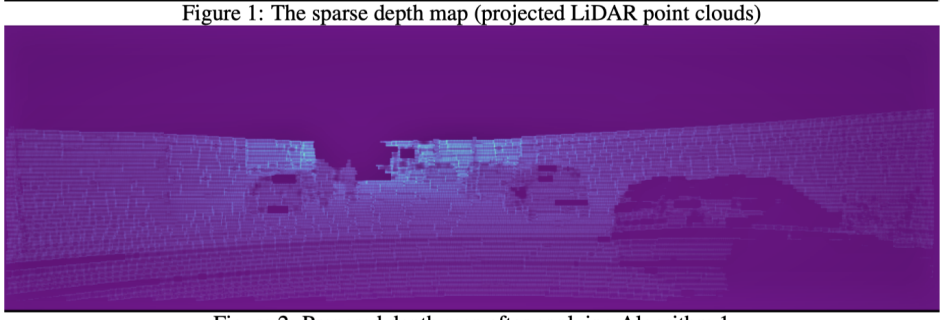


Figure 2: Prepared depth map after applying Algorithm 1

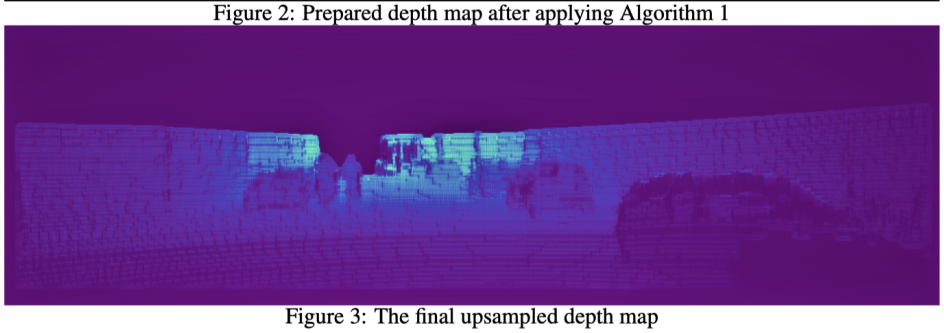


Figure 3: The final upsampled depth map

Figure 3.4: Weighted upsampling of the lidar point cloud. [17]

Following the projection, a sparse 2D depth map is obtained, to be used in the next step. Since the lidar depth map is sparse, it is necessary to extrapolate it. Therefore, by applying the weighted depth computation algorithm (Algorithm 1), figure 3.5 shows the described pseudo code. It is possible to create an image with full depth. However, as can be seen in Figure 1, only a minority of the pixels have non-zero values. Therefore, these pixels are called value pixels and other pixels are

called empty pixels. This particular step is to fill in the empty pixels. With respect to the structure of lidar data, as shown in Figure 1, the sparseness of lidar data in the vertical direction is larger than that in the horizontal direction. Consequently, in order to obtain a monotonic depth map, a function that fills all the empty pixels by searching the entire neighborhood of the pixel with longer vertical step than the horizontal neighborhood is used. However, for each empty pixel, our algorithm searches backward and forward to find all the valued pixels. When the number of valued pixels found is greater than zero in both directions, it assigns a value equal to the weighted average of all valued pixels found, unless it moves to the next pixel without giving it a value. Once the empty points are filled, a complementary algorithm (Algorithm 2), figure 3.6 visualizes the described pseudo code, is used to make the shapes in the depth image more coherent and to fill the small empty points (see Figure 3). A rectangular area around an empty pixel is examined by this algorithm, and if the number of evaluated pixels is greater than a threshold, the empty pixel is given a value equal to the weighted average of all evaluated pixels. A threshold value must be chosen so that every inner pixel of a shape is filled. Similar to Algorithm 1, the weighted structure in Algorithm 2 aims to be smooth but computationally expensive even with large steps.[17]

Algorithm 1 Weighted Depth Filling Algorithm

S_x = step length for X-axis
S_y = step length for Y-axis
for Each pixel in depth map **do**
  if $Pixelvalue = 0$ **then**
    $S = 0$
    $S_c = 0$
    for searched pixels forward with S-x footstep **do**
      if $Pixelvalue \neq 0$ **then**
        $d \leftarrow$ distance between searched pixel and reference pixel
        $S \leftarrow S + (1/d)\times$(searched pixel value)
        $S_c1 \leftarrow S_c1 + (1/d)$
      end if
    end for
    for searched pixels backward with S-x footstep **do**
      if $Pixelvalue \neq 0$ **then**
        $d \leftarrow$ distance between searched pixel and reference pixel
        $S \leftarrow S + (1/d)\times$(searched pixel value)
        $S_c2 \leftarrow S_c2 + (1/d)$
      end if
    end for
    $i \leftarrow i + 2$
  end if
  if $Sc1 > 0$ & $Sc2 > 0$ **then**
    Pixel value$\leftarrow S/(Sc1 + Sc2)$
  end if
end for
Do the same for Y-axis

Figure 3.5: Weighted Depth Filling Algorithm. [17]

Algorithm 2 Complementary Weighted Depth Filling Algorithm

Ss = step length for Searching
for Each pixel in the depth map **do**
  if $Pixelvalue = 0$ **then**
    $S = 0$
    $S_c = 0$
    for search a sqaure around the pixel Ss×Ss **do**
      if $Pixelvalue \neq 0$ **then**
        $d \leftarrow$ distance between searched pixel and reference pixel
        $S \leftarrow S + (1/d)\times$(searched pixel value)
        $S_c \leftarrow S_c + (1/d)$
      end if
    end for
    $i \leftarrow i + 2$
  end if
  if $Sc/(Ss \times Ss) > Thresold$ **then**
    Pixel value$\leftarrow S/Sc$
  end if
end for

Figure 3.6: Complementary Weighted Depth Filling Algorithm. [17]

3.2 Object Detection

Deep Learning based object detectors are split into two categories over the past few years. On the one hand single stage detectors which do everything from the input image to the final bounding box in a single stage. However, two stage detectors have a separate stage at the end to do the sparse prediction. They are usually able to achieve a slightly higher accuracy at the high computational cost. Figure 3.7 visualizes the different architectures. As this project focuses on a real-world applicable pipeline the single-stage-detectors is chosen in regards of the computational cost.

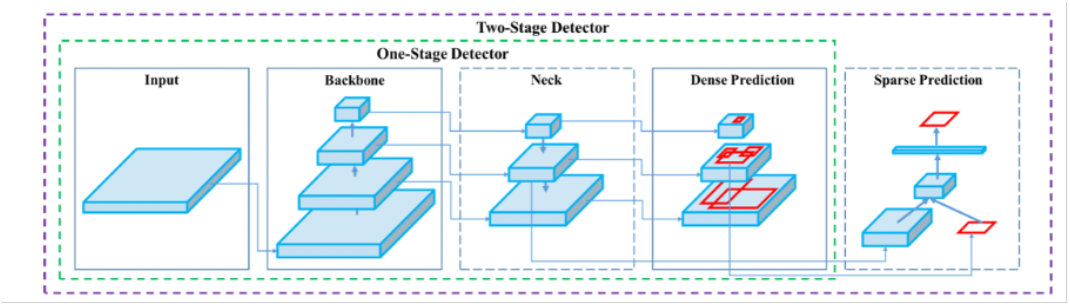


Figure 3.7: One-Stage Detector vs. Two-Stage Detector. [5]

The most well known single stage detector is called YOLO (You Only Look Once) and is continuously improved with small changes to the architecture and training. As shown in figure 3.7, YOLO consist out of a backbone, neck and head. The input Image is usually a 3-channel image. The backbone which is usually an ImageNet pretrained classifier. The more recent version YOLOv4 introduces a custom backbone, called the CSPDarknet53. The neck is considered to be the layers between backbone and head, and these layers are usually used to collect feature maps from different stages. Usually, a neck is composed of several bottom-up paths and several top-down paths. The head stays unchanged compared to YOLOv3 [24].

Figure 3.8 visualizes the architecture of YOLO. The backbone at the beginning for feature extraction and the object detection layer in three different scales as well as the respective shortcut connections to these are highlighted.

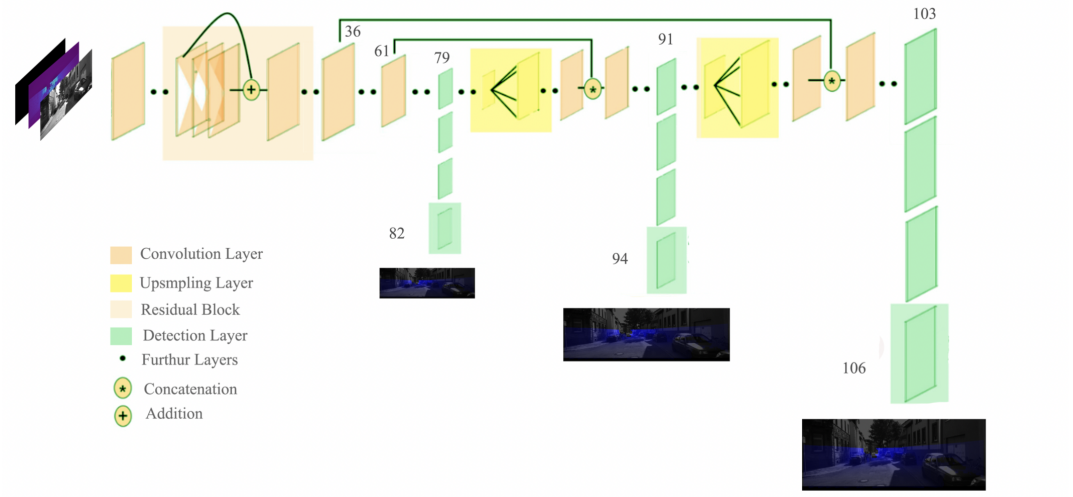


Figure 3.8: Architecture of YOLO.[17]

As discussed, the majority of improvements compared to older version is due to small changes. These improvements are either part of training or a small and fast post-processing step. During training the "Bag of Freebies (BoF)" consist of CutMix and Mosaic data augmentation, DropBlock regularization and Class label smoothing. Furthermore, Self-Adversarial Training is applied, the grid sensitivity is eliminated, multiple anchors for a single ground truth are used and Random training shapes are used. The Bag of Specials (BoS), small improvements with a high return, are based on a new activation function (Mish), Cross-stage partial connections (CSP) and Multiinput weighted residual connections (MiWRC).[5]

Scaled-YOLOv4 (green highlighted in figure 3.9) was chosen for this work, the latest and most accurate version so far. The Scaled YOLOv4 lies on the Pareto optimality curve. For every neural network , there is always such a YOLOv4 network, which is either more accurate at the same speed, or faster with the same accuracy. YOLOv4 is the best in terms of speed and accuracy today reaching up to 1774 FPS on a modern GPU [25].

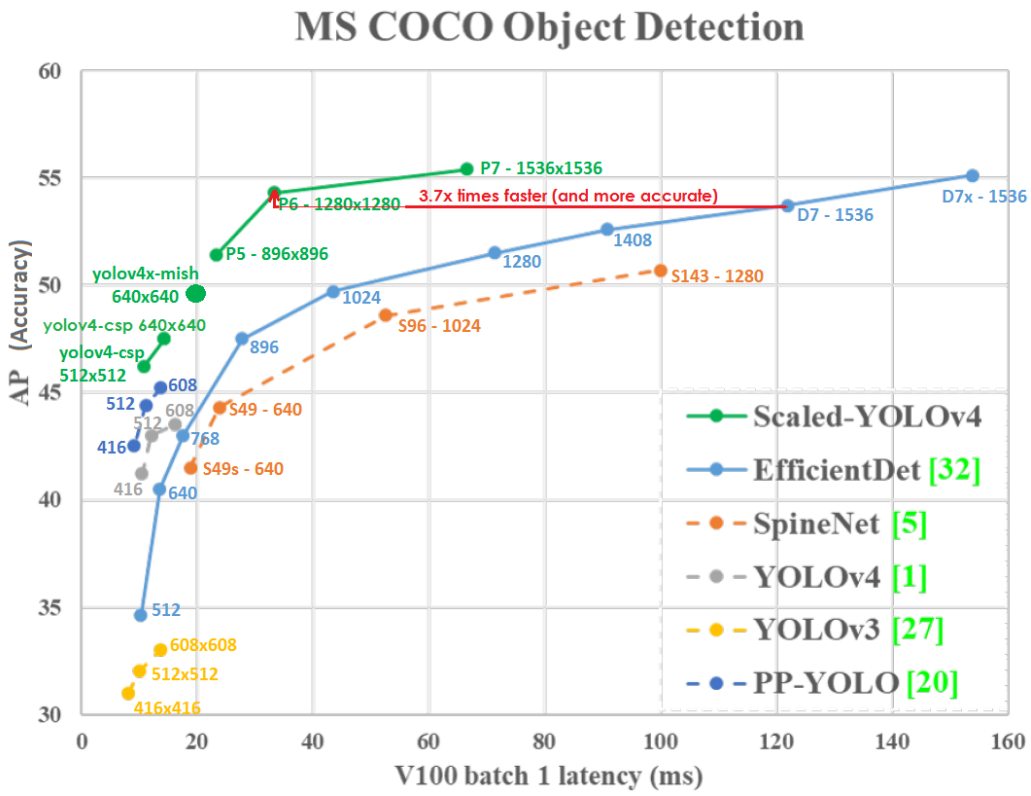


Figure 3.9: MSCOCO Object Deetection Benchmark Evaluation.[17]

Especially noteworthy are the new introduced so called CSP connections which are extremely efficient, simple, and can be applied to any neural network. The general idea is that half of the output signal goes along the main path to generates more semantic information with a large receiving field and the other half of the signal goes bypass in order to preserves more spatial information with a small perceiving field. [5]

3.3 Pre Training Deep Neural Networks for Thermal Imaging

Deep neural networks need a very large amount of training data to avoid overfitting which describes the learning of the training data instead of it's features. An overfitted model will perform poorly on non-training data. Such amount of data is usually not available for the specific object recognition task, possibly due to the long time it takes to manually annotate examples with location information. In an attempt to overcome this, essentially all of the reviewed work on object recognition uses some form of pre-training, which allows the base network to be trained in a different way before it is trained on the recognition task. Most commonly the MSCOCO dataset presented earlier. Here, the intuition is that during pre-training, the network learns very general features that are also applied to similar tasks. It can then use these features to be subsequently trained on the limited amount of recognition data available to the task.

As discussed in [27] thermal object detection in the domain of autonomous driving lacks the ability to generalize. Performing poorly when trained on the KAIST Dataset [6] and tested on the Flir Dataset [10]. The small thermal and non-diverse thermal datasets are not sufficient in order for the network learn the high level appearance of the thermal domain. Furthermore [2] notes that "there is an considerable lack of work undertaken for cyclist detection using multi-spectral data".

Therefore further consideration are being proposed in order to overcome the domain gap especially for underrepresented classes such as cyclists. Creating a diverse dataset at the level of MSCOCO in the thermal domain is an almost impossible task. Since it consists of more then 200.000 completely different images with 1.5 million pixel accurate object bounding boxes. For this reason, the following chapter 4 explores ways to adapt the dataset to the thermal domain and thus exploit the available labels and diverse scenes.

Chapter 4

Transfer Learning for Thermal Imaging

Transfer learning is when a model developed for one task is reused to work on a second task. Object detection networks are usually pretrained on the MSCOCO dataset including more then 200.000 labeled images and 1.5 million object instances. [22] and fine tuned on a smaller set of application specific data. As discussed in 3 opens the thermal imaging domain a gap.

The previously mentioned FLIR dataset, an example image shown in 4.1

Figure 4.1: Example Image from the Dataset with Detected Objects. [10]

is one of the very few and most recent labeled automotive dataset in the thermal domain and therefore used for the following considerations. It is acquired via a RGB and thermal camera mounted on a vehicle with annotations created for 14.452 thermal images. It primarily is captured in streets and highways in Santa Barbara, California, USA from November to May with clear-sky conditions at both day and night. [10]

4.1 Generating Artificial Thermal Images

ThermalWorld and Thermal GAN

In the following section a new method is proposed to close the domain gap. Previous work has dealt with methods to generate artificial thermal images, which have shown to improve re-identification networks. In this context, the ThermalWorld dataset was also created, which contains thermal images as shown 4.2 in the form of absolute temperate and

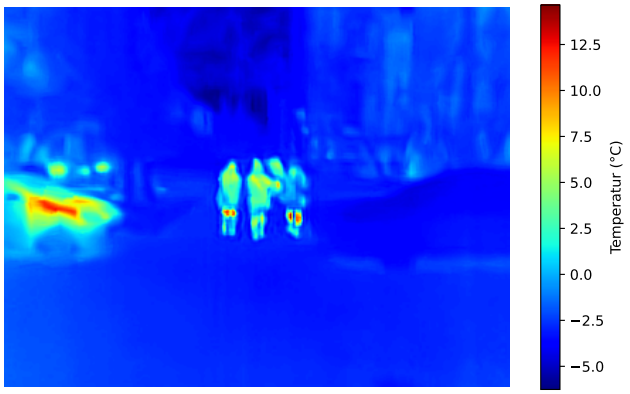


Figure 4.2: Example Image from the Dataset with the Corresponding Temperature.

semantic segmentations as seen in figure 4.3. [20]

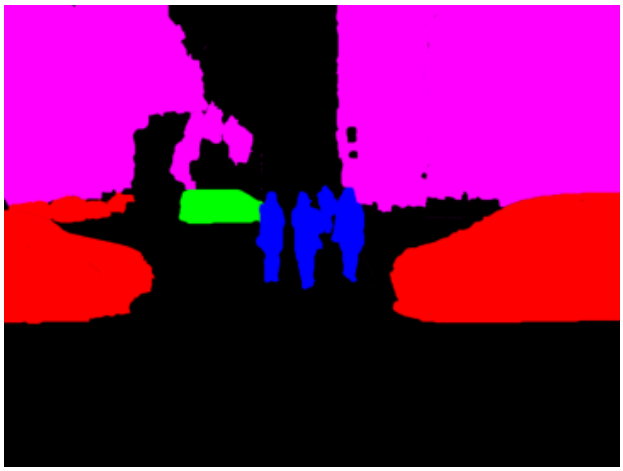


Figure 4.3: Example Image from the Dataset with the Corresponding Semantic Segmentation.

This combination of semantic segmentation, which assigns a class to each pixel, makes it possible to extract the absolute and relative temperature distribution for each class. Figure 4.4 visualizes the temperature distribution of all classes in the entire ThermalWorld dataset.

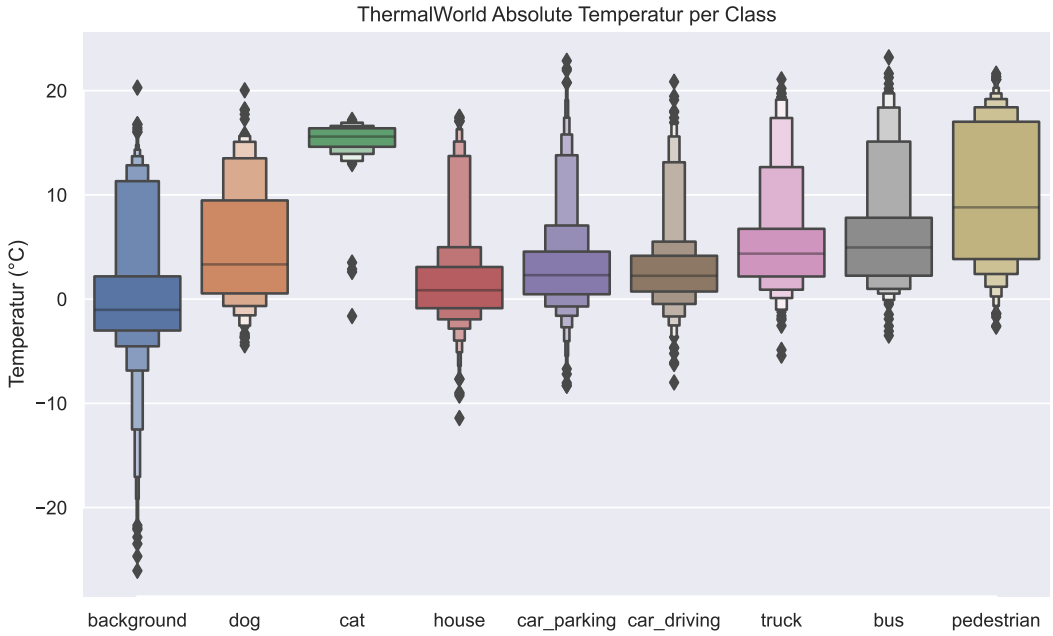


Figure 4.4: Absolute Temperature Distribution of each class in the ThermalWorld dataset.

The original work proposed a constrained GAN (Generative Adversarial Network) which is able to generate artificial thermal images from a temperature segmentation and rgb images. This is being done by predicting the relative temperature gradients instead of its absolute temperature. It is assumed that the thermal segmentation that provides average temperatures of the emitting objects in the scene could resolve possible ambiguities. [19]

Exploiting Panoptical Labels

The newest category of MSCOCO labels is called panoptical labels. These are a combination of semantic and instance segmentation, taking into account the stuff and things categories, which assign a meaningful class to each pixel. [18] Figure 4.5 visualizes these panoptical labels and their corresponding rgb images.

Figure 4.5: Panoptical Labels and Corresponding Images from the MSCOCO Dataset. [18]

It can be seen that they offer an accurate description of the scene. Figure 4.6 visualizes the difference between semantic, instance and panoptic segmentation. It can be clearly seen, that panoptic is an advanced combination of both.

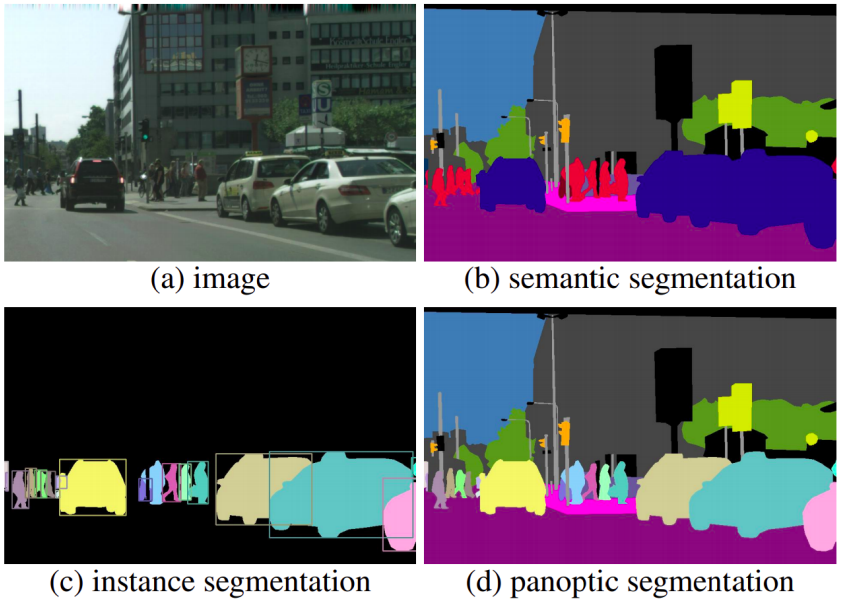


Figure 4.6: RGB Image, Semantic Segmenation, Instance Segmentation and Panoptic Segmentation of the same Scene. [18]

These pixel-precise panoptical labels are combined in the next step with the previously extracted distribution functions of the different classes. These are approximated by a normal distribution, from which a sample is drawn for each instance in the image with the associated class. We start with a background temperature

for the image as shown in figure 4.7.

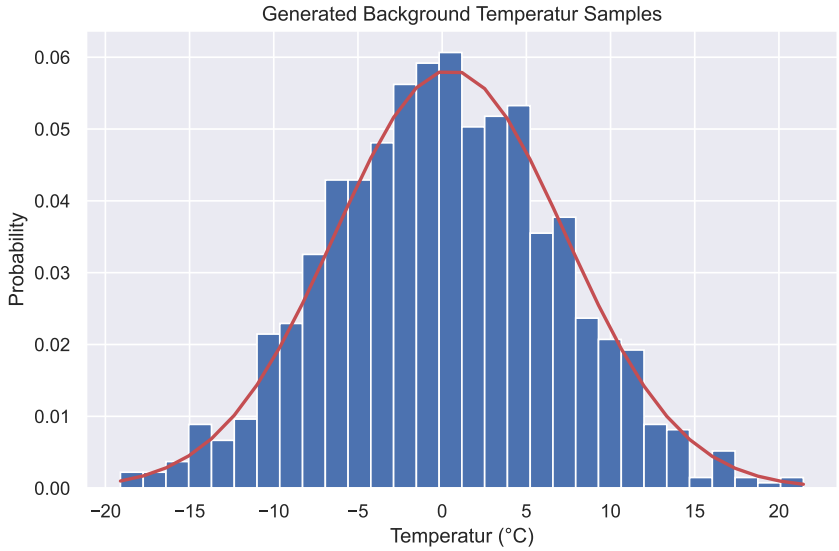


Figure 4.7: Absolute Background Temperature Distribution and Example Values.

Subsequently, based on the background temperature, a sample for the relative temperature belonging to the respective class is drawn. In this way, a dense and realistically distributed thermal temperature segmentation is created.

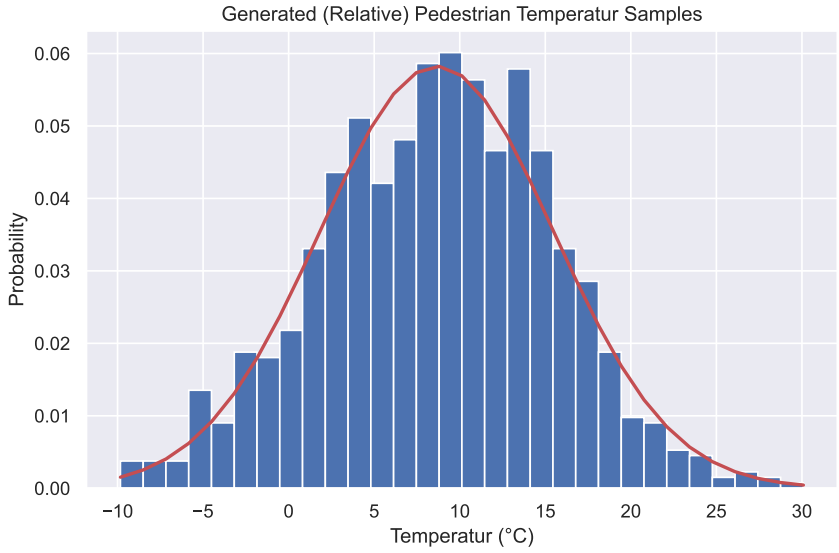


Figure 4.8: Relative Temperature Distribution and Example Values of a Person.

This procedure is applied to the entire MSCOCO dataset in the following step. In this way, it receives additional thermal annotations. This large scale thermal segmented dataset is then extended with the constrained ThermalGAN to complete thermal images.

The following figure 4.9 shows the input (RGB image) and the output (artificial thermal image). Especially persons are displayed very realistically, which is most likely due to the training of the ThermalGAN with focus on person re-id. However, as seen in (b), it also fails slightly and produces images that are difficult to interpret in some situations. In addition, the images lose some of their resolution and detail.

Figure 4.9: MSCOCO Images and Corresponding Generated Thermal Images

4.2 Evaluating MSCOCO Performance

In the following, two identical Scaled-Yolov4-CSP networks were trained. In one case, the network was pre-trained with the normal MSCOCO dataset and tuned to the new domain fine with the Flir thermal dataset. In the second case, the network was additionally pre-trained with the fake thermal images. In addition, the images

lose some of their resolution and detail.

In the following diagram 4.10, the evaluation by means of the test data set is visible. In particular, it is noticeable that the bicycle has a much lower precision than the rest.

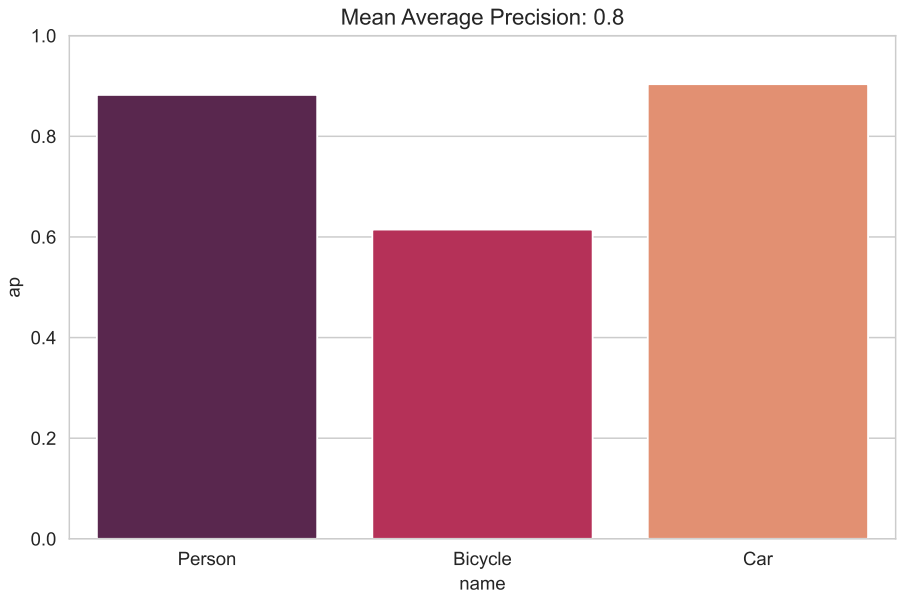


Figure 4.10: Average Precision of MSCOCO + Flir.

Additionally in figure 4.11 the detailed Precision vs. Recall graph, which shows the problem even more clearly. This exactly matches the previously described problem over in the thermal domain.

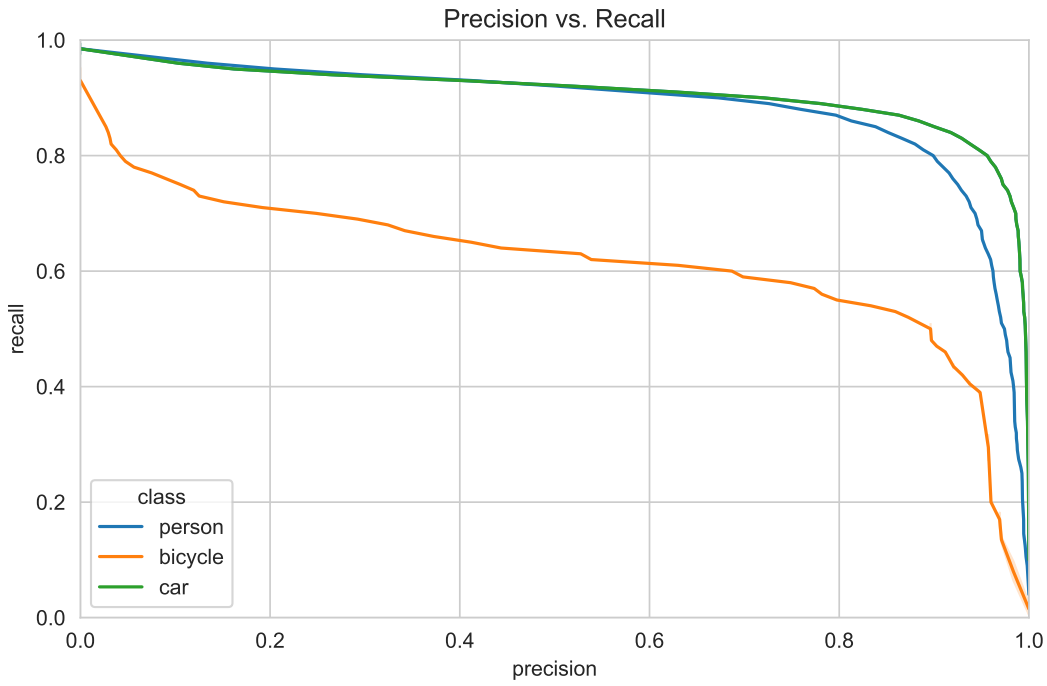


Figure 4.11: Precision vs. Recall of MSCOCO + Flir.

4.3 Evaluating MSCOCO-thermal Performance

In contrast, the following evaluation is carried out using additional artificial thermal images. It is clearly visible that the underrepresented class of the bicycles in the training data set cannot be learned sufficiently in with the normal training. In contrast, with additional artificial thermal images, the network was able to learn a more general representation of the thermal domain and was thus able to adapt much faster with less data to new problems.

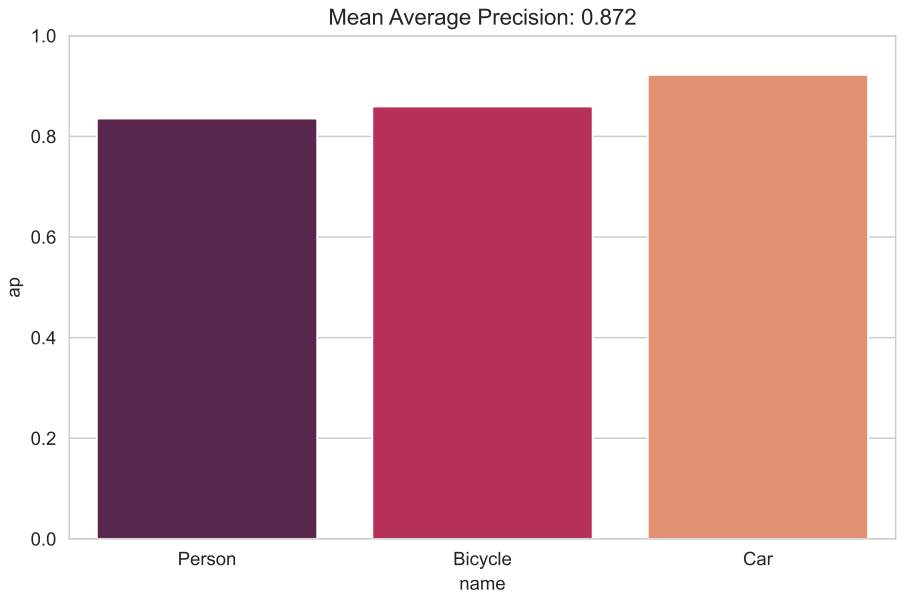


Figure 4.12: Average Precision of MSCOCO thermal + Flir.

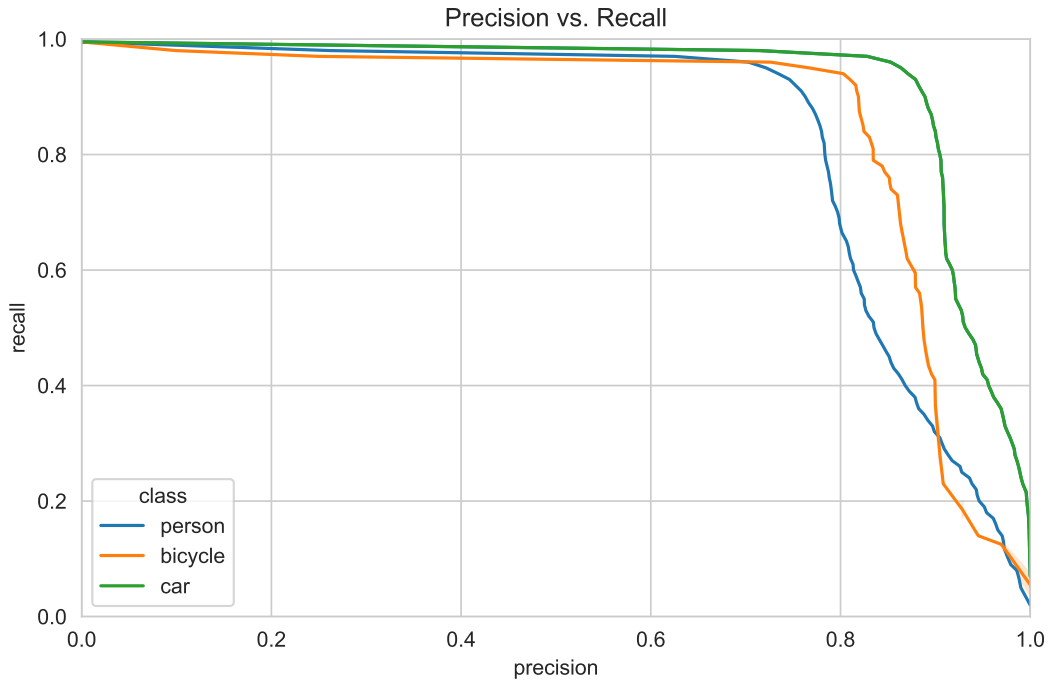


Figure 4.13: Precision vs. Recall of MSCOCO thermal + Flir.

Chapter 5

Evaluating Thermal and Lidar Imaging in Adverse Weather Conditions

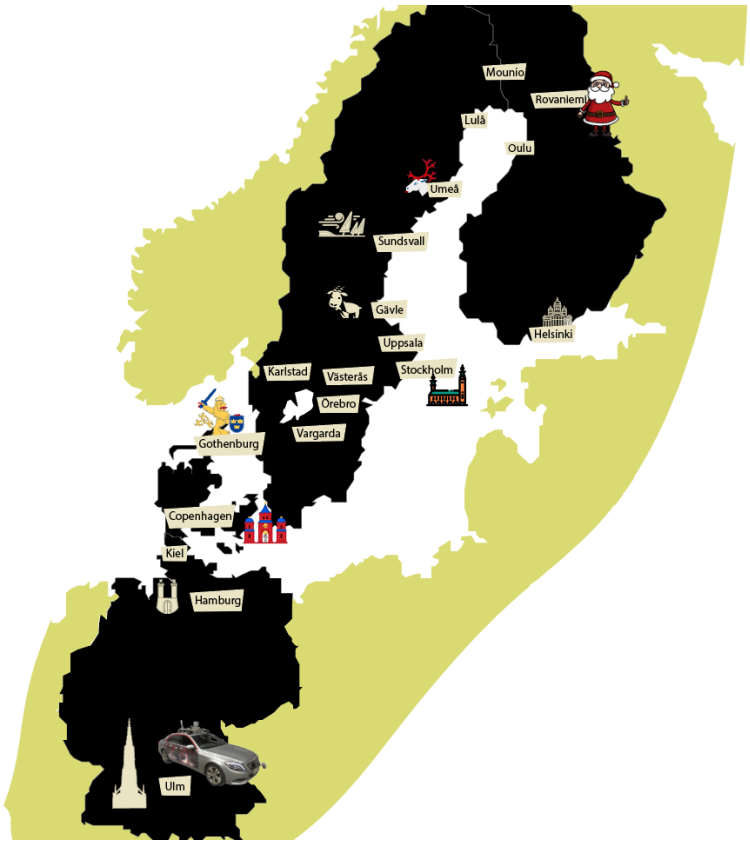


Figure 5.1: Geographical Diversity of the SeeingThroughFog Dataset. [4]

Most of the data sets and test areas of autonomous cars to date take place under optimal weather conditions. However, some work is taking place on domain adaptation and weather conditions. Recently, the project "Seeing Through Fog Without Seeing Fog" produced a series of publications on the topic as well as a matching dataset that includes a variety of sensors and weather conditions. Of particular interest for this work is the fact that thematic images are recorded and included, but have not been used by anyone.

Figure 5.2 shows the sensor setup. Relevant is the thermal camera and the lidar on the roof.

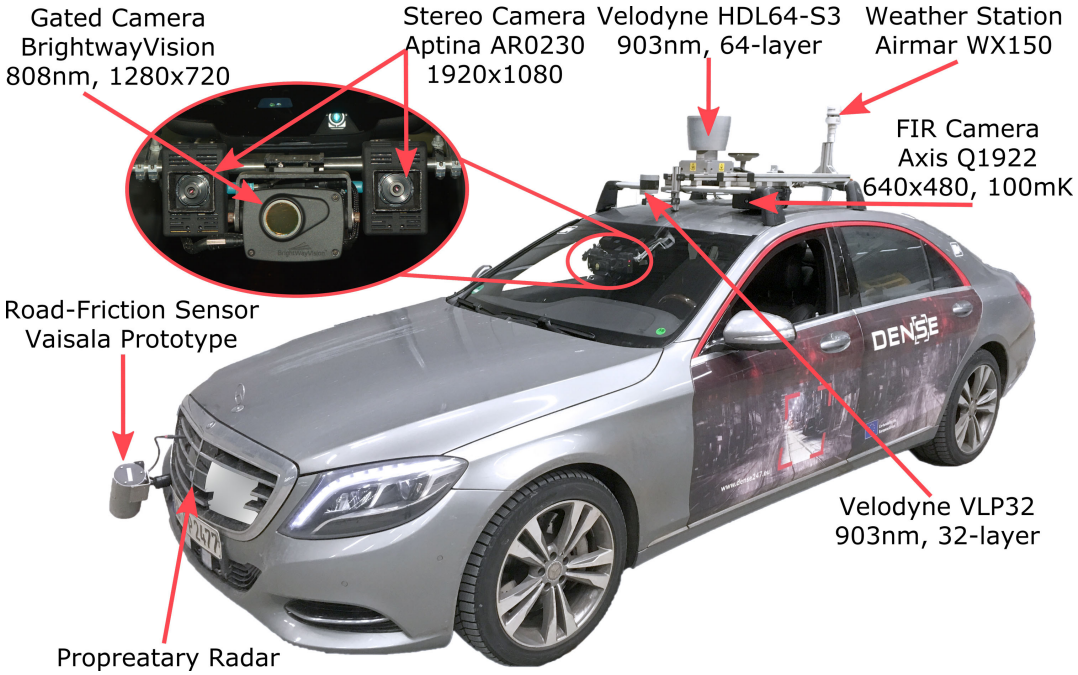


Figure 5.2: Sensor Setup on the Project Car. [4]

This data set forms the basis for the evaluation of Adverse Weather Conditions in the further course. The network is trained with the corresponding training data set and tested with the existing test split.

It was also the goal to use additional self-recorded data, however, in this case a problem has arisen. Already with the relation between camera field of view and the lidar in the data set, the solely depth upsampling does not deliver perfect results. With the more than three lower lidar of the aalborg university the upsmaplling failed completely. The SeeingThroughFog dataset uses the HDL-64e lidar [4] scanner with a vertical resolution of 0.4°[15]. The lidar scanner HDL-32e at Aalborg

University considered for this project has a more then 3 times lower resolution of 1.33°[16] which results in only a few lays being inside the scene which.

The following is the analysis of Adverse Weather Conditions. As a benchmark, RGB + Lidar is used with the same procedure to show the advantages and disadvantages of the different spectra.

5.1 Dense Fog Daytime

Thermal and Lidar

In the dense fog during the day failed far infrared completely. No detail of the environment can be guessed and the image does not contain any relevant information.

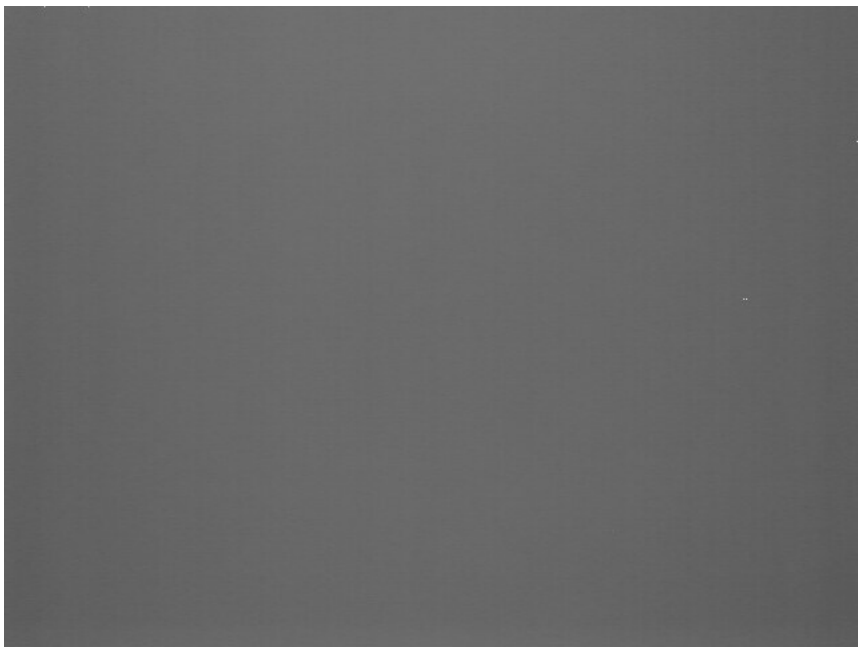


Figure 5.3: Image ID: 200, Thermal Imaging fails in dense fog.

In the dense fog during the day failed far infrared thermal imaging and near infrared lidar sensors both fail. No detail of the environment can be guessed and the image does not contain any relevant information.

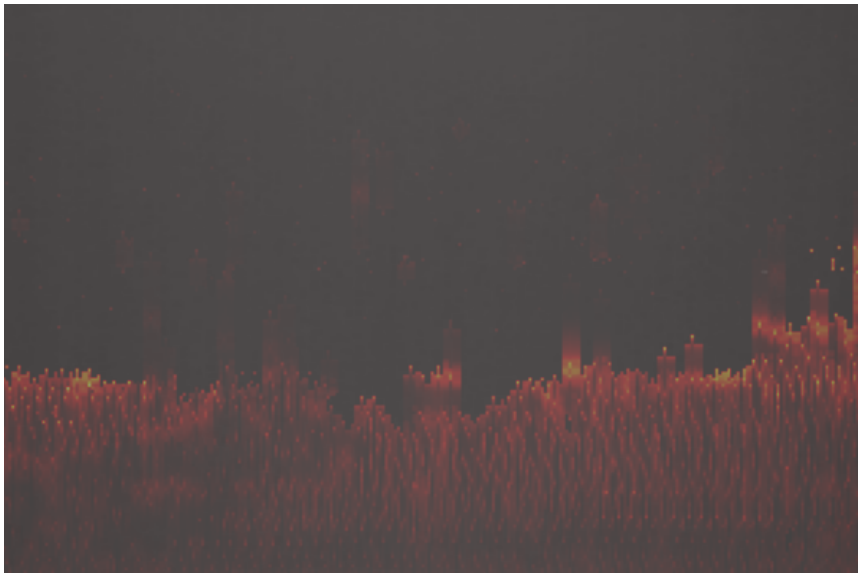


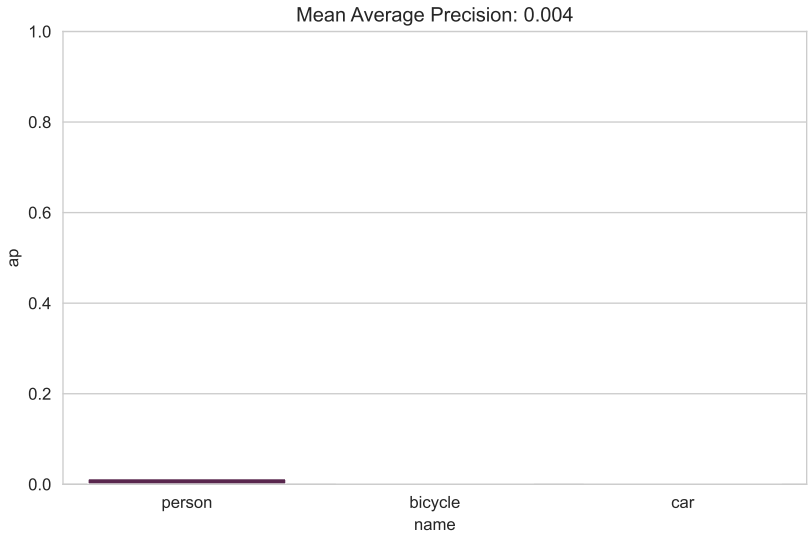
Figure 5.4: Image ID: 200, Also lidar fails ins combination with thermal imaging in dense fog.

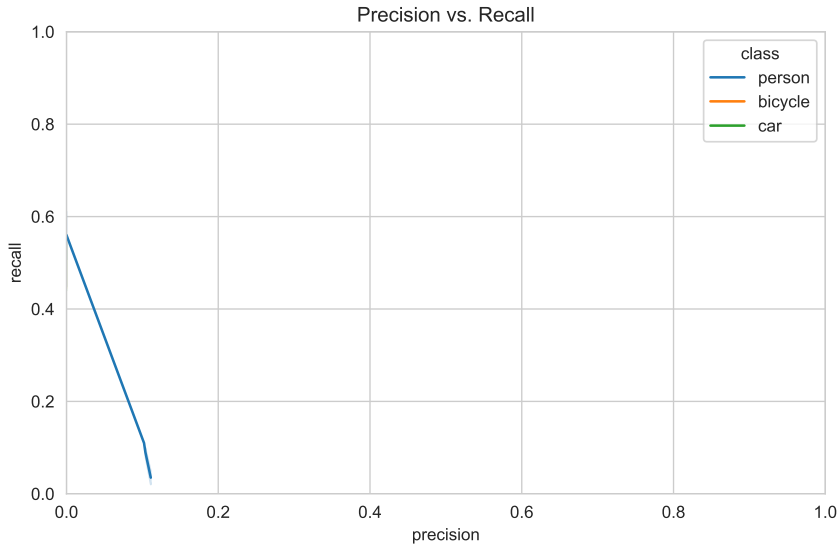
The following image shows an intersection within the downtown area with just a person visible. This scenario represents the best case in dense fog.

Figure 5.5: Image ID: 2190, Thermal Imaging fails in dense fog and can only correctly see close persons.

During the day in dense fog, the combination of thermal imaging and lidar

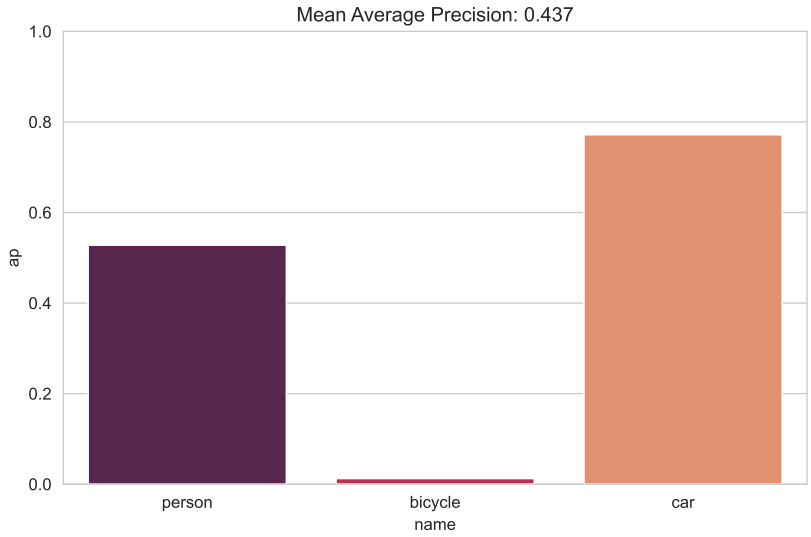
achieves an average precision of only 0.004. Bicycles and cars can not be detected at all.





RGB and Lidar

RGB and lidar, on the other hand, can be people reasonably well, and cars even better. Only bicycles are a problem in this example, but this is most likely due to the fact that there is no representative number and quality of bicycles in heavy snow.



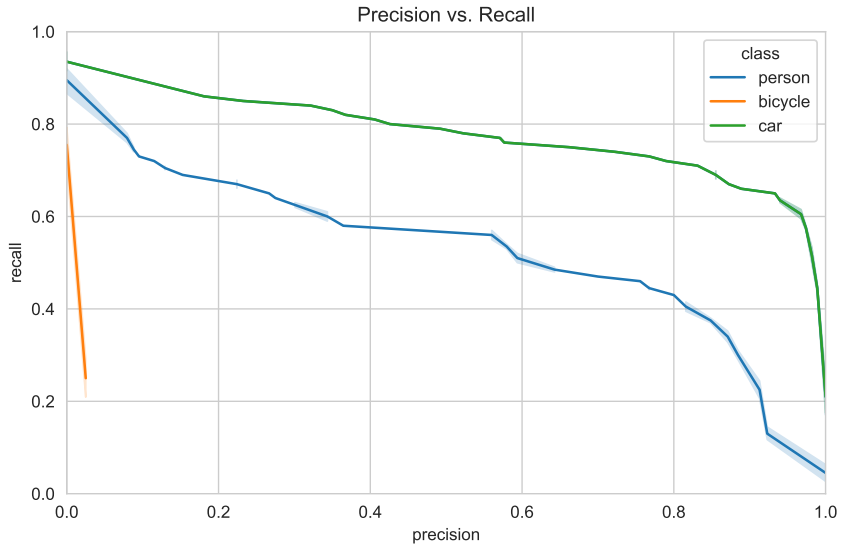


Figure 5.6: Image ID: 200, RGB shows a clear image of the envivroment compared to thermal imaging.

Figure 5.7: Image ID: 2190, RGB shows a clear image of the envivroment compared to thermal imaging.

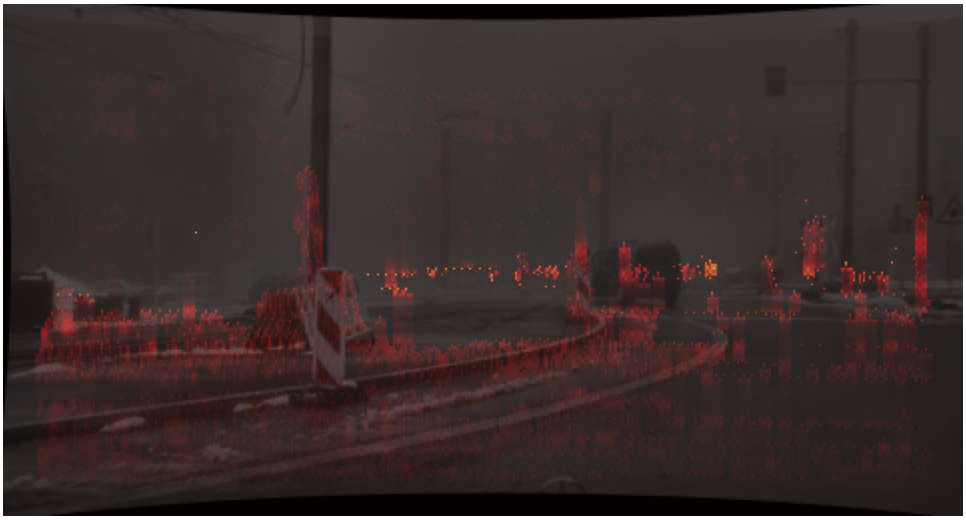
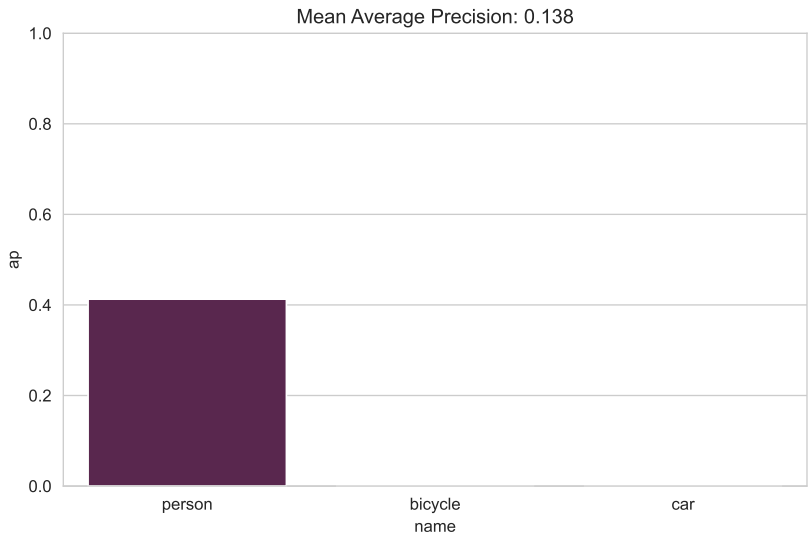


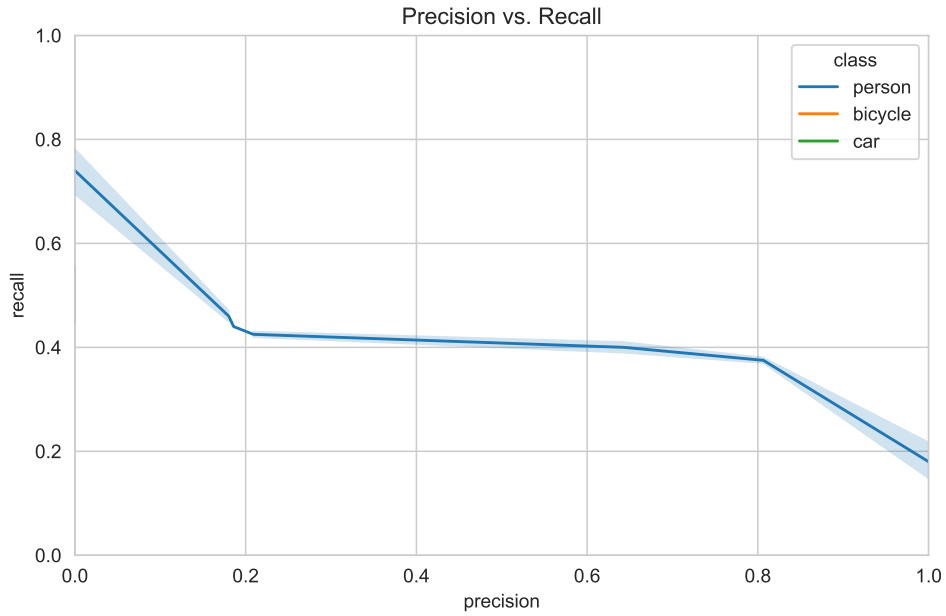
Figure 5.8: Image ID: 2190, Lidar captures mostly noise in these conditions.

5.2 Dense Fog Nighttime

Thermal and Lidar

Dense fog at night shows the same behavior as during the day. Interestingly, however, with a higher precision for the few correctly detected persons.





In dense fog at night, the sitation for thermal imaging does not improve much compared to rgb. On the other hand, the lidar is able to record at least some relevant detections.

RGB and Lidar

As before, the visible domain has clear advantages, especially for cars and bicycles.

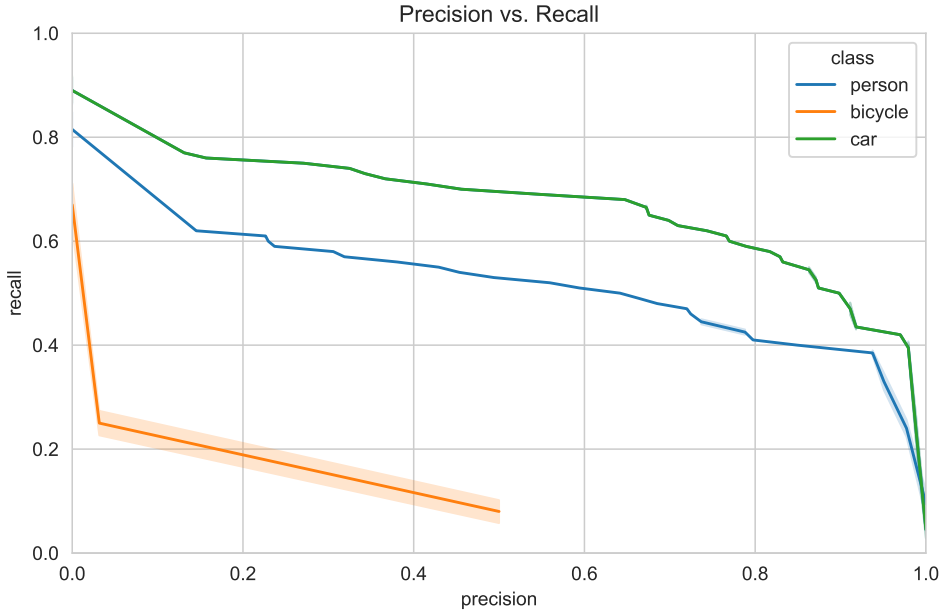
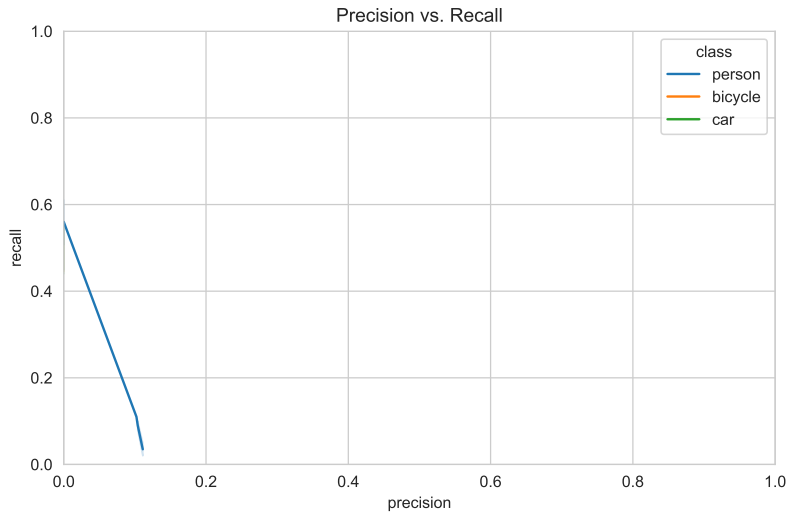


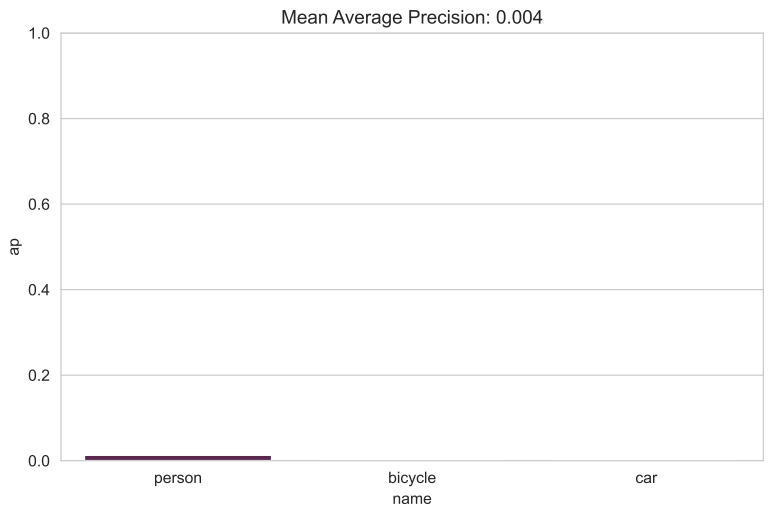
Figure 5.9: Image ID: 900, Lidar captures mostly noise in these conditions.

5.3 Light Fog Daytime

The effects of the combination of low ambient temperature and high humidity described above also make the lidar/thermal combination virtually blind here.

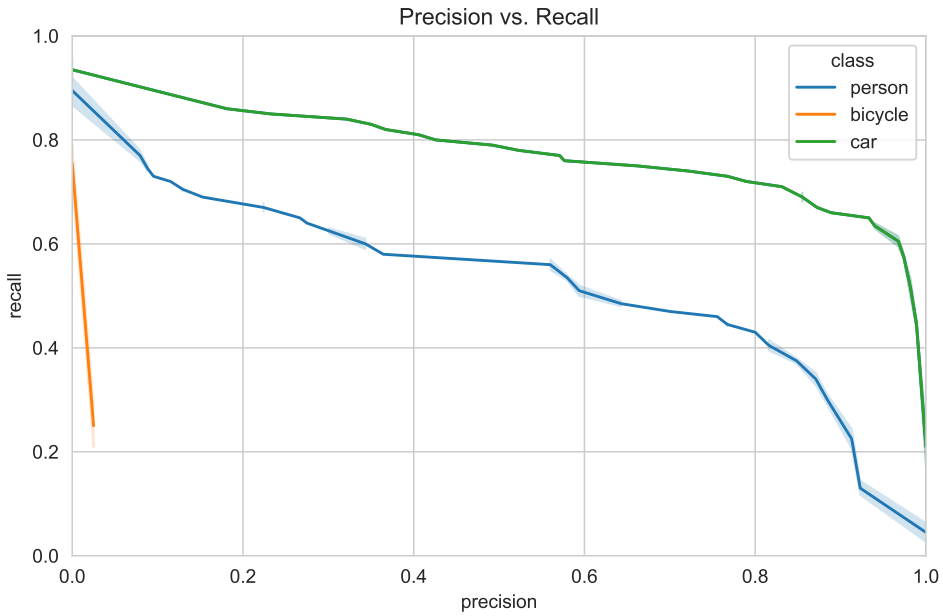
Thermal and Lidar

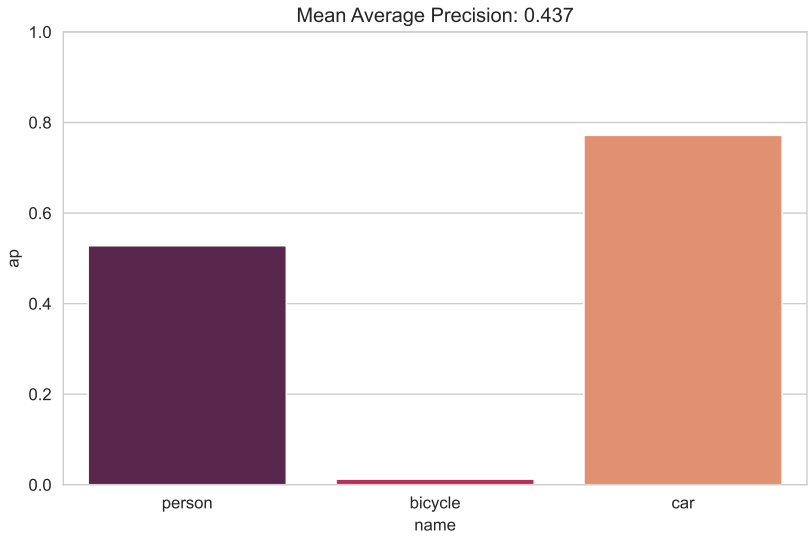




RGB and Lidar

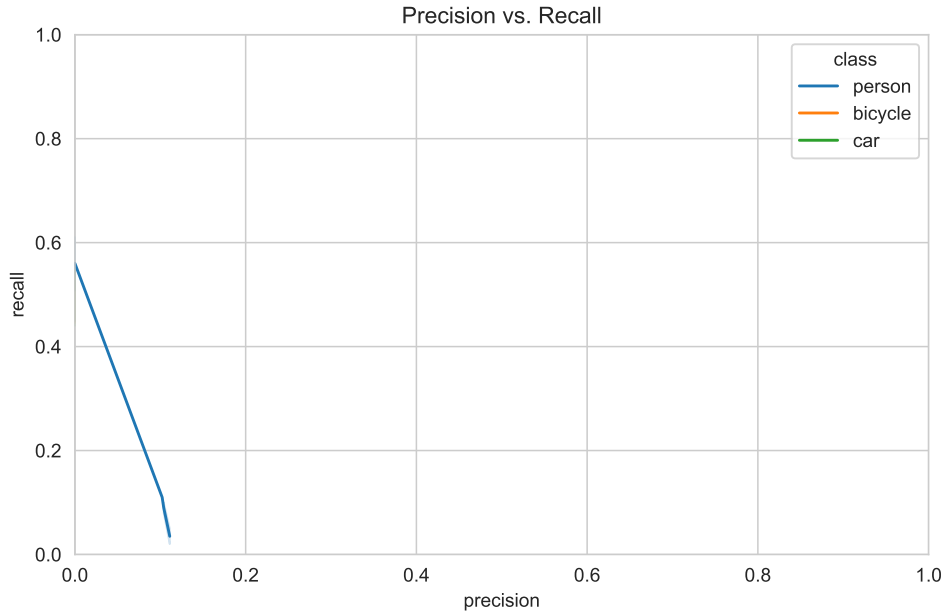
Previously seen effects are again visible here, that in the visible domain objects can still be perceived.

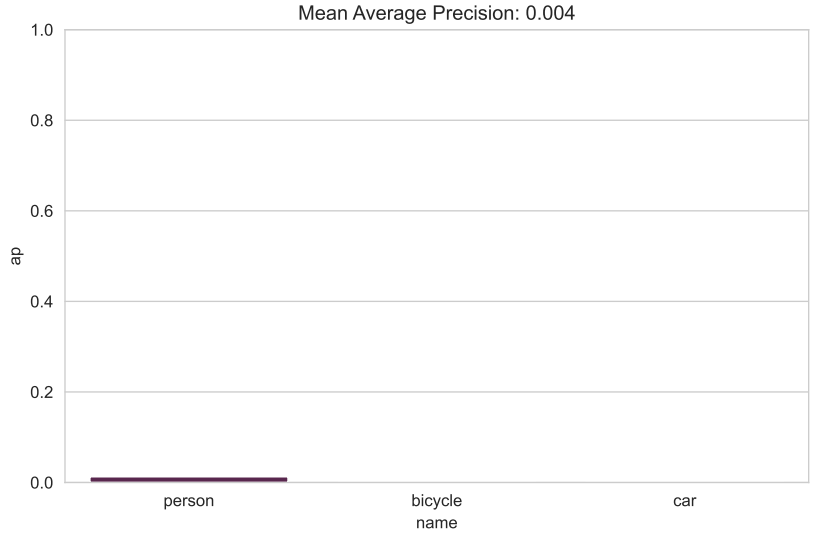




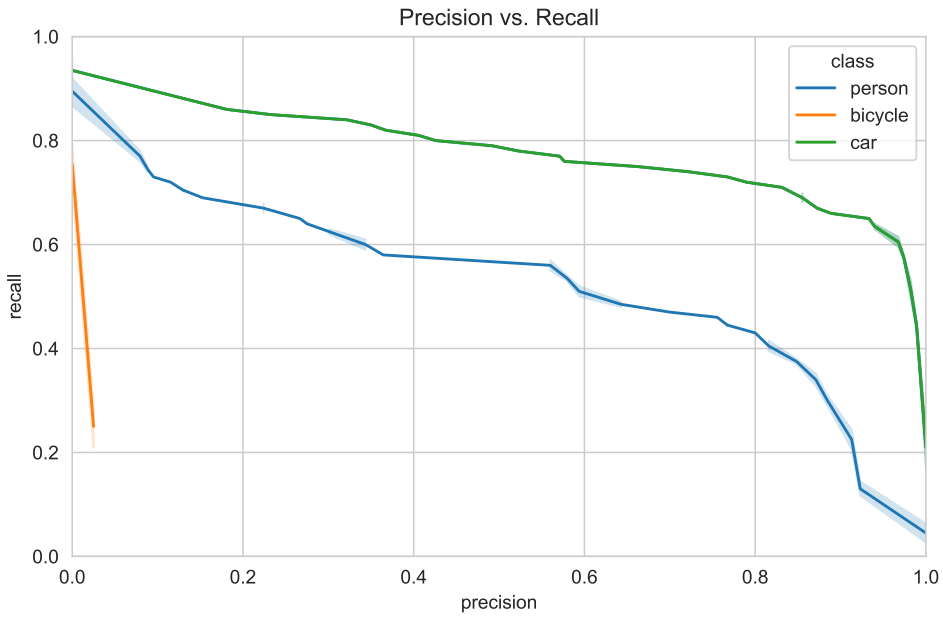
5.4 Light Fog Night

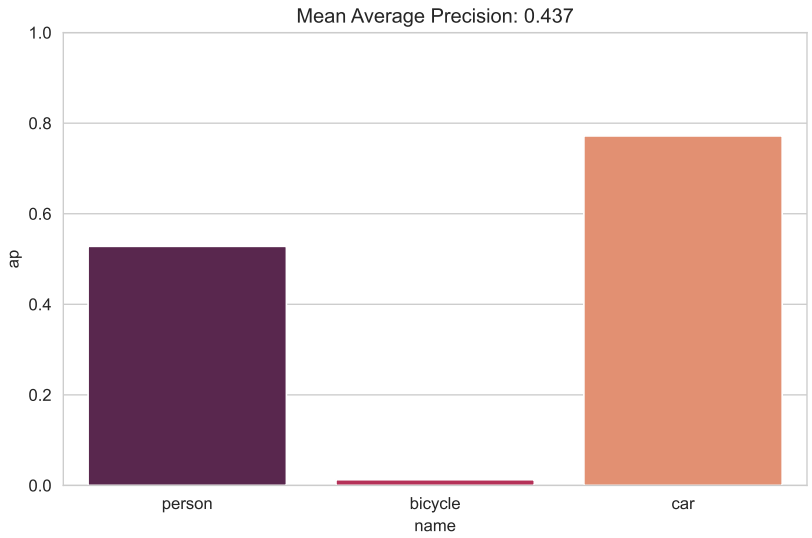
Thermal and Lidar





RGB and Lidar

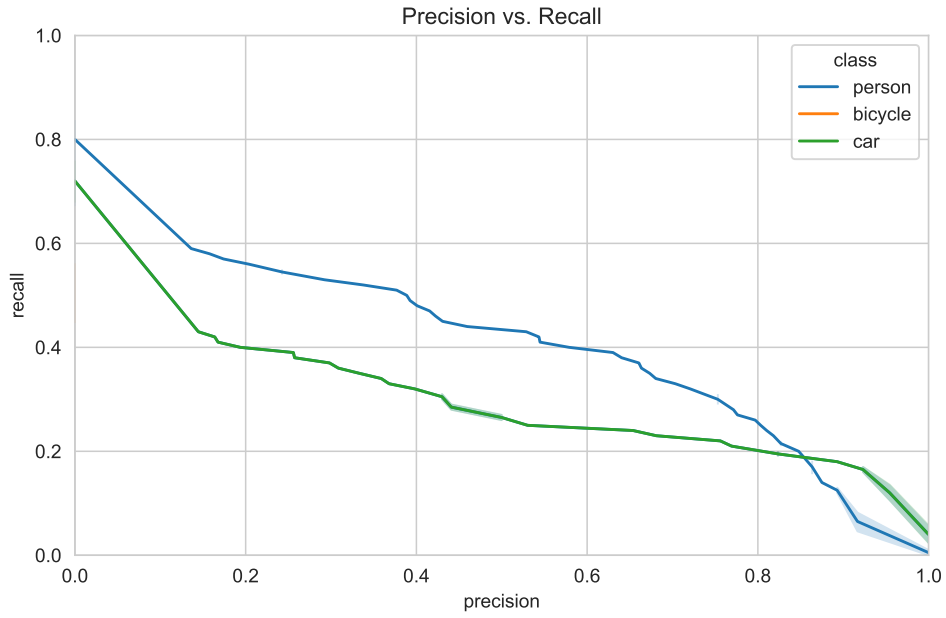


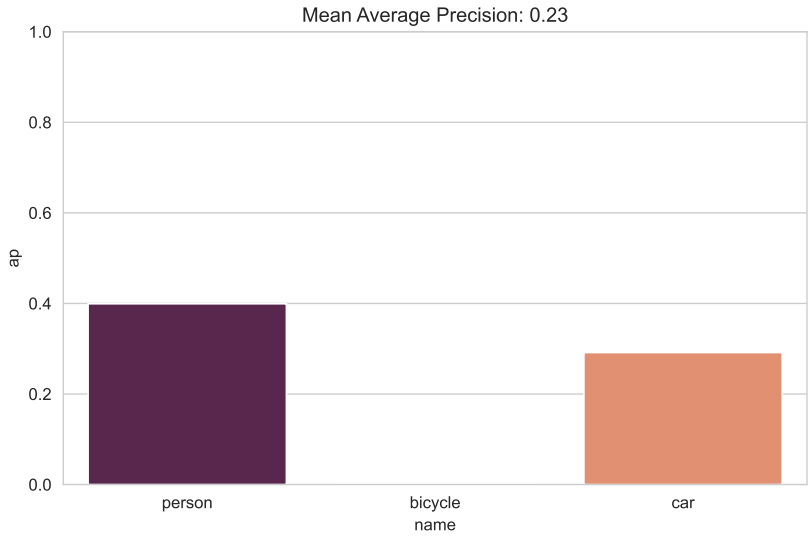


5.5 Snow Day

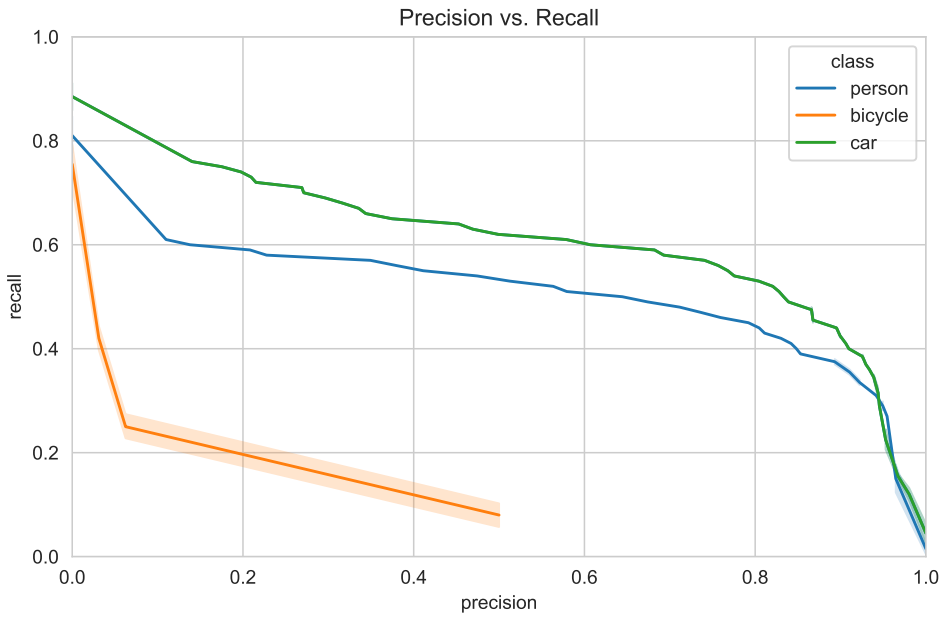
For the sake of completeness, snow by day and by night is also shown below. It can be seen that the basic effects of water particles in the air are the same and reducing the thermal imaging visibility. However, dense snow flakes have a much smaller impact on the thermal domain compared to light fog. The direct view to the environment and it's far infrared emitting surfaces between the snowflakes seems to be advantageous compared to fog where the radiations needs to pass through the water molecules.

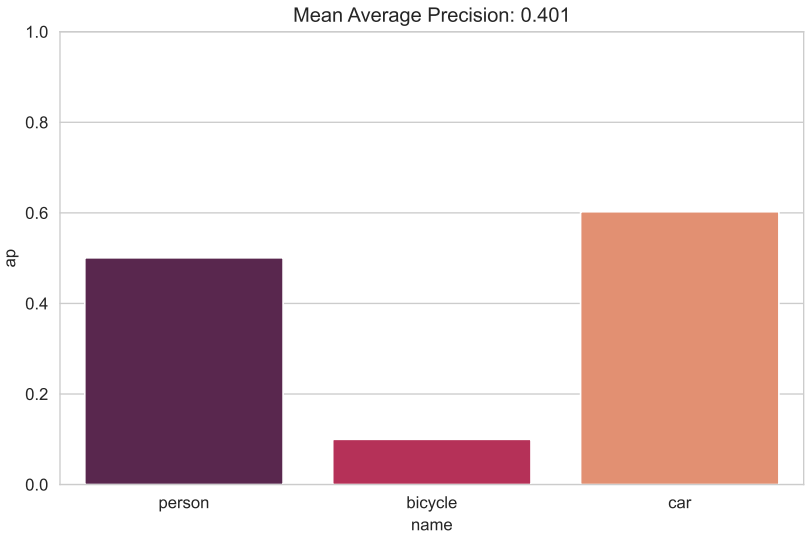
Thermal and Lidar





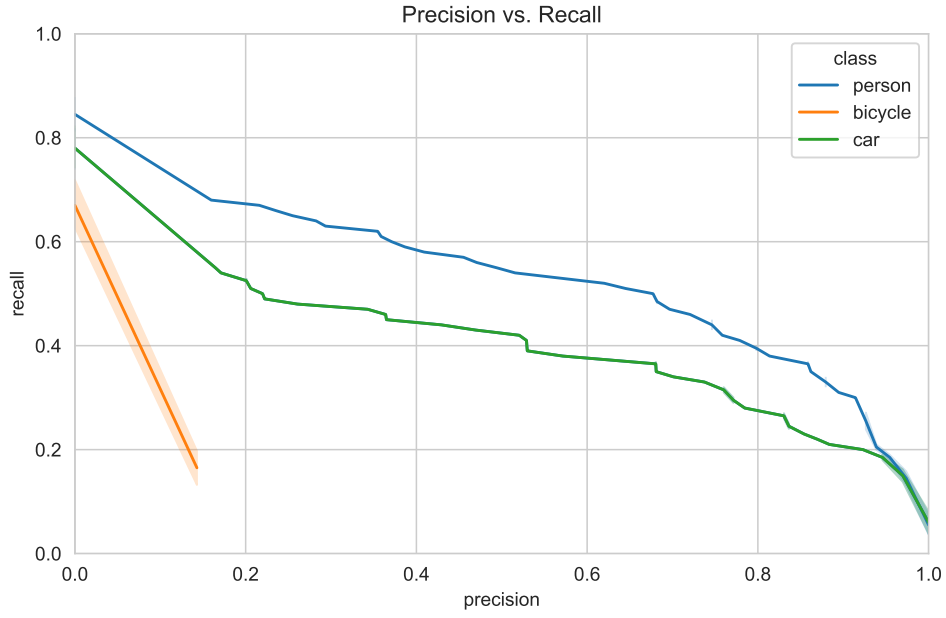
RGB and Lidar

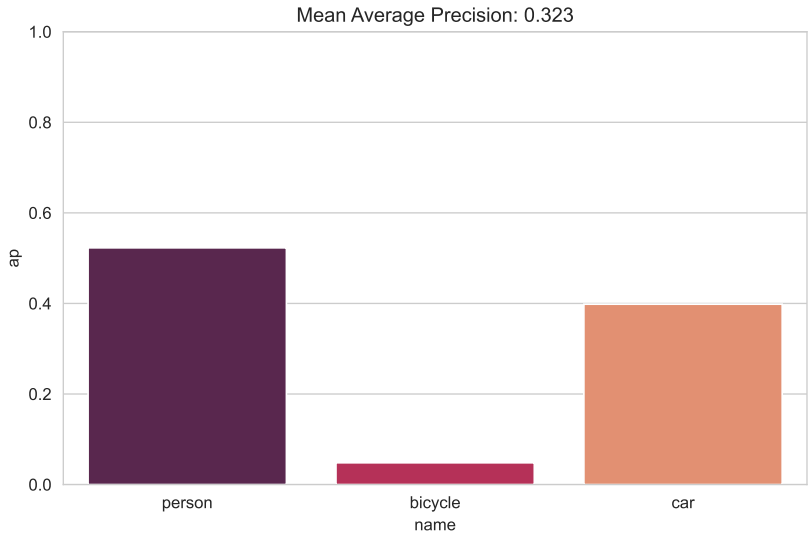




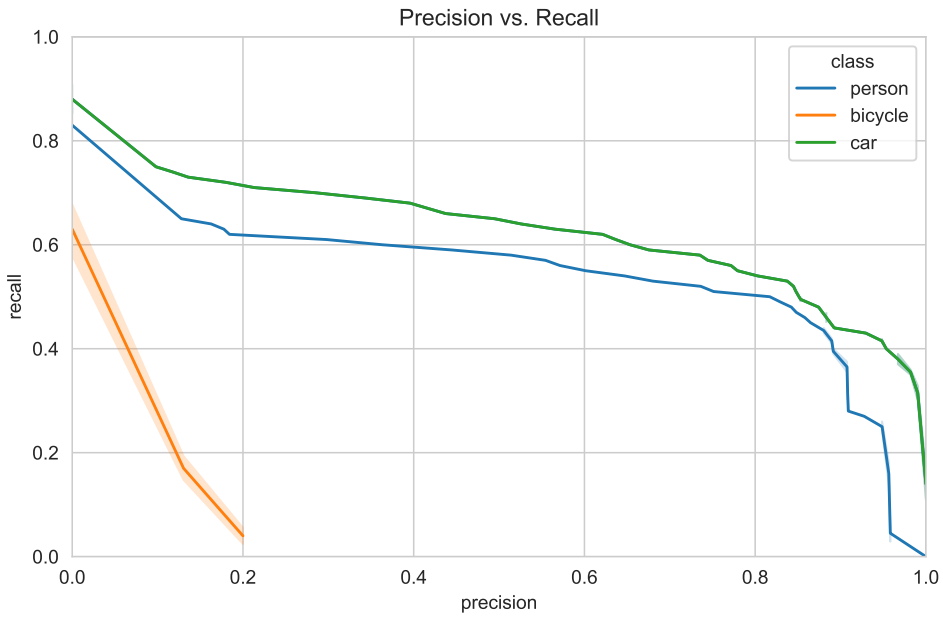
5.6 Snow Night

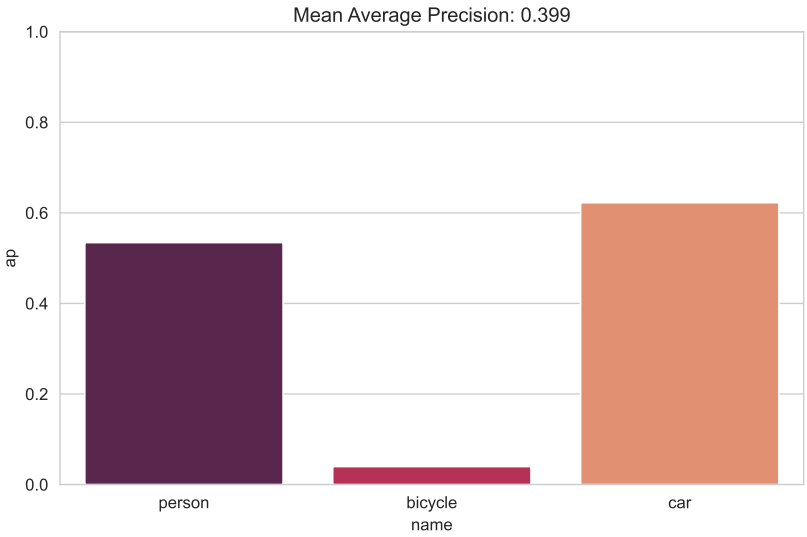
Thermal and Lidar





RGB and Lidar




Mean Average Precision: 0.399
1.0
0.8
0.6
0.4
0.2
0.0
ap
person
bicycle
car
name

Chapter 6

Conclusion and Outlook

6.1 Conclusion

A novel and improved training process for deep neural networks in the thermal domain has been presented which is able to lead to faster training convergence and compensate underrepresented classes.

The main conclusion of this work is that the combination of features from far infrared and lidar images heavily suffers in any weather with water particles in the air. With thermal imaging being mostly completly blinded in dense fog, while with visible light it is still possible to capture the scene. However the results clearly show that thermal imaging in combination with lidar is a strong combination for pedestrian detection during the night on a similar level of actively illuminated scenes which are captures with modern hdr imagers.

In addition to the analyses performed here [21] and [3] discussed that lidar has major problem in adverse weather conditions such as snow and fog with partial failures. Especially modern high resolution lidar sensors fail as shown in this work. Furtgermore in [8] is the Light Transmission in Fog and specifically the influence of wavelength on the extinction coefficient well studied. Wavelengths from 350 nm to 1000 nm had the same behavior in fog, regardless of fog density or type. On the other hand, above 1000 nm, differences may occur. For very dense fogs < 30 m and in particular advection fogs, fog had an impact about 10% higher for wavelengths in the infrared range. [8] These results fit very well with the results observed here in this thesis in that both lidar and thermal imaging have much greater difficulty in fog, rain, and snow.

6.2 Outlook

There are still many fundamentals to uncover in the field of environmental sensing for fully autonomous cars, and no one has yet found the perfect sensor set. Contrary to the original expectations of this work, thermal imaging and lidar have proven to be disadvantageous compared to classic rgb cameras in adverse weather conditions. Especially interesting considering the high price of these two sensors. Recent advances such as hdr imager have made their contribution, which can now reliably capture dark scenes. However, previous work has already shown that thermal imaging can significantly improve pedestrian detection in clear night conditions. Considering the safety that autonomous vehicles need to achieve, thermal imaging may have a place for this situation as well. just no particular advantages in adverse weather.

Bibliography

[1] Michael Aeberhard and Nico Kaempchen. “High-Level Sensor Data Fusion Architecture for Vehicle Surround Environment Perception”. In: (Mar. 2019).

[2] Sarfraz Ahmed et al. “Visual and Thermal Data for Pedestrian and Cyclist Detection”. en. In: *Towards Autonomous Robotic Systems*. Ed. by Kaspar Althoefer, Jelizaveta Konstantinova, and Ketao Zhang. Vol. 11650. Series Title: Lecture Notes in Computer Science. Cham: Springer International Publishing, 2019, pp. 223–234. ISBN: 978-3-030-25331-8 978-3-030-25332-5. DOI: 10.1007/978-3-030-25332-5_20. URL: http://link.springer.com/10.1007/978-3-030-25332-5_20 (visited on 02/09/2021).

[3] Mario Bijelic, Tobias Gruber, and Werner Ritter. “Benchmarking Image Sensors Under Adverse Weather Conditions for Autonomous Driving”. In: *2018 IEEE Intelligent Vehicles Symposium (IV)* (June 2018). arXiv: 1912.03238, pp. 1773–1779. DOI: 10.1109/IVS.2018.8500659. URL: http://arxiv.org/abs/1912.03238 (visited on 02/12/2021).

[4] Mario Bijelic et al. “Seeing Through Fog Without Seeing Fog: Deep Multimodal Sensor Fusion in Unseen Adverse Weather”. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020.

[5] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. “YOLOv4: Optimal Speed and Accuracy of Object Detection”. en. In: (2020), p. 17.

[6] Y. Choi et al. “KAIST Multi-Spectral Day/Night Data Set for Autonomous and Assisted Driving”. In: *IEEE Transactions on Intelligent Transportation Systems* 19.3 (Mar. 2018). Conference Name: IEEE Transactions on Intelligent Transportation Systems, pp. 934–948. ISSN: 1558-0016. DOI: 10.1109/TITS.2018.2791533.

[7] Stüker Dirk. “Heterogene Sensordatenfusion zur robusten Objektverfolgung im automobilen Strassenverkehr”. PhD thesis. 2003.

[8] Pierre Duthon, Michèle Colomb, and Frédéric Bernardin. "Light Transmission in Fog: The Influence of Wavelength on the Extinction Coefficient". en. In: *Applied Sciences* 9.14 (Jan. 2019). Number: 14 Publisher: Multidisciplinary Digital Publishing Institute, p. 2843. DOI: 10.3390/app9142843. URL: https://www.mdpi.com/2076-3417/9/14/2843 (visited on 02/12/2021).

[9] *FLIR ADK™*. URL: https://www.flir.de/products/adk/.

[10] *FLIR Thermal Dataset for ADAS*. 2018. URL: https://www.flir.de/oem/adas/adas-dataset-form/.

[11] Rikke Gade and Thomas Moeslund. "Thermal cameras and applications: A survey". In: *Machine Vision and Applications* 25 (Jan. 2014), pp. 245–262. DOI: 10.1007/s00138-013-0570-5.

[12] Andreas Geiger et al. "Vision meets Robotics: The KITTI Dataset". In: *International Journal of Robotics Research (IJRR)* (2013).

[13] Craig L. Glennie and Derek D. Lichti. "Static Calibration and Analysis of the Velodyne HDL-64E S2 for High Accuracy Mobile Scanning". In: *Remote Sensing* 2 (2010), pp. 1610–1624.

[14] Ibeo Automotive Systems GmbH. *Operating Manual ibeo LUX 2010 Laserscanner*. Ibeo Automotive Systems GmbH, Hamburg, 2010.

[15] *HDL-32E Durable Surround Lidar Sensor*. 2020. URL: https://velodynelidar.com/products/hdl-32e/.

[16] *HDL-64E Durable Surround Lidar Sensor*. 2020. URL: https://velodynelidar.com/products/hdl-64e/.

[17] Meisam Jamshidi Seikavandi, Kamal Nasrollahi, and Thomas B. Moeslund. "Deep car detection by fusing grayscale image and weighted upsampled LiDAR depth". en. In: *Thirteenth International Conference on Machine Vision*. Ed. by Wolfgang Osten, Jianhong Zhou, and Dmitry P. Nikolaev. Rome, Italy: SPIE, Jan. 2021, p. 13. ISBN: 978-1-5106-4040-5 978-1-5106-4041-2. DOI: 10.1117/12.2586908. URL: https://www.spiedigitallibrary.org/conference-proceedings-of-spie/11605/2586908/Deep-car-detection-by-fusing-grayscale-image-and-weighted-upsampled/10.1117/12.2586908.full (visited on 02/09/2021).

[18] Alexander Kirillov et al. "Panoptic Segmentation". en. In: *arXiv:1801.00868 [cs]* (Apr. 2019). arXiv: 1801.00868. URL: http://arxiv.org/abs/1801.00868 (visited on 04/21/2021).

[19] Vladimir V Kniaz et al. “The ThermalGAN Dataset for Thermal Image Synthesis Supplementary Material”. en. In: (), p. 6.

[20] Vladimir V. Kniaz et al. “ThermalGAN: Multimodal Color-to-Thermal Image Translation for Person Re-identification in Multispectral Dataset”. en. In: *Computer Vision – ECCV 2018 Workshops*. Ed. by Laura Leal-Taixé and Stefan Roth. Vol. 11134. Series Title: Lecture Notes in Computer Science. Cham: Springer International Publishing, 2019, pp. 606–624. ISBN: 978-3-030-11023-9 978-3-030-11024-6. DOI: 10.1007/978-3-030-11024-6_46. URL: http://link.springer.com/10.1007/978-3-030-11024-6_46 (visited on 02/12/2021).

[21] You Li et al. “What happens to a ToF LiDAR in fog?” In: *arXiv:2003.06660 [cs, eess]* (June 2020). arXiv: 2003.06660. URL: http://arxiv.org/abs/2003.06660 (visited on 02/12/2021).

[22] Tsung-Yi Lin et al. “Microsoft COCO: Common Objects in Context”. In: *Computer Vision – ECCV 2014*. Ed. by David Fleet et al. Cham: Springer International Publishing, 2014, pp. 740–755. ISBN: 978-3-319-10602-1.

[23] Andreas Pfeuffer and Klaus Dietmayer. “Optimal Sensor Data Fusion Architecture for Object Detection in Adverse Weather Conditions”. In: *arXiv:1807.02323 [cs]* (July 2018). arXiv: 1807.02323. URL: http://arxiv.org/abs/1807.02323 (visited on 02/12/2021).

[24] Joseph Redmon and Ali Farhadi. “YOLOv3: An Incremental Improvement”. en. In: (2018), p. 6.

[25] Micaela Verucchi et al. “A Systematic Assessment of Embedded Neural Networks for Object Detection”. In: *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*. Vol. 1. IEEE. 2020, pp. 937–944.

[26] Hermann Winner et al. *Handbuch Fahrerassistenzsysteme Grundlagen, Komponenten und Systeme für aktive Sicherheit und Komfort*. Springer Fachmedien Wiesbaden GmbH, 2015.

[27] Ravi Yadav et al. *CNN based Color and Thermal Image Fusion for Object Detection in Automated Driving*. July 2020.