HALMSTAD UNIVERSITY

Master of Science in Engineering(MSE)

MASTER THESIS

Semantic Scene Segmentation using RGB-D & LRF fusion

Final report

Harald Lilja

Intelligent systems and digital design

Halmstad University, June 4, 2020–version 4.0

ABSTRACT

In the field of robotics and autonomous vehicles, the use of RGB-D data and LiDAR sensors is a popular practice for applications such as SLAM[14], object classification[19] and scene understanding[5]. This thesis explores the problem of semantic segmentation using deep multimodal fusion of LRF and depth data. Two data set consisting of 1080 and 108 data points from two scenes is created and manually labeled in 2D space and transferred to 1D using a proposed label transfer method utilizing hierarchical clustering. The data set is used to train and validate the suggested method for segmentation using a proposed dual encoder-decoder network based on SalsaNet [1] with gradually fusion in the decoder. Applying the suggested method yielded an improvement in the scenario of an unseen circuit when compared to uni-modal segmentation using depth, RGB, laser, and a naive combination of RGB-D data. A suggestion of feature extraction in the form of PCA or stacked auto-encoders is suggested as a further improvement for this type of fusion. The source code and data set is made publicly available at https://github.com/Anguse/salsa_fusion

ACKNOWLEDGEMENTS

I would like to thank Halmstad University for making this project possible, especially Mikael Hindgren who has been our main contact person with questions regarding resources provided by Halmstad University. I would also like to thank HMS network for sponsoring this project, Eren Erdal Aksoy for being my mentor in this process, and finally Team Apex consisting of Jonathan Jönsson, Felix Stenbäck, Anton Olsson and Felix Rosberg and myself. Without any of the mentioned, this project would not have been possible.

CONTENTS

1 INTRODUCTION 1
- 1.1 Intro 1
- 1.2 Problem formulation 1
- 1.3 Scope 2
  - 1.3.1 Novelty 3
  - 1.3.2 Contribution 3
  - 1.3.3 Questions 3

2 LITTERATURE REVIEW 5
- 2.1 Fusion architectures 5
- 2.2 Encoder-decoder 6
- 2.3 RGB & Depth fusion 7
- 2.4 Depth segmentation 8
- 2.5 LRF & Depth fusion 9

3 METHODOLOGY 11
- 3.1 Overview 11
- 3.2 Implementation 12
  - 3.2.1 Competition format 12
  - 3.2.2 Hardware setup 13
  - 3.2.3 Software platform 13
  - 3.2.4 Data representation 14
  - 3.2.5 Security aspects 14
- 3.3 Fusion Strategy 14
  - 3.3.1 Consideration 14
  - 3.3.2 Network choice 15
  - 3.3.3 Network input 16
  - 3.3.4 Design 16
- 3.4 Data collection 18
  - 3.4.1 Classes 18
  - 3.4.2 Data points 19
  - 3.4.3 Processing 20
  - 3.4.4 Labeling 22
- 3.5 LRF 22
  - 3.5.1 Label transferring 22
  - 3.5.2 Network input 24
- 3.6 Training setup 25

4 RESULTS 27
- 4.1 Performance 27
- 4.2 New circuit 28
- 4.3 Predictions 30
- 4.4 Label transfer 35

5 DISCUSSION 37
- 5.1 Fusion strategy 37

5.2 Unimodal segmentation 38
5.3 Label transfer 38
5.4 Implementation 39
5.5 Hardware 39
6 CONCLUSION 41
6.1 Posed questions 41
6.2 Summary 41
6.3 Future work & improvement 42
A APPENDIX: DISCUSSION ON THE NETWORK ARCHITECTURE 43
BIBLIOGRAPHY 45

LIST OF FIGURES

Figure 1 Fusion strategies; early fusion(a) combines modalities before classification, (b)late fusion classifies using each modality separately before fusion. Intermediate fusion(c) incorporates multi modal fusion within the network layers. 6

Figure 2 Fusenet; A CNN of U-net architecture for semantic segmentation based on RGB-D data, proposed by [8](figure from paper). 8

Figure 3 SalsaNet; an encoder-decoder CNN applied for real-time semantic scene segmentation using point cloud data. Figure from [1] 9

Figure 4 The test bed, figure from [16] 13

Figure 5 Proposed fusion architecture, both laser and depth networks are based on the network SalsaNet[1]. Fusion is conducted by the concatenation of feature maps from the corresponding decoder layer from each modality. 17

Figure 6 Overview of the circuit used for data collection and the resulting map created using laser scans. 18

Figure 7 Representation of data sample from each modality 20

Figure 8 Data collection pipeline; data is collected from the different modalities, an offset is added to laser scans and synchronization is performed. The data is then extracted individually for labeling. 21

Figure 9 Visualization of camera depth estimation and laser scans after added delay. White points represent depth pixels and colored points the LRF scan. 21

Figure 10 Sample label; walls are notated in green, ground in yellow, unknown in red and obstacle in purple. 22

Figure 11 The overlapping LRF and camera FOV 23

Figure 12 Camera label transferred to laser scan 23

Figure 13 The synchronization problem is visible when performing quick turns. The figure above shows the result from a slight turn(left) and a quick turn(right) with walls in blue and obstacles in green. 24

Figure 14 Laser segments generated using hierarchical clustering with a distance threshold of .15 (left) and the adjusted result. This scenario is from the quick turn in figure 13 above. 24
Figure 15 An occupancy grid created from the laser data using ray casting (left) and with applied notations (right). 25
Figure 16 Prediction 1 from each of the networks in the training circuit. 31
Figure 17 Prediction 2 from each of the networks in the training circuit. 32
Figure 18 Prediction 1 from each of the networks in the unseen circuit. 33
Figure 19 Prediction 2 from each of the networks in the unseen circuit. 34
Figure 20 Transferred label results 36

LIST OF TABLES

Table 1 Results 27
Table 2 Results 29

ACRONYMS

Red Green Blue Depth RGB-D

IMU Inertial Measurement Unit

RC Radio Controlled

CNN Convolutional Neural Network

DNN Deep Neural Network

LiDAR Light Detection and Ranging

LRF Laser Rangefinder

ToF Time of Flight

FOV Field of View

ROS Robot Operating System

1 INTRODUCTION

1.1 INTRO

Autonomous vehicles face a wide variety of problems when deciding actions to perform in a driving scenario. These problems are often categorized into three distinct modules; Path-planning, Controlling, and Perception. The perceptive module observes the environment that the car is located in to extract the most relevant information and provide it to the path-planner. The path- planner then uses this data to calculate a strategy for reaching a desired position. The controller's objective is to perform the appropriate actions to achieve the path obtained by the planner. This involves saving the information about the robot's current pose and uncertainties as well as making the vehicle obtain new poses requested by the planner.
In order for the two latter modules to work optimally it is required of the perceptive unit to obtain a sufficient description of the context in which the vehicle is operating. This is especially important in the presence of dynamic obstacles with uncertain movement such as pedestrians or other vehicles.

1.2 PROBLEM FORMULATION

In modern time the usage of RGB-D data in autonomous vehicles and robots has become a widely applied technique. This sensor complements a regular colored image with a gray-scaled intensity map describing the depth information of the observed scene. The technique has been widely applied in fields such as Simultaneous Location and Mapping (SLAM)[14], Object detection[19] and Semantic segmentation[5] to name a few. Active Stereo cameras is one such sensor. They can perceive depth in an environment using a infrared light(IR) pattern which is projected into the scene. The distortion of this pattern is then computed in order to estimate the depth of the image. This technique is very effective in ranges from 0.2-10 meters and produces a dense point cloud of objects within this area. Limitations of this technique resides in computing the depth of a planar homogeneous, transparent or shiny surface.
Another sensor commonly used in robotics is Light detection and ranging (LiDAR) sensors. Apart from stereo cameras, LiDAR sensors traditionally uses time of flight (ToF) technology to measure the range to obstructions. The sensor illuminates the measured area with laser lights and measures the reflected light with a sensor to calculate the

distance to the point. The perks of using this sensor is the high precision in distant objects and its performance in dark areas. It can also make observations in greater FOV in comparison to stereo cameras. In relation to an active stereo cameras, the point density produce by a LiDAR is sparse.
The area of incorporating LRF data with depth is sparsely explored territory. However, we argue that there are relevant information to be retrieved from this combination of modalities. It is well known that LRF sensors have a great accuracy in its range estimations, especially in longer ranges if compared to a camera that estimates depth using binocular disparity. Additionally, the Field-Of-View (FOV) of a LRF is much greater when compared to a depth camera.
One might argue that 3D-LiDAR's could produce an equivalent result in this sense which is partly valid, however the pricing of these type of sensors with similar precision is many times the cost of a LRF-scanner. On this basis there is reason to explore the benefits of this combination of visual information.

1.3 SCOPE

This thesis will focus on fusing 2D LiDAR (LRF) measurements with stereo depth data, leveraging the accuracy and range from the LRF along with the high point density produced by the active stereo camera. The RGB data received from the camera will be used in the labelling process as the reference image. The resulting combined depth information will be used to perform real-time semantic scene segmentation in a racing scenario.

The reason why colored images are excluded from the network input is to create a method that is independent of textures in the scene. That is, we want the method to be applicable in a new environment were the textures may differ from the training scenario but the geocentric principles and spatial information are the same.

Due to the niched area of RC-car racing, data sets with RGB-D and LRF data is not publicly available. Therefore a data set will be created for training of the network.

The method will be applied on a test-bed consisting of a modified RC-car supplied with a computational unit, an active stereo camera, and a LRF. The resulting module will be a part of a fully autonomous system that will be used on the race car in F1tenth[1], a racing competition for autonomous miniature cars.

The resulting pipeline and segmentation procedure will also be evaluated against state-of-the-art semantic segmentation methods utilizing RGB-D data.

1 https://f1tenth.org/

1.3.1 *Novelty*

The use of fused LRF measurements and RGB-D data for the purpose of semantic segmentation.

1.3.2 *Contribution*

- A method for transferring depth labels to LRF scans
- A fusion strategy for segmented LRF scans and depth map

1.3.3 *Questions*

a) Does the LRF data provide additional information that increases the prediction accuracy of segmentation in the scene?

b) Does the exclusion of color data provide a more robust classifier given a scene with unknown/varying texture?

2 LITTERATURE REVIEW

2.1 FUSION ARCHITECTURES

In autonomous vehicles, the use of multiple sensor modalities to achieve a better estimate of the environment is a common principle. This concept is often applied by combining e.g. encoder values, which is a subject to drift due to uncertainties such as tire slippage, with GPS signals that may diverge because of bad signal receptions in order to obtain a better position estimate. The fusion strategy in this area is often comprised of a Kalman filter that computes the combined estimate by observing the estimate of each modality together with corresponding uncertainty. In applications such as this, the estimate is of a low dimensionality which makes the Kalman filter an applicable method. In the field of scene understanding this method is more complex due to the required high dimensionality to describe the scene. By definition a segmented image consists of a series of labels which when combined provide a classification for every single pixel in the image. Dealing with an estimate for every pixel along with its uncertainty in the same manner as in a position estimate would therefore be a computationally expensive task.

Combining multiple modalities for the application of computer vision have three main architectures. Early fusion which is the concatenation of the features from multiple modalities before performing any classification, as implemented by [18]. This procedure may not fully exploit the complementary nature of the modalities and very often lead to large feature spaces to be explored when performing the actual fusion. In contrast, late fusion is performed by combining the classified result of the multiple modalities in a later stage. Each input source is used to train its own machine learning model before performing any fusion. This architecture has been employed frequently in deep multi modal learning during the rise of ensemble classifiers and can be found implemented in [23],[29],[11]. The third architecture is referred to as intermediate fusion. This architecture incorporates fusion strategies within the network layers of the classification model. The majority of work in deep multi modal fusion adopts this architecture. Advantages of this flexible approach is described in [12]. In this work the author introduces a slow fusion model that fuses representations of video streams gradually across the layers during training. This approach is shown to achieve better results for a large-

scale video classification problem in comparison with early and late fusion.

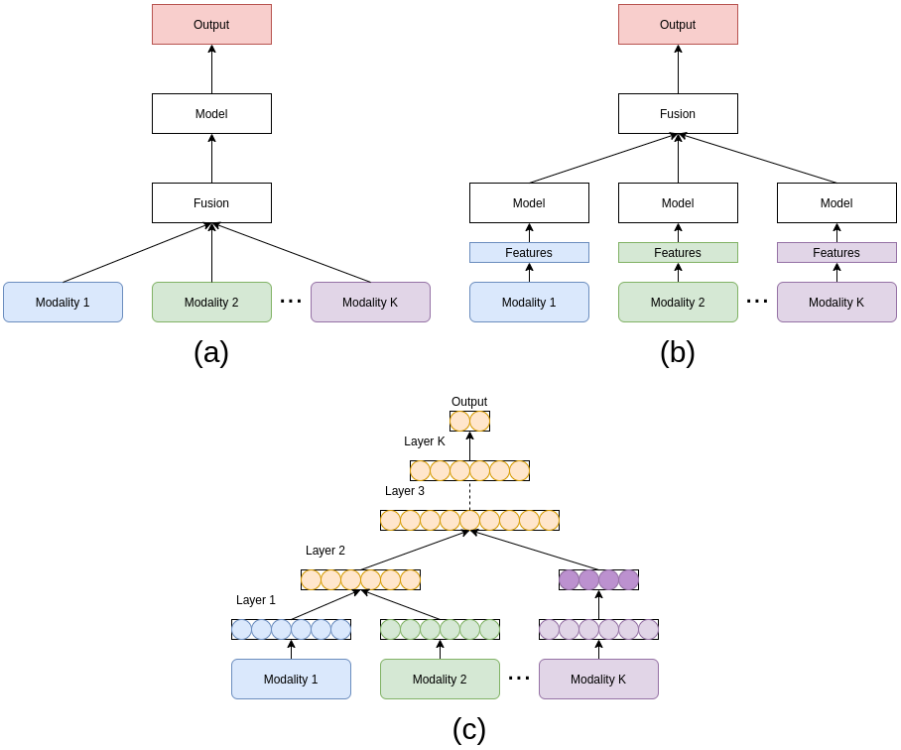


Figure 1: Fusion strategies; early fusion(a) combines modalities before classification, (b)late fusion classifies using each modality separately before fusion. Intermediate fusion(c) incorporates multi modal fusion within the network layers.

It is easy to state that intermediate fusion can be a better approach to a deep multi modal fusion problem, however due to the flexible nature of this architecture it is important to decide how the network should be designed for the applied problem to achieve its benefits. The choice of which modality to fuse and at which depth is usually done based on intuition. The success of different approaches in this decision is based upon the nature of the fused modalities. Additionally, the decision of in what layer and how many times to perform the fusion is also a designing parameter that needs to be considered. This boils down to the problem of network design.

2.2 ENCODER-DECODER

The encoder-decoder architecture is popular in the task of semantic segmentation. The encoder part takes an input vector and gradually comprises feature maps in each layer, encoding the original input as the network grow deeper. The decoder does the opposite, taking the feature map comprised by the encoder and up sampling it to give an

approximation of the target output.
In the training phase, the encoder and decoder are trained together using a loss function based on the delta between the prediction and target. The optimizer will train each part of the network to achieve the lowest delta.
By applying this method, the encoder is therefore trained to generate a feature vector that comprises the most useful features from the image so that the decoder can predict the target image as accurately as possible. A successful implementation of this type of network is [2] where the authors employ an encoder-decoder architecture for semantic segmentation on RGB images. The network uses a up-sampling strategy by extracting pooling indices from the corresponding encoder layer which removes the need for a trained up-sampling layer.

2.3 RGB & DEPTH FUSION

In [31] the authors use RGB and depth information from LiDAR point clouds for semantic segmentation by first making classification with each modality separately and then performing sensor fusion using a fusion classifier. For a feature space $\mathbb{R}^N$, each uni modal classifier P are evaluated on segments which are covered by a single sensor, producing a set of labels Δ^L.

$$P_{img} : \mathbb{R}^{N_{img}} \rightarrow \Delta^L,\ P_{pc} : \mathbb{R}^{N_{pc}} \rightarrow \Delta^L \tag{1}$$

A strategy of early fusion would operate on a feature space of length N_{pc+img}. Comprised of data received from both modalities.

$$P_{earlyfusion} : \mathbb{R}^{N_{pc+img}} \rightarrow \Delta^L \tag{2}$$

In the applied late fusion architecture, fusion classification is applied on all overlapping segments of each modality.

$$P_{latefusion} : \Delta^{2L} \rightarrow \Delta^L \tag{3}$$

By applying this strategy, the learning problem is significantly deduced. An advantage of early fusion is the learning of potentially more expressive information of each modality. However, the learning problem becomes much more demanding in terms of computation.

An example of intermediate fusion applied on a similar problem is described in [8]. The authors discuss the popular use of HHA-encoding[7] that is commonly applied to depth information in order to obtain more discriminative information from the channel. This representation consists of three channels: horizontal disparity, height and angle between normals and the gravity based on the estimated ground floor. HHA encoding significantly improves the accuracy of semantic segmentation but requires high computational cost. The

authors instead argue that this representation is superficial and inefficient and instead proposes the encode-decoder network FuseNet which extracts and fuses depth information with RGB gradually in the encoder part of the network. Instead of preprocessing a single channel depth image into three channels, FuseNet learns high dimensional features from depth end-to-end as the network grows deeper.

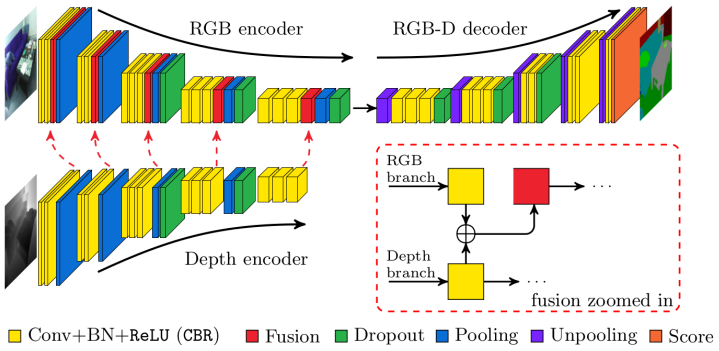


Figure 2: Fusenet; A CNN of U-net architecture for semantic segmentation based on RGB-D data, proposed by [8](figure from paper).

2.4 DEPTH SEGMENTATION

In [1], the authors employ a encoder-decoder CNN, named SalsaNet for real-time semantic notation using only 3D-LiDAR point clouds. The network is trained and evaluated on the KITTI data set [6] which is a publicly available data set consisting of labelled road scenes. The network is trained to distinguish road, vehicles and background. Since the point cloud data in this data set is unlabelled, an auto labeling process for LiDAR data is applied using MultiNet [26], which is a public network pre-trained on the KITTI data set, and Mask R-CNN[10]. Using bounding boxes of vehicles provided in the data set, annotations can then be made in the RGB image and transferred to point cloud domain. Further, a Bird-Eye-View(BOV) of the scene is generated producing a 2D-map with the size of 256x64 cells each containing mean and max elevation, intensity, and number of projected points. The resulting input to the network is a 256x64x4 BEV map.
The proposed network is shown in figure 3 and is comprised of an encoder part of stacked ResNet[9] blocks each of which is followed by dropout and pooling layers. Each convolutional layer has a default kernel size of 3. The number of feature channels are respectively 32,64,128,256 and 256. Up-sampling the feature map, each deconvolutional layer in the decoder part is element-wise added to the corresponding layer in the encoder part via skip connections. After feature addition in the decoder, a stack of convolutional layers are introduced to capture more precise spatial cues to be further propagated to the

higher layers. The following layer applies 1x1 convolution to achieve 3 channels corresponding to the semantics in the scene. These are fed to a soft-max classifier to obtain pixel-wise classification.
Evaluating this network, the authors achieve results better than state-of-the-art methods[27],[28],[22]. The source code is made publicly available at [1].

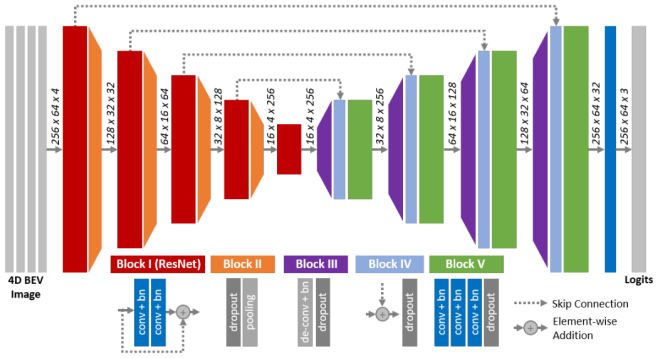


Figure 3: SalsaNet; an encoder-decoder CNN applied for real-time semantic scene segmentation using point cloud data. Figure from [1]

2.5 LRF & DEPTH FUSION

In [21] the use of LRF data with disparity information perceived by a passive stereo camera is combined in the problem of human tracking from a mobile robot. The proposed method is based on low-level sensor fusion. The approach leverages the robustness of a human detector based on depth data with the responsiveness of LRF based detection using a Kalman filter applied in a late fusion fashion. In the depth based detection method, the LRF scan is also incorporated by introducing a trimming method based on the difference in depth estimation between the depth perceived by the camera and laser scanner. Given a set of stereo depth estimation $\mathbf{v} = [v_1, v_2, ..., v_n]$ and corresponding laser scans $l = [l_1, l_2, ..., l_n]$ trimming is performed by:

$$v_i = \begin{cases} l_i, & \text{if } v_i > l_i \\ v_i, & \text{otherwise} \end{cases} \tag{4}$$

Therefore after this trimming achieving:

$$v_i \leqslant l_i, i \in [1, 2, ...n] \tag{5}$$

1 https://gitlab.com/aksoyeren/salsanet.git

The reason for applying this method is to remove false depth estimations perceived by the stereo camera, instead relying on the more accurate LRF.

3 METHODOLOGY

3.1 OVERVIEW

This thesis aims to solve the problem of semantic segmentation in a real-time scenario utilizing stereo disparity and range scans from a LRF. The solution was to be implemented on a miniature race car that would have entered the competition F1Tenth[16]. Due to the current situation of covid-19 pandemic the competition was cancelled. However this solution will instead be used as a reference perception method for possible entries in the future made by Halmstad University. In that regard, the scope of this thesis will not change in terms of implementation.
Since the area that the implementation will be applied on is very specific, it is necessary to generate and label data collected using the built setup in an environment similar to that of the competition circuit. However, since the actual competition environment is not available there is a need for flexibility in the implementation. One issue is that the information of textures in the scene is unknown which could make information of RGB nature misleading. On this basis we chose to exclude colored images in our classification strategy. With this said, as posed in 1.3.3 a), evaluation of the success in the decision of excluding color should be analyzed.
Another implementation issue faced is the problem of classifying opponent vehicles since only one test bed is built and available in this proposition. This is an important issue to be resolved since information of vehicles in the vicinity is a piece of information that is significant in the decision making of the local path-planning and controlling module.

The method of sensor fusion will be evaluated against the performance of networks based on unimodal data of every available modality, that is:

- RGB
- Depth
- LRF

Depending on the success of the network further evaluation against state-of-the art strategies utilizing sensor data within the constraints of our hardware setup will be conducted. Since, to the best of our knowledge, no existing solutions exists utilizing all three modalities the focused area is on the two first mentioned.

3.2 IMPLEMENTATION

Since the solution proposed in this thesis will be applied in a larger system with specific constraints, this information needs to be considered before designing the details of the system. The context that will be considered when performing the implementation can be comprised as:

- Rules of the competition
- Hardware setup
- Circuit design
- Obstacles in the scene
- Additional modules in the autonomous system
- Security aspects

3.2.1 *Competition format*

The F1tenth competition is presented by the authors as part competition and part test bed for researchers around the world to apply studies upon. The main idea is to make research on autonomous vehicles more accessible in relation to competitions such as DARPA grand challenge [3] which is more expensive and hazardous due to the use of real sized vehicles.
The competition format consists of a circuit made up of arbitrary wall segments in which cars about 1/10 the size of a regular race car, compete against each other in two different main stages. A time trial format where the cars each individually occupy the course to attempt to complete as many laps as possible in a given time(typically 3-5 minutes). In this format, both speed and consistency are rewarded while crashing results in penalty. The second race format is head-to-head racing. In order for a car to enter this part of the competition, it must first demonstrate the ability to avoid obstacles placed within the scene. These obstacles can consist of cardboard boxes or foam that will be moved to different places in the course. If the car passes this test it is able to enter the head-to-head racing class. In the actual competitive part of this stage two cars will race against each other under some circumstances which are not mentioned in the public rule description.

In addition to the different competitive stages there are two separate vehicle classes which will compete against each other. One restricted class in which the hardware is limited according to a list of components and an open class where the only restriction is the size of the

car and the limitation of electric powered motors. The car developed in this thesis will enter the first mentioned class.

3.2.2 *Hardware setup*

The main parts of the setup on which all experiments will be conducted is described in the following list:

- Intel realsense D435i
- Hokuyo 10LX
- NVIDIA Jetson TX2
- VESC6 speed controller
- Orbitty carrier
- Traxxas Velineon 3500 brushless motor

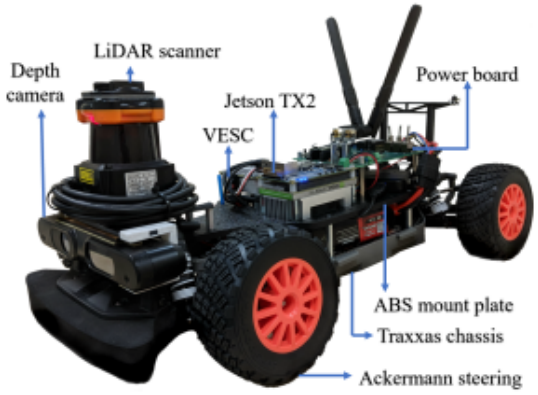


Figure 4: The test bed, figure from [16]

3.2.3 *Software platform*

An autonomous vehicles require a framework for intercommunication between modules in the system. For example, the sharing of read sensor data. A framework suitable for this is Robot Operating System (ROS). This framework utilizes a publish/subscribe architecture where different programs (nodes) publish and/or subscribes to data from a centralized core using topics. This framework is widely used in the field of robotics and therefore include many useful libraries for tasks such as localization, navigation and computer vision.

3.2.4 *Data representation*

How to embed the scene information for use in the system is important to consider. Since the path-planner needs to consider many possible paths to take in order to make the best decision, it is important to keep this information as compressed as possible. A lightweight interpretation would be to have the different segments expressed as a label identifier along with the distance to the closest point of the segment and the angle from the front of the vehicle. Another approach is to present the segments as point clouds and let the path-planner be responsible for compressing the information as pleased. Since the core functionality of a perception module is to supply the system with as detailed scene information as possible, this decision will be left for each subscribing module.

3.2.5 *Security aspects*

Issues concerning security in this implementation mostly consists of the dangers in false measurements from the sensors. If the LRF fails to obtain information of a wall for example, this would most likely result in the car crashing. An approach of handling this is to implement an emergency stop if the LRF fails to obtain any laser scans. As argued in 3.2.4, this is also not really a perception related issue since the decision of actions to perform is done by the controller module.

3.3 FUSION STRATEGY

3.3.1 *Consideration*

The two modalities utilized in this thesis is of different nature in terms of data coverage and representation. Although they share measured information, the information gain from the LRF provides additional information in terms of accuracy at greater range and a much larger FOV. That said, the overlapping area perceived by the camera is highly relatable to the LRF.
One approach to the problem could be to simply adjust the depth points in the vicinity of the area covered by the scanner in an early fusion fashion. This could provide a greater accuracy in terms of spatial position of obstacles in the scene, however would not provide any additional information in the question of the type of object being observed. Additionally, this would put great stress on the displacement relationship between the mounted sensors. This is fragile, especially since the vehicle has shockers which lead to dynamic change in pitch orientation when performing quick turns, acceleration, or breaking. Not to mention the obvious complication of perfect data synchronization.

Fusing the data after performing classification on each modality would be a preferred choice on these terms. The problem here is the different dimensionalities of the output from each classifier. The problem is to notate every pixel of an image but the scanner can only observe a small segment of this image. How can we solve the problem of reshaping the entire prediction made from the depth input based on this segment? This exhaustive procedure must also be able to perform with real-time inference.
As referred to in 2.3 the authors of [8] presents a strategy for fusing different modalities on a network level. The approach taken here is dependent on highly resembling input data; RGB and depth images obtained using one camera. This assures the same dimension resulting in similar feature maps which can be combined to provide more robust results. The fusion strategy of this paper is also exhaustive in terms of element-wise multiplication in multiple layers of the network encoder. In our belief this task would be to heavy computationally when pursuing real-time inference.
In [15] the authors argue that with weakly correlated modalities, a mixture is rarely beneficial until the later stages of the process. Like earlier mentioned there is resemblance in the modalities used in this thesis although the need for alternative representation is most likely to result in a significant divergence at glance. The task will be to find deeper correlations between the two representations.
On this basis the concretization of a network based fusion strategy will be pursued.

3.3.2 *Network choice*

As mentioned in 2.2 the use of a encoder-decoder structured network is very beneficial for image segmentation[2][4] which makes it a more or less given choice for the task at hand. For real-time inference it is also desirable that the network is of minimalistic proportions to be able to keep up with such constraints. Due to time constraints, there exists no room for trial and error in this phase but the network employed shall be able to conduct on both modalities ideally.
Earlier mentioned SalsaNet[1] is employed on data from a 3D LiDAR that is comprised to a topology view grid containing spatial and relative information from points in each grid cell. Additionally it performs with low inference, up to 160 Hz on a 256x64 image containing 4 channels stated by the authors. Segnet[2] is another consideration. This network is conducted on RGB images with an inference of 20 Hz using an image resolution of 360x480 on Nvidia Titan GPU.
Due to the appealing resemblance of application the choice is made to employ SalsaNet as the applied network. One may argue that SegNet has achieved greater overall success, however the deployment of

this network on the inferior GPU utilized on the vehicle could lead to an inference below 15 Hz which is undesirable.

3.3.3 *Network input*

As reasoned in 3.3, the network is to find deeper correlations between the modality representations.

The data is initially comprised of a 640x480 depth image and a vector of 169 laser point scans. In order to combine these modalities it is necessary to find a shared representation. Since the network is a CNN with 2D convolution this is the preferrable input space. This means that there is a need to create a 2D representation of the laser scan. Important to consider here is the size of dimensions. Obviously, too large dimensionality will cause loss in the networks inference time. It is also desired to obtain synchronization between the inputs in terms of dimensionality.

3.3.4 *Design*

Using SalsaNet as the starting point of the network, the idea is to extend this implementation with an additional pipeline for processing laser data. Then the question comes down to how the fusion step is to be conducted. One way of doing this is to simply concatenate the feature maps from each modality. This is a simple solution but may yield improvement over single modality prediction. Mentioned in [20] improvement with this type of fusion can be achieved by dimensionality reduction with the use of principal component analysis (PCA) or stacked autoencoders after the concatenation has been conducted. This is therefore the preferred design choice for the fusion step. An implementation issue with this design is that the employed tensorflow version does not support PCA layers. Due to the time constraints there is unfortunately not enough room to implement such a layer or yet switch to another backend version. For this reason, the initial design will rely on fusion by concatenation of feature maps without any additional dimensionality reduction.

Next consideration is where the fusion is to be performed. As argued in 3.3 fusion of earlier nature is preferred when the modalities is greatly correlated. In our case the 2D grid generated from the laser scanner will contain significant visual differences arguing against fusion at the early stages of the network. Instead a deeper fusion strategy will be applied in the decoder part of the network.

The laser network will be pre-trained before incorporation into the fusion network to perceive the most describing features in terms of classifying the laser data. These weights will be used initially in the fusion network but left trainable as further updating the weights can be relevant for describing the features that are more discriminant for

the final task of full scene segmentation. For simplicity a minimized version of SalsaNet is used as the base for laser segmentation as well as the full network configured with the corresponding resolution for the input depth image.

The SalsaNet encoder is constructed with a series of ResNet[25] blocks. Each ResNet block is followed by dropout and pooling layers except for in the bottleneck. The decoder has a series of deconvolutional layers to upsample the feature maps, each of which is elementwise added with the corresponding layer in the encoder using skip connections. After feature addition, a stack of convolutional layers are conducted to capture more precise spatial cues. The next layer applies a 1x1 convolution to obtain 5 channels corresponding to the classes in section 3.4.1. The output feature map is fed to a soft-max classifier to obtain pixel-wise classification. The full network is illustrated in figure 5 below.

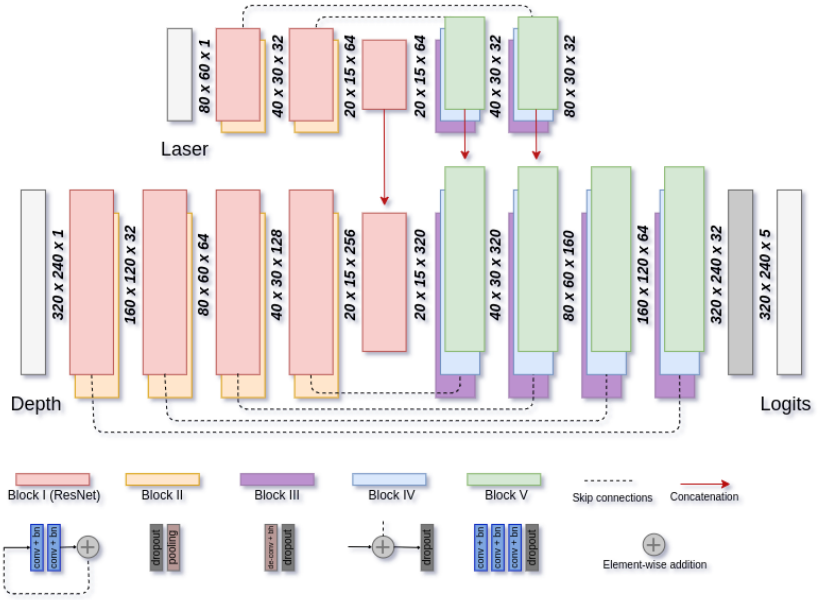


Figure 5: Proposed fusion architecture, both laser and depth networks are based on the network SalsaNet[1]. Fusion is conducted by the concatenation of feature maps from the corresponding decoder layer from each modality.

To adjust for class imbalance in the dataset, the soft-max cross entropy loss is updated with a smoothed frequency of each class. The weight is applied as α_i, resulting in the following expression:

$$\mathcal{L}(y, \hat{y}) = -\sum_{i}^{n} \alpha_i p(y_i) \log(p(\hat{y_i})), \qquad \alpha_i = 1/\sqrt{f_i} \tag{6}$$

Where f_i denotes the frequency of the class i. y_i and $\hat{y_i}$ express the true and predicted labels.

3.4 DATA COLLECTION

The first task before any segmentation can be performed is the collection of data. For this a circuit is built in the same manner as described in the competition rules. The scene which will be explored will consist of walls covering both sides of the vehicles and obstacles placed in arbitrary locations throughout the circuit.

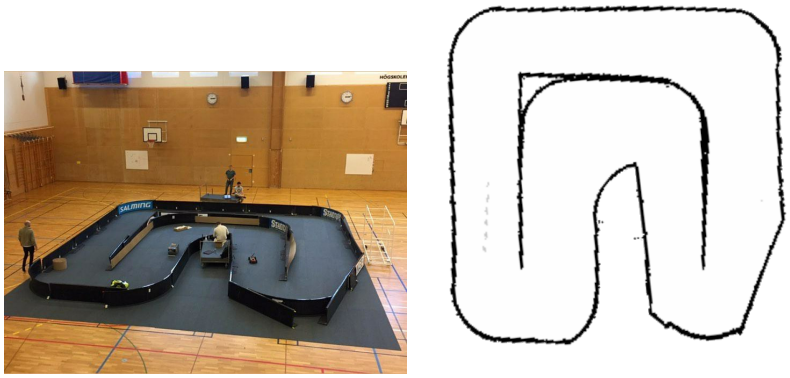


Figure 6: Overview of the circuit used for data collection and the resulting map created using laser scans.

3.4.1 *Classes*

In the scene description there will be as mentioned, 5 types of segments;

- Walls
- Ground
- Obstacle
- Car
- Unknown

In the laser data, the unknown class notates areas where no laser point is present. Using ray casting, it is still possible to determine ground points by projecting a beam towards a point. The ground can be established to be on this path due to the free space and given that the operating area is a confined space. Further explanation of this is presented in 3.5.2. In the depth data, unknown is considered everything that is outside of the circuit.

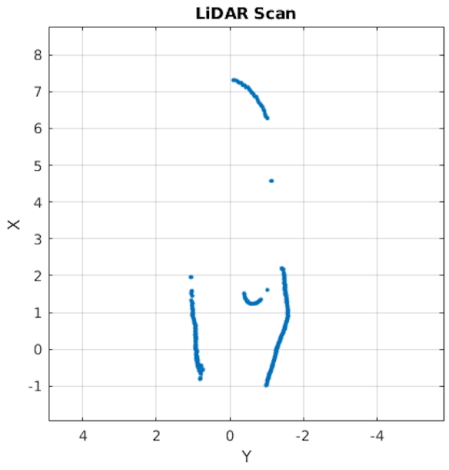
3.4.2 *Data points*

When the data is collected it is necessary to have both modalities synchronized as we want to find correlations between the different types of sensor data. We want to be able to label one modality and have this label be applicable in the other. This covers both the temporal and spatial relationship. To handle the transformation between the mounting points we apply a displacement to the sensors in relation to the middle point of the car.
The intel realsense d435i camera is supplied with a fish-eye lens which makes it possible for the camera to provide a wider field of view in the depth channel. This is the default perceptive mode of depth on the camera, therefore we add the additional topic of depth_aligned which corresponds to the depth when transformed to match the frame of the colored image due to the same reason as argued above.
To synchronize the data the rosbag tools from the ROS framework is applied. This allows for recording of topics being published in ROS creating a bag file containing all data published during the time of recording. This can later be used for playback within ROS in order to visualize or process the data. The recorded topics are as follows:

- /scan - LRF scan
- /camera/color/image_raw - RGB image
- /camera/depth/image_rect_raw - Stereo depth
- /camera/aligned_depth_to_color/image_raw - Stereo depth aligned with RGB image

During the collecting phase, a data set of 1080 data points from respective modality have been collected. A data point consisting of an image from each modality can be viewed in figure 7

(a) RGB. (b) Aligned depth.



(c) LRF.

Figure 7: Representation of data sample from each modality.

3.4.3 *Processing*

Analyzing the data after recording there existed a problem with synchronization between the camera and LRF scans. The recorded camera data was delayed in relation to the laser scans. Applying a static delay to the LRF of 350 milliseconds provided a satisfactory result where the depth aligned with the laser scans. From the ROS visualization tool rviz, the result can be viewed in figure 9. It should be noted that this is a manual configuration made to achieve a better synchronization. Although the result is aligned, it is not perfectly harmonized.
Additionally, the data from the sensors is published using different frequencies resulting in an unbalanced amount of measurements. The LRF has a scan rate of 40 Hz while the frame rate of the camera is 30 fps. To adjust for this a synchronization filter was implemented. This filter considers several topics from within ROS and is implemented to publish a new synchronized topic when all topics publish their data within a certain time interval. This time interval was set to

10ms. With this method we achieve the same amount of data points from each sensor. An overview of the data processing flow can be viewed in figure 8.

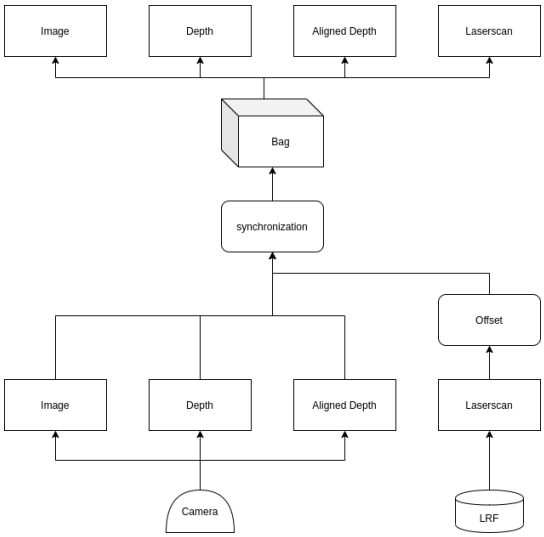


Figure 8: Data collection pipeline; data is collected from the different modalities, an offset is added to laser scans and synchronization is performed. The data is then extracted individually for labeling.

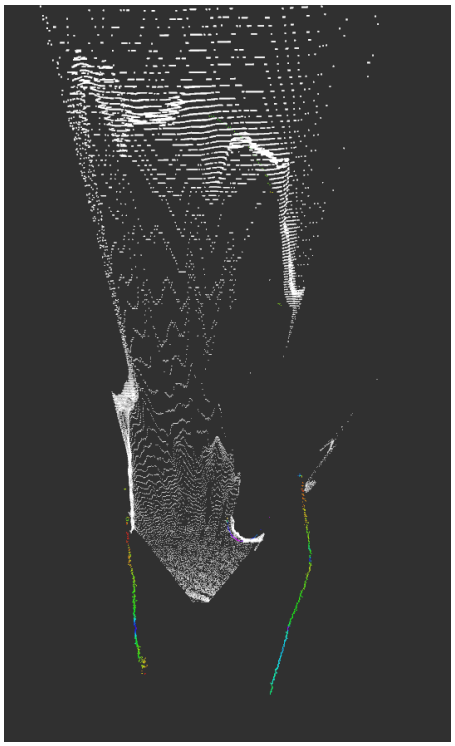
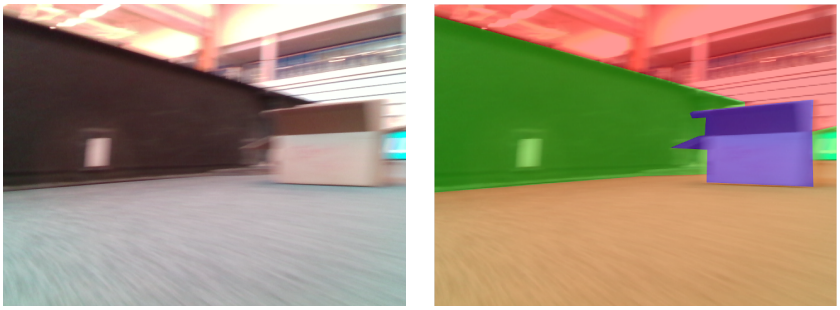


Figure 9: Visualization of camera depth estimation and laser scans after added delay. White points represent depth pixels and colored points the LRF scan.

3.4.4 *Labeling*

For the labelling procedure the RGB images are utilized since they are aligned with the depth images and are more suitable when distinguishing segments. This is done with the classes referred to in 3.4.1. The labels are embedded as an image with a value corresponding to its label. That is the segment class which the corresponding pixel belongs to. An example of a labelled image can be viewed in figure 10 below.



(a) Original data (b) Labeled data

Figure 10: Sample label; walls are notated in green, ground in yellow, unknown in red and obstacle in purple.

3.5 LRF

3.5.1 *Label transferring*

In order to avoid additional labelling, it is desirable to be able to apply the same notations from depth/RGB to the laser scans. To achieve this we propose the following method: Knowing the horizontal FOV of the LRF and camera, we first isolate the overlapping laser segment **L** by removing scans with angles outside this boundary. The camera FOV is 42.5° or $\pm$21.25°, which when isolated in the laser scan can be viewed in figure 11a. By clamping the laser scan with the angular range of the camera we achieve a 169x2 vector consisting of the x and y value of every point. Since the image width is 240 pixels the row is down-sampled using nearest-neighbor interpolation and truncation to fit the laser scan dimension.
In some scans, there are invalid points that occur due to the reflection of the walls. These are notated by an invalid intensity value. These are excluded in the scan as well as the label.
By retrieving the notated pixels in the row corresponding to the height of the LRF we achieve the labels which match the laser scan. Applying the label on the scan we achieve the result shown in figure 12a. Figures 10, 11a and 12a all belong to the same data sample.

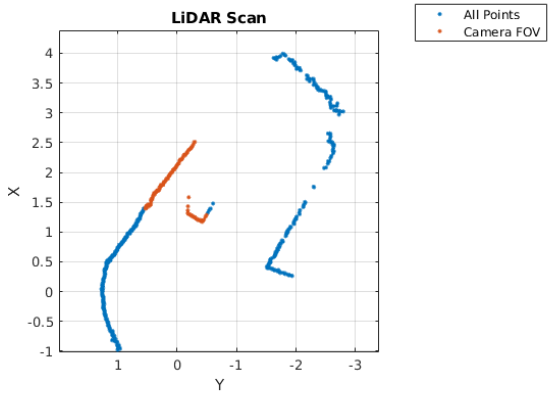


Figure 11: The overlapping LRF and camera FOV

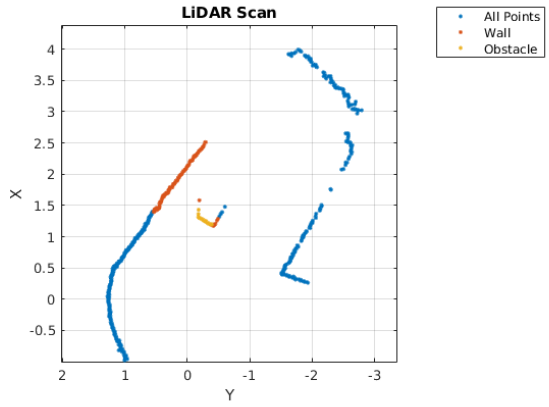
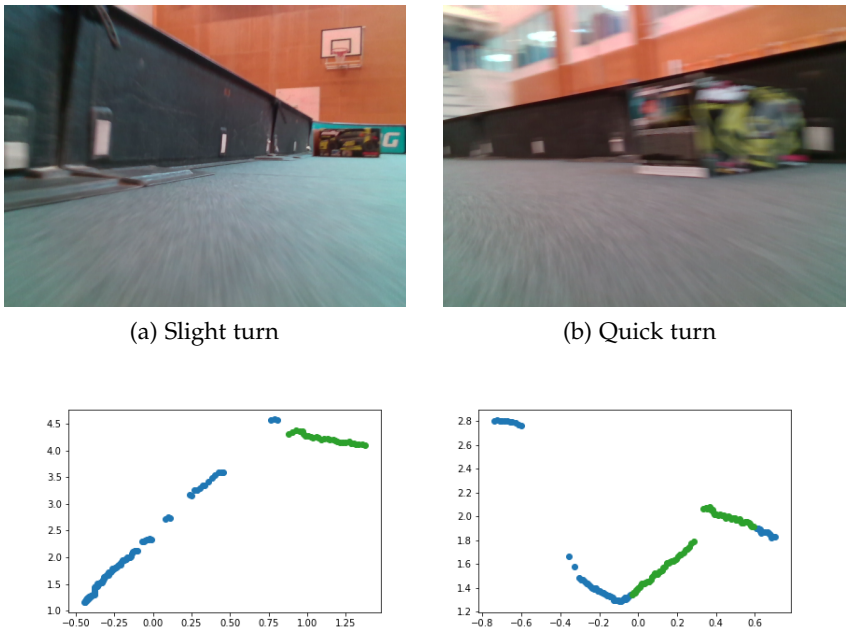


Figure 12: Camera label transferred to laser scan

Simply overlapping the label on the scan gave an excellent result when the car is in a static position or slight turn. However, when the car is making fast turns, the synchronization proved to be inadequate. In figure 13 the scan and the applied label can be viewed in a state where the car is in a slow and quick turn.

To address this issue we need a way of determining the best match from the applied label. To do this we need to separate the different possible segments from the raw laser scan and find the overlapping points from the applied label. We chose to apply hierarchical clustering on the raw laser scan since this is a preferred method when the number of clusters in the data is unknown. A distance threshold of .15 is found satisfactory in this process. An example of this can be viewed in 14. We then correct the label by finding the cluster which contain most of the points belonging to that label class. This is done for the labels obstacle and car followed by filling of the remaining segments with the wall class.



(a) Slight turn (b) Quick turn

Figure 13: The synchronization problem is visible when performing quick turns. The figure above shows the result from a slight turn(left) and a quick turn(right) with walls in blue and obstacles in green.

3.5.2 *Network input*

Since the network input expects a 2D image as input there is need to generate such a representation from the vectorized laser points. Doing so, a confined area must be established so that scans of all ranges can fit within the grid. The size must also be scalable with the depth image with size 320x240. A grid size of 80x60 is established, filling both of these criterias.
In order to create this representation ray casting is conducted by applying Bresenham's line algorithm [17] with the coordinates of each valid point. This method generate an occupancy grid based on the

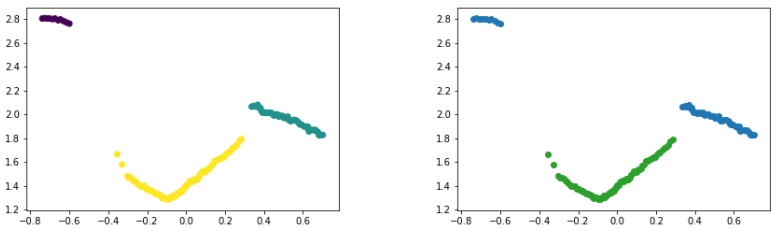


Figure 14: Laser segments generated using hierarchical clustering with a distance threshold of .15 (left) and the adjusted result. This scenario is from the quick turn in figure 13 above.

position of the LRF by projecting a beam towards the observed point. Doing so it can be established that the residing area between the point and the LRF is unoccupied and thus creating a 2D occupancy grid from the points. This is very useful in our case since it provides a notation of ground areas, free of charge.

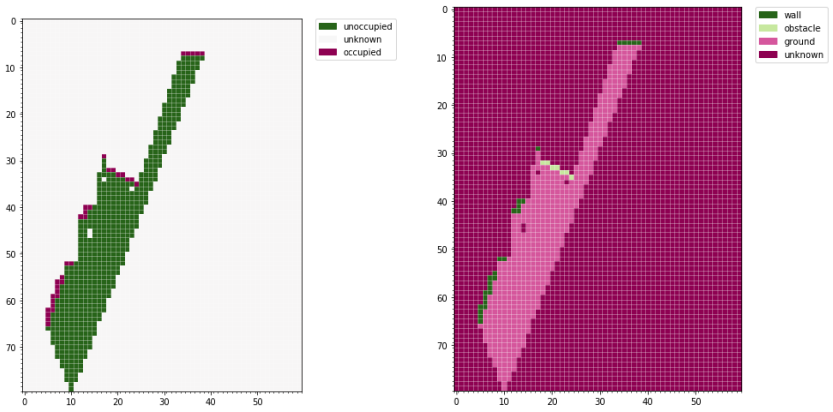


Figure 15: An occupancy grid created from the laser data using ray casting (left) and with applied notations (right).

3.6 TRAINING SETUP

The data retrieved during the collection phase is unbalanced with a lower frequency of the class obstacle and car. To adjust for this we update the softmax cross-entropy loss function with a smoothed frequency of every class as expressed in equation 6. To be able to train the network efficiently for parameter tuning, the model is applied on a Nvidia RTX 2080 GPU. The data is split into train, test, and validation sets using 70,15,15 splits. Data augmentation is conducted on every data point in the form of rotation, gaussian noise, and horizontal flipping. For rotation a degree of -5 to 5 is applied randomly, a variance of 10 is used for the applied noise. In the test set, augmentation is excluded for a more accurate estimation of the performance on the final application.
The fusion network is trained along with plain SalsaNet networks trained on depth, RGB and RGB-D data. The network for processing laser data is also evaluated in the experiment to consider how well this modality alone can classify the scene. The depth network is considered the baseline since the interesting aspect is whether any performance gain can be seen from introducing LRF data in relation to utilizing only depth data.

The fusion network is trained and evaluated both with and without a pre-trained laser module using the laser labels. In the pre-trained model, the laser network is first trained on laser labels in order to

conclude the most significant features from the scan readings. This model is then applied as a starting point for the fusion network. A batch size of 8 images is applied along with a learning rate of 0.01 which decayed by 0.06 after 20 000 iterations. The Adam optimizer [13] is used in the learning process. The dropout probability is set to .5 and the network is trained for 100 epochs except for the laser network which is trained for 50 epochs. The network is evaluated on the average of 100 predictions on the unseen test data.
The results are evaluated based on intersection over union (IoU), precision and recall of every class. Additionally the overall inference is considered. The evaluation metrics for the class i can be expressed as:

$$P_i = \frac{|\hat{Y}_i \cap Y_i|}{|\hat{Y}_i|}, \quad IoU_i = \frac{|\hat{Y}_i \cap Y_i|}{|\hat{Y}_i \cup Y_i|}, \quad R_i = \frac{|\hat{Y}_i \cap Y_i|}{|Y_i|} \tag{7}$$

Where Y_i denotes the ground true area of class i and $\hat{Y}_i$ the predicted area.
The source code in this procedure is made publicly available at [1].

1 https://github.com/Anguse/salsa_fusion

4 RESULTS

4.1 PERFORMANCE

	FUS1	FUS2	DEPTH	LASER	RGB	RGB-D
Iou (%)						
Wall	91.07	90.20	90.94	95.22	96.02	**96.27**
Ground	97.05	96.35	97.07	98.42	**98.62**	98.58
Obstacle	37.80	37.81	36.11	43.29	52.22	**53.78**
Car	8.41	12.63	13.28	8.70	21.15	**22.50**
Unknown	95.73	96.06	95.84	**99.93**	98.18	98.31
Average	66.02	66.58	66.65	69.11	73.24	**73.89**
Precision (%)						
Wall	97.29	94.12	96.78	**99.39**	98.91	98.88
Ground	97.45	98.60	97.64	98.42	98.86	**98.88**
Obstacle	45.83	44.05	41.53	45.99	57.23	**58.32**
Car	11.05	15.64	16.70	9.52	23.91	**25.98**
Unknown	97.73	98.34	98.31	**1.00**	99.09	99.20
Average	69.77	70.15	70.19	70.66	75.60	**76.25**
Recall (%)						
Wall	93.45	95.61	93.79	95.77	97.05	**97.33**
Ground	99.58	97.66	99.40	**1.00**	99.75	99.69
Obstacle	65.91	69.51	69.76	71.26	84.09	**85.78**
Car	22.68	27.83	28.03	9.61	**57.23**	52.54
Unknown	97.91	97.65	97.44	**99.93**	99.07	99.10
Average	75.91	77.65	77.68	75.31	**87.44**	86.89
Inference (ms)	1.19	1.19	1.07	**0.20**	1.18	1.19

Table 1: Quantitative results, training circuit. With (fus1) and without (fus2) pre-training.

Analyzing the results it is obvious that the applied method did not provide a more accurate estimation in any of the observed areas. Looking at the IoU, the average performance of depth in relation to

the proposed fusion strategy is actually worse than when using only depth information. With this said, there is a rather significant gain of almost 1.7% in the prediction of obstacles. Suprisingly, the fusion network without a pre-training phase achieved better results than with pre-trainig.
The RGB-D network dominated in most of the observed classes and evaluation metrics. These results are not surprising due to the fact that this contains the highest concentration of data of the measured approaches. This is closely followed by RGB data, providing very similar results.
The laser scan yielded top performing results on several evaluation metrics, however, it should be noted that the LRF predictions are based on the topology view of the scene which contains much less information and does therefore not fulfill the objective of classifying the entire scene.
Even though RGB data provides significant improvement when compared to depth, this prediction is heavily dependent on the textures in the scene. In our case, we are trying to construct a classifier that is independent of scene textures.
The car class got the lowest score in all measured evaluation metrics. This is largely due to unbalanced sample distribution in the dataset as only 2% of the datapoints contained this class. Additionally, the car and obstacle classes have very similar spatial features but different textures which contributes in favor for the RGB and RGBD networks.
The inference of all networks are kept below 1.2 ms for all networks which is very promising. This corresponds to about 83 Hz which is more than double the rate of the LRF. This is when applied on Nvidia RTX 2080 which is more sophisticated than the Nvidia Jetson, however this shows great promise for real-time performance of inference greater than 15 HZ on the test bed.

4.2 NEW CIRCUIT

To evaluate the robustness of the networks, an additional circuit is constructed with different textures and objects. In this course, the obstacles are made up of red and blue cones instead of boxes. Additionally the walls are covered with cardboard in order to introduce new textures in this area. The ground is kept the same as in the training phase. From this scene an additional 108 data points are collected and labelled for testing. No training is conducted on this data. The tests are conducted in the same manner as on the training circuit.

	FUS1	FUS2	DEPTH	LASER	RGB	RGB-D
Iou (%)						
Wall	82.30	83.49	79.78	**87.99**	33.79	36.39
Ground	93.13	91.90	93.13	**99.71**	92.88	93.84
Obstacle	28.00	**30.00**	25.76	12.14	5.04	1.13
Car	23.62	**27.66**	22.27	0.00	0.68	0.33
Unknown	97.68	98.12	97.73	**99.99**	81.82	86.76
Average	64.95	**66.23**	63.73	59.97	42.84	43.69
Precision (%)						
Wall	**96.00**	93.22	95.32	90.99	69.69	78.74
Ground	93.80	94.51	94.31	**99.71**	94.63	94.85
Obstacle	38.32	**42.35**	32.59	35.58	6.17	1.38
Car	28.68	**32.86**	26.20	0.00	2.33	0.65
Unknown	98.46	98.73	98.74	**100.00**	90.36	94.16
Average	71.05	**72.33**	69.43	65.26	52.64	53.96
Recall (%)						
Wall	85.23	88.89	83.00	**96.32**	39.60	40.39
Ground	99.23	97.09	98.67	**100.00**	98.05	98.87
Obstacle	51.95	48.12	**55.29**	15.68	26.27	9.30
Car	44.35	**45.81**	43.23	0.00	1.80	2.20
Unknown	99.19	99.38	98.97	**99.99**	89.64	91.74
Average	**75.98**	75.86	75.83	62.40	51.07	48.50
Inference (ms)	1.17	1.17	1.11	**0.20**	1.20	1.23

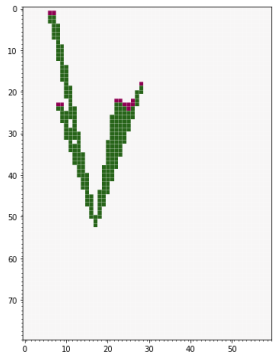
Table 2: Quantitative results, unseen circuit. With (fus1) and without (fus2) pre-training.

In this scene, we see a general improvement with the proposed fusion strategy. The fusion network provides the highest average score on all measured metrics with the network without pre-trainig performing the best. Notable here is the significant difference of car predictions. Comparing depth and fusion to their corresponding performance in the scene used for training, there is a difference of 15% in the fusion network and 9% with depth. The reason for this is believed to be the resemblance between a low box and the car from the training scene. Since cones were used as obstacles in the new scene, miss-classification is less likely to occur between these classes, thus increasing performance in this class.
As expected the RGB and RGB-D methods performed poorly in this

context since there exists novel objects with new texture patterns in the scene. When we compare RGB and RGB-D the increase in performance with depth introduced was significantly small which indicates that the network relies greatly on textures since this provides more descriptive information in the training phase.
An interesting finding is that the laser predictions on the car class in this scene gave 0% in all metrics. This is suspicious as it suggests that not a single pixel was correctly classified in the car class. An explanation for this could not be found by examining the data.

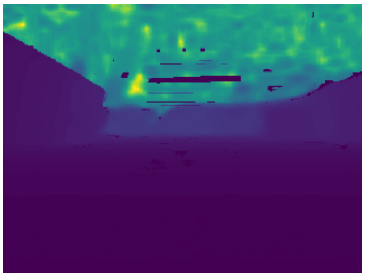
4.3 PREDICTIONS

In figures 16,17,18,19 predictions from each network on both circuits can be viewed. In these visualizations, it is observable that the fusion network in all cases has a better estimate of the position of semantics in the scene. With that said, there is also a noisy area in all samples of fusion predictions. This area varies in size and position from image to image. Looking at the laser scan, this area can roughly be translated to the missed laser points in the middle of the scan. Comparing the predictions made in the training circuit and the unseen circuit, the noise is less significant and occurs more frequently in the upper parts of the predicted image. This is an interesting observation as it would suggest that the network has observed a relationship between the depth image and laser scan in terms of displacement and orientation. With that said, this finding is a visual observation and thus hard to quantify and measure. Since the area of missing scans is quite large this could just be coincidence and while it can be observed in several occasions it is not obvious in all predictions.
Also notable is the amount of missing scans in the laser data on the training set. The reason for this is believed to be the reflectiveness of the walls in the circuit since this behaviour was greatly reduced in the circuit with cardboard walls. This problem was not as frequent in the depth estimations which is likely to be caused because of the different methods applied by the sensors when perceiving depth. The stereo camera utilizes stereo disparity with the help of a laser projector while the LRF relies on time-of-flight technology. When a reflective surface is angled, the laser point bounce of the surface in a different direction which results in the LRF not receiving the laser point and is thus unable to obtain the depth estimate.



(a) Laser input

(b) RGB input



(c) Depth input

(d) Ground truth

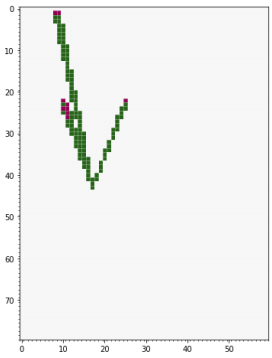
(e) Depth prediction

(f) RGB prediction

(g) RGBD prediction

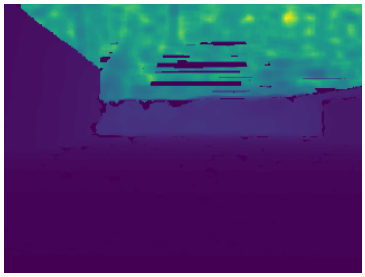
(h) Fusion prediction

Figure 16: Prediction 1 from each of the networks in the training circuit.



(a) Laser input

(b) RGB input



(c) Depth input

(d) Ground truth

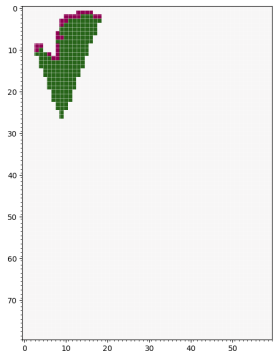
(e) Depth prediction

(f) RGB prediction

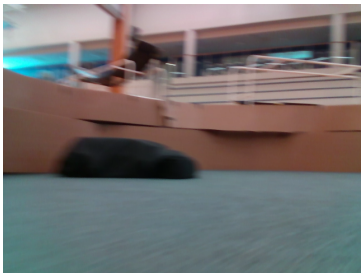
(g) RGBD prediction

(h) Fusion prediction

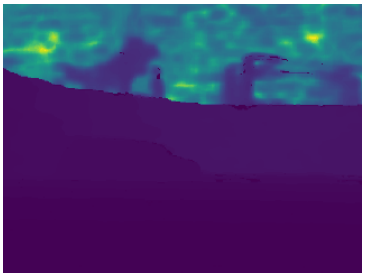
Figure 17: Prediction 2 from each of the networks in the training circuit.



(a) Laser input



(b) RGB input



(c) Depth input

(d) Ground truth

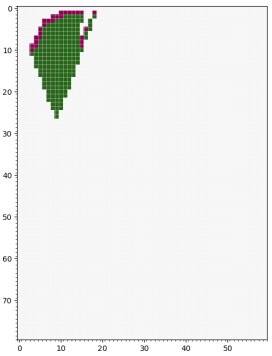
(e) Depth prediction

(f) RGB prediction

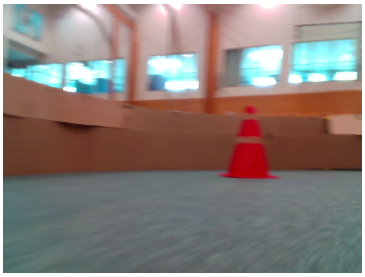
(g) RGBD prediction

(h) Fusion prediction

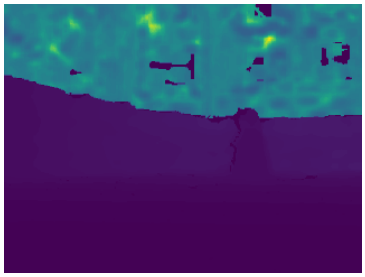
Figure 18: Prediction 1 from each of the networks in the unseen circuit.



(a) Laser input



(b) RGB input



(c) Depth input

(d) Ground truth

(e) Depth prediction

(f) RGB prediction

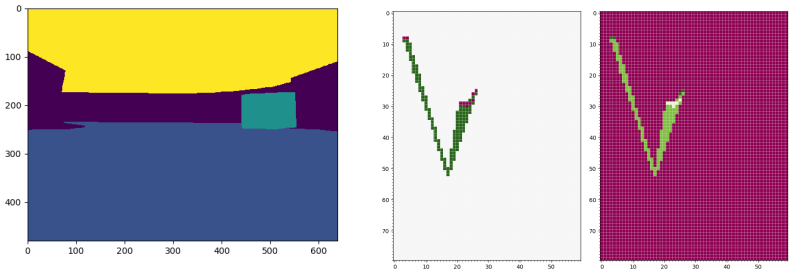
(g) RGBD prediction

(h) Fusion prediction

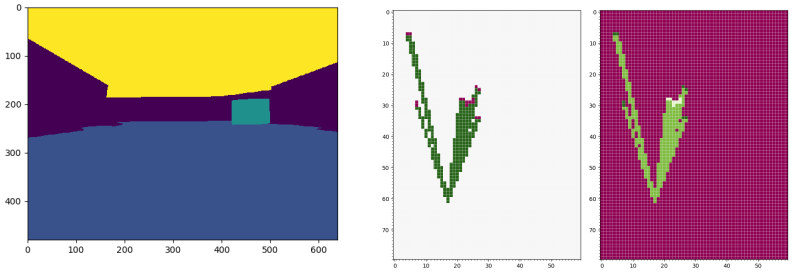
Figure 19: Prediction 2 from each of the networks in the unseen circuit.

4.4 LABEL TRANSFER

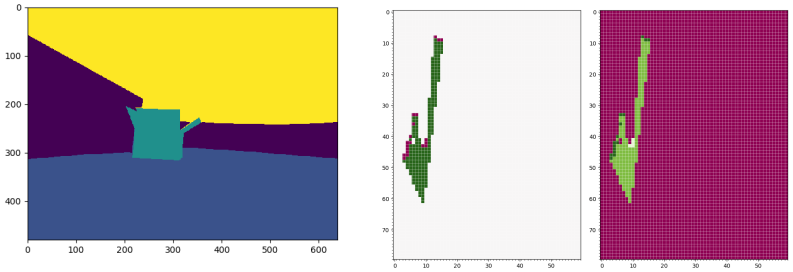
Results of the label transfer method is not easily quantizised without hand labeling the data points which is what the method was created to avoid. A visualization of some examples can be viewed in figure 20. From a visual perspective the method performed satisfactory. In some cases, for example in the last figure, an obstacle label could still appear in the background of the observed object. The reason for this is because of faulty clustering. It is possible to adjust the threshold for the clusters however this instead causes over-segmentation, i.e. multiple segments may appear for the same object. This instead causes problem when applying the label since the amount of points are compared to points in the overlapping cluster.



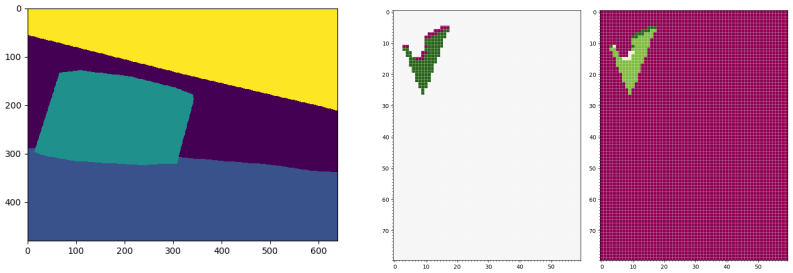
(a) label (b) laser input (c) laser label



(d) input (e) input (f) result



(g) input (h) input (i) result



(j) input (k) input (l) result

Figure 20: Transferred label results

5 DISCUSSION

5.1 FUSION STRATEGY

The proposed strategy of fusing modalities by feature concatenation on a network level yielded an improvement of around 2.5% in IoU in comparison to unimodal depth segmentation when introduced to an unseen circuit. These results show that there are information gains to retrieve from incorporating LRF scans with depth data for the task of semantic scene segmentation. Although this is an improvement, there is reason to believe that a more effective fusion strategy could be conducted for improved results. In the predictions from the fusion network, an area of noise was present in all observed samples which is believed to be correlated to missing laser scans. A factor for this result is believed to be because the feauture contributions from each sensor is not evaluated enough in terms of correlated information gain. As argued in section 3.3.4 the preferred method would have been to apply dimensionality reduction using PCA as suggested in [30] after concatenation is performed. Another approach to fuse the data would be adding an additional encoder-decoder structure in every fusion step. This also yields the desired effect of feature extraction. This would, ofcourse, drastically increase the inference time.
In [24] the authors find that constructing a multimodal fusion layer by feature concatenation yielded poor results arguing against this form of fusion. However, this is directed towards a fusion step which is employed on a single shared representation layer. In our case, we are introducing the fusion gradually and while the feature maps are in an abstract state. The idea behind this is that it could provide a difference due to the decoder learning correlated features between the modalities gradually. The stacked decoders performs up-sampling of the concatenated data however never has the ability to consider the discriminative value of the features from each modality in relation to each-other in terms of predicting the target output.
Another issue that leads to reduced fusion performance is the lack of laser points in the scans which was caused by the reflective walls in the circuit. This resulted in gaps occurring in the laser scan. For this kind of issues a generative module that can compensate for missing data would be a useful complement. Another approach is to extract the corresponding depth data from the stereo camera and fill in the areas which are missing in the laser scan. The main problem with this is the relationship between the depth images aligned with the RGB data and the raw depth which is captured using the fish-eye

lens. The raw depth data is relatable to the LRF scan, however when it is aligned with the RGB image there is loss in data due to different camera intrinsics. Therefore there is no exact way of translating from the aligned depth to raw depth. An estimation may however be possible.
In section 3.4.1 the classes of each modality is established. In the depth data the unknown class is considered all information outside the confinement of the circuit. However in the laser scans, the unknown class is regarded as areas which is unseen by the LRF. This is not entirely equivalent as these laser points often occur within the circuit due to the mentioned problem with loss of points. If there was no missed scans this would be more accurate however, due to occlusion, areas behind an obstacle would still be considered as noise even if it is within the circuit. Addressing a different approach is difficult because it requires knowledge of information behind the observed obstacle. For the depth estimation this is not an issue as this modality offers a 3D representation where the height can be leveraged to determine the unknown areas. A possible solution for the LRF modality would be to implement a generative model which could fill in these gaps so that the residing area could be enclosed and thus it could be concluded that everything outside is considered unknown.

5.2 UNIMODAL SEGMENTATION

Isolating predictions from each modality the RGB and RGB-D networks provided the most accurate predictions in the experiment on the trained circuit. What is important to note here is that these approaches is based on textures in the scene which is undesirable as the method is to perform in a circuit which differ in textures. This is quantified in when applied on an unseen circuit as the performance of these networks is drastically decreased. In this scene, predictions based on depth and laser data instead excel, as expected. Noteable is the possibility of invalid labels which may occur due to the label transferring approach. This could effect the performance of the laser network.

5.3 LABEL TRANSFER

The results of adapting labels from RGB images to the laser data gave visually acceptable result. From an analytic perspective it is difficult to measure the performance due to the mentioned problem of target reference. The prediction of the network is based on the labels so this is not a measurement in those terms. Known issues consists of the approach of the hierarchical clustering not dividing the clusters perfectly due to the distance threshold. This causes the labels to, in a few cases, appearing on the background of the observed

object. Although this issue is only present in very few of the observed samples, for the credibility of this thesis it is important to consider.

5.4 IMPLEMENTATION

The proposed segmentation solution has not yet been tested on the test-bed due to the time constraints. The network has been applied on the vehicle but only using the CPU on the Jetson board as there were difficulties with the version match of CudaNN, a deep learning module for Cuda necessary for 2D convolutional operations. The CPU implementation is not sufficient as this yields an inference of about .7 seconds which is far from real-time. This also consumes the hardware's full computational capacity which disables other necessary processes to run in parallel. To resolve this issue there was need to flash the hardware with a new image of Jetpack which is the SDK provided for the board. This would remove all data from the board which would cause problems for the additional groups of students working on the same hardware.

5.5 HARDWARE

A huge impact on the method conducted in this thesis was the hardware incompability between the stereo camera and test-bed plattform. The Nvidia Jetson is a system utilizing 64 bit ARM architecture which unfortunately is not officially supported by Intel realsense d435i. This is believed to be the main reason causing issues in terms of synchronization between camera and LRF. This led to additional mentioned problems in the label transferring method.
Before the intel camera was integrated in the system, the ZED stereo camera from Stereolabs was initially applied as visual unit. This camera relies on passive stereo disparity when performing depth estimations. This camera could not perform depth estimations on objects closer than 1.5 m which caused the inability to detect the ground in a predictable manner. This caused problems in the early stages of this thesis as data collection was first based on this setup.

6 CONCLUSION

6.1 POSED QUESTIONS

- 1 - **Does the LRF data provide additional information that increases the prediction accuracy of segmentation in the scene?** Yes, the implementation issued in this thesis yielded an improvement of 2.5% when compared to unimodal segmentation using only depth data in an unseen circuit.
- 2 - **Does the exclusion of color data provide a more robust classifier given a scene with unknown/varying texture?** No, the RGB and RGB-D networks provided a significant reduction of around 40% in IoU when applied on an unseen circuit. This resulted in an average IoU reduction of 22% and 21% when compared to the best performing fusion approach.

6.2 SUMMARY

This thesis has explored the problem of semantic segmentation using deep multimodal fusion of LRF and depth data. Two data set consisting of 1080 and 108 data points from different circuits have been created and manually labeled in 2D space and transferred to 1D using proposed label transfer method utilizing hierarchical clustering. The 1D representation of the laser data is used to create a 2D occupancy grid representation using ray casting for constructing the network input. The data sets has been used to train and validate the suggested method for segmentation using a proposed dual encoder-decoder network based on SalsaNet [1] with gradually fusion in the decoder using feature concatenation. Applying the suggested method yielded an improvement of around 2.5% average IoU when compared to unimodal segmentation using only depth data. A suggestion of feature extraction in the form of PCA or stacked auto-encoders is suggested as a further improvement for additional evaluation of this type of fusion. The suggested approach for implementation in the application of scene segmentation in a racing environment as explored in this thesis is utilizing the fusion network as this fulfills the objective of full scene description with a average IoU above 66% and an inference below 2 ms.

6.3 FUTURE WORK & IMPROVEMENT

For future work in this field it is highly recommended to apply a generative model to account for missed laser scans as this is considered one of the major flaws in the implementation. As mentioned in the summary, further feature extraction is advised in the fusion step as the feature correlation between modalities is believed to be insufficiently explored with this implementation. Other areas that would be considered further is increasing the grid of the laser scan for the ability to introduce fusion in additional network layers.

A

APPENDIX: DISCUSSION ON THE NETWORK ARCHITECTURE

As argued in section 3.3 the reason for conducting the presented network strategy depends on the difference of the two modalities. With that said, an early fusion strategy could be adopted by combining the overlapping area with a weighted utilization of modality depending on the intensity of the observed point. Due to the known better performance of LRF in longer ranges, a decision to use this modality on depth estimations with greater intensity can be concluded. At a very basic level, this type of approach would be the implementation of an early fusion strategy. Why the decision is made to not use this type of approach is because it only has the potential to, at its most significant impact, alter the estimations of a very small area of the depth map. What is desired is to create a network which can totally reshape the output depending on an input from the LRF. For this reason, early fusion is deemed insufficient and avoided in this implementation.
As mentioned in [31], late fusion is generally conducted in scenarios where modalities differ more. An approach of a shared representation layer after classification is motivated in this scenario. The authors of [24] find that a multi-modal fusion layer by simple concatenation of incoming connections yielded a worse result revealing that hidden units have strong connections to variables from individual modalities but few units that connect across modalities. In order to account for this a recommended strategy from [31] is applying dimensionality reduction in the shared representation layer. Suggested methods for this is PCA or stacked auto-encoders.
Given the applied network SalsaNet [1] utilizes an encoder-decoder architecture, the idea behind the suggested fusion strategy is that in the up-sampling process, the same type of dimensionality reduction as in an auto-encoder is performed and that therefore this can be leveraged for removing redundancies in the input space. This is also the reason why fusion is performed in the decoder.
In addition to the presented strategy, experiments were conducted with the same strategy of feature concatenation but at different depths and with varying numbers of layers. Two other setups explicitly were performing fusion in the encoder with the same number of layers and performing fusion in both the encoder and decoder around the bottleneck of the network with a total of three layers. Testing these networks on the training circuit, the decoder fusion strategy proved the best performing and thus were utilized in the final model. Tests using the unseen circuit was never performed on the two other men-

tioned approaches, however, an improvement in the training circuit yielded better results in the novel circuit from tests conducted using the fusion network.

BIBLIOGRAPHY

[1] Eren Erdal Aksoy, Saimir Baci, and Selcuk Cavdar. Salsanet: Fast road and vehicle segmentation in lidar point clouds for autonomous driving. In *IEEE Intelligent Vehicles Symposium (IV2020)*, 2020.

[2] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

[3] Martin Buehler, Karl Iagnemma, and Sanjiv Singh. *The 2005 DARPA grand challenge: the great robot race*, volume 36. Springer, 2007.

[4] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018.

[5] Alberto Garcia-Garcia, Sergio Orts-Escolano, Sergiu Oprea, Victor Villena-Martinez, and José García Rodríguez. A review on deep learning techniques applied to semantic segmentation. *CoRR*, abs/1704.06857, 2017. URL http://arxiv.org/abs/1704.06857.

[6] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[7] Saurabh Gupta, Ross Girshick, Pablo Arbeláez, and Jitendra Malik. Learning rich features from rgb-d images for object detection and segmentation. In *European conference on computer vision*, pages 345–360. Springer, 2014.

[8] Caner Hazirbas, Lingni Ma, Csaba Domokos, and Daniel Cremers. Fusenet: Incorporating depth into semantic segmentation via fusion-based cnn architecture. In *Asian conference on computer vision*, pages 213–228. Springer, 2016.

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

[10] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[11] Samira Ebrahimi Kahou, Christopher Pal, Xavier Bouthillier, Pierre Froumenty, Çaglar Gülçehre, Roland Memisevic, Pascal Vincent, Aaron Courville, Yoshua Bengio, Raul Chandias Ferrari, et al. Combining modality specific deep neural networks for emotion recognition in video. In *Proceedings of the 15th ACM on International conference on multimodal interaction*, pages 543–550, 2013.

[12] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.

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

[14] Raul Mur-Artal and Juan D. Tardós. ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras. *CoRR*, abs/1610.06475, 2016. URL http://arxiv.org/abs/1610.06475.

[15] N. Neverova, C. Wolf, G. Taylor, and F. Nebout. Moddrop: Adaptive multi-modal gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(8):1692–1706, 2016.

[16] Matthew O'Kelly, Varundev Sukhil, Houssam Abbas, Jack Harkins, Chris Kao, Yash Vardhan Pant, Rahul Mangharam, Dipshil Agarwal, Madhur Behl, Paolo Burgio, and Marko Bertogna. F1/10: an open-source autonomous cyber-physical platform. *CoRR*, abs/1901.08567, 2019. URL http://arxiv.org/abs/1901.08567.

[17] Michael L. V. Pitteway and Dereck J Watkinson. Bresenham's algorithm with grey scale. *Communications of the ACM*, 23(11): 625–626, 1980.

[18] Soujanya Poria, Erik Cambria, and Alexander Gelbukh. Deep convolutional neural network textual features and multiple kernel learning for utterance-level multimodal sentiment analysis. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 2539–2544, 2015.

[19] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 918–927, 2018.

[20] D. Ramachandram and G. W. Taylor. Deep multimodal learning: A survey on recent advances and trends. *IEEE Signal Processing Magazine*, 34(6):96–108, Nov 2017. ISSN 1558-0792. doi: 10.1109/MSP.2017.2738401.

[21] Danijela RistiÄ-Durrant, Ge Gao, and Adrian Leu. Low-level sensor fusion-based human tracking for mobile robot. *Facta Universitatis, Series: Automatic Control and Robotics*, 1(1):17–32, 2016. ISSN 1820-6425. URL http://casopisi.junis.ni.ac.rs/index.php/FUAutContRob/article/view/1449.

[22] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[23] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014.

[24] Nitish Srivastava and Russ R Salakhutdinov. Multimodal learning with deep boltzmann machines. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 2222–2230. Curran Associates, Inc., 2012. URL http://papers.nips.cc/paper/4683-multimodal-learning-with-deep-boltzmann-machines.pdf.

[25] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-first AAAI conference on artificial intelligence*, 2017.

[26] M. Teichmann, M. Weber, M. ZÃ¶llner, R. Cipolla, and R. Urtasun. Multinet: Real-time joint semantic reasoning for autonomous driving. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1013–1020, June 2018. doi: 10.1109/IVS.2018.8500504.

[27] Bichen Wu, Alvin Wan, Xiangyu Yue, and Kurt Keutzer. Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1887–1893. IEEE, 2018.

[28] Bichen Wu, Xuanyu Zhou, Sicheng Zhao, Xiangyu Yue, and Kurt Keutzer. Squeezesegv2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 4376–4382. IEEE, 2019.

[29] Di Wu, Lionel Pigou, Pieter-Jan Kindermans, Nam Do-Hoang Le, Ling Shao, Joni Dambre, and Jean-Marc Odobez. Deep dynamic neural networks for multimodal gesture segmentation and recognition. *IEEE transactions on pattern analysis and machine intelligence*, 38(8):1583–1597, 2016.

[30] Dong Yi, Zhen Lei, and Stan Z Li. Shared representation learning for heterogenous face recognition. In *2015 11th IEEE international conference and workshops on automatic face and gesture recognition (FG)*, volume 1, pages 1–7. IEEE, 2015.

[31] R. Zhang, S. A. Candra, K. Vetter, and A. Zakhor. Sensor fusion for semantic segmentation of urban scenes. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1850–1857, May 2015. doi: 10.1109/ICRA.2015.7139439.

COLOPHON

This document was typeset using the typographical look-and-feel `classicthesis` developed by André Miede. The style was inspired by Robert Bringhurst's seminal book on typography "*The Elements of Typographic Style*". `classicthesis` is available for both LaTeX and LyX:

http://code.google.com/p/classicthesis/

Happy users of `classicthesis` usually send a real postcard to the author, a collection of postcards received so far is featured here:

http://postcards.miede.de/

.

PO Box 823, SE-301 18 Halmstad
Phone: +35 46 16 71 00
E-mail: registrator@hh.se
www.hh.se