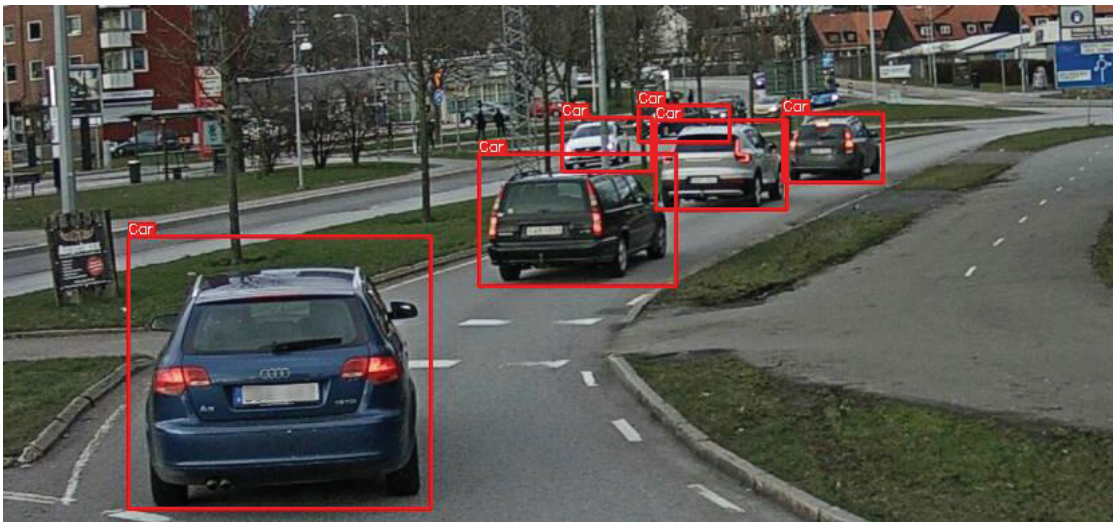
SHARPEN

Skalbara högautomatiserade fordon med robust perception

Publik rapport



Författare: Hans Salomonsson, Eren Erdal Aksoy, Hannes von Essen, Anton Kloek, Tiago Cortinhal, Devdatt Dubhashi

Datum: 2022-07-26

Projekt inom Trafiksäkerhet och automatiserade fordon - FFI - 2018-12-11

FFI Fordonsstrategisk Forskning och Innovation

VINNOVA Energimyndigheten TRAFIKVERKET FKG VOLVO SCANIA VOLVO

Table of Contents

1. Executive Summary

The aim of the project was to improve today's machine learning methods based on deep learning to be more robust in challenging environments, such as night, rain, snow and dirt on the sensors. To achieve this goal, both technology to generate synthetic data, where one can control these variables, as well as the development of new methods that can better handle these situations. The project has also focused on bringing these systems closer to production by compressing them to reduce their resource usage in the vehicle.

We have built an interface to Carla Simulator, which is built in Unreal Engine, with semi-automatic functionality to generate large amounts of 3D worlds with challenging conditions. The sensor array was based on the Kitty open database for the development of autonomous vehicles. Furthermore, we used Nvidia's Jetson Xavier AGX as a hardware platform for accelerated deep learning. Our results show that you can improve today's system with the use of synthetic data and also significantly reduce the cost of annotation of data, as we demonstrated that we can replace 90% of the annotated data with synthetic data.

Furthermore, we have developed methods that can synthesize data from broken sensors in vehicles through machine learning and other sensors. We have also explored the best way of sensor fusion for object recognition. Furthermore, we have shown with methods developed in the project that we can reduce the latency by over 60%.

We also developed a demonstrator on an Nvidia Xavier AGX platform.

Organizationally, monthly meetings and additional technical meetings were held when necessary. The coordinating partner at the start was Volvo GTT. This role was taken over by Embedl, who was added as a partner during the course of the project. Repli5 has been spun off from MIS to commercialize synthetic data generation. Private financing and first customer obtained.

2. Summering

Syftet med projektet var att förbättra dagens maskininlärningsmetoder baserad på djupinlärning att bli robustare i utmanande miljöer, såsom natt, regn snö och smuts på sensorerna. För att uppnå detta mål har både teknologi för att generera syntetisk data, där man kan kontrollera dessa variabler, samt utveckling av nya metoder som bättre kan hantera dessa situationer. Projektet har också fokuserat på att ta dessa system närmre produktion genom att komprimera dem för att minska dess resursanvändande.

Våra resultat visar att man kan förbättra dagens system med användandet av syntetisk data och även signifikant minska kostnaden för annotering av data då vi visat att vi kan ersätta 90% av den annoterade datan med syntetisk data.

Vidare har vi utvecklat metoder som kan syntetisera data från trasiga sensorer i fordon genom maskininlärning och andra sensorer. Vi har också utforskat bästa sättet för sensor fusion för objektigenkänning. Vi har också visat att med metoder utvecklade i projektet att vi kan reducera latensen för samma objektigenkänningssystem med över 60%.

Vi har byggt ett interface till Carla Simulator, som är byggd i Unreal Engine, med semi-automatisk funktionalitet för att generera 3D världar med utmanande förhållanden. Sensorkonfigurationen var

baserad på den öppna databasen Kitty för utveckling av autonoma fordon. Vidare använde vi oss av Nvidias Jetson Xavier AGX som hårdvaruplattform för accelererad djupinlärning.

Organisatoriskt genomfördes månadsmöten och ytterligare tekniska möten vid behov. Koordinerande partner vid start var Volvo GTT. Denna roll togs över av Embedl, som tillkom som partner under projektets gång. Repli5 har knoppats av under projektet från MIS för att kommersialisera den syntetiska datagenerering. Privat finansiering och första kund erhållen.

3. Background

Artificial Intelligence (AI) is the field of developing intelligent agents which perceive the environment and take actions that maximize their chance of achieving the given missions. Machine Learning and more particularly Deep Learning (DL) are rapidly growing subfields of AI. Particularly, DL has become highly popular in the last five years with the availability of vast amounts of data and computational resources in addition to a number of innovations in the field. The field has now reached the peak of inflated expectations according to the latest Gartner hype cycle[1].

DL has become one of the cornerstones in the development of Autonomous Vehicles (AV) and is expected to play a major role in such products. It is therefore of critical importance that the Swedish automotive industry builds cutting edge DL competence and excels in adapting the technology. It is expected that AI technology will be present in most of the automation stack[2] and specifically in the Vehicle Environment Model in the short term[3].

Traditionally, Computer Vision methods have been the dominant choice for building the perception layer. However, these methods were not capable of providing good enough performance for building an L4 ready perception system[3]. Today, there is a general consensus that DNNs have the potential to achieve high enough accuracy in a wide variety of conditions which is seen as the key for an L4 perception system and used for almost all the necessary perception components as the core solution. Most recent AD works have, so far, focused on the autonomous car applications, whereas the autonomous truck approaches, in particular for confined area related applications, have not attracted enough attention.

4. Purpose, Research Question and Method

In order to make autonomous vehicles to operate in tough conditions in Confined Areas a set of needs must be addressed. The purpose of the project was to develop methods concentrated on the following needs:

- **Robust perception performance in tough environmental Conditions**: Weather conditions like rain and snow can drastically reduce the productivity of an autonomous logistic service, thus the ability to safely operate under different weather conditions is a key differentiator. By using a diverse set of sensors including LiDARs, Radars, multiple RGB and infrared cameras we aim at enabling the vehicle to operate during night as well as adverse weather conditions, thus a large and diverse dataset is needed. Furthermore, perception in confined areas poses some additional challenges vs. public roads. For instance, dirt level on sensors is higher and free space and object detection is harder since there is less structure and features in the environment.

[1] Gartner hype cycle 2018: https://tinyurl.com/y78evrb9

[2] The road to artificial intelligence in mobility—smart moves required https://tinyurl.com/ybnuzxna

[3] Mobileye Advanced Technologies, https://tinyurl.com/ycqseluo

- **Sensor Fusion**: Each sensor has different field of view. To create the necessary aggregated view of the world surrounding the vehicle, data readings coming from various sensors need to be fused in a proper format. Such a fusion also needs to be robust in cases of having sensor failures, i.e. when a set of sensors cannot provide any output. Therefore, there is a certain need of having a robust fusion method to merge different kinds of sensory readings to further generate semantics of the perceived surrounding environment.
- **Fail-Safe Operation**: A fault tolerant system increases the functional safety of the vehicle, and as a byproduct the efficiency of the logistic service is improved. A way to address sensor faults is using redundant sensor systems. The novel sensor dropout methodology will enable trained DNNs to operate even in cases when several of the sensors fail to operate correctly. Note that this technique is different from the well-known dropout regularization technique, where network nodes are randomly removed during training to reduce overfitting. Dropping out sensors will prevent feature co-adaptations and yield robust network responses in cases of having sensor failures. For this purpose, we will develop a generative network that learns mapping between different sensor modalities, i.e. generates imaginary representations of the failed sensory readouts by referring to the already available sensors, e.g. creating missing 3D point clouds from real 2D image pixels or vice versa.
- **Data generation**: Data is one of the key needs for developing DNNs, but obtaining it by collecting it and then manually annotating it is also a slow and expensive process. Thus the need for high performing synthetic data generators is crucial in speeding up the development time of such technologies. Another use-case for the data generators is to use them for validation and verification of the overall system design in different domains.

More specifically, the research questions we proposed in the start of the project were:

RQ1. How to fuse multi-sensory information? In the field of autonomous vehicles, there exists a large corpus of work, which rely on uni-modal sensory information. Environment perception using different sensor modalities has also been studied for decades, typically in the form of fusing camera and LiDAR inputs. However, fusing multisensory data streams to gain robust and efficient perception skills independent of the environment and weather conditions is still a challenge since data readouts have vast variations in time scales, dimensions (i.e. 2D versus 3D data), and signal types (i.e. continuous versus discrete). So far, various approaches including early and late fusions have been introduced. Using multimodal deep learning as the sensor fusion module for autonomous driving has not been addressed until recently[4] and is still a non-trivial problem. In this regard, we will investigate a hybrid fusion where both early and late fusion will be intertwined. We will not only enrich the extraction of features in the network early layers by incorporating the results from the late fusion, but will also feed the early network features forward to the late fusion stage. Such a combination of feed forward and feedback connections will leverage the fusion of different sensory modalities in a more semantic form.

RQ2. How to handle different sensor failures? Each sensor has a unique contribution to the perception pipeline of the vehicle. In real-world conditions, one or multiple sensors might naturally lose the functionality, but an intelligent vehicle should autonomously adapt itself to the existing sensor configuration. The project hypothesis is that applying a stochastic regularization, such as sensor dropout, during the network training can help networks learn cross-modality correlations while distinguishing unique contribution of each modality. Thus, the network will not be biased with any sensor features. It is further hypothesized that learning mapping between different sensor modalities can allow vehicles to generate missing sensory readouts in case of facing sensor failures. Such a sensor-tosensor mapping will allow the vehicle to create imaginary representations of the failed sensor readouts and thus leverage the autonomy needed by intelligent vehicles.

[4] Giering, M., Venugopalan, V., & Reddy, K. (2015, September). Multi-modal sensor registration for vehicle perception via deep neural networks. In 2015 IEEE High Performance Extreme Computing Conference (HPEC) (pp. 1-6). IEEE.

RQ3. How to operate in tougher environmental conditions? Weather, light and dirt conditions in the real world can introduce vast changes in the perceived environment, for instance, feature space of the sunny scene will be quite different compared to that of the snowy counterpart. Having a perception system robust to such harsh weather conditions (such as sun glare, foggy, rainy or snowy) is a must for autonomous vehicles. This problem still remains unsolved and is of utmost importance for gaining a full autonomy. In SHARPEN, we therefore address this nontrivial problem by developing robust DNNs which will be trained with a large dataset including such tough conditions.

RQ4. How does challenging conditions affect sensor output and how should this most efficiently be replicated in a simulation environment for realistic synthetic data generation? SHARPEN will cover the important impact of tough conditions, e.g. snow, rain, bad lightning, dust and confined areas, impact sensor data. Synthetic data unfortunately differs from real data, but this gap is 6 something that we are bridging in the on-going project [57] for safer, more robust and cheaper development of autonomous vehicles, where we make heavy use of machine learning to translate the simulated world into a world much closer to reality. By doing this we will show that we can improve the trained perception system's performance evaluated on real data and hence decrease the amount of real data that needs to be annotated and significantly reduce the cost. However, this work does not cover tough environment conditions. This is essential to analyse in detail and replicate in the simulation environment to develop the autonomous systems of tomorrow that need to cope with these conditions.

Moreover, the following related research questions have also been explored in the Project:

- What type of sensor fusion gives the best result for fusion of RGB and lidar data for a state of the art object detection model?
- To what extent does pre-training on synthetic data contribute to a better result when continuing to train on real data?
- How much does the benefit of synthetic pretraining vary depending on the amount of real data available?
- How much can a state of the art object detection model be compressed without dropping significantly in accuracy for an automotive use case?

5. Objective

SHARPEN will deliver results of different character, each providing support for the research questions and measurable goals.

- Robust perception: The developed algorithms will increase the robustness and the autonomy of the vehicles deployed in the confined areas under different environmental conditions. Improved productivity of the logistic solution will be positively impacted from the developed technology.
- Data generation: A mixed dataset including both automatic and manual annotations will be created. The data set will also include the relevant generated data. A tool for generating synthetic data with automatic annotations has great potential to save costs of obtaining the data needed for training and validation.
- Development of prototypes: A truck demonstrator that demonstrates the concept of robust perception will also be developed.
- Academic theses: Halmstad University will perform most of its work through a PhD student, i.e. SHARPEN will form the major part of a doctoral thesis.

The measurable goals of SHARPEN are:

G1. Robust perception under tough environmental conditions
The perception system will be validated by comparing its performance under good conditions vs. bad conditions. Good conditions means all sensors are working, good weather, daylight and clean sensors.

Bad conditions mean any combination of sensor failure, rain or snow, night and dirty sensors. The intersection over union (IoU) of the predictions vs. the ground truth will be calculated for good conditions and for a similar scene in bad conditions. The goal is that when the object or road is in the field of view in both the good and bad condition scene, the IoU degradation in the bad conditions shall be less than 20%.

G2. Scalable and Realistic Data generation
The manually annotated dataset should have at least 150 000 annotations, where an annotation refers to either an object or free space definition. The synthetic dataset should have at least one million images and diverse enough that when training a network on the synthetic data, the accuracy shall be at least 75% of the accuracy that the same network gets when trained on the manual dataset.

Due to changes in the project scope, resulting from the negative impact of the global pandemic, which impacted the extent to which Volvo Group could contribute to the project, no real annotated dataset was developed in the Project. Both G1 and G2 depend on this dataset for quantifying the results of the developed technologies in the project. To mitigate the impact of not developing a dataset, an open dataset developed for autonomous vehicles (Kitti) was used as a replacement[5]. However, there are no tags for e.g. weather conditions in this dataset, which means that comparing between "good and bad" conditions needed for G1 is not possible without a significant amount of manual condition tagging of the dataset. Hence, this experiment is left to be done in the accepted continuation project RoadView. Please note that all the components to perform the experiment for G1 have been developed in SHARPEN, so once the dataset is available, it is trivial to make the experiment. The G2 goal is measured on the Kitti dataset, but mismatch between definitions between classes made it slightly more complicated than first anticipated, but the problem was solved.

[5] http://www.cvlibs.net/

6. Results and Goal Fulfillment

Impressive results have been achieved on various topics. In this chapter we will first present these results in topic specific subsections and then relate these results to FFI:s goals on program and subprogram level.

6.1 Synthetic Data Generation

MIS main goal in this project was to deliver a dataset of synthetic data, this was achieved with a combination of open source and inhouse developed software. To generate the synthetic dataset a simulator designed for driving and support for sensor recording is needed. Carla[6] was chosen as a simulator in combination with improvements, extensions and novel tools for scalable generation of 3d worlds and assets. Extensions to Carla have mainly focused on weather parameters, simulations of weather previously not supported in Carla, dust and dirt simulation, and support for different semantic class definitions then previously supported by carla. The updated class definitions were originally set by Volvo for the scope of "confined areas" but were later updated to the KITTI(SOURCE) dataset classes after the scope of the project was updated. To avoid further issues with changing class definitions a super set of classes was created so that conversion could happen offline to different class definition sets. Since many published datasets have larger nr of classes in training set then they use for evaluation, a limited superset was created that focuses on the evaluation labels rather than the training labels, since tagging all the 3d assets would incur large amounts of manual labor if the class set gets too large.

Figure 1. The Carla Simulator has been extended to be able to generate data with dirt on the sensor, dust clouds as well as snow.

[6] https://carla.org/

Tools for scalable generation of 3d worlds and assets
One of the largest challenges with successfully using synthetic data in training perception systems that will be applied in production, or otherwise evaluated on logged data, is the domain gap between logged and synthetic data. This domain gap is partly due to style difference but mainly lack of variance in the assets and materials in the simulated world. To mitigate this a large amount of assets and world needs to be created, incurring a large cost for manual labor.

To reduce this labor and get better scalability, tools were developed to automate some of the steps in generating 3D worlds. These tools used a combination of procedural algorithms and deterministic algorithms. Procedural solutions were used on lower level descriptions of the world like placement of assets in nature, like rocks, trees, and similar. Deterministic algorithms were used on higher level descriptions of the world like road networks. This gives a level of control when laying out the main features of the world, but still a high level of automation when adding the smaller detail, speeding up the process. This project was internally called WorldGenerator.

The ambitions to stick close to opensource standards, and industry standards whenever possible resulted in picking OpenDRIVE (xodr) [source] as the format for road network descriptions. OpenDRIVE is an XML based roadnetwork description standard, that on top of the network layout can describe road features like lanes, widths, heights, road banking and much more. OpenDRIVE also includes objects adjacent, and connected to the road, like street signs, bus stops and more, but it is not intended to describe an entire area with buildings, nature, and topology like a map would.

Generating 3D Worlds and Scenarios
The first maps and assets created using WorldGenerator were designed to simulate confined areas per the original scope. Later more city like maps were created to better fit the evaluation against the KITTI dataset. Due to time constraints from changing scope, a combination of WorldGenerator worlds, and Carla worlds were used. In some cases WorldGenerator tools were used to update existing Carla worlds to better fit a European scope.

With these worlds, materials that are affected by simulated weather conditions were created. These materials supported rain, snow, and dirt. Since Carla already had decent support for rain, implementing new materials that supported those weather presets was straightforward and followed Carla documentation. With snow and dirt, novel approaches had to be developed and tested. The snow and dirt materials were created for the confined areas and tested in a mine like setting, these were however not later adapted to the change in scope due to lack of time and remaining budget for MIS after project scope update.

The first iteration of sample data was generated with two synthetic lidars and two synthetic RGB cameras to fit the original scope of a construction machine in a confined area scope as set by Volvo. This was updated to instead resemble a simplified version of the KITTI dataset sensor setup when the scope of the project was updated. This updated sensor setup included a single lidar, 2 cameras (stereo), GNSS, and IMU.

Annotations included for the lidar is semantic tagging of each datapoint in the point cloud.
For the cameras the dataset included annotations for semantic segmentation and depth maps.

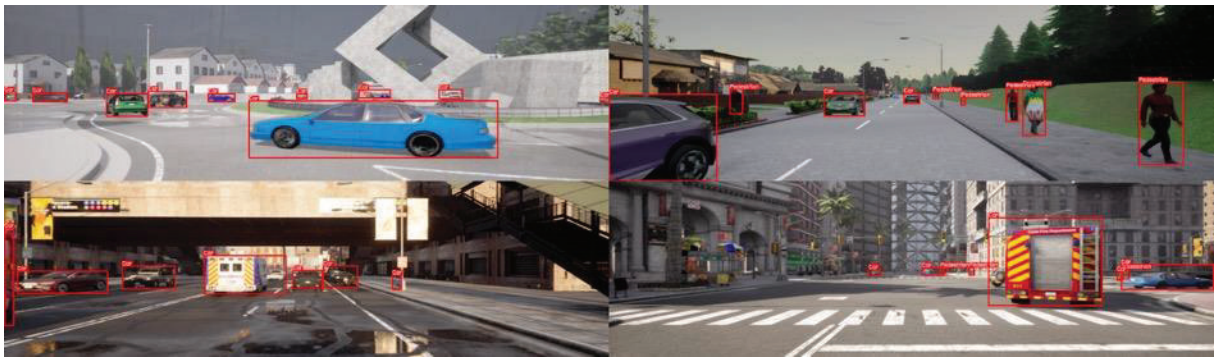


Figure 2. Example images and annotations from the synthetic dataset.

Experiments and Results

The original 1xRGB model was trained on the synthetic dataset from scratch for 10 epochs (note that the synthetic dataset is much larger than KITTI so one epoch corresponds to more images), with the normal hyperparameters. It was then finetuned on varying fractions of the KITTI dataset, again with the normal hyperparameters, but with the number of epochs adjusted such that the same total number of images seen during training remained the same. As a comparison, the same fractions of KITTI were also trained from scratch.

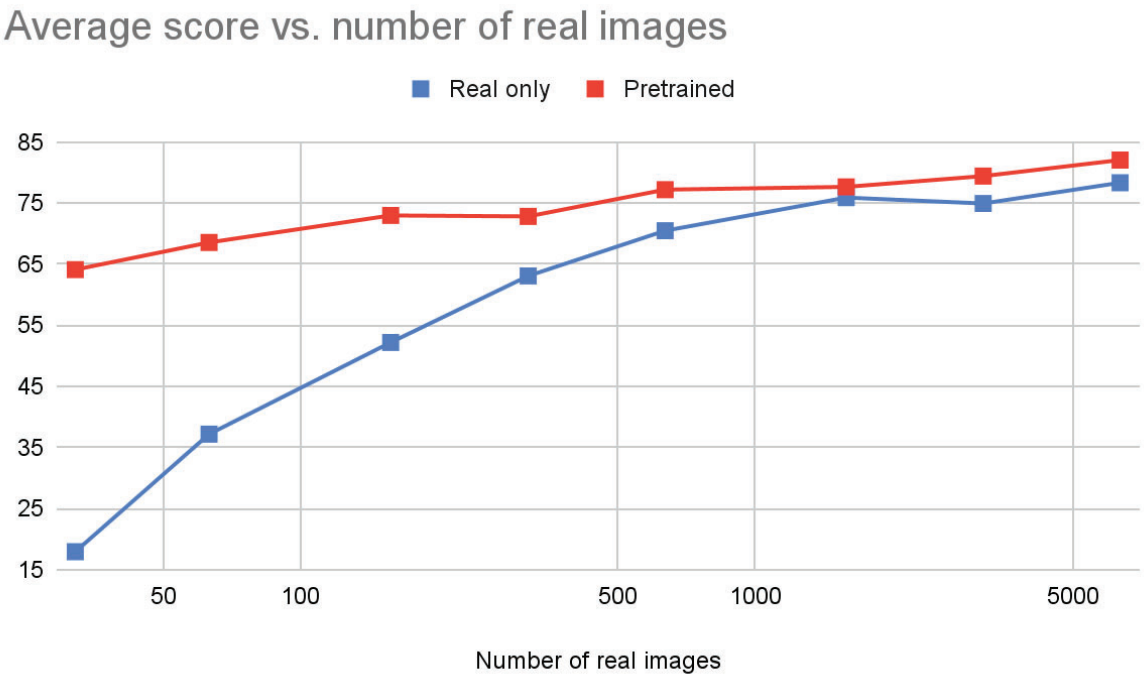


Figure 3. Comparison of training on different amounts of real data with or without synthetic pretraining.

Figure 3 shows the average score when training on different fractions of the real dataset, either from scratch (the blue curve) or after pretraining on the entire synthetic dataset (the red curve). Note the logarithmic scale of the x-axis. The complete metrics are listed in the table below. We can see that starting from the synthetically pretrained model enables us to reach high accuracy even with only a few hundred real images. And even with as few as 32 real images, we reach 82% of the original accuracy.

	# Real images	Car			Pedestrian			Cyclist			Average score
		Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard	
Real only	6347 (100%)	95.3	89.1	87.3	87.4	82.1	76.8	72.0	59.4	55.7	78.3
	3173 (50%)	95.2	89.3	86.8	87.8	81.2	75.6	61.4	49.9	47.2	74.9
	1586 (25%)	95.0	88.8	86.2	81.1	76.0	69.9	72.0	58.8	55.4	75.9
	634 (10%)	94.8	85.1	82.1	81.9	74.2	67.0	54.8	48.7	45.7	70.5
	317 (5%)	94.0	80.8	77.3	76.1	69.2	61.4	40.3	34.8	33.8	63.1
	158 (2.5%)	91.7	78.3	74.0	67.1	61.0	53.3	16.2	14.5	13.6	52.2
	63 (1%)	84.4	63.8	59.6	43.3	38.6	33.4	4.3	3.3	4.0	37.2
	32 (0.5%)	42.0	32.7	27.9	18.1	18.4	16.0	2.3	2.2	2.0	17.9
Pretrained	6347 (100%)	95.0	90.9	88.6	91.5	85.1	80.9	79.2	65.4	61.8	82.0
	3173 (50%)	95.6	89.0	87.3	90.3	83.8	79.3	74.2	59.4	55.9	79.4
	1586 (25%)	95.1	88.2	86.2	88.2	81.3	76.5	68.5	58.7	56.3	77.7
	634 (10%)	95.4	89.4	85.8	86.2	79.4	74.5	71.3	57.7	55.1	77.2
	317 (5%)	94.6	85.9	82.4	89.2	80.4	72.9	56.7	47.8	45.6	72.8
	158 (2.5%)	93.9	85.2	81.2	84.8	75.7	68.9	64.9	53.1	49.2	73.0
	63 (1%)	93.8	79.9	75.3	87.4	76.3	67.4	52.5	43.5	40.9	68.6
	32 (0.5%)	81.2	66.7	58.1	90.0	80.2	71.7	47.8	41.6	39.9	64.1

Table 1. Results for the synthetic data experiments.

We can also see that while the benefit of the pretraining is the largest when we have few real images, we get an improvement even when using the full real dataset, and we need less than half of the original real dataset to reach the same accuracy as the original model trained only on the real dataset.

6.2 Sensor Dropout

Furthermore, HH has intensively worked on developing various robust and accurate perception modules required for the sensor fusion (WP4) and sensor dropout (WP5) modules. In this respect, HH has implemented two advanced neural network models for the semantic segmentation of 3D LiDAR point clouds. The first model is called SalsaNet [2] which takes top-view projection of front-view LiDAR point clouds and returns points that belong free-space and vehicles in the scene. The second network is the next generation of the SalsaNet model, hence, named SalsaNext [3]. As the main contribution, SalsaNext performs uncertainty-aware semantic segmentation of a full 3D LiDAR point cloud in real-time. In other words, SalsaNext can predict not only a class label for each LiDAR point, but also returns a confidence score for each segment prediction. Note that this contribution plays a crucial role for achieving a seamless late sensor fusion, where the final network predictions for different sensor modalities are unified to contribute the delivery of the Sensor Fusion networks in WP4.

More specifically, in [2], HH has studied the joint segmentation of the road, i.e., drivable free-space, and vehicles using 3D LiDAR point clouds only. In this work, HH has proposed a novel “SemAntic Lidar data SegmentAtion Network”, i.e., *SalsaNet*, which has an encoder-decoder architecture where the encoder part contains consecutive ResNet blocks. Decoder part rather upsamples features and combines them with the corresponding counterparts from the early residual blocks via skip connections.

The input for the *SalsaNet* is the Bird-Eye-View image projection of the point cloud. Final output of the decoder is then sent to the pixelwise classification layer to return semantic segments.

In [2], the network's performance was validated on the KITTI dataset[7] which provides 3D bounding boxes for vehicles and a relatively small number of annotated road images (~300 samples). HH has proposed an *auto-labeling* process to automatically label ~11K point clouds in the KITTI dataset. For this purpose, HH has employed the state-of-the-art methods[8,9] to respectively segment road and vehicles in camera images. These segments are then mapped from camera space to LiDAR to automatically generate annotated point clouds.

Quantitative and qualitative experiments on the KITTI dataset show that the proposed *SalsaNet* significantly outperforms other state-of-the-art semantic segmentation approaches in terms of pixel-wise segmentation accuracy while requiring much less computation time. Figure 1 below shows sample segmentation results. In the supplementary video[10], we provide more qualitative results. Source code is also publically available[11].

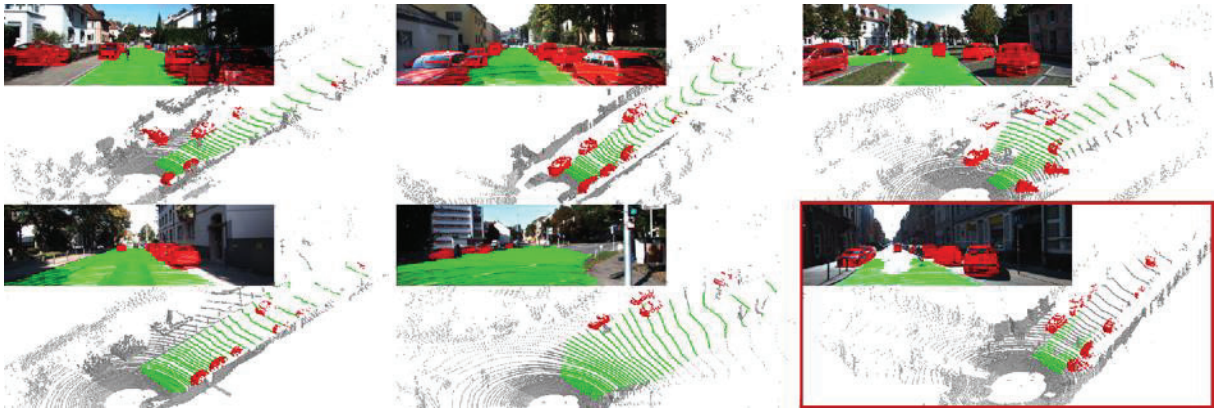


Figure 4 Sample qualitative results showing successes and failures of the proposed SalsaNet model using bird-eye-view [best view in color]. Note that the corresponding camera images on the top left are only for visualization purposes and have not been used in the training. The dark- and light-gray points in the point cloud represent points that are inside and outside the bird-eye-view region, respectively. The green and red points indicate road and vehicle segments.

In [3], HH has introduced a novel neural network architecture to perform uncertainty-aware semantic segmentation of a full 3D LiDAR point cloud in real-time. The proposed network is built upon the SalsaNet model [2], hence, named SalsaNext. The base SalsaNet model has an encoder-decoder skeleton where the encoder unit consists of a series of ResNet blocks and the decoder part upsamples and fuses features extracted in the residual blocks. In SalsaNext, our contributions lie in the following aspects:

---

[7] A Geiger, P Lenz, C Stiller, and R Urtasun. 2013. Vision meets robotics: The KITTI dataset. Int. J. Rob. Res. 32, 11 (September 2013), 1231–1237. DOI:https://doi.org/10.1177/0278364913491297

[8] M. Teichmann, M. Weber, M. Zllner, R. Cipolla, and R. Urtasun, "Multinet: Real-time joint semantic reasoning for autonomous driving," in IEEE Intelligent Vehicles Symposium, 2018, pp. 1013–1020

[9] K. He, G. Gkioxari, P. Doll´ar, and R. B. Girshick, "Mask R-CNN," CoRR, vol. abs/1703.06870, 2017. [Online]. Available: http://arxiv.org/abs/1703.06870

[10] https://www.youtube.com/watch?v=grKnW-uGIys

[11] https://gitlab.com/aksoyeren/salsanet

- To capture the global context information in the full 360-degree LiDAR scan, we introduce a new context module before encoder, which consists of a residual dilated convolution stack fusing receptive fields at various scales.
- To increase the receptive field, we replaced the ResNet block in the encoder with a novel combination of a set of dilated convolutions each of which has different kernel sizes.
- To avoid any checkerboard artifacts in the upsampling process, we replaced the transposed convolution layer in the SalsaNet decoder with a pixel-shuffle layer which directly leverages on the feature maps to upsample the input with less computation.
- To boost the roles of very basic features (e.g., edges and curves) in the segmentation process, the dropout treatment was altered by omitting the first and last network layers in the dropout process.
- To have a lighter model, average pooling was employed instead of having stride convolutions in the encoder.
- To enhance the segmentation accuracy by optimizing the Jaccard index, the weighted cross entropy loss in SalsaNet was combined with the Lovasz-Softmax loss.
- To further estimate the epistemic (model) and aleatoric (observation) uncertainties for each 3D LiDAR point, the deterministic SalsaNet model was transformed into a stochastic format by applying the Bayesian treatment.

All these contributions formed the new SalsaNext model which is the probabilistic derivation of the SalsaNet with a significantly better segmentation performance. The input of SalsaNext is the rasterized image of the full LiDAR scan in the panoramic view. The final network output is the point-wise classification scores together with uncertainty measures. Note that, to the best of our knowledge, this is the first work showing both epistemic and aleatoric uncertainty estimation on the LiDAR point cloud segmentation task.

HH has evaluated the performance of SalsaNext and compared with the state-of-the-art semantic segmentation methods on the large-scale challenging Semantic-KITTI[12] dataset, which provides over 43K LiDAR data. Obtained quantitative results compared to state-of-the-art pointwise and projection-based approaches are reported in [3]. The SalsaNext model considerably outperforms the others by leading to the highest mean IoU score (59.5%) which is +3.6% over the previous state-of-the-art method. In contrast to the original SalsaNet, we also obtain more than 14% improvement.

Figure 5 below shows sample qualitative segmentation and uncertainty results from SalsaNext. In this figure, only for visualization purposes, segmented object points are also projected back to the respective camera image. Note that these camera images have not been used for training of SalsaNext. As depicted in this figure, SalsaNext can, to a great extent, distinguish road, car, and other object points. In this figure, we additionally show the estimated epistemic and aleatoric uncertainty values projected on the camera image for the sake of clarity. We obtain high epistemic uncertainty for rare classes such as other ground. We also observe that high level of aleatoric uncertainty mainly appears around segment

[12] Behley, J., Garbade, M., Milioto, A., Quenzel, J., Behnke, S., Stachniss, C., and Gall, J. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In ICCV, 2019.

boundaries and on distant objects. In the supplementary video[13], we provide more qualitative results. Source code is also available[14].

Figure 5. Sample qualitative segmentation and uncertainty results from SalsaNext.

To address the sensor dropout problem (RQ2, M4b and M8), HH has implemented a deep generative neural network [4] based on SalsaNext [3]. This new generative model can predict dropped sensor readings (e.g., non-functional camera images) by employing other available sensor data (e.g., LiDAR point clouds). See Figure 3 for a sample panoramic color scene image generated from LiDAR point cloud.

More specifically, the work in [4] treats the sensor dropout problem as a multi modal domain translation. Domain translation can be considered a mapping of data samples from an input source domain to a different target domain. For instance, translating sketches to images or segmentation maps to images. The multi-modal domain translation such as synthesizing images from raw 3D point sets remains a challenge since point clouds, e.g., LiDAR scans, are sparse, unstructured, and nonuniformly sampled, which makes the mapping to the structured image space non-trivial.

In [4], we propose a novel multi-modal domain translation framework leveraging the underlying semantics of the perceived scene. Differently from existing works, we argue that mediating the translation between perceptually different sensor readings via semantic scene segments could ease the process to a great extent. More specifically, we propose a modular generative framework that can, for the first time, synthesize a panoramic color image from a full 3D LiDAR scan. See Figure 3 for example. The proposed framework is shown in Figure 4 and starts with SalsaNext [3] to semantically segment the point cloud. The same semantic segmentation is applied to the paired camera image by employing another state-of-the-art model: SDNet[15]. As our main technical contribution, we introduce a new conditional generative model, named TITANNet (generaTive domaIn TrANslation Network), which adversarially learns to translate the predicted LiDAR segment maps to the camera image counterparts. Finally, generated image segments are processed to render the panoramic scene images by a state-of-

[13] https://www.youtube.com/watch?v=MlSaIcD9ItU

[14] https://github.com/Halmstad-University/SalsaNext

[15] Andrew Tao, Karan Sapra, and Bryan Catanzaro. Hierarchical multi-scale attention for semantic segmentation. CoRR, 2020.

the-art model Vid2Vid-Net[16]. We provide extensive quantitative and qualitative evaluations of our framework. Obtained results on the Semantic-KITTI[17] dataset show that our framework outperforms all evaluated strong baselines by a large margin. We provide a video[18] showing the performance of TITANNet on the validation and test splits. The source code[19] is released for public use.

We here note that such multi-modal domain translation has practical uses for autonomous vehicles. Take an example of having a failure in the camera setup. The lack of a modality can severely impair the autonomous vehicle's performance since the subsequent sensor fusion and maneuver planning processes solely rely on these visual readings. Therefore, synthesizing photo-realistic images from other functioning modality readings, e.g., 3D LiDAR clouds, could help overcome a scenario of complete collapse. Another application could be generating additional annotated data in the source domain. By transferring the known labels across different domains, one can generate a new variation of the original scene from a different data distribution with no extra effort. We refer the reader to [4] for various images generated across different domains by TITANNet.

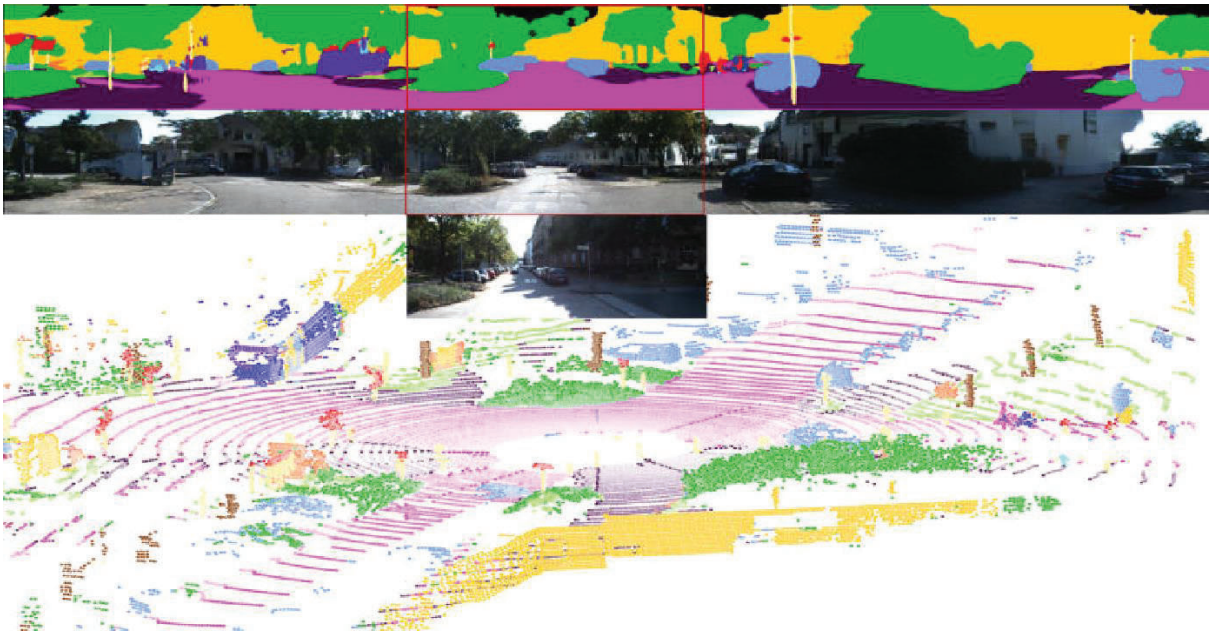


Figure 6 TITAN-Net is a modular generative neural network framework that receives a full 3D LiDAR point cloud and returns the panoramic color image by solely relying on the semantics of the scene. The framework first applies semantic segmentation to the full LiDAR scan (the bottom image). Next, a novel generative network translates the LiDAR segments to the camera semantic segments (the top image), which are then converted back to the panoramic color images (the second image from the top) by an additional generative model. The red frame indicates the region that the ground-truth camera image (the third image from the top) corresponds to. Our framework, for the first time, generates a 360-degree color image of the environment.

[16] T. Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, A. Tao, J. Kautz, and Bryan Catanzaro. Video-to-video synthesis In NeurIPS, 2018.

[17] Behley, J., Garbade, M., Milioto, A., Quenzel, J., Behnke, S., Stachniss, C., and Gall, J. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In ICCV, 2019.

[18] https://www.youtube.com/watch?v=eV510t29TAc

[19] https://github.com/Halmstad-University/TITAN-NET

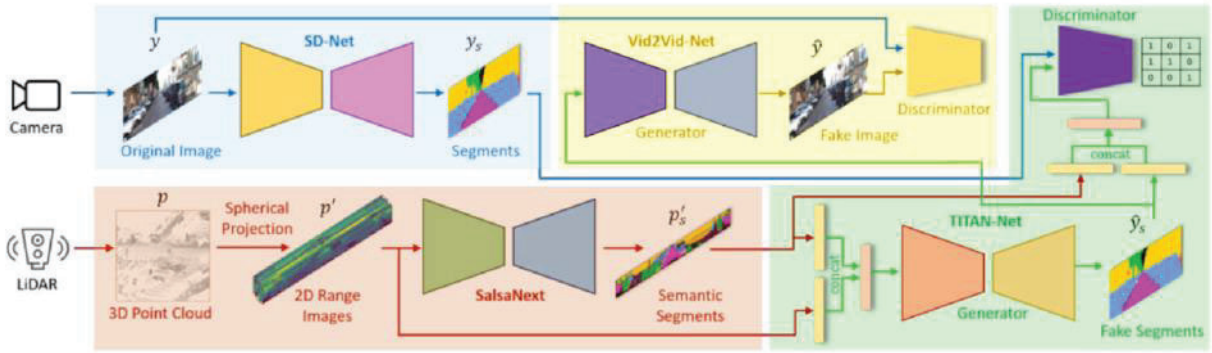


Figure 7. The proposed modular framework has four neural networks. Each is depicted by a unique background color. In the red box, a captured 3D LiDAR point cloud is first projected onto the 2D range image plane to be further processed by SalsaNext to predict semantic segments . Likewise, the corresponding RGB camera image is processed by SD-Net to predict semantic segment maps as depicted by the blue box. The green box highlights our proposed conditional GAN model TITAN-Net, where the Generator is conditioned on the concatenated and to generate the fake camera segment map. The TITAN-Net Discriminator is also conditioned on while comparing with the expected . Finally, as depicted in the yellow box, the fake segment is processed by Vid2Vid-Net to synthesize the realistic RGB image .

To particularly address RQ3 (*How to operate in tougher environmental conditions?),* we evaluated the performance of SalsaNext using a new dataset called WADS[20] (Winter Adverse Driving dataSet), which is the only publicly available dataset that contains labeled point cloud data of snowy weather conditions. WADS provides point-wise semantic labels (e.g., snow, vehicle, building, etc.) for each LiDAR scan. Figure 5 shows a sample point cloud from the WADS dataset. As showing in Figure 8, falling snow points introduce a large amount of noise in the LiDAR scan. Thus, we aimed at de-snowing LiDAR scans using SalsaNext. For this purpose, each LiDAR scan is first converted into binary labels, where white points indicate snow and green points represent every other non-snow points (see Figure 8). We retrained the SalsaNext model using the WADS dataset to detect falling snow points. A sample qualitative result is shown in Figure 8, where SalsaNext, to a large extent, detect and remove snow points.

Table 2 shows obtained quantitative results in contrast to statistical noise filtering algorithms such as DSOR[21] (Dynamic Statistical Outlier Removal), DROR[22] (Dynamic Radius Outlier Removal), and LiOR[23] (Low-intensity Outlier Removal). As Table 1 suggests, SalsaNext outperforms these classical methods by leading to the highest F1 score.

[20] https://digitalcommons.mtu.edu/wads/

[21] Kurup A. and Bos J., "Dsor: A scalable statistical filter for removing falling snow from lidar point clouds in severe winter weather." arXiv preprint arXiv:*2109.07078*, 2021.

[22] Nicholas Charron, Stephen Phillips, and Steven LWaslander. Denoising of lidar point clouds corrupted by snowfall. In *2018 15*th Conference on Computer and Robot Vision (CRV), pages 254–261. IEEE, 2018.

[23] Ji-Il Park, Jihyuk Park, and Kyung-Soo Kim. Fast and accurate desnowing algorithm for lidar point clouds. IEEE Access, 8:160202–160212, 2020

Filter Model	Recall	Precision	F1 score
DSOR	81.64	70.87	75.87
DROR	77.50	60.50	67.95
LIOR	92.51	65.65	76.79
LIOR&DROR	76.52	88.88	82.24
SalsaNext	90.32	91.98	91.14

Table 2. Obtained quantitative results on the WADS test split. All results are presented in %.

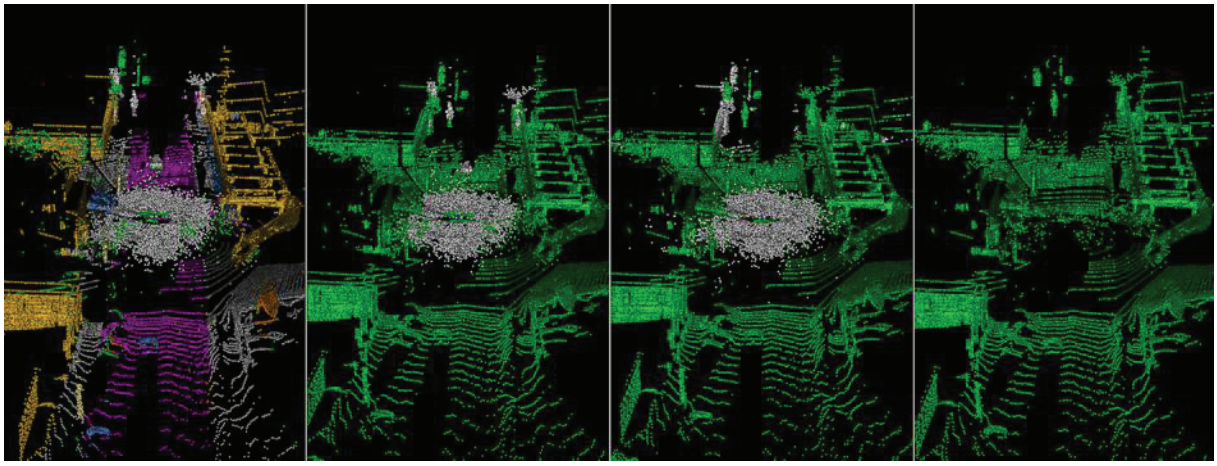


Figure 8 Detecting snow points in LiDAR point cloud data. On the left, a sample LiDAR scan with all the semantic labels is shown from the WADS dataset. In the second left image, the same LiDAR scan is shown with the binary labels where white points indicate snow and green represents the other object points. In the third image, the inference result from SalsaNext is shown. SalsaNext can successfully detect snow points. The very right image shows the LiDAR point cloud data with non-snow points, i.e., the one after removing the detected snow points.

6.3 Sensor Fusion

For the development of the sensor fusion we have used the state of the art object detection model YOLOv4 as a reference architecture. The general structure of YOLOv4 can be summarized by the figure below.

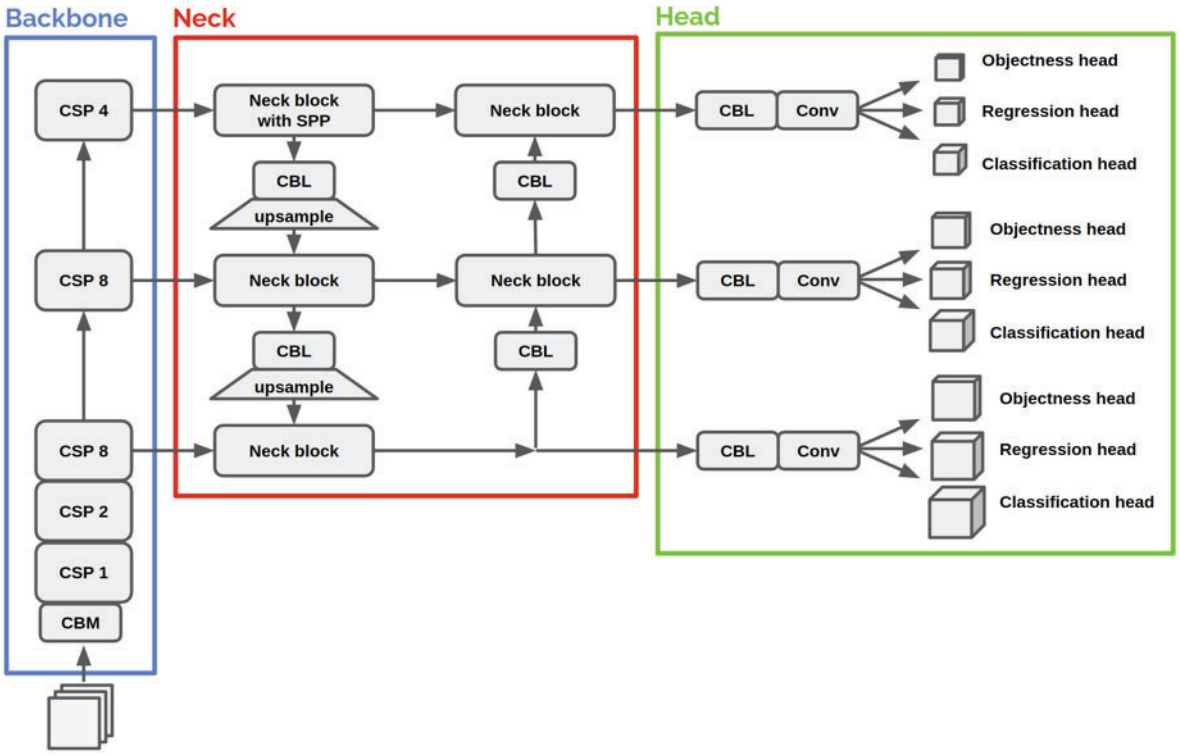


Figure 9. High-level description of the YOLOv4 model architecture.

The "CSP…" blocks are the stages from the CSPDarknet53 backbone, with each block halving the image resolution and doubling the number of channels, and where the number following "CSP" indicates the number of repeated residual units. CBM is a Conv -> BatchNorm -> Mish sequence and CBL is Conv -> BatchNorm -> LeakyRelu. The "upsample" layer is a nearest-neighbor upsampling operation that doubles the image resolution. It is used to enable the concatenation of layers from higher up in the figure with layers coming from the left, when entering a "Neck block". Similarly, the two Neck blocks to the right in the figure have inputs coming from below. These go through CBL blocks with stride 2 to enable them to be concatenated with the layers coming from the left. This neck structure with an upsample pass and a "downsample" pass is called a PAN neck[24].
Finally, the "Head" section processes the three feature maps of different scales from the neck individually, outputting an objectness, regression and classification map for each scale. The top heads in the figure have a smaller image resolution, and are used to predict larger objects in the scene, whereas the lower heads have a larger image resolution and are used to predict smaller objects.

[24] Liu, S., Qi, L., Qin, H., Shi, J., & Jia, J. (2018). Path aggregation network for instance segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 8759-8768).

Chicago

Input fusion
The simplest way to fuse inputs is to combine them before feeding them to the network. In our case we concatenate the images, in the channel dimension. For example, to fuse the left and right stereo camera images, we stack the two sets of RGB channels and get a combined 6 channel image. The only modification required of the network is thus to change the input channels of the first conv layer.

Multi-scale backbone fusion
Next, we explore a network where we have one individual backbone for each input, which we then fuse before feeding into a shared neck and head. We do this fusion at each of the three scales that are fed to the neck, by adding the output of each of the backbone feature maps at that scale.

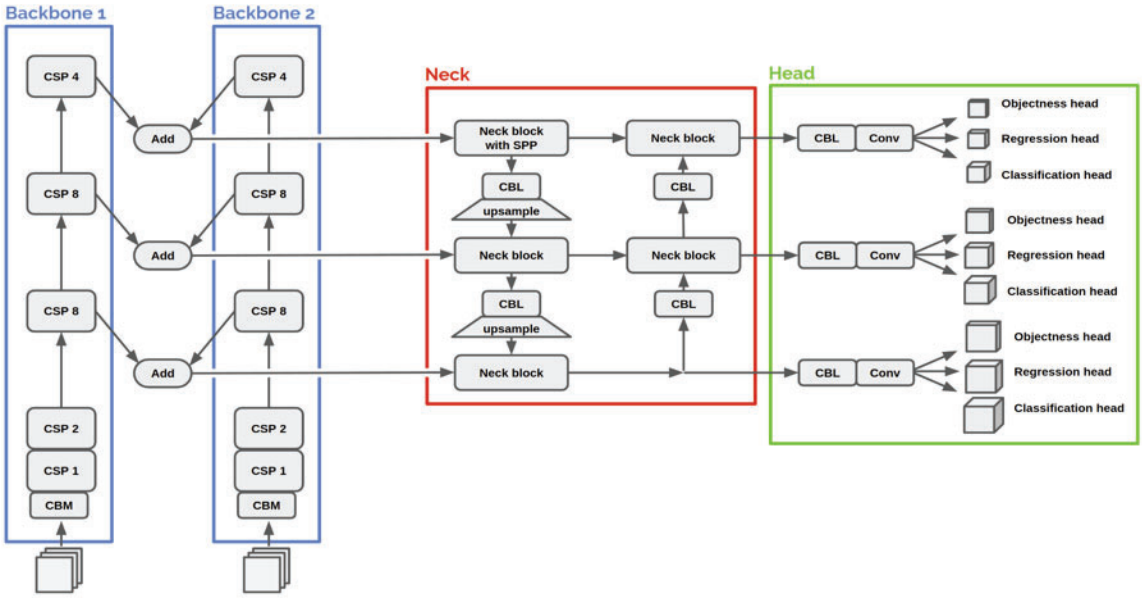


Figure 10. Description of the multi-scale backbone fusion applied on YOLOv4.

Multi-scale neck fusion
In the next variant, we include the neck in the individual parts, so that the only shared part is the "Head" part.

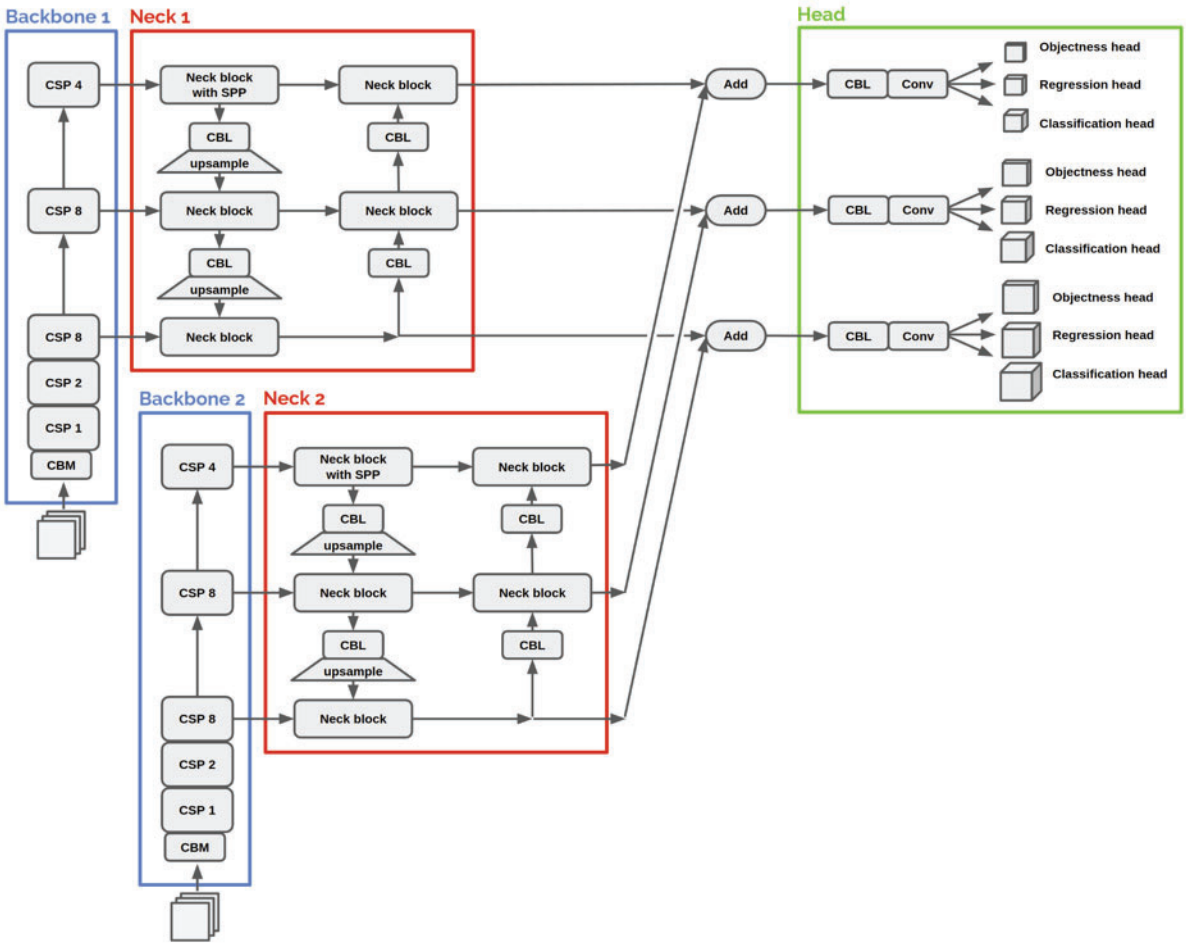


Figure 11. Description of the multi-scale neck fusion applied on YOLOv4.

Multi-scale multi-depth fusion
Finally, in the last variant we allow the network to choose to share information at multiple points in the two backbone-neck parts, in addition to fusing at the necks as in the previous variant. The learned sharing of information within the backbone-necks is implemented using cross fusion[25], denoted as "Weighted sum" in the figure. Each "Weighted sum" node contains two learnable weights, one for each of the two inputs.

[25] Caltagirone, L., Bellone, M., Svensson, L., & Wahde, M. (2019). LIDAR–camera fusion for road detection using fully convolutional neural networks. Robotics and Autonomous Systems, 111, 125-131.

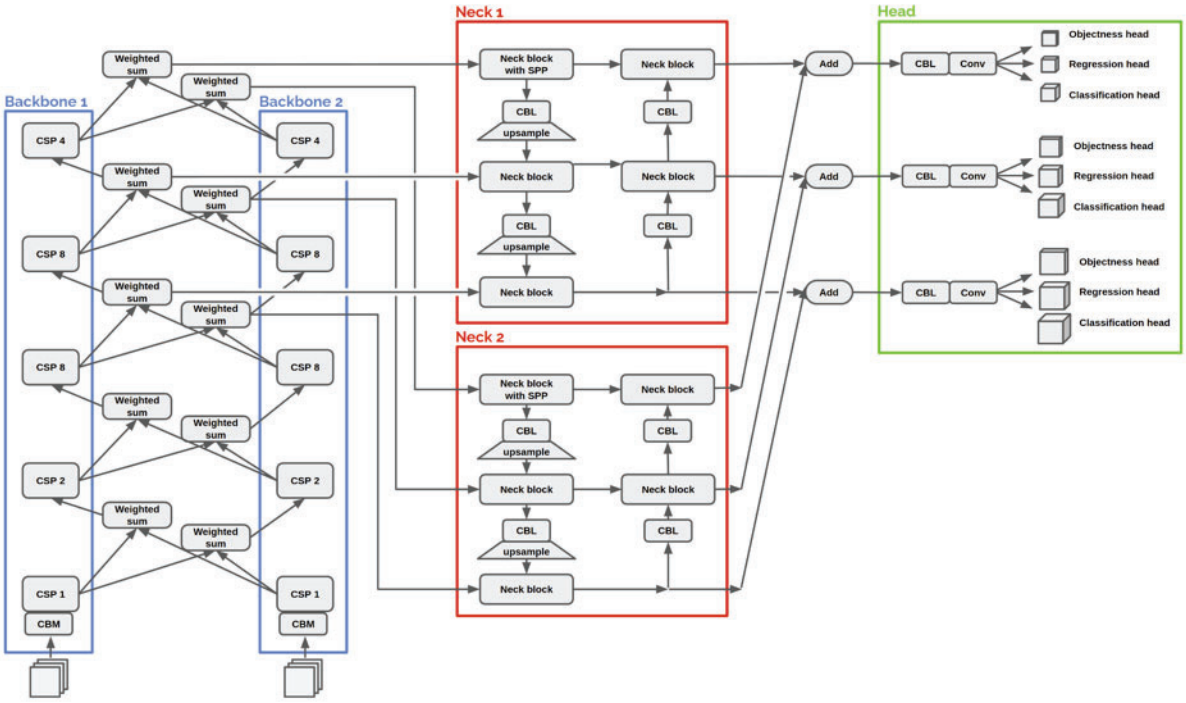


Figure 12. Description of the multi-scale multi-depth fusion applied on YOLOv4.

Fused sensor modalities

In this project, three input types were explored in the fusion experiments: the original (left eye) color image, an additional (right eye) image, and an image with point cloud data projected to the same view as the left eye image, yielding a sparse single-channel depth map. The pixels between the sparse points are then given the value of the closest projected point within a 6 pixel distance, or left empty if the closest point is farther away than that. Figure 13 shows an example of the three respective inputs, annotated with the bounding boxes of the left-eye image.

Different combinations of these inputs were explored in the fusion experiments, as well as a special case where the 1xRGB mono image was combined with itself, which we denote *2xRGB (double-mono)* in the table. This was introduced as a special case to see whether any improvement in the *2xRGB (stereo)* case came from the model utilizing the stereo information or was simply due to the extra computation capacity introduced by the fusion methods.

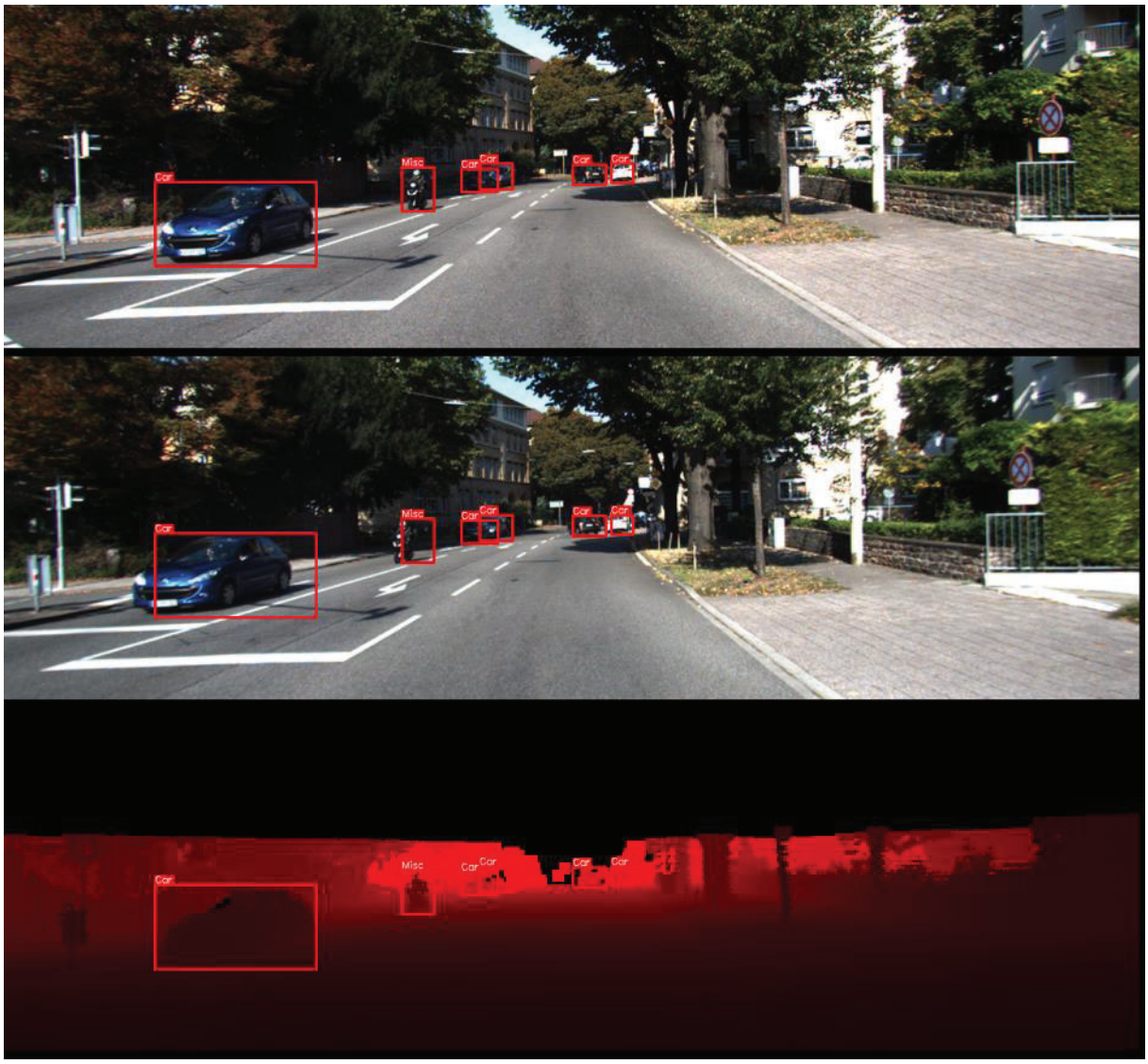


Figure 13. Example of an annotated sample from the dataset: the mono/left stereo image (top), the right stereo image (center), and the depth image (bottom).

Fusion results
The results for the different fusion methods are summarized in Table 3.

Sensors	Fusion	Car			Pedestrian			Cyclist			Average
		Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard	
1xRGB		95.3	89.1	87.3	87.4	82.1	76.8	72.0	59.4	55.7	78.3
2xRGB (stereo)	input	92.9	87.5	85.5	85.6	79.6	74.9	67.7	52.5	50.4	75.2
	backbone	95.4	88.9	87.2	86.8	81.4	76.4	69.2	57.2	53.8	77.4
	neck	97.8	90.2	88.3	90.0	83.2	77.3	71.5	61.5	57.8	79.7
	multi-depth	94.7	88.3	87.1	88.8	82.6	75.6	74.6	62.6	60.3	79.4
2xRGB (double-mono)	backbone	97.6	90.1	88.2	86.3	80.2	74.7	72.3	59.0	55.1	78.2
	neck	95.1	89.2	87.1	87.1	80.3	73.0	70.2	59.2	55.5	77.4
	multi-depth	95.3	89.0	87.2	87.8	81.8	75.2	68.2	56.6	53.6	77.2
1xRGB + lidar	input	95.0	89.1	88.2	86.1	80.7	75.3	70.4	59.4	56.6	77.9
	backbone	95.1	90.0	89.0	89.6	84.0	79.7	74.3	59.9	56.9	79.8
	neck	96.5	90.0	88.5	91.3	83.8	79.0	78.0	63.2	59.7	81.1
	multi-depth	97.3	89.6	88.3	91.7	84.1	79.7	71.1	59.0	55.7	79.6
2xRGB + lidar	input	95.0	90.4	88.5	85.6	81.2	77.6	64.0	52.8	50.4	76.2
	backbone	96.4	90.9	88.8	90.3	84.9	80.9	72.8	56.6	53.2	79.4
	neck	59.4	70.1	71.5	91.4	84.7	80.1	74.1	62.3	59.7	72.6
	multi-depth	97.4	89.8	88.2	90.7	84.0	79.7	70.6	58.5	55.0	79.3

Table 3. Results for the sensor fusion experiments.

We can see that in both the stereo and lidar cases the neck fusion method gave the best performance, and that the best model overall was the **1xRGB + lidar** neck fusion model. One possible explanation for the superiority of doing late fusion rather than input fusion is that the model is encouraged to process the lidar input into something useful since it doesn't initially share its computational resources between the lidar and the color parts.

6.4 Model Compression

Model compression is very important to be able to run real-time perception systems in vehicles due to the limited available computational resources. Both HH and Embedl have made progress in this area during the project.

In [1], HH has introduced a generic deep neural network compression method that is network agnostic (i.e., not dependent on a specific neural network architecture) and has minimal impacts on accuracy, while reducing the inference time and memory footprint of a network. This work is important as deep neural networks have been notorious for being computational expensive. It is because neural networks

are often over-parametrized and most likely have redundant nodes or layers as they are getting deeper and wider. Their demand for hardware resources prohibits their extensive use in embedded devices and puts restrictions on tasks like real time image classification or object detection. Therefore, network compression plays an important role in autonomous driving.

The proposed method in [1] has two stages: pruning and quantization. In the pruning phase, low importance parameters are first removed and then the network is retrained for a small number of epochs. This first stage is repeated until reaching a threshold accuracy drop. The following quantization stage rather projects network parameters into forms that enable more computationally efficient arithmetic by, for instance, migrating from float to integer representation. These two consecutive compression operations reduce the number of required computations while also lowering the memory footprint of deep networks. The proposed model compression method was experimentally evaluated on different image classification datasets and object detection tasks.

In classification networks, the proposed framework reached up to 95% pruning of network parameters. In relatively more complicated object detection networks, the number of model parameters were reduced up to 59.70% without sacrificing much in accuracy. With this proposed method, the proposed framework achieved up to 182x and 110x memory compressions (i.e., reduction rates in memory required for network parameters) for the classification and detection networks, respectively.

Table I shows the performances of five different well known model architectures, three image classifiers and two object detectors at different stages of the proposed model compression pipeline. The column named *Initial* presents the accuracy or the mean average precision (mAP) of the architecture using pre-trained weights before the model compression pipeline is applied. The *Pruned* column indicates the accuracy or mAP after the pruning step of the pipeline. The *Pruned & Quantized* column presents the final accuracy or mAP after the two-stage model compression procedure has been applied. The very last column, called *Percentage Pruned*, indicates the total percentage of the network parameters that were removed.

Along the lines of network compression, HH has also introduced a Multi-Objective Hardware-Aware Quantization (MOHAQ) method [5], which considers hardware efficiency and inference error as objectives for mixed-precision quantization.

Architecture (metric)	Initial	Pruned	Pruned & Quantized	Percentage Pruned
MNIST Classifier (accuracy)	0.990	0.993	0.993	95.00
CIFAR10 Classifier (accuracy)	0.917	0.899	0.898	95.00
ResNet50 Classifier (accuracy)	0.834	0.825	0.747	45.00
YOLOv3 Detector (mAP)	0.589	0.537	0.530	59.70
FasterRCNN Detector (mAP)	0.677	0.662	0.612	52.30

Table 4. Obtained quantitative results on model compression on different tasks.

Embedl used the object detection part of the KITTI dataset as the real dataset, containing 7481 stereo images with bounding box annotations and lidar point clouds. The test split of TensorFlow Datasets[26] was followed, which ensures that images in different splits do not come from the same video sequence.

[26] https://www.tensorflow.org/datasets/catalog/kitti

As base model, the YOLOv4 object detection model[27] was used, but with an input size of 1248x384 to be able to keep the original pixels of the KITTI images without having to resize the image to a square.

All experiments follow the YOLOv4 training setup with a cosine learning rate schedule, and hue/saturation/value and mosaic augmentations, but some parameters were adjusted after a manual hyperparameter search to be suitable for KITTI. The KITTI experiments were trained for 30 epochs with an Adam optimizer, with a max learning rate of 0.0005 and a warmup of 10% of the training time. The original weight decay of 0.0005 and momentum of 0.937 were kept, and the models were trained with a batch size of 4 in order to be able to use the same batch size for even the largest fusion models. In the fusion experiments, the hue/saturation/value augmentation was only applied on the RGB inputs.

The Nvidia Jetson Xavier AGX was used as the hardware platform for the deployment of the neural network. Nvidia provides a tool, TensorRT[28], for compiling neural networks from various binary formats, e.g. Tensorflow protobuf file as well as ONNX[29] the industry standard for model interoperability, to machine code ready to be used by the embedded Xavier platform. Nvidia's command line tool trtexec, shipped with TensorRT, was used for measuring the latency on the platform.

In the compression experiments, we compare Embedl's hardware-aware pruning methods to a *structured magnitude pruning* baseline method. These experiments are run on the original 1xRGB model.

Structured magnitude pruning (baseline method)
In magnitude pruning, the magnitude of the weights are used as a proxy for their importance, where the argument is that removing the weights with the smallest magnitude should have the least negative impact on the accuracy of the network. It has been observed that networks tend to be overparameterized such that a large proportion of weights can be removed (set to zero) without significantly harming the accuracy. Sparse magnitude pruning is however not guaranteed to reduce the latency on most modern accelerators, since they cannot make use of the sparse structure of the data. Instead, in order to get speed improvements it is often necessary to do *structured pruning*, where you for example prune all weights for an entire channel such that the channel can actually be *removed* instead of being zeroed out. In structured magnitude pruning, we thus do not consider each weight magnitude individually, but instead look at the sum of the weight magnitudes for an entire channel.

While structured pruning solves the problem of accelerators not being able to utilize sparse weights, there are other aspects of accelerators that are ignored when pruning simply based on magnitude: the removal of channels of an operation will have different effects on the latency of the model depending on factors such as the input size and number of channels the operation currently has. There might be staircase effects where a channel reduction does not provide any latency reduction until it reaches a certain granularity level, such as steps of 32. And some parts of the accelerator might only work for specific combinations of channel and input sizes, yielding complicated patterns of extra beneficial regions. These intricacies also vary significantly between different hardwares.

Embedl has developed methods that take these effects of the hardware into account when looking for ways to optimally compress the neural network. Embedl has developed different hardware-aware compression methods. In this report, three versions of Embedl's compression are evaluated. The compression results are summarized in the Figure 14 and Table 5 below.

[27] Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934.
[28] https://developer.nvidia.com/tensorrt
[29] https://onnx.ai/

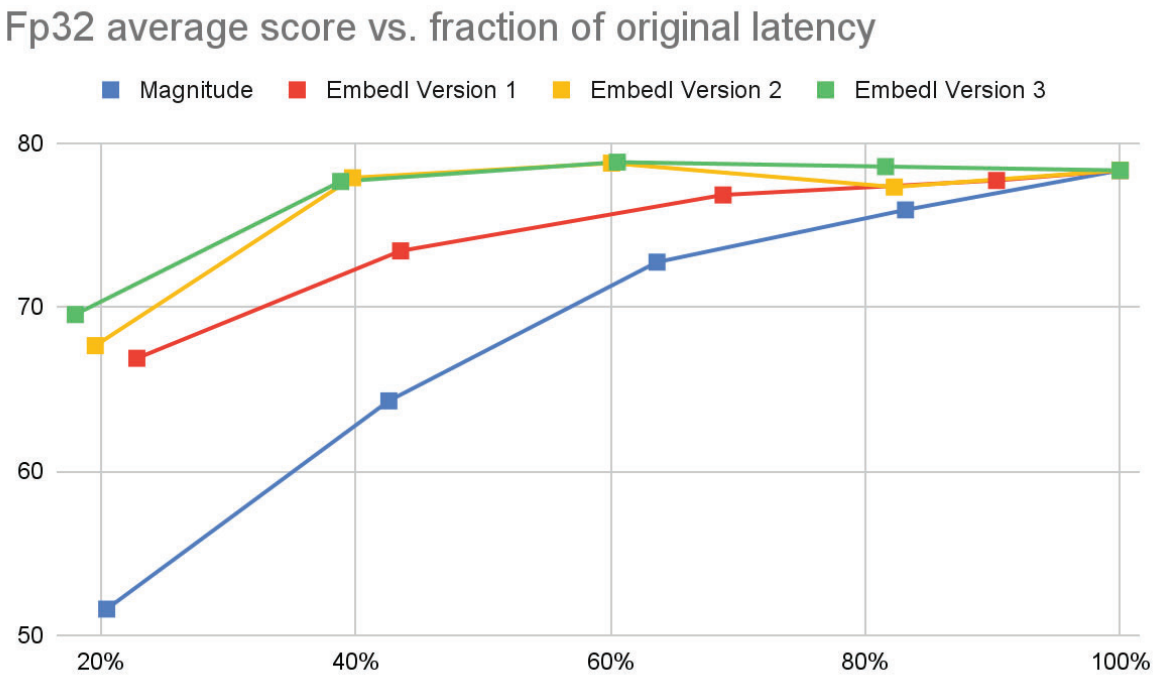


Figure 14. Comparison of the results for the four different compression methods. Embedl's methods clearly outperforms the baseline method.

Sensors	Fusion	Car			Pedestrian			Cyclist			Average
		Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard	
1xRGB		95.3	89.1	87.3	87.4	82.1	76.8	72.0	59.4	55.7	78.3
2xRGB (stereo)	input	92.9	87.5	85.5	85.6	79.6	74.9	67.7	52.5	50.4	75.2
	backbone	95.4	88.9	87.2	86.8	81.4	76.4	69.2	57.2	53.8	77.4
	neck	97.8	90.2	88.3	90.0	83.2	77.3	71.5	61.5	57.8	79.7
	multi-depth	94.7	88.3	87.1	88.8	82.6	75.6	74.6	62.6	60.3	79.4
2xRGB (double-mono)	backbone	97.6	90.1	88.2	86.3	80.2	74.7	72.3	59.0	55.1	78.2
	neck	95.1	89.2	87.1	87.1	80.3	73.0	70.2	59.2	55.5	77.4
	multi-depth	95.3	89.0	87.2	87.8	81.8	75.2	68.2	56.6	53.6	77.2
1xRGB + lidar	input	95.0	89.1	88.2	86.1	80.7	75.3	70.4	59.4	56.6	77.9
	backbone	95.1	90.0	89.0	89.6	84.0	79.7	74.3	59.9	56.9	79.8
	neck	96.5	90.0	88.5	91.3	83.8	79.0	78.0	63.2	59.7	81.1
	multi-depth	97.3	89.6	88.3	91.7	84.1	79.7	71.1	59.0	55.7	79.6
2xRGB + lidar	input	95.0	90.4	88.5	85.6	81.2	77.6	64.0	52.8	50.4	76.2
	backbone	96.4	90.9	88.8	90.3	84.9	80.9	72.8	56.6	53.2	79.4
	neck	59.4	70.1	71.5	91.4	84.7	80.1	74.1	62.3	59.7	72.6
	multi-depth	97.4	89.8	88.2	90.7	84.0	79.7	70.6	58.5	55.0	79.3

Table 5. Results for Embedl's different compression methods compared with the baseline.

We can see that Embedl's hardware aware methods outperform the magnitude pruning method, with the best method being Embedl Version 3. This method maintains most of the accuracy for all pruning ratios except for the most aggressive ratio where only 5.7% of the flops are left.

The image below shows an example of applying the most aggressively pruned model on an image sequence collected by a camera-mounted car in Gothenburg.

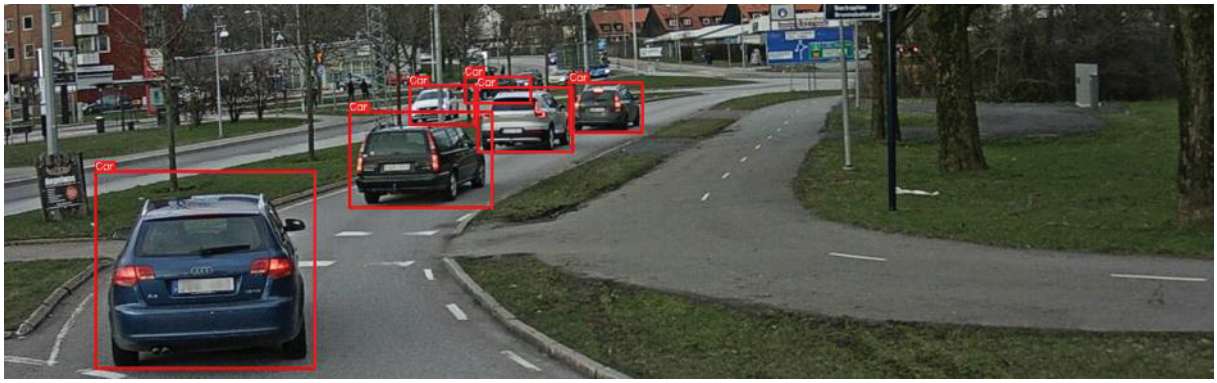


Figure 15. Demo of object detection compression running on target hardware platform. Image from real traffic in Gothenburg.

6.5 Summary and Goal Fulfillment

In the application, four research questions (RQ1-4) and two quantifiable objectives (G1-2) were formulated.

RQ1 How to fuse multi-sensory information?
We have shown that late fusion gives the best result when using several RGB cameras and a lidar sensor.

RQ2. How to handle different sensor failures?
We have developed technology to use machine learning to generate data from the faulty sensor.

RQ3. How to operate in tougher environmental conditions?
We have developed a method to be able to remove noise caused by snow in lidar data. To develop more technology around this research question, we would have needed the dataset in the original project plan. However, there was no possibility for this after AB Volvo reduced its involvement in the project due to the pandemic. However, we will carry out this work in an approved and started EU-funded continuation project, RoadView.

RQ4. How does challenging conditions affect sensor output and how should this most efficiently be replicated in a simulation environment for realistic synthetic data generation?
By studying sensor data in harsh weather and operational conditions, functionality was developed to mimic these effects in synthetic data generated in a 3D graphics engine (Unreal Engine).

G1. Robust perception under tough environmental conditions
To quantitatively measure whether this goal has been achieved, the dataset that was not developed is needed, see RQ3 above. However, we have developed promising technology. This will be evaluated and improved in the RoadView project.

G2. Scalable and Realistic Data generation
To reach the goal of obtaining at least 75% of the original performance, only 0.5% of the real data set (32 images) was needed. With the synthetic data and 10% of the real data set, the original performance could be obtained, i.e. 90% saving in annotation.

7. Dissemination and Publications

7.1 Dissemination and exploitation activities

Hur har/planeras projektresultatet att användas och spridas?	Markera med X	Kommentar
Öka kunskapen inom området	X	
Föras vidare till andra avancerade tekniska utvecklingsprojekt	X	
Föras vidare till produktutvecklingsprojekt	X	
Introduceras på marknaden	X	
Användas i utredningar/regelverk/ tillståndsärenden/ politiska beslut		

Open-source Codes:

- https://gitlab.com/aksoyeren/salsanet
- https://github.com/Halmstad-University/SalsaNext
- https://github.com/Halmstad-University/TITAN-NET

PhD:

- The PhD work of Tiago Cortinhal from Halmstad University is ***fully supported*** by SHAPREN.
- The PhD work of Nesma Rezk from Halmstad University is ***partially supported*** by SHARPEN.

Awards:

- Tiago Cortinhal, a PhD student at Halmstad University, received the *"Best Student Paper Award"* at the IJCAI 2021 Workshop on Artificial Intelligence for Autonomous Driving.

Invited Talks:

- Workshop on Bridging the gap between map-based and map-less driving, IEEE Intelligent Vehicles Symposium (IV2022), USA, October 2022
- A seminar on "Semantics-aware Multi-modal Domain Translation" at IAM, Arizona Commerce Authority, Arizona USA, 2021
- Workshop on Data Driven Intelligent Vehicle Applications, IEEE Intelligent Vehicles Symposium (IV2020), USA, October 2020

- Workshop on synthetic data for AI Sweden's partners, Gothenburg Sweden, September 2021

Projects:

- ROADVIEW – Robust Automated Driving in Extreme Weather: The SHARPEN beneficiaries HH and SDS have received a **Horizon Europe grant** of 7.5 million euros for the ROADVIEW project. ROADVIEW can be considered as the extension of SHARPEN. Halmstad University is the main project coordinator of the consortium of 15 partners. ROADVIEW integrates a complex in-vehicle system-of-systems able to perform advanced environment and traffic recognition and prediction and determine the appropriate course of action of a CAV in a real-world environment, including harsh weather conditions. The project develops an embedded in-vehicle perception and decision-making system based on enhanced sensing, localization, and improved object/person classification (including vulnerable road users). Its ground-breaking innovations are grounded on a cost-effective multisensory setup, sensor noise modelling and filtering, collaborative perception, testing by simulation-assisted methods and integration and demonstration under different scenarios and weather conditions.

7.2 Publications

[1] Tzelepis G., Asif A., Baci S., Cavdar S., and Aksoy E. E., "Deep Neural Network Compression for Image Classification and Object Detection," 2019 18th IEEE International Conference on Machine Learning and Applications (ICMLA), 2019, pp. 1621-1628, doi: 10.1109/ICMLA.2019.00266.

[2] Aksoy E.E., Baci S., and Cavdar S.. "SalsaNet: Fast Road and Vehicle Segmentation in LiDAR Point Clouds for Autonomous Driving", IEEE IVS 2020, [Online]. Available: http://arxiv.org/abs/1909.08291

[3] Cortinhal T., Tzelepis G., and Aksoy E.E. "SalsaNext: Fast, Uncertainty-aware Semantic Segmentation of LiDAR Point Clouds for Autonomous Driving", ISVC 2020, [Online]. Available: https://arxiv.org/pdf/2003.03653.pdf

[4] Cortinhal T., Kurnaz F., and Aksoy E.E., "Semantics-aware multi-modal domain translation: From lidar point clouds to panoramic color images", Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops, 2021, pp. 3032-3048

[5] Rezk N., Nordstrom T., Stathis D., Ul-abdin Z., Aksoy E.E., Hemani A., "MOHAQ: Multi-Objective Hardware-Aware Quantization of Recurrent Neural Networks", 2021, arXiv preprint https://arxiv.org/abs/2108.01192

[6] Englund, C.; Aksoy, E.E.; Alonso-Fernandez, F.; Cooney, M.D.; Pashami, S.; Åstrand, B., "AI Perspectives in Smart Cities and Communities to Enable Road Vehicle Automation and Smart Traffic Control.", Smart Cities 2021, 4, 783-802. https://doi.org/10.3390/smartcities4020040

[7] Cooney M., Orand A., Larsson H., Pihl J., and Aksoy E. E., "Exercising with an "Iron Man": Design for a Robot Exercise Coach for Persons with Dementia," 2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN), 2020, pp. 899-905, doi: 10.1109/RO-MAN47096.2020.9223552.

8. Conclusions and Continued Studies

We have shown that pretraining on the synthetic dataset developed by MIS, can significantly reduce the need for annotated data - saving significant cost in deep learning R&D projects.

We have shown that Embedl's compression methods can be successfully applied in the automotive use case to significantly reduce the latency while maintaining most of the original accuracy, allowing for much more aggressive compression than the magnitude pruning baseline.

We have also explored multi-scale fusion methods particularly designed for multi-scale object detection architectures like YOLOv4. Doing fusion later in the model rather than at the input seems to sometimes be necessary in order for an added sensor to be utilized. One possible explanation could be that the added input is too complicated to be utilized directly, and is ignored in favor of more readily interpretable sensors. While this project has focused on 2d bounding box detection, an interesting future research area would be to explore similar fusion methods for 3d bounding box detection, where the stereo and lidar information is likely to be more useful, and where we might accordingly expect the fusion to provide a larger improvement.

9. Partners and contact details

The project participants and their roles are **Embedl AB** (coordinator), **Halmstad University** , **Volvo GTT, Volvo Construction Equipment** and **Machine Intelligence Sweden AB**.

Partner	Contact	Logotype
Embedl AB (Coordinator)	Hans Salomonsson hans@embedl.com +46730632837	
Halmstad Högskola	Eren Erdal Aksoy eren.aksoy@hh.se	
Machine Intelligence Sweden AB	Devdatt Dubhashi devdatt.dubhashi@machineintelligence.se	

Volvo Group Truck Technology	Magnus Alin magnus.alin@volvo.com	
Volvo Construction Equipment	Andreas Hjertstrom andreas.hjertstrom@volvo.com	