# INFLUENCE OF DOMAIN SHIFT FACTORS ON DEEP SEGMENTATION OF THE DRIVABLE PATH OF AN AUTONOMOUS VEHICLE

R. P. A. Bormans[a], R. C. Lindenbergh[a], F. Karimi Nejadasl[b],

[a] Dept. of Geoscience and Remote Sensing, Delft University of Technology, The Netherlands
- r.p.a.bormans@student.tudelft.nl & r.c.lindenbergh@tudelft.nl
[b] Robot Care System, The Hague, The Netherlands - f.kariminejadasl@robotcaresystems.com

**Commission II, WG II/6**

**KEY WORDS:** LiDAR, Computer Vision, Self-driving cars, Weakly-supervised learning, Convolutional Neural Network, Domain adaptation

**ABSTRACT:**

One of the biggest challenges for an autonomous vehicle (and hence the WEpod) is to see the world as humans would see it. This understanding is the base for a successful and reliable future of autonomous vehicles. Real-world data and semantic segmentation generally are used to achieve full understanding of its surroundings. However, deploying a pretrained segmentation network to a new, previously unseen domain will not attain similar performance as it would on the domain where it is trained on due to the differences between the domains. Although research is done concerning the mitigation of this domain shift, the factors that cause these differences are not yet fully explored. We filled this gap with the investigation of several factors. A base network was created by a two-step fine-tuning procedure on a convolutional neural network (SegNet) which is pretrained on CityScapes (a dataset for semantic segmentation). The first tuning step is based on RobotCar (road scenery dataset recorded in Oxford, UK) while afterwards this network is fine-tuned for a second time but now on the KITTI (road scenery dataset recorded in Germany) dataset. With this base, experiments are used to obtain the importance of factors such as horizon line, colour and training order for a successful domain adaptation. In this case the domain adaptation is from the KITTI and RobotCar domain to the WEpod domain. For evaluation, groundtruth labels are created in a weakly-supervised setting. Negative influence was obtained for training on greyscale images instead of RGB images. This resulted in drops of IoU values up to 23.9% for WEpod test images. The training order is a main contributor for domain adaptation with an increase in IoU of 4.7%. This shows that the target domain (WEpod) is more closely related to RobotCar than to KITTI.

## 1. INTRODUCTION

For a WEpod or a self-driving vehicle in general to safely navigate over the road, it needs to *understand* road scenes that appear in our daily life. The WEpod is an autonomous shuttle (figure 1) and is able to transfer up to six people. As most autonomous vehicles it is equipped with camera, LiDAR and RaDAR sensors. One common way to achieve this awareness of the environment, is to use semantic segmentation. Semantic segmentation is the assignment of each pixel of an image to a semantically meaningful class. A simple reconstruction of the environment can be achieved by identifying three classes: occupancies, drivable path and unknown area.


Figure 1: WUrbie: one of the two WEpods in the Netherlands.

Detecting obstacles is a critical aspect for realising autonomous driving. Static obstacles such as buildings and trees, as well as dynamic obstacles such as other traffic participants have to be detected with great accuracy in order to avoid accidents. A possible trajectory which the vehicle can follow is called the drivable path. This path can play an important role for in-lane localisation. In order to determine this path and avoid accidents, obstacle sensing plays an important role. From the aforementioned definition of drivable path it can be concluded that this term is not necessarily bounded to one ”solution”. Intersections can be thought of as an example of a combination of solutions. Often there will be locations which are neither a drivable path nor an obstacle. Typically these areas correspond to the road area outside the lane which is occupied by the vehicle (including lanes for oncoming traffic), curbstones, empty pavements and ditches (Barnes et al., 2017). It is important to mention the difference with free space, since free space is defined as the space where a vehicle can move freely without colliding with other objects (Lundquist et al., 2009). Hence, unknown area does not represent the same volume as free space although free space often is a part of unknown area.

Initially, no large amount of sensor data was available for the WEpod. Therefore use is made of already available (external) datasets. Weak labels are created for a subset of two large road scenery datasets, KITTI and RobotCar. These datasets not only contain the recorded image sequences but also laser, GPS and IMU data. The created labels are not perfect by means of clear boundaries for all three categories (obstacles, drivable path and unknown area). The quality of these labels is to a certain extent dependent on the sensor quality of the recording platform (both camera and LiDAR). By treating these labels as groundtruth, it is

possible to produce a vast amount of labels which will enable us to create a (large) set of training images.

Convolutional Neural Networks (CNN) have become a dominant player in the world of computer vision during recent years. With this emerging field, a large number of different CNN architectures are available today. When a CNN is trained on a certain source domain (e.g. CityScapes, a dataset for semantic urban scene understanding) and then deployed on a different (target) domain, the network will often execute the task (e.g. segmentation) poorly because of the differences between target and source domain (i.e. the domain shift). This limited ability of a CNN to adapt itself to new domains is a common problem of transfer learning. For our implementation the goal of transfer learning is to transfer the ”knowledge” of the pretrained CNN towards the specific task of segmenting images into the three aforementioned classes. In general transfer learning can be very useful since it limits the required amount of training data and computational time needed to successfully train a CNN. Factors that influence the success of domain adaptation are identified and it is shown how they influence the result. The goal of this paper is to obtain an idea if and how several factors influence the domain shift from the KITTI and RobotCar domain towards the WEpod domain.

The rest of this paper is organised such that section 2. will sum up the most important and/or recent research on CNNs, semantic segmentation and domain adaptation. Section 3. will walk through different aspects of the workflow such as the datasets, labelling technique and the used CNN architecture. All performed experiments are described in section 4. while the results of these experiments with a small discussion can be found in section 5.. The conclusion in section 6. will complete the paper.

## 2. RELATED WORK

Convolutional Neural Networks have proven to achieve impressive results on a wide range of computer vision tasks, such as semantic segmentation (Long et al., 2015) and object recognition (He et al., 2016). Except for new architectures, improvements of already existing networks have been examined via dilated convolution (Yu and Koltun, 2015) and conditional random fields (Chen et al., 2018). Since interest in semantic segmentation increases due to the diverse applications such as autonomous driving, development on CNNs for semantic segmentation is a dynamic area of research. This results in continuously increasing state-of-the-art results such as reported in (Zhao et al., 2017) and in (Huang et al., 2017).

Semantic segmentation is an important tool for several applications because it is enables the understanding of a scene based on an image. However, because fine annotation and quality control for one single image will take up to 1.5 hour (Cordts et al., 2016), most datasets do not have a comprehensive groundtruth set which results in usage of weakly- or semi-supervised labels to boost performance of semantic segmentation. (Pathak et al., 2014) approached this as multiple instance learning and (Papandreou et al., 2015) developed expectation-maximisation methods for semantic segmentation under weakly-supervised and semi-supervised training setups. (Hong et al., 2016) make use of auxiliary weak annotations from semantic segmentation for different categories to assist segmentation of images with only image-level class labels.

All methods, use annotations in both source and target domain except for (Hoffman et al., 2016), who use strong supervision in the source domain but no supervision in the target domain. Our work considers a combination of strong and weakly-supervised

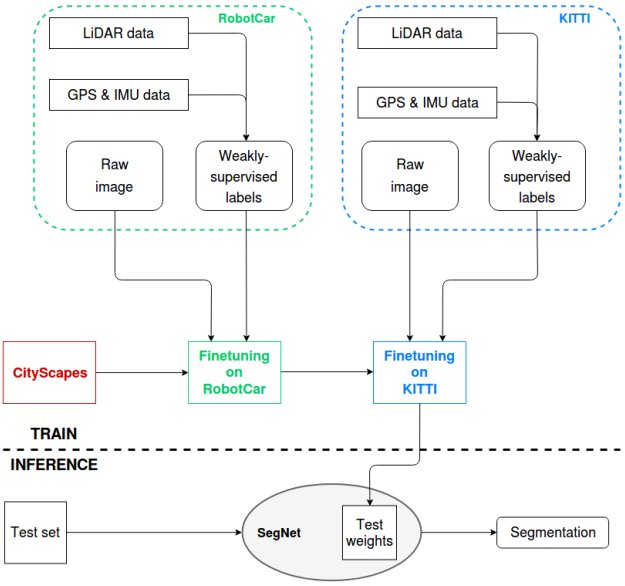


Figure 2: The main workflow of the applied approach.

labels in the source domain and no annotation in the target domain. The goal of domain adaptation is to be able to transfer the knowledge of the source domain to a different but related domain by handling the variation between the two data distributions. Initially domain adaptation has centred around image classification where the domain shift between stock photographs and real world cases of certain objects had to be overcome (Zhuo et al., 2017).

Some approaches for domain adaptation include the aim for maximal domain confusion (making domain distributions as similar as possible) (Tzeng et al., 2015) while others align the features in source and target domain by assuming that the source classifier and target classifier differ by a residual function (Long et al., 2016).

Domain adaptation for semantic segmentation is initiated by (Hoffman et al., 2016) who considered the learning scenario of strong supervision in the source domain while no supervision was available in the target domain with the goal of semantically segmenting images. (Chen et al., 2017) proposed an unsupervised learning approach for road scene segmentation in order to adapt to different environments of cities around the world.

Although research on different approaches to mitigate domain shift is known, only few resources target to explore the factors causing effects of domain shift on semantic segmentation. (Kalogeiton et al., 2016) analysed possible domain shift parameters for object detection by examining four factors. To the best of our knowledge, analysis of domain parameters for semantic segmentation is an untouched field of research. This work can be seen as a first step towards a deeper understanding of the influencing factors of the domain shift within the area of semantic segmentation.

## 3. METHODOLOGY

The main workflow of this project is presented in figure 2. As a first step, LiDAR, GPS and IMU data are used to create *groundtruth* (noted as weakly-supervised labels in figure 2) for the raw input images. This weakly-supervised labelling is done for three datasets, KITTI, RobotCar and WEpod (section 3.1). The groundtruth, together with the input imagery is used to fine-tune a base network. This base network is pretrained

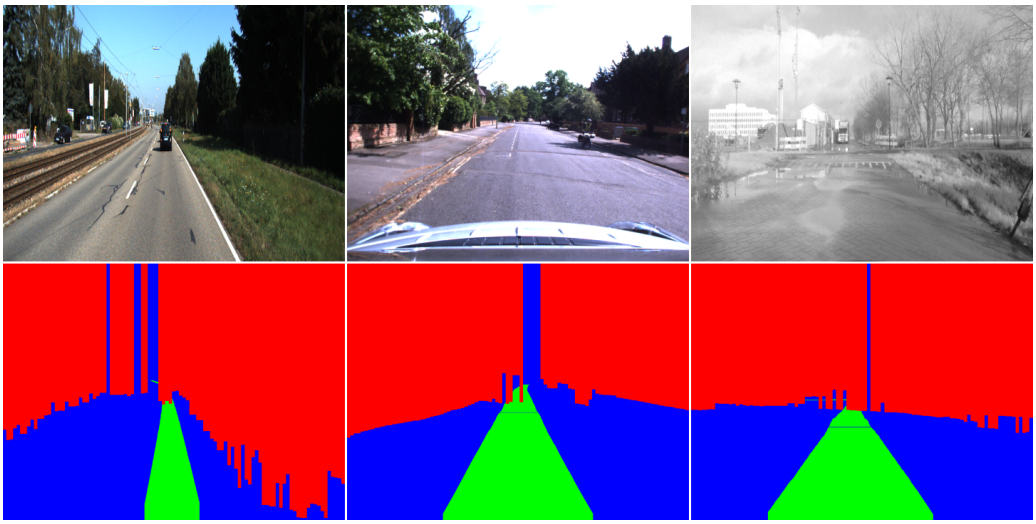

Figure 3: Example images and created labels of the KITTI raw dataset (left), Oxford RobotCar (middle) and the WEpod dataset (right). Red refers to occupancies, blue to unknown area and the drivable path is depicted as green.

on the CityScapes dataset and will be fine-tuned in two stages. The workflow shows that the first fine-tuning step occurs on the RobotCar dataset. The final weights, resulting from a second fine-tuning (on KITTI) will serve as test weights of the CNN for new, unseen imagery (KITTI, RobotCar and WEpod).

### 3.1 Datasets

Three datasets have been used in order to train a neural network based on weakly-supervised labels, to perform pixel-wise labelling. Weakly-supervised is referring to the approach of creating labels and thus creating training images without any manual labelling and is further explained in section 3.2. Our network is pretrained on the CityScapes dataset (Cordts et al., 2016) and fine-tuned on both the RobotCar dataset (Maddern et al., 2017) and the KITTI dataset (Geiger et al., 2013).

The platforms of RobotCar and KITTI are both equipped with a laser scanner, camera, IMU and a GPS navigation system. These sensors are vital for training the network. The laser scanner is used for obstacle sensing. In the case of KITTI, GPS is used to obtain the trajectory of the vehicle and obstacles are sensed by a Velodyne HDL-64E scanner. For RobotCar the trajectory is obtained by means of visual odometry and scanning is performed using a SICK LD-MRS 3D LiDAR.

**CityScapes**
The original CityScapes dataset consists of images with corresponding semantic labels which are subdivided into 5 000 fine annotated images and 20 000 coarsely annotated images. 19 semantic classes are present in the original dataset. This dataset is the base for our network because it is a high quality dataset for semantic segmentation. Additionally it has off-the-shelf weights available for SegNet (section 3.3) on the 11 class version of the CityScapes dataset[1].

**Oxford RobotCar**
(Maddern et al., 2017) have collected more than 1000 km of recorded driving over a period of a year. One route in central Oxford is covered twice a week. This driving scheme lead to large variations in scene appearance due to illumination, weather and seasonal changes and dynamic objects (Janai et al., 2017). Weakly-supervised segmentation is applied on the Oxford RobotCar dataset, labelling a total of 3033 images. These images are randomly subdivided into a training set of 2730 images and a validation set of 303 images.

**KITTI**
The KITTI vision benchmark suite is a well-known dataset related to different computer vision problems. In contrast to RobotCar, KITTI has large diversities in environmental changes but lacks this diversity in seasonal changes and weather conditions. The raw recordings were used for creating weakly-supervised labels. From the raw KITTI dataset (City), a total of 1174 images are labelled. From this total set, 1060 training images and 114 validation images are separated.

**WEpod**
KITTI and the largest part of CityScapes are recorded in Germany, while the United Kingdom is the setting for RobotCar. WEpod is recorded in the Netherlands in different settings. Data obtained from the WEpod is obtained in only one day and therefore does not have the diversity which CityScapes, KITTI (environment) and RobotCar (weather/season) do have. WEpod imagery is only available as greyscale images where all other datasets contain RGB images.

### 3.2 Weakly-supervised labels

Traditionally, a neural network needs a lot of training images in order to make good predictions. Transfer learning is one way to reduce the amount of data that is needed. However, even in the case of transfer learning, annotations are often still required. With an annotation time of 1.5 hour per image, this will lead to a significantly large annotation time for a complete training set. Weakly-supervised labels can be a solution to create a vast amount of groundtruth. These can be created by using sensor data of the recording platform.

In order to create labels in an automated fashion, the segmentation method from (Barnes et al., 2017) is adapted which consists of three parts. First, the drivable path is projected into the image assuming this is equivalent to the actual driven path in consecutive timestamps. This path refers to the outermost points of contact of the tires with the ground.

[1] Obtained from the SegNet Model Zoo:
https://github.com/alexgkendall/SegNet-Tutorial/blob/master/Example_Models/segnet_model_zoo.md

The 64-beam LiDAR used by KITTI also senses the road, these points need to be removed. All points that lie at least 10 cm above the ground plane in the LiDAR coordinate frame (with the origin at the laser scanner) are retained. To represent the slope of the road ahead, the height difference between the future poses is used. After filtering the ground scatter, the image is subdivided in 100 vertical bins. For each bin, all pixels above the lowest projected laser point are labelled as obstacle in that specific bin. When there is no laser point present in a bin, no pixels in that bin are labelled as obstacle. In case a pixel obtains labels both as drivable path and obstacle, the obstacle label is superimposed over the drivable path.

As third and last step, pixels without a label at this point will be labelled as unknown area. These areas often consist of sidewalks, road and depending on the LiDAR (4-beam; RobotCar and WEpod) sometimes will include small areas of other vehicles and buildings. This characterises that we are using weak labels. Figure 3 shows example images from KITTI, RobotCar and WEpod with the corresponding weak labels.

### 3.3 Architecture

The SegNet architecture (Badrinarayanan et al., 2017) is fine-tuned and deployed in this segmentation problem. SegNet has proven to obtain robust accuracies for semantic segmentation and provides real-time deployment, a critical requirement for autonomous vehicles. SegNet is an encoder-decoder network and has a symmetrical shape because encoder and decoder are similar.

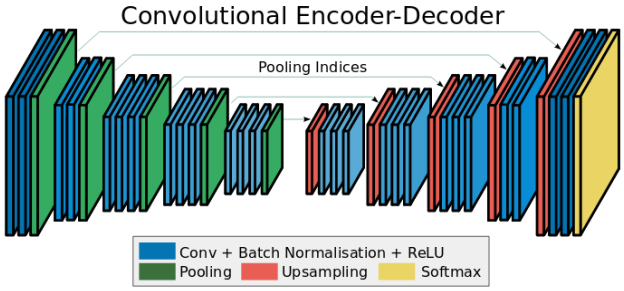


Figure 4: Encoder-decoder architecture of the implemented CNN. Image modified from (Badrinarayanan et al., 2017).

The encoder of SegNet is identical to the VGG16 network (Simonyan and Zisserman, 2014) except for the three fully connected layers of VGG16, these are not present in SegNet. This results in an encoder of 13 convolutional layers. Every convolutional layer (except the last layer in the decoder) is followed by a batch normalisation layer (Ioffe and Szegedy, 2015) and a Rectified Linear Unit (ReLU) activation function. The pipeline of these operations is presented as a blue layer in figure 4. The encoder is subdivided into five *blocks*. These blocks differ by the shape of the input (feature) map. This is realised by a max-pooling layer which appear after the second, fourth, seventh, tenth and thirteenth convolutional layer (the green layers in figure 4). During these downsampling steps, indices of the pooled (maximum) values are transferred to the corresponding upsampling layers (red layers in figure 4) in the decoder, aiming to keep part of the spatial content of the original input (feature) maps. The last layer of the decoder consists of a softmax layer (yellow layer in figure 4) which results in a pixel-wise segmented image.

Equation 1 represents the original cross-entropy loss of one observation where C represents the total number of classes. $y_c$ is a binary indicator. It is equal to 1 when the observation is classified correctly and it takes a value of 0 otherwise. $\hat{y}_c$ is the predicted probability that the observation is classified as class $c$.

To tackle the problem of class imbalance (i.e. more pixels are classified as obstacles and unknown area than drivable path), the original cross-entropy loss is weighted in the SegNet architecture. These weights are class dependent and different for each dataset. The imbalance weights are not trainable which means that they are constant throughout training. Computation of the weights is according to equation 2 which represents median frequency balancing (Eigen and Fergus, 2015).

In equation 2, *f(c)* stands for the frequency of $c$ (i.e. the number of pixels of class $c$ divided by the total amount of pixels in the image). *M(F)* is equal to the median of frequencies of all classes. This weighting procedure results in low weights for the larger classes while the smaller classes (drivable path in our case) has the highest value and hence, will cause the loss to increase. In order to determine these values, we based the weight for occupancy, drivable path and unknown area on a subset of the training data consisting of 256 images.

The objective or cost function of SegNet is a weighted cross-entropy loss, summed over all pixels of the mini-batch and is shown in equation 3 (note that the regularisation term in this loss function is ignored). P is the total amount of pixels in each mini-batch (containing four images).

$$\mathcal{L}(y, \hat{y}) = -\sum_{c=1}^{C} y_c \ln(\hat{y}_c) \tag{1}$$

$$W_c = \frac{M(F)}{f(c)} \tag{2}$$

$$\mathcal{L}_{SegNet}(y, \hat{y}) = -\sum_{p=1}^{P} W_c \cdot \sum_{c=1}^{C} y_c \ln(\hat{y}_c) \tag{3}$$

To make sure the applied network is suitable for the task, as assumed based on the literature (Barnes et al., 2017), some basic implementations are carried out. The baseline setup can be seen in figure 2. After the first fine-tuning step based on RobotCar, a second fine-tuning step is executed with the KITTI dataset. Furthermore, three datasets (KITTI, RobotCar and WEpod) are combined into one mixed dataset and the network is fine-tuned at once on this mixed dataset in order to set an upper bound for the problem.

## 4. EXPERIMENTS

Several experiments are carried out to explore the factors that influence the grade of success of a domain adaptation. All factors that are investigated are explained below.
The network is fine-tuned using the a modified version of the Caffe deep learning framework (Jia et al., 2014) and all experiments are carried out using a NVIDIA GeForce 1080 Ti GPU.

**Baseline and upper bound** As mentioned in section 3.3 a baseline is created through some simple setups. The results of the network on RobotCar are good when the network is fine-tuned on RobotCar while not producing useful results when segmenting KITTI test images. When we fine-tune the network a second time (this time on KITTI) the opposite is true. Tests on KITTI data have high evaluation values while, apparently, the network ”forgot” how to segment RobotCar images and segmentation results are bad.

Besides this baseline, an upper bound is created in order to show to what extent it is possible to achieve good results when all domains are included in the training phase in one fine-tuning step. Results are visible in table 1.

**Number of classes** The initial proposal included three classes as proposed in section 1. Because our target platform is equipped with a LiDAR, it would be possible to deploy it during test phase. Therefore, occupancies can be processed on the fly and only two classes need to be present in the segmentation: drivable path and *non-drivable path*. The idea behind the focus on path proposal estimation is that fusing two chaotic classes will increase the overall accuracy. Occupancies and unknown area are referred to as chaotic because they do not have clear distinguishable features. Reducing the number of classes for the groundtruth of all training images also affects the weight values as calculated in equation 2 since the ratio between the classes is shifted and imbalance weights will therefore be recalculated.

**Colour** Originally, SegNet is intended for RGB images. However the target imagery (WEpod) is only available in greyscale. Initially this was handled by copying the greyscale channel three times such that these images could be used as input for the SegNet trained on RGB images. However, this implies that features which are learned primarily by colour are not obtained when tested on target imagery. Therefore, another setup was made by fine-tuning the neural network on the same images but converted to greyscale. The comparison of these setups will indicate to what extend colour is important for creating features, which is automatically done by the neural network.

**Horizon** The height of the horizon line in the image for a certain camera is the result of the camera height, roll and pitch. KITTI and RobotCar data have a similar horizon height in the images (they only differ by a few pixels). However, WEpod images have a relatively low horizon line due to the significant lower placement of the camera ( 0.8m versus 1.65m for KITTI and 1.52m for RobotCar), resulting in a smaller part of the image containing road and potentially a crucial difference between the datasets. To examine this, the horizon height of the WEpod test images is changed such that it is similar to KITTI and RobotCar.

**Order of training** Another training setup is effecting the order in which the network will be fine-tuned. As a first approach, SegNet is fine-tuned on RobotCar first and later on it is fine-tuned on KITTI. To exclude the effect of tuning order, the setup is reversed; first train on KITTI and later on RobotCar. Exactly the same data is used to train with exactly the same settings. The only difference is the order of tuning and hence the workflow depicted in figure 2 does not match with this setting.

**Left-hand traffic** The RobotCar dataset is recorded in Oxford and consequently trajectories of the vehicle are located on the left side of the road. The opposite is true for KITTI and the target domain of WEpod, thus an experiment where the training and validation images are flipped is executed. An example is shown in figure 5.

**Data equalisation** The initial dataset combination of RobotCar and KITTI consisted of 2730 and 1060 training images respectively. In contrast to the imbalance in classes, the imbalance between these two domains is not resolved in the loss function. Therefore, equalising the number of training images from these datasets will potentially resolve a bias towards the larger dataset in the feature space.

**Cropping** The original resolution differs for all three datasets. Changing this is done by cropping an area equal to the size of the KITTI images before resizing these images such that they can serve as input image for SegNet. As a consequence of cropping, the bonnet and a large part of the sky are eliminated in the RobotCar imagery (this was also done for KITTI before publishing the raw dataset). After resizing, the image has changed as can be seen in figure 5.

Figure 5: Original image of the RobotCar training set (left), after the flipping procedure (middle) and after cropping (right).

### 4.1 Evaluation

Several classical metrics are available to evaluate the performance of semantic segmentation on pixel-level. Since there is no benchmark suite for the semantic classes that are applied, no comparison with respect to the current state-of-the-art concerning semantic segmentation can be made.

To evaluate the results both quantitatively as qualitatively, groundtruth is required for the test images. Groundtruth is created by labelling test images similar to the labelling technique used on the training images (explained in section 3.2).

The following metrics are taken into account for each class separately (Fritsch et al., 2013):

$$\text{Precision} = \frac{TP}{TP + FP} \tag{4}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{5}$$

$$\text{IoU} = \frac{TP}{TP + FP + FN} \tag{6}$$

where TP, FP, TN and FN denote True Positive, False Positive, True Negative and False Negative respectively. Accuracy is left out since it is misleading in most cases, being biased towards large classes. When qualitatively evaluating a drivable path it is important that occupancies are not segmented as drivable path. This situation is potentially more catastrophic than drivable path segmented as occupancies or unknown area as occupancy. Stated otherwise, it is important to have as few false positives as possible and thus, precision is a more informative metric than recall in the case of drivable path estimates.

For occupancies, the opposite is true. Qualitatively it is *better* to have classified too many pixels as occupancy than missing a lot of occupancies. Therefore, it is important to have as few false negatives as possible which means sensitivity is a more informative score than precision.
However, it has to be stressed that the performance cannot be summarized in one metric. It is a combination which will determine the performance. This combination is evaluated according to the Jaccard index, also known as the Intersection-over-Union metric (IoU) and is stated in equation 6.

## 5. RESULTS

A common problem in machine learning is overfitting. This happens when the model is overly complex for the data and usually is

the case when the number of parameters is larger than the number of constraints (images in this setup). Since overfitting is a potential explanation for insufficient adaptation, it is important to check for each fine-tuning step (training) that the model is not overfitting. Overfitting can be recognised by examining the learning curve. A continuously decreasing training loss but simultaneous decreasing validation accuracy is a sign of overfitting. However, when examining the learning curves of both training setups (figure 6), because validation loss is not increasing, it is clear that no overfitting has occurred.

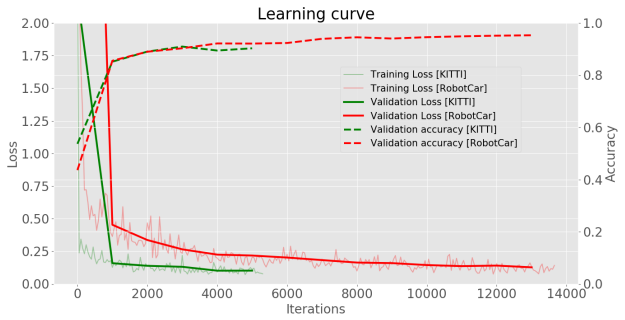


Figure 6: The learning curve for both fine-tuning steps in the baseline network.

The remaining results are divided into two parts. First, some generic outcomes are shown and compared to the baseline we have set. Some specific use cases that show noteworthy results or an exception on the rule are illustrated in subsection 5.2.

### 5.1 Generic results

**Baseline** The original setup is shown in figure 2 and was intended to act as a baseline for the experiments. The evaluation results of this setup are shown in the upper part of table 1. The table represents metric values for drivable path and occupancies tested on RobotCar, KITTI and WEpod data. Unknown area is left out because no conclusions can be made based on this class since this class is the *remainder* of the image and contains unknown space (e.g. road but also sidewalks and even parts of obstacles). Hence this class is not useful for autonomous vehicles. After analysing the results it is clear that tests on the KITTI dataset succeeded the best. This is likely to be caused by the order of training which is confirmed by the experiment where the order of training is reversed. Another confirmation of this reason is the comparison of results after the first and second fine-tuning (not shown in table 1). It is remarkable that differences between RobotCar and WEpod concerning drivable path estimation do not differ more since the network received training images from the RobotCar domain but not from the WEpod domain.
For both RobotCar and KITTI, there is a considerable difference between the metrics for occupancies and metrics for drivable path with higher values for performance on occupancies. However, the opposite is true for the WEpod data.

**Number of classes** A consequence of converting unknown area and occupancies to one mixed class (non-drivable path) for all training images is that the output of the network consists of two classes and hence only the drivable path can be evaluated. A strong signature throughout all datasets is the high values for recall while the IoU and precision values are low. This is caused by only few false negatives (instances where the network missed to classify pixels as drivable path) and a lot of false positives (the network claimed that pixels are drivable path, while they are not) compared to the number of pixels which are classified as drivable path. We do not have an explanation yet for this phenomenon.

**Colour** Adapting the training images from RGB to greyscale, has big influence on the quality of the segmentation of WEpod test images. A small increase in precision (3.6%) and huge increase of recall (46.7%) for occupancies is seen. Even higher increases are experienced for RobotCar and KITTI has a similar trend. However, these increases come at the high cost of dramatically decreasing precision, recall and IoU for the drivable path estimation. IoU values drop with 7.5%, 17.9% and 23.9% for KITTI, RobotCar and WEpod, respectively. In the case of WEpod, the resulting IoU value is 7.2% (while precision is 93.1%) which means that the number of pixels that are falsely negative are considerably larger than the portion of pixels that are correctly classified.

**Horizon** As changing the horizon line only consists of changes in the WEpod dataset, only segmentation on the WEpod is explored. Shifting the horizon line in the target domain results in a positive effect for the occupancies. A precision increase of 3.3% is seen while the recall metric has a significant increase of 20.7%, all relative to the baseline. This translates to less false positives and more importantly less false negatives or alternatively more true positives. Changes concerning drivable path are less sensational. Although, there is an increase of precision (4.4%) the shift of the horizon line also results in a drop of sensitivity of 3.3%.

**Order of training** This setup only requires changes in the training setup. Instead of tuning on RobotCar first, KITTI is now used as first fine-tuning step. During test phase, this reversed approach changed the metrics rigorously for the WEpod dataset. Although there is a slight decrease of recall (3.7%), the strong improvement of precision to 87.8% (increase of 33.7%) results in an increase of IoU of 4.7%. Occupancies even see a larger increase in all metrics. The resulting IoU for the occupancies of 73.5% is close to the upper bound (which is 77.9%) and is the result of a high recall value (97.2%) and high precision value (73.5%). This combination of metric augmentation is visualised by more unified drivable paths and better predictions for occupancies. To show these differences, figure 10 displays a test image of the WEpod dataset as output of the baseline network and as output when the order of training is reversed.

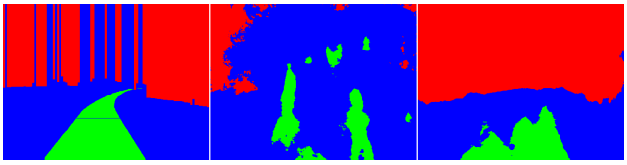

Figure 7: Groundtruth of a test image from the WEpod dataset(left). The same test image predicted through the baseline method (middle) and the output of our network when the order of training is changed in the setup (right).

The influence on RobotCar and KITTI is reversed. Because the last fine-tuning step is done on RobotCar, the evaluation metrics are similar to the metrics for KITTI in the baseline case and vice versa.

**Left-hand traffic** Flipping the training images of RobotCar, will lead to similar results as converting the training images from RGB to greyscale. In the case of WEpod, the resulting precision is very high (92.4%) with a very low IoU (15.3%) value. This is caused by a large number of false negatives compared to the true positives. Otherwise stated, drivable path is only segmented very sparsely. The same is observed on the RobotCar and KITTI test images however, less extreme. Throughout all datasets, the occupancies are better classified when training images of RobotCar are flipped.

| | Metric | RobotCar | | KITTI | | WEpod | |
|---|---|---|---|---|---|---|---|
| | | Drivable path | Occupancy | Drivable path | Occupancy | Drivable path | Occupancy |
| **Baseline** | Precision [%] | 66.2 | 94.9 | 71.0 | 96.7 | 54.1 | 66.2 |
| | Recall [%] | 36.6 | 77.9 | 73.6 | 84.3 | 42.2 | 25.8 |
| | IoU [%] | 31.3 | 74.4 | 58.6 | 82.0 | 31.1 | 23.6 |
| **Upper bound** | Precision [%] | 90.8 | 92.4 | 83.1 | 94.8 | 92.9 | 81.6 |
| | Recall [%] | 77.7 | 97.0 | 75.1 | 92.5 | 85.3 | 94.0 |
| | IoU [%] | 75.4 | 89.8 | 66.8 | 87.9 | 79.7 | 77.9 |

Table 1: Evaluation values for the baseline and upper bound of the experiments.

**Data equalisation** Reducing the number of training images for RobotCar to a total of 1060 and thus setting it equal to the KITTI dataset does not result in an increase of drivable path estimation for the WEpod dataset since it decreases all three metric values. KITTI shows similar behaviour. Although RobotCar only has half of its initial number of images during the training phase, it does see a increase in the performance on drivable path estimation.

**Cropping** The change of aspect ratio for RobotCar training images is evaluated slightly different from the other experiments. The network is first fine-tuned on the RobotCar dataset containing cropped images such that it resembles the KITTI dataset before resizing. After this first fine-tuning, the network is tested on the KITTI dataset and compared to evaluation after the first fine-tuning step in the original setup. Where in the original setup often no drivable path is recognised after the first tuning step, this number reduces by 50% when the training images are cropped to be similar. However, results on recall are still dramatically weak (10.0%) which means that there are only very few true positives compared to the amount of false negatives.

### 5.2 Case study

While results of RobotCar are generally unsatisfactory after the second round of fine-tuning, there is a remarkable case where the network does seem to have some decent output. This is shown in figure 8. This output also results in better metrics when comparing it to the baseline (which is averaged over all test images) of RobotCar. Concerning drivable path, increases of 32.9%, 21.8% and 26.8% are seen for precision, recall and IoU respectively. Although there is a small decrease for the precision of occupancies (5.4%) compared to the baseline, recall (13.2%) and IoU (7.9%) are significantly better. When doing a visual check on why this test image is different from the other test images of RobotCar, the difference in lighting conditions attracts the attention. This should be considered as potential cause of the domain shift.

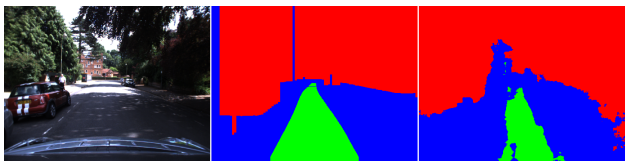

Figure 8: Test image from the RobotCar dataset (left). The same test image but labelled with the weakly-supervised method (middle) and the output of our network fine-tuned on the combined dataset (right).

A small part of the KITTI dataset has groundtruth available segmenting 94 test images into lane and non-lane labels. This is shown in the middle section of figure 9. Black represents non-lane while green equals the lane in the image. From this example, it is shown that even the upper bound network has trouble with predicting turns in an image. The logical cause of this problem is the fact that the vast majority of the training images is dealing with straight roads without any turns. Besides missing the turn, another less severe failure can be seen. The image shows a crossroad where the branch of the road has a steep upward slope. This branch is not recognised by the network but can be explained by the fact that groundtruth labels always show drivable path as the actual driven road and therefore, groundtruth labels do not include the branches themselves. It is suggested that a more extensive training set would resolve these issues because the number of crossroads in the current training set is minimal.

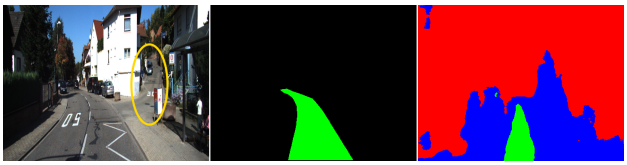

Figure 9: Test image from the KITTI dataset (left). Groundtruth label for the same test image (middle) and the output of our upper bound network (right).

Figure 10 illustrates two remarkable cases. The first is the absence of occupancy in the right part of the predicted segments. Furthermore, the estimated part of drivable path above the horizon stands out. Although, the rest of the drivable path is not perfectly segmented, the segment above the horizon is particularly strange because in none of the groundtruth labels drivable path will occur above the horizon. Thus, the network assumed this part of the image to be similar to the road. Despite this false segmentation, it does recognise the truck which is classified as occupancy. In contrast to the truck, the white building in the background is not recognised.

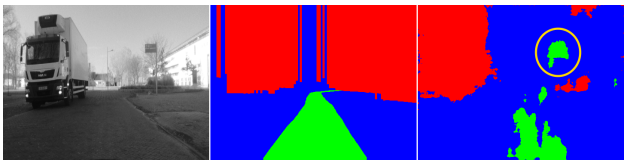

Figure 10: Test image from the WEpod dataset (left). Groundtruth label for the same test image (middle) and the output of our baseline network (right).

## 6. CONCLUSION

In this paper we investigated possible factors for a successful adaptation from the KITTI and RobotCar domain towards the WEpod domain. Modifications to the source datasets such as equalising both KITTI and RobotCar datasets, flipping RobotCar images and cropping RobotCar images did show small negative impact on the success of the domain adaptation. Changing the training images from RGB to greyscale resulted in a bigger decrease of domain adaptation success.

Horizon line change was an adjustment in target domain and showed only minor effect. Reducing the number of classes in the created groundtruth for the source domain has a negative effect on the target evaluation. Changing the setup by reversing the order of training does show an improvement on the target domain. This implies that the WEpod domain is more closely related to the RobotCar domain than to the KITTI domain. This is an indication where to further search the domain bias between the datasets.

Future work can consist of extra experiments such as changing the lighting conditions (gamma correction) of the image. The case study of the RobotCar showed potential in this area. Augmentation of the training data (rotate and scale) is not yet examined and has potential because it generates ”extra” training images without the need for additional unique images. Hence, data augmentation should be considered in future work.

## REFERENCES

Badrinarayanan, V., Kendall, A. and Cipolla, R., 2017. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE transactions on pattern analysis and machine intelligence 39(12), pp. 2481–2495.

Barnes, D., Maddern, W. and Posner, I., 2017. Find your own way: Weakly-supervised segmentation of path proposals for urban autonomy. In: Robotics and Automation (ICRA), 2017 IEEE Int. Conference on, IEEE, pp. 203–210.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K. and Yuille, A. L., 2018. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE transactions on pattern analysis and machine intelligence 40(4), pp. 834–848.

Chen, Y.-H., Chen, W.-Y., Chen, Y.-T., Tsai, B.-C., Wang, Y.-C. F. and Sun, M., 2017. No more discrimination: Cross city adaptation of road scene segmenters. In: 2017 IEEE Int. Conference on Computer Vision (ICCV), IEEE, pp. 2011–2020.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S. and Schiele, B., 2016. The cityscapes dataset for semantic urban scene understanding. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Eigen, D. and Fergus, R., 2015. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In: Proc. of the IEEE Int. Conference on Computer Vision, pp. 2650–2658.

Fritsch, J., Kuehnl, T. and Geiger, A., 2013. A new performance measure and evaluation benchmark for road detection algorithms. In: Int. Conference on Intelligent Transportation Systems (ITSC).

Geiger, A., Lenz, P., Stiller, C. and Urtasun, R., 2013. Vision meets robotics: The KITTI dataset. Int. Journal of Robotics Research (IJRR).

He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. In: Proc. of the IEEE conference on computer vision and pattern recognition, pp. 770–778.

Hoffman, J., Wang, D., Yu, F. and Darrell, T., 2016. Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. arXiv preprint arXiv:1612.02649.

Hong, S., Oh, J., Lee, H. and Han, B., 2016. Learning transferrable knowledge for semantic segmentation with deep convolutional neural network. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3204–3212.

Huang, G., Liu, Z., Weinberger, K. Q. and van der Maaten, L., 2017. Densely connected convolutional networks. In: Proc. of the IEEE conference on computer vision and pattern recognition, Vol. 1, p. 3.

Ioffe, S. and Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: Int. conference on machine learning, pp. 448–456.

Janai, J., Güney, F., Behl, A. and Geiger, A., 2017. Computer vision for autonomous vehicles: Problems, datasets and state-of-the-art. arXiv preprint arXiv:1704.05519.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S. and Darrell, T., 2014. Caffe: Convolutional architecture for fast feature embedding. In: Proc. of the 22nd ACM Int. conference on Multimedia, ACM, pp. 675–678.

Kalogeiton, V., Ferrari, V. and Schmid, C., 2016. Analysing domain shift factors between videos and images for object detection. IEEE transactions on pattern analysis and machine intelligence 38(11), pp. 2327–2334.

Long, J., Shelhamer, E. and Darrell, T., 2015. Fully convolutional networks for semantic segmentation. In: Proc. of the IEEE conference on computer vision and pattern recognition, pp. 3431–3440.

Long, M., Zhu, H., Wang, J. and Jordan, M. I., 2016. Unsupervised domain adaptation with residual transfer networks. In: Advances in Neural Information Processing Systems, pp. 136–144.

Lundquist, C., Schön, T. B. and Orguner, U., 2009. Estimation of the free space in front of a moving vehicle. Technical Report LiTH-ISY-R-2892, Department of Automatic Control, Linköping University.

Maddern, W., Pascoe, G., Linegar, C. and Newman, P., 2017. 1 year, 1000 km: The Oxford RobotCar dataset. The Int. Journal of Robotics Research 36(1), pp. 3–15.

Papandreou, G., Chen, L.-C., Murphy, K. P. and Yuille, A. L., 2015. Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation. In: Proc. of the IEEE Int. conference on computer vision, pp. 1742–1750.

Pathak, D., Shelhamer, E., Long, J. and Darrell, T., 2014. Fully convolutional multi-class multiple instance learning. arXiv preprint arXiv:1412.7144.

Simonyan, K. and Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

Tzeng, E., Hoffman, J., Darrell, T. and Saenko, K., 2015. Simultaneous deep transfer across domains and tasks. In: Computer Vision (ICCV), 2015 IEEE Int. Conference on, IEEE, pp. 4068–4076.

Yu, F. and Koltun, V., 2015. Multi-scale context aggregation by dilated convolutions. arXiv preprint arXiv:1511.07122.

Zhao, H., Shi, J., Qi, X., Wang, X. and Jia, J., 2017. Pyramid scene parsing network. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 2881–2890.

Zhuo, J., Wang, S., Zhang, W. and Huang, Q., 2017. Deep unsupervised convolutional domain adaptation. In: Proc. of the 2017 ACM on Multimedia Conference, ACM, pp. 261–269.