Coarse-to-Fine Semantic Road Segmentation Using Super-Pixel Data Model and Semi-Supervised Modified CycleGAN

Farnoush Zohourian and Josef Pauli
Department of Computer Science and Applied Cognitive Science, Chair of Intelligent Systems, Duisburg-Essen University, Duisburg, Germany
Email: farnoush.zohourian@stud.uni-due.de, josef.pauli@uni-due.de

***Abstract*—Real time road scene understanding is a crucial challenge for vision-based Advanced Driver Assistance Systems (ADAS). In our previous work, we proposed a method to utilize the advantages of enhancement-based segmentation method to improve the road segmentation result at reasonable computational effort. However, the performance is suffered from the poor efficiency and generalizability of Conditional Random Field (CRF) models. To overcome these drawbacks, we propose a novel semi-supervised refinement strategy based on a modified Cycle Generative Adversarial Network (Cycle-GAN). Our contributions are the following: first, our method learns a mapping between unpaired 4 channel images and a label domain. Second, a new pair-wise metric learning for a subset of images is added to improve the robustness of learning procedure. Third, we proposed a generative network with fewer parameters than the original Cycle-GAN. Forth, adversarial learning procedure is limited to the already predicted road boundary obtained from our recent work, that all together boost the segmentation performance. Experiments on KITTI benchmark show the effectiveness of the 4-7% of improvement compares to our previous work based on the super pixel and Convolutional Neural Network (CNN) and achieves comparable performance among the top-performing algorithms of recent un/semi-supervised semantic segmentation tasks.**

***Index Terms*—super-pixel, semantic segmentation, CNN, deep learning, conditional adversarial network, road segmentation, CycleGAN, un(semi)-supervised method**

I. Introduction

Automatic road detection, as one of the main features in urban image understanding, plays a critical role in various Advanced Driver Assistance Systems (ADAS). Several methods have been presented over the years, but the task is still far-off being completely solved.

The variations in illumination and appearance, occlusions, shadows, and unpaved areas are the main elements that cause the road segmentation challenging for autonomous vehicles. In recent years, Deep Convolutional Neural Networks (DCNNs) [1], [2] enabled great achievement towards better visual understanding on the tasks like image classification [3], [4], object detection [5] and image semantic segmentation [6]. Compared to the image classification, semantic segmentation, where each pixel is assigned to an object class in the image, is more challenging due to the combining of local information at the pixel-level with the general information obtained from multi-scale contextual reasoning [5]. Fast and accurate estimation of the pixel labels in a way compatible for embedding into real-time applications is not an easy and straightforward task. While most of the breakthrough deep Convolutional Neural Network (CNN) [7], models boost accuracy mainly by increasing the network size, in practice, they are fairly limited in computational power and memory, and they are infeasible or at least difficult to be used in embedded devices in self-driving cars.

In our previous work [8], we proposed a novel approach to utilize the advantages of CNNs for the task of road segmentation at the suitable computational effort. This method mainly differs from usual semantic segmentation methods in two aspects: first the input data model for the CNN network and second the simple CNN-network layering. The state-of-the-art convolutional neural networks for image segmentation are based on two different input data model: They are based on either patch-wise [9], [10] or pixel-wise dense classification [6]. The most recent improvements in CNN's are profited by using above input data models and increasing the network size, which together require powerful GPUs. Since deeper networks cause large computational costs, they are mostly not suitable for embedded devices in self-driving cars and ADAS. In our proposed work, the runtime benefits from designing a shorten CNN network and using the irregular super pixels [11] as basis for the CNN input rather than regular Patch or full image, which tremendously reduces the input size. This strategy disassembles the pixel grid into super pixels forming the basic units for the classification task by CNN. Reducing the input to the super-pixel domain allows the CNN's structure to stay small and efficient to compute, while keeping the advantage of convolutional layers. Although, this method achieved remarkable low computational time in both training and testing phases, there is still tradeoff between accuracy and computational efficiency. This happens due to two main reasons. First, the lower resolution of the

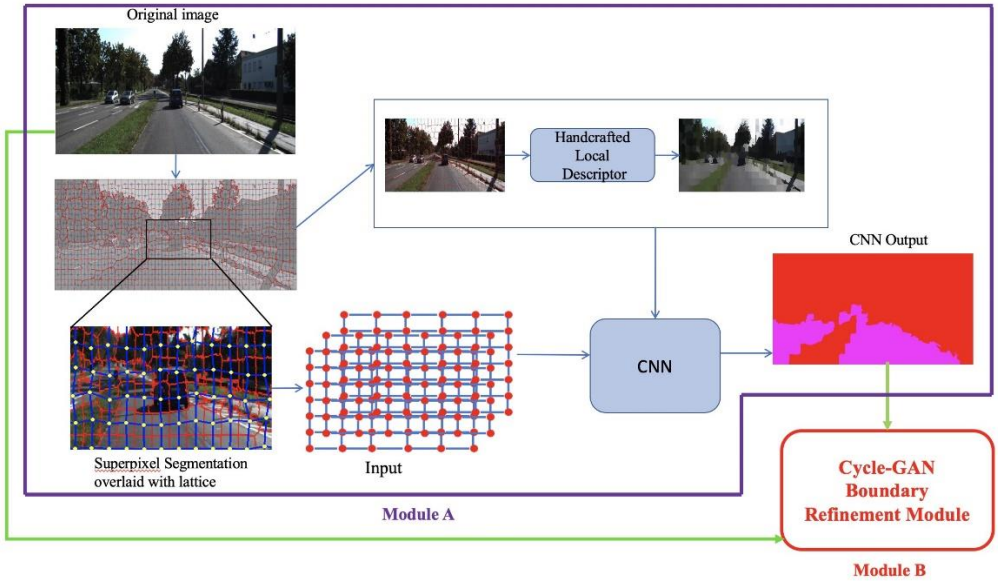
Manuscript received June 22, 2022; revised October 20, 2022.

doi: 10.18178/joig.10.4.132-144

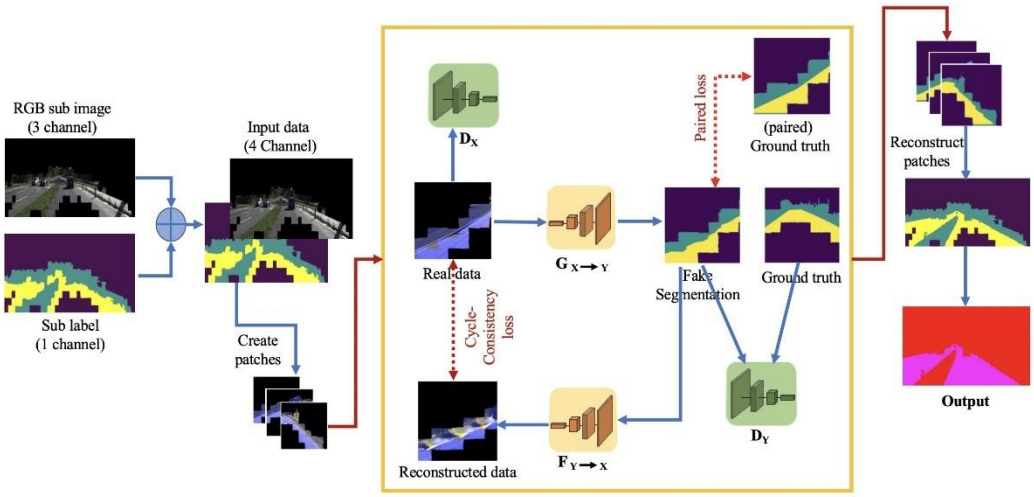
irregular super pixel domain yields naturally lower accuracy compared to high-cost pixel-wise based methods specially for super pixels that are not completely homogeneous which mostly happens in the road border. Second, CNNs have drawbacks to model the interactions and correlation between the output variables directly. All label variables are predicted independently from each other, which affects a good smooth segmentation. Since the independent prediction model does not capture global properties explicitly, various post-processing approaches have been explored to reinforce spatial contiguity in the output label maps. In our recent work [12], [13] we proposed a refinement segmentation method to model global properties like object connectivity, geometric properties, and spatial relationship between objects by using Conditional Random Fields (CRFs) [14]. The CNN-based approach mentioned above is used to define unary potentials and mean-field inference in fully connected CRF [15] is used as the pairwise potential. Such fully connected CRF has been found effective in practice to recover fine details in the output maps. To keep the computational time low, we limited the refinement scope to the super pixels bordered to street boundary, in which estimated in the first step. The refinement procedure applying CRF could effectively improve the performance, however the discrimination of the road pattern in challenging conditions, such as shadow on the road surface, illumination changes or similarity with neighboring patterns like sidewalks are still challenging areas. They suffer from either insufficient training data with high costs of manual annotation or the limitation of higher-order potentials in a pairwise model in CRFs. [16], [17]. To this end, we present a novel structure to enforce a higher-order consistency without being limited to a very specific class of pairwise potential. We explore a semi-supervised approach based on a modified cycle generative adversarial network (CycleGAN) [18], that the parameters of the higher-order potential can be learned, instead of directly being defined and integrated in the CRF model. The new proposed method enhances the predicted segmentation result by learning new label maps from 4-D channel images and ground-truth domain in an un(semi)supervised manner. It can enforce a model of higher-order consistency, that can be obtained neither by a per-pixel cross entropy loss in CNN method, nor pair-wise term in CRF model. The proposed method features the following contributions:

1) We introduce a modified cycle consistence generative adversarial network to enhance the road segmentations result obtained from our proposed super pixel-based CNN approach in semi-supervised learning.
2) The proposed adversarial method enforces cycle consistency to learn the mapping between unpaired 4-D channel images and a label domain. The 4-D channel is the original RGB images together with segmented road area obtained from our super pixel-based CNN road segmentation approach as fourth channel. They are passed through the modified CycleGAN, which extracts fine-grained road. The full architecture is shown in Fig. 1.
3) The computational cost is reduced in two ways; First by redesigning the residual blocks of the original CycleGAN into a shortened structure and reducing their number of parameters to keep the computational effort low. Second, the adversarial learning procedure is limited to the road boundary for boosting the segmentation performance.
4) Contrary to the original CycleGAN, we used a semi(un)paired datasets and we performed road segmentation enhancement by applying L1 loss between the output and target. Consequently, the enhancement quality improves better than the original CycleGAN and previous methods.

The remainder of this paper is organized as follows. Section II introduces previous road segmentation methods in deep learning domain. In Section III we describe our proposed method in detail. Implementation and training of the proposed adversarial model is explained in Section IV. We analyze the experimental results in both accuracy and time-efficiency in Section V. Finally, Section VI concludes our paper.



(a) Architecture of module A



(b) Architecture of module B

Figure 1. Total Architecture of the proposed Method. Each image in Module A is segmented into the inhomogeneous areas (super pixel), projected on a regular lattice structure. This lattice together with a higher dimensional feature descriptor extracted from each irregular super pixel are fed into a designed CNN network (as a pixel-wise two class-label classification task) to segment road regions. In Module B, the coarse road segmentation prediction obtained from previous step is smoothed by applying a semi-supervised modified CycleGAN, which maps a 4D-domain into unpaired label domain.

II. Related Work

A. Semantic Segmentation

Object detection, in particular, semantic segmentation, where each pixel in an image is assigned to a certain semantic class, are early steps in many autonomous driving systems based on computer vision. The inputs in the CNN-based semantic segmentation approaches are mainly in the two categories. They are either patched based [9], [10] where explicitly image patches are passed through the CNN network [7], or they differently use a Fully Convolutional Approach (FCN) [6], where the input images of any size could be accepted as the input and the prediction results equal to the input image-size are restored by using transposed convolution layers. The latter approach is more efficient, due to the prevention of the redundant computation of the low-level filters on each pixel in overlapping patches. The trade-off between input and output resolution of FCN architecture is typically solved by alternative ways such as a) utilizing skip connections from lower layers to the upper layers, which increase the resolution at the layers close to the output [6] or b) using dilated convolutions [19], which increase receptive field without losing resolution and increasing the computation time or c) applying the enhanced methods to integrate lower-level information [12], [13]. Most of the per pixel-labelling methods [20], [21] are too expensive for embedded applications and they require powerful GPUs to be fast enough for achieving the real-time performance. Reducing the computational burden of semantic segmentation is essential to make it feasible for embedded systems and autonomous driving. Image segmentation in real-time is a strong requirement in the self-driving applications to react to new events instantly and to guarantee the safety in execution speed. In [8] a 2D lattice of irregular super pixels is fed into the simple CNN network, which allows for easier extraction of the neighborhood information by the convolutional network in tremendously squeezed time. In U-net [22] like FCN [6], the down-sample part is done with convolutional layers and pooling layers, then the up-sample part is performed by deconvolutional layers to recover feature-map size increasingly. Contrary to the FCN, that combines fine layers and coarse layers at pixel-level to reuse the lower layer feature-maps, U-net combines local feature information with global feature information by concatenating lower layer feature-maps in down-sample phase with deeper layer feature-maps in up-sample phase. Drozdzal *et al.* [23] exchange the basic stacked convolutional blocks by the residual blocks [24] and introduce two types of skip connections to overcome the vanishing gradients problem. The short skip connections within the block alongside with the existing long skip connections between the corresponding feature maps of encoder and decoder modules lead to faster convergence during the training.

Many recent techniques perform CRF-based refinement approaches on the output produced by the convolutional neural network [25], [26]. They combine CNN unary label predictions with certain classes of pairwise potentials. In fully connected CRFs [15] a mean-field inference with millions of variables is computed using recent filter-based techniques. Applying another CNN or recurrent networks to learn pairwise information and drive long-range label interactions are proposed respectively in [26], [27].

B. Adversarial Learning

The presence of occlusions, illumination changes, large paved areas, shadows and overlap between objects are all factors that impact the lack of generalization ability of the segmentation model. Contrary to the great success of the networks mentioned above, based on convolutional neural networks and CRFs for enhanced semantic segmentation, they are highly dependent on the training and testing data, where they have equal underlying distribution. However, having diversity between the training and testing data is

common in the real world. In addition to that, the most state-of-the-art methods are fully supervised, requiring lots of labelled training data for better performance. Data augmentation is generally used as a technique for increasing the number of training data. Nevertheless, versatility in the distribution between the training and testing data sets leads to the unsatisfactory performance. To rectify this problem and adjust the methods for better generalization, unsupervised techniques are merged as a powerful technique to improve the generalizability of deep learning models to the new image domains without using any labelled data in the target [18], [28] or involving recurrent methods or any higher-order terms in the model itself like Conditional Random Fields (CRF). [15], [26]. Generative Adversarial Networks (GANs) [29] have two networks (named Generator G and Discriminator D), which are trained simultaneously in an adversarial way to create new data. The goal of the training is to implicitly find the underlying distribution of the training examples. The various types of GANs were proposed, such as cGAN, [30], DCGAN [31] and Pix2pix [28]. Deep Convolutional Generative Adversarial Networks (DCGANs) [31] use deep convolutional and convolutional-transpose layers in the discriminator and generator, respectively to learn representations from the un-labelled image data. Conditional Generative Adversarial Networks (cGANs) [30] employ image as conditional information to both generator and discriminator. Paired image-to-image translation (Pix2pix) [28] as an extension of the cGAN architecture uses a U-Net-based [22] network as G, and the PatchGAN [28] architecture as the discriminator network. Pix2pix evolved in recent years with the introductions of CycleGAN [18], where utilizes cycle consistency loss to translate from one domain to another, without requiring any paired data. Recently, different GANs are used for the semantic segmentation applications. Luc [32] proposed a convolutional semantic segmentation network along with an adversarial network to improve the segmentation performance compared with the traditional networks. In StreetGAN [33], the arbitrary sized of the road patches in aerial images are trained through GAN network to analyze and enhance the attributes in areas, in which the road extraction is difficult. A modified cycle generative adversarial network was proposed in [34] to improve the semantic segmentation performance for low light images. A dual-hop Generative Adversarial Network (DH-GAN) [35] is proposed to first segment the roads and intersections in aerial images and then a smoothing-based graph optimization procedure is applied to fit a best covering road graph.

III. Overview of Proposed Method

In Section III-B, we explain in detail our proposed semi-supervised CycleGAN is used for the enhancement of the road segmentation (Module B). This module refines the super pixel-based result to a pixel-wise one to increase the precision of the road segmentation. Like our recent proposed work based on the CRF model [12], [13], the refinement procedure is limited to the super-pixels touching the predicted road boundary. Working in this area helps to enhance the segmentation accuracy, while keeping the additional computational effort low. Our segmentation method follows several steps, which briefly are summarized in following steps; (a) segmenting the image into super-pixels, wherein the super-pixels are homogenous image regions comprising most pixels having similar image features. (b) defining an image descriptor for the irregular super-pixels, where each image descriptor comprises a plurality of image features. (c) The super pixels are assigned to corresponding positions of a regular grid structure extending across the image to create neighborhood relations for convolutional operation. (d) This lattice together with the image descriptors are fed into the designed shallow convolutional network based on the assignment to classify the super-pixels of the image according to semantic categories. (e) We create a 4-band imagery dataset by concatenating the sub area around the road boundary in the original images and their CNN label mask, by extracting those super-pixels touching the predicted road boundary. (f) To handle the tradeoff among the segmentation accuracy, memory resources and inference speed for large-scale image size, each 4D image is split automatically to a bunch of overlapping local patches according to the specific dataset, building a new augmented training data set from a single region. (g) Finally, a semi-supervised modified CycleGAN is proposed to enhance segmentation results along the road border by looking at the ground truth domain.

Steps (a) to (d) are covered in module A [8] and the rest are discussed in current work as module B. The proposed system obtained comparable performance among the top performing algorithms on the KITTI [36] road benchmark and its fast inference makes it particularly suitable for deployment in ADAS.

A. Super Pixel-Based Convolutional Neural Network

Super pixel segmentation is a technique, where an image is segmented into the regions with similar features like color, brightness, texture, etc. [37]. Super pixel units reduce the model complexity and computational cost by aggregating more compressed information than pixel units. Well-segmented super pixels preserve the object structures by correctly adjusting the segmented border to the object contour and consequently resulting in the accuracy improvement of the subsequent tasks like semantic segmentation.

For the convolutional operation in CNN network, we need a regular structure (grid format). The irregular super pixels with the different sizes or disordered shaped boundaries are not able to be directly convolved, due to the arbitrary neighborhood relations. In addition, imposing a specific topological structure to the super pixel segmentation mainly prevent the maximum pattern homogeneity inside of each super pixel.

In our previous work [8], we proposed a novel approach to obtain both highest homogeneity and convolutional ability. First, the image is segmented into coherent super pixels with maximum similarity among all pixels within the region. Then, an image descriptor is extracted from each irregular super pixel, which contains a plurality of image features. A super pixel lattice scheme for enabling

convolutional operation between the input data and kernels in CNN convolutional layers is proposed. Each super pixel is projected into a corresponding regular grid structure covering the whole image. This lattice is obtained in the early step of super pixel creation. Eventually, this lattice together with the image descriptors are fed to a shallow convolutional neural network for a pixel-wise classification purpose.

1) Input data model

A modified version of SLIC algorithm [11] is used for the super pixel segmentation. SLIC defines a 5-D space including 3-D spectral space and 2-D spatial space and uses a k-means clustering method for grouping of pixels in the 5-D space. The nearest neighborhood is selected based on the Euclidean distance between the center of sub-segment to the center of each adjacent segment. To avoid having isolated super pixels or disconnected regions, SLIC applies an" Enforce-Connectivity" procedure, which leads to an inconsistency in the total numbers of the super pixels created in each iteration of adopted K-mean clustering. It makes them unsuitable as a direct CNN input model. To address this issue, we do not remove any small region in the modified version. We keep the larger segment and merge the rest into the nearest super pixel. Then, we project the irregular super pixel segmentations from the final step to the lattice centered in the rectangular structure extracted from the first iteration of the modified SLIC method. For each super pixel a high dimensional feature descriptor is defined and comprises 69 image features including of 9 different color channels, 1 position and 59 Local Binary patterns (LBP) [38] to boost the accuracy and reliability. Finally, the provided input data model is fed to a small CNN with the super low computational cost presented in the following.

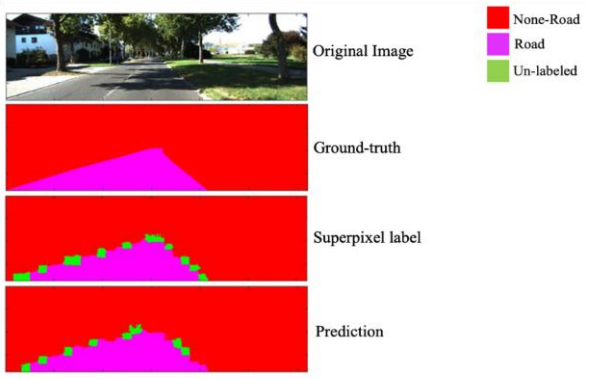


Figure 2. Output result based on super pixel-CNN method [8].

2) CNN network architecture

Contrary to the most of the semantic segmentation approaches, that need deep convolutional network layering to handle large image context, our method does not require a complex network architecture. Since our input data model benefits from larger informative units coming from the super pixels and their feature descriptor, rather than using pixel units which both together lead to a considerable reduction of the computational time. The network has two convolutional layers, two fully connected layers and one drop-out layer with non-linear activation function after each convolutional and fully connected layer. The input of our method is defined by the super pixel lattice on each image with size of *H/S* and *W/S*, where S is the initial super pixel size and *W*, *H* are image width and height respectively. The output is a binary classification with two classes of the *road* and *non-road* [8]. Fig. 2 shows the road segmentation prediction from one sample of KITTI urban scene images based on our proposed method.

B. Segmentation Refinement with Modified Cycle Consistent Adversarial Networks

Even though our super pixel-based convolutional network [8] tremendously reduced the computational effort with acceptable level of accuracy, however the trade-off between accuracy and time efficiency is still significant. The presence of occlusions and natural equivocation, shadows or large paved areas cause accurate road segmentation a difficult task. Unpaved roads mostly have poor conditions. The grass or sidewalk could be sometimes misclassified as the road boundary (See Fig. 3). In addition to that, all label variables in CNNs are predicted independently from each other. They are unable to capture the interactions and correlations between the output variables directly, which are important for a smooth semantic segmentation. Although, applying our refinement method based on CRF technique [12], [13] could mainly solve the above shortcomings, however, the differences in underlying distribution of training and testing data in the real world sometimes led to an unsatisfying performance. Having sufficient training data could solve this issue, but at the high cost of the manual annotation. Moreover, CRF based techniques mostly limited to the specific relational model in their higher-order potentials, which affects the generalization of these methods.

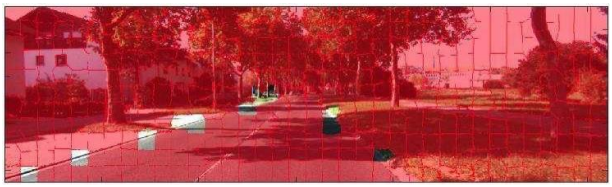


Figure 3. Most wrong prediction is appeared along the road boundary [8].

To this end, we propose in this paper a semi-supervised modified CycleGAN to improve the road segmentation performance. The original CycleGAN [18] uses unpaired data from two different domains to translate image-to-image using two forward and backward models. Instead of aligned image pairs, it learns a mapping $G_{X\rightarrow Y}$ from the data distribution of domain *X* to domain *Y*. However, in the absence of the paired data, optimizing the adversarial objective is difficult and the generated data could be vague. To address this issue, the data was translated back to the original *X* domain via the generator $F_{Y\rightarrow X}$. On the one hand, the generator learns higher-level appearance features, so that the distribution of generated images $G(X)$ is corresponding to the data distribution in the target domain *Y*. On the other hand, input domain *X* and the learned mappings from regenerated back $F(G(X))$ should not contradict from each other. Similarly, *Y* dataset performs the same process above.

In our modified CycleGAN, we aim to enhance the road border segmentation from our proposed super pixel-CNN based network [8]. We define our source domain a 4band imagery as the combination of original images and their CNN mask label along the road border, and unpaired ground truth database as the target domain. In addition, unlike the existing CycleGAN, in our semi-supervised approach we added a paired L1 loss from a subset of our input domain and their corresponding target to improve the enhancement quality of fine segmentation. To improve the efficiency of the proposed method, we modified the generator network of the original CycleGAN with a lower computational cost.

In the remainder of this section, first the discriminator and generator architecture of our modified CycleGAN are explained in detail in Sections III.B.1 and III.B.2, then in Section III.B.4 we discuss the total Adversarial loss function.

1) Discriminator network

Our modified CycleGAN architecture involves the simultaneous training of the two generators and two discriminator models. Discriminators are trained in pairs with the generators to discriminate between real and fake images. Here, we explain in detail the architecture of the two discriminators in our model (See Fig. 4). The input to the discriminator is a real or fake image generated by the generator and the discriminator is trained to classify each image as real or fake. In this paper, we used PatchGAN discriminator [28]. The key advantage of this discriminator is the significant improvement of the resolution and details of the output image. In PatchGAN, the input image, which is either real or fake is fed into the network and returns you back the output in the form of a matrix $N \times N$ image patches instead of a single value.

In our network the output layer of discriminator is a patchGAN of size $N = 32$, representing overlapping image patches of size 70×70. The discriminator is a fully convolutional network consisting of five convolutional layers. In the first three convolutional layers, stride value is set to 2 and the last two convolutional layers have stride equal to 1, which totally reduced the size of feature maps to 1⁄8 and the final output layer produces a one channel prediction map having a value between 0 and 1 in all pixel locations. The convolutional layers are followed by the Leaky ReLU (LReLU) as the activation function and then an instance normalization layer. We have two discriminators D_X and D_Y. D_Y optimizes G to generate data from domain X into outputs indistinguishable from domain Y, and vice versa for D_X and generator F. The input in D_Y is a real image from domain Y (ground truth) or fake image (generated ground truth) having a size of 256×256×1 regarding to our dataset. The final prediction map has the size of 32×32×1, where each single output value has a receptive field of 70×70. It means the 70×70 overlapping image patches, that can be classified as real or fake. For example, if the discriminator trained on real data, loss objective takes the discriminator output when a real image is fed into, and a matrix of ones. In the case of fake data, loss objective takes the output of discriminator on the generated image and a matrix of zeros. Likewise, D_X performs the same process, with the difference, that the discriminator gets the input size as 256×256×4 regarding our dataset and produces the output at the same size as 32×32×1.

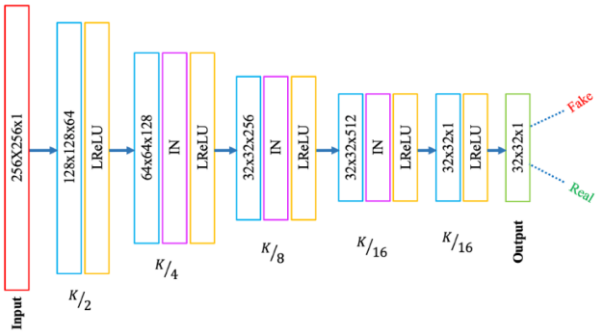


Figure 4. The architecture of our discriminator network. It has 5 convolutional layer of size 4×4. The first three have stride= (2,2) and the last two ones have stride= (1,1).

2) Generator network

The main differences of the architecture of our generator next to the original network in CycleGAN is shown in Fig. 5. The Generator network of original CycleGAN is a fully convolutional encoder-transfer-decoder network. The encoder consists of 3 down sampling blocks (first block is stride-1 7×7 convolution and the rest two blocks are stride-2 3×3 convolution, followed by the instance normalization and ReLU) for reducing the feature map size. The transfer part consists of nine residual blocks to learn residual features with reference to the layer inputs. The decoder mirrors the encoder and expands spatial dimensions of the feature-maps uses three up-sampling block until prediction result has the same scale to the input image.

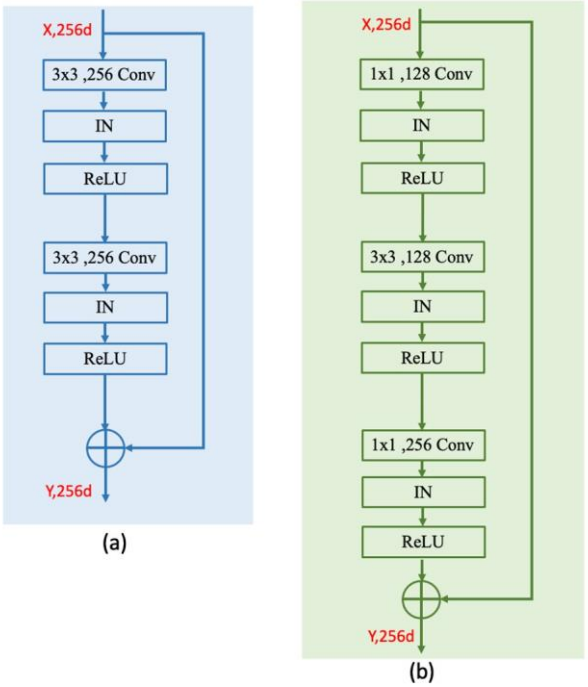


Figure 5. The architecture of the residual blocks in original CycleGAN (a) and our modified residual blocks (b).

To achieve our main goal in this study, to enhance the road segmentation, we modified the residual block of original CycleGAN as shown in Fig. 5. Contrary to the original network, our modified residual block has three convolutional layers in size of 1×1×128, 3×3×128 and 1×1×256 respectively, where each of them followed by instance normalization and Rectified Linear Units (ReLU) after first two layers. The number of filters in the first two convolutional layer is 128, which is reduced to half according to the original architecture and the last layer has 256 dimensions. In addition to that, we used only 6 instead of 9 residual blocks, so that altogether cause having 2,038,337 number of parameters as opposed to the 11,380,289 number of parameters in the generator of original network, which immensely reduces the computational cost by 5X and improves the performance. Regarding to the activation function in the residual blocks, we experimented different types of activation functions. It turns out that other types of activation functions, such as LReLU, are learning some artefacts such as super pixel boundary instead of road boundary.

3) Adversarial training

The power of CycleGANs is how they define the loss function and apply the full cycle loss as a supplementary optimization target. As we explained above, we're dealing with 2 generators and 2 discriminators. The generator is successful, if fake (generated) images are so good that discriminator cannot distinguish those from real images. In other words, the discriminator's output for fake images should be as close to 1 as possible. The total objective function of our generator network is given by equation (1), which is composed of the adversarial loss function of GAN (L_{adv}), cycle consistency loss (L_{Cyl}) and our new added paired loss (L_p) as the L1 distance between the output and the target in a small subsidiary of paired data to enhance the performance of the road segmentation results. Here, X represents our first 4-D domain (RGB information combined with coarse CNN segmentation result) and Y represents target domain (unpaired ground truth). In the original CycleGAN [18], log function-based adversarial loss was used, whereas in our modified version we used Mean-least-square in L_{adv} loss to increase the learning steadiness and convergence speed of our network.

$$\begin{aligned} L_{total}(D,G,F,X,Y) = {} & L_{adv_{(G)}}(G,D_Y,X) + \\ & L_{adv_{(F)}}(F,D_X,Y) + \\ & \lambda L_{cyl}(G,F,X,Y) + \\ & \eta L_P(G,F) \end{aligned} \tag{1}$$

here, G represents the generator and D represents the discriminator. Hyper-parameters λ and η are sequentially weighting scores for the Cycle-consistency and paired losses. Equation (2) calculates the adversarial loss function (L_{adv}). The adversarial loss function attempts to map the distribution generated from GAN to the actual distribution in target domain by playing a game that one player adjusts G to minimize the adversarial learning value, and another player adjusts D to maximize it. The $D_Y(G(X))$ makes the generative ability better in a way that the discriminator D_Y *is* unable to distinguish, whether the sample is coming from real or generated domain by maximizing the probabilities of D_Y (x) = 1 and D_Y (G(X)) = 0. Similar adversarial loss for the mapping function $F:Y \rightarrow X$ and its discriminator D_X i.e., $L_{adv(F)}$ (F, DX, Y) is defined.

$$\begin{aligned} L_{adv_{(G)}}(G,D_Y,X) &= \frac{1}{m}\sum_{i=1}^{m}\left(1 - D_Y\big(G(x_i)\big)\right)^2 \\ L_{adv_{(F)}}(F,D_X,Y) &= \frac{1}{m}\sum_{i=1}^{m}\left(1 - D_X\big(F(y_i)\big)\right)^2 \end{aligned} \tag{2}$$

where m is the number of sub-samples in each domain X and Y and x_i or y_i is one sample of $\{x_1, ..., x_m\}$ from domain X or one sample of $\{y_1, ..., y_m\}$ from domain Y. The adversarial loss alone enforces the generated output to be part of the proper domain but does not impose that the input and output are identified the same. To address this issue, cycle consistency loss is defined. It relies on the expectation that translating an image to the other domain and translating back again, should get something like what you put in. It enforces that $F(G(X)) \approx X$ and $G(F(Y)) \approx Y$. Equation (3) shows the cycle consistency loss by calculating the Mean Absolute Error (MAE) between reconstructed and original input data.

$$\begin{aligned} L_{cyl}(G,F,X,Y) = \frac{1}{m}\sum_{i=1}^{m} & [F(G(x_i) - x_i)] \\ & + [G(F(y_i) - y_i)] \end{aligned} \tag{3}$$

For winning this game by the Discriminators, they should correctly distinguish real and fake images, meaning the real images should be marked as close as to 1 and generated images thus predict 0, (See Equations (4)).

$$\begin{aligned} L_{real}^{(D)} &= \frac{1}{m}\sum_{i=1}^{m}[1 - D_X\big((x_i)\big]^2 + \frac{1}{m}\sum_{i=1}^{m}[1 - D_Y\big((y_i)\big]^2 \\ L_{fake}^{(D)} &= \frac{1}{m}\sum_{i=1}^{m}[D_Y(G(x_i)]^2 + \frac{1}{m}\sum_{i=1}^{m}\big[D_X\big(F(y_i)\big)\big]^2 \end{aligned} \tag{4}$$

The paired L1 loss (L_p) as shown in Equation (5) is the L1 distance between the output and target domain for a subset of paired data. By adding the paired L1 loss, we could train our network to improve the road segmentation performance, which is more like the target image rather than using the original CycleGAN. More details will be discussed in Section V.

$$\begin{aligned} L_P(G,F,X,Y) = \frac{1}{m}\sum_{i=1}^{m} & [\| G(x_i) - y_i \|_1] \\ & + [\| F(y_i) - x_i \|_1] \end{aligned} \tag{5}$$

where X is the 4-D channel input domain, and target Y is the paired ground truth image in a subset of our training domain. $L_{adv(G)}$ (G, D_Y, X) or $L_{adv(F)}$ (F, D_X, Y) losses are calculated based on the outputs of both the generator and discriminator, while cycle consistency loss and paired loss are based only on the outputs of the generator. λ and η are the penalized factors for cycle consistency loss and paired loss respectively. The optimal values were experimentally determined, and they will be discussed in Section V.

IV. Implementation of the Proposed Adversarial Model

A. Data Preparation

To evaluate the reliability of the proposed method using semi-supervised modified CycleGAN, experiments are conducted using open-source KITTI [36] data set. KITTI consists of 502 8-bits RGB images splits in two categories: train and test sets with ground truth label provided only for training set consist of two semantic classes named road and un-road. The segmentation label of the test set is not revealed publicly. The training set has 289 images (95 images with Urban Markings (UM), 96 images with multiple Urban Markings (UMM) and 98 images where the street has no urban markings (UU). The test set has 290 images including (96 UM, 94 UMM and 100 UU) images. The image dimensions are varied in the width size among in {1226, 1238, 1241, 1242} and the height in {370, 374, 375, 376}. We also divided the training set into two subsets named; train and val which is used as the two-fold cross validation to evaluate the learning approach while training. The validation set is 20% of training set consisting of images coming from all three different categories UM, UMM, UU. These images are selected from completely different video sequences, which are not part of the training set.

To keep the fairness of the measurement between the current work and our already proposed methods [8], which resulted the Super pixel-CNN based coarse segmentation, and our recent work [12], [13] for fine-grained road segmentation based on CRF technique, the same conditions and parameter values for data set preparation are used. In [8], SLIC parameters K = 400, m = 35 were used for creating super pixels, resulting in 396 super pixels in each image. Super pixel based segmented image was projected to a 11×36 lattice as CNN input model for the road segmentation. To evaluate the method, the accuracy was calculated on both super pixel and pixel level separately. The corresponding ground truth label in super pixel level was defined based on the majority pixel-labels inside the super pixel.

In current approach, the smoothness procedure is limited to the area around the road boundary. Considering the size of the KITTI images, we automatically extract sub-images, which are included only in the area around the road by selecting those super pixels, which are touching the road border. From each sub-image, we create a maximum of 6 overlapping image patches in size of 256×256 with the stride S = 196. Each RGB sub-image patch is combined with the coarse CNN segmentation as an extra band, that altogether are made our source domain. Same technique is used to create unpaired patches of label considered as the target domain. We discarded those patches, which have less than 2% of boundary pixels. Totally, we generate 1200 samples from the original KITTI training set, where 1020 samples are used for training and 180 samples are separately used as our validation set. In addition to that, we created 1422 samples from the original test set, that in combination with our training set used as the source domain. The sizes of the input images and the ground truth images are set to 256×256. The performance of our approach with respect to the state of art methods and our previous works, is investigated on KITTI dataset on both image perspective and a bird's eye projection.

B. Training Details

For training, we used two subsets of patches. One is the KITTI training set [36] (excluded the validation set), and another is the KITTI test set. We apply unpaired mapping between two domains, however, to improve the learning process we applied the L1 paired loss for a subset of our data. Mini-batch SGD and the adaptive moment estimation (Adam) optimizer [39] are adopted with momentum parameters $\beta_1 = 0.5$, $\beta_2 = 0.999$. Epsilon was set to $1E{-}08$ and the batch size is set to 1. The initial learning rate was set to 0.0002. Furthermore, the balancing parameters λ and η in equation (5) are analytically set to 10 and 1 respectively (for more detail see Sec. V-A). The weights of all filters are initialized from a Gaussian distribution with a mean of 0.0 and a standard deviation of 0.02. The data are normalized in the range [−1, 1]. All experiments are implemented in python, performed on google Colab using the TensorFlow framework (version 2.0).

V. Results and Discussion

In this section, first we analyze our model against some variants and demonstrate their empirical results to clarify, why we decided on our final model. These variates include a) applying different network layering respect to their processing time and accuracies, b) the importance of utilizing the paired loss to calculate the total objective function, c) the number of residual blocks in the Generator networks and lastly d) fine tuning the penalize values of the paired and cycle consistency losses. Next, we evaluate the performance of the current approach based on semi-supervised modified CycleGAN (final model) with the accuracy of the pixel grid obtained from the super pixel-based convolution network [8], our recent work for the enhancement of the road segmentation based on CRF technique [12], [13] and finally to the Original CycleGAN [18]. Regarding to the KITTI evaluation scheme [36], the measurements were done on both the image perspective and a bird's eye projection provided by the KITTI data set.

A. Model Analysis

We perform model analysis on our KITTI validation set (See Sec. IV-A) to evaluate our approach on the image perspective in different scenarios and to find the best model with the highest performance. Table I compare our six dominant storylines named V1 to V6. V1 scenario is the original CycleGAN [18] and V6 is our final best model (SP_CNN_Modi_CycleGAN) proposed in the current paper.

These methods are different in the following aspects: first, their designed generator network layers and different activation functions second, their total objective function third, the number of residual blocks and lastly, the best value for the weighted parameters to strengthen the regularization effect. The first column of the Table I refer to the varied generator architectures, which are shown in

Fig. 6. We designated two main different Generator architectures rather than the generator network in the original CycleGAN [18], to improve the road segmentation accuracy, while boosting the time-efficiency. We also analyzed different activation function to help the network to learn the complex patterns in the data and yield better results.

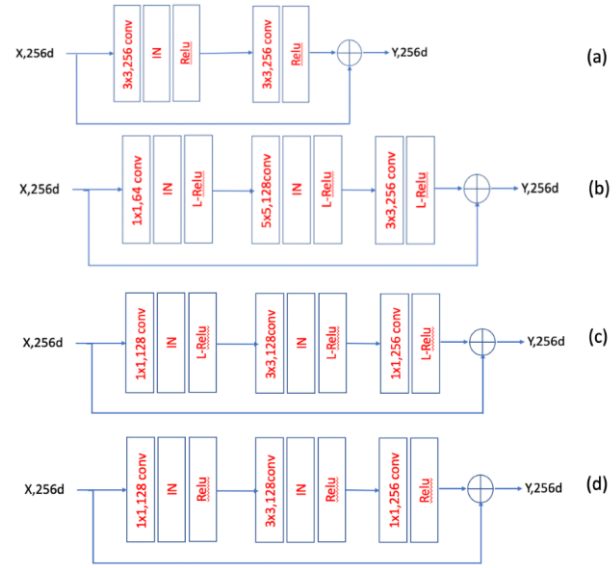


Figure 6. The architecture of the residual Blocks of the generator network in (a) Model V1 (Original CycleGAN), (b) Model V2, (c) Model V3, (d) Model V4 to V6 (SP_CNN_Modi_CycleGAN).

The second column shows the formula of the total loss function for each method. In V1 scenario, which is the original CycleGAN [18], the log function-based adversarial loss was used, whereas in the rest methods (V2 to V6), we used Mean-least-square in L_{adv} loss to increase the learning steadiness and convergence speed of our network. We also combine our proposed L1 paired loss to the total objective function in methods V5 and V6. Third column of the Table I indicate the number of the residual blocks used in the generator networks of each version. Despite of the original CycleGAN [18], we used 6 instead of 9 residual blocks in all scenarios to reduce the computational time.

The different weighted values for the regularization parameters λ and η in the total loss function are shown in the last column. We investigated several values for λ, however as in the CycleGAN paper [18] proposed $\lambda = 10$ was the best. We also tuned the L1 paired loss with various values, which the best two ones are collated in version V5 and V6.

We ran the experiments on the validation set in the image perspective for each scenario to choose the best model with the rational tradeoff between the accuracy and time efficiency. Table II summarizes the average evaluation results for each model and their required computational cost, which is acquired by the calculating the number of the parameters(*#Params*) that the generator network should learn. Fig. 7 presents the road segmentation results of an image from KITTI validation set, obtained from our six different storylines.

TABLE I. THE COMPARISON OF THE DIFFERENT SCENARIOS WHICH LEADS TO PROPOSE OUR FINAL SEMI-SUPERVISED MODIFIED CYCLEGAN. METHOD V1 IS THE ORIGINAL CYCLEGAN AND THE METHOD V6 IS OUR PROPOSED SP_CNN_MODI_CYCLEGAN

Method	Generator network	Total Objective Function	num residual blocks	Regularization Parameters
V1	Fig. 6(a)	$L_{total}(D, G, X, Y) = L_{adv}(G)(G, D_Y, X) + L_{adv(F)}(F, D_X, Y) + \lambda L_{cyl}(G, F, X, Y)$	9	$\lambda = 10$
V2	Fig. 6(b)	$L_{total}(D, G, X, Y) = L_{adv}(G)(G, D_Y, X) + L_{adv(F)}(F, D_X, Y) + \lambda L_{cyl}(G, F, X, Y)$	6	$\lambda = 10$
V3	Fig. 6(c)	$L_{total}(D, G, X, Y) = L_{adv}(G)(G, D_Y, X) + L_{adv(F)}(F, D_X, Y) + \lambda L_{cyl}(G, F, X, Y)$	6	$\lambda = 10$
V4	Fig. 6(d)	$L_{total}(D, G, X, Y) = L_{adv}(G)(G, D_Y, X) + L_{adv(F)}(F, D_X, Y) + \lambda L_{cyl}(G, F, X, Y)$	6	$\lambda = 10$
V5	Fig. 6(d)	$L_{total}(D, G, X, Y) = L_{adv}(G)(G, D_Y, X) + L_{adv(F)}(F, D_X, Y) + \lambda L_{cyl}(G, F, X, Y) + \eta L_P(G, F)$	6	$\lambda = 10$, $\eta = 5$
V6	**Fig. 6(d)**	$\boldsymbol{L_{total}(D, G, X, Y) = L_{adv}(G)(G, D_Y, X) + L_{adv(F)}(F, D_X, Y) + \lambda L_{cyl}(G, F, X, Y) + \eta L_P(G, F)}$	**6**	$\boldsymbol{\lambda = 10, \eta = 1}$

TABLE II. THE COMPARISON OF THE EVALUATION RESULTS ON THE KITTI VALIDATION SET BY APPLYING OUR DIFFERENT PROPOSED STORYLINES: METHOD V1 IS THE ORIGINAL CYCLEGAN AND THE METHOD V6 IS OUR FINAL PROPOSED SP CNN MODI CYCLEGAN METHOD

Method	ACC	F1	PRE	REC	#Params
V1	95.95%	92.74%	93.41%	92.10%	11.38×10^6
V2	96.02%	92.60%	95.15%	90.47%	3.85×10^6
V3	97.24%	95.15%	94.90%	95.41%	2.04×10^6
V4	97.29%	95.21%	95.20%	95.23%	2.04×10^6
V5	97.31%	95.24%	95.28%	95.20%	2.04×10^6
V6	**97.33%**	**95.26%**	95.58%	94.96%	2.04×10^6

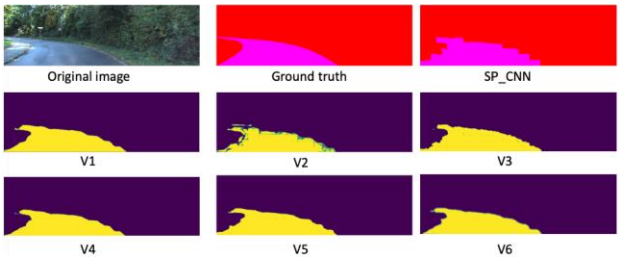


Figure 7. Road segmentation results obtained from models: V1 to V6.

A comparison of the computational costs of all methods shows that the generator architecture designed for V3 to V6 models is more time efficient than V1 (Original CyclGAN) or V2, while the accuracy is increased. Employing the LReLU instead of ReLU as the activation function in the generator network, provides a small negative gradient for the negative inputs. However, it causes that models V2 and V3 become more sensitive to some artifacts and their networks learn unnecessary information. As you can see in Fig. 7 the networks learn the gradient changes along our virtual super pixel boundary, which is created due to the selection of those super pixels touching the road boundary. Moreover, Leaky ReLU is computationally costlier and spends more time to converge to a global optimum. Analyzing the evaluation results obtained from V1 to V4 reveals that combing the L1 paired loss to the total loss function in V5 and V6 leads to the small improvements in the road segmentation performance and robustness of the varying degrees of complexity in the underlying distributions in different images. From both Table II and Fig. 7, the V6 model has the highest performance. Therefore, it has been selected as our final proposed method (SP_CNN_Modi_CycleGAN) in the current study.

B. Evaluation on Birds Eye Perspective

For evaluation in the birds-eye perspective, the images are projected on the ground plane in the KITTI benchmark via the known camera geometry. Table III shows the results on the test set based on our three proposed methods divided into the different road types (UM, UMM, UU, URBAN). Compared to our CNN approach the maximum F-score on all four urban categories improved approximately 4% on the official KITTI test set and around 1% in comparison to our recent approach based on CRF. In one urban category (UMM_ROAD), we had almost

7% improvement relative to the same category in our CNN technique, implying that the weaker accuracy in the SP_ CNN approach induced by inaccurate super pixels on the road border could relatively be fixed. Fig. 8 compares two samples in BEV on the KITTI test set in our SP_CNN approach and modified CycleGAN. Whilst the street is nicely segmented, there are a few false detections that mostly happened, when the segmented area was fooled by a shadow covering the street.

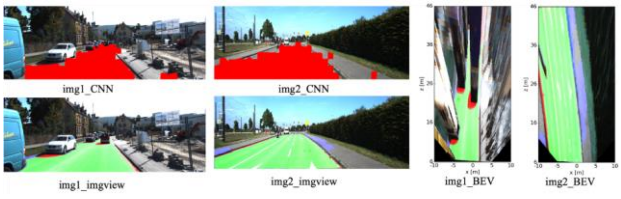


Figure 8. The road segmentation result from the KITTI test set on BEV, obtained from both SP_CNN and our modified CycleGAN approach. Here, blue is false positives, red denotes false negatives and green represents true positives.

TABLE III. EVALUATION RESULTS ON KITTI TEST SET. MAXIMUM F-MEASURE (MAXF), AVERAGE PRECISION (AP), PRECISION (PRE), RECALL (REC), FALSE POSITIVE RATE (FPR), FALSE NEGATIVE RATE (FNR)

Method	Benchmark	MaxF	AP	PRE	REC	FPR	FNR
SP_CNN	UM_ROAD	81.60 %	69.62 %	78.13 %	85.40 %	10.89 %	14.60 %
	UMM_ROAD	85.07 %	79.86 %	85.97 %	84.20 %	15.11 %	15.80 %
	UU_ROAD	78.47 %	65.18 %	74.20 %	83.25 %	9.43 %	16.75%
	URBAN_ROAD	82.36 %	72.31 %	80.48 %	84.33 %	11.27 %	15.67 %
SP_CNN_CRF	UM_ROAD	83.22 %	72.94 %	77.11 %	90.39 %	12.23 %	9.61 %
	UMM_ROAD	90.96 %	84.63 %	87.86 %	94.29 %	14.32 %	5.71 %
	UU_ROAD	80.02 %	67.93 %	77.56 %	82.64 %	7.79 %	17.36 %
	URBAN_ROAD	85.97 %	77.81 %	82.04 %	90.31 %	10.89 %	9.69 %
SP_CNN_Modi_CycleGAN	UM_ROAD	**85.01 %**	76.86 %	86.98 %	83.13 %	5.67 %	16.87 %
	UMM_ROAD	**91.80 %**	89.25 %	92.94 %	90.70 %	7.58 %	9.30 %
	UU_ROAD	**79.49 %**	68.66 %	85.19 %	74.51 %	4.22 %	25.49 %
	URBAN_ROAD	**86.90 %**	79.61 %	89.41 %	84.52 %	5.52 %	15.48 %

C. Evaluation on Image Perspective

Since the ground truth of the KITTI test set is not publicly available, we used the validation set (See Sec. IV-A) to evaluate our approach on the image perspective. Table IV summarizes the average evaluation results of all four urban categories from the validation set, which are obtained from four different methods. The accuracy obtained from the CNN part was 94.41%. In the current approach by applying modified CycleGAN technique, we had around 3% improvement in the accuracy, and we could reach about 97.33%. In comparison to our recently proposed method based on CRF and original CycleGAN, the road segmentation accuracy enhances approximately to 2%. Fig. 9 shows the output of our modified CycleGAN for several patches after the completion of the training process and Fig. 10 depicts one representative result based on investigated different methods.

TABLE IV. THE COMPARISON OF THE EVALUATION RESULTS ON THE KITTI VALIDATION SET BY APPLYING OUR DIFFERENT PROPOSED METHODS: OUR SP CNN CLASSIFIER, OUR ENHANCEMENT CRF METHOD, ORIGINAL CYCLEGAN AND OUR PROPOSED METHOD BASED ON SEMI-SUPERVISED MODIFIED CYCLEGAN

Method	ACC	F1	PRE	REC
SP_CNN	94.41%	95.18%	91.41%	97.34%
SP_CNN_CRF	96.85%	90.94%	92.30%	90.65%
SP_CNN_Orig_CycleGAN	95.95%	92.74%	93.41%	92.10%
SP_CNN_Modi_CycleGAN	**97.33%**	**95.26%**	95.58%	94.96%

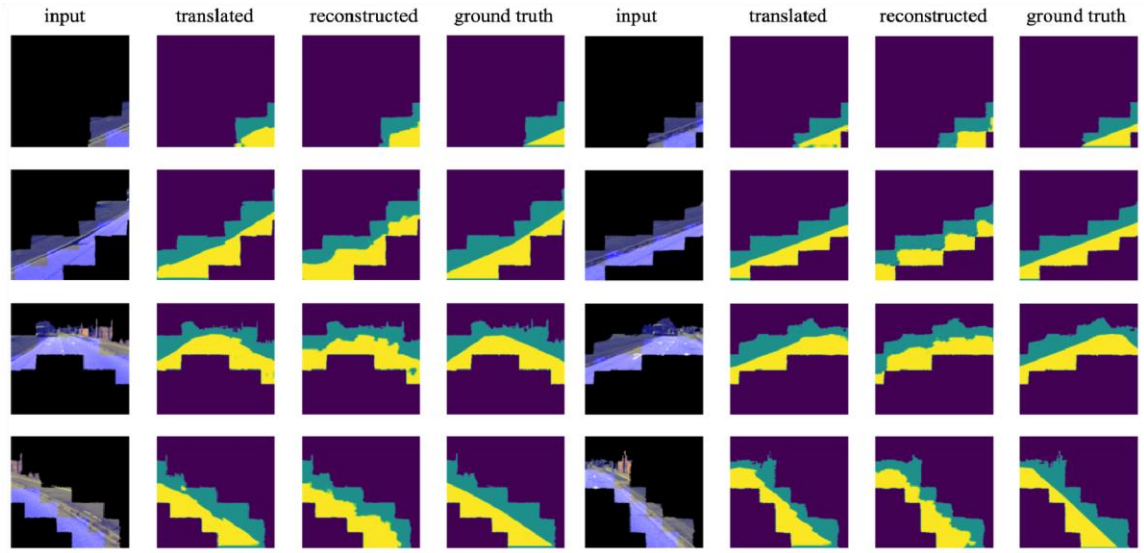


Figure 9. Road segmentation generation from a 4-D domain to unpaired ground truth domain.

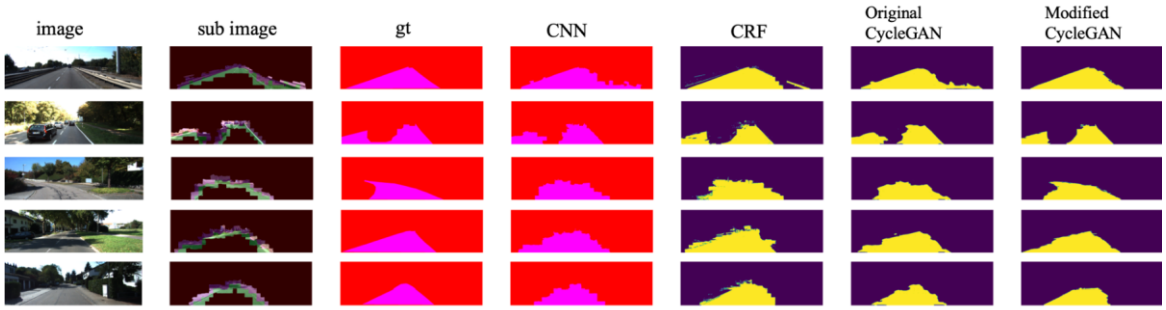


Figure 10. Road segmentation results from KITTI validation set based on all evaluated methods.

D. Run-Time Analysis

In this section, we discuss the computational cost and the required processing time for final approach including the coarse segmentation and the refinement method. First, we analyze the computational cost of our modified CycleGAN compared to the original CycleGAN in terms of the number of parameters(#Params) and the floating-point operations (#FLOPs), that can reveal, which approach is faster and needs fewer computational effort. Table V shows the values of these two evaluation indexes, which are calculated using the profiler library of TensorFlow framework (version 2.6). As already explained in Section.III-B2, we improved the generator networks by modifying the residual blocks into smaller structure and reducing the number of the blocks from nine to six. As shown in Table V, our proposed method has approximately a reduction of 5.58 times and 4.42 times in *#Params* and *#FLOPs* respectively compared to the original CycleGAN. Second, we compare both road segmentation accuracy and average processing time for one image in current study with some of the state-of-arts methods in semantic segmentation. All information is summarized in Table VI. AI Scores in the table rank the ML computational power of GPUs.

TABLE V. THE COMPUTATIONAL COST OF OUR PROPOSED METHOD COMPARED TO THE ORIGINAL CYCLEGAN

Method	#Params	#FLOPs
Original CycleGAN	11.38×10^6	98.85×10^9
Our Modified CycleGAN	2.04×10^6	22.32×10^9

All results are provided in KITTI URBAHN Multi-line ROAD test set. Using super pixels and simple CNN network combined with optimized cycle consistent adversarial technique, which is applied only on the small portion of pixels surrounding the road contour distinctly reduces the computational complexity. Our proposed method required 0.10*s*, that all in together with the required time for super pixel creation and CNN network, the total runtime of our approach amounts to 0.12*s* per image with GPU specification. To sum up, we can emphasize that our designed approach is compatible for real-time systems. Contrary to the various models, it can reduce the performance degradation, while increase the processing speed and keep a reasonable trade-off between accuracy and time-efficiency.

TABLE VI. THE KITTI ROAD SEGMENTATION PERFORMANCE IN % ON URBAN MULTIPLE MARKED (UMM) CATEGORY. ONLY RESULTS OF PUBLISHED METHODS ARE REPORTED. THE AI-SCORES:" HTTPS: //AI − BENCHMARK.COM/RANKINGDEEPLEARNING.HTML"

Method	Processor	MaxF	AP	AI-Scores	Runtime(s)
LidCamNet [20]	NVIDIA GTX1080 GPUs	96.03 %	93.93 %	17383	0.15
RBNet [40]	NVIDIA Tesla K20c 5 GB	94.97 %	91.49 %	-	0.18
LODNN [21]	NVIDIA GTX980Ti GPU, 6GB	94.07 %	92.03%	16038	0.018
UP CONV POLY [41]	NVIDIA Titan X GPU	93.83 %	90.47	20089	0.083
Ours (SP_CNN) [8]	Intel(R) Core (TM) i7-4790K CPU @4GHz	85.07 %	79.86 %	1400	**0.019**
Ours (Modified CycleGAN)	NVIDIA T4 GPU	91.80 %	89.25 %	14558	**0.10**

VI. CONCLUSION

In this study, we discussed an approach to enhance the segmentation map obtained from a super-pixel based convolutional neural network applied for semantic road segmentation. We focused to improve the segmentation accuracy, especially along the road border in challenging conditions, such as shadow on the road surface, illumination changes or similarity with neighboring patterns like sidewalls. We formulated the problem by proposing a semi-supervised modified CycleGAN approach. We defined a new network, that adjusted the original CycleGAN to refine the road segmentation. The proposed modified CycleGAN varied from original one in the generator structure and new added paired loss in total objective function. This enhanced the segmentation performance and reduced the computational effort compared to the original CycleGAN. The comparative experiments using KITTI database shows that our algorithm achieves comparable results in the semantic road segmentation with other state-of-the-art methods. In future, we plan to evaluate this approach for more than 2 classes and extend the pixel-wise classification to different objects such as sidewalls, lanes, traffic signs, vehicles, buildings, sky, etc. In addition, we will focus more on processing time of our refinement model along with our already proposed method based on super pixel-CNN for operating in a real time system.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

Farnoush Zohourian contributed to the conceptualization and methodology, implementation, and writing the paper. Josef Pauli supervised the project, reviewing and editing. All the authors approved the final version.

REFERENCES

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436-444, 2015.

[2] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85-117, 2015.

[3] O. Russakovsky, *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211-252, 2015.

[4] J. Gu, *et al.*, "Recent advances in convolutional neural networks," *Pattern Recognition*, vol. 77, pp. 354-377, 2018.

[5] Z. Q. Zhao, P. Zheng, S. T. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11, pp. 3212-3232, 2019.

[6] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431-3440.

[7] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions (2015)," arXiv preprint arXiv:1511.07122, 2016.

[8] F. Zohourian, B. Antic, J. Siegemund, M. Meuter, and J. Pauli, "Superpixel-based road segmentation for real-time systems using cnn." in *VISIGRAPP (5: VISAPP)*, 2018, pp. 257-265.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84-90, 2017.

[10] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1915-1929, 2013.

[11] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, "Slic superpixels," Tech. Rep., 2010.

[12] F. Zohourian, J. Siegemund, M. Meuter, and J. Pauli, "Efficient fine-grained road segmentation using superpixel-based cnn and crf models," in *Proc. International Conference on Pattern Recognition and Artificial Intelligence*, 2018, pp. 512-517.

[13] F. Zohourian, J. Siegemund, M. Meuter, and J. Pauli, "Efficient fine-grained road segmentation using superpixel-based cnn and crf models," arXiv preprint arXiv:2207.02844, 2022.

[14] J. Lafferty, A. McCallum, and F. C. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proc. the 18th International Conference on Machine Learning*, 2001.

[15] P. Krahenbuhl and V. Koltun, "Efficient inference in fully connected crfs with gaussian edge potentials," *Advances in Neural Information Processing Systems*, vol. 24, pp. 109-117, 2011.
[16] P. Kohli, *et al.*, "Robust higher order potentials for enforcing label consistency," *International Journal of Computer Vision*, vol. 82, no. 3, pp. 302-324, 2009.
[17] L. Ladicky, C. Russell, P. Kohli, and P. H. Torr, "Associative hierarchical crfs for object class image segmentation," in *Proc. IEEE 12th International Conference on Computer Vision*, 2009, pp. 739-746.
[18] J. Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. the IEEE International Conference on Computer Vision*, 2017, pp. 2223-2232.
[19] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," arXiv preprint arXiv:1511.07122, 2015.
[20] L. Caltagirone, M. Bellone, L. Svensson, and M. Wahde, "Lidar-camera fusion for road detection using fully convolutional neural networks," *Robotics and Autonomous Systems*, vol. 111, pp. 125-131, 2019.
[21] L. Caltagirone, S. Scheidegger, L. Svensson, and M. Wahde, "Fast lidar-based road detection using fully convolutional neural networks," in *Proc. IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 1019-1024.
[22] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2015, pp. 234-241.
[23] M. Drozdzal, E. Vorontsov, G. Chartrand, S. Kadoury, and C. Pal, "The importance of skip connections in biomedical image segmentation," in *Deep Learning and Data Labeling for Medical Applications*, Springer, 2016, pp. 179-187.
[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770-778.
[25] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected crfs," arXiv preprint arXiv:1412.7062, 2014.
[26] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. Torr, "Conditional random fields as recurrent neural networks," in *Proc. the IEEE International Conference on Computer Vision*, 2015, pp. 1529-1537.
[27] G. Lin, C. Shen, A. V. D. Hengel, and I. Reid, "Efficient piecewise training of deep structured models for semantic segmentation," in *Proc. the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3194-3203.
[28] P. Isola, J. Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-Image translation with conditional adversarial networks," in *Proc. the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1125-1134.
[29] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. WardeFarley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139-144, 2020.
[30] M. Mirza and S. Osindero, "Conditional generative adversarial nets," arXiv preprint arXiv:1411.1784, 2014.
[31] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," arXiv preprint arXiv:1511.06434, 2015.
[32] P. Luc, C. Couprie, S. Chintala, and J. Verbeek, "Semantic segmentation using adversarial networks," arXiv preprint arXiv:1611.08408, 2016.
[33] S. Hartmann, M. Weinmann, R. Wessel, and R. Klein, "Streetgan: Towards road network synthesis with generative adversarial networks," in *Proc. 25th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision in cooperation with EUROGRAPHICS Association*, 2017.
[34] S. W. Cho, N. R. Baek, J. H. Koo, M. Arsalan, and K. R. Park, "Semantic segmentation with low light images by modified cyclegan-based image enhancement," *IEEE Access*, vol. 8, pp. 93561-93585, 2020.
[35] D. Costea, A. Marcu, E. Slusanschi, and M. Leordeanu, "Creating roadmaps in aerial images with generative adversarial networks and smoothing-based optimization," in *Proc. the IEEE International Conference on Computer Vision Workshops*, 2017, pp. 2100-2109.
[36] J. Fritsch, T. Kuehnl, and A. Geiger, "A new performance measure and evaluation benchmark for road detection algorithms," in *Proc. 16th International IEEE Conference on Intelligent Transportation Systems*, 2013, pp. 1693-1700.
[37] X. Ren and J. Malik, "Learning a classification model for segmentation," in *Proc. IEEE International Conference on Computer Vision*, 2003, pp. 10-17.
[38] T. Ojala, M. Pietikainen, and D. Harwood, "Performance evaluation of texture measures with classification based on kullback discrimination of distributions," in *Proc. 12th International Conference on Pattern Recognition*, 1994, pp. 582-585.
[39] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization. (dec)," arXiv preprint arXiv:1412.6980, 2014.
[40] Z. Chen and Z. Chen, "Rbnet: A deep neural network for unified road and road boundary detection," in *Proc. International Conference on Neural Information Processing*, 2017, pp. 677-687.
[41] G. L. Oliveira, W. Burgard, and T. Brox, "Efficient deep models for monocular road segmentation," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2016, pp. 4885-4891.

Farnoush Zohourian received her M.Eng. in Computer Engineering and Cognitive Systems from Duisburg-Essen university of Germany in 2015. Her master thesis was in the field of medical image processing. She is a Ph.D. student at the chair of Intelligent Systems at the university of Duisburg-Essen, Germany. She has worked from 2015 to 2018 for APTIV (formerly Delphi), Wuppertal, Germany, as an External Doctorate Researcher in Computer vision and AI, working on the utilization of varied sensory inputs like camera and radar for an improved real-time scene parsing in vehicles for autonomous driving industry.

Josef Pauli is a professor at Duisburg-Essen university, Faculty of Engineering, Department of Computer Science and Applied Cognitive, chair of Intelligent Systems. His research interests include Computer/Robot Vision, Cognitive Robot Systems, Neural Network Learning. Current application areas are image processing for medical technology, image and hyper-spectral data analysis for surface inspection, deep learning in particle measurement technology, robot navigation in highly dynamic environments.