Superpixel-based Road Segmentation for Real-time Systems using CNN

Farnoush Zohourian[1], Borislav Antic[2], Jan Siegemund[2], Mirko Meuter[2] and Josef Pauli[1]
[1]*University of Duisburg-Essen, Department of Computer Science and Applied Cognitive Science, 47048, Duisburg, Germany*
[2]*Delphi Electronics & Saftey, D-42119, Wuppertal, Germany*

Keywords: Superpixel, Semantic Segmentation, CNN, Deep Learning, Road Segmentation.

Abstract: Convolutional Neural Networks (CNN) contributed considerable improvements for image segmentation tasks in the field of computer vision. Despite their success, an inherent challenge is the trade-off between accuracy and computational cost. The high computational efforts for large networks operating on the image's pixel grid makes them ineligible for many real time applications such as various Advanced Driver Assistance Systems (ADAS). In this work, we propose a novel CNN approach, based on the combination of super-pixels and high dimensional feature channels applied for road segmentation. The core idea is to reduce the computational complexity by segmenting the image into homogeneous regions (superpixels) and feed image descriptors extracted from these regions into a CNN rather than working on the pixel grid directly. To enable the necessary convolutional operations on the irregular arranged superpixels, we introduce a lattice projection scheme as part of the superpixel creation method, which composes neighbourhood relations and forces the topology to stay fixed during the segmentation process. Reducing the input to the superpixel domain allows the CNN's structure to stay small and efficient to compute while keeping the advantage of convolutional layers. The method is generic and can be easily generalized for segmentation tasks other than road segmentation.

1 INTRODUCTION

One of the long-lasting goals of computer vision is the automated scene understanding from a variety of images. Exposing image specification is useful for applications, like image editing, image search and environment perception for autonomous vehicles. Detecting objects like roads, pedestrians, vehicles, traffic signs, etc. is important for many driver-less cars and driver assistance systems. Due to the variability of different factors like colour, shape, illumination and shadows or obstacles on the road surface, the road detection is a challenging problem. The state of arts techniques to solve this problem are mainly based on deep learning and Convolutional Neural Networks (CNNs) (LeCun et al., 2015; Schmidhuber, 2015). These methods enable towards better visual understanding by applying a semantic segmentation process in which each pixel is assigned to an object category. The segmentation result provides meaningful information to support higher level scene understanding tasks.

Currently, there are two major approaches to train CNN-based image processing systems. The two approaches differ with respect to the input data model. One of the approaches is based on a patch-wise analysis of the images, i.e. an extraction and classification of rectangular regions having a fixed size for every single image (Ciresan et al., 2012; Farabet et al., 2013; Ganin and Lempitsky, 2014; Ning et al., 2005). The other one is based on full image resolution, wherein all pixels of an image in the original size are analyzed (Long et al., 2015). Most recent improvements in both CNN-based methods were accomplished by increasing the network size (Simonyan and Zisserman, 2014; He et al., 2016), whereas deeper networks provoke large computational costs that make them unsuitable for embedded devices in driver assistance systems.

In the current work we apply a superpixel-based CNN method for the specific application of pixel-wise road segmentation that uses superpixels as input data model. To the best of our knowledge, it is the first time that irregular superpixels with regular lattice projection for Convolutional purpose is given as input data model into a CNN network. The proposed method comprises the following steps: first, segmenting the image into superpixels, wherein the superpixels are coherent image regions comprising a plurality of

Zohourian, F., Antic, B., Siegemund, J., Meuter, M. and Pauli, J.
Superpixel-based Road Segmentation for Real-time Systems using CNN.
DOI: 10.5220/0006612002570265
In *Proceedings of the 13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2018) - Volume 5: VISAPP*, pages 257-265
ISBN: 978-989-758-290-5

pixels having similar image features. Then determining image descriptors for the superpixels, wherein each image descriptor comprises a plurality of image features. The superpixels are assigned to corresponding positions of a regular grid structure extending across the image. This lattice together with the image descriptors are fed to the convolutional network based on the assignment to classify the superpixels of the image according to semantic categories.

Feeding a network with almost well segmented "superpixel" units enables the network to learn local information like contrast, shape, texture, etc. much better rather than using raw image pixels. In Comparison to (Long et al., 2015) that is based on full resolution input data and has a deep convolutional network layering (e.g: vgg-19), our method combines larger basic units "super-pixels" with simple network structure. This results in significant reduction of the computational costs for a densely labelled map prediction. Contrary to patch-based semantic segmentation approaches (Farabet et al., 2013), information about spatial context in the proposed method can be preserved preferably due to the usage of superpixels.

The remainder of this paper is organized as follows. Sec.2 presents an overview over related work. Sec.3 focuses on the proposed superpixel-based Convolutional Neural Network approach. We describe the superpixel segmentation method and how it is embedded in Sec. 3.1. Then the feature selection is discussed in Sec. 3.4. The network architecture and parameters is argued further in Sec.3.5. Sec.4 presents experiments and results. Sec.5 draws a conclusions and discusses future works and required improvements.

2 RELATED WORK

Deep Learning is a machine learning concept to model higher level features by learning hierarchies of lower-level features (LeCun et al., 2015). Convolutional neural network is a deep learning technique which has been effectively applied in different computer vision applications, such as n image classification (Krizhevsky et al., 2012), object detection (Girshick et al., 2014; P. Sermanet, 2013), scene labeling (Farabet et al., 2013; Chen et al., 2014). The state of-the-art methods for semantic segmentation are generally fully convolutional networks (Long et al., 2015) which are directly applied to the whole image. This method has been improved further in several newer approaches such as "DeepLab" (Chen et al., 2014). However, most of the per pixel labeling methods are too expensive for embedded applications and they require powerful GPUs to be fast enough for achieving the real-time performance.

In this paper we combine superpixels segmentation with convolutional neural network. Several other methods benefit from this combination too. Gadde (Gadde et al., 2016) embedded superpixels into a newly defined layer that he names "Bilateral Inception" which acts as an edge preserving filter. This layer is substituted with a fully connected layer and propagates label information between superpixels. This results in better segmentation than in exclusively pixel-wise approaches. However, this network still uses full resolution images as the inputs. SuperCNN (He et al., 2015) is a neural network based approach for salient object detection. A sequence of superpixels, instead of a 2D image pattern, is fed into this network. Contrary to this 1-D inputs, our proposed method uses a 2D-grid of superpixels as input to the network which allows for easier extraction of the neighborhood information by convolutional network.

3 SUPERPIXEL-BASED CONVOLUTIONAL NEURAL NETWORK

This work addresses the task of road segmentation from urban scene images. We tackle the problem by segmenting the images into superpixels, deriving road relevant features, and constructing a rational feature model fed into CNN to segment road regions. Figure 1 displays the architecture of our method. Superpixels are extracted using the Simple Linear Iterative Clustering (SLIC) algorithm (See Sec.3.1). The main features extracted from the superpixels are colour, texture, location and histogram of gradients (See Sec.3.4). The applied Convolutional neural network (CNN) model used to segment road and non-road parts is described in Sec. 3.5.

3.1 Superpixel Extraction

Superpixel segmentation methods partition an image into homogeneous regions. Comparing to pixels, superpixels are perceptually more meaningful (Ren and Malik, 2003). Two main benefits of well extracted superpixel properties, that encourage us to choose them as basic units in our approach, are described as follows:

- **Accuracy.** Well segmented superpixels can store more compact information about the color, texture, etc. and they are less ambiguous and sensitive to noise than features extraction at pixel level.

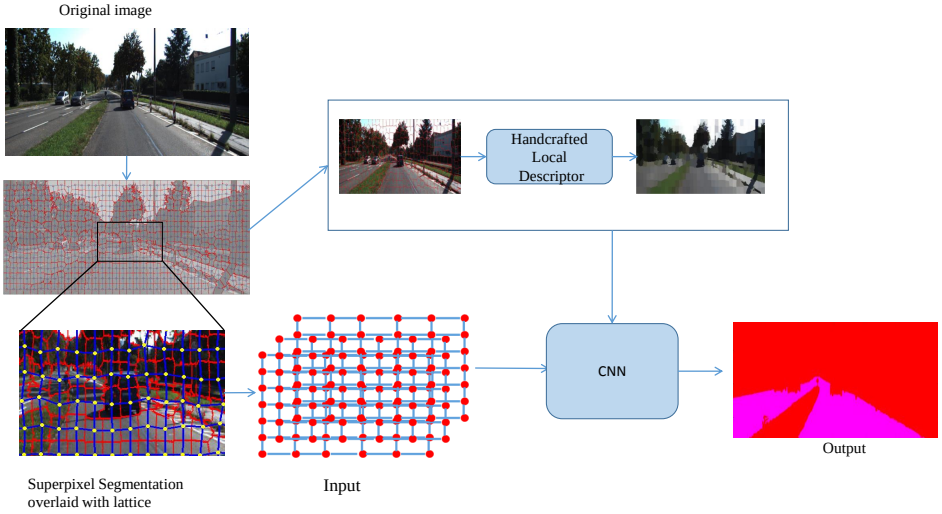

Figure 1: The Proposed SP-CNN Architecture: The original image is segmented into superpixels. Then local descriptors for the superpixels are extracted. The superpixel descriptors are assigned to the respective node in the lattice projection and are fed into a CNN to classify the superpixel according to the semantic categories.

They can preserve the object structures and adjust well to the object contours.

- **Efficiency.** Dealing with millions of pixels and their parameters in large systems can be costly, whereas using superpixels can greatly reduce the model complexity and computation cost especially for real time systems.

On the one hand, using regular superpixel segmentation (superpixels are arranged in a grid structure) methods can preserve the topology. However,regularity mostly inhibits maximum homogeneity of the 'texture' inside each superpixel. On the other hand using irregular superpixel segmentation methods result in different sizes and irregularly shaped boundaries which are not directly applicable as input data for a convolutional network. Therefore, a regular topology is needed to convolve the input data with kernels, and irregular superpixels need to be "realigned" such that a proper input into a convolutional network is possible.

3.2 Original SLIC Method

We used SLIC algorithm (Achanta et al., 2010) for superpixel segmentation. SLIC initiates with equally-sized superpixels arranged in a grid structure. The similarity between pixels is calculated based on two criteria: spectral similarity and spatial proximity that enforces compactness and regularity in the superpixel shapes. The main idea of this approach is to limit the search space to a region proportional to the desired SP size which reduces considerably the calculation time. Superpixels grow by measuring the (spectral-spatial) distance between each pixel to its cluster center and then update the cluster centers based on K-means algorithm. Input parameters of this method are input images with N pixels, a desired number of approximately equally-sized superpixels K and a weighted distance m that combines spectral and spatial proximity to control the compactness of the superpixels. The superpixels are initiated with roughly equal size of $S = \sqrt{N/K}$ and the spatial extent of any superpixel is approximately in S^2 neighborhood.

SLIC only computes distances from each cluster center to pixels within $2S \times 2S$ area that assures us pixels that are associated with this cluster center lie within a $2S \times 2S$ area around the superpixel center on the xy plane and not farther.It leads to reduction of complexity and distance computations and independence from the number of superpixels (Achanta et al., 2010).

For K desired superpixel each cluster centers are specified with a 5D vector. $C_i = [l_i, a_i, b_i, x_i, y_i]$ with $i = [1, K]$ at regular grid intervals S. Distance measurement for the pixels nearest to each cluster center is based on measuring the Euclidean distances in CIELAB color space and spatial pixel distances, however inconsistency in clustering behavior for different superpixel sizes should be controlled (m in Eq.3.1). Larger m resulting superpixels are more compact and smaller m aimed better segmentation but more irregular size and shape. Euclidean distances in CIELAB color space are visually meaningful for small distances. Outweighing pixel color similarities (m) prevents the spatial pixel distances from exceeding this perceptual color distance limit. Hence, instead of using a

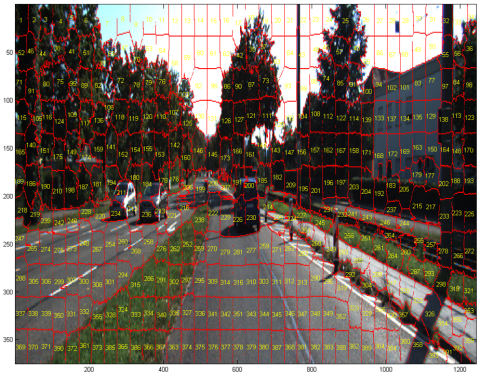

Figure 2: Super-Pixel Segmentation (After) Enforcement Connectivity.

simple Euclidean distance in the 5D space, distance measure D_s from each pixel K to the cluster center i defined as follows:

$$\begin{aligned} d_{color} &= \sqrt{(l_k - l_i)^2 + (a_k - a_i)^2 + (b_k - b_i)^2} \\ d_{Spatial} &= \sqrt{(x_k - x_i)^2 + (y_k - y_i)^2} \\ D_s &= d_{color} + \frac{m}{S} d_{Spatial} \end{aligned} \tag{3.1}$$

where D_s is the sum of the lab color space distance and the xy spatial distance normalized by the grid interval S. After superpixel segmentation, SLIC uses a method to enforce all superpixels to be connected and prevent too small areas or any islands or disconnected area. This method leads to non constant numbers of created superpixels making them unsuitable as direct CNN input model. This inconsistency in number of created superpixel would cause trouble for our proposed approach. To prevent this problem, we used a modified version of SLIC. This version changes the connectivity enforcement algorithm provided by SLIC to keep the number of superpixels constant. This help us to provide the same size of input data for our network while keep the strength of SLIC method for having almost homogeneous superpixels.

3.3 Adapted SLIC Method

In the modified version, we perform following steps for each superpixel. First we find the whole adjacent superpixels, and the label-connected components in 2-D for each superpixel. Then if a certain superpixel has more than one segment with the same label, we keep the larger one and merge the rest into the nearest superpixel which is picked up from neighborhood The nearest superpixel is computed based on euclidean distance between the center of sub-segment to the center of each adjacent segment. Contrary to the original version we do not remove any too small region with only one label-connected area.

The necessity to having a regular topology to be able to convolve the input data with kernels, motivated us to propose a superpixel lattice projection. The lattice is centered in the rectangular structure extracted from the first iteration of SLIC method (defined by the centers of the superpixels). This grid is directly used to establish a regular topology for the final superpixels, i. e. the superpixels generated by the last iteration step.

a) Before a) After

Figure 3: Superpixel segmentation before and after enforce connectivity. It embeds disconnected superpixels into the nearest neighburhood.

3.4 Feature Selection

Feature selection acts as a preprocessing step that enables us to model relevant object characteristics in the image. We tested different combinations of features and decided on a particular combination, which gives the best performance. We considered three different feature groups.

Color Feature. It probably is one of the most informative features often applied by the human visual system for object and scene image classification. We used different color spaces RGB, Lab, HSV and computed the average values of all pixels within each superpixel for each color channel separately. Defining descriptors in different color spaces usually improves the description of object and texture image categories (Verma et al., 2010). They are more robust against image variations such as lighting changes, rotation, and occlusions (Burghouts and Geusebroek, 2009).

Position Feature. Generally, road area can be detected from its surroundings based on the color fea-

ture, however the appearance of shadows or similar pattern to the roads like sidewalks, leads to the relatively difficult adequate prediction. As the color to class distribution may vary for different positions and road is typically located in the bottom, we considered "position" as a second type of representative feature addition to the color feature. The average of vertical coordinates of pixels in each superpixel is selected as the location feature.

Local Binary Pattern (LBP). Some local information like texture and shape can contribute to object and scene image classification. Investigation of our images represent significant changes between the texture of a road and its surroundings, especially compared to houses and trees. Roads tend to be flat and smooth, whereas trees and houses have more complex and compound textures. We use Local Binary Patterns (LBP)(Ojala et al., 1994) to computes correlation and disparity among pixels inside each superpixel. LBP showed to be promising for recognition and classification of texture images.

For each superpixel a high dimensional feature descriptor is defined. Each of the image descriptors comprises 69 image features consisting of 9 color, 1 position and 59 LBP. This provides for high accuracy and reliability. The provided data model is fed to a simple convolutional network presented in the following.

3.5 Network Architecture and Choice of Hyper-parameters

Contrary to most state of the art CNN-based semantic segmentation approaches, our proposed method does not require a complex network architecture to handle large image context, due to the presegmentation which improves computational time. Our proposed network structure consists of 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 superpixel lattice (See Sec.3.1) on each image with size of H/S and W/S,where S is initial superpixel size and W,H are image width and hight. The output is a set of three numbers to indicate which of the three classes of the *road*, *non-road* or *un-labeled* they belong to. We explain them more in detail in Sec. 4

The weights in the fully connected layers as well as the weights of the convolution masks are initialized randomly with setting bias to zero. Softmaxlog is used as loss function and optimization is done with stochastic gradient descent. The learning rate and all other hyper-parameters are optimized on a validation set (learning rate is set to 0.1×10^{-4}, and weight decay is set to 0.0001). We used momentum 0.9 and mini-batches of size 50. All of the above parameters are empirically set to achieve a reasonably good reconstruction loss and error rate and are held constant across all datasets.

4 EXPERIMENTS AND RESULTS

We evaluate our method using two widely-used challenging datasets comprising urban scenarios, i.e., the KITTI (Fritsch et al., 2013) and the Cityscapes (M. Cordts, 2016) datasets. In the following we give a brief description of the datasets followed by the evaluation results.

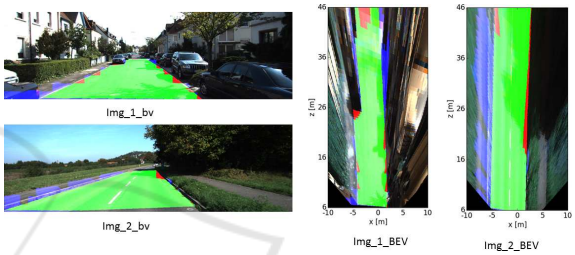

Figure 4: Road segmentation result from official KITTI test set in baseline and bird eye view perspectives. Here, red denotes false negatives, blue is false positives and green represents true positives.

4.1 Datasets

KITTI Dataset

KITTI Road Estimation data set comprises 502 8-bits RGB images splits in train, validation and test sets with ground truth label for three semantic classes. KITTI is one of the most popular datasets for road segmentation in urban scene applications. 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 vary with the width lying in $[1226, 1238, 1241, 1242]$ and their height in $[370, 374, 375, 376]$. We selected 20% of train set images from 3 different categories UM,UMM,uu for the validation set. These images are completely from different video sequences which are not part of the train set.

For KITTI dataset experiment, we chose SLIC parameters $K = 400, m = 35$ resulting in 396 superpixels in each image projected to a 11×36 lattice for CNN input.

Cityscape Dataset
Cityscapes is another new dataset for scene understanding in urban environments (M. Cordts, 2016), comprising pixel-wise ground truth label for nineteen semantic categories including road, car, pedestrian, bicycle, etc. The dataset contains 2975 training, 500 validation, and 1525 test images. All of the images in this dataset are in the same size of 1024×2048 pixels. For evaluation, we report our results on the validation set. In this paper we focused only on road segmentation. Therefore we changed all labels except road to background's label and evaluated our approach only for road segmentation.
Experiment on Cityscape is done by choosing SLIC parameters $K = 2000, m = 10$ on half resolution images resulting in 2048 superpixels for each image and projected to a 32×64 lattice as the input data to CNN.

4.2 Evaluation Results

For training and testing we used the following hardware specifications, CPU: Intel(R) Core(TM) i7-4790K @4GHz. The feature vectors extracted from superpixels were normalized by mean and standard deviation for each channel before fed into the CNN. For this, mean and standard deviation was computed on the training set and applied for both, training and evaluation. Our original task was to segment road from non-road (background). We have evaluated the potential of the proposed approach based on its accuracy in two domains. These are the accuracy on the native pixel gird and the super pixel grid evaluated on the image perspective and a birds eye projection provided by KITTI dataset (Sec.4.1). For evaluation in the superpixel domain, the ground truth for each Superpixel is defined based on the majority pixel-labels inside the superpixel. If the majority reaches less then 80% we assigned new label named *Un-labeled* to our class labeling, yielding three classes in total: *road*, *non-road*, *unlabeled*.

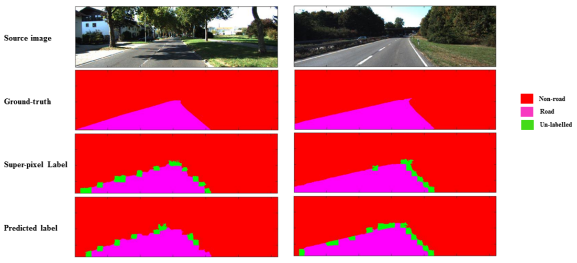

Figure 5: Road segmentation with SP-CNN for two samples: the first horizontal row shows input images. Ground-truth based on pixels and superpixels are shown in the second and third lines.The last row shows the predicted results based on superpixel labeling.

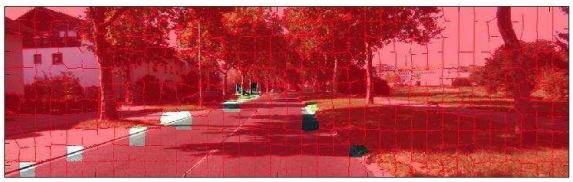

Figure 6: Wrong-predicted area is shown as non-red superpixels.

Evaluation on Image Perspective. We followed the evaluation scheme presented in the KITTI dataset (See (Fritsch et al., 2013)). The ground truth for the test data of the KITTI data set is not publicly available, hence we used the validation set to be able to evaluate our approach on the image perspective. Table 1 shows the average evaluation results on the validation set for all urban categories in KITTI , and cityscape regarding to both superpixel and pixel ground-truths using proposed model and parameters as described in sections 3 and 3.5. We consider *road* as the positive class and *non-road* and *un-labeled* as the negative class.

Figure 5 shows two representative results. Whilst street is nicely segmented, there are a few false detections on the road border. The more detailed view in figure 6 reveals that they originate mostly from inaccurately formed superpixels, which stayed to compact in this scenario rather than clinging to the road border. Further, one superpixel was fooled by a shadow covering the street. These deficiencies are resolved by a refinement step which is not covered in the current paper.

We compared our results by the idea of fitting the super pixels to a prefixed size of super pixels (Patch). To do so, we first segmented all images in KITTI dataset with the same number of irregular superpixel segmentation (396 SP for each image) and then extracted the proposed feature vector from each patch. Table 2 summarized the experiment results. Small difference of around 5% between the overall F-measurement in both cases, happened due to the high correctness of the non-road segmentation, whereas a large gap of almost 14% of average precision emphasizes the low ability of the road segmentation in patch-wise method. This gets even worse by existing shadow or obstacle on the road surface (See Figure:7).

The idea of using resized image as an input model instead of feeding with super-pixels would have two drawbacks. First, you loose more information by decreasing the resolution which makes the prediction worse specially for images with shadow on the road or distinguishing road from sidewalks. Especially if you want to reach the small size of 11x36 of projected lattice. Secondly, our final aim is not only segmenta-

Table 1: Evaluation Results on KITTI and Cityscape validation sets.For the evaluation of the following experiments we used these metrics: accuracy (ACC), F-measure (MaxF), precision (PRE), recall (REC), false positive rate (FPR), false negative rate (FNR).

Dataset	Benchmark	ACC	MaxF	PRE	REC	FPR	FNR
KITTI	Superpixel	97.37%	92.57%	94.21%	91.62%	5.79%	1.86%
	pixel	96.50%	90.08%	92.80%	88.07%	7.20%	2.65%
CityScape	Superpixel	95.75%	92.44%	89.08%	96.76%	10.92%	1.14%
	pixel	94.10%	90.01%	84.41%	97.14%	15.59%	1.89%

Table 2: Evaluation results on both regular and irregular shape of superpixels on KITTI validation set.

Method	Benchmark	MaxF	PRE	REC
Fixed-size SP	Superpixel	87.52%	80.46%	98.07%
	pixel	85.19%	**78.85%**	94.70%
irregular AP	Superpixel	92.57%	94.21%	91.62%
	pixel	90.08%	**92.80%**	88.07%

Table 3: Evaluation Results on KITTI Test set.

Benchmark	MaxF	AP	PRE	REC	FPR	FNR
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 %

Table 4: **CPU**-based Computational Run-Time in seconds per frame for both KITTI and City-scape image resolution: 1)number of superpixels(No_SP) 2)superpixel segmentation(SP),3)Feature Extraction(FE), and 4)Network (CNN).

Dataset	No_SP	SP	FE	CNN	Total
KITTI	396	0.2s	0.2s	0.01s	0.41s
CityScape	2048	0.3s	0.6s	0.2s	1.1s

Table 5: Kitti road benchmark results (in %) on urban road category. only results of published methods are reported. LODNN: (Caltagirone et al., 2017), UP_CONV_POLY (Oliveira et al., 2016), DDN (Mohan, 2014).

Method	Processor	MaxF	AP	Runtime(s)
LODNN	NVIDIA GTX980Ti GPU, 6GB memory	94.07 %	92.03%	0.018
UP_CONV_POLY	NVIDIA Titan X GPU.	93.83 %	90.47	0.083
DDN	NVIDIA GTX980Ti GPU, 6GB memory	93.43 %	89.67 %	2
Ours (un-Optimized runtime)	Intel(R) Core(TM) i7-4790K CPU @4GHz	82.36 %	72.31 %	**0.41**
Ours (Optimized runtime)	Intel(R) Core(TM) i7-3770 CPU @ 3.40GHz	82.36 %	72.31 %	**0.019**

Table 6: Intersection of union(IoU) results for road class on Cityscapes testing set. CRF-RNN (Zheng et al., 2015), Deep-Lab (Chen et al., 2016), FCN (Long et al., 2015).

Method	road (IoU)
DeepLab	97.9%
FCN	97.4%
CRF-RNN	96.3%
Ours	**90.08%**

tion road but also more objects in the images which are currently out of scope of this paper. Resizing of the image is not a suitable solution to segment small object like vehicle, pedestrian, etc. Even if you do this resizing via the down/up sampling you put much effort on computational time.

Evaluation on Birds Eye Perspective. For evaluation in birds-eye perspective in the KITTI benchmark the images are projected on the ground plane via the known camera geometry. The projection is resampled into a regular grid to provide the final image for pixel-wise comparison (Sec.4.1).

We expected the accuracy to drop compared to the image perspective evaluation, as the error induced by inaccurate superpixels on the road border (as mentioned above) spreads over a much larger region due to the ground projection. Table 3 shows the results which are split into the different road types (UM, UMM, UU, URBAN). The accuracy dropped as expected. We can observe the mentioned projection effect on the example results depicted in Figure 4.

Figure 7: Road segmentation prediction with SP-CNN for two samples based on: a) fixed-size superpixels (Patches), b) irregular superpixels.

Cityscape Evaluation. We first evaluate our approach on Cityscape validation set. The results are collected in Table 1. Since, the publicly available Cityscape benchmark evaluates 19 classes, comparison of the average result between our approach (with only road class) and the state of art methods is not fair. Therefore, we compare Per-class scores on Cityscapes testing set. Table 6 presents the results of intersection-over-union metric (IOU) (M. Cordts, 2016) for some of latest approaches on Cityscape. Although, we achieved lower performance among the state of art methods, but it is still comparable while achieved impressive low computational time.

Runtime. Most of the state of the art methods in semantic segmentation are based on GPU and powerful hardware facilities which limit their application to CPU based embedded systems. Our approach using superpixels and simple CNN network aimed real-time performance, which reduces the computational complexity to ease the integration of proposed method into ADAS and autonomous driving systems. We evaluated the computational time for both KITTI and Cityscape datasets in both train and test time. Experiments conducted on KITTI ROAD dataset required only 4.3s training time per epoch for one image (see Sec. 3.5). For testing it achieved the total run-time of 0.41s per image including SP segmentation(0.2s), Feature extraction(LBP 0.2s + position 0.002s) and CNN(0.009s) based on CPU specification and Matlab implementation without any parallel processing (Sec.4.2). For the preprocessing steps (superpixel segmentation and feature extraction), we used non optimized Matlab implementations for experimental reasons. However, runtime optimized versions for superpixel segmentation in CPU processor (Neubert, 2015) reach a performance of 0.008s for an image segmentation into 400 superpixels. Further, an optimized version for LBP feature extraction (López et al., 2014) can reach 0.001s. Using both bears the potential to squeeze the total runtime to 0.019s, which is very fast for semantic segmentation without no usage of GPU and thus, allows for embedding in real time systems. In Table 5 we compared our approach with some of the state of the art methods in semantic segmentation in both accuracy and computational time. All results provided in KITTI URBAHN ROAD test set. Although, the state-of-the-art method (Caltagirone et al., 2017) have a run-time 0.018s, but their method implemented in torch and uses NVIDIA GTX980Ti GPU with 6GB memory, which is quite fast processor, whereas the proposed method uses 4core CPU. The reasoning of low evaluated **MaxF** of the proposed approach which is happened due to the evaluation on Birds Eye Perspective 4.2 could be improved by a post processing step up to the 96% for KITTI and 94% for cityscape dataset that is out of scope of this paper.

To sum up, we can emphasize this point that our approach is compatible for real-time systems with a reasonable trade-off between accuracy and timing cost based on very cheap hardware facilities that make it very fast in both training and testing parts.

5 CONCLUSION AND FUTURE WORKS

We have presented a superpixel-based convolutional neural network for road segmentation. Our main goal focuses on a strategy to reduce the runtime for a pixel-wise classification while still achieving a high level of accuracy to make our approach suitable for deployment on embedded devices for ADAS applications. The core idea is to use superpixel units instead of pixel units as input for a CNN. The proposed method projects a superpixel segmentation on a regular lattice structure to preserve the topology and allow for convolution operations. The object characteristics are extracted from almost homogeneous and irregular-shaped superpixel units. Evaluation show promising accuracy and efficiency in segmentation tasks and significantly reduces the required number of predictions in runtime. This proposed superpixel-wise mechanism can be applied not only for road segmentation but also for semantic understanding of general objects. Future work will initially focus to eliminate the limitation introduced by the so far non-revisable decision of the superpixel segmentation, especially at the road boundary. This might be done by a post processing refinement of either all border superpixels or those marked by the CNN via an attention control.We further plan to improve the discrimination of the road pattern in special examples with challenging conditions; such as shadow on road surface, illumination changes or similarity with neighboring patterns like sidewalks. It is also promising to evaluate this ap-

proach for more than 2 classes and extend the pixel-wise classification to different objects such as side-walks,lane, traffic sign, vehicles, etc.

REFERENCES

Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., and Süsstrunk, S. (2010). Slic superpixels. Technical report.

Burghouts, G. J. and Geusebroek, J.-M. (2009). Performance evaluation of local colour invariants. *Computer Vision and Image Understanding*, 113(1):48–62.

Caltagirone, L., Scheidegger, S., Svensson, L., and Wahde, M. (2017). Fast lidar-based road detection using convolutional neural networks. *arXiv preprint arXiv:1703.03613*.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2014). Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2016). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv preprint arXiv:1606.00915*.

Ciresan, D., Giusti, A., Gambardella, L. M., and Schmidhuber, J. (2012). Deep neural networks segment neuronal membranes in electron microscopy images. In *Advances in neural information processing systems*, pages 2843–2851.

Farabet, C., Couprie, C., Najman, L., and LeCun, Y. (2013). Learning hierarchical features for scene labeling. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1915–1929.

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

Gadde, R., Jampani, V., Kiefel, M., Kappler, D., and Gehler, P. V. (2016). Superpixel convolutional networks using bilateral inceptions. In *European Conference on Computer Vision*, pages 597–613. Springer.

Ganin, Y. and Lempitsky, V. (2014). Nˆ 4-fields: Neural network nearest neighbor fields for image transforms. In *Asian Conference on Computer Vision*, pages 536–551. Springer.

Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778.

He, S., Lau, R. W., Liu, W., Huang, Z., and Yang, Q. (2015). Supercnn: A superpixelwise convolutional neural network for salient object detection. *International Journal of Computer Vision*, 115(3):330–344.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.

Long, J., Shelhamer, E., and Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440.

López, M. B., Nieto, A., Boutellier, J., Hannuksela, J., and Silvén, O. (2014). Evaluation of real-time lbp computing in multiple architectures. *Journal of Real-Time Image Processing*, pages 1–22.

M. Cordts, M. Omran, S. R. T. R. M. E. R. B. U. F. S. R. B. S. (2016). The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Mohan, R. (2014). Deep deconvolutional networks for scene parsing. *arXiv preprint arXiv:1411.4101*.

Neubert, P. (2015). Superpixels and their application for visual place recognition in changing environments.

Ning, F., Delhomme, D., LeCun, Y., Piano, F., Bottou, L., and Barbano, P. E. (2005). Toward automatic phenotyping of developing embryos from videos. *IEEE Transactions on Image Processing*, 14(9):1360–1371.

Ojala, T., Pietikainen, M., and Harwood, D. (1994). Performance evaluation of texture measures with classification based on kullback discrimination of distributions. In *Computer Vision & Image Processing*, pages 582–585. IEEE.

Oliveira, G. L., Burgard, W., and Brox, T. (2016). Efficient deep models for monocular road segmentation. In *Intelligent Robots and Systems (IROS),*, pages 4885–4891. IEEE.

P. Sermanet, D. Eigen, X. Z. M. M. R. F. Y. L. (2013). Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*.

Ren, X. and Malik, J. (2003). Learning a classification model for segmentation. In *ICCV*, volume 1, pages 10–17.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, 61:85–117.

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

Verma, A., Banerji, S., and Liu, C. (2010). A new color sift descriptor and methods for image category classification. In *International Congress on Computer Applications and Computational Science*, pages 4–6.

Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., Huang, C., and Torr, P. H. (2015). Conditional random fields as recurrent neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1529–1537.