# Aerial LaneNet: Lane-Marking Semantic Segmentation in Aerial Imagery Using Wavelet-Enhanced Cost-Sensitive Symmetric Fully Convolutional Neural Networks

Seyed Majid Azimi, Peter Fischer, Marco Körner, *Member, IEEE*, and Peter Reinartz, *Member, IEEE*

***Abstract*— The knowledge about the placement and appearance of lane markings is a prerequisite for the creation of maps with high precision, necessary for autonomous driving, infrastructure monitoring, lanewise traffic management, and urban planning. Lane markings are one of the important components of such maps. Lane markings convey the rules of roads to drivers. While these rules are learned by humans, an autonomous driving vehicle should be taught to learn them to localize itself. Therefore, accurate and reliable lane-marking semantic segmentation in the imagery of roads and highways is needed to achieve such goals. We use airborne imagery that can capture a large area in a short period of time by introducing an aerial lane marking data set. In this paper, we propose a symmetric fully convolutional neural network enhanced by wavelet transform in order to automatically carry out lane-marking segmentation in aerial imagery. Due to a heavily unbalanced problem in terms of a number of lane-marking pixels compared with background pixels, we use a customized loss function as well as a new type of data augmentation step. We achieve a high accuracy in pixelwise localization of lane markings compared with the state-of-the-art methods without using the third-party information. In this paper, we introduce the first high-quality data set used within our experiments, which contains a broad range of situations and classes of lane markings representative of today's transportation systems. This data set will be publicly available, and hence, it can be used as the benchmark data set for future algorithms within this domain.**

***Index Terms*— Aerial imagery, autonomous driving, fully convolutional neural networks (FCNNs), infrastructure monitoring, lane-marking segmentation, mapping, remote sensing, traffic monitoring, wavelet transform.**

Manuscript received March 1, 2018; revised July 28, 2018; accepted October 1, 2018. This work was supported by the German Aerospace Center (DLR), Münchenerstr 20, 82234 Weßling, Germany. *(Corresponding author: Seyed Majid Azimi.)*

S. M. Azimi is with the Department of Photogrammetry and Image Analysis, Remote Sensing Technology Institute, German Aerospace Center (DLR), 82234 Weßling, Germany, and also with the Department of Civil, Geo and Environmental Engineering, Chair of Remote Sensing Technology, Technical University of Munich, 80333 Munich, Germany (e-mail: seyedmajid.azimi@dlr.de).

P. Fischer was with the Department of Photogrammetry and Image Analysis, Remote Sensing Technology Institute, German Aerospace Center (DLR), 82234 Weßling, Germany. He is now with the Department of Sensor Data Fusion (I/EF-24), AUDI AG, Ingolstadt, Germany.

M. Körner is with the Department of Civil, Geo and Environmental Engineering, Chair of Remote Sensing Technology, Technical University of Munich, 80333 Munich, Germany.

P. Reinartz is with the Department of Photogrammetry and Image Analysis, Remote Sensing Technology Institute, German Aerospace Center (DLR), 82234 Weßling, Germany.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TGRS.2018.2878510

## I. Introduction

NOWADAYS, the detailed description of the public transportation network is essential for the generation of accurate road maps and lane-based models. A broad range of current services, e.g., navigation systems and assisted driving, rely on such information. Future applications, such as automated lanewise traffic monitoring, urban management, and city planning, are also asking for high-precision maps at centimeter-level accuracy, particularly built for autonomous driving applications that are called high-definition (HD) maps. At present, autonomous vehicles (AVs) are a research focus in computer vision and remote sensing. In order to achieve autonomy in AVs, one key factor is to localize the vehicle precisely. Very accurate maps containing the location of infrastructures, such as streets, sidewalks, traffic lights, and even lane markings, are a necessity for reaching the goal of fully autonomous driving. Advanced vehicle assistance system comprising features, such as vehicle navigation and lane departure warning, requires not only the road model information but also the precise road lane-marking data, e.g., the lane-marking types and their locations.

Besides the current omnipresent topic of autonomous driving, many more urgent topics can be addressed by HD maps. For instance, the traffic monitoring systems could benefit from the localization of lane markings as the base map. Information about lane-marking locations in open-space parking lots could also result in more complete and therefore more efficient parking lot utilization. In addition, more applications can arise, which will use high-precision maps, as the smart and efficient management of transportation systems is one of the main topics of the 21st century.

At present, the data collection for generating HD maps is mainly carried out by the so-called mobile mapping systems that comprise, in most cases, of a vehicle equipped with a broad range of sensors (e.g., radar, lidar, and cameras). This method comes with some drawbacks, for instance, the ground-based systems can cover only a small part of the map due to the sensor line of sight. Sensor drift and global positioning system shadows in urban canyons lower the spatial accuracy, and traffic flow leads to partial occlusions in the recorded data.

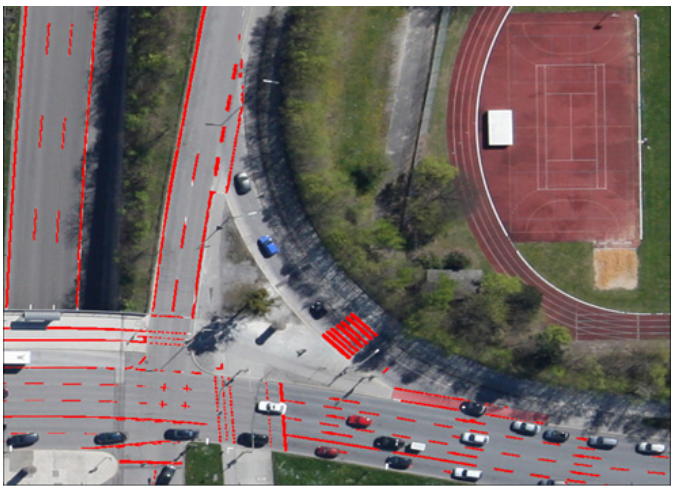


Fig. 1. Sample aerial image patch from AerialLanes18 data set in which lane markings have been annotated. In this task, all classes of lane markings have been considered for pixelwise semantic segmentation.

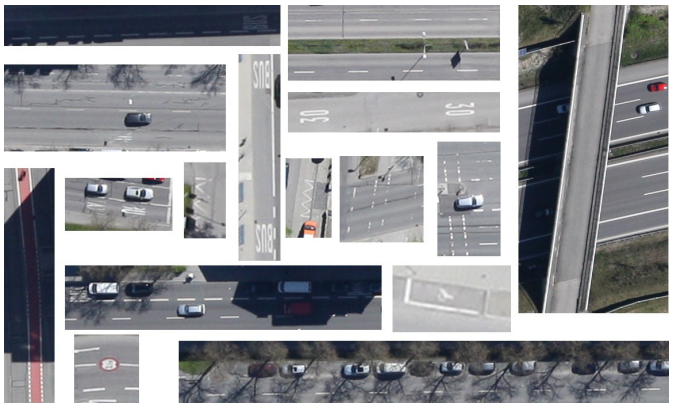


Fig. 2. Challenges in lane-marking segmentation. Light and strong shadow caused by trees and buildings. Examples of rare cases, such as speed limit, the disabled, and bus signs have been indicated. Partial or total occlusion by other objects, such as bridge or tree branches, can be seen.

This issue can be addressed by remote sensing imagery that is intrinsically motivated by the need to capture data from large areas in a short time at a monetary competitive level. More and more airborne and space-borne sensors are recording data in the very-high resolution, e.g., ground sampling distance (GSD) less than 50 cm are in now operational mode. The public sector often offers its data under a free-and-open policy, e.g., aerial imagery of the U.S. Geological Survey in urban regions has ground sampling distance (GSD) less than 30 cm. Data collected by flight campaign with the goal to monitor infrastructure can offer even better GSD. Fig. 1 gives an example of such imagery from the AerialLanes18 data set introduced in this paper, which can be used for the purpose of HD maps creation.

### A. Challenges

Several issues raise the level of difficulty when it comes to image segmentation of aerial imagery for creating HD maps. Some of them are the well-known general problems in the computer vision domain as follows.

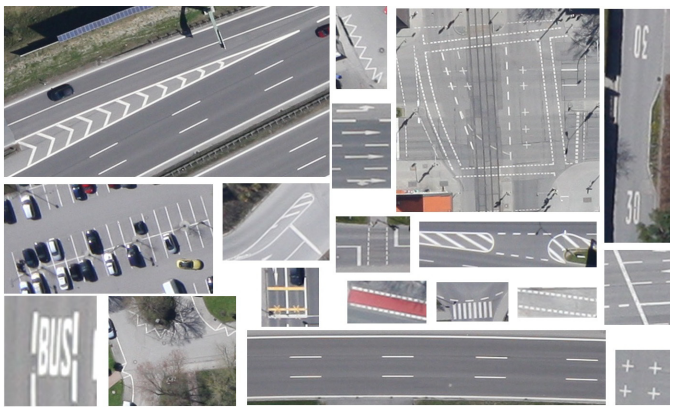


Fig. 3. Different lane-marking classes. Single and double boundary, intersection, boxed junction, turn signs, separator, zigzag, bus and bike sign, speed limit, no-parking zone, and pedestrian crossing.

1) Occlusion (partial or full) changes the appearance of lane markings in the image. Some occlusion cases can be observed in Fig. 2: full occlusion can be caused by other objects such as bridge, tree and so on, while partial occlusion that occurs more often is mostly caused by trees.
2) Shadow creates a different illumination over lane markings causing changes in their appearance. It does not happen often that lane markings are overshadowed, making it a special case. This reason, such as the previous one, could reduce the accuracy of automatic lane-marking algorithms, especially deep learning methods that need a lot of training samples.

Some other challenges are specifically bound to the task of lane-marking segmentation. A short overview is given in the following itemization.

1) *Different Classes:* Generally, lane markings are categorized into different classes, such as single and double boundary, intersection, boxed junction, separator, zigzag, special sign for the disabled, bus and bike sign, speed limit, no-parking zone, pedestrian crossing, and so on. Some of these classes can be seen in Fig. 3.
2) *Small Size:* In airborne imagery, the size of lane markings compared with other objects in the image is, depending on the GSD, quite small. In some cases, a sign of separator could be $5 \times 5$ pixels. This is one of the biggest challenges within the lane-marking mapping task in aerial imagery.
3) *Washed Out Samples:* Not all lane markings are visible in the image; some of them appear washed out partially or completely. This imposes another challenge for the accurate localization of lane markings. On the one hand, in the case of completely washed out lane markings, no visual feature may be captured. Therefore, these cases are ignored. On the other hand, partially occluded objects impose a difficult challenge both in the prediction and data set annotation steps.
4) *Rare Cases:* Lane-marking classes are not equally distributed, as some classes are more frequent than others. Speed limit, bus and bike signs, and parking place for

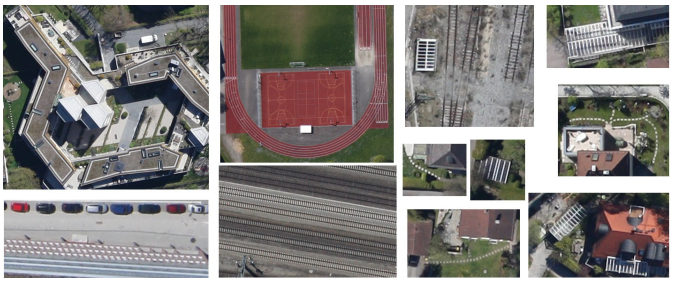


Fig. 4. Complex background. Objects, such as those shown in this figure, share a similar appearance with lane markings. As in some complex background cases, one can name sport field lines, rail ways, roofs of buildings, and so on.

the disabled can be named as rare cases, which can be seen in Fig. 2.

5) The complex background represents an additional hindrance in accurate localization of lane markings. Structures, such as those in Fig. 4, resemble with high-similarity lane markings.

### *B. Related Work*

In spite of the above-mentioned challenges concerning semantic lane-marking segmentation of aerial imagery, another challenge was identified in the early phase of this paper. The usage of aerial images in order to extract valuable data from transportation infrastructure has a rich literature in the remote sensing domain, but as it comes to supervised learning algorithms, we identified the lack of annotated, high-quality data sets. As the lane markings are so small, annotating such objects is difficult and time-consuming. We will, later on, tackle this issue by making our data set easily available.

Concerning aerial imagery, Jin *et al*. [1] first extract the roads. Then, they apply Gabor filters for highlighting the lane markings followed by Otsu's thresholding algorithm for raw binary segmentation. The final result is then given by morphological operators or by using support vector machines [2]. However, by using this approach, some white linear features, such as the ridges of house roofs, may be misclassified if the road extraction is not applied. Also, lines belonging to vehicles or bridges may be misclassified as they are inside the road areas. Furthermore, they did not investigate lane-marking extraction into detail, providing only one resulting image. They also mentioned that objects, such as trees above roads or worn-out/dirty lane markings on the roads, decrease the accuracy of the final results. In order to solve the problem, Jin and Feng [3] propose an approach consisting of three steps to detect lane markings.

1) First, the road centerline is extracted.
2) Then, the road surface is detected.
3) Finally, pavement markings are extracted.

Similar to the previous work, in this paper, also roads are extracted first, and then, lane markings are detected. Even though this method shows a better performance than the previous methods as claimed by the author, it still has the drawback of the previous methods such as not being able to have a good accuracy on lane-marking detection without road extraction. Following this workflow, Jin *et al*. [4] use an unsupervised algorithm to extract the road surface first. Second, Jin *et al*. [4] employed co-occurrence contrast measurements to enhance the lane markings, under the assumption that the contrast between lane marking and road surface is strong and then localized lane markings. Subsequently, morphological closings and openings are applied in order to remove the enhanced edges in the shadow regions. In the last step, the extracted lane-marking features are narrowed by a modified Wang–Zangen algorithm and further fitted to a line by least-square regression. This paper extends lane-marking detection to rural areas. Similar to the previously mentioned works, despite yielding good results in the few provided test images, this paper also suffers from a high rate of false positives in case of not using road extraction step. Further works following this core approach are given by Javanmardi *et al*. [5] and Huang *et al*. [6] who used adaptive threshold in airborne images. Javanmardi *et al*.'s approach [5] contains different steps, such as digital surface model processing, removing vehicles using multiple images, and in the end utilizing a simple adaptive thresholding to extract lane marking. In this method, lane markings are not detected directly as we have done in this paper and the third-party data are used to remove nonlane-marking objects.

Hinz and Baumgartner [7] propose a method to extract lane markings by multiview imagery and context cues and also used the extracted thin lines as a hint for the presence of a road. This method yields very good results. However, this method works only when multiple images have been captured with different views from a place of interest. This method is also similar to previously mentioned works in using the road mask, and therefore, it suffers from low accuracy in case of not applying the road extraction step. Máttyus *et al*. [8] and Gellert *et al*. [9] proposed a method based on Markov random fields and a combined parsing of both ground and aerial images to generate detailed maps. These road models could be used for masking images in order to localize lane markings, but it cannot be used directly for lane-marking localization and only helps to find roads and the boundaries of each line in the roads.

Tournaire *et al.* [10] extract the dashed line and zebra crossing with the use of information obtained by the reconstruction process from the extracted primitives of the image. In contrast to this paper, they only considered rectangle line markings and studied their geometric properties to be able to extract them. Furthermore, they did not use a learning feature approach to detect lane markings as we have done in this paper. More complete overviews about the extraction of roads and road features from airborne images can be found in [11] and [12].

As discussed, no previous work has tried to learn the features of the lane marking through an end-to-end feature learning mechanism, e.g., deep learning methods, to the best of our knowledge. Unlike in remote sensing community, researchers in computer vision community have already applied deep learning methods to extract road infrastructure features in *in situ* images.

Deep learning methods, currently widely used in computer vision, try to learn features rather than using engineered features. During the last few years, deep learning methods

have shown an impressive performance in a variety of computer vision tasks, such as object recognition [13]–[15], detection [16]–[19], and semantic segmentation [20]–[23]. Convolutional neural networks (CNNs), as one of the widely used deep learning methods, have been proven to be very successful for object recognition in images [13]–[15].

However, pixelwise semantic segmentation is a more challenging problem, as each pixel should be classified. Kim and Park [24] propose a sequential transfer learning method based on fully convolutional neural networks (FCNNs) by segmenting the road in the first step and then lane-marking segmentation on the road-masked image. This method is similar to the methodology used in current lane-marking detection algorithms in remote sensing. The main difference is now using FCNNs to extract roads first rather than using nondeep-learning-based methods.

Gurghian *et al*. [25] propose a CNN classification method to localize lane markings on both sides of a vehicle. However, this method is not applicable to remote sensing applications as we are interested to detect lane marking in all regions in the images. Lee *et al*. [26] propose a multitask CNN to localize and classify lane markings in the daytime with different weather conditions as well as during nighttime. This is a very interesting work where the author has developed a method to detect lane markings in different weather conditions. However, this method and other FCNN-based methods in lane-marking detection have been developed for ground imagery processing. Lane markings of small size in image data have not been the focus of most works in this context. In imagery from cars or poles (ground imagery), they are big enough and therefore do not introduce a significant challenge. Having said that in remote sensing imagery, lane markings can be as small as $3 \times 3$ pixels, which are much more difficult to detect.

In order to facilitate the application of supervised learning methods, Caltech Lane [27] and tuSimple [28] data sets were created for lane-marking segmentation, while large-scale data sets for semantic understanding of roads containing a diverse range of classes, including lane markings, have been defined in [29] and [30]. The aforementioned data sets are in ground imagery, and to the best of our knowledge, there is no public data set available for research on lane-marking localization in remote sensing data.

In this paper, we have created the first high-quality annotated data set for lane-marking semantic segmentation in remote sensing imagery specifically in airborne images. We use FCNNs as the baselines of our method. Therefore, this paper is, to our knowledge, the first time using FCNNs to segment lane marking in remote sensing data in contrast to previous methods that mostly detect road first as a hint and second apply edge detection-based methods to segment lane markings. This is one of the main differences of this paper compared with previous works on this task. Unlike the works are done in ground imagery, in this paper, we focus on small-size lane markings by inserting discrete wavelet transforms (DWTs) of input images in different steps into FCNNs to preserve high-frequency information, including lane markings. Wavelet transforms have been widely used both in ground [31] and remote sensing imagery [32]. Recently, Fujieda *et al*. [33] also used DWT combined with CNNs for texture classification. They used CNNs for classification, while in this paper, the focus is on the semantic segmentation task that is a different task from classification. They inserted all DWT decompositions with CNN only in two steps and in the middle of the convolutional layers and did not investigate which insertion place for DWT yields the best results, while in this paper, we use three decompositions and also investigate where is the best place to insert DWT to yield the best results. In their work, DWT decompositions were inserted into CNNs as an input, while in this paper, we still give the RGB image as an input. More importantly, the effect of DWT was not investigated from the point of preserving high-frequency data such as very small objects for semantic segmentation. Moreover, we deploy a weighted loss function as well as symmetric FCNN. Although FCNNs introduced by Long *et al*. [20] are among the first deep learning methods for the semantic segmentation task, its accuracies are still comparable with the state of the art, such as DeepLabv3 [34], DeepLabv3+ [35], PSPNet [22], and ICNet [36], and others with deep backbone networks, such as ResNet [14], ResNext [37], Xception [38], and DenseNet [39]. We choose the FCNN network proposed by Long *et al*. [20] with VGG16 backbone as a baseline of our method due to its simplicity and familiarity of the community with its architecture and yet its accuracy is comparable with the-state-of-the-art methods.

### C. Our Contribution

In this paper, we focus on lane-marking pixelwise semantic segmentation using aerial images. In high-resolution aerial images, the lane markings are easy to identify. Our proposal is based on combining FCNNs with DWT for lane-marking pixelwise semantic segmentation in airborne images. The motivation of using FCNNs as a deep learning method for semantic segmentation is its higher performance compared with nondeep-learning methods.

Unlike traditional methods in which feature extraction and classification steps are performed separately, in FCNNs, features are learned during an end-to-end training and there is no separation between feature extraction and feature classification. FCNNs have been proposed first by Long *et al*. [20] for semantic segmentation in *in situ* imagery with extra upsampling layers (deconvolutional layers). The authors of FCNNs propose multiple pooling layers to be fused with upsampling layers (skip layers) to further refine segmentation boundaries. The authors call their network and its variants FCN32s, FCN16s, and FCN8s. We consider FCN32s as the baseline of this paper.

In order to enhance current network performance, we combine different input images with the FCNN network. The motivation of using DWT is to provide the network with different representations of input objects in different scales as well as full-spectral analysis. DWTs can represent the input image at different scales. While CNNs process the image in the spatial domain and partially in the spectral domain, DWT allows analyzing the images in the full-spectral domain. Therefore, the properties of these algorithms are different.

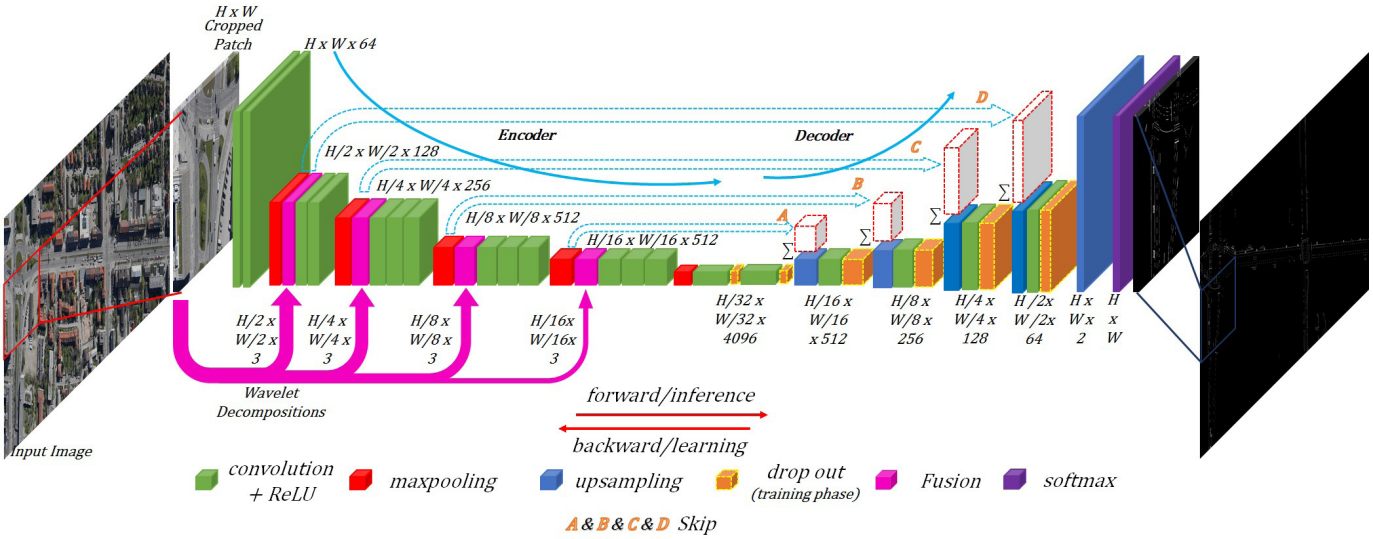



Fig. 5. Aerial LaneNet. Overview of lane-marking segmentation approach using the wavelet-enhanced symmetric cost-sensitive FCNNs. The input image is a high-resolution aerial image. It is cropped first and segmented using the Aerial LaneNet network. In the end, segmented patches are stitched together. $H$ and $W$ represent height and width and the third number is the number of feature maps.

Integrating DWT will enable the network to access the intensity frequency information that is lost in the convolution and average pooling layers, carrying out limited spectral analysis. The intensity frequency information lays in the frequency domain for the pixel intensities variation and not in the different image bands, e.g., in hyperspectral images. Wavelet transform has been investigated for a long time for frequency analysis and also image compression.

In this paper, we have carried out experiments with different combinations of DWT decompositions to be used as an input with a modified version of FCN32s, which we call "Symmetric FCNN." The final result is a pixelwise semantic segmentation of lane marking. Due to the heavily unbalanced task in terms of a number of lane-marking pixels compared with background ones, we have applied a cost-sensitive loss function to impose higher loss for the wrong classification of lane markings as a minor class than loss for the wrong classification of background. As mentioned earlier, we introduce the first high-quality pixelwise annotated data set for lane-marking segmentation and detection in aerial imagery, which shall encourage future works in this area.

The following sections are organized as follows. Section II represents the methodology to enhance FCNN with different DWT decompositions, the cost-sensitive loss function used during the training phase, and the symmetric FCNNs architecture. In Section III, we introduce the data set and its features and properties and report different experiments. In Section IV, the results of the experiments are given and evaluated. In Section V, a conclusion is drawn.

## II. Arial LaneNet: Wavelet-Enhanced Cost-Sensitive Symmetric Fully Convolutional Neural Network

In this paper, we propose a cost-sensitive symmetric FCNN enhanced by DWT, which we call Aerial LaneNet. The overall workflow of our method is illustrated in Fig. 5. Due to the high resolution of aerial images and hardware memory constraint, the original images are chopped into small patches using a sliding window [40]. Then, each patch is processed by Aerial LaneNet in order to predict a semantic segmentation of the input patch.

The output is a binary image that denotes which pixel belongs to lane markings and which one to the background. In the end, patches are stitched together to create the final output with the same resolution as the input image. In the following, we explain our proposed methods in detail.

CNNs are a combination of different layers, such as convolution, pooling, activation function, dropout, and fully connected layers. Input data are convolved with a linear convolution filter in convolution layers

$$(h_k)_{ij} = (W_k * X)_{ij} + b_k \tag{1}$$

where $k = 1, \ldots, K$ is the $k$th feature map in the convolution layer and $(i, j)$ is the index of a neuron in it. $X$ stands for the input data and $W_k$ and $b_k$ are the weights (trainable parameters) of the network and the biases (trainable parameters), respectively.

The output of each neuron in the $k$th feature map has been represented by $(h_k)_{ij}$ at position $(i, j)$. The 2-D convolution between input data and filter mask in the spatial domain is represented by "$*$," which partially includes spectral analysis at low frequencies, while the remaining spectral information is lost.

Considering Fig. 6, parts shown in red in the DWT algorithm can be considered as a convolution function in the traditional CNNs. On the other hand, a wavelet transform is able to capture the full-spectral information of the input in the frequency domain.

Moreover, wavelets can extract multiresolution spectral information from input data at different decomposition levels,

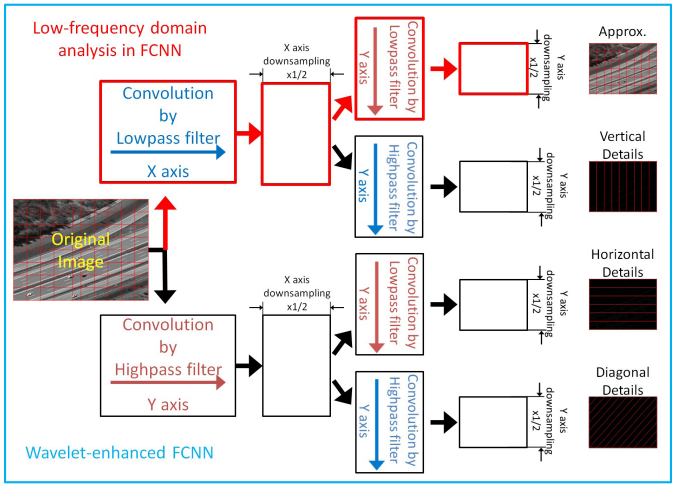



Fig. 6. First-level DWT decomposition workflow. The input gray-scale image is processed by low-pass and high-pass filter in different directions. The output is with a half size of the original image. Afterward, the same operation is applied to each part, resulting in four decomposition parts of the input image in the first-level DWT. In conventional FCNNs, only the low-frequency analysis is carried out shown in red, while DWT offers a full-spectral analysis shown in blue.

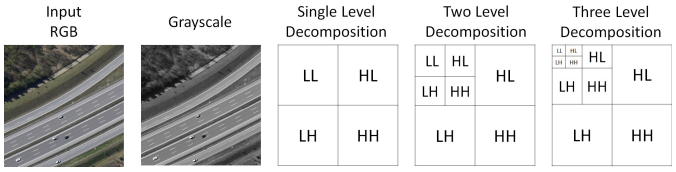



Fig. 7. Different DWT decompositions. The input RGB image is converted to gray scale first. Then, first DWT decomposition is computed followed by next levels. High-pass and low-pass filters are represented by "*H*" and "*L*," respectively. LL stands for two-step low-pass filtering, where HL, LH, and HH contain the horizontal, vertical, and diagonal details, respectively.

as shown in Fig. 7. A multiresolution analysis of the input data would represent the input in different scales similar to a pooling operation. Each subsampling step in wavelet transform can be considered as a different pooling operation.

Therefore, pooling layers could also be replaced by wavelet transforms. Instead of doing so, we merge (fuse) wavelet information of the input with the traditional FCNNs together with pooling layers, which can be done in different ways. In order to add the wavelet decomposition to the network, one can compute wavelet transforms for each image and apply the output to FCNNs. However, in this case, multiscale information of the data is lost. Therefore, the network is not able to learn the lane-marking features at different resolutions. This will lead to a nonscale-invariant method. To address this problem, multiscale input processing is needed.

Each level of wavelet decomposition analyzes the data at different resolutions. Therefore, by combining different decomposition levels of wavelet transforms with FCNNs, low- and high-frequency domain analyses as well as different resolution analyses are achieved.

After applying a wavelet transform on the input image, lane-marking boundaries appear as high-frequency objects in vertical, horizontal, and partially in diagonal details in the wavelet coefficients. Different parts from the first to the third level of the DWT are illustrated in Fig. 7.

### A. Discrete Wavelet Transform (Background)

DWT of a signal $x$ is computed by applying a series of filters and subsampling in subsequent levels [32]. For instance, in the first level of DWT, a low-pass and a high-pass filter are applied simultaneously with impulse responses of $g$ and $h$ resulting in two convolutions of

$$y_{\text{lowpass}}[n] = (x * g)[n] = \sum_{x=-\infty}^{+\infty} x[k]g[n-k]$$
$$y_{\text{highpass}}[n] = (x * h)[n] = \sum_{x=-\infty}^{+\infty} x[k]h[n-k] \quad (2)$$

and the resulting signals are subsampled by a factor of 2, i.e.,

$$y_{\text{lowpass}} = (x * g) \downarrow 2$$
$$y_{\text{highpass}} = (x * h) \downarrow 2. \quad (3)$$

In order to further increase the approximation coefficients and the frequency resolution resulting from low- and high-pass filters and downsamplings, this decomposition is repeated. This results in a tree representation of each decomposition level known as a filter bank, which is illustrated for a two-level decomposition in Fig. 6. We can consider the implementation of wavelet filters as the wavelet coefficients calculation of a discrete set of lower level wavelets for a mother wavelet function $\Psi(x)$. By applying DWT, a discrete function $f(x)$ is converted into a signal of two variables [32]: scale and translation, which can be described as

$$\Psi_{j,k}(x) := \frac{1}{2^{j/2}}\Psi\left(\frac{x-k2^j}{2^j}\right) \quad (4)$$
$$\Phi_{j,k}(x) := \frac{1}{2^{j/2}}\Phi\left(\frac{x-k2^j}{2^j}\right) \quad (5)$$
$$\Psi(x) := \begin{cases} 1, & \text{for } 0 \le x \le 1/2 \\ -1, & \text{for } 1/2 < x \le 1 \\ 0, & \text{otherwise} \end{cases} \quad (6)$$
$$\Phi(x) := \begin{cases} 1, & \text{for } 0 \le x \le 1 \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

in which $\Phi_{j,k}(x)$ is the scaling function for which the box function $\Phi$ has been chosen. $\Psi_{j,k}(x)$ and $\Phi_{j,k}(x)$ have ranges of $[-(1/2^{j/2}), (1/2^{j/2})]$ and $[0, (1/2^{j/2})]$ accordingly with width $2^j$ that starts at $k2^j$. The scale level is represented by $j$ and the shift by $k$. $\Psi_{j,k}(x)$ are scaled and shifted versions of the continuous mother wavelet $\Psi(x)$. In the discrete domain, for a signal of length $N = 2^n$, one considers the $N$ functions $\Phi_{n,0}, \Psi_{n,0} \ldots \Psi_{1,2^{n-1}-1}$. In this paper, we consider the Haar wavelet transform as the first order of the Daubechies wavelet

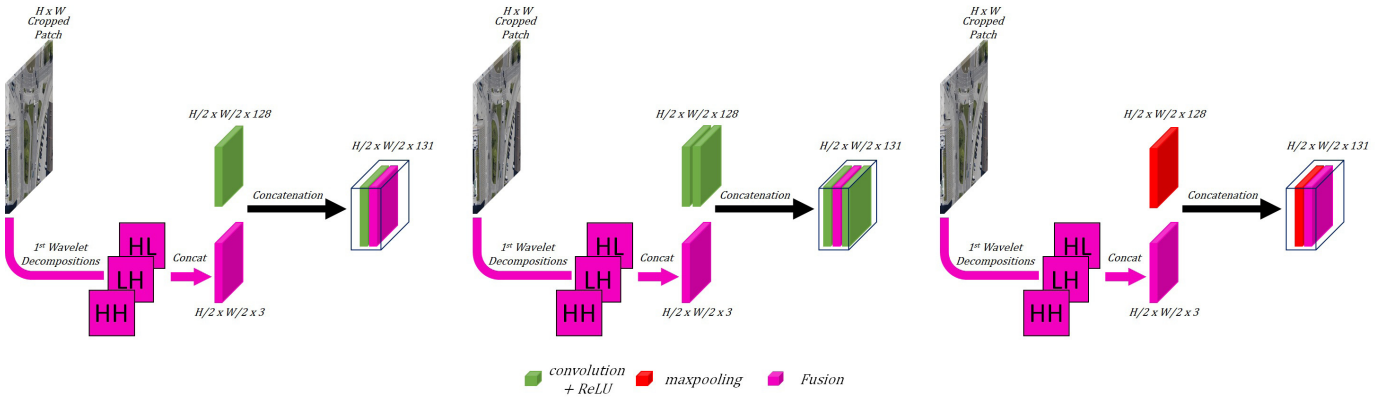



Fig. 8. Different first-level DWT fusions with symmetric FCNNs. There are three fusion variants. (Left) Before pooling layer. (Middle) After convolution layer. (Right) After pooling layer.

family [41] with $n = 2$ and use the basis vectors

$$\begin{aligned}\Phi_{2,0} &= \frac{1}{2}(1,1,1,1)^T \\ \Phi_{2,1} &= \frac{1}{2}(1,1,-1,-1)^T \\ \Phi_{1,0} &= \frac{1}{2}(1,-1,0,0)^T \\ \Phi_{1,1} &= \frac{1}{2}(0,0,1,-1)^T \end{aligned} \tag{8}$$

that yield the coefficients

$$\begin{aligned} c_{j,k} &:= f^T \Phi_{j,k} \\ d_{j,k} &:= f^T \Psi_{j,k} \end{aligned} \tag{9}$$

in which $c_{j,k}$ are coefficients of the scaling vector $\Phi_{j,k}$, and for coarse decomposition, these are low-pass filter coefficients. Similarly, $d_{j,k}$ are the coefficients of the wavelet vector $\Psi_{j,k}$ for detailed decompositions, which are high-pass filter coefficients. In 2-D DWT, it starts first with calculating the wavelet decomposition on a single level in the $x$-direction than in the $y$-direction. Afterward, the next decomposition is performed only in the quadrant part that contains the low-frequency parts (scaling coefficients) for both directions. The decomposition levels proceed until a single pixel is reached.

In order to compress the images as wavelet transform injections, the orthonormal Daubechies wavelet family [41] is selected for their proven success in decomposing images and identifying borders. The Daubechies wavelet family is written as dbN, where $N$ is the order and db is the abbreviation for the Daubechies wavelet family. The db1 wavelet is the same as the Haar wavelet and the first order of Daubechies family with lower computation cost and fewer wavelet filter bank coefficients. The continuous wavelet transform has been presented in (4).

As shown in Fig. 5, DWT decompositions are injected as shown by the paths in pink. Given that the input data are $H$ (Height) and $W$ (Weight) pixels after having changed to gray-scale image shown in Fig. 7, using four levels of the wavelet transform on the input image results in the outputs with $H/2 \times W/2$, $H/4 \times W/4$, $H/8 \times W/8$, and $H/16 \times W/16$ sizes. The input image is first converted to gray scale before DWT computation. In contrast to usual cases in which more data result in a better performance, our preliminary results show that using an RGB input image results in 1.78% intersection over union (IoU) performance decrease. To further investigate this issue, we considered other color spaces including hue, saturation, and value and observed the same effect which we conjecture it could be due to insertion of redundant input data. It is worth mentioning that the parameters of DWT is fixed and are not updated during the training phase. The first-level DWT has an input size of $H \times W$ and four outputs (approximate, horizontal, vertical, and diagonal) with half size capturing different details in the image such as shown in Fig. 6.

The fusion of the first-level wavelet transform has to be done after the first pooling. The reason is that the input size of the image is $H \times W$, while the size of the first-level wavelet decomposition is $H/2 \times W/2$. Hence, due to incompatible size resolution, the first fusion layer is carried out after the first pooling operation.

Inserting the first-level DWT decompositions with a half size of the input image as an input to the network results in losing spatial and spectral information of the original input. Therefore, this scenario is not efficient.

There are different ways of wavelet transform fusion with the FCNN network, as shown in Fig. 8. As mentioned, the wavelet decompositions have to be placed after the pooling layer. We have considered all three illustrated cases to combine the first wavelet decomposition level to the network. The same goes for other DWT levels. A typical cross-entropy loss function in semantic segmentation treats pixels belonging to different classes equally. For a binary classification problem, this can be represented as

$$\begin{aligned} L(\mathbf{W}) = -\frac{1}{N}\sum_{n=1}^{N} & y_n \log \hat{y}(x_n, \mathbf{W}) \\ & + (1 - y_n)(1 - \log \hat{y}(x_n, \mathbf{W})) \end{aligned} \tag{10}$$

where $x_n \in [0, 255]$ is the input pixel value, $y_n \in \{0, 1\}$ is the ground-truth label, $\hat{y}_n \in [0, 1]$ is the prediction probability, $\mathbf{W}$ is the weight matrix of the network, and $L$ denotes the loss function.

In order to classify each pixel, the softmax function is widely used in multiclass classification tasks in FCNNs.

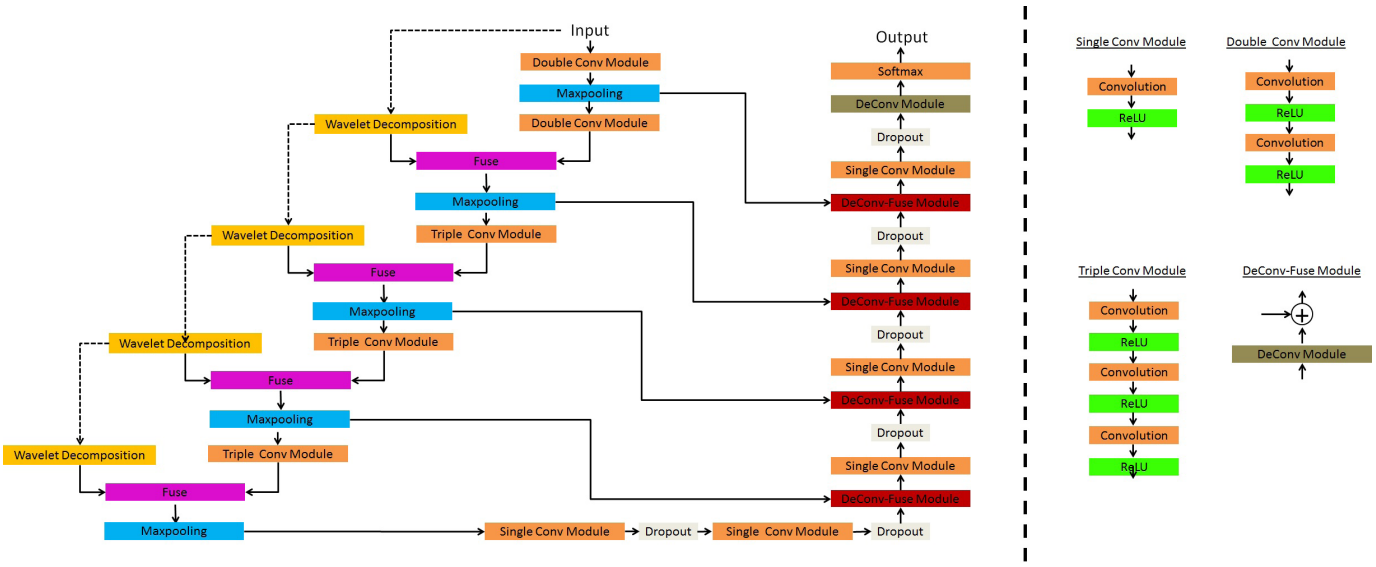



Fig. 9. Aerial LaneNet architecture break down.

The vector of real values between $[0, 1]$ generated by this function denotes a categorical probability distribution.

The softmax function can be expressed as $\hat{y}_j = \text{softmax}(X, W_j) = ((e^{X^T W_j})/(\sum_{k=1}^{K} e^{X^T W_j}))$, in which $W_j$ and $X$ denote the weights of the network (including bias values) and the input data, respectively. The well-known loss layer using the softmax function for multiclass classification is cross-entropy loss.

However, for lane-marking segmentation, the majority of pixels belong to the nonlane-marking class. This makes the problem highly unbalanced. Therefore, we modify the typical cross-entropy loss function by imposing a higher cost on the wrong classification of a lane-marking pixel compared with a background pixel. The defined loss function is

$$L(\mathbf{W}) = -\frac{1}{N}\left(\lambda_{\text{lane}} \sum_{n=1}^{N} y_n \log \hat{y}(x_n, \mathbf{W}) + \sum_{n=1}^{N}(1 - y_n) \times \log\left(1 - \hat{y}(x_n, \mathbf{W})\right)\right) \quad (11)$$

which is cost-sensitive, as it penalizes different class pixels differently. This is done by introducing parameter $\lambda_{\text{lane}}$ in the cross-entropy loss function. This weighted loss function can be easily extended to a multiclass segmentation scenario by inserting a function $\mathbb{1}_{\text{cls}}(x_n)$ which is equal to one if $x_n$ belongs to class $cls$ and zero if it does not. To leverage the capacity of CNNs to perform semantic segmentation, the networks can be modified by replacing fully connected layers with convolution layers that allow CNNs to be applied to images with variable sizes.

This approach will not lead to semantic segmentation with the same resolution as the input image. Therefore, extra upsampling layers (bilinear interpolation) are applied in the baseline network. Bilinear interpolation is differentiable, which makes applying backpropagation during training feasible.

In order to grasp varied visual input information yet keeping input feature map dimensions, the upsampling layer is applied after the last convolution layer to upsample the extracted features to the input dimension size. This can be considered as the encoding of the input data to the first upsampling layer and decoding by upsampling layers, as shown in Fig. 5.

By modification of FCNNs to be more robust to overfitting, we design a symmetric FCNN network. In this methodology, we add convolution and dropout layers after upsampling layers in the baseline network of FCN32s. We do the same for FCN16s and FCN8s network architectures. We also add one additional upsampling layer, which can be seen as a new FCN4s network.

Instead of using average pooling layers, we use max-pooling layers. In FCN4s, we also apply the fusion technique used in the baseline paper, which is a summation of the corresponding pooling layers with the output of the upsampling layers. The motivation to add more convolution layers comes from [13], [14], and [42] where it has been shown that depth has a key role in high-level feature extraction.

Aerial LaneNet is not limited to a fixed input size, i.e., there is no need to resize input images. The only preprocessing step is the subtraction of image mean. Due to the heavily unbalanced data sets for lane marking and the scarcity of such data sets, more dropout layers have been added to the network to prevent overfitting. The deep neural networks are prone to overfitting according to the noise present in the training set samples if that is small.

The inserted layers have been denoted in red in Table I. In Fig. 9, the Aerial LaneNet network architecture is reported in detail. In order to investigate the architecture of the network and its properties such as input and output size, feature map dimension, receptive field, and so on, Table I has been prepared.

## III. Experiments

In this section, we introduce the data set used in the experiments. Then, we explain the experiments and provide the quantitative and qualitative results along with corresponding discussions.

TABLE I

SYMMETRIC FCNN INPUT AND OUTPUT SIZES FOR EACH LAYER AS WELL AS FILTER MAPS AND RECEPTIVE FIELDS. ADDED LAYERS IN SYMMETRIC FCNN TO FCN8S HAVE BEEN SPECIFIED WITH RED COLORS

| Layer | Input | Output | Features | Receptive Field |
|---|---|---|---|---|
| conv1-1 | $960 \times 960 \times 3$ | $960 \times 960 \times 64$ | 64 | $3 \times 3$ |
| conv1-2 | $960 \times 960 \times 64$ | $960 \times 96 \times 64$ | 64 | $5 \times 5$ |
| maxpooling-1/conv2-1 | $960 \times 960 \times 64$ | $480 \times 480 \times 128$ | 128 | $11 \times 11$ |
| conv2-2/1st level Wavelet-fusion | $480 \times 480 \times 128$ | $480 \times 480 \times 131$ | 131 | $13 \times 13$ |
| maxpooling-2/conv3-1 | $480 \times 480 \times 131$ | $240 \times 240 \times 256$ | 256 | $17 \times 17$ |
| conv3-2 | $240 \times 240 \times 256$ | $240 \times 240 \times 256$ | 256 | $19 \times 19$ |
| conv3-3/2nd level Wavelet-fusion | $240 \times 240 \times 256$ | $240 \times 240 \times 259$ | 259 | $21 \times 21$ |
| maxpooling-3/conv4-1 | $240 \times 240 \times 256$ | $120 \times 120 \times 512$ | 512 | $25 \times 25$ |
| conv4-2 | $120 \times 120 \times 512$ | $120 \times 120 \times 512$ | 512 | $27 \times 27$ |
| conv4-3/3rd level Wavelet-fusion | $120 \times 120 \times 512$ | $120 \times 120 \times 515$ | 515 | $29 \times 29$ |
| maxpooling-4/conv5-1 | $120 \times 120 \times 515$ | $60 \times 60 \times 512$ | 512 | $33 \times 33$ |
| conv5-2 | $60 \times 60 \times 512$ | $60 \times 60 \times 512$ | 512 | $35 \times 35$ |
| conv5-3/4th level Wavelet-fusion | $60 \times 60 \times 512$ | $60 \times 60 \times 515$ | 515 | $37 \times 37$ |
| maxpooling-5/conv6-1 | $60 \times 60 \times 515$ | $30 \times 30 \times 4096$ | 4096 | $41 \times 41$ |
| dropout-1 | - | - | - | - |
| conv6-2 | $30 \times 30 \times 4096$ | $30 \times 30 \times 4096$ | 4096 | $43 \times 43$ |
| dropout-2 | - | - | - | - |
| deconv-1/maxpooling-1-fusion | $30 \times 30 \times 4096$ | $60 \times 60 \times 512$ | 512 | $43 \times 43$ |
| conv7 | $60 \times 60 \times 512$ | $60 \times 60 \times 512$ | 512 | $43 \times 43$ |
| dropout-3 | - | - | - | - |
| deconv-2/maxpooling-2-fusion | $60 \times 60 \times 512$ | $120 \times 120 \times 256$ | 256 | $43 \times 43$ |
| conv8 | $120 \times 120 \times 256$ | $120 \times 120 \times 256$ | 256 | $43 \times 43$ |
| dropout-4 | - | - | - | - |
| deconv-3/maxspooling-3-fusion | $120 \times 120 \times 256$ | $240 \times 240 \times 128$ | 128 | $43 \times 43$ |
| conv9 | $240 \times 240 \times 128$ | $240 \times 240 \times 128$ | 128 | $43 \times 43$ |
| dropout-5 | - | - | - | - |
| deconv-4/maxpooling-4-fusion | $240 \times 240 \times 128$ | $480 \times 480 \times 64$ | 64 | $43 \times 43$ |
| conv10 | $480 \times 480 \times 64$ | $480 \times 480 \times 64$ | 64 | $43 \times 43$ |
| dropout-6 | - | - | - | - |
| deconv-5 | $480 \times 480 \times 64$ | $960 \times 960 \times 2$ | 2 | $43 \times 43$ |

### *A. AerialLanes18 Data Set*

The experiments were conducted using images acquired by the German Aerospace Center (DLR) within a flight campaign in the framework of the VABENE++ project. The campaign was carried out over the greater area of the city of Munich on April 26, 2012.

The 3K camera system [43] consisting of three Canon Eos 1Ds Mark III cameras was used for recording the raw data, where two cameras are mounted side looking and one is mounted nadir looking on a flexible platform.

The 3K system is a low-cost camera system used for various remote sensing applications, such as real-time mapping [44], disaster monitoring [45], traffic monitoring [46], and detection of high-density crowds [47].

In total, 20 representative RGB images of size $5616 \times 3744$ pixels have been chosen. The flight height of about 1000 m above ground led to a GSD of approximately 13 cm.

The images depict urban and partly rural areas with highways and first-/second-order roads. Complex traffic situations, such as crossings and congestions, are included. The images served as a starting point for works in the domain of vehicle detection by Liu and Mattyus [46].

### *B. Annotation of AerialLanes18*

The ground truth has been annotated by human experts who marked all kinds of lane markings over roads and highways, such as separate line, continuous line, turn sign, speed limit sign, and even bus and disabled people parking place signs. The annotation was carried out manually by using an in-house annotation software. During annotation, we ignored washed out lane markings. Fig. 10 shows some patches of the mentioned data set. Fig. 11 show the large training images with the overlaid lane-marking annotations.

### *C. Implementation Details*

As the data set does not consist of many images, most likely training a deep neural network on such a small data set from scratch with randomly initialized parameters will lead to overfitting. On the other hand, as annotating small lane-marking objects is difficult and time-consuming, only images of the mentioned data set have been annotated. To address this problem, networks that have already been trained using large data sets, such as ImageNet [48], are used as initialization of parameters in order to transfer the learned information to a new task. This technique is known as "Transfer Learning." Using this technique, we can initialize the weights more efficiently.

Therefore, it can be assumed that the network is already close to one of the optimal solutions and needs far less training data to converge, and by retraining the network known as "fine-tuning" technique, the problem of overfitting can decrease significantly. In our experiments with wavelet transform fusion, we use FCN32s [20] as the baseline. VGG16 proposed by Simonyan and Zisserman [13] is the backbone

Fig. 10. Sample training patches from the AerialLanes18 data set taken by aerial imagery over Munich, Germany. The original image patch is shown with its corresponding annotation. GSD is 13 cm.

Fig. 11. Sample large training image from the AerialLanes18 data set. The original image patch is shown with its corresponding annotation.

main network. However, AlexNet [42], GoogleNet [49], and ResNet-101 [14] are also considered.

We use the patches of $1024 \times 1024$ pixels as an input to the network. We employ the 800 pixels cropping step in the horizontal and vertical directions in the training phase and 1000 pixels in the test phase. For the training step, random flipping patches are applied for data augmentation. We consider one random image as a validation set that consists of 24 patches. In the test set, the number of test patches is 240. Networks are trained on the training set to find the best hyperparameters, and then, both the training and the validation set are used for the final training.

It should be mentioned that in the following experiments, no extra information such as road segmentation or third-party data such as OpenStreetMap [50] has been used.

Aerial LaneNet is trained end-to-end. The optimization problem of finding the minimum value in the loss function is solved by Adam optimizer [51] and backpropagation [52] process. The learning rate of 0.0001 with a batch size of 1 is used. We have trained the final network for about 10 epochs on one Nvidia Titan X Pascal GPU using the Tensorflow [53] framework.

## IV. Results and Evaluation

In our experiments, we compare the final output of the system for each image (not patch) with the corresponding ground truth. Therefore, in lane-marking segmentation, the goal is to classify each pixel as lane-marking class (foreground) or nonlane marking (background). The more pixels are classified correctly, the more accurate the system is. Concerning the evaluation criteria, we use the metrics used by Long *et al.* [20], which are widely used in semantic segmentation tasks. In these metrics, $n_{ij}$ is the pixel number belonging to class $i$, which

TABLE II

EVALUATION OF LANE-MARKING SEGMENTATION USING DIFFERENT BACKBONE NETWORKS FOR SEGMENTATION WITH ONE UPSAMPLING LAYER. WITH VGG16 NETWORK, THIS IS EQUIVALENT WITH FCN32S. IN FINE-TUNING, THE PARAMETERS ARE INITIALIZED BY THE IMAGENET PRETRAINED MODEL RATHER THAN RANDOM INITIALIZATION. IN THIS CASE, ALL OF THE LAYERS ARE RETRAINED. MEAN IOU NUMBERS IN [%]. HIGHER VALUE IS BETTER. MAX STRIDE IS 32 PIXELS

| Network | weighted loss | fine tuned | data augmentation | mean IoU | forward time | conv. layers | param. |
|---|---|---|---|---|---|---|---|
| FCN-AlexNet [20] | - | - | - | 51.08 | 80ms | 8 | 57M |
| FCN-AlexNet | - | - | ✓ | 52.92 | 80ms | 8 | 57M |
| FCN-AlexNet | - | ✓ | ✓ | 55.23 | 80ms | 8 | 57M |
| FCN-AlexNet | ✓ | ✓ | ✓ | 59.06 | 80ms | 8 | 57M |
| FCN-VGG16 [20] | ✓ | ✓ | ✓ | **61.56** | 300ms | 16 | 134M |
| FCN-GoogLeNet [20] | ✓ | ✓ | ✓ | 61.49 | 100ms | 22 | 6M |

has been predicted as class $j$, and $n_{\text{cl}}$ stands for the number of classes with $t_i = \sum_j n_{ij}$ representing the total number of pixels belonging to class $i$. IoU means intersection over union, i.e., it is proportional to the intersection between predictions and ground truth.

We also use the dice similarity coefficient due to the heavy unbalance in the data set. The number of pixels belonging to each class does not have an effect on these two criteria. $P$ and $T$ represent the prediction and ground truth, respectively. The criteria are derived as follows.

1) *Pixel Accuracy*:

$$\frac{\sum_i n_{i,i}}{\sum_i t_i}. \tag{12}$$

2) *Mean Accuracy*:

$$\frac{1}{n_{cl}} \sum_i \frac{n_{i,i}}{t_i}. \tag{13}$$

3) *Mean IoU*:

$$\frac{1}{n_{cl}} \sum_i \frac{n_{i,i}}{t_i + \sum_j n_{j,i} - n_{i,i}}. \tag{14}$$

4) *Frequency Weighted IoU*:

$$\left( \sum_k t_k \right)^{-1} \sum_i \frac{t_i n_{i,i}}{t_i + \sum_j n_{j,i} - n_{i,i}}. \tag{15}$$

5) *Dice Similarity Coefficient*:

$$\frac{2 \mid P \cap T \mid}{\mid P \mid + \mid T \mid} \tag{16}$$

and recall and precision are calculated using the criteria

$$\begin{aligned} \text{Recall} &:= \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \\ \text{Precision} &:= \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}. \end{aligned} \tag{17}$$

The baseline network of FCN32s with AlexNet as a backbone network is trained from scratch, and due to the small and highly unbalanced data set, it classifies lane-marking pixels as background in most areas, with only 51.0% mean IoU accuracy.

Employing weighted loss has increased the performance by almost 2% by penalizing the wrong classification of lane-marking pixels more than the wrong classification of background pixels, alleviating to some extend the challenge posed by an unbalanced data set.

Before applying the customized loss function, fine-tuning using a pretrained model trained on ImageNet [48] and data augmentation are applied due to the small training data set available.

*1) Different Base Network Investigation:* Results in Table II show the performance of Aerial LaneNet in lane-marking segmentation with different network architectures. VGG16 outperforms AlexNet as the shallower network and slightly GoogleNet. The high pixel accuracy of this system should be investigated, as most of pixels belong to the background class rather than lane markings. This phenomenon has two main reasons: first the network is overfitting to the background class due to the small-size data set and second due to the heavily unbalanced data set. As expected, due to the highly unbalanced data set, pixel accuracy and frequency weighted IoU are larger than 99%. These parameters, as mentioned earlier, are not suitable to evaluate the performance of a network using a highly unbalanced task. That is why mean IoU and Dice are more reliable criteria to evaluate an algorithm in such cases.

*2) Effect of $\lambda$:* The value of $\lambda_{\text{lane}}$, which is a hyperparameter, should be tuned. There is no automatic approach to find the best value for this parameter. One approach is considering the default value of $\lambda_{\text{lane}} = 389$ as the ratio of background to lane-marking pixels in the training set. Another method is the grid search that can be applied to refine the default value. We considered the pixel ratio in the test set as well as other setups ranging from 1 to 1000. With this approach, we noticed that the pixel ratio is not the best value to get the best results (Fig. 12). Considering Table III, the best value is achieved with 400 that is higher than the default one and lower than 418 as the ratio in the trainval set. Performance degrades using 308 as the ratio in the test set. This shows that the network has learned this hyperparameter based on the training set. In this case, more research can be devoted to find the best value of $\lambda_{\text{lane}}$ automatically.

*3) Importance of Symmetric FCNN:* As mentioned in Section II, in order to extract higher level features as well as making the network robust to noise in the training set, a symmetric FCNN is designed. The improvement introduced by this algorithm shown in Table IV is almost 3% in terms of mean IoU. Adding more convolution, dropout, and upsampling layers seem to have almost the same impact of around 1%

TABLE III

NUMERICAL RESULTS OF FCN32S-ALEXNET USING DIFFERENT VALUES OF $\lambda_{\text{LANE}}$ DURING TRAINING. THE BASE NETWORK IS VGG16

| $\lambda_{lane}$ value | 1 | 50 | 100 | 200 | 300 | 308 | 350 | 400 | 500 | 1000 |
|---|---|---|---|---|---|---|---|---|---|---|
| mean IoU | 55.23 | 56.77 | 57.22 | 57.93 | 58.12 | 58.21 | 58.45 | **59.06** | 58.76 | 57.32 |

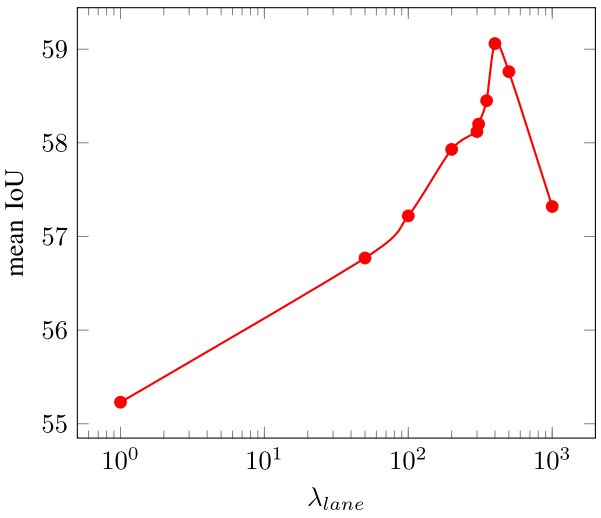



Fig. 12. Performance of FCN32s network with AlexNet as a backbone network on different $\lambda_{\text{lane}}$ values during training. The ratio between lane marking and background pixels in train, trainval, and test sets are 389, 418, and 308, respectively.

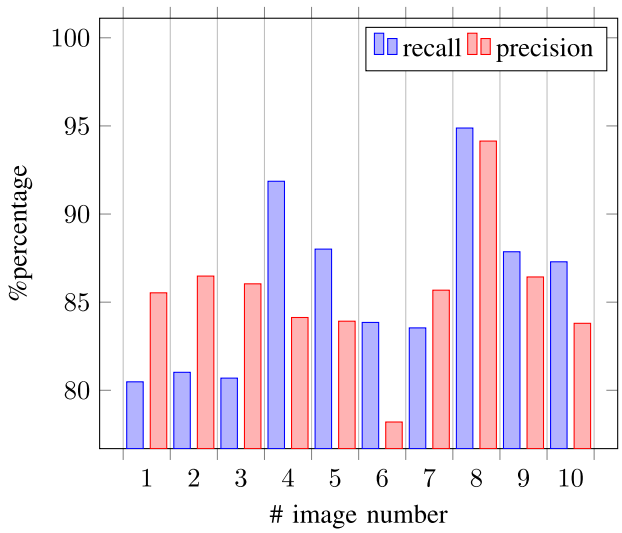



Fig. 13. Evaluation of the Aerial LaneNet network with total recall and precision values for each test image.

TABLE IV

IMPACT OF ADDED CONVOLUTIONS, DROPOUT, AND UPSAMPLING LAYERS TO SHAPE SYMMETRIC FCNN ON THE AERIALLANES18 DATA SET. THE BASE NETWORK IS VGG16

| Network | pixel acc. | mean acc. | mean IoU | f.w. IoU | dice s. c. |
|---|---|---|---|---|---|
| FCN-8s [20] (A,B and C layers) | 99.73 | 66.12 | 62.79 | 99.53 | 51.67 |
| FCNN (A,B,C and D layers) | 99.73 | 67.42 | 63.45 | 99.54 | 52.33 |
| FCNN (A,B,C,D and conv layers) | 99.74 | 68.25 | 64.23 | 99.54 | 53.25 |
| Symmeteric FCNN | **99.74** | **69.57** | **65.10** | **99.55** | **55.08** |

point on the mean IoU. This indicates that even though deeper network could basically improve the performance, the major problem is not their depth. An observation of symmetric FCNN networks shows that even if the network is deep, the algorithm has some difficulty to segment small lane markings. Due to the nature of low-frequency spectral analysis of FCNN, lane markings are smoothed and removed after convolution and average pooling operations. To address this problem, wavelet transform of the input image is inserted into the network.

*4) The Effect of DWT:* A multiresolution analysis using different levels of wavelet transform augments the performance by considering lane-marking objects at different scales. Table V indicates that a combination of the first four DWT decomposition levels results in the best performance, confirming our motivation for multiresolution analysis. In our experiments, we noticed that the addition of a fifth level worsens the results, which could be due to small-size lane markings, since most of their details have already been discarded.

In order to further improve the performance, we replaced the VGG16 base network with the ResNet-101 [14] network, which has a better performance on the ImageNet data set in comparison with VGG16. We inserted DWT levels after the first pooling layer in stage 1 and after the first convolution layer with a stride of 2 in each stage from stage 2 to stage 4. We did not insert DWT's fifth level to stage 5 due to our observation in the DWT's fifth-level insertion after the last pooling layer in VGG16 (see Table V).

As wavelet transform decomposition is made of horizontal, vertical, diagonal details, as well as an approximation component, the investigation is carried out to investigate the effect of each component.

*5) Effect of DWT Components:* According to Table VI, horizontal and vertical components have considerably more impact than the other two. Although the diagonal component also increases mean IoU by almost 2% points, it has less effect than the rather horizontal and vertical components of almost 5%. This indicates that the majority of lane markings are present in the horizontal and the vertical DWT components. The approximation part, however, worsens the performance. This could be due to the fact that this part does not carry sparse information about lane marking as other parts. Experiments with orders of Daubeschies wavelet transforms higher than 1 have resulted in a lower performance of 1.45 mean IoU for

TABLE V

EVALUATION OF AERIAL LANENET FOR FUSION OF EACH LEVEL OF DWT TO SYMMETRIC FCNN WITH COST-SENSITIVE LOSS FUNCTION. IN ADDITION, THE COMPARISON BETWEEN FCN-8S [20] WITH AND WITHOUT FIRST-LEVEL DWT IS PROVIDED

| Network | base network | pixel acc. | mean acc. | mean IoU | f.w. IoU | dice s. c. |
|---|---|---|---|---|---|---|
| FCN-8s [20] | VGG16 | 99.73 | 66.12 | 62.79 | 99.53 | 51.67 |
| FCN-8s - 1st DWT level | VGG16 | 99.75 | 69.67 | 66.24 | 99.56 | 55.14 |
| Aerial LaneNet - 1st DWT level | VGG16 | 99.77 | 75.86 | 70.16 | 99.60 | 61.23 |
| Aerial LaneNet - 1st, 2nd DWT level | VGG16 | 99.79 | 80.83 | 73.57 | 99.62 | 65.55 |
| Aerial LaneNet - 1st, 2nd, 3rd DWT level | VGG16 | 99.80 | 84.32 | 76.72 | 99.65 | 69.61 |
| Aerial LaneNet - 1st, 2nd, 3rd, 4th DWT level | VGG16 | 99.81 | 85.72 | 77.78 | 99.67 | 71.17 |
| Aerial LaneNet - 1st, 2nd, 3rd, 4th DWT level | ResNet-101 | **99.81** | **85.95** | **77.98** | **99.68** | **71.42** |
| Aerial LaneNet - 1st, 2nd, 3rd, 4th, 5th DWT level | VGG16 | 99.80 | 84.01 | 76.64 | 99.65 | 70.25 |

TABLE VI

EVALUATION OF IMPACT OF DIFFERENT DWT DECOMPOSITIONS IN THE FIRST LEVEL ON LANE-MARKING SEGMENTATION, INCLUDING HORIZONTAL, HORIZONTAL AND VERTICAL, HORIZONTAL, VERTICAL, AND DIAGONAL DETAILS AS WELL AS ALL OF DECOMPOSITIONS CONSISTING OF APPROXIMATION PART. THE BASE NETWORK IS VGG16

| Network | pixel acc. | mean acc. | mean IoU | f.w. IoU | dice s. c. |
|---|---|---|---|---|---|
| horizontal | 99.78 | 79.72 | 71.96 | 99.62 | 64.34 |
| horizontal and vertical | 99.80 | 84.03 | 75.84 | 99.65 | 68.56 |
| horizontal, vertical and diagonal | **99.81** | **85.72** | **77.78** | **99.67** | **71.17** |
| horizontal, vertical, diagonal and approximation | 99.80 | 83.21 | 76.02 | 99.65 | 69.23 |

TABLE VII

EVALUATION OF FUSION OF DWT WITH SYMMETRIC FCNN IN DIFFERENT LOCATIONS. THE BASE NETWORK IS VGG16. THE FUSION IS CONCATENATION IN ALL CASES

| Fusion | After first conv | After pooling | Before pooling |
|---|---|---|---|
| mean IoU | 76.23 | **77.78** | 75.42 |

TABLE VIII

CONFUSION MATRIX OF AERIAL LANENET WITH THE BEST PERFORMANCE USING THE VGG16 BASE NETWORK. MATRIX SHOWS THE NUMBER OF SAMPLES FOR EACH CLASS PREDICTED BY THE SYSTEM. DUE TO THE UNBALANCED MULTICLASS PROBLEM, PERCENTAGE NUMBERS FOR EACH CLASS SHOW THE NORMALIZED RECALL RATES. CONFUSION MATRIX SHOWS THE NUMBER OF CORRECT AND WRONG CLASSIFIED PIXELS ALONG WITH NORMALIZED VALUES

| | | **Actual Labels** | | |
|---|---|---|---|---|
| | | Lane Marking | Background | **Class Precision** |
| **Predicted Labels** | Lane Marking | 473313<br>71.55% | 205431<br>0.10% | 69.73% |
| | Background | 188196<br>28.45% | 209396100<br>99.90% | 99.91% |
| | **Class Recall** | 71.55% | 99.90% | **Total Accuracy**<br>mean: 85.72%<br>absolute: 99.81% |

TABLE IX

AERIAL LANENET COMPARISON WITH THE STATE-OF-THE-ART ALGORITHMS. ALL NUMBERS ARE IN [%]

| Network | pixel acc. | mean acc. | mean IoU | f.w. IoU | dice s. c. |
|---|---|---|---|---|---|
| DeepLab[21] | 99.73 | 68.02 | 63.95 | 99.54 | 53.07 |
| UNet[23] | 99.73 | 67.25 | 63.39 | 99.54 | 52.12 |
| FCN-8s [20] | 99.73 | 66.12 | 62.79 | 99.53 | 51.67 |
| DeepLabv3 [34] | 99.68 | 53.79 | 53.24 | 99.38 | 12.26 |
| DeepLabv3+ [35] | 99.79 | 78.23 | 73.18 | 99.62 | 62.71 |
| Aerial LaneNet | **99.81** | **85.95** | **77.98** | **99.68** | **71.42** |

db2, which could be due to less appearance of the lane marking in higher Daubeschies orders.

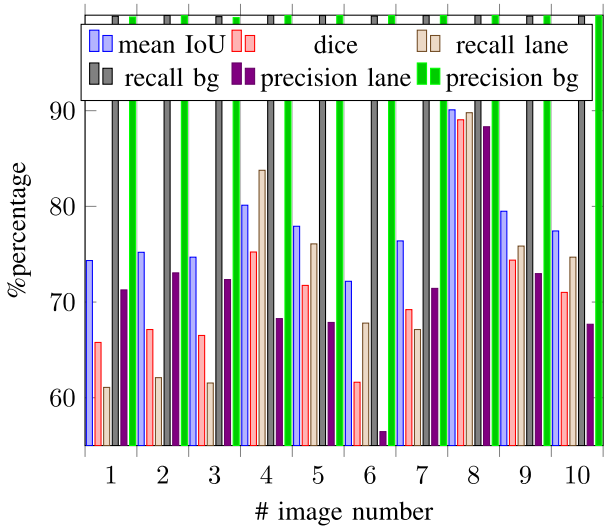



Fig. 14. Evaluation of the Aerial LaneNet network on each test image with mean IoU, dice, and recall and precision values for each class.

*6) Varied Possible Fusions:* As shown in Fig. 8, Table VII reports the result of different DWT fusions with symmetric FCNN. We have considered three different fusion locations. The fusion can be either after the pooling layers or convolution layer or before the pooling layers. Before the first pooling layer, due to dimension incompatibility, the fusion is not possible. Results in Table VII show that placing the fusion right after the pooling layers results in the best performance. The reason for this phenomenon could be the extraction of high-level features by the subsequent convolution layers. On the contrary, the fusion of DWT decomposition before pooling layers leads to a decrease in mean IoU. This could be due to the reason that DWT representation is pooled by the next pooling layer that smooths the representation. However, this degradation is not significant, as lane-marking

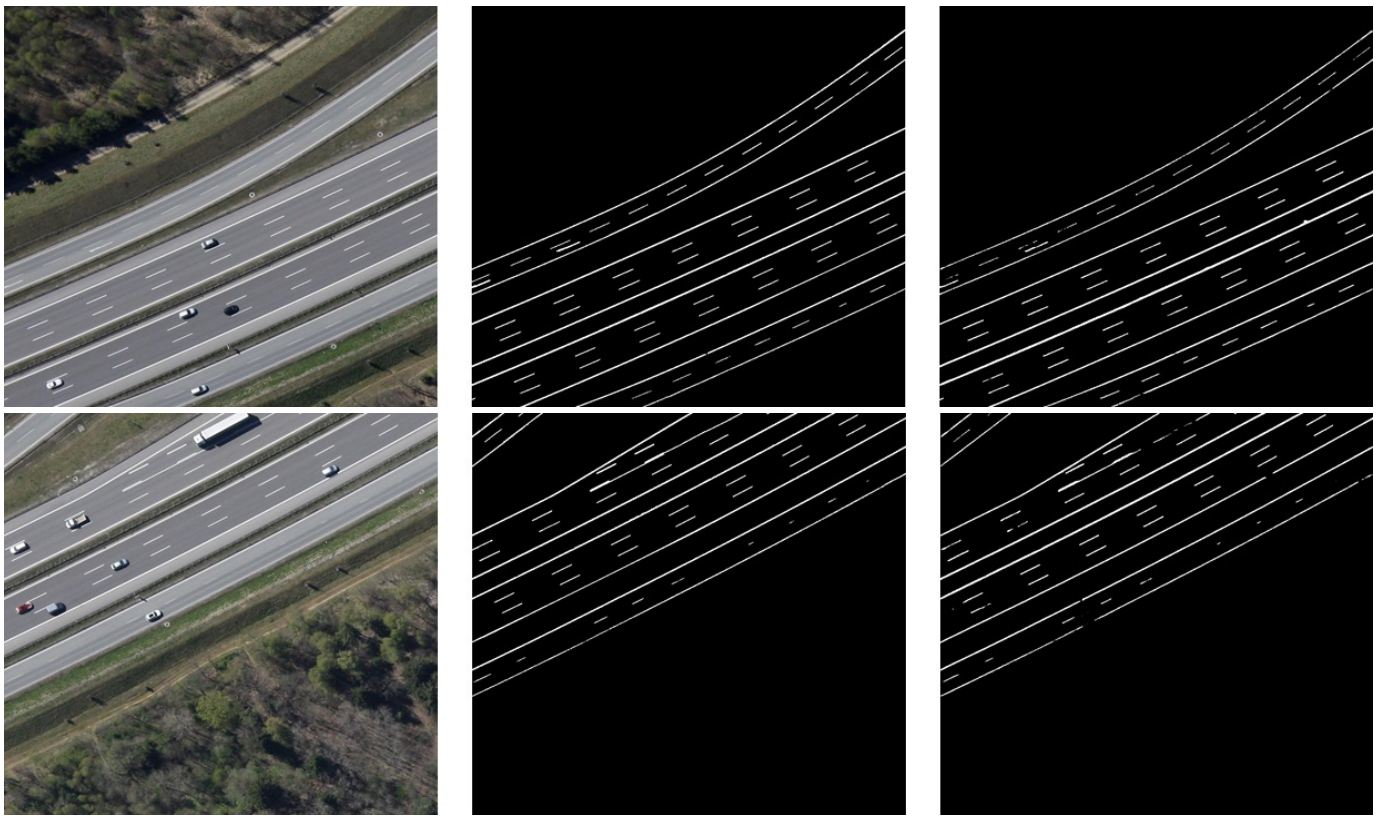


Fig. 15. Examples of results using the Aerial LaneNet approach with the best performance. (Left column) Input images. (Middle column) Ground truth. (Right column) Images are predictions.

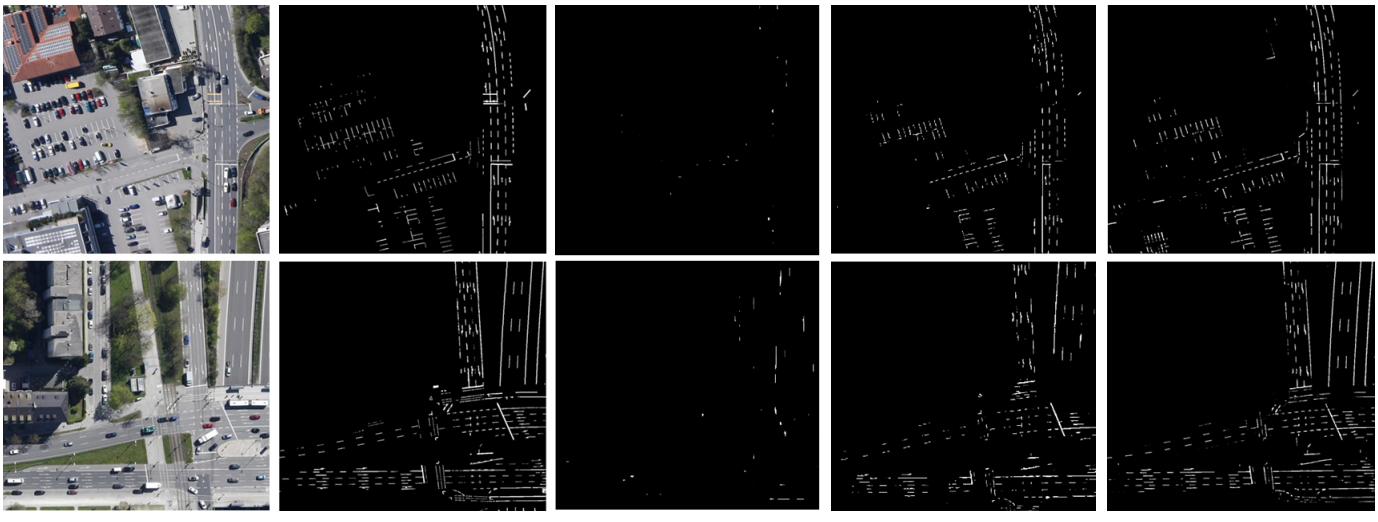



Fig. 16. Qualitative comparison of Aerial LaneNet with ground truth and the state-of-the-art algorithms DeepLabv3 and DeepLabv3+. (a) Input patch. (b) Ground truth. (c) DeepLabv3. (d) DeepLabv3+. (e) Aerial LaneNet.

pixels have higher values compared with neighboring pixels, and in max-pooling operation, the maximum value is chosen.

*7) Confusion Matrix Investigation:* In order to evaluate true and false positives/negatives in our method as well as precision and recall, we have considered the confusion matrix of the configuration for the best performance. Table VIII indicates that in spite of a heavily unbalanced data set, the system is able to achieve a lane-marking pixel (pixelwise) accuracy of 71.55%.

In spite of different illumination conditions introduced by shadows, different shapes, and sizes, the network is able to classify background pixels with 0.1% false positive compared with 99.8% true negative pixels. This indicates how robust the

Fig. 17. Test image with overlaid prediction and ground truth. Ground truth that has not been predicted has been illustrated with dark blue and prediction is depicted with pink.

system is in the presence of the very complex background and objects similar to lane marking. However, the false negatives are still high.

The majority of false negative cases come from straight and dot-shaped lane markings. In the straight lane markings, the output width of the system is almost in all of cases

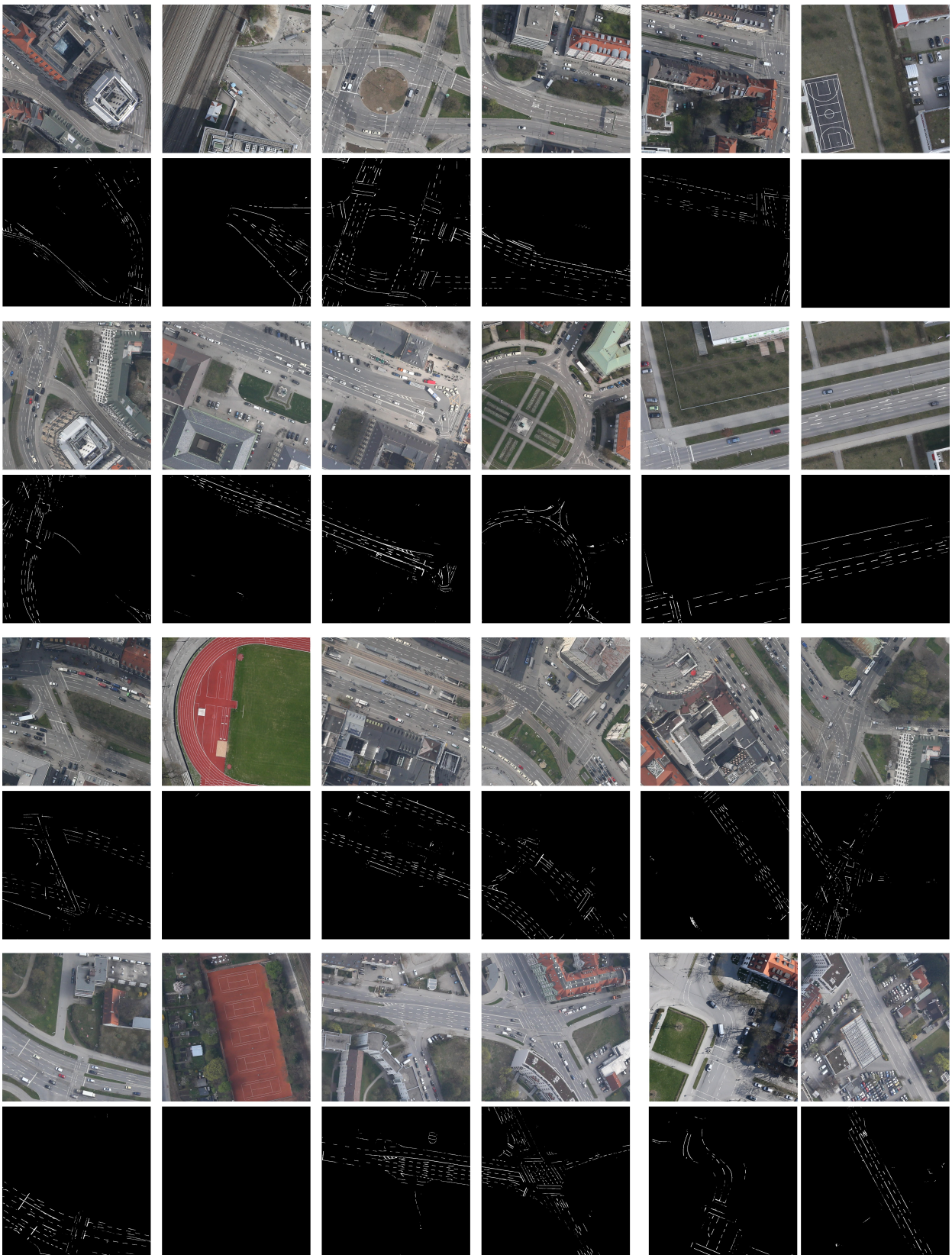


Fig. 18. New test patch images taken in different days, GSD, and camera angles in comparison with the AerialLanes18 data set. Each patch has been shown with the corresponding lane-marking segmentation.

narrower than ground truth. This indicates that this architecture is not able to segment boundaries accurately. Although a morphological operation could increase the performance in this case dramatically, it is not interesting from a research point of view and we do encourage other researchers not to use it in next studies on this data set for benchmarking.

Fig. 19. New test patch images taken in different days, GSD, and camera angles in comparison with the AerialLanes18 data set. Lane-Marking prediction has been overlaid on patches in order to illustrate the localization accuracy of the Aerial LaneNet network.

As mentioned, dot-shaped objects yield a considerable number of false negatives. These objects are as small as 5×5 pixels, which makes them difficult to segment. However, as we do not have access to the information of which pixel belongs to which class in the current annotation, we cannot report a number in this case.

Another and important source of false negative is shadows. As shadows occur rarely, the network has not been able to learn shadows to segment lane markings accordingly. Regarding rare objects, such as "BUS" signs, speed limits, disabled parking places, turn signs, and so on, the same phenomenon is happening. These classes do not occur often, and as in deep convolutional neural networks, a big number of training samples are needed to train the network; the performance in these cases is not high.

*8) Comparison With the State of the Art:* We also compared Aerial LaneNet with FCN-8s, DeepLab [21], UNet [23], and the state-of-the-art method DeepLabv3 [34][1], and its newer version DeepLabv3+ [35][1] in Table IX. Interestingly, there is a big gap between DeepLabv3+ and DeepLabv3. The reason is that DeepLabv3 uses monotonically increasing atrous rates, in which in spite of being effective to obtain large receptive field to segment large-size objects, it severely damages information from small objects, such as lane markings. In contrast, DeepLabv3+ uses a multiscale encoder containing atrous convolutions to obtain a multiscale contextual information, and in the decoder part, a simple yet effective module refines the segmentation outputs to improve the boundary segmentation. The qualitative comparison has been provided in Fig. 16. The multiscale processing helps the DeepLabv3+ to achieve significantly better results than its previous version. This is mostly due to the decoder part that improves the boundary region segmentation. However, it does not have a satisfactory performance on tiny lanemarkings despite its very good performance in the terrestrial images. The results shows that recovering high-frequency information of image pixels by inserting DWT into different levels of CNNs leads to a considerably better performance of 4% mIOU in comparison with the DeepLabv3+ algorithm. Aerial LaneNet outperforms all of these networks in Table IX, showing the high accuracy of our method.

*9) Qualitative Analysis:* In Fig. 13, recall and precision values for each test image are reported. These values are consistent and there is not a big difference between recall and precision. In Fig. 14, mean IoU and dice for each test image as well as recall and precision for each class have been reported. As for total recall and precision values, these criteria are consistent among test images. Recall and precision values for each class have also been computed.

One can notice that precision and recall for background class is very high, which is due to the unbalanced task: there is a big gap between recall and precision for the lane-marking class and for the background class. In order to evaluate the results qualitatively, Fig. 15 illustrates the lane-marking segmentations of different patches of size 1024 × 1024 pixels compared with the ground truth. The left images are input test patches. The middle patches are the ground truth. The patches on the right are the corresponding predictions. Fig. 15 shows a very good performance in the segmentation of both straight and dashed lines in highways. It is very interesting that in some cases, the network has localized correct lane marking, which is not even annotated in the ground truth. However, there are also some failure cases. In the same figures, one can note that shadows, narrower straight lines, very small lane markings, and similar objects in the background are the main reasons for false negative and positive outputs. Fig. 15(a) shows that the shadow caused by a truck has caused degradation in lane-marking segmentation. Objects with a similar appearance still are a challenge, e.g., the roof structures at the bottom-left part of the image in Fig. 15(b), which look similar to lane markings, have been classified as lane marking. Also, in the same image, when it comes to smaller

[1]https://github.com/tensorflow/models/tree/master/research/deeplab

lane-marking objects, the network is not performing as good. In spite of these failure cases, the overall performance proves the concept of effective semantic segmentation of lane marking using enhanced FCNNs with DWT information. In Fig. 17, predictions have been overlaid on the original test images after stitching prediction patches together. In these images, predicted lane-marking pixels and undetected ones are reported in red and blue, respectively. In shadow areas, the network has difficulties to segment lane markings.

*10) Cross-Domain Generalization:* In order to evaluate the robustness of our algorithm to variations: GSD, camera angle view, and illumination conditions, we have considered multiple flights on different days, altitudes, and angles with the DLR 3K camera. Results are reported in Fig. 18.

We have overlaid predictions on the test patches of a new data set in Fig. 19. The performance shows a good generalization capability of the network, which appears robust to most of the challenges mentioned earlier such as small size, different camera angles, and presence of objects similar to lane marking such as lanes in soccer fields.

## V. Conclusion

In this paper, we have introduced a reliable and fast algorithm to segment very small objects, such as lane markings in aerial imagery with high accuracy and robustness. We presented the Aerial LaneNet network based on the idea of enhancing FCNNs with wavelet transformation coefficients for pixelwise semantic segmentation, which enables a full-spectral and multiscale analysis resulting in the considerable improvement compared with our FCNN based-line network. We have shown that although using subsampling layers or atrous convolutions to obtain large receptive fields yields a very good performance in terrestrial images, they cause a vital data lost for pixelwise semantic segmentation of tiny objects, which leads to a considerable performance degradation. Therefore, the lost information should be either injected into the network or be kept by removing subsampling layers to recover the lost data. In this paper, we selected the first strategy showing impressive performance improvement in comparison with the state-of-the-art methods. We conclude that for tiny object segmentation, both high- and low-frequency information of pixels should be analyzed, while CNNs perform mostly low-frequency analysis due to using pooling and convolution layers. The limitations of Aerial LaneNet are in shadow areas, semantic signs on the roads, as well as washed out lane markings. We also introduced the AerialLanes18 data set the first high-quality aerial lane marking data set as a benchmark in this domain. Using different levels of wavelet decomposition leads to a multiresolution data analysis which is important in extracting lane markings, as objects appear at different scales. In the future, we will investigate improving the performance by processing shadow areas differently.

## References


[1] H. Jin, M. Miska, E. Chung, X. Li, and Y. Feng, "Road feature extraction from high resolution aerial images upon rural regions based on multi-resolution image analysis and Gabor filters," in *Remote Sensing Advanced Techniques and Platforms*. Rijeka, Croatia: Intechopen, 2012. [Online]. Available: https://www.intechopen.com/download/pdf/37525

[2] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining Knowl. Discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[3] H. Jin and Y. Feng, "Automated road pavement marking detection from high resolution aerial images based on multi-resolution image analysis and anisotropic Gaussian filtering," in *Proc. ICSPS*, 2010, pp. V1-337–V1-341.

[4] H. Jin, Y. Feng, and M. Li, "Towards an automatic system for road lane marking extraction in large-scale aerial images acquired over rural areas by hierarchical image analysis and Gabor filter," *Int. J. Remote Sens.*, vol. 33, no. 9, pp. 2747–2769, 2012.

[5] M. Javanmardi, E. Javanmardi, Y. Gu, and S. Kamijo, "Towards high-definition 3D urban mapping: Road feature-based registration of mobile mapping systems and aerial imagery," *Remote Sens.*, vol. 9, no. 10, p. 975, 2017.

[6] J. Huang, H. Liang, Z. Wang, Y. Song, and Y. Deng, "Lane marking detection based on adaptive threshold segmentation and road classification," in *Proc. ROBIO*, 2014, pp. 291–296.

[7] S. Hinz and A. Baumgartner, "Automatic extraction of urban road networks from multi-view aerial imagery," *J. Photogramm. Remote Sens.*, vol. 58, nos. 1–2, pp. 83–98, 2003.

[8] G. Máttyus, S. Wang, S. Fidler, and R. Urtasun, "HD maps: Fine-grained road segmentation by parsing ground and aerial images," in *Proc. CVPR*, 2016, pp. 3611–3619.

[9] M. Gellert, L. Wenjie, and U. Raquel, "Deeproadmapper: Extracting road topology from aerial images," in *Proc. ICCV*, 2017, pp. 3458–3466.

[10] O. Tournaire, N. Paparoditis, and F. Lafarge, "Rectangular road marking detection with marked point processes," in *Proc. Conf. Photogramm. Image Anal.*, vol. 3, 2007. [Online]. Available: http://www.isprs.org/proceedings/XXXVI/3-W49/PartA/papers/149_pia07.pdf

[11] H. Mayer, S. Hinz, U. Bacher, and E. Baltsavias, "A test of automatic road extraction approaches," in *Proc. Int. Arch. Photogram., Remote Sens. Spatial Inf. Sci.*, 2006, pp. 209–214.

[12] W. Wang, N. Yang, Y. Zhang, F. Wang, T. Cao, and P. Eklund, "A review of road extraction from remote sensing images," *J. Traffic Transp. Eng.*, vol. 3, no. 3, pp. 271–282, 2016.

[13] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. ICRL*, 2015.[Online]. Available: https://iclr.cc/archive/www/2015.html

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, 2016, pp. 770–778.

[15] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. CVPR*, 2017, pp. 4700–4708.

[16] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[17] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. ICCV*, 2017, pp. 2980–2988.

[18] W. Liu *et al.*, "SSD: Single shot multibox detector," in *Proc. ECCV*, 2016, pp. 21–37.

[19] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. CVPR*, 2017, pp. 6517–6525.

[20] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. CVPR*, 2015, pp. 3431–3440.

[21] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected CRFs," in *Proc. ICLR*, 2014. [Online]. Available: https://arxiv.org/pdf/1412.7062.pdf and https://iclr.cc/archive/www/doku.php%3Fid=iclr2015:accepted-main.htmlunfortunately

[22] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. CVPR*, 2017.

[23] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention—MICCAI*. Cham, Switzerland: Springer, 2015, pp. 234–241.

[24] J. Kim and C. Park, "End-to-end ego lane estimation based on sequential transfer learning for self-driving cars," in *Proc. CVPR Workshops*, 2017, pp. 1194–1202.

[25] A. Gurghian, T. Koduri, S. V. Bailur, K. J. Carey, and V. N. Murali, "DeepLanes: End-to-end lane position estimation using deep neural networks," in *Proc. CVPR Workshops*, 2016, pp. 38–45.

[26] S. Lee *et al.*, "VPGNet: Vanishing point guided network for lane and road marking detection and recognition," in *Proc. ICCV*, 2017, pp. 1965–1973.

[27] M. Aly, "Real time detection of lane markers in urban streets," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2008, pp. 7–12.

[28] (2017). *Tusimple Benchmark*. [Online]. Available: http://benchmark.tusimple.ai

[29] M. Cordts *et al.*, "The cityscapes dataset for semantic urban scene understanding," in *Proc. CVPR*, 2016, pp. 3213–3223.

[30] G. Neuhold, T. Ollmann, S. R. Bulò, and P. Kontschieder, "The mapillary vistas dataset for semantic understanding of street scenes," in *Proc. ICCV*, 2017, pp. 5000–5009.

[31] C. Hu, L.-J. Jiang, and J. Bo, "Wavelet transform and morphology image segmentation algorism for blood cell," in *Proc. 4th IEEE Conf. Ind. Electron. Appl.*, May 2009, pp. 542–545.

[32] S. Mallat, *A Wavelet Tour of Signal Processing*. San Diego, CA, USA: Academic, 2009.

[33] S. Fujieda, K. Takayama, and T. Hachisuka. (2017). "Wavelet convolutional neural networks for texture classification." [Online]. Available: http://arxiv.org/abs/1707.07394

[34] L. Chen, G. Papandreou, F. Schroff, and H. Adam. (2017). "Rethinking atrous convolution for semantic image segmentation." [Online]. Available: https://arxiv.org/abs/1706.05587

[35] L. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. (2018). "Encoder-decoder with atrous separable convolution for semantic image segmentation." [Online]. Available: http://arxiv.org/abs/1802.02611

[36] H. Zhao, X. Qi, X. Shen, J. Shi, and J. Jia. (2017). "ICNet for real-time semantic segmentation on high-resolution images." [Online]. Available: http://arxiv.org/abs/1704.08545

[37] S. Xie, R. B. Girshick, P. Dollár, Z. Tu, and K. He. (2016). "Aggregated residual transformations for deep neural networks." [Online]. Available: http://arxiv.org/abs/1611.05431

[38] F. Chollet. (2016). "Xception: Deep learning with depthwise separable convolutions." [Online]. Available: http://arxiv.org/abs/1610.02357

[39] G. Huang, Z. Liu, and K. Q. Weinberger. (2016). "Densely connected convolutional networks." [Online]. Available: http://arxiv.org/abs/1608.06993

[40] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. CVPR*, 2005, pp. 886–893.

[41] I. Daubechies, "Ten lectures on wavelets," in *Proc. CBMS-NSF Regional Conf. Ser. Appl. Math.*, vol. 61, 1992, no. 4. [Online]. Available: http://bookstore.siam.org/cb61/

[42] A. Krizhevsky, I. Sutskever, and H. E. Geoffrey, "ImageNet classification with deep convolutional neural networks," in *Proc. NIPS*, 2012, pp. 1097–1105.

[43] P. Reinartz, J. Tian, H. Arefi, T. Krauß, G. Kuschk, T. Partovi, and P. d'Angelo, "Advances in DSM generation and higher level information extraction from high resolution optical stereo satellite datam," in *Proc. 34th Earsel Symp.*, Warsaw, Poland, 2014, pp. 16–20.

[44] F. Kurz, S. Türmer, O. Meynberg, D. Rosenbaum, H. Runge, P. Reinartz, and J. Leitloff, "Low-cost optical camera systems for real-time mapping applications," *Photogrammetrie-Fernerkundung-Geoinf.*, vol. 2012, no. 2, pp. 159–176, 2012. [Online]. Available: https://www.ingentaconnect.com/content/schweiz/pfg/2012/00002012/00000002/art00007

[45] F. Kurz, O. Meynberg, D. Rosenbaum, S. Türmer, P. Reinartz, and M. Schroeder, "Low-cost optical camera system for disaster monitoring," *Int. Arch. Photogram., Remote Sens. Spatial Inf. Sci*, vol. B8, pp. 33–37, Jul. 2012.

[46] K. Liu and G. Mattyus, "Fast multiclass vehicle detection on aerial images," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 9, pp. 1938–1942, Sep. 2015.

[47] O. Meynberg, S. Cui, and P. Reinartz, "Detection of high-density crowds in aerial images using texture classification," *Remote Sens.*, vol. 8, no. 6, p. 470, 2016.

[48] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. CVPR*, Jun. 2009, pp. 248–255.

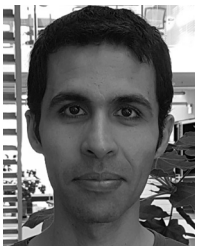
[49] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. CVPR*, Jun. 2015, pp. 1–9.

[50] OpenStreetMap Contributors. (2017). *Planet Dump Retrieved From*. [Online]. Available: https://planet.osm.org

[51] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, 2015.

[52] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," in *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[53] M. Abadi *et al.* (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. [Online]. Available: https://www.tensorflow.org/

**Seyed Majid Azimi** received the B.Sc. degree in electronics engineering from the University of Zanjan, Zanjan, Iran, in 2009, and the M.Sc. degree in computer and communications technology from the University of Saarland, Saarbrücken, Germany, in 2016. He is currently pursuing the Ph.D. degree with the Technical University of Munich, Munich, Germany, with a focus on traffic and infrastructure monitoring from remote sensing data using deep learning methods.

Since 2016, he has been a Scientific Researcher with the Department of Photogrammetry and Image Analysis, Remote Sensing Technology Institute, German Aerospace Center, Weßling, German. His research interests include (embedded) computer vision and machine learning for object detection, segmentation, and tracking.

**Peter Fischer** received the Dipl.Ing. (FH) degree in cartography from the University of Applied Science Munich, Munich, Germany, in 2010, and the M.Sc. degree in geodesy from the Technical University of Munich, Munich, in 2013.

From 2013 to 2017, he was a Scientific Assistant and Project Manager with the Remote Sensing Technology Institute, German Aerospace Center, Weßling, Germany, for five years. In 2018, he joined the Department of Sensor Data Fusion (I/EF-24), AUDI AG, Ingolstadt, Germany. Besides of this he is contributing to the Master course of Geoinformatics at University of Augsburg, Augsburg, Germany. His research interests include machine learning and computational intelligence, especially its applications in the field of cartography and remote sensing.

**Marco Körner** (M'15) received the Diploma (Dipl.Inf.) and Ph.D. (Dr. rer. nat.) degrees in computer sciences (with minor in psychology) from Friedrich Schiller University, Jena, Germany, in 2009 and 2016, respectively.

From 2009 to 2015, he was a member of the Computer Vision Group, Jena. He was a Visiting Researcher with the Centro de Investigación en Computación, Instituto Politecnico Nacional, Mexico City, Mexico, in 2012, and the University of California at San Diego, San Diego, CA, USA, in 2014. Since 2015, he has been a Senior Researcher and the Deputy Head with the Chair of Remote Sensing Technology, Technical University of Munich, Munich, Germany. His research interests include machine learning in computer vision, particularly for application in automotive, remote sensing, and biomedical scenarios.

**Peter Reinartz** (M'09) received the Diploma (Dipl.Phys.) degree in theoretical physics from the University of Munich, Munich, Germany, in 1983, and the Ph.D. (Dr.Ing.) degree in civil engineering from the University of Hannover, Hanover, Germany, in 1989. His Ph.D. dissertation was on optimization of classification methods for multispectral image data.

He is currently with the Head of the Department of Photogrammetry and Image Analysis, Remote Sensing Technology Institute, German Aerospace Center, Weßling, Germany. He holds a professorship for computer science at the University of Osnabrueck, Osnabrück, Germany. He has more than 30 years of experience in image processing and remote sensing. He has over 400 publications in these fields. He is involved in using remote sensing data for disaster management and using high-frequency time series of airborne image data for real-time image processing and for operational use in case of disasters and for traffic monitoring. His research interests include machine learning, stereophotogrammetry and data fusion using space borne and airborne image data, generation of digital elevation models, and interpretation of very-high-resolution data from sensors, such as WorldView, GeoEye, and Pleiades.