This ICCV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Road Anomaly Detection by Partial Image Reconstruction with Segmentation Coupling

Tomas Vojir*† Tomas Sipka† Rahaf Aljundi‡
Nikolay Chumerin‡ Daniel Olmeda Reino‡ Jiri Matas†

Abstract

We present a novel approach to the detection of unknown objects in the context of autonomous driving. The problem is formulated as anomaly detection, since we assume that the unknown stuff or object appearance cannot be learned. To that end, we propose a reconstruction module that can be used with many existing semantic segmentation networks, and that is trained to recognize and reconstruct road (drivable) surface from a small bottleneck. We postulate that poor reconstruction of the road surface is due to areas that are outside of the training distribution, which is a strong indicator of an anomaly. The road structural similarity error is coupled with the semantic segmentation to incorporate information from known classes and produce final per-pixel anomaly scores. The proposed JSR-Net was evaluated on four datasets, Lost-and-found, Road Anomaly, Road Obstacles, and FishyScapes, achieving state-of-art performance on all, reducing the false positives significantly, while typically having the highest average precision for wide range of operation points.

1. Introduction

Autonomous vehicles have quickly become one of the prime application areas of computer vision methods. The range of research topics that have been influenced and stimulated by this rapid development is broad: object detection [33], tracking [31, 52], optical flow estimation [51], stereo [49], monocular depth [9] estimation, semantic segmentation [53], lidar-camera fusion, 3D mapping and self-localisation [50, 23], to name a few. For many of the problems, the best performing methods are, or include, deep neural networks, which have a voracious appetite for training data; currently mainly labeled data. As a consequence, a vast data acquisition and labeling effort has been taking place, together with research into the use of synthetic data [36, 18], virtual environments and simulators [16], and unsupervised learning [2].

In this work, we present an approach to detect road anomalies in a semantic segmentation setting, in the context of spotting arbitrary "stuff" or objects on the road surface. The detection of "stuff" on the road is currently formulated as the detection of the known and "unknown unknown" (unexpected and not considered types of "stuff"), which is typical for one-class classification [42] and outlier/anomaly detection [6] problems. While the semantic segmentation explains the scene, as viewed in an image, decomposing it into a set of known categories, modelling an appearance that is outside the known classes, or is out of distribution, requires additional consideration.

In the proposed approach, we tightly combine information about the known class, "road" in our application, with a strategy for estimating previously unseen objects and stuff. The known class information is captured by a standard segmentation deep neural network. The performance on data close to the training distribution is excellent, but its behaviour on unseen data is variable, as experiments show. We therefore add a reconstruction network module, arguing, and experimentally verifying, that a failure to reconstruct reliably and predictably is an indicator of an anomaly.

The main contributions of the paper are the following: (*i*) a novel use of image reconstruction to distinguish one known semantic class from anomalies and outliers, originating from unknown appearance distributions, by explicitly requiring poor reconstruction outside of the known class, (*ii*) a trainable coupling of information from reconstruction and semantic segmentation that is able to exploit efficiently the two sources of information, (*iii*) a plug-and-play module that can be used with many segmentation networks without the need of re-training the segmentation part, *i.e.*, adding novel functionality without any semantic segmentation performance loss and with a minimal computational and memory overhead. (*iv*) Achieving state-of-the-art results and better generalization to out-of-distribution data than competing methods.

We show quantitative results on three standard and one derivative datasets – Lost-and-found [34],

[*] Corresponding author, vojirtom@fel.cvut.cz

[†] Czech Technical University in Prague, Faculty of Electrical Engineering, Czech Republic

[‡] Toyota Motor Europe, Brussels, Belgium

Road Anomaly [28], Road Obstacles [27] and FishyScapes:LaF [4] – where the proposed method, JSR-Net, outperforms the current state-of-the-art methods often by a large margin.

The rest of the paper is structured as follows: Section 2 discusses related work, Section 3 describes the proposed JSR-Net method and its components, Section 4 provides technical details for reproducibility purposes, Section 5 discusses the experimental results and finally Section 6 summarizes the paper conclusions.

2. Related Work

Anomaly Detection, Out of Distribution detection (OOD) or Novelty Detection [6, 39, 21, 26, 22, 5, 38, 12, 28, 47, 46, 25] describes methods that try to detect input data that is out of a given model task or "knowledge" scope.

Out-of-Distribution (OOD). Recently, OOD detection has gained increased interest, as it can provide deep neural networks with the ability to reject input that does not correspond to the "task" the given model has been trained for, avoiding misleading predictions and their consequences. Most methods, however, focus on the image classification problem. While [21] utilizes the predicted class probability after a softmax layer as a score for in-distribution samples, [26] increases the robustness of this approach by adding a small perturbation to the input beforehand. Le *et al.* [25] proposed a detection method based on the Mahalanobis distance of an input sample features (from different layers of a neural network) to the features of training data. These approaches assume one class per image which limits their applicability to the image classification scenario. Closest to our approach are [1, 3]. Cho [1] employs a variational autoencoder trained on the raw in-distribution data and uses the reconstruction probability by the autoencoder as an "in" score. Bevandić *et al.* [3] uses two heads, one for semantic segmentation and the other to detect outliers for the segmentation head. In this work, we deploy an autoencoder-like approach to localize novel regions or anomalous regions by specifically modeling the road appearance and a coupling module to train the interaction between the semantic segmentation output and the road reconstruction error jointly.

Road Anomaly Detection. There are several clusters of approaches to road anomaly detection if the type of input data is considered. Approaches such as [34, 35] rely on stereo cameras and use Stixel representation to detect the anomalies. Other methods [24, 40, 45, 14] detect anomalies from stereo input by analysing the UV-disparity maps. Recently, methods [19, 41] requiring RGB-D data to localise anomalies have been proposed. In this work, we focus on methods that rely solely on monocular camera images. The most relevant methods [12, 28, 47, 46, 27] are described in detail in the sequel.

The RBM [12] method trains a small Restricted Boltzmann Machine on extracted patches of highways to learn an autoencoding of the road patch through a low-dimensional space. During the evaluation, the input image is split into small patches and each patch is autoencoded through the trained network. The absolute difference between the original and the encoded-decoded image patch is used as an indicator of the presence of an anomaly.

The method of Xue *et al.* [47] detects unknown objects on the road by classifying bounding box proposals established from edge maps. In the first step, the image edges on multiple scales are extracted and merged to form super-pixels [15]. The super-pixel representation is further reduced by detecting occluding edges [30], which includes linking super-pixels in homogeneous areas and adding edges at places with estimated depth discontinuity. From these super-pixels, bounding box proposals are sampled [29] and classified by a random forest [13] using 20 ad-hoc features (*e.g.*, color, objectness, or pseudo-distance) to three classes – road, obstacle, and non-road.

Recently, two studies [28, 32] proposed a neural network method that operates on an RGB image and its semantic segmentation. The neural network is trained to detect the discrepancy of the input image and the image generated by pix2pixHD [43] (or [8]) from the semantic segmentation. Wrongly labeled pixels in the semantic segmentation (*e.g.*, "random" labels on the part of the road) cause the generator to create an image with large visual dissimilarity, which can be identified as "anomalous". Similarly, Xia *et al.* [46] proposed a method for detecting anomalies in semantic segmentation. Using an image synthesis module, the input image is synthesized given the predicted segmentation map. Then the synthesized image is compared with the input image. Regions with large differences are considered as anomalies. The synthesized objects are assumed to be similar to the presented objects in the image unless anomalous. The realism of the generated objects in this approach depends on the quality of a GAN module.

An inpainting based method for detecting road anomalies was proposed in [27]. Selected patches are inpainted with a road-like structure and a discrepancy network is used to detect possible anomalies on the road. In contrast, our approach relies on a pre-trained segmentation model and can be plugged-in without the need to train the complete model. Moreover, we do not rely on a generative model, like GANs, to generate a complete image, but rather focus on learning a more constrained appearance of the road. This way, we avoid mode collapse and optimization stability issues. Since the appearance of the road is modelled in a fully convolutional manner, we avoid pitfalls such as missing a large object due to sub-optimal inpaint window size like [27]. Our method is lightweight and easy to train in comparison to the listed state-of-the-art methods.

3. Methods

The main idea behind the proposed method is to learn a low-dimensional, yet robust, latent representation of road surfaces that have relatively low appearance variation (as opposed to anomalous objects) and there exists an abundance of datasets with labeled roads [11, 17, 48] that can be used for this purpose. The latent road representation is used to perform road reconstruction that, when combined with readily available semantic segmentation networks used in *e.g.*, autonomous vehicles, enables robust detection of arbitrary road anomalies. The same principle can be used in different scenarios *e.g.* in naval drones to detect anomalies on the water surface. To this end, a deep neural network model is proposed to learn jointly a road pixel-level reconstruction and fusion with semantic segmentation. The road reconstruction, trained on Cityscapes [11] dataset, coupled with the semantic segmentation shows good generalization on multiple anomaly detection datasets (see results in Section 5), which is especially important for the task of anomaly detection where the lack of comprehensive training datasets is self-evident. The overall network architecture is illustrated in Fig. 1. The two main modules – reconstruction and segmentation coupling – and a simple, yet effective, data augmentation scheme are described in the following sections and the implementation details are provided in Section 4.

3.1. Reconstruction Module

The goal of the reconstruction module is to learn the the appearance of the road (drivable surfaces) in a discriminative way, meaning the road is to be reconstructed with minimal error while the other environment is required to have a large reconstruction error. To that end, we proposed a deep neural network in the form of a decoder that is connected to the backbone of the fixed segmentation network, which was trained independently beforehand. This formulation of the discriminative reconstruction loss together with a small bottleneck allows us to detect anomalies as poorly reconstructed regions in the in the reconstructed error image. By using the fixed backbone (encoder), it allows us to plug the reconstruction module to any "already in-use" semantic segmentation network and train only the Reconstruction Module and the segmentation coupling (described in the following section).

The reconstruction module consists of three key parts: (*i*) the backbone features dimensionality reduction for the decoder bottleneck. This is achieved by processing the backbone features by an atrous spatial pyramid pooling (ASPP) [7] block. The ASPP block serves to exploit information from larger receptive fields and allows the bottleneck to capture the appearance at different scales, which contrasts with *e.g.*, [27], where the fixed size in-painting window limits the maximum area of the detectable anomalous object. (*ii*) The decoder is used to progressively upsample the feature channels and learn how to reconstruct the road from the bottleneck. It consist of four convolutional blocks. Each block chains twice the following operations: bilinear upsampling, 2D convolution, batchnorm, and ReLU nonlinearity. The number of feature channels is progressively reduced down to the final three channels (RBG) in the last convolution layer. (*iii*) The reconstructed RGB image $\hat{I}$ is compared to the input image I using the structural similarity index measure [44], denoted as SSIM, following the Eq. 1:

$$\text{SSIM}(u_{\hat{I}}^{x,y}, v_I^{x,y}) = \frac{(2\mu_u\mu_v + c_1)(2\sigma_{uv} + c_2)}{(\mu_u^2 + \mu_v^2 + c_1)(\sigma_u^2 + \sigma_v^2 + c_2)} \quad (1)$$

where $u_{\hat{I}}^{x,y}, v_I^{x,y}$ are two local windows of $\hat{I}, I$ centered at location (x, y) and μ, σ are the mean and variance of the local window pixel values. The measure has added constants c_1, c_2 for numerical stability and to set the range of the SSIM output. The Eq. 1 is for single-channel input image, while during evaluation each channel of the RGB image is processed independently and the output is the per-channel SSIM measure averaged over the channel dimension. The SSIM incorporates not only the illumination and contrast parts, but also models the structural dependencies of spatially close pixels, as opposed to *e.g.*, MSE, therefore being more robust to imprecision in reconstructed illumination. There exist an efficient implementation[1], using convolution operations, which was modified and used in this work. Note that since the SSIM is a composite measure averaged over a small neighbourhood, it does not necessarily produce an accurate per-pixel image reconstruction as perceived by humans when used in back-propagation as a loss. The experimental comparison with the standard, widely used, L2 norm is provided in the ablation study in Section 5. The final auxiliary reconstruction loss $\mathcal{L}_R$ used to train the reconstruction module is defined as:

$$\mathcal{L}_R = \frac{1}{|M_r|} \sum_{x,y} \max\left[0, \text{SSIM}\left(u_{\hat{I}}^{x,y}, v_I^{x,y}\right) - \xi\right] M_r^{x,y}$$
$$+ \frac{1}{|M_a|} \sum_{x,y} \max\left[0, 1 - \text{SSIM}\left(u_{\hat{I}}^{x,y}, v_I^{x,y}\right) - \xi\right] M_a^{x,y} \quad (2)$$

where M is a binary mask for the road (r) and the anomalies (a) (not road), $|M|$ is the number of non-zero elements, and the ξ is a slack variable to improve convergence. The

[1] https://github.com/Po-Hsun-Su/pytorch-ssim

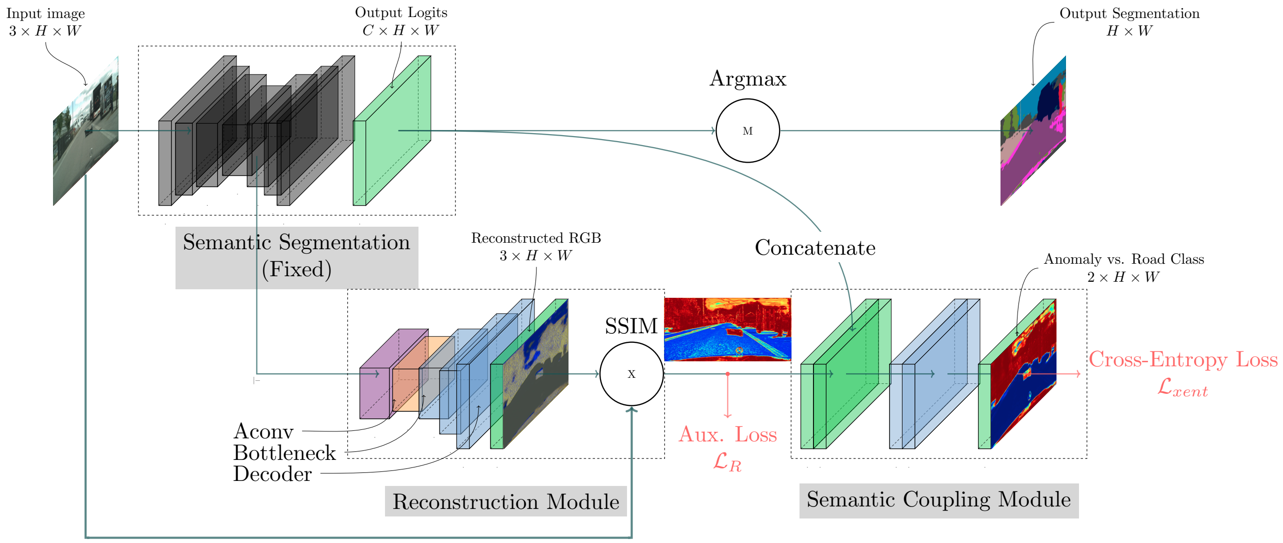

Figure 1. The JSR-Net architecture. The input image is processed by a fixed semantic segmentation network (depicted in gray). The features (from the last layer of the semantic segmentation network backbone) are fed into the reconstruction network, the output of which is a reconstructed version of the input image of the same resolution. The reconstructed image is then compared to the input image using SSIM measure. The per-pixel errors are concatenated with the output logits of the semantic segmentation network and fused by two convolutional blocks. The final output is two maps for the ”road” and ”anomaly” classes.

loss $\mathcal{L}_R$ is scaled down by a factor of two to be in a normalized range $(0, 1)$. Intuitively, the auxiliary reconstruction loss minimizes the reconstruction error on road pixels while maximizing the reconstruction error elsewhere.

3.2. Segmentation Coupling Module

The segmentation coupling module is trained to combine the information of “known classes” encoded in the output logits of the fixed segmentation net with the “unknown anomalies” discovered by the reconstruction module as poorly reconstructed image regions. This trainable coupling of the two sources of information is necessary, since the segmentation networks are often overconfident in the estimation of class likelihood and therefore small anomalous objects on the road are often misclassified (see the results of using only the baseline segmentation networks for anomaly detection in the experiments Section 5).

To learn how to combine the semantic segmentation and reconstruction information, we propose to use a standard convolution block, which is simple but effective for this task. Firstly, the segmentation logits are channel-wise concatenated with the SSIM reconstruction error to form the input to the two convolutional blocks. Each block consists of a 2D convolution layer followed by batch normalization and ReLU non-linearity. The output has two channels corresponding to the anomaly and road class and is normalized by the softmax layer. To train the segmentation coupling, a standard binary cross-entropy loss (Eq. 3) is applied to the two-channel output.

$$\mathcal{L}_{\text{xent}} = -\frac{1}{N}\sum_{n=1}^{N}(1-c^n)\log(1-\hat{c}^n) + c^n\log(\hat{c}^n) \quad (3)$$

where N in number of examples (in our case number of pixels), the $c^n, \hat{c}^n \in \{0, 1\}$ are the true and estimated class labels, respectively, for the n^{th} training example. The final loss is obtained as a sum of $\mathcal{L}_{\text{xent}}$ (Eq. 3) and the scaled auxiliary reconstruction loss $\mathcal{L}_R$ (Eq. 2):

$$\mathcal{L} = \mathcal{L}_{\text{xent}} + 0.5\mathcal{L}_R \quad (4)$$

Note that we do not use explicit weighting of the two losses (only the normalization for the auxiliary loss), since both losses are of a similar scale in normal conditions.

3.3. Synthetic Anomaly Data Augmentation

To further increase the robustness and to alleviate the network spatial bias (*i.e.*, learning that the road label is more likely in the lower part of image), a novel simple augmentation scheme is proposed. The augmentation generates a random number of polygons (up to ten in our case) for a wider variety of possible anomaly shapes. These polygons are then used to crop an image part that belongs to the ”anomaly” label or are filled with a random color and placed randomly inside the road regions. A polygon is created as the convex hull of randomly sampled points inside of a bounding box of random size, in our case the width

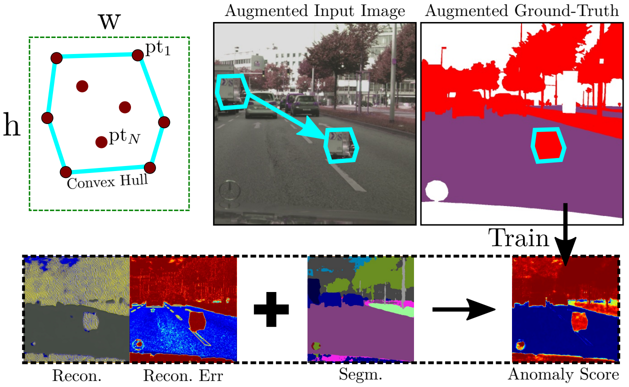

Figure 2. Visualization of the augmentation process. First, N points are randomly sampled inside a bounding box of random size. The convex hull of these points is used as (*i*) a cropping mask for copying a random part of the input image with the anomaly label to a random road location or (*ii*) it is filled with random color with noise. The augmented images with an appropriately modified ground-truth are used to train the proposed method as illustrated by the intermediate results in the last row.

and height are in the range 32 to 256 pixels. This augmentation helps significantly as it prevents the network from overfitting the road regions and, consequently, improves the anomaly detection performance as demonstrated in the ablation study (Section 5). The augmentation process is illustrated in Figure 2.

4. Implementation Details

This section describes the technical details of the proposed method with parameter settings relevant for the reproducibility and clarity of method implementation. The following sections describe in detail (*i*) the network architecture with parameters of individual layers, and (*ii*) the training procedure of the proposed method.

4.1. Network Architecture

We based our method on DeepLabV3 [7] network architecture and used publicly available code[2] which we modified for our purpose. The individual parts of the proposed method consist of these blocks:

- ASPP block – use default parameters, as used in the original implementation, for the dilation steps (*i.e.*, [1, 6, 12, 18]). The number of output channels were set to 4 (*i.e.*, the size of the bottleneck in the reconstruction module).
- Decoder – consist of four blocks of UpScale 2x + Conv + Batch normalization + ReLU + Conv + Batch normalization + ReLU with kernel size set to 3 and stride to 1. The number of feature channels is progressively decreased by a factor of two starting from 128 channels (*i.e.*, 128, 64, 32, 16). The reconstruction image is produced by the final 1×1 convolutional layer that that reduces the 16 channels to 3.
- Segmentation coupling – uses two blocks of Conv + Batch normalization + ReLU. The first block takes as the input concatenation of semantic segmentation logits (19 channels for Cityscapes classes) and the SSIM reconstruction error image (1 channel) and reduces the number of channels to 8 with the kernel size set to 3 and stride set to 1. The second block uses 1×1 convolution and outputs two channels, *i.e.*, "road" and "anomalies" classes.

For the SSIM measure, the default values were used. The local window size was set to 11, meaning 11×11 local windows centered around each pixel location are used to compute the average and variance values. The constants c_1, c_2 were set to 0.01^2 and 0.03^2, *i.e.*, the default values when comparing images with pixel values normalized to $(0, 1)$.

4.2. Training

The learning rate for training the semantic segmentation network was set to 0.01 (the default value of the original codebase). For all training of the proposed method, we set the learning rate to 0.001. Since the proposed model is much smaller, it achieves better performance and convergence with a smaller learning rate. Note that the semantic segmentation networks with different backbone variations were trained separately and fixed for all experiments when the proposed method is involved. The slack variable ξ for the $\mathcal{L}_R$ loss was set to 0.001. For training, the input image size was set to 896×896, when possible (limited by the GPU memory), otherwise 513×513 were used (the default value from the original code). The input image in full resolution was used during evaluation. The training was done on a single *NVIDIA RTX 2080 Ti* GPU.

Furthermore, we fixed the random seeds (set to 42) for PyTorch and NumPy libraries to limit the effect of randomness on the ablation studies, *i.e.*, the data augmentation, shuffling as well as network weight initialization were the same. Note that, even though the network weight initialization starts from the same random seed, if the network architecture is changed, some of the parts of the initialization weights change as well since the number of calls to a random number generator are different.

5. Experiments

There are two main experiments – ablation study and comparison to state-of-the-art methods. The proposed method (or its components) was trained using the same parameters, apart from cases where individual components

[2] https://github.com/jfzhang95/pytorch-deeplab-xception

Segm	Recon	Trained	Aux. L	Avg. AP ↑	Avg. FPR$_{95}$ ↓
✓				31.9	71.5
	✓			62.2 (30.3)	19.4 (52.1)
✓	✓		✓	78.9 (16.7)	5.9 (13.5)
✓	✓	✓		79.1 (0.2)	4.9 (1.0)
✓	✓	✓	✓	82.9 (3.8)	5.1 (-0.2)

Table 1. Ablation study: Components. Performance metrics are averaged over all datasets. Numbers in brackets show improvements in percentage points w.r.t. the previous line. *Segm* denotes segmentation network only (using normalized output logits of merged classes for road/sidewalk *vs*. the rest as output), *Recon* uses the reconstruction module only (SSIM reconstruction error is the output), the *Trained* denotes the use of trained segmentation coupling and the *Aux. L* adds loss on SSIM reconstruction. Note that the simple combination *Segm+Recon* (multiplication of the softmax segmentation merged classes and the reconstruction error) explicitly trains the auxiliary reconstruction loss and therefore the *Aux. L* is checked. The Resnet-101 *checkp1* segmentation model was used in this experiment.

were turned on or off. The segmentation model with the Resnet-101 [20] backbone, using the *checkp1* variation (see details in the ablation study section), was used in all experiments unless stated otherwise. The used datasets, performance measures, and the detailed experiments with results discussion are described in the text below.

5.1. Datasets

Three standard datasets are used for the evaluation – Lost-and-Found (LaF) [34], Road Anomaly (RA) [28] and Road Obstacles (RO) [27]. The datasets contain 1203, 60 and 105 test images respectively. The LaF and RO dataset are taken from a camera mounted in the car, whereas RA combines car mounted camera images and artistic pictures of road-like scenarios. Furthermore, Fishyscapes [4] dataset, which is a subset of Lost-and-Found dataset, is used for result compatibility with [27]. We extend the annotation of the Road Anomaly dataset with a coarse road segmentation, to make available the same ground-truth layer which is provided for the other datasets. Two sets of performance measures are adopted from [34, 4, 27], *i.e*., True Positive Rate (TPR) and False Positive Rate (FPR) and Precision and Recall. These measures are summarised by FPR at 95% TPR (FPR$_{95}$) and average precision (AP), respectively. All evaluation is computed using the road region only, similarly to prior work [28, 27, 34].

5.2. Ablation Study

We show two experiments investigating different aspects of the proposed method – contribution of the individual components, backbone architecture, and design choices. It is important to assess the influence of the backbone architecture and the overall performance of the semantic segmentation network because these components are fixed (not trained) and therefore a different backbone could not be able to capture a necessary features or context for the reconstruction module. All variants of the segmentation network (DeepLab V3 [7]) with different backbones were trained on Cityscapes [11] dataset (evaluations of segmentation networks trained on different datasets are left for future work).

Backbone	Val. mIoU	Avg. AP ↑	Avg. FPR$_{95}$ ↓
Mobilenet v2	61.2	78.7	6.8
Xception	50.3	82.1	5.8
Resnet-101 *checkp1*	51.6	82.9	5.1
Resnet-101 *checkp2*	66.1	83.7	4.4

Table 2. Ablation study: Backbone architecture – influence of segmentation backbones on performance measures. Results for the ResNet backbone are included for two checkpoints. The metrics are averaged over all datasets. The differences in performance are mostly marginal (Avg. AP 81 ± 3, Avg. FPR$_{95}$ 5.5 ± 1.3), supporting the claim that the proposed method helps significantly regardless of the backbone architecture and. Furthermore, is shows the ability of the segmentation coupling to exploit efficiently the given segmentation model, regardless of its strength which is expressed as mean IoU on Cityscape validation set (the Val. mIoU column).

Individual Components. The results in Table 1 show the performance of the method's individual components turned on or off. The tested components are: (*i*) the *Segm* – segmentation network only using normalized output logits of merged classes for road/sidewalk *vs*. rest as the output. The class merging use *max* operator over channels for the merging classes. (*ii*) The *Recon* uses the reconstruction module only with the SSIM reconstruction error as its output, (*iii*) the *Trained* denotes the use of trained segmentation coupling, *i.e*., the fusion of the reconstruction error and the semantic segmentation logits is being trained as described in Section 3 and (*iv*) the *Aux. L* is a meta-tag signalizing the fact that there is some part of the training loss directly applied on to the output of the reconstruction module (*e.g*. SSIM). For example, the loss for training the Segm+Recon+Trained (row 4 in Table 1) is the cross-entropy (Eq. 3) applied to the binary classification output of the segmentation coupling layer – no direct loss on the output of reconstruction module. The "all checked" use Eq. 4, where the reconstruction loss (Eq. 2) is added to the cross-entropy, hence, the Aux. L field is checked. Note that the simple combination *Segm+Recon* uses multiplication of the softmax of merged semantic segmentation logits with the reconstruction error for the anomaly class, which explicitly trains the auxiliary reconstruction loss and therefore the *Aux. L* is checked for this combination as well.

The results show clearly the significance of the reconstruction module, that alone has an average performance comparable to *e.g*., Resynthesis [28] method. By adding the trainable coupling to the semantic segmentation, we

Method		Lost and Found		Road Anomaly		Road Obstacles		Fishyscapes:LaF		Average	
		AP ↑	FPR_{95} ↓	AP ↑	FPR_{95} ↓	AP ↑	FPR_{95} ↓	AP ↑	FPR_{95} ↓	$\overline{\text{AP}}$ ↑	$\overline{\text{FPR}_{95}}$ ↓
SDC-Net	softmax	14.0	58.8	54.6	76.8	3.8	99.3	2.6	92.7	18.8	81.9
	logits	28.4	83.9	55.0	85.3	36.8	87.5	24.0	92.0	36.1	87.2
RBM		13.1	85.3	47.7	91.3	17.7	87.2	11.4	85.4	22.5	87.3
Resynthesis [28]	BayseSegNet	61.9	46.6	70.6	65.8	39.3	16.7	63.8	31.7	58.9	40.2
	PSPNet	62.9	43.1	76.4	48.1	59.2	5.5	66.7	② 3.1	66.3	24.9
Outlier Det. [3]*	combined prob.	–	–	–	–	68.1	19.1	68.3	10.7	*68.2*	*14.9*
	random sz. patches	–	–	–	–	70.6	② 1.0	60.9	27.0	*65.8*	*14.0*
	fixed sz. patches	–	–	–	–	31.4	28.1	50.0	73.9	*40.7*	*51.0*
Erase [27]*		–	–	–	–	75.9 ②	15.8	81.0 ②	9.1	*78.5*	*15.5*
JSR-Net (Ours)	Resnet-101 *checkp1*	79.4 ①	① 3.6	92.7 ②	② 12.6	73.9	1.7	85.5 ①	① 2.7	82.9 ②	② 5.1
	Resnet-101 *checkp2*	78.0 ②	② 4.1	94.4 ①	① 9.2	84.0 ①	① 0.4	78.3	4.0	83.7 ①	① 4.4

Table 3. Performance comparison of the proposed and state-of-the-art methods on three standard datasets and on Fishyscapes:LaF, a subset of Lost and Found. The last column block shows the results averaged over all datasets. The best and the second best results are marked by the corresponding badges. Our method (with Resnet-101 *checkp2*) achieved best results on all but one datasets, in both average precision (AP) and in the false positive rate at the operating point of 95% true positive rate (FPR_{95}). In average performance, it is clearly superior to all competitors. The incomplete results of methods marked by * were taken from [27], see text for details.

can reduce the false positives by incorporating knowledge about the known classes. The detection rate can be further increased by 3.8%, by adding the auxiliary loss $\mathcal{L}_R$ to the reconstruction module with negligible increase of false positives.

Different Backbone Architectures. The results in Table 2 show the anomaly detection performance for different segmentation network backbone architectures. Three different types of backbones were used in the DeepLab-v3 [7] segmentation network, namely, Mobilnet-v2 [37], Xception [10] and Resnet-101 [20]. The results in Table 2 demonstrate the effectiveness of our method, regardless of backbone architecture. Moreover, the effect of the semantic segmentation performance was tested on Resnet-101 architecture using two checkpoints with different segmentation performance, 51.6 *vs*. 66.1 mIoU. The semantic segmentation performance was measured on the Cityscapes [11] validation set. The setup of training the Resnet-101 backbone was intentionally changed to produce two different set of weights to show that the proposed method adapts to the different quality of the segmentation and extracted features within the same backbone architecture. Specifically, the size of training images and number of training epochs were lowered.

All tested backbone architectures, when paired with our proposed method, achieved very high anomaly detection scores outperforming all competitors in average performance over all datasets. The differences in performance of the proposed JSR-Net w.r.t. the different backbones are mostly marginal (in the range Avg. AP 81 ± 3 and Avg. FPR_{95} 5.5 ± 1.3), supporting the claim that the proposed method helps significantly regardless of the backbone architecture. Moreover, the segmentation performance seems not to be a crucial factor, but rather the capacity of the backbone since it is fixed and not trained together with the reconstruction module (see Mobilenet *vs*. Resnet-101 *checkp1*). Conversely, a better semantic segmentation performance improves the anomaly detection by only a small margin (resnet-101 *checkp1 vs. checkp2*, which highlights the ability of the segmentation coupling to exploit efficiently the given segmentation model, regardless of its strength, in our case expressed as the mean IoU on Cityscape validation set.

Design choice	Avg. AP ↑	Avg. FPR_{95} ↓
proposed	**82.9**	**5.1**
$\mathcal{L}_R = \text{L2}$	69.5 (-13.4)	10.2 (-5.1)
w/o augmentation	60.9 (-20.0)	11.5 (-6.4)

Table 4. Ablation study: Design choices – Two main design choices are tested: (*i*) reconstruction error measure (SSIM *vs*. L2 distance measure), and (*ii*) proposed anomaly data augmentation. The *w/o augmentation* row is the performance when excluding the novel augmentation strategy (Section 3.3) of random anomalies ”painted” on the road. The performance is averaged over all datasets. The large performance gain supports the choice of SSIM (over the L2 metric) and the validity of the augmentation strategy.

Design Choices. The results in Table 4 validate two important design choices: (*i*) reconstruction error measure (SSIM *vs*. baseline L2 distance measure), and (*ii*) the proposed simple anomaly data augmentation during training. For the evaluation, everything was kept the same except one assessed design choice. Both the use of a more robust reconstruction measure and the anomaly data augmentation strategy improves the anomaly detection performance significantly, thus supporting the validity of our choices.

5.3. State-of-the-Art Comparison

This experiment compares the proposed JSR-Net to recent state-of-the-art methods for road anomaly detection [28, 27, 12], out-of-distribution detection method [3]

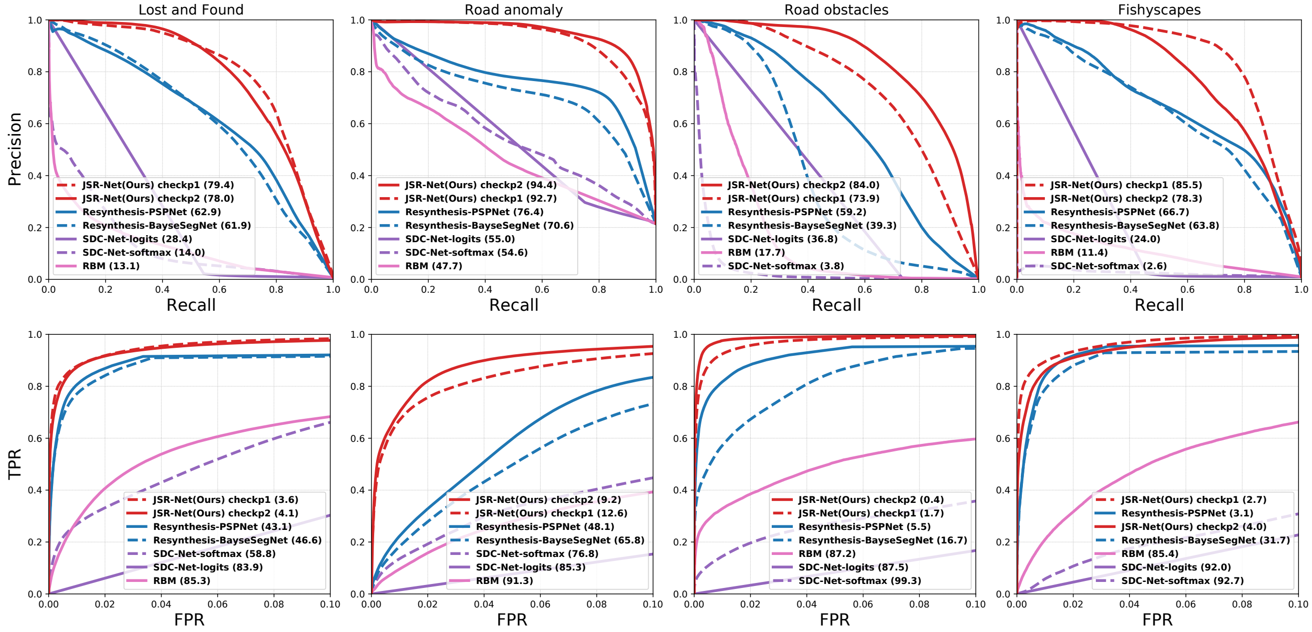

Figure 3. Performance comparison on all datasets in the form of Precision-Recall (top row) and ROC (bottom row) curves. In the legend, the numbers next to each method show the average precision (AP) for the Precision-Recall curves and FPR_{95} for the ROC curves. For the ROC curve we show only the most relevant part – up to 10% FPR – since methods with higher FPR are not usable in real-world application.

(for more details about its variations, readers are referred to [4]) and the baseline state-of-the-art segmentation method [53]. The authors implementation was used for the methods [28, 53] with the provided pre-trained models. The implementation of [12] and the pre-trained model was published in the codebase of [28].

There were several issues with obtaining results for the methods in [27, 46, 3]. The code for the [27] was not published at the time of publication. The method SynthCP [46] was trained only on synthetic data and we did not manage to get reasonable results (higher than SDC-Net baseline) when trained on real data (Cityscapes, Lost and Found). A similar problem with training was encountered with the outlier detection method [3]. For those reasons, we used results on the overlapping datasets from [27], to at least provide partial comparisons. Note that the datasets used in [27] are a subset of datasets used in our evaluation.

The results are summarized in Table 3 and Figure 3, which show the Precision-Recall and ROC curves. The proposed method outperforms the state-of-the-art and significantly improves anomaly detection performance, especially in reducing the false positive rates across a wide range of operation points. Note that our method performs consistently across multiple different datasets as opposed to *e.g.*, Resynthesis [28] method, which achieves very low FPR_{95} ($< 5.6\%$) on two datasets but for other two it is almost ten times larger ($> 43.0\%$).

6. Conclusions

In this paper, we propose a novel method, JSR-Net, for detecting unknown "stuff" (*i.e.*, anomalies) on the road and demonstrate its effectiveness in the context of autonomous driving applications. We formulated the problem as anomaly detection, since the unknown object's appearance cannot be learned directly. To that end, we proposed a reconstruction module that can be used with many existing semantic segmentation networks. The reconstruction module is trained to recognize and reconstruct road surfaces and its inability to reconstruct a part of the road is used as an indicator of an anomaly. The reconstruction error is coupled via a trainable coupling block with the semantic segmentation to incorporate the information from known classes and to produce final per-pixel anomaly scores.

We evaluate our method on three standard datasets and one derivative – Lost and Found, Road Anomaly, Road Obstacles, and Fishyscapes:LaF – demonstrating significant improvements in anomaly detection compared to other state-of-the-art methods on all datasets, except one, where it performed comparatively.

Despite excellent results and the ability to detected anomalies of diverse sizes (e.g. from small bottles to a truck tire), the method still produces false positives, especially on thin structures e.g. long cracks or lane markings. We also observed performance deterioration for lower image quality, e.g. due to strong JPEG artefacts or acquisition trough dirty windows (see the supplementary material).

References

[1] Jinwon An and Sungzoon Cho. Variational autoencoder based anomaly detection using reconstruction probability. *Special Lecture on IE*, 2(1):1–18, 2015.

[2] Yoshua Bengio. Deep learning of representations for unsupervised and transfer learning. In *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, volume 27, pages 17–36, 02 Jul 2012.

[3] Petra Bevandić, Ivan Krešo, Marin Oršić, and Siniša Šegvić. Simultaneous Semantic Segmentation and Outlier Detection in Presence of Domain Shift. In *Pattern Recognition*, pages 33–47. Springer International Publishing, 2019.

[4] H. Blum, P. Sarlin, J. Nieto, R. Siegwart, and C. Cadena. Fishyscapes: A Benchmark for Safe Semantic Segmentation in Autonomous Driving. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 2403–2412, 2019.

[5] Paul Bodesheim, Alexander Freytag, Erik Rodner, and Joachim Denzler. Local novelty detection in multi-class recognition problems. In *2015 IEEE Winter Conference on Applications of Computer Vision*, pages 813–820. IEEE, 2015.

[6] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3), July 2009.

[7] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(4):834–848, 2018.

[8] Qifeng Chen and Vladlen Koltun. Photographic Image Synthesis With Cascaded Refinement Networks. In *Int. Conf. Comput. Vis.*, Oct 2017.

[9] X. Cheng, P. Wang, and R. Yang. Learning Depth with Convolutional Spatial Propagation Network. *IEEE Trans. Pattern Anal. Mach. Intell.*, pages 1–1, 2019.

[10] F. Chollet. Xception: Deep learning with depthwise separable convolutions. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1800–1807, 2017.

[11] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The Cityscapes Dataset for Semantic Urban Scene Understanding. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016.

[12] C. Creusot and A. Munawar. Real-time small obstacle detection on highways using compressive RBM road reconstruction. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 162–167, 2015.

[13] A. Criminisi and J. Shotton. *Decision Forests: A Unified Framework for Classification, Regression, Density Estimation, Manifold, Learning and Semi-Supervised Learning*. Foundations and Trends in Computer Graphics and Vision. Now Pub, 2012.

[14] A. Dairi, F. Harrou, M. Senouci, and Y. Sun. Unsupervised obstacle detection in driving environments using deep-learning-based stereovision. *Robotics and Autonomous Systems*, 100:287 – 301, 2018.

[15] P. Dollár and C. L. Zitnick. Structured forests for fast edge detection. In *International Conference on Computer Vision*, pages 1841–1848, 2013.

[16] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16, 2017.

[17] Jannik Fritsch, Tobias Kuehnl, and Andreas Geiger. A New Performance Measure and Evaluation Benchmark for Road Detection Algorithms. In *International Conference on Intelligent Transportation Systems (ITSC)*, 2013.

[18] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig. VirtualWorlds as Proxy for Multi-object Tracking Analysis. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4340–4349, 2016.

[19] K. Gupta, S. A. Javed, V. Gandhi, and K. M. Krishna. MergeNet: A Deep Net Architecture for Small Obstacle Discovery. In *International Conference on Robotics and Automation (ICRA)*, pages 5856–5862, 2018.

[20] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 770–778, 2016.

[21] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.

[22] Yen-Chang Hsu, Yilin Shen, Hongxia Jin, and Zsolt Kira. Generalized odin: Detecting out-of-distribution image without learning from out-of-distribution data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10951–10960, 2020.

[23] Kaijin Ji and Huiyan Chen , Huijun Di , Jianwei Gong , Guangming Xiong , Jianyong Qi , Tao Yi. CPFG-SLAM:a robust Simultaneous Localization and Mapping based on LIDAR in off-road environment. In *IEEE Intelligent Vehicles Symposium (IV)*, 2018.

[24] R. Labayrade, D. Aubert, and J. . Tarel. Real time obstacle detection in stereovision on non flat road geometry through "v-disparity" representation. In *Intelligent Vehicle Symposium*, volume 2, pages 646–651 vol.2, 2002.

[25] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems*, pages 7167–7177, 2018.

[26] Shiyu Liang, Yixuan Li, and R. Srikant. Enhancing the reli of out-of-distribution image detection in neural networks. In *International Conference on Learning Representations*, 2018.

[27] Krzysztof Lis, Sina Honari, Pascal Fua, and Mathieu Salzmann. Detecting Road Obstacles by Erasing Them, 2020.

[28] K. Lis, K. Nakka, P. Fua, and M. Salzmann. Detecting the Unexpected via Image Resynthesis. In *Int. Conf. Comput. Vis.*, October 2019.

[29] J. Ma, A. Ming, Z. Huang, X. Wang, and Y. Zhou. Object-Level Proposals. In *Int. Conf. Comput. Vis.*, pages 4931–4939, 2017.

[30] A. Ming, T. Wu, J. Ma, F. Sun, and Y. Zhou. Monocular Depth-Ordering Reasoning with Occlusion Edge Detection and Couple Layers Inference. *Intelligent Systems*, 31(2):54–65, 2016.

[31] D. Mykheievskyi, D. Borysenko, and V. Porokhonskyy. Learning Local Feature Descriptors for Multiple Object Tracking. In *ACCV*, 2020.

[32] Toshiaki Ohgushi, Kenji Horiguchi, and Masao Yamanaka. Road Obstacle Detection Method Based on an Autoencoder with Semantic Segmentation. In *ACCV*, November 2020.

[33] S. Pang, D. Morris, and H. Radha. CLOCs: Camera-LiDAR Object Candidates Fusion for 3D Object Detection. In *International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[34] P. Pinggera, S. Ramos, S. Gehrig, U. Franke, C. Rother, and R. Mester. Lost and Found: detecting small road hazards for self-driving vehicles. In *International Conference on Intelligent Robots and Systems (IROS)*, 2016.

[35] S. Ramos, S. Gehrig, P. Pinggera, U. Franke, and C. Rother. Detecting unexpected obstacles for self-driving cars: Fusing deep learning and geometric modeling. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 1025–1032, 2017.

[36] Richter, Stephan R. and Vineet, Vibhav and Roth, Stefan and Koltun, Vladlen. Playing for Data: Ground Truth from Computer Games. In *Eur. Conf. Comput. Vis.*, pages 102–118, 2016.

[37] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. Chen. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4510–4520, 2018.

[38] Alexander Schultheiss, Christoph Käding, Alexander Freytag, and Joachim Denzler. Finding the unknown: Novelty detection with extreme value signatures of deep neural activations. In *German Conference on Pattern Recognition*, pages 226–238. Springer, 2017.

[39] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[40] N. Soquet, D. Aubert, and N. Hautiere. Road segmentation supervised by an extended v-disparity algorithm for autonomous navigation. In *Intelligent Vehicles Symposium*, pages 160–165, 2007.

[41] L. Sun, K. Yang, X. Hu, W. Hu, and K. Wang. Real-Time Fusion Network for RGB-D Semantic Segmentation Incorporating Unexpected Obstacle Detection for Road-Driving Images. *Robotics and Automation Letters*, 5(4):5558–5565, 2020.

[42] David MJ Tax and Robert PW Duin. Uniform object generation for optimizing one-class classifiers. *Journal of machine learning research*, 2(Dec):155–173, 2001.

[43] T. Wang, M. Liu, J. Zhu, A. Tao, J. Kautz, and B. Catanzaro. High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 8798–8807, 2018.

[44] Z. Wang, E. P. Simoncelli, and A. C. Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems Computers, 2003*, volume 2, pages 1398–1402 Vol.2, 2003.

[45] A. Wedel, H. Badino, C. Rabe, H. Loose, U. Franke, and D. Cremers. B-Spline Modeling of Road Surfaces With an Application to Free-Space Estimation. *IEEE Trans. on Intelligent Transportation Systems*, 10(4):572–583, 2009.

[46] Yingda Xia, Yi Zhang, Fengze Liu, Wei Shen, and Alan L. Yuille. Synthesize Then Compare: Detecting Failures and Anomalies for Semantic Segmentation. In *Eur. Conf. Comput. Vis.*, pages 145–161, 2020.

[47] F. Xue, A. Ming, M. Zhou, and Y. Zhou. A Novel Multi-layer Framework for Tiny Obstacle Discovery. In *International Conference on Robotics and Automation (ICRA)*, pages 2939–2945, 2019.

[48] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning, 2020.

[49] F. Zhang, V. Prisacariu, R. Yang, and P. Torr. Ga-net: Guided aggregation net for end-to-end stereo matching. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.

[50] J. Zhang and S. Singh. Visual-lidar Odometry and Mapping: Low drift, Robust, and Fast. In *International Conference on Robotics and Automation (ICRA)*, Seattle, WA, May 2015.

[51] S. Zhao, Y. Sheng, Y. Dong, E. I-C. Chang, and Y. Xu. Mask-Flownet: Asymmetric Feature Matching with Learnable Occlusion Mask. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020.

[52] X. Zhou, V. Koltun, and P. Krähenbühl. Tracking Objects as Points. *Eur. Conf. Comput. Vis.*, 2020.

[53] Y. Zhu, K. Sapra, F. A. Reda, K. J. Shih, S. Newsam, A. Tao, and B. Catanzaro. Improving Semantic Segmentation via Video Propagation and Label Relaxation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, June 2019.