Image-Consistent Detection of Road Anomalies as Unpredictable Patches

Tomáš Vojíř and Jiří Matas
Czech Technical University in Prague, FEE
Technická 2, Prague, Czech Republic
{vojirtom,matas}@fel.cvut.cz

Abstract

*We propose a novel method for anomaly detection primarily aiming at autonomous driving. The design of the method, called DaCUP (**D**etection of **a**nomalies as **C**onsistent **U**npredictable **P**atches), is based on two general properties of anomalous objects: an anomaly is (i) not from a class that could be modelled and (ii) it is not similar (in appearance) to non-anomalous objects in the image. To this end, we propose a novel embedding bottleneck in an auto-encoder like architecture that enables modelling of a diverse, multi-modal known class appearance (*e.g. *road). Secondly, we introduce novel image-conditioned distance features that allow known class identification in a nearest-neighbour manner on-the-fly, greatly increasing its ability to distinguish true and false positives. Lastly, an inpainting module is utilized to model the uniqueness of detected anomalies and significantly reduce false positives by filtering regions that are similar, thus reconstructable from their neighbourhood. We demonstrate that filtering of regions based on their similarity to neighbour regions, using* e.g. *an inpainting module, is general and can be used with other methods for reduction of false positives. The proposed method is evaluated on several publicly available datasets for road anomaly detection and on a maritime benchmark for obstacle avoidance. The method achieves state-of-the-art performance in both tasks with the same hyper-parameters with no domain specific design.*

1. Introduction

The anomaly detection problem, sometimes also referred to as outlier, out-of-distribution or rare event detection, can be intuitively understood as the task of identifying a subset of data that deviates from a statistical model or the general data distribution, in a novel, so far unseen, and thus difficult to predict way. Anomaly detection is an ill-defined problem; there is no generally accepted definition, and most approaches are application driven. For example, in industrial object inspections [2], anomalies are defined as local physical defects or deformities of objects, in medicine [22] anomalies may refer to subtle abnormalities in brain tissue, in streaming data analysis [23, 1] the anomalies are ".... patterns that do not conform to past patterns of behavior for the stream" [23]. In autonomous driving [36, 7] anomalies are most commonly defined as objects on the road that are not from "known classes" (*e.g.* cars, pedestrians, ...).

In the paper, we focus on the latter, *i.e.* anomaly detection in the context of autonomous driving (but are not limited to, as demonstrated in Section 4.3). We propose a novel anomaly detection method that is based on general properties of anomalous objects: (1) *not known*, not from a class that we could model (*i.e.* a labeled class available during training) and (2) *unique*, not similar to non-anomalous objects in the image. The first property follows the common perception [36, 7] of what anomaly is while the second indirectly enforces consistency in anomalous object classification (*e.g.* rocks on the road or lane markings are either all anomalous or all known, but not both at the same time).

Standard methods for anomaly detection [26, 40, 30, 29, 16] rely on pre-trained representations, typically focusing on modelling the non-anomalous classes for which training data are abundant. Naturally, there is a scarcity of diverse anomalous objects in training data as opposed to all novel, so far unseen, unexpected objects that can appear in the real-world. Therefore, training data are usually augmented by synthesized anomalies from random distributions, *e.g.* [8] uses data from the COCO [28] dataset as anomalies, [40] uses random crops from the same image; [29] uses the collection of objects from non-road classes from the training dataset. These pre-trained models are thus limited by the training data with only synthesized anomalies from random distributions and they lack mechanisms to model the aspect of anomaly uniqueness and the consistency in classification which hinders their performance, especially in terms of false positives (as shown experimentally in Table 3). One could even argue that the concept of training for anomalies is self-contradictory, since once something novel is observed in training, it can be modelled and ceases to be an anomaly.

We propose a novel method, called DaCUP that addresses both aspects of the aforementioned definition, i.e. the method is trained to model the non-anomalous class (road) in an explicit embedding space with augmented training data by synthesized anomalies [40] together with mechanisms to promote classification consistency and model the uniqueness of anomalies. To that end, a novel distance-based similarity and inpainting features conditioned on the input image are proposed.

The three main contributions of the paper, one related to the design of the model architecture for robust modelling of the known class (*e.g. road class*), and two tackling the anomaly uniqueness and consistency of classification, are:

- The DaCUP method that models multi-modal appearance of road surfaces by explicitly learning a feature embedding space. The multi-modal aware embedding space allows the method to be trained using multiple diverse datasets (showcased on combining Cityscapes [11] and BDD100k [46]) to learn diverse road surface appearances (*i.e.* know class) which are then utilized to separate the anomalies from the road.

- A novel image-conditioned distance-to-class score. The distance score is computed between embeddings of all image patches and the mean embedding patches segmented as the known class (*e.g.* road) in the current image. The score, used as an additional feature, increases the ability to identify anomalies on never seen surfaces and to decrease false positive detections (demonstrated on results shown in Table 2).

- A novel use of an inpainting mechanism from intermediate features that reduces anomaly false positives, implementing the principle that *an anomaly cannot be predicted from its neighbourhood*. The algorithm is applicable to any method that produces per-pixel anomaly scores. We show its application improves performance of several baseline methods (see Table 3).

2. Related Work

In this section, related work on anomaly detection with a focus on autonomous driving application is discussed. State-of-the-art methods can be broadly split into two main trends: (i) detecting anomalies as an *out-of-distribution* data identification in semantic segmentation networks [26, 8, 16, 24], (ii) detecting discrepancies between re-synthesized and input images [30, 44, 34, 13].

The methods that address the problem of identifying *out-of-distribution* data usually start with a semantic segmentation network with a temperature calibrated [17] softmax output. The method of Liang *et al* [26] combines a temperature calibrated [17] softmax with adversarial perturbations of the input image. The perturbation of the input image has a stronger effect on the *in-distribution* than on *out-of-distribution* data, therefore the separation is more pronounced and a simple thresholding strategy is applied to classify the *out-of-distribution* data. Similarly, Lee *et al* [24] added noise perturbation to the input image and defined a Mahalanobis distance-based score that identifies the *out-of-distribution* pixels by comparing the pixel features (from different network layers) with a class-conditional Gaussian distribution of the closest class estimated from the training data. Instead of adversarial input perturbations, Chan *et al* [8] considers the *out-of-distribution* samples as random objects taken from COCO [28] images and proposes to maximize the softmax entropy for the *out-of-distribution* pixels, hence, producing a uniform distribution over the classes softmax probabilities. The *out-of-distribution* pixels are classified by thresholding normalized softmax entropy. The method of Grcic *et al* [16] trains semantic segmentation network with mixed-content images. The input image is overlaid by normalized flow sampled synthetic negative examples creating *out-of-distribution* pixels. The network is then trained to have low classification error on *in-distribution* pixels and to raise uniform predictions on the *out-of-distribution* pixels. The *out-of-distribution* pixels are identified during inference by computing temperature calibrated softmax and JS divergence with respect to the uniform distribution.

There are multiple recent methods [30, 44, 34, 13] that exploit the success of generative networks [10, 41, 31] to synthesize images from semantic segmentation labels. The method of Lis *et al* [30] first re-synthesize image using generative method [41] from the estimated semantic segmentation of the input image. A discrepancy network is then trained to identify visual and semantic differences between the VGG [38] features of the synthesize and input images and semantic segmentation. Similarly, Xia *et al* [44] synthesize an image from estimated semantic segmentation and then train siamese-like architecture to detect differences between the synthesized and input images. Ohgushi *et al* [34] combines the perceptual difference [21, 15] computed from VGG features extracted from the synthesized image and input images with the entropy of semantic segmentation softmax posteriors to produce the final anomaly score map. Di Biase *et al* [13] effectively combined previous approaches [30, 34] and start by synthesizing image based on estimated semantic segmentation followed by a perceptual difference using VGG features. The intermediate results (*i.e.* input and synthesize image, semantic segmentation, perceptual difference and softmax entropy) are fused together and pass through a decoder that estimates the final per-pixel anomaly score.

An approach proposed by Creusot *et al* [12] trains an auto-encoding Restricted Boltzmann Machine [39] on image patches of highways to learn a low-dimensional representation of highway appearances instead of synthesizing the image (or its patches) from semantic segmentation. During the evaluation, the input image is split into small patches and each patch is passed through the RBM. The absolute difference between the original and the RBM reconstructed image patch is used as an indicator of the presence of an anomaly. Munawar *et al* [33] also models the road patches in a auto-encoder manner and takes into account the appearance of memorized input patches from previous video frames. The absolute difference between the input and reconstructed patches is computed to identify anomalies.

In contrast to the previous two methods, Vojir *et al* [40] takes a holistic approach and instead of reconstructing small image patches use an auto-encoder-like network to learn to reconstruct, in a discriminative way, RGB values of the road pixels. The error between the input and the reconstructed images is computed by structural similarity measure [42]. The reconstruction error is merged with the semantic segmentation logits to produce the final per-pixel anomaly score. The major shortcoming of these auto-encoder methods is the simplicity of the bottleneck to capture the road appearance which results in an inability to utilize the training data efficiently. This is most prominent during training where the road appearance is multi-modal (*e.g*. asphalt, gravel or cobblestones roads). The method proposed in this paper also utilizes an auto-encoder architecture, however, among other things we address this flaw by a proposed novel explicit embedding bottleneck and demonstrate its efficiency in the experimental section.

Instead of explicitly synthesizing images from semantic segmentation labels or learning compact representation of road appearance for image reconstruction, the method proposed by Lis *et al* [29] use semantic segmentation to estimate the road region and inpaint it in a sliding window fashion using general-purpose inpainting algorithm [47]. A discrepancy network utilizing VGG features is then trained to compare the input and inpainted images and detect anomalous regions. In contrast, we propose to use the inpainting in a later stage in targeted areas and pair it with a "fixed" similarity metric to identify discrepancies w.r.t. the input image. The use of a fixed similarity metric in contrast to trained discrepancy network limits the effect and biases of the training data that relies on synthetic anomalies that may or may not generalize for real-world images.

3. Method

This section describes the four main components of the proposed method: the baseline part [40], the embedding bottleneck, the distance-based scoring feature and the inpainting module. The overall structure is shown in Figure 1.

From the baseline, we adopted the idea of a generative-discriminative reconstruction module together with the fixed semantic segmentation network. The reconstruction module can be efficiently trained with only road annotated data and simple synthetic anomaly augmentation – it has shown excellent performance on diverse datasets. Furthermore, the use of a fixed semantic segmentation network allows easy integration into a real-world system that utilizes some form of semantic segmentation in production without the need of re-training the semantic segmentation that could potentially degrade its performance.

However, we identified a major shortcoming of this method – using a small bottleneck to capture the road appearance is not able to exploit the training data efficiently. This is most prominent during training where road appearance is multi-modal (*e.g*. asphalt, gravel or cobblestones roads), which prevents training on larger, and more diverse datasets. Among other things, we address this flaw by the proposed explicit embedding bottleneck.

The proposed network architecture, see Fig. 1, consists of a fixed semantic segmentation network (we adopted DeepLabV3 [9]), the embedding bottleneck, the reconstruction module (adopted from [40]), the embedding scoring feature estimation block, inpainting and coupling modules. The reconstruction module takes features $f \in \mathbb{R}^{C,H_f,W_f}$ from the semantic segmentation backbone as an input and is trained to reconstruct the original pixel values of the drivable surface from the embedding bottleneck (Section 3.1), while inducing large reconstruction error elsewhere. For a detailed explanation of the reconstruction module, the readers are referred to [40].

The coupling module fuses logits $l \in \mathbb{R}^{K,H,W}$ from the semantic segmentation, the image reconstruction error $r_{\text{err}} \in \mathbb{R}^{H,W}$, the embedding scoring feature f^{ds} and f^{dr} (Section 3.2) and the inpainting error $\mathcal{L}_{\text{vgg}}$ (Section 3.3) to produce the per-pixel anomaly score map $s_{\text{map}} \in \mathbb{R}^{H,W}$.

The rest of the section describes the proposed modules in detail. Section 3.1 presents the design to handle the multi-modal appearance of road surfaces. The embedding space enables the use of novel input image conditioned embedding scoring feature (Sec. 3.2), allowing identification of "miss-classified" road regions on-the-fly and to alleviate the issue with previously unseen road appearance. Lastly, a new inpainting technique is introduced in Section 3.3, addressing the issue with consistent classification of anomalies.

3.1. Embedding Bottleneck

To address the data utilization problem of learning multi-modal road appearance in the bottleneck, we propose to inject a small embedding network that takes the backbone features and transform them into a embedding vectors. For each spatial location $f_{x,y} \in \mathbb{R}^C$, an embedding vector $e_{x,y} \in \mathbb{R}^D$ is computed. To that end, an atrous spatial pyramid pooling (ASPP) [9] extracts contextual features for location (x, y). The ASPP block is modified to output the concatenation of the features extracted with different dila-

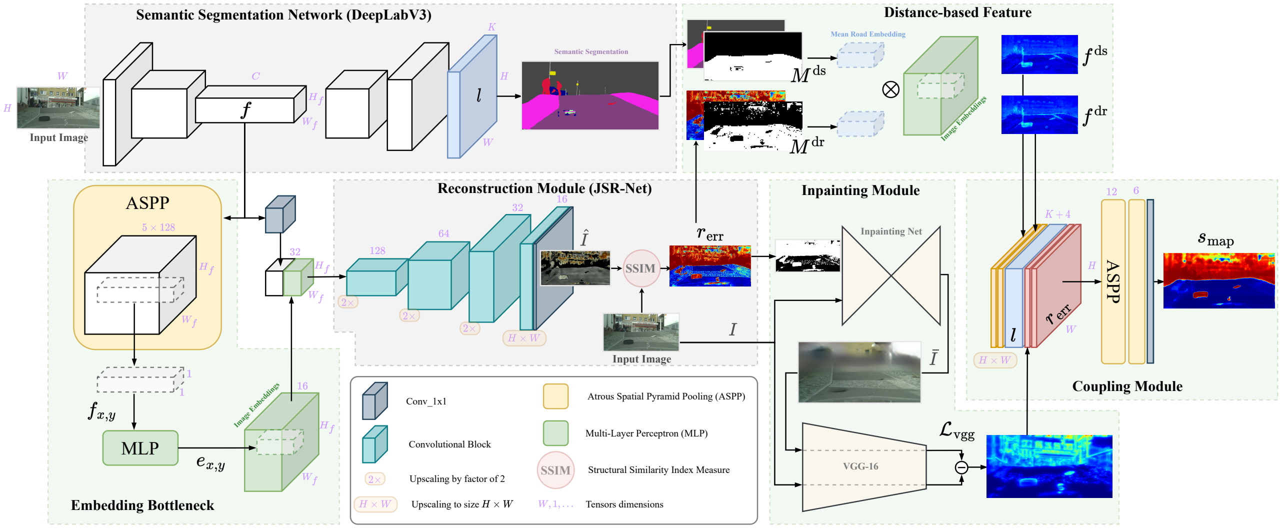

Figure 1. Method overview and its six main blocks. The input image is processed by the *semantic segmentation network* [9] with fixed weights. The intermediate image representation f is processed by the *embedding bottleneck* and also used for the skip connection to the *reconstruction module* [40]. The reconstructed image is compared to the input image using the SSIM to produce the reconstruction error r_{err}. The reconstruction error is used in the *inpainting module* to obtain the inpainting mask (by thresholding) and the VGG features of the inpainted and input images are compared to obtain the perceptual loss map $\mathcal{L}_{\text{vgg}}$. Moreover, the reconstruction error and the semantic segmentation are utilize in the *distance-based features block* to extract the road region masks M^{dr} and M^{ds}. These masks enable to compute the mean road embeddings and the distance-based scoring features f^{dr} and f^{ds}. The aforementioned intermediate results are concatenated in the *coupling module* where the final anomaly score map s_{map} is estimated.

tion (*i.e*. by removing the last 1×1 conv. layer + Batch Normalization + ReLU). By this use of the ASPP, backbone features f with higher resolution can be utilized without losing the additional contextual information from larger receptive fields in later stages. Note that the baseline [40] uses the DeepLab v3 [8] architecture for the segmentation network with the ResNet-101 [19] backbone with features extracted before the last downsampling, *i.e*. at $1/16$ input image scale instead of $1/32$. The ASPP block is followed by three FC layers with ReLU activations to obtain the final D-dimensional embedding $e_{x,y}$.

The major difference, in addition to the modification of the bottleneck, is the explicit loss set on the embeddings. A max-margin triplet loss [37] is used to enforce the margin m distance between embeddings of anchor sample (e^a) and negative (e^n – else) sample while minimizing distance to positive (e^p – road) samples as follows:

$$\mathcal{L}_{\text{tri}} = \frac{1}{|\mathcal{T}|} \sum_{(a,p,n)\in\mathcal{T}} \max\left(||e^a - e^p|| - ||e^a - e^n|| + m, 0\right) + \lambda_d \frac{(e^n - e^a)}{||e^n - e^a||} \cdot \frac{(e^p - e^a)}{||e^p - e^a||}, \tag{1}$$

where $\mathcal{T}$ is the set of triplets and $|\mathcal{T}|$ its size. The last term (the dot product of two normalized vectors) is the directional regularization [32] with λ_d as the weighting factor. Depending on the size of the input image, there can be an excessive number of triplets, *e.g*. if the all mining strategy [14] is used. For this reason, a stochastic sampling procedure is used to randomly choose a limited number of anchor samples (this number can be adjusted based on the available GPU memory, in all our experiments we set it to 1024). To select the positive and negative samples, the ground-truth labels need to be resized to the resolution of the features f. Instead of simple resizing using nearest-neighbour interpolation, a majority vote strategy is employed to address the corner cases of areas where multiple labels meet. The triplets for the training are formed using the semi-hard negative mining [37] strategy to select the negative samples and the easy positive sampling strategy (ESP) [25] for positive samples. These two sampling strategies and the regularization [32] help with a training convergence and, in the case of the ESP, to prevent a class collapse and allow to represent multi-modal distribution of the road class embeddings.

3.2. Distance-based Scoring Feature

The embedding bottleneck is further exploited for the purpose of extracting on-line appearance distance-like scores. This additional feature addresses the problem of increased false positive classifications induced by unknown (*i.e*. not seen during training) road appearances by computing road similarity score w.r.t. the observed image utilizing the structured embedding space. The design is based on a hypothesis that a mean representation of the observed road in the embedding space (even though computed from a noisy and incomplete list of all correct road embeddings in the observed image) would yield a small distance to all correct road embeddings, and therefore, would be able to identify them. To utilize this assumption, a new feature channel $f^d \in \mathbb{R}^{H_f, W_f}$ is introduced. It is computed location-wise, where for each spatial location (y, x) is computed as:

$$\begin{aligned} f^d_{x,y} =& ||\bar{e} - e_{x,y}|| \\ \bar{e} =& \frac{1}{\sum_{i,j} M_{i,j}} \sum_{(i,j) \text{ s.t. } M_{i,j}=1} e_{i,j}, \end{aligned} \qquad (2)$$

where $\bar{e}$ is the mean embedding computed from embeddings with estimated road label indicated by 1 in the mask M.

In the current implementation, two feature channels – f^{ds} and f^{dr} – are used, computed according to Eq. 2. The masks, required for computation of the features, are obtained from i) the semantic segmentation logits l to estimate the mask M^{ds} and the f^{ds} and ii) from the thresholded reconstruction error r_{err} to estimate the mask M^{dr} and the f^{dr}. Note that other source of the mask can be introduced to increase robustness to the failure modes of the respective sources of the road labellings. The effectiveness of this feature is evaluated in the ablation study (Section 4.1).

3.3. Inpainting Module

To address the issue of consistent anomaly classification, we propose to use an inpainting technique. It is motivated by the observation that anomalies are not similar to anything in the image (except themself, *e.g.* fallen rocks on the road) and therefore the inpainting should not be able to recreate the anomaly from its neighbourhood. The inpainting module, in our case, serves two tasks: (i) identifying false positives (a prominent example is the lane markings where part of them are classified as anomalies due to shadows or other appearance degradations) and (ii) refinement of the anomaly segmentation boundary.

For the inpainting technique, the DeepFillv2 [47] method was adopted. This holistic method allows for inpainting of areas defined by a free-form mask (*e.g.* random areas in the image), which is well suited for our application where the inpainting mask comes from the intermediate results obtained from thresholding the reconstruction error r_{err} by learnable threshold and applying a 7 × 7 morphological dilation. The error metric between the input image I and the inpainted version $\bar{I}$ is a perceptual loss [21, 15], which computes the mean Manhattan distance between features extracted from the VGG [38] network at different convolutional layers. We intentionally use fixed VGG features to compute the perceptual loss, instead of *e.g.* training special discrepancy network [30, 29], to limit the source of overfitting and reliance on quality of training data, since the anomalies in the training data are obtained synthetically through augmentation and their quality and variance is low. The inpainting perceptual difference map $\mathcal{L}_{\text{vgg}}$ is a weighted average of the channel-wise mean distance with sigmoid non-linearity $\sigma(\cdot)$ to normalize the error to (0,1) range:

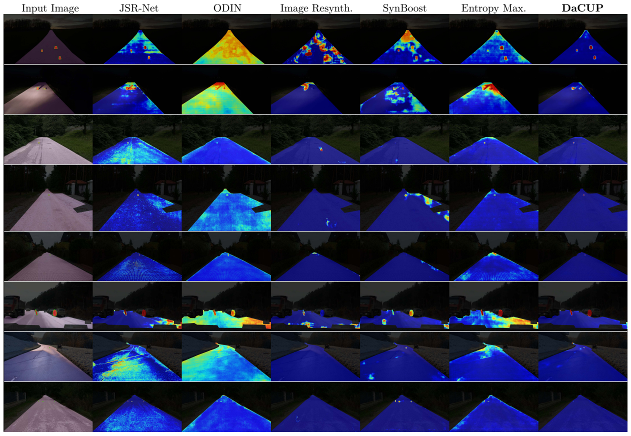

Figure 2. Qualitative results on the *SMIYC* benchmark. Best viewed in color and zoomed-in. The DaCUP shows excellent performance on various surfaces with the ability to avoid false positives induced by road surface texture. The main weakness is the performance for small and distant objects. Ground-truth anomalies are marked red with a green border (left column). Qualitative results for all datasets are included in supplementary materials.

$$\mathcal{L}_{\text{vgg}} = \sigma\left(\sum_{i=1}^{n} U_{\text{bl}}\left(\frac{w_i}{C_i}\sum_{c=1}^{C_i} ||\phi^{(i)}(I) - \phi^{(i)}(\bar{I})||_1\right)\right),$$

where C_i is the number of channels of the i-th layer and function $U_{\text{bl}}(\cdot)$ is bilinear interpolation to resize the distance maps to common size (*i.e.* the size of the input image). Weights $\mathbf{w}_{\text{vgg}} = (w_0, w_0, \ldots, w_n)$, for each used convolutional layer the VGG network, are learned during training.

The inpainting perceptual loss is concatenated in the coupling module with the other features, namely, with logits l from the semantic segmentation network, and with reconstruction error r_{err} from the reconstruction module and the two distance-based embedding score feature, f^{ds} and f^{dr}. Similarly to [40], the coupling module fuses the feature channels and produces the final binary classification road *vs.* anomaly.

We observed that it is beneficial to inject augmentation to the inpainting mask for the coupling module to learn to recognize inpainting of road and other classes which may not be anomalies since it not always results in low perceptual loss and depends on the image context. For that reason, a random free form mask [47] is added to the estimated inpainting mask during training.

3.4. Training Details

The training starts with a warm-up procedure; for the first five epochs only the triplet loss $\mathcal{L}_{\text{final}} = \mathcal{L}_{\text{tri}}$ (Eq. 1) is used. For epochs six to ten, we switch to $\mathcal{L}_{\text{final}} = \mathcal{L}_{\text{tri}} + \mathcal{L}_R$. The auxiliary reconstruction loss, $\mathcal{L}_R$, and computed as:

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

where M is a ground-truth binary mask for the road and not-road (anomalies) pixels respectively and $|M|$ denotes the number of non-zero elements of the mask. The slack variable, ξ, was set to 0.001 as in [40]. The SSIM is the structural similarity index measure [43]. For a pixel at location (x, y) of a single channel image, SSIM is computed as:

$$\text{SSIM}(u_{\hat{I}}^{x,y}, v_I^{x,y}) = \frac{(2\mu_u\mu_v + c_1)(2\sigma_{uv} + c_2)}{(\mu_u^2 + \mu_v^2 + c_1)(\sigma_u^2 + \sigma_v^2 + c_2)} \quad (3)$$

where $u_{\hat{I}}^{x,y}, v_I^{x,y}$ are local patches of reconstructed image $\hat{I}$ and input image I centered at (x, y). The μ, σ are the mean and variance of pixel values in these patches. The constants c_1, c_2 are used to normalize SSIM to the $(0, 1)$ range and are set to the default values 0.01^2 and 0.03^2. Eq. 3 is evaluated for each image channel (R,G,B) separately and averaged to produce the final reconstruction error r_{err}.

After ten epochs, the final loss $\mathcal{L}_{\text{final}} = \lambda_{\text{xent}}\mathcal{L}_{\text{xent}} + \lambda_{\text{tri}}\mathcal{L}_{\text{tri}} + \lambda_R\mathcal{L}_R$ is used. The weights are set to $\lambda_{\text{xent}} = 0.6, \ \ \lambda_{\text{tri}} = 0.2, \ \ \lambda_R = 0.2$. The $\mathcal{L}_{\text{xent}}$ is the negative log likelihood loss applied to the binary classification output of the coupling module as:

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

where N is number of pixels, the c^n and $\hat{c}^n$ are the ground-truth and estimated labels for the n^{th} pixel respectively.

The triplet loss regularization weight λ_d is set to 0.2 per recommendation in [32]. The initial learning rate was set to 0.001 and polynomial learning rate decay with power of 0.9 is used. For optimization, a stochastic gradient descent with momentum (0.9) and weight decay (5e-4) is used.

The perceptual loss is computed using three intermediate layers of the VGG network – *conv1_2*, *conv2_2* and *conv3_3*, and the initial weights $\mathbf{w}_{\text{vgg}}$ are set equally to 0.33.

	Training Data	*LaFRAROFS*		*Obstacle Track+*	
		$\overline{\text{AP}}\uparrow$	$\overline{\text{FPR}}_{95}\downarrow$	$\overline{\text{AP}}\uparrow$	$\overline{\text{FPR}}_{95}\downarrow$
baseline	CityScapes	83.7	4.4	56.2	26.3
w/ emb. space	CityScapes	88.1 (4.4)	3.2 (1.2)	48.4 (7.8)	5.9 (20.4)
baseline	*S*(CityScapes,BDD100k)	85.4	4.5	61.3	10.0
w/ emb. space	*S*(CityScapes,BDD100k)	87.6 (2.2)	3.3 (1.2)	63.0 (1.7)	3.9 (6.1)
baseline	CityScapes,BDD100k	83.5	4.8	52.2	22.9
w/ emb. space	CityScapes,BDD100k	91.2 (7.7)	2.9 (1.9)	78.5 (26.3)	2.7 (20.2)

Table 1. Embedding bottleneck (§3.1) - dependence on training with data varying in size and road appearance diversity. $S(\cdot)$ indicates sub-sampling of the datasets to the size of CityScapes and to have roughly equal number of images from each dataset. The proposed embedding bottleneck lead to improvements in all cases, being especially effective for complex data that required to model diverse road appearance; note the decrease in false positives.

4. Experiments

This section presents two main experiments: (i) an ablation study and (ii) a comparison to the state-of-the-art methods. The ablation study consists of three experiments. The first evaluates the benefit of the explicit embedding space. The second ablation study shows the contributions of the individual proposed components (note that not all components are orthogonal and can be used independently, *e.g*. the embedding channels require the explicit embedding space). Lastly, the benefit of the inpainting module is evaluated.

All versions of the proposed method were trained with the same parameters and training data, unless stated otherwise. We follow the standard evaluation protocol, *i.e*. the evaluation is limited to the road region and the two standard performance measures are adopted from [36, 3, 29]. Namely, the False Positive Rate at 95% of True Positive Rate (FPR$_{95}$) and average precision (AP, *i.e*. area under the Precision-Recall curve).

Despite fixing the random seeds in all experiments, the training procedure is non-deterministic due to the implementation of the *cudnn library*. We train the model without the inpainting module, *i.e*. *baseline* + *emb.space* + *emb.channels*, four times to provide a notion about the performance fluctuation and results significance when comparing different versions of the method. The choice to leave out the inpainting module was purely practical (faster training given our limited resources). We report the results for DaCUP method (without the inpainting module) as the mean performance values with standard deviation and we assume that the other method versions will exhibit similar performance uncertainty.

Datasets. The evaluation was carried out on all commonly available real-data anomaly detection datasets. We created two meta-datasets, specifically: Lost-and-Found (LaF) [36], Road Anomaly (RA) [30], Road Obstacles (RO) [29] and Fishyscapes (FS) [3] collectively denoted as *LaFRAROFS* . The second meta-datasets contains data from benchmark [7], including its validation data to utilize all available images, *i.e*. RO plus new (222) and validation (30) images as described in [7], and we denote it as *Obstacle Track+*. We annotated the *Obstacle Track+* dataset to facilitate the evaluation in the ablation study. Tables 3 and 4 show that the results are correlated with the official

baseline	embedding space	embedding channels	inpainting	*LaFRAROFS*		*Obstacle Track+*	
				$\overline{\text{AP}}\uparrow$	$\overline{\text{FPR}}_{95}\downarrow$	$\overline{\text{AP}}\uparrow$	$\overline{\text{FPR}}_{95}\downarrow$
✓				85.4	4.5	61.3	10.0
✓	✓			87.6	3.3	63.0	3.9
✓	✓	✓		90.9±0.9	2.4±0.3	80.5±2.5	2.7±0.7
✓	✓	✓	✓	91.2	2.4	86.1	1.5

Table 2. Ablation study of the methods building blocks. The third row also shows the standard deviation in performance for multiple training runs to established a significance of result differences.

ones. We intentionally did not use the *LostAndFound NoKnown* track from [7] since it is formed exclusively from the data in *LaFRAROFS*. The performance on the meta-datasets is characterized by performance averaged over the respective sub-datasets. These averaged metrics ($\overline{\text{AP}}$ and $\overline{\text{FPR}}_{95}$) were used in [40] and the results we report are consistent and directly comparable to this prior work (in case of the *LaFRAROFS*). Result for individual sub-datasets are available in the supplementary materials.

For comparison with the state-of-the-art methods, we used the official *SegmentMeIfYouCan* benchmark (*SMIYC*) [7]. The performance results are obtained by submitting the per-image estimated anomaly scores through the benchmark website. The results for *LaFRAROFS* were obtained by evaluating the baseline methods available in the *SMIYC* benchmark toolkit (with available pre-trained models). Qualitative results on selected images are shown in Figure 2. The qualitative results demonstrate clearly the main strength and weakness of the proposed method, *i.e*. excellent performance on various surfaces with the ability to recognize false positives induced by road surface texture and unsatisfactory performance for small distance objects.

4.1. Ablation Study

Explicit Embedding Space. In this experiment, three types of training data are used to demonstrate the effectiveness of the explicit embedding space proposed in Section 3.1. The first two datasets test the ability to model multi-modal road appearance. For that purpose, we used CityScapes [11] and a combined dataset of CityScapes and BDD100k [46] that is subsampled (unless stated otherwise) to have a similar number of training examples as CityScapes alone and a similar number of images from CityScapes and BDD100k. The training images from the respective datasets were chosen at random and fixed for all experiments. The subsampled dataset is denoted as *S*(CityScapes,BDD100k). Lastly, we assess the ability to model road appearance in unbalanced scenarios using the full combination of CityScapes and BDD100k datasets (note: BDD100k has approximately four times more datapoints than Cityscapes). The results are summarized in Table 1, showing the benefit of the proposed embedding bottleneck, especially in effectiveness of data utilization and ability to model diverse road appearances (*i.e*. low false positive).

	LaFRAROFS		*Obstacle Track+*	
	$\overline{\text{AP}}\uparrow$	$\overline{\text{FPR}}_{95}\downarrow$	$\overline{\text{AP}}\uparrow$	$\overline{\text{FPR}}_{95}\downarrow$
Maximum Softmax [20]	29.7	28.9	11.8	21.2
+ inpainted	52.3 (22.6)	26.7 (2.2)	41.3 (29.5)	19.8 (1.4)
Mahalanobis [24]	44.7	32.6	31.6	17.9
+ inpainted	55.2 (10.5)	17.0 (15.6)	45.8 (14.2)	9.2 (8.7)
JSRNet [40]	83.7	4.4	56.2	26.3
+ inpainted	87.1 (2.4)	2.9 (1.5)	65.7 (9.5)	22.0 (4.3)
ODIN [26]	50.7	25.9	20.4	18.7
+ inpainted	71.2 (20.5)	11.8 (14.1)	53.4 (33.0)	7.1 (11.6)
Image Resynthesis [30]	66.3	25.0	54.9	9.8
+ inpainted	66.6 (0.3)	25.0 (0.0)	57.9 (3.0)	9.8 (0.0)
SynBoost [13]	77.5	15.4	68.4	3.4
+ inpainted	80.7 (3.2)	14.3 (1.1)	75.5 (7.1)	2.0 (1.4)
Maximized Entropy [8]	86.3	6.4	86.4	1.9
+ inpainted	85.4 (0.9)	6.2 (0.2)	86.3 (0.1)	1.0 (0.9)
DaCUP (ours) w/o	90.9±0.9	2.4±0.3	80.5±2.5	2.7±0.7
+ inpainted	90.7±0.6 (0.2)	2.3±0.3 (0.1)	80.9±1.6 (0.4)	1.6±0.1 (1.1)
+ inpainted trained	91.2 (0.3)	2.4 (0.0)	86.1 (5.6)	1.5 (1.2)

Table 3. Inpainting module impact on state-of-the-art methods. For all methods, it significantly reduces the false positives ($\overline{\text{FPR}}_{95}$) on all datasets. Besides Maximum Entropy [8], with performance effectively unchanged, average precision ($\overline{\text{AP}}$) is improved too. The last three rows show the benefit of jointly training the inpainting module and the main method. For detailed discussion, see the ablation study section (4.1). The full results for the individual datasets are in supplementary materials.

Component Analysis. For the component analysis, all models are trained using the *S*(CityScapes,BDD100k) dataset, mainly to speed up the training process with limited available resources. The individual tested components are: (i) the baseline, which refers to [40], (ii) *embedding space* Section 3.1, (iii) *embedding channels* Section 3.2 and (iv) *inpainting* Section 3.3. Table 2 shows the results for this experiments and highlights the additive performance gains of the individual contributions.

Inpainting Module. This ablation experiment demonstrates the benefit of the inpainting module itself when used as a post-processing step. Ideally, the inpainting model would be trained jointly with the respective methods, however, due to limited resources we proposed this simpler technique that regardless of the sub-optimal combination yields significant performance improvements. The difference between using the simpler post-processing version and jointly trained (as proposed in Section 3.3) is shown in the last two rows of Table 3. The architecture of the simple version is the same as proposed, however, the trainable weights were set manually to put progressively more weight to the higher-level features ($0.2, 0.3, 0.5$ respectively) and the final anomaly score map in the inpainted regions is computed as an average between the perceptual loss and the original output. The inpainting mask is obtained by thresholding the method's output and the threshold was fine-tuned for each method by grid search using values $(0.1, 0.2, ..., 0.9)$. The results are shown in Table 3. The inpainting module helps in most cases, mainly in the reduction of the false positives. The effect is somewhat diminished for the best performing methods, however, even for the best performing (Maximized Entropy [8]) it helps to reduce the false positives on *Obstacle Track+*by half with negligible impact on

	Obstacle Track					*LostAndFound NoKnown*					Processing speed [s]
	AUPR↑	FPR_{95} ↓	sIoU↑	PPV↑	$\overline{\text{F}}_1$ ↑	AUPR↑	FPR_{95} ↓	sIoU↑	PPV↑	$\overline{\text{F}}_1$ ↑	
Maximum Softmax [20]	15.7	16.6	19.7	15.9	6.3	30.1	33.2	14.2	62.2	10.3	0.47
Mahalanobis [24]	20.9	13.1	13.5	21.8	4.7	55.0	12.9	33.8	31.7	22.1	15.46
ODIN [26]	22.1	15.3	21.6	18.5	9.4	52.9	30.0	39.8	49.3	34.5	3.85
JSRNet [40]	28.1	28.9	18.6	24.5	11.0	74.2	6.6	34.3	45.9	36.0	0.09
Image Resynthesis [30]	37.7	4.7	16.6	20.5	8.4	57.1	8.8	27.2	30.7	19.2	0.65
Road Inpainting [29]	54.1	47.1	57.6	39.5	36.0	82.9	35.8	49.2	60.7	52.3	—
SynBoost [13]	71.3	3.2	44.3	41.8	37.6	81.7	4.6	36.8	72.3	48.7	1.62
Maximized Entropy [8]	85.1	0.8	47.9	62.6	48.5	77.9	9.7	45.9	63.1	49.9	0.43
NFlowJS [16]	85.6	0.4	45.5	49.5	50.4	89.3	0.7	54.6	59.7	61.8	—
DaCUP (ours)	81.5	1.1	37.7	60.1	46.0	81.4	7.4	38.3	67.3	51.1	0.80

Table 4. Comparison with the state-of-the-art on *SegmentMeIfYouCan* benchmark (on two tracks – *Obstacle* and *LaF NoKnown*). The DaCUP is among the top 3 methods in all criteria. The last column reports processing time per-image, averaged over sixty 1280 × 720 images, on NVIDIA GeForce RTX 2080 Ti.

Method	μ_A[px] / μ_R	Pr	Re	TPr	FPr	F1
MaskRCNN [18]	- / -	93.1 (64.0)	41.7 (89.0)	27.1 (0.8)	**2.0** (0.4)	57.6 (74.5)
ENet [35]	78 / 85.9	59.8 (17.2)	**96.3** (96.0)	**62.6** (3.8)	42.0 (18.6)	73.8 (29.1)
DeepLabV3 [9]	21 / 97.0	80.0 (43.5)	92.7 (95.8)	60.2 (3.8)	15.1 (5.0)	85.9 (59.9)
BiSeNet [45]	**17** / **97.6**	90.6 (74.8)	89.9 (94.8)	58.4 (3.8)	6.1 (1.3)	90.3 (83.7)
RefineNet [27]	18 / **97.6**	89.1 (67.3)	93.0 (96.3)	60.4 (3.9)	7.4 (1.9)	91.0 (79.2)
WaSR [4]	21 / 97.1	95.6 (84.2)	87.5 (92.1)	56.8 (3.7)	2.6 (0.7)	**91.4** (88.0)
DaCUP (ours)	42 / 93.0	92.3 (64.7)	84.5 (85.5)	54.9 (3.4)	4.6 (1.9)	88.2 (73.6)
w/o inpaint	28 / 95.6	**96.4** (80.0)	81.6 (92.9)	53.0 (3.7)	**2.0** (0.9)	88.4 (86.0)

Table 5. Comparison with top performing methods on the MODS benchmark. The results for the state-of-the-art methods are from the benchmark publication [6]. Performance in the danger zone (15m around the vehicle) is shown in parentheses. The first (bold) and the second (underline) best results are marked.

average precision. If the inpainting module is trained jointly it can manifest in larger performance gain as demonstrated in our proposed method (last three rows in the Table 3).

4.2. State-of-the-Art Comparison

The proposed DaCUP method is compared with state-of-the-art methods on the *SegmentMeIfYouCan* benchmark (Table 4), and on the *LaFRAROFS* dataset (Table 3). The proposed method performs favorably against other state-of-the-art methods, ranking among top-3 in the *SMIYC* benchmark and top-1 on the *LaFRAROFS* dataset (note that the NFlowJS [16] method is not open-source and therefore only the results for the *SMIYC* benchmark from the official leaderboard are available). The processing speed of state-of-the-art methods with available source code is provided in last column of Table 4. We observed that the main issue of the proposed method is the detection of small, distant objects. Our hypothesis is that it is caused by the low resolution of the embedding space features with no lateral skip connection during upsampling in all stages, *i.e*. in the reconstruction module, in the distance-based features and in the coupling module.

4.3. Anomaly detection in maritime scenario

The proposed DaCUP method is primarily intended for an autonomous driving scenario, however, there are no domain specific design choices and thus it is not limited to the driving scenario. It is possible to apply DaCUP to other applications with similar setting, *i.e*. one known class with limited appearance variability vs. the rest. We demonstrate in a maritime scenario on the task segmenting water and the sky; everything else as an anomaly - an obstacle.

We trained the segmentation network and the DaCUP on the *MaSTr1325 - Maritime Semantic Segmentation Training Dataset* [5] with the same setting and parameters (*water* and *sky* as known classes and everything else as anomalies). The method was evaluated on the MODS benchmark [6]. It performed competitively in obstacle detection (see Table 5) with no domain specific knowledge, *e.g*. IMU or horizon estimation module as WaSR [4]. The performance of DaCUP with the inpainting module is degraded because the inpainting network hallucinates many false positives, since it was not trained using marine domain data, whereas the other parts were re-trained. Both methods used the same single threshold (set to 0.3) to classify to water or obstacle.

5. Conclusions

We proposed the DaCUP method, exploiting general properties of anomalous objects. It efficiently models the multi-modal appearance of road surfaces to localize anomalies, and provides image consistent anomaly detection using an inpainting mechanism and distance-based features. The method achieves state-of-the-art results. The proposed inpainting module can be used by other methods (demonstrated by plugging a simplified version as a post-processing step) to greatly reduce false positives. We applied the DaCUP on different data domain and achieve performance comparable to specialized methods. The main limitation of DaCUP is its lower performance for small and distant objects, most likely caused by the use of low resolution features with no lateral skip connections during upsampling.

Acknowledgements This work was supported by Toyota Motor Europe.

References

[1] Subutai Ahmad, Alexander Lavin, Scott Purdy, and Zuha Agha. Unsupervised real-time anomaly detection for streaming data. *Neurocomputing*, 262:134–147, 2017.

[2] Paul Bergmann, Kilian Batzner, Michael Fauser, David Sattlegger, and Carsten Steger. The MVTec Anomaly Detection Dataset: A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection. *Int. J. Comput. Vis.*, 129(4):1038–1059, 2021.

[3] Hermann Blum, Paul-Edouard Sarlin, Juan Nieto, Roland Siegwart, and Cesar 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.

[4] Borja Bovcon and Matej Kristan. WaSR–A Water Segmentation and Refinement Maritime Obstacle Detection Network. *IEEE Transactions on Cybernetics*, pages 1–14, 2021.

[5] Borja Bovcon, Jon Muhovič, Janez Perš, and Matej Kristan. The MaSTr1325 dataset for training deep USV obstacle detection models. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019.

[6] Borja Bovcon, Jon Muhovič, Duško Vranac, Dean Mozetič, Janez Perš, and Matej Kristan. MODS–A USV-Oriented Object Detection and Obstacle Segmentation Benchmark. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–16, 2021.

[7] Robin Chan, Krzysztof Lis, Svenja Uhlemeyer, Hermann Blum, Sina Honari, Roland Siegwart, Pascal Fua, Mathieu Salzmann, and Matthias Rottmann. SegmentMeIfYouCan: A Benchmark for Anomaly Segmentation, 2021.

[8] Robin Chan, Matthias Rottmann, and Hanno Gottschalk. Entropy Maximization and Meta Classification for Out-of-Distribution Detection in Semantic Segmentation. In *Int. Conf. Comput. Vis.*, pages 5128–5137, October 2021.

[9] 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.

[10] Qifeng Chen and Vladlen Koltun. Photographic Image Synthesis With Cascaded Refinement Networks. In *Int. Conf. Comput. Vis.*, Oct 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] Giancarlo Di Biase, Hermann Blum, Roland Siegwart, and Cesar Cadena. Pixel-Wise Anomaly Detection in Complex Driving Scenes. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 16918–16927, June 2021.

[14] Shengyong Ding, Liang Lin, Guangrun Wang, and Hongyang Chao. Deep feature learning with relative distance comparison for person re-identification. *Pattern Recognition*, pages 2993 – 3003, 2015.

[15] Alexey Dosovitskiy and Thomas Brox. Generating Images with Perceptual Similarity Metrics Based on Deep Networks. In *Adv. Neural Inform. Process. Syst.*, page 658–666, 2016.

[16] Matej Grcić, Petra Bevandić, Zoran Kalafatić, and Siniša Šegvić. Dense anomaly detection by robust learning on synthetic negative data, 2021.

[17] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On Calibration of Modern Neural Networks, 2017.

[18] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Int. Conf. Comput. Vis.*, pages 2980–2988, 2017.

[19] 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.

[20] Dan Hendrycks and Kevin Gimpel. A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks. *Proceedings of International Conference on Learning Representations*, 2017.

[21] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual Losses for Real-Time Style Transfer and Super-Resolution. In *Eur. Conf. Comput. Vis.*, pages 694–711, Cham, 2016.

[22] Weicheng Kuo, Christian Häne, Pratik Mukherjee, Jitendra Malik, and Esther Yuh. Expert-level detection of acute intracranial hemorrhage on head computed tomography using deep learning. *Proceedings of the National Academy of Sciences*, 116:201908021, 10 2019.

[23] Alexander Lavin and Subutai Ahmad. Evaluating Real-Time Anomaly Detection Algorithms – The Numenta Anomaly Benchmark. *IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, pages 38–44, 2015.

[24] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Adv. Neural Inform. Process. Syst.*, NIPS'18, page 7167–7177, 2018.

[25] Elad Levi, Tete Xiao, Xiaolong Wang, and Trevor Darrell. Rethinking Preventing Class-Collapsing in Metric Learning With Margin-Based Losses. In *Int. Conf. Comput. Vis.*, pages 10316–10325, October 2021.

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

[27] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian Reid. RefineNet: Multi-Path Refinement Networks for High-Resolution Semantic Segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, July 2017.

[28] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In *Computer Vision – ECCV 2014*, pages 740–755, 2014.

[29] Krzysztof Lis, Sina Honari, Pascal Fua, and Mathieu Salzmann. Detecting Road Obstacles by Erasing Them, 2021.

[30] Krzysztof Lis, Krishna Nakka, Pascal Fua, and Mathieu Salzmann. Detecting the Unexpected via Image Resynthesis. In *Int. Conf. Comput. Vis.*, October 2019.

[31] Xihui Liu, Guojun Yin, Jing Shao, Xiaogang Wang, and Hongsheng Li. Learning to Predict Layout-to-Image Conditional Convolutions for Semantic Image Synthesis. In *Adv. Neural Inform. Process. Syst.*, 2019.

[32] Deen Dayal Mohan, Nishant Sankaran, Dennis Fedorishin, Srirangaraj Setlur, and Venu Govindaraju. Moving in the Right Direction: A Regularization for Deep Metric Learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, June 2020.

[33] Asim Munawar and Clement Creusot. Structural inpainting of road patches for anomaly detection. In *14th IAPR International Conference on Machine Vision Applications (MVA)*, pages 41–44, 2015.

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

[35] Adam Paszke, Abhishek Chaurasia, Sangpil Kim, and Eugenio Culurciello. ENet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation. *ArXiv*, abs/1606.02147, 2016.

[36] 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.

[37] Florian Schroff, Dmitry Kalenichenko, and James Philbin. FaceNet: A Unified Embedding for Face Recognition and Clustering. In *IEEE Conf. Comput. Vis. Pattern Recog.*, June 2015.

[38] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *International Conference on Learning Representations*, 2015.

[39] Paul Smolensky. Chapter 6: Information Processing in Dynamical Systems: Foundations of Harmony Theory. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations*, page 194–281, 1986.

[40] Tomas Vojir, Tomáš Šipka, Rahaf Aljundi, Nikolay Chumerin, Daniel Olmeda Reino, and Jiri Matas. Road Anomaly Detection by Partial Image Reconstruction With Segmentation Coupling. In *Int. Conf. Comput. Vis.*, pages 15651–15660, October 2021.

[41] 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.

[42] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[43] 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.

[44] Yingda Xia, Yi Zhang, Fengze Liu, Wei Shen, and Alan Yuille. Synthesize then Compare: Detecting Failures and Anomalies for Semantic Segmentation. In *Eur. Conf. Comput. Vis.*, 2020.

[45] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. BiSeNet: Bilateral Segmentation Network for Real-time Semantic Segmentation. In *Eur. Conf. Comput. Vis.*, September 2018.

[46] 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. In *IEEE Conf. Comput. Vis. Pattern Recog.*, June 2020.

[47] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S. Huang. Free-Form Image Inpainting With Gated Convolution. In *Int. Conf. Comput. Vis.*, October 2019.