Master Thesis

Czech
Technical
University
in Prague

F3

Faculty of Electrical Engineering
Department of Computer Science

Distance-aware error evaluation for semantic segmentation

Bc. Maroš Pechník

Supervisor: Ing. Tomáš Vojíř, Ph.D.
Field of study: Open Informatics
Subfield: Data Science
May 2023

MASTER‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Pechník Maroš** Personal ID number: **465991**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Computer Science**

Study program: **Open Informatics**

Specialisation: **Data Science**

II. Master’s thesis details

Master’s thesis title in English:

Distance-aware error evaluation for semantic segmentation

Master’s thesis title in Czech:

Vyhodnocení chyb v závislosti na jejich prostorových pozicích v sémantické segmentaci

Guidelines:

The standard evaluation of semantic segmentation methods does not take into account the spatial distribution of errors (incorrectly classified pixels). While the spatial distribution of error may be of less importance in general semantic segmentation, it becomes crucial in specific applications such as anomaly segmentation in autonomous driving (or other industrial applications where positional errors may results in lost profit). The thesis will study the the distance-aware error evaluation and training regimes.
1) Review a recent state-of-the-art methods for anomaly segmentation (focused on application of autonomous driving), evaluation methodologies and available datasets.
2a) Propose and implement a distance-aware evaluation metric for semantic segmentation and discuss the choices.
2b) Propose a distance-aware regularization loss for training
2c) Implement the loss into at least one state-of-the-art method and train the model with the proposed regularization
3a) Evaluate the state-of-the-art methods on standard benchmarks for road anomaly detection (e.g. [0,1]) using the proposed distance-aware metric.
3b) Compare the results of at least one method that is trained with and without the distance-aware loss using standard and the proposed distance-aware metrics.
4) Discuss and visualize common error types of these methods (from 3b.).

Bibliography / sources:

1. SegmentMeIfYouCan: A Benchmark for Anomaly Segmentation (https://segmentmeifyoucan.com/)
2. Daniel Bogdoll and Maximilian Nitsche and J. Marius Zollner, “Anomaly Detection in Autonomous Driving: A Survey”, IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2022
3. Grcić, Matej and Bevandić, Petra and Šegvić, Siniša, “DenseHybrid: Hybrid Anomaly Detection for Dense Open-set Recognition”, European Conference on Computer Vision (ECCV), 2022
4. Chan, Robin and Rottmann, Matthias and Gottschalk, Hanno, “Entropy Maximization and Meta Classification for Out-Of-Distribution Detection in Semantic Segmentation”, IEEE/CVF International Conference on Computer Vision (ICCV), 2021
5. Tomáš Vojíř, Jiří Matas, “Image-Consistent Detection of Road Anomalies As Unpredictable Patches”, Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), 2023

Name and workplace of master’s thesis supervisor:

Ing. Tomáš Vojíř, Ph.D. Visual Recognition Group FEE

Name and workplace of second master’s thesis supervisor or consultant:

Date of master’s thesis assignment: **20.02.2023** Deadline for master's thesis submission: **26.05.2023**

Assignment valid until: **16.02.2025**

Ing. Tomáš Vojíř, Ph.D. Supervisor’s signature	Head of department’s signature	prof. Mgr. Petr Páta, Ph.D. Dean’s signature

III. Assignment receipt

The student acknowledges that the master’s thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the master’s thesis, the author must state the names of consultants and include a list of references.

.	
Date of assignment receipt	Student’s signature

Acknowledgements

I would like to thank my supervisor Mr. Ing. Tomáš Vojíř, Ph.D. for his guidance, helpful advice, and patient approach. I also thank my family and girlfriend, whose support was tremendous.

Declaration

I declare that the presented work was developed independently and that I have listed all sources of information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of university theses.

In Prague, 26. May 2023

Abstract

This master thesis focuses on computer vision applications, specifically anomaly detection in semantic segmentation models. In some cases, such as visual inspection tasks or autonomous driving, in addition to the correct classification of each input pixel, the spatial distribution of the incorrectly classified pixels plays an important role. This work consists of two contributions to this problem. First, an evaluation metric is proposed that takes into account the spatial distribution of the error. The metric is complementary to standard metrics and provides a different view of model performance. Second, a loss function regularization that forces the semantic segmentation model to make fewer distant errors is proposed. The semantic segmentation model retrained with the distant-aware regularization loss retained the performance of standard metrics and improved the distance-aware evaluation metric. The retrained model performs better at the boundaries of the anomaly and classifies them with higher confidence.

Keywords: semantic segmentation, anomaly detection, distance-aware evaluation metric, distance-aware regularization loss

Supervisor: Ing. Tomáš Vojíř, Ph.D.

Abstrakt

Táto záverečná práca sa zaoberá aplikáciami počítačového videnia, konkrétne detekciou anomálií v modeloch sémantickej segmentácie. V niektorých prípadoch, ako napríklad vo vizuálnych inšpekčných úlohách alebo v autonómnom riadení, je okrem správnej klasifikácie každého pixelu dôležitá aj priestorová distribúcia nesprávne klasifikovaných pixelov. V tejto práci je navrhnutá metrika na vyhodnotenie modelu, ktorá berie v úvahu priestorovú distribúciu chýb. Metrika je komplementárna ku štandardným metrikám a poskytuje iný pohľad na výkonnosť modelu. Taktiež je navrhnutá stratová funkcia, ktorá núti model robiť menej vzdialených chýb. Pôvodný model určený na sémantickú segmentáciu pretrénovaný s navrhnutou stratovou funkciou nezhoršil štandardné metriky ale zlepšil metriku, ktorá model hodnotí podľa vzdialenosti chýb. Pretrénovaný model presnejšie klasifikuje hranice anomálneho objektu a anomálie klasifikuje s väčšou istotou.

Kľúčové slová: sémantická segmentácia, detekcia anomálií, vyhodnocovacia metrika v závislosti na vzdialenosti, stratová funkcia v závislosti na vzdialenosti

Preklad názvu: Vyhodnocení chyb v závislosti na jejich prostorových pozicích v sémantické segmentaci

Contents

Figures

Tables

Chapter 1

Introduction

Artificial intelligence has made tremendous progress in recent years and has become a common part of our daily lives. One of the good applications of machine learning is in the computer vision field. Convolutional neural networks used in computer vision consistently increased their performance in the past years, and several network architectures have become defacto standards. Two main tasks of computer vision are image classification and semantic segmentation. Image classification assigns a category to an entire image, e.g., categorizing images based on the type of scene they depict, such as landscapes, cityscapes, indoor scenes, or nature scenes. Semantic segmentation assigns a category to each pixel of the image and therefore is more complex than image classification. For example, in a street scene, the semantic segmentation model could label pixels as road, pedestrian, car, etc. Semantic segmentation is widely used in numerous applications, including autonomous driving, medical image analysis, and robotics. It plays a crucial role in enabling machines to perceive and understand visual information.

The application of semantic segmentation algorithms is increasingly important in industrial applications, such as visual inspection, or in the automotive industry to make it easier for drivers to drive, increase safety and possibly replace drivers in the future. For this purpose, it is important that the neural networks are able to recognize objects that were not encountered during training. This thesis is focused on anomaly detection semantic segmentation models and proposes a novel distance-aware evaluation metric. Moreover, the distance-aware regularization loss is incorporated into the state-of-the-art semantic segmentation model to force the model to make less distant errors.

Motivation. There exist real-world applications where in addition to the correct classification, the distance of the error from the real object is also important. For example, in inspection tasks, when the machine guided by a vision system has to cut out a defective part from an object, and the object is over-segmented, an unnecessarily large part can be cut out, which may be expensive. Similarly, if the model over-segments the part of the car body that needs to be painted, there is unnecessary spending on painting the part that does not need it. On the other hand, undetected object poses a problem as well. The spatial distribution of the error is also important in classifying anomalies in traffic scenes, which is the main focus of this thesis. Nevertheless, the results can also be applied to other segmentation tasks where the spatial distribution of the errors may be of importance.

1.1 Structure of the thesis

The structure of the thesis is split into five main blocks:

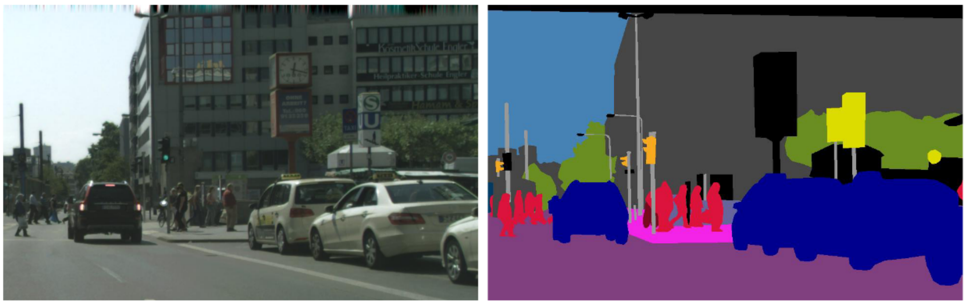
- *Chapter 2* presents a review of recent state-of-the-art methods for semantic segmentation, anomaly detection, evaluation methodologies, and available datasets.
- *Chapter 3* describes the novel distance-aware evaluation metric and discusses the design choices.
- *Chapter 4* proposes a loss regularization for training of semantic segmentation-based models that takes the distance of the error into account.
- *Chapter 5* compares the model trained with and without the distance-aware loss using standard and the proposed distance-aware metrics.
- *Chapter 6* concludes the thesis.

Chapter 2

Review of recent methods for semantic segmentation, datasets, and evaluation metrics.

2.1 Introduction

Semantic segmentation is a widely used technique in processing a digital image. It partitions an image into multiple parts or regions based on the characteristics of the pixels in the image. Labels are used to annotate the data and provide meaningful information about the content of the image. In autonomous driving, labels could be a car, a road, a pedestrian, traffic lights, etc. The goal of the semantic segmentation algorithms is to predict the correct label for each pixel of an input image. It is generally a more complex task than image classification, which predicts a single label for the whole image. In autonomous driving, semantic segmentation is used to help the system identify and locate vehicles and other objects on the road. Semantic segmentation is also widely used in medical image analysis. The illustration of the semantic segmentation is shown in the Figure 2.1.



(a) image (b) semantic segmentation

Figure 2.1: Illustration of the semantic segmentation. The image is courtesy of [1].

In this section, I will provide an overview of recent state-of-the-art methods for semantic segmentation, anomaly detection, evaluation methodologies and available datasets. Similar grouping as in [13, 14] is used.

2.2 Deep learning networks used for semantic segmentation

In this section, I will describe some of the most popular deep neural network architectures that are part of the semantic segmentation models.

- VGG [15] is a deep convolutional neural network proposed by Zisserman and Simonyan from Oxford University designed for image classification. There are multiple versions of VGG depending on the number of layers, such as VGG-13, VGG-16, and VGG-19. VGG is a backbone of several semantic segmentation models [2, 3, 4, 5, 6, 16, 17, 18] .

- ResNet - a residual neural network [19] is a very deep neural network used in various computer vision tasks. It overcomes the difficulty of vanishing gradient in backpropagation for very deep networks by introducing residual blocks. ResNet is the backbone of several semantic segmentation models [16, 17, 18, 20].

- DenseNet [21] is a neural network that has all layers connected with each other. This approach brings advantages such as alleviating the vanishing-gradient problem, strengthening feature propagation, and reducing the number of parameters. DenseNet inspired [6] and is used in [20].

- Vision Transformer (ViT) [22] is a deep-learning model architecture inspired by the success of transformers in Natural Language Processing (NLP). The input image of the ViT model is divided into a sequence of smaller non-overlapping patches that are treated as tokens (words) in NLP. Each patch is then flattened into a vector and processed by a transformer encoder. ViT architechture is used in [23], [24].

2.3 Segmentation models

Besides deep learning segmentation methods mentioned in the previous Section 2.2, many other segmentation methods were proposed, such as statistics-based [25] or geometry-based [26]. In this section, I will focus on deep learning-based segmentation methods. They showed significant improvement in effectiveness compared to traditional segmentation methods. There is a vast number of deep learning techniques. In [13, 14], they analyzed more than a hundred of them and grouped them into categories.

2.3.1 Fully convolutional networks

In one of the first works in the field of deep learning segmentation models, Long *et al.* [2] proposed a fully convolutional network. It significantly improved the accuracy of previous models. Although the accuracy of fully convolutional networks has been overpassed, it inspired many subsequent works.

The model proposed in [2] adapted classification networks AlexNet [27], VGG net [15], and GoogleLeNet [28] into fully convolutional networks and transferred their learned representations by fine-tuning to the segmentation task. A fully convolutional network outputs a spatial segmentation map (Figure 2.2) instead of a classification score returned by AlexNet, VGG net, or GoogleLeNet .

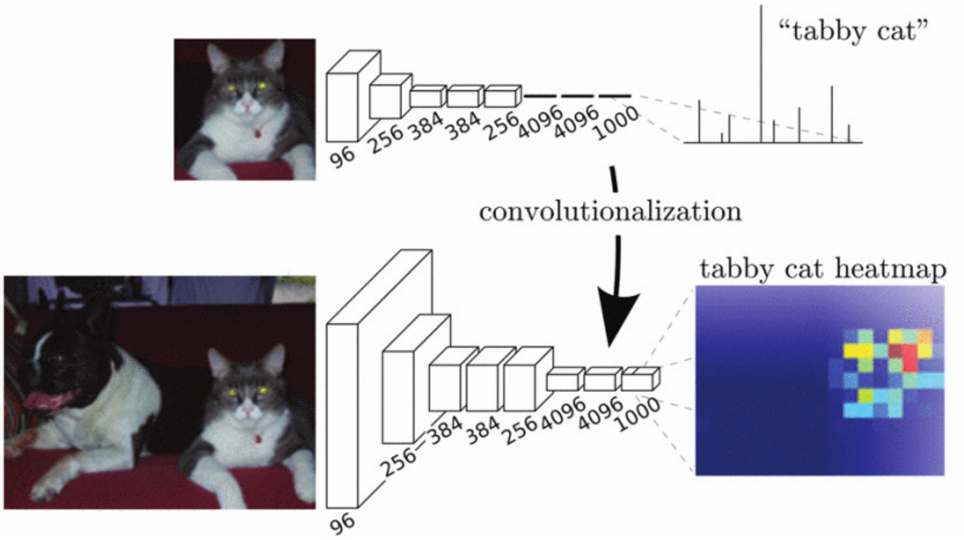


Figure 2.2: Transformation of fully connected layers into convolution layers enables a classification net to output a heatmap. The image is courtesy of [2].

The final classifier layer in each net was decapitated, and all fully connected layers were converted to convolutions. 1×1 convolution was appended to predict scores for each of the PASCAL classes, followed by a deconvolution layer. They defined a novel network architecture, a direct acyclic graph (DAG) with skip connections (Figure 2.3).

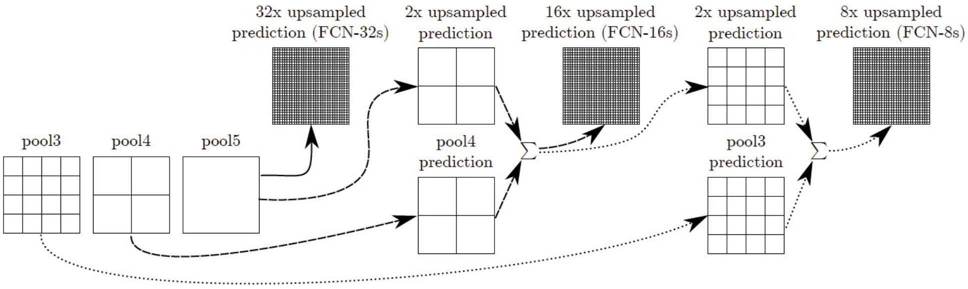

32x upsampled prediction (FCN-32s)
2x upsampled prediction
16x upsampled prediction (FCN-16s)
2x upsampled prediction
8x upsampled prediction (FCN-8s)
pool3
pool4
pool5
pool4 prediction
pool3 prediction

Figure 2.3: DAG networks combining coarse, high layer information with fine, low layer information. The image is courtesy of [2].

Thanks to the skip connections in the graph, the model combines semantic information from coarse layers and appearance information from fine layers in order to produce accurate segmentation. These modifications enabled the network to accept an image of arbitrary size instead of a fixed-size one accepted by classification networks [27, 15, 28]. This work showed that deep learning networks can also be trained for semantic segmentation tasks. This network also has limitations [3] because of using the fixed-size receptive field. It can handle only a single scale semantics within an image, and objects significantly larger or smaller than the receptive field may be fragmented or mislabeled. It also has problems with the correct classification of small objects.

2.3.2 Deconvolution-based models

Traditional neural networks use subsampling operators to reduce the feature map size and increase the receptive field for the final prediction. This approach is not appropriate for semantic image segmentation because it leads to a resolution loss in the output prediction. DeconvNet was the first deconvolution-based segmentation method proposed by Noh *et al.* [3]. It uses convolutional encoder-decoder architecture, which is very popular among DL-based semantic segmentation models. They used the deep network VGG-16 [15], with the last classification layer removed, using its 13 convolutional layers. On top of that, they learned a multi-layer deconvolution network composed of deconvolution, unpooling, and rectified linear unit (RELU) layers. The input of the deconvolution network is an output of the convolution network, feature vector, and the output is a map of pixel-wise class probabilities. The architecture of the DeconvNet is depicted in Figure 2.4. The deconvolution is a mirrored version of the convolution network.

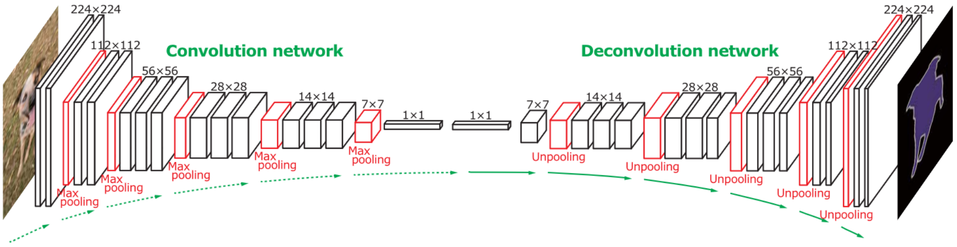


Figure 2.4: Convolution network is followed by a multi-layer deconvolution network to generate the accurate segmentation map. The image is courtesy of [3].

Deconvolution network, as opposed to convolution, enlarges the activations with unpooling and deconvolution operations (Figure 2.5).

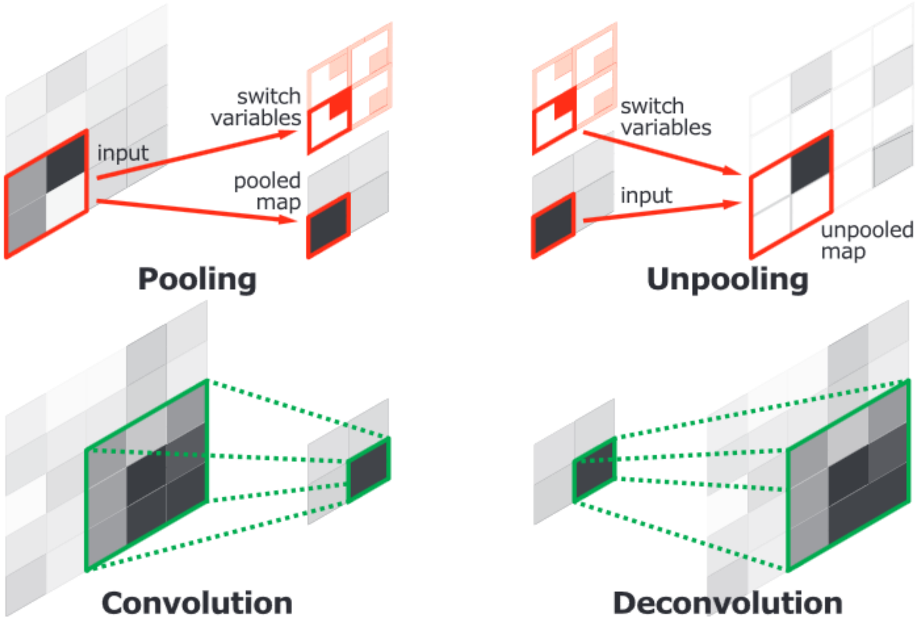


Figure 2.5: Detail of convolution, deconvolution, pooling, and unpooling operations. The image is courtesy of [3].

This algorithm is able to handle object scale variations by eliminating the fixed-size receptive field used by the fully convolutional network (Section 2.3.1). DeconvNet obtained the best accuracy on the PASCAL VOC 2012 dataset in 2015.

Another model based on encoder-decoder architecture is SegNet, proposed by Badrinarayanan *et al.* [4], which was primarily motivated by road scene understanding applications. SegNet has an encoder network and a corresponding decoder network, followed by a final pixel-wise classification layer (Figure 2.6). Similar to DeconvNet, the encoder part of SegNet also consists of 13 convolutional layers corresponding to the first 13 convolutional layers of the VGG-16 network. Removing the last 3 layers of VGG-16 significantly reduces encoder network parameters. The main contribution of SegNet is the decoder part, which upsamples its lower-resolution input feature maps using the memorized max-pooling indices from the corresponding encoder feature maps. This eliminates the need for learning to upsample. In order to produce dense feature maps, the upsampled maps are convolved with trainable filters.

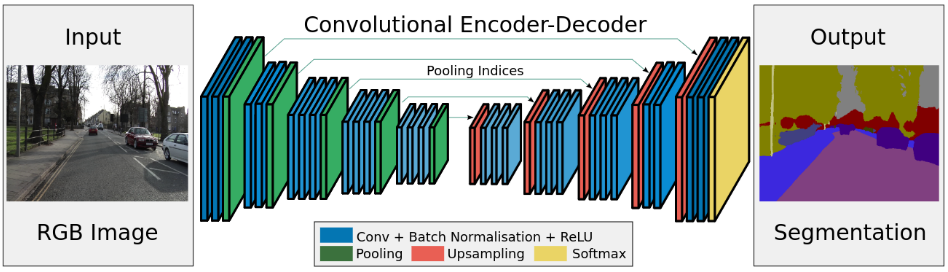


Figure 2.6: SegNet architecture does not consist of fully connected layers; hence, it is only convolutional. The image is courtesy of [4].

The same authors extended the SegNet and proposed a Bayesian SegNet [29]. This model outputs pixel-wise class labels with a measure of model uncertainty for each class.

There are numerous networks based on encoder-decoder architecture. Stacked Deconvolutional Network (SDN) [30] is a deep network consisting of multiple shallow deconvolutional networks that are stacked one by one. U-Net [31] focuses on the segmentation of biomedical images. GridNet [32] has a two-dimensional grid structure that combines accurate prediction and context information.

2.3.3 Recurrent neural networks based models

The recurrent neural network achieved promising results in processing sequential signals, such as speech and text. However, they are helpful in segmentation tasks as well. ReSeg [5] is a segmentation method based on the ReNet [33] network. The architecture of the ReSeg model is shown in the Figure 2.7. ReNet was developed for classification as an alternative to traditional convolution networks. It did not outperform them, but the idea of recurrent neural networks that sweep horizontally and vertically in both directions across the image was altered in [5] for semantic segmentation purposes. Visin *et al.* [5] extended the ReNet architecture and stacked the four ReNet layers on top of pre-trained VGG-16 that extracts generic local features. ReNet layers are followed by up-sampling layers to recover the original image's resolution for the final prediction.

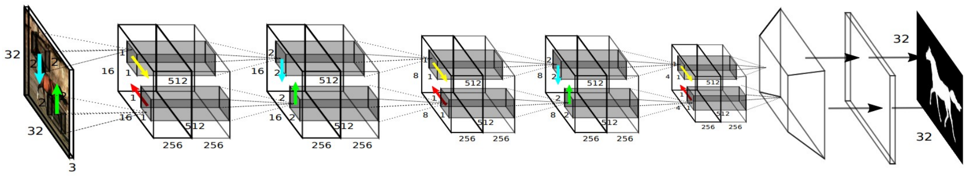
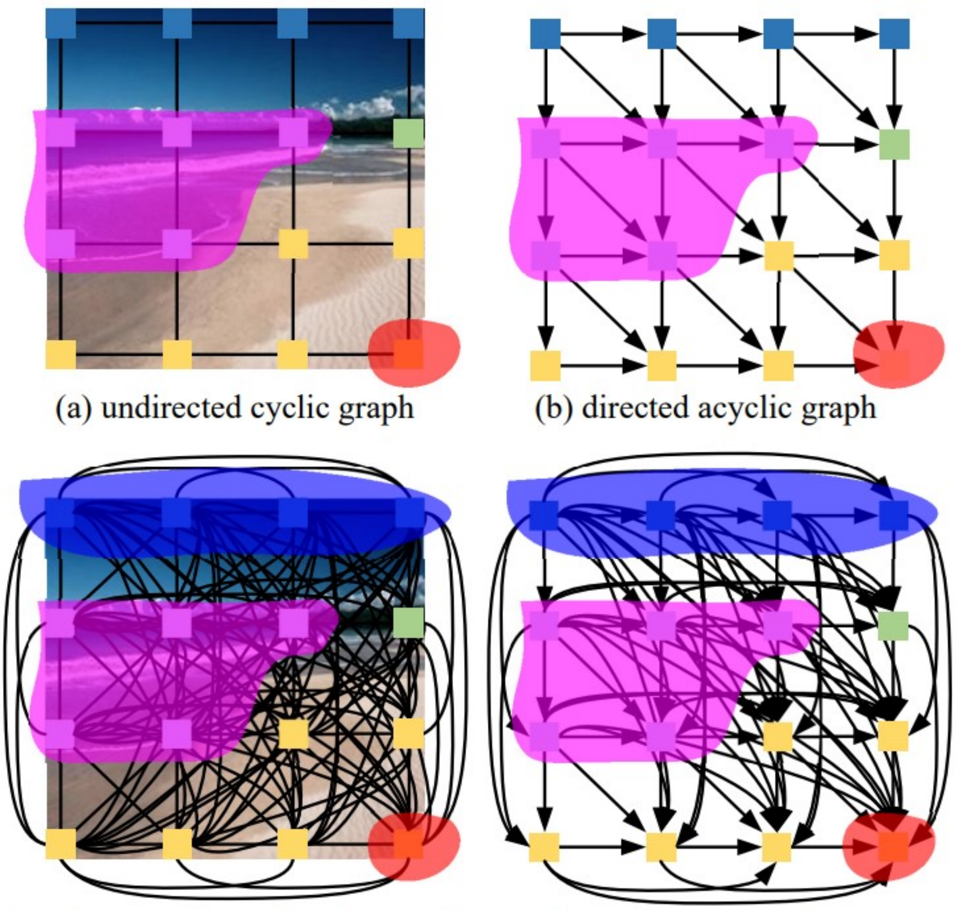


Figure 2.7: The ReSeg model. The pre-trained VGG-16 feature extractor network is not shown. Stacked ReNet layers are followed by by an upsampling and softmax layers. The image is courtesy of [5].

Shuai *et al.* [34] introduced directed acyclic graph RNNs (DAG-RNNs). The proposed network processes DAG-structured images (Figure 2.8b), enabling the network to model long-range semantic dependencies between image units. Local features in an image are considered in a graphical structure. In order to perform segmentation, DAG-RNNs are integrated with convolution and deconvolution layers. In a single feed-forward network pass, the network accepts images of varying sizes and produces the corresponding dense label prediction maps.

Inspired by DenseNet [21], which connects each convolutional layer to every other layer in a feed-forward fashion, Fan *et al.* [6] proposed a dense RNN (Figure 2.9) to capture richer contextual dependencies between image units, which are densely connected with each other (Figure 2.8d). Into dense RNNs, an attention model was introduced in order to assign more importance to helpful dependencies.



(a) undirected cyclic graph (b) directed acyclic graph

(c) dense undirected cyclic graph (d) dense directed acyclic graph

Figure 2.8: A common way to represent the dependencies among image units is to represent an image as an undirected graph. Due to the cycles in an undirected graph, it is difficult to apply RNNs to model dependencies in images directly. The undirected graph is approximated with several directed acyclic graphs (DAGs) to tackle this issue. The image is courtesy of [6].

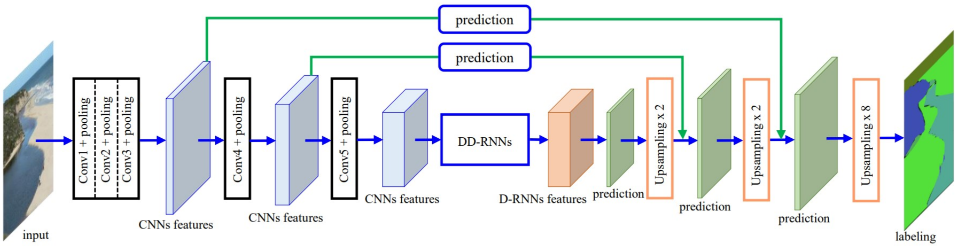


Figure 2.9: The architecture of the model. DAG-structured dense RNNs (DD-RNNs) are placed on top of the 5^{th} layer of VGG-16. DD-RNNs are followed by deconvolution layers to upsample the prediction. The model uses skip strategy to combine low-level and high-level features. The image is courtesy of [6].

Beyon *et al.* [35] worked on recurrent neural networks using Long Short Term Memory (LSTM), mainly composed of the 2D LSTM layer and feed-forward layers. This model addresses the problem of long-range dependencies and takes into account both local and global dependencies in a single process of scene labeling.

2.3.4 Generative adversarial network based models

Generative adversarial networks (GAN) [36] is a type of network frequently used in computer vision. It gained popularity in applications such as style transfer [37], image painting [38], or text-to-image synthesis [39]. Luc *et al.* [7] proposed ANet, the first application of adversarial training to semantic segmentation. It consists of two networks: adversarial and segmentation (Figure 2.10). The segmentation network (generator) partitions the input image into non-overlapping regions. The adversarial network (discriminator) encourages the segmentation model to produce label maps that it could not recognize from the ground-truth labels. The discriminator is trained to binary classify the input image as real or fake.

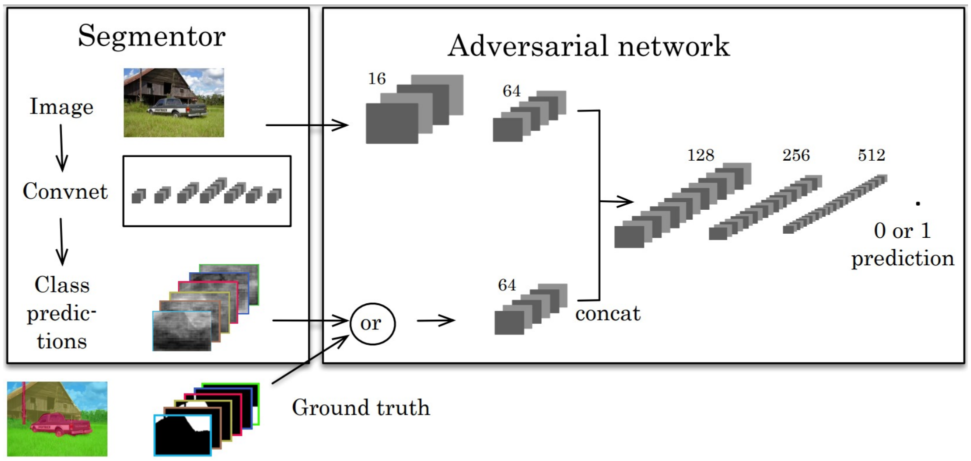


Figure 2.10: The segmentation network (generator) produces per-pixel classification. Labeled image from the segmentation network or ground truth is an input of the adversarial network, which produces class label (1=ground truth, 0=synthetic). The image is courtesy of [7].

Semi-supervised framework based on Generative Adversarial Networks (GANs) was proposed by Souly *et al.* [8]. This model, shown in Figure 2.11, consists of the generative network providing extra training examples and the discriminator network, which labels a training example from K possible classes or labels it as a fake sample (additional class) instead of binary classification of the training example as a real or fake one.

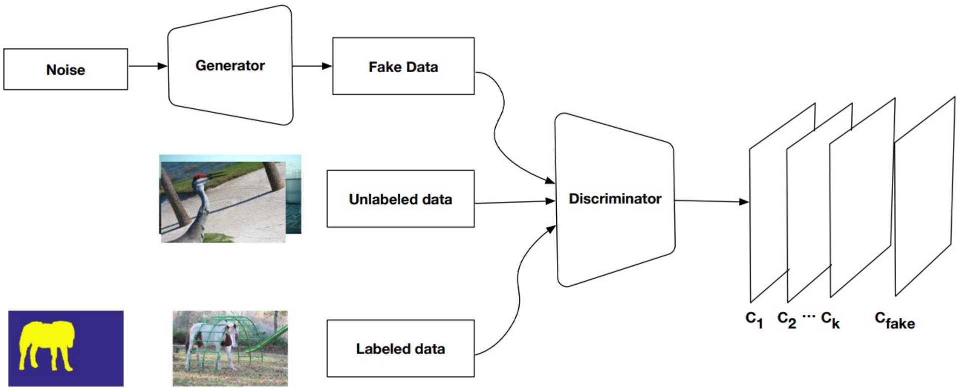


Figure 2.11: Semi-supervised convolutional GAN architecture. The Discriminator creates confidence maps for each class and a label for false data using produced, unlabeled, and labeled data. The image is courtesy of [8].

Hung *et al.* [40] designed a discriminator to distinguish the ground truth from the predicted probability maps. Additionally, semi-supervised learning is enabled by the discriminator that discovers trustworthy regions in predicted results of unlabeled images. SegAN is another example of GAN proposed by Xue *et al.* [41]. As a generator, a fully convolutional encoder-decoder neural network is proposed to generate segmentation label maps. The discriminator has a similar structure to a generator's decoder with a multi-scale loss function. Both parts learn global and local features to capture long-range and short-range dependencies between pixels. SegAN was developed for the segmentation of medical images. There are more GAN architectures for medical purposes, [42] for brain MRI segmentation, [43] for brain tumor segmentation, or [44], which proposes a GAN architecture to mitigate imbalance data problem in medical image semantic segmentation.

Conditional generative adversarial nets (cGAN) [45] is an extension of GAN [36], conditioned on extra information. This extra information, such as class labels, is during training fed to the generator and discriminator as an additional input layer. This way, cGAN is, for example, able to generate images based on class labels.

2.3.5 DeepLab family models

A very famous group of semantic segmentation methods is the DeepLab family [46, 16, 17, 18]. Chen *et al.* [16] re-purposed neural networks designed for classification (VGG-16 [15] and ResNet-101 [19]) by transforming all fully connected layers to fully convolutional layers and by using atrous (dilated) convolution, allowing the computation of the response of any layer at arbitrary resolution. This tackles the problem of low-resolution images outputted by deep neural networks. Atrous convolution can enlarge the size of the receptive field while keeping the number of parameters unchanged. Chen *et al.* experimented with this technique to find the best trade-off between precise localization (small receptive field) and context assimilation (large receptive field). Chen *et al.* [17] upgraded the previous model DeepLabv2 to DeepLabv3. Atrous convolution with different rates is in this model employed in cascade or in parallel to segment various-sized objects. DeepLabv3+ [18] has an encoder-decoder structure with DeepLabv3 as an encoder. The difference between these two models is in the added decoder, which objective is to recover the object boundaries.

2.4 Methods for semantic segmentation anomaly detection

State-of-the-art methods for semantic segmentation are usually trained on the closed set of classes. This is a problem for the classification of previously unseen data, i.e., anomalies or out-of-distribution data. Anomalous objects are usually not from classes available during training (in-distribution) and are usually not visually similar to non-anomalous objects. Detecting these objects is essential for the safety of deployed models in, e.g., automated driving or medical applications.

2.4.1 Methods based on image classification

The first approaches to anomaly detection were developed on deep neural networks proposed for image classification. Lee *et al.* [47] proposed a method for detecting abnormal test samples which can be applied to any pre-trained softmax neural classifier. Noise perturbation was added to the input image and the resulting confidence score is based on Mahalanobis distance. The

ODIN method [48] also does not require any retraining or changes of the pre-trained neural network. Liang *et al.* [49] realized that temperature scaling and adding small perturbations to the input image could increase the softmax score gap between in and out-of-distribution samples. Hendrycks *et al.* [50] demonstrated that predictions produced by softmaxes tend to be higher for correct examples and lower for out-of-distribution samples.

2.4.2 Methods based on learning to detect anomalies

DeVries *et al.* [51] augmented a neural network with a confidence estimation branch and trained the network classifiers that output confidence intervals. These intervals are used to differentiate between in and out-of-distribution samples. By thresholding on the learned confidence intervals, they obtained better results than by thresholding in the softmax prediction probabilities for almost all tested network architectures.

Hendrycks *et al.* [52] proposed an approach called Outlier Exposure. Network heuristics is learned to recognize out-of-distribution examples by seeing them in the training phase.

Bevandic´ *et al.* [20] proposed a multi-task convolutional model with two heads for semantic segmentation and outlier detection. The final output is a combination of these two dense prediction maps. DenseNet [21] or ResNet [19] is used as a convolutional backbone, and both tasks are performed in a single forward pass.

Entropy maximization and meta classification for out-of-distribution detection in semantic segmentation proposed by Chan *et al.* [53] is a method that detects out-of-distribution samples. Chan *et al.* refer to out-of-distribution (OoD) samples as samples not included in the model's semantic space. This approach combines two steps, entropy maximization, and meta classification. In the entropy maximization part, a semantic segmentation network (DeepLabv3+ [18] or DualGCNNet [54]) is re-trained to predict class labels with low confidence scores on OoD samples. Softmax entropy is computed to quantify the level of uncertainty. The loss function is adjusted for OoD samples, producing a negative log-likelihood averaged over all classes. Because of the increased sensitivity in predicting OoD objects, false positive predictions are removed in the meta classification part, where the logistic regression is employed. Removing false positive OoD is based on geometry features (connected components of pixels) and aggregated dispersion measures without access to the ground-truth labels. This model was trained on Cityscapes data, and as a proxy for OoD samples, images from the COCO dataset were randomly picked. As test sets LostAndFound and Fishyscapes datasets were used.

2.4.3 Methods based on image reconstruction

Vojir *et al.* [55] proposed a method called DaCUP (Detection of anomalies as Consistent Unpredictable Patches). The DaCUP method employs an auto-encoder-like architecture with a novel embedding bottleneck. This embedding bottleneck enables the model to capture diverse and multi-modal appearances of known classes, such as the road, and the model can better differentiate between anomalous and non-anomalous objects. Image-conditioned distance-to-class score is used as an additional feature that helps the model to identify anomalies on previously unseen surfaces and to decrease the false positive predictions. An inpainting mechanism is also used to reduce the false positive prediction implementing the principle that anomalies cannot be predicted from their neighborhood and are not similar to anything in the image except themselves. The proposed method achieves state-of-the-art performance.

Generative models can generate high-dimensional feature space from low-dimensional one. Xia et al. [9] proposed SynthCP, a framework consisting of two modules, an image synthesize module, and a comparison module to detect anomalies. Conditional GAN (cGAN) [45] in synthesize module generates a reconstructed image from the segmentation result. The comparison module detects anomalies by comparing reconstructed and input images (Figure 2.12).

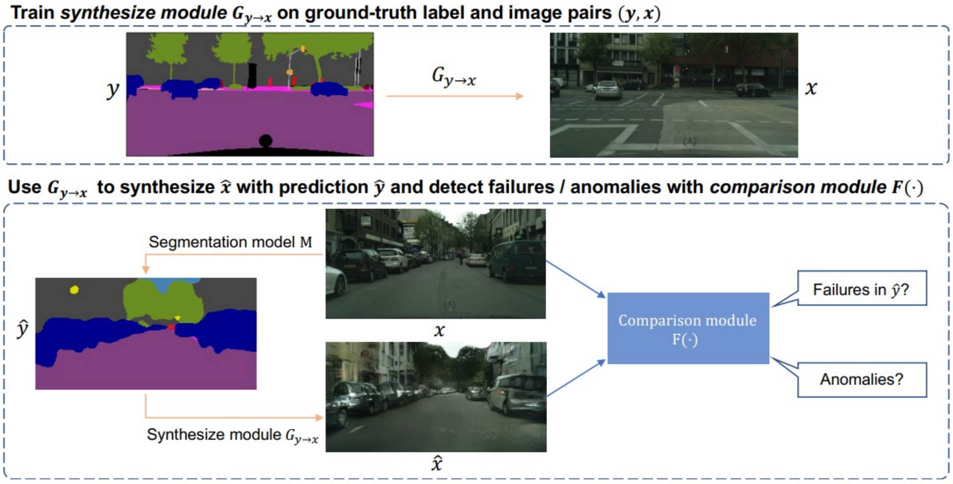

Train *synthesize module* $G_{y\rightarrow x}$ on ground-truth label and image pairs (y, x)
y
$G_{y\rightarrow x}$
x
Use $G_{y\rightarrow x}$ to synthesize $\hat{x}$ with prediction $\hat{y}$ and detect failures / anomalies with *comparison module* $F(\cdot)$
Segmentation model M
$\hat{y}$
x
Synthesize module $G_{y\rightarrow x}$
$\hat{x}$
Comparison module F(·)
Failures in $\hat{y}$?
Anomalies?

Figure 2.12: Detail of the SynthCP framework. The image is courtesy of [9].

Munawar *et al.* [56] limited a generative model by introducing negative learning. The generative model is trained to encode non-anomalous objects into latent representation, then decode it back to the original space, and fail to do the same on anomalous objects. Anomaly is determined based on the similarity between the input and the reconstructed signal. The following works focus on detecting anomalies, specifically on the road.

Creusot and Munawar [57] use a compressive Restricted Boltzman Machine neural network to reconstruct the road and create a deep feature representation. The anomaly detection is performed by comparing the observed and reconstructed road patches.

Lis *et al.* [10] first find image patches and inpaint them with the surrounding road texture. In the second step, they compare the original image and the inpainted one through the discrepancy network to check if they are similar enough (Figure 2.13).

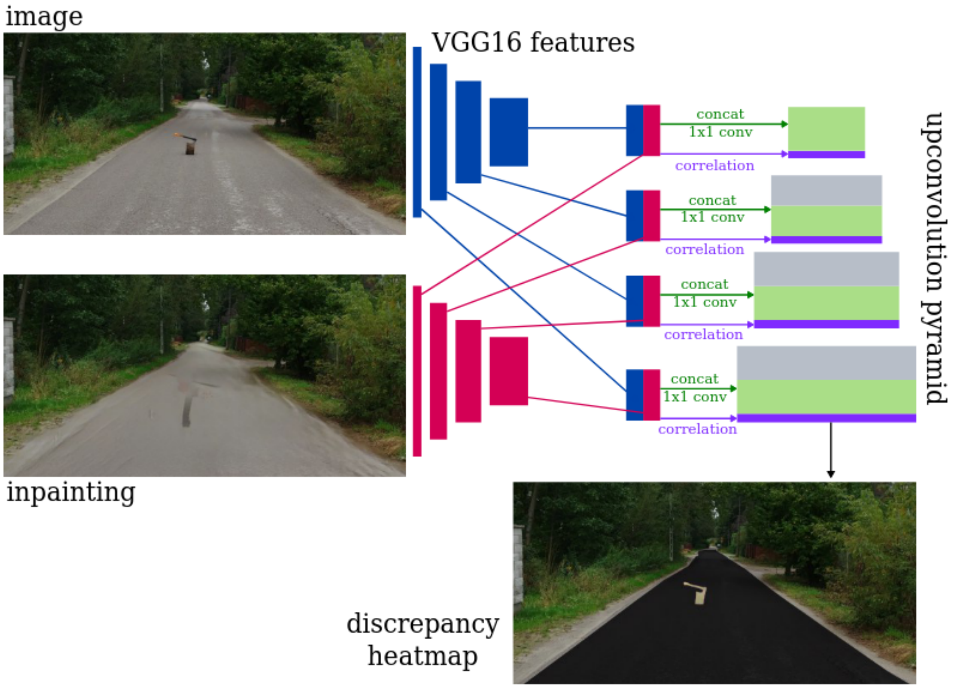


Figure 2.13: Detail of the discrepancy network's architecture. The image is courtesy of [10].

SynBoost proposed by Di Biase *et al.* [58], is a robust framework combining segmentation uncertainty and re-synthesizing the image from the semantic label map. It contains a segmentation module to obtain semantic labeling with the uncertainty of the prediction and synthesis module (cGAN [45]) generating an image from the given semantic map. The dissimilarity module predicts the anomaly segmentation map out of original and generated images and semantic map with uncertainties.

2.5 Datasets

In the following paragraphs, popular datasets used for the training of semantic segmentation neural networks are presented.

2.5.1 Datasets mainly used in semantic segmentation tasks

PASCAL Visual Object Classes (VOC) 2012 [59] is a very popular dataset in the field of computer vision containing well-annotated images. This dataset became a benchmark for object detection; however, it can also be used for classification, segmentation, action recognition, and person layout. The dataset consists of 1464 images for training, 1449 for validation, and 1456 for testing purposes. It includes 20 object classes and one class for background. Most of the segmentation methods described in this section have been evaluated on the PASCAL VOC 2012 dataset. The augmented version of PASCAL VOC 2012 [60] with more than ten thousand images in the training set is more frequently used.

Pascal Context [61] is an extension of the PASCAL VOC 2010 segmentation task containing 10 103 images for training and validation and 9637 for testing. There are 59 semantic classes frequently used in this dataset.

Microsoft COCO: Common Objects in Context [62] is a dataset containing 328 thousand images of complex everyday scenes with 91 object types in their natural context.

ADE20K [63] is a densely annotated dataset with images of complex everyday scenes. The training set comprises more than 20 thousand images, the validation set contains 2 thousand images, and the test set has 3 thousand images. This dataset involves 150 semantic categories.

Kitti [64] is a dataset captured from a moving car. Geiger *et al.* recorded 6 hours of traffic scenarios capturing real-world traffic situations. The original dataset does not contain ground-truth labels.

CamVid [65] is a video-based database with per-pixel ground-truth hand-labeled 32 classes. It contains video sequences of driving scenes. The original dataset is split by Sturgess *et al.* [66] into the train, validation, and test set, and the number of classes is reduced to 11.

2.5.2 Datasets mainly used in anomaly segmentation tasks

Cityscapes [67] is a large-scale dataset comprising diverse video sequences recorded in 50 European cities. It contains 5 thousand well-annotated images and 20 thousand additional coarse-annotated images. Objects are annotated into 30 classes grouped in 8 categories.

LostAndFound [11] contains anomalous objects and obstacles in street scenes in Germany in more than 2000 images. Only anomalies and roads are labeled. Images show limited diversity because they are usually frames of videos captured in single scenes.

Fishysapes [68] is a public validation dataset for anomaly detection in semantic segmentation for urban driving containing 100 images from the original LostAndFound data with refined labels. Anomalous objects can appear everywhere in the image, not only on the road. The issue of low diversity is overcomed by adding synthetic data to the real images.

RoadAnomaly21 [69] is a general anomaly segmentation benchmark in the full street scenes. It is an extension of [70] with corrected labels, removed low-quality images, and added newly collected images. Each image contains at least one anomaly, which can appear anywhere in the image. Added images were collected from the internet and therefore depict various environments. Pixel-level annotations include three classes: anomaly / obstacle, not anomaly / not obstacle, and void.

RoadObstacle21 [69] contains pixel-level annotated images. Obstacles that appear on the road in front of the car can be understood as anomalies. They are at different distances from the car and are surrounded by road pixels. The diversity of the scenes is ensured by different road surfaces, weather, and lighting conditions. Pixel-level annotations are the same as in RoadAnomaly21.

StreetHazards [71] is a synthetic dataset designed for anomaly detection. Diverse foreign objects are re-rendered into the driving streets to ensure the background context and a wide variety of anomalous objects. The dataset contains 7 thousand images with 18 classes.

BDD100K [72] is a large driving video dataset consisting of 100 thousand videos, each about 40 seconds long. They were recorded in different weather conditions and at different times of the day. The dataset contains, for example, the annotations for scene tagging, drivable area, semantic and instance segmentation (22 classes), lane marking, and object bounding boxes.

StreetScenes [73] is a huge dataset video anomaly detection. It consists of more than 56 thousand frames for training and more than 146 thousand frames for testing, which are extracted from the original videos (46 training and 35 testing high-resolution video sequences). Ground-truth annotations for 17 different anomaly types are provided for the test set as bounding boxes around anomalous events.

2.6 Evaluation metrics

In the below section, I will summarize the most popular metrics for semantic and anomaly segmentation. For anomaly segmentation, the class imbalance is typical because anomaly usually covers only a small part of an image. The notation used is as follows: p_{ij} denotes a pixel of class i predicted as a pixel of class j. In other words, p_{ii} denotes True Positive (TP), p_{ji} False Positive (FP), p_{ij} False Negative (FN), p_{jj} True Negative (TN). There are total of K classes.

2.6.1 Pixel-level evaluation

Pixel Accuracy (PA) is defined as the proportion of correctly classified pixels over the total number of pixels.

$$\text{PA} = \frac{\sum_{i=0}^{K} p_{ii}}{\sum_{i=0}^{K}\sum_{j=0}^{K} p_{ij}} \tag{2.1}$$

Mean Pixel Accuracy (MPA) is an extension of PA, where per-class Pixel Accuracy is averaged. It is defined as:

$$\text{MPA} = \frac{1}{K}\frac{\sum_{i=0}^{K} p_{ii}}{\sum_{i=0}^{K}\sum_{j=0}^{K} p_{ij}} \tag{2.2}$$

Precision / Recall are popular metrics for evaluating classical image segmentation algorithms. Precision (also known as Positive Predictive Value (PPV)) measures the ratio between true positive predictions and all positive predictions. It is a likelihood that a positive example is truly positive. Recall is a likelihood that positive example was correctly classified as positive.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \text{ Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2.3}$$

TPR / FPR True Positive Rate (TPR) reflects the model's ability to identify positive instances correctly, while False Positive Rate (FPR) represents the rate of incorrect classifications of negative instances as positive. These metrics are often used together to assess the performance of binary classification models, with the goal of achieving high TPR and low FPR values. TPR is also known as Recall.

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \text{ FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \tag{2.4}$$

FPR$_{95}$ metric measures how many false positive predictions are made to obtain a 95% True Positive Rate.

F1-Score combines precision and recall and is useful for evaluating models where the classes are imbalanced.

$$\text{F}_1 = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}} = 2\frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{2.5}$$

Area Under the Receiver Operating Characteristic curve (AuROC is calulated by measuring the area under the ROC curve. The x-axis of the ROC curve is the False Positive Rate, and the y-axis is the True Positive Rate. TPR and FPR for the ROC curve are obtained by varying the classification threshold. The ROC curve is not suitable for highly imbalanced data.

Area under the precision-recall curve (AuPRC) is useful for classification with imbalanced classes because it emphasizes detecting minority class. The x-axis of a Precision-Recall (PR) curve is the recall, and the y-axis is the precision. The score given by AuPRC is an area under the PR curve obtained by varying thresholds for Precision and Recall computation.

2.6.2 Region-level evaluation

Intersection over Union (IoU) is a widespread metric for semantic segmentation tasks. It is defined as the area of intersection between the ground truth map and the predicted segmentation map divided by the union of those maps.

$$\text{IoU} = \frac{\sum_{i=0}^{K} p_{ii}}{\sum_{i=0}^{K} p_{ii} + \sum_{i=0}^{K} \sum_{j=0}^{K} p_{ij} + \sum_{i=0}^{K} \sum_{j=0}^{K} p_{ji}} \tag{2.6}$$

Mean-IoU is a widely used metric for reporting the performance of segmentation algorithms. It denotes the average of the per-class IoU.

$$\text{MIoU} = \frac{1}{K} \frac{\sum_{i=0}^{K} p_{ii}}{\sum_{i=0}^{K} p_{ii} + \sum_{i=0}^{K} \sum_{j=0}^{K} p_{ij} + \sum_{i=0}^{K} \sum_{j=0}^{K} p_{ji}} \tag{2.7}$$

sIoU is an adjusted version of the component-wise intersection over union (IoU) for ground-truth components proposed by Chan *et al.* [69]. The component of pixels is defined as pixels, where a pixel and its 8 surrounding pixels have the same label. Chan *et al.* denote $\mathcal{Z}_c$ as set of pixels labeled as anomaly, $\mathcal{K} \subseteq \mathcal{P}(\mathcal{Z}_c)$ with $\mathcal{P}(\mathcal{S})$ as the power set of a set $\mathcal{S}$, the set of anomaly components according to the ground-truth. $\hat{\mathcal{K}} \subseteq \mathcal{P}(\mathcal{Z}_c)$ is the set of components that are predicted as anomalous. The sIoU is defined as:

$$\text{sIoU(k)} = \frac{\mid k \cap \hat{K}(k) \mid}{\mid (k \cup \hat{K}(k)) \setminus \mathcal{A}(k) \mid} \text{ with } \hat{K}(k) = \bigcup_{\hat{k} \in \hat{\mathcal{K}}, \hat{k} \cap k \neq \emptyset} \hat{k} \tag{2.8}$$

$k \in \mathcal{K}$ and $\mathcal{A}(k) = \{z \in k' : k' \in \mathcal{K} \setminus \{k\}\}$. In other words, if the predicted component covers multiple ground-truth components, those pixels are excluded from the union.

The metrics mentioned above do not consider the spatial distribution of the error. This is the reason for proposing a distance-based evaluation metric.

Chapter 3

Distance-aware evaluation metric

As mentioned in Section 2.6, commonly used metrics do not take into account the spatial distribution of the error. In this chapter, I propose a generic distant-aware evaluation metric for semantic segmentation tasks that we apply later on for the evaluation of road anomaly detection methods. The distance-based error score (Section 3.2) is inspired by standard metrics using False Positive and False Negative predictions. False predictions are weighted by the distance maps (Section 3.1), which assign a value to each pixel in an image representing the shortest distance between that pixel and a specific reference point(s). To address the problem of class imbalance, normalization is proposed in Section 3.3.

3.1 Distance maps

To represent a tentative error of each pixel used for the computation of the metric, we first compute distance maps. Distance map assigns a value to each pixel in an image, indicating the shortest distance between that pixel and a specific reference point or a set of reference points. Distance maps are later used to calculate the distance-based error score of the prediction. In the proposed distance-aware evaluation metric, one distance map is generated for a positive class and one for a negative class(es). In the special case of road anomaly detection, one distance map is computed for anomaly pixels and one for road pixels. The distance map of anomaly pixels holds for each pixel the value representing the distance between the road pixel and the closest anomalous pixel. Analogously, it also applies to the distance map of road pixels. Formally, it can written as:

$$\mathrm{DM}^{k'}(\mathrm{px}) = \min_{\mathrm{GT}(\mathrm{px'}) \in k'} ||\mathrm{px} - \mathrm{px'}||_2 \qquad (3.1)$$

where *k'* denotes the selected class from the set of available classes k, GT is a ground-truth matrix:

$$\mathbf{GT}; (\mathrm{gt}_{ij}) \in \{k\}^{\mathrm{h} \times \mathrm{w}},$$

and px is the pixel for which the distance score is computed.

In the case of the proposed distance-aware (D-A) regularization metric, the distance map used for the calculation of the D-A error score has the form:

$$\mathrm{DM}^{k'}_{\ln}(\mathrm{px}) = \ln(\mathrm{DM}^{k'} + 1) = \ln(\min_{\mathrm{GT}(\mathrm{px'}) \in k'} ||\mathrm{px} - \mathrm{px'}||_2 + 1). \qquad (3.2)$$

Having two classes: R for the road pixel and A for the anomaly pixel, $k \in \{\mathrm{R}, \mathrm{A}\}$. Distance maps generated for the case of two classes, $k \in \{\mathrm{R}, \mathrm{A}\}$, are shown in Figure 3.2. Anomalous object visualized in distance maps (Figure 3.2) is highlighted in Figure 3.1.

L2 distance is a natural choice in measuring distances; it gives a straight line distance between two points. It is smooth and differentiable compared to the L1 distance. The logarithm function is used in order to limit the effect of the distant errors. Because the logarithm is not defined for zero, one is added to the L2 distance to overcome this issue.

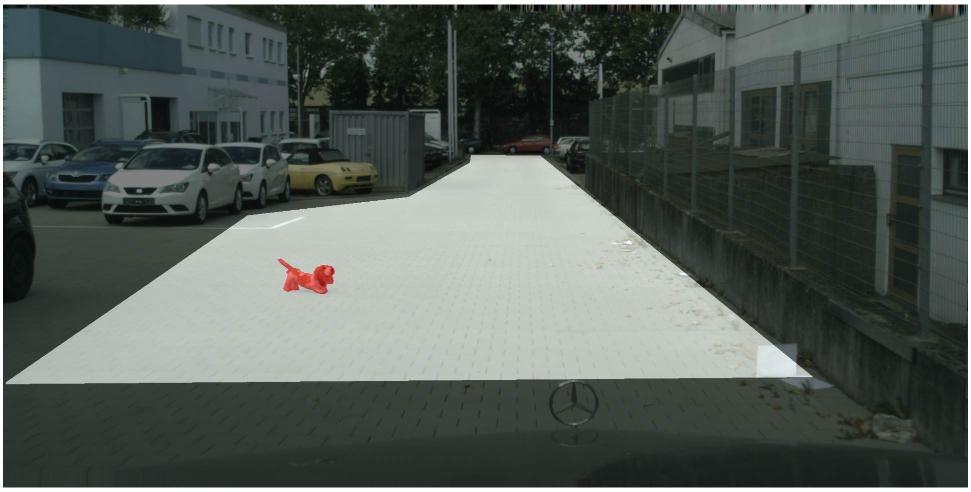
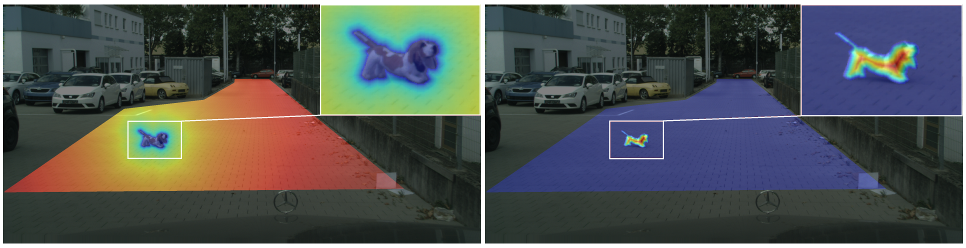


Figure 3.1: Image from the LostAndFound dataset [11]. The region of interest is highlighted in white, with the anomalous object highlighted in red.



(a) : The distance map of anomaly pixels blended over the original image.

(b) : The distance map of road pixels blended over the original image.

Figure 3.2: Visualization of the distance maps using a jet colormap [12] with the detail on the anomaly.

3.2 Distance-based error score

Distance maps (Equation 3.2) are used for the computation of the distance-based error score. Distance-based error score is inspired by standard metrics that use False Positive (FP) and False Negative (FN) predictions. The proposed metric weighs FP and FN by the distance maps. If the false prediction is distant from the true class, it increases the proposed metric more than if the false prediction is made closer to the true class. I propose to decompose the error score into weighted false positive - *wfp* and weighted false negative - *wfn*. For some applications, one type of error may be of more importance, and by the decomposition, one may control the impact of each type of error. The formulas are the following:

$$\mathrm{wfp} = \sum_{\{\mathrm{px}_i : \mathrm{GT}(\mathrm{px}_i) \in \mathrm{kn} | \forall i\}} \mathrm{DM}_{\mathrm{ln}}^{\mathrm{kp}}(\mathrm{px}_i) \cdot [[\mathrm{PI}_\theta(\mathrm{px}_i) = \mathrm{kp}]], \tag{3.3}$$

$$\mathrm{wfn} = \sum_{\{\mathrm{px}_i : \mathrm{GT}(\mathrm{px}_i) = \mathrm{kp} | \forall i\}} \mathrm{DM}_{\mathrm{ln}}^{\mathrm{kn}}(\mathrm{px}_i) \cdot [[\mathrm{PI}_\theta(\mathrm{px}_i) \in \mathrm{kn}]], \tag{3.4}$$

where *kp* denotes the positive class, and *kn* denotes the negative class. There could be more semantic classes belonging to the negative class. The distance map $\mathrm{DM}_{\mathrm{ln}}^{\mathrm{kn}}(\mathrm{px}_i)$ assigns to the pixel px_i the closest distance to the pixel belonging to the negative class. In the case of two classes, *Road* and *Anomaly*, $kp = A$ and $kn = R$.

Matrix PI_θ holds the class predicted by some particular method:

$$\mathbf{PI}_\theta ; (\mathrm{pi}_{\theta ij}) \in \{\mathrm{kp}, \mathrm{kn}\}^{\mathrm{h} \times \mathrm{w}},$$

and double box brackets mean

$$[[true]] = 1,$$
$$[[false]] = 0.$$

Distance-based error score s is

$$\mathrm{s} = \alpha \cdot \mathrm{wfp} + (1 - \alpha) \cdot \mathrm{wfn}. \tag{3.5}$$

The parameter α can be set based on the purpose of the classification task to penalize one type of error more than another. For the sake of simplicity, in the proposed metric, $\alpha = \frac{1}{2}$.

3.3 Distance-based error score with normalization

An apparent drawback of the distance-based error score is that a tiny object does not have a significant distance from its boundary and therefore does not appropriately increase the error score s while misclassified. A natural way of addressing such imbalance is to introduce some form of normalization. The approach with normalization of *wfp* and *wfn* on two classes, R for road and A for the anomaly, in the following equations was tried.

$$\mathrm{wfp} = \frac{1}{\sum_{\mathrm{px}} \mathrm{DM}^{\mathrm{A}}(\mathrm{px})} \sum_{\{\mathrm{px}_i : \mathrm{GT}(\mathrm{px}_i) = \mathrm{R} | \forall i\}} \mathrm{DM}^{\mathrm{A}}(\mathrm{px}_i) \cdot [[\mathrm{PI}_\theta(\mathrm{px}_i) = \mathrm{A}]] \tag{3.6}$$

$$\mathrm{wfn} = \frac{1}{\sum_{\mathrm{px}} \mathrm{DM}^{\mathrm{R}}(\mathrm{px})} \sum_{\{\mathrm{px}_i : \mathrm{GT}(\mathrm{px}_i) = \mathrm{A} | \forall i\}} \mathrm{DM}^{\mathrm{R}}(\mathrm{px}_i) \cdot [[\mathrm{PI}_\theta(\mathrm{px}_i) = \mathrm{R}]] \tag{3.7}$$

However, the normalization caused the *wfn* score to be much more important than the *wfp* score, and the false positive pixels, even far from the anomaly, did not add a considerable score to the final error score, which is the weighted sum of the *wfp* and *wfn*. As a result normalization from equations 3.6 and 3.7 was dropped. Without normalization, the *wfn* score is determined only by the size of the anomaly. If some small anomaly is not detected at all, it does not increase the error score adequately. This was addressed by the normalization of the distance map described in the following section.

3.3.1 Normalization for each anomaly

To eliminate the problem described in the previous section, the distance map of road pixels, used for the computation of the *wfn* score, was computed as follows: first, connected components were found for each anomaly. Then the distance map from the road and ignore pixels to anomaly pixels was calculated. Each component was normalized (divided by the maximum distance within the component) to have the highest error in its center. In order to have the distance maps on the same scale, the distance map of road pixels is divided by its maximum value too. Figure 3.3 shows that each anomaly has the highest error in the center regardless of its size.

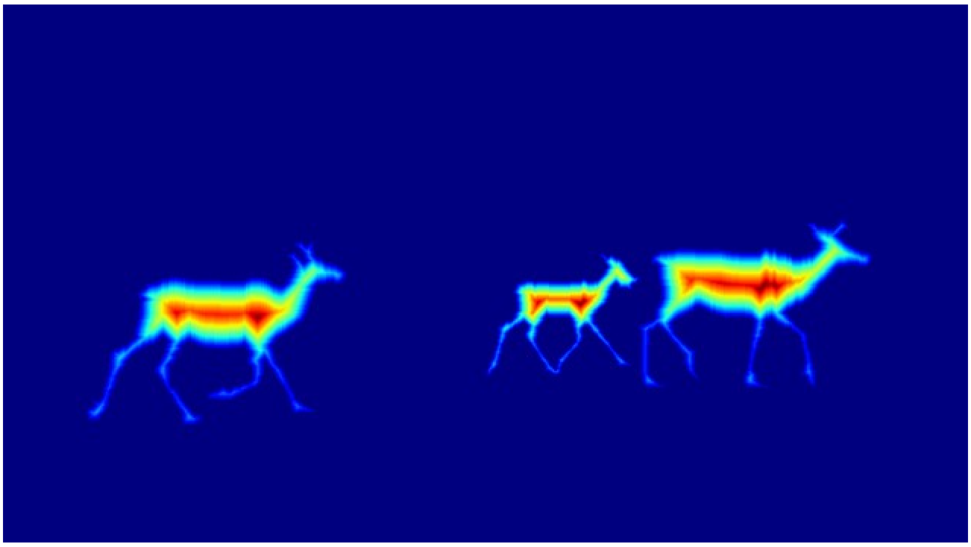


Figure 3.3: The distance map of road pixels. With normalization, each anomaly has the highest error in its center.

Chapter 4

Loss function

4.1 Loss functions used for semantic segmentation

Deep learning algorithms most often use stochastic gradient descent to optimize the objective. In the case of semantic segmentation, the objective is a loss function, which measures the difference between the current output of the algorithm and ground-truth labels. This measurement is sent as a feedback signal to improve the model performance. This way, the model learns. The most common loss functions for semantic segmentation, Cross-entropy loss (Section 4.1.1), and Dice loss (Section 4.1.2) are described in the following paragraphs. However, they do not take the spatial distribution of the error into account.

4.1.1 Cross-entropy

Cross entropy loss measures the model's performance, which outputs a probability value between 0 and 1. The cross-entropy loss increases as the predicted probability diverges from the true label. Cross-entropy loss for M classes is defined as:

$$L_{CE} = -\frac{1}{N} \sum_{n=1}^{N} \sum_{i=1}^{M} y_i^n \log(\hat{y}_i^n), \tag{4.1}$$

where N is the number of pixels and y_i^n is a ground-truth label and $\hat{y}_i^n$ is the predicted probability of the model for the n^{th} pixel and i^{th} class. In case of binary classification, binary cross-entropy is used:

$$L_{BCE}(y^n, \hat{y}^n) = -\frac{1}{N}\sum_{n=1}^{N} y^n \log(\hat{y}^n) + (1-y^n)\log(1-\hat{y}^n), \tag{4.2}$$

4.1.2 Dice loss

Dice loss is based on the dice coefficient, which measures the overlap of two samples. Dice coefficient for 2 classes is calculated as:

$$D_{coef} = \frac{2yp+\epsilon}{y+p+\epsilon}, \tag{4.3}$$

where y is a ground-truth label and p is a predicted label. ϵ is a constant that ensures that the denominator is not a zero. Dice loss is then calculated as:

$$D_{loss} = 1 - D_{coef} \tag{4.4}$$

4.2 Distance-aware regularization loss

Distance-aware regularization loss penalizes mistakes made by the model which are far from the true class more than the ones that are closer to the true class.

4.2.1 Proposed distance-aware regularization loss

Distance maps are used to account for distance in the loss function. Distance map $\mathrm{DM}^k(n)$ for class k is computed for each pixel n of an input image according to the Equation 3.1. The proposed distance-aware loss function has the form:

$$L_{distance} = \frac{1}{d}\sum_{k}^{K}\sum_{n \notin k}^{N} \hat{y}_k^n \cdot \mathrm{DM}^k(n), \tag{4.5}$$

where

$$\hat{y}_k^n = \sigma_k(n)$$

is a softmax output of the model for class k and pixel n, N is the total number of pixels in the input image, and d is the diagonal of the input image. The proposed loss function penalizes high softmax probabilities that are distant from the true class. In the case of two classes, road and anomaly, $k \in \{R, A\}$.

4.3 Implementation of the distance-aware regularization loss into the Dacup model

The distance-aware loss function was implemented into the state-of-the-art model Dacup (version without the inpainting module, which is trained faster), proposed by Vojíř *et al.* [55], described in the Section 2.4.2. For the first five epochs, the Dacup model is trained using only triplet loss $L_{final} = L_{tri}$. Between epochs six and ten, the loss function is $L_{final} = L_{tri} + L_R$. After ten epochs, the loss has a form:

$$L_{final} = \lambda_{xent} L_{xent} + \lambda_{tri} L_{tri} + \lambda_R L_R, \tag{4.6}$$

where $\lambda_{xent} = 0.6$, $\lambda_{tri} = 0.2$, $\lambda_R = 0.2$ are the weights, L_{tri} and L_R are regularizations, for details please see [55], and L_{xent} is the binary negative log likelihood loss (Equation 4.2).

The distance-aware loss (Equation 4.5) was incorporated into the final loss (Equation 4.6) as follows:

$$L_{final} = \lambda_{xent}(L_{xent} + L_{distance}) + \lambda_{tri} L_{tri} + \lambda_R L_R. \tag{4.7}$$

Dacup is trained using the binary negative log likelihood, which penalizes the low confidence of the correct class predicted by the model. $L_{distance}$ penalizes high predicted confidence of the incorrect class proportionally to its distance to the nearest correct class. The aim is to make fewer distant errors.

4.3.1 Retraining the model with the proposed distance-aware loss function

The model was retrained with the same parameters as the original Dacup model. The version without the inpainting module was chosen due to faster training. It was trained on the LostAndFound dataset and evaluated on the test split of LostAndFound (LaF) [11], RoadAnomaly (RA) [69], and

RoadObstacles (RO) [69]. In order to evaluate the distance-aware metric, threshold θ had to be picked. Pixels with softmax score for anomaly class $\hat{y}^n_{k=A}$ higher than the threshold θ are classified as anomalous. The threshold θ was selected based on the LostAndFound validation split as a value, where the True Positive Rate (Sensitivity) was 95%.

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} = 0.95$$

Based on this condition, thresholds for original Dacup and retrained Dacup were selected as follows:

$$\theta_{original} = 0.95,$$

$$\theta_{retrained} = 0.96.$$

To quickly check if the model is minimizing the distance, the error was computed similarly to the $L_{distance}$ (Equation 4.5):

$$\mathrm{error} = \frac{1}{d} \sum_{k}^{K} \sum_{n \notin k}^{N} \mathrm{DM}^k(n) \cdot [[\mathrm{PI}(n) = k]]. \tag{4.8}$$

The error (Equation 4.8) masks out the distance of the incorrectly classified pixels n. $\mathrm{PI}(n)$ denotes the prediction of the model for pixel n, d is the diagonal of the input image, double box brackets mean 1 if the expression inside the brackets is true, 0 otherwise, and $k \in \{R, A\}$.

The figure below shows a comparison of the original Dacup model and the retrained Dacup model with the distance-aware (D-A) loss. Pixels are labeled as anomalous if the softmax output for anomaly class is greater than a selected threshold ($\theta_{original}$ for original Dacup and $\theta_{retrained}$ for retrained Dacup with D-A loss). The figure shows that the number of images with an error greater than 10 dropped with the retrained model, and the number of images with a lower error increased.

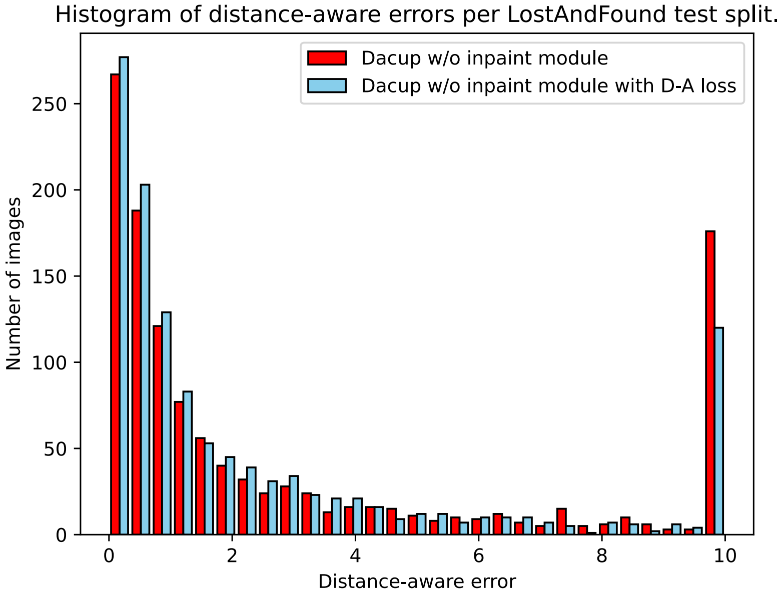


Figure 4.1: Histogram of distance-aware (D-A) errors (Equation 4.8) per LostAndFound test split. Images with error greater than a selected value (in this case, 10) are displayed as their error was 10 due to visualization clarity.

Chapter 5

Results

5.1 Quantitative evaluation

Both models, Dacup and Dacup with D-A loss, were trained for 100 epochs. Standard metrics on which models were evaluated are the False Positive Rate at 95% of True Positive Rate and average precision (AP - area under the Prescision-Recall curve). After the training with the D-A loss, the standard metric did not get significantly worse, as can be observed in the Table 5.1.

Models	LostAndFound		RoadObstacles		RoadAnomaly	
	AP ↑	FPR_{95}	AP ↑	FPR_{95}	AP ↑	FPR_{95}
DACUP	85.24	2.17	86.39	0.17	94.45	6.45
DACUP w/ D-A loss	85.06	3.09	88.31	0.22	94.82	7.16

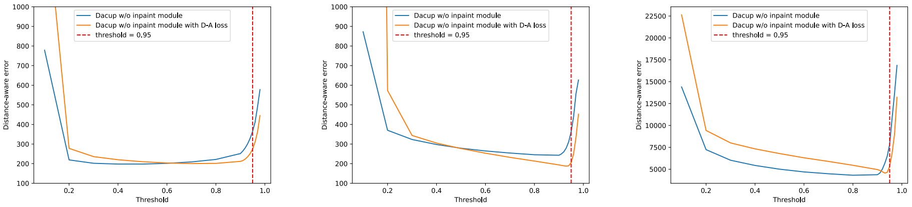
Table 5.1: Comparison of the DACUP model without the inpainting module and the DACUP model without the inpainting module trained with the distance-aware (D-A) loss on the standard metrics - Average Precision and False Positive Rate at 95% True Positive Rate.

Both models were also evaluated using the proposed distance-aware metric (Equation 3.5), with the normalization described in Section 3.3.1. At selected thresholds, the proposed metric is better at retrained model on all observed datasets. The proposed loss forced the Dacup model to make less distant mistakes.

Models	LostAndFound	RoadObstacles	RoadAnomaly
	$\bar{s}\downarrow$	$\bar{s}\downarrow$	$\bar{s}\downarrow$
DACUP	364.97	362.63	7829.31
DACUP with D-A loss	310.72	249.26	6814.47

Table 5.2: Comparison of the DACUP model without the inpainting module and the DACUP model without the inpainting module trained with the distance-aware (D-A) loss on the proposed distance-aware (D-A) metric. D-A metric s is computed for each image and averaged over the dataset.

The following graphs show the dependence of the proposed D-A metric on the threshold for the datasets used. We can observe that for lower thresholds, the original Dacup has better scores of the proposed metric. For higher thresholds, the model trained with the D-A loss outperforms the one not trained with the D-A loss. The retrained model detects anomalies with higher confidence than the original Dacup. On the contrary, it sometimes labels non-anomalous objects as anomalies with low certainty, producing a higher D-A error at lower thresholds. For the selected thresholds, the retrained model is better in all cases.



(a) : Change of the error depending on the threshold for LostAndFound test split.

(b) : Change of the error depending on the threshold for RoadObstacles dataset.

(c) : Change of the error depending on the threshold for RoadAnomaly dataset.

Figure 5.1: Change of the error depending on the evaluated dataset.

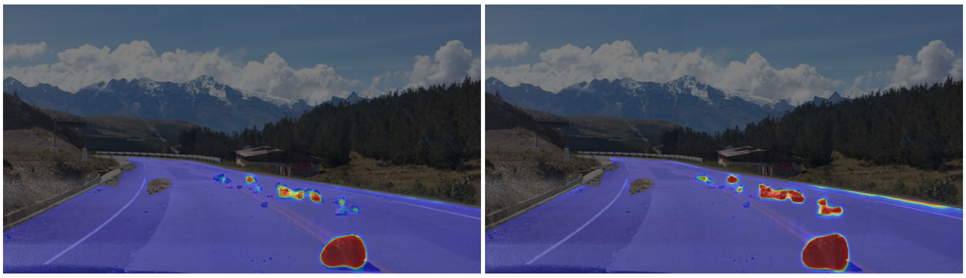
5.2 Visualization of the results

In this section, images from each of the evaluated datasets demonstrating the difference between the Dacup model trained with the D-A loss and the original Dacup model are presented. The model retrained with the D-A loss detects anomalies with higher certainty and gives a higher anomaly score even to the borders of the anomaly object compared to the original model. The original and also the model retrained with D-A loss have problems detecting small and distant anomalies. The colormap used in these visualizations is jet [12].

In the figures 5.2, 5.3, and 5.8 is shown that the anomalous objects detected by the original model with low confidence are detected with higher confidence by the model retrained with the D-A loss regularization. Even the previously undetected anomalies are detected by the retrained model. The retrained model penalizes highly confident predictions of the incorrect class proportionally to the distance from the correct class.

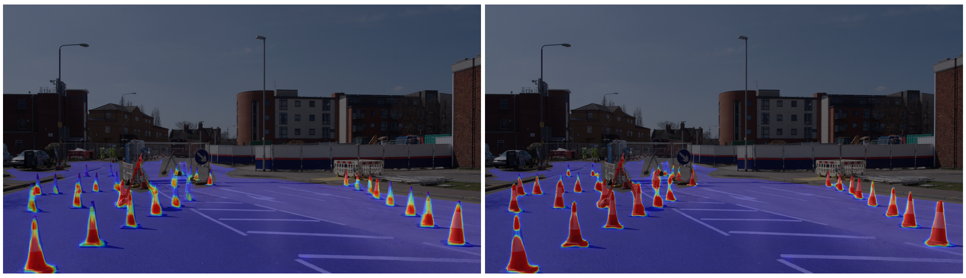
Figures 5.4, 5.5, and 5.9 show that the retrained model refined the boundaries of anomalous objects compared to the original model. Boundaries are recognized with higher confidence. The retrained model minimizes the distance between the pixels with a high softmax road score and the true road pixels.

Figures 5.6, and 5.7 demonstrate that distant false positive predictions made by the original model were decreased or totally removed by the model retrained with the D-A loss.



(a) : Prediction by the original model. **(b) :** Prediction by the retrained model.

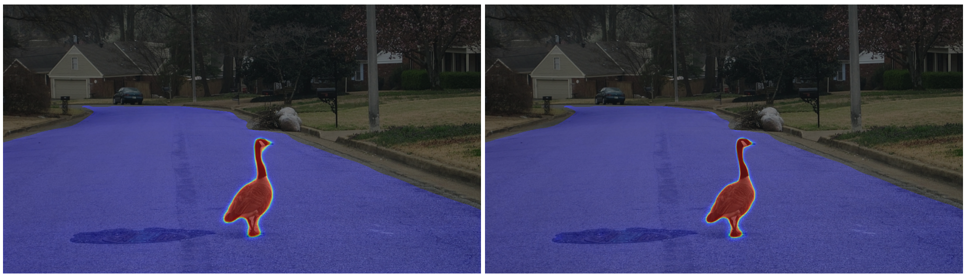
Figure 5.2: The retrained model detects rocks on the road with higher certainty. It also detects rocks undetected by the original model. On the other hand, edges of the road have higher anomaly score but still lower than the selected threshold. The image is from RA dataset.



(a) : Prediction by the original model.

(b) : Prediction by the retrained model.

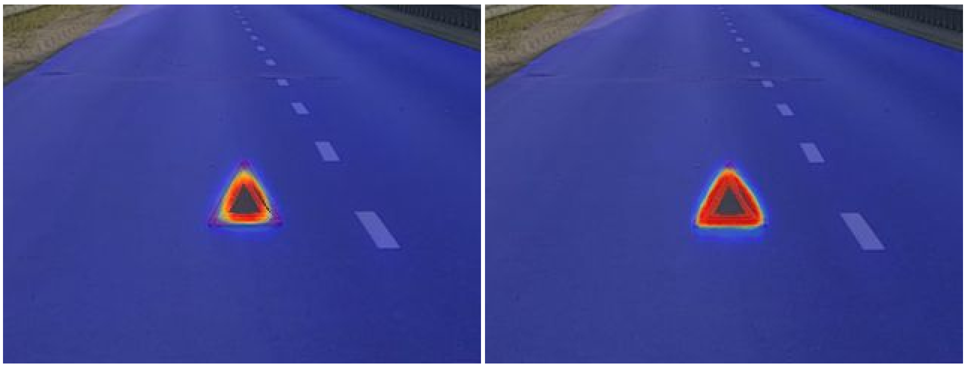
Figure 5.3: The retrained model better detects the entire cone area. The image is from RA dataset.



(a) : Prediction by the original model.

(b) : Prediction by the retrained model.

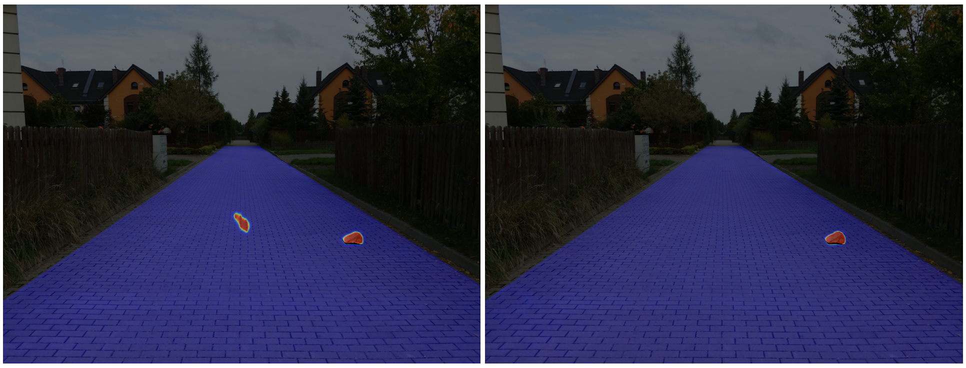
Figure 5.4: Boundaries of the anomaly are refined by the retrained model, and the model is more confident in them. The image is from RA dataset.



(a) : Prediction by the original model.

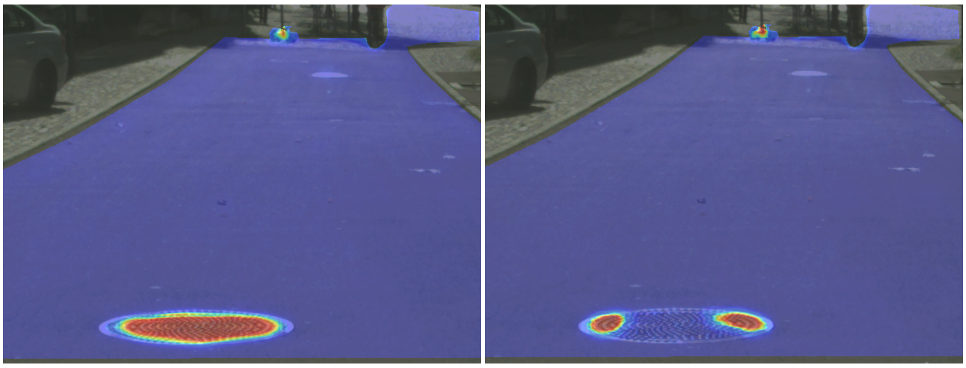
(b) : Prediction by the retrained model.

Figure 5.5: The retrained model better detects the boundaries of the anomalous object. The image is from RO dataset.



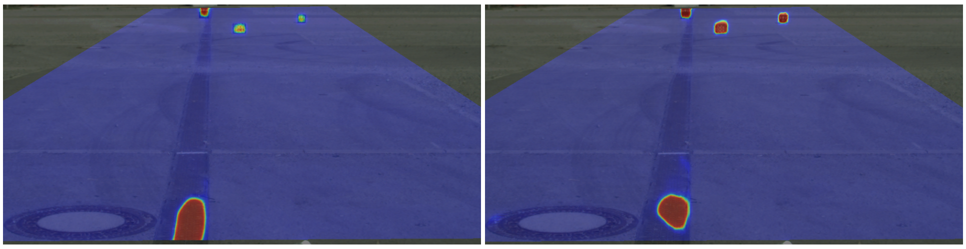
(a) : Prediction by the original model. **(b) :** Prediction by the retrained model.

Figure 5.6: Anomaly is in the right part of an image. The original model detects a false positive anomaly in the left part of an image, which is not falsely detected by the retrained model. The image is from RO dataset.



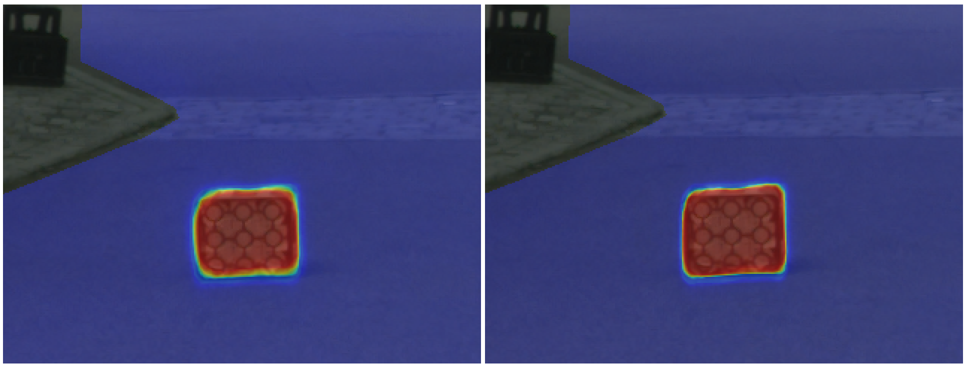
(a) : Prediction by the original model. **(b) :** Prediction by the retrained model.

Figure 5.7: The distant anomaly is not detected very well by either of the models. The sewer is falsely detected as an anomaly by the original model, and the retrained model falsely detects a smaller part of the sewer, which is far from the true anomaly. The image is from LaF dataset.



(a) : Prediction by the original model. **(b) :** Prediction by the retrained model.

Figure 5.8: Three distant anomalous objects are detected with higher confidence by the retrained model. The image is from LaF dataset.

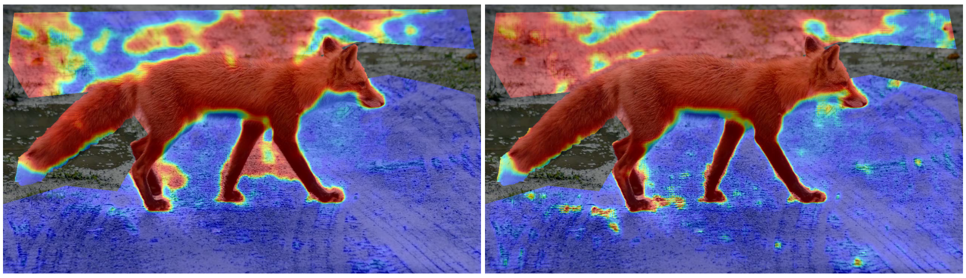


(a) : Prediction by the original model. **(b) :** Prediction by the retrained model.

Figure 5.9: The retrained model detects the boundaries of the anomalous object with higher confidence. The image is from LaF dataset.

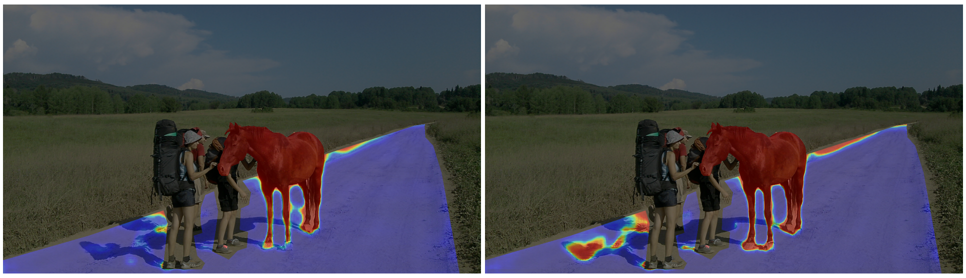
5.2.1 Failure cases

In the following figures 5.10, and 5.11, failure cases of the model retrained with the D-A loss are presented. In both cases, the retrained model makes more false positive predictions than the original model. In the figure 5.10, the legs of the fox are detected more precisely. The increased number of pixels with incorrectly high anomaly score may be caused by the untypical road surface. In the figure 5.11, the legs of the horse are also detected more precisely. However, the number of false positive predictions has increased. The retrained model mistakenly classifies shadows on the road as anomalies.



(a) : Prediction by the original model. **(b) :** Prediction by the retrained model.

Figure 5.10: The retrained model detects the legs of the fox more precisely. On the other hand, more false positive predictions are made. The image is from RA dataset.



(a) : Prediction by the original model. **(b) :** Prediction by the retrained model.

Figure 5.11: The retrained model detects the legs of the horse more precisely. The number of false positive predictions has increased. The image is from RA dataset.

Chapter 6

Conclusion

In industrial applications such as visual inspection or autonomous driving, the spatial distribution of the error plays a key role. The goal of this thesis was to propose the distance-aware evaluation metric for semantic segmentation and to propose the distance-aware regularization loss function for training the semantic segmentation model so the errors of the model were more compact around the anomalous object, making fewer distant errors.

The distance-aware regularization metric, inspired by the standard metrics, which takes into account the spatial distribution of the error, was proposed. Based on the purpose of the application, it can penalize certain type of error more than another. The proposed metric is complementary to standard metrics and provides a different view of the performance of the semantic segmentation model.

In order to retrain the segmentation model to make less distant errors, the distance-aware loss function was proposed. The loss was incorporated into the state-of-the-art model Dacup, which uses the negative log-likelihood loss. The Dacup model retrained with the distant-aware regularization loss retained the performance of the standard metrics and improved the distance-aware evaluation metric. The retrained model performs better at the boundaries of the anomaly and classifies them with higher confidence.

Appendix A

List of Notation

Symbol	Meaning
D-A	Distance-aware
DAG	Direct Acyclic Graph
RA	RoadAnomaly dataset
RO	RoadObstacles dataset
LaF	LostAndFound dataset
AP	Average Precision
TPR	True Positive Rate
FPR	False Positive Rate
AI	Artifical Intelligence
DL	Deep Learning
OoD	Out-of-Distribution
GAN	Generative Adversarial Network
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network

Appendix B

Bibliography

[1] Alexander Kirillov, Kaiming He, Ross B. Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. *CoRR*, abs/1801.00868, 2018.

[2] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *CoRR*, abs/1411.4038, 2014.

[3] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. *CoRR*, abs/1505.04366, 2015.

[4] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *CoRR*, abs/1511.00561, 2015.

[5] Francesco Visin, Kyle Kastner, Aaron C. Courville, Yoshua Bengio, Matteo Matteucci, and KyungHyun Cho. Reseg: A recurrent neural network for object segmentation. *CoRR*, abs/1511.07053, 2015.

[6] Heng Fan and Haibin Ling. Dense recurrent neural networks for scene labeling. *CoRR*, abs/1801.06831, 2018.

[7] Pauline Luc, Camille Couprie, Soumith Chintala, and Jakob Verbeek. Semantic segmentation using adversarial networks. *CoRR*, abs/1611.08408, 2016.

[8] Nasim Souly, Concetto Spampinato, and Mubarak Shah. Semi and weakly supervised semantic segmentation using generative adversarial network. *CoRR*, abs/1703.09695, 2017.

[9] Yingda Xia, Yi Zhang, Fengze Liu, Wei Shen, and Alan L. Yuille. Synthesize then compare: Detecting failures and anomalies for semantic segmentation. *CoRR*, abs/2003.08440, 2020.

[10] Krzysztof Lis, Sina Honari, Pascal Fua, and Mathieu Salzmann. Detecting road obstacles by erasing them. *CoRR*, abs/2012.13633, 2020.

[11] Peter Pinggera, Sebastian Ramos, Stefan Gehrig, Uwe Franke, Carsten Rother, and Rudolf Mester. Lost and found: Detecting small road hazards for self-driving vehicles. *CoRR*, abs/1609.04653, 2016.

[12] Jet colormap array - MATLAB jet — mathworks.com. `https://www.mathworks.com/help/matlab/ref/jet.html`. [Accessed 18-May-2023].

[13] Shervin Minaee, Yuri Boykov, Fatih Porikli, Antonio Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. Image segmentation using deep learning: A survey. *CoRR*, abs/2001.05566, 2020.

[14] Shijie Hao, Yuan Zhou, and Yanrong Guo. A brief survey on semantic segmentation with deep learning. *Neurocomputing*, 406:302–321, 2020.

[15] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2014.

[16] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *CoRR*, abs/1606.00915, 2016.

[17] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *CoRR*, abs/1706.05587, 2017.

[18] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. *CoRR*, abs/1802.02611, 2018.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[20] Petra Bevandic, Ivan Kreso, Marin Orsic, and Sinisa Segvic. Simultaneous semantic segmentation and outlier detection in presence of domain shift. *CoRR*, abs/1908.01098, 2019.

[21] Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. Densely connected convolutional networks. *CoRR*, abs/1608.06993, 2016.

[22] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

[23] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows, 2021.

[24] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything, 2023.

[25] Yibiao Zhao and Song-chun Zhu. Image parsing with stochastic scene grammar. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011.

[26] Stella X. Yu, Hao Zhang, and Jitendra Malik. Inferring spatial layout from a single image via depth-ordered grouping. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–7, 2008.

[27] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

[28] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014.

[29] Alex Kendall, Vijay Badrinarayanan, and Roberto Cipolla. Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. *CoRR*, abs/1511.02680, 2015.

[30] Jun Fu, Jing Liu, Yuhang Wang, and Hanqing Lu. Stacked deconvolutional network for semantic segmentation. *CoRR*, abs/1708.04943, 2017.

[31] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015.

[32] Damien Fourure, Rémi Emonet, Élisa Fromont, Damien Muselet, Alain Trémeau, and Christian Wolf. Residual conv-deconv grid network for semantic segmentation. *CoRR*, abs/1707.07958, 2017.

[33] Francesco Visin, Kyle Kastner, Kyunghyun Cho, Matteo Matteucci, Aaron C. Courville, and Yoshua Bengio. Renet: A recurrent neural network based alternative to convolutional networks. *CoRR*, abs/1505.00393, 2015.

[34] Bing Shuai, Zhen Zuo, Gang Wang, and Bing Wang. Dag-recurrent neural networks for scene labeling. *CoRR*, abs/1509.00552, 2015.

[35] Wonmin Byeon, Thomas M. Breuel, Federico Raue, and Marcus Liwicki. Scene labeling with lstm recurrent neural networks. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3547–3555, 2015.

[36] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

[37] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *CoRR*, abs/1703.10593, 2017.

[38] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S. Huang. Generative image inpainting with contextual attention. *CoRR*, abs/1801.07892, 2018.

[39] Scott E. Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. *CoRR*, abs/1605.05396, 2016.

[40] Wei-Chih Hung, Yi-Hsuan Tsai, Yan-Ting Liou, Yen-Yu Lin, and Ming-Hsuan Yang. Adversarial learning for semi-supervised semantic segmentation. *CoRR*, abs/1802.07934, 2018.

[41] Segan: Adversarial network with multi-scale l 1 loss for medical image segmentation. *Neuroinformatics*, 16(3-4):383–392, 2018.

[42] Pim Moeskops, Mitko Veta, Maxime W. Lafarge, Koen A. J. Eppenhof, and Josien P. W. Pluim. Adversarial training and dilated convolutions for brain MRI segmentation. *CoRR*, abs/1707.03195, 2017.

[43] Mina Rezaei, Konstantin Harmuth, Willi Gierke, Thomas Kellermeier, Martin Fischer, Haojin Yang, and Christoph Meinel. Conditional adversarial network for semantic segmentation of brain tumor. *CoRR*, abs/1708.05227, 2017.

[44] Mina Rezaei, Haojin Yang, and Christoph Meinel. Conditional generative refinement adversarial networks for unbalanced medical image semantic segmentation. *CoRR*, abs/1810.03871, 2018.

[45] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *CoRR*, abs/1411.1784, 2014.

[46] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs, 2014.

[47] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks, 2018.

[48] Shiyu Liang, Yixuan Li, and R. Srikant. Principled detection of out-of-distribution examples in neural networks. *CoRR*, abs/1706.02690, 2017.

[49] Shiyu Liang, Yixuan Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks, 2020.

[50] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *CoRR*, abs/1610.02136, 2016.

[51] Terrance DeVries and Graham W. Taylor. Learning confidence for out-of-distribution detection in neural networks, 2018.

[52] Dan Hendrycks, Mantas Mazeika, and Thomas G. Dietterich. Deep anomaly detection with outlier exposure. *CoRR*, abs/1812.04606, 2018.

[53] Robin Chan, Matthias Rottmann, and Hanno Gottschalk. Entropy maximization and meta classification for out-of-distribution detection in semantic segmentation. *CoRR*, abs/2012.06575, 2020.

[54] Li Zhang, Xiangtai Li, Anurag Arnab, Kuiyuan Yang, Yunhai Tong, and Philip H. S. Torr. Dual graph convolutional network for semantic segmentation. *CoRR*, abs/1909.06121, 2019.

[55] Tomáš Vojíř and Jiří Matas. Image-consistent detection of road anomalies as unpredictable patches. In *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 5480–5489, 2023.

[56] Asim Munawar, Phongtharin Vinayavekhin, and Giovanni De Magistris. Limiting the reconstruction capability of generative neural network using negative learning. *CoRR*, abs/1708.08985, 2017.

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

[58] Giancarlo Di Biase, Hermann Blum, Roland Siegwart, and Cesar Cadena. Pixel-wise anomaly detection in complex driving scenes. *CoRR*, abs/2103.05445, 2021.

[59] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88:303–308, 2009.

[60] Bharath Hariharan, Pablo Arbeláez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *2011 International Conference on Computer Vision*, pages 991–998, 2011.

[61] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 891–898, 2014.

[62] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014.

[63] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ADE20K dataset. *CoRR*, abs/1608.05442, 2016.

[64] A Geiger, P Lenz, C Stiller, and R Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[65] Gabriel J. Brostow, Julien Fauqueur, and Roberto Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, 30(2):88–97, 2009. Video-based Object and Event Analysis.

[66] Paul Sturgess, Alahari Karteek, Lubor Ladicky, and Philip H. S. Torr. Combining appearance and structure from motion features for road scene understanding. In *British Machine Vision Conference*, 2009.

[67] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. *CoRR*, abs/1604.01685, 2016.

[68] Hermann Blum, Paul-Edouard Sarlin, Juan Nieto, Roland Siegwart, and Cesar Cadena. Fishyscapes: A benchmark for safe semantic segmentation in autonomous driving. pages 2403–2412, 10 2019.

[69] Robin Chan, Krzysztof Lis, Svenja Uhlemeyer, Hermann Blum, Sina Honari, Roland Siegwart, Mathieu Salzmann, Pascal Fua, and Matthias Rottmann. Segmentmeifyoucan: A benchmark for anomaly segmentation. *CoRR*, abs/2104.14812, 2021.

[70] Krzysztof Lis, Krishna K. Nakka, Pascal Fua, and Mathieu Salzmann. Detecting the unexpected via image resynthesis. *CoRR*, abs/1904.07595, 2019.

[71] Dan Hendrycks, Steven Basart, Mantas Mazeika, Andy Zou, Joe Kwon, Mohammadreza Mostajabi, Jacob Steinhardt, and Dawn Song. Scaling out-of-distribution detection for real-world settings, 2022.

[72] 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.

[73] Bharathkumar Ramachandra and Michael Jones. Street scene: A new dataset and evaluation protocol for video anomaly detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, March 2020.