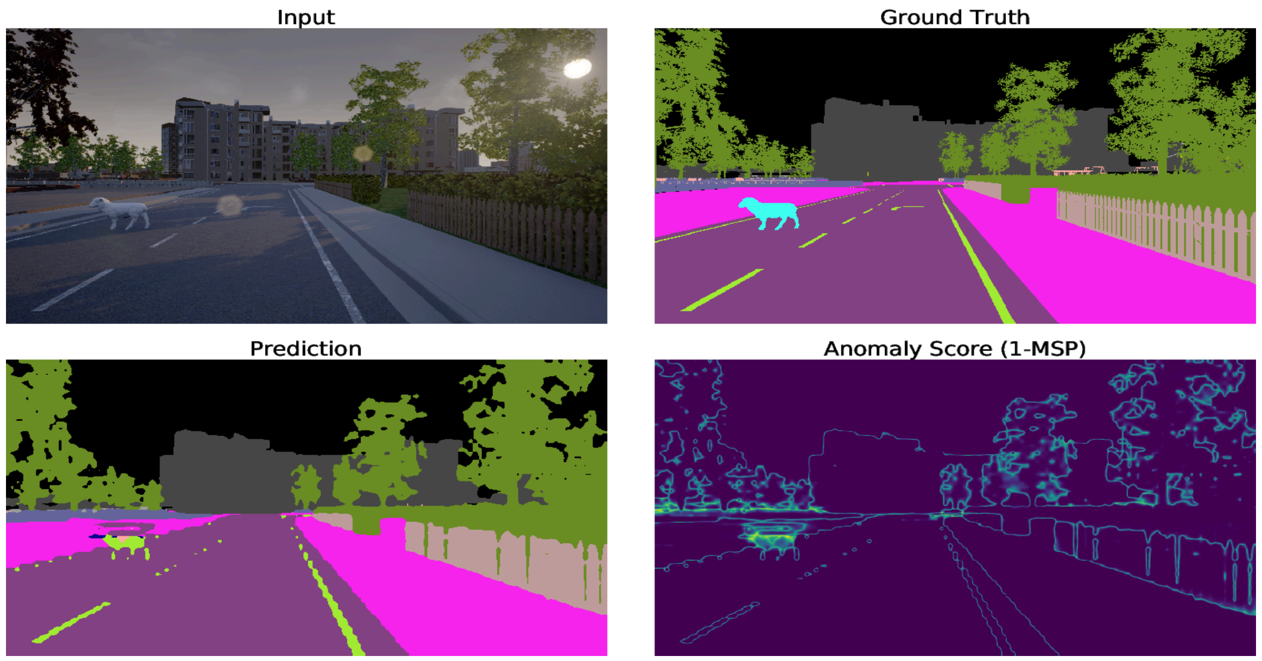
Real-Time Anomaly Segmentation for Road Scenes

TA: Niccolò Cavagnero (Email: niccolo.cavagnero@polito.it)



Existing deep networks, when deployed in open-world settings, perform poorly on Unknown/Anomaly/Out-of-distribution(OoD) objects that were not present during the training. Detecting OoD objects becomes critical for autonomous driving applications and branches of computer vision problems such as continual learning and open-world problems.

In this project, your task is to build **tiny anomaly segmentation** models to segment anomalies that could be deployed in real-time. The goal is for the models to be able to fit in small devices, which represents a realistic memory constraint for an edge application using a smart camera with some small onboard processing capacity.

Dataset

- For training the model, you have to use the Cityscapes Dataset.
- For testing the anomaly segmentation model: Road Anomaly, Road Obstacle, and Fishyscapes dataset. All testing images are provided here [Link].

Starting Code Repository: Github Link

Goals

1. Get acquainted with the task of anomaly segmentation and understand the anomaly segmentation dataset and its complexities.
2. Run initial experiments with a few baseline models for anomaly segmentation.
3. Analyze the anomaly segmentation results by training a void classifier on different semantic segmentation architectures.
4. Propose your extensions to reduce anomaly segmentation model size while maintaining anomaly segmentation performance. You are free to propose your ideas and choose what you want to focus on.

Project Steps

1. **Study the literature and get familiar with the task**

   As a preliminary step to get familiar with the task of anomaly segmentation **[5,6]**, start by studying the problem and dataset and read about some popular lightweight models for semantic segmentation **[1,2,3,4]**. These models will be the basis for the experiments you will carry out in the project, so make sure you understand how they work.

2. **Baselines**

   A. The GitHub repository provides a pre-trained ERF-Net **[7]** model trained on Cityscapes. The students must perform various anomaly inferences using the pre-trained model and anomaly segmentation test dataset provided. The code for MSP has already been provided. Resources for max-entropy and max-logit can be found here [i] [ii].

		SMIYC RA-21		SMIYC RO-21		FS L&F		FS Static		Road Anomaly	
Method	**mIoU**	**AuPRC**	**FPR95**	**AuPRC**	**FPR95**	**AuPRC**	**FPR95**	**AuPRC**	**FPR95**	**AuPRC**	**FPR95**
MSP											
MaxLogit											
Max Entropy											

B. Temperature scaling: is a method for confidence calibration for any classifier which could result in improving anomaly segmentation capabilities of a network. Hence, we use this technique while segmenting anomalies during inference. The goal of this part is to find the value of temperature that gives the best anomaly segmentation results.

		SMIYC RA-21		SMIYC RO-21		FS L&F		FS Static		Road Anomaly	
Method	**mIoU**	**AuPRC**	**FPR95**	**AuPRC**	**FPR95**	**AuPRC**	**FPR95**	**AuPRC**	**FPR95**	**AuPRC**	**FPR95**
MSP											
MSP(t = 0.5)											
MSP(t = 0.75)											
MSP(t = 1.1)											
MSP (best t)											

3. Void Classifier

The cityscapes dataset consists of 19 known category classes + void (background). In this section, we will assume the void class as an anomaly and train the ENet and BiSeNet networks. Then, we will perform the anomaly inference by only selecting the output of the Void class.

		SMIYC RA-21		SMIYC RO-21		FS L&F		FS Static		Road Anomaly	
Network	**mIoU**	**AuPRC**	**FPR95**	**AuPRC**	**FPR95**	**AuPRC**	**FPR95**	**AuPRC**	**FPR95**	**AuPRC**	**FPR95**
ENet											
ERF-Net											
BiSeNet											

4. Project Extension

It is now time for you to put into practice what you have learned so far and propose additional analyses and extensions. The extensions may have different goals, e.g., add a new analysis that investigates some problem, improve the anomaly performance of the model, or reduce the size of the model while trying to keep the same accuracy. Here are a few examples of possible extensions, but feel free to propose your own

(Choose any one extension carefully as some of the extensions could

require more GPU computing):

a) **Analyze the effect of different metrics:** Anomaly segmentation performance can be improved by adopting better metrics:

   i) Compute the mean and covariance of training samples (on Cityscapes) and then a mahalanobis distance-based confidence score. (ref: A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks, eq. 1 and 2.)

b) **Analyze the effect of additional training dataset**: Anomaly segmentation capabilities could be improved by using additional semantic segmentation datasets such as Mapillary. The analysis can be shown in the following ways:

   i) Sequential training of model on cityscapes followed by Mapillary or vice-versa.

   ii) Mixed training dataset: Mix the Mapillary and Cityscapes dataset and then train the model.

c) **Effect of Training Loss function**: Analyze the effect of the training model along with losses that are specifically made for anomaly detection.

   i) Enhanced Isotropy Maximization Loss.

   ii) Logit Normalization loss.

   iii) Analyze the effect of these losses when trained jointly with focal loss and cross-entropy loss.

d) **Self-supervised pre-trained models:** Self-supervised pre-trained models provide strong generalization priors. In this part, the student needs to perform the following steps:

   i) Replace the backbone of the segmentation network with self-supervised pre-trained models of ResNet-50: Barlow-Twins, MoCo-V1/V2, and SimSiam.

   ii) Analyze the effect of freezing the backbone vs fine-tuning the backbone.

iii) Why some pre-trained models are better at anomaly segmentation tasks?

e) **Foundational models (GPU Intensive):** Foundational models such as segment anything, DINO-v2 have pushed the generalization boundaries. In this part, the student is supposed to follow the following steps:

i) Replace the backbone of the segmentation network with DINO, DINO-v2 weights (use ViT-S/14).

ii) Analyze the effect of freezing backbone, fine-tuning, linear-probing, and LP-FT the backbone on anomaly segmentation performance.

iii) ***(Optional)*** Perform all the above experiments with ViT-B backbone of segment-anything or CLIP.

f) **Pruning and Quantization:** Another direction could be to try and reduce the model size and latency by employing quantization and pruning methods to the trained models. [Link] [Link]**.** The experiment result should report mIOU, FLOPS, # of trainable model parameters, and anomaly segmentation performance.

g) Be creative and come up with your ideas and proposals. These extensions don't need to produce very good results as long as you provide good motivations for why you are trying something.

References

1. BiSeNet: Bilateral Segmentation Network for Real-time Semantic Segmentation
2. BiSeNet V2: Bilateral Network with Guided Aggregation for Real-time Semantic Segmentation
3. ICNet for Real-Time Semantic Segmentation on High-Resolution Images
4. Enet: A deep neural network architecture for real-time semantic segmentation
5. SegmentMeIfYouCan: A Benchmark for Anomaly Segmentation
6. The Fishyscapes Benchmark Anomaly Detection for Semantic Segmentation
7. ERFNet: Efficient Residual Factorized ConvNet for Real-Time Semantic Segmentation