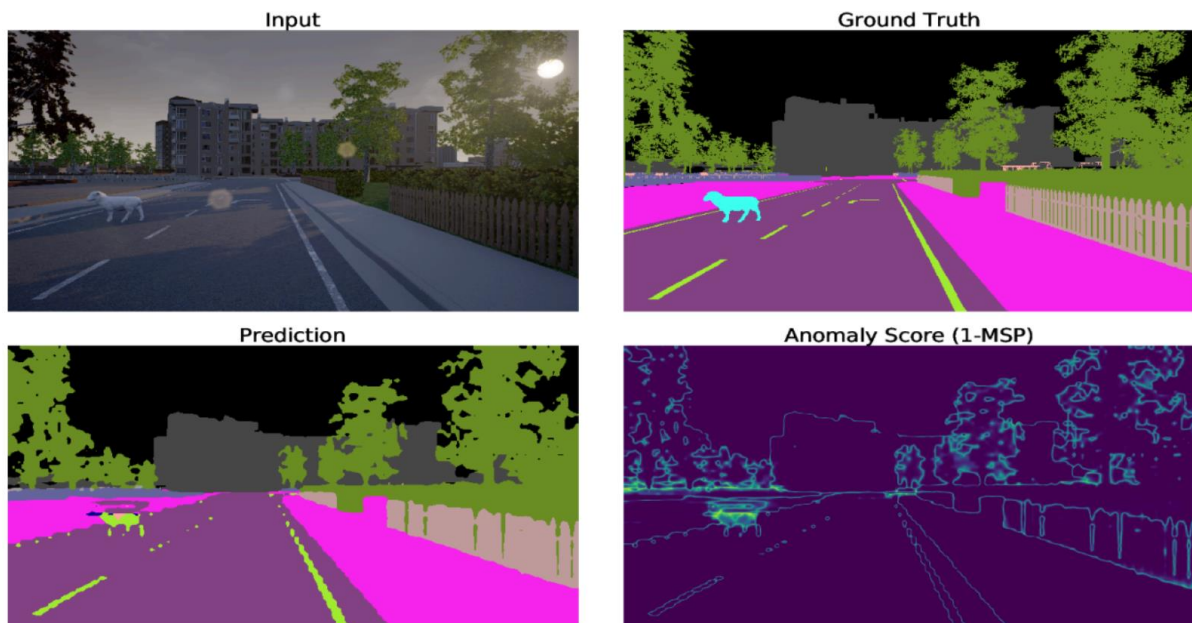


Mask Architecture Anomaly Segmentation for Road Scenes

TA: Alessandro Marinai (Email: alessandro.marinai@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, train, and test your own **anomaly segmentation model** to segment anomalies on road scenes.

Starting code repository: [Github Repository](#)

Project steps

1. Study the mask architecture literature for semantic segmentation

Study a very simple and efficient model for real-time semantic segmentation ERFNet[10].

2. Study the mask architecture literature for semantic segmentation

Study the core concepts behind mask architectures and the first [MaskFormer](#) model [3], understand the key changes in [Mask2former](#), its evolution [4] (not mandatory but important) and understand the use of [DINOv2](#) [5] in the [EoMT](#) model [6].

3. Understand anomaly segmentation task and post-hoc methods

Get familiar with the [task of anomaly segmentation](#), start by studying the problem and datasets [1,2] and then read up the most important anomaly segmentation post-hoc methods [7, 8].

4. Pixel-based baselines

To develop the first baselines let's start from a more conventional approach of using pixel-based approach and apply post-hoc methods using these models. In the [eval folder](#) you can find the method for computing MSP using a ERFNet pretrained model on some anomaly segmentation validation sets (the weights are also on github). Download the datasets [here](#).

The post-hoc methods you need to run inference for are: MSP (already implemented), Max Logit, Max Entropy.

Resources for max-entropy and max-logit can be found here [i] [[ii].

5. Mask-based baselines

In this step you will evaluate the eomt model on the same anomaly segmentation datasets. In the repo you can find the [eomt repository](#) with all you will need for the project.

Adapt the eval code that you used in the previous step to use EoMT pretrained model. Keep in mind that we are switching into a Mask Architecture so the output of the model will be different.

The post-hoc methods you need to run inference for are: MSP, Max Logit, Max Entropy and RbA, that can now be applied since you are using a mask Architecture.

Resources for RbA can be found here [iii].

The baselines results should be reported in a table like this:

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

To get another baseline try [Temperature scaling](#).

It 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.

PRO TIP: To make this fast, you can save the model predictions on the datasets and then try different temperatures with the saved logits. In that way you will need to run the model forward pass once.

		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)											

6. Project Extensions

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 problems, 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:

- **Fine-tune the model with a training loss specifically made for anomaly segmentation**

You should train EoMT on [Cityscapes](#) Dataset (starting code is in the eomt folder) but you need to modify it.

Some possible losses:

- [Enhanced Isotropy Maximization Loss](#).
- [Logit Normalization loss](#).

- **Fine-tune with Outlier Exposure**

Cut-paste some objects from other datasets on [Cityscapes](#) (ex. COCO dataset's objects) and use these pasted outliers to enhance anomaly segmentation performances.

- In the RbA paper, they provide a method with Outlier Exposure, implement it, and play with the hyperparameters.
- Think of your own Outlier Exposure training pipeline, it doesn't need to perform incredibly as long as you can motivate why you tried that technique.

- **Your ideas**

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 motivation for why you are trying something.

Ideas must be agreed with the TA before starting to work on them.

N.B. In case you decide to fine-tune the model, it is a great idea to use AMP (automatic mixed precision) to reduce training time. A good first experiment to start out with is to finetune just the prediction head. Then you can gradually unfreeze the last layers and compare (check Colab time constraints).

Another idea can be to finetune just the learned queries and freeze the remaining part of the model.

Another technique that could be useful to be able to finetune without many resources is LoRA [9].

References:

1. [SegmentMelfYouCan: A Benchmark for Anomaly Segmentation](#)
2. [The Fishyscapes Benchmark Anomaly Detection for Semantic Segmentation](#)
3. [Per-Pixel Classification is Not All You Need for Semantic Segmentation](#), Bowen Cheng et. Al.
4. [Mask2Former: Masked-attention Mask Transformer for Universal Image Segmentation \(CVPR 2022\)](#)
5. [DINOv2: Learning Robust Visual Features without Supervision](#)
6. [Your ViT is Secretly an Image Segmentation Model \(CVPR 2025\)](#)
7. [RbA: Segmenting Unknown Regions Rejected by All](#)
8. [Scaling Out-of-Distribution Detection for Real-World Settings](#)
9. [LoRA: Low-Rank Adaptation of Large Language Models](#)
10. [ERFNet: Efficient Residual Factorized ConvNet for Real-Time Semantic Segmentation](#)