Real-Time Anomaly Segmentation for Road Scene

https://github.com/bred91/Advanced-Machine-Learning-Project

Masetta Dimitri
306130

Pane Raffaele
305485

Tetamo Ruben
317569

Politecnico di Torino, 2024

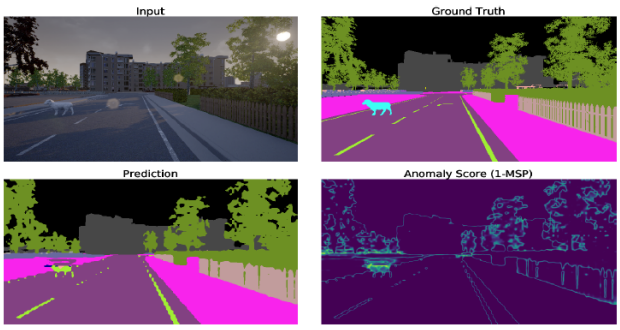


Figure 1. When exposed to an object type unseen during training, a state-of-the-art semantic segmentation model predicts familiar labels(road strip) with high confidence.

Abstract

Existing deep networks excel on semantic segmentation tasks, but when deployed in open-world settings, perform poorly on Out-of-distribution objects that were not present during the training. However, detecting Out-of-distribution objects becomes more critical for autonomous driving applications, where the open-world scenario provide continuously new objects to identify. To address this applications niches we analyze and compare the performance of three state of the art models, capable of a good trade-off between high accuracy and live inference. Furthermore we analyze the effect of training model along with losses that are specifically made for anomaly detection to further increase the accuracy. A comprehensive set of experiments show that ERFNet, integrated with Cross Entropy, outperforms most baselines when tested against five different datasets.

1. Introduction

New deep learning technologies have enable astonishing results in the accuracy of semantic segmentation tasks, that aim to assign a semantic labels to each pixel of the input image. In recent years, the availability of large datasets and computationally-powerful machines have facilitated the development of deep network able to outperform previous state of the art architecture; the rising needs of deployment, of such network, on the edge, have led to an increase of the computational efficiency, so that they can satisfy the requirement of real-time operation on low-power embedded devices, and by so new network architecture like [1,3,6] are been defined.

Yet, all these algorithms are usually trained on a closed set of semantic classes [5, 7], and as such they are ill-equipped to handle class of objects unseen during the training. However, recent interest in self-driving vehicles has created a strong need for neural network capable of detecting and localizing such out-of-distribution object, as their detection is safety critical in the perception of autonomous vehicles. The real world is uncontrollable, and a mis-identification of an input, by an autonomous agents can lead to disastrous consequences[6 fish]. Thus, to reach full autonomy while ensuring safety and reliability, arises the need to be able to estimate uncertainty on the semantic segmentation in case of outliers. As illustrated in Fig.1, deep convolutional neural networks (CNNs) response, to out-of-distribution object is unpredictable, as the algorithm assign to the outliers pixels the semantic labels that give most confidence, without realizing that in reality it is a completely different class.

Previous research on out-of-distribution detection is mainly focused on developing methods on simple images datasets, like MNIST or CIFAR-10 [4, 8, 9, 15–20]. Therefore the main objective of this paper is to analyze the feasibility of using as backbone of-the-shelf semantic segmentation models to segment anomalies, that could be deployed in small devices capable to do inference in real time. Has been also considered the influence of the uses of loss functions, that are specifically made for anomaly detection, as can be seen in SEC. After training the models on Cityscape,

dataset of urban scenarios, a comprehensive set of experiment against five different anomaly dataset proves the feasibility of using such model to accurately perform anomaly segmentation in real time on small devices.

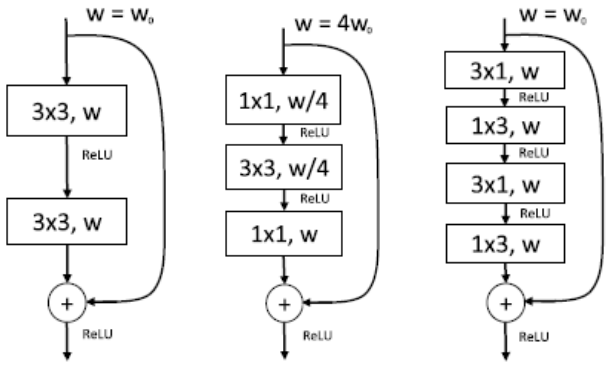
2. Related Work

2.1. ERFNet

ERFNet [6] (Efficient Residual Factorized Network) is a convolutional neural network optimized for accurate real time semantic segmentation. As designated by the name, the core element of the architecture is a novel factorized residual layer design, that facilitates learning and significantly decrease the degradation problem in deep architecture [13]. The original work in [13] proposes a non-bottleneck residual layer design with two 3x3 convolutions, Fig.2a and a bottleneck design with one 3x3 and two 1x1 convolutions, Fig.2b, both with similar amount of parameters, but the second that requires less computational resources; instead of that, [6] proposes a new implementation of the residual layer that make use of only 1D factorized convolutions, to accelerate and reduce the parameters of the original non-bottleneck layer, Fig.2c. This module is computational less expensive and has less parameters than the bottleneck version, while keeping similar accuracy to the non-bottleneck one. The proposed block is thus stacked sequentially to build an encoder-decoder architecture like SegNet [21] and ENet [1], which produces semantic segmentation in the same resolution as the input by first downsampling and then upsampling the information, as shown on Tab.1. In contrast to SegNet [21] that uses a symmetric encoder-decoder architecture, it is used, like ENet [1], a small decoder whose only purpose is to upsample the encoder's output. By using all of this strategies ERFNet is able to obtain a very high segmentation accuracy, while keeping top computational efficiency.

2.2. ENet

ENet [1] (Efficient Neural Network) provide an asymmetric encoder-decoder architecture characterized by compactness and computation efficiency, even for mobile applications. The initial stage, implemented by ENet, contains a single block, that is heavily inspired by [2], which has as its main purpose to translate the input image, with its three channels, in 16 feature maps, as presented in Fig.3a. After the initial stage, the main architecture can therefore being summarized in 2 main phases, encoding and decoding. The encoding phase is characterized by a bottleneck block, shown on Fig.3b, utilizing a similar structure to the one proposed by ResNet [13], in which there is a main branch containing max pooling and a padding layer that merge through an element-wise addition with an extension branch; the extensions branch contains three convolutional



(a) Non-bottleneck module (b) Bottleneck module (c) Non-bottleneck-1D module

Figure 2. The two residual layers originally proposed in [6 erfnet] Non-bottleneck and Bottleneck, and the proposed design Non-bottleneck-1D. (a) Non-bottleneck. (b) Bottleneck. (c) Non-bottleneck-1D.

	Layer	Type	Output size
ENCODER	1	**Downsampler block**	16x512x256
	2	**Downsampler block**	64x256x128
	3-7	5 x **Non-bt-1D**	64x256x128
	8	**Downsampler block**	128x128x64
	9	**Non-bt-1D** (dilated 2)	128x128x64
	10	**Non-bt-1D** (dilated 4)	128x128x64
	11	**Non-bt-1D** (dilated 8)	128x128x64
	12	**Non-bt-1D** (dilated 16)	128x128x64
	13	**Non-bt-1D** (dilated 2)	128x128x64
	14	**Non-bt-1D** (dilated 4)	128x128x64
	15	**Non-bt-1D** (dilated 8)	128x128x64
	16	**Non-bt-1D** (dilated 16)	128x128x64
DECODER	17	**Deconvolution** (upsampling)	64x256x128
	18-19	2 x **Non-bt-1D**	64x256x128
	20	**Deconvolution** (upsampling)	16x512x256
	21-22	2 x **Non-bt-1D**	16x512x256
	23	**Deconvolution** (upsampling)	Cx1024x512

Table 1. ERFNet architecture

layer, the first for dimensionality reduction, then the main convolution and an expansion layer, followed by a Spatial Dropout [12] used as regularization layer. This sequence of layer allows to have greater performance, because the main convolution work on smaller dimensions. After each convolution is placed a Batch Normalization [11] and then a PReLU [14] layer. The decoding phase aims to be of low computational impact, by rapidly upsample the encoder's output, as seen above with ERFNet [6], the only main difference is a final full convolution that takes a considerable portion of the decoder processing time, that map the output on the C output channels. The architecture layer disposal

	Layer	Type	Output size
IN	1	**Initial block**	16x256x256
ENCODER	2	**Bt-module** (downsampling)	64x128x128
	3-6	**Bt-module**	64x128x128
	7	**Bt-module** (downsampling)	128x64x64
	8	**Bt-module**	128x64x64
	9	**Bt-module** (dilated 2)	128x64x64
	10	**Bt-module** (asymmetric 5)	128x64x64
	11	**Bt-module** (dilated 4)	128x64x64
	12	**Bt-module**	128x64x64
	13	**Bt-module** (dilated 8)	128x64x64
	14	**Bt-module** (asymmetric 5)	128x64x64
	15	**Bt-module** (dilated 16)	128x64x64
	16-23	*Repeat layers 8 to 15*	
DECODER	24	**Bt-module** (upsampling)	64x128x128
	25	**Bt-module**	64x128x128
	26	**Bt-module**	64x128x128
	27	**Bt-module** (upsampling)	16x256x256
	28	**Bt-module**	16x256x256
	29	**Full-conv**	Cx512x512

Table 2. ENet architecture

can be seen more in details on Tab.2

2.3. BiSeNetV2

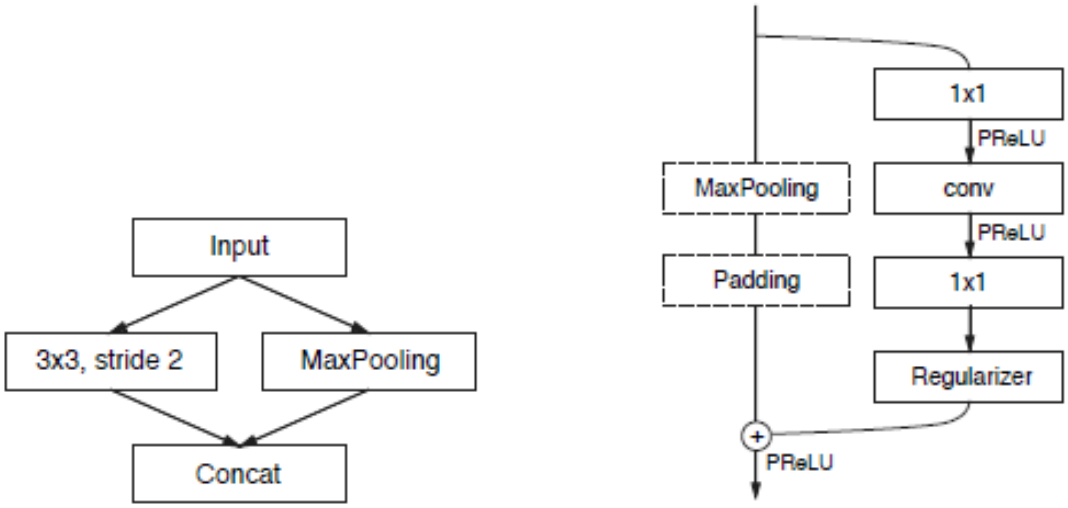
BiSeNetV2 [3] (Bilateral Segmentation Network Version 2), short for "Bilateral Segmentation Network V2," is a state-of-the-art deep learning architecture designed for real-time semantic segmentation tasks. BiSeNetV2 was built upon the original BiSeNet model, aiming to improve both accuracy and efficiency. The architecture is structured as a multi-branch network consisting of three key components: a Detail Branch (DB) and a Semantic Branch (SB), which are merged by an Aggregation Layer (AL). The details of the architectures are presented on Fig.4.
Starting from the Semantic Branch, it is responsible for capturing high-level semantic information from the input image. It adopts the fast-downsampling strategy to promote the level of the feature representation and enlarge the receptive field quickly (High-level semantics require large receptive field).
In parallel to the Semantic Branch, the Detail Branch is designed to capture fine-grained details and local information that might be lost or overlooked by the Semantic Branch due to downsampling operations. It aims to preserve spatial information at higher resolutions and is responsible for the spatial details, which is low-level information. Therefore, this branch requires a rich channel capacity to encode affluent spatial detailed information. Details on Fig.5.
The feature representation of the Detail Branch and the Semantic Branch is complementary, one of which is unaware of the information of the other one. Thus, an Aggregation Layer is designed to merge both types of feature representation to produce the final segmentation result.
To further improve the segmentation accuracy, they also proposed a booster training strategy. As the name implies, through the use of Segmentation Heads, it can enhance the feature representation in the training phase and can be discarded in the inference phase (in this way it has no extra inference cost). Overall, BiSeNetV2 leverages the complementary strengths of the Semantic Branch and the Detail Branch, while the Aggregation Layer effectively integrates the information from both branches to achieve accurate and detailed semantic segmentation results. This architecture strikes a balance between computational efficiency and segmentation accuracy, making it suitable for real-time applications on various platforms.

3. Experiments

We now presents a series of experiments, that we conduct to demonstrate the feasibility of using this model to do real time anomaly segmentation. We first introduce the datasets, and then the existing baseline on ERFNet and how we adapted them to the task of anomaly segmentation. Next we propose the techniques of temperature scaling. We then proceeded to train ENet and BiSeNetV2 networks on the void class, assumed as an anomaly. Finally we test the influence of loss function that are specifically made for anomaly detection.
Cityscape: [9 bise] is a large urban street scene dataset, from a car perspective. It contains 2957 fine annotated images for training, 500 for validation and another 1525 for testing, all annotated to 19 predefined classes.
Road Anomaly 21: [paper 5] is a dataset of 100 test images, with pixel-level annotations of general anomaly in full street scenes.
Road Obstacle 21: [paper 5] is a dataset of 327 test images, with pixel-level annotations of obstacle within the road region of interest.
Lost and Found [paper 6] is a dataset of 100 test images, with pixel-level annotations of small road hazards.
Static [paper 6] is a dataset of 29 test images, with pixel-level annotations of road obstacle.
Road Anomaly [paper 2211] is a dataset of 59 test images, with pixel-level annotations of unusual dangers which can be encountered by a vehicle on the road .

3.1. Baseline

As a first task, we performed several anomaly inference tests, Tab.3, using a pre-trained ERFNET model and the above-mentioned test datasets. Three methods were implemented:



(a) ENet initial block

(b) ENet bottleneck module

Figure 3. (a) ENet initial block. Convolution has 13 filters and Max Pooling return the 3 channels, which sums up to 16 feature maps after concatenation. (b) ENet bottleneck module. The convolution is either a regular, dilated, or full convolution (also known as deconvolution).

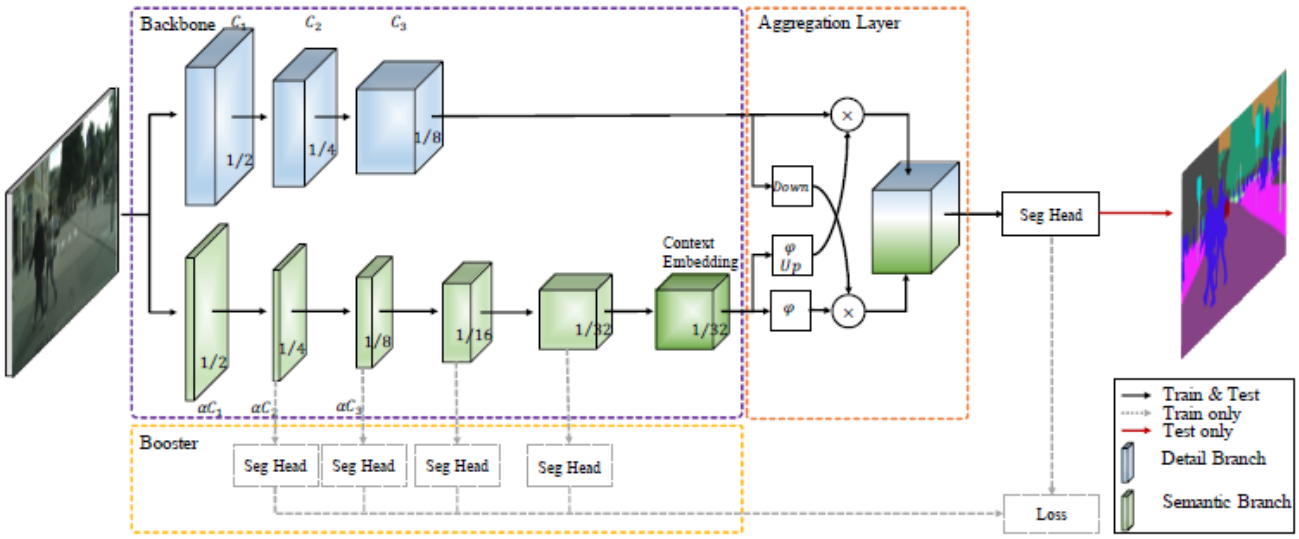


Figure 4. There are mainly three components: two-pathway backbone in the purple dashed box, the aggregation layer in the orange dashed box, and the booster part in the yellow dashed box. The two-pathway backbone has a Detail Branch (the blue cubes) and a Semantic Branch (the green cubes).

- MSP [1] (Maximum Softmax Probability) is a technique often utilized in probabilistic classification tasks where the softmax function is employed to derive class probabilities. In models such as softmax regression or neural networks with softmax output layers, the softmax function converts logits into probabilities. MSP involves selecting the class with the highest probability as the predicted class.

$$MSP = \frac{-max_k * exp(f(x)_k)}{\Sigma_i exp(f(x)_i)} \quad (1)$$

- Max Logit [2] is a classification method commonly employed in neural network models, particularly in scenarios where class prediction is based on the highest logit score. In neural networks, the output

Stage	Detail Branch					Semantic Branch						Output Size
	opr	*k*	*c*	*s*	*r*	*opr*	*k*	*c*	*e*	*s*	*r*	
Input												512×1024
S_1	Conv2d	3	64	2	1	Stem	3	16	-	4	1	256×512
	Conv2d	3	64	1	1							256×512
S_2	Conv2d	3	64	2	1							128×256
	Conv2d	3	64	1	2							128×256
S_3	Conv2d	3	128	2	1	GE	3	32	6	2	1	64×128
	Conv2d	3	128	1	2	GE	3	32	6	1	1	64×128
S_4						GE	3	64	6	2	1	32×64
						GE	3	64	6	1	1	32×64
S_5						GE	3	128	6	2	1	16×32
						GE	3	128	6	1	3	16×32
						CE	3	128	-	1	1	16×32

Figure 5. Instantiation of the Detail Branch and Semantic Branch.

layer typically produces logits, which are unnormalized scores representing the likelihood of each class. MaxLogit operates by selecting the class with the highest logit value as the predicted class. This method is straightforward and efficient, making it a popular choice in various machine learning tasks.

$$ML = -max_k \cdot f(x)_k \tag{2}$$

- Max Entropy [3], also known as Maximum Entropy, is a modeling approach employed for estimating probability distributions under conditions of limited information. The fundamental principle underlying MaxEntropy is to identify the probability distribution that maximizes entropy while satisfying given constraints based on available information.

$$ME = -\Sigma_i \sigma(f^i(x)) \cdot log(\sigma(f^i(x))) \tag{3}$$

3.2. Temperature scaling

Temperature scaling is employed as a confidence calibration technique for enhancing the anomaly segmentation capabilities of any classifier within a network. Consequently, we integrate this method into the anomaly segmentation process during inference. The objective of this phase is to determine the optimal temperature value that yields the most effective anomaly segmentation results.

$$MSP\_t = \frac{MSP}{t} \tag{4}$$

As evident from the results, Tab.4, a discernible positive trend is observed as the temperature decreases, leading to performance improvements. Specifically, there is an increase in the AuPRC, accompanied by a fall in FPR95.

3.3. Void Classifier

The Cityscapes dataset encompasses a diverse array of urban environments, containing 19 distinct category classes in addition to a category representing void or background regions. Within the context of this study, the void class is regarded as an anomaly. Our approach involves training ERFNet, ENet and BiSeNetV2 networks using this dataset, focusing specifically on modeling the void class as an anomaly. Subsequently, during the anomaly inference phase, we isolate and analyze the output pertaining solely to the void class.
The results, Tab.5 from 30 epochs of training, show that while ERFNet achieves a relatively high mIoU score of 21.605, while ENet and BiSeNetV2 exhibit superior AuPRC values in most datasets, indicating strong anomaly detection capabilities with controlled false positive rates. Overall, these findings highlight the nuanced trade-offs between semantic segmentation accuracy and anomaly detection efficacy among different deep learning architectures. For the training part, we used the repository [10] as a starting point, on which we then made the various verifications, modifications and evolutions necessary for our tasks.

3.4. Effect of Training Loss function

To further try to improve the performance of the ERFNet model we try to explore the impact of different training loss functions tailored explicitly for anomaly detection. This analysis includes:

- Logit Normalization (LogitNorm) is a technique proposed as a remedy for a common issue encountered in training neural networks for classification tasks, particularly when using cross-entropy loss. The problem

		SMIYC RA-21		SMIYC RO-21		FS L&S		FS Static		Road Anomaly	
Method	**mIoU**	**AuPRC**	**FPR95**	**AuPRC**	**FPR95**	**AuPRC**	**FPR95**	**AuPRC**	**FPR95**	**AuPRC**	**FPR95**
MSP	72.223	14.585	95.090	0.721	94.769	0.257	95.829	1.982	95.259	9.427	95.301
Max Logit	72.223	13.193	97.015	1.153	86.816	0.214	96.441	1.645	96.461	8.708	93.764
Max Entropy	72.223	14.310	96.716	0.832	94.084	0.228	96.830	1.953	94.052	9.107	95.313

Table 3. Result of Maximum Softmax Probability, Max Logit and Max Entropy performing anomaly segmentation on ERFNet as we vary the input dataset.

		SMIYC RA-21		SMIYC RO-21		FS L&S		FS Static		Road Anomaly	
Method	**mIoU**	**AuPRC**	**FPR95**	**AuPRC**	**FPR95**	**AuPRC**	**FPR95**	**AuPRC**	**FPR95**	**AuPRC**	**FPR95**
MSP	72.223	14.585	95.090	0.721	94.769	0.257	95.829	1.982	95.259	9.427	95.301
MSP(t = 0.1)	72.223	14.773	95.013	0.680	94.978	0.230	95.083	2.066	95.047	9.795	95.037
MSP(t = 0.25)	72.223	14.734	95.029	0.687	94.944	0.255	95.204	2.045	95.108	9.719	95.090
MSP(t = 0.5)	72.223	14.675	95.052	0.699	94.885	0.268	95.406	2.017	95.183	9.606	95.172
MSP(t = 0.75)	72.223	14.626	95.072	0.710	94.827	0.262	95.617	1.997	95.227	9.509	95.244
MSP(t = 1.1)	72.223	14.570	95.098	0.725	94.745	0.255	95.913	1.977	95.267	9.398	95.319

Table 4. Maximum Softmax Probability of ERFnet performing anomaly segmentation as we vary the input dataset and the we scale the temperature.

arises from the tendency of the norm (magnitude) of the logits to increase continually during training. This phenomenon often leads to overconfident predictions from the model. The core idea behind LogitNorm is to control the norm of the logits during training by enforcing a constant vector norm on them. By doing so, LogitNorm aims to decouple the influence of the output's norm during the optimization process of the neural network.

- Jaccard Loss[5] is often used as a loss function in training deep learning models for segmentation tasks. It's calculated as the complement of the Jaccard Index (Intersection over Union), that provides a quantitative assessment of the overlap between the elements of two sets, reflecting how similar or dissimilar they are. The Jaccard Loss (J_{loss}) is computed as follow:

$$J_{\text{loss}} = 1 - IoU \tag{5}$$

  It provides a quantitative assessment of the overlap between the elements of two sets, reflecting how similar or dissimilar they are.

- Focal Loss [6] is designed to address class imbalance in binary classification tasks. It's formulated as follows:

$$\text{FL}(p_t) = -\alpha_t(1-p_t)^{\gamma}\log(p_t) \tag{6}$$

  Where: - p_t is the predicted probability for the true class. - α_t is the balancing parameter for class t, often used to mitigate class imbalance. - γ is the focusing parameter that modulates the rate at which easy examples are down-weighted. A typical value for γ is 2. The term $(1-p_t)^{\gamma}$ down-weights the loss for easy examples (well-classified samples), focusing more on hard examples during training.

Comparing the results obtained, on Tab.6, from Focal Loss, Logit Normalisation and Jaccard Loss against what was obtained with Cross Entropy, we found no improvement. Cross Entropy, in general, gave us better results. With regard to the jointly variants, we note that the losses that perform worse individually benefited from the coupling. Those that were already performing well showed no further improvement. Probably, with a more extensive search of the weights between the jointly losses, better results can be achieved by enhancing more the best performing losses in each pair.

4. Conclusion

In conclusion, this work has shown a broad and interesting overview of real-time anomaly segmentation task for road scene. In our experiments, we tested various models, BisenetV2, Erfnet and Enet, against specific benchmarks for anomaly segmentation. We saw the effects of using temperature scaling and how, as the loss used (whether single or jointly) changes, the results also differ.

Certainly a major limitation were the few available computational resources and the short time. With more time available, it would have been interesting to train the models for many more epochs and make more in-depth compar-

30 epochs		SMIYC RA-21		SMIYC RO-21		FS L&S		FS Static		Road Anomaly	
Method	**mIoU**	**AuPRC**	**FPR95**	**AuPRC**	**FPR95**	**AuPRC**	**FPR95**	**AuPRC**	**FPR95**	**AuPRC**	**FPR95**
ERFNet	21.605	16.445	94.941	0.988	95.145	0.282	94.578	2.030	94.885	9.725	94.984
ENet	12.008	14.230	90.378	0.519	85.233	0.377	84.460	2.223	85.437	8.144	93.843
BiSeNet v2	17.552	21.679	98.508	1.001	99.950	0.322	96.530	2.114	92.463	10.702	97.723

Table 5. Result of anomaly segmentation on ERFNet, ENet and BiSeNetV2 as we train and perfome inference considering the void class as an anomaly class.

30 epochs		SMIYC RA-21		SMIYC RO-21		FS L&S		FS Static		Road Anomaly	
Method	**mIoU**	**AuPRC**	**FPR95**	**AuPRC**	**FPR95**	**AuPRC**	**FPR95**	**AuPRC**	**FPR95**	**AuPRC**	**FPR95**
Cross Entropy (CE)	27.820	21.804	89.356	2.247	95.795	0.302	94.105	2.789	80.597	13.744	81.117
Focal Loss (FL)	24.444	19.164	94.412	0.477	96.313	0.248	92.427	2.742	64.423	11.494	82.063
Logit Norm (LN)	10.410	13.201	94.754	0.872	67.523	0.261	87.99	2.605	82.494	11.588	86.738
Jaccard Loss (JL)	4.760	13.938	91.123	0.646	94.348	0.249	91.008	2.089	85.259	9.385	90.851
JL / CE	27.866	17.422	98.580	1.224	87.967	0.307	92.806	2.498	88.439	12.615	82.758
JL / FL	23.533	13.498	97.789	0.371	99.791	0.252	93.235	2.029	88.368	9.348	94.003
LN / CE	21.841	12.068	97.998	0.885	93.759	0.234	98.445	2.105	88.950	9.312	94.967
LN / FL	23.790	16.908	98.478	0.555	97.190	0.273	88.163	6.003	75.680	12.317	81.431

Table 6. Result of anomaly segmentation on ERFNet, as we train and perform inference considering considering differents loss functions.

isons on the different hyperparameters available. Also for the final task, it would have been interesting to further investigate the use of different weights between the various jointly available losses.

References

[1] Sangpil Kim Eugenio Culurciello Adam Paszke, Abhishek Chaurasia. Enet: A deep neural network architecture for real-time semantic segmentation, 2016. 1, 2

[2] S. Ioffe J. Shlens C. Szegedy, V. Vanhoucke and Z. Wojna. Rethinking the inception architecture for computer vision, 2015. 2

[3] Jingbo Wang Gang Yu Chunhua Shen Nong Sang Changqian Yu, Changxin Gao. Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation, 2020. 1, 3

[4] Jang E. Alemi A.A Choi, H. Waic, but why? generative ensembles for robust anomaly detection., 2018. 1

[5] Omran M. Ramos S. Rehfeld T. Enzweiler M. Benenson R. Franke U. Roth S. Schiele B. Cordts, M. The cityscapes dataset for semantic urban scene understanding., 2016. 1

[6] Luis M. Bergasa Eduardo Romera, José M. Álvarez and Roberto Arroyo. Erfnet: Efficient residual factorized convnet for real-time semantic segmentation, 2018. 1, 2

[7] Lenz P. Urtasun R. Geiger, A. Are we ready for autonomous driving? the kitti vision benchmark suite., 2012. 1

[8] El-Yaniv R Golan, I. Deep anomaly detection using geometric transformations., 2018. 1

[9] Gimpel-K. Hendrycks, D. A baseline for detecting misclassified and out-of-distribution examples in neural networks, 2017. 1

[10] https://github.com/zh320/realtime-semantic-segmentation pytorch. realtime-semantic-segmentation-pytorch. 5

[11] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015. 2

[12] A. Jain Y. LeCun J. Tompson, R. Goroshin and C. Bregler. Efficient object localization using convolutional networks, 2015. 2

[13] S. Ren K. He, X. Zhang and J. Sun. Deep residual learning for image recognition., 2015. 2

[14] S. Ren K. He, X. Zhang and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, 2015. 2

[15] Lee K. Lee H. Shin J. Lee, K. A simple unified framework for detecting out-of-distribution samples and adversarial attacks., 2018. 1

[16] Gales M. Malinin, A. Predictive uncertainty estimation via prior networks., 2018. 1

[17] McDaniel P. Papernot, N. Deep k-nearest neighbors: Towards confident, interpretable and robust deep learning, 2018. 1

[18] Almohsen R. Doretto G. Pidhorskyi, S. Generative probabilistic novelty detection with adversarial autoencoders., 2018. 1

[19] Vandermeulen R. Goernitz N. Deecke L. Siddiqui S.A. Binder A. M¨uller E. Kloft M. Ruff, L. Deep one-class classification., 2018. 1

[20] Khalooei M. Fathy M. Adeli E. Sabokrou, M. Adversarially learned one-class classifier for novelty detection., 2018. 1

[21] A. Handa V. Badrinarayanan and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling., 2015. 2