Real-Time Anomaly Segmentation for Road Scenes

Advance Machine Learning 2024/2025

Francesco Passiatore
Politecnico di Torino
Matricola: s332821
s332821@studenti.polito.it

Marco Pontrandolfo
Politecnico di Torino
Matricola: s331060
s331060@studenti.polito.it

Fabiano Vaglio
Politecnico di Torino
Matricola: s329986
s329986@studenti.polito.it

I. ABSTRACT

This study addresses real-time anomaly segmentation in road scenes, a critical task for applications like autonomous driving. We evaluate baseline segmentation models. The source code of this project is available at https://github.com/FrancescoPassiatore/AnomalySegmentation_CourseProjectBaseCode.

II. INTRODUCTION

Anomaly detection is a fundamental challenge in computer vision, focusing on identifying regions within an image that deviate from expected patterns. This process is crucial in safety-critical applications such as autonomous driving, where the ability to promptly detect anomalies like road debris or unexpected obstacles is essential. Similarly, in industrial settings, anomaly detection plays a vital role in quality control and equipment monitoring.

Among the various approaches, *per-pixel anomaly segmentation* stands out for its ability to precisely localize non-conforming areas. This level of detail is particularly relevant in scenarios like autonomous driving, where systems must quickly detect and respond to potential hazards. However, the complexity of this task lies in distinguishing between data previously encountered during training (*In-Distribution*, ID) and completely new data (*Out-of-Distribution*, OOD).

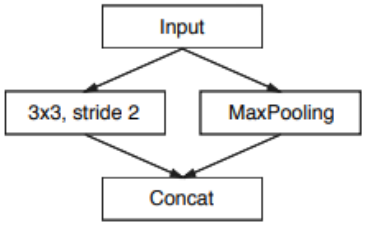
For our experiments, we employ three well-known segmentation models: **BiSeNet**, **ERFNet**, and **ENet**, chosen for their efficiency and widespread application in segmentation tasks. Our goal is to investigate how different training strategies and the use of diverse datasets can enhance the models' ability to detect anomalies, ultimately contributing to the development of more effective and reliable systems for real-world applications.

III. RELATED WORKS

Real-time semantic segmentation is a significant challenge, requiring a balance between computational efficiency and the ability to capture both spatial and contextual features of an image. In this section, we discuss three prominent architectures: **ENet**, **ERFNet**, and **BiSeNet**, each offering distinct advantages in terms of speed and accuracy.

A. *Efficient Neural Network (ENet)*

The ENet architecture [1] is composed of two main components: the encoder and the decoder. The encoder is responsible for feature extraction and image compression. Starting from the initial stage, that contains a single block presented in Figure 1, ENet's initial block, maxPooling is performed with non-overlapping 2×2 windows, and the convolution has 13 filters, which sums up to 16 feature maps after concatenation. A key aspect of the encoder is the early downsampling of the input, which helps to reduce computational costs. ENet's architecture is divided into different stages formed by bottleneck modules, illustrated in Figure 2, where conv is either a regular, dilated, or full convolution (also known as known as deconvolution) with 3×3 filters, or a 5×5 convolution decomposed into two asymmetric ones.



(a)

Fig. 1. ENet initial block.

On the other hand, the decoder is designed to be more lightweight compared to the encoder, focusing on reconstructing the segmented image. It achieves this through upsampling and the use of lightweight convolutions that help to rebuild spatial information effectively. Additionally, skip connections are incorporated to recover details that may have been lost during the encoding process..

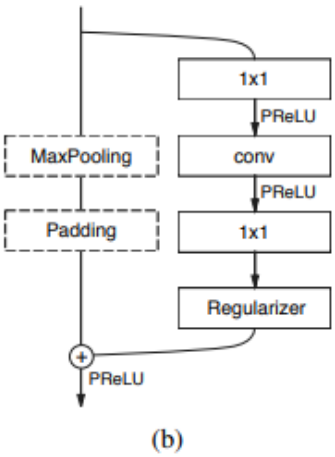


Fig. 2. ENet bottleneck module.

With this design, ENet strikes a balance between efficiency and accuracy, making it an ideal choice for real-time applications on embedded devices.

B. *Efficient Residual Factorized Network (ERFNet)*

ERFNet (Efficient Residual Factorized Network) [2] is a network designed for real-time semantic segmentation, balancing computational efficiency and accuracy. Its architecture follows the trend of using convolutions with residual connections but they introduce an optimized design of residual blocks.

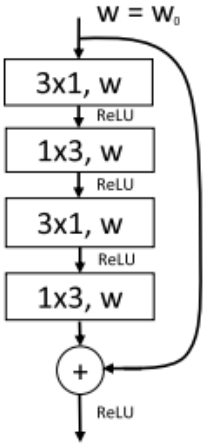


Fig. 3. ERFNet's Non-bottleneck-1D.

The Non-bottleneck-1D illustrated in Figure 3, proposed by ERFNet, replaces 3x3 convolutions with a combination of 3x1 and 1x3 convolutions, reducing computational cost without sacrificing performance. This approach enables ERFNet to provide accurate segmentation with reduced inference time, making it ideal for applications in the automotive and robotics fields.

C. *Bilateral Segmentation Network (BiSeNet)*

BiSeNet (Bilateral Segmentation Network) [3] is an efficient neural network designed for real-time semantic segmentation by combining spatial and contextual information. Its architecture, consists of two paths as shown in Figure 4: the Spatial Path, which preserves high-resolution details such as object boundaries, and the Context Path, which captures global semantic context through progressive downsampling. The integration of these two paths through a Feature Fusion Module (FFM) allows BiSeNet to achieve accurate segmentation while maintaining low computational costs. This design enables the network to process high-resolution images at real-time speeds, making it well-suited for applications like autonomous driving and robotics, where both speed and precision are critical.

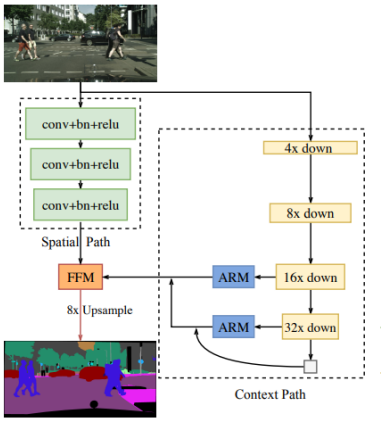


Fig. 4. BiSeNet dual-path architecture

IV. Methods and Metrics

Several approaches have been introduced to detect anomalous samples by analyzing the model's predictive outputs and internal features. This section explores the metrics used and five prominent methods: Maximum Softmax Probability (MSP), Maximum Logit (MaxLogit), Maximum Entropy (MaxEntropy), Void Classifier.

A. *Maximum Softmax Probability (MSP)*

MSP is a baseline method for detecting misclassified and out-of-distribution (OOD) examples in neural networks. It utilizes the softmax output probabilities of a classifier, where the highest softmax probability indicates the predicted class. The anomaly score $s(x)$ for an input x is defined as:

$$s(x) = 1 - \max_c \sigma(f_c^z(x))$$

where $\sigma(f_c(x))$ represents the softmax probability for class c for a pixel x . A higher anomaly score implies that the input is more likely to be misclassified or OOD.

*a) **Temperature Scaling**:* A problem with softmax probabilities is that often they exhibit overconfidence, MSP assumes that the highest softmax probability corresponds to model confidence, at this is not always accurate, but techniques like temperature scaling can avoid this issue. Temperature scaling is a simple post-processing calibration technique used to improve the reliability of confidence scores produced by

deep learning models. It works by scaling the logits of the model's output using a single scalar parameter, referred to as the temperature.

$$s(x) = 1 - \max_c \sigma(f_c^z(x)/T)$$

T is the temperature , choosing a higher value will soften the probability distribution, improving calibration.

B. *Maximum Logit (MaxLogit)*

MaxLogit [4] is a method that uses the logits (a raw prediction score , pre-softmax output) of a neural network to detect OOD samples. The anomaly score $s(x)$ is given by:

$$s(x) = -\max_c f_z^c(x)$$

By directly evaluating the logits, MaxLogit provides a more nuanced measure of confidence compared to softmax probabilities, avoiding the normalization of logits into probabilities and therefore inflated confidence scores for non-true predictions.

C. *Maximum Entropy (MaxEntropy)*

MaxEntropy assesses the uncertainty of a model's predictions by calculating the entropy of the softmax output distribution. The anomaly score $s(x)$ is given by:

$$H(x) = -\sum_c \sigma(f_z^c(x)) \log \sigma(f_z^c(x))$$

Higher entropy values indicate greater uncertainty in the prediction, suggesting that the input may be anomalous. MaxEntropy can effectively identify samples that do not conform to the learned distribution.

D. *Void Classifier*

The Void Classifier method involves augmenting the original classification model with an additional "void" class representing anomalies. During training, the model learns to classify in-distribution samples into their respective classes and assigns OOD samples to the void class. At inference time, the model predicts an input as anomalous if the void class has the highest predicted probability. This approach enables the model to explicitly learn a representation for anomalies, improving detection performance.

E. *AuPRC*

The Area Under the Precision-Recall Curve (AuPRC) evaluates classification performance in imbalanced tasks by quantifying the trade-off between precision (proportion of true positives among predicted positives) and recall (proportion of true positives detected). Unlike accuracy, AuPRC emphasizes this balance, making it especially relevant for skewed class distributions. A higher AuPRC indicates stronger model performance in maintaining both metrics.

F. *FPR95*

The False Positive Rate at 95% True Positive Rate (FPR95) measures the rate of negative samples misclassified as positives when the true positive rate is fixed at 95%. This metric is critical for applications requiring high sensitivity (e.g., anomaly detection), where minimizing false alarms at near-maximal detection rates is essential. Lower FPR95 values denote better robustness.

G. *mIoU*

The mean Intersection over Union (mIoU) is a widely adopted evaluation metric for semantic segmentation tasks. It quantifies segmentation accuracy by averaging the Intersection over Union (IoU) across all classes. For a single class, the IoU is computed as:

$$\text{IoU} = \frac{|A \cap B|}{|A \cup B|},$$

where A and B denote the predicted segmentation mask and ground-truth mask, respectively. Here, $|A \cap B|$ represents the true positives (correctly predicted pixels), and $|A \cup B|$ corresponds to the union of true positives, false positives, and false negatives.

The mIoU generalizes this measure for multi-class segmentation by averaging the class-specific IoU values:

$$\text{mIoU} = \frac{1}{N} \sum_{i=1}^{N} \text{IoU}_i,$$

where N is the total number of classes and IoU_i is the IoU for the i-th class. By equally weighting all classes, mIoU ensures balanced evaluation, even in imbalanced datasets. It is favored over pixel-wise accuracy for its sensitivity to both region consistency and boundary precision, penalizing over- and under-segmentation errors effectively.

V. Loss Functions

In our work, for the evaluation on the impact of Loss Functions on Anomaly Detection we will utilize three different loss functions to optimize the model's performance in anomaly detection: Enhanced IsoMax Loss, Logit Loss, and Focal Loss. Each of these functions are designed to address specific challenges in training deep learning models, particularly to enhance the ability to detect out-of-distribution (OOD) samples and tackle data imbalance issues.

A. *Enhanced IsoMax Loss*

The Enhanced IsoMax Loss [5] is the modified version of IsoMax, which acts as a direct replacement for the conventional Softmax loss used in deep learning models, primarily aimed at improving out-of-distribution (OOD) detection performance. Unlike Softmax, which converts logits into probability distributions and relies on cross-entropy for training, IsoMax enhances this process by leveraging distance-based calculations to improve feature separability and reduce classification errors. By normalizing both protoypes and the fetaures, changing the initialization of the prototypes and adding the

distance scale, a learnable scalar parameter that multiplies the feature-prototype distances, they obtained the Enhanced IsoMax loss (IsoMax+ loss).

$$\mathcal{L}_{\text{isoMax+}} = -\log^{\dagger\dagger}\left(\frac{\exp\left(-E_s|d_s|\left\|\widehat{f_\theta(x)} - \widehat{p_\phi^k}\right\|\right)}{\sum_j \exp\left(-E_s|d_s|\left\|\widehat{f_\theta(x)} - \widehat{p_\phi^j}\right\|\right)}\right)$$

B. *Logit Normalization*

Logit Normalization [6], encourages the direction of the logit to be consistent with the corresponding one-hot label, without optimizing the magnitude of the logit. The logit vector is normalized to be a unit vector with a constant magnitude. The softmax cross-entropy loss is then applied on the normalized logit vector, therefore we'll have the new loss function :

$$\mathcal{L}_{\text{logit\_norm}}\big(f(\boldsymbol{x};\theta), y\big) = -\log\left(\frac{e^{f_y/(\tau\|\boldsymbol{f}\|)}}{\sum_{i=1}^{k} e^{f_i/(\tau\|\boldsymbol{f}\|)}}\right),$$

The Logit Normalization aims to reduce the overconfidence of the models, which leads to poor performance on OOD examples.

C. *Focal Loss*

Focal Loss [7] improves upon the standard cross-entropy loss by addressing class imbalance, particularly in tasks such as object detection and anomaly detection. It assigns greater weight to hard-to-classify samples while down-weighting well-classified ones.

The Focal Loss formula is:

$$\mathcal{L}_{\text{Focal}}(p_t) = -\alpha(1-p_t)^\gamma \log(p_t)$$

where: p_t is the model's predicted probability for the true class, α is a balancing factor to weight the importance of positive/negative classes, γ is the focusing parameter, which adjusts the loss contribution of easy vs. hard examples.

When $\gamma = 0$, the function reduces to standard cross-entropy. Higher values of γ focus learning on challenging examples, enhancing model robustness in imbalanced datasets.

D. *Cross-Entropy*

The Cross-Entropy (CE) loss is a fundamental objective function for classification and semantic segmentation tasks. It measures the dissimilarity between a model's predicted probability distribution $\mathbf{p}$ and the ground-truth distribution $\mathbf{y}$. For a single pixel (or data point) with C classes, the CE loss is defined as:

$$\mathcal{L}_{\text{CE}} = -\sum_{i=1}^{C} y_i \log(p_i),$$

where $y_i \in \{0, 1\}$ is the one-hot encoded ground-truth label (1 for the true class, 0 otherwise), and $p_i \in [0, 1]$ is the predicted probability for class i.

In semantic segmentation, CE is computed independently for each pixel and averaged across an image or batch. For a batch of N pixels, the loss becomes:

$$\mathcal{L}_{\text{CE}} = -\frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{C} y_{n,i}\log(p_{n,i}).$$

VI. Datasets

To train and evaluate our anomaly segmentation framework, we leverage four complementary datasets, each addressing distinct aspects of robustness in autonomous driving scenarios:

A. *Cityscapes Dataset*

Cityscapes [8] serves as our primary training dataset, providing 5,000 high-resolution urban scene images from 50 differenct cities. These include 2,975 training images, 500 validation images, and 1,525 test images, all annotated with fine-grained pixel-level labels for 19 semantic classes. This dataset establishes the baseline segmentation performance for known object categories.

B. *Fishyscapes Dataset*

Fishyscapes [9] provides two complementary evaluation settings:

Lost & Found: derived from the Lost and Found Dataset, it contains images with small anomalous objects.

FS Static: based on the Cityscapes validation set, images are augmented with anomalous objects not present in training set.

Together, these subsets test both natural and artificial anomaly detection across varied contexts, emphasizing robustness to both authentic and adversarial edge cases.

C. *RoadAnomaly*

The RoadAnomaly dataset focuses on detecting anomalies in real-world scenes. It features 60 images with pixel-level annotations, with anomalous objects, placed in unpredictable locations making it a challenging test for anomaly segmentation models.

D. *SegmentMeIfYouCan*

SegmentMeIfYouCan [10], is a benchmark for anomaly segmentation, it introduces two key datasets:

RoadAnomaly21: The road anomaly track benchmarks general anomaly segmentation in full street scenes.

RoadObstacle21: The road obstacle track focuses on safety for automated driving. The objects to segment in the evaluation data always appear on the road ahead, i.e. they represent realistic and hazardous obstacles that are critical to detect.

VII. Experiments

A. *Performance Analysis Across Datasets and use of temperature scaling*

The results obtained across multiple datasets (SMIYC RA-21, SMIYC RO-21, FS-L&F, FS Static, and Road Anomaly), Table I ,reveal several key aspects of the performance of the methods under consideration: MSP, MaxLogit, and Max Entropy.

Method	mIoU	SMIYC RA-21		SMIYC RO-21		FS-L&F		FS Static		Road Anomaly	
		AuPRC	FPR95	AuPRC	FPR95	AuPRC	FPR95	AuPRC	FPR95	AuPRC	FPR95
MSP	72.20%	29.10	62.55	2.71	65.22	1.75	50.59	7.47	41.84	12.42	82.58
MaxLogit	72.20%	**38.32**	**59.34**	**4.63**	**48.44**	**3.30**	**45.49**	**9.50**	**40.30**	**15.58**	**73.25**
Max Entropy	72.20%	30.97	62.66	3.04	65.91	2.58	50.16	8.84	41.55	12.67	82.75

TABLE I
RESULTS ON DATASETS WITH DIFFERENT METHODS.

General Performance Across Datasets

1) Consistent mIoU across methods: All methods achieve 72.20% mIoU, showing identical performance on standard segmentation tasks. This is due to the shared backbone and training, all three are based on the ERFNet model. Differences emerge solely in anomaly detection metrics.

2) MaxLogit outperforms Softmax-based methods: MaxLogit achieves better AuPRC and FPR95 across all datasets, demonstrating its effectiveness for anomaly segmentation. Raw logits preserve uncertainty signals lost in softmax normalization, which instead compresses values into overconfident probabilities. MSP and MaxEntropy rely on softmax outputs, leading to missed anomalies.

Temperature Scaling

Table II shows the impact of temperature scaling parameter t on anomaly detection performance using MSP. As previously mentioned, temperature scaling enables calibration of model confidence, influencing the trade-off between AuPRC and FPR95. While mIoU remains unaffected (72.20% across all configurations), as temperature scaling modifies confidence estimates through post-processing.

3) Effects of Temperature Scaling:

- **Low** t **(0.1–0.25)**: Sharpening confidence scores exacerbates overconfidence, degrading anomaly detection.
- **Moderate** t **(0.5–1.1)**: Intermediate values balance confidence sharpness and calibration.
- **High** t **(1.5–3.0)**: Smoother distributions improve anomaly discrimination in specific contexts Excessive smoothing harms the other datasets.

4) Challenge of Finding an Optimal t: The analysis reveals three fundamental limitations to universal temperature selection:

- **Metric conflict**: Rarely single t values simultaneously optimizes AuPRC and FPR95. For instance:
  - SMIYC RA-21: $t = 1.1$ gives peak AuPRC (29.40) but suboptimal FPR95 (62.65)
  - $t = 0.5$ reduces SMIYC RO-21 FPR95 (63.23 vs 65.22) but sacrifices AuPRC (2.42 vs 2.76)
- **Dataset dependence**: Optimal t varies significantly across benchmarks:
  - RoadAnomaly: Insensitive to t (AuPRC 0.7 across $t = 0.1$-3.0)
  - FS Static: Requires 3x higher t than SMIYC RO-21 for best performance

Therefore identifying a single optimal value of t across all datasets proves challenging due to the intrinsic differences in the characteristics of the datasets. The variation in the optimal t across datasets can be explained by their unique properties:

- Anomaly Distribution: datasets with balanced or well-defined anomalies benefit from moderate temperature adjustments (e.g., SMIYC RA-21)
- Anomaly density and variability: sparse or highly variables anomalies (e.g., FS-L&F, FS Static) require higher t values, reducing the model's overconfidence and enhance robustness to subtle anomalies
- Anomaly simplicity: simple anomalies with clear distinctions from normal regions (e.g., SMIYC RO-21) favor lower t values, which sharpen the model's probabilities for more decisive classification

This variability underlines the importance of tailoring temperature scaling to the specific characteristics of each dataset, rather than adopting a universal value of t.

B. Anomaly Detection via Void Class in Cityscapes

In this section, we explore the potential of semantic segmentation models to detect anomalies by leveraging the void class within the Cityscapes dataset. The void class, representing background or unlabelled regions, is treated as a proxy for anomalies. Specifically, we train segmentation models to include the void class alongside the 19 known category classes in Cityscapes. The anomaly inference is performed by isolating the output of the void class during evaluation, results are in Table III. To this end, we employ three segmentation architectures: ENet, BiSeNet, and ERFNet, leveraging pretrained models and optimizing their training configurations for this task.

We utilized publicly available pretrained models for ENet[1] and BiSeNet[2], sourced from their respective repositories.

Each model underwent a series of training experiments designed to optimize segmentation performance, focusing particularly on the void class. The experiments systematically varied key parameters, including learning rate schedules, data augmentation strategies, optimizer selection, and layer freezing. Performance was evaluated using the Intersection-over-Union (IoU) metric and loss reduction, with the void class serving as a measure of anomaly detection capability.

1) Model-Specific Training Methodologies:

[1]https://github.com/davidtvs/PyTorch-ENet
[2]https://github.com/CoinCheung/BiSeNet

Method	mIoU	SMIYC RA-21		SMIYC RO-21		FS-L&F		FS Static		Road Anomaly	
		AuPRC	FPR95	AuPRC	FPR95	AuPRC	FPR95	AuPRC	FPR95	AuPRC	FPR95
MSP	72.20%	29.10	62.55	2.71	65.22	1.75	50.59	7.47	41.84	**12.42**	**82.58**
MSP ($t = 0.5$)	72.20%	27.06	62.73	**2.42**	**63.23**	1.28	66.73	6.60	43.48	11.92	82.88
MSP ($t = 0.75$)	72.20%	28.16	62.49	2.57	64.13	1.43	51.76	6.93	42.49	12.31	82.81
MSP ($t = 1.1$)	72.20%	**29.40**	**62.65**	2.76	65.87	1.86	50.78	7.13	41.62	12.64	83.25
t **best**	72.20%										
$t = 0.1$	72.20%	19.85	93.47	1.23	93.75	0.63	93.39	3.71	93.40	10.48	94.97
$t = 0.25$	72.20%	23.93	89.82	1.95	89.44	0.96	89.90	5.26	89.35	11.19	93.77
$t = 1.5$	72.20%	30.23	68.70	2.96	68.70	**2.29**	**47.98**	9.68	42.76	12.44	83.40
$t = 2.0$	72.20%	30.61	65.33	3.33	72.67	2.29	66.88	9.86	47.01	12.66	82.97
$t = 2.5$	72.20%	30.64	66.71	2.99	75.93	2.92	66.97	10.78	47.01	12.66	82.97
$t = 3.0$	72.20%	30.54	68.13	2.95	78.77	3.06	66.41	**10.77**	**42.76**	12.65	82.97

TABLE II
RESULTS GIVEN BY DIFFERENT VALUES OF t.

a) ENet: For ENet, we adopted the class-weighting function proposed in the original paper to address class imbalance. We utilized the CrossEntropy2D loss, already used in the ERFNet model. Two main configurations were explored to achieve the best results:

- **Initial Configuration:**
  - Applied the data augmentation strategy from ERFNet's training procedure.
  - Used the Adam optimizer with a LambdaLR scheduler.
  - Trained for 20 epochs with all layers frozen except the final layer.
  - **Results:** Achieved a low IoU ($\sim$ 3% for first epochs).

  Therefore we moved to a different configuration, trying to increase the IoU of the first epochs, aiming at reaching a higher one at later epochs. We tried to adopt the methods used in ENet's repository, changing from a LambdaLR scheduler to a StepLR scheduler, using ENet's augmentation and removing parameter freezing, as it could possibly limit the performances.
- **Improved Configuration:**
  - Adjusted augmentation strategies for improved diversity in training data.
  - Removed parameter freezing and adopted a StepLR scheduler, consistent with ENet's pretrained model training.
  - **Results:** Observed significant performance improvements, with an initial IoU of 18% in the first epoch. The model achieved a final mIoU of 36.38% after 20 epochs.

  We managed to improve significantly the model with these modifications, improving the mIoU. We tried other methods to improve these values by adding augmentations, changing optimizer (AdamW), and modifying the weight decay and learning rate, but no improvements were made. At 36% the mIoU is still quite low, but through an increase of the training epochs and training just the decoder, these values may increase.

2) Models and Training Configurations:

a) BiSeNet: For BiSeNet, a similar methodology was employed with adjustments to training strategies, we used the methods deployed in the repository of the pretrained model, where SGD was preferred to Adam probably to better generalize and to have more stability with BatchNorm. OhmCELoss was used, which combines Cross-Entropy loss with Online Hard Example Mining (Ohem) , this focuses training on the hardest samples (pixels with high loss), improving model performance on difficult cases.

- **Initial Configuration:**
  - Same augmentation as for ERFNet.
  - Utilized the Stochastic Gradient Descent (SGD) optimizer as in the pretrained model.
  - Employed a LambdaLR scheduler and OhemCELoss, consistent with the pretrained repository.
  - Trained for 20 epochs with parameter freezing.
  - **Results:** Achieved a low initial IoU ($\sim$ 18% per initial epochs), indicating limited segmentation capability under this configuration.

  Our first try was with freezed parameters, but didn't achieve the best results , therefore we changed this, trying to improve the IoU per epoch.
- **Improved Configuration:**
  - Removed parameter freezing to allow end-to-end training.
  - **Results:** Marked improvement in performance, with the IoU starting at 34% in the first epoch and increasing over time to reach a final mIoU of 61.72%.

In both ENet and BiSeNet we achieve better results when we don't freeze parameters, the model better adapts to the dataset and to the specific task, so in our case for the void classification and with the void class integrated.

b) ErfNet: The ErfNet model was trained using its original training script without significant modifications. The default configuration was applied:

- **Configuration:**
  - Training applied only on decoder.

Void Classification Model	mIoU	SMIYC RA-21		SMIYC RO-21		FS-L&F		FS Static		Road Anomaly	
		AuPRC	FPR95	AuPRC	FPR95	AuPRC	FPR95	AuPRC	FPR95	AuPRC	FPR95
ENet	36.38%	11.35	93.32	0.42	99.79	2.00	50.44	2.28	85.24	7.30	92.85
ERF-Net	**72.50%**	18.14	82.65	0.84	99.70	**12.13**	**15.99**	**15.72**	**57.59**	8.64	92.52
BiSeNet	61.72%	**30.87**	**73.21**	**2.71**	**86.60**	9.56	64.65	2.00	92.29	**11.33**	**87.65**

TABLE III
RESULTS ON VOID CLASSIFICATION.

- Trained for 20 epochs using the default setup provided in the repository.
- **Results:** Achieved reasonable performance, serving as a benchmark for comparison with ENet and BiSeNet.

With ERFNet we choose to avoid modifying the default settings, only modifying the 19th weight from 0 to 1.0. This configuration showed great results, reaching an mIoU of 72.50% and good results for AuPRC and FPR95 for some datasets.

3) Results with Void Classifier:

a) Overall Void Classification Performance: The results in Table III underscore the importance of architectural design in balancing segmentation accuracy and anomaly detection:

- ERFNet's factorized convolutions and residual connections optimize for general segmentation but limit adaptability to novel anomalies.
- BiSeNet's dual-path architecture prioritizes spatial detail and context fusion, making it robust to rare or irregular anomalies at the cost of slightly lower mIoU.
- ENet's lightweight design sacrifices both segmentation precision and anomaly detection capability for real-time efficiency

BiSeNet excels in 3/5 datasets (SMIYC RA-21, SMIYC RO-21, Road Anomaly), achieving the highest AuPRC and lowest FPR95, highlighting its strengths, despite having a lower mIoU than ERFNet. ERFNet lead in FS-L&F and FS Static, suggesting better adaptation to anomalies resembling known classes.

C. Impact of Loss Functions on Anomaly Detection

In this section, we analyze the impact of incorporating advanced loss functions tailored for anomaly detection into the training of semantic segmentation models. Specifically, we evaluate two loss functions—**Enhanced Isotropy Maximization Loss (IsomaxPlus)** and **Logit Normalization Loss**—in conjunction with standard loss functions **Focal Loss** and **Cross-Entropy Loss (CE)**. The objective of this evaluation is to determine whether these loss functions enhance anomaly detection performance by improving feature separation and increasing confidence in predictions for anomalous regions.

1) Implementation of losses:

a) Enhanced IsoMax: IsoMaxPlus[3] [5] has two components to its loss, the first part should replace the last layer of the used model, but its thought for a linear layer. ERFNet's last layer is a convolutional layer, therefore we had to modify the structure of the first part of the loss to adapt it to the output of the convolutional layer, modifying the shape of the features to (B,H,W,C), and then flattening the features to (B*H*W,C). Other then modifying IsoMaxPlus, we had to use a mixed-precision training which refers to the technique of using both 16-bit (half precision) and 32-bit (single precision) floating-point numbers during model training to improve computational efficiency while maintaining model accuracy. This was needed due to problems of OOM (Out-of-Memory),a situation where a computer or device runs out of available memory (RAM or GPU memory) during a process or operation, causing it to fail or crash, with Google Colab. This was due to the heavy computations needed to compute distances in IsoMaxPlus.

b) Logit Normalization: Logit normalization has been introduced as a technique to constrain the logits to a fixed norm, preventing excessive model confidence. By normalizing the logits, the softmax output is forced to distribute probability mass more evenly, which in turn helps calibrate uncertainty. However, the degree of normalization is influenced by the temperature parameter T , which acts as a scaling factor after normalization. When T is large, the normalized logits are scaled down, resulting in a smoother softmax distribution where no single class dominates. This leads to lower confidence predictions and can help in applications requiring better uncertainty estimation. Conversely, when T is small, the logits are scaled up, sharpening the softmax distribution and producing higher confidence predictions, with one class having a dominant probability.

In our experiments with ERFNet on the Cityscapes dataset we initially applied logit normalization directly to the cross-entropy loss with a default scaling factor $T = 1.0$. However, this resulted in poor performance, with the loss stagnating at high values and a very low mean Intersection over Union (mIoU).

To mitigate this issue, we experimented with a lower temperature value, specifically $T = 0.05$. This adjustment effectively reduced the impact of normalization by amplifying the magnitude of the logits. As a result, The loss started decreasing more effectively and the mIoU improved significantly compared to the previous setting.

This experiment highlights the importance of tuning the temperature parameter when applying logit normalization. Without an appropriate T, normalization can either over-regularize the logits, leading to inefficient learning, or under-regularize them, reducing the intended benefits of uncertainty

[3]https://github.com/dlmacedo/entropic-out-of-distribution-detection/blob/master/losses/isomaxplus.py

TABLE IV
PERFORMANCE COMPARISON ACROSS DATASETS FOR VARIOUS METHODS AND LOSS CONFIGURATIONS.

Method	Loss	mIoU (%)	SMIYC RA-21		SMIYC RO-21		FS-L&F		FS Static		Road Anomaly	
			AuPRC	FPR95	AuPRC	FPR95	AuPRC	FPR95	AuPRC	FPR95	AuPRC	
MSP	Logit Norm + CE	61.08	30.37	86.25	1.12	69.23	2.85	72.56	12.08	40.84	11.80	81.55
	Logit Norm + Focal Loss	63.04	27.48	92.03	1.97	98.63	4.28	73.41	15.00	40.08	11.30	82.89
	IsomaxPlus + CE	**68.21**	30.42	62.29	12.84	19.35	1.44	69.60	9.85	27.56	**16.88**	**72.20**
	IsomaxPlus + Focal Loss	63.48	43.42	59.89	1.87	47.83	1.04	64.38	8.52	33.33	16.41	73.41
MaxLogit	Logit Norm + CE	61.08	28.93	87.34	1.07	77.08	3.18	71.27	13.01	40.93	11.59	81.74
	Logit Norm + Focal Loss	63.04	25.19	94.21	1.62	99.98	4.52	72.86	15.86	39.37	10.95	83.54
	IsomaxPlus + CE	**68.21**	29.88	64.81	16.28	19.22	1.21	71.87	**10.82**	**25.71**	16.72	72.80
	IsomaxPlus + Focal Loss	63.48	39.42	61.46	1.79	53.69	**1.12**	**50.35**	10.19	32.60	16.72	75.85
MaxEntropy	Logit Norm + CE	61.08	29.79	86.28	1.09	69.10	3.64	72.62	13.37	40.93	11.85	81.54
	Logit Norm + Focal Loss	63.04	24.06	91.99	1.61	98.48	8.98	73.44	18.30	40.15	11.28	82.81
	IsomaxPlus + CE	**68.21**	39.02	59.91	**24.88**	**19.34**	1.47	70.71	9.00	25.73	19.34	74.68
	IsomaxPlus + Focal Loss	63.48	**47.07**	**47.07**	2.25	46.83	0.98	74.90	10.21	34.06	18.33	73.16

calibration.

c) Combination with CE and FocalLoss: We then combined both IsoMaxPlus and Logit Normalization with CE and Focal Loss. Both losses incorporate Cross-Entropy in their default format, therefore we didn't need to modify anything for CE. To implement FocalLoss[4] instead we had to apply minor changes, for IsoMaxPlus we removed the second part (which included the Cross-Entropy) and applied the FocalLoss , using $\gamma = 2.0$.

Also for LogitNorm we applied FocalLoss to its output.

2) Anomaly Detection Performance Across Metrics: When comparing the results in Table IV across the three anomaly detection metrics (Maximum Softmax Probability (MSP), MaxLogit, and MaxEntropy), distinct trends emerge:

- MaxLogit and MaxEntropy deliver superior results in terms of AuPRC and FPR95 across most datasets. These methods leverage the ability of advanced loss functions to push the confidence of anomalies further away from in-distribution predictions, thus enabling better anomaly separability.
- On the other hand, MSP consistently underperforms, except for RoadAnomaly. This indicates its limited robustness in detecting anomalies, as it relies solely on softmax probability, which often fails to adequately distinguish between in-distribution and out-of-distribution samples.

3) Best Loss Function Combination: The best loss is the IsomaxPlus, it achieves the best results in general, outperforming Logit Normalization. The combination of IsomaxPlus with Focal Loss generally emerges as the best-performing configuration for anomaly detection. This pairing consistently reduces FPR95 while maintaining competitive AuPRC scores across all datasets. The strength of this approach lies in the complementary mechanisms of the two losses:

- IsomaxPlus Loss encourages isotropic features by pushing the representations of in-distribution samples closer together while maximizing the separation of anomalous data. This results in improved anomaly confidence and separation.
- Focal Loss emphasizes the learning of difficult samples, which is particularly crucial for detecting rare anomalies and improving overall sensitivity to out-of-distribution regions.

Together, these losses enhance the model's ability to correctly identify anomalies while maintaining strong segmentation performance, as demonstrated by low FPR95 values (e.g., 47.07% on SMIYC RA-21 and 50.35% on FS-L&F) and competitive AuPRC scores . The results demonstrate that the choice of loss function significantly impacts both segmentation and anomaly detection performance. The best mIoU is given by the combination of IsoMaxPlus and Cross-Entropy.

VIII. CONCLUSION

Real-time anomaly segmentation for road scenes is a challenging task that requires balancing speed, accuracy, and the ability to detect unexpected objects. Our study shows that while many models perform well on standard segmentation tasks, their ability to identify anomalies varies significantly.

The choice of detection method plays a crucial role. Approaches that avoid overconfidence, such as using raw model outputs instead of softmax probabilities, prove more reliable for spotting anomalies. Architectural design also matters: models that combine detailed spatial information with broader context tend to detect irregularities better, even if they sacrifice some general segmentation accuracy.

Calibration methods, like temperature scaling, can improve performance, but their effectiveness depends heavily on the type of anomalies present. This suggests that adaptive calibration strategies are needed to handle diverse scenarios.

Finally, the loss function used during training has a significant impact. Combining distance-based feature learning with a focus on hard-to-classify examples enhances both segmentation and anomaly detection.

In summary, building robust real-time systems for anomaly segmentation requires a careful balance of efficient architectures, adaptive calibration, and uncertainty-aware training. These insights pave the way for safer and more reliable vision systems in applications like autonomous driving.

[4]https://github.com/clcarwin/focal_loss_pytorch/blob/master/focalloss.py

REFERENCES

[1] Adam Paszke, Abhishek Chaurasia, Sangpil Kim, Eugenio Culurciello, "ENet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation", arXiv:1606.02147, 2016

[2] E. Romera, J. M. Álvarez, L. M. Bergasa and R. Arroyo, "ERFNet: Efficient Residual Factorized ConvNet for Real-Time Semantic Segmentation," in IEEE Transactions on Intelligent Transportation Systems, vol. 19, no. 1, pp. 263-272, Jan. 2018, doi: 10.1109/TITS.2017.2750080.

[3] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, Nong Sang, "BiSeNet: Bilateral Segmentation Network for Real-time Semantic Segmentation", arXiv:1808.00897, 2018

[4] Dan Hendrycks, Steven Basart, Mantas Mazeika, Andy Zou, Joe Kwon, Mohammadreza Mostajabi, Jacob Steinhardt, Dawn Song, "Scaling Out-of-Distribution Detection for Real-World Settings", arXiv:1911.11132, 2019

[5] David Macêdo, Teresa Ludermir, "Enhanced Isotropy Maximization Loss: Seamless and High-Performance Out-of-Distribution Detection Simply Replacing the SoftMax Loss", arXiv:2105.14399, 2022

[6] Hongxin Wei, Renchunzi Xie, Hao Cheng, Lei Feng, Bo An, Yixuan Li, "Mitigating Neural Network Overconfidence with Logit Normalization", arXiv:2205.09310, 2022

[7] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, Piotr Dollar, "Focal Loss for Dense Object Detection", arXiv:1708.02002, 2018

[8] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The Cityscapes Dataset for Semantic Urban Scene Understanding," in Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[9] Blum, H., Sarlin, PE., Nieto, J. et al. The Fishyscapes Benchmark: Measuring Blind Spots in Semantic Segmentation. Int J Comput Vis 129, 3119–3135 (2021). https://doi.org/10.1007/s11263-021-01511-6

[10] Robin Chan, Krzysztof Lis, Svenja Uhlemeyer, Hermann Blum, Sina Honari, Roland Siegwart, Pascal Fua, Mathieu Salzmann, Matthias Rottmann, "SegmentMeIfYouCan: A Benchmark for Anomaly Segmentation", arXiv:2104.14812, 2021