Real-Time Anomaly Segmentation for Road Scenes

Thomas Baracco Riccardo Renda Andrea Sillano

Politecnico di Torino

{s308722, s310383, s314771}@studenti.polito.it

Abstract

Detecting Unknown/Anomaly/Out-of-distribution (OoD) objects is paramount in real-world applications, especially in autonomous driving and various computer vision domains like continual learning and open-world recognition. These objects, not seen during the training phase, pose significant challenges to current models, often resulting in poor performance when they encounter something unexpected.

The criticality of this issue becomes more pronounced in autonomous driving, where the ability to identify and segment anomalies in real-time can be the difference between a safe journey and a potential accident. An autonomous vehicle must be equipped to recognize anything from unexpected road obstacles to unusual weather conditions, ensuring it can make informed decisions without human intervention. Similarly, in the field of computer vision, continual learning and open-world problems require models that can adapt to new information continuously. The ability to detect and segment anomalies is crucial for these models to learn from new data without forgetting previously acquired knowledge, thus evolving.

Given these challenges, our objective is to develop compact anomaly segmentation models capable of real-time deployment in edge computing scenarios. These models must be lightweight enough to fit on small devices, like smart cameras with limited memory and processing capabilities. This constraint is not just a technical requirement but a necessity for widespread adoption in real-world applications, where resources on edge devices are often scarce.

The development of such models opens up a plethora of opportunities for safer and more reliable autonomous systems. By equipping devices with the ability to recognize and respond to the unexpected, we move closer to truly autonomous solutions capable of operating safely and effectively in the complex and unpredictable real world.

The citation [6] contains the details of the original project from which we have derived our work.

The code of the project is provided on the following github repository:

https://github.com/AML-project-AnomalySegmentation/AnomalySegmentation_CourseProjectBaseCode.

1. Introduction

Current models demonstrate proficiency in classification tasks but falter when encountering unknown data or objects not present during their training. This limitation poses significant risks in real-world applications, such as autonomous vehicles, where the stakes involve human safety. To address this challenge, models equipped with anomaly detection through segmentation are proposed as a solution. This approach aims to identify and categorize unusual patterns within data by segmenting it into distinct regions, introducing a new class for out-of-distribution (OOD) objects and labeling unknown entities as anomalies during the training phase.

Among the leading lightweight models for semantic segmentation are ENet [4], ErfNet [7], and BiseNet [9].
These models have been pretrained using the Cityscapes dataset [1], and their performance will be evaluated using the Road Anomaly, Road Obstacle, and Fishyscapes datasets.

The initial phase of this research will focus on assessing the performance of ErfNet by computing metrics such as mean Intersection over Union (mIoU) and False Positive Rate at 95% (FPR95). Subsequent analyses will compare ENet and BiseNet against a version of ErfNet retrained to classify the background as an anomaly.
Furthermore, this study will delve into the impact of incorporating loss functions tailored for anomaly detection into the training process. The losses under consideration include:

- Enhanced Isotropy Maximization Loss
- Logit Normalization Loss

Additionally, the synergy of these specialized losses with more traditional ones, such as focal loss and cross-entropy loss, will be examined. The objective is to ascertain whether these combined loss strategies can further refine the model's ability to discern and segment anomalies, thereby elevating its overall performance and reliability in critical applications. This comprehensive approach to training and evaluation seeks to advance the field of semantic segmentation and anomaly detection, paving the way for safer, more reliable machine learning applications in environments where the unexpected is the norm.

2. Experiment Setup

2.1. Segmentation Models

The principal models used in this project for anomaly segmentation tasks are ERFNet, ENet, and BiSeNet, each with unique features tailored for efficient real-time semantic segmentation:

- **ERFNet [7] (Efficient Residual Factorized Network):** ERFNet stands out for its blend of efficiency and performance in semantic segmentation. This network architecture (Figure 2) utilizes efficient residual factorized convolutions to strike a balance between computational speed and accuracy. Its design caters to the demanding requirements of real-time applications such as autonomous vehicles and robotics, where rapid and reliable segmentation is essential.

- **ENet [4] (Efficient Neural Network):** Designed for speed and efficiency, ENet is a compact convolutional neural network architecture optimized for semantic segmentation tasks that require real-time processing. It achieves an optimal balance between performance and computational load through the use of asymmetric convolutions, strategic pooling, and skip connections. This makes ENet an ideal choice for deployment in environments constrained by computational resources, including mobile devices and edge computing platforms.

- **BiSeNet [9] (Bilateral Segmentation Network):** BiSeNet is engineered specifically for high-speed semantic segmentation, featuring a dual-branch structure that captures both spatial details and contextual information efficiently. The spatial branch focuses on fine-grained details, while the context branch extracts broader contextual cues, enabling the network to provide comprehensive scene understanding. This dual approach ensures that BiSeNet can deliver detailed and semantically rich segmentation results in real-time applications, where understanding both the micro and macro aspects of a scene is critical.

These models represent the forefront of anomaly detection and segmentation in real-time applications, offering a range of solutions tailored to different requirements, from the need for speed and efficiency in ERFNet and ENet to the detailed and context-aware segmentation capabilities of BiSeNet.

2.2. Metrics

In order to evaluate the performance of the model, we took advantage of different kind of metrics.

- **mIoU**: mean Intersection over Union, it is a metric used to evaluate the performance of image segmentation algorithms. It measures the similarity between the predicted segmentation mask from a model and the ground truth segmentation mask. The Intersection over Union (IoU) is calculated for each class, and then the average is taken across all classes. The formula is given by:

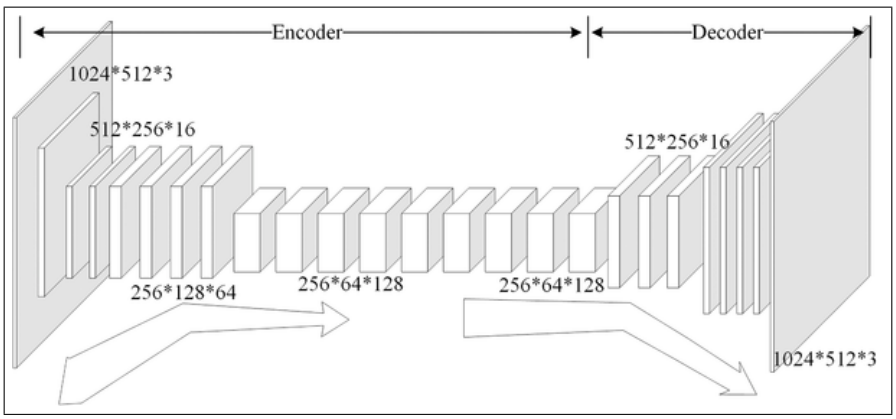
$$mIoU = \frac{1}{N} \sum_{i=1}^{N} \frac{TP_i}{TP_i + FP_i + FN_i} \tag{1}$$

  where:

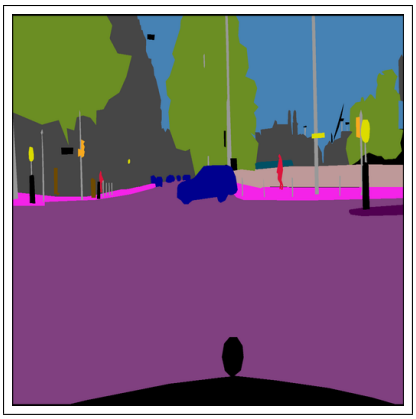
  - N is the total number of classes
  - TP_i is the number of pixels correctly predicted as class i
  - FP_i is the number of pixels erroneously predicted as class i
  - FN_i is the number of pixels belonging to class i but not predicted correctly

  mIoU returns a percentage score, where a higher value indicates better segmentation quality

- **auPRC**: The area under the Precision-Recall Curve (auPRC) is a metric used to evaluate the performance of a binary classification model. It assesses the trade-off between precision and recall across different classification thresholds.
  In a Precision-Recall Curve, precision is plotted on the y-axis, and recall (sensitivity) is plotted on the x-axis. The curve is generated by varying the decision threshold of the model, and for each threshold, precision and recall values are calculated. The area under this curve (auPRC) provides a summary measure of the model's ability to balance precision and recall across different operating points.
  The higher the auPRC, the better the model is at achieving high precision while maintaining high recall, it returns a percentage. A value of 100 represents a perfect model, while a value near to 0 indicates poor performance



(a) Complete architecture of ErfNet [2]



(b) Segmentation mask of cityscape [1]

Figure 1. ErfNet

		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	72.20	14.59	95.09	0.72	94.77	0.25	95.83	1.98	95.26	9.43	95.30
Max Logit	72.20	13.19	97.02	1.15	86.82	0.21	96.44	1.65	96.15	8.71	93.76
Max Entropy	72.20	14.31	96.72	0.83	94.83	0.22	96.79	1.95	94.05	9.10	95.31

Table 1. Results table 1

- **FPR95TPR**: The False Positive Rate at 95% (FPR95) is a performance metric used to evaluate binary classification models, particularly in scenarios where high true positive rates are critical. It represents the rate of false positives when the true positive rate is fixed at 95%.
  FPR95 measures the proportion of incorrectly classified negative instances (false positives) when the model achieves a true positive rate of 95%. This metric is useful when there is a need to control false positives at a specific level, such as in applications where minimizing the number of false positives is crucial

2.3. Dataset

The dataset employed for training the models in this study is Cityscapes, a comprehensive collection designed specifically for urban scene understanding and semantic urban scene parsing. Cityscapes is distinguished by its large scale and the detailed annotations it provides for various urban scenarios. The dataset has been instrumental in advancing the development and evaluation of semantic segmentation models due to its complexity and diversity.

For the experiments conducted in this research, two specific components of the Cityscapes dataset were utilized:

- **gtFine_trainvaltest**: This package contains fine-grained annotations for the training and validation sets, encompassing a total of 3475 images with detailed annotations. These annotations are crucial for training models to recognize and understand a wide variety of urban elements and scenarios accurately. Additionally, the package includes dummy annotations (ignore regions) for the test set, which comprises 1525 images. These dummy annotations are used to indicate areas in the images that should not be considered during the evaluation phase, typically because they do not contain relevant information for the task at hand or are ambiguous.

- **leftImg8bit_trainvaltest**: Complementing the annotations, this package provides the actual visual data with 5000 left 8-bit images across training, validation, and test sets. These images capture the complexity of urban environments in various conditions and are essential for training models to visually interpret urban scenes.

The combination of these packages offers a robust framework for training and evaluating models on semantic segmentation tasks. By leveraging the detailed annotations and high-quality images provided in the Cityscapes dataset, this study aims to enhance models' capabilities in detecting anomalies and understanding urban environments, which is critical for applications such as autonomous driving and urban planning.

3. Baselines

In this phase of the study, our attention is specifically directed towards the ErFNet model, utilizing a version pretrained on the Cityscapes dataset. We undertake a series of anomaly detection tests using this pretrained model in conjunction with a designated anomaly segmentation test dataset. The objective is to ascertain the model's performance in identifying anomalies through the computation of key metrics: mean Intersection over Union (mIoU), Area under the Precision-Recall Curve (AuPRC), and the False Positive Rate at 95% Recall (FPR95). To achieve this, we apply three distinct anomaly inference methods: Maximum Softmax Probability (MSP), Maximum Logit, and Maximum Entropy. Each method offers a unique approach to quantifying the model's ability to segment anomalies, providing a comprehensive understanding of its effectiveness in various scenarios.

3.1. OOD Detectors

The goal of these inferences is to conduct a comparative analysis of three distinct Out-of-Distribution (OOD) detection methodologies: Maximum Softmax Probability (MSP), Maximum Logits, and Maximum Entropy. This comparison aims to elucidate the strengths and weaknesses of each approach in identifying and handling OOD data, thereby determining their efficacy and applicability in various anomaly detection contexts.

3.1.1 MSP

The baseline method in our comparison is the Maximum Softmax Probability (MSP), which utilizes the negative value of the maximum softmax probability as an indicator of anomaly. The anomaly score is formulated as follows:

$$\text{softmax}_p = -\max_k \frac{e^{f(x)_k}}{\sum_i e^{f(x)_i}}, \tag{2}$$

where $f(x)$ represents the unnormalized logits from the classifier f.
MSP is noted for its effectiveness in smaller-scale applications; however, its performance can be hindered in larger-scale settings. This limitation arises because the softmax probability may be distributed across a wide range of classes, leading to a dilution of the confidence level that the model has regarding in-distribution examples. Consequently, in scenarios involving a large number of classes, the task of distinguishing in-distribution instances becomes more challenging. This challenge is not necessarily due to the novelty or unfamiliarity of the images but rather the dispersion of probabilities across the extensive class space, complicating the detection of in-distribution examples.

3.1.2 Max Logit

To mitigate the limitations faced by the Maximum Softmax Probability (MSP) detector in large-scale applications—where it may yield low confidence predictions—a MaxLogit detector has been proposed. This approach calculates the anomaly score using the negative maximum of the unnormalized logits:

$$\text{maxLogit} = -\max_k f(x)_k \tag{3}$$

Given that the logits remain unnormalized, this scoring mechanism is not influenced by the total number of classes. This characteristic makes the MaxLogit detector more adaptable and effective for large-scale applications, as it avoids the dilution of confidence that can occur when probabilities are spread across many classes. By directly relying on the raw, unnormalized logits, the MaxLogit approach offers a robust alternative for anomaly detection in environments with a broad class spectrum.

3.1.3 Max Entropy

The MaxEntropy detector leverages the principle that low entropy scores correlate with high-confidence predictions, whereas high entropy scores are indicative of low confidence. This detector's anomaly score is calculated as follows:

$$\text{maxEntropy} = -\frac{1}{N_c} \sum_{k,i} p(x)_{k,i} \cdot \log(p(x)_{k,i}) \tag{4}$$

Here, $p(x)$ represents the softmax probabilities of the unnormalized logits from the classifier, and N_c is the number of classes.

3.1.4 Inferences

The results of the analysis using the three different detectors (MSP, MaxLogit, and MaxEntropy) are summarized in Table 1. All tests were conducted using the ErfNet model pretrained on the Cityscapes dataset. The evaluation metrics include mean Intersection over Union (mIoU), computed on the Cityscapes dataset, and Area under the Precision-Recall Curve (AuPRC) and False Positive Rate at 95% Recall (FPR95), both computed on the Road Anomaly test dataset.

The results indicate that the performance of the three detectors is comparably similar. This observation could be attributed to the limited size of the test dataset, which may not provide a sufficiently diverse range of examples to highlight the distinct advantages and disadvantages of each method.

		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	72.20	14.59	95.09	0.72	95.77	0.26	95.83	1.98	95.26	9.43	95.30
MSP (t=0.5)	72.20	14.67	95.05	0.70	94.89	0.27	95.41	2.02	95.18	9.61	95.17
MSP (t=0.75)	72.20	14.63	95.07	0.71	94.83	0.26	95.62	2.00	95.23	9.51	95.24
MSP (t=1.1)	72.20	14.57	95.10	0.72	94.75	0.26	95.91	1.98	95.27	9.40	95.32
MSP (best t = 2.13)	72.20	14.44	95.20	0.76	94.53	0.24	96.68	1.94	95.20	9.20	95.41

Table 2. Results table 2

3.2. Temperature scaling

The pursuit of an optimal temperature value is part of our methodology to enhance the anomaly segmentation capabilities of neural networks through confidence calibration. Temperature scaling, a technique for calibrating the confidence levels of a classifier's predictions, plays a pivotal role in this process. It aims to align the network's confidence scores with the actual likelihood of prediction correctness. For instance, if a network assigns a confidence level of 80% to 100 predictions, we expect 80 of those predictions to be accurate. This alignment indicates that the network is well-calibrated.

Temperature scaling acts as a post-processing step, enabling a more reliable interpretation of the neural network's output probabilities. It adjusts the confidence scores by dividing the logits (the inputs to the softmax function) by a learned scalar parameter, T, referred to as the temperature. The adjusted softmax probability is given by:

$$\text{softmax}_p = -\max_k \frac{e^{\frac{f(x)_k}{T}}}{\sum_i e^{\frac{f(x)_i}{T}}}, \tag{5}$$

where T is the temperature parameter used to scale the logits, $f(x)_k$ represents the logits for class k , and the denominator is the sum of the exponentiated, scaled logits for all classes. The goal of adjusting the softmax function in this manner is to find a temperature value (T) that optimizes the model's anomaly segmentation performance, ensuring that the confidence levels it produces are both accurate and trustworthy. This calibration process is crucial for applications where precise risk assessment and decision-making are based on the model's predictions.

3.2.1 Method

The primary goal is to identify the optimal temperature parameter, T, that minimizes the cross-entropy loss [5]. This involves a two-step process where the logSoftMax function is applied to the Maximum Softmax Probability (MSP) values, followed by the computation of the Negative Log Likelihood (NLL).

3.2.2 Negative Log Likelihood

The NLL is conceptualized as the empirical estimate of the negative expected value, $-\mathbf{E}_{Q(x,y)}[\log P(y|x)]$, which can also be interpreted through the distribution function $S_y(x)$ of a deep neural network (DNN):

$$\text{NLL} = -\sum_{(x_i,y_i)} \log(S_y = y_i(x_i)), (x_i, y_i) \sim Q(x, y) \tag{6}$$

Here, $\mathbf{E}$ represents the expectation over the distribution $Q(x, y)$, and the goal is to minimize this expected negative log probability, which ideally occurs when the predicted probability distribution $P(y|x)$ aligns with the true distribution $Q(y|x)$.

To minimize the calibration error, the NLL must be minimized with respect to the parameters that adjust the $S_y(x)$ function. This effectively means reducing the discrepancy between the DNN's output confidence—achieved through applying the logSoftMax function on the MSP values—and the true distribution $Q(y|x)$.
This approach ensures that the model's confidence scores accurately reflect the actual probability of prediction correctness, thereby enhancing the model's reliability for anomaly detection tasks. The calibration process, specifically through temperature scaling, aims to adjust the model's outputs to closely match the true class distribution, thereby improving the model's overall performance and confidence in its predictions.

3.2.3 Inferences

Table 2 presents the outcomes of experiments conducted solely with the Maximum Softmax Probability (MSP) detector, while systematically varying the temperature parameter. These inferences were carried out on the ErfNet model, which had been pretrained on the Cityscapes dataset. The evaluation metrics remain consistent with those used in the previous step.

The findings from these experiments indicate that variations in the temperature parameter have minimal impact on the performance metrics. Specifically, even the optimal temperature identified by the optimizer ($t = 2.13$) yields re-

sults that are closely aligned with those obtained using other temperature values. This observation suggests that while temperature scaling is a method for confidence calibration, its effect on enhancing the anomaly segmentation capabilities of the network, at least in this context with the MSP detector and the chosen datasets, is limited.

4. Void Classifier

In this section, we shift our focus to treating the 'void' class as an anomaly and embark on a comparative analysis between the ENet and BiSeNet networks against ErFNet. The Cityscapes dataset comprises 19 defined classes alongside a 'void' class, which encapsulates segments of the background not categorized into any of the 19 classes. Our strategy involves retraining the three models—ENet, BiSeNet, and ErFNet—with a revised perspective, where the background or 'void' class is explicitly recognized as an anomaly. Subsequently, anomaly detection inferences will be conducted by exclusively considering the outputs pertaining to the 'void' class.

This approach allows us to directly evaluate the efficacy of each network in isolating and identifying anomalous regions represented by the 'void' class, thus enabling a precise assessment of their anomaly detection capabilities. By retraining the models to classify the background as an anomaly, we aim to enhance their sensitivity to out-of-distribution data, providing a robust mechanism for anomaly detection in scenarios where distinguishing between the usual classes and undefined background elements is crucial. This comparative analysis will shed light on the relative strengths and weaknesses of ENet, BiSeNet, and ErFNet in handling anomaly detection tasks, particularly in the context of complex environments like those depicted in the Cityscapes dataset.

4.1. Setup

For this study, we utilized models pre-trained on the Cityscapes dataset, implementing a novel approach to weight initialization. Traditionally, weights are hard-coded; however, in this instance, they were dynamically calculated based on the distribution histogram of the dataset. This process involved counting and normalizing the occurrence of labels for each class.

A crucial modification was applied to the initialization of the weight for the 19^{th} class, corresponding to the 'void' category. Traditionally set to 0, which effectively ignored the 'void' class during training, we adjusted this weight to 1. This change ensures the 'void' class is factored into the training process, encouraging the models to learn and make inferences about these areas, thereby incorporating anomaly detection into their segmentation capabilities.

4.2. Training

Following the initial setup, each model underwent training adjustments. Due to the limitations imposed by the GPU resources available on Google Colab, the training was conducted over three epochs. This constraint necessitated a focused and efficient approach to model optimization.

4.3. Inferences

The evaluation of anomaly detection capabilities for ENet, ErFNet, and BiSeNet, as detailed in Table 3, provides insightful findings into their performance across various metrics, including mean Intersection over Union (mIoU), Area under the Precision-Recall Curve (AuPRC), and False Positive Rate at 95% Recall (FPR95), utilizing the previously established testing dataset.

A significant variance in mIoU scores was observed, with ENet notably underperforming in comparison to ErFNet and BiSeNet. This discrepancy could potentially be attributed to the limited number of training batches for ENet, suggesting that an increase in batch size could lead to improved mIoU outcomes for this model. In contrast, the differences in AuPRC and FPR95 scores among the models were minimal. This indicates a level of consistency in their overall anomaly detection performance, despite some variations in their precise ability to segment and recognize the 'void' class as an anomaly. This consistency across models, especially in terms of AuPRC and FPR95 metrics, underscores their reliability in anomaly detection tasks, albeit with room for optimization in segmentation accuracy, particularly for ENet through adjustments in training batch size.

5. Effect of Training Loss function

In our pursuit to enhance anomaly segmentation capabilities, we investigate the integration of specific loss functions designed to improve anomaly detection during the detection process.
This exploration involves evaluating the impact of two distinct loss functions:

- **Enhanced Isotropy Maximization Loss**
- **Logit Normalization loss**

These will be assessed alone or in conjunction with the focal loss or cross-entropy loss to determine their combined effect on model performance.

5.1. Enhanced Isotropy Maximization Loss

Current methods for out-of-distribution (OOD) detection often come with significant challenges, such as the need for outlier data collection and extensive hyperparameter tuning, which can lead to decreased classification accuracy and

		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	2.48	13.97	95.88	0.59	97.62	0.40	91.24	1.44	97.88	7.44	96.60
ERF-Net	56.22	12.37	97.77	0.93	87.84	0.32	98.31	1.59	98.70	8.15	95.26
BiSeNet	41.15	12.98	88.95	2.87	99.44	0.16	92.77	1.25	98.36	8.41	92.68

Table 3. Results table 3

		SMIYC RA-21		SMIYC RO-21		FS L&F		FS Static		Road Anomaly	
Loss	**mIoU**	**AuPRC**	**FPR95**	**AuPRC**	**FPR95**	**AuPRC**	**FPR95**	**AuPRC**	**FPR95**	**AuPRC**	**FPR95**
Logit	47.46	16.34	94.70	0.97	91.08	0.29	94.16	1.77	94.66	10.95	94.37
Logit + cross	56.16	13.69	95.74	0.91	94.05	0.28	95.27	1.81	96.03	10.08	94.63
Logit + focal	41.15	12.98	88.95	2.87	99.44	0.16	92.77	1.25	98.36	8.41	92.68
IsoMax	2.2	14.80	95.18	0.67	95.08	0.28	95.14	2.08	95.26	9.85	95.11
IsoMax + cross	46.80	15.02	94.89	0.59	96.37	0.29	94.53	1.80	95.97	10.15	94.70
IsoMax + focal	49.92	14.99	95.00	0.66	95.17	0.28	94.95	1.93	95.18	9.81	95.20

Table 4. Results table 4

slower inference. A promising solution to these issues is the entropic out-of-distribution detection method, which leverages the IsoMax loss during training and the entropic score for OOD instance identification. This approach stands out for its simplicity and effectiveness, eliminating the need for model architecture or training procedure modifications.

The IsoMax [3] loss acts as a direct replacement for the traditional SoftMax loss, which combines the output linear layer, SoftMax activation, and cross-entropy loss. The replacement is straightforward, requiring no changes to existing model frameworks or hyperparameter settings. Additionally, the entropic score, which replaces the minimum distance score, simplifies the detection process by eliminating the need for hyperparameter tuning, thanks to its reliance on a global constant entropic scale.

The mathematical formulation of the IsoMax loss is as follows:

$$\mathcal{L}_{IsoMax} = -\frac{1}{N}\sum log\frac{e^{-E_s||f_\theta(x)-p_\phi^k||}}{\sum_j e^{-E_s||f_\theta(x)-p_\phi^j||}} \quad (7)$$

where

- N is the number of batch
- E_s represents the entropic scale
- $||f_\theta(x)-p_\phi^k||$ measures the distance in the IsoMax loss context

The introduction of normalization for both the features $f_\theta(x)$ and the weights p_ϕ^j, denoted as $\hat{f_\theta(x)}$ and $\hat{p_\phi^j}$ respectively, addresses the bias towards classes with low-norm prototypes being mistakenly favored as OOD examples. The features $f_\theta(x)$ and the weights p_ϕ^j are not normalized; due to that, examples from classes that present prototypes with low norms are unjustifiably favored to be considered OOD examples for the same reason. The enhancement consist of replacing $f_\theta(x)$ with its normalized version given by $\hat{f\theta(x)} = f\theta(x)/||f\theta(x)||$.

Furthermore, the inclusion of a distance scale d_s refines the calculation of isometric distances used by the enhanced IsoMax+ loss, leading to the updated loss function:

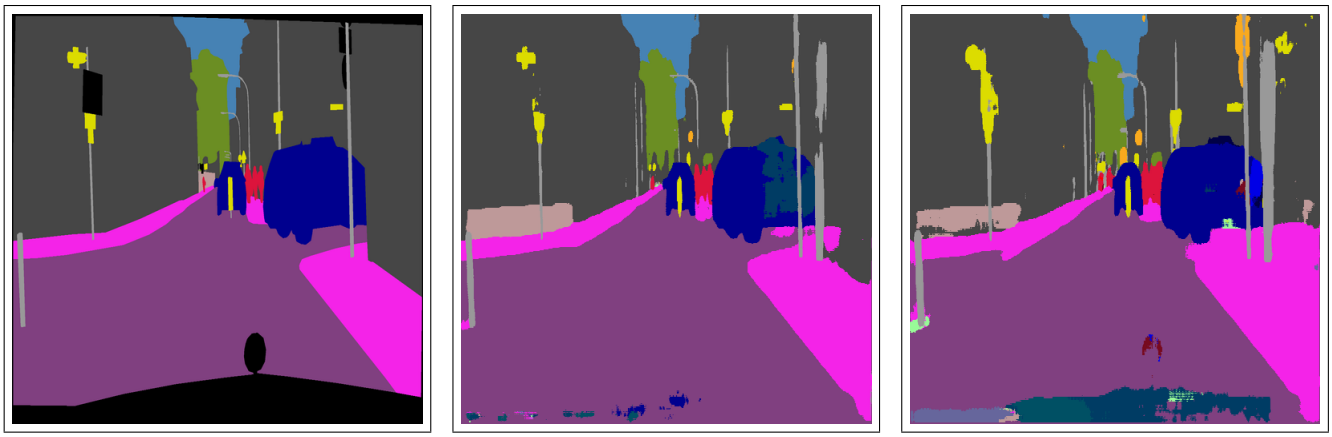
$$\mathcal{L}_{IsoMax} = -\frac{1}{N}\sum log\frac{e^{-E_s|d_s|||\hat{f\theta(x)}-\hat{p_\phi^j}||}}{\sum_j e^{-E_s|d_s|||\hat{f\theta(x)}-\hat{p_\phi^j}||}} \quad (8)$$

These modifications significantly boost OOD detection performance while preserving the IsoMax loss's advantages: no need for hyperparameter tuning, independence from outlier/background data, efficient inference, and no compromise on classification accuracy. This advanced approach presents a compelling solution for enhancing machine learning models' safety and reliability in real-world applications.

5.2. Logit Normalization loss

Detecting out-of-distribution (OOD) inputs is crucial for ensuring the safety of machine learning models in real-world applications. A significant challenge faced by neural networks is their tendency to exhibit overconfidence in their predictions, both for in-distribution and OOD inputs. This study introduces a novel strategy to combat this issue: Logit Normalization [8](LogitNorm). LogitNorm modifies the cross-entropy loss function to maintain a constant vector norm for the logits during training. This approach stems from the observation that the magnitude of the logits tends to increase over the course of training, leading to overconfident outputs from the network.

The key idea behind LogitNorm is to decouple the influence of the logits' magnitude from the network's optimiza-



(a) Orginal segmentation label (b) ErfNet pretrained segmentation (c) ErfNet - LogitNorm and Cross Entropy loss

Figure 2. Example segmentation outputs

tion process. This is achieved by applying the following normalization to the logits:

$$\text{normalized logits} = \frac{f(x)}{\|f(x)\|_2} \cdot T \tag{9}$$

where $f(x)$ are the logits output by the neural network before the softmax activation, $\|f(x)\|_2$ is the L2 norm of the logits, and T is a temperature parameter that adjusts the scale of normalization.

Implementing LogitNorm aims to adjust the neural network's confidence scores, sharpening the distinction between in-distribution and OOD inputs. Through extensive testing across a variety of benchmarks, LogitNorm has demonstrated a significant improvement in the model's ability to discern OOD inputs. Specifically, it has been shown to reduce the False Positive Rate at 95% Recall (FPR95) by up to 42.30%, highlighting LogitNorm's potential to address the overconfidence problem effectively and enhance the reliability of machine learning models in contexts where safety is paramount.

5.3. Inferences

The results of the anomaly detection inferences for the modified Loss functions on ErFNet are detailed in Table 4. The evaluation revisited metrics such as mean Intersection over Union (mIoU), Area under the Precision-Recall Curve (AuPRC), and False Positive Rate at 95% Recall (FPR95) on the previously used testing dataset. As we can see training the model with the added losses seems to improve the results on the majority of the dataset with the most promising result obtained by the model trained with Logit loss

6. Conclusion

In this project, we conducted an in-depth analysis of various models, such as ErFNet, across different detectors, and also explored the impact of retraining these models with distinct loss functions. The performance outcomes of using Maximum Softmax Probability (MSP), Maximum Logit (MaxLogit), and Maximum Entropy (MaxEntropy) detectors were found to be quite similar, indicating that these methods yield comparable results in our context. Additionally, adjustments in the temperature parameter did not notably influence the models' performance, suggesting that this factor might not be critical under the conditions tested. However, retraining the models led to noticeable improvements in their performance metrics, especially when the specific anomaly detection loss function were used. This highlights the potential benefits of selecting and experimenting with different loss functions to enhance model accuracy and efficiency in anomaly detection tasks.

6.1. Limitation

The retraining phase encountered constraints due to the limited GPU resources provided by Google Colab. For more optimal outcomes, a longer training duration would have been beneficial, suggesting that access to more robust computational resources could significantly enhance the results of our models.

References

[1] 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, 2016. 1, 3

[2] Chaoxian Dong. Image semantic segmentation method based on gan network and erfnet model. *The Journal of Engineering*, 2021, 03 2021. 3

[3] David Macêdo and Teresa Bernarda Ludermir. Improving entropic out-of-distribution detection using isometric distances and the minimum distance score. *CoRR*, abs/2105.14399, 2021. 7

[4] Adam Paszke, Abhishek Chaurasia, Sangpil Kim, and Eugenio Culurciello. Enet: A deep neural network architecture for real-time semantic segmentation. *CoRR*, abs/1606.02147, 2016. 1, 2

[5] Geoff Pleiss. Temperature Scaling, 2021. 5

[6] Shyam Nandan Rai. AnomalySegmentation_CourseProjectBaseCode, 2023. 1

[7] Eduardo Romera, José M. Álvarez, Luis M. Bergasa, and Roberto Arroyo. Erfnet: Efficient residual factorized convnet for real-time semantic segmentation. *IEEE Transactions on Intelligent Transportation Systems*, 19(1):263–272, 2018. 1, 2

[8] Hongxin Wei, Renchunzi Xie, Hao Cheng, Lei Feng, Bo An, and Yixuan Li. Mitigating neural network overconfidence with logit normalization, 2022. 7

[9] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation. *CoRR*, abs/1808.00897, 2018. 1, 2