Detecting Anomalies in Semantic Segmentation with Prototypes

Dario Fontanel[1], Fabio Cermelli[1,2], Massimiliano Mancini[3], Barbara Caputo[1,2]

[1]Politecnico di Torino, [2]Italian Institute of Technology, [3]University of Tübingen

{dario.fontanel, fabio.cermelli, barbara.caputo}@polito.it,

massimiliano.mancini@uni-tuebingen.de

Abstract

Traditional semantic segmentation methods can recognize at test time only the classes that are present in the training set. This is a significant limitation, especially for semantic segmentation algorithms mounted on intelligent autonomous systems, deployed in realistic settings. Regardless of how many classes the system has seen at training time, it is inevitable that unexpected, unknown objects will appear at test time. The failure in identifying such anomalies may lead to incorrect, even dangerous behaviors of the autonomous agent equipped with such segmentation model when deployed in the real world. Current state of the art of anomaly segmentation uses generative models, exploiting their incapability to reconstruct patterns unseen during training. However, training these models is expensive, and their generated artifacts may create false anomalies. In this paper we take a different route and we propose to address anomaly segmentation through prototype learning. Our intuition is that anomalous pixels are those that are dissimilar to all class prototypes known by the model. We extract class prototypes from the training data in a lightweight manner using a cosine similarity-based classifier. Experiments on StreetHazards show that our approach achieves the new state of the art, with a significant margin over previous works, despite the reduced computational overhead. Code is available at https://github.com/DarioFontanel/PAnS.

1. Introduction

For machines acting in the real world, it is of the utmost importance recognizing what objects are present in their surroundings, and where. To achieve this objective, multiple works have focused on the task of semantic segmentation [27, 4], where the goal is to assign to each pixel in the image its corresponding semantic label. However, semantic segmentation models are inherently limited by the classes they see annotated at training time. As vast as their training database might be, this limitation is a crucial weakness, as clearly it is not possible to capture in a static collection all the possible semantic classes a system might ever encounter. Ideally, we would like a segmentation model to recognize whether a pixel belongs to one of its known classes, or whether it belongs to an unseen category not in-
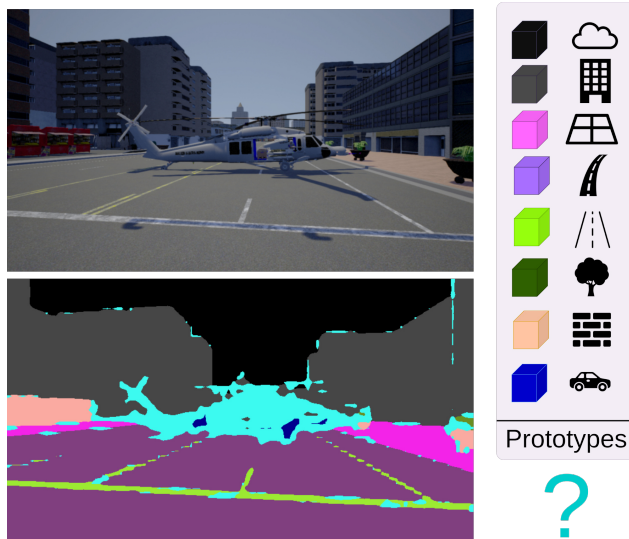

Figure 1: Anomaly segmentation (AS) aims to segment objects unseen to the model. Addressing AS is crucial, especially in autonomous driving applications, where confusing an anomalous object with known ones can be extremely dangerous. In this work, we address AS through prototype learning, where the anomalies (light-blue) are all regions unmatched with any class prototype learned by the model.

cluded in the training set. Note that this capability is important whenever pixels of unknown categories might be a threat for the machine (or the human) making use of a semantic segmentation module. As an example, since no semantic segmentation dataset contains labeled pixels for the class *helicopter*, if an autonomous driving system faces a scene as the one depicted in Figure 1 it will have no chances to avoid a fatal collision, unless it recognizes that there is something unforeseen in the image, *i.e.* an *anomaly*.

In this work, we focus on the problem depicted in Figure 1, namely anomaly segmentation (AS) [13, 37]. The goal of AS is to recognize whether a pixel in an image belongs to a category unseen during training, being unknown to the model. Previous works addressed this problem by either imposing a threshold on the predicted probabilities per pixel [16] or by means of generative approaches, comparing input images (or features) with their reconstructed versions [25, 37]. However, both strategies present some drawbacks.

Figure 2: Qualitative results of SPADE [31] reconstructions on an image from StreetHazards dataset [15] (top). The green box shows the anomaly, correctly not reconstructed by the model. The red box, instead, shows one of the artifacts that the generator introduces: the traffic lights are not reconstructed, thus being predicted as anomalies.

The first ignores that the softmax function blurs the confidence of the model regarding the presence of a certain class, *i.e.* after the softmax, two classes predicted with high scores (logits) have nevertheless equal and low probability).

On the other hand, generating images with high fidelity is particularly difficult in semantic segmentation due to their complex content. Thus, generative approaches tend to create artifacts not only when synthesizing pixels of unknown classes, but also pixels of known ones (see Fig. 2).

In this paper, we argue that it is fundamental to address the anomaly segmentation problem directly at the level of class scores. Our intuition is that if a model learns general but discriminative representation of each class, it can detect anomalies as pixels that are not compatible with any of the class representations. We concretely pursue this idea by means of class-specific prototypes, considering a pixel as anomalous whenever the highest matching score for the set of prototypes of the known classes is below a certain threshold. We train the prototypes using a cosine classifier that bounds the class specific scores while ensuring that the prototypes embody the average pixel features of the corresponding known class. It is important to emphasize that, to avoid the normalization problems of softmax-based approaches, we estimate the anomalies *directly* from the compatibility between a feature vector and the prototypes. We test our model PAnS on the popular StreetHazards bench-

mark [15], showing that it largely surpasses the state of the art.

Contributions. To summarize, our contributions are:

- We propose a novel perspective for the anomaly segmentation problem, revisiting the importance of class-specific scores rather than probabilities in the process of recognizing anomalous pixels.

- We present our approach, **P**rototypical **An**omaly **S**egmentation (PAnS), that computes such scores as the compatibility between a feature vector and class-specific prototypes, learning the latter as weights of a cosine classifier.

- Experiments on the widely adopted StreetHazards showing that our approach surpasses the previous state of the art by a margin.

2. Related works

In this section we review the topics that constitute the building blocks of our work, *i.e.* uncertainty estimation in semantic segmentation, out-of-distribution (OOD) detection, anomaly segmentation and prototype learning.

Semantic segmentation. Modern semantic segmentation architectures [27, 6, 46, 24, 45] are fully-convolutional encoder-decoder networks [27, 1] that differ on the strategy used to integrate contextual information in the pixel-level features. We can categorize these works as belonging mainly to two different approaches: pyramid-based approaches [46, 24, 5, 4, 45, 6], that integrate modules exploiting information at different scales, and attention-based approaches [40, 42, 41, 43, 9, 44] that aggregates the long-range spacial dependencies using attention modules at different levels. A drawback shared by all these architectures is that they require huge amount of training data, which is often time consuming and extremely expensive to collect. Moreover, they only consider an offline setting, *i.e.* once the model has been trained, it is not possible to integrate additional knowledge. Although recent works have tried to move forward and deal with the addition of novel classes [30, 3], none of these approaches deal with anomaly detection.

Out of distribution detection is a topic that has aroused growing interest in the machine learning community in recent years [16, 22, 18].

[16] has established the standard baseline for out-of-distribution (OOD) detection where a threshold applied on the maximum softmax probability (MSP) is used to recognize whether a sample belongs to the training distribution (in-distribution) or not (out-of-distribution). Beside its simplicity and effectiveness, the largest softmax probability is sub-optimal to detect anomalies for two reasons. First, the model may produce high probability values even when the

predictions are incorrect [7]. Second in case of correct predictions with low probabilities, the model may misinterpret them as OOD samples.

Other approaches have attempted to mitigate these issues. [19] preprocesses the training set in order to find class-specific set of images that contain only the training samples predicted with high similarity, filtering out the fraction of sparse samples that may be outliers. Given a test sample, the model detects anomalies through a modified nearest-neighbor classifier, computing the prediction as a ratio of distances between class sets. In [7], an additional neural network is trained on in-distribution data to produce high confidence values when the prediction of the model is correct. This additional architecture helps to model the confidence on the network predictions.

Previous approaches [10, 20] have also used Monte Carlo Dropout (MC-Dropout) for Bayesian approximation. With multiple forward pass, they compute the variance and the entropy as a measure of uncertainty.

ODIN [23] improves MSP by introducing a temperature scaling factor in the softmax operation and a small perturbation on the features before the classification step. The temperature value and the perturbation are computed on a OOD validation set. It is worth noticing that despite the apparent similarity, our work is conceptually different from [23]. First, [23] relies on a softmax function with temperature, while we directly compute the class scores as compatibility with the prototypes. Second, while [23] requires an OOD set, we train our model on in-distribution data only.

Anomaly segmentation. Current mainstream approaches for AS exploit pixel-wise reconstruction loss with auto-encoders (AEs) [2, 13]. The main disadvantage of these approaches is that when the training scene is very complex, as often is the case in roads or streets scenes, AEs are not able to model correctly the in-distribution and therefore they cannot guarantee to generate plausible in-distribution images from out-of-distribution regions. As a matter of fact, the CAOS benchmark [15] demonstrated that MSP [16] outperforms AEs and Bayesian network-based approaches. Recent works [25] and [37] proposed to compare test images containing OOD pixels with their reconstructed versions. In both cases, images are reconstructed from the predicted semantic maps by means of a generative approach, *i.e.* pix2pixHD [36] and SPADE [31] respectively. To detect the OOD pixels, [25] and [37] measure the discrepancy between the original and reconstructed images by means of a discrepancy network and a comparison module respectively. A drawback of these works is that artifacts in the reconstructions process may be wrongly recognized as anomalies, as shown in Fig. 2.

Differently from these works, we do not rely on expensive generative approaches and we directly produce predictions from the compatibility between features and class pro-

totypes. Our approach achieves better results despite being more lightweight and simple than generative-based ones.

Prototype Learning [39, 12, 29], as opposed to softmax-based CNN, learn a metric space in which labeling is achieved by measuring the distance between the test image and prototypes of of class. Recently, prototype learning has been exploited by few-shot [11, 32, 34, 39] and zero-shot [38, 26] learning methods. In this work, we take inspiration from [11, 32], using a cosine-similarity based classification layer that forces the classification weights to represent the class prototypes.

3. Anomaly Segmentation with Prototypes

In this section we first formalize the anomaly segmentation problem (Section 3.1), highlighting the limitations of softmax-based approaches. We then describe how we overcome these limitations with our model, Prototypical Anomaly Segmentation (PAnS), in Section 3.2. An illustration of PAnS is provided in Fig. 3.

3.1. Problem formulation

The goal of anomaly segmentation is to recognize which pixels in the image belong to anomalous objects, unseen during training. Let us denote as $\mathcal{X} \in \mathbb{R}^{|\mathcal{I}|}$ the image space, where $\mathcal{I}$ is the set of pixels. During training, we are given a dataset $\mathcal{T} = \{(x_k, y_k)\}_{i=k}^{N}$ where $x \in \mathcal{X}$ is an image and $y \in \mathcal{Y}$ is its corresponding ground-truth mask. As in standard segmentation, $\mathcal{Y}$ contains pixel-level annotations for a set of semantic classes $\mathcal{C}$, *i.e.* $\mathcal{Y} \in \mathcal{C}^{|\mathcal{I}|}$. Given $\mathcal{T}$, we want to learn a function f mapping an image to its corresponding anomaly score at pixel level, *i.e.* $f : \mathcal{X} \rightarrow \mathbb{R}^{|\mathcal{I}|}$. Without loss of generality, we consider f built on three components. The first is a feature extractor $\varphi : \mathcal{X} \rightarrow \mathcal{Z}$ mapping images into a feature space $\mathcal{Z} \subset \mathbb{R}^{|\mathcal{I}| \times d}$, with d being the feature dimensions. The second is a scoring function $\rho : \mathcal{Z} \rightarrow \mathbb{R}^{|\mathcal{I}| \times |\mathcal{C}|}$ mapping the features in $\mathcal{Z}$ to pixel-level class scores. The third is an anomaly score function $\sigma : \mathbb{R}^{|\mathcal{I}| \times |\mathcal{C}|} \rightarrow \mathbb{R}^{|\mathcal{I}|}$, mapping the class scores to the final anomaly ones. A core component of every anomaly segmentation algorithm is σ, that produces the final anomaly scores. In the following we discuss how previous approaches instantiated the σ function.

Maximum Softmax Probability (MSP). One of the most popular and effective approaches for anomaly segmentation is Maximum Softmax Probability (MSP) [16]. The intuition behind MSP is that the anomaly score of a pixel should depend on the highest probability assigned to any of the known classes. Given an image x and its pixel-level class scores $s = \rho(\varphi(x))$, MSP defines the anomaly score for pixel i, *i.e.* $\sigma_i(s)$, as:

$$\sigma_i(s) = 1 - \max_{c \in \mathcal{C}} \frac{e^{s_i^c}}{\sum_{k \in \mathcal{C}} e^{s_i^k}} \qquad (1)$$
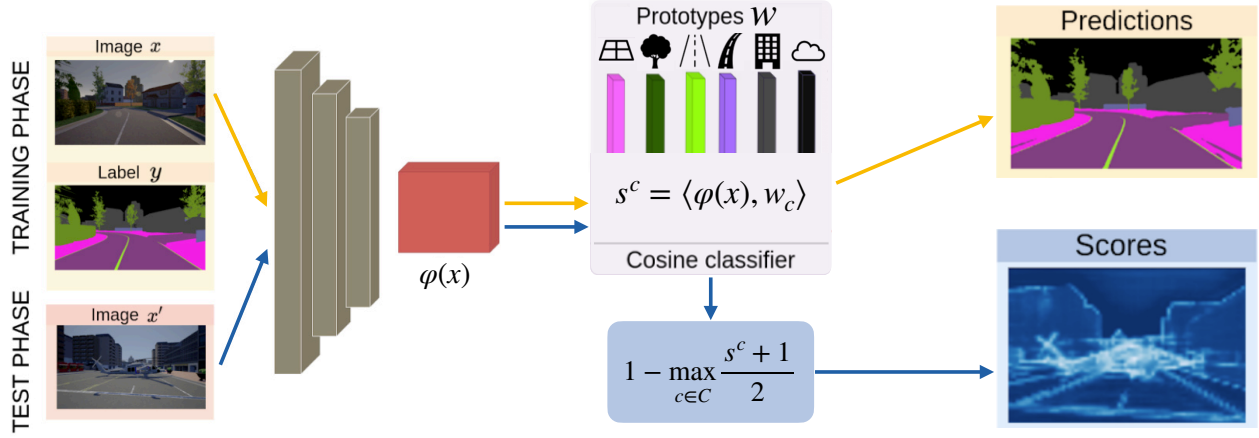
Figure 3: Overview of PAnS. During training, we learn class-specific prototypes using a cosine classifier that computes the cosine similarity between the features $\varphi(x)$ and the classifier weights w. When a test image arrives, we compute the network prediction on the known classes and we compute the anomaly score by means of the cosine classifier scores.

where $s_i^k = \rho_k^i(z)$ is the score for class k in pixel i. Note that anomaly scores are defined as the inverse of the maximum probability assigned to any known class in $\mathcal{C}$, with the probabilities computed through the softmax function. Despite its effectiveness, we argue that using softmax probabilities is not the best choice to estimate the anomaly scores. In fact, the softmax function may smooth the confidence of the model prediction on each pixel, leading to consider uncertain (and thus anomalous) pixels even with high predicted initial scores.

As a toy example, suppose we have two different classes and, given a pixel, the model produces a very high score for the first class and a very low one for the other one. In this case, the probabilities would have a low entropy after applying the softmax function, with a high probability for the class with the highest score and a low probability for the other. The model will correctly consider this pixel as not anomalous. However, if both classes have high but near scores, they will get close probability values after softmax normalization. This high entropy on the pixel prediction indicates that the model is uncertain of the semantic of the pixel, but the high initial scores may hint that the model *is not uncertain* that the pixel belongs to a known class.

Note that we will get this high entropy class scores regardless of the initial magnitude of the scores. Obviously, since the magnitude of the logits before the softmax are unbounded, it is not trivial to assess whether the model is uncertain on the semantic content due to the pixels features being out of distribution or not. In the following, we will show that keeping the class scores independent and constraining them to be bounded in a known range (*i.e.* $[-1, 1]$) through prototype matching is useful to assess the actual confidence of a model in its predictions, improving the recognition of anomalous pixels.

3.2. Prototypical Anomaly Segmentation

In the previous section, we discussed how MSP may fail in recognizing anomalous pixels due to the softmax normalization, which discards information about the confidence of the model. As a consequence, we take a different approach, arguing that it is critical to consider the confidence of each class separately. Ideally, we want to obtain confidence values which: (i) are independent for each class, (ii) do not require extra-computation, and (iii) are bounded in a certain range, such that it is possible to define a threshold on their scores to detect anomalies.

To accomplish this, we propose to represent each class using a prototype. Each class prototype may be considered as a reference feature vector for a certain class. We can then compute class-independent confidence scores by computing the similarity between the features of any pixel and the prototype itself. We take inspiration from few-shot classification learning works [32, 11], and we use a simple yet effective cosine classifier, which implicitly encodes the class-prototypes by means of its classification weights.

Cosine Classifier To effectively extract class-prototypes from the network, we use a cosine classifier, *i.e.* a classifier which uses cosine similarity between the input features and the class weights as class scores. While this classifier has been used for image classification [11, 28, 35, 17] to effectively learn class-prototypes, we are the first to use this classifier with the aim of recognizing anomalies in semantic segmentation. We replace the standard convolutional classifier, with a cosine similarity-based one. In particular, the classification scores for a class c, given an image x and a pixel i, are:

$$s_i^c = \rho_c^i(\varphi(x)) = \langle \varphi_i(x), w_c \rangle = \frac{\varphi_i(x)^\top w_c}{||\varphi_i(x)|| \, ||w_c||}, \quad (2)$$

where $\varphi_i(x)$ is the output of the feature extractor φ at pixel

i of the image x, and $w_c \in \mathbb{R}^d$ is the prototype of class c. We note that the scores s are in the range $[-1, 1]$ due to the normalization effect on the denominator.

To learn the prototypes, we apply the standard cross-entropy loss on the softmaxed probabilities computed on the scores s^c:

$$\ell_{CE}(x, y) = -\frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} \log \frac{e^{\tau s_i^{y_i}}}{\sum_{c \in \mathcal{C}} e^{\tau s_i^c}}, \quad (3)$$

where τ is a scalar value that scales the classification scores in the range $[-\tau, \tau]$ and y_i is the ground-truth at pixel i. Intuitively, minimizing the ℓ_{CE} loss forces the prototype weights to have low cosine distance with features of their respective class, representing them on average.

Computing the Anomaly Scores In this section, we describe how we use the prototypes to detect the anomalies in the input image. To overcome the limitations of softmax function, we argue that it is important to avoid the use of normalized probabilities, but rather to use directly the classification scores. Since the cosine classifier outputs the similarity of each weight with the visual features extracted from the network, it enables the use of the scores s as a confidence measure on the presence of a class. Moreover, we can exploit the fact that the scores of the classifier are bounded in the range $[-1, 1]$ and define the binary probability $\bar{\sigma}$ of a class c to appear in a pixel i of an image x as:

$$\bar{s}_i^c = \frac{s_i^c + 1}{2} \quad (4)$$

and the anomaly score σ using the maximal binary probability, which we define as:

$$\sigma_i(s_i) = 1 - \max_{c \in \mathcal{C}} \bar{s}_i^c. \quad (5)$$

Intuitively, σ_i will produce scores close to 1 when the visual features are far from all the class-prototypes, while close to 0 if at least the prototype of one class is close to them.

With this strategy, we expect that our method is able to effectively represent the known classes, having an high confidence on pixels belonging to them, while we expect no class prototype to be close to the extracted features for pixels of anomalous objects. Moreover, we avoid the aforementioned issues of the softmax function largely boosting the results as we will demonstrate in the experimental section.

4. Experiments

Dataset and baselines We conduct our experiments on the popular StreetHazards [15] database, which has been proposed within CAOS benchmark [15] as a synthetic dataset for anomaly segmentation. It contains 5125 training images with paired semantic labels, 1031 validation images without anomalies and 1500 test images with anomalies. As the
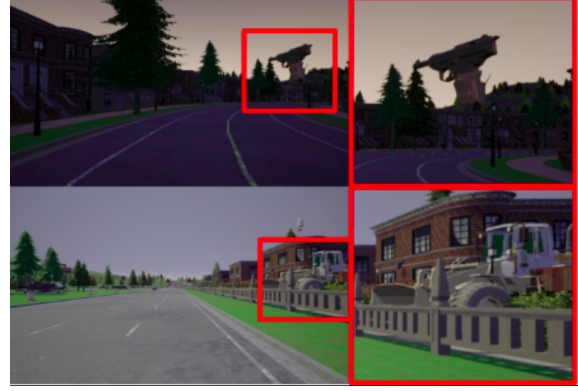

Figure 4: Examples taken from StreetHazards dataset [15].

authors used Unreal Engine along with CARLA simulator [8] to obtain the synthetic images, they selected different towns for the three splits. Test images contain anomalies randomly selected from a set of 250 objects. These objects are placed in the test images trying to reproduce plausible road scenarios. Figure 4 shows examples of images taken from the dataset.

On this benchmark we compare our method with state-of-the-art anomaly segmentation approaches, namely MSP [16], MSP + CRF [15], an auto-encoder (AE) based approach [2], Dropout [10], and the generative approach SynthCP [37].

Metrics. Following [15, 37, 25] we used as anomaly segmentation metrics AUPR, AUROC and FPR95 as they are widely used in out-of-distribution detection [18, 21]. AUPR measures the area under the Precision-Recall curve, AUROC measures the area under the TRP and FPR and FPR95 measures the FPR at 95% of recall. In each of these metrics, anomaly pixels have been considered as positives and all the others as negative.

4.1. Implementation details

Following [37], we used a ResNet-50 architecture [14] as a backbone and PSPNet [46] as the head module. We trained our segmentation module for 40 epochs, with batch size equal to 2 and a learning rate equal to 0.007. We used as learning rate decay policy the polynomial schedule with power equal to 0.9 and a weight decay equal to 0.0001. We also used InPlace-ABN [33] which allows to save up to 50% of GPUs memory. Same as [37], we used multiple scale evaluation at test time and at training time we performed random scale, random crop and random horizontal flip augmentation.

4.2. Comparison with the state of the art

The results of our comparison with the state of the art are reported in Table 1. As the table shows, in this scenario our method achieves the best performances by a margin, under AUROC and FPR95 OOD metrics, being instead

Method	AUPR ↑	AUROC ↑	FPR95 ↓
AE [2]	2.2	66.1	91.7
Dropout [10]	7.5	69.9	79.4
MSP [16]	6.6	87.7	33.7
MSP + CRF [15]	6.5	88.1	29.9
SynthCP [37]	**9.3**	88.5	28.4
PAnS	8.8	**91.1**	**23.2**

Table 1: Results on StreetHazards dataset [15] according to AUPR, AUROC and FPR95 metrics. [15].
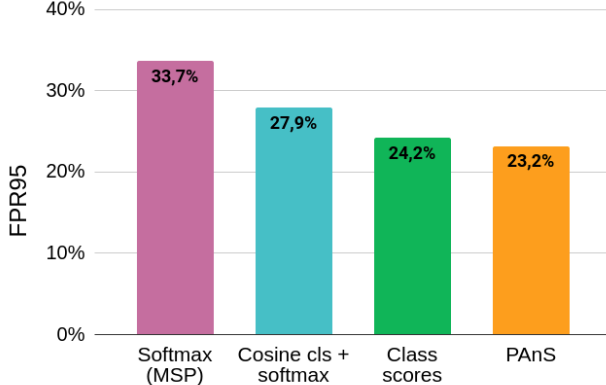

Figure 5: Difference between the direct usage of scores (both the scores produced by a standard classifier and a cosine-based one) on StreetHazard dataset [15].

comparable under AUPR values. Among all, noteworthy is the result of 23.2% achieved under the FPR95 metric which shows that our method is much less prone to confuse pixels of known classes as anomalies. This is due to the fact that our prototype-based classifier betters preserve the original scores for known classes which might be either smoothed by the softmax normalization (as in MSP) or overwritten by inaccurate generations (as in SynthCP). Indeed, under the FPR95 metric, our approach improves the state of the art (SynthCP) by almost +6%. The fact that our prototypes induce a classifier more robust against misclassification of known class pixels is reflected in the other metrics. Our approach reaches 91.1% of AUROC, improving over the best previous method, SynthCP, of 2.6%. These results confirm how our approach achieves the best trade-off between recognizing anomalous pixels while preserving high confidence predictions for pixels of known classes. On the other hand, SynthCP obtains a slightly better AUPR (+0.5 w.r.t. PAnS). However, we highlight that PAnS only requires a single forward pass on the network, without any generative step and without increasing the computation required by the model.

Note also that generative models might be affected by the quality of the generated images. Indeed, using synthetic images often introduces artifacts, as mentioned in [37]. These artifacts might hamper the performance of generative anomaly segmentation models, since they might be wrongly segmented as anomalies (*i.e.* see Fig. 2).

Qualitative results To analyze the impact of our cosine classifier and scores, we report in Fig. 6 some qualitative examples on anomaly scores produced by the softmax-based approach MSP [16] and our approach PAnS for randomly selected samples of StreetHazards. In the figure, white regions indicate higher anomaly scores while blue regions lower ones. As the figure shows, our model is able to correctly assign low scores to the regions where the anomaly is present (*e.g. helicopter* on the top image, *i.e. carriage* in the bottom), while MSP does not, covering only small portions of the anomalies. At the same time, both MSP and our approach tend to assign high anomaly scores to boundaries between known classes (*e.g. street lines* and *road* in the top, *building* and *sidewalk* in the bottom). We believe modeling this highly uncertain regions between two or more known classes, is an open problem for anomaly segmentation algorithms, which would be important to address in future works.

Ablation study of anomaly scores. We report in Fig. 5 an ablation study about the choice of the anomaly score function σ on the StreetHazards dataset [15]. To provide a comprehensive comparison, we considered four variants: the softmax predictions of a standard linear classifier (MSP [16]), the softmaxed predictions of a cosine classifier (*Cosine cls + softmax*), using the unnormalized class scores of a linear classifier (*Class scores*), and finally the unnormalized cosine scores of PAnS. As the figure shows, computing the softmax probabilities using a cosine classifier improves the performances w.r.t. making use of a standard classifier, improving the FPR95 value reached by the latter by 5.8% points. However, we note that using directly the class scores of the network instead of the softmax-normalized probabilities is highly beneficial, improving the standard softmaxed version by 9.5% and the cosine one by 4.7%.

Finally, we note using the unnormalized cosine scores (PAnS) outperforms the use of standard class scores achieving the highest FRP95 value up to 23.2%. We ascribe this improvement to the unbounded nature of scores of a standard classification layer, where defining a threshold values for detecting anomalies is more difficult.

Ablation study of classifiers. While our model PAnS shows promising results on anomaly segmentation, an open question is whether it still maintains the high discrimination capabilities of standard classification modules. In Table 2 we report the IoU achieved by both a standard and a cosine similarity-based classifier on each class of the StreetHazards dataset. Overall, the cosine-based classifier outperforms the standard classifier obtaining a 54.2% mIoU, which is 1% more than standard classifier (53.2% mIoU). Results show that the cosine similarity allows the model to reach higher performances on almost every class, especially on classes that are frequently considered hard, such

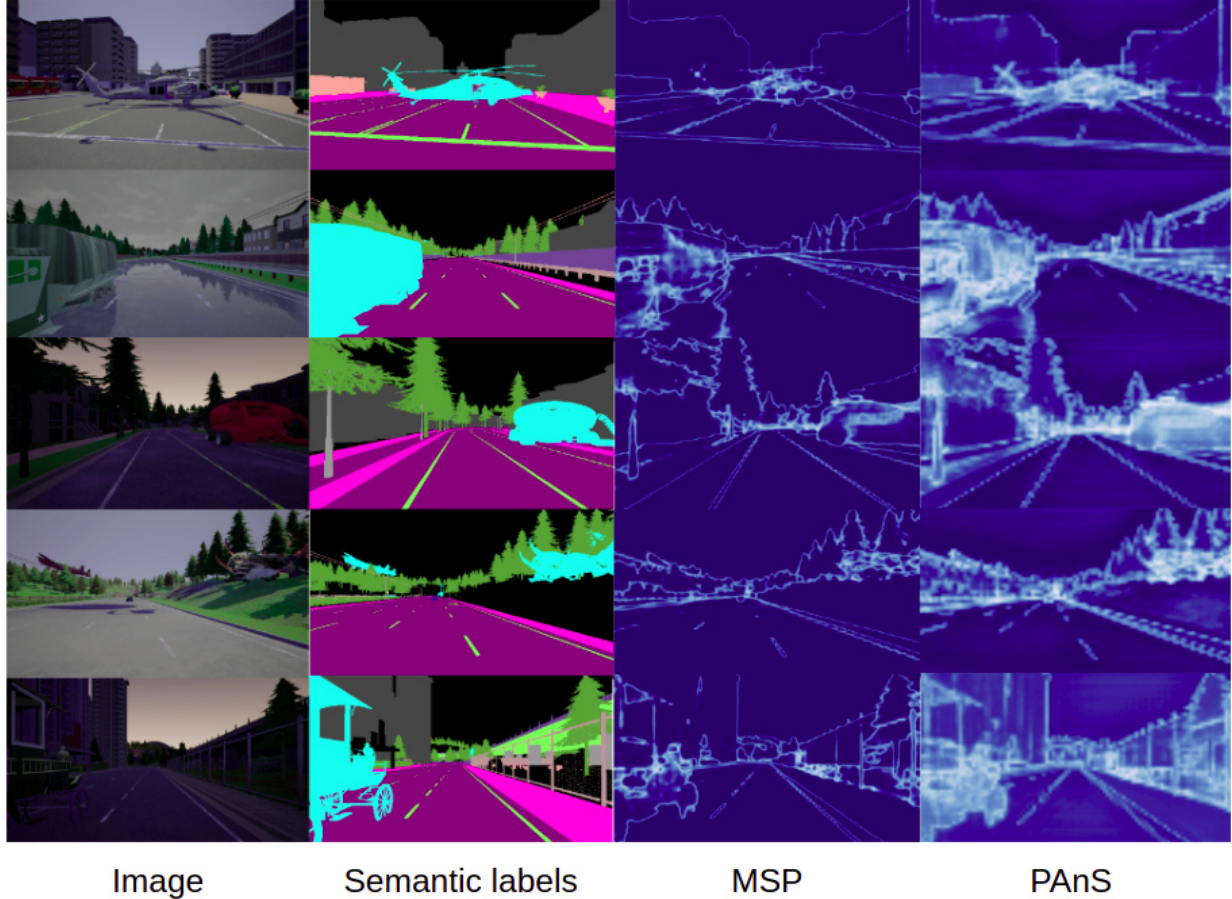| Image | Semantic labels | MSP | PAnS |

Figure 6: **Qualitative comparison** between the use of probabilities (MSP) and our direct scores (PAnS) for segmenting anomalies on StreetHazards [15]. White indicates an high score for the anomaly, while the blue indicates a low score. In the semantic labels the anomaly are represented in cyan.

Classifier	bkg	building	fence	pole	street-line	road	sidewalk	veget.	car	wall	t.sign	mIoU
Standard	84.5	70.9	30.1	**23.6**	**26.7**	92.1	57.4	75.1	53.3	42.9	28.9	53.2
Cosine-based	**84.8**	**72.1**	**30.9**	22.3	**26.7**	**92.5**	**60.0**	**75.3**	**55.2**	**45.7**	**30.3**	**54.2**

Table 2: Comparison on IoU using a standard or the cosine classifier.

as fence, traffic sign, and sidewalk. Only in the *pole* category the model's performance slightly deteriorates w.r.t. the standard classifier, 22.3% against 23.6%. The reason is that in StreetHazards the *pole* class is small and rarely represented, making it difficult for the model to estimate a good prototype for it.

5. Conclusions

In this paper we addressed the problem of anomaly detection in semantic segmentation, an important but yet scarcely researched topic. Previous works either focused on how to model the probability of a pixel belonging to an unknown class, or on generative approaches for detecting anomalies through reconstruction errors. Here, we instead argued that to detect if a pixel is anomalous or not, it is more important to have a model that measures the distance between visual features extracted from a pixel and a general representation of each class, rather than using maximum softmax probabilities. We obtain such general representations by learning class-specific prototypes as weights of a cosine similarity-based classifier. Experiments on the widely used StreetHazards benchmark support our intuition, as we achieve the new state of the art with a significant margin over previous work in two out of three metrics.

Future work will further explore this research avenue, looking into distinguishing between anomalous pixels and pixels at the boundary between two known classes, that are challenging to predict due to their uncertain nature.

References

[1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(12):2481–2495, 2017.

[2] Christoph Baur, Benedikt Wiestler, Shadi Albarqouni, and Nassir Navab. Deep autoencoding models for unsupervised anomaly segmentation in brain mr images. In *International MICCAI Brainlesion Workshop*, pages 161–169. Springer, 2018.

[3] Fabio Cermelli, Massimiliano Mancini, Samuel Rota Bulo, Elisa Ricci, and Barbara Caputo. Modeling the background for incremental learning in semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9233–9242, 2020.

[4] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(4):834–848, 2017.

[5] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. 2017.

[6] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Eur. Conf. Comput. Vis.*, 2018.

[7] Charles Corbière, Nicolas Thome, Avner Bar-Hen, Matthieu Cord, and Patrick Pérez. Addressing failure prediction by learning model confidence. In *Adv. Neural Inform. Process. Syst.*, pages 2902–2913, 2019.

[8] A Dosovitskiy, G Ros, F Codevilla, A Lopez, and V Koltun. Carla: An open urban driving simulator. arxiv 2017. *arXiv preprint arXiv:1711.03938*.

[9] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3146–3154, 2019.

[10] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. pages 1050–1059, 2016.

[11] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4367–4375, 2018.

[12] Samantha Guerriero, Barbara Caputo, and Thomas Mensink. Deepncm: Deep nearest class mean classifiers. In *ICLR-WS*, 2018.

[13] Matthias Haselmann, Dieter P Gruber, and Paul Tabatabai. Anomaly detection using deep learning based image completion. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 1237–1242. IEEE, 2018.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 770–778, 2016.

[15] Dan Hendrycks, Steven Basart, Mantas Mazeika, Mohammadreza Mostajabi, Jacob Steinhardt, and Dawn Song. A benchmark for anomaly segmentation. *arXiv preprint arXiv:1911.11132*, 2019.

[16] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.

[17] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *IEEE Conf. Comput. Vis. Pattern Recog.*, June 2019.

[18] Yen-Chang Hsu, Yilin Shen, Hongxia Jin, and Zsolt Kira. Generalized odin: Detecting out-of-distribution image without learning from out-of-distribution data. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 10951–10960, 2020.

[19] Heinrich Jiang, Been Kim, Melody Guan, and Maya Gupta. To trust or not to trust a classifier. In *Adv. Neural Inform. Process. Syst.*, pages 5541–5552, 2018.

[20] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Adv. Neural Inform. Process. Syst.*, pages 5574–5584, 2017.

[21] Shiyu Liang, Yixuan Li, and Rayadurgam Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. *arXiv preprint arXiv:1706.02690*, 2017.

[22] Shiyu Liang, Yixuan Li, and R Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *Int. Conf. Learn. Represent.*, 2018.

[23] Shiyu Liang, Yixuan Li, and R Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *International Conference on Learning Representations*, 2018.

[24] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.

[25] Krzysztof Lis, Krishna Nakka, Pascal Fua, and Mathieu Salzmann. Detecting the unexpected via image resynthesis. In *Int. Conf. Comput. Vis.*, pages 2152–2161, 2019.

[26] Zhizhe Liu, Xingxing Zhang, Zhenfeng Zhu, Shuai Zheng, Yao Zhao, and Jian Cheng. Convolutional prototype learning for zero-shot recognition. *Image and Vision Computing*, 98:103924, 2020.

[27] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2015.

[28] Chunjie Luo, Jianfeng Zhan, Xiaohe Xue, Lei Wang, Rui Ren, and Qiang Yang. Cosine normalization: Using cosine similarity instead of dot product in neural networks. In *International Conference on Artificial Neural Networks*, pages 382–391. Springer, 2018.

[29] Pascal Mettes, Elise van der Pol, and Cees GM Snoek. Hyperspherical prototype networks. 2019.

[30] Umberto Michieli and Pietro Zanuttigh. Incremental learning techniques for semantic segmentation. In *ICCV-WS*, pages 0–0, 2019.

[31] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2337–2346, 2019.

[32] Hang Qi, Matthew Brown, and David G Lowe. Low-shot learning with imprinted weights. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5822–5830, 2018.

[33] Samuel Rota Bulò, Lorenzo Porzi, and Peter Kontschieder. In-place activated batchnorm for memory-optimized training of dnns. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018.

[34] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Adv. Neural Inform. Process. Syst.*, pages 4077–4087, 2017.

[35] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, koray kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29, pages 3630–3638, 2016.

[36] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8798–8807, 2018.

[37] Yingda Xia, Yi Zhang, Fengze Liu, Wei Shen, and Alan Yuille. Synthesize then compare: Detecting failures and anomalies for semantic segmentation. *arXiv preprint arXiv:2003.08440*, 2020.

[38] Wenjia Xu, Yongqin Xian, Jiuniu Wang, Bernt Schiele, and Zeynep Akata. Attribute prototype network for zero-shot learning. 2020.

[39] Hong-Ming Yang, Xu-Yao Zhang, Fei Yin, and Cheng-Lin Liu. Robust classification with convolutional prototype learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3474–3482, 2018.

[40] Changqian Yu, Jingbo Wang, Changxin Gao, Gang Yu, Chunhua Shen, and Nong Sang. Context prior for scene segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 12416–12425, 2020.

[41] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Learning a discriminative feature network for semantic segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1857–1866, 2018.

[42] Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. *Eur. Conf. Comput. Vis.*, 2020.

[43] Yuhui Yuan and Jingdong Wang. Ocnet: Object context network for scene parsing. *arXiv preprint arXiv:1809.00916*, 2018.

[44] Hang Zhang, Kristin Dana, Jianping Shi, Zhongyue Zhang, Xiaogang Wang, Ambrish Tyagi, and Amit Agrawal. Context encoding for semantic segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 7151–7160, 2018.

[45] Zhenli Zhang, Xiangyu Zhang, Chao Peng, Xiangyang Xue, and Jian Sun. Exfuse: Enhancing feature fusion for semantic segmentation. In *Eur. Conf. Comput. Vis.*, 2018.

[46] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.