Consensus Synergizes With Memory: A Simple Approach for Anomaly Segmentation in Urban Scenes

Jiazhong Cen, Zekun Jiang, Lingxi Xie, Dongsheng Jiang, Wei Shen, Member, IEEE, and Qi Tian, Fellow, IEEE

Abstract—Anomaly segmentation is a critical task for safety-critical applications, such as autonomous driving in urban environments. Its objective is to detect out-of-distribution (OOD) samples with unseen categories, given a pre-trained segmentation model. The core challenge of this task is how to distinguish hard in-distribution samples from OOD samples, which has not been explicitly discussed in previous research. In this paper, we propose a simple yet effective approach named CosMe (Consensus Synergizes with Memory) to address this challenge. CosMe consists of two key components: 1) building a memory bank comprising seen prototypes extracted from multiple layers of the given segmentation model, and 2) training an auxiliary model that mimics the behavior of the given model and using the consensus of their mid-level features as complementary cues that synergize with the memory bank. The former serves as a baseline that can detect all potential outliers, including both OOD and hard in-distribution samples; the latter assists in distinguishing between these two types of outliers. Experimental results on several urban scene anomaly segmentation datasets demonstrate that CosMe outperforms previous approaches by a significant margin.

Index Terms—Semantic segmentation, anomaly detection, clustering.

I. INTRODUCTION

S EMANTIC segmentation is an important technology for many vision-based applications. Current studies [1], [2], [3], [4], [5], [6] on semantic segmentation mainly focused on designing complex segmentation networks with higher segmentation capacities on in-distribution samples whose categories are seen during training, while they paid less attention to out-of-distribution (OOD) samples, a.k.a., anomalous samples, whose categories are unknown during training. Consequently, they are incapable of identifying anomalous samples. Instead, they can only predict an anomalous sample as seen categories. This issue greatly impedes their uses in safety-critical applications such as autonomous driving in urban scenes. For example, the anomalous samples (marked by yellow boxes) in Fig. 1 are predicted as a road by a segmentation network, which may lead to accidents. To address this issue, anomaly segmentation, a task to detect and segment out unseen anomalous samples with a given pre-trained segmentation model, is attracting more and more attention.

Previous approaches have attempted to address the task of anomaly segmentation by relying on prediction incorrectness of given segmentation networks, such as uncertainties over categories and re-synthesis from errors caused by segmentation failures. However, these approaches lack a mechanism to distinguish hard in-distribution samples from anomalies, leading to high false positive rates. Thus, the core challenge of anomaly segmentation has become how to differentiate between the two. In this paper, we start with the following experience: vision models often produce similar response on trained objects, but their output can be uncertain on unseen objects. This also aligns with the behavior of humans.

Inspired by this intriguing observation, we propose a simple yet effective framework named Consensus Synergizes with Memory (CosMe) for anomaly segmentation, which can achieve a better trade-off between the discovery of anomalies and the classification of hard in-distribution samples. To maximize the use of information embedded in the segmentation model, we propose a strong memory-based baseline called Multi-layer Memory (MulMem). MulMem is inspired by the uncertainty-based approaches. It utilizes K-means [8] clustering to extract feature prototypes from multiple intermediate layers of the pre-trained segmentation model on in-distribution data to memorize seen samples with varying scales. To tackle the hard in-distribution problem in MulMem, we then present a consensus-based module, named Auxiliary Consensus (Aux-Con), in which an auxiliary network is trained to keep reaching a consensus with the given segmentation model in a self-supervised manner. This is achieved by explicitly maintaining hierarchical consistency between the auxiliary network and the given segmentation model on the feature representations of samples from seen categories.

Intuitively, whether an in-distribution sample is normal or hard can be determined by its distance to the prototypes in the MulMem memory bank (i.e., in-distribution samples farther away from all prototypes are regarded as hard in-distribution, and vice versa). By this means, we divide the in-distribution samples into normal and hard sets. Then we compute the
Fig. 1. The aim of anomaly segmentation is to detect samples whose categories are unseen during the training phase, which are called out-of-distribution (OOD) samples, using a pre-trained segmentation model. However, when faced with OOD samples, existing segmentation models can only predict them as seen categories, leading to dangerous outcomes in applications like autonomous driving. For instance, OOD samples (marked by yellow boxes) can be mistakenly recognized as in-distribution samples, such as road and people, and cause accidents. On the other hand, hard in-distribution samples (marked by red boxes) can also cause segmentation failures and be misidentified as OOD samples in anomaly segmentation.

Fig. 2. Statistical analysis of MulMem anomaly scores and AuxCon anomaly scores on two anomaly segmentation datasets: Fishyscapes Lost & Found [7] and Fishyscapes Static. “ID” and “OOD” denote in-distribution samples and out-of-distribution samples, respectively. Normal in-distribution samples and hard in-distribution samples are distinguished by thresholding with the MulMem anomaly score at 70% TPR. That is, when 70% out-of-distribution samples are correctly classified, the in-distribution samples whose anomaly scores given by MulMem are beyond the threshold of OOD are regarded as hard in-distribution samples. We can observe that the ability of MulMem to differentiate hard in-distribution samples from OOD samples is hardly guaranteed since the averaged MulMem anomaly score over hard in-distribution samples is even higher than that over OOD samples, while AuxCon shows a complementary ability to address this issue.

As illustrated in Fig. 2, the memory-based module can easily differentiate normal in-distribution samples from OOD samples according to MulMem (the minimum distance to prototypes) and AuxCon (the feature inconsistency), respectively.

As illustrated in Fig. 2, the memory-based module can easily differentiate normal in-distribution samples from OOD samples, but its ability to distinguish hard in-distribution samples is hardly guaranteed. In contrast, for these hard in-distribution samples, the consensus-based module shows a clearly better differentiation ability, while its overall discrimination between in-distribution samples and OOD samples is relatively smaller. These observations indicate the good complementarity between MulMem and AuxCon. And thus, a simple combination of them, i.e., CosMe, has a strong ability for anomaly segmentation, especially can favorably distinguish hard in-distribution samples from anomalies.

Experimental results show that CosMe achieves consistent and substantial improvements over the previous state-of-the-art anomaly segmentation approaches under the same setting on several urban scene datasets. In particular, on Fishyscapes benchmark [7], CosMe achieves a 41.95% AP, outperforming the previous SOTA by a significant margin of 10.9%. Moreover, the performance of CosMe is even on par with those methods using extra OOD samples for re-training.

In summary, our main contribution is to design a mechanism that can effectively distinguish hard in-distribution samples from anomalies in anomaly segmentation, which leads to a simple yet effective solution to the challenging problem of anomaly segmentation in urban scenes.

The remainder of this paper unfolds as follows: Section II delves into a review of literature pertaining to anomaly segmentation, establishing the context and precedence for our work. Following this, Section III-B details our proposed approach, CosMe, elucidating its intricate design and methodological underpinnings. Subsequently, we devote Section IV to a thorough experimental analysis, with the aim of demonstrating the effectiveness and applicability of CosMe in real-world scenarios. In Section V, we engage in an in-depth discussion concerning the hard in-distribution problem, shedding light on potential hurdles and proposed solutions. Finally, Section VI serves as the conclusion, where we encapsulate the main
findings of our study, their implications, and suggestions for future research.

II. RELATED WORK

The problem of detecting and segmenting unseen anomalous samples has been receiving increasing attention in recent years, and is closely related to the task of out-of-distribution (OOD) detection. In this section, we provide a brief overview of OOD detection, followed by a description of previous approaches for anomaly segmentation in urban scenes.

A. Out-of-Distribution (OOD) Detection

OOD detection is a broad concept that refers to the task of detecting instances that do not conform to the distribution of the training data. It encompasses various tasks such as open set recognition [9], novelty detection [10], [11], industry defect detection [12], [13], and video anomaly detection [14], [15], [16]. Since the first deep-learning-based OOD detection baseline was proposed by Hendryck and Gimpel [17] in 2017, numerous approaches have been developed, which can be categorized into uncertainty-based [18], [19], clustering-based [20], reconstruction-based [21], [22], and density-based [23], [24], among others.

B. Anomaly Segmentation

1) Uncertainty-Based Approaches: Detecting anomalies based on uncertainties is intuitive, since they are unseen during model training. Hendryck and Gimpel [17] proposed a baseline for OOD detection named “maximum softmax probability” (MSP), which measures anomaly scores by the maximum softmax probability outputted by the softmax classifier. Then they proposed “maxlogit” [25]. In maxlogit, the logits, i.e., the inputs of the softmax classifier, were used as the anomaly scores instead. Jung et al. [18] proposed “standardized maximum logit” (SML) by improving “maxlogit”. They used the statistics of the training set to standardize the “maxlogit” scores for each seen category, leading to a large improvement. To further boost the performance, some other approaches tried to first make the segmentation networks more sensitive to anomalous samples by either re-training them with a new loss function [19] or re-designing them by a new network architecture [26], [27], then applied the uncertainty measures. Specifically, Liang et al. [27] proposed GMMSeg, which replaces the final 1 × 1 convolutional classifier with Gaussian mixture models. However, the segmentation networks modified by these strategies may sacrifice their performance on seen categories and cannot be generalized to other existing segmentation networks. Chan et al. [28] and Liu et al. [29] utilized samples from the COCO dataset [30] as OOD proxy for urban scenes and introduced an extra training objective to maximize the uncertainty on these samples. However, in practice, the categories of anomalous samples are unknown and inaccessible. To address the challenge of incorporating off-the-shelf OOD data, several studies [31], [32], [33] have proposed the utilization of synthetic OOD data for training the anomaly segmentation module. However, the extent to which these methods effectively generalize to truly out-of-domain samples still raises uncertainties and requires further investigation. Moreover, we point out that, in essence, uncertainty-based approaches are a kind of clustering-based approaches, where the cluster centroids are the parameters of the last 1 × 1 convolution layer. The anomaly scores assigned by such approaches are various functions of the logit values, which in essence are the distances (i.e. vector dot-product) between features and the centroids. From this point of view, most existing uncertainty-based approaches only make use of the last memory embedded in the last convolution layer and ignore the information embedded in intermediate layers. These observations inspire us to propose the multi-layer memory baseline, which adopts the k-means clustering algorithm to memorizing the intermediate features of in-distribution samples generated by the segmentation model.

2) Synthesis-Based Approaches: Recently, thanks to the rapid development of generative adversarial networks (GANs) [34], [35], one can reconstruct complex urban scenes from semantic segmentation results. Intuitively, samples belonging to unseen categories cannot be re-synthesized. Accordingly, some recent studies [21], [22], [36] propose to re-synthesize an image from a segmentation result and then compare it to the original input image to localize the anomalous samples. However, synthesis-based approaches still have limitations in distinguishing between hard in-distribution samples and OOD samples, as segmentation results for hard samples are usually incorrect and will lead to reconstruction errors. Furthermore, their anomaly segmentation performance is heavily reliant on the training quality of GANs and may be compromised by artifacts and style shifts in the synthesized images. Moreover, the time-consuming sequential processing, which involves segmenting, synthesizing, and comparing, makes this approach impractical for real-time applications.

3) Hybrid Approaches: To achieve better anomaly segmentation results, Di Biase et al. [37] proposed a hybrid approach, which combines both uncertainty-based approach and synthesis-based approach. However, the hybrid approach inherits the drawbacks of both uncertainty-based and synthesis-based methods, such as vulnerability to hard in-distribution samples, dependence on GAN quality, and time-consuming segment-synthesize-compare operations. Even worse, their approach made use of OOD samples to train a discriminator to combine the various anomaly scores, which may limit its generalization ability to novel anomalies not represented in the training data.

III. CONSENSUS SYNERGIZES WITH MEMORY

In this section, we present the details of our proposed approach, CosMe, for anomaly segmentation. We first formulate the problem of anomaly segmentation and then provide an overview of the framework. Next, we describe the two key components of CosMe: the memory-based baseline Multi-layer Memory (MulMem) in Section III-C, and the consensus-based module Auxiliary Consensus (AuxCon) in Section III-D.

A. Problem Setup

We first define the problem of anomaly segmentation. Given a semantic segmentation model $M$ with parameters $\Theta$ that
Fig. 3. Overall framework of CosMe. It consists of two main modules: the Multi-layer Memory (MulMem) and the Auxiliary Consensus (AuxCon). MulMem stores feature prototypes extracted from multiple intermediate layers of the pre-trained segmentation model, while AuxCon introduces an auxiliary model that aims to maintain consistency with the segmentation model on in-distribution data. Specifically, MulMem assigns an anomaly score to each feature vector based on its minimum distance to the prototypes, while AuxCon calculates an anomaly score based on the inconsistency between the segmentation model and the auxiliary model. These two types of anomaly scores are combined to generate the final anomaly prediction.

has been trained on a training set $T = (X^{(n)}, Y^{(n)})_{n=1}^{N}$ consisting of images and their corresponding ground-truth segmentation masks, the goal is to segment out pixels in a testing image $X$ that belong to an unseen category set $U$ ($U \cap C = \emptyset$), based on fixed $M$ and $T$. This is achieved by assigning an anomaly score $\varepsilon_{i,j}(X)$ to each pixel $(i,j)$, where $\varepsilon_{i,j}(X)$ is small when $y_{i,j} \in C$, but large when $y_{i,j} \in U$.

Some of the state-of-the-art anomaly segmentation approaches [19], [27], [31] resort to re-designing or re-training the segmentation model, which can have a detrimental impact on the performance of in-distribution segmentation and restrict the generalizability of the model. Other approaches [32], [33] employ out-of-distribution (OOD) data, either synthetic or sourced from other datasets, for training an anomaly segmentation head. However, the categories of OOD data are often unknown and unpredictable in real-world applications, significantly reducing the reliability and practicality of these methods in real-world scenarios.

In contrast, we follow a strict setting in which the segmentation model is frozen and no information about the anomalies is available.

B. Overall Framework

The overall framework of CosMe, depicted in Fig. 3, comprises two main components: Multi-layer Memory (MulMem) and Auxiliary Consensus (AuxCon). Along with the pre-trained segmentation model, these modules work together to address the challenge of distinguishing hard in-distribution samples from OOD samples while leveraging the memories embedded in the segmentation model. In the following sections, we provide a brief overview of MulMem and AuxCon.

- The Multi-layer Memory (MulMem) consists of a memory bank that comprises several sub-branches. Each sub-branch stores representative features, i.e., prototypes, that are outputted from a specific layer of the given segmentation network. To determine whether a sample is anomalous or not, the minimum feature distance between the sample and all prototypes in the sub-branches is computed.

Fig. 4. Many decision boundaries in the high-dimensional feature space can perform consistently on specific data. But for anomalies, they may show high inconsistency.

- The Auxiliary Consensus (AuxCon) is an auxiliary model that shares information with the pre-trained segmentation model. Its goal is to imitate the behavior of the pre-trained model on in-distribution data. Compared to completely unseen anomalies, hard in-distribution samples exhibit more consistent features with normal in-distribution samples. These shared similarities facilitate the auxiliary model in easily learning to imitate the behavior of the segmentation model on hard in-distribution data. Conversely, when presented with anomalies, the auxiliary model demonstrates significant inconsistencies with the pre-trained segmentation model. In other words, although the auxiliary model is trained to mimic the segmentation model, the differences in network architecture and learning objectives result in distinct decision boundaries between the two models, as depicted in Fig. 4. Consequently, the mimicking error of the auxiliary model naturally serves as a kind of anomaly score.

Empirically, vision models often produce similar response on trained objects, but their output can be uncertain on unknown objects. This also aligns with the behavior of humans. This intriguing property serves as the motivation behind the design of MulMem and AuxCon.
Algorithm 1 Memory Learning for Sub-Branch $S^{(l)}$

Require: Training batch $B$, coefficient $m$, initialized $S^{(l)}$  
Ensure: Updated $K^{(l)}$ prototypes $S^{(l)} = \{p_k\}_{k=1}^{K^{(l)}}$

1: $F_B \leftarrow \{f^{(l)}_s(X; \Theta)\mid X \in B\}$  
2: for $f \in F_B$ do  
3: \quad $Q_{S^{(l)}}, f \leftarrow \{\phi(p, f)\mid \forall p \in S^{(l)}\}$  
4: end for  
5: for $p \in S^{(l)}$ do  
6: \quad $S_p^{(l)} \leftarrow \{f \mid f \in F_B, p = \arg \max_{p \in S^{(l)}} Q_{S^{(l)}, f}\}$  
7: \quad $p \leftarrow m \cdot p + (1 - m) \cdot \frac{1}{|S_p^{(l)}|} \sum_{f \in S_p^{(l)}} s^{(l)} f_{p}$  
8: end for

C. Multi-Layer Memory

Let $F^{(l)}(X; \Theta) \in \mathbb{R}^{H^{(l)} \times W^{(l)} \times C^{(l)}}$ be the feature map for an input image $X$, outputted from layer $l$ of the given segmentation model $\mathbb{M}$. $H^{(l)}, W^{(l)}, C^{(l)}$ denote the height, width, and channel number of the feature map respectively. Let $f^{(l)}_s(X; \Theta)$ denote the feature vector at the location $(i, j)$ of this feature map. Our goal is to build a memory bank $\mathbb{M} = \{S^{(l)}_i \mid i \in \mathcal{L}\}$ to store prototype features of seen samples from the training set $T$, where $S^{(l)}_i$ is a sub-branch of the bank to store prototype features outputted from layer $l$, and $\mathcal{L}$ is the set of layer of interests.

To memorize seen samples, a straightforward way is performing clustering on the training set to generate prototypes. Since the segmentation network processes data samples batch-wise, we adopt mini-batch K-means [8] algorithm to generate the prototypes.

Without loss of generality, we describe this prototype extraction process by taking one sub-branch $S^{(l)}_i$ as an example. Let $K^{(l)}_i$ denote the size of $S^{(l)}_i$, we randomly initialize each element of the $K^{(l)}$ prototypes from a uniform distribution $U(0, 1)$.

After initialization, we learn to update the prototypes in $S^{(l)}_i$ by the mini-batch K-means algorithm, which is driven by momentum update. Let $m$ be the pre-defined momentum coefficient for training $S^{(l)}_i$. Specifically, for each prototype $p$ in $S^{(l)}_i$, we update $p$ by the features that are closest to $p$, i.e., the features have the highest cosine similarity with $p$. This can be achieved by maintaining a set $S_p^{(l)}$ to store such features for prototype $p$:

$$S_p^{(l)} \leftarrow \{f \mid f \in F_B, p = \arg \max_{p \in S^{(l)}} Q_{S^{(l)}, f}\},$$

where $\phi(\cdot, \cdot)$ denotes the cosine similarity between two vectors, $F_B \leftarrow \{f^{(l)}_s(X; \Theta)\mid X \in B\}$ is the set containing all the features of the images in a mini-batch $B$ and $Q_{S^{(l)}, f} \leftarrow \{\phi(p, f)\mid \forall p \in S^{(l)}\}$ is the set containing all cosine similarities between each prototype $p \in S^{(l)}$ and a feature $f$. Finally, $p$ is computed by a momentum update:

$$p \leftarrow m \cdot p + (1 - m) \cdot \frac{1}{|S_p^{(l)}|} \sum_{f \in S_p^{(l)}} s^{(l)} f_{p}.$$  

The detail of this memory learning algorithm is shown in Algorithm 1.

With the learned sub-branch $S^{(l)}$, given an input image $X$, an anomaly score map $\Gamma^{(l)}(X)$ for this input image is computed by

$$\gamma^{(l)}_{i,j}(X) = 1 - \max_{p \in S^{(l)}} \phi(p, f^{(l)}_i(X; \Theta))(\forall p \in S^{(l)}),$$

where $\gamma^{(l)}_{i,j}(X)$ denotes the anomaly score at the location $(i, j)$ of the anomaly score map $\Gamma^{(l)}(X)$.

Note that in our experiments the coefficient $m$ is set to 0.9 empirically. In practice, $m$ can also be determined adaptively according to the training iteration $i$ and the training batch size $|B|$. That is:

$$m_i = 1 - \frac{1}{(i + 1) \cdot |B|}.$$  

Since several sub-branches form the memory bank $\mathbb{M} = \{S^{(l)}_i \mid i \in \mathcal{L}\}$ of MulMem, we compute the anomaly score map $\tilde{\Gamma}(X)$ given by the memory bank $\mathbb{M}$ by a simple combination of the anomaly score maps given by each sub-branch:

$$\tilde{\gamma}_{i,j}(X) = \prod_{l \in \mathcal{L}} \gamma^{(l)}_{i,j}(X),$$

where $\tilde{\gamma}_{i,j}(X)$ is the MulMem anomaly score at the location $(i, j)$ of the anomaly score map $\tilde{\Gamma}(X)$. We additionally adopt the standardization strategy in [18] to normalize the MulMem anomaly scores in $\tilde{\Gamma}(X)$.

D. Auxiliary Consensus

As shown in Fig. 3, the auxiliary consensus module explicitly ensures feature consistency between the given segmentation model $\mathbb{M}$ and an auxiliary model $\mathbb{M}'$ (parameterized by $\Theta'$). This is achieved by self-supervised learning without using any segmentation annotations of the training set $T$. For this purpose, $\mathbb{M}'$ has the same down-sampling schedule as $\mathbb{M}$, so that for a layer $l$ in $\mathbb{M}$ we can find its corresponding layer $l'$ in $\mathbb{M}'$. For example, ResNet50 [38] can be the backbone of an auxiliary model for a segmentation model with ResNet101.

Let $L_s$ be a set of layers of $\mathbb{M}$, e.g., for ResNet, $L_s$ can be the last Conv layers of the five Conv blocks $L_s = \{C1, C2, C3, C4, C5\}$ which is used to supervise the corresponding layers of $\mathbb{M}'$. For each layer $l \in L_s$, $l'$ is its corresponding layer in $\mathbb{M}'$. Our learning purpose is to enforce the feature map $G^{(l)}(X; \Theta')$ outputted by layer $l'$ of $\mathbb{M}'$ to approach $F^{(l)}(X; \Theta)$.

Towards this end, we fix $\Theta$, and minimize the following loss function on the training set $T$:

$$L = \sum_{X \in T} \sum_{l \in L_s} \frac{1}{H^{(l)} \times W^{(l)}} \|F^{(l)}(X, \Theta) - G^{(l)}(X; \Theta')\|_F^2,$$

where $\|\cdot\|_F$ means the Frobenius norm of a matrix. The overall training algorithm is shown by Algorithm 2.

During inference, to compute anomaly scores, we select a subset $L_s'$ from $L_s$ as an evaluation set. Given a testing image $X$, the anomaly score $\psi_{i,j}^{(l)}$ at each location $(i, j)$ is computed by:

$$\psi_{i,j}^{(l)} = \frac{1}{C^{(l)}} \|F^{(l)}_{i,j}(X; \Theta) - g^{(l)}_{i,j}(X; \Theta')\|_2^2,$$
we also introduce an exponential addition strategy, where

calculated by

have a larger magnitude. Multiplication can eliminate this
score basically depend on AuxCon, since its anomaly scores
have different magnitude. Addition makes the final anomaly
combination strategy for AuxCon and MulMem to generate
in which

and multiplication. In practice, we find the two indicators
we consider two simple and intuitive strategies – addition
and multiplication. Consequently, we use an auxiliary model to address this
task rather than segmentation, thereby assisting the MulMem
module in detecting anomalies.

However, it is also important to note that even with the
incorporation of the auxiliary model, the limitations of the
information bottleneck still persist. This is because the training
objective of the auxiliary model remains unrelated to anomaly
segmentation. Nonetheless, our experiments demonstrate that
an auxiliary model specifically designed to mimic the segment-
tion model can serve as a valuable complement for anomaly
segmentation. We hope that future studies will delve deeper
into this problem to develop more comprehensive solutions.

IV. EXPERIMENTS

In this section, we present the datasets utilized in our experiments, as well as implementation details, evaluation
metrics, and the experimental results.

A. Datasets

We conduct our experiments on four widely-used
anomaly segmentation datasets [40]: Fishyscapes Lost &
Found [7], Fishyscapes Static [7], Road Anomaly [21] and
Streethazards [25].

1) Fishyscapes Lost & Found: Fishyscapes (FS) Lost &
Found [7] is a high-quality image dataset that contains realistic
road obstacles. It is built upon the original Lost & Found [41]
dataset and follows the same setting as Cityscapes [42], which
is a widely used dataset for urban-scene segmentation. It con-
sists of real urban images with 37 kinds of unexpected road
obstacles and 13 different street scenarios (e.g., varying road
surface appearances, strong illumination changes, etc). FS Lost &
Found provides a public validation set of 100 images and a
hidden test set of 275 images for benchmarking purposes.

2) Fishyscapes Static: Fishyscapes (FS) Static is based on
the validation set of Cityscapes [42]. It was constructed by
superimposing anomalous samples collected from PASCAL
VOC [43] onto Cityscapes, ensuring seamless style matching
with the Cityscapes dataset. The dataset consists of a publicly
available validation set with 30 images and a hidden test set
with 1,000 images for benchmarking purposes.

3) Road Anomaly: The Road Anomaly dataset [21] captures
hazardous scenarios faced by vehicles on roads. It consists of
60 images collected from the internet, depicting unusual
objects such as animals, rocks, etc. on roads, with a resolution
of 1280 \times 720. Due to the fact that this dataset was not
collected under conditions similar to Cityscapes, there exists a
significant domain gap between the two datasets. This
dataset can be utilized to evaluate the domain generalization
capabilities of an anomaly segmentation method.

F. Relation to Information Bottleneck Theory

The information bottleneck theory [39] aims to conceptual-
ize deep neural networks as bottleneck models. Deep neural
networks convert input variables into latent variables and
output predictions based on these latent variables. According
to this theory, the objective of training neural networks is
to maximize the mutual information between the latent vari-
ables and the predictions, while simultaneously constraining
the mutual information between the input variables and the
predictions. According to this theory, deep neural networks
effectively acts as an information bottleneck, filtering out
irrelevant information in line with the training target.

In semantic segmentation, the training objective for the
neural network is to generate accurate predictions for all inputs
within the target distribution. This implies that during the for-
ward process of the network, information related to anomaly
segmentation is filtered out. Memory-based approaches in this
context involve comparing intermediate dense feature vectors
with extracted feature prototypes to determine whether a given
feature belongs to anomalies. However, due to the filtering
effect, it becomes challenging to distinguish between hard
in-distribution data and out-of-distribution data.

As a result, we use an auxiliary model to address this
challenge by emulating the behaviors of the pre-trained
segmentation model. This auxiliary model, called AuxCon,
is equipped with additional parameters that focus on a different
task rather than segmentation, thereby assisting the MulMem
module in detecting anomalies.

However, it is also important to note that even with the
incorporation of the auxiliary model, the limitations of the
information bottleneck still persist. This is because the training
objective of the auxiliary model remains unrelated to anomaly
segmentation. Nonetheless, our experiments demonstrate that
an auxiliary model specifically designed to mimic the segment-
tion model can serve as a valuable complement for anomaly
segmentation. We hope that future studies will delve deeper
into this problem to develop more comprehensive solutions.
TABLE I

| Method | Utilizing OOD Data | Requiring Re-training | FS Lost & Found | FS Static |
|--------|--------------------|-----------------------|-----------------|-----------|
| Synboost [9] | ✓ | × | 15.79 | 18.75 |
| Density - Logistic Regression [2] | ✓ | ✓ | 24.36 | 13.39 |
| Bayesian DeepLab [36] | × | ✓ | 38.46 | 15.50 |
| OoD Training - Void Class [2] | ✓ | ✓ | 22.11 | 19.40 |
| Discriminative Outlier Detection Head [1] | ✓ | ✓ | 19.02 | 0.29 |
| Dirichlet Deeplab [32] | ✓ | ✓ | 47.43 | 84.60 |
| GMMSeg [26] | × | ✓ | 6.61 | 10.49 |
| DenseHybird [16] | ✓ | ✓ | 6.18 | 5.51 |
| SLEEG [44] | ✓ | × | 6.69 | 5.5 |
| PEBAL [42] | ✓ | ✓ | 7.58 | 1.73 |
| CosMe (Ours) | × | × | 13.32 | 5.74 |

4) Streethazards: The Streethazards dataset (cited in [25]) is an anomaly segmentation dataset generated using the Unreal Engine and the CARLA simulation environment. It includes 5,125 image and semantic segmentation ground-truth pairs for training the segmentation model, 1,031 non-anomalous pairs for validation, and 1,500 test pairs with anomalies. There are a total of 250 unique anomaly models of various types, and 12 classes of objects utilized for training.

B. Implementation Details

1) Pre-Trained Segmentation Model: For fair comparison, we follow [19], [22], and [25] to adopt PSPnet [44] with ResNet101 [38] as the given pre-trained segmentation model on Streethazards and follow [18] and [37] to adopt DeepLabV3+ [45] with ResNet101 as the given pre-trained segmentation model on the other three datasets.

2) Multi-Layer Memory: We maintain three memory sub-branches, in which prototypes are extracted from the outputs of C4 layer and C5 layer of the ResNet101 backbone as well as the last hidden layer (LH) of the segmentation model. Their sizes are set to 1024, 2048 and 256 (equal to the feature channel number), respectively. The momentum coefficient $m$ is set to 0.9.

3) Auxiliary Consensus: The auxiliary models adopt the same down-sampling schedule as their corresponding given segmentation models but with different backbones, i.e., (ResNet101 $\rightarrow$ ResNet50 [38]). For all experiments, the supervision layer set is $L_s = \{C2, C3, C4, C5, LH, O\}$, where O is the output layer of the segmentation model, i.e., the last $1 \times 1$ Conv layer to compute the outputted logits over categories. The evaluation layer set for computing AuxCon anomaly score is $L_e = \{C5\}$. We train the auxiliary model for 100 epochs. The learning rate is 0.01, batch size is 8. Our training is done on 4 Nvidia GeForce RTX 3090 GPUs. Note that we take no data augmentation to make sure the auxiliary model can reach consensus with the segmentation model only on in-distribution samples.

4) Evaluation Metrics: Following [18], [22], and [25], three metrics are used for evaluation: area under receiver operating curve (AUROC), false positive rate at 95% true positive rate (FPR95) and average precision (AP). Since anomalous samples are much less than in-distribution samples, the data imbalance suggests FPR95 and AP are major evaluation metrics.

C. Comparison With Previous Approaches

1) Fishyscapes Test Sets: We first compare CosMe with other anomaly segmentation approaches on Fishyscapes (FS Lost & Found and FS Static) test sets. Note that, Fishyscapes test sets are unavailable to the public to prevent approaches from overfitting to specific “anomalies”. We submit our code to the dataset holder and get the results from Fishyscapes Leaderboard. According to Table I, we achieve a new SOTA performance compared with approaches without re-training or extra OOD data and outperform them by large margins.

1https://fishyscapes.com
Table II: Comparison on Fishyscapes Validation Sets and Road Anomaly. ¶ and ♯ indicate re-training and extra OOD data, respectively.

| Method         | FS Lost & Found | FS Static | Road Anomaly |
|----------------|-----------------|-----------|--------------|
|                | FPR5  | AUROC   | AP  | FPR5  | AUROC   | AP  | FPR5  | AUROC   | AP  |
| LDN_BIN [1]    | 23.97 | 95.59   | 45.71 | -     | -      | -   | -     | -      | -   |
| GMMSeg [26]    | 13.11 | 97.34   | 43.47 | -     | -      | -   | 34.42 | 47.90  |
| DenseHybrid [16] | 6.0    | -       | 60.5  | 4.2   | -      | 63.1 | 43.2  | 63.9 | |
| SLEEG [44]     | 10.9  | 98.3    | 70.9  | 3.85  | 99.22  | 77.23 | 39.40 | 89.00  | 52.70 |
| PEBAL [42]     | 6.49  | 99.09   | 59.83 | 6.81  | 99.23  | 82.73 | 28.29 | 92.51  | 62.37 |
|                |        |         |       |       |        |      |       |        |     |
| MSP [21]       | 45.63 | 86.99   | 6.02  | 34.10 | 89.94  | 14.24 | 68.44 | 73.76  | 20.59 |
| MaxLogit [20]  | 38.13 | 92.00   | 18.77 | 28.50 | 92.80  | 27.99 | 64.85 | 77.97  | 24.44 |
| SynthCP [46]   | 45.95 | 88.34   | 6.54  | 34.02 | 89.90  | 23.22 | 64.69 | 76.08  | 24.87 |
| SML [23]       | 14.53 | 96.88   | 36.55 | 16.75 | 96.69  | 48.67 | 49.74 | 81.96  | 25.82 |

| MulMem (Ours)  | 14.47 | 97.39   | 41.73 | 5.07  | 98.87  | 65.61 | 63.38 | 80.02  | 29.49 |
| AuxCon (Ours)  | 18.68 | 95.79   | 19.52 | 5.45  | 98.76  | 62.01 | 51.33 | 84.61  | 38.40 |
| CosMe (Ours)   | 9.43  | 98.35   | 52.37 | 1.72  | 99.49  | 73.40 | 51.04 | 85.92  | 41.11 |

Table III: Comparison Results on Streethazards.

| Method         | Utilizing OOD Data | Re-training | FPR5  | AUROC   | AP  |
|----------------|-------------------|-------------|-------|---------|-----|
| Dropout [13]   | ✓                 | ✓           | 79.4  | 69.9    | 7.5 |
| DML [5]        | ✓                 | ✓           | 17.3  | 93.7    | 14.7 |
| LDN_BIN [1]    | ✓                 | ✓           | 30.9  | 89.7    | 18.8 |
| DenseHybrid [16] | ✓                | ✓           | 13.0  | 95.6    | 30.2 |
| MSP [21]       | X                 | X           | 33.7  | 87.7    | 6.6 |
| SynthCP [46]   | X                 | X           | 28.4  | 88.5    | 9.3 |
| MaxLogit [20]  | ✓                 | ✓           | 26.5  | 89.3    | 10.6 |
| CosMe (PSPnet) | X                 | X           | 23.2  | 91.3    | 16.8 |
| CosMe (DeepLabV3+) | X         | X           | 15.5  | 94.6    | 19.7 |

5) Comparison on Inference Time: When anomaly segmentation approaches are evaluated, their inference time should also be taken into consideration, since real applications, such as autonomous driving, require real-time performance. Uncertainty-based approaches (e.g., SML [18]) are the fastest, since they need almost no additional computation. CosMe also enjoys a faster inference speed, which is close to uncertainty-based approaches. This is because that, once training is finished, there is no dependency between the pre-trained segmentation model and the auxiliary model in CosMe. Compare with synthesis-based approaches (e.g., SynthCP [22]), CosMe can be parallelized. To process a 2048 × 1024 image on a NVIDIA GeForce RTX 3090 GPU, the original segmentation network needs 60.54 ms. SML, serial CosMe, parallel CosMe, and SynthCP need extra 14.48 ms, 70.24 ms, 18.82 ms and 86.36 ms, respectively.

D. Qualitative Analysis

Fig. 5 visualizes some anomaly segmentation results at TPR95 (white pixels indicate anomalies). From the comparison between the second row (MulMem) and the third row (CosMe), we can see that CosMe significantly suppresses responses caused by hard in-distribution samples. We also provide visualization comparison with SLEEG [32], which requires synthetic OOD data to train. As shown in Fig. 6, our CosMe show comparable performance with SLEEG.
E. Ablation Study

In Table II, we presented the ablation results of the two sub-modules of CosMe. In this section, we further investigate the impact of different design choices within each sub-module on the FS Lost & Found validation set.

1) Ablation on Layer Set $\mathcal{L}$ for MulMem Sub-Branches: In Section III-C, we introduce $\mathcal{L}$ as the set of layers used for prototype learning in the memory bank. The ablation results, presented in the upper part of Table IV, demonstrate the impact of this choice solely on MulMem, without the assistance of AuxCon. Notably, the best performance is achieved when $\mathcal{L} = \{C4, C5, LH\}$, indicating that there are valuable memories embedded in different stages of the segmentation network that have not been fully leveraged by previous approaches.

It is widely recognized that shallow layers of convolutional networks capture low-level visual features shared among different objects, making them less effective for distinguishing anomalies. In our approach, we concentrate on the information embedded in deeper layers, which contain higher-level information and better discriminate anomalies. This insight can inspire future research in anomaly segmentation.

2) Ablation on AuxCon Evaluation Layer Set $\mathcal{L}_{e}$: By fixing $\mathcal{L} = \{C4, C5, LH\}$, we then conduct ablation on evaluation layer set $\mathcal{L}_{e}$. As shown in the lower part of Table IV, when $\mathcal{L}_{e} = \{C5\}$, CosMe reaches its best performance. The choice of $\mathcal{L}_{e}$ depends on the difficulty of the auxiliary model to imitate the segmentation model. Intuitively, deeper network layers extract more complex and high-level features, which are harder to imitate, since the error of imitation accumulates as information propagates through the network layers. As a result, anomaly scores derived from features extracted from deeper layers, such as LH, tend to be elevated for both in-distribution and out-of-distribution samples, making them unsuitable for anomaly segmentation. Similarly, anomaly scores calculated by C4 tend to be low, making $\mathcal{L}_{e} = C5$ the optimal choice. Please note that the suitability of this choice may vary depending on the specific segmentation network architectures used.

3) Ablation on MulMem Sub-Branch Size $K^{(l)}$: Note that the ablations of $K^{(l)}$ and $m$ are also done solely on MulMem. Without loss of generality, we focus on the effect of $K^{(C4)}$ with other parameters fixed. They are: $\mathcal{L} = \{C4, C5, LH\}$, $K^{(C5)} = 2048$, $K^{(LH)} = 256$ and $m = 0.9$. Table V show the results. We find the sub-branch $S^{(C4)}$ with larger $K^{(C4)}$ shows better performance, similar results are observed in other sub-branches. In general, MulMem is robust to the choice of $K^{(l)}$ when $K^{(l)}$ is large enough. It is worth noting that the inherent bias of K-means towards spherical or circular clusters limits its performance on clusters with non-standard shapes, particularly elongated or irregular shapes. Since the primary
Table IV
Ablation Results for the Selection of $L$ (Upper Part) and $L_e$ (Lower Part)

| Layers in Set | FPR95 ↓ | AUROC ↑ | AP ↑ |
|--------------|---------|---------|------|
| $\{C_4\}$   | 34.55   | 93.17   | 9.81 |
| $\{C_5\}$   | 36.49   | 92.97   | 22.11|
| $\{L_H\}$   | 20.84   | 95.74   | 12.32|
| $\{C_4, C_5\}$ | 23.94   | 95.34   | 27.38|
| $\{C_4, C_5, L_H\}$ | 14.47 | 97.39   | 41.73|
| $\{C_4\}$   | 10.58   | 98.13   | 44.99|
| $\{C_5\}$   | 9.43    | 98.35   | 52.37|
| $\{L_H\}$   | 15.45   | 97.04   | 35.91|
| $\{O\}$     | 16.90   | 96.50   | 27.95|
| $\{C_4, C_5\}$ | 10.39   | 98.24   | 49.96|
| $\{C_4, C_5, L_H\}$ | 13.09   | 97.67   | 42.74|
| $\{C_4, C_5, L_H, O\}$ | 15.07   | 97.00   | 32.28|

Table V
Ablation Results for $K^{(C_4)}$ (Upper Part) and $m$ (Lower Part)

| $K^{(C_4)}$ | FPR95 ↓ | AUROC ↑ | AP ↑ |
|-------------|---------|---------|------|
| 1024        | 14.47   | 97.39   | 41.73|
| 768         | 16.38   | 97.21   | 41.14|
| 512         | 13.95   | 97.51   | 40.34|
| 256         | 13.62   | 97.48   | 41.34|
| 128         | 12.66   | 97.68   | 40.71|
| 64          | 18.01   | 96.49   | 28.46|

| $m$ | FPR95 ↓ | AUROC ↑ | AP ↑ |
|-----|---------|---------|------|
| 0.99| 15.78   | 97.21   | 39.73|
| 0.9 | 14.47   | 97.39   | 41.73|
| 0.8 | 14.67   | 97.22   | 40.74|
| 0.7 | 14.74   | 97.23   | 37.58|
| 0   | 15.23   | 97.23   | 36.77|

Table VI
Ablation Results for Different Backbones of the AuxCon

| Backbone Arch | FPR95 ↓ | AUROC ↑ | AP ↑ |
|---------------|---------|---------|------|
| ResNet11      | 12.54   | 97.88   | 45.68|
| ResNet50      | 9.43    | 98.35   | 52.37|
| ResNet101     | 12.98   | 97.69   | 43.63|
| ResNet152     | 13.73   | 97.75   | 46.97|

Table VII
Ablation Results for Different Combination Strategies for AuxCon and MulMem

| Strategy         | FPR95 ↓ | AUROC ↑ | AP ↑ |
|------------------|---------|---------|------|
| Addition         | 10.46   | 98.07   | 45.62|
| Multiplication   | 9.43    | 98.35   | 52.37|
| E-addition (a=0.5; b=0.5) | 9.83 | 98.22   | 49.10|
| E-addition (a=0.3; b=0.3) | 9.59 | 98.28   | 50.74|

The goal of MulMem is to memorize dense features rather than categorize them into distinct clusters, we address this limitation by using a larger value of $K$ to enhance its memorization capability. By increasing the number of clusters, MulMem has a better chance of capturing the diverse variations in the data, thus overcoming the limitations associated with spherical or circular clusters. This is evident from the experimental results presented in Table V, where it can be observed that the performance rapidly declines when $K^{(C_4)} = 64$.

In practice, we adopt an intuitively superior setup that let $K^{(l)} = C^{(l)}$ (recall that $C^{(l)}$ denotes the channel number of the intermediate outputted by layer $l$).

4) Ablation on MulMem Momentum Coefficient $m$: According to Table V, within a reasonable range, MulMem has better performance with a relatively larger $m$. Equation 2 indicates prototypes generated with larger $m$ focus more on the whole dataset. Conversely, prototypes generated with smaller $m$ focus more on the last batch of images, which leads to unstable performance.

5) Ablation on Different AuxCon Backbones: Table VI shows the effect of different backbones of AuxCon.

"ResNet11" is a small network with the same down-sampling schedule as ResNet101, which has only 11 layers. According to Table VI, AuxCon is relatively robust to the choice of different auxiliary models. The anomaly segmentation performance is not improved by employing a larger auxiliary model. This can be attributed to the fact that our intention is for the auxiliary model to specifically learn to mimic the behavior of the segmentation model on in-distribution data. Increasing the network capacity of the auxiliary model may inadvertently encourage it to learn to mimic potential out-of-distribution data, ultimately leading to a degradation in the anomaly segmentation performance. On the contrary, if the auxiliary network is too small, it may struggle to adequately learn the behavior of the segmentation model on in-distribution data, resulting in poor performance. Therefore, striking the right balance in terms of auxiliary model size is crucial for achieving optimal anomaly segmentation performance.

6) Ablation on Different Combination Strategies: We consider three strategies to combine the AuxCon anomaly score with MulMem anomaly score: 1) addition; 2) multiplication; 3) exponential addition (E-addition). After adopting exponent to adjust the magnitude of the two items, addition strategy can reach similar performance with multiplication. However, it needs to choose two additional hyper-parameters. Consequently, simple multiplication is effective and economical.

7) AuxCon Synergizes With SML: We argue that uncertainty-based approaches can be regarded as special cases of MulMem, where the only memory sub-branch is the last $1 \times 1$ convolution kernel. Such assumption suggests that the MulMem can be replaced with another uncertainty-based approach. Take the previous SOTA SML [18] as an example, the results are shown in Table VIII. With the help of AuxCon, the performance of SML is further improved. The results demonstrate the good generalization ability of AuxCon and its potential to synergize with other uncertainty-based approaches.
TABLE VIII
THE RESULTS OF SML WITH AUXCON ON FISHYSCAPES VALIDATION SETS

| Method    | FS Lost & Found PPR95 ↓ AUROC ↑ AP ↑ | FS Static PPR95 ↓ AUROC ↑ AP ↑ |
|-----------|-------------------------------------|----------------------------------|
| SML       | 21.44 96.30 36.97 13.47 96.59 49.11 |                                  |
| SML + AuxCon | 9.40 98.16 45.49 3.06 99.39 80.22 |                                  |
| CosMe (Ours) | 9.43 98.35 52.37 1.47 99.58 79.25 |                                  |

V. DISCUSSION

We attempt to address the challenging hard in-distribution problem in anomaly segmentation with the assistance of an auxiliary model, which emulates the behavior of the original segmentation model on in-distribution data. Although this approach achieves impressive results on existing anomaly segmentation benchmarks, there is still a trade-off in differentiating hard in-distribution samples from out-of-distribution samples. The hard in-distribution problem can be attributed to various factors, such as a lack of training samples. When faced with insufficient training data, AuxCon is powerless as the auxiliary model also requires a significant amount of data for adequate training. This highlights the issue of detecting hard in-distribution samples lies in the training of the segmentation model. Moreover, as discussed in Section III-F, the impact of the information bottleneck is known to limit the performance of anomaly segmentation. One potential solution to address this issue is utilizing more advanced segmentation architectures that can effectively reduce the loss of information related to anomaly segmentation. We hope that future studies will delve into these challenges and propose innovative solutions.

VI. CONCLUSION

In this paper, we pointed out the core challenge of anomaly segmentation is the existence of hard in-distribution samples. Inspired by the psychology finding of consensus processes in group recognition memory performance, we proposed “Consensus Synergizes with Memory” (CosMe), which utilizes inconsistency with an auxiliary model to complement the memory-based prototype-level distance for anomaly segmentation. Our approach was verified on various datasets and achieved superior results on all of them. Note that, our approach has no constraint on the segmentation network and can be parallelized. This merit shows its potential in practical applications.

REFERENCES

[1] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2015, pp. 3431–3440.

[2] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional networks for biomedical image segmentation,” in Medical Image Computing and Computer-Assisted Intervention—MICCAI, Cham, Switzerland: Springer, 2015.

[3] M. Orsić and S. Šegvić, “Efficient semantic segmentation with pyramidal fusion,” Pattern Recognit., vol. 110, Feb. 2021, Art. no. 107611.

[4] X. Li et al., “FoveaNet: Perspective-aware urban scene parsing,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 784–792.

[5] S. Choi, J. T. Kim, and J. Choo, “Cars can’t fly up in the sky: Improving urban-scene segmentation via height-driven attention networks,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 9370–9380.

[6] J. Fu et al., “Dual attention network for scene segmentation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 3141–3149.

[7] H. Blum, P.-E. Sarlin, J. Nieto, R. Siegwart, and C. Cadena, “The fishyscapes benchmark: Measuring blind spots in semantic segmentation,” Int. J. Comput. Vis., vol. 129, pp. 3119–3135, Nov. 2021.

[8] J. MacQueen, “Classification and analysis of multivariate observations,” in Proc. 5th Berkeley Symp. Math. Statist. Probab., I, Berkeley, CA: University of California Press, 1967, pp. 281–297.

[9] W. I. Scheiner, A. de Rezende Rocha, A. Sapkota, and T. E. Boult, “Toward open set recognition,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 7, pp. 1757–1772, Jul. 2013.

[10] C. P. Diehl and J. B. Hampshire, “Real-time object classification and novelty detection for collaborative video surveillance,” in Proc. Int. Joint Conf. Neural Networks (IJCNN), 2002, pp. 2620–2625.

[11] H. Idrees, M. Shah, and R. Surette, “Enhancing camera surveillance using computer vision: A research note,” Policing, Int. J., vol. 41, no. 2, pp. 292–307, Apr. 2018.

[12] Q. Zhou, S. He, H. Liu, T. Chen, and J. Chen, “Pull & push: Leveraging differential knowledge distillation for efficient unsupervised anomaly detection and localization,” IEEE Trans. Circuits Syst. Video Technol., vol. 33, no. 5, pp. 2176–2189, May 2023.

[13] K. Wu, L. Zhu, W. Shi, W. Wang, and J. Wu, “Self-attention memory-augmented wavelet-CNN for anomaly detection,” IEEE Trans. Circuits Syst. Video Technol., vol. 33, no. 3, pp. 1374–1385, Mar. 2022.

[14] Y. Zhong, X. Chen, Y. Hu, P. Tang, and F. Ren, “Bidirectional spatio-temporal feature learning with multiscale evaluation for video anomaly detection,” IEEE Trans. Circuits Syst. Video Technol., vol. 32, no. 12, pp. 8285–8296, Dec. 2022.

[15] J. Guo, P. Zheng, and J. Huang, “Efficient privacy-preserving anomaly detection and localization in bigrang video,” IEEE Trans. Circuits Syst. Video Technol., vol. 30, no. 9, pp. 3268–3281, Sep. 2020.

[16] X. Mo, V. Monga, R. Bala, and Z. Fan, “Adaptive sparse representations for video anomaly detection,” IEEE Trans. Circuits Syst. Video Technol., vol. 24, no. 4, pp. 631–645, Apr. 2014.

[17] D. Hendrycks and K. Gimpel, “A baseline for detecting misclassified and out-of-distribution examples in neural networks,” in Proc. Int. Conf. Learn. Represent., 2017, pp. 1–11.

[18] S. Jung, J. Lee, D. Gwak, S. Choi, and J. Choo, “Standardized max logits: A simple yet effective approach for identifying unexpected road obstacles in urban-scene segmentation,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2021, pp. 15405–15414.

[19] J. Cen, P. Yun, J. Cai, M. Yu Wang, and M. Liu, “Deep metric learning for open world semantic segmentation,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2021, pp. 15313–15322.

[20] K. G. Mehrotra, C. K. Mohan, and H. Huang, “Detecting the unexpected via image resynthesis,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 2152–2161.

[21] Z. Dong, Y. Zhi, L. Fengze, S. Wei, and Y. Alan, “Synthesize then compare: Detecting failures and anomalies for semantic segmentation,” in Proc. Eur. Conf. Comput. Vis. (ECCV), 2020, pp. 145–161.

[22] C. Leys, O. Klein, Y. Dominicy, and C. Ley, “Detecting multivariate outliers: Use a robust variant of the Mahalanobis distance,” J. Exp. Social Psychol., vol. 74, pp. 150–156, Jan. 2018.

[23] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, “LOF: Identifying density-based local outliers,” in Proc. ACM SIGMOD Int. Conf. Manage. Data, 2000, pp. 93–104.

[24] D. Hendrycks et al., “Scaling out-of-distribution detection for real-world settings,” 2019, arXiv:1911.11132.

[25] P. Bevandič, I. Krčo, M. Oršić, and S. Šegvić, “Dense outlier detection and open-set recognition based on training with noisy negative images,” 2021, arXiv:2101.09193.

[26] Z. Li et al., “Detecting failures and anomalies for semantic segmentation,” in Proc. Adv. Neural Inf. Process. Syst., vol. 32, pp. 1–12.

[27] R. Chan, M. Rottmann, and H. Gottschalk, “Entropy maximization and meta classification for out-of-distribution detection in semantic segmentation,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2021, pp. 5108–5117.

[28] Y. Liu et al., “Residual pattern learning for pixel-wise out-of-distribution detection in semantic segmentation,” 2022, arXiv:2211.14512.

[29] T.-Y. Lin et al., “Microsoft COCO: Common objects in context,” in Proc. ECCV, 2014, pp. 1–12.
[31] M. Gricic, P. Bevandic, and S. Segvic, “DenseHybrid: Hybrid anomaly detection for dense open-set recognition,” in Proc. ECCV, 2022, pp. 500–517.

[32] Y. Tu et al., “Self-supervised likelihood estimation with energy guidance for anomaly segmentation in urban scenes,” 2023, arXiv:2302.06815.

[33] Y. Tian, Y. Liu, G. Pang, F. Liu, Y. Chen, and G. Carneiro, “Pixel-wise energy-based abstention learning for anomaly segmentation on complex urban driving scenes,” in Proc. ECCV, 2022, pp. 246–263.

[34] I. J. Goodfellow et al., “Generative adversarial networks,” Commun. ACM, vol. 63, no. 11, pp. 139–144, Nov. 2020, doi: 10.1145/3422622.

[35] T. Park, M. Liu, T. Wang, and J. Zhu, “Semantic image synthesis with spatially-adaptive normalization,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 2332–2341.

[36] D. Haldimann, H. Blum, R. Siegwart, and C. Cadena, “This is not what I imagined: Error detection for semantic segmentation through visual dissimilarity,” 2019, arXiv:1909.00976.

[37] G. Di Biase, H. Blum, R. Siegwart, and C. Cadena, “Pixel-wise anomaly detection in complex driving scenes,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 16913–16922.

[38] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 770–778.

[39] N. Tishby, F. C. Pereira, and W. Bialek, “The information bottleneck method,” 2000, arXiv:physics/0004057.

[40] D. Bogdoli, S. Uhlemeyer, K. Kowol, and J. Marius Zöllner, “Perception datasets for anomaly detection in autonomous driving: A survey,” 2023, arXiv:2302.02790.

[41] P. Pinggera, S. Ramos, S. Gehrig, U. Franke, C. Rother, and R. Mester, “Lost and found: Detecting small road hazards for self-driving vehicles,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS), Oct. 2016, pp. 1099–1106.

[42] M. Cordts et al., “The cityscapes dataset for semantic urban scene understanding,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 3213–3223.

[43] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, “The Pascal visual object classes (VOC) challenge,” Int. J. Comput. Vis., vol. 88, pp. 303–338, Jun. 2010.

[44] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, “Pyramid scene parsing network,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 6230–6239.

[45] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, “Encoder–decoder with atrous separable convolution for semantic image segmentation,” in Proc. ECCV, 2018, pp. 801–818.

[46] I. Mukhoti and Y. Gal, “Evaluating Bayesian deep learning methods for semantic segmentation,” 2018, arXiv:1811.12709.

[47] A. Malinin and M. J. Gales, “Predictive uncertainty estimation via prior networks,” in Proc. NeurIPS, 2018, pp. 1–11.

[48] Y. Gal and Z. Ghahramani, “Dropout as a Bayesian approximation: Representing model uncertainty in deep learning,” in Proc. IML, 2016, pp. 1050–1059.

[49] Shiyi An, W. Grathwohl, K. Wang, J. Jacobsen, D. Duvenaud, M. Norouzi, and K. Swersky, “Your classifier is secretly an energy based model and you should treat it like one,” in Proc. ICLR, 2020, pp. 1–23.