Efficient Unknown Object Detection with Discrepancy Networks for Semantic Segmentation

Ryo Kamoi*
The University of Texas at Austin
Department of Computer Science
ryokamoi@utexas.edu

Takumi Iida
SenseTime Japan

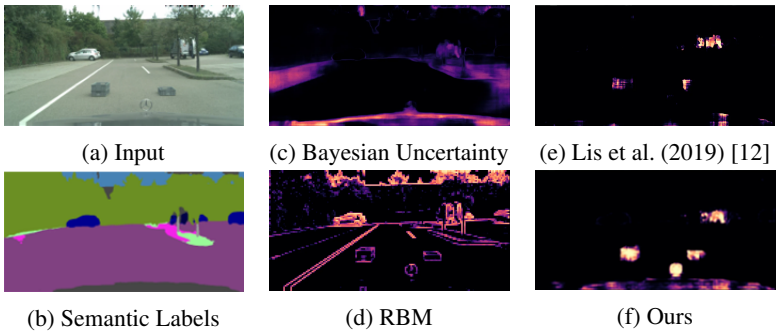
Kaname Tomite
SenseTime Japan

Abstract

Detecting unknown objects such as lost cargo is essential for improving the safety of self-driving cars. This is the first work focusing on reducing the computational cost of discrepancy networks for unknown object detection on monocular camera images. We propose an efficient discrepancy networks based solely on semantic segmentation, which has 50% fewer parameters and is 140% faster inference speed compared to an existing method, while improving detection performance by a large margin. In a major departure from prior work, we remove GANs from discrepancy networks. While previous studies have used GANs as a necessary component, our model outperforms them without using it. We further improve detection performance by analyzing intermediate representations and introducing *feature selection* and *deep supervision*. Our experiments on three datasets for obstacle detection show significant improvement of more than 5% in AUROC.

1 Introduction

Computer vision tasks such as object detection and semantic segmentation play important roles in the development of self-driving systems. However, semantic segmentation or object detection systems often cannot detect unknown objects that are not in training data. Once self-driving cars become used in the real world, they will have to deal with objects on the road that the systems



(a) Input (c) Bayesian Uncertainty (e) Lis et al. (2019) [12]
(b) Semantic Labels (d) RBM (f) Ours

Figure 1: Example detection results.

*This work was done during an internship at SenseTime Japan.

Machine Learning for Autonomous Driving Workshop at the 35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.

have never seen before to ensure safe driving. Unknown object detection, a method to detect objects that did not appear during training, is essential for improving the safety of self-driving systems.

There are many works of obstacle detection based on informative inputs such as LiDAR [7] or stereo camera [1, 14]. However, methods relying on geometric information from these devices are often computationally expensive and affected by the ability of the equipment. (e.g. maximum measurable distance of LiDAR). Unknown object detection on monocular camera images is a challenging task, but one possible approach to alleviating these problems. Recently, improvements have been made in this area by a new method named discrepancy networks [12, 13, 2], a model comparing images by using the intermediate representations from feature extractors such as VGG-16. However, they improve performance by using expensive models such as GANs or providing additional information such as uncertainty. Due to the limited computational resources and demands for latency on autonomous driving, fast and efficient models with a lower number of parameters are desired. This work is the first work focusing on reducing the computational cost of discrepancy networks for detecting unknown objects without degrading performance.

We propose a new unknown object detection method based on discrepancy networks by locating errors in semantic segmentation estimation, which is computationally efficient while achieving better performance. The proposed method performs better without using GANs, while prior work in discrepancy networks [12, 17, 13, 2] and general anomaly detection [15, 6] often rely on resynthesized images from GANs. We propose three methods to improve the detection performance. By making use of the low computational cost of our model without GAN, we propose *dataset regeneration*, a simple training technique that improves detecting performance and generalizability of the model significantly. In addition, we analyzed intermediate representations of our discrepancy networks, and propose to apply **feature selection** and **deep supervision** [18, 9]. Our observation reveals that too shallow features detect surface texture such as edges, and too deep features ignore unknown objects (Figure 6). To make matters worse, we observe that too deep features can harm the overall model performance significantly. Therefore, we propose *feature selection* to remove harmful features, and choose proper features that detect unknown objects effectively. We also find that discrepancy networks tend to focus on information from shallower features and are often sensitive to surface textures such as edges, and propose to apply *deep supervision* to force discrepancy networks to use information from intermediate features.

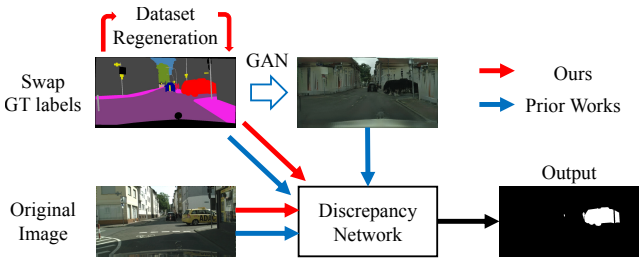


Figure 2: Inputs to our model and models of previous studies.

Algorithm 1: Training Procedure of Our Model

input : Discrepancy networks $\mathcal{D}$ with parameters θ,
Pretrained feature extractror $\mathcal{F}$,
Indices of features S (*feature selection*),
Training dataset X

$\theta \leftarrow$ initialization

for $epoch \leftarrow 1$ **to** n **do**
  Dataset Regeneration – generate discrepancy dataset X_{epoch} from X by swapping semantic segmentation labels of randomly selected instances
  for $x, y \leftarrow$ random batch from X_{epoch} **do**
    extract features $f_i, i \in S$ from $\mathcal{F}(x)$
    $\theta \leftarrow Adam(\theta, L_\theta(x, y))$ (Eq. 3)
  end
end

We evaluate our model on three datasets for obstacle detection and demonstrates that it outperforms prior models with a larger number of parameters. Figure 1 shows example detection results of our model and baseline methods, showing ours detect the obstacles most correctly. Our model has 50% fewer parameters and 140% faster inference speed compared to an existing method, while improving detection performance by more than 5% in the area under the ROC curve (AUROC, AUC).

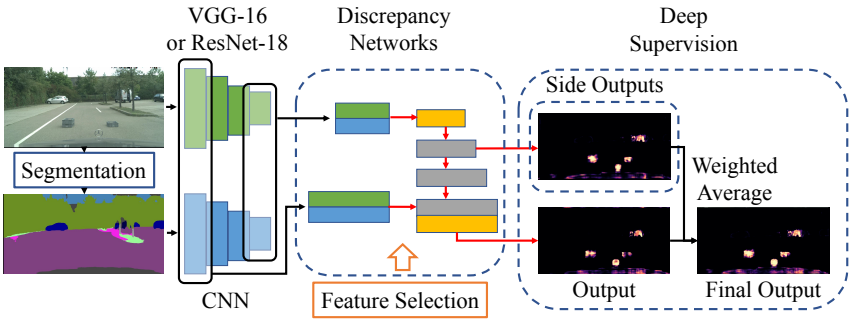


Figure 3: Overall structure of our model. It compares features of the original image and semantic labels to detect errors in semantic segmentation estimation. *Deep supervision* using side outputs from intermediate layers of discrepancy networks makes effective use of information in intermediate representations. We can use any combination of features from feature extractors, and two features $S = \{1, 4\}$ are used in this figure. Red arrows represent convolutional networks.

2 Semantic Segmentation Discrepancy Networks

Discrepancy Networks in Prior Work Discrepancy networks [12, 17, 13, 2] is a model in unknown object detection, motivated by the idea that unknown objects cannot be properly resynthesized by GANs. During inference, the model predicts semantic segmentation labels of an input image, and GAN restores the input image from the predicted semantic segmentation. Since there is no correct semantic segmentation label for unknown objects, GAN cannot restore the region where there are unknown objects. Finally, discrepancy networks compare intermediate representations of feature extractors of the original and GAN-resynthesized image, and semantic segmentation estimation to detect unknown objects. In many works, pretrained feature extractors such as VGG-16 are used for the original and GAN-resynthesized images, and a CNN is used as a feature extractor for semantic segmentation.

This method does not require datasets with obstacles (fallen objects like tires and bumpers), which are difficult to prepare on a large scale, during training. For training, this method makes a discrepancy dataset by swapping semantic segmentation labels of randomly chosen instances and resynthesizing images with GAN from the mistaken labels (Figure 2). As a result, regions where semantic segmentation labels are swapped cannot be restored properly by GAN. The discrepancy network is trained to detect regions where the semantic segmentation labels are swapped. This method can perform supervised learning because we know regions where the labels are swapped. Consequently, discrepancy networks are trained to detect regions that are not properly restored, and are expected to detect unknown object when an image with obstacles are given.

While this method achieves state-of-the-art performance on monocular camera images, the GAN introduces large computational cost, and also there is much room to improve detection performance. For instance, we observe that this method often detects unknown objects partially, and seems to be sensitive to surface structures such as edges.

Semantic Segmentation Discrepancy Networks We propose discrepancy networks solely based on semantic segmentation. While GANs play an important role in existing unknown object detection [12, 17, 13, 2], we theoretically and experimentally show that GANs are not necessary for discrepancy networks. In the discrepancy networks in prior work [12] (Figure 2), images resynthesized by GAN are only dependent on semantic segmentation estimation. Thus the original image and GAN image are conditionally independent given semantic segmentation estimation. In other words, GAN images provide no information when semantic segmentation labels are given. This observation motivates us to propose discrepancy networks without GANs.

Dataset Regeneration *Dataset regeneration* is a method that modifies the training procedure of discrepancy networks to generate a discrepancy dataset (swap labels of randomly chosen instances) after every epoch. Previous work [12] generates discrepancy dataset once before training because resynthesis by GANs is computationally expensive. In contrast, our model just swaps semantic

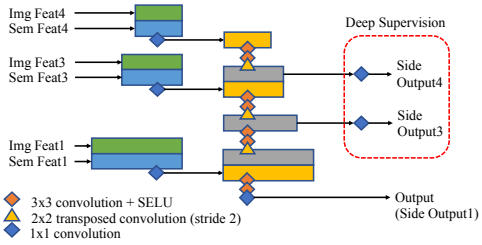


Figure 4: The structure of our discrepancy networks. $S = \{1, 3, 4\}$ in this figure.

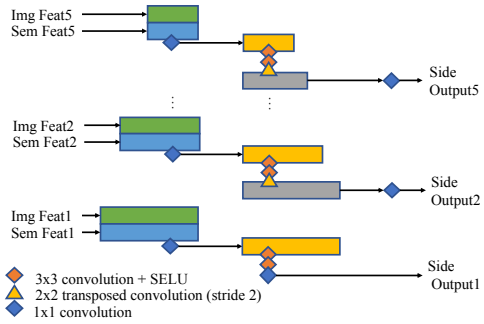


Figure 5: Feature selection model.

segmentation labels of randomly selected instances, is able to generate a dataset before every epoch (Algorithm 1). This simple idea significantly improves unknown object detection performance.

3 Deep Supervision

Deep supervision [18, 9], a method of applying supervised training and utilizing side outputs from intermediate layers, improves the performance of various computer vision tasks including saliency detection. Since unknown object detection is a task with similar characteristics to saliency detection, which requires the detection of arbitrary objects, it would be effective to refer to the ideas in the models used for saliency detection. As the first study in this direction, we propose to apply *deep supervision* by generating side outputs from intermediate layers of discrepancy networks and applying supervised learning for them (Figure 3).

In this section, we describe our deep supervision for discrepancy networks. Side outputs $side_i(x), i \in S$, where x is the input image, are generated by discrepancy networks as described in Figure 4. In specific, 1×1 convolution is applied to intermediate features to generate side outputs. Finally, bilinear interpolation is applied to side outputs to match the size to the original image. Two types of losses, a side loss $L_{side}(x, y|\theta)$ and fusion loss $L_{fuse}(x, y|\theta, \mathbf{w})$, will be used during training. The side loss is the sum of cross entropy losses for all side-outputs, for training so that all side-outputs to be close to the target. In addition, this model returns a weighted average of side outputs as the fused output,

$$s_{fuse}(x) = \sum_{i \in S} w_i side_i(x), \tag{1}$$

and the weights $\mathbf{w} = \{w_i\}$ will be also trained. The fuse loss is designed so that the weighted sum, which will be the final output, will be close to the target. The resulting training loss is the sum of the fusion and side losses

$$L_{final}(x, y|\theta, \mathbf{w}) = L_{fuse}(x, y|\theta, \mathbf{w}) + L_{side}(x, y|\theta) \tag{2}$$

$$= l(s_{fuse}(x), y) + \frac{1}{|S|} \sum_{i \in S} l(side_i(x), y) \tag{3}$$

where θ denotes parameters of discrepancy networks, y is the target label, and l is cross entropy loss.

However, we experimentally observe that the trained weighted average s_{fuse} is not the best output for test data with unknown objects, and found that side outputs from deep layers are often useful for test data. Therefore, as the final output for test data, we take the pixel-wise maximum score of the learned weighted average $s_{fuse}(x)$ and the side output from the deepest feature $s_m(x)$

$$s_{test}(x) = \max(s_{fuse}(x), s_m(x)). \tag{4}$$

The difference between the inference and learning situations, where the training data does not contain any unknown objects, can cause unpredictable behavior and requires attention.

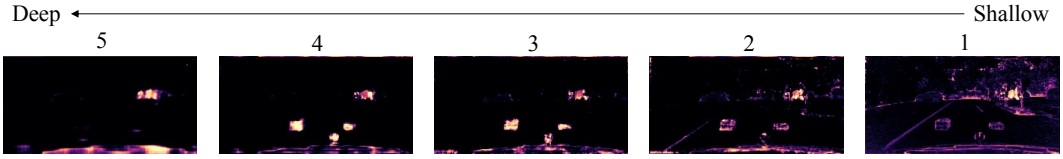


Figure 6: Different features from the feature extractor exhibit different properties towards unknown objects. While the third and forth features properly detect boxes on the road (unknown objects), the deepest (fifth) feature loses them. Shallow (first and second) features detect edges of almost all objects. The results are generated by using the model described in Figure 5 with VGG-16. Please refer to Figure 1 for the original image and semantic segmentation results.

4 Feature Selection

We find unexpected behaviors of intermediate representations of discrepancy networks towards unknown objects. This observation motivates us to propose *feature selection* for discrepancy networks, and further justifies the use of *deep supervision*.

4.1 Different Layers Behave Differently Towards Unknown Objects

We find that intermediate representations from different layers of feature extractors such as VGG-16 exhibit unexpected behaviors to unknown objects when used in discrepancy networks. Specifically, shallow features are sensitive to surface textures such as edges, too deep features ignore unknown objects, and specific intermediate features are useful for unknown object detection.

We analyze the behavior of each feature by using a model without using results from the previous layers (Figure 5) and training the model in the same way as the discrepancy networks with deep supervision (Equation 3). Figure 6 shows side outputs of the model in Figure 5, suggesting that shallow layers (1, 2) detect surface structure such as edges, intermediate features (3, 4) properly detect unknown objects (boxes), and deeper feature (5) ignores the boxes. Our further experiments show that this phenomenon occurs with other unknown objects as well (Table 5). Moreover, we observe that the overall model performance is harmed when too deep features ignoring unknown objects are included in the model (Table 6), while they work properly for training data. As an example, networks using five layers from VGG-16 ignores unknown objects while the model using four layers works well. This phenomenon is probably caused by the difference between the situation during training and inference because unknown objects are not used in training of the discrepancy networks, and cannot be solved by using training data which is not including obstacles. This is an alarming observation that affects all discrepancy networks.

4.2 Feature Selection

The above observations suggest that unknown object detection performance of discrepancy networks is sensitive to *feature selection*. Although discrepancy networks can be trained on datasets that do not contain obstacles, it is not possible to perform *feature selection* by using training data because harmful features can detect known instances in training data properly. However, it is computationally expensive to train and evaluate models with all possible combinations of features. We propose a method to choose proper features by using the model we used for the analysis in the previous section f_{selec} (Figure 5). Our *feature selection* use side-outputs from the model f_{selec}, whose each layer does not receive information from previous layers. Same as the previous chapter, we train this model in the same way as our discrepancy networks including *deep supervision* (Equation 3, and evaluate side-outputs on data containing unknown objects D_{eval}. Our *feature selection* utilizes detection performance (we use AUROC as a metric) of the output from the features of each layer, and chooses features with top-k scores. However, we choose to always include the shallowest feature $i = 1$ because using only deep features would result in low resolution. Finally, our *feature selection* selects features indices as

$$S = \{1\} \cup \underset{i \in \{2,\ldots,n\}}{\arg\,\text{top-}k}\, \text{AUROC}(side_i(\cdot, f_{selec}),\, D_{eval}), \tag{5}$$

where $side_i(\cdot, f_{selec})$ denotes the i-th side output from *feature selection* model. In Section 5, we demonstrate this method can choose proper features that work well on unseen data by using relatively small dataset with obstacles.

Finally, we use a generalized discrepancy network in which any combination of feature layers S from feature extractors can be used (e.g. $S = \{1, 4\}$ in Figure 3 and $S = \{1, 3, 4\}$ in Figure 4). For the i-th $(i \in S)$ layer, features from original image and semantic segmentation labels are concatenated and fused by 1×1 convolution. The fused features are passed to the discrepancy network, and it is concatenated with the previous result (if it is not the first feature). Then, they are fused by using convolutional networks. For the i-th $(i \notin S)$ component in discrepancy networks, new features are not concatenated and convolutions are applied to the current feature (Figure 4).

5 Experiments

We show that our method has a smaller number of parameters and reduces inference time significantly, while improving the detecting performance by a large margin. We also analyze *feature selection* and show it is necessary and effective.

5.1 Datasets and Implementation Details

In this experiment, our discrepancy networks are trained on Cityscapes dataset [4]. In addition, our model requires a pretrained semantic segmentation model. We use Bayesian SegNet [10] and PSP Net [20] trained on BDD100K dataset [19]. We evaluate our unknown object detection method on three obstacle detection datasets: Lost and Found (LAF) [14], Small Obstacle Dataset (SOD) [16], and Road Anomaly (RA) [12].

In our model, side outputs are generated from features just before being concatenated with the next feature from features extractors. We apply bilinear interpolation to side outputs to match the size to the final output. For training of our model, we use weighted cross entropy loss in all settings. Ego vehicle is ignored during training and evaluation. We train 100 epochs with Adam optimizer [11]. The initial learning rate is $1\mathrm{e}{-4}$ and reduce by a factor of ten when there is no improvement after the initial five epochs.

5.2 Baseline Methods

We compare our model with four baseline methods. The first one is the discrepancy networks by [12]. Compared to our model, it compares the original image to a resynthesized image by GAN. We also evaluate two uncertainty models: Bayesian uncertainty and ensemble. We use Bayesian SegNet and PSP Net, which are used in discrepancy networks, to evaluate uncertainty. The use of uncertainty to unknown object detection has been proposed in previous work [3, 8]. Finally, we evaluate the Restricted Boltzmann Machine (RBM) autoencoder to resynthesize road texture corrupted by Gaussian noise [5].

5.3 Parameter Size and Inference Time

Compared to a previous method using discrepancy networks with GANs, our model reduces the number of parameters significantly to less than half, even after *deep supervision* is added. Here we compare the computational cost of our model with the method by [12]. For both models, we use Bayesian SegNet for semantic segmentation and VGG-16 for feature extraction. Table 1 shows our method reduces parameter size and inference time, while improving detection performance significantly as described later. The GAN model in [12] has 182 million parameters, but our *deep supervision* adds only one thousand parameters. As a result, the overall size of parameters is reduced from 372 million to 184 million (reduced by 51%). Inference time for one image is also reduced from 3.6 seconds to 2.6 seconds (reduced by 28%). We evaluate our inference time on 51 images in Lost and Found dataset on NVIDIA TITAN V. Semantic segmentation takes a long time because we use a Bayesian model in our experiments, so we can save time even more by using a single model.

5.4 Ablation Study and Comparison to Baseline Methods

Table 1: Parameter size and inference time of our model and [12]. We use Bayesian SegNet as a semantic segmentation model, and VGG-16 as a feature extractor. For Comparison purpose, we use feature one to four $S = \{1, 2, 3, 4\}$. We show an average inference time for 51 images in Lost and Found dataset on NVIDIA TITAN V.

(a) Parameter Size

Model	Ours	[12]
GAN	—	182,543,619
Deep Supervision	1,304	—
Discrepancy Net	5,808,770	6,163,266
SemSeg	39,799,609	39,799,609
VGG-16	138,423,208	138,423,208
Sum	184,032,891	372,738,472
Ratio	49	100

(b) Inference time for one image (seconds)

Model	Ours	[12]
GAN	—	1.0
Discrepancy Net	0.2	0.2
SemSeg	2.4	2.4
Sum	2.6	3.6
Ratio	72	100

Table 2: AUROC of unknown object detection of our models and models proposed by [12] using four layers $S = \{1, 2, 3, 4\}$. ReGen and DSup denote *dataset regeneration* at every epoch and *deep supervision* proposed in this paper. LAF, SOD, and RA denote three datasets we evaluate.

(a) VGG-16

GAN	ReGen	DSup	Bayesian SegNet			PSP Net		
			LAF	SOD	RA	LAF	SOD	RA
✓			83.24(±1.89)	78.99(±6.51)	72.69(±2.47)	82.78(±1.58)	80.08(±5.97)	76.65(±3.02)
			80.95(±3.45)	73.22(±7.12)	73.72(±1.31)	80.99(±3.07)	74.11(±7.56)	75.32(±2.44)
	✓		88.35(±1.31)	83.43(±3.11)	78.33(±0.45)	87.92(±1.30)	83.91(±2.86)	81.16(±0.58)
	✓	✓	**91.88(±0.77)**	**89.27(±0.61)**	**79.25(±0.69)**	**91.59(±0.68)**	**89.44(±0.40)**	**81.87(±0.65)**

(b) ResNet-18

GAN	ReGen	DSup	Bayesian SegNet			PSP Net		
			LAF	SOD	RA	LAF	SOD	RA
✓			79.21(±3.30)	67.54(±1.53)	75.55(±1.64)	78.75(±2.30)	67.20(±2.11)	76.39(±2.50)
			82.80(±3.61)	77.88(±4.85)	72.76(±1.69)	82.52(±3.10)	78.21(±4.63)	73.94(±4.25)
	✓		87.09(±0.41)	**83.24(±2.46)**	78.13(±0.66)	86.67(±0.43)	**83.52(±2.74)**	80.66(±0.73)
	✓	✓	**87.48(±2.69)**	78.09(±2.74)	**81.78(±0.86)**	**88.08(±2.43)**	79.32(±2.83)	**83.55(±1.02)**

Table 3: AUROC of unknown object detection of our models using five layers $S = \{1, 2, 3, 4, 5\}$. Compared to results in Table 2 for four layers $S = \{1, 2, 3, 4\}$, detection performance is reduced when the fifth feature is used. This result shows that *feature selection* is necessary for discrepancy networks. We do not use GANs in this experiment.

Feature	ReGen	DSup	Bayesian SegNet			PSP Net		
			LAF	SOD	RA	LAF	SOD	RA
VGG-16	✓		79.67(±0.88)	78.90(±3.66)	74.79(±1.39)	80.64(±0.97)	78.08(±4.53)	78.23(±1.71)
ResNet-18	✓		77.72(±2.27)	74.86(±3.66)	73.64(±1.03)	79.21(±1.96)	74.28(±3.78)	78.03(±2.38)

We compare our model to baseline methods. For discrepancy network models, we train five times and show means and standard deviations of AUROC. Based on the *feature selection* results explained later, we use $S = \{1, 2, 3, 4\}$ for all discrepancy networks.

Table 4: AUROC of baseline methods. We use Bayesian uncertainty for Bayesian SegNet model, and ensemble uncertainty for PSP Net model.

	LAF	SOD	RA
Uncertainty (Bayesian SegNet)	67.72	72.81	70.64
Uncertainty (PSP Net)	63.64	70.16	66.89
RBM	71.43	83.32	58.70

Table 2 shows ablation study and comparison to the discrepancy network by [12]. The model on the top lines (using GAN, no *dataset regeneration*) is the model by [12], and the rest are our models. The results show that our model outperforms the baseline model when trained

Table 5: AUROC of side outputs from different features of *feature selection* model in Figure 5. Feature 1 is from the shallowest layer, and layer 5 is from the deepest layer. VGG-16 and ResNet-18 are feature extractors, Bayesian SegNet and PSP Net are semantic segmentation models, LAF, SOD, and RA are test datasets.

Feature	VGG-16						ResNet-18					
	Bayesian SegNet			PSP Net			Bayesian SegNet			PSP Net		
	LAF	SOD	RA	LAF	SOD	RA	LAF	SOD	RA	LAF	SOD	RA
1	79.56	69.14	72.70	79.97	67.66	76.04	64.22	59.53	70.39	69.55	56.92	72.01
2	91.20	88.09	77.63	90.62	87.90	81.44	85.16	74.10	79.00	85.53	74.89	79.87
3	**92.59**	**94.94**	**79.34**	**92.38**	**95.27**	**82.12**	**88.08**	**93.63**	**80.63**	**87.63**	**93.26**	**83.65**
4	89.20	88.03	78.03	88.88	88.62	76.22	85.18	88.54	75.69	85.30	88.48	76.82
5	81.14	75.92	69.21	80.88	77.11	66.68	78.05	80.86	72.90	78.72	80.75	74.07

Table 6: AUROC of unknown object detection using different sets of features. "ReGen" and "DSup" stand for *dataset regeneration* and *deep supervision*.

Feature S	ReGen	DSup	VGG-16						ResNet-18					
			Bayesian SegNet			PSP Net			Bayesian SegNet			PSP Net		
			LAF	SOD	RA	LAF	SOD	RA	LAF	SOD	RA	LAF	SOD	RA
1, 2	✓	✓	90.71	88.46	78.73	90.36	89.21	76.43	88.89	89.52	78.17	88.80	89.59	76.79
1, 3	✓	✓	**93.04**	**95.00**	**82.43**	**92.79**	**94.95**	**82.98**	**91.16**	**95.13**	82.12	**90.46**	**95.06**	**84.06**
1, 4	✓	✓	91.61	89.94	78.61	91.16	90.65	81.28	88.84	78.65	80.36	88.89	79.31	81.73
1, 5	✓	✓	81.30	74.41	74.36	81.21	73.72	76.45	71.24	73.60	**82.45**	72.89	74.29	82.77
1, 2	✓		90.61	88.50	79.00	90.36	89.20	73.94	88.58	87.04	80.95	88.41	87.42	79.55
1, 3	✓		**91.68**	**93.67**	**82.44**	**91.35**	**93.75**	82.02	**91.79**	**94.47**	**81.91**	**91.52**	**94.16**	**82.17**
1, 4	✓		89.27	86.15	81.17	88.72	87.38	**83.61**	86.94	85.32	78.70	86.52	85.69	81.93
1, 5	✓		75.22	63.50	72.92	76.40	64.17	77.84	76.37	60.10	74.01	77.19	59.70	75.93

with *dataset regeneration* and *deep supervision*. Additionally, in comparison to baseline methods in Table 4, our method outperforms by a large margin.

The results also show that the effect of GAN is limited, which is consistent with my argument showing that GAN is theoretically unnecessary. The reason for the slight improvement can be attributed to the fact that the GAN image is in the same space as the original image, making it easier to compare. The performance difference of models with and without GAN in our experiment is smaller than what reported by [12]; 83.24 and 80.95 in our experiments, and 82 and 77 in the original paper. While the original paper only showed the results of a single training session, we observed that the discrepancy networks by [12] has a large variation in results, and the differences in the average of multiple training showed are small.

The ablation study of our method shows that both *dataset regeneration* and *deep supervision* improve detection performance in almost all settings, and more effective than adding expensive GANs. In addition, we observe that *dataset regeneration* reduces the deviation of the final performance.

5.5 Feature Selection

First, we demonstrate that *feature selection* is necessary for discrepancy networks. Table 2 shows results using four layers $S = \{1, 2, 3, 4\}$ from feature extractors, and Table 3 shows results for $S = \{1, 2, 3, 4, 5\}$. The results show that models using five layers perform worse than those using four layers. For example, models using VGG-16 achieve 88.35 in AUROC on LAF dataset when four layers are used, but it is reduced to 79.67 when five layers are used. These results suggest that we have to remove harmful features before training. We empirically observe that too deep features can harm the final results, but shallow features do not affect the final results significantly. This behavior may come from the difference between training and inference of our discrepancy networks model. We cannot fix this problem by using training data since the harmful features still work well on training data, so we have to apply *feature selection* by using dataset containing obstacles.

Next, we show that our *feature selection* method can remove harmful features properly. Table 5 shows detection performance of each side output from our *feature selection* model described in Figure 5. For both VGG-16 and ResNet-18, the third side-output exhibits the highest performance on all test data,

and the fifth side-output seems to be too deep and harmful. Therefore, our *feature selection* concludes that the third feature is useful for unknown instance detection, and the fifth side-output can be harmful. To show that our *feature selection* model is correctly working, we evaluate discrepancy networks using two features (i.e. $k = 1$ in Equation 5) from feature extractors (Table 6). We use the shallowest feature in all cases because it has the highest resolution i.e. $S = \{1, i\}, i = \{2, 3, 4, 5\}$ as explained in Equation 5. For both VGG-16 and ResNet-18, $S = \{1, 3\}$ exhibit the highest performances and $S = \{1, 5\}$ exhibit the worst performances in most cases. These results suggest that our *feature selection* is effective, and independent of semantic segmentation models and datasets.

6 Conclusion

We propose a new unknown instance detection method using discrepancy in semantic segmentation, which outperforms prior models while reducing the number of parameters and inference time. Specifically, our method does not use GANs, which have been often used in unknown instance detection and general anomaly detection methods. To improve performance without increasing the amount of computation significantly, we analyze behaviors of intermediate representations in feature extractors for unknown objects, and propose *feature selection* method and *deep supervision* for discrepancy networks.

Our observation suggests that deep features of pretrained feature extractors can exhibit unexpected behaviors towards unknown objects. This result motivates further work not only in unknown object detection, but in all tasks related to the robustness of pretrained feature extractors.

References

[1] Nicola Bernini, Massimo Bertozzi, Luca Castangia, Marco Patander, and Mario Sabbatelli. Real-Time Obstacle Detection using Stereo Vision for Autonomous Ground Vehicles: A Survey. In *International Conference on Intelligent Transportation Systems (ITSC)*, 2014.

[2] Giancarlo Di Biase, Eth Zurich, Hermann Blum, Roland Siegwart, and Cesar Cadena. Pixel-wise Anomaly Detection in Complex Driving Scenes. *arXiv preprint arXiv:2103.05445*, 2021.

[3] Hermann Blum, Paul-Edouard Sarlin, Juan Nieto, Roland Siegwart, and Cesar Cadena. The Fishyscapes Benchmark: Measuring Blind Spots in Semantic Segmentation. *arXiv preprint arXiv:1904.03215*, 2019.

[4] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, Bernt Schiele, Daimler Ag R&d, and T U Darmstadt. The Cityscapes Dataset for Semantic Urban Scene Understanding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[5] Clement Creusot and Asim Munawar. Real-time small obstacle detection on highways using compressive RBM road reconstruction. In *IEEE Intelligent Vehicles Symposium (IV)*, 2015.

[6] Lucas Deecke, Robert Vandermeulen, Lukas Ruff, Stephan Mandt, and Marius Kloft. Image Anomaly Detection with Generative Adversarial Networks. In *European Conference on Machine Learning*, 2018.

[7] Jaehyun Han, Dongchul Kim, and Myoungho Sunwoo. Enhanced Road Boundary and Obstacle Detection Using a Downward-Looking LIDAR Sensor. *IEEE Transactions on Vehicular Technology*, 61(3), 2012.

[8] Dan Hendrycks, Steven Basart, Mantas Mazeika, Mohammadreza Mostajabi, Jacob Steinhardt, and Dawn Song. A Benchmark for Anomaly Segmentation. *arXiv preprint arXiv:1911.11132*, 2019.

[9] Qibin Hou, Ming-Ming Cheng, Xiaowei Hu, Ali Borji, Zhuowen Tu, and Philip H S Torr. Deeply Supervised Salient Object Detection with Short Connections. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(4):815–828, 2019.

[10] Alex Kendall, Vijay Badrinarayanan, and Roberto Cipolla. Bayesian SegNet: Model Uncertainty in Deep Convolutional Encoder-Decoder Architectures for Scene Understanding. In *British Machine Vision Conference (BMVC)*, 2017.

[11] Diederik P Kingma, Jimmy Ba, and Jimmy Lei Ba. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations (ICLR)*, 2014.

[12] Krzysztof Lis, Krishna Nakka, Pascal Fua, and Mathieu Salzmann. Detecting the Unexpected via Image Resynthesis. In *International Conference on Computer Vision (ICCV)*, 2019.

[13] Toshiaki Ohgushi, Kenji Horiguchi, and Masao Yamanaka. Road Obstacle Detection Method Based on an Autoencoder with Semantic Segmentation. In *Asian Conference on Computer Vision (ACCV)*, 2020.

[14] Peter Pinggera, Sebastian Ramos, Stefan Gehrig, Uwe Franke, Carsten Rother, and Rudolf Mester. Lost and Found: Detecting Small Road Hazards for Self-Driving Vehicles. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016.

[15] Thomas Schlegl, Philipp Seeböck, Sebastian M Waldstein, Ursula Schmidt-Erfurth, and Georg Langs. Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery. In *Conference on Information Processing in Medical Imaging (IPMI)*, 2017.

[16] Aasheesh Singh, Aditya Kamireddypalli, Vineet Gandhi, and K Madhava Krishna. LiDAR guided Small obstacle Segmentation. *arXiv preprint arXiv:2003.05970*, 2020.

[17] Yingda Xia, Yi Zhang, Fengze Liu, Wei Shen, and Alan L Yuille. Synthesize then Compare: Detecting Failures and Anomalies for Semantic Segmentation. *arXiv preprint arXiv:2003.08440*, 2020.

[18] Saining Xie and Zhuowen Tu. Holistically-Nested Edge Detection. In *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[19] Fisher Yu, Wenqi Xian, Yingying Chen, Fangchen Liu, Mike Liao, Vashisht Madhavan, and Trevor Darrell. BDD100K: A Diverse Driving Video Database with Scalable Annotation Tooling. *arXiv preprint arXiv:1805.04687*, 2018.

[20] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid Scene Parsing Network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

A Implementation Details

In this section, we describe the implementation details of our model.

A.1 Discrepancy Networks

Figure 7 shows the full structure of our discrepancy networks for $S = \{0, 1, 2, 3, 4\}$. When we remove layers from this model, the corresponding features are not concatenated, and the corresponding side outputs are not generated. However, we do not change the size of the output channel.

A.2 Semantic Segmentation

Table 7 describes a feature extractor for semantic segmentation labels, made of simple convolutional layers. This model is trained concurrently with the training of discrepancy networks.

Table 8 shows the list of semantic segmentation labels used in our model. Since we use the same configuration as [12], our model does not use some labels in Cityscapes such as egovehicle. The loss function of discrepancy networks also ignores these labels.

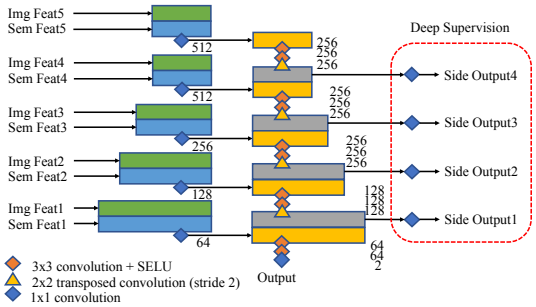


Figure 7: The full structure of our discrepancy networks for $S = \{0, 1, 2, 3, 4\}$. The numbers next to layers represent the size of the output channel.

Table 7: Feature extractor for semantic segmentation labels.

	kernel	output channel	output
1	7×7	32	sem feat 1
2	3×3	64	sem feat 2
3	3×3	128	sem feat 3
4	3×3	256	sem feat 4
5	3×3	256	sem feat 5

Table 8: Semantic segmentation label of our model. Some labels such as egovehicle in Cityscapes are ignored in our model. This setting is identical to [12].

label	name
0	road
1	sidewalk
2	building
3	wall
4	fence
5	pole
6	traffic light
7	traffic sign
8	vegetation
9	terrain
10	sky
11	person
12	rider
13	car
14	truck
15	bus
16	train
17	motorcycle
18	bicycle
(ignored)	unlabeled, egovehicle, rectification border, out of roi, static, dynamic, ground, parking, rall track, guard rail, bridge, tunnel, polegroup, caravan trailer

B Experiments

This section shows additional experimental results.

B.1 On Road

The detection performance on the road is more critical. Table 9 shows detection performance only on the road. In this result, the model performs inference on the entire picture, but the AUROCs are evaluated only road areas. This result also shows that our dataset regeneration and deep supervision are effective.

B.2 Example Outputs

We show example outputs of our model and a baseline method [12]. Figure 8 and 9 are detection performance for challenging cases. The color of boxes in Figure 8 are similar in color to the road, and the baseline method [12] misses the right box. The baseline method also misses a box on the nighttime road in Figure 9. In contrast, our method succeeded in detecting both obstacles.

Figure 10 shows the limitations of our method, missing the white box on the right. Our method often misses white objects, probably because we train it not to detect white lines on the road. We would have to fix our semantic segmentation labeling strategy to solve this problem.

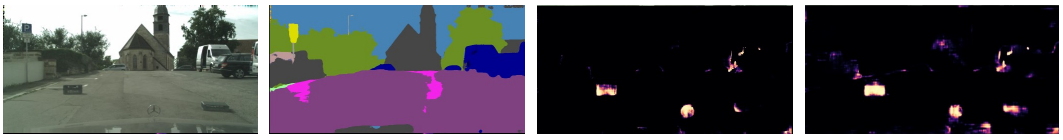
Table 9: AUROC of unknown instance detection evaluated only on road region.

GAN	ReGen	DSup	SegNet		ResNet	
			LAF	SOD	LAF	SOD
o			86.13(±2.12)	78.51(±6.20)	86.23(±1.99)	80.90(±5.80)
			84.17(±3.08)	73.02(±6.57)	84.63(±2.83)	74.45(±6.93)
	o		91.68(±1.11)	83.86(±2.24)	91.85(±1.16)	84.50(±2.16)
	o	HED	**95.60(±0.59)**	**88.99(±0.67)**	**95.75(±0.55)**	**89.56(±0.32)**

(a) VGG-16

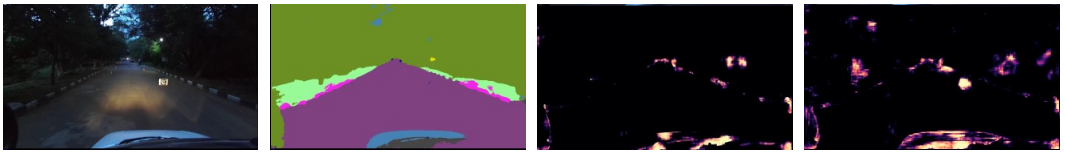
GAN	ReGen	DSup	SegNet		ResNet	
			LAF	SOD	LAF	SOD
o			86.13(±2.12)	78.51(±6.20)	86.23(±1.99)	80.90(±5.80)
			84.17(±3.08)	73.02(±6.57)	84.63(±2.83)	74.45(±6.93)
	o		91.68(±1.11)	83.86(±2.24)	91.85(±1.16)	84.50(±2.16)
	o	HED	**95.60(±0.59)**	**88.99(±0.67)**	**95.75(±0.55)**	**89.56(±0.32)**

(b) ResNet-34



Input Image | Semantic Segmentation | Lis et al. (2019) | Ours

Figure 8: Example Result from Lost and Found [14]



Input Image | Semantic Segmentation | Lis et al. (2019) | Ours

Figure 9: Example Result from Small Obstacle Dataset [16]

Input Image | Semantic Segmentation | Lis et al. (2019) | Ours

Figure 10: Example Result from Lost and Found [14] dataset (failure)