Improving Performance and Compute Efficiency of Anomaly Segmentation Models with COCO-DAG

Liyuan Mao Yilin Sun Zherui Huang
Shanghai Jiao Tong University

Abstract

Deep neural networks(DNNs) have achieved success on the semantic segmentation tasks, yet they must train on a pre-defined distribution of object classes, hence their performance decay when facing actual everyday scenes. Anomaly segmentation task aiming to detect out-of-distribution(OOD) objects requires the DNNs to successfully distinguish OOD objects from those in-distribution ones instead of classifying them as any given classes. Autonomous driving is an area where OOD detection is of great significance for the sake of safety. We propose an approach through variation of dilation and erosion operation(VDE) to refine the standardized max logits method which aligns the different distributions and reflects relative meanings of max logits within each predicted class. Moreover, we propose a method based on the moment generating function as the characteristic function for a random variable. Based on this method, we are able to collect all information corresponding to given random variable. We further propose OOD detection with moment configuration(OODMC), a method which can improve the performance of existing models. We also improved existing training datasets with novel ways of data augmentation upon the COCO dataset(COCO-DAG), which highly improve the compute efficiency of existing training procedure. We conducted experiments with self-built datasets simulating actual scenes with OOD samples and showed that our method is a betterment of existing methods.

1. Introduction

Modern deep neural networks have shown their abilities on various computer vision tasks. In particular, the advancement of deep learning techniques have enabled many semantic segmentation models to work, however, these models are generally trained on a predefined closed set of objects classes [4, 10]. These training procedure assumes that the training data and the actual data has the same kind of distributions, hence when facing actual everyday scenes the performace of these models significantly degrades. Anomaly segmentation is a special region of our interest where safety comes to the first priority. Take autonomous driving in urban scenes for an example, the goal of anomaly segmentation is to distinguish the so called *out-of-distribution*(OOD) targets which never appear on the training dataset. Modern systematic datasets [4] have detailed categorization of objects among various types, yet models overfitting on these datasets will try to classify unseen objects to existing categories, causing fatal catastrophe under certain situations.

Many approaches have been proposed to solve the anomaly segmentation tasks. These methods have achieved their success proven by prevalent evaluating metrics. Yet the computational overhead, hardware requirement and expenditure cannot be neglected. Also, there is still space of improvement for these methods.

In this project, we aim to improve the performance and lower the computational costs of modern anomaly segmentation models from various perspectives. We implemeted data augmentation by introducing a self-built dataset(COCO-DAG) from the prevalent COCO and cityscape datasets and conducted experiments on our dataset, exploring the possibility of increasing computational efficiency during the training process. We further create algorithms focusing on the intrinsic characteristics of the OOD pixels, or objects and evaluate their ability of improving the performance of developed pipeline of modules for anomaly segmentation tasks. Besides, based on SML, dilation ,and erosion, We propose a method that needs neither OOD data nor extra networks but perform well. Our exploration and methods show that datasets and algorithms are both significant for tackling current computer vision tasks, either for the sake of training costs or with the purpose of improving performance.

2. Related Works

Previous methods have achieved success on the anomaly segmentation task. The techniques include but not limited to applying various characteristic functions to extract features from different kinds of objects, i.e. the OOD and ID ones; usage of generative models featuring GANs [1, 6, 12]

and autoencoders and pixel-wise anomaly detection framework. In this section, we will summary the papers we have carefully studied or roughly skimmed, providing inspiration for our whole project.

The modern baseline of detecting OOD examples dates back to the method proposed by Dan Hendrycks [8]. They considered two related problems of detecting if an instance is misclassified or out-of-distribution. They provided the baseline by using the traditional softmax probabilities and their method applies to various regions including but not limited to computer vision, natural language processing and automatic speech recognition. Their method, tackling the problem when the training andtest distributions differ, may not work well for recent pixel-level OOD detection tasks. Yet their overall methodology provides a baseline which can be further developed and improved and fits to many tasks.

The usage of generative models such as the deep convolutional generative adversarial network (DCGAN) to image generation has already been introduced to the tasks of semantic segmentation and out-of-distribution detection. David Haldimann [7] proposed a method which can leverage generative models to detect wrongly segmented instances. They first generate an RGB image, then learn a dissimilarity metric that compares the generated image with the original input and detects inconsistencies introduced by the semantic segmentation. Their method requires a GAN based dissimilarity detector and works pretty well on existing datasets.

Finding that autoencoders are prone to simply generate a lower-quality version of input images, Krzysztof Lis [11] introduce a drastically novel approach by reformulating the problem of segmenting unknown classes as one of identifying the differences between the original input image and the one resynthesized from the predicted semantic map. Their method, also requiring the usage of generative networks [13], introduces a new discrepancy network to the fully pipelined architecture and shows that the unexpected objectes can be more reliably detected than the uncertainty- and autoencoder-based methods.

More recent methods have focused on the implementation of pixel-wise anomaly detection framework. Giancarlo Di Biase and Hermann Blum's method [5] uses uncertainty maps to improve over existing re-synthesis methods in finding dissimilarities between the input and generated images and works as a general framework around already trained segmentation networks, which ensures anomaly detection without compromising segmentation accuracy. Moreover, their method is widely appliable because it does not put any constraint on the segmentation network, and therefore can be used with any already trained state-of-the-art segmentation model.

Standardized max logits are proposed by Sanghun Jung and Jungsoo Lee [9] to align the different distributions and reflect the relative meanings of max logits within each predicted class. They aim to alleviate the performance degradation of max logits as their distribution of each predicted class is significantly different from each other. By aligning the ranges of max logits in each predicted class via standardization, they improved the performance of detecting anomalous objects. Their simple yet effective method achieved an improved performance on many benchmark such as Fishyscapes Lost and Found.

Entropy Maximization and Meta Classification [3] proposed by Robin Chan is the core method upon which we build and improve our methods. They utilized samples from the COCO dataset as OOD sample and introduced another training objective to maximize the softmax entropy on these proxies. The idea is to start from pretrained semantic segmentation networksand to retrain a number of DNNs on different in-distribution dataset. They also applied meta classification to introduce a transparent post-processing step to discard false positive OOD samples. Their method significantly improves the detection efficiency via softmax entropy thresholding, leading to better performance compared to existing methods.

3. Methodology

Our project is built upon the framework of Entropy Maximization and Meta Classification(MaxEnt) [3], and the one of Standardized Max Logits(SML) [9]. In this section, we will describe in detail how we understand and improve these algorithms. We will first illustrate how the data augmentation is implemented and how we built our own training dataset, which is crucial to improving the compute efficiency. We then further propose an algorithm for OOD detection with moment configuration. Finally we will show our refinement of SML methods.

3.1. Data Augmentation(COCO-DAG)

The training data of the original algorithm proposed by MaxEnt [3] consists of two parts, one is simply cityscape dataset, another is COCO images. There are two main flaws in this training data, firstly, cityscape pictures are $1024*2048$, but we only need $480*480$ of them(by cropping from a full cityscape image), so we will waste a lot of GPU memory because we must load the whole picture into GPU and then start transforming the picture. Secondly, there are no image contain both OOD pixels and in-distribution(ID) pixels, but the task is to detect OOD pixels in one single image which contains both OOD and ID pixels.So we provide a new way of mixing OOD pixels and ID pixels by synthesizing the images from COCO and cityscape datasets together. Our method is able to make training pictures significantlt smaller and make them contain both OOD and ID pixels in one single image. During training, we just mix

these augmented data with original data to better train the model.

Our data augmentation can be described by three stages. First we extract polygon targets from the COCO images to serve as the OOD proxy for our dataset 1, assuming that these randomly selected and extracted objects form a well distribution of OOD objects in actual city scenes.

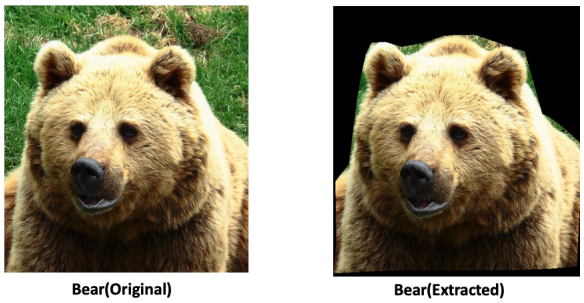

Figure 1. Extract polygon object from COCO dataset

We then synthesize these images with original images from the cityscape dataset 2 and replace them with the images used in the training dataset. Here the aforementioned performance degradation appears because under the standard training procedure, the original method would randomly crop the images into $480 * 480$ ones to resemble the sampling for OOD objects. This alleviates the performance of our data augmentation because the cropping region does not necessarily intersect the region where we added COCO targets.

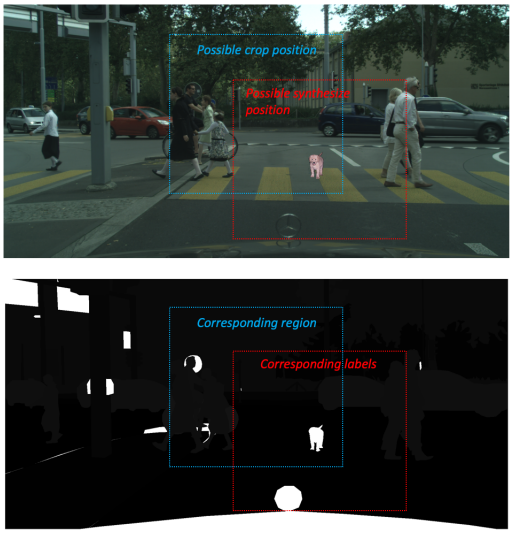

Figure 2. Synthesize cityscape images with COCO objects

So we improved our method of data augmentation by introducing an algorithm that guarantees the sampling of OOD targets. After the synthesization, we would locate the center of COCO targets 3, and split the image into four parts 4. From this approach, we will be able to generate a new dataset that guarantees the co-existence of OOD and ID targets in the same image, and the location of the OOD target in the image is evenly distributed, namely, four directions from the center of the COCO target. We then use this self-built dataset to train the models.

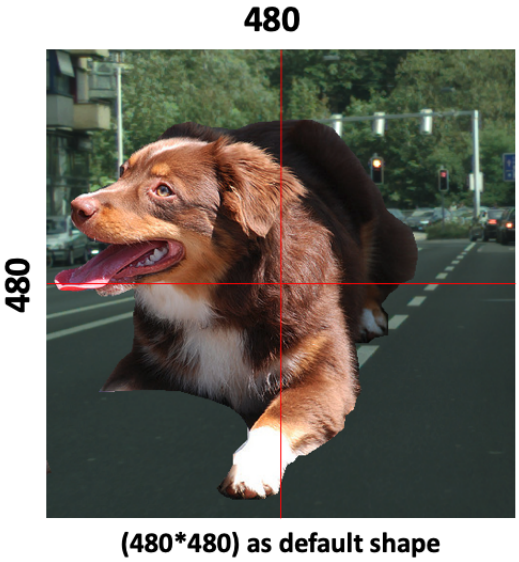

Figure 3. Split the image to get better features

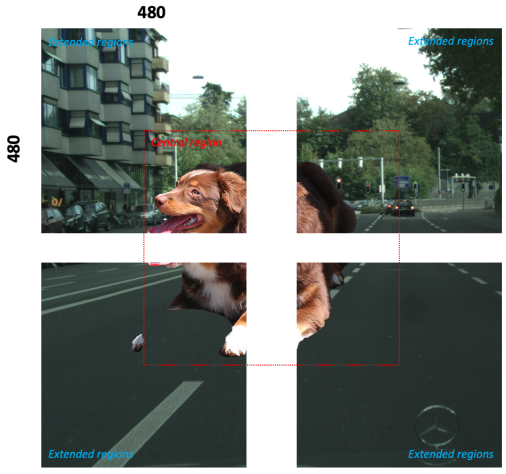

Figure 4. Images synthesized after split

3.2. Towards the intrinsic characteristics of OOD pixels with moment

3.2.1 Motivation and Introduction

We have tried many ways to distinguish OOD pixels from ID pixels, one is based on the assumption that neural net-

work suffers more from adversarial attack when attack is performed on the information that neural network truly learnt, but suffers less when attack is performed on the information that it didn't truly learn. So we performed FGSM on both OOD pixels and ID pixels and saw how their max logit of different classes would change, the expected results are that max logit of ID pixels drastically change while max logit of OOD pixels change a little bit so we can distinguish them from the amount of change.

Unfortunately, we forgot to turn the mode of model into eval mode when performing adversarial attack, so we can't distinguish OOD and ID pixels because the model is non-deterministic, even we forward identical image two times, without any attack, many pixels' max logits will change quite a bit, no matter it's OOD or ID.

At that time, we also found that this uncertainty of train mode model gives me another view of the task, from the following picture 5 we can clearly see that there's a Benz trademark lower in the picture, this heatmap represents the max logits' change when forward identical image two times, although there are still many ID pixels whose change is quite large, some absolutely OOD pixels(like Benz trademark) have significantly larger change in max logit, so why don't we use this *uncertainty* as another dimension of classifying OOD and ID pixels?

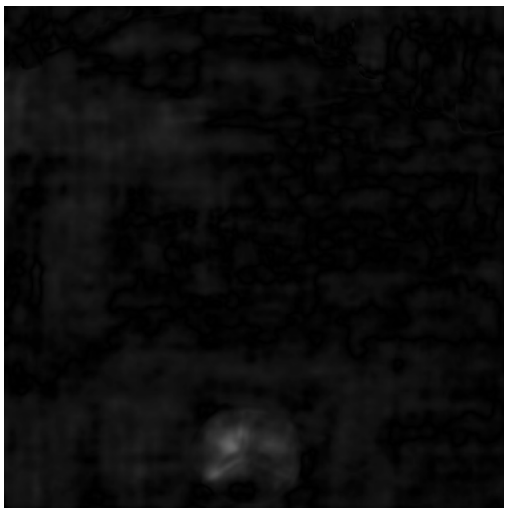

Figure 5. Heatmap of certain image

Besides that, can we take a step more forward and find more dimension or information to classify OOD and ID pixels? We think that for the purpose of finding more information for classifying, we should view *logit* differently.

Let's take the assumption that the output of each class's logit is independent, this is not a strong assumption because different logit is generated by different neurons, and for the sake of brief all the 'model' that we reference below is non-deterministic model like dropout model, then what is forward one time truly does? For each pixel in the picture, it's logits' value can be viewed as samples from some unknown distribution $[X_1, X_2, \cdots, X_c]$, which means each bit of logits simply represent a random variable X_c, it has its mean, variance, entropy and other statistic information, forward a image for one time represents sampling one time from these random variables, the variables' randomness are fully determined by the dropout models' structure when forward.

Let's take a 3 class task for example, for a specific pixel, it's logits could be like $[1.9, 3.8, 8.8]$, and could also be like $[2.2, 4.0, 9.1]$ and so on, most of the traditional classify algorithms just use one sample of them to determine which class this pixel belongs to, they would probably use softmax to get the highest probability and take this pixel as the highest-probability class, or they would calculate the entropy of softmax probabilities and use it as a criterion of determining whether a pixel is OOD or not, but do these traditional algorithms capture all the information in the logits that model output? we believe not, and what we are trying to do is fully utilizing the information underneath the logits.

3.2.2 Our insights

Threshold-based methods are popular in OOD detection area, not only because of its efficiency compared with generation-based methods, but also because they can be better explained through networks' characteristics, the aforementioned MaxEnt [3] and SML [9] papers all applied these methods, using some information underneath the output logits, like how large the max logit is or how high the entropy is. What if we view their methods at that certain angle we referenced before, the *random variable* view of logits?

The answer is clear, if we consider the same example, the 3 class task, we also forward 2 times and get $[1.9, 3.8, 8.8]$ and $[2.2, 4.0, 9.1]$, then SML just combine them in some way (because SML only use inference mode model and inference model can be seen as some kind of combinatory of dropout model) and get the max logit, if we assume that inference model only take mean of two times' results, then SML just take 8.95 out of $[2.05, 3.9, 8.95]$, this is just the first order information of these two samples: the mean information, then what about higher-order information of these random variables? SML just discards them. This disaster also happens in SML [9], they just use the entropy of logits, which is another aspect of these random variables, and discard all other information.

Questions like *how can you be so sure that the combined information of dropout models are more useful than one single inference model* might be raised and our reasonings are as follows.

First of all, inference model never shows up during training, which means there is actually no guarantee for inference model's performance, we are always minimizing the

loss between target value and what dropout models output and thus their performance are guaranteed, the reason why inference model performs better than dropout model is that inference model inherit most of the structure of the dropout models and can be seen as some kind of ensemble of all the dropout models. So although single dropout model performs worse than inference model, if we use all the information of all the dropout models, we are likely to get the same amount (or larger amount) of information that inference model could get. Besides that, although inference model may get better combination of dropout models' output logits, it's still using just first-order information in SML and just entropy in MaxEnt. It's pretty clear that if one could classify OOD and ID pixels in one dimension, then one could classify them in higher-order space including that dimension. What we are trying to do is using all those variables information to add more dimensions to the classification task and make them more separable. So how exactly we do to accomplish that goal? The answer is moment. In the next section, we will introduce our method based on moment.

3.2.3 OOD Detection with Moment Configuration(OODMC)

In statistic, every random variable have its characteristic function,

$$g_X(t) = E(e^{itX})$$

which can be seen as the Laplace transform of the random variable, with the knowledge of Laplace transform we know that every random variable has its unique characteristic function, so to portray a random variable we just have to calculate its characteristic function. It's easier said than done because it's untrackable to calculate with imagine number when doing deep learning task, but we do have another surrogate function, the moment generating function.

$$M_X(t) = E(e^{tX})$$

Although moment generating function can also be seen as a way of Laplace transform, without the help of imagine number we can't make sure this function always exists, but let me make the assumption that it exists in this task, and with the help of Taylor's series expansion we have

$$M_X(t) = E(e^{tX}) = E(\sum_{k=0}^{\infty} \frac{(tX)^k}{k!}) = \sum_{k=0}^{\infty} \frac{E(X^k)}{k!} t^k$$

which means if we can get all orders' moment of the random variable $E(X^n)$ We can then reconstruct all the information that this random variable holds.

Back to this OOD detection task, the goal is to utilize more information of logits to better distinguish OOD pixels from ID pixels, and we have made it clear that SML and MaxEnt are just utilizing some kind of information, while extra information should come from all the moments of logits. we propose OODMC(OOD Detection with Moment Configuration), which significantly boost the performance. The algorithm is as below.

Algorithm 1 OOD Detection with Moment Configuration: OODMC

Require: any non-deterministic model $\mathcal{M}$, any kind of classifier $\mathcal{C}$

First train classifier:

for I, T in train dataset **do**

Forward 128 times to sample all needed information for each pixel

Calculate entropy

Calculate N order moment of all logits X_c, $moment^n = [m^n_{X_1}, m^n_{X_2}, \cdots, m^n_{X_c}]$ ▷ each order of moment is C dim, here 19

Concatenate [$entropy$, $moment^1$ ..., $moment^n$]

Use all the information to fit classifier $\mathcal{C}$

Predict OOD label

for I, T in test dataset **do**

Calculate all information through the same procedure

Use classifier $\mathcal{C}$ to predict whether a pixel is OOD

There are two key points of this algorithm. Firstly, this is an algorithm only for backend processing, so any pre-trained classification model on cityscape dataset is alright, but since this algorithm's performance depends on the classification model's performance, it's recommended to choose a strong one. Secondly, the idea of using *more information* can be found throughout the whole algorithm, If we choose entropy as the only information representation and set the classifier as a threshold classifier, we get back to the ordinary MaxEnt algorithm, if we choose max logit as the only information representation and also set the classifier as a threshold classifier, we get back to the ordinary SML. There is no guarantee that how more separable can those pixels be when adding these dimensions generated with each order's moment, but it's always strictly more separable in higher dimension space, so we can use these information with relief.

3.3. Improving Performance Based on SML

3.3.1 Our Inspiration

In most related works, researchers tend to solve OOD identification problems through approaches based on deep learning. They usually extract features from images and fuse them, train extra deep neuron networks, and expect the model can tell anomaly objects. Of all the methods for iden-

tifying unexpected road obstacles, Standardized Max Logits [9](SML) proposed by Jung et al. is very impressive due to its simplicity, effectiveness, and efficiency. SML doesn't use any additional network and training. No out-of-distribution data is needed either. However, it reaches a pretty good effect that exceeds our expectations. Compared to other methods, SML uses prior knowledge of the segmentation model and the images more directly to solve problems. For example, it suppresses objects' boundaries to avoid high anomaly scores caused by abrupt changes in the boundaries. And this makes us pay more attention to the structure of images, where our inspiration comes in.

3.3.2 Irrelevant Regions Elimination with Variation of Dilation and Erosion(VDE)

SML [9] is a simple yet effective approach to identifying unexpected road obstacles. It nails two key points. Firstly, choosing softmax function as the final layer results in the so-called "overconfidence" of the model. Secondly, the segmentation model's output max logits of different categories are in different ranges. It also tries two ways to reduce the interference of images on the model's judgment: Boundary Suppress and Gaussian Smooth, which we are interested in.

As figure6 shows, the two ways reduce the number of pixels of misjudgment. But there are still many wrongly judged pixels left. Most of them are in some "irrelevant" regions, such as the air or streets inside. In figure7, we can see that, with a low threshold applied, many in-distribution pixels will be misjudged because the model is too suspicious, while with a high threshold, the model losses the ability to tell OOD pixels. Obstacles in these irrelevant regions seriously interfere with the model but have a low probability of threatening driving. Ignoring them sometimes helps the model has a larger available threshold range to tell real anomaly items from in-distribution objects. If we can focus on the area of roads, the performance of the method will be better.

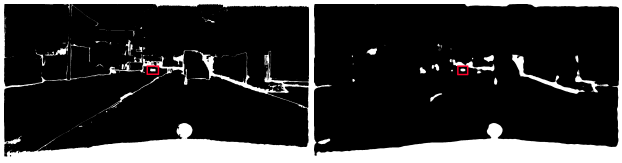

Figure 6. White pixels are those that be judged to be unexpected ones. The left doesn't apply methods Boundary Suppress and Gaussian Smooth while the right applies them. Overall, the right achieves better results. The object in the red box is an OOD obstacle.

So how can we do that? Figure8 tells us the answer. The standardized max logits values of roads and unexpected pixels are sharply divided, which means we can leverage part of the output to improve itself. The process of VDE is: Firstly, select pixels that are classified as roads. Secondly, set a high threshold to obtain in-distribution pixels and eliminate incorrectly classified pixels. Thirdly, to fill in discontinuous pixel parts(usually OOD pixels) surrounded by road pixels, do a dilation operation. Then, do an erosion operation in order to restore the unnecessary expansion. Finally, according to processed results, raise or lower the confidence of the semantic segmentation model to some pixels. The process is shown in figure9.

Different from normal dilation and erosion, VDE adjusts the algorithms respectively to fit this task. For dilation, after each value of a pixel updating, VDE synchronizes the update to the original image, which can make sure to fully fill the holes surrounded by road pixels with a small size of the dilation kernel. For erosion, in each row, VDE does an erosion operation until a real road pixel is eroded, then updates the original image and continues to the next row directly. For each row, the erosion operator will be executed twice, from left to right and from right to left respectively. Considering the usual location and angle of the camera placed, the dilation and erosion operation will be executed from upper rows to lower rows.

Note that VDE needs neither OOD data nor additional training, which is the same as SML. We keep the improving method light and simple on purpose.

4. Experiments

4.1. Experiment Configuration

We built our experiment recipes based on the scripts provided by MaxEnt [3]. Besides the given parameters from the original paper, we introduce our own parameters for training.

To choose the ratio of augmented data mixed with origin data, use parameter `embedding_img_interval`, which indicates the frequency of origin data replaced by augmented data. The parameter `optim_target` can be chosen between *entropy* and *logit* to select the front-end model of SML algorithm.

As for the parameters of OODMC, `moment_num` indicates the number of samples used to calculate statistic information of logits, `svm_points_num` indicates the number of total pixels used to train the classifier(default as SVM), `moment_order` indicates the number of orders used to calculate logits' moment, each order's moment is an array of `class_num` bits,`moment_weight` is the coefficient that measures how much OODMC relys on entropy since OODMC also uses it.

Finally, `SVM_eval_subsize` is introduced to depict how many pictures the model will analyze and we recommend that this parameter be modified depending on the actual GPU capability of user.

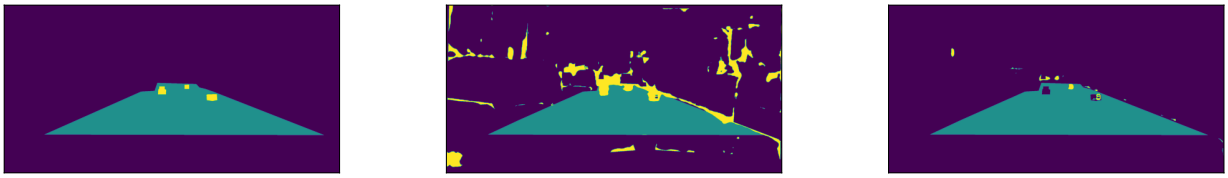

Figure 7. Yellow pixels are those who be judged to be OOD ones. Green pixels are the main part of the road as a reference. The left is the ground truth. The middle is with a low threshold applied while the right is with a high one.

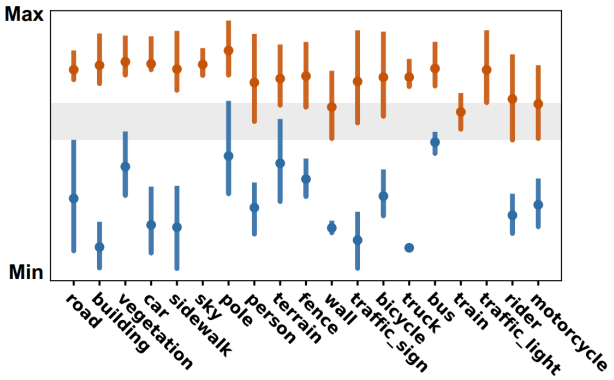

Figure 8. X-axis denotes the classes which are sorted by the occurrences of pixels in the training phase. Y-axis denotes the SML values. Red and blue represent the distributions of values in in-distribution pixels and unexpected pixels, respectively. This figure is from [9].

Other configuration are inherited from the MaxEnt [3] paper and all these configurations can by found in our codes.

4.2. Overall results

For the limitation of semester time and computing resources, we were only able to conduct some experiments. Below are selected experiment results. Our expriment results show that COCO data augmentation(COCO-DAG) does improve the performance of existing OOD detection models by increasing F1 scores, precision rate, area under ROC and PRC curves and by decreasing false position rate and FPR95. To sum up, DAG surpasses non-DAG methods, entropy surpasses logit.

Method	Entropy without DAG	Entropy with DAG
F1 Score ↑	0.9831	**0.9893**
Precision ↑	0.9922	**0.9954**
FP Rate ↓	0.1651	**0.0970**

Table 1. Pixel level evaluation results on LostandFound

Method	Logit without DAG	Entropy without DAG
AUROC ↑	0.5395	0.9371
FPR95 ↓	1.0	0.4279
AUPRC ↑	0.0114	0.5729
Method	Logit with DAG	Entropy with DAG
AUROC ↑	**0.6070**	**0.9464**
FPR95 ↓	1.0	**0.3066**
AUPRC ↑	**0.0166**	**0.5940**

Table 2. Stage evaluation results on LostandFound

The results are pretty good but not surprising. We only analyze at pixel level because of the following two reasons: Firstly, segment level evaluation is not a trending metric, which is another aspect of OOD detection task. Secondly, our algorithm OODMC does not do anything related to segment classification, it will probably harm the original algorithm designed for segment level classification in MaxEnt's paper [3].

For VDE, we compare it to SML [9] in Fishyscapes [2] validation sets. For a fair comparison, we reproduce SML in our experimental environment. And we also list the performance reported in the paper.

Method	AP↑	AUROC↑	FPR @TPR95↓
SML (reported)	**36.55**	96.88	14.53
SML (reproduced)	24.25	96.35	17.17
VDE	33.43	**97.92**	**10.25**

Table 3. Comparison of VDE and SML

In our environment, VDE archives better results. More-

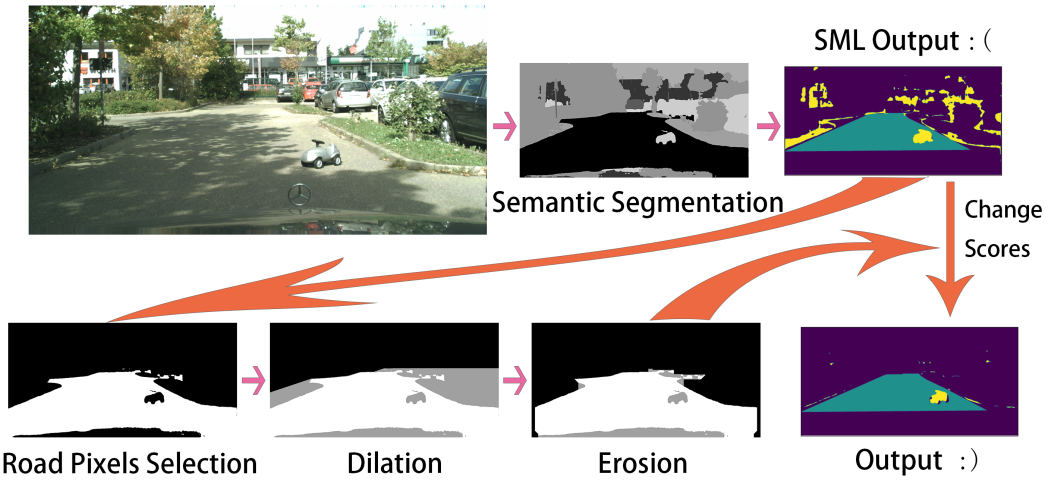

Figure 9. The process of VDE. By taking road areas as the main concern, the model has a larger available range of output values to distinguish OOD pixels from ID ones. Then the selection of the threshold for them becomes more reasonable and the performance of the method gets better naturally.

over, its AUROC and FPR@TPR95 are better than SML's reported in the paper, reaching a pretty good effect.

5. Conclusion

We have introduced data augmentation from COCO dataset mixed with cityscape images(COCO-DAG), a data augmentation which can be used to improve the computing efficiency of existing methods. We also propose the OOD detection with moment configuration(OODMC), an algorithm which can enhance the performance of anomaly segmentation models using the intrinsic characteristics of OOD pixels, and variation of dilation and erosion(VDE), an approach that doesn't need OOD data and additional training but has great performance.

Our insights from careful analysis of existing algorithms and concrete experiments are listed as follows:

- The structure, quality and extensibility of datasets are crucial to modern sophisticated computer vision tasks. Proper data augmentation and efficient training recipes are important to improve the performance of any DNNs. Our COCO-DAG dataset built from carefully designed algorithm is able to improve the computing efficiency of existing anomaly segmentation models.
- Our OODMC algorithm provides a method for backend processing, which can be applied to any pretrained classification model on cityscape dataset. In particular, our OODMC algorithm works better with a stronger classification models.
- In order to distinguish the OOD pixels from the ID ones, intrinsic characteristics must be exploited to fully get the information representation of the pixels. By treating OOD pixels as random variables, we are able to use moment to extract their intrinsic characteristics and hence improve the performance of existing anomaly segmentation models.
- By analyzing output values of the semantic segmentation model and the prior knowledge of the location where OOD obstacles appear with different probability, we refine SML to VDE, achieving great performance and keep it light, which means a complete lack of extra data and training.

To sum up, we have treated the anomaly segmentation task from various perspectives, including direct data augmentation methods, algorithms aiming at extracting intrinsic features of the OOD pixels, and attention to the structure of images. We believe that our datasets and algorithms can be widely applied to current well-performing anomaly segmentation models and improve their performance and computing efficiency altogether.

References

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 214–223. PMLR, 06–11 Aug 2017. 1

[2] Hermann Blum, Paul-Edouard Sarlin, Juan Nieto, Roland Siegwart, and Cesar Cadena. Fishyscapes: A benchmark for safe semantic segmentation in autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019. 7

[3] Robin Chan, Matthias Rottmann, and Hanno Gottschalk. Entropy maximization and meta classification for out-of-distribution detection in semantic segmentation. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5108–5117, 2021. 2, 4, 6, 7

[4] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 1

[5] Giancarlo Di Biase, Hermann Blum, Roland Siegwart, and Cesar Cadena. Pixel-wise anomaly detection in complex driving scenes. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16913–16922, 2021. 2

[6] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 1

[7] David Haldimann, Hermann Blum, Roland Siegwart, and Cesar Cadena. This is not what I imagined: Error detection for semantic segmentation through visual dissimilarity. *CoRR*, abs/1909.00676, 2019. 2

[8] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. 2

[9] Sanghun Jung, Jungsoo Lee, Daehoon Gwak, Sungha Choi, and Jaegul Choo. Standardized max logits: A simple yet effective approach for identifying unexpected road obstacles in urban-scene segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15425–15434, 2021. 2, 4, 6, 7

[10] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014. 1

[11] Krzysztof Lis, Krishna K. Nakka, Pascal Fua, and Mathieu Salzmann. Detecting the unexpected via image resynthesis. *CoRR*, abs/1904.07595, 2019. 2

[12] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. 1

[13] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. *CoRR*, abs/1711.11585, 2017. 2