ObjectAug: Object-level Data Augmentation for Semantic Image Segmentation

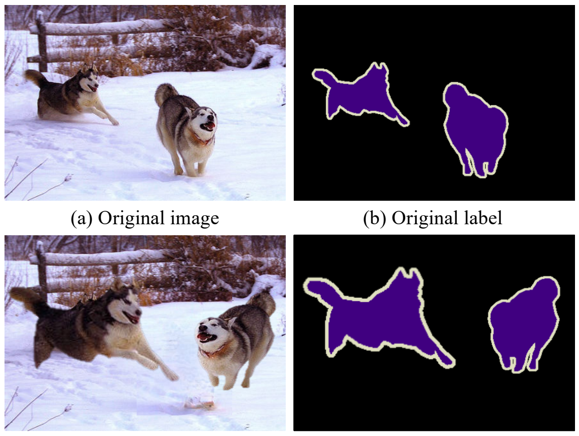
Jiawei Zhang*†, Yanchun Zhang†‡, Xiaowei Xu§
*Shanghai key Laboratory of Data Science, School of Computer Science, Fudan University, Shanghai, China
†Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou, China
‡College of Engineering and Science, Victoria University, Melbourne, Australia
§ Guangdong Cardiovascular Institute, Guangdong Provincial People's Hospital, Guangzhou, China
Email: 17110240008@fudan.edu.cn, yanchun.zhang@vu.edu.au, xiao.wei.xu@foxmail.com

***Abstract*—Effective training of deep neural networks (DNNs) usually requires labeling a large dataset, which is time and labor intensive. Recently, various data augmentation strategies like regional dropout and mix strategies have been proposed, which are effective as the augmented dataset can guide the model to attend on less discriminative parts. However, these strategies operate only at the image level, where the objects and the background are coupled. Thus, the boundaries are not well augmented due to the fixed semantic scenario. In this paper, we propose ObjectAug to perform object-level augmentation for semantic image segmentation. Our method first decouples the image into individual objects and the background using semantic labels. Second, each object is augmented individually with commonly used augmentation methods (e.g., scaling, shifting, and rotation). Third, the pixel artifacts brought by object augmentation are further restored using image inpainting. Finally, the augmented objects and background are assembled as an augmented image. In this way, the boundaries can be fully explored in the various semantic scenarios. In addition, ObjectAug can support category-aware augmentation that gives various possibilities to objects in each category, and can be easily combined with existing image-level augmentation methods to further boost the performance. Comprehensive experiments are conducted on both natural image and medical image datasets. Experiment results demonstrate that our ObjectAug can effectively improve segmentation performance.**

***Index Terms*—Object-level, Data Augmentation, Image Segmentation**

I. Introduction

Semantic segmentation with the goal to assign semantic labels to target objects in an image is one of the fundamental topics in computer vision [1], [2]. Recently, deep neural networks (DNNs) have been widely adopted in semantic segmentation with boosted performance [3]–[12]. Typical DNN requires a large amount of labeled data for training [13], [14], which is a time and labour intensive process. To get around this, machine learning practitioners typically either transfer knowledge from other tasks (i.e. transfer learning [15]) or generate synthetic data based on labeled data (i.e. data augmentation). Various classical data augmentation strategies, including random rotation, random scaling and random cropping [7], [16] artificially expanded the size of datasets by orders of magnitude, thus with improved performance in most vision tasks.



(a) Original image (b) Original label

(c) Augmented image (d) Augmented label

Fig. 1. Illustration of ObjectAug. ObjectAug operates at the object level, and can perform various augmentation methods for each object as shown in (c) and (d). The left husky is scaled and shifted, while the right one is flipped and shifted. Thus, the boundaries are extensively augmented to boost the performance of semantic segmentation.

Recently, several works [17]–[24] proposed to mix data at the image level (a.k.a mixed images) to augment the data which can guide the model to attend on less discriminative parts. For example, CutOut [17] and random erase [20] randomly masked out square regions of the input during training, which improves the robustness and overall performance of convolutional neural networks. CutMix [19] generated a new training sample by randomly combining two cropped training samples, and achieves a better performance than CutOut. However, both methods operate at the image level, where the objects and the background are coupled in the image. This means that the boundaries between objects and the background are fixed, which are important to obtain precise segmentation [25], [26]. In this way, the critical boundaries are not augmented well. It occurs to us: is it possible to effectively augment the boundaries in semantic segmentation to further boost the performance?

In this paper, we propose an object-level augmentation method, ObjectAug, to address the above question. As shown in Figure 1, instead of dealing with the image directly, ObjectAug utilizes the segmentation annotations to decouple the

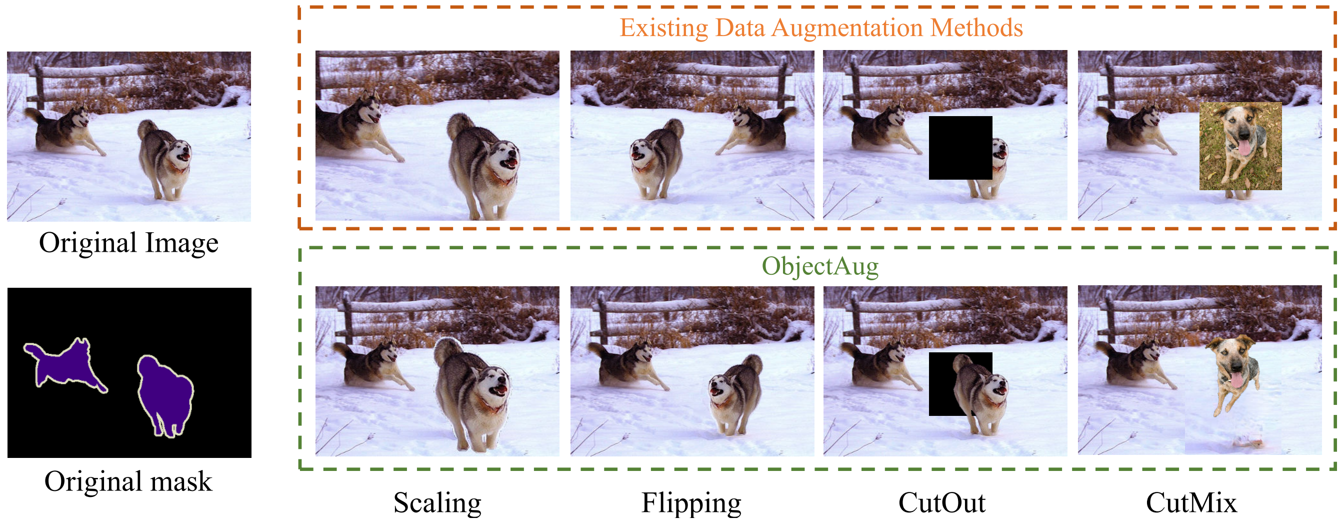


Fig. 2. Comparison of ObjectAug and existing data augmentation methods. All the existing methods operate at the image level, while ObjectAug at the object level. ObjectAug can adopt existing methods to augment each object in the image.

image into the background and objects. The object-removed background can be further restored using image inpainting, while the objects can be augmented independently. The annotations are also processed in a similar way as that of their corresponding images. Finally, the augmented objects and the background are assembled as an augmented image in which objects can be placed flexibly on the background. During the augmentation, ObjectAug usually misaligns areas of the original object and the enhanced object, which would cause pixel artifacts (the area that belongs to the original object and not to the enhanced object). Since the augmentations for each object are performed independently, category-aware augmentation can be easily achieved, e.g., objects in some categories are augmented in every augmented image, while others remain (unaugmented) in some augmented images. On the other hand, our ObjectAug breaks the consistency among the objects and diversifies their compositions, the robustness is therefore further improved. The performance can be further boosted by combining the existing augmentation methods since they are on image level, which works at a higher level than that of our objectAug (object level). We present extensive evaluations of ObjectAug on various DNN architectures and semantic image segmentation datasets. Experiments across various architectures and datasets demonstrate that our ObjectAug can effectively improve the segmentation performance.

In summary, our contributions follow,

- We propose an object-level data augmentation method, ObjectAug, which can be easily combined with other existing image-level augmentation methods to further boost performance.
- ObjectAug can perform category-aware augmentation to mitigate the category imbalance problems, which performs diversified augmentation methods to objects belonging to different categories.
- Extensive experiments across various networks and datasets demonstrate that ObjectAug outperforms existing data augmentation methods and significantly improves the performance of segmentation.

II. Related work

Existing augmentation methods can be divided into two general approaches: traditional augmentation methods, and DNN-orientated augmentation methods.

The traditional augmentation approach has been widely used including random crops, horizontal flipping, and color augmentation [27], which could improve the robustness of translated, reflected, and illuminated objects, respectively. Random scaling [7] as well as random rotations and affine transformations are also widely adopted. Geometric distortions or deformations are commonly used to increase the number of samples for training DNNs to balance datasets. The above methods have proven to be fast, reproducible, and reliable, and its implementation code is relatively easy and available for the most known deep learning frameworks, which makes it even more widespread [28].

The DNN-orientated augmentation approach is an emerging trend, which takes the learning characteristics of DNNs into consideration. Mixup [18] uses information from two images. Rather than implanting one portion of an image inside another, Mixup produces an element-wise convex combination of two images. An adaptive mixing policy [29] was proposed to improve Mixup preventing manifold intrusion. CutOut [17] randomly masks out square regions of the input during training, which improves the robustness and overall performance. Rather than occluding a portion of an image, CutMix [19] generates a new training sample by randomly combining two cropped training samples. Moreover, Generative Adversarial Network (GAN) [30] is getting popular in data augmentation domain. For example, [31] employed a GAN to generate realistic images by reconstructing the image, while our method aims on augmenting the true objects in the image rather than

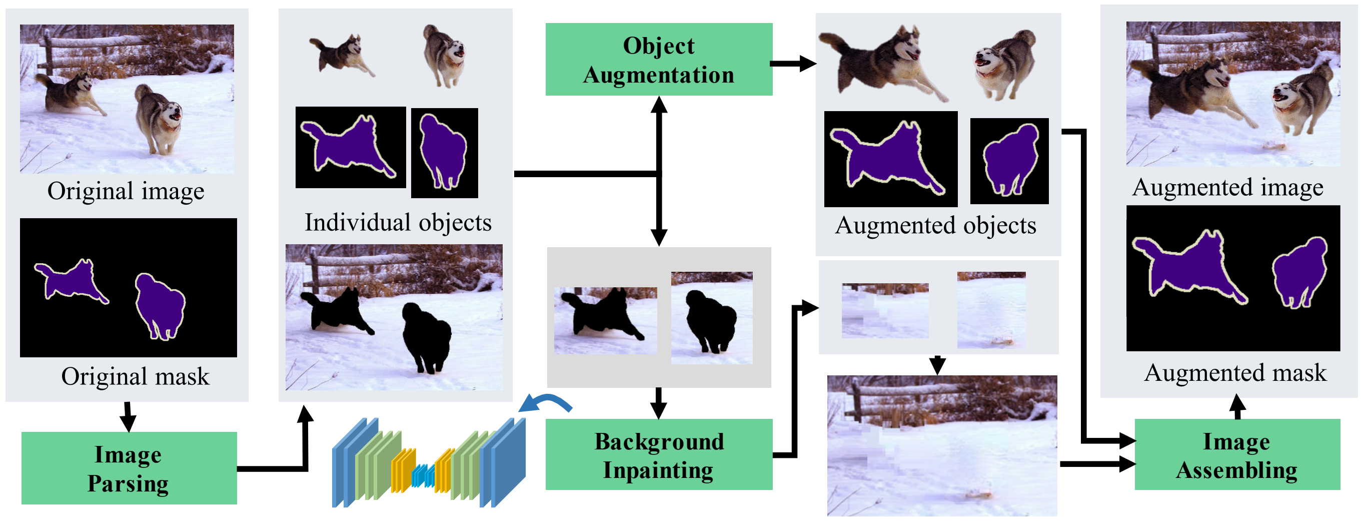


Fig. 3. Overall flow of ObjectAug. ObjectAug includes four modules: image parsing, object augmentation, background inpainting, and image assembling. First, image parsing decouples the image and extracts multiple objects in the image. Next, object augmentation adopts various data augmentation methods to augment the objects. At the same time, background inpainting restores the object-removed background to deal with the pixel artifacts. Finally, the augmented objects with the restored background and their masks are assembled to obtain the augmented image and mask.

generating the new objects. The methods are out of the scope of this paper.

In this paper, we compared ObjectAug with CutOut and CutMix, which are the most popular approaches in DNN-oriented augmentation method. Note that CutOut and CutMix are primarily for classification and detection tasks, which can be easily revised for segmentation tasks (detailed in the experiment section). Illustration of comparison between ObjectAug, traditional data augmentation methods (scaling and flipping), CutOut and CutMix are shown in Figure 2.

III. Method

A. Overview

The overall flow of ObjectAug is shown in Figure 3, and Algorithm 1 details the process of ObjectAug. ObjectAug includes four modules: image parsing, object augmentation, background inpainting, and image assembling. First, image parsing decouples the image and extracts the multiple objects in the image. Next, the object augmentation performs various data augmentation methods to augments the objects. At the same time, the background inpainting restores the object-removed background to deal with the pixel artifacts. Finally, the augmented objects, the corresponding masks, and the restored background are assembled by the image assembling module as the augmented image and mask. We detail the four modules of ObjectAug as follows.

B. Image Parsing

Image parsing decouples the image and extracts the objects from the image. For ease of discussion, a training image and its mask are denoted as $\mathbf{I}$ and $\mathbf{M}$, respectively. By utilizing the segmentation annotation, we can easily decouple the mask $\mathbf{M}$ into $\mathbf{M}^k \in \{0,1\}^{W\times H}$, which is the mask of the k-th object. It associates binary values in the mask to pixels in the image so that $\mathbf{M}^k_{x,y} = 1$ if the pixel at (x, y) belongs to the k-th object. Note that since one pixel can corresponds to only one object, the masks are constrained by Eq. (1). $\mathbf{1}$ presents the all-ones matrix.

$$\sum_{k=1}^{n} \mathbf{M}^k = \mathbf{1}, \quad \mathbf{M}^n = \mathbf{1} - \sum_{k=1}^{n-1} \mathbf{M}^k. \tag{1}$$

The background masks $\mathbf{M}_n$ can be easily retrieved from the object masks. The process of extracting the k-th object is denoted as $\mathbf{I}^k \in \mathbb{R}^{W\times H\times C}$. The decouple of the objects in an image is computed as follows,

$$\mathbf{I}^k = \mathbf{M}^k \odot \mathbf{I}, \quad \sum_{k=1}^{n} \mathbf{I}^k = \mathbf{I}. \tag{2}$$

where $\odot$ is dot product. Meanwhile, if the traditional data augmentation methods such as rotation or flipping are used directly to the object with its size unchanged, its position may change considerably, which may result in pixel artifacts in the image. In order to minimize the pixel artifacts brought by ObjectAug, we crop each object $\mathbf{I}^k$ and its corresponding ground truth $\mathbf{M}^k$, and obtain the cropped object patch $\mathbf{I}^k_c$ and the ground truth $\mathbf{M}^k_c$. ψ denotes the crop parameter, which takes the center of the object as the cropping center, and the crop size is based on the size of the object.

$$\mathbf{I}^k_c, \mathbf{M}^k_c = f_c(\mathbf{I}^k, \mathbf{M}^k|\psi). \tag{3}$$

C. Object Augmentation

The object augmentation module augments the extracted objects from the image parsing module individually. Given a series of traditional data augmentation methods including scaling, rotation, shifting, flipping, brightening and etc., denoted as $[f_1, f_2, ..., f_m]$, they are performed to each object with a set of corresponding probabilities $\mathbf{P}$=$[p_1, p_2, ..., p_m]$. With the sequentially performing the augmentations, we obtain the composed augmentation function f_{ObjAug}. We then apply f_{ObjAug} to the extracted object and mask patches, and obtain the augmented objects.

$$f_{ObjAug}(\mathbf{I}|\mathbf{P}) = f_1(\mathbf{I}|p_1) \circ f_2(\mathbf{I}|p_2) \circ \ldots \circ f_m(\mathbf{I}|p_m). \quad (4)$$

$$\mathbf{I}^k_{aug}, \mathbf{M}^k_{aug} = f_{ObjAug}(\mathbf{I}^k_c, \mathbf{M}^k_c|\mathbf{P}). \quad (5)$$

As each object can be augmented individually, the object augmentation module supports category-aware augmentation. Note that the category imbalance is a common issue in semantic segmentation, where the categories with the less and difficult objects to segment are more critical in training. Meanwhile, the data augmentation not only brings generalization to the train samples, it also makes the training more difficult. Therefore, an intuitive idea is that the method can be inclined to more rare cases called rarity-driven coefficient so that the model can alleviate under-fitting because of too few examples. However, we actually find that the rarity of objects is not directly proportional to the final segmentation performance. For example, the number of object in category people is far more than that of all other category objects, but its segmentation performance is ranked 13-th. Then we also use the segmentation performance as the criterion of the category-aware coefficient called hard-driven coefficient. We will detail the comparison of two coefficients in ablation analysis. For the hard-driven coefficient α, p_j represents the performance in the previous experiment of the j-th category, and $\hat{p}$ represents the median of the performance. In rarity-driven coefficients β, $\hat{N}$ represents the median of the number of objects and N_j represents the number of objects belonging to the j-th category. α_j/β_j is defined as the hard-driven/rarity-driven coefficient for the j-th category as follows,

$$\alpha_j = \frac{\hat{p}}{p_j}, \beta_j = \frac{\hat{N}}{N_j}, \quad (6)$$

Finally, the probability of the j-th category using the augmentation method m and the hard-driven coefficient α is defined as,

$$p_{j,m} = p_m \times \alpha_j. \quad (7)$$

D. Background Inpainting

The image parsing module brings pixel artifacts in the object-removed background which can not always be covered completely by the augmented object I^k_{aug}. In addition to filling random noise, we use image inpainting methods to fill the pixel artifacts. To prevent the inpainting methods from being affected by objects, we inpaint the background with the object removed. Particularly, we perform morphological processing

Algorithm 1 Process of ObjectAug.

```
Require:
    An original image $\mathbf{I}$ and its annotation $\mathbf{M}$ containing $n$ objects; Selected data augmentation methods $[f_1, f_2, ..., f_m]$ with probabilities $[p_1, p_2, ..., p_m]$, category-aware coefficient $[\alpha_1, \alpha_2, ..., \alpha_J]$, $J$ is the number of class;
Ensure:
    Augmented image $\mathbf{I}$ and its annotation $\mathbf{M}$;
1:  Parse $\sum_{k=1}^{n} \mathbf{I}^k = \mathbf{I}$                                # Parsing Stage
2:  Parse $\sum_{k=1}^{n} \mathbf{M}^k = \mathbf{1}$
3:  $\mathbf{I}^k_c, \mathbf{M}^k_c \leftarrow f_c(\mathbf{I}^k, \mathbf{M}^k, \psi)$
4:  for all objects $I^k$ do                                     # Augmenting Stage
5:      $\mathbf{I}^k_{aug}, \mathbf{M}^k_{aug} \leftarrow f_{ObjectAug}(\mathbf{I}^k_c, \mathbf{M}^k_c, p_{j,m})$
6:      $\mathbf{I}^k_i \leftarrow \Phi(\mathbf{I}_c, \mathbf{M}^k_c)$                        # Inpainting Stage
7:      $\mathbf{I}_c, \mathbf{M}_c \leftarrow f_c(\mathbf{I}, \mathbf{M}, \psi)$                    # Assembling Stage
8:      $\widehat{\mathbf{M}}^k_c \leftarrow \mathbf{M}^k_c \cup \mathbf{M}^k_{aug}$,   $\widehat{\mathbf{M}}^{k'}_c \leftarrow \widehat{\mathbf{M}}_c - \mathbf{M}^k_{aug}$
9:      $\mathbf{I}_{asm} \leftarrow \mathbf{I}_c \odot (\mathbf{1} - \widehat{\mathbf{M}}_c) + \mathbf{I}^k_{aug} + \mathbf{I}^k_i \odot \widehat{\mathbf{M}}''_c$
10:     $\mathbf{M}_{asm} \leftarrow \mathbf{M}_c \odot (\mathbf{1} - \widehat{\mathbf{M}}_c) + \mathbf{M}^k_{aug}$.
```

to expand extracted object masks thus to remove residual edges. We employ the inpainting methods from [32] and denote it as Φ, which uses DNNs with partial convolutions to deal with the irregular holes. The DNN models for image inpainting is trained on ImageNet [33]. The background inpainting can be described as,

$$\mathbf{I}^k_i = \Phi(\mathbf{I}_c, \mathbf{M}^k_c). \quad (8)$$

where $\mathbf{I}^k_i$ is the inpainted result from the original cropped image $\mathbf{I}_c$ with selected object $\mathbf{M}^k_c$ removed. In the inpainting process, the images are all scaled to a uniform size. Meanwhile, since the object is removed and we only inpaint the background, the annotation does not need to be processed accordingly.

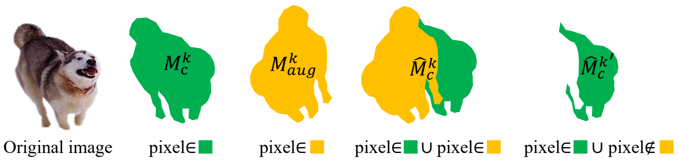


Fig. 4. Visualization of masks $\widehat{\mathbf{M}}^k_c$ and $\widehat{\mathbf{M}}^{k'}_c$ in the image assembling step.

E. Image Assembling

In image assembling, the augmented objects and the restored background are combined into the augmented images and masks. First, we use the identical cropping parameter ψ to directly crop the original image $\mathbf{I}$ and annotation $\mathbf{M}$ to obtain the cropped image $\mathbf{I}_c$ and annotation $\mathbf{M}_c$.

$$\mathbf{I}_c, \mathbf{M}_c = f_c(\mathbf{I}, \mathbf{M}|\psi). \quad (9)$$

Next, the masks of the augmented objects are processed following,

$$\widehat{\mathbf{M}}^k_c = \mathbf{M}^k_c \cup \mathbf{M}^k_{aug}, \quad \widehat{\mathbf{M}}^{k'}_c = \widehat{\mathbf{M}}_c - \mathbf{M}^k_{aug}, \quad (10)$$

where $\widehat{\mathbf{M}}_c^k$ denotes the union of the original annotation and the augmented annotation for the current image patch $\mathbf{I}_c$; $\widehat{\mathbf{M}}_c^{k'}$ is the pixel artifacts introduced by the misaligned original object and augmented object. For image assembling, the corresponding image area of $\widehat{\mathbf{M}}_c^k$ is reset (to be pixel artifacts), while the corresponding image area of $\widehat{\mathbf{M}}_c^{k'}$ is restored using image inpainting.

Finally, we assemble the augmented object I_{aug}^k into the cropped original image as Eq. (11). Meanwhile, the pixel artifacts is filled with the inpainting results $\mathbf{I}_i^k$.

$$\mathbf{I}_{asm}^{'} = \mathbf{I}_c \odot (\mathbf{1} - \widehat{\mathbf{M}}_c^k) + I_{aug}^k + \mathbf{I}_i^k \odot \widehat{\mathbf{M}}_c^{k'}. \quad (11)$$

$$\mathbf{M}_{asm} = \mathbf{M}_c \odot (\mathbf{1} - \widehat{\mathbf{M}}_c^k) + \mathbf{M}_{aug}^k. \quad (12)$$

By assembling the assembled patches $\mathbf{I}_{asm}$ into the image with the relevant area removed, we get the image with the k-th object augmented. Similarly, the corresponding object-level augmented ground truth is obtained accordingly. We then repeat the above process until all the objects are processed, and the augmented image and annotation are obtained.

IV. Experiments

In this section, ObjectAug is extensively evaluated and compared with the existing methods on public datasets. We first describe our experimental setup in detail. Then we compare ObjectAug with CutOut and CutMix, and proceed ablation analysis of hyper-parameters, image inpainting, and category-aware coefficient on the PASCAL VOC 2012 segmentation dataset. At last, we conduct extended experiments on the Cityscapes dataset and CRAG dataset to show the generalizability of our method.

A. Experiment Setup

Datasets Three image segmentation datasets including PASCAL VOC 2012, Cityscapes, and CRAG are used for evaluation.

- **PASCAL VOC 2012.** The PASCAL VOC 2012 semantic segmentation benchmark [34] contains 20 foreground object classes. The original dataset contains 1, 464 (train), 1, 449 (val), and 1, 456 (test) pixel-level annotated images. The dataset is augmented by the extra annotations provided by [35], resulting in 10, 582 (trainaug) training images;
- **Cityscapes.** The Cityscapes dataset [14] is a large-scale dataset containing high-quality pixel-level annotations of 5000 images (2975, 500, and 1525 for training, validation, and test, respectively) and about 20000 coarsely annotated images;
- **CRAG.** The Colorectal Adenocarcinoma Gland (CRAG) dataset [36] has a total of 213 H&E CRA images taken from 38 WSIs. Images are at 20$\times$ magnification and are mostly of size 1512$\times$1516 pixels, with corresponding instance-level ground truth. The CRAG dataset is split into 173 training images and 40 test images with different cancer grades.

Implementation Details All experiments were implemented and evaluated on PyTorch platform with Titan X (Pascal) with 12 GB memory. For the PASCAL VOC 2012 and cityscapes datasets, we followed the same training settings as in [11]. In short, we employed the same learning rate schedule (i.e., "poly" policy [37] and same initial learning rate 0.007), cropped size 513 $\times$ 513, fine-tuning batch normalization parameters with output stride 16, and random scale data augmentation during training. Besides, we also used some traditional data augmentation methods including random scale, random shift, random rotation, and random horizontal flip to augment objects in ObjectAug. The specific setting of ObjectAug details in the result and discussion section. The performance was measured in terms of pixel intersection-over-union averaged across all classes (mIOU). For image inpainting, we utilized ResNet50 based U-Net as our model and the loss function from [32]. We implemented the model with the pre-trained model on ImageNet [13] and finetuned it on current datasets to enhance the inpainting performance.

TABLE I
Performance comparison of models with and without ObjectAug on PASCAL VOC 2012.

Method	Model	ObjectAug	mIoU (%)
DeepLab V3	MobileNet	×	70.1
		✓	**71.9**
	ResNet-50	×	76.9
		✓	**77.8**
	ResNet-101	×	77.3
		✓	**78.4**
DeepLab V3plus	MobileNet	×	71.4
		✓	**73.8**
	ResNet-50	×	77.2
		✓	**78.8**
	ResNet-101	×	78.3
		✓	**79.6**

B. Results on PASCAL VOC 2012

1) Effectiveness of ObjectAug: We evaluated ObjectAug based on DeepLab V3 [11] and DeepLab V3plus [12] with three backbone model MobileNet [38], ResNet-50 [4], and ResNet-101 on the PASCAL VOC 2012 dataset. Note that all baseline are implemented with traditional data augmentation methods including random scaling, random rotation and random flipping. As shown in Table I, we can see that ObjectAug improves the mIoU of MobileNet, ResNet-50 and ResNet-101 based DeepLab V3plus by 1.8%, 0.9%, and 1.1%, respectively. Meanwhile, MobileNet, ResNet-50 and ResNet-101 based DeepLab V3 obtain an improvement of 2.4%, 1.6%, and 1.3%, respectively.

2) Comparison with Traditional Data Augmentation Methods: In this part, we select four widely-used traditional data augmentation methods including random rotation, random shifting, random scaling, and random flipping for comparison. Note that the above operations can be performed at two levels: image level (the traditional approach) and object-level in ObjectAug. As shown in Table II, the operation at

TABLE II
PERFORMANCE COMPARISON OF OBJECTAUG AND TRADITIONAL DATA AUGMENTATION METHODS ON PASCAL VOC 2012.

Method	Image level	Object level	mIoU (%)
Baseline			68.5
+ R.Rotation	✓		69.8
		✓	69.5
+ R.Scaling	✓		69.9
		✓	70.3
+ R.Flipping	✓		70.1
		✓	69.6
+ R.Shifting	✓		70.3
		✓	70.7
Baseline + All	✓		71.4
Baseline + All		✓	71.2
Baseline + All	✓	✓	73.8

* All means the combination of the above four data augmentation methods.

the object level achieves competitive performance with the methods at the image level. Surprisingly, ObjectAug with random scaling at the object level outperforms the methods at the image level slightly. On the other hand, we also evaluated the combination of object-level augmentation and image-level augmentation. Notably, we can observe that the combination approach achieves much better results than each of them, which indicates that ObjectAug has no conflict with traditional data augmentation methods. This is due to the fact that the two kinds of augmentation methods operate at different levels, and thus they can exploit semantic information at different scales without overlap.

3) Comparison with DNN-based Methods: We compared our ObjectAug with the current widely-used regularization technique, CutOut [17] and CutMix [19]. Table III details the comparison of our ObjectAug, CutOut, and CutMix. ObjectAug outperforms CutOut by 1.5%, and CutMix by 1.1%, respectively. ObjectAug achieves 73.8% mIoU on PASCAL VOC 2012, which is 2.5% higher than the baseline of 71.4%. Particularly, the state-of-the-art performance of 74.1% is achieved by combining ObjectAug and CutOut. Figure 5 shows the mIoU curve of them during training, and we can discover that the improvement of ObjectAug is rather stable. Qualitative comparison are shown in Figure 6. Compared with CutOut and CutMix, ObjectAug obtains better performance especially on the boundaries.

TABLE III
PERFORMANCE COMPARISON OF OBJECTAUG AND DNN-BASED DATA AUGMENTATION METHODS ON PASCAL VOC 2012. WE USE MOBILENET BASED DEEPLAB V3PLUS MODEL AS OUR BASELINE.

Method	mIoU (%)
Baseline	71.4
+ CutOut (16×16, p = 0.5)	71.9
+ CutOut (16×16, p = 1)	72.3
+ CutMix (p = 0.5)	72.7
+ CutMix (p = 1)	72.4
+ ObjectAug	**73.8**
+ CutOut (16×16, p=0.5) + ObjectAug	73.9
+ CutMix (p=0.5) + ObjectAug	**74.1**

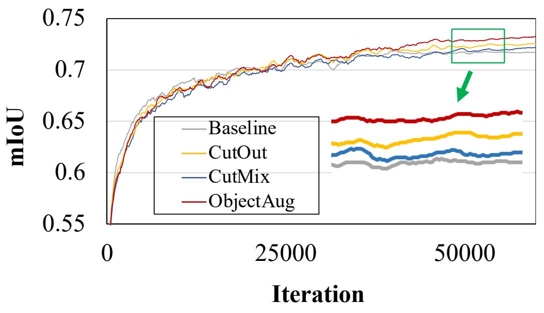


Fig. 5. Comparison of mIoU curves over iteration times between CutOut, CutMix, and ObjectAug.

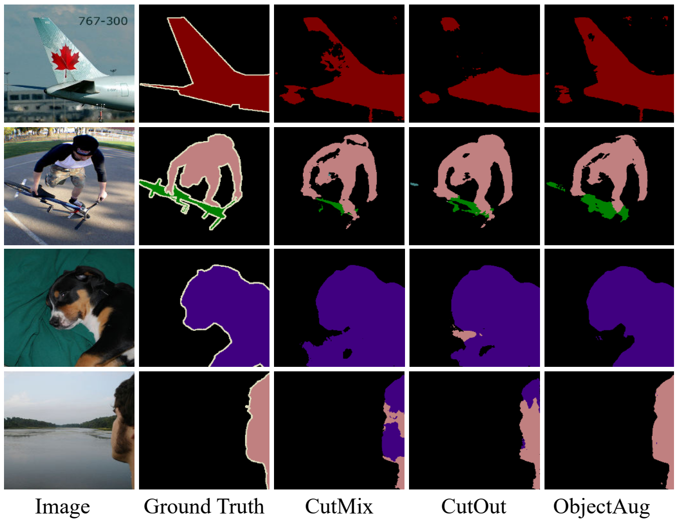


Fig. 6. Qualitative comparison of ObjectAug and DNN-based data augmentation methods (CutOut, CutMix) on PASCAL VOC 2012.

TABLE IV
ABLATION RESULTS OF IMAGE INPAINTING.

Method	Speed (s/iter)	mIoU (%)
Baseline	0.698	71.4
+ ObjectAug(w/o fill)	0.721	73.1
+ ObjectAug(random noises)	0.729	73.3
+ ObjectAug(inpainting)	0.827	73.8

4) Ablation Analysis of Image Inpainting: CutOut and CutMix take different operations on the pixel artifacts. In this part, the ablation analysis of the image inpainting module is discussed. As shown in Table IV, we can see that even if there is no processing on the pixel artifacts, our ObjectAug can still obtain a significant improvement by 1.7%. In fact, the pixel artifacts brought by this method can be treated as a unique form of CutOut. Random noise filling also enhances the robustness to some extent. Meanwhile, the inpainting method achieves the best performance among the three processing methods. On the other hand, we also evaluated the training speed of the three methods. We can see that although ObjectAug using the inpainting method can achieve the best performance among the

TABLE V
RESULT OF IMPACT DISCUSSION OF CATEGORY-AWARE COEFFICIENT.

Method	H	NH	R	NR	Total
Baseline	57.2	84.3	70.1	72.6	71.4
Rarity-driven	59.3	85.8	70.9	75.3	73.2
Hard-driven	60.2	86.2	70.6	76.7	73.8

three methods, the speed is 18.5% slower than the baseline.

5) Impact of Category-aware Strategy: In this section, we discuss two category-aware strategies: the rarity-driven coefficient and the hard-driven coefficient. We introduce two divisions for the 20 categories in PASCAL VOC 2012. In the first division, the first ten (with small number of images) are divided into rare categories group (R), while the last ten (with large number of images) are divided into non-rare categories group (NR) to analyze the effect of rarity-driven coefficient. Similarly, the hard categories (H) and the non-hard categories (NH) are obtained to evaluate hard-driven coefficient in the second division. The comparison of two strategies is shown in Table V. Experimental results show that the segmentation performance of both categories is improved by 1.8-2.4%, and the method shows more effectiveness on the hard-based rank (0.8% higher). This is due to the fact that hard cases are more critical to effectively train DNNs.

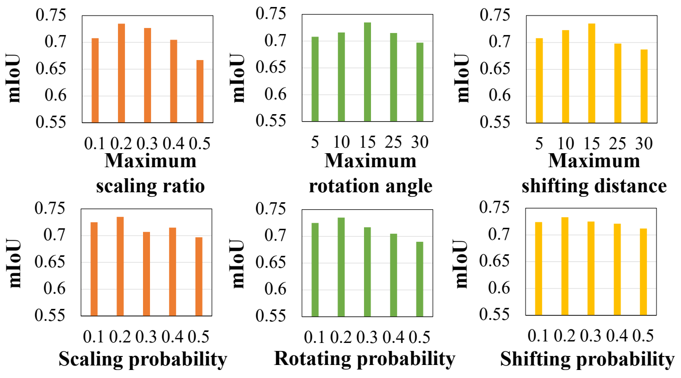


Fig. 7. Result of impact discussion of hyper-parameters including random scaling, random rotation and random shift.

6) Impact of Hyper-parameters: To study the impact of hyper-parameters, we need to assign two groups of hyper-parameters, configurations of each augmentation method and their possibilities. The first group includes the maximum scaling ratio M_z in random scaling, maximum rotation angle M_s in random rotation, and maximum shifting distance M_s in random shifting. The second is their corresponding probabilities P_z, P_r, P_s. To demonstrate the impact of these hyper-parameters on the performance, hyper-parameter settings are discussed on DeepLab V3plus (Mobile-Net). We set $M_z = 1.2, M_r = 15°, M_s = 10$ as the base setting. Results are shown in Figure 7. Notably, M_z, M_r, and M_z achieve the optimal performance with $M_z = 0.2$, $M_r = 15$, and $M_s = 5$. Meanwhile, their probabilities of $[P_z, P_r, P_s]$ demonstrate a better performance with $[0.2, 0.2, 0.1]$.

TABLE VI
PERFORMANCE COMPARISON OF MODELS WITH AND WITHOUT OBJECTAUG ON CITYSCAPES.

Method	Model	ObjectAug	mIoU (%)
DeepLab V3plus	MobileNet	×	72.0
		✓	73.5
	ResNet-50	×	75.6
		✓	76.9
	ResNet-101	×	77.4
		✓	78.5

TABLE VII
PERFORMANCE COMPARISON OF OBJECTAUG AND EXISTING DATA AUGMENTATION METHODS ON CITYSCAPES.

Method	mIoU (%)
Baseline	72.0
+ CutOut (16×16, p = 1)	72.8
+ CutMix (p = 1)	72.6
+ ObjectAug	**73.5**

C. Extended Results on Cityscapes and CRAG

We further evaluated the effectiveness of ObjectAug on hard-to-segment data on the Cityscapes dataset, and its generalization on the CRAG dataset.

1) Results on Cityscapes: In Cityscapes, we consider the hard-to-segment objects as human, vehicle, object, and construction categories, which are augmented using ObjectAug. As shown in Table VI, our ObjectAug improves MobileNet, ResNet-50 and ResNet-101 based DeepLab V3plus by 1.5%, 1.3% and 0.9% respectively. Table VII shows that ObjectAug also outperforms CutOut and CutMix by 0.7-0.9%. Compared with CutOut and CutMix with an improvement of 0.8% and 0.6% respectively, the improvement of ObjectAug almost doubles.

TABLE VIII
PERFORMANCE COMPARISON OF MODELS WITH AND WITHOUT OBJECTAUG ON CRAG.

Method	ObjectAug	mIoU (%)
FCN	×	82.2
	✓	85.5
U-Net	×	84.6
	✓	87.2
PSPNet	×	86.3
	✓	89.1
DeepLab V3	×	87.1
	✓	89.7

2) Results on CRAG: Table VIII shows the segmentation performance of FCN, U-Net [5], PSPNet [10], and DeepLab V3 [11] with and without ObjectAug on the CRAG dataset. We can see that ObjectAug can evidently improve the segmentation performance by 2.6-3.3%. As shown in Table IX, compared with other methods, ObjectAug can further improve the performance by 0.7-0.9%. Compared with CutOut and

TABLE IX
PERFORMANCE COMPARISON OF OBJECTAUG AND EXISTING DATA AUGMENTATION METHODS ON CRAG.

Method	mIoU (%)
Baseline	84.6
+ CutOut (16×16, p = 1)	85.5
+ CutMix (p = 1)	85.3
+ ObjectAug	**86.2**

CutMix with an improvement of 0.9% and 0.7% respectively, the improvement of ObjectAug almost doubles.

V. CONCLUSION

In this paper, we proposed a data augmentation method ObjectAug for semantic image segmentation. The proposed ObjectAug works at object level, which is completely different as the existing methods operating at image level. In addition, ObjectAug can support category-aware augmentation, and can be easily combined with the existing image-level augmentation methods to further enhance performance. The comprehensive experiments are conducted on both natural image and medical image datasets. Experiments cross various models and datasets demonstrate that our ObjectAug outperforms other augmentation methods and improve the segmentation performance.

VI. ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China (No.62006050).

REFERENCES

[1] X. Xu, Q. Lu, L. Yang, S. Hu, D. Chen, Y. Hu, and Y. Shi, "Quantization of fully convolutional networks for accurate biomedical image segmentation," in *CVPR*, 2018, pp. 8300–8308.
[2] X. Xu, Y. Ding, S. X. Hu, M. Niemier, J. Cong, Y. Hu, and Y. Shi, "Scaling for edge inference of deep neural networks," *Nature Electronics*, vol. 1, no. 4, pp. 216–222, 2018.
[3] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *CVPR*, 2015, pp. 3431–3440.
[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.
[5] O. Ronneberger, P. Fischer, and T. Brox, *U-Net: Convolutional Networks for Biomedical Image Segmentation*. Springer International Publishing, 2015.
[6] J. Zhang, Y. Jin, J. Xu, X. Xu, and Y. Zhang, "Mdu-net: Multi-scale densely connected u-net for biomedical image segmentation," *arXiv preprint arXiv:1812.00352*, 2018.
[7] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
[8] J. Zhang, Y. Zhang, and X. Xu, "Pyramid u-net for retinal vessel segmentation," *arXiv preprint arXiv:2104.02333*, 2021.
[9] J. Zhang, Y. Zhang, S. Zhu, and X. Xu, "Constrained multi-scale dense connections for accurate biomedical image segmentation," in *BIBM*. IEEE, 2020, pp. 877–884.
[10] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *CVPR*, 2017, pp. 2881–2890.
[11] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017.
[12] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the ECCV*, 2018, pp. 801–818.
[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NeurIPS*, 2012, pp. 1097–1105.
[14] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *CVPR*, 2016, pp. 3213–3223.
[15] L. Torrey and J. Shavlik, "Transfer learning," in *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*. IGI global, 2010, pp. 242–264.
[16] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.
[17] T. DeVries and G. W. Taylor, "Improved regularization of convolutional neural networks with cutout," *arXiv preprint arXiv:1708.04552*, 2017.
[18] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.
[19] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, "Cutmix: Regularization strategy to train strong classifiers with localizable features," in *ICCV*, 2019, pp. 6023–6032.
[20] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation." in *AAAI*, 2020, pp. 13 001–13 008.
[21] D. Hendrycks, N. Mu, E. D. Cubuk, B. Zoph, J. Gilmer, and B. Lakshminarayanan, "Augmix: A simple data processing method to improve robustness and uncertainty," *arXiv preprint arXiv:1912.02781*, 2019.
[22] H. Guo, Y. Mao, and R. Zhang, "Mixup as locally linear out-of-manifold regularization," in *AAAI*, vol. 33, 2019, pp. 3714–3722.
[23] J.-H. Kim, W. Choo, H. Jeong, and H. O. Song, "Co-mixup: Saliency guided joint mixup with supermodular diversity," *arXiv preprint arXiv:2102.03065*, 2021.
[24] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "Autoaugment: Learning augmentation policies from data," *arXiv preprint arXiv:1805.09501*, 2018.
[25] H. Ding, X. Jiang, A. Q. Liu, N. M. Thalmann, and G. Wang, "Boundary-aware feature propagation for scene segmentation," in *ICCV*, 2019, pp. 6819–6829.
[26] H. Kervadec, J. Bouchtiba, C. Desrosiers, E. Granger, J. Dolz, and I. B. Ayed, "Boundary loss for highly unbalanced segmentation," in *MIDL*, 2019, pp. 285–296.
[27] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NeurIPS*, 2012, pp. 1097–1105.
[28] L. Perez and J. Wang, "The effectiveness of data augmentation in image classification using deep learning," *arXiv preprint arXiv:1712.04621*, 2017.
[29] H. Guo, Y. Mao, and R. Zhang, "Mixup as locally linear out-of-manifold regularization," in *AAAI*, vol. 33, 2019, pp. 3714–3722.
[30] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *arXiv preprint arXiv:1406.2661*, 2014.
[31] S. Liu, J. Zhang, Y. Chen, Y. Liu, Z. Qin, and T. Wan, "Pixel level data augmentation for semantic image segmentation using generative adversarial networks," in *ICASSP*. IEEE, 2019, pp. 1902–1906.
[32] G. Liu, F. A. Reda, K. J. Shih, T.-C. Wang, A. Tao, and B. Catanzaro, "Image inpainting for irregular holes using partial convolutions," in *Proceedings of the ECCV*, 2018, pp. 85–100.
[33] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *CVPR*. Ieee, 2009, pp. 248–255.
[34] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *International journal of computer vision*, vol. 111, no. 1, pp. 98–136, 2015.
[35] B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, and J. Malik, "Semantic contours from inverse detectors," in *ICCV*. IEEE, 2011, pp. 991–998.
[36] S. Graham, H. Chen, J. Gamper, Q. Dou, P.-A. Heng, D. Snead, Y. W. Tsang, and N. Rajpoot, "Mild-net: minimal information loss dilated network for gland instance segmentation in colon histology images," *Medical image analysis*, vol. 52, pp. 199–211, 2019.
[37] W. Liu, A. Rabinovich, and A. C. Berg, "Parsenet: Looking wider to see better," *arXiv preprint arXiv:1506.04579*, 2015.
[38] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *CVPR*, 2018, pp. 4510–4520.