Data Augmentation for Object Detection via Controllable Diffusion Models

Haoyang Fang[1], Boran Han[2], Shuai Zhang[2], Su Zhou[2], Cuixiong Hu, Wen-Ming Ye
AWS AI
Bellevue, US
{haoyfang, boranhan, shuaizs, zhousu, tonyhu, wye}@amazon.com

Abstract

Data augmentation is vital for object detection tasks that require expensive bounding box annotations. Recent successes in diffusion models have inspired the use of diffusion-based synthetic images for data augmentation. However, existing works have primarily focused on image classification, and their applicability to boost object detection's performance remains unclear. To address this gap, we propose a data augmentation pipeline based on controllable diffusion models and CLIP. Our approach involves generating appropriate visual priors to control the generation of synthetic data and implementing post-filtering techniques using category-calibrated CLIP scores. The evaluation of our approach is conducted under few-shot settings in MSCOCO, full PASCAL VOC dataset, and selected downstream datasets. We observe the performance increase using our augmentation pipeline. Specifically, the mAP improvement is ***+18.0%/+15.6%/+15.9%*** *for COCO 5/10/30-shot,* ***+2.9%*** *on full PASCAL VOC dataset, and* ***+12.4%*** *on average for selected downstream datasets.*

1. Introduction

End-to-end trained deep learning models are the main workhorse behind state-of-the-art object detection methods [12, 35, 48]. A somewhat brute-force but effective recipe for further performance enhancement is to simply train these models with a larger and more diverse *annotated* dataset. However, object detection requires not only labels of the objects within each image but also accurate bounding boxes that snugly encloses each object. This extra work makes the curation of such datasets for training object-detection models substantially more laborious and less cost-effective than the image classification counterpart.

An alternative to annotating new datasets is *data augmentation* which creates more training examples by bootstrapping an existing dataset. Traditional data augmentation for object detection involves rotation, scaling, flipping and other manipulation of each image which encourages the model to learn more invariant features hence improving the robustness of the trained model. More advanced augmentation techniques involves image erasing methods (random erasing [52], GridMask [6], FenceMask [22], Cutout [8], etc.), image mix methods (Mosaic [12], Mixup [47], CutMix [46], etc.), or copy-paste methods that replicate image samples [13]. Even more advanced data-augmentation methods are *generative* — they leverage the recent advances in generative models such as CLIP and stable diffusion models [15, 32, 36, 41] to create synthetic training images.

Intuitively, generative data-augmentation adds diversity, realism and novel visual features in the augmented examples. These are impossible with non-generative data-augmentation methods. Not surprisingly, they result in major performance gains in downstream vision tasks [1, 14, 23, 38, 39, 53].

However, unlike traditional data-augmentation methods where the bounding box annotations can be calculated in a straightforward manner, it is unclear how to perform generative data-augmentation with bounding box labels. For this reason, all aforementioned work that utilize generative data-augmentation are restricted to image classification tasks. Admittedly, there are specialized generative models trained to generate data with bounding boxes: (1) layout-to-image models [7, 17, 50] usually requires a dense bounding box distribution for training and does not apply to object detection tasks [5]. While [5] presented its performance for data augmentation on a downstream object detection dataset [2], we are not able to find their code for more experiments, nor compare on that dataset [2] due to the license restriction. (2) copy-paste with diffusion models [11, 51] generate images of target objects guided by text only on a plain diffusion model and utilize extra off-the-shelf segmentation models [26, 27, 29, 30, 42] to cut the objects off and paste to a real image. Thus the data generated is less realistic [5]. Note that all aforementioned methods require training on object detection or segmentation datasets.

[1] Corresponding author.
[2] These authors contributed equally to this work.

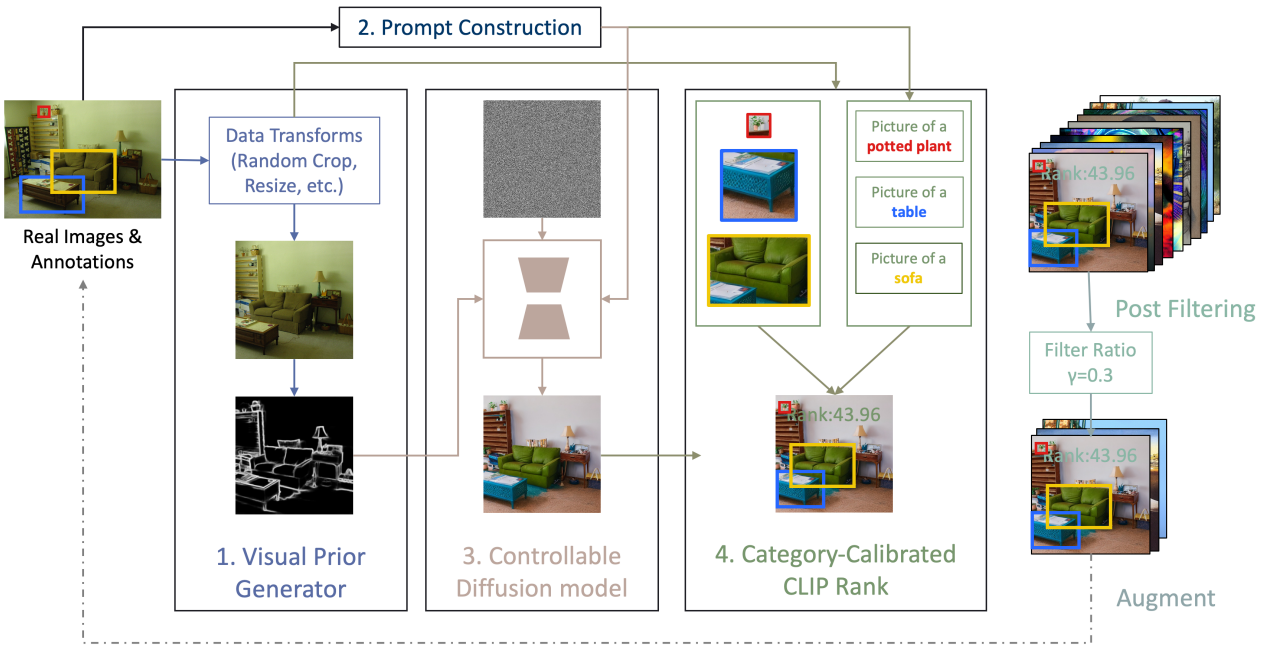

Figure 1. The data augmentation pipeline for object detection based on controllable diffusion model: 1. Generate visual priors 2. Construct prompts for the whole image and for each bounding boxes 3. Generate synthetic data via the controllable diffusion model 4. Compute category-calibrated CLIP rank and perform post filtering.

In this paper, we address the following natural question:

Can we perform *generative* data augmentation for object detection via diffusion models with more fine-grained control and without human annotations?

Our first idea is to specify a bounding box and then used diffusion-based inpainting approaches [36, 45] to generate the object within. In this way, we have both the object and its bounding box. A minor caveat is that the object might not fill the bounding box tightly. We will evaluate this approach as a baseline.

Our second — arguably more interesting — idea combines controllable diffusion models for guided text-to-image generation [49] with visual priors such as HED boundary [44], semantic segmentation masks [21] that we obtain from each image of the original annotated dataset. The generated image thus inherits the high quality bounding box annotation but with different style, lighting, or even completely new objects inside. We also propose a novel category-calibrated CLIP scores [31] to filter out those images where the object inside the bounding box is not compatible with the prompt. Further performance gains were obtained by integrating our first idea that uses inpainting based methods into our pipeline.An illustration of our proposed method is shown Figure 1.

To evaluate the effectiveness of our approach, we conduct extensive experiments under both few-shot settings with the MSCOCO [24] dataset, standard settings with the PASCAL VOC [9] dataset, and several downstream datasets [16, 19, 37, 40]. The few-shot settings reflect scenarios with limited annotated data, while the full data settings represent the training scenarios with ample annotations. Our comprehensive evaluation aims to showcase the versatility and robustness of our approach across different data regimes.

Summary of results. Our main contributions are:

- Designing a simple but effective method to generate synthetic image with high quality bounding boxes annotation using a carefully controlled diffusion model.
- Automatic data-quality control by filtering with a category-calibrated CLIP scores.
- Integrating an inpainting method [45] into our pipeline to further improve the detector's performance.
- Systematic evaluation of our method in both few-shot and full data settings. Improve the YOLOX detector's mAP result by **+18.0%/+15.6%/+15.9%** for COCO 5/10/30-shot, **+2.9%** for full PASCAL VOC dataset, and **+12.4%** on average for downstream datasets.

Related work: The idea of *generative* data augmentation originates from [1, 14, 23, 38, 39, 53], but their methods are restricted to classification tasks. While copy-paste methods

and layout-to-image methods require off-the-shelf segmentation models or training on data with dense bounding box layouts, we aim to perform data augmentation for *object detection* efficiently via controllable diffusion model while *no human curated annotations is needed*. The resulting object detection model significantly advances the existing state-of-the-art in the few-show setting. The main components of our method leverages the exciting recent advances in generative AI [31, 36, 41, 49], especially the larger foundation models trained with multi-modal image-text corpora [31, 36, 49]. We emphasize that neither controllable diffusion models [49] nor CLIP scores [31] are new, but the application of them for generating synthetic images that come with high-quality bounding box annotations as well as the use of these images to enhance object detection are new to this paper.

2. Method

In this section, we break down the key components of our proposed image data augmentation pipeline for object detection.

This pipeline consists of (1) A visual prior generator; (2) A prompt constructor; (3) A controllable diffusion model; (4) A post filter with category-calibrated CLIP [31] rank. Figure 1 illustrates how different components coordinate and generate the synthetic data for object detection. Below we will describe each component in details.

2.1. Prior Extractor

Given a set of N image-annotation pairs $(x_i, y_i)_{i=1}^N$ for training, with ground truth categories $c \in C$, we randomly sample M image-annotation pairs, and then perform regular data transforms on the image and the annotation. Then we use visual prior extractor to get the M "visual prior"-annotation pairs $(v_j, \hat{y}_j)_{j=1}^M$, where each v_j is of pixel size 512x512. Our default visual prior extractor is HED edge detector [44] to balance the visual diversity and bounding box quality. In Section 4.7, we discuss using other visual prior extractors, such as Canny edge detector [3] for better bounding box quality, segmentation mask [21] for better visual diversity.

2.2. Prompt Construction

We then construct the prompt p_j based on each annotation $\hat{y}_j$. Annotation is composed of one or several "bounding box"-category pairs. Our default strategy is to take all the category labels in $\hat{y}_j$, concatenating them in a sentence, seperated by comma. We also explored several other strategies for prompt construction, and found that the default one is simple yet effective. More details are discussed in Section 4.3.

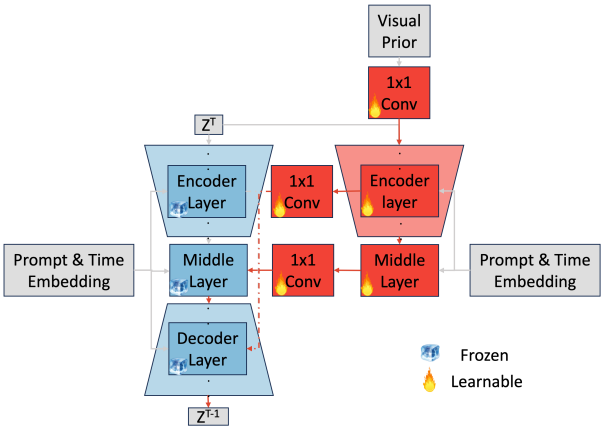

Figure 2. Structure of the controllable diffusion model. Z^T is the latent representation at time T of a latent diffusion model [36]. Weights of blue blocks are frozen and only weights of red blocks are updated during training. Red lines are the paths that gradients flow.

2.3. Controllable Diffusion Model

The controllable diffusion model follows [49] and is shown in Figure 2, where we keep two copies of the pretrained diffusion model, one with all parameters frozen, and the other with only encoder blocks and a middle block, connecting with the skip connections and the middle layer of the frozen copy by 1x1 convolutions initialized with zero. Note that since we use the visual prior to force the structure of the synthetic image, the annotation $\hat{y}_j$ can still be used as ground truth. Thus given the prompt, we can use controllable diffusion model F to generate synthetic images for each (v_j, p_j) pair: $\hat{x}_j = F(v_j, p_j)$, and thus get M synthetic image-annotation pair $(\hat{x}_j, \hat{y}_j)$.

2.4. Post Filter with Category-Calibrated CLIP Rank

For each annotation $\hat{y}_j$, it contains some "bounding box"-label pairs $(b_j^k, l_j^k)_{k=1,2,...}$. We cropped $\hat{x}_j$ with the bounding box b_j^k to get image content inside the bounding box $\hat{x}_j^k$, and use CLIP [31] to compute the similarity score: $s_j^k = CLIP(\hat{x}_j^k, p_j^k)$, where p_j^k is the text prompt constructed based on the label l_j^k.

Then we collect similarity scores for all bounding boxes in all annotations $\hat{y}_j$ for each category: $B_{c \in C}$. And for each bounding box label pairs (b_j^k, l_j^k), we compute r_j^k: the descending order rank of b_j^k in $B_{l_j^k}$. Then we compute the rank score R_j for the synthetic data pair $(\hat{x}_j, \hat{y}_j)$ by averaging all the $r_{b_j^k}$ in the annotation $\hat{y}_j$.

For post filtering, we define the filtering ratio as γ, and we keep top γM synthetic data pairs ranked by R_j. And we define the augmentation ratio α as the number of synthetic

	5 shot		10 shot		30 shot	
	mAP	AP50	mAP	AP50	mAP	AP50
YOLOX-S	5.0	10.1	9.6	18.1	14.2	26.7
+ SDInpaint	5.3	11.1	10.6	19.9	14.6	26.4
+ PbE	5.5	**11.4**	9.8	18.9	14.7	26.5
+ Ours	**5.9 (+18.0%)**	**11.4 (+12.8%)**	**11.1 (+15.6%)**	**20.6 (+13.8%)**	**15.9 (+12.0%)**	**27.8 (+4.1%)**
DINO-SwinL	18.6	26.0	24.3	33.7	**35.8**	**49.5**
+ Ours	**20.3 (+9.1%)**	**28.1 (+8.1%)**	**26.0 (+7.0%)**	**36.8 (+9.2%)**	35.0 (-2.2%)	48.8 (-1.4%)

Table 1. We evaluate our data augmentation approach with a one-stage lightweight detector YOLOX-S [12] and a high performance transformer based detector DINO-SwinL [48] on COCO under 5/10/30-shot, and report the improvement on mAP and AP50 metrics.

images over real images: $\alpha = \frac{\gamma M}{N}$.

3. Main Results

In this section, we conduct our experiments under few shot settings on COCO [24] dataset, and standard settings with full data used for training on PASCAL VOC [9] and other selected downstream datasets [16, 19, 40] to verify the effectiveness of the proposed pipeline on different domains.

For object detectors, we choose YOLOX [12] for extensive experiments due to efficiency concerns. It extends [33, 34] by introducing a series of optimizations such as Mosaic and Mixup augmentation, Decoupled Head, and SimOTA, to enhance both speed and accuracy, and achieves SOTA detection accuracy among one-stage detectors while maintaining real-time inference speeds. We also include results for DINO [48] to show our improvement on SOTA detectors. DINO [48] is based on further previous transformer based detectors [4, 20, 25] and adds contrastive denoising, mixed query selection with dynamic anchors and static content queries to further improve the performance and reduce time for convergence.

3.1. Experiment Settings

Our default setting for synthetic data generation is to use filtering ratio $\gamma = 30\%$, augmentation ratio $\alpha = 1$, synthetic image size of 512x512, DDIM sampler [41] with 50 steps and guidance scale of 9.0, and HED edges [44] as visual prior.

For YOLOX-S [12] detector, we use SGD optimizer with batch size of 64, learning rate of 1e-2, momentum of 0.9, weight decay of 5e-4, and 200 epochs for pre-training. Same optimization but with learning rate of 5e-3, backbone frozen, and 20k iterations for finetuning.

For DINO-SwinL [48] detector, we use AdamW optimizer with batch size of 16, learning rate of 1e-4, weight decay of 1e-4, and 36 epochs for base training. Same optimizer but with backbone frozen and 10k iterations for finetuning.

We report both COCO-standard mAP and VOC-standard AP50 for comparison, and also include AP75, mAP-small, mAP-medium, mAP-large as a supplement in some experiments. All experiments are conducted with random seeds set to 1 on an AWS EC2 p3dn.24xlarge server with 8x V100(32G) GPUs.

3.2. Few Shot

We first evaluate our data augmentation pipeline under Few-Shot Object Detection (FSOD) setting on COCO dataset [24]. FSOD is an emerging and challenging area in computer vision that addresses the problem of detecting objects in images with very limited labeled training data [10, 18, 43]. It explores the scenario where the model needs to generalize to novel objects it has never seen during pretraining, given only a few examples of each new class.

It comprises 80 object categories divided into 60 base categories and 20 novel categories that are identical to the 20 classes in PASCAL VOC dataset [9]. The base category data from the COCO [24] training sets are used to pretrain the model. To simulate the few-shot scenario, K-shot instances (where K = 5, 10, or 30) are randomly sampled from the previously unseen novel classes. We are skipping the 1/3 shot because the results with this number of shots suffer from very high variance as shown in [43].

As shown in Table 1, we compared our method with baseline (base training + few shot finetuning [43]) on YOLOX-S [12] and DINO-SwinL [48] on COCO [24] with 5/10/30 shots. We also extended our analysis to include a comparison with the inpainting Stable Diffusion (SDInpaint) [36] and Paint-by-Example (PbE) [45], where we randomly sample and mask a bounding box and perform inpainting to produce a synthetic image.

We notice that as the number of available shots increases, the improvement of inpainting methods significantly diminishes. This is because when the data size is limited, synthetic objects with loosely fitting bounding boxes can still yield favorable results for the model. However, as the amount of real data grows, the accuracy of the ground truth bounding box becomes crucial, and the loose bounding boxes in synthetic data may even cause a performance degrade: there is a drop in AP50 under 30-shot for SDInpaint [36] and PbE [45].

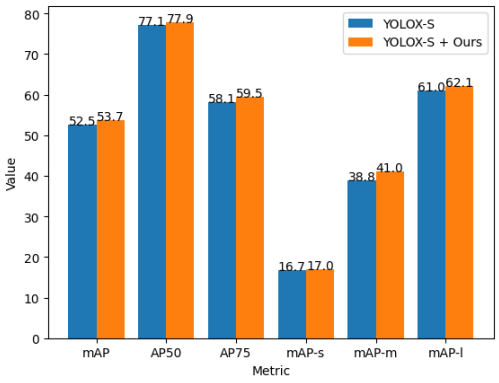

Figure 3. We evaluate our approach on full PASCAL VOC dataset, and show the performance improvement with YOLOX-S.

As we are using visual priors to supervise the generation in our approach, it constantly outputs precise bounding boxes. Though it may sacrifice the diversity of the object appearance, it still generally outperforms baseline by a large margin on both YOLOX [12], a one-stage lightweight detector, and DINO [48], a transformer based high-performance detector.

3.3. PASCAL VOC Dataset

We evaluate the performance on full PASCAL VOC dataset [9]. We show that our approach is able to boost the performance of the detector even with ample annotations. The result is shown in Figure3, where we use VOC0712trainval [9] to train YOLOX-S [12] with SGD for 300 epochs with learning rate of 0.01, Nesterov momentum of 0.9, and weight decay of 0.0005, and evaluate on VOC07 [9] test set. YOLOX-S has a significant improvement with our augmentation: mAP **+1.2**, mAP50 **+0.8**, and mAP75 **+1.4**.

3.4. Downstream Object Detection Tasks

Using the same detector settings, we further evaluate the performance on a few downstream object detection datasets selected from [16, 19, 37, 40] (due to license issues we cannot cover all) to prove the generalization capability of our appoach. The result is shown in Table 2. In general our approach improves the detector's performance by a large margin.

4. Discussion and Analysis

We further evaluate our proposed data augmentation pipeline to answer the following questions:

1. How important is our post filtering with category-calibrated CLIP rank?
2. What is an appropriate augmentation ratio?
3. How should we construct the prompts?
4. Does it work well with other augmentations?
5. Given sufficient synthetic data, do we still need real data for training?
6. Is is possible to combine inpainting methods with our approach?
7. How does other type of visual priors work?

	mAP		AP50	
	YOLOX-S	+Ours	YOLOX-S	+Ours
Watercolor [16]	11.6	**16.5**	26.2	**35.7**
Raccoon [19]	22.8	**37.5**	70.1	**78.8**
Thermal [19]	61.0	**72.2**	89.6	**94.0**
Plantdoc [40]	**39.8**	38.6	**54.0**	53.4
deepfruits [37]	**57.6**	51.5	**87.2**	80.0
comic [16]	10.1	**12.0**	22.2	**26.2**
Avg.	33.8	**38.0**	58.2	**61.4**

Table 2. We evaluate our approach on several downstream datasets in some interesting domains without any finetuning or adaptation for the diffusion model.

4.1. Post Filtering is Necessary

	5 shot		10 shot		30 shot	
	mAP	AP50	mAP	AP50	mAP	AP50
no filter	5.2	10.7	10.5	19.9	15.0	26.4
50%	5.8	**11.4**	11.0	20.3	15.3	27.0
30%	**5.9**	**11.4**	**11.1**	**20.6**	**15.9**	**27.8**
20%	5.5	10.8	10.4	19.4	15.8	**27.8**
10%	4.5	9.4	10.5	19.2	15.4	26.8

Table 3. Results of YOLOX-S on COCO dataset under 5/10/30-shot setting with different filtering ratio γ. "no filter" refers to performance without post filtering. Note the data volumes after post filtering are fixed.

We investigate how different post-filtering ratios affect the performance of the augmentation. As shown in Table 3, we generate synthetic images with the same hyperparameters and same random seeds, and apply different post-filtering ratios, while making sure the augmentation ratios after post-filtering are the same, i.e. $\alpha = 1$. And report the mAP and AP50 for 3/5/10 shot on COCO. We notice that after post filtering the performance is generally better than unfiltered data. Especially when the filtering ratio is 30%, it outperforms unfiltered data by a large margin: **+13.5%** mAP for 5 shot, **+5.7%** mAP for 10 shot, **+6.0%** mAP for 30 shot.

But we also notice that when the filtering ratio is further lower, the performance drops, which contradicts the intuition that when the ratio is lower, the synthetic data quality should be better, and thus, the improvement should be

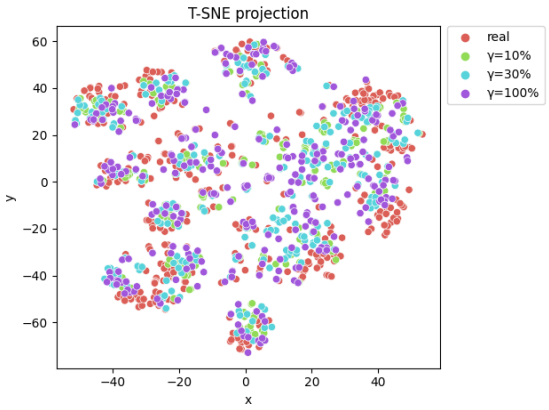

Figure 4. Distribution of real data and synthetic data with different filtering ratio γ under 30-shot.

larger. Figure 4 shows the distribution of real data and synthetic data of different filtering ratios and explains this well. When the filtering ratio is too low, synthetic data from some "bad" real image disappear, and the synthetic images are mostly crowded around a few "good" real images, and sacrificing excessive generalization that cannot be compensated for by higher data quality.

4.2. Choose A Moderate Augmentation Ratio

After we produced high quality synthetic data, a critical question arises: how much synthetic data should be utilized? What should be a good value for the augmentation ratio α?

Figure 5 presents the ablation study on the choice of different augmentation ratio used during training. Observations reveal that optimal performance is generally achieved when $1 \leq \alpha \leq 4$. To balance performance and efficiency, the default augmentation ratio is set to $\alpha = 1$ in all other sections of this work.

Intuitively, it is commonly perceived that an increased volume of data correlates positively with enhanced performance. But we notice that when augmentation ratio goes further larger, the performance drops. Reason behind this is also related to domain shifts, similar to the findings in Section 4.1. Since we do not adapt the diffusion model to target object detection data domain, the domain gap exists, mostly in image styles, between the synthetic and real images. The introduction of an excessive amount of synthetic data changes the overall data distribution, thereby potentially confuse the object detector.

4.3. Simple Prompts Work Well

Previous diffusion based data augmentation work mostly involves only one category per prompt, for example, [51] using a fixed prompt "a photo of a single

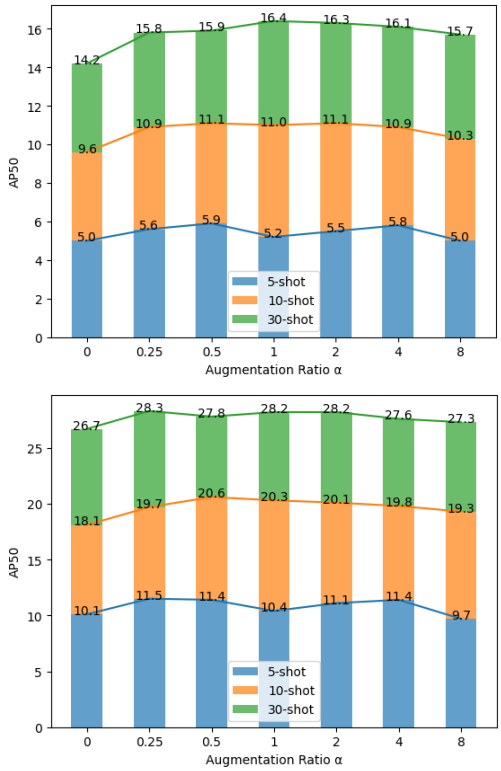

Figure 5. Ablation study on different augmentation ratios under 5/10/30-shot settings. We observe that optimal performance is generally achieved when $1 \leq \alpha \leq 4$.

$\langle category\_name \rangle$", [11] using a mix of six fixed patterns, and [1] simply using the category name. In our case, a prompt with multiple category names are required, so we extend previous methods to explore different ways of constructing a prompt based as described in Table 4, where "concatenat" refers to simply concatenate category name of all bounding boxes separated by comma with duplication, "and" referring to use the word "and" for separation, "shuffledset" referring to use the category names in a random order separated by comma with duplication removed, "shuffledsetand" referring to further use "and" for separation, "img" referring to add "An image of" at the beginning of the concatenated sentence, and "mix" referring to use a mix of all above strategies with random selections.

And we show the result in Figure 6. It turns out that simply concatenating all the category names, either separated by comma or word "and", works constantly well. While adding more strategies causes a decrease in the robustness.

Inspired by this finding, we further experiment on adding additional prompt keywords that may increase the visual quality of the synthetic images [28]. We used the proposed best prompts, the prompts with top-15 frequency in [28] comparing with no additional prompts. Synthetic image samples are show in Figure 7 and evaluation results on

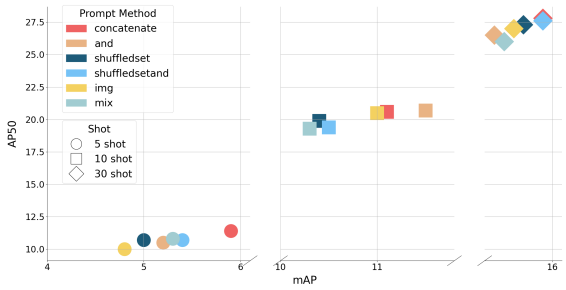

Figure 6. We show the 5/10/30-shot performance for different prompt construction strategies. Each color represents a strategy as shown at top left corner. And circles, squares, and diamonds shape represent 5/10/30-shot respectively.

COCO 10-shot are shown in Table 5.

	prompt
concatenate	"A, A, B, C, ..."
and	"A and A and B and C ..."
shuffledset	"B, C, A, ..."
shuffledsetand	"C and A and B..."
img	"An image of A, A, B, C, ..."
mix	A mix of strategies above

Table 4. Different prompt construction strategies. A, B, C are category names of objects in the image. Note that one image may have multiple objects of a same category.

	mAP	AP50	AP75	mAP-s	mAP-m	mAP-l
None	11.1	20.6	10.5	3.2	9.8	17.3
BestPrompt	10.4	18.8	10.2	3.1	8.4	16.3
Top15Prompt	10.4	19.3	9.7	3.0	8.4	16.3

Table 5. Using visual enhancement prompts from [28] for data generation does not further improve the detector's performance.

4.4. Works Well With Other Augmentations

We explore the combination of our approach and other data augmentations and compare the results. Specifically, we test on YOLOX-S with and without mosaic augmentation [12], and on DINO-SwinL with and without random choice resizing [48]. The result is shown in Figure 8. We observe that our approach remains consistent irrespective of the presence of additional data augmentations.

4.5. Real Data is Important

We compare the results of mixing all synthetic data with different percentages of real data in VOC dataset [9] in Table 7. We use the percentages of 0%, 1%, 10%, 50%, and 100%, and report the mAP, AP50, AP75, and mAP-small/medium/large metrics. It shows that mAP grows from

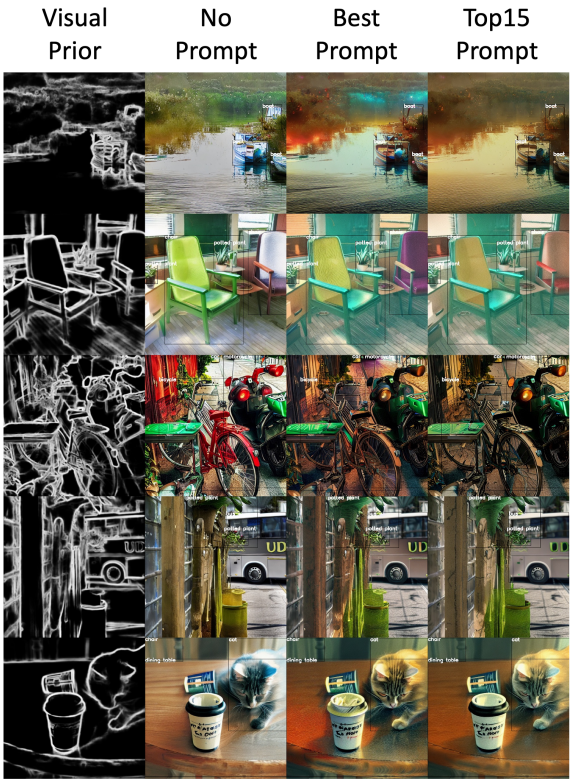

Figure 7. Synthetic images with guidance of different additional prompts.

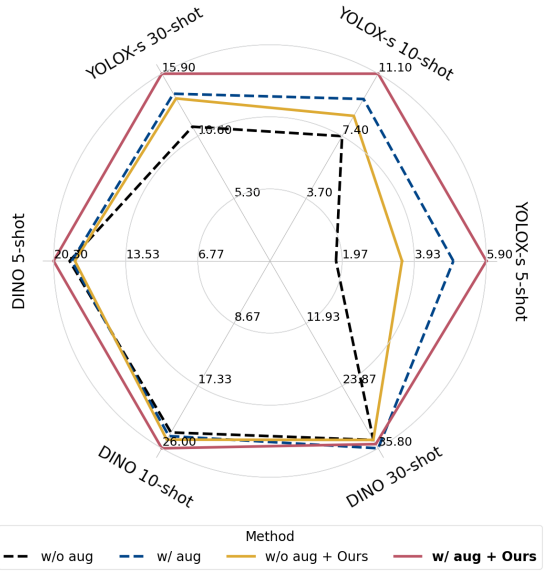

Figure 8. We evaluate on YOLOX-S and DINO with and without other augmentations.

	mAP	AP50	AP75	mAP-s	mAP-m	mAP-l
PbE	8.8	17.7	7.9	2.9	7.0	13.7
PbE + HED	9.4	19.1	8.2	2.6	7.5	14.5
PbE + 30%PF	11.1	20.9	10.4	3.1	8.8	**17.5**
PbE + HED + 30%PF	11.0	**21.1**	10.4	**3.2**	9.7	16.7
PbE(less) + HED(more) + 30%PF	**11.2**	21.0	**10.8**	**3.2**	**9.9**	17.0

Table 6. We further integrate PbE with different approaches and notice an improvement while we generate a few PbE data, then more data controlled by HED and filtered at a ratio of 30%.

22.6 to 52.5 as real data percentage rises fro 0% to 100%. Thus the detector's performance is heavily relied on the amount of real data. This suggests that our approach should be used as an enrichment but cannot replace real data.

	mAP	AP50	AP75	mAP-s	mAP-m	mAP-l
0%	22.6	43.6	21.0	3.0	12.0	29.1
1%	26.3	49.6	25.4	5.3	15.4	32.9
10%	31.9	57.1	31.9	7.2	19.7	39.3
50%	43.0	70.5	46.1	14.9	31.0	50.4
100%	52.5	77.1	58.1	16.7	38.8	61.0

Table 7. Detection results on PASCAL VOC [9] dataset with different percentage of real data for training. There is a significant drop as real data become less.

4.6. Integration of Inpainting

Inpainting method can introduce more object varieties to the data due to its large scale pretraining on as much as billions of image-caption pairs [36]. However, currently it is infeasible to enforce the inpainting algorithm draw an object to fill the bounding box, which causes the box not tight enough and thus introduces noise to training data. In Section 3.2, we present how our approach outperforms inpainting methods on COCO dataset [24] under few shot settings. Here we futher explore the possibility to integrate inpainting method into our pipeline to further increase performance.

We experiment on COCO dataset under 10-shot setting on integrating inpainting method PbE [45] into our approach. And we show in Table 6 that (1) PbE: Using PbE alone cannot improve the detector's performance. (2) PbE + Ours: Adding synthetic images from our approach can further improve PbE's performance, and beats real data only in AP50 by +1.0, but got -0.1 in mAP. (3) PbE + 30%PF: Using PbE with 30% post filtering with category-calibrated clip scores increase the performance by a large margin. (4) PbE + Ours + 30%PF: Adding synthetic images from our approach and post filtering can help the model perform better at detecting small/medium objects. (5) PbE(less) + Ours(more) + 30%PF: Decreasing the amount of synthetic data from PbE and increasing synthetic data from ours while keeping the total synthetic data amount unchanged reach the best performance for augmentation.

4.7. Other Visual Priors

We compare using different visual priors, i.e. HED edge [44], canny edge [3], segmentation mask generated by Uniformer [21], and Scribble generated with HED [44,49]. We compare those method on COCO under few shot settings and the result is shown in Table 8. Comparing with HED edge, Canny edge is less robust due to it is more vulnerable to noises, although it outperforms HED edge on mAP in 10-shot, it generally produces synthetic data with lower quality and has inferior performance. While mask and scribble visual prior produce more diverse object appearance, they also suffer from more synthetic features. Note that only Uniformer [21] that generates mask visual priors is trained on segmentation datasets.

	5-shot		10-shot		30-shot	
	mAP	AP50	mAP	AP50	mAP	AP50
Canny	5.2	10.6	11.2	20.4	14.0	25.4
HED	**5.9**	**11.4**	11.1	20.6	**15.9**	**27.8**
Uniformer	5.2	10.8	**11.5**	**20.9**	14.3	25.5
Scribble	5.1	10.6	9.7	18.5	13.9	24.7

Table 8. We compare the augmentation results controlled by different visual priors. From top to down the control of visual priors turns from fine to coarse.

5. Conclusion

We introduce a novel data augmentation approach designed for object detection tasks based on controllable diffusion model and CLIP. Our evaluations are conducted on COCO datasets under few-shot settings, full PASCAL VOC dataset, and downstream object detection datasets. The results demonstrate that our approach significantly enhances the performance of object detectors.

We delve into various interesting questions of our methodology, and further show that the integration of inpainting methods further elevates its effectiveness. Given the synergy observed between our approach and other data augmentation techniques, we note that our method can be combined with other data augmentation methods to further increase the performance. We hope this can be a strong baseline for future work.

References

[1] Shekoofeh Azizi, Simon Kornblith, Chitwan Saharia, Mohammad Norouzi, and David J Fleet. Synthetic data from diffusion models improves imagenet classification. *arXiv preprint arXiv:2304.08466*, 2023. 1, 2, 6

[2] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 1

[3] John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986. 3, 8

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 4

[5] Kai Chen, Enze Xie, Zhe Chen, Lanqing Hong, Zhenguo Li, and Dit-Yan Yeung. Integrating geometric control into text-to-image diffusion models for high-quality detection data generation via text prompt. *arXiv preprint arXiv:2306.04607*, 2023. 1

[6] Pengguang Chen, Shu Liu, Hengshuang Zhao, and Jiaya Jia. Gridmask data augmentation. *arXiv preprint arXiv:2001.04086*, 2020. 1

[7] Jiaxin Cheng, Xiao Liang, Xingjian Shi, Tong He, Tianjun Xiao, and Mu Li. Layoutdiffuse: Adapting foundational diffusion models for layout-to-image generation. *arXiv preprint arXiv:2302.08908*, 2023. 1

[8] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017. 1

[9] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88:303–338, 2010. 2, 4, 5, 7, 8

[10] Qi Fan, Wei Zhuo, Chi-Keung Tang, and Yu-Wing Tai. Few-shot object detection with attention-rpn and multi-relation detector. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4013–4022, 2020. 4

[11] Yunhao Ge, Jiashu Xu, Brian Nlong Zhao, Laurent Itti, and Vibhav Vineet. Dall-e for detection: Language-driven context image synthesis for object detection. *arXiv preprint arXiv:2206.09592*, 2022. 1, 6

[12] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*, 2021. 1, 4, 5, 7

[13] Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D Cubuk, Quoc V Le, and Barret Zoph. Simple copy-paste is a strong data augmentation method for instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2918–2928, 2021. 1

[14] Ruifei He, Shuyang Sun, Xin Yu, Chuhui Xue, Wenqing Zhang, Philip Torr, Song Bai, and Xiaojuan Qi. Is synthetic data from generative models ready for image recognition? *arXiv preprint arXiv:2210.07574*, 2022. 1, 2

[15] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 1

[16] Naoto Inoue, Ryosuke Furuta, Toshihiko Yamasaki, and Kiyoharu Aizawa. Cross-domain weakly-supervised object detection through progressive domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5001–5009, 2018. 2, 4, 5

[17] Naoto Inoue, Kotaro Kikuchi, Edgar Simo-Serra, Mayu Otani, and Kota Yamaguchi. Layoutdm: Discrete diffusion model for controllable layout generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10167–10176, 2023. 1

[18] Bingyi Kang, Zhuang Liu, Xin Wang, Fisher Yu, Jiashi Feng, and Trevor Darrell. Few-shot object detection via feature reweighting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 4

[19] Chunyuan Li, Haotian Liu, Liunian Harold Li, Pengchuan Zhang, Jyoti Aneja, Jianwei Yang, Ping Jin, Houdong Hu, Zicheng Liu, Yong Jae Lee, and Jianfeng Gao. Elevater: A benchmark and toolkit for evaluating language-augmented visual models. *Neural Information Processing Systems*, 2022. 2, 4, 5

[20] Feng Li, Hao Zhang, Shilong Liu, Jian Guo, Lionel M Ni, and Lei Zhang. Dn-detr: Accelerate detr training by introducing query denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13619–13627, 2022. 4

[21] Kunchang Li, Yali Wang, Junhao Zhang, Peng Gao, Guanglu Song, Yu Liu, Hongsheng Li, and Yu Qiao. Uniformer: Unifying convolution and self-attention for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 2, 3, 8

[22] Pu Li, Xiangyang Li, and Xiang Long. Fencemask: a data augmentation approach for pre-extracted image features. *arXiv preprint arXiv:2006.07877*, 2020. 1

[23] Zheng Li, Yuxuan Li, Penghai Zhao, Renjie Song, Xiang Li, and Jian Yang. Is synthetic data from diffusion models ready for knowledge distillation? *arXiv preprint arXiv:2305.12954*, 2023. 1, 2

[24] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 2, 4, 8

[25] Shilong Liu, Feng Li, Hao Zhang, Xiao Yang, Xianbiao Qi, Hang Su, Jun Zhu, and Lei Zhang. Dab-detr: Dynamic anchor boxes are better queries for detr. *arXiv preprint arXiv:2201.12329*, 2022. 4

[26] Timo Lüddecke and Alexander Ecker. Image segmentation using text and image prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7086–7096, 2022. 1

[27] Timo Lüddecke and Alexander Ecker. Image segmentation using text and image prompts. In *Proceedings of*

the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 7086–7096, 2022. 1

[28] Nikita Pavlichenko and Dmitry Ustalov. Best prompts for text-to-image models and how to find them. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2067–2071, 2023. 6, 7

[29] Lu Qi, Jason Kuen, Yi Wang, Jiuxiang Gu, Hengshuang Zhao, Philip Torr, Zhe Lin, and Jiaya Jia. Open world entity segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 1

[30] Xuebin Qin, Zichen Zhang, Chenyang Huang, Masood Dehghan, Osmar R Zaiane, and Martin Jagersand. U2-net: Going deeper with nested u-structure for salient object detection. *Pattern recognition*, 106:107404, 2020. 1

[31] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 2, 3

[32] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 1

[33] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 4

[34] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 4

[35] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015. 1

[36] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 1, 2, 3, 4, 8

[37] Inkyu Sa, Zongyuan Ge, Feras Dayoub, Ben Upcroft, Tristan Perez, and Chris McCool. Deepfruits: A fruit detection system using deep neural networks. *sensors*, 16(8):1222, 2016. 2, 5

[38] Mert Bulent Sariyildiz, Karteek Alahari, Diane Larlus, and Yannis Kalantidis. Fake it till you make it: Learning transferable representations from synthetic imagenet clones. In *CVPR 2023–IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 1, 2

[39] Jordan Shipard, Arnold Wiliem, Kien Nguyen Thanh, Wei Xiang, and Clinton Fookes. Diversity is definitely needed: Improving model-agnostic zero-shot classification via stable diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 769–778, 2023. 1, 2

[40] Davinder Singh, Naman Jain, Pranjali Jain, Pratik Kayal, Sudhakar Kumawat, and Nipun Batra. Plantdoc: A dataset for visual plant disease detection. In *Proceedings of the 7th ACM IKDD CoDS and 25th COMAD*, pages 249–253. 2020. 2, 4, 5

[41] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 1, 3, 4

[42] Yukun Su, Jingliang Deng, Ruizhou Sun, Guosheng Lin, Hanjing Su, and Qingyao Wu. A unified transformer framework for group-based segmentation: Co-segmentation, co-saliency detection and video salient object detection. *IEEE Transactions on Multimedia*, 2023. 1

[43] Xin Wang, Thomas E Huang, Trevor Darrell, Joseph E Gonzalez, and Fisher Yu. Frustratingly simple few-shot object detection. *arXiv preprint arXiv:2003.06957*, 2020. 4

[44] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *Proceedings of the IEEE international conference on computer vision*, pages 1395–1403, 2015. 2, 3, 4, 8

[45] Binxin Yang, Shuyang Gu, Bo Zhang, Ting Zhang, Xuejin Chen, Xiaoyan Sun, Dong Chen, and Fang Wen. Paint by example: Exemplar-based image editing with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18381–18391, 2023. 2, 4, 8

[46] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019. 1

[47] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 1

[48] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M Ni, and Heung-Yeung Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. *arXiv preprint arXiv:2203.03605*, 2022. 1, 4, 5, 7

[49] Lvmin Zhang and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. *arXiv preprint arXiv:2302.05543*, 2023. 2, 3, 8

[50] Bo Zhao, Lili Meng, Weidong Yin, and Leonid Sigal. Image generation from layout. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8584–8593, 2019. 1

[51] Hanqing Zhao, Dianmo Sheng, Jianmin Bao, Dongdong Chen, Dong Chen, Fang Wen, Lu Yuan, Ce Liu, Wenbo Zhou, Qi Chu, et al. X-paste: Revisiting scalable copy-paste for instance segmentation using clip and stablediffusion. 2023. 1, 6

[52] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 13001–13008, 2020. 1

[53] Yongchao Zhou, Hshmat Sahak, and Jimmy Ba. Training on thin air: Improve image classification with generated data. *arXiv preprint arXiv:2305.15316*, 2023. 1, 2