CVPR
#4792

ForAug: Mitigating Biases and Improving Vision Transformer Training by Recombining Foregrounds and Backgrounds

Anonymous CVPR submission

Paper ID 4792

Abstract

Transformers, particularly Vision Transformers (ViTs), have achieved state-of-the-art performance in large-scale image classification. However, they often require large amounts of data and can exhibit biases, such as center or size bias, that limit their robustness and generalizability. This paper introduces ForAug, a novel data augmentation operation that addresses these challenges by explicitly imposing invariances into the training data, which are otherwise part of the neural network architecture. ForAug is constructed by using pretrained foundation models to separate and recombine foreground objects with different backgrounds. This recombination step enables us to take fine-grained control over object position and size, as well as background selection. We demonstrate that using ForAug significantly improves the accuracy of ViTs and other architectures by up to 4.5 percentage points (p.p.) on ImageNet, which translates to 7.3 p.p. on downstream tasks. Importantly, ForAug not only improves accuracy but also opens new ways to analyze model behavior and quantify biases. Namely, we introduce metrics for background robustness, foreground focus, center bias, and size bias and show that using ForAug during training substantially reduces these biases. In summary, ForAug provides a valuable tool for analyzing and mitigating biases, enabling the development of more robust and reliable computer vision models. Our code and dataset are publicly available at `<url>`.

1. Introduction

Image classification, a fundamental task in computer vision (CV), involves assigning labels to images from a set of categories. It underpins a wide range of applications, like medical diagnosis [37, 48], autonomous driving [51], and object recognition [2, 14, 16] and facilitates large-scale pretraining [9, 29, 45], and progress evaluation in CV [21, 34]. The advent of large-scale datasets, particularly ImageNet [7], served as a catalyst for the rise of large-scale CV models [15, 23] and remains the most important CV benchmark for more than a decade [15, 23, 46, 53]. While traditionally, convolutional neural networks (CNNs) have been the go-to architecture in CV, Transformers [47], particularly the Vision Transformer (ViT) [9], have emerged as a powerful alternative and go-to architecture, demonstrating superior performance in various vision tasks, including image classification [2, 50, 53, 56, 61].

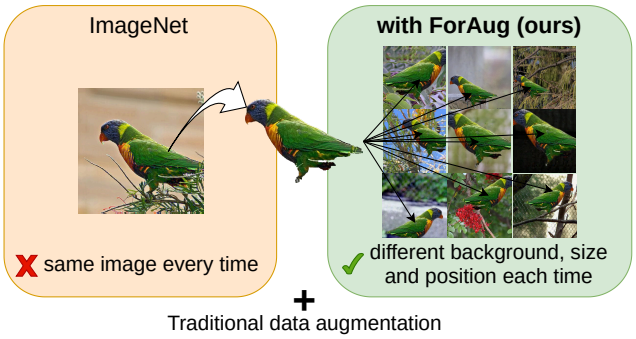


Figure 1. Comparison of traditional image classification training and training when using *ForAug*. *ForAug* recombines foreground objects with different backgrounds each epoch, thus creating a more diverse training set. We still apply strong traditional data augmentation afterwards.

Data augmentation is a key technique for training image classification models. Traditional augmentation methods, such as cropping, flipping, or color shifts, are commonly employed to increase data diversity [40, 55], but remain bound to existing image compositions. While these preserve the images' semantic meaning, their ability to teach spatial invariances is limited. While combinations of these data augmentations are still used today, they originally were proposed to benefit CNNs. However, the architectural differences of CNNs and Transformers suggest that the latter might benefit from different data augmentation strategies. In particular, the self-attention mechanism, unlike a CNN, is not translation equivariant [8, 36], meaning that the model is not designed to understand the spatial relationships between pixels.

Recognizing that Transformers need to learn spatial relationships directly from data, we propose *ForAug*, a data augmentation method that makes these relationships explicit by recombining foreground objects with diverse backgrounds. Thus, *ForAug* goes beyond existing image compositions and encodes desired invariances directly into the training data (see Figure 1). Applying *ForAug* to a dataset like ImageNet is a two-step process: (1) We separate the foreground objects in ImageNet from their backgrounds, using an open-world object detector [35] and fill in the background in a neutral way using an object removal model [41, 43]. (2) This allows us to then recombine any foreground object with any background on the fly, creating a highly diverse training set. By exploiting the control over foreground size and position during recombination, *ForAug* explicitly teaches spatial invariances that image classification models typically must learn implicitly. We show that using *ForAug* additionally to strong traditional data augmentation increases the model accuracy of Transformers by up to 4.5 p.p. on ImageNet and reduces the error rate by up to 7.3 p.p. in downstream tasks.

Beyond training, *ForAug* becomes a diagnostic tool for analyzing model behavior and biases, when used during evaluation. We utilize our control over the image distribution to measure a model's background robustness (by varying the choice of background), foreground focus (by leveraging our knowledge about the placement of the foreground object), center bias (by controlling position), and size bias (by controlling size). These analyses provide valuable insights into model behavior and biases, which is crucial for model deployment and future robustness optimizations. We show that training using *ForAug* significantly reduces all of these biases. We make our code for *ForAug* and the output of *ForAug*'s segmentation phase on ImageNet publicly available[1] to facilitate further research.

Contributions

- We propose *ForAug*, a novel data augmentation scheme, that recombines objects and backgrounds. *ForAug* allows us to move beyond the (possibly biased) image compositions in the dataset while preserving label integrity.
- We show that training a standard ViT using *ForAug* leads to up to 4.5 p.p. improved accuracy on ImageNet-1k and 7.3 p.p. on downstream tasks.
- We propose novel *ForAug*-based metrics to analyze and quantify fine-grained biases of trained models: Background Robustness, Foreground Focus, Center Bias, and Size Bias. We show that *ForAug* significantly reduces these biases by encoding invariance that benefits ViT into the training data.

[1]Link will go here.

2. Related Work

Data Augmentation for Image Classification Data augmentation is a crucial technique for improving the performance and generalization of image classification models. Traditional augmentation strategies rely on simple geometric or color-space transformations like cropping, flipping, roatation, blurring, color jittering, or random erasing [60] to increase the diversity of the training data without changing their semantic meaning. With the advent of Vision Transformers, new data augmentation operations like PatchDropout [28] have been proposed. Other transformations like Mixup [59], CutMix [57], or random cropping and patching [44] combine multiple input images. These simple transformations are usually bundled to form more complex augmentation policies like AutoAugment [4] and RandAugment [5], or 3-augment [46] which is optimized to train a ViT. For a general overview of data augmentation techniques for image classification, we refer to Shorten and Khoshgoftaar [40], Xu et al. [55].

We build upon these general augmentations by introducing a novel approach to explicitly separate objects and backgrounds for image classification, allowing us to – unlike these basic transformations – move beyond dataset image compositions. Our approach is used additionally to strong traditional techniques to improve performance and reduce biases.

Copy-Paste Augmentation The copy-paste augmentation [13], which is used only for object detection [13, 39] and instance segmentation [26, 52], involves copying segmented objects from one image and pasting them onto another. While typically human annotated segmentation masks are used to extract the foreground objects, other foregound sources have been explored, like 3D models [18] and pretrained object-detection models for use on objects on white background [10] or synthetic images [11]. [19] apply copy-paste as an alternative to CutMix in image classification, but they do not shift the size or position of the foregrounds and use normal dataset images as backgrounds.

Unlike prior copy-paste methods that overlay objects, *ForAug* extracts foregrounds and replaces their backgrounds with semantically neutral fills, thereby preserving label integrity while enabling controlled and diverse recombination.

Model robustness evaluation Evaluating model robustness to various image variations is critical for understanding and improving model generalization. Datasets like ImageNet-C [17] and ImageNet-P [17] introduce common corruptions and perturbations. ImageNet-E [25] evaluates model robustness against a collection of distribution shifts. Other datasets, such as ImageNet-D [58], focus on varying background, texture, and material, but rely on synthetic data.

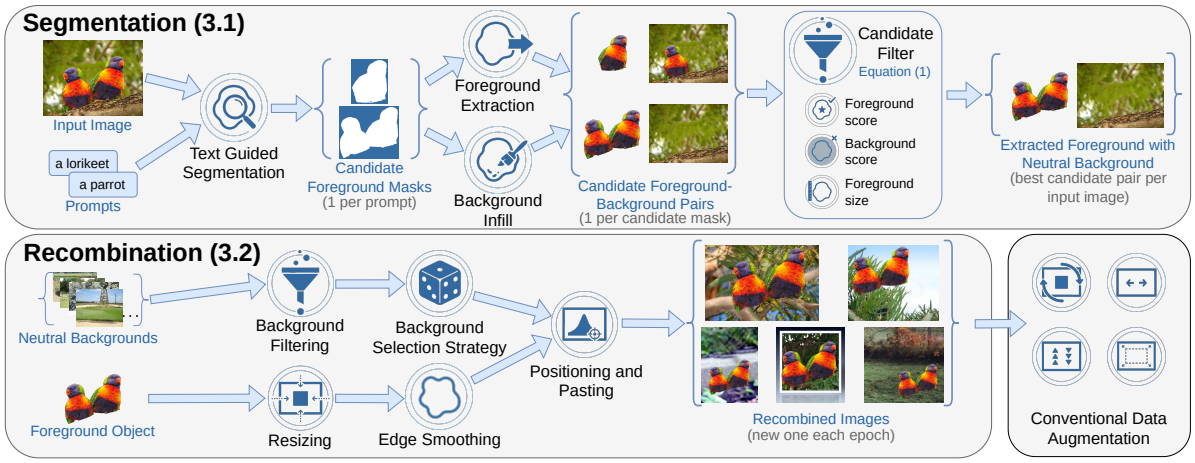


Figure 2. Overview of *ForAug*. The data creation consists of two stages: Segmentation (offline, Section 3.1), where we segment the foreground objects from the background and fill in the background. Recombination (online, Section 3.2), where we combine the foreground objects with different backgrounds to create new samples. After recombination, we apply strong, commonly used augmentation policies.

Stylized ImageNet [12] investigates the impact of texture changes. ImageNet-9 [54] explores background variations using segmented images, but backgrounds are often artificial.

In contrast to these existing datasets, which are used only for evaluation, *ForAug* provides fine-grained control over foreground object placement, size, and background selection, enabling a precise and comprehensive analysis of specific model biases within the context of a large-scale, real-world image distribution. As *ForAug* also provides controllable training set generation, it goes beyond simply measuring robustness to actively improving it through training.

3. *ForAug* (Method)

We introduce *ForAug*, a data augmentation designed to enhance Transformer training by embedding spatial invariances–which Transformers would otherwise need to learn implicitly–directly into the training data. *ForAug* comprises two distinct stages: Segmentation and Recombination. Both stages are illustrated in Figure 2.

3.1. Segmentation

The segmentation stage isolates the foreground objects and their corresponding backgrounds. We then fill the background using a pretrained object-removal model, producing visually plausible [41], neutral scenes ready for recombination. This stage is computed once offline and the results are stored for the recombination stage.

First, foreground objects are detected and segmented from their backgrounds using a prompt-based segmentation model to exploit the classification datasets labels. We use the state-of-the-art Grounded SAM [35], which is based on Grounding DINO [27] and SAM [22]. The prompt we use is "`a <class name>, a type of <object category>`", where `<class name>` is the specific name of the objects class as defined by the dataset and `<object category>` is a the broader category of the object. The `<object category>` guides the segmentation model towards the correct object in case the `<class name>` alone is too specific. This can be the case with prompts like "sorrel" or "guenon", where the more general name "horse" or "monkey" is more helpful. We derive the `<object category>` from the WordNet hierarchy, using the immediate hypernym.

We iteratively extract n foreground masks for each dataset-image, creating prompts by going one hypernym up the WordNet-tree each step (e.g. "a sorrel, a type of horse", "a horse, a type of equine", ...). Masks that are very similar, with a pairwise IoU of at least 0.9, are merged. The output is a set of masks delineating the foreground objects and the backgrounds. We select the best mask per image (according to Equation (1)) in a later filtering step, described below.

First, an inpainting model that is specifically optimized to remove objects from images, such as LaMa [43] or Attentive Eraser [41], is used to inpaint the foreground regions in the backgrounds. Then, to ensure the quality of the foregrounds and the neutral background images, we select a foreground/background pair (for each dataset-image) from the $\leq n$ variants we have extracted and infilled in the previous steps. Using an ensemble E of six ViT, ResNet, and Swin Transformer models pretrained on the original dataset, we select the foreground/background pair that maximizes foreground performance while minimizing the performance on the background and size of the foreground. For each model $m \in E$, we predict the score of the ground truth class c on the foreground fg and background bg and weigh these

with the size $\mathrm{size}(\cdot)$ in number of pixels according to:

$$\begin{aligned}\mathrm{score}(\mathrm{fg},\mathrm{bg},c) = &\log\left(\frac{1}{|E|}\sum_{m\in E}\mathbb{P}[m(\mathrm{fg})=c]\right)\\ &+\log\left(1-\frac{1}{|E|}\sum_{m\in E}\mathbb{P}[m(\mathrm{bg})=c]\right)\\ &+\lambda\log\left(1-\left|\frac{\mathrm{size}(\mathrm{fg})}{\mathrm{size}(\mathrm{bg})}-\varepsilon\right|\right).\end{aligned} \quad (1)$$

We run a hyperparameter search using a manually annotated subset of foreground/background variants to find the factors in Equation (1): $\lambda = 2$ and $\varepsilon = 0.1$.

Finally, we filter out backgrounds that are largely infilled, as these tend to be overly synthetic and do not carry much information (see the supplementary material). Although the segmentation stage is computational overhead, it is a one-time cost with results that can be reused across experiments (see the supplementary material for details). In summary, we factorize the dataset into a set of foreground objects with a transparent background and a set of diverse backgrounds per class. The next step is to recombine these, before applying other common data augmentation operations during training.

3.2. Recombination

The recombination stage, performed online during training, combines the foreground objects with different backgrounds to create new training samples. For each object, we follow the pipeline of: Pick an appropriate background, resize it to a fitting size, and place it in the background image. Through this step, we expose the model to variations beyond the image compositions of the dataset.

For each foreground object, we sample a background using one of the following strategies: (1) the original image background, (2) the set of backgrounds from the same class, or (3) the set of all possible backgrounds. These sets are trading off the amount of information the model can learn from the background against the diversity of new images created. In each epoch, each foreground object is seen exactly once, but a background may appear multiple times.

The selected foreground is resized based on its relative size within its original image and the relative size of the original foreground in the selected background image. The final size is randomly selected from a 30% range around upper and lower limits (s_u and s_l), based on the original sizes. To balance the size of the foreground and that of the backgrounds original foreground, the upper and lower limit s_u and s_l are set to the mean or range of both sizes, depending on the foreground size strategy: *mean* or *range*.

The resized foreground is then placed at a random position within the background image. To more seamlessly integrate the foreground, we apply a Gaussian blur with $\sigma \in [\frac{\sigma_{\max}}{10}, \sigma_{\max}]$, inspired by the standard range for the Gaussian blur operation in [46], to the foreground's alpha-mask.

Table 1. Ablation of the design decisions in the segmentation phase of *ForAug* on TinyImageNet. The first line is our baseline, while the other lines are using *ForAug*. We use basic settings with the *same* background strategy during recombination for this experiment.

Detect. Prompt	Infill Model	TinyImageNet Accuracy [%] ViT-Ti	ViT-S
TinyImageNet		66.1 ± 0.5	68.3 ± 0.7
specific	LaMa [43]	65.5 ± 0.4	71.2 ± 0.5
general	LaMa [43]	66.4 ± 0.6	72.9 ± 0.6
general	Att. Eraser [41]	67.5 ± 1.2	72.4 ± 0.5

We can apply standard data augmentation techniques in two modes: Either we apply all augmentations to the recombined image, or we apply the cropping and resizing to the background only and then apply the other augmentations after recombination. The first mode mirrors standard augmentation practice, whereas the second one ensures the foreground object remains fully visible.

We experiment with a constant mixing ratio, or a linear or cosine annealing schedule that increases the amount of images from the original dataset over time. The mixing ratio acts as a probability of selecting an image from the original dataset; otherwise, an image with the same foreground is recombined using *ForAug*, ensuring each object is seen once per epoch. The recombination stage is designed to be parallelized on the CPU during training and thus does not impact training time (see supplementary material for details).

4. Experiments

We conduct a comprehensive suit of experiments to validate the effectiveness of our approach, comparing ImageNet-training with and without *ForAug* for 10 different models. Furthermore, we assess the impact of using *ForAug* for pre-training on multiple fine-grained downstream datasets. Finally, we exploit *ForAug*'s control over the image distribution to quantify model behaviors and biases. We always report the mean and standard deviation of three independent training runs.

4.1. Design Choices of ForAug

We start by ablating the design choices of *ForAug* on TinyImageNet [24], a subset of ImageNet containing 200 categories with 500 images each. Table 1 presents ablations for segmentation and Table 2 for recombination.

Prompt. First, we evaluate the type of prompt used to detect the foreground object. Here, the *general* prompt, which contains the class and the more general object category, outperforms only having the class name (*specific*).

Inpainting. Among inpainting models, Attentive Eraser [41] produces slightly better results compared to LaMa [43] (+0.5 p.p. on average). For inpainting examples,

Table 2. Ablation of the recombination phase of *ForAug* on TinyImageNet (top) and ImageNet (bottom). The first experiments use the initial segmentation settings with LaMa [43].

FG. size	Augment. Order	BG Strat.	BG. Prune	Original Mixing	Edge Smooth.	Accuracy [%] ViT-Ti	ViT-S
TinyImageNet						66.1 ± 0.5	68.3 ± 0.7
mean	crop→paste	same	-	-	-	64.6 ± 0.5	70.0 ± 0.6
range	crop→paste	same	-	-	-	65.5 ± 0.4	71.2 ± 0.5
range	crop→paste	same	-	-	-	67.5 ± 1.2	72.4 ± 0.5
range	paste→crop	same	-	-	-	67.1 ± 1.2	72.9 ± 0.5
range	paste→crop	same	1.0	-	-	67.0 ± 1.2	73.0 ± 0.3
range	paste→crop	same	0.8	-	-	67.2 ± 1.2	72.9 ± 0.8
range	paste→crop	same	0.6	-	-	67.5 ± 1.0	72.8 ± 0.7
range	paste→crop	same	0.8	$p = 0.2$	-	69.8 ± 0.5	75.0 ± 0.3
range	paste→crop	same	0.8	$p = 0.33$	-	69.5 ± 0.4	75.2 ± 1.0
range	paste→crop	same	0.8	$p = 0.5$	-	70.3 ± 1.0	74.2 ± 0.2
range	paste→crop	same	0.8	linear	-	70.1 ± 0.7	74.9 ± 0.8
range	paste→crop	same	0.8	reverse lin.	-	67.6 ± 0.2	73.2 ± 0.3
range	paste→crop	same	0.8	cos	-	71.3 ± 1.0	75.7 ± 0.8
range	paste→crop	same	0.8	cos	$\sigma_{\max} = 4.0$	70.0 ± 0.8	75.5 ± 0.7
range	paste→crop	orig.	0.8	cos	$\sigma_{\max} = 4.0$	67.2 ± 0.9	69.9 ± 1.0
range	paste→crop	all	0.8	cos	$\sigma_{\max} = 4.0$	70.1 ± 0.7	77.5 ± 0.6
ImageNet						-	79.1 ± 0.1
range	paste→crop	same	0.8	cos	-	-	80.5 ± 0.1
range	paste→crop	same	0.8	cos	$\sigma_{\max} = 4.0$	-	80.7 ± 0.1
range	paste→crop	all	0.8	cos	$\sigma_{\max} = 4.0$	-	81.4 ± 0.1

see the supplementary material.

Foreground size significantly impacts performance. Employing a *range* of sizes during recombination, rather than a fixed *mean* size, boosts accuracy by approximately 1 p.p. This suggests that the added variability is beneficial.

Order of data augmentation. Applying all augmentations after foreground-background recombination (*paste→crop→color*) improves ViT-S's performance compared to applying crop-related augmentations before pasting (*crop→paste→color*). ViT-Ti results are ambiguous.

Background pruning. When it comes to the backgrounds to use, we test different pruning thresholds (t_{prune}) to exclude backgrounds with large inpainting. A threshold of $t_{\text{prune}} = 1.0$ means that we use all backgrounds that are not fully infilled. Varying t_{prune} has minimal impact. We choose $t_{\text{prune}} = 0.8$ to exclude predominantly artificial backgrounds.

Mixing *ForAug*-augmented samples with the original ImageNet data proves crucial. While constant and linear mixing schedules improve performance over no mixing by $2-3$ p.p. compared to only augmented samples, the cosine annealing schedule proves optimal, boosting accuracy by $3-4$ p.p.

Edge smoothing. We evaluate the impact of using Gaussian blurring to smooth the edges of the foreground masks. For larger models, this gives us a slight performance boost on the full ImageNet (second to last line in Table 2).

Background strategy. Another point is the allowed choice of background image for each foreground object. We compare using the original background, a background from the same class, and any background. These strategies go from low diversity and high shared information content between the foreground and background to high diversity and low shared information content. For *ViT-Ti*, the latter two strategies perform comparably, while *ViT-S* benefits from the added diversity of using any background. The same is true when training on the full ImageNet.

Table 3. Accuracy of ViT-S on TinyImageNet (TIN) in percent using *ForAug* with different foreground position distributions by varying the Bates parameter η. The best performance is achieved when using the uniform distribution ($\eta = 1$) for training.

Bates Parameter during training	TIN w/o *ForAug*	TIN w/ *ForAug* $\eta=-3$	-2	$1/-1$	2	3
Baseline	68.9	60.5	60.2	60.8	62.6	63.1
$\eta = -3$	71.3	79.3	79.5	79.1	79.3	79.1
$\eta = -2$	71.5	80.0	78.7	79.3	79.1	78.8
$\eta = 1/-1$	72.3	79.5	78.9	80.2	79.7	80.4
$\eta = 2$	71.3	78.2	77.8	79.1	79.6	79.9
$\eta = 3$	71.4	77.2	76.9	78.6	79.6	79.7

Table 4. Dataset statistics for TinyImageNet and ImageNet with and without *ForAug*. For *ForAug* we report the number of foreground/background pairs.

Dataset	Classes	Training Images	Validation Images
TinyImageNet	200	100 000	10 000
TinyImageNet + *ForAug*	200	99 404	9915
ImageNet	1000	1 281 167	50 000
ImageNet + *ForAug*	1000	1 274 557	49 751

Foreground position. Finally, we analyze the foreground object's positioning in the image, using a generalization of the Bates distribution [1] with parameter $\eta \in \mathbb{Z}$. The Bates distribution presents an easy way to sample from a bounded domain with just one hyperparameter that controls its concentration. $\eta = 1/-1$ corresponds to the uniform distribution; $\eta > 1$ concentrates the distribution around the center; and for $\eta < -1$, the distribution is concentrated at the borders (see supplementary material for details). When sampling more towards the center of the image, the difficulty of the task is reduced, which reduces performance on TinyImageNet (Table 3). This is reflected in the performance when evaluating using *ForAug* with $\eta = 2$ and $\eta = 3$ compared to $\eta = -1/1$. We observe a similar reduction for $\eta < -1$.

After fixing the optimal design parameters in Tables 1 and 2 (last rows), we run *ForAug*'s segmentation step on the entire ImageNet dataset. Table 4 shows the resulting dataset statistics. The slightly reduced image count for *ForAug* is due to instances where Grounded SAM fails to produce valid segmentation masks.

4.2. Image Classification Results

Table 5 compares the ImageNet performance of models trained with and without *ForAug*. We adopt the training setup of [31] and [46] for training ViT [9], Swin [29] and ResNet [15] (representing CNNs) models as well as the setup of DeiT [45] for that model. Both setups are using strong

Table 5. ImageNet results of models trained on ImageNet with and without *ForAug*. *ForAug* improves the performance of most models, with a larger gain for larger models.

Model	ImageNet Accuracy [%]		Delta
	w/o *ForAug*	w/ *ForAug*	
ViT-S	79.1 ± 0.1	81.4 ± 0.1	+2.3
ViT-B	77.6 ± 0.2	81.1 ± 0.4	+3.5
ViT-L	75.3 ± 0.4	79.8 ± 0.1	+4.5
DeiT-S	80.1 ± 0.1	80.0 ± 0.3	−0.1
DeiT-B	81.9 ± 0.3	81.9 ± 0.2	±0.0
DeiT-L	79.3 ± 2.3	82.4 ± 0.1	+3.1
Swin-Ti	77.9 ± 0.2	79.7 ± 0.1	+1.8
Swin-S	79.4 ± 0.1	80.6 ± 0.1	+1.2
ResNet-50	78.3 ± 0.1	78.8 ± 0.1	+0.5
ResNet-101	79.4 ± 0.1	80.4 ± 0.1	+1.0

Table 6. Comparison of *ForAug* and simple Copy-Paste methods. We train ViT-S on ImageNet using the same 3-augment data augmentation on top of the copy-paste augmentation.

Augmentation	labels	Accuracy [%]	Delta to Prev.
Baseline + **Simple Copy-Paste**	bg	31.3 ± 0.6	
+ mixed labels	fg + bg	32.0 ± 0.8	+0.7
+ fg labels	fg	31.6 ± 0.9	−0.4
+ *range* foreground size variation	fg	43.0 ± 1.2	**+11.4**
+ infilled backgrounds	fg	68.7 ± 0.2	**+25.7**
+ *cos* mixing strategy	fg	81.2 ± 0.1	**+12.5**
+ edge smoothing	fg	81.3 ± 0.1	+0.1
+ background pruning= ***ForAug***	fg	81.4 ± 0.1	+0.1

data augmentations like RandAugment, CutMix, and Mixup optimized for Transformers (details in supplementary material). Notably, *ForAug* improves performance across all tested architectures, including the ResNet models, demonstrating benefits beyond Transformers. For DeiT we only observe benefits on ImageNet for the larger models. For other transformers, we observe improvements from 1.2 p.p. to 4.5 p.p. with increasing gains for larger models. *ForAug*'s improvements counteract the drop in performance for increasing model sizes. Without *ForAug* this drop is 3.8 p.p. (ViT-S to L), while with *ForAug* it is reduced to 1.6 p.p. For DeiT there is a drop of 0.8 p.p. from small to large while when using *ForAug* there is a *gain* of 2.4 p.p.

Comparison to Simple Copy-Paste. We compare *ForAug* to a simple adaption of the Copy-Paste augmentation inspired by [11, 13, 39] in Table 6. Contrary to semantic segmentation we do not have foreground masks available. Thus, we paste the extracted foreground objects from *ForAug's segmentation stage* onto normal ImageNet images. We observe 3 large jumps in accuracy: (**1**) From our *range* foreground size variation (+11.4%), (**2**) from using our infilled backgrounds instead of images from the dataset (+25.7%), and (**3**) from our *cos* mixing strategy with non-augmented images (+12.5%). *ForAug*'s changes to the naive copy-paste augmentation are thus imperative for good classification performance.

Table 7. Downstream accuracy in percent when finetuning on other datasets. Models are pretrained on ImageNet with and without *ForAug*. Pretraining using *ForAug* increases transformer downstream accuracy.

Model	*ForAug*	Aircraft	Cars	Flowers	Food	Pets
ViT-S	✗	72.4 ± 1.0	89.8 ± 0.3	94.5 ± 0.2	89.1 ± 0.1	93.8 ± 0.2
ViT-S	✓	78.6 ± 0.5	92.2 ± 0.2	95.5 ± 0.2	89.6 ± 0.1	94.5 ± 0.2
		+6.2	+2.4	+1.0	+0.5	+0.7
ViT-B	✗	71.7 ± 0.5	90.0 ± 0.2	94.8 ± 0.4	89.8 ± 0.2	94.1 ± 0.4
ViT-B	✓	79.0 ± 2.2	93.3 ± 0.1	96.5 ± 0.1	90.9 ± 0.1	95.1 ± 0.4
		+7.3	+3.3	+1.7	+1.1	+1.0
ViT-L	✗	72.1 ± 1.0	88.8 ± 0.3	94.4 ± 0.3	90.1 ± 0.2	94.2 ± 0.4
ViT-L	✓	77.6 ± 1.2	89.1 ± 0.2	96.6 ± 0.1	91.3 ± 0.1	95.1 ± 0.1
		+5.5	+0.3	+2.2	+1.2	+0.9
DeiT-S	✗	75.3 ± 0.4	91.1 ± 0.2	94.8 ± 0.4	89.2 ± 0.2	92.4 ± 0.2
DeiT-S	✓	76.8 ± 0.8	91.9 ± 0.2	95.2 ± 0.3	89.1 ± 0.2	92.3 ± 0.4
		+1.5	+0.8	+0.4	−0.1	−0.1
DeiT-B	✗	77.0 ± 1.2	92.9 ± 0.2	96.1 ± 0.2	91.2 ± 0.1	93.3 ± 0.4
DeiT-B	✓	79.3 ± 0.3	93.1 ± 0.1	96.4 ± 0.2	91.3 ± 0.1	93.3 ± 0.1
		+2.3	+0.2	+0.3	+0.1	±0.0
DeiT-L	✗	72.8 ± 5.5	92.8 ± 1.0	95.8 ± 1.5	90.5 ± 2.6	92.4 ± 2.0
DeiT-L	✓	78.8 ± 0.8	93.8 ± 0.2	97.0 ± 0.2	92.0 ± 0.2	93.5 ± 0.2
		+6.0	+1.0	+1.2	+1.5	+1.1
Swin-Ti	✗	77.0 ± 0.1	91.3 ± 0.6	95.9 ± 0.1	90.0 ± 0.2	94.2 ± 0.1
Swin-Ti	✓	81.1 ± 0.8	92.8 ± 0.4	96.2 ± 0.1	90.4 ± 0.3	94.8 ± 0.5
		+4.1	+2.5	+0.3	+0.4	+0.6
Swin-S	✗	75.7 ± 1.4	91.0 ± 0.3	95.9 ± 0.5	91.1 ± 0.2	94.4 ± 0.1
Swin-S	✓	81.4 ± 0.2	93.1 ± 0.2	96.3 ± 0.3	91.2 ± 0.2	94.9 ± 0.3
		+5.7	+2.1	+1.4	+0.1	+0.5
ResNet-50	✗	78.2 ± 0.5	89.8 ± 0.2	91.7 ± 0.4	84.4 ± 0.2	93.7 ± 0.3
ResNet-50	✓	80.3 ± 0.4	90.4 ± 0.2	91.7 ± 0.2	84.5 ± 0.2	93.7 ± 0.3
		+2.1	+0.6	±0.0	+0.1	±0.0
ResNet-101	✗	78.4 ± 0.6	90.3 ± 0.1	91.2 ± 0.5	86.0 ± 0.2	94.3 ± 0.2
ResNet-101	✓	81.4 ± 0.5	91.3 ± 0.1	92.9 ± 0.2	86.3 ± 0.1	94.0 ± 0.3
		+3.0	+1.3	+1.7	+0.3	−0.3

Downstream tasks. To assess the transferability of *ForAug*-trained models, we finetune models pretrained on ImageNet with and without *ForAug* on five fine-grained datasets: FGVC-Aircraft [30], Stanford Cars [6], Oxford Flowers [32], Food-101 [20], and Oxford-IIIT Pets [33]. In Table 7 we see transformer accuracies improve on all these datasets by up to 7.3 p.p. Notably, training with *ForAug* boosts the downstream performance of DeiT-S and DeiT-B, despite similar ImageNet results. This shows the improved representations from training with *ForAug* translate to gains beyond better ImageNet scores.

4.3. Bias and Robustness Evaluation

Beyond its use for training, *ForAug*'s unique properties and controlled data generation capabilities make it a powerful tool for analyzing behavior and biases of black-box models.

Background Robustness. We assess the robustness of models to shifts in the background distribution from a class-related background to any background. Figure 3 presents the background robustness results for three datasets: Im-

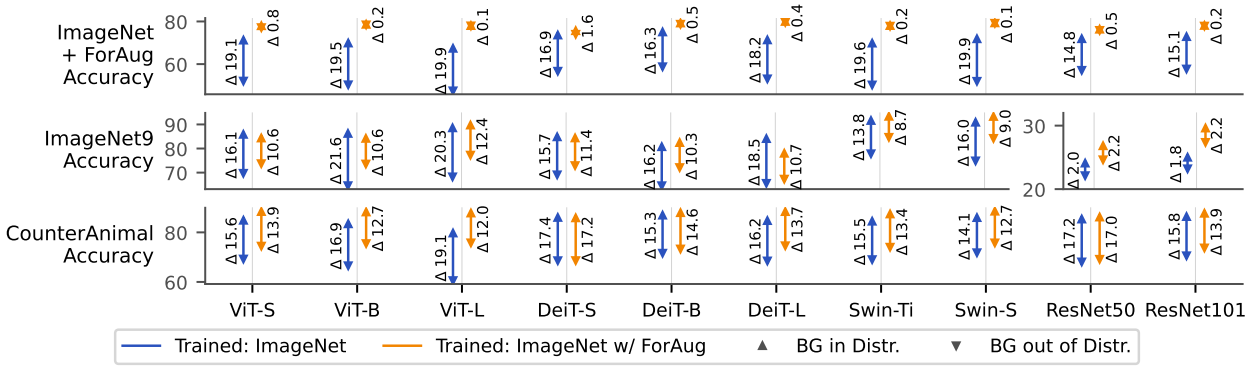


Figure 3. Evaluation of background robustness on ImageNet + *ForAug*, ImageNet9 and CounterAnimal. We plot the in-distribution (top of arrow) and the out-of-distribution (bottom of arrow) accuracy when training with and without *ForAug*. We annotate each arrow with its length Δ. Training with *ForAug* improves the background robustness of all transformers by mostly boosting the out-of-distribution accuracy.

ageNet with *ForAug* (all backgrounds vs. backgrounds of same class), ImageNet9 [54] (random backgrounds vs. original backgrounds), and CounterAnimal [49] (counter vs. common background). The top triangle of each arrow represents the in-distribution backgrounds and the bottom triangle represents the out-of-distribution ones. We follow ImageNet9 and CounterAnimal and assess the background robustness in terms of the accuracy gap when evaluating a model on images of normal background distribution compared to out-of-distribution backgrounds (length of each arrow; Δ). Crucially, *ForAug* improves the background robustness of all models and across datasets, reducing the background-gap by boosting the performance on the out-of-background-distribution samples more than the in-distribution ones. These findings highlight the generalization benefits of *ForAug* to unusual image compositions.

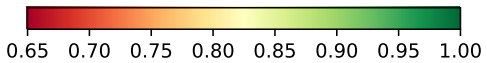
Foreground Focus. Leveraging our inherent knowledge of the foreground masks when using *ForAug*, as well as common XAI techniques [3, 38, 42], we can evaluate a model's focus on the foreground object. We can directly evaluate ImageNet-trained models, but this technique can also be extended to other datasets without relying on manually annotated foreground masks. To evaluate the foreground focus, we employ Grad-CAM [38], Grad-CAM++ [3] and IntegratedGradients (IG) [42] to compute the per-pixel importance of an image for the model's prediction. The foreground focus is defined to be the ratio of the foreground's relative importance to its relative size in the image:

$$\text{FG Focus(img)} = \frac{\text{Area(img) Importance(fg)}}{\text{Area(fg) Importance(img)}} \qquad (2)$$

If all pixels uniformly receive the same importance value, the foreground focus is one. The foreground focus of a model is its average focus over all test images. Figure 4 presents our findings. Using *ForAug* significantly increases the foreground focus of ViT, DeiT and ResNet across all XAI metrics. We hypothesize Swin's below-uniform foreground focus with GradCam is due to its specific implementation.

Table 8. Accuracy relative to the center accuracy of multiple instantiations of the models when the foreground objects is in different cells of a 3×3 grid. We calculate center bias according to Equation (3). Using *ForAug* significantly reduces models' center bias.

Model	Center Bias [%] when trained w/o *ForAug*	Center Bias [%] when trained w/ *ForAug*	Delta
ViT-S	25.5 ± 0.8	22.0 ± 0.3	-3.5
ViT-B	25.4 ± 0.4	19.0 ± 0.2	-6.4
ViT-L	24.3 ± 1.1	11.7 ± 0.7	-12.6
DeiT-S	20.4 ± 0.2	21.2 ± 0.1	$+0.8$
DeiT-B	19.0 ± 0.7	19.0 ± 0.2	± 0.0
DeiT-L	21.2 ± 0.2	18.0 ± 0.2	-3.2
Swin-Ti	25.0 ± 0.7	16.5 ± 0.2	-8.5
Swin-S	23.2 ± 0.1	15.6 ± 0.2	-7.6
ResNet50	26.3 ± 0.3	19.7 ± 0.3	-6.6
ResNet101	23.0 ± 0.3	19.9 ± 0.2	-3.1



Center Bias. With *ForAug* we have unique control over the position of the foreground object in the image. This lets us quantify the center bias of models trained with and

CVPR
#4792

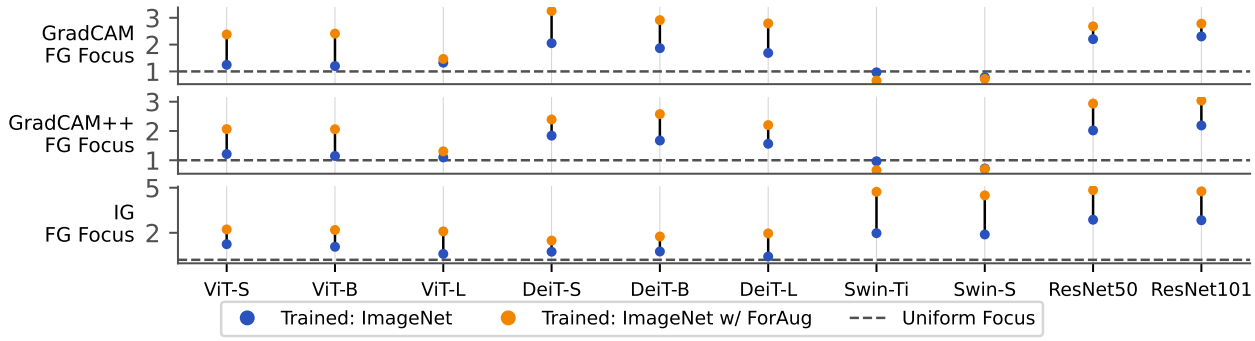


Figure 4. Evaluation of the foreground focus (Equation (2)) using GradCam, GradCam++ and IntegratedGradients (IG) of models trained on ImageNet. Training with *ForAug* improves the foreground focus of almost all models.

without *ForAug*. We divide the image into a 3×3 grid and evaluate model accuracy when the (scaled-down) foreground object is in each of the 9 grid cells. Each cell's accuracy is divided by the accuracy in the center cell for normalization, which gives us the relative performance drop when the foreground is in each part of the image. The center bias is calculated as one minus the average of the minimum performance of a corner cell and the minimum performance of a side cell:

$$\text{Center Bias} = 1 - \frac{\min_{c \in \text{sides}} \text{Acc}(c) + \min_{c \in \text{corners}} \text{Acc}(c)}{2\text{Acc}(c_{\text{center}})} \quad (3)$$

Table 8 visualizes the center bias of three instantiations of each model. Performance is generally highest in the center and lowest in the four corners. Interestingly, ImageNet-trained models perform slightly better when the foreground object is on the right side of the image, compared to the left side, despite our use of random flipping with a probability of 0.5 during training. Using *ForAug* significantly reduces center bias across models, with a more uniform performance especially across the middle row. Thus, *ForAug* makes the model recognize objects across a wider spatial distribution, counteracting the center-bias of ImageNet.

Size Bias. Finally, we evaluate the impact of different sized foreground objects on the accuracy. For this evaluation, we use the *mean* foreground size strategy. We introduce a size factor f_{size} by which we additionally scale the foreground object before pasting it onto the background. Results are normalized by the accuracy when using $f_{\text{size}} = 1.0$. Figure 5 shows the size bias curves of models trained with and without *ForAug*. Models trained using *ForAug* maintain perform better, especially with smaller foreground objects. Therefore, *ForAug*-training improves robustness to variations in object scale, especially for larger models.

5. Discussion & Conclusion

We introduce *ForAug*, a novel data augmentation scheme that facilitates improved Transformer training for image classification. By explicitly separating and recombining foreground objects and backgrounds, *ForAug* enables controlled data augmentation beyond existing image compositions, leading to significant performance gains on ImageNet and downstream fine-grained classification tasks. Furthermore, *ForAug* provides a powerful framework for analyzing model behavior and quantifying biases, including background robustness, foreground focus, center bias, and size bias. Our experiments demonstrate that training using *ForAug* not only boosts accuracy but also significantly reduces these biases, resulting in more robust and generalizable models. In the future, we see *ForAug* be also applied to other datasets and tasks, like video recognition or segmentation. *ForAug*'s ability to both improve performance and provide insights into model behavior makes it a valuable tool for advancing CV research and developing more reliable AI systems.

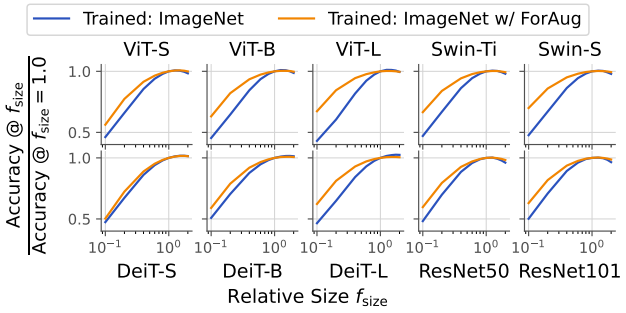


Figure 5. Evaluation of the size bias of models trained on ImageNet. We plot the accuracy relative to the accuracy when using the default size ($f_{\text{size}} = 1.0$).

Acknowledgements

Will be in the final paper.

References

[1] G.E. Bates. Joint distributions of time intervals for the occurrence of successive accidents in a generalized polya urn

scheme. *Annals of Mathematical Statistics*, 26:705–720, 1955. 5

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. 2020. 1

[3] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 839–847, 2018. 7

[4] Ekin D. Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V. Le. Autoaugment: Learning augmentation policies from data. 2018. 2

[5] Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. Randaugment: Practical automated data augmentation with a reduced search space. 2019. 2

[6] Afshin Dehghan, Syed Zain Masood, Guang Shu, and Enrique. G. Ortiz. View independent vehicle make, model and color recognition using convolutional neural network. 2017. 6

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009. 1

[8] Peijian Ding, Davit Soselia, Thomas Armstrong, Jiahao Su, and Furong Huang. Reviving shift equivariance in vision transformers. 2023. 1

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. 1, 5

[10] Debidatta Dwibedi, Ishan Misra, and Martial Hebert. Cut, paste and learn: Surprisingly easy synthesis for instance detection. 2017. 2

[11] Yunhao Ge, Jiashu Xu, Brian Nlong Zhao, Neel Joshi, Laurent Itti, and Vibhav Vineet. Beyond generation: Harnessing text to image models for object detection and segmentation. *ArXiv*, abs/2309.05956, 2023. 2, 6

[12] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. 2018. 3

[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. 2020. 2, 6

[14] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. 2013. 1

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1, 5

[16] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. 2017. 1

[17] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. 2019. 2

[18] Stefan Hinterstoisser, Olivier Pauly, Hauke Heibel, Marek Martina, and Martin Bokeloh. An annotation saved is an annotation earned: Using fully synthetic training for object detection. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 2787–2796, 2019. 2

[19] Ji-Soo Kang and Kyungyong Chung. Staug: Copy-paste based image augmentation technique using salient target. 10: 123605–123613. 2

[20] Parneet Kaur, Karan Sikka, and Ajay Divakaran. Combining weakly and webly supervised learning for classifying food images. 2017. 6

[21] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *ACM Computing Surveys*, 54(10s):1–41, 2022. 1

[22] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. 2023. 3

[23] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2012. 1

[24] Yann Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015. 4

[25] Xiaodan Li, Yuefeng Chen, Yao Zhu, Shuhui Wang, Rong Zhang, and Hui Xue. Imagenet-e: Benchmarking neural network robustness via attribute editing. 2023. 2

[26] Evan Ling, Dezhao Huang, and Minhoe Hur. Humans need not label more humans: Occlusion copy & paste for occluded human instance segmentation. 2022. 2

[27] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. 2023. 3

[28] Yue Liu, Christos Matsoukas, Fredrik Strand, Hossein Azizpour, and Kevin Smith. Patchdropout: Economizing vision transformers using patch dropout. 2022. 2

[29] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9992–10002, Los Alamitos, CA, USA, 2021. IEEE Computer Society. 1, 5

[30] S. Maji, J. Kannala, E. Rahtu, M. Blaschko, and A. Vedaldi. Fine-grained visual classification of aircraft. Technical report, 2013. 6

[31] Tobias Christian Nauen, Sebastian Palacio, and Andreas Dengel. Which transformer to favor: A comparative analysis of efficiency in vision transformers. In *Proceedings of the Winter Conference on Applications of Computer Vision (WACV)*, pages 6955–6966, 2025. 5

[32] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing*, 2008. 6

[33] Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012. 6

[34] Gabriela Rangel, Juan C. Cuevas-Tello, Jose Nunez-Varela, Cesar Puente, and Alejandra G. Silva-Trujillo. A survey on convolutional neural networks and their performance limitations in image recognition tasks. 2024(1), 2024. 1

[35] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, Zhaoyang Zeng, Hao Zhang, Feng Li, Jie Yang, Hongyang Li, Qing Jiang, and Lei Zhang. Grounded sam: Assembling open-world models for diverse visual tasks. 2024. 2, 3

[36] Renan A. Rojas-Gomez, Teck-Yian Lim, Minh N. Do, and Raymond A. Yeh. Making vision transformers truly shift-equivariant. 2023. 1

[37] Edward Sanderson and Bogdan J. Matuszewski. *FCN-Transformer Feature Fusion for Polyp Segmentation*, pages 892–907. Springer International Publishing, 2022. 1

[38] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. 128(2):336–359, 2016. 7

[39] Ang Jia Ning Shermaine, Michalis Lazarou, and Tania Stathaki. Image compositing is all you need for data augmentation. 2025. 2, 6

[40] Connor Shorten and Taghi M. Khoshgoftaar. A survey on image data augmentation for deep learning. 6(1), 2019. 1, 2

[41] Wenhao Sun, Benlei Cui, Xue-Mei Dong, and Jingqun Tang. Attentive eraser: Unleashing diffusion model's object removal potential via self-attention redirection guidance. 2024. 2, 3, 4

[42] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. 2017. 7

[43] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. 2021. 2, 3, 4, 5

[44] Ryo Takahashi, Takashi Matsubara, and Kuniaki Uehara. Data augmentation using random image cropping and patching for deep cnns. 30(9):2917–2931, 2018. 2

[45] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training data-efficient image transformers & distillation through attention. In *Proceedings of the 38th International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021. 1, 5

[46] Hugo Touvron, Matthieu Cord, and Hervé Jégou. Deit iii: Revenge of the vit. In *Computer Vision – ECCV 2022*, pages 516–533, Cham, 2022. Springer Nature Switzerland. 1, 2, 4, 5

[47] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017. 1

[48] Ioannis A. Vezakis, Konstantinos Georgas, Dimitrios Fotiadis, and George K. Matsopoulos. Effisegnet: Gastrointestinal polyp segmentation through a pre-trained efficientnet-based network with a simplified decoder. 2024. 1

[49] Qizhou Wang, Yong Lin, Yongqiang Chen, Ludwig Schmidt, Bo Han, and Tong Zhang. A sober look at the robustness of CLIPs to spurious features. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 7

[50] Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, and Furu Wei. Image as a foreign language: Beit pretraining for all vision and vision-language tasks. 2022. 1

[51] Wenhai Wang, Jifeng Dai, Zhe Chen, Zhenhang Huang, Zhiqi Li, Xizhou Zhu, Xiaowei Hu, Tong Lu, Lewei Lu, Hongsheng Li, Xiaogang Wang, and Yu Qiao. Internimage: Exploring large-scale vision foundation models with deformable convolutions. 2022. 1

[52] Levi Kassel Michael Werman. Deepaste – inpainting for pasting. 2021. 2

[53] Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *Proceedings of the 39th International Conference on Machine Learning*, pages 23965–23998. PMLR, 2022. 1

[54] Kai Xiao, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. Noise or signal: The role of image backgrounds in object recognition. 2020. 3, 7

[55] Mingle Xu, Sook Yoon, Alvaro Fuentes, and Dong Sun Park. A comprehensive survey of image augmentation techniques for deep learning. 137:109347, 2023. 1, 2

[56] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *Transactions on Machine Learning Research*, 2022. 1

[57] Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Seong Joon Oh, Youngjoon Yoo, and Junsuk Choe. CutMix: Regularization strategy to train strong classifiers with localizable features. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, 2019. 2

[58] Chenshuang Zhang, Fei Pan, Junmo Kim, In So Kweon, and Chengzhi Mao. Imagenet-d: Benchmarking neural network robustness on diffusion synthetic object. 2024. 2

[59] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018. 2

[60] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. 2017. 2

[61] Zhuofan Zong, Guanglu Song, and Yu Liu. Detrs with collaborative hybrid assignments training. 2022. 1