ForAug: Recombining Foregrounds and Backgrounds to Improve Vision Transformer Training with Bias Mitigation

Anonymous ICCV submission

Paper ID 6426

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 that limit their robustness and generalizability. This paper introduces ForAug, a novel data augmentation scheme that addresses these challenges and explicitly includes inductive biases, which commonly are part of the neural network architecture, into the training data. ForAug is constructed by using pretrained foundation models to separate and recombine foreground objects with different backgrounds, enabling fine-grained control over image composition during training. It thus increases the data diversity and effective number of training samples. We demonstrate that training on ForNet, the application of ForAug to ImageNet, significantly improves the accuracy of ViTs and other architectures by up to 4.5 percentage points (p.p.) on ImageNet and 7.3 p.p. on downstream tasks. Importantly, ForAug enables novel ways of analyzing model behavior and quantifying biases. Namely, we introduce metrics for background robustness, foreground focus, center bias, and size bias and show that training on ForNet substantially reduces these biases compared to training on ImageNet. 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>`.

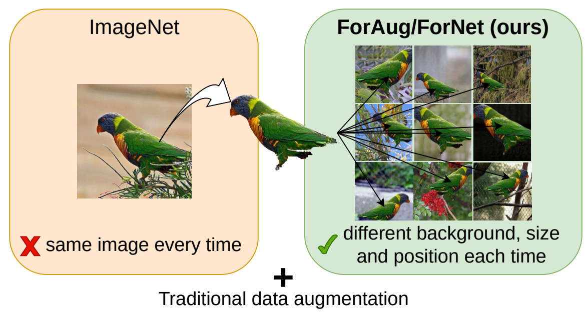

Figure 1. Comparison of *ForNet* and ImageNet. *ForNet* recombines foreground objects with different backgrounds each epoch, thus creating a more diverse training set. We still apply traditional data augmentation afterwards.

1. Introduction

Image classification, a fundamental task in computer vision (CV), involves assigning a label to an image from a predefined set of categories. This seemingly simple task underpins a wide range of applications, including medical diagnosis [39, 50], autonomous driving [52], and object recognition [3, 15, 17]. Furthermore, image classification is used for large-scale pretraining of vision models [10, 31, 47] and to judge the progress of the field of CV [22, 36]. The advent of large-scale datasets, particularly ImageNet [8], containing millions of labeled images across thousands of categories, has been instrumental in driving significant progress in this field. ImageNet served as a catalyst for the rise of large-scale CV models [16, 25] and remains the most important CV benchmark for more than a decade [16, 25, 48, 54].

While traditionally, convolutional neural networks (CNNs) have been the go-to architecture for image classification, Transformers [49], particularly the Vision Transformer (ViT) [10], have emerged as a powerful alternative. These attention-based models have demonstrated superior performance in various vision tasks, including image classification [3, 51, 54, 57, 62].

Data augmentation is a key technique for training image classification models. Traditional data augmentation methods, such as random cropping, flipping, and color jittering, are commonly employed to increase the diversity of the training data and improve the model's performance [42, 56]. These basic transformations, originally designed for CNNs, change the input images in a way that preserves their semantic meaning [1]. However, the architectural differences of CNNs and Transformers suggest that the latter might benefit from different data augmentation strategies. In particular, the Transformers self-attention mechanism is not translation equivariant [9, 38], meaning that the model does not inherently understand the spatial relationships between pixels.

Inspired by this inductive bias of CNNs, that is not inherent to ViTs, we propose *ForAug*, a novel data augmentation scheme for image classification which makes the translation equivariance of CNNs explicit in the training data by recombining foreground objects at varying positions with different backgrounds. Applying *ForAug* to ImageNet gives rise to *ForNet*, a novel dataset that enables this data augmentation with with fine-grained control over the image composition. Recognizing that Transformers need to learn the spatial relationships from data, since they are not inherently translation invariant, and in general are usually trained on larger datasets [24], we separate the foreground objects in ImageNet from their backgrounds, using an open-world object detector [37], and fill in the background in a plausible way using an object removal model [43, 45]. This allows us to recombine any foreground object with any background on the fly, creating a highly diverse training set. During recombination, we can control important parameters, like the size and position of the foreground object, to help the model learn the spatial invariances necessary for image classification. We show that training on *ForNet* instead of ImageNet increases the model accuracy of Transformers by up to 4.5 p.p. on ImageNet and an up to 39.3% reduction in error rate on downstream tasks.

Additionally, *ForAug* is a useful tool for analyzing model behavior and biases, when used during the evaluation phase. We utilize our control over the image distribution to quantify 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 the object's position), and size bias (by controlling object size). These analyses provide insights into model behavior and biases, which is crucial for model deployment and future robustness optimizations. We show that training on *ForNet*, instead of ImageNet, significantly reduces all of these biases, completely removing the models' dependence on the background distribution. We make our code for *ForAug* and the *ForNet*-dataset publicly available[1] to facilitate further research.

Contributions

- We propose *ForAug*, a novel data augmentation scheme, that recombines objects and backgrounds to train Transformers for image classification.
- We show that training on *ForNet*, the ImageNet instantiation of *ForAug*, leads to 4.5 p.p. improved accuracy on ImageNet and 7.3 p.p. on downstream tasks.
- We propose novel *ForAug*-based metrics to analyze and quantify fine-grained biases trained models: Background Robustness, Foreground Focus, Center Bias, and Size Bias. Training on *ForNet*, instead of ImageNet, significantly reduces these biases.

[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 [61] to increase the diversity of the training data without changing their semantic meaning. With the advent of Transformers, new data augmentation operations like PatchDropout [30] have been proposed. Other transformations like Mixup [60], CutMix [58], or random cropping and patching [46] combine multiple input images. These simple transformations are usually bundled to form more complex augmentation policies like AutoAugment [5] and RandAugment [6], which automatically search for optimal augmentation policies or 3-augment [48] which is optimized to train a ViT. For a general overview of data augmentation techniques for image classification, we refer to [42, 56].

We build upon these general augmentation techniques by introducing a novel approach to explicitly separate and recombine foregrounds and backgrounds for image classification. Our approach is used in tandem with traditional data augmentation techniques to improve model performance and reduce biases.

Copy-Paste Augmentation The copy-paste augmentation [14], which is used for object detection [14, 41] and instance segmentation [28, 53], 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 [19] and pretrained object-detection models for use on objects on white background [11] or synthetic images [12]. DeePaste [53] focuses on using inpainting for a more seamless integration of the pasted object.

Unlike these methods, *ForNet* focuses on image classification. While for detection and segmentation, objects are pasted onto another image (with a different foreground) or on available or rendered background images of the target scene, we extract foreground objects and fill in the resulting holes in the background in a semantically neutral way. This way, we can recombine any foreground object with a large variety of neutral backgrounds from natural images, enabling a controlled and diverse manipulation of image composition.

Model robustness evaluation Evaluating model robustness to various image variations is critical for understanding and improving model generalization. Datasets like ImageNet-C [18] and ImageNet-P [18] introduce common corruptions and perturbations. ImageNet-E [27] evaluates model robustness against a collection of distribution shifts. Other datasets, such as ImageNet-D [59], focus on varying background, texture, and material, but rely on synthetic data. Stylized ImageNet [13] investigates the impact of texture changes. ImageNet-9 [55] explores background variations using segmented images, but the backgrounds are often artificial.

In contrast to these existing datasets, which are used only for evaluation, *ForNet* 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 *ForNet* also provides controllable training set generation, it goes beyond simply measuring robustness to actively improving it through training.

3. RecombiNet (Method)

We introduce *ForAug*, a data augmentation scheme designed to enhance Transformer training by explicitly separating and recombining foreground objects and backgrounds. *ForAug* involves two stages: Segmentation and Recombination, both visualized in Figure 2.

Segmentation

The segmentation stage isolates the foreground objects and their corresponding backgrounds. We then fill in the background in a visually plausible way [43] using a pretrained object-removal model. 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 [37], which is based on Grounding DINO [29] and SAM [23]. 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 up to n foreground masks for each dataset-image, using different more and more general prompts based on the more general synsets of WordNet (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.

An inpainting model that is specifically optimized to remove objects from images, such as LaMa [45] or Attentive Eraser [43], is used to inpaint the foreground regions in the backgrounds. To ensure the quality of the foreground and background images (for each dataset-image), we select a foreground/background pair from the $\leq n$ variants we have extracted and infilled in the previous steps. Using an ensemble 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 according to:

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

Here, E is the ensemble of models and m is a pretrained model, c is the correct foreground class, fg, and bg are the foreground and background and $\text{size}(\cdot)$ is the size in number of pixels. We ran a hyperparameter search using a manually annotated subset of foreground/background variants to find the factors in Equation (1): $\lambda = 2$ and $\varepsilon = 0.1$. The *optimal foreground size* of 10% of the full image balances the smallest possible foreground size that encompasses all the respective class information in the image with still conveying the foreground information after pasting it onto another background. This filtering step ensures we segment all the relevant foreground objects.

Finally, we filter out backgrounds that are more than 80% infilled, as these tend to be overly synthetic, plain and don't carry much information (see the supplementary material). We ablate this choice in Section 4.1. 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 them as data augmentation before applying common data augmentation operations during training.

Recombination

The recombination stage, which is performed online, 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

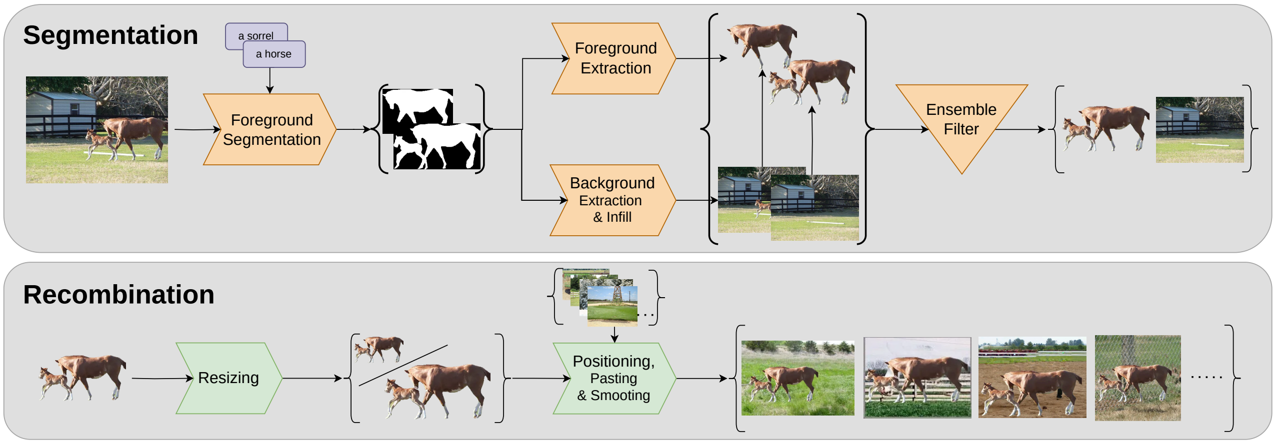

Figure 2. Overview of *ForNet*. The data creation consists of two stages: (1, offline) Segmentation, where we segment the foreground objects from the background and fill in the background. (2, online) Recombination, where we combine the foreground objects with different backgrounds to create new samples.

fitting size, place it in the background image, smooth the transition edge, and apply other data augmentations.

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:

$$s \sim \mathcal{U}\left[(1-0.3)s_l, (1+0.3)s_u\right]. \quad (2)$$

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. This position is sampled from a generalization of the Bates distribution [2] with parameter $\eta \in \mathbb{N}$, visualized in Figure 3. We choose the bates distribution, as it presents an easy way to sample from a bounded domain with just one hyperparameter that controls the concentration of the distribution. $\eta = 1$ corresponds to the uniform distribution; $\eta > 1$ concentrates the distribution around the center; and for $\eta < -1$, the distribution is concentrated at the borders. To more seamlessly integrate the foreground, we apply a Gaussian blur with $\sigma \in [\frac{\sigma_{\text{max}}}{10}, \sigma_{\text{max}}]$, inspired by the standard range for the Gaussian blur operation in [48], to the foreground's alpha-mask.

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 second mode ensures the foreground object remains fully visible, while the first mode mirrors standard data augmentation practices.

We experiment with a constant mixing ratio, or a linear or cosine anealing 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*. Thus, we still ensure each foreground is seen once per epoch.

4. Experiments

We conduct a comprehensive suit of experiments to validate the effectiveness of our approach. We compare training on *ForNet*, the ImageNet instantiation of *ForAug*, to training on ImageNet for 7 different models. Furthermore, we assess the impact of using *ForNet* for pretraining on multiple fine-grained downstream datasets. Additionally, we use *ForAug*'s control over the image distribution to quantify some model behaviors and biases.

4.1. Design Choices of *ForAug*

We start by ablating the design choices of *ForAug*. For this, we revert to TinyImageNet [26], a subset of ImageNet containing 200 categories with 500 images each, and Tiny*ForNet*, a version of *ForAug* derived from TinyImageNet. Table 1 presents the results of these ablations.

Dataset	Detect. prompt	Infill Model	FG. size	Augmentation Order	BG. strategy	BG. pruning	edge smoothing	original image mixing	TinyImageNet Accuracy ViT-Ti [%]	ViT-S [%]
TinyImageNet									66.1 ± 0.5	68.3 ± 0.7
Tiny*ForNet*	specific	LaMa [45]	mean	crop→paste→color	same	-	-	-	64.6 ± 0.5	70.0 ± 0.6
Tiny*ForNet*	specific	LaMa [45]	range	crop→paste→color	same	-	-	-	65.5 ± 0.4	71.2 ± 0.5
Tiny*ForNet*	general	LaMa [45]	range	crop→paste→color	same	-	-	-	66.4 ± 0.6	72.9 ± 0.6
Tiny*ForNet*	general	Att. Eraser [43]	range	crop→paste→color	same	-	-	-	67.5 ± 1.2	72.4 ± 0.5
Tiny*ForNet*	general	Att. Eraser [43]	range	paste→crop→color	same	-	-	-	67.1 ± 1.2	72.9 ± 0.5
Tiny*ForNet*	general	Att. Eraser [43]	range	paste→crop→color	same	1.0	-	-	67.0 ± 1.2	73.0 ± 0.3
Tiny*ForNet*	general	Att. Eraser [43]	range	paste→crop→color	same	0.8	-	-	67.2 ± 1.2	72.9 ± 0.8
Tiny*ForNet*	general	Att. Eraser [43]	range	paste→crop→color	same	0.6	-	-	67.5 ± 1.0	72.8 ± 0.7
Tiny*ForNet*	general	Att. Eraser [43]	range	paste→crop→color	same	0.8	$\sigma_{\max} = 2.0$	-	67.2 ± 0.4	72.9 ± 0.5
Tiny*ForNet*	general	Att. Eraser [43]	range	paste→crop→color	same	0.8	$\sigma_{\max} = 4.0$	-	65.9 ± 0.5	72.4 ± 0.6
Tiny*ForNet*	general	Att. Eraser [43]	range	paste→crop→color	same	0.8	-	$p = 0.2$	69.8 ± 0.5	75.0 ± 0.3
Tiny*ForNet*	general	Att. Eraser [43]	range	paste→crop→color	same	0.8	-	$p = 0.33$	69.5 ± 0.4	75.2 ± 1.0
Tiny*ForNet*	general	Att. Eraser [43]	range	paste→crop→color	same	0.8	-	$p = 0.5$	70.3 ± 1.0	74.2 ± 0.2
Tiny*ForNet*	general	Att. Eraser [43]	range	paste→crop→color	same	0.8	-	linear	70.1 ± 0.7	74.9 ± 0.8
Tiny*ForNet*	general	Att. Eraser [43]	range	paste→crop→color	same	0.8	-	reverse lin.	67.6 ± 0.2	73.2 ± 0.3
Tiny*ForNet*	general	Att. Eraser [43]	range	paste→crop→color	same	0.8	-	cos	71.3 ± 1.0	75.7 ± 0.8
Tiny*ForNet*	general	Att. Eraser [43]	range	paste→crop→color	same	0.8	$\sigma_{\max} = 4.0$	cos	70.0 ± 0.8	75.5 ± 0.7
Tiny*ForNet*	general	Att. Eraser [43]	range	paste→crop→color	orig.	0.8	$\sigma_{\max} = 4.0$	cos	67.2 ± 0.9	69.9 ± 1.0
Tiny*ForNet*	general	Att. Eraser [43]	range	paste→crop→color	all	0.8	$\sigma_{\max} = 4.0$	cos	70.1 ± 0.7	77.5 ± 0.6
ForNet	general	Att. Eraser [43]	range	paste→crop→color	same	0.8	-	cos	-	80.5 ± 0.1
ForNet	general	Att. Eraser [43]	range	paste→crop→color	same	0.8	$\sigma_{\max} = 4.0$	cos	-	80.7 ± 0.1
ForNet	general	Att. Eraser [43]	range	paste→crop→color	all	0.8	$\sigma_{\max} = 4.0$	cos	-	81.3 ± 0.1

Table 1. Ablation of design decisions of Tiny*ForNet* on TinyImageNet and *ForNet* on ImageNet.

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. Attentive Eraser [43] produces superior results compared to LaMa [45] (see the supplementary for examples).

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*) slightly improves ViT-S's performance compared to applying crop-related augmentations before pasting (*crop→paste→color*). For ViT-Ti, the results are ambiguous.

Background pruning. When it comes to the choice of backgrounds to use, we test two pruning thresholds (t_{prune}) to exclude backgrounds with excessive 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. Therefore, we choose $t_{\text{prune}} = 0.8$ to exclude predominantly artificial backgrounds. Similarly, applying edge smoothing to foreground masks with Gaussian blurring actually hurts performance on Tiny*ForNet*, but slightly improves it on *ForNet*.

Mixing *ForNet* 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 using Tiny*ForNet*, the cosine annealing schedule yields the best results, boosting accuracy by another $0.5-1$ p.p.

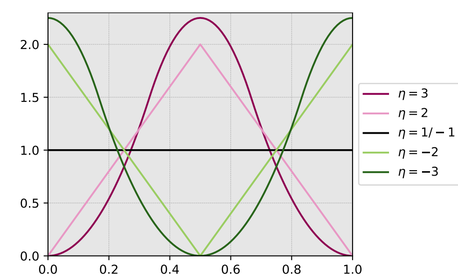

Figure 3. Plot of the probability distribution function (PDF) of the extended Bates distribution for different parameters η. Higher values of η concentrate the distribution around the center.

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) version of *ForNet*.

Foreground position. Finally, we analyze the foreground object's positioning in the image. We utilize an extended Bates distribution to sample the position of the foreground

Training Set/ Bates Parameter	TIN	Tiny*ForNet*				
		$\eta = -3$	-2	$1/-1$	2	3
TinyImageNet	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 2. Accuracy of ViT-S trained on TinyImageNet (TIN) and Tiny*ForNet* with different foreground position distributions by varying the parameter of a Bates distribution η. The best performance is achieved using the uniform distribution ($\eta = 1$).

Dataset	Classes	Training Images	Validation Images
TinyImageNet	200	100,000	10,000
Tiny*ForNet*	200	99,404	9,915
ImageNet	1,000	1,281,167	50,000
ForNet	1,000	1,274,557	49,751

Table 3. Dataset statistics for TinyImageNet, Tiny*ForNet*, ImageNet, and *ForNet*. For *ForNet* and Tiny*ForNet* we report the number of foreground/background pairs.

object. The Bates distribution [2] with parameter $\eta \geq 1$ is the mean of η independent uniformly distributed random variables [20]. Therefore, the larger η, the more concentrated the distribution is around the center. We extend this concept to $\eta \leq -1$ by defining $X \sim \text{Bates}(\eta) :\Leftrightarrow s(X) \sim \text{Bates}(-\eta)$ for $\eta \leq 1$ with s being the sawtooth function on $[0, 1]$:

$$s(x) = \begin{cases} x + 0.5 & \text{if } 0 < x < 0.5 \\ x - 0.5 & \text{if } 0.5 \leq x \leq 1 \end{cases} \quad (3)$$

Note that $s \circ s = \text{id}$ on $[0, 1]$. This way, distributions with $\eta \leq -1$ are more concentrated around the borders. $\eta = 1$ and $\eta = -1$ both correspond to the uniform distribution. The PDF of this extended Bates distribution is visualized in Figure 3.

When sampling more towards the center of the image, the difficulty of the task is reduced, which then reduces the performance on TinyImageNet. This is reflected in the performance when evaluating on Tiny*ForNet* with $\eta = 2$ and $\eta = 3$ compared to $\eta = -1/1$. We observe a similar reduction for $\eta < -1$. This experiment is conducted using the LaMa infill model.

After fixing the optimal design parameters in Table 1 (last row), we construct the full *ForNet* dataset using the entire ImageNet dataset. Table 3 compares the dataset statistics of ImageNet and *ForNet*. The slightly reduced image count in *ForNet* is due to instances where Grounded SAM failed to produce valid object detections.

Model	ImageNet Accuracy when trained on		Delta
	ImageNet	*ForNet*	
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
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 4. ImageNet results of models trained on *ForNet* and on ImageNet directly. *ForNet* improves the performance of all models in our test.

4.2. Image Classification Results

Table 4 compares the ImageNet performance of models trained on *ForNet* and ones trained directly on ImageNet. We adopt the training setup of [33] and [48] (details in the supplementary material) for training ViT [10], Swin [31] and ResNet [16] models. Notably, *ForNet* improves performance across all tested architectures, including the ResNet models (up to 1 p.p.), demonstrating benefits beyond Transformers. For Transformer models, we observe improvements from 1.2 p.p. to 4.5 p.p. This improvement is more substantial for the larger models, with ViT-L gaining 4.5 p.p. in accuracy. *ForNet*'s improvements mostly counteract the drop in performance due to overfitting for large models. When training on ImageNet, this drop is 3.8 p.p. from ViT-S to ViT-L, while for *ForNet* it is reduced to 1.6 p.p.

To assess the transferability of *ForNet*-trained models, we finetune models pretrained on ImageNet and *ForNet* on five fine-grained datasets: FGVC-Aircraft [32], Stanford Cars [7], Oxford Flowers [34], Food-101 [21], and Oxford-IIIT Pets [35]. While for ResNets, the performance of both training datasets is about the same, for every Transformer, we see the accuracy improve on all downstream dataset by up to 7.3 p.p. and a reduction of error rate of up to 39.3%. In summary, these results demonstrate that the improved representation learning achieved by training on *ForNet* translates to superior performance not only on ImageNet, but also on a variety of fine-grained image classification tasks.

4.3. Further Model Evaluation

Beyond its use for training, *ForNet*'s unique properties and controlled data generation capabilities make it a powerful tool for analyzing model behavior and biases.

Background Robustness We assess the robustness of models to shifts in the background distribution from a class-

Model	Aircraft	Cars	Flowers	Food	Pets
ViT-S @ ImageNet	72.4 ± 1.0	89.8 ± 0.3	94.5 ± 0.2	89.1 ± 0.1	93.8 ± 0.2
ViT-S @ *ForNet*	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 @ ImageNet	71.7 ± 0.5	90.0 ± 0.2	94.8 ± 0.4	89.8 ± 0.2	94.1 ± 0.4
ViT-B @ *ForNet*	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 @ ImageNet	72.1 ± 1.0	88.8 ± 0.3	94.4 ± 0.3	90.1 ± 0.2	94.2 ± 0.4
ViT-L @ *ForNet*	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
Swin-Ti @ ImageNet	77.0 ± 0.1	91.3 ± 0.6	95.9 ± 0.1	90.0 ± 0.2	94.2 ± 0.1
Swin-Ti @ *ForNet*	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 @ ImageNet	75.7 ± 1.4	91.0 ± 0.3	95.9 ± 0.5	91.1 ± 0.2	94.4 ± 0.1
Swin-S @ *ForNet*	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 @ ImageNet	78.2 ± 0.5	89.8 ± 0.2	91.7 ± 0.4	84.4 ± 0.2	93.7 ± 0.3
ResNet-50 @ *ForNet*	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	+0.1	± 0
ResNet-101 @ ImageNet	78.4 ± 0.6	90.3 ± 0.1	91.2 ± 0.5	86.0 ± 0.2	94.3 ± 0.2
ResNet-101 @ *ForNet*	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

Table 5. Downstream accuracy in percent when finetuning on other datasets. Models were pretrained on *ForNet* and ImageNet. Pretraining on *ForNet* increases Transformer downstream accuracy on all datasets.

Model	Background Robustness when trained on		Delta
	ImageNet	*ForNet*	
ViT-S	0.73 ± 0.01	0.99 ± 0.01	+0.26
ViT-B	0.72 ± 0.01	1.00 ± 0.01	+0.28
ViT-L	0.70 ± 0.01	1.00 ± 0.01	+0.30
Swin-Ti	0.72 ± 0.01	1.00 ± 0.01	+0.28
Swin-S	0.72 ± 0.01	1.00 ± 0.01	+0.28
ResNet-50	0.79 ± 0.01	0.99 ± 0.01	+0.20
ResNet-101	0.79 ± 0.01	1.00 ± 0.01	+0.21

Table 6. Evaluation of the background robustness of models trained on *ForNet* and on ImageNet directly. Training on *ForNet* improves the background robustness of all model to ≈ 1.00, meaning the model is indifferent to the choice of background.

related background to any background. Background robustness is defined to be the ratio of accuracy on *ForNet* with same-class backgrounds to accuracy with any background:

$$\text{Background Robustness} = \frac{\text{Acc}(\textit{ForNet}_{\text{all}})}{\text{Acc}(\textit{ForNet}_{\text{same}})} \quad (4)$$

It represents the relative drop in performance under a background distribution shift. Table 6 presents the background robustness of various models. When trained on ImageNet, smaller models generally exhibit greater robustness to changes in the background distribution than larger models and ResNet is more robust than the tested Transformer models. Crucially, training on *ForNet* instead of ImageNet improves the background robustness of all models to ≈ 1.00, meaning that these models are agnostic to the choice of background and only classify based on the foreground. These findings highlight the generalization benefits of *ForNet*.

Model	Foreground Focus when trained on					
	IN	FN	IN	FN	IN	FN
	GradCam		GradCam++		IG	
ViT-S	1.2 ± 0.1	2.3 ± 0.3	1.2 ± 0.1	2.1 ± 0.4	1.9 ± 0.1	2.7 ± 0.1
ViT-B	1.2 ± 0.1	2.4 ± 0.7	1.1 ± 0.1	2.1 ± 0.1	1.7 ± 0.1	2.7 ± 0.1
ViT-L	1.3 ± 0.1	1.6 ± 0.1	1.1 ± 0.1	1.3 ± 0.1	1.3 ± 0.1	2.6 ± 0.1
Swin-Ti	0.9 ± 0.1	0.7 ± 0.1	1.0 ± 0.3	0.7 ± 0.3	2.5 ± 01	4.8 ± 0.3
Swin-S	0.8 ± 0.1	0.7 ± 0.1	0.7 ± 0.1	0.7 ± 0.4	2.4 ± 0.1	4.6 ± 0.3
ResNet-50	2.2 ± 0.1	2.7 ± 0.1	2.0 ± 0.1	2.9 ± 0.1	3.2 ± 0.1	4.9 ± 0.2
ResNet-101	2.3 ± 0.1	2.8 ± 0.1	2.2 ± 0.1	3.0 ± 0.1	3.2 ± 0.1	4.8 ± 0.1

Table 7. Evaluation of the foreground focus using GradCam, GradCam++ and IntegratedGradients of models trained on *ForNet* (FN) and on ImageNet (IN) directly. Training on *ForNet* improves the foreground focus of almost all models.

Foreground Focus Leveraging our inherent knowledge of the foreground masks when using *ForNet*, as well as common XAI techniques [4, 40, 44], 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 [40], Grad-CAM++ [4] or IntegratedGradients (IG) [44] 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:

$$\text{FG Focus}(\text{img}) = \frac{\text{Area}(\text{img})\ \text{Importance}(\text{fg})}{\text{Area}(\text{fg})\ \text{Importance}(\text{img})} \quad (5)$$

The foreground focus of a model is its average foreground focus over all test images. Table 7 presents our findings. Training on *ForNet* significantly increasees the foreground focus of ViT and ResNet across all metrics used. For Swin, the foreground focus stagnates when measured using GradCam and GradCam++, but almost doubles when using IG.

Center Bias With *ForNet* we have unique control over the position of the foreground object in the image. This lets us quantify the center bias of ImageNet- and *ForNet*-trained models. We divide the image into a 3×3 grid and evaluate model accuracy when the 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

Model	Center Bias when trained on		Delta
	ImageNet	*ForNet*	
ViT-S	0.255 ± 0.008	0.220 ± 003	-0.035
ViT-B	0.254 ± 0.004	0.190 ± 0.002	-0.064
ViT-L	0.243 ± 0.011	0.117 ± 0.007	-0.126
Swin-Ti	0.250 ± 0.007	0.165 ± 0.002	-0.085
Swin-S	0.232 ± 0.001	0.156 ± 002	-0.076
ResNet50	0.263 ± 0.003	0.197 ± 0.003	-0.066
ResNet101	0.230 ± 0.003	0.199 ± 002	-0.031

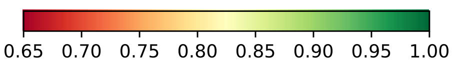

Table 8. Evaluation of the position bias. We plot the accuracy relative to the center accuracy of multiple instantiations of the models when the foreground objects is in different cells a 3×3 grid. Training on *ForNet* significantly reduces a models center bias.

cell and the minimum performance of a side cell:

$$\text{Center Bias} = 1 - \frac{\min\limits_{a,b\in\{0,2\}} \text{Acc}(\text{cell}_{(a,b)}) + \min\limits_{\substack{a=1 \text{ or } b=1 \\ a\neq b}} \text{Acc}(\text{cell}_{(a,b)})}{2\text{Acc}(\text{cell}_{(1,1)})} \tag{6}$$

Table 8 visualizes the center bias of three instantiations of each model. Performance is generally highest in the center and the center top and bottom and center left and right cells, 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. Training on *ForNet* significantly reduces center bias across all models. This demonstrates that *ForNet* promotes a more uniform spatial attention distribution. Their accuracy is higher in the center left and right cells than in the center top and bottom ones, which is not the case for ImageNet-trained models.

Size Bias Finally, we evaluate the impact of different-sized foreground objects on the accuracy. For this evaluation, we

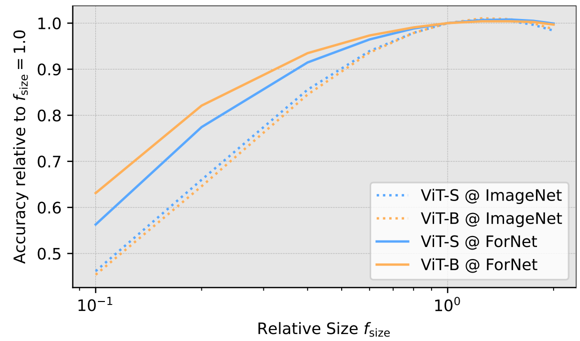

Figure 4. Evaluation of the size bias of models trained on *ForNet*. We plot the accuracy relative to the accuracy when using the mean foreground size.

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 again normalized by the accuracy when using the mean foreground size ($f_{\text{size}} = 1.0$). Figure 4 shows the size bias curves of ViT-S and ViT-B when trained on ImageNet and *ForNet*. Models trained on *ForNet* maintain better performance even with smaller foreground objects, when ImageNet-trained models exhibit a more rapid performance decline. Therefore, *ForNet*-training improves robustness to variations in object scale.

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, 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 on *ForNet*, the instantiation of *ForAug* on ImageNet, 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.

Acknowledgements

Will be in the final paper.

References

[1] Khaled Alomar, Halil Ibrahim Aysel, and Xiaohao Cai. Data augmentation in classification and segmentation: A survey and new strategies. 9(2):46. 1

[2] 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. 4, 6

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

[4] 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

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

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

[7] 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

[8] 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

[9] Peijian Ding, Davit Soselia, Thomas Armstrong, Jiahao Su, and Furong Huang. Reviving shift equivariance in vision transformers. 2023. 2

[10] 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, 6

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

[12] 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

[13] 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

[14] 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

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

[16] 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, 6

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

[18] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. 2019. 3

[19] 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

[20] Norman L. Jonhson, Samuel Kotz, and N. Balakrishnan. *Continuous Univariate Distributions*. Wiley, 2 edition, 1995. Wiley series in probability and mathematical statistics. 6

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

[22] 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

[23] 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

[24] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. In *Computer Vision – ECCV 2020*, pages 491–507, Cham, 2020. Springer International Publishing. 2

[25] 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

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

[27] Xiaodan Li, Yuefeng Chen, Yao Zhu, Shuhui Wang, Rong Zhang, and Hui Xue. Imagenet-e: Benchmarking neural network robustness via attribute editing. 2023. 3

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

[29] 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

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

ICCV #6426

[31] 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, 6

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

[33] Tobias Christian Nauen, Sebastian Palacio, and Andreas Dengel. Which transformer to favor: A comparative analysis of efficiency in vision transformers, 2023. 6

[34] 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

[35] 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

[36] 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). 1

[37] 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

[38] Renan A. Rojas-Gomez, Teck-Yian Lim, Minh N. Do, and Raymond A. Yeh. Making vision transformers truly shift-equivariant. 2023. 2

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

[40] 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

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

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

[43] 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, 5

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

[45] 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, 5

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

[47] 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

[48] 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, 6

[49] 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

[50] 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

[51] 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

[52] 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

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

[54] 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

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

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

[57] 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

[58] 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. 2

[59] Chenshuang Zhang, Fei Pan, Junmo Kim, In So Kweon, and Chengzhi Mao. Imagenet-d: Benchmarking neural network robustness on diffusion synthetic object. 2024. 3

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

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

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