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

Anonymous submission

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>`.

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 (Sanderson and Matuszewski 2022; Vezakis et al. 2024), autonomous driving (Wang et al. 2022b), and object recognition (Carion et al. 2020; He et al. 2017; Girshick et al. 2013) and facilitates large-scale pretraining (Dosovitskiy et al. 2021; Liu et al. 2021; Touvron et al. 2021), and progress evaluation in CV (Khan et al. 2022; Rangel et al. 2024). The advent of large-scale datasets, particularly ImageNet (Deng et al. 2009), served as a catalyst for the rist of large-scale CV models (Krizhevsky, Sutskever, and Hinton 2012; He et al. 2016) and remains the most important CV benchmark for more than a decade (Krizhevsky, Sutskever, and Hinton 2012; Touvron, Cord, and Jégou 2022; Wortsman et al. 2022; He et al. 2016). While traditionally, convolutional neural networks (CNNs) have been the go-to architecture in CV, Transformers (Vaswani et al. 2017), particularly the Vision Transformer (ViT) (Dosovitskiy et al. 2021), have emerged as a powerful alternative, demonstrating superior performance in various vision tasks, including image classification (Wortsman et al. 2022; Yu et al. 2022; Carion et al. 2020; Zong, Song, and Liu 2022; Wang et al. 2022a).

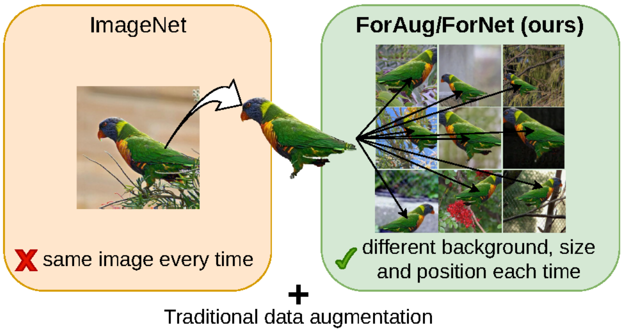

ImageNet
ForAug/ForNet (ours)
✗ same image every time
✓ different background, size and position each time
+
Traditional data augmentation

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.

Data augmentation is a key technique for training image classification models. Traditional data augmentation methods, such as random cropping, flipping, and color changes, are commonly employed to increase the diversity of the training data and improve the model's performance (Xu et al. 2023; Shorten and Khoshgoftaar 2019). These basic transformations, originally designed for CNNs, change the input images in a way that preserves their semantic meaning (Alomar, Aysel, and Cai 2023), but are limited to existing image compositions. 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 Transformers self-attention mechanism, unlike a CNN, is not translation equivariant (Rojas-Gomez et al. 2023; Ding et al. 2023), meaning that the model does not inherently understand the spatial relationships between pixels.

Recognizing that Transformers need to learn the spatial relationships from data and in general are usually trained on larger datasets (Kolesnikov et al. 2020), we propose *ForAug*, a novel data augmentation scheme that recombines foreground objects with different backgrounds. Thus, *ForAug* goes beyond existing image compositions and encodes desired invariances directly into the training data (see Figure 1). 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. We separate the foreground objects in ImageNet from their backgrounds, using an open-world object detector (Ren et al. 2024), and fill in the background in a plausible way using an object removal model (Sun et al. 2024; Suvorov et al. 2021). This allows us to then 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 in 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 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. 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, moving beyond the (possibly biased) image compositions in the dataset while preserving label integrity.
- We show that training ViT 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.

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 (Zhong et al. 2017) to increase the diversity of the training data without changing their semantic meaning. With the advent of Transformers, new data augmentation operations like PatchDropout (Liu et al. 2022) have been proposed. Other transformations like Mixup (Zhang et al. 2018), CutMix (Yun et al. 2019), or random cropping and patching (Takahashi, Matsubara, and Uehara 2018) combine multiple input images. These simple transformations are usually bundled to form more complex augmentation policies like AutoAugment (Cubuk et al. 2018) and RandAugment (Cubuk et al. 2019), or 3-augment (Touvron, Cord, and Jégou 2022) which is optimized to train a ViT. For a general overview of data augmentation techniques for image classification, we refer to (Shorten and Khoshgoftaar 2019; Xu et al. 2023).

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

Copy-Paste Augmentation The copy-paste augmentation (Ghiasi et al. 2020), which is used for object detection (Shermaine, Lazarou, and Stathaki 2025; Ghiasi et al. 2020) and instance segmentation (Werman 2021; Ling, Huang, and Hur 2022), 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 (Hinterstoisser et al. 2019) and pretrained object-detection models for use on objects on white background (Dwibedi, Misra, and Hebert 2017) or synthetic images (Ge et al. 2023). (Kang and Chung) 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.

While these methods paste objects 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 are preserving label integrity while having diverse, neutral backgrounds available for recombination, 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 (Hendrycks and Dietterich 2019) and ImageNet-P (Hendrycks and Dietterich 2019) introduce common corruptions and perturbations. ImageNet-E (Li et al. 2023) evaluates model robustness against a collection of distribution shifts. Other datasets, such as ImageNet-D (Zhang et al. 2024), focus on varying background, texture, and material, but rely on synthetic data. Stylized ImageNet (Geirhos et al. 2018) investigates the impact of texture changes. ImageNet-9 (Xiao et al. 2020) 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 fore-

[1] Link will go here.

ground 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 *ForAug* (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 (Sun et al. 2024) 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 (Ren et al. 2024), which is based on Grounding DINO (Liu et al. 2023) and SAM (Kirillov et al. 2023). 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 (Suvorov et al. 2021) or Attentive Eraser (Sun et al. 2024), 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:

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

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 $\varepsilon = 0.1$. This filtering step ensures we segment all the relevant foreground objects.

Finally, we filter out backgrounds that are largely infilled, as these tend to be overly synthetic and don't carry much information (see the supplementary material). 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.

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 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. This position is sampled from a generalization of the Bates distribution (Bates 1955) with parameter $\eta \in \mathbb{N}$. 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

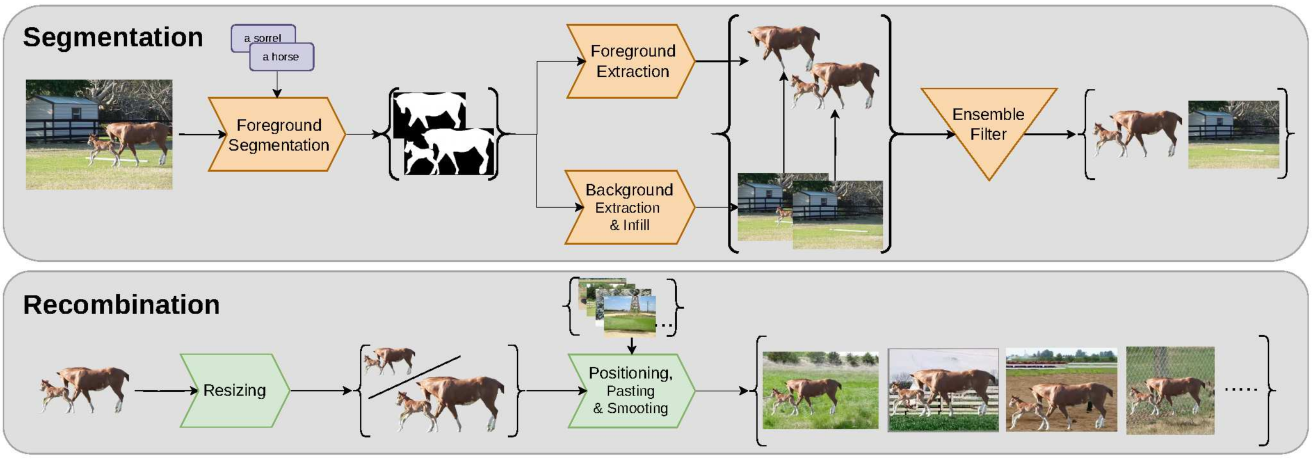


Figure 2: Overview of *ForNet*. The data creation consists of two stages: (1, offline) Segmentation, where we segment the foreground objects from the background and fill in the background. (2, online) Recombination, where we combine the foreground objects with different backgrounds to create new samples.

Gaussian blur with $\sigma \in \left[\frac{\sigma_{\max}}{10}, \sigma_{\max}\right]$, inspired by the standard range for the Gaussian blur operation in (Touvron, Cord, and Jégou 2022), 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 (Le and Yang 2015), 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 ablations for the segmentation phase and Table 2 for the recombination phase.

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*).

Dataset	Detect. Prompt	Infill Model	TinyImageNet Accuracy [%]	
			ViT-Ti	ViT-S
TinyImageNet			66.1 ± 0.5	68.3 ± 0.7
Tiny*ForNet*	specific	LaMa (Suvorov et al. 2021)	65.5 ± 0.4	71.2 ± 0.5
Tiny*ForNet*	general	LaMa (Suvorov et al. 2021)	66.4 ± 0.6	72.9 ± 0.6
Tiny*ForNet*	general	Att. Eraser (Sun et al. 2024)	67.5 ± 1.2	72.4 ± 0.5

Table 1: Ablation of the design decisions in the segmentation phase of *ForAug* on TinyImageNet.

Inpainting. Attentive Eraser (Sun et al. 2024) produces superior results compared to LaMa (Suvorov et al. 2021) (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, 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

FG. size	Augment. Order	BG Strat.	BG. Prune	Original Mixing	Edge Smooth.	Accuracy [%] ViT-Ti	Accuracy [%] ViT-S
TinyImageNet							
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	-	-	-	66.4 ± 0.6	72.9 ± 0.6
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							
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	$\sigma_{\max} = 4.0$	-	81.3 ± 0.1

Table 2: Ablation of design decisions of the recombination phase of *ForAug* on TinyImageNet (top) and ImageNet (bottom).

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 3: Accuracy of ViT-S trained on TinyImageNet (TIN) and Tiny*ForNet* with different foreground position distributions by varying the Bates parameter 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 4: Dataset statistics for TinyImageNet, Tiny*ForNet*, ImageNet, and *ForNet*. For *ForNet* and Tiny*ForNet* we report the number of foreground/background pairs.

Model	ImageNet Accuracy when trained on ImageNet	ImageNet Accuracy when trained on *ForNet*	Delta
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			
DeiT-B			
DeiT-L			
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 5: ImageNet results of models trained on *ForNet* and on ImageNet directly. *ForAug /ForNet* improves the performance of all models in our test.

best results, 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, especially on the full *ForNet*.

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 object. The larger η, the more concentrated the distribution is around the center, $\eta < -1$ concentrate the distribution at the edges. When sampling more towards the center of the image, the difficulty of the task is reduced, which then reduces the performance on TinyImageNet (see Table 3). 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 $\eta < -1$.

After fixing the optimal design parameters in Tables 1 and 2 (last rows), we construct the full *ForNet* dataset using the entire ImageNet dataset. Table 4 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.

4.2 Image Classification Results

Table 5 compares the ImageNet performance of models trained on *ForNet* and on ImageNet. We adopt the training setup of (Nauen, Palacio, and Dengel 2025) and (Touvron, Cord, and Jégou 2022) (details in the supplementary material) for training ViT (Dosovitskiy et al. 2021), Swin (Liu et al. 2021) and ResNet (He et al. 2016) models. Notably, *ForNet* improves performance across all tested architectures, including the ResNet models, 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. (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

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
DeiT-S @ ImageNet					
DeiT-S @ *ForNet*					
DeiT-B @ ImageNet					
DeiT-B @ *ForNet*					
DeiT-L @ ImageNet					
DeiT-L @ *ForNet*					
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	+0.1	±0.0
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 6: 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.

fine-grained datasets: FGVC-Aircraft (Maji et al. 2013), Stanford Cars (Dehghan et al. 2017), Oxford Flowers (Nilsback and Zisserman 2008), Food-101 (Kaur, Sikka, and Divakaran 2017), and Oxford-IIIT Pets (Parkhi et al. 2012). 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%. These results demonstrate that the improved representations from training on *ForNet* translate to superior performance not only on ImageNet, but also on fine-grained image classification tasks.

4.3 Bias and Robustness 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-related background to any background. Figure 3 presents the background robustness results for three datasets: ForNet (all backgrounds vs. backgrounds of same class), ImageNet9 (Xiao et al. 2020) (random backgrounds vs. original backgrounds), and CounterAnimal (Wang et al. 2024) (counter vs. common background). 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. Crucially, training on *ForNet* instead of ImageNet improves the background robustness of all models, reducing the background-gap by boosting the performance of the out-of-background-distribution samples more than the in-distribution ones. These findings highlight the generalization benefits of *ForNet*.

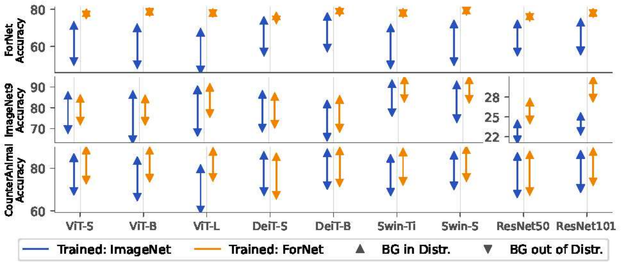


Figure 3: Evaluation of the background robustness on *ForNet*, ImageNet9 and CounterAnimal of models trained on *ForNet* and on ImageNet directly. Training on *ForNet* improves the background robustness of all models, reducing the background distribution-gap (arrow length).

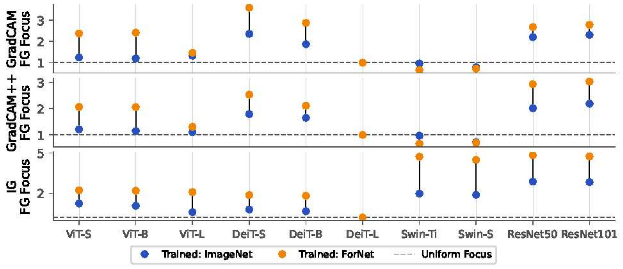
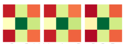
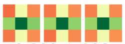
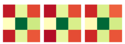
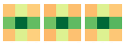
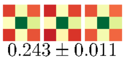
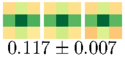
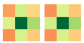
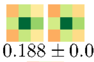
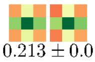
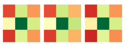
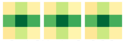
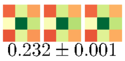
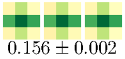
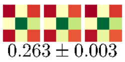
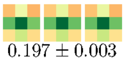
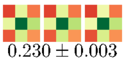
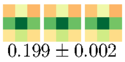


Figure 4: Evaluation of the foreground focus using GradCam, GradCam++ and IntegratedGradients of models trained on *ForNet* (FN) and on ImageNet (IN) directly. Training on *ForNet* improves the foreground focus of almost all models.

Foreground Focus Leveraging our inherent knowledge of the foreground masks when using *ForNet*, as well as common XAI techniques (Selvaraju et al. 2016; Chattopadhay et al. 2018; Sundararajan, Taly, and Yan 2017), 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 (Selvaraju et al. 2016), Grad-CAM++ (Chattopadhay et al. 2018) or IntegratedGradients (IG) (Sundararajan, Taly, and Yan 2017) 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:

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

The foreground focus of a model is its average foreground focus over all test images. Figure 4 presents our findings. Training on *ForNet* significantly increasees the foreground focus of ViT and ResNet across all metrics used. For Swin,

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
DeiT-S	 ≈ 0.202 ± 0.001	 ≈ 0.230 ± 0.004	
DeiT-B	 ≈ 0.188 ± 0.004	 ≈ 0.213 ± 0.010	
DeiT-L	 ≈ 0.203		
Swin-Ti	 0.250 ± 0.007	 0.165 ± 0.002	-0.085
Swin-S	 0.232 ± 0.001	 0.156 ± 0.002	-0.076
ResNet50	 0.263 ± 0.003	 0.197 ± 0.003	-0.066
ResNet101	 0.230 ± 0.003	 0.199 ± 0.002	-0.031

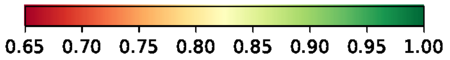


Table 7: We plot accuracy relative to the center accuracy of multiple instantiations of the models when the foreground objects is in different cells of = a 3 × 3 grid. Training on *ForNet* significantly reduces a models center bias.

the foreground focus stagnates when measured using Grad-Cam 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 cell and the minimum performance of a side cell:

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

Table 7 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.

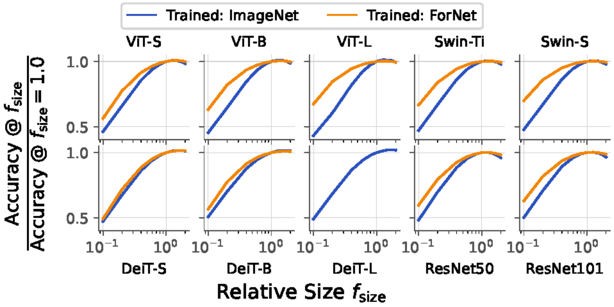


Figure 5: Evaluation of the size bias of models trained on *ForNet*. We plot the accuracy relative to the accuracy when using the mean foreground size.

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 again normalized by the accuracy when using the mean foreground size ($f_{\text{size}} = 1.0$). Figure 5 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 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 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

Alomar, K.; Aysel, H. I.; and Cai, X. 2023. Data Augmentation in Classification and Segmentation: A Survey and New Strategies. 9(2): 46.

Bates, G. 1955. Joint distributions of time intervals for the occurrence of successive accidents in a generalized Polya urn scheme. *Annals of Mathematical Statistics*, 26: 705–720.

Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; and Zagoruyko, S. 2020. End-to-End Object Detection with Transformers.

Chattopadhay, A.; Sarkar, A.; Howlader, P.; and Balasubramanian, V. N. 2018. Grad-CAM++: Generalized Gradient-Based Visual Explanations for Deep Convolutional Networks. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 839–847.

Cubuk, E. D.; Zoph, B.; Mane, D.; Vasudevan, V.; and Le, Q. V. 2018. AutoAugment: Learning Augmentation Policies from Data.

Cubuk, E. D.; Zoph, B.; Shlens, J.; and Le, Q. V. 2019. RandAugment: Practical automated data augmentation with a reduced search space.

Dehghan, A.; Masood, S. Z.; Shu, G.; and Ortiz, E. G. 2017. View Independent Vehicle Make, Model and Color Recognition Using Convolutional Neural Network.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE.

Ding, P.; Soselia, D.; Armstrong, T.; Su, J.; and Huang, F. 2023. Reviving Shift Equivariance in Vision Transformers.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; and Houlsby, N. 2021. 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.

Dwibedi, D.; Misra, I.; and Hebert, M. 2017. Cut, Paste and Learn: Surprisingly Easy Synthesis for Instance Detection.

Ge, Y.; Xu, J.; Zhao, B. N.; Joshi, N.; Itti, L.; and Vineet, V. 2023. Beyond Generation: Harnessing Text to Image Models for Object Detection and Segmentation. *ArXiv*, abs/2309.05956.

Geirhos, R.; Rubisch, P.; Michaelis, C.; Bethge, M.; Wichmann, F. A.; and Brendel, W. 2018. ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness.

Ghiasi, G.; Cui, Y.; Srinivas, A.; Qian, R.; Lin, T.-Y.; Cubuk, E. D.; Le, Q. V.; and Zoph, B. 2020. Simple Copy-Paste is a Strong Data Augmentation Method for Instance Segmentation.

Girshick, R.; Donahue, J.; Darrell, T.; and Malik, J. 2013. Rich feature hierarchies for accurate object detection and semantic segmentation.

He, K.; Gkioxari, G.; Dollár, P.; and Girshick, R. 2017. Mask R-CNN.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hendrycks, D.; and Dietterich, T. 2019. Benchmarking Neural Network Robustness to Common Corruptions and Perturbations.

Hinterstoisser, S.; Pauly, O.; Heibel, H.; Martina, M.; and Bokeloh, M. 2019. 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)*, 2787–2796.

Jonhson, N. L.; Kotz, S.; and Balakrishnan, N. 1995. *Continuous Univariate Distributions*. Wiley series in probability and mathematical statistics. Wiley, 2 edition. ISBN 0-471-58494-0. Wiley series in probability and mathematical statistics.

Kang, J.-S.; and Chung, K. ???? STAug: Copy-Paste Based Image Augmentation Technique Using Salient Target. 10: 123605–123613.

Kaur, P.; Sikka, K.; and Divakaran, A. 2017. Combining Weakly and Webly Supervised Learning for Classifying Food Images.

Khan, S.; Naseer, M.; Hayat, M.; Zamir, S. W.; Khan, F. S.; and Shah, M. 2022. Transformers in Vision: A Survey. *ACM Computing Surveys*, 54(10s): 1–41.

Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A. C.; Lo, W.-Y.; Dollár, P.; and Girshick, R. 2023. Segment Anything.

Kolesnikov, A.; Beyer, L.; Zhai, X.; Puigcerver, J.; Yung, J.; Gelly, S.; and Houlsby, N. 2020. Big Transfer (BiT): General Visual Representation Learning. In *Computer Vision – ECCV 2020*, 491–507. Cham: Springer International Publishing. ISBN 978-3-030-58558-7.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In Pereira, F.; Burges, C.; Bottou, L.; and Weinberger, K., eds., *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc.

Le, Y.; and Yang, X. 2015. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7): 3.

Li, X.; Chen, Y.; Zhu, Y.; Wang, S.; Zhang, R.; and Xue, H. 2023. ImageNet-E: Benchmarking Neural Network Robustness via Attribute Editing.

Ling, E.; Huang, D.; and Hur, M. 2022. Humans need not label more humans: Occlusion Copy & Paste for Occluded Human Instance Segmentation.

Liu, S.; Zeng, Z.; Ren, T.; Li, F.; Zhang, H.; Yang, J.; Jiang, Q.; Li, C.; Yang, J.; Su, H.; Zhu, J.; and Zhang, L. 2023. Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection.

Liu, Y.; Matsoukas, C.; Strand, F.; Azizpour, H.; and Smith, K. 2022. PatchDropout: Economizing Vision Transformers Using Patch Dropout.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In *2021 IEEE/CVF*

International Conference on Computer Vision (ICCV), 9992–10002. Los Alamitos, CA, USA: IEEE Computer Society.

Maji, S.; Kannala, J.; Rahtu, E.; Blaschko, M.; and Vedaldi, A. 2013. Fine-Grained Visual Classification of Aircraft. Technical report.

Nauen, T. C.; Palacio, S.; and Dengel, A. 2025. Which Transformer to Favor: A Comparative Analysis of Efficiency in Vision Transformers. In *Proceedings of the Winter Conference on Applications of Computer Vision (WACV)*, 6955–6966.

Nilsback, M.-E.; and Zisserman, A. 2008. Automated Flower Classification over a Large Number of Classes. In *Indian Conference on Computer Vision, Graphics and Image Processing*.

Parkhi, O. M.; Vedaldi, A.; Zisserman, A.; and Jawahar, C. V. 2012. Cats and Dogs. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Rangel, G.; Cuevas-Tello, J. C.; Nunez-Varela, J.; Puente, C.; and Silva-Trujillo, A. G. 2024. A Survey on Convolutional Neural Networks and Their Performance Limitations in Image Recognition Tasks. 2024(1).

Ren, T.; Liu, S.; Zeng, A.; Lin, J.; Li, K.; Cao, H.; Chen, J.; Huang, X.; Chen, Y.; Yan, F.; Zeng, Z.; Zhang, H.; Li, F.; Yang, J.; Li, H.; Jiang, Q.; and Zhang, L. 2024. Grounded SAM: Assembling Open-World Models for Diverse Visual Tasks.

Rojas-Gomez, R. A.; Lim, T.-Y.; Do, M. N.; and Yeh, R. A. 2023. Making Vision Transformers Truly Shift-Equivariant.

Sanderson, E.; and Matuszewski, B. J. 2022. *FCN-Transformer Feature Fusion for Polyp Segmentation*, 892–907. Springer International Publishing. ISBN 9783031120534.

Selvaraju, R. R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; and Batra, D. 2016. Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. 128(2): 336–359.

Shermaine, A. J. N.; Lazarou, M.; and Stathaki, T. 2025. Image compositing is all you need for data augmentation.

Shorten, C.; and Khoshgoftaar, T. M. 2019. A survey on Image Data Augmentation for Deep Learning. 6(1).

Sun, W.; Cui, B.; Dong, X.-M.; and Tang, J. 2024. Attentive Eraser: Unleashing Diffusion Model's Object Removal Potential via Self-Attention Redirection Guidance.

Sundararajan, M.; Taly, A.; and Yan, Q. 2017. Axiomatic Attribution for Deep Networks.

Suvorov, R.; Logacheva, E.; Mashikhin, A.; Remizova, A.; Ashukha, A.; Silvestrov, A.; Kong, N.; Goka, H.; Park, K.; and Lempitsky, V. 2021. Resolution-robust Large Mask Inpainting with Fourier Convolutions.

Takahashi, R.; Matsubara, T.; and Uehara, K. 2018. Data Augmentation using Random Image Cropping and Patching for Deep CNNs. 30(9): 2917–2931.

Touvron, H.; Cord, M.; Douze, M.; Massa, F.; Sablayrolles, A.; and Jegou, H. 2021. Training data-efficient image transformers & distillation through attention. In Meila, M.; and Zhang, T., eds., *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, 10347–10357. PMLR.

Touvron, H.; Cord, M.; and Jégou, H. 2022. DeiT III: Revenge of the ViT. In Avidan, S.; Brostow, G.; Cissé, M.; Farinella, G. M.; and Hassner, T., eds., *Computer Vision – ECCV 2022*, 516–533. Cham: Springer Nature Switzerland.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. In Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Vezakis, I. A.; Georgas, K.; Fotiadis, D.; and Matsopoulos, G. K. 2024. EffiSegNet: Gastrointestinal Polyp Segmentation through a Pre-Trained EfficientNet-based Network with a Simplified Decoder.

Wang, Q.; Lin, Y.; Chen, Y.; Schmidt, L.; Han, B.; and Zhang, T. 2024. A Sober Look at the Robustness of CLIPs to Spurious Features. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Wang, W.; Bao, H.; Dong, L.; Bjorck, J.; Peng, Z.; Liu, Q.; Aggarwal, K.; Mohammed, O. K.; Singhal, S.; Som, S.; and Wei, F. 2022a. Image as a Foreign Language: BEiT Pretraining for All Vision and Vision-Language Tasks.

Wang, W.; Dai, J.; Chen, Z.; Huang, Z.; Li, Z.; Zhu, X.; Hu, X.; Lu, T.; Lu, L.; Li, H.; Wang, X.; and Qiao, Y. 2022b. InternImage: Exploring Large-Scale Vision Foundation Models with Deformable Convolutions.

Werman, L. K. M. 2021. DeePaste – Inpainting for Pasting.

Wortsman, M.; Ilharco, G.; Gadre, S. Y.; Roelofs, R.; Gontijo-Lopes, R.; Morcos, A. S.; Namkoong, H.; Farhadi, A.; Carmon, Y.; Kornblith, S.; and Schmidt, L. 2022. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In Chaudhuri, K.; Jegelka, S.; Song, L.; Szepesvari, C.; Niu, G.; and Sabato, S., eds., *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, 23965–23998. PMLR.

Xiao, K.; Engstrom, L.; Ilyas, A.; and Madry, A. 2020. Noise or Signal: The Role of Image Backgrounds in Object Recognition.

Xu, M.; Yoon, S.; Fuentes, A.; and Park, D. S. 2023. A Comprehensive Survey of Image Augmentation Techniques for Deep Learning. 137: 109347.

Yu, J.; Wang, Z.; Vasudevan, V.; Yeung, L.; Seyedhosseini, M.; and Wu, Y. 2022. CoCa: Contrastive Captioners are Image-Text Foundation Models. *Transactions on Machine Learning Research*.

Yun, S.; Han, D.; Chun, S.; Oh, S. J.; Yoo, Y.; and Choe, J. 2019. CutMix: Regularization Strategy to Train Strong Classifiers With Localizable Features. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE.

Zhang, C.; Pan, F.; Kim, J.; Kweon, I. S.; and Mao, C. 2024. ImageNet-D: Benchmarking Neural Network Robustness on Diffusion Synthetic Object.

Zhang, H.; Cisse, M.; Dauphin, Y. N.; and Lopez-Paz, D. 2018. mixup: Beyond Empirical Risk Minimization. In *International Conference on Learning Representations*.

Zhong, Z.; Zheng, L.; Kang, G.; Li, S.; and Yang, Y. 2017. Random Erasing Data Augmentation.

Zong, Z.; Song, G.; and Liu, Y. 2022. DETRs with Collaborative Hybrid Assignments Training.