Proper Reuse of Image Classification Features Improves Object Detection

Cristina Vasconcelos, Vighnesh Birodkar, Vincent Dumoulin
Google Research, Brain Team
{crisnv,vighneshb,vdumoulin}@google.com

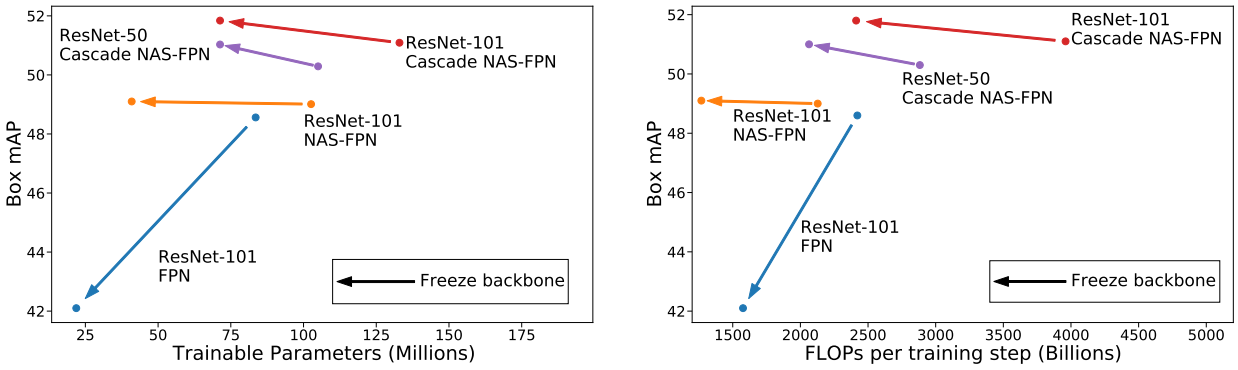

Figure 1. Impact on object detection performance of preserving and not fine-tuning features learned on ImageNet. The arrows indicate impact on each model, when trained with frozen backbone weights. As long as the remaining detector components have enough capacity **(left)**, freezing increases performance while significantly reducing resources used during training (**right**).

Abstract

A common practice in transfer learning is to initialize the downstream model weights by pre-training on a data-abundant upstream task. In object detection specifically, the feature backbone is typically initialized with ImageNet classifier weights and fine-tuned on the object detection task. Recent works show this is not strictly necessary under longer training regimes and provide recipes for training the backbone from scratch. We investigate the opposite direction of this end-to-end training trend: we show that an extreme form of knowledge preservation—freezing the classifier-initialized backbone— consistently improves many different detection models, and leads to considerable resource savings. We hypothesize and corroborate experimentally that the remaining detector components capacity and structure is a crucial factor in leveraging the frozen backbone. Immediate applications of our findings include performance improvements on hard cases like detection of long-tail object classes and computational and memory resource savings that contribute to making the field more accessible to researchers with access to fewer computational resources.

1. Introduction

Transfer learning [35] is a widely adopted practice in deep learning, especially when the target task has a smaller dataset; the model is first pre-trained on an upstream task in which a larger amount data is available and then fine-tuned on the target task. Transfer learning from ImageNet or even larger [5, 16, 47] or weakly labeled [33] datasets was repeatedly shown to yield performance improvements across various vision tasks, architectures, and training procedures [23, 33, 47].

For object detection [20] it is common practice to initialize the model's backbone (see Figure 2 for a model diagram) with weight values obtained by pretraining on an image classification task, such as ImageNet [44]. Traditionally, the backbone is fine-tuned while training the other detector components from scratch. Two lines of work have recently made seemingly contradictory observations on transfer learning for object detection. On one hand, Sun et al. [47] show that object detectors benefit from the amount of classification data used in pre-training. On the other hand, more recent papers have reported that the performance gap between transferring from a pre-trained backbone initialization and training the backbone from scratch with smaller, in-domain datasets vanishes with longer training [8, 14, 26, 45].

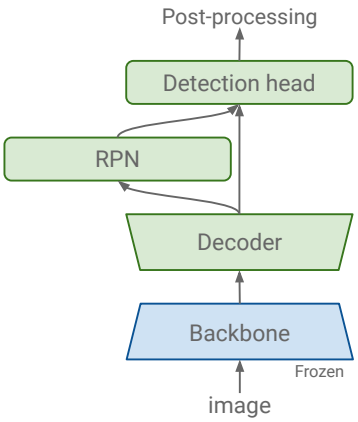

Figure 2. A sketch of the Faster-RCNN detection model that serves as an example. Our proposed training procedure freezes the backbone after initializing it from a classification task and trains the other components from scratch.

We revisit transfer learning in its simplest form, where the backbone's classifier initialization is frozen during detection training. This allows us to better understand the usefulness of the pre-trained representation without confounding factors resulting from fine-tuning. This approach has many advantages: it is simple, resource saving, and easy to replicate. Moreover, using this approach, we are able to make the following two observations:

1. Longer training is a confounding factor in investigating the usefulness of the pre-trained representation, because the weights of a fine-tuned backbone move further away from their pre-trained initialization.
2. This is important, because our ablation studies show that the representation learned on the upstream classification task is better for object detection than the one obtained from fine-tuning or training from scratch on the object detection task itself using a smaller in-domain dataset.

When we preserve the pre-trained representation through freezing the backbone, we observe a consistent performance improvement from pre-training on larger datasets. Provided that the subsequent detector components have enough capacity, the models trained with a frozen backbone even exceed the performance of their fine-tuned or "from scratch" counterparts.

As an immediate contribution of our findings, we show that it is possible to train an off-the-shelf object detection model with similar or superior performance while significantly reducing the need for computational resources, both memory-wise and computationally-wise (FLOPs)(Figure 1). The performance benefits of the proposed upstream task knowledge preservation are even more clear when stratifying results by classes and the number of annotations available. Our results show that our extreme formulation of model reuse has a clear positive impact on classes with a low number of annotations, such as those found in long-tail object recognition.

2. Related work

Benefits of object detector backbone pre-training. The hypothesis that it is possible to improve performance on vision tasks by training a better base model has been largely investigated [23, 29, 42]. Huang et al. [19] focus on the close correlation between architectures with different capacity, their classification performances on ImageNet and their corresponding object detection performance. Sun et al. [47] shifts the discussion away from model comparison to focus on the impact of pre-training data. Their ablations contrast backbone pretraining on ImageNet versus JFT-300M classification tasks to corroborate the hypothesis that large-scale data helps in representation learning that ultimately improves transfer learning performance in classification, detection, segmentation, and pose estimation tasks. Importantly for our discussion, they point out that the benefit of pre-training from a larger dataset is capacity bounded.

Unsupervised, weakly-supervised and self-supervised pre-training. Reducing the need for task-specific annotations is especially relevant for tasks requiring fine-grained annotations such as detection and segmentation. This motivates an active area of research on unsupervised [1, 6], self-supervised [21, 43, 46, 52] and weakly supervised methods for object detection (pre-)training [24, 27, 34, 39]. In a self-training formulation, Barret et al. [53] propose to discard the original labels from ImageNet and obtain pseudo-labels using a detector trained on MSCOCO. The pseudo-labeled ImageNet and labeled COCO data are then combined to train a new model. Closer to our work, Dhruv et al. [34] investigate pre-training on billions of weakly-labeled images using social media hashtags. They show that when using large amounts of pre-training data, detection performance is bounded by model capacity. Gains on smaller models are small or negative, but as model capacity increases the larger pre-training datasets yield consistent improvements. On the impact of pre-training using noisy labels, they conjecture that gains from weakly supervised pre-training compared to supervised pre-training (ImageNet) may be primarily due to improved object classification performance, rather than spatial localization performance. The contrast between gains in AP versus AP50 is used as a proxy for this analysis.[1]

Training from scratch. He et al. [14] show that it is possible to train an object detector from scratch from random initialization using in-domain supervision only but using a

[1] AP measures the average precision at different intersection over union (IoU) thresholds ranging from 0.5:0.95, while AP@50 is computed as the average precision computed at IoU threshold 0.5 only.

longer training schedule and with proper regularization and normalization. Li et al. [26] aim to reduce resources and increase stability and propose to split the "training from scratch" procedure into a two-step procedure with progressively increasing input resolution, both executed on the target dataset. The combination of training from scratch and data augmentation was explored in [7,12]. Ghiasi et al. [12], show that the simple data augmentation mechanism of pasting objects randomly onto an image provides solid gains over previous state of the art approaches for both MSCOCO and LVIS detection and segmentation. During their ablation studies, smaller models were trained using strong data augmentation and trained from scratch, while larger models (presenting the best performance) were fine-tuned from a model pre-trained on ImageNet. Du et al. [7] present a set of training strategies to improve the detector's performance combining data augmentation (large-scale jittering [50]), a smooth activation function, regularization, backbone architecture changes, and normalization techniques.

Importance of resource savings. The field of deep learning is becoming increasingly resource-conscious. From an environmental perspective, the carbon footprint of training deep learning models is non-negligible [51], and our findings contribute to reducing that footprint. From a social perspective, Obando-Ceron & Castro [4] note that the large-scale standardized deep reinforcement learning benchmarks such as the Arcade Learning Environment [2] have the unfortunate effect of partitioning the field into groups with access to large-scale computational resources and groups without such resources. They argue that smaller-scale environments can still yield valuable scientific insights while being accessible to more researchers. The large batch sizes used in recent object detection work [8, 12] arguably have a similar deterring effect. While we don't claim to solve the issue in its entirety for object detection, our findings contribute to reducing the computational resources necessary to achieve strong object detection performance and take a step in the direction of allowing researchers with varied levels of access to resources to contribute to the field.

3. Methodology

Our main hypothesis is that the features learnt by training on large-scale image classification tasks are better for the object detection task than those obtained from comparatively smaller, in-domain datasets. The classification datasets that we consider (ImageNet (1.2 M) and JFT-300M (300 M)) contain orders of magnitude more images than common detection datasets like MSCOCO (118K) and LVIS (100 K). They key insight we propose is to freeze the weights learnt on the classification task and choose the remaining components such that they have enough capacity to learn the detection-specific features.

3.1. Preserving classification features

To preserve the knowledge learnt during classification, we use the most natural and obvious strategy of freezing the weights of the classification network (also called backbone). The common practice in literature [20, 29, 42] is to train all the weights in the model after the backbone has been initialized. We instead consider the alternative strategy of freezing all of the backbone weights. Not only does this save compute and speed up training, but as we discover, improves the performance of many modern detection architectures.

3.2. Detection-specific capacity

Adapting classification networks to the detection task, typically requires the addition of detection specific components (Figure 2), like a Region Proposal Network [42], a Feature Pyramid Network [28] and more recently Detection Cascades [3]. We observe that the capacity of detection components plays a large role in the ability of networks to generalize, particularly when we initialize from a classification task. We show that when the detection-specific components have enough capacity, initializing from a classification task and freezing those weights performs better than fine-tuning or training from scratch (as is common in the literature). Moreover, we see that the performance gain increases when we pre-train on a more diverse classification dataset.

3.3. Data augmentation

We use Large Scale Jittering (LSJ) [50] for all of our experiments and Copy-and-paste augmentation [12] for our best results with EfficientNet [49]. We note that our proposed technique is complementary to both these data augmentation strategies. Moreover, for the experiments with frozen backbones, the data augmentation techniques are able to improve results *only* by helping the detection specific components.

4. Experiments

Our experiments connect two apparently contradictory conclusions in the object detection literature.[2] Sun et al. [47] advocate for pre-training on image classification datasets, and observe benefits that increase with the pre-training dataset's scale. Going in the opposite direction, the most recent trend of papers follows He et al. [14]'s findings that support training from scratch under longer training regimes.

We resolve this contradiction by pointing out that a backbone fine-tuned for longer can move further away from its

[2] See https://github.com/tensorflow/models/blob/master/official/projects/backbone_reuse/README.md for open-sourced code and instructions.

pre-trained initialization. If—as our evidence suggests—fine-tuning is detrimental to the learned backbone representation in terms of detection performance, then this would explain why the benefits of pre-training appear to vanish in longer training regimes. However, this degradation can be prevented if the pre-trained backbone is frozen during detection training. This lets us realize the benefits of pre-trained initialization even under longer training schedules. Doing so is simple, easy to replicate, yields performance improvements on all investigated architectures coupled with expressive-enough subsequent detector components (Figure 1), and saves considerable computational resources during training.

4.1. Experimental Setting

Architecture. Our baselines were built on top of two open-sourced codebases. The ablation studies presented using ResNet [15] models were built on top of Du et al. [9]'s codebase,[3] motivated by the strong performance shown on baselines trained from scratch. Building on this, our first group of experiments (subsection 4.2) explores the impact of feature preservation on a combinations of small backbones (ResNet-50, ResNet-101) with Fast-RCNN-based detectors. We then introduce detectors with increasing amounts of capacity by varying the feature pyramids [28] (FPN and NAS-FPN) and adding Cascade heads [3].

The second group of experiments (subsection 4.3) applies the insights gained in a competitive setting, namely a Mask-RCNN with an EfficientNet-B7 backbone and NAS-FPN and Cascade heads using Ghiasi et al. [12]'s codebase and covers both detection and segmentation tasks.[4] We also investigate how Ghiasi et al.'s strong Copy+Paste data augmentation approach interacts with backbone freezing. In comparison to [9]'s baseline, [12]'s implementation replaces NAS-FPN's convolution layers with ResNet bottleneck blocks [15] and also adds one extra convolutional layers on the RPN head.

Training Parameters. All our models trained with frozen backbones adopt exactly the same hyper-parameters as the corresponding non-frozen ones. No changes were made to Ghiasi et al. [12]'s code, while changes to Du et al. [9]'s code are described in the next paragraph. Further details can be found in the original references.

During our study on ResNets models, we target the use and replication of our findings on environments with lower resources available for training. Therefore, the batch size was reduced from Du et al. [9]'s original value of 256 down to 64. Consequently, the learning rate was also adjusted to 0.08. Results presented in the main text use the same longer training schedule as in their original formulation (600 epochs for ResNets and 390 epochs for EfficientNet-B7s). Our results with shorter training schedules are described in Appendix C.

[3] https://github.com/tensorflow/models/blob/master/official/vision/beta/MODEL_GARDEN.md

[4] https://github.com/tensorflow/tpu/tree/master/models/official/detection/projects/copy_paste

Model	Pretraining	mAP	AP @ 50
ResNet-101 + FPN	From scratch	48.4	70.1
	ImageNet	48.6	70.5
	+ Freeze backbone	(−6.5) 42.1	(−5.2) 65.3
	JFT-300M	48.7	70.5
	+ Freeze backbone	(−5.6) 43.1	(−3.3) 67.2
ResNet-101 + NAS-FPN	From scratch	47.2	68.2
	ImageNet	49.0	70.0
	+ Freeze backbone	(+0.1) 49.1	(+0.3) 70.3
	JFT-300M	49.1	70.2
	+ Freeze backbone	(+1.0) 50.1	(+1.5) 71.8
ResNet-50 + NAS-FPN + Cascade	From scratch	50.0	68.1
	ImageNet	50.3	68.0
	+ Freeze backbone	(+0.7) 51.0	(+1.1) 69.1
	JFT-300M	50.4	68.1
	+ Freeze backbone	(+1.7) 52.1	(+2.3) 70.4
ResNet-101 + NAS-FPN + Cascade	From scratch	50.4	68.4
	ImageNet	51.1	69.1
	+ Freeze backbone	(+0.8) 51.8	(0.8) 69.9
	JFT-300M	51.1	69.0
	+ Freeze backbone	(+1.7) 52.8	(+2.1) 71.1

Table 1. With a powerful-enough detector, freezing the backbone to its pre-trained initialization during detection training outperforms fine-tuning the backbone or training it from scratch.

Datasets. We perform ablation studies on detection and segmentation with the MSCOCO (2017) [30] and LVIS 1.0 [13] datasets. MSCOCO has $118k$ images covering 91 classes (80 used in practice). The LVIS dataset was designed to simulate the long-tail distribution of classes in natural images. The 1.0 version used in this paper contains $100k$ images covering 1203 classes.

Due to large costs associated with annotating localization information, image classification datasets are much larger than object detection [36] and segmentation [30] datasets. Since we investigate the benefits associated with dataset scale, we use two image classification datasets with increasing amounts of data. ImageNet contains $1M$ labeled images based on 1000 categories while JFT-300M [17] contains more than $300M$ images labeled with 18291 categories. Labeling error in JFT-300M is estimated to be around 20% [47], but this is offset by a larger visual diversity than ImageNet. We use ImageNet or JFT-300M to pretrain the backbone on classification tasks, except when training from scratch. Neither is used to train the detector.

4.2. Revisiting ResNet reuse for object detection

We start by investigating the impact of the backbone training strategy (trained from scratch, fine-tuned from a pre-trained initialization, or frozen at its pre-trained initialization) while controlling for pre-training dataset size

Paper	Pre-training	Schedule	Freeze?	mAP
Pre-training on larger classification datasets helps.				
Sun et al. [47]	ImageNet	Short	Yes	47.8
	JFT	Short	Yes	49.0
Pre-training does not help with longer training schedules.				
He et al. [14]	ImageNet	Long	No	49.0
	JFT	Long	No	49.1
Backbone freezing & high-capacity detector components help.				
Ours	JFT	Long	Yes	**50.1**

Table 2. We revisit the conclusions from [14, 47] under modern training regimes and best practices. We use their main conclusions and re-train these models for a fair comparison. All results are reported on a ResNet-101 backbone with a NAS-FPN. Note that Sun et al. [47] did not run experiments with NAS-FPN.

(ImageNet, JFT-300M), backbone architecture (ResNet-50, ResNet-101), detector architecture (FPN, NAS-FPN, NAS-FPN + Cascade), and training schedule (72 epochs, 600 epochs). Tabular results for the longer training schedule are presented and discussed here (Table 1), and results using the shorter schedule are presented in Appendix C.

The relative benefit from pre-training on larger classification datasets is clear when freezing the backbone. The comparison between *ImageNet + Freeze backbone* and *JFT-300M + Freeze backbone* in Table 1 shows a consistent improvement in performance ($+0.9$ mAP in most cases) across the different backbones and detector components tested. See Appendix A (Figure 6) for an alternative visualization.

Proper reuse of image classification features improves performance. Our main finding is that the advantage gained from pre-trained backbone networks gets lost with a longer training schedule. Notice that in Table 2, the improvement from JFT-300M which was apparent in the short training schedule (blue-shaded region), disappears with the long training schedule (red-shaded region). In contrast, we show that with freezing the backbone, and thus preserving the pre-training knowledge, we maintain the improvement due to JFT-300M across both the training schedules.

Our results corroborate Sun et al. [47]'s observation that pre-training on larger datasets is beneficial when fine-tuning the backbone for a shorter schedule (Table 2, shadowed blue), and Appendix C's Table 12 shows this is consistent across architectures. Our results complement Sun et al. [47] and He et al. [14]: by freezing the backbone, we see that both can be explained through the lens of pre-trained knowledge preservation. In addition, we notice that the benefit from the re-use of the knowledge from large scale image classification datasets is bounded by the capacity of remaining components, as we show next.

Detection model	#Params (Million)		Δ mAP
	Original	Trained	
FPN	83.5	(26.1%) 21.9	−6.5
NAS-FPN	102.6	(39.9%) 40.9	+0.1
Cascade + NAS-FPN	132.9	(53.6%) 71.3	+0.7

Table 3. Number of parameters that are trained when keeping the backbone frozen (a ResNet-101) with various detection models. We report Δ mAP as the performance gap from fine-tuned counterparts. Weights initialized from an ImageNet pre-trained model.

Feature preservation benefits are bounded by the remaining trainable capacity. Sun et al. [47] conjecture that the overall benefit from pre-training is bounded by the capacity of the whole model (backbone + detection components). Using an FPN detector, they present experiments freezing backbone weights under short training regimes and show that those models under-perform their fine-tuned counterparts. Our results using FPN detectors confirm the same decrease in performance (Figure 3) on both short and long training schedules.

In a new direction of investigation, we refine the original conjecture to pinpoint the remaining detector components as the principal capacity bottleneck in terms of benefiting from pre-trained features. Figure 1 plots the performance observed across models built with components of increasing capacity and trained with either backbone freezing or fine-tuning. Table 3 shows the number of parameters of the models ablated and the corresponding number of parameters trained on models with frozen backbones. Object detection models with higher capacity in the remaining trainable components clearly surpass their fine-tuned counterparts. See Appendix A for more capacity ablations.

To summarize, in typical settings (e.g. using an FPN) we don't see benefits from freezing the backbone because the remaining detector components don't have enough capacity. In other words, the representation learned from pre-training on the large classification dataset is better, provided that there are enough learnable parameters (like with NAS-FPN). Consequently, the comparable performances achieved by fine-tuning the backbone and training it from scratch under a long schedule could simply be a consequence of the fact that fine-tuning for longer moves it further away from the good representation found by pre-training on the classification task.

Through backbone freezing and extensive experimentation, we (i) disentangle the benefit of using pre-trained backbones when compared to training from scratch under long training schedules; and (ii) show the benefit on the final detector of using even larger classification datasets. Additionally, our results suggest that the knowledge contained in pre-trained weights can and should be preserved during longer training regimes.

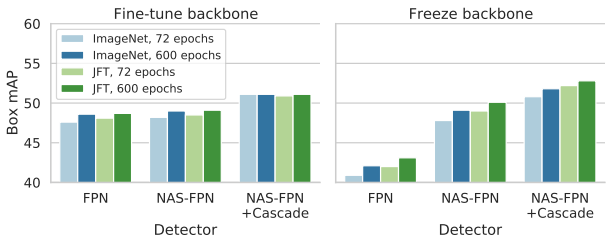

Figure 3. Detectors using a ResNet-101 backbone. Sufficient capacity on components trained is important in realizing the benefits of preserving the representation from a large classification dataset.

4.3. EfficientNet-B7's reuse for object detection

Next, we present our results on freezing features learned from a classification dataset (ImageNet) on EfficientNet-B7 based detectors. The ablations also investigate how freezing the backbone complements strong data augmentation techniques designed specifically for localization tasks. The models compared in this section adopt both large scale jittering [50] and Copy+Paste [12] data augmentation (previously shown to improve training under fine-tuning and training from scratch regimes). The comparison of our results with other strong detectors are presented in Table 4.

Freezing EfficientNet-B7 backbones improves performance on both MSCOCO and LVIS datasets. Table 4 presents our results when applied to the same architectures and training regimes as Ghiasi et al. [12]'s baselines for MSCOCO and LVIS. It shows a clear gain in performance for frozen models observed both on detection and instance segmentation tasks. On the MSCOCO baseline, backbone freezing outperforms fine-tuning by $+0.9$ Box mAP. Note that this number on the impact of freezing alone is comparable to the benefit of using Copy+Paste augmentation ($+1.1$). From the same table, the results on MSCOCO instance segmentation highlight the benefit of feature preservation even more, with a larger benefit derived ($+0.7$ Mask mAP) than from Copy+Paste ($+0.3$).

Similarly, freezing classification features also presented a positive impact on performance for LVIS for both detection and instance segmentation. This strengthens our claim that feature preservation is important, since LVIS is harder than MSCOCO due to its long-tailed classed distribution.

When the backbone is frozen, strong data augmentations improves the remaining detector components even further. The benefit of feature preservation is even more clear when combined with localization-specific data augmentations. For MSCOCO detection, the improvement obtained by feature preservation is of $+1.1$ mAP and reaches $+1.5$ mAP for segmentation masks (Table 4). This increase in segmentation performance is larger than the original $+0.3$ gain obtained by using the Copy+Paste augmentation itself ($+1.1$ Box mAP and $+0.3$ Mask mAP). That is, by combining feature preservation and Copy+Paste augmentation the total gain over the baseline numbers is $+2.2$ Box mAP and $+1.8$ Mask AP. Note that in this case the data augmentations only affect the detection-specific components, since the backbone remains frozen.

MSCOCO (val)	**Box mAP**	**Mask mAP**
Swin Transformer [31]*	**57.1**	**49.5**
Cascade Eff-B7 NAS-FPN (1280) [12]	54.8	46.9
+ freeze backbone (ours)	(+0.9) 55.7	(+0.7) 47.6
+ Copy-Paste	55.9	47.2
+ freeze backbone (ours)	(+1.1) 57.0	(+1.5) 48.7
+ Self-training Copy-Paste	57.0	48.9
Soft Teacher + Swin-L [52]	59.1	51.0

LVIS (val)	**Box mAP**	**Mask mAP**
cRT (ResNeXt-101-32×8d) [22]	—	27.2
LVIS Challenge 2020 Winner [48]†	41.1	38.8
Eff-B7 NAS-FPN (1280) [12]	37.2	34.7
+ freeze backbone (ours)	(+2.2) 39.4	(+2.5) 37.2
+ Copy-Paste	41.6	38.1
+ freeze backbone (ours)	(+1.5) **43.1**	(+1.8) **39.9**

Table 4. Freezing the backbone improves two strong baselines on MSCOCO and LVIS. when training only on the `train2017` split. "Self-training" and [52] use the extra training data. *ImageNet-22K pre-training. †No test-time augmentation. All results are reported on the validation sets.

The combination of feature preservation and data augmentation produces even larger improvements in performance on LVIS (Table 4). This can be observed on both detection ($+1.5$ over Copy-Paste and $+5.9$ over the baseline) and segmentation ($+1.8$ over Copy-Paste and $+5.2$ over the baseline). We further stratify the results obtained on LVIS by number of annotations (Table 5) and object size (Table 6). Our EfficientNet-B7 models with a frozen pre-trained backbone show improvements across all object sizes, numbers of annotations, and detection and segmentation tasks.

Our LVIS results are obtained by following Ghiasi et al. [12]'s two-step training procedure: after training the detector using unbalanced loss, the classifier head is further tuned using a class balanced loss. The goal of the second stage is to improve performance on rare classes. In the original fine-tuning setting the second training stage causes a drop in performance on the *frequent* classes. However, with feature preservation, all three groups (rare, common and frequent classes) improve in performance in the second stage.

LVIS	Box				Mask			
	mAP	**mAP$_r$**	**mAP$_c$**	**mAP$_f$**	**mAP**	**mAP$_r$**	**mAP$_c$**	**mAP$_f$**
First stage results: regular training								
Copy and Paste [12]	38.5	19.3	37.3	48.2	35.0	19.5	34.9	42.1
+ Freeze backbone	(+1.2) 39.7	(+2.9) 22.2	(+1.5) 38.8	(+0.1) 48.3	(+1.5) 36.5	(+2.9) 22.4	(+1.9) 36.8	(+0.4) 42.5
Second stage: tunes detection-classifier final layer using class-balanced loss								
Copy and Paste [12]	41.6	31.5	39.8	48.0	38.1	32.1	37.1	41.9
+ Freeze backbone	(+1.5) 43.1	(+1.7) 33.2	(+2.1) 41.9	(+0.7) 48.7	(+1.1) 39.9	(+1.5) 33.6	(+2.5) 39.6	(+1.0) 42.9

Table 5. Performance using EfficientNet-B7 + NAS-FPN. Freezing the backbone has the strongest positive performance impact on rare ($\mathrm{mAP_r}$) and common ($\mathrm{mAP_c}$) classes, while still improving frequent ($\mathrm{mAP_f}$) classes. Original first phase results are provided by the authors of [12]. Results without Copy-Paste augmentations can be found in Appendix D.

LVIS	Box				Mask			
	mAP	**mAP$_s$**	**mAP$_m$**	**mAP$_l$**	**mAP**	**mAP$_s$**	**mAP$_m$**	**mAP$_l$**
First stage results: regular training								
Copy and Paste [12]	38.5	30.5	48.2	55.5	35.0	25.7	45.6	53.2
+ frozen backbone	(+1.2) 39.7	(+0.2) 30.7	(+1.4) 49.6	(+2.3) 57.8	(+1.5) 36.5	(+0.3) 26.0	(+1.6) 47.2	(+2.8) 56.0
Second stage: tunes detection-classifier final layer using class-balanced loss								
Copy and Paste [12]	41.6	33.5	51.5	58.1	38.1	28.4	49.0	55.6
+ frozen backbone	(+1.5) 43.1	(+0.6)34.1	(+1.8) 53.3	(+2.3) 60.4	(+1.8) 39.9	(+0.7) 29.1	(+2.1) 51.1	(+2.7) 58.3

Table 6. Performance using EfficientNet-B7 + NAS-FPN. Freezing the backbone has the strongest positive performance impact on large objects ($\mathrm{mAP_l}$), then medium-sized objects ($\mathrm{mAP_m}$), and finally small objects ($\mathrm{mAP_s}$). Original first phase results are provided by the authors of [12]. Results without Copy-Paste augmentations can be found in Appendix D.

4.4. How does preserving pre-trained representations help?

So far we have established that given sufficient detector capacity it is better to freeze the pre-trained backbone than to fine-tune it or train it from scratch on detection data, but it is not obvious why that is the case.

We explore this question by visualizing how freezing the backbone impacts performance on classes with different numbers of training annotations. Figure 4 shows class-wise box mAPs relative to those obtained by fine-tuning the backbone for the *freeze* and *from scratch* training strategies. As before, we observe that with a lower-capacity detector (FPN) training the backbone from scratch is comparable in performance to fine-tuning it from a pre-trained initialization, and that freezing the backbone underperforms fine-tuning it. Interestingly, fine-tuning from a JFT initialization outperforms training from scratch for classes with fewer annotations. As we move towards larger detectors, performance across the three strategies remains similar for classes with larger amounts of annotations while freezing the backbone and training it from scratch become increasingly beneficial and detrimental (respectively) for classes with fewer annotations.

We observe a similar behaviour when comparing *freeze* and *fine-tune* in a more competitive setting (Figure 5, Table 5), where the benefits of backbone freezing concentrate mostly on classes with fewer annotations.

Given these observations, we conjecture that the pre-trained representation contains features beneficial for detection that require many annotations to be learned from the detection data alone in addition to being brittle to fine-tuning on detection data. While not identical to catastrophic forgetting [11], this phenomenon bears some resemblance to it: in fine-tuning on the detection task, the object detection model appears to struggle to preserve knowledge not only beneficial to the upstream classification task, but *beneficial to the downstream detection task itself*.

4.5. Beyond backbone freezing

Throughout this work we presented backbone freezing as a knowledge preservation strategy that yields performance benefits for object detection. What this shows is that there are better ways of using a pre-trained model for downstream object detection applications, but it does not mean (nor do we claim) that backbone freezing is itself an optimal strategy.

To demonstrate this, we present initial results using a lightweight alternative strategy in the form of residual adapters [40, 41], which have been successfully applied to adapt to downstream tasks such as cross-domain few-shot image classification [25], natural language processing [18,32,37], and transfer learning with expert image classifiers [38].

Our initial results (Table 7) show that equipping the

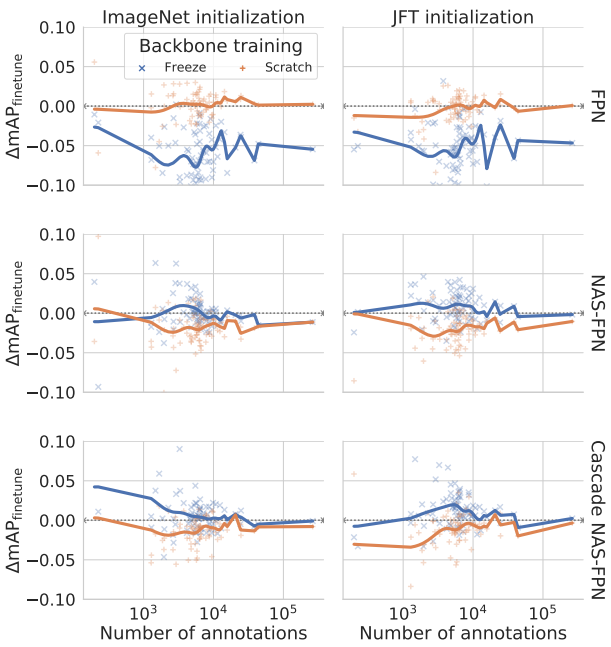

Figure 4. Class-wise mAP as a function of the number of training annotations *relative to fine-tuning the backbone*. Smoothed curves are obtained by applying Gaussian smoothing with $\sigma = 1000$. The ResNet-101 backbone is combined with detectors of varying capacities (rows) and initializations (columns). Freezing the backbone is increasingly beneficial as the detector capacity increases (top to bottom rows) and as the number of annotations decreases.

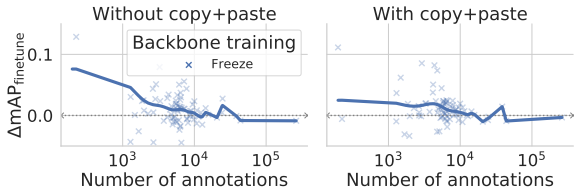

Figure 5. Class-wise mAP as a function of the number of training annotations *relative to fine-tuning the backbone*. Smoothed curves are obtained by applying Gaussian smoothing with $\sigma = 1000$. The EfficientNet-B7 architecture is trained without (left) and with (right) copy+paste augmentations.

frozen backbone with residual adapter layers that are trained alongside the detector components yields further improvements over full backbone fine-tuning. We conjecture that residual adapters and similar adaptation approaches such as feature-wise transformations [10] can better incorporate the object detection training signal while preserving aspects of the image classifier representation that are useful to the object detection task. See Appendix E for more results on the use of adapters.

More generally, adjacent fields such as transfer learning, multi-task learning, few-shot classification, and domain adaptation have all tackled the problem of using pre-trained models for downstream applications, and we believe that object detection would benefit from their insights.

Model	Pretraining	mAP	AP @ 50
ResNet-101 + NAS-FPN + Cascade	ImageNet	51.1	69.1
	+ Freeze backbone	(+0.8) 51.8	(+0.8) 69.9
	+ Res. adapters	(+1.9) 53.0	(+2.4) 71.5
	JFT-300M	51.1	69.0
	+ Freeze backbone	(+1.7) 52.8	(+2.1) 71.1
	+ Res. adapters	(+2.5) 53.6	(+3.0) 72.0

Table 7. Adding residual adapters to the frozen backbone and training them along with the subsequent detector components yields another performance increase.

5. Conclusions

We cast a new light on the re-use of pre-trained representations obtained from a large-scale classification task in a downstream detection setting. Particularly, we show that preserving the backbone representation obtained from training on a large-scale classification task is beneficial to object detection and instance segmentation. We also show how this ties together the two seemingly contradictory observations of [47] and [14], the missing piece being the fact that the longer training schedule moves the backbone further away from a good initial representation.

We investigate backbone freezing as a simple approach to knowledge preservation and demonstrate its benefits when coupled with sufficient detection-specific component capacity through extensive experiments across multiple combinations of backbones, detection models, datasets, and training schedules. This approach easy to implement and reproduce and requires significantly less computational resources during training.

While the need to control for resources used in previous work means that SOTA is not yet accessible to most, we demonstrate demonstrate computation and memory savings in all settings, meaning that practitioners are able to train larger models with larger batch sizes given the same amount of resources, and accessibility could be further improved through future work on reducing the number of training epochs required. We also believe that tapping into the rich model re-use literature in adjacent fields represents a promising direction for future work. Finally, our findings can be used in future neural architecture search work to take advantage of pre-trained and frozen classification based features to ultimately do better than NAS-FPN.

References

[1] Amir Bar, Xin Wang, Vadim Kantorov, Colorado J Reed, Roei Herzig, Gal Chechik, Anna Rohrbach, Trevor Darrell, and Amir Globerson. Detreg: Unsupervised pretraining with region priors for object detection, 2021. 2

[2] Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, 2013. 3

[3] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6154–6162, 2018. 3, 4

[4] Johan Samir Obando Ceron and Pablo Samuel Castro. Revisiting rainbow: Promoting more insightful and inclusive deep reinforcement learning research. In *International Conference on Machine Learning*, pages 1373–1383. PMLR, 2021. 3

[5] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017. 1

[6] Zhigang Dai, Bolun Cai, Yugeng Lin, and Junying Chen. Up-detr: Unsupervised pre-training for object detection with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1601–1610, June 2021. 2

[7] Xianzhi Du, Barret Zoph, Wei-Chih Hung, and Tsung-Yi Lin. Simple training strategies and model scaling for object detection. *CoRR*, abs/2107.00057, 2021. 3

[8] Xianzhi Du, Barret Zoph, Wei-Chih Hung, and Tsung-Yi Lin. Simple training strategies and model scaling for object detection, 2021. 1, 3

[9] Xianzhi Du, Barret Zoph, Wei-Chih Hung, and Tsung-Yi Lin. Simple training strategies and model scaling for object detection. *arXiv preprint arXiv:2107.00057*, 2021. 4

[10] Vincent Dumoulin, Ethan Perez, Nathan Schucher, Florian Strub, Harm de Vries, Aaron Courville, and Yoshua Bengio. Feature-wise transformations. *Distill*, 3(7):e11, 2018. 8

[11] Robert M. French. Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, 3(4):128–135, 1999. 7

[12] 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. *arXiv preprint arXiv:2012.07177*, 2020. 3, 4, 6, 7, 13, 14

[13] Agrim Gupta, Piotr Dollár, and Ross B. Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 5356–5364. Computer Vision Foundation / IEEE, 2019. 4

[14] Kaiming He, Ross Girshick, and Piotr Dollár. Rethinking imagenet pre-training. In *Proceedings of the International Conference on Computer Vision*, pages 4918–4927, 2019. 1, 2, 3, 5, 8, 12

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society, 2016. 4

[16] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 1

[17] Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015. 4

[18] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. *CoRR*, abs/1902.00751, 2019. 7

[19] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, and Kevin Murphy. Speed/accuracy trade-offs for modern convolutional object detectors. *CoRR*, abs/1611.10012, 2016. 2

[20] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, and Kevin Murphy. Speed/accuracy trade-offs for modern convolutional object detectors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 1, 3

[21] Jisoo Jeong, Seungeui Lee, Jeesoo Kim, and Nojun Kwak. Consistency-based semi-supervised learning for object detection. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. 2

[22] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *Eighth International Conference on Learning Representations (ICLR)*, 2020. 6

[23] Simon Kornblith, Jonathon Shlens, and Quoc V. Le. Do better imagenet models transfer better? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1, 2

[24] Jason Kuen, Federico Perazzi, Zhe Lin, Jianming Zhang, and Yap-Peng Tan. Scaling object detection by transferring classification weights. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6044–6053, 2019. 2

[25] Wei-Hong Li, Xialei Liu, and Hakan Bilen. Improving task adaptation for cross-domain few-shot learning. *CoRR*, 2021. 7

[26] Yang Li, Hong Zhang, and Yu Zhang. Rethinking training from scratch for object detection, 2021. 1, 3

[27] Xiaodan Liang, Si Liu, Yunchao Wei, Luoqi Liu, Liang Lin, and Shuicheng Yan. Towards computational baby learning: A weakly-supervised approach for object detection. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 999–1007, 2015. 2

[28] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 3, 4

[29] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 2, 3, 13

[30] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2014. cite arxiv:1405.0312Comment: 1) updated annotation pipeline description and figures; 2) added new section describing datasets splits; 3) updated author list. 4

[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. *arXiv preprint arXiv:2103.14030*, 2021. 6

[32] Rabeeh Karimi Mahabadi, James Henderson, and Sebastian Ruder. Compacter: Efficient low-rank hypercomplex adapter layers. *CoRR*, abs/2106.04647, 2021. 7

[33] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens Van Der Maaten. Exploring the limits of weakly supervised pretraining. In *Proceedings of the European conference on computer vision (ECCV)*, pages 181–196, 2018. 1

[34] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, pages 185–201, Cham, 2018. Springer International Publishing. 2

[35] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 2009. 1

[36] Dim P. Papadopoulos, Jasper R. R. Uijlings, Frank Keller, and Vittorio Ferrari. Extreme clicking for efficient object annotation. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 4940–4949, 2017. 4

[37] Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. Adapterfusion: Non-destructive task composition for transfer learning, 2021. 7

[38] Joan Puigcerver, Carlos Riquelme, Basil Mustafa, Cédric Renggli, André Susano Pinto, Sylvain Gelly, Daniel Keysers, and Neil Houlsby. Scalable transfer learning with expert models. *ArXiv*, abs/2009.13239, 2021. 7

[39] Vignesh Ramanathan, Rui Wang, and Dhruv Mahajan. Predet: Large-scale weakly supervised pre-training for detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2865–2875, 2021. 2

[40] Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Learning multiple visual domains with residual adapters. *CoRR*, abs/1705.08045, 2017. 7, 13

[41] Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Efficient parametrization of multi-domain deep neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 7, 13

[42] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28:91–99, 2015. 2, 3

[43] Chuck Rosenberg, Martial Hebert, and Henry Schneiderman. Semi-supervised self-training of object detection models. In *2005 Seventh IEEE Workshops on Applications of Computer Vision (WACV/MOTION'05) - Volume 1*, volume 1, pages 29–36, 2005. 2

[44] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 1

[45] Zhiqiang Shen, Zhuang Liu, Jianguo Li, Yu-Gang Jiang, Yurong Chen, and Xiangyang Xue. Object detection from scratch with deep supervision. *IEEE transactions on pattern analysis and machine intelligence*, 42(2):398–412, 2019. 1

[46] Kihyuk Sohn, Zizhao Zhang, Chun-Liang Li, Han Zhang, Chen-Yu Lee, and Tomas Pfister. A simple semi-supervised learning framework for object detection. In *arXiv:2005.04757*, 2020. 2

[47] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *ICCV*, pages 843–852. IEEE Computer Society, 2017. 1, 2, 3, 4, 5, 8, 12

[48] Jingru Tan, Gang Zhang, Hanming Deng, Changbao Wang, Lewei Lu, Quanquan Li, and Jifeng Dai. 1st place solution of lvis challenge 2020: A good box is not a guarantee of a good mask. *arXiv preprint arXiv:2009.01559*, 2020. 6

[49] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019. 3

[50] Mingxing Tan, Ruoming Pang, and Quoc V. Le. Efficientdet: Scalable and efficient object detection. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 10778–10787. Computer Vision Foundation / IEEE, 2020. 3, 6, 12

[51] Neil C Thompson, Kristjan Greenewald, Keeheon Lee, and Gabriel F Manso. Deep learning's diminishing returns: The cost of improvement is becoming unsustainable. *IEEE Spectrum*, 58(10):50–55, 2021. 3

[52] Mengde Xu, Zheng Zhang, Han Hu, Jianfeng Wang, Lijuan Wang, Fangyun Wei, Xiang Bai, and Zicheng Liu. End-to-end semi-supervised object detection with soft teacher. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 2, 6

[53] Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin D Cubuk, and Quoc V Le. Rethinking pre-training and self-training. In *Advances in Neural Information Processing Systems*, pages 3833–3845, 2020. 2