A Decade's Battle on Dataset Bias: Are We There Yet?

Zhuang Liu Kaiming He*

Meta AI Research, FAIR

Abstract We revisit the "dataset classification" experiment suggested by Torralba and Efros a decade ago [51], in the new era with large-scale, diverse, and hopefully less biased datasets as well as more capable neural network architectures. Surprisingly, we observe that modern neural networks can achieve excellent accuracy in classifying which dataset an image is from: *e.g.*, we report 84.7% accuracy on held-out validation data for the three-way classification problem consisting of the YFCC, CC, and DataComp datasets. Our further experiments show that such a dataset classifier could learn semantic features that are generalizable and transferable, which cannot be simply explained by memorization. We hope our discovery will inspire the community to rethink the issue involving dataset bias and model capabilities.

1 Introduction

In 2011, Torralba and Efros [51] called for a battle against dataset bias in the community, right before the dawn of the deep learning revolution [26]. Over the decade that followed, progress on building diverse, large-scale, comprehensive, and hopefully less biased datasets (*e.g.*, [27,30,40,42,49]) has been an engine powering the deep learning revolution. In parallel, advances in algorithms, particularly neural network architectures, have achieved unprecedented levels of ability on discovering concepts, abstractions, and patterns—including *bias*—from data.

In this work, we take a renewed "*unbiased look at dataset bias*" [51] after the decade-long battle. Our study is driven by the tension between building less biased datasets versus developing more capable models—the latter was less prominent at the time of the Torralba and Efros paper [51]. While efforts to reduce bias in data may lead to progress, the development of advanced models could better exploit dataset bias and thus counteract the promise.

Our study is based on a fabricated task we call *dataset classification*, which is the "*Name That Dataset*" experiment designed in [51] (Figure 1). Specifically, we randomly sample a large number (*e.g.*, as large as one million) of images from each of several datasets, and train a neural network on their union to classify from which dataset an image is taken. The datasets we experiment with are presumably among the most diverse, largest, and uncurated datasets in the wild, collected from the Internet. For example, a typical combination we study, referred to as "YCD", consists of images from YFCC [49], CC [4], and DataComp [15] and presents a 3-way dataset classification problem.

* Work done at Meta; now at MIT. Code: github.com/liuzhuang13/bias

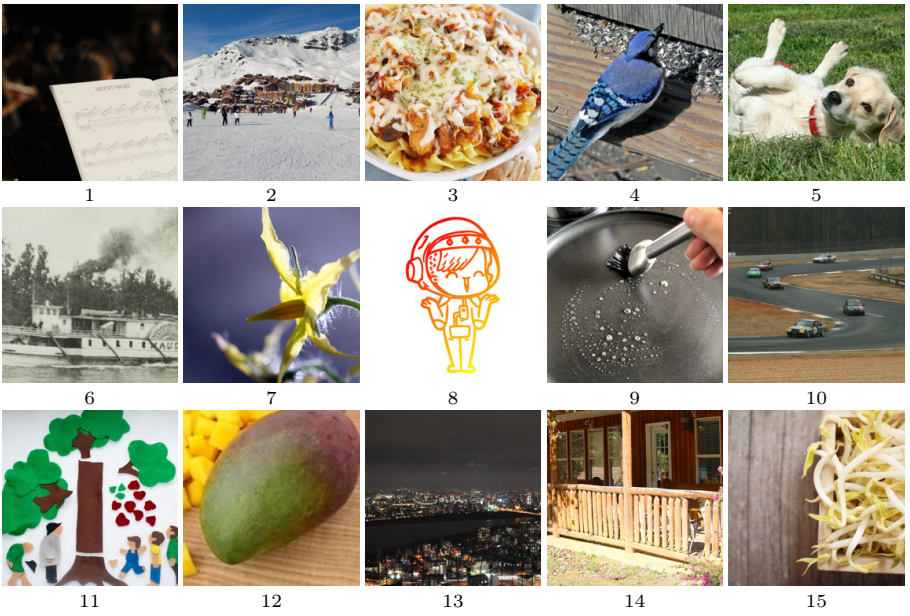

Figure 1: The "Name That Dataset" game [51] in 2024: These images are sampled from three modern datasets: YFCC [49], CC [4], and DataComp [15]. *Can you specify which dataset each image is from*? While these datasets appear to be less biased, we discover that neural networks can easily accomplish this "*dataset classification*" task with surprisingly high accuracy on the held-out validation set.
Answer: YFCC: 1, 4, 7, 10, 13; CC: 2, 5, 8, 11, 14; DataComp: 3, 6, 9, 12, 15

To our (and many of our initial readers') surprise, modern neural networks can achieve excellent accuracy on such a dataset classification task. Trained in the aforementioned YCD set that is challenging for human beings (Figure 1), a model can achieve >84% classification accuracy on the *held-out* validation data, *vs*. 33.3% of chance-level guess. This observation is highly robust, over a large variety of dataset combinations and across different generations of architectures [11,20,26,31,44], with very high accuracy (*e.g.*, over 80%) achieved in most cases.

Intriguingly, for such a dataset classification task, we have a series of observations that are analogous to those observed in *semantic* classification tasks (*e.g.*, object classification). For example, we observe that training the dataset classifier on *more* samples, or using *stronger* data augmentation, can *improve* accuracy on held-out validation data, even though the training task becomes harder. This is similar to the generalization behavior in semantic classification tasks. This behavior suggests that the neural network attempts to discover dataset-specific patterns—a form of bias–to solve the dataset classification task. Further experiments suggest that the representations learned by classifying datasets carry some semantic information that is transferrable to image classification tasks.

As a comparison, if the samples of different datasets were unbiasedly drawn from the same distribution, the model should not discover any dataset-specific

bias. To check this, we study a pseudo-dataset classification task, in which the different "datasets" are uniformly sampled from a single dataset. We observe that this classification task quickly becomes intractable, as the only way for the classifier to approach this task is to memorize every single instance and its subset identity. As a result, increasing the number of samples, or using stronger data augmentation, makes memorization more difficult or intractable in experiments. No transferability is observed. These behaviors are strikingly contrary to those of the real dataset classification task.

More surprisingly, we observe that even *self-supervised* learning models are capable of capturing certain bias among different datasets. Specifically, we pre-train a self-supervised model on the union of different datasets, without using any dataset identity as the labels. Then with the pre-trained representations frozen, we train a linear classifier for the dataset classification task. Although this linear layer is the only layer that is tunable by the dataset identity labels, the model can still achieve a surprisingly high accuracy (*e.g.*, 78%) for dataset classification. This *transfer learning* behavior resembles the behaviors of typical self-supervised learning methods (*e.g.*, for image classification).

In summary, we report that modern neural networks are surprisingly capable of discovering hidden bias from different datasets. This observation is true even for modern datasets that are very large, diverse, less curated, and presumably less biased. The neural networks can solve this task by discovering generalizable patterns (*i.e.*, generalizable from training data to validation data, or to downstream tasks), exhibiting behaviors analogous to those observed in semantic classification tasks. Comparing with the game of "*Name That Dataset*" in the Torralba and Efros paper [51] a decade ago, this game even becomes way easier given today's capable neural networks. In this sense, the issue involving dataset bias has not been relieved. We hope our discovery will stimulate new discussion in the community regarding the relationship between dataset bias and the continuously improving models.

2 A Brief History of Datasets

Pre-dataset Eras. The concept of "datasets" did not emerge directly out of the box in the history of computer vision research. Before the advent of computers (*e.g.*, see Helmholtz's book of the 1860s [57]), scientists had already recognized the necessity of "test samples", often called "stimuli" back then, to examine their computational models about the human vision system. The stimuli often consisted of synthesized patterns, such as lines, stripes, and blobs. The practice of using synthesized patterns was followed in early works on computer vision.

Immediately after the introduction of devices for digitizing photos, researchers were able to *validate* and justify their algorithms on one or very few real-world images [39]. For example, the *Cameraman* image [41] has been serving as a standard test image for image processing research since 1978. The concept of using data (which was not popularly referred to as "datasets") to *evaluate* computer vision algorithms was gradually formed by the community.

Datasets for Task Definition. With the introduction of *machine learning* methods into the computer vision community, the concept of "datasets" became clearer. In addition to the data for the validation purpose, the application of machine learning introduced the concept of *training data*, from which the algorithms can optimize their model parameters.

As such, the training data and validation data put together inherently *define a task* that is of interest. For example, the MNIST dataset [29] defines a 10-digit classification task; the Caltech-101 dataset [14] defines an image classification task of 101 object categories; the PASCAL VOC suite of datasets [13] define a family of classification, detection, and segmentation tasks of 20 object categories.

To incentivize more capable algorithms, more challenging tasks were defined, albeit *not* necessarily of greater interest in everyday life. The most notable example of this kind, perhaps to the surprise of today's readers, is the ImageNet dataset [8]. ImageNet has over one million images defined with 1000 classes (many of them being fine-grained animal species), which is nontrivial even for normal human beings to recognize [24]. At the time when ImageNet was proposed, algorithms for solving this *task* appeared to be cumbersome—*e.g.*, the organizers provided SIFT features [32] pre-computed to facilitate people studying this problem, and typical methods back then may train 1000 SVM classifiers, which in itself is a nontrivial problem [56]. If ImageNet were to remain as a task on its own, we wouldn't be able to witness the deep learning revolution.

But a paradigm shift awaited.

Datasets for Representation Learning. Right after the deep learning revolution in 2012 [26], the community soon discovered that the neural network representations learned on large-scale datasets like ImageNet are *transferrable* (*e.g.*, [10,17,62]). The discovery brought in a paradigm shift in computer vision: it became a common practice to pre-train representations on ImageNet and transfer them to downstream tasks.

As such, the ImageNet dataset was no longer a task of its own; it became a pinhole of the *universal visual world* that we want to represent. Consequently, the used-to-be cumbersome aspects became advantages of this dataset: it has a larger number of images and more diversified categories than most (if not all) other datasets at that time, and empirically it turned out that these properties are important for learning good representations.

Encouraged by ImageNet's enormous success, the community began to pursue more general and ideally universal visual representations. Tremendous effort has been paid on building larger, more diversified, and hopefully less biased datasets. Examples include YFCC100M [49], CC12M [4], and DataComp-1B [15]—the main datasets we study in this paper—among many others [9, 42, 45, 46]. It is intriguing to notice that the building of these datasets does *not* always define a task of interest to solve; actually, many of these large-scale datasets do not even provide a split of training/validation sets. It is with the goal of *pre-training* in mind that these datasets were built.

On Dataset Bias. Given the increasing importance of datasets, the bias introduced by datasets has drawn the community's attention. Torralba and Efros [51]

presented the dataset classification problem and examined dataset bias in the context of hand-crafted features with SVM classifiers. Tommasi *et al*. [50] studied the dataset classification problem using neural networks, specifically focusing on linear classifiers with pre-trained ConvNet features [10]. The datasets they studied are smaller in scale and simpler comparing with today's web-scale data.

The concept of classifying different datasets has been further developed in domain adaption methods [16,53]. These methods learn classifiers to adversarially distinguish features from different domains, where each domain can be thought of as a dataset. The problems studied by these methods are known to have significant domain gaps. On the contrary, the datasets we study are presumably less distinguishable, at least for human beings.

Another direction on studying dataset bias is to replicate the collection process of a dataset and examine the replicated data. ImageNetV2 [38] replicated the ImageNet validation set's protocol. It observed that this replicated data still clearly exhibits bias as reflected by accuracy degradation. The bias is further analyzed in [12].

Many benchmarks [21,22,25,64] have been created for testing models' generalization under various forms of biases, such as common corruptions and hazardous conditions. There is also a rich line of work on mitigating dataset bias. Training on multiple datasets [28,33] can potentially mitigate dataset bias. Methods that adapt models to data with different biases at test time [47,59] have also gained popularity recently.

Bias in datasets also has significant social implications. Several well-known datasets have been identified with biases in demographics [3,61] and geography [43]. They also contain harmful societal stereotypes [36,54,68]. Addressing these biases is critical for fairness and ethical considerations. Tools like REVISE [58] and Know Your Data [18] offer automatic analysis for potential bias in datasets. Debiasing approaches, such as adversarial learning [65] and domain-independent training [60], have also shown promise in reducing the effects of dataset bias.

3 Dataset Classification

The dataset classification task [51] is defined like an image classification task, but each dataset forms its own class. It creates an N-way classification problem where N is the number of datasets. The classification accuracy is evaluated on a validation set consisting of held-out images sampled from these datasets.

3.1 On the Datasets We Use

We intentionally choose the datasets that can make the dataset classification task challenging. We choose our datasets based on the following considerations: (1) They are large in scale. Smaller datasets might have a narrower range of concepts covered, and they may not have enough training images for dataset classification. (2) They are general and diversified. We avoid datasets that are about a specific scenario (*e.g.*, cities [6], scenes [69]) or a specific meta-category of objects (*e.g.*,

dataset	description
YFCC [49]	100M Flickr images
CC [4]	12M Internet image-text pairs
DataComp [15]	1B image-text pairs from Common Crawl [1]
WIT [45]	11.5M Wikipedia images-text pairs
LAION [42]	2B image-text pairs from Common Crawl [1]
ImageNet [8]	14M images from search engines

Table 1: Datasets used in our experiments.

flowers [34], pets [35]). (3) They are collected with the intension of pre-training generalizable representations, or have been used with this intension. Based on these criteria, we choose the datasets listed in Table 1.

Although these datasets are supposedly more diverse, there are still differences in their collection processes that potentially contribute to their individual biases. For example, their sources are different: Flickr is a website where users upload and share photos, Wikipedia is a website focused on knowledge and information, Common Crawl is an organization that crawls the web data, and the broader Internet involves a more general range of content than these specific websites. Moreover, different levels of curation have been involved in the data collection process: *e.g.*, LAION was collected by reverse-engineering the CLIP model [37] and reproducing its zero-shot accuracy [42].

Despite our awareness of these potential biases, a neural network's excellent ability to capture them is beyond our expectation. In particular, we note that we evaluate a network's dataset classification accuracy by applying it to each validation image *individually*, which ensures that the network has no opportunity to exploit the underlying statistics of several images.

3.2 Main Observation

We observe surprisingly high accuracy achieved by neural networks in this dataset classification task. This observation is robust across different settings. By default, we randomly sample 1M and 10K images from each dataset as training and validation sets, respectively. We train a ConvNeXt-T model [31] following common practice of supervised training (implementation details are in Appendix). We observe the following behaviors in our experiments:

High accuracy is observed across *dataset combinations*. In Table 2 (top panel), we enumerate all 20 (C_6^3) possible combinations of choosing 3 out of the 6 datasets listed in Table 1. In summary, in all cases, the network achieves >62% dataset classification accuracy; and in 16 out of all 20 combinations, it even achieves >80% accuracy. In the combination of YFCC, CC, and ImageNet, it achieves the highest accuracy of **92.7%**. Note that the chance-level guess gives 33.3% accuracy.

In Table 2 (bottom panel), we study combinations involving 3, 4, 5, and all 6 datasets. As expected, using more datasets leads to a more difficult task, reflected

YFCC	CC	DataComp	WIT	LAION	ImageNet	accuracy
✓	✓	✓				84.7
✓	✓		✓			83.9
✓	✓			✓		85.0
✓	✓				✓	92.7
✓		✓	✓			85.8
✓		✓		✓		72.1
✓		✓			✓	90.2
✓			✓	✓		86.6
✓			✓		✓	86.7
✓				✓	✓	91.9
	✓	✓	✓			83.6
	✓	✓		✓		62.8
	✓	✓			✓	82.8
	✓		✓	✓		84.3
	✓		✓		✓	91.3
	✓			✓	✓	84.1
		✓	✓	✓		71.5
		✓	✓		✓	88.9
		✓		✓	✓	68.2
			✓	✓	✓	90.7
✓	✓	✓				84.7
✓	✓	✓	✓			79.1
✓	✓	✓	✓	✓		67.4
✓	✓	✓	✓	✓	✓	69.2

Table 2: Dataset classification yields high accuracy in all combinations. Top panel: all 20 combinations that involve 3 datasets out of all 6. **Bottom panel**: combinations with 3, 4, 5, or 6 datasets. All results are with 1M training images sampled from each dataset.

by the decreasing accuracy. However, the network still achieves 69.2% accuracy when all 6 datasets are included.

High accuracy is observed across *model architectures.* In Table 3, we report the results on the YCD combination using different generations of representative model architectures: AlexNet [26], VGG [44], ResNet [20], ViT [11], and ConvNeXt [31].

We observe that *all architectures can solve the task excellently*: 4 out of the 5 networks achieve excellent accuracy of >80%, and even the now-classical AlexNet achieves a strong result of 77.8%.

This result shows the neural networks are extremely good at capturing dataset biases, regardless of their concrete architectures. There has been

model	accuracy
AlexNet	77.8
VGG-16	83.5
ResNet-50	83.8
ViT-S	82.4
ConvNeXt-T	84.7

Table 3: Different model architectures all achieve high accuracy. Results are on the YCD combination with 1M images each.

significant progress in network architecture design after the AlexNet paper, including normalization layers [2,23], residual connections [20], self-attention [11,55]. The "inductive bias" in network architectures can also be different [11]. Nevertheless, none of them appears to be indispensable for dataset classification (*e.g.*, VGG [44] has none of these components): the ability to capture dataset bias may be inherent in deep neural networks, rather than enabled by specific components.

High accuracy is observed across different *model sizes*. By default, we use ConvNeXt-Tiny (27M parameters) [31]. The term "Tiny" is with reference to the modern definition of ViT sizes [11,52] and is comparable to ResNet-50 (25M) [20]. In Figure 2, we report results of models with different sizes by varying widths and depth.

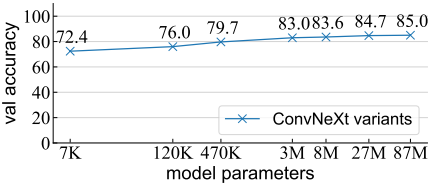

Figure 2: Models of different sizes all achieve very high accuracy, while they can still be substantially smaller than the sizes of typical modern networks. Here the models are variants of ConvNeXt [31], whose "Tiny" size has 27M parameters. Results are on the YCD combination with 1M training images from each set.

To our further surprise, even *very small* models can achieve strong accuracy for the dataset classification task. A ConvNeXt with as few as 7K parameters (3/10000 of ResNet-50) achieves 72.4% accuracy on classifying YCD. This suggests that neural networks' structures are very effective in learning the underlying dataset biases. Dataset classification can be done without a massive number of parameters, which is often credited for deep learning's success in conventional visual recognition tasks.

We also observe that larger models get increasingly better, although the return becomes diminishing. This is consistent with observations on conventional visual recognition tasks. Moreover, we have not observed overfitting behaviors to the extent of the model sizes and dataset scales we have studied. This implies that there may exist generalizable patterns that help the models determine dataset identities and the model is not trying to memorize the training data. More investigations on generalization and memorization are presented next.

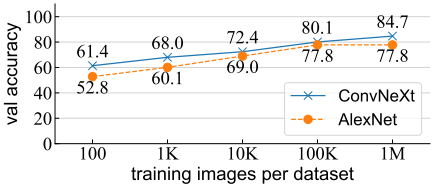

Figure 3: Dataset classification accuracy increases with the number of training images. This behavior suggests that the model is learning certain patterns that are generalizable, which resembles the behavior observed in typical semantic classification tasks. Results are on the YCD combination.

Dataset classification accuracy benefits from *more training data*. We vary the number of training images for YCD classification and present results in Figure 3.

Intriguingly, models trained with *more* data achieve *higher* validation accuracy. This trend is consistently observed in both the modern ConvNeXt and the classical AlexNet. While this behavior appears to be natural in *semantic* classification tasks, we remark

that this is not necessarily true in dataset classification: in fact, if the models *were* struggling with *memorizing* the training data, their generalization performance on the validation data might decrease. The observed behavior—*i.e.*, more training data improves validation accuracy—suggests that the model is learning certain semantic patterns that are generalizable to unseen data, rather than memorizing and overfitting the training data.

Dataset classification accuracy benefits from *data augmentation*. Data augmentation [26] is expected to have similar effects as increasing the dataset size (which is the rationale behind its naming). Our default training setting uses random cropping [48], RandAug [7], MixUp [67], and CutMix [63] as data augmentations. Table 4 shows the results of using reduced or no data augmentations.

augmentation / training images per dataset	10K	100K	1M
no aug	43.2	71.9	76.8
w/ RandCrop	66.1	74.5	84.2
w/ RandCrop, RandAug	70.2	78.0	85.0
w/ RandCrop, RandAug, MixUp/CutMix	72.4	80.1	84.7

Table 4: Data augmentation improves dataset classification accuracy, similar to the behavior of semantic classification tasks. Results are on the YCD combination.

Adding data augmentation makes it more difficult to memorize the training images. However, using stronger data augmentation consistently *improves* the dataset classification accuracy. This behavior remains largely consistent regardless of the number of training images per dataset. Again, this behavior mirrors that observed in *semantic* classification tasks, suggesting that dataset classification is approached not through memorization, but by learning patterns that are generalizable from the training set to the unseen validation set.

Summary. In sum, we have observed that neural networks are highly capable of solving the dataset classification task with good accuracy. This observation holds true across a variety of conditions, including different combinations of datasets, various model architectures, different model sizes, dataset sizes, and data augmentation strategies.

4 Analysis

In this section, we analyze the model behaviors in different modified versions involving the dataset classification task. This reveals more intriguing properties of neural networks for dataset classification.

4.1 Low-level Signatures?

There is a possibility that the high accuracy is simply due to low-level signatures, which are less noticeable to humans but are easily identifiable by neural networks.

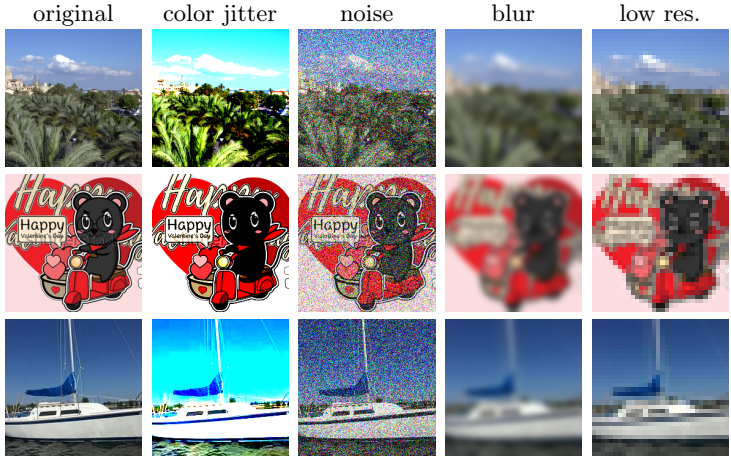

Figure 4: Different corruptions for suppressing low-level signatures. We apply a certain type of corruption to both the training and validation sets, on which we train and evaluate our model.

Potential signatures could involve JPEG compression artifacts (*e.g.*, different datasets may have different compression quality factors) and color quantization artifacts (*e.g.*, colors are trimmed or quantized depending on the individual dataset). We design a set of experiments that help us preclude this possibility.

Specially, we apply a certain type of image corruption to both the training and validation sets, on which we train and evaluate our model. In other words, we perform the dataset classification task *on corrupted data*.[1] We consider four types of image corruption: (i) color jittering [26], (ii) adding Gaussian noise with a fixed standard deviation (std); (iii) blurring the image by a fixed-size Gaussian kernel; and (iv) reducing the image resolution. Figure 4 shows some example images for each corruption. We note that we apply one type of corruption at a time, resulting in one distinct data of dataset classification.

corruption (on train+val)	accuracy
none	84.7
color jittering (strength: 1.0)	81.1
color jittering (strength: 2.0)	80.2
Gaussian noise (std: 0.2)	77.3
Gaussian noise (std: 0.3)	75.1
Gaussian blur (radius: 3)	80.9
Gaussian blur (radius: 5)	78.1
low resolution (64×64)	78.4
low resolution (32×32)	68.4

Table 5: High accuracy are achieved on different corrupted versions of the dataset classification task. This suggests that low-level signature is not a main responsible factor. Results are on the YCD combination.

Table 5 shows the dataset classification results for each image corruption. As expected, corruption reduces the classification accuracy, as both training and

[1] It is worth noticing that this is different from *data augmentation*, which applies random image corruption to the training data.

validation sets are affected. Despite degradation, strong classification accuracy can still be achieved, especially when the degree of corruption is weaker. Introducing these different types of corruption should effectively disrupt low-level signatures, such as JPEG or color quantization artifacts. The results imply that the models attempt to solve the dataset classification task beyond using low-level biases.

4.2 Memorization or Generalization?

In Sec. 3.2, we have shown that the models learned for dataset classification behave like those learned for semantic classification tasks (Figure 3 and Table 4), in the sense that they exhibit *generalization* behaviors. This behavior is in sharp contrast with the *memorization* behavior, as we discuss in the next comparison.

We consider a *pseudo*-dataset classification task. In this scenario, we manually create multiple pseudo-datasets, all of which are sampled without replacement from the same source dataset. We expect this process to give us multiple pseudo-datasets that are truly unbiased.

Table 6 reports the *training* accuracy of a model trained for this pseudo-dataset classification task, using different numbers of training images per set, without *vs*. with data augmentation. When the task is relatively simple, the model achieves 100% training accuracy; however, when the task becomes more difficult (more training images or stronger augmentation), the model fails to converge, as reflected by unstable, non-decreasing loss curves.

imgs per set	w/o aug	w/ aug
100	100.0	100.0
1K	100.0	100.0
10K	100.0	fail
100K	fail	fail

Table 6: *Training* accuracy on a pseudo-dataset classification task. Here we create 3 pseudo-datasets, all of which are sampled without replacement from the same source dataset (YFCC). This *training* task becomes more difficult for the network to solve if given more training images and/or stronger data augmentation. Validation accuracy is ~33% as no transferrable pattern is learned.

This phenomenon implies that the model attempts to *memorize* individual images and their labels to accomplish this pseudo-dataset classification task. Because the images in these pseudo-datasets are unbiased, there should be no shared patterns that can be discovered to discriminate these different sets. As a result, the model is forced to memorize the images and their random labels, similar to the scenario in [66]. But memorization becomes more difficult when given more training images or stronger augmentation, which fails the training process after a certain point.

This phenomenon is unlike what we have observed in our real dataset classification task (Figure 3 and Table 4). This again suggests that the model attempts to capture shared, generalizable patterns in the real dataset classification task.

Although it may seem evident, we note that the model trained for the pseudo-dataset classification task does *not* generalize to validation data (which is held out and sampled from each pseudo-dataset). Even when the training accuracy is 100%, we report a chance-level accuracy of ~33% in the validation set.

4.3 Self-supervised Learning

Thus far, all our dataset classification results are presented under a *fully-supervised* protocol: the models are trained end-to-end with full supervision (using dataset identities as the labels). Next, we explore a *self-supervised* protocol, following the common protocol used for semantic classification tasks in the scenario of self-supervised learning.

Formally, we pre-train a self-supervised learning model MAE [19] without using any labels. Then we freeze the features extracted from this pre-trained model, and train a linear classifier using supervision for the dataset classification task. This is referred to as the linear probing protocol. We note that in this protocol, *only the linear classifier layer is tunable* under the supervision of the dataset classification labels. Linear probing presents a more challenging scenario.

Table 7 shows the results under the self-supervised protocol. Even with MAE pre-trained on standard ImageNet (which involves *no* YCD images), the model achieves 76.2% linear probing accuracy for dataset classification. In this case, only the linear classifier layer is exposed to the dataset classification data.

Using MAE pre-trained on the same YCD training data, the model achieves higher accuracy of 78.4% in linear probing. Note that although this MAE is pre-trained on the same target data, it has no prior knowledge that the goal is for dataset classification. Nevertheless, the pre-trained model can learn features that are more discriminative (for this task) than those pre-trained on the different dataset of ImageNet. This transfer learning behavior again resembles those observed in semantic classification tasks.

case	accuracy
fully-supervised	82.9
linear probing w/	
MAE trained on IN-1K	76.2
MAE trained on YCD	78.4

Table 7: Self-supervised pre-training, followed by linear probing, achieves high accuracy for dataset classification. Here, we study MAE [19] as our self-supervised pre-training baseline, which uses ViT-B as the backbone. The fully-supervised baseline for dataset classification is with the same ViT-B architecture (82.9%). Results are on the YCD combination.

case	transfer acc
random weights	6.7
Y+C+D	27.7
Y+C+D+W	34.2
Y+C+D+W+L	34.2
Y+C+D+W+L+I	34.8
MAE [19]	68.0
MoCo v3 [5]	76.7

Table 8: Features learned by classifying datasets can achieve nontrivial results under the linear probing protocol. Transfer learning (linear probing) accuracy is reported on ImageNet-1K, using ViT-B as the backbone in all entries. The acronyms follow the first letter of each dataset in Table 2.

4.4 Features Learned by Classifying Datasets

We have shown that models trained for dataset classification can well generalize to unseen validation data. Next we study how well these models can be transferred

to semantic classification tasks. To this end, we now consider dataset classification as a pretext task, and perform linear probing on the frozen features on a semantic classification task (ImageNet-1K classification). Table 8 shows the results of our dataset classification models pre-trained using different combinations of datasets.[2]

Comparing with the baseline of using random weights, the dataset classification models can achieve non-trivial ImageNet-1K linear probing accuracy. Importantly, using a combination of more datasets can increase the linear probing accuracy, suggesting that *better features are learned by discovering the dataset biases across more datasets.*

As a reference, it should be noted that the features learned by dataset classification are significantly worse than those learned by specialized self-supervised learning methods, such as MAE [19], MoCo v3 [5], and others, which is as expected. Nevertheless, our experiments reveal that *the dataset bias discovered by neural network models are relevant to semantic features that are useful for image classification.*

5 User Study

To have a better sense of the dataset classification task, we further conduct a user study to assess how well humans can do this task and to learn their experience.

Settings. We ask our users to classify individual images sampled from the YCD combination. Because users may not be familiar with these datasets, we provide an interface for them to *unlimitedly* browse the training images (with ground-truth labels of their dataset identities) when they attempt to predict every validation image. We ask each user to classify 100 validation images, which do not overlap with the training set provided to them. We do not limit the time allowed to be spent on each image or on the entire test.

Users. A group of 20 volunteer participants participated in our user study. All of them are researchers with machine learning background, among which 14 have computer vision research experience.

User study results. Figure 5 shows the statistics of the user study results on the dataset classification task. In summary, 11 out of all 20 users have 40%-45% accuracy, 7 users have 45%-50%, and only 2 users achieve over 50%. The mean is 45.4% and the median is 44%.

The human performance is higher than the chance-level guess (33.3%), suggesting that there exist patterns that humans can discover to distinguish these datasets. However, the human performance is much lower than the neural network's 84.7%.

We also report that the 14 users who have computer vision research experience on average perform no better than the other users. Among these 14 users, we also

[2] In this comparison, we search for the optimal learning rate, training epoch, and the layer from which the feature is extracted, following the common practice in the self-supervised learning community.

ask the question "What accuracy do you expect a neural network can achieve for this task?" The estimations are 60% from 2 users, and 80% from 6 users, and 90% from 1 user; there were 5 users who chose not to answer. The users made these estimations before becoming aware of our work.

There are 15 participants who describe the difficulty of the task as "difficult". No participant describes the task as "easy". 2 participants commented that they found the task "interesting".

We further asked the users what dataset-specific patterns they have discovered and used to solve this task. We summarize their responses below, in which the brackets indicate how many users mentioned the same pattern:

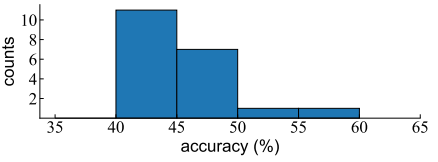

Figure 5: User study results on humans performing the dataset classification task. Humans generally categorize images from YCD with 40-70% accuracy.

- YFCC: people (6), scenery (3), natural lighting, plants, lifestyle (2), real-world, sport, wedding, high resolution (2), darker, most specific, most new, cluttered;
- CC: cartoon (2), animated, clothing sample, product, logo, concept, explanatory texts, geography, furniture, animals, low resolution, colorful, brighter, daily images, local images, single person, realistic, clean background;
- DataComp: white background (3), white space, transparent background, cleaner background, single item (2), product (2), merchandise, logo-style, product showcase, text (2), lots of words, artistic words, ads, stickers, animated pictures (2), screenshots, close-up shot, single person, people, non-realistic icons, cartoon, retro;

In these user responses, there are some simple types of bias that can be exploited (*e.g.*, "white background" for DataComp), which can help increase the user prediction accuracy over chance-level guess. However, many types of the bias, such as the inclusion of "people" in images, are neither sufficient nor meaningful for identifying the images (*e.g.*, all datasets contain images with people presented).

6 Conclusion

We revisit the dataset classification problem in the context of modern neural networks and large-scale datasets. We observe that the datasets bias can still be easily captured by modern neural networks. This phenomenon is robust across models, dataset combinations, and many other settings.

It is worth pointing out that the concrete forms of the bias captured by neural networks remain largely unclear. We have discovered that such bias may contain some generalizable and transferrable patterns, and that it may not be easily noticed by human beings. We hope further effort will be devoted to this problem, which would also help build datasets with less bias in the future.

Acknowledgements. We thank Yida Yin, Mingjie Sun, Saining Xie, Xinlei Chen, and Mike Rabbat for valuable discussions and feedback, and all volunteers for participating in our user study.

References

1. Common Crawl. https://commoncrawl.org.
2. Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arxiv preprint arXiv:1607.06450*, 2016.
3. Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on fairness, accountability and transparency*, 2018.
4. Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12M: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *CVPR*, 2021.
5. Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised Vision Transformers. In *ICCV*, 2021.
6. Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016.
7. Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *CVPR Workshops*, 2020.
8. Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009.
9. Karan Desai, Gaurav Kaul, Zubin Aysola, and Justin Johnson. RedCaps: Web-curated image-text data created by the people, for the people. In *NeurIPS Datasets and Benchmarks Track*, 2021.
10. Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *ICML*, 2014.
11. Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.
12. Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Jacob Steinhardt, and Aleksander Madry. Identifying statistical bias in dataset replication. In *ICML*, 2020.
13. Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The PASCAL Visual Object Classes (VOC) Challenge. *IJCV*, 2010.
14. Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *CVPR workshops*, 2004.
15. Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, et al. Datacomp: In search of the next generation of multimodal datasets. In *NeurIPS Datasets and Benchmarks Track*, 2023.
16. Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *JMLR*, 2016.

17. Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.
18. Google People + AI Research. Know your data. 2021.
19. Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022.
20. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.
21. Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *ICLR*, 2018.
22. Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *CVPR*, 2021.
23. Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.
24. Andrej Karpathy. What I learned from competing against a ConvNet on ImageNet. https://karpathy.github.io/2014/09/02/what-i-learned-from-competing-against-a-convnet-on-imagenet/, 2014. Accessed: October 21, 2023.
25. Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *ICML*, 2021.
26. Alex Krizhevsky, Ilya Sutskever, and Geoff Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012.
27. Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *IJCV*, 2020.
28. John Lambert, Zhuang Liu, Ozan Sener, James Hays, and Vladlen Koltun. Mseg: A composite dataset for multi-domain semantic segmentation. In *CVPR*, 2020.
29. Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998.
30. Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.
31. Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *CVPR*, 2022.
32. David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004.
33. Thao Nguyen, Gabriel Ilharco, Mitchell Wortsman, Sewoong Oh, and Ludwig Schmidt. Quality not quantity: On the interaction between dataset design and robustness of clip. *NeurIPS*, 2022.
34. Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics & Image Processing*, 2008.
35. Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *CVPR*, 2012.
36. Vinay Uday Prabhu and Abeba Birhane. Large image datasets: A pyrrhic win for computer vision? *arXiv preprint arXiv:2006.16923*, 2020.
37. Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

38. Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *ICML*, 2019.
39. Lawrence Gilman Roberts. *Machine Perception of Three-Dimensional Solids*. PhD thesis, Massachusetts Institute of Technology, 1963.
40. Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 2015.
41. William F Schreiber. Image processing for quality improvement. *Proceedings of the IEEE*, 1978.
42. Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade W Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa R Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. LAION-5b: An open large-scale dataset for training next generation image-text models. In *NeurIPS Datasets and Benchmarks Track*, 2022.
43. Shreya Shankar, Yoni Halpern, Eric Breck, James Atwood, Jimbo Wilson, and D Sculley. No classification without representation: Assessing geodiversity issues in open data sets for the developing world. *arXiv preprint arXiv:1711.08536*, 2017.
44. Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.
45. Krishna Srinivasan, Karthik Raman, Jiecao Chen, Michael Bendersky, and Marc Najork. Wit: Wikipedia-based image text dataset for multimodal multilingual machine learning. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021.
46. Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *ICCV*, 2017.
47. Yu Sun, Xiaolong Wang, Liu Zhuang, John Miller, Moritz Hardt, and Alexei A. Efros. Test-time training with self-supervision for generalization under distribution shifts. In *ICML*, 2020.
48. Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, 2015.
49. Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. YFCC100M: The new data in multimedia research. *Communications of the ACM*, 2016.
50. Tatiana Tommasi, Novi Patricia, Barbara Caputo, and Tinne Tuytelaars. A deeper look at dataset bias. *arxiv preprint arXiv:1505.01257*, 2015.
51. Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *CVPR*, 2011.
52. Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. *arxiv preprint arXiv:2012.12877*, 2020.
53. Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.
54. CWJ van Miltenburg. Stereotyping and bias in the flickr30k dataset. In *Workshop on multimodal corpora: computer vision and language processing*, 2016.
55. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

56. Andrea Vedaldi and Andrew Zisserman. Efficient additive kernels via explicit feature maps. *IEEE transactions on pattern analysis and machine intelligence*, 2012.
57. Hermann Von Helmholtz. *Optique physiologique*. Masson, 1867.
58. Angelina Wang, Alexander Liu, Ryan Zhang, Anat Kleiman, Leslie Kim, Dora Zhao, Iroha Shirai, Arvind Narayanan, and Olga Russakovsky. REVISE: A tool for measuring and mitigating bias in visual datasets. *IJCV*, 2022.
59. Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *ICLR*, 2021.
60. Zeyu Wang, Klint Qinami, Ioannis Christos Karakozis, Kyle Genova, Prem Nair, Kenji Hata, and Olga Russakovsky. Towards fairness in visual recognition: Effective strategies for bias mitigation. In *CVPR*, 2020.
61. Kaiyu Yang, Klint Qinami, Li Fei-Fei, Jia Deng, and Olga Russakovsky. Towards fairer datasets: Filtering and balancing the distribution of the people subtree in the imagenet hierarchy. In *Conference on fairness, accountability and transparency*, 2020.
62. Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *arxiv preprint arXiv:1411.1792*, 2014.
63. Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, 2019.
64. Oliver Zendel, Katrin Honauer, Markus Murschitz, Daniel Steininger, and Gustavo Fernandez Dominguez. Wilddash-creating hazard-aware benchmarks. In *ECCV*, 2018.
65. Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. Mitigating unwanted biases with adversarial learning. *arXiv preprint arXiv:1801.07593*, 2018.
66. Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.
67. Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018.
68. Dora Zhao, Angelina Wang, and Olga Russakovsky. Understanding and evaluating racial biases in image captioning. In *ICCV*, 2021.
69. Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 2017.

A Implementation Details

For image-text datasets (CC, DataComp, WIT, LAION), we only use their images. The LAION dataset was filtered before usage. We uniformly sample the same number of images from each dataset to form the train / val sets for dataset classification. If a dataset already has pre-defined train / val splits, we only sample from its train split. 1M images for each dataset is used as the default unless otherwise specified. This is not a small collection, yet it still only represents a tiny portion of images (e.g., <10%) for most datasets we study. To speed up image loading, the shorter side of each image is resized to 500 pixels if the original shorter side is larger than this. We observe this has minimal effect on the performance of models.

We train the models for the same number of samples seen as in a typical 300-epoch supervised training on ImageNet-1K classification [31], regardless of the number of training images. This corresponds to the same number of iterations as in [31] since the same batch size is used. The complete training recipe is shown in Table 9.

config	value
optimizer	AdamW
learning rate	1e-3
weight decay	0.3
optimizer momentum	β_1, β_2=0.9, 0.95
batch size	4096
learning rate schedule	cosine decay
warmup epochs	20 (ImageNet-1K)
training epochs	300 (ImageNet-1K)
augmentation	`RandAug` (9, 0.5) [7]
label smoothing	0.1
mixup [67]	0.8
cutmix [63]	1.0

Table 9: Training settings for dataset classification.

For the linear probing experiments on ViT-B in Section 4.3 and 4.4, we follow the settings used in MAE [19]. For Section 4.4, we use the checkpoint from epoch 250, and sweep for a base learning rate from {0.1, 0.2, 0.3}, and a layer index for extracting features from {8, 9, 10}.

During inference, an image is first resized so that its shortest side is 256 pixels, maintaining the aspect ratio. Then the model takes its 224×224 center crop as input. Therefore, the model cannot directly exploit the different distributions of resolutions and/or aspect ratios for different datasets as a shortcut for predicting images' dataset identities. The model takes randomly augmented crops of 224×224 images as inputs in training.

B Additional Results

Training Curves. We plot the training loss and validation accuracy for ConvNeXt-T YCD classification in Figure 6. The training converges quickly to a high accu-

racy level in the initial phases. This again demonstrates neural networks' strong capability in capturing dataset bias.

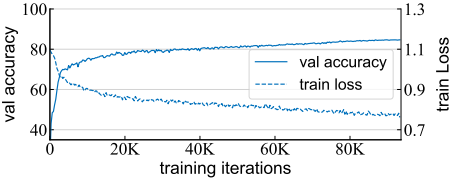

Figure 6: Training curves for YCD classification. The model converges quickly.

ImageNet vs. ImageNetV2. ImageNetV2 [38] attempts to create a new validation set trying to follow the exact collection process of ImageNet-1K's validation set. As such, the images look very much alike. We find a classifier could reach 81.8% accuracy classifying ImageNetV2 and ImageNet-1K's validation set, substantially higher than 50%, despite only using 8K images for training from each. This again demonstrates how powerful neural networks are at telling differences between seemingly close image distributions.