The Dog Ate Our Benchmark: Generalizability and Leakage in Fine-Grained Image Classification Benchmarking

Anonymous authors
Paper under double-blind review

Abstract

Dog breed identification is a classic benchmark in fine-grained image classification (FGIC). Several recent studies have reported top-1 accuracies over 90%. To examine whether these results generalize across benchmarks within the same domain, we replicated a recent workflow that achieved 94.45% accuracy on Stanford Dogs and extended it to the Columbia Dogs and Tsinghua Dogs benchmarks. Performance dropped to 91.20% and 87.17%, respectively, suggesting that high accuracy on Stanford Dogs does not necessarily reflect broad domain generalization.

The replicated workflow used frozen features from four convolutional neural networks (CNNs), all distributed as ImageNet-1K-pretrained models. Because Stanford Dogs is derived from ImageNet, we audited image provenance relative to the ImageNet-1K train split using SHA-256 hashing. We found 100% of the unique images in the Stanford Dogs train and test splits were byte-identical to ImageNet-1K training images. This overlap also affects Tsinghua Dogs, where 23.8% of training images and 59.6% of validation images appear in ImageNet-1K train. Overlap in Columbia Dogs was much lower, at 1.5% across the analyzed splits. These results support leakage as a possible explanation for cross-benchmark performance differences.

We further tested this explanation by replacing the CNN feature extractors with OpenCLIP and DINOv2, models with different pretraining provenance. These models achieved higher accuracy on Columbia Dogs than on either ImageNet-1K-contaminated benchmark. These findings raise concerns about a common FGIC practice: combining ImageNet-1K pretraining with evaluation on ImageNet-derived benchmarks. Our provenance audit shows that estimates of FGIC accuracy on dog breed benchmarks can reflect leakage between evaluation sets and pretraining corpora rather than independent task generalization.

1 Introduction

Dog breed identification has long been used as a fine-grained image classification benchmark because dog breeds combine subtle between-class differences with substantial within-class variation. Early work used detection, part localization, and landmark-based representations (Liu et al., 2012; Wang et al., 2014); later work shifted toward convolutional neural network (CNN) architectures, pretrained representations, attention mechanisms, and downstream optimization. In many cases, the CNNs evaluated were pretrained on ImageNet (Deng et al., 2009), a large-scale image database organized according to the WordNet (Miller, 1995) hierarchy. However, ImageNet pretraining raises a benchmark-validity question because dog-breed benchmarks are themselves derived from ImageNet.

Here, we investigate the generalizability of research in FGIC when using ImageNet-pretrained models and ImageNet-derived benchmarks. We contextualize the common dog breed benchmarking datasets within the broader ecosystem of visual data, taking a step back from architecture-level optimization to examine current research from a benchmark-validity perspective. In particular, we ask whether these benchmarks offer independent evaluations of model performance and how shared provenance between pretraining corpora and evaluation benchmarks affects published results and their interpretation.

1.1 Contributions

This paper makes three main contributions. First, we apply a recent high-performing multi-model workflow (Cui et al., 2024) to multiple dog breed classification benchmarks to demonstrate that the high performance achieved on Stanford Dogs by CNN-based workflows may not generalize to other datasets. Second, using exact file-level hashing to identify byte-identical duplicate image files, we reveal non-independence among dog breed benchmarks and demonstrate varying amounts of data leakage from ImageNet pretraining into these benchmarks, including into the test splits. Third, we reveal an interactive effect between model pretraining and dataset provenance using an ablation design, demonstrating that apparent model performance is dependent on the provenance of both the model and the benchmark itself. Together, these analyses show how benchmark construction, dataset reuse, and pretrained representations can shape apparent progress in FGIC.

2 Background and Related Works

2.1 Datasets for FGIC: History and Curation

A pivotal development in the history of FGIC in the age of deep learning was the development of ImageNet (Deng et al., 2009), a large-scale corpus of images often used to pretrain transfer learning models for computer vision (Ridnik et al., 2021). To build ImageNet, reviewers examined 80,000 concepts across various levels of specificity (e.g., at the level of species (*Felis catus*) and at the level of breed (e.g., Burmese cat) (Deng et al., 2009). These reviewers were recruited through Amazon's Mechanical Turk, a marketplace that crowds-sources human judgments by providing monetary compensation to people for completing tasks. ImageNet is fundamentally linked to dog breed identification because it was subsequently used to develop the benchmarking datasets used in this domain.

Stanford Dogs. Stanford Dogs was developed in 2011 and is explicitly a subset of ImageNet (Khosla et al., 2011). Specifically, the curators examined ImageNet's dog (*Canis familiaris*) class and extracted all child nodes that were also leaves. Classes that could include multiple breeds, like "working dog" or "puppy", were then excluded, as were low-quality images and images smaller than 200 x 200 pixels. Each image was checked "to confirm whether or not it matched images from Wikipedia and shared similar features to the other images in the same category" (Khosla et al., 2011). In the end, the dataset included 20,580 images representing 120 classes (breeds), with between 150 and 200 images per breed.

Columbia Dogs. Soon after, in 2012, the Columbia Dogs with Parts dataset was developed by retrieving images from the photo-sharing site Flickr and Google image search (Liu et al., 2012). Curators combined these images with images from ImageNet. The only filtering they undertook was to remove images that did not include both of a dog's eyes. The breed label for each image was determined by ratings from "multiple" Mechanical Turk workers, with the final dataset including 8,351 images across 133 AKC-recognized breeds.

Tsinghua Dogs. Released in 2020, Tsinghua Dogs was designed to be a benchmarking dataset for dog breed classification in the age of deep learning (Zou et al., 2020). This dataset contains 70,428 images spanning 130 dog breeds. Image collection was achieved with a "data capture system" involving user-uploaded photos from three Chinese cities. Additional images were collected from image searches, Baidu (a social networking site), and ImageNet. Low-quality images were filtered out, and images were deduplicated based on structural similarity as defined by Wang et al. (2004). Images were assigned a breed label via "expert review" and using an active learning framework (Zou et al., 2020). The curators intended their dataset to represent Chinese dog breed diversity both in terms of which breeds are represented and the number of images of each breed, ranging from 200 to nearly 7,500.

2.2 Recent Advances in FGIC

Dog breed classification has been a major benchmarking task used to evaluate model architectures and optimization for the last 15 years. As the methodologies used for FGIC have evolved, task performance

Table 1: Selected reports of high top-1 accuracy on dog breed benchmarking tasks across datasets

Citation	Approach	Dataset	Accuracy
Ráduly et al. (2018)	CNN (Inception-ResNet-v2)	Stanford Dogs	90.69%
Borwarnginn et al. (2019)	CNN (Inception V3)	Columbia Dogs	96.75%
Haswar et al. (2022)	CNN (architecture unspecified)	Columbia Dogs	97.81%
Cui et al. (2024)	CNN (multi-model feature extraction) + PCA + feature selection + SVM	Stanford Dogs	94.45%
Khan et al. (2025)	YOLOv8N + dual-stream Vision Transformer	Tsinghua Dogs	90.04%

accuracies have increased, with several recent studies reporting top-1 accuracies above 90%. Papers that achieved such results without manual curation and while benchmarking on a full dataset are summarized in Table 1. We were able to locate publicly available code for one of these studies: Cui et al. (2024), which achieved a maximum accuracy of 94.45% on the full Stanford Dogs dataset using feature extraction from multiple CNN architectures. Our analysis replicates and extends the work of Cui et al. (2024).

3 Experimental Resources

3.1 Target Workflow

Cui et al. (2024) evaluated the performance of several hybrid models on the Stanford Dogs dataset. For their highest-performing model, they combined frozen weights from several CNN architectures: Inception V3 (Szegedy et al., 2016), NASNet (Zoph et al., 2018), PNASNet (Liu et al., 2018), and Inception-ResNet-v2 (Szegedy et al., 2017). Each model was used to extract features from the Stanford Dogs training and test datasets (separately), and they used principal component analysis (PCA) to reduce the number of features to 4,000. Next, they used a Grey Wolf Optimizer (GWO) (Mirjalili et al., 2014) to select the most informative features, following Basak et al. (2021), and reported the highest accuracy achieved for each run using an RBF-kernel SVM with probability estimates enabled trained on the selected features and evaluated on the held-out test set.

For our analysis, we used the code for the top-performing model published in their repository[1]. However, we noted some discrepancies between the available code and the methods described in the manuscript. The released script matches the hyperparameters described in the paper for PCA, GWO, and SVM, but it does not fully match the described CNN preprocessing and fine-tuning procedure. In particular, the released script appears to perform feature extraction from ImageNet-pretrained CNNs rather than loading fine-tuned model weights, and it applies a uniform 331×331 resize without the paper-described normalization or model-specific input sizes. Because our goal was to reproduce the executable workflow made available by the authors, when the manuscript and released code conflicted, we treated the released code as the primary operational specification. Our first objective was therefore to ensure that we could achieve a prediction accuracy consistent with what was reported in the published analysis.

We downloaded the weights for InceptionV3 from the source identified by Cui et al. (2024) (`http://data.lip6.fr/cadene/pretrainedmodels`) but were unable to download the remaining models from this resource due to SSL certificate issues. Instead, the pretrained weights for NASNet-A (Large), PNASNet-5-Large, and InceptionResNetV2 were downloaded from Kaggle (`https://www.kaggle.com/datasets/mhiro2/pytorch-pretrained-models`.) Outputs from each model's feature extractor were adaptively average-pooled, flattened, and concatenated per the published workflow. We used 4,000 dimensions for

[1]Model `nasnet_pnasnet_inceptionresnetv2_inception_v3_svm_PCA_GWO.py` from `https://download.big.ac.cn/idog/dogvc/code/`

PCA and 30 search agents, 5 iterations, and 30 independent runs for GWO. Finally, we examined the highest accuracy achieved across all thirty runs.

In line with the published results, the maximum test accuracy obtained was 94.60%. While this does not exactly match the 94.45% accuracy reported under these conditions by Cui et al. (2024), the difference is negligible, especially in context of the fact that no seed was provided for the non-deterministic GWO step.

3.2 Refactoring the Target Workflow

In the original code, the entire pipeline was run from a single script. We refactored the code to mitigate some of the challenges of running multiple trials and to support experimentation via modular changes to configuration, streamlined recovery, and decreased runtime. First, we split the script into five phases run from a single harness: feature extraction, feature concatenation, dimensionality reduction, GWO, and prediction. Each phase included the checkpointing of temporary files, allowing the pipeline to resume from any phase. We also turned off shuffling in the torch loader to support reproducibility and moved all identified modifiable parameters out of the scripts themselves and into a `config.yml` file. We configured the feature extraction step to use a GPU if available via `torch.device` and added some unit testing (e.g., to ensure that the train and test lists were read in with the same class ordering). Finally, we added random seeding to the GWO step to allow for consistency across trials and parallelized the independent optimization runs under a map-reduce paradigm to make use of the computational resources available.

We ran the refactored pipeline using the same parameters as in the original code from Cui et al. (2024). The reported accuracy was 94.42%. A second run with a different random seed returned an accuracy of 94.62%. Based on these results, we were satisfied that the refactor had a negligible effect on performance proportional to the stochasticity of the code, as these closely matched the published value (Cui et al., 2024) and our original replication using their code, respectively. For all experiments, we assumed the refactored pipeline to be a representative approximation of the workflow developed by Cui et al. (2024).

3.3 Datasets

In our analysis, we work with three different datasets. The first is Stanford Dogs, which was the original data source used by Cui et al. (2024). We downloaded Stanford Dogs on February 7, 2023 from the Stanford Dogs website[2].

We also examined two additional datasets: Tsinghua Dogs and Columbia Dogs. Tsinghua Dogs was downloaded on November 4, 2024 following the URL[3] from the paper (Zou et al., 2020). The URL[4] identified in the Columbia Dogs paper Liu et al. (2012) is no longer publicly accessible. We therefore based our analyses of this dataset on a locally archived copy that was downloaded on February 8, 2023 from Udacity via `https://github.com/udacity/dog-project`[5].

For each dataset, we calculated the number of images in the train and test sets, as well as the per-class splits (Appendix A). Stanford Dogs and Tsinghua Dogs both have two splits that are divided 80/20 and 93/7, respectively. The Columbia Dogs distribution used in our analysis included separate train, validation, and test splits. We combined the validation and test splits into a single split for testing, resulting in an 80/20 train/test split.

3.4 Computing Infrastructure

All classification experiments were run on a university high-performance computing cluster with CPU and GPU nodes. The analyses used Python 3.10.13 in a Conda environment using PyTorch 2.5.1, Torchvision 0.20.1, Pillow 9.4.0, NumPy 2.2.6, scikit-learn 1.7.2, Matplotlib 3.9.1, tqdm 4.67.3, and `pretrainedmodels`

[2] `http://vision.stanford.edu/aditya86/ImageNetDogs/`

[3] `https://cg.cs.tsinghua.edu.cn/ThuDogs/`

[4] `faceserv.cs.columbia.edu/DogData/`

[5] Though the dataset is no longer publicly available through this repository, it is still hosted at `https://s3-us-west-1.amazonaws.com/udacity-aind/dog-project/dogImages.zip` as of the time of writing

Table 2: Comparison of accuracies obtained across datasets. Full/Original reports the maximum test accuracy observed under each dataset's original train/test configuration. Subsample statistics summarize results across 10 subsamples, each containing 25 train and 6 test images per class.

	Full/Original	Accuracy Over 10 Subsamples			
Dataset	**Accuracy**	**Mean**	**SD**	**Min.**	**Max.**
Columbia	91.20%	90.23%	1.00%	88.35%	91.73%
Stanford	94.62%	94.57%	0.98%	92.92%	95.83%
Tsinghua	87.17%	86.74%	1.01%	85.51%	88.72%

0.7.4. Cui et al. (2024)'s GWO feature-selection code used modules downloaded from Basak et al. (2021); we also downloaded them from the associated repository[6].

3.5 Data Availability

After the double-blind phase, all of the code, run configurations, and final run outputs used in these analyses will be made publicly available via GitHub.

4 Experiment 1: Generalizability of Prior Work

We first tested whether the Stanford Dogs performance reported by Cui et al. (2024) generalized to other dog-breed benchmarks. We ran the replicated workflow on Stanford Dogs, Columbia Dogs, and Tsinghua Dogs using each dataset's original splits. We then repeated the analysis under matched sample-size conditions, using 10 random subsamples with 25 training and 6 test images per class, to determine whether cross-dataset differences persisted after controlling for dataset size and split structure.

4.1 Approach

In this analysis, we re-ran the reproduced pipeline described above on two additional dog breed benchmarking datasets, Columbia Dogs and Tsinghua Dogs. For each dataset, we conducted two experiments. First, we ran the pipeline on the full datasets using the published train/test splits (Table 4), except that we combined the Columbia Dogs `test` and `valid` splits. We used 4,000 PCs, corresponding to the highest-performing condition reported by Cui et al. (2024).

Given the differences in the sizes of the datasets as well as their component splits and classes, we wanted to test whether the experimental design contributed to differences in performance. Therefore, we ran the pipeline on each dataset again while standardizing the number of images across classes and across datasets via subsampling. For 10 randomly seeded trials, 25 train and 6 test images were randomly sampled from each dataset. Because this analysis was designed to compare datasets under matched sample-size conditions rather than to reproduce the original benchmark splits, subsamples were drawn from all images available for each class. The 25/6 train/test split was selected based on the minimum number of images per class across all three datasets (Appendix A). We then compared the distribution of the 10 maximum accuracies for each dataset (mean, standard deviation, minimum, and maximum) against the overall maximum accuracy achieved on the dataset as a whole.

4.2 Results

Running the full dataset with the original train/test splits revealed that the pipeline did not achieve the high accuracy seen on Stanford Dogs with either of the other benchmarks (Table 2). These differences persisted even when the number of images was standardized across datasets (Figure 1): across the ten

[6]We used commit `6c23e1ea719e0e3d69155fd9d9397a4d14f6fe1` from `https://github.com/DVLP-CMATERJU/Two-Step-Feature-Enhancement`

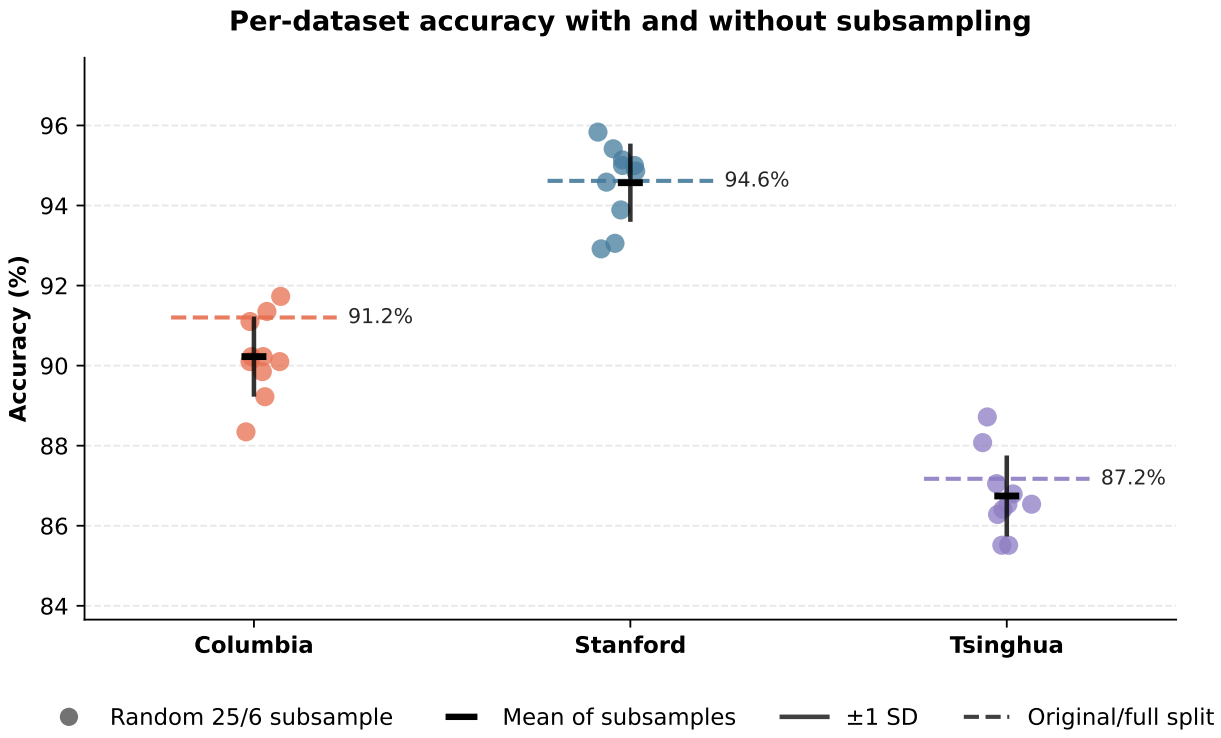


Figure 1: To quantify the effect of the differences in dataset size, we compared the performance on each dataset as a whole (dashed line) against standardized subsamples of 25 train and 6 test images per class. The trends in cross-dataset accuracy persisted under standardized split sizes.

trials subsampled from each of the datasets, the mean maximum accuracies across trials followed the same trend as the maximum accuracies from the full-dataset analyses (Table 2; Figure 1). The overall estimated accuracy for Stanford (94.62%) was the closest to the mean across trials (94.57%), but across all three datasets, the standard deviations were approximately 1% and the whole-dataset maximum accuracy fell within ± 1 standard deviation of the mean. This analysis suggested that the differences in performance across the full datasets were not driven by differences in the structure or size of the train/test splits, given that the differences persisted even when the splits were standardized. The fact that this workflow achieved a higher performance on Stanford Dogs even under standardized conditions thus cannot be explained simply by dataset characteristics such as training set size, test set size, or train/test proportions.

5 Experiment 2: Auditing Dataset Artifacts

To contextualize these cross-benchmark performance differences, we investigated the provenance of the images in the datasets. Stanford Dogs is derived directly from ImageNet (Khosla et al., 2011), so we wanted to establish whether the Stanford Dogs test set remained strictly held out for models pretrained on ImageNet.

Many FGIC pipelines use CNN architectures distributed with pretrained weights. One of the most common pretraining corpora is ImageNet-1K, which is the 1,000-class benchmark associated with the ImageNet Large Scale Visual Recognition Challenge 2012 (Russakovsky et al., 2015). Because ImageNet can refer either to the broader WordNet-organized image database or to the 1,000-class ILSVRC classification benchmark commonly used for supervised pretraining, we distinguish ImageNet from ImageNet-1K throughout this analysis.

The four CNN feature extractors used in Experiment 1 are distributed as ImageNet-pretrained models, though the architecture papers and downstream model artifacts vary in how precisely they document the

provenance of the released weights. Inception-v3 and Inception-ResNet-v2 were introduced and evaluated in the context of ILSVRC/ImageNet classification (Szegedy et al., 2016; 2017), while NASNet and PNASNet were developed through architecture-search workflows that included ImageNet-scale evaluation (Zoph et al., 2018; Liu et al., 2018). Downstream model documentation identifies the artifacts used here as ImageNet-1K pretrained models (Google, 2020d;c;b;a). We therefore treat the feature extractors as ImageNet-1K-pretrained for purposes of leakage analysis, but note that finer-grained artifact provenance could not be independently verified.

The goal of this analysis was to identify potential data leakage between the pretraining and benchmarking stages. Leakage complicates interpretation of accuracy results and could be a contributing factor to the observed cross-dataset performance patterns (Figure 1). We therefore investigated the relationships among the three benchmarking datasets and with ImageNet-1K in order to characterize data leakage at the levels of dataset and split. We also examined leakage at the split level because there are different interpretability implications to leakage from pretraining into feature extraction for the train versus test splits.

5.1 Approach

To characterize overlap between ImageNet-1K training data and the benchmarking datasets, we identified exact file-level duplicates by computing SHA-256 hashes over the raw bytes of each file. Because SHA-256 is collision-resistant, the probability of two distinct image files producing the same hash is negligible; we therefore treated matching SHA-256 hashes as evidence that two files were byte-identical duplicates. This procedure was designed to detect exact file-level duplicates and does not detect visually identical or near-duplicate images that have been resized, cropped, recompressed, or otherwise modified. Therefore, our overlap estimates should be interpreted as conservative with respect to visual duplication because they exclude near-duplicates and transformed copies.

We used the class labels from Stanford Dogs to identify the WordNet IDs (wnid) associated with dog breeds, per the methods of Khosla et al. (2011). We then extracted only those synsets from the ImageNet-1K training split for comparison to the benchmarking datasets. We computed the hashes for each dataset (Stanford Dogs, Columbia Dogs, Tsinghua Dogs, and the selected synsets of the ImageNet-1K training split) and all available splits within Stanford Dogs, Columbia Dogs, and Tsinghua Dogs using the `sha256` function from Python's `hashlib` library (Python Software Foundation, 2026). Files were read incrementally in 1 MiB chunks, and the resulting hashes were stored as hexadecimal strings using `hexdigest`. We retained the hash, dataset name, split, class label, and file path for each image.

We then conducted pairwise comparisons of sets of image hashes. The hexadecimal strings representing the images in each dataset were loaded into Python `set` objects. For each pair of datasets, we computed the intersection of the sets. For dataset-level comparisons with ImageNet-1K, percent overlap was calculated as the number of shared hashes divided by the number of unique hashes in the benchmarking dataset. We also quantified overlap between individual splits and the dog synsets of ImageNet-1K train by computing the intersection of pairs of sets. Finally, overlap among benchmarking datasets was visualized as a Venn diagram using the `venn` functionalities of `matplotlib` (Hunter, 2007). We computed mutually exclusive Venn regions using set intersections and set differences over SHA-256 hashes.

5.2 Results

Comparison of unique SHA-256 hashes revealed substantial overlap among datasets. Most notably, 100% of the Stanford Dogs test split was byte-identical to images in the ImageNet-1K training split, as was the Stanford Dogs training split. For Tsinghua Dogs, the dataset-level proportion was lower but still non-trivial (18,559 out of 70,310 unique hashes in the dataset as a whole, or 26.4%). In Columbia Dogs, on the other hand, only 127 of 8,345 unique hashes in the dataset (1.5%) were identified in the ImageNet-1K training set. These three datasets therefore vary significantly in the risk of leakage from pretraining into benchmarking.

Additionally, individual splits, even within datasets, varied significantly in the fraction of images overlapping ImageNet-1K (Table 3). As indicated above, 100% of the images across both splits in Stanford Dogs were present in the ImageNet-1K training data. For Columbia Dogs, the rates of overlap with ImageNet-1K were

Table 3: Exact file-level overlap between dog-breed benchmark splits and ImageNet-1K train. Counts of byte-identical image files within each split are shown. Four images present in the Tsinghua image archive did not appear in the provided train or validation manifests and were therefore treated as unlisted in split-level analyses.

Dataset	Split	Hashes in ImageNet-1K	Unique Hashes in Split	Overlap
Stanford Dogs	Train	11,960	11,960	100.0%
Stanford Dogs	Test	8,568	8,568	100.0%
Columbia Dogs	Train	100	6,675	1.5%
Columbia Dogs	Test	9	836	1.1%
Columbia Dogs	Validation	18	835	2.2%
Tsinghua Dogs	Train	15,468	65,120	23.8%
Tsinghua Dogs	Validation	3,099	5,199	59.6%
Tsinghua Dogs	Unlisted	0	4	0.0%

low across all three splits. In Tsinghua Dogs, ImageNet-overlapping files represented 23.8% of the training split but 59.6% of the validation split.

The benchmarking datasets themselves overlapped to varying extents (Figure 2). Once again, Columbia Dogs was the most distinct from other datasets. Only 118 images were shared by all three datasets, and Columbia also shared an additional 7 images with Stanford Dogs. On the other hand, a large subset of Stanford Dogs (91%) also appeared in Tsinghua Dogs, though the majority of the Tsinghua Dogs dataset (74%) is not shared with any of the other datasets.

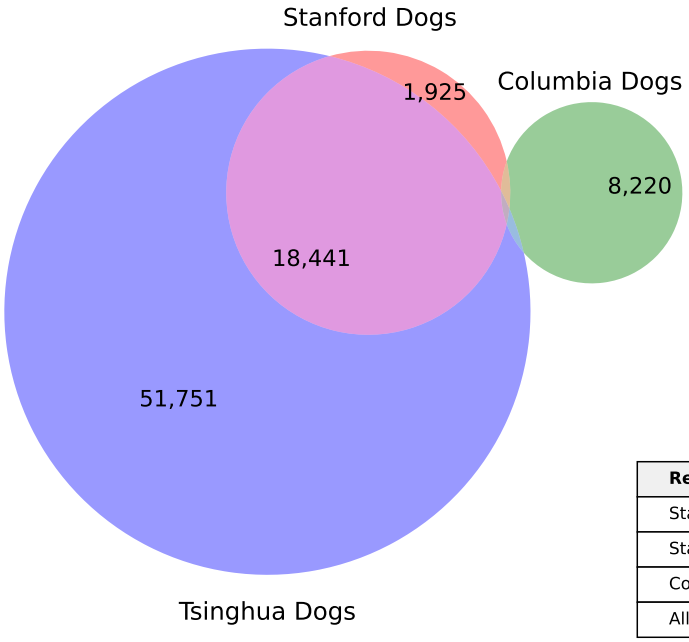
Because Table 3 reports unique SHA-256 hashes within each split, while Figure 2 reports unique SHA-256 hashes after collapsing across each dataset, the totals differ when the same byte-identical image appears in more than one split of the same dataset. This pattern revealed cross-split duplicates in all three datasets: 37 duplicate hashes between the Stanford Dogs train and test splits, 58 duplicate hashes between the Tsinghua Dogs train and validation splits, and 1 duplicate hash across Columbia Dogs splits. Thus, in addition to overlap with ImageNet-1K, all three benchmarks exhibited some degree of within-dataset leakage across splits.

This experiment revealed leakage at two levels: between ImageNet-1K pretraining and benchmarking and between splits within each benchmark. The overlap is particularly notable for Stanford Dogs: there is 100% overlap with ImageNet-1K train, and 37 byte-identical images appeared in both the Stanford training and test splits. Therefore, for any CNN feature extractor pretrained on the standard ImageNet-1K training split, the Stanford Dogs test set contains images encountered during pretraining. This pattern combined with the substantially higher performance of the target workflow on Stanford Dogs than on the other datasets suggested data leakage could plausibly contribute to inflated accuracy estimates when benchmarking on Stanford Dogs.

This explanation is especially plausible when comparing Stanford Dogs to Columbia Dogs, which had negligible overlap with ImageNet-1K. Tsinghua Dogs presented a more complex case: ImageNet-1K overlap was substantially higher in the validation split than in the training split. This result reveals that leakage is not merely a dataset-level property; because the validation split has substantially higher ImageNet-1K overlap than the training split, benchmark performance may also be affected by split-specific provenance differences. Thus, after this audit, leakage remained a potential explanation for at least some of the patterns in performance observed in Experiment 1, especially the high performance of Stanford Dogs relative to Columbia Dogs.

6 Experiment 3: Characterizing Performance Artifacts of Pretraining Corpora

To investigate the effects of the leakage from ImageNet-1K training into the benchmarking datasets, we used an ablation design to examine whether Stanford Dogs' performance advantage would persist when the feature extractors had different pretraining provenance. We were particularly interested in the patterns of



Region	Count
Stanford ∩ Columbia only	7
Stanford ∩ Tsinghua only	18,441
Columbia ∩ Tsinghua only	0
All three	118

Figure 2: Exact file-level overlap among dog breed benchmark datasets. Counts indicate mutually exclusive regions of byte-identical image files identified using SHA-256 hashes. The large Stanford–Tsinghua overlap is primarily ImageNet-mediated; small overlap regions are reported in the inset table.

performance for the Stanford versus Columbia datasets. We compared cross-dataset performance using two additional image classification models: Contrastive Language–Image Pretraining (CLIP) (Radford et al., 2021) and DIstillation with NO labels version 2 (DINOv2) (Oquab et al., 2024). This design interrogated whether leakage between the pretraining and testing phases contributed to performance differences seen in Experiment 1.

CLIP CLIP is a weakly supervised framework that offers a complementary approach to the CNNs traditionally evaluated for dog breed classification. We used the OpenCLIP implementation of CLIP (Ilharco et al., 2021), specifically a `convnext_base_w` checkpoint pretrained on data from LAION-2B, which is an image–text corpus constructed from web data retrieved by Common Crawl (`https://commoncrawl.org`) (Schuhmann et al., 2022). While we make no claim that LAION-2B is completely independent from ImageNet, we chose this model because its pretraining corpus is not the ImageNet-1K supervised training dataset implicated in the leakage observed in Experiment 2. This reduced dependence on ImageNet allows us to contrast OpenCLIP's performance across datasets with the patterns seen in Experiment 1 to examine whether Stanford Dogs remained advantaged. We predicted that Stanford Dogs (100% leakage) would show less of an advantage when features were extracted with OpenCLIP, specifically in comparison to the Columbia dataset (negligible leakage).

DINOv2 DINOv2 is a self-supervised approach to image representation learning that was pretrained on LVD-142M, a curated corpus constructed by retrieving web images visually close to images from several curated datasets, including ImageNet-22K, the ImageNet-1K training split, Google Landmarks, and several fine-grained datasets (Oquab et al., 2024). Although the DINOv2 pipeline removed near-duplicates of images

in the validation and test sets of the benchmarks evaluated in Oquab et al. (2024), its use of ImageNet-derived and fine-grained datasets as retrieval anchors raises a provenance question in the context of dog-breed benchmarks that are themselves derived from ImageNet. We therefore treated DINOv2 as an intermediate case: unlike the CNN workflow, it was not trained as a supervised ImageNet-1K classifier, but its pretraining corpus was constructed using ImageNet-derived retrieval anchors. In this analysis, we therefore sought to compare trends in performance across benchmarking datasets when using feature extractors with and without documented ImageNet-1K-supervised pretraining.

6.1 Approach

In this analysis, we used the same target workflow adapted from Cui et al. (2024) in Experiment 1 but employed an alternative model (OpenCLIP or DINOv2) for feature extraction. For the alternative feature extractors, the PCA dimensionality was set to 640 because the OpenCLIP embedding contained 640 features; this allowed the same downstream GWO/SVM workflow to be applied to both OpenCLIP and DINOv2 without changing the OpenCLIP representation size. For the CNN workflow, we report the results from Experiment 1 using the 4,000-PC configuration corresponding to the highest-performing condition reported by Cui et al. (2024). Otherwise, we kept the workflow configuration as similar as possible to Experiment 1. Because our goal was not to optimize or directly rank models, but rather to compare within-model dataset performance patterns, we did not conduct additional hyperparameter tuning.

In this experiment, we evaluated two pretrained vision models: OpenCLIP and DINOv2. For OpenCLIP, we used the `open_clip` implementation distributed through the `open-clip-torch` package. Specifically, we evaluated the `convnext_base_w` architecture with weights pretrained on `laion2b_s13b_b82k`. For DINOv2, we retrieved the `vitb14` model from `facebookresearch/dinov2` using the PyTorch Hub interface. In both cases, pretrained weights were used as fixed feature extractors: model parameters were not fine-tuned on any of the dog datasets. Images were resized and normalized according to preprocessing parameters specified in the experiment configuration. For the replicated CNN workflow, these parameters were chosen to match Cui et al. as closely as possible. We used the same configured input size for DINOv2, applying bicubic resizing, center cropping, and ImageNet mean/std normalization. We retained the Cui et al. (2024) input size for DINOv2 to isolate the effect of feature-extractor provenance while minimizing changes to the surrounding pipeline. For OpenCLIP, preprocessing used the transform returned by `open_clip.create_model_and_transforms` for the specified model and pretrained weights.

We extracted image-level feature embeddings from each model and used these embeddings as inputs for the downstream classification pipeline (Experiment 1). The original train/test splits associated with each dataset were used, except that the Columbia Dogs `test` and `valid` were combined into a single `test` split. Accuracy was computed as in Experiment 1, by recording the highest accuracy achieved across the 30 GWO runs.

6.2 Results

The relative performance of the alternative models over the three datasets revealed an interaction between model and dataset that deviated from the pattern observed in Experiment 1. The clearest interaction appears in model behavior across Stanford and Columbia. In Experiment 1, the ImageNet-pretrained CNN workflow performed better on Stanford (94.6%) than on Columbia (91.2%), with no overlap between the distributions over 10 trials (Figure 1). Here, both OpenCLIP and DINOv2 achieved higher accuracy on Columbia than on Stanford (Figure 3), with Columbia showing a 5% advantage for DINOv2 and 7.5% for OpenCLIP. The pattern was consistent with our predictions: the high performance on Stanford seen with the ImageNet-pretrained CNN workflow did not replicate with OpenCLIP or DINOv2. The flipped performance on Stanford vs Columbia is consistent with the interpretation that evaluating ImageNet-pretrained FGIC workflows on Stanford Dogs can inflate estimates of performance relative to benchmarks with substantially less ImageNet-1K leakage.

For Tsinghua Dogs, performance was slightly lower than for Stanford with DINOv2 (difference of 2.4%) and slightly higher with OpenCLIP (0.7%). The complex provenance of the train versus test splits of the Tsinghua dataset complicates interpretation of this pattern. However, given the 8% advantage for Stanford compared

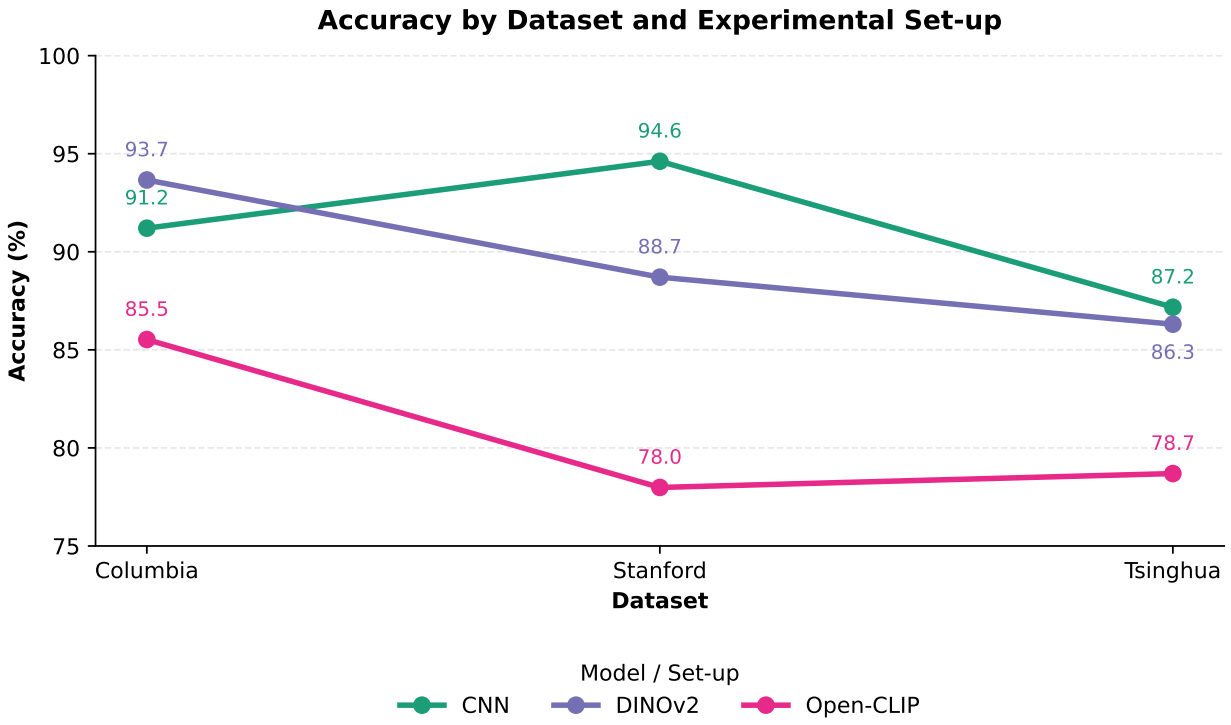


Figure 3: Comparison of accuracies achieved by the ImageNet-pretrained CNN workflow and two alternative pretrained feature extractors across the three datasets. CNN results use the 4,000-PC configuration from Experiment 1; OpenCLIP and DINOv2 results use a 640-PC configuration to accommodate the dimensionality of the OpenCLIP feature embeddings.

to Tsinghua with the CNN-based workflow, the similarity in performance across these datasets by DINOv2 and OpenCLIP is notable, suggesting again that CNN performance may be inflated when benchmarking on Stanford Dogs in particular.

7 Discussion

Taken together, these three experiments highlight the importance of considering data provenance when selecting and interpreting benchmarks. The high accuracy achieved by an ImageNet-pretrained CNN workflow on Stanford Dogs did not reflect a stable pattern of performance across datasets, even within the same task domain. Instead, the results support an alternative explanation: overlap between ImageNet-1K training data and Stanford Dogs can inflate performance estimates when using ImageNet-1K-pretrained feature extractors.

We thus demonstrate that Stanford Dogs is not an appropriate benchmark for evaluating such models, despite its widespread use for this purpose. The entire Stanford Dogs test split is contained in ImageNet-1K train. The Stanford Dogs performance advantage observed with the CNN workflow did not persist when using features extracted from OpenCLIP and DINOv2, which use different datasets for pretraining. This leakage suggests that evaluations on Stanford Dogs using ImageNet-1K-pretrained models violate a core tenet of machine learning model development: the test set used to evaluate performance must be held out. Holding out the test set allows researchers to estimate generalization beyond the data encountered during training; here, we see major shifts in between-dataset performance when replacing the supervised ImageNet-1K CNN feature extractors with models that have different pretraining provenance. Once the supervised ImageNet-1K CNN feature extractors were replaced, performance on Stanford Dogs fell in line with performance on the other datasets.

Tsinghua Dogs also shows significant leakage from ImageNet-1K and should not be treated as an independent evaluation benchmark for ImageNet-1K-pretrained models. Unlike Stanford Dogs, however, this leakage does not appear to produce a straightforward performance advantage. Across all three models, relative performance was lowest on Tsinghua Dogs among the three datasets. One possible explanation is that Stanford Dogs is entirely ImageNet-1K-derived, whereas Tsinghua Dogs contains over 50,000 unique images, many of which were collected from social media users in China (Zou et al., 2020) approximately a decade after the other two datasets were curated. Additionally, the Tsinghua validation split contains a much higher proportion of ImageNet-derived images (59.6%) than the training split does (23.8%). If the novel Tsinghua images reflect a distribution shift relative to ImageNet, any advantage from ImageNet-pretrained representations may be partially offset by variation between the training and evaluation partitions. In this sense, Tsinghua Dogs may be simultaneously contaminated by ImageNet-1K and distribution-shifted away from ImageNet's dog images.

A number of additional questions remain open. First, we investigated only leakage from the pretraining to testing phase and did not estimate the effect of duplicated images across splits within datasets. We also did not investigate why the Tsinghua Dogs curator's efforts to deduplicate their dataset with SSIM did not remove these duplicates. Second, additional analysis is needed to confirm whether distribution shift explains the patterns in performance seen with the Tsinghua dataset. Third, this analysis is limited in scope to dog breed identification and does not investigate whether similar issues arise in other FGIC domains. Finally, our DINOv2 results, especially in light of the high accuracy achieved by Khan et al. (2025) on Tsinghua Dogs, suggested that modern representation models may perform strongly on dog breed identification tasks without extensive task-specific tuning. However, we did not optimize for absolute accuracy because our goal was to compare evaluation behavior across datasets.

These analyses add context to the past 15 years of dog breed work in FGIC: provenance must be taken into account. In fact, a recent review raised concerns about the external validity of the breed labels assigned to the images in ImageNet and the dog breed FGIC benchmarking datasets (Perez et al., 2025), but here we identify a more fundamental issue. Data leakage presents a major impediment to reproducibility in machine learning (Kaufman et al., 2012; Kapoor & Narayanan, 2023). While we were able to computationally reproduce the results of Cui et al. (2024) because they followed important reproducibility practices, including making their code publicly available, the present analysis reveals a deeper reproducibility issue in FGIC that has arisen as the relationships between models, pretrained weights, and benchmark datasets have evolved over time. Stanford Dogs was presented as a novel dataset for FGIC at the first workshop on fine grained visual categorization at CVPR in 2011 (Khosla et al., 2011). It thus predates the ImageNet-1K (Russakovsky et al., 2015) dataset used to pretrain many CNNs today. While we did not investigate the provenance of these images across different ImageNet-derived datasets, we note that originally the research community was aware of the conflict: Krause et al. (2016), for example, explicitly excluded the dog classes within ImageNet from pretraining when benchmarking on Stanford Dogs. Today, this conflicting provenance is often overlooked; as a result, FGIC benchmarking studies that used Stanford Dogs along with ImageNet pretraining cannot be assumed to generalize outside of this specific context.

8 Conclusion

We argue that researchers should avoid using Stanford Dogs or Tsinghua Dogs as independent benchmarks for models pretrained on ImageNet-1K. Stanford Dogs is fully compromised for this purpose because its entire test split is contained in ImageNet-1K train. Tsinghua Dogs is also inappropriate as an independent evaluation benchmark because a majority of its validation split is contained in ImageNet-1K train, and because ImageNet-1K overlap differs substantially between its training and validation splits. More generally, the anomalies in data provenance identified here span several categories of data leakage, as defined by Kapoor & Narayanan (2023), and demonstrate that data leakage in FGIC is not merely a theoretical concern. Benchmark datasets should therefore be accompanied by provenance documentation, split manifests, and image-level hashes that allow future researchers to audit overlap with common pretraining corpora. Such precautions are necessary to ensure that benchmarks remain instruments of accountability.

References

Hritam Basak, Rohit Kundu, Sukanta Chakraborty, and Nibaran Das. Cervical cytology classification using PCA and GWO enhanced deep features selection. *SN Computer Science*, 2(5):369, 2021.

Punyanuch Borwarnginn, Kittikhun Thongkanchorn, Sarattha Kanchanapreechakorn, and Worapan Kusakunniran. Breakthrough conventional based approach for dog breed classification using CNN with transfer learning. In *2019 11th International Conference on Information Technology and Electrical Engineering (ICITEE)*, pp. 1–5. IEEE, 2019.

Ying Cui, Bixia Tang, Gangao Wu, Lun Li, Xin Zhang, Zhenglin Du, and Wenming Zhao. Classification of dog breeds using convolutional neural network models and support vector machine. *Bioengineering*, 11 (11):1157, 2024.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 248–255, 2009.

Google. Inception V3, 2020a. URL `https://www.kaggle.com/models/google/inception-v3`. Kaggle Models. Accessed 8 June 2026.

Google. PNASNet, 2020b. URL `https://www.kaggle.com/models/google/pnasnet`. Kaggle Models. Accessed 8 June 2026.

Google. Inception ResNet V2, 2020c. URL `https://www.kaggle.com/models/google/inception-resnet-v2`. Kaggle Models. Accessed 8 June 2026.

Google. NASNet, 2020d. URL `https://www.kaggle.com/models/google/nasnet`. Kaggle Models. Accessed 8 June 2026.

Parthesh Haswar, Aditya Iyer, Prachiti Godhane, Amisha Jadhav, and Ankita Malhotra. Dog breed identification and age detection using neural networks. *International Journal of Innovative Research in Technology*, 8(11):125–131, 2022.

J. D. Hunter. Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

Gabriel Ilharco, Mitchell Wortsman, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, et al. OpenCLIP. Zenodo, 2021.

Sayash Kapoor and Arvind Narayanan. Leakage and the reproducibility crisis in machine-learning-based science. *Patterns*, 4(9), 2023.

Shachar Kaufman, Saharon Rosset, Claudia Perlich, and Ori Stitelman. Leakage in data mining: Formulation, detection, and avoidance. *ACM Transactions on Knowledge Discovery from Data*, 6(4), December 2012.

Noman Khan, Mi Young Lee, and Jakyoung Min. Head-body guided deep learning framework for dog breed recognition. *Computers, Materials & Continua*, 85(2):2935–2958, 2025.

Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Fei-Fei Li. Novel dataset for fine-grained image categorization: Stanford dogs. In *Proceedings of the CVPR Workshop on Fine-Grained Visual Categorization*, volume 1, 2011.

Jonathan Krause, Benjamin Sapp, Andrew Howard, Howard Zhou, Alexander Toshev, Tom Duerig, James Philbin, and Li Fei-Fei. The unreasonable effectiveness of noisy data for fine-grained recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 301–320, September 2016.

Chenxi Liu, Barret Zoph, Maxim Neumann, Jonathon Shlens, Wei Hua, Li-Jia Li, Li Fei-Fei, Alan Yuille, Jonathan Huang, and Kevin Murphy. Progressive neural architecture search. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 19–34, September 2018.

Jiongxin Liu, Angjoo Kanazawa, David Jacobs, and Peter Belhumeur. Dog breed classification using part localization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 172–185, 2012.

George A. Miller. WordNet: A lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

Seyedali Mirjalili, Seyed Mohammad Mirjalili, and Andrew Lewis. Grey wolf optimizer. *Advances in Engineering Software*, 69:46–61, 2014.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

Glenvelis Perez, Yixuan He, Zihan Lyu, Yilin Chen, Nicholas R. Howe, and Halie M. Rando. Standardizing canine breed data in veterinary records is challenging, but computer vision offers an alternative perspective on breed assignment. *American Journal of Veterinary Research*, 86(S1):S38–S45, 2025.

Python Software Foundation. hashlib — secure hashes and message digests, 2026. URL `https://docs.python.org/3/library/hashlib.html`. Python Standard Library documentation. Accessed June 1, 2026.

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 *Proceedings of the International Conference on Machine Learning (ICML)*, Proceedings of Machine Learning Research, pp. 8748–8763, 2021.

Zalán Ráduly, Csaba Sulyok, Zsolt Vadászi, and Attila Zölde. Dog breed identification using deep learning. In *2018 IEEE 16th International Symposium on Intelligent Systems and Informatics (SISY)*, pp. 271–276. IEEE, 2018.

Tal Ridnik, Emanuel Ben-Baruch, Asaf Noy, and Lihi Zelnik-Manor. ImageNet-21K pretraining for the masses. In *Advances in Neural Information Processing Systems Datasets and Benchmarks Track*, 2021.

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. *International Journal of Computer Vision*, 115(3):211–252, 2015.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa 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 *Advances in Neural Information Processing Systems*, volume 35, pp. 25278–25294, 2022.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the Inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826, 2016.

Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A. Alemi. Inception-v4, Inception-ResNet and the impact of residual connections on learning. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pp. 4278–4284, 2017.

Xiaolong Wang, Vincent Ly, Scott Sorensen, and Chandra Kambhamettu. Dog breed classification via landmarks. In *2014 IEEE International Conference on Image Processing (ICIP)*, pp. 5237–5241. IEEE, 2014.

Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V. Le. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Ding-Nan Zou, Song-Hai Zhang, Tai-Jiang Mu, and Min Zhang. A new dataset of dog breed images and a benchmark for fine-grained classification. *Computational Visual Media*, 6(4):477–487, 2020.

A Appendix: Per-Dataset Class Counts

In order to ascertain the number of images in each class across all splits, we used the fact that the datasets are all structured as one class per folder and used `rglob` to enumerate all of the files in each directory. We then counted the number of files ending in either `.jpg` or `.jpeg`. These extensions were verified as sufficient using the Linux command `sed 's/.*\.//' | uniq` to identify the file extensions present in each class folder across all splits. For Columbia Dogs, each class-level directory name appears once in the `train` split, once in the `valid` split, and once in the `test` split. For Stanford and Tsinghua, the splits were identified by mapping image file names against metadata text files from the original datasets.

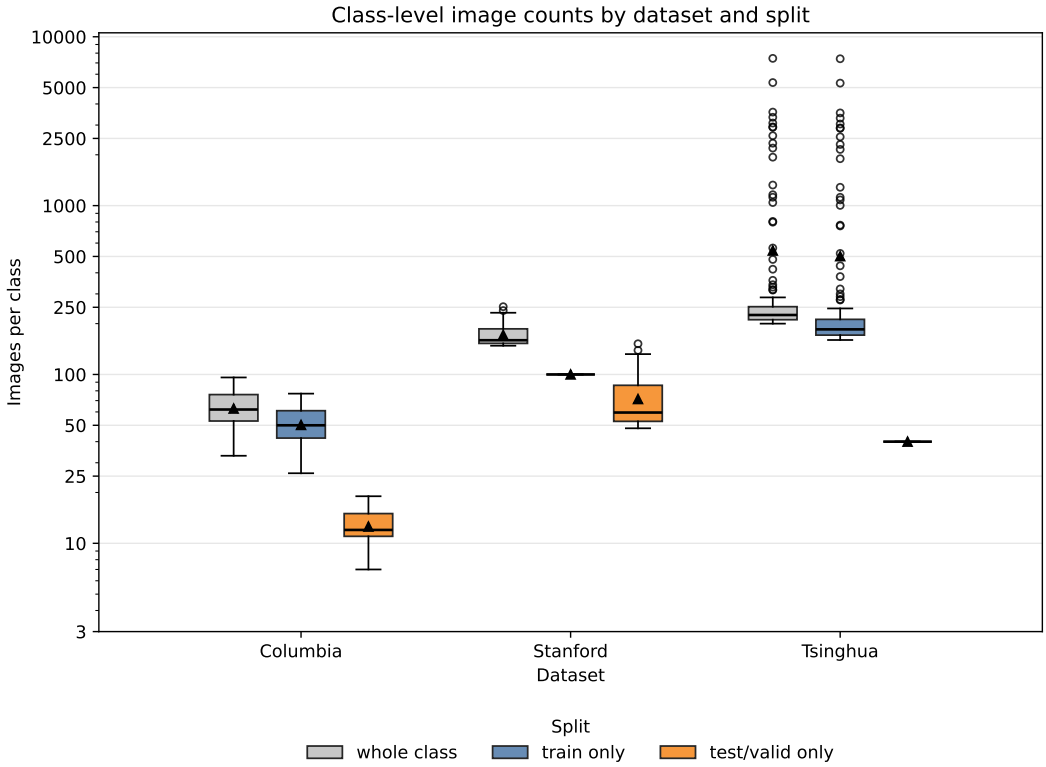


Figure 4: Class-level image counts by dataset and split. The y-axis is visualized on a log scale.

The results of the per-class image count analysis are visualized (Figure 4) and reported (Table 4) here. These results were used as the basis for the decision to compare 25/6 train/test splits across all datasets in Experiment 1, as the minimum number of images included in any train split, across all datasets and all classes, was 26, and the minimum number of images included in any test split across all classes was 7 (when combining the Columbia Dogs `test` and `valid` splits). By selecting 25/6, we were able to ensure that no splits were deterministic across samples.

Table 4: Statistics of image counts per class, by dataset and split.

Dataset	**Split**	**Num. Classes**	**Mean**	**SD**	**Min.**	**Max.**
Columbia	Whole class	133	62.80	14.85	33	96
Columbia	Train only	133	50.23	11.86	26	77
Columbia	Test only	133	6.29	1.71	3	10
Columbia	Valid only	133	6.28	1.35	4	9
Columbia	Test+valid	133	12.56	3.01	7	19
Stanford	Whole class	120	171.50	23.22	148	252
Stanford	Train only	120	100.00	0.00	100	100
Stanford	Test only	120	71.50	23.22	48	152
Tsinghua	Whole class	130	541.75	1005.70	200	7449
Tsinghua	Train only	130	501.75	1005.70	160	7409
Tsinghua	Test only	130	40.00	0.00	40	40